KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İSTATİSTİK VE BİLGİSAYAR BİLİMLERİ ANABİLİM DALI

SAĞLIK ALANINDA GRADYAN TEMELLİ MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE TRİAJ BELİRLEME

YÜKSEK LİSANS TEZİ

İhsan ARVAS

EKİM 2024
TRABZON

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İSTATİSTİK VE BİLGİSAYAR BİLİMLERİ ANABİLİM DALI

SAĞLIK ALANINDA GRADYAN TEMELLİ MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE TRİAJ BELİRLEME

İhsan ARVAS

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünce
"YÜKSEK LİSANS (İSTATİSTİK VE BİLGİSAYAR BİLİMLERİ)"
Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : 11 / 09 / 2024
Tezin Savunma Tarihi : 15 / 10 / 2024

Tez Danışmanı : Dr. Öğr. Üyesi Tolga BERBER

Trabzon 2024

ÖNSÖZ

“SAĞLIK ALANINDA GRADYAN TEMELLİ MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE TRİAJ BELİRLEME” isimli bu tez Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü İstatistik ve Bilgisayar Bilimleri Anabilim Dalı, Yüksek Lisans Programı’nda hazırlanmıştır.

Tez çalışmam boyunca hiçbir desteğini esirgemeyen ve bilim insanı olma yolundaki ilk adımlarımı atarken kıymetli bilgileriyle ve tecrübesiyle yolumu aydınlatan danışman hocam Sayın Dr. Öğr. Üyesi Tolga BERBER’e teşekkürü bir borç bilirim. Ayrıca, yapıcı eleştirileri ve önerileri ile tezime büyük katkıları bulunan saygıdeğer hocam Sayın Dr. Öğr. Üyesi Uğur ŞEVİK’e, ve tez çalışması süresinde de hiçbir yardımdan kaçınmayan arkadaşlarım Ebru TOYGAR’a, İsmail Hakkı ŞENGÜL’e, İbrahim Ethem BALABAN’a, Ertuğrul Eşol’a, Arş. Gör. Aleyna YILMAZ’a ve Arş. Gör. Beyzanur SİYAH’a teşekkür ederim.

Tez çalışması sürecinde bana desteklerini esirgemeyen Türkiye Hastanesi yönetimine, Genel Müdür yardımcı Mustafa DOĞRU’ya İstatistik ve Bilgisayar Bölümü’ndeki tüm değerli hocalarıma ve asistan arkadaşlarıma teşekkürlerimi sunarım.

Son olarak, doğduğum günden beri elimi hiç bırakmayan, hayatım boyunca aldığım tüm kararlarda bilgi ve tecrübelerinden faydalandığım ve her kararımda arkamda olan, maddi ve manevi desteklerini hiçbir zaman esirgemeyen ve bugünlere gelmemde en büyük paya sahip olan annem Hamide ARVAS, babam Ahmet Adnan ARVAS, abim Mehmet Rıdvan ARVAS’a teşekkür eder, minnettarlığımı sunarım.

Bu tez çalışmasının bundan sonraki çalışmalara katkı sağlamasını temenni ederim.

İhsan ARVAS

Trabzon 2024

.

TEZ ETİK BEYANNAMESİ

Yüksek Lisans Tezi olarak sunduğum “SAĞLIK ALANINDA GRADYAN TEMELLİ MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE TRİAJ BELİRLEME” başlıklı bu çalışmayı baştan sona kadar danışmanım Dr. Öğr. Üyesi Tolga BERBER’in sorumluluğunda tamamladığımı, verileri/örnekleri Özel Türkiye Hastanesi'nin onayıyla KVKK ya uygun topladığımı, deneyleri/analizleri ilgili laboratuvarlarda yaptığımı/yaptırdığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 15 /10/2024

İhsan ARVAS

İÇİNDEKİLER

Sayfa No

ÖNSÖZ III
TEZ ETİK BEYANNAMESİ IV
İÇİNDEKİLER V
ÖZET VIII
SUMMARY VIII
ŞEKİLLER DİZİNİ IX
TABLOLAR DİZİNİ X
SEMBOLLER VE KISALTMALAR DİZİNİ XI
1. GENEL BİLGİLER 1
1.1. Literatür Taraması 5
1.1.1. Uygulama Faydaları 7
1.1.2. Uygulama Zorlukları ve Başarı Kriterleri 8
1.1.3. Sağlık Hizmetlerinde Etkisi 10
1.2. Makine Öğrenmesi 11
1.2.1. Makine Öğrenimi Modellerinin Türleri 14
1.2.2. Naive Bayes Sınıflandırma Yöntemi 21
1.2.3. Destek Vektör Makineleri 21
1.2.4. Yapay Sinir Ağları 23
1.2.5. Ağaç Yöntemleri 25
1.2.6. Model Değerlendirmesi 32
1.3. Karar Destek Sistemleri 35
2. YAPILAN ÇALIŞMALAR 36
2.1. Araştırma Tasarımı 36
2.2. Veri İşleme ve Hazırlık Süreçleri 37
2.3. Modelleme Yaklaşımı 40
2.4. Modelin Değerlendirilmesi 40
2.5. Veri Analizi 40
2.6. Kullanılan Yazılım ve Araçlar 44
3. BULGULAR VE TARTIŞMA 45
4. SONUÇ VE ÖNERİLER 47

5. KAYNAKLAR 48
6. EKLER 52
ÖZGEÇMİŞ

Yüksek Lisans Tezi

ÖZET

SAĞLIK ALANINDA GRADYAN TEMELLİ MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE TRİAJ BELİRLEME

İhsan ARVAS

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
İstatistik ve Bilgisayar Bilimleri Anabilim Dalı
Danışman: Dr. Öğr. Üyesi Tolga BERBER
2024, 51 Sayfa, 12 sayfa Ek

Bu tez çalışmasında, hastaların şikayetlerine göre ilgili uzmanlık alanına yönlendirilmesi amacıyla makine öğrenimi algoritmalarının karar destek sistemlerine entegrasyonunu incelemektedir. Özel Türkiye Hastanesi'nden elde edilen yaklaşık bir milyon anonim hasta kaydını kullanarak hastaları hastane girişinde doğru bölüme yönlendirecek bir model geliştirilmiştir. Naive Bayes, Destek Vektör Makineleri, Yapay Sinir Ağları, Rastgele Orman, AdaBoost, Gradient Boosting, CatBoost, LightGBM ve XGBoost gibi çeşitli algoritmalar karşılaştırılmış ve %97 doğruluk oranıyla en yüksek performansı XGBoost algoritması göstermiştir. Hasta şikayetleri gibi metin verileri TF-IDF yöntemleriyle sayısal forma dönüştürülmüştür. Modelin başarısı doğruluk metrikleri ile değerlendirilmiş ve sınıf bazında performans, karışıklık matrisi kullanılarak analiz edilmiştir. Bulgular, triaj süreçlerinin otomatikleştirilmesi ve sağlık profesyonellerine destek sağlanması açısından önemli bir potansiyeli ortaya koymakta ve yapay zeka ile makine öğreniminin sağlık hizmetlerinde etkili birer karar destek aracı olarak rolünü vurgulamaktadır.

Anahtar Kelimeler: Yapay zekâ, Makine öğrenimi, Sağlık hizmetleri, Karar destek sistemleri, XGBoost, Veri analizi, Sınıflandırma modelleri, Sağlık veri seti

Master Thesis

SUMMARY

TRIAGE DETERMINATION WITH GRADIENT-BASED MACHINE LEARNING METHODS IN HEALTH FIELD

İhsan ARVAS

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Statistics and Computer Sciences Graduate Program
Supervisor: Asst. Prof. Dr. Tolga BERBER
2024, 51 Pages, 12 Pages Appendix

In this thesis, examines the integration of machine learning algorithms into decision support systems for directing patients to the appropriate specialty based on their complaints. A model was developed to accurately guide patients to the correct department at the hospital entrance, using approximately one million anonymized records from Özel Türkiye Hospital. Various algorithms, including Naive Bayes, Support Vector Machines, Neural Networks, Random Forest, AdaBoost, Gradient Boosting, CatBoost, LightGBM, and XGBoost, were compared, with XGBoost demonstrating the highest performance at 97% accuracy. Text data, such as patient complaints, were converted into numerical form using TF-IDF methods. The model's success was assessed through accuracy metrics and class-level performance analyzed with a confusion matrix. Findings highlight the potential for automating triage processes and supporting healthcare professionals, showcasing the role of AI and machine learning as effective decision support tools in healthcare.

Key Words: Artificial intelligence, Machine learning, Healthcare services, Decision support systems, XGBoost, Data analysis, Classification models, Health dataset

ŞEKİLLER DİZİNİ

Sayfa No

Şekil 1. Karar sisteminin gösterimi 2
Şekil 2. Makine öğrenimi yöntemleri 14
Şekil 3. Denetimli öğrenme 15
Şekil 4. Denetimsiz öğrenme 18
Şekil 5. Yaş dağılımı 42
Şekil 6. Cinsiyet dağılımı pasta grafiği 43
Şekil 7. Bölümlere göre hasta dağılımı 44
Şekil 8. Uygulama ekranı 47

TABLOLAR DİZİNİ

Sayfa No

Tablo 1. Karışıklık matrisi 32
Tablo 2. Veriler 39
Tablo 3. Yaş değerleri 42
Tablo 4. Cinsiyet yüzdeleri 43
Tablo 5. Bölüme göre hasta dağılımı (ilk 5) 44
Tablo 6. Makine öğrenimi algoritmaları değerlendirilmesi 46

SEMBOLLER VE KISALTMALAR DİZİNİ

ADA	: Amerikan Diyabet Derneği
AI	: Yapay Zekâ
EHR	: Elektronik Sağlık Kayıtları
ED	: Acil Servis
ICD	: Uluslararası Hastalık Sınıflandırması
ICU	: Yoğun Bakım Ünitesi
KDS	: Karar Destek Sistemi
KNN	: K-En Yakın Komşu
ML	: Makine Öğrenimi
NLP	Doğal Dil İşleme
PCA	: Temel Bileşenler Analizi
SVM	: Destek Vektör Makineleri
SMOTE	: Sentetik Azınlık Aşırı Örneklem Tekniği
TF-IDF	: Terim Frekansı - Ters Belge Frekansı
t-SEN	: t-Dağıtılmış Stokastik Komşu Gömme
YSA	: Yapay Sinir Ağları

1. GENEL BİLGİLER

Karar verme, kararı verecek bireylerin tüm mevcut ihtimaller arasından amaç ve hedeflere uygun olarak en ideal seçeneğe yönelmesi olarak tanımlanmaktadır (Özata & Aslan 2004). Bu kararların verilebilmesi, bireylerin yetenekleri gibi unsurlara bağlıdır. Bireyler tüm koşulları sağlıyor olabilseler bile çevresel etmenler, ruh hali gibi sebeplerden dolayı doğru kararlar alınamaması veya doğru alınmış kararların yanlış zamanlarda uygulanması karar verme sürecini etkileyebilir. Bu süreçleri daha etkin ve koordineli bir şekilde yürütebilmek için karar destek sistemlerine (KDS) ihtiyaç duyulmaktadır. Karar destek sistemleri karar verecek kişilere yardımcı olmak amacıyla teknoloji, algoritma, belge ve verileri kullanarak, karar vericiye yardımcı problemi daha iyi tanımlayan ve çözümleyici yaklaşımlarda bulunan bilgisayar sistemidir (Hersh, 1999).

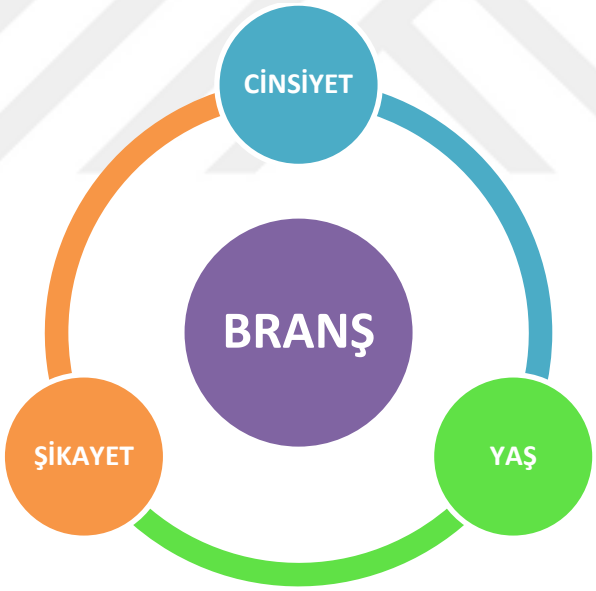
Sağlık sektöründe tanı ve teşhisin sadece insan gücüyle yönetilmesi mümkün değildir (ADA, American Diabetes Association 2014). Triaj, sınırlı kaynakların bulunduğu ortamlarda, hastaların durumlarına göre önceliklendirilmesi ve en uygun tedavi yolunun belirlenmesi sürecidir. Doğru bir triaj sistemi, hastaların hızlı bir şekilde değerlendirilmesine olanak tanır. Kritik durumda olan hastaların en kısa sürede tedavi edilmesini sağlar ve hastane kaynaklarının etkin bir şekilde kullanılmasına katkıda bulunur. Yanlış veya gecikmiş kararlar, hasta sonuçlarını olumsuz etkileyebilir ve sağlık sisteminin genel işleyişini bozabilir.

Bu kritik süreçte karar destek sistemlerinin önemi giderek artmaktadır. Karar destek sistemleri, klinisyenlere hastaların durumlarını değerlendirme ve uygun tedavi seçeneklerini belirleme konusunda yardımcı olan araçlar sunar. Bu sistemler, hasta verilerini analiz ederek ve belirli algoritmalara dayanarak önerilerde bulunur böylece klinisyenlerin daha bilinçli ve hızlı kararlar almasını sağlar. Ancak, bu sistemlerin etkinliği, kullanılan verilerin kalitesine ve sistemlerin doğru bir şekilde yapılandırılmasına bağlıdır.

Makine öğrenimi, son yıllarda tıp alanında karar destek sistemlerinin geliştirilmesinde önemli bir rol oynamaktadır. Makine öğrenimi algoritmaları, büyük miktarda hasta verisini analiz ederek ve bu verilerden kalıplar çıkararak, triaj sürecinde klinisyenlere daha doğru ve objektif önerilerde bulunabilir. Özellikle derin öğrenme teknikleri, karmaşık tıbbi verileri anlamlandırmada ve daha önce fark edilmemiş ilişkileri ortaya çıkarmada güçlü bir araç haline gelmiştir. Bu bağlamda, makine öğrenimi

teknolojilerinin tıpta triaj süreçlerine entegrasyonu, karar destek sistemlerinin performansını ve doğruluğunu artırma potansiyeline sahiptir.

Yapay zekâ destekli bu sistemler, hasta bakımında standartlaşmayı sağlayabilir, insan hatalarını minimize edebilir ve en nihayetinde hasta sonuçlarını iyileştirebilir. Mevcut literatür, bu alandaki gelişmelerin hem teorik hem de pratik açıdan büyük bir önem taşıdığını göstermektedir ve bu tez, sağlık hizmetlerinde triaj sürecini otomatikleştirmek ve hızlandırmak için karar destek sisteminin kullanımını incelemektedir. Özellikle, karar destek sistemi hastaların cinsiyetini, yaşını ve şikayetlerini alarak hangi branşa yönlendirileceklerini belirleyen bir triaj sistemi olarak tasarlanmıştır. Bu sistem, hastaların sağlık hizmetlerine daha hızlı ve etkin bir şekilde erişimini ve sağlık profesyonellerinin zamanını daha verimli bir şekilde kullanmalarını sağlamaktadır. Bu süreç, hem hastaların hızlı ve etkili bir şekilde tedavi edilmesini sağlar hem de sağlık profesyonellerinin kaynaklarını en verimli şekilde kullanmalarına olanak tanır.



Şekil 1. Karar destek sisteminin gösterimi

Branş seçim süreci genellikle bir Pratisyen hekim tarafından gerçekleştirilir ve birkaç aşamadan oluşur. İlk olarak, hasta, klinik durumuna ve belirtilerine göre değerlendirilir. Bu değerlendirme genellikle hastanın şikayetleri, cinsiyeti ve yaşına göre hangi branşa gideceğini belirler.

Sağlık sistemlerinde hasta yoğunluğunun artması, takip ve tedavi süreçlerinin hızlı ve etkili bir şekilde yönetilmesini gerektirmektedir. Triaj, hastaların durumlarının aciliyetine göre sınıflandırılması ve önceliklendirilmesi sürecidir. Geleneksel triaj süreçleri genellikle sağlık profesyonelleri tarafından yürütülmekte ve bu da zaman alıcı olabilmektedir. Bu noktada, yapay zekâ (AI) ve karar destek sistemleri, sağlık hizmetlerinde triaj süreçlerini iyileştirmek için umut verici çözümler sunmaktadır.

Karar destek sistemleri (KDS), sağlık hizmetlerinde klinik karar alma süreçlerini iyileştirmeyi amaçlayan bilgisayar tabanlı araçlardır. Bu sistemler, klinik verilere dayalı olarak sağlık profesyonellerine tanı, tedavi ve hasta yönetimi konusunda yardımcı olur. Karar destek sistemlerinin amacı, klinik kararların doğruluğunu artırmak, tedavi süreçlerini standardize etmek ve sağlık hizmetlerinde hasta güvenliğini sağlamaktır. Tanı ve tedavi süreçlerinde karar destek sistemleri, klinisyenlere hastaların mevcut semptomları, tıbbi geçmişleri ve laboratuvar sonuçları gibi verileri analiz ederek olası tanılar ve tedavi seçenekleri sunar. Bates vd. (2003) çalışması, karar destek sistemlerinin doğru tanı konulmasında klinisyenlere yardımcı olarak tıbbi hataları önemli ölçüde azalttığını ve hasta sonuçlarını iyileştirdiğini göstermektedir. Benzer şekilde, Garg vd., (2005), tarafından yapılan bir meta-analiz, karar destek sistemlerinin klinik kararların kalitesini artırdığını ve hasta bakım süreçlerini iyileştirdiğini ortaya koymaktadır.

Karar destek sistemleri, sağlık hizmetlerinin çeşitli alanlarında geniş çapta benimsenmektedir. Hastaların ilk değerlendirmelerini yaparak doğru branşa yönlendirilmesi, klinik kararların desteklenmesi ve hastaların sağlık durumlarının izlenmesi gibi birçok alanda kullanılmaktadır. Bu teknolojilerin gelecekte daha da yaygınlaşması ve gelişmesi beklenmektedir. Karar destek sistemlerinin gelişimi, sağlık hizmetlerinin genel verimliliğini artırmak ve hasta sonuçlarını iyileştirmek için kritik bir adım olarak görülmektedir.

Makine öğrenimi algoritmaları, büyük miktarda klinik veriyi analiz ederek, Karar destek sistemlerinin hastalar hakkında daha doğru ve öngörücü kararlar almasını sağlar. Bu entegrasyon, hastaların semptomlarının geçmiş verilere ve mevcut klinik bilgilere dayanarak değerlendirilmesini mümkün kılar, bu da daha kişiselleştirilmiş ve hassas tıbbi önerilerin

sunulmasına olanak tanır. Örneğin, makine öğrenimi modelleri, benzer semptomlar gösteren geçmiş hastaların sonuçlarını inceleyerek, mevcut hastanın tedavi sürecini optimize edebilir ve olası komplikasyonları öngörebilir. Makine öğrenimi destekli karar destek sistemleri, klinik karar süreçlerini sürekli olarak öğrenip geliştirebilir. Bu sistemler, sağlık hizmetleri alanında karşılaşılan karmaşık ve dinamik problemlere uyum sağlayarak, değişen hasta ihtiyaçlarına ve yeni ortaya çıkan tıbbi bilgilere hızla yanıt verebilir. Ayrıca, bu tür karar destek sistemleri, sağlık hizmetlerinde veri tabanlı karar alma süreçlerini otomatikleştirerek, klinisyenlerin rutin görevlerden ziyade daha karmaşık ve kritik kararlarla ilgilenmesine olanak tanır. Karar destek sistemlerinin makine öğrenimi ile entegrasyonu, sağlık hizmetlerinin verimliliğini artırmak ve hasta bakımını daha güvenli ve etkili hale getirmek için önemli bir adım olarak öne çıkmaktadır.

Yapay zekâ, dijital triaj süreçlerinde giderek daha fazla kullanılmaktadır. Yapay zekâ tabanlı sistemler, büyük veri kümelerini analiz ederek, hastaların semptomlarını daha doğru bir şekilde değerlendirme ve yönlendirme yeteneğine sahiptir. Berry vd. (2018), tarafından yapılan bir çalışmada, yapay zekâ tabanlı dijital triaj sistemlerinin klinik kararların doğruluğunu artırarak, hasta bakım süreçlerinde önemli iyileşmeler sağladığını göstermektedir. Yapay zekânın doğal dil işleme (NLP) yetenekleri, hastaların şikayetlerini anlamlandırarak doğru tıbbi müdahalelerin önerilmesini sağlar. Yapay zekâ ve karar destek sistemlerinin birleşimi, dijital triaj süreçlerinde devrim niteliğinde bir gelişme sunmaktadır. Bu iki teknolojinin bir araya gelmesi, hasta değerlendirme ve yönlendirme süreçlerinde yüksek doğruluk ve hız sağlar. Yapay zekâ, hastalardan toplanan büyük veri kümelerini analiz ederken, karar destek sistemleri bu analizleri kullanarak en uygun tedavi yollarını önerir. Bu entegrasyon, sağlık profesyonellerine güvenilir bir destek sunar ve hasta bakımında kaliteyi artırır (Moreno vd., 2020).

Bu tez çalışmasının temel amacı, sağlık hizmetlerinde triaj süreçlerini otomatikleştirmek ve daha hızlı, doğru bir şekilde hasta yönlendirmesi yapabilen bir karar destek sistemi geliştirmektir. Özellikle sağlık alanında, hastaların uygun branşlara doğru ve zamanında yönlendirilmesi, tedavi süreçlerinin etkinliğini artırmak için kritik bir öneme sahiptir. Bu bağlamda, makine öğrenimi algoritmaları kullanılarak geliştirilen model, hasta şikayetleri, yaş ve cinsiyet gibi verileri analiz ederek hastaların hangi bölüme başvurması gerektiğini belirlemeyi hedeflemektedir. Çalışmada, farklı makine öğrenimi algoritmaları değerlendirilmiş ve en yüksek doğruluk oranı sağlayan yöntem belirlenerek, triaj süreçlerinin iyileştirilmesi amaçlanmıştır.

Gelecekte, dijital triaj sistemlerinin, yapay zekâ ve makine öğrenimi ile desteklenmiş karar destek sistemleri daha da yaygınlaşması beklenmektedir. Literatürde, bu sistemlerin sağlık hizmetlerinde standart hale gelmesinin, hasta bakımında daha yüksek kalite ve doğruluk sunacağı, sağlık profesyonellerinin iş yükünü azaltarak, daha kritik kararlarla ilgilenmelerine olanak tanıyacağı vurgulanmaktadır (Sutton vd., 2019), Dijital triaj süreçleri, otomasyon, yapay zekâ, karar destek sistemleri ve makine öğrenimi gibi ileri teknolojilerle birleştiğinde, sağlık hizmetlerinde devrim niteliğinde yenilikler sunmaktadır.

1.1. Literatür Taraması

Sağlık hizmetlerinde dijital triaj süreçleri, otomasyon, yapay zekâ, karar destek sistemleri (KDS) ve makine öğrenimi gibi teknolojilerin entegrasyonu, sağlık hizmetlerinin verimliliğini artırmak ve hasta bakımını iyileştirmek için önemli bir potansiyel sunmaktadır. Literatürde bu teknolojilerin sağlık hizmetlerine entegrasyonu, hasta değerlendirme süreçlerini optimize etmek, karar alma süreçlerini desteklemek ve genel sağlık hizmetlerinin kalitesini artırmak açısından büyük bir ilgi görmektedir.

Dijital triaj süreçleri, hastaların semptomlarının hızlı ve etkili bir şekilde değerlendirilmesi için dijital platformlar aracılığıyla gerçekleştirilen ön tarama süreçlerini ifade eder. Bu süreçler, özellikle acil servisler ve birinci basamak sağlık hizmetlerinde önemlidir. Literatürde, dijital triaj sistemlerinin sağlık hizmetlerinde verimliliği artırdığı ve hasta memnuniyetini iyileştirdiği belirtilmektedir (Moreno vd., 2020).

Otomasyonun dijital triaj süreçlerine entegrasyonu, bu sistemlerin hızını ve doğruluğunu artıran önemli bir faktördür. Otomatik sistemler, hastaların belirtilerini hızlı bir şekilde analiz ederek standart bir değerlendirme sunar ve insan hatasını minimize eder. Literatürde, triaj süreçlerinin otomasyonu sayesinde acil durumlarda daha hızlı ve etkili müdahalelerin mümkün olduğu vurgulanmaktadır (Sutton vd., 2019).

Karar destek sistemleri (KDS), dijital triaj süreçlerinin önemli bir bileşeni olarak öne çıkmaktadır. Karar destek sistemleri, hastalardan toplanan bilgileri analiz ederek, triaj kararlarını destekler ve sağlık profesyonellerine en uygun müdahale yöntemlerini önerir. Karar destek sistemlerinin dijital triajla entegrasyonu, hasta değerlendirmesinde yüksek doğruluk ve hız sağlayarak, sağlık hizmetlerinin genel verimliliğini artırır (Mehta vd., 2021). Bu sistemler, klinik karar süreçlerini optimize ederek, hasta bakımında daha güvenilir ve etkili bir yol sunar.

Makine öğrenimi, dijital triaj ve karar destek sistemlerinin gücünü artıran bir diğer önemli teknolojidir. Literatürde, makine öğrenimi algoritmalarının büyük veri kümeleri üzerinde eğitim alarak hasta semptomlarını daha doğru bir şekilde değerlendirdiği belirtilmektedir (Li vd., 2020). Bu algoritmalar, geçmiş hasta verileri ve klinik yönergelerle karşılaştırmalar yaparak, daha öngörücü ve kişiselleştirilmiş tıbbi öneriler sunar. Makine öğrenimi ile desteklenen karar destek sistemleri, sağlık profesyonellerine daha fazla bilgi ve analiz imkânı sunarak, daha bilinçli kararlar almalarına yardımcı olur.

Etkili dijital triaj araçlarının geliştirilmesi, hastaların semptomlarını doğru ve kolay bir şekilde girebilmesini sağlamak için büyük önem taşır. Bu araçların tasarımında, teşhis doğruluğunun kritik durumları kaçırma riskiyle dengelenmesi gerekmektedir. Kullanıcı geri bildirimlerine dayalı özelleştirme ve yinelemeli geliştirme süreçleri, erişilebilir ve kullanışlı araçlar oluşturmak için kritik öneme sahiptir. Bu araçların gerçek yaşam senaryolarında test edilerek değerlendirilmesi, etkinlik ve güvenliklerinin sağlanması açısından gereklidir. (Gottliebsen vd., 2020; FitzGerald vd., 2009)

Dijital triaj sistemlerinin etkinliğini ölçmek için sağlam denetleme mekanizmaları gereklidir. Bu değerlendirme, sistemlerin teşhis doğruluğunu ve yüksek ile düşük aciliyet vakalarını doğru bir şekilde tanımlama yeteneklerini içermelidir. Yaygın olarak kullanılan referans standartlar arasında acil servis (ED) ve yoğun bakım ünitesi (ICU) kabul oranları yer almakta olup, bu ölçütler hasta aciliyetini tahmin etmek için kullanılmaktadır (BMJ Open).

Karar destek sistemleri, klinik yönergelerin daha tutarlı bir şekilde uygulanmasında da önemli bir rol oynamaktadır. Sağlık hizmetlerinde kanıta dayalı uygulamaların yaygınlaştırılması, tedavi süreçlerinin standardizasyonu açısından büyük bir öneme sahiptir. Kawamoto vd. (2005) çalışmasında, karar destek sistemlerinin klinik yönergelerin uygulanmasını destekleyerek, hastaların daha tutarlı ve etkili bir şekilde tedavi edilmesini sağladığı vurgulanmaktadır. Bu durum, özellikle kompleks ve nadir hastalıkların tedavisinde klinisyenlere önemli bir destek sunmaktadır.

Karar destek sistemleri, yalnızca mevcut klinik verileri analiz etmekle kalmaz, aynı zamanda yapay zekâ ve makine öğrenimi teknikleri kullanarak gelecekteki sağlık sonuçlarını tahmin etme yeteneğine de sahiptir. Topol (2019) çalışmasında, yapay zekâ destekli karar destek sistemlerinin, hastaların sağlık durumlarını ve tedavi yanıtlarını tahmin ederek, klinisyenlere proaktif tedavi planları sunabileceği belirtmektedir. Bu tür tahmin yetenekleri,

özellikle kronik hastalıkların yönetiminde ve yeniden hastaneye yatışların önlenmesinde değerli bir araç olarak öne çıkmaktadır.

Son yıllarda, yapay zekâ destekli karar destek sistemlerinin sağlık hizmetlerinde kullanımı giderek artmıştır. Bu teknolojiler, hasta tiraj ve branş belirleme süreçlerinde önemli roller üstlenmektedir. Karar destek sistemleri, hastaların semptomlarını değerlendirerek ve klinik yönergelerle karşılaştırarak, uygun branşa yönlendirme sürecinde etkili bir araç olarak kullanılmaktadır. Yapay zekâ tabanlı bu sistemler, doğal dil işleme yetenekleri sayesinde hastaların şikayetlerini anlamlandırabilir, gerekli tıbbi müdahaleleri önerir ve bu bilgiler doğrultusunda triaj kararlarını destekler.

1.1.1. Uygulama Faydaları

Makine öğrenimi ve yapay zekâ tabanlı karar destek sistemleri, hastaların semptomlarını ve tıbbi geçmişlerini analiz ederek daha kişiselleştirilmiş tedavi planları sunar ve her hastanın özel ihtiyaçlarına uygun tedavi yöntemlerinin belirlenmesini sağlar. Li vd. (2020) tarafından yapılan bir araştırmada, yapay zekâ tabanlı karar destek sistemlerinin, hastalara daha kişisel ve etkili bakım sunarak sağlık hizmetlerinin kalitesini artırdığı gösterilmiştir. Bu sistemler, hastaların sağlık durumlarına daha uygun müdahalelerin yapılmasını sağlayarak tedavi sonuçlarını iyileştirir.

Dijital triaj sistemleri ve karar destek sistemleri, hastaların hızlı ve doğru bir şekilde değerlendirilmesini sağlayarak, hasta memnuniyetini artırır. Sutton vd. (2019), bu sistemlerin hastaların bekleme sürelerini azalttığını ve doğru tedaviye yönlendirildiklerini belirterek, hasta memnuniyetinin önemli ölçüde iyileştirildiğini ifade etmektedir. Ayrıca, hastaların sağlık sorunlarının hızlı bir şekilde değerlendirilmesi ve tedaviye yönlendirilmesi, hasta deneyimini olumlu yönde etkiler.

Makine öğrenimi algoritmaları, sürekli olarak yeni verilerle beslenerek öğrenmeye devam eder ve zamanla daha doğru sonuçlar üretir. Bu, dijital tiraj ve karar destek sistemlerinin sürekli olarak kendilerini geliştirmesini sağlar. Li vd. (2020), makine öğrenimi tabanlı sistemlerin, her geçen gün daha fazla veri ile daha hassas ve doğru tahminlerde bulunarak, sağlık hizmetlerinin kalitesini artırdığını vurgulamaktadır. Bu sistemler, sağlık hizmetlerinin dinamik doğasına uyum sağlayarak, uzun vadede sürdürülebilir bir çözüm sunar.

Dijital tiraj sistemlerinin uygulanması, karışık sonuçlar doğurabilir. Örneğin, İngiltere'deki eConsult sistemi, personel iş yükünü azaltmada sınırlı etki göstermiş, ancak hastalar tarafından aile hekimleriyle alternatif bir iletişim yöntemi sağladığı için takdir edilmiştir. Hollanda'daki web tabanlı triaj ise, hastalar arasında daha iyi bireysel tedavi yönetimi teşvik ederek birinci basamak sağlık hizmeti kullanımında önemli bir azalma sağlamıştır.

1.1.2. Uygulama Zorlukları ve Başarı Kriterleri

Dijital triaj sistemleri ve karar destek sistemlerinin uygulanmasındaki en büyük zorluklardan bazıları, gerekli teknolojik altyapının sağlanmaması ve mevcut sistemlerle entegre edilememesidir. Moreno vd. (2020) çalışmasında, sağlık kuruluşlarının eski sistemleri ve altyapıları nedeniyle yeni dijital çözümleri entegre etmede ciddi zorluklar yaşadıkları belirtmektedir. Özellikle farklı sağlık bilgi sistemlerinin uyumsuzluğu, veri akışını ve sistemler arası iletişimi engelleyebilir. Bu durum, dijital triaj ve karar destek sistemleri uygulamalarının tam potansiyelini kullanmayı zorlaştırır.

Dijital triaj ve karar destek sistemlerinin uygulanmasında karşılaşılan diğer önemli zorluklar, veri güvenliği ve hasta mahremiyetidir. Mehta vd. (2021) tarafından yapılan araştırma, büyük veri kümelerinin yönetimi ve korunmasında yaşanan güvenlik zorluklarının, hastaların kişisel bilgilerinin korunmasını zorlaştırdığını vurgulamaktadır. Sağlık hizmetleri sağlayıcıları, veri sızıntıları ve ihlaller karşısında yasal yükümlülüklerle karşı karşıya kalabilirler ve bu da dijital sistemlerin güvenliğini sağlamak için ek önlemler alınmasını gerektirir.

Dijital triaj ve karar destek sistemleri sistemlerinin uygulanması genellikle yüksek maliyetli olabilir. Li vd. (2020), yapay zekâ ve makine öğrenimi gibi ileri teknolojilerin geliştirilmesi ve sürdürülmesinin büyük mali yatırımlar gerektirdiğini belirtmektedir. Ayrıca, bu sistemlerin sürekli güncellenmesi ve bakımı da ek maliyetler getirir. Sağlık kuruluşları, bu teknolojilere yatırım yaparken, kısıtlı bütçelerini etkili bir şekilde yönetmek zorundadır.

Dijital triaj ve karar destek sistemleri sistemlerinin başarıyla uygulanabilmesi için sağlık profesyonelleri ve hastalar tarafından kabul görmesi gerekmektedir. Sutton vd. (2019), sağlık çalışanlarının yeni teknolojilere karşı direnç gösterebileceğini ve bu sistemlerin karmaşıklığı nedeniyle eğitim ve uyum süreçlerinin zorlayıcı olabileceğini

belirtmiştir. Hastaların bu sistemlere olan güveni de kritik önem taşır; teknolojiye olan şüphe, sistemlerin etkin kullanımını sınırlayabilir.

Makine öğrenimi ve yapay zekâ algoritmaları, her zaman tutarlı performans göstermeyebilir. Berry vd. (2018), dijital triaj sistemlerinin bazı durumlarda yanlış değerlendirmeler yapabileceğini ve bu hataların ciddi sağlık sonuçlarına yol açabileceğini vurgulamıştır. Sistemlerin yanlış pozitif veya yanlış negatif sonuçlar üretmesi, hasta bakımı üzerinde olumsuz etkilere neden olabilir. Bu, özellikle karmaşık tıbbi durumlar ve nadir hastalıklar söz konusu olduğunda daha büyük bir sorun oluşturur.

Dijital triaj ve karar destek sistemleri uygulamaları, yasal ve etik konuları da beraberinde getirir. Hastaların verilerinin kullanımı, paylaşılması ve korunması, çeşitli yasal düzenlemelere tabidir. Sağlık hizmeti sağlayıcıları, bu teknolojileri kullanırken, hasta haklarına ve mahremiyet standartlarına uygun davranmak zorundadır. Mehta vd. (2021), bu yasal ve etik zorlukların, dijital triaj ve karar destek sistemleri sistemlerinin geniş çapta benimsenmesini zorlaştırabileceğini ifade etmiştir.

Dijital triaj ve karar destek sistemlerinin etkinliği, farklı kültürel ve demografik faktörlere göre değişebilir. Sutton vd. (2019), bu sistemlerin farklı popülasyon grupları arasında farklı sonuçlar üretebileceğini ve bu durumun, belirli gruplarda daha az etkili olabileceğini belirtmiştir. Ayrıca, teknolojiye erişimdeki eşitsizlikler, bu sistemlerin bazı hastalar tarafından kullanımını sınırlayabilir. Bu faktörler, dijital triaj ve karar destek sistemlerinin kapsayıcılığını ve adaletli dağılımını etkileyebilir.

Dijital triaj sistemlerinin başarılı bir şekilde entegrasyonu, dikkatli planlama ve destek gerektirir. Süper kullanıcıların varlığı, pazarlama beklentileriyle uygulama uyumu ve mevcut sistemlerle entegrasyon etkili uygulama için kritik faktörlerdir. Bunlar karar destek sistemleri ve yapay zekâ uygulamalarının sağlık hizmetlerine entegrasyonunun karmaşık ve zorlu bir süreç olduğunu göstermektedir. Ancak, bu zorlukların üstesinden gelindiğinde, bu teknolojiler sağlık hizmetlerinin pratik kullanımını dönüştürme potansiyeline sahiptir. Literatürde, bu zorlukların farkında olmak ve onları yönetme stratejileri geliştirmek, bu sistemlerin başarılı bir şekilde uygulanmasının anahtarı olarak vurgulanmaktadır. Ayrıca, dijital araçlar kullanıcı dostu olmalı ve hastaları doğru bir şekilde tedavi yolculukları boyunca yönlendirebilmelidir.

1.1.3. Sağlık Hizmetlerinde Etkisi

Yapay zekâ (AI), karar destek sistemleri (KDS) ve makine öğrenimi (ML), sağlık hizmetlerinin pek çok yönünü dönüştürmekte ve geliştirmekte büyük rol oynamaktadır. Bu teknolojiler, veri analizi, teşhis ve tedavi süreçlerinde etkili araçlar olmak üzere kullanılarak, sağlık hizmetlerinin verimliliğini, erişilebilirliğini ve kalitesini artırmaktadır.

Yapay zekâ ve makine öğrenimi, büyük sağlık veri kümelerini analiz ederek değerli bilgiler çıkarmada kullanılmaktadır. Bu analizler, sağlık trendlerini anlama, risk faktörlerini belirleme ve hasta tahlil sonuçlarını iyileştirme konularında kritik öneme sahiptir. Moreno vd. (2020), bu sistemlerin sağlık hizmetlerinde verimliliği nasıl artırdığını ve hasta bakımını nasıl iyileştirdiğini vurgulamıştır. Hastalıkların seyrini tahmin edebilir ve potansiyel salgınları önceden belirleyebilir. Bu yetenek, özellikle pandemi gibi küresel sağlık krizlerinde, sağlık kaynaklarının önceden planlanması ve etkin bir şekilde yönlendirilmesi için hayati önem taşır. Li vd. (2020), Yapay zekânın hastalık tahmin modellerinin doğruluğunu ve bu modellerin sağlık politikalarına nasıl rehberlik ettiğini incelemiştir. Teşhis koyma süreçlerinde hız ve doğruluk sağlar. Yapay zekâ tabanlı teşhis sistemleri, görüntüleme teknikleri ve biyomarker analizlerinde kullanılarak, hastalıkların erken ve doğru bir şekilde teşhis edilmesine olanak tanır. Bu teknolojiler, aynı zamanda kişiselleştirilmiş tedavi planlarının oluşturulmasında da kullanılmaktadır.

Karar destek sistemleri, sağlık profesyonellerine kritik karar verme aşamalarında yardımcı olur. Yapay zekâ destekli karar destek sistemleri, hastanın tıbbi geçmişi, mevcut sağlık durumu ve benzer vakaları analiz ederek en uygun tedavi önerilerini sunar. Berry vd. (2018), karar destek sistemlerinin karar verme sürecinde sağladığı doğruluk ve tutarlılığın, hasta sonuçlarını nasıl iyileştirdiğini ortaya koymuştur.

Yapay zekâ tabanlı sağlık platformları, hastalara evlerinden çıkmadan tıbbi danışmanlık alabilme imkânı sunarak, sağlık hizmetlerinin erişilebilirliğini artırmaktadır. Sutton vd. (2019) bu teknolojilerin, özellikle kırsal ve ulaşımı zor bölgelerde yaşayan hastalar için faydalarını vurgulamıştır. Sağlık profesyonellerinin eğitiminde ve sürekli profesyonel gelişiminde kullanılmaktadır. Bu teknolojiler, karmaşık tıbbi bilgileri anlamada ve uygulamada yardımcı olur; ayrıca simülasyonlar ve interaktif öğrenme platformları aracılığıyla pratik becerilerin geliştirilmesini destekler.

Yapay zekânın sağlık hizmetlerinde kullanımı, etik ve yönetimsel zorlukları da beraberinde getirir. Veri gizliliği, güvenlik ve karar verme sürecinin şeffaflığı gibi konular,

bu teknolojilerin etik ve sorumlu bir şekilde kullanılması gerektiğini vurgular. Mehta vd. (2021), bu zorlukların üstesinden gelmek için gereken düzenleyici çerçeveleri ve politikaları tartışmıştır. Özet olarak yapay zekâ, karar destek sistemleri ve makine öğreniminin sağlık hizmetlerindeki uygulamaları sürekli gelişmektedir. Gelecekte, bu teknolojilerin daha da sofistike hale gelmesi ve geniş bir yelpazede sağlık uygulamalarına entegre edilmesi beklenmektedir. Bu gelişmeler, sağlık hizmetlerinin daha da kişiselleştirilmesine ve global sağlık sorunlarına çözüm üretilmesine katkı sağlayacaktır. Yapay zekâ, karar destek sistemleri ve makine öğrenimi sağlık hizmetlerinin kalitesini artırmakta ve erişilebilirliğini genişletmektedir. Ancak, yapay zekânın sağlık hizmetlerinde kullanımı konusunda etik ve güvenlik sorunları da bulunmaktadır. Yapay zekâ, sağlık sektöründe önemli değişiklikler yaratabilecek güçlü bir araçtır. Yapay zekânın sağlık hizmetlerine etkisi, yapay zekânın tıbbi teşhislerdeki kullanımı ve yapay zekânın sağlık hizmetlerinde karşılaşılan etik ve güvenlik sorunları hakkında daha fazla araştırma yapılması gerekmektedir.

1.2. Makine Öğrenmesi

Makine öğrenimi, ilk kez Alan Turing'in 1950'lerde önerdiği öğrenebilen ve yapay olarak zeki olabilen makineler konseptine dayanmaktadır. O zamandan bu yana, makine öğrenimi yüz tanıma gibi güvenlik hizmetlerinden kamu taşımacılığında verimliliği artırma ve riski azaltmaya kadar çeşitli alanlarda kullanılmaktadır. Son dönemde, sağlık hizmetleri ve biyoteknoloji alanlarında da önemli uygulamalara sahiptir. Yapay zekâ ve makine öğrenimi, iş süreçlerinde önemli değişiklikler yaratmış ve günlük yaşamları dönüştürmüştür. Benzer dönüşümlerin sağlık hizmetlerinde de gerçekleşmesi beklenmektedir.

Bu alandaki son gelişmeler, doktorların yükünü azaltmak ve doğruluk, tahmin ve bakım kalitesini artırmak için büyük fırsatlar ve ilerlemeler sunmaktadır. Sağlık hizmetlerinde makine öğreniminin mevcut ilerlemeleri, genellikle doktor veya analist rollerini destekleyici nitelikte olup ve hastalık tahmin modelleri geliştirmekte kullanılmaktadır. Büyük tıbbi kuruluşlarda, makine öğrenimi tabanlı yaklaşımlar elektronik sağlık kayıtlarının düzenlenmesinde, kan örneklerindeki, organlardaki ve kemiklerdeki düzensizliklerin tespit edilmesinde ve robot yardımlı cerrahilerde artan verimlilik sağlamak amacıyla uygulanmaktadır.

Yapay zekâ, insan zekâsını öğrenen ve taklit eden herhangi bir bilgisayarlı zekâ anlamına gelen şemsiye bir terimdir. Yapay zekâ, en çok robotlar ve kendi kendine giden

arabalar gibi özerk makinelerle tanınır, ancak kişiselleştirilmiş reklamlar ve web aramaları gibi günlük uygulamalara da nüfuz etmektedir. Son yıllarda, yapay zekâ geliştirme ve uygulamaları inanılmaz ilerlemeler kaydetmiş ve yüksek karar verme düzeyleri, doğruluk, problem çözme yeteneği ve hesaplama becerileri nedeniyle birçok alanda uygulanmıştır.

Yapay zekâ algoritmalarının geliştirilmesinde, elde edilen veriler genellikle iki gruba ayrılır: Eğitim ve test veri seti. Bu ayrım, güvenilir öğrenmeyi, temsilci popülasyonları ve tarafsız tahminleri sağlamak amacıyla yapılır. Eğitim seti, algoritmanın eğitiminde kullanılan özellikler ve bunlara karşılık gelen tahminlerden (denetimli öğrenme durumunda) oluşur. Test veri seti ise algoritmaya yeni olup, algoritmanın yeteneklerini test etmek için kullanılır. Bu yöntem, eğitim veri seti tarafından algoritmanın test edilmesinde oluşabilecek yanlılığı ortadan kaldırmak için alınır. Algoritma, eğitim ve test aşamalarından kabul edilebilir sonuçlarla geçtikten sonra sağlık hizmetleri ortamlarında uygulanır.

Makine öğrenimi, sağlık hizmetlerinde hastalık tahmini, hastane sonuçlarının belirlenmesi ve görüntü tespiti gibi çeşitli alanlarda geniş çapta uygulanmaktadır. Örneğin, Liu, Zhang ve Razavian, elektronik sağlık kayıtlarını (EHR) kullanarak hastalıkların başlangıcını tahmin etmek için bir derin öğrenme algoritması geliştirmiştir. Bu algoritma, yapılandırılmış verilerle birlikte EHR'den elde edilen serbest metin verilerini kullanarak, kalp yetmezliği, böbrek yetmezliği ve inme gibi hastalıkların başlangıcını tahmin edebilmiştir.

Büyük veri setlerini kullanarak makine öğrenimi algoritmaları ile popülasyon düzeyinde sağlık sonuçları tahmin edilebilir. Makine öğrenimi algoritmaları, büyük veri setleri ve sonuç ile diğer bağımsız değişkenler arasındaki doğrusal olmayan ilişkiler olduğunda avantajlıdır. Birçok makine öğrenimi algoritması tekrarlanabilir ve bu algoritmaların gerçek özü, altta yatan otantik ve yüksek kaliteli verilere dayanır. Sağlık hizmetlerinde tahmin modelleri yeni değildir; örneğin, 1967'de formüle edilen Framingham sağlık skoru. Parametrik istatistikler geçmişte çeşitli popülasyon düzeyindeki sonuçları tahmin etmek için kullanılmıştır; ancak bu sonuçları tahmin etmek için çeşitli makine öğrenimi algoritmalarını kullanma ihtiyacı ve ilgisi son zamanlarda artmıştır. Makine öğrenimi modellerinin formüle edilmesinde kullanılan teorik bilgi ve ilkeler bir süredir var olmasına rağmen, çeşitli halk sağlığı organizasyonlarında büyük miktarda değişken verinin yüksek hızda üretilmesi sürecinde son zamanlarda bir patlama olmuştur. Ek olarak, çeşitli donanım ve bulut çözümleri aracılığıyla verileri işleme yeteneğinin geliştirilmesi, karmaşık algoritmaların büyük veri için etkili bir şekilde kullanılma yeteneği artırmıştır.

Makine öğrenimi (ML) teknolojileri, klinik deneme araştırmalarını geliştirmek için geniş bir kullanım yelpazesi sunmaktadır. Klinik deneme adaylarına karmaşık öngörücü analitikler uygulayarak, tıbbi profesyoneller daha geniş bir veri yelpazesini değerlendirebilir, tıbbi testlerin maliyetini ve süresini azaltabilirler. Makine öğrenimi uygulamaları, klinik deneme verimliliğini artırmada çeşitli yollar sunmaktadır. Örneğin, optimal örnek boyutlarını belirlemeye yardımcı olabilir ve elektronik sağlık kayıtları (EHR) kullanarak veri hatalarını azaltabilir. Bu yöntem, dünya çapında iyi eğitimli radyologların giderek azaldığı sağlık alanında önemli bir sorunu ele almaktadır.

Sağlık iş dünyasında makine öğreniminin bir başka faydası, deneme katılımcılarına gerçek zamanlı olarak hizmet vermektir. Bu tür teknolojiler, araştırmacıların test için optimal örnek boyutunu belirlemelerine ve elektronik kayıtları kullanarak veri tabanı hatalarını ortadan kaldırmalarına da yardımcı olabilir. Makine öğrenimi teknolojileri, klinik denemelerde daha doğru ve güvenilir sonuçlar elde edilmesini sağlayarak, sağlık hizmetlerinin genel kalitesini artırabilir ve maliyetleri düşürebilir.

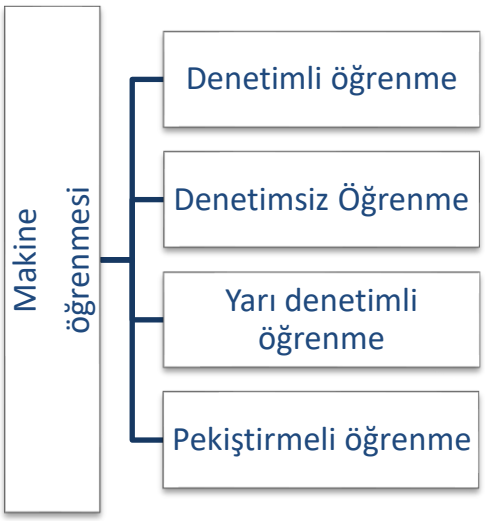
Tıbbi sektörde doğrudan yer almasının yanı sıra, makine öğrenimi kritik bir önleyici rol oynar. Bu teknoloji, uzmanların şu anda görünmeyen ancak hayatımızı tehdit edebilecek olası kusurları belirlemelerine olanak tanır. Bu, ortaya çıkan hastalıklar arasında yer alabilir ve bunları manuel olarak tespit etmek zordur; makine öğrenimi, hastanın rahatsızlığını teşhis etmede, sağlığını izlemekte ve gerekli adımları önermede önemli bir rol oynar. Küçük rahatsızlıklardan kanser gibi erken tespiti zor olan ciddi hastalıklara kadar değişebilir.

Küresel olarak veya belirli demografik sektörler arasında psikolojik sağlık sorunlarını öğrenmek ve tahmin etmek, sağlık alanında makine öğrenimi kullanım örneklerinden biridir. Bu, psikolojik sağlık uzmanlarının tanımlamalarını sağlar ve bu alanda karşılaşılan zorlukları anlamalarına yardımcı olur. Klinik denemelerin düzenlenmesi daima zaman alıcı ve emek yoğun bir işlem olmuştur ve birçok faktörü göz önünde bulundurmak gereklidir. Bu nedenle, potansiyel klinik deneme adayları, çeşitli kriterlere göre titizlikle incelenmelidir. Bu teknoloji, ilaçların güvenli ve etkili olduğundan emin olmak için büyük miktarda veriyi sürekli olarak izler ve değerlendirir. Makine öğrenimi, makinelerin kavramları öğrenmesine, verileri yorumlamasına ve temel yaşam tarzı değişiklikleri önermesine olanak tanır.

1.2.1. Makine Öğrenimi Modellerinin Türleri

Makine öğrenimi (ML), çeşitli veri tipleri ve uygulama alanları için farklı öğrenme modelleri sunarak geniş bir yelpazede değer yaratır. Temel olarak, bu modeller öğrenme süreçlerine göre sınıflandırılır ve her model, belirli bir veri seti veya problem türüne göre özelleştirilmiş çözümler üretmek için tasarlanmıştır. Günümüzde teknolojinin ilerlemesiyle birlikte, sağlık, finans, otomasyon ve daha birçok alanda bu modellerin rolü giderek artmakta; böylece daha akıllı, etkili ve öngörülebilir sistemlerin geliştirilmesine olanak sağlanmaktadır. Bu çeşitlilik, makine öğreniminin adaptasyon kabiliyetini ve farklı zorluklara uyum sağlama potansiyelini göstermektedir.

Makine öğrenimi modelleri genellikle denetimli, denetimsiz, yarı denetimli ve pekiştirmeli öğrenme olmak üzere dört ana kategori altında incelenir. Denetimli öğrenme, belirli görevleri yerine getirirken doğru sonuçları üretmek için etiketli verilerden faydalanırken, denetimsiz öğrenme verilerdeki gizli yapıları ve grupları keşfetmeye odaklanır. Yarı denetimli öğrenme, her iki modelin en iyi yanlarını birleştirerek, etiketli ve etiketlenmemiş verileri bir arada kullanır. Son olarak, pekiştirmeli öğrenme, belirli hedeflere ulaşmak için ödül mekanizmalarını temel alan bir süreç izler. Bu çeşitlilik, makine öğrenimi modellerinin farklı veri setleri ve uygulama senaryoları için nasıl uyum sağlayabildiğini ve çözüm sunduğunu ortaya koymaktadır.



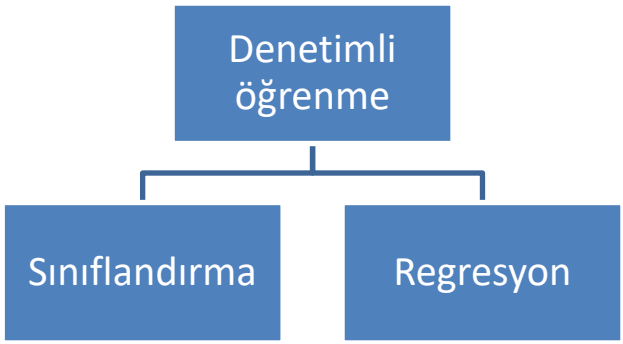
Şekil 2. Makine Öğrenim Yöntemleri

Denetimli Öğrenme (Supervised Learning)

Denetimli öğrenme, makine öğrenimi algoritmalarının en yaygın kullanılan türlerinden biridir. Bu öğrenme modeli, her bir giriş verisi (input) için önceden belirlenmiş çıkış verileri (output) ile eğitilir. Eğitim sürecinde, algoritma giriş ve çıkış verileri arasındaki ilişkiyi modelleyerek bir tahmin fonksiyonu oluşturur. Bu süreç, algoritmanın yeni, görülmemiş verilere karşı doğru tahminler yapabilmesini sağlar. Temel amacı, verilen giriş verilerine en uygun çıkışı üretebilmek olan bu model, özellikle etiketli veri gerektiren durumlar için idealdir.

Denetimli öğrenme, genellikle iki ana kategori altında incelenir: sınıflandırma ve regresyon. Sınıflandırma, giriş verilerini önceden belirlenmiş kategorilere ayırır. Örneğin, e-postaların spam ya da spam olmayan olarak sınıflandırılması veya bir görüntünün içerdiği objenin tespit edilmesi bu türe örnek verilebilir. Regresyon ise sürekli değerlerin tahmin edilmesiyle ilgilenir; bir evin değerini tahmin etmek veya bir aracın yakıt tüketimini hesaplamak gibi.

Sağlık sektörü, denetimli öğrenme modellerinin en kritik uygulama alanlarından biridir. Bu alanda, denetimli öğrenme algoritmaları, hastalıkların teşhisinden tedavi yöntemlerinin belirlenmesine kadar geniş bir kullanım alanına sahiptir. Örneğin, tıbbi görüntüler üzerinde yapılan çalışmalarda, çeşitli görüntü işleme teknikleri ile hastalıklı doku ve sağlıklı doku arasındaki farkların sınıflandırılması sağlanabilir. Ayrıca, hastaların gelecekteki sağlık durumlarının tahmin edilmesi gibi regresyon uygulamaları da bu alanda sıkça tercih edilir.



Şekil 3. Denetimli Öğrenme

Sınıflandırma (Classification)

Bu model türü, etiketli veri setleri kullanarak, her bir girişin hangi kategoriye ait olduğunu öğrenir ve bu bilgiyi yeni verileri sınıflandırmak için kullanır. Tipik bir sınıflandırma problemi, verileri iki veya daha fazla sınıfa ayırmayı gerektirir; örneğin, bir banka işleminin dolandırıcılık içerip içermediğini tespit etmek. Algoritma, giriş özelliklerine dayanarak, her veri noktasını en uygun sınıfa atar, bu süreç genellikle bir puanlama veya olasılık tahmini içerir.

Sınıflandırma algoritmaları arasında karar ağaçları, destek vektör makineleri (SVM) ve naive bayes gibi yöntemler bulunur. Her biri farklı veri yapıları ve ihtiyaçları için özelleştirilmiş çözümler sunar. Örneğin, karar ağaçları, karar kuralları serisiyle sınıflandırma yaparken; SVM, veri sınıfları arasında en iyi ayrımı sağlayacak hiper-düzlemi bulmaya çalışır. Bu algoritmalar, tıptan finansa, perakendecilikten siber güvenliğe kadar çeşitli alanlarda uygulanabilir. Sağlık sektöründe, bu algoritmalar tıbbi teşhislerde kullanılarak belirli hastalıkları sınıflandırmak için etkili sonuçlar verir.

Regresyon (Regression)

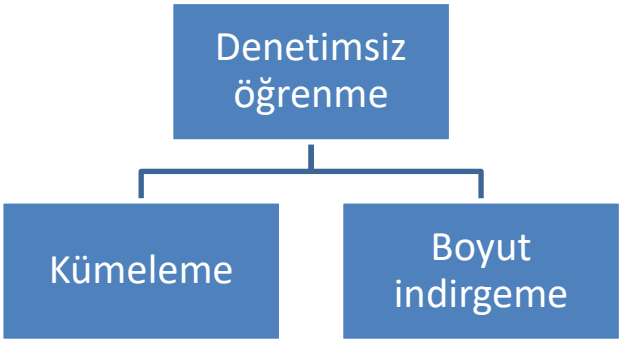
Regresyon, denetimli öğrenme modelleri içinde yer alır ve sürekli (sayısal) değerlerin tahmin edilmesine odaklanır. Bu yöntem, giriş verileri ile bir veya daha fazla bağımsız değişken arasındaki ilişkiyi modelleyerek, bir çıktı (bağımlı değişken) değerini tahmin etmeye çalışır. Temel regresyon modellerinden biri olan lineer regresyon, veriler arasındaki doğrusal ilişkileri belirlemek için kullanılır. Regresyon analizi, özellikle ekonomi, mühendislik, biyoloji ve tıp gibi alanlarda, sürekli veriler üzerinde tahmin yürütmek ve değişkenler arası ilişkileri anlamak için yaygın olarak kullanılır.

Regresyon teknikleri arasında en çok bilinenler lineer regresyon, lojistik regresyon, polinom regresyon ve ridge regresyon gibi yöntemlerdir. Lineer regresyon, iki değişken arasındaki doğrusal ilişkiyi ifade ederken, lojistik regresyon özellikle ikili sonuçlar (evet/hayır, 0/1 gibi) tahmin etmek için kullanılır. Polinom regresyon, veriler arasındaki ilişkinin daha karmaşık olduğu durumlarda, yüksek dereceli polinomlar kullanarak bu ilişkileri modellemeye çalışır. Bu teknikler, fiyat tahminleri, hava durumu tahmini, hastalık risk analizi ve daha birçok alanda kullanılarak, karar verme süreçlerine büyük katkılar sağlar. Özellikle sağlık sektöründe, hasta sonuçlarının tahmini ve tedavi başarısının değerlendirilmesi gibi kritik uygulamalar için önemlidir.

Denetimsiz Öğrenme (Unsupervised Learning)

Denetimsiz öğrenme, etiketlenmemiş verilerle çalışarak veri setinin içerisindeki gizli yapıları, grupları veya desenleri ortaya çıkarmayı amaçlayan bir makine öğrenimi yaklaşımıdır. Bu öğrenme türü, veri üzerinde herhangi bir önceden belirlenmiş etiket veya kategori olmadan işlem yapar bu nedenle model, veriler arasındaki ilişkileri ve benzerlikleri kendiliğinden keşfetmeye çalışır. Denetimsiz öğrenme, genellikle büyük ve karmaşık veri kümeleri üzerinde yararlı bilgileri ve örüntüleri bulmak için kullanılır. Kümeleme ve boyut azaltma, denetimsiz öğrenmenin en yaygın iki yöntemidir.

Denetimsiz öğrenmenin iki ana dalı olan kümeleme ve boyut azaltma, farklı amaçlara hizmet eder. Kümeleme, benzer özelliklere sahip veri noktalarını gruplar halinde organize etmek için kullanılır. Bu yöntem, müşteri segmentasyonu, genetik kümelenme veya sosyal ağ analizi gibi alanlarda uygulanabilir. Öte yandan, boyut azaltma, veri setindeki özellik sayısını azaltarak daha basit ve anlaşılır bir veri yapısı elde etmeyi amaçlar; bu, özellikle yüksek boyutlu verilerin görselleştirilmesi ve analiz edilmesi sırasında faydalıdır. Principal Component Analysis (PCA) ve t-Distributed Stochastic Neighbor Embedding (t-SNE), boyut azaltma için kullanılan popüler tekniklerdendir. Denetimsiz öğrenme yöntemleri, verilerdeki gizli kalıpları ortaya çıkarma potansiyeline sahip olduğundan, bilimsel araştırma, müşteri analitikleri ve biyoinformatik gibi çeşitli disiplinlerde değerli içgörüler sağlar. PCA, verilerin varyansını maksimize ederek en önemli özelliklerini korurken, veri boyutunu düşürür. Bu yöntem, genellikle lineer veri setlerinde etkilidir ve finans, biyoenformatik ve sosyal bilimlerde kullanılır.



Şekil 4. Denetimsiz Öğrenme

Kümeleme (Clustering)

Kümeleme, denetimsiz öğrenme tekniklerinden biri olarak, veri noktalarını benzerliklerine göre gruplara ayırmayı amaçlar. Bu teknik, herhangi bir etiketleme veya sınıflandırma girişi olmadan veriler arasındaki doğal gruplamaları keşfetmek için kullanılır. Kümeleme algoritmaları, veri seti içindeki benzer özellikleri taşıyan veri noktalarını tespit eder ve bu noktaları bir araya getirerek gruplar oluşturur. Bu gruplar, veri içindeki yapıları, desenleri ve ilişkileri anlamak için temel teşkil eder. Örneğin, müşteri davranış verileri kullanılarak yapılan kümeleme, benzer alışveriş eğilimlerine sahip müşterileri belirlemek için kullanılabilir.

Kümeleme algoritmaları arasında k-means, hiyerarşik kümeleme ve DBSCAN gibi yöntemler yer alır. K-means algoritması, verileri k sayıda kümeye ayırırken, Hiyerarşik Kümeleme, veri noktalarını adım adım birleştirerek veya bölerek ağaç benzeri bir yapı oluşturur. DBSCAN, yoğunluk bazlı bir yaklaşım kullanarak, veri noktalarının yoğun olduğu bölgeleri tespit eder ve bu bölgeleri ayrı kümeler olarak gruplandırır. Bu algoritmalar, genetik araştırmalarda, pazar segmentasyonunda, sosyal ağ analizlerinde ve hatta astronomi ve ekoloji gibi alanlarda kullanılarak, geniş veri setleri içinde anlamlı bilgilerin çıkarılmasında etkili olurlar. Kümeleme, bu çeşitli uygulamalarıyla, veri bilimindeki en temel ve kritik tekniklerden biridir.

Boyut İndirgeme (Dimensionality Reduction)

Boyut indirgeme, yüksek boyutlu veri setlerini daha yönetilebilir ve anlaşılır hale getirmek için kullanılan bir denetimsiz öğrenme tekniğidir. Bu süreç, veri setindeki özellik sayısını azaltırken, orijinal veri setinin önemli bilgilerini mümkün olduğunca korumayı amaçlar. Boyut indirgeme, özellikle görselleştirme, veri saklama ve işleme süreçlerini kolaylaştırmak ve yüksek boyutlu verilerde sıkça karşılaşılan bilgi kaybı problemini hafifletmek için kritik öneme sahiptir. Bu teknik, veri setlerinde gürültünün azaltılmasına ve model performansının artırılmasına yardımcı olur, böylece daha hızlı ve daha doğru analizler yapılabilir.

En yaygın boyut indirgeme teknikleri arasında Principal Component Analysis (PCA) ve t-Distributed Stochastic Neighbor Embedding (t-SNE) bulunmaktadır. t-SNE özellikle yüksek boyutlu veri setlerindeki karmaşık yapıları modelleyerek veri noktaları arasındaki benzerlikleri koruyacak şekilde düşük boyutlu temsiller oluşturur; bu sayede veri görselleştirmesi için idealdir. Boyut indirgeme teknikleri, genetik sekans analizi, yüz tanıma sistemleri ve müşteri segmentasyonu gibi çeşitli uygulamalarda kullanılarak değerli içgörüler sağlar ve büyük veri setlerinin analizini kolaylaştırır.

Yarı Denetimli Öğrenme (Semi-Supervised Learning)

Yarı denetimli öğrenme hem etiketli hem de etiketlenmemiş verileri kullanarak model eğitimi gerçekleştiren bir makine öğrenimi yaklaşımıdır. Bu öğrenme türü, genellikle etiketli veri miktarının sınırlı olduğu ancak büyük miktarda etiketlenmemiş verinin mevcut olduğu durumlar için idealdir. Yarı denetimli öğrenme, etiketli verilerle modelin temel yapısını kurarken, etiketlenmemiş verileri kullanarak modelin genelleme yeteneğini artırmak ve daha geniş veri setleri üzerinden öğrenimini derinleştirmek için kullanılır. Bu yöntem, veri etiketleme maliyetlerini azaltırken, geniş veri setlerinden elde edilen bilgiyi maksimize etme avantajı sunar.

Yarı denetimli öğrenme teknikleri arasında, grafik tabanlı modeller ve yarı denetimli destek vektör makineleri (Semi-Supervised SVM) bulunmaktadır. Bu yöntemler, özellikle doğal dil işleme, görüntü tanıma ve biyoenformatik gibi alanlarda yaygın olarak kullanılmaktadır. Örneğin, bir dil modelinin eğitiminde sadece bir kısmı etiketlenmiş büyük metin korpuslarından yararlanılabilir. Yarı denetimli öğrenme, etiketli verilerin kısıtlı olduğu tıbbi görüntüleme ve genetik sınıflandırma çalışmalarında da önemli rol oynar; bu sayede daha az etiketle daha fazla veri noktası kullanılarak daha güçlü ve doğru modeller oluşturulabilir.

Pekiştirmeli Öğrenme (Reinforcement Learning)

Pekiştirmeli öğrenme, bir ajanın (agent) çevresiyle etkileşim içinde bulunarak ve gerçekleştirdiği eylemler sonucunda aldığı ödüller veya cezalar aracılığıyla optimal davranış stratejileri geliştirmesini amaçlayan bir makine öğrenimi yaklaşımıdır. Bu öğrenme türü, ajanın karşılaştığı duruma bağlı olarak en uygun eylemi seçmesini ve uzun vadeli ödülleri maksimize etmesini hedefler. Pekiştirmeli öğrenme modelleri, deneme-yanılma yoluyla ve ödül mekanizması üzerinden kendini sürekli olarak geliştirir. Bu yöntem, karar verme süreçlerinin dinamik olduğu ve ajanın çevresinden sürekli geri bildirim aldığı uygulamalarda özellikle etkilidir.

Pekiştirmeli öğrenme teknikleri arasında Q-Learning, Deep Q Networks (DQN), ve Policy Gradient gibi yöntemler bulunur. Bu teknikler, video oyunlarından otomatik araç sürüş sistemlerine, robotikten finansal strateji geliştirmeye kadar geniş bir yelpazede

uygulanabilir. Örneğin, video oyunlarında karakterlerin çevreyle etkileşim kurarak görevleri tamamlaması veya otomatik araçların trafikte güvenli ve etkili bir şekilde manevra yapması için pekiştirmeli öğrenme kullanılabilir. Ayrıca, e-ticarette de pekiştirmeli öğrenme modelleri piyasa koşullarına göre alım-satım kararları alarak finansal getiriyi maksimize etme stratejileri geliştirebilir.

1.2.2. Naive Bayes Sınıflandırma Yöntemi

Naive Bayes, olasılık temelli bir sınıflandırma tekniğidir ve Bayes Teoremi'ne dayanır. Bu model, özellikler arasındaki bağımlılıkları göz ardı eden (naive) basit bir yaklaşım benimser, bu da algoritmayı uygulamayı ve hesaplamayı kolaylaştırır. Her bir sınıfın verilen bir girişe ait olma olasılığını hesaplamak için kullanılır. Naive Bayes, metin sınıflandırma, spam tespiti ve duygu analizi gibi alanlarda yaygın olarak tercih edilen etkili bir yöntemdir çünkü büyük veri kümeleri üzerinde hızlı ve etkili sonuçlar üretebilir.

Bayes teoremi, $P(C|X)$ yani X veri kümesi gözlemlendiğinde C sınıfının olasılığı,

$$P(C|X) = \frac{(P(X|C).P(C))}{P(X)} \quad (1)$$

biçiminde hesaplanır. Burada $P(C|X)$, X veri kümesi verildiğinde C sınıfının olasılığı; $(P(X|C)$, C sınıfı verildiğinde X veri kümesinin gözlemlenme olasılığı; $P(C)$, C sınıfının önsel olasılığı; $P(X)$, X veri kümesinin toplam olasılığıdır.

1.2.3. Destek Vektör Makineleri

Destek Vektör Makineleri (SVM), sağlık verilerinin sınıflandırılmasında kullanılan güçlü bir makine öğrenme algoritmasıdır. SVM, özellikle tıbbi karar destek sistemlerinde, hastalık sınıflandırması ve triaj belirleme gibi kritik uygulamalarda etkili sonuçlar sunmaktadır. Bu algoritmanın temel amacı, veriyi en iyi şekilde ayıran hiper düzlemi bulmaktır. Bu hiper düzlem, farklı sınıflara ait tıbbi veriler arasındaki mesafeyi maksimuma çıkararak daha doğru ve güvenilir sınıflandırmalar yapmayı sağlar.

SVM, veri noktalarını sınıflandırmak için optimal bir hiper düzlem bulur. Bu hiper düzlem, matematiksel olarak

$$\omega . x + b = 0 \tag{2}$$

burada ω hiper düzlemin normal vektörünü, x veri noktalarını ve b ise bir sabiti temsil eder. Algoritma, bu hiper düzlemi belirlerken, iki sınıf arasındaki marjini maksimuma çıkaran ω ve b değerlerini optimize eder. Bu optimizasyon problemi, aşağıdaki gibi formüle edilir:

$$Minimize \frac{1}{2} \|w\|^2 \tag{3}$$

Bu optimizasyon, marjin koşulu

$$y_i(\omega . x_i + b) \geq 1 \tag{4}$$

Altında gerçekleştirilir. Burada y_i her bir veri noktasının sınıf etiketini, x_i ise veri noktalarını ifade eder. Amaç, hiper düzlemi sınıflar arasındaki mesafeyi maksimuma çıkaracak şekilde yerleştirerek tıbbi verilerin doğru bir şekilde sınıflandırılmasını sağlamaktır.

SVM, doğrusal olarak ayrıştırılamayan tıbbi veriler için de genişletilebilir. Bu durumda, veriler daha yüksek boyutlu bir uzaya haritalanarak doğrusal bir ayrım sağlanır. Bu işlem, çekirdek (kernel) fonksiyonları kullanılarak gerçekleştirilir. Özellikle tıbbi verilerin karmaşık yapısı göz önüne alındığında, çekirdek fonksiyonları SVM'in başarısını artırır. Yaygın olarak kullanılan çekirdek fonksiyonları arasında doğrusal, polinomial, radyal bazlı fonksiyon (RBF) ve sigmoid yer alır.

Örneğin, RBF çekirdek fonksiyonu, iki veri noktası arasındaki mesafeyi Gaussian benzeri bir biçimde hesaplar ve şu şekilde ifade edilir:

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|\right)^2 \tag{5}$$

Burada γ bir parametre olup, çekirdek fonksiyonunun etkisini kontrol eder. Bu fonksiyon, tıbbi verilerin karmaşık ve doğrusal olmayan yapılarında bile yüksek doğruluk oranlarıyla sınıflandırma yapmayı sağlar.

1.2.4. Yapay Sinir Ağları

Yapay Sinir Ağları (Artificial Neural Networks, ANN, YSA), biyolojik sinir sistemlerinden ilham alarak geliştirilmiş, tıbbi verilerin işlenmesi ve analizi için kullanılan güçlü bir makine öğrenme yöntemidir. Yapay sinir ağları, karmaşık ve çok boyutlu verilerle başa çıkabilme yeteneği sayesinde, sağlık alanında hastalık teşhisi, tedavi önerileri ve triaj belirleme gibi çeşitli görevlerde yaygın olarak kullanılmaktadır. YSA, birçok nörondan oluşan katmanlar aracılığıyla verileri işler ve öğrenir. Her bir nöron, diğer nöronlarla bağlantılı olup, bu bağlantıların ağırlıkları öğrenme sürecinde optimize edilir.

YSA, temel olarak üç katmandan oluşur: girdi katmanı, gizli katmanlar ve çıktı katmanı. Girdi katmanı, tıbbi verileri (örneğin hasta yaşı, cinsiyeti, şikayetleri) alır ve bu verileri gizli katmanlara iletir. Gizli katmanlar, verileri işleyerek karmaşık ilişkileri öğrenir ve bu ilişkiler aracılığıyla sağlık verilerinin sınıflandırılması gerçekleştirilir. Son olarak, çıktı katmanı, sınıflandırma sonucunu üretir. Bu işlem, aşağıdaki formül ile gösterilebilir:

$$y = f\left(\sum_{i=1}^{n} w_i . x_i + b\right) \tag{6}$$

Burada y nöronun çıktısını, x_i girdi verilerini, w_i ağırlıkları ve b ise bias terimini temsil eder. f ise aktivasyon fonksiyonudur ve nöronun çıktısının belirlenmesinde kullanılır.

YSA'nın öğrenme süreci, ileri besleme (feedforward) ve geri yayılım (backpropagation) algoritmaları aracılığıyla gerçekleştirilir. İleri besleme aşamasında, tıbbi veriler girdi katmanından geçirilir ve her katman arasında hesaplanan ağırlıklarla işlenir. Geri yayılım aşamasında ise, çıktı katmanından elde edilen hata geri beslenir ve bu hata kullanılarak ağırlıklar güncellenir. Bu süreç, ağırlıkların optimize edilmesi ve modelin doğruluğunun artırılması amacıyla tekrar tekrar uygulanır. Geri yayılım algoritması, ağırlıkları güncellemek için aşağıdaki gibi bir türev hesaplama işlemi gerçekleştirir:

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i} \tag{7}$$

burada Δw_i ağırlık değişimini, η öğrenme hızını ve E hata fonksiyonunu temsil eder. Bu işlem, ağırlıkların modelin performansını iyileştirecek şekilde güncellenmesini sağlar.

Yapay Sinir Ağları (YSA), farklı katman sayıları ve yapılarıyla sığ ve derin olmak üzere iki ana gruba ayrılabilir. Sığ YSA, genellikle yalnızca bir veya iki gizli katmandan oluşan sinir ağlarıdır. Bu tür ağlar, basit ve daha az karmaşık veriler üzerinde etkili olabilir, ancak tıbbi veriler gibi karmaşık ve çok boyutlu veri setlerinde performansları sınırlı kalabilir. Sığ YSA'nın avantajı, daha hızlı eğitilebilmesi ve daha az hesaplama gücü gerektirmesidir. Ancak, bu tür ağlar karmaşık ilişkileri ve yüksek boyutlu verileri modellemekte yetersiz kalabilir.

Sığ Yapay Sinir Ağları

Sığ YSA'nın matematiksel modeli, giriş verilerinin bir gizli katman üzerinden geçirilerek işlenmesini içerir. Her bir gizli katmandaki nöron, giriş verileriyle ilgili belirli bir özelliği öğrenir ve bu özellikler toplandığında, çıktı katmanında nihai karar verilir. Bir sığ YSA'nın çıktısı şu şekilde ifade edilebilir:

$$y = f\left(\sum_{i=1}^{n} w_i . x_i + b\right) \tag{8}$$

Sığ YSA'nın avantajları arasında daha hızlı eğitim süresi ve daha düşük hesaplama maliyeti bulunur. Özellikle sınırlı işlem gücü veya zaman kısıtlaması olan durumlarda önemlidir. Ayrıca, sığ YSA'lar, genellikle daha az miktarda eğitim verisiyle etkili bir şekilde çalışabilir. Ancak, bu tür ağların karmaşık ve çok boyutlu verilerde yeterli performansı sağlayamayabileceği unutulmamalıdır. Sığ yapının sınırlı sayıda katmanı, verilerdeki karmaşık ilişkilerin tam olarak öğrenilememesine yol açabilir.

Tıbbi verilerin sınıflandırılmasında sığ YSA, özellikle basit ilişkilerin ve temel kalıpların keşfedilmesi gerektiğinde kullanışlıdır. Örneğin, belirli bir hastalık için risk faktörlerinin hızlıca belirlenmesi veya hastaların temel demografik bilgilerine dayalı olarak ön sınıflandırma yapılması gibi görevlerde sığ YSA etkili olabilir. Ancak, daha karmaşık klinik verilerin işlenmesi gerektiğinde, derin YSA veya diğer gelişmiş yöntemlerin kullanılması gerekebilir.

Derin Yapay Sinir Ağları

Derin Yapay Sinir Ağları (Deep Neural Networks, DNN), çok sayıda gizli katmandan oluşan ve karmaşık veri yapılarının işlenmesinde kullanılan ileri düzey makine öğrenme modelleridir. Bu ağlar, her bir katmanda veriyi farklı seviyelerde işleyerek daha yüksek soyutlama düzeyleri elde eder. Derin YSA'lar, özellikle tıbbi verilerin karmaşık yapıları ve çok boyutlu ilişkileri üzerinde yüksek doğruluk oranlarıyla sınıflandırma yapabilme yetenekleriyle öne çıkar. Büyük veri setleri ve karmaşık desenler içeren veriler için derin YSA, en uygun ve etkili çözümlerden biri olarak kabul edilir.

Derin YSA'nın matematiksel modeli, çok sayıda gizli katmanın ardışık olarak veri işlediği bir yapıdadır. Bu modelde, her katmandaki nöronlar bir önceki katmandan gelen girdileri işler ve bu süreç her katman boyunca devam eder. Sonuç olarak, derin YSA'nın çıktısı aşağıdaki gibi formüle edilebilir:

$$y = f^{(L)}\left(W^L . f^{(L-1)}(W^{L-1} . \dots f^1(W^1 . x + b^1) + b^2) + b^L\right) \quad (9)$$

Burada L gizli katman sayısını W^i i'inci katmanın ağırlıklarını, f^i ise her bir katmanda kullanılan aktivasyon fonksiyonunu temsil eder. Derin YSA'ların karmaşık veri yapılarını öğrenme yeteneği, her bir katmanda veriyi işleyen bu çok katmanlı yapıdan gelir.

Derin YSA'ların en büyük avantajı, karmaşık ve çok boyutlu verilerdeki ilişkileri modelleyebilme kapasitesidir. Tıbbi verilerin analizi, hastalık teşhisi, genetik analizler, görüntü işleme (örneğin, MRI, CT taramaları) ve triaj belirleme gibi alanlarda derin YSA'lar, yüksek doğruluk oranları ve başarılı sonuçlar sunar. Bununla birlikte, derin YSA'lar, eğitim sürecinde büyük veri setlerine ve yüksek hesaplama gücüne ihtiyaç duyar. Ayrıca, aşırı uyum (overfitting) gibi sorunlar da dikkatle yönetilmelidir.

1.2.5. Ağaç Yöntemleri

Karar ağaçları, Breiman vd. (1984) tarafından tanıtılan ve Quinlan (1986) tarafından geliştirilmiş olan temel sınıflandırma algoritmalarından biridir. Bu algoritma, veri madenciliği ve makine öğrenmesi alanlarında geniş bir kabul görmüş ve özellikle anlaşılabilirliği, açıklanabilirliği ve basitliği sayesinde popüler olmuştur. Karar ağaçları,

klinik karar destek sistemlerinde de kullanılmış ve birçok çalışmada hasta teşhis ve tedavi süreçlerinin modellenmesinde etkili olduğu düşünülmüştür (Loh, 2011).

Ancak, literatürde yer alan çalışmalarda, karar ağaçlarının bazı sınırlamaları olduğu da ortaya konmuştur. Özellikle Duda ve Hart (1973), tarafından vurgulanan aşırı uyum problemi, karar ağaçlarının genelleme yeteneklerini zayıflatmış ve karmaşık veri setleri üzerinde yetersiz kalmalarına yol açmıştır. Bu nedenle, literatürde daha güçlü ve genellenebilir modellerin geliştirilmesine yönelik çalışmalar artmış ve bu arayışlar, Boosting gibi topluluk yöntemlerinin doğmasına neden olmuştur.

Boosting, Schapire (1990) tarafından önerilen ve daha sonra Freund ve Schapire (1997) tarafından geliştirilen bir topluluk öğrenme yöntemidir. Bu yöntem, bir dizi zayıf öğreniciyi (genellikle karar ağaçları) birleştirerek güçlü bir model oluşturmayı hedefler. Literatürde, Boosting yöntemlerinin karar ağaçlarına kıyasla daha yüksek doğruluk ve genelleme yeteneği sağladığı birçok çalışmada gösterilmiştir (Hastie, vd., 2009).

Boosting algoritmaları, karmaşık veri setlerinde, dengesiz veri dağılımlarında ve kritik süreçlerde yüksek doğruluk ve genelleme yeteneği sunarak karar ağaçlarının sınırlamalarını aşmıştır. Bu durum Boosting yöntemlerini, modern makine öğrenmesi uygulamalarında vazgeçilmez hale getirmiştir.

Rastgele Orman

Rastgele orman algoritmasının temel prensibi, bir veri kümesinin farklı alt örnekleri üzerinde birçok karar ağacı eğitmek ve ardından bu ağaçların sonuçlarını birleştirmektir. Bu süreç, modelin performansını artırır ve karar ağaçlarının bireysel olarak sahip olabileceği varyansı azaltır. Her bir ağacın tahmininin bir oy olarak kabul edildiği bu yöntem, sınıflandırma problemlerinde en çok oyu alan sınıfı, regresyon problemlerinde ise tahminlerin ortalamasını son tahmin olarak seçer. Rastgele orman algoritmasının matematiksel temeli şu şekildedir:

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} h_t(x) \tag{10}$$

Burada T oluşturulan karar ağaçlarının sayısını, $h_t(x)$ t karar ağacı tarafından yapılan tahmini, $\hat{y}$ modelin nihai tahminini ifade eder. Rastgele orman, özellikle yüksek boyutlu veri setlerinde ve çok sayıda özellik içeren durumlarda etkili sonuçlar verir.

Adaboost (Adaptive Boosting)

Adaboost, boosting algoritmalarının en bilinenlerinden biridir. Adaboost, her iterasyonda zayıf bir öğreniciyi eğitir ve bu öğrenicinin hatalarını düzelterek güçlü bir sınıflandırıcı oluşturur. Adaboost'un temel formülü şu şekildedir:

Başlangıç ağırlıkları belirlenir,

$$w_i^{(1)} = \frac{1}{N}(i = 1, 2, \ldots, N) \tag{11}$$

burada $w_i^{(1)}$ her bir örneğin başlangıç ağırlığıdır ve N toplam örnek sayısını temsil eder. Her iterasyonda t, zayıf bir öğrenici $h_t(x)$ eğitilir ve hata oranı ϵ_t hesaplanır,

$$\epsilon_t = \sum_{i=1}^{N} w_i^{(t)} I\big(y_i \neq h_t(x_i)\big) \tag{12}$$

burada $I\big(y_i \neq h_t(x_i)\big)$ bir gösterge fonksiyonu olup, yanlış sınıflandırılan örnekler için 1, doğru sınıflandırılanlar için 0 değerini alır. Zayıf öğrenicinin ağırlığı hesaplanır,

$$a_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \tag{13}$$

Zayıf öğrenicinin ağırlığının hesaplanır,

$$w_i^{(t+1)} = w_i^{(1)} \exp\left(-\alpha_t y_i h_t(x_i)\right. \tag{14}$$

Ağırlıklar normalize edilerek toplamlarının 1 olmasına dikkat edilir:

$$w_i^{(t+1)} = \frac{w_i^{(t+1)}}{\sum_{j=1}^{N} w_j^{t+1}} \tag{15}$$

Nihai model, tüm zayıf öğrenicilerin ağırlıklı çoğunluk kararı olarak hesaplanır:

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right) \tag{16}$$

Gradient Boosting

Gradient Boosting, bir kayıp fonksiyonunu minimize ederek çalışır ve hataları düzeltmek için gradyan iniş algoritmasını kullanır. Bu yöntemin temel adımları şunlardır (Friedman, 2001):
Başlangıç modeli genellikle sabit bir değer olarak seçilir:

$$F_0(X) = \arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, \gamma) \tag{17}$$

burada L kayıp fonksiyonu temsil eder. Her iterasyonda t ve gradyanlar hesaplanır:

$$r_i^{(t)} = -\left[\frac{\partial L\left(y_i, F(x_i)\right)}{\partial F(x_i)}\right]_{F=F_{t-1}} \tag{18}$$

Gradyanlara dayalı olarak zayıf öğrenici $h_t(x)$ eğitilir:

$$h_t(x)\ 0 \arg\min_{h} \sum_{i=1}^{N} \left(r_i^{(t)} - h(x_i)\right)^2 \tag{19}$$

Model güncellenir:

$$F_t(x) = F_{t-1}(x) + v.h_t(x) \tag{20}$$

Burada v öğrenme oranıdır ve modeli fazla uyumdan (overfitting) korumak için küçük bir değer seçilir.

CatBoost

CatBoost, özellikle kategorik verilerle çalışmak üzere optimize edilmiş bir gradient boosting algoritmasıdır. Yandex tarafından geliştirilen bu algoritma, kategorik değişkenlerle etkili bir şekilde çalışabilme yeteneğiyle öne çıkar ve bu sayede yüksek doğruluk oranları ve hızlı eğitim süreleri sağlar. CatBoost, özellikle karmaşık ve yüksek boyutlu veri setlerinde başarılı performans sunar ve aşırı uydurma (overfitting) riskini minimize eden güçlü düzenlileştirme teknikleri içerir.

CatBoost'un en önemli özelliklerinden biri, kategorik değişkenleri otomatik olarak işleyebilmesidir. Klasik gradient boosting algoritmalarında kategorik verilerin sayısal verilere dönüştürülmesi gerekirken, CatBoost bu dönüşümü kendi içinde gerçekleştirir. Bu, modelin performansını artırırken veri ön işleme sürecini de basitleştirir. CatBoost, bu işlemi "ordered boosting" yöntemiyle yapar. Bu yöntemde, veri seti sıralı bir şekilde işlenir ve önceki veriler, daha sonraki veriler için kategorik özelliklerin kodlanmasında kullanılır.

Matematiksel olarak, CatBoost modelinin öğrenme süreci şu şekilde ifade edilebilir:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta . f_t(x_i) \tag{21}$$

Burada $\hat{y}_i^{(t)}$ i'inci veri noktası için t'inci iterasyonda yapılan tahmin, $\hat{y}_i^{(t-1)}$ Bir önceki iterasyondaki tahmin, η öğrenme oranı, $f_t(x_i)$ t'inci iterasyonda öğrenilen yeni zayıf model. CatBoost, ayrıca verideki sıra hatalarını azaltmak ve modelin genelleme kabiliyetini artırmak için bir rastgeleleştirme süreci uygular. Bu süreçte, veriler rastgele sırayla işlenir ve her iterasyonda farklı bir sıralama ile kategorik değişkenler kodlanır. Bu, modelin overfitting yapmasını önlemeye yardımcı olur ve genel performansını artırır.

CatBoost, klinik karar destek sistemlerinde (KDS) de yaygın olarak kullanılmaya başlanmıştır. Özellikle, büyük miktarda kategorik veri içeren tıbbi veri setlerinde CatBoost, hızlı ve etkili çözümler sunar. Örneğin, hastaların demografik bilgileri ve geçmiş hastalık verileri gibi kategorik değişkenlerin yoğun olduğu durumlarda, CatBoost algoritması yüksek doğruluk oranlarıyla etkili bir sınıflandırma yapabilir.

LightGBM

LightGBM, Microsoft tarafından geliştirilen, özellikle büyük veri setlerinde ve yüksek boyutlu problemlerde etkili çalışan bir gradient boosting algoritmasıdır (Ke vd., 2017). LightGBM, bölünmüş kazanç tabanlı öğrenme (leaf-wise tree growth) ve histogram tabanlı hesaplama gibi tekniklerle eğitim hızını artırır ve bellek kullanımını optimize eder. Bu algoritma, özellikle büyük ve karmaşık veri setleri üzerinde hızlı eğitim süreleri ve yüksek doğruluk oranları sunarak öne çıkar.

LightGBM'in temel amacı, önceki modellerin hatalarını azaltarak doğruluğu artırmaktır. Her iterasyonda, kayıp fonksiyonunu minimize eden yeni bir model eklenir. Formül, gradient boosting yapısıyla benzerdir:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta . f_t(x_i)$$

Burada $\hat{y}_i^{(t)}$ i'inci veri noktası için t'inci iterasyonda yapılan tahmin, $\hat{y}_i^{(t-1)}$ Bir önceki iterasyondaki tahmin, η öğrenme oranı, $f_t(x_i)$ t'inci iterasyonda öğrenilen yeni zayıf model. LightGBM, aşırı öğrenmeyi (overfitting) önlemek için L1 ve L2 düzenlileştirme (regularization) tekniklerini de uygular. Hedef fonksiyonu şu şekildedir:

$$Obj(t) = \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{k=1}^{t} \Omega f(k) \tag{22}$$

Burada $L\left(y_i, \hat{y}_i^{(t)}\right)$ Gerçek değer y_i ile tahmin edilen değer $\hat{y}_i^{(t)}$ arasındaki kayıp fonksiyonu, $\sum_{k=1}^{t} \Omega f(k)$ ağaçların karmaşıklığını azaltan düzenleme terimi. LightGBM, hız, doğruluk ve ölçeklenebilirlik açısından güçlü bir boosting algoritmasıdır. Özellikle büyük ve karmaşık veri setleri ile çalışırken, eğitim süresini optimize etmek isteyen araştırmacılar için ideal bir seçimdir. Klinik karar destek sistemleri gibi veri yoğun uygulamalarda kullanıldığında, doğru tahmin oranlarını artırarak hasta sonuçlarını iyileştirebilir.

XGBoost (Extreme Gradient Boosting)

XGBoost, gradient boosting algoritmasının optimize edilmiş ve daha hızlı çalışan bir versiyonudur. Bu algoritma, düzenlileştirme (regularization) teknikleri ve paralel hesaplama

özellikleri sayesinde yüksek doğruluk oranları ve hızlı eğitim süreleri sağlar (Chen ve Guestrin, 2016). XGBoost, özellikle büyük veri setleri ve karmaşık modeller üzerinde etkili sonuçlar verir. Algoritma, her bir adımda yeni bir ağaç ekleyerek hataları azaltır ve modelin genel performansını artırır.

Gradient boosting, bir kayıp fonksiyonunu (loss function) minimize etmek için ardışık modellerin bir araya getirildiği bir süreçtir. Bu süreçte her adımda, önceki modelin hatalarını düzeltmeye odaklanan bir yeni model eklenir. XGBoost, bu süreçteki iyileştirmeyi aşağıdaki formülle gerçekleştirir:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta . f_t(x_i) \tag{23}$$

burada $\hat{y}_i^{(t)}$ i'inci veri noktası için t'inci iterasyonda yapılan tahmin, $\hat{y}_i^{(t-1)}$ Bir önceki iterasyondaki tahmin, η öğrenme oranı, $f_t(x_i)$ t'inci iterasyonda öğrenilen yeni zayıf model (genellikle bir karar ağacı). XGBoost, bu tahmin sürecinde L2 düzenlileştirme (ridge regularization) ile overfitting riskini azaltır ve modelin genelleme kabiliyetini artırır. Düzenlileştirilmiş hedef fonksiyonu şu şekilde tanımlanır:

$$Obj(t) = \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{k=1}^{t} \Omega f(k) \tag{24}$$

Burada $L\left(y_i, \hat{y}_i^{(t)}\right)$ Gerçek değer y_i ile tahmin edilen değer $\hat{y}_i^{(t)}$ arasındaki kayıp fonksiyonu, $\sum_{k=1}^{t} \Omega f(k)$ ağaçların karmaşıklığını azaltan düzenleme terimi.

Boosting algoritmaları, klinik karar destek sistemlerinde (KDS) yaygın olarak kullanılmaktadır. Bu sistemler, hastalıkların teşhisi, tedavi planlaması ve hasta yönetimi gibi konularda sağlık profesyonellerine yardımcı olur. Örneğin, deri kanseri teşhisinde kullanılan bir karar destek sistemi, hastaların cilt lezyonlarını analiz ederek doğru teşhis oranlarını artırabilir (Esteva vd., 2017).

Boosting algoritmaları, sağlıkta karar destek sistemlerinde önemli bir rol oynamaktadır. Bu algoritmalar, zayıf öğrenicileri bir araya getirerek güçlü ve yüksek doğruluk oranlarına sahip modeller oluşturur. Adaboost, Gradient Boosting ve XGBoost gibi çeşitli boosting teknikleri, sağlıkta teşhis ve tedavi süreçlerini iyileştirmede etkili bir şekilde

kullanılmaktadır. Gelecekte, bu algoritmaların daha geniş uygulama alanları bulması ve sağlık hizmetlerinin kalitesini artırması beklenmektedir.

1.2.6. Model Değerlendirmesi

Makine öğrenimi modellerinin sağlık verileri üzerindeki performansını doğru bir şekilde değerlendirebilmek, sağlık alanında geliştirilen uygulamaların güvenilirliği açısından kritik öneme sahiptir. Model değerlendirmesi, geliştirilen algoritmanın genel doğruluğunu, genelleme kabiliyetini ve sağlık uygulamalarındaki geçerliliğini anlamamıza yardımcı olur. Brinkmann vd.. (2017), model değerlendirmesinin, özellikle klinik karar destek sistemlerinde doğru teşhis ve tedavi süreçlerinin belirlenmesi için hayati olduğunu belirtmektedir.

Tablo 1. Karışıklık Matrisi

	Gerçek Pozitif (P)	Gerçek Negatif (N)
Tahmin Pozitif (P')	Doğru Pozitif (TP)	Yanlış Pozitif (FP)
Tahmin Negatif (N')	Yanlış Negatif (FN)	Doğru Negatif (TN)

Performans Ölçütleri

Makine öğrenimi modellerinin performansını doğru bir şekilde değerlendirebilmek için kullanılan çeşitli başarı ölçütleri, modelin doğruluğunu, hassasiyetini ve genelleme kabiliyetini ölçmek açısından büyük önem taşır. Sağlık alanında bu başarı ölçütleri, hastalık teşhisi, tedavi planlaması ve hasta yönetimi gibi kritik karar süreçlerinde modelin ne kadar güvenilir olduğunu anlamak için gereklidir. Powers (2011), başarı ölçütlerinin, model değerlendirmesindeki rolünü ve sağlık uygulamalarındaki önemini vurgulamaktadır. En yaygın kullanılan başarı ölçütlerinden biri doğruluk (accuracy) oranıdır. Doğruluk, modelin tüm tahminlerinin ne kadarının doğru olduğunu gösterir. Ancak, doğruluk oranı, özellikle sınıflar arasında dengesizlik olduğunda tek başına yeterli bir ölçüt olmayabilir. Japkowicz ve Stephen (2002), dengesiz veri setlerinde doğruluğun yanıltıcı olabileceğini ve bu durumda kesinlik ve geri çağırma gibi ölçütlerin daha uygun olacağını belirtmiştir. Doğruluk oranı şu şekilde hesaplanır:

$$Doğruluk(\text{accuracy}) = \frac{\left(Doğru\ Pozitif\ (TP) + Doğru\ Negatif\ (TN)\right)}{Toplam\ Örneklem\ Sayısı\ (N)} \quad (25)$$

Bunun yanı sıra, özellikle dengesiz veri setlerinde kesinlik ve geri çağırma gibi ölçütler modelin daha detaylı bir değerlendirmesini sağlar. Kesinlik, modelin pozitif olarak sınıflandırdığı örneklerin gerçekten pozitif olma oranını ifade ederken, geri çağırma, modelin pozitif sınıfı ne kadar iyi yakaladığını gösterir. Bu iki ölçütün harmonik ortalaması olan F1 skoru, dengesiz veri setlerinde dengeli bir performans ölçütü olarak kullanılır:

$$Kesinlik\ (Precision) = \frac{\left(Doğru\ Pozitif\ (TP)\right)}{Doğru\ Pozitif\ (TP) + Yanlış\ Pozitif\ (FP)} \quad (26)$$

$$Geri\ Çağırma\ (Recall) = \frac{\left(Doğru\ Pozitif\ (TP)\right)}{Doğru\ Pozitif\ (TP) + Yanlış\ Negatif\ (FN)} \quad (27)$$

$$F1\ Skoru = 2.\frac{Kesinlik.Geri\ Çağırma}{Kesinlik + Geri\ Çağırma} \quad (28)$$

Klinik karar destek sistemleri gibi uygulamalarda, modelin güvenilirliğini ve genel başarısını değerlendirmek için çapraz doğrulama (cross-validation) teknikleri de kullanılır. Çapraz doğrulama, veri setinin çeşitli alt gruplara ayrılarak modelin her alt grup üzerinde ayrı ayrı eğitilmesi ve test edilmesi işlemidir. Bu yöntem, modelin aşırı uyum (overfitting) yapmasını önlemeye ve genelleme kabiliyetini artırmaya yardımcı olur. 3 katlı çapraz doğrulama (3-fold cross-validation) gibi yöntemler, veri seti üzerindeki rastgele etkileri dengeleyerek daha güvenilir sonuçlar elde edilmesini sağlar.

Çapraz Doğrulama

Bu yöntem, modelin yeni, görülmemiş veriler üzerinde nasıl performans göstereceğini tahmin etmek için kritik bir araçtır. Sağlık alanında, modelin genelleme kabiliyeti, hasta verilerinin doğru ve güvenilir bir şekilde işlenmesi açısından büyük önem taşır. Bu

bağlamda, çapraz doğrulama, klinik karar destek sistemlerinde modellerin doğruluğunu ve güvenilirliğini sağlamak için hayati bir rol oynar.

Çapraz doğrulamanın temel prensibi, veri setinin belirli bir kısmının model eğitimi için, geri kalanının ise modelin test edilmesi için kullanılmasıdır. Bu yöntem, veri setinin farklı alt kümelere ayrılması ve her alt kümenin bir kez test kümesi olarak kullanılması suretiyle gerçekleştirilir. Stone (1974) tarafından önerilen bu yöntem, modelin çeşitli veri alt kümeleri üzerindeki performansını değerlendirmeyi mümkün kılar ve bu sayede modelin genelleme kabiliyeti hakkında daha güvenilir sonuçlar elde edilir. Çapraz doğrulamanın temel formülü şu şekildedir:

$$Genel\ Performans = \frac{1}{k}\sum_{i=1}^{k} M_i \tag{29}$$

Burada k toplam kat sayısını, M_i her bir katmanda elde edilen performans ölçütünü (örneğin doğruluk, F1 skoru) ifade eder.

En yaygın kullanılan çapraz doğrulama yöntemi k-katlı çapraz doğrulama (k-fold cross-validation) olarak bilinir. Bu yöntemde, veri seti k alt kümeye (fold) bölünür ve model, $k-1$ katmanında eğitilirken kalan 1 katman test için kullanılır. Bu işlem k defa tekrarlanır ve her defasında farklı bir katman test kümesi olarak seçilir. Sonuç olarak, k farklı performans ölçümü elde edilir ve bu ölçümler ortalanarak modelin genel performansı hesaplanır. Geisser (1975) ve Browne (2000), bu yöntemin modelin performansını değerlendirmede üstün olduğunu ve özellikle küçük veri setlerinde aşırı uyumu önlediğini belirtmiştir.

Sağlık alanında, veri setleri genellikle sınırlı ve dengesiz olabilir. Bu durum, modelin eğitim sürecinde aşırı uyuma neden olabilir, yani model eğitim verisine çok fazla uyum sağlarken yeni veriler üzerinde düşük performans gösterebilir. Çapraz doğrulama, modelin yalnızca eğitim verisine değil, aynı zamanda test verisine de uyum sağlama yeteneğini ölçer. Böylece, modelin genelleme kabiliyeti artırılır ve gerçek hayattaki veri setlerine uygulanabilirliği sağlanır. Kohavi (1995), k-katlı çapraz doğrulamanın dengesiz veri setlerinde model performansını değerlendirmek için en etkili yöntemlerden biri olduğunu göstermiştir.

1.3. Karar Destek Sistemleri

İşletmeler, faaliyetleri boyunca karşılaştıkları basit, komplike, olağan ve olağandışı sorunların çözümü için çeşitli kararlar vermek zorundadır. İşletme yöneticileri, karar verme sürecinde çevresel faktörler, psikolojik unsurların etkisi altında kalarak doğru karar verememe endişesi yaşayabilmektedirler. Bu nedenle, karar verme süreci yöneticiler için oldukça zorlu bir süreç haline gelmektedir. Karar destek sistemleri (KDS), bu zorlu süreci kolaylaştırmak amacıyla oluşturulmuş teknoloji destekli altyapılardır. Ancak bu yapıyı açıklamadan önce, ‘karar’, ‘karar verme süreci’ ve ‘yönetim’ kavramlarını ele almak gerekmektedir.

Türk Dil Kurumu, "karar" kavramını “bir iş veya sorun hakkında düşünülerek verilen kesin yargı” olarak tanımlamaktadır. Mintzberg (1976) ise kararı, “eylem için gerekli unsurların tanımlanmasıyla başlayıp, eylem için çok özel bir yargının oluşmasıyla biten dinamik faktörler ve faaliyetler serisi” olarak tanımlar (Tekin ve Ehtiyar, 2010). Karar verme süreci, problemlerin çözümüne yönelik çeşitli çözüm alternatiflerinin belirlenmesi, değerlendirilmesi ve en uygun seçeneğin seçilmesi aşamalarını kapsar (Çelik, 2018). Bu süreci farklı bir şekilde ifade edecek olursak, karar verme; karar vericilerin, belirlenen hedefe ulaşmak için alternatifler arasında en uygun olanını seçme sürecidir (Çelikten ve diğerleri, 2019).

Sağlık hizmetlerinin kalitesi ve etkinliği, hastaların doğru teşhis ve tedaviye zamanında ulaşabilmesiyle doğrudan ilişkilidir. Bu noktada, karar destek sistemleri (KDS), sağlık profesyonellerine hastalıkların teşhisi, tedavi planlaması ve hasta yönetimi konularında yardımcı olarak önemli bir rol oynamaktadır. KDS Kullanıcıların karar verme süreçlerini desteklemek için veri, modeller ve kullanıcı dostu arayüzler sunan bilgisayar tabanlı bilgi sistemleridir (Turban ve Aronson, 2001).

Klinik karar destek sistemlerinin kökenleri, 1960'lı yıllara dayanmaktadır. İlk klinik karar destek sistemlerinden biri olan MYCIN, 1970'lerde Stanford Üniversitesi'nde geliştirilmiştir ve enfeksiyon hastalıklarının tedavisinde antibiyotik önerileri sunmuştur (Shortliffe, 1976). Bu dönemde geliştirilen sistemler, kural tabanlı olup, belirli klinik kurallara dayanarak önerilerde bulunmaktaydı. Günümüzde, karar destek sistemleri çeşitli alanlarda kullanılmaktadır. Elektronik sağlık kayıtları (EHR) ile entegre edilen sistemler, hasta verilerini analiz ederek klinik uyarılar, teşhis önerileri ve tedavi planları sunar (Bates ve Gawande, 2003). Örneğin, sepsis teşhisinde kullanılan karar destek sistemleri, hastanın

vital bulgularını ve laboratuvar sonuçlarını analiz ederek erken teşhis ve tedavi sağlar (Jones vd., 2016).

Makine öğrenimi ve yapay zekâ, sağlıkta karar destek sistemlerinin etkinliğini artırmak için yaygın olarak kullanılmaktadır. Bu teknolojiler, büyük veri setlerini analiz ederek karmaşık kalıpları tanımlar ve klinik karar verme sürecini destekler (Esteva vd., 2019).

2. YAPILAN ÇALIŞMALAR

Bu çalışma, makine öğrenimi algoritmalarının triaj sürecinde karar destek sistemlerine entegrasyonunu inceleyen retrospektif bir araştırma olarak tasarlanmıştır. Çalışmanın temel amacı, hasta verilerini kullanarak, hastaların yaş, cinsiyet ve şikayetlerine göre hangi sağlık bölümüne yönlendirilmesi gerektiğini tahmin eden bir model geliştirmektir. Araştırma, tek bir sağlık kurumu olan Özel Türkiye Hastanesi'nden elde edilen ve anonimleştirilmiş hasta kayıtlarına dayanmaktadır.

Araştırma, geniş bir veri seti üzerinde gerçekleştirilmiş olup, verilerin elektronik sağlık kayıtlarından anonimleştirilerek toplanması sayesinde geniş bir hasta popülasyonunu temsil etme özelliğine sahiptir. Veri seti, hastaların demografik bilgilerini, şikayetlerini, başvurdukları bölümleri ve tanı kodlarını içermektedir. Bu veriler, makine öğrenimi algoritmalarıyla analiz edilerek, triaj kararlarının otomatikleştirilmesi ve sağlık profesyonellerine destek sağlanması hedeflenmiştir.

Araştırmanın tasarımı, farklı makine öğrenimi algoritmalarının karşılaştırmalı performans analizine dayanmakta olup, elde edilen sonuçlara göre en uygun algoritmanın seçilmesi planlanmıştır. Bu bağlamda, çalışmanın nihai hedefi, triaj sürecinde kullanılabilecek etkili bir karar destek sistemi geliştirmektir.

Tüm işlemler, Intel® Core (TM) i5-9500H CPU 2.70 GHz işlemcili, 16 GB RAM özellikli bir bilgisayar ortamında ve Python (3.11) yazılımı kullanılarak gerçekleştirilmiştir.

2.1. Araştırma Tasarımı

Bu çalışmada kullanılan veri seti, Özel Türkiye Hastanesi'nden elde edilen yaklaşık bir milyon hasta kaydından oluşmaktadır. Veri seti, her bir hastayı benzersiz bir şekilde tanımlayan PatientID, hastaların şikayetlerini detaylandıran EpicrisisReports, yaş bilgilerini içeren Age, cinsiyet bilgilerini belirten Gender, hastaların başvurduğu bölümleri gösteren Department, tanı kodlarını içeren ICDCode ve tanı tanımlarını içeren ICDDefinition sütunlarını içermektedir. Toplam 1.030.780 kayıttan, düşük sayıda kayda sahip “Çocuk Gastroenteroloji”, “Çocuk Romatoloji” ve “Tıbbi Biyokimya” bölümleri çıkarılarak, analiz için 1.030.652 kayıt kullanılmıştır.

Veri işleme süreci, eksik verilerin giderilmesi, veri tiplerinin dönüştürülmesi ve metin verilerinin sayısal formata dönüştürülmesini kapsamaktadır. Şikayetlerin metinsel ifadeleri, doğal dil işleme (NLP) teknikleriyle işlenmiş ve model girişleri için uygun formatta hazırlanmıştır.

Araştırmada, Naive Bayes, Destek Vektör Makineleri (SVM), Yapay Sinir Ağları (ANN), Rastgele Orman, AdaBoost, Gradient Boosting, CatBoost, LightGBM ve XGBoost gibi makine öğrenimi algoritmaları kullanılmıştır. Her bir modelin performansı, doğruluk metriği değerlendirilmektedir. Modellerin eğitimi ve değerlendirilmesi sırasında, veri seti eğitim ve test olarak ikiye ayrılmış, çapraz doğrulama yöntemi kullanılarak model performansları test edilmiştir.

Seçilen en iyi model, Python programlama dili ve Streamlit kütüphanesi kullanılarak bir uygulamaya dönüştürülmüştür. Bu uygulama, sağlık profesyonellerinin triaj kararlarını daha etkili ve hızlı bir şekilde verebilmelerini sağlamak için kullanıcı dostu bir arayüz sunmaktadır.

2.2. Veri İşleme ve Hazırlık Süreçleri

Bu çalışmada, kullanılan veri seti bir SQLite veritabanından alınmış olup, çeşitli hasta verilerini içermektedir. Veriler, hasta şikayetlerini PatientID, (EpicrisisReports), yaş, cinsiyet, ilgili departman, ICD kodları gibi tıbbi bilgileri kapsamaktadır. Veri işleme ve hazırlık süreci, modelleme aşamasına geçmeden önce verinin temizlenmesi, dönüştürülmesi ve analize hazır hale getirilmesi için kritik öneme sahiptir. Bu süreçte izlenen adımlar aşağıda ayrıntılı olarak açıklanmıştır:

Tablo 2. Veriler

EpicrisisReports	Age	Gender	Department
Şikayeti:Ateş,boğaz ağrısı,	15	ERKEK	Çocuk Sağlığı ve Hastalıkları
Şikayeti:Yakın da zorlanma Hikayesi:1 hafta Fizik Muayene:To:16/18 Gk:Sağ/sol 1,0 Bio.sağ/sol normal F:Sağ/sol normal Tedavi:Grv:ss +0,75 yakın	44	KADIN	Göz Hastalıları
Şikayeti: Halsizlik ve kolay yorulma ile el ayak şişmesi, nefes almada zorlanma ve yutkunma güçlüğü. Boyunda önde hafif şişlik hissediyor. Ağrı tanımlamıyor.	53	KADIN	İç Hastalıkları
Şikayeti:KAŞINTI SIRTTA YANMA Hikayesi:3-4 AYDIR OLAN SIRT AĞRISI KAŞINTI ŞİKAYETİ VAR KİLO KAYBI VAR	71	ERKEK	Gastroenteroloji
Şikayeti:SOL ALT KADRANDA AĞRI SOL ALT KADRANDA HASSAS	62	ERKEK	Genel Cerrahi

Veriler, SQLite veritabanından pandas kütüphanesi kullanılarak read_sql fonksiyonu ile çekilmiştir. Bu aşamada, SQL sorgusu ile veritabanındaki ilgili tablo (patient_records) pandas DataFrame yapısına dönüştürülmüştür. Veritabanından veri çekilirken, potansiyel olarak eksik ya da hatalı kayıtlar göz önünde bulundurulmuş ve gerekli filtrelemeler yapılmıştır.

Veri setinde bulunan eksik ya da hatalı veriler, analiz sürecini olumsuz etkileyebileceğinden, bu verilerle ilgili bir inceleme yapılmıştır. Eksik değerler, eğer küçük bir yüzdede ise, veri setinden çıkarılmış ya da uygun istatistiksel yöntemlerle doldurulmuştur. Hatalı veriler ise analiz sürecine dahil edilmeden önce düzeltici işlemlerden geçirilmiştir.

Veri setinde yer alan ‘Gender (Cinsiyet)’ değişkeni kategorik bir veri olup, modelin bu veriyi işleyebilmesi için sayısal değerlere dönüştürülmüştür. Bu dönüşüm, Python'da category veri tipi kullanılarak yapılmıştır. Kategorik veriler için LabelEncoder kullanılarak her bir kategoriye sayısal bir kod atanmıştır. Bu sayede, modelin bu değişkeni doğrudan kullanabilmesi sağlanmıştır.

Veri setinde bulunan EpicrisisReports (Epikriz Raporları) sütunu, hasta şikayetlerini içeren metin verisidir. Bu tür verilerin modellenebilmesi için sayısal bir forma dönüştürülmesi gerekmektedir. Bu amaçla, metin verileri üzerinde TfidfVectorizer kullanılmıştır. TF-IDF (Term Frequency-Inverse Document Frequency), metin içerisindeki kelimelerin önem derecesini belirleyen bir sayısal vektör oluşturma yöntemidir. Bu işlem, metin verilerinin model tarafından işlenebilir hale getirilmesi için kullanılmış ve 1000 özelliğe kadar olan TF-IDF vektörleri oluşturulmuştur.

TF-IDF ile dönüştürülen metin verileri, yaş ve cinsiyet gibi numerik ve kategorik özelliklerle birleştirilmiştir. Bu süreçte, hstack fonksiyonu kullanılarak TF-IDF matrisleri ve diğer özellikler yatay olarak birleştirilmiş, modelin giriş verisi olarak kullanılacak nihai özellik matrisi oluşturulmuştur.

Veri setinde, bazı sınıfların diğerlerine göre daha az sayıda gözlem içermesi nedeniyle sınıf dengesizliği sorunu tespit edilmiştir. Bu durumu gidermek için, SMOTE (Synthetic Minority Over-sampling Technique) yöntemi uygulanmıştır. SMOTE, azınlık sınıflarına ait sentetik örnekler oluşturarak veri setindeki dengesizliği azaltmakta ve modelin azınlık sınıflarını daha iyi öğrenmesini sağlamaktadır. Bu yöntem, eğitim verisi üzerinde uygulanmış ve dengesizliğin giderilmesi amaçlanmıştır.

2.3. Modelleme Yaklaşımı

Bu çalışmada, çok sınıflı bir sınıflandırma problemi üzerine odaklanılmış olup, farklı makine öğrenimi modelleri incelenmiştir. Modelleme sürecinde, en iyi performansı gösteren modelin seçilmesi amacıyla çeşitli algoritmalar kullanılmış ve bunlar arasından en uygun olanı belirlenmiştir. Bu süreçte değerlendirilen modeller arasında Naive Bayes, Destek Vektör Makineleri (SVM), Yapay Sinir Ağları (ANN), AdaBoost, Gradient Boosting, CatBoost, LightGBM ve XGBoost yer almaktadır. Her bir modelin avantajları ve dezavantajları göz önünde bulundurularak kapsamlı bir karşılaştırma yapılmıştır.

2.4. Modelin Değerlendirilmesi

Modelin performansı, test veri seti üzerinde çeşitli metrikler kullanılarak değerlendirilmiştir. Bu değerlendirme süreci, modelin genel doğruluğunu ve sınıf bazında performansını analiz etmek için önemli bir aşamadır. Modelin değerlendirilmesinde kullanılan başlıca metrikler aşağıda ayrıntılı olarak açıklanmıştır:

Modelin genel doğruluğu, doğru tahmin edilen örneklerin toplam örnek sayısına oranı olarak hesaplanmıştır. Doğruluk, sınıflandırma problemlerinde yaygın olarak kullanılan bir metrik olmakla birlikte, sınıf dengesizliği olan veri setlerinde yanıltıcı olabilir. Bu nedenle, doğruluk metriği, modelin performansını anlamak için tek başına yeterli görülmemiş ve diğer metriklerle desteklenmiştir. Modelin sınıf bazında performansı, doğruluk metriği ile değerlendirilmiştir.

Modelin tahminlerinin ne kadar doğru olduğunu ve hangi sınıflarda hatalar yaptığını görselleştirmek amacıyla bir karışıklık matrisi oluşturulmuştur. Karışıklık matrisi, her bir sınıfın doğru ve yanlış tahminlerinin sayısını gösteren bir tablodur. Bu tablo, modelin güçlü ve zayıf yönlerini sınıf bazında analiz etmeyi ve modelde yapılacak iyileştirmeler için yol göstermeyi amaçlamaktadır.

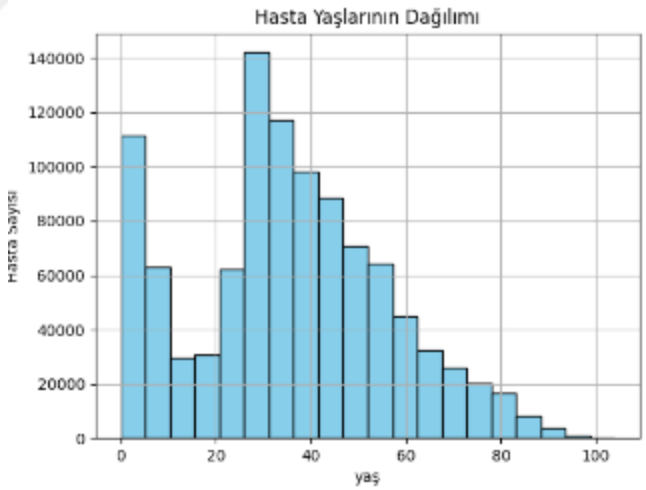
2.5. Veri Analizi

Çalışmada kullanılan veri kümesinin demografik ve bölüm bazlı dağılımına ilişkin analizler sunulmuştur. Elde edilen veriler, çalışmanın temelini oluşturan hasta bilgilerini içermekte olup, yaş, cinsiyet ve bölüm gibi temel değişkenler üzerinden değerlendirilmiştir.

Hastaların yaş dağılımını anlamak, veri kümesinin demografik yapısını belirlemek açısından önemlidir. Bu analizde, veri kümesindeki hastaların yaş ortalaması, en küçük ve en büyük yaş değerleri hesaplanmıştır. Ayrıca, yaş dağılımını görselleştirmek amacıyla bir histogram oluşturulmuştur.

Tablo 3. Yaş değerleri

İstatistik	**Değer**
En küçük yaş	0
En büyük yaş	104
Yaş ortalaması	35

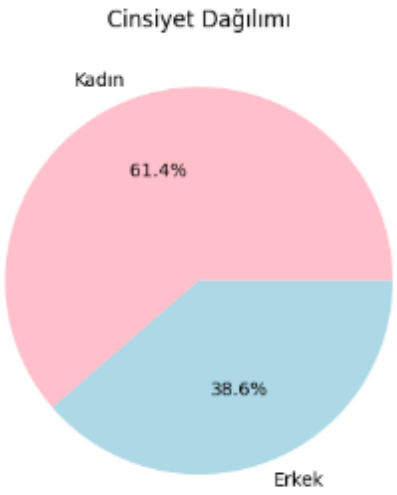


Şekil 5. Yaş dağılımı (Bu histogramın Python ile nasıl oluşturulduğu "EK-2" bölümünde detaylı olarak açıklanmıştır.)

Cinsiyet dağılımı, veri kümesindeki kadın ve erkek hastaların oranlarını belirlemek açısından analiz edilmiştir. Bu analiz, veri kümesinde cinsiyetler arasındaki dengenin değerlendirilmesine olanak tanımaktadır.

Tablo 4. Cinsiyet yüzdeleri

Cinsiyet	**Yüzde (%)**
Kadın	51.4
Erkek	38.6

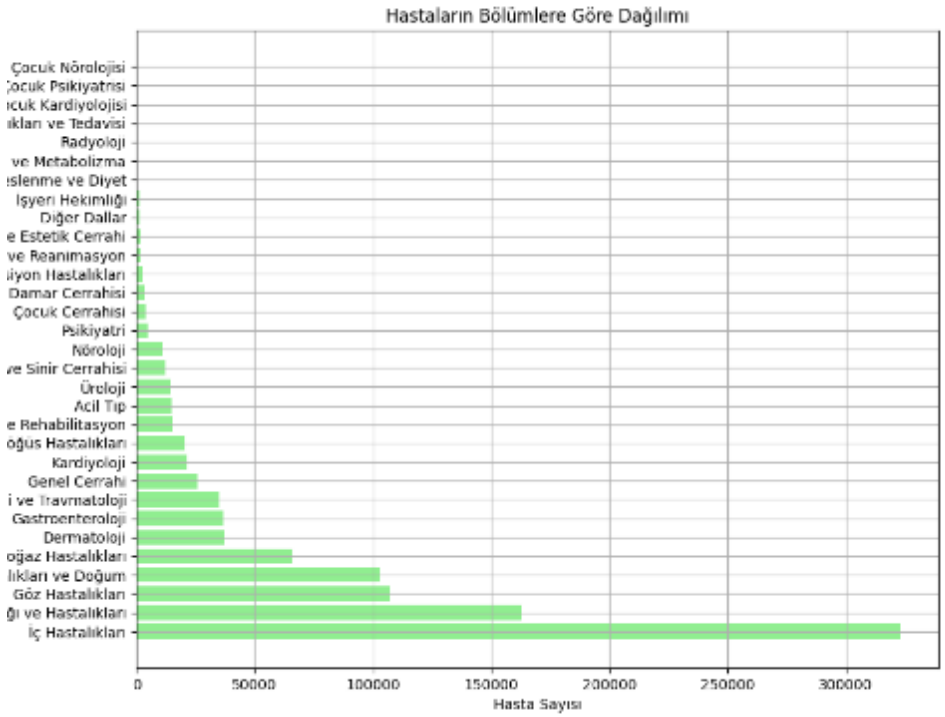


Şekil 6. Cinsiyet Dağılımı Pasta Grafiği (Bu histogramın Python ile nasıl oluşturulduğu " EK-2" bölümünde detaylı olarak açıklanmıştır.)

Bu bölümde, hastaların farklı tıbbi bölümlere göre dağılımı analiz edilmiştir. Bu analiz, veri kümesindeki hastaların hangi bölümlerde yoğunlaştığını ve bu yoğunluğun hastane hizmetlerine olan etkisini ortaya koymaktadır.

Tablo 5. Bölüme göre hasta dağılımı (İlk 5)

Bölüm	**Hasta sayısı**
İç Hastalıkları	322.984
Çocuk Sağlığı ve Hastalıkları	162.648
Göz Hastalıkları	107.414
Kadın Hastalıkları ve Doğum	102.715
Kulak-Burun-Boğaz Hastalıkları	66.213



Şekil 7. Bölümlere Göre Hasta Sayısı (Bu histogramın Python ile nasıl oluşturulduğu " EK-2" bölümünde detaylı olarak açıklanmıştır.)

2.6. Kullanılan Yazılım ve Araçlar

Araştırmanın tüm evreleri Python programlama dili kullanılarak gerçekleştirilmiştir. Veri işleme ve analiz süreçleri için pandas, numpy gibi kütüphaneler; modelleme için XGBoost; metin işleme için scikit-learn kütüphanesinin TF-IDF vectorizer'ı tercih edilmiştir. Sonuçların interaktif bir şekilde sunulması ve modelin uygulanabilirliğinin artırılması amacıyla Streamlit kütüphanesi ile bir web uygulaması geliştirilmiştir. matplotlib.pyplot kütüphanesi ile görselleştirilmesi yapılmıştır.

3. BULGU VE TARTIŞMA

Bu çalışmada, makine öğrenimi algoritmalarının sağlık hizmetlerinde triaj süreçlerine entegrasyonunu sağlamak amacıyla, hasta verilerini kullanarak hangi sağlık bölümüne yönlendirilmesi gerektiğini tahmin eden bir model geliştirilmiştir. Araştırma kapsamında, Naive Bayes, Destek Vektör Makineleri (SVM), Yapay Sinir Ağları (ANN), Rastgele Orman, AdaBoost, Gradient Boosting, CatBoost, LightGBM ve XGBoost gibi farklı makine öğrenimi algoritmaları değerlendirilmiş ve performansları karşılaştırılmıştır. Elde edilen sonuçlar, bu modellerin triaj sürecine nasıl entegre edilebileceği konusunda önemli bilgiler sunmuştur.

Tablo 6. Makine öğrenimi algoritmaları değerlendirilmesi

Makine öğrenimi algoritması	**Doğruluğu**
Naive Bayes	%89
LightGBM	%90
Destek Vektör Makineleri	%91
Yapay Sinir Ağları (ANN)	%91
Rastgele Orman	%94
AdaBoost	%93
Gradient Boosting	%95
CatBoost	%94
XGBoost	%97

Özellikle XGBoost algoritması, %97 doğruluk oranı ile diğer tüm modellerden üstün performans göstermiştir. Bu sonuç, XGBoost'un sağlık verilerini etkili bir şekilde işleyerek doğru sınıflandırmalar yapabilme kapasitesini ortaya koymaktadır. XGBoost'un yüksek

doğruluk oranı, modelin sağlık hizmetlerinde triaj kararlarını desteklemede ne kadar etkili olabileceğini göstermektedir. Ayrıca, bu algoritmanın performansı, triaj süreçlerinin otomatikleştirilmesi ve sağlık profesyonellerine destek sağlanması açısından büyük bir potansiyele sahip olduğunu doğrulamaktadır.

Ancak, modelin düşük performans gösterdiği alanlarda yapılan hataların nedenleri üzerine yapılan analizler, veri setindeki dengesizliklerin ve belirli hastalık durumlarının nadir olmasının etkili olabileceğini göstermektedir. Bu tür durumlar, modelin öğrenme sürecini olumsuz etkileyebilir ve bazı hastalık sınıflarını yanlış sınıflandırmasına neden olabilir.

Ayrıca, modelin uygulanabilirliği ve ölçeklenebilirliği açısından, gerçek dünya verileriyle yapılan testler ve bu testler sırasında elde edilen geribildirimler, modelin daha da iyileştirilmesi için önemli fırsatlar sunmaktadır. Özellikle, veri setinin sürekli olarak güncellenmesi ve genişletilmesi, modelin farklı popülasyonlar üzerindeki genel geçerliliğini artıracak ve triaj süreçlerindeki etkinliğini daha da iyileştirecektir.

Çalışmamız, yapay zekâ ve makine öğreniminin acil servis yönlendirmeleri ve hastane içi triaj gibi kritik sağlık hizmetlerinde potansiyel olarak devrim yaratabilecek araçlar olarak kullanılabilirliğini göstermektedir. Gelecekte yapılacak çalışmalar, modelin kapsamını genişleterek daha fazla hastalık durumunu ve farklı coğrafi bölgelerden gelen verileri içermelidir. Bu, modelin evrenselliğini ve farklı hasta demografilerine uygulanabilirliğini test etmek için önemlidir.

Şekil 8. Uygulama ekranı (Bu uygulama Python ile nasıl oluşturulduğu " EK-3" bölümünde detaylı olarak açıklanmıştır)

4. SONUÇ VE ÖNERİLER

Bu çalışma, XGBoost algoritması kullanılarak geliştirilen bir tahmin modelinin, hastane bölüm yönlendirmelerinde etkili bir araç olarak kullanılabileceğini göstermiştir. Model, test seti üzerinde %97 gibi yüksek bir doğruluk oranı elde etmiş, bu da makine öğrenimi yöntemlerinin sağlık sektöründe karar destek sistemleri olarak potansiyel taşıdığını ortaya koymuştur. Özellikle, veri setindeki çeşitlilik ve kapsamlı veri ön işleme adımları, modelin güvenilirliğini ve genel geçerliliğini artırmıştır. Bununla birlikte, modelin performansında gözlemlenen bazı sınırlılıklar, özellikle nadir görülen hastalık durumları ve veri dengesizlikleri nedeniyle bazı bölüm tahminlerinde hataların yaşanması, modelin iyileştirilmesi gereken alanları belirlemiştir:

- Elektronik Sağlık Kayıt Sistemlerinin Güncellenmesi: Elektronik Sağlık Kayıtlarının verileri Karar Destek Sistemlerine entegre edip senkron şekilde çalışılması önerilir.
- Eğitim Veri Setinin Genişletilmesi: Modelin daha geniş hastalık spektrumunu kapsayabilmesi ve nadir durumlarla daha iyi başa çıkabilmesi için veri setinin çeşitlendirilmesi ve genişletilmesi önerilir.
- Sürekli Öğrenme ve Güncelleme: Modelin zaman içinde değişen sağlık trendlerine ve yeni tıbbi bilgilere uyum sağlayabilmesi için sürekli öğrenme mekanizmalarının senkronizasyonunu sağlanmalıdır.
- Karar Destek Sistemleri Teorisi: Bu çalışma, karar destek sistemlerinin geliştirilmesinde kullanılan yöntemlerin teorik temellerine katkıda bulunmuş ve XGBoost'un sağlık alanındaki uygulanabilirliğini somut örneklerle desteklemiştir.
- Makine Öğrenimi Modellerinin Denetlemesi: Çalışma, model değerlendirme süreçlerinin, özellikle sağlık sektöründe, nasıl yürütülmesi gerektiğine dair metodolojik bir çerçeve sunar.
    - Coğrafi Çeşitliliğin İncelenmesi: Modelin farklı coğrafi bölgelerdeki hastane verileri üzerinde test edilmesi, modelin evrenselliği ve adaptasyon kapasitesi hakkında daha fazla bilgi sağlayabilir.

5. KAYNAKLAR

Özata, M., & Aslan, Ş. (2004). Klinik Karar Destek Sistemleri ve Örnek Uygulamalar. Kocatepe Tıp Dergisi, 5(2), 11-17.

Özata, M., & Aslan, Ş. (2004). Karar verme süreci ve karar verme teknikleri. Selçuk Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 11, 487-506.

Sucu, M. (2021). Karar Destek Sistemleri ve İş Zekâsı Uygulamalarının İşletmeler Açısından Önemi: Bir Literatür Araştırması. Pamukkale Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 44, 261-283.

Veranyurt, Ü., Deveci, A., Esen, M. F., & Veranyurt, O. (2020). Makine öğrenmesi teknikleriyle hastalık sınıflandırması: Random forest, k-nearest neighbour ve adaboost algoritmaları uygulaması. Uluslararası Sağlık Yönetimi vae Stratejileri Araştırma Dergisi, 6(2), 275-286.

Alanazi, A. (2022). Using machine learning for healthcare challenges and opportunities. Informatics in Medicine Unlocked, 30, 100924. https://doi.org/10.1016/j.imu.2022.100924

Alonso, D. H., Wernick, M. N., Yang, Y., Germano, G., Berman, D. S., & Slmoka, P. (2018). Prediction of cardiac death after adenosine myocardial perfusion SPECT based on machine learning. Journal of Nuclear Cardiology. https://doi.org/10.1007/s12350-017-0924-x

Alonso, S. G., de la Torre Díez, I., Zapiraín, B. G., & Rodrigues, J. J. P. C. (2018). A systematic review of techniques and sources of big data in the healthcare sector. Journal of Medical Systems, 42(12), 214. https://doi.org/10.1007/s10916-018-1081-7

American Diabetes Association. (2014). Standards of medical care in diabetes—2014. Diabetes Care, 37(Supplement_1), S14-S80.

Arshad, M., Rasool, M., & Ahmad, M. (2003). Anderson Darling and modified Anderson Darling tests for generalized Pareto distribution. Pakistan Journal of Applied Sciences, 3(2), 85-88.

Bates, D. W., & Gawande, A. A. (2003). Improving safety with information technology. The New England Journal of Medicine, 348(25), 2526-2534. https://doi.org/10.1056/NEJMsa020847

Berry, J. G., Ziniel, S. I., Freeman, L., Porter, S., O'Toole, J. K., & Gay, J. C. (2018). Hospital readmission and parent perceptions of their child's hospital discharge. JAMA Pediatrics, 172(2), 118-126. https://doi.org/10.1001/jamapediatrics.2017.3963

Campbell, E. M., Sittig, D. F., Ash, J. S., Guappone, K. P., & Dykstra, R. H. (2006). Types of unintended consequences related to computerized provider order entry. Journal of the American Medical Informatics Association, 13(5), 547-556. https://doi.org/10.1197/jamia.M2042

Celebi, M. E., & Inal, M. E. (2019). Ethics in artificial intelligence: An overview. AI Matters, 5(2), 10-14. https://doi.org/10.1145/3334306.3334312

Char, D. S., Shah, N. H., & Magnus, D. (2018). Implementing machine learning in health care—addressing ethical challenges. The New England Journal of Medicine, 378(11), 981-983. https://doi.org/10.1056/NEJMp1714229

Cichosz, S. L., Johansen, M. D., & Hejlesen, O. K. (2015). Toward big data analytics: Review of predictive models in management of diabetes and its complications. Journal of Diabetes Science and Technology, 10(1), 27-34. https://doi.org/10.1177/1932296815608728

Cichosz, S. L., & Hejlesen, O. (2022). Classification of Gastroparesis from Glycemic Variability in Type 1 Diabetes: A Proof-of-Concept Study. Journal of Diabetes Science and Technology, 16(5), 1190-1195. https://doi.org/10.1177/19322968211015206

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639), 115-118. https://doi.org/10.1038/nature21056

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1), 119-139. https://doi.org/10.1006/jcss.1997.1504

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29(5), 1189-1232. https://doi.org/10.1214/aos/1013203451

Garg, A. X., Adhikari, N. K., McDonald, H., Rosas-Arellano, M. P., Devereaux, P. J., Beyene, J., ... & Haynes, R. B. (2005). Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: A systematic review. JAMA, 293(10), 1223-1238. https://doi.org/10.1001/jama.293.10.1223

Grant, R. W., & Kirkman, M. S. (2015). Trends in the evidence level for the American Diabetes Association's “Standards of Medical Care in Diabetes” from 2005 to 2014. Diabetes Care, 38(1), 6-8.

Haenssle, H. A., Fink, C., Schneiderbauer, R., Toberer, F., Buhl, T., Blum, A., ... & Reader, J. (2018). Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists. Annals of Oncology, 29(8), 1836-1842. https://doi.org/10.1093/annonc/mdy166

Habehh, H., & Gohel, S. (2021). Machine learning in healthcare. Current Genomics, 22(4), 291-300.

Hersh, W. (1999). A stimulus to define informatics and health information technology. Journal of the American Medical Informatics Association, 6(1), 73-77. https://doi.org/10.1136/jamia.1999.0060073

Hersh, R. H. (1999). Generating Ideals and Transforming Lives: A Contemporary Case for the Residential Liberal Arts College. Daedalus, 128(1), 173-194.

Javaid, M., Haleem, A., Singh, R. P., Suman, R., & Rab, S. (2022). Significance of machine learning in healthcare: Features, pillars and applications. International Journal of Intelligent Networks, 3, 58-73. https://doi.org/10.1016/j.ijin.2022.06.003

Jones, S. S., Thomas, A., Evans, R. S., Welch, S. J., Haug, P. J., & Snow, G. L. (2016). Secondary use of EHR data: Spontaneous trigger design for early sepsis detection. Journal of Biomedical Informatics, 52, 92-102. https://doi.org/10.1016/j.jbi.2014.04.013

Kawamoto, K., Houlihan, C. A., Balas, E. A., & Lobach, D. F. (2005). Improving clinical practice using clinical decision support systems: A systematic review of trials to identify features critical to success. BMJ, 330(7494), 765. https://doi.org/10.1136/bmj.38398.500764.8F

Li, W., Wang, Z., Zhu, Y., Li, Z., Chen, X., & Hu, X. (2020). Predictive models of hospital readmission risk: A systematic review. Journal of the American Medical Informatics Association, 27(4), 613-622. https://doi.org/10.1093/jamia/ocz228

Mehta, N., Harish, V., Tavarez, T., & Monash, R. (2021). Development and implementation of artificial intelligence-based decision support systems. BMJ Health & Care Informatics, 28(1), e100251. https://doi.org/10.1136/bmjhci-2021-100251

Moreno, R. P., Rhodes, A., Donchin, Y., & Lippert, A. (2020). Clinical decision support systems in critical care. Intensive Care Medicine, 46(9), 1707-1717. https://doi.org/10.1007/s00134-020-06156-3

Narula, J., Chandrashekhar, Y., & Ahmadi, A. (2017). Artificial intelligence and machine learning in cardiovascular imaging. JACC: Cardiovascular Imaging, 10(10 Pt B), 1308-1311. https://doi.org/10.1016/j.jcmg.2017.09.007

Shortliffe, E. H. (1976). Computer-Based Medical Consultations: MYCIN. Elsevier.

Sutton, R. T., Pincock, D., Baumgart, D. C., Sadowski, D. C., Fedorak, R. N., & Kroeker, K. I. (2019). An overview of clinical decision support systems: Benefits, risks, and strategies for success. NPJ Digital Medicine, 2(1), 1-10. https://doi.org/10.1038/s41746-019-0131-0

Topol, E. J. (2019). High-performance medicine: The convergence of human and artificial intelligence. Nature Medicine, 25(1), 44-56. https://doi.org/10.1038/s41591-018-0300-7

Tran, V. T., Riveros, C., Ravaud, P., & Perrodeau, E. (2019). Patients' perspectives on how to improve the adoption of digital health tools: Qualitative study. Journal of Medical Internet Research, 21(5), e11665. https://doi.org/10.2196/11665

Tran, V. T., Porcher, R., Tran, V. C., & Ravaud, P. (2019). Predicting data saturation in qualitative surveys with mathematical models from ecological research. Journal of Clinical Epidemiology, 105, 28-35. https://doi.org/10.1016/j.jclinepi.2018.03.008

Turban, E., & Aronson, J. E. (2001). Decision Support Systems and Intelligent Systems. Prentice Hall.

6. EKLER

EK-1 MAKİNE ÖĞRENİMİ ALGORİTMALARI

Naive bayes

```
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score, classification_report, precision_recall_fscore_support
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import MultinomialNB
from scipy.sparse import hstack
from sqlalchemy import create_engine
import pickle
# Veritabanından verileri yükleme
engine = create_engine('sqlite:///health_data.db')
data = pd.read_sql('patient_records', engine)
# Verilerin İşlenmesi
data['Gender'] = data['Gender'].astype('category')
label_encoder_department = LabelEncoder()
data['Department'] = label_encoder_department.fit_transform(data['Department'].astype(str))
vectorizer = TfidfVectorizer(max_features=1000)
X_epicrisis = vectorizer.fit_transform(data['EpicrisisReports'])
X_other = data[['Age', 'Gender']].apply(lambda x: x.cat.codes if x.dtype.name == 'category' else x)
X = hstack([X_other, X_epicrisis])
y = data['Department']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Naive Bayes için Parametre Grid
param_grid = {
    'alpha': [0.5, 1.0, 1.5]
}
# Naive Bayes Modeli Tanımlanması
model = MultinomialNB()
# GridSearchCV ile Modelin Eğitimi
grid_search = GridSearchCV(model, param_grid, cv=3, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train)
# En İyi Modelin Seçimi ve Test Verisi Üzerindeki Performansı
best_model = grid_search.best_estimator_
y_pred = best_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
precision, recall, f1, _ = precision_recall_fscore_support(y_test, y_pred, average='weighted')
# Sonuçların Yazdırılması
print(f"Model: Naive Bayes")
print(f"Best Parameters: {grid_search.best_params_}")
print(classification_report(y_test, y_pred, target_names=label_encoder_department.classes_))
# Sonuçları DataFrame olarak kaydetme ve gösterme
results = {
    'Model': 'Naive Bayes',
    'Best Params': grid_search.best_params_,
    'Accuracy': accuracy * 100,
    'Precision': precision * 100,
    'Recall': recall * 100,
    'F1-Score': f1 * 100
}
results_df = pd.DataFrame([results])
print(results_df)
# En iyi modelin, vektörleştiricinin ve label encoder'ın kaydedilmesi
with open('best_model_naive_bayes.pkl', 'wb') as file:
    pickle.dump(best_model, file)
with open('vectorizer.pkl', 'wb') as file:
    pickle.dump(vectorizer, file)
with open('label_encoder_department.pkl', 'wb') as file:
    pickle.dump(label_encoder_department, file)
```

Destek vektör makineleri (SVM)

```
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy score, classification report,
precision_recall_fscore_support
from sklearn.preprocessing import LabelEncoder
from sklearn.svm import SVC
from scipy.sparse import hstack
from sqlalchemy import create_engine
import pickle
# Veritabanından verileri yükleme
engine = create_engine('sqlite:///health_data.db')
data = pd.read_sql('patient_records', engine)
# Verilerin İşlenmesi
data['Gender'] = data['Gender'].astype('category')
label_encoder_department = LabelEncoder()
data['Department'] = label_encoder_department.fit_transform(data['Department'].astype(str))
vectorizer = TfidfVectorizer(max_features=1000)
X_epicrisis = vectorizer.fit_transform(data['EpicrisisReports'])
X_other = data[['Age', 'Gender']].apply(lambda x: x.cat.codes if x.dtype.name == 'category'
else x)
X = hstack([X_other, X_epicrisis])
y = data['Department']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# SVM için Parametre Grid
param_grid = {
    'C': [0.1, 1, 10],
    'gamma': [0.001, 0.01, 0.1],
    'kernel': ['linear', 'poly', 'rbf']
}
# SVM Modeli Tanımlanması
model = SVC(probability=True)

# GridSearchCV ile Modelin Eğitimi
grid_search = GridSearchCV(model, param_grid, cv=3, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train)
# En İyi Modelin Seçimi ve Test Verisi Üzerindeki Performansı
best_model = grid_search.best_estimator_
y_pred = best_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
precision, recall, f1, _ = precision_recall_fscore_support(y_test, y_pred,
average='weighted')
# Sonuçların Yazdırılması
print(f"Model: Support Vector Machines (SVM)")
print(f"Best Parameters: {grid_search.best_params_}")
print(classification_report(y_test, y_pred,
target_names=label_encoder_department.classes_))
# Sonuçları DataFrame olarak kaydetme ve gösterme
results = {
    'Model': 'Support Vector Machines',
    'Best Params': grid_search.best_params_,
    'Accuracy': accuracy * 100,
    'Precision': precision * 100,
    'Recall': recall * 100,
    'F1-Score': f1 * 100
}
results_df = pd.DataFrame([results])
print(results_df)
# En iyi modelin, vektörleştiricinin ve label encoder'ın kaydedilmesi
with open('best_model_svm.pkl', 'wb') as file:
    pickle.dump(best_model, file)
with open('vectorizer.pkl', 'wb') as file:
    pickle.dump(vectorizer, file)
with open('label_encoder_department.pkl', 'wb') as file:
    pickle.dump(label_encoder_department, file)
```

Yapay sinir ağları (YSA)

```
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy score, classification report,
precision_recall_fscore_support
from sklearn.preprocessing import LabelEncoder
from sklearn.neural_network import MLPClassifier
from scipy.sparse import hstack
from sqlalchemy import create_engine
import pickle
# Veritabanından verileri yükleme
engine = create_engine('sqlite:///health_data.db')
data = pd.read_sql('patient_records', engine)
# Verilerin İşlenmesi
data['Gender'] = data['Gender'].astype('category')
label_encoder_department = LabelEncoder()
data['Department'] = label_encoder_department.fit_transform(data['Department'].astype(str))
vectorizer = TfidfVectorizer(max_features=1000)
X_epicrisis = vectorizer.fit_transform(data['EpicrisisReports'])
X_other = data[['Age', 'Gender']].apply(lambda x: x.cat.codes if x.dtype.name == 'category'
else x)
X = hstack([X_other, X_epicrisis])
y = data['Department']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Yapay Sinir Ağları için Parametre Grid
param_grid = {
    'hidden_layer_sizes': [(10,), (20,), (100, 100, 100), (50, 100, 50)],
    'activation': ['relu', 'tanh'],
    'learning_rate_init': [0.001, 0.01]
}
# Yapay Sinir Ağları Modeli Tanımlanması
model = MLPClassifier(max_iter=1000)
# GridSearchCV ile Modelin Eğitimi
grid_search = GridSearchCV(model, param_grid, cv=3, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train)
# En İyi Modelin Seçimi ve Test Verisi Üzerindeki Performansı
best_model = grid_search.best_estimator_
y_pred = best_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
precision, recall, f1, _ = precision_recall_fscore_support(y_test, y_pred,
average='weighted')
# Sonuçların Yazdırılması
print(f"Model: Artificial Neural Networks (ANN)")
print(f"Best Parameters: {grid_search.best_params_}")
print(classification_report(y_test, y_pred,
target_names=label_encoder_department.classes_))
# Sonuçları DataFrame olarak kaydetme ve gösterme
results = {
    'Model': 'Artificial Neural Networks',
    'Best Params': grid_search.best_params_,
    'Accuracy': accuracy * 100,
    'Precision': precision * 100,
    'Recall': recall * 100,
    'F1-Score': f1 * 100
}
results_df = pd.DataFrame([results])
print(results_df)
# En iyi modelin, vektörleştiricinin ve label encoder'ın kaydedilmesi
with open('best_model_ann.pkl', 'wb') as file:
    pickle.dump(best_model, file)
with open('vectorizer.pkl', 'wb') as file:
    pickle.dump(vectorizer, file)
with open('label_encoder_department.pkl', 'wb') as file:
    pickle.dump(label_encoder_department, file)
```

Rastgele orman

```
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score, classification_report,
precision_recall_fscore_support
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from scipy.sparse import hstack
from sqlalchemy import create_engine
import pickle
# Veritabanından verileri yükleme
engine = create_engine('sqlite:///health_data.db')
data = pd.read_sql('patient_records', engine)
# Verilerin İşlenmesi
data['Gender'] = data['Gender'].astype('category')
label_encoder_department = LabelEncoder()
data['Department'] = label_encoder_department.fit_transform(data['Department'].astype(str))
vectorizer = TfidfVectorizer(max_features=1000)
X_epicrisis = vectorizer.fit_transform(data['EpicrisisReports'])
X_other = data[['Age', 'Gender']].apply(lambda x: x.cat.codes if x.dtype.name == 'category'
else x)
X = hstack([X_other, X_epicrisis])
y = data['Department']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Random Forest için Parametre Grid
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'bootstrap': [True, False]
}
# Random Forest Modeli Tanımlanması
model = RandomForestClassifier(random_state=42)

# GridSearchCV ile Modelin Eğitimi
grid_search = GridSearchCV(model, param_grid, cv=3, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train)
# En İyi Modelin Seçimi ve Test Verisi Üzerindeki Performansı
best_model = grid_search.best_estimator_
y_pred = best_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
precision, recall, f1, _ = precision_recall_fscore_support(y_test, y_pred,
average='weighted')
# Sonuçların Yazdırılması
print(f"Model: Random Forest Classifier")
print(f"Best Parameters: {grid_search.best_params_}")
print(classification_report(y_test, y_pred,
target_names=label_encoder_department.classes_))
# Sonuçları DataFrame olarak kaydetme ve gösterme
results = {
    'Model': 'Random Forest',
    'Best Params': grid search.best params ,
    'Accuracy': accuracy * 100,
    'Precision': precision * 100,
    'Recall': recall * 100,
    'F1-Score': f1 * 100
}
results_df = pd.DataFrame([results])
print(results_df)
# En iyi modelin, vektörleştiricinin ve label encoder'ın kaydedilmesi
with open('best_model_rf.pkl', 'wb') as file:
    pickle.dump(best_model, file)
with open('vectorizer.pkl', 'wb') as file:
    pickle.dump(vectorizer, file)
with open('label_encoder_department.pkl', 'wb') as file:
    pickle.dump(label_encoder_department, file)
```

Adaboost (Adaptive Boosting)

```
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score, classification_report,
precision_recall_fscore_support
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier
from scipy.sparse import hstack
from sqlalchemy import create_engine
import pickle
# Veritabanından verileri yükleme
engine = create_engine('sqlite:///health_data.db')
data = pd.read_sql('patient_records', engine)
# Verilerin İşlenmesi
data['Gender'] = data['Gender'].astype('category')
label_encoder_department = LabelEncoder()
data['Department'] = label_encoder_department.fit_transform(data['Department'].astype(str))
vectorizer = TfidfVectorizer(max_features=1000)
X_epicrisis = vectorizer.fit_transform(data['EpicrisisReports'])
X_other = data[['Age', 'Gender']].apply(lambda x: x.cat.codes if x.dtype.name == 'category'
else x)
X = hstack([X_other, X_epicrisis])
y = data['Department']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# AdaBoost için Parametre Grid
param_grid = {
    'n_estimators': [50, 100, 200],
    'learning_rate': [0.01, 0.1, 1.0],
    'base_estimator': [DecisionTreeClassifier(max_depth=1),
DecisionTreeClassifier(max_depth=3)]
}
# AdaBoost Modeli Tanımlanması
model = AdaBoostClassifier()
# GridSearchCV ile Modelin Eğitimi
grid_search = GridSearchCV(model, param_grid, cv=3, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train)
# En İyi Modelin Seçimi ve Test Verisi Üzerindeki Performansı
best_model = grid_search.best_estimator_
y_pred = best_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
precision, recall, f1, _ = precision_recall_fscore_support(y_test, y_pred,
average='weighted')
# Sonuçların Yazdırılması
print(f"Model: AdaBoost")
print(f"Best Parameters: {grid_search.best_params_}")
print(classification_report(y_test, y_pred,
target_names=label_encoder_department.classes_))
# Sonuçları DataFrame olarak kaydetme ve gösterme
results = {
    'Model': 'AdaBoost',
    'Best Params': grid_search.best_params_,
    'Accuracy': accuracy * 100,
    'Precision': precision * 100,
    'Recall': recall * 100,
    'F1-Score': f1 * 100
}
results_df = pd.DataFrame([results])
print(results_df)
# En iyi modelin, vektörleştiricinin ve label encoder'ın kaydedilmesi
with open('best_model_adaboost.pkl', 'wb') as file:
    pickle.dump(best_model, file)
with open('vectorizer.pkl', 'wb') as file:
    pickle.dump(vectorizer, file)
with open('label_encoder_department.pkl', 'wb') as file:
    pickle.dump(label_encoder_department, file)
```

Gradient Boosting

```
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy score, classification report,
precision_recall_fscore_support
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import GradientBoostingClassifier
from scipy.sparse import hstack
from sqlalchemy import create_engine
import pickle
# Veritabanından verileri yükleme
engine = create_engine('sqlite:///health_data.db')
data = pd.read_sql('patient_records', engine)
# Verilerin İşlenmesi
data['Gender'] = data['Gender'].astype('category')
label_encoder_department = LabelEncoder()
data['Department'] = label_encoder_department.fit_transform(data['Department'].astype(str))
vectorizer = TfidfVectorizer(max_features=1000)
X_epicrisis = vectorizer.fit_transform(data['EpicrisisReports'])
X_other = data[['Age', 'Gender']].apply(lambda x: x.cat.codes if x.dtype.name == 'category'
else x)
X = hstack([X_other, X_epicrisis])
y = data['Department']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Gradient Boosting için Parametre Grid
param_grid = {
    'n_estimators': [100, 200, 300],
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 5, 7]
}
# Gradient Boosting Modeli Tanımlanması
model = GradientBoostingClassifier()
# GridSearchCV ile Modelin Eğitimi
grid_search = GridSearchCV(model, param_grid, cv=3, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train)
# En İyi Modelin Seçimi ve Test Verisi Üzerindeki Performansı
best_model = grid_search.best_estimator_
y_pred = best_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
precision, recall, f1, _ = precision_recall_fscore_support(y_test, y_pred,
average='weighted')
# Sonuçların Yazdırılması
print(f"Model: Gradient Boosting")
print(f"Best Parameters: {grid_search.best_params_}")
print(classification_report(y_test, y_pred,
target_names=label_encoder_department.classes_))
# Sonuçları DataFrame olarak kaydetme ve gösterme
results = {
    'Model': 'Gradient Boosting',
    'Best Params': grid_search.best_params_,
    'Accuracy': accuracy * 100,
    'Precision': precision * 100,
    'Recall': recall * 100,
    'F1-Score': f1 * 100
}
results_df = pd.DataFrame([results])
print(results_df)
# En iyi modelin, vektörleştiricinin ve label encoder'ın kaydedilmesi
with open('best_model_gradientboost.pkl', 'wb') as file:
    pickle.dump(best_model, file)
with open('vectorizer.pkl', 'wb') as file:
    pickle.dump(vectorizer, file)
with open('label_encoder_department.pkl', 'wb') as file:
    pickle.dump(label_encoder_department, file)
```

CatBoost

```
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score, classification_report,
precision_recall_fscore_support
from sklearn.preprocessing import LabelEncoder
from catboost import CatBoostClassifier
from scipy.sparse import hstack
from sqlalchemy import create_engine
import pickle
# Veritabanından verileri yükleme
engine = create_engine('sqlite:///health_data.db')
data = pd.read_sql('patient_records', engine)
# Verilerin İşlenmesi
data['Gender'] = data['Gender'].astype('category')
label_encoder_department = LabelEncoder()
data['Department'] = label_encoder_department.fit_transform(data['Department'].astype(str))
vectorizer = TfidfVectorizer(max_features=1000)
X_epicrisis = vectorizer.fit_transform(data['EpicrisisReports'])
X_other = data[['Age', 'Gender']].apply(lambda x: x.cat.codes if x.dtype.name == 'category'
else x)
X = hstack([X_other, X_epicrisis])
y = data['Department']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# CatBoost için Parametre Grid
param_grid = {
    'iterations': [100, 200, 300],
    'depth': [4, 6, 8],
    'learning_rate': [0.01, 0.1, 0.3],
    'l2_leaf_reg': [1, 3, 5]
}
# CatBoost Modeli Tanımlanması
model = CatBoostClassifier(loss_function='MultiClass', verbose=0)
# GridSearchCV ile Modelin Eğitimi
grid_search = GridSearchCV(model, param_grid, cv=3, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train)
# En İyi Modelin Seçimi ve Test Verisi Üzerindeki Performansı
best_model = grid_search.best_estimator_
y_pred = best_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
precision, recall, f1, _ = precision_recall_fscore_support(y_test, y_pred,
average='weighted')
# Sonuçların Yazdırılması
print(f"Model: CatBoost")
print(f"Best Parameters: {grid_search.best_params_}")
print(classification_report(y_test, y_pred,
target_names=label_encoder_department.classes_))
# Sonuçları DataFrame olarak kaydetme ve gösterme
results = {
    'Model': 'CatBoost',
    'Best Params': grid_search.best_params_,
    'Accuracy': accuracy * 100,
    'Precision': precision * 100,
    'Recall': recall * 100,
    'F1-Score': f1 * 100
}
results_df = pd.DataFrame([results])
print(results_df)
# En iyi modelin, vektörleştiricinin ve label encoder'ın kaydedilmesi
with open('best_model_catboost.pkl', 'wb') as file:
    pickle.dump(best_model, file)
with open('vectorizer.pkl', 'wb') as file:
    pickle.dump(vectorizer, file)
with open('label_encoder_department.pkl', 'wb') as file:
    pickle.dump(label_encoder_department, file)
```

LightGBM

```
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split, RandomizedSearchCV
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score, classification_report, precision_recall_fscore_support,
confusion_matrix
from sklearn.preprocessing import LabelEncoder, StandardScaler
import lightgbm as lgb
from scipy.sparse import hstack
import seaborn as sns
from sqlalchemy import create_engine
import pickle
# Veritabanından verileri yükleme
engine = create_engine('sqlite:///health_data.db')
data = pd.read_sql('patient_records', engine)
# Verilerin İşlenmesi
label_encoder_gender = LabelEncoder()
data['Gender'] = label_encoder_gender.fit_transform(data['Gender'])
label_encoder_department = LabelEncoder()
data['Department'] = label_encoder_department.fit_transform(data['Department'])
# Metin verilerini TF-IDF ile sayısal hale getirme
vectorizer = TfidfVectorizer(max_features=1000)
X_epicrisis = vectorizer.fit_transform(data['EpicrisisReports'])
# Sayısal verileri standardize etme (özellikle model performansı için)
scaler = StandardScaler()
X_other = scaler.fit_transform(data[['Age', 'Gender']].to_numpy())
# Verilerin birleştirilmesi
X = hstack([X_other, X_epicrisis])
y = data['Department']
# Eğitim ve test verilerinin ayrılması
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# LightGBM için Parametre Grid (RandomizedSearch için)
param_dist = {
    'max_depth': [5, 7, 10],  # Modelin daha derin bölünmeler yapabilmesi için
    'learning_rate': [0.01, 0.1, 0.3],
    'n_estimators': [100, 200, 300],
    'subsample': [0.7, 0.8, 0.9],
    'num_leaves': [50, 100, 150],  # Daha fazla yaprak sayısı denemesi
    'min_data_in_leaf': [10, 20, 30],  # Yaprak başına minimum veri
    'min_split_gain': [0.0, 0.01, 0.1],  # Split kazancı
    'class_weight': ['balanced']  # Sınıf dengesizliği için otomatik ağırlık
}
# LightGBM Modeli Tanımlanması
model = lgb.LGBMClassifier(objective='multiclass')
# RandomizedSearchCV ile Modelin Eğitimi
random_search = RandomizedSearchCV(model, param_distributions=param_dist, n_iter=10, cv=3,
scoring='accuracy', n_jobs=-1, random_state=42)
random_search.fit(X_train, y_train, categorical_feature=[1])  # Kategorik özellik 'Gender' burada
belirtiliyor
# En İyi Modelin Seçimi ve Test Verisi Üzerindeki Performansı
best_model = random_search.best_estimator_
y_pred = best_model.predict(X_test)
# Modelin Performansının Değerlendirilmesi
accuracy = accuracy_score(y_test, y_pred)
precision, recall, f1, _ = precision_recall_fscore_support(y_test, y_pred, average='weighted')
# Sonuçların Yazdırılması
print(f"Model: LightGBM")
print(f"Best Parameters: {random_search.best_params_}")
print(classification_report(y_test, y_pred, target_names=label_encoder_department.classes_))
# Sonuçları DataFrame olarak kaydetme ve gösterme
results = {
    'Model': 'LightGBM',
    'Best Params': random_search.best_params_,
    'Accuracy': accuracy * 100,
    'Precision': precision * 100,
    'Recall': recall * 100,
    'F1-Score': f1 * 100
}
results_df = pd.DataFrame([results])
print(results_df)
# En iyi modelin, vektörleştiricinin ve label encoder'ın kaydedilmesi
with open('best_model_lightgbm.pkl', 'wb') as file:
    pickle.dump(best_model, file)
with open('vectorizer.pkl', 'wb') as file:
    pickle.dump(vectorizer, file)
with open('label_encoder_department.pkl', 'wb') as file:
    pickle.dump(label_encoder_department, file)
```

XGBosst (Extreme Gradient Boosting)

```
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score, classification_report,
precision_recall_fscore_support
from sklearn.preprocessing import LabelEncoder
import xgboost as xgb
from scipy.sparse import hstack
from sqlalchemy import create_engine
import pickle
# Veritabanından verileri yükleme
engine = create_engine('sqlite:///health_data.db')
data = pd.read_sql('patient records', engine)
# Verilerin İşlenmesi
data['Gender'] = data['Gender'].astype('category')
label_encoder_department = LabelEncoder()
data['Department'] = label_encoder_department.fit_transform(data['Department'].astype(str))
vectorizer = TfidfVectorizer(max_features=1000)
X_epicrisis = vectorizer.fit_transform(data['EpicrisisReports'])
X_other = data[['Age', 'Gender']].apply(lambda x: x.cat.codes if x.dtype.name == 'category'
else x)
X = hstack([X_other, X_epicrisis])
y = data['Department']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# XGBoost için Parametre Grid
param_grid = {
    'max_depth': [3, 5, 7],
    'learning_rate': [0.01, 0.1, 0.3],
    'n_estimators': [100, 200, 300],
    'subsample': [0.7, 0.8, 0.9]
}
# XGBoost Modeli Tanımlanması
model = xgb.XGBClassifier(objective='multi:softmax', num_class=len(y.unique()),
enable_categorical=True)
# GridSearchCV ile Modelin Eğitimi
grid_search = GridSearchCV(model, param_grid, cv=3, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train)
# En İyi Modelin Seçimi ve Test Verisi Üzerindeki Performansı
best_model = grid_search.best_estimator_
y_pred = best_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
precision, recall, f1, _ = precision_recall_fscore_support(y_test, y_pred,
average='weighted')
# Sonuçların Yazdırılması
print(f"Model: XGBoost")
print(f"Best Parameters: {grid_search.best_params_}")
print(classification_report(y_test, y_pred,
target_names=label_encoder_department.classes_))
# Sonuçları DataFrame olarak kaydetme ve gösterme
results = {
    'Model': 'XGBoost',
    'Best Params': grid_search.best_params_,
    'Accuracy': accuracy * 100,
    'Precision': precision * 100,
    'Recall': recall * 100,
    'F1-Score': f1 * 100
}
results_df = pd.DataFrame([results])
print(results_df)
# En iyi modelin, vektörleştiricinin ve label encoder'ın kaydedilmesi
with open('best_model_xgboost.pkl', 'wb') as file:
    pickle.dump(best_model, file)
with open('vectorizer.pkl', 'wb') as file:
    pickle.dump(vectorizer, file)
with open('label_encoder_department.pkl', 'wb') as file:
    pickle.dump(label_encoder_department, file)
```

EK-2 VERİ ANALİZİ

```
import pandas as pd
import matplotlib.pyplot as plt
from sqlalchemy import create_engine
# Veritabanına bağlanma
engine = create_engine('sqlite:///health_data.db')
data = pd.read_sql('patient_records', engine)
# Yaş ile ilgili analizler
age_mean = data['Age'].mean()
age_min = data['Age'].min()
age_max = data['Age'].max()
# Yaş analiz sonuçlarını tablo haline getirme
summary_df = pd.DataFrame({
    'Statistic': ['Mean Age', 'Min Age', 'Max Age'],
    'Value': [age_mean, age_min, age_max]
})
print("Yaş Analizi Sonuçları:")
print(summary_df)
# Yaş dağılımını görselleştirme
plt.hist(data['Age'], bins=20, color='skyblue', edgecolor='black')
plt.title('Hasta Yaşlarının Dağılımı')
plt.xlabel('yaş')
plt.ylabel('Hasta Sayısı')
plt.grid(True)
plt.show()
# Cinsiyet ile ilgili analizler
gender_counts = data['Gender'].value_counts()
total_gender = gender_counts.sum()
female_percentage = (gender_counts['KADIN'] / total_gender) * 100 if 'KADIN' in
gender_counts else 0
male_percentage = (gender_counts['ERKEK'] / total_gender) * 100 if 'ERKEK' in
gender_counts else 0

# Cinsiyet analiz sonuçlarını tablo haline getirme
gender_summary_df = pd.DataFrame({
    'Gender': ['Kadın', 'Erkek'],
    'Percentage': [female_percentage, male_percentage]
})
print("\nCinsiyet Dağılımı Sonuçları:")
print(gender_summary_df)
# Cinsiyet dağılımını görselleştirme
plt.pie(gender_summary_df['Percentage'], labels=gender_summary_df['Gender'],
autopct='%1.1f%%', colors=['pink', 'lightblue'])
plt.title('Cinsiyet Dağılımı')
plt.show()
# Bölüme göre dağılım
department_counts = data['Department'].value_counts()
# Bölüm analiz sonuçlarını tablo haline getirme
department_summary_df = pd.DataFrame({
    'Department': department_counts.index,
    'Number of Patients': department_counts.values
})
print("\nBölüme Göre Hasta Sayısı:")
print(department_summary_df)
# Bölüm dağılımını görselleştirme
plt.figure(figsize=(10, 8))
plt.barh(department_summary_df['Department'], department_summary_df['Number of
Patients'], color='lightgreen')
plt.xlabel('Hasta Sayısı')
plt.ylabel('Branş')
plt.title('Hastaların Bölümlere Göre Dağılımı')
plt.grid(True)
plt.show()
```

EK-3 UYGULAMA

```
import streamlit as st
import pickle
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from scipy.sparse import hstack

# Model, vectorizer ve label encoder yüklemesi
with open('best_model_xgboost.pkl', 'rb') as file:
    model = pickle.load(file)
with open('vectorizer.pkl', 'rb') as file:
    vectorizer = pickle.load(file)
with open('label_encoder_department.pkl', 'rb') as file:
    label_encoder_department = pickle.load(file)

# Kullanıcı girdilerini alacak bir fonksiyon
def user_input_features():
    age = st.number_input("Yaş", min_value=0, max_value=120, value=30)
    gender = st.selectbox("Cinsiyet", ("KADIN", "ERKEK"))
    complaint = st.text_area("Şikayet")

    # Cinsiyeti sayısal değere dönüştürme
    gender_num = 0 if gender == "KADIN" else 1

    # Girdi verilerini DataFrame olarak oluşturma
    data = pd.DataFrame([[age, gender_num, complaint]], columns=["Age", "Gender",
"EpicrisisReports"])
    return data

# Streamlit uygulama başlığı
st.title("Departman Tahmin Uygulaması")

# Kullanıcı girdilerini al
input_data = user_input_features()

# Epikriz raporlarını TF-IDF vektörlerine dönüştürme
tfidf_data = vectorizer.transform(input_data['EpicrisisReports'])
input_data = input_data.drop('EpicrisisReports', axis=1)
final_input = hstack([input_data, tfidf_data])

# Tahmin yap ve sonucu göster
if st.button("Tahmin Et"):
    prediction = model.predict(final_input)
    predicted_department_name =
label_encoder_department.inverse_transform([prediction])[0]
    st.write(f"Tahmin edilen departman: {predicted_department_name}")
```

EK-4 ÇIKTI ÖRNEKLERİ

Şekil 9. Uygulama Çıktısı

ÖZGEÇMİŞ

İhsan ARVAS, ilköğrenimini Rauf Orbay İlköğretim Okulu'nda bitirdikten sonra lise eğitimini 2012 yılında Ümit Meriç Kız Anadolu İmam Hatip Lisesi'nde tamamladı. 2016-2017 öğretim yılında Karadeniz Teknik Üniversitesi Fen Fakültesi İstatistik ve Bilgisayar Bilimleri bölümünde lisans eğitimine başladı. 2020 yılında bu bölümden mezun oldu. Aynı yıl Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstatistik ve Bilgisayar Bilimleri Anabilim dalında tezli yüksek lisans programına başladı.