T.C.
MARMARA ÜNİVERSİTESİ
TÜRKİYAT ARAŞTIRMALARI ENSTİTÜSÜ
BİLGİ VE BELGE YÖNETİMİ ANABİLİM DALI

DİJİTAL ARŞİV ÖRÜNTÜLERİNİN HESAPLAMALI ARŞİV BİLİMİ İLE İNCELENMESİ

DOKTORA TEZİ

MEHMET OYTUN CİBAROĞLU

İSTANBUL, 2025

T.C.
MARMARA ÜNİVERSİTESİ
TÜRKİYAT ARAŞTIRMALARI ENSTİTÜSÜ
BİLGİ VE BELGE YÖNETİMİ ANABİLİM DALI

DİJİTAL ARŞİV ÖRÜNTÜLERİNİN HESAPLAMALI ARŞİV BİLİMİ İLE İNCELENMESİ

DOKTORA TEZİ

MEHMET OYTUN CİBAROĞLU

TEZ DANIŞMANI: DOÇ. DR. BAHATTİN YALÇINKAYA

İSTANBUL, 2025

İÇİNDEKİLER

İÇİNDEKİLER ..... **I**
ÖNSÖZ ..... **IV**
ÖZET ..... **VI**
ABSTRACT ..... **VIII**
ŞEKİLLER LİSTESİ ..... **X**
TABLOLAR LİSTESİ ..... **XV**
SEMBOLLER LİSTESİ ..... **XVII**
EKLER LİSTESİ ..... **XVIII**
KISALTMALAR LİSTESİ ..... **XIX**
GİRİŞ **1**
BİRİNCİ BÖLÜM **5**
1. TEZİN AMACI, HİPOTEZLERİ, YÖNTEMİ VE KAPSAMI **5**
1.1. Konu ve Önem 5
1.2. Amaçlar ve Hedefler 8
1.3. Özgün Değer 9
1.4. Problem İfadesi, Araştırma Soruları ve Hipotezler 10
1.5. Yöntem 11
1.6. Kapsam 12
1.7. Tezin Düzeni 12
1.8. Kısıtlar 12
1.9. Yararlanılan Kaynaklar 13
İKİNCİ BÖLÜM **17**
2. ARŞİV BİLİMİ VE YAPAY ZEKÂ GENEL LİTERATÜR TARAMASI **17**
2.1. Bilim Kavramının Genel Değerlendirilmesi 17
2.2. Hipotez ve Teori Kavramlarının Genel Değerlendirilmesi 21
2.3. Arşiv Biliminin Genel Değerlendirilmesi 25
2.3.1. Arşiv Teorisi ve Teorinin Dönüşümü 29
2.3.1.1. Provenans İlkesi 36
2.3.1.2. Fonlara Saygı İlkesi 45
2.3.1.3. Orijinal Düzen İlkesi 46
2.4. Arşivsel Düşünme 49
2.5. Hesaplamalı Düşünme 55
2.6. Hesaplamalı Arşiv Bilimi ve Çalışmaları 59
2.7. Modern Arşivciliğin Yeni Paradigması Olarak Hesaplamalı Arşiv Bilimi 72
2.8. Yapay Zekâ 74
2.8.1. Makine Öğrenmesi 84
2.8.1.1. Denetimli Öğrenme 85
2.8.1.1.1. Doğrusal Regresyon 88
2.8.1.1.2. Lojistik Regresyon 90
2.8.1.1.3. Karar Ağaçları 92
2.8.1.1.4. Destek Vektör Makineleri 98
2.8.1.1.5. k-En Yakın Komşu 111
2.8.1.1.6. Naive Bayes 116
2.8.1.1.7. Rastgele Orman 119
2.8.1.1.8. Yapay Sinir Ağları 122
2.8.1.1.8.1. Tek Katmanlı Algılayıcı 124
2.8.1.1.8.2. Çok Katmanlı Algılayıcı 128
2.8.1.2. Denetimsiz Öğrenme 131
2.8.1.2.1. k-Ortalamalar 134
2.8.1.2.2. Hiyerarşik Kümeleme 135

2.8.1.2.3. DBSCAN 139
2.8.1.2.4. Temel Bileşen Analizi 141
2.8.1.2.5. Birliktelik Kuralı Öğrenmesi 143
2.8.1.3. Pekiştirmeli Öğrenme 144
2.8.2. Derin Öğrenme 146
2.8.2.1. Aktivasyon Fonksiyonları 148
2.8.2.1.1. Doğrusal Aktivasyon Fonksiyonu 148
2.8.2.1.2. Sigmoid Aktivasyon Fonksiyonu 149
2.8.2.1.3. Hiperbolik Tanjant Aktivasyon Fonksiyonu .. 150
2.8.2.1.4. ReLU Aktivasyon Fonksiyonu 151
2.8.2.1.5. Leaky ReLU Aktivasyon Fonksiyonu 153
2.8.2.1.6. Parametrik ReLU Aktivasyon Fonksiyonu 153
2.8.2.1.7. ELU Aktivasyon Fonksiyonu 154
2.8.2.1.8. GELU Aktivasyon Fonksiyonu 155
2.8.2.1.9. Softmax Aktivasyon Fonksiyonu 157
2.8.2.2. Derin Öğrenmede Düzenlileştirme Teknikleri 158
2.8.2.2.1. Veri Artırma 159
2.8.2.2.2. Sentetik Veri Üretme 160
2.8.2.2.3. Gürültü Ekleme 160
2.8.2.2.4. Bırakma 161
2.8.2.2.5. L1 ve L2 Ağırlık Azalması 161
2.8.2.2.6. Seyrek-Yoğun-Seyrek Eğitim 162
2.8.2.2.7. Erken Durdurma 163
2.8.2.3. Bias-Varyans İlişkisi 165
2.8.2.4. Hiperparameteler 167
2.8.2.5. Evrişimli Sinir Ağları 168
2.8.2.5.1. Giriş Katmanı 170
2.8.2.5.2. Evrişim Katmanı 171
2.8.2.5.2.1. Kenar Algılama 172
2.8.2.5.2.2. Piksel Ekleme 175
2.8.2.5.2.3. Adım Kaydırma 177
2.8.2.5.3. Havuzlama Katmanı 178
2.8.2.5.4. Aktivasyon Katmanı 182
2.8.2.5.5. Tam Bağlantılı Katman 182
2.8.2.5.6. Çıkış Katmanı 183
2.8.2.5.7. Düzleştirme İşlemi 183
2.8.2.6. Evrişimli Sinir Ağları Modelleri 184
2.8.2.6.1. LeNET 184
2.8.2.6.2. AlexNET 185
2.8.2.6.3. VGG16 186
2.8.2.6.4. ResNET 188
2.8.2.6.5. EfficientNET 189
2.8.2.7. Özyinelemeli Sinir Ağları 190
2.8.2.8. Transfer Öğrenmesi 193
ÜÇÜNCÜ BÖLÜM 195
3. UYGULAMA FAZLARINA YÖNELİK METODOLOJİLER 195
3.1. Veri Setleri 195
3.1.1. Amerika Birleşik Devletleri İç Hat Uçuş Veri Seti 195
3.1.2. Hava Görüntüsü Veri Seti (Aerial Image Dataset-AID) 197
3.2. Donanım 198
3.3. Platform ve Araçlar 198
3.4. Araştırmanın Modeli 201

3.5. Makine Öğrenmesi ve Derin Öğrenme Model Değerlendirme Metrikleri... 202
3.5.1. Sınıflandırma Metrikleri ... 202
3.5.2. Regresyon Metrikleri ... 205
3.5.3. Kümeleme Metrikleri ... 207
3.5.4. Boyut Azaltma Metrikleri ... 208
DÖRDÜNCÜ BÖLÜM ... 210
4. BİRİNCİ FAZ (MAKİNE ÖĞRENMESİ) ANALİZLER VE BULGULAR ... 210
4.1. Veri Ön İşleme ... 210
4.2. Temel İstatistiki Göstergeler ... 220
4.3. Makine Öğrenmesi Uygulamaları ... 231
4.3.1. Denetimli Öğrenme Uygulamaları ... 231
4.3.1.1 Regresyon Analizleri ... 231
4.3.1.2. Sınıflandırma Analizleri ... 244
BEŞİNCİ BÖLÜM ... 265
5. İKİNCİ FAZ (DERİN ÖĞRENME) ANALİZLER VE BULGULAR ... 265
5.1. Evrişimli Sinir Ağları Uygulaması ... 265
5.1.1. Görüntü Boyutu Bazlı Karşılaştırmalar ... 271
5.1.2. Optimizasyon Algoritması ve Görüntü Boyutu Bazlı Karşılaştırmalar ... 273
5.1.3. Eğitim Kümesi Bazlı Karşılaştırmalar ... 275
5.1.4. Öğrenme Oranı Bazlı Karşılaştırmalar ... 296
5.2. Transfer Öğrenmesi ... 304
5.2.1. Temel Modeller ... 304
5.2.2. Optimize Edilmiş ve Düzenlileştirilmiş Modeller ... 316
ALTINCI BÖLÜM ... 335
6. PROVENANS VERİ HATTI VE ARCHIVEMATICA ... 335
6.1. Provenans Veri Hattı ... 335
6.2. Archivematica ... 339
YEDİNCİ BÖLÜM ... 348
7. SONUÇ, TARTIŞMA VE DEĞERLENDİRME ... 348
7.1. Sonuç ... 348
7.1.1. Makine Öğrenmesi ... 348
7.1.2. Evrişimli Sinir Ağları ... 349
7.1.3. Transfer Öğrenmesi ... 350
7.1.4. Dijital Provenans ve Archivematica Uygulaması ... 351
7.2. Tartışma ... 351
7.2.1. Makine Öğrenmesi Analizleri ... 351
7.2.2. Transfer Öğrenmesi ve Evrişimli Sinir Ağları Analizleri ... 352
7.2.3. Dijital Provenans ve Archivematica Uygulaması ... 355
7.3. Değerlendirme ... 356
KAYNAKÇA ... 360
EKLER ... 423

ÖNSÖZ

Bilgi ve belge yönetimi alanında güncel bir yaklaşım olarak ortaya çıkan hesaplamalı arşiv bilimi, sadece tarihi değere sahip belge ve materyalleri değil, arşivsel dijital veri bağlamında da kabul edilebilecek büyük veri setlerini makine öğrenmesi ve derin öğrenme yöntemleriyle inceleyerek arşivsel süreçlerin daha şeffaf, izlenebilir ve yeniden üretilebilir hale gelmesine katkı sağlamayı da amaçlamaktadır. Böyle bir amaç, aynı zamanda bilgisayar bilimi literatüründe de popüler bir kavram olan provenansa yönelik yaklaşımların entegrasyonunu da sağlama potansiyeline sahiptir. Arşiv bilimi ile bilgisayar biliminin ortak bir kavramı olarak değerlendirilmesi gereken provenans, her iki bilim dalındaki yöntem ve teknikler aracılığıyla disiplinlerüstü bir yaklaşım oluşturma hedefinde olan hesaplamalı arşiv bilimini alanda oldukça önemli bir konuma taşımıştır.

Bu bağlamda, yapay zekâ yöntem ve tekniklerinin bilgi ve belge yönetimi alanına uygulanması bir süreçler bütünü olarak değerlendirilmeli; retrospektif ve prospektif şekilde tüm uygulama süreçleri, dijital ortamda küresel standartlara uygun koşullarda nitelikli bir veri havuzu biçiminde saklanabilmelidir.

Akademik çalışmalarda yönlendirici fikirleri ve yol göstericiliği ile doktora tez konusunu belirlememi sağlayan değerli tez danışmanım Doç. Dr. Bahattin YALÇINKAYA'ya destekleri için minnettarım.

Teze değerli katkıları için Doç. Dr. Buket DOĞAN'a ve Prof. Dr. Niyazi ÇİÇEK'e çok teşekkür ederim. Ayrıca, Doç. Dr. Elif YILMAZ ŞENTÜRK ve Doç. Dr. Lale ÖZDEMİR ŞAHİN ile Dr. Öğr. Üyesi Banu Fulya YILDIRIM'a destekleri için çok teşekkür ederim.

Analiz süreçlerinde fikir alışverişinde bulunduğum ve yapıcı önerileriyle uygulamaların ve tezin son halini almasını sağlayan Dr. Öğr. Üyesi Muhammet Emin GEDİKLİ'ye ve Öğr. Gör. Dr. Özhan SAĞLIK'a çok teşekkür ederim.

Akılcı önerileriyle tezimi yazarken beni hep yönlendiren ve konsantrasyonumu üst düzeyde tutan Arş. Gör. Nilay KAÇAR'a minnettarım. İyi ki varsın.

Hayatımın her anında yanımda olan ve destekleri ile motivasyonumu kuvvetlendiren aileme çok teşekkür ederim.

Bu doktora tezi, 2022 yılının Ağustos ayında aramızdan ayrılan canımız Layka'ya adanmıştır.

Mehmet Oytun CİBAROĞLU

Layka’ya…

ÖZET

Arşiv bilimi; büyük veri ve yapay zekâ odaklı oldukça geniş çaplı bir dönüşümün içinde yer almaya başlamıştır. Bu doğrultuda hesaplamalı arşiv bilimi adı verilen yeni bir paradigma ortaya çıkmıştır. Bu alan, bilgisayar bilimi ile arşiv biliminin kolektif bilgisini birleştirerek dijital arşiv materyallerinin analizi, uzun süreli korunması ve erişimi için hesaplamalı yöntemler sunmaktadır. Bu tezde, hesaplamalı arşiv biliminin temel yaklaşımları benimsenerek; yapılandırılmış (TSV) ve yapılandırılmamış (uydu tarafından çekilen yer görüntüleri) dijital arşiv veri setleri üzerinde yapay zekâ tabanlı analizler gerçekleştirilmiş, hesaplamalı süreçlere ait provenans verisi oluşturulmuş ve tüm çıktılar, Archivematica programı aracılığıyla uzun süreli dijital korumaya alınmıştır. Çalışmanın ilk aşamasında, 1990-2009 yılları arasına ait 3,6 milyon örnek ve 11 öznitelikten oluşan ABD iç hat uçuş verileri analiz edilmiştir. Bu veri setinin seçilmesinin nedeni ise dijital arşiv verisi niteliğinde ve açık veri olmasıdır. Kapsamlı önişleme adımları (eksik değerlerin KNNImputer ile doldurulması, log dönüşümü, IQR yaklaşımı ile aykırı değerlerin giderilmesi, kırpma ve standardizasyon) sonrasında regresyon ve sınıflandırma görevleri gerçekleştirilmiştir. Yolcu sayısına ilişkin en başarılı regresyon modeli, çoklu doğrusal regresyonda %91,97 R^2 ve yapay sinir ağı ile de %99,90 R^2 elde etmiştir. Sınıflandırma analizlerinde mesafe kategorilerine göre %99,90 doğrulukla yığınlama modeli, uçuş başına yolcu sayısı sınıflandırmasında ise yapay sinir ağı modeli %99,59 doğrulukla en başarılı sonuçları vermiştir. Derin öğrenme analizleri kapsamında ise, 30 sınıflı AID yer görüntüleri veri seti, temel bir evrişimli sinir ağı (CNN) modeli oluşturularak işlenmiş; çeşitli giriş boyutları, optimizasyon algoritması türleri ve öğrenme oranları bazlı karşılaştırmalar yapılmıştır. Temel CNN modelinde en iyi sonuç, Adamax optimizasyon algoritması ile 250×250 giriş boyutu ve %91,40 doğrulukla elde edilmiştir. Veri artırımı ve hiperparametre optimizasyonu da model başarısına oldukça katkı sağlamıştır. Transfer öğrenmesi kapsamında ise EfficientNet mimarisinin (B0-B3, V2B0-V2B3) 8 farklı varyantı kullanılmış, öncelikle ağırlıksız olarak temel modeller eğitilmiş, ardından RandomSearch Tuner tekniği ile en iyi hiperparametre aramaları yapılmıştır. Optimize edilen modellerde B3, V2B2 ve V2B3 varyantları %97,20 doğruluk ve sırasıyla 0,9713, 0,9716 ve 0,9718 F1 skorları ile en iyi sonucu vermiştir. Tüm hesaplamalı süreçler, PROV uyumlu JSON formatında provenans verisi olarak yapılandırılmış; analizlere dair işlem bilgileri, kullanılan yöntemler, hiperparametreler ve çıktılar zaman damgalı şekilde kayıt altına alınmıştır. Örnek olarak seçilen iki analiz süreci, BagIt kütüphanesi ile paketlenmiş, orijinal ve türev veri setleriyle birlikte Archivematica sistemine yüklenmiş ve burada SIP, AIP ve DIP

paketlerine dönüştürülerek uzun süreli koruma sağlanmıştır. Ayrıca genel işlem akışı, PROV-ML düğümleri ile diyagram olarak gösterilmiştir. Tezin alana temel katkısı; hesaplamalı yöntem ve tekniklerle üretilen veri ve bilgilerin, tüm girdi-çıktı boyutlarıyla birlikte uzun süreli korumaya alınabilecek nitelikli dijital arşiv nesnelerine (nitelikli veri havuzu) dönüştürülebileceğidir. Bu katkıya ek olarak yapay zekâ tabanlı süreçlere dair arşivsel provenansın oluşturulması, tüm hesaplamalı süreçlerin izlenebilirliği ve şeffaflığını da sağlamaktadır. Tez, bu yönüyle alanda dijital koruma, süreç belgeleme ve veri temelli karar destek mekanizmalarına katkı sağlamakta ve bu bağlamda kurumlar, araştırmacılar ve kullanıcılar için yeni bir arşivsel değerin üretilebileceğini savunmaktadır.

Anahtar Kelimeler: Hesaplamalı Arşiv Bilimi, Yapay Zekâ, Dijital Provenans, Makine Öğrenmesi, Derin Öğrenme, Transfer Öğrenmesi

ABSTRACT

Archival science has increasingly become part of a broad transformation driven by big data and artificial intelligence. In this context, a new paradigm known as computational archival science has emerged. This field integrates the collective knowledge of computer science and archival science to offer computational methods for the analysis, long-term preservation and access of digital archival materials. In this thesis, core approaches of computational archival science are adopted to conduct artificial intelligence-based analyses on both structured (TSV) and unstructured (satellite-captured aerial imagery) digital archival datasets. Provenance information related to the computational processes is generated, and all outputs are preserved through Archivematica, a digital preservation system. In the first stage of the study, a U.S. domestic flight dataset containing 3,6 million records and 11 features from 1990 to 2009 is analyzed. This dataset was selected due to its nature as a digital archival source and its public availability. After extensive preprocessing steps (missing value imputation with KNNImputer, logarithmic transformation, outlier removal via the IQR method, clipping and standardization) both regression and classification tasks were performed. For predicting the number of passengers, multiple linear regression yielded an R^2 of 91,97%, while the artificial neural network model achieved 99,90% R^2. In classification analyses, the stacking model achieved 99,90% accuracy in predicting distance categories, while the artificial neural network model reached 99,59% accuracy in classifying passenger numbers per flight. For deep learning analyses, the 30-class AID (Aerial Image Dataset) was processed using a baseline convolutional neural network (CNN) model. Various input dimensions, optimizasyon algoritması types and learning rates were compared. The best result in the CNN model was obtained using the Adamax optimizasyon algoritması, with an input size of 250×250 and an accuracy of 91,40%. Data augmentation and hyperparameter optimization significantly contributed to model performance. As part of transfer learning, eight variants of the EfficientNet architecture (B0-B3 and V2B0-V2B3) were utilized. Initially, base models were trained without preloaded weights, followed by hyperparameter tuning using the RandomSearch Tuner. In the optimized models, the B3, V2B2, and V2B3 variants achieved the best performance, with an accuracy of 97.20% and F1 scores of 0,9713, 0,9716, and 0,9718, respectively. All computational processes were recorded as provenance data in PROV-compliant JSON format. Information regarding analytical methods, hyperparameters, outputs, and timestamps were systematically recorded. Two example analysis workflows were packaged using the BagIt library, uploaded to the Archivematica system together with both

original and derived datasets, and transformed into SIP, AIP, and DIP packages for long-term preservation. In addition, the overall process flow was visualized as a diagram using PROV-ML nodes. The fundamental contribution of this thesis is the demonstration that data and information generated through computational methods and techniques can be transformed - together with all their input-output dimensions- into qualified digital archival objects (qualified data repository) that are suitable for long-term preservation. In addition to this, the creation of archival provenance for AI-based processes ensures the traceability and transparency of all computational workflows. In this respect, the thesis contributes to digital preservation, process documentation, and data-driven decision-making mechanisms in the field, and argues that a new archival value can thereby be produced for institutions, researchers, and users.

Keywords: Computational Archival Science, Artificial Intelligence, Digital Provenance, Machine Learning, Deep Learning, Transfer Learning

ŞEKİLLER LİSTESİ

Şekil 1. Mashup Provenans Gösteriminde Yönlendirilmiş Çevrimsiz Çizge Yapısı 40
Şekil 2. Arşivsel Düşünmenin Dört Yönü 51
Şekil 3. Hesaplamalı Düşünmenin Dört Bileşeni 59
Şekil 4. Yapay Zekânın Gelişiminde Önemli Evreler 82
Şekil 5. Makine Öğrenmesi Türleri 85
Şekil 6. Denetimli Öğrenme Algoritmaları 86
Şekil 7. Denetimli Öğrenme Akış Diyagramı 87
Şekil 8. Basit Doğrusal Regresyon Grafiği 88
Şekil 9. Çok Değişkenli Doğrusal Regresyon Grafiği 90
Şekil 10. Lojistik Regresyon Grafiği 91
Şekil 11. Karar Ağaçları Grafiği 94
Şekil 12. Karar Sınırları Grafiği 95
Şekil 13. Aşırı Öğrenmiş ve Budanmış Regresyon Ağaçları Grafiği 97
Şekil 14. Aşırı Öğrenmiş ve Budanmış Regresyon Ağaçlarında Tahmin 98
Şekil 15. Doğrusal Destek Vektör Makinesi Sınıflandırma Grafiği 99
Şekil 16. Sert Marjlı Destek Vektör Makinesi Sınıflandırma Grafiği 102
Şekil 17. Yumuşak Marjlı Destek Vektör Makinesi Sınıflandırma Grafiği 103
Şekil 18. Polinomal Destek Vektör Makinesi Sınıflandırma Grafiği 105
Şekil 19. 3 Boyutlu Polinomal Çekirdek Fonksiyonu Dönüşümü 105
Şekil 20. Gamma Parametreli RBF Çekirdek Fonksiyonu 107
Şekil 21. Sigmoid Çekirdek Fonksiyonu Sınıflandırma Grafiği 108
Şekil 22. Doğrusal Destek Vektör Regresyon Grafiği 110
Şekil 23. Doğrusal Olmayan Destek Vektör Regresyon Grafiği 111
Şekil 24. k-En Yakın Komşu Algoritması Sınıflandırma Grafiği 113
Şekil 25. k-En Yakın Komşu Algoritması Regresyon Grafiği 114
Şekil 26. k-En Yakın Komşu Algoritması Ters Mesafe Ağırlıklandırması Grafiği 115
Şekil 27. Gauss Naive Bayes Sınıflandırma Grafiği 117
Şekil 28. Çok Terimli Naive Bayes 1 118
Şekil 29. Çok Terimli Naive Bayes Grafiği 2 119
Şekil 30. Rastgele Orman Karar Sınırları Grafiği 120
Şekil 31. Rastgele Orman Ağaç Yapısı Grafiği 121
Şekil 32. Ağaç Yapısının Değiştiği Sınıflandırma Grafiği 121
Şekil 33. Rastgele Orman Regresyon Grafiği 122
Şekil 34. Canlılardaki Sinir Hücrelerinin Biyolojik Görünümü 123
Şekil 35. Sinir Hücrelerinin Matematiksel Modellenmesi 123
Şekil 36. Tek Katmanlı Algılayıcı Modeli 125
Şekil 37. Düzlemde Sınıf Ayrımı 125
Şekil 38. İkiden Fazla Sınıf Olduğunda Sınırların Belirlenmesi 126
Şekil 39. Çok Katmanlı Algılayıcı Mimarisi 129
Şekil 40. Denetimsiz Öğrenme Türleri 132
Şekil 41. Denetimsiz Öğrenme Akış Diyagramı 133
Şekil 42. k-Ortalamalar Grafiği 135
Şekil 43. Hiyerarşik Kümeleme Grafiği 138
Şekil 44. DBSCAN Kümeleme Grafiği 141
Şekil 45. Temel Bileşen Analizi Boyut Azaltma Grafiği 143
Şekil 46. Yapay Sinir Ağı Yapısı 147
Şekil 47. Doğrusal Aktivasyon Fonksiyonu 149
Şekil 48. Sigmoid Aktivasyon Fonksiyonu 150
Şekil 49. Hiperbolik Tanjant Aktivasyon Fonksiyonu 151
Şekil 50. ReLU Aktivasyon Fonksiyonu 152

Şekil 51. Leaky ReLU Aktivasyon Fonksiyonu 153
Şekil 52. PReLU Aktivasyon Fonksiyonu 154
Şekil 53. ELU Aktivasyon Fonksiyonu 155
Şekil 54. GELU Aktivasyon Fonksiyonu 156
Şekil 55. Softmax Aktivasyon Fonksiyonu 157
Şekil 56. Aşırı Öğrenme ve Eksik Öğrenme 159
Şekil 57. Bırakma 161
Şekil 58. Yoğun-Seyrek-Yoğun Eğitim 163
Şekil 59. Erken Durdurma Grafiği 164
Şekil 60. Bias-Varyans İlişkisi 1 166
Şekil 61. Bias-Varyans İlişkisi 2 167
Şekil 62. Evrişimli Sinir Ağları Çalışma Prensibi 168
Şekil 63. Çok Katmanlı Evrişim İşlemi 169
Şekil 64. Giriş Katmanı İçin Örnek Görüntüler 171
Şekil 65. Modifiye Edilmiş Giriş Görüntüleri 171
Şekil 66. Kenar Tespit İşlemi 173
Şekil 67. Piksel Ekleme İşlemi 176
Şekil 68. Adım Kaydırma İşlemi 177
Şekil 69. Havuzlama 1 - Orijinal Matris 179
Şekil 70. Havuzlama 2 - Maksimum Havuzlama 179
Şekil 71. Havuzlama 3 - Ortalama Havuzlama 180
Şekil 72. Havuzlama 4 - Global Havuzlama 181
Şekil 73. LeNET Mimarisi 185
Şekil 74. AlexNET Mimarisi 186
Şekil 75. VGG16 Mimarisi 187
Şekil 76. ResNET Mimarisi 188
Şekil 77. EfficientNET Mimarisi 189
Şekil 78. Temel RNN Mimarisi 192
Şekil 79. Makine Öğrenmesi ve Transfer Öğrenmesi Farklılıkları 194
Şekil 80. Tezin Kavramsal Modeli 201
Şekil 81. AUC-ROC Eğrisi 204
Şekil 82. Orijinal Veri Setinin İlk 5 Satır Görüntüsü 210
Şekil 83. YAML İçerisindeki Metaveri Bilgileri 210
Şekil 84. Metaveri Ekleme 1 211
Şekil 85. Metaveri Ekleme 2 211
Şekil 86. Metaveri Adlarının Türkçe'ye Çevrilmiş Hali 211
Şekil 87. Veri Seti 1'in Son Görünümü 211
Şekil 88. Veri Seti 1 Temel İstatistiki Bilgiler 212
Şekil 89. Veri Seti 1'de Bulunan Veri Tipleri 212
Şekil 90. Veri Seti 1'de Bulunan 0 Değerleri 213
Şekil 91. Veri Seti 1 Histogram ve Kutu Grafiği 215
Şekil 92. Veri Seti 1 Çarpıklık ve Basıklık Grafiği 216
Şekil 93. Veri Seti 1 Çarpıklık ve Basıklık Grafiği (Kontrol) 218
Şekil 94. Veri Seti 1 Histogram ve Kutu Grafiği (Kontrol) 219
Şekil 95. Veri Seti 1 Korelasyon Matrisi 220
Şekil 96. Eyalet Bazlı En Çok Uçuş Yapılan Varış Noktaları 222
Şekil 97. En Çok Uçuş Yapılan Varış Havaalanları 222
Şekil 98. En Az Uçuş Yapılan Varış Havaalanları 223
Şekil 99. Aylık Bazdaki Uçuş Sayıları 223
Şekil 100. Yıllık Bazdaki Uçuş Sayıları 224
Şekil 101. Varış Eyaletlerine Göre Yolcu Sayıları 224
Şekil 102. Farklı Şehirler Arası En Uzun Uçuş Mesafesi 225

Şekil 103. Farklı Şehirler Arası En Kısa Uçuş Mesafesi 225
Şekil 104. En Fazla Uçağın Kalktığı Havaalanları..... 226
Şekil 105. En Az Uçağın Kalktığı Havaalanları..... 226
Şekil 106. En Fazla Uçağın Kalktığı Şehir ve Eyaletleri 227
Şekil 107. Yıl Bazlı Yolcu Sayıları ve Toplam Yolcu Sayıları..... 227
Şekil 108. En Az ve En Fazla Yolcuya Sahip Kalkış Şehirleri 228
Şekil 109. En Çok Yolcuya Sahip İlk 50 Rota 228
Şekil 110. En Çok Uçuşa Sahip İlk 50 Rota 229
Şekil 111. New York-Chicago Arası Aylık Yolcu Trendi 229
Şekil 112. Toplam Hava Trafiği ve Yolcu Sayısının Yıllık Bileşik Büyüme Oranı (YBBO) 230
Şekil 113. Her 100.000 Kişiye Düşen Yıllık Uçuş Oranı 231
Şekil 114. Basit Doğrusal Regresyon Grafiği 1 232
Şekil 115. Basit Doğrusal Regresyon Grafiği 2 234
Şekil 116. Basit Doğrusal Regresyon Grafiği 3 235
Şekil 117. Çoklu Doğrusal Regresyon Grafiği 1..... 236
Şekil 118. Çoklu Doğrusal Regresyon Grafiği 2..... 238
Şekil 119. Karar Ağaçları Dağılım ve Artık Değer Grafiği 240
Şekil 120. k-En Yakın Komşu Dağılım ve Artık Değer Grafiği 240
Şekil 121. Rastgele Orman Dağılım ve Artık Değer Grafiği 241
Şekil 122. Yapay Sinir Ağı Dağılım ve Artık Değer Grafiği 241
Şekil 123. Karar Ağaçları Öğrenme Eğrisi 242
Şekil 124. k-En Yakın Komşu Öğrenme Eğrisi 243
Şekil 125. Rastgele Orman Öğrenme Eğrisi 243
Şekil 126. Yapay Sinir Ağı Öğrenme Eğrisi 244
Şekil 127. Lojistik Regresyon Sınıflandırma Karmaşıklık Matrisi 245
Şekil 128. Lojistik Regresyon Öğrenme Eğrisi..... 246
Şekil 129. Lojistik Regresyon AUC-ROC Eğrisi..... 247
Şekil 130. Stratified k-Fold Lojistik Regresyon Öğrenme Eğrisi 248
Şekil 131. Sınıflandırma Modelleri Karmaşıklık Matrisi..... 250
Şekil 132. Sınıflandırma Modelleri AUC-ROC Eğrisi 251
Şekil 133. Topluluk Modelleri ve Temel Modellerin Öğrenme Eğrileri..... 252
Şekil 134. PCA Destekli 5 Sınıflı Rastgele Orman Modeli Karmaşıklık Matrisi 254
Şekil 135. PCA Destekli 5 Sınıflı Rastgele Orman Modeli AUC-ROC Eğrisi..... 255
Şekil 136. PCA Destekli 5 Sınıflı Karar Ağaçları Modeli Karmaşıklık Matrisi 257
Şekil 137. PCA Destekli 5 Sınıflı Karar Ağaçları Modeli AUC-ROC Eğrisi..... 257
Şekil 138. PCA Destekli 5 Sınıflı KNN Modeli Karmaşıklık Matrisi 259
Şekil 139. PCA Destekli 5 Sınıflı KNN Modeli AUC-ROC Eğrisi..... 259
Şekil 140. PCA Destekli 5 Sınıflı Naive Bayes Modeli Karmaşıklık Matrisi 261
Şekil 141. PCA Destekli 5 Sınıflı Naive Bayes Modeli AUC-ROC Eğrisi 261
Şekil 142. PCA Destekli 5 Sınıflı Yapay Sinir Ağı Modeli Karmaşıklık Matrisi..... 262
Şekil 143. PCA Destekli 5 Sınıflı Yapay Sinir Ağı Modeli AUC-ROC Eğrisi 263
Şekil 144. PCA Destekli 5 Sınıflı Yapay Sinir Ağı Modeli Öğrenme Eğrisi 264
Şekil 145. Veri Seti 2 Görüntüleri ve Sınıfları..... 267
Şekil 146. Temel CNN Modeli Kayıp ve Doğruluk Grafiği 268
Şekil 147. Temel CNN Modeli Karmaşıklık Matrisi 269
Şekil 148. Veri Artırımı Yapılmış CNN Modeli Kayıp ve Doğruluk Grafiği 270
Şekil 149. Veri Artırımı Yapılmış CNN Modeli Karmaşıklık Matrisi..... 271
Şekil 150. Kayıp ve Doğruluk Grafiği 1 (250x250)..... 272
Şekil 151. Kayıp ve Doğruluk Grafiği 2 (250x250)..... 272
Şekil 152. %90-%5-%5 Bölümlendirmesi Kayıp ve Doğruluk Grafiği 273
Şekil 153. %90-%5-%5 Bölümlendirmesi Karmaşıklık Matrisi 276
Şekil 154. %90-%5-%5 Bölümlendirmesi AUC-ROC Eğrisi..... 277

Şekil 155. %90-%5-%5 Veri Artırımı Yapılmış Model Kayıp ve Doğruluk Grafiği 278
Şekil 156. %90-%5-%5 Veri Artırımı Yapılmış Model Karmaşıklık Matrisi 279
Şekil 157. %90-%5-%5 Veri Artırımı Yapılmış Model AUC-ROC Eğrisi 279
Şekil 158. %80-%10-%10 Bölümlendirmesi Kayıp ve Doğruluk Grafiği 280
Şekil 159. %80-%10-%10 Bölümlendirmesi Karmaşıklık Matrisi 281
Şekil 160. %80-%10-%10 Bölümlendirmesi AUC-ROC Eğrisi 281
Şekil 161. %80-%10-%10 Veri Artırımı Yapılmış Model Kayıp ve Doğruluk Grafiği 282
Şekil 162. %80-%10-%10 Veri Artırımı Yapılmış Model Karmaşıklık Matrisi 283
Şekil 163. %80-%10-%10 Veri Artırımı Yapılmış Model AUC-ROC Eğrisi 283
Şekil 164. %90-%5-%5 Bölümlendirmesi Kayıp ve Doğruluk Grafiği 284
Şekil 165. %90-%5-%5 Bölümlendirmesi Karmaşıklık Matrisi 285
Şekil 166. %90-%5-%5 Bölümlendirmesi AUC-ROC Eğrisi 285
Şekil 167. %90-%5-%5 Veri Artırımı Yapılmış Model Kayıp ve Doğruluk Grafiği 286
Şekil 168. %90-%5-%5 Veri Artırımı Yapılmış Model Karmaşıklık Matrisi 287
Şekil 169. %90-%5-%5 Veri Artırımı Yapılmış Model AUC-ROC Eğrisi 287
Şekil 170. %80-%10-%10 Bölümlendirmesi Kayıp ve Doğruluk Grafiği 288
Şekil 171. %80-%10-%10 Bölümlendirmesi Karmaşıklık Matrisi 289
Şekil 172. %80-%10-%10 Bölümlendirmesi AUC-ROC Eğrisi 289
Şekil 173. %80-%10-%10 Veri Artırımı Yapılmış Model Kayıp ve Doğruluk Grafiği 290
Şekil 174. %80-%10-%10 Veri Artırımı Yapılmış Model Karmaşıklık Matrisi 291
Şekil 175. %80-%10-%10 Veri Artırımı Yapılmış Model AUC-ROC Eğrisi 291
Şekil 176. %80-%10-%10 Bölümlendirmesi Kayıp ve Doğruluk Grafiği 292
Şekil 177. %80-%10-%10 Bölümlendirmesi Karmaşıklık Matrisi 293
Şekil 178. %80-%10-%10 Bölümlendirmesi AUC-ROC Eğrisi 293
Şekil 179. %80-%10-%10 Veri Artırımı Yapılmış Model Kayıp ve Doğruluk Grafiği 294
Şekil 180. %80-%10-%10 Veri Artırımı Yapılmış Model Karmaşıklık Matrisi 295
Şekil 181. %80-%10-%10 Veri Artırımı Yapılmış Model AUC-ROC Eğrisi 295
Şekil 182. Öğrenme Oranı 0,0001 Kayıp ve Doğruluk Grafiği 296
Şekil 183. Öğrenme Oranı 0,0001 Karmaşıklık Matrisi 297
Şekil 184. Öğrenme Oranı 0,0001 AUC-ROC Eğrisi 297
Şekil 185. Veri Artırımı Yapılmış Öğrenme Oranı 0,0001 Kayıp ve Doğruluk Grafiği 298
Şekil 186. Veri Artırımı Yapılmış Öğrenme Oranı 0,0001 Karmaşıklık Matrisi 299
Şekil 187. Veri Artırımı Yapılmış Öğrenme Oranı 0,0001 AUC-ROC Eğrisi 299
Şekil 188. Öğrenme Oranı 0,00001 Kayıp ve Doğruluk Grafiği 300
Şekil 189. Öğrenme Oranı 0,00001 Karmaşıklık Matrisi 301
Şekil 190. Öğrenme Oranı 0,00001 AUC-ROC Eğrisi 301
Şekil 191. Veri Artırımı Yapılmış Öğrenme Oranı 0,00001 Kayıp ve Doğruluk Grafiği 302
Şekil 192. Veri Artırımı Yapılmış Öğrenme Oranı 0,00001 Karmaşıklık Matrisi 303
Şekil 193. Veri Artırımı Yapılmış Öğrenme Oranı 0,00001 AUC-ROC Eğrisi 303
Şekil 194. Temel EfficientNetB0 Modeli Kayıp ve Doğruluk Grafiği 305
Şekil 195. Temel EfficientNetB0 Modeli Karmaşıklık Matrisi 305
Şekil 196. Temel EfficientNetB1 Modeli Kayıp ve Doğruluk Grafiği 306
Şekil 197. Temel EfficientNetB1 Modeli Karmaşıklık Matrisi 307
Şekil 198. Temel EfficientNetB2 Modeli Kayıp ve Doğruluk Grafiği 308
Şekil 199. Temel EfficientNetB2 Modeli Karmaşıklık Matrisi 308
Şekil 200. Temel EfficientNetB3 Modeli Kayıp ve Doğruluk Grafiği 309
Şekil 201. Temel EfficientNetB3 Modeli Karmaşıklık Matrisi 310
Şekil 202. Temel EfficientNetV2B0 Modeli Kayıp ve Doğruluk Grafiği 310
Şekil 203. Temel EfficientNetV2B0 Modeli Karmaşıklık Matrisi 311
Şekil 204. Temel EfficientNetV2B1 Modeli Kayıp ve Doğruluk Grafiği 312
Şekil 205. Temel EfficientNetV2B1 Modeli Karmaşıklık Matrisi 312
Şekil 206. Temel EfficientNetV2B2 Modeli Kayıp ve Doğruluk Grafiği 313

Şekil 207. Temel EfficientNetV2B2 Modeli Karmaşıklık Matrisi 314
Şekil 208. Temel EfficientNetV2B3 Modeli Kayıp ve Doğruluk Grafiği 314
Şekil 209. Temel EfficientNetV2B3 Modeli Karmaşıklık Matrisi 315
Şekil 210. Optimize Edilmiş EfficientNetB0 Modeli Kayıp ve Doğruluk Grafiği 319
Şekil 211. Optimize Edilmiş EfficientNetB0 Modeli Karmaşıklık Matrisi 320
Şekil 212. Optimize Edilmiş EfficientNetB1 Modeli Kayıp ve Doğruluk Grafiği 321
Şekil 213. Optimize Edilmiş EfficientNetB1 Modeli Karmaşıklık Matrisi 322
Şekil 214. Optimize Edilmiş EfficientNetB2 Modeli Kayıp ve Doğruluk Grafiği 323
Şekil 215. Optimize Edilmiş EfficientNetB2 Modeli Karmaşıklık Matrisi 323
Şekil 216. Optimize Edilmiş EfficientNetB3 Modeli Kayıp ve Doğruluk Grafiği 324
Şekil 217. Optimize Edilmiş EfficientNetB3 Modeli Karmaşıklık Matrisi 325
Şekil 218. Optimize Edilmiş EfficientNetV2B0 Modeli Kayıp ve Doğruluk Grafiği 326
Şekil 219. Optimize Edilmiş EfficientNetV2B0 Modeli Karmaşıklık Matrisi 327
Şekil 220. Optimize Edilmiş EfficientNetV2B1 Modeli Kayıp ve Doğruluk Grafiği 328
Şekil 221. Optimize Edilmiş EfficientNetV2B1 Modeli Karmaşıklık Matrisi 329
Şekil 222. Optimize Edilmiş EfficientNetV2B2 Modeli Kayıp ve Doğruluk Grafiği 330
Şekil 223. Optimize Edilmiş EfficientNetV2B2 Modeli Karmaşıklık Matrisi 331
Şekil 224. Optimize Edilmiş EfficientNetV2B3 Modeli Kayıp ve Doğruluk Grafiği 332
Şekil 225. Optimize Edilmiş EfficientNetV2B3 Modeli Karmaşıklık Matrisi 333
Şekil 226. Örnek Makine Öğrenmesi Süreci Provenans Veri Hattı Gösterimleri 336
Şekil 227. Örnek Transfer Öğrenmesi Süreci Provenans Veri Hattı Gösterimleri 337
Şekil 228. Her İki Örnek Sürece Ait BagIt Yapısı ve Metaverileri 338
Şekil 229. Archivematica Transfer Süreci Gösterimleri 1 341
Şekil 230. Archivematica Transfer Süreci Gösterimleri 2 342
Şekil 231. Archivematica Alım (Ingest) Aşaması 343
Şekil 232. İçeriklerin Normalizasyonu 343
Şekil 233. İçeriklere Metaveri Eklenmesi 344
Şekil 234. SIP Kayıtları 344
Şekil 235. DIP Yükleme Adımı 344
Şekil 236. AIP METS Dosyası 345
Şekil 237. Tamamlanmış AIP Süreci 345
Şekil 238. Hesaplamalı Arşiv Süreçleri Genel Provenans Şeması 346

TABLOLAR LİSTESİ

Tablo 1. Arşivlerdeki Paradigma Değişiminin Yönleri 28
Tablo 2. Arşiv Bilimi ve Bilgisayar Biliminde Provenans 41
Tablo 3. Provenans ve Fonlara Saygı İlkeleri Arasındaki Temel Farklılıklar 46
Tablo 4. Orijinal Düzen ve Provenans İlkeleri Arasındaki Temel Farklılıklar 48
Tablo 5. Orijinal Düzen ve Fonlara Saygı İlkeleri Arasındaki Temel Farklılıklar 49
Tablo 6. Fiziksel ve Dijital Ortamdaki Veri Kavramının Farkları 60
Tablo 7. Arşiv Uzmanları Tarafından Önerilen Araştırma Alanları 67
Tablo 8. Çok Disiplinli Perspektif Bağlamında Disiplinlerin Odakları 68
Tablo 9. Bazı Hesaplamalı Arşiv Bilimi Projeleri 71
Tablo 10. Yapay Zekâ Türleri ve Karşılaştırmaları 83
Tablo 11. ResNET Varyantları 189
Tablo 12. EfficientNET Varyantları 190
Tablo 13. Veri Seti 1 Hakkında Bilgiler 195
Tablo 14. Veri Seti 2 Hakkında Bilgiler 197
Tablo 15. Tezde Kullanılan Bazı Örnek Kütüphaneler 199
Tablo 16. Tensorflow Alt Kütüphane Örnekleri 199
Tablo 17. Karmaşıklık Matrisi 202
Tablo 18. Veri Seti 1 Kolmogorov-Smirnov Test Sonuçları 213
Tablo 19. Öznitelik Mühendisliği ile Türetilen Yeni Öznitelikler 214
Tablo 20. Basit Doğrusal Regresyon 1 Ölçüm Değerleri 232
Tablo 21. Basit Doğrusal Regresyon 2 Ölçüm Değerleri 233
Tablo 22. Basit Doğrusal Regresyon 3 Ölçüm Değerleri 234
Tablo 23. Çoklu Doğrusal Regresyon 1 Ölçüm Değerleri 236
Tablo 24. Çoklu Doğrusal Regresyon 2 Ölçüm Değerleri 237
Tablo 25. Regresyon Modelleri Değerlendirme Sonuçları 239
Tablo 26. Lojistik Regresyon Modeli Sınıflandırma Raporu 246
Tablo 27. Sınıflandırma Modellerinin Karşılaştırılması 249
Tablo 28. Uçuş Başına Yolcu Sayısı Sınıf Dağılımları 253
Tablo 29. PCA Destekli 5 Sınıflı Rastgele Orman Modeli Sınıflandırma Raporu 254
Tablo 30. PCA Destekli 5 Sınıflı Karar Ağaçları Modeli Sınıflandırma Raporu 256
Tablo 31. PCA Destekli 5 Sınıflı KNN Modeli Sınıflandırma Raporu 258
Tablo 32. PCA Destekli 5 Sınıflı Naive Bayes Modeli Sınıflandırma Raporu 260
Tablo 33. PCA Destekli 5 Sınıflı Yapay Sinir Ağı Modeli Sınıflandırma Raporu 262
Tablo 34. Tüm Sınıflandırma Modellerinin Performans Karşılaştırması 264
Tablo 35. Veri Seti 2 Temel CNN Mimarisi 265
Tablo 36. Görüntü Boyutu Bazlı Karşılaştırmalar 1 272
Tablo 37. Görüntü Boyutu Bazlı Karşılaştırmalar 2 273
Tablo 38. Adam Optimizasyon Algoritması Bazlı Karşılaştırmalar 274
Tablo 39. Adamax Optimizasyon Algoritması Bazlı Karşılaştırmalar 274
Tablo 40. Adagrad Optimizasyon Algoritması Bazlı Karşılaştırmalar 274
Tablo 41. RMSProp Optimizasyon algoritması Bazlı Karşılaştırmalar 275
Tablo 42. SGD Optimizasyon algoritması Bazlı Karşılaştırmalar 275
Tablo 43. Bölümlendirme Bazlı Karşılaştırmalar 1 275
Tablo 44. %90-%5-%5 Bölümlendirmesi Değerlendirme Sonuçları 278
Tablo 45. Bölümlendirme Bazlı Karşılaştırmalar 2 280
Tablo 46. %80-%10-%10 Bölümlendirmesi Değerlendirme Sonuçları 282
Tablo 47. Bölümlendirme Bazlı Karşılaştırmalar 3 284
Tablo 48. %90-%5-%5 Bölümlendirmesi Değerlendirme Sonuçları 286
Tablo 49. Bölümlendirme Bazlı Karşılaştırmalar 4 288
Tablo 50. %80-%10-%10 Bölümlendirmesi Değerlendirme Sonuçları 290

Tablo 51. Bölümlendirme Bazlı Karşılaştırmalar 5 292
Tablo 52. %80-%10-%10 Bölümlendirmesi Değerlendirme Sonuçları 294
Tablo 53. Öğrenme Oranı Bazlı Karşılaştırmalar 1 296
Tablo 54. Veri Artırımı Yapılmış Öğrenme Oranı 0,0001 Değerlendirme Sonuçları 298
Tablo 55. Öğrenme Oranı Bazlı Karşılaştırmalar 2 300
Tablo 56. Veri Artırımı Yapılmış Öğrenme Oranı 0,00001 Değerlendirme Sonuçları 302
Tablo 57. EfficientNetB0 Eğitim Kümesi Bazlı Model Analiz Sonuçları 304
Tablo 58. EfficientNetB1 Eğitim Kümesi Bazlı Model Analiz Sonuçları 306
Tablo 59. EfficientNetB2 Eğitim Kümesi Bazlı Model Analiz Sonuçları 307
Tablo 60. EfficientNetB3 Eğitim Kümesi Bazlı Model Analiz Sonuçları 309
Tablo 61. EfficientNetV2B0 Eğitim Kümesi Bazlı Model Analiz Sonuçları 310
Tablo 62. EfficientNetV2B1 Eğitim Kümesi Bazlı Model Analiz Sonuçları 311
Tablo 63. EfficientNetV2B2 Eğitim Kümesi Bazlı Model Analiz Sonuçları 313
Tablo 64. EfficientNetV2B3 Eğitim Kümesi Bazlı Model Analiz Sonuçları 314
Tablo 65. Optimize Edilmemiş EfficientNet Modelleri Sonuçlar 315
Tablo 66. Modellere Uygulanan Düzenlileştirme Tekniği ve Değerleri 318
Tablo 67. EfficientNetB0 Modeline Ait En İyi Hiperparametreler ve Değerleri 319
Tablo 68. Optimize Edilmiş EfficientNetB0 Modeline Ait Analiz Sonuçları 319
Tablo 69. EfficientNetB1 Modeline Ait En İyi Hiperparametreler ve Değerleri 320
Tablo 70. Optimize Edilmiş EfficientNetB1 Modeline Ait Analiz Sonuçları 321
Tablo 71. EfficientNetB2 Modeline Ait En İyi Hiperparametreler ve Değerleri 322
Tablo 72. Optimize Edilmiş EfficientNetB2 Modeline Ait Analiz Sonuçları 323
Tablo 73. EfficientNetB3 Modeline Ait En İyi Hiperparametreler ve Değerleri 324
Tablo 74. Optimize Edilmiş EfficientNetB3 Modeline Ait Analiz Sonuçları 324
Tablo 75. EfficientNetV2B0 Modeline Ait En İyi Hiperparametreler ve Değerleri 326
Tablo 76. Optimize Edilmiş EfficientNetV2B0 Modeline Ait Analiz Sonuçları 326
Tablo 77. EfficientNetV2B1 Modeline Ait En İyi Hiperparametreler ve Değerleri 327
Tablo 78. Optimize Edilmiş EfficientNetV2B1 Modeline Ait Analiz Sonuçları 328
Tablo 79. EfficientNetV2B2 Modeline Ait En İyi Hiperparametreler ve Değerleri 329
Tablo 80. Optimize Edilmiş EfficientNetV2B2 Modeline Ait Analiz Sonuçları 330
Tablo 81. EfficientNetV2B3 Modeline Ait En İyi Hiperparametreler ve Değerleri 331
Tablo 82. Optimize Edilmiş EfficientNetV2B3 Modeline Ait Analiz Sonuçları 332
Tablo 83. Transfer Öğrenmesi Genel Değerlendirme Tablosu 333
Tablo 84. Transfer Öğrenmesi Literatür Çalışmaları ve Karşılaştırmalar 354

SEMBOLLER LİSTESİ

e	:	Euler sayısı
ε	:	İstatistiksel hata terimi
$\sum$	:	Toplam
$\forall$	:	Bütün
ξ	:	Esnek değişken
ϕ	:	Özellik dönüşüm fonksiyonu
exp	:	Üstel fonksiyon
γ	:	Yayılım katsayısı
σ	:	Bant genişliği (düzeltme katsayısı)
$\prod$	:	Çarpma
$\propto$	:	Orantı (sabit oran)
μ	:	Ortalama
$\mid$	:	Koşullu olasılık
a	:	Aktivasyon değeri
θ	:	Modelin öğrenilen parametreleri
$J(\theta)$	:	Kayıp fonksiyonu
δ	:	Geri yayılım hata değeri
∇	:	Gradyan
$argmin$	:	Minimum değer indisi
cos	:	Kosinüs benzerlik fonksiyonu
ε	:	Komşuluk yarıçapı (DBSCAN)
$\exists$	:	Varlık niceleyicisi
$\cup$	:	Birleşim
$\notin$	:	Elemanı değil
λ	:	Özdeğer (eigenvalue)
v	:	Özvektör (eigenvector)
V	:	Özvektör matrisi
$\Phi(x)$	:	Standart Gauss kümülatif dağılım fonksiyonu
max	:	En büyük değer
erf	:	Hata fonksiyonu
π	:	Pi sayısı
λ	:	Düzenleme (ceza) parametresi (L1 ve L2)
F	:	Evrişim filtresi
$*$	:	Evrişim işlemi
θ	:	Gradyan (kenar) yönü (Prewitt)
$atan$	:	Arktanjant fonksiyonu
∇^2	:	Laplasyen

EKLER LİSTESİ

Ek 1. Temel CNN Modeli Yapısı Python Gösterimi 423
Ek 2. Temel CNN Modeli Katmanları 424
Ek 3. Veri Seti 2 Normalizasyon 425
Ek 4. Temel CNN Modeli Sınıflandırma Raporu 425
Ek 5. Veri Artırımı Yapılmış CNN Modeli Sınıflandırma Raporu 426
Ek 6. %90-%5-%5 Bölümlendirmesi Adam Opt. 1. Model Sınıflandırma Raporu 427
Ek 7. %90-%5-%5 Bölümlendirmesi Adam Opt. 2. Model Sınıflandırma Raporu 428
Ek 8. %80-%10-%10 Bölümlendirmesi Adamax Opt. 1. Model Sınıflandırma Raporu 429
Ek 9. %80-%10-%10 Bölümlendirmesi Adamax Opt. 2. Model Sınıflandırma Raporu 430
Ek 10. %90-%5-%5 Bölümlendirmesi Adagrad Opt. 1. Model Sınıflandırma Raporu 431
Ek 11. %90-%5-%5 Bölümlendirmesi Adagrad Opt. 2. Model Sınıflandırma Raporu 432
Ek 12. %80-%10-%10 Bölümlendirmesi RMSProp Opt. 1. Model Sınıflandırma Raporu 433
Ek 13. %80-%10-%10 Bölümlendirmesi RMSProp Opt. 2. Model Sınıflandırma Raporu 434
Ek 14. %80-%10-%10 Bölümlendirmesi SGD Opt. 1. Model Sınıflandırma Raporu 435
Ek 15. %80-%10-%10 Bölümlendirmesi SGD Opt. 2. Model Sınıflandırma Raporu 436
Ek 16. Öğrenme Oranı 0,0001 1. Model Sınıflandırma Raporu 437
Ek 17. Öğrenme Oranı 0,0001 2. Model Sınıflandırma Raporu 438
Ek 18. Öğrenme Oranı 0,00001 1. Model Sınıflandırma Raporu 439
Ek 19. Öğrenme Oranı 0,00001 2. Model Sınıflandırma Raporu 440
Ek 20. Temel EfficientNetB0 Modeli Sınıflandırma Raporu 441
Ek 21. Temel EfficientNetB1 Modeli Sınıflandırma Raporu 442
Ek 22. Temel EfficientNetB2 Modeli Sınıflandırma Raporu 443
Ek 23. Temel EfficientNetB3 Modeli Sınıflandırma Raporu 444
Ek 24. Temel EfficientNetV2B0 Modeli Sınıflandırma Raporu 445
Ek 25. Temel EfficientNetV2B1 Modeli Sınıflandırma Raporu 446
Ek 26. Temel EfficientNetV2B2 Modeli Sınıflandırma Raporu 447
Ek 27. Temel EfficientNetV2B3 Modeli Sınıflandırma Raporu 448
Ek 28. Hiperparametreler İçin Belirlenen Aralıklar 449
Ek 29. Optimize Edilmiş EfficientNetB0 Sınıflandırma Raporu 450
Ek 30. Optimize Edilmiş EfficientNetB1 Sınıflandırma Raporu 451
Ek 31. Optimize Edilmiş EfficientNetB2 Sınıflandırma Raporu 452
Ek 32. Optimize Edilmiş EfficientNetB3 Sınıflandırma Raporu 453
Ek 33. Optimize Edilmiş EfficientNetV2B0 Sınıflandırma Raporu 454
Ek 34. Optimize Edilmiş EfficientNetV2B1 Sınıflandırma Raporu 455
Ek 35. Optimize Edilmiş EfficientNetV2B2 Sınıflandırma Raporu 456
Ek 36. Optimize Edilmiş EfficientNetV2B3 Sınıflandırma Raporu 457

KISALTMALAR LİSTESİ

ABD	**:**	Amerika Birleşik Devletleri
API	**:**	Application Programming Interface
AUC-ROC	**:**	Area under the Curve - Receiver Operating Characteristic Curve
BIOS	**:**	Basic Input/Output System
BİT	**:**	Bilgi ve İletişim Teknolojileri
CAS	**:**	Computational Archival Science
CASES	**:**	Computational Archival Science Educational System
CD-ROM	**:**	Compact Disk-Read Only Memory
DOAJ	**:**	Directory of Open Access Journals
DVD-ROM	**:**	Digital Versatile Disk-Read Only Memory
EEPROM	**:**	Electrically Erasable Programmable Read-Only Memory
GPU	**:**	Graphics Processing Unit
HACA	**:**	The Housing Authority of the City of Asheville
IBM	**:**	International Business Machines Corporation
IEEE	**:**	Institute of Electrical and Electronics Engineers
INHS	**:**	Illinois Natural History Survey
JPEG	**:**	Joint Photographic Experts Group
JSTOR	**:**	Journal Storage
LSTM	**:**	Long Short-Term Memory
MASK R-CNN	**:**	Mask Region-based Convolutional Neural Network
MHCP	**:**	Morgenthau Holocaust Collections Project
NARA	**:**	National Archives and Records Administration
NER	**:**	Named Entity Recognition
NLP	**:**	Natural Language Processing
OCR	**:**	Optical Character Recognition
PDF	**:**	Portable Document Format
PDF/A	**:**	Portable Document Format Archival
PREMIS	**:**	Preservation Metadata Implementation Strategies
PROV	**:**	Provenance
PROV-ML	**:**	Provenance - Machine Learning
RAR	**:**	Roshal Archive
RDF	**:**	Resource Description Framework
RI	**:**	The Roosevelt Institute
RiC-CM	**:**	Records in Contexts-Conceptual Model
RNN	**:**	Recurrent Neural Network
SGD	**:**	Stochastic Gradient Descent
SVM	**:**	Support Vector Machine
SVR	**:**	Support Vector Regressor
TIFF	**:**	Tag Image File Format
TSV	**:**	Tab Separated Values
TXT	**:**	Text
ULAKBİM	**:**	Ulusal Akademik Ağ ve Bilgi Merkezi
USB	**:**	Universal Serial Bus
VGG	**:**	Visual Geometry Group
WAV	**:**	Waveform Audio File Format
YÖK	**:**	Yükseköğretim Kurulu

GİRİŞ

İnsanoğlunun yaşam serüveni; kısa ve uzun vadeli ihtiyaçlarını karşılama süreçlerinde yüz yüze kaldığı problemler ile deneme-yanılma yoluyla geliştirdiği çözümleri kayıt altına alarak günümüze taşınmıştır. Bu kayıtlar, tarihsel süreçte mağara duvarlarından kil tabletlere, papirüs ve kâğıttan EEPROM[1], CD-ROM[2] ve USB[3] gibi dijital ortamlara kadar farklı fiziksel ve dijital biçimlerde varlık göstermiştir. Günümüzde bu süreç, Bilgi Çağı olarak adlandırılan bir dönemde devam etmektedir. Kayıtlı bilginin taşıdığı değer, insanlık tarihi boyunca arşivlerin önemini artırmıştır. Özellikle resmi belgelerin ispat ve kanıt niteliği taşıması, arşivlerin kurumsal güvenceye alınmasını gerekli kılmıştır. Bu nedenle arşivler, sadece belgelerin saklandığı alanlar değil; toplumun yönetimsel, hukuksal ve kültürel sürekliliğini sağlayan yapılar olarak değerlendirilmelidir (Yeo, 2018).

Yeo (2021) ilk çağlarda Mezopotamya, Mısır, Yunanistan ve Roma İmparatorluğu gibi birçok bölgede büyük miktarda kayıt ve belge üretilmiş olmasına rağmen; savaşlar ve ihmaller nedeniyle bu kayıt ve belgelerin sadece küçük bir kısmının günümüze ulaştığını vurgulamıştır. Orta Çağ'dan itibaren ise matbaanın icadı, coğrafi keşifler, Rönesans, Aydınlanma ve Sanayi Devrimi gibi gelişmeler belge üretimini önemli ölçüde artırmış; bu durum arşivcilik uygulamalarının giderek kurumsallaşmasına zemin hazırlamıştır. Özellikle modern devletlerin ortaya çıkışıyla birlikte arşivler, yönetimsel işleyişin vazgeçilmez bir parçası hâline gelmiş ve bu süreçte fonlara saygı, provenans ve orijinal düzen gibi temel arşivcilik ilkeleri şekillenmiştir. 20. yüzyıla gelindiğinde ise özellikle kamu ve araştırma hizmetleri için belge yönetimi sistemleri geliştirilmiş; arşivler yalnızca belge saklama işleviyle sınırlı kalmayıp, erişim, sınıflandırma ve uzun süreli koruma gibi çok boyutlu işlevler üstlenen bilgi merkezlerine dönüşmüştür.

İnternetin özellikle 1990'lı yılların ortalarından itibaren yaygınlaşmaya başlaması ile dijital ortamda oluşan veri ve bilginin miktarı da artmaya başlamıştır. Web 1.0'dan Web 4.0'a kadar gelişen internet teknolojileriyle içerik oluşturma becerileri de artmış ve böylece veri üretimi üstel (exponential) büyüme göstermeye başlamıştır. Bu gelişmeler ise büyük veri (big data) kavramını ortaya çıkarmıştır. Her ne kadar kavramın kökleri, ilişkisel veritabanlarının ve ilk nesil veri merkezlerinin geliştirilmeye başlandığı 1960'lı yıllara dayansa da (Alvarado, 2023,

[1] Electrically Erasable Programmable Read-Only Memory
[2] Compact Disk-Read Only Memory
[3] Universal Serial Bus

s. 26) günümüzde dijital anlamda yönetilemez ölçekteki veri yığınlarını tanımlamak için kullanılmaktadır.

Arşivsel süreçler, tarih boyunca fiziksel belgeler üzerine kuruluyken, dijital verilerin ve büyük veri kavramının yükselişi, arşiv bilimi için önemli bir dönüşümü zorunlu kılmıştır. Artan veri hacmi ve çeşitliliği, teorisyen ve uygulamacıları yeni yöntemler geliştirmeye zorlamış; geleneksel ilkelerle bu ölçekteki dijital gerçekliğin yönetilemeyeceği açıkça görülmüştür. Bu bağlamda büyük veri, arşiv bilimi açısından bir kaos değil, veriyle etkileşimi yeniden tanımlayabilecek bir potansiyel alan olarak değerlendirilmelidir.

Arşiv biliminin odağı, çok kısa bir zaman diliminde fiziksel belgelerden dijital bilgi ve belge sistemlerine geçiş yapmıştır. Bu geçiş, hem teknolojik bir dönüşüm hem de metodolojik ve kavramsal bir yapılanma olarak görülmelidir. Verinin "21. yüzyılın petrolü" olarak tanımlandığı yayınlar (Possler, Bruns ve Niemann-Lenz, 2019; Quigley vd., 2021; Stach, 2023; Gbeassor, 2024) dijital verinin stratejik değerini açıkça ortaya koymaktadır. Artık, arşiv biliminin odağı sadece belgelerin korunmasına değil; dijital verinin işlenmesi, yorumlanması, anlamlandırılması ve uzun süreli saklanmasına doğru evrilmiştir.

Bu bağlamda; geleneksel arşiv ilkelerinin dijital ortama uyarlanabilirliği (Conway, 2010, s. 65), sanal depolama mantığına arşivsel ilke ve yöntemlerin dahil edilmesi (Yakel vd., 2011, s. 25), ölçeklenebilirlik sorunsalının mantıksal modellenmesi (Hedstrom, 1997) ve zaman ve mekândan bağımsız erişimin tam anlamıyla sağlanması (Jablonski, 2016) gibi uyumlaştırma çalışmaları gerçekleştirilmiştir.

Dijital arşivlerde, klasik arşiv anlayışındaki fiziksel orijinallik kavramı geçerliliğini yitirmiştir. Bunun yerine, bir materyalin "stored record[4]" ile "manifested record[5]" arasındaki ayrımı önem kazanmıştır[6] (Trace, 2020; Sağlık, 2021). Bu bağlamda, örüntü analizleri gibi işlemler, genellikle belgenin "manifested" kopyaları ya da türevleri üzerinden yürütülmektedir. Bu sayede veriler arasındaki ilişkisellik açığa çıkarılarak, "nitelikli veri havuzları" oluşturulabilir. Özellikle yapılandırılmış veya yapılandırılmamış dijital arşiv verilerinin içinde barındırdığı örüntülerin keşfi, erişilebilirliğinin artırılması ve dijital provenansının oluşturulması ile korunması gibi konular, geleneksel arşivcilik yaklaşımlarından farklı olarak yeni teknolojik yöntemleri zorunlu kılmaktadır.

[4] Koruma amacıyla saklanan asıl kopya
[5] Erişime sunulan ve işlem gören türev
[6] Bu ayrım, dijital ortamlarda belgenin orijinalliği yerine özgünlüğü ve bütünlüğünün korunması anlayışına dayanmaktadır (Duranti ve Thibodeau, 2006).

Günümüzde yapay zekâ teknolojisi, büyük veri niteliğindeki arşivlerde örüntüleri tespit etmek ve erişilebilirliği artırmak için başvurulan temel araçlar arasında yer almaktadır. Bu gelişmelerle birlikte arşiv bilimi, diğer disiplinlerle (özellikle bilgisayar bilimi) kurduğu ilişkiler aracılığıyla disiplinlerarası[7] ve hatta disiplinlerüstü[8] bir niteliğe kavuşmuştur. Gerek dijital doğan gerekse dijitalleştirilen verilerde, bu tür analizler arşivsel anlamı bozmadan yeni yapılar keşfetmeye olanak sağlamaktadır. Bu, alanın henüz tam anlamıyla içselleştiremediği[9] "veri manipülasyonu[10]" kavramı çerçevesinde, bilgiye daha derinlemesine erişim ve analiz olanakları sunmaktadır.

Klasik arşiv biliminde bilgi ve belgelere (içerikleri dahil) orijinallik ve bütünlük ilkeleri gereği doğrudan müdahale edilmemektedir. Ancak dijital veri arşivlerinde, belgenin orijinal formatından türetilen reprodüksiyonlar (örneğin, biraz önce bahsedilen "manifested records" kavramı) üzerinden analizler gerçekleştirilebilmektedir. Bu durum teknik bir "kopyalama" sürecinden ziyade belgenin bağlamsal ilişkilerini ortaya koymaya yönelik hesaplamalı analiz olanaklarını da beraberinde getirmektedir. Bu tür uygulamalar; provenans gibi temel ilkelere zarar verilmeden arşivsel bağlam korunarak gerçekleştirilebilmektedir. Arşiv bilimi özelinde düşünüldüğünde bu yöntemler; geleneksel arşivcilik ilkelerini tamamlayan, alanın disiplinlerarası gelişimine katkı sunan ve disiplinlerüstü bir konuma çıkaran perspektif sağlamaktadır. Böylece dijital ortamdaki arşivsel problemlerin çözümünde, hesaplamalı yöntemler arşiv biliminin mantıksal ortakları hâline gelmektedir. Bu bağlamda ise şu sorular ortaya çıkmaktadır: Hesaplamalı yöntemler kullanılarak dijital arşiv verilerinde nasıl örüntü keşfi yapılabilir? Keşfedilen bu örüntüler, veri bütünlüğü korunarak uzun süreli arşivlemeye nasıl entegre edilebilir?

Bu tez, hesaplamalı yöntemlerin (makine öğrenmesi ve derin öğrenme) arşiv bilimi bağlamında nasıl uygulanabileceğini ortaya koymayı amaçlamaktadır. Çalışma kapsamında, dijital arşiv[11]

[7] Disiplinlerarası çalışma; iki ya da daha fazla disiplinin kavram, yöntem veya yaklaşımlarının bir araya getirilerek ortak bir sorunun çözümlenmesini ifade etmektedir. Arşiv bilimi bağlamında ise, örneğin delil değeri ya da belgenin hukuki statüsü gibi konularda hukuk, bilgi bilimi ve tarih gibi alanlarla iş birliği yapılması bu duruma örnektir.

[8] Disiplinlerüstü çalışma ise mevcut disipliner sınırların ötesine geçen, kendi teorik çerçevesini oluşturan ve yeni bilgi alanları yaratan yaklaşımları tanımlaktadır. Yapay zekâ ile arşivciliğin birleştiği alan olan hesaplamalı arşiv bilimi, otomatik dijital provenans oluşturma, algoritmik dokümantasyon ya da makine öğrenmesi temelli belgesel analiz gibi uygulamalar, arşiv bilimini disiplinlerüstü bir bilgi üretim sahasına dönüştürmektedir.

[9] Hesaplamalı Arşiv Bilimi ve Çalışmaları bölümünde detaylı açıklanmıştır.

[10] Orijinal veri setinin bütünlüğünü ve otantikliğini bozmadan, veri setinin kopyasından türetilen versiyonlarda analiz işlemlerinin yapılabilmesini ifade etmektedir.

[11] Tezde, dijital arşiv terimi; dijital ortamda üretilen ya da dijitalleştirme süreçleri sonucunda oluşturulan bilgi kaynaklarının, içerik ve bağlam bütünlüğü korunarak uzun süreli erişim ve kullanımı amacıyla oluşturulan arşivsel sistemler bütünü anlamında kullanılmıştır. Dijital arşivler, materyallere ait provenans, işlem bilgisi ve teknik metaveri öğelerini de kapsamakta ve böylece güvenilirlik, bütünlük ve sürdürülebilirliği sağlamayı hedeflemektedir. Örnek olarak; data.gov, Kaggle Datasets, NASA Earthdata, Google Earth Engine Datasets, The Digital Public Library of America (DPLA), Europeana ve Library of Congress Digital Collections verilebilir.

verilerindeki örüntülerin açığa çıkarılması ve ardından uzun süreli dijital korumaya uygun veri paketlerine dönüştürülmesi hedeflenmektedir. Tüm bu süreç, arşivsel standartlarla uyumlu biçimde bir model olarak yapılandırılmıştır.

BİRİNCİ BÖLÜM

1. TEZİN AMACI, HİPOTEZLERİ, YÖNTEMİ VE KAPSAMI

1.1. Konu ve Önem

Bu tez, dijital arşiv verilerinin büyük veri çağında nasıl daha etkin yönetilebileceğini araştırmakta ve hesaplamalı yöntemlerin arşiv bilimine entegrasyonuna odaklanmaktadır. Çalışmada, yapılandırılmış ve yapılandırılmamış dijital arşiv verilerinin makine öğrenmesi ve derin öğrenme teknikleriyle analiz edilmesi, analizler sonucunda elde edilen çıktıların provenansının yapılandırılması ve uzun süreli saklamaya uygun veri paketlerinin oluşturulması amaçlanmaktadır. Bu yönüyle tez, hesaplamalı arşiv bilimi yaklaşımının hem teorik hem de uygulamalı bir örneğini sunmaktadır.

Bilinen tarihin başlangıcından itibaren, belirli bir amaç doğrultusunda, iş ve işlemlere dair süreçlerin kanıtı için oluşturulan her türlü kayıt malzemesine, sonradan ihtiyaç duyulması halinde sistematik bir biçimde ve kolayca erişilebilmesi için oluşturulan arşivlere, oldukça uzun bir süre boyunca fiziksel taşıyıcı ortamlar (kil tablet, ipek ve bambu, papirüs, parşömen ve M.Ö. 2. yüzyıldan beri kullanımı bilinen kâğıt) hakim olmuştur. 15. yüzyılda coğrafi keşiflerin başlaması ve matbaanın icadı, 16. yüzyılda Reform hareketleri ve Rönesans, 17. yüzyılda Aydınlanma Çağı ve 18. yüzyılda Sanayi Devrimi gibi gelişmeler; bilgi üretimi, iletişim ve yönetim süreçlerinde köklü dönüşümlere yol açmış ve bu dönüşümler sonucunda 19. yüzyılda modern bürokratik yapılar bir zorunluluk olarak ortaya çıkmıştır. Zaman içerisinde giderek kalabalıklaşan dünya nüfusunun da tüm bu gelişmelere eklenmesiyle, iş ve işlemlerin kanıtı olan belgeler, çarpıcı bir biçimde çoğalmış; bu durum ise, belgelerin kolay erişim için tanımlanması ve düzenlenmesi ile değerine göre ayıklanması, değersiz görülenlerin ise imha edilmesi ihtiyacını doğurmuştur.

Her ne kadar belge tanımlama, düzenleme, ayıklama ve imha uygulamalarına 20. yüzyıldan önce -örneğin 19. yüzyıl Osmanlı arşiv uygulamalarında- çeşitli biçimlerde rastlansa da; bu işlemler, özellikle 20. yüzyıldan itibaren artan belge üretimi ve bilgi yoğunluğu nedeniyle daha sistematik, kurumsal ve normatif bir hale gelmiş, böylece belirli bir değere sahip belgelerin seçilmesi, erişimin kolaylaştırılması ve arşivsel düzenin sağlanması yönünde daha yoğun bir ihtiyaç doğmuştur. Özellikle 2. Dünya Savaşı ile birlikte "belge yönetimi" kavramı geliştirilmiş ve var olan temel paradigma olan arşiv yönetimi ile birincil düzeyde ilişkilendirilmiştir. Arşivlerin ve belgelerin etkin ve verimli bir yönetim felsefesine kavuşması için yapılan bu uğraşlar, doğal olarak başlangıçta sadece fiziksel ortamda bulunan çeşitli taşıyıcı ortamlara

(kâğıt ve türevleri, mikrofilm, mikrofiş vb.) yönelik olmuştur. 20. yüzyılın ortalarında başlayıp, son çeyreğinden itibaren ivme kazanan Bilgi ve İletişim Teknolojilerindeki (BİT) gelişmeler ile birlikte, internetin 30 Ocak 1993 tarihinde küresel erişime açılması (Berners-Lee ve Fischetti, 2000; Ring, 2023) bir kırılma noktası oluşturmuştur. Yeni bir dönemin başlangıcını temsil eden bu kırılma noktası ile arşivler artık sadece fiziksel değil, etkileşimli dijital ortamlarda da oluşmaya başlamıştır. Bu tarihten önce de dijital arşivler mevcuttu; ancak erişim çoğunlukla kurumsal yapı ile sınırlıydı. Günümüze kadar, arşiv biliminin karakteristik özellikleri dijitalleşmenin gereklilikleri karşısında değişime uğramıştır. Dijital ortamda veriler çoğunlukla bilgisayar ve ağ bağlantıları aracılığıyla üretilmekte ve iletilmektedir. Bu süreçte, verilerin bir araya gelmesiyle oluşan bilgiler, tanımlayıcı metaverileriyle birlikte, klasik arşivcilik anlayışı açısından daha önce örneği görülmemiş yeni bir dijital yapı[12] meydana getirmiştir.

İnternetin yaygınlaşmasıyla birlikte dijital veri üretimi katlanarak artmış ve yüzyıllardır belge ve arşivlerin temel taşıyıcı ortamı olarak bilinen kâğıt, dijital ortama aktarılmaya başlanmıştır. Bunun yanında arşivciler de geleneksel arşiv metodolojilerini gözden geçirerek dijital ortama uyum sağlamaya çalışmışlardır. Arşivler için geliştirilen ilk nesil veritabanları ve elektronik arşiv yönetim sistemleri gibi bilgisayar tabanlı sistemler, daha büyük hacimli verilerin işlenmesini ve belgelerin yönetilmesini kolaylaştırmıştır. Ancak bu türden ilk araçlar, tamamıyla yapılandırılmış veriler (elektronik tablo verileri) için tasarlanmış olup; yapılandırılmamış ve karmaşık (e-posta, sosyal medya iletileri, multimedya dosyaları ve veritabanları vb.) dijital materyaller için düşünülmemiştir. Çünkü ilk nesil bilgi işlem ortamları çoğunlukla yapılandırılmış veri içermekteydi. 2010'lu yıllardan itibaren mobil teknolojilerin (özellikle akıllı telefonlar ve sensörler) de oldukça fazla kullanılmasıyla üretilen, saklanan ve paylaşılan büyük miktarda dijital veriyi yönetmek, anlamlandırmak ve korumak için yenilikçi yöntemlere ihtiyaç duyulmuştur. Çünkü dijital ortam, sürdürülebilir olmayan veri miktarının[13] hem depolanması hem de analiz edilmesini zorlaştırmaktadır.

Arşivcilik alanında önde gelen ülkelerde bulunan uygulamacılar ve teorisyenler, veri patlaması karşısında hem arşivleri geliştirmek hem de daha gelişmiş hesaplama tekniklerini arşiv uygulamalarına dahil etmenin gerekliliği konusunda hemfikir olmuşlardır (Blanke, 2024). Böylece, bu temel zorluğa yanıt olarak; sanal ortamda yapılandırılan arşivlerin, barındırdığı verilerle birlikte işlenmesi, korunması ve erişime açılabilmesi konularında, halihazırda

[12] Tamamen bitlerden oluşan ve 0-1 mantığında çalışan bilgisayar ortamında.
[13] Veri patlaması

bilgisayar biliminin temeli olan ve diğer birçok bilim dalında da kullanılan hesaplamalı yöntemlerin, arşiv biliminin yöntem ve ilkelerinde de kullanılabilmesini amaçlayan hesaplamalı arşiv bilimi[14] alanı ortaya çıkmıştır. Hesaplamalı arşiv bilimi; bilgisayar bilimi, bilgi bilimi ve arşiv teorisini birleştirmeyi amaçlayan ve disiplinlerüstü yaklaşımla ele alınan yeni bir bilim dalıdır (Payne, 2018). Hesaplamalı arşiv bilimi; arşivsel problemlere çözüm olarak güncel teknolojiler olan makine öğrenmesi, derin öğrenme, doğal dil işleme, veri madenciliği vb. gibi teknikleri kullanmaktadır. Sadece yardımcı "teknikler" olarak değil, karşılıklı olarak bu tekniklerin arkasındaki teorilerden de beslenmeyi amaçlayan bir bilim dalı olarak hesaplamalı arşiv bilimi; arşivcilerin ve araştırmacıların büyük ölçekli dijital veri içeren arşivleri daha verimli bir şekilde ele almalarını, erişilebilirliği artırmalarını ve elle keşfedilmesi pek mümkün olmayan karmaşık ve yüksek hacimli arşiv materyallerinden yeni içgörüler ve dolaysıyla yorumlanabilir stratejik bilgiler ortaya çıkarmalarını sağlamaktadır. Bu noktada; klasik fiziksel ortamdan farklı olarak dijital materyallerin, hızla değişen yazılım ve donanım nedeniyle çabuk eskimeleri ve veri özgünlüğünün sağlanamama ihtimali (Wieringa, 2017; Frank, 2020) gibi dezavantajlara sahip olduğu da unutulmamalıdır.

Günümüzde bir arşivci veya alan araştırmacısının, dijital arşivi oluşturan verilerin anlamını ve oluşturulma sürecini bilmesi, Bilgi Çağı'nda ayırt edici bir yetkinlik olarak görülmeli ve mutlaka göz önünde bulundurulmalıdır. Bu durum, çok yönlü bilgi teknolojisi kullanımını gerektirmektedir. Ayrıca, arşivsel verilerin analizinden elde edilen bilgilerin; verinin orijinal saklama özelliklerinin yanı sıra, erişime açılmadan önce geçtiği aşamalar hakkında özet bilgiler sunması da mümkündür. Böyle bir yapı, kullanıcıların veri bütünü içinde gizli kalmış ilişkiler hakkında ön bilgilere kolayca erişmesini sağlayacaktır. Bu ön bilgiler, arşivsel aşamalar olabileceği gibi; doğrudan verinin çeşitli analizlerini de içerebilir. Böyle bir yapının oluşturulabilmesi için öncelikle hesaplamalı düşünme ile arşivsel düşünme kavramlarının anlaşılması, sonra da hesaplamalı yöntemler ile entegrasyonunun sağlanması gerekmektedir. Bu tür bir yaklaşımın uygulanabilmesi için ya mevcut bir dijital arşiv probleminin[15] bulunması ya da iyileştirme, keşif veya otomasyon gibi amaçlarla hesaplamalı tekniklerin kullanımına açık bir veri yapısının mevcut olması gerekmektedir. Böylece problemi çözmeye yönelik bir

[14] Computational Archival Science-CAS

[15] Dijital ortamda üretilen, depolanan veya erişime sunulan verilerin yönetimi, korunması, erişilebilirliği, bütünlüğü, doğrulanabilirliği ya da bağlamsal anlamının sürdürülebilirliğiyle ilgili karşılaşılan yapısal ya da teknik zorluklar. Örnek olarak sosyal medya iletilerinin arşivlenmesinde biçimsel bütünlüğün korunamaması, büyük görsel ve metinsel arşivlerde nesne tanımanın manuel olarak mümkün olmaması, belgelerin üretim süreçlerine dair provenans bilgilerinin kaybolması riski veya büyük veri arşivlerinde anlamlı örüntülerin tespit edilememesi gibi durumlar verilebilir.

yapı; hesaplamalı düşünmeyi, arşivsel düşünmeyi, hesaplamalı yöntemleri kullanabilmeyi ve arşiv biliminin temel gerekliliklerini anlamayı gerektirecektir.

Bu tez, arşiv bilimi ile hesaplamalı yöntemlerin entegrasyonunu içeren uygulamalı bir model geliştirmeyi amaçlamaktadır. Çalışma üç temel aşamadan oluşmaktadır:

- Yapılandırılmış veri içeren bir veri setinde, makine öğrenmesi yöntemleri kullanılarak gizli örüntülerin ortaya çıkarılması,
- Uydu tarafından çekilen yer görüntülerinden oluşan bir dijital fotoğraf arşivinin, derin öğrenme tekniklerinden evrişimli sinir ağları ve transfer öğrenmesi algoritmaları ile sınıflandırılması,
- Her iki uygulama fazına ait analiz süreçlerine ait provenans bilgilerinin JSON formatında kaydedilerek, arşivsel standartlara uygun şekilde veri paketlerinin oluşturulması ve bu paketlerin açık kaynaklı bir uzun süreli dijital koruma sisteminde saklanması.

Bu kapsamda tez, hesaplamalı arşiv bilimi yaklaşımı çerçevesinde hem örüntü keşfine dayalı bilgi üretimini hem de uzun süreli dijital saklama süreçlerini içeren bütüncül bir uygulama modeli önermektedir. Tezin çıktısı ise, hem analiz süreçlerini hem de veri paketlerini belgeleyen ve arşivlenebilir bir yapı olan "nitelikli veri havuzu"dur.

1.2. Amaçlar ve Hedefler

Bu tezde, tamamen elektronik ortamda oluşturulmuş ve 1990-2009 yılları arasında Amerika Birleşik Devletleri'nde yapılan iç hat uçuşlarına ait sefer tarihleri, kalkış-varış havaalanı kodları, yolcu sayısı, koltuk sayısı, mesafe gibi operasyonel bilgileri içeren bir dijital arşiv nesnesi (TSV) makine öğrenmesi yöntemleri ile analiz edilmiş; bu veri setinde yer alan değişkenler arasındaki gizli örüntüler ortaya çıkarılmaya çalışılmıştır. Ayrıca ikinci bir dijital arşiv nesnesi olan yüksek çözünürlüklü (600x600) uydu yer görüntüleri arşivi derin öğrenme yöntemlerinden evrişimli sinir ağları sınıflandırma algoritmaları ve transfer öğrenmesi yöntemleri ile analiz edilmiştir. Tüm analizlerde programlama dili olarak Python, IDE[16] olarak ise Jupyter Lab[17] kullanılmıştır. Tezin amaçları aşağıdakiler gibi sıralanabilir:

- Hesaplamalı arşiv bilimi kapsamında geliştirilen yöntem ve tekniklerin, arşiv bilimi literatürüne ve uygulama alanına disiplinlerarası bir yaklaşımla veri analizi, örüntü tanıma ve hesaplamalı düşünme temelli yaklaşımlar kazandırarak, alanın hesaplamalı tarafına dikkat çekmek ve bu yöndeki dönüşümüne katkı sağlamak,

[16] Integrated Development Environment
[17] Özellikle veri bilimi ve bilimsel araştırmalarda etkileşimli hesaplama için tasarlanmış hafif ve web tabanlı bir IDE.

- Hesaplamalı arşiv bilimi yaklaşımının uygulanabilirliğine dair bir örnek model sunarak, dijital arşiv verilerinin yeni nesil hesaplamalı yöntemlerle analizine yönelik bir uygulama örneği geliştirmek,
- Dijital arşiv verilerinin, analiz süreçlerinden geçirilen orijinal formatları, türevleri ve çıktıları ile birlikte uzun süreli saklanması sağlanarak "nitelikli veri havuzu" oluşturmak ve bu havuzu açık kaynak kodlu bir uygulama üzerinde saklamak.

Tüm bu amaçlara ulaşmak için üç hedef belirlenmiştir:

- Hesaplamalı arşiv bilimi uygulamalarının gerçekleştirilmesine imkân tanıyacak, veri analitiği, makine öğrenmesi ve derin öğrenmede yaygın kullanılan bir programlama dili ve kütüphane ekosisteminin belirlenmesi,
- Uygulama sürecinde kullanılan hesaplamalı işlemlerin mantıksal ilişkilerini ve aşamalarını modelleyebilecek kavramsal bir yapının oluşturulması,
- Üretilen analiz çıktılarının, orijinal veri setleriyle birlikte uzun süreli koruma ilkeleri doğrultusunda saklanabilmesi için gerekli yazılım altyapısının belirlenmesi ve kriterler çerçevesinde açık kaynaklı uygun bir uzun süreli saklama sisteminin seçilmesi.

Analizler sonucunda elde edilen çıktılar, girdileri ile birlikte "provenans veri hattı[18]" biçiminde yapılandırılmıştır. Bu veri hattında; dijital provenansı açıkça gösteren ve her iki veri setine uygulanan tüm yapay zekâ algoritmaları, hiperparametreler, girdi birimleri, çıktı bitimleri, donanım ve diğer tüm süreç detayları, işlem aşamaları şeklinde ifade edilmiştir.

1.3. Özgün Değer

Akademik çalışmalarda uzun süredir makine öğrenmesi ve derin öğrenme uygulamalarının fen bilimleri, sosyal bilimler ve tıp gibi farklı disiplinlerarası alanlarda kullanıldığı bilinmektedir. Bu tez, bilgisayar bilimi yöntemlerinin dijital arşiv verileri üzerinde hesaplamalı arşiv bilimi yaklaşımıyla uygulanmasına yönelik Türkiye'de yapılmış ilk kapsamlı örneklerden biri olması bakımından özgün bir katkı sunmaktadır. Çalışma, arşiv bilimi ile yapay zekâ tekniklerinin entegrasyonunda disiplinlerarası bir yaklaşımla, hesaplamalı arşiv biliminin uygulanabilirliğini incelemeyi amaçlamaktadır. Tez ayrıca; dijital çağda bir arşivcinin mevcut sorumluluklarına ek olarak, dijital arşiv koleksiyonunda barındırılan çeşitli formatlardaki verinin ne olduğu, neyi ifade ettiği, erişime sunulmaya uygun olup olmadığı ve veriler arasındaki ilişkilerin açığa çıkarılması yoluyla bir "nitelikli veri havuzu" oluşturabilme becerisine dikkat çekmeye

[18] Provenance Pipeline

odaklanmıştır. Böylece bir arşivci, arşivi oluşturan verileri ve bu veriler arasındaki ilişkileri yapay zekâ yöntemleri ile açığa çıkarabilecektir.

1.4. Problem İfadesi, Araştırma Soruları ve Hipotezler

Tez kapsamında ele alınacak olan temel araştırma problemi, "yapılandırılmış veri ile yapılandırılmamış veri niteliğindeki arşiv materyallerinin barındırdığı gizli ilişkilerin ve sahip olduğu örüntülerin analitik yöntemlerle açığa çıkarılması, orijinal formatı ile birlikte çıktısı sağlanan tüm süreçlerin uzun süreli korunmasının sağlanması ve geleceğe yönelik kullanılabilirliğinin sürdürülebilmesinin yanında tüm özellikleri ve kapsamı ile bir "nitelikli veri havuzu" özelliğinde arşivsel referans noktası olabilmesi için arşiv bilimi mantığında hangi geliştirmelerin veya iyileştirmelerin yapılması gerekmektedir?" şeklinde belirlenmiştir.

Bu tezde cevaplanmaya çalışılan araştırma soruları, genel ve özel araştırma sorusu olarak ikiye ayrılmaktadır. Genel araştırma soruları aşağıdakiler gibi belirlenmiştir:

- Arşivlerin giderek dijitalleşmesi arşiv bilimi için hangi zorlukları doğurmaktadır?
- Dijital veriler, yeni nesil arşivler için nasıl tetikleyici bir temel güç oluşturmaktadır?
- Hesaplamalı arşiv bilimi bağlamında düşünüldüğünde, arşiv bilimciler hangi fırsatları yakalayabilir?
- Hesaplamalı arşiv bilimi uygulamaları için arşivci ve belge yöneticilerine yönelik temel politika ve stratejiler neler olmalıdır?

Özel araştırma soruları ise şu şekildedir:

- Bir arşivsel problemi hesaplamalı arşiv bilimi bağlamında tanımlayabilmek için gerekli kriterler nelerdir?
- Bilinen bir arşivsel problemin çözümüne yönelik hangi hesaplamalı bilim yöntemleri nasıl seçilmelidir?
- Uygulanacak yönteme uygun altyapı oluşturma süreçleri hangi adımları kapsamaktadır?

Temel araştırma problemi ile araştırmanın genel ve özel soruları bağlamında geliştirilen hipotezler ise şu şekilde belirlenmiştir:

- **Hipotez 1:** Dijital arşiv verisinin işlenmesi, analiz edilmesi ve uzun süreli saklanması süreçleri; makine öğrenmesi, derin öğrenme ve hesaplamalı analiz gibi yeni teknolojiler aracılığıyla etkin biçimde gerçekleştirilebilir.

- **Hipotez 2:** Dijital arşiv verisi üzerinde gerçekleştirilen hesaplamalı analizler, girdi ve çıktılarıyla birlikte orijinal veri seti ve manipüle edilmiş türevleri ile versiyonlanarak saklandığında, dijital provenans bütünlüğü korunabilir.
- **Hipotez 3:** Provenans verisi, tüm işlem adımlarıyla birlikte temsil edilerek dijital arşiv sistemlerinde bütüncül bir şekilde saklanabilir.

1.5. Yöntem

Bu tezde kullanılan yöntemler, veri seti özelinde farklılaşmaktadır. İlk olarak, yapılandırılmış TSV formatında veri seti, 1990-2009 arası Amerika Birleşik Devletleri (ABD) iç hat uçuş verilerini içermektedir. 3.606.083 örnekten oluşan ve aylık bazda iç hat uçuş kayıtlarını içeren bu dosyanın metaverileri; kalkış havaalanı kodu, varış havaalanı kodu, kalkış havaalanının bulunduğu şehrin adı, varış havaalanının bulunduğu şehrin adı, her bir uçuş için yolcu sayısı, her bir uçuş için uçaktaki koltuk sayısı, sefer sayısı, mesafe, uçuş tarihi, kalkış havaalanının bulunduğu şehrin nüfusu, varış havaalanının bulunduğu şehrin nüfusu olmak üzere 11 özniteliktir. Python programlama dili kullanılarak çeşitli makine öğrenmesi algoritmaları (doğrusal regresyon, lojistik regresyon, naive bayes, karar ağaçları, k-en yakın komşu, rastgele orman, yapay sinir ağları vb.) aracılığı ile regresyon ve sınıflandırma analizleri yapılmıştır. İkinci olarak, Xia vd. (2017) tarafından geliştirilen ve uydu tarafından çekilen yer görüntülerinden oluşan yapılandırılmamış veri seti; 10.000 adet RGB[19] ve 600x600 piksel boyutlarındaki görüntüleri içermektedir. Veri seti; havaalanı, çıplak arazi, beyzbol sahası, kumsal, köprü, şehir merkezi, kilise, ticari alan, yoğun yerleşim alanı, çöl, tarım arazisi, orman, endüstriyel alan, çayır, orta yoğun yerleşim alanı, dağ, park, otopark, oyun alanı, gölet, liman, tren istasyonu, tatil köyü, nehir, okul, seyrek yerleşim alanı, meydan, stadyum, depolama tankları ve viyadük olmak üzere 30 sınftan oluşmaktadır. Bu veri seti için de Python programlama dili ile birlikte Tensorflow kütüphanesi kullanılarak derin öğrenme yöntemlerinden evrişimli sinir ağları ve transfer öğrenmesi aracılığı ile sınıflandırma ve tahminler yapılmıştır.

Tüm analizler tamamlandıktan sonra, arşivsel verinin orijinal formatı ile birlikte, veri setine uygulanan yapay zekâ algoritmalarının çıktıları da dâhil edilerek, tüm süreci açıklayan bir provenans yapısı oluşturulmuştur. Bu yapı JSON formatında konfigüre edilmiş ve açık kaynak kodlu, uzun süreli dijital koruma yazılımı olan Archivematica içerisinde depolanmıştır.

[19] Red Green Blue

1.6. Kapsam

Bu tezin kapsamı, belirli bir arşivsel problemin çözümüne yönelik olarak yapılandırılmış ve yapılandırılmamış veri setlerinin içerdiği örüntülerin yapay zekâ yöntemleriyle incelenmesi ve ardından süreç çıktıları, orijinal ve türev veri seti ile bir paket halinde uzun süreli koruma standartları sağlayan programda saklanmasını içermektedir. Başka bir deyişle; bu tezin ele aldığı konu; ne sadece arşiv bilimi ne de sadece hesaplamalı bilim olacaktır. Hesaplamalı arşiv bilimi mantığının sunduğu doğrultuda ele alınacak bir arşivsel problemin, hesaplamalı yöntemlerle çözüm aşamaları açıklanmış ve gösterimi sağlanmıştır.

1.7. Tezin Düzeni

Bu tez, yedi ana bölümden oluşmaktadır. Birinci ana bölümde sırasıyla; araştırmanın amaçları ve hedefleri, özgün değeri, temel problem ifadesi, soruları ve hipotezleri, yöntemi, kapsamı ve kısıtları, düzeni, terminolojisi ve kaynakları bulunmaktadır. İkinci ana bölüm; tezin kapsamını oluşturan temel alanların (arşiv bilimi, arşivsel düşünme, hesaplamalı düşünme, hesaplamalı arşiv bilimi, makine öğrenmesi, derin öğrenme vb.) teorik çerçevesi ile literatürde yapılan çalışmaları kapsamaktadır. Üçüncü ana bölüm; tezde izlenen metodoloji hakkında bilgiler vermektedir. Bu bölümde araştırmanın yönteminden ve uygulamaya yönelik olarak oluşturulan modelden detaylı olarak bahsedilmiştir. Dördüncü ana bölümde; birinci faz analizler (yapılandırılmış veriye yönelik makine öğrenmesi yöntemleri) yapılmış ve bulgular değerlendirilmiştir. Beşinci ana bölümde; ikinci faz analizler (yapılandırılmamış görüntü verisine derin öğrenme yöntemleri) yapılmış ve bulgular değerlendirilmiştir. Altıncı ana bölüm ise provenans bilgilerinin oluşturulup paketlenmesi ve Archivematica programına yüklenerek transfer işlemlerinin yapılmasını içermektedir. Yedinci ana bölümde ise her iki fazdaki analizler ile Archivematica ve provenans uygulamasının genel bir değerlendirilmesi yapılarak, literatür bağlamında tartışılmış ve sonuçlandırılmıştır.

1.8. Kısıtlar

Tezdeki kısıtlar; kapsam, veri seti, programlama dili ve kütüphaneleri bağlamında belirlenmiştir. Kısıtlar aşağıdaki maddelerde açıklanmıştır:

- Tez, arşiv süreçlerinin tamamını kapsamamaktadır. Sadece hesaplama, analiz ve saklama süreçlerine odaklanılmıştır. Başka bir deyişle tez; dijital arşiv materyalinin barındırdığı gizli ilişkilerin açığa çıkarılması, bu kapsamda ilgili analizlerin yapılması, açığa çıkarılan ilişkilerin anlamlandırılması, yorumlanması ve uzun süreli saklanmasına yöneliktir.

- Seçilen veri setleri, yapılandırılmış ve yapılandırılmamış özellikte olup, yarı-yapılandırılmış veriler (XML ve JSON, log, metin verisi vb.) bu tezin konusunu oluşturmadığı için analizlere dahil edilmemiştir.
- Analizlerde kullanılan Python programlama diline ait kütüphaneler, sürekli güncelleme ve yapısal değişiklikler nedeniyle gelecekteki çalıştırmalarda uyumsuzluk gösterebilir. Bu durum, araştırmanın tekrarlanabilirliğini ve sürdürülebilirliğini olumsuz etkileyebilecek potansiyel bir kısıt olarak değerlendirilmiştir.
- Tezde 2 farklı dijital nesne kullanılmıştır. Bu nesneler TSV dosyası ile JPEG uzantılı uydu fotoğraflarından oluşan bir derlemdir. Bu türlerden başka metin içeren çeşitli formatlarda (örneğin; gazete arşivleri) da analizler yapılabilmekte fakat bu tezin konusunu oluşturmamaktadır.
- Tez, veri setlerine uygulanacak makine öğrenmesi ve derin öğrenme yöntemlerinin tamamını kapsamamaktadır. Veri setine uygulanacak yöntemler, veri setinin karakteristik özelliklerine (yapı, bağlam, öznitelik vb.) ve hesaplama maliyetlerine göre sınırlı sayıda belirlenmiştir. Ayrıca, veri seti 1'in büyüklüğünden ötürü destek vektör makineleri analizlerinde RAM hataları alınmış ve bu algoritma kullanılamamıştır.

1.9. Yararlanılan Kaynaklar

Bu tezin hazırlanması sürecinde incelenen kaynaklara erişim sağlayabilmek amacıyla, aşağıda yer alan dergi, veritabanı, dijital kütüphane ve akademik platformlardan yararlanılmıştır. Kaynaklar, türlerine göre aşağıdaki gibi sınıflandırılmıştır:

A. Akademik Dergiler

1. American Archivist
2. Archivaria
3. Archival Issues
4. Archival Science
5. Archives and Manuscripts
6. Archival Practice
7. Archive Journal
8. Archives and Records

9. Arşiv Dünyası
10. Bilgi Dünyası
11. Bilgi Yönetimi
12. Journal of Archival Organization
13. Journal of the Association for Information Science and Technology
14. Journal of Contemporary Archival Studies
15. Journal of the South African Society of Archivists
16. Records Management Journal
17. Society of Florida Archivists Journal
18. Türk Kütüphaneciliği

B. Ulusal ve Uluslararası Akademik Veritabanları

1. ULAKBİM TR Dizin Veritabanı
2. ULAKBİM Sosyal Bilimler Veritabanı
3. ULAKBİM Mühendislik ve Temel Bilimler Veritabanı
4. Web of Science
5. Web of Knowledge
6. Scopus

C. Akademik Yayın Platformları ve Dijital Kütüphaneler

1. JSTOR
2. SAGE
3. ScienceDirect
4. SpringerLink
5. Taylor & Francis
6. Wiley Online Library
7. Emerald Insight
8. IEEE Xplore

9. DOAJ (Directory of Open Access Journals)
10. DergiPark
11. EBSCOhost
12. Ebrary

D. Ulusal ve Uluslararası Tez Arşivleri

1. YÖK Tez Merkezi
2. ProQuest Dissertations & Theses Global

Ayrıca, bu tezin hazırlanma sürecinde arama motorları da kullanılmış olup, bu arama motorlarında kullanılan anahtar kelimelerin listesi, Türkçe karşılıkları ile birlikte aşağıda verilmiştir:

1. Arcival Science - Arşiv Bilimi
2. Computational Archival Science - Hesaplamalı Arşiv Bilimi
3. Archival Thinking - Arşivsel Düşünme
4. Computational Science - Hesaplamalı Bilim
5. Computational Thinking - Hesaplamalı Düşünme
6. Computational Social Science - Hesaplamalı Sosyal Bilimler
7. Machine Learning Implementations on Archival Data - Arşiv Verisi Üzerinde Makine Öğrenmesi Uygulamaları
8. Deep Learning Implementations on Archival Data - Arşiv Verisi Üzerinde Derin Öğrenme Uygulamaları
9. Archives in the Digital Age - Dijital Çağda Arşivler
10. Machine Learning in Archive Collections - Arşiv Koleksiyonlarında Makine Öğrenmesi
11. Neural Networks on Archival Images - Arşiv Görsellerinde Sinir Ağları
12. Digital Provenance and Machine Learning Pipeline - Dijital Provenans ve Makine Öğrenmesi Veri Hattı
13. Computer Science and Digital Provenance - Bilgisayar Bilimi ve Dijital Provenans
14. Archival Provenance - Arşivsel Provenans

15. Digital Provenance - Dijital Provenans

Arama motorlarında, belirtilen anahtar kelimelerle yapılan taramalarda, “hesaplamalı arşiv bilimi”, “makine öğrenmesi” ve “dijital provenans” gibi anahtar kelimeler hem İngilizce hem Türkçe kaynaklarda görece daha fazla sonuç verirken; “arşivsel düşünme” kavramının sınırlı sayıda akademik çalışma ile temsil edildiği görülmüştür. Bu farkın temel nedenlerinden biri, ilk gruptaki kavramların daha geniş, yerleşik ve disiplinlerarası kullanımı olan terimler olmasıdır. Makine öğrenmesi ve dijital provenans gibi terimler; arşiv bilimi, yapay zekâ ve bilişim gibi birçok alanda ortak bir dil hâline gelmiştir. Buna karşılık “arşivsel düşünme” terimi, daha çok teorik arşivcilik içinde gelişen özgün bir terim olarak görünmektedir.

İKİNCİ BÖLÜM

2. ARŞİV BİLİMİ VE YAPAY ZEKÂ GENEL LİTERATÜR TARAMASI

Arşiv bilimi ile hesaplamalı yöntemler sunan yapay zekâ kavramlarına ait teorik temellerin bütünsel açıdan değerlendirilebilmesi ve tez kapsamındaki kavramsal ilişkilerin doğru anlaşılabilmesi için öncelikle "bilim", "hipotez" ve "teori" gibi temel kavramların neleri ifade ettiğinin bilinmesi önem arz etmektedir. Bu bağlamda alt başlıklarda sunulan açıklamalar, hem çalışmanın teorik arka planını oluşturan "arşiv bilimi", "hesaplamalı arşiv bilimi, "yapay zekâ", "makine öğrenmesi" ve "derin öğrenme" gibi teorik arka planları netleştirmek hem de ilerleyen bölümlerde yapılan hesaplama ve çıkarımlara teorik bir zemin hazırlama amacı taşımaktadır.

2.1. Bilim Kavramının Genel Değerlendirilmesi

Eski çağlardan beri dünyayı anlamak, uyum sağlayabilmek ve hayatta kalabilmek için sayısız çabalar sergilenmiştir. Bu çabaların bazıları döneminin ihtiyaçlarını karşılamada yeterli olurken, bazıları ise başarısızlıkla sonuçlanmıştır. Ancak tarihsel süreçte birikimli öğrenme yoluyla insanlık, bilgi üretiminde ve yaşam koşullarını iyileştirmede sürekli bir ilerleme kaydetmiştir. Zaman geçtikçe avcı-toplayıcı yaşamdan yerleşik hayata geçilmiş, daha korunaklı evler inşa edilmiş, tarım ve hayvancılık yapılmaya başlanmış, hesaplama ve mühendislik bilgisi gerektiren dev yapılar inşa edilmiş, nehirlere drenaj yapılmış, ordular kurulmuş, makineler icat edilmiş, toplumsal kurallar geliştirilmiştir. Teleskop icat edilmiş ve gökyüzü ile uzay gözlemlenmiş, mikroskop icat edilmiş ve hücre ve atom gibi yeni keşifler yapılmıştır. Var olan gemi üretim teknolojisi geliştirilerek okyanuslarda yol alabilen yeni gemiler icat edilmiş; elektronik devreler geliştirilerek elektrik, elektronik ve bilgisayar gibi bilim dallarında veri iletiminde bir devrim yapılmıştır. Tıp alanındaki ilerlemeler sayesinde de hastalıklara çözümler bulunmuş, aşılar keşfedilmiş ve yeni tedavi yöntemleri geliştirilmiştir. Daha sayılamayacak kadar çok geliştirme, ilerleme ve icat yapılarak yaşam daha da kolaylaştırılmış ve gün geçtikçe de kolaylaşmaya devam etmektedir. Görüldüğü üzere; tüm bu çabaların ortak yönü yaşam konforunu sağlayabilmek için dünyayı anlamak ve anlamlandırmaktır. Özellikle 16. yüzyılda başlayan ve modern bilimin temellerini oluşturan bilimsel devrim ile gelen gözlem, deney ve eleştirel akıl yürütme; bilimsel süreçlerin bir sistematiğe oturtulmasını sağlamıştır. Bilimsel devrimin başlangıcı olarak literatürde Nicolaus Copernicus'un 1543 tarihli "*De Revolutionibus Orbium Coelestium*[20]" adlı yayını

[20] Göksel Kürelerin Devrimleri Üzerine

gösterilmektedir (Capek, 1968, s. 114; Bussard, 2005, s. 691; Valdez, 2014, s. 367; Teich, 2015, s. 11; Casadevall ve Fang, 2016, s. 1; Shapin, 2018, s. 3; Rankin ve Weise, 2022, s. 54). Ardından 17. yüzyılda başlayan Aydınlanma Çağı ile bilim; gerçekliğin altında yatan ilkeleri, kalıpları ve yasaları ortaya çıkarmaya çalışarak, insanlığın doğa ile etkileşim biçimini sistematiğe dökmüştür. Tüm bu gelişmeler, bilimin özünde bir ampirik bir uğraş olduğunu; yani kanıtlara dayanması, ölçülebilmesi ve gözlemlenebilmesi nedeniyle verilere ve hesaplamalara dayandığını göstermektedir. Böylece, elde edilen sonuçlar ve sahip olunan bilgiler zaman içerisinde düzeltilip güncellenebilir ve bilimsel anlayış giderek tekrarlanabilir ve doğru hale gelir. Bu durum, metodolojiye ve nesnelliğe olan bağlılık ile rahatlıkla açıklanabilmektedir. Nesnellik, bilimsel süreçlerde duygu ve düşüncelere bağlılıktan uzaklaşma ve bilimsel bir bilginin evrensel olarak uygulanabilir hale getirilmesi demektir (Stamenkovic, 2023). Tüm bu sayılanlara ek olarak; bilimsel süreçler sonunda elde edilen bilgiler günümüzde çeşitli endüstrilerde, tıpta, teknolojide ve günlük yaşamda kullanılmaktadır. Örneğin; tıp alanındaki bilimsel gelişmeler neticesinde yaşam süresi ve kalitesini önemli ölçüde artmış; fizik ve kimya alanındaki ilerlemeler sayesinde de elektrik, elektronik, bilgisayar bilimleri, inşaat ve çevre sistemleri gibi modern toplumu şekillendiren teknolojiler ortaya çıkmıştır. Bilimin, olgular veya teoriler toplamından fazlası olmasıyla açıklanabilecek bu durum, sorunlara sistematik yaklaşma biçiminin yansıması olarak düşünülebilir (Kaushik ve Walsh, 2019).

McGraw Hill sözlüğünde bilim; *"doğanın ve doğal olayların incelendiği; gerçeklerin gözlemlendiği, sınıflandırıldığı ve genellikle nicel yasaların formüle edilip doğrulandığı bir kavramdır. Ayrıca mantıksal akıl yürütme ve veri analizinin doğal olaylara uygulanmasını da içerir"* (Bridgman ve Holton, 2024) şeklinde tanımlanmaktadır. Oxford Advanced Learner's sözlüğünde ise bilim; *"fiziksel ve doğal dünya ile toplumun yapısı ve davranışlarının, özellikle gözlem ve deney yoluyla incelenmesi"* olarak tanımlanmıştır (Hornby, 1995, s. 1050). Merriam-Webster sözlüğünde ise bilim; *"bilgi sahibi olmak; bir çalışma nesnesine sahip bir bilgi alanı; özellikle bilimsel yöntemle elde edilen ve test edilen genel doğruları veya genel yasaların işleyişini kapsayan bilgi"* olarak tanımlanmıştır (Mish, 2004, s. 646). Amerika Birleşik Devletleri Ulusal Bilimler Akademisi[21] bilimi; *"doğal olayların test edilebilir açıklamalarını ve tahminlerini oluşturmak için kanıtların kullanılması ve bu süreçle üretilen bilgi"* olarak tanımlamıştır (National Academy of Sciences, 2008, s. 10).

[21] National Academy of Sciences

Mickens ve Patterson (2016, s. 2) bilim kavramını; ilk insanların çeşitli fiziksel olgulara açıklama aradığından hareketle tanımlamaya çalışmıştır:

- Güneşin doğuşu ve batışı,
- Biri diğerinden daha büyük olan iki nesnenin çarpışması,
- Su gibi bir sıvıya uygulanan ısının, sıvının ısınmasına neden olması.

Böylece bilim; fiziksel evrende var olan örüntülerin sistematik olarak gözlemlenmesi, üretilmesi, analiz edilmesi ve modellenmesi olarak tanımlanabilir.

Yukarıda verilen tanımlar, bilimin kolektif bir yapıda profesyonel süreçlerle ilerleyen bir nitelikte olduğunu göstermektedir. Çünkü doğayı anlamlandırmak için yapılan tüm çalışmalar belirli bir sistematiklik ve bölümlendirme içerisinde yürütülmezse hangi verinin doğru analiz edilip edilmediği veya bilimsel süreçler sonunda elde edilen bilgilerin geçerli olup olmadığı konusunda ikilemlerle karşılaşılır. Böyle bir kaos durumunda da bilimden bahsedilemez. Ayrıca günümüzdeki gibi tıp, fen bilimleri, mühendislik ve sosyal bilimler alanlarının ayrılması, bu sistematikliğin güzel bir örneği olarak düşünülmelidir. İşte bu yüzden bilimin tüm süreçlerinin tekrarlanabilir, kaydedilebilir, paylaşılabilir, doğrulanabilir ve eleştirilebilir olması bilimsel ilerlemeler için mutlaka gereklidir. Böylece, Silva'ya göre (2024) bilimin genel bir bilgi arayışından ziyade daha özel, belirli bir bilginin aranması süreci olarak tanımlanması yanlış olmayacaktır. Clark, Golder ve Golder'ın da aktardığı üzere (2018, s. 58); Karl Popper'ın deyimiyle bilim; bir bilim insanının fikirlerini sürekli olarak mantık ve kanıtın soğuk ışığına tabi tuttuğu "pervasızca eleştirel" bir bilgi arayışıdır.

Yukarıdaki tanımlar ve açıklamalar neticesinde bilim kavramının karakteristik özellikleri aşağıdaki gibi özetlenebilir:

1. ***Bilim evrenseldir:*** Bilim, evrende bulunan temel kuralların her yerde aynı olduğu tek bir sistem varsayar. Evrenin bir bölümü incelenerek elde edilen bilgi, diğer bölümlere de uygulanabilir (Drenth, 2004, s. 238).
2. ***Bilimsel fikirler ve sonuçlar sabit değildir, değişebilir:*** Bilim, mevcut teorilerin başarısız veya kusurlu olduğu ispatlandığında değişir. Bilimsel sonuçlar sürekli test ve değerlendirme döngüsü ile gözden geçirilir. Bir teori çürütülebilir, değişebilir ya da yeni teoriler eskilerinin yerini alabilir. Böylece bilim kendi kendini düzelterek ilerler. Bilimde düşüncenin esnekliği, bilimsel sürecin özüdür. Başka bir deyişle bilim, dogmatizmi reddeder (Pitt, 1998).

3. ***Bilimin sonuçları tutarlı ve güvenilirdir:*** Bilim, deneysel süreçlerin tekrarlanabilir şekilde tasarlandığı bir yöntem izler. Bu, deney koşullarının sabit olduğu farklı zamanlarda yapılan çalışmaların da aynı sonuçları vermesi gerektiği anlamına gelmektedir. Deneyler arasındaki tutarlılık, bulguların güvenilirliğini doğrulamaya yardımcı olur. Bilimsel sonuçlar; kanıta, gözleme ve mantıksal akıl yürütmeye dayandıkları için güvenilir kabul edilmektedir (Munafò ve Flint, 2010).
4. ***Bilim sağlam kanıtlar gerektirir:*** Bilim, süreçlere dair her aşamada tekrarlanabilir, ölçülebilir ve doğrulanabilir kanıtlara, yani doğru verilere dayanır. Bilimsel bir kanıt, teori veya hipotezi desteklemeye veya çürütmeye yarar. Bu tür kanıtların deneysel kanıt olması ve bilimsel yönteme uygun olarak yorumlanabilir olması gereklidir (Taper ve Subhash, 2004, s. 529).
5. ***Bilim duygu ve düşüncelerden etkilenmez:*** Bilimsel süreçler, gerçeği arama yolunda tüm duygusal tepkilerden ayrılmalıdır. Dolaysıyla; bilimsel ilerlemeler önyargı, kişisel inanç, politik görüş, estetik yargılar vb. insani faktörlerden arınmıştır ve bunlardan etkilenmez (Conca, 2014; Barbalet, 2002).
6. ***Bilim mantık ve hayal gücü ile yapılanır:*** Bir bilimsel sürecin başlangıcı, o sürece ait önce hayal gücü ve sonra mantıksal düşünme ilkelerine uygunluk gerektirmektedir. Sadece verilerin doğruluğu, bilimsel gelişme ve ilerleme için yeterli değildir. Bilimsel sonuçlar, kendiliğinden ortaya çıkamaz. Mantık, belirli hipotezler veya modellerle tutarlı tahminler yapmak için gereklidir. Fikirleri test ederken mantığa ihtiyaç vardır, ancak önce test edilecek fikirlerin ortaya konması gerekmektedir. Bir bilimsel süreçte veriler kavramsal bir şema açısından yorumlanmalıdır (Taber, 2017, s. 32).
7. ***Bilim tahmin ederek ilerler:*** Tahmin, bilim sürecinin merkezinde yer alır ve bilimsel yöntemin temellerinden birisidir. Bilim insanları bir teoriye dayalı tahmini, doğadaki veya laboratuvardaki gerçek olaylarla karşılaştırarak fikirlerini test eder (Sarewitz ve Pielke, 1999, s. 123).

Bir bilimsel süreç sistematik ve çok yönlüdür. Sistematikliği sağlayan şey de sürecin mantıksal akıl yürütme hiyerarşisine sahip olmasıdır. Başka bir deyişle bir bilimsel süreç, belli başlı mantıksal aşamalardan oluşmakta ve tüm bu aşamalar tekrarlanabilir niteliğe sahip olmaktadır. Bu bağlamda bir bilimsel süreç, aşağıdaki adımlardan oluşmalıdır:

- Gözlemler neticesinde tespit edilen problemin tanımlanması,
- Bu probleme yönelik bir hipotez geliştirilmesi,
- Bu hipotezin sürekli test edilmesi,

- Hipotezin desteklenmesi durumunda teori geliştirilmesi,
- Teorinin kesin olarak ispatlanması durumunda bilimsel bir "yasa"ya dönüşmesi ve değişmemesi,
- Hipotezin desteklenmemesi durumunda ya hipotezin değiştirilmesi ya da test aşamasına geri dönülmesi.

2.2. Hipotez ve Teori Kavramlarının Genel Değerlendirilmesi

Tarihte bilimsel atılımların, hayal gücü ve mantıksal akıl yürütme ile başladığı bir gerçektir. Ancak, bunun devam edebilmesi için gözlemler, kanıtlar ve tutarlı verilerle desteklenmesi gereklidir. Tüm bu gereklilikler, belirli bir araştırmaya yönelik sürekli sorular sormak ve cevaplarının aranması ile sağlanır. Belirli bir probleme yönelik ortaya atılan bu tür sorular bağlam olarak bir hipotez oluşturur. Başka bir deyişle; araştırma sorularına yönelik test edilebilen önermeler, hipotez olarak adlandırılabilir. Bir hipotez; Polit ve Beck'e göre (2017, s. 130) gözlemlenen veya gözlemlenemeyen değişkenler arasındaki ilişki hakkında bir tahmin ifadesidir. A Complete Dictionary of the English sözlüğünde hipotez; *"bir varsayım, kanıtlanmamış bir prensip altında oluşturulmuş bir sistem"* olarak tanımlanmaktadır (Sheridan, 1798, s. 329). Ary vd.'ne göre (2009, s. 7) ise hipotez, değişkenler arasındaki ilişkileri tanımlayan ve geçici olarak doğru olduğu varsayılan bir ifade olarak tanımlanmıştır. Tanımların ortak özelliği; bir hipotez oluşturabilmek için belirli bir probleme yönelik olarak değişkenler ve değişkenlerin arasında bir ilişki kurulmasının gerekliliğidir. Örneğin; ilköğretim öğrencilerinin kitap okuma alışkanlıkları ile ilgili bir araştırma yapan bilim insanı, araştırmasına şu soru ile başlayabilir: Düzenli kitap okuma alışkanlığı, öğrencilerin derslerindeki başarıyı nasıl etkilemektedir? Bu araştırma sorusu, şu şekilde bir hipoteze dönüşebilir: Düzenli olarak günde 2 saat kitap okuyan öğrencilerin derslerindeki başarı, günde 2 saatten az kitap okuyan öğrencilere göre daha yüksektir. Bu hipotezde, iki olgu dikkat çekmektedir: Kitap okuma alışkanlığı ve ders başarısı. Başka bir deyişle; bu iki olgu da bir değişken olarak nitelendirilebilir. Çünkü bir değişkendeki farklılık, diğer değişkeni yapısal olarak etkilemektedir. Ayrıca, değişkenleri kendi içinde sınıflandırmak da sürecin sağlıklı sonuçlar vermesi açısından bir gerekliliktir. Bu örnekte kitap okuma alışkanlığı, ders başarısını etkileyeceği için bağımsız değişken; ders başarısı ise kitap okuma alışkanlığından etkileneceği için bağımlı değişken olarak nitelendirilebilir. Bu bağlamda bir bağımlı değişken, bağımsız değişkenden etkilenen değişkendir. Başka bir deyişle bağımlı değişken, bağımsız değişkene bağlıdır (Kaur, 2013, s. 36). Bağımsız değişken ise Dictionary of Statistics&Methodology sözlüğünde; *"bir çalışmada varsayılan neden; başka bir değişkenin değerlerini tahmin etmek*

veya açıklamak için kullanılabilen bir değişken; araştırmacı tarafından manipüle edilebilen ve bu manipülasyonun başka bir değişken -bağımlı değişken- üzerinde bir etkisi olacağını öngören bir değişken" olarak tanımlanmıştır (Vogt, 1993, s. 151). Bağımlı değişkenin bağımlılığı; açıkça bağımsız değişkene bağlı olmasından dolayıdır. Literatürde, araştırmacıların çoğunlukla anlamaya çalıştığı değişken, bağımlı değişkendir (Perepiczka, Chandler ve Becerra, 2011; Carter, Brown ve Simpson, 2017; Motta, 2019; Berndt vd., 2021; MacArthur ve Santo, 2022; Dani ve Al Quraan, 2023; Casinillo ve Tavera, 2023). Bu bağlamda düşünüldüğünde hipotezin temel mantığı; araştırma sorusuna muhtemel bir cevap sunması ve cevaba yönelik tahmini test etmesidir.

Teori ise, genel bir davranış veya yapı hakkında önemli ölçüde geniş bir spesifik örnek yelpazesinde geçerli olduğu varsayılan sıralı bir iddialar kümesidir (Sutherland, 1975, s. 9). Stewart, Harte ve Sambrook'a göre (2011, s. 222) teori; *"bir şeyi açıklamaya yönelik bir varsayım ya da fikirler sistemi"* olarak tanımlanmıştır. Collins English Dictionary sözlüğünde ise teori; *"bir sonuç üretmek için kullanılan kurallar, prosedürler ve varsayımlar sistemi; soyut bilgi veya akıl yürütme; spekülatif veya varsayımsal bir görüş veya fikir; ideal veya varsayımsal bir durum; genel anlamda çok çeşitli bağlantılı olguları açıklamak ve tahmin etmek için mantıksal veya matematiksel argümanlarla ilişkilendirilen bir dizi hipotez; hipotez için teknik olmayan bir isim"* olarak tanımlanmıştır (Treffry, 1998, s. 1587). Tanımlardan, bir teorinin; olguların veya kavramların nasıl ve neden var olduklarına dair açıklamalar sağladığı çıkarılabilir. Nixon (2004, s. 34), teori kavramının üç farklı yönü olduğunu öne sürmektedir:

- ***Niyet ve eylemin analitik yorumu:*** Teori; eylemlerin anlamını, niyetini ve bu eylemlerin olası etkilerini anlayabilmek için kavramsal araçlar sağlama potansiyeline sahiptir.
- ***Alternatif eylem yollarının değerlendirilmesi:*** Teori; alternatif hareket tarzları ve bu hareket tarzlarını değerlendirmek için gerekli kaynakları sağlar. Ayrıca; şimdiye kadar karmaşık olmayan bir şekilde uyumlu olduğu varsayılan eylem biçimlerinin uyumsuzluklarına da işaret edebilir. Başka bir deyişle; seçeneklerin tartışılmasını sağlar.
- ***Uygulamanın altında yatan ilkelerin gerekçelendirilmesi:*** Teori, yalın ifadelerle olguların ve kavramların açıklanmasına yardımcı olur. Böylece teori; hem şeffaflık sağlar hem de iletişimsel bir özellik kazanır. Bu özellik, eylemlerin nasıl anlamlandırılması gerektiğini gösterir.

Bu bağlamda bir teorinin; kavramsal bir model özelliğinde olduğu üzerinde durulabilir. Teori; belirli bir yapı içerisindeki çeşitli olgular veya öğelerin birbirleriyle olan ilişkileri hakkında bir

akıl yürütme sağlayarak, gerçekliği birleştirir. Bu duruma örnek olarak; Alfred Wegener tarafından 1912'de ortaya atılan levha tektoniği teorisi verilebilir. Bu teori kısaca, kıtaların sürüklendiğini ve birbirine baskı yaptığını; ayrıca ayrı birer levhaya sahip olduğunu öne sürmektedir. Böylece, kıtaların hareketlerindeki değişimleri anlamak için, jeoloji alanında bir kavramsal model sağlar. Teorinin bir başka üzerinde durulması gereken özelliği ise bir hipotezler dizisi olmasıdır. Teoriler çoğunlukla deneysel olarak test edilebilir, doğrulanabilir veya yanlışlanabilir hipotezler üretir. Bir teorinin sahip olması gereken evrensel özellikler aşağıda açıklanmaktadır:

1. ***Teoriler içsel tutarlılığa sahip olmalıdır:*** Bir teori, teoriyi tanımlayan ifadelerinin (önermeler, tahminler, varsayımlar ve kavramlar) hiçbirinin birbiriyle çelişmediği durumda içsel tutarlılığa sahip olarak nitelendirilebilir. Örneğin, bir ifade ikinci bir ifadenin tersini iddia ettiğinde, doğal olarak mantıksal bir çelişki oluşur. Bu durumda en fazla yalnızca biri doğru olabilir. Bir teori, hem bir ifadenin hem de o ifadenin tersinin doğru olduğunu söylediğinde, o teori tutarsız olarak nitelendirilir çünkü her iki ifadenin de doğru olması imkânsızdır. Teoriler, ele alınan olgunun açık ve içsel olarak tutarlı bir açıklamasını mutlaka sunmalıdır (McCrae vd., 2010). Teorinin bu özelliği kritik olarak nitelendirilebilir çünkü bir teorinin tanımlayıcı ve açıklayıcı gücü ile güvenilirliği, her zaman mantıksal içyapısına ve ifadelerinin iç uyumuna dayanır.

2. ***Teoriler deneysel kanıtlara dayanmalıdır:*** Bir teorinin yerleşmesi, kabul görmesi ve bir yasaya dönüşebilmesi için titizlikle test edilmesi ve doğru tahminlerde bulunulduğuna dair çok sayıda gözlemsel veya deneysel kanıta sahip olması gerekir (Edgar ve Manz, 2017, s. 222). Teorinin sağlamlığı, deneylerden elde edilen kanıtlarla doğru orantılıdır.

3. ***Teorilerin tahmin gücü yüksek olmalıdır:*** Bir teorinin değeri genellikle gelecekteki olayları veya davranışları tahmin etme kabiliyeti ile ölçülür. Bir olayın meydana gelip gelmeyeceği kesin olarak bilinmese de, meydana gelme olasılığı belirlenebilmelidir, çünkü bu şekilde daha iyi tahmin gücüne sahip bir teori sağlanmış olur (Hyland, 1981, s. 18). Örneğin; Albert Einstein tarafından 1915 yılında geliştirilen genel görelilik teorisi ile; ışığın sapması, kara delikler ve kütleçekim dalgaları konularında, hem geçmiş hem de gelecek olaylarla ilgili şaşırtıcı derecede isabetli tahminler yapılmıştır. Arthur Stanley Eddington, 29 Mayıs 1919'daki güneş tutulması sırasında yaptığı gözlemler ile, genel göreliliğin ışığın kütleçekim alanında bükületeceği tahminini doğrulamıştır (Kennefick, 2012).

4. ***Teoriler yanlışlanabilmelidir:*** Filozof Karl Popper (1962, s. 36), "Conjectures and Refutations: The Growth of Scientific Knowledge" adlı eserinde; *"bir teorinin her gerçek testi onu çürütme veya reddetme girişimidir. Test edilebilirlik, yanlışlanabilirliktir; ancak test edilebilirliğin dereceleri vardır: Bazı teoriler diğerlerine göre daha test edilebilir, daha fazla çürütülmeye maruz kalır; deyim yerindeyse daha büyük riskler alır"* diyerek, teorinin testler yoluyla çürütülmeye açık olması gerektiğini vurgulamıştır. Eğer bir teori yanlışlanamaz bir özelliğe bürünürse, bilimsel olmaktan çıkar ve dogmatik bir hale dönüşme riski oluşur.

5. ***Teoriler genelleştirilebilmelidir:*** Teorilerin genellenebilirlik özelliği, onların yalnızca belirli durumlar veya sınırlı örnekler için değil, daha geniş olgu veya olgu kümeleri için de geçerli olabilmesini ifade eder. Bir teorinin uygulanabilirlik alanı ne kadar genişse, o teorinin gücü ve açıklayıcılığı o kadar artar (Wacker, 1998, s. 365). Örneğin, arşiv teorisinin yalnızca belirli bir materyal türüne (örneğin kağıt belgeler) değil, dijital belgeler, görsel-işitsel materyaller ve veri kümeleri dahil olmak üzere tüm belge ve bilgi türlerine uygulanabilir olması, teorinin genellenebilirlik niteliğini pekiştirmektedir. Bu durum, teorinin farklı bağlamlarda ve teknolojik gelişmeler karşısında geçerliliğini sürdürebilme kapasitesini göstermektedir.

6. ***Teoriler uyarlanabilir olmalıdır:*** Bilimsel ilerlemelerde yeni kanıtlar bulundukça, teoriler de bu yeni gelişmelere uyum sağlayacak kadar esnek olmalıdır. Çalışmalarda, yeni bulguları destekleyemeyen teoriler revize edilmeli veya değiştirilmelidir. Böylece uyarlanabilirlik özelliği ile teoriler, bilimsel araştırmanın ilerlemesini sağlamış olur (Renzulli, 2020).

Bilimsel araştırmalar hiç durmadan devam ettiğine göre, teorilerin de durağan olmadığı, bir devinim içerisinde olduğu rahatlıkla söylenebilir. Araştırmalar sonucunda yeni kanıtlar ortaya çıktıkça, teoriler güncellenebilir veya evrim geçirebilir. Bir teori, ortaya atıldığı andan itibaren çok sayıda teste tabi tutularak daha da geliştirilebilir ya da güncel kanıtlar karşısında başarısız olursa terk edilebilir ya da değiştirilebilir. Filozof Thomas Kuhn (1962) tarafından yayımlanan "The Structure of Scientific Revolutions[22]" adlı kitapta; bilimsel ilerlemelerin doğrusal olarak değil, devrimsel değişimlerle gerçekleştiğini savunulmaktadır. Başka bir deyişle; bir teorinin, bilimsel bir topluluk tarafından onaylandığında bir "paradigma" oluştuğunu; zamanla

[22] Bilimsel Devrimlerin Yapısı

karşılaşılan anormallikler ve yeni kanıtlarla çeliştiğinde ise mevcut paradigmanın yeni bir paradigma ile yer değiştirdiğini ifade edilmektedir.

Bilimsel teorilerin zaman içinde yeni bilgi ve ihtiyaçlara göre evrim geçirdiğini ortaya koyan bu yaklaşım, yalnızca doğa bilimleriyle sınırlı değildir. Aynı paradigma değişimi, bilgi ve belge yönetimi alanında da gözlemlenmektedir (Cibaroğlu ve Yalçınkaya, 2019; Binici, 2019). Bu bağlamda, arşiv bilimi de tarihsel süreçte toplumsal, teknolojik ve yönetsel gereksinimlere göre sürekli dönüşüm geçiren bir alan olmuştur. Özellikle dijitalleşme ve büyük veri çağında, geleneksel arşiv ilkelerinin revizyonu ve yeni teknolojilerle entegrasyonu zorunlu hâle gelmiştir. İşte bu noktada, arşiv biliminin teorik temelinin anlaşılması ve çağdaş yaklaşımlarla entegrasyonunun ortaya konulması, bu tezin hem teorik hem de uygulamalı boyutlarının kavranması açısından büyük önem taşımaktadır. Aşağıdaki bölümlerde, bu çerçevede arşiv biliminin temel kavramları ve arşiv teorisinin dinamikleri açıklanmıştır.

2.3. Arşiv Biliminin Genel Değerlendirilmesi

Belirli bir form üzerinde bulunan kayıtlı bilgilerin en eski kanıtları, Mezopotamya'da M.Ö. 8000 civarında tarımın ortaya çıkışıyla aynı zamana denk gelmektedir (Beynon-Davies, 2009). Bu türden ilk kayıtlar, tarımsal ürünlerin hem miktarını hem de türlerini tespit edebilmek ve ayrıca bunları, daha sonra tekrar kullanmak için çeşitli şekillerde simgeler içermekteydi. Yazının ortaya çıkışı; hafıza oluşturmayı ve bilgi birikimini daha da etkinleştirmiş ve binyıllar içinde büyük ve karmaşık sosyal yapıların geliştirilmesini sağlamıştır. M.Ö. 4000'lerin sonlarında yazının bulunması; küçük grupların yerleşik yaşama geçmesiyle şehir devletlerinin kurulması, ilk karmaşık sosyal ilişki ve hiyerarşilerin başlangıcı ve tarımsal üretime yönelik emek kavramının ortaya çıkışıyla aynı zamana denk gelmiştir (Kiger, 2025). Birlikte ve gruplar halinde yaşamaya başlamanın getirdiği kurallar bütününün de ortaya çıkmasıyla, özellikle devlet gücü ve otoritesinin, askeri ve tarımsal üretim başta olmak üzere, gruplar üzerindeki sosyal ilişkileri düzenleyici rolü ile; sözlü bilginin kaydedilmesi, saklanması ve gerektiği durumlarda tekrar erişilebilmesi sağlanmıştır (Blerk, 2018). Yazılı dilin ortaya çıkmasıyla birlikte, kaydedilebilecek ve iletilebilecek her türlü işin kapsamı; zaman içerisinde simgelerin verdiği anlamı aştığından dolayı, bir form (biçim) üzerine kaydedilen bilgilerin hem belirli bir şablonu olan belgelere dönüşmesini sağlamış hem de bu belgelerin rollerini ve amaçlarını genişlemiştir (Woods, 2010, s. 19). Bu bağlamda; arşivlerin ilk köklerinin, antik uygarlıklarda idari ve hukuki yönetim sistemlerin temel faaliyetleri kapsamında ortaya çıktığı söylenebilir. Yüzyıllar içerisinde ise; yazının da bulunmasının etkisiyle idari, yasal ve tarihi önemi olan belgelerin toplanması, resmi otorite olan devlet güvencesinde güvenilir bir kanıt niteliğinde

korunması ve bunların yönetimini içeren ilk uygulama örneklerinin, bilimsel bir nitelikten yoksun, ancak tamamen pratik ve pragmatik amaçlarla başladığı söylenebilir.

Antik çağlardan beri özellikle belirli bir kurumsal yapıya sahip organizasyonların (şehir devletleri, imparatorluklar, modern çağlarda şirketler vb.) zaman içerisinde yapmış oldukları faaliyetleri gerek belgelendirmek gerekse tekrar kullanmak için sakladıkları bilinen bir gerçektir. Bunun altında yatan temel neden; organizasyonun devamlılığının yapılan faaliyetlere bağlı olması ve bu faaliyetlerin ayrıntılarına istenildiğinde eksiksiz bir biçimde ulaşma isteği olarak gösterilebilir. Organizasyonun türüne bakılmadan, bu faaliyetlerin yüzyıllar boyunca kesintili veya kesintisiz olarak sürdüğü düşünülürse, zaman içerisinde bir yığın haline dönüşen belgelerin yönetiminin (düzenlenmesi, seçilmesi, ayıklanması, imhası vb.) hangi koşullara göre yapılacağı önem arz etmektedir. Peki, çok fazla sayıda olmasının yanında çeşitli özelliklerinden ötürü önem arz eden belgeler, nasıl bir anlayış ile yönetilmelidir? Arşiv bilimi, bu gibi sorulara yanıt arayan bir bilim dalıdır ve genel olarak arşivlerin yönetilmesi ile ilgilenir. Ayrıca, kalıcı bir değere sahip fiziksel veya elektronik formatta bulunan her türlü materyalin (kayıt, belge, fotoğraf, e-posta vb.) yönetimine dair teori, kavram ve uygulamalar bütünü olarak da düşünülebilir. Biraz daha detaylandırıldığında ise arşiv bilimi; arşivleri ve materyalleri koruma, düzenleme, yönetme ve erişime açmaya yönelik hem teorik hem uygulama mantığını içermektedir. Ayrıca; tarihi öneme sahip değerli el yazmaları, resmi belgeler, uluslararası anlaşmalar, tarihi tablolar ve günümüzde oldukça yüksek kullanıma sahip elektronik araştırma verileri gibi daha pek çok materyali gelecek nesillerin erişebilmesi için uzun süreli korumaya dair yöntemler de sağlamaktadır. Arşiv bilimi; belgelerin ve arşivlerin sistematik yönetimi için ilkeler ve teoriler geliştirmek amacıyla 19. yüzyılın sonlarında şekillenmeye başlamıştır.

Arşiv bilimi; Duranti ve MacNeil'e göre (1996, s. 47) 19. yüzyılda diplomatik biliminden doğan; materyallerin belgesel ve fonksiyonel ilişkileri ile kontrol edilme ve iletilme biçimleri açısından incelenmesine yönelik bir kavramlar ve yöntemler bütünüdür. Encyclopedia of Archival Science içerisinde ise arşiv bilimi; *"insan düşüncelerinin, duygularının ve faaliyetlerinin hem ürünü hem de aracısı olarak sürece bağlı bilginin çeşitli bağlamlarda bilimsel incelenmesini içeren akademik ve uygulamalı bir disiplindir. Çalışma alanı; arşiv kurumlarını, kişisel belgeleri, resmi belgeleri, çeşitli toplulukların, devlet kurumlarının ve diğer resmi kuruluşların arşivlerini ve genel olarak arşiv malzemeleri ile hem belgelerin kendisini hem de üretme, yönetim, kullanım ve sosyokültürel bağlamlarını kapsar"* şeklinde tanımlanmış ve açıklanmıştır (Duranti ve Franks, 2015a, s. 84).

Thomassen'e göre (2001, s. 382) arşiv biliminin nesnesi hem içeriğin kendisi hem de bu içeriği üreten ve yapılandıran süreçlerdir. Arşiv biliminin amaçları bu bağlamda arşiv kalitesinin oluşturulması ve sürdürülmesidir. Bu; belgelerin dayanıklılığını, bunları üreten iş süreçlerini ve karşılıklı bağlarını kapsamaktadır. Arşiv biliminin metodolojisi ise bir yandan kaydedilen bilginin işlevi ile diğer yandan biçimi, yapısı ve provenans bağlamı arasındaki bağlantıların analizi, kaydedilmesi ve bunların sürdürülmesidir. Arşiv metodolojisi; arşiv teorisinden geliştirilen ilke ve prosedürlerin uygulanması yoluyla belge yönetimi için gerekliliklerin oluşturulması, belgelerin korunması ve kullanımı ile politikalar geliştirmeyi de içerir (Duranti, 1999).

Çeşitli faaliyetler sonucunda kendiliğinden oluşan belgelerin, bir düzen olmadan, uzun yıllar boyunca saklandığını; belirli bir yığın haline dönüştükten sonra ihtiyaç duyulduğunda, istenilen bilgiyi ve belgeyi bulma konusunda büyük problemlere yol açtığını ve bunların yanında doğal şartlardan ötürü saklama ortamında bulunan belgelerin deformasyona (örneğin; tarihi bir belgedeki mürekkebin erimesi) uğrayabileceği düşünüldüğünde, bu yığınların detaylı bir analizden geçirilerek; öncelikle hangi belgelerin saklanması gerektiğine karar verilmesi, ardından saklama ortamında nasıl bir sınıflandırmaya tabi tutulacağı, bunların ne kadar süre ile saklanması gerektiği ve hangilerinin ne şartlarda imha edileceğine yönelik olarak karar verilmesini sağlayan politika ve ilkelere ihtiyaç duyulmaktadır. Oldukça uzun zamandır çeşitli şekillerde uygulanan bu politika ve ilkeler, ilk kez 19. yüzyılda Fransız ve Alman arşivciler tarafından belgelerin oluşturulduğu anda meydana gelen ilişki biçimlerine uygun olarak düzenlenmesini öngören ve arşivlere devredilen belgelerin bütünlüklerinin korunmasına dayanan provenans ilkesi aracılığıyla oluşturulmuştur (Ciechanowski, 2021).

Cook, (2001, s. 4) arşiv biliminde bir paradigma değişiminin meydana geldiğini vurguladıktan sonra değişimin merkezinde; belgeleri statik fiziksel nesneler olarak görmektense, onları dinamik sanal kavramlar olarak anlamaya doğru bir geçişin yer aldığını; belgelerin insani veya idari faaliyetin pasif çıktıları olmaktansa, insani ve kurumsal hafızanın oluşumunda aktif aktörler olduğunu; belgelerin oluşturulduğu bağlamı istikrarlı hiyerarşik organizasyonlar içinde görmektense, belgeleri iş akışı işlevselliğinin akıcı ağları içinde değerlendirmenin gerekliliğini vurgulamaktadır. Arşivciler için ise bu kabuk değişimi; kendilerini belgesel mirasın pasif koruyucuları olarak tanımlamaktansa, kollektif hafızayı aktif olarak şekillendirmedeki rollerine geçişi vurgulamaktadır. Başka bir deyişle teorik arşiv; çıktıdan sürece; yapıdan fonksiyona; arşivlerden arşivlemeye; belgeden belge bağlamına; idari faaliyetin doğal kalıntısı veya pasif yan ürününden toplumsal hafızanın bilinçli olarak inşa

edilmesine ve aktif olarak arşivlenmesine doğru ilerlemiştir. Arşivlerdeki değişimin yönleri, Tablo 1'de gösterilmektedir.

Tablo 1. Arşivlerdeki Paradigma Değişiminin Yönleri

	Eski Anlayış	**Yeni Paradigma**
Belgeler	Statik fiziksel nesneler	Dinamik sanal kavramlar
	Faaliyetlerin pasif çıktıları	Hafızanın oluşumunda aktif aktörler
Belge Bağlamı	Hiyerarşik organizasyonlar	İş akışı işlevselliği
Arşivciler	Pasif koruyucular	Kollektif hafızayı aktif olarak şekillendirici

Klasnic (2019, s. 11), arşiv biliminin yalnızca arşiv belgelerine yönelik araştırmalarla sınırlı olmadığını, ancak bu belgelerin içeriklerinin de disiplinin doğrudan inceleme alanı içerisinde yer almadığını ifade etmektedir.

Jacob von Rammingen, 1540 yılında Augsburg şehri belgelerinin düzenlenmesi için bir el yazması hazırlamış ve böylece bilinen en eski arşiv el kitabı ortaya çıkmıştır. Bu el kitabını daha da önemli kılan şey ise, arşiv bilimi ile ilgili olarak "yayımlanan" ilk çalışma olmasıdır çünkü Rammingen'in bu el kitabını hazırlarken, belgelerin kaydedilmesi ile ilgili daha önceki literatüre atıfta bulunduğu bilinmektedir (Weidling, 2013, s. 270).

1632'de ise İtalyan avukat Bishop Baldassare Bonifacio, Venedik'te "De Archivis Liber Singularis"i yayınlamıştır. Bir broşür şeklinde hazırlanan bu kitapçık, Jacob von Rammingen'den sonra arşivler için hazırlanan ikinci bir ayrı yayın olarak görülmektedir. Kitapçık arşivlerin tarihi, önemi ve yönetimi hakkında kısa bilgiler içermektedir (Born, 1941).

Hollandalı arşivciler Muller, Feith ve Fruin (2003/1898) tarafından 1898 yılında yazılan "Manual for the Arrangement and Description of Archives[23]" kitabı ile, arşiv biliminin modern olarak bir bilimsel disiplin haline dönüşmesinin temeli atılmıştır. Hollandalı arşivciler, o zamanlar hâlihazırda uygulanmakta olan kuralları belirli bir bilimsel sistematik yapı içerisinde açıklayarak arşiv biliminin kurucuları olarak kabul edilmişlerdir. Arşiv bilimi, kısmen entelektüel kısmen de uygulamalı bir bilimdir. Arşivciler, arşivsel süreçlerdeki faaliyetlerini arşiv teorisine dayandırır ve bu çerçeve bağlamında uygulama yapar. Arşiv materyalleri, bunları üreten veya oluşturan insanların ve kurumların tüm karakteristik özelliklerini içeren "temsilciler" olduğundan, arşivciler teorik bazlı uygulamaları pratik gerçeklikleri göz önünde

[23] Arşivlerin Düzenlenmesi ve Tanımlanması Kılavuzu

bulundurarak süreçlere uyarlamalıdır. Bu nedenle arşiv biliminin; soyut ve somut ile teorik ve pratik bir yapısal karışım olduğu söylenebilir.

2.3.1. Arşiv Teorisi ve Teorinin Dönüşümü

Arşiv bilimi; önceki bölümde de belirtildiği üzere arşiv belgelerini yönetme, koruma ve erişilebilir hale getirme tekniklerini içeren akademik ve uygulamalı bir disiplindir. Arşiv teorisi ise bu disiplin içerisindeki uygulamaların nasıl yapılması gerektiğine dair bir kavramsal çerçeve sağlayan yapıdır. Başka bir deyişle teorinin temel rolü; uygulamayı ve metodolojileri yönlendiren ilkeler oluşturmaktır ve bu ilkeler arşivcilere metodolojik temeller sağlar. Örneğin; kâğıt tabanlı belgelerden elektronik belgelere geçiş; dayanıklılık, erişim ve metaveri yönetimi gibi unsurların yeniden değerlendirilmesine yol açmıştır (Manoff, 2004; Yalçınkaya, 2017; Çiçek Akgün ve Çiçek, 2022; Casenave ve Kane, 2024).

Duranti'ye göre (1994, s. 330) arşiv teorisi; arşiv malzemesinin ne olduğu ile ilgili fikirlerin bütünü olup, bu malzemelerin nasıl yönetilmesi gerektiğine dair metodolojileri içerir. MacNeil (1994, s. 7), arşiv teorisini bir model bağlamında tanımlamış ve bu modelde teori; arşivlerin doğası hakkındaki fikirlerin analiziyken; metodoloji, arşivlerin nasıl ele alınacağına dair fikirlerin analizidir. Uygulama ise metodolojinin belirli durumlara yansımasının bir sonucudur. Ancak uygulama, çoğunlukla kurumsal bir bağlama özgüdür. Binark ise (1980, s. 6) arşiv teorisini; arşivin uygulaması, kuruluşu, organizasyonu, hizmetlerin planlanması, belgelerin arşive girişi, sınıflanması, katalog ve envanterlerinin hazırlanması ile kullanıma sunulmasına dair işlemler olarak tanımlamıştır. Eastwood (1994, s. 125), arşiv teorisinin ilk amacının arşiv belgelerinin doğasını anlamak olduğunu vurgulamaktadır. Bilimsel bir arşiv disiplini, arşiv belgelerinin doğası hakkında bilgi ve belgelerin sahip oldukları özellikleri korumak için sistematik yollarla hareket etmekten oluşmaktadır. Bu nedenle, arşiv teorisinin temel sorusu; belgelerin korunması gereken özelliklerin neler olduğu ve neden korunması gerektiğidir.

Bu bağlamda teorinin, hangi malzemelerin arşiv malzemesi sayılabileceğine dair bir çıkarım temelinde oluşturulduğu söylenebilir. Teori, kesin yargılar içermemekle birlikte, esnek bir çatı model önermektedir. Teorinin kapsamını anlayabilmek için, arşiv materyallerinin doğası hakkındaki fikirlerin analiz edilerek, kendi içlerinde tutarlı olup olmadıklarını tespit etmek gereklidir. Eastwood (1994, s. 127-128), teoriyi analiz ederek, dört temel fikri şu şekilde açıklamaktadır:

1. ***Tarafsızlık:*** Arşiv belgeleri bir eylemi ifade etmenin aracı ve bu eylemin bir ürünü olarak üretildiklerinden dolayı, önyargı şüphesinden uzaktır. Fakat bu durum, üreticilerinin

önyargıdan uzak olduğu anlamına gelmez. Sadece üretilmelerinin nedenleri ve koşulları, sonraki nesillerin çıkarları veya bilgileri için yazılmadığını gösterir. Belge bu anlamda tarafsızsa onun gerçeklere ve eylemlerine olan sadakatine güvenilebilir. Arşiv belgeleri, gerçeğe ve eyleme sadık olma iddiasıyla oluşturulduklarından dolayı kimi zaman bazı kişi ya da kurumların çıkarlarına zarar verebilecek gerçekleri açığa çıkarabilme potansiyeli taşır. Belgelerin bozulmadan kalması ve tarafsızlığının mümkün olduğunca korunması için yöntem ve uygulamalar tasarlaması gerekenler ise arşivcilerdir.

2. ***Özgünlük:*** Bir belgenin özgünlüğü; belgenin oluşturulma ve koruma olgularına bağlıdır. Arşivler, yalnızca bu özellikler aracılığıyla oluşturulduklarında ve üreticisi ile yasal ardılları tarafından düzenlendikçe ve onu saklama gücüne sahip olan kurumlar tarafından saklandıkça özgün kalabilir. Başka bir deyişle belgeler; gerçek ve eylemin sadık tanığı olarak korunup saklandıklarında özgün olabilir. Belgelerin, geçmiş faaliyetlerin özgün kanıtları olabilmesi için onaylanmış düzenli prosedürlere göre de oluşturulması ve saklanması gerekmektedir. Doğal olarak, belgeye özgünlük kazandıran bu koşullar, belgenin kendisinde değil, prosedürlerde gözlemlenebilir. Bu nedenle arşiv teorisinin kapsamı; belgelerin ötesine geçerek, oluşturulması ve saklanmasına dair prosedürel bağlamının dikkate alınmasını kapsar. Başka bir deyişle; sadece tarihi arşivin yöntem ve uygulamalarına değil, aynı zamanda belgelerin oluşturulduğu andan itibaren ve varoluşları boyunca üretildiği ve saklandığı yöntem ve uygulamalarıyla da ilgilenir. Pek çok arşiv belgesi, idari ve hukuki denetim süreçlerinin dışında kaldığında, delil niteliğini kısmen ya da tamamen yitirebilir. Her ne kadar bu belgeler geçmişe dair sembolik veya kültürel değer taşımaya devam etse de, delil olarak güvenilirlikleri zayıflamıştır. Belgenin üretim bağlamına ilişkin bilgiler yeniden inşa edilmeye veya içerik çözümlemeleriyle eksikler giderilmeye çalışılsa da; bu tür belgeler, sürekli kurumsal saklama altında korunan arşiv belgeleriyle karşılaştırıldığında, hukuki ya da bilimsel açıdan daha az güvenilir kabul edilmektedir.
3. ***Doğallık ve karşılıklı ilişki:*** Arşivlerdeki belgelerin durumu, mevcut kurumsal veya bireysel ihtiyaçlara göre işlerin yürütülmesi sırasında birikme biçimiyle ilgilidir. Her ne kadar bu birikim, farkında olunmadan ait olunan hukuki sistem, ilgili ülkede geçerli olan siyasi rejim ve kültür gibi unsurlar neticesinde şekillense de belgeler yapılan işte ihtiyaç duyulan bilgi açığını gidermek için aralarında organik bağ kurularak bir araya getirilir. Bu bağ, belgenin ait olduğu faaliyet ve onun bir üst sekmesi olan fonksiyon bağlamında değerlendirilir. Dolaysıyla belgeler onları oluşturan süreçlerin kanıtlarıdır, çünkü onları oluşturan idari ihtiyaçların dışında bir amaç için toplanmaz ve bir müzedeki materyaller

veya bir kütüphane koleksiyonundaki belgeler gibi, bu ihtiyaçlar dışında bir şeye hizmet etmek üzere bir şemaya göre bir araya getirilmez. Herhangi bir arşivdeki belgelerin ilişkileri, işlerin yürütülmesi ve ihtiyaçlarına göre kurulur. Belgeler arasındaki ilişkiler ve belgelerin üretildiği işlemlerle olan bağları, tek bir arşiv belgesinin geçmiş faaliyetleri tam anlamıyla temsil edemeyeceğini ortaya koyar. Her bir belge, anlamını diğer belgelerle birlikte kazanır ve ancak bu bağlamda ilgili faaliyetin kanıtı olarak işlev görebilir. Bu nedenle arşiv teorisi, belgeler ile onları üreten idari süreçler arasındaki yapısal ve işlevsel ilişkiyi temel alır. Hem belgelerin ortaya çıkmasına neden olan işlevin hem de yapıların anlaşılması; yöntem ve uygulamanın geliştirilmesi açısından hayati önem taşımaktadır. Belgeler arasındaki doğallık ve karşılıklı ilişkilerin dinamiği anlaşılmadan yapılan işlemler, işlevsel ve yapısal bağları zedeleyebilir; bu da belgenin anlamını, önemini ve kanıt değerini taşıyan bütünlüğünü bozma riski taşır. Bu karakteristiği doğru biçimde algılayabilmek için teorik bir çerçeve gerekir; onu koruyabilmek için ise bu teoriye dayalı bir yöntemle hareket edilmelidir.

4. ***Tekillik:*** Her arşiv belgesi, oluşturulduğu bağlam, içerdiği bilgiler ve arşivsel düzen içindeki yeri açısından kendine özgüdür. Aynı bilginin farklı arşivlerde ya da belgelerde bulunması mümkündür; ancak her belgenin üretim süreci, ilişkisel bağlamı ve saklandığı fon içindeki yeri, onu özgün ve ayırt edici kılar. Bu bağlamda, bir belgenin nüshaları bulunsa dahi, her biri kendi arşivsel bağlamında tekil olarak kabul edilir. Ancak günümüzde, belgelerde yer alan bilgilerin büyük bir kısmına arşiv dışındaki ortamlardan, daha erişilebilir ve işlevsel formatlarda ulaşmak da mümkündür. Bu durum, arşiv belgelerinin değerini yalnızca taşıdığı bilgiye indirgememek gerektiğini; bağlamı, düzeni ve üretim sürecine dayalı özelliklerinin de göz önünde bulundurulması gerektiğini ortaya koymaktadır.

Avrupa ve Kuzey Amerika geleneğindeki arşivciler tarafından farklı uygulamaları gerçekleştirilen teori, 20. yüzyılın büyük bir bölümünde modern arşivlerin ihtiyaçlarını etkili bir biçimde karşılamıştır. Ancak, özellikle BİT temelli gelişmeler (ağ, internet ve bilgisayarlaşmanın hızlanması vb.) teorinin temel varsayımlarına meydan okumaya başladıkça arşivsel işlemlerin (değerlendirme, tanımlama, erişim, koruma vb.) temel kavramlarında etkili bir sorgulama başlatmıştır. Bu sorgulama, temel kavramların (provenans, orijinal düzen, fonlara saygı vb.) değiştirilmesi ve geliştirilmesine yönelik birçok çalışma ve projeyi içermektedir (Cook, 2011b, s. 605). Gerek provenans ilkesi gerekse orijinal düzen ve fonlara saygı ilkeleri; fiziksel belgelerin tanımlanması, düzenlenmesi ve erişimine yönelik olması

nedeniyle, dijital ortamda oluşturulan veri, bilgi ve belgelerin yönetimsel süreçlerine uymamakta ve uygulamada fazlasıyla boşluklara yol açmaktadır.

Hilary Jenkinson'ın 1922'de yayımlanan ve 1937'de gözden geçirilen "A Manual of Archive Administration[24]" adlı eseri, arşiv teorisi üzerine İngilizce yayımlanan ve yaygın olarak okunan öncü çalışmalardan biridir. Kitapta, teorik yapıların çoğu Muller, Feith ve Fruin'in (2003/1898) "Manual for the Arrangement and Description of Archives[25]" adlı kitabından alınsa da, Jenkinson daha fazla açıklama, sınırların çizilmesi ve önerilerle Hollandalı arşivcilerin temel pratik varsayımlarını geliştirmiştir. Jenkinson'un arşiv teorisinin en kapsamlı ve belki de benzersiz yönü, belgelerin ve arşivlerin tanımıdır. Jenkinson'un terimleriyle arşivin ve belgelerin tanımı, İngiliz arşivciliğini düşük seviyeli bürokratik işlerden; büyük miktarda belgeyi, zamana duyarlı bir şekilde arşivleme kapasitesine sahip, tamamen işlevsel ve entelektüel açıdan titiz bir çabaya dönüştürecek bir katalizör haline gelmiştir (Ridener, 2009, s. 41). Jenkinson, bir arşivcinin tipik olarak pasif bir koruyucu[26] görevi yapmasını ve her materyalin saklanması gerektiğini savunmaktadır. Ridener'e göre (2009) Jenkinson'ın teorisindeki ana unsurlar; arşiv belgesinin nesnelliği, provenans ilkesi, fonlara saygı ilkesi ve belgelerin birbirine bağlılığının önemi, arşiv belgelerinin üretilmesinde, korunmasında ve diğer belgelerle ilişkilerinde organik yapısı, arşivin önemini korumak için sürekli saklama ve arşiv belgelerinin kontrolünü içermektedir.

Eastwood (2004, s. 40), post-modernist düşünceden ilham alan arşivcilerin, neyin saklanıp neyin imha edileceğine karar verdiklerinde, arşivlerin ne ifade ettiği ve bir arşivin aslında ne içerdiğine dair güçlü bir etken olmaya başladıklarını vurgulamaktadır. Böyle bir durum, Jenkinson mantığına aykırıdır. Jenkinson'a göre belgeler, kurum veya kuruluşun çıkarları düşünülerek düzenlenir, gelecek nesillerin çıkarları bu durumda düşünülmez.

Jenkinson'a göre arşivcinin asli görevi; belgelerin üretildikleri bağlamdaki bütünlük**,** özgünlük ve tarafsızlık niteliklerini muhafaza etmek ve bu belgelerin arşivsel değerlerini müdahalesiz ve nesnel bir biçimde koruma altına almaktır. Arşivci, bu yönüyle belgelerin yalnızca fiziksel değil, aynı zamanda ahlaki bütünlüklerinin de sorumluluğunu üstlenir. Jenkinson; değerlendirme aşamasının, önemsiz belgelerin ayıklanması ve imha edilmesi de dahil, arşive transfer edilmeden önce gerçekleştirilmesi gerektiğini ve bu görevin arşivcilerin değil, belgeyi üretenlerin sorumluluğunda yapılması gerektiğini savunmuştur. Holmes (1961, s. 346) ve

[24] Arşiv Yönetimi El Kitabı
[25] Arşivlerin Düzenlenmesi ve Tanımlanması Kılavuzu
[26] Custody

Hohmann (2016, s. 16); Jenkinson'ın ayrıca arşivcinin içeriğin bir analisti olmadığını, sadece belgeleri incelemek isteyenler için bir kanıtın koruyucusu olduğu fikrini savunduğunu belirtmiştir. Jenkinson'ın temel amacı, arşiv belgelerinin niteliğini ve özelliklerini araştırmak; mevcut ve geleceğin arşivlerinin oluşturulmasına rehberlik edebilecek arşiv ilkelerinin anlaşılmasını sağlamaktır. Arşiv tanımında Jenkinson, saklama tarihlerini, organik yapılarını ve doğal süreçlerle birikimlerini vurgulamıştır: *"Arşiv sınıfına ait olduğu söylenebilecek bir belge, kendisinin bir parçası olduğu idari ya da yürütme işlemi sırasında (kamuya açık ya da özel) hazırlanmış ya da kullanılmış olan ve daha sonra bu işlemden sorumlu kişi ya da meşru halefler tarafından kendi bilgileri için kendi gözetiminde muhafaza edilir"* (Jenkinson, 1937, s. 11). Jenkinson, arşivciler tarafından tarihi veya başka nedenlerle uzun vadeli koruma için belgelerin seçilmesinin, arşivlerin bütünlüğünü ve özellikle tarafsızlıklarını veya oluşturuldukları konular hakkında kanıt sağlama niteliklerini zedeleyeceğini düşünmektedir (Eastwood, 2004, s. 40). Jenkinson, arşiv belgelerinin üretilme şeklinin esas alınarak bir değerlendirme yapılmasını da savunmuştur. Ona göre düzenli faaliyetler sırasındaki doğal birikim; belgelerin saklanmak için seçilmesinde bir kriter değildir. Aksine, oluşturucuları tarafından üretilmesi olgusu dikkate alınarak, belge üreticilerinin kendi özel ve gelecekteki kullanımları, arşivlere tarafsızlık ve özgünlük nitelikleri kazandırmaktadır. Bu nitelikler; arşivlere kendi özel değerlerini geçmişin kanıtı olarak vermesini sağlamıştır. Jenkinson ayrıca arşivlerin birbiriyle ilişkili belgelerden oluştuğunu ve bunun anlam ifade eden ve korunması gereken bağlamsal bütün olduğunu fark etmiştir.

1950'lerde, Jenkinson'a karşıt bir düşüncenin öncülüğünü ise Schellenberg yapmıştır. Tschan'a göre (2002, s. 177), Schellenberg idealist fakat daha pragmatik bir müdahaleci ve imha programının kurucusu olarak görülmektedir. Schellenberg'in özellikle ABD'deki arşiv teorisine katkısı; arşiv uygulamalarında değerlendirmenin merkeziyetini vurgulamak, seçim sürecini arşivcinin birincil rolü yapmak ve oldukça fazla hacime sahip belgelerin imhasını gerçekleştirmektir. "Modern Archives: Principles and Techniques[27]" adlı kitabında, Schellenberg'in (1956) arşiv anlayışı birçok bakımdan geleneksel teoriden beslenmiştir. Schellenberg; arşivlerin, belgelerin organik yapısının korunmasından sorumlu olduğunu savunmuş ve fonlara saygı ilkesinin merkeziyetçiliğini desteklemiştir. Schellenberg ayrıca, belirli amaçlara ulaşmak için faaliyetlerin yapıldığı sırada arşivlerin doğal olarak oluşturulduğunu ve arşiv belgelerinin; herhangi bir müdahale, değişiklik veya yetkisiz imha olmadan belirli bir bütünlükte tutulması gerektiğini vurgulamıştır. Schellenberg, devlet

[27] Modern Arşivler: İlkeler ve Teknikler

kurumları tarafından oluşturulan çok büyük hacimdeki belgelerin, daha efektif kullanılabilmesi adına azaltılması gerektiğinin de farkındaydı. Smith'e göre (1981, s. 316), Schellenberg belgelerin etkin bir şekilde azaltılması için prosedürlerin geliştirilmesinde öncü bir rol oynamış ve Ulusal Arşivler'de ilk imha programlarının hazırlanmasına büyük katkı sunmuştur. Schellenberg; belgelerin, üreticileri tarafından belirli bir ihtiyacı karşılamaya yönelik oluşturulmalarına rağmen, kalıcı olarak saklanmak üzere seçilmelerinin altında yatan nedenin "ihtiyaç" olmadığına inanmaktaydı. Yine Tschan'a göre (2002), Schellenberg, arşivlerin o sırada uğraşmakta olduğu büyük belge hacimleri ile başa çıkmanın en etkili yolunun, bir belgenin üreticisine hem kanıt değeri hem de fayda değeri sunan birincil değer ve üreticisi dışındakiler için gerekli olacak tarihsel ve kültürel işlevleriyle ilgili ikincil değer arasında ayrım yapmak olduğunu ve böylece, belgelerin ikincil değerine dayanarak nispi değerini belirlediğini savunmuştur. Schellenberg ayrıca ikincil değeri; kuruluşların fonksiyonlarını yerine getirme şeklinin kanıtı olan delil değeri ve belirli kuruluşlar tarafından ele alınan çeşitli konularla ilgili bilgi değeri olmak üzere iki alt boyuta indirgemiştir. Bu farklılaşma, belgeler ve arşivler arasında daha büyük bir ayrıma neden olacağı için Schellenberg tarafından önemsenmekteydi. Ona göre, belgelerin üreticileri için geçerli olarak sadece birincil değeri vardı; ancak arşivler, belgenin üreticileri için geçerli olan orijinal değerlerinin dışında, bir arşivci tarafından önemli olarak görülen ikincil değere sahip belgelerden de oluşmaktaydı. Sonuç olarak, belgelerin, belge yöneticisinin gözetimi altında olduğu ve sadece gelecekteki kullanıcılar için (özellikle gelecekteki tarihsel araştırmalar için) değeri olan belgeler, arşivlerin bir parçası haline gelecekti (National Archives, 2007). Schellenberg; modern belgeleri eski arşivlerden ayıran ve yeni arşiv teorisinin ilkelerini ortaya çıkaran en önemli nedenin, belgelerin hacmi olduğunu belirtmiştir. Schellenberg, ABD'de giderek artan miktarda biriken dağınık belge yığınları karşısında organizasyonel verimliliği optimum seviyeye çıkarmak adına, 1934'te Birleşik Devletler Ulusal Arşivi'nin kuruluşunu oldukça önemli bir gelişme olarak değerlendirmektedir (Schellenberg, 1956). Schellenberg'in arşiv teorisine katkısı; kalıcı olarak değerli olan belge kütleleri arasından seçim yapılarak, bu seçimlerin araştırmacılar için akıllıca kullanılabilmesi önerisi olmuştur. Schellenberg, arşiv belgelerinin seçim sürecini, arşivcinin rolünün merkezine oturtmuş ve arşiv tanımını şu şekilde merkezileştirmiştir: *"Herhangi bir kamu veya özel kuruluşun, referans ve araştırma amacıyla sürekli olarak saklanmaya hak kazanmış ve bir arşiv kurumuna devredilmek üzere seçilen ve gönderilen veya oluşturulan belgeler"* (Schellenberg, 1956, s. 16).

Ernst Posner ise, Avrupa arşiv geleneğinin ABD'ye getirilmesinde kritik bir rol oynamıştır. O'Toole (2004), Posner'ın antik çağ arşivlerinde, üzerinde çalıştığı çok farklı medeniyetler arasında bağlayıcı doku görevi gören bir dizi tutarlı tema bulduğunu ve onu en çok etkileyen durumun; belgelerin ve arşiv uygulamalarının uzun süreli devamlılığı olduğunu belirtmektedir. Posner'a göre; günümüzde aşina olunan belgelerin temel özellikleri, antik çağlardan beri saklanan belgeler ile neredeyse aynı özellikleri göstermektedir. Başka bir deyişle, Posner'ın başlangıç noktasının, antik çağ arşivlerinin kendi zamanımızdaki arşivlerle birçok ortak noktası olduğu rahatlıkla söylenebilir. Örneğin Posner; bu arşivleri oluşturan kültür veya kurumdan bağımsız olarak "belge üretiminde sabit olarak adlandırılabilecek türleri" tanımlamanın mümkün olduğunu belirtmektedir: Bu türler; yasalar, idari faaliyet belgeleri, finansal ve muhasebe belgeleri, arazi mülkiyeti ve vergi belgeleri, askerlik ve işgücü gibi alanlarda "insanlar üzerinde kontrolü" kolaylaştıran belgeler ile devletin ve bireylerin özel işlemlerini onayladığı ve koruduğu "noter" belgeleridir (Posner, 1972). Bunların yanında, her türden mali belgenin de sayıca en fazla olduğu ve etkilerinin uzun süre boyunca devam ettiği vurgulanmıştır. Makleff'e göre (2017) Posner; günümüzde güncel belgeler (dosyalar) ile güncel olmayan belgeler (arşivler) arasındaki ayrımın antik dünyada olduğu kadar keskin olmadığı ve bu nedenle antik çağlardaki arşiv süreçlerini nispeten düzensiz ve hatta kısmen tartışmalı olarak tanımlamaktadır. Böyle bir durum değerlendirmesi oldukça normal ve olağan karşılanmalıdır. Çünkü antik çağlarda, tıpkı günümüzdeki gibi güncel belgeleri içeren belge yönetimi disiplini ile güncel olmayan belgeleri içeren arşiv yönetimi disiplini "bilimsel" olarak var olmamıştır. Bunun nedeni ise o zamanki iş ve süreçlerin tamamen pratik çıkarlar doğrultusunda düşünülmesidir. Posner (1955, s. 210) ayrıca, belge yönetim süreçleri ile arşiv yönetimi arasındaki ilişkinin benzerliğinden yola çıkarak, ABD kamu kurum ve kuruluşlarının belge yönetimi ve arşiv yönetimini tek bir sorumluluk noktasında birleştirilebileceğini belirtmekte ve belge yöneticisi ile arşivcinin bir Amerikan olgusu olarak görülmesi gerektiğini vurgulamaktadır. Aynı şekilde Posner, antik çağdaki belgelerin yönetiminin çoğu yönünün temelde şimdiki olanlarla aynı olduğunu öne sürmektedir: *"Halihazırda bildiğimiz tüm arşiv organizasyonu ve uygulamalarının, basit depolama tesislerinden kayda değer büyüklükteki arşiv tesislerine kadar antik çağdaki Yakın Doğu'da uygulanmakta"* olduğunu vurgulayan Posner, O'Toole'ye göre (2004, s. 165) belgelere erişim konusunda, arşivcilerin materyalleri düzenli bir şekilde arşivde saklayabilmek adına yöneticilerle yakın bir şekilde çalışmak zorunda olduklarını da belirtmiştir.

Duranti (1994); arşiv teorisinin temel kavramlarının Roma İmparatorluğu hukukuna gömülü olan kavramlara dayandığını öne sürmekte ve bu kavramlardan "kalıcı bellek" ve "kamuoyu inancı"nın, çağımızda değerlendirme ile ilişkilendirilen fikirlere doğrudan etkisi olduğunu belirtmektedir. Kalıcı bellek kavramı ilk olarak arşiv belgelerine ve daha sonra arşivlere bağlandığında, sürekli bir koruma yükümlülüğü taşımayıp, yalnızca güvenilirliğin bir ifadesini göstermekteydi. Hukuki anlamda kamuoyu inancı kavramı ise, arşiv teorisinde kalıcı belleğe göre daha fazla önem taşımaktadır. Kalıcı bellek fikrinin arşiv belgeleri ile doğru ifade ettiği gerçekler arasındaki ilişkiyi ve kamuoyu inancının ise arşivler ile hizmet edilen toplum arasındaki ilişkiyi ifade ettiği söylenebilir. Eski ve Ortaçağ Batı dünyasında, arşiv belgelerinin bir arşivde saklanmasıyla güvenilirliği sağlanmıştır ve bu bağlamda arşiv teriminin, arşiv belgelerinin üreticileri tarafından tutulduğu yeri ifade ettiğini unutmamak gerekmektedir.

2.3.1.1. Provenans İlkesi

Bu ilke, arşiv teorisinin en önemli ilkelerinden biridir. Temel arşiv işlemlerinin organizasyonu, 19. yüzyılın ikinci yarısından itibaren bu ilke bağlamında yürütülmüştür. Arşivlerde, belgelerin sınıflandırılması ve tanımlanması, provenans ilkesinin uygulanmasına dayanmaktadır. Arşivlerden elde edilebilecek her türlü içgörü, bu ilke sayesinde anlam kazanmaktadır. Bu ilkenin önemi, belgelerin orijinal düzeninin korunmasında da ortaya çıkmaktadır. Özellikle Fransız Devrimi ve Aydınlanma Çağı'ndan sonra arşivlerin halka açılmasıyla birlikte, belgeler konusal ayrıma gidilerek kategorize edilmiş ve bu şekilde düzenlenmiştir. Bunun sonucunda ise çeşitli kurumlardan gelen belgeler tek bir grup olarak ele alınmış ve orijinal düzen göz ardı edilmiştir. Fransız Ulusal Arşivleri'nin ilk yıllarında benimsediği konusal kategorizasyon, arşiv sistemlerinin tamamen parçalanması gibi ciddi sorunları beraberinde getirmiş, bu sorun daha sonra provenans ilkesinin formüle edilmesiyle çözülmüştür (Guimarães ve Tognoli, 2015, s. 565).

Brenneke (1968, s. 75), ilkenin teorik kökeninin Ernst Spiess tarafından kaleme alınan 1777 tarihli "Von Archiven" çalışmasına dayandığını; buna göre belgelerin nasıl düzenlenmesi ve ne tür bir düzene sahip olması gerektiğini bilebilmek için belgelerin hikâyesinin bilinmesi gerektiğini; ayrıca ilkenin, gelecekte her fonun içeriğini ve organizasyonun yönteminin belirleneceği "tümevarımsal" bir yöntem olduğunu aktarmaktadır. İlkenin özü; üreticisi belirli olan ve belirli bir fonksiyon veya görev kapsamında üretilen arşiv malzemelerinin, aynı üreticiye ait olup başka fonksiyonlar çerçevesinde oluşan arşiv malzemeleri ile karıştırılmaması gerektiğidir. Arşiv materyalleri; üretildikleri kişi ve kurumların hiyerarşik yapıları doğrultusunda, üretim bağlamları bozulmadan ve birbirinden ayrılmadan

saklanmalıdır. Bu yaklaşım, belgenin fonksiyonel bütünlüğünü korumayı amaçlar. (Yıldız, 2006, s. 7). İlke; arşivsel analiz veya işlemlerin belge bazından ziyade belge setlerine göre yapılmasına olanak tanımakta ve belgelerin, kurumsal iş ve işlemlerin bir sonucu olarak oldukça doğal bir şekilde ilişkili olduğu organik bir yapıya bağlı olduğunu göstermektedir.

Arşiv ve Belge Terminolojisi Sözlüğü'nde (Pearce-Moses, 2005, s. 317), provenans; *"bir şeyin kökeni veya kaynağı; bir öğenin veya koleksiyonun kökeni, muhafazası ve sahipliğine ilişkin bilgi"* şeklinde tanımlanmaktadır. İlke, bir koleksiyondaki öğelerin üreticisine veya bunları alan kişi, aile veya kuruluşa atıfta bulunur ve farklı provenanslara ait belgelerin bağlamlarının korunabilmesi için ayrı tutulmasını gerektirir. Burada amaç; sahiplik veya geçmişi belgelemek ve korumak ile üreticinin geçmişine odaklanmaktır. Arşivciler, tanımlama aşamasının bir parçası olarak belgelerin provenansını kaydeder ve araştırmacıların her belgenin ve koleksiyonun sahipliğini veya oluşturulma geçmişini izleyebilmelerini sağlar. Arşivcilik Terimleri Sözlüğü'ne göre ise provenans ilkesi; *"arşiv belgelerini, işlem gördüğü tarihteki oluşum ve ilişkisel biçimlerine uyarak, işlevsel ve kurumsal faaliyetleri yansıtacak biçimde düzenleme yöntemi"* olarak tanımlanmaktadır (Karakaş, Rukancı ve Anameriç, 2009, s. 35).

Schellenberg (1965, s. 90) ilkeyi, Amerikan arşivcilik mesleğine bir Avrupa katkısı olarak görmüş ve ilkenin benzer türdeki kurumlardan gelen belgelerin fonlar içinde gruplandırılmasını öngören Fransız "fonlara saygı" ilkesinden geldiğini vurgulamıştır. Ona göre ilke; Alman arşivciler tarafından modern provenans ilkesinin eşdeğeri olarak kabul edilen Provenienzprinzip[28] formüle edilerek, arşivsel süreçlerin uygulanmasında daha da kesin ve tanımlayıcı bir hale getirilmiştir. Alman arşivciler ise ilkeyi; bir fonun belirli idari üreticisiyle ilişkisinin öneminin vurgulandığı Provenienzprinzip ve belirli bir fon içindeki belgelerin orijinal düzeninin korunduğu Registraturprinzip[29] olmak üzere iki şekilde formüle etmişlerdir. Hollandalı arşivciler, 1898'de Hollanda Arşivciler Derneği himayesinde "Arşivlerin Düzenlenmesi ve Tanımlanması Kılavuzu"nu yayımlayarak provenans ilkesinin sağlamlaştırılmasına önemli bir katkıda bulunmuşlardır. Kılavuz yalnızca; Fransızların fonlara saygı ve provenans ilkelerinin bu iki Alman kaynaklı bileşenine yaptığı vurguyu kabul etmekle kalmamış, aynı zamanda uygulamadaki sorunlardan kaynaklanan en önemli pratik sorulara da yanıt vermiştir. Bu temelden yola çıkılarak ilke; ilk defa 1910 yılında Brüksel'de düzenlenen Uluslararası Arşivciler ve Kütüphaneciler Kongresi'nde; arşivlerin düzenlemesinde,

[28] Provenans ilkesi

[29] Kayıt ilkesi. Özellikle Almanca bağlamda, arşiv bilimi ve belge yönetiminde öncelikli olarak kullanılan bir kavramdır. Bir arşiv sistemi içindeki belgelerin sistematik bir şekilde düzenlenmesi, depolanması ve alınması anlamına gelir. Bu ilke, tüm belgelerin yapılandırılmış bir şekilde tutulmasını ve genellikle orijinal organizasyon yapısını veya idari süreci yansıtmasını sağlar.

tanımlanmasında ve tarihsel araştırmalarda uygulanabilecek en iyi yöntem olarak oybirliğiyle onaylanmıştır (Duranti ve Franks, 2015b, s. 285).

Horsman (1994, s. 51) provenans ilkesinin, arşiv teorisinin hem dışsal (arşive; bir kişi, grup veya varlık tarafından bir bütün olarak üretildiği haliyle saygı göstermek-fonlara saygı) hem de içsel (belgelere üreticileri tarafından verilen düzene saygı göstermek-orijinal düzen) olarak iki boyutta hareket edebilen tek ilkesi olduğunu iddia etmektedir. Her iki boyut da ayrılmaz bir bütün olarak hareket ettiğinde, tam bir provenans kontrolü sağlanabilir.

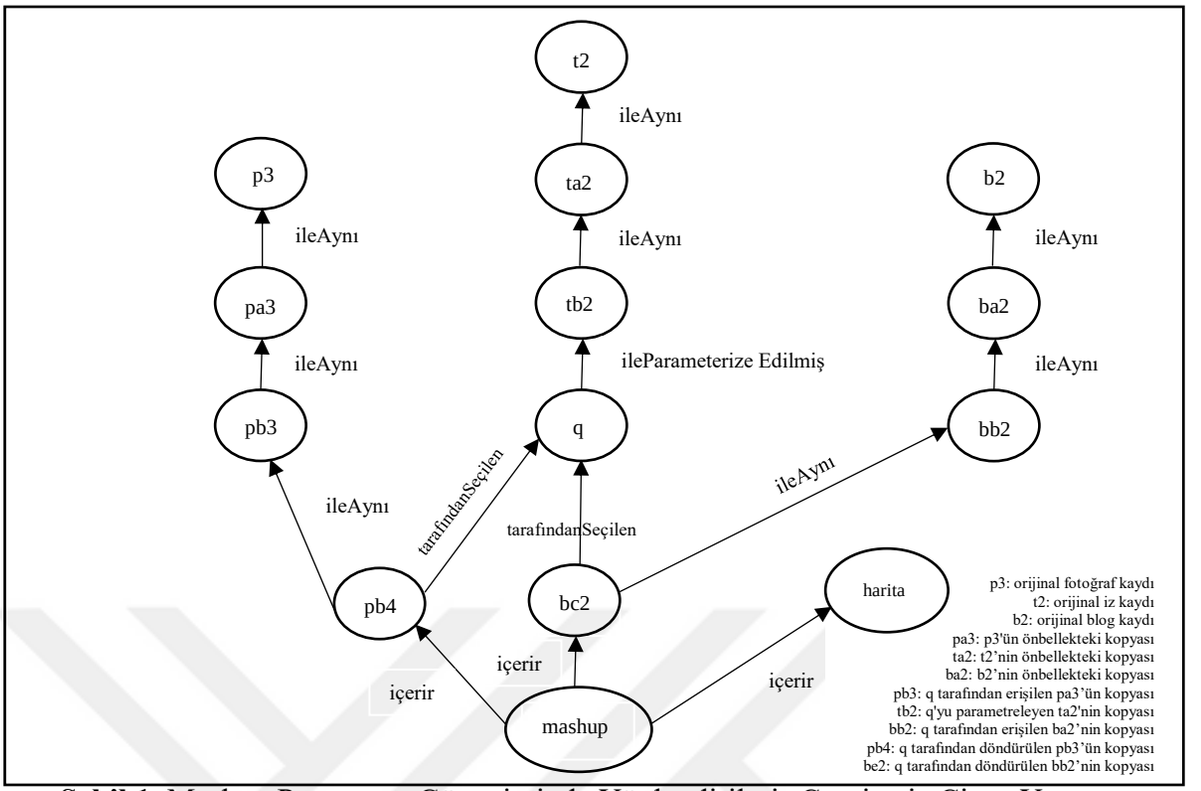
Dijitalleşmenin her geçen gün artması, daha çok bilgi ve belgenin elektronik ortamda temsil edilmesini sağlamıştır. Böylece, arşivsel süreçlere dahil edilen dijital bilgi ve belgenin -daha çok verileştirilmiş bir biçimde- düzenlenmesinde ilkenin de bir değişime -veya evrime- uğramasına yönelik birçok çalışma ve bu bağlamda öneriler mevcuttur (Nesmith, 2006; MacNeil, 2008; Meehan, 2009; Douglas, 2017; Barclay vd., 2020; Massari vd., 2023).

Stephano (2024); fon, provenans veya orijinal düzen kavramlarının arşivlerdeki paradigma değişimine artık tam anlamıyla bir karşılık veremediğini belirterek, Records in Contexts (RiC) kavramsal modeli bağlamında provenans ilkesinin rolünü incelemiş ve RiC kavramsal modelinin provenans anlayışının kavramın genişlemesine dair olmadığını iddia etmiştir. Ona göre fon, provenans ve orijinal düzen kavramları, tekillik ilkesine sahiptir ve yalnızca bir fon, tek bir provenans ve tek bir düzenin varlığını göstererek provenans ilkesini bir tekillik ilkesi haline getirir. Çalışmada, RiC kavramsal modelinin provenans ilkesine yönelik postmodern fikirler ve eleştiriler üzerine yapılandırıldığını ve bu yapılandırmanın fon, provenans ve orijinal düzen kavramlarıyla bağlantılı olmadığı da vurgulanmıştır. RiC kavramsal modelinde provenans ilkesi, diğer kavramların önem kazanması ve aralarındaki ilişkinin değişmesi nedeniyle azalmıştır. Provenansın, modeli içindeki 85 muhtemel ilişkiden yalnızca birkaçını temsil ettiği ve böylece modellenebilen çok sayıda olası ilişki ve bağlam içinde, baskın bir rol oynamadığını vurgulanmaktadır. Yazar, model içindeki bağlamsal ilişkilere atıf yaparak provenansın genişlemesi mantığının yerine, farklı bir şekilde çerçevelemeyi ve diğerleri arasında bir özelliğe indirgenmesini önermektedir. Başka bir deyişle provenans, tekil bir ana bağlam olmak yerine birçok bağlam arasında sadece bir bağlam veya bağlam öğesi haline dönüşebilecektir. Yazar, arşivsel süreçlerde merkezi referans ve yapılandırma öğesi rolüne sahip fon kavramının da önemini kaybettiğini; modelde daha soyut bir boyutta; bir niteliğin muhtemel değerlerinden biri olarak varlığını korumakta olduğunu belirtmektedir. Ayrıca, günümüzde fon kavramının artık evrensel geçerliliği olan bağlayıcı bir ilke olarak görülmediği de ifade edilerek, bu kavramın yerine fonlara karşılık gelen belge grupları, fiziksel bir bütünlük

içinde değil; bağlamsal ilişkiler temelinde tanımlanan ve her belge grubunu kapsayabilen yeni bir belge seti anlayışıyla ele alınmaktadır. Bu yeni yaklaşımda, belgelerin fiziksel olarak bir arada bulunması gerekliliği ortadan kalkmakta; belgeler aynı anda birden fazla bağlamla ilişkilendirilebilmekte ve eşzamanlı düzenlemelere olanak tanınmaktadır. Böylece, fiziksel orijinal düzen anlayışı, yerini çok boyutlu bağlamsal modellere bırakmaktadır. Yazarın bu fikri, provenansın dijital ortamda, klasik formata göre giderek "önemsizleştiğini" gösterse de, RiC modelindeki gibi "mantıksal ilişkilerden" sadece biri olmasına rağmen, provenansın aslında model içerisindeki diğer tüm özellik ve ilişkilerin ortak bir kavramı olabileceği anlamına gelebilir.

Provenans ilkesi, bilgisayar biliminde de dijital verilerin yaşam döngüsü boyunca geçmişini ve dönüşümünü tanımlayan metaverileri ifade etmektedir. Provenans; özellikle bilimsel iş akışları, veritabanları veya dağıtık sistemler gibi bağlamlarda verilerin türetilmesini izlemekle ilgilenir (Levine ve Liberatore, 2009; Palmer, Bubendorfer ve Welch, 2011). Örneğin, bir veri seti çeşitli algoritmalar aracılığıyla analiz edildiğinde, temizlendiğinde ve işlendiğinde provenans sistemi hangi işlemlerin kim tarafından ve ne zaman uygulandığını kaydeder; bu da veri-yoğun hesaplama platformlarında şeffaflık, yeniden üretilebilirlik ve hesap verebilirlik sunar.

Moreau (2010, s. 123) bilgisayar bilimi açısından provenansı *"bir veri parçasının provenansı, o veri parçasına yol açan süreç"* olarak tanımlamıştır. Bu tanımda amaçlanan kavramsal provenans; bir programın çalıştırılması, programa ait girdi verileri, programın yapılandırılma süreçleri, donanım olarak bilgisayar, bilgisayara güç veren elektrik ve son kullanıcılar gibi "sürece" dahil olan her bir boyuttaki bilgileri, potansiyel olarak "yakalama" kapasitesine sahiptir. Başka bir deyişle, burada amaçlanan; provenansı her boyutta yakalayabilecek bir bilgisayar tabanlı bir gösterim tasarlamaktır.



Şekil 1. Mashup Provenans Gösteriminde Yönlendirilmiş Çevrimsiz Çizge Yapısı
Kaynak: (Moreau, 2010, s. 127)

Moreau (2010), provenansı süreç bazlı olarak yorumlarken, Web 2.0 Mashup örneğinde provenansın, son kullanıcının ilgili içeriği görüntülenmesiyle sonuçlanan bir hesaplama süreci olduğunu belirtmektedir. Şekil 1'de, Mashup provenans yakalama süreci, Yönlendirilmiş Çevrimsiz Çizge[30] temelli bir yaklaşım ile gösterilmektedir. Burada her düğüm; hesaplama sürecinde bulunan veri öğelerini, kenarlar ise veri türevlerini ifade etmektedir. Örneğin; sorgu, q düğümü ile temsil edilirken; b2, bir blog içerisindeki veri gruplarını göstermektedir; ba2, b2'nin önbellekten alınan kopyasını ve bb2 ise sorgu motoru (q) tarafından erişilen diğer kopyasını temsil etmektedir. Şekilde Mashup, bir harita uygulaması aracılığı ile sorgu motoru tarafından üretilen sonucun birleştirilmesiyle oluşturulmuştur. Bu motor, son kullanıcının belirlediği bir sorguyu çalıştırmakta ve blogtaki önbelleklerden veri çekmektedir. Başka bir deyişle; bir hesaplama aracılığıyla Mashup'a bağlanabilen herhangi bir veri, olay veya kullanıcı eylemi potansiyel olarak Mashup'ın provenansına ait olduğu söylenebilir.

Arşivsel provenans, belgelerin sadece içeriklerine değil, fonksiyonlarına göre nasıl değerlendirileceğini, düzenleneceğini ve tanımlanacağını gösterirken; bilgisayar biliminde provenans daha çok süreç odaklı olup ve hesaplamalı iş akışları aracılığıyla verilere ne olduğuna dair bir anlayış sağlamaktadır. Her iki disiplin de bağlamsal bütünlüğe önem verse

[30] Directed Acyclic Graph - DAG

de odak, farklılık göstermektedir. Bilgisayar bilimi genellikle yazılım sistemlerine gömülü otomatik provenans yakalama mekanizmalarını kullanırken, arşiv biliminde provenans genellikle elle değerlendirme ve bağlamsal analiz yoluyla belirlenmektedir (Missier vd., 2013; Tennis, 2015).

Tablo 2. Arşiv Bilimi ve Bilgisayar Biliminde Provenans

	Arşiv Bilimi	**Bilgisayar Bilimi**
Tanım	Arşiv materyallerinin temeli, saklanması ve sahipliğinin belgesidir ve bu materyalleri kimin, hangi koşullar altında nasıl oluşturduğunu ve nasıl saklandığını gösterir	Bir veri nesnesinin, modelinin veya dijital materyalin üretilme geçmişini (yani nasıl oluşturulduğunu, dönüştürüldüğünü ve kullanıldığını) tanımlayan metaverilerdir
Odak	Saklama geçmişi, üretici, oluşturulan bağlam ve zaman içinde bütünlük	Verilerin ve hesaplama süreçlerinin izlenebilirliği (örneğin; girdi/çıktı dönüşümleri, kod bağımlılıkları, model durumları vb.)
Amaç	Materyallerin gerçekliğini, bütünlüğünü ve delil değerinin sağlanması ile yasal ve kurumsal sorumluluğun güçlendirilmesi	Bilimsel sonuçların yeniden üretilebilirliğinin sağlanması, iş akışlarının hata ayıklamasının, denetiminin, uyumluluğunun ve optimizasyonunun desteklenmesi
Örnek Durum	Resmi bir belgenin gerçek ve değiştirilmemiş olduğunun doğrulanması ve kurumsal arşivlerde gözetim zincirinin oluşturulması	Makine öğrenmesi uygulamasının yeniden üretilmesi ve model davranışının açıklanması (örneğin; öznitelik türetme yolları ve hiperparametreler)
Detay	Çoğunlukla fon, seri veya belge grubu düzeyinde tanımlanır; materyalin üreticileriyle olan ilişkileri ile arşiv hiyerarşisini gösterir	Makine öğrenmesi uygulamasındaki fonksiyonların, kütüphane versiyonlarının, hiperparametreler ve zaman damgası vb. düzeyinde izlenebilirliğini gösterir
Terim	Çoğunlukla bağlamsal metaveriler	Düğümler (varlıklar, aktiviteler, aracılar) ve kenarlar (tarafından üretildi, tarafından kullanıldı vb.) ile grafiksel yapılar (örneğin; W3C PROV)
Zamansallık	Uzun vadeli saklama ve tarihsel bağlamın vurgulanması	Operasyonel soyağacı, verilerin geçici hesaplamalar sırasında, çoğunlukla gerçek zamanlı veya iş akışına özgü bağlamlarda izlenmesi
Modeller	Fonlara saygı, orijinal düzen, ISAD(G), ICA-AtoM, DACS, RiC vb.	W3C PROV, PROV-ML, Açık Kaynak Modeli, RO-Crate, ReproZip vb.
Anlamsallık	Toplumsal, yasal ve kurumsal bağlama bağlılık	Verinin dönüştürülme mantığı ile yeniden üretilebilirliği, semantik ve ontolojik şemalar aracılığıyla gösterilir

Kaynak: (Horsman, 1994; Groth vd., 2012; Missier vd., 2013; Duranti ve Franks, 2015b; Guimarães ve Tognoli, 2015; Curcin, 2016; Michetti, 2016; Douglas, 2017; Cowan, 2018; Fuhrmeister Hopp, 2019; Barclay vd., 2020; Ciechanowski, 2021; Massari vd., 2023; Bak, 2024; Chapman vd., 2024)

Tablo 2'de provenans kavramının arşiv bilimi ile bilgisayar bilimi arasındaki farkları açıklanmaktadır. Tablodan da görüldüğü üzere arşiv bilimi; üreticiyi, oluşturma bağlamını ve belgelerin zaman içindeki sürekliliğini vurgular. Buradaki temel amaç, arşiv fonunun bütünlüğünü korumaktır. Öte yandan, bilgisayar bilimi ise girdi/çıktı ilişkileri, model

durumları ve hesaplama adımları gibi veri dönüşümlerinin izlenebilirliğine odaklanmaktadır. Metaveri yapısı açısından incelendiğinde ise arşiv bilimi provenansı genellikle belge grubu veya seri düzeyinde tanımlar ve hiyerarşi içindeki üreticiler ve belgeler arasındaki ilişkileri yansıtırken, bilgisayar bilimi; tekil veri öğelerini, zaman damgalarını ve sistem bileşenlerini vb. izleyen ayrıntılı provenans kullanmaktadır. Arşivsel açıdan düşünüldüğünde üreticinin adı, üretim tarihi ve serisi gibi bağlamsal metaveriler önemliyken, bilgisayar biliminde provenans, düğümleri (varlıklar, aktiviteler, aracılar vb.) ve kenarları (kullanılan, oluşturulan vb.) içerir ve genellikle grafiksel olarak görselleştirilir (örneğin, W3C PROV modeli). Zamansallık açısından da arşiv bilimi, provenansı uzun süreli koruma ve tarihsel süreklilik için kullanırken, bilgisayar biliminde çoğunlukla operasyonel log kayıtları ve iş akışlarının olay tabanlı izlenmesi de dahil olmak üzere gerçek zamanlı izlenme kullanılır (Groth vd., 2012). Her ne kadar iki kavramın da farklılıkları olsa da, bilgisayar biliminin provenansı, arşiv biliminde dijital olarak talep edilen yapıyı da sağlayabilmektedir. Dolaysıyla, arşivsel provenansın bilgisayar bilimi aracılığıyla da sağlanabileceği söylenebilir.

Bak (2024) ise dijital teknolojilere odaklanarak, arşiv teorisi ve pratiğinde, provenans ilkesine yönelik belge oluşturma ve saklama teknolojilerinin nasıl düşünülebileceği ve bunlara nasıl yanıt verebileceği konusunda bir çözümleme sunmaktadır. Dijital provenansın yeni bir provenans türü ve kavramın yeniden icat edilmesi olmadığını vurgulayarak, dijital belge yönetim sistemine ve teknolojisine uygulanması olduğunu belirtmiştir. Dijital provenansın anlaşılması; arşivcilerin dijital belgeleri işleyebilmesi için gereklidir; ancak daha da önemli olan; arşivcilerin ve kullanıcıların dijital belgelerin bağlamını ve içeriğini anlamalarıdır. PREMIS[31] ve PROV[32] gibi projeler bağlamında provenansı belgelemek için metaveri standartları, uygulamalara bağlı olarak birden fazla provenansa izin verebilir, ancak bu yapılandırmalar sadece koruma faaliyetlerini desteklemek için tasarlanmıştır ve belgelere erişimi ve belgelerin temsilini desteklememektedir. Dijital provenans; belge, erişim ve koruma teknolojilerinin maddeselliğini ve anlamlarının nasıl anlaşıldığına dair teknik, sosyal ve kültürel boyutlar eklenerek veri provenansı veya veri soyağacı anlayışını derinleştirebilir. Dijital provenans ayrıca, dijital koruma literatüründe arşivsel korumanın öncelikli olarak bilgi odaklı olduğuna dair bir yanlış anlamayı da değiştirme hedefindedir. Bak (2024, s. 852-859) dijital provenansın 3 boyutunu detaylarıyla açıklamıştır:

[31] Preservation Metadata Implementation Strategies
[32] Provenance

1. ***Teknik Boyut:*** Dijital sistemlerin oluşumunu sağlayan donanım ve yazılımlar, yalnızca teknik tasarım ve mühendislik ürünü değil; aynı zamanda tarihsel bilgi birikimi ve kültürel düşüncenin ürünleridir. Örneğin; TCP/IP gibi protokoller, internetin işleyişini mümkün kılan teknik standartlar olmasının yanında, geliştirildikleri dönemin ihtiyaçlarını, toplumsal beklentilerini ve iletişim anlayışını da yansıtır. Bu bağlamda, bu tür teknik yapıların kültürel değeri, bir şiir ya da resim gibi sanatsal eserlerle karşılaştırıldığında daha az değildir. Teknolojilerin kültür ifadeleri olarak anlaşılması, en temel işlemler de dahil olmak üzere hesaplama cihazlarının tüm yönlerine taşınır. Kısacası donanım, bir roman kadar kültürün bir ifadesidir. Tüm bunlar, dijital teknolojilerin en teknik hallerinde bile kültürel olduğunun bir göstergesidir. Bununla birlikte, teknik uyumluluk diğer her şey için temel bir gereklilik olmaya devam etmektedir. Bu nedenle, teknik gelişimin kültürel statüsünü inkâr etmeden, dijital teknolojilerin uygulanması ve kullanımına teknik bağımlılıklar getiren belirli donanım ve yazılım yapılandırmalarını, kodlama dillerini ve yapıları ile üretim ve bakım tekniklerini de anlamak gerekir. Bu teknik bağımlılıklar ve kültürel koşullar, dijital provenansın teknik boyutunun bir parçasıdır.
2. ***Sosyo-Teknik Boyut:*** Kullanım ve kullanıcı geçmişlerini temsil eder ve özellikle gerekli örtük bilgilere odaklanır. Bu boyut, kullanıcıların teknolojilerin nasıl yayıldığını, benimsendiğini ve bir kullanıcı topluluğu veya bir hesaplama topluluğu olarak adlandırılabilecek olgu içinde nasıl kullanıldığının etkisinden bahseder.
3. ***Sosyal Boyut:*** Dijital sistemlerin ve bunlar üzerinde oluşturulan belgelerin bir kültür içinde sahip olduğu anlamları kapsamaktadır. Bu çağrışımlar genel olarak dijital sistemlerle, tüm cihaz ve yazılım sınıflarıyla veya sistemlerin belirli örnekleriyle ilgili olabilir. Tüm arşiv belgeleri, içerikleri aracılığıyla iletilen doğrudan anlamın ve belgelerin kanıtladığı veya etkinleştirdiği işlevlerin ötesinde anlamlara sahiptir. Bir belgenin toplumsal anlamı ve değeri, bir kültür içindeki daha geniş toplumsal anlam ve değer sistemleriyle bağlantı kurduğunda, provenansın toplumsal boyutundan bahsetmek mümkün olabilmektedir. Dijital maddesellik diğer maddesellik biçimlerine göre kavramsal olarak daha zordur ve dijital bir dosyayı oluşturan bitlerin maddeselliğini, bitlere erişmek ve bunları işlemek için gerekli altyapıyı ve dijital temsilin kendisinin yapılandırmasını içerir. Dijital belgelerin toplumsal değeri, dijital maddesellikle ilişkilendirilebilir; örneğin belgeler, bunları üretmek için kullanılan dijital sistemler nedeniyle belirli bir toplumsal değere sahiptir. Drake (2016), dijital teknolojideki gelişmelere sosyal bir bakış açısından bakarak, provenansın teorik ve pratik

uygulamalarının iki nedenden ötürü bozulduğunu belirtmektedir. Birincisi, BİT sayesinde dünya nüfusunun büyük yüzdesinin yaşanmış deneyimleri hakkında yazma ve belge oluşturma gücüne ve potansiyeline sahip olduğudur. İkincisi ise bu artan etkenin insanlara ve topluluklara kendilerini adlandırma ve konumlandırma fırsatı vermesidir; bu, arşivsel süreçlerde provenansın belirlenmesi için olmazsa olmaz bir süreçtir. Dijital provenansın sosyal boyutu; teknik ve sosyo-teknik boyutlar gibi, belgelerin kendisiyle olduğu kadar, bilgisayar sistemlerinin geçmişiyle, sosyal ve kültürel değeriyle ve kullanımlarıyla da bağlantılıdır.

Sonuç olarak bir belge, çeşitli yönleriyle toplumsal öneme sahip olma potansiyeline sahip olabilir ve dijital provenans; bu potansiyeli arşivcilere dijital sistemlerin dünyayı nasıl şekillendirdiğine dair yönleri düşünmeleri ve keşfetmeleri için bir çerçeve sunarak sağlamaktadır.

Provenans, arşivler için kilit erişim noktalarından biri olarak bilgiye erişimde de oldukça önemli bir rol oynamaktadır (Bearman ve Lytle, 1985; Lemieux ve imProvenance Group, 2016). Bu durumda provenans, bir keşif aracı olarak kullanıma oldukça açıktır. Arşivsel keşif aracı olarak provenansın etkinliği; bilgi veya belgenin kapsamı ve standardizasyonu ile artmaktadır. Bir arşivci, bir derlemin oluşturulması ve saklanmasıyla ilgili ne kadar çok şey bilir ve aktarırsa, araştırmacıların bilgi ve belgeleri kullanabilmesi de o kadar kolay olacaktır.

Moreau (2010) ise; bilgisayar bilimi perspektifinden provenansın, sistem içerisindeki veri ve bilgilerin bağlamının anlaşılabilmesi için oldukça etkili bir kontrol aracı olduğunu, böyle bir aracın da sistem yapısını şeffaflaştıracağını öne sürmektedir. Böylece, veri ve bilgilerin hem yönetimi hem de erişilebilirliği ve kullanımının, belirli kurallar bütünü dahilinde arşivsel standartlara ve süreçlere uygunluğu belirlenebilmektedir. Şeffaflık; sistem ve içerdiği veri ile bilgilerin hesap verilebilirliğini sağlamaktadır. Bunun için de provenansın özgünlüğü ve güvenilirliğinin, belirli yaklaşımlar bağlamında oluşturulması gerekmektedir. Başka bir deyişle özellikle yönetişim bağlamında provenans ile ilgili veri ve bilgiler, resmi belgelerin de güvenilirliğini sağlamaktadır.

Makine öğrenmesi ve derin öğrenme gibi yöntemlerle analiz edilen veri setlerinde, her veri noktası veya çıktının hangi işlemlerden geçtiği, hangi algoritmalarla işlendiği, hangi kaynak dosyadan geldiği gibi bilgilerle birlikte tanımlanabilir. Bu bağlamda dijital provenans; hem materyalin üreticisinin kim olduğunu hem de veri işleme sürecine dair tüm adımların şeffaf bir biçimde izlenmesini mümkün kılar.

2.3.1.2. Fonlara Saygı İlkesi

Yapılan bir işin, belirli bir idari birime karşılık geldiği ve belgelerin buna göre korunması gerektiği inancından doğan fonlara saygı[33] ilkesi, 1840'lı yıllarda ilk defa Belçika ve Fransa'da ilan edilmiş ve sonraki yıllarda kıtada yaygınlaşmıştır (Sweeney, 2008). İlke son halini tek bir fon içindeki belgelerin, ilgili kurumun resmi faaliyetleri sırasında oluşturduğu düzene göre saklanması gerektiğini belirten 19. yüzyıl Alman ve Hollandalı arşivcilerin doktrini ile almıştır. Ayrıca, ilke; fonların kaynağı ve aynı kurumdan gelen diğer fonlarla ilişkisini açık bir şekilde yansıtacak genel bir düzenleme veya sınıflandırma şemasına yerleştirilmesini savunan provenans kavramıyla da genişletilmiştir. Hollandalı arşivciler Muller, Feith ve Fruin tarafından yazılan kitap, bu ilkeye son onayı vermiş ve teorideki değişiklik, arşivlerin ve arşivcilerin çalışmalarının karakterinde değişikliğe yol açmıştır. Arşivler; idari özelliklerini vurgulamaya, belgeleri devralmaları beklenen idari kurumlarla yeniden bağlantı kurmaya ve bunu sürdürmeye, kurumlardan düzenli transferler talep etmeye ve hangi belgelerin saklanacağına karar vermede önemli bir role kavuşmuştur (Posner, 1940, s. 167-168).

Bu ilke, tek bir kaynaktan (bir organizasyon, aile veya birey gibi) gelen arşiv belgelerinin diğer arşiv belge koleksiyonlarıyla karıştırılmaması ve tek bir birim olarak bir arada tutulması gerektiğini belirten bir ilkedir. Buradaki amaç, belgelerin oluşturulduğu bağlamı korumaktır. Çünkü bağlam, belgenin oluşma koşullarını ortaya çıkardığından dolayı korunması gerekmektedir. Uygulamada ise bu yaklaşım, benzer belgeler birden fazla koleksiyonda mevcut olsa bile, her koleksiyonun provenansını yansıtacak şekilde bozulmadan kalması anlamına gelmektedir. Başka bir deyişle, belgelerin arşivlere geldiği andaki bütünlüğünün sürekli korunmasını gerektirmektedir.

Fonlara saygı ilkesi, adından da anlaşılacağı üzere fonların sistematikliğine ve doğallığına vurgu yapmaktadır. Zelenyj (1996, s. 126), ilkenin anlaşılmasının zor olduğunu ve çoğunlukla da yanlış anlaşıldığını öne sürse de; 1841 yılında Fransa Ulusal Arşivleri Başkanı Natalis de Wailly tarafından yayımlanan "14 numaralı Genelge" ile *"bir kurum, kuruluş, aile ya da bireyin tüm belgelerinin bir koleksiyonunu oluşturma ve farklı fonları belirli bir düzene göre düzenleme"* talimatı verilmiştir. Böylece arşivsel düzenlemenin temel yöntemi olarak fonların belirli bir sistemde saklanmasının temelleri atılmıştır. Fonlara saygı, sahiplik kavramını arşiv belgelerinin düzenleyici ilkesi haline getirmiş ve koleksiyonların oluşturulmasından sorumlu kurum, kuruluş, aile ya da bireye göre oluşturulmasını benimsetmiştir. Bu yeni sistem, arşiv

[33] Respect des Fonds

belgesinin nerede, hangi süreçlerden geçtiği, hangi amaçla, kim için, muhatabı tarafından ne zaman ve nasıl alındığının bilinmesi ihtiyacını da gidermiştir (Duchein, 1983, s. 67). İlke; belge gruplarının üreticiliğe göre sıralanmasının, belgeler arasındaki ilişkileri kanıtlayarak ve üretildikleri çevreyi daha iyi göstererek özgünlük ve bağlam sağladığını göstermektedir.

Arşiv ve Belge Terminolojisi Sözlüğü (Pearce-Moses, 2005, s. 317, 346), fonlara saygı ilkesi ile provenans ilkesini birbirine eş olarak tanımlasa da, odaklandıkları olgular farklıdır.

Tablo 3. Provenans ve Fonlara Saygı İlkeleri Arasındaki Temel Farklılıklar

	Fonlara Saygı	**Provenans**
Odak	Üreticinin belgelerinin bir birim olarak bütünlüğünün korunması	Belgelerin üreticisinin belirlenmesi ve belgelenmesi
Uygulama Seviyesi	Organizasyonel - Belgeleri gruplandırmak ve düzenlemek	Tanımlayıcı - Kataloglama ve bağlamsal tanımlama
Amaç	Orijinal bağlamı korumak	Belgelerin geçmiş süreçleri
Çıktı	Belgelerin fiziksel veya entellektüel bütünlüğü	Tarihsel ve bağlamsal veriler

Tablo 3'te, literatürde zaman zaman birbirlerinin yerine kullanılabilen her iki ilkenin de farklılıkları gösterilmektedir. Fonlara saygı; koleksiyonların orijinal bağlamlarına göre bozulmadan saklanmasıyla ilgiliyken provenans ise belgelerin üreticisini veya kaynağını tanımlamayı ve kaydetmeyi vurgulamaktadır. Aralarındaki ince farklar, düzenleme faaliyetlerinde özellikle açığa çıkmaktadır. Fonlara saygı; belgelerin fiziksel ve entellektüel olarak nasıl gruplandırılacağını gösterirken; provenans ise araştırmacılar için bağlam sağlayan tanımlayıcı metaverileri izler. Bu farklılıklar göz önüne alındığında; fonlara saygı yapısal bir ilke iken provenans tanımlayıcı bir ilke olarak görülmektedir (Duchein, 1983).

2.3.1.3. Orijinal Düzen İlkesi

Bu ilke, sistematik bir biçimde ilk defa Hollandalı arşivciler Muller, Feith ve Fruin (2003/1898, s. 52) tarafından 1898 yılında yayımlanan "Arşivlerin Düzenlenmesi ve Tanımlanması Kılavuzu"nda şu şekilde bahsedilmiştir: *"Düzenleme sistemi, esas olarak onu üreten idari organın organizasyonuna karşılık gelen arşiv koleksiyonunun orijinal organizasyonuna dayanmalıdır"*.

Arşivcilik Terimleri Sözlüğü'nde "Asli Düzen" olarak belirtilen ilke; *"belgelerin güncel kullanımda olduğu süreç içerisinde uygulanan düzen içinde saklanması prensibi"* olarak tanımlanmıştır (Karakaş, Rukancı ve Anameriç, 2009, s. 7).

Arşiv ve Belge Terminolojisi Sözlüğü'nde ilke, şu şekilde tanımlamaktadır: *"Belgelerin üreticisi tarafından oluşturulan organizasyonu ve sırası"* (Pearce-Moses, 2005, s. 280-281).

Sözlüğe göre ilke; arşivlerin temel bir ilkesidir ve belgeleri orijinal düzende tutmanın iki amaca hizmet ettiğini belirtir:

- Belgelerin bağlamından çıkarılabilecek mevcut ilişkileri ve kanıtsal önemi koruması,
- Belge üreticisini, belgelere erişmek için yeni erişim araçları oluşturma zahmetinden kurtarması.

Schellenberg (1961, s. 18) ilkeyi, belgelerin mevcut yaşamları boyunca kendilerine uygulanan düzende tutulması gerektiği şeklinde tanımlamakta ve ilkenin, Alman arşivciler tarafından 1881'de Prusya Devlet Arşivleri'ndeki düzenleme çalışmalarında kullanılmak üzere formüle edilen "Registraturprinzip"in bir sonucu olduğunu belirtmektedir. İlke, tam olarak resmi belgelerin, ilgili birimin resmi faaliyetleri sırasında oluşturuldukları düzene göre saklanması gerektiğini vurgulamaktadır. İlke; belgeler arşive devredilmeden önce resmi olarak uygun şekilde düzenlendiğinde uygulanabilmektedir.

Orijinal düzen ilkesi kapsamında; yanlış dosyalanmış öğeler, uygun yerlerinde yeniden dosyalanabilir. Üretici, öğeleri gelişigüzel bir şekilde saklamışsa, bir koleksiyon anlamlı bir düzene sahip olmayabilir. Bu gibi durumlarda, arşivciler düzenlemeyi ve tanımlamayı kolaylaştırmak için belirli bir düzen uygular. Dolaysıyla bir arşiv koleksiyonunun düzenlenmesinde, ilk önce mümkün olduğunca orijinal düzen yeniden kurulmalıdır. Ancak bundan sonra, bu düzenin dışına çıkmanın arzu edilip edilmeyeceği ve ne ölçüde arzu edilir olduğuna karar verilebilir (Muller, Feith ve Fruin, 1898/2003, s. 58). Bu kural, bir arşiv koleksiyonunun eski organizasyonuna sıkı sıkıya bağlılığına dayanmaz ve eski düzenin kendi başına iyileştirmeye açık olmadığı gerekçesiyle yeniden kurulmasını talep etmez. Ancak, bu eski düzenin dışına çıkma fırsatı olabileceğini kabul eder. Önceki bölümde belirtilen ilkeye uygun olarak genel eski düzenin, eski organizasyonun ihtiyaçlarına göre kurulduğu ve ona sıkı sıkıya bağlı olduğu gerçeğini vurgular. Bunun için öncelikle arşiv koleksiyonunun canlı bir organizma iken nasıl oluştuğunu ve nasıl değiştiğini incelemek gerekmektedir (Douglas, 2013).

Boles (1982, s. 29); ilkenin uzunca bir zamandır kullanıldığını belirtmekte ve bu durumu da uygulama kolaylığına ve teorik gerekçelere bağlamaktadır. Ona göre uygulamada, ilkenin iki dayanağı bulunmaktadır. Birincisi; çoğunlukla işe yaramasıdır. İlke, evrensel olarak uygulanabilir olmasa da ve bu duruma çok sayıda istisna yapılması gerekse de, yine de tamamen uygulanamayacak kadar kusurlu değildir. İkincisi ise ekonomisidir. Sürekli olarak yetersiz ödenek sağlanan bütçelerle karşı karşıya kalan arşivciler, teorik değerlendirmeler ne olursa olsun, büyük belge grupları üzerinde alternatif organizasyon şemaları uygulama

yeteneğine sahip olamamış ve böylece bu ilke zorunlu olarak kullanılmaya devam etmiştir. Teorik gerekçeler ise; tarihsel ve arşivsel olmak üzere iki amaca hizmet eder. Tarihsel olarak; belgelerin üreticilerine göre düzenlenmesi, belgelerin içeriğinden bağımsız olarak belge üreticisinin karakteri ve organizasyonu hakkında bilgi vermektedir. Böylece, ilkenin kanıt değeri yansıtılmış olur. İlke ayrıca, provenans ilkesini önemli ölçüde genişletmesi bakımından arşivsel bir değere de sahiptir. Provenans, temelde arşivcilerin bir grup belgeyi ayrı bir birim şeklinde muhafaza ederek, dosyalara ne ekleme ne de çıkarma yaparak bir belge bütününün üreticisine saygı göstermeleri konusunda ısrar eder. Belgelerin orijinal düzeninin korunması da böylece bu saygıyı güçlendirir. Yalnızca üreticinin belge yapısının bütünlüğüne değil, aynı zamanda üreticinin düzenine de saygı gösterilir. Bu çerçevede orijinal düzen, provenans ilkesinin organik bir bileşeni olarak değerlendirilebilir.

Arşiv teorisinin uygulanma durumu, görünürde net olan süreci daha da karmaşıklaştırmaktadır. Bir arşivci, provenans ve orijinal düzen arasındaki ilişkiden güçlü bir şekilde etkilendiğinde, orijinal düzenin kesinlikle bozulmaması gerektiğini bilir çünkü eğer böyle bir bozma eylemi yapılırsa; sadece kanıtlar kaybolmakla kalmayacak, aynı zamanda arşivsel gerçekler de bozulacaktır. Sonuç olarak arşivci, kanıtsal olarak üstün ve otantik belgelere erişim ne kadar zayıf olursa olsun, orijinal düzeni olduğu gibi bırakma eğiliminde olacaktır (Greene ve Meissner, 2005).

Tablo 4'te orijinal düzen ile provenans ilkesi arasındaki farklılıklar özetlenmiştir. Orijinal düzen ilkesi, belgelerin üreticisi tarafından aktif olarak kullanıldığı zamankiyle aynı iç düzenlemede tutulması gerektiğini vurgulaması nedeniyle, odaklandığı alanlar açısından provenans ilkesinden kesin bir şekilde ayrılır. Provenans ilkesi ise belgelerin orijinal üreticisine bağlı kalmasını sağlamanın yanında, buna bağlı olarak bağlamın ve özgünlüğün korunmasını amaçlar.

Tablo 4. Orijinal Düzen ve Provenans İlkeleri Arasındaki Temel Farklılıklar

	Orijinal Düzen	**Provenans**
Odak	Üretici tarafından düzenleme	Belgelerin üreticisinin belirlenmesi ve belgelenmesi
Uygulama Seviyesi	Koruyucu - Orijinal organizasyonun bozulmadan korunması sağlanır	Tanımlayıcı - Kataloglama ve bağlamsal tanımlamada uygulanır
Amaç	İşlevsel ilişkilerin korunması	Belgelerin geçmişinin ve kökeninin izlenmesi
Çıktı	Dosyalama sisteminin korunması	Tarihsel ve bağlamsal veriler

Tablo 4'ten görüldüğü üzere; orijinal düzen, belgelerin oluşturulduğu andaki düzenlerinin korunmasına yoğunlaşırken; provenans, belgelerin hangi kişi ya da kurum tarafından

üretildiğini ortaya koymaya odaklanmaktadır. Orijinal düzen, belgelerin ilk oluşturuldukları sistem veya yapı bozulmadan korunmasını hedeflerken, provenans ise daha çok tanımlama, kataloglama ve bağlamsal tanımlama çalışmalarında kullanılır. Orijinal düzen, belgeler arasındaki işlevsel bağlantıları korumayı amaçlarken; provenans, belgelerin geçmişini ve kökenini takip etmeyi amaçlar. Orijinal düzenin sonucunda belgeler, ilk oluşturuldukları dosyalama sistemine yakın şekilde saklanırken, provenansta ise belgelerin geçmişi ve bağlamına dair bilgiler üretilir.

Tablo 5. Orijinal Düzen ve Fonlara Saygı İlkeleri Arasındaki Temel Farklılıklar

	Orijinal Düzen	**Fonlara Saygı**
Odak	Üretici tarafından düzenleme	Üreticinin belgelerinin bir birim olarak bütünlüğünün korunması
Uygulama Seviyesi	Koruyucu - Orijinal organizasyonun bozulmadan korunması sağlanır	Organizasyonel - Belgeleri gruplandırmak ve düzenlemek
Amaç	İşlevsel ilişkilerin korunması	Orijinal bağlamı korumak
Çıktı	Dosyalama sisteminin korunması	Belgelerin fiziksel veya entellektüel bütünlüğü

Tablo 5'te ise orijinal düzen ve fonlara saygı ilkeleri arasındaki farklılıklar gösterilmektedir. Orijinal düzende; belgeler arasındaki işlevsel ilişkilerin korunması hedeflenerek, belge üreticileri tarafından belirlenen düzeninin korunması odak noktasıdır. Fonlara saygı ilkesi ise; üreticinin belgelerinin tamamını tek bir birim olarak korumaya odaklanır. Orijinal bağlamı korumak amacıyla, belgeler üreticilerine göre gruplandırılarak organizasyonel düzeyde etkinlik sağlanmaktadır.

2.4. Arşivsel Düşünme

Önce ortaçağ belgeleri, ardından giderek daha fazla sayıda diğer kamu belgeleri ulusal anlamda önemli hale gelince devletler, arşivlerin "yetenekli" koruyucuların bakımına emanet edilmesi gerektiğini farketmiştir (Posner, 1940). Zaman içerisindeki bu değişim, belgelerin korunması ve yönetiminin organize bir uygulamaya dönüştürüldüğü bugünkü arşiv kurumlarının başlangıcına öncülük etmiştir. Koruyucu olarak arşivcinin rolü de zaman içerisinde sadece belgelerin fiziksel bütünlüğünü korumakla kalmayıp erişilebilirliğini ve gelecekteki kullanımlar için uygun bağlamı sağlamak olmuştur. Bu dönüşüm, arşiv teorisindeki temel ilkelerle, özellikle de belgelerin doğrudan idari kullanımının ötesinde uzun vadeli önemini vurgulayan arşiv değeri kavramıyla da uyumludur. Arşivler, zaman içinde tarihin ve mirasın daha geniş toplumsal ve politik yönlerini yansıtacak şekilde düzenlendiği bir alan haline gelmiş ve arşivciler, tarihi belgelerin özgünlüğünü ve sürekliliğini koruma konusunda kritik bir

sorumluluk üstlenerek daha çok toplumsal kapsayıcılık bağlamına ulaşmışlardır (Booms, Joldersma ve Klumpenhouwer, 1987, s. 78). Yakın bir zamandan itibaren bu durum, arşivsel düşünmenin temelini oluşturmuştur.

Lowry ve MacNeil (2021), arşivsel süreçlerin tarihi belgeleri muhafaza etmekten çok daha fazlası olduğunu vurgulayarak; geçmişin nasıl anlaşıldığının, bugünün nasıl korunduğunun ve geleceğin nasıl şekillenebileceğine dair anahtar bakış açısının arşivsel düşünme kavramı ile elde edilebileceğini öne sürmüştür. Çünkü belgeleri korumanın ve yönetmenin etik, tarihi ve sosyal sonuçlarının eleştirel ve katılımcı bir şekilde değerlendirilmesi gerekmektedir.

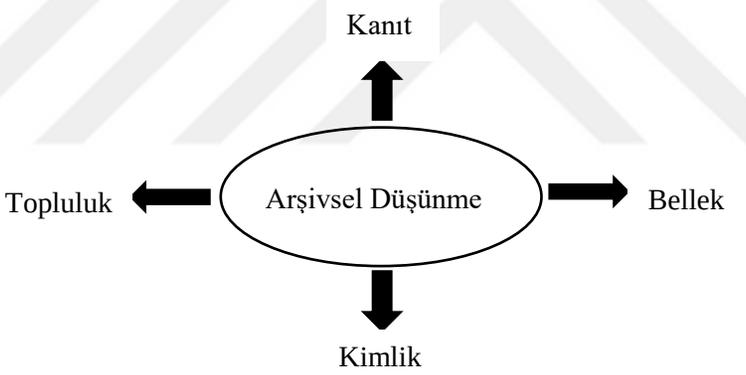
Arşivsel düşünme terimi, arşiv çalışmalarında belgelerin korunması ve işlenmesiyle ilgili teori, ilke ve eylemleri ifade etmek için kullanılmaktadır (Ahmed, 2021, s. 10). Başka bir deyişle arşivsel düşünme; arşivcilerin ve bilgi profesyonellerinin arşivleri nasıl yönettiklerini, yorumladıklarını ve kavramsallaştırdıklarını etkileyen entelektüel yaklaşımı ve zihniyeti ifade etmektedir. Ayrıca, dijital veya fiziksel materyallerin oluşturulması, düzenlenmesi, korunması ve erişilebilirliği ile ilgili kararları oluşturan ilkeler, teoriler ve çerçeveleri de kapsamaktadır. Arşivsel düşünme, sadece tarihi nitelikteki belgelerle nasıl etkileşim kurulduğunu değil, bu belgelerin hangi ortama kaydedildiğinden bağımsız olarak bilgiyi gelecek için korumanın önemini de vurgulamaktadır.

Sassoon (2007, s. 41) arşivsel düşünmenin; arşivciler tarafından materyallerin arşivsel süreçlerde nasıl düşünüldüğü bağlamında anlaşılması gerektiğini vurgulamaktadır. Böyle bir düşünme tarzı, arşivler için yapılanların temelini oluşturur. Jones (2016, s. 214) ise, arşivsel düşünme anlayışının temel yönlerini; arşiv koleksiyonlarını anlayabilmek için materyallerin toplu tanımlanması; materyallerin saklandığı ve kullanıldığı düzenin anlamına katkıda bulunduğu inancı, provenans bilgisinin yakalanması, belgelerin oluşturulduğu ve yönetildiği bağlamın veya bağlamların anlaşılmasının önemi, tüm bu yönlerin her birinin, her bir bileşene anlam kazandırmaya yardımcı olan bir ilişki ve bağlantı sisteminin parçası olduğu anlayışı, bu sistemlerin uzun süreli erişilebilir ve anlaşılır kalabilecek şekilde belgelenmesini ve yönetilmesini sağlayabilmek için sürdürülebilirliğe odaklanma olarak belirtmektedir.

Weld'e göre (2014, s. 236) arşivsel düşünme, arşivsel süreçlerin de ötesine geçen eleştirel bir soyutlama olarak da görülebilir. Bu anlamda, hangi materyallerin nasıl korunacağı; arşivlerin geçmişi, bugün için nasıl şekillendirilebileceği, toplulukların seslerini nasıl duyurabileceği ve erişim ile gizlilik arasındaki dengenin nasıl sağlanacağı gibi sorularla ilgilenmektedir.

Arşivsel düşünme kavramı uzun süredir arşivlerin nesnel ve tarafsız alanlar olduğu fikrine karşı çıkan başta post-modern teori olmak üzere çeşitli fikirlerden etkilenmiştir (Cook, 2001; Harris, 2005; Jimerson, 2007). Post-modern arşiv teorisyenleri, arşiv eyleminin doğası gereği öznel olduğunu; sosyal, siyasi ve kültürel faktörler tarafından şekillendirildiğini savunmaktadır (Schwartz ve Cook, 2002; Carter, 2006; Caswell, 2023). Bu durum zaman içerisinde arşiv çalışmalarına daha eleştirel bir yaklaşım getirmiş, Jenkinsoncu yaklaşıma sahip "arşivcilerin, belgelerin sadece pasif koruyucuları" oldukları anlayışı terkedilerek "tarihin yorumcuları" şeklindeki yeni rollerini tanımlamaya başlamıştır.

Cook (2013), entelektüel akımların modern öncesi dönemden moderne, post-modernden dönemden çağdaşa doğru değiştikçe, arşivsel düşünmenin odağının da kanıttan belleğe, kimlik ve topluluğa doğru kaydığını vurgulamaktadır. Bu bağlamda arşiv paradigmasının eskiden yeniye doğru dört boyutunu kanıt, bellek, kimlik ve topluluk olarak belirlemiştir. Ona göre arşivciler; farklı yerlerde hemen hemen aynı süreçleri (değerlendirme, edinme, işleme, tanımlama, koruma, erişime sunma vb.) gerçekleştirdiklerinden dolayı; sorumlulukları, toplumsal önemleri ve etkisi, bir ideale ve bireysellikten ziyade toplumsallığa ulaşmalıdır.



Şekil 2. Arşivsel Düşünmenin Dört Yönü

Şekil 2'de gösterildiği üzere kanıt, ilk arşiv paradigmasıdır ve 1930'lara kadar profesyonel söyleme hâkim olmuş ve günümüzde de önemli bir arşivcilik kaygısı olarak devam etmektedir. Paradigma, arşivlerde biriken "doğal" kalıntıların kanıt olarak korunması etrafında şekillenmiştir ve tarafsız bir koruyucu olarak arşivcinin temel profesyonel odağı, hukuki özelikteki resmi belgelerin gelecek kuşaklar tarafından otantik ve güvenilir belgesel kaynaklar olarak kullanılması ve anlaşılması için bir bağlama oturtulması amacıyla düzenlenmesi ve tanımlanması olmuştur (Cook, 2013, s. 106).

Bellek, ikinci arşiv paradigmasıdır ve 1930'lardan 1970'lere kadar paradigmanın anahtar kavramı olmuştur. Bu paradigma, belgelerin tarihsel kaynaklar olarak değerlendirilmesiyle birlikte, arşivcinin tarihçi özelliğine miras kalan hukuki bir kanıtı korumak yerine, öznel olarak

kültürel bir bellek kaynağı üretmesiyle belirgin bir şekilde ilgilidir. Bu bellek kaynağı; modern iş araçları ve süreçleri kullanılarak, daha büyük olan varlıklar için daha verimli bir şekilde yönetilmiş ve arşivcilerin değerlendirme ve tanımlamaya yönelik yeni yaklaşımları desteklemek için belgelerin ve üreticilerinin tarihine ilişkin ayrıntılı araştırmalarını yansıtmıştır. Ortaya çıkan arşiv, elbette halen insani ve kurumsal faaliyetlerin kanıtıdır, ancak bu kanıtların üretildiği, değerlendirildiği, edinildiği, tanımlandığı ve anlaşıldığı bağlam artık değişmiştir (Cook, 2013, s. 108).

Kimlik ise üçüncü arşiv paradigmasıdır ve 1970'lerden sonraki yıllarda arşivcinin profesyonel uzmanlığında anahtar kavramdır. Bu paradigma, belirgin bir şekilde arşivlerin toplumsal bir kaynak olarak sayılması gerektiğine odaklanmıştır. Arşivler, daha önceki zihinsel çerçevelere hâkim olan monolitik kalıplardan ziyade postmodern ve dijital dünyanın çoğulcu ve belirsiz doğasına giderek daha fazla saygı duyan bir toplumsal kaynak olarak değerlendirilebilir. Arşivler bu dönemde aynı zamanda akademik elitleri destekleyen bir kültürel miras kaynağı olmaktan çıkıp kimlik ve adalet için toplumsal bir temel haline dönmüşmüştür. Arşivcinin kendi kimliği, toplumun kimliğini daha karmaşık kanıt kavramlarına dayanan ortak anılar aracılığıyla bulmasına öncülük eden uzman olmaya doğru genişlemiştir. Artık arşivci, arşivsel belleğe başvurarak toplumun kendi çoklu kimliklerini oluşturmasına yardımcı olan bilinçli bir arabulucu ve hızla değişen toplumsal organizasyonların ve dijital ortamın baş döndürücü karmaşıklığı karşısında kanıtları koruyan aktif bir uzman olarak kendi kimliğinin arayışındadır (Cook, 2013, s. 112).

Topluluk, dördüncü arşiv paradigmasıdır ve günümüzde oldukça baskındır. Yeni dijital, politik ve çoğulcu evrende profesyonel arşivcilerin, yerleşik arşivlerdeki materyalleri edinmek yerine, arşivsel süreçleri toplumdaki birçok kişiyle paylaşılan katılımcı bir süreç olarak teşvik etmek için toplum içinde çalışan mentörler ve koçlar haline dönüştükleri bir dönem olarak vurgulanmaktadır. Arşivciler; toplulukların kanıt üretme ve doğrulama, hikâye anlatma, bellek oluşturma ve genellikle farklı olan ilişkilerini belgeleme konusunda sahip oldukları yeni yöntemleri öğrenmelidir. Topluluk arşivleri; uzmanlık ve bilgiyi her iki yönde de paylaşarak, bellek ve kimlik için genişletilmiş, canlı, kullanılabilir ve bağlamsallaştırılmış belgelere sahip olma konusunda kendi çıkarlarına ve nihayetinde toplumun çıkarlarına hizmet etmek için arşivlerini tutmaya teşvik eder. Arşivciler aynı zamanda toplumun ilgili üyelerini arşivler ve onların topluluk arşivleri ile etkileşimli diyaloglara dahil edebilir ve baskın kültür dilinden, terminolojisinden ve tanımlarından "öteki"ne karşı duyarlılığa ve belgelerin idari ve hukuki

önemleri kadar toplumlar için taşıdığı duygusal, dini, sembolik ve kültürel değerlere ilişkin keskin bir farkındalığa geçiş yapmaları gerekmektedir (Cook, 2013, s. 115).

Punzalan ve Caswell (2016), arşiv bilimi alanında sosyal adalet konusunda yapılan çalışmaları; toplumun yeterince temsil edilmeyen ve marjinalleştirilmiş kesimlerinin dahil edilmesi, arşiv kavramlarının yeniden yorumlanması ve genişletilmesi, topluluk arşivlerinin geliştirilmesi, arşiv eğitimi ve öğretiminin yeniden düşünülmesi ve insan hakları ihlallerini belgeleme çabaları olarak sınıflandırmıştır. Arşiv alanındaki çalışmaların daha çok sosyal adalet konularını kapsayabilmesi için gelecekteki çalışmaları çerçeveleyebilecek üç temel alanı; "sosyal adalet anlayışında hak temelli yaklaşımların ötesine geçilmesi", "marjinalleşmenin ırksal ve ekonomik yapılarının incelenmesi" ve "dijitalleşme ile açık erişimin etki ve sonuçlarının anlaşılması" şeklinde belirlemiştir.

Dijital çağda kavram; verilerin oluşturulması, depolanması ve erişilebilirliğinde teknolojinin dönüştürücü etkisi nedeniyle önemli ölçüde evrimleşmiştir. Dijital teknolojilerin yaygınlaşması; kalıcılık, maddesellik ve erişilebilirlik konusundaki geleneksel kavramları da derinden etkileyerek bozmuştur. Materyallerin giderek daha fazla dijital oluşturulması dijital formatların, metaverilerin ve dosya sistemlerinin anlaşılmasını karmaşıklaştırmaktadır. Buna bağlı olarak da içgörü, anlam ve bağlam gibi süreç bazlı ifadelerin anlaşılması da zorlaşmıştır. Dolaysıyla; dijital ortamdaki bilgilerin biçim olarak yeniden düşünülüp tasarlanmasına ve hatta yeni yaklaşımlar doğmasına yönelik çalışmalar gitgide artmaktadır (Klareld ve Gidlund, 2017; Jensen, 2020; Anderson, 2020; Goudarouli, 2023). Makine öğrenmesi ve derin öğrenme gibi yapay zekâ teknikleri; sınıflandırma, içgörü çıkarma ve metaveri oluşturma gibi süreçlerin otomatikleştirilmesini ve arşivcilerin daha üst düzey görevlere odaklanmasını sağlayabilir. Ayrıca, dijital platformların keşfedilebilirliğinin artması, arşivler ile kullanıcılar arasındaki bağı kuvvetlendirebilir ve erişimin daha fazla demokratikleştirilmesini sağlayabilir.

Literatürdeki çalışmalar incelendiğinde, arşivsel düşünme çalışmalarının daha çok aşağıdaki alanlarda yapıldığı görülebilir:

- ***Provenans Araştırması:*** Arşivsel düşünmenin de temellerinden biri olarak; kişi, topluluk veya kurumların, belgelerini nasıl oluşturduğunu ve bu belgelerin nereden geldiğinin anlaşılması bütünlüklerinin korunmasında anahtar görevi görür. Dijital belgelerin mevcut teknolojide provenansını izleyebilmenin zorluklarını ön plana çıkaran ve bu durumun iyileştirilebilmesi için Kaynak Tanımlama Çerçevesi (RDF)[34] ve RiC gibi

[34] Resource Description Framework

standart ve araçlarla provenansın izlenebileceğinin vurgulanması (Michetti, 2016); hükümetler ve yerel yönetimlerin yapıları ve prosedürleri üzerinde daha yoğun bir şekilde düşünülmesi (Fuhrmeister ve Hopp, 2019); yerli toplulukların belleklerinin izlenmesi (Cowan, 2018); sömürge dönemi müze eserlerinin fiziksel kalıntılarını inceleme ve iade etme sürecinde dönem bilgisini ve erişilebilirliği yeniden üretme riskinin ele alınması (Kuever, 2024) gibi çalışmalar, provenans araştırmalarında öne çıkmaktadır.

- ***Değerlendirme:*** Arşivsel düşünme, aynı zamanda hangi belgelerin kalıcı değere sahip olduğunu ve gelecek için korunması gerektiği konusunda eleştirel karar vermeyi içerir. Resmi ve toplumsal önceliklerin zaman içerisinde değişebileceğinden hareketle arşivsel değerin değişebileceği ve değerlendirme kriterlerinin farklılaşabileceği ihtimali her zaman bulunmaktadır. Cook (2013) tarafından önerilen "arşivsel düşünmenin dört paradigması"nın değerlendirme süreci ile bağdaştırılması (Foscarini, 2017); dijital insani bilimler alanındaki çağdaş uygulamaların, arşiv belgelerini analitik ve dijital geçmiş anlatısı amaçları için "ham" malzeme olarak nasıl kullanabileceğinin araştırılması (Bauer ve Wrisley, 2020); arşivleri uzun süredir "rahatsız eden" sessizliklerin[35] duyulabilmesi için çeşitli topluluklar ile birlikte katılımcı değerlendirme (Cook, 2011a) gibi çalışmalar ise değerlendirme alanında ön plana çıkmaktadır.
- ***Erişim ve Etik:*** Etik hususlar arşivsel düşünmenin merkezinde yer alan boyutlardan biridir. Arşivciler; gizliliğe ve yasal yükümlülüklere uyarken, materyalleri koruma ihtiyacı ile bunları toplumun erişimine sunma sorumluluğu arasında denge kurmak durumdadır. Dijital platformların ortaya çıkmasıyla birlikte, bilgi ve belgelerin topluluklara nasıl erişilebilir hale getirilebileceği önemli bir sorunsal olarak ortaya çıkmıştır. Bazı arşivcilerin, koleksiyonları tüm topluma sunmanın etik olup olmadığına ve kendilerini kültürün "koruyucuları" olarak adlandıran bazı arşivcilerin hâlâ nadir olsa da kısıtlama taraftarı olduklarına yönelik (Prelinger, 2007) ve arşivlerdeki gerçeğin doğasının karmaşık olduğundan hareketle kurumların ve arşivcilerin üretken gücünün analizi ve hassas bilgileri keşfetmenin etik ikileminin incelenmesi (Tesar, 2014) gibi çalışmalar bulunmaktadır.
- ***Bellek:*** Arşivsel düşünme, arşivlerin kolektif hafızayı nasıl şekillendirdiğini ve güç yapılarını nasıl etkilediğini aynı zamanda eleştirel bir şekilde de incelemektedir. Chazan,

[35] Yerli halkın varlığı

Baldwin ve Madokoro'ya göre (2015) arşivler tarafsız depolar değildir; onları üreten ve muhafaza edenlerin bakış açılarını ve önceliklerini yansıtır. Yazarlara göre marjinal grup ve toplulukların seslerini dışlayan bir arşivcilik yerine bu grup ve toplulukların geçmişlerindeki boşlukları, sessizlikleri ve önyargıları kırmak için yeni nesil arşivsel düşünme savunulmalıdır. İki temel depolama biçimi olan bellek ve arşiv kavramlarının birbirleriyle ilişkilerinin maddesellik ve epistemolojik incelenmesi (Jaspert vd., 2024); arşivin, olayları kaydettiği kadar ürettiği varsayımıyla arşiv uygulamaları ve arşiv belgeleri arasındaki ilişkinin "anı" oluşturmaya yardımcı olması (Mauthner ve Gardos, 2015); kişisel, kolektif, tarihsel ve arşivsel olmak üzere dört tür hafıza bağlamında, toplumdaki güç ilişkilerinin incelendiği ve arşivcilerin, toplumdaki zayıf grupların belgelerine yönelik önyargıyı fark ederek ve bunun üstesinden gelerek gelecek nesiller için evrensel kültürün tüm seslerinin incelemesini sağlaması zorunluluğu (Jimerson, 2003) gibi çalışmalar ön plana çıkmaktadır.

- ***Eşitlik:*** Arşivsel düşünme dijital uçurum, temsil ve kültürel duyarlılık gibi konularda kapsayıcı ve eşitlikçi yaklaşımlar benimsenmesini gerektirmektedir. Böylece; seslerini duyuramayan çeşitli toplulukların tarihlerinin korunması ve otantik bir şekilde temsil edilebilirliği sağlanmış olur. Çeşitlilik, eşitlik, kapsayıcılık ve sosyal adaletin arşivcilik lisansüstü müfredatına eklenmesi (Poole ve Todd-Diaz, 2023); tanımlama süreçlerindeki dil yaklaşımının gözden geçirilerek, eşitlik ve uzlaştırma odaklı düşüncenin arşivsel uygulamalara nasıl entegre edilebileceğinin incelenmesi (Robichaud, 2021) gibi çalışmalar bulunmaktadır.

2.5. Hesaplamalı Düşünme

İnsanın temel özellikleri arasında düşünme yetisi ile çeşitli durumlar karşısında yargıda bulunabilme kapasitesi yer almaktadır. Düşünme; bireyin zihinsel temsiller üzerinde işlem yapmasına olanak tanıyan, analiz, sentez, problem çözme, karar verme ve eleştirel değerlendirme gibi becerileri içeren bilinçli bir bilişsel süreçtir (Wang ve Chiew, 2010; Saiz ve Rivas, 2023). Bu temel özellikler; insanlığın ilk varoluşundan itibaren kendisine doğa karşısında zamanla avantaj sağlamıştır. İnsanlık; deneme-yanılma yöntemiyle uyguladığı av teknikleri ile veya tesadüfen bulduğu ateş ile doğa karşısında daha uzun süreler ayakta kalmayı başarabilmiştir. Yüzbinlerce yıl boyunca bu temel özellikler, daha da gelişmiş ve günümüze kadar ulaşan devasa bir ortak miras üretilmesinde temel olgular olmuştur. Düşünme kavramı ile eşzamanlı olarak işleyen, kimi zaman basit kimi zaman ise karmaşık biçimlerde ortaya çıkan bir "hesaplama" süreci de söz konusudur. Bu süreç, yalnızca soyut düşünceyle sınırlı kalmayıp,

problemlerin çözümünde tarih boyunca önemli bir rol oynamıştır. Örneğin Antik Mısır'da, Nil Nehri'nin taşkınlarının öngörülmesi ve piramitlerin inşası, dönemin teknik olanakları çerçevesinde belirli hesaplamaların yapılmasını gerektirmiştir. Özellikle modern çağın başlangıcından itibaren bilim insanları, bilimsel ve teknolojik yeniliklerin topluma yayılması adına sistematik çalışmalar yürütmüş; bu çalışmalar neticesinde günümüz yaşamının konforuna zemin hazırlanmıştır. Tüm bu gelişmelerin temelinde ise insanın düşünme yetisi ile hesaplama kapasitesi yer almaktadır.

Hesaplama teknolojisinin neredeyse her çalışma alanına uygulanması, zaman içerisinde işlerin yapılma şeklini değiştirmiştir. İnsan zihni bugüne kadar sahip olunan en güçlü problem çözme aracı olmasına rağmen; insan düşüncesinin, gücünü bilgisayar ve diğer dijital araçlarla genişletme yeteneği, günlük hayatın ve işlerin önemli bir parçası haline gelmiştir. Bu bağlamda, bilgisayarların ve dijital teknolojilerin hangi koşullarda, ne şekilde ve ne zaman problem çözme süreçlerine katkı sağlayabileceğini anlamak büyük önem taşımaktadır (Bar, Harrison ve Conery, 2011, s. 23). Dolayısıyla, bilgi teknolojilerine yönelik temel yetkinliklerin yanı sıra güçlü iletişim becerileri de çağdaş dijital okuryazarlığın temel bileşenleri arasında yer almaktadır.

Son yıllarda, hızlı bir şekilde dijitalleşme gerçekleşmekte ve yaşamın neredeyse tüm alanlarında hesaplama mantığı ve ilkeleriyle uyumlu bir şekilde düşünme yeteneğinin artık bir zorunluluk haline geldiği açıkça görülmektedir (Aho, 2012; Kalelioğlu, Gülbahar ve Kukul, 2016; Li vd., 2020; Denning ve Tedre, 2021). Biraz önce de bahsedildiği gibi tarihin başlangıcından günümüze kadar devam eden ve gelecekte de devam edecek olan her türlü bilimsel ve teknolojik gelişmede mutlaka bir "hesaplama" arka planının olduğu bir gerçektir. Bu bağlamda hesaplamalı düşünme kavramının; özellikle günümüzde insanın düşünme, akıl yürütme ve problem çözme konusundaki doğal yeteneğinin bir uzantısı haline geldiği söylenebilir. Bu kavram, bireylerin karmaşık durumlara sistematik bir şekilde yaklaşmasına, sorunları daha küçük ve yönetilebilir parçalara ayırmasına, örüntüleri tanımasına ve günlük problem çözme süreçlerine benzer biçimde adım adım çözümler geliştirmesine olanak tanır. Hesaplamalı düşünme sayesinde insanlık; yeni cihazlar icat etmek, bilimsel ilkeleri keşfetmek ya da verimliliği artırmak amacıyla araçlar tasarlamak gibi süreçleri sistematik olarak planlayabilir, analiz edebilir ve etkili çözümler üretebilir duruma gelmiştir. Böyle bir sistematik yaklaşım, karmaşık sorunların daha net anlaşılmasını sağlayarak üreticiliği de geliştirmektedir. Hesaplamalı düşünme; insan zekâsı ile makine arasındaki boşluğu doldurarak; karar verme ve problem çözmede teknolojiden daha etkin bir şekilde

yararlanılmasını sağlamaktadır (Tadeu ve Brigas, 2022). Başka bir deyişle; kavram bir anlamda, insanın doğal hesaplama ve yargılama eğilimini daha sade ve güçlü bir süreçle birleştirmektedir.

Hesaplamalı düşünme kavramı, ilk olarak Papert (1980) tarafından "prosedürel düşünme" olarak anılmış ve geometrik problemlerin çözümünde bilgisayar ve yazılım kullanımı üzerine yaptığı bir araştırmada, bir problem ile çözümü arasındaki ilişkinin tanımlanmasında ve verilerin yapılandırılmasında hesaplamalı düşünmenin kullanılabileceği iddia edilmiş olsa da (Kılıçarslan Cansu ve Cansu, 2019, s. 1), günümüzdeki kullanımıyla kavram, Wing (2006, s. 1) tarafından açıklanmış ve *"... görünüşte zor olan bir problemi; indirgeme, yerleştirme, dönüştürme veya simülasyon yoluyla nasıl çözeceğimizi bildiğimiz bir problemi yeniden formüle etmektir"* şeklinde tanımlanmıştır. Başka bir deyişle hesaplamalı düşünme, sorunların sistematik ve mantıksal bir şekilde çözülmesini sağlayan bilişsel bir süreçtir. Papert'a göre prosedür kavramı ise; *"bir cihaz tarafından gerçekleştirilebilecek adım adım bir dizi talimat"* olarak tanımlanmıştır (Selby ve Woollard, 2010, s. 3). Bu, algoritma kavramı ile eşdeğer olarak görülebilir. Kavram, literatürde çoğunlukla ilköğretim müfredatı odaklı görülse de (Faber vd., 2017; Kakavas ve Ugolini, 2019; Mukasheva ve Omirzakova, 2021; Camacho ve Grané, 2023) hayatın diğer alanlarında da uygulanabilirliği kaçınılmazdır.

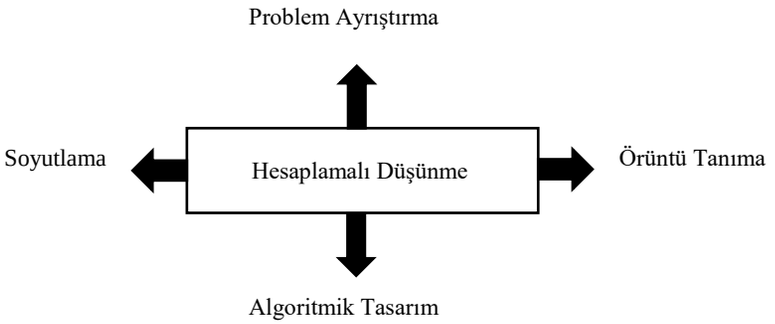
Hesaplamalı düşünme; Bar, Harrison ve Conery'ye göre (2011, s. 2) aşağıdaki adımları içeren bir problem çözme süreci olarak ele alınmıştır:

- Bilgisayar ve diğer araçları kullanarak sorunları çözmeye yardımcı olacak şekilde formüle etmek,
- Verileri mantıksal olarak organize ve analiz etmek,
- Verileri modeller ve simülasyonlar gibi soyutlamalar yoluyla temsil etmek,
- Algoritmik düşünme yoluyla çözümleri otomatikleştirmek,
- Adımların ve kaynakların en verimli ve etkili birleşimine ulaşmak amacıyla olası çözümleri belirlemek, analiz etmek ve uygulamak,
- Problem çözme sürecini çok çeşitli problemlere genellemek ve aktarmak.

Bu bağlamda hesaplamalı düşünme kavramının sadece bilim insanlarının veya teknoloji uzmanlarının bir kabiliyeti olarak değerlendirilemeyeceği açıktır. Daha önce de bahsedildiği üzere, bu kavram tüm insanlığın kabiliyetidir ve her alanda kullanılmaktadır. Kavrama dair yanlış bilinenler ve açıklamaları ise aşağıdaki gibi özetlenebilir:

- ***Hesaplamalı düşünme bir programlama değildir, kavramsallaştırmadır:*** Bilgisayar biliminin sadece bilgisayar programlama olmadığı açıktır. Bir bilgisayar bilimcisi gibi düşünmek, bir bilgisayarı programlayabilmekten daha fazlasını ifade eder.
- ***Hesaplamalı düşünme ezbere dayalı beceri değildir, temel beceridir:*** Temel beceri; her insanın modern toplumda iş yapabilmek için bilmesi gereken bir olgudur; düşünmek gibi. Ezber, mekanik bir rutin anlamına gelmektedir.
- ***Hesaplamalı düşünme bilgisayarların değil, insanların düşündüğü bir yoldur:*** Hesaplamalı düşünme, insanların bilgisayarları kullanarak sorunları çözmesinin bir yoludur. Kavram, insanların bilgisayarlar gibi düşünmesini sağlamaya çalışmamaktadır. Bilgisayarlar yalnızca kendilerine verilen komutları işler; bilinçli değildir. İnsanlar ise zekidir ve hayal kurabilir.
- ***Hesaplamalı düşünme matematik ve mühendislik düşüncesini tamamlar ve birleştirir:*** Gerçek dünyayla etkileşime giren sistemler oluşturulduğu göz önüne alındığında birçok alan, mühendislik düşüncesinden yararlanmaktadır. İnsanları sadece matematiksel olarak değil, hesaplamalı olarak düşünmeye zorlayan şey, temeldeki hesaplama cihazlarının kısıtlamalarıdır; aynı şekilde insanları fiziksel dünyanın ötesinde sistemler tasarlama konusunda özgür kılan şey ise, sanal dünyalar inşa etme yeteneğidir.
- ***Hesaplamalı düşünme bir ürün değil, fikirlerdir:*** Hesaplamalı düşünme somut bir ürün değil; günlük yaşamı yönlendiren, sorunlara yaklaşımı şekillendiren ve insan-insan ile insan-teknoloji etkileşimlerini etkileyen bir düşünme biçimidir. Gelecekte, sadece geliştirilen yazılım ve donanımlar değil, bu sistemlerin temelini oluşturan hesaplamalı düşünme ilkeleri de bireylerin yaşamlarına her düzeyde dokunmaya devam edecektir.
- ***Hesaplamalı düşünme herkes için, her yerde ve her zaman vardır:*** Hesaplamalı düşünme felsefe olarak içselleştirildiğinde, insan çabalarının ayrılmaz bir parçası olacak ve bir gerçeklik haline gelecektir (Wing, 2006, s. 3-4).

Hesaplamalı düşünmenin; bilgisayar bilimi veya programlama ile sınırlı olmadığını bilmek ve bunun farkına varmak önemlidir. Başka bir deyişle, programlamanın kendisiyle ilgili olmaktan çok, bir bilgisayarın izleyebileceği şekillerde düşünme yöntemleriyle ilgilidir. Hesaplamalı düşünme, denilebilir ki, problemin ifade edilmesinde ve bu probleme yönelik çözümlerin tasarlanmasında kesinlik ve netliğin önemini vurgulamaktadır. Kavramın, giderek dijitalleşen ve birbirine bağlanan bir dünyada yeniliği teşvik etmek ve bilinçli karar almayı kolaylaştırmak için bir temel görevi gördüğü rahatlıkla söylenebilir.



Şekil 3. Hesaplamalı Düşünmenin Dört Bileşeni

Şekil 3'te hesaplamalı düşünmenin dört bileşeni gösterilmektedir. Bu bileşenlerin açıklamaları aşağıda verilmiştir:

1. ***Problem Ayrıştırma:*** Karmaşık problemlerin daha küçük ve daha anlaşılabilir parçalara ayrılması olarak tanımlanabilir. Böylece yapılandırılmış etkili bir analiz ve çözüm yolu sağlanmış olur (Rich, Egan ve Ellsworth, 2019).
2. ***Örüntü Tanıma:*** Problemdeki benzerliklerin ve düzenliliklerin belirlenmesine olanak tanıyarak genelleştirilebilir çözümler sunar (Rosali ve Suryadi, 2021).
3. ***Algoritmik Tasarım:*** Problem çözmeye yönelik adım bazlı prosedürler oluşturma süreci olarak tanımanabilen algoritma, sistematik çözümlere olanak sağlar (Skiena, 2008).
4. ***Soyutlama:*** Bilgileri korurken gereksiz ayrıntıların filtrelenmesini sağlayan ve sorunları etkili bir şekilde ele alacak modellerin ve çerçevelerin oluşturulmasına yardımcı olan bir tekniktir (Ward, 1995). Başka bir deyişle kısıtlamaları kaldırma ve önemsiz bilgileri gizleme süreci de denilebilir.

Arşivcilik mesleği, diğer tüm meslekler gibi bilgi teknolojileri temelli uygulamalar ile birlikte çalışmaya doğru evrilmiştir. Alanda devam eden yenilikçi uygulamalar ile birlikte arşiv materyallerinin korunması, eski zamanlara nazaran daha fazla sağlanabilir hale gelmiştir. Bu duruma dijitalleşmenin etkisi ile gelindiğini söylemek yanlış olmaz. Arşiv materyallerinin metaveri standardizasyonu (Yalçınkaya, 2014), büyük veri analitiği bağlamında EBYS'lerin log analizleri (Yalçınkaya, Gedikli ve Cibaroğlu, 2019), yeni oluşan belge uzantıları, uzun süreli korumada yapay zekâ ve makine öğrenmesinden yararlanma vb. gibi yeni gelişen uygulamalar, alanın hesaplamalı tarafına dikkati çekmektedir.

2.6. Hesaplamalı Arşiv Bilimi ve Çalışmaları

Arşiv biliminin önündeki en önemli zorlukların; BİT'in hızlı bir şekilde gelişmesi sonucunda ortaya çıkan büyük veri kavramı ile ilişkili olduğu önceki bölümlerde belirtilmişti. Klasik

formattaki her türden materyali arşivsel süreçlerde yönetmek için bir yöntem ve kurallar bütününü içeren arşiv bilimi, büyük veri ortamındaki her türlü bilgiyi, belgeyi, içeriği ve hatta veriyi saklamak ve yönetmek için, bilgisayar destekli uygulamalara ihtiyaç duymaktadır. Çünkü çoğunlukla kâğıt tabanlı materyallere uygulanabilen klasik arşiv biliminin varsayımları, dijital ortamda bulunan veri, bilgi ve belgelerin farklı yapılara sahip olmasından dolayı yetersiz kalmaktadır. Bu nedenle, dijital arşivlerde yer alan materyallerin yönetimi için veri madenciliği, yapay zekâ ve makine öğrenmesi gibi modern teknolojilere dayalı yeni yaklaşımlara ihtiyaç duyulmaktadır. Arşiv bilimi, teorisyenler ve uygulamacılar tarafından elektronik ortamdaki veri ve bilgilerin yönetimi için yazılımsal süreçler bağlamında değişime zorlanmaktadır. Bu değişim, yeni geliştirilen çeşitli uygulamalarda görülebilir. Örneğin; Jeba vd. (2024), resmi belgelerin yönetiminde güvenli, şeffaf ve erişilebilir bir ekosistem sağlamak amacıyla blokzincir tabanlı eVault programı geliştirmiştir. Akıllı sözleşmelerin kullanıldığı program, yetkilendirme bazlı çalışmaktadır. Programın özellikleri arasında katı erişim kontrolleri, izin yönetimi ve işlem şeffaflığı ile resmi belgelerin bütünlüğü ve güvenilirliğinin sağlanması bulunmaktadır.

Tablo 6. Fiziksel ve Dijital Ortamdaki Veri Kavramının Farkları

Fiziksel Ortam	**Dijital Ortam**
Taşıyıcı ortama bağlıdır	Soyuttur ve biçimden bağımsızdır
İnsan merkezlidir	Makine tarafından okunabilir ve insan desteklidir
Durağandır ve sabit düzeni vardır	Dinamiktir, modüler ve kod odaklıdır
Genellikle bağlama bağlıdır	Açık izleme gerektirir
Elle işlenir	Otomatik işlenir, karmaşıktır ve geri döndürülebilir

Tablo 6'da fiziksel ortamdaki veri kavramı ile dijital ortamdaki veri kavramının farkları özetlenmiştir. Geleneksel formatta veri, taşıyıcı ortamından ayrılamaz özelliktedir (İçerik - metin, resim, el yazısı vb.- ve taşıyıcı ortama -örneğin; kağıt, film, parşömen vb.- fiziksel olarak yerleştirilmiştir). Bu taşıyıcı ortamda verinin düzeni, sırası ve anlamı genellikle biçime bağımlıdır. Bu anlamda, yorumlanması insan algısına bağlıdır. Yani veriler, bir kod çözme sistemine ihtiyaç duyulmadan doğrudan insanlar tarafından okunur ve anlaşılabilir. Dijital ortamda ise veri soyuttur ve ikili (0 ve 1) şeklinde kodlanmıştır; ayrıca okunabilmesi ve anlaşılabilmesi için çeşitli yazılım sistemleri gerekmektedir. Aynı veri, birçok cihaz ve platforma kopyalanabilir, taşınabilir ve bu platformlarda işlenebilir. Veri ayrıca dinamik olarak birbirine bağlanabilir.

Bu farklılıklardan hareketle, dijital ortamda üretilen belgelerin içerikleriyle birlikte saklanması ile fiziksel belgelerin korunması arasında yöntemsel ve kavramsal farklılıklar olduğu açıktır.

Bu durum, arşiv teorisinin günümüz dijital üretim ve erişim koşullarına uyum sağlayacak şekilde yeniden yapılandırılmasını gerektiren bir gerçeklik olarak karşımıza çıkmaktadır. Belge formatları ve taşıyıcıları dönüşürken, arşiv bilimi de bu dönüşüme paralel biçimde evrilmek zorundadır. Materyallerin üretildikten sonra kamuya erişime açılması, tarihsel olarak arşiv teorisinin temel önceliklerinden biri olmuştur. Bu bağlamda, erişim kavramı arşiv teorisinde merkezi bir yere sahiptir. Güvenlik veya gizlilik gerektirmediği hâlde, saklanan ve belirli ya da potansiyel değere sahip dijital bilgi ve belgelerin araştırmacıların ve kamunun erişimine sunulmaması, arşiv teorisinin erişim ilkeleriyle çelişmektedir. Dolayısıyla, dijital belgelerin ya da arşiv gruplarının (örneğin, dijitalleştirilen el yazmaları ya da haritalar) uygun biçimde düzenlenerek son kullanıcıya erişilebilir hâle getirilmesi, güncel arşiv uygulamaları açısından da bir gereklilik hâline gelmiştir.

Hesaplamalı arşiv bilimi disiplini, bu anlamda oldukça umut dolu bir gündeme sahiptir. Disiplin, salt bilgisayar bilimi ile arşiv biliminin birleşmesini içermemektedir. Bu yeni disiplin, bir disiplinlerüstü yaklaşıma sahip olarak her iki disiplinin de birbirlerini etkileyebileceği ve besleyebileceğini öne sürmektedir (Marciano, 2022). Disiplin aynı zamanda arşivcilik okullarına yeni müfredat önerileri de getirmektedir. Değişen ve gelişen dünyada nerdeyse her bilim dalında olduğu gibi, arşiv biliminde de BİT ile ilgili güncel dersler ve konular bu kapsamda müfredatlara eklenmektedir.

Maryland Üniversitesi Bilgi Çalışmaları Fakültesi'nden Prof. Dr. Richard Marciano ve Dr. William Underwood, British Columbia Üniversitesi Bilişim Fakültesi'nden[36] Prof. Dr. Victoria Lemieux ile King's College Londra Dijital İnsani Bilimler Fakültesi'nden Prof. Dr. Mark Hedges öncülüğünde oluşan bir çalışma grubu, Nisan 2016'da bir sempozyum düzenleyerek hesaplamalı arşiv bilimi kavramını yeni bir bilimsel alan olarak önermişler ve aynı yıl IEEE büyük veri konferansında ilk defa hesaplamalı arşiv bilimi çalıştayı düzenlemişlerdir. Bugüne kadar her yıl yapılan çalıştayların dokuzuncusu 17 Aralık 2024 Salı günü Washington'da yapılmıştır. Kavramın uzun bir tanımı bulunmaktadır:

"Değerlendirme, düzenleme ve tanımlama, koruma ve erişim kararlarını desteklemek için verimliliği, üretkenliği ve duyarlılığı artırmak amacıyla büyük ölçekli belgelerin/arşivlerin işlenmesi, analizi, depolanması, uzun süreli korunması ve erişimine hesaplamalı yöntemlerin ve kaynakların uygulanmasıyla ilgilenen disiplinlerüstü bir alan. Amaç, arşiv materyalleriyle araştırma yapmak ve bu araştırmaları yürütmek ve ayrıca bilgisayar bilimi ile arşiv biliminin

[36] iSchool

kolektif bilgisini, yeni teknolojilerin belgelerin oluşturulmasını, kullanımını, depolanmasını ve korunmasını nasıl değiştirdiğini ve bu değişikliklerin arşiv fonksiyonları ve otantik dijital belgelerin toplumsal ve örgütsel kullanımı ve korunması üzerindeki etkilerini anlamak için uygulamaktır. Bu da hesaplamalı arşiv biliminin hesaplamalı ve arşivsel düşünmenin bir karışımı olduğunu göstermektedir" (Marciano vd., 2018, s. 181).

Hesaplamalı arşiv biliminin odaklandığı temel konu; arşivlerin giderek daha dijital bir hale dönüşmesi neticesinde, arşiv koleksiyonlarındaki verilerin bağlamsallaştırılmasına hesaplamalı yöntemlerin uygulanmasıdır. Klasik anlayışta arşivsel düşünmenin temeli belge iken, hesaplamalı arşiv bilimi bağlamında arşivsel düşünmenin temeli veridir. Belge yönetimi ve arşivcilik ilkelerinin veri yönetimine uygulanması ile karşılıklı etkileşim[37] hedeflenmektedir. Bu yeni bilim alanı, bilgisayar bilimi ve arşiv bilimi disiplinlerinin teorik temellerinin geliştirilmesine katkıda bulunmak, tüm sektörleri ve alanları desteklemek için eğitim kurumlarını tasarlamak ve eğitim vermek, işbirliği ortamında bilimsel gelişmeleri test etmek ve uygulamak için sanal ve fiziksel bir laboratuvar geliştirmek amacını gütmektedir (Marciano vd., 2018). Başka bir deyişle alan, temel olarak ortaya çıkan yöntem ve teknolojilerin büyük veri etrafında birleştirilmesi ve sonuçlarının arşiv uygulamaları ve yeni analiz biçimleri ile arşivlerle tarihsel, sosyal, bilimsel ve kültürel araştırma ilişkisini ortaya koymaktadır. Bunun yanında büyük verinin yorumlanması ve kullanımı hakkında alanın gitgide artan sorunları ve endişeleri de ele alınmaktadır.

Carter vd. (2022), çalışmasında Roosevelt Kütüphanesi'ndeki Holokost dönemi arşiv materyallerine hesaplamalı arşiv bilimi ve yeni dijital metodolojileri uygulamıştır. Çalışmada veri seti olarak ABD eski Hazine Bakanı Henry Morgenthau'ya ait 12 yıllık günlük iş kayıtlarını ve notlarını içeren ve 864 ciltten (285.000 sayfa) oluşan bir koleksiyon seçilmiştir. Veri setinin mikrofilm formatı, 2014 yılında dijital ortama aktarılmıştır. Çalışma kapsamında dijitalleştirilen mikrofilm içerikleri PDF dosyalarına dönüştürülmüş; bu dosyalar, görsel analiz yapılabilmesi amacıyla da sayfa bazlı görüntülere dönüştürülmüştür. Çalışmada; Ulusal Arşivler (NARA)[38] idaresinde olan Franklin D. Roosevelt Başkanlık Kütüphanesi ve Müzesi'nin paydaşı Roosevelt Enstitüsü (RI)[39] tarafından finanse edilen bir proje olan "Morgenthau Holokost Koleksiyonları Projesi (MHCP)[40]" kapsamında, koleksiyonda gizli kalmış değerli veri ve bilgilerin çıkarılmasında tahmine dayalı modelleme uygulamak için

[37] Intertwine
[38] National Archives and Records Administration
[39] The Roosevelt Institute
[40] Morgenthau Holocaust Collections Project

makine öğrenmesi yöntemleri kullanılmıştır. Nesne algılama metodolojilerine dayalı iki aşamalı denetimli makine öğrenmesi algoritması kullanılarak 864 cildin tamamı için yeni bir ana dizin oluşturulmasını sağlamak üzere dosyalar ve metaverileri toplayabilmek için Ulusal Arşivler Katalog API[41]'si ve FRANKLIN[42] araştırma araçları veritabanı kullanılmıştır. Tarihsel konu dizini içeriği için belge nesne şeması veritabanları kullanılmış ve bu içerik NLP[43] ve NER[44] metodolojileri kullanılarak analiz edilmiştir. Koleksiyona, koleksiyonları "veri" olarak gören bir bakış açısıyla yaklaşılmış, böylece bu önemli birincil kaynağa erişimi artırmak ve hem dijital araştırmanın hem de Holokost'un daha iyi anlaşılması amaçlanmıştır.

Breen vd. (2023), çalışmasında bir müze arşivinde bulunan balık cinslerinin görüntülerinin otomatik olarak ana hatlarını çıkarmak ve bu görüntüleri segmentlere ayırmak için görüntü işleme tekniklerini kullanmıştır. Veri seti, Prairie Araştırma Enstitüsü bünyesindeki Illinois Doğa Tarihi Araştırması'ndan (INHS)[45] alınan 1.071 balık görüntüsünden oluşmaktadır. Veri seti 7 balık cinsinden (Notropis, Phenacobius, Gambusia, Cyprinus, Esox, Lepomis, Noturus) oluşmakta ve her cins 153 örnek içermektedir. Çalışmanın amaçları; müze arşivi görüntülerinden ana hatların çıkarılması için geliştirilmiş bir görüntü işleme tekniği geliştirmek; eliptik Fourier[46] tanımlayıcılarına dayalı ayırt edici bir örnek anahat gösterimini keşfetmek; hesaplamalı arşiv bilimi yaklaşımının etkinliğini ve balık örneklerinin cinsini yüksek doğrulukla tahmin edebilen örnek bir akış analizi ile göstermek şeklinde belirlenmiştir. Analizler için; 2 boyutlu görüntü ana hatlarını, makine öğrenmesi teknikleriyle analiz edilebilecek 1 boyutlu vektörlere dönüştüren eliptik Fourier tanımlayıcıları kullanılmıştır. Çalışmada, ilk olarak balık cinsi görüntülerinin otomatik metaverilerini oluşturabilmek için koleksiyon içindeki örnekler ve diğer ilgili nesneler tespit edilmiştir. Nesne tespiti için Facebook AI Research (FAIR) tarafından geliştirilen tespit aracı ve Mask R-CNN[47] mimarisi kullanılmıştır. Müze arşiv görüntülerinin ana hat oluşturma teknikleri, farklı cinslerden bir dizi balık görüntülerine uygulanmıştır. Uygulama sonucu ortaya çıkan özellik vektörleri, bir balığın cinsini 2 boyutlu taslağından tahmin etmek için bir sınıflandırma görevinde kullanılmıştır. Verilere; destek vektör makineleri, çok katmanlı algılayıcılar, xgboost ve k-en yakın komşu algoritması gibi sınıflandırma algoritmaları uygulanmış ve tüm yöntemler kabul edilebilir

[41] Application Programming Interface
[42] http://www.fdrlibrary.marist.edu/archives/collections/franklin/
[43] Natural Language Processing
[44] Named Entity Recognition
[45] Illinois Natural History Survey
[46] Bir görüntüdeki konturları ve silüetleri ölçmek için konturlar üzerinde elipslerin toplamını kullanan matematiksel bir tekniktir. Bu teknik bilgisayar görüntüsü ve görsel desen eşleştirmede yaygın olarak kullanılır.
[47] Mask Region-based Convolutional Neural Network

sonuçlar üretirken, SVM'nin[48] en yüksek doğrulukla tahminler ürettiği görülmüştür. Çalışmanın genel tahmin doğruluğu ise %96,3 ± 1,5 olarak açıklanmıştır.

Lee vd. (2017), çalışmasında "Büyük Verinin İnsan Yüzü Projesi[49]" kapsamında ABD'nin güneydoğusunda bulunan ve kentsel dönüşümden etkilenen en büyük alan olan East Riverside semtinden (Southside) gelen belgelere odaklanmaktadır. Belgeler halen Kuzey Karolina Üniversitesi D.H. Ramsey Kütüphanesi özel koleksiyonlarında ve üniversite arşivlerinde saklanmaktadır. Koleksiyon, Asheville Şehri Konut İdaresi'nin (HACA)[50] belgelerini içermekte ve yaklaşık 39,6 metre uzunluğunda 129 kutu belgeden oluşmaktadır. Veri seti, Southside semtinde kentsel dönüşüm süreçlerinin parçası olan yaklaşık 1000 mülkün satın alma sürecinin resmi belgelerini içermektedir. Her bir vaka dosyası 20 ile 200 sayfa arasında hacime sahiptir ve her bir mülkün tüm satış sürecini kapsamaktadır. Çalışmada ilk olarak tapularda bulunan parsel verileri, dijital bir harita üzerinde işaretlenmiştir. Bu dijital haritaya dayalı olarak, kullanıcı arayüzlerini ve veritabanlarını tasarlamak için senaryo tabanlı yöntem kullanılmıştır. Harita üzerinde coğrafi referanslama için ArcMap ve Photoshop uygulamaları kullanılmıştır. Ardından mülk sahipleri belirlenmiş ve senaryo bağlamında anonim olarak hem harita üzerine hem de veritabanına kaydedilmiştir. Asheville haritasında, dijital kürasyon süreçlerinden sonra satın alma süreçlerinden geçen parsellerin yaşam döngüsü "zaman çubuğu" ile interaktif olarak görülebilmektedir. Proje, yeni bir teorik temel oluşturabilecek ileri düzey hesaplamalı düşünme süreci olarak değerlendirilmektedir.

Hamouda vd. (2019), sahte videoları tespit etmeye ve sınıflandırmaya yardımcı olmak için arşiv biliminin temel varsayımlarından olan güvenilirlik faktörünü kullanarak yeni bir çerçeve tasarımı önermiştir. Başka bir deyişle, hesaplamalı arşiv bilimi aracılığıyla sahte videolar sorununu çözmeye yönelik yeni bir yaklaşım sunmuştur. Çalışmada ilk olarak sahte videoları tespit etmek için, videolarda bir güvenilmezlik özelliği oluşturmaya ve sahte bir tipolojiyi genellemeye yönelik çalışma gerçekleştirilmiştir. Çalışma ekibi, daha sonra sahte bir videoyu tespit etmek için kullanılabilecek altı benzersiz test kategorisi oluşturmuştur. Testler, videoların üç temel bileşenini (görsel, ses ve metaveri) içermektedir. Videoyu doğrulayabilmek için, her biri iki adımdan oluşan (iç tutarlılık ve dış tutarlılık) iki test yapılmıştır. Görsel testlerde; videonun gerçekliği (yani kimliği ve bütünlüğü) değerlendirilmiştir. Eğer videoda görsel tutarsızlıklar tespit edilirse, bu, videonun kimliğinin tahrif edildiğini gösterir çünkü bu durumda videonun bazı dış unsurları (örneğin; videonun görsel görüntüleri veya kare hızı)

[48] Support Vector Machine
[49] The Case of the Human Face of Big Data Project
[50] The Housing Authority of the City of Asheville

değiştirilmiş demektir. Bu tür görüntü tutarsızlıkları, video oluşturulduğunda iletilmesi amaçlanan mesajın anlamını değiştirirse, videonun bütünlüğü olumsuz etkilenir. Aynı durum ses testleri için de geçerlidir. Videoda ses ve görsel bileşenler arasında tutarsızlıklar tespit edilirse, bu, videonun bazı dış unsurlarının (örneğin; görüntü, videonun kare hızı, ses hızı) değiştirilmiş olduğu anlamına gelir. Metaveri testi de videonun gerçekliğinin değerlendirilmesine yardımcı olmaktadır. Metaveriler ve görsel bileşenler arasında tutarsızlıklar tespit edilirse, bu, videonun kimliğinin tahrif edildiğini gösterir, çünkü bu durumda da videonun benzersiz öğeleri (örneğin; başlık, açıklama, konum veya tarih gibi metaveriler) değiştirilmiş demektir. Çalışmada yaklaşım; videoların güvenilirliğine yönelik daha düşük bir güveni işaret edebilecek veya olası bir sahtekârlığı gösterebilecek tutarsızlıkları belirlemek için videonun kendi içindeki bileşenlerinin ikili karşılaştırmalarından oluşan bir dizi test önermektedir.

Randby ve Marciano (2020), çalışmasında derin metaveri çıkarımı otomasyonuna olanak sağlayan bir tahmin modeli geliştirmek için denetimli öğrenme algoritmaları ile deneyler gerçekleştirmiştir. Çalışmada veri setinin 1. serisi, Morgenthau'ya ait günlük iş kayıtlarını kapsamakta ve çoğunlukla kronolojik sıraya göre düzenlenmiş 864 cilt günlüğe karşılık gelen 881 PDF dosyasından oluşmaktadır. 3. seri ise alfabetik konulara göre düzenlenmiş 65 PDF dosyasından oluşmakta ve 30.000'den fazla konu dizini kartına karşılık gelmektedir. Ciltlerin her biri içindekiler tablosu içermektedir. İlk olarak, 30.000'den fazla konu dizini işlenmiş ve OpenRefine yazılımı kullanılarak toplamda 6361 benzersiz konu başlığı çıkarılmıştır. Konu dizinlerini tanımlamak için ilk adım olarak nesne algılama algoritmaları kullanılmıştır. Çalışmada makine öğrenmesi stratejisi Google AutoML Vision ile Nanonets[51] derin öğrenme ve OCR'dan[52] oluşan 2 aşamalı bir yaklaşımından oluşmuştur. Nanonets; görüntü verilerinde 6 benzersiz etiketten (kitap, sayfa, başlık, içerik, tarih ve el yazısı) birine karşılık gelen nesneleri bulmak için eğitilmiştir. 2 aşamalı makine öğrenmesi sürecinde toplam 3579 "içindekiler sayfası" JPEG formatında işlenmiş ve toplam 22.667 konu dizini kutusu keşfedilmiştir. Tüm bu etiketli verilerin bir veritabanı ayrıca oluşturulmuştur. Model eğitilmiş ve %96,5 duyarlılık skoru elde edilmiştir.

Marciano, Jansen ve Underwood (2020), çalışmasında hesaplamalı düşünme kavram ve uygulamalarının kütüphane ve arşiv bilimi eğitiminde yüksek lisans düzeyinde müfredata nasıl dahil edilebileceğini araştırmıştır. Bunun için dijital belgeler üzerinde arşiv süreçlerini

[51] Belge işleme ve veri çıkarma iş akışlarını otomatikleştiren yapay zekâ destekli bir sistem.
[52] Optical Character Recognition

gerçekleştirebilmek için yapılanları kaydeden açık kaynak kodlu ve bulut tabanlı Jupyter Notebook uygulaması ve bunlara erişim sağlamak için CASES[53] adlı bir çevrimiçi havuz oluşturulmuştur. Burada amaç; hesaplamalı örnek olay çalışmaları ve ders planları oluşturup yayarak birbirlerinden öğrenen eğitimciler ve uygulayıcılardan oluşan işbirliğine dayalı bir ağ oluşturmaktır. CASES web platformu, arşivcilerin ve araştırmacıların dijital arşiv materyallerine giderek daha fazla uyguladıkları hesaplama araçlarını sergilemek, paylaşmak ve öğretmek için bir hizmet olarak sunulmaktadır. Çalışmada; en iyi uygulamaların sınıf ve laboratuvar ortamlarında benimsendiğini göstermek amaçlanmış ve başarı da Jupyter Notebook hesaplamalı not defterinin izlenmesi ve kullanılması yoluyla ölçülmüştür. Jupyter Notebook'un, dijital belge ve veriler üzerinde arşiv görevlerini gerçekleştirmek için oldukça yararlı bir araç olduğu da görülmüştür.

Lin (2017), Tayvan Yerli Halkları Açık Araştırma Verileri adı verilen ve 2013-2017 yıllarındaki güncel nüfus sayımı ve hane halkı kayıt veri setlerine dayanan büyük bir arşiv verisinin oluşturulmasına yönelik demografik özellikleri içeren bir çalışma gerçekleştirmiştir. Bu kapsamda tekil dijital belgelerin toplanması, temizlenmesi, işlenmesi, keşfedilmesi ve zenginleştirilmesinde hesaplamalı arşiv bilimi çerçevesi ve veri bilimi kullanılmıştır. Veri seti; kategorik veriler, demografik özellik verileri (hane yapısı ve özellikleri) ile zaman serisi verileri olmak üzere üç kategoriden oluşmaktadır. Demografik veriler ise; bağlantılı nüfus dinamikleri verileri, yaşam tabloları, hane istatistikleri, soyağacı verileri, evlilik ve etnik kimlik verileri, iç göç verileri ile coğrafi kodlanmış verilerden oluşmaktadır. Çalışmada, iki adet üst düzey Xeon 2680 v2 CPU ve 256 GB DDR3-16000 ECC RAM'e sahip yüksek performanslı bir iş istasyonu ve I/O veri yolu ile dahili bellek bilgi aktarım hızını ayarlamaya olanak tanıyan BIOS[54] kullanılmıştır. Programlama dili olarak Pascal üzerinde hesaplama kodları geliştirmiş ve kodlar Embarcadero RAD Studio v.10.1 derleyicisi ile derlenmiştir. Yapılan analizler sonucunda 210.000 yerli haneden %97,8'inin yeri tespit edilebilmiştir. Kalan kısım ise, başarılı bir şekilde yeri tespit edilen komşu yerli hanelerin bilgileri kullanılarak tahmin edilmiştir. Çalışma; yerli haneler (sayıları yaklaşık 210.000) ve yerli bireyler (toplamda yaklaşık 560.000 kişi) olmak üzere 2 grupta tamamlanmıştır. Yerli halkların tespit edilen nüfus merkezleri ise dört bölgesel yerli nüfus merkezi, 25 il ve şehir nüfus merkezi (ilçe düzeyine eşdeğer), 365 ilçe yerli nüfus merkezi ile 8700 köy yerli nüfus merkezi olmak üzere toplam 4 bölgedir. Ayrıca, 486 yerli kabile ve topluluk etnik gruplara göre tespit edilmiştir. Çalışma ayrıca, yerli halkın

[53] Computational Archival Science Educational System
[54] Basic Input/Output System

bireysel etnik kimlik oluşumunun %97'sini tanımlanmıştır. Etnik kimliğin %50,6'sı tek etnik kimlikten, %19,4'ü babasoylu etnik kimlikten ve %27,0'ı ise anasoylu etnik kimlikten kaynaklanmaktadır. Çalışmanın başlıca katkıları ise; Tayvan yerli halklarının görünür hale gelmesinin sağlanması; açık veri setlerinin yerli nüfus çalışmalarıyla ilgilenen araştırmacılar için analiz engellerini azaltması; araştırmanın uluslararası akademik topluluğa açılması; araştırmanın gizli ve sabit kalmış verilerinin "mikro ve dinamik" verilere dönüşmesidir.

Feng vd. (2021), çalışmasında Çin kökenli ve uluslararası bilim insanlarının görüşlerini çevrimiçi çalıştay düzenleyerek 5 yıl içinde arşiv alanında hangi konuların trend çalışmaları kapsayacağını belirlemeye çalışmıştır. Toplamda 16 uluslararası ve 15 Çin kökenli uzmandan çevrimiçi mülakat yoluyla alınan cevaplarda; hesaplamalı arşiv bilimi ve yenilikçi teknolojilerin arşivlere uygulanması ön plana çıkmıştır. Çalışma kapsamında görüşülen bilim insanları; belge yöneticileri ve arşivcilerin dijital belgelerin bolluğu ve karmaşıklığı ile başa çıkmak için manuel yaklaşımları uygulamada büyük zorluk yaşadıkları görüşünde bulunmuş; bu nedenle kritik arşiv zorluklarını ele almak için yenilikçi teknolojiler uygulanması gerektiğinden bahsetmişlerdir. Bu konuya ek olarak arşiv biliminin geleceğinin sadece teknolojilere ve teknoloji destekli araçlara dayanamayacağını, profesyonel mesleki kimliğin korunmasının oldukça önemli olduğu da belirtilmiştir. Diğer trend olabilecek konular arasında; algoritmaların hesap verebilirliği ve uçtan uca elektronik belge yönetimi ön plana çıkmaktadır. Tablo 7'de, trend olabilecek konular gösterilmektedir.

Tablo 7. Arşiv Uzmanları Tarafından Önerilen Araştırma Alanları

Çinli Arşiv Uzmanlarının Görüşleri	**Uluslararası Arşiv Uzmanlarının Görüşleri**
Hesaplamalı arşiv bilimi	Hesaplamalı arşiv bilimi
Büyük veri, yapay zekâ, blokzincir, bulut bilişim gibi yenilikçi teknolojilerin arşivlere yönelik uygulamaları	Yapay zekâ veya diğer yenilikçi teknolojilerin arşivlere yönelik uygulamaları ve araç geliştirmeleri
Yeni teknoloji ortamında uçtan uca elektronik belge yönetimi	Yapay zekâ veya diğer teknolojilerin uygulanmasında etik
Dijital arşivler	Algoritmik sorumluluk
Çin arşiv düşüncesinin tarihi ve teorisinin gelişimi	Dijital süreklilik
Diplomatik	Dijital paradigma
Arşiv yönetimi	Arşivlerin tarihi ve arşiv disiplini
Arşivler ve dijital insani bilimler	Verilendirilmiş ve dağıtılmış ortamlar için arşiv diplomatikleri
Arşivler ve hafıza	Arşivler ve zorluklar
Arşivcilik eğitimi	Arşivsel çoklu evrenin uygulanması
	Belge ve bilgi yönetişimi
	Katılımcı arşiv paradigması
	Kişisel arşivler

Kaynak: (Feng vd., 2021, s. 402)

Giannini ve Bowen (2022) çalışmasında; hem karmaşık yapıda hem de yeni, kapsayıcı ve çeşitli kültürel dokulara entegre olan ve "hesaplamalı" temelde yükselen bir arşiv sistemi

öngörmektedir. Bu öngörüsü temel olarak; dijital ve fiziksel sistemlerin artık iç içe geçtiği; insanın varoluş, kimlik ve kültürel boyutlarını bütünleştirdiği bir yapıyı referans almaktadır. "Hesaplamalı" yaşamın, arşivlere yönelik mevcut baskın kültürü giderek daha fazla tanımlayan dijital dünyaya katılması için acil bir ihtiyaca yol açtığını da öne sürmektedir. Bu durum, arşivlere olan "geleneksel" bakış yerine, 21. yüzyılda yeniden tasarlanması gereken "modern" arşivi işaret etmektedir. Geçmiş yaşamı belgeleyen arşiv materyallerini; yapay zekâ, makine öğrenmesi, sinir ağları ve görüntü tanıma vb. teknikler tarafından yönlendirilen kapsayıcı bir "hesaplamalı" sisteme bağlayan ve arşivleri insan merkezli bir yapıya yöneltecek olan yeni bir "arşiv çerçevesi" tasarlanması gerektiğini de vurgulamıştır. Böylece kullanıcılar geçmişte, günümüzde ve gelecekte arşiv materyallerinin içeriği ve bağlamı hakkında yeni ve güncel içgörüler elde edebileceklerdir.

Payne (2018), çalışmasında; özellikle son 10 yıldaki dijital dönüşüm ile büyük veri teknolojilerindeki ilerlemelerin; bilgisayar bilimciler, bilgi bilimciler ve arşiv bilimciler tarafından yapılan çalışmalar üzerinde önemli bir etkiye sahip olduğundan hareketle, bu üç disiplini hesaplamalı arşiv bilimindeki boşlukları doldurmak için karşılaştırmıştır. Öncelikle, bu üç disiplinin odak noktalarını tablolaştırarak bir analiz yapmıştır.

Tablo 8. Çok Disiplinli Perspektif Bağlamında Disiplinlerin Odakları

	Disiplinler		
	Arşiv Bilimi	**Bilgi Bilimi**	**Bilgisayar Bilimi**
Yapılması Gereken	Belgelerin zaman içinde nasıl kalıcı hale getirildiğini anlamak	Bilgi ve teknoloji etkileşiminin insani yönünü anlamak	Hesaplama teorisini ve bir sistemin optimal tasarımını anlamak
Yapılmak İstenen	Belgeye odaklanmak ve onu korumak	Bilgi ağlarının sosyo-teknik yapısını anlamak	En iyi tasarım ve hesaplama uygulamalarını mümkün kılan anlayışı problemlere uygulamak
Yapılabilen	Bir eylemle ilgili gerçekleri anlamak	Çok çeşitli ortamlarda etkileşimde bulunan insanların bilgi davranışlarını anlamak	Algoritmaların, sistemlerin ve ağların fizibilitesini, yapısını, ifadesini ve mekanizasyonunu anlamak
Bilinebilen	Orijinallik ve doğruluk	İnsanların bilgiyi nasıl işlediği, deneyimlerini ve etkileşimlerini nasıl optimize ettiği	Bilginin en etkili şekilde nasıl elde edileceği, temsil edileceği, işleneceği, saklanacağı, iletileceği ve erişileceği

Kaynak: (Payne, 2018, s. 4)

Tablo 8'de; disiplinlerin birbirlerine kıyasla odak, hedef ve yaklaşımlardaki farklılıklar açık bir şekilde görülmektedir. Bu farklılıklar, hesaplamalı arşiv biliminin kendi potası içerisinde eritilip özümsenmesi gereken çeşitli araştırma alanlarının (büyük veri vb.) çoğu ile ilgilidir. Hesaplamalı arşiv bilimi; uzun süreli koruma, düzenleme ve tanımlama; erişimin nasıl sağlanacağına ilişkin kararlar ve hesaplamalı yöntemlerin ve kaynakların uygulanması ile hesaplamalı sistem tasarımının anlaşılmasına odaklanmaktadır. Çoğunlukla uygulamalı bakış

açısından, özellikle dijital arşiv verisine (Roke ve Tillman, 2022), her türden veri setlerinin makine öğrenmesi ve derin öğrenme bazlı analizlerine (Hutchinson, 2018; Hutchinson, 2020; Meurers vd., 2021) ve metin madenciliğine (Zucco vd., 2019; Choi vd., 2022; Yang vd., 2022; Ellahi vd., 2023; Gedikli, 2023) yönelik çok sayıda çalışma alanının halihazırda hesaplamalı arşiv bilimi bağlamında oluşturulduğu da görülmektedir.

Dickerson ve Yun (2023) çalışmasında; eleştirel ve toplum merkezli hesaplamalı arşiv bilimine olan ihtiyacın her zamankinden daha fazla olduğunu öne sürerek, arşivsel verilerin oldukça büyük ölçeği nedeniyle, dijital veri ve belgelerin yanlış bir "belgesel kapsam"da değerlendirildiğini vurgulamaktadır. Üretken yapay zeka ve dil öğrenme modellerinin gücüne atıfta bulunarak yazarlar; özselleştirme, ötekileştirme ve yok etme eğilimlerine direnme ve toplum merkezli arşiv ilkelerinin mutlaka hesaplamalı arşiv bilimine dahil edilmesi gerektiğini vurgulamaktadır. Ayrıca araştırmanın bulguları, büyük ölçekli dijital topluluk arşivlerinde yeterince temsil edilmeyen bireylerin ve toplulukların seslerinin duyurulması ihtiyacına ilişkin kanıtların, hesaplamalı düşünme ve hesaplamalı arşiv bilimi aracılığıyla toplanabileceğini ve erişime açılabileceğini ortaya koymaktadır.

Hesaplamalı arşiv bilimine farklı yaklaşımlar da bulunmaktadır. Ambacher ve Conrad (2021) hesaplamalı arşiv bilimindeki bazı uygulamaların (veri oluşturma ve verileri değiştirme gibi), geleneksel arşiv uygulamalarıyla pek uyumlu olmadığını vurgulayarak; bu tür işlemlerin arşivcilerin tipik olarak yaptığı bir şey olmadığını ve arşivlerin bir üreticiden aldıkları verilerle uyumlu olarak destekleyici ve açıklayıcı bilgiler oluşturduğunu; ancak arşivcilerin; alınan ve uzun süreli saklanan verileri, orijinal verilere bağlı izlenemeyecek şekilde manipüle edemeyeceğini veya değiştiremeyeceğini savunmaktadır. Böyle bir değişikliğin, arşivin temel görevi olan kanıtsal bilgilerin gerçekliğini sorgulayabileceği uyarısında bulunmaktadır. Yazarlara göre arşivciler; araştırmacıların, verileri kendi amaçları için bulmalarına ve yeniden kullanmalarına yardımcı olmakla görevlidir. Arşivciler; arşivsel bilgilerin muhtemel ikincil kullanımları veya bir çalışma verileriyle ilgilenebilecek topluluklar hakkında sadece tavsiyelerde bulunabilir ve gerektiğinde bilgi işlem uzmanları ile işbirliği yapar. Stančić (2018) ise büyük veri kavramının; verilerin, bilgilerin veya belgelerin hacimlerinden çok, nasıl kullanıldıklarıyla ilgili olduğunu öne sürerek; dijital veri, bilgi veya belgelerin doğru bir şekilde yorumlanması ya da kullanıcılar tarafından yorumlanabilir hale getirilmesi gerektiğini vurgulamıştır. Bunun anlamı ise; bir dijital bilgi veya belgenin, sadece tanımlanması veya düzenlenmesinin yeterli olmadığı, aynı zamanda veri bazlı analizler ile önemli bilgilerin tanımlanıp etiketlenerek, anlamsal ve bağlamsal bir zenginliğe kavuşturulmasıdır. Böyle bir

durum ise, hesaplamalı arşiv biliminin uygulamalarında gizlidir. Yazar, "hesaplamalı arşiv bilimi" teriminin kalıcı olup olmayacağının henüz bilinemeyeceğini de öne sürerek; bu alanın "yeni bir bilimsel disiplin mi" yoksa "sadece BİT'e ait araç ve yöntemlerin arşiv biliminin sürekli genişleyen yönlerine uygulanması mı" olduğuna yönelik bir soru sormaktadır. Bu soruya kendi cevabı ise "BİT'in arşiv bilimine uygulandığı"dır. Yazar, hesaplamalı arşiv bilimini bir yeni disiplin olarak görmemekte ve terimin bir süre daha kalıcı olabileceğinden bahsederek; arşiv biliminin desteklenmesine ve dijital materyallerin daha kolay kullanılabilir hale getirilmesine de yardımcı olabileceğini belirtmektedir.

Thibodeau (2018) ise daha farklı bir kavram türeterek "arşiv mühendisliği"ne giden yolu açıklamaktadır. Ona göre arşiv biliminin değeri, hem ürün hem de hizmet olarak arşiv kaynaklarının, kullanıcılarına sağladığı faydalarla anlaşılabilir. Yazar, arşiv mühendisliğinin arşiv biliminden ayrılabileceğini öne sürerek; kavramı, *"arşivsel ürün ve hizmetlerin sağlanmasında en optimal değeri sağlamak için arşiv bilimini sistematik olarak uygulayan bir disiplin"* şeklinde tanımlamıştır. Arşivsel süreçlerin çıktıları olan "ürünler" hem arşivlerde korunan materyallerden hem de bunlara ilişkin bilgilerden oluşmaktadır. Arşivsel süreçlerdeki kilit noktalardan biri olan "bağlam", bu durumda; hedef geçmişi açıklığa kavuşturan bir dizi tutarlı iddia olarak tanımlanabilir. Bağlamsal tanımlama ise; varlığını veya özelliklerini açıklığa kavuşturan bir dizi koşulla ilişkili olarak değerlendirilen bir nesne ve bir veya daha fazla bağlamsal unsuru içermektedir. Arşiv mühendisliği; arşiv bilimindeki kavramları temel alıp, bunları daha geniş bir kapsama genişleterek, dijital veri ve bilginin daha önce görülememiş yönlerini kapsayacak şekilde uyarlayabilir, otomatik işlemeyi kolaylaştırabilir ve nicel testler yoluyla doğrulamayı mümkün kılarak hedef geçmişlerin oluşturulmasını ve sonuçların değerlendirilmesini iyileştirme potansiyeli sunmaktadır.

Yapılan çalışmalar incelendiğinde, hesaplamalı arşiv bilimi uygulamalarının çok yönlü olduğu rahatlıkla görülebilir. Çoğunlukla makine öğrenmesi ve derin öğrenme gibi yapay zekâ uygulamaları temelinde; Twitter[55] paylaşımlarından duygu çıkarımı, web arşivleme metaverileri, dijital doğan ve sonradan dijitalleştirilen arşiv ve müze koleksiyonlarının örüntü analizleri, elektronik belgelerin otomatik sınıflandırılması, bilimsel veri setlerindeki gizli ilişkiler analizi, istatistiksel süreçlerin otomasyonu, arşivsel süreçlerin (değerlendirme, tanımlama, erişim, imha vb.) otomatikleştirilmesi ve daha birçok alanında çalışmalar gerçekleştirilmiştir. Arşiv biliminin BİT uygulamalarıyla entegrasyonu ilk paradigma değişimi

[55] X

olarak ele alınabilirken, yapay zeka ile birleşimi ikinci paradigma değişimi olarak nitelendirilmelidir.

Tablo 9. Bazı Hesaplamalı Arşiv Bilimi Projeleri

Proje Adı	Yıl	Amaç	Kaynak
Linked Data at the Swiss Federal Archives	2021	Arşiv bilgi sistemlerinden tanımlayıcı metaverileri, bağlantılı veri kümelerine dönüştürmek ve birleştirmek. Bunu yaparken, mevcut metaverileri bağlantılı veriye dönüştürmek için "en iyi uygulamaları" formüle etmek. Bağlantılı metaverilerin üçüncü taraflarca yeniden kullanılma potansiyelini göstermek.	(Cochard, 2021)
Piloting Network	2018	Jupyter Notebook etkileşimli hesaplamalı öğrenme platformu aracılığıyla hesaplamalı vaka çalışmaları ve ders planlarının paylaşılmasını ve yayılmasını sağlamak ve eğitimciler ve uygulayıcılardan oluşan çevrimiçi ulusal bir iş birliği ağı geliştirmek.	(Computational Archival Science Research Group, 2019)
CASES (Computational Archival Science Educational System)	2019	Arşivcilerin ve araştırmacıların dijital arşiv materyallerine uyguladığı hesaplamalı uygulamaları sergilemek, paylaşmak ve öğretmek.	https://cases.umd.edu/
Talent Network	2022	Jupyter Notebook kullanarak müfredat geliştirme ve değerlendirme yapmak. Hesaplamalı ve algoritmik düşünmeden kaynaklanan sosyal ve etik kaygıları ele almak. Bilgisayar bilimi, mühendislik, veri bilimi ve eğitim bilimleri ile Kütüphane ve Bilgi Bilimi bölümlerini yakınlaştırarak dijital ve hesaplama becerilerine sahip arşivciler ve kütüphaneciler için önemli fırsatlar sağlamak.	https://ai-collaboratory.net/projects/talent-network/
Machine Learning (ML) for FDR Presidential Library Collections (ML-FDR)	2020	Amerika Birleşik Devletleri'nin Holokost'a verdiği tepkileri inceleyen akademisyenler tarafından yeterince kullanılamayan ve FDR Başkanlık Kütüphanesi'nde bulunan 2. Dünya Savaşı dönemi belgeleriyle ilgili ulaşılması zor arşivsel içeriğin ortaya çıkarılması. Projede makine öğrenmesi teknikleri kullanılarak dijital kültürel varlıkların hesaplamalı işlemleri yapılmıştır.	https://ischool.umd.edu/projects/machine-learning-strategies-for-fdr-presidential-library-collections-ml-fdr/
InterPares	1999	Dijital belgelerin uzun vadeli korunmasını ve güvenilirliğini sağlamak. Otantikliği[56] korumak, standartlar geliştirmek, disiplinlerarası işbirliğini desteklemektir. Proje; statik ve dinamik dijital belgeler, bulut tabanlı ve yapay zeka tarafından yönetilen belgeleri de kapsamaktadır.	http://interparestrust.org/

[56] Özgünlük

Tablo 9'da ise hesaplamalı arşiv bilimi kapsamında değerlendirilebilecek uluslararası projelerden örnekler gösterilmektedir.

2.7. Modern Arşivciliğin Yeni Paradigması Olarak Hesaplamalı Arşiv Bilimi

Buraya kadar ele alınan bölümler; arşivlerin tarihsel gelişimini, dönüşüm süreçlerini ve modern yönelimlerini kapsamlı bir şekilde açıklamaktadır. Sosyal ve insani bilimlerden farklı olarak arşiv bilimi, teknolojik gelişmelere daha duyarlı olduğundan kabuk değiştirmektedir. Bu değişim; dijital veri ve bilgilerin doğruluğu ile bilgiye en uygun koşullarda erişim vb. olgular açısından titizlikle ele alınması gereken yeni bir boyut sunmaktadır. Arşiv biliminin bu dönüşümünü doğru şekilde değerlendirebilmenin ilk koşulu, bilim dallarının artık yalnızca kendi iç dinamikleriyle yetinemeyeceğini kabul etmek ve disiplinlerarası entegrasyon ile inovasyonun gerekliliğini anlamaktır. Örneğin, günümüzde bilgisayar bilimi metodolojileri ve hesaplama araçları olmadan arşiv bilimini düşünmek, bu alanı yalnızlaştırır ve gelişiminin önünü tıkar. Alanda, metodolojinin son 30 yılda giderek artan bir şekilde bilgisayar destekli uygulamalar çerçevesinde şekillendiği açıkça gözlemlenmektedir. Bu alandaki dönüşüm süreci, birçok çalışmada farklı perspektiflerden ele alınmış; özellikle dijitalleştirme, bilişim entegrasyonu ve hesaplama temelli uygulamalar gibi konular bağlamında analiz edilmiştir (Delmas, 1993; Zakharov, 1998; Duranti, 2001; Delmas, 2001; Rayward, 2002; Runardotter, 2007; Cox, 2007; Bunn, 2016; Yao, 2016; Li, 2022; Yancheva, 2024; Goynov vd., 2024).

Örneğin; arşivlerdeki tanımlama faaliyetlerinde, belgeler arasında ilişki kurmak ve bu ilişki üzerinden belge içeriğini anlamak amaçlanmaktadır. Bu süreç, belge bağlamı yanında tarihsel veya semantik[57] düzeyde gerçekleşen süreç örüntüsünün parçaları olarak nasıl bütünleştiklerini ortaya koymaktadır. Tanımlama faaliyetindeki temel hedef, belgelerin birbirleriyle bağlantısının kurularak daha kapsayıcı bir bağlam oluşturulmasını sağlamaktır. Bu, arşivcilik açısından sadece belgelerin fiziksel saklanması değil, bilginin erişilebilir ve anlamlı kılınması için de bir çerçeve oluşturmaktadır. Bu düşünce, yapay zekâ uygulamaları ile de önemli bir paralellik taşımaktadır. Özellikle bilgi grafikleri, ontolojiler veya semantik ağlar vb. yapılar kullanıldığında, yapay zekânın amacı da genellikle parçadan (arşiv örneğinde belge) bütüne doğru bir anlam örgüsü oluşturmaktır. Arşiv bilimi uygulamaları, giderek artan bir şekilde veri bilimi temelli olarak düşünülmekte (Akmon, Zimmerman ve Daniels, 2011; Colavizza vd, 2021; Shinde vd., 2024), özellikle veri provenansı (Payne, 2018; Davet, Hamidzadeh ve Franks, 2023; Yeo, 2024) ve otomatik analiz süreçlerinden faydalanan bir hesaplama

[57] Anlam bilimi

paradigmasına doğru evrilmektedir. Başka bir deyişle arşivsel yapılar; durağan (statik) bir dijital depolama özelliğinden aktif veri modellemesi, anlamsal zenginleştirme ve uzun vadeli izlenebilirliğe doğru genişlemiştir.

Arşiv materyallerinde saklı duran bu örüntüler, yapay zekânın örüntüleri tanıyabilme yeteneği ile arşiv süreçlerini daha verimli ve etkili hale getirilebilir. Örneğin, bir yapay zekâ modeli belgeler arasındaki tarihsel ilişkileri analiz edebilir, semantik bağlamları çıkarabilir ve bu bağlamda yeni ilişkilendirme önerileri sunabilir. Bu süreç, arşivcilerin hem iş yükünü azaltır hem de ilişkileri daha hızlı ve daha büyük bir doğruluk oranıyla belirleyebilir.

Arşiv bilimi ile yapay zekâ uygulamalarındaki bağlamsal ilişki oluşturma süreçleri arasındaki benzerlik, hem her iki alanın da ortak bir işbirliğini hem de bilginin yapılandırılması ve anlamlandırılması konusundaki ihtiyacı açıklamaktadır. Özetle, her iki alanın da nihai hedefi parçaların bütünleştirilerek daha büyük bir anlam kazanması ve bu anlamın kullanıcılara erişilebilir hale getirilmesidir.

Dolaysıyla son teknolojiler, bu bağlamda sadece birer "araç" olarak değerlendirilmemeli, aksine bilimsel yöntemlerin ve veri işleme süreçlerinin dönüşümünü sağlayan temel bileşenler olarak ele alınmalıdır. Dijital arşivlerin yönetimi yalnızca belgelerin dijital ortama aktarılması süreci değildir; aynı zamanda verinin anlamlandırılması, yapılandırılması ve içgörü oluşturulmasını da içeren daha geniş bir bağlamı ifade etmektedir. Bu bağlamda hesaplamalı arşiv bilimi, arşivciliğin temel ilkelerini bilgisayar bilimi ile bütünleştirerek, arşiv süreçlerinin dijital çağda nasıl yeniden şekillendiğinin anlaşılmasına katkı sunmaktadır. Bu disiplin, yalnızca bilimsel yöntemlerin karşılıklı bilgi ve yöntem aktarımını değil, veri işleme mantığının ve hesaplama perspektifinin arşiv uygulamalarına entegrasyonunu da amaçlamaktadır. Örneğin; yazılım geliştirme süreçlerinde provenans kavramının kullanılması, arşiv bilimindeki geleneksel bir yöntemin teknolojik sistemlerle nasıl entegre edilebileceğine dair önemli bir örnektir. Bu noktada, veri işleme süreçlerinde yapay zekâ ve ilgili teknolojilerin sunduğu olanaklar, arşivcilik alanındaki metodolojik dönüşümün ayrılmaz bir parçası haline gelmiştir. Yapay zekâ, dijital arşivlerde yalnızca büyük veri setlerinin yönetimini kolaylaştırmakla kalmamakta, karmaşık belge yapılarının analizini ve içerik bağlamlandırmasını da mümkün kılmaktadır. Bu doğrultuda ise, yapay zekânın arkasındaki mantığın, hesaplamaların ardındaki matematiksel formüllerin ve diğer temel gerekliliklerin anlaşılması, alan için özellikle günümüzde ve gelecekte son derece büyük önem arz etmektedir. Bu durum için öncelikle yapay zekânın tarihçesi, mantığı, işleyişi, hangi teorik temellere

dayandığı ve uygulamalarının hangi süreçlerden nasıl geçtiğinin anlaşılması gerekmektedir. 2.8. Yapay Zekâ bölümü, bu temel dâhilinde oluşturulmuş ve detaylı açıklamaları yapılmıştır.

2.8. Yapay Zekâ

Tarih boyunca birçok teknolojik ve bilimsel ilerlemenin birtakım kaygı, korku ve direnç oluşturduğu görülmüştür. Örneğin; matbaanın icadı ve sanayi devrimi, insan emeğinin yerini alacağı gerekçesi ile şiddetli direnç gösterilmesine neden olmuştur. Bilgi teknolojilerinin devrimsel ilerlemesi ile kişisel bilgisayar ve internet çağının başlaması da, işlerin makineler (çoğunlukla bilgisayar) tarafından kontrol edileceği ve insan emeğinin yerini, rolünü ve verimliliğini alacağı düşünülerek kaygıya yol açmıştır (Mokyr, Vickers ve Ziebarth, 2015).

Fakat bu yenilikler, uzun süren tecrübeler neticesinde insanın özellikle düşünsel emeğine çoğunlukla muhtaç olarak, daha yeni fırsatlar ve kolaylıklar sağlamıştır. 2. Dünya Savaşı'nın sonlarından itibaren yaklaşık 80 yıllık teknolojik ilerlemenin doruk noktası olarak nitelendirilebilecek yapay zekâ kavramı da, ilk andan itibaren sürekli evrimleşerek, yerini alabildiği emek türlerini kökten değiştirmiştir.

Literatürde yapay zekâ kavramına dair çok çeşitli tanımlar mevcuttur. The International Dictionary of Artificial Intelligence Sözlüğü'nde (Raynor, 1999, s. 13) yapay zekâ; *"bilgisayarların insan gibi zeki bir organizmaya benzer bir şekilde davranmasını sağlayacak teknikler geliştirmekle ilgilenen bir alan"* olarak tanımlanmaktadır. Tanımın devamında ise yapay zekânın amaçları; *"bir programın beklenenden "biraz daha akıllı" göründüğü zayıf uçtan; tamamen bilinçli, akıllı ve bilgisayar tabanlı bir varlık geliştirme girişiminin olduğu güçlü uca kadar değişmektedir"* şeklinde açıklanmaktadır.

Collins Dictionary of Artificial Intelligence Sözlüğü'nde (Smith, 1990, s. 22) ise *"insan zekâsının özelliklerini taklit edebilen bilgisayar tabanlı teknolojiyi anlamaya ve uygulamaya çalışan bilgisayar bilimi alanı"* olarak tanımlanan yapay zekânın temel görevinin; *"sorunları çözmek için deneyimlerden öğrenme yeteneğine sahip bilgisayar programları geliştirmek ve bilgiyi, bilgisayarın çıkarım yapabilmek için sembolleri kullanabileceği temsil yöntemleri oluşturmak"* şeklinde vurgulanmaktadır.

Sciubba (2009, s. 20) yapay zekâyı; *"sembolik ve algoritmik olmayan akıl yürütmeyi inceleyen ve bilgiyi makine çıkarımı örnekleri üretmek için kullanılabilecek bir biçimde temsil etmeye çalışan bilgisayar bilimi bölümü"* olarak tanımlamıştır.

OECD (2024, s. 4) ise yapay zekâyı *"bir yapay zekâ sistemi; açık veya örtük hedefler için aldığı girdiden, fiziksel veya sanal ortamları etkileyebilecek tahmin, içerik, öneriler veya*

kararlar gibi çıktıları nasıl üreteceğini çıkaran makine tabanlı bir sistemdir. Farklı yapay zekâ sistemleri, dağıtımdan sonra özerklik ve uyarlanabilirlik seviyeleri bakımından farklılık gösterir." şeklinde tanımlayarak, bir sistem bağlamına vurgu yapmaktadır. Oldukça genel bir kavram olan yapay zekâ; insan benzeri bilişsel becerilere sahip olan teknolojiyi belirtmek için genelleştirilmiş bir soyut terim olarak kullanılmaktadır. Yapay zekâ; makine öğrenmesi ve derin öğrenme olmak üzere 2 temel alt alanda uygulanmaktadır.

Günümüzde oldukça popüler bir alan hâline gelen yapay zekâ, temelde insan beyninin işleyiş biçimini bilgisayarlara uyarlayarak benzer bir zekâ düzeyine ulaşmayı hedefleyen çok disiplinli bir çalışma alanı olarak tanımlanmaktadır (European Commission, 2018; Sheikh, Prins ve Schrijvers, 2023; El Hadi, 2023). Yapay zekâ sistemleri farklı türden veri ve bilgileri işleyerek, insanların gerçekleştirdiği görevleri çok daha kısa sürede yerine getirebilme potansiyeline sahiptir. Başka bir deyişle yapay zekâ, insan zihnini taklit ederek belirli düzeyde özerklikle bilgi türetmeyi amaçlamaktadır. Bu doğrultuda, aşağıdan yukarıya (bottom-up) yaklaşımı temel alan modellerde, büyük veri kümeleri kullanılarak öğrenme süreçleri gerçekleştirilmektedir. 2. Dünya Savaşı sonrası gelişme alanı bulan hesaplamalı bilimler ve özellikle bilgisayar bilimi ile yapay zekânın köklerinin ortaya çıkışı da hemen hemen aynı döneme rastlamaktadır. Çünkü makinelere insan zekâsının çalışma prensibinin öğretilmesi süreci, mutlaka bir hesaplama cihazına ihtiyaç duymaktaydı. Böyle bir ihtiyaç ise programlanabilir bilgisayarların ortaya çıkışına denk gelmektedir. Yapay zekâ veya daha genel olarak bilgisayarların insan gibi düşünebileceği fikri, bilgisayar bilimci Alan Turing'in (1950), makinelerin insandan ayırt edilemeyecek zeki bir davranış sergileme yeteneğini değerlendirmeyi amaçlayan öncü Turing Testi ile başlayan çalışmalarından bu yana, yaklaşık 70 yıldan fazla bir süredir yapay zekânın insan benzeri düşünme kapasitesi bağlamında nasıl tanımlandığı ve bu tanımların hangi teorik çerçevelerde şekillendiği, literatürde oldukça geniş ve çok yönlü şekilde ele alınmıştır. Bu alandaki farklı yaklaşımlar geniş ölçekli çalışmalarda (Bostrom ve Yudkowsky, 2014; Jordan ve Mitchell, 2015; Jiang vd., 2017; McLaren vd., 2017; Kaplan ve Haenlein, 2019; Wamba vd., 2021; Bawack vd., 2022; Arseniev-Koehler ve Foster, 2022) sıklıkla tartışılmaktadır.

1980'lerde yapay zekâ, genellikle *"karar kurallarıyla birlikte alan uzmanlarının sezgisel bilgisine dayanan"* (Orhaug, 1987, s. 167) uzman sistemler biçiminde, karar desteği sunan karmaşık ve deterministik yapılara indirgenmişti. Bu dönemde işlem gücünün sınırlı olması, bazı alanlarda yapay zekâ uygulamalarının, geniş arama alanlarına dayalı sezgisel olmayan problem çözme stratejileriyle sınırlandırılmasına yol açmıştır. Yapay zekâya yönelik erken

dönem yaklaşımların çoğu, iyi tanımlanmış ve düzenli veri yapıları eşliğinde, makinelerin mantıksal çıkarımlar yapabilmesini sağlayacak kuralları belirlemeye odaklanmıştır (Henderson, 2009, s. 26).

Yapay zekânın tarihçesi incelendiğinde, belirli dönüm noktalarının alanda oynadığı son derece önemli rol rahatlıkla görülebilir. McCulloch ve Pitts (1943), sinir ağlarına dair ilk matematiksel modeli açıklayan "A Logical Calculus of the Ideas Immanent in Nervous Activity" adlı makalesinde; sinir ağlarının davranışını tanımlamak için mantıksal bir hesaplamanın geliştirilebileceğini öne sürmüştür. Nöronların davranışı hakkında belirli varsayımlar (tepkiler, sinaptik gecikmeler[58], eşikler vb.) verildiğinde sinir ağı aktivitesi, önermesel mantık[59] kullanılarak temsil edilebilir. Başka bir deyişle, sinir aktivitesinin "sıfır veya bir" karakteri nedeniyle, sinir olayları ve bunlar arasındaki ilişkiler önermesel mantık yoluyla ele alınabilir.

Turing (1950), "Computing Machinery and Intelligence" başlıklı makalesinde, "makineler düşünebilir mi?" sorunsalını ortaya atmış ve bu sorunun daha test edilebilir bir biçimi olarak "taklit oyunu"nu[60] önermiştir. Turing, bu oyunun bir versiyonunda bir erkeğin (A), bir kadının (B) ve bir sorgulayıcının (C) yer aldığını belirtir. Sorgulayıcı, yazılı iletişim yoluyla hangisinin erkek, hangisinin kadın olduğunu belirlemeye çalışır. Bu kurgu, bir makinenin A'nın yerine geçmesiyle yeniden yapılandırılır. Böylece soru şu hâli alır: Sorgulayıcı, makinenin yer aldığı bir oyunda insanlar arasındaki oyunda yaptığı kadar hata yapar mı? Turing, bu soru üzerinden makinelerin düşünme yetisine sahip olup olmadığını sorgulamış ve matematiksel bilinç, öğrenme ve bilinçsizlik gibi çeşitli itirazları inceleyerek, yaklaşık elli yıl içinde makinelerin taklit oyununu insanlardan ayırt edilemeyecek düzeyde oynayabileceğini ileri sürmüştür.

McCarthy vd., (1955), 1956 yazında Dartmouth/New Hampshire'da düzenleyecekleri konferans için ilk defa "yapay zekâ" terimini ortaya atarak, bu terimi bir araştırma disiplini olarak çalıştay öncesinde önermiştir. Konferansta, insan zekâsını simüle edebilen makineler geliştirmeyi amaçlayan yeni bir çalışma alanı öngörülmüştür. Bu konferans; makine öğrenmesi, muhakeme ve algı olanaklarını keşfetmek için matematik, mühendislik, psikoloji ve bilgisayar bilimi gibi farklı alanlardan bilim insanları ve araştırmacıları bir araya getirerek, yapay zekâyı meşru bir akademik disiplin ve bilimsel uğraş olarak resmileştirmiştir. Moor (2006, s. 87); o tarihlerde alanın genel bir teorisi ve özellikle de genel bir öğrenme teorisi

[58] Presinaptik membrandan içe doğru akımın zirvesi ile postsinaptik membrandan içe doğru akımın başlangıcı arasındaki zaman aralığı.
[59] Cümle mantığı veya ifade mantığı olarak da bilinir. Tüm önermeleri, ifadeleri veya cümleleri; daha karmaşık önermeler, ifadeler veya cümleler oluşturmak için birleştirme veya değiştirme yollarını ve ifadeleri birleştirme veya değiştirme yöntemlerinden türetilen mantıksal ilişkileri ve özellikleri inceleyen mantık dalı.
[60] Imitation game

konusunda bir fikir birliğinin olmadığını belirterek, yapay zeka alanının metodoloji, problem seçimi veya genel teori konusunda bir fikir birliğiyle değil, sadece "bilgisayarların zeki görevler gerçekleştirebileceği ortak vizyonu" ile başlatıldığını öne sürmektedir. Böyle bir vizyon ise, konferans teklif metninde şu şekilde ifade edilmiştir: *"Çalışma, öğrenmenin her yönünün veya zekanın herhangi bir diğer özelliğinin prensipte o kadar kesin bir şekilde tanımlanabileceği varsayımına dayanarak ilerleyecektir ki, bir makine bunu simüle edebilir."* (McCarthy, Minsky, Rochester ve Shannon, 1955, s. 2).

Weizenbaum (1966), "ELIZA - A Computer Program for the Study of Natural Language Communication between Man and Machine" başlıklı makalesinde, Rogerian psikoterapisini[61] taklit eden bir bilgisayar programını tanıtmış ve bu basit sistemin insanlar tarafından gerçek zekâya sahipmiş gibi algılanmasının ne denli kolay olduğunu ortaya koymuştur. Çalışma, insan-makine etkileşiminde doğabilecek bilişsel yanılsamalara ve etik risklere dikkat çekmesi bakımından öncü bir nitelik taşımaktadır. ELIZA'nın simülasyonu, kullanıcı girdisinde yer alan anahtar kelimeleri tespit ederek ve bu kelimelere karşılık gelen önceden tanımlanmış dönüşüm kurallarına göre yanıtlar üretmek üzerine kuruludur. Herhangi bir anahtar kelime eşleşmesi bulunmadığında, genel geçer ifadelerle iletişim sürdürmeye çalışılır ya da önceki kurallardan türetilen yapılar kullanılır. Programın temel işlevi, metni dizisel olarak ayrıştırmak ve yeniden yapılandırmak yoluyla basit bir metin manipülasyonu gerçekleştirmektir. ELIZA'nın tasarımı, anlamaya dayalı bir işlemden çok, kullanıcının anlam veriyormuş izlenimini sürdürmesini amaçlayan bir anlayış yanılsamasına odaklanmaktadır.

Colmerauer vd. (1972), Fransızca doğal dil işleme projesi kapsamında Prolog programlama dilini geliştirerek ayrıntılı biçimde tanımlamışlardır. Bu süreçte, Q-sistemleri[62] temel alınmış; önceden belirlenmiş seçim sırasına sahip bir çözümleme stratejisi ve mantıksal çıkarıma dayalı çözüm ilkeleri uygulanmıştır. Prolog'un geliştirilmesinde ayrıca, teorem ispatlayıcıların programlama dillerine uyarlanması, birleştirme (unification) ve geri izleme (backtracking) gibi hesaplama mekanizmalarının tanıtılması önemli rol oynamıştır. Prolog, özellikle doğrusal çözüm stratejisi ile doğal dil işleme ve otomatik teorem ispatı alanlarında uygulanabilirliğini ortaya koymuştur.

[61] Kişi merkezli terapi olarak da bilinir. 1940'ların başında Carl Rogers tarafından önerilmiştir. İnsanların pozitif psikolojik işleyiş elde etmeye doğal olarak motive oldukları fikrine dayanır. Empati, uyum ve koşulsuz pozitif saygı gibi temel koşullar kullanılarak danışan, kendi potansiyeline erişebilecek kadar güvende hissedebilir.

[62] 1967-1970 yılları arasında Alain Colmerauer tarafından, doğal dil işlemede kullanılmak üzere Montreal Üniversitesi'nde geliştirilen ve verilen gramer kurallarına göre yönlendirilmiş grafik dönüşümlerini yapan yöntemler bütünü.

1980'lere gelindiğinde, "yapay zekâ kışı[63]" olarak bilinen zorlu bir dönem başlamıştır. Bu terim, teknolojinin abartılı beklentileri karşılayamaması nedeniyle yapay zekâ araştırmalarına olan ilgi ve yatırımın önemli ölçüde azaldığı bir zamanı ifade etmektedir (Floridi, 2020; Toosi vd., 2021; Haigh, 2023). İlk yapay zekâ sistemleri, özellikle uzman sistemler ve sembolik yapay zekâ; ölçeklenebilirlik, uyarlanabilirlik ve gerçek dünya performansında sınırlamalarla karşı karşıya olduğundan dolayı kuruluşlar ve hükümetler projelere ayrılan fonları zamanla azaltmış ve bu eksiklikler belirginleştikçe alandaki coşku ve heyecan da azalmıştır (Negnevitsky, 1997). İnsan seviyesinde zekâ vaadi gibi kritik projeler ve abartılı tahminler gerçekleşmemiş ve bu durum yaygın bir şüpheciliğe yol açmıştır. Ancak, 1990'lara doğru tekrardan bir canlanma gerçekleşmiştir. Araştırmacılar, hesaplama gücü ve algoritmalardaki gelişmelerin yardımıyla sinir ağlarını daha ciddi bir şekilde incelemeye başlamışlardır. Çok katmanlı sinir ağlarını eğitmek için bir yöntem olan geri yayılım[64] algoritması bu dönemde öne çıkmıştır. Böyle bir atılım, makine öğrenmesinin sembolik yapay zekânın bazı sınırlamalarını aşma potansiyelini göstererek, 1990'larda ve sonrasında yeniden canlanmasına katkı sunmuştur.

1997 yılında IBM tarafından geliştirilen Deep Blue adlı süper bilgisayarın, dünya satranç şampiyonu Garry Kasparov'u altı maçlık seride 3,5-2,5 mağlup etmesi, yapay zekâ tarihinde simgesel bir dönüm noktası olarak kabul edilmektedir. Russell ve Norvig (2016, s. 185), bu gelişmeyi yapay zekânın belirli görevlerdeki hesaplama gücüne dayalı başarısını gösteren önemli örneklerden biri olarak değerlendirmekte; ancak Deep Blue'nun yalnızca kurallara dayalı deterministik arama algoritmalarıyla çalışan, öğrenme ya da genelleme yeteneği bulunmayan bir sistem olduğunu da vurgulamaktadır. Nitekim bu nedenle bazı araştırmacılar, söz konusu başarıyı "gerçek" yapay zekâ ile ilişkilendirmemektedir (Heßler, 2017). Deep Blue, saniyede milyonlarca satranç pozisyonunu analiz edebilen özel donanım ve yazılım altyapısına sahip olup, yalnızca satranç oynamaya özgü olarak tasarlanmıştır. Bu başarı, paralel hesaplama, algoritmik iyileştirmeler ve hesaplama gücündeki ilerlemelerin birleşimiyle mümkün olmuş; aynı zamanda yapay zekânın dar ama etkili uygulama alanlarındaki potansiyelini gözler önüne sermiştir (Bory, 2019).

[63] AI winter
[64] Back propagation

Hinton, Osindero ve Teh (2006), "A Fast Learning Algorithm for Deep Belief Nets" adlı makalesinde; derin inanç ağlarını[65] eğitmek için geliştirilmiş karşıtsal ıraksama[66] algoritmasını tanıtmışlardır. Bu algoritma, daha hızlı ve katman katman ilerleyen[67] bir eğitim yöntemidir. Algoritma, çoklu gizli katmanlara sahip yoğun şekilde birbirine bağlı inanç ağlarındaki etkileri açıklamanın getirdiği zorlukların üstesinden gelmek için "tamamlayıcı önselleri[68]" kullanmaktadır. Bu, daha sonra wake-sleep algoritmasının[69] karşılaştırmalı bir versiyonu kullanılarak rafine edilen verimli katman eğitime olanak tanımaktadır. Ortaya çıkan üretken model, MNIST[70] veri kümesindeki önde gelen ayrımcı yöntemlere kıyasla çok daha iyi rakam sınıflandırma sonuçları elde etmiştir. Çalışmada; karşıtsal ıraksama öğrenmesinin, her katmanda farklı ağırlıklara sahip derin ve çok katmanlı ağları eğitmek ve "hırslı[71]" bir algoritmanın ağırlıkları birer katman halinde öğrenmek için kullanılabileceği bulgularına ulaşılmıştır. Önerilen ağ, MNIST veri setinde %1,25'lik bir hata oranına ulaşarak, ileri beslemeli sinir ağları ve destek vektör makineleri gibi diğer makine öğrenmesi tekniklerinden daha iyi performans göstermiştir. Diğer önemli bulgular arasında ise; üretken modellerin etiketli veri gerektirmeden düşük düzeyli özellikleri öğrenebilmesi, aşırı öğrenme riski olmadan ayrımcı modellere kıyasla daha fazla parametreyi temsil edebilmesi ve derin katmanlardaki doğrusal olmayan dağıtılmış gösterimlerin, görüntü üretimi yoluyla yorumlanabilir hâle gelmesi yer almaktadır.

Mart 2016'da ise Google DeepMind tarafından sadece Go oyunu için geliştirilen bir yapay zekâ programı olan AlphaGo, Go şampiyonlarından Lee Sedol'u beş setlik bir karşılaşmada yenmiştir. Çin kökenli eski bir masa oyunu olan Go, bilgisayarlar için satranç gibi oyunlara kıyasla yüksek seviyede oynanmakta ve dolaysıyla satranç hamlelerinden çok daha zor olasılıklarla hareket edilmesini getirmektedir. Go; sezgi, örüntü tanıma ve derin stratejik düşünme gerektiren bir oyun olarak kabul edildiğinden, programın bu zaferi yapay zekânın insana özgü niteliklerden stratejik düşünme potansiyelinin önemli bir göstergesi olarak

[65] Makine öğrenmesinin bir alt kümesi olan derin öğrenme alanında kullanılan karmaşık yapay sinir ağlarıdır. Büyük veri kümeleri içindeki kalıpları otomatik olarak keşfetmek ve öğrenmek için tasarlanmıştır. Her katmanın bir öncekinden alınan bilgileri anlamlandırabildiği ve genel veriler hakkında karmaşık bir anlayış geliştirebildiği çok katmanlı ağlar olarak düşünülebilir.
[66] Contrastive divergence
[67] Greedy
[68] Complementary prior. Özellikle derin öğrenmede kullanılan olasılıksal grafik modellerinde, gözlenen verilerle birleştiğinde (likelihood) kolaylıkla çözülebilen bir arka dağılım (posterior) elde etmeye olanak tanıyan, özel olarak seçilmiş veya öğrenilmiş önsel dağılım.
[69] Özellikle Helmholtz Makineleri olmak üzere derin üretken modeller için gözetimsiz bir öğrenme algoritmasıdır.
[70] Modified National Institute of Standards and Technology Database. Çeşitli görüntü işleme sistemlerinin eğitimi için yaygın olarak kullanılan el yazısıyla yazılmış rakamların büyük bir veritabanı.

değerlendirilmektedir (Hölldobler, Möhle ve Tigunova, 2017). AlphaGo, derin sinir ağları ve pekiştirmeli öğrenmeyi temel alarak, hem gerçek oyunların hem de kendi kendine oynadığı oyunların bir kombinasyonundan öğrenmeyi gerçekleştirmiştir. Programın mimarisi; oyun pozisyonlarını değerlendirmek ve karar vermek için Monte Carlo Ağaç Araması (MCTS[72]), minimax algoritması, alfa-beta budaması ve yinelemeli derinleştirme gibi çeşitli derin öğrenme tekniklerini kapsamaktadır. Programın bu başarısı; makinelerin sadece hesaplama ve örüntü tanımaya dayalı oyunlarda ustalaşmakla kalmayıp, aynı zamanda daha önce yeteneklerinin ötesinde olduğu düşünülen stratejik derinlik sergileyebileceğini göstermesi açısından da yapay zekâ için bir dönüm noktası olarak değerlendirilmiştir (Granter, Beck ve Papke, 2017; Fazi, 2021).

OpenAI tarafından 2018 yılında sunulan yeni bir dil modeli olan GPT[73]-1, doğal dil işlemeyi geliştirmede temel bir standart uygulama haline gelen GPT serisinin başlangıcını oluşturmaktadır. Model, doğal dilin işlenebilmesinde RNN[74] veya LSTM[75] gibi önceki yöntemlerden daha verimli bir şekilde öz dikkat mekanizmalarından yararlanan dönüştürücü mimariye dayanmaktadır (Kamnis, 2023). 117 milyon parametre ile eğitilen model, GPT-2 ve GPT-3 gibi sonraki versiyonlarıyla karşılaştırıldığında oldukça düşük bir parametre sayısına sahiptir. GPT-1 modelinin getirdiği yenilik; iki aşamalı eğitim sürecinde kendisini göstermektedir: Ön eğitimde model, bir cümlede bulunan sonraki kelimeleri tahmin etmeyi öğrenmek amacıyla büyük bir metin verisi gövdesi üzerinde denetimsiz olarak eğitilmiştir. Metin sınıflandırması veya soru cevaplama gibi etiketli verilerle yapılan belirli görevlerde ise ince ayar yapılmıştır (Yenduri, vd., 2024). Bu yaklaşım, büyük ve önceden eğitilmiş bir dil modelinin, minimum görev ve özel eğitimle etkileyici bir performans elde edebileceğini göstermiştir. Yaklaşık 175 milyar parametreyle eğitilen GPT-3 dil modeli ise; diyalog, metin tamamlama, çeviri ve yazma gibi görevlerde neredeyse mükemmelleşerek dil modelleri için yeni bir standart haline gelmiştir. Bu başarıya, bağlamı doğru şekilde anlayabilmesi ve çok çeşitli konularda tutarlı ve anlamlı yanıtlar verebilme yeteneği sayesinde ulaşmıştır (Imamguluyev, 2023; Yenduri vd., 2024).

2019'a gelindiğinde ise yine Google Deep Mind tarafından geliştirilen AlphaStar yapay zekâ programı, gerçek zamanlı strateji oyunu StarCraft 2'de tüm oyuncuların %99,8'ini geride bırakarak "büyük usta[76]" statüsüne ulaşmıştır. Program, derin pekiştirmeli öğrenme ve taklit

[72] Monte Carlo Tree Search
[73] Generative Pre-trained Transformer
[74] Recurrent Neural Network
[75] Long Short-Term Memory
[76] Grand master

öğrenmesini kullanarak, milyonlarca oyuncunun nasıl oynadığını gözlemledikten sonra oyunu kendi kendine oynayıp stratejilerini geliştirmiş ve simüle etmiştir. Programda oyunun hem mekansal (harita düzeni) hem de zamansal (olay dizisi) yönlerini içeren bir sinir ağı mimarisi kullanmıştır (Vinyals vd., 2019; Garisto, 2019; Risi ve Preuss, 2020).

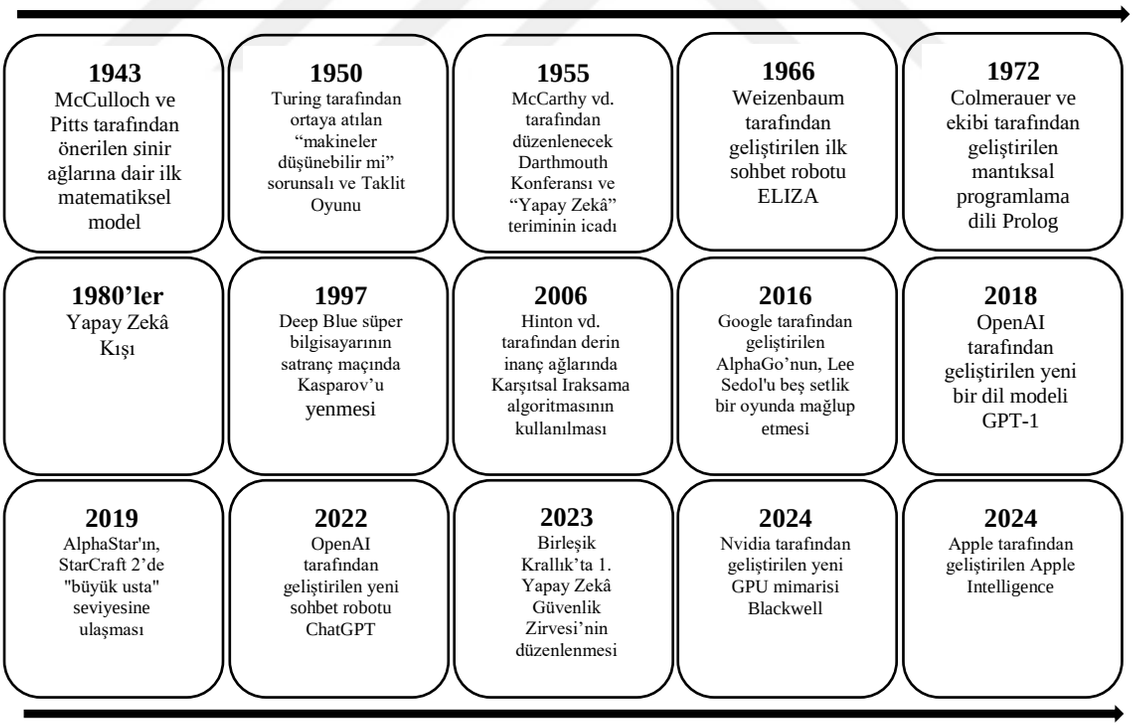
2022'de ise OpenAI, geliştirdiği GPT modellerini sohbet robotuna dönüştürerek insan benzeri konuşma yeteneklerini simüle etmek için tasarlanan ChatGPT'yi tanıtmıştır. Uygulama, önceki sürümler olan GPT-3.5 ve GPT-4 mimarilerinin bir türevi olarak ortaya çıkmıştır. Soru cevaplama, içerik oluşturma ve problem çözme gibi görevlerde oldukça başarılı bir yeterlilik göstererek, kural tabanlı ve sezgisel odaklı sohbet robotlarının önceki sınırlamalarını aşmıştır. ChatGPT-4 yaklaşık 1.8 trilyon parametre ile eğitilmiştir (Howarth, 2024). ChatGPT modeli, doğal dil işleme ve yapay konuşma zekâsının evriminde önemli bir dönüm noktasını temsil etmektedir. ChatGPT, metin dizilerini verimli bir şekilde işlemek için öz-dikkat mekanizmaları kullanan transformatör mimarisi üzerine inşa edilmiştir (Mechkaroska vd., 2024). Parametreler, eğitim sırasında öğrenilen ağırlıkları ve önyargıları temsil ettikleri için bir modelin kapasitesini belirlemede oldukça önemlidir. Sadece ölçeklendirmeye güvenmeden performansı artırmak için ince ayar ve insan geri bildiriminden takviyeli öğrenme (RLHF[77]) gibi optimizasyon teknikleri de kullanılmıştır (Hadi, Abdulredha ve Hasan, 2023).

2023'te, Birleşik Krallık hükümetinin liderliğinde düzenlenen ve yapay zekânın hızlı ilerlemesi sonucu oluşan veya oluşabilecek tüm riskleri ele almak ve bu risklere dair çözümleri tartışmak için resmi ve özel kuruluşlar ile sivil toplum temsilcilerinin katıldığı 1. Yapay Zekâ Güvenlik Zirvesi düzenlenmiştir. Zirvenin yapıldığı yer olan Bletchley Park, 2. Dünya Savaşı sırasında Almanların kullandığı Enigma şifreleme cihazının kodlarının kırıldığı yer olması bakımından da dikkat çekicidir (Burki, 2024, s. e91). Zirvedeki başlıuca tartışılan konular arasında yapay zekâ riskleriyle ilgili artan endişeler, varoluşsal riskler, öngörülemeyen davranışlar, silahlaştırma ve yanlış bilgilendirme bulunmaktadır. Bu konular bağlamında zirvenin temel amacı; yapay zekanın kötü niyetli kullanımı, çeşitli önyargıların üretilmesi, diğer sistemlerle öngörülemeyen entegrasyon-etkileşim veya kontrol kaybı nedeniyle olumsuz sonuçlar üretebileceği çeşitli senaryolar önermek ve bunlara çözüm aramaktır (UK Government, t.y.).

[77] Reinforcement Learning From Human Feedback

2024'te Nvidia, ekran kartlarının hesaplama özelliklerini daha da geliştiren yeni bir mimariyi duyurmuştur. Blackwell adını alan mimari, tek bir GPU[78] üzerinde 20 Petaflops[79] değerinde performans sergilemektedir. Böyle bir yapı, trilyon ve üzeri parametreye sahip mimariler de dahil olmak üzere karmaşık yapay zeka modellerinin hızlı bir şekilde işlenmesini sağlayarak daha hızlı eğitim (epoch) ve çıktı (output) sürecini son derece kolaylaştırmaktadır. Mimari ayrıca saniyede 7,2 terabaytlık çift yönlü bir bant genişliği elde ederek minimum gecikme ve optimum veri çıkışı sağlamaktadır (Hinsche, 2024). Bu özellik ise büyük ölçekli dağıtık hesaplama yapısında darboğazı minimuma çekerek genel sistem verimliliğini ve ölçeklenebilirliği artırmaktadır.

Yine 2024'te, Apple, "Apple Intelligence" adlı yapay zekâ programını bilgisayar ve telefonlardaki işletim sistemlerine entegre etmiştir. Program, yaklaşık 3 milyar parametreye sahip bir sunucu tabanlı üretken dil modeli şeklinde geliştirilmiştir. Dil modeli içerisinde; metin yazma ve iyileştirme, bildirimleri önceliklendirme ve özetleme, aile ve arkadaşlarla yapılan sohbetler için görseller oluşturma ve uygulama içi eylemler gerçekleştirme gibi kullanıcı deneyimlerine yönelik alanlar bulunmaktadır (Gunter vd., 2024). Tüm bu bilgiler ışığında, yapay zekânın geçirdiği evreler Şekil 4'te gösterilmiştir.



Şekil 4. Yapay Zekânın Gelişiminde Önemli Evreler

[78] Graphics Processing Unit

[79] Bir bilgisayar sisteminin bir saniyede gerçekleştirebileceği matematiksel işlem sayısını niceliksel olarak ifade eden ölçü birimi. Özellikle yüksek performanslı hesapalama ve yapay zekâ uygulamalarında hesaplama performansını ölçmek için kullanılır. 1 petaflop kapasitesindeki bir sistem, saniyede 10^{15} işlemi gerçekleştirebilir.

Zaman içerisinde disiplinlerarası bir çalışma alanına dönüşen yapay zekâ; günümüzde matematik, sinirbilim, tıp, çeşitli mühendislik dalları ve sosyal bilimler gibi alanlarla kesişmiş ve böylece çok boyutlu bir yapı özelliği kazanmıştır.

Yapay zekâ, genel olarak üç türe ayrılmakta ve bunlar "yapay dar zekâ" , "yapay genel zekâ" ve" yapay süper zekâ" olarak adlandırılmaktadır. Sadece belirli görevlere sınırlı kapsamda uygulanabilen ve "yapay dar zekâ[80]" olarak adlandırılan birinci nesil sistemler ve uygulamalar; öneri sistemlerinden (Netflix, Amazon Prime vb.) dil modellerine (ChatGPT, Copilot vb.); sağlık uygulamalarından sanal asistanlara (Siri, Alexa vb.) kadar neredeyse her yerde aktif olarak kullanılmaktadır (Aljaber ve Almushaili, 2022). Bu sistemler son derece uzmanlaşmış olup, önceden tanımlanmış parametrelerle çalışmaktadır. Yapay dar zekâ, programlamanın dışındaki görevleri gerçekleştirme yeteneğine sahip değildir. Yapay genel zekâ[81]; bir insanın yapabileceği herhangi bir entelektüel görevi yerine getirebilen makineleri ifade etmektedir. Bu sistemler; yeni ortamlara uyum sağlayarak ve daha önce karşılaşmadıkları sorunları çözerek çok çeşitli görevlerde bilgiyi öğrenme, anlama ve uygulama yeteneğine sahiptir. Ayrıca insanlar gibi düşünebilir, akıl yürütebilir ve uyum sağlayabilir. Günümüzde hipotetik[82] olarak değerlendirilmektedir (Dambrot, 2020). 2001: A Space Odyssey filmindeki HAL 9000, yapay genel zekâya örnek olarak gösterilebilir. Yapay süper zekâ[83] ise; üreticilik, karar verme ve problem çözme gibi durumlar karşısında her bakımdan insan zekâsını aşan teorik bir yapay zekâ sistemini ifade etmektedir. Tahminlere göre yapay süper zekâ, kendisinin farkında olacak (bir benlik ve içgörü) ve üstel bir oranda kendini geliştirebilecek kapasiteye erişebilecektir (Aleixo vd., 2024). Yapay süper zekâ otonom kararlar alabilir ve insanlık için büyük bir tehdit oluşturabilir. Günümüzde ve yakın gelecekte kullanılması mümkün görünmemektedir. Tablo 10'da yapay zekânın üç türüne dair karşılaştırmalar verilmiştir.

Tablo 10. Yapay Zekâ Türleri ve Karşılaştırmaları

Tür	Yetenekler	Örnek Uygulamalar	Varlık Durumu
Yapay Dar Zekâ	Sadece belirli bir göreve özgü uygulamalar Belirli alanlarda insanlardan daha iyi veya eşit performans gösterir	Siri, Netflix öneri sistemi, ChatGPT	Günümüzde aktif olarak kullanımda olan tek yapay zekâ türü
Yapay Genel Zekâ	İnsan benzeri zekâya sahiptir ve bilgiyi öğrenir, anlamlandırır ve uygular Çok çeşitli alanlara uygulanabilir Birçok alanda insanlardan daha iyi veya eşit performans gösterir	(Teoride) İnsansı bir robotun ofis işlerini yapabilmesi veya yemek hazırlaması	Yok
Yapay Süper Zekâ	İnsanın zekâ seviyesini aşar ve öz-bilinçli hale gelir	Hayal gücüne dayalıdır ve	Yok

[80] Artificial Narrow Intelligence (ANI)
[81] Artificial General Intelligence (AGI)
[82] Varsayıma dayalı
[83] Artifical Super Intelligence (ASI)

	Herhangi bir alana uygulanabilir	Terminatör film serisindeki SkyNet sistemi gibi bilimkurgu filmleri	
	Sorunları anında çözebilme yeteneğine sahiptir		
	Tüm alanlarda insanlardan çok daha iyi performans gösterir		

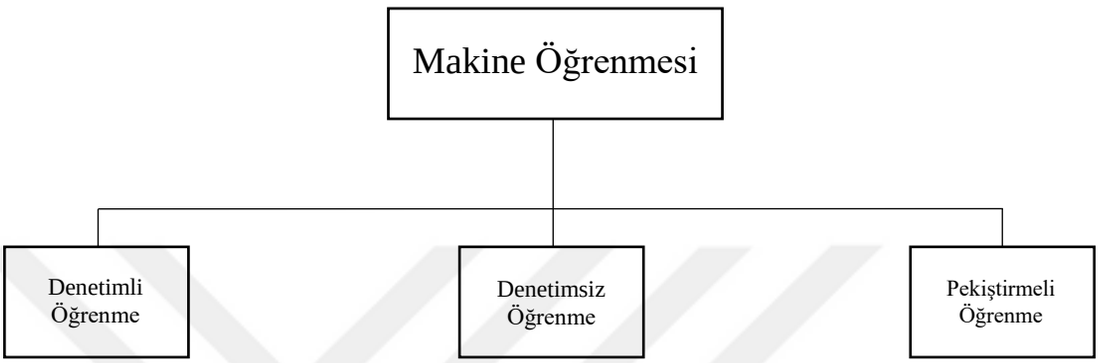
Tablo 10'da belirtildiği üzere yapay dar zekâ, "şimdilik" kullanımda olan tek yapay zekâ türü olarak karşımıza çıkmaktadır. Çoğunlukla görev odaklı problem çözümünde kullanılan bu tür, popüler alanlarda sıklıkla destekleyici görevlerde uygulanmaktadır (Beaulac ve Larribe, 2017; Kuusi ve Heinonen, 2022; Shadbolt, 2022; Sheikh, Prins ve Schrijvers, 2023). Yapay genel zekâ ise, yapay dar zekânın oldukça gelişmiş bir versiyonudur ve insan benzeri zekâya sahiptir. Henüz kullanımda olmasa da, yapay zeka çalışmalarının temel hedeflerinden biri olarak değerlendirilmektedir (Müller, 2014; Goertzel, 2014; Gobble, 2019; McLean vd., 2021; Fei, vd. 2022; Williams, 2022; Chehreghani, 2024). Yapay süper zekâ ise insan zekâsını aşarak kendi öz bilincini oluşturabilir ve sürekli gelişerek insanlık için yıkıcı bir tehdite dönüşebilir. Bilim kurgu filmlerine konu olan bu tür yapay zeka, uzak bir gelecekte bile uygulanabilir görülmemektedir (Gill, 2016; Sergievskii, 2020; Alfonseca vd., 2021; Zohuri, 2023).

2.8.1. Makine Öğrenmesi

Büyük verinin ortaya çıkışıyla birlikte yaşanan veri -ve dolayısıyla bilgi- patlaması, hesaplamalı yöntemlerin analitik süreçlerde, bilgi keşfinde ve karmaşık problemlerin çözümüne yönelik karar verme mekanizmalarında daha sık kullanılmasını sağlamıştır. Bu hesaplamalı yöntemlerden biri de, yapay zekâ bağlamında öne çıkan makine öğrenmesi alanıdır. Bartneck vd. (2021, s. 11) makine öğrenmesini *"bir görev sınıfına ilişkin deneyimi ve bu görevdeki performanslarını iyileştirmek için bir performans ölçüsü biçimindeki geri bildirimi kullanan algoritmaların oluşturulmasına odaklanan bir yapay zekâ alt alanı"* olarak tanımlamaktadır. Ayodele (2010, s. 9), makine öğrenmesini *"deneyimle otomatik olarak iyileşen ve bir öğrenme süreci uygulayan bilgisayar sistemleri oluşturma süreci"* olarak tanımlamış ve *"faydalı olasılıksal modeller oluşturarak bir veri gövdesinden yararlı bilgilerin otomatik olarak çıkarılması"* ile *"genel bir teorinin olmadığı, çok fazla verinin olduğu alanlar için ideal"* bir uygulama olduğunu öne sürmüştür.

Makine öğrenmesi, Mitchell'e göre (1997, s. 2); deneyim ve daha fazla verinin ortaya çıkmasıyla otomatik olarak iyileşen bilgisayar algoritmalarının incelenmesi olarak tanımlanmakta ve geleneksel programlama algoritmalarının kullanılmasının pratik olmadığı yerlerde kullanılmaktadır. Makine öğrenmesi temelde; büyük ölçüde kural tabanlı geleneksel

programlama ve hesaplama yöntemlerinin aksine, verilerden otonom[84] olarak örüntüler ve çeşitli yapılar öğrenerek; gözlemlenen verilere dayalı kararlar, tahminler ve sınıflandırmalar yapılmasını sağlamaktadır. Kural tabanlı programlamadan veri odaklı programlamaya geçiş, yaşamın neredeyse tüm alanlarını (sağlık, finans, eğitim, ofis işleri, üretim, tüketim vb.) önemli ölçüde etkilemiştir. Başka bir deyişle yapay zekâ, neredeyse tüm rutin görevleri devralarak, daha yüksek katma değerli stratejik çalışmalara odaklanılmasını sağlamıştır.



Şekil 5. Makine Öğrenmesi Türleri

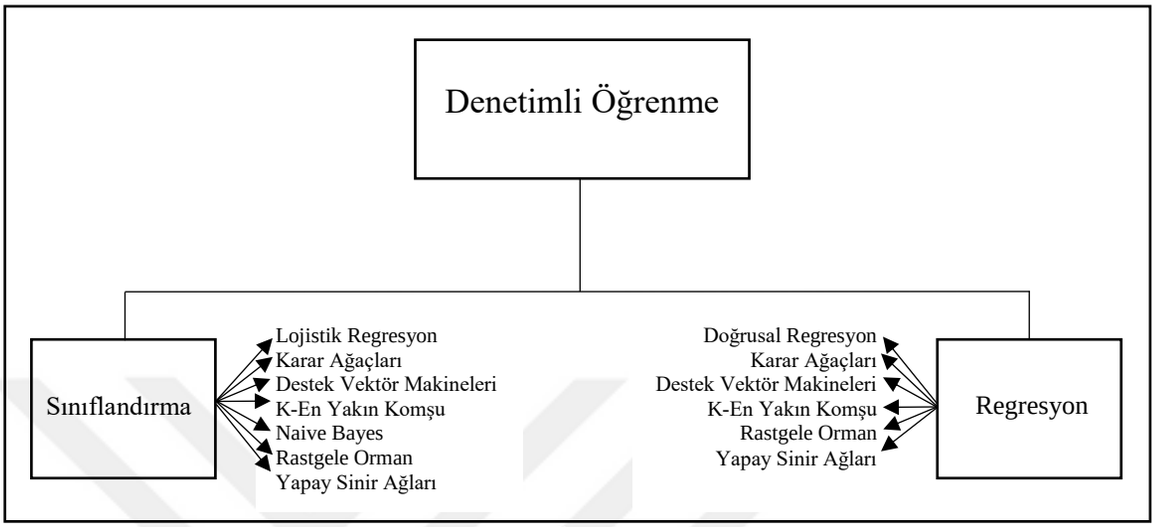
Makine öğrenmesi; derin öğrenme alt alanını da içerecek şekilde günümüzde konuşma ve ses tanıma, örüntü tanıma, tahmin, sınıflandırma, otonom sürüş ve oyun oynama gibi görevlerde, daha önceleri deneyimlenemeyen ve oldukça yüksek doğruluk ve verimlilik seviyelerine ulaşmış, büyük veriden ve özellikle derin sinir ağı mimarilerinden yararlanarak adeta yeni bir BİT devrimini sağlamıştır (Haeb-Umbach, Beyerlein ve Geller, 1995; Siahaan ve Siahaan, 2016; Stavropoulos vd., 2023; Dahlström ve Salminen, 2024; Neha, Yadav ve Goyal, 2024). Makine öğrenmesi; Şekil 5'te gösterildiği üzere, literatürde yöntem olarak çoğunlukla denetimli öğrenme, denetimsiz öğrenme ve pekiştirmeli öğrenme olmak üzere üç alt bölümlendirmeye ayrılmıştır. İstatistik, optimizasyon teorisi ve cebir gibi matematiksel temel ve araçların da yardımıyla, bu üç yöntemin kullanıldığı öğrenme modelleri tasarlanmakta ve bu modellerin başarımları ölçülebilir hale gelmektedir.

2.8.1.1. Denetimli Öğrenme

Denetimli öğrenme, bilinen sonuçlara sahip veriden öğrenen bir algoritma türüdür (Sarker, 2021). Bu öğrenme türünde geliştirilen model, etiketli veriler üzerinde eğitilir. Böylece veri kümesindeki her bir girdiye bir çıktı eşlenerek, belirli bir etiket veya sonuç atanır. Bu işlem, algoritmaya tahminlerinde hedefleyeceği bir ölçüt sağlar ve öğrendikçe doğruluğunu artırır. He, Yan ve Tan'a göre (2022) denetimli öğrenme algoritmalarının amacı; istatistiksel hata

[84] Kendi kendine

oranını genelleştirilebilir ve istenen bir sonuçla daha yakından eşleşecek şekilde ayarlamaktır. Başka bir deyişle, model tarafından önceden görülmemiş verilere iyi bir genelleme yapmaya yarayan eşleme işlevini öğrenmek ve böylece minimum hata ile maksimum doğruluk sağlamaktır.

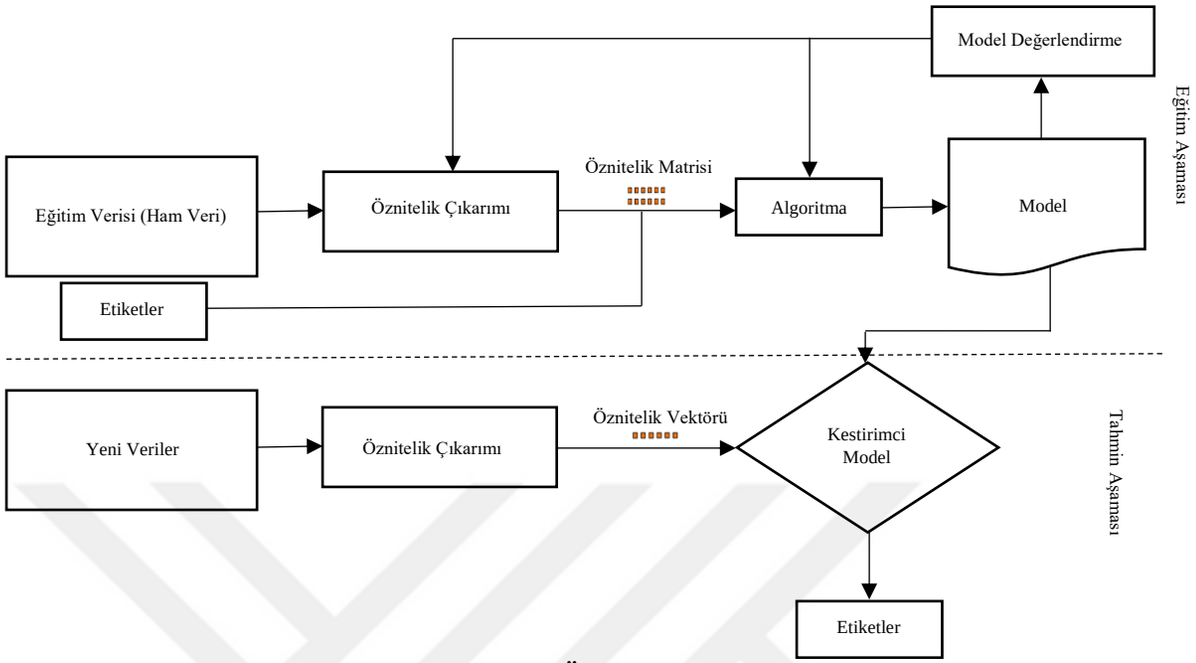


Şekil 6. Denetimli Öğrenme Algoritmaları

Şekil 6'dan da görüldüğü üzere denetimli öğrenme; sınıflandırma ve regresyon olmak üzere iki farklı şekilde uygulanabilmektedir. Lojistik regresyon, karar ağaçları, destek vektör makineleri, k-en yakın komşu, Naive Bayes, rastgele orman ve yapay sinir ağları gibi algoritmalar sınıflandırma algoritmaları olarak kullanılmaktadır.

Sınıflandırma algoritmalarından bazıları ise (örneğin; karar ağaçları, destek vektör makineleri, k-en yakın komşu, rastgele orman, yapay sinir ağları) regresyon algoritmalarında da kullanılabilmektedir. Denetimli öğrenme algoritmaları; literatürde de sıklıkla vurgulandığı şekilde e-posta spam tespiti gibi sınıflandırma problemleri (Dada vd., 2019; Jain ve Sharma, 2022; Nicolas, 2022; Alsuwit, Haq ve Aleisa, 2024) ile hisse senedi veya konut fiyatlarını tahmin etme (Nti, Adekoya ve Weyori, 2020; Chen, 2023; Milosavljevic, Nedovic ve Spasenic, 2024; Chen vd., 2024) gibi sürekli değerleri tahmin eden regresyon problemlerine odaklanmaktadır. Denetimli öğrenmede sınıflandırma görevi; etiketli verinin sınıflara bölünerek eğitilmesini ve verinin ilgili sınıflara atanmasını ifade etmektedir. Süreç, bir dizi etiketli eğitim verisinden modelin öğrenmesini ve daha sonra modelin, etiketlenmemiş veriler için sınıf etiketini tahmin etmek üzere uygulanmasından oluşmaktadır (Taye, 2023). Regresyon görevi ise, sınıf etiketinin sürekli bir değişken olduğu sınıflandırmanın genelleştirilmesidir (Sen, Hajra ve Ghosh, 2020). Tüm makine öğrenmesi modellerinin başarımında olduğu gibi,

denetimli öğrenmede de başarım; yüksek kalitede etiketli verilerin ve uygun öznitelik seçiminin kullanılabilirliğine bağlıdır.



Şekil 7. Denetimli Öğrenme Akış Diyagramı
Kaynak: (Nguyen vd., 2017, s. 3)

Şekil 7'de, denetimli öğrenme algoritmasının akış süreci gösterilmektedir. Birinci adımda; ham eğitim verisi ön işleme yoluyla temizleme ve ölçeklendirme işlemlerinden sonra, kullanılabilir öznitelikler arasından faydalı olanların seçilebilmesi için öznitelik çıkarımına tabi tutulur ve eğitim veri kümesi ile test veri kümesi olmak üzere ayrılır. Ardından, etiketler ile çıkarılan öznitelikler; girdileri istenen çıktılara eşleştiren iyi bir model belirlemek için makine öğrenmesi algoritmalarının kullanıldığı eğitim aşamasında işlenir. Değerlendirme aşaması ise model doğruluğunu iyileştirmek ve ayarlamak[85] için öznitelik çıkarımı ve öğrenme aşamalarına geri bildirim sağlar. Eğitim süreci, istenen doğruluk düzeyine ulaşılana kadar epochlar[86] aracılığıyla tekrarlanır. Model son olarak, tamamen eğitildikten sonra daha önceden hiç görmediği yeni verilerin etiketini tahmin etmek için kullanılır.

Denetimli öğrenme algoritmaları; arşiv biliminde uzun süreli korumada risk profilinin çıkarılması, otomatik metaveri veya belge sınıflandırması, dijitalleştirilmiş koleksiyonlarda yinelenen öğelerin tespit edilmesi ve dijital arşiv materyallerine erişim kısıtlaması tahmini gibi uygulamalarda kullanılabilmektedir.

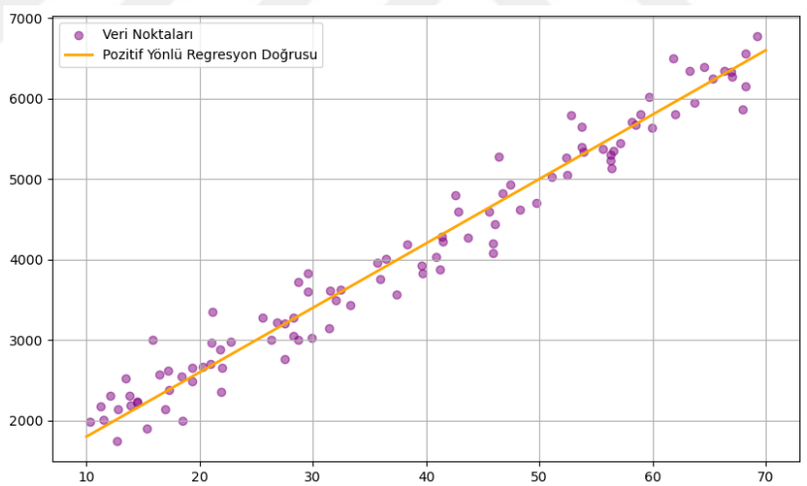
[85] Fine-tuning
[86] Devir

2.8.1.1.1. Doğrusal Regresyon

Doğrusal regresyon, bağımlı bir değişken (Y) ile bağımsız bir değişken (X) arasındaki doğrusal ilişkiyi açıklamak için kullanılan istatistiksel bir tekniktir (Jayasudha, 2020). Yöntemin arka planı; her çıktının, girdisinde bir ilişkiye sahip olmasıdır. Bu nedenle, bağımlı ve bağımsız değişkenlerin belirlenmesinden önce, X'in Y değerini etkileyip etkilemediğini anlaşılmalıdır (Schneider, Hommel ve Blettner, 2010). Modelin temel formülü şu şekildedir:

$$Y = \beta_0 + \beta_1 X + \varepsilon \tag{2.1}$$

Formülde Y, bağımlı değişkeni, başka bir deyişle tahmin edilmek istenen değeri; X bağımsız değişkeni, yani tahmin için kullanılan değeri; β_0, sabit terimi, yani regresyon doğrusunun Y eksenini kestiği noktayı; β_1, eğilim katsayısını, yani X bağımsız değişkenindeki bir birimlik artışın Y bağımlı değişkeni üzerindeki etkisini; ε, hata terimini, yani modelin tahmin edemediği rastgele sapmaları ifade etmektedir. β_0, $X = 0$ olduğunda, Y'nin beklenen değeri ortaya çıkar. β_1, bağımsız değişkendeki bir birimlik artışın, bağımlı değişken üzerinde oluşturduğu ortalama değişimi ifade etmektedir. Örneğin, $\beta_1 = 3$ ise, X'teki her bir birimlik artış, Y'de ortalama 3 birimlik bir artış anlamına gelir.



Şekil 8. Basit Doğrusal Regresyon Grafiği

Şekil 8'de, Python programlama dilinde çizilen örnek bir pozitif yönlü basit doğrusal regresyon grafiği gösterilmektedir. Bu grafikte, x ekseni ile y ekseni arasında bulunan 100 veri noktasının dağılımı ile doğrusal bir ilişkiye sahip bu dağılımların oluşturduğu regresyon doğrusu pozitif bir ilişki içerisinde görülebilmektedir. Burada regresyon doğrusunu gösteren turuncu çizgi, modelin tahmin ettiği doğrusal ilişkiyi temsil etmektedir, başka bir deyişle x değeri arttıkça y değeri de artmaktadır.

Bir diğer tür doğrusal regresyon modeli ise çok değişkenli doğrusal regresyon olarak bilinen, birden fazla değişkenin olduğu türü ifade etmektedir. Çok değişkenli doğrusal regresyon modelinde, bir bağımlı değişkenin (Y) birden fazla bağımsız değişken (X_1, X_2, … , X_n) ile olan doğrusal ilişkisi modellenir (Jankovic, 2022). Modelin temel formül şu şekildedir:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \varepsilon \quad (2.2)$$

Formülde Y, bağımlı değişkeni; β_0, modelin sabit terimini; $\beta_1\ \beta_2, \ldots . \beta_n$, modelin regresyon katsayılarını, yani her bir bağımsız değişkenin Y üzerindeki etkisini; X_1, X_2, ... X_n, bağımsız değişkenleri, ε, modelin hata terimini, yani modelin açıklayamadığı varyansı ifade etmektedir. Çok değişkenli doğrusal regresyon modeli, yukarıdaki gösterimden başka, matris yapısında da gösterilerek formüle edilebilir:

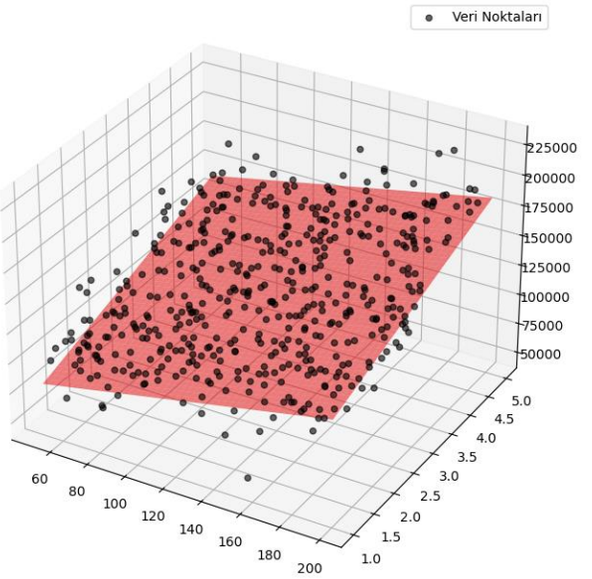
$$Y = X\beta + \varepsilon \quad (2.3)$$

Formülde Y, $n\ x\ 1$ boyutundaki bağımlı değişken vektörünü; X, $n\ x\ (p + 1)$ boyutundaki bağımsız değişkenler matrisini; β, $(p + 1)\ x\ 1$ boyutundaki regresyon katsayıları vektörünü ve ε ise $n\ x\ 1$ boyutundaki hata terimi vektörünü ifade etmektedir.

Basit doğrusal regresyonda olduğu gibi çoklu doğrusal regresyonda da bağımlı değişkenin tahmin edilmesi ve bağımsız değişkenlerden hangilerinin bağımlı değişkeni ne kadar etkilediğini bulmak ve her iki tür değişkenin de aralarında olan ilişkiyi tanımlama amacı bulunmaktadır (Osborne ve Waters, 2002).

Çoklu doğrusal regresyon modelinde; bağımlı ve bağımsız değişkenler arasındaki doğrusal ilişki, gözlem noktalarının birbirinden bağımsız olması, bağımsız değişkenlerin tüm seviyelerinde hataların eş varyanslılığı, hataların normal dağılımı ve bağımsız değişkenlerin birbirleriyle minimum ilişkili olması (başka bir deyişle çoklu doğrusallığın[87] olmaması) gibi temel varsayımlar bulunmaktadır (Özdemir ve Esin, 2007; Williams, Grajales ve Kurkiewicz, 2013).

[87] Multicollinearity



Şekil 9. Çok Değişkenli Doğrusal Regresyon Grafiği

Şekil 9'da, Python programlama dilinde oluşturulan örnek çok değişkenli doğrusal regresyon grafiği gösterilmektedir. Bu grafikte her bir siyah nokta, gözlemlenen veri noktasını temsil etmektedir. Noktalar, gerçek verilerin bağımsız değişkenlere göre nasıl değiştiğini göstermektedir. Kırmızı düzlem, değişkenler arasındaki öngörülen ilişkiyi modelleyen regresyon düzlemi olup çok değişkenli regresyon denkleminden türetilmiştir. Düzlem, farklı X_1 ve X_2 bağımsız değişkenlerine dayalı beklenen Y değerini görsel olarak temsil etmektedir. Başka bir deyişle, her iki bağımsız değişkenin birleşik etkisini göstermektedir.

2.8.1.1.2. Lojistik Regresyon

Lojistik regresyon ikili (0,1) bir değişkeni, öngörücü[88] olarak adlandırılan bir veya daha fazla değişkene dayalı modellemek için kullanılmaktadır (Hilbe, 2015). Modellenen ikili değişken; genellikle bağımlı değişken olarak adlandırılmaktadır. Bağımlı değişken (çıktı) sürekli olmaktan ziyade kategorik olduğunda kullanılmaktadır. Bir lojistik regresyon modelinin verilere iyi uyum sağlayabilmesi için; öngörücülerin ve gözlem noktalarının birbiriyle herhangi bir ilişkiye sahip olmaması ve bağımlı değişkenle önemli ölçüde ilişkili olması gibi temel varsayımlar bulunmaktadır (Harris, 2021).

Lojistik regresyonda modele; bir lojistik fonksiyon (sigmoid fonksiyon olarak da bilinir) uygulanarak, özellikler (giriş değişkenleri) ile ikili hedef değişken arasındaki ilişki öğretilir (Wang vd., 2019). Sigmoid fonksiyon, olasılıksal olarak yorumlanabilen 0 ile 1 arasında bir değer üretir. Fonksiyonun temel formülü şu şekildedir:

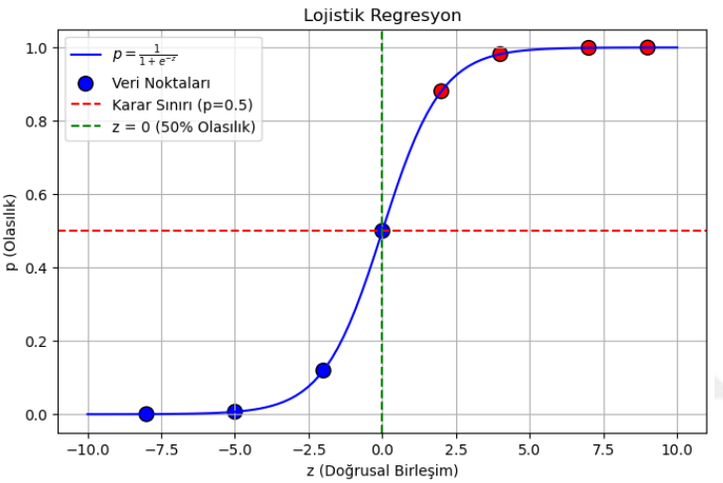
[88] Predictor

$$p = \frac{1}{1 + e^{-z}} \tag{2.4}$$

Formülde p, çıktının sınıf 1 (pozitif sınıf) olma olasılığını; e, Euler sayısını[89]; z, modele giriş özelliklerinin doğrusal birleşimini ifade etmektedir ve şu şekilde formüle edilir:

$$z = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_b x_n \tag{2.5}$$

Formülde b_0, önyargı (bias) değerini; $b_1, b_2, \ldots b_n$, verilerden öğrenilen katsayıları (ağırlıklar); $x_1, x_2, \ldots, x_n$ ise bağımsız değişkenlerin giriş özelliklerini ifade etmektedir. Sigmoid fonksiyonu, z değerini 0 ile 1 arasındaki bir aralığa sıkıştırır. Bu çıktı, girdinin 1. sınıfa ait olma olasılığını temsil eder. Eğer $p > 0{,}5$ ise, 1. sınıf (pozitif sınıf); $p \leq 0{,}5$ ise 0. sınıf (negatif sınıf) demektir.



Şekil 10. Lojistik Regresyon Grafiği

Şekil 10'da sigmoid fonksiyonu kullanılarak oluşturulan lojistik regresyon grafiği gösterilmektedir. Grafikte mavi eğri, sigmoid fonksiyonunu temsil etmektedir. Fonksiyon, herhangi bir gerçek sayı z'yi (x ekseni) 0 ile 1 arasında bir olasılık olan p'ye (y ekseni) eşler ve z değeri arttıkça olasılık 1'e (pozitif sınıf), azaldıkça da olasılık 0'a (negatif sınıf) yaklaşır. Böylece fonksiyon S şeklinde bir yapıya bürünür. Yani z=0'a yakın küçük değişimler büyük bir etkiye sahipken, uç değerler daha çok düzleşme eğilimindedir. Kesikli yatay kırmızı çizgi; p=0,5'teki karar sınırını temsil etmektedir. Eğer p>0,5 ise, model sınıf 1'i (pozitif sınıf) tahmin eder. p≤0,5 ise, model sınıf 0'ı (negatif sınıf) tahmin eder. Bu sınır, ikili sınıflandırma görevlerinde anahtar niteliği taşımaktadır. Dikey kesikli yeşil çizgi ise; modelin p=0,5'i tahmin ettiği noktadır. Bu nokta, sonucun 0 veya 1 olarak sınıflandırma olasılığının eşit olduğu noktaya karşılık gelir. Lojistik regresyonda, temel olarak karar sınırı z=0'da sabitlenir ve bu

[89] ~2.71

değer $\frac{1}{2}$'ye karşılık gelir. Grafikte mavi ve kırmızı ile gösterilen veri noktaları ise; sigmoid eğrisi üzerine çizilen gerçek gözlemleri temsil etmektedir. Grafikteki kırmızı renkli veri noktaları, 0,5 çizgisinin üzerinde olduğundan dolayı birinci sınıfa ait olarak görülmektedir. Mavi veri noktalarının ise 0,5 çizgisinin altında kaldığından dolayı 0. sınıfa ait olduğu görülmektedir.

2.8.1.1.3. Karar Ağaçları

Sayıcı'ya göre (2013, s. 4) karar ağaçları; bir karar vericinin karşılaştığı problem hakkında sahip olduğu bilgiyi oldukça anlaşılır biçimde gösteren, karar seçeneklerini ve olasılıklara bağlı durumları belirli bir düzen içerisinde sıralayıp ortaya koyan görsel bir modelleme yöntemidir. Hem sınıflandırma hem de regresyon görevleri için kullanılan bir algoritmadır. Karar ağaçları tabanlı modeller veriyi öznitelik değerlerine göre yinelemeli bölerek her bir iç düğümün bir özniteliğe dayalı bir kararı ve yaprak düğümlerinin son tahminlerini temsil ettiği ağaç yapısı oluşturur.

Sınıflandırma görevlerinde kullanılan karar ağacı, ayrık sınıf etiketlerini tahmin etmede kullanılmaktadır. Burada amaç, her adımda verileri mümkün olduğunca saf hale getiren bölünmeler oluşturmaktır. Her bir düğümdeki en iyi bölünmeyi gerçekleştirebilmek için entropi (bilgi kazanımı) ve gini indeksi gibi safsızlık[90] ölçütleri kullanılmaktadır.

Entropi, bir veri kümesindeki belirsizlik veya safsızlığın bir ölçüsü olup her düğümdeki verilerin nasıl bölüneceğine karar vermek için kullanılmaktadır. Formülü şu şekilde ifade edilmektedir:

$$\mathrm{H(S)} = -\sum_{i=1}^{c} p_i log_2\, p_i \qquad (2.6)$$

Formülde H(S), S veri kümesinin entropi değerini; c, sınıf sayısını, p_i, i sınıfının olasılığını, log_2, 2 tabanında logaritmayı ifade etmektedir. log_2, bilginin bit cinsi olarak ölçüldüğünden dolayı kullanılmaktadır. Bir sınıf çok baskınsa (p_i 1'e yakınsa), o zaman entropi düşüktür. Eğer tüm sınıflar eşit derecede olasılığa sahipse (p_i=0,5, 0,5) entropi yüksektir.

Entropi, bölünmeden sonra belirsizliğin ne kadar azaldığını ölçmekte ve formülü şu şekilde ifade edilmektedir:

$$\mathrm{IG(S, A)} = \mathrm{H(S)} - \sum_{v \in A} \frac{|S_v|}{|S|}\, H(S_v) \qquad (2.7)$$

[90] Impurity

Formülde $\mathrm{IG(S, A)}$, S veri setinin A özelliğini kullanarak bölündüğündeki bilgi kazanımını; $\mathrm{H(S)}$, tüm S veri setinin bölünmeden önceki entropisini; A, veri kümesini bölmek için kullanılan bir özniteliği; v, A özniteliği içerisindeki benzersiz bir değeri veya kategoriyi; $\frac{|S_v|}{|S|}$; S_v alt kümesindeki veri noktalarının oranını; $H(S_v)$ ise s_v alt kümesinin entropi değerini ifade etmektedir.

Gini indeksi ise bir diğer safsızlık ölçütüdür. Düğümlerin nasıl bölüneceğine karar verilmesi ve bir kaynağın bir popülasyonda ne kadar eşit bir şekilde dağıtıldığının ölçülmesinde kullanılmaktadır (Farris, 2010, s. 851). İndeks ayrıca, bir veri kümesindeki sınıfların ne kadar "karışık" olduğunu ortaya çıkarır. Formülü şu şekilde ifade edilmektedir:

$$\mathrm{Gini(S)} = 1 - \sum_{i=1}^{c} p_i^2 \tag{2.8}$$

Formülde $\mathrm{Gini(S)}$, veri seti S'nin gini safsızlık değerini; c, veri setindeki toplam sınıf sayısını; p_i, i sınıfındaki örneklerin oranını ifade etmektedir. Daha düşük gini indeksi, modelde daha saf düğümler olduğu anlamına gelmektedir.

Gini indeksi ve entropi genellikle benzer sonuçlar vermektedir ancak gini indeksinin hesaplanması daha hızlı gerçekleşir (Saud, Shakya ve Neupane, 2021).

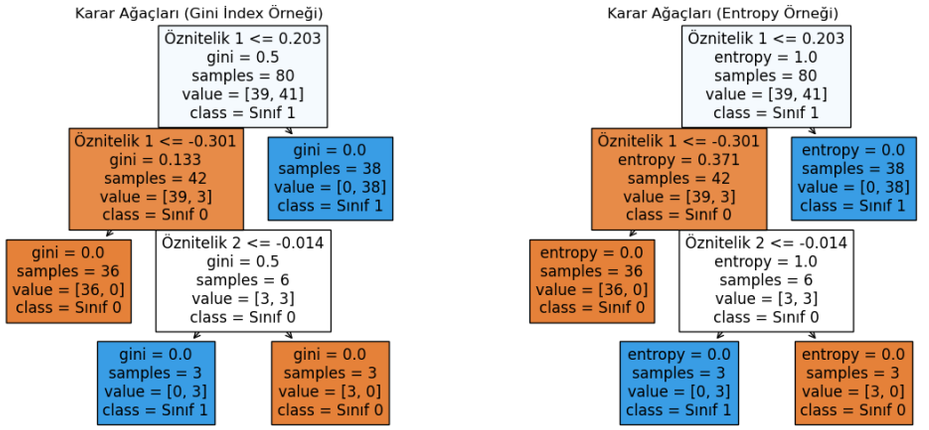
Bir veri seti bir özelliğe göre bölündüğünde, her bir alt düğüm için gini safsızlık değerlerinin ağırlıklı toplamı hesaplanabilmektedir. Burada daha büyük alt kümeler nihai safsızlığa daha fazla katkıda bulunur. Daha düşük bölünmüş gini değeri, daha saf alt kümeler oluşturduğu için daha iyi bir bölünme olduğu anlamına gelir. En iyi bölünme ise, gini bölünmesini en aza indiren değerdir. Formülü şu şekilde ifade edilmektedir:

$$Gini_{split} = \sum_{v \in A} \frac{|S_v|}{|S|} \, Gini(S_v) \tag{2.9}$$

Formülde A, seçilen özniteliğin olası değerlerinin bir kümesini; $|S_v|$, alt kümedeki örnek sayısını; $|S|$, bölünmeden önceki toplam örnek sayısını; $Gini(S_v)$ ise S_v alt kümesinin gini safsızlık değerini ifade etmektedir. Her bir alt kümenin gini safsızlık değeri; orijinal veri kümesine göre büyüklüğü ile ağırlıklandırılır. Böylece, tıpkı bilgi kazanımı gibi, gini kazanımı değeri de bulunabilir. Formülü şu şekildedir:

$$Gini_{gain} = Gini(S) - Gini_{split} \tag{2.10}$$

Formülde $Gini(S)$, bölünmeden önceki gini indeks değerini; $Gini_{split}$ ise bölünmeden sonraki alt düğümlerin gini indeks değerlerinin ağırlıklı toplamını ifade etmektedir.

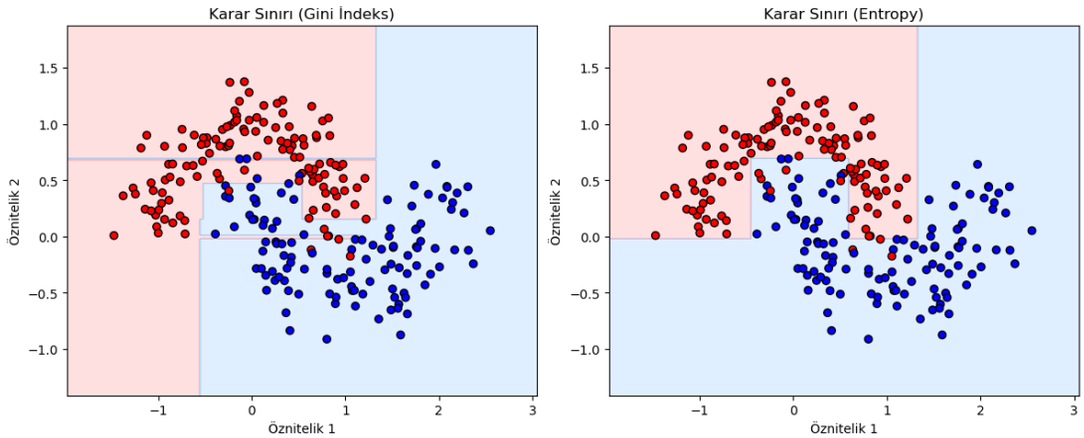


Şekil 11. Karar Ağaçları Grafiği

Şekil 11'de örnek karar ağaçları algoritması grafiği, gini indeksi ve entropi kullanılarak gösterilmektedir. Sol tarafta bulunan gini indeksi hesaplanarak oluşturulmuş karar ağacı grafiğinde amaç; en iyi özellik bölünmesini seçerek safsızlığı en aza indirmektir. Burada her bir dikdörtgen düğüm, bir özelliğin bölündüğü bir karar noktasını temsil etmektedir. Her düğümdeki gini skoru, o alt kümenin safsızlık derecesini gösterir. Daha düşük gini skoru (0 veya 0'a yakın) daha saf sınıflar anlamına gelir. Daha yüksek gini skoru (0,5 ve 0,5'e yakın) daha fazla safsızlık (yani daha karışık sınıflar) anlamına gelmektedir. Örnekler (samples), o düğümdeki veri noktalarının sayısını göstermektedir. Değer (value), her sınıftan alınan örnek sayısını gösterir. Her düğümün altındaki sınıf (class) etiketi ise, o bölünme için tahmin edilen sınıfı temsil etmektedir. Entropide ise temel amaç; en yüksek bilgiye göre en iyi öznitelik bölünmesinin seçilmesidir. Yapılanma, gini indeks ağacına benzer, ancak gini skorları yerine, düğümlerde entropi değerleri bulunur. Entropi değeri 0 ile 1 arasında değişmektedir. 0 tamamen saf bir sınıf (mükemmel bölünme) ve 1 maksimum safsızlık anlamına gelir (sınıfların eşit dağılımı). Bilgi kazanımı her bölünmede ayrı hesaplanır. Daha yüksek bilgi kazanımı daha iyi bir bölünme anlamına gelmektedir. Entropi, gini indeksine göre sınıf dağılımındaki değişikliklere daha çok duyarlıdır (Gajowniczek vd., 2020).

Entropi ve gini indeksi, bazı durumlarda aynı sonuçları vermektedir, fakat çoğu zaman farklı sonuçlar üretir. Gini indeksi ve entropi arasındaki temel fark; safsızlık derecesinin ölçüm yönteminde ve en iyi bölünmeye nasıl karar verdiğindedir. Veri seti dengesizse, entropi, bilgi kazanımını farklı bir şekilde hesapladığı için farklı bir sınır üretebilir. Gini indeks hesaplamalarında logaritmik dönüşümler yapılmadığından dolayı daha hızlı sonuçlar

üretilirken, entropide logaritma kullanıldığından dolayı ise daha yavaş sonuçlar üretilmektedir (Biró ve Néda, 2020).



Şekil 12. Karar Sınırları Grafiği

Şekil 12'de ise karar ağaçlarındaki karar sınırı grafiksel olarak gösterilmektedir. Grafiğin sol tarafında bulunan gini indeksi hesaplamalı karar sınırı gösteriminde karar bölgeleri, her düğümdeki gini safsızlık değeri en aza indirerek oluşturulmuştur. Sağdaki grafikte entropi hesaplamalı grafik bulunmaktadır. Burada ağaç yapısı, bilgi kazanımını en üst düzeye çıkaran bölünmeleri seçmektedir, başka bir deyişle ortaya çıkan alt kümelerin saflığı daha dikkatli bir şekilde ele alınmaktadır. Karar sınırı bazı bölgelerde gini skoruna göre daha pürüzsüz veya farklı olabilir. Gini indeksinin çoğunluk sınıfı ön plana alabileceği durumlarda, entropi hesaplamalarında sınıf dağılımları daha dengeli bir şekilde ele alınır. Veri kümesi iyi dengelenmişse, her iki hesaplama türü de çok benzer karar sınırları verebilir. Sınıf dağılımları normalden farklı ise entropi, giniye kıyasla daha farklı bir karar sınırı üretebilir. Gini indeksi, baskın sınıflara doğru biraz daha önyargılı bölünmeler üretme eğilimindedir, fakat entropi genel belirsizliği daha kapsamlı bir şekilde ele alır. Veri kümesindeki dengesizlik ileri boyutlarda ise, entropi giniye kıyasla daha sade bir bölünme üretmektedir.

Karar ağaçları aynı zamanda regresyon görevlerinde de kullanılan bir denetimli öğrenme algoritması türüdür. Bir regresyon ağacı, sürekli değerleri (örneğin, hisse senedi fiyatları) tahmin eder. Algoritmanın temel amacı; veri setini homojen alt kümelere yinelemeli olarak bölerek sayısal sonuçları tahmin etmeyi ve tahmin hatasını en aza indirmek için en iyi bölünmeyi bulmaktır (Breiman vd., 1984).

Regresyon ağacı; bir kök düğüm, iç karar düğümleri ve uç yaprak düğümlerinden oluşmaktadır. Her bir iç düğüm, bir bölme kriterine dayalı bir kararı temsil ederken, yaprak

düğümleri tahmin edilen sayısal değerleri içerir. Algoritmada verileri bölmek için ikili[91] yinelemeli bölme yaklaşımı kullanılır. Burada amaç; her bölmedeki varyansı en aza indirmek ve her alt kümedeki hedef değişkenin homojenliğini en üst düzeye çıkarmaktır (Rokach ve Maimon, 2005).

Bir regresyon ağacının temel amacı, veri setini, hedef değişkenin varyansının en aza indirildiği bölmelere bölmektir. Bölme için kullanılan en yaygın kriter, kare hatalarının toplamını (SSE[92]) en aza indirme formülüne dayanır (Mienye ve Jere, 2024).

X_i'nin öznitelik vektörlerini ve y_i'nin karşılık gelen hedef değerleri olduğu bir veri kümesi $D = \{(X_i, y_i)\}_{i=1}^{N}$ verildiğinde, bölme kuralı, aşağıdaki formülün çıktısını en aza indiren öznitelik x_i üzerinde bir eşik s bulmayı amaçlar:

$$\sum_{i:X_i \in R_1(s)} (y_i - \bar{y}_1)^2 + \sum_{i:X_i \in R_2(s)} (y_i - \bar{y}_2)^2 \tag{2.11}$$

Formülde $R_1(s)$ ve $R_2(s)$, $x_i = s$ noktasında bölünerek oluşan iki bölgeyi; $\bar{y}_1$ ve $\bar{y}_2$ sırasıyla R_1 ve R_2 bölgelerindeki y'nin ortalama değerlerini ifade etmektedir.

Yeni bir girdi X için tahmin, X'in bulunduğu uç düğüm (yaprak) belirlenerek yapılır. Çıktı ise, o yapraktaki tüm hedef değerlerin ortalaması olup formülü şu şekilde ifade edilmektedir:

$$\hat{y} = \frac{1}{|R_m|} \sum_{i:X_i \in R_m} y_i \tag{2.12}$$

Formülde $\hat{y}$, girdi X için hedef değişkenin tahmin edilen değerini; R_m, X'in içinde bulunduğu bölgeyi (yaprak düğüm); $|R_m|$, o bölgedeki gözlem sayısını; y_i ise R_m içerisindeki hedef değişkenin gerçek değerlerini ifade etmektedir. Regresyon ağaçları sürekli değerleri tahmin ettiğinden, $\hat{y}$ değeri, ayrı bir sınıf etiketi yerine gerçek bir sayı olur.

Karar ağaçları, aşırı öğrenmeye[93] oldukça meyillidir. Bir karar ağacı, özellik alanını yinelemeli olarak bölümlendirme doğası gereği aşırı öğrenmeye kolayca yatkın haline gelebilir (Bramer, 2016, s. 121). Başka bir deyişle; bir karar ağacı modeli aşırı karmaşık hale geldiğinde, eğitim verilerindeki gürültüyü veya alakasız örüntüleri yakaladığında ve görülmeyen verilere genellemeyi iyi yapamadığında aşırı öğrenme meydana gelir (Sani, Ben-Musa ve Haladu, 2021). Ağaç yapısı çok derin bir şekilde büyüdüğünde ve çok az gözlem içeren küçük yaprak

[91] Binary
[92] Sum of Squares Errors
[93] Overfitting

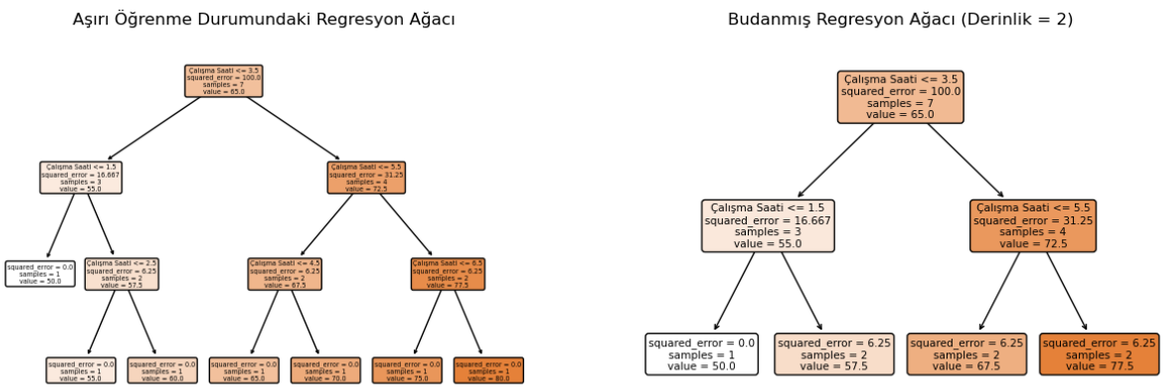
düğümleri oluşturduğunda da aşırı öğrenme meydana gelmektedir. Böylece modelin başarımı; görülmeyen verilere yönelik zayıf genellemeyle sonuçlanır.

Aşırı öğrenmeyi engelleyebilmek için, budama tekniklerinden maliyet-karmaşıklık budaması tekniği kullanılmaktadır. Önce tamamen genişletilmiş bir ağaç yapısı oluşturulduktan sonra doğruluk skoruna çok az katkıda bulunan dallar kaldırılarak aşırı öğrenmenin önüne geçilmiş olur. Budama süreci, maliyet-karmaşıklık fonksiyonunu kullanarak model karmaşıklığı ve hata azaltma arasında bir takas içerir (Ying, 2018). Burada; ağaç karmaşıklığını ve doğruluğunu dengelemek için bir düzenleme parametresi (a) eklenir. Formül şu şekilde ifade edilmektedir:

$$C(T) = \sum_{m=1}^{M} \sum_{i=X_i \in R_m} (y_i - \bar{y}_m)^2 + a|T| \qquad (2.13)$$

Formülde $C(T)$, toplam maliyeti (hata+karmaşıklık cezası); M, terminal düğümlerinin sayısını; R_m, her bir bölgeyi; $\bar{y}_m$, R_m bölgesindeki ortalama hedef değeri; a, budama kuvvetini kontrol eden bir düzenlileştirme parametresini; $|T|$ ise yaprak düğümlerinin sayısını ifade etmektedir. Burada a değeri arttıkça, gereksiz bölünmeler budandığı için ağaç yapısı küçülür ve aşırı öğrenme dizginlenmiş olur.

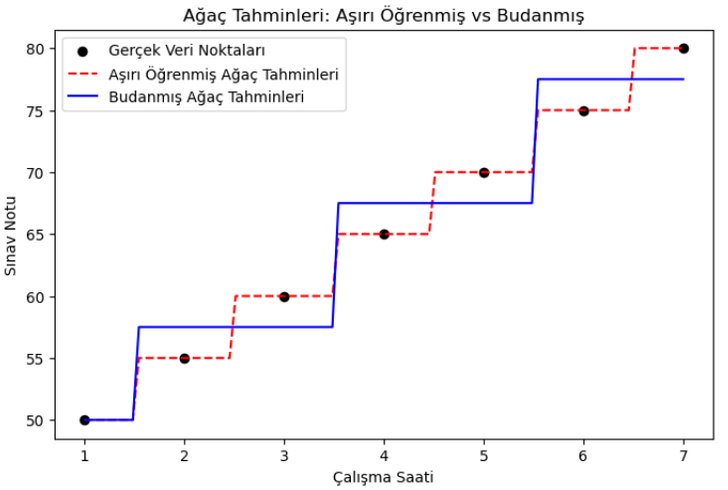
Regresyon ağaçları parametrik olmayan bir yöntemdir. Güçlü yönleri arasında doğrusalsızlık ve etkileşimleri doğal olarak ele almaları yer alırken; aşırı öğrenmeye karşı hassastır ve budama, iyileştirilmiş performans için olmazsa olmaz bir koşuldur (Elith, Leathwick ve Hastie, 2008).



Şekil 13. Aşırı Öğrenmiş ve Budanmış Regresyon Ağaçları Grafiği

Şekil 13'te aşırı öğrenme durumunda olan bir regresyon ağacı örneği (solda) ile aşırı öğrenme durumundan maliyet-karmaşıklık fonksiyonu kullanılarak budanmış ağaç yapısına geçen (sağda) örnek bir gösterim bulunmaktadır. Aşırı öğrenmiş ağaç yapısı birçok dala sahipken,

sağdaki budanmış ağaç yapısı daha az dala sahiptir. Ayrıca, budanmış ağaç yapısının, tam derinlikli ağaç yapısından daha küçük bir derinliğe sahip olduğu da görülmektedir.



Şekil 14. Aşırı Öğrenmiş ve Budanmış Regresyon Ağaçlarında Tahmin

Şekil 14'te ise aşırı öğrenmiş regresyon ağacının, eğitim verilerini neredeyse bire bir takip ederek oldukça keskin adımlar oluşturduğu görülmektedir. X değişkenindeki her benzersiz değerin kendine özgü tahmini bir ŷ değeri vardır ve bu değer verilere neredeyse tam doğru bir şekilde uyan parça parça sabit bir fonksiyon oluşturur. Fakat buradaki temel sorun; varyans yüksekliğidir ve model gürültüyü yakaladıktan sonra görülmemiş verilere iyi bir genelleme yapamaz. Budanmış ağaç yapısı, her bir noktayı tam olarak uydurmak yerine benzer veri noktalarını daha geniş bölgelere gruplandırır. Böylece aşırı bölünmeler oluşmaz ve bu da modeli eğitim verilerindeki küçük dalgalanmalara karşı daha az hassas hale getirir. Başka bir deyişle, kırmızı kesikli çizgi (aşırı öğrenmiş ağaç yapısı) veri noktalarını yakından takip ederken, mavi düz çizgi (budanmış ağaç yapısı) daha pürüzsüz ve genelleştirilmiştir. Budama işlemi, bir miktar önyargı getirir ancak varyansı azaltarak test doğruluğunu artırır ve bu durum daha önemli görülmelidir (Esposito vd., 1997).

2.8.1.1.4. Destek Vektör Makineleri

Destek vektör makineleri[94]; Boser, Guvon ve Vapnik (1992) tarafından geliştirilmiştir. Algoritma, Srivastava ve Bhambhu'ya göre (2010, s. 1) genellikle diğer sınıflandırma algoritmalarına göre daha yüksek doğruluk performansı sağlamaktadır. Başlangıçta ikili sınıflandırma için geliştirilen SVM'ler, çok sınıflı sınıflandırma ve regresyon problemlerini ele almak üzere genişletilmiştir. Metin kategorizasyonu (Joachims, 1998), el yazısıyla yazılmış rakam tanıma (Al-Hamadani, 2023), ton tanıma (Liu, Wang ve Wang, 2016), görüntü

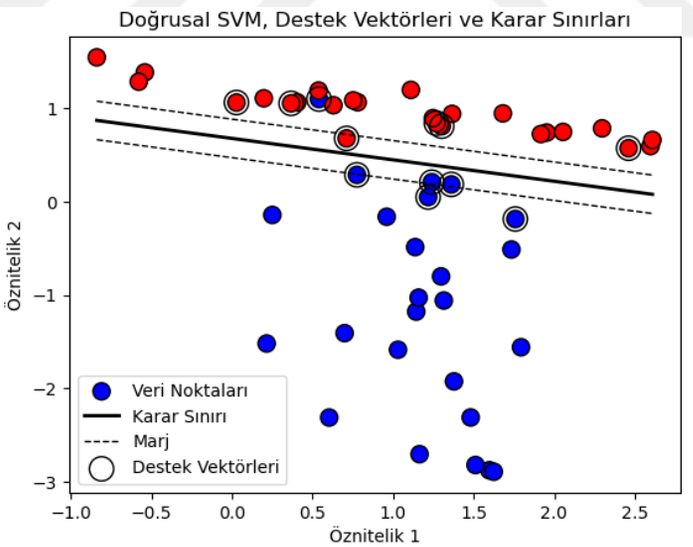
[94] Support Vector Machines - SVM

sınıflandırması (Santosh ve Moguram, 2019), nesne algılama (Kurilová, vd., 2021), genetik veri analizi (Bouslah ve Taleb, 2020) gibi çok çeşitli problemlerde kullanılmıştır. Ancak bazı veri setleri için SVM'nin performansı; maliyet ve çekirdek parametrelerinin nasıl ayarlandığına karşı çok hassastır. Bu hassasiyeti koruyabilmek için çoğunlukla çapraz doğrulama işleminin yapılması gerekmektedir. Çekirdek fonksiyonları, Gauss çekirdeğinin standart sapması, bağıl ağırlıklar ve eğitim örneklerinin sayısı gibi hiperparametre ayarları ile en optimum sonuç bulunmaya çalışılmaktadır.

Sınıflandırma görevlerinde SVM algoritmasının temel amacı; farklı sınıflar arasındaki marjı en üst düzeye çıkaran hiperdüzlemi bulmaktır. Bu hiperdüzlem, her sınıfın en yakın veri noktalarından eşit uzaklıkta olacak şekilde seçilir ve bunlar destek vektörleri olarak bilinir (Awad ve Khanna, 2015). Doğrusal SVM, hem sınıflandırma hem de regresyon görevlerinde kullanılmaktadır. Formülü şu şekilde ifade edilmektedir:

$$f(x) = w \times x + b \tag{2.14}$$

Formülde w, hiperdüzlemi dik kesen ağırlık vektörünü; x, öznitelik vektörünü, yani girdi noktasını; b, önyargıyı ve $f(x)$ ise veri noktalarını sınıflandırmak için kullanılan karar fonksiyonunu ifade etmektedir.



Şekil 15. Doğrusal Destek Vektör Makinesi Sınıflandırma Grafiği

Şekil 15'te SVM'nin sınıflandırma örnek grafiği gösterilmektedir. Bu dağılım grafiğinde; kırmızı ve mavi veri noktalarını iki sınıfa ayıran düz siyah bir çizgi ile her iki tarafta kesikli kenar çizgileri görülmektedir. Kesikli kenar çizgileri her iki yöne doğru uzayan marjı ifade ederken; marj sınırına en yakın veri noktaları ise destek vektörleri olarak adlandırılır.

Hiperdüzlem, sınıflandırma problemlerinde iki sınıfı ayıran karar sınırı olarak tanımlanır. İki boyutlu uzayda bu sınır bir doğru şeklindeyken, üç boyutlu uzayda düzlem hâlini alır. Daha yüksek boyutlu uzaylarda ise hiperdüzlem, uzayın boyut sayısına göre tanımlanan (n−1) boyutlu bir altuzay olarak varlığını sürdürür (Lawan, 2023). Bir hiperdüzlemin denklemi şu şekilde ifade edilmektedir:

$$f(x) = w^T x + b = 0 \tag{2.15}$$

Denklemde w, hiperdüzlemin yönelimini belirlerken, b ise hiperdüzlemi yukarı veya aşağı yönlerde kaydırır. w ağırlık vektörü, karar sınırının yönünü tanımlar. w büyükse, sınıflar arasındaki marj küçük olur; w küçük ise, marj daha büyüktür ve model daha iyi genelleme yapabilir. $b = 0$ olduğunda, hiperdüzlem orijinden geçer; $b > 0$ olduğunda, hiperdüzlem yukarı doğru; $b < 0$ olduğunda ise, hiperdüzlem aşağı doğru kayar. Karar fonksiyonu yeni veri noktalarını sınıflandırmak için kullanılır. Eğer $f(x) > 0$ ise, veri noktası +1. sınıfa aittir. $f(x) < 0$ olduğunda da veri noktası -1. sınıfa aittir. $f(x) = 0$ olduğunda ise, nokta hiperdüzlemde yer almaktadır.

Jun'a göre (2021) marj, karar sınırı (hiperdüzlem) ile en yakın veri noktaları (destek vektörleri) arasındaki mesafe olarak tanımlanabilir. SVM'nin amacı, daha iyi genelleme için marjı maksimize etmektir. Formülü şu şekildedir:

$$\mathrm{M} = \frac{2}{\|w\|} \tag{2.16}$$

Formül 2.16'da görüldüğü üzere ağırlık vektörü w'nun normunun 2'ye bölünmesi ile marj değeri bulunmaktadır. Daha büyük bir marj, daha iyi genelleme ve daha az aşırı öğrenme anlamına gelmektedir. Ağırlık vektörünün büyüklüğü, kenar boşluğunun genişliğini belirler. Ağırlık vektörünün normu ise şu şekilde hesaplanabilir:

$$\|w\| = \sqrt{w_1^2 + w_2^2 + \cdots + w_n^2} \tag{2.17}$$

SVM, farklı sınıflar arasındaki marjı maksimize eden optimum bir karar sınırı bulmayı amaçlar. Veri setine bağlı olarak, iki temel yaklaşım bulunmaktadır: Sert marjlı SVM ve yumuşak marjlı SVM.

Sert marjlı SVM, veri seti mükemmel bir şekilde doğrusal olarak ayrılabilir olduğunda, yani herhangi bir yanlış sınıflandırma olmadan tüm veri noktalarını sınıflandırabilen bir hiperdüzlem mevcut olduğunda uygulanabilir. Sert marj, veri setinin sınıfa göre doğrusal olarak ayrılabilir olduğunu varsayar (Santana vd., 2022). Bu durumda temel amaç, marjı (yani, hiperdüzlem ile en yakın veri noktaları arasındaki mesafeyi) maksimize etmek ve hiçbir veri

noktasının marj içinde veya hiperdüzlemin yanlış tarafında olmamasını sağlamaktır. Optimizasyon problemini çözen formül şu şekilde ifade edilebilir:

$$min_{w,b} \frac{1}{2} \|w\|^2 \quad (2.18)$$

Formülde w, hiperdüzlemin yönelimini tanımlayan ağırlık vektörü ve b ise önyargıdır. Formülün bu kısmı, sınıfları ayıran en büyük farkın bulunmasını amaçlar.

Her bir veri noktasının doğru şekilde sınıflandırıldığından emin olunabilmesi için şu kısıtlamalar uygulanmalıdır:

$$y_i \, (w \times x_i + b) \geq 1, \quad \forall_i \quad (2.19)$$

Formülde y_i, veri noktası i'nin sınıf etiketidir ve +1 (bir sınıf için) veya -1 (diğer sınıf için) değerlerini alır. x_i, i. veri noktasının öznitelik vektörünü; w, karar sınırının yönelimini tanımlayan ağırlık vektörünü, b ise karar sınırını kaydıran önyargıyı ifade etmektedir. Sınıflandırma koşulu olarak $y_i \, (w \times x_i + b)$ terimi, her bir veri noktası için işlevsel marj olarak kabul edilir. Doğru sınıflandırılmış noktalar için:

- Nokta, +1 sınıfına aitse (yani, $y_i = +1$), $(w \times x_i + b)$ ifadesi pozitif olmalıdır.
- Nokta, -1 sınıfına aitse (yani, $y_i = -1$), $(w \times x_i + b)$ ifadesi negatif olmalıdır.

Marj koşulu olarak ise eşitsizliğin sağ tarafındaki 1 değeri marj sınırını temsil etmektedir. Kısıtlama şunları sağlar:

- Sınıf +1 skoru için; $(w \times x_i + b) \geq 1$ (bu noktalar karar sınırından en az 1 birim uzakta olmalıdır).
- Sınıf -1 skoru için; $(w \times x_i + b) \leq -1$ (bu noktalar karar sınırından en az 1 birim uzakta olmalıdır).

Bu kısıtlamalar; hiçbir veri noktasının kenar boşluğunun içinde olmasına veya yanlış sınıflandırılmasına izin verilmemesini sağlar. Böylece; sınıflar arasındaki kenar boşluğu en üst düzeye çıkarılır ve her bir veri noktası, kendi sınıfına ait marj bölgesinin doğru tarafında yer alır. Sert marjlı SVM modelleri, doğrusal olarak ayrılabilen ve gürültüsüz veri kümelerinde etkili olsa da, gürültülü ya da sınıflar arası örtüşmenin bulunduğu veri kümelerine uygulanamaz (Li vd., 2013).

Yumuşak marjlı SVM gerçek dünya problemlerinde, mükemmel ayrımın her zaman mümkün olmadığını göstermekte ve sert marjlı SVM'nin sınırlamalarını ele almak üzere tasarlanmıştır.

Teknikte, bazı yanlış sınıflandırılmış noktalara esnek değişkenler (ξ_i) atanmaktadır (Marin vd., 2022). Formülü şu şekilde ifade edilmektedir:

$$min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_i \xi_i \tag{2.20}$$

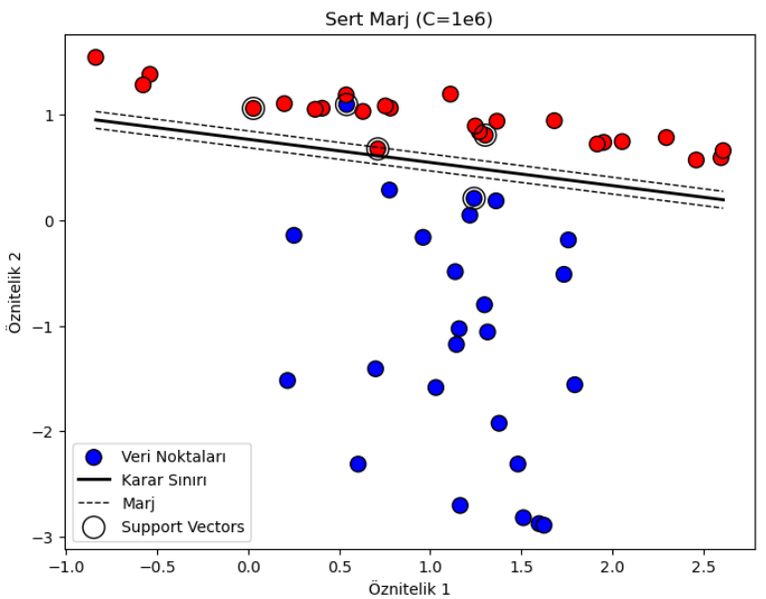
ξ_i, bir noktanın marjı ne kadar ihlal ettiğini ölçen esnek değişkeni[95] temsil etmektedir. C ise marj boyutu ile sınıflandırma doğruluğu arasındaki dengeyi kontrol eden düzenleme parametresini ifade etmektedir. Kısıtlaması ise şu şekilde formülleştirilebilir:

$$y_i (w \times x_i + b) \geq 1 - \xi_i \qquad \forall_i, \xi_i \geq 0 \tag{2.21}$$

Formül, çoğu noktanın doğru sınıflandırılmasını sağlar, ancak gerekirse biraz esneklik (ξi) bırakılabilir:

- $\xi_i = 0$ olduğunda noktalar doğru sınıflandırılmıştır ve marjın dışında kalır.
- $0 < \xi_i \leq 1$ olduğunda da noktalar marjın içinde ama yine de istenilen tarafta kalır.
- $\xi i > 1$ olduğunda ise noktalar yanlış sınıflandırılmıştır, yani hiperdüzlemin yanlış tarafında konumlanmıştır.

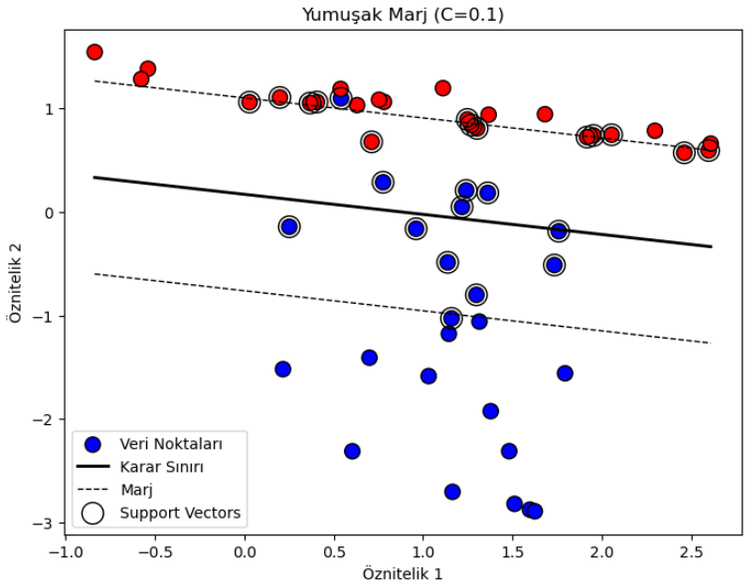
$\xi_i \geq 0$ ifadesi, esnek değişkenin hiçbir zaman negatif olmamasını sağlar. Noktaların marj içinde hareket etmesine veya yanlış sınıflandırılmasına izin verilir ancak bunların ters yöndeki kısıtlamaların ihlal edilmesi de önlenmiş olur.



Şekil 16. Sert Marjlı Destek Vektör Makinesi Sınıflandırma Grafiği

Şekil 16'da sert marjlı SVM grafiği gösterilmektedir. Şekilde de görüldüğü üzere, marj sınırları oldukça dar ve sıkışık bir pozisyondadır.

[95] Slack variable



Şekil 17. Yumuşak Marjlı Destek Vektör Makinesi Sınıflandırma Grafiği

Şekil 17'de gösterilen grafikte ise yumuşak marj hesaplamalarının, marj sınırını son derece genişlettiği ve hatalı verileri de içerisine alacak şekilde sınıflandırmayı yaptığı görülebilmektedir. Algoritmanın amacı esnek sınırlar dahilinde genel yapıyı etkilemeyecek az sayıda hatalı sınıflandırmayı da marj sınırları içinde göstermektir.

Gerçek dünya verilerinin çoğu doğrusal olarak ayrılabilir değildir; bu yüzden iki sınıfı ayırmak için düz bir çizgi (2 boyutta) ya da hiperdüzlem (daha yüksek boyutlarda) yeterli olmaz. Bu sorunu çözmek amacıyla doğrusal olmayan SVM, verileri daha yüksek boyutlu bir öznitelik uzayına dönüştürerek doğrusal bir ayırıcı elde etmeye çalışır. Ancak bu dönüşüm, çekirdek fonksiyonu aracılığıyla veri noktalarının yüksek boyutlu uzaydaki iç çarpımları yapılarak hesaplanır. Bu yönteme çekirdek hilesi[96] denir ve dönüşümün açık biçimde hesaplanmasına gerek kalmadan doğrusal ayrım yapılmasını mümkün kılar (Du vd., 2024).

Doğrusal olmayan şekilde ayrılabilen verilerde, her iki sınıfı doğru şekilde ayıran tek bir doğrusal sınır bulmanın bir yolu yoktur. Doğrusal olmayan SVM'deki temel fikir, orijinal girdi verilerini doğrusal bir ayırıcının bulunabileceği daha yüksek boyutlu bir alana eşlemektir (Mammone, Turchi ve Cristianini, 2009). Orijinal alanda doğrusal bir hiperdüzlem bulmak yerine, önce giriş alanı, $\phi(x)$ fonksiyonu kullanılarak daha yüksek boyutlu bir özellik alanına dönüştürülür. Bu daha yüksek boyutlu alanda, veriler doğrusal olarak ayrılabilir hale gelebilir. Daha sonra dönüştürülmüş alanda doğrusal SVM uygulanır. Orijinal n boyutlu girdi alanını

[96] Kernel trick

daha yüksek m boyutlu bir özellik alanına dönüştürüp verileri kolayca ayrılabilir hale getirmek için bir fonksiyon tanımlanmaktadır: ϕ:$R_n \rightarrow R_m$, m > n olmak üzere,

$$w \times \phi(x) + b \tag{2.22}$$

Formülde $\phi(x)$, x'i daha yüksek boyutlu bir uzaya eşler. Ancak, $\phi(x)$ işlemini yüksek boyutlu uzayda hesaplamak maliyet açısından pahalıdır ve çoğu zaman da uygulanamaz. Bu işlem yerine kullanılan çekirdek hilesi, $\phi(x)$ dönüşümünü gerçekleştirmeden, daha yüksek boyutlu uzayda noktaların hesaplanmasını sağlar. $\phi(x) \times \phi(x')$ ifadesini hesaplamak yerine, bir çekirdek fonksiyonu kullanılır:

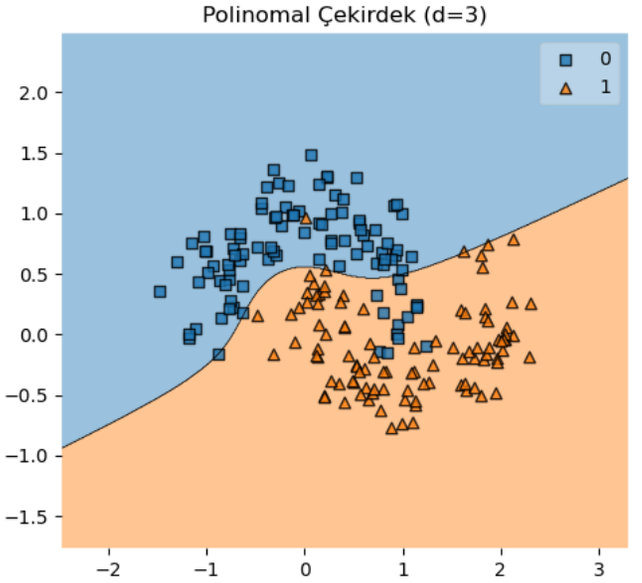
$$K(x, x') = \phi(x) \times \phi(x') \tag{2.23}$$

Böylece $\phi(x)$ ve $\phi(x')$ dönüşümlerini açıkça hesaplanmadan, yalnızca bu dönüşümlerin iç çarpımını temsil eden $K(x, x')$ çekirdek fonksiyonu kullanılarak yüksek boyutlu uzayda sınıflandırma gerçekleştirilir. Polinomal çekirdek, gaussian radyal tabanlı fonksiyon çekirdek (RBF) ve sigmoid çekirdek olmak üzere 3 farklı çekirdek fonksiyonu kullanılmaktadır.

Polinomal çekirdek; iki vektör arasında d dereceli çekirdeğin hesaplanmasını sağlar ve iki vektör arasındaki benzerliği gösterir. Bir polinomal çekirdek, dönüşümü hesaplamadan verileri daha yüksek dereceli bir polinom uzayına eşlemeye olanak tanır (Pradnyana vd., 2021). Örneğin, düz bir çizginin iki sınıfı ayıramadığı bir veri kümesini düşünüldüğünde, veriler polinom fonksiyonları kullanılarak daha yüksek boyutlu bir uzaya yansıtılırsa, sınıfları ayırabilen bir hiperdüzlem bulunabilir. Fonksiyon şu şekilde ifade edilebilir:

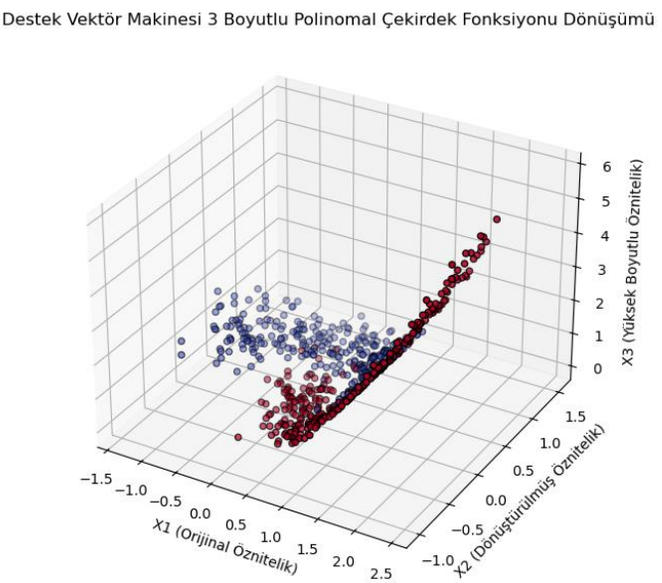
$$K(x, x') = (x \times x' + c)^d \tag{2.24}$$

Formülde x ve x', öznitelik vektörlerini, yani girdi noktalarını; $x \times x'$, iki vektör arasındaki nokta (skaler) çarpımını; c, sabiti (aynı zamanda bir katsayı olarak da adlandırılır ve genellikle karar sınırının esnekliğini kontrol etmek için kullanılır); d ise polinomun derecesini (karar sınırının ne kadar karmaşık olacağını belirler) ifade etmektedir. Doğrusalsızlığı sağlamak için, $x \times x'$ ifadesi d kuvvetine yükseltilir. Eğer c=0 seçilirse, çekirdek fonksiyon, homojen bir polinomal çekirdek hâlini alır. Daha yüksek d değeri model esnekliğini artırır ancak aşırı öğrenme riskine yol açar.



Şekil 18. Polinomal Destek Vektör Makinesi Sınıflandırma Grafiği

Şekil 18'de gösterilen polinomal SVM grafiği, iki sınıfı etkili bir şekilde ayıran eğimli bir karar sınırını göstermektedir. Sınır, veri şekline uyum sağlayarak doğrusalsızlığı etkili bir şekilde yakalar. Sınıflandırma, doğrusal modele kıyasla daha doğrudur. Karar bölgeleri daha pürüzsüzdür ve veri yapısına daha iyi uyar. Polinomal çekirdek SVM, veri kümesinin şeklini takip eden ve doğrusal olmayan bir karar sınırını öğrenerek doğrusal çekirdek SVM'den daha iyi performans gösterir (Tharwat, 2019).



Şekil 19. 3 Boyutlu Polinomal Çekirdek Fonksiyonu Dönüşümü

Şekil 19'da ise; X (X_1), Y (X_2) ve Z (X_3) eksenleri polinomal çekirdek tarafından uygulanan farklı özellik dönüşümlerini temsil etmektedir. Bu alanda iki sınıf daha ayrılabilirdir, yani basit bir doğrusal düzlem artık onları ayırabilir. SVM modeli artık bu dönüştürülmüş alanda doğrusal bir karar sınırı bulabilir. Polinomal çekirdek, verileri örtük olarak daha yüksek

boyutlu bir alana eşler. Bu alanda, doğrusal olmayan şekilde ayrılabilir olan veriler, artık doğrusal olarak ayrılabilir hale gelmiştir. SVM dönüşümü burada açıkça hesaplamaz; bu etkiyi elde etmek için yalnızca çekirdek işlevi kullanılır.

Gaussian radyal temelli çekirdek fonksiyonu (RBF[97]) ise, karmaşık karar sınırlarını modelleme yeteneği nedeniyle en yaygın kullanılan çekirdek fonksiyonlarından biridir (Chittora ve Mishra, 2012; Govidarajani, 2013; Ding vd., 2021). Formülü şu şekilde ifade edilmektedir:

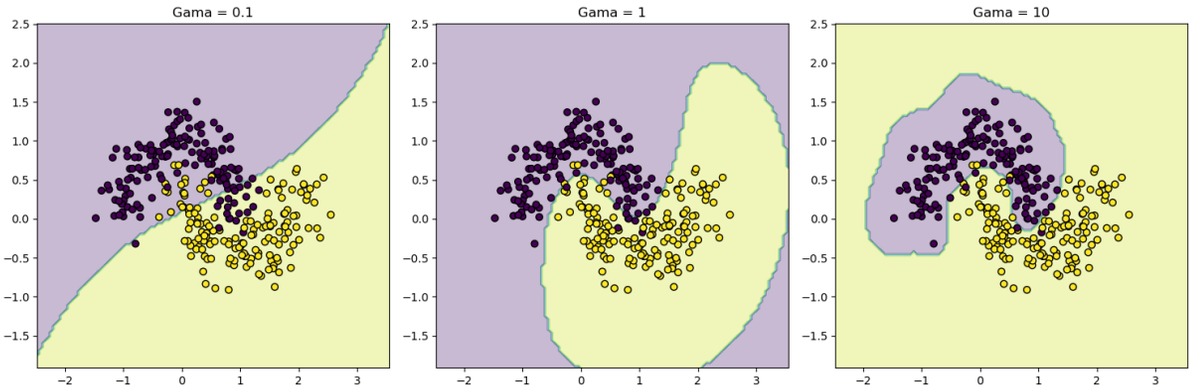
$$\mathrm{K}(x_i,x_j) = exp\,(-\frac{\|x_i - x_j\|^2}{2\sigma^2}) \tag{2.25}$$

Formülde $\mathrm{K}(x_i,x_j)$, iki veri noktası x_i ve x_j arasındaki çekirdek benzerliğini; $\|x_i - x_j\|^2$, iki nokta arasındaki karesel Öklid uzaklığını; σ (sigma), fonksiyonun yayılımını kontrol eden parametreyi; *exp* ise üstel fonksiyonu ifade etmektedir. Gamma parametresini (γ) kullanan bir diğer eşdeğer formülasyon ise şu şekilde ifade edilmektedir:

$$\mathrm{K}(x_i,x_j) = exp\,(-\gamma\| x_i - x_j\|^2) \tag{2.26}$$

Formülde $\gamma = \frac{1}{2\sigma^2}$ eşitliğine denktir. γ parametresi özellikle scikit-learn gibi kütüphanelerde yaygın olarak kullanılmaktadır. İki nokta birbirine yakınsa (yani $\| x_i - x_j\|^2$), çekirdek değeri de 1'e yakındır (yani yüksek benzerlik). İki nokta birbirinden çok uzaktaysa çekirdek değeri 0'a yaklaşır (yani düşük benzerlik). Sigma (σ) veya gamma (γ) parametresi tek bir eğitim örneğinin etkisini kontrol eder. Küçük σ ve büyük γ değerlerinde fonksiyon hızla azalır, yani yalnızca çok yakın noktalar benzer kabul edilir. Büyük σ ve küçük γ değerlerinde ise fonksiyon yavaşça azalır, yani uzak noktalar da benzer kabul edilir (Scholkopf vd., 1997). Gaussian RBF çekirdek fonksiyonu; girdileri sonsuz boyutlu bir uzaya örtük olarak eşleyerek doğrusal olmayan şekilde ayrılabilir verileri işlediği, çok yönlü ve akıcı olduğu ve polinom çekirdeklerinin aksine özellik mühendisliği gerektirmediği için sıklıkla kullanılmaktadır (Yu vd., 2011). Genellikle, optimum γ'yi bulabilmek için çapraz doğrulama[98] ile ızgara araması[99] gibi hiperparametre ayarlama teknikleri kullanılır.

[97] Radial Basis Function
[98] Cross validation
[99] Grid search



Şekil 20. Gamma Parametreli RBF Çekirdek Fonksiyonu

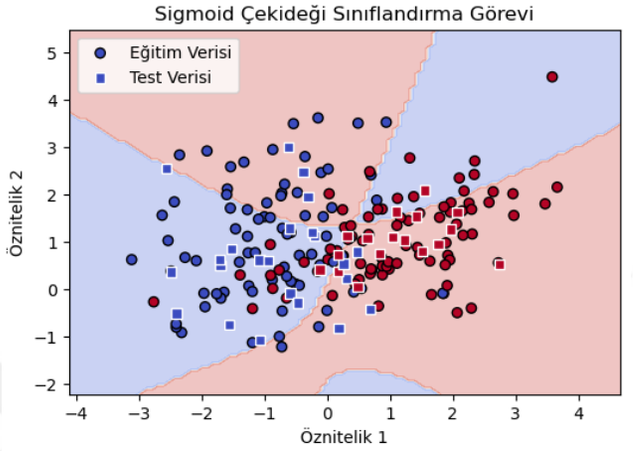
Şekil 20'de, gamma katsayısı 0.1 olan en soldaki grafikte karar sınırı düzgündür, dolaysıyla verilerin ince ayrıntıları pek yakalanamaz. Model, birbirinden uzaktaki noktaların yine de benzer olabileceğini varsaymaktadır. Sınıflandırıcı çok basittir ve bu da yüksek önyargıya (yani uyumsuzluğa) yol açar. Sınıra yakın bazı noktalar, yetersiz esneklik nedeniyle yanlış sınıflandırılmıştır. Gamma katsayısı 1 olan ortadaki grafikte ise karar sınırı daha esnektir ve verilerin şekline iyi uyum sağladığı görülmektedir. Bu değer, genellikle önyargı ve varyans arasında iyi bir denge sağlar. Model, aşırı karmaşıklıktan kaçınırken çoğu noktayı doğru şekilde sınıflandırır. En sağda bulunan grafikte ise gamma katsayısı 10'dur ve grafikte karar sınırı oldukça karmaşıktır, yani eğitim verilerine çok sıkı bir şekilde uymaktadır. Model, tekil veri noktalarına karşı çok hassastır ve bu da aşırı öğrenmeye yol açar. Karar bölgelerinde keskin kıvrımlar ve dönüşler vardır, bu da modelin genel kalıplar yerine gürültüyü yakaladığı anlamına gelir. Bu, görülmeyen verilerde zayıf genellemeye neden olur.

Sigmoid çekirdek fonksiyonu, SVM'de kullanılan doğrusal olmayan çekirdeklerden biridir ve yapay sinir ağlarındaki aktivasyon fonksiyonlarından esinlenilerek geliştirilmiştir (Nurwaha, 2020). Bu fonksiyon, giriş özelliklerini doğrusal olmayan bir dönüşüm yoluyla başka bir uzaya eşleyerek karmaşık karar sınırlarının oluşturulmasına olanak tanır. Sigmoid çekirdek şu şekilde tanımlanır:

$$\mathrm{K}(x_i,x_j) = \tanh(\alpha \times (x_i \times x_j) + \mathrm{c}) \qquad (2.27)$$

$\mathrm{K}(x_i,x_j)$, x_i ve x_j veri noktaları arasındaki benzerliği ölçen çekirdek fonksiyonu olmak üzere; $x_i \times x_j$, iki özellik vektörü arasındaki skaler çarpımını; α, ölçek parametresini, yani girdinin çıktıyı ne kadar etkileyeceğini kontrol eden parametreyi; c, önyargıyı, yani doğrusalsızlığı kontrol etmek için karar sınırını kaydıran parametreyi; tanh ise değerleri (-1,1) aralığına eşleyen hiperbolik tanjant fonksiyonunu ifade etmektedir.

Sigmoid çekirdek fonksiyonu, giriş verilerini yapay sinir ağlarında kullanılan aktivasyon fonksiyonlarına benzer şekilde doğrusal olmayan bir uzaya dönüştürerek, eğri karar sınırları oluşturulmasına olanak tanır. Yapısı itibarıyla, aktivasyon fonksiyonu olarak görev yaptığı iki katmanlı bir yapay sinir ağına benzer şekilde çalışır. Doğru seçilen hiperparametreler (α ve c) ile, sigmoid çekirdek fonksiyonu karmaşık veri yapıları üzerinde etkili bir şekilde sınıflandırma yapabilir (Du vd., 2024).



Şekil 21. Sigmoid Çekirdek Fonksiyonu Sınıflandırma Grafiği

Şekil 21, bir sigmoid çekirdek fonksiyonunun iki boyutlu bir vektörel uzayda iki veri sınıfının nasıl sınıflandırdığını göstermektedir. Arka plan renkleri model tarafından atanan karar bölgelerini temsil etmektedir. SVM, alanı her biri bir sınıfa karşılık gelen iki bölgeye ayırır. Kırmızı ve mavi renkli her bir daire eğitim setinden bir veri noktasını; kırmızı ve mavi kare şekiller ise test verisini göstermektedir. Test verileri, modeli eğitmek için kullanılmıştır. Kırmızı renk 1. sınıfı; mavi renk 2. sınıfı göstermektedir. SVM fonksiyonu, bu noktaları karar sınırını öğrenmek için kullanır.

SVM, regresyon görevlerinde doğrusal ile doğrusal olmayan fonksiyonlar bazında kullanılmaktadır. Doğrusal SVR[100], model karmaşıklığını en aza indirirken, epsilon (ϵ) adı verilen önceden tanımlanmış bir tolerans marjı içinde modele uymaya çalışır (Lee, Hsieh ve Huang, 2005).

$D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ veri seti olmak üzere; $x_i \in R^d$, giriş öznitelik vektörünü; $y_i \in R$ ise hedef çıktıyı ifade etmektedir. Amaç; hedef değerlere yaklaşan doğrusal bir fonksiyon olan $f(x)$'i bulmaktır:

$$f(x) = w^T x + b \tag{2.28}$$

[100] Support Vector Regressor

Formülde $w \in R^d$ ağırlık vektörünü ve $b \in R$ ise önyargıyı temsil etmektedir.

SVR, ϵ tolerans marjını kullanarak y'yi minimum hata ile tahmin eden bir $f(x)$ fonksiyonu bulmayı amaçlar. Kare hataların toplamını en aza indirmek yerine, SVR, w normunu en aza indirirken çoğu veri noktasının ϵ duyarsız bir marj içinde kalmasını sağlar, yani küçük hatalar göz ardı edilir. Optimizasyon problemi şu şekilde formüle edilmektedir:

$$min_{w,b,\xi,\xi^*} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{2.29}$$

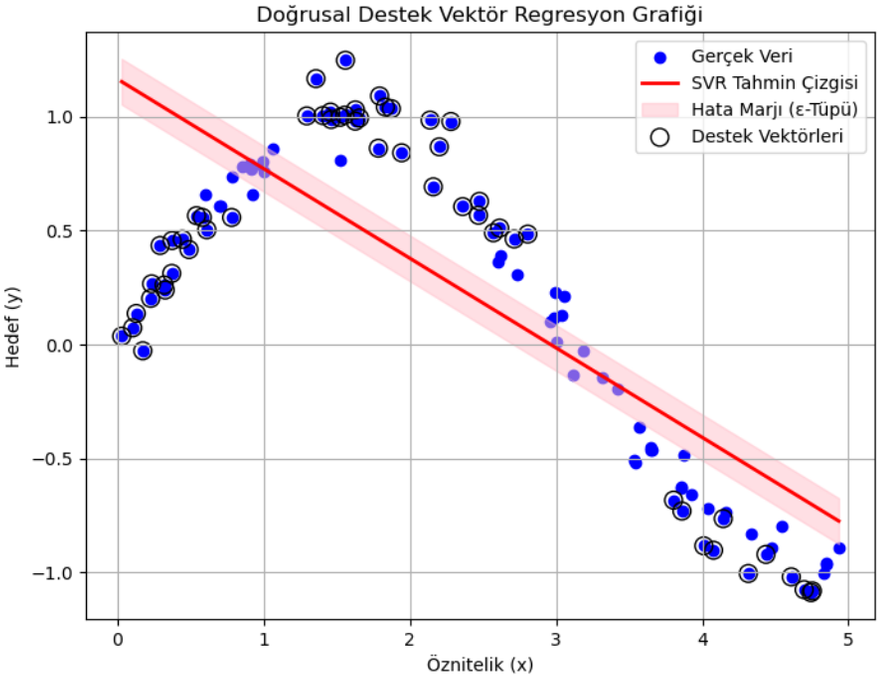
Formülde $\|w\|^2$, modelin aşırı karmaşık olmasını engelleyen düzenleme terimini (ağırlık değerlerini en aza indirerek ve aşırı öğrenmeyi önleyerek daha basit bir model sağlar); C, marj genişliği ile esneklik cezaları arasındaki dengeyi kontrol eden hiperparametreyi; ξ_i ve ξ_i^* ise ϵ marjından sapmalara izin veren esnek değişkenlerini ifade etmektedir.

Optimizasyondaki kısıtlamalar, ϵ marjındaki hataların göz ardı edilmesini, daha büyük sapmaların ise esnek değişkenler kullanılarak cezalandırılmasını zorunlu kılmaktadır:

$$y_i - w^T x_i - b \leq \epsilon + \xi_i$$
$$w^T x_i + b - y_i \leq \epsilon + \xi_i^*$$
$$\xi_i, \xi_i^* \geq 0, \quad \forall_i = 1, \ldots, n$$

Bu kısıtlamalar ile SVR modeli, aşırı öğrenmeyi engelleyici bir özellik kazanır. Bir veri noktası ϵ marjında bulunursa, yani tahmin edilen değerin ϵ değeri içindeyse, cezalandırılmaz. Bir veri noktası ϵ marjının dışında olursa, esnek değişkenler ξ_i ve ξ_i^* sapmayı hesaba katar ve bu sapma, amaç fonksiyonunda cezalandırılır. Başka bir deyişle ϵ marjı, regresyon fonksiyonunun etrafında hiçbir ceza uygulanmayan bir "alan" tanımlar (Wei ve He, 2023).

SVR, tahmin hatalarının belirli bir tolerans (ϵ) aralığında göz ardı edilmesine ve bu toleransı aşan durumların esnek değişkenlerle kontrol edilmesine olanak tanır. Bu yapı, doğrusal modelin sadeliğini korurken, gürültülü veriler ve yüksek boyutlu özellik uzaylarında esnek ve dayanıklı bir regresyon performansı sunar (Ho ve Lin, 2012).



Şekil 22. Doğrusal Destek Vektör Regresyon Grafiği

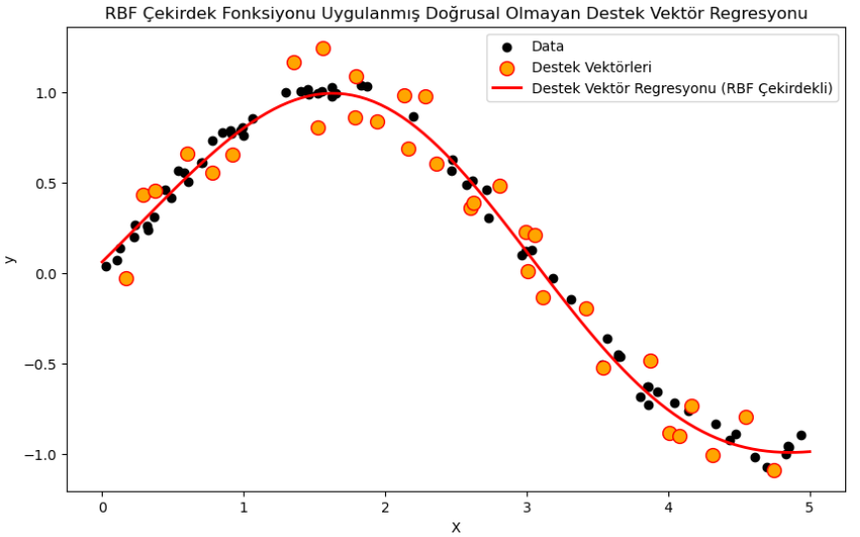
Şekil 22'deki örnek doğrusal SVR grafiği; tolerans marjının korunarak, modelin gürültüye karşı nasıl daha dayanıklı hale getirildiğini göstermektedir. Grafikte mavi noktalar, gürültülü gerçek veri noktalarını; kırmızı çizgi, doğrusal SVR modelinden tahmin edilen regresyon çizgisini; pembe alan, hataların göz ardı edildiği ϵ marjını (hata marjı); siyah daireler ise destek vektörlerini (yani marjı tanımlayan kritik veri noktaları) ifade etmektedir.

Doğrusal olmayan SVR ise girdi alanını doğrusal bir regresyonun gerçekleştirilebileceği daha yüksek boyutlu bir özellik alanına dönüştürür. Model, değişkenler arasındaki karmaşık ve doğrusal olmayan ilişkileri yakalamak için çeşitli çekirdek fonksiyonlarını (örneğin polinom veya gauss radyal baz fonksiyonu) kullanır. Doğrusal olmayan ilişkiler için SVR, giriş verisi x'i doğrusal regresyonun yapılabileceği daha yüksek boyutlu bir özellik uzayı ϕ(x)'e eşleyen çekirdek hilesi kullanılarak genişletir (Geng vd., 2020).

Doğrusal olmayan SVR formülü aşağıdaki eşitliği kullanır:

$$f(x) = w^T \phi(x) + b \qquad (2.30)$$

$\phi(x)$'i doğrudan hesaplamak, oldukça maliyetli olduğundan, skaler çarpımı $(\phi(x_i)^T \phi(x_j)$ dolaylı olarak hesaplayan çekirdek fonksiyonları $K(x_i,x_j)$ kullanılır (Smola ve Schölkopf, 2004).



Şekil 23. Doğrusal Olmayan Destek Vektör Regresyon Grafiği

Şekil 23'te, RBF çekirdek fonksiyonu uygulanmış bir doğrusal olmayan SVR grafiği gösterilmektedir. 100 rastgele veri noktası oluşturulan grafikte RBF çekirdeği kullanılmış ve düzenlileştirme değeri C=100 alınarak iyi bir uyum sağlanmaya çalışılmıştır. Gamma=0,5 alınarak da bir destek vektörünün etkisinin ne kadar uzağa ulaştığı kontrol edilmiştir. Grafikteki siyah noktalar gürültülü gerçek verileri; turuncu noktalar ise destek vektörlerini; kırmızı eğri ise SVR tarafından öğrenilen regresyon fonksiyonunu ifade etmektedir.

2.8.1.1.5. k-En Yakın Komşu[101]

KNN hem sınıflandırma hem de regresyon görevleri için kullanılabilen basit ve örnek tabanlı bir makine öğrenmesi algoritmasıdır. Bir veri noktasını, komşularının nasıl sınıflandırıldığına göre sınıflandırır (Larose, 2005). Etiketli noktalara sahip bir veri kümesi $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ verildiğinde; x_i bir öznitelik vektörü ve y_i sınıf etiketi olmak üzere, KNN algoritmasının hesaplama adımları aşağıdaki gibidir:

- ***k değerinin seçilmesi:*** Dikkate alınacak komşu sayısı (k), çoğunlukla küçük bir tam sayı (3, 5, 7 vb.) olarak seçilebilir.
- ***Mesafe ölçümü:*** Sorgu noktası x (yani, sınıflandırılmak istenen nokta) ile eğitim setindeki diğer tüm veri noktaları arasındaki mesafe hesaplanır. Kullanılan en yaygın mesafe ölçümü Öklid Mesafesi[102]'dir:

$$\mathrm{d}(x, x_i) = \sqrt{\sum_{j=1}^{m} (x_j - x_{ij})^2} \qquad (2.31)$$

[101] k-Nearest Neighbors - KNN
[102] L2 norm olarak da bilinir.

Formülde x, sorgu noktasını; x_i, i. eğitim noktasını; m, özniteliklerin (yani boyutların) sayısını; x_j ve x_{ij} sırasıyla sorgu noktasının j. ve i. eğitim noktasının öznitelik değerlerini göstermektedir.

- ***En yakın k komşunun bulunması:*** x ile veri kümesindeki tüm noktalar arasındaki mesafeler sıralanır ve x'e en yakın olan k eğitim noktası seçilir.
- ***Sınıf oylaması:*** En yakın k komşu belirlendikten sonra, sorgu noktası x için sınıf etiketi çoğunluk oyu ile belirlenir. k komşu arasında en çok görünen sınıf, x'e atanır. Örneğin; k=5 ve 3 komşu A sınıfına, 2'si B sınıfına ait ise, x sorgu noktası A sınıfı olarak sınıflandırılır. Eğer k komşu sayısı belirlendiyse sınıflandırma formülü şu şekildedir:

$$y(x) = çoğunluk\ oyu\ \{y_1, y_2, \dots, y_K\} \tag{2.32}$$

Formülde $y(x)$, x sorgu noktası için öngörülen sınıf etiketini; $y_1, y_2, \dots, y_K$ ise K en yakın komşunun sınıf etiketlerini ifade etmektedir.

- ***Son Sınıflandırma:*** Çoğunluk oyu alan sınıf, sorgu noktasına atanır (Hu vd., 2016; Raschka, 2020).

KNN, "boyutluluk laneti" nedeniyle mesafe değerlerinin daha az anlamlı hale geldiği yüksek boyutlu veriler için uygun değildir (Peng, Gui ve Wu, 2023).

Bir diğer mesafe ölçüm metriği ise Manhattan mesafesidir[103]. Taksi mesafesi olarak da adlandırılır ve her eksen boyunca noktalar arasındaki toplam mutlak fark ölçülür. Düz bir çizgi mesafesi yerine, yalnızca yatay ve dikey yönlerdeki hareketlerin toplamı hesaplanır (Dove vd., 2023). Formülü şu şekildedir:

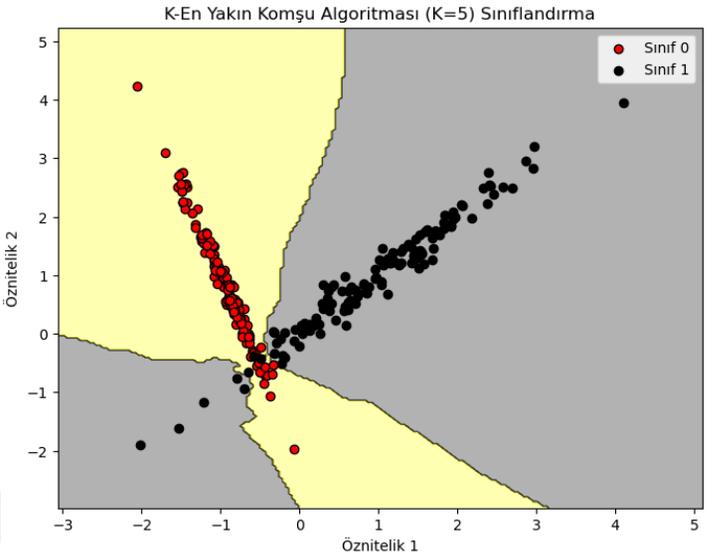
$$\mathrm{d}(x, x_i) = \sum_{j=1}^{m} |x_j - x_{ij}| \tag{2.33}$$

Tıpkı Öklid mesafesinde olduğu gibi her bir özellik üzerinde ayrı ayrı işlem yapılır; ancak burada farkların karesi yerine mutlak değeri alınır. Manhattan mesafesi, çoğu durumda yüksek boyutlu alanlar için Öklid mesafesinden daha iyi sonuç verir çünkü aykırı özelliklerin etkisini azaltır. Mesafeleri fazla tahmin ettiği için çapraz hareket önemli olduğunda kullanımı uygun değildir (Asgar, Hidayat ve Bejo, 2023).

Özellikler iyi ölçeklendirilmişse ve daha "doğal" bir mesafeye ihtiyaç varsa, mutlaka Öklid mesafesi seçilmelidir. Özellikler değişken ölçeklere sahipse veya yüksek boyutluysa,

[103] L1 norm olarak da bilinir.

Manhattan mesafesi genellikle daha iyi performans göstermektedir (François, Wertz ve Verleysen, 2011). Ayrıca, çapraz doğrulama ile de hangi mesafe katsayısının seçileceği belirlenebilir.



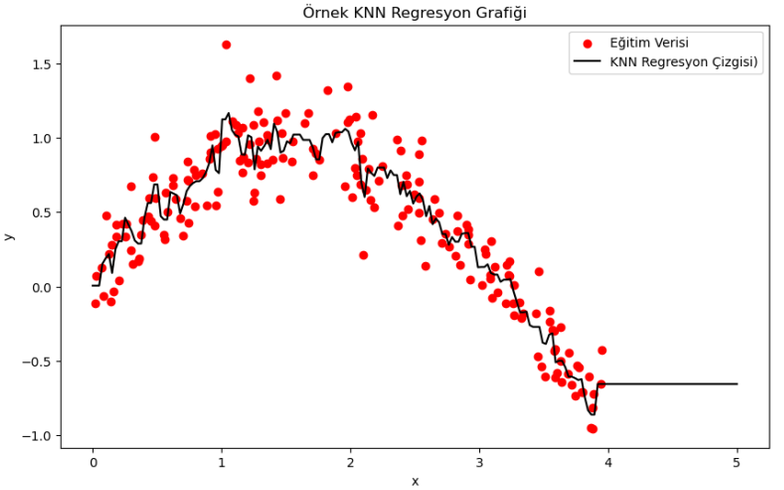
Şekil 24. k-En Yakın Komşu Algoritması Sınıflandırma Grafiği

Şekil 24'te k=5 seçilerek oluşturulan KNN sınıflandırmasına ait örnek bir grafik gösterilmektedir. Veri kümesi, sınıf 0 ve sınıf 1 olarak ayrılmıştır. Sınıflandırıcıda, yeni gözlemlerin sınıfını tahmin etmek için k=5 komşu değeri seçilmiştir. Bölgeler arasındaki sınırın doğrusal olmadığı görülmektedir. Kümelerin net bir şekilde ayrılması, algoritmanın iki sınıf arasında etkili bir şekilde ayrım yapabileceğini göstermektedir, ancak sınıf dağılımlarının örtüştüğü sınıra yakın yerlerde olası yanlış sınıflandırmalar da meydana gelebilir. Sınırın düzgünlüğü, k hiperparametresinin seçiminden etkilenir. Daha küçük k değerleri daha düzensiz sınırlara yol açabilirken, daha büyük değerler karar bölgelerini aşırı düzgünleştirebilir ve potansiyel olarak önyargıyı artırabilir (Levada, Nielsen ve Haddad, 2024).

Regresyon görevlerinde ise, x_0 sorgu noktası için tahmin edilen $\hat{y}_0$ değeri aşağıdaki formül ile ifade edilir:

$$\hat{y}_0 = \frac{1}{k}\sum_{i=1}^{k} y_i \qquad (2.34)$$

Formülde y_i, i. en yakın komşunun gerçek değerini; k ise komşu sayısını ifade etmektedir.



Şekil 25. k-En Yakın Komşu Algoritması Regresyon Grafiği

Şekil 25'te klasik bir KNN regresyon tahmin grafiği gösterilmektedir. Grafik, k=4 olarak belirlenen komşu sayısına göre tekil ağırlıklar atanarak çizilmiştir. Kırmızı noktalar; rastgele gürültü eklenmiş eğitim veri kümesini temsil etmektedir. Siyah regresyon eğrisi ise KNN regresyon modelinin tahminidir. Eğri, KNN regresyonunun çalışma şekli nedeniyle basamaklı görünmektedir. Her bir tahmin, en yakın 4 komşunun ortalamasına dayanmaktadır. Eğri, k değeri artırılırsa daha düzgün hale gelir ancak verilere uyum sağlamada yetersiz kalır. Değerin azaltılması durumunda ise, çizgi daha engebeli hale gelir ve gürültüleri de içine alacak şekilde aşırı öğrenmeye yol açar.

Klasik KNN regresyonunda, sorgu noktası için tahmin edilen değer, en yakın k komşunun hedef değerlerinin eşit ağırlıklı ortalaması olarak hesaplanır. Ancak bu yaklaşım, tüm komşuların eşit derecede etkili olduğunu varsaydığı için, uzak komşuların gereksiz yere etkili olmasına neden olabilir. Ağırlıklı KNN, bu sorunu çözmek amacıyla, her bir komşuya uzaklığa bağlı olarak ağırlık atar ve tahmini, ağırlıklı ortalama ile hesaplar (Wang, Qin ve Wang, 2021). Ağırlıklı KNN regresyonunun temel fikri, yeni bir girdi x için sürekli bir hedef $\hat{y}$'yi, her bir komşunun uzaklığına ters orantılı bir faktörle ağırlıklandırıldığı, k en yakın komşusunun hedef değerlerinin ortalaması alınarak tahmin etmektir (Mukid vd., 2018). Tahmin $\hat{y}$ şu şekilde hesaplanmaktadır:

$$\hat{y} = \frac{\sum_{i=1}^{k} w_i \times y_i}{\sum_{i=1}^{k} w_i} \tag{2.35}$$

Formülde k, dikkate alınan en yakın komşuların sayısını; w_i, i. komşuya atanan ağırlığı; y_i ise i. komşunun hedef değerini ifade etmektedir. Ağırlıklar genellikle mesafe tabanlı fonksiyonlar kullanılarak hesaplanmaktadır. Bu yöntemlerden biri olan ters mesafe ağırlıklandırması formülü şu şekilde ifade edilir:

$$w_i = \frac{1}{d(x, x_i) + \epsilon} \tag{2.36}$$

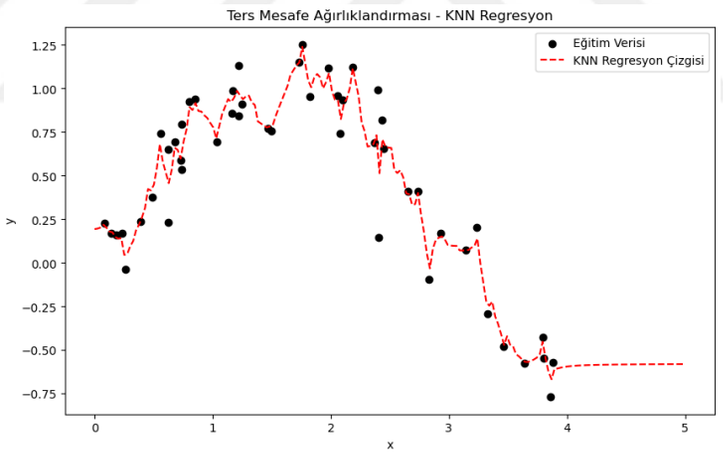
Formülde $d(x, x_i)$, sorgu noktası x ile komşusu x_i arasındaki mesafeyi; ϵ ise sıfıra bölünmeyi önlemek için pozitif sabit sayıyı ifade etmektedir.

Gauss ağırlıklandırma formülü ise şu şekilde ifade edilir:

$$w_i = exp(-\frac{d(x, x_i)^2}{2\sigma^2}) \tag{2.37}$$

Formülde $d(x, x_i)$, sorgu noktası x ile komşusu x_i arasındaki mesafeyi; σ ise ağırlığın mesafeye bağlı olarak azalma hızını kontrol eden bant genişliği (veya düzeltme) parametresini ifade etmektedir.

Ağırlık, mesafe arttıkça üstel olarak azalır. Daha yakın noktalar ağırlıklarını 1'e yakınlaştırırken, uzak noktalar ağırlıklarını 0'a yakınlaştırır. σ parametresi ağırlıkların ne kadar hızlı azaldığını belirler. Küçük σ değerinde ağırlıklar hızla düşerek çok yakın noktalara daha fazla önem verilir. Büyük σ değerinde ise ağırlıklar daha yavaş azalır ve daha uzaktaki noktalar önemli kabul edilir (Halder vd., 2024).



Şekil 26. k-En Yakın Komşu Algoritması Ters Mesafe Ağırlıklandırması Grafiği

Şekil 26'da ise, gürültülü bir veri kümesine uygulanan mesafe tabanlı ağırlıklandırma ile KNN regresyon modeli gösterilmektedir. Siyah noktalar, gürültü eklenerek oluşturulan eğitim verilerini temsil etmektedir. Veri noktalarının x ekseninde 0'dan 4'e kadar yayıldığı görülebilir. Kırmızı kesikli eğri, 4 komşulu KNN modeli tarafından yapılan tahminleri göstermektedir. Mesafe tabanlı ağırlıklandırma kullanıldığından daha yakın komşular, daha uzaktaki noktalara kıyasla tahminler üzerinde daha güçlü bir etkiye sahiptir (Karabulut, Arslan ve Ünver, 2019). Seyrek alanlarda, tahmini etkileyen daha az komşu nedeniyle model, eğriden daha fazla sapabilir. Komşular çok az ise aşırı öğrenme meydana gelebilir, ancak mesafe

tabanlı ağırlıkların kullanılması bunu hafifletmeye yardımcı olur. Eğitim aralığının ötesindeki tahminler (x=4'ten sonra) daha az güvenilir hale gelmektedir.

2.8.1.1.6. Naive Bayes

Naive Bayes, Bayes teoremine dayanan basit ama güçlü olasılıksal algoritmalar ailesidir (Sowjanya ve Veeralaksmi, 2024). "Naive[104]" olarak adlandırılırmasının nedeni; sınıf etiketi verildiğinde özelliklerin birbirinden bağımsız olduğunu varsaymasıdır. Naive Bayes'in özü, Bayes Teoremi'dir ve formülü şu şekilde ifade edilebilir:

$$P(C|X) = \frac{P(X|C) \times P(C)}{P(X)} \tag{2.38}$$

Formülde $P(C|X)$, X özellikleri verildiğinde C sınıfının olasılığını; $P(X|C)$, C sınıfı verildiğinde X özelliklerinin gözlemlenme olasılığını; $P(C)$, sınıf C'nin meydana gelme olasılığını ve $P(X)$, X özelliklerinin ortaya çıkma olasılığının toplam değerini ifade etmektedir. Bir özellik vektörü X=(x_1,x_2,...,x_n) için saf varsayım (özellik bağımsızlığı) şu formül ile gösterilir:

$$P(X|C) = \prod_{i=1}^{n} P(x_i|C) \tag{2.39}$$

Böylece Bayes teoremi aşağıdaki formülle gösterilir hale gelir:

$$P(C|X) = \frac{P(C) \times \prod_{i=1}^{n} P(x_i|C)}{P(X)} \tag{2.40}$$

$P(X)$ tüm sınıflar için sabit olduğundan, sınıflandırma için yalnızca paya ihtiyaç duyulmaktadır:

$$P(C|X) \propto P(C) \times \prod_{i=1}^{n} P(x_i|C) \tag{2.41}$$

Bu formül Naive Bayes sınıflandırıcısının çekirdeğidir ve gerçek olasılıkları hesaplamak yerine, her sınıf için karşılaştırılabilir bir skor üretmeyi amaçlar. Formülde $P(x_i|C)$, sınıfın C olduğu varsayıldığında x_i özelliğinin ne kadar olası olduğunu ifade etmektedir. Verilen bir girdi X'in (x_1,x_2,...,x_n) bir C sınıfına ait olma olasılığını hesaplar (Ismail, Hassan ve Bafjaish, 2020).

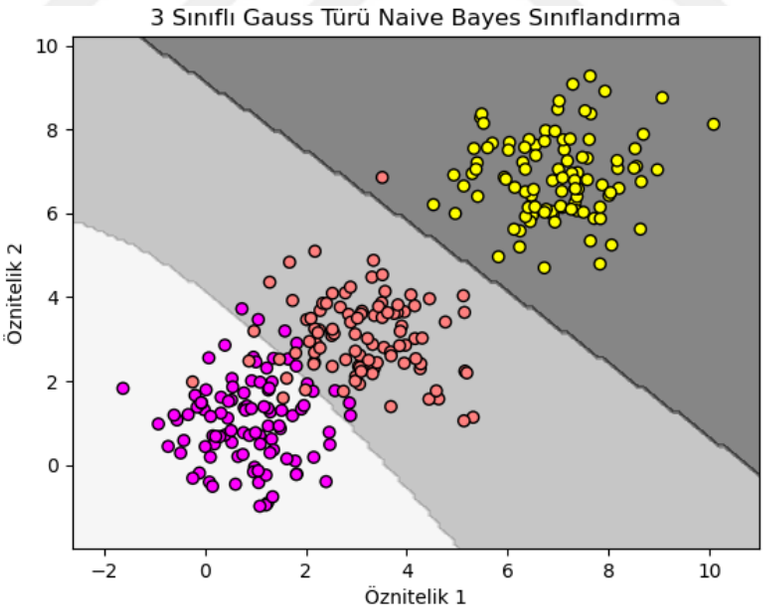
Sürekli veriler için de Gauss Naive Bayes kullanılır ve özelliklerin normal dağılıma uyduğu varsayılır. Olasılık formülü şu şekilde ifade edilmektedir:

[104] Saf

$$P(x_i|C) = \frac{1}{\sqrt{2\pi\sigma_C^2}}\, exp(-\frac{(x_i - \mu_C)^2}{2\sigma_C^2}) \qquad (2.42)$$

Bu formül, özelliğin normal (Gauss) dağılımını izlediğini varsayarak, C sınıfına ait bir özellik değeri x_i'nin gözlemlenme olasılığını hesaplamaktadır. Formülde x_i, i. özelliğin gözlemlenen değerini; C, göz önüne alınan sınıfı; μ_C, C sınıfındaki tüm örnekler için x_i özelliğinin ortalama değerini ($\mu_C = \frac{1}{N}\sum_{k=1}^{N} x_k$ şeklinde hesaplanır; formülde N, C sınıfından gelen örnek sayısını ifade etmektedir); σ_C^2, C sınıfındaki özellik değerlerinin ortalamanın etrafında ne kadar yayıldığını ölçen varyansı; $\sqrt{2\pi\sigma_C^2}$, olasılık dağılımını toplam olasılığın 1'e eşit olması için ölçeklendiren değerini; $exp(-\frac{(x_i-\mu_C)^2}{2\sigma_C^2})$, gözlemlenen değer x_i'nin ortalamadan ne kadar uzakta olduğuna bağlı olarak ne kadar olası olduğunu belirleyen formülü ifade etmektedir (Bouckaert, 2004). Formül çan şeklindeki bir eğriyi tanımlar. Eğer x_i ortalama μ_C'ye yakınsa, $P(x_i|C)$ yüksektir. x_i uzaklaştıkça da olasılık azalır.

Naive Bayes sınıflandırıcısı, hesaplama açısından verimli olması ve yüksek boyutlu veri kümelerinde dahi etkili performans sergilemesi nedeniyle metin madenciliği, e-posta filtreleme, tıbbi teşhis gibi pek çok alanda sıklıkla kullanılmaktadır.



Şekil 27. Gauss Naive Bayes Sınıflandırma Grafiği

Şekil 27'de gösterilen grafik, Gauss Naive Bayes sınıflandırıcısının karar sınırlarını, üç sınıf için veri noktalarıyla birlikte vermektedir. Sarı noktaların oldukça iyi bir şekilde ayrıldığı görülmektedir. Diğer iki sınıfın birbirine daha yakın ve kısmen örtüşmekte olduğu da görülebilir. Başka bir deyişle karar bölgeleri arasında daha yumuşak bir geçiş vardır.

Süreksiz verilerde ise (örneğin, metin verileri) çok terimli Naive Bayes kullanılır. Burada olasılık, kelime sıklıklarından hesaplanır. Çok terimli Naive Bayes[105], özellikle spam filtreleme ve duygu analizi gibi metin sınıflandırma problemlerinde sınıflandırma görevleri için kullanılan olasılıksal bir makine öğrenme algoritmasıdır (Hayatin vd., 2022).

X, sözcük sayılarından oluşan bir belge ise, olasılık $P(X|C_k)$ çok terimli bir dağılım kullanılarak hesaplanabilir:

$$P(X|C_k) = \prod_{j=1}^{n} P(w_j|C_k)^{x_j} \tag{2.43}$$

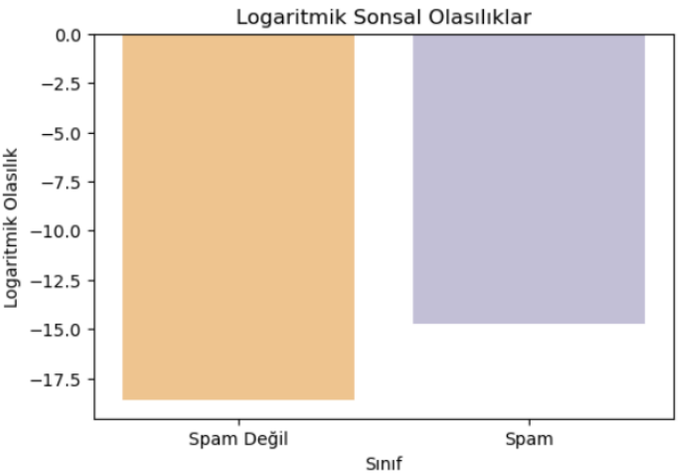
Formülde n, sözlükteki tekil kelime sayısını; w_j, sözlükteki j. kelimeyi; x_j, X belgesindeki w_j kelimesinin sıklığını ve $P(w_j|C_k)$, w_j kelimesinin C_k sınıfında bulunma olasılığını ifade etmektedir.

```
Sözlük: ['akademik' 'araba' 'boş' 'ders' 'edilmiştir' 'eğitimleri' 'fırsat' 'ilan'
 'iphone' 'için' 'jackpot' 'kazandınız' 'kazanma' 'kitap' 'okurum'
 'saatleri' 'sitesinde' 'son' 'web' 'yazım' 'zamanlarda' 'çekilişi'
 'şansı']
Özellik Matrisi:
 [[0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0]
 [0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0]
 [0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1]
 [1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0]
 [0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0]
 [0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0]]
Cümle: Araba çekilişi için son fırsat
Tahmin Edilen Sınıf (1=Spam, 0=Spam Değil): 1
Logaritmik Sonsal Olasılık (Spam): -14.70994908509262
Logaritmik Sonsal Olasılık (Spam Değil): -18.610741872840492
Tahmin: Spam
```

Şekil 28. Çok Terimli Naive Bayes 1

Şekil 28'de, örnek çok terimli Naive Bayes sınıflandırıcısı için oluşturulan sözlük ve kelimeler, kelimelerin matris gösterimi ve seçilen bir cümle içerisinde bulunan kelimelerin analizi yoluyla cümlenin spam olup olmadığı gösterilmektedir. Burada yapılan hesaplamalar sonucunda "araba çekilişi için son fırsat" cümlesinin "spam" olarak tahmin edildiği görülmektedir. Yapılan hesaplamadaki skorlar, logaritmik sonsal olasılık değeri spam ve spam değil sınıfı için hangisinde daha yüksek olursa, o sınıfa atanmaktadır. Örnekte -14.70'lik "spam" skoru, -18,61'lik "spam değil" skorundan daha yüksek olduğu için cümle spam olarak sınıflandırılmıştır.

[105] Multinomial Naive Bayes - MNB



Şekil 29. Çok Terimli Naive Bayes Grafiği 2

Şekil 29'da ise ilgili cümlenin spam sınıfına dahil olduğunu gösteren grafik bulunmaktadır.

2.8.1.1.7. Rastgele Orman

Rastgele orman algoritması, hem sınıflandırma hem de regresyon görevleri için kullanılan güçlü bir topluluk öğrenme yöntemidir. Eğitim sırasında çok sayıda karar ağacı oluşturarak ve tekil ağaçların modunu (sınıflandırma) veya ortalama tahminini (regresyon) vererek çalışır. Algoritmanın temeli; rastgele alt düzlemler için birden çok ağaç yapısı oluşturmaktır (Cutler, Cutler ve Stevens, 2012). Karar ağaçlarından farkı; bölmelerin rastgeleliğidir.

Rastgele ormandaki her ağaçta, her bir bölünme için rastgele bir özellik alt kümesi seçilir ve sınıflandırma görevlerinde Gini safsızlık katsayısı kullanılır (Strobl, Boulesteix ve Augustin, 2007). Formülü şu şekilde ifade edilir:

$$\text{Gini} = 1 - \sum_{k=1}^{K} p_k^2 \tag{2.44}$$

Formülde p_k, k sınıfına ait örneklerin oranını; K ise sınıf sayısını vermektedir. Gini safsızlık katsayısı, rastgele seçilen bir küme elemanının ne sıklıkla yanlış sınıflandırıldığını ölçmektedir. Daha düşük Gini değeri, daha saf bölünmeleri gösterir.

Seçilen bölme, alt düğümlerdeki Gini safsızlık değerlerinin ağırlıklı toplamını en aza indirir:

$$Gini_{split} = \frac{N_L}{N} Gini_L + \frac{N_R}{N} Gini_R \tag{2.45}$$

Formülde N_L, N_R, sol ve sağ alt düğümlerdeki örnek sayısını; $Gini_L$, $Gini_R$, sağ ve sol çocuk düğümlerinin gini safsızlık katsayısını; N ise düğümdeki toplam örnek sayısını ifade etmektedir ve bölünmenin kalitesi değerlendirilirken kullanılır.

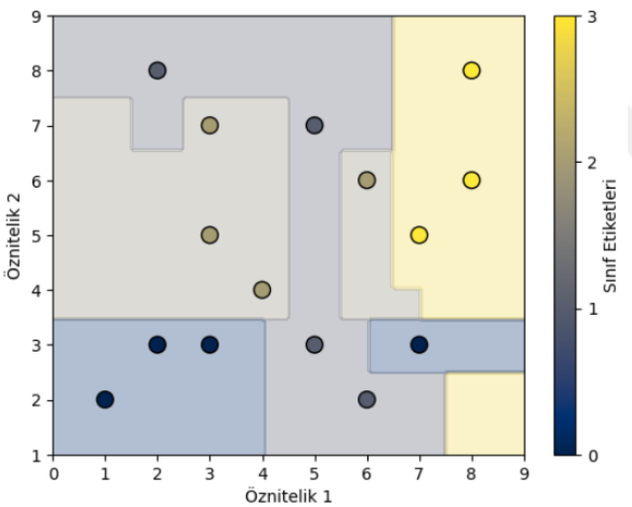
Entropi ve entropiye bağlı bilgi kazanımı aşağıdaki formüllerle ifade edilmektedir:

$$\text{Entropi} = -\sum_{k=1}^{K} p_k log_2\,(p_k) \tag{2.46}$$

Formülde K, sınıf sayısını; p_k, k sınıfının olasılığını; $\log_2$, 2 tabanında logaritmayı ifade etmektedir.

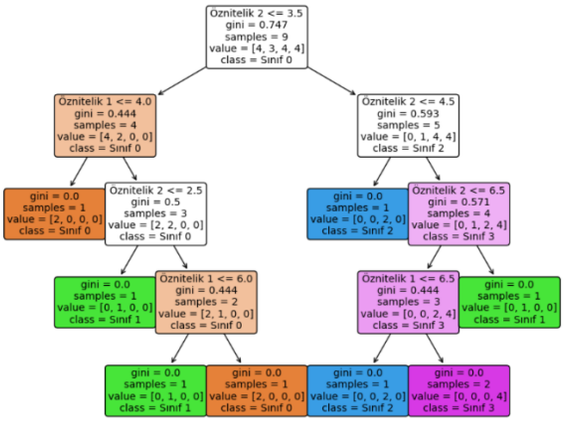
$$IG = \text{Entropi} - \sum_{j=1}^{m} \frac{|S_j|}{|S|}\, Entropi(S_j) \tag{2.47}$$

Formülde S üst düğümü; S_j, alt düğümü; m ise alt düğüm sayısını ifade etmektedir ve bir özelliğin karar ağacında ne kadar bilgi sağladığını ölçer. Entropi ne kadar azalırsa, özellik o kadar iyi bir ayrım sağlar ve en yüksek bilgi kazancına sahip özellik, ağacı bölmek için seçilir. Başka bir deyişle, ilk entropi genel sistem belirsizliği, alt düğümlerdeki ağırlıklı entropi ise bölünmeden kalan belirsizliktir ve aralarındaki fark, ne kadar bilginin kazanıldığını verir (Largeron, Moulin ve Géry, 2011).



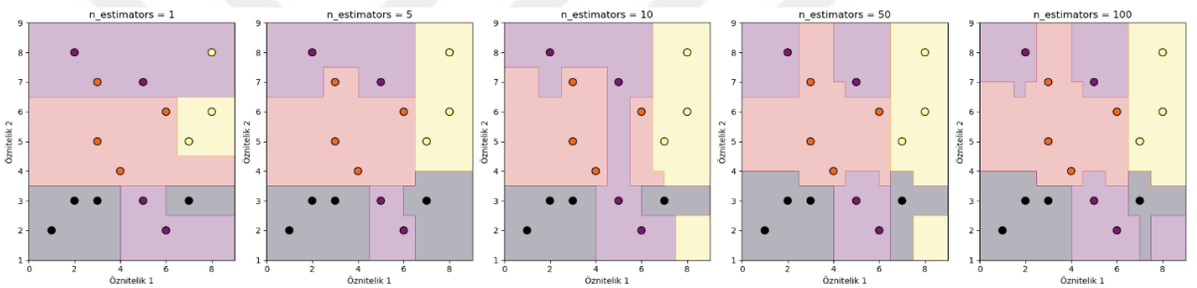
Şekil 30. Rastgele Orman Karar Sınırları Grafiği

Şekil 30'da rastgele orman algoritmasına ait karar sınırlarının örnek gösterimi bulunmaktadır. Grafikteki veri noktaları, dört farklı sınıfa (0, 1, 2, 3) aittir. Sınıflandırıcının amacı, verilerden örüntüler öğrenmek ve iki boyutlu uzaydaki herhangi bir yeni nokta için doğru sınıfı tahmin etmektir. Sınıflandırıcı; doğrusal olmayan, parça parça bir karar sınırı oluşturmak için ağaç topluluğundan öğrenilen karar kurallarını kullanır. Karar sınırının dalgalı yapısı, tahminler üzerinde birden fazla karar ağacının yapısından kaynaklanır ve bu da daha esnek ve genelleştirilebilir bir sınıra yol açar. Daha büyük bölgelerin içinde farklı renklerde görülen bazı küçük cepler, tekil ağaçların etkisinden kaynaklanmaktadır.



Şekil 31. Rastgele Orman Ağaç Yapısı Grafiği

Şekil 31'de ise, bir önceki şekilde bulunan karar sınırlarının ağaç yapısı gösterimi bulunmaktadır. Görüldüğü üzere 4 sınıflı ve 10 ağaçlı yapı, sınıflandırmayı başarılı bir şekilde gerçekleştirmiştir.



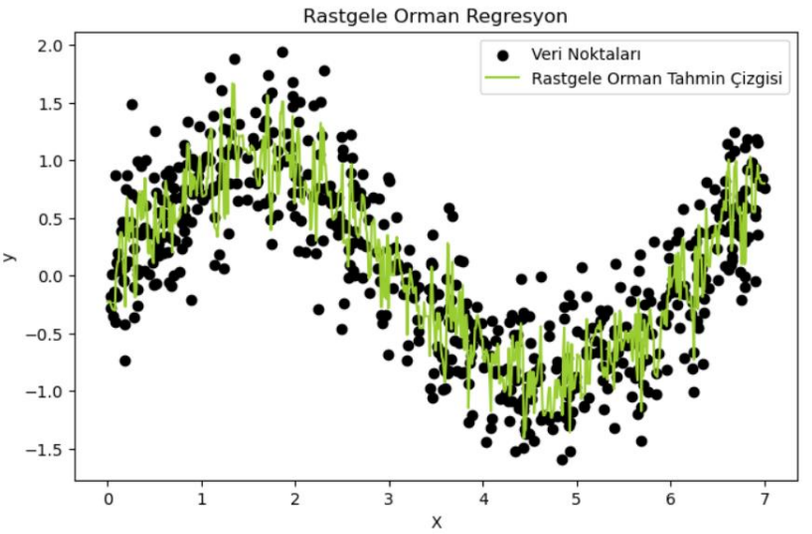
Şekil 32. Ağaç Yapısının Değiştiği Sınıflandırma Grafiği

Şekil 32'de ise, 1, 5, 10, 50 ve 100 ağaç için karar sınırlarının nasıl değiştiğine dair bir gösterim bulunmaktadır. Sınıflandırıcıdaki karar sınırının ağaç sayısı belirli bir noktanın ötesine geçtikçe daha kararlı hale gelmektedir.

Rastgele ormanın nihai tahmini ise, tüm B karar ağaçlarının çıktılarının ortalaması alınarak elde edilir:

$$\hat{y}_{RF}(X) = \frac{1}{B}\sum_{b=1}^{B} f_{(b)}X \qquad (2.48)$$

Formülde $\hat{y}_{RF}(X)$, X girdisi için son tahmin edilen değeri; B, toplam ağaç sayısını ve $f_{(b)}X$, b. ağaçtan X girdisi için gelen tahmindir (Wang, 2019).



Şekil 33. Rastgele Orman Regresyon Grafiği

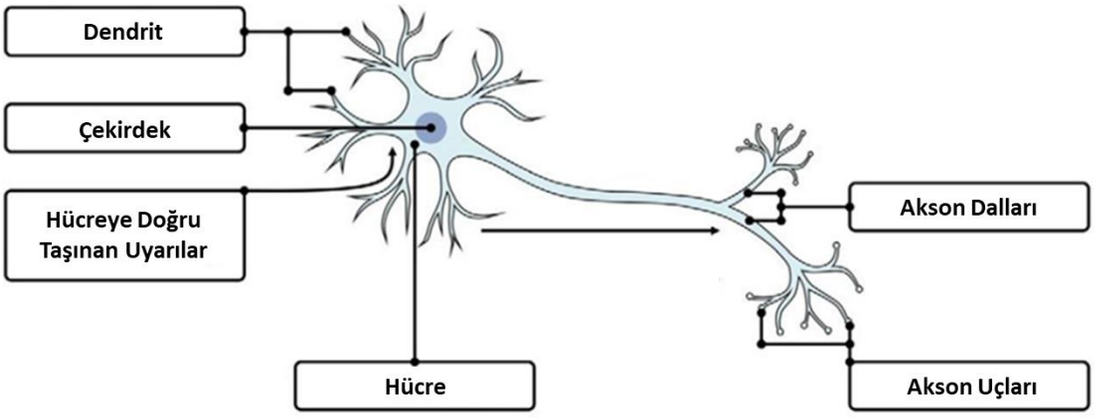
Şekil 33'te ise gürültü eklenmiş bir rastgele orman regresyon fonksiyonu kullanılarak 600 veri noktası üretildikten sonra veriler eğitim (%80) ve test (%20) olarak ayırılmıştır. 50 ağaçlı bir rastgele orman regresyon modeli eğitilmiş ve test kümesinde tahminlerde bulunulmuştur. Ortalama karesel hata (MSE) (0,22) ve R^2 skoru (0,79) kullanılarak model performansı değerlendirilmiş ve gerçek (siyah veri noktaları) ve tahmin edilen değerler (yeşil) çizdirilmiştir. Yeşil tahmin eğrisi, verilerin doğrusal olmayan eğilimini yakalamıştır.

2.8.1.1.8. Yapay Sinir Ağları

Yapay zekâ bölümünde de belirtildiği üzere, insan beyninin düşünce sürecini taklit eden bir sinir ağının ilk matematiksel modeli 1943'te Walter Pitts ve Warren McCulloch tarafından geliştirilmiştir. 1957'de Rosenblatt (1957), "The Perceptron and a Perceiving and Recognizing Automation" başlıklı teknik raporunda ilk defa algılayıcı sinir hücrelerini taklit eden ilk öğrenme modeli olan algılayıcıyı[106] tanıtmış; ardından 1958'de yayımlanan "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain" başlıklı makalesiyle de bu modeli geniş kitlelere duyurarak yapay sinir ağları alanında önemli bir dönüm noktasını oluşturmuştur (Rosenblatt, 1958). Ivakhnenko ve Lapa (1965) ise çok değişkenli veri setlerini kullanarak bilgisayar tabanlı matematiksel bir model oluşturmuş ve bu modelini sinir ağlarına uyarlayarak çalışan ilk derin öğrenme ağlarını geliştirmiştir. Ivakhnenko (1971) ayrıca, sekiz katmanlı bir derin öğrenme ağını da tanıtmıştır. Bu çalışmalarla hedeflenen ise; eğitilebilir, adaptif ve kendi kendine organize olup öğrenebilen,

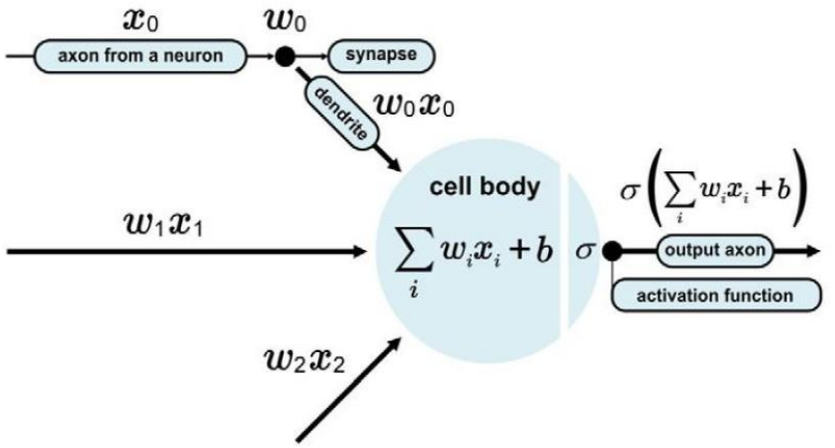
[106] Perceptron

değerlendirme yapabilen sinir ağları ile insan beyninin öğrenme ve uygulama yapısını modellemektir.



Şekil 34. Canlılardaki Sinir Hücrelerinin Biyolojik Görünümü
Kaynak: (Zhou vd., 2022, s. 20)

Şekilde 34'te gösterildiği üzere; her nöron, dendritlerinden giriş sinyalleri alır ve ardından akson boyunca çıkış sinyalleri üretir ve akson sonunda kademeli olarak dallanır ve sinapslar aracılığıyla diğer nöronların dendritlerine bağlanır.



Şekil 35. Sinir Hücrelerinin Matematiksel Modellenmesi
Kaynak: (Zhou vd., 2022, s. 20)

Şekil 35'te ise yapay bir nöronun matematiksel hesaplama modeli gösterilmektedir. Burada, akson boyunca yayılan sinyaller (x), sinapsların[107] sinaptik gücüne (w) bağlı olarak diğer nöronların dendritleriyle (matris işlemi wx) etkileşime girer. Sinaptik güç; bir nöronun diğer nöron üzerindeki etki gücünü ve yönünü; o nöronu uyararak (pozitif ağırlık) veya bastırarak (negatif ağırlık) kontrol edebilir. Çıkış nöronuna ise bir aktivasyon fonksiyonu eklenir. Bu şekilde, sinirler arası iletişim sağlanmaktadır. Sinir ağları, basit olarak; giriş katmanı, gizli

[107] İki sinir hücresi (nöron) veya bir nöron ile bir kas hücresi arasında elektriksel sinir uyarılarının iletildiği yer.

katman ve çıkış katmanından oluşur. Birçok nörondan ve gizli katmandan oluşan sinir ağına ise çok katmanlı sinir ağı denir. Eğer bir sinir ağı, tek bir katmandan oluşuyorsa tek katmanlı sinir ağı olarak tanımlanır (López, López ve Crossa, 2022). Her iki tür sinir ağının da yetenekleri farklılık göstermektedir.

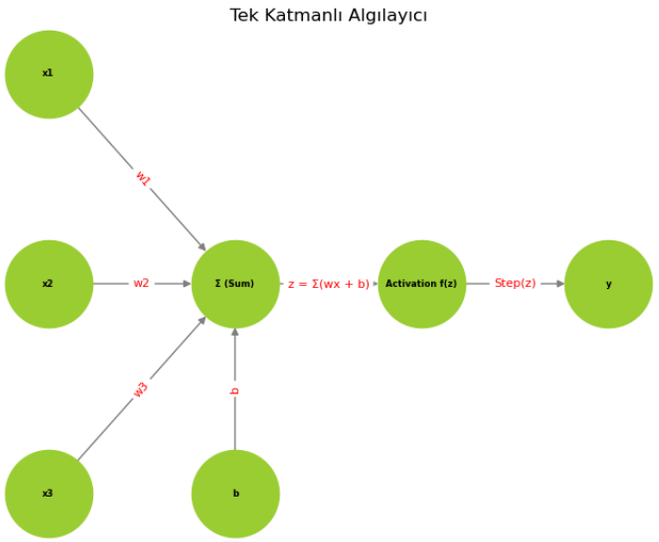
2.8.1.1.8.1. Tek Katmanlı Algılayıcı

Tek katmanlı algılayıcı[108], yapay sinir ağının en basit türüdür ve çoğunlukla ikili sınıflandırma için kullanılır ve mantık devrelerinde kullanılan kapılar gibi düşünülebilir (Wu, 2014). Girdisi ya doğrudan çıkışa ya da bir sonraki katmana eklenir. Genellikle iki girdiye ve bir çıktıya sahiptir. Girişler, sinapslar boyunca uzanır ve ağırlıkları ile çarpılarak bir eşik değerinden geçirilir. Bir girdinin, girdilerin ağırlıklı toplamı ve aktivasyon fonksiyonu uygulanmış çıkışına göre bir sınıfa mı yoksa diğer sınıfa mı ait olduğuna karar veren bir tür doğrusal sınıflandırıcı olarak da değerlendirilebilir (Du vd., 2022). Bir sinir hücresinin çıktı formülü şu şekilde ifade edilir:

$$z = \sum_{i=1}^{n} w_i x_i + b\,, \qquad a = f(z) \tag{2.49}$$

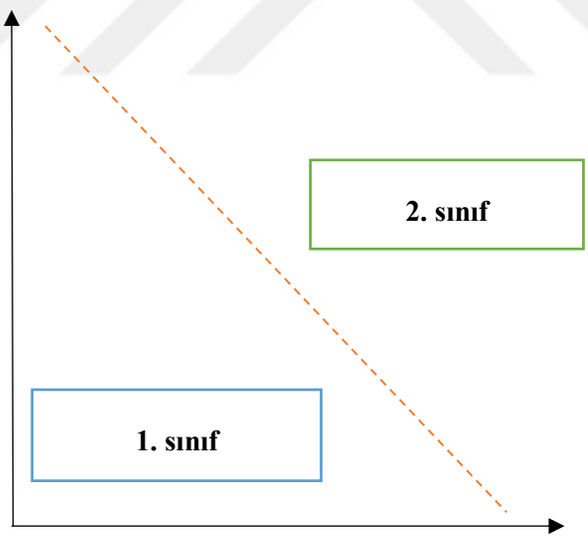
Formülde x_i, i. girişin özniteliğini; w_i, her bir girişle ilişkili ağırlıkları; b, önyargıyı[109] (bias, aktivasyon fonksiyonunu kaydırmaya yardımcı olur); z, ağırlıkların ve girdilerin doğrusal kombinasyonunu; $f(z)$, z'ye uygulanan aktivasyon fonksiyonunu ve a ise nöronun çıktısını (aktivasyon) ifade etmektedir. Giriş değeri x_i, dendritteki ağırlık w_i ile çarpıldıktan sonra sinir hücresine iletilir. Sinir hücresinde de tüm dendritlerden gelen ağırlık ve girdi değerlerinin çarpımları ($w_1x_1, w_2x_2, \dots, w_ix_i$) toplanır ve ardından bir b önyargı değeri eklenir. Bu adımdan sonra da bir aktivasyon fonksiyonu aracılığı ile çıkışa aktarılır. Çıkış; ya nihai çıkış noktası ya da başka bir sinir hücresinin girdisi olabilir. Her bir sinir hücresi, formül (2.49) doğrultusunda giriş değerleri ve ağırlıkların doğrusal kombinasyonu ile hesaplanır. Yapay sinir ağlarında nöronlar, katmanlar hâlinde yapılandırılır; katmanlar arasında bilgi akışı seri biçimde ilerlerken, aynı katmandaki nöronlar paralel olarak çalışır.

[108] Perceptron
[109] Bias



Şekil 36. Tek Katmanlı Algılayıcı Modeli

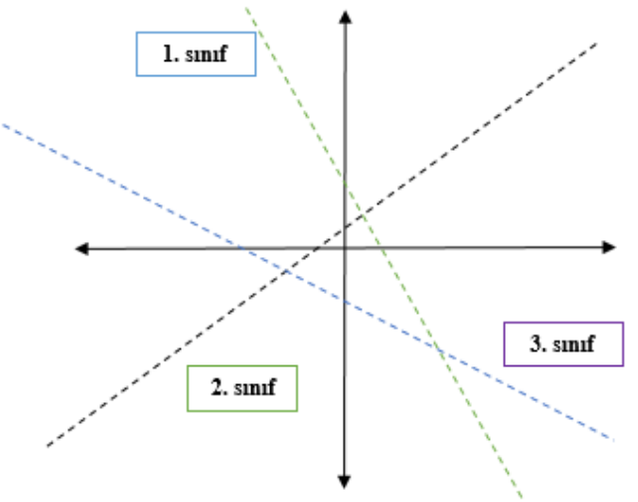
Şekil 36'da formül (2.49)'da gösterildiği üzere tek katmanlı algılayıcı modeli çizilmiştir. SVM algoritmasını geliştiren Cortes ve Vapnik (1995); sınıf dağılımlarını tahmin etmenin, sınırları tahmin etmekten çok daha zor bir iş olduğunu ve kolay bir problemi çözmek için daha zor bir problemi çözmenin gereksiz olduğunu belirtmiştir. Başka bir deyişle; sınırları tahmin etmek, sınıf dağılımlarını tahmin etmekten daha kolaydır.



Şekil 37. Düzlemde Sınıf Ayrımı

Şekil 37'de gösterildiği üzere, düzlemde 1. ve 2. sınıfın dış çevresini tahmin etmeye çalışmak yerine, 1. ve 2. sınıfı birbirinden ayıran sınırı tahmin etmeye çalışmak, daha kolay ve doğrusal bir problem halinde tanımlanabilir. Bu durumun geçerli olabilmesi için de problemin sınırlarının kolayca belirlenebiliyor olması gerekmektedir.

İki sınıfı ayıran bir doğru, tek katmanlı sinir ağı (perceptron) ile çözülebilirken, ikiden fazla sınıf varsa, birden çok doğru; ayrı ayrı birer katmanlı sinir ağı birleştirerek elde edilebilir.



Şekil 38. İkiden Fazla Sınıf Olduğunda Sınırların Belirlenmesi

Şekil 38'deki gibi, eğer birden çok sınıf var ise, tek katmanlı birkaç sinir ağı ile gerçekleştirilen ayırıcı çizgiler (yani, diskriminantlar[110]) ile bu sınırlar ayrılabilir. Ayırıcı çizgiler, karar doğrusu olarak da adlandırılır. Diskriminantların x girdisine bağlı ve doğrusal olduğu varsayılırsa, hesaplanması gereken diskriminant, ağırlıklara ve giriş değerine bağlı bir şekilde, d kadar toplam yapılarak aşağıdaki formül ile gösterilebilir:

$$g(x|w_i, w_{i0}) = w_i^T x + w_{i0} = \sum_{j=1}^{d} w_{ij} x_j + w_{i0} \qquad (2.50)$$

Formülde x, girdi; w, ağırlıklar; w_{i0}, i sınıfına ait bias; $\sum_{j=1}^{d} w_{ij} x_j$, girdilerle ağırlıkların doğrusal kombinasyonunu ve d ise d boyutlu öznitelik vektörü üzerindeki toplamları ifade etmektedir. Bu diskriminant denkleminin geçerli olabilmesi için, problemin sınırlarının kolayca belirlenebilmesi gerekmektedir. Doğrusal diskriminant, daha çok basitliği nedeniyle tercih edilen bir yöntemdir. Eğer; x_j girdisinin ağırlığı büyükse, ilgili j değeri, daha önemli anlamına gelir. Çıktı, farklı özniteliklerin toplamı olarak değerlendirilebilir. Ağırlığı büyük olan giriş, kendisinde varolan önemli özellikleri taşır (Li, Doroslovački ve Loew, 2019). Bu model genelleştirilecek olursa; x^T'nin W ile çarpılmış hali denilebilir:

$$g(x) = x^T W x + w^T x + w_0 \qquad (2.51)$$

[110] Gerçel katsayılı 2. dereceden polinom denklemlerin çözümünde kullanılmaktadır.

Formülde x^T, x giriş vektörünün transpozunu[111] ifade etmektedir. Transpoz işlemi, matris çarpımının kurallarına uygun olacak şekilde boyutları eşleştirmek içindir. Aksi halde çarpma işlemi yapılamaz.

Birinci dereceden diskriminant, bazı problemlerde yeterince esnek olmayabilir. Bunun için, ikiden fazla dereceli diskriminantlar kullanılabilir. Fakat derece arttıkça daha büyük öğrenme kümesine ihtiyaç duyulmaktadır. Eğer küçük öğrenme verisi ve yüksek dereceli diskriminantlar ile bir model kurulursa, aşırı öğrenme[112] riski doğabilir (Bing ve Wegkamp, 2023).

Diskriminant fonksiyonu g ile tanımlanacak olursa:

$$g_i(x) = \sum_{j=1}^{k} w_j \, \theta_{ij}(x) \tag{2.52}$$

Formülde $g_i(x)$, x girdisi için sınıf i'ye ait diskriminant fonksiyonunu; w_j, öğrenilen ağırlıkları (yani parametreler); $\theta_{ij}(x)$, x girdisi üzerine uygulanan temel fonksiyonlar veya özellik dönüşümlerini ifade etmektedir. Formül ayrıca, ağırlıkların x'e bağlı diskriminantlarının ağırlıklı toplamı şeklinde ifade de edilebilir. Girişe, çarpım terimleri de eklenip doğrusal modellerin doğrusal olmayan ilişkileri yakalayabilmesi sağlanabilir. Örneğin; x_1 ve x_2 giriş değerleri ise:

$$\begin{gathered} z_1 = x_1, z_2 = x_2, z_3 = x_1^2, z_4 = x_2^2, z_5 = x_1 x_2 \\ z = [z_1, z_2, z_3, z_4, z_5] \end{gathered} \tag{2.53}$$

5 boyutlu z uzayında oluşturulacak doğrusal diskriminantlar; 2 boyutlu x uzayında doğrusal olmayan bir diskriminant olarak tanımlanacak ve aslında burada yapılan, ilk uzayda doğrusal olmayan bir diskriminant ya da regresyon fonksiyonu öğrenmek yerine doğrusal olmayan bir dönüşümle yeni bir uzaya geçip orada doğrusal bir fonksiyon tanımlamaktır. Başka bir deyişle, doğrusal olmayan bir işlemi doğrusal bir fonksiyonla ayırmak mümkündür.

Doğrusal olarak ayrılabilen iki sınıf varsa, bunun için bir diskriminant denklemi kullanmak yeterlidir. İki sınıf için tanımlanan diskriminant denklemlerinden tek bir diskriminant denklemine geçiş yapılabilir:

$$\begin{gathered} g(x) = g_1(x) - g_2(x) \\ = (w_1^T x + w_{10}) - (w_2^T x + w_{20}) \\ = (w_1 - w_2)^T x + (w_{10} - w_{20}) \\ = w^T x + w_0 \end{gathered} \tag{2.54}$$

[111] Matrisin satırları ile sütunlarının karşılıklı yer değiştirilmesiyle elde edilen yeni matris.
[112] Overfitting

Sonuç olarak; elde edilen $g(x)$ fonksiyonuna "0'dan büyük" ve "0'dan küçük eşit" koşulları atanır:

$$\hat{y} = \begin{cases} C_1, eğer\ g(x) > 0 \\ C_2, eğer\ g(x) \leq 0 \end{cases} \quad (2.55)$$

Gösterimde $g(x)$, 0'dan büyükse C_1 sınıfına; 0'dan küçük ya da eşit ise C_2 sınıfına aittir. Bu işlemler yapılırken, diskriminant denklemine bir sigmoid fonksiyonu uygulandığında olasılıksal sınıflandırma modeli elde edilir:

$$y = sigmoid(w^T x + w_0) = sigmoid(g(x)) \quad (2.56)$$

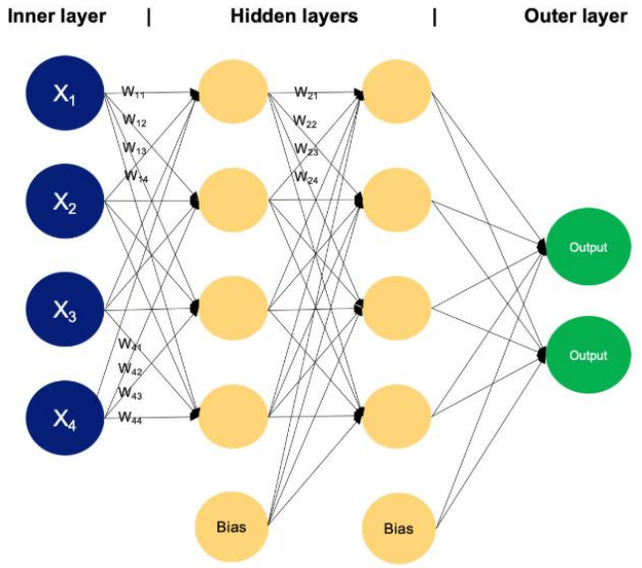
2.8.1.1.8.2. Çok Katmanlı Algılayıcı

Çok katmanlı algılayıcı, sınıflandırma ve regresyon görevlerinde kullanılabilen ve birden fazla nöron katmanından oluşan yapay sinir ağıdır. Ağ mimarisinde döngü yoktur ve ileri beslemeli sinir ağı olarak değerlendirilir (Popescu vd., 2009). Hem model hem de eğitim türü bakımından tek katmanlı sinir ağından farklıdır. Amaç; beyindeki sinir ağlarının modelini oluşturmak ve buna bağlı benzetimler[113] yaparak çeşitli problemlere insansı çözüm üretmektir. Malik ve Solanki'ye göre (2020, s. 152) beyin; görüntü, konuşma tanıma ve öğrenme gibi yetenekleri açısından şu anaki mühendislik ürünlerinden oldukça üstündür. Bu yetenekler, bilgisayar üzerinde gerçekleştirildiğinde ekonomik açıdan önemli olabilecek uygulamalar ortaya çıkmaktadır. Eğer; beynin bu fonksiyonları nasıl yerine getirdiği tam olarak anlaşılırsa, algoritmalar haline getirilip yapay olarak gerçekleştirilebilir.

İnsan beyni, bilgisayardan oldukça farklıdır. Bilgisayarların çoğunlukla tek bir işlemcisi varken, insan beyninde aynı anda çalışan çok sayıda (10^{11}) sinir hücresi (işlemci) bulunur ve her bir işlemci saniyede yaklaşık 200 işlem yapabilir, bu nedenle insan beyni saniyede yaklaşık 20 milyar işlem hesaplayabilir (Pastur-Romay vd., 2017, s. 1647). Her ne kadar ayrıntılı bilinmese de, bu işlemci hücrelerin, bilgisayar işlemcisine göre daha basit ve daha yavaş olduğu söylenebilir. Fakat beyni farklı kılan ve ona yüksek hesap gücü veren şey, bağlantısallığıdır. Beyindeki bir sinir hücresinin, ortalama olarak yaklaşık 10^4 farklı sinir hücresine bağlantısı vardır. Böylece, beyindeki işlem ve belleğin birlikte ağ üzerinde dağıtıldığı düşünülmektedir (Owens ve Tanner, 2017). İşlemler, ağ üzerindeki hücrelerde

[113] Simülasyon

gerçekleştirilirken, bellek bilgileri de aralarındaki bağlantılarda bulunur. Böyle bir yapı, çok katmanlı doğrusal olmayan regresyon[114] gerçeklemelerinde kullanılabilir (Taye, 2023).



Şekil 39. Çok Katmanlı Algılayıcı Mimarisi
Kaynak: (Villanueva, 2024)

Şekil 39'da görüldüğü üzere çok katmanlı algılayıcı; bir giriş katmanı, iki gizli katman ve bir çıkış katmanından oluşmaktadır. Giriş katmanı, bir veri kümesindeki giriş örneklerinden çıkarılan öznitelikleri alır. Giriş katmanındaki her nöron bir özniteliği temsil eder. Giriş katmanındaki nöronlar herhangi bir hesaplama yapmaz; sadece giriş değerlerini ilk gizli katmandaki nöronlara iletir. Gizli katmanlar, giriş verileri üzerinde hesaplamalar yapan ve birbirine bağlı nöronlardan oluşur. Gizli katmandaki her nöron, bir önceki katmandaki tüm nöronlardan girdi alır. Girdiler, ağırlıklarla (w) çarpılır. Ağırlıklar, bir nöronun girdisinin diğerinin çıktısı üzerinde ne kadar etkisi olduğunu belirler. Ağırlıklara ek olarak, gizli katmandaki her nöronun b olarak gösterilen ilişkili bir önyargısı vardır. Önyargı, nörona ek bir girdi sağlayarak çıktı eşiğini ayarlamasına olanak tanır. Ağırlıklar gibi önyargılar da eğitim sırasında öğrenilir. Gizli veya çıktı katmanındaki her nöron için, girdilerinin ağırlıklı toplamı hesaplanır. Bu, her girdinin karşılık gelen ağırlığıyla çarpılmasını, toplanmasını ve önyargının eklenmesini içerir (Amarbayasgala vd., 2021; Lunge, 2024). İleri beslemeli olarak nitelendirilen bu işlemin formülü şu şekilde gösterilir:

l katmanındaki j nöronu için:

$$z_j^{(l)} = \sum_i w_{ij}^{(l)} a_i^{(l-1)} + b_j^{(l)} \tag{2.57}$$

[114] Non-linear regression

Formülde $w_{ij}^{(l)}$, bir önceki katmandaki i nöronundan l katmanındaki j nöronuna giden ağırlığı; $a_i^{(l-1)}$, bir önceki katmandaki nöron i'nin aktivasyonunu; $b_j^{(l)}$ ise önyargıyı[115] ifade etmektedir. Bu formül, önceki katmanın çıktılarının (aktivasyon) ve ağırlıklarının doğrusal bir kombinasyonunu hesaplar, ardından bir önyargı ekler.

Aktivasyon fonksiyonu ise şu şekilde gösterilir:

$$a_j^{(l)} = f(z_j^{(l)}) \tag{2.58}$$

Formülde $a_j^{(l)}$, l katmanındaki j nöronunun aktivasyonunu (çıktısı); $f(z_j^{(l)})$ ise $z_j^{(l)}$'ye uygulanan aktivasyon fonksiyonunu ifade etmektedir. Aktivasyon fonksiyonu doğrusalsızlığı ortaya çıkarır. ReLU, Sigmoid veya Softmax sıklıkla kullanılan aktivasyon fonksiyonlarıdır (Dubey, Singh ve Chaudhuri, 2022).

İleri yayılım sonucunda oluşan hataları hesaplamak ve doğruluğu daha da artırmak için amacıyla geri yayılım parametreleri kullanılarak ağırlıklar ayarlanabilir. Geri yayılım algoritması, ileri yayılım sonucunda oluşan hataları temel alarak her katmandaki ağırlıkların gradyanlarını hesaplar ve bu gradyanlara göre ağırlıklar güncellenerek modelin doğruluğu artırılır. Bu bağlamda kayıp fonksiyonu, tahmin edilen ve gerçek değerler arasındaki farkı ölçer. Sınıflandırma görevlerindeki kayıp fonksiyonunda çapraz entropi kaybı formülü şu şekilde ifade edilmektedir:

$$J(\theta) = -\sum_{i}^{n} y_i \log(\hat{y}_i) \tag{2.59}$$

Formülde $J(\theta)$, kayıp fonksiyonunu; y_i, gerçek sınıf etiketini (doğru sınıf için 1, değilse 0); $\hat{y}_i$ ise modelden tahmin edilen olasılığı ifade etmektedir (Elharrouss vd., 2025).

Regresyon için de ortalama kare hatası[116] kullanılır. Formülü şu şekildedir:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.60}$$

Formülde N, eğitim örneklerinin sayısını; y_i, gerçek değeri; $\hat{y}_i$, modelin tahmin ettiği değeri ve $(y_{i-}\hat{y}_i)^2$ ise her bir örneğin hata karesini ifade etmektedir. Burada hedef, ağırlıkların ayarlanarak kayıp fonksiyonunun en aza indirilmesidir (Geron, 2021).

[115] Bias
[116] MSE

Geri yayılım[117] boyunca hesaplanan bir başka paramete de gradyanların hesaplanmasıdır. Çıktı katmanında hata için şu formül kullanılır:

$$\delta_j^{(L)} = \frac{\partial \mathrm{L}}{\partial z_j^{(L)}} = \frac{\partial \mathrm{L}}{\partial a_j^{(L)}} f'(z_j^{(L)}) \tag{2.61}$$

Formülde $\delta_j^{(L)}$, çıkış katmanı L'deki nöron j için hatayı; L, kayıp fonksiyonunu; $a_j^{(L)}$, çıktı aktivasyonunu; $z_j^{(L)}$, ağırlıklı toplamı; $\frac{\partial \mathrm{L}}{\partial a_j^{(L)}}$, kayıp fonksiyonunun çıktıya göre ne kadar değiştiğini; $f'z_j^{(L)}$ ise aktivasyon fonksiyonunun türevini ifade etmektedir. Türev, z değeri değiştiğinde nöronun çıktısının ne kadar değiştiğini gösterir. Formül, çıkışın ne kadar yanlış olduğunu söyleyen hatayı hesaplamaktadır (Goodfellow, Bengio ve Courville, 2016/2018).

Gizli katmanlara yayılan hata ise şu şekilde hesaplanır:

$$\delta_j^{(l)} = f'(z_j^{(l)}) \sum_k w_{kj}^{(l+1)} \delta_k^{(l+1)} \tag{2.62}$$

Formülde $\delta_j^{(l)}$, gizli katman l'deki nöron j için hatayı; $\delta_k^{(l+1)}$, bir sonraki katmandan $(l+1)$ gelen hatayı; $w_{kj}^{(l+1)}$, l katmanındaki j nöronundan katman $(l+1)$'deki k nöronuna giden ağırlığı; $z_j^{(l)}$, nöron j için ağırlıklı toplam girdiyi; $f'z_j^{(l)}$ ise $z_j^{(l)}$'deki aktivasyon fonksiyonunun türevini ifade etmektedir. Bu formül ise hatayı gizli katmanlara geri yayar (Goodfellow, Bengio ve Courville, 2016/2018).

Gradyan inişi[118] kullanılarak ağırlıkların güncellenmesi ise şu şekilde olur:

$$w^{t+1} = w^t - a\nabla \mathrm{J(w)} \tag{2.63}$$

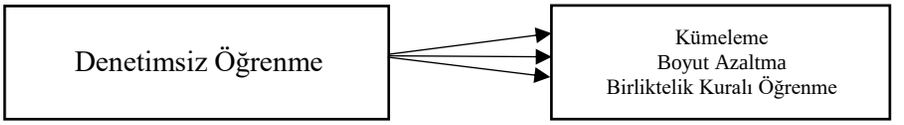
Formülde w^{t+1}, $t+1$ iterasyonunda güncellenen ağırlığı; w^t, t iterasyonundaki geçerli ağırlığı; a, öğrenme oranını; $\nabla \mathrm{J(w)}$ ise w ağırlıklarına ait kayıp fonksiyonunun gradyanını ifade etmektedir. Bu formül ise ağırlıkları, kaybı azaltacak şekilde günceller (Goodfellow, Bengio ve Courville, 2016/2018).

2.8.1.2. Denetimsiz Öğrenme

Her ne kadar makine öğrenmesi uygulamalarının çoğu, günümüzde denetimli öğrenmeye dayalı olsa da[119] kullanılabilir verinin büyük bir çoğunluğu etiketsizdir (Geron, 2021, s. 245). Denetimsiz öğrenme, etiketlenmemiş verilerle ilgilenen bir makine öğrenmesi türüdür.

[117] Back propagation
[118] Gradient Descent
[119] Çünkü yatırımların çoğu bu alana yapılmaktadır.

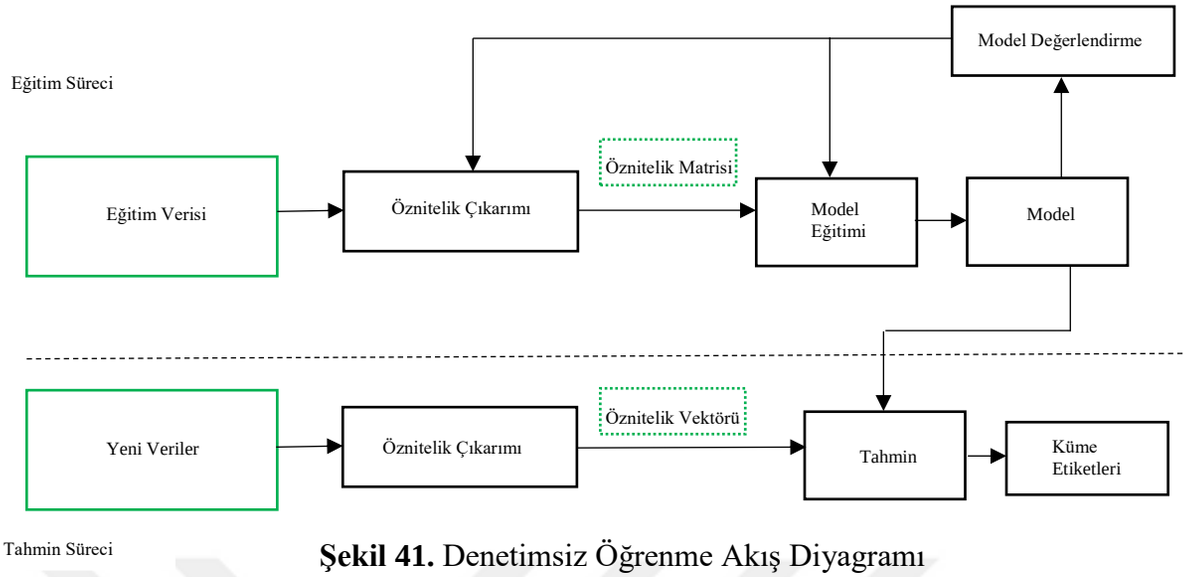
Denetimli öğrenme algoritmalarında veriler, belirli bir kategori veya sonuçla etiketlenirken, denetimsiz öğrenmede verilerin anlamına dair herhangi bir ön bilgi olmadan örüntüler ve ilişkiler bulunur. Başka bir deyişle, denetimsiz makine öğrenmesi algoritmaları, herhangi bir insan müdahalesi olmadan gizli örüntüleri ve verileri bulur, yani modele önceden çıktı verilmez. Eğitim modeli yalnızca girdi parametre değerlerine sahiptir ve grupları veya kalıpları kendi başına keşfeder (Wang ve Biljecki, 2022).



Şekil 40. Denetimsiz Öğrenme Türleri

Denetimsiz öğrenme, Şekil 40'ta gösterildiği gibi kümeleme, boyut azaltma ve birliktelik kuralı öğrenme olmak üzere 3 farklı türde yapılabilmektedir. Kümeleme algoritmalarına örnek olarak k-ortalamalar[120], polinomal ve hiyerarşik kümeleme verilebilir. Boyut azaltma algoritmalarına örnek olarak da temel bileşenler analizi ve çekirdek temel bileşenler analizi verilebilir. Birliktelik kuralı öğrenmesinde ise apriori algoritması örnek olarak verilebilir. K-ortalamalar ve hiyerarşik kümeleme gibi algoritmalar, benzer veri noktalarını içsel özelliklerine göre gruplandırır. Temel bileşen analizi, temel bilgileri korurken verilerin karmaşıklığını azaltmaya yardımcı olur. Denetimsiz öğrenme algoritmalarında, doğruluğu direkt olarak ölçebilmek için etiketli çıktılar olmadığından net değerlendirme metrikleri yoktur. Bu nedenle sonuçları değerlendirmek daha zordur ve dolaylı yollarla yapılır. Araştırmacılar, öğrenilen kalıpların etkinliğini değerlendirmek için genellikle görselleştirme teknikleri (dağılım grafikleri veya dendrogramlar) ile küme doğrulama yöntemleri (Sihouette skoru ve Davies-Bouldin indeksi vb.) ile doğruluk skorunu ölçmeye çalışır (Palacio-Nino ve Berzal, 2019). Denetimsiz öğrenme algoritmaları; görüntü işleme (Samiappan vd., 2018; Qiao, 2022; Ördek ve Borgianni, 2023), doğal dil işleme (Solan vd., 2005; Méndez-Cruz vd., 2024), veri sıkıştırma (Grzymala-Busse ve Than, 1993; Peñaranda, Reaño ve Silla, 2022), veri kümeleme (Greene, Cunningham ve Mayer, 2008; Chander ve Vijaya, 2021) ve anomali tespiti (Kızılkaya, 2019; Bouman, Bukhsh ve Heskes, 2024) gibi görevlerde kullanılmaktadır.

[120] k-Means



Şekil 41. Denetimsiz Öğrenme Akış Diyagramı
Kaynak: (Ghareeb, vd., 2022, s. 5)

Eğitim verisi çoğunlukla veritabanları, sensörler, günlük dosyaları (loglar) veya sosyal medya gibi çeşitli kaynaklardan elde edilip, model eğitimi öncesinde verinin analize uygun hâle getirilmesi amacıyla bir dizi ön işleme adımı gerçekleştirilir. Bu adımlar; eksik veya hatalı değerlerin tespiti ve düzeltilmesi, uygun veri türlerine dönüştürme, aykırı değerlerin yönetimi ve ölçekleme işlemlerini içerir. Öznitelik çıkarımı aşamasında, etiketlenmiş veri bulunmadığı için, algoritmalar veri setinde yer alan örnekler arasındaki yapısal benzerlikleri, örüntüleri ya da gizli yapıları kendiliğinden keşfetmeye çalışır. Öznitelik seçimi sürecinde ise, yüksek boyutluluk problemine karşı etkili olacak biçimde, anlamlı ve bilgi açısından zengin değişkenlerin belirlenmesi için çeşitli istatistiksel yöntemler kullanılabilir. Bu hazırlık sürecinin ardından, uygun denetimsiz öğrenme algoritmaları (örneğin, kümeleme veya boyut indirgeme yöntemleri) veri setine uygulanır. Model, önceden tanımlanmış sınıf etiketleri olmadan, veri içindeki doğal yapıları keşfederek öğrenme sürecini gerçekleştirir. Böylece, insan müdahalesi olmadan, veri setindeki örüntüler, benzerlikler veya yapılar ortaya çıkarılır.

Denetimsiz öğrenme için kullanılan etiketsiz veriler X[121], Instagram, Reddit ve Facebook gibi sosyal medya platformlarında bulunabilir. Bu etiketsiz verilerin gizli kalmış anlamlarını keşfedebilmek için, sistemin bulduğu kalıplara veya kümelere göre bu verileri sınıflandıran bir algoritmaya ihtiyaç duyulmaktadır. Kümeleme analizi, veri setlerindeki grupları veya gizli kalmış ilişkileri açığa çıkarmak için sıklıkla kullanılmaktadır (Oyewole ve Thopil, 2023).

[121] Twitter

Denetimsiz öğrenme algoritmaları da benzer dijital arşiv materyallerinin (veri, görüntü vb.) kümelenmesinde ve anomali tespitinde kullanılabilmektedir.

2.8.1.2.1. k-Ortalamalar

k-Ortalamalar; genellikle çok boyutlu bir veri setinin, önceden belirlenmiş bazı kriterlere göre birbirlerine benzeyecek şekilde gruplara (kümelere) ayrıldığı ve bu şekilde sınıflandırıldığı denetimsiz bir makine öğrenmesi algoritmasıdır (Oti vd. 2021, s. 64). Başka bir deyişle bu algoritma, veriyi k kümeye böler ve kümeler içindeki varyansı en aza indirip küme merkezlerini yinelemeli olarak iyileştirir. Ayrıca, merkez[122] tabanlı bir öğrenme algoritması olarak da bilinir (Mawati, Sumertajaya ve Afendi, 2014).

Algoritma, öncelikle $\mu_1, \mu_2, \ldots \mu_k$ k başlangıç küme merkezlerinin atanmasıyla başlar. Ardından her bir veri noktası en yakın merkeze atanır, sonra güncelleme adımında merkezler, o kümeye atanmış tüm noktaların ortalaması olarak yeniden hesaplanır ve merkezler sabitlenene kadar atama ve güncelleme adımları tekrar tekrar uygulanır. Veri noktalarının en yakın merkeze atanmasında mesafeler hesaplanır. Her bir veri noktası x_i, Öklid mesafesine göre merkez noktası en yakın olan kümeye atanır:

$$c_i = arg_{k\in\{1,2,\ldots,\mathrm{k}\}} \min \|x_i - \mu_k\|^2 \tag{2.64}$$

Formülde c_i, x_i noktasına atanan kümeyi; μ_k, k kümesinin ağırlık merkezini ve $\|x_i - \mu_k\|^2$ ise Öklid mesafesinin karesini ifade etmektedir. Hesaplamalar sonunda minimum mesafeyi sağlayan küme indeksi (k) bulunmuş olur (Singh, 2021).

Her bir küme k için yeni ağırlık merkezi, o kümeye atanan tüm noktaların ortalaması olarak hesaplanır:

$$\mu_k = \frac{1}{|S_k|} \sum_{x_i \in S_k} x_i \tag{2.65}$$

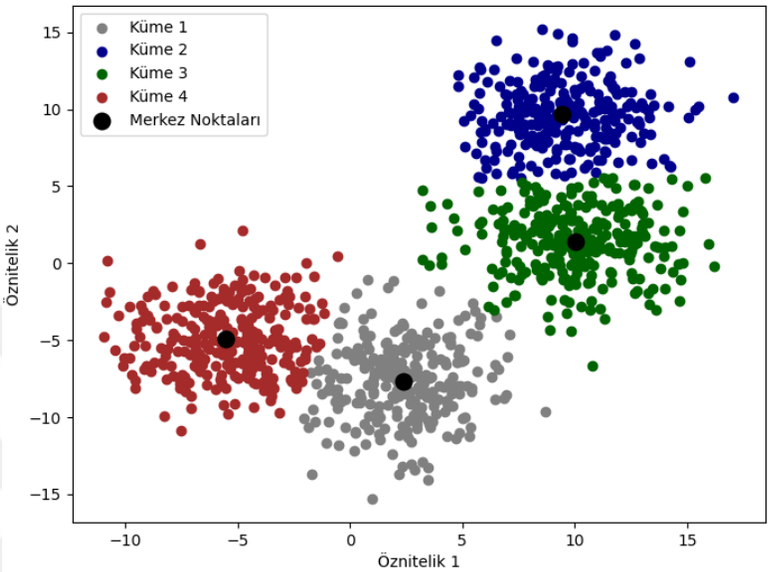
Formülde S_k, k kümesine atanan tüm noktaların kümesini; $|S_k|$, k kümesindeki nokta sayısını ve x_i ise k kümesindeki veri noktalarını ifade etmektedir. Bu adım ise, algoritmanın yeniden atama ve merkez güncelleme döngüsünün bir parçasıdır. İşlem, merkezler sabitlenene kadar tekrarlanır.

k-Ortalamalar, küme içi kareler toplamını en aza indirir, bu aynı zamanda bozulma[123] olarak da bilinir. Bu durumda amaç fonksiyonu şu şekilde formüle edilir (Ding vd., 2016):

[122] Centroid
[123] Distortion

$$J = \sum_{k=1}^{K} \sum_{x_i \in S_k} \|x_i - \mu_k\|^2 \quad (2.66)$$

Formülde J toplam bozulmayı ve $\|x_i - \mu_k\|^2$, bir nokta ile ona atanmış küme merkezi arasındaki Öklid uzaklığının karesini ifade etmektedir. Algoritma, J'nin önemli ölçüde azalmayı bıraktığı noktada yakınsamaktadır.



Şekil 42. k-Ortalamalar Grafiği

Şekil 42'de örnek bir k-ortalamalar grafiği gösterilmektedir. Yaklaşık 1000 veri noktası kullanılarak elde edilen diyagramda k değeri 4 seçilerek 4 ayrı veri grubu kümelenmiştir. Her bir kümenin merkez noktası ise siyah nokta ile işaretlenmiştir. Grafikte, kümelerin iyi bir ayrıma tabi tutulduğu görülmektedir. İç içe geçmiş küme grupları da bulunmadığından, modelin, gözlemleri benzerliğe göre oldukça iyi gruplandırdığı söylenebilir.

2.8.1.2.2. Hiyerarşik Kümeleme

Hiyerarşik kümeleme, verileri küme hiyerarşisine göre bölümlendirmek için kullanılan denetimsiz makine öğrenmesi algoritmalarından biridir. Algoritmada herhangi bir küme sayısının önceden belirlenmesi gerekmemektedir (Arslan, 2021, s. 352). Hiyerarşik kümeleme algoritması, çeşitli küme grupları elde etmek için farklı seviyelerde kesilebilen ve dendrogram[124] adı verilen ağaç benzeri bir yapı oluşturur. Toplayıcı[125] ve bölücü[126] olmak üzere iki farklı türü bulunmaktadır. Toplayıcı hiyerarşik kümelemede veri setindeki her bir

[124] Kümeleme algoritmalarında ayrı veri noktalarının çeşitli benzerlik veya mesafe düzeylerinde nasıl birleştirildiğini veya bölündüğünü gösteren diyagram.
[125] Agglomerative
[126] Divisive

eleman, ayrı bir küme olarak kabul edilir. Daha sonra ise benzer özniteliklere sahip kümeler bir araya gelerek tek bir küme elde edilmeye çalışılır. Tüm veri noktaları bir kümeye ait olana kadar en yakın kümeler yinelemeli olarak birleştirilir (Roux, 2018). Toplayıcı hiyerarşik kümeleme adımları şu şekilde sıralanabilir:

- Uzaklık matrisinin hesaplanması (her veri noktası çifti arasındaki benzerlik/benzersizlik ölçüsü),
- Her bir veri noktasının tek bir küme olarak başlatılması,
- Bir bağlantı kriterine göre en yakın iki kümenin birleştirilmesi,
- Uzaklık matrisinin güncellenmesi,
- Sadece bir küme kalana kadar işlemlerin tekrarlanması (Smarika vd., 2015).

Herhangi bir bağlantı yöntemi uygulanmadan önce, tekil veri noktaları arasındaki mesafelerin ölçülmesi gerekmektedir. Yaygın olarak ölçüm metrikleri arasında Öklid, Manhattan ve Kosinüs benzerliği mesafesi bulunmaktadır. İlgili metriklerin formülleri sırasıyla şu şekilde açıklanabilir:

$$d(A,B) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2} \tag{2.67}$$

Formülde (A, B), iki veri noktasını; (A_i, B_i), A ve B'nin öznitelik değerlerini; n ise öznitelik sayısını ifade etmektedir. Bu formül, iki nokta arasındaki doğrusal uzaklığı hesaplar.

$$d(A,B) = \sum_{i=1}^{n} |A_i - B_i| \tag{2.68}$$

Manhattan mesafesi, iki nokta arasındaki eksen mutlak farklarının toplamını verir.

$$\cos(A,B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2.69}$$

Kosinüs benzerliği ise, iki vektör arasındaki açısal benzerliği ölçer, özellikle metin madenciliği ve yüksek boyutlu verilerde kullanılır.

Hiyerarşik kümelemede bağlantı kriterleri ise noktalar arasındaki ikili uzaklıklara dayalı olarak kümeler arasındaki uzaklığın nasıl ölçüleceğini tanımlar. Bağlantı kriterleri; tekil bağlantı, tam bağlantı, ortalama bağlantı, ağırlık merkezi bağlantı ve Ward bağlantı yöntemleri olarak sınıflandırılabilir (Vijaya, Sharma ve Batra, 2019).

Tekil bağlantı iki küme arasındaki mesafeyi, bir kümenin herhangi bir üyesi ile diğer kümenin herhangi bir üyesi arasındaki minimum mesafe olarak tanımlar ve formülü şu şekilde ifade edilir:

$$d_{tekil\ bağlantı}(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y) \tag{2.70}$$

Formülde C_i ve C_j iki kümeyi; $d(x, y)$ ise x ve y noktaları arasındaki mesafeyi ifade etmektedir.

Tam bağlantı ise farklı kümelerde bulunan noktalar arasındaki maksimum mesafeyi kullanır:

$$d_{tam\ bağlantı}(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y) \tag{2.71}$$

Tam bağlantı, en kötü durumdaki (en uzak) mesafeyi dikkate aldığı için daha sıkı kümeler oluşturma eğilimindedir.

Ortalama bağlantı kümeler arasındaki mesafeyi, her kümeden bir tane olmak üzere tüm nokta çiftleri arasındaki ortalama mesafe olarak hesaplar ve formülü şu şekilde ifade edilir:

$$d_{ortalama\ bağlantı}(C_i, C_j) = \frac{1}{|C_i|\,|C_j|} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y) \tag{2.72}$$

Formülde $|C_i|$ ve $|C_j|$, C_i ve C_j kümelerindeki nokta sayısını ifade etmektedir. Bu yöntem, tekil bağlantı ile tam bağlantı arasında bir denge kurar.

Ağırlık merkezi bağlantı, iki kümenin ortalama vektörleri arasındaki mesafeyi hesaplar ve formülü şu şekilde ifade edilir:

$$\mu_{C_i} = \frac{1}{|C_i|} \sum_{x \in C_i} x\,,\ \mu_{C_j} = \frac{1}{|C_j|} \sum_{y \in C_j} y \ \ ve\ d_{ağırlık\ merkezi\ bağlantı}(C_i, C_j) = d(\mu_{C_i}, \mu_{C_j}) \tag{2.73}$$

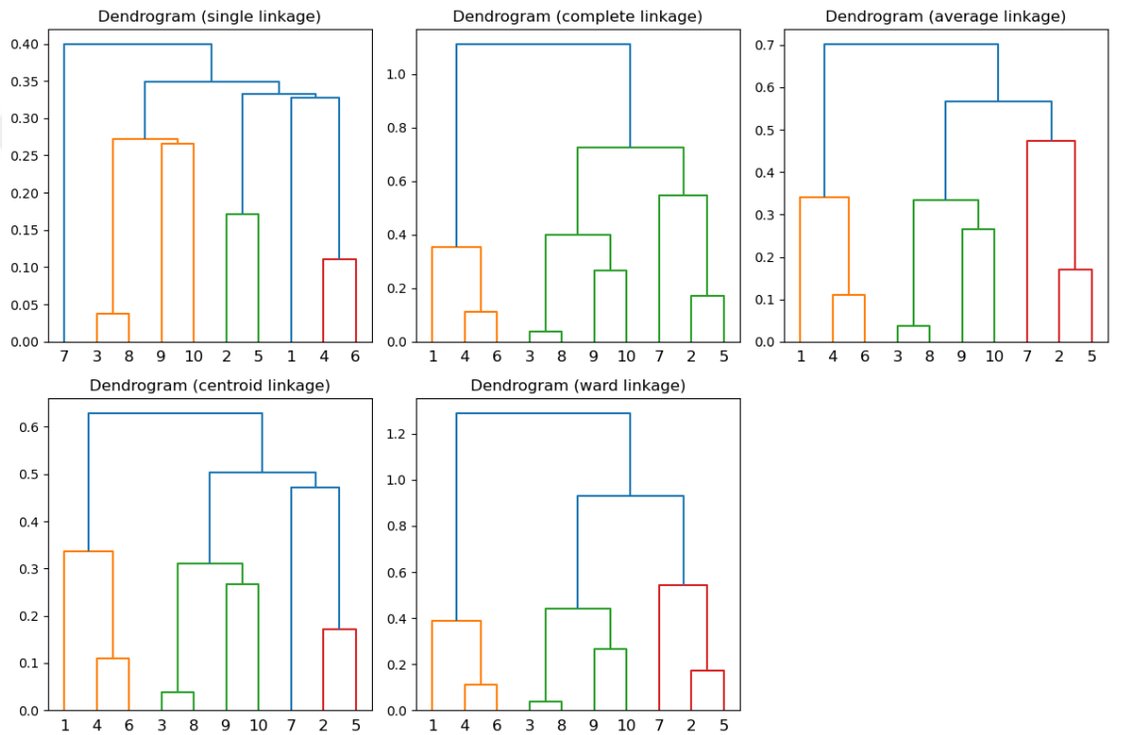
Formülde μ_{C_i}ve μ_{C_j}, C_i ve C_j kümelerinin merkezlerini ifade etmektedir.

Ward bağlantı yöntemi ise toplam küme içi varyansı en aza indirir. C_i ve C_j kümeleri birleştirilirken, toplam küme içi kareler toplamındaki artış hesaplanır. Birleştirme kriteri, varyansta en küçük artışa neden olan çifttir:

$$\Delta(C_i, C_j) = \frac{|C_i||C_j|}{|C_i|+|C_j|}\, d^2(\mu_{C_i}, \mu_{C_j}) \tag{2.74}$$

Formülde $\Delta(C_i, C_j)$, C_i ve C_j kümeleri birleştirildiğinde küme içi varyanstaki artışı; $|C_i|$ ve $|C_j|$, $C_i$ve C_j kümelerindeki hacmi, yani nokta sayısını; μ_{C_i}, μ_{C_j}, kümelerin merkezlerini ve $d^2(\mu_{C_i}, \mu_{C_j})$ ise ağırlık merkezleri arasındaki Öklid uzaklığının karesini ifade etmektedir. Bu yöntem, nispeten eşit büyüklükte kümeler oluşturma eğilimindedir ve kümeler küresel bir şekle sahip olduğunda daha etkili olmaktadır.

Dendrogram, hiyerarşik kümelemede birleştirme işlemini gösteren ağaç benzeri bir diyagramdır. Dalların yüksekliği, kümelerin birleştirildiği mesafeyi temsil eder ve seçilen bir yükseklikte kesilerek farklı küme grupları oluşturulabilir.



Şekil 43. Hiyerarşik Kümeleme Grafiği

Şekil 43'te hiyerarşik kümelemenin farklı uzaklık ölçüm metrikleri ile hesaplanan değerlerinin örnek dendogramları görülmektedir. Her bir ölçüm metriğinin farklı sayıda dallara ayrıldığı görülebilir. Grafiklerde ortalama-merkezi ile tam-ward bağlantı çiftlerinin hemen hemen aynı bölümlendirmelere sahip olduğu görülebilirken, hesaplama farklılıkları nedeniyle kümelerin boyut ve renk bağlantıları değişmektedir.

Bölücü[127] hiyerarşik kümeleme ise toplayıcı kümelemenin tam tersidir. Bu yöntemde daha küçük kümeleri birleştirmek yerine algoritma, tüm veri noktalarını içeren tek bir büyük küme ile başlar ve ardından veri noktaları, her biri tek bir gözlem içerecek şekilde veya bir durdurma

[127] Divisive

kriteri sağlanana kadar özyinelemeli olarak daha küçük alt kümelere ayrılır. Bölücü kümeleme hesaplama açısından maliyetlidir. Dolaysıyla uygulamalarda pek tercih edilmemektedir (Ezugwu vd., 2022).

2.8.1.2.3. DBSCAN[128]

DBSCAN, noktaların yoğunluğuna dayalı olarak verilerdeki kümeleri tanımlamak için kullanılan yoğunluk tabanlı bir kümeleme algoritması olup, rastgele şekle sahip kümeleri otomatik olarak keşfeder ve gürültü noktalarını (aykırı değerler) tanımlar (Bhattacharjee ve Mitra, 2021). DBSCAN iki temel parametreye dayalı olarak çalışır:

- ***ε (Epsilon):*** Noktaların komşu olarak kabul edildiği yarıçap,
- ***MinPts (Minimum Noktalar):*** Yoğun bir bölge (küme) oluşturmak için gereken minimum nokta sayısı.

DBSCAN algoritmasında çekirdek nokta, kendi ε (epsilon) yarıçapındaki komşuluğunda, kendisi de dahil olmak üzere en az MinPts kadar noktaya sahip olan veri noktasıdır. Sınır noktası ise bir çekirdek noktasının ε komşuluğunda yer alan, ancak kendi komşuluğunda MinPts sayısını sağlayamadığı için çekirdek nokta olamayan noktadır. Gürültü ya da aykırı değer olarak adlandırılan noktalar ise herhangi bir çekirdek noktanın ε komşuluğunda yer almayan ve sınır nokta özelliği de göstermeyen veri noktalarıdır (Schubert vd. 2017, s. 2).

Bir p noktasının ε komşuluğu şu şekilde tanımlanabilir:

$$N_{\varepsilon}(p) = \{q \in D \mid d(p,q) \leq \varepsilon\} \tag{2.75}$$

Eşitlikte D, veri setini; $d(p,q)$, mesafe fonksiyonunu; ε ise komşuluk yarıçapını ifade etmektedir. Bu formül p'den ε uzaklıktaki tüm q noktalarını bulur. Başka bir deyişle, p'nin komşuluk alanı belirlenmiş olur.

Bir p noktası ise, aşağıdaki durumlarda çekirdek nokta olarak kabul edilir:

$$|N_{\varepsilon}(p)| \geq MinimumNokta \tag{2.76}$$

Burada $|N_{\varepsilon}(p)|$, p'nin ε komşuluğundaki nokta sayısını ve $MinimumNokta$ ise yoğunluk için gereken kullanıcı tanımlı minimum nokta sayısını ifade etmektedir. Eğer bir nokta bu koşulu sağlıyorsa, bir kümenin temelini oluşturur.

Bir q noktası, aşağıdaki durumlarda bir çekirdek nokta p'den doğrudan yoğunluğa erişebilirdir:

$$q \in N_{\varepsilon}(p) \; ve \; |N_{\varepsilon}(p)| \geq MinimumNokta \tag{2.77}$$

[128] Density-Based Spatial Clustering of Applications with Noise

Başka bir deyişle, doğrudan yoğunluğa erişebilirlik tek yönlü olup sadece çekirdek noktadan diğer noktalara doğru tanımlıdır.

Bir q noktası, p noktasından yoğunluk-erişilebilir ise, aşağıdaki koşulları sağlayan $p_1, p_2, \dots p_n$ dizisi vardır:

$$p_1 = p, p_n = q \; ve \; her \; p_{i+1} \; noktası, p_i \; noktasından \; doğrudan \; yoğunluğa \; erişilebilirdir \quad (2.78)$$

Başka bir ifadeyle bu koşul; kümelerin yoğunluk bağlantılı noktaların bir araya getirilmesiyle oluşturulduğu anlamına gelmektedir. Yani, bir çekirdek noktadan başlayarak birbirine zincirleme şeklinde doğrudan bağlı noktalardan oluşan bir yol varsa, hedef nokta yoğunluk erişilebilir kabul edilir (Ester vd., 1996).

İki p ve q noktası, p ve q'nun her ikisinin de o noktasından yoğunluğa erişilebilir olduğu üçüncü bir o noktası varsa yoğunluk bağlantılı olarak kabul edilir.

$$\exists o \in D \; öyle \; ki \; p \; ve \; q, o \; noktasından \; yoğunluğa \; erişilebilirdir \quad (2.79)$$

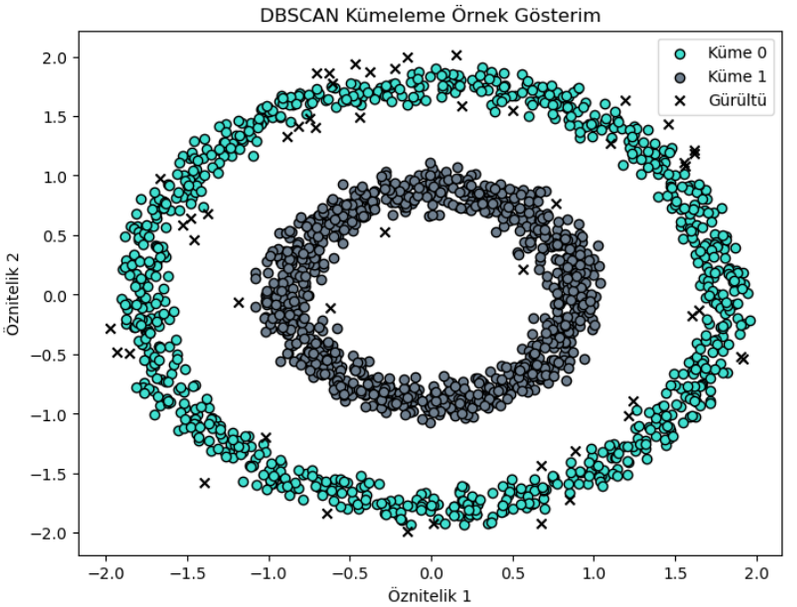
Bu koşul, doğrudan erişilebilir olmasa bile p ve q noktalarının aynı üçüncü bir merkezden (çekirdek nokta) erişilebilir olmaları durumunda aynı kümeye ait kabul edilebileceğini göstermektedir (Lasek, t.y.).

Eğer bir nokta p, herhangi bir çekirdek noktadan yoğunluğa erişilebilir değilse gürültü (aykırı değer) olarak kabul edilir:

$$p \notin \bigcup_{tüm \; kümeler} Küme \quad (2.80)$$

Bu gösterim, p noktasının hiçbir kümeye ait olmadığını ve bu nedenle algoritma tarafından küme dışı olarak işaretlendiğini ifade etmektedir.

DBSCAN; küme sayısını belirtmeden kümeleri otomatik olarak bulması, k-ortalama algoritmasından farklı olarak rastgele şekillerdeki kümeleri tespit edebilmesi ve gürültü noktalarını tanımlaması gibi özellikleri nedeniyle sıklıkla kullanılmaktadır (Hu vd., 2021).



Şekil 44. DBSCAN Kümeleme Grafiği

Şekil 44'te DBSCAN algoritması kullanılarak oluşturulan görsel bulunmaktadır. Burada dikkat edilmesi gereken nokta; kümelemenin otomatik olarak bulunması ve desen şeklinin dairesel olmasıdır. ε değeri 0,1 seçilerek ve gürültüler eklenerek görselin daha anlaşılır olması hedeflenmiştir.

2.8.1.2.4. Temel Bileşen Analizi[129]

Temel bileşen analizi, denetimsiz öğrenmede kullanılan bir boyut azaltma tekniğidir. Yüksek boyutlu verileri mümkün olduğunca fazla varyansı koruyarak daha düşük boyutlu bir uzaya dönüştürür. Bu özelliği sayesinde görselleştirmede, gürültü azaltmada ve modellerin verimliliğini artırmada sıkıklıka kullanılmaktadır (Oja, 2002). Temel bileşen analizi, verideki varyasyonu en fazla yakalayan yeni eksenler (yani temel bileşenler) bularak çalışır. Bu bileşenler korelasyonsuzdur ve açıkladıkları varyans miktarına göre sıralanır. Algoritma şu adımlardan oluşur:

- Verilerin standardizasyonu,
- Kovaryans matrisininin hesaplanması,
- Özdeğerlerin ve özvektörlerin hesaplanması,
- Ana bileşenlerin sıralanması ve seçilmesi,
- Verilerin yeni uzaya dönüştürülmesi (Greenacre vd., 2023).

[129] Principal Component Analysis-PCA

Temel bileşen analizi uygulanmadan önce, her bir özelliğin ortalaması 0 ve varyansı 1 olacak şekilde veri seti standartlaştırılabilir. Böylece tüm özelliklerin modele eşit şekilde katkıda bulunması sağlanmış olur:

$$X_{\text{ölçeklendirilmiş}} = \frac{X - \mu}{\sigma} \tag{2.81}$$

Formülde X, orijinal veri matrisini ($m \times n$ boyutlarında, burada m örnek sayısı ve n öznitelik sayısıdır); μ, her bir özniteliğin ortalamasını ve σ ise her bir özniteliğin standart sapmasını ifade etmektedir.

Kovaryans matrisi kullanılarak yapılan korelasyon şu şekilde hesaplanır:

$$C = \frac{1}{m-1} X^T_{\text{ölçeklendirilmiş}} X_{\text{ölçeklendirilmiş}} \tag{2.82}$$

Formülde C, $n \times n$ boyutlarındaki kovaryans matrisini; $X^T_{\text{ölçeklendirilmiş}}$, standartlaştırılmış veri matrisinin transpozunu; m ise örnek sayısını ifade etmektedir (Sneha ve Fong, 2023).

Kovaryans matrisindeki her bir C_{ij} elemanı, i ve j özellikleri arasındaki kovaryansı temsil eder:

$$C_{ij} = \frac{1}{m-1} \sum_{k=1}^{m} (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j) \tag{2.83}$$

Formülde C_{ij}, i ve j özelliklerinin kovaryansını; x_{ki}, k. gözlemdeki i. özellik değerini; $\bar{x}_i$, i. özelliğin ortalamasını; m, gözlem sayısını ifade etmektedir. Eğer $i = j$ ise; C_{ii}, i. özelliğin varyansını temsil eder.

Özdeğerler ile özvektörlerin hesaplanması ise şu eşitlikle sağlanır:

$$C_v = \lambda v \tag{2.84}$$

Denklemde C, nxn boyutlarındaki kovaryans matrisini; v, özvektörü[130] (temel bileşen); λ ise özdeğeri[131] (açıklanan varyans) ifade etmektedir.

Özdeğerler azalan sırada sıralanır ve karşılık gelen özvektörler temel bileşenleri oluşturur. k. temel bileşen tarafından açıklanan varyans yüzdesinin formülü aşağıdaki gibidir:

$$\frac{\lambda_k}{\sum_{i=1}^{n} \lambda_i} \tag{2.85}$$

[130] Eigenvector
[131] Eigenvalue

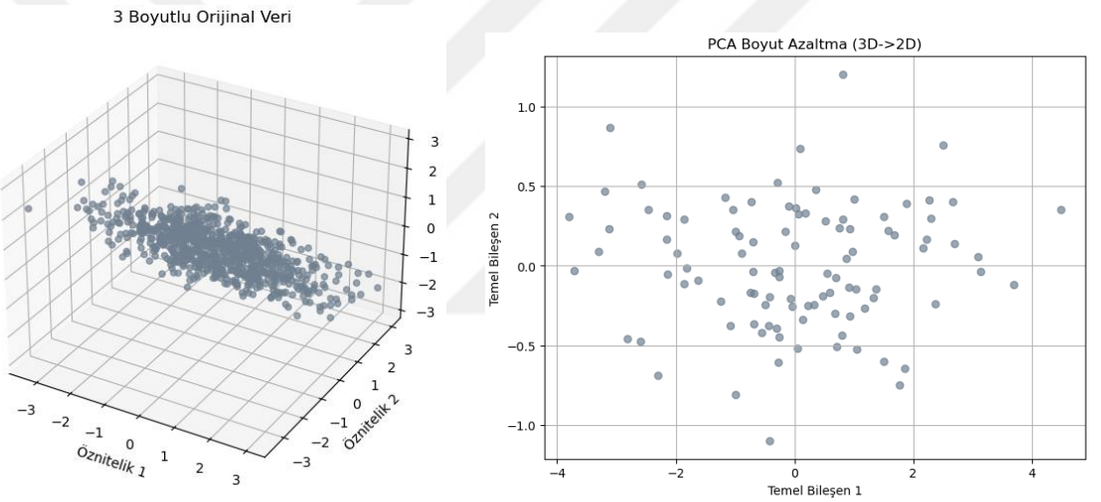
Örnek olarak; boyutluluk azaltılırken, en büyük p temel bileşeni seçildikten sonra varyansın %95'i korunabilir:

$$\sum_{i=1}^{p} \frac{\lambda_i}{\sum_{i=1}^{n} \lambda_i} \approx 0.95 \tag{2.86}$$

Orijinal veri seti kullanılarak da yeni düşük boyutlu uzaya yansıtılmış olur.

$$Z = X_{\text{ölçeklendirilmiş}} \times V_p \tag{2.87}$$

Formülde Z, yeni uzayda mxp boyutlarında dönüştürülmüş veriyi; V_p ise en büyük p özvektörün matrisini (nxp) ifade etmektedir. Bu işlem ile orijinal veri, varyansı en fazla koruyacak şekilde seçilen temel bileşenlere yansıtılır ve verilerin daha düşük boyutlu bir gösterimini sağlar (Sankar, 2021).



Şekil 45. Temel Bileşen Analizi Boyut Azaltma Grafiği

Şekil 45'te 3 boyutlu bir verinin temel bileşen analizi uygulanarak 3 boyuttan 2 boyuta dönüştürüldüğü gösterimler yer almaktadır.

2.8.1.2.5. Birliktelik Kuralı Öğrenmesi

Birliktelik kuralı öğrenmesi, büyük veri kümelerinde gizli kalmış ilişkileri ve örüntüleri keşfetmek için kullanılan denetimsiz öğrenme türüdür. Bazı yayınlarda ilişki kuralı analizi ve ilişki kuralı madenciliği olarak da bilinir (Liu, 2025, s. 317). Genellikle pazar sepeti analizinde, öneri sistemlerinde ve dolandırıcılık tespitinde kullanılan bu öğrenme türü, önceden tanımlanmış etiketler gerektirmemektedir. Bunun yerine algoritma, sık görülen öğe kümelerini bulur ve öğelerin birbirleriyle nasıl ilişkili olduğunu tanımlayan kurallar türetir. Birliktelik kuralı öğrenmesi, diğer makine öğrenmesi türlerine göre daha güçlü bir veri analizi avantajına

sahiptir. Makine öğrenmesindeki veri analizi; otomatik analitik model kurulumu ve verilerdeki ilişkileri analiz etmeye daha az vurgu yapması nedeniyle veri madenciliğinden ayrılmaktadır (Haleem vd., 2022; Rane vd., 2024). Başka bir deyişle veri analizinde veriler arasında gizli ilişkilerin aranması, aynı zamanda örüntülerin belirlenmesi birincil odak noktasıdır. Birliktelik kuralı öğrenmesi, boyut azaltma algoritmasına daha çok benzer (Cassisi vd., 2012).

2.8.1.3. Pekiştirmeli Öğrenme[132]

Pekiştirmeli öğrenme, bir öğrenicinin deneme yanılma yoluyla eylemlerini optimize etmeyi öğrendiği ve ardışık karar almaya odaklanan bir makine öğrenmesi alt alanıdır. Pekiştirmeli öğrenme, bir ödül sinyalini en üst düzeye çıkarmak için ne yapılacağını, yani durumları eylemlere nasıl eşleyeceğini öğrenmek üzerine kuruludur (Ming vd., 2009). Algoritmada, öğreniciye hangi eylemleri yapması gerektiği söylenmez, bunun yerine hangi eylemlerin en fazla ödülü getirdiği denenerek keşfedilmelidir. Zorlayıcı durumlarda, eylemler yalnızca anlık ödülü değil, aynı zamanda bir sonraki durumu ve bunun aracılığıyla tüm sonraki ödülleri de etkileyebilir. Deneme-yanılma süreci ve gecikmeli ödül, pekiştirmeli öğrenmenin en önemli iki ayırt edici özelliğidir (Sutton ve Barto, 2018, s. 1-2). Bu yönleriyle pekiştirmeli öğrenme diğer makine öğrenmesi türlerinden ayrılmaktadır.

Pekiştirmeli öğrenme, genellikle önceden tanımlı verilerin bulunmadığı ortamlarda, otonom bir öğrenicinin sezgisel kararlar alıp eylemlerinin sonuçlarından geri bildirim yoluyla öğrenmesini de esas almaktadır (Ghasemi ve Ebrahimi, 2024). Pekiştirmeli öğrenmede temel kavramlar aşağıda açıklanmıştır:

- ***Durum:*** Bir durum; belirli bir zamanda, öğrenici tarafından algılandığı şekliyle, ortamın belirli bir koşulunu veya yapılandırmasını temsil eder. Durum, öğrenicinin seçimler yapması ve eylemleri seçmesi için sahneyi hazırlar ve bu, bir öğrenicinin meydana gelen her eylem için seçebileceği tüm durumları tanımlar. Örneğin; satrançta bir durum, tahtadaki belirli bir taş düzeni olabilir. Durumlar, bir öğrenicinin içinde bulunabileceği ve kararlar almak için gözlemleyebileceği durumlardır. Durumlar ayrık ve sürekli (örneğin, bir robotun x ve y koordinatları açısından konumu) olabilir (Sallans ve Hinton, 2004; Pandey, Pandey ve Kumar, 2010).
- ***Eylem***: Eylem, bir öğrenicinin çevreyle etkileşim halindeyken yapabileceği olası hareketler veya kararlar kümesidir. Seçili eylem, öğrenicinin mevcut durumlarına ve politikasına göre istenen hedeflere ulaşmak için izlediği stratejinin bir parçasıdır.

[132] Reinforcement Learning

Örneğin, satrançta bir taşı bir kareden diğerine hareket ettirmek bir eylemdir. Durumlara benzer şekilde, eylemler de ayrık ve sürekli olabilir. Satranç örneği, ayrık eylemleri gösterir. Sürekli eylemler için de robot hareketleri verilebilir. Bir robot, her ikisi de sürekli eylem olan koordinatlarını ve hareket hızını değiştirebilir (Sutton ve Barto, 2018; Chandak, 2019).

- ***Politika:*** Politika, çevrenin algılanan durumlarını eylemlere eşleyerek öğrenicinin davranışını yönlendirir ve basit bir fonksiyon, bir arama tablosu veya karmaşık bir hesaplama olabilir. Politikalar stokastik[133] olabilir ve belirli eylemleri gerçekleştirme olasılığını tanımlayabilir. Stokastik bir politika, belirli bir durum için olası eylemler üzerinde bir olasılık dağılımı tanımlar. Tek bir eylem seçmek yerine, eylemleri politika tarafından tanımlanan olasılıklara göre örnekler (Lehmann, 2024; Ghasemi ve Ebrahimi, 2024).
- ***Ödül:*** Ödül, pekiştirmeli öğrenmede oldukça kritik bir faktördür çünkü her bir zaman adımında öğrenici bir hedef sağlar ve öğrenicinin zaman içinde ulaşmayı amaçladığı hem yerel hem de küresel hedefleri tanımlar. Ödüller olumlu olayları olumsuz olaylardan ayırır ve eylemlerin sonuçlarına göre politikaların güncellenmesine yardımcı olur. Ödüller, ortamın durumuna ve gerçekleştirilen eylemlere bağlıdır. Satranç örneğinde, oyunu kazanmak olumlu bir ödüle yol açarken, puan kaybetmek olumsuz bir ödüle yol açar. Ayrıca, oyun berabere bitebilir ve bu da ödül alınmamasına yol açabilir (Guo, 2017; Icarte vd., 2022).
- ***Geçiş Dinamikleri:*** Geçiş dinamikleri fonksiyonu, mevcut durum ve eylem verildiğinde yeni bir duruma ulaşma olasılığını tanımlamaktadır (Ghasemi ve Ebrahimi, 2024).
- ***Çevre Modeli:*** Çevre modeli, çevrenin bir tahminidir ve belirli bir durum ve eylem girdisi verildiğinde neyin mevcut olacağını (bir sonraki durum ve ödül) haritalandırır. Bu modeller, olası gelecekteki olaylara dayanarak hangi eylemlerin yapılması gerektiğini belirler ve planlamaya yardımcı olur. Model kullanan pekiştirmeli öğrenme yaklaşımları model tabanlı yöntemler şeklinde adlandırılırken, sadece deneme-yanılma şeklinde öğrenme ise modelsiz yöntemler olarak adlandırılmaktadır. Bu özellik, soruna uygun algoritmaları seçmek için önemli bir faktördür. Çevrenin dinamikleri, geçiş ve ödül fonksiyonu olarak temsil edilebilir (Viano vd., 2020; Epperlein vd., 2022).

[133] Bir sürecin, rastgele değişkenler ve olasılıksal sonuçlar içermesi durumu. Stokastik sistemler aynı koşullar altında bile farklı sonuçlar üretebilir.

2.8.2. Derin Öğrenme

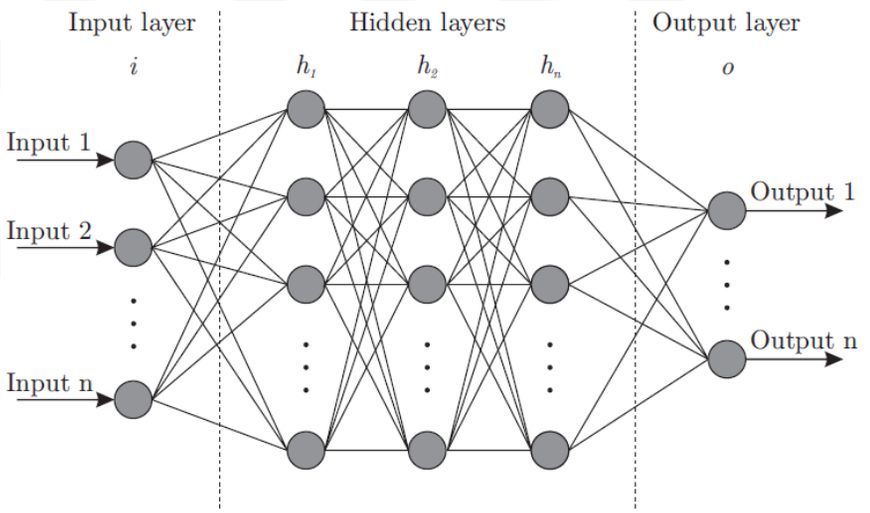
Bir işin bilgisayar aracılığı ile gerçekleştirilebilmesi için yazılım süreçlerine yönelik algoritmasının bilinmesi gerekmektedir. Algoritma; girdiyi çıktıya çevirmek için uygulanabilecek temel komut dizilerinin tamamı olarak tanımlanabilir. Ancak, bazı problemleri çözebilmek için herhangi bir bilinen algoritma olmayabilir. Problem parametreleri, istenen veya istenmeyen durumlarda değişiklik gösterebilir ya da kullanıcıya göre değişkenlik gösteren çeşitli uygulamalar, sabit bir algoritmaya sahip olmayabilir. Hem istenen hem de istenmeyen binlerce örnekten bir sistemin kolayca öğrenmesi, derin öğrenme algoritmaları ile sağlanabilir. Günümüz teknolojisinde veri toplama cihazları büyük ölçüde dijital veri ilettiğinden, verilerin güvenilir bir şekilde erişilebilir, saklanabilir ve işlenebilir olması, derin öğrenme dünyası için bir avantajdır.

Derin öğrenme, makine öğrenmesi alt sınıfı olarak kabul edilmektedir (Kumar ve Manash, 2018). Derin öğrenme algoritmaları, verileri işlerken ve karar almada kullanılmak üzere örüntüler oluştururken insan beynini taklit eder. Klasik bir derin öğrenme ağı, üç veya daha fazla katmana sahip bir sinir ağından oluşur. Ek gizli katmanlara sahip sinir ağlarının kullanımı, yaklaşık tahminler yapabilen tek katmanlı sinir ağlarına kıyasla doğruluk açısından daha iyi sonuçlar sağlamaktadır. Derin öğrenme, üzerinde çalışmak için büyük miktarda veriye ve karmaşık yapısı nedeniyle bu verileri işlemek için çok güçlü donanımlara (işlemci, ekran kartı, RAM vb.) ihtiyaç duyar. Bir derin öğrenme modeli, biyolojik nöronları taklit eden çoklu nöronlardan oluşur. Dünyadaki tüm verilerin %90'ının son 2 yılda üretilmiş olduğu açıklanmıştır (Marr, 2018). Bu oran günümüzde oldukça artmış ve süresi daha da kısalmış durumdadır. Özellikle düzenli ve etiketlenmiş veriler hem arşiv bilimi hem de derin öğrenme için çok önemlidir.

Makine öğrenmesi algoritmalarında öznitelik çıkarma[134] ve seçme[135] işlemleri, modelin başarımını doğrudan etkileyen ve titizlikle yürütülmesi gereken süreçlerdir. Öznitelikler, veri setindeki problemi anlatan önemli özelliklerdir. Makine öğrenmesinde öznitelik çıkarımı ve seçimi genellikle önceden tanımlanmış parametreler aracılığıyla elle gerçekleştirilirken, derin öğrenme modellerinde bu süreç, modelin katmanları boyunca otomatik ve hiyerarşik bir biçimde kendiliğinden öğrenilmektedir. Buradaki kritik nokta ise derin öğrenmede kullanılacak yapay sinir ağı mimarisinin uygun şekilde tasarlanmasıdır. Yapay sinir ağları, insan beyninin çalışma prensiplerinden esinlenerek geliştirilmiş, ağırlıklı bağlantılar

[134] Feature extraction
[135] Feature selection

aracılığıyla birbirine bağlanan ve her biri kendi belleğine sahip işlem birimlerinden oluşan paralel ve dağıtık bilgi işleme yapılarıdır (Elmas, 2021, s. 25). Yapay sinir ağlarında katman sayısı, nöron sayısı, optimizasyon algoritması ve öğrenme oranı gibi hiperparametrelere oldukça dikkatli bir şekilde karar verilmesi gerekmektedir. İlk seferde hiperparametreleri en efektif şekilde seçmek çok zordur. Çünkü veriler ve kullanılan donanım gibi birçok değişken, bu seçimlerden aynı anda etkilenmektedir. Bir hiperparametre; veriden öğrenenen modeller tasarlarken modelin nasıl olması gerektiği, çeşitli koşullar göz önüne alınarak modeli tasarlayan kişiye bırakılan; probleme, veri setine ve donanıma bağlı olarak değişkenlik gösteren parametreler olarak tanımlanabilir (Yadav vd., 2023). Derin öğrenme modelini tasarlarken; makine öğrenmesi modellerine göre çok daha fazla veriye ihtiyaç duyulmaktadır (Degadwala ve Vyas, 2024). Bunun yanında veri seti, gerçek dünya verisini yansıtacak şekilde çeşitli olmalıdır.



Şekil 46. Yapay Sinir Ağı Yapısı
Kaynak: (Bre, Gimenez ve Fachinotti, 2018, s. 1430)

Şekil 46'da klasik bir yapay sinir ağının genel yapısı gösterilmektedir. Sinir ağının modeli üç ana katmandan oluşmaktadır: Giriş katmanı, gizli katmanlar ve çıkış katmanı. Giriş katmanı (i), dış kaynaklardan ham veri alan nöronlardan oluşur. Her bir giriş nöronu, veri setinden bir özniteliği veya değişkeni temsil eder. Ağ, hiyerarşik özellik öğrenmesini sağlayan birden fazla gizli katman (h_1,h_2,...,h_n) içerir. Gizli katmandaki her bir nöron, önceki ve sonraki katmanlardaki tüm nöronlara bağlanarak tam bağlantılı veya yoğun bir katman oluşturur. Bu bağlantılar ağırlıklıdır ve her nöron bir aktivasyon fonksiyonu uygular ve modelin karmaşık örüntüleri öğrenmesi sağlanır. Çıktı katmanı (o) ise, ağın son tahminlerini ürettiği yerdir. Çıktı nöronlarının sayısı problem türüne bağlı olarak değişkenlik gösterir. Sınıflandırma görevleri

için çıktı nöronları sınıf olasılıklarını temsil ederken; regresyon görevleri için sürekli değerler üretilir (Weiss vd., 2022).

Yapay sinir ağı geliştirirken dengesiz veri kümesi, eğitim/doğrulama/test yapısının doğru oranda ayarlanamaması, bias-varyans sorunu[136], aşırı öğrenme[137] ve eksik öğrenme[138] gibi sorunlarla karşılaşılmaktadır (Khoei, Slimane ve Kaabouch, 2023). Bu sorunların giderilmesine ilişkin tezin ilerleyen bölümlerinde bilgiler verilmiştir.

2.8.2.1. Aktivasyon Fonksiyonları

Aktivasyon fonksiyonları, yapay sinir ağlarında özellikle bir giriş sinyalini, çıkış sinyaline dönüştürmek için kullanılır ve bir sonraki katmana giriş olarak beslenir. Başka bir deyişle, aktivasyon fonksiyonu, nöronları aktif veya deaktif etmek için kullanılır. Bir sinir ağının tahmin doğruluğu, kullanılan katman sayısına ve daha da önemlisi kullanılan aktivasyon fonksiyonunun türüne bağlıdır. En sık kullanılan aktivasyon fonksiyonları doğrusal olmayan aktivasyon fonksiyonlarıdır. Sinir ağlarında düğüm[139]; giriş özelliklerinin değerlerini içeren giriş sinyalinin ve bu değerlerin çarpıldığı ağırlıkların ve önyargı değerlerinin toplandığı yerdir (López, López ve Crossa, 2022). Aktivasyon fonksiyonu tam olarak bu noktada işlevini gerçekleştirir. Toplanan ağırlıklı girişi, bir sonraki bağlı nörona veya nöron kümesine, yani bir sonraki katmana geçiren bir çıktı değerine dönüştürür. Bu nedenle, düğümlerin "ateşlenmesinden" sorumlu olan katman, aktivasyon katmanıdır (Rajanand ve Singh, 2024). Fonksiyon giriş değerlerini işledikten sonra, çıktının eşiği geçip geçmediğini kontrol eder ve eğer geçiyorsa, bir sonraki bağlı katmanın nöronlarına gitmesine izin verir. Bu nedenle, aktivasyon fonksiyonu yapay bir nöronda oldukça önemli bir rol oynamaktadır.

2.8.2.1.1. Doğrusal Aktivasyon Fonksiyonu

Doğrusal aktivasyon fonksiyonu, çıktının girdiye doğrudan orantılı olduğu en basit aktivasyon fonksiyonudur. Formülü şu şekilde ifade edilir:

$$f(x) = ax + b \tag{2.88}$$

Formülde x, girdiyi; a, bir sabit sayıyı ve b, önyargıyı ifade etmektedir. En basit haliyle $f(x) = x$, yani çıktının girdiyle aynı olması demektir. Fonksiyon, eğimi a olan düz bir çizgidir, yani doğrusal modellerde kullanılabilir.

[136] Eğitim hatası azalırken test hatasının artması veya tersi.
[137] Overfitting
[138] Underfitting
[139] Node



Şekil 47. Doğrusal Aktivasyon Fonksiyonu

Şekil 47'de örnek bir doğrusal aktivasyon fonksiyonunun grafiği gösterilmektedir. Özellikle regresyon problemlerinde sıklıkla kullanılmaktadır. Fonksiyon düz bir çizgi şeklinde olduğundan, çoklu doğrusal katmanların yığılması, modelin kapasitesini artırmaz. Matematiksel olarak, doğrusal fonksiyonların birleşimi hala doğrusal bir fonksiyondur (Shpilrain ve Sosnovski, 2016). Fakat fonksiyon, karmaşık karar sınırlarını ele alamaz ve görüntü tanıma veya dil modelleme gibi problemlerdeki örüntüleri yakalayamaz. Fonksiyonun kullanıldığı derin sinir ağı, sadece girdinin doğrusal değişikliklerine uyum sağlayabildiğinden sinir ağlarının hatalı çıktılar vermesine neden olur. Bu nedenle, sinir ağlarında doğrusal aktivasyon fonksiyonu yerine çoğunlukla doğrusal olmayan aktivasyon fonksiyonları tercih edilmektedir (Sharma, Sharma ve Athaiya, 2020).

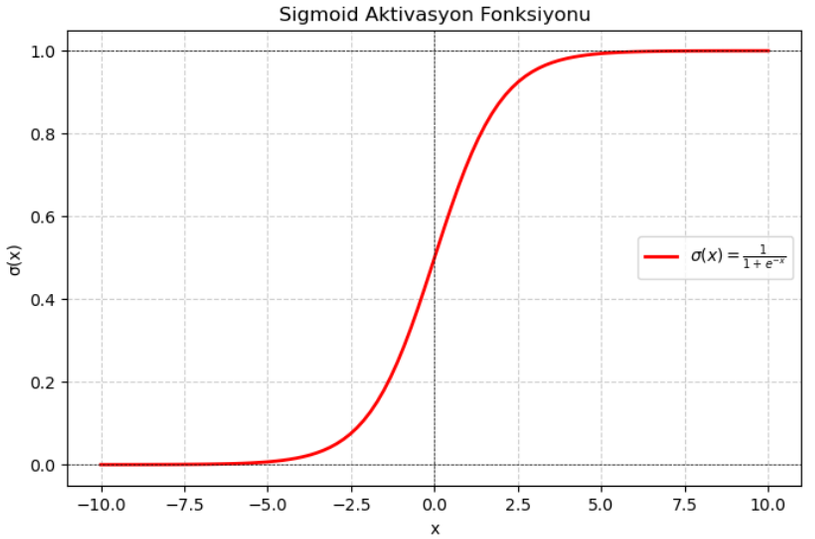
2.8.2.1.2. Sigmoid Aktivasyon Fonksiyonu

Sigmoid aktivasyon fonksiyonu, bir girdiyi 0 ile 1 arasında bir değere eşleyen matematiksel bir fonksiyon olup genellikle ikili sınıflandırma problemlerinde kullanılmaktadır (Hamdan ve Roach, 2022). Formülü şu şekilde gösterilmektedir:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.89}$$

Formülde e, Euler sayısını[140]; x ise modelin girdisini ifade etmektedir. $f(x)$'in etki alanı gerçek sayılar kümesi olup, $-\infty < x < +\infty$ ve aralığı (0,1)'dir. Giriş değeri ne kadar büyükse (daha pozitifse), çıktı değeri 1'e o kadar yakın olur. Giriş değeri ne kadar küçükse (daha negatifse), çıktı değeri 0'a daha yakın olur.

[140] ~2.71



Şekil 48. Sigmoid Aktivasyon Fonksiyonu

Şekil 48'de örnek bir Sigmoid aktivasyon fonksiyonu grafiği gösterilmektedir. Narayan'a göre (1997, s. 70) sürekli değere sahip bir çıktının istendiği görevler için oldukça uygundur. Fonksiyonun değeri her zaman artmaktadır, yani daha büyük girdiler daha büyük çıktılara karşılık gelir. Büyük veya küçük x değerleri için türev 0'a yaklaşır ve bu durum ağırlık güncellemelerinin çok küçük yapılmasına neden olur, başka bir deyişle kaybolan gradyan problemine yol açar. Böyle bir durum, derin sinir ağlarda eğitimi yavaşlatır. Çıkış aralığının (0,1) olması ise gradyan güncellemeleri için dengeli olmayabilir ve daha yavaş yakınsamaya neden olur.

2.8.2.1.3. Hiperbolik Tanjant Aktivasyon Fonksiyonu

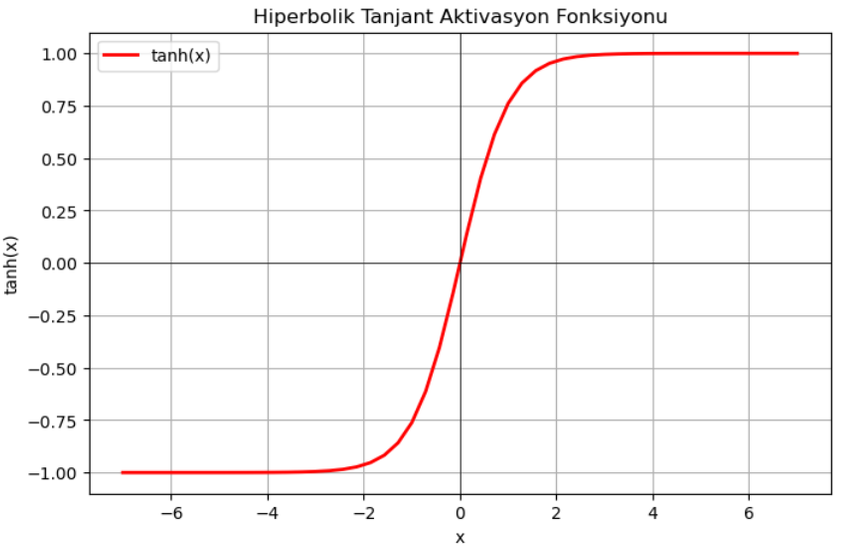
Hiperbolik tanjant aktivasyon fonksiyonu, (-1,1) aralığında değerler üretir. Özellikle sinir ağlarının gizli katmanlarında yaygın olarak kullanılmaktadır. Giriş verilerinin dönüştürülmesi ve gradyan tabanlı optimizasyon hesaplamalarında oldukça kullanışlıdır. Fonksiyon, özellikle özyinelemeli sinir ağlarında[141] tercih edilmektedir. Derin ağlarda kullanımı daha az yaygındır, çünkü çoğunlukla ReLU aktivasyon fonksiyonu daha iyi gradyan yayılımı nedeniyle tercih edilmektedir (Alkhouly, Mohammed ve Hefny, 2021). Mollahosseini, Chan ve Mahoor (2015) hiperbolik fonksiyonun sigmoidal olduğunu öne sürmektedir. Başka bir deyişle hiperbolik tanjant fonksiyonu, sigmoid fonksiyon eğrisinin genişletilmiş halidir. Dolaysıyla, hiperbolik fonksiyonların negatif girişleri negatif bir çıkışa eşleneceği gibi, sıfıra yaklaşan giriş değerleri de sıfıra yaklaşan çıkış değerlerine eşlenecektir. Böylece, sinir ağı eğitim sırasında kenar

[141] Recurrent Neural Networks

değerlerine takılıp kalmaz (Szandala, 2021). Matematiksel formülü şu şekilde ifade edilmektedir:

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.90}$$

Formülde e, Euler sayısını; x ise x aktivasyon fonksiyonunun girdisini ifade etmektedir. Türev $x = 0$ civarında en yüksek seviyededir ve kenarlara doğru azalır, bu da büyük $|x|$ değerleri için gradyanların kaybolması problemine neden olabilir.



Şekil 49. Hiperbolik Tanjant Aktivasyon Fonksiyonu

Şekil 49'da örnek bir hiperbolik tanjant aktivasyon fonksiyonu grafiği gösterilmektedir. Sigmoid fonksiyonuna benzemekle birlikte, orijine göre simetriktir. Sıfır ortalaması vardır ve bu da eğimleri daha dengeli hale getirerek optimizasyona yardımcı olur. Değer büyürse, pozitif x için, $tahn(x)$ fonksiyonu 1'e yaklaşır. Değer daha da küçülürse fonksiyon -1'e yaklaşır. x'in 0 olduğu durumda ise fonksiyon 0 olur. Sıfır merkezli çıktı, eğitim sırasında önyargı kaymalarını azaltmaya yardımcı olur (Hu, Zhang ve Ge, 2021). Sigmoid fonksiyona kıyasla daha pürüzsüz gradyan inişine sahiptir. Sinir ağlarının daha derin katmanlarda Sigmoid fonksiyona göre daha hızlı öğrenmeye yardımcı olur. Büyük veya küçük girdiler için gradyanların kaybolması sorununa yol açabilmektedir. Ayrıca hesaplama açısından ReLU aktivasyon fonksiyonundan daha maliyetlidir.

2.8.2.1.4. ReLU Aktivasyon Fonksiyonu

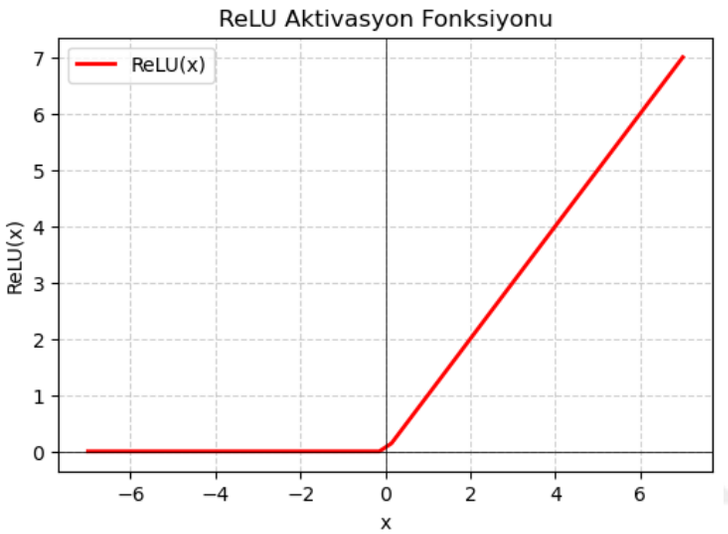
ReLU[142] aktivasyon fonksiyonu, özellikle evrişimli sinir ağlarında en yaygın kullanılan aktivasyon fonksiyonlarından biridir. Yaygın olarak kullanılmasının en önemli nedenlerinden

[142] Rectified Linear Unit - Düzeltilmiş Doğrusal Birim

biri de gradyanların kaybolması sorununa çözüm getirmesidir. Sigmoid ve hiperbolik tanjant aktivasyon fonksiyonları; monoton, türevlenebilir ve zamanla doygunluğa uğradıkları ve bu nedenle de gradyanların kaybolması sorununa yol açtıkları için, ReLU aktivasyon fonksiyonu geliştirilmiştir (Schmidt-Hieber, 2020; Vasanthakumari, Nair ve Krishnappa, 2023). Formülü şu şekilde gösterilir:

$$f(x) = \max(0, x)$$
$$eğer\ x > 0\ ise\ f(x) = x \qquad (2.91)$$
$$eğer\ x \leq 0\ ise\ f(x) = 0$$

Formül negatif değerlerin sıfıra indirgerken pozitif değerlerin sabit kalmasını sağlar.



Şekil 50. ReLU Aktivasyon Fonksiyonu

Şekil 50'de ReLU aktivasyon fonksiyonunun grafiği gösterilmektedir. Fonksiyon, eşikleme işlemi[143] gerektirdiğinden, sigmoid veya tanh fonksiyonlarına göre daha hızlıdır. Birçok nöron sıfır çıkış verir, bu da seyrek aktivasyonlara ve daha iyi genellemeye yol açmaktadır. Sigmoid ve hiperbolik tanjant fonksiyonları, değerleri sınırlı bir aralığa sıkıştırırken; ReLU, değerlerin doğrusal olarak büyümesine izin verir ve bu da derin ağların daha detaylı öğrenilmesine yardımcı olur. ReLU ile ilgili temel sorun, tüm negatif değerlerin sıfıra dönmesidir[144]; bu da modelin verilerden düzgün bir şekilde uyum sağlama yeteneğini azaltır. Bu gibi dezavantajların önüne geçebilmek için Leaky ReLU, Parametrik ReLU (PReLU), Üstel Doğrusal Birim (ELU), ve GELU gibi türev fonksiyonlar geliştirilmiştir (Bai, 2022).

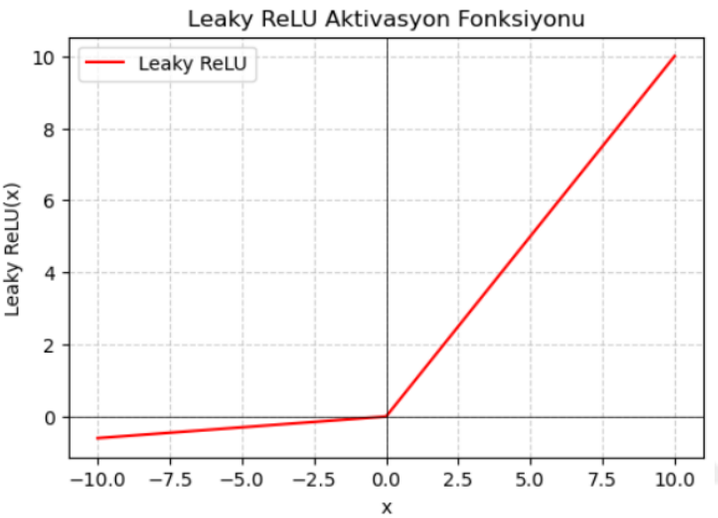
[143] Negatif değerleri sıfıra indirgenmesi.
[144] Ölü nöron problemi - Dead neuron problem

2.8.2.1.5. Leaky ReLU Aktivasyon Fonksiyonu

Leaky ReLU, klasik ReLU fonksiyonunun dezavantajlarından biri olan "ölü nöron" problemine çözüm olarak geliştirilmiş bir aktivasyon fonksiyonudur. Bu aktivasyon fonksiyonu, sıklıkla üretici çekişmeli ağlarda (GAN)[145] kullanılmaktadır. Formülü şu şekilde gösterilir:

$$f(x) = \max(0.01x, x) \tag{2.92}$$

Formül; negatif değerlerin sıfıra ayarlanması yerine, küçük bir doğrusal terim eklenerek (β<1) negatif değerler için düz bir eğim yerine oldukça küçük bir eğimin oluşmasını sağlar (Vallés-Pérez vd., 2023).



Şekil 51. Leaky ReLU Aktivasyon Fonksiyonu

Şekil 51'de, örnek bir Leaky ReLU aktivasyon fonksiyonu gösterilmektedir. Negatif değerler için küçük bir eğime izin vererek, geri yayılım sırasında eğimin korunmasına yardımcı olur ve bu da eğitim sırasında daha kararlı ve daha hızlı bir yakınsamaya yol açar. Fonksiyon, negatif girdileri tamamen göz ardı etmek yerine bu girdilerden de öğrenerek, negatif aktivasyonların anlamlı bilgi taşıyabileceği belirli mimarilerde faydalı olabilir. Negatif girdiler için küçük eğim, özellikle daha derin ağlarda standart ReLU'ya kıyasla aşırı öğrenmeyi azaltmaya yardımcı olabilecek bir düzenleme biçimi sağlar. Tıpkı ReLU gibi, basit eşikleme ve çarpma içerdiğinden hesaplama açısından verimlidir (Maniatopoulos ve Mitianoudis, 2021).

2.8.2.1.6. Parametrik ReLU Aktivasyon Fonksiyonu

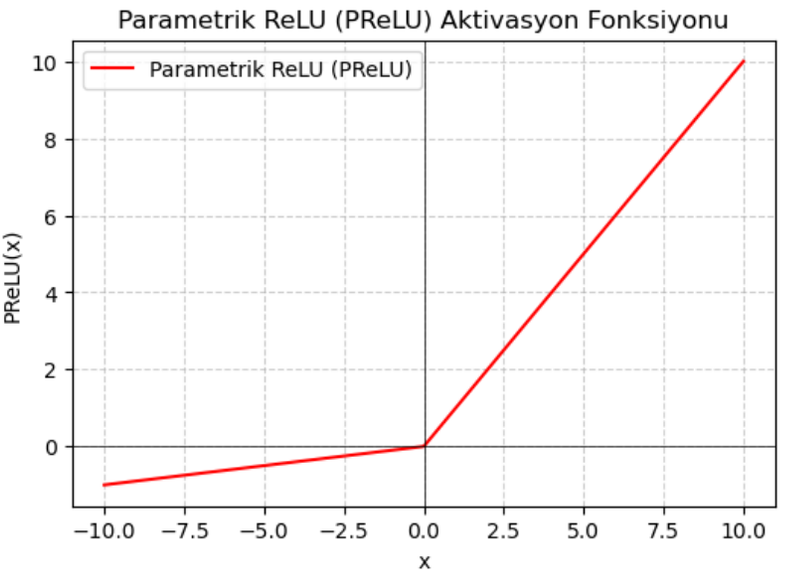
He vd. (2015) tarafından Leaky ReLU fonksiyonunun dinamik bir evrimi olarak geliştirilmiştir. Leaky ReLU fonksiyonuna benzerdir ancak negatif eğim, eğitim sırasında öğrenilir. Leaky

[145] Generative Adversarial Network

ReLU'dan daha esnektir. Formülü şu şekilde gösterilir:

$$f(x) = \max(\alpha x, x) \tag{2.93}$$

Formülde negatif giriş değerlerinin eğimi α değeri aracılığıyla sağlanmaktadır. Geri yayılım sırasında α değeri ayarlanır ve en iyi değerin öğrenilmesi sağlanır. Fonksiyon, eğitim sırasında her birim için sızıntı parametresini dinamik olarak öğrenerek ağın uzmanlaşmasını kolaylaştırır. ImageNet üzerindeki denemelerde, PReLU kullanan ağların ReLU ve Leaky ReLU fonksiyonlarını kullanan ağlardan daha iyi performans gösterdiği ortaya koyulmuştur (Russakovsky vd., 2014; Pusztahazi, Eigner ve Csiszar, 2024).



Şekil 52. PReLU Aktivasyon Fonksiyonu

Şekil 52'de parametrik ReLU aktivasyon fonksiyonu gösterilmektedir. Şekilde görüldüğü üzere, Leaky ReLU fonksiyonuna göre negatif alanda daha fazla esneklik sağlanmaktadır.

2.8.2.1.7. ELU[146] Aktivasyon Fonksiyonu

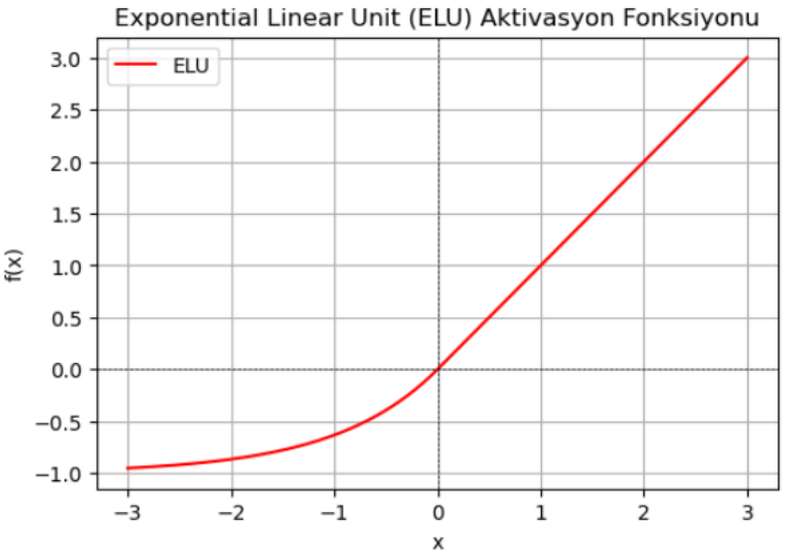
Clevert, Unterthiner ve Hochreiter (2016) tarafından geliştirilen ve doğrusal olmayan yapının korunması aracılığı ile gradyanların kaybolması problemini ele alarak derin sinir ağlarının öğrenme hızını ve doğruluğunu artırmak için tasarlanmış bir aktivasyon fonksiyonudur. ReLU fonksiyonun aksine ELU, negatif girişler için sıfırdan küçük değerler üretir, bu da daha düşük hesaplama karmaşıklığı ile ortalama birim aktivasyonlarını toplu normalleştirme gibi sıfıra yaklaştırmalarına izin verir. Ortalamanın sıfıra doğru kayması, azaltılmış bir önyargı kayması

[146] Exponential Linear Unit

etkisi nedeniyle normal gradyanı birim doğal gradyana yaklaştırarak öğrenmeyi hızlandırır. Formülü şu şekildedir:

$$f(x) = \begin{cases} x & ,eğer\ \ x \geq 0 \\ \alpha(e^x - 1), eğer\ x < 0 \end{cases} \tag{2.94}$$

Formülde α, negatif girdiler için doyma noktasını kontrol eden bir hiperparametreyi ve e ise Euler sayısını ifade etmektedir. Yaygın olarak kullanılan α değeri ise 1'dir (Kim, Kim ve Kim, 2020).



Şekil 53. ELU Aktivasyon Fonksiyonu

Şekil 53'te ELU fonksiyonu gösterilmektedir. ReLU'dan farklı olarak ELU, keskin bir sıfır kesme noktası yerine küçük negatif değerlere izin verir. Sıfır olmayan ortalama çıktıları nedeniyle önyargı etkisini azaltarak öğrenmeyi hızlandırma eğilimindedir. Genellikle ReLU ve türevlerinden daha iyi sınıflandırma doğruluğu ve daha hızlı yakınsama sağlar. İçerdiği üstel fonksiyon nedeniyle hesaplama açısından ReLU'dan daha yoğundur. Özellikle büyük negatif girdiler varsa, gradyanların kaybolması sorununa yol açar. Ayarlanması gereken ek bir hiperparametre (alfa) gerektirdiği için, model seçim sürecine bir karmaşıklık eklenmiş olur. Pozitif değerler için normal bir ReLU gibi davranır ve giriş değerini korurken; negatif girişler için, üstel fonksiyon kullanılarak elde edilen çıktılar sıfıra değil, −α değerine doğru yaklaşır (Morozov, Levkivskyi ve Plechystyy, 2024).

2.8.2.1.8. GELU Aktivasyon Fonksiyonu

Hendrycks ve Gimpel (2016) tarafından geliştirilen yüksek performanslı bir sinir ağı aktivasyon fonksiyonudur. Gauss hatası doğrusal birimi olarak anılır. GELU doğrusalsızlığı; nöron değerlerine saygı duyarak, bırakma ve bölge dışı sezgilerini birleştirerek, kimlik veya

sıfır haritasını rastgele uygulayan stokastik bir sürecin dönüşümüdür. Bazı deneylerde, hem ReLU hem de ELU aktivasyon fonksiyonlarından daha iyi performans göstermiştir (Lee, 2023). Pürüzsüz ve olasılıksal davranışı nedeniyle BERT ve GPT gibi transformatör tabanlı mimarilerde yaygın olarak kullanılmaktadır. Formülü şu şekildedir:

$$f(x) = x\Phi(x) \tag{2.95}$$

Formülde $\Phi(x)$, standart Gauss kümülatif dağılım fonksiyonunu ifade etmektedir.

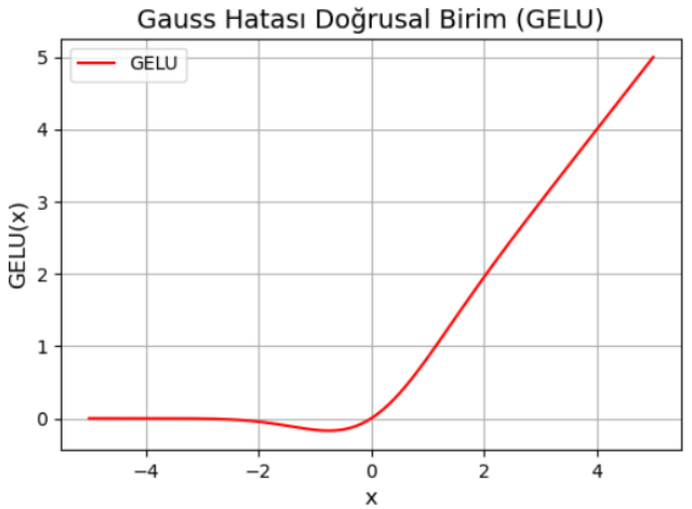
Standart Gauss kümülatif dağılım fonksiyonunun kapalı formda bir çözümü olmadığından, genellikle hata fonksiyonu (erf) kullanılarak ifade edilir:

$$\Phi(x) = \frac{1}{2}\left(1 + erf(\frac{x}{\sqrt{2}})\right) \tag{2.96}$$

Hata fonksiyonunun (erf) formülü ise şu şekilde gösterilir:

$$erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt \tag{2.97}$$

Formülde $\int_0^x e^{-t^2}dt$ integrali, Gauss eğrisinin altındaki alanı 0'dan x'e kadar hesaplar. e^{-t^2} fonksiyonu ise normalizasyon olmadan çan biçimindeki gauss fonksiyonudur. Gauss fonksiyonunun kapalı form integrali olmadığından, erf(x) sayısal olarak hesaplanmalı veya yaklaşık olarak hesaplanmalıdır. $\frac{2}{\sqrt{\pi}}$ faktörü ise, x -∞'dan +∞'a giderken hata fonksiyonu -1 ile +1 arasında değişecek şekilde integrali ölçeklendirir. Normalleştirme faktörü $\sqrt{\pi}$ ise Gauss integralinden gelmektedir (Hendrycks ve Gimpel, 2016).



Şekil 54. GELU Aktivasyon Fonksiyonu

Şekil 54'te örnek bir GELU aktivasyon fonksiyonu grafiği gösterilmektedir. Nöronların yok olması sorunu yaşanmadan gradyanların kaybolması sorununu ele alması ve giriş değerlerinin tüm aralığında pürüzsüzlüğün olması ile monoton olmayan karmaşık desenleri yakalayabilmesi gibi avantajlara sahiptir. Ancak; artan hesaplama karmaşıklığı ve bu duruma bağlı olarak azalan yorumlanabilirlik ve yüksek performansın göreve özgü olabilmesi gibi dezavantajları da bulunmaktadır (Lee, 2023).

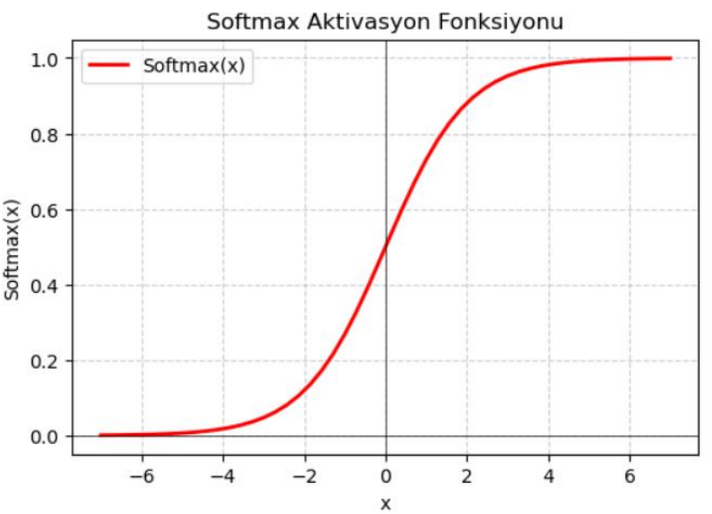
2.8.2.1.9. Softmax Aktivasyon Fonksiyonu

Softmax fonksiyonu, çok sınıflı sınıflandırma problemlerinin çıktı katmanında yaygın olarak kullanılan bir aktivasyon fonksiyonudur. Üstelliğin normalleştirilmiş fonksiyonu olarak da bilinmektedir (Bishop, 2006 s. 115; Sundaram ve Sivanandam, 2018). Ham puanları (logitleri) olasılıklara dönüştürerek çıktıların toplamının 1 olmasını sağlar. Özellikle, geniş sözlüklerden bir kelime veya n-gram tahmin eden dil modelleri için bir yapı taşı olarak doğal dil işlemede popülerlik kazanmıştır (Jozefowicz vd., 2024).

$x = (x_1, x_2, \dots, x_n)$ giriş vektörü olmak üzere, softmax fonksiyonu şu şekilde tanımlanır:

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{2.98}$$

Formülde x_i, i sınıfı için ham puanı (logit); e^{x_i}, ham puana uygulanan üstel fonksiyonu, $\sum_{j=1}^{n} e^{x_j}$, tüm sınıflar için üstel değerlerin toplamını (normalizasyon faktörü); n ise toplam sınıf sayısını ifade etmektedir. Bu fonksiyon, her çıktı değerinin 0 ile 1 arasında olmasını ve tüm çıktıların toplamının tam olarak 1 olmasını sağlayarak, olasılıksal yorumlanabilirliği gerçekleştirir. Üstel fonksiyon her ham puana uygulanır. Tüm değerleri pozitif yapar ve daha büyük değerleri yükseltirken daha küçük değerleri azaltır (Franke ve Degen, 2023).



Şekil 55. Softmax Aktivasyon Fonksiyonu

Şekil 55'te örnek bir softmax aktivasyon fonksiyonu grafiği gösterilmektedir. Fonksiyonun temel avantajı, en büyük değerleri vurgulaması ve maksimum değerin önemli ölçüde altında olan değerleri bastırmasıdır. Bu özellik; çok sınıflı sınıflandırma problemleri için son derece faydalıdır. Her sınıfa belirli olasılıkları atadığı için, bir giriş örneğinin en olası sınıfına karar vermeye yardımcı olur. Softmax fonksiyonu türevlenebilirdir, yani giriş değerlerinin gradyanını hesaplayabilir. Bu nedenle, derin öğrenme modellerini eğitmek için önemli olan gradyan tabanlı optimizasyon yöntemleriyle birlikte kullanılabilir (Shatravin, Shashev ve Shidlovskiy, 2023). Fonksiyonun birkaç dezavantajı da mevcuttur. Aykırı değerlere ve gürültülü verilere karşı hassastır. Düşük olasılıklar, geri yayılım sırasında çok küçük eğimlere neden olarak öğrenmeyi yavaşlatabilir ve yanlış sınıflara yüksek olasılıklar atayabilir. Üs alma ve normalizasyon gerektirdiği için de hesaplama açısından maliyetlidir. Bir örneğin birden fazla sınıfa ait olabileceği çok etiketli görevler için uygun değildir (Lu ve Wu, 2022).

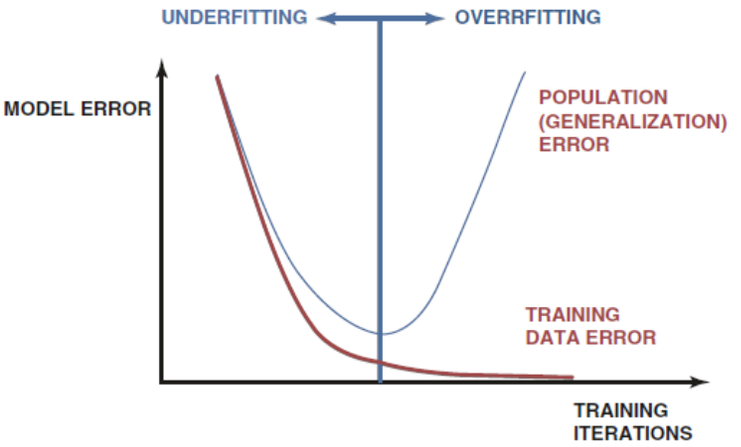
2.8.2.2. Derin Öğrenmede Düzenlileştirme Teknikleri

Düzenlileştirme kavramı; özellikle aşırı veya eksik öğrenmeyi önlemek ve bir sinir ağı modelinin yeni ve daha önceden görülmemiş verilere yönelik iyi bir şekilde genelleştirilmesini sağlamak için uygulanan tekniklerin bütününü ifade etmektedir. Bu teknikler, modelin karmaşıklığına kısıtlamalar getirmeyi ve verilerdeki en temel örüntülere odaklanılmasını amaçlamaktadır (Jung, 2022). Temelinde, eğitim sırasında kayıp fonksiyonuna bir ceza terimi eklemek vardır. Bu ceza, modelin çok karmaşık hale gelmesini veya daha büyük parametre değerlerine sahip olmasını engeller ve böylece modelin eğitim verilerindeki gürültüye uyum sağlama ve daha iyi doğru tahmin etme yeteneği kontrol altına alınmış olur (Khanday ve Sofi, 2021). Aşırı veya eksik öğrenme; dengesiz veri seti, test/doğrulama/eğitim kümesi seçimi, doğruluk metriklerinin seçimi ve çalışma süresi[147] gibi parametrelerin eksik veya fazla seçildiğinde meydana gelir (Nusrat ve Jang, 2018).

Aşırı öğrenme, bir modelin yalnızca temel verideki örüntüyü değil, aynı zamanda eğitim verilerinde bulunan gürültüyü de öğrenmesi ve böylece daha önceden görmediği test verilerinde zayıf performansa yol açması durumu olarak tanımlanabilir. Aşırı öğrenmede eğitim hatası düşük ancak test hatası yüksektir. Eksik öğrenme ise, eğitim verilerini modelleyemeyen veya yeni verilere genelleştiremeyen bir modeli ifade eder (Sehra, Flores ve Montanez, 2021). Eksik öğrenen bir makine öğrenmesi modeli uygun bir model değildir ve eğitim verilerinde zayıf performans göstereceğinden önceden belli olur. Eksik öğrenmede

[147] Runtime

yüksek eğitim ve yüksek test hatası vardır. İyi bir performans metriği verildiğinde tespit edilmesi kolay olduğu için genellikle tartışılmaz. Çözüm ise alternatif makine öğremesi algoritmalarını denemektir. Aşırı öğrenme ve eksik öğrenme, genellikle önyargı-varyans[148] dengesinin belirtileri olarak açıklanır (Geman, Bienenstock ve Doursat, 1992; Kohavi ve Wolpert, 1996).



Şekil 56. Aşırı Öğrenme ve Eksik Öğrenme
Kaynak: (Aliferis ve Simon, 2024, s. 481)

Şekil 56'da aşırı ve eksik öğrenme grafiği gösterilmektedir. Ağırlık güncelleme tekrarlarının sayısı arttıkça, model eğitim verilerini oldukça iyi öğrenir ve genelleme performansı da artar, böylece model hatası azalır. Gereğinden daha fazla eğitimin, eğitim verilerindeki doğruluğu artırdığı ancak genelleme performansını azalttığı bir "kırılma noktası" vardır. Şekilde, bu kırılma noktasının sol tarafındaki model eksik öğrenmeye uğramış ve sağ tarafındaki model ise, örüntüleri aşırı öğrenmiştir.

Veri artırma, sentetik veri üretme, gürültü ekleme, bırakma[149], L1 ve L2 düzenlileştirmeleri, seyrek-yoğun-seyrek eğitim, erken durdurma ve transfer öğrenmesi gibi düzenlileştirme teknikleri bulunmaktadır.

2.8.2.2.1. Veri Artırma

Veri artırma, gözlemlenmemiş verilerin veya gizli değişkenlerin tanıtılması yoluyla yinelemeli optimizasyon veya örnekleme algoritmaları oluşturma yöntemlerini ifade eder (van Dyk ve Meng, 2001). Bu teknikte, daha geniş bir görüş aralığı sağlamak için var olan verilerin görünümüne çeşitli dönüşümler uygulanarak veya türü değiştirilerek veri setinin boyutu yapay olarak genişletilir. Böylece model, aynı verinin farklı varyasyonlarını da inceleyerek daha iyi

[148] Bias-variance
[149] Dropout

bir genelleştirmeyi sağlamış olur (Ding, vd., 2024). Başka bir deyişle, veri artırma ile mevcut verilerin farklılaştırılmış bir simülasyonu ve orijinal verinin değiştirilmiş sürümleri modele verilir. Dönüşüm türleri ise kullanılan veri türüne bağlı olarak değişiklik göstermektedir. Örneğin; görüntü verileri için çeşitli geometrik dönüşümler (döndürme, dikey veya yatay çevirme, ölçeklendirme, kırpma vb.) ve renk dönüşümleri (parlaklık derecesi ayarlama, kontrast ayarlama, ton ve doygunluk ayarlama vb.) uygulanabilir. Metin verileri için eş anlamlı kelime değiştirme, geri çeviri, belirli kelimeleri rastgele silme uygulanabilirken ses verisi için sesi yavaşlatma veya hızlandırma ve arka plan gürültüsü ekleme gibi teknikler uygulanabilir. Bu faydalarının yanında bilgi kaybı riski ve hesaplama yükünün artması gibi dezavantajları da bulunmaktadır (Kumar vd., 2023).

2.8.2.2.2. Sentetik Veri Üretme

Sentetik veri, gerçek verilerin özelliklerine dayalı üretici bir işlemden elde edilmektedir (Assefa vd., 2020). Sınıflandırıcıları düzenlemek, veri kümelerini anonimleştirmek, dengesiz veri kümelerindeki frekansları dengelemek ve azınlık sınıfların öğrenmesini güçlendirmek vb. amacıyla kullanılmaktadır (Fonseca, Douzas ve Bacao, 2021). Çekişmeli üretici ağ[150], sentetik veri üretiminde etkili yollardan biri olup, orijinal veri setinin örüntüsünde yatan dağılımı öğrenerek gerçeğe yakın veriler üretebilir (Ramzan, vd., 2024). Bir diğer yöntem ise varyasyonel otokodlayıcılardır[151]. Özellikle veri üzerindeki gürültünün giderilmesi ve yeniden yapılandırılmasında faydalıdır (Mostofi, Tokdemir ve Toğan, 2024).

2.8.2.2.3. Gürültü Ekleme

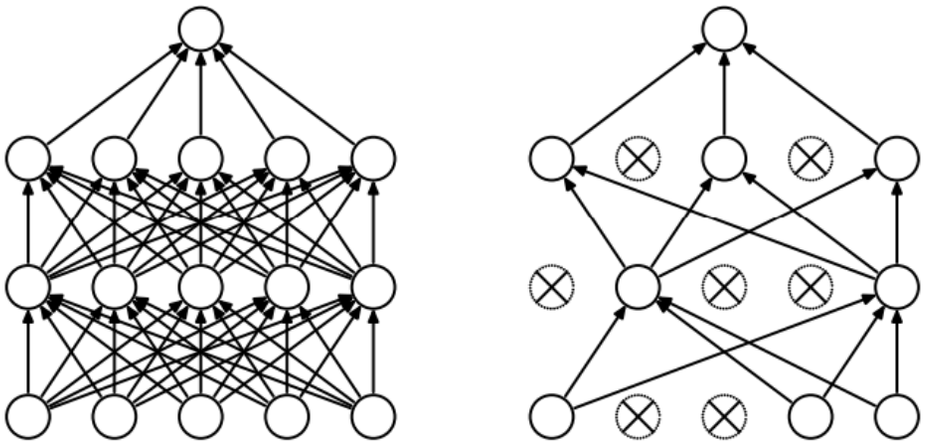
Gürültü; bir görüntüde, bir seste veya bir metinde üretilen ve veri üzerindeki örüntüyü tanımada modeli tetikleyici özelliği olan bir kavramdır (Ganie ve Dadvadipour, 2024). Uygulama türüne bağlı olarak verinin kalitesini farklı şekillerde etkiler. Güdüleyici gürültü; görüntü sinyalindeki ani değişiklikler nedeniyle oluşur ve görüntüye siyah ve beyaz noktalar eklenir. Beyaz renk, görüntü pikselinin renk değerinin en yüksek değere; siyah renk ise en düşük değere indirilmesiyle oluşmaktadır (Boncelet ve Bovik, 2005). Gauss gürültüsü ise doğada bulunabilen bir gürültü türü olup Gauss dağılımına göre karakterize edilmektedir. Görüntüdeki her bir piksel değeri ile Gauss dağılımına sahip gürültünün rastgele değerinin toplamı ile oluşur (Patidar, vd. 2010). Poisson gürültüsü ise Poisson dağılımı tabanlı ve

[150] Generative Adversarial Network - GAN
[151] Variational Autoencoder

sensörün algıladığı foton sayısından istatistiksel bilgi sağlamak için yeterli olmadığında kullanılan gürültü türüdür (Sriwong, Kerdprasop, Kerdprasop, 2021).

2.8.2.2.4. Bırakma[152]

Tekniğin ardındaki ana fikir; modelin eğitimi sırasında nöronları, ilgili tüm bağlantılarıyla sinir ağından rastgele çıkarmaktır. Böylece aşırı öğrenmenin önüne geçilmiş olur. İşlem, eğitim sırasında, üstel sayıda farklı "inceltilmiş" ağdan örnekler alır. Örneğin; her bir nöron diğer nöronlardan bağımsız sabit bir olasılıkla (p) tutulur; p, bir doğrulama kümesi kullanılarak seçilebilir veya 0,5 olarak ayarlanabilir (Srivastava vd., 2014).



Şekil 57. Bırakma
Kaynak: (Srivastava vd., 2014, s. 1930)

Şekil 57'de bir sinir ağının nöronlarının rastgele çıkarıldığı görsel bulunmaktadır. Görselin sol tarafındaki model, orijinal gizli katmanlara sahip standart bir sinir ağını temsil etmektedir. Sağ taraftaki görselde ise sinir ağındaki nöronlara rastgele bırakma uygulandıktan sonraki inceltilmiş hali görülmektedir.

2.8.2.2.5. L1 ve L2 Ağırlık Azalması

Ağırlık azalması olarak da bilinen L1[153] ve L2[154] düzenlileştirmesi, kayıp fonksiyonuna bir ceza terimi eklenmesiyle oluşur ve sinir ağındaki büyük ağırlık değerleri oluşmasının önüne geçer. L1 düzenlileştirmesi, Tibshirani (1996) tarafından geliştirilmiştir. Hem sürekli küçülmeyi hem de otomatik değişken seçimini aynı anda yapan fonksiyon, artık kareler toplamının en aza indirilmesiyle elde edilmektedir. L1, katsayıların mutlak değerinin toplamını kısıtlar:

[152] Dropout
[153] Lasso regresyon
[154] Ridge regresyon

$$L1 = maliyet\ fonksiyonu + \lambda \sum_i |w_i| \tag{2.99}$$

Formülde λ (lambda), cezanın gücünü kontrol eden düzenleme parametresini; w, modelin ağırlıklarını ve $\sum_i |w_i|$ ise ağırlıkların mutlak değerlerinin toplamını ifade etmektedir. Bu yöntem, bazı ağırlıkları sıfıra düşürerek seyrekliği sağlar. L1 regresyonu, öznitelik seçiminde sıklıkla kullanılır çünkü alakasız veya gereksiz öznitelikleri etkili bir şekilde kaldırmaktadır.

L2 düzenlileştirmesi ise kayıp fonksiyonuna katsayıların büyüklüğünün karesine eşit bir ceza ekler. Büyük katsayıların modele girmesini engelleyen bu işlem sonucunda model karmaşıklığı azaltılmış olur (Hoerl ve Kennard, 2000). Formülü şu şekilde gösterilir:

$$L2 = maliyet\ fonksiyonu + \lambda \sum_i w_i^2 \tag{2.100}$$

L2 düzenlileştirmesi, daha küçük ağırlık değerlerini teşvik eder. Bunun yanında öğrenmeyi baskın özniteliklere aşırı derecede bağımlı kılmak yerine birden fazla özniteliğe dağıtır.

L1 ve L2 düzenlileştirmesinin aynı anda kullanıldığı yöntem ise elastik ağ düzenlileştirmesi olarak adlandırılır (Zou ve Hastie, 2005). Formülü şu şekilde gösterilir:

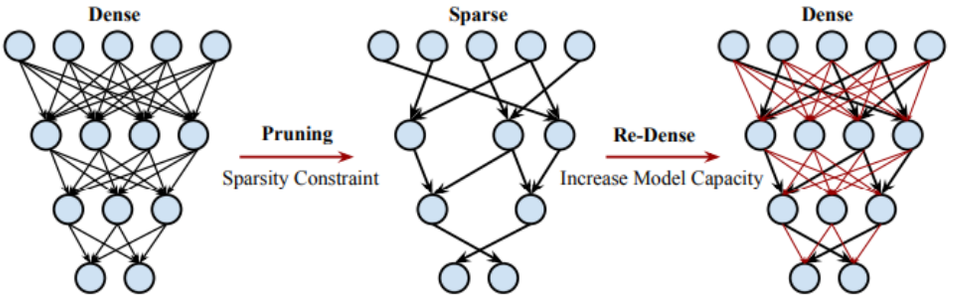
$$EAD = maliyet\ fonksiyonu + \lambda_1 \sum |w_i| + \lambda_2 \sum w_i^2 \tag{2.101}$$

2.8.2.2.6. Seyrek-Yoğun-Seyrek Eğitim

Eksik öğrenme ve yüksek bir önyargı değerine yol açan çeşitli uygulamalar, modelin düşük başarım elde etmesine neden olur. Ayrıca varyans ve önyargıyı aynı anda optimize etmek zordur. Bu sorunu çözmek için, yoğun bir modelden başlayan, ardından modeli seyrek optimizasyonla düzenleyen ve son olarak budanmış ağırlıkları geri yükleyip yeniden eğiterek model kapasitesini artıran bir teknik olarak Han vd. (2016) tarafından geliştirilmiştir. Teknik, modelin hem genellemesini hem de optimizasyon verimliliğini iyileştirir ve son model, orijinal ilk model ile aynı mimariye ve boyuta sahip olduğundan hesaplama yükü oluşturmamaktadır. Teknik, üç aşamada dönüşümlü olarak gerçekleşmektedir:

1. ***Yoğun Aşama:*** Standart bir ağ, tüm parametreler tamamen başlatılmış olarak eğitilir. Bu aşama, modelin zengin bir öznitelikle başlamasını sağlar.
2. ***Seyrek Aşama:*** Yoğun aşamadan sonra, en az büyüklüğe sahip ağırlıklar budanır (sıfıra ayarlanır) ve seyrek bir ağ oluşturulur. Kalan ağırlıklar korunarak eğitime devam edilir ve bu seyreklik, düzenlileştirmeyi zorlayarak aşırı öğrenmeyi azaltmaya yardımcı olur. Ağ, önemli bağlantılara odaklanarak daha sağlıklı durumları öğrenir.

3. ***Yoğun Aşama:*** Budanan ağırlıklar orijinal durumlarına geri yüklenir ve tekrar güncellenmelerine izin verilir (Han vd. (2016).

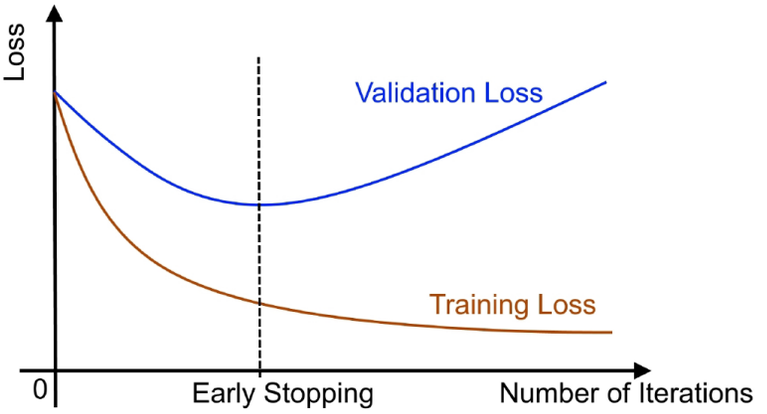


Şekil 58. Yoğun-Seyrek-Yoğun Eğitim
Kaynak: (Srivastava vd., 2014, s. 1930)

Şekil 58'de yoğun-seyrek-yoğun eğitim akışı gösterilmektedir. İlk aşamada orijinal sinir ağı yoğun olarak öğrenmeye başlar. Burada amaç; hangi ağırlıkların daha önemli olduğunu tespit etmektir. Bu ağırlıklar belirlendikten sonra önemsiz görülen ağırlıklar ve nöronlar bırakılır. İkinci aşamada seyrek eğitim, modeli düzenlileştirir ve üçüncü aşamadaki yoğun eğitim katmanı, kırmızı ile boyalı budanmış ağırlıkları geri yükleyerek modeli aşırı öğrenmeden uzaklaştırır ve model kapasitesini artırır.

2.8.2.2.7. Erken Durdurma

Erken durdurma, modelin doğrulama kümesindeki performansı düşmeye başladığında eğitim sürecini durdurarak aşırı öğrenmeyi önlemek için kullanılan bir düzenlileştirme tekniğidir (Hagiwara, 2002). Model, eğitim sırasında her bir epochtan sonra hem eğitim kümesinde hem de ayrı bir doğrulama kümesinde değerlendirilir. Başlangıçta hem eğitim hem de doğrulama kayıpları daha az gerçekleşir. Ancak, belirli bir noktadan sonra, doğrulama kaybı artmaya başlarken eğitim kaybı azalmaya devam edebilir ve bu da aşırı öğrenmeyi gösterir. Başka bir deyişle, aradaki makasın açılmaya başladığı an, aşırı öğrenmenin başladığı andır. Erken durdurma tekniğinde, doğrulama kaybı önceden tanımlanmış bir epoch sayısı için arttığında, eğitim durdurulur ve en düşük doğrulama kaybına sahip epochtaki model, tüm ağırlıklarıyla birlikte geri yüklenir.



Şekil 59. Erken Durdurma Grafiği
Kaynak: (Naushad, Kaur ve Ghaderpour, 2021, s. 6)

Şekil 59'da erken durdurma grafiği gösterilmektedir. Grafikte x ekseni epochları, y ekseni ise tahminlerdeki hata (kayıp) oranını göstermektedir. Daha düşük y değeri, daha iyi model performansını göstermektedir. Kahverengi eğri, eğitim kaybını temsil etmektedir. Eğitim kaybı, ilk epochlarda yüksek başlar ve model öğrendikçe sürekli azalır. Erken durdurma noktasından sonra da azalmaya devam edebilir ve bu da modelin eğitim verilerine daha iyi uyum sağladığınının bir göstergesidir. Mavi eğri ise doğrulama kaybını göstermektedir. Doğrulama kaybı ise başlangıçta azalır ve bu da modelin iyi bir genelleme yaptığını gösterirken, belirli bir noktada minimum değerine ulaşır. Bu noktadan sonra artmaya başlar ve bu da aşırı öğrenmenin bir işaretidir. Model, eğitim verilerine özgü kalıpları öğreniyor ancak daha önce hiç karşılaşmadığı veriler için bir genelleme yapamıyorsa aşırı öğrenme gerçekleşmiş olur. Erken durdurma noktası ise eğitimi durdurmak için en iyi noktayı gösterir. Bu noktadan sonra ise, model artan doğrulama kaybıyla birlikte aşırı öğrenir.

Bu yöntem oldukça popüler olmasına ve "patience" parametresinin farklı problemler için uygulanabilir olmasına rağmen, sadece doğrulama kaybını[155] izler ve eğitim kaybındaki değişiklik göz ardı edilir. Bu eksikliklerin üstesinden gelmek için, probleme daha iyi ayarlanabilen korelasyona dayalı durdurma kriteri[156]; Miseta, Fodor ve Vathy-Fogarassy (2024) tarafından geliştirilmiştir. Bu kriter, eğitim ve doğrulama veri kümelerindeki model performansını aynı anda izlemektedir.

[155] Validation loss
[156] Correlation-Driven Stopping Criterion

2.8.2.3. Bias-Varyans İlişkisi

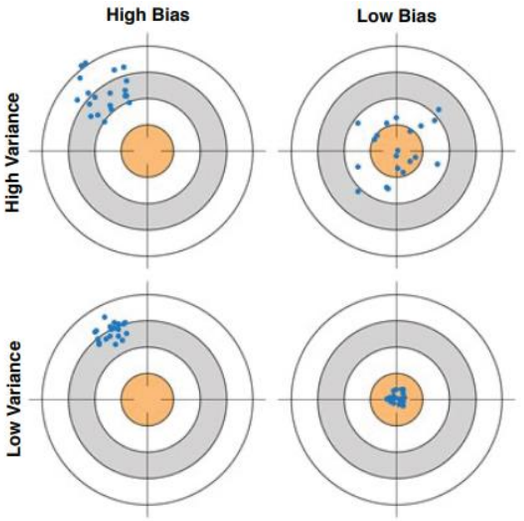
Önyargı, modelleme sonucunda tahmin edilen verilerin değerlerinin, gerçek değerinden sapmayı; varyans ise belirli bir veri noktası için model tahmininin değişkenliğini veya verilerin nasıl yayıldığını ölçmektedir (Goodfellow, Bengio ve Courville, 2016/2018). Önyargı ve varyansın dengede olması, istenen bir durumdur ancak çoğu zaman modellerin yapısı, bu iki değerin dengede olmasını başaramaz. Başka bir tanımda önyargı, modelin gerçek değerden sistematik sapmasını; varyans ise modelin eğitim verisindeki küçük değişikliklere gösterdiği hassasiyeti ifade eder. Bu iki unsur genellikle birbirine ters orantılıdır ve denge içinde olmaları, modelin genelleme başarısı açısından önemlidir (Ranglani, 2024). Önyargı, basitleştirilmiş bir model kullanılarak oldukça karmaşık bir problemi modellemeye çalışmaktan kaynaklanan hata olarak da değerlendirilebilir. Yüksek önyargıya sahip model güçlü varsayımlarda bulunur ancak, çoğunlukla eksik öğrenmeyle sonuçlanır. Örneğin; doğrusal olmayan bir ilişkiyi tahmin eden doğrusal bir model, yüksek bir önyargıya sahip olacak ve tahminde sistematik hatalar yapacaktır. Varyans ise, bir modelin verilerdeki küçük değişikliklere ne kadar duyarlı olduğunu gösterir. Bir model yüksek çeşitliliğe sahipse, çok karmaşıktır ve eğitim verilerine çok yakındır. Örneğin derin bir sinir ağı, eğitim verilerindeki gürültüyü ezberleyecek ve bu da yüksek varyansa ve yeni verilerde zayıf genellemeye yol açacaktır.

Model karmaşıklığı arttıkça, önyargı hatası azalacaktır çünkü model verilerdeki gerçek eğilimleri yakalayacak kadar esnek olacaktır. Öte yandan, varyans da artacaktır çünkü modelin artan esnekliği aynı zamanda gürültüyü de içine alacak şekilde aşırı öğrenmeye meyilli olduğu anlamına gelir. Bu nedenle, öyle bir model karmaşıklığı seçilmelidir ki, mümkün olan önyargı-varyans dengesi optimum noktada olsun. Gürültü hatası ise verilerde her zaman mevcuttur ve bu nedenle indirgenemez hata olarak da adlandırılır (Deprez ve Robinson, 2024).

Yüksek varyansa sahip modeller genellikle aşırı öğrenme eğilimindedir. Böyle bir durum, veri seti için çok karmaşık bir model oluşturmaktan kaynaklanır. Yüksek önyargı genellikle daha karmaşık bir modelle azaltılabilir. Ancak bu, varyansı artırabileceği için dikkatli şekilde dengelenmelidir. Eksik öğrenme, genellikle yüksek önyargıya sahip basit modellerle ilişkilidir. Bu modeller eğitim verisine duyarsızdır ve karmaşık örüntüleri öğrenemez. Bu nedenle veri kümesinin boyutunu artırmak, değişime yanıt veremeyeceği için modeli önemli ölçüde iyileştirmeyecektir. Model varyansı azaltıldığında, tahminlerin kararlılığı artar ancak önyargı da artabileceği için genel hata yine yükselebilir (Deng vd., 2015).

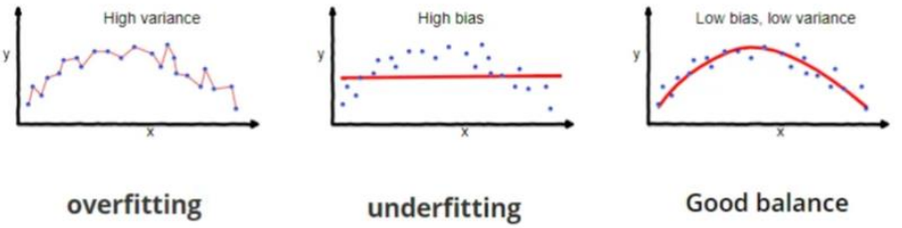
Optimizasyon yapılırken dikkat edilmesi gereken noktalardan biri de öğrenmenin eğitim ve test hatasının birlikte minimum olduğu noktada gerçekleştiğinin bilinmesidir (Chen vd., 2022).

Çünkü eğitim hatası zamanla azalmaya devam etse de, belirli bir noktadan sonra test hatası artmaya başlar. Bu durum, modelin eğitim verisine aşırı uyum sağladığını ve genelleme yeteneğini kaybettiğini gösterir, yani aşırı öğrenme gerçekleşmiştir. Aşırı öğrenme, arzu edilen bir durum değildir. Başka bir deyişle, model aşırı öğrendiğinde eğitim hatası düşer fakat test hatası bir yerden sonra sürekli yükselir.



Şekil 60. Bias-Varyans İlişkisi 1
Kaynak: (Brady ve Brockmeier, 2021, s. 212)

Şekil 60'ta önyargı-varyans ilişkisini açıklayan bir diyagram görülmektedir. Burada hedefin merkezi, bir test noktasındaki (x,y) gerçek değerin en iyi tahminini; hedefin merkezinden uzaktaki bölgeler ise hatalı tahminleri temsil etmektedir. Hedef üzerinde bulunan her bir nokta, modelin yansımasıdır. Noktaların sıkı bir şekilde kümelenme derecesi, modelin varyansını göstermektedir. Nokta kümesinin merkezinin hedefin merkezine yaklaşma derecesi ise modelin önyargısını gösterir. Şeklin sol üst tarafında bulunan hedef tahtası, hem yüksek yanlılığa hem de yüksek varyansa sahip noktaların bir dağılımını göstermektedir. Bu noktalar, gerçek değerden uzakta konumlanmış ve geniş bir şekilde dağılmıştır. Sol alttaki hedef tahtası da benzer olarak, yüksek önyargıya sahip ancak düşük varyansı olan bir modeli işaret etmektedir. Sağdaki hedef tahtaları, noktaların gerçek değerini doğru bir şekilde tahmin eden ve bir merkez etrafında kümelenen modelin düşük önyargıya sahip olduğunu göstermektedir. Sağ alttaki hedef tahtasında modelin, tahminleri yüksek güvenilirlikle ve doğru bir şekilde tahmin ettiği görülmektedir. Eksik öğrenmeye uğrayan modeller genellikle yüksek önyargıya ve düşük varyansa sahipken, aşırı öğrenmiş modeller ise yüksek varyansa ve düşük önyargıya sahiptir.



Şekil 61. Bias-Varyans İlişkisi 2
Kaynak: (Derras ve Makhoul, 2022, s. 9)

Şekil 61'de ise önyargı-varyans durumlarının aşırı öğrenme, eksik öğrenme ve iyi denge durumları gösterilmektedir. İyi bir dengenin sağlanabilmesi için düzenlileştirme tekniklerinden L1 ve L2, k-katlı çapraz doğrulama ve topluluk yöntemlerinden bagging ve boosting ile hiperparametre ayarlama tekniklerinden ızgara araması veya rastgele arama kullanılabilir.

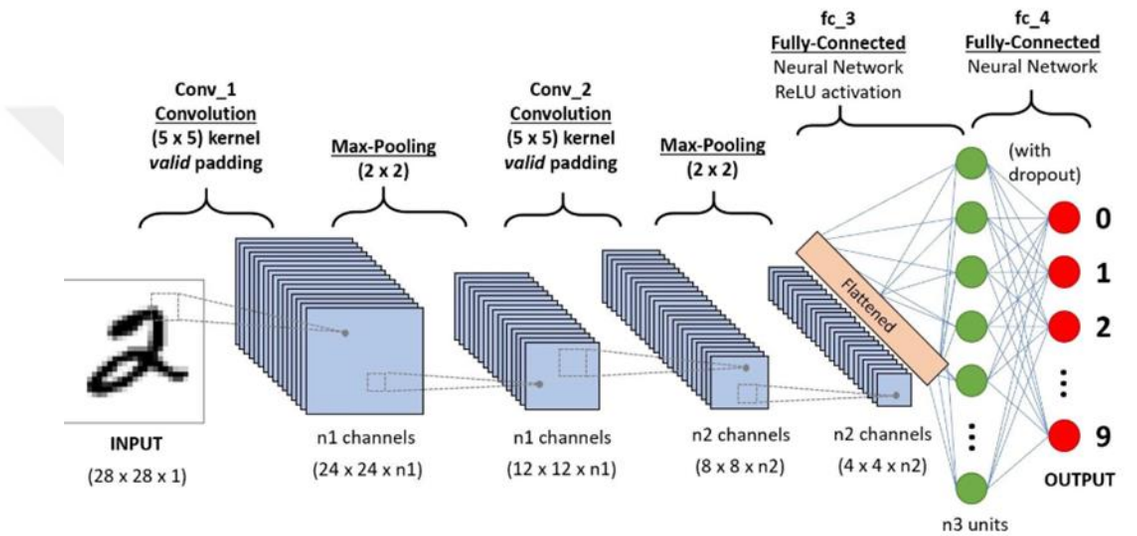
2.8.2.4. Hiperparameteler

Makine öğrenmesi ve derin öğrenme algoritmaları kendi içlerinde önceden belirlenmiş değerlerle model eğitimlerinde kullanılmaktadır. Bu değerler, modelin çeşitli özelliklerine göre değiştirilebilir olup, hiperparametre olarak bilinmektedir. Hiperparametrelerin ne ifade ettiğine geçmeden önce, birbirleriyle sıklıkla karıştırılan parametre kavramı ile arasındaki farkların açıklanması, konunun daha iyi anlaşılmasını sağlayacaktır. Bir parametre, modelin eğitimi sırasında model tarafından öğrenilen değerdir (Dey ve Williams, 2024). Parametreler, çeşitli teknikler (optimizasyon) aracılığıyla eğitim süresi boyunca otomatik olarak güncellenir ve modelin başarımını doğrudan etkiler. Başka bir deyişle parametre, modelden öğrenilebilen değerdir. Parametrelere örnek olarak; ağırlıklar (w) ve bias (b) verilebilir. Hiperparametreler ise eğitim başlamadan önce elle veya otomatik olarak ayarlanabilen harici yapılandırmalardır (Arnold vd., 2024). Hiperparemetreler, modelin nasıl öğrendiğini kontrol etmekle görevli olup, eğitim sırasında güncellenememektedir. Başka bir deyişle, model eğitimi için önceden belirlendiğinden dolayı, eğitim sırasında sabittir. Bu hali ile hiperparametreler, modelden öğrenemez. Hiperparametrelere örnek olarak; öğrenme oranı, yığın boyutu[157], epoch, katman, nöron sayısı ve bırakma oranı vb. verilebilir. Hiperparametrelerin doğru bir şekilde ayarlanması, modelin yüksek bir başarım göstermesi için hayati öneme sahiptir. Eğer bu değerler düzgün bir şekilde ayaralanamazsa modelin başarımı veya tahmin gücü önemli ölçüde etkilenir.

[157] Batch size

2.8.2.5. Evrişimli Sinir Ağları

Evrişimli sinir ağları[158] çok katmanlı algılayıcıların mimarisi temelinde geliştirilen ve özellikle görüntüleri analiz etmek için kullanılan bir sinir ağları türüdür. Görüntü sınıflandırma, nesne algılama ve segmentasyon gibi görevlerde oldukça etkilidir. Ağ, girdi verilerinden özniteliklerin uzamsal hiyerarşilerini otomatik ve uyarlanabilir bir şekilde öğrenmek için tasarlanmıştır. Klasik bir evrişimli sinir ağı; giriş katmanı, evrişim katmanı, aktivasyon katmanı, havuzlama katmanı, tam bağlantılı katman ve çıkış katmanından oluşmaktadır. Daha gelişmiş mimarilerinde bu katmanlara ek olarak bırakma[159] ve toplu normalizasyon[160] gibi yardımcı katmanlar da kullanılmaktadır (Garbin, Zhu ve Marques, 2020; Ayeni, 2022).



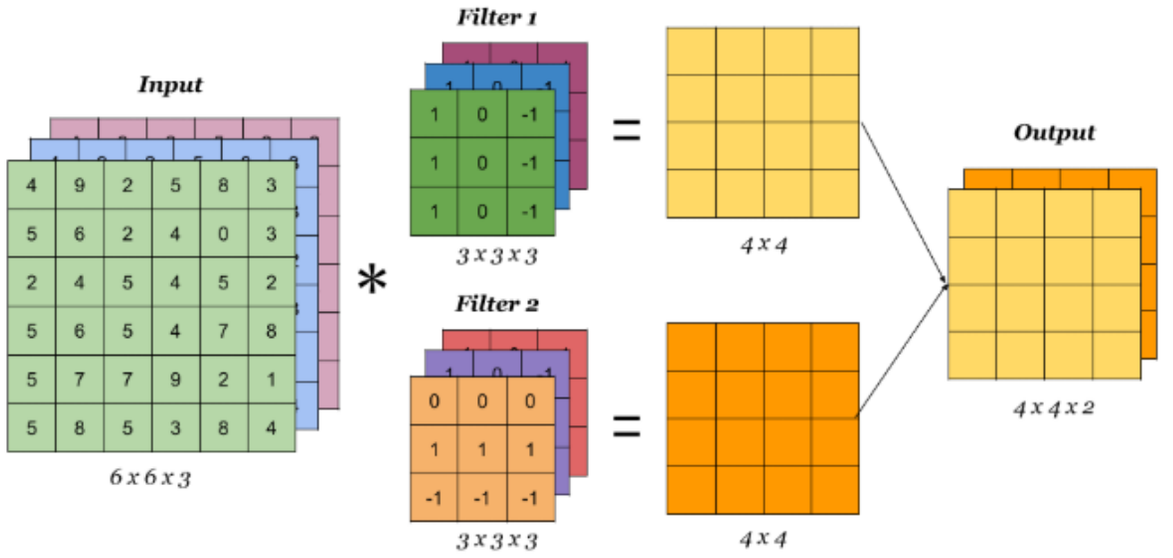
Şekil 62. Evrişimli Sinir Ağları Çalışma Prensibi
Kaynak: (Altan, 2019, s. 323)

Şekil 62'de örnek bir evrişimli sinir ağının mimarisi gösterilmektedir. 28×28×1 piksellik (gri tonlamalı) bir giriş görüntüsü ile işlemler başlamaktadır. Mimarinin örüntüsü; giriş → evrişim → havuzlama → evrişim → havuzlama → tam bağlı katman → tam bağlı katman → çıktı sırasını takip eder. Birinci evrişim katmanında (Conv_1); piksel ekleme (dolgu) ile 5×5 evrişim çekirdeği uygulanır ve 24×24×n_1 boyutunda n_1 özellik haritası üretilir. İlk havuzlama katmanında 2×2'lik maksimum havuzlama yöntemi ile uzamsal boyut 12×12×n_1'e düşürülür. İkinci evrişim katmanında (Conv_2); diğer bir 5×5 evrişim çekirdeği, 8×8×n_2 boyutunda n_2 özellik haritası üretir. Bu aşamada daha yüksek seviyeli özellikler (desenler ve şekiller) örüntüden çıkarılmaktadır. İkinci havuzlama katmanında tekrardan 2×2'lik maksimum havuzlama yöntemi ile havuzlama işlemi gerçekleştirilir ve boyutlar 4×4×n_2'ye düşürülür.

[158] Convolutional Neural Networks - CNN
[159] Dropout
[160] Batch normalization

Düzleştirme[161] adımında ise $4\times4\times n_2$ öznitelik haritaları tam bağlantılı katmanlarda tek bir vektöre dönüştürülür. Ardından birinci tam bağlantılı katmanda (fc_3), ReLU aktivasyonu uygulanır. İkinci tam bağlantılı katmanda (fc_4) ise aşırı öğrenmeyi önlemek için bırakma işlemi uygulanarak bazı nöronlar çıkarılır. Çıktı katmanında ise sınıflandırma etiketlerine karşılık gelen 10 sınıflı (0-9) etiket oluşturulur.

Derin öğrenmede, genellikle çok kanallı görüntüler üzerinde çalışılmaktadır. Çok kanallı bir görüntünün en basit hali RGB'den oluşan görüntü tipidir. RGB'ye örnek olarak günlük hayatta çekilen fotoğraflar verilebilir. Fotoğraf, gri tonlamalı ise burada sadece tek bir kanal bulunur. Fotoğraf renkli ise, bu bir RGB görüntüdür ve kırmızı, yeşil ve mavi olmak üzere üç kanal bulunur. Matris gösterimi ise (H[162], W[163], 3) şeklinde üç boyutlu tensör olarak verilir. Evrişimli ağlarda ara katmanlar birçok kanala sahip öznitelik haritaları üretir [(H, W, C_in), C_in giriş kanallarının sayısını gösterir]. Görüntü 6x6 boyutunda ve 3 kanallı renkli bir görüntü ise gösterim 6x6x3 şeklinde olur. Yani; kanal sayısı boyut sonuna eklenir. Böyle bir görüntü; ardara gelen üç matristen oluşan bir görüntü olarak da tanımlanabilir. Bu görüntüye, bir evrişim işlemi uygulandığında, işlem mutlaka üç kanallı olmak zorundadır. Yani; filtrenin kanal boyutu, eğer giriş görüntüsü üç kanallı ise, üç kanallı olmak zorundadır.



Şekil 63. Çok Katmanlı Evrişim İşlemi
Kaynak: (Cornell Bowers, 2024)

Şekil 63'te çok kanallı bir evrişim işlemi görülmektedir. Şekilde giriş, klasik RGB katmanlarına sahiptir. Bu giriş görüntüsüne evrişim işlemi uygulandığında 3 kanallı filtre, giriş

[161] Flatten
[162] Height
[163] Width

matrisinin ilk 3 boyutunun tam üzerinde eşlenir ve tüm piksel değerleri evrişim işlemine alınıp hesaplanır. Filtre matrisindeki her bir katman, örneğin filtredeki 1. katman R alanını, 2. katman G alanını ve 3. katman da B alanını evrişim işlemine alıp, tüm değerler toplandığında, çıkışta tek bir değer verir. Tüm elemanlar her katman için bir defa olmak üzere toplamda 3 defa aynı işlem yapılmış olur. Aynı işlem, alt katmandaki filtre için de uygulanır. Bu süreçte ne kadar filtre kullanılıyorsa, o kadar öznitelik çıkarılıyor demektir. Başka bir deyişle her bir filtre, bir öznitelik çıkarır. Örneğin; 1. filtre, yatay kenar bulmak için, 2. filtre de dikey kenar bulmak için tasarlanmış olursa, bu 2 filtre ardarda uygulandığında, bir matriste yatay kenarlar, diğer matriste de dikey kenarlar bulunuş olur. Bunlar, öznitelikleri simgeler. İşlemler sonucunda elde edilen değerler, 4x4x1'lik 2 matris boyutunda olur. 4 değeri çıkış matrisi formülünden (n+2p-f+1) gelmiştir. Tüm işlemlerin sonucunda da 4x4x2'lik bir çıkış matrisi elde edilir ve bu çıkış matrisi bir tensördür. Çıkış matrisi bir görüntüyü simgelemeyebilir ancak, bir öznitelik çıkarma işlemi yapılmış olur.

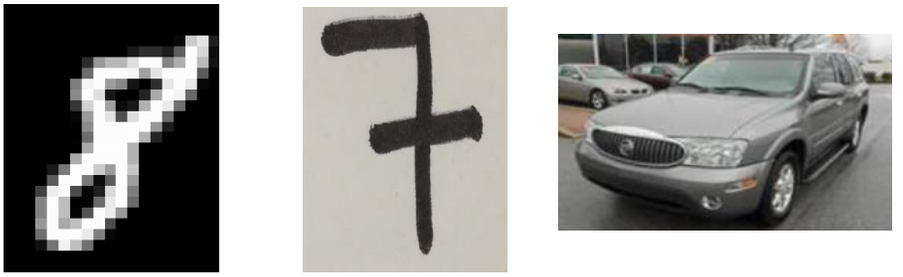
Evrişim işlemine filtre uygulandığında, görüntü siyah-beyaz veya koyu renkli ise; en baştaki basit özniteliklerde dikey, yatay veya rotasyona sahip kenarlar, spesifik renk geçişleri ve daha derin katmanlarda daha nitelikli öznitelikler öğrenilir. Manuel işlemlerde ise filtreler uygulayıcı tarafından belirlenir (Sobel, Prewitt, Robert filtreleri vb.). Derin öğrenmede ve evrişimli sinir ağlarında filtre tasarımı yapılmaz. Model, öznitelikleri katmanlar boyunca kendisi öğrenir. Klasik bir yapay sinir ağı; çıkışı hesaplayabilmek için giriş katmanı, aktivasyon fonksiyonu, bias değeri ve ağırlıklara ihtiyaç duymaktadır. Evrişimli sinir ağında da temel olarak mantık aynıdır. Yapay sinir ağından farklı olarak girişler, vektörü değil, bir görüntü matrisini simgelerr ve bu görüntü matrisi, tek veya çok kanallı olabilir. Bu yüzden işlem yükü biraz daha artar.

2.8.2.5.1. Giriş Katmanı

Bir evrişimli sinir ağının giriş katmanı, genellikle ham görüntü verisini alan ve bu veriyi ağ tarafından detaylı işleme için hazırlayan ilk katmandır. Bu katmanda, herhangi bir evrişim veya aktivasyon işlemi gibi bir hesaplama gerçekleşmemektedir. Veriler sadece ağın işlemesi için yapılandırılır veya biçimlendirilir. Genellikle çok boyutlu bir dizi (tensör) olarak üç temel bileşeni bulunmaktadır:

- Yükseklik (H): Dikey eksen piksel sayısı,
- Genişlik (W): Yatay eksen piksel sayısı,

- Kanal (C): Renk kanallarının sayısı (gri tonlamalı görüntüler için bir kanallı, RGB görüntüler için üç kanallı) (Goodfellow, Bengio ve Courville, 2016/2018).



Şekil 64. Giriş Katmanı İçin Örnek Görüntüler

Şekil 64'te üç farklı giriş katmanı örnek görüntüsü gösterilmektedir. Siyah ve gri temalı görüntüler tek kanallı iken, araba görüntüsü ise üç kanallıdır.

Giriş katmanında ön işleme gibi işlemler yapılabilmektedir. Ağ, giriş verilerinin tutarlı aralıklara sahip olması durumunda daha iyi performans göstermektedir. Bunun için bir görüntüyü ağa aktarmadan önce normalizasyon uygulanabilir. Yaygın normalizasyon teknikleri arasında piksel değerlerini yeniden ölçeklendirme (0-255 → 0-1) ve standardizasyon (ortalamanın standart sapmaya bölünmesi) bulunmaktadır (Triwijoyo, Adil ve Anggrawan, 2021; Hoque vd., 2024). Görüntüler ağa verilmeden önce çeşitli açılarda döndürme, çevirme, yakınlaştırma, kırpma ve ışık şiddetini ayarlama gibi dönüşümler de uygulanarak modelin daha iyi genelleme yapması sağlanabilir.



Şekil 65. Modifiye Edilmiş Giriş Görüntüleri

Şekil 65'te ise yaklaşık 20° sola doğru döndürülmüş ve düşük ışık oranı artırılmış görüntüler bulunmaktadır. Veri artırımı yoluyla modelin, görüntüleri farklı açılardan ve ışık oranlarından daha iyi tanıması beklenmektedir.

2.8.2.5.2. Evrişim Katmanı

Evrişim katmanında temel işlemler gerçekleştirilir. Evrişimli işlemler, bilgisayarlı görüde[164] iki boyutta yapılmaktadır (López, López ve Crossa, 2022). Bu durumun nedeni ise görüntülerin

[164] Computer Vision

ağda matris olarak ifade edilmesidir. Bu matrislerin üzerinde işlem yapılabilmesi için de evrişim işlemini iki boyutta tanımlamak gerekmektedir. Ayrıca, derin öğrenmede ekran kartlarının ön plana çıkmasının nedenlerinden biri de bu durumdur çünkü ekran kartlarının kapasitesi, matris işlemlerini hızlandırmaktadır.

Evrişim katmanında, giriş görüntüsüne filtreler (veya çekirdekler) uygulanarak bir evrişim işlemi gerçekleştirilir. Filtre; görüntü üzerinde kayan, piksel bazında çarpma işlemi gerçekleştiren ve sonuçları toplayan küçük bir matris özelliğindedir. İşlemler sonucunda görüntüdeki kenarlar, dokular veya basit şekiller gibi belirli örüntüleri ortaya çıkaran bir öznitelik haritası[165] üretilir. Filtreler eğitim süreci sırasında öğrenilmektedir.

Ağdaki temel işlem; bir filtrenin (veya çekirdeğin) bir görüntüye (veya önceki bir katmandan bir öznitelik haritasına) uygulandığı evrişim işlemidir. Bir giriş görüntüsü I ve bir filtre F verildiğinde, evrişim işlemi matematiksel olarak şu formül ile gösterilebilir:

$$S(i,j) = (I * \mathrm{F})(i,j) = \sum_{m}\sum_{n} I(i-m, j-n)\ x\ F(m,n) \qquad (2.102)$$

Formülde $S(i,j)$, (i,j) konumunda ortaya çıkan öznitelik haritasını; I, girdi görüntüsü veya öznitelik haritasını; F, evrişim filtresini; (i,j), hesaplanmakta olan öznitelik haritasının konumunu; (m,n), F filtresinin boyutlarını; m ve n üzerindeki toplam ise, filtre ile giriş görüntüsünün yerel bir bölgesi arasındaki piksel bazında çarpımını ifade etmektedir. Filtre, giriş görüntüsü üzerindeki tüm piksellerde belirli aralıklarla gezdikten sonra, iki boyutlu bir matris (öznitelik haritası) üretir. Sonuçta da görüntünün belirli özellikleri (kenarlar, dokular vb.) tespit edilmiş olur.

2.8.2.5.2.1. Kenar Algılama[166]

Bir kenar, genellikle görüntü yoğunluğunda veya birinci türevinde bir süreksizlikle ilişkili olan önemli bir yerel değişikliktir (Sponton ve Cardelino, 2015). Kenar algılama işlemi, görüntü üzerinde en temel bilgilerin elde edilmesini sağlamaktadır. Kenar bilgileri, görüntüden elde edilen öznitelikler içinde en çok ihtiyaç duyulanlardan birisidir (Mohapatra, 2019). Kenarlar; giriş görüntüsünde bulunan yüksek frekanslı alanları simgeler. Bu tür bilgiler elde edilirken yatay, dikey veya farklı rotasyonlarda çeşitli filtre ya da maskeler görüntü üzerinde işleme alınır. Bu işlem, evrişim işlemidir. Bu işlemin sonundaki çıkışta, yüksek frekanslı yeni görüntünün kenar bilgileri tespit edilmiş olur. Geleneksel yöntemlerle (Sobel, Prewitt, Gabor,

[165] Aktivasyon haritası
[166] Edge detection

Robert) elle oluşturulan matrislerle bu işlemler yapılırken (Chaple, Daruwala ve Gofane, 2015), evrişimli sinir ağlarında böyle bir işlemin yapılmasına gerek yoktur. Çünkü evrişim işlemi boyunca, özellikle ilk katmanlarda ağın kendisi bu tür bilgileri otomatik olarak çıkarır (Znamenskaya ve Doroshchenko, 2021).

Kenar algılama işlemi, matris hesaplamalarındaki gibidir, sadece burada özel bir matris bulunmaktadır. Örneğin; 0 ve 10 değerlerinden oluşan 6x6'lık matriste, 10 ve 0'ın kesiştiği nokta, kenar bilgisini verecektir. Çünkü matris elemanları arasında hızlı geçişin olduğu yer frekansın yüksek olduğu alandır (Canny, 2009). Bir kenar bulma filtresi ile giriş matrisine (6x6) evrişim işlemi uygulandığında işlemler aşağıdaki gibi olacaktır:

10	10	10	10	10	10
10	10	10	10	10	10
10	10	10	10	10	10
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

*

1	1	1
0	0	0
-1	-1	-1

=

0	0	0	0
30	30	30	30
30	30	30	30
0	0	0	0

Şekil 66. Kenar Tespit İşlemi

Şekil 66'da kenar tespit işleminin kısa bir örneği gösterilmektedir. Giriş matrisindeki gri renkteki alan kenar bölgesinin kesişimini; kırmızı renkli alan ise evrişim işleminin başlayacağı alanı göstermektedir. Kenar bulma filtresi ise 3x3'lük bir matriste en alt satır en koyu renk, orta satır orta koyu renk, en üst satır ise açık renk olmak üzere konumlandırılmıştır. Görüntü piksellerle ifade edildiği zaman, küçük değerler koyu rengi, büyük değerler ise açık rengi ifade eder. Elde edilen çıkış matrisinde, kenar kesişimi olan bölgede 30; diğer aralıklarda ise 0 değeri elde edilmiştir. 0 değeri, daha koyu alanları simgelerken, 30 değeri daha açık renkli alanları simgelemektedir. Başka bir deyişle kenar, açık renkle ifade edilecektir.

Evrişimli sinir ağlarında kenar tespitinin ardındaki düşünce; hızlı yoğunluk değişiminin olduğu alanları vurgulayan filtrelerin kullanılmasıdır (Muntarina vd., 2023). Elle yapılan kenar algılama filtrelerinden Sobel, Prewitt ve Gauss Laplasyen (LoG) filtreleri yaygın olarak kullanılmaktadır. Bu filtreler genellikle evrişim işlemi kullanılarak uygulanır.

- ***Sobel Operatörü:*** 1968 yılında Irwin Sobel tarafından geliştirilmiştir (Nilsson, 2010, s. 131). Operatör, her ikisi de yoğunluk değişimlerine duyarlı 3x3 dikey ve yatay çekirdekten oluşan iki filtreleme penceresi kullanmaktadır. Görüntü işleme ve bilgisayarlı görüde yaygın olarak kullanılan bir kenar algılama yöntemidir ve kenarlara karşılık gelen yüksek mekânsal frekans bölgelerini vurgulamak için tasarlanmıştır**.** Yatay kenarların algılanması aşağıdaki gibi gösterilir:

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad (2.103)$$

Dikey kenarların algılanması ise şu şekildedir:

$$G_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad (2.104)$$

Bu filtrelerin görüntü I'ya uygulandığında çıktı genellikle gradyan büyüklüğü olarak gösterilir:

$$G = \sqrt{(G_x * I)^2 + (G_y * I)^2} \quad (2.105)$$

Formülde G_x ve G_y, yatay ve dikey Sobel filtrelerini; $*$, evrişim işlemini; $G_x * I$ ve $G_y * I$, sırasıyla yatay ve dikey Sobel filtreleri ile görüntünün evrişimi sonucu elde edilen sonuçları; G ise her pikseldeki kenar kuvvetini temsil etmektedir. G gradyanının büyüklüğü, görüntüdeki ilgili noktadaki kenarın gücünü temsil eder ve daha büyük değerler daha güçlü kenarlara karşılık gelir.

- ***Prewitt Operatörü:*** Prewitt (1970) tarafından geliştirilmiştir. Operatör, görüntüyü yatay ve dikey yönlerde küçük, ayrılabilir ve tamsayı değerli bir filtreyle evrişim işlemine aktarır ve bu nedenle Sobel operatörüne kıyasla daha düşük hesaplama maliyetine sahiptir. Ancak, Sobel operatörüne göre daha kaba bir gradyan tahmini yapar, bu nedenle yüksek frekanslı kenarları algılamada daha az hassas olabilir.

  Prewitt operatörü, iki 3x3 çekirdekli evrişim kullanılarak görüntü yoğunluk fonksiyonunun gradyanının hesaplanmasına dayanır. Yatay kenarları bulmak için kullanılan maske aşağıdaki gibidir:

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad (2.106)$$

  Dikey kenarları bulmak için kullanılan maske ise aşağıdaki gibidir:

$$G_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad (2.107)$$

  Görüntüdeki her piksel, her iki yöndeki gradyanları hesaplamak için bu maskelerle sarılır. G_x ve G_y gradyanları elde edildikten sonra, genel kenar büyüklüğü G şu şekilde hesaplanır:

$$G = \sqrt{G_x^2 + G_y^2} \tag{2.108}$$

Kenar yönü θ ise şu şekilde gösterilir:

$$\theta = atan^2(G_Y, G_x) \tag{2.109}$$

Formülde yer alan arktanjant ($atan$) fonksiyonu, görüntü üzerindeki bir pikselin gradyan vektörünün yönünü, yani kenarın uzanma açısını belirlemektedir. Bu açı, kenarın yatay eksene göre yaptığı açıyı temsil eder ve genellikle radyan ya da derece cinsinden ifade edilir.

- ***Gaussian Laplasyen (LoG):*** Marr ve Hildreth (1980) tarafından geliştirilmiştir. Marr-Hildreth Operatörü olarak da bilinir. Görüntünün Gauss filtresiyle yumuşatılmasından sonra bir Laplasyen filtresinin uygulanmasını içerir. LoG filtresi aşağıdaki gibi uygulanır:

$$LoG\ (x, y) = \nabla^2(G(x, y)) \tag{2.110}$$

Formülde ∇^2, ikinci dereceden türevleri içeren Laplasyen operatörünü; $G(x, y)$, Laplasyen uygulanmadan önce görüntüyü yumuşatan Gauss fonksiyonunu ifade etmektedir.

LoG filtresi, bir Gauss farkı (DoG) filtresi kullanılarak da hesaplanabilir:

$$DoG\ (x, y) = G_{\sigma_1}(x, y) - G_{\sigma_2}(x, y) \tag{2.111}$$

Formülde $G_{\sigma_1}(x, y)$ ve $G_{\sigma_2}(x, y)$, farklı standart sapmaları (σ_1 ve σ_2) olan Gauss filtrelerini ifade etmektedir.

2.8.2.5.2.2. Piksel Ekleme[167]

Evrişim işleminden sonra, çıkış matrisi ile giriş matrisi arasında oluşan boyut farkı yönetilebilir. Bazı durumlarda; giriş ve çıkış matrisi eşit boyutlarda olmayabilir. Her iki matrisi aynı boyutlara getirmek için, giriş matrisine fazladan pikseller eklemek gerekmektedir (Joyce ve Verschelde, 2024). Piksel ekleme işlemi yapılmadan önce, piksel elemanlarının değerinin ne olacağına karar vermek gerekmektedir. Piksel ekleme; genellikle evrişim işleminden sonra uzamsal boyutları korumak amacıyla giriş görüntüsüne uygulanır (Liu vd., 2022). Piksel ekleme işlemiyle, evrişim işleminin uzamsal boyutunun çok hızlı bir şekilde azalmaması sağlanır, bu durum ise özellikle derin ağlarda önemli bir ayrıntıdır.

[167] Padding

p, giriş görüntüsündeki her bir kenara eklenecek piksel değerini belirtmek üzere:

$$p = \frac{K-1}{2} \tag{2.112}$$

Formülü ile p değeri elde edilir. Formülde K, filtrenin boyutunu ifade etmektedir. Piksel ekleme işlemi uygulandıktan sonra ise, çıktı boyutu aşağıdaki formül ile gösterilir:

$$o = (n + 2p - f + 1) \times (n + 2p - f + 1) \tag{2.113}$$

Formülde n, giriş boyutunu; f, filtre boyutunu ve p ise dolgu miktarını ifade etmektedir.

Piksel ekleme işlemi aşağıdaki gibi yapılabilir:

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	1	2	5	6	3	0	0
0	0	2	3	4	6	7	5	0	0
0	0	8	7	6	5	7	6	0	0
0	0	2	3	5	6	7	8	0	0
0	0	4	5	3	2	1	6	0	0
0	0	1	4	5	3	2	6	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

2	2	2	3	4	6	7	5	5	5
1	1	1	1	2	5	6	3	3	3
1	1	1	1	2	5	6	3	3	6
3	2	2	3	4	6	7	5	5	7
8	7	8	7	6	5	7	6	6	7
3	2	2	3	5	6	7	8	8	7
5	4	4	5	3	2	1	6	6	1
4	1	1	4	5	3	2	6	6	2
1	1	1	4	5	3	2	6	6	6
4	4	4	5	3	2	1	6	6	6

Şekil 67. Piksel Ekleme İşlemi

Şekil 67'de, piksel ekleme işleminin 2 farklı yolu gösterilmektedir. Örneğin; p değeri 2 olduğunda, sol tarafta bulunan matrisi çevreleyen 2 katmanlı bir matris (2x2) 0'lar ile doldurulabilir veya sağ tarafta bulunan 2 katmanlı bir matrisin en dış iki katmanı 1. ve 2. sırada kopyalanıp genişletilebilir.

Giriş matrisi elemanlarının nxn şeklinde kopyalanarak, etrafının nxn hayali matris şeklinde doldurulması, giriş matrisinin etrafının 0'lar ile doldurulmasından daha maliyetli olmasına rağmen daha fazla duyarlılık sağladığı için tercih edilebilir. Çünkü evrişim işlemi yapıldığında, matrisin köşeleri 0 ile doluysa, çıkış matrisinin de dış kısımlarında 0'lar, çarpma ve toplama işlemi yapıldığı için toplam değeri sürekli olarak aşağıya çekecektir. Çarpma sırasında 0 değerleri, giriş matrisindeki ana değerleri veya filtredeki değerleri 0 yapacaktır. Değerler 0 olduğu zaman, elde edilen çıkış matrisinde birbirinden uzakta ayrık değerler oluşacaktır. Piksel değerlerinin dışa doğru kopyalanması işlemi yapıldığında ise, matrisin dış kısmına eklenen hayali piksellerden oluşan matris, görüntüye dair piksellerden oluştuğu için fark, 0 değerlerinin uygulanmasındaki gibi çok olmayacaktır. Dolaysıyla, evrişim işleminin çıkışında da mutlaka görüntüye dair bilgilere yakın çıkışlar elde edilecektir. Yani daha hassas bir çıkış matrisine sahip olunacaktır.

2.8.2.5.2.3. Adım Kaydırma[168]

Adım, filtrenin evrişim işlemi sırasında üzerinde hareket ettiği piksel sayısını ifade etmektedir. n adım, filtrenin bir seferde n piksel hareket ettiği anlamına gelir ve modelin performansı ile verimliliğini etkileyen önemli bir hiperparametre olarak kabul edilmektedir. Adım işlemi, evrişimli filtrelerin giriş verileriyle nasıl etkileşime girdiğini kontrol eder ve çıktı özellik haritalarının boyutunu etkiler (Riad vd., 2022). Çıktı boyutları $W_{çıktı}$ ve $H_{çıktı}$, aşağıdaki formülle hesaplanır:

$$W_{çıktı} = \frac{W_{girdi} + 2P - F_W}{S} + 1$$
$$H_{çıktı} = \frac{H_{girdi} + 2P - F_H}{S} + 1 \quad (2.114)$$

Formülde W_{girdi} ve H_{girdi}, giriş görüntüsünün genişliği ve yüksekliğini; F_W ve F_H, filtrenin genişliğini ve yüksekliğini; S, adımı ve P giriş görüntüsüne uygulanan dolguyu[169] ifade etmektedir.

Adım kaydırma işleminde filtre, görüntü matrisi üzerinde evrişim işlemine alınırken, filtrenin kaç piksel aralıkta kaydırılacağı bulunur. Örneğin; 9x9'luk bir giriş matrisine 3x3'lük bir filtre matrisi adım sayısı s=2 olacak şekilde uygulandığında, oluşacak çıkış matrisinin boyutu değişecektir.

3	7	1	2	6	4	0	3	5
5	0	8	3	5	8	6	1	4
7	6	9	7	5	2	7	9	3
9	0	2	1	3	0	2	5	1
3	5	4	2	1	7	9	8	4
8	8	5	4	2	1	7	9	0
1	8	7	0	9	2	0	1	2
0	2	1	2	1	5	6	3	5
8	3	2	9	0	7	0	2	1

Şekil 68. Adım Kaydırma İşlemi

Şekil 68'de adım sayısı s=2 seçildiğinde işlem, başlangıç noktasından (şeklin en sol üstü) 2 piksel sonra başlar. Yani işlem; giriş matrisinin ilk satırındaki 3. eleman olan 1'den başlayacaktır. 2 piksellik gruplar halinde ilerleyecek olan adım kaydırma işlemi, ilk 2 satırı

[168] Stride
[169] Padding

bitirdiği zaman yeni bir satıra geçtiğinde ilk sütundaki 3. eleman olan 7'den itibaren işleme başlayacaktır. 2'li, 3'lü veya 5'li kaydırma işlemi yapıldığında, bilgi kaybı daha fazla olur ve boyut ise biraz daha küçülür. Başka bir deyişle; adım değeri ile çıkış matrisinin boyutları birbiriyle ters orantılıdır. Adım değeri büyük seçilirse, çıkış matrisinin boyutu daha da küçülecektir (Riad vd., 2022).

Formül (2.114) bu örnek için uygulandığında:

$W_{girdi} = 9$

$P = 0$

$F_W = 3$

$S = 2$

$$W_{çıktı} = \frac{9 + 0 - 3}{2} + 1$$

$$= 4$$

Çıkış matrisi $W_{çıktı}$; 4x4 boyutunda olacaktır.

Adım kaydırma işlemi, evrişim işleminin daha hızlı yapılmasını sağlar. Çünkü tüm matris daha kısa sürede taranır. Fakat çıkış matrisinin boyutunu azalttığı için de duyarlılık kaybedilmiş olur.

2.8.2.5.3. Havuzlama Katmanı[170]

Havuzlama katmanı, öznitelik haritalarının uzamsal boyutunu azaltmak için kullanılır, böylece hesaplama yükü ve aşırı öğrenme riski azaltılarak genelleştirmenin iyileştirilmesine yardımcı olur. Ayrıca bu katman, modelin veri setinden farklı özellikleri öğrenmesini de sağlar. İşlem, bir ağın çıktısını belirli bir konumda tek bir değerle özetlemeye izin veren ve doğrusal olmayan bir dönüşümdür (Sharma ve Mehra, 2019). Havuzlama işlemi, evrişim katmanının çıktı matrisinden belirli bir alt örnekleme[171] yaparak bu çıktının özetlenmesini sağlar. Bu sayede, sonraki katmanlarda işlenecek parametre sayısı ve hesaplama yükü azalmış olur. Havuzlama katmanları, modellerin giriş verilerinde meydana gelen küçük dönüşüm, çevirme veya ölçekleme gibi değişikliklere karşı daha değişmez[172] hale gelmesini sağlar. Bu işlem, modeli bilgisayarlı görü için uygun hale getirerek benzer görüntü desenlerini tanımlamasına yardımcı

[170] Pooling Layer
[171] Downsampling
[172] Invariant

olur (Yamashita vd., 2018; Nirthika vd., 2022; Zafar vd., 2022). Maksimum, ortalama ve global havuzlama olmak üzere üç farklı türü bulunmaktadır:

- **Maksimum Havuzlama:** En yaygın kullanılan havuzlama yöntemidir. Çıktı, giriş öznitelik haritası üzerinde kayan tanımlı bir penceredeki (genellikle 2×2 veya 3×3) maksimum değerdir (Gholamalinezhad ve Khosravi, 2020). Formülü şu şekildedir:

$$Y(i,j) = \max_{(m,n) \in R(i,j)} X(m,n) \tag{2.115}$$

Formülde $Y(i,j)$, havuzlanmış özellik haritasında (i,j) konumundaki çıktı değerini ifade etmek üzere; $X(m,n)$, giriş özellik haritası değerlerini; $R(i,j)$, havuzlamanın uygulandığı bölgeyi (örneğin, 2×2 pencere); max ise; maksimum fonksiyonu ifade etmektedir. Dört satır ve dört sütundan oluşan iki boyutlu bir matris Şekil 69'da gösterilmektedir.

2	3	1	4
5	6	7	8
9	10	11	12
13	14	15	16

Şekil 69. Havuzlama 1 - Orijinal Matris

Matrise; iki boyutlu çekirdek (filtre) ile piksel ekleme işlemi yapılmadan (dolgusuz) iki adım şeklinde havuzlama işlemi yapılıp, her pencereden maksimum değer seçildiğinde Şekil 70'deki gibi bir matris elde edilmiştir.



Şekil 70. Havuzlama 2 - Maksimum Havuzlama

Matris, her pencereden sadece en büyük değer alınarak oluşturulmuştur. Böylece matris, alt örneklemeye tabi tutulmuş ve maksimum havuzlama çıktısı olarak daha küçük bir 2x2 matris elde edilmiştir.

- **Ortalama Havuzlama:** Bu yöntem maksimum değeri seçmek yerine, pencere içindeki değerlerin ortalamasını hesaplar. Ortalama havuzlamada, penceredeki tüm değerlerin

ortalaması alınarak işlem yapılır (Gholamalinezhad ve Khosravi, 2020). Formülü şu şekilde gösterilmektedir:

$$Y(i,j) = \frac{1}{|R(i,j)|} \sum_{(m,n)\in R(i,j)} X(m,n) \tag{2.116}$$

Formülde $Y(i,j)$, havuzlanmış özellik haritasında (i,j) konumundaki çıktı değerini ifade etmek üzere; $|R(i,j)|$, havuzlama bölgesindeki eleman sayısını ve $\sum X(m,n)$ ise havuzlama bölgesindeki tüm değerlerin toplamını ifade etmektedir.



Şekil 71. Havuzlama 3 - Ortalama Havuzlama

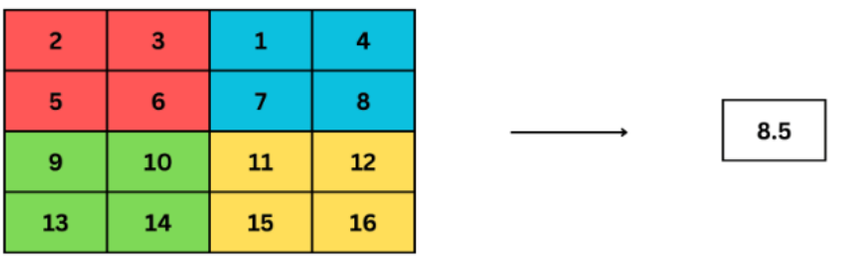
Şekil 71'de 4x4'lük bir matris gösterilmektedir. Bu giriş matrisine 2x2'lik bir filtre uygulandığında, giriş matrisinin ilk tarama alanında bulunan 4 piksel değeri toplanmış ve ağırlıklı ortalamaları aldıktan sonra son değeri çıkış matrisine yansıtılmıştır. Başka bir deyişle buradaki 4 piksel, tek bir piksel ile ifade edilmiştir. Tıpkı maksimum havuzlamada olduğu gibi burada da boyut azaltma işlemi yapılmıştır. Filtre, giriş matrisindeki ikinci işlemde de 2 piksel kaydırıldıktan sonra ikinci dört piksel değerini tek bir pikselde gösterilecek şekilde ortalamasını almıştır. Üçüncü ve dördüncü işlemler de aynı şekilde ilerledikten sonra, çıkış matrisi 2x2 boyutunda oluşmuştur. Ortalama havuzlama işlemi yapıldığında, bu piksellerin tamamının bir değeri elde edilmiş olur. Böylece maksimum havuzlama yöntemine kıyasla daha hassas bir sonuç elde edilir. Ortalama havuzlama, her havuzlama bölgesindeki piksellerin ortalama değerini aldığından, görüntüdeki ince ayrıntıların kaybolmasına neden olur. Birden fazla piksel tek bir pikselde birleştirildiğinden, havuzlanmış öznitelik haritası orijinal görüntüde bulunan ince değişiklikleri ve kenar bilgilerini doğru bir şekilde koruyamaz. Ortalama havuzlama uzamsal olarak değişmez bir işlemdir; yani özelliğin görüntüdeki konumundan bağımsız olarak, ortalama havuzlama sonucu aynı olacaktır. Böyle bir durum da nesne algılama görevleri gibi konumsal bilgilerin önemli olduğu belirli durumlarda istenmeyebilir (Zhao ve Zhang, 2024).

- ***Global Havuzlama:*** Global havuzlamada, normal havuzlama katmanlarından farklı olarak pencere, çekirdek boyutu veya adım kavramı yoktur. Tüm matris bir bütün olarak dikkate alınır. Global havuzlama, tüm bir özellik haritasını kanal başına tek bir değere indirger (Boxue vd., 2018). Formülü şu şekildedir:

$$Y = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X(i,j) \tag{2.117}$$

Formülde Y, kanal başına tek skaler çıkışı; H ve W, özellik haritasının yüksekliği ve genişliğini; $X(i,j)$ ise özellik haritasının piksel değerlerini ifade etmektedir.

Global havuzlama katmanları, yoğun katmanlardan önce doğrudan kullanılır. Evrişimli katmanlar, çekirdek yinelemelerine ve adımlara bağlı olarak her görüntüyü alt örnekleme yapar. Ancak, farklı boyutlardaki görüntülere uygulanan aynı evrişimler, farklı şekillerde bir çıktıyla sonuçlanacaktır (Bulusu vd., 2021). Tüm görüntülere aynı oranda alt örnekleme yapıldığından, daha büyük görüntüler daha büyük çıktıya sahip olacaktır. Bu, boyut uyumsuzluğuna neden olabileceğinden sınıflandırma için yoğun katmanlara geçirilirken bir sorun oluşturabilir. Hiperparametrelerde veya model mimarisinde değişiklik yapılmadan, tüm görüntü şekillerine uygulanabilir bir model uygulamak zor olabilir. Bu tür sorunlar, global havuzlama yöntemi kullanılarak çözülebilir.



Şekil 72. Havuzlama 4 - Global Havuzlama

Şekil 72'de global havuzlama örneği gösterilmektedir. Burada tüm özellik haritası bir bütün olarak ele alınmış ve tüm değerlerin ortalaması alınarak tek bir skaler çıktı değeri elde edilmiştir.

Havuzlama katmanlarında öğrenilebilir parametre bulunmamaktadır (Shah ve Bhavsar, 2022). Çünkü bu katmanlar, giriş verilerinde yalnızca sabit işlemler yapabilir. Böylece; eğitim sırasında herhangi bir ağırlık veya önyargı ayarlanmadan boyut azaltma sağlanmış olur.

2.8.2.5.4. Aktivasyon Katmanı

Aktivasyon katmanı, doğrusal olmayan öznitelik ilişkilerini ve karmaşık örüntüleri öğrenebilmek için evrişimli sinir ağlarında bir anahtar görevi görür. Aktivasyon katmanı teknik olarak "katman" değildir çünkü aktivasyon katmanında hiçbir parametre veya ağırlık öğrenilmez. Çoğunlukla, aktivasyon fonksiyonlarının uygulanmasından hemen sonra evrişim işlemi geldiği için ağ mimarisi diyagramlarından çıkarılır (Manessi ve Rozza, 2019). Aktivasyon fonksiyonu, ağa doğrusal olmayan bir özellik kazandırır ve girdilerden çıktılara karmaşık eşlemeler öğrenilmesini sağlar. Ağ, bir aktivasyon fonksiyonuna sahip olmadan doğrusal bir regresyon modeli gibi davranır ve karmaşık örüntüleri yakalayamaz.

Bir aktivasyon katmanı, $W_{girdi}\ x\ H_{girdi}\ x\ D_{girdi}$ boyutunda bir giriş kabul eder ve ardından ilgili aktivasyon fonksiyonunu uygular. Fonksiyon, eleman bazında uygulandığından, çıktı her zaman giriş boyutuyla aynı olmak durumundadır:

$$W_{girdi} = W_{\text{çıktı}}, H_{girdi} = H_{\text{çıktı}}, D_{girdi} = D_{\text{çıktı}} \tag{2.118}$$

2.8.2.5.5. Tam Bağlantılı Katman

Tam bağlantılı katmanda, ilgili katmandaki her bir nöron bir önceki katmandaki tüm nöronlarla bağlantılıdır. Ağlarda, mimari derinliğe bağlı olarak bir veya daha fazla tam bağlantılı katman bulundurulur (Basha vd., 2020). Bu katmanlar genellikle evrişim ve havuzlama katmanlarından sonra gelir. Birkaç evrişim ve havuzlama katmanından sonra, çıktı genellikle bir vektör boyutunda düzleştirilir[173] ve bir veya daha fazla tam bağlantılı katmandan geçirilir. Burada, önceki katmanlar tarafından öğrenilen tüm özelliklerdeki bilgiler konsolide edilerek entegrasyon işlemleri gerçekleştirilir. Tam bağlantılı bir katmanda; her bir giriş nöronu her bir çıkış nöronuna bağlıdır. Her bağlantının bir ağırlığı ve her nöronun bir önyargısı vardır. Her nöronun çıktısı da bir aktivasyon fonksiyonundan geçirilen girdilerinin ağırlıklı toplamıdır. Tam bağlantılı katmanın formülü aşağıdaki gibi ifade edilmektedir:

$$\begin{aligned} z &= W \times X + b \\ a &= aktivasyon(z) \end{aligned} \tag{2.119}$$

Formülde X, n boyutlu katmanın giriş vektörünü; W, katmanın ağırlık matrisini $(m \times n)$; b, katmanın önyargı vektörünü; z, aktivasyon fonksiyonu uygulanmadan önce girdilerin ağırlıklı toplamını ve a, aktivasyon fonksiyonunu uyguladıktan sonra katmanın çıktısını ifade etmektedir.

[173] Flatten

Örneğin; X giriş vektörü olmak üzere, tam bağlantılı katmandaki durumu için:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{2.120}$$

Ağırlık matrisi W:

$$W = \begin{bmatrix} w_{11}\ w_{12}\ w_{13} \\ w_{21}\ w_{22}\ w_{23} \end{bmatrix} \tag{2.121}$$

Önyargı vektörü b:

$$b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{2.122}$$

Çıktı z (aktivasyondan önce):

$$z = W \times X + b = \ W = \begin{bmatrix} w_{11}\ w_{12}\ w_{13} \\ w_{21}\ w_{22}\ w_{23} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{2.123}$$

Aktivasyon fonksiyonu uygulandıktan sonra çıktı a şu şekilde olur:

$$a = aktivasyon(z) \tag{2.124}$$

a, tam bağlantılı katmanın son çıktısıdır.

2.8.2.5.6. Çıkış Katmanı

Evrişimli sinir ağında çıkış katmanı, son katmandır. Modeldeki eğitim örnekleri arasında oluşturulan hatayı hesaplamak için çıktı katmanında çeşitli kayıp fonksiyonları kullanılır. Bu hata, gerçek çıktı ile öngörülen çıktı arasındaki farkı ortaya çıkarır. Ardından ağ, öğrenme süreci aracılığıyla optimize edilecektir[174]. Son katman olan çıkış katmanında sınıflandırma problemleri için genellikle softmax aktivasyon fonksiyonu kullanılır. Softmax fonksiyonu, ham çıktı puanlarını sınıflar üzerinde bir olasılık dağılımına dönüştürür.

2.8.2.5.7. Düzleştirme[175] İşlemi

Düzleştirme işlemi; önceki katmandan gelen çok boyutlu çıktıyı, genellikle daha fazla işlem için tam bağlantılı bir katmana beslemeden önce tek boyutlu bir vektöre dönüştürür. Bu vektör daha sonra sınıflandırma veya regresyon işlemleri için tam bağlantılı (yoğun) katmanlara beslenir (Jeczmionek ve Kowalski, 2021). Genellikle evrişim ve havuzlama katmanlarından

[174] Geri yayılım (backpropagation) ve optimizasyon algoritmaları ile hatalar azaltılmaya çalışılır.
[175] Flatten

sonra kullanılır. Uzamsal özellikleri çıkaran evrişim ve havuzlama katmanları ile sınıflandırma veya regresyon görevlerini gerçekleştiren tam bağlantılı katmanlar arasında bir köprü görevi görür ve aynı zamanda hesaplama verimliliğini artırır (Naidu vd., 2023). Evrişim ve havuzlama katmanlarından iki veya üç boyutlu çıkan matrisler, tam bağlantılı katmanlara geçemez çünkü tam bağlantılı katmanlar için 1 boyutlu giriş gerekmektedir. Düzleştirme işleminden önce örneğin 128 filtreye sahip 12x12 boyutlarında 64 görüntü (64, 12x12, 128) içeren üç boyutlu bir tensör, düzleştirme işlemi ile bir boyutlu vektöre dönüşür: (64, 18432). Burada 18432 değeri, 12x12x128'den gelmektedir.

2.8.2.6. Evrişimli Sinir Ağları Modelleri

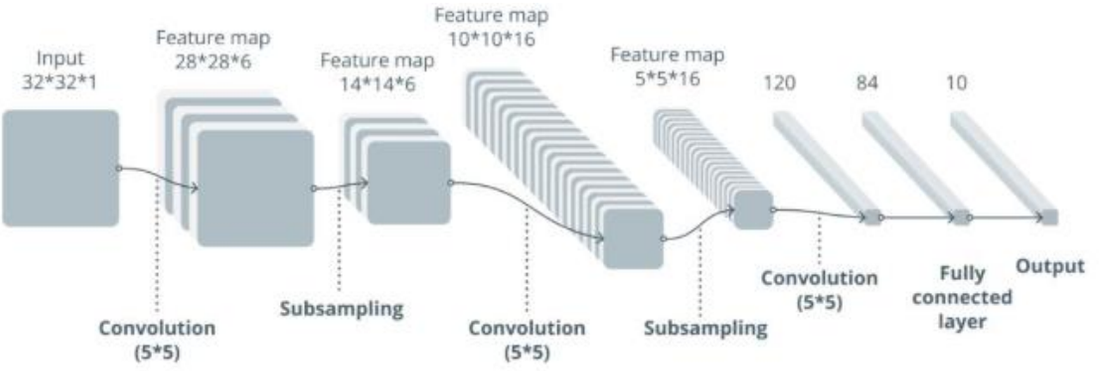
Evrişimli sinir ağlarında gerçekleştirilen iyileştirmelerle farklı özelliklere, yeteneklere ve performanslara sahip mimariler geliştirilmiştir. Bu mimariler; görüntü sınıflandırması ve nesne algılama gibi görevlerin ilerlemesinde oldukça önemli bir rol oynamıştır. Bu bölümde LeNet AlexNet, VGG ve EfficientNet modelleri açıklanmıştır.

2.8.2.6.1. LeNET

LeNET mimarisi (LeNet-5) Lecun vd. (1998) tarafından geliştirilmiştir. 1980'lerden beri üzerinde çalışılan bir konu olarak oldukça önem arz etmektedir. Evrişimli sinir ağının kullanıldığı model; el yazısı ile oluşturulan rakamların tanınmasını amaçlar. Modelde 32x32x1'lik bir görüntü girişi kullanılmıştır.

Ağ için kullanılan veri setinde sadece 1 kanallı, gri seviyeli ve 0-9 aralığında el yazısı rakamlar bulunmaktadır. Giriş ve çıkış katmanı dahil 8 katmandan oluşmaktadır. Modelde ilk etapta bir evrişim işlemi yapılmaktadır. Evrişim katmanında 6 adet filtre uygulanmış ve böylece 28x28x6'lık bir tensör[176] elde edilmiştir. Başka bir deyişle, 1 kanallı görüntü, evrişim katmanında 28x28x6'lık bir öznitelik haritasına dönüşmüştür. Daha sonra ise ortalama havuzlama yöntemi kullanılmıştır. Bu katmanda ise görüntü boyutu 14x14x6'lık bir matrise düşürülmüştür. Havuzlama katmanında kanal sayısı değişmemektedir. Bu aşamadan sonra tekrar evrişim işlemi yapılmış ve 16 kanallı bir filtre kullanılmıştır. Dolaysıyla çıkış 16 kanallı bir tensör olmuştur. Tensörün yükseklik ve genişlik değerleri ise 10x10'dur. Daha sonra tekrar ortalama havuzlama yöntemi kullanılmış ve ardından 2 adet tam bağlantılı katmandan sonra, veri seti 10 sınıflı olduğundan dolayı softmax aktivasyon fonksiyonu uygulanmış ve 10 sınıflı bir sınıflandırıcı elde edilmiştir.

[176] Birden fazla kanalı olan matrisler dizisi.



Şekil 73. LeNET Mimarisi
Kaynak: (Saxena, 2025)

Şekil 73'te LeNET ağının katmanları gösterilmektedir. Modelin en önemli özelliği ise aktivasyon fonksiyonu olarak Sigmoid ve Hiperbolik tanjant kullanılması ve önerilmesidir. Ancak; günümüzde aktivasyon fonksiyonu olarak evrişimli sinir ağlarında ReLu tercih edimektedir (Goh vd., 2024). Ayrıca LeNET, evrişimli sinir ağlarında geri yayılım kullanılarak eğitilen ilk ağlardan birisidir. İlk pratik derin öğrenme modellerinden biri haline gelmiş; AlexNet, VGG ve ResNet gibi modern evrişimli sinir ağlarının temelini oluşturmuştur.

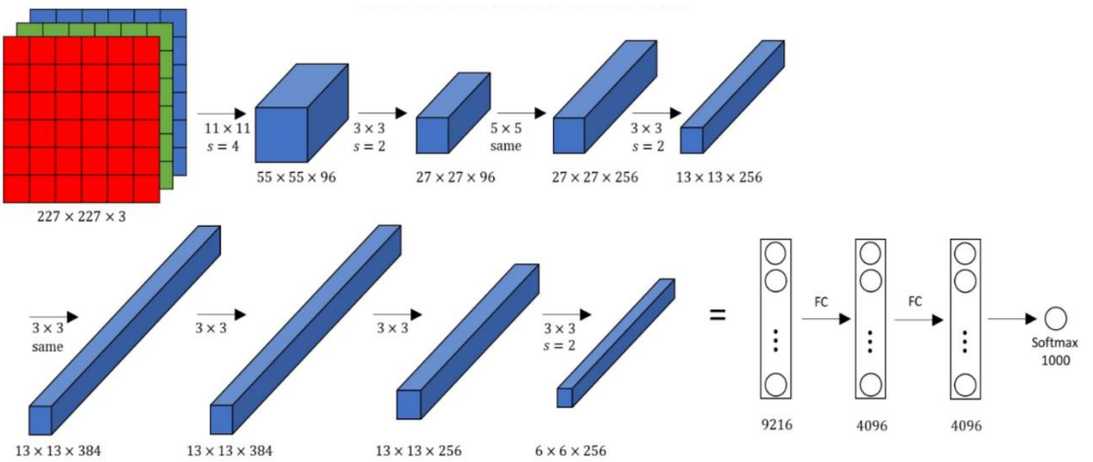
2.8.2.6.2. AlexNET

AlexNet, bilgisayarlı görü[177] alanında bir devrime yol açan derin bir evrişimli sinir ağıdır. Krizhevsky, Sutskever ve Hinton (2012) tarafından geliştirilmiştir. Model, 2012'deki ImageNet büyük ölçekli görsel tanıma yarışmasını kazandığında yaygın bir şekilde tanınmış ve önceki yöntemlerden önemli ölçüde daha iyi performans göstermiştir. AlexNet'ten önce, geleneksel makine öğrenmesi modelleri ve daha sığ evrişimli sinir ağları, büyük ölçekli görüntü sınıflandırmasında yüksek doğruluk elde etmekte zorlanıyordu. AlexNET, ikiye bölünerek her bir parça, bir adet GTX 580 ekran kartı üzerinde eğitilmiştir. Yaklaşık 60 milyon parametresinin çoğu, tam bağlantılı katmanlarda bulunmaktadır. Giriş ve çıkış katmanları dahil toplamda 10 katmandan (5 evrişim+3 tam bağlantılı) oluşmaktadır. Modelin getirdiği yenilikler arasında; her bir evrişim ve tam bağlantılı katmandan sonra ReLU aktivasyon fonksiyonunun kullanılması, boyutları azaltmak için bazı evrişim katmanlarından sonra maksimum havuzlama yönteminin kullanılması, aşırı öğrenmeyi önlemek için tam bağlantılı katmanlarda bırakma işlemi ve çok sınıflı sınıflandırma için çıkış katmanında Softmax aktivasyon fonksiyonunun kullanılması bulunmaktadır.

[177] Computer vision

AlexNET'i daha önceki modellerden ayıran en önemli özellik, giriş görüntüsünü 2 paralel evrişimli ağda eğitmesidir. Yani; bir evrişimli modelin yanı sıra ikinci bir evrişimli modelin varlığıdır. Bu modeller en son aşamada birleştirilmiştir.

Girişte, 227x227x3'lük bir renkli görüntü kullanılmıştır. Birinci evrişim katmanında 4 adımlı ve 11x11 boyutunda 96 çekirdek kullanılmış, bu çekirdekler ReLU aktivasyon fonksiyonu ile işlenmiş ve ardından maksimum havuzlama yöntemi uygulanmıştır. İkinci evrişim katmanı ise, birinci evrişim katmanının çıktısını girdi olarak kullanmış ve 5x5x48 boyutunda 256 çekirdek üzerinde işlem görmüştür. Üçüncü evrişim katmanında 3x3x256 boyutunda 384 çekirdekli; dördüncü evrişim katmanında 3x3x192 boyutunda 384 çekirdekli ve beşinci evrişim katmanında ve 3x3x192 boyutunda 256 çekirdek olarak işlem görmüştür. Modelde üçüncü, dördüncü ve beşinci katmanlarda havuzlama veya normalizasyon işlemi yapılmamıştır. Tam bağlantılı katmanların her biri 4096 nörona sahiptir. Çıkış katmanı, 1000 sınıflı etiketin olasılıklarını hesaplayan Softmax katmanıdır.



Şekil 74. AlexNET Mimarisi
Kaynak: (Gündüz ve Cedimoğlu, 2019, s. 13)

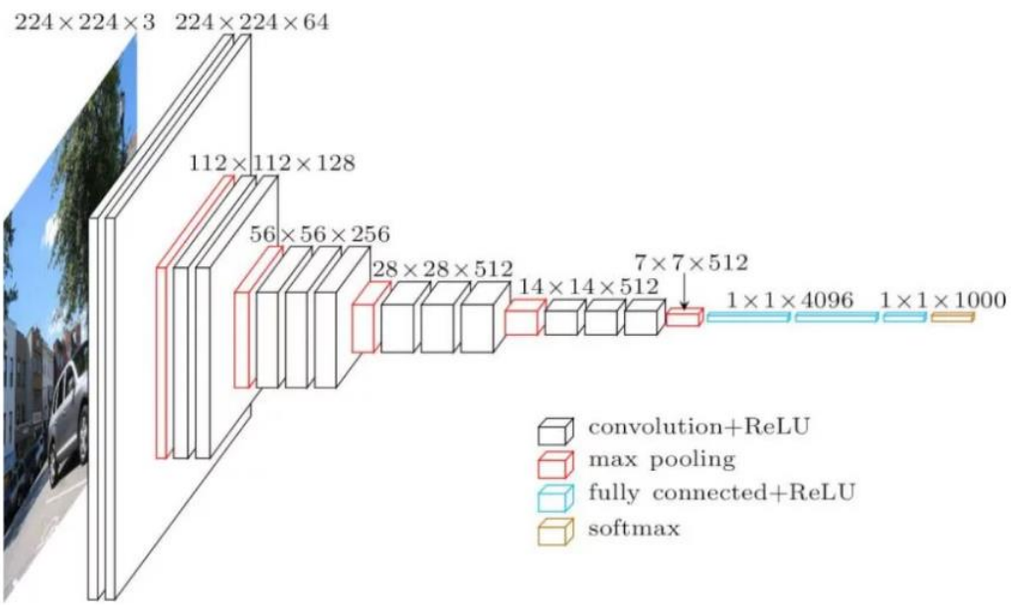
Şekil 74'te AlexNET ağının mimarisi gösterilmektedir. 2012'den önce, klasik modeller kullanılırken elde edilen eğitim hatası %25-%30 aralığında iken; AlexNET'te bu değer %15,3 olmuştur (Krizhevsky, Sutskever ve Hinton, 2012). Katman sayısı arttıkça, eğitim hatası da azalmaktadır. Ancak; eğitim hatasının azalması ile katman sayısının artması, kesinlikle ters orantılı olarak düşünülmemelidir. Katmanların çok olması, modelin başarılı olması ile doğru orantılı değildir (Sun vd., 2016).

2.8.2.6.3. VGG16

Simonyan ve Zisserman (2014) tarafından geliştirilen model, 2014 ImageNet büyük ölçekli görsel tanıma yarışmasında tanıtılmıştır. 13 evrişimli ve 3 tam bağlantılı katman olmak üzere

toplam 16 katmandan oluşmaktadır. Yaklaşık 138 milyon parametreye sahip olan modelde her bir evrişim katmanından sonra ReLU aktivasyon fonksiyonu uygulanmıştır. Giriş katmanında 224x224x3'lük bir görüntü ile işlemlere başlanmaktadır. Evrişim katmanları sırasıyla 64, 128, 256, 512 ve 512 filtreden oluşmaktadır. Toplamda 5 adet olmak üzere her bir evrişim katmanından sonra boyutları azaltmak için 2x2'lik maksimum havuzlama yöntemi kullanılmıştır. Her biri 4096 nörona sahip olmak üzere 3 tam bağlantılı katman ile Softmax aktivasyon fonksiyonu uygulanan 1000 sınıflı çıkış katmanı bulunmaktadır.

VGG16'nın ayırt edici özelliği, LeNet ve AlexNet'ten farklı olarak her evrişim bloğunda birden fazla ardışık evrişim katmanına yer vermesidir. Özellikle giriş katmanından sonra 2 adet 64 filtreli evrişim katmanının ardarda uygulanması ve hemen ardından havuzlama yapılması, modelin derinliğini ve özellik çıkarma kapasitesini artırmıştır.



Şekil 75. VGG16 Mimarisi
Kaynak: (Datta vd., 2024)

Şekil 75'te VGG16 ağı gösterilmektedir. Özellik çıkarmayı iyileştirmesi ve transfer öğrenmesindeki etkinliğinin yanında çok fazla parametre içerdiğinden hesaplama açısından maliyetli olması ve yüksek bellek ihtiyacı gibi dezavantajları da mevcuttur.

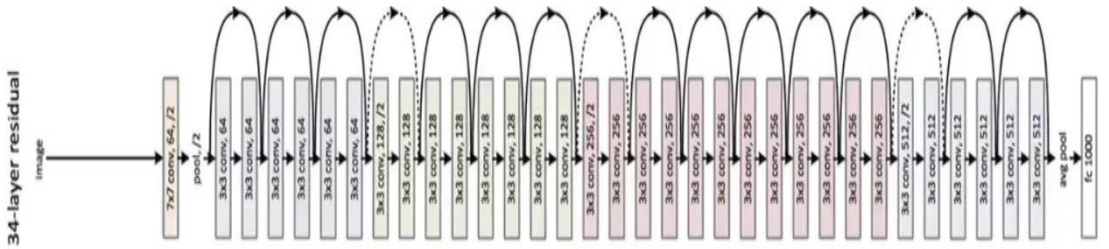
Temel VGG modelden çok çeşitli varyantlar üretilmiştir:

- ***VGG19:*** İşleyişi VGG16'ya benzer ancak 19 katmana sahiptir (16 evrişimli + 3 tam bağlantılı katman). Daha iyi doğruluk skoruna sahip olmasına rağmen daha yüksek hesaplama maliyeti bulunmaktadır (Sudha ve Ganeshbabu, 2021).

- ***Toplu Normalizasyonlu VGG16 (VGG16-BN):*** Her evrişimli katmandan sonra toplu normlizasyon katmanlarının eklendiği varyanttır. Eğitim kararlılığını artırır ve aşırı öğrenmeyi azaltır (Pattanajak ve Malekmohamadi, 2019).
- ***VGG-Face:*** Yüz tanıma için büyük ölçekli yüz veri kümeleri üzerinde eğitilen bir VGG16 çeşididir. VGGFace2 gibi veri kümelerinde önceden eğitilmiştir (Cao vd., 2018).
- ***Azaltılmış Parametreli VGG16 (Hafif VGG):*** Bazı araştırmacılar, doğruluğu korurken parametre sayısını azaltmak için VGG16'yı değiştirmiştir. Çoğunlukla mobil ve uç bilişim uygulamalarında kullanılmaktadır (Ye vd., 2021).

2.8.2.6.4. ResNET

ResNet[178], He vd. (2015) tarafından geliştirilmiş ve derin sinir ağlarının eğitimi sırasında karşılaşılan gradyanların kaybolması[179] problemine çözüm getirerek bilgisayarlı görü alanında adeta bir devrime yol açmıştır. Özellikle geleneksel derin ağların katman sayısı artırıldığında performans düşüşü gözlenirken, ResNet bunun önüne geçmiştir. Modelin temel özelliği, katmanlar arasında artık[180] bağlantılar kullanılmasıdır. Bu yapılar sayesinde model, her katmanda öğrenilecek çıktıyı doğrudan değil, o katmanın girişinden çıkışına kadar olan farkı öğrenmeye çalışır. Bu yapıya ise ResBlock denilmektedir (Wang vd., 2023). ResNet modelleri genellikle Stokastik Gradyan İnişi[181] ile eğitilir, ancak bazı varyantlarında Momentum ve Adam gibi optimizasyon algoritmaları da kullanılabilir.



Şekil 76. ResNET Mimarisi
Kaynak: (Souai, 2024)

Şekil 76'da ResNET mimarisi gösterilmektedir. Düz modelde çok verimli bir sonuç elde edilemezken; ResBlock eklemesi yapıldığında yüzden fazla katmanla birlikte daha iyi sonuçlar elde edilmiştir (Oyedotun vd., 2020). ResBlock uygulaması ile, gradyanların yok olması problemine çözüm getirilmiştir (Shah vd., 2016). Çünkü gradyanlar kaybolduğunda öğrenme azalmakta ve öğrenme azaldığında da teoride beklenilen optimizasyon durumu

[178] Residual Network
[179] Vanishing gradient
[180] Residual
[181] SGD

gerçekleşmemektedir. ResNET'in birçok farklı sayda katmandan oluşan versiyonları da bulunmaktadır.

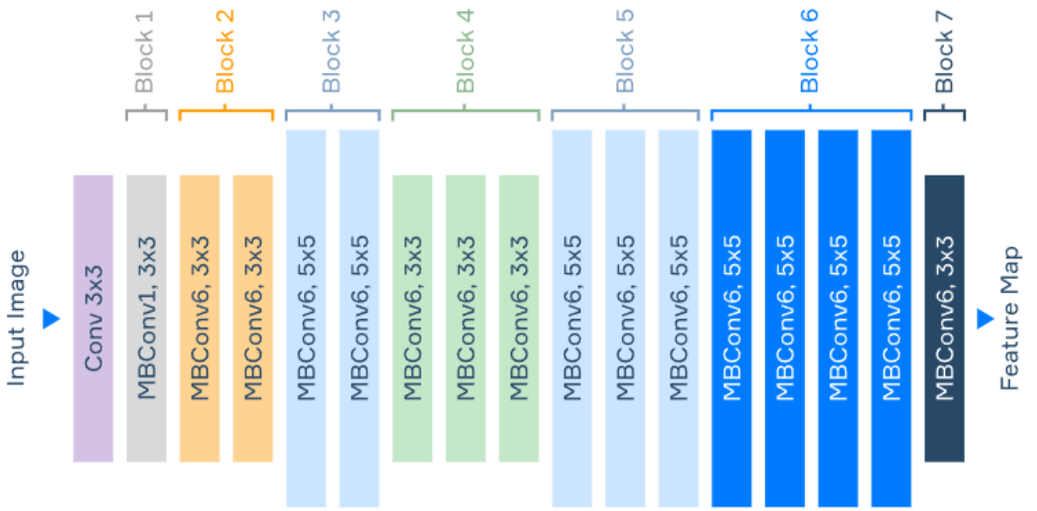
Tablo 11. ResNET Varyantları

Model	Katman Sayısı	Parametre Sayısı
ResNET18	18	11.7 milyon
ResNET34	34	21.8 milyon
ResNET50	50	25.6 milyon
ResNET101	101	44.5 milyon
ResNET152	152	60.2 milyon

Tablo 11'de temel ResNET modelinden üretilen varyant modeller görülmektedir. En düşük boyutlu ResNET 18 katmandan oluşup 11.7 milyon parametreye sahipken; en yüksek boyutlu katman sayısı 152 olup toplamda 60.2 milyon parametresi bulunmaktadır.

2.8.2.6.5. EfficientNET

Tan ve Le (2019) tarafından geliştirilen evrişimli sinir ağları ailesidir. Geleneksel evrişimli sinir ağları mimarilerine kıyasla daha az parametre ve hesaplamaya sahip olup, görüntü sınıflandırma görevlerinde doğruluğu ve verimliliği artırmak amacıyla geliştirilmiştir. EfficientNet'in sunduğu temel yenilik; ağın derinliğini, genişliğini ve çözünürlüğünü dengeli bir şekilde sistematik olarak ölçekleyen bileşik ölçekleme yöntemidir.



Şekil 77. EfficientNET Mimarisi
Kaynak: (Khatri vd., 2023)

Şekil 77'de EfficientNET temel mimarisi görülmektedir. Model, minimum parametrelerle performansı optimize etmek için Nöral Mimari Arama[182] (NAS) kullanılarak tasarlanmıştır. 7

[182] Makine öğrenmesi alanında yaygın olarak kullanılan bir model olan yapay sinir ağlarının tasarımını otomatikleştirmek için kullanılan bir tekniktir. Bir insanın sinir ağını manuel olarak ayarlaması ve neyin iyi çalıştığını öğrenmesi sürecini ele alır ve bu görevi daha karmaşık mimarileri keşfetmek amacıyla otomatikleştirir.

bloktan oluşan model, 3×3'lük evrişim işleminden sonra Mobil Ters Darboğaz Evrişimi[183] katmanının izlediği 3×3 evrişim katmanıyla devam eder. Her blok, değişen boyutlarda (3x3 veya 5x5) bir veya daha fazla MBConv katmanından oluşmaktadır. Model, ilk blokta MBConv1'i ve sonraki bloklarda MBConv6'yı kullanmıştır. Buradaki 1 ve 6 değerleri genişleme oranını göstermektedir. Yaklaşık 5.3 milyon parametreden oluşan temel model, ağ karmaşıklığı ve kaynak kullanımını dengeler. Model ve türevleri, ImageNet ve çeşitli transfer öğrenme görevlerinde diğer evrişimli sinir ağı mimarilerinden daha iyi performans göstermiştir (Tan ve Le, 2019; Chen, Isa ve Liu, 2025).

EfficientNet modelleri, B0'ın temel model ve B7'nin en gelişmiş model olmak üzere toplamda 8 adettir.

Tablo 12. EfficientNET Varyantları

Model	Katman Sayısı	Parametre Sayısı
EfficientNet-B0	18	5.3 milyon
EfficientNet-B1	18	7.8 milyon
EfficientNet-B2	19	9.1 milyon
EfficientNet-B3	20	12 milyon
EfficientNet-B4	23	19 milyon
EfficientNet-B5	27	30 milyon
EfficientNet-B6	31	43 milyon
EfficientNet-B7	38	66 milyon

Tablo 12'de EfficientNET varyantlarının katman ve parametre sayıları gösterilmektedir. Temel model olan B0, 18 katman ve 5.3 milyon parametre ile en küçük model olurken; B7, 38 katman ve 66 milyon parametre ile en büyük modeldir.

2.8.2.7. Özyinelemeli Sinir Ağları

Özyinelemeli sinir ağları[184], zaman serisi veya sıralı verileri işlemek için tasarlanmış bir derin öğrenme sınıfıdır. Modelin temelleri Werbos (1990) tarafından 1974'te geliştirilen ve yinelemeli sinir ağlarını eğitmek için temel oluşturan BPTT[185] kavramına dayanmaktadır. İleri beslemeli sinir ağlarının aksine, girdi dizilerini işlemek için kendi belleğini kullanarak önceki girdilerin içeriğini koruma gibi oldukça dikkat çekici bir özelliğe sahiptir (Baruah ve Organero, 2024). Böyle bir özellik, bağlamın ve veri noktalarının sırasının son derece önem arz ettiği doğal dil işleme, konuşma tanıma ve zaman serisi tahmini gibi uygulamalar için oldukça önemlidir. Ses işleme (Phan vd., 2017), konuşma tanıma (Graves, Mohamed ve Hinton, 2013),

[183] MBConv. Giriş kanallarını genişleten, derinlik evrişimi uygulayan ve daha sonra bunları genellikle kalıntı bağlantılarla daha düşük bir boyuta geri yansıtan, derin sinir ağlarında verimli hesaplama için tasarlanmış bir evrişim bloğu.
[184] Recurrent Neural Networks - RNN
[185] BackPropagation Through Time

metin analizleri (Oyelakin vd., 2022), video çözümleme (Xu vd., 2019) kestirimci bakım (Ramu ve Narayanan, 2023), duygu örüntüleri tanıma (Kurniasari ve Setyanto, 2020) gibi çok çeşitli alanlarda kullanılmaktadır. Veri türlerinin ve görevlerin son yıllarda giderek karmaşıklaşmaya başlaması, diğer mimarilerde olduğu gibi özyinelemeli sinir ağları ve varyantlarında da sürekli bir yeniliğe yol açmıştır. Sıralı veri analizi, tahmini ve sınıflandırmasından çok modlu öğrenme ve gerçek zamanlı karar alma sistemleri gibi karmaşık görevlere doğru genişleyen bir konu alanı için özyinelemeli sinir ağları tabanlı çözümler geliştirilmiş ve denenmiştir (Tsantekidis, Passalis ve Tefas, 2022).

Temel mimarisi, tıpkı derin ağlarda olduğu gibi bir girdi katmanı, bir gizli katman ve bir çıktı katmanından oluşmaktadır. Ağlar içinde bilginin döngüye girmesine izin veren yinelemeli bağlantılara sahiptir. t, her bir andaki adım olmak üzere h_t gizli durumunun güncellenmesi şu formül ile gösterilmektedir (Tjandra, Sakti ve Nakamura, 2017):

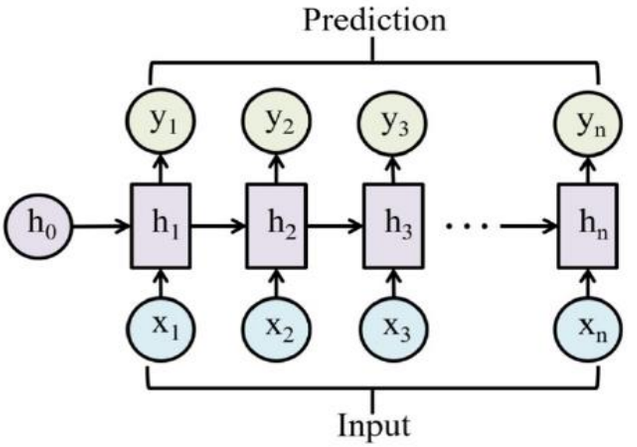
$$h_t = \sigma_h(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{2.125}$$

Formülde h_t, hem mevcut girdiden hem de önceki gizli durumdan bilgi alan t zaman adımındaki gizli durumu; σ_h, aktivasyon fonksiyonunu; W_{xh}, giriş ve gizli katman arasındaki ağırlık matrisini; x_t, t zaman adımındaki girdiyi (bir kelime veya özellik vektörü); W_{hh}, tekrarlayan bağlantı için kullanılan ağırlık matrisini; h_{t-1}, geçmiş bilgileri taşıyan önceki gizli durumu ve b_h, önyargı vektörünü ifade etmektedir. Formül, yeni gizli durum h_t'nin mevcut girdi x_t ve önceki gizli durum h_{t-1} kullanılarak nasıl hesaplandığını açıklamaktadır.

y_t, gizli durum h_t'ye bağlı olan t zaman adımındaki çıktı olmak üzere aşağıdaki formülle gösterilir (Levine, 2021):

$$y_t = \sigma_y(W_{hy}h_t + b_y) \tag{2.126}$$

Formülde σ_y, çıkış katmanının aktivasyon fonksiyonunu; W_{hy}, gizli katman ve çıktı katmanları arasındaki ağırlık matrisini; b_y, modelin çıktıyı ağırlıklı gizli durumdan bağımsız olarak ayarlamasına olanak tanıyan önyargı terimini ifade etmektedir. Gizli durum h_t, geçmiş ve şimdiki girdilerden bilgi yakalar. Bu bilgi W_{hy} tarafından dönüştürülür ve b tarafından ayarlanır. Son olarak, aktivasyon fonksiyonu σ_y, sonucu işleyerek nihai çıktı y_t'yi üretir.



Şekil 78. Temel RNN Mimarisi
Kaynak: (Mienye, Swart ve Obaido, 2024, s. 4)

Şekil 78'de özyinelemeli sinir ağının temel mimarisi gösterilmektedir. Diyagramda $(x_1, x_2, x_3, ..., x_n)$, sıralı veri girişlerini ifade etmektedir. Her bir x_n, t zaman diliminde cümledeki bir kelime, videodaki bir kare veya zaman serisindeki bir değer olabilir. $(h_0, h_1, h_2, ...)$ ise gizli durumları göstermektedir. Gizli durumlar, önceki zaman adımlarından bilginin yakalanmasını sağlar. Her bir zaman adımı t'deki gizli durum h_t, önceki gizli durum h_{t-1} ve girdi x_t kullanılarak hesaplanır. Ağda bulunan yinelemeli bağlantılar, modelin geçmiş bağımlılıkları saklamasına olanak tanımaktadır. $(y_1, y_2, y_3, \dots, y_n)$ ise çıktıları ifade etmekte ve her bir gizli durum h_t, çıktı y_t'yi üretmek için kullanılmaktadır. Çıktı, önceki tüm girdilerden gelen bilgileri kodlayan aktif gizli duruma bağlıdır. Ağın temel özelliği, yinelemeli bağlantılar aracılığıyla geçmiş bilgileri korurken sıralı girdileri işlemesinde yatmaktadır. Ağdaki gizli durum, temelde normal bir ileri beslemeli ağdaki gizli bir katman gibi düşünülebilir.

Ağ işlemlerinin özü, güncel girdiyi önceki gizli durumla bütünleştiren gizli durumun yinelemeli hesaplanmasını içermesidir (Rezk vd., 2020). Bu yinelemeli hesaplama, ağın dinamik zamansal davranış sergilemesini sağlamaktadır. Aktivasyon fonksiyonu σ_h'nin seçimi, ağın davranışında önemli bir rol oynar ve ağın verilerdeki karmaşık desenleri öğrenmesini sağlar (Mienye, Swart ve Obaido, 2024). Aktivasyon fonksiyonu olarak da hiperbolik tanjant (tanh) aktivasyon fonksiyonu sıklıkla kullanılmaktadır. Ancak, geri yayılım sırasında gradyanların üstel olarak büyüdüğü veya küçüldüğü "gradyanların kaybolması" gibi bir dezavantajı bulunmaktadır (Lalapura, Amudha ve Satheesh, 2021).

Özyinelemeli sinir ağı türleri aşağıdakiler gibi açıklanabilir:

- ***Çift Yönlü RNN (Bidirectional RNN-BiRNN):*** Schuster ve Paliwal (1997) tarafından geliştirilmiştir. İki katman kullan ağ, biri ileri yönde girdiyi işlerken diğeri ters yönde

işler. Konuşma tanıma görevlerinde diğer yapay sinir ağları mimarilerinden daha iyi sonuçlar vermiştir.

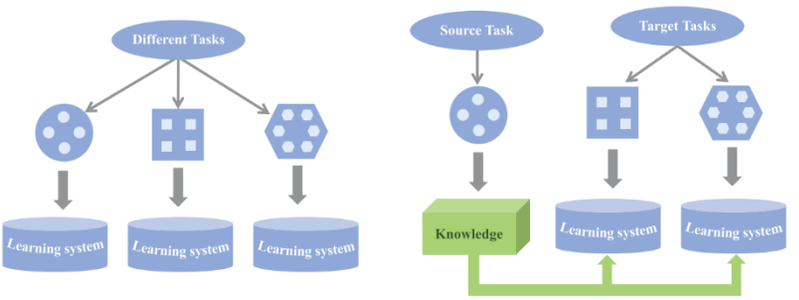
- ***Derin RNN (Deep RNN-DRNN):*** Daha karmaşık örüntüleri öğrenmek için birden fazla katman kullanan mimaridir.
- ***Geçitli Yinelemeli Birim (Gated Recurrent Unit-GRU):*** Cho vd. (2014) tarafından geliştirilmiştir. Bilgi akışını kontrol etmek için bir geçit mekanizması (sıfırlama ve güncelleme kapıları) sunar. LSTM'lerden hesaplama açısından daha verimlidir.
- ***Uzun Kısa Süreli Bellek (Long Short-Term Memory-LSTM):*** Hochreiter ve Schmidhuber (1997) tarafından geliştirilmiştir. Gradyanların kaybolması sorununu ele almak için bir bellek hücresi ve üç kapı (giriş, unutma ve çıkış) kullanır.

2.8.2.8. Transfer Öğrenmesi

Bilgi[186] transferi olarak da bilinen transfer öğrenmesi, makine öğrenmesi alanında sıklıkla kullanılan ön eğitim yöntemlerinden biri olup bir modeli, önceki benzer ilişkili görevlerin eğitilmiş modeliyle yeniden eğiten bir teknik olarak tanımlanabilir (Hosna vd., 2022). Büyük miktarda veri ve hesaplama gücü gerektiren derin bir sinir ağını baştan eğitmek yerine, transfer öğrenmesi teknikleri ile daha az veri ve daha az çalışma zamanı ile yüksek performans elde edilebilir. İşlem, önceden eğitilmiş modellerden yararlanır. Tekniğin arka planındaki temel düşünce; modelin hedef alandaki etiketsiz verileri, kaynak alandaki etiketli verilere dönüştürmesini sağlamaktır (Shen vd., 2022). Büyük veri setlerinde önceden eğitilmiş bir model zaten yararlı özellikleri öğrendiğinden, yeni bir görevde eğitim daha az epoch ve daha az hesaplama maliyeti ile sonuca ulaşacaktır. Örneğin; ImageNet veri setinde eğitilmiş bir model, yeni bir probleme uyarlanabilen genel özellikleri öğrenmiştir ve genellikle daha iyi doğruluk skoruna erişir.

Klasik makine öğrenmesi yöntemleri, test verileriyle aynı öznitelik alanına ve aynı dağılıma sahip çok sayıda eğitim örneğinin olması gibi bir gerekliliğe sahiptir. Dağılım veya öznitelikler değiştiğinde ise, birçok modelin yeni eğitim verileri kullanılarak en baştan oluşturulması gerekmektedir ve buna ek olarak yeterli eğitim verisi toplamak birçok görev için zaman alıcı ve maliyetlidir. Buna benzer durumlarda hem maliyeti azaltmak hem de yüksek başarımlı modellerin ağırlık ve parametreleri ile etkili bir eğitim ve test süreci gerçekleştirmek için transfer öğrenmesi metodları kullanılmaktadır (Pan ve Yang, 2009; Zhao vd., 2024).

[186] Knowledge



Şekil 79. Makine Öğrenmesi ve Transfer Öğrenmesi Farklılıkları
Kaynak: (Wang vd., 2019, s. 5)

Şekil 79'da klasik makine öğrenmesi ile transfer öğrenmesi süreçleri arasındaki farklılıklar gösterilmektedir. Klasik makine öğrenmesi teknikleri, her bir farklı görevi en baştan öğrenmeye çalışırken, transfer öğrenmesi teknikleri ise daha iyi eğitim verisi olduğunda, çeşitli görevlerden gelen bilgiyi hedef göreve aktarmaya çalışır. Makine öğrenmesi bağlamında transfer öğrenmesi; önceden çok çeşitli ağırlıklar ile eğitilmiş bir model ve parametrelerinin, tüm ağırlıkların korunarak kaynak modelden yeni ve daha farklı bir modele başarıyla aktarılabilme potansiyeli olarak da tanımlanabilir (Weiss, Khoshgoftaar ve Wang, 2016; Wang vd., 2019; Zhuang vd., 2020).

Transfer öğrenmesinde eğitim süreci, öncelikle büyük bir veri kümesi üzerinde (örneğin ImageNet) temel bir modelin eğitilmesini ve ardından optimum performansı elde etmek için daha küçük bir hedef veri kümesi kullanılarak ince ayar[187] yapılması şeklinde gerçekleşmektedir. Transfer öğrenmesinin en önemli faydası ise aşırı öğrenmeyi son derece etkili bir şekilde ortadan kaldırması, eğitim süreçlerini hızlandırması ve yeniden eğitim için gerekli veri kümesi boyutunu önemli ölçüde azaltmasıdır (Zhao vd., 2024; Yan vd., 2024).

Önceki bölümlerde bahsedilen evrişimli sinir ağı modelleri (LeNET, AlexNET, VGG16, ResNET ve EfficientNET) aynı zamanda birer ön eğitimli modellerdir ve transfer öğrenmesinde sıklıkla kullanılmaktadır.

[187] Fine tuning

ÜÇÜNCÜ BÖLÜM

3. UYGULAMA FAZLARINA YÖNELİK METODOLOJİLER

Bu bölümde, tezin literatür araştırması bölümünde bahsedilen teorik kısımların uygulama yöntemleri açıklanmıştır. Tezde iki farklı yapay zekâ tekniği kullanılmıştır: Birinci faz analizler için makine öğrenmesi yöntemleri; ikinci faz analizler için ise derin öğrenme yöntemleri.

3.1. Veri Setleri

3.1.1. Amerika Birleşik Devletleri İç Hat Uçuş Veri Seti

Birinci faz analizler, hesaplamalı arşiv biliminin uygulama alanına dahil olmakla birlikte, literatürde çoğunlukla belge tabanlı veriler üzerinde yoğunlaşan mevcut çalışmalardan farklı olarak yapılandırılmış veri setlerinin kullanımına odaklanmıştır. "Veri seti 1" olarak adlandırılacak olan bu set, Amerika Birleşik Devletleri'nde 1990-2009 yıllarını kapsayan ve aylık bazda olmak üzere 3.606.803 örnek ve 11 öznitelikten barındıran iç hat uçuş kayıtlarından oluşmaktadır. Veri seti 1 yapısal bütünlüğe sahip olması, geniş bir zaman aralığına dair veri içermesi, açık veri olması ve ayrıca arşivsel veri özelliği nedeniyle hesaplamalı arşiv bilimi perspektifinden anlamlı bir örnek teşkil ettiğinden dolayı tercih edilmiştir. Veriler, ABD Ulaştırma Bakanlığı tarafından düzenli olarak toplanmış, zaman içinde dijital ortama aktarılmış ve araştırmacıların kullanımına açık hâle getirilmiştir[188]. Veri seti, Academic Torrents veri paylaşım platformundan temin edilmiştir. Platform, büyük hacimli bilimsel veri setlerinin açık erişimle paylaşılmasını amaçlayan ve kar amacı gütmeyen "Institute for Reproducible Research" tarafından geliştirilmiştir (Cohen ve Lo, 2014; Lo ve Cohen, 2016). Platform ayrıca ABD (The University of Texas, University of Michigan, Carnegie Mellon University, Standford University vb.) ve İngiltere'de (University of Bristol, Imperial Collage London vb.) bulunan çok sayıda üniversite tarafından da kullanılmaktadır. Veri setine ait metaveriler ve açıklamaları aşağıdaki tabloda gösterilmektedir.

Tablo 13. Veri Seti 1 Hakkında Bilgiler

No	Sütun Adları	Veri Tipi	Açıklama
1	Kalkış	object	Kalkış havaalanının üç harfli kodu
2	Varış	object	Varış havaalanının üç harfli kodu
3	Kalkış Şehri	object	Kalkış şehir adı
4	Varış Şehri	object	Varış şehir adı
5	Yolcu Sayısı	int64	Kalkış noktasından varış noktasına taşınan yolcu sayısı
6	Koltuk Sayısı	int64	Uçaklardaki toplam koltuk sayısı
7	Uçuş Sayısı	int64	Kalkış ve varış noktası arasındaki uçuş sayısı
8	Mesafe	float64	Kalkış ve varış noktası arasında uçulan mesafe (mil)
9	Uçuş Tarihi	int64	Uçuş tarihi (format: yyyymm)

[188] https://academictorrents.com/details/a2ccf94bbb4af222bf8e69dad60a68a29f310d9a

10	Kalkış Şehri Nüfusu	int64	ABD Nüfus Sayım Bürosu tarafından rapor edilen kalkış şehrinin nüfusu
11	Varış Şehri Nüfusu	int64	ABD Nüfus Sayım Bürosu tarafından rapor edilen varış şehrinin nüfusu

Tablo 13'ten de görüleceği üzere orijinal veri setinde öznitelikler 11 sütuna yayılmıştır. Uçuşların başladığı ve sona erdiği havaalanlarının kodları ve isimleri, havaalanlarının bulunduğu şehir ve eyalet isimleri ile nüfusları, her bir uçuşta bulunan koltuk ve yolcu sayıları, uçuş sayısı, mil cinsinden kalkış ve varış noktaları arasındaki mesafe ile uçuşların yapıldığı tarih (ay ve yıl) bilgisi bulunmaktadır. Orijinal veri seti TSV uzantılıdır. Veri setinin içerisinde metaveri bulunmamaktadır. Metaveriler, sonradan elle eklenmiştir. Veri setinin boyutu 353 MB civarındadır.

Veri setine uygulanacak olan ilk işlem, ön işleme adımlarından oluşmaktadır. Bu veri seti özelinde yapılacak ön işlemler şu şekilde belirtilmiştir:

- Veri setinin satır ve sütun görünümünün ayarlanması,
- Metaveri bilgilerinin girilmesi,
- Sütun adlarının Türkçe'ye çevrilmesi,
- Tanımsız (NA), boş (null) ve 0 (sıfır) değerlerinin tespiti,
- Çapıklık-basıklık ve histogram-kutu grafiği analiziyle aykırı değer tespiti,
- Tanımsız, boş ve 0 değerlerin KNNImpute yöntemi ile doldurulması,
- Veri tipi dönüşümleri (object-string ve float-integer),
- Origin city sütununda bulunan şehir ve eyalet bilgisinin ayrılması,
- Destination city sütununda bulunan şehir ve eyalet bilgisinin ayrılması,
- Fly Date sütununda birleşik halde bulunan ay ve yıl değerlerinin ayrılması,
- Ayrılan tüm değerlerin farklı sütunlarda birleştirilmesi vb.

Ön işleme adımları tamamlandıktan sonra veri setine dair çeşitli tanımlayıcı istatistikler çıkarılmıştır. Bunlar arasında değişkenlerin birbirleriyle korelasyonunu gösteren korelasyon matrisi, temel standart istatistikler olan mod, medyan, standart sapma, standart hata, çeyreklik dilimleri, en küçük ve en büyük değerler bulunmaktadır.

Tanımlayıcı istatistiklerden sonra ise denetimli makine öğrenmesi algoritmalarından regresyon görevlerinde; doğrusal regresyon, yapay sinir ağları, rastgele orman, karar ağaçları ve k-en yakın komşu algoritmaları kullanılmıştır. Sınıflandırma görevleri için lojistik regresyon, yığınlama, rastgele orman, Naive Bayes, karar ağaçları, k-en yakın komşu, ve yapay sinir ağı algoritmaları kullanılmıştır.

3.1.2. Hava Görüntüsü Veri Seti (Aerial Image Dataset-AID)

İkinci faz analizlerde ise AID veri seti kullanılmıştır. Veri setinde 30 farklı görüntü sınıfı ve her sınıfta 600x600 boyutunda 198-378 arası örnek bulunmakta olup toplamda 10.000 adettir. Veri seti Xia vd. (2017) tarafından oluşturulmuştur. Bu set ise "veri seti 2" olarak adlandırılmıştır.

Tablo 14. Veri Seti 2 Hakkında Bilgiler

No	Orijinal Sınıf Adları	Boyut	Görüntü Sayısı
1	Havaalanı	600x600	324
2	Çıplak Arazi	600x600	279
3	Beyzbol Sahası	600x600	198
4	Plaj	600x600	360
5	Köprü	600x600	324
6	Merkez	600x600	234
7	Kilise	600x600	216
8	Ticari Alan	600x600	315
9	Yoğun Yerleşim	600x600	369
10	Çöl	600x600	270
11	Tarım Arazisi	600x600	333
12	Orman	600x600	225
13	Endüstriyel Bölge	600x600	351
14	Çayır	600x600	252
15	Orta Yerleşim	600x600	261
16	Dağ	600x600	306
17	Park	600x600	315
18	Otopark	600x600	351
19	Oyun Alanı	600x600	333
20	Gölet	600x600	378
21	Liman	600x600	342
22	Tren İstasyonu	600x600	234
23	Tatil Köyü	600x600	261
24	Nehir	600x600	369
25	Okul	600x600	270
26	Seyrek Yerleşim	600x600	270
27	Meydan	600x600	297
28	Stadyum	600x600	261
29	Depolama Tankları	600x600	324
30	Viyadük	600x600	378
Toplam			10.000

Tablo 14'ten de görüleceği üzere orijinal veri setindeki sınıflar; havaalanı, çıplak arazi, beyzbol sahası, plaj, köprü, merkez, kilise, ticari alan, yoğun yerleşim, çöl, tarım arazisi, orman, endüstriyel bölge, çayır, orta yerleşim, dağ, park, otopark, oyun alanı, gölet, liman, tren istasyonu, tatil köyü, nehir, okul, seyrek yerleşim, meydan, stadyum, depolama tankları ve viyadük olmak üzere 30 adettir. Veri setinin boyutu 2.55 GB civarındadır.

Veri seti 2'ye uygulanacak olan ilk işlem, ön işleme adımlarından oluşmaktadır. Bu veri seti özelinde yapılacak ön işlemler ise şu şekilde belirtilmiştir:

- Görüntülerin girdi ve çıktı yollarının oluşturulması,
- Görüntülerin yeniden boyutlandırılması,
- Sınıfların değişkene kaydedilmesi,

- 30 sınıfın bulunduğu her bir klasöre erişim ve işletim sistemi işlemleri,
- Görüntü etiketlerinin oluşturulması vb.

Ön işleme adımları tamamlandıktan sonra veri seti 2'ye temel bir evrişimli sinir ağı (Convolutional Neural Networks-CNN) modeli uygulanmıştır. CNN modelleri uygulanırken çeşitli ek işlemler (düzenlileştirme ve optimizasyon) de yapılmıştır. Ayrıca, temel CNN modelinin başarımını daha da artırmak için EfficientNET modelleri transfer öğrenmesi kapsamında uygulanmıştır. Tüm modellere ait değerlendirme metrikleri oluşturulmuş, sonuçlar değerlendirilmiş ve yorumlanmıştır.

3.2. Donanım

Bu tezde makine öğrenmesi ve derin öğrenme kapsamında yapılan tüm analizler AMD Ryzen 9 serisi, 16 çekirdekli, 4.2 GHz (max. 5.7 GHz) hızında ve L3 128 MB önbellek ile çalışan 7950x3D işlemci, 128 GB DDR5 RAM ve Nvidia RTX4080 16 GB GPU RAM görüntü belleğinden oluşan Windows 10 64 Bit işletim sistemine sahip masaüstü bilgisayarda gerçekleştirilmiştir. Ayrıca, Archivematica uygulaması için sanal bir sunucu üzerinde 16 GB DDR4 RAM, 8 çekirdekli, 20MB önbellekli ve 2.40 GHz hızında Xeon ES2630v3 işlemci ile çalışan Ubuntu 18.04.6 LTS işletim sistemi kurulmuştur. Programın versiyonu ise 1.12.2'dir.

3.3. Platform ve Araçlar

Tez kapsamında gerçekleştirilen tüm analizler, veri bilimi uygulamaları için geliştirilmiş Anaconda[189] 2023 dağıtımı içerisinde bulunan Jupyter Lab platformu aracılığı ile yürütülmüştür. Jupyter Lab platformu içerisinde ise Jupyter Notebook kullanılmıştır. Anaconda dağıtımı, birçok Python kütüphanesini dahili olarak barındırdığı için tercih edilmiştir. Çalışmada makine öğrenmesi uygulamaları Python programlama dilinin en son sürümü ile gerçekleştirilirken, derin öğrenme uygulamaları ise çeşitli bağımlılıklardan dolayı Python 3.10 sürümü üzerinden ilerlemiştir. Bu bağımlılıklar arasında en önemlisi Tensorflow kütüphanesinin 2.10 sürümü sonrası Windows üzerinde doğrudan GPU desteğinin kaldırılmasıdır. Bu nedenle, GPU temelli analizler için Anaconda üzerinde ayrı bir çalışma ortamı (environment) oluşturulmuş ve Python 3.10, cudatoolkit 11.8, cudnn 8.8.0 ve Tensorflow 2.10 sürümleri kurulmuştur. Ayrıca diğer tüm gerekli kütüphaneler de çalışma

[189] Veri bilimi, makine öğrenmesi ve sayısal hesaplamalar için yaygın olarak kullanılan; Python ve R programlama dillerine dayalı açık kaynak kodlu bir dağıtım. İçerisinde Jupyter Lab, NumPy, Pandas, Matplotlib, SciPy ve Scikit-learn gibi birçok kütüphane ve araç önceden entegre olarak gelmektedir (https://www.anaconda.com).

ortamı içerisine yüklenmiştir. Tezde kullanılan kütüphane sayısı çok fazla olduğundan, örnek kütüphaneler aşağıdaki tabloda gösterilmiştir.

Tablo 15. Tezde Kullanılan Bazı Örnek Kütüphaneler

Kütüphane	Versiyon	Amaç
Numpy	1.23.5	Bilimsel Hesaplama
Pandas	2.2.3	Veri Analizi ve Manipülasyonu
Seaborn	0.13.2	Veri Görselleştirme
Matplotlib	3.8.4	Veri Görselleştirme
Scikit-Learn	1.6.1	Makine Öğrenmesi Algoritma Hesaplamaları
Tensorflow	2.10	Derin Öğrenme Algoritma Hesaplamaları
PyTorch	2.5.1	GPU Entegrasyonu
BagIt	1.8.1	Arşiv Verisi Paketleme

Tablo 15'te tezde kullanılan bazı örnek kütüphaneler gösterilmektedir. Özellikle Tensorflow, derin öğrenme hesaplamalarına yönelik geniş bir alt kütüphane yapısına sahiptir. Keras kütüphanesi ise Tensorflow içerisinde olduğundan dolayı tabloda gösterilmemiştir. Tensorflow'a ait örnek alt kütüphaneler ise Tablo 16'da özetlenmiştir.

Tablo 16. Tensorflow Alt Kütüphane Örnekleri

from	import
tensorflow.keras.preprocessing.image	ImageDataGenerator
	img_to_array
	load_img
tensorflow.keras.optimizasyon algoritmasıs.legacy	*
tensorflow.keras.utils	to_categorical
tensorflow.keras.models	Sequential
	BatchNormalization
	Conv2D
	MaxPooling2D
	Activation
	Flatten
	Dropout
	Dense

Yapılan tüm işlem ve analizler sonucunda elde edilen veriler, önce JSON formatında kaydedilerek süreç verisi (provenans) oluşturulması sağlanmıştır. Bu amaçla; her bir model eğitimi, hiperparametre seçimi ve test aşaması sonunda ilgili girdiler (veri seti bilgileri, model parametreleri, sonuç metrikleri vb.) olarak JSON dosyasına yazdırılmış; böylece tüm süreç yapılandırılmış hâle getirilmiştir. Ardından, tüm sonuçlar ve gerekli dokümanlar (grafik, tablo vb.) ile orijinal veri setlerinin yanında türevleri de BagIt kütüphanesi ile paketlenerek Archivematica açık kaynak kodlu programına yüklenmiştir. Programda ise çeşitli arşivsel işlemler (ön işlemler ile AIP, SIP ve DIP paketleri) yapılarak süreç sonlandırılmıştır. Ayrıca tüm süreç bilgisi, PROV-ML tabanlı diyagramda bağlamsal olarak gösterilmiştir.

BagIt kütüphanesi, disk veya ağ tabanlı depolama ile her türlü dijital içeriğin aktarımını desteklemek için Boyko vd. (2012) tarafından geliştirilen hiyerarşik bir dosya paketleme biçimidir. Kütüphane, yük[190] ve etiketten[191] oluşmaktadır. Yükün içeriği, dijital materyallerden oluşmaktadır. Etiket ise, paketin depolanmasını ve aktarılmasını kolaylaştırmak ve belgelemek için tasarlanmış metaveri dosyalarını ifade etmektedir. Kütüphanenin altyapısı aşağıdaki zorunlu bileşenlerden oluşmaktadır:

- ***Paket Bildirimi (Bag Declaration):*** Bir paketin okunması ve işlenebilmesi için önyüklemede gerekli etiketleri içerir ve "bagit.txt" formatındadır.
- ***Yük Dizini (Payload Directory):*** Paketin içindeki saklanacak veriyi (içerik) barındırır.
- ***Yük Bildirimi (Payload Manifest):*** Belirli bir paket özet değeri[192] algoritması kullanılarak oluşturulan yük dosyalarını ve bu dosyaların özet değerlerini listeleyen bir etiket dosyasıdır (Boyko vd. (2012).

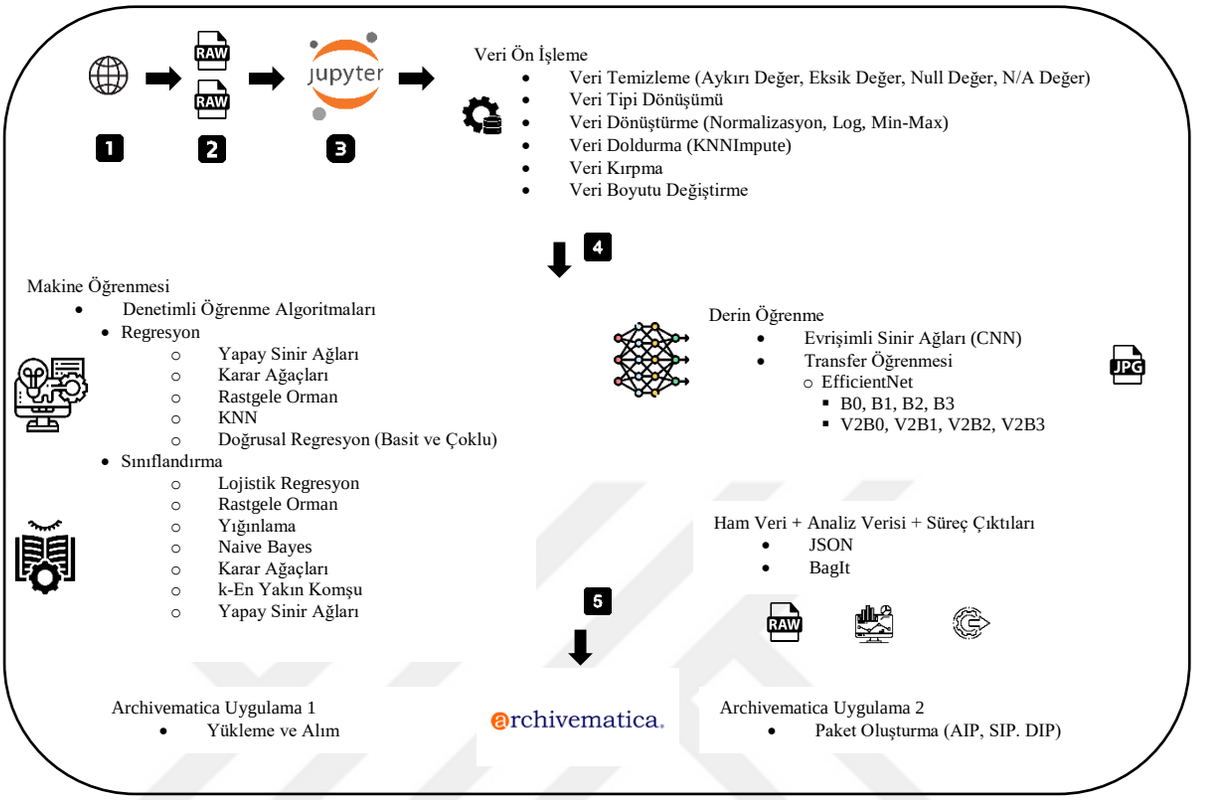
Zorunlu bileşenlerin yanında etiket bildirimi[193], paket metaverisi[194], işlem dosyası[195] gibi opsiyonel bileşenler de kullanılabilmektedir.

Archivematica açık kaynak kodlu programı ise dijital materyallerin OAIS[196] modeline uygun olarak sisteme dahil edilmesinden erişime kadar işlenmesine olanak tanıyan entegre ve açık kaynak kodlu bir yazılım paketidir. Program, tüm servislerin web tabanlı olarak kontrol edilebileceği şekilde geliştirilmiştir. METS, PREMIS, Dublin Core, Kongre Kütüphanesi ve BagIt spesifikasyonu vb. standartları kullanarak dijital depoda otantik, güvenilir ve sistemden bağımsız Arşiv Bilgi Paketi[197] oluşturulmasını sağlar. Program, özelleştirilmiş Xubuntu Linux dağıtımının üzerine inşa edilen ve dijital bilgi paketlerini bir dosya sistemi dizininden geçiren veri hattı[198] özelliğindedir (van Garderen, 2010).

[190] Payload - Burada, "paketlenen veri" anlamında kullanılmıştır.
[191] Tag
[192] Checksum
[193] Tag manifest
[194] Bag metadata
[195] Fetch file
[196] Open Archival Information System
[197] AIP
[198] Pipeline

3.4. Araştırmanın Modeli

Tezin işleyişi ve aynı zamanda modeli, aşağıda gösterilmektedir.



Şekil 80. Tezin Kavramsal Modeli

Şekil 80'de tezin kavramsal modeli görülmektedir. İnternet kaynaklarından elde edilen iki orijinal veri seti (yapılandırılmış tablo verisi ve görüntü) antivirüs kontrolünden geçirildikten sonra Jupyter Lab içerisinde bulunan Notebook platformunda veri ön işleme aşamalarından geçirilmiştir. Bu adımda; yapılandırılmış ham veriye eksik değer ve aykırı değer kontrolü uygulanmış ve veri tipi dönüşümleri gerçekleştirilmiştir. Sonradan kullanılmak üzere ise kırpılmış ve dönüştürülmüş (normalizasyon ve log dönüşümü) versiyonları da oluşturulmuştur. Veri kırpma işlemi ile veri boyutu değiştirme işlemi, görüntüler için de uygulanmıştır. Her iki verinin de ön işleme adımlarının tamamlanmasının ardından, birinci ve ikinci faz uygulamalara geçilmiştir. Birinci faz analizlerde yapılandırılmış tablo verisi, çeşitli denetimli öğrenme algoritmaları ile işlenmiş ve süreç verileri (provenans) JSON formatında kaydedilmiştir. İkinci faz analizlerde ise görüntülere derin öğrenme yöntemlerinden evrişimli sinir ağları uygulanmış ve ardından transfer öğrenmesi yöntemleri ile model doğruluğunun artırılması sağlanmıştır. Bu analizlerde de süreç verileri (provenans) JSON formatında kaydedilmiştir. Ardından, JSON dosyaları BagIt kütüphanesi ile paketlenmiş ve süreç verileri ile orijinal veri setleri,

Archivematica açık kaynak kodlu programına uzun süreli saklama için yüklenmiştir. Burada ise AIP, SIP ve DIP arşiv paketleri oluşturulmuştur.

3.5. Makine Öğrenmesi ve Derin Öğrenme Model Değerlendirme Metrikleri

Çalışma kapsamında oluşturulan tüm modellerin değerlendirilebilmesi için çeşitli değerlendirme metrikleri kullanılmıştır. Bu metrikler arasında sınıflandırma metrikleri (doğruluk, kesinlik, duyarlılık, F1, AUC-ROC eğrisi), regresyon metrikleri (ortalama mutlak hata, ortalama kare hatası, ortalama kare hatası kökü, R^2 vb.), kümeleme metrikleri (Silhouette, Calinski-Harabasz, Davies-Bouldin) ile boyut azaltma metrikleri (EVR, Trustworthiness) bulunmaktadır.

3.5.1. Sınıflandırma Metrikleri

Bu metrikler, bir modelin sınıf etiketlerini ne kadar iyi tahmin ettiğini göstermektedir. Karışıklık matrisi[199], bir sınıflandırma modelinin performansını değerlendirmek için kullanılan tablodur (Vujovic, 2021). Matris, gerçek etiketleri tahmin edilen etiketlerle karşılaştırır, modelin ne kadar iyi performans gösterdiğini ve nerede hata yaptığını gösterir.

Tablo 17. Karmaşıklık Matrisi

		Tahmin Edilen	
Gerçek		Pozitif	Negatif
	Pozitif	**TP**	**FN**
	Negatif	**FP**	**TN**

Tablo 17'de gösterilen karmaşıklık matrisinin elemanları ve açıklamaları şu şekildedir:

- ***TP (True Positive - Doğru Pozitif):*** Modelin pozitif bir örneği olması gerektiği şekilde doğru olarak tahmin ettiğinin göstergesidir.
- ***FN (False Negative - Yanlış Negatif):*** Modelin negatif örneği yanlış bir şekilde tahmin ettiğinin göstergesidir ve Tip II hatası olarak da bilinir.
- ***FP (False Positive - Yanlış Pozitif):*** Modelin pozitif örneği yanlış bir şekilde tahmin ettiğinin göstergesidir ve Tip I hatası olarak da bilinir.
- ***TN (True Negative - Doğru Negatif):*** Modelin negatif örneği olması gerektiği şekilde doğru olarak tahmin ettiğinin göstergesidir.

Bu göstergelerden ise aşağıdaki metrikler türetilmiştir:

[199] Confusion matrix

- ***Doğruluk (Accuracy):*** Tüm veri setindeki doğru tahminlerin sayısının toplam tahmin sayısına bölümü ile elde edilir. Formülü şu şekilde ifade edilmektedir:

$$Doğruluk = \frac{TP + TN}{TP + FN + FP + TN} \tag{2.127}$$

- ***Kesinlik (Precision):*** Modelin pozitif olarak tahmin ettiği örneklerin ne kadarının gerçekten doğru olduğunu ölçer. Formülü şu şekilde ifade edilmektedir:

$$Kesinlik = \frac{TP}{TP + FP} \tag{2.128}$$

- ***Duyarlılık (Recall):*** Gerçek pozitif örneklerin, model tarafından ne kadarının doğru tahmin edildiğini ölçer. Formülü şu şekilde ifade edilmektedir:

$$Duyarlılık = \frac{TP}{TP + FN} \tag{2.129}$$

- ***F1 Skoru:*** Kesinlik (Precision) ve Duyarlılık (Recall) skorlarının harmonik ortalamalarının hesaplandığı metriktir. Özellikle sınıflar arasında bir dengesizlik olduğunda, faydalıdır. Formülü şu şekilde ifade edilmektedir:

$$F1 = 2 \times \frac{Kesinlik \times Duyarlılık}{Kesinlik + Duyarlılık} \tag{2.130}$$

- ***Fβ Skoru:*** F1 skorunun genelleştirilmiş bir versiyonudur. β parametresini kullanarak kesinlik ve duyarlılık skorlarının dengelenmesini sağlar. Formülü şu şekilde ifade edilmektedir:

$$F\beta = (1 + \beta^2) \times \frac{Kesinlik \times Duyarlılık}{\beta^2 \times (Kesinlik + Duyarlılık)} \tag{2.131}$$

β < 1 olduğunda, doğru pozitif tahminler daha fazla göz önünde bulundurulurken, β > 1 olduğunda ise pozitif tahminlere daha fazla önem verilmektedir.

- ***AUC[200] ROC[201] Eğrisi:*** Özellikle ikili sınıflandırma problemlerinde sıklıkla kullanılan bir değerlendirme metriğidir. Modelin iki sınıf arasında ayrım yapma yeteneğini analiz ederek model performansını değerlendirir. Alıcı İşletim Karakteristiği (ROC) eğrisi, bir

[200] Area under the curve
[201] Receiver operating characteristic

modelin pozitif ve negatif sınıflar arasında ayrım yapma yeteneğinin grafiksel bir gösterimidir. Şu formüller bağlamında çizilir:

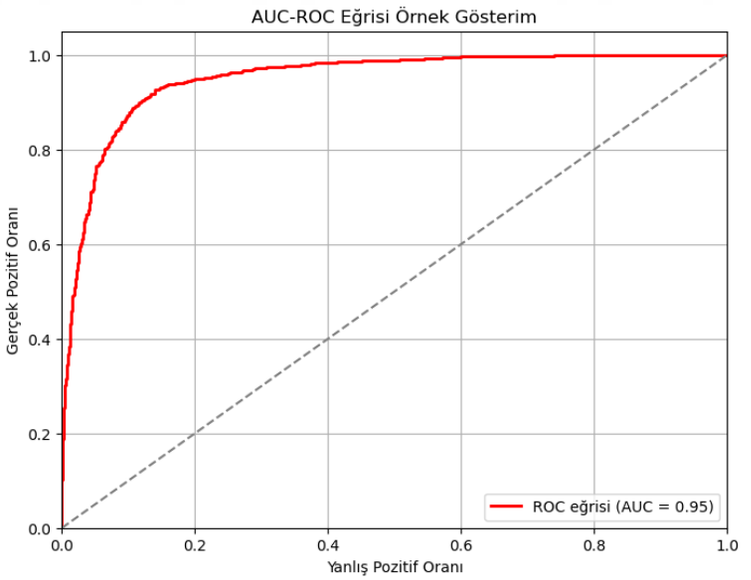
$$TPR = \frac{TP}{TP + FN} \tag{2.132}$$

$$FPR = \frac{FP}{FP + TN} \tag{2.133}$$

Formüllerde TPR (True Positive Rate) gerçek pozitif oranını; FPR (False Positive Rate) ise yanlış pozitif oranını ifade etmektedir. Bu değerlerle çizilen ROC eğrisinin altında kalan alan (AUC), yaklaşık olarak hesaplanabilir. Çünkü doğrudan hesaplanması zor olduğundan, AUC genellikle trapezoidal kuralı kullanılarak yaklaşık olarak hesaplanır (Bradley, 1997). Formülü ise şu şekilde ifade edilmektedir:

$$AUC \approx \sum_{i=1}^{N} (FPR_i - FPR_{i-1}) \times \frac{(TPR_i + TPR_{i-1})}{2} \tag{2.134}$$

Formülde $(FPR_i - FPR_{i-1})$, yanlış pozitif orandaki küçük adım değişimlerini; $\frac{(TPR_i + TPR_{i-1})}{2}$ ise bu küçük aralıktaki ortalama gerçek pozitif oranını ifade etmektedir.



Şekil 81. AUC-ROC Eğrisi

Şekil 81'de örnek bir AUC-ROC eğrisi gösterilmektedir. ROC eğrisi, bir sınıflandırma modelinin farklı eşik ayarlarında performansını gösteren grafiksel bir gösterimdir.

AUC[202] ise, modelin pozitif ve negatif sınıfları ayırt etme konusundaki genel yeteneğini niceliksel olarak belirten bir metriktir. X eksenindeki yanlış potizif oranı, 0 (mükemmel model) ile 1 (en kötü model) değerleri arasında değişirken Y eksenindeki gerçek pozitif oranı da aynı şekilde 0-1 arasında değişen değerler almaktadır. Kırmızı eğri, modelin farklı eşik değerlerinde pozitif ve negatif sınıfları nasıl ayırt ettiğini göstermektedir. Köşegeni gösteren gri kesikli çizgi ise, rastgele bir sınıflandırıcıyı göstermektedir. ROC eğrisi (kırmızı) bu çizgiye yakınsa, model iyi performans göstermiyor demektir. Grafikte sol üst köşe (0,1) ideal noktayı ifade etmektedir (Yang ve Berdine, 2017). Başka bir deyişle modelde hiç yanlış pozitif yoktur (FPR=0) ve tüm pozitifler doğru şekilde tanımlanmıştır (TPR=1).

AUC skor aralığı şu şekilde değerlendirilebilir:

- AUC=1,0 noktasında ise mükemmel sınıflandırıcı (ideal),
- AUC>0,9 ise çok iyi sınıflandırıcı,
- AUC=0,5 ise rastgele tahmin (yararsız model),
- AUC<0,5 ise rastgele tahminden daha kötü (yanlış eğitilmiş model) (Yang ve Berdine, 2017).

Daha yüksek bir AUC skoru, daha iyi sınıflandırma performansı anlamına gelmektedir.

3.5.2. Regresyon Metrikleri

Regresyon metrikleri, bir modelin sürekli değerleri ne kadar iyi tahmin ettiğinin değerlendirilmesini sağlar.

- ***Ortalama Mutlak Hata (Mean Absolute Error):*** Gerçek ve tahmin edilen değerler arasındaki ortalama mutlak farkı ölçer. Tahmin edilen ve orijinal değerler arasındaki ortalama mesafeyi verir. Temelde tahminlerin gerçek değerlerden ortalama ne kadar saptığını gösterir ancak en büyük kısıtı; verilerin az mı yoksa çok mu tahmin edildiği konusunda ve hatanın yönü hakkında hiçbir fikir vermez ve büyük hataları ağır şekilde cezalandırmaz. Formülü şu şekilde gösterilir:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \qquad (2.135)$$

[202] Area under Curve - Eğrinin Altındaki Alan

Formülde n, gözlem sayısını; y_i, gerçek değeri ve $\hat{y}_i$, tahmin edilen değeri ifade etmektedir.

- ***Ortalama Hata Karesi (Mean Squared Error):*** Gerçek ve tahmin edilen değerler arasındaki ortalama kare farkını ölçer. Ortalama mutlak hatadan farkı, tahmin edilen ve gerçek değerler arasındaki karelerin ortalamasının alınmasıdır. Eğimi, ortalama mutlak hataya göre daha kolay hesaplar. Hataların karesini alarak daha büyük hataları, daha küçük hatalardan daha fazla vurgular. Böylece büyük hataları daha fazla cezalandırır. En önemli kısıtı aykırı değerlere karşı hassastır. Formülü şu şekilde gösterilir:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.136}$$

- ***Ortalama Hata Karesinin Karekökü (Root Mean Squared Error):*** MSE'nin karekökünün alınarak orijinal değere geri dönülen metriktir. Tıpkı MSE metriği gibi aykırı değerlere karşı hassastır. Modeldeki büyük hatalara daha fazla ağırlık verir. Regresyon problemlerinde yaygın olarak kullanılır. Formülü şu şekilde gösterilmektedir:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.137}$$

- ***R^2(Belirleme veya Determinasyon Katsayısı):*** Modelin verideki varyansı ne kadar iyi açıkladığını ölçer. Belirleme katsayısı[203] olarak da adlandırılır. Genellikle doğrusal regresyon modelinin performansını değerlendirmede kullanılır. Başka bir deyişle modelin, veri üzerindeki toplam varyansı ne ölçüde açıklayabildiğini ve tahmin ettiği değerlerin gerçek değerlere ne kadar yakın olduğunu gösterir. Formülü şu şekilde gösterilir:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \tag{2.138}$$

Formülde $\bar{y}$, gerçek değerlerin ortalamasını ifade etmektedir. Yüksek R^2 skoru, modelin bağımsız ve bağımlı değişkenler arasındaki ilişkiyi açıklamada oldukça iyi olduğunu gösterirken, düşük bir R^2 skoru ise modelin önemli bilgileri yakalayamadığını gösterir.

[203] Coefficient of Determination

3.5.3. Kümeleme Metrikleri

Kümeleme modellerinin değerlendirilmesinde Silhouette, Calinski-Harabasz ve Davies-Bouldin metrikleri kullanılmaktadır.

- ***Silhouette Skoru:*** Örneklerin ne kadar iyi kümelendiğini ve küme içindeki verilerin diğer kümelere göre benzerliğini ölçer. Silhouette skoru -1 ile 1 arasında değişir. 1 değeri her kümenin diğerlerinden tamamen farklı olduğu; -1 değeri ise tüm verilerin yanlış kümeye atandığı anlamına gelir. 0'a yakın değer, verinin kümelere net olarak ayrılamadığını veya gözlemin kendi kümesine ve diğer kümeye olan uzaklıklarının birbirine yakın olduğunu gösterir. 1'e yakın skorlar ise kümelemenin oldukça iyi yapıldığını göstermektedir (Januzaj, Beqiri ve Luma, 2023). Formülü şu şekilde ifade edilmektedir:

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{2.139}$$

Formülde $a(i)$, i. noktanın aynı küme içindeki ortalama mesafesini; $b(i)$ ise ortalama en yakın kümeye olan ortalama mesafeyi ifade etmektedir.

- ***Calinski-Harabasz İndeksi:*** Kümeleme sonuçlarının kalitesini değerlendirmek için kullanılan bir metriktir. Kümelerin ne kadar iyi ayrılmış ve kompakt olduğunu ölçer. Daha yüksek skor, daha iyi tanımlanmış ve daha iyi ayrılmış kümeleri; daha düşük skor, zayıf kümelenmeyi (üst üste binen veya dağılmış kümeler) ifade eder. Varyans oranı kriteri olarak da bilinir. Ancak skorun üst sınırı bulunmamaktadır, bu da bu metriğin sonucu olduğu gibi yorumlamaktan ziyade farklı k sayılarını değerlendirmek için daha iyi olduğu anlamına gelir. Bir başka kısıtı da küme sayısına duyarlı olmasıdır. Daha fazla küme sayısı da daha yüksek bir skora yol açabilir. Bu nedenle, sonucu doğrulamak için diğer metrikler ile birlikte kullanılması tavsiye edilmektedir (Lima ve Cruz, 2020). Formülü şu şekilde gösterilir:

$$CHI = \frac{T_r(B_k)}{T_r(W_k)} \times \frac{N - k}{k - 1} \tag{2.140}$$

Formülde $T_r(B_k)$, kümeler arası dağılım matrisinin izini (kümelerin birbirinden ne kadar uzakta olduğunu ölçer); $T_r(W_k)$, küme içi dağılım matrisinin izini (her kümenin ne kadar yoğun olduğunu ölçer); N, toplam veri noktası sayısını ve k ise küme sayısını ifade etmektedir. CHI skorunun düşük olması, kümelerin birbirinden net şekilde ayrılamadığını ve iç yoğunluklarının zayıf olduğunu gösterir.

- ***Davies-Bouldin İndeksi:*** Küme sıkılığını ve ayrımını ölçer ve her küme için en benzer diğer kümeyle olan oranı alır ve tüm kümeler için bu değerlerin ortalamasını hesaplar. Kümeler ne kadar birbirinden uzak ve daha az dağılmışsa daha iyi skorlar oluşur. Olabildiğince daha düşük bir puana sahip olmayı hedefler. Puan ne kadar düşükse, her küme o kadar ayrıdır (Wijaya vd., 2021). Formülü şu şekilde gösterilir:

$$DBI = \frac{1}{k}\sum_{i=1}^{k} max_{j \neq i}\left(\frac{\sigma_i + \sigma_j}{d_{ij}}\right) \tag{2.141}$$

  Formülde k, küme sayısını; σ_i, küme i'deki noktalar ile küme merkezi arasındaki ortalama uzaklığı; d_{ij} ise küme merkezleri i ve j arasındaki uzaklığı ifade etmektedir. Özellikle k-ortalamalarda, kümelerin optimum sayısını seçmek için yaygın olarak kullanılmaktadır.

3.5.4. Boyut Azaltma Metrikleri

Kümelemenin aksine boyut azaltma, orijinal bilgileri mümkün olduğunca korurken öznitelik sayısını azaltmayı amaçlar. Bu nedenle, boyut azaltmadaki değerlendirme metriklerinin çoğu bilginin korunmasıyla ilgildir.

- ***Açıklanan Varyans Oranı (Explained Variance Ratio-EVR):*** Her bir temel bileşenin ne kadar varyansı yakaladığını ölçer. Daha yüksek değerler daha fazla bilginin saklandığını gösterir. Formülü şu şekilde ifade edilmektedir:

$$EVR = \frac{\lambda_i}{\sum \lambda_i} \tag{2.142}$$

  Formülde λ_i, özdeğerleri ifade etmektedir.

- ***Güvenilirlik (Trustworthiness):*** Azaltılmış boyutun, orijinal verinin en yakın komşusunu ne kadar iyi koruduğunu ölçer. Temel olarak ölçüm, boyut azaltma tekniğinin orijinal verinin yerel yapısını koruyarak verileri ne kadar iyi koruduğunu görmeye çalışır. Güvenilirlik ölçüm skoru 0 ile 1 arasında değişmektedir. 1'e yakın değerler, azaltılmış boyut veri noktalarına yakın olan komşunun, orijinal boyutta da çoğunlukla yakın olduğu anlamına gelir (Roy, 2024). Özellikle t-SNE[204] ve UMAP[205] gibi doğrusal olmayan boyut azaltma teknikleri için kullanılışlı olup, modelin yerel komşuluk yapılarını koruma

[204] t-distributed Stochastic Neighbor Embedding, yüksek boyutlu verilerin daha düşük boyutlu uzaya indirgenerek görselleştirilmesini sağlayan bir algoritma.
[205] Uniform Manifold Approximation and Projection

yeteneğini değerlendirmek için temel bileşen analizinde de uygulanmaktadır. Formülü şu şekilde ifade edilmektedir:

$$T(k) = 1 - \frac{2}{nk \times (2n - 3k - 1)} \sum_{i=1}^{n} \sum_{j \in U_k(i)} (r(i,j) - k) \quad (2.143)$$

Formülde n, toplam veri noktası sayısını; k, indirgenmiş boyuttaki komşu sayısını; $r(i,j)$, orijinal yüksek boyutlu uzayda j noktasının i noktasına göre uzaklığını; $U_k(i)$, indirgenmiş uzayda i'nin en iyi k komşusu içinde yer alan ancak orijinal uzayda en iyi k komşusu arasında yer almayan noktalar kümesini ifade etmektedir.

$T(k)$ değerinin skor aralıkları şu şekilde özetlenebilir:

- $T(k)$ değerinin 1'e eşit olması durumunda, mükemmel güvenilirlik (tüm komşuluk ilişkilerinin korunması),
- $T(k)$ değerinin 0,8-1,0 aralığında olması durumunda, iyi korunma,
- $T(k)$ değerinin 0,5'ten küçük olması durumunda ise yerel yapının önemli ölçüde bozulması ve kaybı (Roy, 2024).

DÖRDÜNCÜ BÖLÜM

4. BİRİNCİ FAZ (MAKİNE ÖĞRENMESİ) ANALİZLER VE BULGULAR

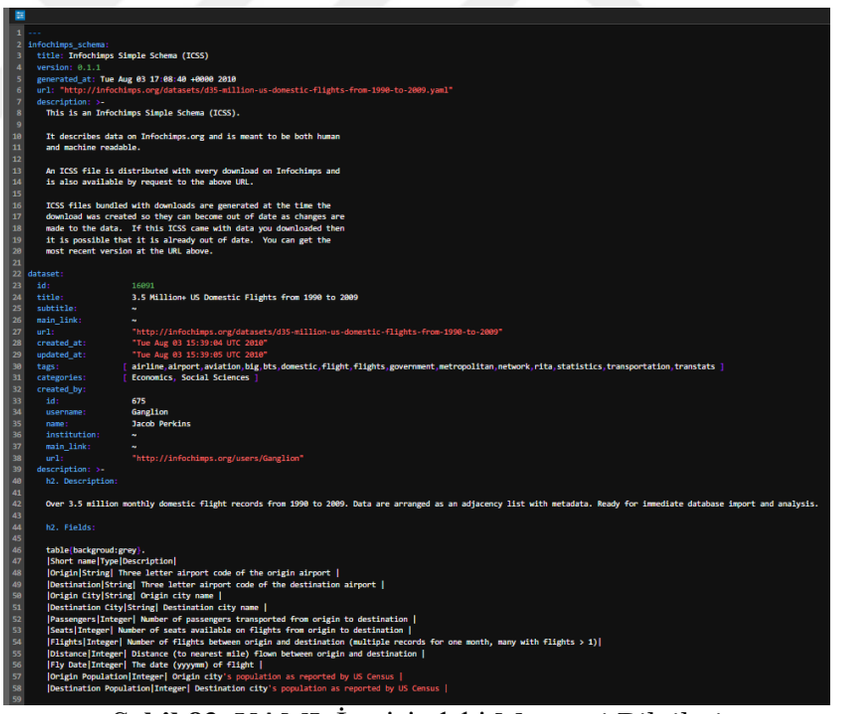
4.1. Veri Ön İşleme

TSV uzantılı veri seti 1, Jupyter Lab içerisindeki Notebook'ta Pandas kütüphanesinin "pd.read_csv()" metodu kullanılarak okunmuştur. Orijinal veri setinin ilk 5 satır görüntüsü Jupyter Lab Notebook üzerinde aşağıdaki gibi görülmektedir.

	MHK	AMW	Manhattan, KS	Ames, IA	21	30	1	254.0	200810	122049	86219
0	EUG	RDM	Eugene, OR	Bend, OR	41	396	22	103.0	199011	284093	76034
1	EUG	RDM	Eugene, OR	Bend, OR	88	342	19	103.0	199012	284093	76034
2	EUG	RDM	Eugene, OR	Bend, OR	11	72	4	103.0	199010	284093	76034
3	MFR	RDM	Medford, OR	Bend, OR	0	18	1	156.0	199002	147300	76034
4	MFR	RDM	Medford, OR	Bend, OR	11	18	1	156.0	199003	147300	76034

Şekil 82. Orijinal Veri Setinin İlk 5 Satır Görüntüsü

Şekil 82'de görüldüğü üzere veri seti 1'de herhangi bir metaveri bulunmamaktadır. Metaveriler, veri setine ait çeşitli bilgiler içeren YAML[206] uzantılı dosya içerisinden alınmıştır. YAML içeriği aşağıdaki gibidir.



Şekil 83. YAML İçerisindeki Metaveri Bilgileri

Metaveriler, 0. satır bilgisine işlenmiştir.

[206] YAML Ain't Markup Language

	Origin	Destination	Origin City	Destination City	Passengers	Seats	Flights	Distance	Fly Date	Origin Population	Destination Population
0	EUG	RDM	Eugene, OR	Bend, OR	41	396	22	103.0	199011	284093	76034
1	EUG	RDM	Eugene, OR	Bend, OR	88	342	19	103.0	199012	284093	76034
2	EUG	RDM	Eugene, OR	Bend, OR	11	72	4	103.0	199010	284093	76034
3	MFR	RDM	Medford, OR	Bend, OR	0	18	1	156.0	199002	147300	76034
4	MFR	RDM	Medford, OR	Bend, OR	11	18	1	156.0	199003	147300	76034

Şekil 84. Metaveri Ekleme 1

0. satır bilgileri ise dataframe (df) şeklinde oluşturulan yeni_satir değişkenine kaydedilmiş ve ardından "pd.concat" metodu ile veri setine birleştirilmiştir. Son görünüm aşağıdaki gibidir.

	Origin	Destination	Origin City	Destination City	Passengers	Seats	Flights	Distance	Fly Date	Origin Population	Destination Population
0	MHK	AMW	Manhattan, KS	Ames, IA	21	30	1	254.0	200810	122049	86219
1	EUG	RDM	Eugene, OR	Bend, OR	41	396	22	103.0	199011	284093	76034
2	EUG	RDM	Eugene, OR	Bend, OR	88	342	19	103.0	199012	284093	76034
3	EUG	RDM	Eugene, OR	Bend, OR	11	72	4	103.0	199010	284093	76034
4	MFR	RDM	Medford, OR	Bend, OR	0	18	1	156.0	199002	147300	76034

Şekil 85. Metaveri Ekleme 2

Metaveri adları veri setinin karakteristiğine uygun bir şekilde Türkçe'ye çevrilmiştir.

	kalkis_havaalani_kodu	varis_havaalani_kodu	kalkis_sehri	varis_sehri	yolcu_sayisi	koltuk_sayisi	ucus_sayisi	mesafe_mil	ucus_tarihi	kalkis_sehri_nufusu	varis_sehri_nufusu
0	MHK	AMW	Manhattan, KS	Ames, IA	21	30	1	254.0	200810	122049	86219
1	EUG	RDM	Eugene, OR	Bend, OR	41	396	22	103.0	199011	284093	76034
2	EUG	RDM	Eugene, OR	Bend, OR	88	342	19	103.0	199012	284093	76034
3	EUG	RDM	Eugene, OR	Bend, OR	11	72	4	103.0	199010	284093	76034
4	MFR	RDM	Medford, OR	Bend, OR	0	18	1	156.0	199002	147300	76034

Şekil 86. Metaveri Adlarının Türkçe'ye Çevrilmiş Hali

"kalkis_sehri" ile "varis_sehri" değişkenleri, hem şehir hem de eyalet bilgisini içerdiğinden bu iki sütun, Pandas kütüphanesinde "str.split()" fonksiyonu kullanılarak şehirleri ve eyaletlerini ayrı ayrı sütunlarda gösterecek şekilde ayrılmıştır. Ayrıca "uçuş_tarihi" değişkeni de "YYYYMM" formatını izlediği için bu bilgi de ay ve yıl olarak iki ayrı sütunda gösterilmiştir. Mesafe sütunu mil cinsinden olduğu için, değerler km'ye dönüştürülmüş ve ondalık değere en yakın tam sayıya çevrilmiştir. Tüm metaverilerin birbirleriyle ilişki sıralanmış son hali ise aşağıdaki gibidir.

	kalkis_havaalani_kodu	kalkis_sehri	kalkis_eyaleti	kalkis_sehri_nufusu	varis_havaalani_kodu	varis_sehri	varis_eyaleti	varis_sehri_nufusu	yolcu_sayisi	koltuk_sayisi	ucus_sayisi	mesafe_km	ucus_tarihi(ay)	ucus_tarihi(yil)
0	MHK	Manhattan	KS	122049	AMW	Ames	IA	86219	21	30	1	409	10	2008
1	EUG	Eugene	OR	284093	RDM	Bend	OR	76034	41	396	22	166	11	1990
2	EUG	Eugene	OR	284093	RDM	Bend	OR	76034	88	342	19	166	12	1990
3	EUG	Eugene	OR	284093	RDM	Bend	OR	76034	11	72	4	166	10	1990
4	MFR	Medford	OR	147300	RDM	Bend	OR	76034	0	18	1	252	2	1990

Şekil 87. Veri Seti 1'in Son Görünümü

Veri seti 1'e ait temel istatistiki değerler ise aşağıda gösterilmiştir.

	count	mean	std	min	25%	50%	75%	max
kalkis_sehri_nufusu	3606803.0	5.871502e+06	7.858062e+06	13005.0	1030597.0	2400193.0	8613622.0	38139592.0
varis_sehri_nufusu	3606803.0	5.897982e+06	7.906127e+06	12887.0	1025470.0	2400193.0	8635706.0	38139592.0
yolcu_sayisi	3606803.0	2.688910e+03	4.347617e+03	0.0	109.0	1118.0	3503.0	89597.0
koltuk_sayisi	3606803.0	4.048297e+03	6.200871e+03	0.0	156.0	1998.0	5370.0	147062.0
ucus_sayisi	3606803.0	3.722890e+01	4.961970e+01	0.0	2.0	25.0	55.0	1128.0
mesafe_km	3606803.0	1.122723e+03	9.727222e+02	0.0	440.0	836.0	1492.0	8200.0
ucus_tarihi(ay)	3606803.0	6.544309e+00	3.467370e+00	1.0	4.0	7.0	10.0	12.0
ucus_tarihi(yil)	3606803.0	2.000568e+03	5.746467e+00	1990.0	1996.0	2001.0	2006.0	2009.0

Şekil 88. Veri Seti 1 Temel İstatistiki Bilgiler

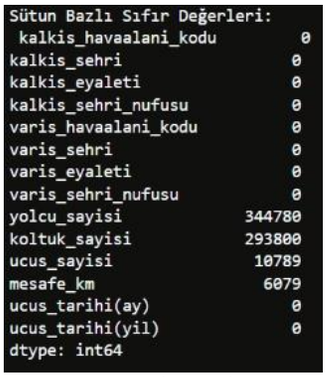
Son işlemlerden sonra veri seti 1'de bulunan veri tipleri, Pandas kütüphanesinde bulunan "astype()" fonksiyonu ile "object"-"string" ve "float64"-"int64" dönüşümleri de yapılarak, son hali "df.info()" komutu ile listelenmiştir.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3606803 entries, 0 to 3606802
Data columns (total 14 columns):
 #   Column                Dtype
---  ------                -----
 0   kalkis_havaalani_kodu  string
 1   kalkis_sehri           string
 2   kalkis_eyaleti         string
 3   kalkis_sehri_nufusu    int64
 4   varis_havaalani_kodu   string
 5   varis_sehri            string
 6   varis_eyaleti          string
 7   varis_sehri_nufusu     int64
 8   yolcu_sayisi           int64
 9   koltuk_sayisi          int64
 10  ucus_sayisi            int64
 11  mesafe_km              int64
 12  ucus_tarihi(ay)        int64
 13  ucus_tarihi(yil)       int64
dtypes: int64(8), string(6)
memory usage: 385.2 MB
```

Şekil 89. Veri Seti 1'de Bulunan Veri Tipleri

Şekil 89'da da görüldüğü üzere, toplam 14 sütun ve 3.606.803 satırlık veri, 6 adet string ve 8 adet int64 tipinde veri içermektedir.

Veri seti 1'deki değerler içerisinde boş (null), mevcut değil (na), yinelenen (duplicated) ve 0 değer olup olmadığının kontrolleri sırasıya "isnull().any()", "isna().any()", "duplicated().any()" ve "(df==0).sum()" komutlarıyla yapılmıştır. Veri setinde herhangi bir "boş" veya "mevcut değil" değer bulunmamış, fakat 41.753 yinelenen değer tespit edilmiştir. Yinelenen değerlerin tamamı "drop_duplicates(inplace=True)" komutu ile veri setinden silinmiştir. Yinelenen değerler silindikten sonra geriye toplam 3.565.050 satır veri kalmıştır. Veri seti 1'de bulunan 0 değerleri ise sütun bazlı olarak aşağıdaki şekilde gösterilmektedir.



Şekil 90. Veri Seti 1'de Bulunan 0 Değerleri

En fazla 0 değeri 344.780 kayıt ile yolcu sayısı sütununda bulunmaktadır. İkinci sırada 293.800 ile koltuk sayısı gelmektedir. Uçuş sayısı sütununda ise 10.789 adet 0 değeri bulunmaktadır. En az 0 değeri olan sütun, 6079 ile mesafe değişkenidir. Yolcu sayısının 0 olması, uçuşların yolcular olmadan yapıldığı anlamına gelebilir. Koltuk sayısının da 0 olması ise, ya yolcu adına kayıtlı koltuğun olmaması (yolcusuz uçuş) ya da veri hatası olarak yorumlanabilir. Uçuş sayısındaki 0 değerleri ise kayıtlı uçuşların herhangi bir sebepten dolayı (örneğin; hava durumu veya motor arızası) yapılamadığını gösterebilir. Mesafe değişkenindeki 0'lar ise yine uçuşların yapılmadığının veya veri hatasının bir göstergesi olabilir.

Makine öğrenmesi algoritmaları, çeşitli matematiksel işlemler içerdiğinden dolayı ve 0 değerlerinin temel işlemler ve ileri seviye hesaplamalarda tanımsız değerlere yol açması, bu değerlerin dönüştürülmesi veya silinmesi ile ilgili olarak aksiyon alınmasını gerekli kılmaktadır. Veri seti 1 uçuş verilerini içerdiğinden dolayı herhangi bir değerinin 0 olması normalde beklenmedik bir durum olarak değerlendirilebilir. Çalışmada bu yaklaşım benimsenmiş ve 0 değeri olan hücrelerin silinmesi yerine, ilgili sütunlardaki veri dağılım örüntüsüne uygun bir biçimde değiştirilmesine karar verilmiştir. Çeşitli metrikler (mod, medyan, ortalama vb.) bazında uygulamaların yapılabilmesi için verinin öncelikle normal dağılıma benzer bir örüntüde olması gerekmektedir. Bu tür uygulamaların gerçekleştirilebilmesi için de öncelikle veri seti 1'in normal dağılıp dağılmadığının testi Kolmogorov-Smirnov metodu ile yapılmıştır. Bu metod, bir örneğin tamamen belirlenmiş sürekli dağılıma sahip bir popülasyondan gelip gelmediğine karar vermek için kullanılır (Drezner, Turel ve Zerom, 2010).

Tablo 18. Veri Seti 1 Kolmogorov-Smirnov Test Sonuçları

Sütun	K-S	p	Dağılım
kalkis_sehri_nufusu	1	0	Normal Değil
varis_sehri_nufusu	1	0	Normal Değil
yolcu_sayisi	0,90	0	Normal Değil
koltuk_sayisi	0,91	0	Normal Değil
ucus_sayisi	0,83	0	Normal Değil

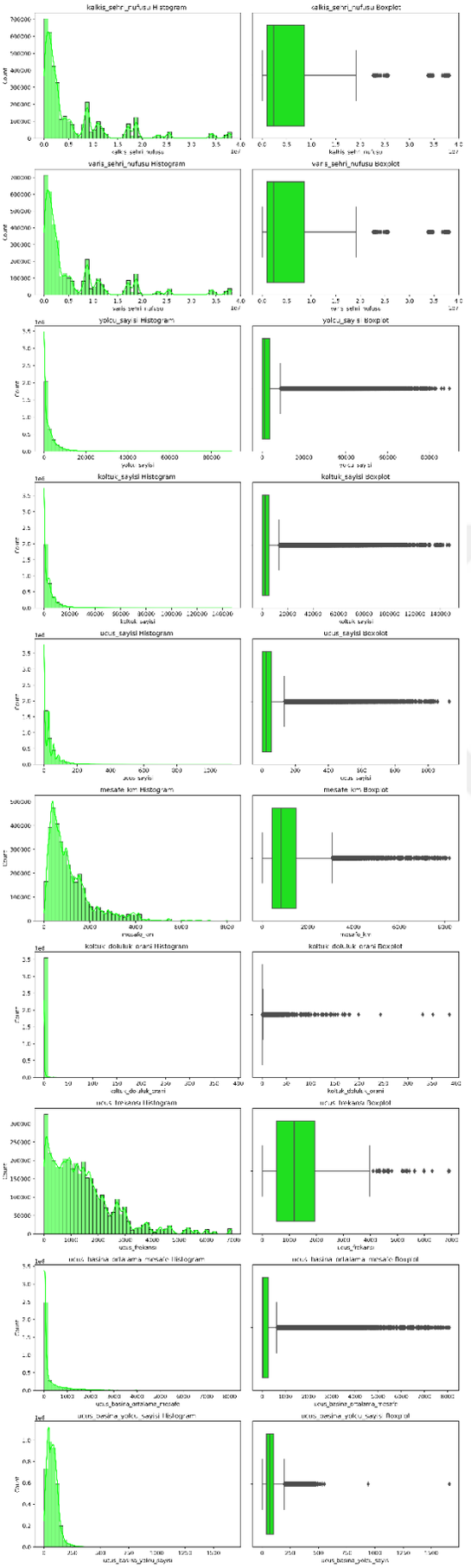
mesafe_km	0,99	0	Normal Değil
ucus_tarihi(ay)	0,89	0	Normal Değil
ucus_tarihi(yil)	1	0	Normal Değil

Tablo 18'de görülen test sonuçlarına göre veri seti 1'deki nümerik değerlerin normal dağılmadığı görülmüştür. Bu durumda 0 içeren hücrelerin mod, medyan veya ortalama ile doldurulması yanlış bir uygulama olacaktır. Dolayısıyla, bu aşamada literatürde de sıklıkla kullanılan yöntemlerden KNN Impute tekniği kullanılarak ilgili sütunlarda bulunan 0 değerleri doldurulmuştur. Bu teknik, çoğunlukla Öklid mesafesine dayalı olarak ve en benzer k sayıda komşu bularak eksik değerleri tahmin etmek için kullanılan parametrik olmayan bir yöntemdir. Yöntem, benzer özelliklere sahip örneklerin eksik özellik için benzer değerlere sahip olma olasılığının yüksek olduğunu varsaymaktadır (Beretta ve Santaniello, 2016). Bu kapsamda, "yolcu_sayisi", "koltuk_sayisi", "ucus_sayisi" ve "mesafe_km" özelliklerinde bulunan 0 değerleri, k=4 değeri seçilerek doldurulmuştur. Bu işlemden sonra ise öznitelik mühendisliği ile yeni öznitelikler oluştrulmuştur.

Tablo 19. Öznitelik Mühendisliği ile Türetilen Yeni Öznitelikler

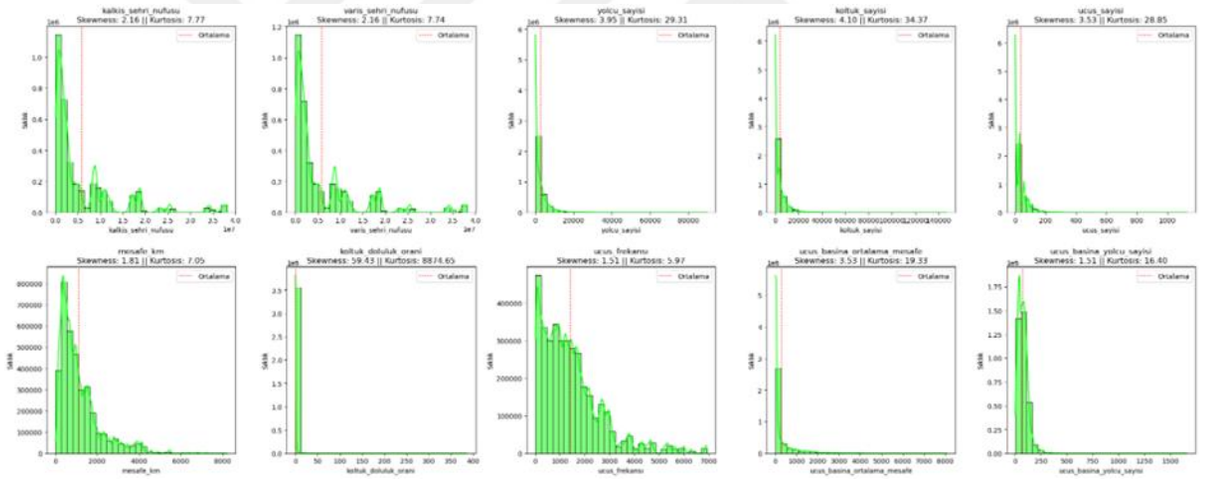
Yeni Türetilen Öznitelik	**Hangi Özniteliklerden Türetildiği**
uçuş başına yolcu sayısı	yolcu sayısı / uçuş sayısı
koltuk doluluk oranı	yolcu sayısı / koltuk sayısı
uçuş başına ortalama mesafe	mesafe / uçuş sayısı
rota eyaletler arası	kalkış eyaleti + varış eyaleti
rota şehirlerarası	kalkış sehri + varış sehri
uçuş frekansı	rota şehirlerarası

Tablo 19'dan görüldüğü üzere uçuş frekansı özniteliği, "rota şehirlerarası" özniteliğinden alınarak türetilmiş ve value_counts().reset_index() komutu kullanılarak her bir uçuş rotasının ne sıklıkla gerçekleştiği hesaplanmıştır. Diğer öznitelikler ise sayısal değerler için bölme, string veriler için toplama ve birleştirme işlemleri yoluyla türetilmiştir. Mevcut öznitelikler ile yeni türetilen özniteliklerin normal dağılıma uyup uymadıklarının bir diğer tespiti ise histogram ve kutu grafiği kullanılarak yapılmıştır.



Şekil 91. Veri Seti 1 Histogram ve Kutu Grafiği

Şekil 91'de veri seti 1'e ait tüm sayısal özniteliklerin ait histogram ve kutu grafikleri gösterilmektedir. Her iki tür grafikten de hemen hemen tüm özniteliklerin yüksek sayıda aykırı değer içerdikleri görülebilmektedir. Kalkış şehri nüfusu ile varış şehri nüfusu, sağa çarpıktır. Başka bir deyişle, veri seti 1'de bulunan çoğu şehir az sayıda nüfusa sahiptir ancak büyük şehirlerin de varlığı göz önüne alındığında gerçek dünya verisi için normal bir sonuç ortaya çıkmaktadır. Yolcu sayısı ve koltuk sayısı öznitelikleri de aynı şekilde fazla sayıda aykırı değer içermektedir. Her iki özellik de sağa çarpıktır. Uçuş sayısı özniteliği sağa çarpıktır ve çok sayıda yüksek frekanslı uçuş kaydının bulunduğunu göstermektedir. Mesafe özelliği ise çoğu uçuşun kısa ve orta mesafelerde olduğunu göstermektedir. Koltuk doluluk oranı özelliği de, türetilmesine rağmen çok sayıda düşük değere sahip olup, sağa doğru çarpıktır. Uçuş frekansı da aynı şekilde çok sayıda düşük frekanslı rotanın olduğunu göstermektedir. Uçuş başına ortalama mesafe özniteliği, hem mesafe hem de uçuş sayısının özelliklerini yansıtmaktadır. Uçuş başına yolcu sayısı özniteliği ise alt sınıra yakın bir dağılım göstermekte ve nispeten az sayıda aykırı değer içermektedir.



Şekil 92. Veri Seti 1 Çarpıklık ve Basıklık Grafiği

Şekil 92'de ise veri seti 1'in tüm sayısal özniteliklerine ait çarpıklık[207] ve basıklık[208] grafikleri gösterilmektedir. Çarpıklık, bir veri setinde özniteliklerin sahip olduğu değerlerin ortalaması etrafındaki dağılımındaki asimetri derecesini ifade eden bir terimdir (Kim, 2013). Makine öğrenmesi uygulamalarında çarpıklık, özniteliklerin altta yatan dağılımını anlamak ve uygun veri ön işlemeyi sağlamak için oldukça önemlidir. Çarpıklık değeri 0 ise dağılım simetriktir. Normal dağılım simetrik bir dağılım olduğundan çarpıklığı da sıfırdır. Eğer çarpıklık sağ tarafa doğru ise, kuyruk daha yüksek değerlere doğru kayar ve veri noktalarının çoğu sol tarafta

[207] Skewness
[208] Kurtosis

(düşük değerlerde) yoğunlaşır. Eğri sol tarafa doğru ise, kuyruk daha düşük değerlere doğru çekilir ve veri noktalarının çoğu sağ tarafta kümelenir (Chiang, 2016).

Basıklık ise bir veri setindeki özelliklerin olasılık dağılımına göre kuyruk durumunu ölçen bir metriktir. Çarpıklık asimetriyi tanımlarken, basıkılık ise kuyrukların normal dağılımdan ne kadar farklı olduğunu tanımlar. Normal bir dağılımın basıklık değeri 3 olarak kabul edilmektedir. Yüksek basıklık[209] değeri (>3), ağır kuyruklar ve keskin bir tepe noktasını gösterirken düşük basıklık[210] (<3) değeri, hafif kuyrukları ve daha düz bir tepe noktasını ifade eder (Berkane ve Bentler, 1987).

Şekil incelendiğinde neredeyse tüm değişkenlerin sağ tarafa doğru çarpık olduğu görülebilir. koltuk_doluluk_orani (çarpıklık=59,43, basıklık=8874,65), yolcu_sayisi (çarpıklık=9,35, basıklık=29,31) ve koltuk_sayisi (çarpıklık=4,10, basıklık=34,37) değişkenleri yüksek çarpıklık ve basıklık skorlarına sahiptir. ucus_sayisi, ucus_frekansı, mesafe_km öznitelikleri ise sağa doğru orta derecede çarpıklık göstermektedir. Değişkenlerin hepsinin az veya çok basıklık özelliği gösterdiği söylenebilir. Özellikle koltuk_doluluk_orani (8874,65) değişkeni aşırı basıklık göstermektedir. Uçuş frekansı (basıklık=5,97) da diğer değişkenler gibi yüksek basıklık özelliğine sahiptir.

Sınıflandırma ve regresyon görevlerinde güçlü modeller geliştirebilmek için aykırı değer probleminin çözülmesi gerekmektedir. Gereğinden çok sayıda aykırı değer, modelin genelleme özelliğinin oldukça kaybolmasına ve zayıf temsile neden olmaktadır. Problemin çözümü için öncelikle IQR[211] tekniği uygulanmıştır. Bu teknik, veri setindeki aykırı değerleri tespit etmek ve işlemek için kullanılan ve istatistiksel olarak oldukça güçlü bir tekniktir. Değerlerin orta %50'sinin yayılımına dayanır ve üçüncü çeyrek (Q3) ile birinci çeyrek (Q1) arasındaki fark olarak hesaplanır. Veri noktaları; genellikle Q1-(1.5xIQR) altında veya Q3+(1.5xIQR) üstünde kalırsa, aykırı değer olarak kabul edilmektedir (McGill, Tukey ve Larsen, 1978).

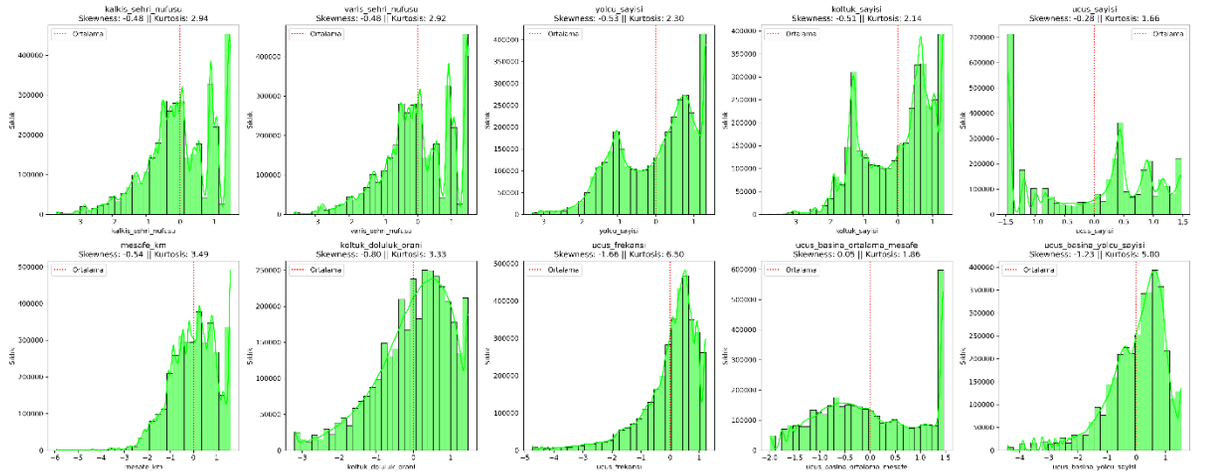
IQR tekniği uygulandıktan sonra logaritmik dönüşüm işlemi gerçekleştirilmiştir. Böylece uç ve dip değerlerin etkisinin azaltılması ve çarpıklığın azaltılarak normalliğin artırılması hedeflenmiştir. Osborne'a (2010) göre, log ve log1p gibi logaritmik dönüşümler, çarpıklığı azaltmak ve normal dağılım göstermeyen verileri normal dağılıma yaklaştırmak için yaygın olarak kullanılmaktadır.

[209] Leptokurtik
[210] Platykurtik
[211] Çeyrekler açıklığı

Logaritmik dönüşüm işlemlerinden sonra ise kırpma[212] işlemi uygulanmıştır. İşlem öznitelikleri, önceden tanımlanmış belirli eşik değerleri içinde tutarak uç (aykırı) değerlerin model üzerindeki etkisini azaltmak için kullanılan bir veri ön işleme tekniğidir. Genellikle normalizasyon veya standardizasyon adımlarından önce kullanılmaktadır. Teknik, faydalı olabilecek bilgileri içeren örneklerin çıkarılması yerine, seçilen belirli aralığın dışına düşen örnekleri sınır değerlerine indirger (Sharma ve Chatterjee, 2021; Alexandropoulos, Kotsiantis ve Vrahatis, 2019). Uygulamada alt ve uç değerler, belirlenen 1. ve 99. yüzdelik değerler ile değiştirilmiştir. Bu yöntem, veri setinin hem boyutunu korumuş hem de uç ve alt değerlerin model eğitimi üzerindeki olası olumsuz etkisini azaltmıştır. Böylece modelin kararlılığı artırılmıştır.

Son olarak verilerde standardizayonu sağlayabilmek için Scikit-Learn kütüphanesindeki "StandardScaler" fonksiyonu uygulanmıştır. Bu teknik, özelliklere ait ortalamayı her bir veri noktasından çıkarıp standart sapmaya böldükten sonra, verilerin ortalamasının 0 ve standart sapmasının 1 olmasıyla sonuçlanır. Standardizasyon olmadan, daha geniş sayısal aralıklara sahip özellikler daha fazla önyargılı model performansına veya yavaş yakınsamaya yol açabilir (Ahsan vd., 2021).

Tüm dönüşüm işlemlerinden sonra histogram ve kutu grafiği yaklaşımları ile aykırı değer ve normallik kontrolleri tekrardan yapılmıştır.

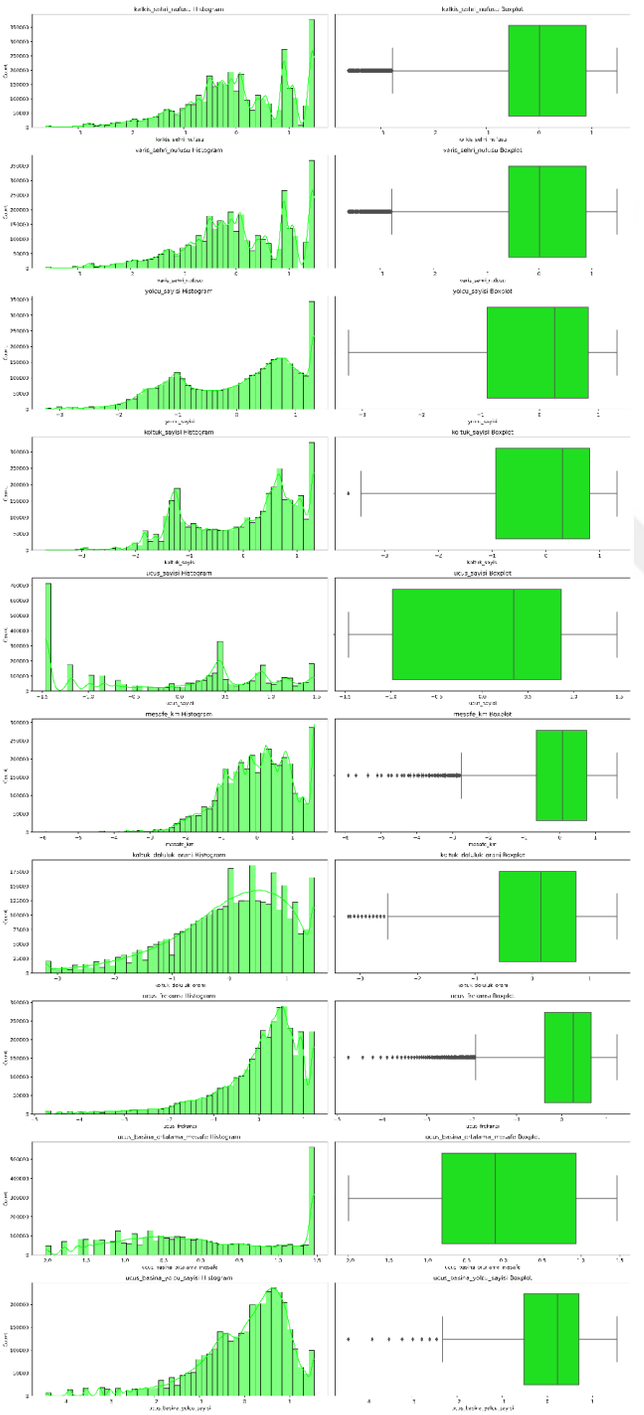


Şekil 93. Veri Seti 1 Çarpıklık ve Basıklık Grafiği (Kontrol)

Şekil 93'ten de görüldüğü üzere IQR tabanlı aykırı değer azaltma, logaritmik dönüşüm, kırpma ve standardizasyon uygulandıktan sonra sayısal değişkenlerin dağılımlarında hem çarpıklık hem de basıklık olarak önemli iyileştirmeler gerçekleşmiştir. kalkis_sehri_nufusu,

[212] Clipping

varis_sehri_nufusu, yolcu_sayisi, koltuk_sayisi, ucus_sayisi ve koltuk_doluluk_orani gibi çoğu değişken -0,8 ile 0,28 arasında çarpıklık ve 1,66-3,33 arası basıklık skorlarına çekilmiştir. Bu durum, bu değişkenlerin yaklaşık olarak normal dağıldığını göstermektedir. Özellikle, ucus_basina_ortalama_mesafe değişkeni neredeyse mükemmel bir simetri skoruna (çarpıklık≈0,05) ulaşmış ve mesafe_km ve koltuk_doluluk_orani kabul edilebilir basıklık durumuna yaklaşmıştır. Bazı değişkenlerde (ucus_frekansı ve ucus_basina_yolcu_sayisi) yüksek basıklık değerleri olmasına rağmen çarpıklık skorları oldukça iyi durumdadır.

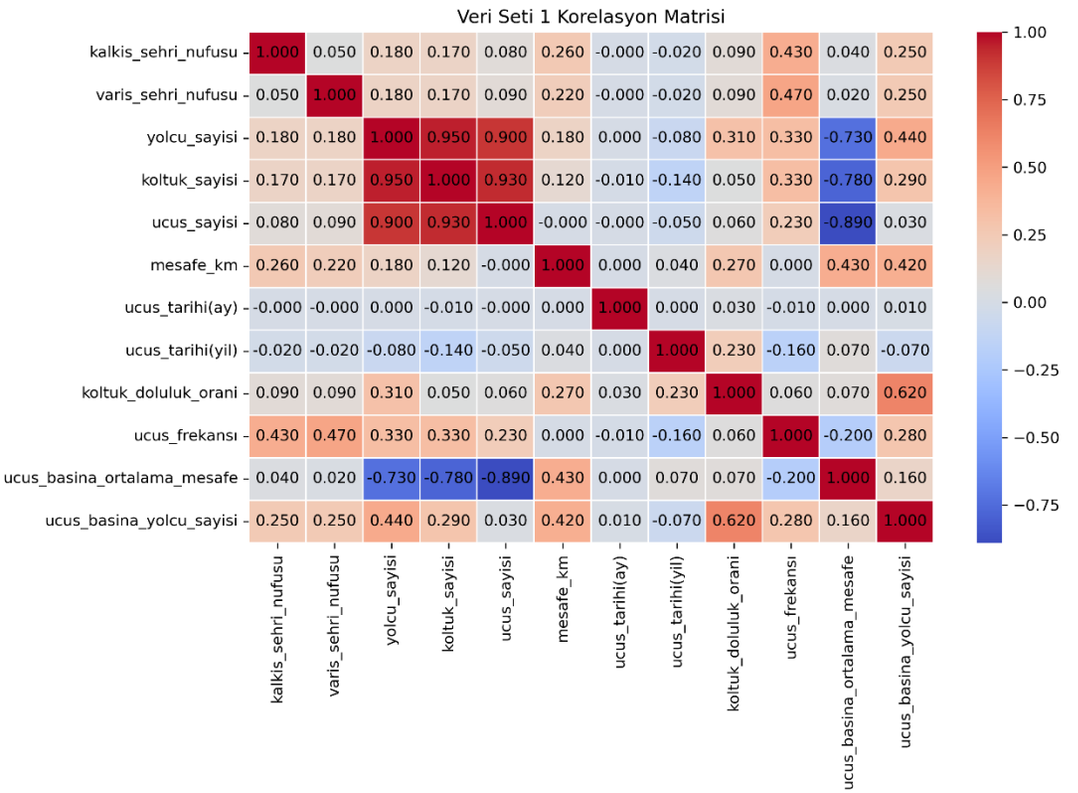


Şekil 94. Veri Seti 1 Histogram ve Kutu Grafiği (Kontrol)

Şekil 94'te ise aykırı değerlerin orijinal verilere kıyasla önemli ölçüde azaldığı görülmektedir. Verilerin bağlı olduğu aralıklar, standardizasyon ile uyumlu bir şekilde merkezileştirilmiştir. Veri seti 1'de önemli ölçüde azaltıldığı görülen aykırı değerlerin tamamının yok edilememesi, gerçek dünya verilerinin neredeyse hiç normal dağılıma sahip olmadığını kanıtlamaktadır. Az sayıdaki aykırı değerin veri setinde tutulması, modellerin gerçek dünya koşullarındaki gibi genelleştirilmesine katkısının olacağı düşünülmüştür.

4.2. Temel İstatistiki Göstergeler

Veri seti 1 için ilk tanımlayıcı istatistik korelasyon matrisidir.



Şekil 95. Veri Seti 1 Korelasyon Matrisi

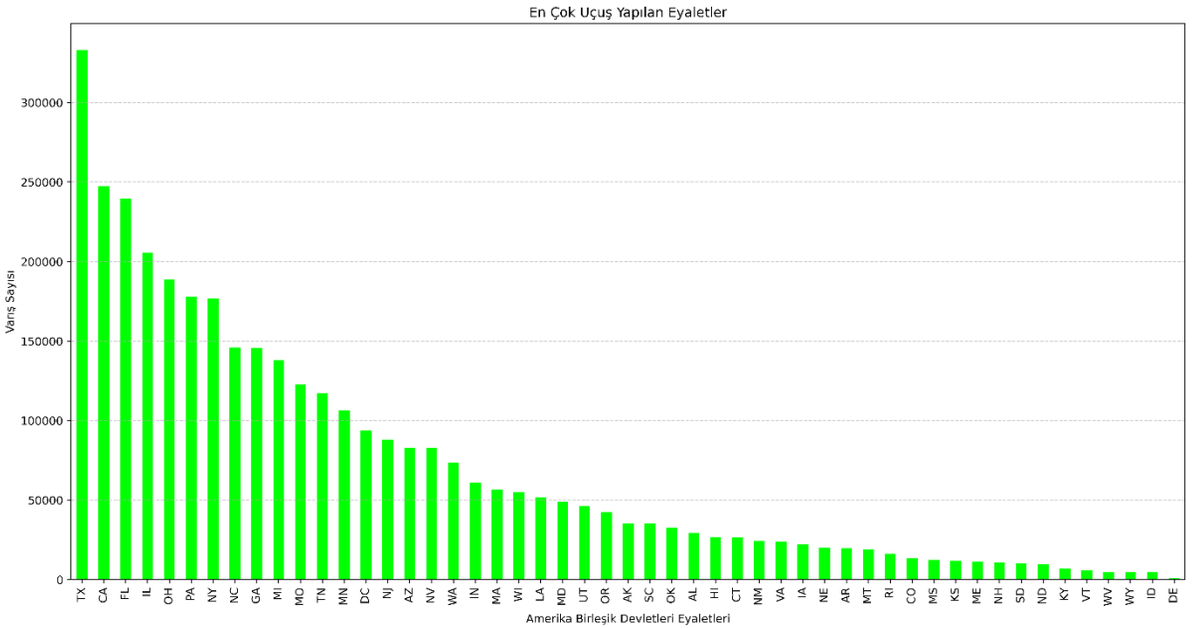
Şekil 95'te veri seti 1 değişkenlerine ait korelasyon matrisi görülmektedir. Korelasyon matrisi, değişkenler arasındaki korelasyon katsayılarını gösteren bir tablo olup ve keşifsel veri analizlerinde (EDA[213]) sıklıkla kullanılmaktadır. Makine öğrenmesinde ise öznitelik seçiminin dikkatli bir şekilde yapılmasında önemli oynamaktadır (Kumar ve Chong, 2018). Alabileceği değerler -1 ile 1 arasında değişmektedir. -1'e yakın değer; değişkenler arasında ters orantılı bir korelasyon olduğunu gösterirken, +1'e daha yakın değerler, değişkenler arasında doğru orantılı bir koreleasyon olduğunu gösterir. Ayrıca 0'a yakın değerler, doğrusal ilişkinin olmadığını veya zayıf ilişkiyi gösterir (Chan, 2003). Şekil incelendiğinde, yolcu_sayisi ve koltuk_sayisi

[213] Exploratory data analysis

(r=0,95) arasında oldukça yüksek bir pozitif korelasyon görülmektedir. Bunun anlamı ise daha fazla koltuğa sahip uçuşların doğal olarak daha fazla yolcu taşımasıdır. Ancak, böyle bir durumun özellikle regresyon modellerinde çoklu doğrusallığa[214] yol açma ihtimali bulunmakta ve tahmini modellemede, varyansın yükselmesi gibi sonuçlara yol açabilmektedir. Çözüm olarak ise temel bileşen analizi ile boyut azaltma veya düzenlileştirme teknikleri kullanılabilir. Bir başka yüksek pozitif korelasyon, koltuk_sayisi ve ucus_sayisi değişkenlerinde görülebilir (r=0,93). Bu durum ise daha fazla uçuşa sahip rotaların, daha yüksek koltuk kapasitesine sahip olma eğilimini açıklamaktadır. yolcu_sayisi ve ucus_sayisi değişkenleri arasında ise yine yüksek bir korelasyon görülmektedir (r=0,90). Doğal olarak, daha fazla uçuş daha fazla yolcunun taşınmasını gerektirecektir. Pozitif korelasyonların yanı sıra negatif korelasyon gösteren değişkenler de bulunmaktadır. Bunlardan biri ucus_basina_ortalama_mesafe ve ucus_sayisi değişkenleridir. -0,89'luk negatif korelasyon, uzun mesafeli uçuşların daha az sayıda olma eğilimini göstermektedir. ucus_basina_ortalama_mesafe ve koltuk_sayisi değişkenleri ise yine yüksek bir negatif korelasyona sahiptir (r=-0,78). Bu durumun anlamı ise daha büyük uçaklara -yani daha fazla koltuğa- sahip uçuşların, daha kısa ortalama mesafeli uçuş eğilimini göstermektedir. ucus_basina_ortalama_mesafe ve yolcu_sayisi değişkenleri de negatif korelasyona sahip görünmektedir (r=-0,73). Yani, toplam yolcu hacmi, genellikle daha kısa uçuşlarda daha yüksek olma eğilimindedir. koltuk_doluluk_orani ve ucus_basina_yolcu_sayisi değişkenleri ise orta seviyede pozitif korelasyona sahiptir (r=0,62). Bu durum da daha yüksek koltuk sayısının, uçuş başına daha fazla yolcu ile ilişkilendirilebileği anlamına gelebilir. ucus_frekansı değişkeni, kalkis_sehri_nufusu (r=0,43) ve varis_sehri_nufusu (r=0,47) değişkenleri ile orta düzeyde bir pozitif korelasyon göstermektedir. Bu, daha büyük nüfusa sahip şehirlerin daha sık uçuşlara sahip olma eğiliminde olduğunu göstermektedir. koltuk_doluluk_orani değişkeni de yolcu_sayisi (r=0,31) ve mesafe_km (r=0,27) değişkenleri ile orta seviyede pozitif korelasyon göstermektedir. Matriste birbiriyle hemen hemen hiç korelasyon göstermeyen değişkenler de bulunmaktadır.

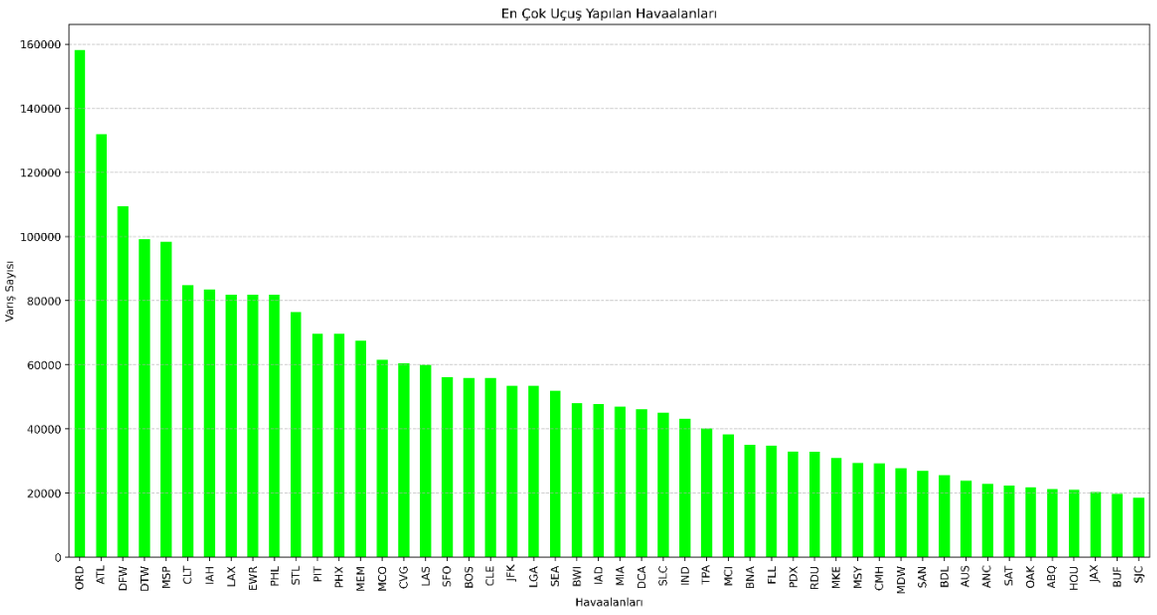
Veri setinin daha iyi anlaşılabilmesi için çeşitli tanımlayıcı istatistiklerin yanında özniteliklerin çeşitli istatistiki özellikleri grafiksel olarak gösterilmiştir.

[214] Multicollinearity



Şekil 96. Eyalet Bazlı En Çok Uçuş Yapılan Varış Noktaları

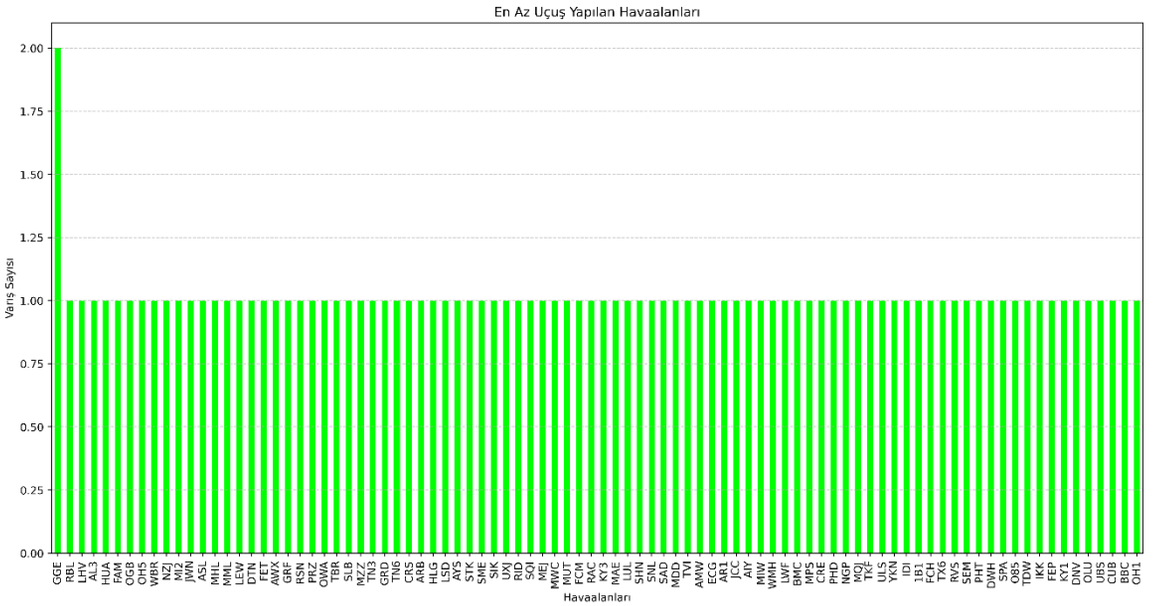
Şekil 96'da en çok uçuş yapılan eyaletlerin sıralaması gösterilmektedir. Texas eyaleti 333.020 uçuş ile ilk sırada yer alırken California eyaleti 247.512 uçuş ile ikinci sırada yer almaktadır. Üçüncü sırada ise 239.828 uçuşla Florida eyaleti bulunmaktadır. Delaware eyaleti 634 uçuş ile son sırada yer alırken Idaho eyaleti 4540 uçuş ve Wyoming eyaleti de 4765 uçuş ile son sıraları paylaşmaktadır.



Şekil 97. En Çok Uçuş Yapılan Varış Havaalanları

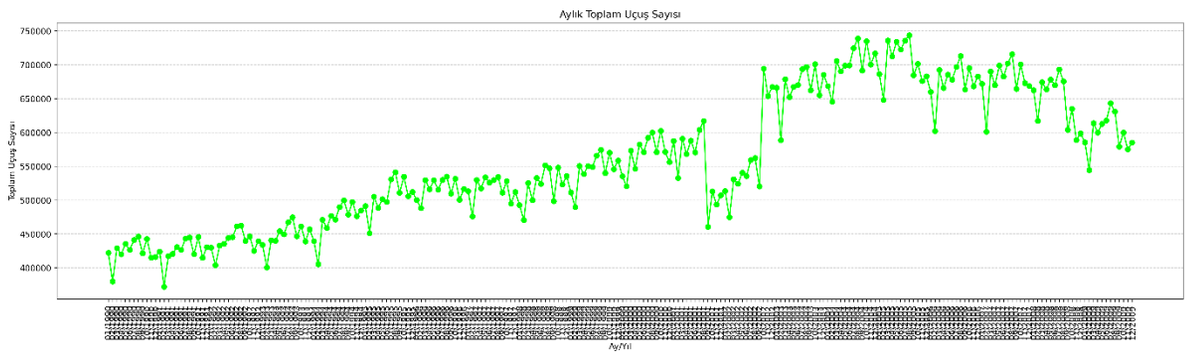
Şekil 97'de en çok uçuş yapılan havaalanlarının sıralaması gösterilmektedir. Chicago O'Hare Uluslararası Havaalanı, 158.214 uçuş ile en çok uçuş yapılan havaalanı olmuştur. İkinci sırada

131.947 uçuş ile Atlanta Hartsfield-Jackson Uluslararası Havaalanı gelmektedir. Üçüncü sırada ise 109.405 uçuş ile Dallas Fort Worth Uluslararası Havaalanı gelmektedir.



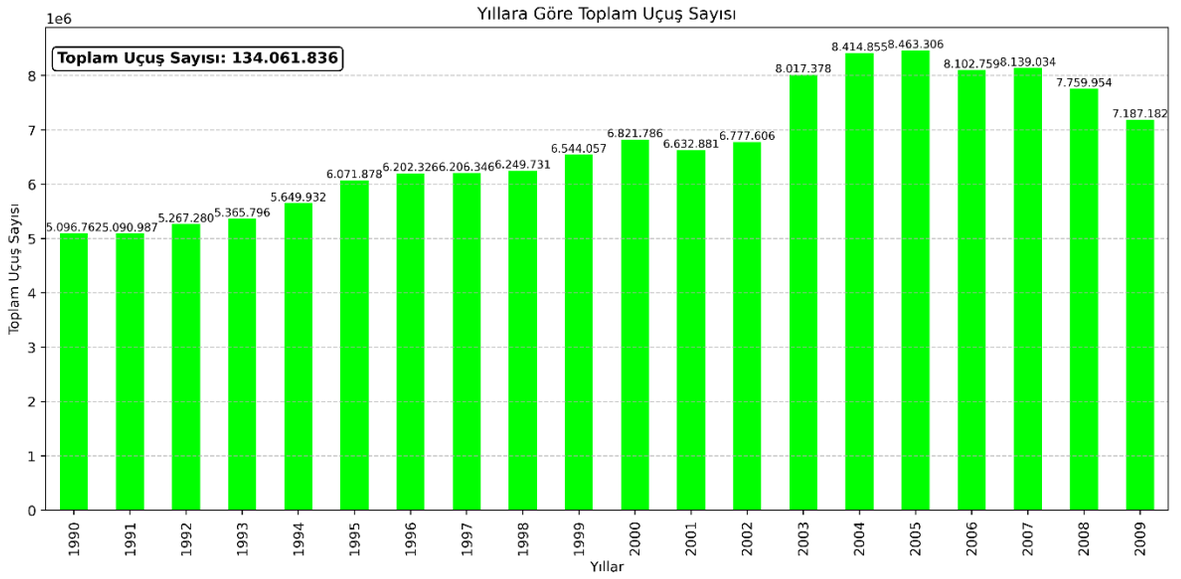
Şekil 98. En Az Uçuş Yapılan Varış Havaalanları

Şekil 98'de ise en az uçuş yapılan havaalanlarının sıralaması gösterilmektedir. 89 havaalanına sadece 1 defa uçuş yapılmıştır. Bu havaalanları arasında Kaliforniya'da bulunan Red Bluff Municipal Havaalanı, Güney Carolina'da bulunan Orangeburg Municipal Havaalanı gibi bölgesel ve dar kapsamlı havaalanları bulunmaktadır.



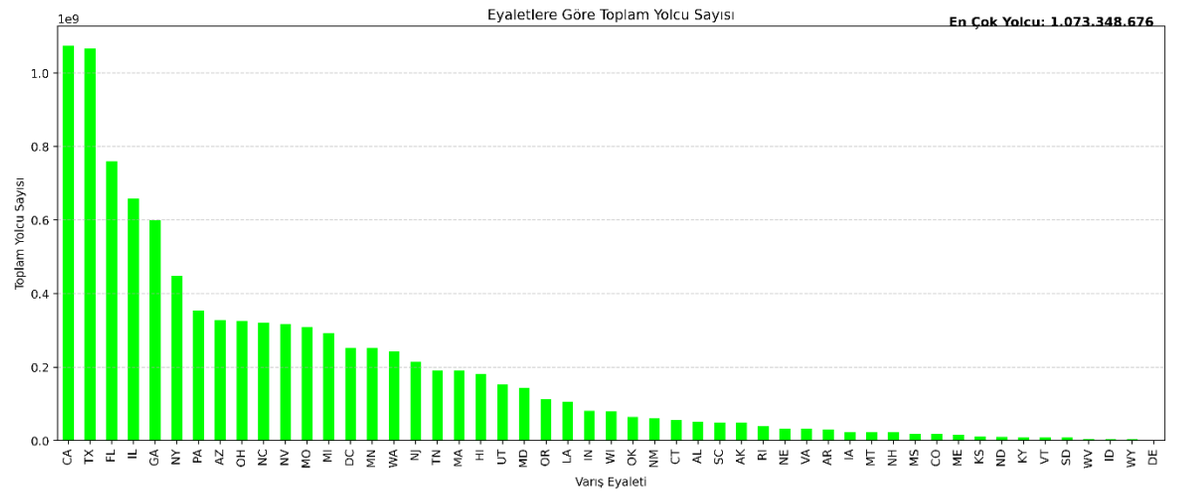
Şekil 99. Aylık Bazdaki Uçuş Sayıları

Şekil 99'da ise aylık bazdaki uçuş sayıları gösterilmektedir. En çok uçuş yapılan ay ve yıl Ağustos 2005 olup, ilgili ayda toplam 743.671 uçuş yapılmıştır. En az uçuş ise Şubat 1991'de 372.149 uçuş olarak gerçekleşmiştir.



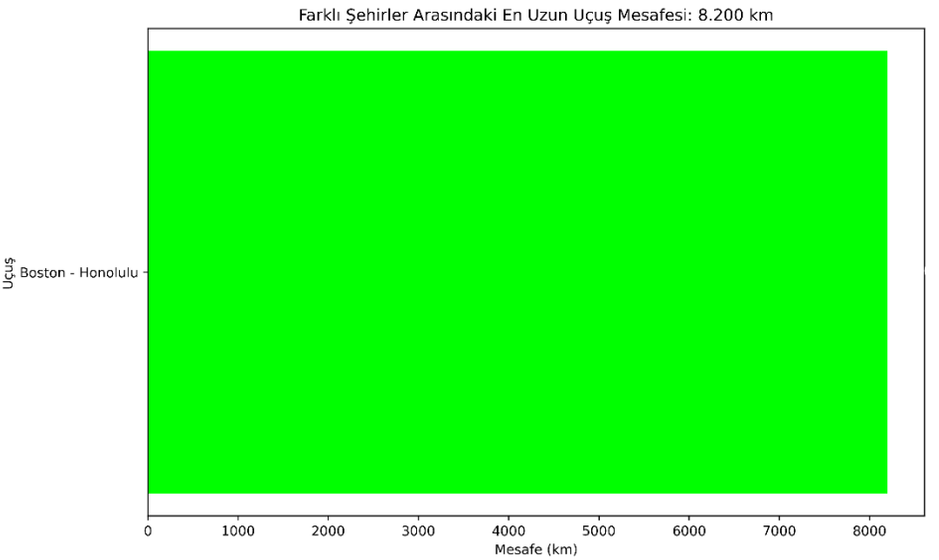
Şekil 100. Yıllık Bazdaki Uçuş Sayıları

Şekil 100'de ise yıllık bazda uçuş sayıları gösterilmektedir. Grafik, aylık bazdaki uçuş sayılarına göre daha sade ve anlaşılır özelliktedir. 2005 yılı, en çok uçuş yapılan yıl olarak göze çarpmakta ve yıl boyunca toplam 8.463.306 adet uçuşun gerçekleştiği görülmektedir. Yıl bazlı en az uçuş ise 1990 yılı olup toplamda 5.096.762 olarak gerçekleşmiştir. 1990-2009 yılları arası toplam 134.061.836 uçuş gerçekleştirilmiştir. 20 yılın ortalaması ise 6.703.092 uçuş olmuştur.



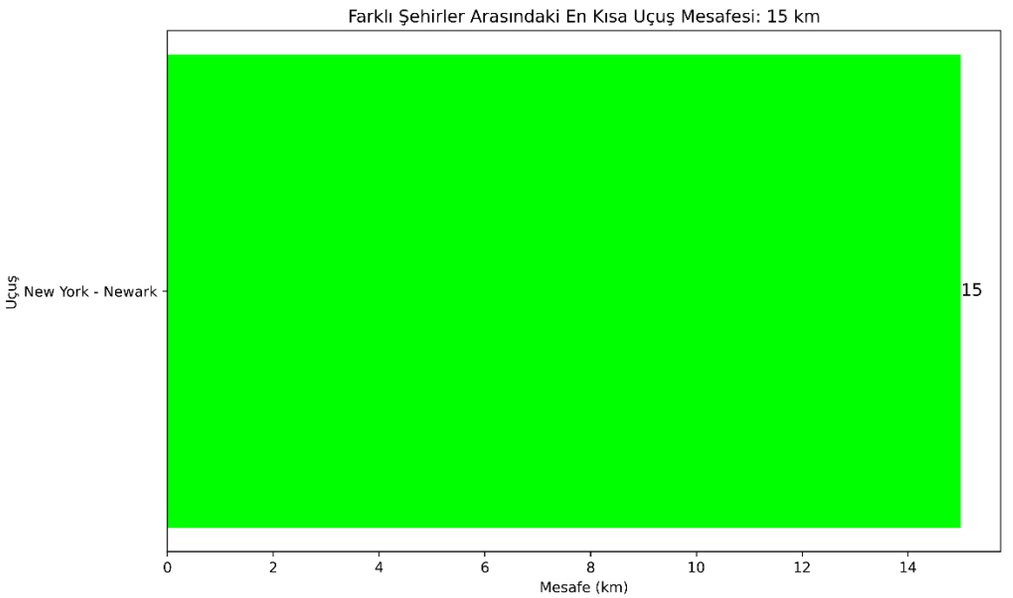
Şekil 101. Varış Eyaletlerine Göre Yolcu Sayıları

Şekil 101'de ise varış eyaletlerine göre toplam yolcu sayıları gösterilmektedir. En çok gelen yolcu sayısı, 1.073.348.676 ile California eyaletindedir. Texas eyaletinde ise bu sayı 1.066.021.714'tür. Üçüncü sırada ise Florida eyaleti 758.803.852 yolcu ile gelmektedir. En az yolcular ise sondan itibaren sırasıyla 28.375 yolcu ile Delaware eyaleti, 3.069.597 yolcu ile Wyoming eyaleti ve 3.293.543 yolcu ile Idaho eyaleti gelmektedir.



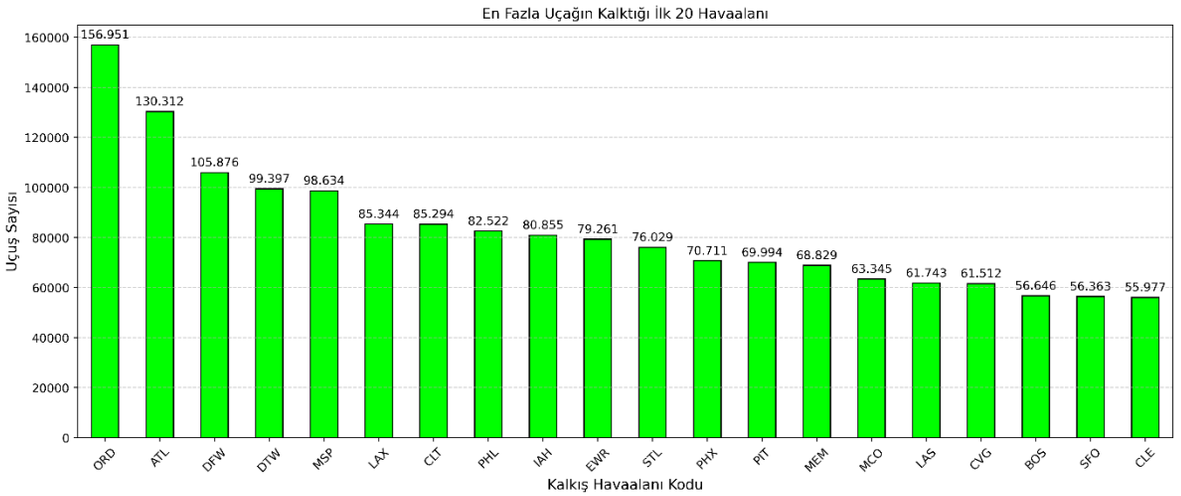
Şekil 102. Farklı Şehirler Arası En Uzun Uçuş Mesafesi

Şekil 102'de görüldüğü üzere farklı şehirler arasındaki en uzun uçuş mesafesi 8200 km ile Boston-Honolulu şehirleri arasında gerçekleşmiştir.



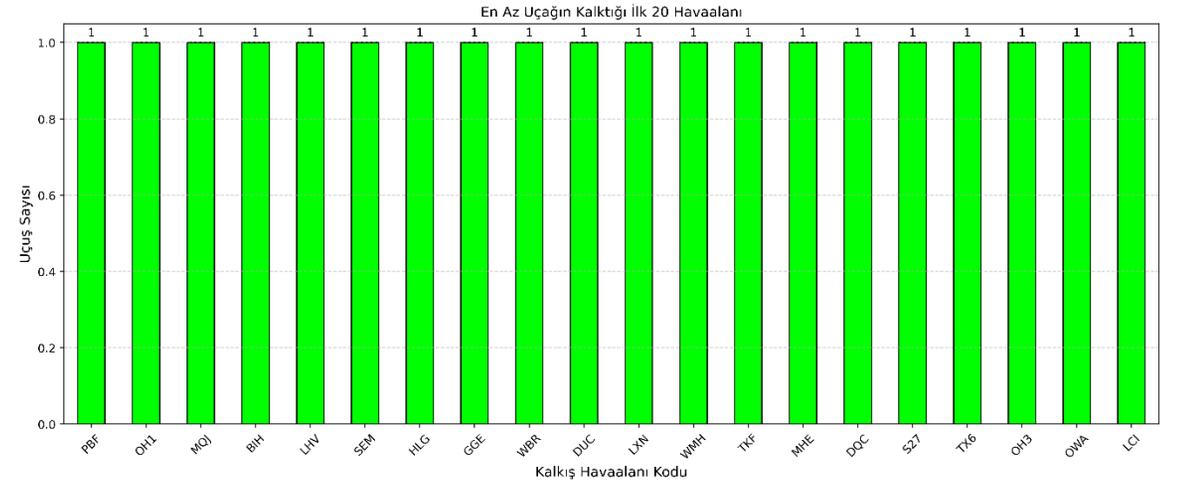
Şekil 103. Farklı Şehirler Arası En Kısa Uçuş Mesafesi

Şekil 103'te farklı şehirler arasındaki en kısa uçuş mesafesinin 15 km ile New York-Newark şehirleri arasında gerçekleştiği görülmektedir.



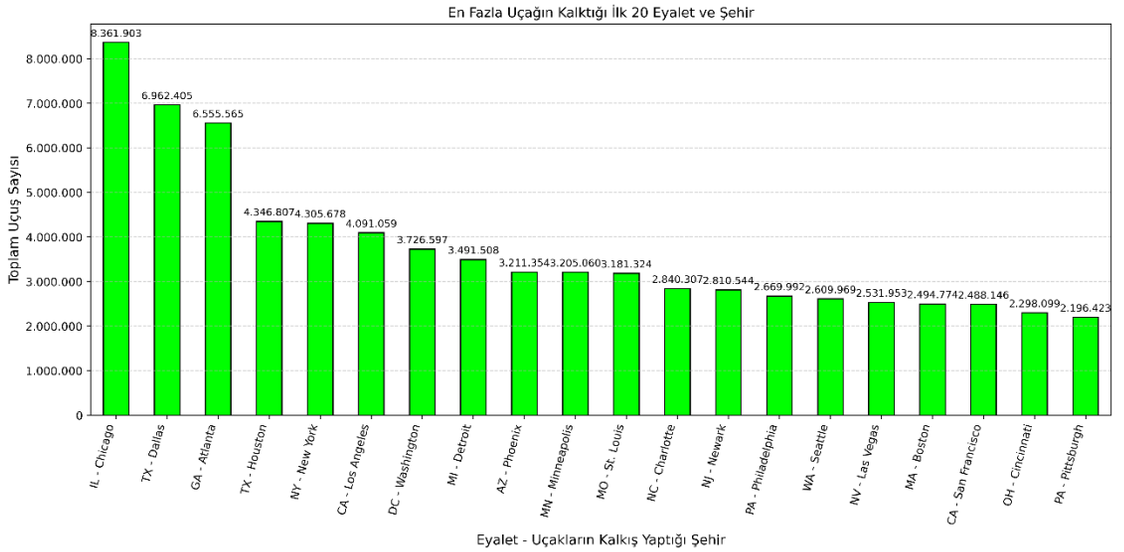
Şekil 104. En Fazla Uçağın Kalktığı Havaalanları

Şekil 104'te, en fazla kalkış yapılan havaalanlarının sıralaması gösterilmektedir. İlk sırada Chicago O'Hare Uluslararası Havaalanı (ORD) 156.951 uçuş ile gelmektedir. İkinci sırada 130.312 uçuş ile Hartsfield-Jackson Uluslararası Havaalanı (ALT) gelmektedir. Üçüncü sırada ise 105.876 uçuş ile Dallas Fort Worth Uluslararası Havaalanı (DFW) gelmektedir.



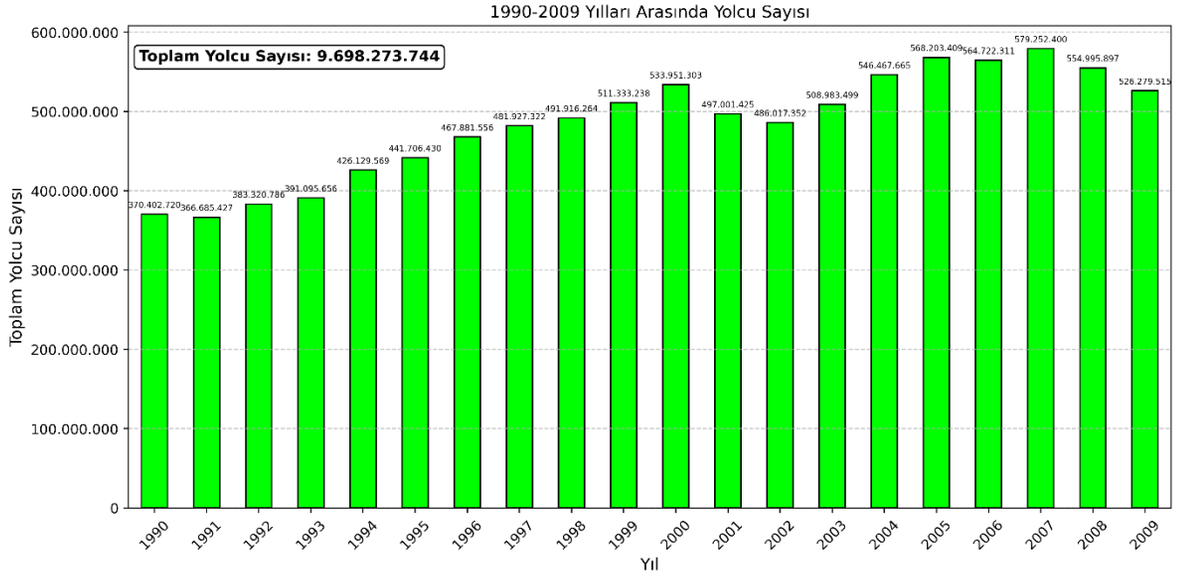
Şekil 105. En Az Uçağın Kalktığı Havaalanları

Şekil 105'te, en az kalkış yapılan havaalanlarının sıralaması gösterilmektedir. 63 adet küçük ölçekli havaalanından sadece 1'er defa uçuş yapılmıştır. Bu havaalanları arasında Nebraska eyaletinde bulunan Jim Kelly Field Havaalanı, South Carolina eyaletine bulunan Georgetown İlçe Havaalanı ve Arkansas eyaletinde bulunan Grider Field Havalanı bulunmaktadır.



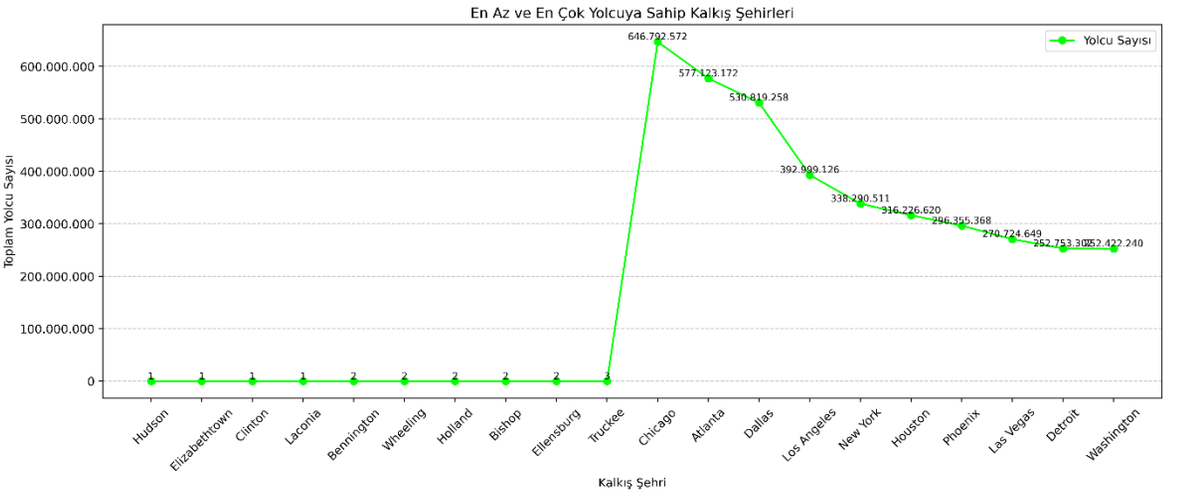
Şekil 106. En Fazla Uçağın Kalktığı Şehir ve Eyaletleri

Şekil 106'da en fazla uçağın kalkış yaptığı şehir ve şehirlerin bağlı olduğu eyaletler sıralanmaktadır. En fazla uçuş, 8.361.903 adet ile Illinois eyaletindeki Chicago şehrinden yapılmıştır. İkinci sırada ise 6.962.405 uçuş ile Texas eyaletindeki Dallas şehri bulunmaktadır. Üçüncü sırada ise 6.555.565 uçuş ile Georgia eyaletindeki Atlanta şehri gelmektedir. New York şehri ise 4.305.678 uçuş ile beşinci sırada bulunmaktadır.



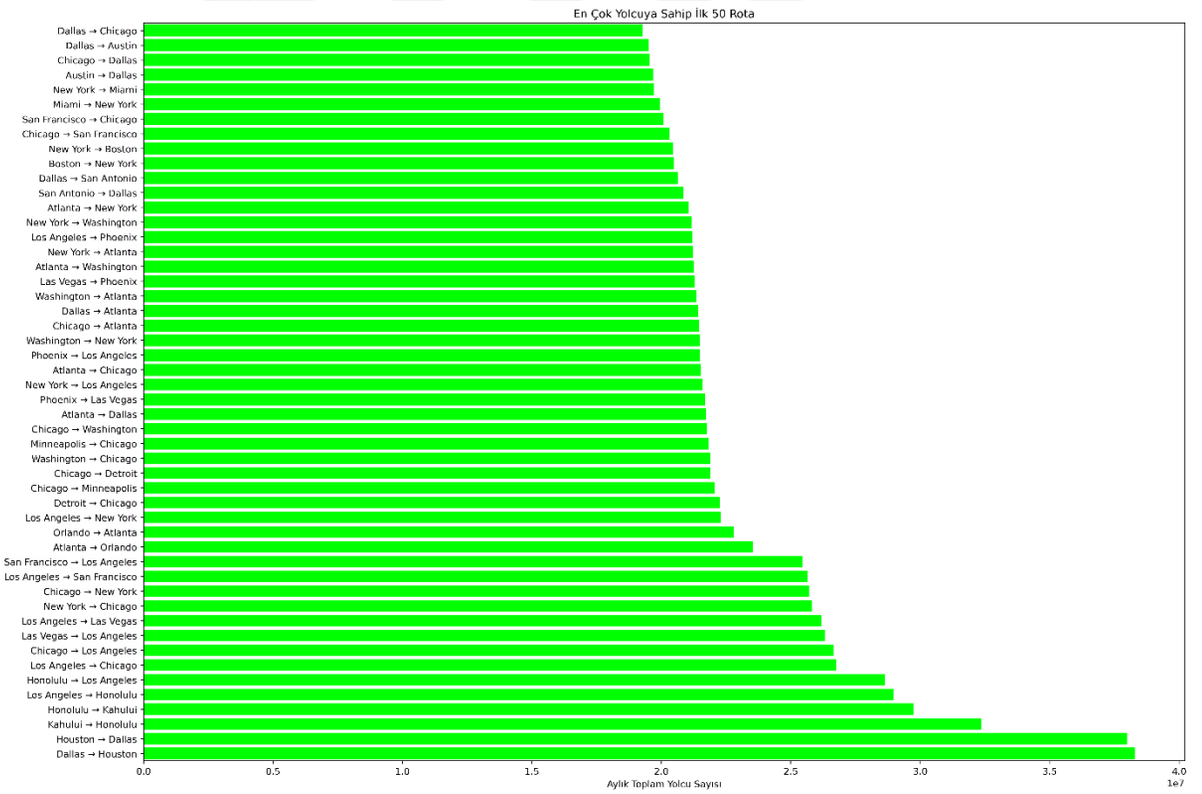
Şekil 107. Yıl Bazlı ve Toplam Yolcu Sayıları

Şekil 107'de ise yıllar bazında toplam yolcu sayıları gösterilmektedir. En az yolcu sayısı, 1990 yılında 370.402.720 yolcu ile gerçekleşmiştir. En çok yolcu sayısı ise 579.252.400 ile 2007 yılına aittir. Tüm yıllar bazında ise toplam yolcu sayısı 9.698.273.744 olarak gerçekleşmiştir. Yıllık ortalama yolcu sayısı ise 484.913.687'dir.



Şekil 108. En Az ve En Fazla Yolcuya Sahip Kalkış Şehirleri

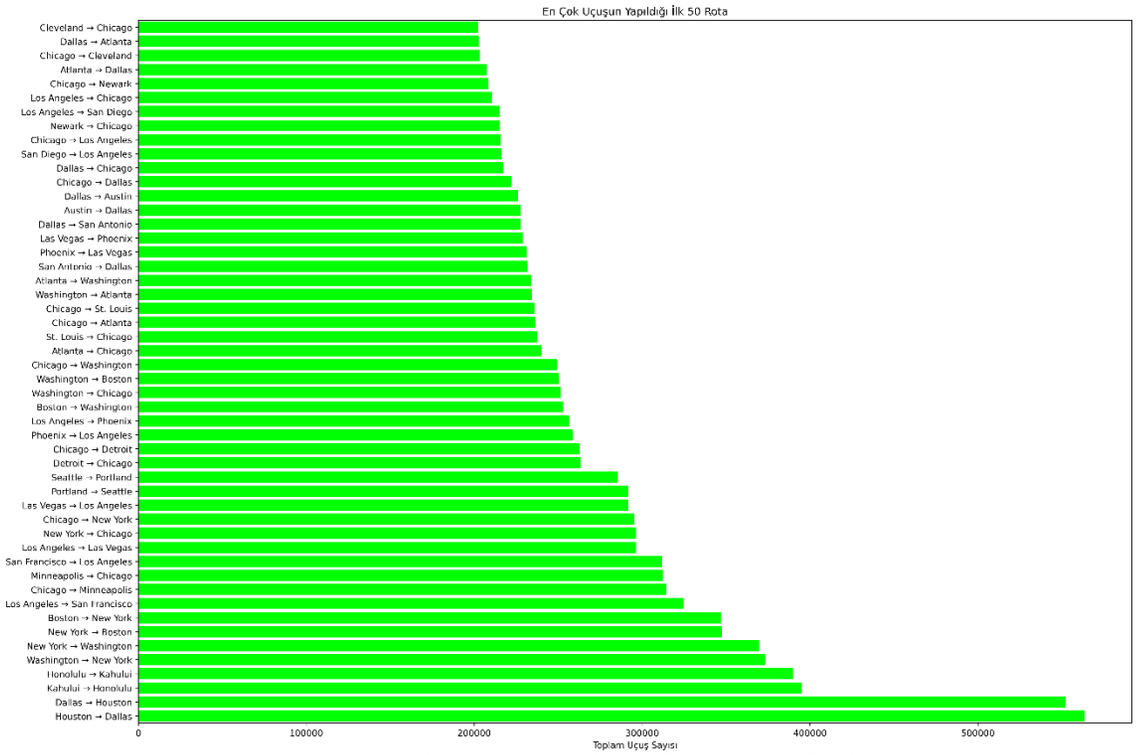
Şekil 108'de en az ve en çok yolcuya sahip olan kalkış havaalanlarının bulunduğu şehirler sıralanmaktadır. Hudson, Elizabethtown ve Clinton gibi küçük şehirler veya ilçelerde sadece 1'er yolcu uçuş gerçekleştirmişken, Chicago, Atlanta ve Dallas gibi büyük şehirlerden sırasıyla 646.792.572, 577.123.172 ve 530.819.258 yolcu uçuş gerçekleştirmiştir.



Şekil 109. En Çok Yolcuya Sahip İlk 50 Rota

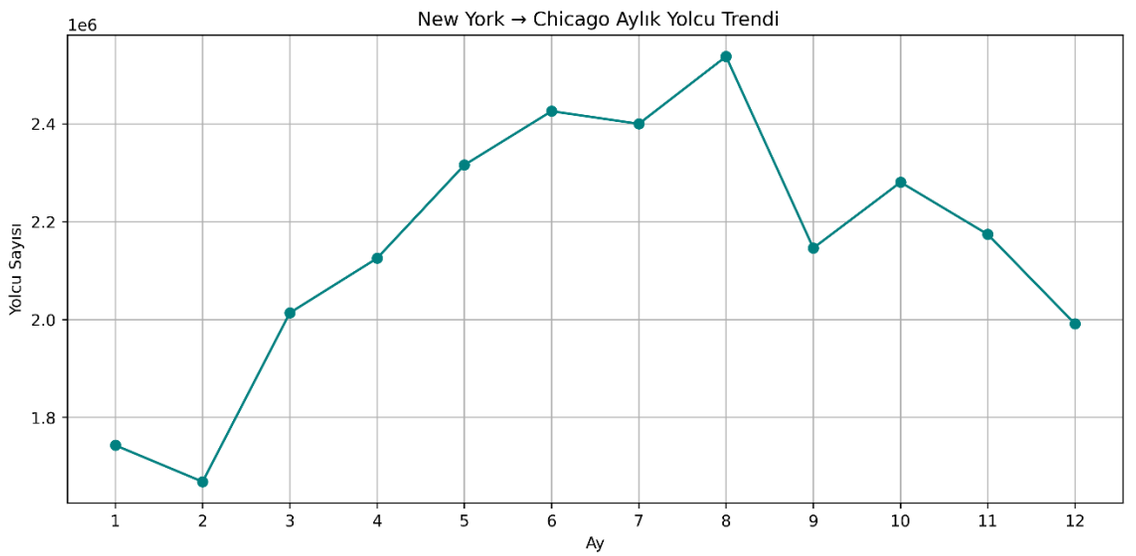
Şekil 109'da en çok yolcu sirkülasyonuna sahip ilk 50 rota belirlenmiştir. Tüm yıllar baz alındığında, aylık en çok yolcu sirkülasyonunun karşıklıklı olarak Dallas ve Houston şehirleri

arasında olduğu görülebilir. Bu karşılıklı rotadaki aylık toplam yolcu sayısı her bir yön için yaklaşık 38.000.000 olmak üzere toplam yaklaşık 76.000.000'dur.



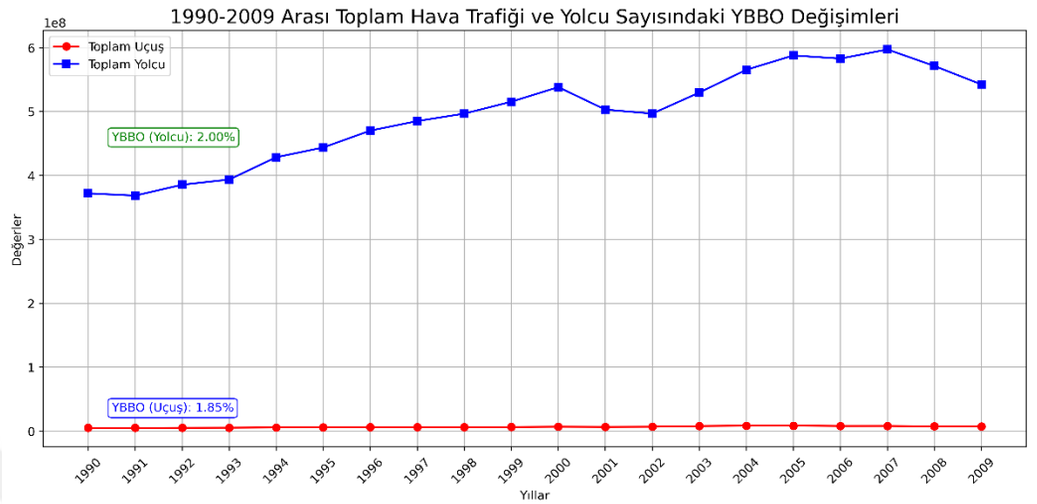
Şekil 110. En Çok Uçuşa Sahip İlk 50 Rota

Bir önceki grafikle doğru orantılı olarak, en çok uçuşa sahip olan rota Houston ve Dallas şehirleri arasındadır. Her bir yön için yaklaşık 570.000 uçuş olmak üzere toplam yaklaşık 1.040.000 uçuş, 1990-2009 yıllarında gerçekleştirilmiştir.



Şekil 111. New York-Chicago Arası Aylık Yolcu Trendi

Şekil 111'de ise en çok kullanılan ilk 50 rotadan New York-Chicago arası aylık yolcu sayısı trendi görülmektedir. En fazla yolcu sayısı yaklaşık 2.500.000 ile Ağustos aylarında gerçekleşmiştir. En az yolcu sayısı ise yaklaşık 1.700.000 ile Şubat aylarında gerçekleşmiştir.

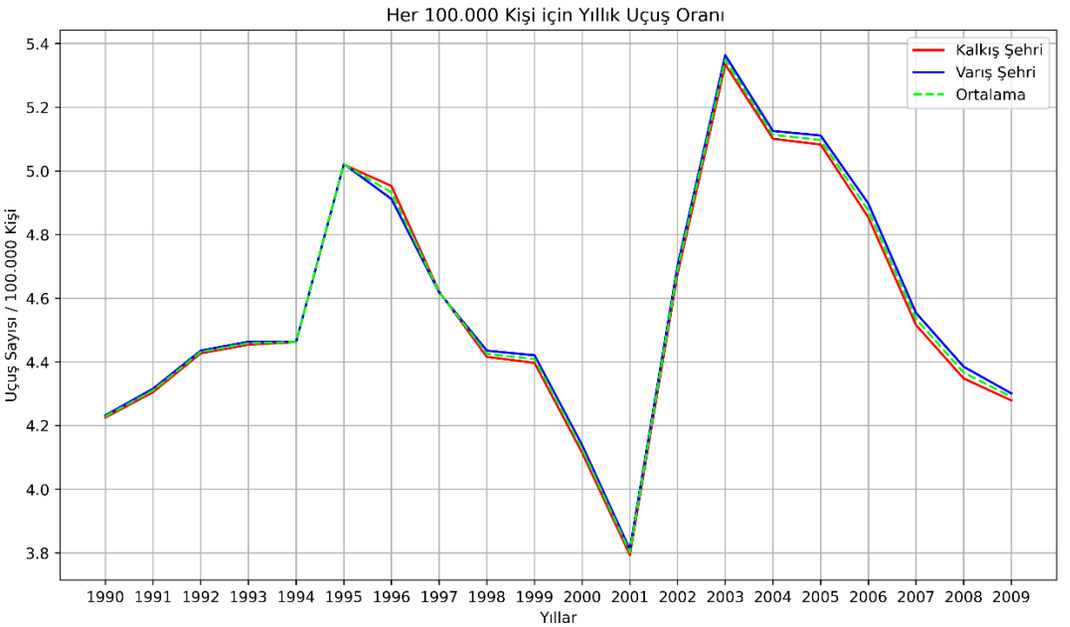


Şekil 112. Toplam Hava Trafiği ve Yolcu Sayısının Yıllık Bileşik Büyüme Oranı (YBBO)

Şekil 112'de, toplam uçuş ve toplam yolcu hacmi gösterilmektedir. Yolcu ve uçuş sayılarının uzun vadeli eğilimlerini özetlemek için her iki metriğe de bileşik Yıllık Bileşik Büyüme Oranı (YBBO) uygulanmıştır. Formülü şu şekilde ifade edilmektedir:

$$YBBO = (\frac{V_f}{V_i})^{\frac{1}{n}} - 1 \qquad (4.1)$$

Formülde V_f, dönem sonundaki değeri; V_i, dönem başlangıcındaki değeri; n, V_f ile V_i arasındaki dönem sayısını ifade etmektedir. Formül, bir değerin n dönem boyunca yıllık bileşik büyümesinin ortalamasını vermektedir. Şekil incelendiğinde, toplam uçuş sayısının 20 yıl boyunca oldukça az bir artışa sahip olduğu görülebilir. Uçuş sayısının YBBO değeri %1,85 olarak hesaplanmıştır. Yani, başlangıç dönemi dahil 20 dönem boyunca toplam uçuş sayılarında oldukça az bir yıllık artış bulunmaktadır. Yolcu hacmi ise özellikle 1990'ların sonu ve 2000'lerin başında güçlü bir büyüme eğilimi göstermiştir. Yolcu sayısının YBBO değeri ise uçuş sayısından biraz daha fazla (%2,00) olarak hesaplanmış ve bu durum da yıllık bazda az da olsa daha dolu uçuşlara doğru bir eğilimi yansıtmıştır. 2001 yılı civarında ve yine 2007 yılı sonrasında yolcu sayısındaki belirgin düşüş, sırasıyla 11 Eylül saldırıları ve küresel ekonomik kriz bağlamında yorumlanabilir. Toplam uçuş sayısına göre yolcu hacmindeki daha yüksek YBBO değeri, uçak kapasitelerinde veya rotalardaki iyileştirmeleri yansıtabilir. YBBO, aritmetik ortalamaya göre dönem bazında yaşanan dalgalanmaları yumuşattığından dolayı tercih edilmiştir.



Şekil 113. Her 100.000 Kişiye Düşen Yıllık Uçuş Oranı

Şekil 113'te ise 1990-2009 yılları arasında her 100.000 kişiye düşen yıllık uçuş oranı rota bazlı gösterilmektedir. Şekilden de görüldüğü üzere, sert iniş ve çıkışların 1995-2001, 2001-2003 ve 2005-2009 yılları arasında olduğu görülmektedir. ABD'de yaşanan 11 Eylül 2001 saldırıları hem iç hem de dış hava trafiğinde güvenlik gerekçeleri nedeniyle ani ve büyük bir kesintiye yol açmıştır (Ito ve Lee, 2005). 11 Eylül saldırıları nedeniyle uçuş sayıları ve uçakla seyehat eden nüfusun oranı nüfus oranı 2001'de en düşük noktasına ulaşırken, 2002 ve 2003 yıllarındaki ani yükseliş, havacılık sektöründe hızlı bir toparlanma yaşandığını gösterebilir. 2004 yılından başlayıp 2009 yılına kadar olan sürede kademeli bir düşüş gözlemlenmiştir. Sun vd. (2024) bu durumu 2008 küresel krizinin öncü etkileri ve yakıt maliyetlerinin yükselmesi ile açıklamaktadır.

4.3. Makine Öğrenmesi Uygulamaları

Bu bölümde denetimli ve denetimsiz öğrenme algoritmaları ile çeşitli analizler yapılmış ve bulgular değerlendirilmiştir.

4.3.1. Denetimli Öğrenme Uygulamaları

Denetimli öğrenme algoritmalarında ise veri seti 1'e öncelikle regresyon analizleri, hemen ardından ise sınıflandırma analizleri yapılmıştır.

4.3.1.1. Regresyon Analizleri

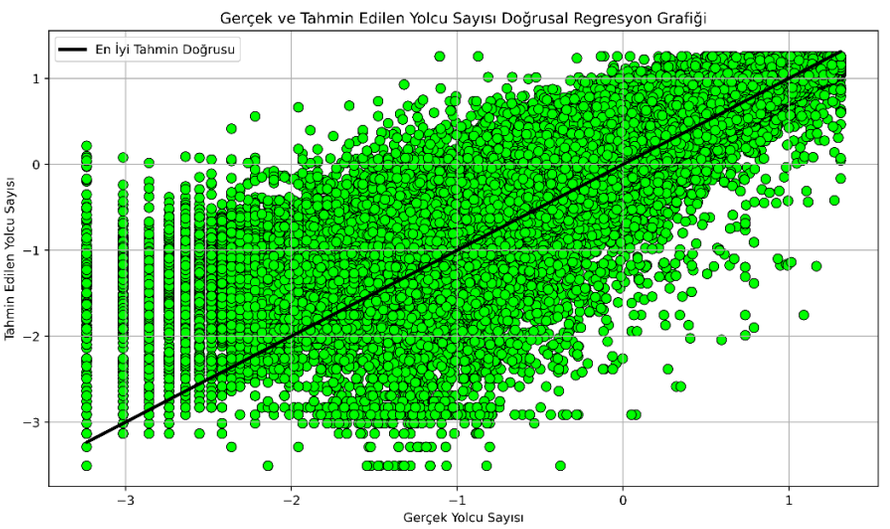
Regresyon analizlerine basit doğrusal regresyon ile başlanmıştır. "yolcu_sayisi" değişkenin bağımlı, "koltuk_sayisi" değişkenin bağımsız değişken olduğu ilk modelde amaç; koltuk

sayısının yolcu sayısına olan etkisini ölçmektir.

Tablo 20. Basit Doğrusal Regresyon 1 Ölçüm Değerleri

Metrik	Değer	Bağımsız Değişken	Bağımlı Değişken
R^2	0,9115	Koltuk Sayısı	Yolcu Sayısı
Düzeltilmiş R^2	0,9115		
RMSE	0,2975		
MSE	0,0885		
MAE	0,1899		
EVS	0,9115		
MAPE (filtrelenmiş)	0,6962		
Katsayı (Coefficient)	0,9547		
Kesme Noktası (Intercept)	0,000017		

Tablo 20'de basit doğrusal regresyon modelinin değerlendirme metrikleri ve elde edilen sonuçlar gösterilmektedir. R^2 skoru incelendiğinde modelin, yolcu sayısındaki toplam varyansın ~%91,15'ini açıkladığı görülmektedir. Başka bir deyişle, uçaklardaki koltuk sayısı ile yolcu sayısı arasında oldukça güçlü bir doğrusal ilişki bulunmaktadır. Düzeltilmiş R^2 skoru da R^2 ile aynıdır. Eğer modelde sadece bir bağımsız değişken varsa, düzeltilmiş R^2 ile R^2 birbirine çok yakın ve hatta eşit olabilir (Bhandari, 2025). Modelde koltuk sayısı değişkeninin, yolcu sayısının oldukça güçlü bir kestirimcisi olduğu söylenebilir. RMSE (0,2975) ve MAE (0,1899) skorları da modelin düşük ortalama hata ile doğru tahmin ürettiğini göstermektedir. MSE skoru (0,0885) da oldukça düşüktür. 0,9547'lik model katsayısı, her bir ek koltuk için ortalama yolcu sayısının 0,9547 kişi arttığını göstermektedir. Kesme noktası skoru (0,000017) neredeyse sıfırdır, bu da modelin sıfır koltuk olduğunda neredeyse sıfır yolcu tahmin ettiği anlamına gelmektedir.



Şekil 114. Basit Doğrusal Regresyon Grafiği 1

Şekil 114'te basit doğrusal regresyon modelinin dağılım grafiği görülmektedir. X ekseni, standardize edilmiş gerçek yolcu sayısını, Y ekseni ise tahmin edilen yolcu sayısını göstermektedir. Grafiği iki eşit parçaya bölen siyah çizgi ise tahminlerin gerçek değerlere eşit

olduğu ideal doğruyu ifade etmektedir. Yeşil noktalar ise modelin ürettiği tahminlerdir. Şekil incelendiğinde, dağılım noktalarının en iyi tahmin doğrusu etrafında kümelendiği görülebilir. Başka bir deyişle model, koltuk_sayisi değişkeni ile yolcu_sayisi değişkeni arasındaki doğrusal ilişkiyi yakalamıştır. Grafiğin sol alt bölgesindeki[215] seyrek ve sapmış noktalar, modelin düşük yolcu sayısı içeren örneklerde tahmin performansının azaldığını göstermektedir. Bu durum, örneğin kısa mesafeli veya düşük talep gören uçuşlarda modelin genelleme yeteneğinin zayıfladığına işaret olabilir.

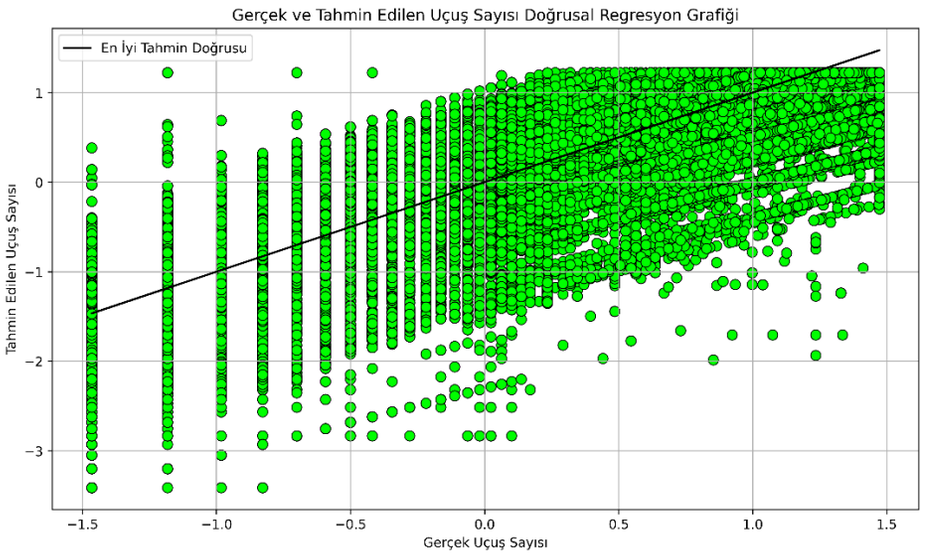
Bir diğer basit doğrusal regresyon modeli ise, bağımsız değişken olarak "koltuk_sayisi", bağımlı değişken olarak da "ucus_sayisi" seçilerek oluşturulmuştur. Bu modelde ise amaç; koltuk sayısının uçuş sayısına olan etkisini ölçmektir. Analizler sonucunda değerlendirme metrikleri ve sonuçları aşağıdaki tabloda gösterilmiştir.

Tablo 21. Basit Doğrusal Regresyon 2 Ölçüm Değerleri

Metrik	Değer	Bağımsız Değişken	Bağımlı Değişken
R^2	0,8618	Koltuk Sayısı	Uçuş Sayısı
Düzeltilmiş R^2	0,8618		
RMSE	0,3716		
MSE	0,1381		
MAE	0,2855		
EVS	0,8618		
MAPE (filtrelenmiş)	0,7262		
Katsayı (Coefficient)	0,9285		
Kesme Noktası (Intercept)	-0,00015158		

Tablo 21'de ise ikinci basit doğrusal regresyon modeline ait değerlendirme metrikleri ve elde edilen sonuçlar gösterilmiştir. 0,8618'lik R^2 skoru, uçuş sayısındaki varyansın %86,18'inin koltuk sayısıyla açıklanabildiğini göstermektedir. İlk model kadar olmasa da değişkenler arasında güçlü bir doğrusal ilişki vardır denilebilir. 0,3716'lık RMSE ve 0,2855'lik MAE skorları, nispeten düşük ortalama tahmin hatalarını ve modelin genelde iyi yapılandığını göstermektedir. MSE skoru da (0,1381) düşük görünmektedir. Model katsayısı skoru (0,9285), her ek koltuk için uçuş sayısının ~0,93 arttığını ve uçakların koltuk kapasitesi ile seferlerin sıklığı arasında doğrusal bir ilişki olduğunu göstermektedir. Kesme noktası ise -0,00015'lik oldukça düşük skor ile modelin koltuk olmadığında neredeyse sıfır sefer tahmin ettiğini göstermektedir.

[215] Hem gerçek hem de tahmin edilen değerlerin düşük olduğu alan.



Şekil 115. Basit Doğrusal Regresyon Grafiği 2

Şekil 115'te ikinci basit doğrusal regresyon modelinin dağılım grafiği görülmektedir. Şekil incelendiğinde, dağılımların hem dikey hem de yukarı doğru yatay bir eğilim gösterdiği söylenebilir. Yani model, koltuk sayısı ile uçuş sayısı arasında doğrusal bir ilişki yakalamıştır ancak grafiğin sol alt bölgesinde, gerçek uçuş sayıları düşük olduğunda tahminlerde daha fazla varyans görülmektedir.

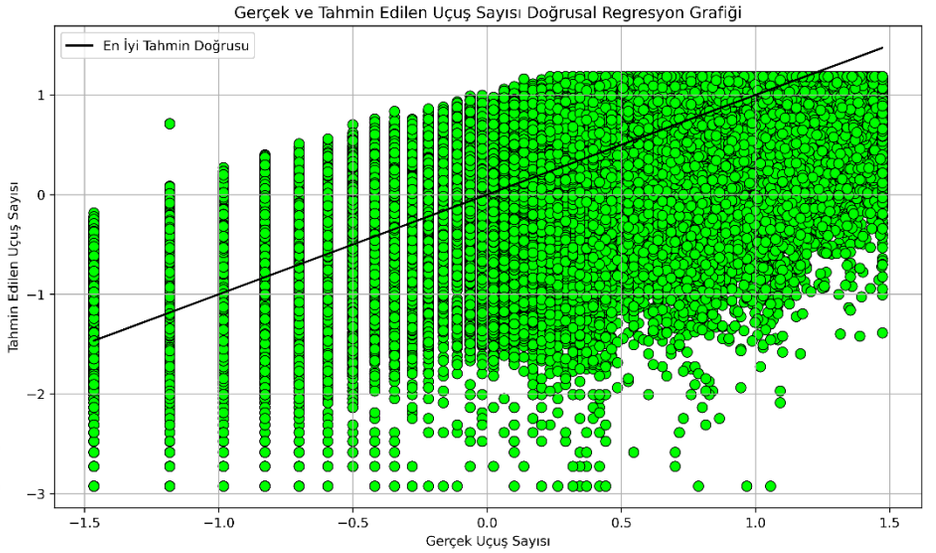
Üçüncü basit doğrusal regresyon modeli ise bağımsız değişken "yolcu_sayisi", bağımlı değişken ise "ucus_sayisi" seçilerek oluşturulmuştur. Buradaki hedef ise yolcu sayısının uçuş sayısına olan etkisini ölçmektir. Analizler sonucunda değerlendirme metrikleri ve sonuçları aşağıdaki tabloda gösterilmiştir.

Tablo 22. Basit Doğrusal Regresyon 3 Ölçüm Değerleri

Metrik	Değer	Bağımsız Değişken	Bağımlı Değişken
R^2	0,8160	Yolcu Sayısı	Uçuş Sayısı
Düzeltilmiş R^2	0,8160		
RMSE	0,4289		
MSE	0,1839		
MAE	0,3295		
EVS	0,8160		
MAPE (filtrelenmiş)	0,8197		
Katsayı (Coefficient)	0,9036		
Kesme Noktası (Intercept)	0,0001721		

Tablo 22'de ise üçüncü basit doğrusal regresyon modeline ait değerlendirme metrikleri ve elde edilen sonuçlar sunulmuştur. R^2 metriği, 0,8160'lık değeri ile uçuş sayısındaki varyansın %81,60'ının yolcu sayısıyla açıklanabileceğini göstermektedir. RMSE (0,4289) ve MAE (0,3295) skorları, bir önceki modelde kullanılan koltuk_sayisi değişkenine göre daha yüksek tahmin hatasını yansıtmaktadır. Filtrelenmiş MAPE skoru (0,8197) modelin ortalama yüzde hatasının nispeten yüksek olduğunu göstermektedir. Ancak veriler standardize edildiğinden,

bu yüksek skor ham birim hatalarından daha çok orantılı hataları yansıtmaktadır. Model katsayı skoru (0,9036), her ek yolcu için uçuş sayısının ~0,90 oranında arttığını ifade etmektedir. Kesme noktası skoru (−0,0001721) neredeyse sıfır olup, modelin, herhangi bir yolcu olmadığında uçuş sayısının da neredeyse sıfır olacağını göstermektedir.



Şekil 116. Basit Doğrusal Regresyon Grafiği 3

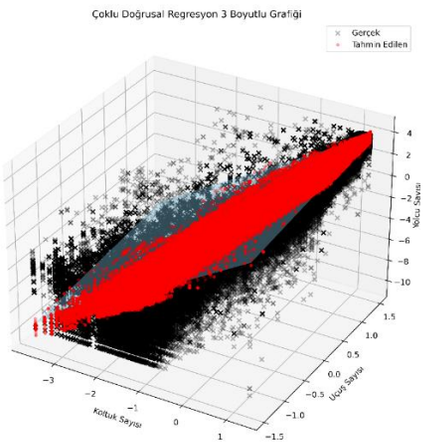
Şekil 116'da üçüncü basit doğrusal regresyon modelinin dağılım grafiği görülmektedir. Diğer iki modele göre daha düşük R^2 skoruna sahip olan üçüncü model, yine de doğrusal bir eğilim sergilemektedir. Grafik incelendiğinde gerçek uçuş sayıları arttıkça, tahmin edilen uçuş sayısının da arttığı görülebilir. X eksenindeki 0 noktasından itibaren gerçek uçuş sayısı arttıkça, tahminlerin çoğunun en iyi tahmin doğrusunun altında kaldığı görülmektedir. Bu durum, modelin özellikle daha yüksek uçuş sayısı için, daha az tahmin yeteneğine sahip olduğunun bir işaretidir.

Çoklu doğrusal regresyon modeli ise "ucus_sayisi", "mesafe_km" ve "koltuk_sayisi" bağımsız değişkenleri ile "yolcu_sayisi" bağımlı değişkeni seçilerek oluşturulmuştur. Bu modeldeki amaç ise uçuş sayısı, mesafe ve koltuk sayısının yolcu sayısını ne derecede etkileyebildiğini ölçmektir. Analizler sonucunda değerlendirme metrikleri ve sonuçları aşağıdaki tabloda gösterilmiştir.

Tablo 23. Çoklu Doğrusal Regresyon 1 Ölçüm Değerleri

Metrik		Değer	Bağımsız Değişken	Bağımlı Değişken
R^2		0,9197	Uçuş Sayısı Mesafe Koltuk Sayısı	Yolcu Sayısı
Düzeltilmiş R^2		0,9197		
RMSE		0,2833		
MSE		0,0803		
MAE		0,1839		
EVS		0,9197		
MAPE (filtrelenmiş)		0,6709		
Katsayı (Coefficient)	koltuk_sayisi	0,7670		
	ucus_sayisi	0,1914		
	mesafe_km	0,0836		
Kesme Noktası (Intercept)		0,0000		

Tablo 23'te birinci çoklu doğrusal regresyon modeline ait değerlendirme metrikleri ve elde edilen sonuçlar gösterilmektedir. Model 0,9197'lik R^2, Düzeltilmiş R^2 ve EVS skorları ile; bağımlı değişken olan yolcu sayısındaki varyansın ~%91,97'sinin üç bağımsız değişken (koltuk_sayisi, ucus_sayisi, mesafe_km) tarafından açıklanabildiğini göstermektedir. Modelin 0,2833'lük RMSE ve 0,1839'luk MAE skorları, nispeten düşük kabul edilebilir. 0,0803'lük MSE skoru, gerçek verilerden ortalama karesel sapmayı göstermektedir. 0,6709'luk filtrelenmiş MAPE skoru ise verilerdeki ortalama mutlak yüzdelik hatanın biraz yüksek olduğunu göstermektedir. Burada da veriler standardize edildiğinden, yüksek MAPE skoru ham birim hatalarından daha çok orantılı hataları yansıtmaktadır. Modelin katsayıları incelendiğinde ise; "koltuk_sayisi" değişkeninin 0,7670'lık katsayı skoru, güçlü bir ilişkiyi işaret etmektedir. Yani, koltuk sayısındaki bir standart sapma artışı, diğer değişkenler sabit tutulduğunda, tahmin edilen yolcu sayısında 0,7670'lik standart sapma artışına karşılık gelmektedir. "ucus_sayisi" değişkeni için katsayı skoru ise 0,1914 olup, model içerisinde koltuk sayısı değişkeninden daha az etkili olduğu görülebilir. Mesafe_km değişkeni için ise katsayı skoru 0,0836'dır. Kesme noktası skorunun 0 olması, regresyon çizgisinin orijinden geçtiği anlamına gelmektedir (Steenbergen, 2016, s. 11). Böylece, tüm bağımsız değişkenler 0 olduğunda, bağımlı değişkenin tahmin edilen değeri de 0 olur.



Şekil 117. Çoklu Doğrusal Regresyon Grafiği 1

Şekil 117'de birinci çoklu doğrusal regresyon modelinin üç boyutlu tahmin grafiği gösterilmektedir. Grafikte modelde en etkili iki tahmin edici olan koltuk sayısı ve uçuş sayısı ile bağımlı değiken yolcu sayısı gösterilmektedir. Diğer bağımsız değişken olan mesafe bağımsız değişkeni, boyut kısıtlamasından dolayı ortalaması alınarak model içerisinde temsil edilmiştir. x simgesi gerçek yolcu sayılarını temsil ederken, kırmızı renkli noktalar modelin tahmin edilen değerlerini göstermektedir. Şeffaf olarak görülen açık mavi düzlem ise regresyon düzlemidir ve modelin iki bağımsız değişken ile bağımlı değişken arasındaki tahmini ilişkiyi göstermektedir. Regresyon yüzeyinin eğimli yapısı, koltuk_sayisi ve ucus_sayisi değişkenleri ile (yolcu_sayisi) arasında doğrusal bir ilişki olduğunu göstermektedir. Kırmızı olarak işaretlenen veri noktalarının gerçek değerler ile çoğunlukla örtüşmesi, iyi bir genelleme ve güçlü bir uyumu ifade etmektedir.

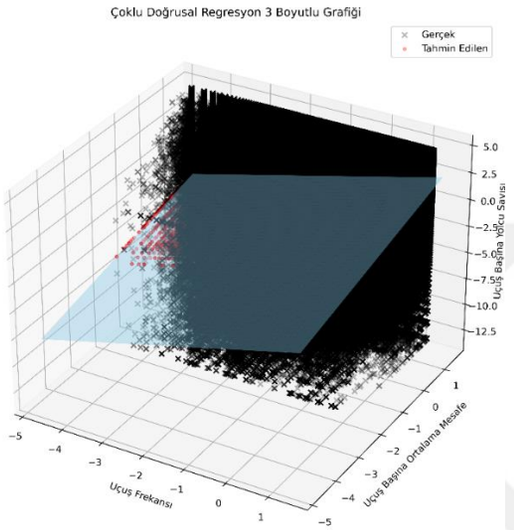
İkinci çoklu doğrusal regresyon modeli ise ucus_frekansi ve ucus_basina_ortalama_mesafe bağımsız değişkenleri ile ucus_basina_yolcu_sayisi bağımlı değişkeni seçilerek oluşturulmuştur. Bu modeldeki amaç ise uçuş başına ortalama mesafe ve seferlerin sıklığının (uçuş frekansı) sefer başına yolcu sayısına (ucus_basina_yolcu_sayisi) olan etkisinin ölçülmesidir. Analizler sonucunda değerlendirme metrikleri ve sonuçları aşağıdaki tabloda gösterilmiştir.

Tablo 24. Çoklu Doğrusal Regresyon 2 Ölçüm Değerleri

Metrik		Değer	Bağımsız Değişken	Bağımlı Değişken
R^2		0,1252	Uçuş Frekansı Uçuş Başına Ortalama Mesafe	Uçuş Başına Yolcu Sayısı
Düzeltilmiş R^2		0,1252		
RMSE		0,9365		
MSE		0,8771		
MAE		0,7132		
EVS		0,1252		
MAPE (filtrelenmiş)		1,4692		
Katsayı (Coefficient)	ucus_frekansi	0,3212		
	ucus_basina_ortalama_mesafe	0,2212		
Kesme Noktası (Intercept)		0,0000		

Tablo 24'te ikinci çoklu doğrusal regresyon modeline ait değerlendirme metrikleri ve elde edilen sonuçlar gösterilmektedir. Sonuçlar incelendiğinde, modelin istatistiki olarak zayıf performans gösterdiği söylenebilir. R^2 skoru (0,1252), bağımlı değişkenin sadece %12,52'sinin iki bağımsız değişken tarafından açıklanabildiğini göstermektedir. Düzeltilmiş R^2 ve EVS skorları da aynıdır. RMSE (0,9365) ve MAE (0,7132) yüksek hata skorları olarak göze çarpmaktadır. MSE (0,8771) ve filtrelenmiş MAPE (1,4692) skorları ise oldukça yüksek olarak değerlendirilebilir. Bu değerler, tahminlerdeki hataların yüksek olduğunu ve modelin genelleme yapmakta zorlandığını göstermektedir. Katsayılar incelendiğinde ise "ucus_frekansi" değişkeni için katsayı skoru 0,3212 olup sınırlı bir katkıyı göstermektedir.

Benzer şekilde, “ucus_basina_ortalama_mesafe” değişkeninin katsayısı ise 0,2212’dir ve modele en az katkı veren değişken olarak değerlendirilebilir. Bir önceki modelde olduğu gibi bu modelde de kesme noktası değeri 0 olarak görülmektedir. Genel olarak değerlendirildiğinde ise gerek modelin zayıf açıklama gücü gerekse yüksek tahmin hataları nedeniyle bağımsız değişkenler ile bağımlı değişken arasındaki ilişkinin tatmin edici olmadığı görülmektedir. Bağımlı değişkenin açıklanabilmesi için farklı bağımsız değişkenlerin denenmesi bir çözüm olarak sunulabilir.



Şekil 118. Çoklu Doğrusal Regresyon Grafiği 2

Şekil 118’de ise ikinci çoklu doğrusal regresyon modelinin üç boyutlu tahmin grafiği gösterilmektedir. Şekilde ilk dikkati çeken durum, tahmin noktalarının azlığı ve regresyon düzleminin nispeten düz görüntüsüdür. Modelin yetersiz uyum gösterdiği grafikten anlaşılabilir. Gerçek veri noktaları, regresyon düzleminin üstünde ve altında yaygın bir şekilde dağılmış görünmektedir. Bu durum, büyük tahmin hatalarını göstermektedir. Başka bir deyişle, tahminler, gerçek değerlerden önemli ölçüde sapmıştır. Grafik incelendiğinde değişkenler arası ilişkinin doğrusal olmayabileceği veya mevcut değişkenlerle modelin ancak zayıf seviyede açıklanabileceği de görülmektedir.

Yeni türetilen özniteliklerden uçuş başına ortalama mesafenin bağımlı değişken, diğer tüm özniteliklerin bağımsız değişken olduğu regresyon modelinde ise, özellikle kategorik değişkenler “one hot encoding”[216] işlemine tabi tutulduktan sonra oluşan eğitim seti boyutu (~2.850.000, 31.352) nedeniyle RAM hatası alınmıştır. Bu durumun nedeni ise çok fazla sayıda kategorik değişkenin ve tekil değerin (örneğin; havaalanı kodları ve şehirler) olmasıdır.

[216] Kategorik değişkenleri ikili (binary) biçime dönüştürme yöntemi. Yöntemde, her kategori için kategorinin mevcut olduğu (1) ve mevcut olmadığı (0) anlamına geldiği yeni sütunlar oluşturulur. Burada temel amaç, sıralılığı (ordinality) ortadan kaldırmak, model performansını iyileştirmek ve algoritmalara uyum sağlamaktır.

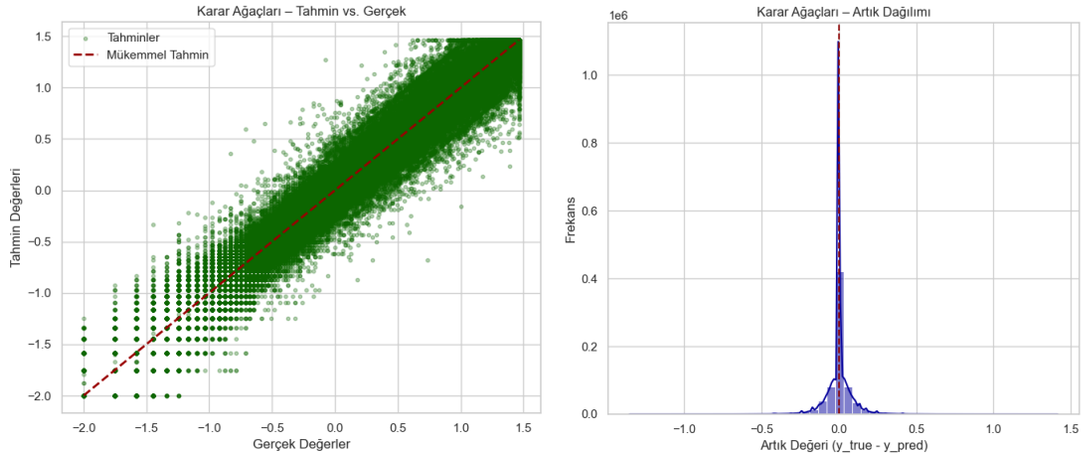
"OneHotEncoder", kategori başına her öznitelik için bir sütun oluşturduğundan, devasa sütun sayısı ile analizlere devam edilememiştir. Çözüm için ise yüksek kardinalite[217] gösteren öznitelikler ile diğer öznitelikler elendikten sonra "kalkis_eyaleti" ve "varis_eyaleti" değişkenleri bağımsız değişken olarak seçilmiş ve işlemlere devam edilmiştir. Veri setine boyut azaltma tekniği (PCA) uygulanmış, ancak hata alınmıştır. Hatanın nedeni ise "OneHotEncoder" fonksiyonunun (sparse=True) özelliğini kullanması ve PCA'nın seyrek girdileri desteklememesidir. Çözüm olarak ise "TruncatedSVD[218]" boyut azaltma yöntemi kullanılmıştır. Özellik alanı 100 bileşene sabitlendikten sonra analizlere devam edilmiştir. Regresyon görevi için yapay sinir ağı (çok katmanlı algılayıcı), rastgele orman, karar ağaçları ve k-en yakın komşu algoritmaları kullanılmış ve analiz sonuçları değerlendirme metrikler bazında karşılaştırılmıştır.

Tablo 25. Regresyon Modelleri Değerlendirme Sonuçları

Model	R^2	Düzeltilmiş R^2	RMSE	MSE	MAE	MAPE (%)	EVS
Yapay Sinir Ağı	0,9990	0,9990	0,0300	0,0009	0,0162	4,8718	0,9990
Rastgele Orman	0,9979	0,9979	0,0448	0,0020	0,0267	14,3992	0,9979
Karar Ağaçları	0,9935	0,9935	0,0804	0,0064	0,0418	24,7859	0,9935
K-En Yakın Komşu	0,9879	0,9879	0,1099	0,0120	0,0714	39,5348	0,9879

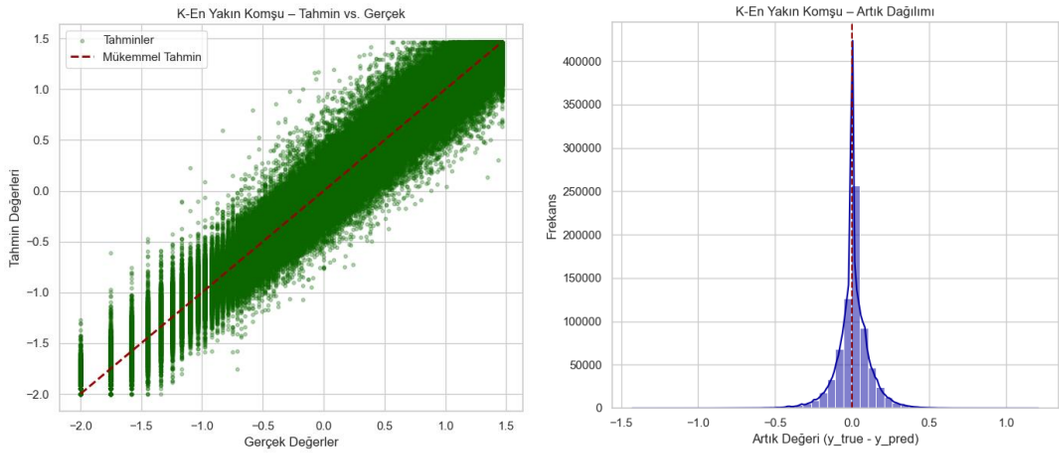
Tablo 25'teki değerlendirme metrikleri, yapay sinir ağı[219] modelinin diğer modellere göre daha tutarlı ve başarılı bir performansa sahip olduğunu göstermektedir. En yüksek R^2 skoruna (0,9990), en düşük MSE (0,0009) ve RMSE (0,0300) değerlerine sahip olan model, tahmin edilen ve gerçek değerler arasındaki minimum sapmaya sahiptir. Bu değerlere ek olarak model, en düşük MAE (0,0162) ve MAPE (%4,87) skorlarına da sahiptir. Rastgele orman modeli, 0,9979'luk R^2 skoru ile ve diğer metrikler bazında karar ağaçlarına kıyasla en güçlü ikinci performansı göstermiştir. Buna karşılık, k-en yakın komşu[220] modeli, tüm modeller içinde en düşük genelleme kapasitesini gösterse bile %98,79'luk R^2 skoru oldukça iyidir. Genel anlamda ise, tüm modeller yüksek R^2 değerlerine (>0,98) sahiptir ve bu durum da her bir modelin bağımlı değişkendeki varyansın oldukça önemli bir kısmını açıklayabildiğini göstermektedir. Tablodan çıkarılabilecek bir diğer sonuç ve genelleme ise büyük ölçekli yapılandırılmış veri setleri için tahmin edici modellemede sinir ağlarının oldukça başarılı sonuçlar vermesidir.

[217] Bir sütundaki veri değerlerinin benzersizliği. Çok fazla benzersiz değere sahip olan veya benzersiz değeri olmayan sütunların çok az varyansı vardır. Yüksek kardinaliteye (çok sayıda benzersiz değer) sahip değerlerin herhangi bir deseni belirlenemez.
[218] Kesik tekil değer ayrıştırma (SVD) yoluyla doğrusal boyut indirgemesi gerçekleştiren tekniktir. PCA'nın aksine, tekil değer ayrıştırmayı hesaplamadan önce verileri merkezlemez. Böylece seyrek matrisleri rahatlıkla işler.
[219] Multilayer Perceptron - MLP
[220] KNN



Şekil 119. Karar Ağaçları Dağılım ve Artık Değer Grafiği

Şekil 119'da sol tarafta bulunan dağılım grafiği; gerçek bağımlı değişken değerlerini, model tarafından tahmin edilen değerlerle karşılaştırmaktadır. Kırmızı kesikli çizgi (y=x) mükemmel tahmin çizgisini temsil etmektedir. Dağılım grafiği incelendiğinde, tahminlerin çoğunlukla mükemmel tahmin çizgisini takip ettiği görülebilir. Model, genel eğilimi etkili bir şekilde yakalamıştır. Sağ taraftaki grafikte ise artık[221] değer dağılımı gösterilmektedir. Grafikte, artık değerlerin sıfırın etrafında düzenli bir biçimde kümelendiği görülmektedir. Uzun ve keskin bir tepe noktasına sahip olan dağılım, tahminlerin gerçek değerlerine yakın olduğunu göstermektedir. Başka bir deyişle model, tahmin hatalarını en aza indirmede oldukça iyi bir performans göstermiştir.

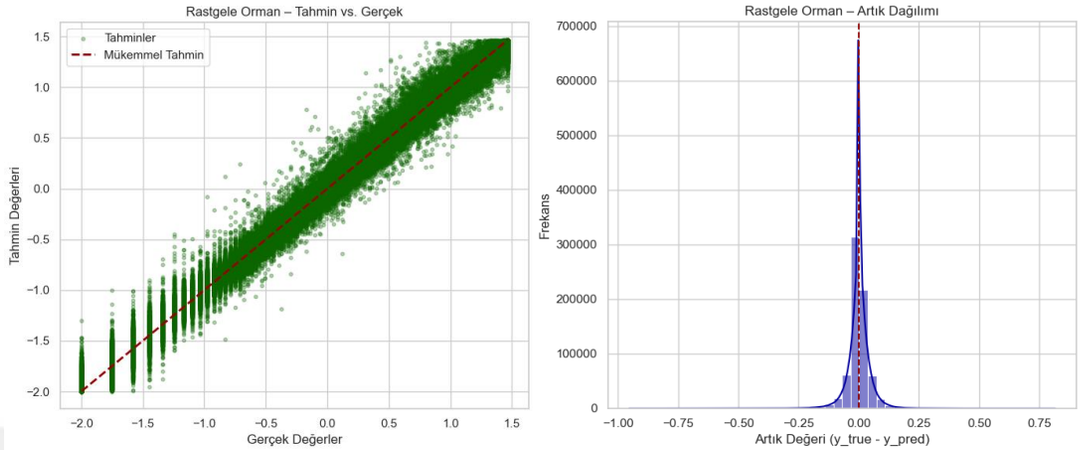


Şekil 120. k-En Yakın Komşu Dağılım ve Artık Değer Grafiği

Şekil 120'de sol tarafta bulunan dağılım grafiğinde, tıpkı karar ağaçları modelinde olduğu gibi, tahminlerin çoğunlukla y=x çizgisi boyunca hizalandığı görülmektedir. -2,0 ile -1,0 arasındaki kesikli bölgeler, modelin seyrek verilerdeki performans düşüklüğünü göstermektedir. Model, yüksek veri yoğunluğuna sahip bölgelerde (-0,5-1,5) oldukça iyi bir performans göstermesine

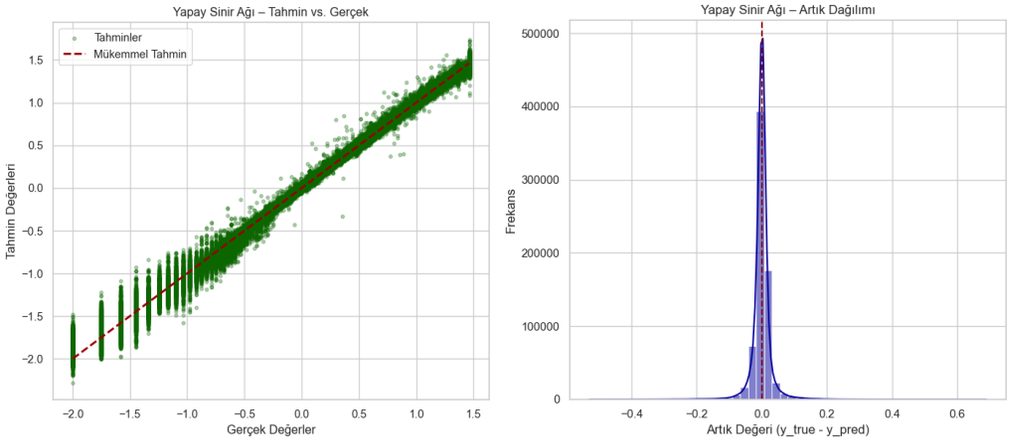
[221] Residual

rağmen, düşük frekanslı gözlemlerde performanslı değildir. Artık değer dağılımını gösteren sağ taraftaki grafikte ise, tıpkı karar ağaçları modelinde olduğu gibi artık değerler sıfıra yakın ve keskin bir şekilde tepe oluşturmuştur.



Şekil 121. Rastgele Orman Dağılım ve Artık Değer Grafiği

Şekil 121'de sol tarafta bulunan dağılım grafiği, rastgele orman modelinin dağılımını göstermektedir. Tahminlerin kırmızı kesikli çizgi boyunca yoğun ve dar bir yapı oluşturduğu görülebilir. Bu ise gerçek ve tahmin edilen değerler arasında güçlü bir uyumu işaret etmektedir. Başka bir deyişle, gerçek değerler ile tahmin edilen değerlerin yoğunluğu, düşük varyans ve düşük önyargıyı göstermektedir. Artık değer dağılımını gösteren sağ taraftaki grafikte ise, diğer iki modelde olduğu gibi artık değerlerin yüksek ve dar bir tepe ile sıfıra yakın bir şekilde toplandığı görülmektedir. Dağılım düşük önyargı ve dengeli hata oranını işaret etmektedir.

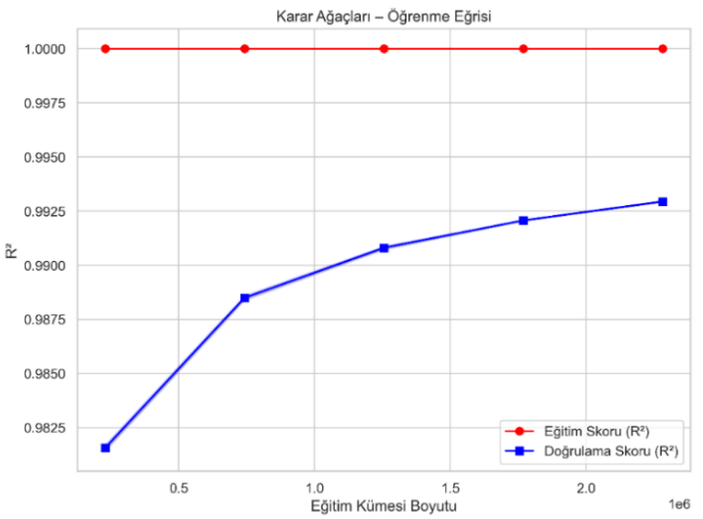


Şekil 122. Yapay Sinir Ağı Dağılım ve Artık Değer Grafiği

Şekil 122'de sol tarafta bulunan dağılım grafiğinde ise, yapay sinir ağı modelinin dört model arasında neden en başarılı model olduğu görülebilmektedir. Modelin tahminleri, minimum dağılımla y=x çizgisi boyunca hizalanmıştır. Tahminlerin mükemmel tahmin çizgisi boyunca kümelenmesi, modelin yüksek tahmin doğruluğuna ve mükemmel bir genellemeye sahip

olduğunu göstermektedir. Artık değer dağılımını gösteren sağ taraftaki grafikte ise, artıklar sıfır etrafında çok dar bir şekilde toplanmış olup keskin ve simetrik bir tepe oluşturmuştur.

Modellerin performansını ölçmek için ayrıca öğrenme eğrileri oluşturulmuş ve değerlendirilmiştir. Buradaki temel amaç ise aşırı öğrenme veya eksik öğrenme durumları ile ideal öğrenmenin gerçekleşip gerçekleşmediğinin ölçülmesidir. Öğrenme eğrisi[222], eğitim seti boyutu arttıkça modelin performansını göstermek ve veriden artarak öğrenen algoritmaları teşhis etmek için kullanılmaktadır (Meek, Thiesson ve Heckerman, 2002). Öğrenme eğrileri oluşturulurken bir diğer amaç ise modellerin standart sapmasının da hesaplanarak eğrilerin ne kadar gölge oluşturdukları veya oluşturmadıklarını tespit etmektir. Öğrenme eğrilerinde, eğitim boyutu arttıkça model performansının daha isabetli olarak tahminler üretebilmesi için çapraz doğrulama kullanılmaktadır (Mohr ve van Rijn, 2023). cv=5 seçilerek uygulanan öğrenme eğrileri, veri bölümlemesinin neden olduğu varyansın azalmasını sağlamaktadır.

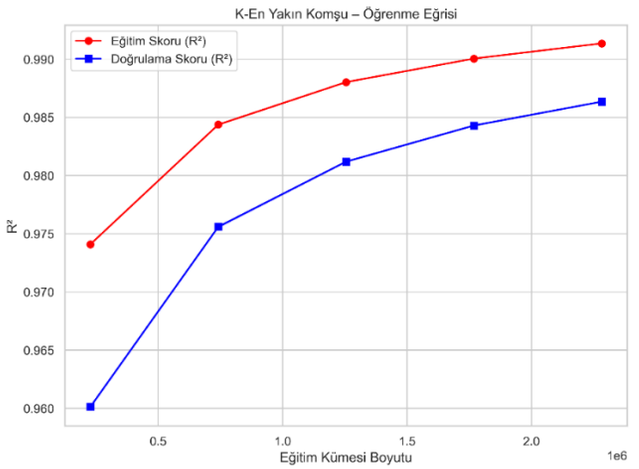


Şekil 123. Karar Ağaçları Öğrenme Eğrisi

Şekil 123'te, karar ağaçları modelinin öğrenme eğrisi görülmektedir. Karar ağaçlarında genellikle, belirli durumlarda (budama yapılmamış model, varsayılan değerlerle seçilen DecisionTreeRegressor vb.) her yaprak saf olana kadar büyür (Bishnoi ve Hooda, 2022). Çünkü ağaç, her örnek için bir bölme oluşturarak eğitim verilerini ezberleyebilir ve bu da sıfır eğitim hatası ile ve $R^2=1$'e yol açabilir. Şekilde görülen eğitim doğruluğu da 1 seviyelerinde görülmektedir. Doğrulama doğruluğu ile eğitim doğruluğu arasında makasın (~0,0075) birbirine çok yakın olması, modelin oldukça iyi bir genelleme yeteneğine sahip olduğunu ve aşırı öğrenmeye uğramadığını göstermektedir. Şekilde eğitim boyutu büyüdükçe (~0,9815'ten

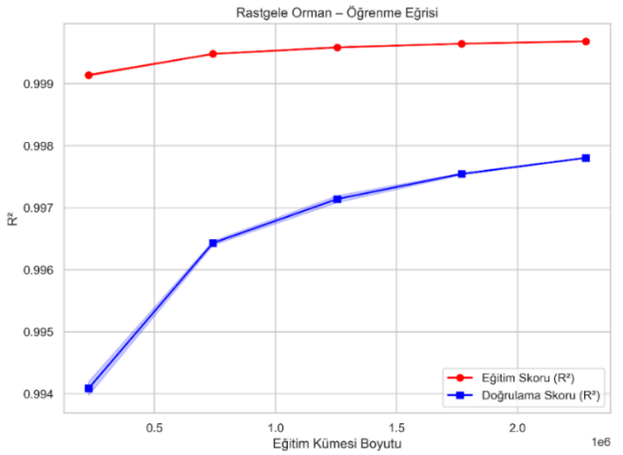
[222] Learning curve

~0,9930'a) R^2 değerinin yükseldiği görülebilir. Şekilde ayrıca kırmızı eğride hiç gölgenin olmaması standart sapmanın olmadığını da göstermektedir. Ancak mavi eğride, eğitimin başlarından itibaren oldukça az bir gölgenin görülmesi ve eğitimin ortalarından itibaren kaybolması, çok küçük bir standart sapmanın olduğunu ve eğitim seti büyüdükçe kaybolduğunu göstermektedir. Başka bir deyişle şekil, modelin deterministik olduğunu, nerdeyse hiç varyansın bulunmadığı ve doğrulama performansının stabil olduğunu göstermektedir.



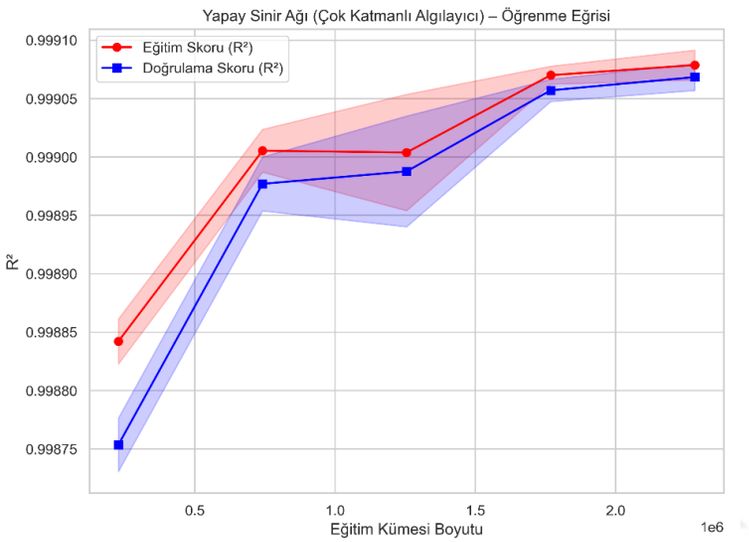
Şekil 124. k-En Yakın Komşu Öğrenme Eğrisi

Şekil 124'te k-en yakın komşu modelinin öğrenme eğrisi görülmektedir. Eğitim skorunun ~0,974 seviyesinden başladığı ve eğitim seti boyutu arttıkça yükseldiği görülmektedir. Eğitim setinin sonlarında ise ~0,991 skoruna erişmiştir. Doğrulama skoru ise 0,960 civarında başlamış ve ~0,986'ya kadar çıkmıştır. Eğitim ve doğrulama skorları arasındaki küçük fark (~0,015), daha fazla veri kullanıldıkça gitgide küçülmüştür (~0,005). Bu durum ise modelin anlamlı ve genelleştirilebilir bir özelliğe sahip olduğunu göstermektedir.



Şekil 125. Rastgele Orman Öğrenme Eğrisi

Şekil 125'te, rastgele orman modeline ait öğrenme eğrisi görülmektedir. Şekilde, eğitim skorunun ~0,999 seviyesinde başladığı ve eğitim setinin boyutu arttıkça neredeyse 0,9998'e ulaştığı görülmektedir. Doğrulama skoru da benzer bir örüntüyü takip etmiş ve ~0,994'ten ~0,998'e yükselmiştir. Tıpkı bir önceki modelde olduğu gibi, eğitim ve doğrulama skorları arasındaki fark neredeyse 0'dır (~0,0019) ve her iki eğri de oldukça sabit özellik göstermektedir. Doğrulama skorunda, başlarda hafif bir gölgenin görülmesi, varyansı işaret etmektedir ancak bu varyans önemsiz olarak görülebilir. Veri seti boyutu arttıkça da gölge kaybolmuştur.



Şekil 126. Yapay Sinir Ağı Öğrenme Eğrisi

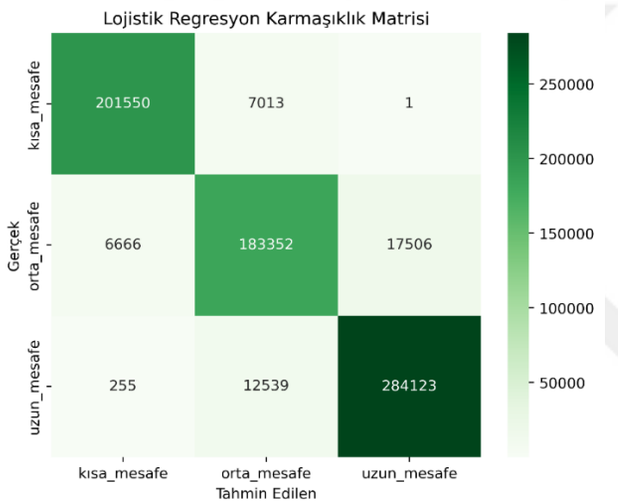
Şekil 126'da ise yapay sinir ağı[223] modeline ait öğrenme eğrisi görülmektedir. Şekilde, eğitim skorunun ~0,99885'ten başladığı ve veri seti boyutu arttıkça ~0.99908'e kadar çıktığı görülmektedir. Doğrulama skoru ise ~0,99875'ten başlayıp ~0,99906'ya ulaşmıştır. Görüldüğü üzere, eğitim ve doğrulama eğrileri arasında neredeyse hiç boşluk bulunmamaktadır. Bu durum, modelin neredeyse mükemmel bir genellemeye sahip olduğunu göstermektedir. Her iki eğride de varyansların diğer modellere göre daha fazla olduğu söylenebilir. Ayrıca, eğitimin başlamasından kısa bir süre sonra üst üste binmiş görünen kırmızı ve mavi gölgeler (standart sapma), model performansının veri bölmelerine duyarlı olmadığını göstermektedir.

4.3.1.2. Sınıflandırma Analizleri

Regresyon analizlerinden sonra sınıflandırma işlemlerine başlanmıştır. Sınıflandırma görevi için mesafe_km değişkeni kendi içerisinde kısa, orta ve uzun mesafe olarak 3 ayrı alt sınıfa ayrılmış ve ilk model, lojistik regresyon kullanılarak analiz edilmiştir. Model oluşturulurken

[223] Çok Katmanlı Algılayıcı

mesafe_km ile mesafe_km_kategori değişkenleri, hedef değişkenler olduğundan dolayı sınıflandırıcı modelinden çıkarılmıştır. Ayrıca, modele anlamlı bir şekilde katkıda bulunamayabilecek, aşırı öğrenmeye yol açabilecek, boyutu yükseltebilecek ve yoğun kardinalite gösteren rota_eyaletlerarasi ve rota_sehirlerarasi değişkenleri ile tanımlayıcı değişkenler (kalkis_havaalani_kodu ve varis_havaalani_kodu) de çıkarılmıştır. Modelde bulunan kategorik özellikler (kalkis_sehri, kalkis_eyaleti, varis_sehri, varis_eyaleti) OneHotEncoder kullanılarak kodlanmıştır. Veri seti, train_test_split fonksiyonu aracılığıyla katmanlı bir şekilde %80-%20 yaklaşımı ile bölünmüştür. Model, Softmax kayıp fonksiyonuna sahip "lbfgs" çözücüsü ve "max_iter=5000" parametreleri kullanılarak eğitilmiştir. Eğitim tamamlandıktan sonra model; karmaşıklık matrisi, sınıflandırma raporu, öğrenme eğrisi ve AUC-ROC eğrisi bazında değerlendirilmiştir.



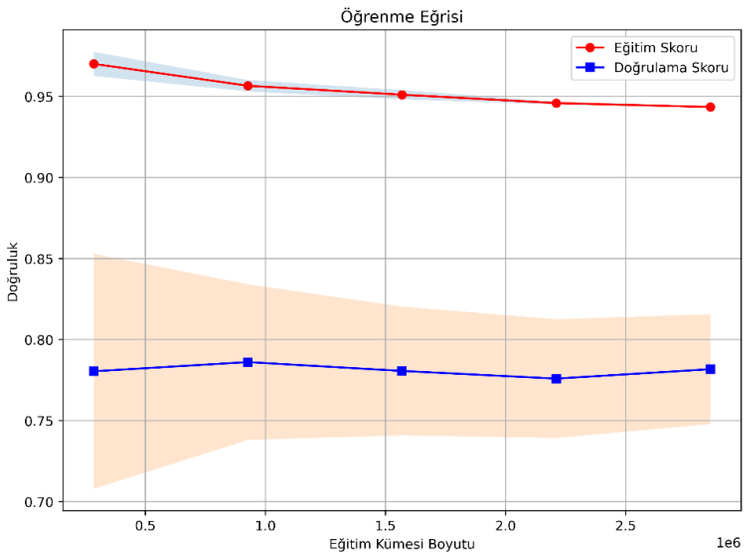
Şekil 127. Lojistik Regresyon Sınıflandırma Karmaşıklık Matrisi

Şekil 127, mesafe_km değişkeninin 3 sınıflı karmaşıklık matrisini göstermektedir. 201.550 kısa_mesafe, 183.352 orta_mesafe ve 284.123 uzun_mesafe örneklerinin doğru bir şekilde sınıflandırıldığı görülebilir. Model, sınıfların ayrımını yüksek seviyede öğrenme yeteneğine sahip olup, özellikle uzun mesafeli uçuşları doğru bir şekilde tahmin etmede oldukça başarılıdır. Modelin az sayıda yanlış sınıflandırma yaptığı da görülmektedir. Örneğin; 7013 kısa mesafeli uçuş, orta mesafeli olarak sınıflandırılmıştır. 17.506 uçuş ise gerçekte orta mesafe olarak sınıflandırılması gerekirken uzun mesafe olarak sınıflandırılmıştır. Bu tür hatalar, örüntüde birbirine yakın sınıflarda bir karışıklık yapıldığını göstermektedir. En az yanlış sınıflandırma olarak sadece 1 uçuşun gerçekte kısa mesafe olarak sınıflandırılması gerekirken uzun mesafe şeklinde sınıflandırıldığı görülebilir.

Tablo 26. Lojistik Regresyon Modeli Sınıflandırma Raporu

	Kesinlik	Duyarlılık	F1	Örnek Sayısı
kısa_mesafe	0,9668	0,9664	0,9666	208.564
orta_mesafe	0,9036	0,8835	0,8935	207.524
uzun_mesafe	0,9420	0,9569	0,9494	296.917
Doğruluk			**0,9383**	713.005
Makro Ortalama	0,9375	0,9356	0,9365	713.005
Ağırlıklı Ortalama	0,9381	0,9383	0,9381	713.005

Tablo 26'da modelin sınıflandırma raporu görülmektedir. Her üç sınıfın da hemen hemen dengeli dağıldığı söylenebilir, sadece uzun mesafe sınıfı, dağılımda biraz daha baskın görünmektedir. Kısa mesafe sınıfı tüm metriklerde en yüksek performansı göstermiştir (kesinlik=0,9668, duyarlılık=0,9664, F1=0,9666). Uzun mesafeli uçuşların da 0,9420 kesinlik, 0,9569 duyarlılık ve 0,9494 F1 skorları ile başarılı bir şekilde sınıflandırıldığı görülebilir. Orta mesafeli uçuşlar ise diğer iki sınıfa göre daha düşük değerlendirme skorlarına sahiptir (kesinlik=0,9036, duyarlılık=0,8835, F1=0,8935). Modelin genel doğruluk skoru ise 0,9383 olup neredeyse her 100 uçuştan yaklaşık 94'ü doğru sınıfta tahmin edilmiştir. Modelde ayrıca logaritmik kayıp[224] metriği de hesaplanmıştır. Log kaybı, tahminlerin doğru olup olmadığı ile modelin ne kadar güvenilir olduğunun değerlendirilmesini sağlayan bir metriktir. Mükemmel bir modelin log kaybı 0'dır (Terven vd., 2025, s. 24). 0,1601'lik log kaybı skoru modelin doğru tahminlerin yanında doğru sınıflara yüksek olasılıkları atadığını açıklamaktadır.

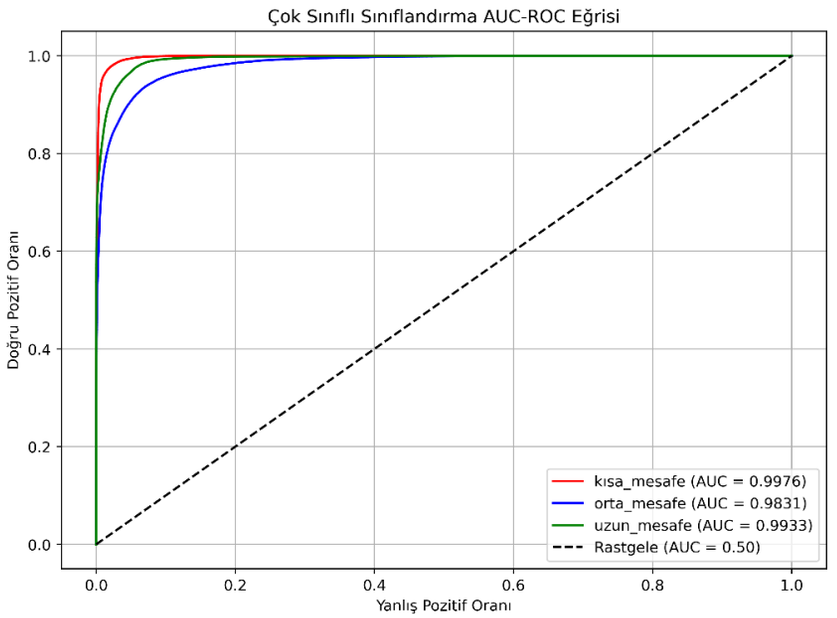


Şekil 128. Lojistik Regresyon Öğrenme Eğrisi

Şekil 128'de lojistik regresyon modelinin öğrenme eğrisi görülmektedir. cv=5 seçilerek uygulanan öğrenme eğrisi sonucunda eğitim doğruluk skorunun genellikle 0,94-0,96 aralığında

[224] Log loss

seyrettiği ve modelin eğitim verilerine çok iyi uyum sağladığı görülebilir. Doğrulama doğruluğu skorunun ise eğitim doğruluğu skoruna göre önemli ölçüde daha düşük olduğu görülmektedir (0,77-0,78). Eğitim kümesi boyutu arttıkça da bu fark hemen hemen aynı kalmıştır. Eğitim ve doğrulama doğruluğuna ait skorlar arasındaki bu fark, modelin aşırı öğrenme yaptığını göstermektedir. Başka bir deyişle model, eğitim verilerini çok iyi öğrenmiş ancak daha önceden hiç görmediği doğrulama verilerine eşit derecede genelleme yapamamıştır. Tablo 25'te belirtilen 0,9383'lük test doğruluğu ile öğrenme eğrisinde görülen aşırı öğrenme durumu bir karmaşıklığa yol açmaktadır. Buradaki temel ayrım, doğruluğun nasıl ölçüldüğü ile açıklanabilir. Modelin 0,9383'lük genel doğruluğu, tahmin edilen olasılıkların argmax fonksiyonu kullanılarak bütün test setinde hesaplanmıştır. Sınıflandırıcı, her bir test örneği için, uçuşun her üç sınıfa ait olma olasılığını hesapladıktan sonra, en yüksek olasılığa sahip sınıfı seçmektedir. Ardından doğru yapılan tahminleri toplam tahminlere böldükten sonra bir doğruluk skoru verir. Öğrenme eğrisinde ise tek bir test seti doğruluğu yoktur. Eğitim setinin farklı bölümlerinde, çapraz doğrulama[225] kullanılarak birden fazla eğitim ve doğrulama adımı gerçekleştirildiğinden, sınıfların ne kadar iyi genelleştirildiği detaylı olarak ölçülebilir. Bu nedenle, model test setinde çok iyi performans gösterse de, öğrenme eğrisi aşırı öğrenmenin tespit edilmesine yardımcı olur. Aşırı veya eksik öğrenme, istenilen bir durum olmadığı için, çeşitli teknikler kullanılarak modelin hem eğitim hem de doğrulama performansı iyileştirilmelidir.



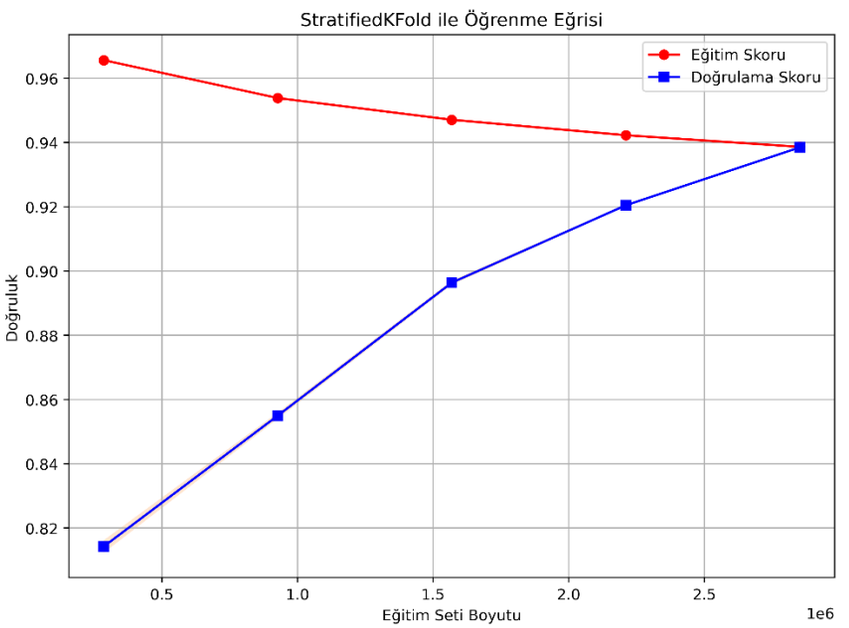
Şekil 129. Lojistik Regresyon AUC-ROC Eğrisi

[225] Cross validation

Şekil 129'da modele ait AUC-ROC eğrisi gösterilmektedir. Grafikte kısa mesafe sınıfına ait eğrinin (AUC=0,9976), diğer iki sınıf ile neredeyse tam bir ayrıma sahip olduğu görülebilir. Başka bir deyişle model, bu sınıfta neredeyse hiç yanlış pozitif veya yanlış negatif sınıflandırma yapmamıştır. Orta mesafe sınıfına ait eğri (AUC=0,9831) ise diğer iki sınıf ile daha fazla örtüşme olduğunu göstermektedir. Uzun mesafe sınıfına ait eğri (AUC=0,9933) ise, yüksek ayrımı göstermekle birlikte modelin uzun mesafeli uçuşları minimum hata ile belirleyebileceğini ifade etmektedir.

Modeldeki aşırı öğrenme sorununu gidermek için Stratified k-Fold tekniği kullanılmıştır. Bu teknik, çoğunlukla sınıflandırma problemlerinde kullanılmakta olup, katmanlı (tabakalı) çapraz doğrulama ile veri setini benzer büyüklükteki k gruplara böler. Teknik, her katmanın tüm veri setiyle aynı sınıf dağılımını korumasını sağlamaktadır (Marcot ve Hanea, 2021; Prusty, Patnaik ve Dash, 2022). Örneğin; Widodo, Brawijaya ve Samudi (2022), servikal kanser tahmini için dengesiz bir veri kümesinde sınıflandırma algoritmalarını değerlendirmek üzere Stratified k-Fold tekniğini uygulamış ve model doğruluğunun değişmeden aşırı öğrenme probleminin giderildiğini tespit etmişlerdir. Yani teknik, modelin tahmin performansını doğrudan etkilemeden değerlendirme sürecini iyileştirmektedir.

Teknik, k=5 seçilerek eğitim seti üzerinde uygulanmış ve aşırı öğrenme problemi giderilmiştir. Modelin doğruluk skoru aynı kalmış, eğitim ve doğrulama doğruluğu arasındaki fark ise kapanmıştır.



Şekil 130. Stratified k-Fold Lojistik Regresyon Öğrenme Eğrisi

Şekil 130'da Stratified k-Fold tekniği uygulanmış lojistik regresyon öğrenme eğrisi gösterilmektedir. Eğitimin başlarında eğitim ve doğrulama skorları arasında belirgin bir boşluk görülmektedir (eğitim skoru=~%96,5, doğrulama skoru=~%81,5). Fakat eğitim seti boyutu arttıkça eğitim doğruluğu beklendiği gibi az bir miktar azalmış ve doğrulama doğruluğu yükselerek eğitim doğruluğuna yaklaşmıştır. Grafikte, tüm eğitim setinde ise her iki eğrinin de yaklaşık 0,94'te birleştiği görülebilir. Yani, modelin aşırı öğrenme sorunu giderilmiştir.

Sınıflar, son olarak dört farklı model (rastgele orman, yığınlama, lojistik regresyon, Naive Bayes) üzerinde analiz edilmiş ve sonuçları karşılaştırılmıştır. Yığınlama[226], modellerin tahmin performansını iyileştirmek için birden fazla temel modeli[227] ve bir meta modeli[228] birleştiren bir topluluk öğrenme tekniğidir (Alexandropoulos vd., 2019). Teknikte, öncelikle seviye 0 modeller eğitilir ve bu modellerin eğitiminden elde edilen veriler ile, doğrulama seti üzerinde çapraz doğrulama yapılarak tahminlerde bulunulur. Son olarak seviye 1 modeller meta[229] model olarak eğitilir ve seviye 0 modellerden gelen tahminler, meta modelde girdi olarak kullanılır. Böylece farklı modellerin güçlerinin birleştirilmesiyle hem model önyargısı hem de varyansın azaltılması sağlanır (Ahrens vd., 2022). Yığınlama modelinde kullanılan seviye 0 modeller; lojistik regresyon, karar ağaçları, k-en yakın komşu ve çok katmanlı algılayıcıdır. Meta model olarak ise hist gradient boosting[230] sınıflandırıcısı kullanılmış ve cv=5 seçilerek modeller çalıştırılmıştır.

Tablo 27. Sınıflandırma Modellerinin Karşılaştırılması

Model	Doğruluk	Kesinlik	Duyarlılık	F1	Log Kaybı
Yığınlama	0,9990	0,9990	0,9990	0,9990	0,0046
Rastgele Orman	0,9983	0,9983	0,9982	0,9983	0,0303
Lojistik Regresyon	0,9385	0,9376	0,9358	0,9366	0,1600
Naive Bayes	0,4996	0,5656	0,4362	0,3962	17,8613

Tablo 27'de modellerin değerlendirme metrikleri ile sonuçları gösterilmektedir. Yığınlama modeli, en başarı model olarak göze çarpmaktadır. Doğruluk skoru (0,9990), modelin diğer metriklerinin skorlarıyla aynıdır. Yani, modelin hemen hemen mükemmel bir sınıflandırma yaptığı söylenebilir. Ayrıca oldukça düşük model log kaybı (0,0046), oldukça iyi olasılıksal tahmini ve dengelemeyi göstermektedir. Rastgele orman sınıflandırıcısı da yığınlama modeline oldukça yakın doğruluk (0,9983), kesinlik (0,9983), duyarlılık (0,9982) ve F1 (0,9983) skorları ile oldukça başarılıdır. Skorların hemen hemen birbirine eşit olması, oldukça bir istikrarlı

[226] Stacking
[227] Seviye 0 modeli olarak adlandırılır
[228] Seviye 1 modeli olarak adlandırılır
[229] Üst
[230] Özellikle büyük veri kümeleri için hesaplama verimliliği sağlayan, gradyan artırmanın optimize edilmiş versiyonudur. Teknik, sürekli özellikleri tamsayı bölmelere ayırmakta ve böylece RAM kullanımını azaltılmaktadır.

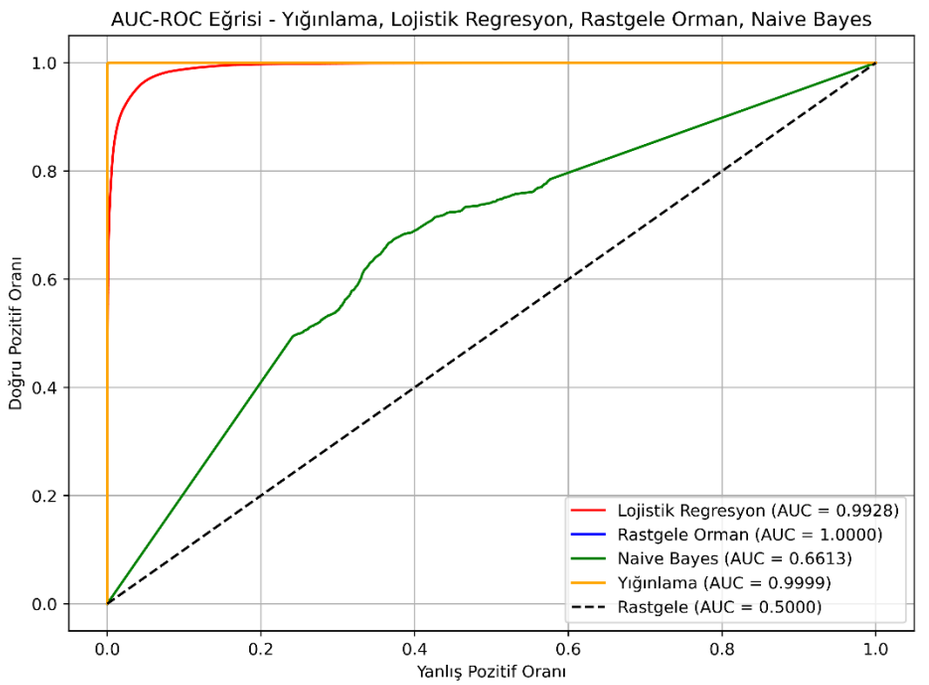
tahmin gücünü göstermektedir. Logaritma kaybı, yığınlama modeline göre yüksek (0,0303) görünmektedir. Ancak genel olarak değerlendirildiğinde düşük kalmaktadır. Lojistik regresyon sınıflandırıcısı daha önceden elde edilen 0,9385'lik doğruluk skoru ile diğer iki modele göre biraz daha düşük performans göstermiştir (kesinlik=0,9376, duyarlılık=0,9358, F1= 0,9366). Logaritma kaybı (0,1600) ise diğer iki modele göre oldukça yüksek kalmaktadır. Naive Bayes sınıflandırıcısı ise, bu modeller içinde en kötü performansı göstermektedir. Yalnızca 0,4996'lık doğruluk skoru ile tıpkı bir "rastgele tahminci" gibi davranmış ve kesinlik (0,5656), duyarlılık (0,4362) ve F1 (0,3962) skorları ile oldukça başarısız bir model olarak göze çarpmıştır. Modelin log kaybı (17.8613) oldukça yüksektir.



Şekil 131. Sınıflandırma Modelleri Karmaşıklık Matrisi

Şekil 131'de dört farklı sınıflandırma modelinin karmaşıklık matrisi gösterilmektedir. Rastgele orman sınıflandırıcısı, her üç sınıfta da oldukça başarılı olmuştur. Modelin, oldukça az sayıda yanlış sınıflandırma yaptığı görülmektedir. Örneğin; kısa mesafe sınıfında yalnızca 222 adet yanlış sınıflandırma bulunmaktadır. Orta mesafe sınıfında ise 164 kısa ve 540 uzun mesafe olmak üzere toplamda 704 yanlış sınıflandırma vardır. Uzun mesafe sınıfında toplamda 283 yanlış sınıflandırılma görülmektedir. Yanlış sınıflandırmalar, test setinin boyutu göz önüne alındığında nerdeyse önemsiz olarak yorumlanabilir. Yığınlama sınıflandırıcı ise, rastgele orman sınıflandırıcısından da iyi bir performans göstermiştir. Model; kısa mesafe sınıfında 130, orta mesafe sınıfında 338 ve uzun mesafe sınıfında da 221 yanlış sınıflandırma yaparak,

mnimum hata ile en iyi performansı göstermiştir. Model, topluluk yöntemleri ile birden fazla modelin gücünden yararlandığı için en yüksek sınıflandırma kalitesine ulaşmıştır. Lojistik regresyon sınıflandırıcısında, ilk modelden farklı olarak en iyi C[231] parametresi[232] RandomSearchCV ile aranmış ve ayrıca "max_iter=15000" ve "solver=saga" seçilerek model çalıştırılmıştır. İlk modele göre oldukça az bir iyileşme görülmüştür. Naive Bayes sınıflandırıcısı ise performans açısından en zayıf model olarak göze çarpmaktadır. Örneğin; kısa mesafe sınıfında sadece 34.342 doğru sınıflandırılmış görünmektedir. Aynı sınıfa ait yanlış sınıflandırmalar ise 50.943 orta mesafe ve 123.279 uzun mesafe olarak görülmektedir. Naive Bayes, özellikler arasında bağımsızlığı varsaydığından, karmaşık sınıfları modelleyememiş ve bu da özellikle kısa ve orta mesafeler için yüksek yanlış sınıflandırma sayılarına yol açmıştır.

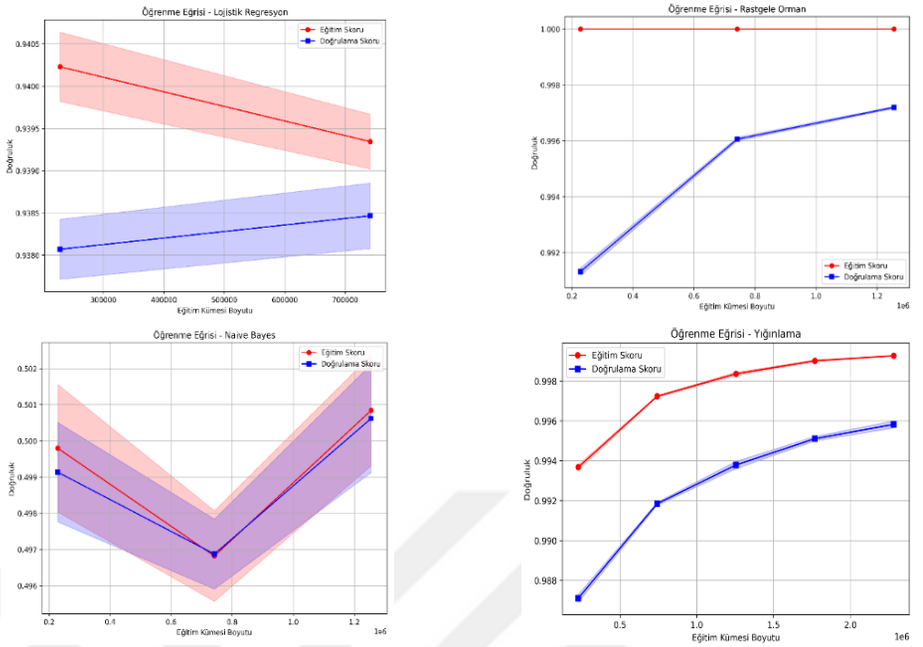


Şekil 132. Sınıflandırma Modelleri AUC-ROC Eğrisi

Şekil 132'de sınıflandırma modellerine ait AUC-ROC eğrisi gösterilmektedir. Yığınlama ve rastgele orman modelleri, sırasıyla 0,9999 ve 1,0000 AUC skorları elde ederek en başarılı modeller olmuştur. Rastgele orman eğrisi, hemen hemen aynı skorlara sahip olmasından dolayı yığınlama eğrisi ile üst üste binmiş görünmektedir. Lojistik regresyon modeli ise 0,9928'lik AUC skoru ile oldukça iyi bir performansa sahiptir. Naive Bayes modeli ise 0,6613'lük oldukça düşük bir AUC skoru ile sınıflar arasında ayrım yapma yeteneğinin sınırlı olduğunu

[231] Aşırı öğrenmeyi önlemek için modelin katsayılarına uygulanan cezanın derecesini kontrol eden düzenleme parametresi.
[232] 166.81005372000558

göstermektedir. Diğer şekil ve grafiklerde olduğu gibi, topluluk modellerinin (yığınlama ve rastgele orman) en etkili modeller olduğu görülebilmektedir.



Şekil 133. Topluluk Modelleri ve Temel Modellerin Öğrenme Eğrileri

Şekil 133'te ise dört sınıflandırma modeline ait öğrenme eğrileri gösterilmektedir. Rastgele orman sınıflandırıcı modeli, tüm eğitim seti boyutlarında en yüksek eğitim (~1,0) ve doğrulama doğruluğu (0,9983) skoru yakalamıştır. Başka bir deyişle, modelin genelleme özelliğinin oldukça başarılı olduğu ve düşük önyargı ile düşük varyans barındırdığı söylenebilir. Eğitim kümesinin boyutu arttıkça eğitim ve doğrulama performansı arasındaki boşluğun yakınlaşmaya başladığı görülmektedir. Yani model, daha fazla veriden faydalandığında daha iyi ölçeklenmektedir. Topluluk öğrenmesi yöntemlerinden yığınlama[233] modeli ise hemen hemen rastegele orman sınıflandırıcısına yakın bir doğrulama performansı göstermektedir. Oldukça yüksek bir eğitim performansı (0,9990) ve doğrulama doğruluğu (0,9960) yakalamıştır. Rastegele orman modelinde olduğu gibi, daha fazla eğitim verisi ile daha yüksek doğrulama ve eğitim skoruna ulaşılmıştır. Modelin etkili bir şekilde genelleştirebildiğini söylemek mümkündür. Modelde önyargı ve varyans dengesi de bu özellikten dolayı dengede görülmektedir. Lojistik regresyon ise, diğer iki modele göre biraz daha düşük eğitim (0,9394) ve doğrulama (0,9385) doğruluğuna sahiptir. Eğitim kümesi boyutu arttıkça eğitim skoru düşmeye ve doğrulama skoru yükselmeye başlamaktadır. Ayrıca, eğriler arasındaki açıklık eğitim kümesi boyunca kapanmaktadır. Naive Bayes sınıflandırıcı modeli ise dört model içinde en zayıf genel performansı göstermektedir. Düşük eğitim (0,501) ve doğrulama (0,501)

[233] Stacking

doğruluğuna sahip olan modelde hem eğitim hem de doğrulama eğrileri, eğitim kümesi boyutu artsa da değişkenlik göstermemektedir. Naive Bayes algoritması, öznitelikler arasında güçlü bir bağımsızlık varsaydığından dolayı, veri seti 1 için yetersiz bir modeldir çünkü veri setinde çoğu öznitelik birbiriyle ilişkilidir. Dolaysıyla, model bu ilişkilerdeki örüntüyü iyi öğrenememiş ve böylece yüksek önyargı özelliği göstermiştir.

Bir diğer sınıflandırma görevi için ucus_basina_yolcu_sayisi değişkeni, kendi içerisinde "çok düşük", "düşük", "orta", "yüksek" ve "çok yüksek" olarak beş ayrı alt sınıfa ayrılmış ve rastgele orman algoritması kullanılarak analiz edilmiştir. Sınıf dağımlarına ilişkin tablo, aşağıda gösterilmektedir.

Tablo 28. Uçuş Başına Yolcu Sayısı Sınıf Dağılımları

No	Sınıf	Dağılım
0	Çok Düşük	734.769
1	Düşük	704.461
2	Orta	714.445
3	Yüksek	705.069
4	Çok Yüksek	706.281

Tablo 28'den de görüldüğü üzere, sınıf dağılımları oldukça dengeli görünmektedir. Bu durum, hesaplama süreçlerinde standardizasyonu sağlamıştır. Makine öğrenmesinde algoritmalar, kategorik veriler üzerinde doğrudan çalışmadığından, tüm kullanılabilir kategorik özellikler (kalkis_havaalani_kodu, kalkis_sehri, kalkis_eyaleti, varis_havaalani_kodu, varis_sehri, varis_eyaleti, mesafe_km_kategori, rota_eyaletlerarasi) sayısal değerlere dönüştürülmüştür. Dönüştürme işlemi, one hot encoding tekniğiyle yapılmış ve bu işlem sonucunda tüm öznitelik sayısı 24'ten 4093'e çıkmıştır. Toplam boyut ise (3.565.025, 24)'ten (3.565.025, 4.093)'e yükselmiştir. 4093 öznitelik, hesaplama karmaşıklığını ve maliyetini artıracağından, hem öznitelikleri azaltmak hem de çoklu doğrusallığı[234] ortadan kaldırmak için temel bileşen analizi[235] uygulanmıştır. Bileşen sayısı, açıklanan varyansın %95'ini korumak için otomatik olarak seçilmiş ve işlem sonrasında boyut, 4093'ten 62'ye düşmüştür. Analiz sonuçlarına göre, sonuç matrisi X_pca, minimum bilgi kaybıyla veri setinin varyansının çoğunu korumuş ve hesaplama verimliliğini önemli ölçüde iyileştirmiştir. Ayrıca, one-hot encoding tekniği ile kodlanan kategorik veriler ile diğer sayısal değerleri içeren dataframe, hesaplama maliyetinin azaltılması için float64'ten float32'ye çevrilmiş ve dataframe, Numpy dizisine dönüştürülmüştür.

[234] Multicollinearity
[235] Principal Component Analysis - PCA

Çok sınıflı sınıflandırma görevi için ilk olarak, rastgele orman sınıflandırıcısı seçilmiş ve bu sınıflandırıcıda kullanılmak üzere hiperparametre optimizasyonu için RandomizedSearchCV tekniği tüm veri setine uygulanmıştır. Yapılan ilk denemelerde daha yüksek k değeri ve daha yüksek iterasyon değerleri, RAM hatalarına neden olmuş ve bu yüzden arama işlemi, 3 katlı çapraz doğrulama ve 20 iterasyon ile gerçekleştirilebilmiştir. Arama sonucunda bulunan en iyi değerler şu şekilde gösterilebilir:

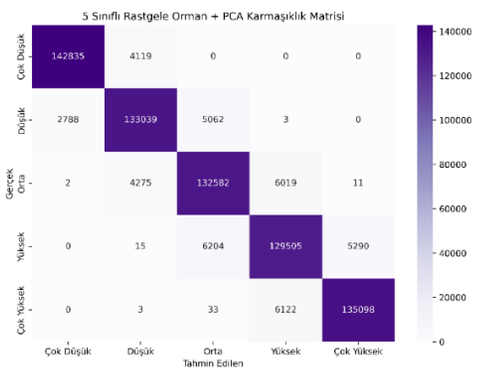
- n_estimators: 200
- min_samples_split: 2
- min_samples_leaf: 1
- max_depth: None
- bootstrap: False

Bulunan en iyi hiperparametreler ile oluşturulan model test edilmiştir.

Tablo 29. PCA Destekli 5 Sınıflı Rastgele Orman Modeli Sınıflandırma Raporu

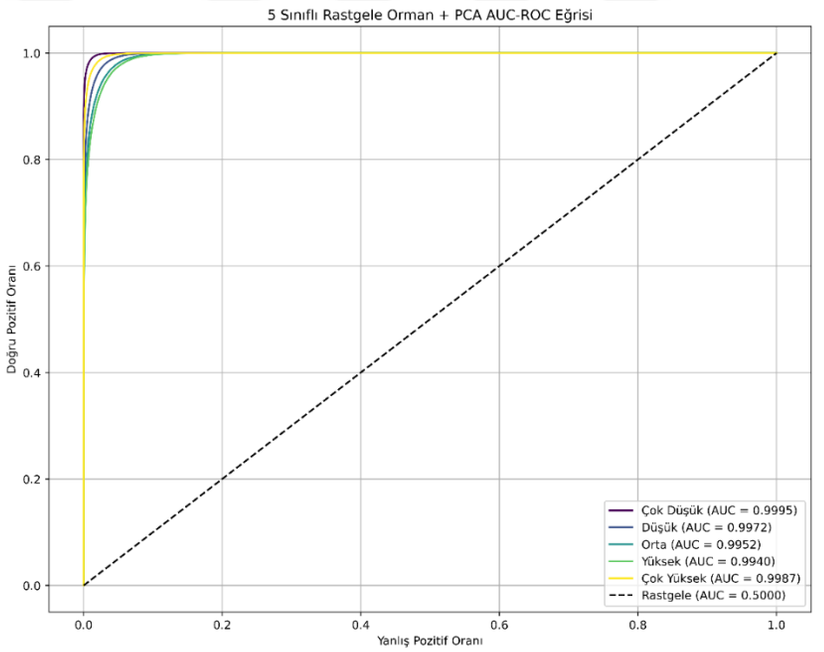
	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
Çok Düşük	0,9808	0,9720	0,9764	146.954
Düşük	0,9405	0,9443	0,9424	140.892
Orta	0,9215	0,9279	0,9247	142.889
Yüksek	0,9143	0,9184	0,9163	141.014
Çok Yüksek	0,9622	0,9564	0,9593	141.256
Doğruluk			**0,9440**	713.005
Makro Ortalama	0,9439	0,9438	0,9438	713.005
Ağırlıklı Ortalama	0,9441	0,9440	0,9440	713.005

Tablo 29'da, rastgele orman sınıflandırıcısının sınıflandırma raporu gösterilmektedir. Tabloya göre, test setinde (%20) yapılan toplam tahminlerin %94,40'ı doğrudur. Tüm sınıflar genelinde hem makro hem de ağırlıklı ortalama skorları, sınıf dağılımımın da dengeli olmasından dolayı, modelin genelleştirilebilir bir performansa sahip olduğunu göstermektedir. Özellikle "çok düşük" ve "çok yüksek" sınıfları yüksek F1 skorlarına (~0,96-0,98) sahiptir. "Yüksek" sınıfı, tüm sınıflar arasında en düşük F1 puanına (0,9163) sahip olsa da yine de kabul edilebilir seviyede olduğu söylenebilir.



Şekil 134. PCA Destekli 5 Sınıflı Rastgele Orman Modeli Karmaşıklık Matrisi

Şekil 134'te ise modele ait karmaşıklık matrisi gösterilmektedir. Matrise bakıldığında, sınıfların büyük çoğunlukla doğru tahmin edildiği görülmektedir. "Çok düşük", "düşük" ve "çok yüksek" sınıfları, yüksek doğrulukla sınıflandırılmıştır. "Yüksek" sınıfında olması gereken 5290 veri, "çok yüksek" olarak sınıflandırılmıştır. "Orta" sınıfından 4275 veri ise "düşük" olarak sınıflandırılmıştır. Yanlış sınıflandırmalar, modelin yeterli performansı gösterememesinden dolayı değil, gerçek dünya verilerinin olması gerektiği gibi dalgalı ve birbirine yakın olmasından dolayıdır. Ayrıca, aralığı birbirine yakın sınıflarda yanlış tahmin sayılarının nispeten yüksek olduğu söylenebilir. Matriste gözlemlenen bir başka önemli nokta ise aşırı uçlar arasında dikkate değer yanlış sınıflandırmaların olmamasıdır. Örneğin, "çok düşük" sınıfı hiçbir zaman "yüksek" ve "çok yüksek" olarak tahmin edilmemiştir. Bu da, temel bileşen analizinin, boyutları oldukça indirgemesine rağmen, sınıfların ayrımını iyi koruduğunu göstermektedir.



Şekil 135. PCA Destekli 5 Sınıflı Rastgele Orman Modeli AUC-ROC Eğrisi

Şekil 135'te ise sınıflandırma işlemine dair modelin AUC-ROC eğrisi gösterilmektedir. Şekil incelendiğinde, tüm sınıfların 0,99'dan büyük AUC skorları elde ettiği görülmektedir. AUC skoru 1'e ne kadar yakınsa o kadar güçlü sınıf ayrılabilirliğini temsil eder. Başka bir deyişle tüm tahminlerde düşük yanlış pozitif oranlarını göstermektedir. "Çok düşük" sınıfı 0,9995 AUC skoru ile en yüksek başarımı gösterirken, "yüksek" sınıfı ise 0,9940'lık bir skor yakalamıştır. Yüksek skorlar modelin birbirine çok yakın sınıf aralıklarında bile dengeli ve sabit bir performans sürdürdüğünü göstermektedir. Düzlemi ortadan ikiye ayıran siyah kesikli çizgi, rastgele bir sınıflandırıcıyı (AUC=0,5000) temsil etmektedir. Tüm eğrilerin, bu çizginin

oldukça üzerinde ve 1'e yakın olması, modelin oldukça iyi bir tahmin performansına sahip olduğunu göstermektedir.

Sınıflandırma işleminde ikinci olarak karar ağaçları sınıflandırıcısı seçilmiş ve bu sınıflandırıcıda kullanılmak üzere hiperparametre optimizasyonu için RandomizedSearchCV tekniği tüm veri setine uygulanmıştır. Arama işlemi, 3 katlı çapraz doğrulama ve 20 iterasyon ile gerçekleştirilmiştir. Arama sonucunda bulunan en iyi değerler şu şekilde gösterilebilir:

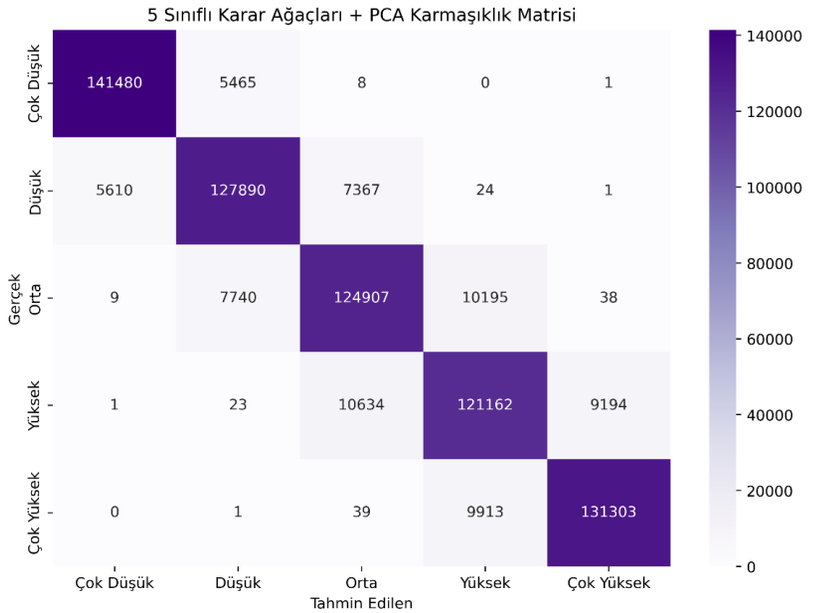
- min_samples_split: 10
- min_samples_leaf: 1
- max_depth: 40
- criterion: entropy

Bulunan en iyi hiperparametreler ile oluşturulan model test edilmiştir.

Tablo 30. PCA Destekli 5 Sınıflı Karar Ağaçları Modeli Sınıflandırma Raporu

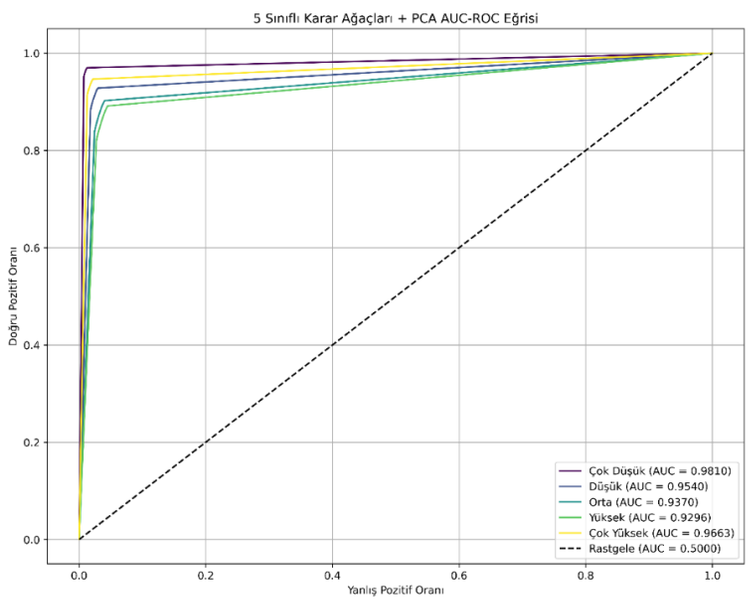
	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
Çok Düşük	0,9618	0,9628	0,9623	146.954
Düşük	0,9063	0,9077	0,9070	140.892
Orta	0,8738	0,8742	0,8740	142.889
Yüksek	0,8575	0,8592	0,8584	141.014
Çok Yüksek	0,9343	0,9295	0,9319	141.256
Doğruluk			**0,9071**	713.005
Makro Ortalama	0,9067	0,9067	0,9067	713.005
Ağırlıklı Ortalama	0,9071	0,9071	0,9071	713.005

Tablo 30'da karar ağaçları sınıflandırıcısının sınıflandırma raporu gösterilmektedir. Tabloda "çok düşük" sınıfı, kesinlik (0,9618), duyarlılık (0,9628) ve F1 (0,9623) metriklerinde diğer modellere göre en yüksek performansı göstermiştir. Başka bir deyişle model, uçuş başına çok düşük sınıfındaki yolcu sayısını belirlemede diğer sınıflara göre daha başarılıdır. "Düşük" ve "çok yüksek" sınıfları da sırasıyla 0,9070 ve 0,9319 F1 puanlarıyla güçlü bir performans göstermektedir. "Orta" ve "yüksek" sınıfları, diğer sınıflara göre daha düşük performansa sahiptir. Bu durum, gerçek dünya verilerinde oldukça yaygın olan komşu sınıflarla örtüşmeler veya karışıklıklar olduğunu gösterebilir. Model genel olarak değerlendirildiğinde ise %90,71'lik bir doğruluk skoruna sahiptir.



Şekil 136. PCA Destekli 5 Sınıflı Karar Ağaçları Modeli Karmaşıklık Matrisi

Şekil 136'da karar ağaçları modelinin karmaşıklık matrisi gösterilmektedir. Matriste çoğunlukla doğru tahminler yapıldığı görülse de, birbirine yakın sınıflarda yanlış sınıflandırmalar yapılmıştır. Örneğin; "orta" sınıfında olması gereken 10.195 örnek, "yüksek" olarak sınıflandırılmıştır. Bu durum, iki sınıf arasındaki benzerliği açıklamaktadır. Başka bir deyişle model, "orta" ve "yüksek" sınıflarını ayırt etmekte zorlanmaktadır. Bu duruma benzer olarak "çok yüksek" ve "yüksek" sınıfı ile "düşük" ve "çok düşük" sınıfları örnek gösterilebilir. Yanlış sınıflandırmaların çoğunun birbirine çok yakın sınıflar arasında meydana geldiği görülebilir. Model, genelleştirmeyi oldukça iyi bir şekilde yapsa da birbirine yakın kategoriler arasındaki nüansları ayırt etmede kısmen zorlanmaktadır.



Şekil 137. PCA Destekli 5 Sınıflı Karar Ağaçları Modeli AUC-ROC Eğrisi

Şekil 137'de ise modele ait AUC-ROC eğrisi gösterilmektedir. Model, "çok düşük" (0,9810) ve "çok yüksek" (0,9663) sınıflarını, diğer sınıflara göre neredeyse mükemmel bir şekilde ayırabilmektedir. "Yüksek" (0,9296) ve "orta" (0,9370) sınıfları ise diğer sınıflara göre düşük performansa sahip sınıflar olarak görülmektedir. Genel olarak incelendiğinde ise tüm AUC skorlarının 0,90'ın üzerinde olduğu ve modelin tüm sınıflar arasında iyi bir ayrım gücüne sahip olduğu söylenebilir.

Üçüncü olarak KNN sınıflandırıcısı seçilmiş ve bu sınıflandırıcıda kullanılmak üzere hiperparametre optimizasyonu için RandomizedSearchCV tekniği tüm veri setine uygulanmıştır. Arama işlemi, 3 katlı çapraz doğrulama ve 20 iterasyon ile gerçekleştirilmiştir. Arama sonucunda bulunan en iyi değerler şu şekilde gösterilebilir:

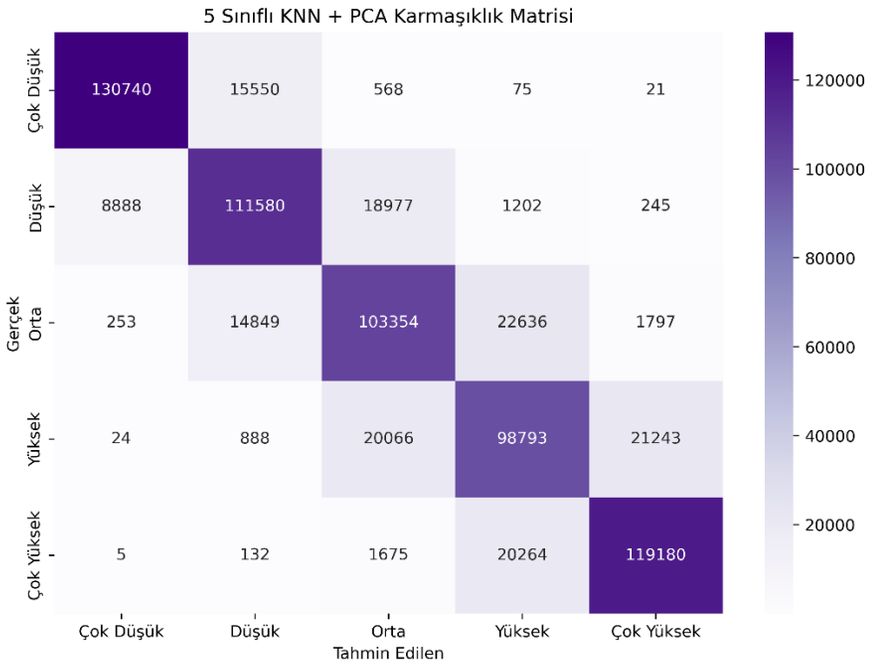
- weights: Distance
- n_neighbors: 7
- metric: Manhattan

Bulunan en iyi hiperparametreler ile oluşturulan model test edilmiştir.

Tablo 31. PCA Destekli 5 Sınıflı KNN Modeli Sınıflandırma Raporu

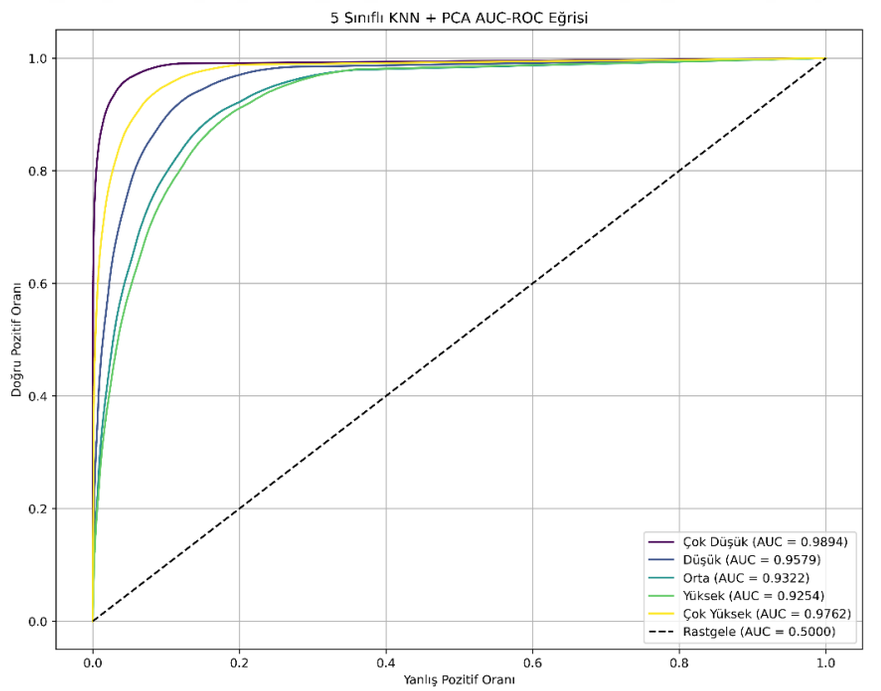
	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
Çok Düşük	0,9345	0,8897	0,9115	146.954
Düşük	0,7803	0,7920	0,7861	140.892
Orta	0,7146	0,7233	0,7189	142.889
Yüksek	0,6910	0,7006	0,6958	141.014
Çok Yüksek	0,8364	0,8437	0,8401	141.256
Doğruluk			**0,7905**	713.005
Makro Ortalama	0,7913	0,7898	0,7905	713.005
Ağırlıklı Ortalama	0,7924	0,7905	0,7913	713.005

Tablo 31'de, KNN sınıflandırıcısına ait sınıflandırma raporu gösterilmektedir. Tablo incelendiğinde, "çok düşük" sınıfının tüm değerlendirme metriklerinde en yüksek performansı gösterdiği görülmektedir (kesinlik=0,9345, duyarlılık=0,8897, F1=0,9115). Model, ilgili sınıfı belirlemede diğer sınıflardan daha başarılıdır. "Çok yüksek" sınıfı, "çok düşük" sınıfı hariç diğer sınıflara göre daha başarılıdır ancak yine de genel anlamda yetersiz denilebilir (kesinlik=0,8364, duyarlılık=0,8437, F1=0,8401). Modelin bu iki uç sınıfı diğer sınıflardan etkili bir şekilde ayırdığı söylenebilir. Model, "düşük", "orta" ve "yüksek" sınıflarını ayırt etmekte ise oldukça zorlanmaktadır. En iyi hiperparametrelerle çalıştırılmasına rağmen model, nispeten orta seviyede bir performans göstermektedir. Genel doğruluk skoru (0,7905), test setinin %79,05'inin beş sınıfı doğru şekilde sınıflandırıldığını göstermektedir. Sonuçlar, sınıflara ait sınırların belirsiz olduğunu ve örtüştüğünü (çok düşük ve çok yüksek sınıfları hariç) göstermektedir.



Şekil 138. PCA Destekli 5 Sınıflı KNN Modeli Karmaşıklık Matrisi

Şekil 138'de modele ait karmaşıklık matrisi görülmektedir. Şekilde, sınıfların birbirlerine yaklaşmaya başladıklarında yanlış sınıflandırmaların arttığı görülebilir. Örneğin; "orta" sınıfı, "yüksek" (22.636) ve "düşük" (14.849) sınıfları ile oldukça yüksek bir karışıklık göstermektedir. Bunun nedeni ise bu üç sınıfın örtüşen sınırlara sahip olmasıdır.



Şekil 139. PCA Destekli 5 Sınıflı KNN Modeli AUC-ROC Eğrisi

Şekil 139'da ise modele ait sınıfların AUC-ROC eğrisi gösterilmektedir. Özellikle "çok düşük" (AUC=0,9894) ve "çok yüksek" (AUC=0,9762) sınıfları en iyi sınıf ayrımı gösteren sınıflar

olarak görünmektedir. “Düşük” (AUC=0,9579) sınıfı ise en iyi performansı gösteren üçüncü sınıf olarak görülebilir. “Orta” (AUC=0,9322) ve “yüksek” (0,9254) sınıflarının AUC değerleri daha düşük görülmektedir. Skorlardaki düşüş, sınıf sınırlarının ayırt edilmesinin daha zor olduğunu göstermektedir.

Dördüncü olarak Naive Bayes sınıflandırıcısı seçilmiş ve bu sınıflandırıcıda da hiperparametre optimizasyonu için RandomizedSearchCV tekniği seçilmiş ve tüm veri setine uygulanmıştır. Arama işlemi, 3 katlı çapraz doğrulama ve 20 iterasyon ile gerçekleştirilmiştir. Arama sonucunda bulunan en iyi değerler şu şekilde gösterilebilir:

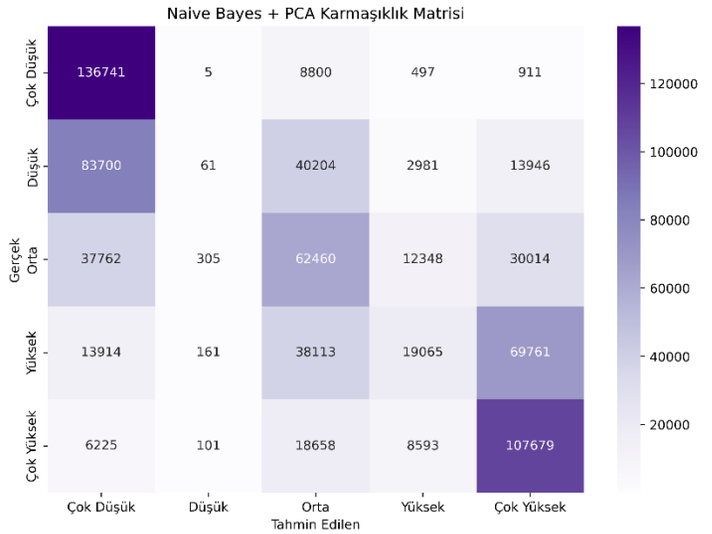
- var_smoothing: 8,286427728546843e-08

Tablo 32. PCA Destekli 5 Sınıflı Naive Bayes Modeli Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
Çok Düşük	0,4913	0,9305	0,6430	146.954
Düşük	0,0964	0,0004	0,0009	140.892
Orta	0,3713	0,4371	0,4015	142.889
Yüksek	0,4384	0,1352	0,2067	141.014
Çok Yüksek	0,4844	0,7623	0,5923	141.256
Doğruluk			**0,4572**	713.005
Makro Ortalama	0,3763	0,4531	0,3689	713.005
Ağırlıklı Ortalama	0,3774	0,4572	0,3714	713.005

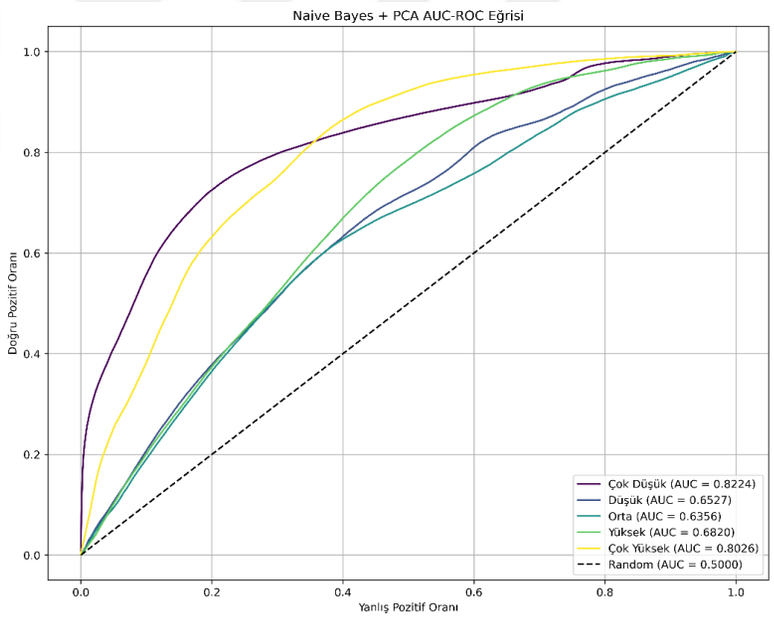
Tablo 32’de Naive Bayes sınıflandırıcısına ait sınıflandırma raporu gösterilmektedir. Şu ana kadarki modeller arasında 0,4572 genel doğruluk skoru ve diğer metriklerdeki (makro ortalama ve ağırlıklı F1 skorları=~0,37) en düşük performansı göstermiştir. “Çok düşük” sınıfı (kesinlik=0,4913, duyarlılık=0,9305, F1=0,6430) yüksek duyarlılık skoruna sahip olsa da[236], “düşük” sınıfı ise neredeyse tamamen başarısızdır. Yani model bu sınıfı neredeyse hiç ayırt edememiştir. “Orta”, “yüksek” ve “çok yüksek” sınıflarında da dengesiz ve yanlış sınıflandırma oldukça yüksek görünmektedir. Naive Bayes sınıflandırıcısının oldukça düşük performans göstermesi, olasılıksal bir sınıflandırıcı olması ve özelliklerin birbirinden bağımsız olduğunu varsaymasından kaynaklanmaktadır. Ancak dayandığı bu temel, gerçek dünya uygulamalarında çoğunlukla geçerli değildir (Zhang, 2004). Veri seti 1’de bulunan yolcu sayısı, uçuş mesafesi ve koltuk sayısı gibi özellikler doğrudan birbirleriyle ilişkidir. Böylece, modelin, veri seti 1 için uygun bir sınıflandırıcı olmadığı söylenebilir.

[236] Gerçek “çok düşük” sınıfına ait örneklerin çoğunun yakalandığını göstermektedir.



Şekil 140. PCA Destekli 5 Sınıflı Naive Bayes Modeli Karmaşıklık Matrisi

Şekil 140'ta Naive Bayes sınıflandırıcısının oldukça dengesiz ve yanlış sınıflandırma yaptığı açıkça görülmektedir.



Şekil 141. PCA Destekli 5 Sınıflı Naive Bayes Modeli AUC-ROC Eğrisi

Şekil 141'de ise tüm eğrilerin tepe noktasından oldukça uzak bir konumda oldukları görülmektedir.

Son olarak çok katmanlı algılayıcı (yapay sinir ağları) sınıflandırıcısı seçilmiş ve bu sınıflandırıcıda kullanılmak üzere hiperparametre optimizasyonu için RandomizedSearchCV tekniği, tüm veri setine uygulanmıştır. Arama işlemi, 3 katlı çapraz doğrulama ve 20 iterasyon ile gerçekleştirilmiştir. Arama sonucunda bulunan en iyi değerler şu şekilde gösterilebilir:

- solver: Adam

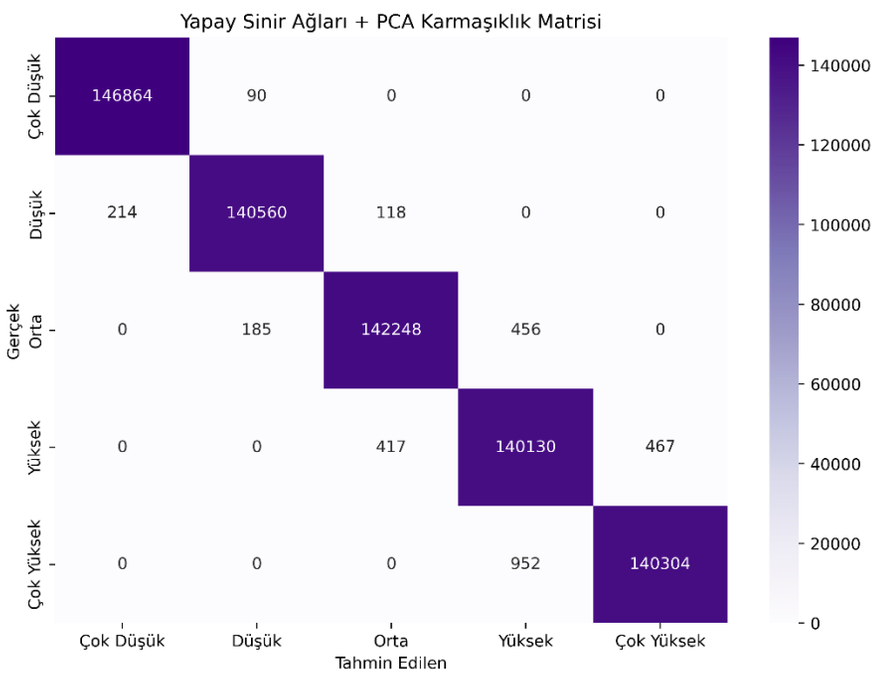
- learning_rate: Adaptive
- hidden_layer_sizes: (128, 64)
- alpha: 0,001
- activation: ReLu

Bulunan en iyi hiperparametreler ile oluşturulan model test edilmiştir.

Tablo 33. PCA Destekli 5 Sınıflı Yapay Sinir Ağı Modeli Sınıflandırma Raporu

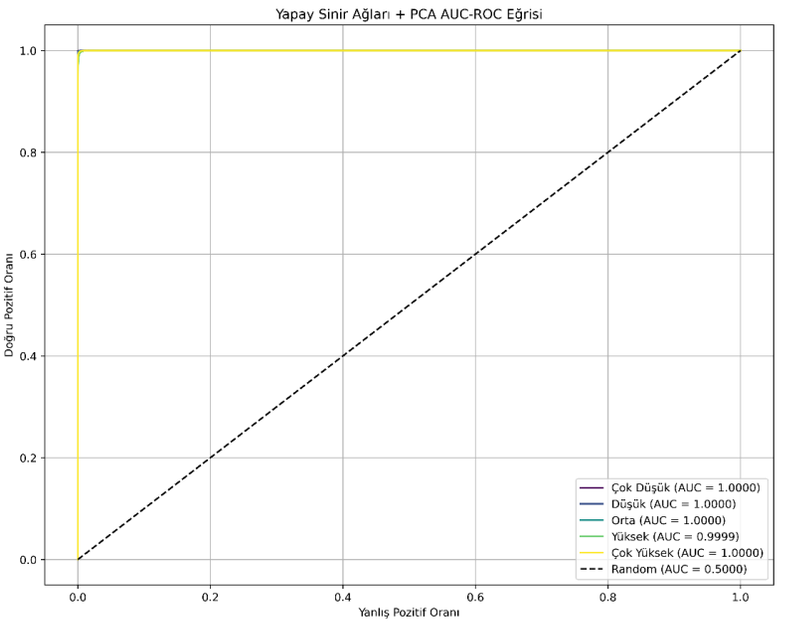
	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
Çok Düşük	0,9985	0,9994	0,9990	146.954
Düşük	0,9980	0,9976	0,9978	140.892
Orta	0,9963	0,9955	0,9959	142.889
Yüksek	0,9901	0,9937	0,9919	141.014
Çok Yüksek	0,9967	0,9933	0,9950	141.256
Doğruluk			**0,9959**	713.005
Makro Ortalama	0,9959	0,9959	0,9959	713.005
Ağırlıklı Ortalama	0,9959	0,9959	0,9959	713.005

Tablo 33'te, yapay sinir ağına ait sınıflandırma raporu gösterilmektedir. Skorlar, modelin diğer modellerden tüm metriklerde daha iyi bir performansa sahip olduğunu ve neredeyse mükemmel bir sınıflandırma yaptığını göstermektedir (doğruluk=0,9959). Birbirine yakın sınıflarda diğer modellerin zorlandığı göz önüne alındığında, modelin ne kadar değerli olduğu anlaşılabilir. Çok katmanlı algılayıcının başarısı; doğrusal olmayan ve karmaşık karar sınırlarını öğrenebilmesinde yatmaktadır (Taud ve Mas, 2018). Veri seti 1'deki özellikler arasında bulunan gizli ve tespit edilemeyen örüntüler, model tarafından kolaylıkla tanımlanabilmiştir.



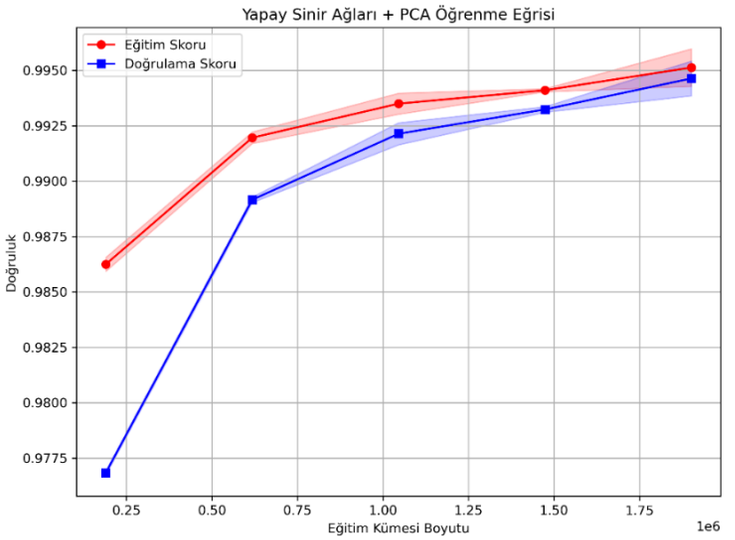
Şekil 142. PCA Destekli 5 Sınıflı Yapay Sinir Ağı Modeli Karmaşıklık Matrisi

Şekil 142'de yapay sinir ağı sınıflandırıcısına ait karmaşıklık matrisinde tüm sınıfların son derece yüksek doğrulukla sınıflandırıldığı kolaylıkla görülebilmektedir. Test seti büyüklüğü ile kıyaslandığında son derece nadir olduğu görülen yanlış sınıflandırmalar, istatistiki olarak önemsiz görülebilir. Örneğin; sadece 214 "düşük" sınıfına ait örnek, "çok düşük" olarak sınıflandırılmıştır. En çok yanlış sınıflandırma ise 952 adet ile "çok yüksek" sınıfının "yüksek" olarak sınıflandırılması ile yapılmıştır. Bu miktar, ilgili sınıf için toplam tahminlerin %1'inden daha azını göstermektedir.



Şekil 143. PCA Destekli 5 Sınıflı Yapay Sinir Ağı Modeli AUC-ROC Eğrisi

Şekil 143'te modele ait AUC-ROC eğrisi gösterilmektedir. Grafikte, tüm sınıfların kusursuz bir sınıflandırma performansı ortaya koyduğu görülebilmektedir. Beş sınıfa ait eğriler, çoğunlukla üst üste binmiş durumda ve tepe noktası hizasında birleşmektedir. AUC skorunun 1 olması, modelin tüm pozitif örnekleri herhangi bir negatif örnekten hata olmadan daha üst sıraya koyduğu anlamına gelmektedir. Temel bileşen analizi ile boyutların azaltılmasına rağmen ortaya çıkan bu sonuç, model sağlamlığını ve genelleme yeteneğinin neredeyse hatasız olduğunu göstermektedir.



Şekil 144. PCA Destekli 5 Sınıflı Yapay Sinir Ağı Modeli Öğrenme Eğrisi

Şekil 144'te ise modele ait öğrenme eğrisi görülmektedir. Eğitim skorunun 0,9863 seviyelerinde başlayıp eğitim kümesi boyutu arttıkça maksimum 0,9950 seviyesinde sabit kaldığı görülebilir. Doğrulama skorunun ise oldukça yüksek ancak eğitim skoruna göre bir miktar daha düşük başladığı görülmektedir (~0,9763). Eğitim kümesi boyutu arttıkça doğrulama skoru da yükselmiş ve eğitim skoruna yaklaşmıştır. Başka bir deyişle model, minimum varyansa sahiptir.

Tablo 34. Tüm Sınıflandırma Modellerinin Performans Karşılaştırması

Model	Doğruluk	Kesinlik	Duyarlılık	F1
Yapay Sinir Ağı	0,9959	0,9959	0,9959	0,9959
Rastgele Orman	0,9440	0,9439	0,9438	0,9438
Karar Ağaçları	0,9071	0,9067	0,9067	0,9067
KNN	0,7905	0,7913	0,7898	0,7905
Naive Bayes	0,4572	0,3763	0,4531	0,3689

Tablo 34'te ise tüm modellerin performansları karşılaştırılmıştır. Modellerin etkinliğinde önemli farklar bulunmaktadır. Yapay sinir ağı %99,59'luk doğruluk ve kesinlik, duyarlılık ve F1 skorları ile en iyi performans gösteren model olmuştur. İkinci sırada rastgele orman modeli gelmektedir. Model, %94,40 doğruluk, %94,39 kesinlik, %94,38 duyarlılık ve F1 skorları ile oldukça başarılı bir performans sergilemiştir. Üçüncü sırada ise karar ağaçları modeli gelmektedir. %90,71 doğruluk ve %90,67 kesinlik, duyarlılık ve F1 skorları ile iyi bir performansa sahiptir. KNN modeli, daha düşük boyutlu alanlarda iyi performans gösterme eğiliminde olduğundan (Halder vd., 2024), veri seti 1'de yeteri kadar iyi bir sınıflandırma performansı gösterememiştir (doğruluk=0,7905). Naive Bayes modeli ise en zayıf performansı gösteren modeldir. Daha önce de belirtildiği gibi oldukça zayıf performans, özellikler arasındaki bağımsızlık varsayımına bağlanmaktadır. Ayrıca Naive Bayes, doğrusal olmayan sınırları öğrenme kapasitesinden yoksundur.

BEŞİNCİ BÖLÜM

5. İKİNCİ FAZ (DERİN ÖĞRENME) ANALİZLER VE BULGULAR

5.1. Evrişimli Sinir Ağları Uygulaması

Veri seti 2'de, öncelikle klasik evrişimli sinir ağları kullanılarak temel bir model oluşturulmuş ve ardından sınıflandırma işlemi yapılmıştır. Temel model oluşturulduktan sonra, aşırı veya eksik öğrenme durumlarında kullanılmak üzere çeşitli optimizasyon ve düzenlileştirme teknikleri uygulanmıştır. Hem temel model hem de optimize edilen model karşılaştırılmıştır.

Veri seti 2, VGG benzeri bir yapıda ancak her evrişim işleminden sonra "batch normalization[237]" eklenerek temel bir CNN modeli ile sıfırdan eğitime tabi tutulmuştur. CNN modelinin giriş katmanı 224x224x3'lük renkli bir görüntüden oluşmaktadır. Birinci evrişim bloğunda 32 filtrenin (nöron sayısı) her biri 3×3 çekirdekten oluşmakta ve bu blokta ReLU aktivasyonu kullanılmaktadır. Piksel ekleme işlemi parametresi "same" olarak ayarlanmıştır, yani çıktı boyutu ile girdi boyutu eşit olacaktır. Havuzlama katmanında ise 2x2'lik bir alt örnekleme uygulanarak boyut yarı yarıya azaltılmıştır. İkinci evrişim bloğunda 64 filtre bulunmakta ve ilk bloktaki gibi özelliklere sahiptir. Üçünü evrişim bloğunda ise aynı özellikler korunmakta olup, filtre sayısı 128'e çıkarılmıştır. Dördüncü evrişim bloğunda 256, beşinci evrişim bloğunda ise 512 filtre bulunmaktadır. Bu bloktan sonraki düzleştirme katmanında ise üç boyutlu özellik haritaları bir boyuta indirgemiştir (3×3×512=4608). Birinci tam bağlantılı katman (yoğun) 128, ikinci tam bağlantılı katman ise 256 nörondan oluşmakta ve her iki blokta da ReLU aktivasyon fonksiyonu kullanılmaktadır. Bırakma katmanında ise 512 nöron bulunmakta ve nöronların %50'si rastgele devre dışı bırakılacak şekilde ayarlanmıştır. Böylece ağın aşırı öğrenmesinin önüne kısmen de olsa geçilebilecektir. Çıkış katmanında ise 30 sınıf bulunmakta ve bu nedenle çok sınıflı sınıflandırma problemlerinde sıklıkla kullanılan Softmax aktivasyon fonksiyonu seçilmiştir. Modelin mimarisi Tablo 35'te gösterilmiş olup, uygulamaya ait Python kodu ek 1'de verilmiştir.

Tablo 35. Veri Seti 2 Temel CNN Mimarisi

Katman	Nöron Sayısı	Çekirdek Boyutu	Aktivasyon Fonksiyonu	Çıkış Şekli
Giriş	-	-	-	(224, 224, 3)
1. Evrişim	32	(3×3)	ReLU	(224, 224, 32)
1. Havuzlama	-	(2×2)	-	(112, 112, 32)
2. Evrişim	64	(3×3)	ReLU	(112, 112, 64)
2. Havuzlama	-	(2×2)	-	(56, 56, 64)

[237] Toplu normalizasyon

3. Evrişim	128	(3×3)	ReLU	(56, 56, 128)
3. Havuzlama	-	(2×2)	-	(28, 28, 128)
4. Evrişim	256	(3×3)	ReLU	(28, 28, 256)
4. Havuzlama	-	(2×2)	-	(14, 14, 256)
5. Evrişim	512	(3×3)	ReLU	(14, 14, 512)
5. Havuzlama	-	(2×2)	-	(7, 7, 512)
Düzleştirme	-	-	-	(4608,)
1. Yoğun	256	-	ReLU	(256)
2. Yoğun	512	-	ReLU	(512)
Bırakma	512	-	-	-
Çıkış	30	-	Softmax	(30)

Modelde toplam 8.142.302 parametre olup, bunların 8.140.318'i eğitilebilir, 1984'ü ise eğitimez durumdadır. Eğitilemeyen 1984 parametre, genellikle toplu normalizasyon adımlarından gelmektedir. Modelin katmanları ayrıca ek 2'de verilmiştir. Model, eğitim sürecine dahil edilmeden önce ön işlemeye tabi tutulmuştur. Yapılan denemelerde, orijinal boyutu 600x600 olan görüntüler, hem donanım hem de sistem uyumluğu için en uygun görüntü giriş değeri olan 224x224 boyutuna düşürülmüş ve işlemler bu doğrultuda gerçekleştirilmiştir. Ardından, veri normalizasyonu yapılarak tüm piksel değerleri [0,1] aralığına indirgenmiştir.

Dizinde bulunan her bir klasör adı, içinde barındırdığı görüntülerin sınıfını temsil edecek şekilde ayarlanmıştır. Ardından for döngüsü ile her bir sınıfta (klasör) yineleme yapılarak, tüm görüntü isimleri toplanmış ve her bir görüntünün yolu[238] oluşturulmuştur. Böylece sınıf etiketlerinin (diziler) tam bir listesi elde edilmiştir. Dosya yolları ve etiket listeleri DataFrame formatına dönüştürülmüştür. Etiketler string formatından integer formatına dönüştürülmüş ve her bir sınıf ismini benzersiz bir tamsayı dizinine eşleyen bir sözlük oluşturulmuştur. Ardından mapping yapılarak DataFrame'de kayıtlı sınıf isimlerine karşılık gelen tamsayı değerleriyle değiştirilmiştir.

Veri seti bölümlendirmesi için "sklearn.model_selection" kütüphanesinden "train_test_split" tekniği kullanılmış ve veri kümesi rastgele bölünmüştür. %80-%10-%10 kuralı uygulanarak 10.000 görüntünün 8000'inin eğitim, 1000'inin doğrulama ve 1000'inin test seti olarak ayrılması sağlanmıştır. Ayrıca, sınıf dağılımının tüm bölümlemelerde dengeli kalmasını sağlamak için tabakalı[239] yöntem kullanılmıştır.

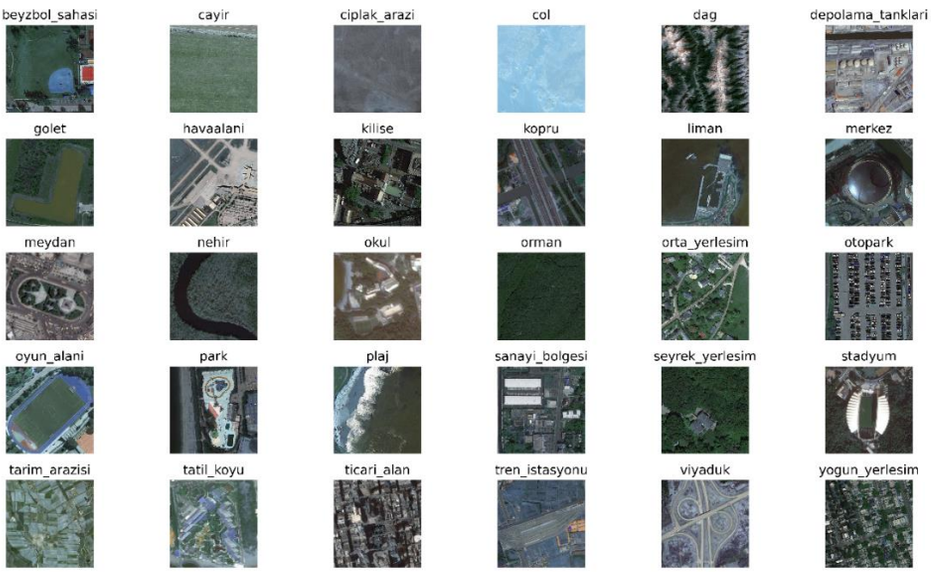
Tensorflow kütüphanesinin, görüntüleri daha hızlı yüklemesini sağlamak için AutoTune fonksiyonu kullanılmıştır. Bu fonksiyon paralel iş parçacıklarının sayısını otomatik olarak

[238] Path
[239] Stratify

ayarlanmasını sağlamaktadır. Yığın boyutu[240] 16 olarak ayarlanmıştır. Böylece, modele verilen her bir yığın, 16 görüntü ve etiketten oluşacaktır. Görüntüler, "from_tensor_slices" komutu ile eğitim görüntülerinin bulunduğu DataFrame üzerinden "dosya yolu, etiket" çifti ile alınarak Tensorflow veri setine dönüştürülmüştür. Buna ek olarak modelin herhangi bir sırada görüntüleri almasını engellemek için veri seti, 1000 görüntüden oluşan bir ara belleğe alınarak rastgele karıştırılmıştır[241]. Son olarak "prefetch" komutu ile, modelin eğitildiği mevcut yığın işlemleri devam ederken, diğer yığının önceden hazırlanması sağlanmıştır. Böylece hesaplama hızı ve verimliliğinin artması hedeflenmiştir. Doğrulama ve test kümelerinin tutarlılığını korumak için karıştırma işlemi uygulanmamıştır. Bu işlem sadece eğitim setine uygulanmıştır.

Görüntüler, "tf.image.decode_jpeg" komutu ile JPEG uzantılı görüntüleri 3 kanallı bir RGB görüntü tensörüne dönüştürülmüş ve "tf.image.resize" komutu ile de görüntüler 224×224 piksel olarak yeniden boyutlandırılmıştır. Bu boyutun seçilmesinin nedeni çok sayıda ön eğitimli modelin (ResNet, EfficientNet, VGG vb.) bu değerler ile verileri eğitmesi ve başarılı olmasıdır. Son olarak görüntülerin piksel değerleri "img = tf.image.resize(img, IMG_SIZE) / 255.0" komutu ile [0, 1] aralığına indirgenerek normalizasyon uygulanmıştır. Normalizasyona ilişkin Python kodu ek 3'te verilmiştir.

Veri setindeki örnek görüntüler ve sınıfları orijinal boyutlarında (600x600) Şekil 145'te gösterilmektedir.

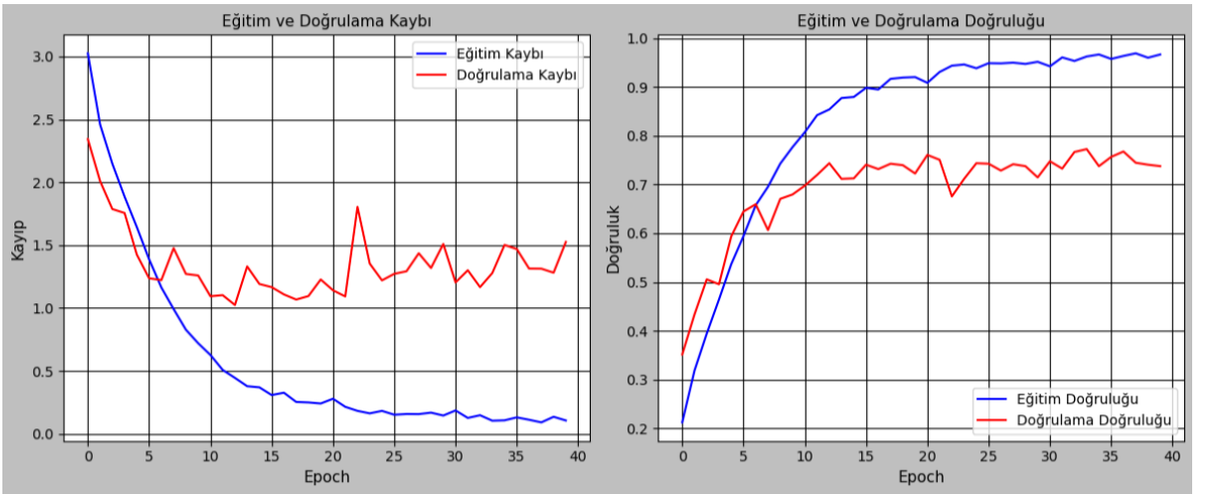


Şekil 145. Veri Seti 2 Görüntüleri ve Sınıfları

[240] Batch size
[241] Shuffle

Model, ilk olarak herhangi bir optimizasyon algoritması olmadan eğitilmiş ve eğitim doğruluğu %99,76 iken, test doğruluğu %57 olarak gerçekleşmiştir. Bu skor, aşırı öğrenme olduğunu göstermektedir. Evrişimli sinir ağlarının eğitimi sırasında mutlaka bir optimizasyon algoritmasının kullanılması gerekmektedir. Optimizasyon algoritmaları, geri yayılım ile elde edilen gradyanlar doğrultusunda ağın ağırlık ve bias parametrelerini güncelleyerek kayıp fonksiyonunun minimize edilmesini sağlar. Bu süreç, modelin veriden öğrenebilmesi için temel bir mekanizmadır.

Mevcut modele Adam optimizasyon algoritması eklenerek analizlere devam edilmiştir. Öğrenme oranı 0,0001, epoch değeri ise 50 olarak belirlenmiştir. Kayıp fonksiyonu, sınıf aralıkları 0-29 aralığında olduğundan "one hot encoding" işlemleri yapılmamış ve bu yüzden "sparse_categorical_crossentropy" olarak seçilmiştir. Eğitim seti, doğrulama seti ve test seti bölümlendirmesi ise %80-%10-%10 kuralı uygulanarak 8000 eğitim görüntüsü, 1000 doğrulama görüntüsü ve 1000 test görüntüsü şeklinde belirlenmiştir. "batch_size" değeri ise 16'dır. Modelde ayrıca öğrenme oranı optimize edici olan ReduceLROnPlateau kütüphanesi kullanılmıştır. Bu kütüphane eğitim sırasındaki gelişmeyi modelin güncel performansına bağlı olarak ayarlar. Başka bir deyişle, öğrenme sürecini optimize eder. İlerleme yavaşladığında adım boyutunu azaltarak modelin daha iyi bir şekilde yakınsamasına yardımcı olur. Model eğitimi tamamlandıktan sonra elde edilen sonuçlar aşağıdaki grafik ve tabloda gösterilmektedir.

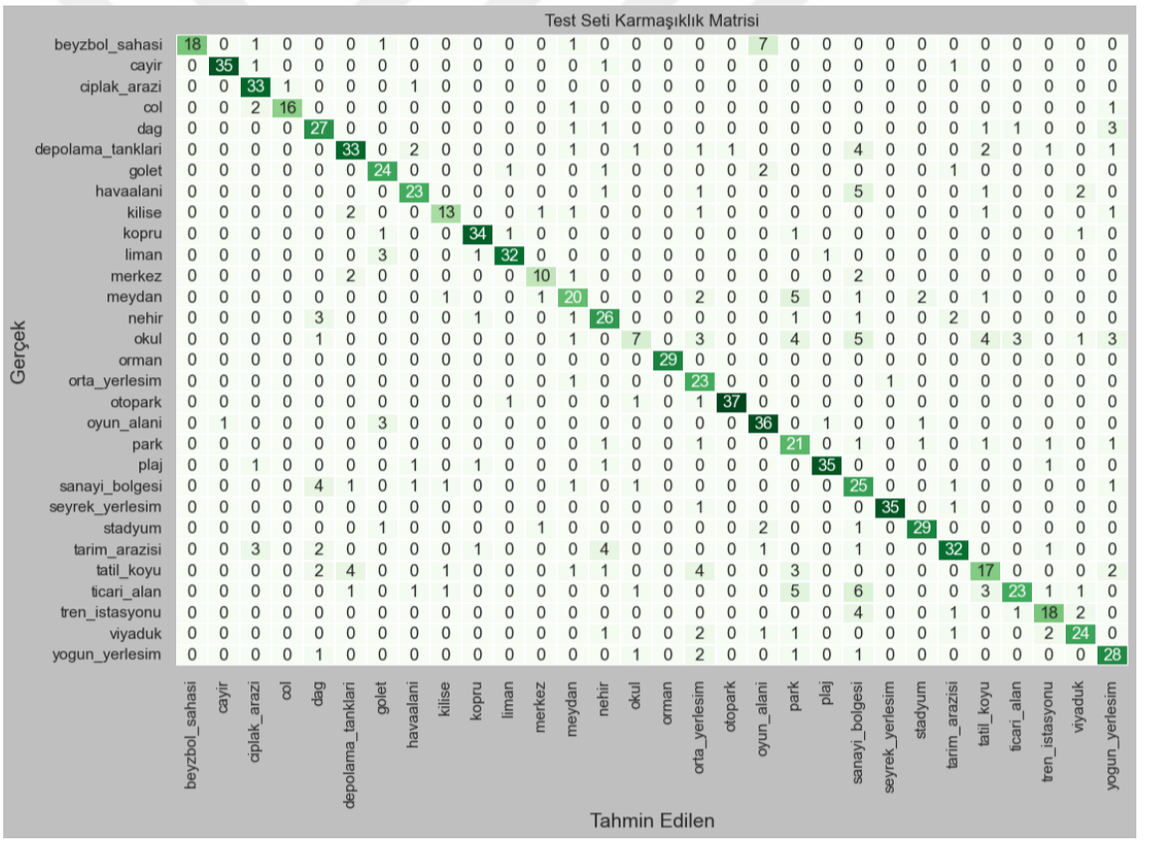


Şekil 146. Temel CNN Modeli Kayıp ve Doğruluk Grafiği

Şekil 146'da CNN modeline ait kayıp ve doğruluk grafikleri gösterilmektedir. ReduceLROnPlateau fonksiyonu, eğitim süreci boyunca 18 defa öğrenme oranına müdahale edip düşürmüştür. Sol tarafta görülen kayıp grafiğinde eğitim kaybının yaklaşık 40 epoch sonunda 0'a yakın olarak azaldığı görülmektedir. Yani, model eğitim setini etkili bir şekilde

öğrenmiştir. Doğrulama kaybı ise 10. epochtan sonra dalgalanmaya başlamış görünmektedir. Eğitim ve doğrulama kayıpları arasında görülen açıklık, aşırı öğrenmeyi işaret etmektedir. Başka bir deyişle model, daha önce hiç görmediği doğrulama setinde yeteri kadar genelleme yapamamıştır. Sağ tarafta görülen grafikte ise eğitim doğruluğunun ilk epochtan itibaren hızlı bir şekilde arttığı ve son epoch itibariyle %99'a yaklaştığı görülmektedir. Tıpkı kayıp grafiğinde olduğu gibi, modelin eğitim verilerine oldukça uyduğu söylenebilir. Doğrulama skoru ise %70-75 arasına ulaştıktan sonra küçük dalgalanmalar göstermektedir. Eğitim ve doğrulama skorları arasındaki yaklaşık %20'lik fark da aşırı öğrenmeyi kanıtlamaktadır. Başka bir deyişle model, sınırlı bir genelleme yeteneğine sahiptir.

Eğitim sırasında model çok iyi bir öğrenme oranı (~%96) yakalasa da, daha önce hiç görmediği test verilerinde %76,3'lük bir doğru tahmin oranı yakalamıştır. Bu sonuçlar, modelin optimize edilmesini gerektirmektedir. Modele ait sınıflandırma raporu ek 4'te verilmiştir.



Şekil 147. Temel CNN Modeli Karmaşıklık Matrisi

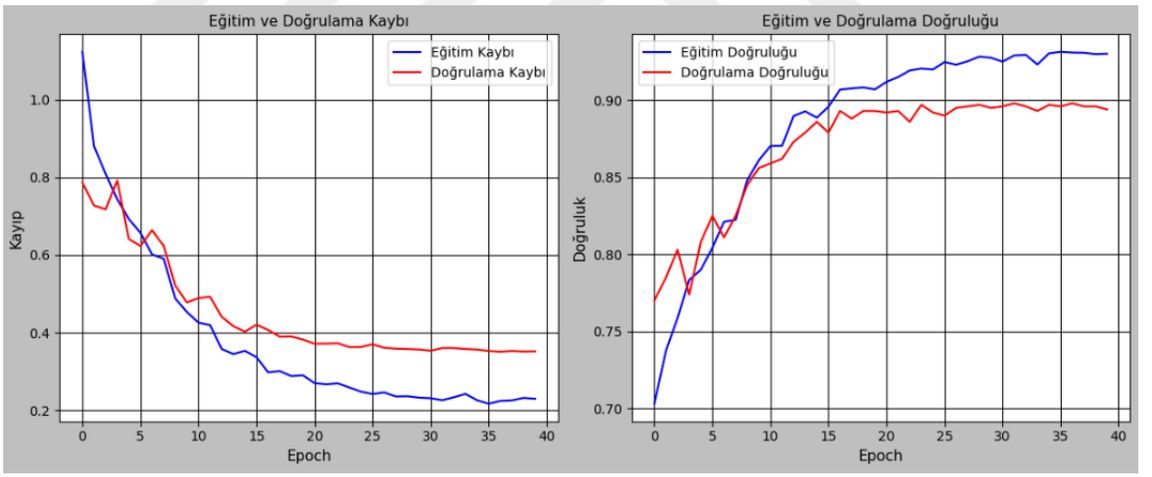
Şekil 147'de ise modele ilişkin test seti karmaşıklık matrisi gösterilmektedir. Matristen de görüleceği üzere, modelin istenilen seviyede sınıflandırma yapamadığı görülmektedir. 1000 test görüntüsünde 227 adet hatalı sınıflandırma yapılmıştır. En düşük sınıflandırma performansına sahip sınıflar okul, park ve sanayi bölgesi gibi sınıflardır. Diğer yanlış sınıflandırmalar, görsel benzerlikten dolayı kaynaklanmaktadır. Bu gibi durumların önüne

geçebilmek için modele veri artırımı yapılmış ve çeşitli açılardan üretilen yeni görüntülerle model tekrardan eğitilmiştir.

Veri artırımı yapılan modelde ise erken durdurma[242] fonksiyonu, ReduceLROnPlateau ile birlikte kullanılmıştır. Öğrenme oranı başlangıç değeri 0,0001 olarak belirlenmiş ve optimizasyon algoritması olarak ilk modeldeki gibi Adam kullanılmıştır. Kayıp fonksiyonu ise "sparse_categorical_crossentropy" olarak seçilmiştir.

Veri artırımı için bir fonksiyon oluşturulmuş ve görüntülere rastgele dönüşümler uygulanarak eğitim verilerinin çeşitliliği yapay olarak artırılmaya çalışılmıştır. Böylece, modelin daha iyi genelleştirilmesi ve aşırı öğrenmenin mümkün olduğunca azaltılması hedeflenmiştir. Görüntüler, rastgele olarak 0 dereceden 270 dereceye kadar 90 derecelik açılarla yatay ve dikey olarak döndürülmüştür. Ayrıca, görüntülerin parlaklık değeri %20 oranında artırılmış ve kontrast değerleri 0,8-1,2 aralığında rastgele değiştirilmiştir. Son olarak görüntüler 250x250 boyutuna kadar doldurulmuş[243] ve 224×224'lük bir bölge rastgele kesilip yakınlaştırılmıştır.

İlk ön işleme adımında olduğu gibi veri artırımı yoluyla eğitilecek olan model, AutoTune fonksiyonu ile paralel iş parçacığı sayısının optimum olarak belirlenmesi sağlanmıştır.



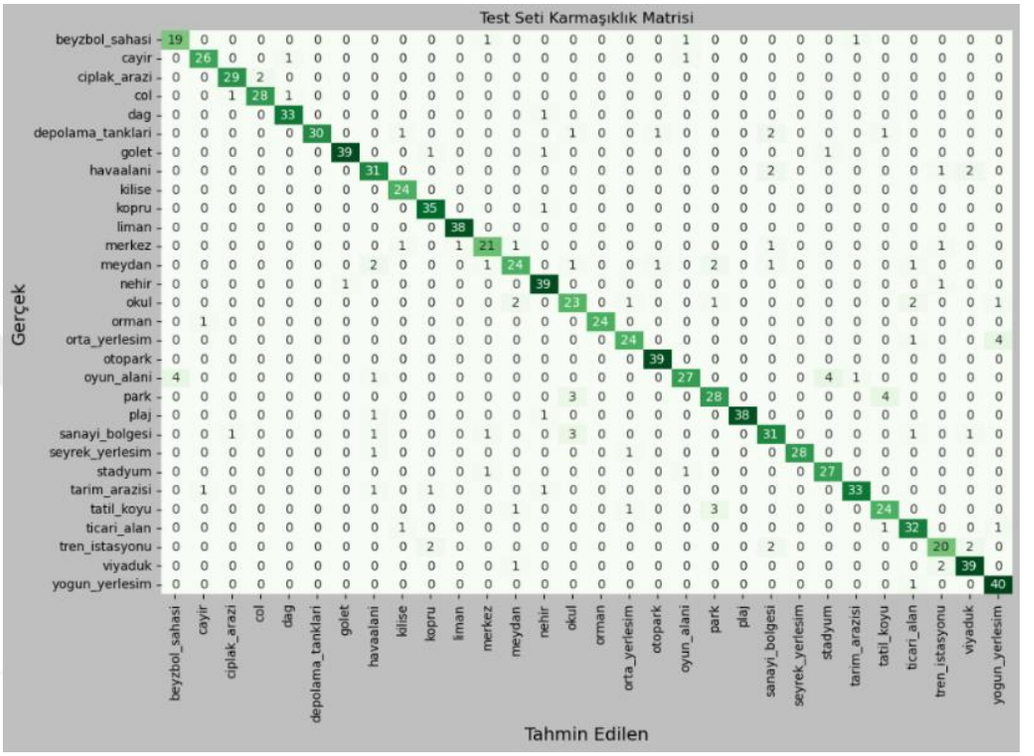
Şekil 148. Veri Artırımı Yapılmış CNN Modeli Kayıp ve Doğruluk Grafiği

Şekil 148'de, veri artırımı yapılmış modelin kayıp ve doğruluk grafiği görülmektedir. Eğitim kaybının 40 epoch sonunda yaklaşık 0,2'ye ulaştığı ve doğrulama kaybının da özellikle 10. epochtan sonra eğitim kaybına yaklaştığı görülebilir. Açıklığın kapanması, modelin aşırı öğrenmeue uğramadığını göstermektedir. Sağ taraftaki doğruluk grafiğine bakıldığında ise 40 epoch sonunda eğitim doğruluğunun %93, doğrulama doğruluğunun ise %89 civarında sabitlendiği görülmektedir. İlk modeldeki %20'lik fark, oldukça küçülerek yaklaşık %4'e

[242] Early stopping
[243] Padding

kadar düşmüştür. Sonuç olarak veri artırımı yapılmış model, hem yüksek eğitim ve doğrulama doğruluğunu sağlamış hem de aşırı öğrenmeyi ortadan kaldırmıştır.

Analiz sonucunda ilk modele göre bir iyileşme gerçekleşmiştir. Modelin genel doğruluk skoru 0,8930'a yükselmiştir. Veri artırımı yapılmış modelin sınıflandırma raporu ise ek 5'te verilmiştir.



Şekil 149. Veri Artırımı Yapılmış CNN Modeli Karmaşıklık Matrisi

Şekil 149'da ise veri artırımı yapılmış modelin karmaşıklık matrisi gösterilmektedir. 1000 test görüntüsünde 103 adet hatalı sınıflandırma yapılmış ve bu değer ilk modeldeki 227 hatalı sınıflandırmaya göre oldukça iyi bir performanstır.

Yukarıda bahsedilen tüm işlemler temel CNN modeli ile temel modelin başarımını artırmak için yapılan düzenlileştirme ve optimizasyon tekniklerini içermektedir. Şimdi gösterilecek işlemler ise görüntülerin tahmin doğruluğuna ilişkin farklı hiperparametrelerle (optimizasyon algoritması, yığın boyutu, görüntü boyutu, eğitim-test bölümlendirmesi) yapılmıştır. Burada da öğrenme oranı iyileştirici ve veri artırımı gibi teknikler uygulanmıştır.

5.1.1. Görüntü Boyutu Bazlı Karşılaştırmalar

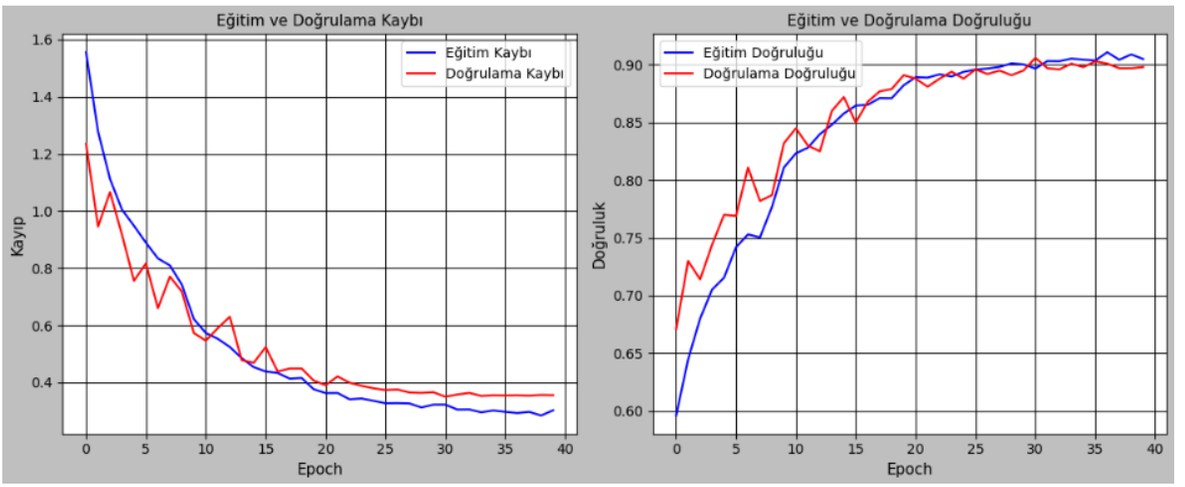
Temel CNN modelinin görüntü boyutu bazlı karşılaştırmaları öncelikle batch_size hiperparametresi 8 ayarlanarak yapılmış, ardından 16 değeri verilerek 64x64, 128x128, 200x200, 250x250 ve 300x300 piksel boyutlarında analizler gerçekleştirilmiş ve doğruluk,

kesinlik, duyarlılık ve F1 skoları arasında karşılaştırmalar yapılmıştır. Tüm modellerin öğrenme oranı başlangıç değeri 0,0001 olarak seçilmiş ve tüm modellerde öğrenme oranı ayarlayıcı kullanılmıştır. Temel modellerin değerlendirme metrikleri (doğruluk, kesinlik, duyarlılık ve F1 skorları) alındıktan sonra, veri artırımı yapılan modellerin değerlendirme skorları verilmiş ve tabloda nihai sonuçlar gösterilmiştir.

Tablo 36. Görüntü Boyutu Bazlı Karşılaştırmalar 1

Görüntü Boyutu	Öğrenme Oranı Başlangıç Değeri	Yığın Boyutu	Optimizasyon algoritması	Doğruluk	Kesinlik	Duyarlılık	F1
64x64	0,0001	8	Adam	0,8050	0,8148	0,8027	0,8047
128x128	0,0001	8	Adam	0,8830	0,8836	0,8795	0,8799
200x200	0,0001	8	Adam	0,9050	0,9051	0,9017	0,9019
250x250	**0,0001**	**8**	**Adam**	**0,9090**	**0,9073**	**0,9064**	**0,9059**
300x300	0,0001	8	Adam	0,8730	0,8753	0,8658	0,8681

Tablo 36'da batch_size parametresi 8 olarak ayarlanan modellerin görüntü boyutu bazlı sonuçları gösterilmektedir. Batch size değerinin 8 ayarlanması ile, modele 8'li gruplar halinde verilen görüntüler daha hızlı işlenmektedir. Buna göre en iyi sonucu veren boyut, 250x250 olup, 0,9090 doğruluk, 0,9073 kesinlik, 0,9064 duyarlılık ve 0,9059 F1 skorlarına sahiptir. 250x250 boyutlarına sahip ilk modelin doğruluk skoru 0,7780 idi ve aşırı öğrenme belirtileri bulunmaktaydı.



Şekil 150. Kayıp ve Doğruluk Grafiği 1 (250x250)

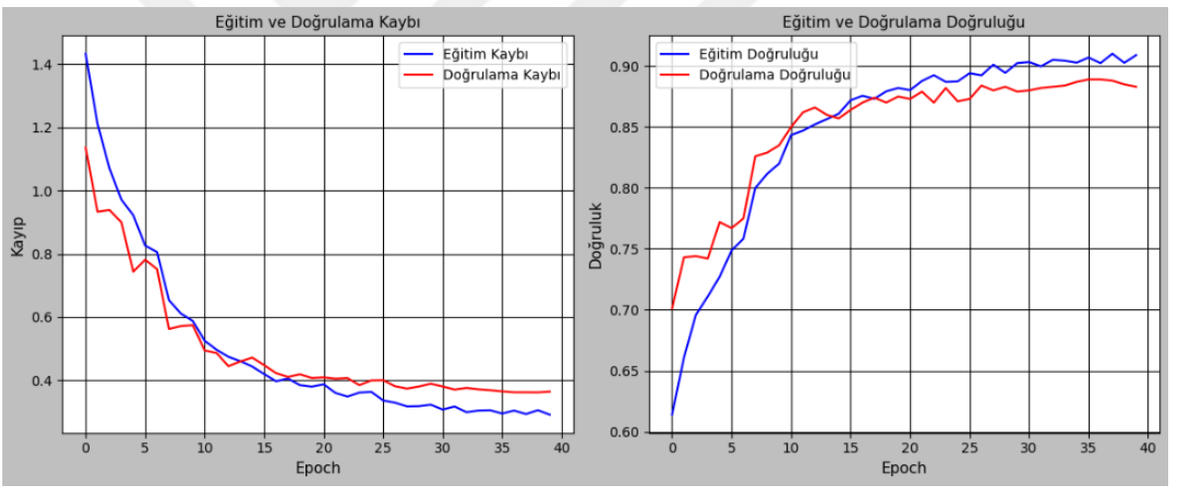
Şekil 150'de veri artırımı yapılan en iyi modelin kayıp ve doğruluk grafikleri gösterilmektedir. Eğitim ve doğrulama kayıpları incelendiğinde, her iki eğrinin de birbirlerine yakın seyrederek azaldığı görülebilir. Doğruluk eğrileri incelendiğinde ise her iki eğrinin de yine birbirlerine yakın olduğu ve 30. epoch itibariyle hemen hemen sabit oldukları söylenebilir. Doğruluk skorlarının ~0,90 seviyelerinde ve kayıp skorlarının ~0,30-0,37 seviyelerinde olması, modelin

genelleme performansının iyi olduğunu göstermektedir. İlk modeldeki aşırı öğrenme problemi, bu modelde çözülmüştür.

Tablo 37. Görüntü Boyutu Bazlı Karşılaştırmalar 2

Görüntü Boyutu	Öğrenme Oranı Başlangıç Değeri	Yığın Boyutu	Optimizasyon algoritması	Doğruluk	Kesinlik	Duyarlılık	F1
64x64	0,0001	16	Adam	0,8160	0,8309	0,8145	0,8186
128x128	0,0001	16	Adam	0,8680	0,8683	0,8644	0,8641
200x200	0,0001	16	Adam	0,8960	0,8968	0,8924	0,8932
250x250	**0,0001**	**16**	**Adam**	**0,9030**	**0,9038**	**0,9010**	**0,9010**
300x300	0,0001	16	Adam	0,8900	0,8898	0,8856	0,8863

Tablo 37'de ise batch_size parametresi 16 olarak ayarlanan modellerin görüntü boyutu bazlı sonuçları gösterilmektedir. Buna göre en iyi sonucu veren boyut, yine 250x250 boyutlarına sahip model olup, 0,9030 doğruluk, 0,9038 kesinlik, 0,9010 duyarlılık ve 0,9010 F1 skorlarına sahiptir. 250x250 boyutlarına sahip ilk modelin doğruluk skoru 0,8030 idi ve yine aşırı öğrenme belirtileri göstermekteydi.



Şekil 151. Kayıp ve Doğruluk Grafiği 2 (250x250)

Şekil 151'de veri artırımı yapılan en iyi modelin kayıp ve doğruluk grafikleri gösterilmektedir. Eğitim ve doğrulama kayıpları incelendiğinde, her iki eğrinin de bir önceki modeldeki gibi birbirlerine yakın seyrederek azaldığı görülmektedir. Doğruluk eğrileri incelendiğinde ise her iki eğrinin de yine birbirlerine yakın olduğu ve 25. epoch itibariyle aralarında oldukça az bir açıklık olduğu söylenebilir. Eğitim doğruluğu skoru ~0,92 civarında iken doğrulama doğruluğu skoru ~0,88 civarındadır. Yine de, modelin genelleme performansının iyi seviyede olduğu söylenebilir. İlk modeldeki aşırı öğrenme problemi, bu modelde de çözülmüş görünmektedir.

5.1.2. Optimizasyon Algoritması ve Görüntü Boyutu Bazlı Karşılaştırmalar

Optimizasyon algoritması bazlı karşılaştırmalarda giriş değeri 64x64-300x300 aralığında

seçilmiş ve öğrenme oranı başlangıç değeri ise 0,001 olarak uygulanmıştır. Modellerde ayrıca öğrenme oranı iyileştirici ve veri artırma tekniği de kullanılmıştır.

Tablo 38. Adam Optimizasyon Algoritması Bazlı Karşılaştırmalar

Optimizasyon algoritması	Öğrenme Oranı Başlangıç Değeri	Yığın Boyutu	Görüntü Boyutu	Doğruluk	Kesinlik	Duyarlılık	F1
Adam	0,001	16	64x64	0,8310	0,8367	0,8279	0,8295
Adam	0,001	16	128x128	0,8810	0,8810	0,8756	0,8766
Adam	**0,001**	**16**	**200x200**	**0,8850**	**0,8849**	**0,8784**	**0,8797**
Adam	0,001	16	250x250	0,8550	0,8606	0,8515	0,8536
Adam	0,001	16	300x300	0,8630	0,8651	0,8582	0,8595

Tablo 38'de Adam optimizasyon algoritması ile farklı görüntü boyutlarında sınıflandırma skorları karşılaştırılmıştır. Adam ile en iyi skoru veren model 200x200 boyutunda olup %88,50 doğruluk, %88,49 kesinlik, %87,84 duyarlılık ve %87,97 F1 skorlarına sahiptir.

Tablo 39. Adamax Optimizasyon Algoritması Bazlı Karşılaştırmalar

Optimizasyon algoritması	Öğrenme Oranı Başlangıç Değeri	Yığın Boyutu	Görüntü Boyutu	Doğruluk	Kesinlik	Duyarlılık	F1
Adamax	0,001	16	64x64	0,8490	0,8584	0,8465	0,8480
Adamax	**0,001**	**16**	**128x128**	**0,9140**	**0,9139**	**0,9102**	**0,9112**
Adamax	0,001	16	200x200	0,9010	0,9009	0,8968	0,8975
Adamax	0,001	16	250x250	0,9130	0,9126	0,9085	0,9097
Adamax	0,001	16	300x300	0,9050	0,9058	0,9002	0,9015

Tablo 39'da Adamax optimizasyon algoritması ile yapılan analizlerde en iyi skoru 128x128 boyutunda olan model vermiştir. Model, %91,40 doğruluk, %91,39 kesinlik, %91,02 duyarlılık ve %91,12 F1 skorlarına sahiptir.

Tablo 40. Adagrad Optimizasyon Algoritması Bazlı Karşılaştırmalar

Optimizasyon algoritması	Öğrenme Oranı Başlangıç Değeri	Yığın Boyutu	Görüntü Boyutu	Doğruluk	Kesinlik	Duyarlılık	F1
Adagrad	0,001	16	64x64	0,6860	0,7093	0,6854	0,6838
Adagrad	0,001	16	128x128	0,8120	0,8223	0,8121	0,8138
Adagrad	0,001	16	200x200	0,8270	0,8310	0,8245	0,8253
Adagrad	0,001	16	250x250	0,8380	0,8420	0,8331	0,8336
Adagrad	**0,001**	**16**	**300x300**	**0,8450**	**0,8514**	**0,8415**	**0,8436**

Tablo 40'ta gösterilen Adagrad optimizasyon algoritmasında ise en iyi skoru 300x300 boyutunda olan model vermiş olup, model %84,50 doğruluk, %85,14 kesinlik, %84,15 duyarlılık ve %84,36 F1 skoruna sahiptir.

Tablo 41. RMSProp Optimizasyon Algoritması Bazlı Karşılaştırmalar

Optimizasyon algoritması	Öğrenme Oranı Başlangıç Değeri	Yığın Boyutu	Görüntü Boyutu	Doğruluk	Kesinlik	Duyarlılık	F1
RMSProp	0,001	16	64x64	0,7720	0,7784	0,7655	0,7679
RMSProp	**0,001**	**16**	**128x128**	**0,8390**	**0,8440**	**0,8341**	**0,8360**
RMSProp	0,001	16	200x200	0,8160	0,8129	0,8125	0,8096
RMSProp	0,001	16	250x250	0,8360	0,8405	0,8320	0,8330
RMSProp	0,001	16	300x300	0,6310	0,6800	0,6251	0,6093

Tablo 41'deki RMSProp optimizasyon algoritmasında ise en iyi skoru 128x128 boyutunda olan model vermiştir. Model, %83,90 doğruluk, %84,40 kesinlik, %83,41 duyarlılık ve %83,60 F1 skoruna sahiptir.

Tablo 42. SGD Optimizasyon Algoritması Bazlı Karşılaştırmalar

Optimizasyon algoritması	Öğrenme Oranı Başlangıç Değeri	Yığın Boyutu	Görüntü Boyutu	Doğruluk	Kesinlik	Duyarlılık	F1
SGD	0,001	16	64x64	0,7170	0,7280	0,7180	0,7168
SGD	0,001	16	128x128	0,8290	0,8384	0,8265	0,8278
SGD	0,001	16	200x200	0,8320	0,8372	0,8281	0,8289
SGD	0,001	16	250x250	0,8350	0,8399	0,8309	0,8330
SGD	**0,001**	**16**	**300x300**	**0,8350**	**0,8395**	**0,8326**	**0,8330**

Tablo 42'deki SGD optimizasyon algoritmasında ise en iyi skoru 250x250 boyutunda olan model vermiştir. Model, %83,50 doğruluk, %83,95 kesinlik, %83,26 duyarlılık ve %83,30 F1 skoruna sahiptir.

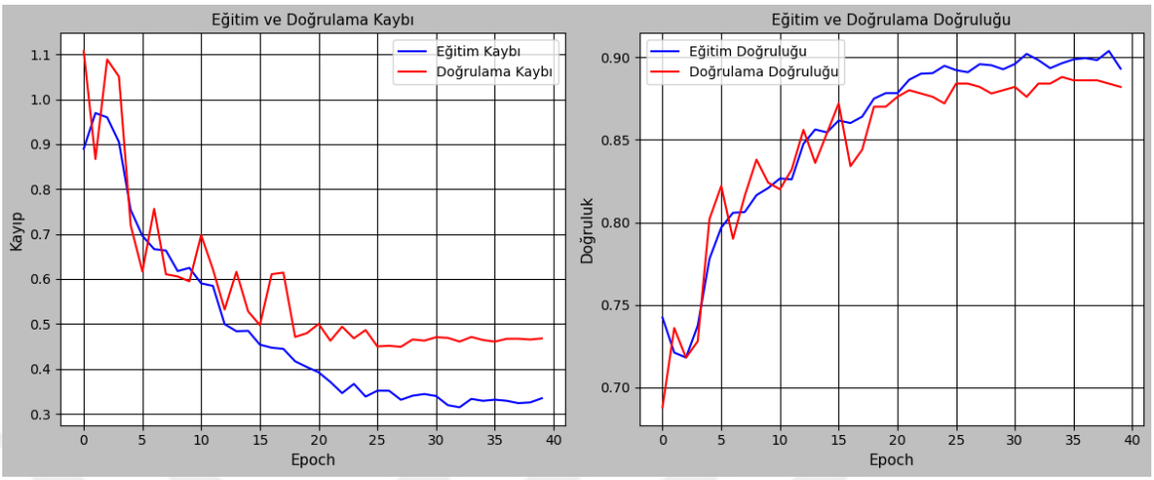
5.1.3. Eğitim Kümesi Bazlı Karşılaştırmalar

Bu bölümdeki analizlerde ise eğitim-doğrulama-test setleri, %90-%5-%5, %80-%10-%10, %70-%15-%15, %60-%20-%20 ve %50-%25-%25 yaklaşımları bazında bölümlendirilerek uygulanmıştır. Her bir analizde, en iyi performansı gösteren bölümlendirme bazlı modelin veri artırımları yapılarak başarımın artırılması hedeflenmiştir. Analizlerde ilk olarak Adam optimizasyon algoritması kullanılmıştır.

Tablo 43. Bölümlendirme Bazlı Karşılaştırmalar 1

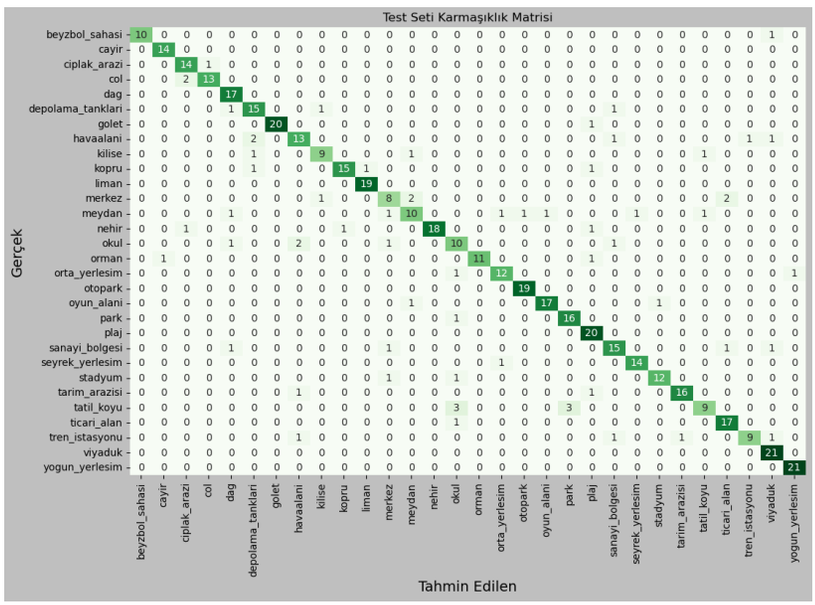
Bölümlendirme (%)	Optimizasyon algoritması	Öğrenme Oranı Başlangıç Değeri	Yığın Boyutu	Görüntü Boyutu	Doğruluk	Kesinlik	Duyarlılık	F1
90-5-5	Adam	0,001	16	224x224	**0,8680**	**0,8689**	**0,8591**	**0,8605**
80-10-10					0,8090	0,8096	0,7985	0,8002
70-15-15					0,8133	0,8104	0,8084	0,8079
60-20-20					0,8080	0,8021	0,7989	0,7976
50-25-25					0,8064	0,8075	0,8029	0,7990

Tablo 43'ten de görüldüğü üzere, en iyi performans gösteren bölümlendirmenin %90-%5-%5 olduğu görülmektedir. Bu ilk modelin doğruluk skoru 0,8680, kesinlik skoru 0,8689, duyarlılık skoru 0,8591, F1 skoru ise 0,8605 olarak gerçekleşmiştir.



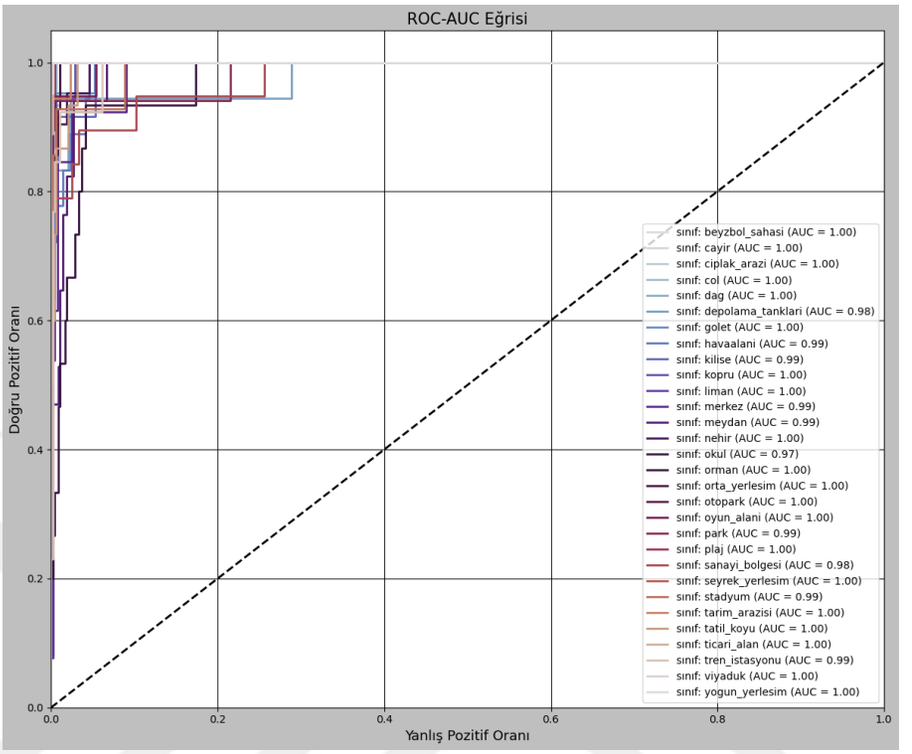
Şekil 152. %90-%5-%5 Bölümlendirmesi Kayıp ve Doğruluk Grafiği

Şekil 152'de %90-%5-%5 yaklaşımıyla oluşturulan modelin kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiği incelendiğinde eğitim kaybının hızla azaldığı ve doğrulama kaybının da hemen hemen aynı eğilimi izlediği görülebilir, ancak doğrulama kaybında dalgalanma fazladır. 20. epochtan sonra eğitim ve doğrulama kaybı arasındaki açıklık, modelin, daha önce görmediği görüntüler üzerinde yeteri kadar genelleme yapamadığını göstermektedir. Başka bir deyişle aşırı öğrenme durumu vardır denilebilir. Sağ taraftaki doğruluk grafiğinde ise eğitim doğruluğunun 40. epoch itibariyle 0,90 seviyesine ulaştığı görülebilir. Doğrulama doğruluğu ise eğitim doğruluğu ile hemen hemen aynı hizadadır ancak 0,88 seviyesinde sabitlenmiştir.



Şekil 153. %90-%5-%5 Bölümlendirmesi Karmaşıklık Matrisi

Şekil 153'te ilk modelin karmaşıklık matrisi görülmektedir. Tatil köyü sınıfının okul sınıfı ile karıştırıldığı 3 örnek ve merkez sınıfının meydan sınıfı olarak tahmin edildiği 2 örnek göze çarpmaktadır. İlk modelin sınıflandırma raporu ise ek 6'da verilmiştir.



Şekil 154. %90-%5-%5 Bölümlendirmesi AUC-ROC Eğrisi

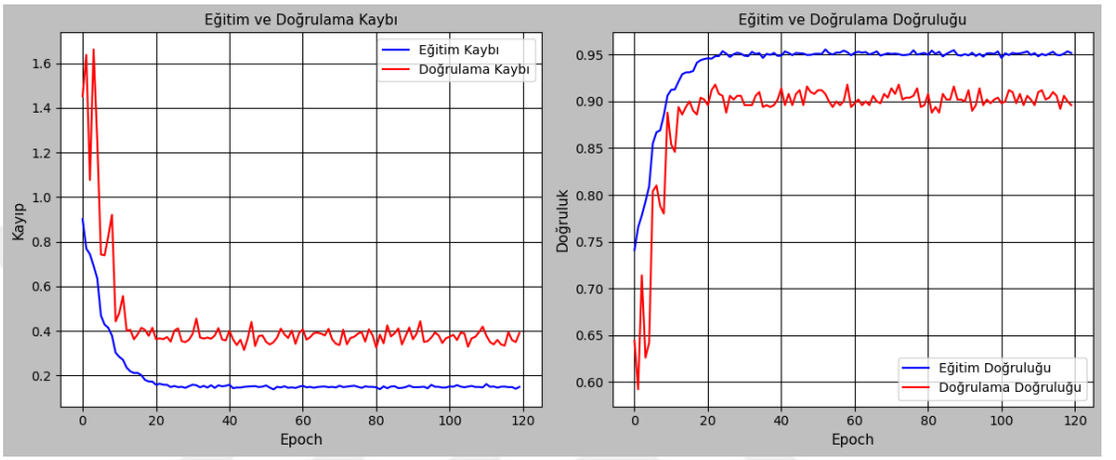
Şekil 154'te en iyi modelin her bir sınıfına ait AUC-ROC eğrileri gösterilmektedir. X ekseni, yanlış pozitif oranını Y ekseni ise doğru pozitif oranını göstermektedir. Düzlemi ikiye ayıran simetrik çizgi ise rastgele tahmini temsil etmektedir (AUC=0,5). Bu çizginin üzerindeki herhangi bir eğri, rastgele sınıflandırmadan daha iyi bir sınıflandırmayı işaret eder. Şekil incelendiğinde modelin, tüm sınıflarda yüksek sınıflandırma performansı gösterdiği görülebilir. Veri setinde bulunan 30 sınıftan 22 tanesi AUC=1,00 skoruna ulaşmıştır. Kalan sınıfların (örneğin, havaalani, okul, meydan, tatil_koyu vb.) AUC skorları 0,95-0,99 arasındadır. Bu skor aralığı da oldukça güçlü bir ayrım gücünü göstermektedir. ROC alanının sol üst köşesine yakın bir yerde gözlemlenen çok düşük FPR'ler ve yüksek TPR'ler, yanlış sınıflandırmaların minimum düzeyde olduğunu göstermektedir. Örneğin; okul sınıfı (AUC=0,95), modeldeki en düşük AUC skorunu göstermektedir. İlgili sınıfın, bazı örtüşen özellikler veya görsel benzerlik nedeniyle model tarafından bir veya daha fazla sınıfla karıştırıldığı söylenebilir.

İlk modelin performansını daha da artırmak için veri artırma tekniği uygulanmış, epoch değeri 120 seçilerek uzun vadeli eğitimlerdeki değişimlerin daha net gözlemlenmesi hedeflenmiştir.

Tablo 44. %90-%5-%5 Bölümlendirmesi Değerlendirme Sonuçları

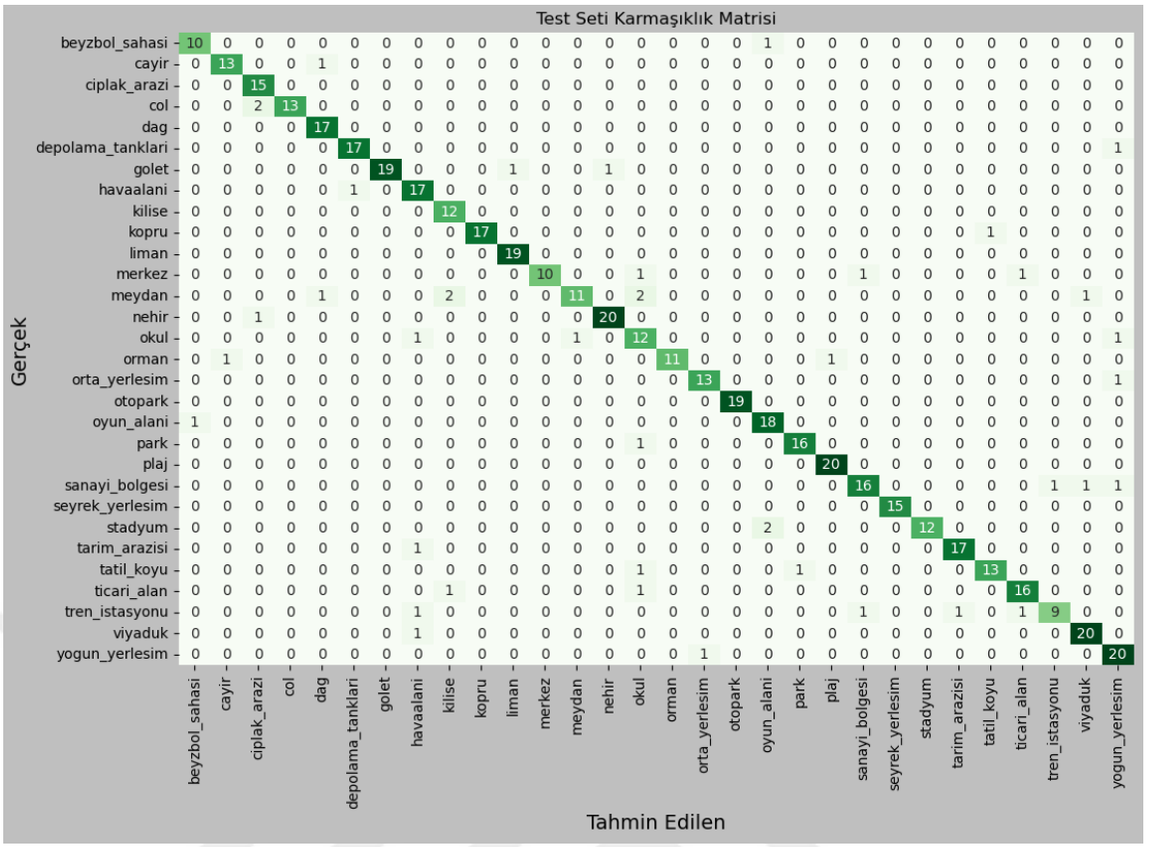
Bölümlendirme (%)	Optimizasyon algoritması	Öğrenme Oranı Başlangıç Değeri	Yığın Boyutu	Görüntü Boyutu	Doğruluk	Kesinlik	Duyarlılık	F1
90-5-5	Adam	0,001	16	224x224	0,9140	0,9193	0,9090	0,9103

Tablo 44'te, veri artırımı yapılmış modelin değerlendirme sonuçları gösterilmektedir. Doğruluk skoru 0,9140, kesinlik 0,9193, duyarlılık 0,9090 ve F1 skoru da 0,9103 olarak gerçekleşmiştir. Doğruluk skoru ilk modele göre yaklaşık %5 artmıştır.



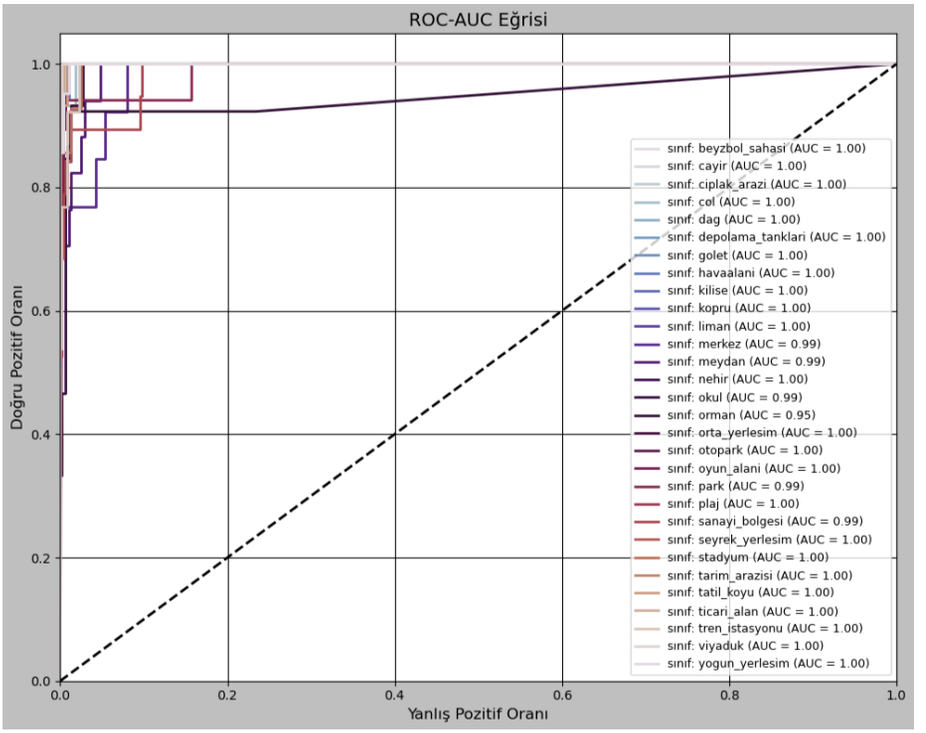
Şekil 155. %90-%5-%5 Veri Artırımı Yapılmış Model Kayıp ve Doğruluk Grafiği

Şekil 155'te veri artırımı yapılmış modelin doğruluk ve kayıp grafikleri gösterilmektedir. Kayıp grafiği incelendiğinde, eğitim kaybının 20. epoch itibariyle düşük bir değer (~0,2) seviyesine yakın olarak sabitlendiği görülmektedir. Doğrulama kaybı da 20. epochtan itibaren oldukça hafif dalgalanmalarla ~0,35 seviyesinde sabitlenmiştir. Doğruluk grafiğinde de benzer şekilde, eğitim doğruluğunun 20. epochtan itibaren ~0,95'te sabitlendiği görülebilir. Doğrulama doğruluğu ise ~0,90 seviyesinde sabitlenmiştir. Doğruluk grafiğindeki açıklık, kayıp grafiğindeki açıklıktan biraz daha fazla olsa da, eğrilerin sabit ve düz olması, modelin aşırı öğrenme yapmadığını göstermektedir.



Şekil 156. %90-%5-%5 Veri Artırımı Yapılmış Model Karmaşıklık Matrisi

Şekil 156'da ise ikinci modelin karmaşıklık matrisi görülmektedir. Meydan sınıfının kilise sınıfı ile karıştırıldığı 2 örnek ve stadyum sınıfının oyun alanı sınıfı olarak tahmin edildiği 2 örnek dikkati çekmektedir. İkinci modelin sınıflandırma raporu ise ek 7'de verilmiştir.



Şekil 157. %90-%5-%5 Veri Artırımı Yapılmış Model AUC-ROC Eğrisi

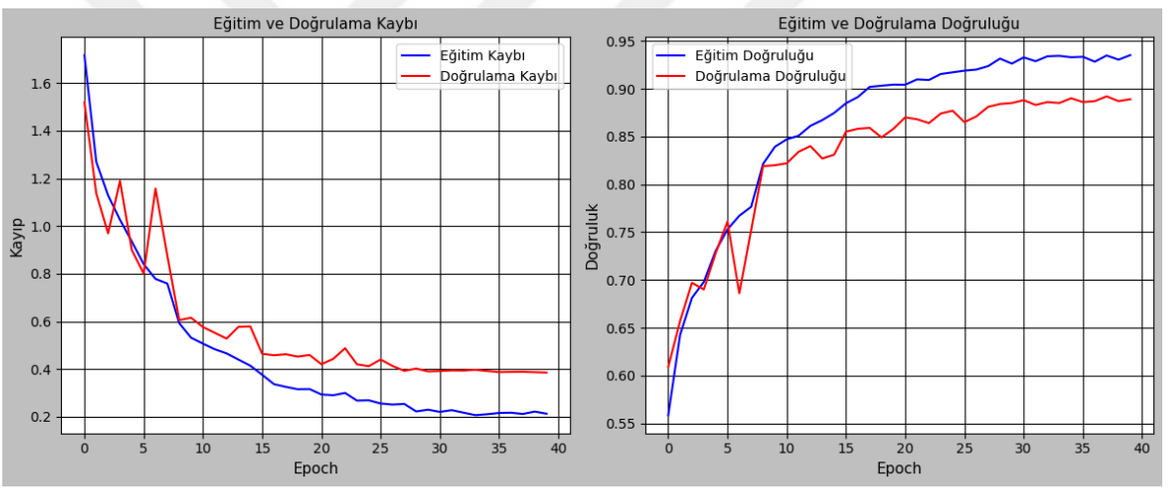
Şekil 157'de ikinci modele ait ROC eğrileri ve AUC skorları gösterilmektedir. Sınıfların çoğunun (24) AUC skoru 1 iken; 5 sınıfın 0,99 ve 1 sınıfın da 0,95'tir.

Analizlerde ikinci olarak Adamax optimizasyon algoritması kullanılmıştır.

Tablo 45. Bölümlendirme Bazlı Karşılaştırmalar 2

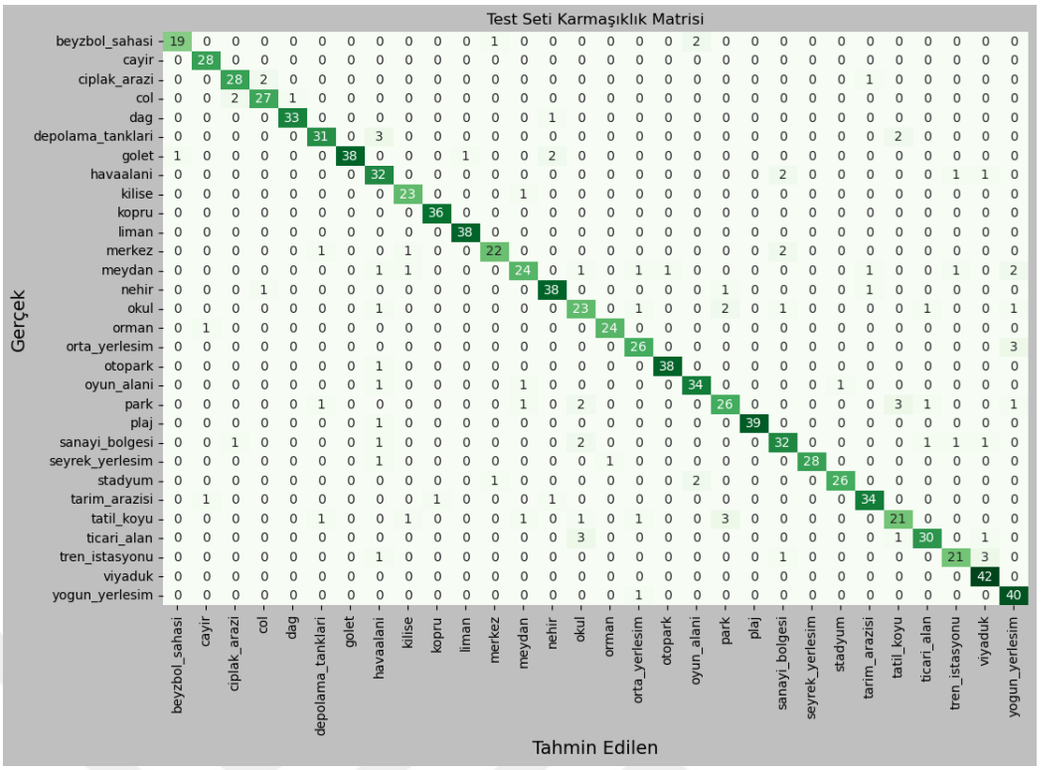
Bölümlendirme (%)	Optimizasyon algoritması	Öğrenme Oranı Başlangıç Değeri	Yığın Boyutu	Görüntü Boyutu	Doğruluk	Kesinlik	Duyarlılık	F1
90-5-5	Adamax	0,001	16	224x224	0,8960	0,9016	0,8890	0,8894
80-10-10					**0,9010**	**0,9021**	**0,8973**	**0,8984**
70-15-15					0,8913	0,8914	0,8885	0,8887
60-20-20					0,8890	0,8880	0,8850	0,8850
50-25-25					0,8756	0,8756	0,8726	0,8729

Tablo 45'te, en başarılı performansı gösteren model bölümlendirmesinin %80-%10-%10 olduğu görülmektedir. Model, 0,9010'luk doğruluk, 0,9021'lik kesinlik, 0,8973 duyarlılık ve 0,8984'lük F1 skorlarına sahiptir.



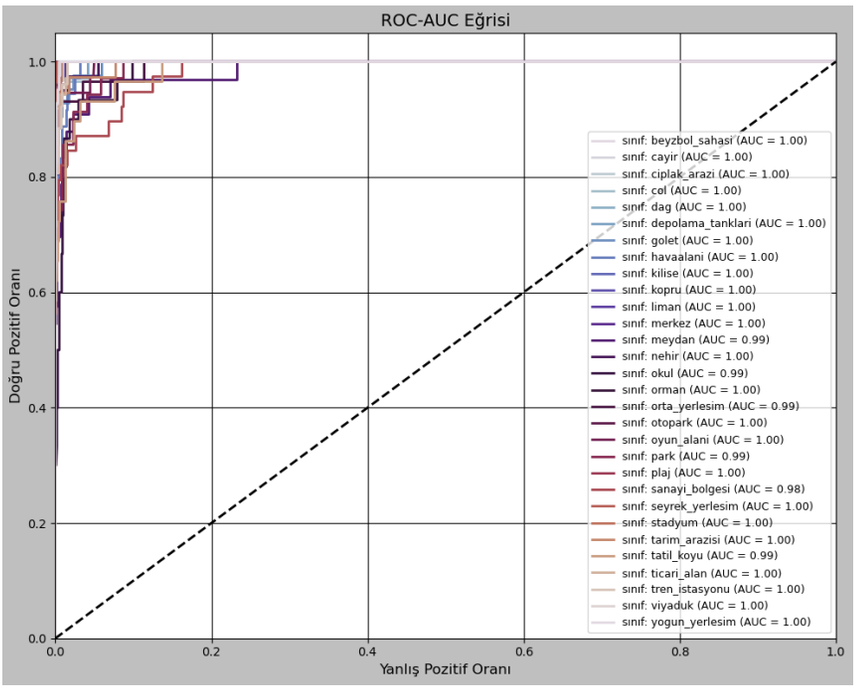
Şekil 158. %80-%10-%10 Bölümlendirmesi Kayıp ve Doğruluk Grafiği

Şekil 158'de sol tarafta görülen kayıp grafiğinde eğitim kaybının 40. epoch itibariyle 0,20 seviyesinde sabitlendiği görülmektedir. Doğrulama kaybı ise hemen hemen 0,40'ta sabitlenmiştir. Sağ taraftaki doğruluk grafiğinde ise eğitim doğruluğu 0,94 seviyesinde görünmekteyken doğrulama doğruluğu da 0,89 seviyesinde durmuştur. Temel modelde, bu durumda herhangi bir aşırı öğrenmenin olduğu söylenemez. Her iki grafikteki açıklıkların farkı kabul edilebilir seviyededir.



Şekil 159. %80-%10-%10 Bölümlendirmesi Karmaşıklık Matrisi

Şekil 159'da ilk modelin karmaşıklık matrisi görülmektedir. Tatil köyü sınıfının park sınıfı ile karıştırıldığı 3 örnek ve orta yerleşim sınıfının yoğun yerleşim sınıfı olarak tahmin edildiği 3 örnek göze çarpmaktadır. İlk modelin sınıflandırma raporu ise ek 8'de verilmiştir.



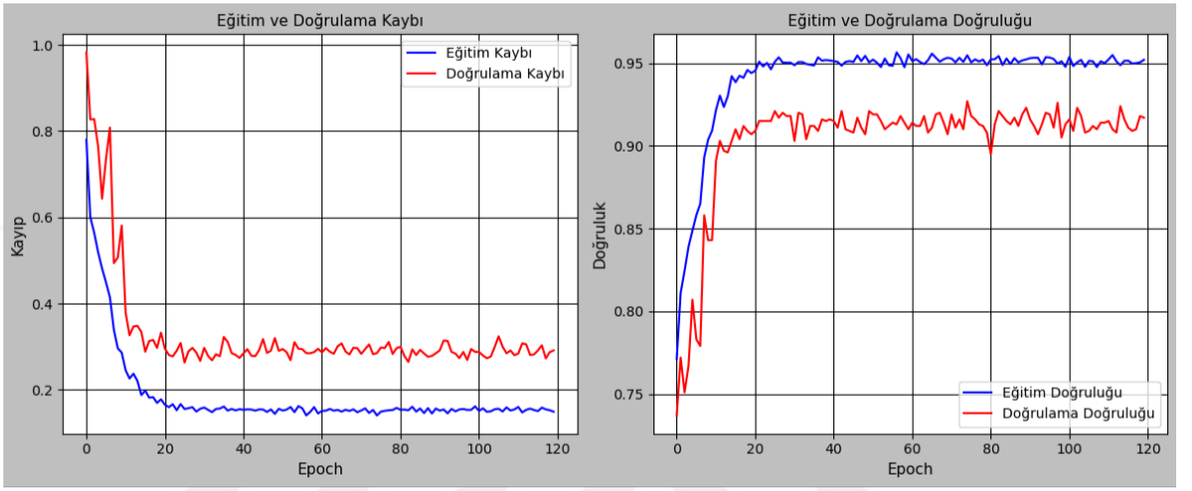
Şekil 160. %80-%10-%10 Bölümlendirmesi AUC-ROC Eğrisi

Şekil 160'ta ilk modele ait ROC eğrileri ve AUC skorları gösterilmektedir. Sınıfların çoğunun (24) AUC skoru 1 iken; 5 sınıfın 0,99 ve 1 sınıfın da 0,98'dir.

Tablo 46. %80-%10-%10 Bölümlendirmesi Değerlendirme Sonuçları

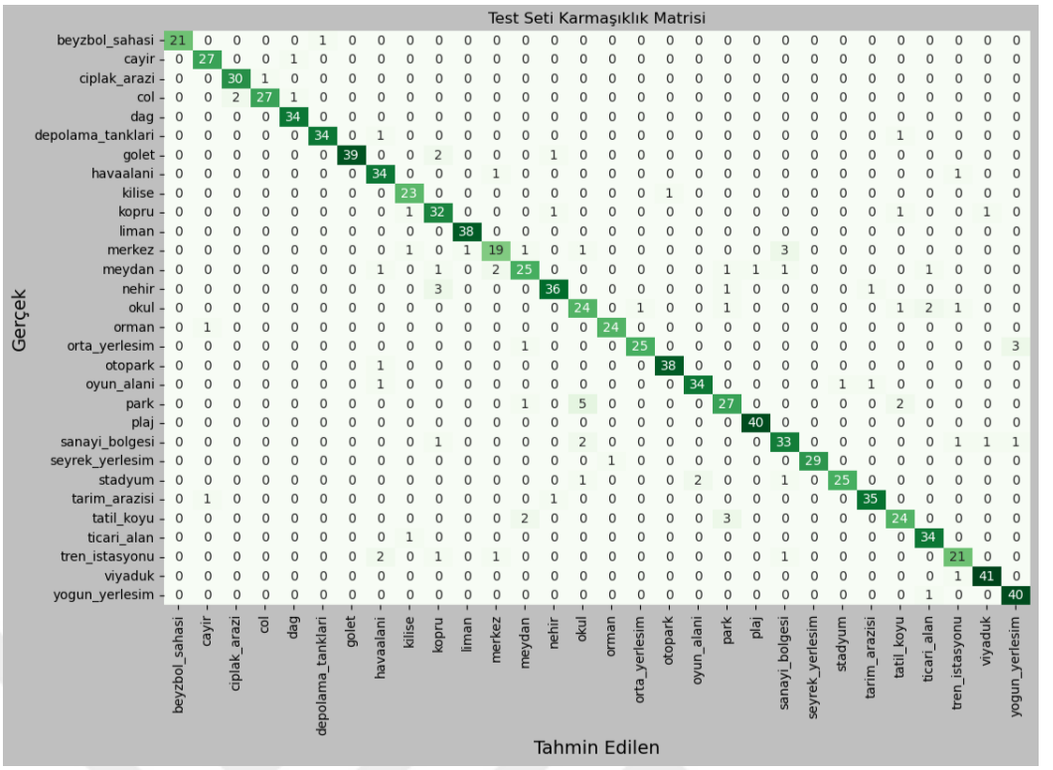
Bölümlendirme (%)	Optimizasyon algoritması	Öğrenme Oranı Başlangıç Değeri	Yığın Boyutu	Görüntü Boyutu	Doğruluk	Kesinlik	Duyarlılık	F1
80-10-10	Adamax	0,001	16	224x224	0,9130	0,9127	0,9095	0,9101

Tablo 46'da ise en iyi modelin veri artırımı yapılmış versiyonunun değerlendirme sonuçları gösterilmektedir. 0,9130'luk doğruluk, 0,9127'lik kesinlik, 0,9095'lik duyarlılık ve 0,9101'lik F1 skoru gerçekleşmiştir. Doğruluk skoru ilk modele göre yaklaşık %1,2 oranında artmıştır.



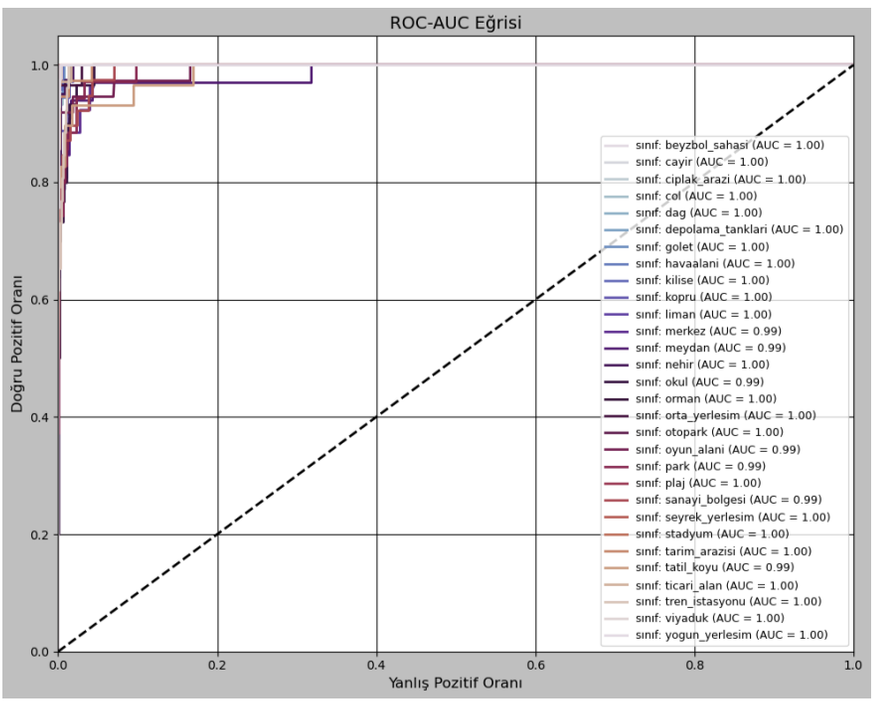
Şekil 161. %80-%10-%10 Veri Artırımı Yapılmış Model Kayıp ve Doğruluk Grafiği

Şekil 161'de ise ikinci modele ait kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiği incelendiğinde, eğitim kaybının 20. epochtan itibaren 0,10 seviyelerinde sabitlendiği görülmektedir. Doğrulama kaybı da 0,30 seviyelerinde sabitlenmiştir. Bu durum, modelin istikrarlı ve stabil olduğunu göstermektedir. Sağ taraftaki doğruluk grafiğinde ise eğitim doğruluğunun yine 20. epoch itibariyle 0,95 seviyelerinde sabitlendiği görülmektedir. Doğrulama doğruluğu ise nispeten biraz daha fazla dalgalanma gösterse de 0,92 seviyelerinde sabitlenmektedir. Her iki grafikte de yaklaşık 0,20'lik açıklık farkı, modelin iyi genelleştirildiğini göstermektedir. Yine de, veri artırımının modeli az da olsa iyileştirdiği söylenebilir.



Şekil 162. %80-%10-%10 Veri Artırımı Yapılmış Model Karmaşıklık Matrisi

Şekil 162'de ikinci modelin karmaşıklık matrisi görülmektedir. Park sınıfının okul sınıfı ile karıştırıldığı 5 örnek ve nehir sınıfının köprü sınıfı olarak tahmin edildiği 3 örnek bulunmaktadır. İkinci modelin sınıflandırma raporu ise ek 9'da verilmiştir. Veri artırımı yapılmış modelin karmaşıklık matrisinde de görüldüğü üzere ilk model ile hemen hemen aynı sınıflandırma performansının gerçekleştiği söylenebilir.



Şekil 163. %80-%10-%10 Veri Artırımı Yapılmış Model AUC-ROC Eğrisi

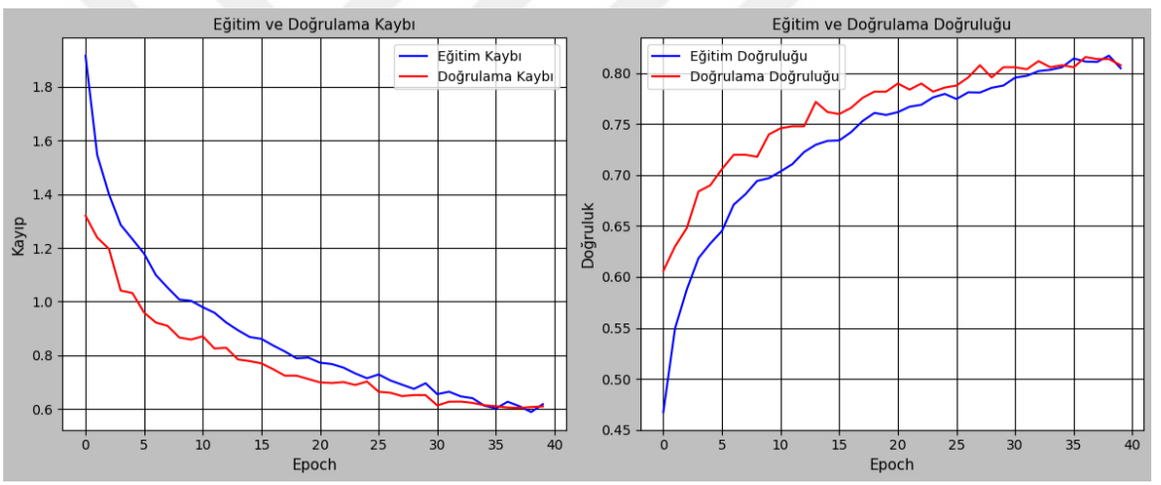
Şekil 163'te ikinci modele ait ROC eğrileri ve AUC skorları gösterilmektedir. Sınıfların çoğunun (23) AUC skoru 1 iken; 7 sınıfın 0,99'dur.

Üçüncü olarak, Adagrad optimizasyon algoritması kullanılmıştır.

Tablo 47. Bölümlendirme Bazlı Karşılaştırmalar 3

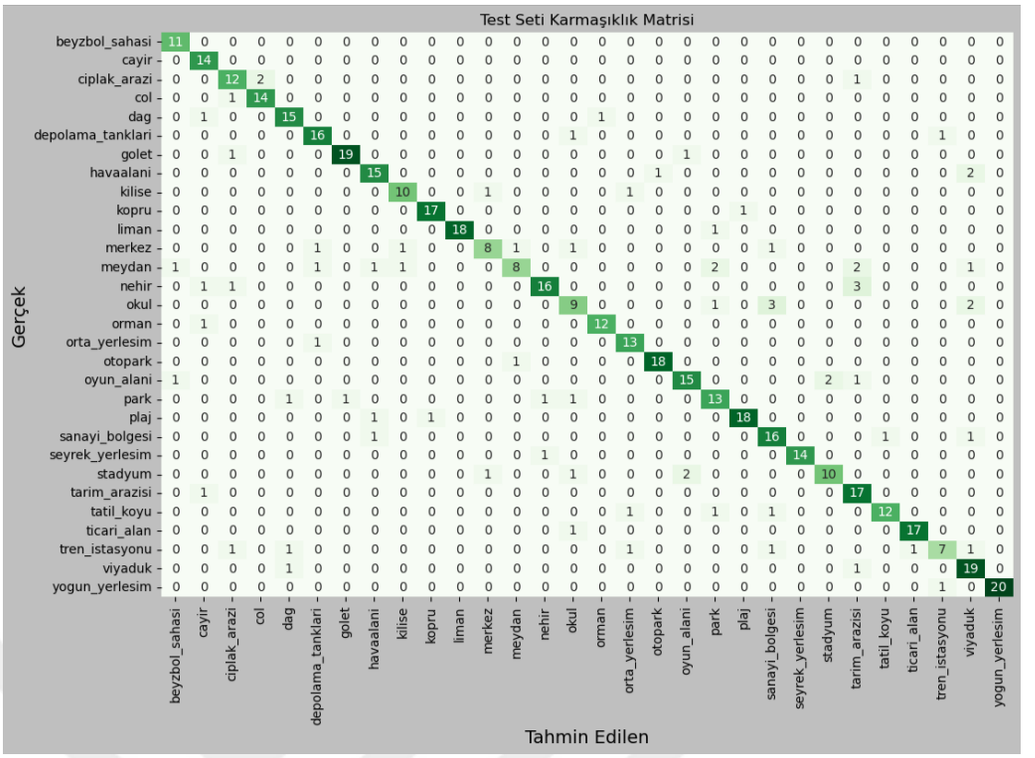
Bölümlendirme (%)	Optimizasyon algoritması	Öğrenme Oranı Başlangıç Değeri	Yığın Boyutu	Görüntü Boyutu	Doğruluk	Kesinlik	Duyarlılık	F1
90-5-5	Adagrad	0,001	16	224x224	**0,8460**	**0,8473**	**0,8414**	**0,8391**
80-10-10					0,8160	0,8213	0,8137	0,8138
70-15-15					0,7880	0,7980	0,7835	0,7839
60-20-20					0,7805	0,7878	0,7764	0,7760
50-25-25					0,7356	0,7470	0,7310	0,7303

Tablo 47'de, en başarılı model bölümlendirmesinin %90-%5-%5 olduğu görülmektedir. Adam ve Adamax'a göre biraz daha fazla performans kaybı yaşanan Adagrad kullanımında, doğruluk 0,8460, kesinlik 0,8473, duyarlılık 0,8414 ve F1 0,8391 skorları almıştır.



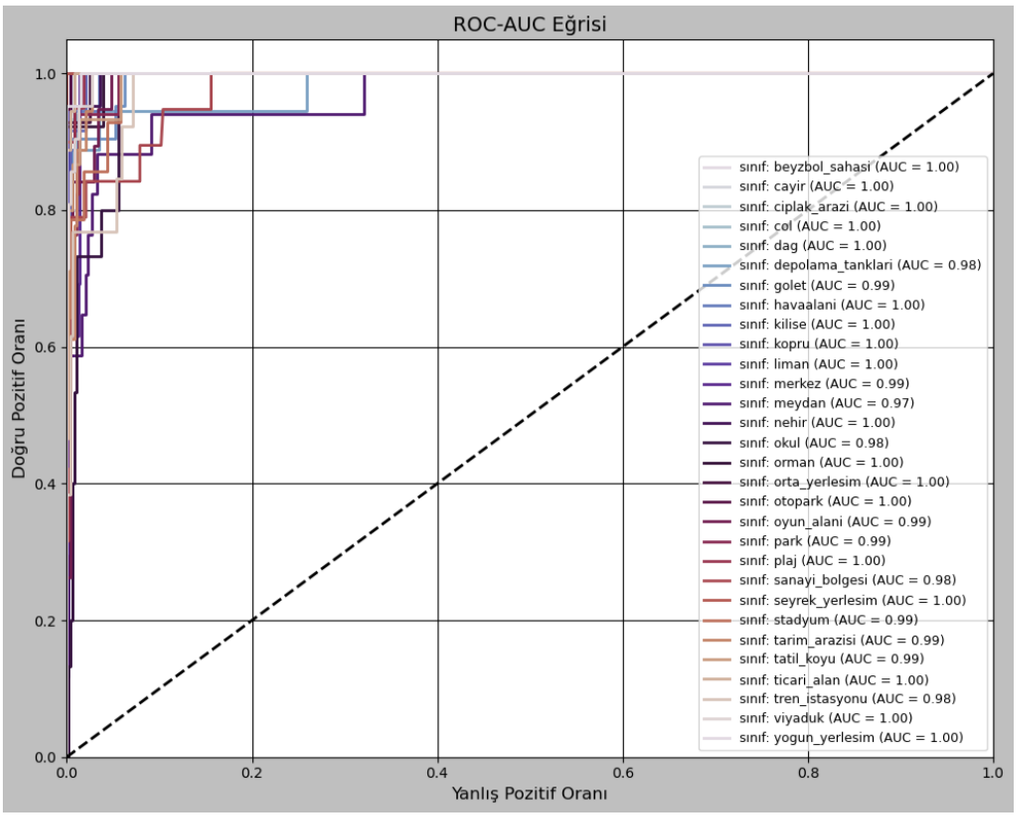
Şekil 164. %90-%5-%5 Bölümlendirmesi Kayıp ve Doğruluk Grafiği

Şekil 164'te ise ilk modele ait kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiği incelendiğinde, eğitim kaybının 40 epoch sonunda 0,60'ta sabitlendiği görülmektedir. Doğrulama kaybında ise bu oran da hemen hemen aynıdır. Sağ taraftaki doğruluk grafiğinde ise hem eğitim hem de doğrulama doğruluğunun 40 epoch sonunda 0,81 seviyesinde birleştikleri görülmektedir. Her iki grafikte de, açıklıkların 25. epochtan itibaren birleşmeye başladıkları görülse de, kayıp skorları yüksek kalmakta, doğruluk skorları ise 0,80 seviyesinde bulunmaktadır. Başka bir deyişle, Adagrad optimizasyon algoritması kullanılan model, diğer iki modele göre daha düşük genel performans ve genelleme yeteneğine sahip olmuştur.



Şekil 165. %90-%5-%5 Bölümlendirmesi Karmaşıklık Matrisi

Şekil 165'te ilk modelin karmaşıklık matrisi görülmektedir. Okul sınıfının sanayi bölgesi sınıfı ile karıştırıldığı 3 örnek ve nehir sınıfının tarım arazisi sınıfı olarak tahmin edildiği 3 örnek bulunmaktadır. İlk modelin sınıflandırma raporu ise ek 10'da verilmiştir.



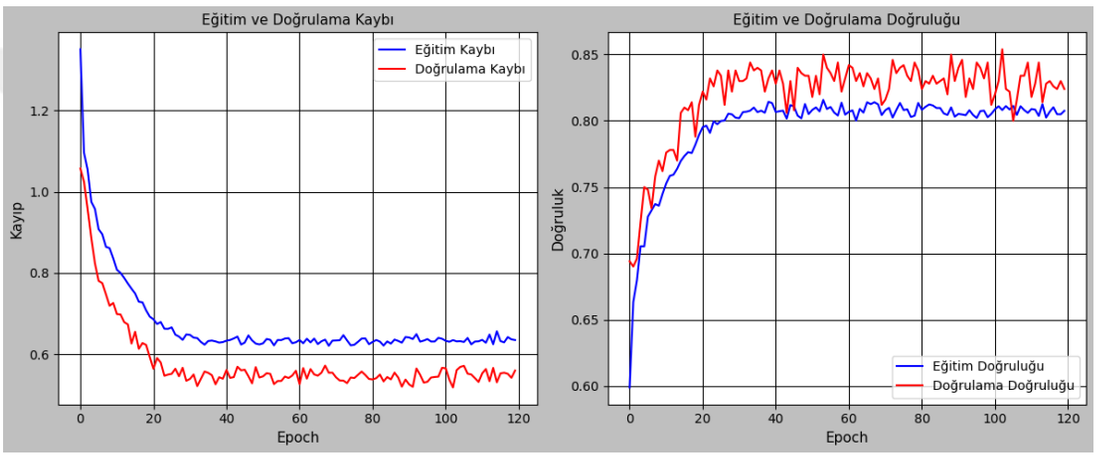
Şekil 166. %90-%5-%5 Bölümlendirmesi AUC-ROC Eğrisi

Şekil 166'da ilk modele ait ROC eğrileri ve AUC skorları gösterilmektedir. Sınıfların çoğunun (18) AUC skoru 1 iken; 7 sınıfın 0,99, 4 sınıfın 0,98 ve 1 sınıfın 0,97'dir.

Tablo 48. %90-%5-%5 Bölümlendirmesi Değerlendirme Sonuçları

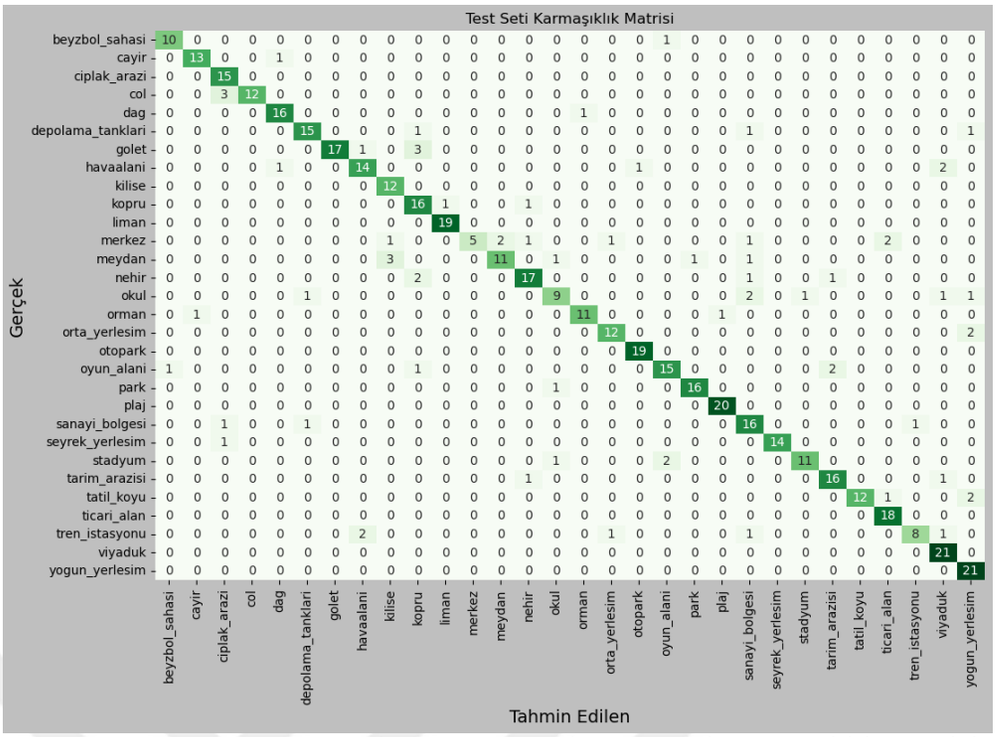
Bölümlendirme (%)	Optimizasyon algoritması	Öğrenme Oranı Başlangıç Değeri	Yığın Boyutu	Görüntü Boyutu	Doğruluk	Kesinlik	Duyarlılık	F1
90-5-5	Adagrad	0,001	16	224x224	0,8620	0,8754	0,8543	0,8544

Tablo 48'de ise en iyi modelin veri artırımı yapılmış versiyonunun değerlendirme sonuçları gösterilmektedir. Doğruluk skoru 0,8620, kesinlik 0,8754, duyarlılık 0,8543 ve F1 skoru ise 0,8544 olarak gerçekleşmiştir. Modelin doğruluk skoru, ilk modele göre yaklaşık %1,2 oranında artmıştır.



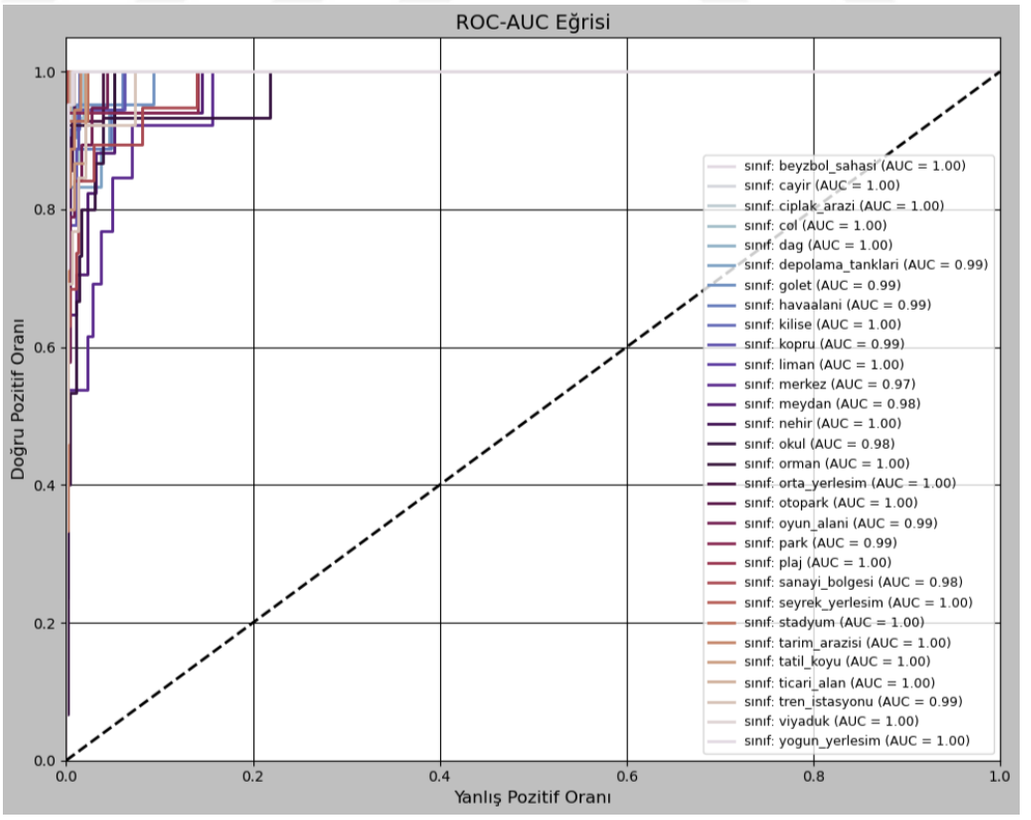
Şekil 167. %90-%5-%5 Veri Artırımı Yapılmış Model Kayıp ve Doğruluk Grafiği

Şekil 167'de ise ikinci modele ait kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde eğitimn kaybının 20. epochtan itibaren 0,62 seviyelerinde sabitleşmeye başladığı görülmektedir. Bu değer, doğrulama kaybı için 0,52 seviyelerindedir. Sağ tarafta bulunan doğruluk grafiğinde ise eğitim ve doğrulama doğruluklarının da 20. epochtan itibaren 0,81-0,85 arasında sabitleştiği görülebilmektedir. Her iki grafikte de, eğriler arasındaki açıklığın oldukça az olması, modelin stabilliğini ve aşırı öğrenme durumunun olmadığını gösterse de, eğitim doğruluğunun doğrulama doğruluğunun altında kalması ve nispeten düşük eğitim ve doğrulama skorları, Adagrad'ın yapısından kaynaklanmaktadır.



Şekil 168. %90-%5-%5 Veri Artırımı Yapılmış Model Karmaşıklık Matrisi

Şekil 168'de ikinci modelin karmaşıklık matrisi görülmektedir. Gölet sınıfının köprü sınıfı ile karıştırıldığı 3 örnek ve çöl sınıfının çıplak arazi sınıfı olarak tahmin edildiği 3 örnek bulunmaktadır. İlk modelin sınıflandırma raporu ise ek 11'de verilmiştir.



Şekil 169. %90-%5-%5 Veri Artırımı Yapılmış Model AUC-ROC Eğrisi

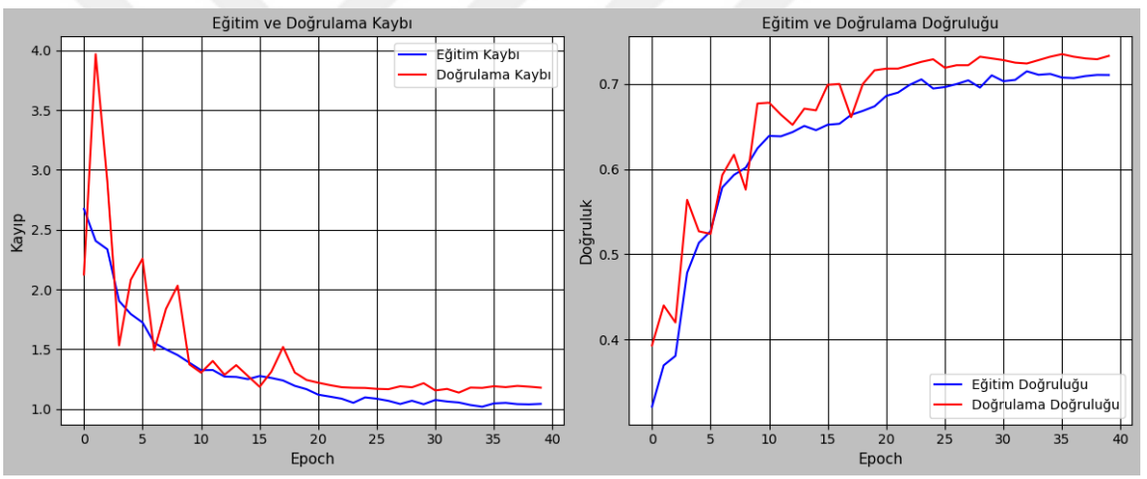
Şekil 169'da ikinci modele ait ROC eğrileri ve AUC skorları gösterilmektedir. Sınıfların çoğunun (18) AUC skoru 1 iken; 7 sınıfın 0,99, 3 sınıfın 0,98 ve 1 sınıfın 0,97'dir.

Dördüncü olarak, RMSProp optimizasyon algoritması kullanılmıştır.

Tablo 49. Bölümlendirme Bazlı Karşılaştırmalar 4

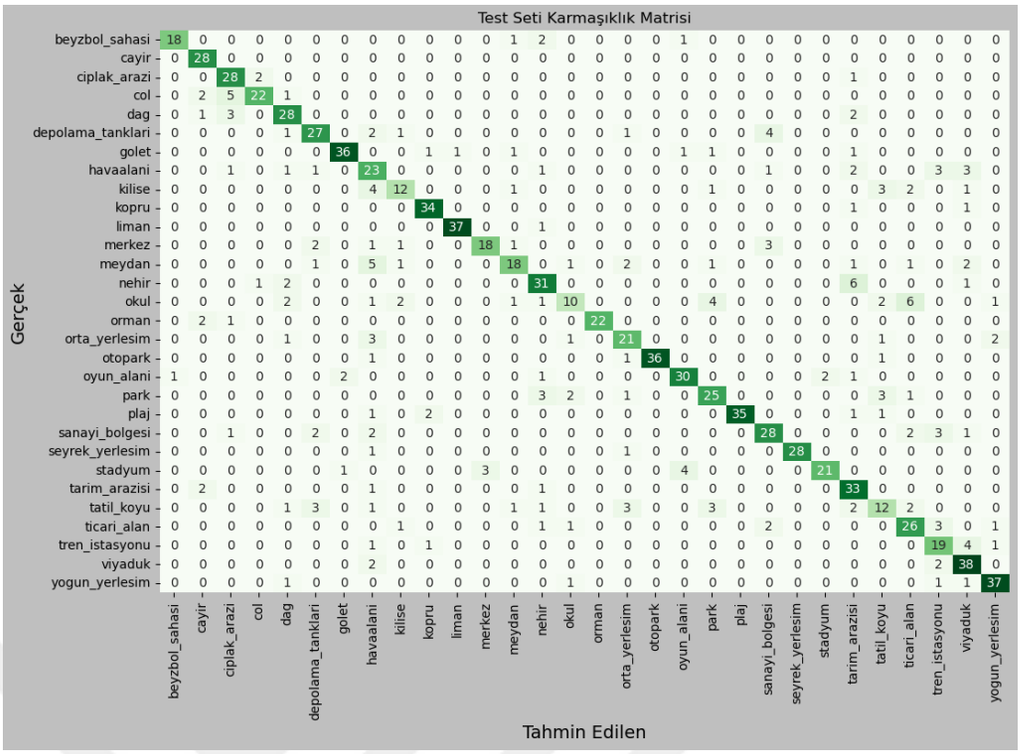
Bölümlendirme (%)	Optimizasyon algoritması	Öğrenme Oranı Başlangıç Değeri	Yığın Boyutu	Görüntü Boyutu	Doğruluk	Kesinlik	Duyarlılık	F1
90-5-5	RMSProp	0,001	16	224x224	0,7480	0,7522	0,7392	0,7365
80-10-10					**0,7810**	**0,7885**	**0,7720**	**0,7736**
70-15-15					0,7547	0,7582	0,7470	0,7468
60-20-20					0,7175	0,7332	0,7111	0,7115
50-25-25					0,7560	0,7560	0,7506	0,7431

Tablo 49'dan da görüldüğü üzere, en iyi performansı gösteren model %80-%10-%10 bölümlendirmesidir. Diğer üç modele göre düşük kalan model, 0,7810'luk doğruluk, 0,7885'lik kesinlik, 0,7720'lik duyarlılık ve 0,7736'lık F1 skorlarına sahiptir.



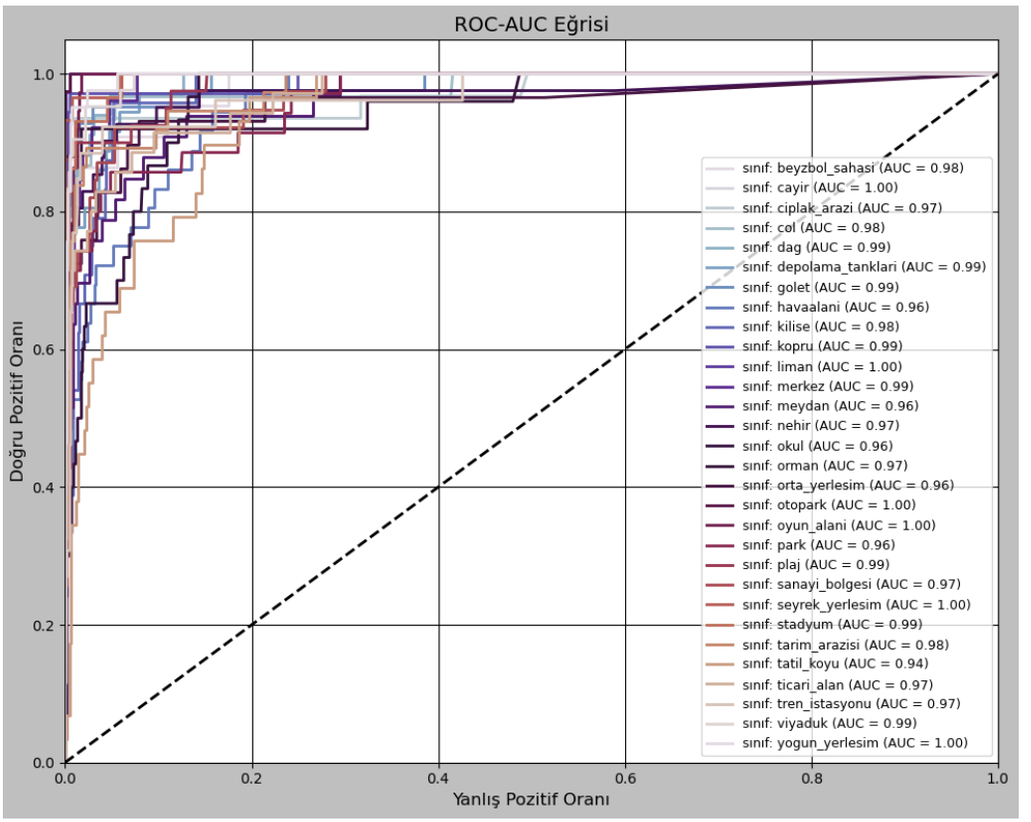
Şekil 170. %80-%10-%10 Bölümlendirmesi Kayıp ve Doğruluk Grafiği

Şekil 170'te ise ilk modele ait kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde eğitim kaybının, 23. epoch itibariyle 1,0 seviyesinde sabitlendiği görülmektedir. Doğrulama kaybı ise, 40 epoch sonunda 1,2 seviyesinde sabit kalmıştır. Model kayıp değerlerindeki bu yükseklik, doğruluk skorlarını da etkilemektedir. Kayıp grafiği bağlamında modelin yüksek kaybına rağmen aşırı öğrenme durumuna girmediği söylenebilir. Sağ tarafta bulunan doğruluk grafiğinde ise eğitim doğruluğu 40 epoch sonucunda 0,71'e ulaşmış görünmektedir. Bu değer doğrulama doğruluğunda 0,74 seviyesindedir. Model, tam olarak orta düzeyde bir performans göstermektedir.



Şekil 171. %80-%10-%10 Bölümlendirmesi Karmaşıklık Matrisi

Şekil 171'de ilk modelin karmaşıklık matrisi görülmektedir. Nehir sınıfının tarım arazisi sınıfı ile karıştırıldığı 6 örnek ve okul sınıfının ticari alan sınıfı olarak tahmin edildiği 6 örnek bulunmaktadır. İlk modelin sınıflandırma raporu ise ek 12'de verilmiştir.



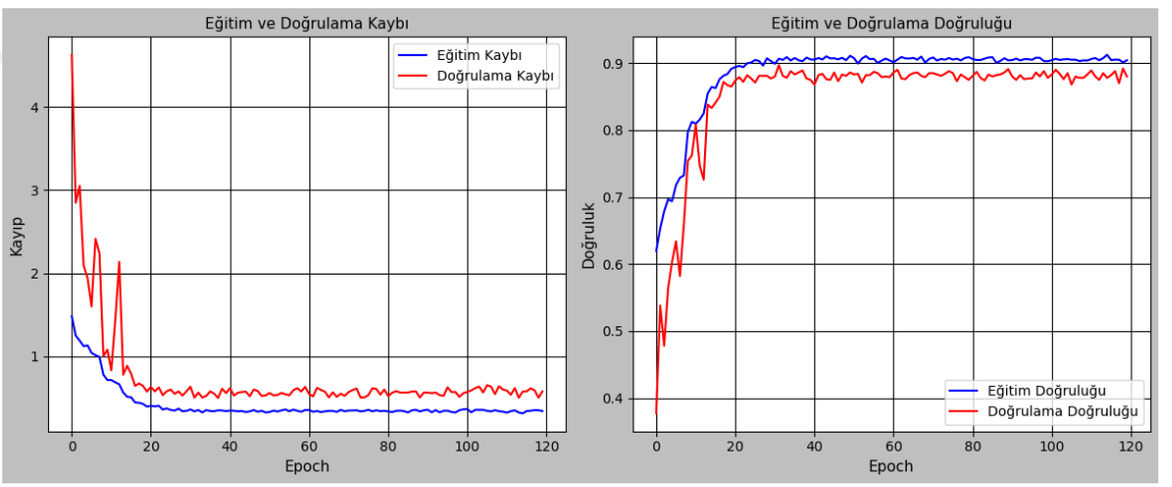
Şekil 172. %80-%10-%10 Bölümlendirmesi AUC-ROC Eğrisi

Şekil 172'de ilk modele ait ROC eğrileri ve AUC skorları gösterilmektedir. Sınıfların bir kısmının (6) AUC skoru 1 iken; 8 sınıfın 0,99, 4 sınıfın 0,98 ve 6 sınıfın 0,97, 5 sınıfın 0,96 ve 1 sınıfın 0,94'tür.

Tablo 50. %80-%10-%10 Bölümlendirmesi Değerlendirme Sonuçları

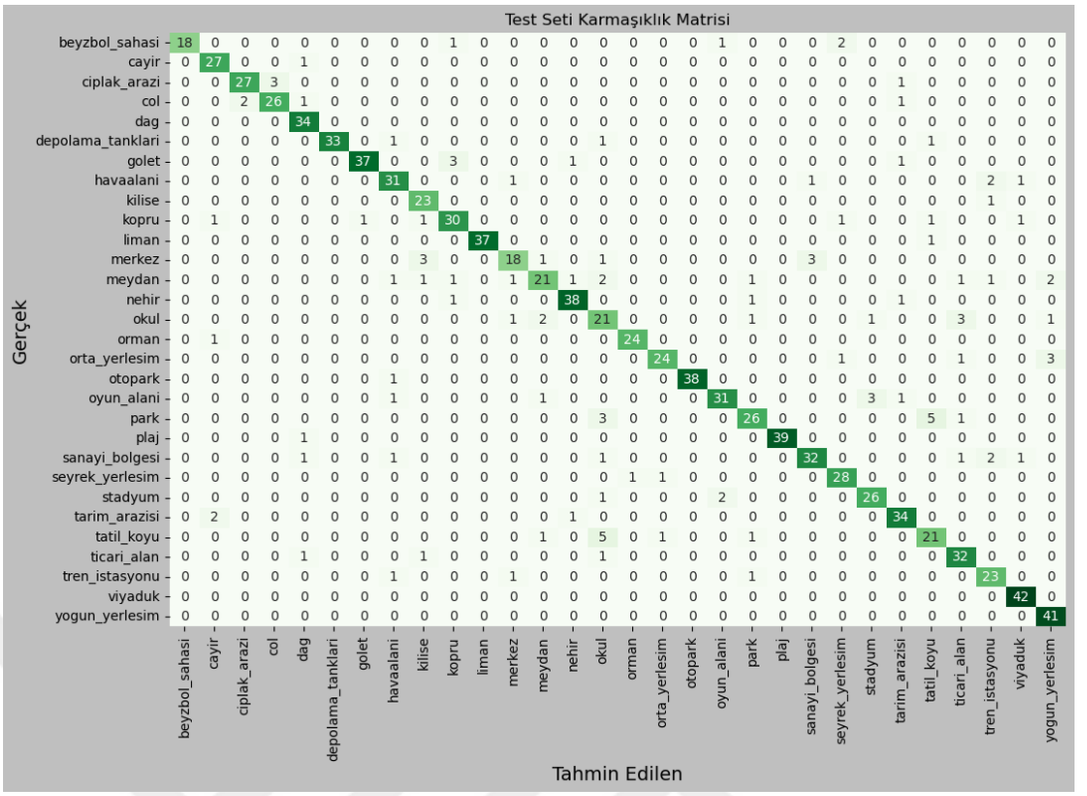
Bölümlendirme (%)	Optimizasyon algoritması	Öğrenme Oranı Başlangıç Değeri	Yığın Boyutu	Görüntü Boyutu	Doğruluk	Kesinlik	Duyarlılık	F1
80-10-10	RMSProp	0,001	16	224x224	0,8820	0,8808	0,8770	0,8765

Tablo 50'den de görüldüğü üzere veri artırımı yapılan modelin başarımının %10 arttığı görülmektedir. Doğruluk skoru 0,8820, kesinlik 0,8808, duyarlılık 0,8770 ve F1 skoru 0,8765 olarak gerçekleşmiştir.



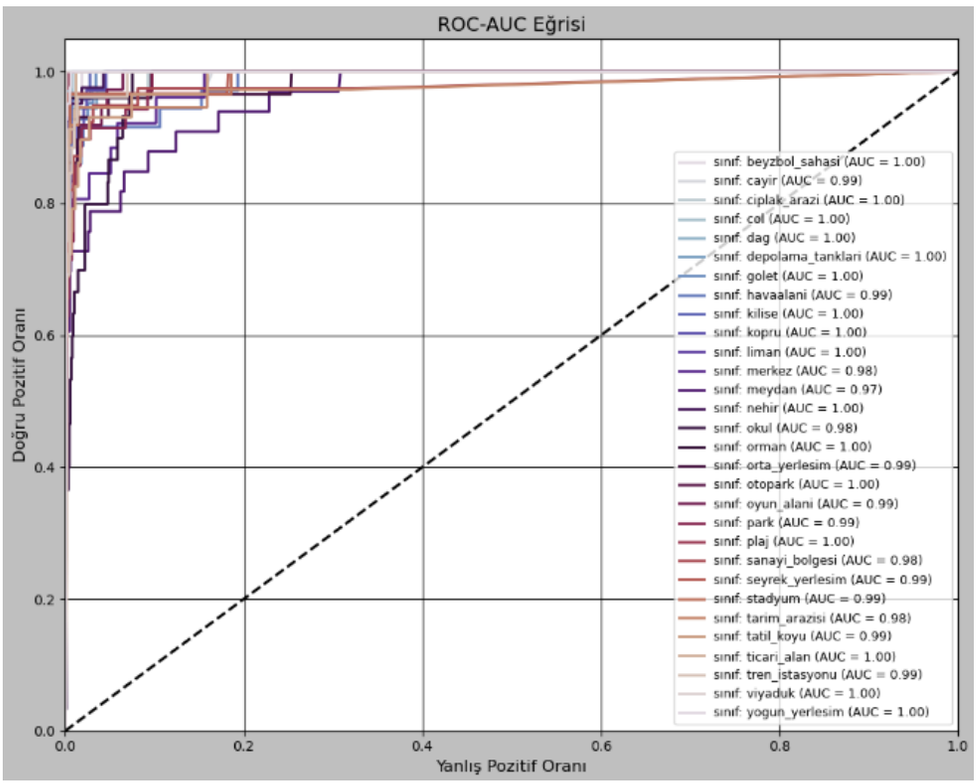
Şekil 173. %80-%10-%10 Veri Artırımı Yapılmış Model Kayıp ve Doğruluk Grafiği

Şekil 173'te ise ikinci modele ait kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde, eğitim kaybı ile doğrulama kaybının 20. epochtan itibaren 0,4-0,5 seviyesinde sabit kaldığı görülebilir. Sağ tarafta bulunan doğruluk grafiğinde ise hem eğitim hem de doğrulama eğrilerinin 0,90 seviyesinde sabit kaldığı görülmektedir. Bu durum, hem aşırı öğrenmenin olmadığını hem de ilk modele göre daha iyi bir genelleştirme özelliğine sahip olduğunu göstermektedir.



Şekil 174. %80-%10-%10 Veri Artırımı Yapılmış Model Karmaşıklık Matrisi

Şekil 174'te ikinci modelin karmaşıklık matrisi görülmektedir. Tatil köyü sınıfının okul sınıfı ile karıştırıldığı 5 örnek ve park sınıfının tatil köyü sınıfı olarak tahmin edildiği 5 örnek bulunmaktadır. İlk modelin sınıflandırma raporu ise ek 13'te verilmiştir.



Şekil 175. %80-%10-%10 Veri Artırımı Yapılmış Model AUC-ROC Eğrisi

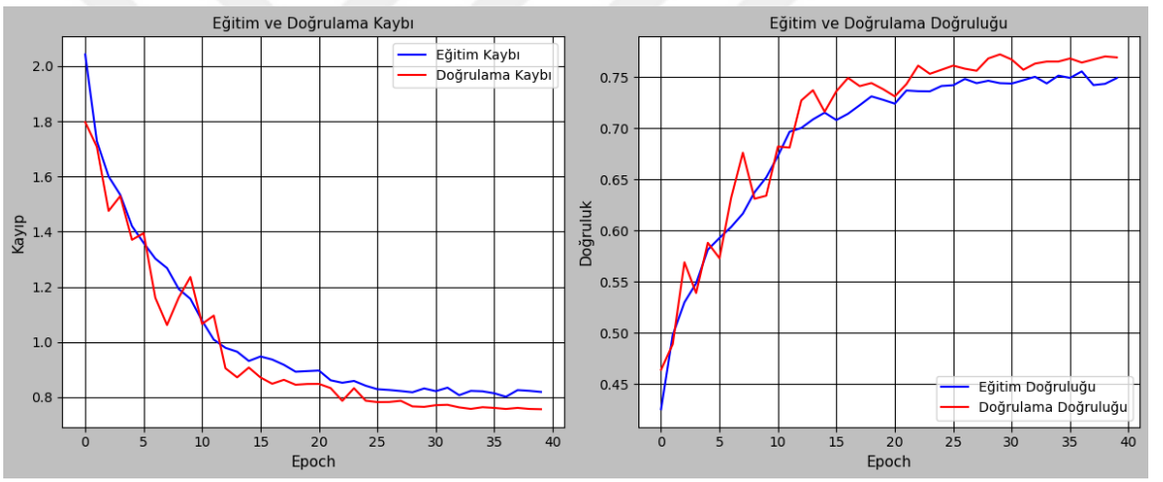
Şekil 175'te ikinci modele ait ROC eğrileri ve AUC skorları gösterilmektedir. Sınıfların bir kısmının (16) AUC skoru 1 iken; 9 sınıfın 0,99, 4 sınıfın 0,98 ve 1 sınıfın 0,97'dir.

Beşinci olarak SGD optimizasyon algoritması kullanılmıştır.

Tablo 51. Bölümlendirme Bazlı Karşılaştırmalar 5

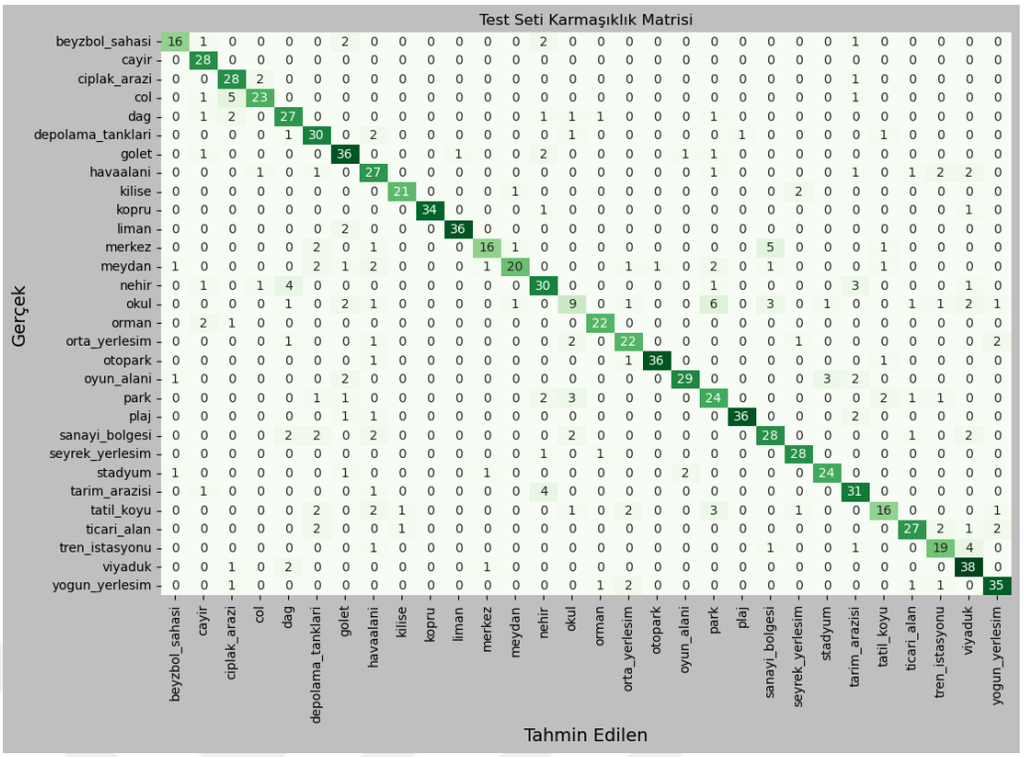
Bölümlendirme (%)	Optimizasyon algoritması	Öğrenme Oranı Başlangıç Değeri	Yığın Boyutu	Görüntü Boyutu	Doğruluk	Kesinlik	Duyarlılık	F1
90-5-5	SGD	0,001	16	224x224	0,7900	0,7999	0,7856	0,7851
80-10-10					**0,7960**	**0,7997**	**0,7904**	**0,7904**
70-15-15					0,7353	0,7494	0,7319	0,7327
60-20-20					0,7010	0,7166	0,6981	0,6943
50-25-25					0,7080	0,7193	0,7044	0,7007

Tablo 51'e göre %80-%10-%10 bölümlendirmesinin daha başarılı sonuç verdiği görülmektedir. Modelin doğruluk skoru 0,7960, kesinliği 0,7997, duyarlılığı 0,7904 ve F1 skoru 0,7904 olarak gerçekleşmiştir.



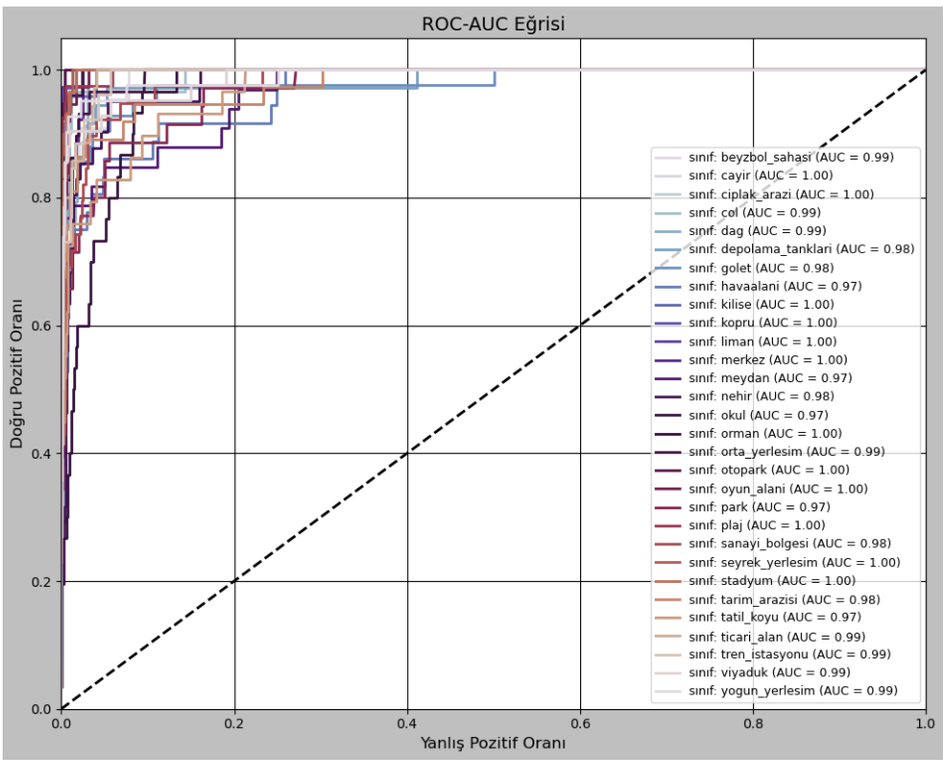
Şekil 176. %80-%10-%10 Bölümlendirmesi Kayıp ve Doğruluk Grafiği

Şekil 176'da ise ilk modele ait kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta gösterilen kayıp grafiğinde, eğitim ve doğrulama kaybının 40 epoch sonunda 0,8 seviyesinde kaldığı görülmektedir. Sağ tarafta bulunan doğruluk grafiğinde ise eğitim doğruluğunun 0,75, doğrulama doğruluğunun ise 0,77 seviyesinde olduğu görülebilir. Eğitim ve doğrulama eğrileri arasındaki açıklık oldukça az görünse de, modelin genelleştirme özelliği yeterli değildir. Ancak genel doğruluk orta seviyededir.



Şekil 177. %80-%10-%10 Bölümlendirmesi Karmaşıklık Matrisi

Şekil 177'de ilk modelin karmaşıklık matrisi görülmektedir. Okul sınıfının park sınıfı ile karıştırıldığı 6 örnek ve merkez sınıfının sanayi bölgesi olarak tahmin edildiği 5 örnek bulunmaktadır. İlk modelin sınıflandırma raporu ise ek 14'te verilmiştir.



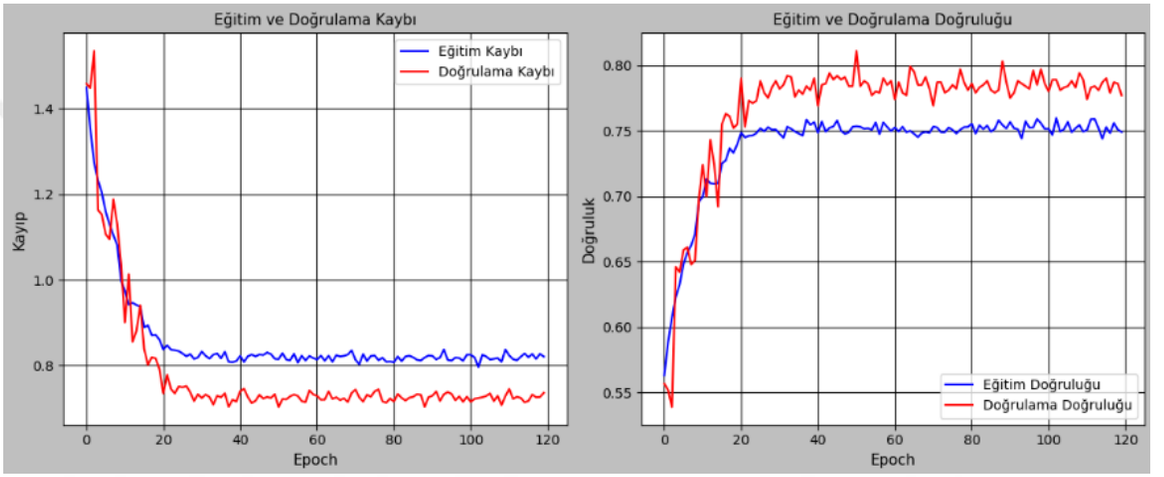
Şekil 178. %80-%10-%10 Bölümlendirmesi AUC-ROC Eğrisi

Şekil 178'de ilk modele ait ROC eğrileri ve AUC skorları gösterilmektedir. Sınıfların bir kısmının (12) AUC skoru 1 iken; 8 sınıfın 0,99, 5 sınıfın 0,98 ve 5 sınıfın 0,97'dir.

Tablo 52. %80-%10-%10 Bölümlendirmesi Değerlendirme Sonuçları

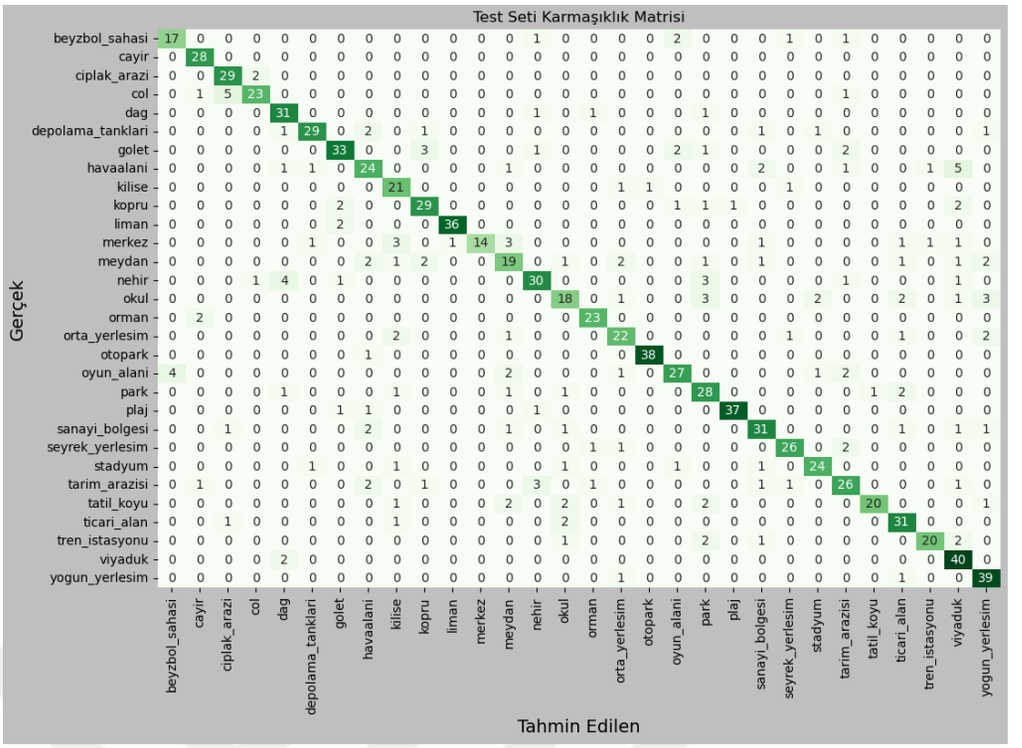
Bölümlendirme (%)	Optimizasyon algoritması	Öğrenme Oranı Başlangıç Değeri	Yığın Boyutu	Görüntü Boyutu	Doğruluk	Kesinlik	Duyarlılık	F1
80-10-10	SGD	0,001	16	224x224	0,8130	0,8209	0,8089	0,8091

Tablo 52'den de görüldüğü üzere veri artırımı yapılan modelin başarımının yaklaşık %2 arttığı görülmektedir. Doğruluk skoru 0,8130'a, kesinliği 0,8209'a, duyarlılığı 0,8089'a ve F1 skoru da 0,8091'e yükselmiştir.



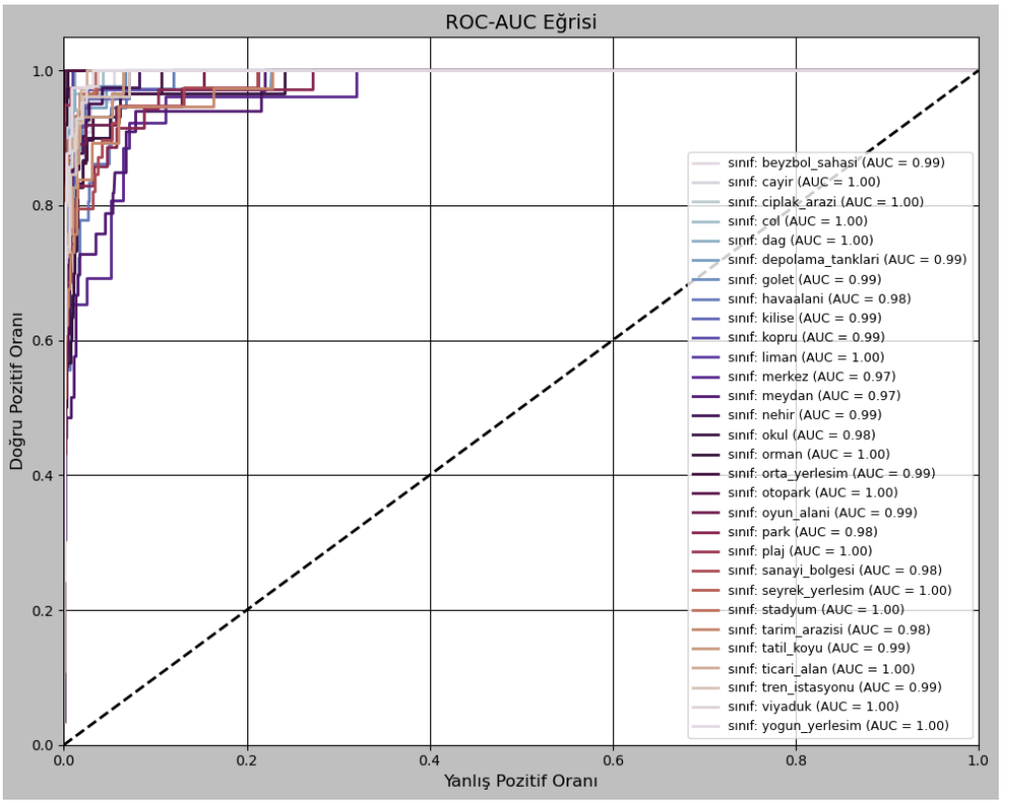
Şekil 179. %80-%10-%10 Veri Artırımı Yapılmış Model Kayıp ve Doğruluk Grafiği

Şekil 179'da ise ikinci modele ait kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde eğitim (0,81) ve doğrulama kaybının (0,70) 20. epochtan itibaren sabit kalmaya başladığı görülmektedir. Açıklık farkı ise 0,11 gibi bir değer almıştır. Sağ tarafta bulunan doğruluk grafiğinde ise 120 epoch sonunda eğitim doğruluğu (0,75) ile doğrulama doğruluğu (0,78) arasındaki açıklığın 0,03 olduğu görülmektedir. İlk modele göre az da olsa bir iyileşme olduğu söylenebilir. Ancak, genel performans olarak RMSProp ile son iki sırayı paylaşmaktadır.



Şekil 180. %80-%10-%10 Veri Artırımı Yapılmış Model Karmaşıklık Matrisi

Şekil 180'de ikinci modelin karmaşıklık matrisi görülmektedir. Havaalanı sınıfının viyadük sınıfı ile karıştırıldığı 5 örnek ve nehir sınıfının dağ olarak tahmin edildiği 4 örnek bulunmaktadır. İkinci modelin sınıflandırma raporu ise ek 15'te verilmiştir.



Şekil 181. %80-%10-%10 Veri Artırımı Yapılmış Model AUC-ROC Eğrisi

Şekil 181'de ikinci modele ait ROC eğrileri ve AUC skorları gösterilmektedir. Sınıfların bir kısmının (13) AUC skoru 1 iken; 10 sınıfın 0,99, 5 sınıfın 0,98 ve 2 sınıfın 0,97'dir.

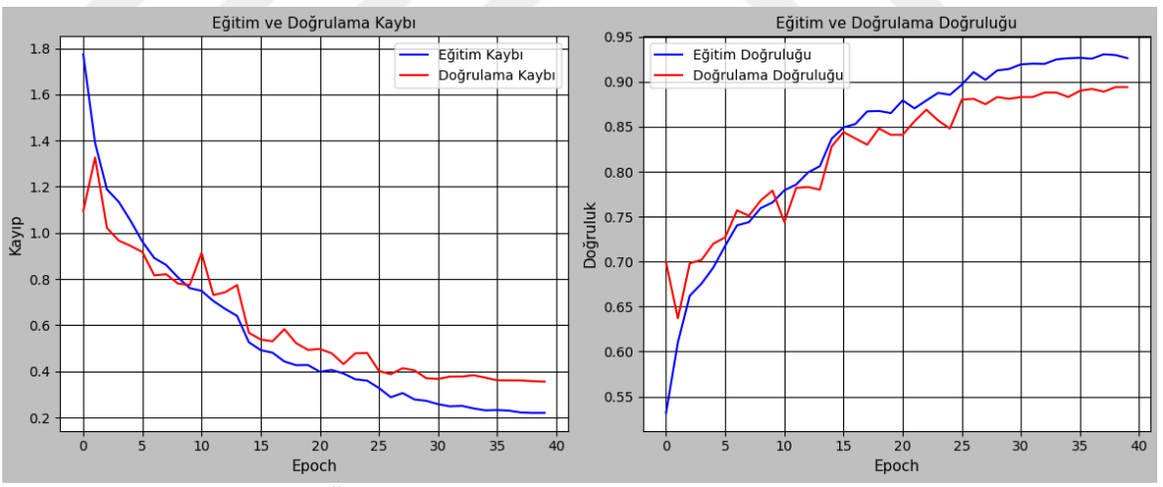
5.1.4. Öğrenme Oranı Bazlı Karşılaştırmalar

Bu bölümde ise veri seti 2, öğrenme oranı bazlı karşılaştırmalara tabi tutulmuştur. İki farklı öğrenme oranı (0,0001 ve 0,00001) seçilmiş ve her bir öğrenme oranında en iyi performansı gösteren temel model, veri artırımı ve ağ yapısındaki güncellemeler yoluyla iyileştirilmeye çalışılmıştır. İlk olarak öğrenme oranı 0,0001 seçilerek analizlere başlanmıştır.

Tablo 53. Öğrenme Oranı Bazlı Karşılaştırmalar 1

Öğrenme Oranı Başlangıç Değeri	Yığın Boyutu / Optimizasyon algoritması	Görüntü Boyutu	Bölümlendirme	Doğruluk	Kesinlik	Duyarlılık	F1
0,0001	16	224x224	9000-500-500	0,8920	0,8990	0.8859	0,8893
			8000-1000-1000	**0,8970**	**0,8961**	**0,8915**	**0,8927**
			7000-1500-1500	0,8767	0,8768	0,8703	0,8720
	Adam		6000-2000-2000	0,8450	0,8474	0,8418	0,8427
			5000-2500-2500	0,8384	0,8378	0,8345	0,8336

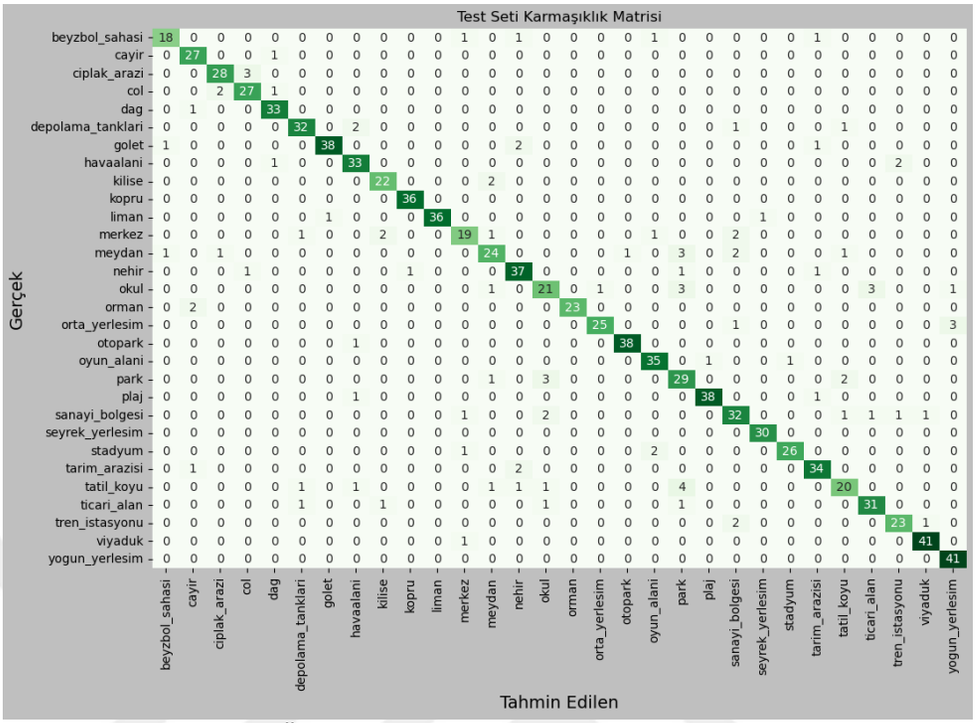
Tablo 53'e göre 0,0001 öğrenme oranında en iyi skoru, %80-%10-%10 bölümlendirmesi vermiştir. Bu modelin doğruluk skoru 0,8970, kesinliği 0,8961, duyarlılığı 0,8915 ve F1 skoru 0,8927'dir.



Şekil 182. Öğrenme Oranı 0,0001 Kayıp ve Doğruluk Grafiği

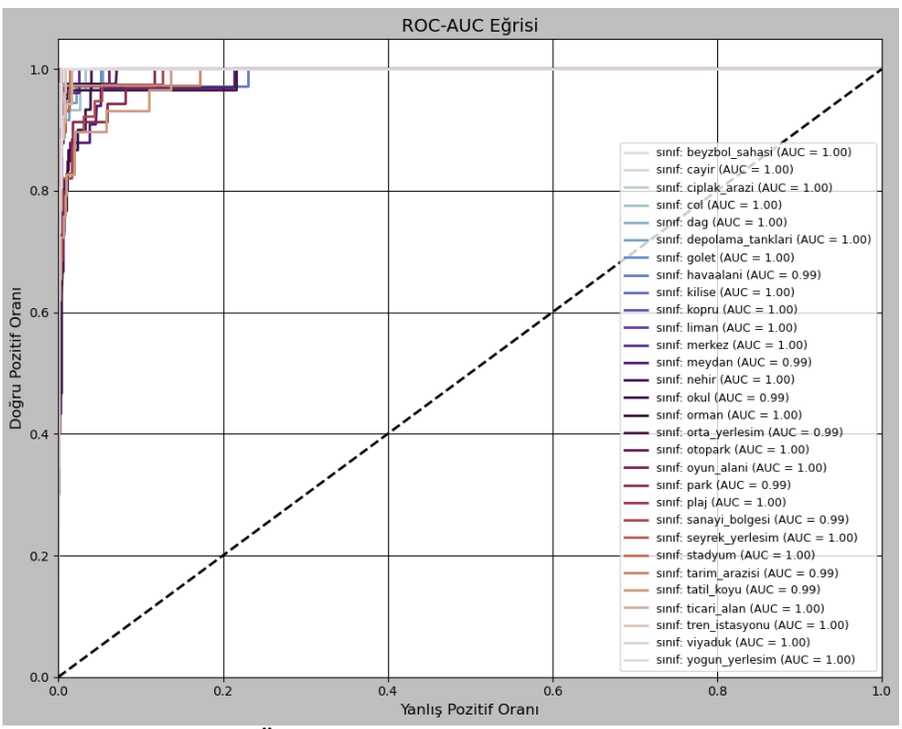
Şekil 182'de öğrenme oranı 0,0001 olan modelin kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde, eğitim kaybının 40 epoch sonunda 0,2 seviyesine, doğrulama kaybının ise 0,4 seviyesine indiği görülebilir. İki eğrinin açıklık farkı ise yaklaşık 0,2'dir. Her iki eğrinin birbirine paralel düşüşü, aşırı öğrenmenin olmadığını göstermektedir. Sağ taraftaki doğruluk grafiğinde ise 40. epoch itibariyle eğitim doğruluğunun 0,93, doğrulama doğruluğunun ise 0,90 seviyesinde olduğu görülmektedir. Eğitim ve doğrulama doğruluğu

arasındaki açıklık farkının oldukça küçük (0,03) olması, genelleme yeteneğinin iyi olduğu ve aşırı öğrenmenin neredeyse olmadığını göstermektedir.



Şekil 183. Öğrenme Oranı 0,0001 Karmaşıklık Matrisi

Şekil 183'te ilk modelin karmaşıklık matrisi görülmektedir. Tatil köyü sınıfının park sınıfı ile karıştırıldığı 4 örnek ve park sınıfının okul sınıfı olarak tahmin edildiği 3 örnek bulunmaktadır. İlk modelin sınıflandırma raporu ise ek 16'da verilmiştir.



Şekil 184. Öğrenme Oranı 0,0001 AUC-ROC Eğrisi

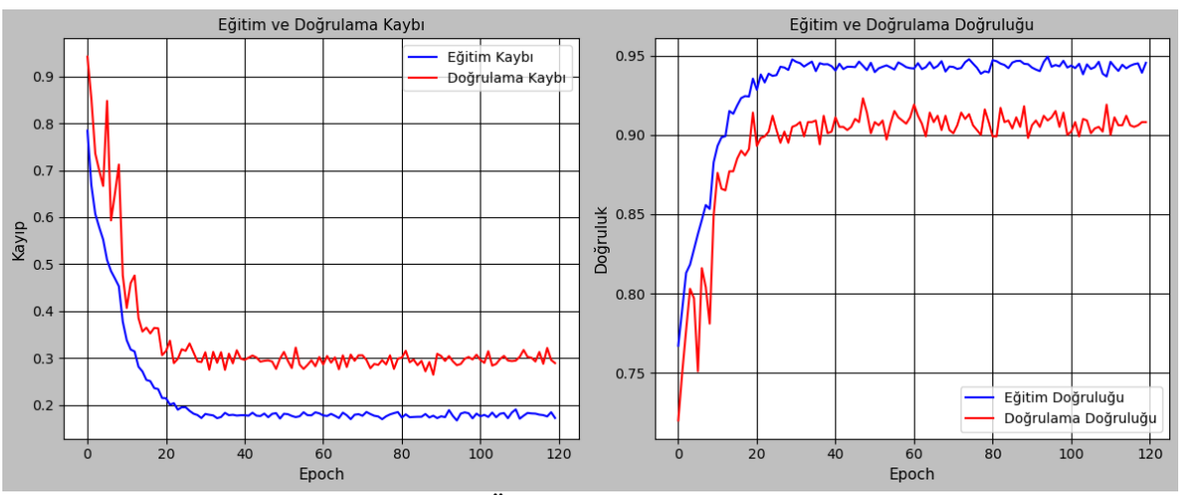
Şekil 184'te ilk modele ait ROC eğrileri ve AUC skorları gösterilmektedir. Sınıfların büyük çoğunluğunun (22) AUC skoru 1'dir. Diğer 8 sınıf ise 0,99 olarak görülmektedir.

Model iyileştirmeleri kapsamında, ikinci modele GlobalAveragePooling2D eklenmiş ve düzleştirme katmanı çıkarılmıştır. Daha önce de belirtildiği üzere düzleştirme katmanı, 2 boyutlu özellik haritalarını genellikle yoğun katmanlara beslemeden önce 1 boyutlu vektöre dönüştürmede kullanılmaktadır (Suriya, Chandran ve Sumithra, 2019). GlobalAveragePooling2D (GAP) tekniği ise tüm değerleri düzleştirmek yerine, her bir özellik haritasının ortalamasını hesaplayarak parametre sayısını azaltır ve aşırı öğrenme potansiyelini düşürür. Yani, GAP tekniğinden sonra düzleştirme katmanı eklemeye gerek kalmamaktadır. Analizler, 120 epoch değeri ile bu şekilde yeniden yapılmıştır.

Tablo 54. Veri Artırımı Yapılmış Öğrenme Oranı 0,0001 Değerlendirme Sonuçları

Öğrenme Oranı Başlangıç Değeri	Yığın Boyutu / Optimizasyon algoritması	Görüntü Boyutu	Bölümlendirme	Doğruluk	Kesinlik	Duyarlılık	F1
0,0001	16 / Adam	224x224	8000-1000-1000	0,9140	0,9122	0,9089	0,9094

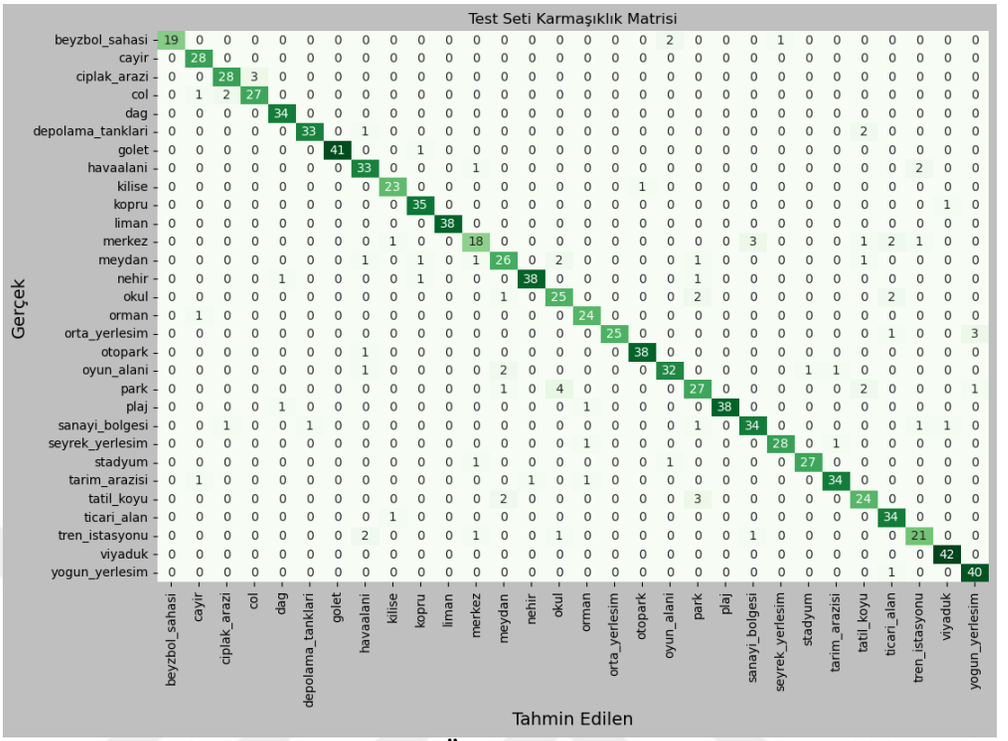
Tablo 54'ten de görüldüğü üzere ağın yapısındaki değişiklikler, %1,70 oranında iyileştirme sağlamıştır. Doğruluk skorunun 0,9140'a, kesinliğin 0,9122'ye, duyarlılığın 0,9089'a ve F1'in 0,9094'e yükseldiği görülebilir.



Şekil 185. Veri Artırımı Yapılmış Öğrenme Oranı 0,0001 Kayıp ve Doğruluk Grafiği

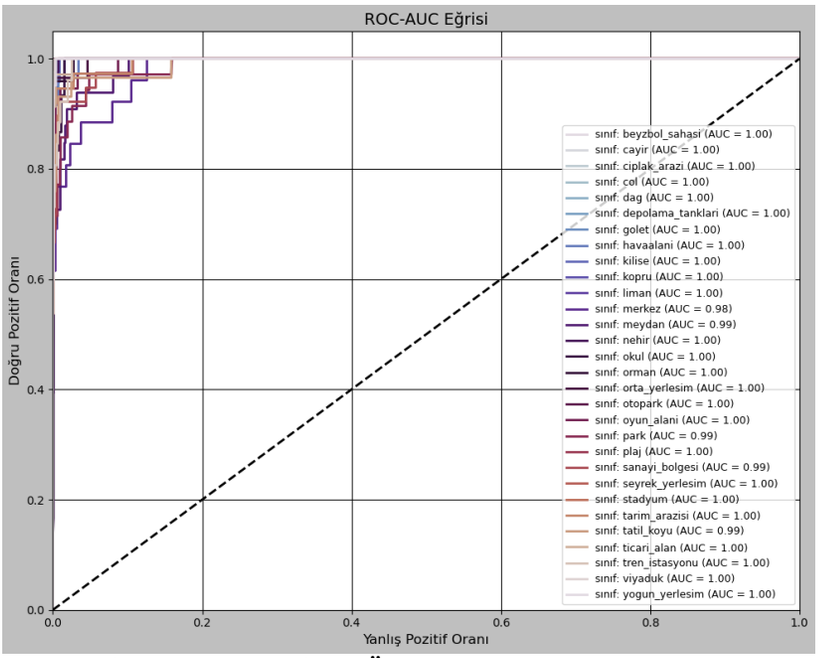
Şekil 185'te ikinci modelin kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde, eğitim kaybının 20. epochtan itibaren 0,20'nin altına düştüğü görülmektedir. Doğrulama kaybı ise aynı epochtan itibaren 0,30 seviyesinde sabitlenmiştir. Eğitim ve doğrulama kaybı arasındaki 0,10 puanlık açıklık bulunmaktadır. Sağ taraftaki doğruluk grafiğinde ise eğitim doğruluğu 0,94 seviyesinde sabitken, doğrulama doğruluğu da 0,91

seviyesinde sabitlenmiştir. Grafikteki 0,30 puanlık açıklık da aşırı öğrenme durumunu yansıtmamaktadır.



Şekil 186. Veri Artırımı Yapılmış Öğrenme Oranı 0,0001 Karmaşıklık Matrisi

Şekil 186'da ikinci modelin karmaşıklık matrisi görülmektedir. Çıplak arazi sınıfının çöl sınıfı ile karıştırıldığı 3 örnek ve orta yerleşim sınıfının yoğun yerleşim sınıfı olarak tahmin edildiği 3 örnek bulunmaktadır. İkinci modelin sınıflandırma raporu ise ek 17'de verilmiştir.



Şekil 187. Veri Artırımı Yapılmış Öğrenme Oranı 0,0001 AUC-ROC Eğrisi

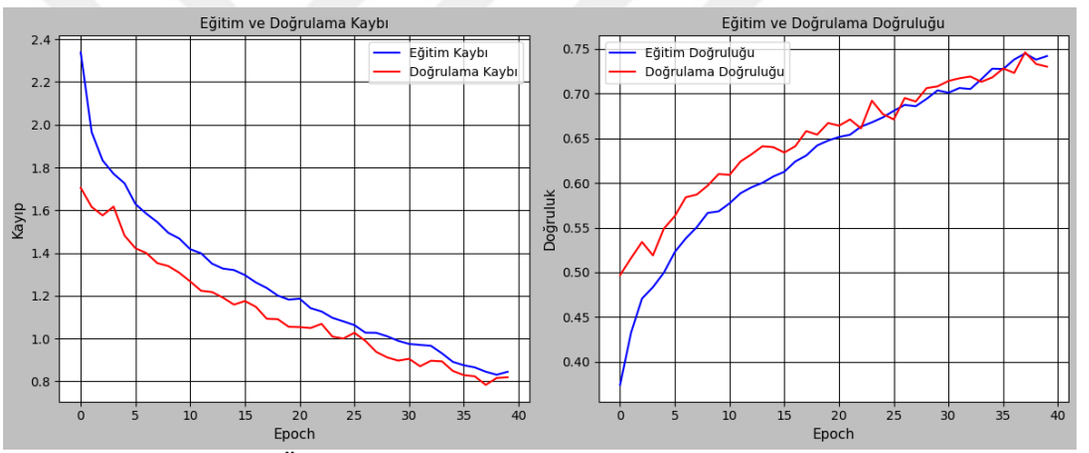
Şekil 187'de ikinci modele ait tüm sınıflar için ROC eğrileri ve AUC skorları gösterilmektedir. Sınıfların büyük çoğunluğunun (24) AUC skoru 1'dir. Diğer 6 sınıf ise 0,98-0,99 aralığındadır.

Son olarak 0,00001 öğrenme oranı ile analizler yapılmıştır.

Tablo 55. Öğrenme Oranı Bazlı Karşılaştırmalar 2

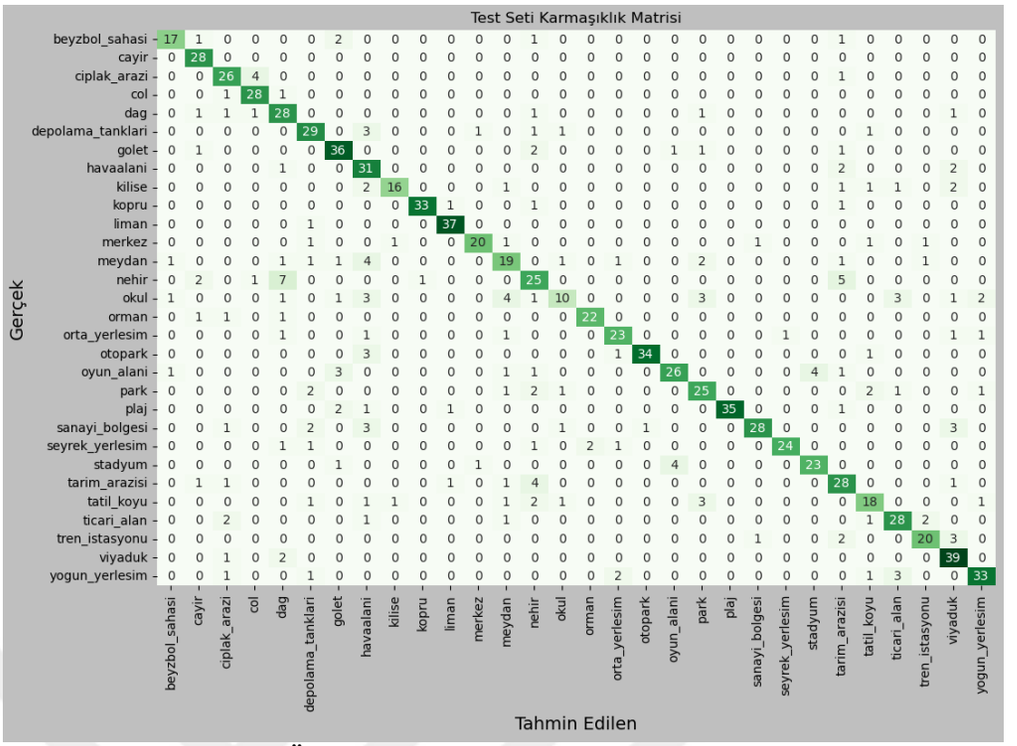
Öğrenme Oranı Başlangıç Değeri	Yığın Boyutu / Optimizasyon algoritması	Görüntü Boyutu	Bölümlendirme	Doğruluk	Kesinlik	Duyarlılık	F1
0,00001	16	224x224	9000-500-500	0,7460	0,7554	0,7419	0,7400
			8000-1000-1000	**0,7890**	**0,8023**	**0,7855**	**0,7873**
			7000-1500-1500	0,7200	0,7359	0,7139	0,7132
	Adam		6000-2000-2000	0,6780	0,6951	0,6725	0,6691
			5000-2500-2500	0,6872	0,7030	0,6859	0,6822

Tablo 55'ten de görüldüğü üzere, %80-%10-%10 bölümlendirmesinin daha başarılı sonuç verdiği görülmektedir. Modelin doğruluk skoru 0,7890, kesinlik skoru 0,8023, duyarlılık skoru 0,7855 ve F1 skoru 0,7873 olarak görülmektedir.



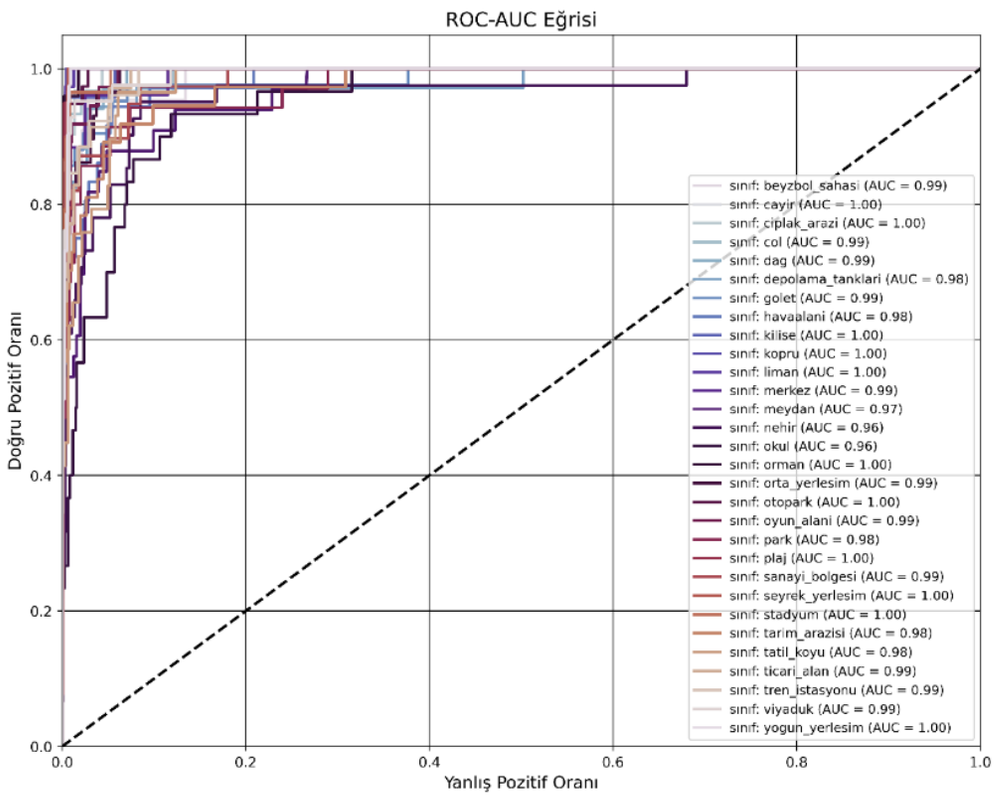
Şekil 188. Öğrenme Oranı 0,00001 Kayıp ve Doğruluk Grafiği

Şekil 188'de ilk modelin kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp eğrisinde, eğitim kaybının 40 epoch sonunda 0,85 seviyesine indiği görülmektedir. Doğrulama kaybının ise hemen hemen benzer seviyede durduğu görülebilir. Eğriler arasındaki açıklık neredeyse yok görünmekte ancak kayıplar yüksek seviyededir denilebilir. Sağ tarafta bulunan doğruluk grafiğinde ise eğitim doğruluğunun 0,74'e ulaştığı ve doğrulama doğruluğunun ise 0,73 seviyesine geldiği görülebilir. Ancak, görüldüğü üzere aşırı öğrenme durumu olmasa da modelin performansı yetersiz görünmektedir. Bu duruma yol açan temel etken ise optimizasyon algoritmasının model ağırlıklarında yeterince büyük güncellemeler yapmasını engelleyen son derece düşük öğrenme oranı (0,00001) olabilir. Başka bir deyişle, seçilen öğrenme oranı, yakınsamayı önemli ölçüde yavaşlatmıştır.



Şekil 189. Öğrenme Oranı 0.00001 Karmaşıklık Matrisi

Şekil 189'da ise ilk modelin karmaşıklık matrisi görülmektedir. Nehir sınıfının dağ sınıfı ile karıştırıldığı 7 örnek ve tarım arazisi sınıfının nehir olarak tahmin edildiği 4 örnek bulunmaktadır. İlk modelin sınıflandırma raporu ise ek 18'de verilmiştir.



Şekil 190. Öğrenme Oranı 0,00001 AUC-ROC Eğrisi

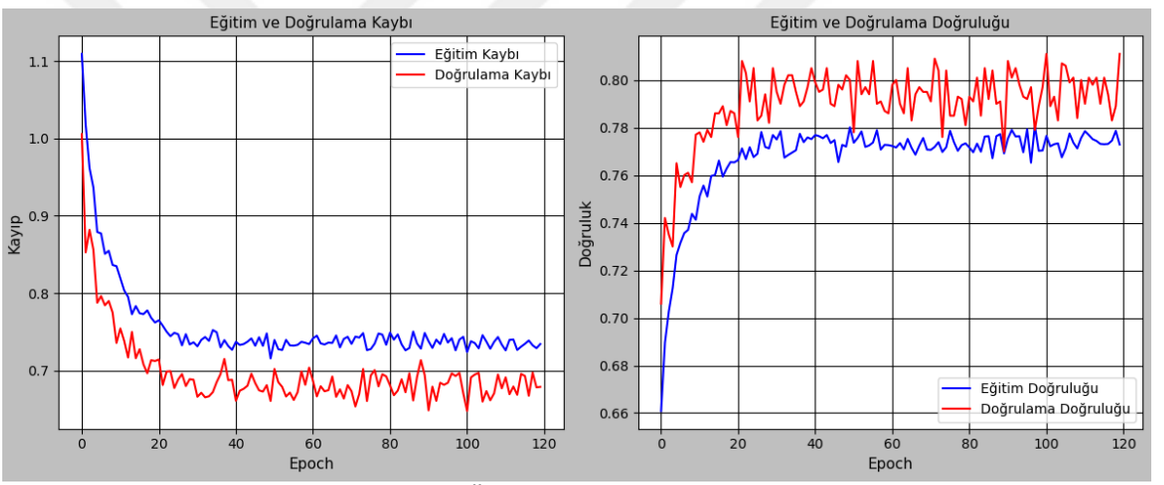
Şekil 190'da ilk modele ait sınıfların ROC eğrileri ile AUC skorları gösterilmektedir. Sınıflardan 10 adedinin AUC skoru 1'dir ve sınıf arasında mükemmel ayrılabilirliği göstermektedir. Çoğu sınıf ise 0,96-0,99 aralığında sıralanmaktadır.

Bu modelde de düzleştirme katmanı çıkarılıp yerine GlobalAveragePooling2D eklenmiştir.

Tablo 56. Veri Artırımı Yapılmış Öğrenme Oranı 0,00001 Değerlendirme Sonuçları

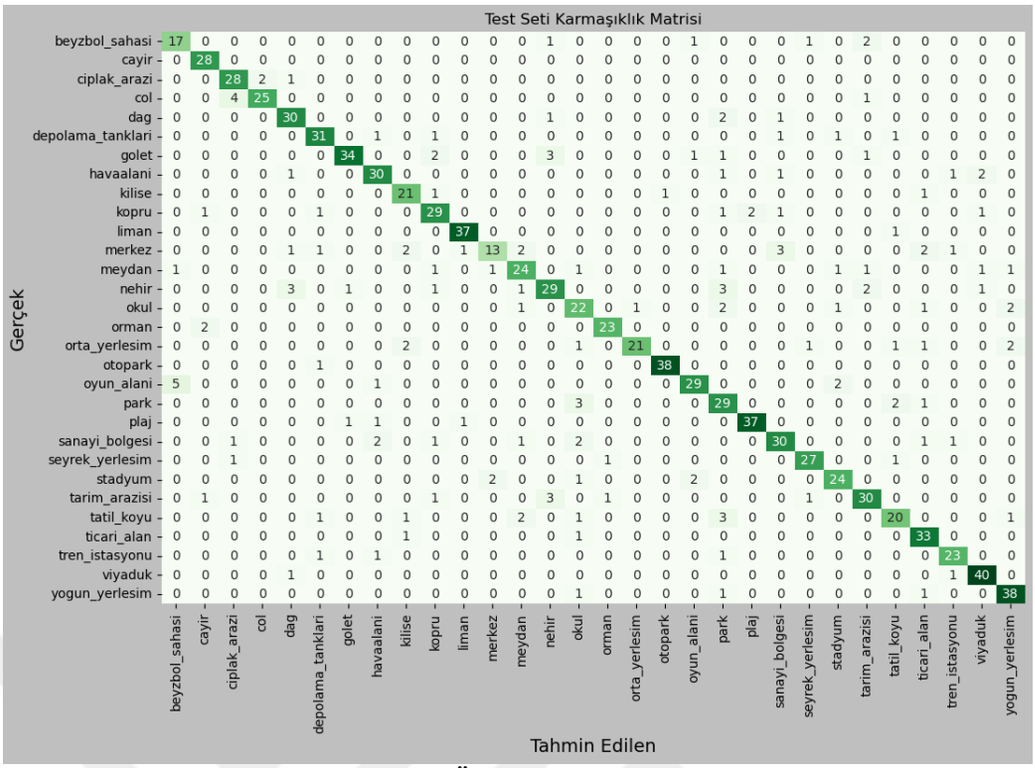
Öğrenme Oranı Başlangıç Değeri	Yığın Boyutu / Optimizasyon algoritması	Görüntü Boyutu	Bölümlendirme	Doğruluk	Kesinlik	Duyarlılık	F1
0,00001	16 / Adam	224x224	8000-1000-1000	0,8400	0,8403	0,8359	0,8347

Tablo 56'dan da görüldüğü üzere, ağın yapısında yapılan değişiklikler, doğruluk skorunun 0,8400'a kesinliğin 0,8483'e, duyarlılığın 0,8359'a ve F1'in 0,8347'ye yükselmesini sağlamıştır. Başka bir deyişle, toplamda %5 oranında iyileştirme sağlandığı söylenebilir.



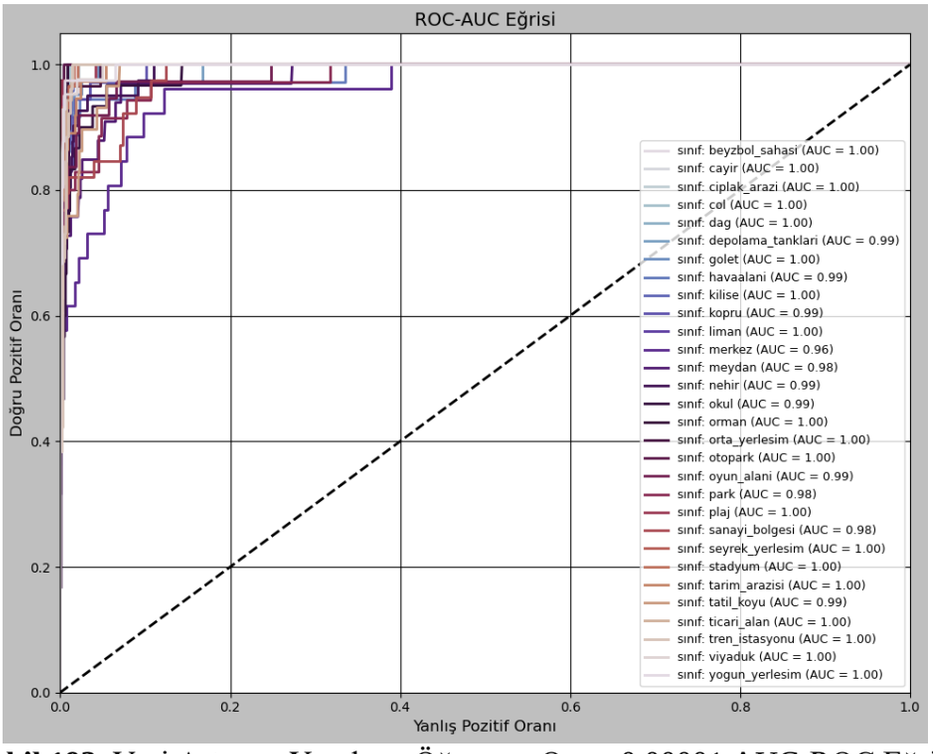
Şekil 191. Veri Artırımı Yapılmış Öğrenme Oranı 0,00001 Kayıp ve Doğruluk Grafiği

Şekil 191'de ikinci modelin kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde eğitim kaybının, zorlanarak da olsa 0,8 altına indiği görülebilir. Doğrulama kaybı ise 0,7'nin altına inmiştir. Kısa aralıklarla görülen bu zorlanma, modelin istikrarsızlığını değil, öğrenme oranından kaynaklanan yakınsama çabalarını göstermektedir. Sağ tarafta bulunan doğruluk grafiğinde ise eğitim doğruluğunun 120 epoch sonunda 0,78 seviyesine çıktığı görülebilir. Doğrulama doğruluğunun ise 0,82'de sonlandığı görülmektedir. İlk modelle karşılaştırıldığında daha yüksek doğrulama doğruluğu elde edilmiştir.



Şekil 192. Veri Artırımı Yapılmış Öğrenme Oranı 0,00001 Karmaşıklık Matrisi

Şekil 192'de ise ikinci modelin karmaşıklık matrisi görülmektedir. Çöl sınıfının çıplak arazi sınıfı ile karıştırıldığı 4 örnek ve oyun alanı sınıfının beyzbol sahası olarak tahmin edildiği 5 örnek bulunmaktadır. Ayrıca, ilk modele göre yanlış sınıflandırmaların azaldığı görülebilir. İkinci modelin sınıflandırma raporu ise ek 19'da verilmiştir.



Şekil 193. Veri Artırımı Yapılmış Öğrenme Oranı 0,00001 AUC-ROC Eğrisi

Şekil 193'te ikinci modele ait tüm sınıflar için ROC eğrileri ve AUC skorları gösterilmektedir. Sınıfların büyük çoğunluğunun AUC skoru 1'dir ve pozitif sınıf ile diğer tüm sınıflar arasında mükemmel ayrılabilirliği göstermektedir. Diğer sınıfları ise 0,96-0,99 arasındadır.

5.2. Transfer Öğrenmesi

Bu bölümde ise veri seti 2, EfficientNet modelinin B0-B3 ve v2B0-v2B3 modelleri ile çalıştırılmış ve sonuçlar her dört ön eğitilmiş model için karşılaştırılmıştır. Her model öncelikle temel model olarak baştan ve ağırlıksız eğitilmiş, ardından optimizasyon ve düzenlileştirme teknikleri uygulanarak tekrar eğitim sürecine dahil edilmiştir. Temel modeller de kendi içerisinde 5 ayrı alt modele ayrılmış ve bu modellerde diğer tüm parametreler sabitken, eğitim-doğrulama-test setlerinin bölümlendirilmesinde %90-%5-%5, %80-%10-%10, %70-%15-%15, %60-%20-%20 ve %50-%25-%25 yaklaşımları kullanılmıştır.

Görüntü boyutları, her bir EfficientNet modelinde belirtilen şekliyle modele verilmiştir. Yapılan denemelerde, görüntü boyutunun EfficientNet modelleri geliştikçe artması, ekran kartı hafıza problemlerine yol açmış ve yığın boyutu düzenlemesi EfficientNetB4 modelinden sonra işe yaramamıştır. Görüntü boyutu arttıkça, yığın boyutu hiperparametresi 64'ten 32'ye ve daha sonra 16'dan 8'e düşürülmüş fakat bu çözüm de işe yaramamıştır. Dolaysıyla, görüntüler sabit bir boyuta (224x224) ayarlanarak farklı modeller için eşit bir şekilde karşılaştırma olanağı sağlanmıştır.

Analiz yapılan temel modellerde yığın boyutu hiperparametresi 8 olarak seçilmiştir. Ayrıca, temel model eğitimleri kendi arasında 60 epoch değeri ile sabit tutularak, daha anlaşılır ve karşılaştırılabilir yorumların yapılması sağlanmıştır. Modele temel özellik çıkarıcıdan sonra 128 nöron ve ReLU aktivasyonu ile tam bağlantılı katman eklenmiştir.

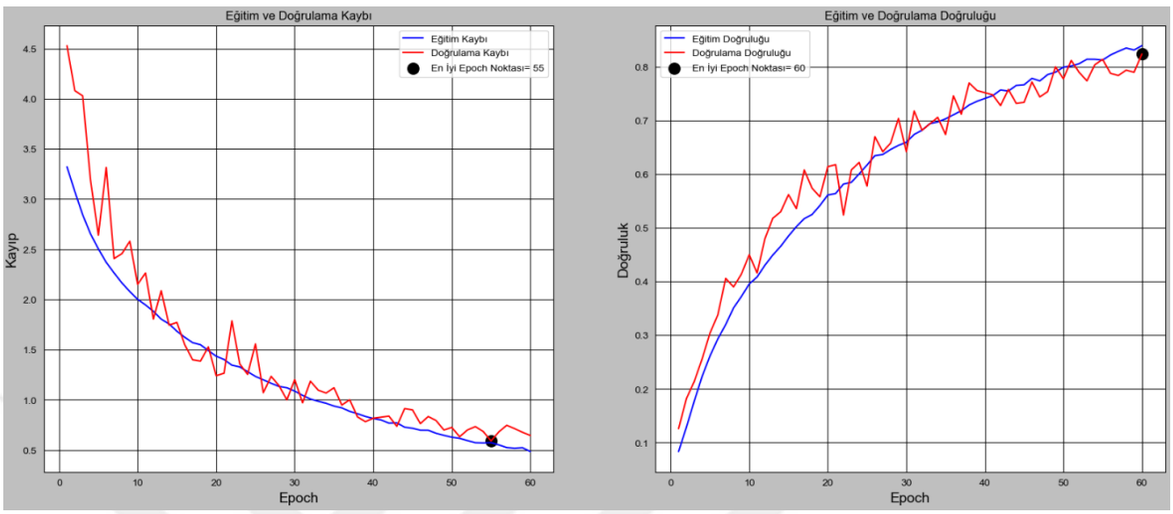
5.2.1. Temel Modeller

İlk transfer öğrenmesi uygulaması, EfficientNetB0 modeli ile gerçekleştirilmiştir. Bu aşamada, ImagetNet ağırlıkları seçilmemiş, eğitim sürecinde ise görüntüler karıştırılarak epochlar başlatılmıştır. Analiz sonuçları aşağıdaki tabloda gösterilmektedir.

Tablo 57. EfficientNetB0 Eğitim Kümesi Bazlı Model Analiz Sonuçları

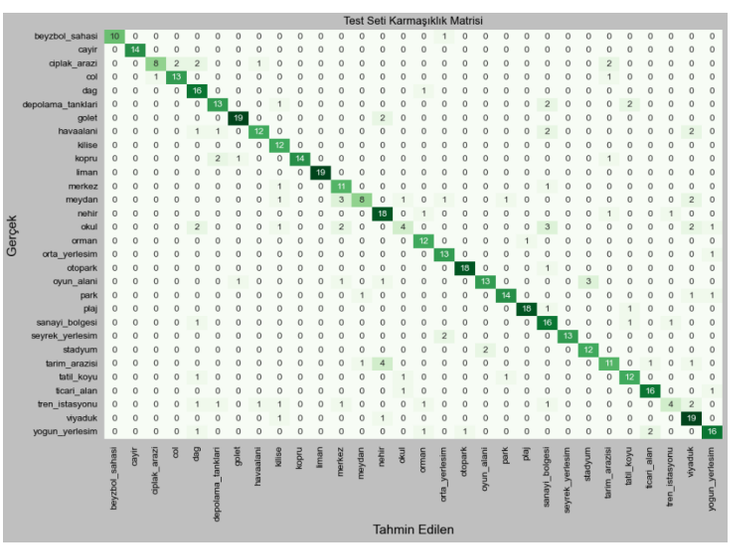
Transfer Öğrenmesi	Model	Epoch	Yığın Boyutu	Eğitim-Doğrulama-Test	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1
EfficientNetB0	**1**	**60**	**8**	**9000-500-500**	**0,7960**	**0,8070**	**0,7936**	**0,7867**
	2	60	8	8000-1000-1000	0,7950	0,8035	0,7978	0,7911
	3	60	8	7000-1500-1500	0,7693	0,7855	0,7699	0,7598
	4	60	8	6000-2000-2000	0,7360	0,7524	0,7364	0,7313
	5	60	8	5000-2500-2500	0,6968	0,7241	0,6968	0,6887

Tablo 57'de, EfficientNetB0 ön eğitimli modelin değerlendirme sonuçları gösterilmektedir. %90-%5-%5 bölümlendirmesine sahip 1 numaralı model; 0,7960 doğruluk, 0,8070 kesinlik, 0,7936 duyarlılık ve 0,7867 F1 skorları ile diğer bölümlendirmelerden daha iyi performans göstermiştir.



Şekil 194. Temel EfficientNetB0 Modeli Kayıp ve Doğruluk Grafiği

Şekil 194'te temel B0 modelinin kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde, eğitim kaybının 60 epoch sonunda 0,50 seviyesine indiği görülmektedir. Doğrulama kaybı ise 0,70 seviyesine kadar inmiştir. Süreç boyunca doğrulama kaybının sürekli olarak dalgalandığı görülebilir. Sağ taraftaki doğruluk grafiğinde ise 60 epoch sonunda eğitim doğruluğunun 0,85, doğrulama doğruluğunun ise 0,84 seviyesinde birleştiği görülmektedir. Eğriler arasındaki açıklık son derece küçüktür (0,01). Başka bir deyişle, modelin genelleme yeteneği iyi görünmekte ancak ağırlıklar olmadan eğitildiği için yeteri kadar genelleme yapamamakta ve doğruluk skorları yeteri kadar yükselememektedir. Modele ait sınıflandırma raporu, ek 20'de verilmiştir.



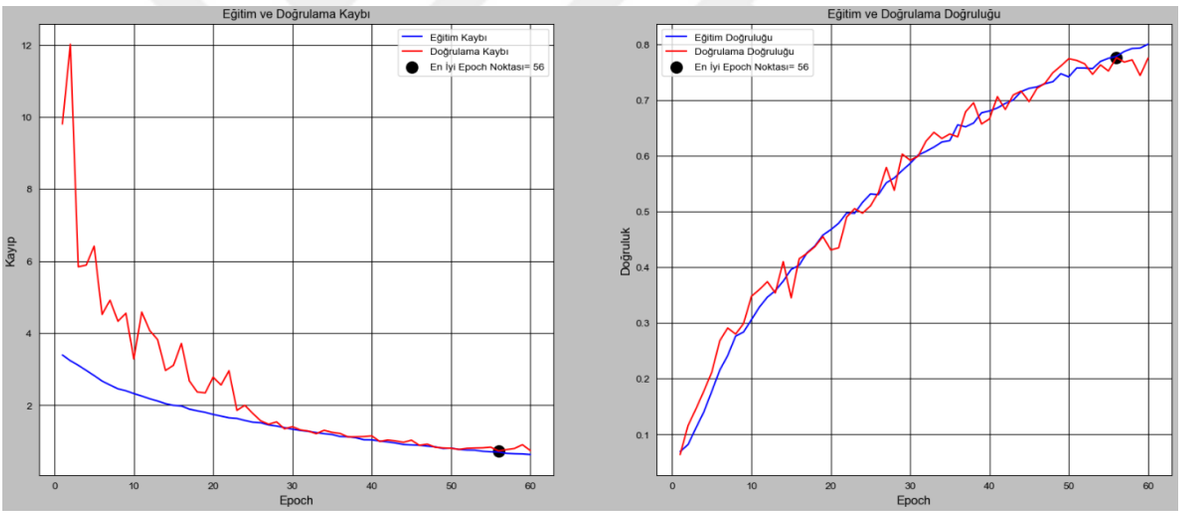
Şekil 195. Temel EfficientNetB0 Modeli Karmaşıklık Matrisi

Şekil 195'te ise modele ait test seti karmaşıklık matrisi gösterilmektedir. 500 test görüntüsünde 102 hatalı sınıflandırma yapılmıştır. Matriste, tarım arazisi sınıfından 4 örneğin nehir; okul sınıfından 3 örneğin sanayi bölgesi; meydan sınıfından 3 örneğin de merkez olarak sınıflandırıldığı göze çarpmaktadır.

Tablo 58. EfficientNetB1 Eğitim Kümesi Bazlı Model Analiz Sonuçları

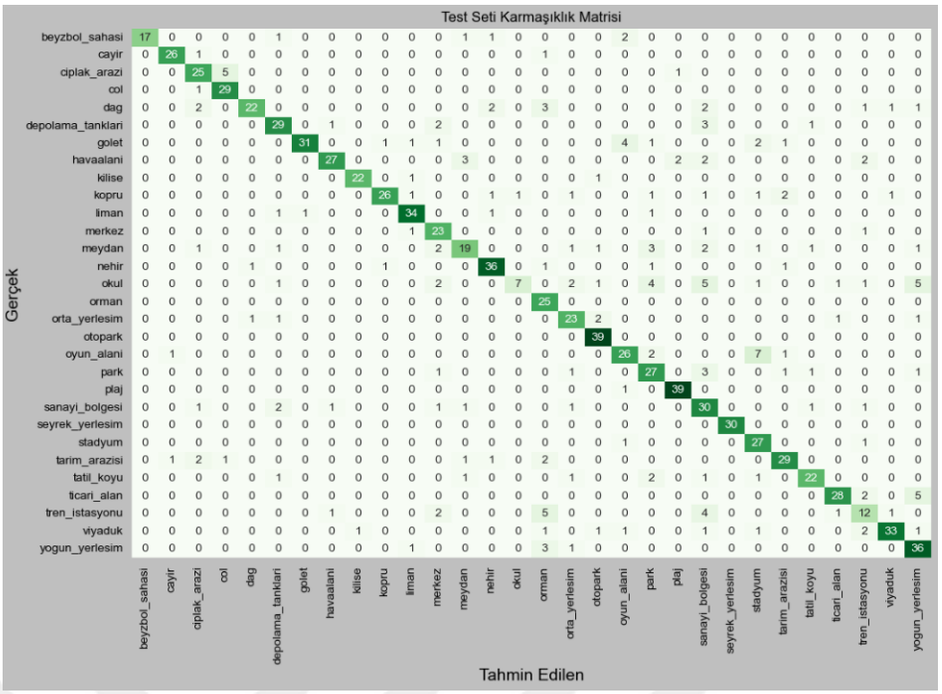
Transfer Öğrenmesi	Model	Epoch	Yığın Boyutu	Eğitim-Doğrulama-Test	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1
EfficientNetB1	1	60	8	9000-500-500	0,7820	0,7917	0,7762	0,7714
	2	**60**	**8**	**8000-1000-1000**	**0,7990**	**0,8146**	**0,7977**	**0,7924**
	3	60	8	7000-1500-1500	0,7473	0.7640	0,7505	0,7444
	4	60	8	6000-2000-2000	0,7120	0,7267	0,7122	0,7023
	5	60	8	5000-2500-2500	0,7036	0,7215	0,7031	0,6945

Tablo 58'de, EfficientNetB1 ön eğitimli modelin değerlendirme sonuçları gösterilmektedir. %80-%10-%10 bölümlendirmesine sahip 2 numaralı model; 0,7990 doğruluk, 0,8146 kesinlik, 0,7977 duyarlılık ve 0,7924 F1 skorları ile diğer bölümlendirmelerden daha iyi performans göstermiştir. Modele ait kayıp-doğruluk grafiği ile karmaşıklık matrisi sonuçları aşağıda gösterilmiştir.



Şekil 196. Temel EfficientNetB1 Modeli Kayıp ve Doğruluk Grafiği

Şekil 196'da temel B1 modelinin kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde, hem eğitim hem de doğrulama kaybının 60 epoch sonunda 0,70 seviyesine indiği görülmektedir. Her iki eğri arasında da sapma veya açıklık olmadığı için de aşırı öğrenme söz konusu değildir. Sağ taraftaki doğruluk grafiğinde ise 60 epoch sonunda eğitim doğruluğunun 0,80, doğrulama doğruluğunun ise 0,77 seviyesinde birleştiği görülmektedir. Bu grafikte de eğriler arasında açıklık bulunmamaktadır. Modelin genelleme yeteneği, ağırlıklar olmadan eğitildiği için yeterli başarımı gösterememiştir. Modele ait sınıflandırma raporu, ek 21'de verilmiştir.



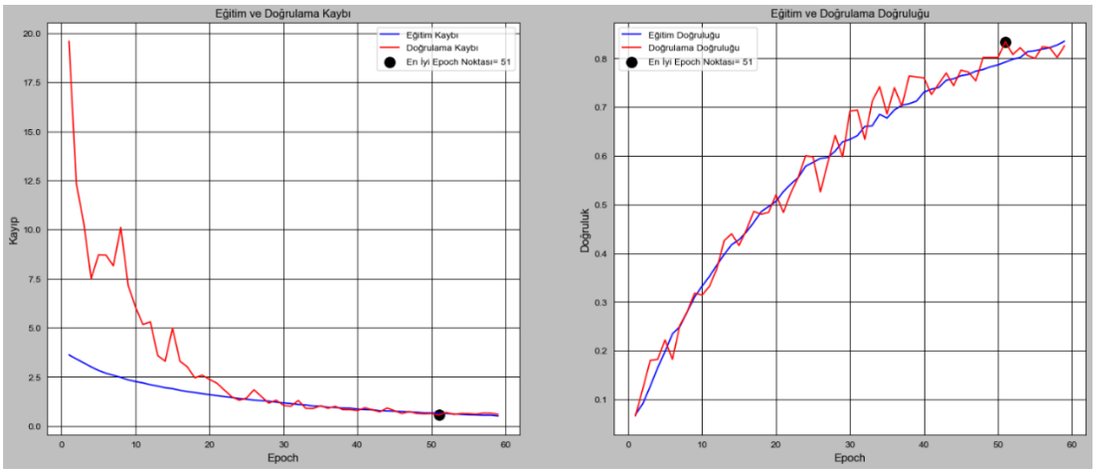
Şekil 197. Temel EfficientNetB1 Modeli Karmaşıklık Matrisi

Şekil 197'de ise modele ait test seti karmaşıklık matrisi gösterilmektedir. 1000 test görüntüsünde 201 hatalı sınıflandırma yapılmıştır. Matriste, oyun alanı sınıfından 7 örneğin stadyum; çıplak arazi sınıfından 5 örneğin çöl; okul sınıfından 5 örneğin sanayi bölgesi; ticari alan sınıfından 5 örneğin yoğun yerleşim olarak sınıflandırıldığı göze çarpmaktadır.

Tablo 59. EfficientNetB2 Eğitim Kümesi Bazlı Model Analiz Sonuçları

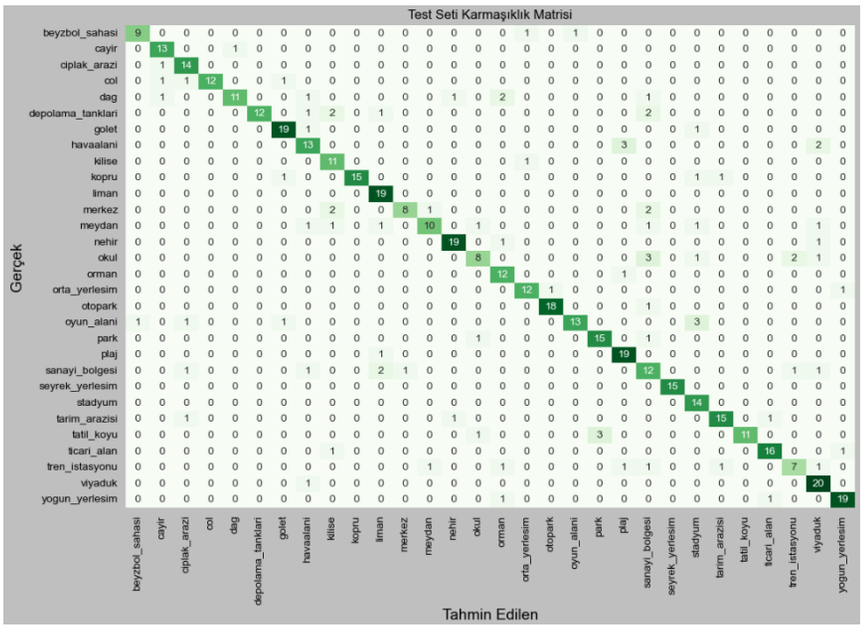
Transfer Öğrenmesi Modeli	Model Adı	Epoch	Yığın Boyutu	Eğitim-Doğrulama-Test	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1
EfficientNetB2	**1**	**60**	**8**	**9000-500-500**	**0,8220**	**0,8365**	**0,8180**	**0,8168**
	2	60	8	8000-1000-1000	0,7990	0,8166	0,8008	0,7984
	3	60	8	7000-1500-1500	0,7353	0,7679	0,7352	0,7297
	4	60	8	6000-2000-2000	0,7320	0,7523	0,7304	0,7292
	5	60	8	5000-2500-2500	0,6404	0,6745	0,6422	0,6339

Tablo 59'da, temel B2 ön eğitimli modelin değerlendirme sonuçları gösterilmektedir. %90-%5-%5 bölümlendirmesine sahip 1 numaralı model; 0,8220 doğruluk, 0,8365 kesinlik, 0,8180 duyarlılık ve 0,8168 F1 skorları ile diğer bölümlendirmelerden daha iyi performans göstermiştir. Modele ait kayıp-doğruluk grafiği ile karmaşıklık matrisi sonuçları aşağıda gösterilmiştir.



Şekil 198. Temel EfficientNetB2 Modeli Kayıp ve Doğruluk Grafiği

Şekil 198'de temel B2 modelinin kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde, hem eğitim hem de doğrulama kaybının 60 epoch sonunda 1,00 seviyesine indiği görülmektedir. İki eğri arasında 28. epochtan itibaren açıklık yoktur ve hemen hemen eşit seviyede sonlanmıştır. Doğrulama kaybının başlarda çok, 30. epochtan sonra ise oldukça az düzeyde olması, öğrenmenin iyi sayılabilecek seviyede olduğunu göstermektedir. Her iki eğri arasında da sapma veya açıklık olmadığı için de aşırı öğrenme söz konusu değildir. Sağ taraftaki doğruluk grafiğinde ise 60 epoch sonunda hem eğitim hem de doğrulama doğruluğunun 0,84 seviyesinde birleştiği görülmektedir. Bu grafikte de eğriler arasında açıklık bulunmamaktadır. Başka bir deyişle, modelin genelleme yeteneği iyi görünmekte ancak ağırlıklar olmadan eğitildiği için başarımı istenilen seviyede yapamamaktadır. Ayrıca, doğrulama doğrulunun sürekli dalgalanması, öğrenme oranı ayarlayıcısının modeli belirli bir seviyede tuttuğunu göstermektedir. Modele ait sınıflandırma raporu, ek 22'de verilmiştir.



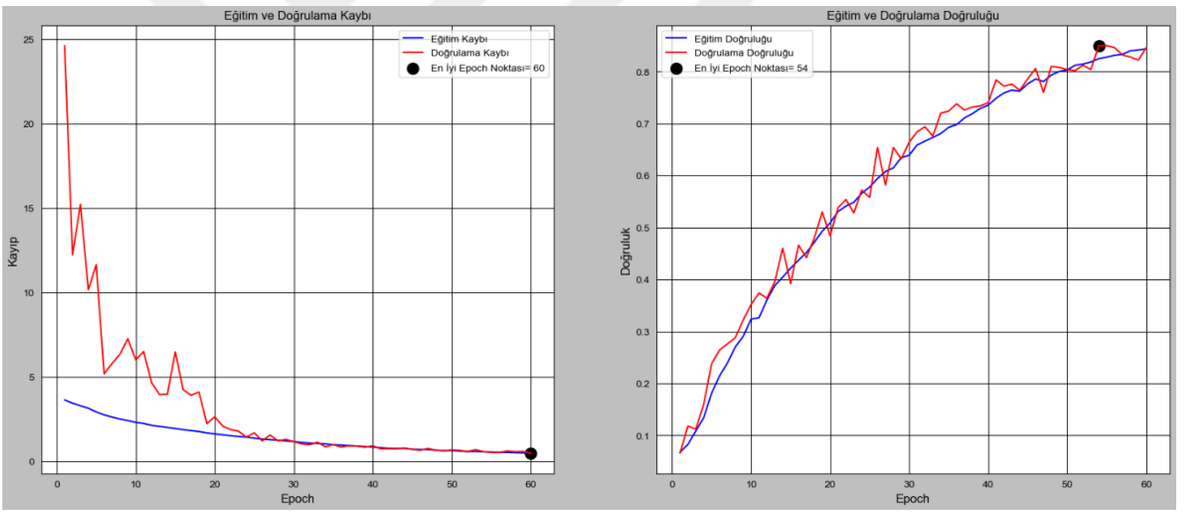
Şekil 199. Temel EfficientNetB2 Modeli Karmaşıklık Matrisi

Şekil 199'da ise modele ait test seti karmaşıklık matrisi gösterilmektedir. 500 test görüntüsünde 89 hatalı sınıflandırma yapılmıştır. Matriste, havaalanı sınıfından 3 örneğin plaj; okul sınıfından 3 örneğin sanayi bölgesi; tatil köyü sınıfından da 3 örneğin park olarak sınıflandırıldığı göze çarpmaktadır.

Tablo 60. EfficientNetB3 Eğitim Kümesi Bazlı Model Analiz Sonuçları

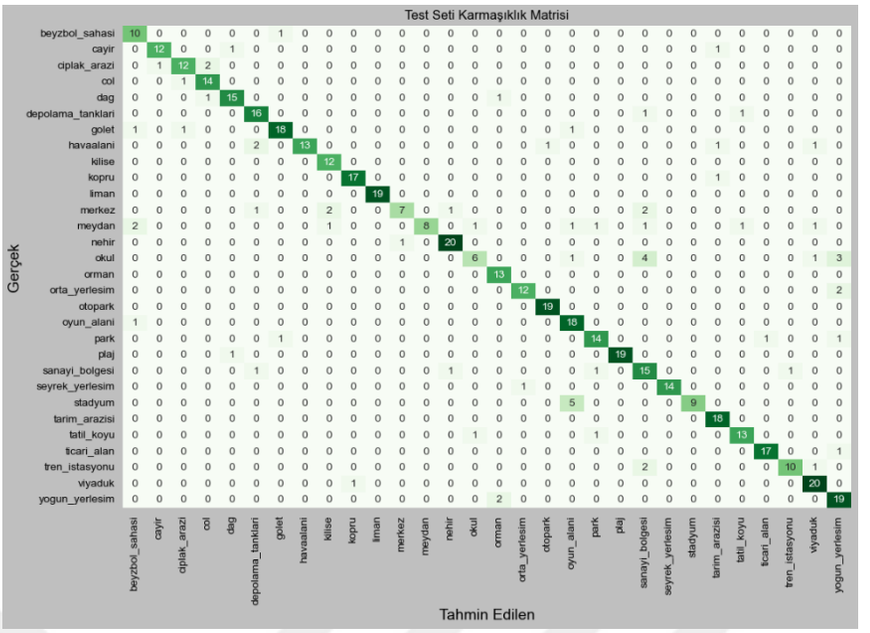
Transfer Öğrenmesi	Model	Epoch	Yığın Boyutu	Eğitim-Doğrulama-Test	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1
EfficientNetB3	**1**	**60**	**8**	**9000-500-500**	**0,8580**	**0,8707**	**0,8512**	**0,8492**
	2	60	8	8000-1000-1000	0,8290	0,8362	0,8296	0,8257
	3	60	8	7000-1500-1500	0,7600	0,7894	0,7592	0,7560
	4	60	8	6000-2000-2000	0,7490	0,7728	0,7460	0,7370
	5	60	8	5000-2500-2500	0,6840	0,7107	0,6842	0,6782

Tablo 60'ta temel B3 ön eğitimli modelin eğitim sonuçları gösterilmektedir. %90-%5-%5 bölümlendirmesine sahip 1 numaralı model; 0,8580 doğruluk, 0,8707 kesinlik, 0,8512 duyarlılık ve 0,8492 F1 skorları ile diğer bölümlendirmelerden daha iyi performans göstermiştir. Modele ait kayıp-doğruluk grafiği ile karmaşıklık matrisi sonuçları aşağıda gösterilmiştir.



Şekil 200. Temel EfficientNetB3 Modeli Kayıp ve Doğruluk Grafiği

Şekil 200'de temel B3 modelinin kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde, hem eğitim hem de doğrulama kaybının 60 epoch sonunda 0,8 seviyesine indiği görülmektedir. İki eğri arasında 25. epochtan itibaren açıklık yoktur ve hemen hemen eşit seviyede sonlanmıştır. Her iki eğri arasında da sapma veya açıklık olmadığı için aşırı öğrenme söz konusu değildir. Sağ taraftaki doğruluk grafiğinde ise 60 epoch sonunda hem eğitim hem de doğrulama doğruluğunun 0,85 seviyesinde birleştiği görülmektedir. Bu grafikte de eğriler arasında açıklık bulunmamaktadır. Başka bir deyişle, diğer önceki modellerdeki gibi genelleme yeteneği iyi sayılmasına rağmen, ağırlıklar olmadan eğitildiği için istenilen seviyede bir başarımı gerçekleştirememiştir. Modele ait sınıflandırma raporu, ek 23'te verilmiştir.



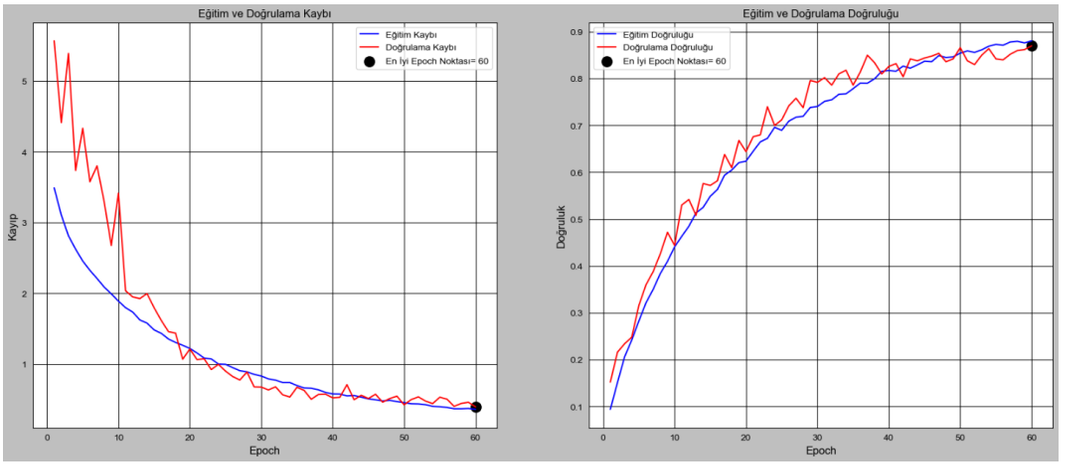
Şekil 201. Temel EfficientNetB3 Modeli Karmaşıklık Matrisi

Şekil 201'de ise modele ait test seti karmaşıklık matrisi gösterilmektedir. 500 test görüntüsünde 71 hatalı sınıflandırma yapılmıştır. Matriste, stadyum sınıfından 5 örneğin oyun alanı; okul sınıfından da 4 örneğin sanayi bölgesi olarak sınıflandırıldığı göze çarpmaktadır.

Tablo 61. EfficientNetV2B0 Eğitim Kümesi Bazlı Model Analiz Sonuçları

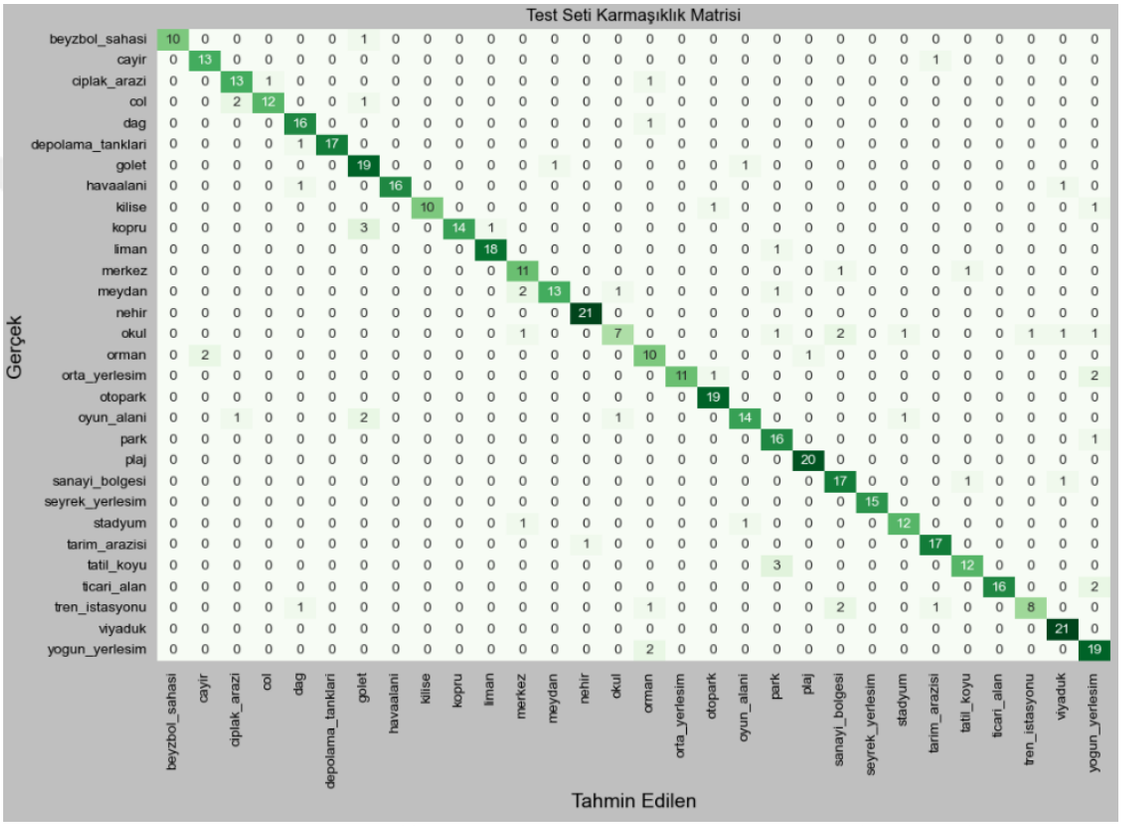
Transfer Öğrenmesi	Model	Epoch	Yığın Boyutu	Eğitim-Doğrulama-Test	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1
EfficientNetV2B0	**1**	**60**	**8**	**9000-500-500**	**0,8740**	**0,8840**	**0,8653**	**0,8685**
	2	60	8	8000-1000-1000	0,8580	0,8654	0,8558	0,8543
	3	60	8	7000-1500-1500	0,8187	0,8288	0,8163	0,8119
	4	60	8	6000-2000-2000	0,7965	0,8217	0,7926	0,7944
	5	60	8	5000-2500-2500	0,7544	0,7694	0,7495	0,7469

Tablo 61'de, temel V2B0 ön eğitimli modelin eğitim sonuçları gösterilmektedir. %90-%5-%5 bölümlendirmesine sahip 1 numaralı model; 0,8740 doğruluk, 0,8840 kesinlik, 0,8653 duyarlılık ve 0,8685 F1 skorları ile diğer bölümlendirmelerden daha iyi performans göstermiştir. EfficientNetB0 modeline göre ise daha yüksek bir doğruluk oranına sahiptir.



Şekil 202. Temel EfficientNetV2B0 Modeli Kayıp ve Doğruluk Grafiği

Şekil 202'de temel V2B0 modelinin kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde, hem eğitim hem de doğrulama kaybının 60 epoch sonunda 0,30 seviyesine indiği görülmektedir. Doğrulama kaybının başlarda çok, son epochlara doğru ise hafif dalgalandığı görülmektedir. Her iki eğri arasında da sapma veya açıklık olmadığı için de aşırı öğrenme söz konusu değildir. Sağ taraftaki doğruluk grafiğinde ise 60 epoch sonunda hem eğitim hem de doğrulama doğruluğunun 0,87 seviyesinde birleştiği görülmektedir. Bu grafikte de eğriler arasında açıklık bulunmamaktadır. Modele ait sınıflandırma raporu, ek 24'te verilmiştir.



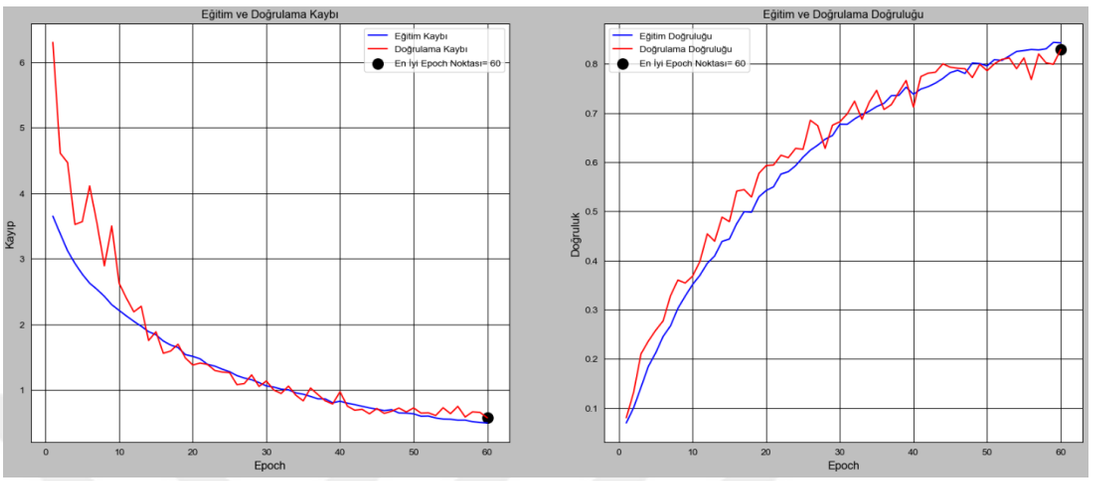
Şekil 203. Temel EfficientNetV2B0 Modeli Karmaşıklık Matrisi

Şekil 203'te ise modele ait test seti karmaşıklık matrisi gösterilmektedir. 500 test görüntüsünde 63 hatalı sınıflandırma yapılmıştır. Matriste, köprü sınıfından 3 örneğin tren istasyonu ve tatil köyü sınıfından da 3 örneğin park olarak sınıflandırıldığı göze çarpmaktadır.

Tablo 62. EfficientNetV2B1 Eğitim Kümesi Bazlı Model Analiz Sonuçları

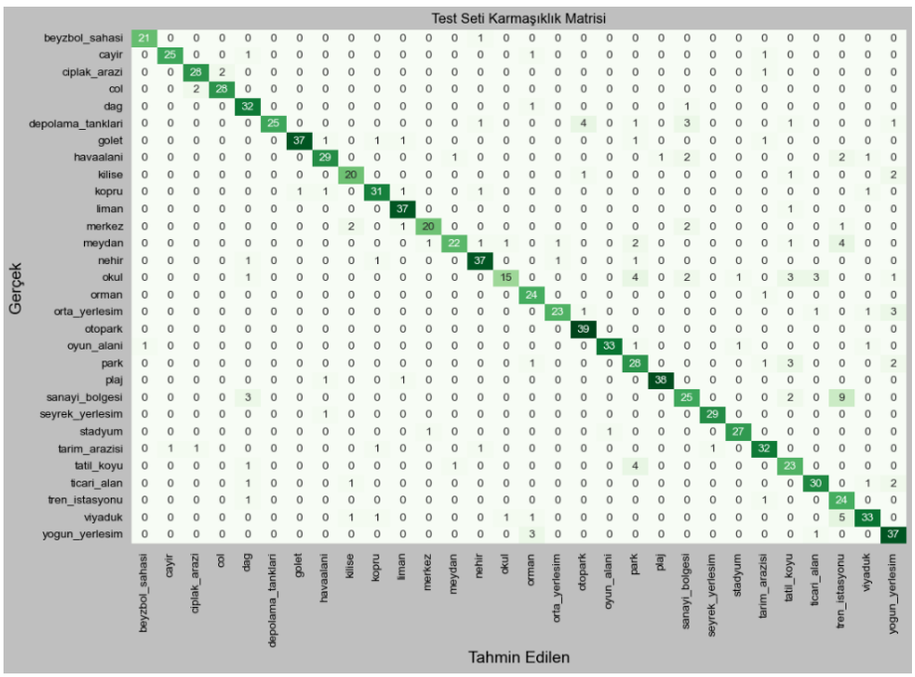
Transfer Öğrenmesi	Model	Epoch	Yığın Boyutu	Eğitim-Doğrulama-Test	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1
EfficientNetB1v2	**1**	**60**	**8**	**9000-500-500**	**0,8500**	**0,8637**	**0,8452**	**0,8435**
	2	60	8	8000-1000-1000	0,8520	0,8637	0,8524	0,8509
	3	60	8	7000-1500-1500	0,8133	0,8233	0,8093	0,8052
	4	60	8	6000-2000-2000	0,7855	0,7987	0,7838	0,7800
	5	60	8	5000-2500-2500	0,7560	0,7788	0,7595	0,7561

Tablo 62'de, EfficientNetV2B1 ön eğitimli modelin eğitim sonuçları gösterilmektedir. %90-%5-%5 bölümlendirmesine sahip 1 numaralı model; 0,8500 doğruluk, 0,8637 kesinlik, 0,8452 duyarlılık ve 0,8435 F1 skorları ile diğer bölümlendirmelerden daha iyi performans göstermiştir. EfficientNetB1 modeline göre ise daha başarılı bir doğruluk oranına sahiptir.



Şekil 204. Temel EfficientNetV2B1 Modeli Kayıp ve Doğruluk Grafiği

Şekil 204'te temel V2B1 modelinin kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde, hem eğitim hem de doğrulama kaybının 60 epoch sonunda 0,40 seviyesine indiği görülmektedir. Her iki eğri arasında da sapma veya açıklık olmadığı için aşırı öğrenme söz konusu değildir. Sağ taraftaki doğruluk grafiğinde ise 60 epoch sonunda hem eğitim hem de doğrulama doğruluğunun 0,85 seviyesinde olduğu görülmektedir. Bu grafikte de eğriler arasında açıklık bulunmamaktadır. Modele ait sınıflandırma raporu, ek 25'te verilmiştir.



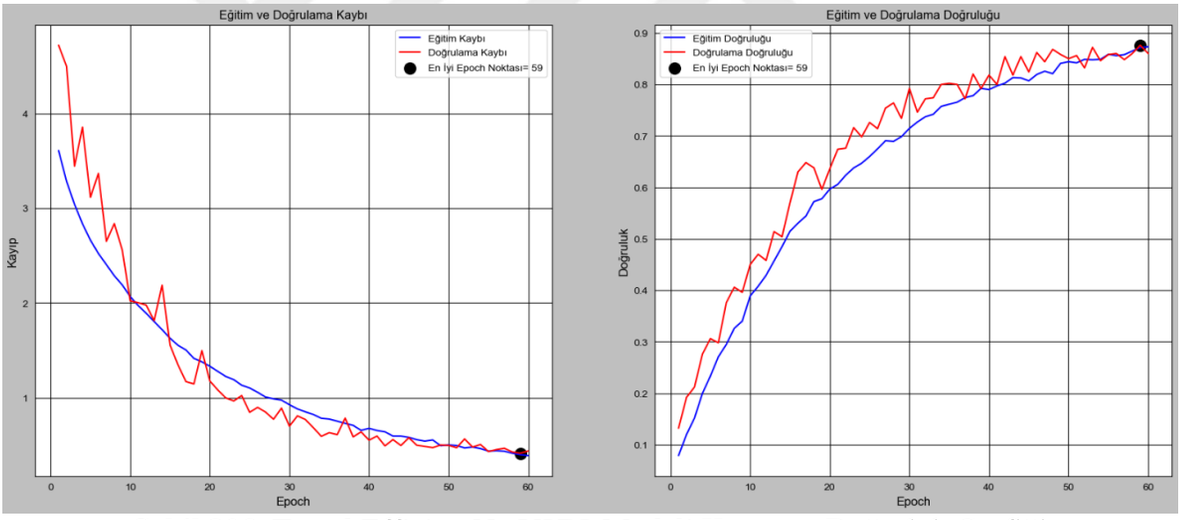
Şekil 205. Temel EfficientNetV2B1 Modeli Karmaşıklık Matrisi

Şekil 205'te ise modele ait test seti karmaşıklık matrisi gösterilmektedir. 500 test görüntüsünde 75 hatalı sınıflandırma yapılmıştır. Matriste, sanayi bölgesi sınıfından 9 örneğin tren istasyonu; viyadük sınıfından 5 örneğin tren istasyonu; depolama tankları sınıfından 4 örneği de otopark olarak sınıflandırıldığı göze çarpmaktadır.

Tablo 63. EfficientNetV2B2 Eğitim Kümesi Bazlı Model Analiz Sonuçları

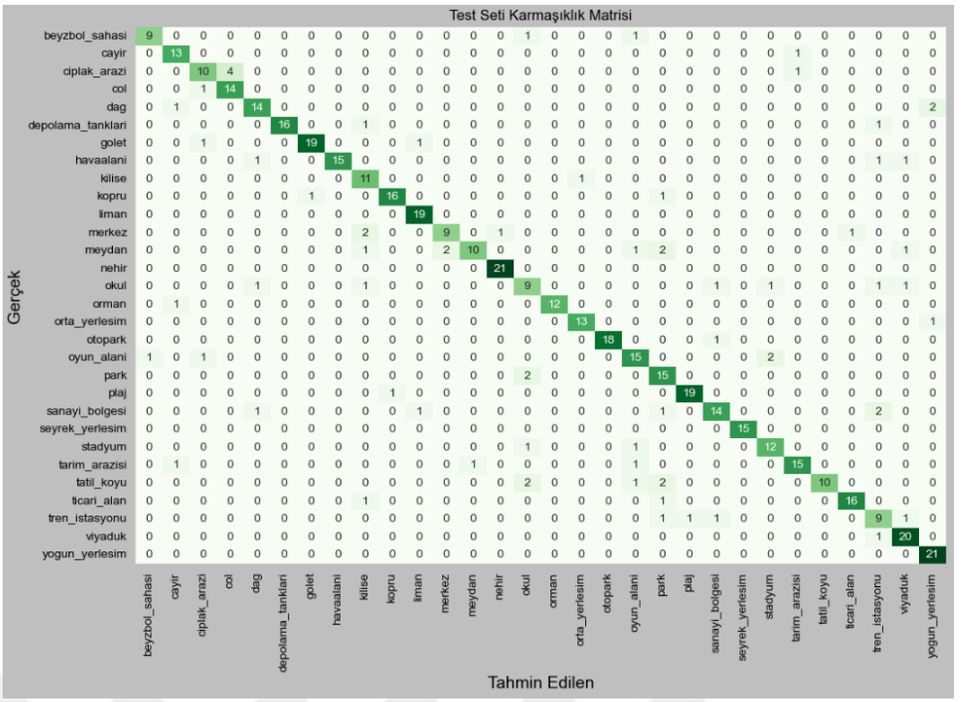
Transfer Öğrenmesi	Model	Epoch	Yığın Boyutu	Eğitim-Doğrulama-Test	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1
EfficientNetB2v2	**1**	**60**	**8**	**9000-500-500**	**0,8580**	**0,8615**	**0,8511**	**0,8508**
	2	60	8	8000-1000-1000	0.8180	0,8421	0,8125	0,8081
	3	60	8	7000-1500-1500	0,8120	0,8227	0,8074	0,8037
	4	60	8	6000-2000-2000	0,7605	0,7899	0,7588	0,7540
	5	60	8	5000-2500-2500	0,7624	0,7912	0,7602	0,7598

Tablo 63'te, temel v2B2 ön eğitimli modelin eğitim sonuçları gösterilmektedir. %90-%5-%5 bölümlendirmesine sahip 1 numaralı model; 0,8580 doğruluk, 0,8615 kesinlik, 0,8511 duyarlılık ve 0,8508 F1 skorları ile diğer bölümlendirmelerden daha iyi performans göstermiştir. Modele ait kayıp-doğruluk grafiği ile karmaşıklık matrisi sonuçları aşağıda gösterilmiştir.



Şekil 206. Temel EfficientNetV2B2 Modeli Kayıp ve Doğruluk Grafiği

Şekil 206'da temel V2B2 modelinin kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde, hem eğitim hem de doğrulama kaybının 60 epoch sonunda 0,3 seviyesine indiği görülmektedir. Her iki eğri arasında da sapma veya açıklık olmadığı için de aşırı öğrenme söz konusu değildir. Sağ taraftaki doğruluk grafiğinde ise 60 epoch sonunda hem eğitim hem de doğrulama doğruluğunun 0,876 seviyesinde olduğu görülmektedir. Bu grafikte de eğriler arasında açıklık bulunmamaktadır. Modele ait sınıflandırma raporu, ek 26'da verilmiştir.



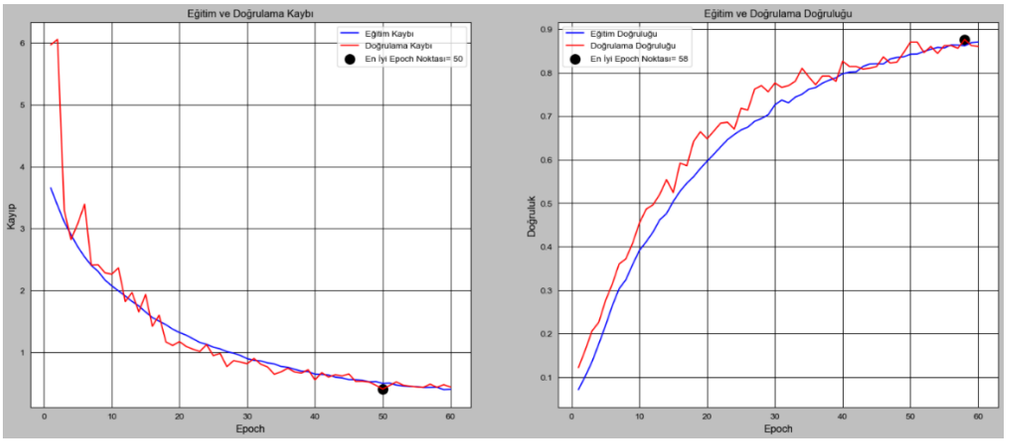
Şekil 207. Temel EfficientNetV2B2 Modeli Karmaşıklık Matrisi

Şekil 207'de ise modele ait test seti karmaşıklık matrisi gösterilmektedir. 500 test görüntüsünde 71 hatalı sınıflandırma yapılmıştır. Matriste, çıplak arazi sınıfından 4 örneğin çöl olarak sınıflandırıldığı göze çarpmaktadır.

Tablo 64. EfficientNetV2B3 Eğitim Kümesi Bazlı Model Analiz Sonuçları

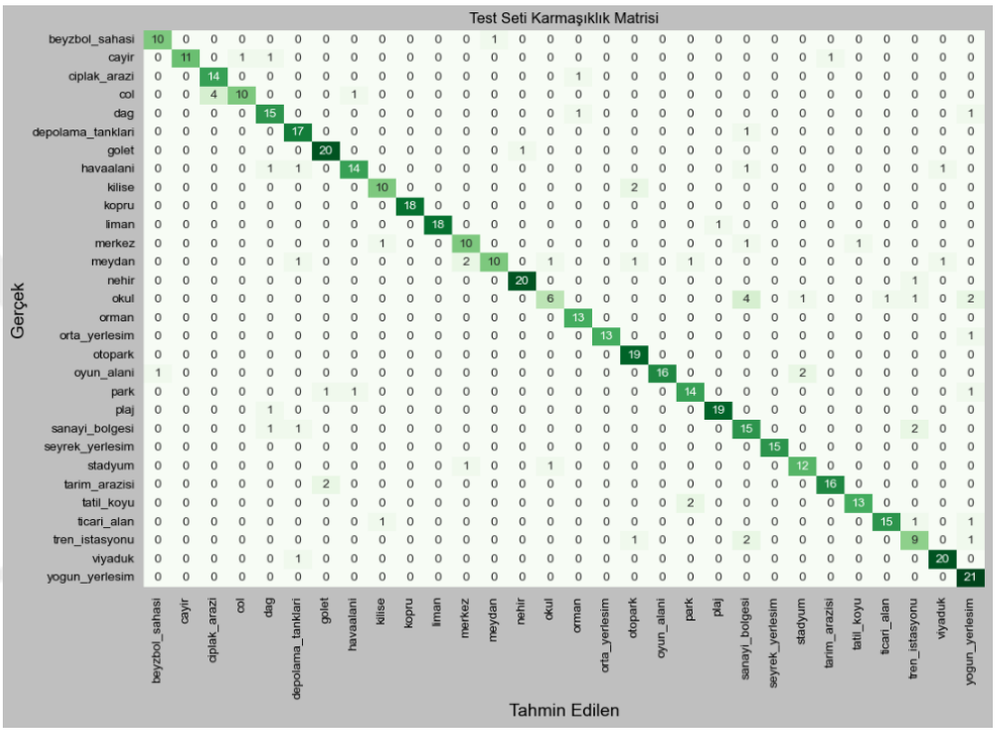
Transfer Öğrenmesi	Model	Epoch	Yığın Boyutu	Eğitim-Doğrulama-Test	Doğruluk	Kesinlik	Duyarlılık	F1
EfficientNetB3v2	**1**	**60**	**8**	**9000-500-500**	**0,8660**	**0,8718**	**0,8589**	**0,8588**
	2	60	8	8000-1000-1000	0,8420	0,8550	0,8417	0,8370
	3	60	8	7000-1500-1500	0,8273	0,8321	0,8255	0,8221
	4	60	8	6000-2000-2000	0,7825	0,7973	0,7836	0,7790
	5	60	8	5000-2500-2500	0,7140	0,7589	0,7134	0,7067

Tablo 64'te, temel V2B3 ön eğitimli modelin eğitim sonuçları gösterilmektedir. %90-%5-%5 bölümlendirmesine sahip 1 numaralı model; 0,8660 doğruluk, 0,8718 kesinlik, 0,8589 duyarlılık ve 0,8588 F1 skorları ile diğer bölümlendirmelerden daha iyi performans göstermiştir. EfficientNetB3 modeline göre ise daha başarılı bir doğruluk oranına sahiptir.



Şekil 208. Temel EfficientNetV2B3 Modeli Kayıp ve Doğruluk Grafiği

Şekil 208'de temel V2B3 modelinin kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde, hem eğitim hem de doğrulama kaybının 60 epoch sonunda 0,30 seviyesine indiği görülmektedir. Her iki eğri arasında da sapma veya açıklık olmadığı için de aşırı öğrenme söz konusu değildir. Sağ taraftaki doğruluk grafiğinde ise 60 epoch sonunda hem eğitim hem de doğrulama doğruluğunun 0,96 seviyesinde olduğu görülmektedir. Bu grafikte de eğriler arasında açıklık bulunmamaktadır. Modele ait sınıflandırma raporu, ek 27'de verilmiştir.



Şekil 209. Temel EfficientNetV2B3 Modeli Karmaşıklık Matrisi

Şekil 209'da ise modele ait test seti karmaşıklık matrisi gösterilmektedir. 500 test görüntüsünde 67 hatalı sınıflandırma yapılmıştır. Matriste, çöl sınıfından 4 örneğin çıplak arazi ve okul sınıfından da 4 örneğin sanayi bölgesi olarak sınıflandırıldığı göze çarpmaktadır.

Tablo 65. Optimize Edilmemiş EfficientNet Modelleri Sonuçlar

Optimize Edilmemiş ve Düzenlileştirilmemiş EfficientNet Modelleri								
	B0	**B1**	**B2**	**B3**	**V2B0**	**V2B1**	**V2B2**	**V2B3**
Parametre Sayısı	5.330.571	7.856.239	9.177.569	12.320.535	7.200.312	8.212.124	10.178.374	14.467.622
Bölümlendirme	9000-500-500	8000-1000-1000	9000-500-500	9000-500-500	9000-500-500	8000-1000-1000	9000-500-500	9000-500-500
Doğruluk	0,7960	0,7990	0,8220	0,8580	**0,8740**	0,8520	0,8580	0,8660
Kesinlik	0,8070	0,8146	0,8365	0,8707	0,8840	0,8637	0,8615	0,8718
Duyarlılık	0,7936	0,7977	0,8180	0,8512	0,8653	0,8524	0,8511	0,8589
F1	0,7867	0,7924	0,8168	0,8492	0,8685	0,8509	0,8508	0,8588
Doğrulama Kaybı	0,6458	0,7398	0,6548	0,4772	**0,3968**	0,5732	0,4357	0,4355

Tablo 65'te sekiz temel EfficientNet modelinin değerlendirme tablosu gösterilmektedir. Temel modellerde en başarılı performansı V2B0 varyantı göstermiştir. Ağırlıksız, ince ayar ve

hiperparametre optimizasyonu yapılmamış olmasına rağmen varyant, 0,8740'lık doğruluk, 0,8840'lık kesinlik, 0,8653'lük duyarlılık ve 0,8685'lik F1 skoruna sahiptir. Varyantın doğrulama kaybı (0,3968) ise değerlendirilen modeller içinde en düşük skora sahiptir ve en iyi genelleme kapasitesini göstermektedir. Diğer başarılı performans gösteren varyantlar arasında EfficientNetB3 (doğruluk=%85,80) ve EfficientNetV2B3 (doğruluk=%86,60) yer almaktadır, ancak bu varyantların doğrulama kayıpları (sırasıyla 0,4772 ve 0,4355) V2B0 varyantından daha yüksektir. Daha büyük modellerin (B3, V2B3) daha fazla derinliğe sahip olduğu görülse de, bu durum en başarılı performansın gösterilmesine yetmemektedir. Bir diğer gözlem ise V2 varyantlarının, genellikle doğruluk ve doğrulama kaybı açısından, orijinal EfficientNet modellerinden daha iyi performans göstermesidir. Örneğin; V2B0, B1 modelinden daha az parametre sayısına sahip olmasına rağmen, B1 modelinden iyi skorlar elde etmiştir. Bu durum, V2 mimarisin iyileştirmeleri (örneğin; birleştirilmiş MBConv blokları ve kademeli öğrenme) aracılığı ile mümkün olmaktadır.

5.2.2. Optimize Edilmiş ve Düzenlileştirilmiş Modeller

Baştan ve ağırlıksız eğitilen temel modellerin başarımını artırmak için, Imagenet ağırlıkları modele eklenmiş, optimizasyon ve düzenlileştirme teknikleri, yoğun katman ekleme ve veri artırma[244] teknikleri uygulanmıştır. Bu tekniklere ek olarak Keras Tuner kütüphanesi ile en iyi hiperparametre değerlerinin bulunması amaçlanmıştır. Veri setinin büyüklüğünden dolayı Grid Search (Izgara Araması) yapılırken ekran kartı hafıza problemleri ile karşılaşılmasından dolayı Random Search (Rastgele Arama) tekniği kullanılmış ve bu doğrultuda en iyi hiperparametre değerlerinin bulunması sağlanmıştır. Random Search ile aranan hiperparametreler; öğrenme oranı, optimizasyon algoritması, yoğun katman nöron sayısı, L1 ve L2 çekirdek düzenlileştiricisi, L2 bias düzenlileştiricisi ve L2 aktivite düzenlileştiricisi olarak belirlenmiştir.

Çekirdek düzenleyicileri, hiperparametrelerin ayarlanması sırasında, sinir ağındaki ağırlıklara düzenlileştirme uygulamak için kullanılmaktadır. Temel işlevleri ise büyük ağırlık değerlerine ceza ekleyerek aşırı öğrenmeyi önlemek ve modelin daha iyi genelleme yapmasını sağlamaktır (Bietti vd., 2019). Feinman ve Lake (2019) ise, modern CNN mimarilerinin çekirdek ağırlıklarını saflaştırarak örnek karmaşıklıklarını iyileştirebilmek için, mekansal korelasyonları artıran bir çekirdek düzenleyicisi önermişlerdir. Bu düzenleyicinin korelasyon parametreleri, ağın önceki deneyimlerden öğrenilmektedir. Tezde, Feinman ve Lake (2019) tarafından

[244] Data augmentation

yapılan çalışmanın kodlarının bulunduğu GitHub platformunda[245] gösterildiği üzere, çekirdek düzenleyicileri kullanılmıştır. Bias (önyargı) düzenleyicisi ise katmanlardaki önyargı değerlerine düzenlileştirme uygular. Temel işlevi, büyük önyargı değerlerini cezalandırarak aşırı uyumu önlemektir (Sohaee, 2023). Önyargı, ağırlıklardan ve önceki aktivasyonlardan gelen sonucu topladıktan sonra nöronun pozitif olup olmadığının görülmesi için bir eşik değeri gibi davranır. Bu nedenle, düzenlileştirmede çoğunlukla ağırlıklara odaklanıldığından, çekirdek düzenleyicilerine göre çok daha az kullanılmaktadır. Tezde kullanılan önyargı düzenleyicisi, Keras dokümantasyonunda[246] gösterildiği gibi analizlerde kullanılmıştır. Aktivite düzenleyicisi de, ağırlık veya bias değerleri yerine ağdaki katmanların çıktı aktivasyonlarına düzenlileştirme uygular. Temel işlevi ise büyük aktivasyon değerlerini cezalandırarak seyrekliği ve genelleştirmeyi iyileştirebilecek aktivasyonların küçük kalmasını sağlamaktır (Shi vd., 2019).

Transfer öğrenmesi sırasında ön eğitimli modeller, oldukça genel veri setleri üzerinden eğitildiğinden, kullanılan veri setlerinin özelliklerini bire bir taşımayabilir. Dolaysıyla, modele fazladan yoğun katman[247] eklemek, ön eğitimli modelin veri setine özgü özellikleri ve üst düzey ilişkileri öğrenmesini daha iyi sağlayabilir (Yosinski vd., 2014).

Öğrenme oranı,[248] modelin eğitim sırasında ağırlıklarını ne kadar güncellediğini kontrol ettiği için en önemli hiperparametrelerden biri olarak değerlendirilmektedir (Wu ve Liu, 2023, s. 1; IIemobayo vd., 2024, s. 391; Peng, 2024, s. 184). Ön eğitimli modellerin başarımını optimize etmek için, doğru bir öğrenme oranını seçme işi ise oldukça önemlidir. Ön eğitimli modeller, ImageNet gibi büyük veri kümelerinden genel görüntü özelliklerini öğrenmiştir ancak yüksek bir öğrenme oranı seçimi, verinin faydalı özelliklerini çok hızlı bir şekilde kaybedebilirken, düşük öğrenme oranı seçimi ise öğrenilen bilgiyi korur ve ince ayarın daha sağlıklı yapılmasını sağlayabilir. Ancak; öğrenme oranı olması gerekenden daha da düşükse, eğitim çok yavaş olacaktır; öğrenme oranı çok yüksek olursa da, model salınabilir ve yakınsamayabilir.

Optimizasyon algoritmaları, eğitim sırasında kayıp fonksiyonunu en aza indirmek için modelin ağırlıklarını güncelleddiklerinden son derece önemlidir. Transfer öğrenmesi uygulamasında kullanılan optimizasyon algoritmalarından Adam[249], sinir ağlarının eğitimi sırasında kayıp fonksiyonunu en aza indirmek için kullanılan yinelemeli bir optimizasyon algoritmasıdır.

[245] https://github.com/rfeinman/SK-regularization
[246] https://keras.io/api/layers/regularizers/
[247] Dense layer
[248] Learning rate
[249] Adaptive Moment Estimation

RMSprop ve momentumlu Stokastik Gradyan İnişi'nin bir kombinasyonu olarak değerlendirilmektedir (Reyad, Sarhan ve Arafa, 2023, s. 17100; Hassan vd., 2023, s. 16604). Adamax ise, Adam'ın bir varyantı olan ancak güncellemeleri ölçekleyebilmek için L2 normu yerine sonsuzluk normunu (L_∞) kullanan uyarlanabilir bir optimizasyon algoritmasıdır. Adam'a kıyasla genellikle daha kararlı ağırlık güncellemeler sağlar (Yi, Ahn ve Ji, 2020, s. 2). Adagrad (Adaptive Gradient Algorithm), geçmiş gradyan büyüklüklerine göre her parametrenin öğrenme oranını uyarlayan bir optimizasyon algoritmasıdır ve seyrek verilerdeki özellikleri tanımlamada başarılıdır. Seyrek parametrelere daha büyük güncellemeler ve sık parametrelere daha küçük güncellemeler verir (Liang vd., 2024). RMSprop ise Adagrad'ın agresif öğrenme oranındaki düşüşü aşmak için tasarlanmıştır. Adagrad gibi, RMSprop da her parametrenin öğrenme oranını geçmiş gradyanlara göre ölçeklendirir, ancak tüm geçmiş gradyanların karesini biriktirmek yerine, kare gradyanların üstel olarak azalan ortalamasını korur (Xu vd., 2021).

Çalışmada kullanılan hiperparametreler için belirlenen aralıklar ek 28'de verilmiştir. Eğitim süreçlerinde optimizasyon tekniklerinden erken durdurma ve öğrenme oranı ayarlayıcısı (ReduceLROnPlateau) kullanılmıştır. Keras Tuner Random Search üzerinden elde edilen en iyi öğrenme oranı belirlendikten sonra, öğrenme oranı ayarlayıcısına bu değer verilmiştir. Erken durdurma tekniğinde ise "patience" değeri 3, "versobe" değeri 1 ve "restore_best_weights" değeri ise "True" olarak belirlenmiştir.
Düzenlileştirme tekniklerinden veri artırma (ImageDataGenerator) da kullanılmış olup, değerleri aşağıdaki tabloda gösterilmiştir.

Tablo 66. Modellere Uygulanan Düzenlileştirme Tekniği ve Değerleri

Teknik	Parametre	Değer
ImageDataGenerator	horizontal_flip	True
	rotation_range	25
	width_shift_range	0,25
	height_shift_range	0,25
	zoom_range	0,25

Modelde kayıp fonksiyonu olarak "categorical_crossentropy" kullanılmıştır. Problem türü çok sınıflı sınıflandırma olduğu için bu kayıp fonksiyonu sıklıkla kullanılmaktadır.

Temel modellere yapılan bu optimizasyon ve düzenlileştirme teknikleri, tahmin skorlarını daha da geliştirmek ve aşırı öğrenme veya eksik öğrenmenin önüne geçilmesini sağlayarak, modellerin çok daha sağlıklı yorumlanabilmesine yardımcı olmaktadır.

İlk analiz temel model B0 üzerinde uygulanmıştır. EfficientNetB0 temel modeli için aramalar sonucunda elde edilen en iyi değerler Tablo 67'de gösterilmektedir.

Tablo 67. EfficientNetB0 Modeline Ait En İyi Hiperparametreler ve Değerleri

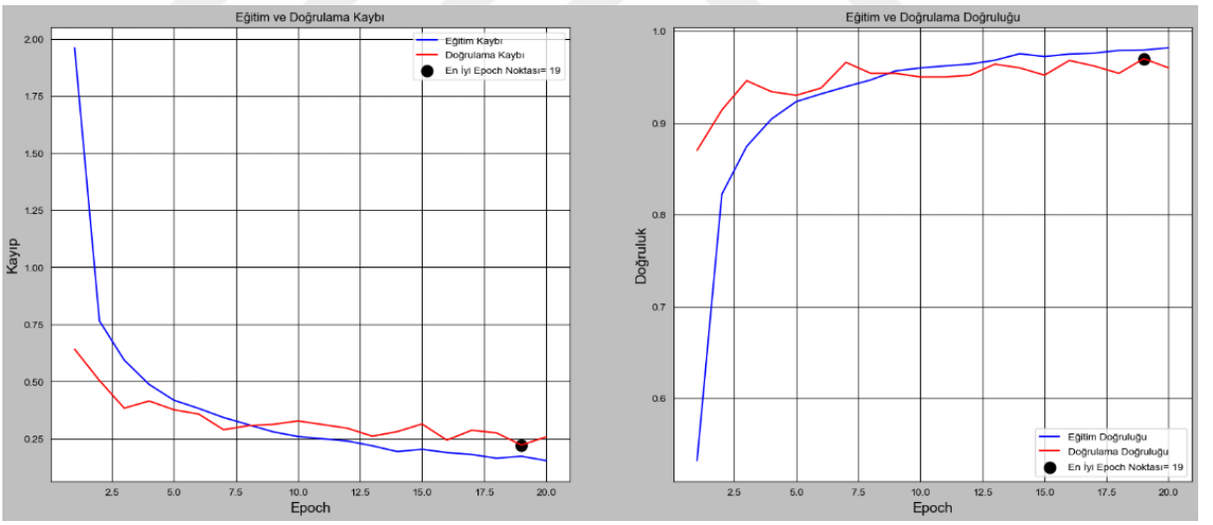
Hiperparametre	En İyi Değer
kernel_regularizer L1	2,9630903518381 59e-05
kernel_regularizer L2	0,0002528402402636253
bias_regularizer L2	0,0003188261203409189
activity_regularizer L2	2,0407320985338325e-05
learning_rate	0,001
optimizasyon algoritması	Adamax

Aramada en iyi dense katmanı nöron değeri ise 384 olarak belirlenmiştir. kernel_regularizer L2 değeri, L1 değerinden daha iyi skor verdiği için tercih edilmiştir. Yukarıda gösterilen en iyi hiperparametre değerleri ile yapılan eğitim sonuçları aşağıdaki tabloda gösterilmektedir.

Tablo 68. Optimize Edilmiş EfficientNetB0 Modeline Ait Analiz Sonuçları

Transfer Öğrenmesi Modeli	Model No	Epoch	Yığın Boyutu	Eğitim-Doğrulama-Test	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1
EfficientNetB0	1	20	16	9000-500-500	0,9640	0,9651	0,9600	0,9608
			32	9000-500-500	0,9640	0,9641	0,9594	0,9600

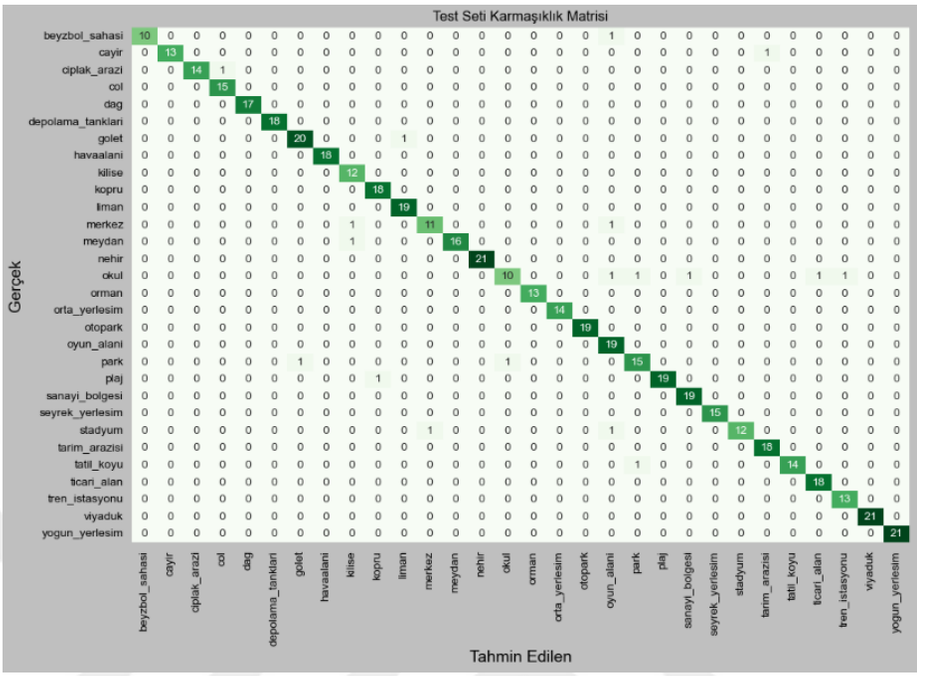
Tablo 68'de görüldüğü üzere, temel modeldeki 0,7960'lık doğruluk oranı, 0,9640'a çıkmıştır. Buna ek olarak "batch_size" parametresi 16 olan model, 32 olan modele göre daha iyi performans göstermiştir. Optimize edilen modele dair kayıp doğruluk grafiği ve test seti karmaşıklık matrisi aşağıda gösterilmiştir.



Şekil 210. Optimize Edilmiş EfficientNetB0 Modeli Kayıp ve Doğruluk Grafiği

Şekil 210'da optimize edilmiş B0 modelinin kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde, eğitim kaybının 20 epoch sonunda 0,14 seviyesine, doğrulama kaybının ise 0,25 seviyesine indiği görülebilir. İki eğrinin açıklık farkı ise yaklaşık 0,1'dir. Her iki eğri de birbirinden sapmadığı için aşırı öğrenme söz konusu değildir. Sağ taraftaki doğruluk grafiğinde ise 20. epoch itibariyle eğitim doğruluğunun 0,97, doğrulama doğruluğunun ise 0,95 seviyesinde olduğu görülmektedir. Eğitim ve doğrulama doğruluğu arasındaki açıklık farkının oldukça küçük (0,02) olması, genelleme yeteneğinin oldukça iyi

olduğunu göstermektedir. Eğriler ve skorlardan da görüldüğü üzere, düzenleme ve ince ayar teknikleri hem eğitim hem de doğrulama aşamalarında oldukça düzgün ve kararlı bir yakınsamayı sağlamıştır. Modele ait sınıflandırma raporu, ek 29'da verilmiştir.



Şekil 211. Optimize Edilmiş EfficientNetB0 Modeli Karmaşıklık Matrisi

Şekil 211'de ise modele ait karmaşıklık matrisi gösterilmektedir. İlk model ile karşılaştırıldığında oldukça yüksek bir doğruluk oranına sahip olan matriste yanlış sınıflama oranının oldukça azaldığı görülebilir. Optimize edilmiş modelde 500 test görüntüsünden sadece 18 tanesi hatalı olarak sınıflandırılmıştır. Bu oran ilk modelde 102 idi.

İkinci analiz temel model B1 üzerinde uygulanmıştır. EfficientNetB1 temel modeli için aramalar sonucunda elde edilen en iyi değerler Tablo 69'da gösterilmektedir.

Tablo 69. EfficientNetB1 Modeline Ait En İyi Hiperparametreler ve Değerleri

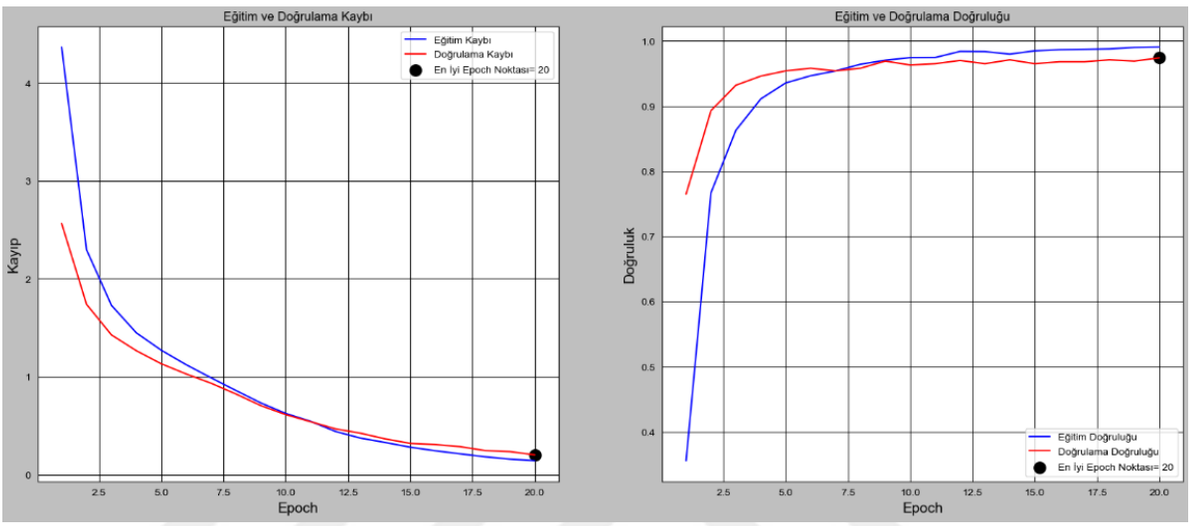
Hiperparametre	En İyi Değer
kernel_regularizer L1	8,237059732496302e-05
kernel_regularizer L2	0,0005879021722490936
bias_regularizer L2	1,4461073827695563e-05
activity_regularizer L2	0,0007512035157174944
learning_rate	0,0001
optimizasyon algoritması	RMSprop

Aramalar sonucunda en iyi dense nöron sayısı 512 olarak bulunmuştur. L1 çekirdek düzenlileştiricisi, L2 düzenlileştiricine göre daha iyi performans gösterdiğinden tercih edilmiştir. Yukarıda gösterilen en iyi hiperparametre değerleri ile yapılan eğitim sonuçları Tablo 70'te gösterilmektedir.

Tablo 70. Optimize Edilmiş EfficientNetB1 Modeline Ait Analiz Sonuçları

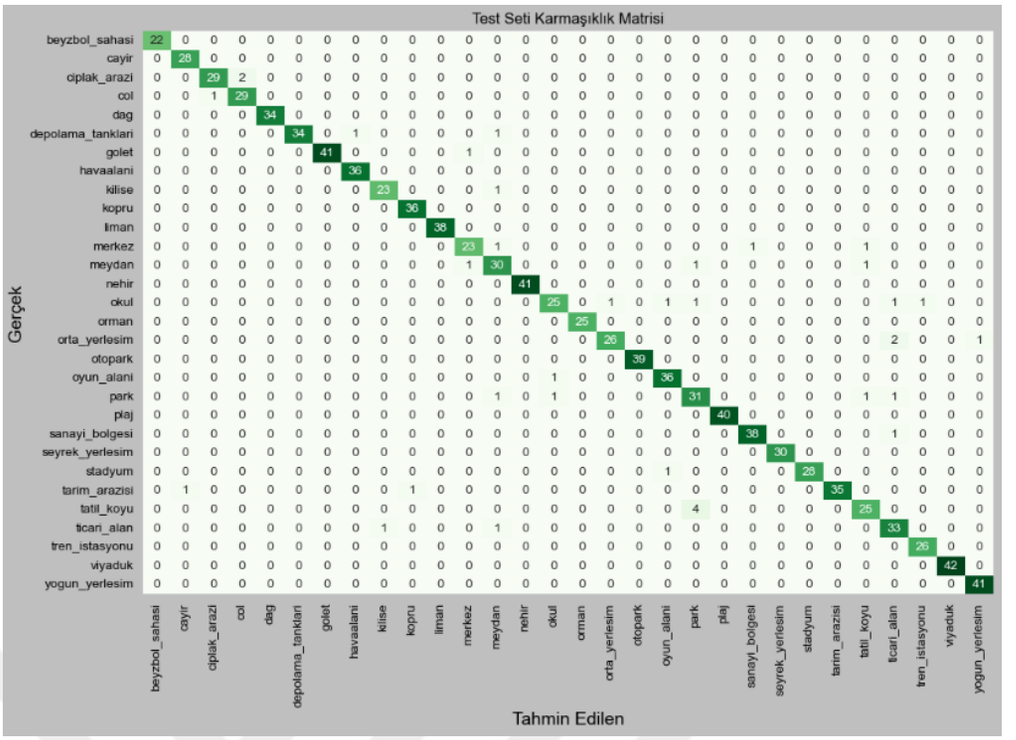
Transfer Öğrenmesi Modeli	Model Adı	Epoch	Yığın Boyutu	Eğitim-Doğrulama-Test	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1
EfficientNetB1	Temel2	20	16	8000-1000-1000	0,9640	0,9633	0,9618	0,9622
	Temel2	20	32	8000-1000-1000	0,9590	0,9593	0,9562	0,9562

Tablo 70'te görüldüğü üzere, yığın boyutu 16 olan model, 32 olana göre daha iyi doğruluk skoru vermiştir. Optimize edilen modele dair kayıp doğruluk grafiği ve test seti karmaşıklık matrisi aşağıda gösterilmiştir.



Şekil 212. Optimize Edilmiş EfficientNetB1 Modeli Kayıp ve Doğruluk Grafiği

Şekil 212'de optimize edilmiş B1 modelinin kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde, hem eğitim hem de doğrulama kaybının 20 epoch sonunda 0,2 seviyesine indiği görülmektedir. İki eğrinin açıklık farkı bulunmamaktadır. Kayıp grafiğindeki bu durum, neredeyse mükemmel genelleştirilebilirliği göstermektedir. Sağ taraftaki doğruluk grafiğinde ise 20. epoch itibariyle eğitim doğruluğunun 0,99, doğrulama doğruluğunun ise 0,97 seviyesinde olduğu görülmektedir. Her iki grafikte de eğrilerin pürüzsüz ve stabil olması, düzenleme ve ince ayar tekniklerinin veri üzerindeki genelleme ve temsilde oldukça etkili olduğunu göstermektedir. Modele ait sınıflandırma raporu, ek 30'da verilmiştir.



Şekil 213. Optimize Edilmiş EfficientNetB1 Modeli Karmaşıklık Matrisi

Şekil 213'te ise modele ait karmaşıklık matrisi gösterilmektedir. İlk model ile karşılaştırıldığında oldukça yüksek bir doğruluk oranına sahip olan matriste yanlış sınıflama oranının oldukça azaldığı görülebilir. Optimize edilmiş modelde 1000 test görüntüsünden sadece 36 tanesi hatalı olarak sınıflandırılmıştır. Bu oran ilk modelde 201 idi. Matriste, tatil köyü sınıfının 4 örneğinin, park olarak sınıflandırıldığı göze çarpmaktadır.

Üçüncü analiz temel model B2 üzerinde uygulanmıştır. EfficientNetB2 temel modeli için aramalar sonucunda elde edilen en iyi değerler Tablo 71'de gösterilmektedir.

Tablo 71. EfficientNetB2 Modeline Ait En İyi Hiperparametreler ve Değerleri

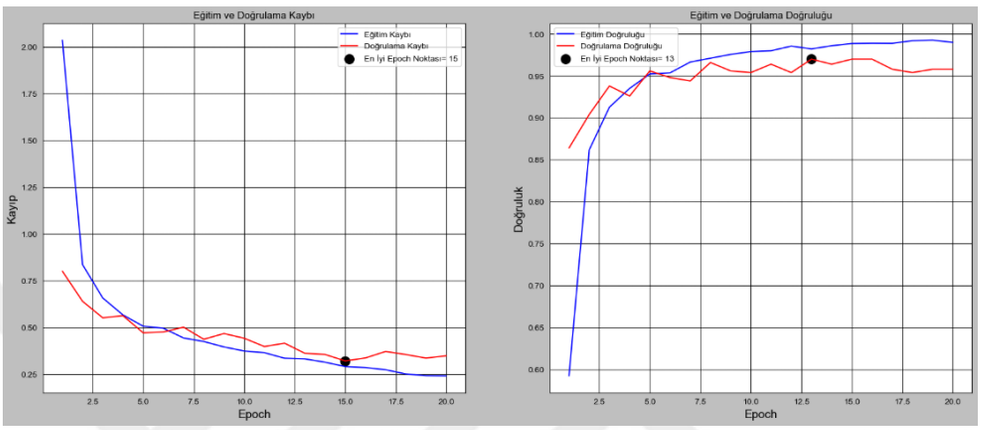
Hiperparametre	En İyi Değer
kernel_regularizer L1	1,5661588358752978e-05
kernel_regularizer L2	6,175710404430179e-05
bias_regularizer L2	0,00011694328713687794
activity_regularizer L2	6,948789957181235e-05
learning_rate	0,001
optimizasyon algoritması	Adamax

Aramalar sonucunda en iyi yoğun katman nöron değeri 512 olarak bulunmuştur. L1 çekirdek düzenlileştiricisi, L2 düzenlileştiricsine göre daha iyi performans gösterdiğinden tercih edilmiştir. Yukarıda gösterilen en iyi hiperparametre değerleri ile yapılan eğitim sonuçları Tablo 72'de gösterilmektedir.

Tablo 72. Optimize Edilmiş EfficientNetB2 Modeline Ait Analiz Sonuçları

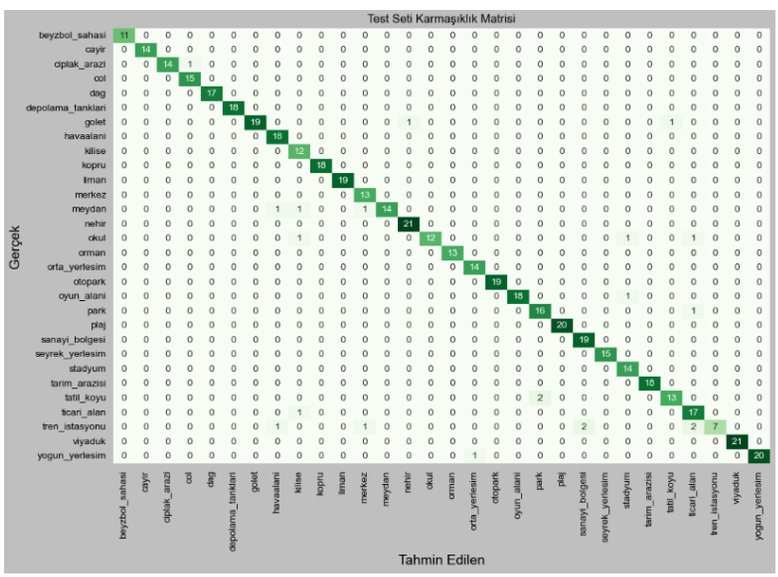
Transfer Öğrenmesi Modeli	Model Adı	Epoch	Yığın Boyutu	Eğitim-Doğrulama-Test	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1
EfficientNetB2	Temel1	20	16	9000-500-500	0,9540	0,9558	0,9491	0,9503
EfficientNetB2	Temel1	20	32	9000-500-500	0,9580	0,9600	0,9551	0,9532

Tablo 72'den de görüldüğü üzere, yığın boyutu 32 olan model, 16 olana göre daha iyi doğruluk skoru vermiştir. Optimize edilen modele dair kayıp doğruluk grafiği ve test seti karmaşıklık matrisi aşağıda gösterilmiştir.



Şekil 214. Optimize Edilmiş EfficientNetB2 Modeli Kayıp ve Doğruluk Grafiği

Şekil 214'te optimize edilmiş B2 modelinin kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde, eğitim kaybının 20 epoch sonunda 0,25 seviyesine, doğrulama kaybının ise 0,35'e indiği görülmektedir. İki eğrinin açıklık farkı yaklaşık 0,1'dir. En düşük doğrulama kaybının, optimum nokta olarak seçilen 15. epoch'ta gerçekleştiği görülebilir. Sağ taraftaki doğruluk grafiğinde ise 20. epoch itibariyle eğitim doğruluğunun 0,99, doğrulama doğruluğunun ise 0,96 seviyesinde olduğu görülmektedir. Eğriler arasındaki 0,03'lük bir açıklık bulunmaktadır. Modele ait sınıflandırma raporu, ek 31'de verilmiştir.



Şekil 215. Optimize Edilmiş EfficientNetB2 Modeli Karmaşıklık Matrisi

Şekil 215'te, ilk model ile karşılaştırıldığında oldukça yüksek bir doğruluk oranına sahip olan matriste yanlış sınıflama oranının da çok azaldığı görülebilir. Optimize edilmiş modelde, modelin daha önceden hiç görmediği 500 test görüntüsünden sadece 21 tanesi hatalı olarak sınıflandırılmıştır. Bu oran ilk modelde 89 idi. Matriste göze çarpan durum ise tren istasyonu sınıfından 6 örneğin yanlış sınıflandırılmasıdır.

Dördüncü analiz temel model B3 üzerinde uygulanmıştır. EfficientNetB3 temel modeli için aramalar sonucunda elde edilen en iyi değerler Tablo 73'te gösterilmektedir.

Tablo 73. EfficientNetB3 Modeline Ait En İyi Hiperparametreler ve Değerleri

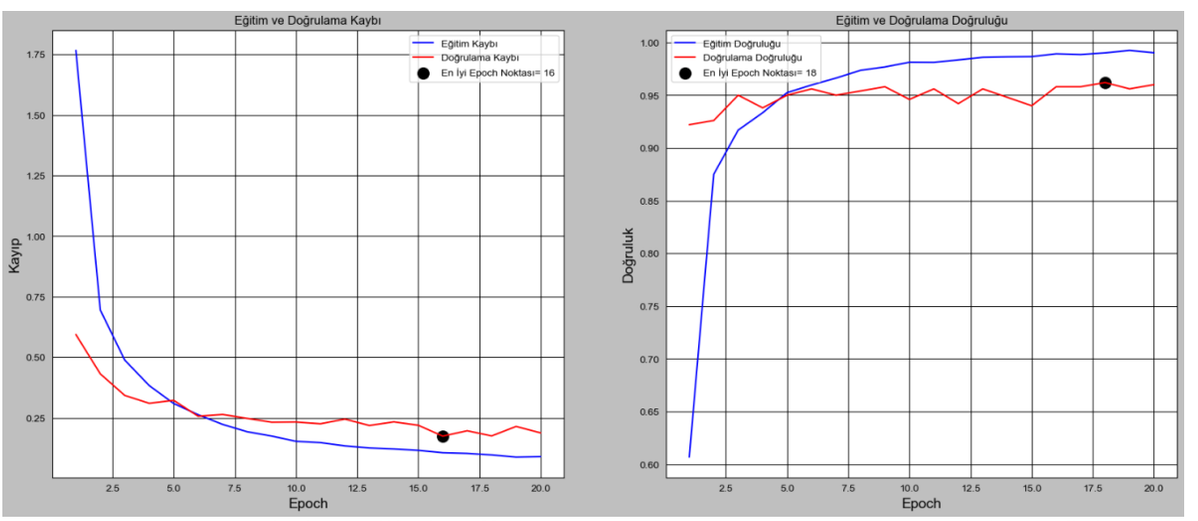
Hiperparametre	En İyi Değer
kernel_regularizer L1	2,039753016236578e-06
kernel_regularizer L2	4,075389089290891e-05
bias_regularizer L2	2,2244311682341678e-05
activity_regularizer L2	0,0005660488174994738
learning_rate	0,001
optimizasyon algoritması	Adamax

Aramalar sonucunda model için en iyi yoğun katman nöron sayısı 512 olarak bulunmuştur. L2 çekirdek düzenlileştiricisi, L1 düzenlileştiricisine göre daha iyi performans gösterdiğinden dolayı tercih edilmiştir.

Tablo 74. Optimize Edilmiş EfficientNetB3 Modeline Ait Analiz Sonuçları

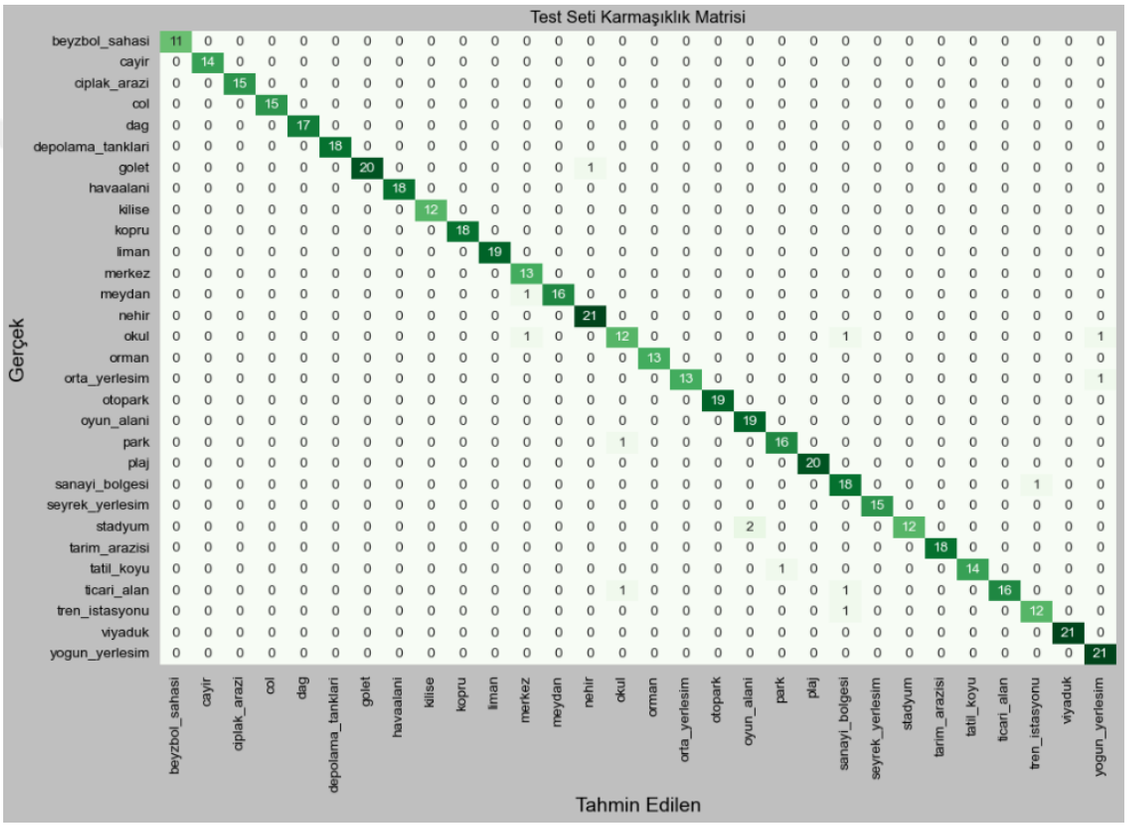
Transfer Öğrenmesi Modeli	Model Adı	Epoch	Yığın Boyutu	Eğitim-Doğrulama-Test	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1
EfficientNetB3	Temel1	20	16	9000-500-500	0,9720	0,9739	0,9704	0,9713
EfficientNetB3	Temel1	20	32	9000-500-500	0,9620	0,9624	0,9593	0,9585

Tablo 74'te görüldüğü üzere bu modelde en iyi sonucu yığın boyutu 16 olan model vermiştir. Optimize edilen modele dair kayıp doğruluk grafiği ve test seti karmaşıklık matrisi aşağıda gösterilmiştir.



Şekil 216. Optimize Edilmiş EfficientNetB3 Modeli Kayıp ve Doğruluk Grafiği

Şekil 216'da optimize edilmiş B3 modelinin kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde, eğitim kaybının 20 epoch sonunda 0,10 seviyesine, doğrulama kaybının ise 0,18'e indiği görülmektedir. İki eğrinin açıklık farkı yaklaşık 0,08'dir. Her iki eğri arasındaki açıklığın son derece az olması, aşırı öğrenmenin olmadığı ve oldukça iyi bir genelleme performansını göstermektedir. En düşük doğrulama kaybının, optimum nokta olarak seçilen 16. epochta gerçekleştiği görülebilir. Sağ taraftaki doğruluk grafiğinde ise 20. epoch itibariyle eğitim doğruluğunun 0,99, doğrulama doğruluğunun ise 0,96 seviyesinde olduğu görülmektedir. Eğriler arasında 0,03'lük bir açıklık bulunmaktadır. Model yüksek genelleştirebilirliğe sahiptir. Modele ait sınıflandırma raporu, ek 32'de verilmiştir.



Şekil 217. Optimize Edilmiş EfficientNetB3 Modeli Karmaşıklık Matrisi

Şekil 217'de modelin karmaşıklık matrisi gösterilmektedir. Optimize edilmiş modelde, modelin daha önceden hiç görmediği 500 test görüntüsünden sadece 14 tanesi hatalı olarak sınıflandırılmıştır. Bu oran ilk modelde 71 idi. Matriste, stadyum sınıfından 2 örneğin oyun alanı olarak sınıflandırıldığı göze çarpmaktadır.

Beşinci analiz temel model V2B0 üzerinde uygulanmıştır. EfficientNetV2B0 temel modeli için aramalar sonucunda elde edilen en iyi değerler Tablo 75'te gösterilmektedir.

Tablo 75. EfficientNetV2B0 Modeline Ait En İyi Hiperparametreler ve Değerleri

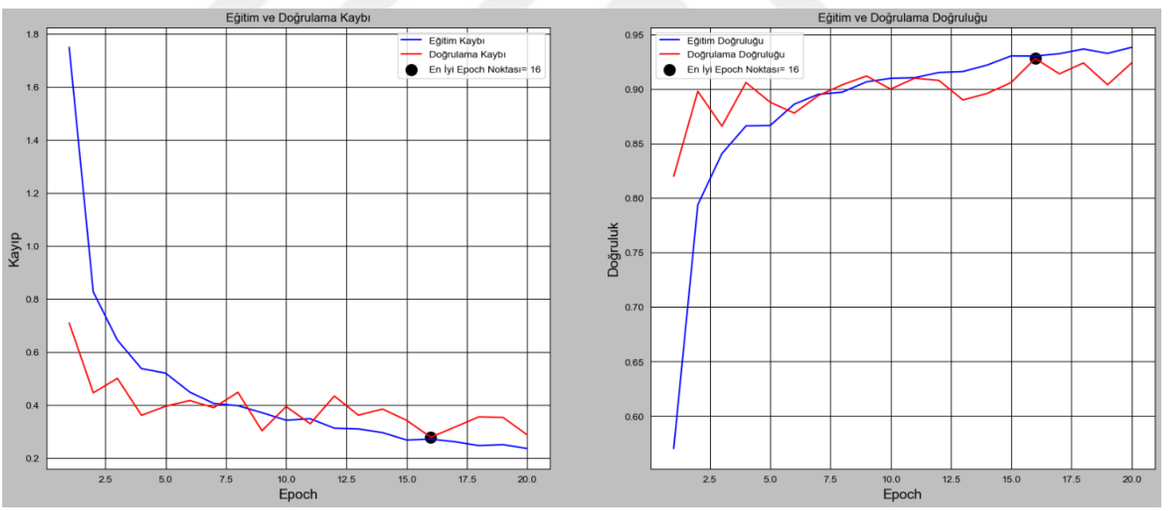
Hiperparametre	En İyi Değer
kernel_regularizer L1	1,3792248662960861e-06
kernel_regularizer L2	8,901416500660485e-06
bias_regularizer L2	2,845295893599345e-06
activity_regularizer L2	0,0001633138352209446
learning_rate	0,001
optimizasyon algoritması	Adam

Aramalar sonucunda model için en iyi yoğun katman nöron sayısı 384 olarak bulunmuştur. L2 çekirdek düzenlileştiricisi, L1 düzenlileştiricisine göre daha iyi performans gösterdiğinden tercih edilmiştir.

Tablo 76. Optimize Edilmiş EfficientNetV2B0 Modeline Ait Analiz Sonuçları

Transfer Öğrenmesi Modeli	Model Adı	Epoch	Yığın Boyutu	Eğitim-Doğrulama-Test	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1
EfficientNetB0v2	Temel1	20	16	9000-500-500	0,9320	0,9347	0,9271	0,9289
	Temel1	20	32	9000-500-500	0,9040	0,9101	0,8981	0,8978

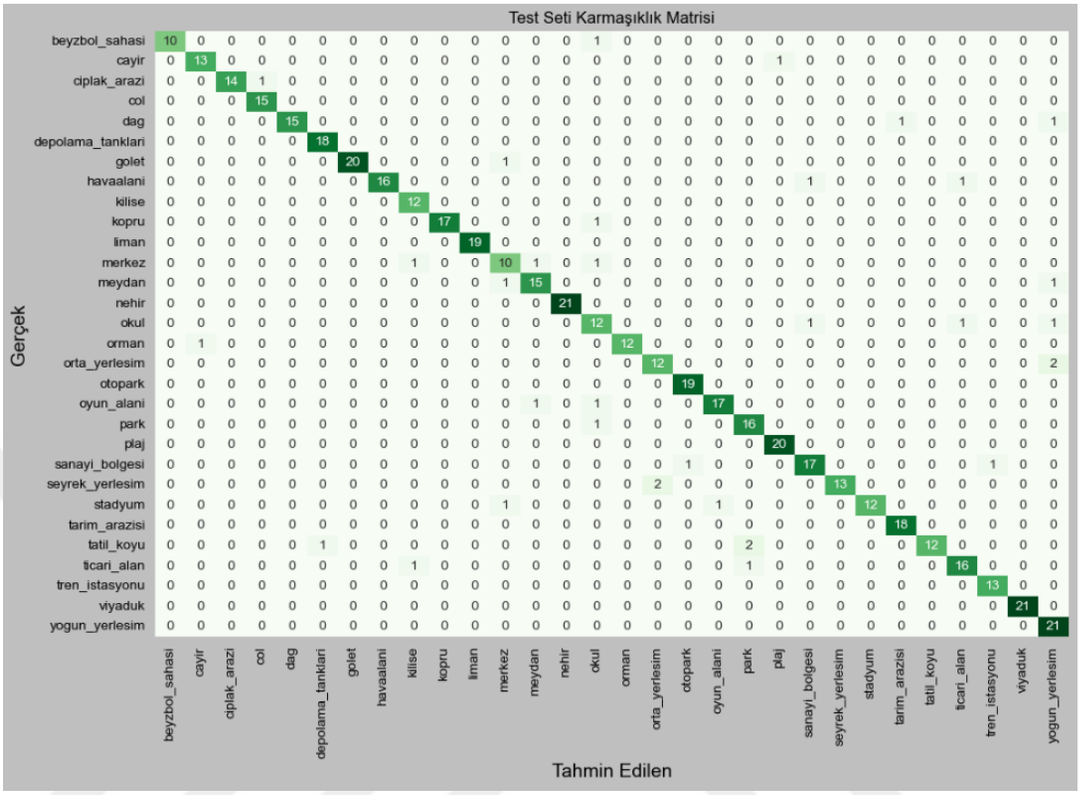
Tablo 76'da görüldüğü üzere bu modelde en iyi sonucu yığın boyutu 16 olan model vermiştir. Optimize edilen modele dair kayıp doğruluk grafiği ve test seti karmaşıklık matrisi aşağıda gösterilmiştir.



Şekil 218. Optimize Edilmiş EfficientNetV2B0 Modeli Kayıp ve Doğruluk Grafiği

Şekil 218'de optimize edilmiş V2B0 modelinin kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde, eğitim kaybının 20 epoch sonunda 0,25 seviyesine, doğrulama kaybının ise 0,30'a indiği görülmektedir. İki eğrinin açıklık farkı yaklaşık 0,05'dir. Her iki eğri arasındaki açıklığın son derece az olması, aşırı öğrenmenin olmadığı ve oldukça iyi bir genelleme performansını göstermektedir. Grafikte dikkati çeken nokta ise doğrulama doğruluğunun gereğinden fazla dalgalanmasıdır. Dalgalanmalara rağmen, eğitim ve doğrulama kaybı arasında büyük bir sapma yoktur. En düşük doğrulama kaybının, optimum nokta olarak seçilen 16. epoch'ta gerçekleştiği görülebilir. Sağ taraftaki doğruluk grafiğinde ise 20. epoch

itibariyle eğitim doğruluğunun 0,94, doğrulama doğruluğunun ise 0,93 seviyesinde olduğu görülmektedir. Eğriler arasında neredeyse hiç açıklık yoktur. Modele ait sınıflandırma raporu, ek 33'te verilmiştir.



Şekil 219. Optimize Edilmiş EfficientNetV2B0 Modeli Karmaşıklık Matrisi

Şekil 219'da modele ait karmaşıklık matrisi görülmektedir. İlk model ile karşılaştırıldığında oldukça yüksek bir doğruluk oranına sahip olan matriste yanlış sınıflama oranının da çok azaldığı görülebilir. Optimize edilmiş modelde, modelin daha önceden hiç görmediği 500 test görüntüsünden 34 tanesi hatalı olarak sınıflandırılmıştır. Bu oran ilk modelde 63 idi. Matriste seyrek yerleşim (2), tatil köyü (2), orta yerleşim (2) sınıflarının, sırasıyla orta yerleşim, park ve yoğun yerleşim olarak sınıflandırıldığı göze çarpmaktadır.

Altıncı analiz temel model V2B1 üzerinde uygulanmıştır. EfficientNetV2B1 temel modeli için aramalar sonucunda elde edilen en iyi değerler aşağıdaki tabloda gösterilmektedir.

Tablo 77. EfficientNetV2B1 Modeline Ait En İyi Hiperparametreler ve Değerleri

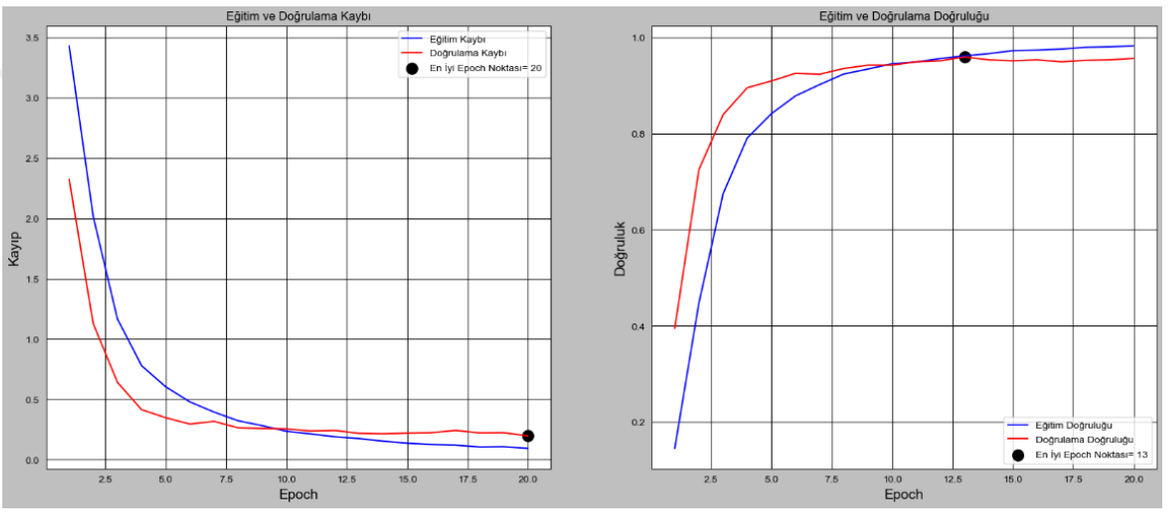
Hiperparametre	En İyi Değer
kernel_regularizer L1	2,674585427220742e-06
kernel_regularizer L2	0,00017909910243922317
bias_regularizer L2	5,2134657783780125e-06
activity_regularizer L2	8,02368684716831e-05
learning_rate	0,0001
optimizasyon algoritması	RMSprop

Tablo 77'ye göre aramalar sonucunda model için en iyi yoğun katman nöron sayısı 128 olarak bulunmuştur. L1 çekirdek düzenlileştiricisi, L2 düzenlileştiricisine göre daha iyi performans gösterdiğinden tercih edilmiştir.

Tablo 78. Optimize Edilmiş EfficientNetV2B1 Modeline Ait Analiz Sonuçları

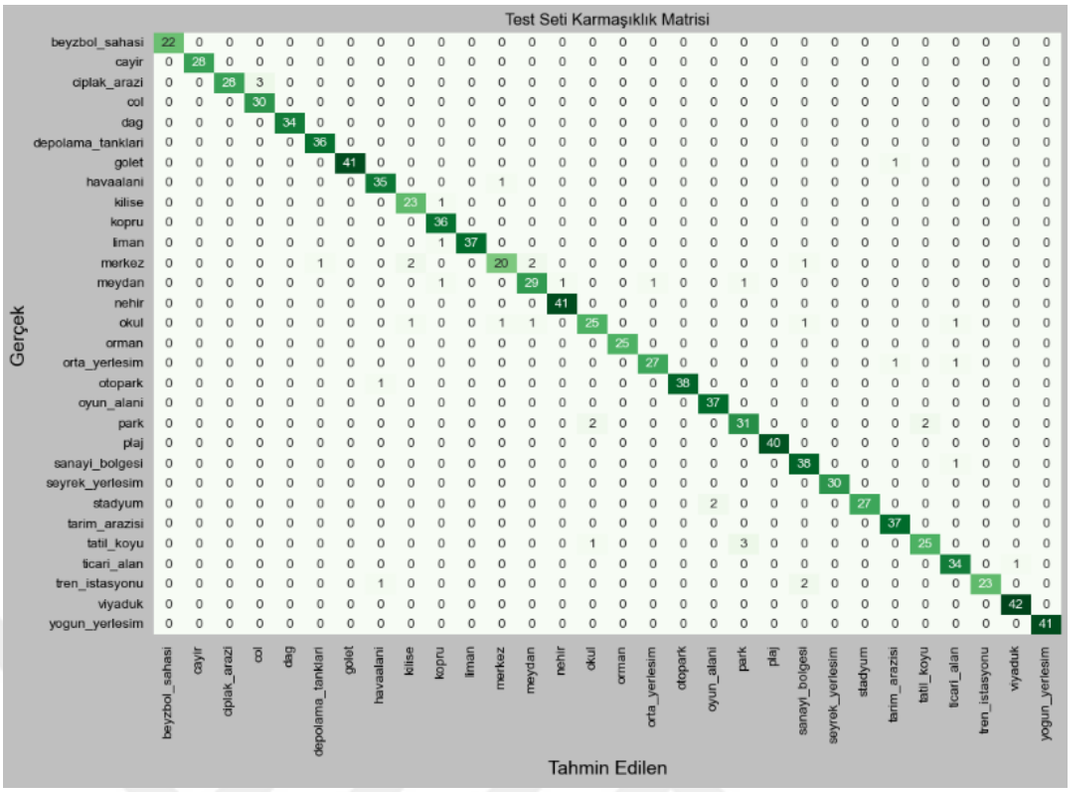
Transfer Öğrenmesi Modeli	Model Adı	Epoch	Yığın Boyutu	Eğitim-Doğrulama-Test	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1
EfficientNetv2B1	Temel2	20	16	8000-1000-1000	0,9480	0,9492	0,9432	0,9429
	Temel2	20	32	8000-1000-1000	0,9600	0,9598	0,9560	0,9570

Tablo 78'de görüldüğü üzere bu modelde en iyi sonucu yığın boyutu 32 olan model vermiştir. Optimize edilen modele dair kayıp doğruluk grafiği ve test seti karmaşıklık matrisi aşağıda gösterilmiştir.



Şekil 220. Optimize Edilmiş EfficientNetV2B1 Modeli Kayıp ve Doğruluk Grafiği

Şekil 220'de optimize edilmiş V2B1 modelinin kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde, eğitim kaybının 20 epoch sonunda 0,10 seviyesine, doğrulama kaybının ise 0,20'ye indiği görülmektedir. İki eğrinin açıklık farkı yaklaşık 0,1'dir. Her iki eğri arasındaki açıklığın son derece az olması, aşırı öğrenmenin olmadığı ve oldukça iyi bir genelleme performansını göstermektedir. Ayrıca model, oldukça düzgün eğrilerden oluşmaktadır. Sağ taraftaki doğruluk grafiğinde ise 20. epoch itibariyle eğitim doğruluğunun 0,98, doğrulama doğruluğunun ise 0,94 seviyesinde olduğu görülmektedir. Eğriler arasındaki 0,04'lük bir açıklık bulunmaktadır. Modele ait sınıflandırma raporu, ek 34'te verilmiştir.



Şekil 221. Optimize Edilmiş EfficientNetV2B1 Modeli Karmaşıklık Matrisi

Şekil 221'de ise modele ait test seti karmaşıklık matrisi gösterilmektedir. İlk model ile karşılaştırıldığında oldukça yüksek bir doğruluk oranına sahip olan matriste yanlış sınıflama oranının da çok azaldığı görülebilir. Optimize edilmiş modelde, modelin daha önceden hiç görmediği 1000 test görüntüsünden sadece 40 tanesi hatalı olarak sınıflandırılmıştır. Bu oran ilk modelde 148 idi. Matriste, çıplak arazi sınıfından 3 örneğin çöl olarak sınıflandırıldığı göze çarpmaktadır.

Yedinci analiz temel model V2B2 üzerinde uygulanmıştır. EfficientNetV2B2 temel modeli için aramalar sonucunda elde edilen en iyi değerler Tablo 79'da gösterilmektedir.

Tablo 79. EfficientNetV2B2 Modeline Ait En İyi Hiperparametreler ve Değerleri

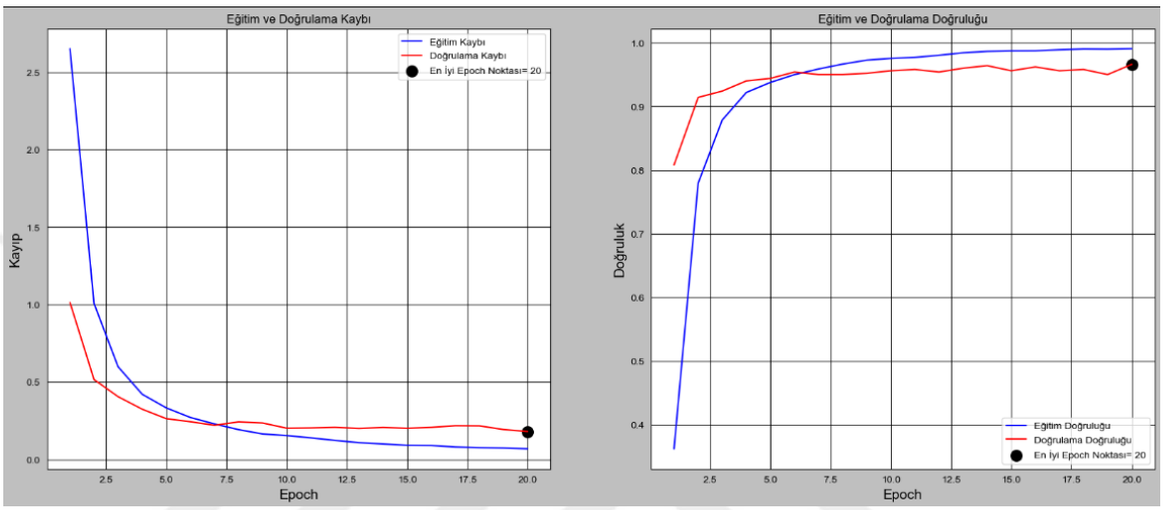
Hiperparametre	En İyi Değer
kernel_regularizer L1	1,4682754255180836e-06
kernel_regularizer L2	3,3675252019936284e-05
bias_regularizer L2	1,6393450100072133e-06
activity_regularizer L2	0,0005824802393366627
learning_rate	0,0001
optimizasyon algoritması	RMSprop

Aramalar sonucunda model için en iyi yoğun katman nöron sayısı 512 olarak bulunmuştur. L1 çekirdek düzenlileştiricisi, L2 düzenlileştiricisine göre daha iyi performans gösterdiğinden tercih edilmiştir.

Tablo 80. Optimize Edilmiş EfficientNetV2B2 Modeline Ait Analiz Sonuçları

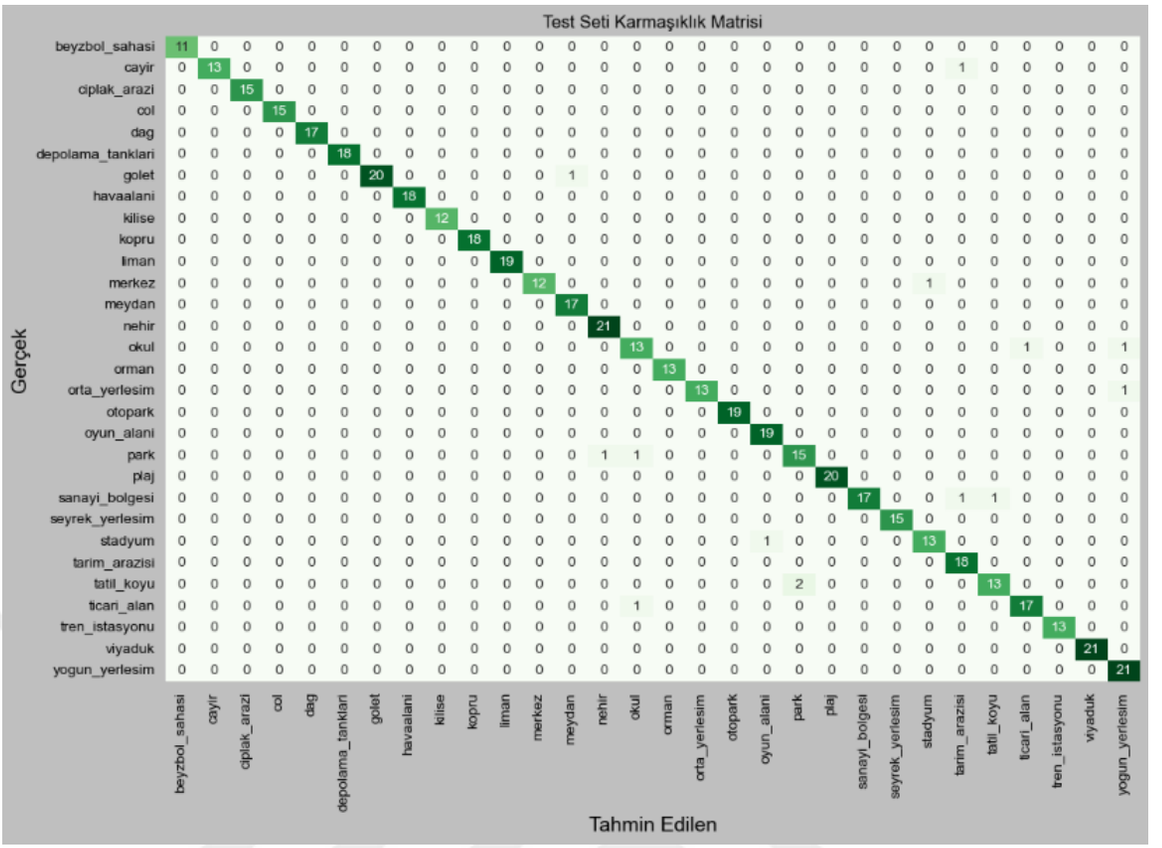
Transfer Öğrenmesi Modeli	Model Adı	Epoch	Yığın Boyutu	Eğitim-Doğrulama-Test	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1
EfficientNetB2v2	Temel1	20	16	9000-500-500	0,9720	0,9738	0,9705	0,9716
EfficientNetB2v2	Temel1	20	32	9000-500-500	0,9620	0,9639	0,9586	0,9589

Tablo 80'de görüldüğü üzere bu modelde en iyi sonucu yığın boyutu 16 olan model vermiştir. Optimize edilen modele dair kayıp doğruluk grafiği ve test seti karmaşıklık matrisi aşağıda gösterilmiştir.



Şekil 222. Optimize Edilmiş EfficientNetV2B2 Modeli Kayıp ve Doğruluk Grafiği

Şekil 222'de optimize edilmiş V2B2 modelinin kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde, eğitim kaybının 20 epoch sonunda 0,10 seviyesine, doğrulama kaybının ise 0,20'ye indiği görülmektedir. İki eğrinin açıklık farkı yaklaşık 0,1'dir. Her iki eğri arasındaki açıklığın son derece az olması, aşırı öğrenmenin olmadığı ve oldukça iyi bir genelleme performansını göstermektedir. Sağ taraftaki doğruluk grafiğinde ise 20. epoch itibariyle eğitim doğruluğunun 0,99, doğrulama doğruluğunun ise 0,96 seviyesinde olduğu görülmektedir. Doğrulama doğruluğunun 12. epochtan itibaren hafif bir dalgalanma yaptığı görülmektedir. Eğriler arasındaki 0,03'lük bir açıklık bulunmaktadır. Modele ait sınıflandırma raporu, ek 35'te verilmiştir.



Şekil 223. Optimize Edilmiş EfficientNetV2B2 Modeli Karmaşıklık Matrisi

Şekil 223'te, modele ait test seti karmaşıklık matrisi gösterilmektedir. İlk model ile karşılaştırıldığında oldukça yüksek bir doğruluk oranına sahip olan matriste yanlış sınıflama oranının çok azaldığı görülebilir. Optimize edilmiş modelde, modelin daha önceden hiç görmediği 500 test görüntüsünden sadece 14 tanesi hatalı olarak sınıflandırılmıştır. Bu oran ilk modelde 71 idi. Matriste, tatil köyü sınıfından 2 örneğin park olarak sınıflandırıldığı göze çarpmaktadır.

Sekizinci ve son analiz temel model V2B3 üzerinde uygulanmıştır. EfficientNetV2B3 temel modeli için aramalar sonucunda elde edilen en iyi değerler Tablo 81'de gösterilmektedir.

Tablo 81. EfficientNetV2B3 Modeline Ait En İyi Hiperparametreler ve Değerleri

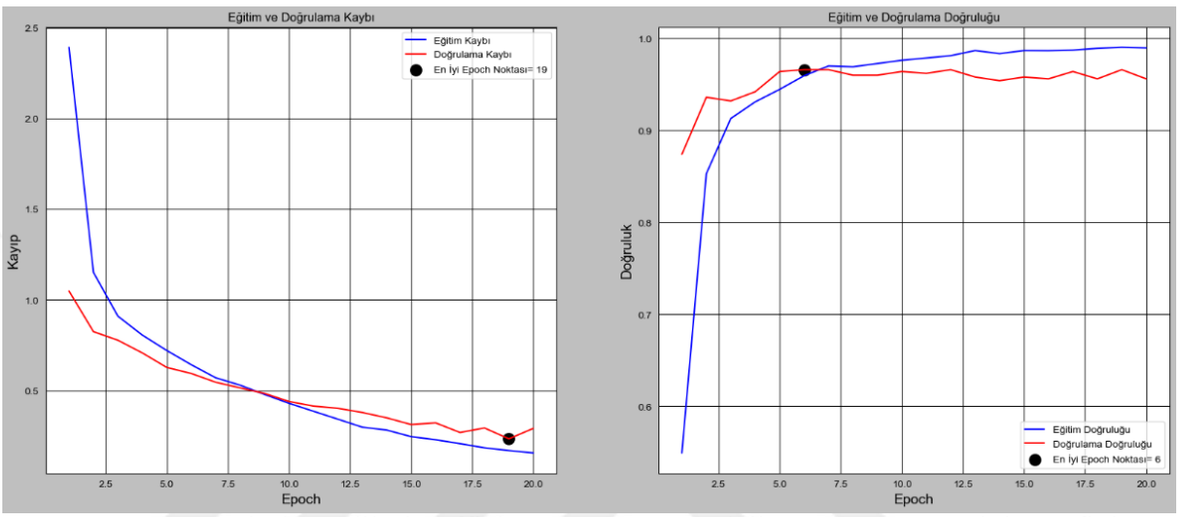
Hiperparametre	**En İyi Değer**
kernel_regularizer L1	3,88772957273848e-06
kernel_regularizer L2	0,0007554696804105081
bias_regularizer L2	3,21603893942531e-05
activity_regularizer L2	0,00016397618885493213
learning_rate	0,001
optimizasyon algoritması	Adamax

Aramalar sonucunda model için en iyi yoğun katman nöron sayısı 256 olarak bulunmuştur. L1 çekirdek düzenlileştiricisi, L2 düzenlileştiricisine göre daha iyi performans gösterdiğinden tercih edilmiştir.

Tablo 82. Optimize Edilmiş EfficientNetV2B3 Modeline Ait Analiz Sonuçları

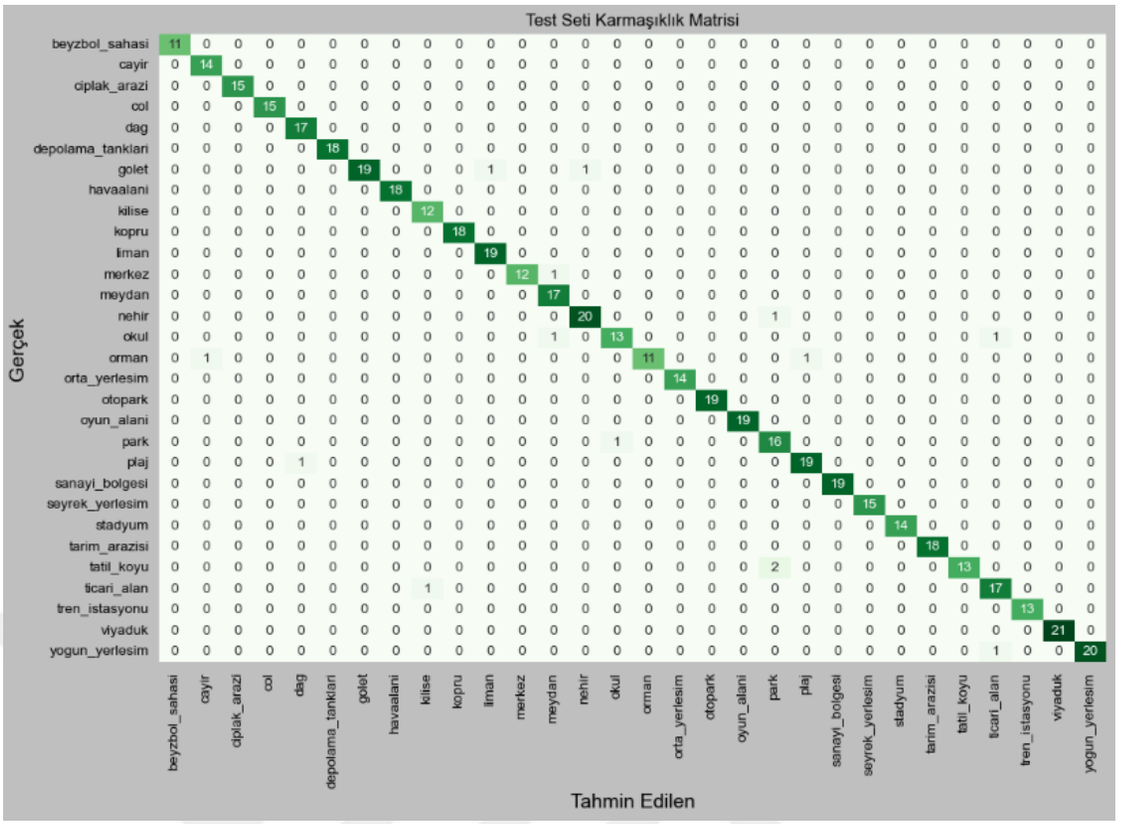
Transfer Öğrenmesi Modeli	Model Adı	Epoch	Yığın Boyutu	Eğitim-Doğrulama-Test	Doğruluk (Accuracy)	Kesinlik (Precision)	Duyarlılık (Recall)	F1
EfficientNetB3v2	Temel1	20	16	9000-500-500	0,9680	0,9684	0,9657	0,9652
	Temel1	20	32	9000-500-500	0,9720	0,9738	0,9716	0,9718

Tablo 82'de görüldüğü üzere bu modelde en iyi sonucu yığın boyutu 32 olan model vermiştir. Optimize edilen modele dair kayıp doğruluk grafiği ve test karmaşıklık matrisi aşağıda gösterilmiştir.



Şekil 224. Optimize Edilmiş EfficientNetV2B3 Modeli Kayıp ve Doğruluk Grafiği

Şekil 224'te optimize edilmiş V2B3 modelinin kayıp ve doğruluk grafikleri gösterilmektedir. Sol tarafta bulunan kayıp grafiğinde, eğitim kaybının 20 epoch sonunda 0,15 seviyesine, doğrulama kaybının ise 0,25'e indiği görülmektedir. İki eğrinin açıklık farkı yaklaşık 0,1'dir. Her iki eğri arasındaki açıklığın son derece az olması, aşırı öğrenmenin olmadığı ve oldukça iyi bir genelleme performansını göstermektedir. Ancak, doğrulama kaybının son epochta hafif bir yükselişi gözlemlenmiştir. Sağ taraftaki doğruluk grafiğinde ise 20. epoch itibariyle eğitim doğruluğunun 0,99, doğrulama doğruluğunun ise 0,96 seviyesinde olduğu görülmektedir. Eğriler arasındaki 0,03'lük bir açıklık bulunmaktadır. Son epoch itibariyle doğrulama doğruluğunun hafifçe düşme eğiliminde olduğu gözlemlenmiştir. Modele ait sınıflandırma raporu, ek 36'da verilmiştir.



Şekil 225. Optimize Edilmiş EfficientNetV2B3 Modeli Karmaşıklık Matrisi

Şekil 225'te modele ait test seti karmaşıklık matrisi gösterilmektedir. İlk model ile karşılaştırıldığında oldukça yüksek bir doğruluk oranına sahip olan matriste yanlış sınıflama oranının çok azaldığı görülebilir. Optimize edilmiş modelde, modelin daha önceden hiç görmediği 500 test görüntüsünden sadece 14 tanesi hatalı olarak sınıflandırılmıştır. Bu oran ilk modelde 67 idi. Matriste tatil köyü sınıfının park olarak sınıflandırıldığı 2 örnek göze çarpmaktadır.

Tablo 83. Transfer Öğrenmesi Genel Değerlendirme Tablosu

Optimize Edilmemiş ve Düzenlileştirilmemiş EfficientNet Modelleri								
	B0	**B1**	**B2**	**B3**	**V2B0**	**V2B1**	**V2B2**	**V2B3**
Parametre Sayısı	5.330.571	7.856.239	9.177.569	12.320.535	7.200.312	8.212.124	10.178.374	14.467.622
Bölümlendirme	9000-500-500	8000-1000-1000	9000-500-500	9000-500-500	9000-500-500	8000-1000-1000	9000-500-500	9000-500-500
Doğruluk	0,7960	0,7990	0,8220	0,8580	0,8740	0,8520	0,8580	0,8660
Kesinlik	0,8070	0,8146	0,8365	0,8707	0,8840	0,8637	0,8615	0,8718
Duyarlılık	0,7936	0,7977	0,8180	0,8512	0,8653	0,8524	0,8511	0,8589
F1	0,7867	0,7924	0,8168	0,8492	0,8685	0,8509	0,8508	0,8588
Doğrulama Kaybı	0,6458	0,7398	0,6548	0,4772	0,3968	0,5732	0,4357	0,4355
Optimize Edilmiş ve Düzenlileştirilmiş EfficientNet Modelleri								
	B0	**B1**	**B2**	**B3**	**V2B0**	**V2B1**	**V2B2**	**V2B3**
Parametre Sayısı	5.330.571	7.856.239	9.177.569	12.320.535	7.200.312	8.212.124	10.178.374	14.467.622
Bölümlendirme	9000-500-500	8000-1000-1000	9000-500-500	9000-500-500	9000-500-500	8000-1000-1000	9000-500-500	9000-500-500
Doğruluk	0,9640	0,9640	0,9580	**0,9720**	0,9320	0,9600	**0,9720**	**0,9720**
Kesinlik	0,9651	0,9633	0,9600	0,9739	0,9347	0,9598	0,9738	0,9738
Duyarlılık	0,9600	0,9618	0,9551	0,9704	0,9271	0,9560	0,9705	0,9716
F1	0,9608	0,9622	0,9532	0,9713	0,9289	0,9570	0,9716	0,9718
Doğrulama Kaybı	0,2571	0,2019	0,3480	0,1879	0,2872	0,1971	**0,1803**	0,2921

Tablo 83'te, veri seti 2'ye ait EfficientNet ve EfficientNetV2 model varyantlarının (B0-B3; v2B0-v2B3) sonuçları karşılaştırılmıştır. İlk aşama analizlerde tespit edilen en iyi modeller çeşitli düzenlileştirme ve optimizasyon teknikleriyle birlikte eğitilmiş ve doğruluk, kesinlik, duyarlılık ve F1 skorları ile doğrulama kaybı metrikleri bazında değerlendirilmiştir. Her bir modeldeki parametre sayısı, modelin derinliğine ve karmaşıklığına bağlı olarak 5.330.571 (B0) ile 14.467.622 (v2B3) arasında değişmektedir. Veri seti 2'nin optimize edilmemiş versiyonunda B3, yüksek bir doğrulama kaybına (0,4772) sahip olmasına rağmen, 0,8580'lik doğruluk skoruna erişerek diğer temel modellerden daha iyi performans göstermiştir. EfficientNetV2 modelleri arasında ise v2B0, 0,8740'lık doğruluk oranına erişmiştir. Bu modelin kesinlik (0,8840), duyarlılık (0,8653) ve F1 (0,8685) skorları da diğer V2 modellerine göre daha başarılı sonuç vermiştir. Doğrulama kaybı (0,3968) açısından da en başarılı V2 modelidir. Optimize edilmiş ve düzenlileştirilmiş modellerin performansına bakıldığında ise oldukça yüksek seviyede iyileştirmelerin olduğu görülebilir. En başarılı sonuç, 0,9720'lik doğruluk skoru ile B3 tarafından sağlanmıştır. Bu modelin kesinlik (0,9739), duyarlılık (0,9704) ve F1 (0,9713) skorları da diğer temel modellere göre daha başarılı olduğu görülmektedir. Modelin doğrulama kaybı da 0,1879 olarak en düşük seviyede gerçekleşmiştir. v2B2 ve v2B3 modelleri, 0,9720'lik doğrulama skorları ile oldukça başarılı diğer modeller olarak göze çarpmaktadır. Bu modellerin kesinlik, duyarlılık ve F1 skorları da birbirine yakın seviyede gerçekleşmiştir. Doğrulama kayıpları da her iki modelde de sırasıyla 0,1803 ve 0,2921'dir. Başka bir deyişle, optimize edilen ve düzenlileştirme uygulanan modeller, yüksek seviyede genelleştirilmiştir. Genel olarak, optimize edilmiş V2 modellerinin ilk oluşturulan temel modellere göre daha iyi performans gösterdiği görülmektedir. V2B3 modeli en yüksek parametre sayısına (14.467.622) sahip olmasına rağmen, doğrulama kaybı (0,2921) diğer geliştirilmiş modellere karşı yüksek kalmıştır. En başarılı modellerin çoğunlukla %90-%5-%5 eğitim-doğrulama-test seti olarak ayrılanlarda olduğu görülmektedir.

ALTINCI BÖLÜM

6. PROVENANS VERİ HATTI VE ARCHIVEMATICA

Bu bölümde, öncelikle tüm analiz süreçlerini kapsayan bir provenans veri hattının[250] oluşturulma aşamaları açıklanmıştır. Ardından Archivematica programının arşiv bilgi paketleri ve transfer süreci açıklanmış ve süreç bilgileri detaylı olarak gösterilmiştir. Son olarak genel akış verileri, PROV-ML tabanlı çizilmiştir.

6.1. Provenans Veri Hattı

Archivematica uygulamasına geçmeden önce, tez kapsamında elde edilen tüm sonuçların JSON formatında provenans veri hattı oluşturulmuştur. Provenans veri hattı; sonuçların hangi hiperparametrelerle üretildiğini, analizler için hangi veri setlerinin girdi olarak kullanıldığı gibi çok çeşitli süreç bilgilerini içermektedir. Tezde provenans veri hattı; hesaplamalı arşiv bilimi bağlamında, "çıktıları sağlayan sürecin kendisi" olarak değerlendirilmiştir. Başka bir deyişle, "sürecin kendisi"; tüm ön işlemleri, tüm analizleri, analizlere dair tüm sonuçları, veri setlerini, hiperparametreleri, hataları, sistem uyarılarını, yazılım ve donanım bileşenleri ile tüm hesaplamalı süreçleri kapsamaktadır. Tüm süreç verileri, sırasıyla oluşturulduktan sonra BagIt kütüphanesi ile paketlenmiş ve Archivematica programına yüklenmiştir.

Makine öğrenmesi ve transfer öğrenmesine ait 2 örnek provenans veri hattı, Python programlama dilinde oluşturulmuş ve ağaç yapısı şeklinde zaman damgalı olarak gösterilmiştir.

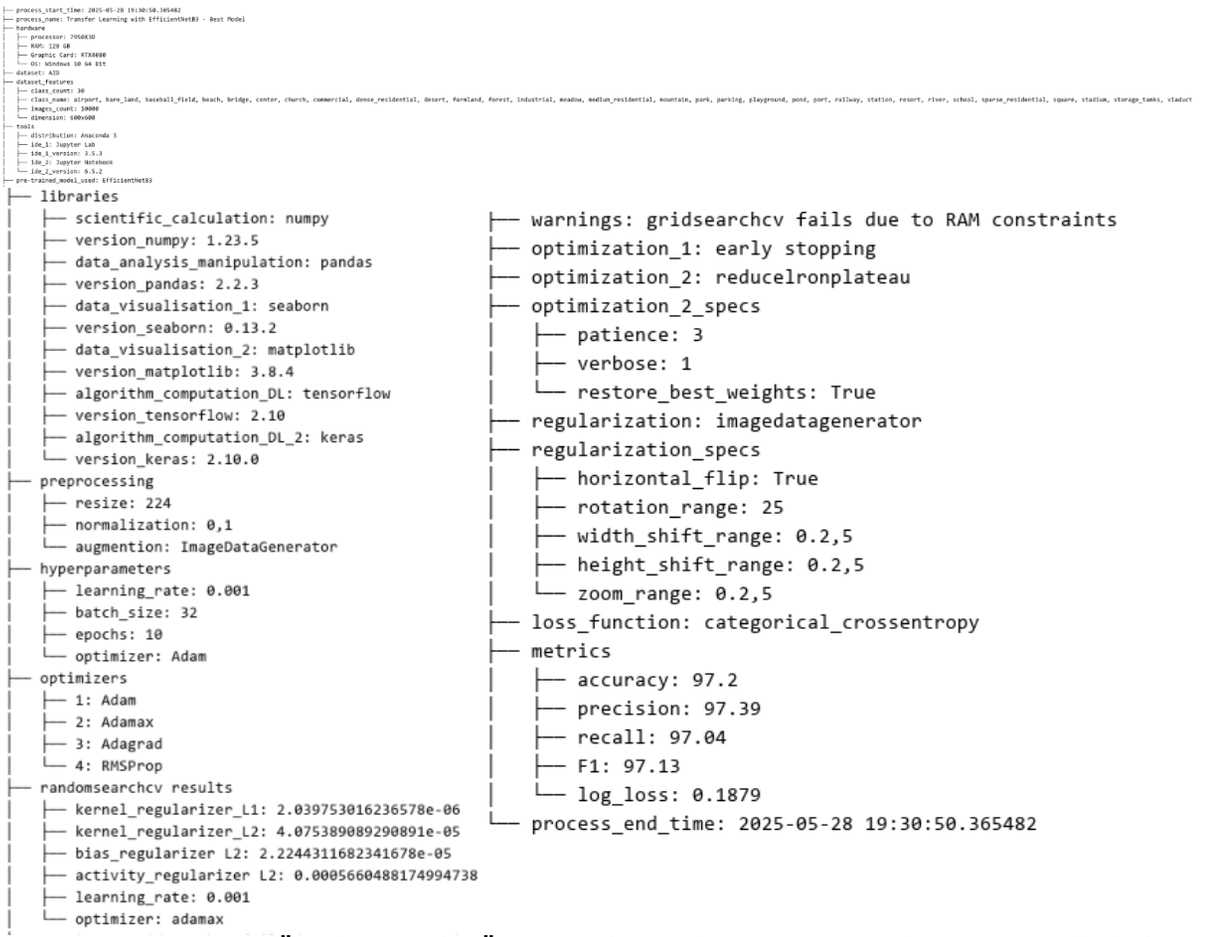
```
├── process_start_time: 2025-05-29 09:46:03.659513
├── process_name: Number of Passengers per Flight
├── hardware
│   ├── processor: 7950X3D
│   ├── RAM: 128 GB
│   ├── Graphic Card: RTX4080
│   └── OS: Windows 10 64 Bit
├── dataset: US Domestic Flights from 1990 to 2009
├── dataset URL: https://academictorrents.com/details/a2ccf94bbb4af222bf8e69dad60a68a29f310d9a
├── dataset_features
│   ├── rows: 3606803
│   ├── columns: 11
│   └── columns_names: Origin, Destination, Origin City, Destination City, Passengers, Seats, Flights, Distance, Fly Date, Origin Population, Destination Population
├── tools
│   ├── distribution: Anaconda 3
│   ├── ide_1: Jupyter Lab
│   ├── ide_1_version: 3.5.3
│   ├── ide_2: Jupyter Notebook
│   └── ide_2_version: 6.5.2
├── models_used: PCA+Random Forest, PCA+Decision Tree, PCA+KNN, PCA+Naive Bayes, PCA+MLP
├── libraries
│   ├── scientific_calculation: numpy
│   ├── version_numpy: 1.23.5
│   ├── data_analysis_manipulation: pandas
│   ├── version_pandas: 1.5.3
│   ├── data_visualisation_1: seaborn
│   ├── version_seaborn: 0.12.2
│   ├── data_visualisation_2: matplotlib
│   ├── version_matplotlib: 3.7.0
│   ├── algorithm_computation_DL: sklearn
│   └── version_tensorflow: 1.2.1
```

[250] Provenance Pipeline



Şekil 226. Örnek Makine Öğrenmesi Süreci Provenans Veri Hattı Gösterimleri

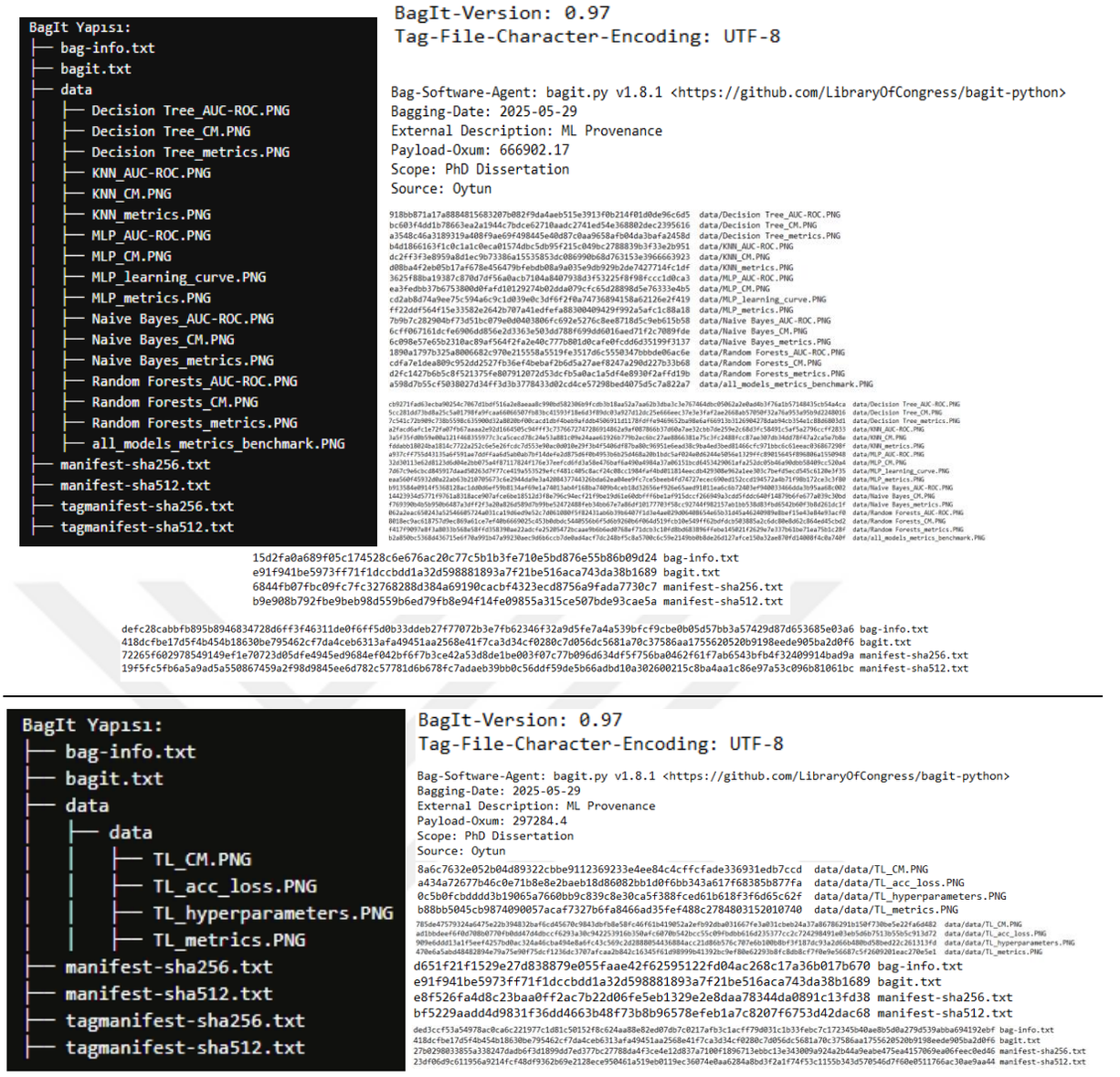
Şekil 226’da makine öğrenmesi analizlerinde sınıflandırma görevinde kullanılan 5 sınıflı “uçuş başına yolcu sayısı” özelliğine ait tüm analiz süreçlerini içeren örnek bir provenans veri hattı gösterilmektedir. Bu veri hattında; veri seti özellikleri, ön işleme adımları, kullanılan kütüphaneler ve versiyonları, model eğitiminde kullanılan algoritmalar vb. daha birçok metaveri, değerleri ve sonuçları ile birlikte verilmektedir.



Şekil 227. Örnek Transfer Öğrenmesi Süreci Provenans Veri Hattı Gösterimleri

Şekil 227’de ise, transfer öğrenmesi analizlerinden EfficientnetB3 modeli için oluşturulan örnek provenans veri hattı görülmektedir. Bu veri hattında da en baştan tüm analiz süreçlerinin metaveri bilgileri ile değerleri ve sonuçları gösterilmektedir.

Makine öğrenmesi örnek süreci ile transfer öğrenmesi örnek sürecine ait çıktılar (grafik ve tablolar) BagIt kütüphanesi ile paketlenmiştir. Paketleme sürecine dair çıktılar görselleştirilmiştir.



Şekil 228. Her İki Örnek Sürece Ait BagIt Yapısı ve Metaverileri

Oluşturulan klasör yapısında bulunan dosyalar ve bilgileri aşağıda özetlenmiştir:

- ***bag-info.txt:*** Kaynak organizasyonu, yazar ve paketin açıklaması gibi üst düzey tanımlayıcı metaverileri içermektedir.
- ***bagit.txt:*** BagIt spesifikasyonuna ait sürüm ve kod bilgisini içermektedir.
- ***data:*** Dijital nesneleri içermektedir.
- ***manifest-sha256.txt ve manifest-sha512.txt:*** Bu dosyalar, her bir dijital materyalin kriptografik özet değerlerini[251] (SHA-256, SHA-512) içermektedir.

[251] Checksum

- ***tagmanifest-sha256.txt ve tagmanifest-sha512.txt:*** Etiket dosyalarının (bag-info.txt, bagit.txt, manifest-sha256.txt, manifest-sha512.txt) özet değerlerini içermekte ve metaveriler ile doğrulama mekanizmalarının tamlığını sağlamaktadır.

Bu işlemlerden sonra; orijinal veri setleri, türev veri setleri, oluşturulan provenans veri hatları ve grafikler, Archivematica programına RAR formatında yüklenmiştir. Transferler ön işlemlerden geçirilerek öncelikle AIP, ardından SIP ve alım[252] adımından sonra gerekli işlemler yapıldığında DIP oluşmaktadır.

6.2. Archivematica

Archivematica, her türlü dijital materyal içeren koleksiyonların uzun süreli erişimlerinin sağlanabilmesi amacıyla 2008'de Artefactual Systems tarafından geliştirilen, Açık Arşiv Bilgi Sistemi Standardı (OAIS), METS, PREMIS ve BagIt ile uyumlu, ücretsiz ve açık kaynak kodlu bir dijital koruma ekosistemidir (Bogosavljevic, 2018). Archivematica projesinin amacı; sınırlı teknik ve finansal kapasiteye sahip arşivler ile kütüphanelere dijital materyallerin korunmasını sağlayabilmek için araç ve metodoloji sunmaktır. Program sadece Ubuntu ve Linux tabanlı sistemlerde çalışmaktadır. Windows desteği ise henüz bulunmamaktadır. Tüm işlemlerin veri hattı[253] tabanlı bir mikro hizmet modeli ile gerçekleştirildiği sistemde, bir dijital materyalin tam anlamıyla korunabilmesi için 4 adımlı bir süreç bulunmaktadır:

- ***Transfer:*** İlk işlem, dijital materyalin "Transfer" dizinine yerleştirildiğinde başlatılır. Burada içerik, standart bir klasör yapısı kullanılarak paketlenir. Bu aşamada sistem; dosya türlerini belirler, virüs taraması yapar, bütünlük doğrulaması için özet değerleri üretir ve metaveri çıkarımı yapar.
- ***Alım***: Transfer onaylanıp işlendikten sonra içerik, korumaya uygun formatlar bazında normalizasyona (word belgelerinin PDF/A veya TIFF'e dönüştürülmesi gibi) tabi tutulur. Bu aşamada program Arşiv Bilgi Paketi (AIP) oluşturur; orijinal ve normalizasyonlu dosyalara, PREMIS bazlı teknik metaveriler de eklenir ve METS, Dublin Core vb. kullanılarak açıklayıcı metaveriler yapılandırılır. AIP, daha sonra uzun vadeli depolarda saklanır.
- ***Depolama:*** Program, çeşitli harici depolama sistemlerine (DSpace, Fedora, LOCKSS vb.) entegre olabilir. AIP'ler ise metaverileri ve log kayıtlarıyla birlikte saklanır. Dosyalar BagIt kullanılarak paketlenir, özet değerleri ile veri bütünlüğü sağlanır.

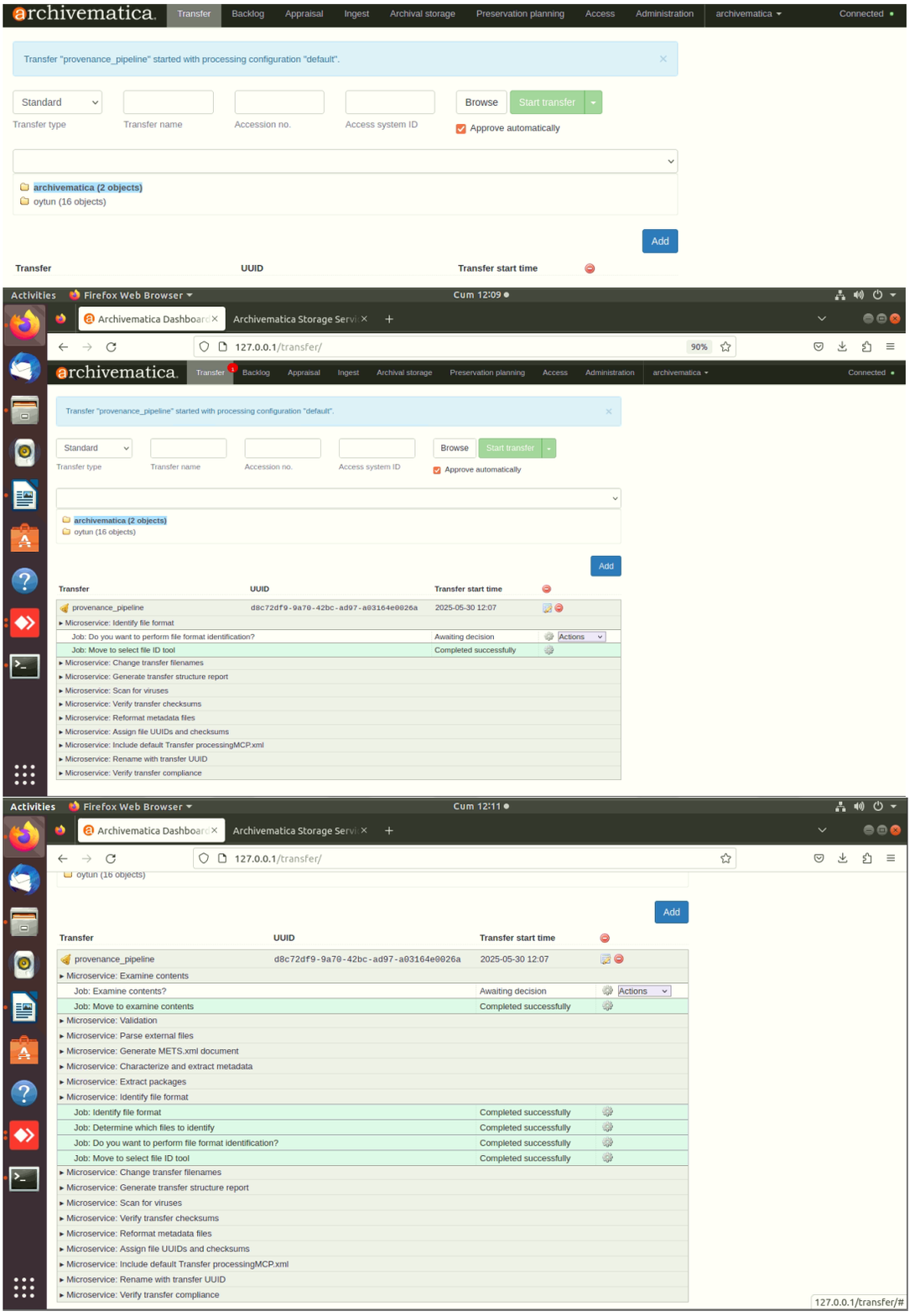
[252] Ingest
[253] Pipeline

- ***Erişim:*** Bu aşamada erişimin sağlanabilmesi amacıyla Dağıtım Bilgi Paketi (DIP) üretilir. DIP, videoların veya görüntülerin kopyaları gibi hafif sürümleri içerir.

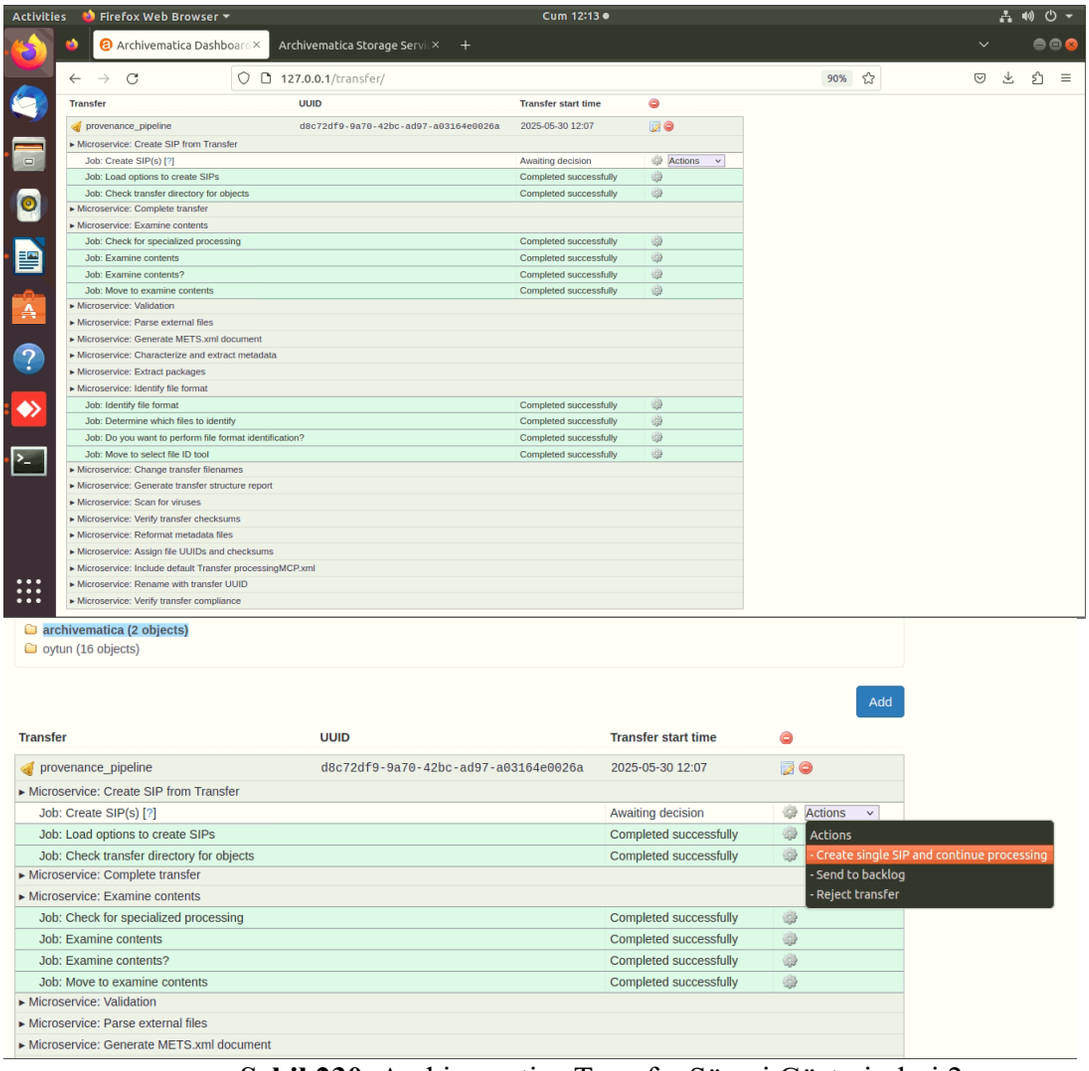
Program bağlamında bir gönderim bilgi paketi (SIP[254]), dijital arşive veya sisteme alınan dijital materyaller ve ilişkili metaverilerdir. SIP, üretici tarafından oluşturulabilir veya üreticinin sağladığı içerik temelli olarak sistem yöneticileri tarafından oluşturulabilir. Arşiv bilgi paketi (AIP[255]) ise dijital bir depoda uzun süreli saklanan ve korunan dijital materyalin sürümü olarak tanımlanabilir. Paket içeriğinde saklanacak dosyalar, açıklayıcı metaveriler, saklama metaverileri, yapısal metaveriler ile tüm süreci ilişkilendiren METS dosyası bulunmaktadır. Dağıtım bilgi paketi (DIP[256]) ise erişim veya dağıtım için hazırlanan dijital materyalin sürümü olup uzun süreli saklama için AIP'ten türetilmiştir. Archivematica, sürecin işlenmesi sırasında AIP ile birlikte bir DIP oluşturur. Bir DIP; dijital materyallerin erişim kopyaları ile tanımlayıcı metaverilerini içerebilir (Navale ve McAuliffe, 2018; Bogosavljevic, 2018; Archivematica, 2025).

Oluşturulan paketler, Ubuntu işletim sistemi içerisinde kurulu bulunan Archivematica programının "Transfer" bölümüne standart tipte yüklenmiştir. Yüklemeler "Transfer_ML_TL" klasörü içinde bulunan "Provenance_Pipeline_ML.rar" ve "Provenance_Pipeline_TL.rar" dosyalarını kapsamaktadır.

[254] Submission Information Package
[255] Archival Information Package
[256] Dissemination Information Package

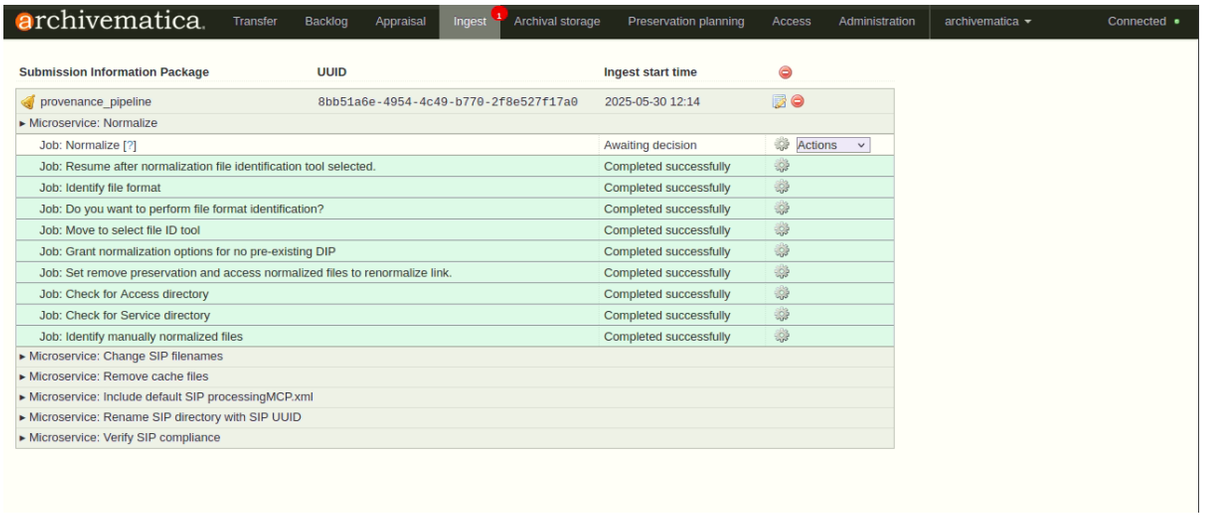


Şekil 229. Archivematica Transfer Süreci Gösterimleri 1



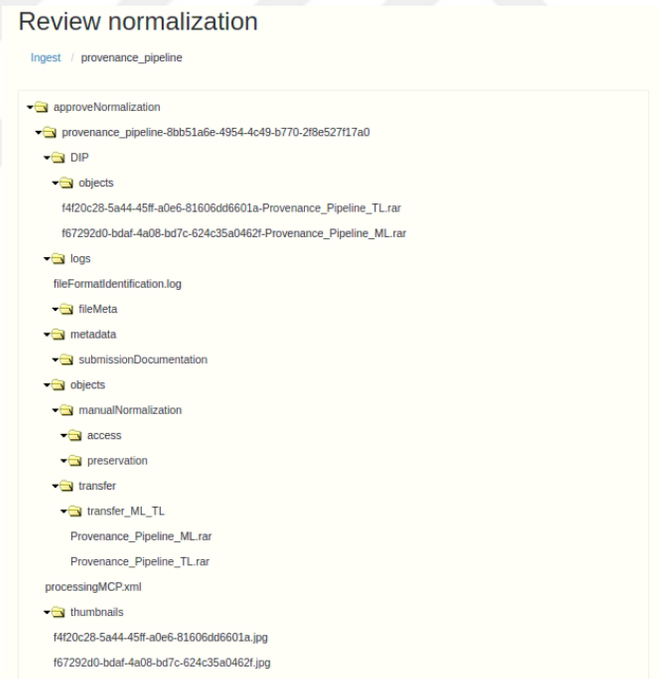
Şekil 230. Archivematica Transfer Süreci Gösterimleri 2

Program, transfer işlemleri sırasında mikro hizmetlere dayalı bir iş akışı kullanmaktadır. Örneğin; dijital nesnelerin içerikleri ile ilgili olarak transfer uyumluluğunun tespit edilmesi için bir mikro hizmet başlatılmıştır. Ardından içerik formatlarının tanımlanması işlemi gerçekleştirilmiştir. Bunun yanında, tüm içerik virüs taramasından geçirilmiştir. Süreç ve içeriklere ait bir METS.xml dosyası da oluşturulmuştur. Son olarak, SIP oluşturma işlemi tamamlandığında, alım aşamasına geçilmiştir.



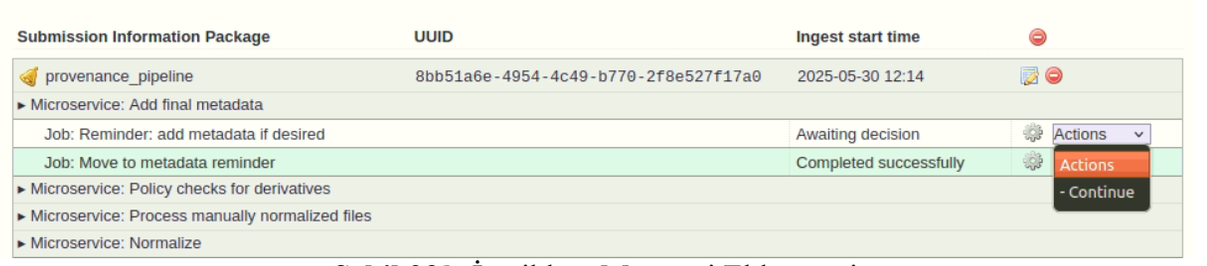
Şekil 231. Archivematica Alım (Ingest) Aşaması

Alım aşaması, transfer edilen ve doğrulanan dosyaların uzun süreli korunması için SIP'e dönüştürüldüğü aşamadır. Örneğin; normalizasyon adımı, dijital materyalleri programın içinde korumaya uygun biçimlere (örneğin TIFF, PDF/A, TXT, WAV) dönüştürme işlemidir. Bu bağlamda, içeriklerin uzantıları dönüştürülmüştür.



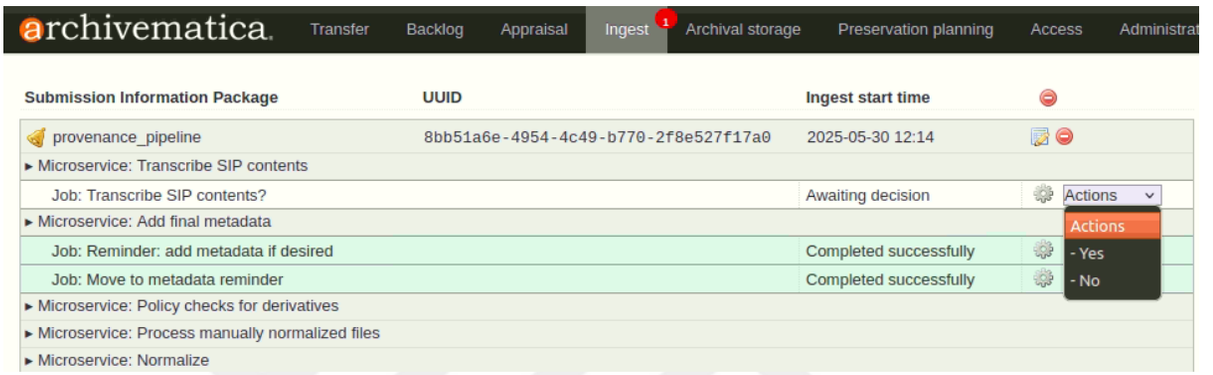
Şekil 232. İçeriklerin Normalizasyonu

Şekil 232'de görüldüğü üzere normalizasyona uğrayacak içeriklerin ağaç yapısı formatındaki görüntüsü de elde edilebilmektedir.



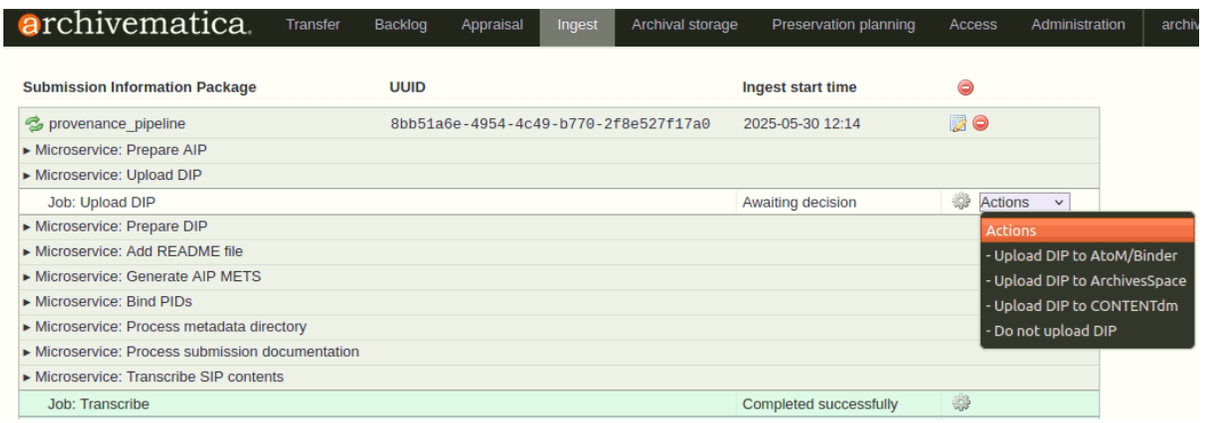
Şekil 233. İçeriklere Metaveri Eklenmesi

Şekil 233'te, SIP oluşturulmadan önce son metaverilerin eklenmesi işlemi için akışın onay beklediği görülebilir.



Şekil 234. SIP Kayıtları

Şekil 234'te ise SIP içeriklerinin kaydedilmesi için akışın onay beklediği görülmektedir. SIP; dizin ve dosya yapısı, dosya adları ve yolları, dosya UUID[257]'leri ve özet değerleri ile PREMIS metaverilerini içermektedir. Bu işlem, alım sırasında SIP içerisindeki dosyaların neler olduğunu belgeler.

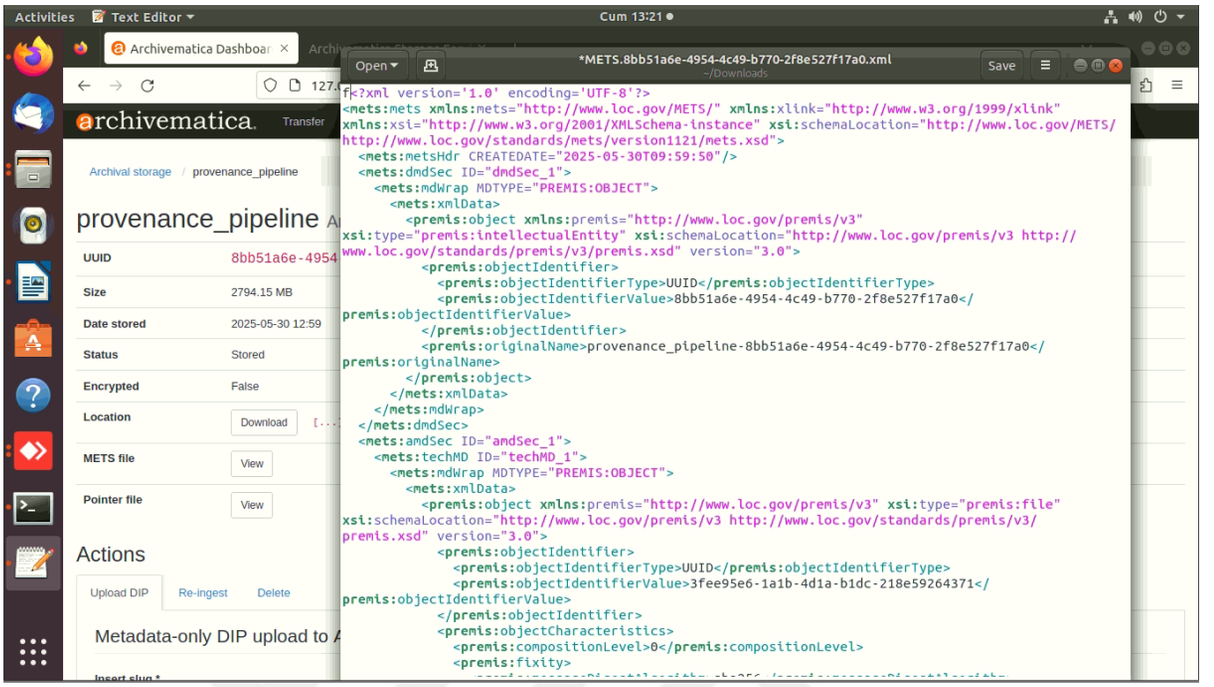


Şekil 235. DIP Yükleme Adımı

Şekil 235'te ise alım sürecinin son aşaması görülmektedir. Program, DIP oluşturulduktan sonra erişim ve kullanım için harici erişim sistemlerine (Atom/Binder ya da ArchivesSpace vb.)

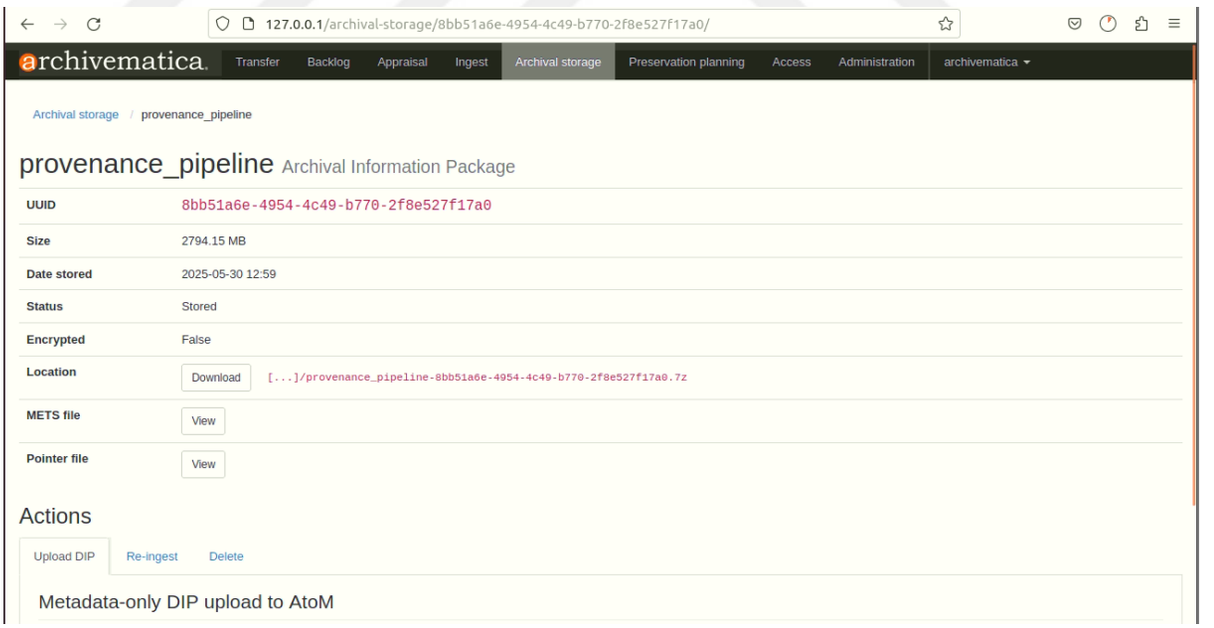
[257] Universally Unique Identifier (Evrensel Tekil Tanımlayıcı). Archivematica programındaki iş akışlarının güvenilir bir şekilde izlenebilmesi ve yönetilmesi amacıyla farklı dijital materyallere, paketlere ve işlemlere atanan tanımlayıcı.

aktarılmasına olanak tanımaktadır. Bu aşamada seçim, DIP'in yüklenmemesi ve sadece Archivematica içinde kalmasıdır.



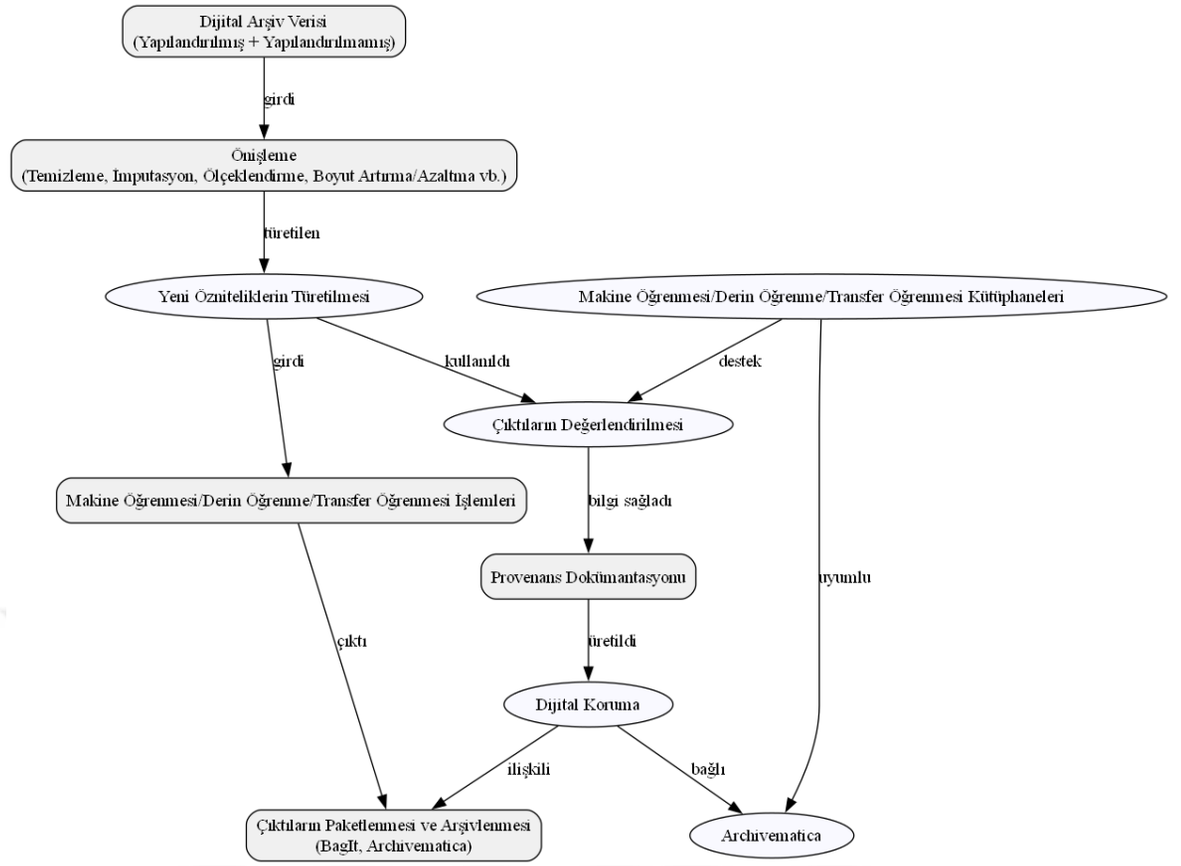
Şekil 236. AIP METS Dosyası

AIP paketinin oluşturma sürecine dair veriler, METS formatında da oluşturulmuştur.



Şekil 237. Tamamlanmış AIP Süreci

Şekil 237'da ise AIP bilgileri görülmektedir. SIP ise geçici oluşturulan bir paket olup, alım aşamasında kullanılmakta ve AIP oluşturulduktan sonra silinmektedir.



Şekil 238. Hesaplamalı Arşiv Süreçleri Genel Provenans Şeması

Şekil 238'de, bir arşivsel problemin hesaplamalı arşiv bilimi ile çözüm adımlarını PROV-ML[258] tabanlı özetleyen ve aynı zamanda bu sürecin provenansını gösteren şema bulunmaktadır. Diyagram, dijital arşiv verisinin girdi olarak kabul edilmesiyle başlamaktadır. Tezde, hem yapılandırılmış veri seti (TSV uzantılı uçuş bilgileri veri seti) hem de yapılandırılmamış veri seti (AID uydu görüntüleri veri seti) kullanılmıştır. Önişleme düğümünde, her iki veri setinde de çeşitli veri önişleme adımları (veri temizleme, eksik değerleri doldurma [imputasyon] vb.) yapılmıştır. Önişleme adımından sonra, öznitelik mühendisliği ile orijinal veri setinden yeni öznitelikler türetilmiştir. Tüm bu işlemlerden sonra ise makine öğrenmesi, derin öğrenme ve transfer öğrenmesi model ve algoritmaları ile hesaplamalar yapılmıştır. Hesaplamaların ardından tüm çıktılar detaylı bir şekilde değerlendirilmiş ve provenans için girdi sağlamıştır. Provenans veri hattı, analizlerin en baştan tüm süreç bilgilerini de içerecek şekilde JSON formatında oluşturulmuş ve görselleştirilmiştir. Böylece, arşivsel iş akışlarında dijital provenans fonksiyonel hale getirilmiştir. Oluşturulan tüm provenans veri hatları BagIt kütüphanesi ile paketlenmiş ve veri setleri ile tüm süreçlerin

[258] W3C tarafından geliştirilen ve PROV standardından türetilen prensipleri kullanarak makine öğrenmesi iş akışlarının temsili için tasarlanan provenans veri modelidir. Model, tüm makine öğrenmesi süreçlerinin yaşam döngüsünü dokümante etmek için yapılandırılmış bir bağlam sunmaktadır. Ayrıca; şeffaflık, yeniden üretilebilirlik ve hesap verebilirliği de sağlar.

korunması için Arhcivematica programına yüklenmiştir. Böylece, uzun süreli korumanın sürdürülebilirliği hedeflenmiştir.

YEDİNCİ BÖLÜM

7. SONUÇ, TARTIŞMA VE DEĞERLENDİRME

Bu bölümde, analizler sonucunda elde edilen bulguların sonuçları açıklanmış, sonuçlar ayrıca literatür destekli karşılaştırılmış ve tüm çalışma değerlendirildikten sonra çıkarımlar yapılmıştır.

7.1. Sonuç

7.1.1. Makine Öğrenmesi

İlk olarak, 3.6 milyondan fazla örnek ve 11 öznitelikten oluşan ve ABD iç hat uçuş verilerini (1990-2009) içeren büyük ölçekli veri setinde denetimli ve denetimsiz öğrenme algoritmaları uygulanmıştır. Analiz süreçleri, veri ön işleme ile başlamıştır. Orijinal veriler, YAML dosyasından çıkarılan metaverilerle biçimlendirilmiştir. Yolcu sayısı, koltuk sayısı, uçuş sayısı ve mesafe gibi kritik değişkenlerdeki sıfır değerleri silinmemiş, aralarındaki korelasyonları koruyan ve bilgi kaybını en aza indiren KNNImpute yöntemi kullanılarak doldurulmuştur. Ayrıca, Kolmogorov-Smirnov testleri ile basıklık-çarpıklık testleri de yapılmış ve veri setinin normal dağılıma uymadığı tespit edilmiştir. Veri setinde bulunan çok sayıda aykırı değer; çeyrekler arası aralık, logaritmik dönüşüm, kırpma ve standardizasyon işlemlerine tabi tutulmuştur. Önişleme adımları ile, tüm sayısal özniteliklerde çarpıklık ve basıklık skorlarında önemli iyileştirmeler sağlanmıştır. Ayrıca, öznitelik mühendisliği ile koltuk doluluk oranı, uçuş sıklığı ve uçuş başına metrikler gibi değişkenler türetilmiş ve model yorumlanabilirliği artırılmıştır. Keşifsel istatistiksel analiz ile değişkenler arası korelasyonlar ve çeşitli temel istatistikler gösterilmiştir. Basit doğrusal regresyon analizinde koltuk sayısı ve yolcu sayısı değişkeni arasındaki ilişkiler incelenmiş ve %91,15'lik R^2 skoru elde edilirken; koltuk sayısı ve uçuş sayısı arasındaki ilişkide ise %86,18'lik R^2 skoru yakalanmıştır. Yolcu sayısı ve uçuş sayısı ile oluşturulan model ise %81,60'lık R^2 skoru elde edilmiştir. Çoklu doğrusal regresyonda ise öncelikle uçuş sayısı-mesafe-koltuk sayısı değişkenlerinin yolcu sayısı ile ilişkisi incelenmiş ve %91,97'lik R^2 skoru elde edilmiştir. Uçuş frekansı ve uçuş başına ortalama mesafe ile uçuş başına yolcu sayısı değişkenleri arasındaki ilişki ise neredeyse bulunamamıştır (R^2=12,52). Sınıflandırma analizlerinde ise lojistik regresyon ile mesafe değişkeni kendi içerisinde kısa, orta ve uzun mesafe olarak 3 ayrı alt sınıfa ayrılmış ve %93,83'lük doğruluk skoru elde edilmiştir. Öğrenme eğrisi ile model performansı da test edilmiş ve aşırı öğrenme durumları tespit edilmiştir. Çözüm olarak ise Stratified k-Fold CV yaklaşımı uygulanmış (cv=5) ve aşırı öğrenme ile yüksek bias (önyargı) durumları

giderilmiştir. Ayrıca; mesafe değişkeni, dört farklı model (rastgele orman, yığınlama, lojistik regresyon, Naive Bayes) üzerinde analiz edilmiş ve sonuçları karşılaştırılmıştır. Yığınlama modeli en başarılı model olmuştur (doğruluk skoru=0,9990). Rastgele orman sınıflandırıcısı ise 0,9983'lük doğruluk skoru yakalayarak başarılı bir performans göstermiştir. Lojistik regresyon sınıflandırıcısı 0,9385'lik doğruluk skoru ile diğer iki modele göre biraz daha düşük performans gösterirken Naive Bayes sınıflandırıcısı %49,96'lık doğruluk skoru ile en başarısız performansa sahip olmuştur. Dört modelin de öğrenme eğrileri gözlemlenmiş ve herhangi bir aşırı öğrenme durumuna rastlanmamıştır. Bir diğer sınıflandırma görevi ise uçuş başına yolcu sayısı değişkeninin kendi içinde 5 ayrı alt sınıfa ayrılmasıyla oluşturulmuştur. Veri setindeki kategorik değerler, one hot encoding tekniğiyle dönüştürülmüş ve oluşabilecek çoklu doğrusallığı ortadan kaldırmak için temel bileşen analizi[259] uygulanmıştır. Ayrıca, one-hot encoding tekniği ile kodlanan kategorik veriler ile diğer sayısal değerleri içeren dataframe, hesaplama maliyetinin azaltılması için float64'ten float32'ye çevrilmiş ve dataframe, Numpy dizisine dönüştürülmüştür. Rastgele orman, karar ağaçları, KNN, Naive Bayes ve yapay sinir ağları modelleri ile sınıflandırma işlemleri gerçekleştirilmiştir. Her bir model için RandomizedSearchCV ile en iyi hiperparametreler aranmış ve modeller en iyi değerlerle tekrar eğitilmiştir. Analizler sonucunda rastgele orman algoritması %94,40, karar ağaçları %90,71, KNN %79,05, Naive Bayes %45,72 ve yapay sinir ağları %99,59 doğruluk skorları elde etmiştir. Bu duruma göre en başarılı iki model yapay sinir ağları ve rastgele orman sınıflandırıcıları olmuştur.

7.1.2. Evrişimli Sinir Ağları

İkinci olarak; 30 farklı sınıfa sahip, 600x600 piksel boyutlarında ve 10.000 uydu yer görüntüsünden oluşan AID veri seti, sınıflandırma görevleri için kullanılmıştır. Bu kapsamda başlangıçta, temel bir CNN mimarisi geliştirilmiştir. Model; 32,64,128,256 ve 512 nörondan oluşan 5 evrişim bloğu ile 256 ve 512 nörondan oluşan 2 tam bağlantılı katman ve 0,5'e ayarlanmış bırakma katmanından oluşmaktadır. Model, ayrıca toplamda 8.142.302 parametreye sahiptir. Veri önişleme kapsamında ise görüntü boyutları 224x224 boyutuna düşürülmüş ve daha sonra normalizasyon uygulanarak tüm piksel değerleri [0,1] aralığına indirgenmiştir. Model, ilk olarak optimizasyon algoritması olmadan test edilmiş ve aşırı öğrenme göstererek %57 doğruluk skoru üretmiştir. Adam optimizasyon algoritması seçilerek tekrar eğitilen modele, öğrenme oranını dinamik olarak ayarlamak için ReduceLROnPlateau entegre edilmiştir. Yapılan bu optimizasyonlar ile test doğruluğu %76,3'e ulaşarak model biraz

[259] PCA

daha iyileştirilmiştir. Modelin gücünü daha da artırmak ve aşırı öğrenmeyi minimumda tutmak veya ortadan kaldırmak için veri artırma teknikleri kullanılmıştır. Bu kapsamda; rastgele döndürmeler, parlaklık artırma, kontrast değişimi ve dolgu ekleme ile rastgele kırpma işlemleri yapılmıştır. Veri artırımı yapılan model, %89,30 doğruluk skoru elde ederek en yüksek seviyesine ulaşmıştır. Model ayrıca, optimum giriş yapılandırmalarını belirlemek için çeşitli çözünürlüklerde (64×64-300×300) ve yığın boyutlarında (8-16) eğitilmiştir. 250×250 piksel çözünürlük ve 8 yığın boyutu ile %90,90 test doğruluğu elde edilmiştir. Bunların yanında; Adam, Adamax, Adagrad, RMSProp ve SGD dahil olmak üzere çeşitli optimizasyon algoritmaları karşılaştırmalı olarak değerlendirilmiştir. Adamax optimizasyon algoritması, 128x128 giriş değeri ile %91,40'a kadar test doğruluğu sağlayarak en iyi performansa ulaşmıştır. Modelin eğitim-doğrulama-test bölümlendirmesine olan duyarlılığını değerlendirmek için beş veri seti bölümlendirme tekniği (%90-%5-%5, %80-%10-%10, %70-%15-%15, %60-%20-%20 ve %50-%25-%25) ile de eğitimler gerçekleştirilmiş ve %90-%5-%5 ve %80-%10-%10 bölümlendirmeleriyle eğitilen modellerin diğer bölümlendirmelere göre daha yüksek performans sağladığı görülmüştür. Başka bir deyişle, daha büyük eğitim seti bölümlendirmesi, daha iyi genellemeyi sağlamıştır. Bir diğer analiz ise öğrenme oranı (0,0001-0,00001) bazlı gerçekleştirilmiştir. Bu analizde ise düzleştirme katmanı, GlobalAveragePooling2D ile değiştirilerek, hem toplam parametre sayısı azaltılarak genelleştirme artırılmış hem de potansiyel aşırı öğrenme riski azaltılmıştır. 0,0001 öğrenme oranıyla eğitilen modeller, diğer öğrenme oranına göre daha iyi performans göstermiştir. Performans değerlendirmeleri ve karşılaştırmaları kayıp ve doğruluk grafikleri, karmaşıklık matrisleri ve AUC-ROC eğrileri kullanılarak yapılmıştır.

7.1.3. Transfer Öğrenmesi

Üçüncü olarak, transfer öğrenmesi kapsamında EfficientNet modelinin B0, B1, B2, B3 ve güncelleştirilmiş varyantları V2B0, V2B1, V2B2 ve V2B3 kullanılarak sınıflandırma analizleri yapılmıştır. Bu kapsamda 8 model, 224x224'lük giriş değerleriyle öncelikle ImageNet ağırlıkları olmadan temel modeller olarak eğitilmiştir. Modele temel özellik çıkarıcıdan sonra 128 nöron içeren ve ReLU aktivasyonu ile tam bağlantılı katman eklenmiştir. EfficientNet tabanı dondurulmuş ve yalnızca en üstteki yoğun katmanlar, %90-%5-%5, %80-%10-%10, %70-%15-%15, %60-%20-%20 ve %50-%25-%25 eğitim-doğrulama-test bölümlendirmesi şeklinde eğitilmiştir. İlk analizlerde en yüksek doğruluk skoruna sahip modeller, Keras Tuner Random Search tekniği ile en iyi çekirdek düzenleyici, bias (önyargı) düzenleyici, aktivite düzenleyici, öğrenme oranı, optimizasyon algoritması ve yoğun katman değerleri bulunana

kadar arama yapılmış ve en iyi hiperparametre değerleri ile modeller eğitilmeden önce veri artırma tekniği eklenmiştir. Analizler sonucunda B3, V2B2 ve V2B3 modelleri %97,20 doğruluk skoru elde etmiştir.

7.1.4. Dijital Provenans ve Archivematica Uygulaması

Uygulamalar sonucunda elde edilen çıktılar ve bu süreçlerin tamamı, JSON formatında provenans verisi olarak yapılandırılmış ve paketlenmiştir. Bu kapsamda, orijinal veri setlerinin yanı sıra, bu setlerden türetilmiş varyantlar, çeşitli grafiksel çıktılar ve Python programlama dili ile geliştirilen provenans kodları, uzun süreli koruma amacıyla Archivematica programına yüklenmiştir. Bu sürecin temel amacı, hem hesaplamalı modeller geliştirmek hem de bu işlemlerin her aşamasının belgelenmesini ve korunmasını sağlamaktır. Bu bağlamda, Archivematica'nın kullanılması, hesaplamalı süreç verilerinin uzun süreli dijital korumaya alınmasını sağlamıştır. Archivematica uygulaması aracılığıyla tüm analiz süreçleri ve çıktılarının AIP, DIP ve SIP şeklinde oluşturulması sağlanmış ve böylece çıktıların uzun süreli korunabilirliğinin sürdürülmesi hedeflenmiştir. Ancak, Archivematica programının kapsamlı yapısı göz önüne alındığında, bu tezde sadece ön işleme ve paketlerin oluşturulması aşamaları detaylandırılmıştır.

7.2. Tartışma

7.2.1. Makine Öğrenmesi Analizleri

Literatürde doğrudan veri seti 1 kullanılarak yapılan herhangi bir bildiri, tez, makale, kitap ve kitap bölümü çalışmasına rastlanılmamış olup, ilk makine öğrenmesi uygulamaları bu tezde verilmiştir. Ancak; iç ve dış hat uçuşlara, farklı veri setleri kullanılarak makine öğrenmesi algoritmalarının uygulandığı çalışmalar mevcuttur. Sismanidou, Tarradellas ve Suau-Sanchez (2022) çalışmasında ABD'de bulunan ve 2018 Temmuz ayındaki 21 havaalanındaki gecikmeli uçuşları, bağlantılı yolcular üzerinde incelemiştir. Makine öğrenmesi yöntemlerinden regresyon ve yapay sinir ağları ile eğitilen modellerde gecikmeler tahmin edilmiş ve karşılaştırılmıştır. Vadlamani, Shafiq ve Baysal (2022) çalışmasında Southwest Havayollarının yeni pazarlara giriş modellerini analiz ve tahmin edebilmek için lojistik regresyon ve karar ağaçları algoritmaları kullanmıştır. Wild vd. (2021), çalışmasında hava taşımacılığı talebini modellemede çeşitli makine öğrenmesi algoritmalarının (çoklu doğrusal regresyon ve yapay sinir ağları) performansını karşılaştırmıştır. Fırat, Yıltaş-Kaplan ve Şamlı (2021) tarafından gerçekleştirilen çalışmada ise, herhangi iki ülke arasındaki uçuş operasyonunda en yüksek yük faktörünü sağlayabilecek optimum koltuk kapasitesini belirlemek için bir yöntem tanıtılmıştır.

Yapay sinir ağı, doğrusal regresyon, gradient boosting ve rastgele orman vb. teknikler kullanılarak makine öğrenmesi yöntemleri uygulanmış ve problemi çözmek için bir yazılım geliştirilmiştir. Çalışmada Dünya Bankası veritabanından türetilen veri seti kullanılmıştır. Tang (2022), çalışmasında uçuş gecikmelerini tahmin etmek üzere denetimli makine öğrenmesi modelleri uygulamıştır. Veri seti, Kaggle platformundan elde edilen ve JFK Havaalanı kalkışlı bir yıllık uçuş bilgilerini içermektedir. Uçuş gecikmelerinin ikili sınıflandırmasını tamamlamak üzere lojistik regresyon, KNN, Gauss Naive Bayes, karar ağacı, destek vektör makinesi, rastgele orman ve gradyan destekli ağaç algoritmaları kullanılmıştır.

7.2.2. Transfer Öğrenmesi ve Evrişimli Sinir Ağları Analizleri

Xia vd. (2017) tarafından yapılan çalışmada AID veri seti, uydu görüntülerini sınıflandırma modellerini değerlendirmek için önerilmiştir. Çalışmada CaffeNet, GoogLeNet ve VGG-VD-16 gibi ön eğitimli modeller kullanılmıştır. GoogLeNet ön eğitimli modeli kullanılarak %50 eğitim %50 test bölümlendirmesinde %86,39; %20 eğitim %80 test bölümlendirmesinde ise %83,44'lük bir test doğruluğu elde edilmiştir. CaffeNet ön eğitimli modeli kullanılarak %50 eğitim %50 test bölümlendirmesinde ise %89,53; %20 eğitim %80 test bölümlendirmesinde ise %86,86'lık bir test doğruluğu elde edilmiştir. VGG-VD-16 modelinde ise %50 eğitim %50 test bölümlendirmesinde %89,64; %20 eğitim %80 test bölümlendirmesinde ise %86,59'luk bir test doğruluğu elde edilmiştir.

Jin vd. (2018), orijinal AID veri setinin tanıtılmasından yaklaşık 1 yıl sonra, AID++ olarak adlandırılan, veri seti, 400.000'den fazla görüntü ve 46 sınıftan oluşmaktadır. Çalışmada, bazı ResNet-50 modelleri çeşitli veri setleri (AID, AID++, ImageNet, NWPU, RSI-CB) üzerinde eğitilmiş ve ön eğitimli bu modeller, küçük ölçekli bir veri seti olan WHU19 görüntülerini sınıflandıran bir özellik çıkarıcı olarak kabul edilmiştir. Bu doğrultuda yapılan testlerde AID veri seti %90,45'lik doğruluk skoruna erişirken AID++, %97,38'lik bir doğruluk skoruna sahip olmuştur.

Tong vd. (2020) tarafından yapılan çalışmada, görüntü sınıflandırma performansını artırmak için kanal dikkati tabanlı bir DenseNet (CAD) mimarisi önerilmiştir. Çalışmada DenseNet-121'e dikkat modülü entegre edilmiş ve sınıflar arası benzerlik ile aşırı öğrenmeyi azaltmak için çapraz entropi kayıp fonksiyonu kullanılmıştır. Çalışma; 288x288 görüntü boyutu, 16 yığın boyutu ve Pytorch kütüphanesi kullanılarak gerçekleştirilmiştir. %50 eğitim %50 test bölümlendirmesinde %97,16'lık doğruluk skoru elde edilmiştir.

Alhichri vd. (2021) tarafından yürütülen çalışmada, EfficientNet-B3-Attn-2 adı verilen dikkat mekanizması entegre edilerek geliştirilmiş bir sınıflandırma çerçevesi önerilmiştir. Çalışma Google Colab ortamında yürütülmüş ve 32 yığın boyutu, 0,001 öğrenme oranı (ilk 15 epoch - sonraki 15 epoch 0,0001) ve 256x256 görüntü boyutunda hesaplama işlemleri yapılmıştır. %50 eğitim ve %50 test bölümlendirmesinde temel EfficientNetB3 modeli %94,12, EfficientNetB3-Attn-2 modelinde ise %96,56'lık doğruluk skorları elde edilmiştir. Ayrıca, %20 eğitim ve %80 test bölümlendirmesinde ise EfficientNetB3-Attn-2 modeli ise %94,45 doğruluk skoru vermiştir. Çalışma, Google Colab ortamında gerçekleştirildiğinden, çeşitli kısıtlamalar (hesaplama, RAM, depolama vb.) nedeniyle %80 eğitim %20 test bölümlendirmesi için eğitim gerçekleştirilmemiş ve bu yüzden sonuçlar sağlanamamıştır.

İnik (2023) tarafından gerçekleştirilen çalışmada ise AID veri seti, transfer öğrenmesi modelleri (VGG19, AlexNet, GoogLeNet, ResNet-50, ResNet-101) kullanılarak sınıflandırılmıştır. Çalışmada önerilen çerçeve; budanmış VGG19 olup, diğer ön eğitimli modeller ile sonuçları karşılaştırılmıştır. İşlemler; 50 epoch, 64 yığın boyutu, 0,001 öğrenme oranı başlangıcı ve SGDM optimizasyon algoritması seçilerek başlatılmıştır. %50 eğitim %50 test bölümlendirmesinde en başarılı sonucu %94,26 doğruluk skoru ile VGG19 ön eğitimli modeli vermiştir. Budanmış VGG19 ön eğitimli modeli ise %66,50 daha az parametre içermesine rağmen, sadece %0,70'lik bir doğruluk skoru kaybı ile yine de yüksek sonuç vermiştir.

Bi vd. (2020) çalışmasında uydu yer görüntülerinin sınıflandırılmasında kalıntı dikkat tabanlı yoğun bağlı evrişimli sinir ağı (RADC-Net) çerçevesi önermiş ve AID ile diğer veri setleri üzerinde eğitimler gerçekleştirmiştir. Çalışmada ağırlık parametreleri için rastgele başlatma kullanılmış ve tüm bias (önyargı) parametreleri 0,001 olarak ayarlanmıştır. Öğrenme oranı ilk 100 epoch için 0,001 şeklinde ayarlanmış; son epocha kadar öğrenme oranı her epochta 10'a bölünmüştür. Eğitim süreci, öğrenme oranı 1e-6 (1×10^{-6})'ya düştüğünde sonlanmaktadır. Aşırı öğrenme sorununu önlemek için, L2 normalizasyonu (5×10^{-6}) olarak seçilmiş ve bırakma[260] oranı 0,2 olarak ayarlanmıştır. %20 eğitim %80 test bölümlendirmesinde %88,12; %50 test %50 eğitim bölümlendirmesinde ise %92,35'lik doğruluk skoru yakalanmıştır.

Diğer benzer çalışmalar ise Tablo 84'te özetlenmiştir.

[260] Dropout

Tablo 84. Transfer Öğrenmesi Literatür Çalışmaları ve Karşılaştırmalar

Çalışma	Metod	Yığın Boyutu	Öğrenme Oranı	Eğitim	Doğrulama	Test	Doğruluk
İnik (2023)	Budanmış VGG19	64	0,001	%50	-	%50	%94,26
Alhichri vd. (2021)	EfficientNet-B3-Attn-2	32	0,001-0,0001	%50		%50	%96,56
Bahri vd. (2020)	Transfer Öğrenmesi (MLP+NASNet Mobile)	16	0.0001	%70	-	%30	%98,10
Tong vd. (2020)	Kanal Dikkati Tabanlı DenseNET	16	-	%50	-	%50	%97,16
Bi vd. (2020)	RADC-Net		0,001	%50	-	%50	%92,35
Cao vd. (2021)	Öz-Dikkat Tabanlı Derin Özellik Füzyonu (SAFF	-	-	%50	-	%50	%93,83
Cheng vd. (2018)	Ayrımcı CNN (D-CNN)	-	0,01-0,001	%50	-	%50	%96,89
Anwer vd. (2018)	TEX-Nets	-	0,001	%50	-	%50	%95,73
Jin vd. (2018)	ResNet-50	-	-	-	-	-	%90,45
	GoogleNet	-	-	-	-	-	%97,38 (AID++)
Xia vd. (2017)	VGG-VD-16	-	-	%50	-	%50	%89,64
	GoogLeNet	-	-	%50	-	%50	%86,39
Bu Çalışma	**EfficientNetV2B2**	**8**	**0,001**	**%50**	**%25**	**%25**	**%76,24**
	EfficientNetV2B2 (Kernel[L1+L2]+Bias[L2]+Aktivite Düzenleyici[L2])	**16**	**0,0001**	**%90**	**%5**	**%5**	**%97,20**

Tablodan da görüldüğü üzere, tezde %90 eğitim %5 doğrulama %5 test bölümlendirmesinde EfficientNetV2B2[261] modelinde, çekirdek-önyargı-aktivite düzenleyicileri destekli olarak %97,20'lik sınıflandırma doğruluğu elde edilmiştir. Her bir model için en iyi çekirdek, bias, aktivite düzenleyici vb. değerleri Keras Tuner RandomSearch ile bulunmuş ve uygulanmıştır. Literatürde incelenen çalışmaların nerdeyse tamamında, her ne kadar hem yöntem hem de veri setlerindeki bölümlendirme teknikleri farklı olsa da (çoğunlukla %50-%50 ve %20-%80), modellere dikkat mekanizmaları eklemeden yapılan optimizasyonlar ile düzenleyici tekniklerin eklenmesiyle de oldukça başarılı sonuçların elde edilebileceği gösterilmiştir.

Chen vd. (2018) tarafından önerilen evrişimli sinir ağı (CNN) 4 evrişim katmanı, 3 havuzlama katmanı ve 1 tam bağlantılı katmandan oluşmaktadır. Modelin girdi boyutu 100x100'lük RGB bir görüntüdür ve kademeli olarak hiyerarşik özellikleri (kenarlar, dokular, şekiller ve daha soyut desenler) çıkardıktan sonra 225 boyutlu çıkış vermektedir. Her deneyde, tam bağlı katmanın çıktısının görüntü özellikleri olarak çıkarıldığı bir özellik çıkarıcı olarak CNN kullanılmış ve sınıflandırıcı olarak doğrusal destek vektör makinesi benimsenmiştir. Temel CNN modeli, AID veri setinde %78,95 doğruluk oranına ulaşmıştır. Tezde ise veri seti 2 için sıfırdan oluşturulan VGG benzeri temel bir evrişimli sinir ağı modeli, optimize edildikten ve düzenlileştirildikten sonra maksimum %91,40'lık bir doğruluk skoruna erişmiştir. Bu fark,

[261] V2B2 modelinin yanı sıra B3 ve V2B3 modellerinde de %97,20 doğruluk skoruna ulaşılmış, ancak diğer tüm metrikler ve doğrulama kaybı skorunda en iyi sonucu V2B2 modeli verdiğinden tabloda gösterilmiştir.

evrişimli sinir ağları arasındaki mimari farklılıklara, (beş evrişim katmanı), düzenlileştirme için bırakma katmanlarının eklenmesine ve çeşitli optimizasyon stratejilerine bağlanabilir.

7.2.3. Dijital Provenans ve Archivematica Uygulaması

Schlegel ve Sattler (2023), Git ve MLflow kullanan makine öğrenmesi geliştirme hatlarından W3C PROV uyumlu provenans verisini çıkarmak için tasarlanmış MLflow2PROV adlı yeni bir araç önermiştir. Yazarlar dijital provenansın yakalanmasında; model yaşam döngüsü boyunca tüm iş akışlarının izlenebilirliği, yorumlanabilirliği ve doğrulanabilirliğinin bu araç ile sağlanabileceğini iddia etmektedir. GitHub platformunda[262] kodları ve örnek uygulamaları bulunabilir.

Turnau, vd. (2023) ise Datalog tabanlı sorgu araçlarından yararlanarak makine öğrenmesi modelleri için provenans odaklı bir çerçeve önermektedir. Önerilen modelin temel yeniliği; provenans yakalayıcısının doğrudan modelin eğitim aşamasına yerleştirilmesi ve bu provenansı SHAP değerlerini türetmek için yeniden kullanılmasına olanak sağlanmasıdır.

Souza vd. (2019) ise ham bilimsel verilerden eğitilmiş makine öğrenmesi modellerine kadar provenansın izlenmesinin çoğunlukla manuel, parçalı ve yüksek maliyetli olduğunu belirttikten sonra uçtan uca bir çözüm önermiştir. Çözüm yaklaşımı ise; makine öğrenmesi yaşam döngülerinin verilerle birbirine bağlı birden fazla iş akışı olarak modellenmesi ve iş akışları yürütülürken provenansın izlenmesidir. Bunun için iş akışlarına provenans yakalama ajanları eklenerek yaşam döngüsünün süreçlerine dair üretilen veri ile alana özel verileri birleştiren daha kapsamlı provenans analizleri gerçekleştirilmiştir.

Samuel, Löffler ve König-Ries (2021), makine öğrenmesinin yönetimsel kararlara desteğinden bilimsel keşfe kadar birçok alanda ayrılmaz bir parça haline geldiğini, ancak yayınlanmış sonuçların yeniden üretilmesinin yetersiz dokümantasyon, versiyon eksikliği ve yapılandırılmış metaverilerinin yokluğu nedeniyle, yeteri kadar etkili kullanılmadığını belirterek, makine öğrenmesi süreçlerine FAIR veri ilkeleri eklemeyi önermiştir. İlkenin sadece veri setlerine değil, kullanılan araçlara, algoritmalara, iş akışlarına da genişletilmesini savunarak, Jupyter Notebook uzantılı dosyalarda provenansı yakalayan, çalışmaları ve sonuçları karşılaştıran ve kodlara ait metaverileri depolayabilen "ProvBook" provenans aracını da tanıtmıştır. ProvBook, modelin her çalışmasındaki provenansı otomatik olarak çalıştırdıktan sonra, kod ortamını da birlikte sağlayarak yeniden üretilebilirliğe yardımcı olmaktadır.

[262] https://github.com/mariusschlegel/mlflow2prov

Bahsedilen çalışmaların yanında, makine öğrenmesi veri hatlarındaki provenansı yakalayabilmek için çok çeşitli uygulamalar geliştirildiği de görülmüştür (Curcin, 2016; Rupprecht vd., 2020; Walchshofer vd., 2023; Chapman vd., 2024; Schelter, Guha ve Grafberger, 2024).

Tezde ise, daha önce belirtildiği üzere; arşiv bilimi ile bilgisayar biliminin ortak noktalarından biri olan "provenans" kavramının hesaplamalı arşiv bilimi bağlamında oluşturulması sağlanmıştır. Hesaplamalı süreçler böylece, tanımlayıcı metaveriler ile provenansının oluşturulması sonucunda, araştırma çıktılarının hem yorumlanabilmesi hem de arşivsel bağlamda erişilebilirliği sağlanmıştır.

7.3. Değerlendirme

Tezin temel araştırma sorusu şu şekildeydi:

"Yapılandırılmış veri ile yapılandırılmamış veri niteliğindeki arşiv yığınlarının barındırdığı gizil ilişkilerin ve sahip olduğu örüntülerin analitik yöntemlerle açığa çıkarılması, orijinal formatı ile birlikte çıktısı sağlanan tüm süreçlerin uzun süreli korumasının sağlanması ve geleceğe yönelik kullanılabilirliğinin sürdürülebilmesinin yanında tüm öznitelikleri ve kapsamı ile bir "nitelikli veri havuzu" özelliğinde arşivsel referans noktası olabilmesi için arşiv bilimi mantığında hangi geliştirmelerin veya iyileştirmelerin yapılması gerekmektedir?"

Arşiv bilimi uygulamaları, giderek artan bir şekilde veri bilimi temelli olarak düşünülmekte özellikle veri provenansı ve otomatik analiz süreçlerinden faydalanan bir hesaplama paradigmasına doğru evrilmektedir. Arşivsel yapılar; günümüzde durağan depolama ve saklamanın yanında, etkileşimli veri modellemesi, anlamsal zenginleştirme ve uzun süreli izlenebilirliği de barındırmalıdır. Bu paradigma değişimi ile, geleneksel arşiv ilkeleri değişime uğrayarak arşivsel içeriklerin işlenmesi, zenginleştirilmesi ve hesaplamalı kullanımını savunan hesaplamalı arşiv bilimi mutlaka benimsenmelidir. Bu bağlamda tezde, dijital veri setleri üzerinde yapay zeka teknikleri ile örüntü tanıma ve öznitelikler arasında gizli kalan ilişkiler ortaya çıkarılmış; hesaplamalı süreçlerin dijital arşiv standartlarına uygun olarak provenansları JSON formatında oluşturulmuş, paketlenmiş ve orijinal formatları ile birlikte türetilen tüm veri setleriyle birlikte Archivematica programına yüklenmiş ve arşiv paketleri üretilmiştir. Temel araştırma sorusunda vurgulanan "nitelikli veri havuzu" tüm bu aşamalardan sonra başarıyla oluşturulmuştur.

Genel araştırma soruları ve cevapları ise aşağıda gösterilmektedir:

1) Arşivlerin artan dijitalleşmesi arşiv bilimi için hangi zorlukları ortaya çıkarmaktadır?

Arşiv biliminin bu konudaki en büyük zorluğu, hacim ve çeşitlilik ile mücadele etmektir. Dijital verilerin analizi donanım gücüne bağlı olduğundan ölçeklenebilirlik de bir dezavantaj olarak ortaya çıkmaktadır. Bir diğer zorluk ise özgünlüğün kaybıdır. Dijital materyallerin standart dışı manipülasyonu, provenans ve özgünlük risklerini artırmaktadır. Ayrıca, dijital ortamın eskimesi ve kırılganlığı da verilere erişimde sorunlara yol açmaktadır.

2) Dijital veriler, yeni nesil arşivler için temel bir itici gücü nasıl oluşturmaktadır?

Dijital veriler, arşivlerin durağan bir depo olma özelliklerini geliştirerek, dinamik bilgi altyapılarına dönüşmesine zorlamaktadır. Hacim ve çeşitlilik bazında elle yapılması mümkün olmayan analizlerin (örneğin; otomatik sınıflandırma, örüntü tanıma ve trend analizi) yapılabilmesine olanak tanıyan programlama dilleri ve yapay zeka ile verinin tüm yaşam döngüsü ve kullanılan kaynakların gösterimi sağlanarak hem bilimsel üretilebilirliğe hem de arşiv bilimine katkı sağlanmaktadır.

3) Hesaplamalı arşiv bilimi bağlamında ele alındığında, arşivciler hangi fırsatları yakalayabilir?

Arşivciler, hesaplamalı arşiv bilimi uygulamaları ile her türlü dijital materyalde; tahmin analizleri, sınıflandırma, otomatik provenans çıkarımı ve bilgi keşfi gibi yenilikçi uygulamalar ile arşivleri daha da zenginleştirebilir.

4) Hesaplamalı arşiv bilimi uygulamaları için arşivci ve belge yöneticilerine yönelik temel politika ve stratejiler neler olmalıdır?

Bu bağlamda, lisans ve lisansüstü düzeyde hesaplamalı arşiv bilimi ile ilgili yeni müfredatlar (programlama dilleri, yapay zekâ, veri modelleme, dijital provenans, hesaplamalı düşünme vb.) eklenebilir. Özellikle Jupyter Lab ve Jupyter Notebook araçlarının kullanım becerisi kazandırılabilir. Bilgisayar bilimciler, istatistikçiler, vb. ile daha fazla işbirliği planlanabilir.

Özel araştırma soruları ve cevapları ise şu şekildedir:

1) Hesaplamalı arşiv bilimi bağlamında bir arşivsel problemi tanımlamak için gereken kriterler nelerdir?

Bir dijital arşiv problemi; öncelikle korunması, analiz edilmesi veya yorumlanması gereken yapılandırılmış, yarı yapılandırılmış veya yapılandırılmamış dijital materyalleri içermelidir. Problemin niteliği belirlenirken, ilgili verinin provenansı detaylı olarak incelenmelidir. Eğer mevcut provenans bilgisi yetersiz durumdaysa, araştırma sürecinde yeni bir provenans kaydı oluşturulmalıdır. Ayrıca, problemin tanımlanması aşamasında hesaplamalı yöntemler mutlaka

etkin biçimde kullanılmalıdır. Tanımlanan problem, aynı zamanda dijital arşiv değerleri olan özgünlük, bütünlük, erişilebilirlik ve kullanılabilirlik vb. ile tutarlı olmalıdır.

2) Bilinen bir arşivsel problemin çözümü için hangi hesaplamalı yöntemler nasıl seçilmelidir?

Yöntem, öncelikle veri türüne göre seçilmelidir. Örneğin; JPEG veya PNG uzantılı görüntüler için evrişimli sinir ağı veya ön eğitimli ağlar; tablo verileri için denetimli makine öğrenmesi modelleri seçilebilir. Yöntem ayrıca sorunun kapsamına göre değişkenlik göstermektedir. Örneğin; değişkenler arası ilişkilerin tanımlanması regresyon görevlerinin ilgi alanına girerken; değişken içerisindeki sınıfların tespiti için kümeleme veya sınıflandırma algoritmaları kullanılmalıdır.

3) Uygulanacak yönteme göre bir altyapı oluşturma süreçlerinde hangi adımlar yer almalıdır?

Yöntem seçildiğinde, yapılması gereken standart süreçler bulunmaktadır. İlk olarak mutlaka veri önişleme adımlarının tamamlanması gerekmektedir. Çünkü arşiv verisi, orijinal formatı ile saklandığından dolayı, algoritma ve modellerin efektif çalışabilmesi için mutlaka versiyonlanıp temizlenmesi ve düzenlileştirilmesi gerekmektedir (manifested kopyalar üzerinde). Örneğin; dijital materyal bir tablo verisi ise; eksik değer, 0, NA değer veya INF değer kontrolü yapılmalıdır. Ardından, aykırı değer kontrolü ile çeşitli istatistiksel dağılım uygulamaları da yapılabilir. Öznitelik mühendisliği ile yeni değişkenlerin türetilmesi, veri içerisinde bulunan ancak elle tespit edilemeyen ilişkilerin açığa çıkarılmasını sağlar. Ardından hedeflere göre makine öğrenmesi veya derin öğrenme modelleri seçilerek sonuçlar elde edilir. Tüm süreci belgeleyen bir provenans veri hattı oluşturulduktan sonra ise uzun süreli saklama için çeşitli standartlar bağlamında işlemler gerçekleştirilmelidir.

Hipotezlerin tamamı doğrulanmıştır:

Hipotez 1: Dijital arşiv verisinin işlenmesi, analiz edilmesi ve uzun süreli saklanması süreçleri; makine öğrenmesi, derin öğrenme ve hesaplamalı analiz gibi yeni teknolojiler aracılığıyla etkin biçimde gerçekleştirilebilir.

Dijital arşiv verileri, sınıflandırma ve tahmin gibi görevlerde güncel makine öğrenmesi ve derin öğrenme kütüphaneleri kullanılarak analiz edilmiş ve yüksek doğruluk skorlarına ve anlamlı ilişkilere erişilmiştir. Analiz süreçleri ve sonuçlar, provenans verisi olarak kaydedilmiş ve uzun süreli saklama programına transfer edilmiştir.

Hipotez 2: Dijital arşiv verisi üzerinde gerçekleştirilen hesaplamalı analizler, girdi ve çıktılarıyla birlikte orijinal veri seti ve manipüle edilmiş türevleri ile versiyonlanarak saklandığında, dijital provenans bütünlüğü korunabilir.

Orijinal veri seti, kopyalanarak çoğaltıldıktan sonra analizler bağlamında çok çeşitli türev veri setleri oluşturulmuş ve analizlerde kullanılmıştır. Ayrıca versiyonlanan veri setleri, bütünlüğün korunması amacıyla Archivematica mikro servislerinden geçirilmeden önce BagIt kütüphanesi ile paketlenmiştir.

Hipotez 3: Provenans verisi, tüm işlem adımlarıyla birlikte temsil edilerek dijital arşiv sistemlerinde bütüncül bir şekilde saklanabilir.

Veri setlerine uygulanan analizlerin süreç bilgileri, çıktıları ile birlikte JSON formatında kaydedilerek, Archivematica içerisinde standardizasyona tabi tutulmuştur. Sürece dair tüm bilgiler, çeşitli tekniklerle (PREMIS ve METS) gösterilmekte ve kullanılabilmektedir. Donanım ve yazılım yapılandırması, model mimarisi, hiperparametreler, veri seti ön işleme, değerlendirme metrikleri vb. çok sayıda veri, zaman damgalı olarak program içerisinde saklanmaktadır.

Tezin bütünü değerlendirildiğinde, arşiv biliminin hesaplamalı süreçlerle entegrasyonu, arşivcilere mesleki uygulamalarda yeni kavramsal çerçeveler, araçlar ve yöntemler benimseme olanağı sunarak, dijital varlık yöneticilerine dönüşümünü teşvik edecektir. Bu dönüşüm, kod tabanlı veya kodsuz makine öğrenmesi süreçlerinin ve dijital metodolojilerin anlaşılmasını zorunlu kılmakta; böylece hesaplamalı düşünmenin merkeze alınmasıyla geleneksel rollerin ötesine geçilerek, 21. yüzyılda toplumların ve kültürlerin dijital hafızasını şekillendirmede aktif birer aktör haline gelinmesi sağlanacaktır.

Son olarak, gelecek çalışmalarda arşivsel provenansın otomatik yakalanmasını sağlayacak araçlar geliştirilip, Archivematica programı ile entegrasyonu sağlanabilir. Ayrıca, yarı yapılandırılmış verilerin analizleri yapılarak provenans veri hatlarının Atom, Dspace vb. uygulamalarında dinamik bir şekilde gösterimi yapılabilir.

KAYNAKÇA

Ahmed, S. (2021). People of Remembrance: Archival Thinking and Religious Memory in Sufi Communities. *Archival Science*, 21, 9-23, doi: https://doi.org/10.1007/s10502-020-09346-9

Aho, A.V. (2012). Computation and Computational Thinking. *The Computer Journal*, 55(7), 832-835, doi: https://doi.org/10.1093/comjnl/bxs074

Ahrens, A. vd. (2022). An Introduction to Stacking Regression for Economists. S. Sriboonchitta, V. Kreinovich, W. Yamaka (eds.). *Credible Asset Allocation, Optimal Transport Methods, and Related Topics* içinde (ss. 7-29), 429, Springer: Cham, doi: https://doi.org/10.1007/978-3-030-97273-8_2

Ahsan, M.M. vd. (2021). Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance. *Technologies*, 9, 1-17, doi: https://doi.org/10.3390/technologies9030052

Akmon, D., Zimmerman, A. ve Daniels, M. (2011). The Application of Archival Concepts to a Data-Intensive Environment: Working with Scientists to Understand Data Management and Preservation Needs. *Archival Science*, 11, 329-348, doi: https://doi.org/10.1007/s10502-011-9151-4

Aleixo, J.E. vd. (2024). Artificial Intelligence Applied to Digital Marketing. S. Teixeira ve J. Remondes (eds.), *The Use of Artificial Intelligence in Digital Marketing: Competitive Strategies and Tactics* içinde (ss. 21-72). Hershey: IGI Global Scientific Publishing, doi: https://doi.org/10.4018/978-1-6684-9324-3.ch002

Alexandropoulos, S.A.N. vd. (2019). Stacking Strong Ensembles of Classifiers. J. MacIntyre, I. Maglogiannis, L. Iliadis, E. Pimenidis (eds.). *Artificial Intelligence Applications and Innovations* içinde (ss. 545-556), 559, Springer: Cham, doi: https://doi.org/10.1007/978-3-030-19823-7_46

Alexandropoulos, S.A.N., Kotsiantis, S.B. ve Vrahatis, M.N. (2019). Data Preprocessing in Predictive Data Mining. *The Knowledge Engineering Review*, 34(e1), 1-33, doi: 10.1017/S026988891800036X

Alfonseca, M. vd. (2021). Superintelligence Cannot be Contained: Lessons from Computability Theory. *Journal of Artificial Intelligence Research*, 70, 65-76.

Al-Hamadani, M.N.A. (2023). Classification and Analysis of the MNIST Dataset Using PCA and SVM Algorithms. *Military Technical Courier*, 71(2), 221-235, doi: https://doi.org/10.5937/vojtehg71-42689

Alhichri, H. vd. (2021). Classification of Remote Sensing Images Using EfficientNet-B3 CNN Model with Attention. *IEEE Access*, 9, 14078-14094, doi: 10.1109/ACCESS.2021.3051085

Aliferis, C. ve Simon, G. (2024). Overfitting, Underfitting and General Model Overconfidence and Under-Performance Pitfalls and Best Practices in Machine Learning and AI. G.J. Simon, C. Aliferis (eds.). *Artificial Intelligence and Machine Learning in Health Care and Medical Sciences* içinde (ss. 477-524). Cham: Springer, doi: https://doi.org/10.1007/978-3-031-39355-6_10

Aljaber, S. ve Almushaili, T. (2022). Artificial Intelligence. *International Journal of Engineering Research and Applications*, 12(12), 52-57, doi: 10.9790/9622-12125257

Alkhouly, A.A, Mohammed, A. ve Hefny, H.A. (2021). Improving the Performance of Deep Neural Networks Using Two Proposed Activation Functions. *IEEE Access*, 9, 82249-82271, doi: 10.1109/ACCESS.2021.3085855

Alsuwit, M.H., Haq, M.A. ve Aleisa, M.A. (2024). Advancing e-Mail Spam Classification using Machine Learning and Deep Learning Techniques. *Engineering, Technology & Applied Science Research*, 14(4), 14994-15001, doi: https://doi.org/10.48084/etasr.7631

Altan, G. (2019). DeepGraphNet: Grafiklerin Sınıflandırılmasında Derin Öğrenme Modelleri. *Avrupa Bilim ve Teknoloji Dergisi*, Özel Sayı, 319-329, doi: https://doi.org/10.31590/ejosat.638256

Alvarado, R.C. (2023). Data Science from 1963 to 2012. *ArXiv: 2311.03292*, 1-48, doi: https://doi.org/10.48550/arXiv.2311.03292

Amarbayasgala, T. vd. (2021). An Efficient Prediction Method for Coronary Heart Disease Risk Based on Two Deep Neural Networks Trained on Well-Ordered Training Datasets. *IEEE Access*, 9, 135210-135223, doi: https://doi.org/10.1109/ACCESS.2021.3116974

Ambacher, B. ve Conrad, M. (2021). Computational Archival Science is a Two-Way Street. Proceedings of *2021 IEEE International Conference on Big Data*, 2192-2199, doi: 10.1109/BigData52589.2021.9671873

Anderson, C.B. (2020). Preservation and Archiving of Digital Media. *Information Services and Use*, 40(3), 201-208, doi:10.3233/ISU-200085

Anwer, R.M. (2018). Binary Patterns Encoded Convolutional Neural Networks for Texture Recognition and Remote Sensing Scene Classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 138, 74-85, doi: https://doi.org/10.1016/j.isprsjprs.2018.01.023

Archivematica. (2025). Ingest. 02.02.2025 tarihinde https://www.archivematica.org/en/docs/archivematica-1.12/user-manual/ingest/ingest/#create-sip adresinden erişildi.

Arnold, C. vd. (2024). The Role of Hyperparameters in Machine Learning Models and How to Tune Them. *Political Science Research and Methods*, 1-8, doi: 10.1017/psrm.2023.61

Arseniev-Koehler, A. ve Foster, J.G. (2022). Machine Learning as a Model for Cultural Learning: Teaching an Algorithm What it Means to be Fat. *Sociological Methods & Research*, 51(4), 1484-1539, doi: https://doi.org/10.1177/00491241221122603

Arslan, İ. (2021). *Python ile Veri Bilimi*. İstanbul: Pusula.

Ary, D. vd. (2009). *Introduction to Research in Education*. Belmont: Wadsworth.

Asgar, M.R.G., Hidayat, R. ve Bejo, A. (2023). Comparison Euclidean Distance and Manhattan Distance as Classification in Speech Recognition System. *Proceedings of the International Conference on Educational Management and Technology*, 454-463, doi: https://doi.org/10.2991/978-2-494069-95-4_54

Assefa, S.A. vd. (2020). Generating Synthetic Data in Finance: Opportunities, Challenges and Pitfalls. *Proceedings of the First ACM International Conference on AI in Finance*, 44, 1-8, doi: https://doi.org/10.1145/3383455.3422554

Awad, M. ve Khanna, R. (2015). *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. New York: Apress Open, doi: http://dx.doi.org/10.1007/978-1-4302-5990-9

Ayeni, J.A. (2022). Convolutional Neural Network (CNN): The Architecture and Applications. *Applied Journal of Physical Science*, 4(4), 42-50, doi: https://doi.org/10.31248/AJPS2022.085

Ayodele, T.O. (2010). Machine Learning Overview. Y. Zhang (ed.). *New Advances in Machine Learning* içinde (ss. 9-18), Rijeka: IntechOpen, doi: 10.5772/9374

Bahri, A. (2020). Remote Sensing Image Classification via Improved Cross-Entropy Loss and Transfer Learning Strategy Based on Deep Convolutional Neural Networks. *IEEE Geoscience and Remote Sensing Letters*, 17(6), 1087-1091, doi: 10.1109/LGRS.2019.2937872

Bai, Y. (2022). RELU-Function and Derived Function Review. *SHS Web of Conferences*, 144, 1-5, doi: https://doi.org/10.1051/shsconf/202214402006

Bak, G. (2024). Digital Provenance. *Archival Science*, 24, 847-869, doi: https://doi.org/10.1007/s10502-024-09462-w

Barbalet, J. (2002). Science and Emotions. *The Sociological Review*, 50(S2), 132-150, doi: 10.1111/j.1467-954X.2002.tb03595.x

Barclay, I. vd. (2020). Certifying Provenance of Scientific Datasets with Self-Sovereign Identity and Verifiable Credentials. *ArXiv*, 1-6, doi: https://doi.org/10.48550/arXiv.2004.02796

Barr, D., Harrison, J. ve Conery, L. (2011). Computational Thinking: A Digital Age. *Learning & Leading with Technology*, Mart/Nisan 2011, 20-23. 13.11.2023 tarihinde https://files.eric.ed.gov/fulltext/EJ918910.pdf adresinden erişildi.

Bartneck, C. vd. (2021). *An Introduction to Ethics in Robotics and AI*. Cham: Springer, doi: https://doi.org/10.1007/978-3-030-51110-4

Baruah, R.D. ve Organero, M.M. (2024). Explicit Context Integrated Recurrent Neural Network for applications in Smart Environments. *Expert Systems with Applications*, 255(D), 1-14, doi: https://doi.org/10.1016/j.eswa.2024.124752

Basha, S.H.S. vd. (2020). Impact of Fully Connected Layers on Performance of Convolutional Neural Networks for Image Classification. *Neurocomputing*, 378, 112-119, doi: https://doi.org/10.1016/j.neucom.2019.10.008

Bastian, J.A. (2017). GLAMs, LAMs and Archival Perspectives. Heather MacNeil and Terry Eastwood (eds.), *Currents of Archival Thinking 2nd ed*. içinde (ss. 327-350) Santa Barbara, California: Libraries Unlimited.

Bauer, B. ve Wrisley, D.J. (2020). Digital Humanities Practices and Archival Appraisal in the Contemporary Arab World. Égypte/Monde Arabe, 22, 71-83, doi: https://doi.org/10.4000/ema.13166

Bawack, R.E. vd. (2022). Artificial Intelligence in e-Commerce: A Bibliometric Study and Literature Review. *Electronic Markets*, 32, 297-338, doi: https://doi.org/10.1007/s12525-022-00537-z

Bearman, D.A. ve Lytle, R.H. (1985). The Power of the Principle of Provenance. *Archivaria*, 21, 14-27.

Beaulac, C. ve Larribe, F. (2017). Narrow Artificial Intelligence with Machine Learning for Real-Time Estimation of a Mobile Agent's Location Using Hidden Markov Models. *International Journal of Computer Games Technology*, 1, 1-10, doi: https://doi.org/10.1155/2017/4939261

Beretta, L. ve Santaniello, A. (2016). Nearest Neighbor Imputation Algorithms: A Critical Evaluation. *BMC Medical Informatics and Decision Making*, 74, 197-208, doi: https://doi.org/10.1186/s12911-016-0318-z

Berkane, M. ve Bentler, P.M. (1987). Distribution of Kurtoses, with Estimators and Tests of Homogeneity of Kurtosis. *Statistics & Probability Letters*, 5, 201-207, doi: https://doi.org/10.1016/0167-7152(87)90040-X

Berndt, M. vd. (2021). Investigating Statistical Literacy and Scientific Reasoning & Argumentation in Medical, Social Sciences and Economics Students. *Learning and Individual Differences*, 86, 1-9, doi: https://doi.org/10.1016/j.lindif.2020.101963

Berners-Lee, T. ve Fischetti, M. (2000). *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. New York: HarperBusiness.

Beynon-Davies, P. (2009). Neolithic Informatics: The Nature of Information. International *Journal of Information Management*, 29(1), 3-14, doi: https://doi.org/10.1016/j.ijinfomgt.2008.11.001

Bhandari, A. (2025, Nisan 4). R-squared vs. Adjusted R-squared: Key Differences in Regression Analysis. 10.04.2025 tarihinde https://www.analyticsvidhya.com/blog/2020/07/difference-between-r-squared-and-adjusted-r-squared/ adresinden erişildi.

Bhattacharjee, P. ve Mitra, P. (2021). A Survey of Density Based Clustering Algorithms. *Frontiers of Computer Science*, 15, 1-27, doi: https://doi.org/10.1007/s11704-019-9059-3

Bi, Q. vd. (2020). RADC-Net: A Residual Attention Based Convolution Network For Aerial Scene Classification. *Neurocomputing*, 377, 345-359, doi: https://doi.org/10.1016/j.neucom.2019.11.068

Bietti, A. vd. (2019). A Kernel Perspective for Regularizing Deep Neural Networks. *ArXiv:1810.00363*, 1-20, doi: https://doi.org/10.48550/arXiv.1810.00363

Binark, İ. (1980). *Arşiv ve Arşivcilik Bilgileri*. Ankara: T.C. Başbakanlık Devlet Arşivleri Genel Müdürlüğü.

Bing, X. ve Wegkamp, M. (2023). Interpolating Discriminant Functions in High-Dimensional Gaussian Latent Mixtures. *ArXiv:2210.14347*, 1-29, doi: https://doi.org/10.48550/arXiv.2210.14347

Binici, K. (2019). Makine Öğrenmesi Yaklaşımıyla e-Belgelere Standart Dosya Plan Numaralarının Otomatik Olarak Atanması Üzerine Bir Çalışma. *Bilgi Yönetimi*, 2(2), 116-126, doi: https://doi.org/10.33721/by.654464

Biró, T.S. ve Néda, Z. (2020). Gintropy: Gini Index Based Generalization of Entropy. *Entropy*, 22(8), 1-13, doi: https://doi.org/10.3390/e22080879

Bishnoi, S. ve Hooda, B.K. (2022). Decision Tree Algorithms and Their Applicability in Agriculture for Classification. *Journal of Experimental Agriculture International*, 44(7), 20-27, doi: https://doi.org/10.9734/jeai/2022/v44i730833

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning*. New York: Springer.

Blanke, T. (2024). Reassembling Digital Archives - Strategies for Counter-Archiving. *Humanities and Social Sciences Communications*, 11, 1-12, doi: https://doi.org/10.1057/s41599-024-02668-4

Blerk, N.J. (2018). The Emergence of Law in Ancient Egypt: The Role of Maat. *Fundamina*, 24(1), 69-88, doi: https://doi.org/10.17159/2411-7870/2018/v24n1a4

Bogosavljevic, M. (2018). Digitization in Archives - Archivematica, The Practical Review of the Software for Managing Archival Records on the Example of the Personal Fonds Rista Odavic at the Archives of Serbia. *Journal of Archival Theory and Practice*, 1(1), 233-248.

Boles, F. (1982). Disrespecting Original Order. *The American Archivist*, 45(1), 26-32.

Boncelet, C. ve Bovik, A.C. (2005). Image Noise Models. A. Bovik (ed.). *Handbook of Image and Video Processing* içinde (ss. 325-335). Texas: Elsevier.

Booms, H., Joldersma, H. ve Klumpenhouwer, R. (1987). Society and the Formation of a Documentary Heritage: Issues in the Appraisal of Archival Sources. *Archivaria*, 24, 69-107.

Born, L.K. (1941). Baldassare Bonifacio and His Essay De Archivis. *The American Archivist*, 4(4), 221-237, doi: https://doi.org/10.17723/aarc.4.4.36u35457n6g45825

Bory, P. (2019). Deep New: The Shifting Narratives of Artificial Intelligence from Deep Blue to AlphaGo. *Convergence*, 25(4), 627-642, doi: https://doi.org/10.1177/1354856519829679

Boser, B.E., Guyon, I.M. ve Vapnik, V.N. (1992). A Training Algorithm for Optimal Margin Classifiers. *Proceedings of the 5th Annual Workshop on Computational Learning Theory* içinde (ss. 144-152), doi: https://doi.org/10.1145/130385.130401

Bostrom N. ve Yudkowsky, E. (2014). The Ethics of Artificial Intelligence. K. Frankish ve W.M. Ramsey (eds.). *The Cambridge Handbook of Artificial Intelligence* içinde (ss. 316-334). Cambridge: Cambridge University Press, doi: https://doi.org/10.1017/CBO9781139046855.020

Bouckaert, R.R. (2004). Naive Bayes Classifiers that Perform Well with Continuous Variables. Webb, G.I., Yu, X. (eds.). *Lecture Notes in Computer Science* içinde (ss. 1089-1094), 3339, Berlin: Springer, doi: https://doi.org/10.1007/978-3-540-30549-1_106

Bouman, R., Bukhsh, Z. ve Heskes, T. (2024). Unsupervised Anomaly Detection Algorithms on Real-world Data: How Many Do We Need? *Journal of Machine Learning Research*, 25, 1-34.

Bouslah, A. ve Taleb, N. (2020). A Genetic Approach Wrapped Support Vector Machine for Feature Selection Applied to Parkinson's Disease Diagnosis. *International Journal of Informatics and Applied Mathematics*, 3(1), 54-69.

Boxue, Z. vd. (2018). AlphaMEX: A Smarter Global Pooling Method for Convolutional Neural Networks. *Neurocomputing*, 321, 36-48, doi: https://doi.org/10.1016/j.neucom.2018.07.079

Boyko, A. vd. (2012). The BagIt File Packaging Format (v0.97). *Network Working Group*, 1-26, 10.03.2025 tarihinde https://www.digitalpreservation.gov/documents/bagitspec.pdf adresinden erişildi.

Bradley, A.P. (1997). The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms. *Pattern Recognition*, 30(7), 1145-1159, doi: https://doi.org/10.1016/S0031-3203(96)00142-2

Brady, J. ve Brockmeier, D.R. (2021). Bias-Variance Tradeoff: A Property-Casualty Modeler's Perspective. *Casualty Actuarial Society*, 13(2), 207-232.

Bramer, M. (2016). *Principles of Data Mining 2nd ed*. London: Springer.

Bre, F., Gimenez, J.M. ve Fachinotti, V.D. (2018). Prediction of Wind Pressure Coefficients on Building Surfaces Using Artificial Neural Networks. *Energy and Buildings*, 158, 1429-1441, doi: https://doi.org/10.1016/j.enbuild.2017.11.045

Breen, D.E. vd. (2023). Specimen Outlining: A Computational Archival Science Approach. *2023 IEEE International Conference on Big Data*, 2004-2009, doi: 10.1109/BigData59044.2023.10386985

Breiman, L. vd. (1984). Classification and Regression Trees (1st ed.). Boca Raton: Chapman and Hall/CRC, doi: https://doi.org/10.1201/9781315139470

Brenneke, A. (1968). *Archivistica: Contributo Alla Teoria Ed Alla Storia Archivistica Europea*. Milano: Giuffrè.

Bridgman, P.W. ve Holton, G. (2024). Science. *McGraw Hill AccessScience* içinde. 01.10.2024 tarihinde https://www.accessscience.com/content/article/a607100 adresinden erişildi.

Bulusu, S. vd. (2021). Generalization Capabilities of Translationally Equivariant Neural Networks. *ArXiv:2103.14686*, 1-28, doi: https://doi.org/10.48550/arXiv.2103.14686

Bunn, J. (2016). Archival Description and Automation: A Brief History of Going Digital. *Archives and Records*, 37(1), 65-78, doi: https://doi.org/10.1080/23257962.2016.1145577

Burki, T. (2024). Crossing the Frontier: The First Global AI Safety Summit. *The Lancet Digital Health*, 6(2), e91-e92, doi: https://doi.org/10.1016/S2589-7500(24)00001-3

Bussard, A.E. (2005). A Scientific Eevolution? *EMBO Reports*, 6(8), 691-694, doi: https://doi.org/10.1038/sj.embor.7400497

Camacho, R.S. ve Grané, M. (2023). Computational Thinking Programs in Primary Education: A Systematic Review. *Digital Education Review*, 44, 133-145, doi: https://doi.org/10.1344/der.2023.44.133-145

Canny, J. (2009). A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), 679-698, doi: https://doi.org/10.1109/TPAMI.1986.4767851

Cao, Q. vd. (2018). VGGFace2: A Dataset for Recognising Faces Across Pose and Age. *Proceedings of 13th IEEE International Conference on Automatic Face & Gesture Recognition*, 67-74, doi: https://doi.org/10.1109/FG.2018.00020

Cao, R. vd. (2021). Self-Attention-Based Deep Feature Fusion for Remote Sensing Scene Classification. *IEEE Geoscience and Remote Sensing Letters*, 18(1), 43-47, doi: 10.1109/LGRS.2020.2968550

Capek, M. (1968). The Second Scientific Revolution. *Diogenes*, 16(63), 114-133, doi: https://doi.org/10.1177/039219216801606306

Carter, J., Brown, M. ve Simpson, K. (2017). From the Classroom to the Workplace: How Social Science Students are Learning to do Data Analysis for Real. *Statistics Education Research Journal*, 16(1), 80-101, doi: https://doi.org/10.52041/serj.v16i1.218

Carter, K.S. vd. (2022). Using AI and ML to Optimize Information Discovery in Under-Utilized, Holocaust-Related Records. *AI & Society*, 37, 837-858, doi: https://doi.org/10.1007/s00146-021-01368-w

Carter, R.G.S. (2006). Of Things Said and Unsaid: Power, Archival Silences and Power in Silence. *Archivaria*, 61, 215-233.

Casadevall, A., ve Fang, F.C. (2016). Revolutionary Science. *American Society for Microbiology*, 7(2), 1-6, doi: https://doi.org/10.1128/mbio.00158-16

Casenave, J. ve Kane, A. (2024). Ethics And Deontology in Archival Practice: Uses and Epistemological Issues. *Knowledge Organization*, 51(4), 249-261, doi: http://dx.doi.org/10.5771/0943-7444-2024-4-249

Casinillo, L. ve Tavera, G. (2023). On Regression Modeling for Students' Attitude towards Statistics Online Learning in Higher Education. *St. Theresa Journal of Humanities and Social Sciences*, 9(2), 73-87.

Cassisi, C. vd. (2012). Similarity Measures and Dimensionality Reduction Techniques for Time Series Data Mining. A. Karahoca (ed.). *Advances in Data Mining Knowledge Discovery and Applications* içinde (ss. 71-96). InTechOpen, doi: 10.5772/49941

Caswell, M. (2023). "The Archive" is not an Archives: Acknowledging the Intellectual Contributions of Archival Studies. A. Prescott ve A. Wiggins (eds.), *Archives: Power, Truth and Fiction* içinde (ss. 21-36), New York: Oxford University Press, doi: https://doi.org/10.1093/oxfordhb/9780198829324.013.0002

Chan, Y.H. (2003). User's Guide to Correlation Coefficients. *Singapore Medical Journal*, 44(12), 614-619.

Chandak, Y. vd. (2019). Learning Action Representations for Reinforcement Learning. *ArXiv:1902.00183*, 1-18, doi: https://doi.org/10.48550/arXiv.1902.00183

Chander, S. ve Vijaya, P. (2021). Unsupervised Learning Methods for Data Clustering. D. Binu ve B.R. Rajakuma (eds.). *Artificial Intelligence in Data Mining: Theories and Applications* içinde (ss. 41-64), London: Academic Press, doi: https://doi.org/10.1016/B978-0-12-820601-0.00002-1

Chaple, G.N., Daruwala, R.D. ve Gofane, M.S. (2015). Comparisions of Robert, Prewitt, Sobel Operator Based Edge Detection Methods for Real Time Uses on FPGA. *2015 International Conference on Technologies for Sustainable Development*, 1-4, doi: 10.1109/ICTSD.2015.7095920

Chapman, A. vd. (2024). Supporting Better Insights of Data Science Pipelines with Fine-Grained Provenance. *ACM Transactions on Database Systems*, 49(2), 1-42, doi: https://doi.org/10.1145/3644385

Chazan, M., Baldwin, M. ve Madokoro, L. (2015). Aging, Activism, and the Archive: Feminist Perspectives for the 21st Century. *Archivaria*, 80, 59-87.

Chehreghani, M.H. (2024). The Embeddings World and Artificial General Intelligence. *Cognitive Systems Research*, 84, 1-6, doi: https://doi.org/10.1016/j.cogsys.2023.101201

Chen, C., Isa, N.A.M. ve Liu, X. (2025). A review of Convolutional Neural Network Based Methods for Medical Image Classification. *Computers in Biology and Medicine*, 185, 1-41, doi: https://doi.org/10.1016/j.compbiomed.2024.109507

Chen, J. vd. (2018). Remote Sensing Scene Classification Based on Convolutional Neural Networks Pre-Trained Using Attention-Guided Sparse Filters. *Remote Sensing*, 10(2), 1-16, doi: https://doi.org/10.3390/rs10020290

Chen, T. vd. (2022). Learning to Optimize: A Primer and a Benchmark. *Journal of Machine Learning* Research, 23, 1-59, doi: https://doi.org/10.48550/arXiv.2103.12828

Chen, W. vd. (2024). Leveraging Machine Learning for Sophisticated Rental Value Predictions: A Case Study from Munich, Germany. *Applied Sciences*, 14(20), 1-26, doi: https://doi.org/10.3390/app14209528

Chen, X. (2023). Stock Price Prediction Using Machine Learning Strategies. *BCP Business & Management*, 36, 488-497, doi: https://doi.org/10.54691/bcpbm.v36i.3507

Cheng, G. vd. (2018). When Deep Learning Meets Metric Learning: Remote Sensing Image Scene Classification via Learning Discriminative CNNs. *IEEE Transactions on Geoscience and Remote Sensing*, 56(5), 2811-2821, doi: 10.1109/TGRS.2017.2783902

Chiang, H.E. (2016). Skewness and Coskewness in Bond Returns. *The Journal of Financial Research*, 39(2), 145-178, doi: 10.1111/jfir.12093

Chittora, A. ve Mishra, O. (2012). Face Recognition Using RBF Kernel Based Support Vector Machine. *International Journal of Future Computer and Communication*, 1(3), 280-283.

Cho, K. vd. (2014). On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *ArXiv:1409.1259*, 1-9, doi: https://doi.org/10.48550/arXiv.1409.1259

Choi, C. vd. (2022). Big-Data-Based Text Mining and Social Network Analysis of Landscape Response to Future Environmental Change. *Land*, 11(12), 1-21, doi: https://doi.org/10.3390/land11122183

Cibaroğlu, M.O. ve Yalçınkaya, B. (2019). Belge ve Arşiv Yönetimi Süreçlerinde Büyük Veri Analitiği ve Yapay Zekâ Uygulamaları. *Bilgi Yönetimi*, 2(1), 44-58, doi: https://doi.org/10.33721/by.570634

Ciechanowski, H.M. (2021). The Principle of Provenance as the Principle of Rationalization of Archives. *Archiwa*, 12(14), 49-65, doi: https://doi.org/10.12775/AKZ.2021.003

Clark, W.R., Golder, M. ve Golder, S.N. (2018). *Principles of Comparative Politics, 3rd ed.* California: SAGE Publications.

Clevert, D.A., Unterthiner, T. ve Hochreiter, S. (2016). Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *ArXiv:1511.07289*, 1-14, doi: https://doi.org/10.48550/arXiv.1511.07289

Cochard, J.L. (2021). Linked Data at the Swiss Federal Archives. 06.03.2025 tarihinde https://ceur-ws.org/Vol-3019/LinkedArchives_2021_paper_21.pdf adresinden erişildi.

Cohen, J.P. ve Lo, H.Z. (2014). Academic Torrents: A Community-Maintained Distributed Repository. *Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment*, 2, 1-2, doi: https://doi.org/10.1145/2616498.2616528

Colavizza, G. vd. (2021). Archives and AI: An Overview of Current Debates and Future Perspectives. *ACM Journal on Computing and Cultural Heritage*, 15(1), 1-15, doi: https://doi.org/10.1145/3479010

Colmerauer, A. vd. (1972). *A Communication System in French*. Preliminary Report at the end of the IRIA Contract, Artificial Intelligence Group, Faculty of Sciences of Luminy, Aix-Marseille II University.

Computational Archival Science Research Group. (2019). Toward a Computational Framework for Library and Archival Education: Report on Preliminary Literature and Curriculum Review. 06.03.2025 tarihinde https://ai-collaboratory.net/wp-content/uploads/2020/04/imlsCASresearchGroupReport.pdf adresinden erişildi.

Conca, J. (2014, 3 Ağustos). Science is not Democratic. 03.10.2024 tarihinde https://www.forbes.com/sites/jamesconca/2014/08/03/science-is-not-democratic/ adresinden erişildi.

Conway, P. (2010). Preservation in the Age of Google: Digitization, Digital Preservation, and Dilemmas. *The Library Quarterly: Information, Community, Policy*, 80(1), 61-79, doi: https://doi.org/10.1086/648463

Cook, T. (2001). Archival Science and Postmodernism: New Formulations for Old Concepts. *Archival Science*, 1, 3-24, doi: https://doi.org/10.1007/BF02435636

Cook, T. (2011a). We Are What We Keep; We Keep What We Are: Archival Appraisal Past, Present and Future. *Journal of the Society of Archivists*, 32(2), 173-189, doi: https://doi.org/10.1080/00379816.2011.619688

Cook, T. (2011b). The Archive(s) is a Foreign Country: Historians, Archivists, and the Changing Archival Landscape. *The American Archivist*, 74(2), 600-632.

Cook, T. (2013). Evidence, Memory, Identity and Community: Four Shifting Archival Paradigms. *Archival Science*, 13, 95-120, doi: https://doi.org/10.1007/s10502-012-9180-7

Cornell Bowers. (2024). Convolutional Neural Networks, CS4782: Intro to Deep Learning. 17.05.2024 tarihinde https://www.cs.cornell.edu/courses/cs4782/2024sp/lectures/pdfs/week_2_0.pdf adresinden erişildi.

Cortes, C. ve Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20, 273-297, doi: https://doi.org/10.1007/BF00994018

Cowan, L.M. (2018). *Decolonizing Provenance: An Examination of Types of Provenance and their Role in Archiving Indigenous Records in Canada*. Yüksek Lisans Tezi. Lisansüstü Çalışmalar Enstitüsü, Manitoba Üniversitesi, Kanada.

Cox, R.J. (2007). Machines in the Archives: Technology and the Coming Transformation of Archival Reference. *First Monday*, 12(11), doi: https://doi.org/10.5210/fm.v12i11.2029

Curcin V. (2016). Embedding Data Provenance into the Learning Health System to Facilitate Reproducible Research. *Learning Health Systems*, 1(2), 1-12, doi: https://doi.org/10.1002/lrh2.10019

Cutler, A., Cutler, D.R. ve Stevens, J.R. (2012). Random Forests. C. Zhang ve Y. Ma (eds.). *Ensemble Machine Learning* içinde (ss. 157-175), New York: Springer, doi: https://doi.org/10.1007/978-1-4419-9326-7_5

Çiçek Akgün, A. ve Çiçek, N. (2022). Dijital Çağda Değişen Belge Olgusunun Arşivcilikte Düzenleme ve Tanımlamaya Etkisi: Literatüre Dayalı Bir İnceleme. *Bilgi ve Belge Araştırmaları*, (17), 33-58, doi: https://doi.org/10.26650/bba.2022.17.1127615

Dada, E.G. vd. (2019). Machine Learning for e-Mail Spam Filtering: Review, Approaches and Open Research Problems. *Heliyon*, 5(6), 1-23, doi: https://doi.org/10.1016/j.heliyon.2019.e01802

Dahlström, K.M. ve Salminen, T.A. (2024). Apprehensions and Emerging Solutions in ML-Based Protein Structure Prediction. *Current Opinion in Structural Biology*, 86, 1-6, doi: https://doi.org/10.1016/j.sbi.2024.102819

Dambrot, S.M. (2020). Theoretical and Hypothetical Pathways to Real-Time Neuromorphic AGI/Post-AGI Ecosystems. *Procedia Computer Science*, 169, 110-122, doi: https://doi.org/10.1016/j.procs.2020.02.122

Dani A. ve Al Quraan, E. (2023). Investigating Research Students' Perceptions about Statistics and its Impact on Their Choice of Research Approach. *Heliyon*, 9(10), 1-10, doi: 10.1016/j.heliyon.2023.e20423

Datta, A. vd. (2024). Survey on Deep Learning Technique on Maize Leaves Infected by Fall Armyworms. D.K. Kole, S.R. Chowdhury, S. Basu, D. Plewczynski, D. Bhattacharjee (eds.). *Proceedings of 4th International Conference on Frontiers in Computing and Systems*, 975 içinde (ss. 521-530). Singapore: Springer, doi: https://doi.org/10.1007/978-981-97-2614-1_36

Davet, J., Hamidzadeh, B. ve Franks, P. (2023). Archivist in the Machine: Paradata for AI-Based Automation in the Archives. *Archival Science*, 23, 275-295, doi: https://doi.org/10.1007/s10502-023-09408-8

Degadwala, S. ve Vyas, D. (2024). Systematic Analysis of Deep Learning Models vs. Machine Learning. *International Journal of Scientific Research in Computer Science Engineering and Information Technology*, 10(4), 60-70, doi: https://doi.org/10.32628/CSEIT24104108

Delmas, B. (1993). Archival Science and Information Technologies. A. Menne-Haritz (ed.). *Information Handling in Offices and Archives* içinde (ss. 168-176). München: K.G. Saur, doi: https://doi.org/10.1515/9783111503363.168

Delmas, B. (2001). Archival Science Facing the Information Society. *Archival Science*, 1, 25-37, doi: https://doi.org/10.1007/BF02435637

Deng, B.C. vd. (2015). A New Strategy to Prevent Over-Fitting in Partial Least Squares Models Based on Model Population Analysis. *Analytica Chimica Acta*, 880, 32-41, doi: http://dx.doi.org/10.1016/j.aca.2015.04.045

Denning, P.J. ve Tedre, M. (2021). Computational Thinking: A Disciplinary Perspective. *Informatics in Education*, 20(1), 361-390, doi: 10.15388/infedu.2021.21

Deprez, M. ve Robinson, E.C. (2024). *Machine Learning for Biomedical Applications*. London: Elsevier, doi: https://doi.org/10.1016/C2019-0-04088-5

Derras, B. ve Makhoul, N. (2022). An Overview of the Infrastructure Seismic Resilience Assessment Using Artificial Intelligence and Machine-Learning Algorithms. *Proceedings of 3rd International Conference on Natural Hazards & Infrastructure*, 1-12.

Dey, N. ve Williams, J.P. (2024). Valid Inference for Machine Learning Model Parameters. *ArXiv:2302.10840*, 1-35, doi: https://doi.org/10.48550/arXiv.2302.10840

Dickerson, M. ve Yun, A.E. (2023). Critical Community-Centeredness: Ethical Considerations for Computational Archival Studies. *2023 IEEE International Conference on Big* Data, 2032-2035, doi: https://doi.ieeecomputersociety.org/10.1109/BigData59044.2023.10386092

Ding, B. vd. (2024). Data Augmentation using LLMs: Data Perspectives, Learning Paradigms and Challenges. *Findings of the Association for Computational Linguistics*, 1679-1705, doi: https://doi.org/10.18653/v1/2024.findings-acl.97

Ding, H. vd. (2016). k-Means Clustering with Distributed Dimensions. *Proceedings of the 33rd International Conference on Machine Learning*, 48, 1339-1348.

Ding, X. vd. (2021). Random Radial Basis Function Kernel-Based Support Vector Machine. *Journal of the Franklin Institute*, 358(18), 10121-10140, doi: https://doi.org/10.1016/j.jfranklin.2021.10.005

Douglas, J. (2013). What We Talk About When We Talk About Original Order in Writers' Archives. *Archivaria*, 76, 7-25.

Douglas, J. (2017). Origins: Evolving Ideas about the Principle of Provenance. T. Eastwood ve H. MacNeil (eds.), *Currents of Archival Thinking 2nd ed.* içinde (ss. 25-52). Santa Barbara: Libraries Unlimited.

Dove, S. vd. (2023). A User-Friendly Guide to Using Distance Measures to Compare Time Series in Ecology. *Ecology and Evolution*, 13(10), 1-32, doi: https://doi.org/10.1002/ece3.10520

Drake J.M. (2016). RadTech Meets RadArch: Towards a New Principle for Archives and Archival Description. 09.12.2024 tarihinde https://medium.com/on-archivy/radtech-meets-radarch-towards-a-new-principle-for-archives-and-archival-description-568f133e4325 adresinden erişildi.

Drenth, P.J.D. (2004). Scientific Values: How Universal? International Symposium on Universal Values, Academy of Athens, May 26-28, 237-246, 01.10.2024 tarihinde https://www.allea.org/wp-content/uploads/2016/02/Drenth_Scientific_Values.pdf adresinden erişildi.

Drezner, Z., Turel, O. ve Zerom, D. (2010). A Modified Kolmogorov-Smirnov Test for Normality. *Communications in Statistics - Simulation and Computation*, 39(4), 693-704, doi: https://doi.org/10.1080/03610911003615816

Du, K.L. vd. (2022). Perceptron: Learning, Generalization, Model Selection, Fault Tolerance, and Role in the Deep Learning Era. *Mathematics*, 10(24), 1-46, doi: https://doi.org/10.3390/math10244730

Du, K.L. vd. (2024). Exploring Kernel Machines and Support Vector Machines: Principles, Techniques, and Future Directions. *Mathematics*, 12(24), 1-58, doi: https://doi.org/10.3390/math12243935

Dubey, S.R., Singh, S.K. ve Chaudhuri, B.B. (2022). Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark. *Neurocomputing*, 503, 92-108, doi: https://doi.org/10.1016/j.neucom.2022.06.111

Duchein, M. (1983). Theoretical Principles and Practical Problems of Respect des Fonds in Archival Science. *Archivaria*, 16, 64-82.

Duranti, L. (1994). The Concept of Appraisal and Archival Theory. *The American Archivist*, 57(2), 328-344, doi: https://doi.org/10.17723/aarc.57.2.pu548273j5j1p816

Duranti, L. (1999). Concepts and Principles for the Management of Electronic Records, or Records Management Theory is Archival Diplomatics. *Records Management Journal*, 9(3), 149-171, doi: https://doi.org/10.1108/EUM0000000007248

Duranti, L. (2001). The Impact of Digital Technology on Archival Science. *Archival Science* 1, 39-55, doi: https://doi.org/10.1007/BF02435638

Duranti, L. ve Franks, P.C. (2015a). Archival Science. *Encyclopedia of Archival Science* içinde (1. bs., s. 84).). New York: Rowman & Littlefield.

Duranti, L. ve Franks, P.C. (2015b). Principle of Provenance. *Encyclopedia of Archival Science* içinde (1. bs., s. 285). New York: Rowman & Littlefield.

Duranti, L. ve MacNeil, H. (1996). The Protection of the Integrity of Electronic Records: An Overview of the UBC-MAS Research Project. *Archivaria*, 42, 46-67.

Duranti, L. ve Thibodeau, K. (2006). The Concept of Record in Interactive, Experiential and Dynamic Environments: the View of InterPARES. *Archival Science*, 6(1), 13-68, doi: https://doi.org/10.1007/s10502-006-9021-7

Eastwood, T. (1994). What is Archival Theory and Why is it Important? *Archivaria*, 37, 122-130.

Eastwood, T. (2004). Jenkinson's Writings on Some Enduring Archival Themes. *The American Archivist*, 67(1), 31-44.

Edgar, T.W. ve Manz, D.O. (2017). *Research Methods for Cyber Security*. Massachusetts: Elsevier Syngress, doi: https://doi.org/10.1016/B978-0-12-805349-2.00007-8

El Hadi, M.M. (2023). Artifical Intelligence Background, Definnitions, Challenges and Benefits. *CompuNet*, 31, 5-13.

Elharrouss, O. vd. (2025). Loss Functions in Deep Learning: A Comprehensive Review. *ArXiv:2504.04242*, 1-36, doi: https://doi.org/10.48550/arXiv.2504.04242

Elith, J., Leathwick, J.R. ve Hastie, T. (2008). A Working Guide to Boosted Regression Trees. *Journal of Animal Ecology*, 77, 802-813, doi: https://doi.org/10.1111/j.1365-2656.2008.01390.x

Ellahi, A. vd. (2023). Applying Text Mining and Semantic Network Analysis to Investigate Effects of Perceived Crowding in the Service Sector. *Cogent Business & Management*, 10(2), 1-18, doi: https://doi.org/10.1080/23311975.2023.2215566

Elmas, Ç. (2021). *Yapay Zekâ Uygulamaları*. Ankara: Seçkin.

Epperlein, J.P. vd. (2022). Reinforcement Learning with Algorithms from Probabilistic Structure Estimation. *Automatica*, 144, 1-12, doi: https://doi.org/10.1016/j.automatica.2022.110483

Esposito, F. vd. (1997). A Comparative Analysis of Methods for Pruning Decision Trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5), 476-491, doi: 10.1109/34.589207

Ester, M. vd. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 226-231,

European Commission. (2018). A Definition of AI: Main Capabilities and Scientific Disciplines. *The European Commission's High-Level Expert Group on Artificial Intelligence*, 1-7, 18.09.2023 tarihinde https://ec.europa.eu/futurium/en/system/files/ged/ai_hleg_definition_of_ai_18_december_1.pdf adresinden erişildi.

Ezugwu, A.E. vd. (2022). A Comprehensive Survey of Clustering Algorithms: State-of-the-Art Machine Learning Applications, Taxonomy, Challenges, and Future Research Prospects. *Engineering Applications of Artificial Intelligence*, 110, 1-43, doi: https://doi.org/10.1016/j.engappai.2022.104743

Faber, H.H. vd. (2017). Teaching Computational Thinking to Primary School Students via Unplugged Programming Lessons. *Journal of the European Teacher Education Network*, 12, 14-24.

Farris, F.A. (2010). The Gini Index and Measures of Inequality. *The American Mathematical Monthly*, 117(10), 851-864, doi: https://doi.org/10.4169/000298910X523344

Fazi, M.B. (2021). Beyond Human: Deep Learning, Explainability and Representation. *Theory, Culture & Society*, 38(7-8), 55-77, doi: https://doi.org/10.1177/0263276420966386

Fei, N. vd. (2022). Towards Artificial General Intelligence via a Multimodal Foundation Model. *Nature Communications*, 13(1), 1-13, doi: https://doi.org/10.1038/s41467-022-30761-2

Feinman, R. ve Lake, B.M. (2019). Learning a Smooth Kernel Regularizer for Convolutional Neural Networks. *ArXiv:1903.01882*, 1-7, doi: https://doi.org/10.48550/arXiv.1903.01882

Feng, H. vd. (2021). Retrospect and Prospect: The Research Landscape of Archival Studies. *Archival Science*, 21, 391-411, doi: https://doi.org/10.1007/s10502-021-09364-1

Fırat, M., Yıltaş-Kaplan, D. ve Şamlı, R. (2021). Forecasting Air Travel Demand for Selected Destinations Using Machine Learning Methods. *Journal of Universal Computer Science*, 27(6), 564-581, doi: 10.3897/jucs.68185

Floridi, L. (2020). AI and its New Winter: From Myths to Realities. *Philosophy & Technology*, 33, 1-3, doi: https://doi.org/10.1007/s13347-020-00396-6

Fonseca, J., Douzas, G. ve Bacao, F. (2021). Improving Imbalanced Land Cover Classification with k-Means Smote: Detecting and Oversampling Distinctive Minority Spectral Signatures. *Information*, 12(7), 1-20, doi: https://doi.org/10.3390/info12070266

Foscarini, F. (2017). Archival Appraisal in Four Paradigms. H. MacNeil, T. Eastwood (eds.). *Currents of Archival Thinking* içinde (ss. 107-133). Santa Barbara: Libraries Unlimited.

François, D., Wertz, V. ve Verleysen, M. (2011). Choosing the Metric: A Simple Model Approach. N. Jankowski, W. Duch, K. Grabczewski (eds.). *Meta-Learning in Computational Intelligence* içinde (ss. 97-115), Berlin: Springer, doi: https://doi.org/10.1007/978-3-642-20980-2_3

Frank, R.D. (2020). The Social Construction of Risk in Digital Preservation. *Journal of the Association for Information Science and Technology*, 71(4), 474-484, doi: https://doi.org/10.1002/asi.24247

Franke, M. ve Degen, J. (2023). The Softmax Function: Properties, Motivation and Interpretation, 1-23, doi: https://doi.org/10.31234/osf.io/vsw47

Fuhrmeister, C. ve Hopp, M. (2019). Rethinking Provenance Research. *Getty Research Journal*, 11, 213-231.

Gajowniczek, K. vd. (2020). Semantic and Generalized Entropy Loss Functions for Semi-Supervised Deep Learning. *Entropy*, 22, 1-17, doi: 10.3390/e22030334

Ganie, A.G. ve Dadvadipour, S. (2024). Exploring Noise-Induced Techniques to Strengthen Deep Learning Model Resilience. *Pollack Periodica*, 19(3), 8-13, doi: https://doi.org/10.1556/606.2024.01068

Garbin, C., Zhu, X. ve Marques, O. (2020). Dropout vs. Batch Normalization: An Empirical Study of Their Impact to Deep Learning. *Multimedia Tools and Applications*, 79, 12777-12815, doi: https://doi.org/10.1007/s11042-019-08453-9

Garisto, D. (2019, 30 Kasım). Google AI Beats Top Human Players at Strategy Game Starcraft II. *Nature*. 14.01.2025 tarihinde https://linkcuts.com/nature adresinden erişildi.

Gbeassor, L.A. (2024). Data as the New Oil: Extractivism in Neptune Frost. *Matatu*, 55(2), 249-277, doi: https://doi.org/10.1163/18757421-bja00011

Gedikli, M.E. (2023). *Elektronik Belge Yönetim Sistemlerinde Veri Madenciliği ve Makine Öğrenmesi Yöntemlerinin Kullanımı*. Marmara Üniversitesi Türkiyat Araştırmaları Enstitüsü, İstanbul.

Geman, S., Bienenstock, E. ve Doursat, R. (1992). Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, 4(1), 1-58.

Geng, J. vd. (2020). Support Vector Machine Regression (SVR)-Based Nonlinear Modeling of Radiometric Transforming Relation for the Coarse-Resolution Data-Referenced Relative Radiometric Normalization (RRN). *Geo-Spatial Information Science*, 23(3), 237-247, doi: https://doi.org/10.1080/10095020.2020.1785958

Geron, A. (2021). *Scikit-Learn, Keras ve Tensorflow ile Uygulamalı Makine Öğrenmesi*. (Çev. B. Aksoy, Ö. Kaya). Ankara: Buzdağı Yayınevi. (Orijinal yayın tarihi, 2019).

Ghareeb, S. vd. (2022). Evaluating Student Levelling Based on Machine Learning Model's Performance. *Discovery Internet of Things*, 2(3), 1-25, doi: https://doi.org/10.1007/s43926-022-00023-0

Ghasemi, M. ve Ebrahimi, D. (2024). Introduction to Reinforcement Learning. *ArXiv:2408.07712*, 1-19, doi: https://doi.org/10.48550/arXiv.2408.07712

Gholamalinezhad, H. ve Khosravi, H. (2020). Pooling Methods in Deep Neural Networks, a Review. *ArXiv:2009.07485*, 1-16, doi: https://doi.org/10.48550/arXiv.2009.07485

Giannini, T. ve Bowen, J.P. (2022). Computational Culture: Transforming Archives Practice and Education for a Post-Covid World. *Journal on Computing and Cultural Heritage*, 15(3), 1-18, doi: https://doi.org/10.1145/3493342

Gill, K.S. (2016). Artificial Super Intelligence: Beyond Rhetoric. *AI & Society*, 31, 137-143, doi: https://doi.org/10.1007/s00146-016-0651-x

Gobble, M.M. (2019). The Road to Artificial General Intelligence. *Research-Technology Management*, 62(3), 55-59, doi: https://doi.org/10.1080/08956308.2019.1587336

Goertzel, B. (2014). Artificial General Intelligence: Concept, State of the Art and Future Prospects. *Journal of Artificial General Intelligence*, 5(1) 1-46, doi: 10.2478/jagi-2014-0001

Goh, K.W. vd. (2024). Comparison of Activation Functions in Convolutional Neural Network for Poisson Noisy Image Classification. *Emerging Science Journal*, 8(2), 592-602, doi: http://dx.doi.org/10.28991/ESJ-2024-08-02-014

Goodfellow, I., Bengio, Y. ve Courville, A. (2018). *Derin Öğrenme*. (Çev. F.Yarman Vural, R.G. Cinbiş, S. Kalkan). Ankara: Buzdağı Yayınevi. (Orijinal yayın tarihi, 2016).

Goudarouli, E. (2023). Digital Innovation and Archival Thinking. A.Prescott, A.Wiggins (eds.). *Archives: Power, Truth, and Fiction* içinde (ss. 267-281), Oxford: Oxford University Press, doi: https://doi.org/10.1093/oxfordhb/9780198829324.013.0019

Govidarajani, M. (2013). A Hybrid Framework using RBF and SVM for Direct Marketing. *International Journal of Advanced Computer Science and Applications*, 4(4), 121-126.

Goynov, M. vd. (2024). CultIS: Web-Based Platform for Intelligent Cultural Content Management. *Digital Presentation and Preservation of Cultural and Scientific Heritage*, 14, 19-36, doi: https://doi.org/10.55630/dipp.2024.14.1

Görentaş, M.B. ve Uçkan, T. (2023). Makine Öğrenmesi Yöntemleri Kullanılarak Mahkeme Kararlarlarının Kümelenmesi. *Journal of Computer Science*, 8(2), 148-158, doi: https://doi.org/10.53070/bbd.1318518

Granter, S.R., Beck, A.H. ve Papke, D.J. (2017). AlphaGo, Deep Learning, and the Future of the Human Microscopist. *Archives of Pathology & Laboratory Medicine*, 141(5), 619-621, doi: https://doi.org/10.5858/arpa.2016-0471-ED

Graves, A., Mohamed, A.R. ve Hinton, G. (2013). Speech Recognition with Deep Recurrent Neural Networks. *ArXiv:1303.5778*, 1-5, doi: https://doi.org/10.48550/arXiv.1303.5778

Greenacre, M. vd. (2023). Principal Component Analysis. *Economics Working Paper Series*, 1856, 1-24. 17.06.2024 tarihinde https://repositori-api.upf.edu/api/core/bitstreams/9a4459af-9a9a-4c96-8eae-d97731cbf0c6/content adresinden erişildi.

Greene, D., Cunningham, P. ve Mayer, R. (2008). Unsupervised Learning and Clustering. M. Cord ve P. Cunningham (eds.). *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval* içinde (ss. 51-87), Berlin: Springer.

Greene, M. ve Meissner, D. (2005). More Product, Less Process: Revamping Traditional Archival Processing. *The American Archivist*, 68(2), 208-263, doi: 10.17723/aarc.68.2.c741823776k65863

Groth, P. vd. (2012). Requirements for Provenance on the Web. *International Journal of Digital Curation*, 7(1), 39-56, doi: https://doi.org/10.2218/ijdc.v7i1.213

Grzymala-Busse, J.W. ve Than, S. (1993). Data Compression in Machine Learning Applied to Natural Language. *Behavior Research Methods, Instruments & Computers*, 25, 318-321, doi: https://doi.org/10.3758/BF03204518

Guimarães, J.A.C. ve Tognoli, N.B. (2015). Provenance as a Domain Analysis Approach in Archival Knowledge Organization. *Knowledge Organization*, 42(8), 562-569.

Gunter, T. vd. (2024). Apple Intelligence Foundation Language Models. *arXiv*, 2407.21075v1, 1-47, doi: https://doi.org/10.48550/arXiv.2407.21075

Guo, X. (2017). *Deep Learning and Reward Design for Reinforcement Learning*. Doktora Tezi. Computer Science and Engineering, University of Michigan.

Gündüz, G. ve Cedimoğlu, İ.H. (2019). Derin Öğrenme Algoritmalarını Kullanarak Görüntüden Cinsiyet Tahmini. *Sakarya University Journal of Computer and Information Sciences*, 2(1), 9-17, doi: https://doi.org/10.35377/saucis.02.01.517930

Hadi, M.A., Abdulredha, M.N. ve Hasan, E. (2023). Introduction to ChatGPT: A New Revolution of Artificial Intelligence with Wachine Learning Algorithms and

Cybersecurity. *International Research Journal of Current Science*, 4(4), 276-285, doi: https://doi.org/10.47587/SA.2023.4406

Haeb-Umbach, R., Beyerlein, P. ve Geller, D. (1995). Speech Recognition Algorithms for Voice Control Interfaces. *Philips Journal of Research*, 49(4), 381-397.

Hagiwara, K. (2002). Regularization Learning, Early Stopping and Biased Estimator. *Neurocomputing*, 48(1-4), 937-955, doi: https://doi.org/10.1016/S0925-2312(01)00681-6

Haigh, T. (2023). There was no 'First AI Winter'. *Communications of the ACM*, 66(12), 35-39, doi: https://doi.org/10.1145/3625833

Halder, R.K. vd. (2024). Enhancing K-Nearest Neighbor Algorithm: A Comprehensive Review and Performance Analysis of Modifications. *Journal of Big Data*, 11, 1-55, doi: https://doi.org/10.1186/s40537-024-00973-y

Haleem, A. vd. (2022). Artificial Intelligence (AI) Applications for Marketing: A Literature-Based Study. *International Journal of Intelligent Networks*, 3, 119-132, doi: https://doi.org/10.1016/j.ijin.2022.08.005

Hamdan, M.H. ve Roach, D.C. (2022). The Sigmoid Neural Network Activation Function and its Connections to Airy's and the Nield-Kuznetsov Functions. *Proof*, 2, 108-114, doi: 10.37394/232020.2022.2.13

Hamouda, H. vd. (2019). Extending the Scope of Computational Archival Science - A Case Study on Leveraging Archival and Engineering Approaches to Develop a Framework to Detect and Prevent Fake Video. *2019 IEEE International Conference on Big Data*, 3087-3097, doi: 10.1109/BigData47090.2019.9006170

Han, S. vd. (2016). DSD: Regularizing Deep Neural Networks with Dense-Sparse-Dense Training Flow. *ArXiv:1607.04381*, 1-13, doi: https://doi.org/10.48550/arXiv.1607.04381

Harris, J.K. (2021). Primer on Binary Logistic Regression. *Family Medicine and Community Health*, 9, 1-7, doi: :10.1136/fmch-2021-001290

Harris, V. (2005). Archives, Politics and Justice. M. Procter, M. Cook ve C. Williams (eds.). *Political Pressure and the Archival Record* içinde (ss. 173-183) Chicago: Society of American Archivists.

Hassan, E. vd. (2023). The Effect of Choosing Optimizasyon algoritması Algorithms to Improve Computer Vision Tasks: A Comparative Study. *Multimedia Tools and Applications*, 82, 16591-16633, doi: https://doi.org/10.1007/s11042-022-13820-0

Hayatin, N. vd. (2022). Sentiment Analysis Based on Probabilistic Classifier Techniques in Various Indonesian Review Data. *Jordanian Journal of Computers and Information Technology*, 8(3), 271-282.

He, H., Yan, H. ve Tan, V.Y.F. (2022). Information-Theoretic Characterization of the Generalization Error for Iterative Semi-Supervised Learning. *Journal of Machine Learning Research*, 23(1), 1-52.

He, K. vd. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *arXiv:1502.01852*, 1-11, doi: https://doi.org/10.48550/arXiv.1502.01852

He, K. vd. (2015). Deep Residual Learning for Image Recognition. *ArXiv:1512.03385*, 1-12, doi: https://doi.org/10.48550/arXiv.1512.03385

Hedstrom, M. (1997). Digital Preservation: A Time Bomb for Digital Libraries. *Computers and the Humanities*, 31(3), 189-202.

Henderson, H. (2009). Artificial Intelligence. *Encyclopedia of Computer Science and Technology* içinde (Gözden Geçirilmiş Basım, ss. 26-28). New York: Infobase Publishing.

Hendrycks, D. ve Gimpel, K. (2016). Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units. *ArXiv:1606.08415*, 1-6, doi: https://doi.org/10.48550/arXiv.1606.08415

Heßler, M. (2017). The Triumph of "Stupidity" : Deep Blue`s Victory over Garri Kasparov. The Controversy about its Impact on Artficial Intelligence Research. *Naturwissenschaften, Technik und Medizin*, 25(1), 1-33, doi: https://doi.org/10.1007/s00048-017-0167-6

Hilbe, J.M. (2015). *Practical Guide to Logistic Regression*. Boca Raton: CRC Press.

Hinsche, R. (2024). Blackwell Customer Deck. 16.01.2025 tarihinde https://mcomputers.cz/wp-content/uploads/2024/04/NVIDIA-Blackwell-Praha-PDF.pdf adresinden erişildi.

Hinton, G.E., Osindero, S. ve Teh, Y.W. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation* 18, 1527-1554.

Ho, C.H. ve Lin, C.J. (2012). Large-Scale Linear Support Vector Regression. *Journal of Machine Learning Research*, 13, 3323-3348.

Hochreiter, S. ve Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735-1780, doi: https://doi.org/10.1162/neco.1997.9.8.1735

Hoerl, A.E. ve Kennard, R.W. (2000). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 42(1), 80-86, doi: https://doi.org/10.2307/1271436

Hohmann, P. (2016). On Impartiality and Interrelatedness: Reactions to Jenkinsonian Appraisal in the Twentieth Century. *The American Archivist*, 79(1), 14-25, doi: https://doi.org/10.17723/0360-9081.79.1.14

Holmes, O. (1961) Sir Hilary Jenkinson, 1882-1961. *The American Archivist*, 24(3), 345-347, doi: https://doi.org/10.17723/aarc.24.3.e521g515xp0n60u1

Hoque, Z. vd. (2024). Stain Normalization Methods for Histopathology Image Analysis: A Comprehensive Review and Experimental Comparison. *Information Fusion*, 1-22, doi: https://doi.org/10.1016/j.inffus.2023.101997

Hornby, A.S. (1995). Science. *Oxford Advanced Learner's Dictionary of Current English* içinde (5.bs., s. 1050). Oxford: Oxford University Press.

Hosna, A. vd. (2022). Transfer Learning: A Friendly Introduction. *Journal of Big Data*, 9(1), 1-19, doi: https://doi.org/10.1186/s40537-022-00652-w

Horsman, P. (1994). Taming the Elephant: An Orthodox Approach to the Principle of Provenance. A. Kerstin ve J. Sydbeck (eds.). *The Principle of Provenance: Report from the First Stockholm Conference on the Archival of Principle of Provenance* içinde (ss. 51-63), Stockholm: Swedish National Archives.

Howarth, J. (2024, 6 Ağustos). Number of Parameters in GPT-4 (Latest Data). 06.01.2025 tarihinde https://explodingtopics.com/blog/gpt-parameters adresinden erişildi.

Hölldobler, S., Möhle, S. ve Tigunova, A. (2017). Lessons Learned from AlphaGo. S. Hölldobler, A. Malikov ve C. Wernhard (eds.). *YSIP2 Proceedings of the Second Young Scientist's International Workshop on Trends in Information Processing* içinde (ss.92-

101), Dombai, Rusya, 16-20 Mayıs 2017. 06.01.2025 tarihinde https://ceur-ws.org/Vol-1837/paper14.pdf adresinden erişildi.

Hu, L. vd. (2021). KR-DBSCAN: A Density-Based Clustering Algorithm Based on Reverse Nearest Neighbor and Influence Space. *Expert Systems with Applications*, 186, 1-8, doi: https://doi.org/10.1016/j.eswa.2021.115763

Hu, L.Y. vd. (2016). The Distance Function Effect on k-Nearest Neighbor Classification for Medical Datasets. *SpringerPlus*, 5, 1-9, doi: https://doi.org/10.1186/s40064-016-2941-7

Hu, Z., Zhang, J. ve Ge, A.Y. (2021). Handling Vanishing Gradient Problem Using Artificial Derivative. *IEEE Access*, 9, 22371-22377, doi: 10.1109/ACCESS.2021.3054915

Hutchinson, T. (2018). Protecting Privacy in the Archives: Supervised Machine Learning and Born Digital Records. *2018 IEEE International Conference on Big Data*, 2696-2701, doi: 10.1109/BigData.2018.8621929

Hutchinson, T. (2020). Natural Language Processing and Machine Learning as Practical Toolsets for Archival Processing. *Records Management Journal*, 30(2), 155-174, doi: http://dx.doi.org/10.1108/RMJ-09-2019-0055

Hyland, M. (1981). *Introduction to Theoretical Psychology*. London: The Macmillan Press.

Icarte, R.T. vd. (2022). Reward Machines: Exploiting Reward Function Structure in Reinforcement Learning. *Journal of Artificial Intelligence Research*, 73, 173-208, doi: https://doi.org/10.1613/jair.1.12440

IIemobayo, J.A. vd. (2024). Hyperparameter Tuning in Machine Learning: A Comprehensive Review. *Journal of Engineering Research and Reports*, 26(6), 388-395, doi: https://doi.org/10.9734/jerr/2024/v26i61188

Imamguluyev, R. (2023). The Rise of GPT-3: Implications for Natural Language Processing and Beyond. *International Journal of Research Publication and Reviews*, 4(3), 4893-4903, doi: https://doi.org/10.55248/gengpi.2023.4.33987

Ismail, M., Hassan, N. ve Bafjaish, S.S. (2020). Comparative Analysis of Naive Bayesian Techniques in Health-Related forClassification Task. *Journal of Soft Computing and Data Mining*, 1(2), 1-10, doi: https://doi.org/10.30880/jscdm.2020.01.02.001

Ito, H. ve Lee, D. (2005). Assessing the Impact of the September 11 Terrorist Attacks on U.S. Airline Demand. *Journal of Economics and Business*, 57(1), 75-95, doi: https://doi.org/10.1016/j.jeconbus.2004.06.003

Ivakhnenko, A.G. ve Lapa, V.G. (1965). *Cybernetic Predicting Devices*. United States Joint Publications Research Service. New York: CCM Information Corp.

Ivakhnenko, A.G. (1971). Polynomial Theory of Complex Systems. *IEEE Transactions on Systems, Man and Cybernetics*, 4, 364-378, doi: 10.1109/TSMC.1971.4308320

İnik, Ö. (2023). Classification of Scenes in Aerial Images with Deep Learning Models. *Türk Doğa ve Fen Dergisi*, 12(1), 37-43, doi: https://doi.org/10.46810/tdfd.1225756

Jablonski, A. (2016). Scalability of Sustainable Business Models in Hybrid Organizations. *Sustainability*, 8(3), 1-36, doi: https://doi.org/10.3390/su8030194

Jain, N. ve Sharma, B.K. (2022). e-Mail Spam Detection using Machine Learning. *International Journal of Mechanical Engineering*, 7(12), 490-499, doi: https://doi.org/10.56452/7-12-46

Jankovic, S.M. (2022). The Multivariate Statistical Analysis - Multiple Linear Regression. *International Journal on Biomedicine and Healthcare*, 10(4), 253-255, doi: 10.5455/ijbh.2022.10.173-175

Januzaj, Y., Beqiri, E. ve Luma, A. (2023). Determining the Optimal Number of Clusters using Silhouette Score as a Data Mining Technique. *International Journal of Online and Biomedical Engineering*, 19(4), 174-182, doi: https://doi.org/10.3991/ijoe.v19i04.37059

Jaspert, N. vd. (2024). Memory and Archive. N. Dietrich, L. Lieb, N. Schneidereit (eds.). *Theory and Classification of Material Text Cultures* içinde (ss. 113-153), Berlin: De Gruyter, doi: https://doi.org/10.1515/9783111325514-004

Jayasudha, J. (2020). A Simple Linear Regression with Statistical Approach for Building Machine Learning Model and Prediction. *International Journal of Innovative Research in Science, Engineering and Technology*, 9(7), 6038-6043.

Jeba, N. vd. (2024). e-Vault for Legal Records. *ArXiv:2403.01186*, 1-7, doi: https://doi.org/10.48550/arXiv.2403.01186

Jeczmionek, E. ve Kowalski, P.A. (2021). Flattening Layer Pruning in Convolutional Neural Networks. *Symmetry*, 13(7), 1-13, doi: https://doi.org/10.3390/sym13071147

Jenkinson, H. (1937). A Manual of Archive Administration (2nd ed.). London: P. Lund, Humphries & Co.

Jensen, H.S. (2020). Digital Archival Literacy for (All) Historians. *Media History*, 27(2), 251-265, doi: https://doi.org/10.1080/13688804.2020.1779047

Jiang, F. vd. (2017). Artificial Intelligence in Healthcare: Past, Present and Future. *Stroke and Vascular Neurology*, 2(e000101), 230-243, doi: 10.1136/svn-2017-000101

Jimerson, R.C. (2003). Archives and Memory. *OCLC Systems & Services*, 19(3), 89-95, doi: https://doi.org/10.1108/10650750310490289

Jimerson, R.C. (2007). Archives for All: Professional Responsibility and Social Justice. *The American Archivist*, 70(2), 252-281.

Jin, P. vd. (2018). AID++: An Updated Version of AID on Scene Classification. *ArXiv:1806.00801*, 1-4, doi: https://doi.org/10.48550/arXiv.1806.00801

Joachims, T. (1998). Text Categorization with Support Vector Machines. 10.02.2025 tarihinde https://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf adresinden erişildi, doi: http://dx.doi.org/10.17877/DE290R-5097

Jones, M. (2016). Documenting Things: Bringing Archival Thinking to Interdisciplinary Collaborations. *The Australian Library Journal*, 65(3), 213-223, doi: https://doi.org/10.1080/00049670.2016.1204976

Jordan, M.I. ve Mitchell, T.M. (2015). Machine Learning: Trends, Perspectives and Prospects. *Science*, 349 (6245), 255-260, doi: 10.1126/science.aaa8415

Joyce, J. ve Verschelde, J. (2024). Algebraic Representations for Faster Predictions in Convolutional Neural Networks. *ArXiv:2408.07815*, 1-19, doi: https://doi.org/10.48550/arXiv.2408.07815

Jozefowicz, R. vd. (2024). Exploring the Limits of Language Modeling. *ArXiv:1602.02410*, 1-11, doi: https://doi.org/10.48550/arXiv.1602.02410

Jun, Z. (2021). The Development and Application of Support Vector Machine. *Journal of Physics: Conference Series*, 1748(5), 1-6, doi: 10.1088/1742-6596/1748/5/052006

Jung, A. (2022). *Machine Learning: The Basics*. Gateway East: Springer, doi: https://doi.org/10.1007/978-981-16-8193-6_7

Kakavas, P. ve Ugolini, F.C. (2019). Computational Thinking in Primary Education: A Systematic Literature Review. *Research on Education and Media*, 11(2), 64-94, doi: 10.2478/rem-2019-0023

Kalelioğlu, F., Gülbahar, Y. ve Kukul, V. (2016). A Framework for Computational Thinking Based on a Systematic Research Review. *Baltic Journal of Modern Computing*, 4(3), 583-596.

Kamnis, S. (2023). Generative Pre-Trained Transformers (GPT) for Surface Engineering. *Surface and Coatings Technology*, 466, 1-9, doi: https://doi.org/10.1016/j.surfcoat.2023.129680

Kaplan, A. ve Haenlein, M. (2019). Siri, Siri, in My Hand: Who's the Fairest in the Land? On the Interpretations, Illustrations and Implications of Artificial Intelligence. *Business Horizons*, 62, 15-25, doi: https://doi.org/10.1016/j.bushor.2018.08.004

Karabulut, B., Arslan, G. ve Ünver, H.M. (2019). A Weighted Similarity Measure for k-Nearest Neighbors Algorithm. *Celal Bayar University Journal of Science*, 15(4), 393-400, doi: 10.18466/cbayarfbe.618964

Karakaş, H.S., Rukancı, F. ve Anameriç, H. (2009). Organik Yöntem (Provenance). *Belge Yönetimi ve Arşiv Terimleri Sözlüğü* içinde (1.bs., s. 35). Ankara: Başbakanlık Basımevi.

Kaur, S.P. (2013). Variables in Research. *Indian Journal of Research and Reports in Medical Sciences*, 3(4), 36-38.

Kaushik, V. ve Walsh, C.A. (2019). Pragmatism as a Research Paradigm and Its Implications for Social Work Research. *Social Sciences*, 8(9), 1-17, doi: https://doi.org/10.3390/socsci8090255

Kennefick, D. (2012). Not Only Because of Theory: Dyson, Eddington and the Competing Myths of the 1919 Eclipse Expedition. Lehner, C., Renn, J., Schemmel, M. (eds.). *Einstein and the Changing Worldviews of Physics, Einstein Studies, vol 12* içinde (ss. 201-232), Boston: Birkhäuser, doi: https://doi.org/10.1007/978-0-8176-4940-1_9

Khanday, N.Y. ve Sofi, S.A. (2021). Taxonomy, State-of-the-Art, Challenges and Applications of Visual Understanding: A Review. *Computer Science Review*, 40, 1-23, doi: https://doi.org/10.1016/j.cosrev.2021.100374

Khatri, S. vd. (2023). A Deep Learning Approach for Detection of Disease in Plant Leaves. A.K. Das, J. Nayak, B. Naik, S. Vimal, D. Pelusi (eds.). *Computational Intelligence in Pattern Recognition* 725 içinde (ss. 169-181). Singapore: Springer, doi: https://doi.org/10.1007/978-981-99-3734-9_15

Khoei, T.T., Slimane, H.O. ve Kaabouch, N. (2023). Deep Learning: Systematic Review, Models, Challenges and Research Directions. *Neural Computing and Applications*, 35, 23103-23124, doi: https://doi.org/10.1007/s00521-023-08957-4

Kılıçarslan Cansu, S. ve Cansu, F.K. (2019). An Overview of Computational Thinking. *International Journal of Computer Science Education in Schools*, 3(1), 1-11, doi: 10.21585/ijcses.v3i1.53

Kızılkaya, B. (2019). *Unsupervised Anomaly Detection Algorithms*. (Yayımlanmamış Yüksek Lisans Tezi). Dokuz Eylül University Graduate School of Natural and Applied Sciences, İzmir.

Kiger, P.J (2025, Mayıs 28). How Mesopotamia Became the Cradle of Civilization. 28.05.2025 tarihinde https://www.history.com/articles/how-mesopotamia-became-the-cradle-of-civilization adresinden erişildi.

Kim, D., Kim, J. ve Kim, J. (2020). Elastic Exponential Linear Units for Convolutional Neural Networks. *Neurocomputing*, 406, 253-266, doi: https://doi.org/10.1016/j.neucom.2020.03.051

Kim, H.Y. (2013). Statistical Notes for Clinical Researchers: Assessing Normal Distribution (2) Using Skewness and Kurtosis. *Restorative Dentistry & Endodontics*, 38(1), 52-54, doi: https://doi.org/10.5395/rde.2013.38.1.52

Klareld, A.S. ve Gidlund, K.L. (2017). Rethinking Archives as Digital: The Consequences of "Paper Minds" in Illustrations and Definitions of e-Archives. *Archivaria*, 83, 81-108.

Klasnic, P.P. (2019). Archival Science-Today, *Atlanti+*, 29(1), 10-18, doi: https://doi.org/10.33700/2670-4579.29.1.10-18(2019)

Kohavi, R. ve Wolpert, D.H. (1996). Bias plus variance decomposition for zero-one loss functions. *Machine Learning Proceedings of the Thirteenth International Conference*, 96, 275-283.

Krizhevsky, A., Sutskever, I. ve Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. F. Pereira, C.J. Burges, L. Bottou, K.Q. Weinberger

(eds.). *Advances in Neural Information Processing Systems 25* içinde (ss. 1097-1105). New York: Morgan Kaufmann Publishers.

Kuever, J. (2024). Framing Participatory Methods in Provenance Research: From the Restitution of Objects to a Collaborative Production of Knowledge. *ICOFOM Study Series*, 52(1), 74-88, doi: https://doi.org/10.4000/11zlt

Kuhn, T. (1962). The Structure of Scientific Revolutions. Chicago: The University of Chicago Press.

Kumar, P.R. ve Manash, E.B.K. (2018). Deep Learning: A Branch of Machine Learning. *Journal of Physics: Conference Series*, 1228, 1-9, doi: 10.1088/1742-6596/1228/1/012045

Kumar, S. ve Chong, I. (2018). Correlation Analysis to Identify the Effective Data in Machine Learning: Prediction of Depressive Disorder and Emotion States. *International Journal of Environmental Research and Public Health*, 15(12), 1-24, doi: https://doi.org/10.3390/ijerph15122907

Kumar, T. vd. (2023). Image Data Augmentation Approaches: A Comprehensive Survey and Future Directions. *ArXiv:2301.02830*, 1-32, doi: https://doi.org/10.48550/arXiv.2301.02830

Kurilová, V. vd. (2021). Support Vector Machine and Deep-Learning Object Detection for Localisation of Hard Exudates. *Scientific Reports*, 11, 1-9, doi: https://doi.org/10.1038/s41598-021-95519-0

Kurniasari, L. ve Setyanto, A. (2020). Sentiment Analysis Using Recurrent Neural Network-LSTM in Bahasa Indonesia. *Journal of Engineering Science and Technology*, 15(5), 3242-3256.

Kuusi, O. ve Heinonen, S. (2022). Scenarios from Artificial Narrow Intelligence to Artificial General Intelligence: Reviewing the Results of the International Work/Technology 2050 Study. *World Futures Review*, 14(1), 65-79, doi: https://doi.org/10.1177/19467567221101637

Lalapura, V.S., Amudha, J. ve Satheesh, H.S. (2021). Recurrent Neural Networks for Edge Intelligence: A Survey. *ACM Computing Surveys*, 54(4), 1-38, doi: https://doi.org/10.1145/3448974

Largeron, C., Moulin, C. ve Géry, M. (2011). Entropy based Feature Selection for Text Categorization. *Proceedings of the 2011 ACM Symposium on Applied Computing*, 924-928, doi: https://doi.org/10.1145/1982185.1982389

Larose, D. (2005). *Discovering Knowledge in Data*. New Jersey: John Wiley & Sons.

Lasek, P. (t.y.). *Instance-Level Constraints in Density-Based Clustering*. 28.10.2023 tarihinde https://ceur-ws.org/Vol-1492/Paper_26.pdf adresinden erişildi.

Lawan, A.A. vd. (2023). Fundamentals of Machine-Learning Modeling for Behavioral Screening and Diagnosis of Autism Spectrum Disorder. A.S. El-Baz ve J.S. Suri (eds.). *Neural Engineering Techniques for Autism Spectrum Disorder volume 2: Diagnosis and Clinical Analysis* içinde (ss. 253-268), London: Elsevier, doi: https://doi.org/10.1016/B978-0-12-824421-0.00020-5

Lecun, Y. vd. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11), 2278-2324, doi: 10.1109/5.726791

Lee, M. (2023). GELU Activation Function in Deep Learning: A Comprehensive Mathematical Analysis and Performance. *ArXiv:2305.12073*, 1-19, doi: https://doi.org/10.48550/arXiv.2305.12073

Lee, M. vd. (2017). Heuristics for Assessing Computational Archival Science (CAS) Research: The Case of the Human Face of Big Data Project. *2017 IEEE International Conference on Big Data*, 2262-2270, doi: 10.1109/BigData.2017.8258179

Lee, Y.J., Hsieh, W.F. ve Huang, C.M. (2005). epsilon-SSVR: A Smooth Support Vector Machine for epsilon-Insensitive Regression. IE*EE Transactions on Knowledge and Data Engineering*, 17 (5), 678-685, doi: https://doi.org/10.1109/TKDE.2005.77

Lehmann, M. (2024). The Definitive Guide to Policy Gradients in Deep Reinforcement Learning: Theory, Algorithms and Implementations. *ArXiv:2401.13662*, 1-49, doi: https://doi.org/10.48550/arXiv.2401.13662

Lemieux, V.L. ve imProvenance Group (2016). Provenance: Past, Present and Future in Interdisciplinary and Multidisciplinary Perspective. V.L. Lemieux (ed.). *Building Trust in Information - Perspectives on the Frontiers of Provenance* içinde (ss. 3-45), Washington: Springer International Publishing, doi: 10.1007/978-3-319-40226-0_1

Levada, A.L.M., Nielsen, F. ve Haddad, M.F.C. (2024). Adaptive k-Nearest Neighbor Classifier based on the Local Estimation of the Shape Operator. *ArXiv:2409.05084*, 1-18, doi: https://doi.org/10.48550/arXiv.2409.05084

Levine, B.N. ve Liberatore, M. (2009). DEX: Digital Evidence Provenance Supporting Reproducibility and Comparison. *Digital Investigation*, 6, 48-56, doi: https://doi.org/10.1016/j.diin.2009.06.011

Levine, S. (2021). Designing, Visualizing and Understanding Deep Neural Networks: Discussion 6. 12.05.2024 tarihinde https://cs182sp21.github.io/static/discussions/dis6-sol.pdf?utm adresinden erişildi.

Li, H.X. vd. (2013). Probabilistic Support Vector Machines For Classification of Noise Affected Data. *Information Sciences*, 221, 60-71, doi: https://doi.org/10.1016/j.ins.2012.09.041

Li, J. (2022). Design of an Effective Archive Management System with a Compression Approach for Network Information Technology. *Wireless Communications and Mobile Computing*, 1, 1-10, doi: https://doi.org/10.1155/2022/3503841

Li, L., Doroslovački, M. ve Loew, M.H. (2019). Discriminant Analysis Deep Neural Networks. *53rd Annual Conference on Information Sciences and Systems*, 1-6, doi: 10.1109/CISS.2019.8692803

Li, Y. vd. (2020). Computational Thinking is More about Thinking than Computing. *Journal for STEM Education Research*, 3, 1-18, doi: https://doi.org/10.1007/s41979-020-00030-2

Liang, Y. vd. (2024). Multi-Dimensional Adaptive Learning Rate Gradient Descent Optimization Algorithm for Network Training in Magneto-Optical Defect Detection. *International Journal of Network Dynamics and Intelligence*, 3(3), 1-14, doi: https://doi.org/10.53941/ijndi.2024.100016

Lima, S.P. ve Cruz, M.D. (2020). A Genetic Algorithm Using Calinski-Harabasz Index for Automatic Clustering Problem. *Revista Brasileira De Computação Aplicada*, 12(3), 97-106, doi: https://doi.org/10.5335/rbca.v12i3.11117

Lin, J.P. (2017). An Infrastructure and Application of Computational Archival Science to Enrich and Integrate Big Digital Archival Data: Using Taiwan Indigenous Peoples Open Research Data (TIPD) as an Example. *2017 IEEE International Conference on Big Data*, 2279-2287, doi: 10.1109/BigData.2017.8258181

Liu, G. vd. (2022). Partial Convolution for Padding, Inpainting, and Image Synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5), 6096-6110, doi: 10.1109/TPAMI.2022.3209702

Liu, Q., Wang, J. ve Wang, M. (2016). An Improved Tone Model for SVM Based Tone Recognition. *International Journal of Hybrid Information Technology*, 9(10), 161-166, doi: http://dx.doi.org/10.14257/ijhit.2016.9.10.16

Liu, Z. (2025). *Artificial Intelligence for Engineers*. Cham: Springer, doi: https://doi.org/10.1007/978-3-031-75953-6_13

Lo, H.Z. ve Cohen, J.P. (2016). Academic Torrents: Scalable Data Distribution. *ArXiv:1603.04395*, 1-2, doi: http://arxiv.org/abs/1603.04395

López, O.A.M., López, A.M. ve Crossa, J. (2022). *Multivariate Statistical Machine Learning Methods for Genomic Prediction*. Cham: Springer, doi: https://doi.org/10.1007/978-3-030-89010-0_10

Lowry, J. ve MacNeil, H. (2021). Archival Thinking: Archaeologies and Genealogies. *Archives and Museum Informatics*, 21(1), 1-8, doi: https://doi.org/10.1007/s10502-020-09355-8

Lu, J. ve Wu, B. (2022). A Loss Function Base on Softmax for Expression Recognition. *Mobile Information Systems*, 1, 1-6, doi: https://doi.org/10.1155/2022/8230154

Lunge, N. (2024, Mart 24). A Deep Architecture: Multi-Layer Perceptron. 30.05.2024 tarihinde https://medium.com/@nlunge786/a-deep-architecture-multi-layer-perceptron-164bc5ff3842 adresinden erişildi.

MacArthur, K.R. ve Santo, J.B. (2022). A Multi-Level Analysis of the Effects of Statistics Anxiety/Attitudes on Trajectories of Exam Scores. *Journal of Statistics and Data Science Education*, 31(1), 102-112, doi: https://doi.org/10.1080/26939169.2022.2093805

MacNeil, H. (1994). Archival Theory and Practice: Between Two Paradigms. *Archivaria*, 37, 6-20.

MacNeil, H. (2008). Archivalterity: Rethinking Original Order. *Archivaria*, 66, 1-24.

Makleff, R. (2017). Sovereignty and Silence: The Creation of a Myth of Archival Destruction, Liège, 1408. 05.03.2024 tarihinde https://www.archivejournal.net/essays/sovereignty-and-silence/ adresinden erişildi.

Malik, N. ve Solanki, A. (2021). Simulation of Human Brain: Artificial Intelligence-Based Learning. S. Verma, P. Tomar (eds.). *Impact of AI Technologies on Teaching, Learning, and Research in Higher Education* içinde (ss. 150-160). London: IGI Global Scientific Publishing, doi: https://doi.org/10.4018/978-1-7998-4763-2.ch009

Mammone, A., Turchi, M. ve Cristianini, N. (2009). Support Vector Machines. *WIREs Computational Statistics*, 1(3), 283-289, doi: https://doi.org/10.1002/wics.49

Manessi, F. ve Rozza, A. (2019). Learning Combinations of Activation Functions. *ArXiv:1801.09403*, 1-6, doi: https://doi.org/10.48550/arXiv.1801.09403

Maniatopoulos, A. ve Mitianoudis, N. (2021). Learnable Leaky ReLU (LeLeLU): An Alternative Accuracy-Optimized Activation Function. *Information*, 12, 1-16, doi: https://doi.org/10.3390/info12120513

Manoff, M. (2004). Theories of the Archive from Across the Disciplines. *Libraries and the Academy*, 4(1), 9-25, doi: http://dx.doi.org/10.1353/pla.2004.0015

Marciano, R. (2022). AFTERWORD: Towards a new Discipline of Computational Archival Science (CAS). L. Jaillant (ed.). *Archives, Access and Artificial Intelligence: Working with Born-Digital and Digitized Archival Collections* içinde (ss. 205-218). Bielefeld: Bielefeld University Press, doi: https://doi.org/10.1515/9783839455845-009

Marciano, R. vd. (2018). Archival Records and Training in the Age of Big Data. J. Purcell, L.C. Sarin, P.T. Jaeger, J.C. Bertot (eds.), *Re-Envisioning the MLS: Perspectives on the Future of Library and Information Science Education Advances in Librarianship, Volume 44B* içinde (ss. 179-199). West Yorkshire: Emerald Publishing, doi: https://doi.org/10.1108/S0065-28302018000044B010

Marciano, R., Jansen, G. ve Underwood, W. (2020). Developing a Framework to Enable Collaboration in Computational Archival Science Education. 14.11.2023 tarihinde https://www2.archivists.org/sites/all/files/Research_Forum%20_2019_Marciano_final.pdf adresinden erişildi.

Marcot, B.G. ve Hanea, A.M. (2021). What is an Optimal Value of k in k-Fold Cross-Validation in Discrete Bayesian Network Analysis? Computational Statistics, 36, 2009-2031, doi: https://doi.org/10.1007/s00180-020-00999-9

Marin, A. vd. (2022). The Soft-Margin Support Vector Machine with Ordered Weighted Average. *Knowledge-Based Systems*, 237, 1-11, doi: https://doi.org/10.1016/j.knosys.2021.107705

Marr, B. (2018, 10 Aralık). How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read. 16.02.2025 tarihinde https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/ adresinden erişildi.

Marr, D. ve Hildreth, E. (1980). Theory of Edge Detection. *Proceedings of the Royal Society B*, 207, 187-217, doi: https://doi.org/10.1098/rspb.1980.0020

Massari, A. vd. (2023). Representing Provenance and Track Changes of Cultural Heritage Metadata in RDF: A Survey of Existing Approaches. *ArXiv*, 1-23, doi: https://doi.org/10.48550/arXiv.2305.08477

Mauthner, N.S. ve Gardos, J. (2015). Archival Practices and the Making of "Memories". *New Review of Information Networking*, 20(1-2), 155-169, doi: https://doi.org/10.1080/13614576.2015.1114825

Mawati, R., Sumertajaya, M. ve Afendi, F.M. (2014). Modified Centroid Selection Method of K-Means Clustering. *IOSR Journal of Mathematics*, 10(2), 49-53, doi: http://dx.doi.org/10.9790/5728-10234953

McCarthy, J. vd. (1955). A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence. 05.01.2025 tarihinde http://jmc.stanford.edu/articles/dartmouth/dartmouth.pdf adresinden erişildi.

McCrae, R.R. vd. (2010). Internal Consistency, Retest Reliability, and Their Implications for Personality Scale Validity. *Personality and Social Psychology Review*, 15(1), 28-50, doi: https://doi.org/10.1177/1088868310366253

McCulloch, W.S. ve Pitts, W. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, 5, 115-133.

McGill, R., Tukey, J.W. ve Larsen, W.A. (1978). Variations of Box Plots. *The American Statistician*, 32(1), 12-16, doi: https://doi.org/10.1080/00031305.1978.10479236

McLaren, B.M. vd. (2017). A Computer-Based Game that Promotes Mathematics Learning More than a Conventional Approach. *International Journal of Game-Based Learning*, 7(1), 36-56. doi:10.4018/IJGBL.2017010103

McLean, S. vd. (2021). The Risks Associated with Artificial General Intelligence: A Systematic Review. *Journal of Experimental & Theoretical Artificial Intelligence*, 35(5), 649-663, doi: https://doi.org/10.1080/0952813X.2021.1964003

Mechkaroska, D. vd. (2024). Architectural Scalability of Conversational Chatbot: The Case of ChatGPT. Arai, K. (ed.). *Advances in Information and Communication FICC 2024 Lecture Notes in Networks and Systems* içinde (ss. 54-71), 919, Springer, Cham, doi: https://doi.org/10.1007/978-3-031-53960-2_5

Meehan, J. (2009). Making the Leap from Parts to Whole: Evidence and Inference in Archival Arrangement and Description. *The American Archivist*, 72(1): 72-90, doi: https://doi.org/10.17723/aarc.72.1.kj672v4907m11x66

Meek, C., Thiesson, B. ve Heckerman, D. (2002). The Learning-Curve Sampling Method Applied to Model-Based Clustering. *Journal of Machine Learning Research*, 2, 397-418.

Méndez-Cruz C.F vd. (2024). Unsupervised Learning and Natural Language Processing Highlight Research Trends in a Superbug. *Frontiers in Artificial Intelligence*, 7, 1-11, doi: 10.3389/frai.2024.1336071

Meurers, T. vd. (2021). A Scalable Software Solution for Anonymizing High-Dimensional Biomedical Data. *GigaScience*, 10(10), 1-13, doi: https://doi.org/10.1093/gigascience/giab068

Michetti, G. (2016). Provenance: An Archival Perspective. V. Lemieux (ed.). *Building Trust in Information* içinde (ss. 59-68). Cham: Springer, doi: https://doi.org/10.1007/978-3-319-40226-0_3

Mickens, R. ve Patterson, C. (2016). What is Science?, *Georgia Journal of Science*, 74(2), 1-5.

Mienye, I.D. ve Jere, N. (2024). A Survey of Decision Trees: Concepts, Algorithms, and Applications. *IEEE Access*, 12, 86716-86727, doi: 10.1109/ACCESS.2024.3416838

Mienye, I.D., Swart, T.G. ve Obaido, G. (2024). Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants and Applications. *Information*, 15, 1-34, doi: https://doi.org/10.3390/info15090517

Milosavljevic, M., Nedovic, K. ve Spasenic, Z. (2024). Predicting Stock Prices Using Deep Learning Algorithms: A Case of Food-Processing Industry. *Athens Journal of Technology & Engineering*, 11, 1-15, doi: https://doi.org/10.30958/ajte.X-Y-Z

Ming, Q.F. vd. (2009). A Reinforcement Learning Algorithm Based on Minimum State Method and Average Reward. Proceedings of *WRI World Congress on Computer Science and Information Engineering*, 534-538, doi: 10.1109/CSIE.2009.433

Miseta, T., Fodor, A. ve Vathy-Fogarassy, A. (2024). Surpassing Early Stopping: A Novel Correlation-Based Stopping Criterion for Neural Networks. *Neurocomputing*, 567, 1-13, doi: https://doi.org/10.1016/j.neucom.2023.127028

Mish, F.C. (2004). Science. *The Merriam-Webster Dictionary* içinde (1.bs., s. 646.). Springfield: Merriam-Webster Inc.

Missier, P. vd. (2013). Provenance and Data Differencing for Workflow Reproducibility Analysis. *Concurrency and Computation: Practice and Experience*, 22(17), 2345-2359, doi: https://doi.org/10.1002/cpe.3035

Mitchell, T.M. (1997). *Machine Learning*. Redmond: McGraw-Hill.

Mohapatra, B.N. (2019). Image Edge Detection Techniques. *Transactions on Image Processing and Computer Vision*, 5(15), 15-19, doi: http://dx.doi.org/10.19101/TIPCV.2019.515008

Mohr, F. ve van Rijn, J.N. (2023). Fast and Informative Model Selection Using Learning Curve Cross-Validation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8), 9669-9680, doi: 10.1109/TPAMI.2023.3251957

Mokyr, J., Vickers, C. ve Ziebarth, N.L. (2015). The History of Technological Anxiety and the Future of Economic Growth: Is This Time Different? *Journal of Economic Perspectives*, 29(3), 31-50, doi: 10.1257/jep.29.3.31

Mollahosseini, A., Chan, D. ve Mahoor, M.H. (2015). Going Deeper in Facial Expression Recognition using Deep Neural Networks. *ArXiv: 1511.04110*, 1-10, doi: https://doi.org/10.48550/arXiv.1511.04110

Moor, J. (2006). The Dartmouth College Artificial Intelligence Conference: The Next Fifty Years. *AI Magazine*, 27(4), 87-91.

Moreau, L. (2010). The Foundations for Provenance on the Web. *Foundations and Trends in Web Science*, 2(2-3), 99-241, doi: https://doi.org/10.1561/1800000010

Morozov, A.V., Levkivskyi, V.L. ve Plechystyy, D.D. (2024). Activation Functions in Neural Networks: Overview and Comparison. *Informatics, Computing and Automation*, 35(3), 144-151, doi: https://doi.org/10.32782/2663-5941/2024.3.1/22

Mostofi, F., Tokdemir, O.B. ve Toğan, V. (2024). Generating Synthetic Data with Variational Autoencoder to Address Class Imbalance of Graph Attention Network Prediction Model for Construction Management. *Advanced Engineering Informatics*, 62(A), 1-17, doi: https://doi.org/10.1016/j.aei.2024.102606

Motta, V. (2019). Estimating Poisson Pseudo-Maximum-Likelihood rather than Log-Linear Model of a Log-Transformed Dependent Variable. *RAUSP Management Journal*, 54(4), 508-518, doi: 10.1108/RAUSP-05-2019-0110

Mukasheva, M. ve Omirzakova, A. (2021). Computational Thinking Assessment at Primary School in the Context of Learning Programming. *World Journal on Educational Technology: Current Issues*, 13(3), 336-353, doi: https://doi.org/10.18844/wjet.v13i3.5918

Mukid, M.A. vd. (2018). Credit Scoring Analysis Using Weighted k Nearest Neighbor. *Journal of Physics: Conference Series*, 1025, 1-7, doi: 10.1088/1742-6596/1025/1/012114

Muller, S., Feith, J.A. ve Fruin, R. (2003). *Manual for the Arrangement and Description of Archives, 2nd ed*. (Çev. A.H. Leavitt). New York: H.W. Wilson Co. (Orijinal Yayın Tarihi 1898).

Munafò, M.R. ve Flint, J. (2010). How Reliable are Scientific Studies? *British Journal of Psychiatry*, 197(4), 257-258, doi: 10.1192/bjp.bp.109.069849

Muntarina, K. vd. (2023). MultiResEdge: A Deep Learning-Based Edge Detection Approach. *Intelligent Systems with Applications*, 20, 1-14, doi: https://doi.org/10.1016/j.iswa.2023.200274

Müller, V.C. (2014). Risks of General Artificial Intelligence. *Journal of Experimental & Theoretical Artificial Intelligence*, 26(3), 297-301, doi: https://doi.org/10.1080/0952813X.2014.895110

Naidu, M.S.R. vd. (2023). Feature Extraction and Dense Layer Classification Using Convolutional Neural Network. *International Research Journal of Modernization in Engineering Technology and Science*, 5(4), 3244-3248, doi: https://www.doi.org/10.56726/IRJMETS36257

Narayan, S. (1997). The Generalized Sigmoid Activation Function: Competitive Supervised Learning. *Information Sciences*, 99(1-2), 69-82, doi: https://doi.org/10.1016/S0020-0255(96)00200-9

National Academy of Sciences. (2008). *Science, Evolution and Creationism*. Washington, D.C.: The National Academies Press.

National Archives. (2007). Appraisal Policy of the National Archives. 20.04.2024 tarihinde https://www.archives.gov/records-mgmt/scheduling/appraisal adresinden erişildi.

Naushad, R., Kaur, T. ve Ghaderpour, E. (2021). Deep Transfer Learning for Land Use and Land Cover Classification: A Comparative Study. *Sensors*, 21(23), 1-13, doi: https://doi.org/10.3390/s21238083

Navale, V. ve McAuliffe, M. (2018). Long-Term Preservation of Biomedical Research Data. *F1000Research*, 7, 1-16, doi: https://doi.org/10.12688/f1000research.16015.1

Negnevitsky, M. (1997). The History of Artificial Intelligence or from the "Dark Ages" to the Knowledge-Based Systems. *WIT Transactions on Information and Communication Technologies*, 19, 1-21.

Neha, S.V.S.T., Yadav, Y. ve Goyal, Y. (2024). Introduction to Machine Learning. *International Journal of Advanced Research in Science, Communication and Technology*, 4(2), 10-104, doi: 10.48175/IJARSCT-15723

Nesmith, T. (2006). Reopening Archives: Bringing New Contextualities into Archival Theory and Practice. *Archivaria*, 60, 259-274.

Nguyen, D. vd. (2017). Joint Network Coding and Machine Learning for Error-Prone Wireless Broadcast. *2017 IEEE 7th Annual Computing and Communication Workshop and Conference*, Las Vegas, NV, USA, 1-7, doi: 10.1109/CCWC.2017.7868415

Nicolas, C. (2022). e-Mail Spam Detection Using Machine Learning Based Text Analysis [Master Thesis, Universitat Politècnica de Catalunya]. Barcelona. 31.01.2025 tarihinde https://upcommons.upc.edu/bitstream/handle/2117/386385/EMAIL%20SPAM%20DETECTION%20USING%20MACHINE%20LEARNING%20BASED%20TEXT%20ANALYSIS.pdf;jsessionid=409CD81238094A1ACC001C5CF7EFF73E?sequence=6 adresinden erişildi.

Nilsson, N.J. (2010). *The Quest for Artificial Intelligence: A History of Ideas and Achievements*. New York: Cambridge University Press.

Nirthika, R. vd. (2022). Pooling in Convolutional Neural Networks for Medical Image Analysis: A Survey and an Empirical Study. *Neural Computing and Applications*, 34, 5321-5347, doi: https://doi.org/10.1007/s00521-022-06953-8

Nixon, J. (2004). What is Theory? *Educar*, 34, 27-37.

Nti, I.K., Adekoya, A.F. ve Weyori, B.A. (2020). A Comprehensive Evaluation of Ensemble Learning for Stock-Market Prediction. *Journal of Big Data*, 7(20), 1-40, doi: https://doi.org/10.1186/s40537-020-00299-5

Nurwaha, D. (2020). Comparison of Kernel Functions of Support Vector Machines: A Case Study for the Solar Cell Output Power Prediction. *International Journal of Energy Applications and Technologies*, 7(1), 1-6, doi: 10.31593/ijeat.630789

Nusrat, I. ve Jang, S.B. (2018). A Comparison of Regularization Techniques in Deep Neural Networks. *Symmetry*, 10(11), 1-18, doi: https://doi.org/10.3390/sym10110648

OECD. (2024). Explanatory Memorandum on the Updated OECD Definition of an AI System. OECD Artificial Intelligence Papers, March 2024, no. 8. OECD Publishing. 29.01.2025 tarihinde https://www.oecd.org/content/dam/oecd/en/publications/reports/2024/03/explanatory-memorandum-on-the-updated-oecd-definition-of-an-ai-system_3c815e51/623da898-en.pdf adresinden erişildi.

Oja, E. (2002). Unsupervised Learning in Neural Computation. *Theoretical Computer Science*, 287, 187-207.

Onorato, G. (2024). Bayesian Optimization for Hyperparameters Tuning in Neural Networks. *ArXiv:2410.21886*, 1-57, doi: https://doi.org/10.48550/arXiv.2410.21886

Orhaug, T. (1987). Computer Applications in Monitoring and Verification Technologies. A. M. Din (ed.), *Arms and Artificial Intelligence: Weapons and Arms Control Applications of Advanced Computing* içinde (ss.165-178), Oxford: Oxford University Press.

Osborne, J. (2010). Improving Your Data Transformations: Applying the Box-Cox transformation. *Practical Assessment, Research, and Evaluation*, 15(1), 1-9, doi: https://doi.org/10.7275/qbpc-gk17

Osborne, J.W. ve Waters, E. (2002). Four Assumptions of Multiple Regression that Researchers Should Always Test. *Practical Assessment, Research and Evaluation*, 8(2), 1-5, doi: https://doi.org/10.7275/r222-hv23

Oti, E.U. vd. (2021). Comprehensive Review of k-Means Clustering Algorithms. *International Journal of Advances in Scientific Research and Engineering*, 7(8), 64-68, doi: http://doi.org/10.31695/IJASRE.2021.34050

O'Toole, J.M. (2004). Back to the Future: Ernst Posner's Archives in the Ancient World. *The American Archivist*, 67(2), 161-175, doi: https://doi.org/10.17723/aarc.67.2.h124276213041315

Owens, M.T. ve Tanner, K.D. (2017). Teaching as Brain Changing: Exploring Connections between Neuroscience and Innovative Teaching. *CBE Life Sciences Education*, 16(2), 1-9, doi: https://doi.org/10.1187/cbe.17-01-0005

Oyedotun, O.K. vd. (2020). Going Deeper with Neural Networks without Skip Connections. *2020 IEEE International Conference on Image Processing*, 1756-1760, doi: 10.1109/ICIP40778.2020.9191356

Oyelakin, B.P. vd. (2022). Text Classification Using Recurrent Neural Network and Support Vector Machine on a Customer Review Dataset. J*ournal of Theoretical and Applied Information Technology*, 100(4), 948-957.

Oyewole, G.J. ve Thopil, G.A. (2023). Data Clustering: Application and Trends. *Artificial Intelligence Review*, 56, 6439-6475, doi: https://doi.org/10.1007/s10462-022-10325-y

Ördek, B. ve Borgianni, Y. (2023). Application of Unsupervised Learning and Image Processing into Classification of Designs to be Fabricated with Additive or Traditional Manufacturing. *Proceedings of the Design Society*, 3, 613-622, doi: 10.1017/pds.2023.62

Özdemir, Y.A. ve Esin, A.A. (2007). Parameter Estimation in Multiple Linear Regression Models Using Ranked Set Sampling. *Communications Faculty of Sciences University of Ankara Series A1 Mathematics and Statistics*, 56(1), 7-20, doi: https://doi.org/10.1501/Commua1_0000000194

Öztürk, F.E ve Demirel, N. (2023). Comparison of the Methods to Determine Optimal Number of Cluster. *Veri Bilim Dergisi*, 6(1), 34-45.

Palacio-Nino, J.O. ve Berzal, F. (2019). Evaluation Metrics for Unsupervised Learning Algorithms. *ArXiv:1905.05667*, 1-9, doi: https://doi.org/10.48550/arXiv.1905.05667

Palmer, B., Bubendorfer, K. ve Welch, I. (2011). Verifying Digital Provenance in Web Services. *2011 Fourth IEEE International Conference on Utility and Cloud Computing, Melbourne*, 445-450, doi: 10.1109/UCC.2011.73

Pan, S.J. ve Yang, Q. (2009). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10). 1345-1359, doi: 10.1109/TKDE.2009.191

Pandey, P., Pandey, D. ve Kumar, S. (2010). Reinforcement Learning by Comparing Immediate Reward. *International Journal of Computer Science and Information Security*, 8(5), 1-5, doi: https://doi.org/10.48550/arXiv.1009.2566

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas.* Newyork: Basic Books.

Pastur-Romay, L.A. vd. (2017). Parallel Computing for Brain Simulation. *Current Topics in Medicinal Chemistry*, 17(14), 1646-1668.

Patidar, P. vd. (2010). Image De-Noising by Various Filters for Different Noise. *International Journal of Computer Applications*, 9(4), 45-50.

Pattanajak, N. ve Malekmohamadi, H. (2019). Improving a 3-D Convolutional Neural Network Model Reinvented from VGG16 with Batch Normalization. *Proceedings of 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*, 45-50, doi: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00052

Payne, N. (2018). Stirring the Cauldron: Redefining Computational Archival Science (CAS) for the Big Data Domain. *2018 IEEE International Conference on Big Data*, 2743-2752, doi: 10.1109/BigData.2018.8622594

Pearce-Moses, R. (2005). *A Glossary of Archives and Records Terminology*. Chicago: Society of American Archivists.

Peñaranda, C., Reaño, C. ve Silla, F. (2022). Exploring the Use of Data Compression for Accelerating Machine Learning in the Edge With Remote Virtual Graphics Processing Units. *Concurrency and Computation: Practice and Experience*, 35(20), 1-19, doi: https://doi.org/10.1002/cpe.7328

Peng, C. (2024). Comprehensive Analysis of the Impact of Learning Rate and Dropout Rate on the Performance of Convolutional Neural Networks on the CIFAR-10 Dataset. *Applied and Computational Engineering*, 102, 183-192, doi: https://doi.org/10.54254/2755-2721/102/20241161

Peng, D., Gui, Z. ve Wu, H. (2023). Interpreting the Curse of Dimensionality from Distance Concentration and Manifold Effect. *ArXiv:2401.00422*, 1-17, doi: https://doi.org/10.48550/arXiv.2401.00422

Perepiczka, M., Chandler, N. ve Becerra, M. (2011). Relationship Between Graduate Students' Statistics Self-Efficacy, Statistics Anxiety, Attitude Toward Statistics and Social Support. *The Professional Counselor*, 1(2), 99-108, doi: 10.15241/mpa.1.2.99

Phan, H. vd. (2017). Audio Scene Classification with Deep Recurrent Neural Networks. *ArXiv:1703.04770*, 1-5, doi: https://doi.org/10.48550/arXiv.1703.04770

Pitt, J.C. (1998). Explaining Change in Science. *Society for Philosophy and Technology Quarterly Electronic Journal*, 3(3), 135-140, doi: https://doi.org/10.5840/techne19983326

Polit, D.F. ve Beck, C.T. (2017). *Nursing Research: Generating and Assessing Evidence for Nursing Practice, 10th ed*. Philadelphia: Wolters Kluwer.

Prusty, S., Patnaik, S. ve Dash, S.K. (2022). SKCV: Stratified K-fold Cross-Validation on ML Classifiers for Predicting Cervical Cancer. *Frontiers in Nanotechnology*, 4, 1-12, doi: https://doi.org/10.3389/fnano.2022.972421

Poole, A.H. ve Todd-Diaz, A. (2023). Where Are Diversity, Equity, and Inclusion (DEI) and Social Justice in North American Graduate Archival Education? *Journal of Education for Library and Information Science*, 64(4), 434-457, doi: https://doi.org/10.3138/jelis-2022-0054

Popescu, M.C. vd. (2009). Multilayer Perceptron and Neural Networks. *WSEAS Transactions on Circuits and Systems*, 7(8), 579-588.

Popper, K.R. (1962). *Conjectures and Refutations: The Growth of Scientific Knowledge*. New York: Basic Books.

Posner, E. (1940). Some Aspects of Archival Development Since the French Revolution. *The American Archivist*, 3(3), 159-172.

Posner, E. (1955). The National Archives and the Archival Theorist. *The American Archivist*, 18(3), 207-216, doi: https://doi.org/10.17723/aarc.18.3.g503072842452823

Posner, E. (1972). *Archives in the Ancient World*. Massachusetts: Harvard University Press.

Possler, D., Bruns, S. ve Niemann-Lenz, J. (2019). Data is the New Oil-But How Do We Drill It? Pathways to Access and Acquire Large Data Sets in Communication Science. *International Journal of Communication*, 13, 3894-3911.

Pradnyana, G.A. vd. (2021). Performance Analysis of Support Vector Machines with Polynomial Kernel for Sentiment Polarity Identification: A Case Study in Lecturer's Performance Questionnaire. *Journal of Physics: Conference Series*, 1810, 1-8, doi: 10.1088/1742-6596/1810/1/012033

Prelinger, R. (2007). Archives and Access in the 21st Century. *Cinema Journal*, 46(3), 114-118, doi: https://dx.doi.org/10.1353/cj.2007.0027

Prewitt, J.M.S. (1970). Object Enhancement and Extraction. B.S. Lipkin (ed.). *Picture Processing and Psychopictorics* içinde (ss. 75-149). New York: Academic Press.

Punzalan, R.L. ve Caswell, M. (2016). Critical Directions for Archival Approaches to Social Justice. *The Library Quarterly*, 86(1), 1-18, doi: https://doi.org/10.1086/684145

Pusztahazi, L.S., Eigner, G. ve Csiszar, O. (2024). Parametric Activation Functions for Neural Networks: A Tutorial Survey. *IEEE Access*, 12, 168626-168644, doi: 10.1109/ACCESS.2024.3474574

Qiao, Q. (2024). Image Processing Technology Based on Machine Learning. *IEEE Consumer Electronics Magazine*, July/August 2024, 90-99. 21.02.2025 tarihinde https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9712416 adresinden erişildi.

Quigley, E. vd. (2021). Data is the New Oil: Citizen Science and Informed Consent in an Era of Researchers Handling of an Economically Valuable Resource. *Life Sciences, Society and Policy*, 17, 1-13, doi: https://doi.org/10.1186/s40504-021-00118-6

Rajanand, A. ve Singh, P. (2024). ErfReLU: Adaptive Activation Function for Deep Neural Network. *Pattern Analysis and Applications*, 27(68), 1-11, doi: https://doi.org/10.1007/s10044-024-01277-w

Ramu, K. ve Narayanan, K. (2023). Predictive Maintenance Using RNN and LSTM Models. *Journal of Theoretical and Applied Information Technology*, 101(11), 4782-4790.

Ramzan, F. vd. (2024). Generative Adversarial Networks for Synthetic Data Generation in Finance: Evaluating Statistical Similarities and Quality Assessment. *AI*, 5(2), 667-685, doi: https://doi.org/10.3390/ai5020035

Randby, T. ve Marciano, R. (2020). Digital Curation and Machine Learning Experimentation in Archives. *2020 IEEE International Conference on Big Data*, 1904-1913, doi: 10.1109/BigData50022.2020.9377788

Rane, N.L. vd. (2024). Machine Learning and Deep Learning for Big Data Analytics: A Review of Methods and Applications. *Partners Universal International Innovation Journal* 2(3), 172-197, doi: https://doi.org/10.5281/zenodo.12271006

Ranglani, H. (2024). Empirical Analysis of the Bias-Variance Tradeoff Across Machine Learning Models. *Machine Learning and Applications: An International Journal*, 11(4), 1-12, doi:10.5121/mlaij.2024.11401

Rankin, J. ve Weise, C. (2022). *World History Since 1500: An Open and Free Textbook*. Johnson City: East Tennessee State University.

Raschka, S. (2020). STAT 451: Introduction to Machine Learning Lecture Notes. 26.09.2024 tarihinde https://sebastianraschka.com/pdf/lecture-notes/stat451fs20/02-knn__notes.pdf adresinden erişildi.

Raynor, W.J. (1999). Artificial Intelligence. *The International Dictionary of Artificial Intelligence* içinde (1.bs., s. 13). Chicago: Glenlake Publishing Company.

Rayward, W.B. (2002). A History of Computer Applications in Libraries: Prolegomena. *IEEE Annals of the History of Computing*, 24(2), 4-15, doi: 10.1109/MAHC.2002.1010066

Renzulli, J.S. (2020). The Catch-A-Wave Theory of Adaptability: Core Competencies for Developing Gifted Behaviors in the Second Machine Age of Technology. *International Journal for Talent Development and* Creativity, 8(1), 79-95, doi: https://doi.org/10.7202/1076749ar

Reyad, M., Sarhan, A. ve Arafa, M. (2023). A Modified Adam Algorithm for Deep Neural Network Optimization. *Neural Computing and Applications*, 35, 17095-17112, doi: https://doi.org/10.1007/s00521-023-08568-z

Rezk, N. vd. (2020). Recurrent Neural Networks: An Embedded Computing Perspective. *IEEE Access*, 81(1): 57967-57996, doi: https://doi.org/10.1109/ACCESS.2020.2982416

Riad, R. vd. (2022). Learning Strides in Convolutional Neural Networks. *ArXiv:2202.01653*, 1-17, doi: https://doi.org/10.48550/arXiv.2202.01653

Rich, P.J., Egan, G. ve Ellsworth, J. (2019). A Framework for Decomposition in Computational Thinking. *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, 416-421, doi: https://doi.org/10.1145/3304221.3319793

Ridener, J. (2009). *From Polders to Postmodernism: A Concise History of Archival Theory*. Duluth, MN: Litwin Books.

Ring, J. (2023, 30 Nisan). 30 Years Ago, One Decision Altered the Course of Our Connected World. 25.05.2023 tarihinde https://www.npr.org/2023/04/30/1172276538/world-wide-web-internet-anniversary adresinden erişildi.

Risi, S. ve Preuss, M. (2020). Behind DeepMind's AlphaStar AI that Reached Grandmaster Level in StarCraft II. - *Künstliche Intelligenz*, 34, 85-86, doi: https://doi.org/10.1007/s13218-020-00642-1

Robichaud, D. (2021). Integrating Equity and Reconciliation Work into Archival Descriptive Practice at the University of Waterloo. *Archivaria*, 91, 74-103.

Rokach, L. ve Maimon, O. (2005). *Data Mining and Knowledge Discovery Handbook*. Boston: Springer, doi: https://doi.org/10.1007/0-387-25465-X_9

Roke, E.R. ve Tillman, R.K. (2022). Pragmatic Principles for Archival Linked Data. *The American Archivist*, 85(1), 173-201, doi: https://doi.org/10.17723/2327-9702-85.1.173

Rosali, D. ve Suryadi, D. (2021). An Analysis of Students' Computational Thinking Skills on The Number Patterns Lesson during The Covid-19 Pandemic. F*ormatif: Jurnal Ilmiah Pendidikan MIPA*, 11(2), 217-232, doi: http://dx.doi.org/10.30998/formatif.v11i2.9905

Rosenblatt, F. (1957). The Perceptron and a Perceiving and Recognizing Automation. *Project PARA Report No: 85-460-1*. 24.02.2025 tarihinde https://bpb-us-e2.wpmucdn.com/websites.umass.edu/dist/a/27637/files/2016/03/rosenblatt-1957.pdf adresinden erişildi.

Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, 65(6), 386-408, doi: https://doi.org/10.1037/h0042519

Roux, M. (2018). A Comparative Study of Divisive and Agglomerative Hierarchical Clustering Algorithms. *Journal of Classification*, 35, 345-366, doi: https://doi.org/10.1007/s00357-018-9259-9

Roy, S. (2024). Trustworthy Dimensionality Reduction. *ArXiv:2405.05868*, 1-51, doi: https://doi.org/10.48550/arXiv.2405.05868

Runardotter, M. (2007). *Information Technology, Archives and Archivists: An Interacting Trinity for Long-term Digital Preservation*. Yüksek Lisans Tezi. Social Informatics, Luleå University of Technology, İsveç.

Rupprecht, L. vd. (2020). Improving Reproducibility of Data Science Pipelines through Transparent Provenance Capture. *Proceedings of the VLDB Endowment*, 13(12), 3354-3368, doi: https://doi.org/10.14778/3415478.3415556

Russakovsky, O. vd. (2014). ImageNet Large Scale Visual Recognition Challenge. *ArXiv:1409.0575*, 1-43, doi: https://doi.org/10.48550/arXiv.1409.0575

Russell, S.J. ve Norvig, P. (2016). *Artificial Intelligence: A Modern Approach 3rd ed*. Harlow: Pearson.

Sağlık. Ö. (2021). *Elektronik Belge Yönetimi Uygulamalarındaki Koşullar Işığında e-İmzalı Belgelerin Delil Değerinin Arşivsel Güvenilirlik Açısından İncelenmesi*. İstanbul Üniversitesi Sosyal Bilimler Enstitüsü, İstanbul.

Sağlık, Ö. (2025). Trustworthiness of Digital Records through NDSA Levels of Digital Preservation. *Archives and Records*, 46(1), 37-50, doi: https://doi.org/10.1080/23257962.2024.2440745

Saiz, C. ve Rivas, S.F. (2023). Critical Thinking, Formation, and Change. *Journal of Intelligence*, 11(12), 1-26, doi: https://doi.org/10.3390/jintelligence11120219

Sallans, B. ve Hinton, G.E. (2004). Reinforcement Learning with Factored States and Actions. *Journal of Machine Learning Research*, 5, 1063-1088.

Samiappan, D. vd. (2018). Machine Learning Based Image Processing Using Unsupervised Approach. *Pakistan Journal of Biotechnology*, 15(3), 751-755.

Samuel, S., Löffler, F. ve König-Ries, B. (2021). Machine Learning Pipelines: Provenance, Reproducibility and FAIR Data Principles. *ArXiv:2006.12117*, 1-2, doi: https://doi.org/10.48550/arXiv.2006.12117

Sani, A.M., Ben-Musa, A.S. ve Haladu, M. (2021). In-Depth Study of Decision Tree Model. *International Journal of Science and Research*, 10(11), 705-709, doi: 10.21275/MR211102051237

Sankar, A. (2021, 29 Eylül). Principal Component Analysis Part 1: The Different Formulations. 23.07.2024 tarihinde https://medium.com/data-science/principal-component-analysis-part-1-the-different-formulations-6508f63a5553 adresinden erişildi.

Santana, İ. vd. (2022). Support Vector Machines with the Hard-Margin Loss: Optimal Training via Combinatorial Benders' Cuts. *ArXiv:2207.07690*, 1-23, doi: https://doi.org/10.48550/arXiv.2207.07690

Santosh, A. ve Moguram, A. (2019). Image Classification Using Support Vector Machine. *Journal of Emerging Technologies and Innovative Research*, 6(2), 73-78.

Sarewitz, D. ve Pielke, R. (1999). Prediction in Science and Policy. *Technology in Society*, 21(2), 121-133, doi: https://doi.org/10.1016/S0160-791X(99)00002-0

Sarker, I.H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, 1-21, doi: https://doi.org/10.1007/s42979-021-00592-x

Sassoon, J. (2007). Sharing Our Story: An Archaeology of Archival Thought. *Archives and Manuscripts*, 35(2), 40-54.

Saud, A., Shakya, S. ve Neupane, B. (2021). Analysis of Depth of Entropy and GINI Index Based Decision Trees for Predicting Diabetes. *Indian Journal of Computer Science*, 6(6), 19-28, doi: 10.17010/ijcs/2021/v6/i6/167641

Saxena, S. (2025, 1 Mayıs). The Architecture of Lenet-5. 01.05.2025 tarihinde https://www.analyticsvidhya.com/blog/2021/03/the-architecture-of-lenet-5/ adresinden erişildi.

Sayıcı, G. (2013). *Karar Ağaçları, Bayes Ağları ve Etki Diyagramları Aracılığı ile Bilgi Keşfi ve Karar Verme*. İstanbul Üniversitesi Sosyal Bilimler Enstitüsü, İstanbul.

Schellenberg, T.R. (1956). *Modern Archives: Principles & Techniques*. Chicago: The Society of American Archivists.

Schellenberg, T.R. (1961). Archival Principles of Arrangement. *The American Archivist*, 24(1), 11-24.

Schellenberg, T.R. (1965). *The Management of Archives*. New York: Columbia University Press.

Schelter, S., Guha, S. ve Grafberger, S. (2024). Automated Provenance-Based Screening of ML Data Preparation Pipelines. *Datenbank Spektrum*, 24, 187-196, doi: https://doi.org/10.1007/s13222-024-00483-4

Schlegel, M. ve Sattler, K.U. (2023). MLflow2PROV: Extracting Provenance from Machine Learning Experiments. *2023 International Workshop on Data Management for End-to-End Machine Learning Bildiriler Kitabı* içinde (ss. 1-4), doi: https://doi.org/10.1145/3595360.3595859

Schmidt-Hieber, J. (2020). Nonparametric Regression Using Deep Neural Networks with ReLU Activation Function. *The Annals of Statistics*, 48(4), 1875-1897, doi: https://doi.org/10.1214/19-AOS1875

Schneider, A., Hommel, G. ve Blettner, M. (2010). Linear Regression Analysis. *Deutsches Ärzteblatt*, 107(44), 776-782, doi: 10.3238/arztebl.2010.0776

Scholkopf, B. vd. (1997). Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classifiers. *IEEE Transactions on Signal Processing*, 45(11), 2758-2765, doi: 10.1109/78.650102

Schubert, E. vd. (2017). DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM Transactions on Database Systems*, 42(3), 1-21, doi: https://doi.org/10.1145/3068335

Schuster, M. ve Paliwal, K.K. (1997). Bidirectional Recurrent Neural Networks. *IEEE Transactions on Signal Processing*, 45(11), 2673-2681, doi: 10.1109/78.650093

Schwartz, J.M. ve Cook, T. (2002). Archives, Records and Power: The Making of Modern Memory. *Archival Science* 2, 1-19, doi: https://doi.org/10.1007/BF02435628

Sciubba, E. (2009). Artificial Intelligence and Expert Systems in Energy Systems Analysis. C.A. Frangopoulos (ed.). *Exergy, Energy System Analysis and Optimization vol. 3* içinde (ss. 1-29). Oxford: EOLSS Publishers Co.

Sehra, S., Flores, D. ve Montanez, G.D. (2021). Undecidability of Underfitting in Learning Algorithms. *Proceedings of 2nd International Conference on Computing and Data Science*, 591-594, doi: 10.1109/CDS52072.2021.00107

Selby, C.C. ve Woollard, J. (2010). Computational Thinking: The Developing Definition. *SIGCSE'14: The 45th ACM Technical Symposium on Computer Science Education* Atlanta, Georgia, USA, March 5-8, 2014, Association for Computing Machinery, New York: United States. 02.07.2024 tarihinde https://core.ac.uk/download/pdf/17189251.pdf adresinden erişildi.

Sen, P.C., Hajra, M. ve Ghosh, M. (2020). Supervised Classification Algorithms in Machine Learning: A Survey and Review. J.K. Mandal, D. Bhattacharya (eds.). *Emerging Technology in Modelling and Graphics* içinde (ss. 99-111), Singapore: Springer, https://doi.org/10.1007/978-981-13-7403-6_11

Sergievskii, V. (2020). Super Strong Artificial Intelligence and Human Mind. *Procedia Computer Science*, 169, 458-460, doi: https://doi.org/10.1016/j.procs.2020.02.225

Shadbolt, N. (2022). From so Simple a Beginning: Species of Artificial Intelligence. *Daedalus*, 151(2), 28-42, doi: https://doi.org/10.1162/daed_a_01898

Shah, A. vd. (2016). Deep Residual Networks with Exponential Linear Unit. *ArXiv:1604.04112*, 1-7, doi: https://doi.org/10.48550/arXiv.1604.04112

Shah, B. ve Bhavsar, H. (2022). Time Complexity in Deep Learning Models. *Procedia Computer Science*, 215, 202-210, doi: https://doi.org/10.1016/j.procs.2022.12.023

Shahapure, K.R. ve Nicholas, C. (2020). Cluster Quality Analysis Using Silhouette Score. *IEEE 7th International Conference on Data Science and Advanced Analytics Bildirileri içinde* (ss. 747-748), Sydney, Avusturalya, doi: 10.1109/DSAA49011.2020.00096

Shapin, S. (2018). *The Scientific Revolution 2nd ed*. Chicago: University of Chicago Press.

Sharma, S. ve Mehra, R. (2019). Implications of Pooling Strategies in Convolutional Neural Networks: A Deep Insight. *Foundations of Computing and Decision Sciences*, 44(3), 303-330, doi: 10.2478/fcds-2019-0016

Sharma, S., Sharma, S. ve Athaiya, A. (2020). Activation Functions in Neural Networks. *International Journal of Engineering Applied Sciences and Technology*, 4(12), 310-316.

Sharma, S. ve Chatterjee, S. (2021). Winsorization for Robust Bayesian Neural Networks. *Entropy*, 23(11), 1-48, doi: https://doi.org/10.3390/e23111546

Shatravin, V., Shashev, D. ve Shidlovskiy, S. (2023). Implementation of the SoftMax Activation for Reconfigurable Neural Network Hardware Accelerators. *Applied Sciences*, 13(23), 1-15, doi: https://doi.org/10.3390/app132312784

Sheikh, H., Prins, C. ve Schrijvers, E. (2023). *Mission AI: The New System Technology*. The Hauge: Springer, doi: https://doi.org/10.1007/978-3-031-21448-6_2

Shen, J. vd. (2022). Transfer Learning of Phase Transitions in Percolation and Directed Percolation. *Physical Review E*, 105, 1-11, doi: https://doi.org/10.1103/PhysRevE.105.064139

Sheridan, T. (1798). Hypothesis. *A Complete Dictionary of the English Language, 2nd ed.* içinde (2.bs., s. 329). London: Charles Dilly.

Shi, G. vd. (2019). Enhance the Performance of Deep Neural Networks via L2 Regularization on the Input of Activations. *Neural Processing Letters*, 50, 57-75, doi: https://doi.org/10.1007/s11063-018-9883-8

Shinde, G. vd. (2024). AI in Archival Science - A Systematic Review. *ArXiv:2410.09086*, 1-24, doi: https://doi.org/10.48550/arXiv.2410.09086

Shpilrain, V. ve Sosnovski, B. (2016). Compositions of Linear Functions and Applications to Hashing. *Groups Complexity Cryptology*, 8(2), 155-161, doi: https://doi.org/10.1515/gcc-2016-0016

Siahaan, M.D.L. ve Siahaan, A.P.U. (2016). Fingerprint Pattern Recoqnition Using LVQ. *IOSR Journal of Computer Engineering*, 18(6), 85-92, doi: 10.9790/0661-1806048592

Silva, J.G.C. (2024). *Scientific Research: An Introduction*. London: B.P. International.

Simonyan, K. ve Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv:1409.1556*, 1-14, doi: https://doi.org/10.48550/arXiv.1409.1556

Singh, S.P. (2021). Study of k-Means and Enhanced k-Means Clustering Algorithm. *International Journal of Research in Computer Science*, 4(10), 103-107.

Sismanidou, A., Tarradellas, J. ve Suau-Sanchez, P. (2022). The Uneven Geography of US Air Traffic Delays: Quantifying the Impact of Connecting Passengers on Delay Propagation. *Journal of Transport Geography*, 98, 1-12, doi: https://doi.org/10.1016/j.jtrangeo.2021.103260

Skiena, S.S. (2008). *The Algorithm Design Manual 2nd ed*. New York: Springer.

Smarika, N. vd. (2015). Agglomerative Hierarchical Clustering Technique for Partitioning Patent Dataset. *Proceedings of 4th International Conference on Reliability, Infocom Technologies and Optimization*, 1-4, doi: 10.1109/ICRITO.2015.7359281

Smith, J.F. (1981). Theodore R. Schellenberg: Americanizer and Popularizer. *The American Archivist*, 44(4), 313-326.

Smith, R. (1990). Artificial Intelligence. *Collins Dictionary of Artificial Intelligence* içinde (1.bs., s. 22). New York: Facts on File Publications.

Smola, A.J. ve Schölkopf, B. (2004). A Tutorial on Support Vector Regression. *Statistics and Computing*, 14, 199-222, doi: https://doi.org/10.1023/B:STCO.0000035301.49549.88

Sneha, K.A. ve Fong, W.H. (2023). Covariance Matrix Method in Principal Component Analysis. *Proceedings of Science and Mathematics*, 15, 8-15.

Sohaee, N. (2023). Error and Optimism Bias Regularization. *Journal of Big Data*, 10, 1-12, doi: https://doi.org/10.1186/s40537-023-00685-9

Solan, Z. vd. (2005). Unsupervised Learning of Natural Languages. *Proceedings of the National Academy of Sciences*. 102(33), 11629-11634, doi: https://doi.org/10.1073/pnas.0409746102

Souai, W. (2024, Haziran 15). Convolutional Neural Network:[Updated 2024]. 11.07.2024 tarihinde https://medium.com/ubiai-nlp/convolutional-neural-network-updated-2024-130ebb6885ed adresinden erişildi.

Souza, R. vd. (2019). Provenance Data in the Machine Learning Lifecycle in Computational Science and Engineering. *2019 IEEE/ACM Workflows in Support of Large-Scale Science*, 1-10, doi: https://doi.org/10.1109/WORKS49585.2019.00006

Sowjanya, N. ve Veeralaksmi, Y. (2024). An Efficient Method for Text Classification Using Naive Bayes. *International Journal of Modern Engineering Research*, 14(4), 143-148.

Sponton, H. ve Cardelino, J. (2015). A Review of Classic Edge Detectors. *Image Processing On Line*, 5, 90-123, doi: https://doi.org/10.5201/ipol.2015.35

Srivastava, D.K. ve Bhambhu, L. (2010). Data Classification Using Support Vector Machine. *Journal of Theoretical and Applied Information Technology*, 12(1), 1-7.

Srivastava, N. vd. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15, 1929-1958.

Sriwong, K., Kerdprasop, K. ve Kerdprasop, N. (2021). The Study of Noise Effect on CNN-Based Deep Learning from Medical Images. *International Journal of Machine Learning*, 11(3), 202-207, doi: 10.18178/ijmlc.2021.11.3.1036

Stach, C. (2023). Data is the New Oil-Sort of: A View on Why This Comparison is Misleading and Its Implications for Modern Data Administration. *Future Internet*, 15(2), 1-49, doi: https://doi.org/10.3390/fi15020071

Stamenkovic, P. (2023). Facts and Objectivity in Science. *Interdisciplinary Science Reviews*, 48(2), 277-298, doi: 10.1080/03080188.2022.2150807

Stančić, H. (2018). Computational Archival Science. *Moderna Arhivistika*, 1(2), 323-330, doi: https://doi.org/10.54356/ma/2018/iyln2017

Stavropoulos, V. vd. (2023). Deep Learning(s) in Gaming Disorder through the User-Avatar Bond: a Longitudinal Study Using Machine Learning. *Journal of Behavioral Addictions*, 12(4), 878-894, doi: https://doi.org/10.1556/2006.2023.00062

Steenbergen, M.R. (2016). *Regression Analysis: A Primer for the Social Sciences*. Zurich: Soziologisches Institut.

Stephano, A.A. (2024). Recontextualization of Provenance: Records in Contexts and the Principle of Provenance. *Archival Science*, 24, 783-800, doi: https://doi.org/10.1007/s10502-024-09460-y

Stewart, J., Harte, V. ve Sambrook, S. (2011). What is Theory?, J*ournal of European Industrial Training*, 35(3), 221-229, doi: https://doi.org/10.1108/03090591111120386

Strobl, C., Boulesteix, A.L. ve Augustin, T. (2007). Unbiased Split Selection for Classification Trees Based on the Gini Index. *Computational Statistics & Data Analysis*, 52(1), 483-501, doi: https://doi.org/10.1016/j.csda.2006.12.030

Sudha, V. ve Ganeshbabu, T.R. (2021). A Convolutional Neural Network Classifier VGG-19 Architecture for Lesion Detection and Grading in Diabetic Retinopathy Based on Deep Learning. *Computers, Materials & Continua* 66(1), 827-842, doi: https://doi.org/10.32604/cmc.2020.012008

Sun, S. vd. (2016). On the Depth of Deep Neural Networks: A Theoretical View. *Proceedings of the 13rd AAAI Conference on Artificial Intelligence*, 2066-2072, doi: https://doi.org/10.48550/arXiv.1506.05232

Sun, X. vd. (2024). Airline Competition: A Comprehensive rReview of Recent Research. *Journal of the Air Transport Research Society*, 2, 1-17, doi: https://doi.org/10.1016/j.jatrs.2024.100013

Sundaram, N.M. ve Sivanandam, S.N. (2018). Softmax Activation Function for Neural Network Multiclass Classifiers. *Karpagam Jes*, 12(4), 144-152.

Suriya, M., Chandran, V. ve Sumithra, M.G. (2019). Enhanced Deep Convolutional Neural Network for Malarial Parasite Classification. *International Journal of Computers and Applications*, 44(12), 1113-1122, doi: https://doi.org/10.1080/1206212X.2019.1672277

Sutherland, J.W. (1975). *Systems: Analysis, Administration and Architecture*. New York: Van Nostrand Reinhold Company.

Sutton, R.S. ve Barto, A.G. (2018). *Reinforcement Learning: An Introduction, 2nd ed*. Cambridge: MIT Press.

Sweeney, S. (2008). The Ambiguous Origins of the Archival Principle of "Provenance". *Libraries & the Cultural Record*, 43(2), 193-213.

Szandala, T. (2021). Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks. A.K. Bhoi, P.K. Mallick, C.M. Liu, V.E. Balas (eds.). *Bio-inspired Neurocomputing* içinde (ss. 203-224), Studies in Computational Intelligence, vol 903. Gateway East: Springer, doi: https://doi.org/10.1007/978-981-15-5495-7_11

Taber, K.S. (2017). Reflecting the Nature of Science in Science Education. K.S. Taber, B. Akpan (Eds.), *Science Education: An International Course Companion* içinde (ss. 23-37). Rotterdam: Sense Publishers, doi: https://doi.org/10.1007/978-94-6300-749-8

Tadeu, P. ve Brigas, C. (2022). Multiple Intelligence's and Computational Thinking. Journal of Computer and Education Research, 10(9), 1-17, doi: https://doi.org/10.18009/jcer.1027934

Tan, M. ve Le, Q.V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *ArXiv:1905.11946*, 1-11, doi: https://doi.org/10.48550/arXiv.1905.11946

Tang, Y. (2022). Airline Flight Delay Prediction Using Machine Learning Models. *Proceedings of the 5th International Conference on E-Business and Internet*, 151-154, doi: https://doi.org/10.1145/3497701.3497725

Taper, M.L. ve Subhash R.L. (2004). The Nature of Scientific Evidence: A Forward-Looking Synthesis, M.L. Taper ve S.R. Lele (eds.), *The Nature of Scientific Evidence: Statistical, Philosophical and Empirical Considerations* içinde (ss. 527-551) Chicago: University of Chicago Press, doi: https://doi.org/10.7208/chicago/9780226789583.003.0016

Taud, H. ve Mas, J. (2018). Multilayer Perceptron (MLP). M.T.C. Olmedo, M. Paegelow, J.F. Mas, F. Escobar (eds.). *Geomatic Approaches for Modeling Land Change Scenarios* içinde (ss. 451-455), Cham: Springer, doi: https://doi.org/10.1007/978-3-319-60801-3_27

Taye, M.M. (2023). Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions. *Computers*, 12(5), 1-26, doi: https://doi.org/10.3390/computers12050091

Teich, M. (2015). *The Scientific Revolution Revisited*. Cambridge: Open Book Publishers, doi: http://dx.doi.org/10.11647/OBP.0054

Tennis, J.T. (2015). Archival Metadata for Digital Cultural Heritage Conceptual Provenance, Contextual Forensics, and the Authority of the Found Digital Object. *2015 Digital Heritage, Granada, Spain*, 399-402, doi: 10.1109/DigitalHeritage.2015.7419533

Terven, J. vd. (2025). A Comprehensive Survey of Loss Functions and Metrics in Deep Learning. *Artificial Intelligence Review*, 58, 1-172, doi: https://doi.org/10.1007/s10462-025-11198-7

Tesar, M. (2014). Ethics and Truth in Archival Research. *History of Education*, *44*(1), 101-114, doi: https://doi.org/10.1080/0046760X.2014.918185

Tharwat, A. (2019). Parameter Investigation of Support Vector Machine Xlassifier with Kernel Functions. *Knowledge and Information Systems*, 61, 1269-1302, doi: https://doi.org/10.1007/s10115-019-01335-4

Thibodeau, K. (2018). Computational Archival Practice: Towards a Theory for Archival Engineering. *2018 IEEE International Conference on Big Data*, 2753-2760, doi: 10.1109/BigData.2018.8622174

Thomassen, T. (2001). A First Introduction to Archival Science. *Archival Science*, 1, 373-385, doi: https://doi.org/10.1007/BF02438903

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society B*, 58, 267-288.

Tjandra, A., Sakti, S. ve Nakamura, S. (2017). Compressing Recurrent Neural Network with Tensor Train. *ArXiv:1705.08052*, 1-8, doi: https://doi.org/10.48550/arXiv.1705.08052

Tong, W. vd. (2020). Channel-Attention-Based DenseNet Network for Remote Sensing Image Scene Classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13, 4121-4132, doi: 10.1109/JSTARS.2020.3009352

Toosi, A. vd. (2021). A Brief History of AI: How to Prevent Another Winter (A Critical Review). *PET Clinics*, 16(4), 449-469, doi: https://doi.org/10.1016/j.cpet.2021.07.001

Trace, C.B. (2020). Maintaining Records in Context? Disrupting the Theory and Practice of Archival Classification and Arrangement. *The American Archivist*, 83, 2, 322-372.

Treffry, D. (1998). Theory. *Collins English Dictionary: Millennium Edition, 4th ed.* içinde (4. bs., s. 1587). Glasgow: HarperCollins Publishers.

Triwijoyo, B.K., Adil, A. ve Anggrawan, A. (2021). Convolutional Neural Network with Batch Normalization for Classification of Emotional Expressions Based on Facial Images. *MATRIK: Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, 21(1), 197-204, doi: https://doi.org/10.30812/matrik.v21i1.1526

Tsantekidis, A., Passalis, N. ve Tefas, A. (2022). Recurrent Neural Networks. A. Iosifidis, A. Tefas (eds.). *Deep Learning for Robot Perception and Cognition* içinde (ss. 101-115), London: Elsevier, doi: https://doi.org/10.1016/B978-0-32-385787-1.00010-5

Tschan, R. (2002). A Comparison of Jenkinson and Schellenberg on Appraisal. *The American Archivist*, 65(2), 176-195, doi: https://doi.org/10.17723/aarc.65.2.920w65g3217706l1

Turing, A.M. (1950). Computing Machinery and Intelligence. *Mind*, 59(236), 433-460, doi: https://doi.org/10.1093/mind/LIX.236.433

Turnau, J. vd. (2023). Provenance-Based Explanations for Machine Learning (ML) Models. 2023 IEEE 39th International Conference on Data Engineering Workshops, 40-43, doi: https://doi.org/10.1109/ICDEW58674.2023.00011

UK Government. (t.y.). Introduction to the AI Safety Summit. 15.01.2025 tarihinde https://assets.publishing.service.gov.uk/media/65255655244f8e00138e7362/introduction_to_the_ai_safety_summit.pdf adresinden erişildi.

Vadlamani, S.L., Shafiq, M.O. ve Baysal, O. (2022). Using Machine Learning to Analyze and Predict Entry Patterns of Low-Cost Airlines: A Study of Southwest Airlines. *Machine Learning with Applications*, 10, 1-15, doi: https://doi.org/10.1016/j.mlwa.2022.100410

Valdez, J. (2014). *The Snow Cone Diaries: A Philosopher's Guide to the Information Age*. Bloomington: AuthorHouse LLC.

Vallés-Pérez, I. vd. (2023). Empirical Study of the Modulus as Activation Function in Computer Vision Applications. *Engineering Applications of Artificial Intelligence*, 120, 1-8, doi: https://doi.org/10.1016/j.engappai.2023.105863

van Dyk, D.A. ve Meng, X.L. (2001). The Art of Data Augmentation. *Journal of Computational and Graphical Statistics*, 10(1), 1-50.

van Garderen, P. (2010). Archivematica: Using Micro-Services and Open-Source Software to Deliver a Comprehensive Digital Curation Solution. *IPRES 2010 Session 4b Preservation Services*, 1-5, 10.03.2025 tarihinde https://vangarderen.net/assets/pdfs/iPres2010_Using_MicroServices_and_Open_Source_Software.pdf adresinden erişildi.

Vasanthakumari, R.K., Nair, R.V. ve Krishnappa, V.G. (2023). Improved Learning by Using a Modified Activation Function of a Convolutional Neural Network in Multi-Spectral Image Classification. *Machine Learning with Applications*, 14, 1-13, doi: https://doi.org/10.1016/j.mlwa.2023.100502

Viano, L. vd. (2020). Robust Inverse Reinforcement Learning under Transition Dynamics Mismatch. *ArXiv:2007.01174*, 1-45, doi: https://doi.org/10.48550/arXiv.2007.01174

Vijaya, N. Sharma, S. ve Batra, N. (2019). Comparative Study of Single Linkage, Complete Linkage and Ward Method of Agglomerative Clustering. *Proceedings of 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel* Computing, 568-573, doi: 10.1109/COMITCon.2019.8862232

Villanueva, K.C.N. (2024). Multilayer Perceptrons (MLP) for Nonlinear Classification: A Case Study on Handwritten Digit Recognition. 12.08.2024 tarihinde https://medium.com/@kisetzuu/multilayer-perceptrons-mcp-for-nonlinear-classification-a-case-study-on-handwritten-digit-2925b5d41908 adresinden erişildi.

Vinyals, O. vd. (2019). Grandmaster Level in StarCraft II Using Multi-Agent Reinforcement Learning. *Nature*, 575, 350-354, doi: https://doi.org/10.1038/s41586-019-1724-z

Vogt, W.P. (1993). Independent Variable. *Dictionary of Statistics and Methodology: A Nontechnical Guide for the Social Sciences* içinde (3.bs., s. 151). Thousand Oaks: Sage Publications.

Vujovic, Z. (2021). Classification Model Evaluation Metrics. *International Journal of Advanced Computer Science and Applications*, 12(6), 599-606, doi: 10.14569/IJACSA.2021.0120670

Wacker, J.G. (1998). A Definition of Theory: Research Guidelines for Different Theory-Building Research Methods in Operations Management. *Journal of Operations Management*, 16, 361-385, doi: https://doi.org/10.1016/S0272-6963(98)00019-9

Walchshofer, C. vd. (2023). Provectories: Embedding-Based Analysis of Interaction Provenance Data. *IEEE Transactions on Visualization and Computer Graphics*, 29(12), 4916-4831, doi: 10.1109/TVCG.2021.3135697

Wamba, S.F. vd. (2021). Industry Experiences of Artificial Intelligence (AI): Benefits and Challenges in Operations and Supply Chain Management. *Production Planning & Control*, 33(16), 1493-1497, doi: https://doi.org/10.1080/09537287.2021.1882695

Wang, D. vd. (2019). Comprehensive Eye Diagram Analysis: A Transfer Learning Approach. *IEEE Photonics Journal*, 11(6), 1-19, doi: 10.1109/JPHOT.2019.2947705

Wang, G., Qin, W. ve Wang, Y. (2021). Cyclic Weighted k-means Method with Application to Time-of-Day Interval Partition. *Sustainability*, 13(9), 1-13, doi: https://doi.org/10.3390/su13094796

Wang, J. ve Biljecki, F. (2022). Unsupervised Machine Learning in Urban Studies: A Systematic Review of Applications. *Cities*, 129, 1-20, doi: https://doi.org/10.1016/j.cities.2022.103925

Wang, L. vd. (2023). Residual Depth Feature-Extraction Network for Infrared Small-Target Detection. *Electronics*, 12(12), 1-20, doi: https://doi.org/10.3390/electronics12122568

Wang, Q. vd. (2019). Overview of Logistic Regression Model Analysis and Application. *Chinese Journal of Preventive Medicine*, 53(9), 955-960, doi: http://dx.doi.org/10.3760/cma.j.issn.0253-9624.2019.09.018

Wang, S. (2019). Introduction to Machine Learning Lecture 6: Decision Trees and Random Forests. 22.05.2024 tarihinde https://web.stanford.edu/class/cme250/files/cme250_lecture6.pdf adresinden erişildi.

Wang, Y. ve Chiew, V. (2010). On the Cognitive Process of Human Problem Solving. *Cognitive Systems Research*, 11(1), 81-92, doi: https://doi.org/10.1016/j.cogsys.2008.08.003

Ward, M. (1995). A Definition of Abstraction. *Journal of Software Maintenance: Research and Practice*, 7(6), 1-6, doi: https://doi.org/10.1002/smr.4360070606

Wei, J. ve He, X. (2023). Support Vector Regression Model with Variant Tolerance. *Measurement and Control*, 56(9-10), doi: https://doi.org/10.1177/00202940231180620

Weidling, T. (2013). The Oldest Archival Science: Jacob Von Rammingen's Two Manuals of Registry and Archival Management from 1571, and a Monography on Archives from 1632 by Baldassare Bonifacio. *Scandinavian Journal of History*, 38(2), 270-271, doi: https://doi.org/10.1080/03468755.2013.776405

Weiss, K., Khoshgoftaar, T.M. ve Wang, D. (2016). A Survey of Transfer Learning. *Journal of Big Data*, 3, 1-40, doi: https://doi.org/10.1186/s40537-016-0043-6

Weiss, R. vd. (2022). Applications of Neural Networks in Biomedical Data Analysis. *Biomedicines*, 10(7), 1-30, doi: https://doi.org/10.3390/biomedicines10071469

Weizenbaum, J. (1966). ELIZA - A Computer Program for the Study of Natural Language Communication between Man and Machine. *Communications of the ACM*, 9(1), 36-45, doi: https://doi.org/10.1145/365153.365168

Weld, K. (2014). *Paper Cadavers: The Archives of Dictatorship in Guatemala*. Durham: Duke University Press.

Werbos, P. (1990). Backpropagation Through Time: What it does and How to do it. *Proceedings of the IEEE*, 78, 1550-1560, doi: https://doi.org/10.1109/5.58337

Widodo, S., Brawijaya, H. ve Samudi, S. (2022). Stratified K-fold Cross Validation Optimization on Machine Learning for Prediction. *Sinkron: Jurnal Penelitian Teknik Informatika*, 6(4), 2407-2414, doi: https://doi.org/10.33395/sinkron.v7i4.11792

Wieringa, M. (2017). The Fragility of Digital Media Content: On Preservation and Loss: Sketching the Pilgrimage of Future Scholars to Recover Our Digital Vellum. *Junctions*, 2(2), 27-38, doi: 10.33391/jgjh.33

Wijaya, Y.A. vd. (2021). Davies Bouldin Index Algorithm for Optimizing Clustering Case Studies Mapping School Facilities. *TEM Journal*, 10(3), 1099-1103, doi: 10.18421/TEM103-13

Wild, G. vd. (2021). Machine Learning for Air Transport Planning and Management. *ArXiv:2112.01301*, 1-9, doi: https://doi.org/10.48550/arXiv.2112.01301

Williams, A.E. (2022). Approximating an Artificial General Intelligence or a General Collective Intelligence. *International Journal of Collaborative Intelligence*, 2(3), 210-223, doi: https://doi.org/10.1504/IJCI.2021.122708

Williams, M.N., Grajales, C.A. ve Kurkiewicz, D. (2013). Assumptions of Multiple Regression: Correcting Two Misconceptions. *Practical Assessment, Research and Evaluation*, 18(1), 1-14, doi: https://doi.org/10.7275/55hn-wk47

Wing, J.M. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33-35 doi: https://doi.org/10.1145/1118178.1118215

Woods, C. (2010). Visible Language: The Earliest Writing Systems. C. Woods (ed). *Visible Language: Inventions of Writing in the Ancient Middle East and Beyond* içinde (ss. 15-28). Chicago: The Oriental Institute of the University of Chicago.

Wu, Y. (2014). Introduction to Neural Network Algorithm. 15.10.2024 tarihinde https://einsteinmed.edu/uploadedFiles/labs/Yaohao-Wu/Lecture%209.pdf adresinden erişildi.

Wu, Y. ve Liu, L. (2023). Selecting and Composing Learning Rate Policies for Deep Neural Networks. *ACM Transactions on Intelligent Systems and Technology*, 14(2), 1-25, doi: https://doi.org/10.1145/3570508

Xia, G.S. vd. (2017). AID: A Benchmark Dataset for Performance Evaluation of Aerial Scene Classification. *ArXiv:1608.05167*, 1-25, doi: https://doi.org/10.48550/arXiv.1608.05167

Xu, D. vd. (2021). Convergence of the RMSProp Deep Learning Method with Penalty for Nonconvex Optimization. *Neural Networks*, 139, 17-23, doi: https://doi.org/10.1016/j.neunet.2021.02.011

Xu, N. vd. (2019). Dual-Stream Recurrent Neural Network for Video Captioning. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(8), 2482-2493, doi: 10.1109/TCSVT.2018.2867286

Yadav, P. vd. (2023). Exploring Hyper-Parameters and Feature Selection for Predicting Non-Communicable Chronic Disease Using Stacking Classifier. *IEEE Access*, 11, 80030-80055, doi: 10.1109/ACCESS.2023.3299332

Yakel, E. vd. (2011). Digital Curation for Digital Natives. J*ournal of Education for Library and Information Science*, 52(1), 23-31.

Yalçınkaya, B. (2014). *e-Devlet Üstveri Standardının Oluşturulması ve Türkiye Için Modellenmesi*. (Yayımlanmamış Doktora Tezi). Marmara Üniversitesi, Türkiyat Araştırmaları Enstitüsü, İstanbul.

Yalçınkaya, B. (2017). Elektronik Belgelerin Geleceği: Standartlaşmaya Yönelik Bir Model Olarak Üstveri. Ö. Külcü, T. Çakmak, Eroğlu, Ş. (eds.), *Kamusal Alan Olarak Bilgi Merkezleri ve Yenilikçi Yaklaşımlar* içinde (ss. 221-261). İstanbul: Hiperyayın.

Yalçınkaya, B., Gedikli, M.E. ve Cibaroğlu M.O. (2019). Elektronik Belge Yönetim Sisteminde Log Analizi: İstatistiksel Bir Değerlendirme. B. Yalçınkaya, M. A. Ünal, B. Yılmaz, F. Özdemirci (eds.), *Bilgi Yönetimi ve Bilgi Güvenliği: eBelge- eArşiv- eDevlet- Bulut Bilişim-Büyük Veri-Yapay Zekâ* içinde (ss. 61-77). Ankara: Ankara Ünivrsitesi Yayınevi.

Yamashita, R. vd. (2018). Convolutional Neural Networks: An Overview and Application in Radiology. *Insights Imaging*, 9, 611-629, doi: https://doi.org/10.1007/s13244-018-0639-9

Yan, P. vd. (2024). A Comprehensive Survey of Deep Transfer Learning for Anomaly Detection in Industrial Time Series: Methods, Applications and Directions. *ArXiv:2307.05638*, 1-27, doi: https://doi.org/10.48550/arXiv.2307.05638

Yancheva, G. (2024). A Research Project for Mapping the Digital Initiatives of Bulgarian Libraries. *Digital Presentation and Preservation of Cultural and Scientific Heritage*, 14, 273-280, doi: https://doi.org/10.55630/dipp.2024.14.26

Yang M. vd. (2022). News Text Mining-Based Business Sentiment Analysis and its Significance in Economy. *Frontiers in Psychology*, 13, 1-7, doi: 10.3389/fpsyg.2022.918447

Yang, S. ve Berdine, G. (2017). The Receiver Operating Characteristic (ROC) Curve. *The Southwest Respiratory and Critical Care Chronicles*, 5(19), 34-36, doi: https://doi.org/10.12746/swrccc.v5i19.391

Yao, C. (2016). The Application of Computer Science and Technology in Libraries. P. Fournier-Viger (ed.). *Proceedings of the 2016 International Conference on Education, Management, Computer and Society* içinde (ss. 1422-1424), doi: 10.2991/emcs-16.2016.355

Ye, M. vd. (2021). A Lightweight Model of VGG-16 for Remote Sensing Image Classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, 6916-6922, doi: 10.1109/JSTARS.2021.3090085

Yenduri, G. vd. (2024). GPT (Generative Pre-Trained Transformer) - A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges and Future Directions. *IEEE Access*, 12, 54608-54649, 2024, doi: 10.1109/ACCESS.2024.3389497

Yeo, G. (2018). Records, Information and Data: Exploring the Role of Record Keeping in an Information Culture. London: Facet Publishing.

Yeo, G. (2021). Record-Making and Record-Keeping in Early Societies. New York: Routledge.

Yeo, G. (2024). Artificial Intelligence and the Future(s) of Archival Theory and Practice. *Archeion*, 125, 10-32, doi: https://doi.org/10.4467/26581264ARC.24.008.20572

Yıldız, A. (2006). Provenans Sisteminin Amaçları ve Uygulanması. *Arşiv Dünyası*, 8, 7-10.

Yi, D., Ahn, J. ve Ji, S. (2020). An Effective Optimization Method for Machine Learning Based on ADAM. *Applied Sciences*, 10(3), 1-20, doi: https://doi.org/10.3390/app10031073

Ying, X. (2018). An Overview of Overfitting and its Solutions. *Journal of Physics: Conference Series*, 1168, 1-7, doi: 10.1088/1742-6596/1168/2/022022

Yosinski, J. vd. (2014). How Transferable are Features in Deep Neural Networks? *ArXiv:1411.1792*, 1-14, doi: https://doi.org/10.48550/arXiv.1411.1792

Yu, H. vd. (2011). Advantages of Radial Basis Function Networks for Dynamic System Design. *IEEE Transactions on Industrial Electronics*, 58(12), 5438-5450, doi: 10.1109/TIE.2011.2164773

Zafar, A. vd. (2022). A Comparison of Pooling Methods for Convolutional Neural Networks. *Applied Sciences*, 12(17), 1-21, doi: https://doi.org/10.3390/app12178643

Zakharov, V. (1998). Developing of a Computerized Information System on Conservation and Preservation. J. Whiffin ve J. Havermans (eds.). *Library Preservation and Conservation in the '90s: Proceedings of the Satellite Meeting of the IFLA Section on Preservation and Conservation* içinde (ss. 175-176). Berlin: De Gruyter Saur, doi: https://doi.org/10.1515/9783110954128.175

Zelenyj, D. (1996). Linchpin Imperilled: The Functional Interpretation of Series and the Principle of Respect des Fonds. *Archivaria*, 42, 126-136.

Zhang, H. (2004). The Optimality of Naive Bayes. 17. Uluslararası Florida Yapay Zeka Araştırma Topluluğu Konferansı (FLAIRS 2004) Bildiriler Kitabı içinde (ss. 1-6). California: The AAAI Press.

Zhao, L. ve Zhang, Z. (2024). A Improved Pooling Method for Convolutional Neural Networks. *Scientific Reports*, 14, 1-22, doi: https://doi.org/10.1038/s41598-024-51258-6

Zhao, Z. vd. (2024). A Comparison Review of Transfer Learning and Self-Supervised Learning: Definitions, Applications Advantages and Limitations. *Expert Systems With Applications*, 242, 1-38, doi: https://doi.org/10.1016/j.eswa.2023.122807

Zhou, X. vd. (2022). *Deep Learning on Edge Computing Devices: Design Challenges of Algorithm and Architecture*. Oxford: Elsevier, doi: https://doi.org/10.1016/B978-0-32-385783-3.00009-0

Zhuang, F. vd. (2020). A Comprehensive Survey on Transfer Learning. *ArXiv:1911.02685*, 1-31, doi: https://doi.org/10.48550/arXiv.1911.02685

Znamenskaya, I.A. ve Doroshchenko, I. (2021). Edge Detection and Machine Learning for Automatic Flow Structures Detection and Tracking on Schlieren and Shadowgraph Images. *Journal of Flow Visualization and Image Processing*, 28(4), 1-26, doi: 10.1615/JFlowVisImageProc.2021037690

Zohuri, B. (2023). Artificial Super Intelligence (ASI): The Evolution of AI Beyond Human Capacity. *Current Trends in Engineering Science*, 3(6), 1-5, doi: 10.54026/CTES/1049

Zou, H. ve Hastie, T. (2005). Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society Series B*, 67, 301-320.

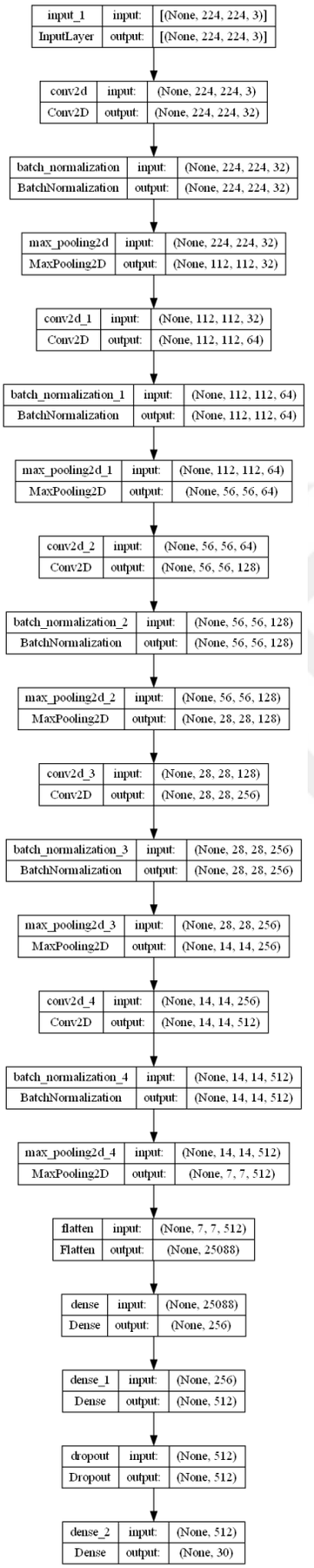
Zucco, C. vd. (2019). Sentiment Analysis for Mining Texts and Social Networks Data: Methods and Tools. *WIREs Data Mining and Knowledge Discovery*, 10(1), 1-32, doi: https://doi.org/10.1002/widm.1333

EKLER

Ek 1. Temel CNN Modeli Yapısı Python Gösterimi

```
img_shape = (224,224,3)
num_classes = len(classlist)
inputs = tf.keras.Input(shape=img_shape)

x = tf.keras.layers.Conv2D(32, (3,3), activation="relu", padding="same")(inputs)
x = BatchNormalization()(x)
x = MaxPooling2D(pool_size=(2,2))(x)

x = tf.keras.layers.Conv2D(64, (3,3), activation="relu", padding="same")(x)
x = BatchNormalization()(x)
x = MaxPooling2D(pool_size=(2,2))(x)

x = tf.keras.layers.Conv2D(128, (3,3), activation="relu", padding="same")(x)
x = BatchNormalization()(x)
x = MaxPooling2D(pool_size=(2,2))(x)

x = tf.keras.layers.Conv2D(256, (3,3), activation="relu", padding="same")(x)
x = BatchNormalization()(x)
x = MaxPooling2D(pool_size=(2,2))(x)

x = tf.keras.layers.Conv2D(512, (3,3), activation="relu", padding="same")(x)
x = BatchNormalization()(x)
x = MaxPooling2D(pool_size=(2,2))(x)

x = Flatten()(x)

x = Dense(256, activation="relu")(x)
x = Dense(512, activation="relu")(x)

x = Dropout(rate=0.5)(x)

outputs = Dense(len(classlist), activation="softmax")(x)
model = Model(inputs, outputs)
```

Ek 2. Temel CNN Modeli Katmanları



Ek 3. Veri Seti 2 Normalizasyon

```python
# Normalizasyon
def load_image(filepath, label):
    image = tf.io.read_file(filepath)
    image = tf.image.decode_jpeg(image, channels=3)
    image = tf.image.resize(image, (224 ,224)) / 255.0
    return image, label
```

Ek 4. Temel CNN Modeli Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	1.0000	0.6430	0.7827	28
cayir	0.9722	0.9211	0.9459	38
ciplak_arazi	0.8049	0.9429	0.8684	35
col	0.9412	0.8000	0.8649	20
dag	0.6750	0.7941	0.7297	34
depolama_tanklari	0.7674	0.7021	0.7333	47
golet	0.7273	0.8276	0.7742	29
havaalani	0.7931	0.6970	0.7419	33
kilise	0.7647	0.6500	0.7027	20
kopru	0.8947	0.8947	0.8947	38
liman	0.9143	0.8649	0.8889	37
merkez	0.7692	0.6667	0.7143	15
meydan	0.6452	0.6061	0.6250	33
nehir	0.6842	0.7429	0.7123	35
okul	0.5833	0.2188	0.3182	32
orman	1.0000	1.0000	1.0000	29
orta_yerlesim	0.5476	0.9200	0.6866	25
otopark	0.9737	0.9250	0.9487	40
oyun_alani	0.7347	0.8571	0.7912	42
park	0.5000	0.7500	0.6000	28
plaj	0.9459	0.8750	0.9091	40
sanayi_bolgesi	0.4386	0.6944	0.5376	36
seyrek_yerlesim	0.9722	0.9459	0.9589	37
stadyum	0.8788	0.8529	0.8657	34
tarim_arazisi	0.8000	0.7111	0.7529	45
tatil_koyu	0.5484	0.4857	0.5152	35
ticari_alan	0.8214	0.5349	0.6479	43
tren_istasyonu	0.7200	0.6923	0.7059	26
viyaduk	0.7742	0.7500	0.7619	32
yogun_yerlesim	0.6829	0.8235	0.7467	34
Doğruluk			**0.7630**	1000
Makro Ortalama	0.7758	0.7597	0.7575	1000
Ağırlıklı Ortalama	0.7800	0.7630	0.7618	1000

Ek 5. Veri Artırımı Yapılmış CNN Modeli Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	0.8261	0.8636	0.8444	22
cayir	0.9286	0.9286	0.9286	28
ciplak_arazi	0.9355	0.9355	0.9355	31
col	0.9333	0.9333	0.9333	30
dag	0.9429	0.9706	0.9565	34
depolama_tanklari	1.0000	0.8333	0.9091	36
golet	0.9750	0.9286	0.9512	42
havaalani	0.8158	0.8611	0.8378	36
kilise	0.8889	1.0000	0.9412	24
kopru	0.8974	0.9722	0.9333	36
liman	0.9744	1.0000	0.9870	38
merkez	0.8400	0.8077	0.8235	26
meydan	0.8276	0.7273	0.7742	33
nehir	0.8864	0.9512	0.9176	41
okul	0.7419	0.7667	0.7541	30
orman	1.0000	0.9600	0.9796	25
orta_yerlesim	0.8889	0.8276	0.8571	29
otopark	0.9512	1.0000	0.9750	39
oyun_alani	0.9000	0.7297	0.8060	37
park	0.8235	0.8000	0.8116	35
plaj	1.0000	0.9500	0.9744	40
sanayi_bolgesi	0.7949	0.7949	0.7949	39
seyrek_yerlesim	1.0000	0.9333	0.9655	30
stadyum	0.8438	0.9310	0.8852	29
tarim_arazisi	0.9429	0.8919	0.9167	37
tatil_koyu	0.8000	0.8276	0.8136	29
ticari_alan	0.8421	0.9143	0.8767	35
tren_istasyonu	0.8000	0.7692	0.7843	26
viyaduk	0.8864	0.9286	0.9070	42
yogun_yerlesim	0.8696	0.9756	0.9195	41
Doğruluk			**0.8930**	1000
Makro Ortalama	0.8919	0.8904	0.8898	1000
Ağırlıklı Ortalama	0.8947	0.8930	0.8924	1000

Ek 6. %90-%5-%5 Bölümlendirmesi Adam Opt. 1. Model Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	1.0000	0.9091	0.9524	11
cayir	0.9333	1.0000	0.9655	14
ciplak_arazi	0.8235	0.9333	0.8750	15
col	0.9286	0.8667	0.8966	15
dag	0.8095	1.0000	0.8947	17
depolama_tanklari	0.7895	0.8333	0.8108	18
golet	1.0000	0.9524	0.9756	21
havaalani	0.7647	0.7222	0.7429	18
kilise	0.8182	0.7500	0.7826	12
kopru	0.9375	0.8333	0.8824	18
liman	0.9500	1.0000	0.9744	19
merkez	0.6667	0.6154	0.6400	13
meydan	0.7143	0.5882	0.6452	17
nehir	1.0000	0.8571	0.9231	21
okul	0.5882	0.6667	0.6250	15
orman	1.0000	0.8462	0.9167	13
orta_yerlesim	0.8571	0.8571	0.8571	14
otopark	0.9500	1.0000	0.9744	19
oyun_alani	0.9444	0.8947	0.9189	19
park	0.8421	0.9412	0.8889	17
plaj	0.8000	1.0000	0.8889	20
sanayi_bolgesi	0.7895	0.7895	0.7895	19
seyrek_yerlesim	0.9333	0.9333	0.9333	15
stadyum	0.9231	0.8571	0.8889	14
tarim_arazisi	0.9412	0.8889	0.9143	18
tatil_koyu	0.8182	0.6000	0.6923	15
ticari_alan	0.8500	0.9444	0.8947	18
tren_istasyonu	0.9000	0.6923	0.7826	13
viyaduk	0.8400	1.0000	0.9130	21
yogun_yerlesim	0.9545	1.0000	0.9767	21
Doğruluk			**0.8680**	500
Makro Ortalama	0.8689	0.8591	0.8605	500
Ağırlıklı Ortalama	0.8712	0.8680	0.8662	500

Ek 7. %90-%5-%5 Bölümlendirmesi Adam Opt. 2. Model Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	0.9091	0.9091	0.9091	11
cayir	0.9286	0.9286	0.9286	14
ciplak_arazi	0.8333	1.0000	0.9091	15
col	1.0000	0.8667	0.9286	15
dag	0.8947	1.0000	0.9444	17
depolama_tanklari	0.9444	0.9444	0.9444	18
golet	1.0000	0.9048	0.9500	21
havaalani	0.8095	0.9444	0.8718	18
kilise	0.8000	1.0000	0.8889	12
kopru	1.0000	0.9444	0.9714	18
liman	0.9500	1.0000	0.9744	19
merkez	1.0000	0.7692	0.8696	13
meydan	0.9167	0.6471	0.7586	17
nehir	0.9524	0.9524	0.9524	21
okul	0.6667	0.8000	0.7273	15
orman	1.0000	0.8462	0.9167	13
orta_yerlesim	0.9286	0.9286	0.9286	14
otopark	1.0000	1.0000	1.0000	19
oyun_alani	0.8571	0.9474	0.9000	19
park	0.9412	0.9412	0.9412	17
plaj	0.9524	1.0000	0.9756	20
sanayi_bolgesi	0.8889	0.8421	0.8649	19
seyrek_yerlesim	1.0000	1.0000	1.0000	15
stadyum	1.0000	0.8571	0.9231	14
tarim_arazisi	0.9444	0.9444	0.9444	18
tatil_koyu	0.9286	0.8667	0.8966	15
ticari_alan	0.8889	0.8889	0.8889	18
tren_istasyonu	0.9000	0.6923	0.7826	13
viyaduk	0.9091	0.9524	0.9302	21
yogun_yerlesim	0.8333	0.9524	0.8889	21
Doğruluk			**0.9140**	500
Makro Ortalama	0.9193	0.9090	0.9103	500
Ağırlıklı Ortalama	0.9199	0.9140	0.9135	500

Ek 8. %80-%10-%10 Bölümlendirmesi Adamax Opt. 1. Model Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	0.9500	0.8636	0.9048	22
cayir	0.9333	1.0000	0.9655	28
ciplak_arazi	0.9032	0.9032	0.9032	31
col	0.9000	0.9000	0.9000	30
dag	0.9706	0.9706	0.9706	34
depolama_tanklari	0.9118	0.8611	0.8857	36
golet	1.0000	0.9048	0.9500	42
havaalani	0.7442	0.8889	0.8101	36
kilise	0.8846	0.9583	0.9200	24
kopru	0.9730	1.0000	0.9863	36
liman	0.9744	1.0000	0.9870	38
merkez	0.9167	0.8462	0.8800	26
meydan	0.8571	0.7273	0.7869	33
nehir	0.9048	0.9268	0.9157	41
okul	0.7188	0.7667	0.7419	30
orman	0.9600	0.9600	0.9600	25
orta_yerlesim	0.8667	0.8966	0.8814	29
otopark	0.9744	0.9744	0.9744	39
oyun_alani	0.8947	0.9189	0.9067	37
park	0.8125	0.7429	0.7761	35
plaj	1.0000	0.9750	0.9873	40
sanayi_bolgesi	0.8421	0.8205	0.8312	39
seyrek_yerlesim	1.0000	0.9333	0.9655	30
stadyum	0.9630	0.8966	0.9286	29
tarim_arazisi	0.9189	0.9189	0.9189	37
tatil_koyu	0.7778	0.7241	0.7500	29
ticari_alan	0.9091	0.8571	0.8824	35
tren_istasyonu	0.8750	0.8077	0.8400	26
viyaduk	0.8750	1.0000	0.9333	42
yogun_yerlesim	0.8511	0.9756	0.9091	41
Doğruluk			**0.9010**	1000
Makro Ortalama	0.9021	0.8973	0.8984	1000
Ağırlıklı Ortalama	0.9029	0.9010	0.9006	1000

Ek 9. %80-%10-%10 Bölümlendirmesi Adamax Opt. 2. Model Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	1.0000	0.9545	0.9767	22
cayir	0.9310	0.9643	0.9474	28
ciplak_arazi	0.9375	0.9677	0.9524	31
col	0.9643	0.9000	0.9310	30
dag	0.9444	1.0000	0.9714	34
depolama_tanklari	0.9714	0.9444	0.9577	36
golet	1.0000	0.9286	0.9630	42
havaalani	0.8500	0.9444	0.8947	36
kilise	0.8846	0.9583	0.9200	24
kopru	0.8000	0.8889	0.8421	36
liman	0.9744	1.0000	0.9870	38
merkez	0.8261	0.7308	0.7755	26
meydan	0.8333	0.7576	0.7937	33
nehir	0.9231	0.8780	0.9000	41
okul	0.7273	0.8000	0.7619	30
orman	0.9600	0.9600	0.9600	25
orta_yerlesim	0.9615	0.8621	0.9091	29
otopark	0.9744	0.9744	0.9744	39
oyun_alani	0.9444	0.9189	0.9315	37
park	0.8182	0.7714	0.7941	35
plaj	0.9756	1.0000	0.9877	40
sanayi_bolgesi	0.8462	0.8462	0.8462	39
seyrek_yerlesim	1.0000	0.9667	0.9831	30
stadyum	0.9615	0.8621	0.9091	29
tarim_arazisi	0.9459	0.9459	0.9459	37
tatil_koyu	0.8276	0.8276	0.8276	29
ticari_alan	0.8947	0.9714	0.9315	35
tren_istasyonu	0.8400	0.8077	0.8235	26
viyaduk	0.9535	0.9762	0.9647	42
yogun_yerlesim	0.9091	0.9756	0.9412	41
Doğruluk			**0.9130**	1000
Makro Ortalama	0.9127	0.9095	0.9101	1000
Ağırlıklı Ortalama	0.9143	0.9130	0.9128	1000

Ek 10. %90-%5-%5 Bölümlendirmesi Adagrad Opt. 1. Model Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	0.8462	1.0000	0.9167	11
cayir	0.7778	1.0000	0.8750	14
ciplak_arazi	0.7500	0.8000	0.7742	15
col	0.8750	0.9333	0.9032	15
dag	0.8333	0.8824	0.8571	17
depolama_tanklari	0.8421	0.8889	0.8649	18
golet	0.9500	0.9048	0.9268	21
havaalani	0.8333	0.8333	0.8333	18
kilise	0.8333	0.8333	0.8333	12
kopru	0.9444	0.9444	0.9444	18
liman	1.0000	0.9474	0.9730	19
merkez	0.8000	0.6154	0.6957	13
meydan	0.8000	0.4706	0.5926	17
nehir	0.8889	0.7619	0.8205	21
okul	0.6429	0.6000	0.6207	15
orman	0.9231	0.9231	0.9231	13
orta_yerlesim	0.8125	0.9286	0.8667	14
otopark	0.9474	0.9474	0.9474	19
oyun_alani	0.8333	0.7895	0.8108	19
park	0.7222	0.7647	0.7429	17
plaj	0.9474	0.9000	0.9231	20
sanayi_bolgesi	0.7273	0.8421	0.7805	19
seyrek_yerlesim	1.0000	0.9333	0.9655	15
stadyum	0.8333	0.7143	0.7692	14
tarim_arazisi	0.6800	0.9444	0.7907	18
tatil_koyu	0.9231	0.8000	0.8571	15
ticari_alan	0.9444	0.9444	0.9444	18
tren_istasyonu	0.7778	0.5385	0.6364	13
viyaduk	0.7308	0.9048	0.8085	21
yogun_yerlesim	1.0000	0.9524	0.9756	21
Doğruluk			**0.8460**	500
Makro Ortalama	0.8473	0.8414	0.8391	500
Ağırlıklı Ortalama	0.8512	0.8460	0.8436	500

Ek 11. %90-%5-%5 Bölümlendirmesi Adagrad Opt. 2. Model Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	0.9091	0.9091	0.9091	11
cayir	0.9286	0.9286	0.9286	14
ciplak_arazi	0.7500	1.0000	0.8571	15
col	1.0000	0.8000	0.8889	15
dag	0.8889	0.9412	0.9143	17
depolama_tanklari	0.8824	0.8333	0.8571	18
golet	1.0000	0.8095	0.8947	21
havaalani	0.8235	0.7778	0.8000	18
kilise	0.7500	1.0000	0.8571	12
kopru	0.6957	0.8889	0.7805	18
liman	0.9500	1.0000	0.9744	19
merkez	1.0000	0.3846	0.5556	13
meydan	0.8462	0.6471	0.7333	17
nehir	0.8500	0.8095	0.8293	21
okul	0.7500	0.6000	0.6667	15
orman	0.9167	0.8462	0.8800	13
orta_yerlesim	0.8571	0.8571	0.8571	14
otopark	0.9500	1.0000	0.9744	19
oyun_alani	0.8333	0.7895	0.8108	19
park	0.9412	0.9412	0.9412	17
plaj	0.9524	1.0000	0.9756	20
sanayi_bolgesi	0.6957	0.8421	0.7619	19
seyrek_yerlesim	1.0000	0.9333	0.9655	15
stadyum	0.9167	0.7857	0.8462	14
tarim_arazisi	0.8421	0.8889	0.8649	18
tatil_koyu	1.0000	0.8000	0.8889	15
ticari_alan	0.8571	1.0000	0.9231	18
tren_istasyonu	0.8889	0.6154	0.7273	13
viyaduk	0.8077	1.0000	0.8936	21
yogun_yerlesim	0.7778	1.0000	0.8750	21
Doğruluk			**0.8620**	500
Makro Ortalama	0.8754	0.8543	0.8544	500
Ağırlıklı Ortalama	0.8729	0.8620	0.8582	500

Ek 12. %80-%10-%10 Bölümlendirmesi RMSProp Opt. 1. Model Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	0.9474	0.8182	0.8780	22
cayir	0.8000	1.0000	0.8889	28
ciplak_arazi	0.7179	0.9032	0.8000	31
col	0.8800	0.7333	0.8000	30
dag	0.7368	0.8235	0.7778	34
depolama_tanklari	0.7500	0.7500	0.7500	36
golet	0.9231	0.8571	0.8889	42
havaalani	0.4694	0.6389	0.5412	36
kilise	0.6667	0.5000	0.5714	24
kopru	0.8947	0.9444	0.9189	36
liman	0.9737	0.9737	0.9737	38
merkez	0.8571	0.6923	0.7660	26
meydan	0.7500	0.5455	0.6316	33
nehir	0.7209	0.7561	0.7381	41
okul	0.6250	0.3333	0.4348	30
orman	1.0000	0.8800	0.9362	25
orta_yerlesim	0.7000	0.7241	0.7119	29
otopark	1.0000	0.9231	0.9600	39
oyun_alani	0.8333	0.8108	0.8219	37
park	0.7143	0.7143	0.7143	35
plaj	1.0000	0.8750	0.9333	40
sanayi_bolgesi	0.7368	0.7179	0.7273	39
seyrek_yerlesim	1.0000	0.9333	0.9655	30
stadyum	0.9130	0.7241	0.8077	29
tarim_arazisi	0.6471	0.8919	0.7500	37
tatil_koyu	0.5217	0.4138	0.4615	29
ticari_alan	0.6500	0.7429	0.6933	35
tren_istasyonu	0.6129	0.7308	0.6667	26
viyaduk	0.7308	0.9048	0.8085	42
yogun_yerlesim	0.8810	0.9024	0.8916	41
Doğruluk			**0.7810**	1000
Makro Ortalama	0.7885	0.7720	0.7736	1000
Ağırlıklı Ortalama	0.7901	0.7810	0.7794	1000

Ek 13. %80-%10-%10 Bölümlendirmesi RMSProp Opt. 2. Model Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	1.0000	0.8182	0.9000	22
cayir	0.8710	0.9643	0.9153	28
ciplak_arazi	0.9310	0.8710	0.9000	31
col	0.8966	0.8667	0.8814	30
dag	0.8718	1.0000	0.9315	34
depolama_tanklari	1.0000	0.9167	0.9565	36
golet	0.9737	0.8810	0.9250	42
havaalani	0.8378	0.8611	0.8493	36
kilise	0.7931	0.9583	0.8679	24
kopru	0.8333	0.8333	0.8333	36
liman	1.0000	0.9737	0.9867	38
merkez	0.8182	0.6923	0.7500	26
meydan	0.8077	0.6364	0.7119	33
nehir	0.9268	0.9268	0.9268	41
okul	0.5833	0.7000	0.6364	30
orman	0.9600	0.9600	0.9600	25
orta_yerlesim	0.9231	0.8276	0.8727	29
otopark	1.0000	0.9744	0.9870	39
oyun_alani	0.9118	0.8378	0.8732	37
park	0.8387	0.7429	0.7879	35
plaj	1.0000	0.9750	0.9873	40
sanayi_bolgesi	0.8889	0.8205	0.8533	39
seyrek_yerlesim	0.8750	0.9333	0.9032	30
stadyum	0.8667	0.8966	0.8814	29
tarim_arazisi	0.8718	0.9189	0.8947	37
tatil_koyu	0.7241	0.7241	0.7241	29
ticari_alan	0.8205	0.9143	0.8649	35
tren_istasyonu	0.7931	0.8846	0.8364	26
viyaduk	0.9333	1.0000	0.9655	42
yogun_yerlesim	0.8723	1.0000	0.9318	41
Doğruluk			**0.8820**	1000
Makro Ortalama	0.8808	0.8770	0.8765	1000
Ağırlıklı Ortalama	0.8858	0.8820	0.8817	1000

Ek 14. %80-%10-%10 Bölümlendirmesi SGD Opt. 1. Model Sınıflandırma Raporu

	Kesinlik	Duyarlılık	F1	Örnek Sayısı
beyzbol_sahasi	0.8421	0.7273	0.7805	22
cayir	0.7778	1.0000	0.8750	28
ciplak_arazi	0.7368	0.9032	0.8116	31
col	0.8519	0.7667	0.8070	30
dag	0.7105	0.7941	0.7500	34
depolama_tanklari	0.7143	0.8333	0.7692	36
golet	0.7500	0.8571	0.8000	42
havaalani	0.6429	0.7500	0.6923	36
kilise	0.9130	0.8750	0.8936	24
kopru	1.0000	0.9444	0.9714	36
liman	0.9730	0.9474	0.9600	38
merkez	0.8421	0.6154	0.7111	26
meydan	0.8696	0.6061	0.7143	33
nehir	0.6977	0.7317	0.7143	41
okul	0.4737	0.3000	0.3673	30
orman	0.8800	0.8800	0.8800	25
orta_yerlesim	0.7586	0.7586	0.7586	29
otopark	0.9730	0.9231	0.9474	39
oyun_alani	0.9062	0.7838	0.8406	37
park	0.6154	0.6857	0.6486	35
plaj	0.9730	0.9000	0.9351	40
sanayi_bolgesi	0.7368	0.7179	0.7273	39
seyrek_yerlesim	0.8750	0.9333	0.9032	30
stadyum	0.8571	0.8276	0.8421	29
tarim_arazisi	0.7209	0.8378	0.7750	37
tatil_koyu	0.7273	0.5517	0.6275	29
ticari_alan	0.8438	0.7714	0.8060	35
tren_istasyonu	0.7308	0.7308	0.7308	26
viyaduk	0.7451	0.9048	0.8172	42
yogun_yerlesim	0.8537	0.8537	0.8537	41
Doğruluk			**0.7960**	1000
Makro Ortalama	0.7997	0.7904	0.7904	1000
Ağırlıklı Ortalama	0.7999	0.7960	0.7935	1000

Ek 15. %80-%10-%10 Bölümlendirmesi SGD Opt. 2. Model Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	0.8095	0.7727	0.7907	22
cayir	0.8750	1.0000	0.9333	28
ciplak_arazi	0.8056	0.9355	0.8657	31
col	0.8846	0.7667	0.8214	30
dag	0.7750	0.9118	0.8378	34
depolama_tanklari	0.9062	0.8056	0.8529	36
golet	0.8462	0.7857	0.8148	42
havaalani	0.7059	0.6667	0.6857	36
kilise	0.6774	0.8750	0.7636	24
kopru	0.8056	0.8056	0.8056	36
liman	0.9730	0.9474	0.9600	38
merkez	1.0000	0.5385	0.7000	26
meydan	0.6333	0.5758	0.6032	33
nehir	0.8108	0.7317	0.7692	41
okul	0.6667	0.6000	0.6316	30
orman	0.8846	0.9200	0.9020	25
orta_yerlesim	0.7333	0.7586	0.7458	29
otopark	0.9744	0.9744	0.9744	39
oyun_alani	0.8182	0.7297	0.7714	37
park	0.6667	0.8000	0.7273	35
plaj	0.9737	0.9250	0.9487	40
sanayi_bolgesi	0.7949	0.7949	0.7949	39
seyrek_yerlesim	0.8667	0.8667	0.8667	30
stadyum	0.8571	0.8276	0.8421	29
tarim_arazisi	0.7222	0.7027	0.7123	37
tatil_koyu	0.9524	0.6897	0.8000	29
ticari_alan	0.7750	0.8857	0.8267	35
tren_istasyonu	0.9091	0.7692	0.8333	26
viyaduk	0.7273	0.9524	0.8247	42
yogun_yerlesim	0.7959	0.9512	0.8667	41
Doğruluk			**0.8130**	1000
Makro Ortalama	0.8209	0.8089	0.8091	1000
Ağırlıklı Ortalama	0.8202	0.8130	0.8114	1000

Ek 16. Öğrenme Oranı 0,0001 1. Model Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	0.9000	0.8182	0.8571	22
cayir	0.8710	0.9643	0.9153	28
ciplak_arazi	0.9032	0.9032	0.9032	31
col	0.8710	0.9000	0.8852	30
dag	0.9167	0.9706	0.9429	34
depolama_tanklari	0.9143	0.8889	0.9014	36
golet	0.9744	0.9048	0.9383	42
havaalani	0.8684	0.9167	0.8919	36
kilise	0.8800	0.9167	0.8980	24
kopru	0.9730	1.0000	0.9863	36
liman	1.0000	0.9474	0.9730	38
merkez	0.8261	0.7308	0.7755	26
meydan	0.8000	0.7273	0.7619	33
nehir	0.8605	0.9024	0.8810	41
okul	0.7500	0.7000	0.7241	30
orman	1.0000	0.9200	0.9583	25
orta_yerlesim	0.9615	0.8621	0.9091	29
otopark	0.9744	0.9744	0.9744	39
oyun_alani	0.8974	0.9459	0.9211	37
park	0.7073	0.8286	0.7632	35
plaj	0.9744	0.9500	0.9620	40
sanayi_bolgesi	0.8000	0.8205	0.8101	39
seyrek_yerlesim	0.9677	1.0000	0.9836	30
stadyum	0.9630	0.8966	0.9286	29
tarim_arazisi	0.8947	0.9189	0.9067	37
tatil_koyu	0.8000	0.6897	0.7407	29
ticari_alan	0.8857	0.8857	0.8857	35
tren_istasyonu	0.8846	0.8846	0.8846	26
viyaduk	0.9535	0.9762	0.9647	42
yogun_yerlesim	0.9111	1.0000	0.9535	41
Doğruluk			**0.8970**	1000
Makro Ortalama	0.8961	0.8915	0.8927	1000
Ağırlıklı Ortalama	0.8981	0.8970	0.8965	1000

Ek 17. Öğrenme Oranı 0,0001 2. Model Sınıflandırma Raporu

	Kesinlik	Duyarlılık	F1	Örnek Sayısı
beyzbol_sahasi	1.0000	0.8636	0.9268	22
cayir	0.9032	1.0000	0.9492	28
ciplak_arazi	0.9032	0.9032	0.9032	31
col	0.9000	0.9000	0.9000	30
dag	0.9444	1.0000	0.9714	34
depolama_tanklari	0.9706	0.9167	0.9429	36
golet	1.0000	0.9762	0.9880	42
havaalani	0.8462	0.9167	0.8800	36
kilise	0.9200	0.9583	0.9388	24
kopru	0.9211	0.9722	0.9459	36
liman	1.0000	1.0000	1.0000	38
merkez	0.8182	0.6923	0.7500	26
meydan	0.8125	0.7879	0.8000	33
nehir	0.9744	0.9268	0.9500	41
okul	0.7812	0.8333	0.8065	30
orman	0.8889	0.9600	0.9231	25
orta_yerlesim	1.0000	0.8621	0.9259	29
otopark	0.9744	0.9744	0.9744	39
oyun_alani	0.9143	0.8649	0.8889	37
park	0.7714	0.7714	0.7714	35
plaj	1.0000	0.9500	0.9744	40
sanayi_bolgesi	0.8947	0.8718	0.8831	39
seyrek_yerlesim	0.9655	0.9333	0.9492	30
stadyum	0.9643	0.9310	0.9474	29
tarim_arazisi	0.9444	0.9189	0.9315	37
tatil_koyu	0.8000	0.8276	0.8136	29
ticari_alan	0.8500	0.9714	0.9067	35
tren_istasyonu	0.8400	0.8077	0.8235	26
viyaduk	0.9545	1.0000	0.9767	42
yogun_yerlesim	0.9091	0.9756	0.9412	41
Doğruluk			**0.9140**	1000
Makro Ortalama	0.9122	0.9089	0.9094	1000
Ağırlıklı Ortalama	0.9155	0.9140	0.9137	1000

Ek 18. Öğrenme Oranı 0,00001 1. Model Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	0.8500	0.7727	0.8095	22
cayir	0.8000	1.0000	0.8889	28
ciplak_arazi	0.7429	0.8387	0.7879	31
col	0.8235	0.9333	0.8750	30
dag	0.6364	0.8235	0.7179	34
depolama_tanklari	0.7436	0.8056	0.7733	36
golet	0.7826	0.8571	0.8182	42
havaalani	0.5849	0.8611	0.6966	36
kilise	0.8889	0.6667	0.7619	24
kopru	0.9706	0.9167	0.9429	36
liman	0.9250	0.9737	0.9487	38
merkez	0.9091	0.7692	0.8333	26
meydan	0.6129	0.5758	0.5938	33
nehir	0.5952	0.6098	0.6024	41
okul	0.6667	0.3333	0.4444	30
orman	0.9167	0.8800	0.8980	25
orta_yerlesim	0.8214	0.7931	0.8070	29
otopark	0.9714	0.8718	0.9189	39
oyun_alani	0.8387	0.7027	0.7647	37
park	0.7143	0.7143	0.7143	35
plaj	1.0000	0.8750	0.9333	40
sanayi_bolgesi	0.9333	0.7179	0.8116	39
seyrek_yerlesim	0.9600	0.8000	0.8727	30
stadyum	0.8519	0.7931	0.8214	29
tarim_arazisi	0.6222	0.7568	0.6829	37
tatil_koyu	0.6923	0.6207	0.6545	29
ticari_alan	0.7778	0.8000	0.7887	35
tren_istasyonu	0.8333	0.7692	0.8000	26
viyaduk	0.7358	0.9286	0.8211	42
yogun_yerlesim	0.8684	0.8049	0.8354	41
Doğruluk			**0.7890**	1000
Makro Ortalama	0.8023	0.7855	0.7873	1000
Ağırlıklı Ortalama	0.7999	0.7890	0.7880	1000

Ek 19. Öğrenme Oranı 0,00001 2. Model Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	0.7391	0.7727	0.7556	22
cayir	0.8750	1.0000	0.9333	28
ciplak_arazi	0.8235	0.9032	0.8615	31
col	0.9259	0.8333	0.8772	30
dag	0.8108	0.8824	0.8451	34
depolama_tanklari	0.8611	0.8611	0.8611	36
golet	0.9444	0.8095	0.8718	42
havaalani	0.8333	0.8333	0.8333	36
kilise	0.7778	0.8750	0.8235	24
kopru	0.7838	0.8056	0.7945	36
liman	0.9487	0.9737	0.9610	38
merkez	0.8125	0.5000	0.6190	26
meydan	0.7742	0.7273	0.7500	33
nehir	0.7838	0.7073	0.7436	41
okul	0.6667	0.7333	0.6984	30
orman	0.9200	0.9200	0.9200	25
orta_yerlesim	0.9545	0.7241	0.8235	29
otopark	0.9744	0.9744	0.9744	39
oyun_alani	0.8788	0.7838	0.8286	37
park	0.6444	0.8286	0.7250	35
plaj	0.9487	0.9250	0.9367	40
sanayi_bolgesi	0.8108	0.7692	0.7895	39
seyrek_yerlesim	0.9000	0.9000	0.9000	30
stadyum	0.8276	0.8276	0.8276	29
tarim_arazisi	0.8108	0.8108	0.8108	37
tatil_koyu	0.7692	0.6897	0.7273	29
ticari_alan	0.8049	0.9429	0.8684	35
tren_istasyonu	0.8519	0.8846	0.8679	26
viyaduk	0.8889	0.9524	0.9195	42
yogun_yerlesim	0.8636	0.9268	0.8941	41
Doğruluk			**0.8400**	1000
Makro Ortalama	0.8403	0.8359	0.8347	1000
Ağırlıklı Ortalama	0.8437	0.8400	0.8388	1000

Ek 20. Temel EfficientNetB0 Modeli Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	1.0000	0.9091	0.9524	11
cayir	1.0000	1.0000	1.0000	14
ciplak_arazi	0.8889	0.5333	0.6667	15
col	0.8667	0.8667	0.8667	15
dag	0.6400	0.9412	0.7619	17
depolama_tanklari	0.7647	0.7222	0.7429	18
golet	0.9048	0.9048	0.9048	21
havaalani	0.8571	0.6667	0.7500	18
kilise	0.6667	1.0000	0.8000	12
kopru	1.0000	0.7778	0.8750	18
liman	1.0000	1.0000	1.0000	19
merkez	0.6111	0.8462	0.7097	13
meydan	0.8000	0.4706	0.5926	17
nehir	0.6923	0.8571	0.7660	21
okul	0.5714	0.2667	0.3636	15
orman	0.7500	0.9231	0.8276	13
orta_yerlesim	0.7647	0.9286	0.8387	14
otopark	0.9474	0.9474	0.9474	19
oyun_alani	0.8667	0.6842	0.7647	19
park	0.8750	0.8235	0.8485	17
plaj	0.9474	0.9000	0.9231	20
sanayi_bolgesi	0.5926	0.8421	0.6957	19
seyrek_yerlesim	1.0000	0.8667	0.9286	15
stadyum	0.8000	0.8571	0.8276	14
tarim_arazisi	0.6875	0.6111	0.6471	18
tatil_koyu	0.7500	0.8000	0.7742	15
ticari_alan	0.8421	0.8889	0.8649	18
tren_istasyonu	0.6667	0.3077	0.4211	13
viyaduk	0.6552	0.9048	0.7600	21
yogun_yerlesim	0.8000	0.7619	0.7805	21
Doğruluk			**0.7960**	500
Makro Ortalama	0.8070	0.7936	0.7867	500
Ağırlıklı Ortalama	0.8085	0.7960	0.7896	500

Ek 21. Temel EfficientNetB1 Modeli Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	1.0000	0.7727	0.8718	22
cayir	0.9286	0.9286	0.9286	28
ciplak_arazi	0.7576	0.8065	0.7812	31
col	0.8286	0.9667	0.8923	30
dag	0.9167	0.6471	0.7586	34
depolama_tanklari	0.7838	0.8056	0.7945	36
golet	0.9688	0.7381	0.8378	42
havaalani	0.9000	0.7500	0.8182	36
kilise	0.9565	0.9167	0.9362	24
kopru	0.9286	0.7222	0.8125	36
liman	0.8718	0.8947	0.8831	38
merkez	0.6765	0.8846	0.7667	26
meydan	0.7308	0.5758	0.6441	33
nehir	0.8571	0.8780	0.8675	41
okul	0.8750	0.2333	0.3684	30
orman	0.6098	1.0000	0.7576	25
orta_yerlesim	0.7419	0.7931	0.7667	29
otopark	0.8667	1.0000	0.9286	39
oyun_alani	0.7429	0.7027	0.7222	37
park	0.6429	0.7714	0.7013	35
plaj	0.9286	0.9750	0.9512	40
sanayi_bolgesi	0.5455	0.7692	0.6383	39
seyrek_yerlesim	1.0000	1.0000	1.0000	30
stadyum	0.6585	0.9310	0.7714	29
tarim_arazisi	0.8286	0.7838	0.8056	37
tatil_koyu	0.8462	0.7586	0.8000	29
ticari_alan	0.9032	0.8000	0.8485	35
tren_istasyonu	0.5217	0.4615	0.4898	26
viyaduk	0.9167	0.7857	0.8462	42
yogun_yerlesim	0.7059	0.8780	0.7826	41
Doğruluk			**0.7990**	1000
Makro Ortalama	0.8146	0.7977	0.7924	1000
Ağırlıklı Ortalama	0.8173	0.7990	0.7953	1000

Ek 22. Temel EfficientNetB2 Modeli Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	0.9000	0.8182	0.8571	11
cayir	0.8125	0.9286	0.8667	14
ciplak_arazi	0.7778	0.9333	0.8485	15
col	1.0000	0.8000	0.8889	15
dag	0.9167	0.6471	0.7586	17
depolama_tanklari	1.0000	0.6667	0.8000	18
golet	0.8636	0.9048	0.8837	21
havaalani	0.6842	0.7222	0.7027	18
kilise	0.6471	0.9167	0.7586	12
kopru	1.0000	0.8333	0.9091	18
liman	0.7917	1.0000	0.8837	19
merkez	0.8889	0.6154	0.7273	13
meydan	0.8333	0.5882	0.6897	17
nehir	0.9048	0.9048	0.9048	21
okul	0.7273	0.5333	0.6154	15
orman	0.7059	0.9231	0.8000	13
orta_yerlesim	0.8571	0.8571	0.8571	14
otopark	0.9474	0.9474	0.9474	19
oyun_alani	0.9286	0.6842	0.7879	19
park	0.8333	0.8824	0.8571	17
plaj	0.7917	0.9500	0.8636	20
sanayi_bolgesi	0.5000	0.6316	0.5581	19
seyrek_yerlesim	1.0000	1.0000	1.0000	15
stadyum	0.6667	1.0000	0.8000	14
tarim_arazisi	0.8824	0.8333	0.8571	18
tatil_koyu	1.0000	0.7333	0.8462	15
ticari_alan	0.8889	0.8889	0.8889	18
tren_istasyonu	0.7000	0.5385	0.6087	13
viyaduk	0.7407	0.9524	0.8333	21
yogun_yerlesim	0.9048	0.9048	0.9048	21
Doğruluk			**0.8220**	500
Makro Ortalama	0.8365	0.8180	0.8168	500
Ağırlıklı Ortalama	0.8392	0.8220	0.8207	500

Ek 23. Temel EfficientNetB3 Modeli Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	0.7143	0.9091	0.8000	11
cayir	0.9231	0.8571	0.8889	14
ciplak_arazi	0.8571	0.8000	0.8276	15
col	0.8235	0.9333	0.8750	15
dag	0.8824	0.8824	0.8824	17
depolama_tanklari	0.8000	0.8889	0.8421	18
golet	0.9000	0.8571	0.8780	21
havaalani	1.0000	0.7222	0.8387	18
kilise	0.8000	1.0000	0.8889	12
kopru	0.9444	0.9444	0.9444	18
liman	1.0000	1.0000	1.0000	19
merkez	0.8750	0.5385	0.6667	13
meydan	1.0000	0.4706	0.6400	17
nehir	0.9091	0.9524	0.9302	21
okul	0.7500	0.4000	0.5217	15
orman	0.8125	1.0000	0.8966	13
orta_yerlesim	0.9231	0.8571	0.8889	14
otopark	0.9500	1.0000	0.9744	19
oyun_alani	0.6923	0.9474	0.8000	19
park	0.8235	0.8235	0.8235	17
plaj	1.0000	0.9500	0.9744	20
sanayi_bolgesi	0.6000	0.7895	0.6818	19
seyrek_yerlesim	1.0000	0.9333	0.9655	15
stadyum	1.0000	0.6429	0.7826	14
tarim_arazisi	0.8571	1.0000	0.9231	18
tatil_koyu	0.8667	0.8667	0.8667	15
ticari_alan	0.9444	0.9444	0.9444	18
tren_istasyonu	0.9091	0.7692	0.8333	13
viyaduk	0.8333	0.9524	0.8889	21
yogun_yerlesim	0.7308	0.9048	0.8085	21
Doğruluk			**0.8580**	500
Makro Ortalama	0.8707	0.8512	0.8492	500
Ağırlıklı Ortalama	0.8716	0.8580	0.8537	500

Ek 24. Temel EfficientNetV2B0 Modeli Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	1.0000	0.9091	0.9524	11
cayir	0.8667	0.9286	0.8966	14
ciplak_arazi	0.8125	0.8667	0.8387	15
col	0.9231	0.8000	0.8571	15
dag	0.8421	0.9412	0.8889	17
depolama_tanklari	1.0000	0.9444	0.9714	18
golet	0.7308	0.9048	0.8085	21
havaalani	1.0000	0.8889	0.9412	18
kilise	1.0000	0.8333	0.9091	12
kopru	1.0000	0.7778	0.8750	18
liman	0.9474	0.9474	0.9474	19
merkez	0.7333	0.8462	0.7857	13
meydan	0.9286	0.7647	0.8387	17
nehir	0.9545	1.0000	0.9767	21
okul	0.7778	0.4667	0.5833	15
orman	0.6667	0.7692	0.7143	13
orta_yerlesim	1.0000	0.7857	0.8800	14
otopark	0.9048	1.0000	0.9500	19
oyun_alani	0.8750	0.7368	0.8000	19
park	0.7273	0.9412	0.8205	17
plaj	0.9524	1.0000	0.9756	20
sanayi_bolgesi	0.7727	0.8947	0.8293	19
seyrek_yerlesim	1.0000	1.0000	1.0000	15
stadyum	0.8571	0.8571	0.8571	14
tarim_arazisi	0.8947	0.9444	0.9189	18
tatil_koyu	0.8571	0.8000	0.8276	15
ticari_alan	1.0000	0.8889	0.9412	18
tren_istasyonu	0.8889	0.6154	0.7273	13
viyaduk	0.8750	1.0000	0.9333	21
yogun_yerlesim	0.7308	0.9048	0.8085	21
Doğruluk			**0.8740**	500
Makro Ortalama	0.8840	0.8653	0.8685	500
Ağırlıklı Ortalama	0.8833	0.8740	0.8728	500

Ek 25. Temel EfficientNetV2B1 Modeli Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	0.9545	0.9545	0.9545	22
cayir	0.9615	0.8929	0.9259	28
ciplak_arazi	0.9032	0.9032	0.9032	31
col	0.9333	0.9333	0.9333	30
dag	0.7805	0.9412	0.8533	34
depolama_tanklari	1.0000	0.6944	0.8197	36
golet	0.9737	0.8810	0.9250	42
havaalani	0.8788	0.8056	0.8406	36
kilise	0.8333	0.8333	0.8333	24
kopru	0.8857	0.8611	0.8732	36
liman	0.9024	0.9737	0.9367	38
merkez	0.9091	0.7692	0.8333	26
meydan	0.9167	0.6667	0.7719	33
nehir	0.8810	0.9024	0.8916	41
okul	0.8824	0.5000	0.6383	30
orman	0.7742	0.9600	0.8571	25
orta_yerlesim	0.9200	0.7931	0.8519	29
otopark	0.8667	1.0000	0.9286	39
oyun_alani	0.9706	0.8919	0.9296	37
park	0.6667	0.8000	0.7273	35
plaj	0.9744	0.9500	0.9620	40
sanayi_bolgesi	0.7143	0.6410	0.6757	39
seyrek_yerlesim	0.9667	0.9667	0.9667	30
stadyum	0.9310	0.9310	0.9310	29
tarim_arazisi	0.8421	0.8649	0.8533	37
tatil_koyu	0.6571	0.7931	0.7188	29
ticari_alan	0.8571	0.8571	0.8571	35
tren_istasyonu	0.5333	0.9231	0.6761	26
viyaduk	0.8684	0.7857	0.8250	42
yogun_yerlesim	0.7708	0.9024	0.8315	41
Doğruluk			**0.8520**	1000
Makro Ortalama	0.8637	0.8524	0.8509	1000
Ağırlıklı Ortalama	0.8657	0.8520	0.8522	1000

Ek 26. Temel EfficientNetV2B2 Modeli Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	0.9000	0.8182	0.8571	11
cayir	0.8125	0.9286	0.8667	14
ciplak_arazi	0.7692	0.6667	0.7143	15
col	0.7778	0.9333	0.8485	15
dag	0.8235	0.8235	0.8235	17
depolama_tanklari	1.0000	0.8889	0.9412	18
golet	0.9500	0.9048	0.9268	21
havaalani	1.0000	0.8333	0.9091	18
kilise	0.6471	0.9167	0.7586	12
kopru	0.9412	0.8889	0.9143	18
liman	0.9048	1.0000	0.9500	19
merkez	0.8182	0.6923	0.7500	13
meydan	0.9091	0.5882	0.7143	17
nehir	0.9545	1.0000	0.9767	21
okul	0.6000	0.6000	0.6000	15
orman	1.0000	0.9231	0.9600	13
orta_yerlesim	0.9286	0.9286	0.9286	14
otopark	1.0000	0.9474	0.9730	19
oyun_alani	0.7500	0.7895	0.7692	19
park	0.6522	0.8824	0.7500	17
plaj	0.9500	0.9500	0.9500	20
sanayi_bolgesi	0.8235	0.7368	0.7778	19
seyrek_yerlesim	1.0000	1.0000	1.0000	15
stadyum	0.8000	0.8571	0.8276	14
tarim_arazisi	0.8824	0.8333	0.8571	18
tatil_koyu	1.0000	0.6667	0.8000	15
ticari_alan	0.9412	0.8889	0.9143	18
tren_istasyonu	0.6000	0.6923	0.6429	13
viyaduk	0.8333	0.9524	0.8889	21
yogun_yerlesim	0.8750	1.0000	0.9333	21
Doğruluk			**0.8580**	500
Makro Ortalama	0.8615	0.8511	0.8508	500
Ağırlıklı Ortalama	0.8683	0.8580	0.8580	500

Ek 27. Temel EfficientNetV2B3 Modeli Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	0.9091	0.9091	0.9091	11
cayir	1.0000	0.7857	0.8800	14
ciplak_arazi	0.7778	0.9333	0.8485	15
col	0.9091	0.6667	0.7692	15
dag	0.7895	0.8824	0.8333	17
depolama_tanklari	0.8095	0.9444	0.8718	18
golet	0.8696	0.9524	0.9091	21
havaalani	0.8750	0.7778	0.8235	18
kilise	0.8333	0.8333	0.8333	12
kopru	1.0000	1.0000	1.0000	18
liman	1.0000	0.9474	0.9730	19
merkez	0.7692	0.7692	0.7692	13
meydan	0.9091	0.5882	0.7143	17
nehir	0.9524	0.9524	0.9524	21
okul	0.7500	0.4000	0.5217	15
orman	0.8667	1.0000	0.9286	13
orta_yerlesim	1.0000	0.9286	0.9630	14
otopark	0.8261	1.0000	0.9048	19
oyun_alani	1.0000	0.8421	0.9143	19
park	0.8235	0.8235	0.8235	17
plaj	0.9500	0.9500	0.9500	20
sanayi_bolgesi	0.6250	0.7895	0.6977	19
seyrek_yerlesim	1.0000	1.0000	1.0000	15
stadyum	0.8000	0.8571	0.8276	14
tarim_arazisi	0.9412	0.8889	0.9143	18
tatil_koyu	0.9286	0.8667	0.8966	15
ticari_alan	0.9375	0.8333	0.8824	18
tren_istasyonu	0.6429	0.6923	0.6667	13
viyaduk	0.9091	0.9524	0.9302	21
yogun_yerlesim	0.7500	1.0000	0.8571	21
Doğruluk			**0.8660**	500
Makro Ortalama	0.8718	0.8589	0.8588	500
Ağırlıklı Ortalama	0.8740	0.8660	0.8635	500

Ek 28. Hiperparametreler İçin Belirlenen Aralıklar

```python
# L1 Çekirdek, L2 Çekirdek, L2 Bias ve L2 Aktivite Hiperparametreleri
kernel_l1 = hp.Float("kernel_l1", min_value=1e-6, max_value=1e-4, sampling="log")
kernel_l2 = hp.Float("kernel_l2", min_value=1e-6, max_value=1e-3, sampling="log")
bias_l2 = hp.Float("bias_l2", min_value=1e-6, max_value=1e-3, sampling="log")
activity_l2 = hp.Float("activity_l2", min_value=1e-6, max_value=1e-3, sampling="log")
```

```python
# Yoğun Katman Nöron Sayısı
dense_units = hp.Int("dense_units", min_value=128, max_value=512, step=128)
```

```python
# Öğrenme Oranı ve Optimize Edici Hiperparametreleri
learning_rate = hp.Choice("learning_rate", values=[1e-2, 1e-3, 1e-4])
optimizer_name = hp.Choice("optimizer", values=["adam", "adamax", "adagrad", "rmsprop"])

optimizers = {"adam": Adam(learning_rate=learning_rate), "adamax": Adamax(learning_rate=learning_rate),"adagrad": Adagrad(learning_rate=learning_rate),
              "rmsprop": RMSprop(learning_rate=learning_rate)}
opt = optimizers[optimizer_name]
```

Ek 29. Optimize Edilmiş EfficientNetB0 Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	1.0000	0.9091	0.9524	11
cayir	1.0000	0.9286	0.9630	14
ciplak_arazi	1.0000	0.9333	0.9655	15
col	0.9375	1.0000	0.9677	15
dag	1.0000	1.0000	1.0000	17
depolama_tanklari	1.0000	1.0000	1.0000	18
golet	0.9524	0.9524	0.9524	21
havaalani	1.0000	1.0000	1.0000	18
kilise	0.8571	1.0000	0.9231	12
kopru	0.9474	1.0000	0.9730	18
liman	0.9500	1.0000	0.9744	19
merkez	0.9167	0.8462	0.8800	13
meydan	1.0000	0.9412	0.9697	17
nehir	1.0000	1.0000	1.0000	21
okul	0.9091	0.6667	0.7692	15
orman	1.0000	1.0000	1.0000	13
orta_yerlesim	1.0000	1.0000	1.0000	14
otopark	1.0000	1.0000	1.0000	19
oyun_alani	0.8261	1.0000	0.9048	19
park	0.8824	0.8824	0.8824	17
plaj	1.0000	0.9500	0.9744	20
sanayi_bolgesi	0.9500	1.0000	0.9744	19
seyrek_yerlesim	1.0000	1.0000	1.0000	15
stadyum	1.0000	0.8571	0.9231	14
tarim_arazisi	0.9474	1.0000	0.9730	18
tatil_koyu	1.0000	0.9333	0.9655	15
ticari_alan	0.9474	1.0000	0.9730	18
tren_istasyonu	0.9286	1.0000	0.9630	13
viyaduk	1.0000	1.0000	1.0000	21
yogun_yerlesim	1.0000	1.0000	1.0000	21
Doğruluk			**0.9640**	500
Makro Ortalama	0.9651	0.9600	0.9608	500
Ağırlıklı Ortalama	0.9659	0.9640	0.9633	500

Ek 30. Optimize Edilmiş EfficientNetB1 Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	1.0000	1.0000	1.0000	22
cayir	0.9655	1.0000	0.9825	28
ciplak_arazi	0.9667	0.9355	0.9508	31
col	0.9355	0.9667	0.9508	30
dag	1.0000	1.0000	1.0000	34
depolama_tanklari	1.0000	0.9444	0.9714	36
golet	1.0000	0.9762	0.9880	42
havaalani	0.9730	1.0000	0.9863	36
kilise	0.9583	0.9583	0.9583	24
kopru	0.9730	1.0000	0.9863	36
liman	1.0000	1.0000	1.0000	38
merkez	0.9200	0.8846	0.9020	26
meydan	0.8571	0.9091	0.8824	33
nehir	1.0000	1.0000	1.0000	41
okul	0.9259	0.8333	0.8772	30
orman	1.0000	1.0000	1.0000	25
orta_yerlesim	0.9630	0.8966	0.9286	29
otopark	1.0000	1.0000	1.0000	39
oyun_alani	0.9474	0.9730	0.9600	37
park	0.8378	0.8857	0.8611	35
plaj	1.0000	1.0000	1.0000	40
sanayi_bolgesi	0.9744	0.9744	0.9744	39
seyrek_yerlesim	1.0000	1.0000	1.0000	30
stadyum	1.0000	0.9655	0.9825	29
tarim_arazisi	1.0000	0.9459	0.9722	37
tatil_koyu	0.8929	0.8621	0.8772	29
ticari_alan	0.8684	0.9429	0.9041	35
tren_istasyonu	0.9630	1.0000	0.9811	26
viyaduk	1.0000	1.0000	1.0000	42
yogun_yerlesim	0.9762	1.0000	0.9880	41
Doğruluk			**0.9640**	1000
Makro Ortalama	0.9633	0.9618	0.9622	1000
Ağırlıklı Ortalama	0.9648	0.9640	0.9640	1000

Ek 31. Optimize Edilmiş EfficientNetB2 Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	1.0000	1.0000	1.0000	11
cayir	1.0000	1.0000	1.0000	14
ciplak_arazi	1.0000	0.9333	0.9655	15
col	0.9375	1.0000	0.9677	15
dag	1.0000	1.0000	1.0000	17
depolama_tanklari	1.0000	1.0000	1.0000	18
golet	1.0000	0.9048	0.9500	21
havaalani	0.9000	1.0000	0.9474	18
kilise	0.8000	1.0000	0.8889	12
kopru	1.0000	1.0000	1.0000	18
liman	1.0000	1.0000	1.0000	19
merkez	0.8667	1.0000	0.9286	13
meydan	1.0000	0.8235	0.9032	17
nehir	0.9545	1.0000	0.9767	21
okul	1.0000	0.8000	0.8889	15
orman	1.0000	1.0000	1.0000	13
orta_yerlesim	0.9333	1.0000	0.9655	14
otopark	1.0000	1.0000	1.0000	19
oyun_alani	1.0000	0.9474	0.9730	19
park	0.8889	0.9412	0.9143	17
plaj	1.0000	1.0000	1.0000	20
sanayi_bolgesi	0.9048	1.0000	0.9500	19
seyrek_yerlesim	1.0000	1.0000	1.0000	15
stadyum	0.8750	1.0000	0.9333	14
tarim_arazisi	1.0000	1.0000	1.0000	18
tatil_koyu	0.9286	0.8667	0.8966	15
ticari_alan	0.8095	0.9444	0.8718	18
tren_istasyonu	1.0000	0.5385	0.7000	13
viyaduk	1.0000	1.0000	1.0000	21
yogun_yerlesim	1.0000	0.9524	0.9756	21
Doğruluk			**0.9580**	500
Makro Ortalama	0.9600	0.9551	0.9532	500
Ağırlıklı Ortalama	0.9626	0.9580	0.9567	500

Ek 32. Optimize Edilmiş EfficientNetB3 Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı k**
beyzbol_sahasi	1.0000	1.0000	1.0000	11
cayir	1.0000	1.0000	1.0000	14
ciplak_arazi	1.0000	1.0000	1.0000	15
col	1.0000	1.0000	1.0000	15
dag	1.0000	1.0000	1.0000	17
depolama_tanklari	1.0000	1.0000	1.0000	18
golet	1.0000	0.9524	0.9756	21
havaalani	1.0000	1.0000	1.0000	18
kilise	1.0000	1.0000	1.0000	12
kopru	1.0000	1.0000	1.0000	18
liman	1.0000	1.0000	1.0000	19
merkez	0.8667	1.0000	0.9286	13
meydan	1.0000	0.9412	0.9697	17
nehir	0.9545	1.0000	0.9767	21
okul	0.8571	0.8000	0.8276	15
orman	1.0000	1.0000	1.0000	13
orta_yerlesim	1.0000	0.9286	0.9630	14
otopark	1.0000	1.0000	1.0000	19
oyun_alani	0.9048	1.0000	0.9500	19
park	0.9412	0.9412	0.9412	17
plaj	1.0000	1.0000	1.0000	20
sanayi_bolgesi	0.8571	0.9474	0.9000	19
seyrek_yerlesim	1.0000	1.0000	1.0000	15
stadyum	1.0000	0.8571	0.9231	14
tarim_arazisi	1.0000	1.0000	1.0000	18
tatil_koyu	1.0000	0.9333	0.9655	15
ticari_alan	1.0000	0.8889	0.9412	18
tren_istasyonu	0.9231	0.9231	0.9231	13
viyaduk	1.0000	1.0000	1.0000	21
yogun_yerlesim	0.9130	1.0000	0.9545	21
Doğruluk			**0.9720**	500
Makro Ortalama	0.9739	0.9704	0.9713	500
Ağırlıklı Ortalama	0.9736	0.9720	0.9720	500

Ek 33. Optimize Edilmiş EfficientNetV2B0 Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	1.0000	0.9091	0.9524	11
cayir	0.9286	0.9286	0.9286	14
ciplak_arazi	1.0000	0.9333	0.9655	15
col	0.9375	1.0000	0.9677	15
dag	1.0000	0.8824	0.9375	17
depolama_tanklari	0.9474	1.0000	0.9730	18
golet	1.0000	0.9524	0.9756	21
havaalani	1.0000	0.8889	0.9412	18
kilise	0.8571	1.0000	0.9231	12
kopru	1.0000	0.9444	0.9714	18
liman	1.0000	1.0000	1.0000	19
merkez	0.7692	0.7692	0.7692	13
meydan	0.8824	0.8824	0.8824	17
nehir	1.0000	1.0000	1.0000	21
okul	0.7059	0.8000	0.7500	15
orman	1.0000	0.9231	0.9600	13
orta_yerlesim	0.8571	0.8571	0.8571	14
otopark	0.9500	1.0000	0.9744	19
oyun_alani	0.9444	0.8947	0.9189	19
park	0.8421	0.9412	0.8889	17
plaj	0.9524	1.0000	0.9756	20
sanayi_bolgesi	0.8947	0.8947	0.8947	19
seyrek_yerlesim	1.0000	0.8667	0.9286	15
stadyum	1.0000	0.8571	0.9231	14
tarim_arazisi	0.9474	1.0000	0.9730	18
tatil_koyu	1.0000	0.8000	0.8889	15
ticari_alan	0.8889	0.8889	0.8889	18
tren_istasyonu	0.9286	1.0000	0.9630	13
viyaduk	1.0000	1.0000	1.0000	21
yogun_yerlesim	0.8077	1.0000	0.8936	21
Doğruluk			**0.9320**	500
Makro Ortalama	0.9347	0.9271	0.9289	500
Ağırlıklı Ortalama	0.9369	0.9320	0.9325	500

Ek 34. Optimize Edilmiş EfficientNetV2B1 Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	1.0000	1.0000	1.0000	22
cayir	1.0000	1.0000	1.0000	28
ciplak_arazi	1.000	0.9032	0.9492	31
col	0.9091	1.0000	0.9524	30
dag	1.0000	1.0000	1.0000	34
depolama_tanklari	0.9730	1.0000	0.9863	36
golet	1.0000	0.9762	0.9880	42
havaalani	0.9459	0.9722	0.9589	36
kilise	0.8846	0.9583	0.9200	24
kopru	0.9231	1.0000	0.9600	36
liman	1.0000	0.9737	0.9867	38
merkez	0.9091	0.7692	0.8333	26
meydan	0.9062	0.8788	0.8923	33
nehir	0.9762	1.0000	0.9880	41
okul	0.8929	0.8333	0.8621	30
orman	1.0000	1.0000	1.0000	25
orta_yerlesim	0.9643	0.9310	0.9474	29
otopark	1.0000	0.9744	0.9870	39
oyun_alani	0.9487	1.0000	0.9737	37
park	0.8857	0.8857	0.8857	35
plaj	1.0000	1.0000	1.0000	40
sanayi_bolgesi	0.9048	0.9744	0.9383	39
seyrek_yerlesim	1.0000	1.0000	1.0000	30
stadyum	1.0000	0.9310	0.9643	29
tarim_arazisi	0.9487	1.0000	0.9737	37
tatil_koyu	0.9259	0.8621	0.8929	29
ticari_alan	0.9189	0.9714	0.9444	35
tren_istasyonu	1.0000	0.8846	0.9388	26
viyaduk	0.9767	1.0000	0.9882	42
yogun_yerlesim	1.0000	1.0000	1.0000	41
Doğruluk			**0.9600**	1000
Makro Ortalama	0.9598	0.9560	0.9570	1000
Ağırlıklı Ortalama	0.9607	0.9600	0.9596	1000

Ek 35. Optimize Edilmiş EfficientNetV2B2 Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	1.0000	1.0000	1.0000	11
cayir	1.0000	0.9286	0.9630	14
ciplak_arazi	1.0000	1.0000	1.0000	15
col	1.0000	1.0000	1.0000	15
dag	1.0000	1.0000	1.0000	17
depolama_tanklari	1.0000	1.0000	1.0000	18
golet	1.0000	0.9524	0.9756	21
havaalani	1.0000	1.0000	1.0000	18
kilise	1.0000	1.0000	1.0000	12
kopru	1.0000	1.0000	1.0000	18
liman	1.0000	1.0000	1.0000	19
merkez	1.0000	0.9231	0.9600	13
meydan	0.9444	1.0000	0.9714	17
nehir	0.9545	1.0000	0.9767	21
okul	0.8667	0.8667	0.8667	15
orman	1.0000	1.0000	1.0000	13
orta_yerlesim	1.0000	0.9286	0.9630	14
otopark	1.0000	1.0000	1.000	19
oyun_alani	0.9500	1.0000	0.9744	19
park	0.8824	0.8824	0.8824	17
plaj	1.0000	1.0000	1.0000	20
sanayi_bolgesi	1.0000	0.8947	0.9444	19
seyrek_yerlesim	1.0000	1.0000	1.0000	15
stadyum	0.9286	0.9286	0.9286	14
tarim_arazisi	0.9000	1.0000	0.9474	18
tatil_koyu	0.9286	0.8667	0.8966	15
ticari_alan	0.9444	0.9444	0.9444	18
tren_istasyonu	1.0000	1.0000	1.0000	13
viyaduk	1.0000	1.0000	1.0000	21
yogun_yerlesim	0.9130	1.0000	0.9545	21
Doğruluk			**0.9720**	500
Makro Ortalama	0.9738	0.9705	0.9716	500
Ağırlıklı Ortalama	0.9729	0.9720	0.9719	500

Ek 36. Optimize Edilmiş EfficientNetV2B3 Sınıflandırma Raporu

	Kesinlik	**Duyarlılık**	**F1**	**Örnek Sayısı**
beyzbol_sahasi	1.0000	1.0000	1.0000	11
cayir	0.9333	1.0000	0.9655	14
ciplak_arazi	1.0000	1.0000	1.0000	15
col	1.0000	1.0000	1.0000	15
dag	0.9444	1.0000	0.9714	17
depolama_tanklari	1.0000	1.0000	1.0000	18
golet	1.0000	0.9048	0.9500	21
havaalani	1.0000	1.0000	1.0000	18
kilise	0.9231	1.0000	0.9600	12
kopru	1.0000	1.0000	1.0000	18
liman	0.9500	1.0000	0.9744	19
merkez	1.0000	0.9231	0.9600	13
meydan	0.8947	1.0000	0.9444	17
nehir	0.9524	0.9524	0.9524	21
okul	0.9286	0.8667	0.8966	15
orman	1.0000	0.8462	0.9167	13
orta_yerlesim	1.0000	1.0000	1.0000	14
otopark	1.0000	1.0000	1.0000	19
oyun_alani	1.0000	1.0000	1.0000	19
park	0.8421	0.9412	0.8889	17
plaj	0.9500	0.9500	0.9500	20
sanayi_bolgesi	1.0000	1.0000	1.0000	19
seyrek_yerlesim	1.0000	1.0000	1.0000	15
stadyum	1.0000	1.0000	1.0000	14
tarim_arazisi	1.0000	1.0000	1.0000	18
tatil_koyu	1.0000	0.8667	0.9286	15
ticari_alan	0.8947	0.9444	0.9189	18
tren_istasyonu	1.0000	1.0000	1.0000	13
viyaduk	1.0000	1.0000	1.0000	21
yogun_yerlesim	1.0000	0.9524	0.9756	21
Doğruluk			**0.9720**	500
Makro Ortalama	0.9738	0.9716	0.9718	500
Ağırlıklı Ortalama	0.9736	0.9720	0.9720	500