T.C.
MEHMET AKİF ERSOY ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

PROBİYOTiKLERİN SINIFLANDIRILMASINDA YAPAY ZEKA VE MAKİNE ÖĞRENMESİ TEKNİKLERİNİN UYGULAMASI VE KARŞILAŞTIRILMASI

LİSANS TEZİ

GÖKÇEN DİLEK ALAK - ÖZGÜR YİĞİT AŞİT

BURDUR, 2025

T.C.
MEHMET AKİF ERSOY ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ

ETİK BEYAN

Mehmet Akif Ersoy Üniversitesi Fen Bilimler Enstitüsü Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğine göre hazırlamış olduğum " **Probiyotiklerin Sınıflandırılmasında Yapay Zeka ve Makine Öğrenmesi Tekniklerinin Uygulanması ve Karşılaştırılması** " adlı tezin hazırlanması sürecinde akademik etik ilkeleri ihlal etmediğimi taahhüt eder, tezimin kağıt ve elektronik kopyalarının Mehmet Akif Ersoy Üniversitesi Fen Bilimler Enstitüsü arşivlerinde aşağıda belirttiğim koşullarda saklanmasına izin verdiğimi onaylarım.

Fen Bilimler Enstitüsü Lisansüstü Eğitim-Öğretim Yönetmeliğinin ilgili maddeleri
uyarınca gereğinin yapılmasını arz ederim.

□ Tezimin tamamı her yerden erişime açılabilir.

□ Tezim sadece Mehmet Akif Ersoy Üniversitesi yerleşkelerinde erişime açılabilir.

□ Tezimin 3 yıl süreyle erişime açılmasını istemiyorum. Bu sürenin sonunda uzatma için başvuruda bulunmadığım takdirde, tezimin tamamı her yerden erişime açılabilir.

Adı Soyadı
ÖZGÜR YİĞİT AŞİT – GÖKÇEN DİLEK ALAK

Tarih ve İmza

ÖNSÖZ

Bu çalışma, lisans öğrenimimiz süresince edindiğimiz bilgi ve tecrübeleri pekiştirme amacıyla gerçekleştirilmiştir. Tezimizin her aşamasında bizlere yol gösteren, bilgi ve birikimiyle katkı sunan değerli danışmanımız **Doktor Öğretim Üyesi Tülay Turan** ve **Prof. Dr. Gülden Başyigit Kılıç**'a en içten teşekkürlerimizi sunarız. Kendisinin desteği ve yönlendirmeleri, çalışmamızın her aşamasında bizlere ilham kaynağı olmuştur.

Ayrıca, verilerin sağlanmasında ve çeşitli konularda değerli katkılarını esirgemeyen **Prof. Dr. Gülden Başyigit Kılıç**'a teşekkür ederiz. Çalışmamızın kapsamının genişlemesine ve derinleşmesine olanak tanıyan destekleri bizim için çok kıymetli olmuştur.

Çalışmamızın, alanındaki bilgi birikimine katkı sağlamasını ve ilerleyen dönemlerde yapılacak çalışmalara ışık tutmasını temenni ederiz.

Özgür Yiğit AŞİT

Gökçen Dilek ALAK

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

KISALTMALAR VE SİMGELER DİZİNİ

AdaBoost	Adaptive Boosting
CART	Classification and Regression Trees
EFB	Exclusive Feature Bundling
Extra Trees	Extremely Randomized Trees
GOSS	Gradient-based One-Side Sampling
GRU	Gated Recurrent Unit
HistGradient Boosting	Histogram-based Gradient Boosting
LightGBM	Light Gradient Boosting Machine
LSTM	Long Short-Term Memory
PCA	Principal Component Analysis
PyTorch	(Python Deep Learning Kütüphanesi)
Random Forest	(Karar Ağaçları Topluluğu Algoritması)
RNN	Recurrent Neural Network
Scikit-learn (sklearn)	(Python Makine Öğrenmesi Kütüphanesi)
SVM	Support Vector Machine
TensorFlow	(Google tarafından geliştirilen Derin Öğrenme Kütüphanesi)
TSNE	t-Distributed Stochastic Neighbor Embedding
XGBoost	Extreme Gradient Boosting
YZ.	Yapay Zeka

ÖZET

PROBİYOTiKLERİN SINIFLANDIRILMASINDA YAPAY ZEKA VE MAKİNE ÖĞRENMESİ TEKNİKLERİNİN UYGULAMASI VE KARŞILAŞTIRILMASI
LİSANS TEZİ
GÖKÇEN DİLEK ALAK - ÖZGÜR YİĞİT AŞİT
BURDUR MEHMET AKİF ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI
(TEZ DANIŞMANI: Doktor Öğretim Üyesi TÜLAY TURAN)
BURDUR, HAZİRAN - 2025

Mikroorganizmalar, doğada yaygın olarak bulunan ve çeşitli biyolojik süreçlerde önemli rollere sahip olan canlılardır. Bu mikroskobik canlılar, bakteriler, mantarlar, arkeler ve protistler gibi çeşitli gruplara ayrılır. Mikroorganizmaların sınıflandırılması, biyolojik çeşitliliğin anlaşılması, patojenlerin tanımlanması ve endüstriyel uygulamalarda kullanılmaları açısından büyük önem taşır. Geleneksel sınıflandırma yöntemleri, mikroorganizmaların morfolojik, biyokimyasal ve genetik özelliklerine dayanır. Ancak, son yıllarda Fourier Dönüşümlü Kızılötesi (FTIR) spektroskopisi gibi modern analitik teknikler, mikroorganizmaların hızlı ve doğru bir şekilde sınıflandırılmasında giderek daha fazla kullanılmaktadır. Bu projenin amacı, endüstriyel alanlar içerisinde kullanılan *Enterococcus faecium, Lactobacillus plantarum ve Lactobacillus fermentum* probiyotiklerinin sınıflandırılması sırasındaki insan kaynaklı hataların minimize edilmesi ve yapay zeka ve makine öğrenmesi algoritmaları kullanılarak bu üç mikroorganizma ile ilgili model eğitiminin yapılması ve bundan sonraki mikroorganizma laboratuvar sonuçlarının hızlı bir şekilde sınıflandırılmasına katkıda bulunulmaktır. Bu çalışmada çeşitli bireysel ve hibrit olan makine öğrenmesi ve derin öğrenme modelleri kullanılmıştır. CNN modeli percision, doğruluk ve recall değerlerinde yüksek performans göstermiştir. Bu bulgu, derin öğrenme tabanlı CNN modelinin mikroorganizma sınıflandırma probleminde üstün bir performans sunduğunu ve veri artırma tekniklerinin model başarımını önemli ölçüde iyileştirdiğini göstermektedir.

Anahtar Kelimeler: Mikroorganizma sınıflandırma, yapay zeka, makine öğrenmesi, derin öğrenme, probiyotikler.

ABSTRACT

APPLICATION AND COMPARISON OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING TECHNIQUES IN THE CLASSIFICATION OF PROBIOTICS
BACHELOR'S THESIS
ÖZGÜR YİĞİT AŞİT - GÖKÇEN DİLEK ALAK
BURDUR MEHMET AKİF ERSOY UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF COMPUTER ENGINEERING
(THESIS ADVISOR: Assistant Professor TÜLAY TURAN)
BURDUR, JUNE - 2025

Microorganisms are widely found in nature and play significant roles in various biological processes. These microscopic organisms are classified into different groups, including bacteria, fungi, archaea, and protists. The classification of microorganisms is crucial for understanding biological diversity, identifying pathogens, and facilitating industrial applications. Traditional classification methods rely on morphological, biochemical, and genetic characteristics. However, in recent years, modern analytical techniques such as Fourier Transform Infrared (FTIR) spectroscopy have been increasingly used for the rapid and accurate classification of microorganisms. The aim of this project is to minimize human-induced errors in the classification of *Enterococcus faecium, Lactobacillus plantarum,* and *Lactobacillus fermentum* probiotics, which are used in industrial applications. Additionally, this study seeks to train models using artificial intelligence and machine learning algorithms to improve the rapid classification of future laboratory results involving microorganisms. Various individual and hybrid machine learning and deep learning models have been utilized in this study. The CNN model demonstrated high performance in terms of precision, accuracy, and recall. These findings suggest that the deep learning-based CNN model provides superior performance in microorganism classification tasks and that data augmentation techniques significantly enhance model efficiency.

Keywords: Microorganism classification, artificial intelligence, machine learning, deep learning, probiotics.

1. GİRİŞ

Probiyotikler, sağlık üzerindeki olumlu etkileri nedeniyle gıda, ilaç ve biyoteknoloji endüstrilerinde geniş çapta kullanılmaktadır. Özellikle *Enterococcus faecium*, *Lactobacillus plantarum* ve *Lactobacillus fermentum* türleri, endüstride en yaygın kullanılan ve doğru sınıflandırılmasına en çok ihtiyaç duyulan probiyotikler arasında yer almaktadır. Ancak, bu üç spesifik probiyotiğin sınıflandırılması üzerine doğrudan odaklanan herhangi bir çalışma literatürde bulunmamaktadır. Bu boşluğu doldurmak amacıyla, çeşitli hibrit ve bireysel modelleri kullanarak bir karşılaştırma yaparak en yüksek performans gösteren model bulunmuştur. Derin öğrenme tabanlı Convolutional Neural Network (CNN) algoritması kullanarak bu bakterilerin sınıflandırılması gerçekleştirilmiştir. Önerilen yöntem, yüksek doğruluk oranıyla (*%92*) bu probiyotiklerin tanımlanmasını sağlayarak hem akademik araştırmalara hem de endüstriyel uygulamalara önemli bir katkı sunmaktadır. Bu çalışma, probiyotiklerin otomatik tanımlanması ve sınıflandırılmasına yönelik yeni bir yaklaşım önererek, sağlık ve biyoteknoloji alanındaki gelecekteki araştırmalar için önemli bir referans niteliği taşımaktadır. Bu çalışmaya yardımcı olan mikroorganizmaların sınıflandırılması alanında yapılmış çeşitli çalışmalar literatür taraması bölümünde belirtilmiştir.

1.1. Probiyotikler

Probiyotikler, belirli sayılarda alındıklarında sağlık üzerinde olumlu etkiler yaratan canlı mikroorganizmalar olarak tanımlanmaktadır. Genellikle bakterilerden oluşan bu mikroorganizmalar, özellikle laktik asit bakterileri (Lactobacillus ve Bifidobacterium türleri gibi) ve mayalar (Saccharomyces boulardii gibi) içermektedir. Probiyotikler, bağırsak florasını dengelemekte ve sindirim sistemi sağlığını desteklemektedir (Başyiğit Kılıç, 2009).

Probiyotik kavramı, ilk olarak 20. yüzyılın başlarında Nobel ödüllü Rus bilim insanı Elie Metchnikoff tarafından ortaya atılmıştır (Metchnikoff, 1907). Metchnikoff, Bulgar köylülerinin uzun ömürlü olmalarını, yoğurt ve fermente süt ürünleri tüketmelerine bağlamıştır (Metchnikoff, 1907). Bu ürünlerde bulunan laktik asit bakterilerinin bağırsak sağlığını iyileştirdiğini ve böylece genel sağlığı olumlu yönde

etkilediğini öne sürmüştür (Özen & Dinleyici, 2015). Bu gözlemler, probiyotiklerin modern tıp ve beslenme biliminde önemli bir yer edinmesine yol açmıştır (Hill vd., 2014).

Probiyotiklerin hayatımızdaki önemi oldukça büyüktür. Bağırsak sağlığını korumakta ve sindirim sistemi problemlerini önlemektedir. İrritabl bağırsak sendromu (IBS), inflamatuar bağırsak hastalıkları (IBD) ve ishal gibi durumlarda etkili olmaktadır. Bağışıklık sistemini güçlendirmekte ve enfeksiyonlara karşı direnci artırmaktadır. Ayrıca, probiyotiklerin mental sağlık üzerinde de olumlu etkileri bulunmaktadır; anksiyete ve depresyon semptomlarını hafifletebilmektedir. Probiyotikler, gıda takviyeleri, fermente gıdalar (yoğurt, kefir, lahana turşusu gibi) ve bazı içecekler yoluyla alınabilmektedir. Bu nedenle, probiyotikler modern beslenme ve sağlık uygulamalarında önemli bir rol oynamaktadır (Başyiğit Kılıç, 2009).

1.1.1. Enterococcus faecium

Enterococcus faecium, probiyotik özelliklere sahip bir bakteri türü olup, insan ve hayvan bağırsak florasında doğal olarak bulunmaktadır. Gram-pozitif, fakültatif anaerobik bir mikroorganizma olan bu bakteri, laktik asit bakterileri grubuna dahil edilmekte ve bağırsak sağlığını desteklemektedir. İlk olarak 20. yüzyılın başlarında tanımlanan *Enterococcus faecium*, bağırsak florasını dengelemekte, sindirim sistemi sağlığını iyileştirmekte ve ishal, kabızlık, irritabl bağırsak sendromu (IBS) gibi durumlarda etkili olmaktadır. Bağışıklık sistemini güçlendirerek enfeksiyonlara karşı direnci artırmakta, patojen bakterilere karşı antimikrobiyal maddeler üretmekte ve besin emilimini iyileştirmektedir. Gıda takviyeleri, probiyotik yoğurtlar, kefir ve diğer fermente ürünlerde kullanılan bu bakteri, hayvan yemlerine eklenerek hayvan sağlığını desteklemekte ve verimliliği artırmaktadır. Genellikle güvenli kabul edilen Enterococcus faecium'un kullanımı, bağışıklık sistemi zayıf olan bireylerde ve belirli sağlık koşullarında doktor tavsiyesi alınarak yapılmalıdır. Antibiyotik direnci gibi potansiyel riskler nedeniyle, kontrollü ve bilinçli bir şekilde kullanılması önem taşımaktadır (Başyiğit Kılıç, 2009).

1.1.2. Lactobacillus plantarum

Lactobacillus plantarum, bitki bazlı fermente gıdalarda yaygın olarak bulunan ve insan bağırsak sistemi ile ağız boşluğunda da doğal olarak yaşayan çok yönlü bir probiyotik bakteridir. Bu gram-pozitif, çubuk şeklindeki bakteri, geniş bir pH aralığında hayatta kalabilme ve çeşitli karbonhidratları fermante edebilme özelliğiyle öne çıkmakta ve güçlü antimikrobiyal aktivitesi sayesinde bakteriyosin üretimi yoluyla patojen mikroorganizmaların büyümesini engellemektedir. L. plantarum, bağırsak bariyer fonksiyonunun güçlendirilmesi, sindirim sağlığının iyileştirilmesi, enflamatuar bağırsak hastalıklarının semptomlarının hafifletilmesi ve bağışıklık modulasyonu gibi sağlık yararları göstermektedir. Ayrıca turşu, zeytin, ekşi hamur ve lahana turşusu gibi fermente gıdaların üretiminde önemli bir rol oynamakta ve günümüzde fermente gıda üretiminde starter kültür olarak ve çeşitli probiyotik preparatlarda yaygın şekilde kullanılmaktadır (Başyiğit Kılıç, 2009).

1.1.3. Lactobacillus fermentum

Lactobacillus fermentum, insan bağırsak sisteminde doğal olarak bulunan, gram-pozitif, çubuk şeklinde ve fakültatif anaerobik bir laktik asit bakterisidir. Bu probiyotik, karbonhidratları fermante ederek laktik asit ve asetik asit üretme kabiliyetiyle bilinmekte ve bu özelliği sayesinde patojen mikroorganizmaların büyümesini engelleyerek bağırsak pH'sını düşürmektedir. L. fermentum, özellikle antimikrobiyal bileşikler üreterek patojenlere karşı koruma sağlamasının yanı sıra, bağışıklık sistemini güçlendirme, kolesterol seviyelerini düşürme ve oksidatif strese karşı koruma sağlama gibi sağlık yararları göstermektedir. Ayrıca fermente gıdalarda, ekşi hamurda ve bazı geleneksel süt ürünlerinde doğal olarak bulunabilmekte ve günümüzde çeşitli probiyotik takviyelerde ve fonksiyonel gıdalarda aktif bir bileşen olarak kullanılmaktadır (Başyiğit Kılıç, 2009).

1.2. FTIR Spektroskopisi

Fourier Dönüşümlü Kızılötesi Spektroskopisi (FTIR), moleküllerin titreşimsel hareketlerini inceleyerek kimyasal bağ yapılarını ve fonksiyonel gruplarını belirlemede kullanılan ileri bir analitik tekniktir (Gazi Üniversitesi, t.y.). Bu teknik, günümüzde kimya, malzeme bilimi, biyoloji, eczacılık ve adli tıp gibi çeşitli alanlarda yaygın şekilde uygulanmaktadır (Mettler Toledo, 2023).

FTIR spektroskopisinin tarihsel gelişimi 19. yüzyılın başlarına dayanmaktadır. 1800 yılında William Herschel tarafından kızılötesi radyasyonun keşfi ile bu alandaki çalışmalar başlamıştır (Portable Analytical Solutions, 2023). Ancak modern FTIR tekniğinin temelleri 1880'lerde Albert Abraham Michelson tarafından geliştirilen Lutz, M. (2013). *Learning Python* (5th ed.). O'Reilly Media.interferometre ile atılmıştır. Michelson interferometresi, ışık dalgalarının girişim özelliklerini kullanan ve daha sonra FTIR cihazlarının çalışma prensibini oluşturacak olan temel cihaz olarak kabul edilmektedir.

1950'li yıllarda bilgisayar teknolojisinin gelişmesiyle birlikte Fourier dönüşümü hesaplamalarının pratik uygulaması mümkün hale gelmiştir. 1960'larda James Cooley ve John Tukey tarafından geliştirilen Hızlı Fourier Dönüşümü (FFT) algoritması, FTIR teknolojisinin ticari olarak uygulanabilir hale gelmesini sağlamıştır. İlk ticari FTIR cihazları 1970'lerde piyasaya sürülmüş ve bu tarihten itibaren teknik sürekli olarak geliştirilmektedir.

FTIR spektroskopisi tekniği, klasik dispersif infrared spektroskopisinden farklı olarak çalışmaktadır. Bu teknikte, örnek üzerine tüm dalga boylarını içeren infrared ışığı gönderilmekte ve örneğin bu ışığı soğurması sonucu elde edilen interferogram, matematiksel bir işlem olan Fourier dönüşümü ile spektruma dönüştürülmektedir (Gazi Üniversitesi, t.y.). Bu yaklaşım, veri toplama hızını artırmakta ve sinyal-gürültü oranını iyileştirmektedir (Gazi Üniversitesi, t.y.).

Fourier Dönüşümlü Kızılötesi (FTIR) spektroskopisi, organik ve inorganik bileşiklerin kimyasal yapılarının belirlenmesinde yaygın olarak kullanılan bir tekniktir (Başyiğit Kılıç ve Karahan, 2010). Bu yöntem, moleküllerin karakteristik titreşimlerini analiz ederek onların fonksiyonel gruplarını ve bağ yapılarını tespit eder (Başyiğit Kılıç ve Karahan, 2010). FTIR spektroskopisi, hızlı ve hassas sonuçlar vermesi, numuneye

zarar vermemesi ve küçük miktarlardaki örneklerle analiz yapabilme avantajları nedeniyle tercih edilmektedir (Çırak, 2011).

Mikroorganizmaların tanımlanması ve sınıflandırılması, geleneksel olarak morfolojik ve biyokimyasal özelliklerine dayanırken, son yıllarda FTIR spektroskopisi gibi modern teknikler bu alanda önemli katkılar sağlamıştır (Albayrak, 2010). FTIR, mikroorganizmaların hücresel bileşenlerinin (proteinler, lipitler, nükleik asitler ve karbonhidratlar) özgün infrared absorpsiyon bantlarını tespit ederek, her bir mikroorganizma için benzersiz bir "parmak izi" spektrumu oluşturur (Albayrak, 2010). Bu parmak izi spektrumları, mikroorganizmaların tür veya alt tür seviyesinde doğru bir şekilde tanımlanmasına olanak tanır (Albayrak, 2010).

FTIR spektroskopisinin mikrobiyolojideki uygulamaları arasında laktik asit bakterilerinin (LAB) tanımlanması da bulunmaktadır (Başyiğit Kılıç ve Karahan, 2010). Yapılan çalışmalar, FTIR spektroskopisinin LAB'nin tanısında rutin olarak kullanılabileceğini göstermiştir (Başyiğit Kılıç ve Karahan, 2010). Bu teknik, LAB'nin hızlı ve doğru bir şekilde tanımlanmasını sağlayarak, gıda endüstrisinde kalite kontrol süreçlerine önemli katkılar sunmaktadır (Başyiğit Kılıç ve Karahan, 2010).

FTIR spektroskopisinin mikroorganizmaların sınıflandırılmasındaki etkinliği, çok değişkenli analiz yöntemleriyle desteklendiğinde daha da artmaktadır (Albayrak, 2010). Özellikle, ana bileşen analizi (PCA) ve doğrusal diskriminant analizi (LDA) gibi istatistiksel yöntemler, FTIR spektrumlarından elde edilen verilerin yorumlanmasında kullanılarak, mikroorganizmaların doğru bir şekilde sınıflandırılmasına yardımcı olur (Albayrak, 2010).

Sonuç olarak, FTIR spektroskopisi, mikroorganizmaların hızlı, güvenilir ve tahribatsız bir şekilde tanımlanması ve sınıflandırılması için güçlü bir araçtır (Başyiğit Kılıç & Karahan, 2010). Bu teknik, özellikle gıda, ilaç ve biyoteknoloji endüstrilerinde kalite kontrol ve araştırma-geliştirme süreçlerinde yaygın olarak kullanılmaktadır (Çırak, 2011).

1.3. Projede Kullanılan Kodlama Dili - PYTHON

Python programlama dili, Guido van Rossum tarafından 1980'lerin sonunda geliştirilmeye başlanmış ve ilk olarak 1991 yılında piyasaya sürülmüştür. Python okunabilirliği yüksek, basit ve anlaşılır bir dil olarak tasarlanmıştır. Python'un ismi, Guido van Rossum'un sevdiği bir komedi grubu olan Monty Python'dan esinlenilmiştir. Python'un temel amaçları arasında, programcıların daha az kod yazarken daha fazla iş yapabilmesini sağlamak ve kodun okunabilirliğini artırmak yer almaktadır (Van Rossum ve Drake, 2009; Lutz, 2013).

Günümüzde Python, özellikle makine öğrenmesi ve yapay zeka alanlarında büyük bir popülerlik kazanmıştır (Müller ve Guido, 2016; Géron, 2019). Bunun başlıca nedenleri şunlardır:

- Geniş Kütüphane Desteği: Python, makine öğrenmesi ve yapay zeka için TensorFlow, PyTorch, Keras, Scikit-learn gibi güçlü kütüphanelere sahiptir (Goodfellow, Bengio, & Courville, 2016; Géron, 2019). Bu kütüphaneler, karmaşık algoritmaların kolayca uygulanabilmesini sağlamaktadır.
- Kolay Öğrenilebilirlik: Python'un sözdizimi basit ve anlaşılırdır. Bu da yeni başlayanlar için öğrenmeyi kolaylaştırır ve hızlı prototip oluşturmayı mümkün kılmaktadır (Lutz, 2013).
- Çok Yönlülük: Python, veri analizi, web geliştirme, otomasyon, bilimsel hesaplamalar ve daha birçok alanda kullanılabilmektedir (Müller & Guido, 2016). Bu çok yönlülük, özellikle veri bilimcileri ve yapay zeka araştırmacıları için büyük bir avantaj sağlamaktadır.
- Entegrasyon ve Uyumluluk: Python, diğer diller ve sistemlerle kolayca entegre olabilmektedir. Örneğin, C/C++ ile yazılmış kütüphanelerle birlikte kullanılabilmektedir (Van Rossum & Drake, 2009).

Python, makine öğrenmesi ve yapay zeka projelerinde hızlı geliştirme ve test etme imkanı sunmaktadır (Géron, 2019). Ayrıca, büyük veri setleri üzerinde çalışırken veri manipülasyonu ve analizi için Pandas ve NumPy gibi kütüphanelerle güçlü bir altyapı sağlamaktadır (McKinney, 2012; Oliphant, 2006).

1.3.1. Kullanılan Kütüphaneler

- **Temel Kütüphaneler:** Python'da veri analizi ve görselleştirme için sıkça kullanılan kütüphaneler bulunmaktadır. **Pandas**, veri manipülasyonu ve analizi işlemlerini gerçekleştirmektedir (McKinney, 2012). **NumPy**, bilimsel hesaplamalar için büyük diziler ve matrisler üzerinde işlem yapma imkanı sunmaktadır (Oliphant, 2006). **Matplotlib** ve **Seaborn**, veri görselleştirme işlemlerini gerçekleştirmektedir; Matplotlib temel grafikler oluştururken (Hunter, 2007), Seaborn daha estetik ve istatistiksel grafikler sunmaktadır (Waskom, 2021). **Time** kütüphanesi ise zamanla ilgili işlemler için kullanılmaktadır; örneğin kodun çalışma süresini ölçmek gibi işlemler yapmaktadır (Python Software Foundation, 2023).
- **İstatistiksel Analiz & Sinyal İşleme:** Bu kütüphaneler, veriler üzerinde istatistiksel analizler ve sinyal işleme işlemlerini gerçekleştirmektedir. SciPy.stats, normalite testleri (Shapiro-Wilk, D'Agostino's K^2), çarpıklık (skew) ve basıklık (kurtosis) hesaplamaları gibi istatistiksel testler sunmaktadır. SciPy.signal ise sinyal işleme işlemlerini gerçekleştirmektedir; örneğin, Savitzky-Golay filtresi (savgol_filter) ve tepe noktalarını bulma (find_peaks) gibi işlemler yapmaktadır (Virtanen vd., 2020).

- **Önişleme & Veri Dönüşümü:** Veri önişleme, makine öğrenmesi modellerinin başarısı için kritik bir adım olarak değerlendirilmektedir. Sklearn.preprocessing, verileri standartlaştırma (StandardScaler), normalizasyon (MinMaxScaler) ve etiket kodlama (LabelEncoder) gibi işlemlerle hazırlamaktadır. Sklearn.model_selection ise veri setini eğitim ve test setlerine ayırma işlemini gerçekleştirmektedir (train_test_split), bu da modelin performansını değerlendirmek için önemli bir adım olarak görülmektedir (Pedregosa vd., 2011).

- **Makine Öğrenmesi - Sınıflandırma Algoritmaları:** Bu kütüphaneler, sınıflandırma problemlerini çözmek için kullanılan çeşitli algoritmaları içermektedir. Sklearn.ensemble, RandomForestClassifier, GradientBoostingClassifier gibi ensemble yöntemlerini sunmaktadır. Sklearn.svm, Destek Vektör Makineleri (SVM) için kullanılmaktadır.

Sklearn.neighbors, K-en yakın komşu (KNN) algoritmasını içermektedir. Sklearn.linear_model, lojistik regresyon gibi lineer modelleri sunmaktadır ve Sklearn.metrics, model performansını değerlendirme işlemlerini gerçekleştirmektedir (Pedregosa vd., 2011).

- **Makine Öğrenmesi - Kümeleme Algoritmaları:** Kümeleme, benzer veri noktalarını gruplamak için kullanılan bir teknik olarak değerlendirilmektedir (Jain, 2010**).** Sklearn.cluster, KMeans, DBSCAN ve AgglomerativeClustering gibi kümeleme algoritmalarını içermektedir (Pedregosa vd., 2011). Scipy.cluster.hierarchy, hiyerarşik kümeleme işlemlerini gerçekleştirmekte ve dendrogram gibi görselleştirme araçları sunmaktadır (Virtanen vd., 2020).

- **Ekstra Makine Öğrenmesi Modelleri:** Bu kütüphaneler, gelişmiş makine öğrenmesi modelleri ve teknikleri sunmaktadır. XGBoost ve LightGBM**,** gradient boosting algoritmalarının optimize edilmiş versiyonlarını içermektedir (Chen ve Guestrin, 2016; Ke vd., 2017). Imblearn.over_sampling**,** dengesiz veri setlerini dengeleme işlemlerini gerçekleştirmektedir (Lemaître, Nogueira, & Aridas, 2017). Sklearn.ensemble**,** birden fazla modeli birleştirme işlemlerini sunmaktadır (Pedregosa vd., 2011).

- **Derin Öğrenme - TensorFlow & Keras:** Derin öğrenme, büyük ve karmaşık veri setleri üzerinde yüksek performanslı modeller oluşturmak için kullanılmaktadır (Goodfellow, Bengio, & Courville, 2016). TensorFlow, derin öğrenme modelleri oluşturma ve eğitme işlemlerini gerçekleştirmektedir (Abadi vd., 2016). Keras**,** TensorFlow üzerinde çalışan ve kullanımı kolay bir arayüz sunmaktadır (Chollet, 2015). TensorFlow.keras.models**,** Sequential gibi modeller oluşturma imkanı sağlamaktadır. TensorFlow.keras.layers**,** Dense, Dropout, LSTM gibi çeşitli katmanlar ekleme işlemlerini gerçekleştirmektedir. TensorFlow.keras.optimizers**,** Adam gibi optimizasyon algoritmalarını içermektedir (Kingma ve Ba, 2015).

- **Derin Öğrenme - PyTorch Kütüphaneleri:** PyTorch, derin öğrenme modelleri oluşturmak ve eğitmek için kullanılan güçlü bir kütüphane olarak değerlendirilmektedir (Paszke vd., 2019). torch modülü, temel tensor işlemlerini ve hesaplamalarını gerçekleştirmektedir. torch.nn modülü, sinir ağı katmanlarını ve modellerini tanımlamak için kullanılmaktadır. torch.optim modülü, optimizasyon algoritmalarını içermekte ve model eğitimi sırasında kullanılmaktadır. torch.utils.data modülü, veri setlerini yönetmek ve veri yükleyicileri (DataLoader) oluşturmak için kullanılmaktadır. Bu kütüphaneler, derin öğrenme projelerinde esnek ve etkili bir çalışma ortamı sunmaktadır.

- **Boyut Azaltma & Özellik Çıkarma Kütüphaneleri:** Bu kütüphaneler, veri setlerinin boyutunu azaltmak ve özellik çıkarma işlemlerini gerçekleştirmek için kullanılmaktadır. sklearn.decomposition modülü, PCA (Principal Component Analysis) ve TruncatedSVD (Truncated Singular Value Decomposition) gibi yöntemleri içermektedir (Pedregosa vd., 2011). PCA, verilerin varyansını koruyarak boyut azaltma işlemi gerçekleştirmektedir. TruncatedSVD ise özellikle seyrek matrisler üzerinde boyut azaltma işlemleri yapmaktadır. sklearn.manifold modülü, TSNE **(**t-Distributed Stochastic Neighbor Embedding) ve Isomap gibi yöntemleri sunmaktadır. TSNE**,** yüksek boyutlu verileri iki veya üç boyuta indirgeyerek görselleştirme imkanı sağlamaktadır (Maaten ve Hinton, 2008). Isomap ise verilerin geometrik yapısını koruyarak boyut azaltma işlemi gerçekleştirmektedir (Tenenbaum, De Silva, ve Langford, 2000). Bu yöntemler, özellikle büyük ve karmaşık veri setlerinde etkili bir şekilde kullanılmaktadır.

1.4. Sınıflandırma Modellerinin Çalışma Prensipleri

1.4.1. Random Forest (Rastgele Orman)

Random Forest, karar ağaçları topluluğuna dayanan güçlü bir denetimli öğrenme algoritmasıdır (Breiman, 2001). Bu model, çok sayıda karar ağacını

paralel olarak eğitip sonuçlarını birleştirerek çalışmaktadır. Random Forest, bootstrap örnekleme ile orijinal veri setinden rastgele örnekler alarak her ağaç için yeni eğitim veri setleri oluşturmaktadır. Öznitelik rastgeleleştirme tekniği ile her düğümde tüm öznitelikler yerine rastgele seçilen bir alt küme değerlendirilmektedir. Her ağaç, kendi bootstrap örneği üzerinde, öznitelik rastgeleleştirme kullanılarak maksimum derinliğe kadar büyütülmektedir. Sınıflandırma problemlerinde çoğunluk oylaması, regresyon problemlerinde ise ortalama alma yoluyla bireysel ağaçların tahminleri birleştirilmektedir. Random Forest, yüksek boyutlu verilerde iyi performans göstermekte, aşırı öğrenmeye karşı dirençli olmakta ve öznitelik önem derecelerini belirleyebilmektedir (Breiman, 2001).

1.4.2. Gradient Boosting (Gradyan Artırma)

Gradient Boosting, zayıf öğrenicileri (genellikle sığ karar ağaçları) sıralı bir şekilde ekleyerek model performansını aşamalı olarak artıran bir topluluk yöntemidir. Bu algoritmada, basit bir model ile başlanmakta ve mevcut modelin hataları (artıklar) hesaplanmaktadır. Bu artıkları tahmin etmek için yeni bir zayıf öğrenici eğitilmekte ve öğrenme oranı ile ölçeklendirilerek mevcut modele eklenmektedir. Ardından artıklar yeniden hesaplanmakta ve belirli bir iterasyon sayısına ulaşılana kadar bu süreç tekrarlanmaktadır. Gradient Boosting, yüksek tahmin doğruluğu sağlamakta ancak hiperparametre ayarlarına karşı duyarlı olmakta ve aşırı öğrenme riski taşıyabilmektedir (Friedman, 2001).

1.4.3. Extra Trees (Extremely Randomized Trees)

Extra Trees, Random Forest'a benzer ancak rastgeleleştirmeyi bir adım ileri taşıyan bir topluluk yöntemidir (Geurts, Ernst, & Wehenkel, 2006). Extra Trees, genellikle tüm orijinal eğitim setini her ağaç için kullanmakta ve bootstrap örnekleme yapmamaktadır. En önemli farkı, her öznitelik için en iyi bölme noktasını seçmek yerine, tamamen rastgele bölme noktaları belirlemesidir. Random Forest gibi, her düğümde rastgele seçilen bir öznitelik alt kümesi

değerlendirilmektedir. Bu ekstra rastgeleleştirme, hesaplama verimliliğini artırmakta ve genellikle aşırı öğrenmeyi daha da azaltmaktadır. Extra Trees, gürültülü verilerde iyi performans gösterebilmekte ve eğitimi Random Forest'a göre daha hızlı olmaktadır (Geurts vd., 2006).

1.4.4. HistGradient Boosting (Histogram-tabanlı Gradyan Artırma)

HistGradient Boosting, Gradient Boosting'in daha hızlı ve verimli bir versiyonudur (Pedregosa vd., 2011; Ke vd., 2017). Özellikle büyük veri setlerinde performans artışı sağlamaktadır. Bu model, sürekli değişkenleri eğitim sırasında binlere ayırarak histogramlar oluşturmakta ve bölme noktalarını belirleme işlemini bu histogramlar üzerinden yapmaktadır. Dahili olarak eksik değerleri ele alabilmekte ve ayrı bir önişleme gerektirmemektedir. Doğrulama setindeki performans artışı belirli bir eşiğin altına düştüğünde eğitimi durdurabilmektedir. Seviye-bilge büyüme yerine her seferinde en yüksek kazancı sağlayan yaprak düğümü bölmeyi tercih etmektedir. HistGradient Boosting, büyük veri setlerinde XGBoost veya LightGBM'e benzer performans göstermekte ve Scikit-learn kütüphanesinde doğrudan erişilebilmektedir (Pedregosa vd., 2011).

1.4.5. AdaBoost (Adaptive Boosting)

AdaBoost, ilk popüler boosting algoritmalarından biridir ve zayıf öğrenicileri (genellikle karar kütükleri) sıralı bir şekilde birleştirmektedir (Freund ve Schapire, 1997). Bu algoritma, başlangıçta tüm eğitim örneklerine eşit ağırlık atamakta ve mevcut örnek ağırlıklarını kullanarak bir zayıf öğrenici eğitmektedir. Zayıf öğrenicinin hata oranı hesaplanmakta ve bu hata oranına dayalı olarak model ağırlığı belirlenmektedir. Yanlış sınıflandırılan örneklerin ağırlıkları artırılmakta, doğru sınıflandırılanların ağırlıkları azaltılmaktadır. Belirli bir iterasyon sayısına veya yeterli performansa ulaşılana kadar bu süreç tekrarlanmaktadır. Final tahmin aşamasında, tüm zayıf öğrenicilerin tahminleri, model ağırlıklarına göre ağırlıklandırılmış oylama ile birleştirilmektedir. AdaBoost, özellikle gürültüsüz verilerde iyi performans göstermekte, ancak aykırı

değerlere karşı duyarlı olmakta ve karmaşık veri setlerinde diğer boosting algoritmaları kadar iyi performans göstermeyebilmektedir (Freund ve Schapire, 1997).

1.4.6. XGBoost (Extreme Gradient Boosting)

XGBoost, gradient boosting çerçevesinin yüksek performanslı ve optimize edilmiş bir uygulamasıdır (Chen ve Guestrin, 2016). Bu model, aşırı öğrenmeyi önlemek için L1 (Lasso) ve L2 (Ridge) düzenlileştirme tekniklerini kullanmaktadır. Temel öğrenici olarak CART (Classification and Regression Trees) ağaçlarını kullanmakta ve her yaprağa bir sürekli skor atamaktadır (Chen ve Guestrin, 2016). Kayıp fonksiyonunu ikinci dereceden Taylor açılımı ile yaklaşık olarak hesaplayarak optimizasyon sürecini hızlandırmaktadır. Kullanıcılar tarafından tanımlanabilen kayıp fonksiyonlarını desteklemektedir. Ağaç oluşturma sırasında çok çekirdekli işlem yapabilmekte ve veri setindeki eksik değerleri otomatik olarak ele alabilmektedir. Önce maksimum derinliğe kadar ağaç büyütüp sonra negatif kazanç sağlayan dalları geriye doğru budamaktadır (Chen ve Guestrin, 2016). XGBoost, birçok veri bilimi yarışmasında kazanan modellerin parçası olarak popülerlik kazanmıştır ve özellikle yapılandırılmış/tablo verilerinde üstün performans göstermektedir (Nielsen, 2016).

1.4.7. LightGBM (Light Gradient Boosting Machine)

LightGBM, Microsoft tarafından geliştirilen, özellikle büyük veri setlerinde hızlı ve verimli çalışmak üzere tasarlanmış bir gradient boosting çerçevesidir (Ke vd., 2017). Sürekli özellikleri ayrık bölmelere ayırarak histogram-tabanlı algoritmalar kullanmakta ve hesaplama verimliliğini artırmaktadır. Geleneksel seviye-bilge yaklaşım yerine, her seferinde en büyük kazanç sağlayan yaprağı bölen yaprak-bilge büyüme stratejisi kullanmaktadır (Ke vd., 2017). Bu, daha derin, asimetrik ağaçlar oluşturarak daha iyi doğruluk sağlamaktadır. GOSS (Gradient-based One-Side Sampling) tekniği ile örnek

sayısını azaltmak için yüksek gradyanlı örneklere odaklanıp düşük gradyanlı örnekleri rastgele almaktadır. EFB (Exclusive Feature Bundling) yöntemi ile özellik sayısını azaltmak için yüksek korelasyonlu ve seyrek özellikleri birleştirmektedir. Kategori özelliklerini doğrudan desteklemekte ve bu özellikleri verimli bir şekilde ele almaktadır (Ke vd., 2017). LightGBM, XGBoost'a benzer doğruluk sağlarken çok daha az bellek kullanımı ve daha hızlı eğitim süresi sunmaktadır, özellikle milyon veya milyarlarca örnek içeren veri setlerinde etkili olmaktadır.

1.4.8. SVM (Support Vector Machine - Destek Vektör Makinesi)

SVM, yüksek boyutlu uzaylarda veri noktalarını sınıflandırmak için kullanılan güçlü bir algoritmadır. Bu algoritma, sınıfları ayıran optimal bir hiper-düzlem (düz çizgi, düzlem veya yüksek boyutlu eşdeğeri) bulmayı amaçlamaktadır. Hiper-düzlemi her iki sınıfın en yakın veri noktalarından (destek vektörleri) maksimum uzaklıkta olacak şekilde konumlandırmakta ve böylece marjin maksimizasyonu sağlamaktadır. Kernel hilesi (Kernel Trick) ile doğrusal olarak ayrılamayan verileri daha yüksek boyutlu bir uzaya dönüştürerek orada doğrusal olarak ayrılabilir hale getirmektedir. Yaygın kernel fonksiyonları arasında doğrusal kernel, polinomiyal kernel, Radyal Tabanlı Fonksiyon (RBF) kernel ve sigmoid kernel bulunmaktadır. C parametresi ile yumuşak marjin uygulanarak, bazı örneklerin yanlış sınıflandırılmasına izin verilmekte ve gürültülü verilerde daha iyi genelleme yapılabilmektedir. SVM, sınıflandırmanın yanı sıra regresyon (SVR - Support Vector Regression) için de kullanılabilmektedir. Bu model, özellikle yüksek boyutlu verilerde, örnek sayısından daha fazla özellik olduğunda ve sınıfların belirgin sınırları olduğu durumlarda iyi performans göstermektedir.

1.4.9. Logistic Regression (Lojistik Regresyon)

Adına rağmen Logistic Regression bir sınıflandırma algoritmasıdır ve özellikle ikili sınıflandırma problemleri için kullanılmaktadır. Temelde doğrusal

bir model olup, girdi özelliklerinin doğrusal kombinasyonunu hesaplamaktadır. Doğrusal kombinasyon sonucu, bir sınıfa ait olma olasılığına dönüştürmek için sigmoid fonksiyonundan geçirilmektedir. Model parametreleri, log olabilirlik fonksiyonunu maksimize edecek şekilde optimize edilmektedir. Aşırı öğrenmeyi önlemek için L1 (Lasso) veya L2 (Ridge) düzenlileştirme teknikleri kullanılabilmektedir. İkili sınıflandırma için tasarlanmış olmasına rağmen, "one-vs-rest" veya "multinomial" yaklaşımlarıyla çok sınıflı problemlere genelleştirilebilmektedir. Logistic Regression, yorumlanabilirliği, eğitim hızı ve olasılıksal çıktıları nedeniyle popülerdir. Ayrıca, özellik önemi hakkında doğrudan bilgi sağlamakta ve yüksek boyutlu, seyrek veri setlerinde iyi performans gösterebilmektedir (Hosmer, Lemeshow, & Sturdivant, 2013; James, Witten, Hastie, & Tibshirani, 2021; Kleinbaum & Klein, 2010; Ng, 2004; Bishop, 2006).

1.5. Derin Öğrenme Modellerinin Çalışma Prensipleri

1.5.1. CNN (Convolutional Neural Network - Evrişimli Sinir Ağı)

CNN, özellikle görüntü işleme ve tanıma görevlerinde kullanılan ileri düzey bir derin öğrenme mimarisidir. Bu model, insanların görsel korteksinden esinlenen bir yapıya sahip olmaktadır. CNN'lerin çalışma prensibi, görsel verilerdeki uzamsal ilişkileri yakalamak üzere tasarlanmış özel katmanlar içermektedir. Temel bileşenleri arasında evrişim katmanları, havuzlama katmanları ve tam bağlantılı katmanlar bulunmaktadır. Evrişim katmanlarında, giriş verisi üzerinde çeşitli filtreler (veya çekirdekler) kaydırılmakta ve her konumda bir iç çarpım hesaplanarak özellik haritaları oluşturulmaktadır. Bu filtreler, eğitim sırasında kenarlar, dokular ve daha karmaşık görsel özellikler gibi önemli özellikleri otomatik olarak öğrenmektedir. Havuzlama katmanları, özellik haritalarının boyutsal indirgenmesini sağlayarak hesaplama verimliliğini artırmakta ve aynı zamanda konumsal değişimlere karşı dayanıklılık kazandırmaktadır. En yaygın havuzlama yöntemi olarak maksimum havuzlama

(max pooling) kullanılmakta, bu işlemde belirli bir pencere içindeki en yüksek değer seçilmektedir. Aktivasyon fonksiyonları, genellikle ReLU (Rectified Linear Unit), ağa doğrusal olmayan özellikler kazandırmaktadır. Derin CNN mimarileri, bu temel katmanların çok sayıda tekrarlanmasıyla oluşturulmaktadır. Son olarak, tam bağlantılı katmanlar, çıkarılan özellikleri kullanarak sınıflandırma veya regresyon görevlerini gerçekleştirmektedir. Modern CNN mimarileri arasında AlexNet, VGGNet, GoogLeNet, ResNet ve EfficientNet gibi modeller bulunmakta, bunlar görüntü sınıflandırma, nesne tespiti, segmentasyon ve yüz tanıma gibi çeşitli görevlerde kullanılmaktadır. CNN'ler ayrıca görüntü işlemenin yanı sıra, ses tanıma, doğal dil işleme ve zaman serisi analizi gibi alanlarda da uygulanabilmektedir.

1.5.2. LSTM (Long Short-Term Memory - Uzun-Kısa Vadeli Bellek)

LSTM, zaman serisi verileri ve sıralı verileri işlemek için tasarlanmış özel bir tekrarlayan sinir ağı (Recurrent Neural Network - RNN) türüdür (Hochreiter ve Schmidhuber, 1997). Standart RNN'lerin uzun vadeli bağımlılıkları öğrenme konusundaki sınırlamalarını aşmak için geliştirilmiştir. LSTM'leri özel kılan, uzun süreli bilgileri hatırlama ve ilgisiz bilgileri unutma yeteneğidir. LSTM hücreleri, üç kapıdan oluşan karmaşık bir iç yapıya sahiptir: unutma kapısı, giriş kapısı ve çıkış kapısı. Unutma kapısı, hücre durumundan hangi bilgilerin atılacağına karar vermektedir. Sigmoid fonksiyonu kullanarak 0 ile 1 arasında değerler üreterek, önceki hücre durumundaki her bilgi parçasının ne ölçüde korunacağını belirlemektedir. Giriş kapısı, hangi yeni bilgilerin hücre durumuna ekleneceğini kontrol etmektedir. Bu kapı, iki bölümden oluşmaktadır: sigmoid katmanı hangi değerlerin güncelleneceğini belirlemekte ve tanh katmanı yeni aday değerler oluşturmaktadır. Çıkış kapısı, hücre durumunun hangi kısımlarının çıktı olarak verileceğini kontrol etmektedir. Sigmoid fonksiyonu, hücre durumunun hangi kısımlarının çıktıya gideceğini belirlemekte ve tanh fonksiyonu, hücre durumunu -1 ile 1 arasında ölçeklendirmektedir. Bu kapı yapısı, LSTM'lerin uzun vadeli bağımlılıkları öğrenmesini sağlamakta ve gradyan kaybı veya patlaması sorunlarını hafifletmektedir (Hochreiter ve Schmidhuber, 1997). LSTM'ler, doğal dil işleme, konuşma tanıma, metin oluşturma, makine çevirisi, duygu analizi, zaman serisi tahmini, müzik kompozisyonu ve video analizi gibi sıralı verilerin

işlenmesini gerektiren çeşitli uygulamalarda yaygın olarak kullanılmaktadır (Lipton vd., 2015). LSTM'lerin varyantları arasında, hesaplama verimliliği için daha basit bir mimari kullanan GRU (Gated Recurrent Unit) (Cho vd., 2014) ve çift yönlü LSTM (Bidirectional LSTM) bulunmaktadır. Çift yönlü LSTM'ler, bir sıralı veriyi hem ileri hem de geri yönde işleyerek, gelecekteki ve geçmişteki bağlamı birleştirmektedir (Schuster ve Paliwal, 1997).

1.6. Ensemble Learning (Topluluk Öğrenmesi)

Ensemble Learning, makine öğrenmesi modellerinin doğruluk oranını ve genelleştirme kapasitesini artırmak amacıyla birden fazla modelin birlikte kullanılmasını sağlayan bir yöntemdir (Zhou, 2012). Tek bir modelin sınırlılıklarını ve hata oranlarını azaltmak için farklı modellerin çıktıları birleştirilmekte ve daha güvenilir tahminler elde edilmektedir. Makine öğrenmesi algoritmaları, veri setlerindeki belirli örüntüleri öğrenirken çeşitli avantajlar ve dezavantajlar barındırmaktadır. Örneğin, bazı modeller aşırı öğrenme (overfitting) eğilimi gösterirken, bazıları ise genelleme konusunda yetersiz kalabilmektedir. Ensemble Learning, farklı algoritmaların tahminlerini bir araya getirerek model hatalarını en aza indirmeyi ve tahmin performansını artırmayı amaçlamaktadır. Bu yöntemin temel ilkesi, tek bir model yerine birden fazla modelin kullanılmasıyla sistemin dayanıklılığının ve doğruluk seviyesinin yükseltilmesidir (Dietterich, 2000). Özellikle büyük veri setleri, karmaşık veri yapıları ve makine öğrenmesi yarışmaları gibi alanlarda yaygın olarak tercih edilmektedir.

Ensemble Learning'in başarısının temelinde modellerin çeşitliliği ve bağımsızlığı yatmaktadır. Kullanılan modellerin birbirinden farklı özelliklere sahip olması durumunda hata oranı azalmaktadır ve daha yüksek doğruluk seviyesine ulaşılmaktadır (Opitz ve Maclin, 1999). Bu yöntem, dört ana teknik ile uygulanmaktadır: **Bagging**, **Boosting**, **Stacking** ve **Blending**.

Stacking, farklı türde makine öğrenmesi modellerinin tahminlerini birleştirerek daha güçlü bir model oluşturmayı amaçlayan bir ensemble öğrenme tekniği olarak öne çıkmaktadır (Wolpert, 1992). Bagging ve boosting yöntemlerinden farklı olarak, stacking modeli çeşitliliğini artırmaya odaklanmakta ve genellikle farklı algoritmaların bir arada kullanılmasıyla uygulanmaktadır. Bu yöntemin temel çalışma prensibi, farklı algoritmaların (örneğin, karar ağaçları, lojistik regresyon, destek vektör makineleri ve

XGBoost gibi) aynı eğitim veri seti üzerinde eğitilmesi ve her modelin kendi tahminlerini üretmesi üzerine kuruludur. Daha sonra, bu tahminler bir araya getirilerek ikinci bir model **(**meta-learner veya üst model**)** tarafından analiz edilmekte ve nihai tahmin yapılmaktadır.

Meta-model, alt modellerin ürettiği tahminleri giriş verisi olarak almakta ve bu veriler üzerinden daha yüksek doğruluk oranına sahip bir çıktı üretmeye çalışmaktadır (Wolpert, 1992).

Stacking yönteminin en büyük avantajı, farklı modellerin güçlü yönlerinden yararlanarak tahmin performansını artırmasıdır. Örneğin, belirli bir model belirli veri türlerinde yüksek doğruluk sağlarken, farklı bir model başka türdeki verilerde daha iyi performans gösterebilmektedir. Stacking, bu farklı modellerin avantajlarını bir araya getirerek daha güçlü ve daha genelleştirilebilir bir model oluşturmaktadır (Sill vd., 2009). Ancak, bu yöntemin hesaplama maliyeti yüksek olmakta ve dikkatli hiperparametre ayarları gerektirmektedir. Doğru şekilde uygulandığında, stacking yöntemi, makine öğrenmesi modellerinin tahmin başarısını önemli ölçüde artırabilmektedir.

2. LİTERATÜR TARAMASI

Sun ve arkadaşları (2022) tarafından yapılan araştırmada, destek vektör makineleri (SVM) modeli, probiyotik türlerin DNA dizilimlerinden sınıflandırılması için uygulanmıştır. Model, özellik uzayındaki karmaşıklığı azaltmak için boyut indirgeme algoritmalarıyla desteklenmiştir. Bu yöntem, özellikle bağırsak mikrobiotasında türlerin tespitinde kesinlik düzeyini artırmış ve sınıflandırma işlemini hızlandırmıştır. Ayrıca, bağırsak mikrobiyotasının yapısındaki heterojenlik modelin eğitimi sırasında dikkate alınmıştır.

Ayoola ve arkadaşları (2023) tarafından yapılan araştırmada, rastgele ormanlar (random forests) algoritması, probiyotik ve patojen türlerin çevresel faktörlere göre etkileşimlerini tahmin etmek için uygulanmıştır. Model, çevresel verilerin mikrobiyal ekosistem üzerindeki etkilerini analiz etmek için regresyon teknikleri ile entegre edilmiştir. Örneğin, sıcaklık ve pH seviyelerindeki değişikliklerin probiyotik çeşitliliği

üzerindeki etkisi tahmin edilmiştir. Çalışma, hem mikrobiyal dengenin korunması hem de zararlı etkilerin en aza indirgenmesi adına öneriler geliştirmiştir.

Wu ve arkadaşları (2024) tarafından yapılan araştırmada, probiyotiklerin metagenomik verilerden çıkarımını kolaylaştırmak için karar ağaçları ve XGBoost algoritmaları uygulanmıştır. Karar ağaçları, probiyotik biyobelirteçlerin sınıflandırılmasında temel yapı taşlarını oluştururken, XGBoost algoritması, büyük ve karmaşık veri kümelerindeki örüntüleri daha doğru bir şekilde modellemiştir. Çalışma, özellikle metagenomik veri analizi sırasında biyolojik gürültünün etkilerini azaltmaya odaklanmıştır.

Shakibania ve arkadaşları (2024) tarafından yapılan çalışamada , konakçı-patojen etkileşimlerini modellemek için convolutional neural networks (CNNs) uygulanmıştır. CNN modeli, metin tabanlı biyobelirteçler ve mikrobiyal görüntü verileri gibi çok boyutlu verilerle eğitilmiştir. Bu, probiyotik türlerin patojenlere karşı koruyucu etkilerini anlamada önemli bir araç olmuştur. Model, biyolojik verilerin uzamsal ilişkilerini çıkararak sınıflandırma doğruluğunu artırmıştır.

Marcos-Zambrano ve arkadaşları (2023) tarafından yapılan araştırmada, gözetimsiz makine öğrenimi araçlarından k-means ve hiyerarşik kümeleme yöntemleri uygulanmıştır. Mikrobiyom gruplarını sınıflandırmak için, verilerden çıkarılan metriklerle çok boyutlu benzerlik analizleri yapılmıştır. Özellikle, bağırsak mikrobiyotasındaki farklılıkları belirlemek için kullanılan bu yöntemler, hasta grupları arasındaki varyasyonları anlamada etkili olmuştur.

Venkatesh ve arkadaşları (2024) tarafından yapılan araştırmada, kişiselleştirilmiş probiyotik ürün geliştirme amacıyla, Bayes ağları ve markov modelleri uygulanmıştır. Bu algoritmalar, bireylerin mikrobiyom profiline ve genetik yapılarına dayanarak probiyotik kombinasyonlarını optimize etmek için uygulanmıştır. Çalışma, öneri sistemlerinin güvenilirliğini artırmak için genetik ve yaşam tarzı verilerini harmanlamıştır.

Shakibania ve arkadaşları (2022) tarafından yapılan araştırmada, çevresel faktörlerin mikrobiyom içindeki probiyotik türlerin dağılımına olan etkisini analiz etmek için destek vektör makineleri (SVM) uygulanmıştır. Model, özellikle sıcaklık, nem ve diyet değişikliklerinin probiyotik tür çeşitliliği üzerindeki etkisini tahmin etmek üzere yapılandırılmıştır.

Talon ve ark. (2002), yoğurttaki Streptococcus salivarius subsp. thermophilus ve Lactobacillus delbrueckii subsp. bulgaricus popülasyonlarının miktarlarının belirlenmesi amacıyla yaptıkları araştırmada; beraber kullandıkları piroliz-kütle spektrometrisi (PyMS) ve yapay sinir ağları (YSA) tekniklerinin, yoğurttaki bu iki bakteri türünün hızlı ve doğru bir şekilde belirlenmesinde etkili bir yöntem olduğunu belirtmişlerdir.
Zhang ve ark. (2021), yaptıkları araştırmada görüntü analizi tabanlı mikroorganizma sayımında yapay zeka tekniklerinin kullanıldığında daha etkili ve hızlı sonuçlar verdiğini ortaya koymuştur.

Dziuba (2013), Fourier dönüşümü kızılötesi spektroskopisi (FTIR) ve yapay sinir ağları (YSA) kullanarak Propionibacterium türlerini %93 doğrulukla tanımlamıştır. Bu çalışma, FTIR spektrumlarının YSA ile analiz edilmesinin mikroorganizma sınıflandırmasında etkili bir yöntem olduğunu göstermiştir.

Ma ve ark. (2022), yapay zeka ve optik görüntüleme kullanarak gıdalardaki bakterilerinin tespitini hızlandırma konulu yaptıkları araştırmada, Escherichia coli(E. coli) tespitinde derin öğrenme modeli olan gerçek zamanlı nesne algılama ve sınıflandırma algoritması YOLOv4(You Only Look Once, Version 4) kullanıldığında %94 hassasiyet ile hızlı bir şekilde E. coli yi tespit ettiği gözlenmiştir.

Kumar ve ark. (2024), gıda kaynaklı hastalıklardaki patojenlerin(Salmonella, Escherichia coli, Campylobacter, Clostridium and Listeria) tahmini için kullanılan yöntemleri, karar ağaçları, rastgele ormanlar (random forests), k-En Yakın Komşu (k-Nearest Neighbors), stokastik gradyan inişi (stochastic gradient descent) ve son derece rastgele ağaçlar (extremely randomized trees) gibi teknikler ile tüm bu yaklaşımları birleştiren bir topluluk (ensemble) modeliyle karşılaştırmıştır. Deneysel sonuçlar, önerilen yeni topluluk modelinin diğer sınıflandırıcılardan daha iyi performans gösterdiğini ortaya koymuştur. Bu model, ortalama olarak %97,26 doğruluk, 0,22 RMSE, %97,77 geri çağırma, %97,66 kesinlik ve %98,44 F1 skoru ile en yüksek performansı elde etmiştir.

Pan ve ark. (2024), tarımsal mikroorganizmalar için oluşturulan ilk veri kümesi olan SMAD' ı tanıtarak, havada taşınan mikroorganizmaların tespiti için bir açık küme tanıma yöntemi geliştirmiştir. Önerilen SPARC (Spatial Attention on Region Hierarchy)

modülü, yerel ve küresel özellikleri birleştirerek görüntü bozulmalarını azaltmış ve bilinmeyen sınıfların doğru şekilde sınıflandırılmasını sağlamıştır.

Park ve ark. (2023), deniz mikroorganizmalarının metagenomik sınıflandırmasında derin öğrenme modellerinin potansiyelini değerlendirilmiş olup DeepMicrobes ve ResNet mimarileri kullanılarak mevcut sınıflandırma araçlarına alternatif yöntemler önerilmiştir.

Smith ve arkadaşlarının (2020) çalışmasında, yapay zekânın (YZ) klinik mikrobiyoloji laboratuvarlarına entegrasyonu detaylı bir şekilde incelenmiş olup YZ ve makine öğrenimi (MÖ) tekniklerinin, özellikle görüntü analizi görevlerinde tanı süreçlerini nasıl iyileştirebileceğini araştırılmıştır. Örneğin, Gram boyama analizlerinde, Evrişimsel Sinir Ağları (Convolutional Neural Networks - CNN) kullanılarak kan kültürü Gram boyama preparatlarının otomatik yorumlanması üzerine bir çalışma sunulmuştur. Bu çalışmada, CNN modeli, 1000 farklı nesneyi ayırt edebilecek şekilde eğitilmiş ve bu eğitim süreci yoğun bir işlem gerektirmiştir. Ayrıca, parazitoloji alanında, morfolojik analizlerin hala tercih edilen tanı yöntemi olduğu ve YZ'nin bu alanda da uygulanabilirliği vurgulanmıştır. Bakteriyel kültür plakalarının dijital görüntü analizi için YZ'nin kullanımı ele alınmış ve bu alanda otomasyonun iş akışı verimliliğini ve kaliteyi artırdığı belirtilmiştir. MALDI-TOF MS verilerinin ileri analizlerinde YZ'nin uygulanması, bu teknolojinin klinik mikrobiyolojiye "kara kutu" teşhis konseptini tanıttığı ve YZ'nin bu alanda doğal bir sonraki adım olduğu ifade edilmiştir. Son olarak, YZ ve MÖ yöntemlerinin, mikroorganizmaların tüm genom dizileme verilerinin analizinde, özellikle antimikrobiyal direnç tahmininde umut verici olduğu belirtilmiştir.

Derin öğrenme temelli bir yaklaşımda, metagenomik verilerden 16S rRNA genlerine dayalı olarak bakterilerin sınıflandırılması için konvolüsyonel sinir ağları (CNN) ve derin inanç ağları (DBN) uygulanmıştır. Bu modeller, sınıftan cins seviyesine kadar sınıflandırmaya izin vererek, farklı ekosistemlerdeki bakteriyel toplulukları doğru bir şekilde sınıflandırmıştır (Fiannaca ve ark., 2018)

U-Net derin öğrenme modeli, Rift Valley virüsünün sınıflandırılması ve segmentasyonu için uygulanmıştır. Bu model, görüntü analizinde üstün bir performans göstermiş ve Dice skoru %90, IOU skoru %83,1 olarak rapor edilmiştir (Matuszewski ve

Sintorn, 2018). Transfer öğrenme tekniği kullanılarak bakteri sınıflandırması gerçekleştirilmiştir. Xception modeli, insanlar için ölümcül olabilecek yedi bakteri çeşidinin sınıflandırılmasında %97,5 doğrulukla başarı elde etmiştir (Wahid ve ark., 2019).

Cordovana ve arkadaşları (2022), Fourier-Transform Infrared (FTIR) Spektroskopisi tabanlı IR Biotyper sistemi kullanarak Salmonella enterica O-serogrup tiplendirmesi için makine öğrenmesine dayalı bir yöntem geliştirmişlerdir. 958 Salmonella izolatını içeren geniş bir veri seti üzerinde yapılan testler sonucunda, sistemin %94,7 ile %99,2 arasında değişen doğruluk oranlarıyla başarılı sonuçlar verdiği görülmüştür. Bu sonuçlar, FTIR tabanlı IR Biotyper sisteminin, serogrup seviyesinde Salmonella tiplendirmesi için umut verici ve kullanıcı dostu bir araç olduğunu göstermektedir. Makine öğrenmesi algoritmalarının uygulanmasıyla, insan kaynaklı hatalardan arındırılmış otomatik bir analiz ve sonuç yorumlama süreci geliştirilmiştir (Cordovana ve ark., 2022).

Alakuş (2023), mikroorganizmaların sınıflandırılmasında tekrarlayıcı sinir ağları (TSA) ve uzun/kısa süreli bellek (LSTM) modellerinin kullanılmasını ele almaktadır. Çalışma dört aşamalı bir süreç izlemektedir: veri toplama, önişleme, model tasarımı ve sınıflandırma. TSA ile %92,53, LSTM ile %99,85 doğruluk skorları elde edilmiştir. Bu çalışma, mikroorganizmaların sınıflandırılmasında derin öğrenme yöntemlerinin etkinliğini ortaya koymakta ve mikroorganizmaların özelliklerini anlamada önemli bir katkı sağlamaktadır.

Ergüven ve Ökten (2022), yaptıkları araştırmada, bakteri tanı ve teşhisinde yapay zeka teknolojilerinin kromojenik agar üzerinde bakteri kolonilerinin sayılması ve sınıflandırılması konusunda manuel yöntemlere göre daha kısa sürede sonuç verdiğini göstermiştir. Ayrıca, mikrobiyom analizlerinde yapay zeka, bakterilerin taksonomik olarak sınıflandırılmasına ve tedavi yöntemlerinin belirlenmesine katkı sağlamıştır. Virüslerin tanı ve teşhisinde, özellikle COVID-19 pandemisi döneminde, yapay zeka modelleri, yüksek doğruluk oranları ile teşhis süreçlerini hızlandırmış ve biyometrik veriler üzerinden COVID-19 varlığını tespit edebilen sistemler geliştirilmiştir. Mantar ve parazitlerin tanı süreçlerinde ise yapay zeka kullanımı, tanı sürecini hızlandırmış ve biyokimyasal tanımlama gereksinimlerini azalttığı belirtilmiştir.

Kandilci, Yakıcı ve Kayar (2024), yapay zekanın mikroorganizma tanı ve teşhisinde önemli rol oynadığını, özellikle görüntü analizi destekli yöntemlerin hızlı ve doğru teşhis sağladığını belirtmektedir. Ayrıca, makine öğrenimi ve derin öğrenme gibi yapay zekanın alt dallarının, mikroorganizmaların sınıflandırılmasında ve enfeksiyon kontrolü süreçlerinde etkin bir şekilde kullanıldığı vurgulanmıştır.

Graf ve ark. (2024), yapay zekanın mikrobiyoloji laboratuvarlarında slayt tabanlı görüntü analizlerinde kullanılabileceğini öne sürmüşlerdir. Çalışmalarında, Gram boyama yorumlaması ve kültür büyümesi gibi analizlerde yapay zeka tabanlı algoritmalar kullanılmış ve %90'ın üzerinde bir doğruluk elde edilmiştir. Bu bulgu, yapay zekanın tanısal süreçlerde insan hatasını büyük ölçüde azaltabileceğini göstermektedir. Çalışmada, kültür plaklarındaki mikroorganizmaların sayısının ve türlerinin YZ algoritmalarıyla belirlenmesi amaçlanmıştır. Random Forest algoritması kullanılarak yapılan analizde, manuel yöntemlere göre %95 oranında daha hızlı sonuçlar elde edilmiştir. Bu, yapay zekanın laboratuvar verimliliğini önemli ölçüde artırabileceğini ortaya koymaktadır.

Liu ve ark. (2021), dondurularak kurutulan kalp kapakçıklarında oksidatif hasarın tespiti amacıyla Fourier Dönüşüm Kızılötesi Spektroskopisi (FTIR) ve yapay sinir ağı (YSA) sınıflandırma modellerinin birleştirilmesiyle çalışma yapılmıştır. Decellularized domuz aortik kalp kapakçıkları, dondurma işlemi öncesinde ve sonrasında nitroblue tetrazolium (NBT) boyama yöntemi ve FTIR kullanılarak analiz edilmiştir. Ayrıca, sukrozun (şeker) dokuları oksidatif hasardan koruma potansiyeli de incelenmiştir. Sonuçlar, %40 sukroz çözeltisinin oksidatif hasarı önemli ölçüde azalttığını ve depolamanın biyokimyasal yapıda ciddi değişiklik yaratmadığını göstermiştir. Yapay sinir ağı modeli, H_2O_2 ve $FeCl_3$ ile işlem görmüş numunelerde oksidatif hasarı %100 doğrulukla sınıflandırabilmiştir. Ancak, taze, dondurulmuş ve saklanmış numuneler arasındaki biyokimyasal farklılıkların minimal olması nedeniyle bu grupların sınıflandırılması daha zorlu olmuştur.

Fourier Transform Infrared (FTIR) Spektroskopisi, spektral yorumlamanın zaman alıcı bir süreç olmasına rağmen, bileşiklerdeki işlevsel grupların belirlenmesinde önemli bir tekniktir. Enders ve arkadaşlarının çalışmasında (Enders ve ark., 2024), FTIR spektrumlarının daha hızlı ve etkili bir şekilde analiz edilmesi amacıyla Evrişimli Sinir

Ağları (CNN'ler) tabanlı bir makine öğrenimi modeli geliştirilmiştir. Bu araştırmada, NIST spektral veri tabanından elde edilen gaz fazındaki organik moleküllerin spektrumları, görüntü formatına dönüştürülmüş ve işlevsel grupların tespiti için CNN modelleri başarılı bir şekilde eğitilmiştir. Söz konusu modeller, FTIR analizlerinin işlevsel grup tespiti konusunda daha geniş bir uygulama alanı kazanmasına katkı sağlamıştır.

Sofu ve arkadaşlarının 2007 yılında yayımladıkları "Gıda Bilimi ve Teknolojisi Alanında Yapay Zekâ Uygulamaları" başlıklı makalede, yapay zekâ ve bulanık mantık tekniklerinin gıda mikrobiyolojisinde mikroorganizmaların sınıflandırılması, gıda güvenliği ve kalite kontrolü ile proses optimizasyonu gibi uygulamalarda kullanıldığı belirtilmektedir.

Escherichia coli O157:H7'nin sınıflandırılması üzerine yapılan bu çalışmada, bulanık mantık ve sinir ağlarının füzyonu ile tanıma oranları artırılmıştır (Wang vd., 1998). Bu yaklaşımlar, mikroorganizmaların tanımlanmasında makine öğrenmesi tabanlı yöntemlerin uygulanabilirliğini vurgulamaktadır.

Mikrobiyal sistemlerin davranışlarını modellemek için yapay zeka ve sibernetik modeller gibi alternatif yöntemler geliştirilmiştir (Patnaik, 2009).

3. GEREÇ VE YÖNTEM

Probiyotiklerin doğru ve etkili sınıflandırılması, sağlık ve endüstri alanlarında büyük bir öneme sahiptir. Bu çalışmada, *Enterococcus faecium, Lactobacillus plantarum ve Lactobacillus fermentum* türlerinin ayrımını yapmak için güçlü bir Convolutional Neural Network (CNN) modeli geliştirilmiştir. Bu modelin seçilme sebebi denenmiş olan tüm hibrit ve bireysel algoritmalar arasından en yüksek performansı CNN algoritmasının göstermiş olmasıdır. Bu projede kullanılmış olan *Enterococcus faecium, Lactobacillus plantarum ve Lactobacillus fermentum* türlerinin ölçümleri FTIR tekniği ile ölçülmüştür.

FTIR spektroskopisi, moleküllerin belirli frekanslardaki kızılötesi ışığı soğurmasına dayanmaktadır. Bu frekanslar, molekül içindeki farklı kimyasal bağların titreşimleriyle ilişkilidir.

Konvansiyonel spektroskopide, ışığın belirli dalga boylarında numune tarafından nasıl soğurulduğu incelenirken, FTIR spektroskopisinde ışığın tüm dalga boyları eşzamanlı olarak analiz edilir. Bu analiz Fourier dönüşümü (FT) adı verilen matematiksel bir işlemle gerçekleştirilir ve spektrum elde edilir.

FTIR cihazı aşağıdaki temel bileşenlerden oluşur:

- IR Işık Kaynağı: Geniş bir spektral aralıkta kızılötesi ışık üretir.
- Michelson İnterferometresi: FTIR cihazının temel bileşenidir. Hareketli ve sabit aynalar ile bir ışın ayırıcıdan oluşur.
- Dedektör: Gelen sinyali kaydeder ve bilgisayara iletir.
- Bilgisayar: Fourier dönüşümünü uygular ve nihai spektrumu oluşturur.

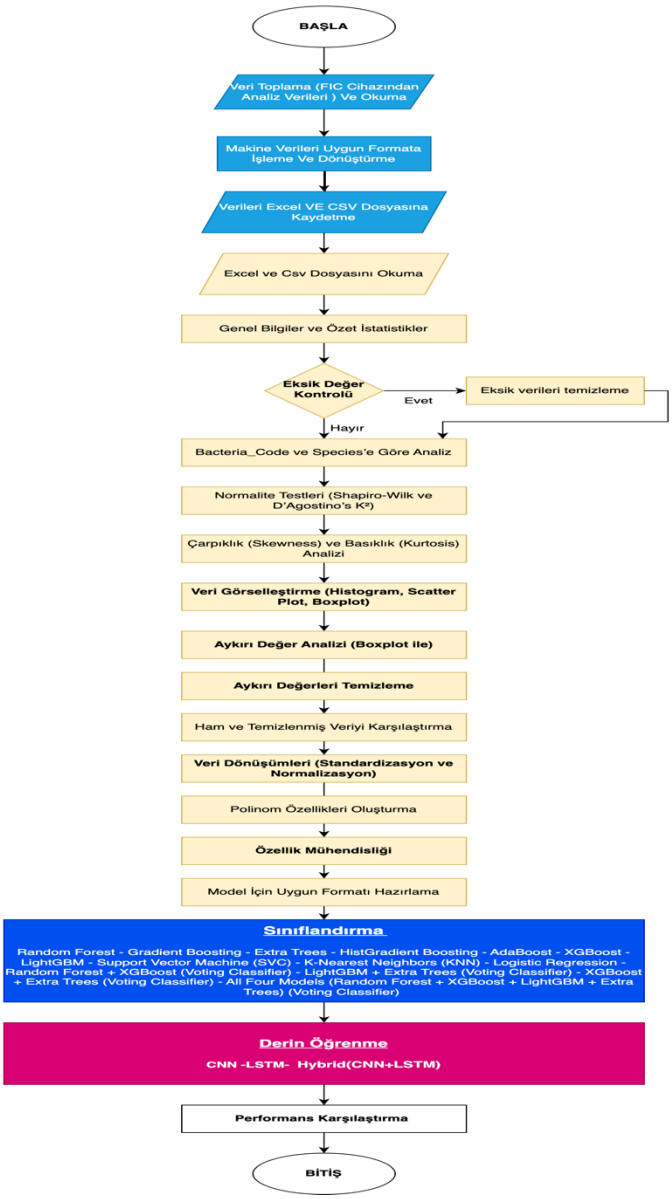
Ölçüm sırasında, kızılötesi ışık kaynağından çıkan ışın, Michelson interferometresinden geçerek numuneye yönlendirilir. Numune, belirli dalga boylarındaki ışığı soğurur ve geriye kalan ışık dedektör tarafından algılanır. Fourier dönüşümü ile zaman alanından frekans alanına çevrilen veriler, FTIR spektrumu olarak adlandırılan grafiksel bir çıktı haline getirilir.

FTIR spektrumu, numuneden geçen veya numune tarafından soğurulan ışığın dalga sayısı (cm^{-1}) ve soğurma şiddeti (transmittans veya absorbans) arasındaki ilişkiyi gösterir.

- Fonksiyonel Grup Bölgesi (4000-1500 cm^{-1}): Hidroksil (-OH), karbonil (C=O), amin (-NH) gibi karakteristik bağların bulunduğu bölgedir.
- Parmak İzi Bölgesi (1500-400 cm^{-1}): Her bileşik için benzersiz olan ve kimyasal yapıyı tanımlayan bölgedir.

Modelin başarısını artırmak amacıyla veri analizi, ön işleme, makine öğrenmesi ve derin öğrenme süreçlerinde geniş bir kütüphane yelpazesi kullanılmıştır. Bu projede, Prof. Dr. Gülden Başyiğit Kılıç'ın 2019 yılında Süleyman Demirel Üniversitesi'nde tamamladığı "Bazı Laktobasil Suşlarının Genetik Tanısının Yapılması ve Faj Dirençliliklerinin Belirlenmesi" başlıklı doktora tezindeki mikroorganizma analiz

sonuçları (**Fourier Dönüşümlü Kızılötesi Spektroskopisi (FTIR)** analizlerinden elde edilen toplam **108.275** veri noktası) kullanılmıştır. Bu veriler, *Enterococcus faecium* (44.375 veri noktası), *Lactobacillus plantarum* (37.275 veri noktası) ve *Lactobacillus fermentum* (26.625 veri noktası) türlerine aittir. Proje aşamasında yapılan işlemler sırasıyla Şekil 3.1. 'de belirtilmiştir.



Şekil 3.1. Proje Akış Şeması

Veri manipülasyonu ve görselleştirme için Pandas, NumPy, Matplotlib ve Seaborn gibi temel kütüphanelerden yararlanılmıştır. Verilerin istatistiksel analizi ve sinyal işleme aşamalarında SciPy fonksiyonları kullanılmış, önişleme ve dönüşüm

süreçleri Scikit-learn ile gerçekleştirilmiştir. Çeşitli makine öğrenmesi sınıflandırma ve kümeleme algoritmalarının yanı sıra, XGBoost ve LightGBM gibi güçlü modeller de değerlendirilmiştir. Derin öğrenme aşamasında ise TensorFlow/Keras ve PyTorch kullanılarak CNN modeli eğitilmiş ve optimize edilmiştir. Ayrıca, boyut azaltma ve özellik çıkarma teknikleri uygulanarak modelin verimliliği artırılmıştır.

Elde edilen sonuçlar, geliştirilen CNN modelinin %92 doğruluk oranıyla probiyotik sınıflandırmasında başarılı olduğunu göstermektedir. Bu çalışma, hem akademik hem de endüstriyel alanda probiyotik analizine yönelik yapay zeka tabanlı yeni bir yaklaşım sunarak literatürdeki boşluğu doldurmayı hedeflemektedir. Aşağıda bu projenin geliştirilmesinde kullanılmış olan tüm kütüphaneler, algoritmalar, algoritmaların karşılaştırılması, karşılaştırma sonuçları ve gerekli bilgiler tablolarla desteklenerek verilmiştir.

Proje geliştirme aşamasında kullanılan kütüphaneler hizmet ettikleri amaca göre:

- Temel Kütüphaneler
    - Pandas
    - Numpy
    - Matplotlib.pyplot
    - Seaborn
    - Time
- İstatiksel Analiz & Sinyal İşleme
    - scipy.stats içerisinden shapiro, normaltest, skew ve kurtosis
    - scipy.signal içerisinden savgol_filter ve find_peaks
- Önişleme & Veri Dönüşümü
    - sklearn.preprocessing içerisinden StandardScaler, MinMaxScaler, LabelEncoder
    - sklearn.model_selection içerisinden train_test_split
- Makine Öğrenmesi - Sınıflandırma Algoritmaları
    - sklearn.ensemble içerisinden RandomForestClassifier, GradientBoostingClassifier, ExtraTreesClassifier, HistGradientBoostingClassifier, AdaBoostClassifier
    - sklearn.svm içerisinden SVC
    - sklearn.neighbors içerisinden KNeighborsClassifier

    - sklearn.linear_model içerisinden LogisticRegression
    - sklearn.metrics içerisinden classification_report ve accuracy_score
- Makine Öğrenmesi - Kümeleme Algoritmaları,
    - sklearn.cluster içerisinden KMeans, DBSCAN ve AgglomerativeClustering
    - scipy.cluster.hierarchy içerisinden ndrogderam ve linkage
- Ekstra Makine Öğrenmesi Modelleri
    - xgboost içerisinden XGBClassifier
    - lightgbm içerisinden LGBMClassifier
    - imblearn.over_sampling içerisinden SMOTE
    - sklearn.ensemble içerisinden VotingClassifier
- Derin Öğrenme - Tensorflow & Keras
    - Tensorflow
    - tensorflow içerisinden keras
    - tensorflow.keras.models içerisinden Sequential
    - tensorflow.keras.layers içerisinden Dense, Dropout, LSTM, Conv1D, MaxPooling1D ve Flatten
    - tensorflow.keras.optimizers içerisinden Adam
- Derin Öğrenme - Pytorch
    - Torch
    - Torch.nn
    - Torch.optim
    - torch.utils.data içerisinden Dataset ve DataLoader
- Boyut Azaltma & Özellik Çıkarma
    - sklearn.decomposition içerisinden PCA ve TruncatedSVD
    - sklearn.manifold içerisinden TSNE ve Isomap

FTIR spektral verilerle mikroorganizma sınıflandırması için ön işleme ve analiz sonuçları :

Şekil 3.2. ’de görüldüğü üzere, veri setindeki eksik değerler kontrol edilerek, her sütundaki eksik değerlerin sayısı hesaplanmaktadır. Daha sonra, "Bacteria_Code" ve "Species" sütunlarına göre istatistiksel dağılım analizi yapılmakta ve her tür için benzersiz bakteri kodlarının sayısı hesaplanmaktadır. Ardından, "Y_Absorbans" sütunu üzerinde

türlere göre istatistiksel analiz yapılarak, betimsel istatistikler elde edilmektedir. Son olarak, "X_Dalga_Boyu" ve "Y_Absorbans" sütunları için normalite testleri (Shapiro-Wilk ve D'Agostino's K^2) uygulanarak, verilerin normal dağılıma uygun olup olmadığı kontrol edilmektedir. Bu analizler, veri setinin yapısını anlamak ve sonraki adımlar için temel oluşturmak amacıyla kullanılmaktadır.

```
# 4. Eksik Değer Kontrolü
missing_values = df_combined.isnull().sum()
print("\nEksik Değerler:")
print(missing_values)

# 5. Bacteria_Code ve Species'e Göre İstatistiksel Analiz
print("\nBacteria_Code ve Türlere Göre İstatistiksel Dağ□1□m:")
bacteria_species_stats = df_combined.groupby("Species")["Bacteria_Code"].nunique()
print(bacteria_species_stats)

# 6. Bacteria_Code ve Species'e Göre Y_Absorbans Analizi
print("\nBacteria_Code ve Türlere Göre Y_Absorbans İstatistiksel Analizi:")
bacteria_species_absorbance_stats = df_combined.groupby("Species")["Y_Absorbans"].describe()
print(bacteria_species_absorbance_stats)

# 7. Normalite Testleri
print("\nNormalite Testleri (Shapiro-Wilk ve D'Agostino's K^2):")
shapiro_test_x = shapiro(df_combined["X_Dalga_Boyu"])  # Shapiro-Wilk testi
shapiro_test_y = shapiro(df_combined["Y_Absorbans"])
normaltest_x = normaltest(df_combined["X_Dalga_Boyu"])  # D'Agostino's K^2 testi
normaltest_y = normaltest(df_combined["Y_Absorbans"])
print(f"Shapiro-Wilk X: {shapiro_test_x}, Shapiro-Wilk Y: {shapiro_test_y}")
print(f"D'Agostino X: {normaltest_x}, D'Agostino Y: {normaltest_y}")
```

Şekil 3.2. İstatistiksel Analiz ve Normallik Testlerine Ait Python Kod Parçası

Şekil 3.3. 'te gösterilen kod kısmı, bir veri setindeki "X_Dalga_Boyu" ve "Y_Absorbans" sütunları için çarpıklık (skewness) ve basıklık (kurtosis) analizlerini gerçekleştirmektedir. Çarpıklık, verilerin simetrik dağılımdan ne kadar saptığını ölçerken, basıklık ise verilerin kuyruklarının kalınlığını ve zirve noktasının keskinliğini değerlendirmektedir. Kodda, her iki sütun için çarpıklık ve basıklık değerleri hesaplanmakta ve bu değerler ekrana yazdırılmaktadır.

```python
# 8. Skewness (Çarpıklık) ve Kurtosis (Basıklık) Analizi
print("\nÇarpıklık (Skewness) ve Basıklık (Kurtosis) Analizi:")
skew_x = skew(df_combined["X_Dalga_Boyu"])
skew_y = skew(df_combined["Y_Absorbans"])
kurt_x = kurtosis(df_combined["X_Dalga_Boyu"])
kurt_y = kurtosis(df_combined["Y_Absorbans"])
print(f"X Çarpıklık: {skew_x}, Y Çarpıklık: {skew_y}")
print(f"X Basıklık: {kurt_x}, Y Basıklık: {kurt_y}")
```

Şekil 3.3. X ve Y Değişkenlerinin Çarpıklık ve Basıklık Değerlerinin Hesaplandığı Python Kod Bloğu

Şekil 3.4. 'de gösterilen kod parçası, bir veri setindeki "X_Dalga_Boyu" ve "Y_Absorbans" sütunlarının dağılımlarını görselleştirmek için kullanılmaktadır. Kod, iki alt grafikten oluşan bir figür oluşturur: ilk grafikte "X_Dalga_Boyu" değerlerinin histogramı, ikinci grafikte ise "Y_Absorbans" değerlerinin histogramı çizilmektedir. Her iki histogram da 50 bin ile oluşturulmuş olup, kenar renkleri ve şeffaflık ayarları ile görsel netlik artırılmıştır. Grafikler, x ve y eksenlerinde ilgili değerlerin etiketlerini ve frekans bilgisini içermektedir.

```python
# 9. Veri Görselleştirme
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
plt.hist(df_combined["X_Dalga_Boyu"], bins=50, edgecolor="black", alpha=0.7)
plt.xlabel("X Değeri (Dalga Boyu)")
plt.ylabel("Frekans")
plt.title("X Değerlerinin Dağılımı")
plt.subplot(1, 2, 2)
plt.hist(df_combined["Y_Absorbans"], bins=50, edgecolor="black", alpha=0.7, color="orange")
plt.xlabel("Y Değeri (Absorbans)")
plt.ylabel("Frekans")
plt.title("Y Değerlerinin Dağılımı")
plt.tight_layout()
plt.show()
```

Şekil 3.4. X (Dalga Boyu) ve Y (Absorbans) Değişkenlerinin Histogram ile Görselleştirilmesi

Şekil-3.5. 'de gösterilen kod parçası, bir veri setindeki "X_Dalga_Boyu" ve "Y_Absorbans" değerleri arasındaki ilişkiyi ve aykırı değerleri analiz etmek için kullanılmaktadır. İlk olarak, "X_Dalga_Boyu" ve "Y_Absorbans" arasındaki ilişkiyi görselleştirmek için bir scatter plot (dağılım grafiği) oluşturulur. Bu grafik, iki değişken

arasındaki potansiyel ilişkiyi ve yoğunlukları anlamaya yardımcı olmaktadır. Daha sonra, her iki değişken için boxplot (kutu grafiği) çizilerek aykırı değerler analiz edilmektedir. Boxplot'lar, verilerin dağılımını, medyanını ve aykırı değerlerin varlığını göstermek için kullanılmaktadır.

```
# 10. Scatter Plot: X-Y İlişkisi
plt.figure(figsize=(8, 5))
plt.scatter(df_combined["X_Dalga_Boyu"], df_combined["Y_Absorbans"], alpha=0.3, s=1)
plt.xlabel("X Değeri (Dalga Boyu)")
plt.ylabel("Y Değeri (Absorbans)")
plt.title("X - Y İlişkisi")
plt.show()

# 11. Aykırı Değer Analizi (Boxplot)
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
sns.boxplot(y=df_combined["X_Dalga_Boyu"], color="lightblue")
plt.title("X Değerlerinin Boxplot Analizi (Dalga Boyu)")
plt.subplot(1, 2, 2)
sns.boxplot(y=df_combined["Y_Absorbans"], color="salmon")
plt.title("Y Değerlerinin Boxplot Analizi (Absorbans)")
plt.tight_layout()
plt.show()
```

Şekil 3.5. X ve Y Değişkenlerinin İlişkisel Dağılımı ve Aykırı Değer Analizi

Şekil 3.6 'da gösterilen kod parçası, bir veri setindeki aykırı değerlerin temizlenmesi ve temizlenmiş veri seti ile orijinal veri seti arasındaki karşılaştırmaları içermektedir. İlk olarak, "Y_Absorbans" sütunundaki aykırı değerler, çeyreklikler (Q1 ve Q3) ve IQR (Interquartile Range) kullanılarak belirlenmekte ve bu değerlerin dışında kalan veriler temizlenmektedir. Daha sonra, orijinal ve temizlenmiş veri setleri arasında "X_Dalga_Boyu" ve "Y_Absorbans" sütunlarının ortalama ve standart sapma değerleri karşılaştırılmaktadır. Ayrıca, bakteri türlerinin dağılımı da orijinal ve temizlenmiş veri setleri arasında karşılaştırılmaktadır. Son olarak, temizlenmiş veri seti, indeksleri sıfırlanarak son haline getirilmekte ve bu veri setinin ilk birkaç satırı ekrana yazdırılmaktadır.

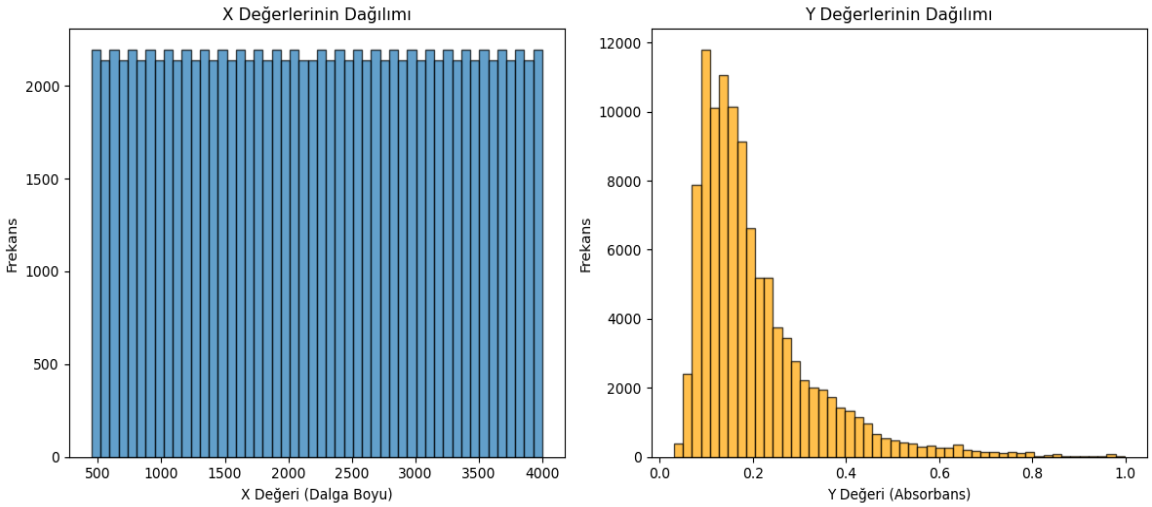
```python
# 12. Aykırı Değerlerden Arındırılmış Veri Seti
Q1 = df_combined["Y_Absorbans"].quantile(0.25)
Q3 = df_combined["Y_Absorbans"].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
df_cleaned = df_combined[(df_combined["Y_Absorbans"] >= lower_bound) & (df_combined["Y_Absorbans"] <= upper_bound)]

# 13. Ham ve Temizlenmiş Veri Karşılaştırması
stats_comparison = pd.DataFrame({
    "Özellik": ["X (Mean)", "X (Std)", "Y (Mean)", "Y (Std)", "Satır Sayısı"],
    "Ham Veri": [df_combined["X_Dalga_Boyu"].mean(), df_combined["X_Dalga_Boyu"].std(), df_combined["Y_Absorbans"].mean(), df_combined["Y_Absorbans"].std(), df_combined.shape[0]],
    "Temizlenmiş Veri": [df_cleaned["X_Dalga_Boyu"].mean(), df_cleaned["X_Dalga_Boyu"].std(), df_cleaned["Y_Absorbans"].mean(), df_cleaned["Y_Absorbans"].std(), df_cleaned.shape[0]]
})
print("\nHam ve Temizlenmiş Veri Karşılaştırması:")
print(stats_comparison)

species_distribution = pd.DataFrame({
    "Bakteri Türü": df_combined["Species"].value_counts().index,
    "Ham Veri Sayısı": df_combined["Species"].value_counts().values,
    "Temizlenmiş Veri Sayısı": df_cleaned["Species"].value_counts().values
})
print("\nBakteri Türlerine Göre Dağılım Karşılaştırması:")
print(species_distribution)

# 14. Ön İşlemlerden geçirilmiş Veri
df_final = df_cleaned.dropna().reset_index(drop=True)
print("\nÖn İşlemlerden geçirilmiş Veri:")
print(df_final.head())
```

Şekil 3.6. Aykırı Değerlerin Filtrelenmesi ve Ham–Temizlenmiş Verilerin İstatistiksel Karşılaştırması

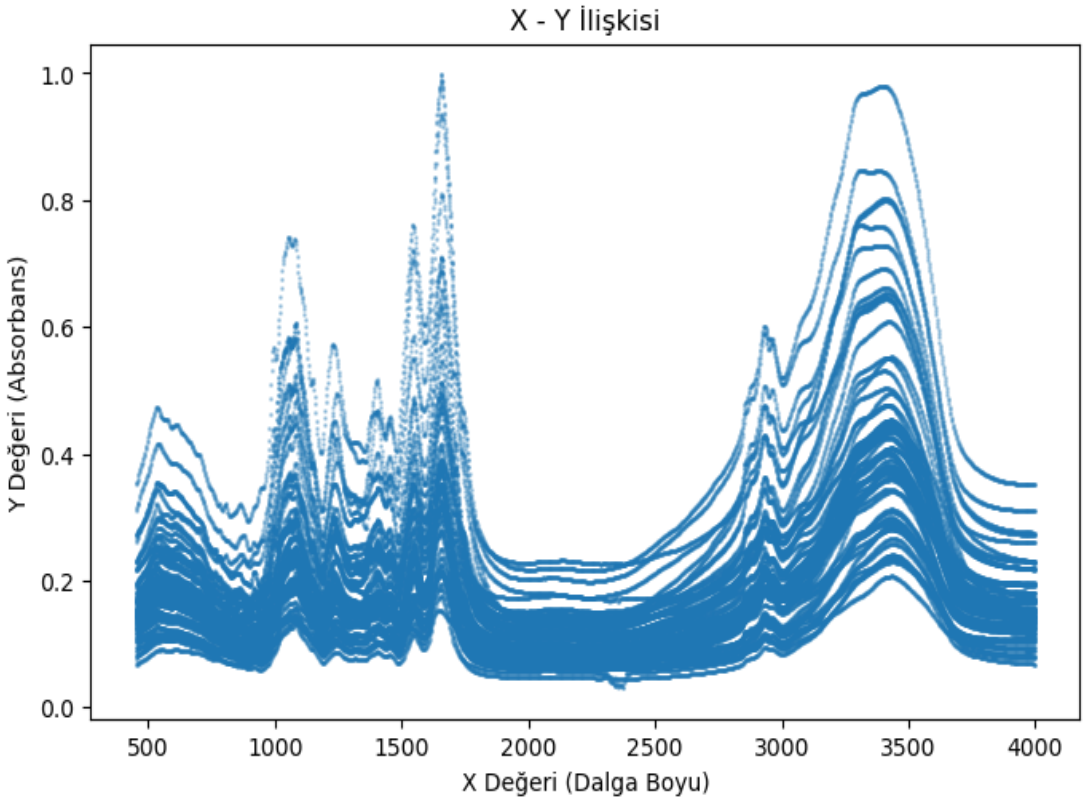
```
Veri setinin genel bilgileri:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 108275 entries, 0 to 108274
Data columns (total 4 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   Bacteria_Code  108275 non-null  object
 1   X_Dalga_Boyu   108275 non-null  float64
 2   Y_Absorbans    108275 non-null  float64
 3   Species        108275 non-null  object
dtypes: float64(2), object(2)
memory usage: 3.3+ MB

Veri setinin özet istatistikleri:

İstatistiksel Analiz:
               count         mean          std         min          25%  \
X_Dalga_Boyu  108275.0  2224.500000  1024.801298  450.500000  1336.500000
Y_Absorbans   108275.0     0.204947     0.130305    0.030611     0.116532

                     50%          75%         max
X_Dalga_Boyu  2224.500000  3112.500000  3998.500000
Y_Absorbans      0.166814     0.249507     0.997961

Eksik Değerler:
...

Çarpıklık (Skewness) ve Basıklık (Kurtosis) Analizi:
X Çarpıklık: 0.0, Y Çarpıklık: 1.952456438218794
X Basıklık: -1.200000761753862, Y Basıklık: 5.010879933350964
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
/opt/anaconda3/lib/python3.12/site-packages/scipy/stats/_axis_nan_policy.py:531: UserWarning: scipy.stats.shapiro: For N > 5000, computed p-value may not be accurate. Current N is 108275.
  res = hypotest_fun_out(*samples, **kwds)
```

Şekil 3.7. FTIR spektral verilerle mikroorganizma sınıflandırması için ön işleme ve analiz sonuçları

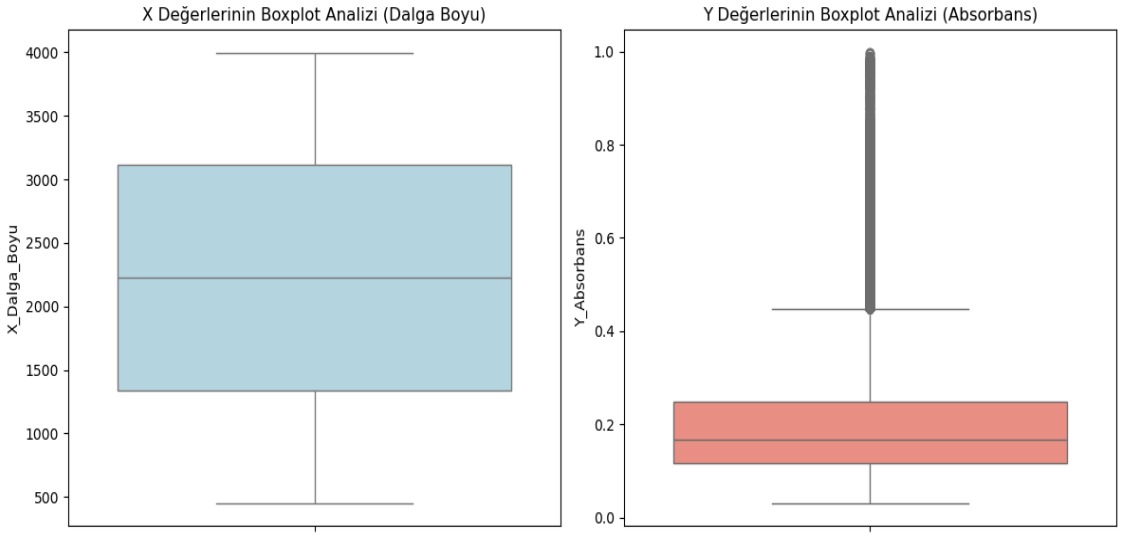
X (Dalga Boyu) ve Y (Absorbans) değerlerinin dağılımı, X – Y ilişkisi, X ve Y değerlerinin Boxplot analizleri sırasıyla Şekil 3.8, Şekil 3.9. , Şekil 3.10 'da gösterilmiştir.



Şekil 3.8. X (Dalga Boyu) ve Y (Absorbans) değerlerinin dağılımı



Şekil 3.9. X (Dalga boyu) – Y (Absorbans) ilişkisi



Şekil 3.10. X ve Y değerlerinin Boxplot Analizi

Ham ve Temizlenmiş Veri karşılaştırması bilgileri, bakteri türlerine göre dağılım karşılaştırması ve Ön işlemlerden geçirilmiş veri bilgileri Şekil-3.11' de gösterilmiştir.

```
Ham ve Temizlenmiş Veri Karşılaştırması:
        Özellik       Ham Veri  Temizlenmiş Veri
0      X (Mean)    2224.500000       2190.991527
1       X (Std)    1024.801298       1022.347664
2      Y (Mean)       0.204947          0.182694
3       Y (Std)       0.130305          0.089485
4  Satır Sayısı  108275.000000     102379.000000

Bakteri Türlerine Göre Dağılım Karşılaştırması:
             Bakteri Türü  Ham Veri Sayısı  Temizlenmiş Veri Sayısı
0    Enterococcus Faecium            44375                    42819
1  Lactobacillus Plantarum           37275                    34950
2  Lactobacillus fermentum           26625                    24610

Ön İşlemlerden geçirilmiş Veri:
  Bacteria_Code  X_Dalga_Boyu  Y_Absorbans                  Species
0        AK5-22         450.5     0.173908  Lactobacillus fermentum
1        AK5-22         452.5     0.174840  Lactobacillus fermentum
2        AK5-22         454.5     0.175818  Lactobacillus fermentum
3        AK5-22         456.5     0.176491  Lactobacillus fermentum
4        AK5-22         458.5     0.177256  Lactobacillus fermentum
```

Şekil-11. Ham ve Temizlenmiş Veri karşılaştırması bilgileri, baketri türlerine göre dağılım karşılaştırması ve Ön işlemlerden geçirilmiş veri bilgileri

3.1. SINIFLANDIRMA

3.1.1. Sınıflandırma Algoritmaları

Sınıflandırma algoritması olarak Random Forest, Gradient Boosting, Extra Trees, HistGradient Boosting, AdaBoost, XGBoost, LightGBM, Support Vector Machines (SVM), Logistic Regression kullanılmıştır. Yapılan işlemler sırasıyla; Veri dönüşümleri, Gelişmiş özellik mühendiisliği, model için en uygun formatı hazırlama, SMOTE uygulama, veri setini ayırma, model listesi, model eğitimi ve testi ve sonuçları görselleştirme işlemleri yapılmıştır. Modeller Şekil-3.12'de, model eğitim ve testi Şekil-3.13'te belirtilmiştir.

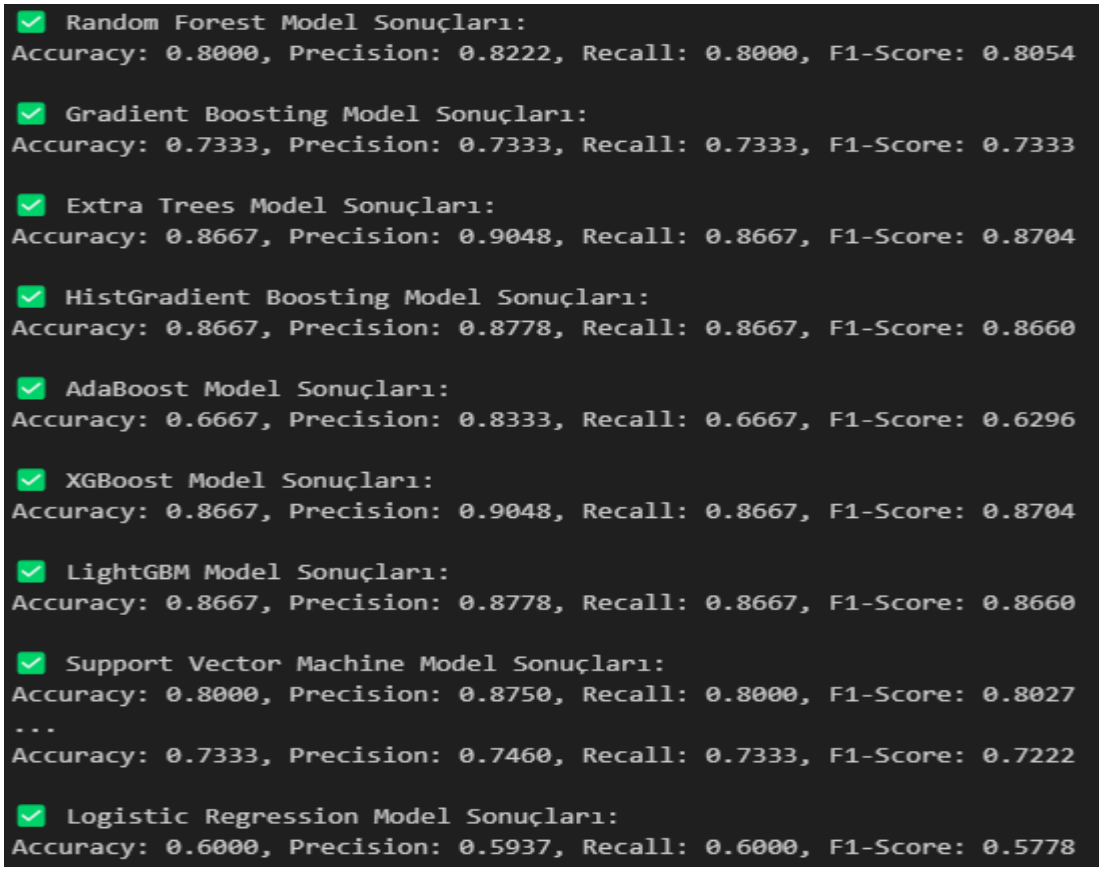
```python
best_models = {
    "Random Forest": RandomForestClassifier(n_estimators=500, max_depth=10, min_samples_split=2, min_samples_leaf=1, random_state=42),
    "Gradient Boosting": GradientBoostingClassifier(n_estimators=100, learning_rate=0.05, max_depth=3, random_state=42),
    "Extra Trees": ExtraTreesClassifier(n_estimators=200, random_state=42),
    "HistGradient Boosting": HistGradientBoostingClassifier(max_iter=50, random_state=42),
    "AdaBoost": AdaBoostClassifier(n_estimators=100, learning_rate=0.05, algorithm="SAMME", random_state=42),
    "XGBoost": XGBClassifier(n_estimators=500, max_depth=10, learning_rate=0.01, eval_metric='mlogloss', random_state=42),
    "LightGBM": LGBMClassifier(n_estimators=100, max_depth=3, verbose=-1),
    "Support Vector Machine": SVC(kernel='rbf', C=1, gamma='scale'),
    "K-Nearest Neighbors": KNeighborsClassifier(n_neighbors=10, weights="distance"),
    "Logistic Regression": LogisticRegression(max_iter=5000, solver="lbfgs")
}
```

Şekil 3.12. Kullanılan Modeller

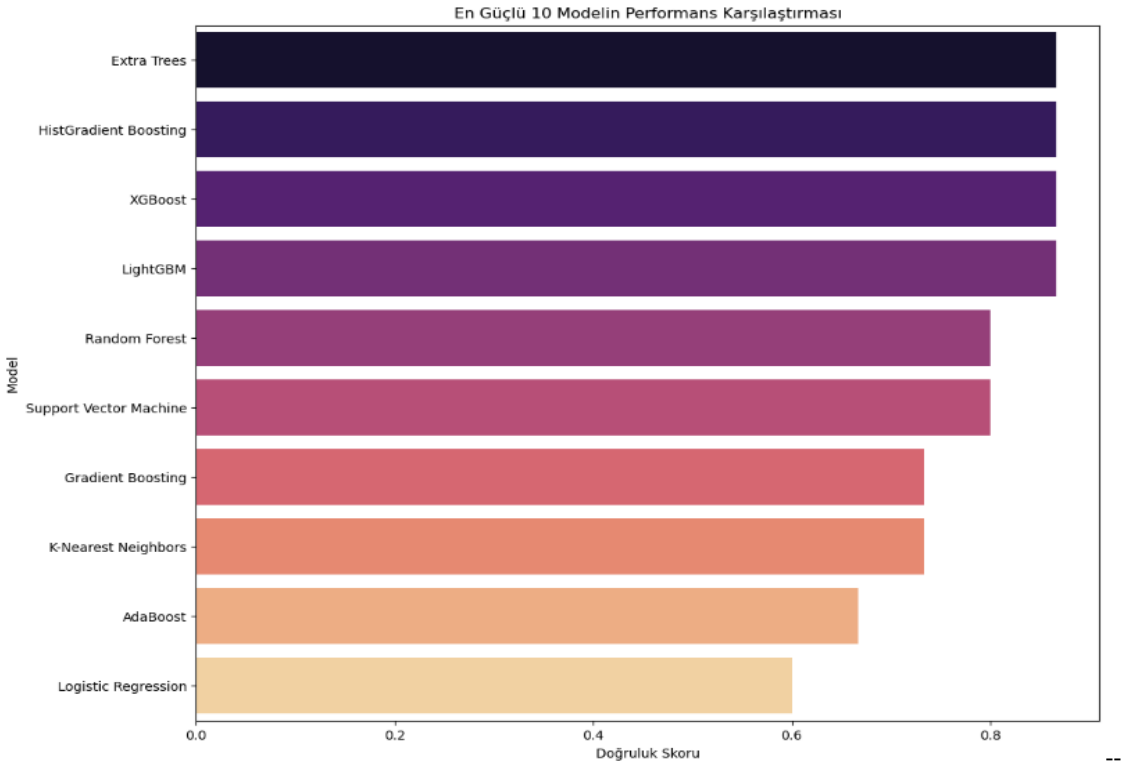
```python
results = []

for name, model in best_models.items():
    try:
        start_time = time.time()
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        acc = accuracy_score(y_test, y_pred)
        report = classification_report(y_test, y_pred, output_dict=True, zero_division=1)
        end_time = time.time()
        duration = end_time - start_time

        results.append({
            "Model": name,
            "Accuracy": acc,
            "Precision": report["weighted avg"]["precision"],
            "Recall": report["weighted avg"]["recall"],
            "F1-Score": report["weighted avg"]["f1-score"],
            "Training Time (sec)": round(duration, 3)
        })
        print(f"\n✅ {name} Model Sonuçları: \nAccuracy: {acc:.4f}, Precision: {report['weighted avg']['precision']:.4f}, Recall: {report['weighted avg']['recall']:.4f}, F1-Score: {report['weighted avg']['f1-score']:.4f}")
    except Exception as e:
        print(f"\n❌ {name} Modeli hata verdi! Hata: {str(e)}")
```

Şekil 3.13. Sınıflandırma Modellerinin Eğitim, Test ve Performans Değerlendirme Kod Bloğu

Random Forest, Gradient Boosting, Extra trees, HistGradient Boosting, AdaBoost, XGBoost, LightGBM, Support Vector Machines (SVM), Logistic Regression modellerinin Accuracy, Precision, Recall ve F1-Score değerleri Şekil 3.14, grafiği Şekil 3.15' te verilmiştir.



Şekil-3.14. Accuracy, Precision, Recall ve F1-Score değerleri



Şekil-3.15. Modellerin Performans Karşılaştırma Grafiği

3.1.2. Sınıflandırma Algoritmalarının Kombinasyonları ve Performansları

Kombinasyonlar Random Forest + XGBoost, LightGBM + Extra Trees, XGBoost + Extra Trees, All Four Models şeklindedir.

3.1.2.1. Random Forest + XGBoost

Şekil 3.1.2.1 'te gösterilen kod, makine öğrenmesi alanında sıkça kullanılan bir teknik olan topluluk öğrenmesi (ensemble learning) uygulamasını göstermektedir. Kod, önceden eğitilmiş en iyi Random Forest ve XGBoost modellerini bir VotingClassifier içinde birleştirerek daha güçlü bir sınıflandırıcı oluşturmaktadır. "soft" oylama parametresi, her modelin tahmin olasılıklarını kullanarak daha hassas bir sonuç elde edilmesini sağlar.

```
"Random Forest + XGBoost": VotingClassifier(
    estimators=[('rf', best_models["Random Forest"]), ('xgb', best_models["XGBoost"])], voting='soft'),
```

Şekil 3.1.2.1. Random Forest ve XGBoost Modellerinin VotingClassifier ile Topluluk Modeli Olarak Birleştirilmesi

3.1.2.2. LightGBM + Extra Trees

Şekil-3.1.2.2. 'de gösterilen kod, makine öğrenmesi tahmin performansını artırmak için topluluk öğrenmesi (ensemble learning) tekniğini uygulamaktadır. "LightGBM + Extra Trees" olarak adlandırılan bu modelde, önceden eğitilmiş en iyi LightGBM ve Extra Trees sınıflandırıcıları VotingClassifier yapısı içinde birleştirilmektedir. "soft" oylama parametresi, her bir modelin tahmin olasılıklarını değerlendirerek nihai kararı vermeyi sağlar - bu da keskin sınıf etiketleri yerine daha nüanslı tahminler üretir.

```
"LightGBM + Extra Trees": VotingClassifier(
    estimators=[('lgbm', best_models["LightGBM"]), ('et', best_models["Extra Trees"])], voting='soft'),
```

Şekil 3.1.2.2. LightGBM + Extra Trees Ensemble Modeli (Soft Voting)

3.1.2.3. XGBoost + Extra Trees

Şekil-3.1.2.3. 'te gösterilen kod, VotingClassifier kullanarak XGBoost ve Extra Trees modellerini birleştiren bir ensemble yöntemini gösteriyor.Şekil-*'te, soft oylama yöntemi kullanılmış, yani her modelin sınıf olasılıkları dikkate alınarak nihai tahmin yapılmaktadır.

```
"XGBoost + Extra Trees": VotingClassifier(
    estimators=[('xgb', best_models["XGBoost"]), ('et', best_models["Extra Trees"])], voting='soft'),
```

Şekil 3.1.2.3. XGBoost + Extra Trees Ensemble Modeli (Soft Voting)

3.1.2.4. All Four Models

Şekil 3.1.2.4. 'te gösterilen kod, Random Forest, XGBoost, LightGBM ve Extra Trees modelleri bir araya getirilmiş ve soft oylama yöntemi kullanılmış. Soft oylama, her modelin sınıf olasılıklarını dikkate alarak nihai tahmini yapar.

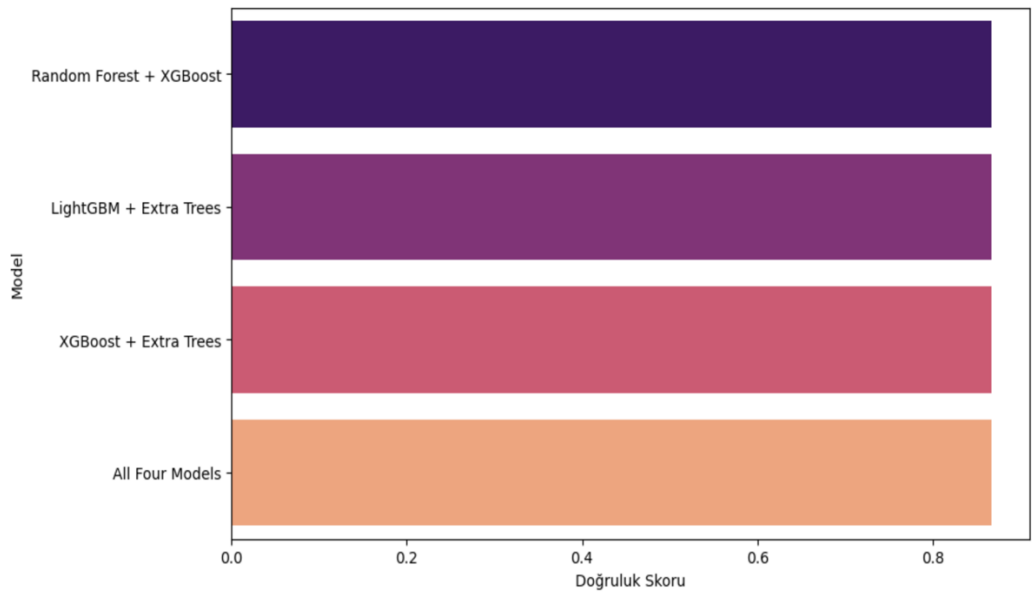
```
"All Four Models": VotingClassifier(
    estimators=[('rf', best_models["Random Forest"]), ('xgb', best_models["XGBoost"]),
                ('lgbm', best_models["LightGBM"]), ('et', best_models["Extra Trees"])], voting='soft')
```

Şekil 3.1.2.4. LightGBM + Extra Trees+ XGBoost+Random Forest Ensemble Modeli (Soft Voting)

Sınıflandırma modellerinde yüksek performanslı Random Forest ve XGBoost, LightGBM ve Extra Trees, XGBoost ve Extra Trees kombinasyonları denenmiştir. Bu modellerin kombinasyonların performansları Çizelge 3.1.2 ve Şekil 3.1.2' de belirtilmiştir.

Çizelge 3.1.2. Random Forest, XGBoost, LightGBM ve Extra Trees modellerinin çeşitli kombinasyonlarıyla elde edilen sınıflandırma performansı metrikleri.

Model Combination	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Random Forest + XGBoost	0.8667	0.9048	0.8667	0.8704
LightGBM + Extra Trees	0.8667	0.8778	0.8667	0,8660
XGBoost + Extra Trees	0.8667	0.9048	0.8667	0.8704
All Four Models	0.8667	0.9048	0.8667	0.8704



Şekil 3.1.2. Farklı Essemble Modellerin Karşılaştırılması

3.2. Derin Öğrenme Modelleri

Projeni geliştirilme aşamasında derin öğrenme modelleri nde CNN, LSTM, ve hibrit bir model olan CNN+LSTM kullanılmıştır. Sırasıyla yapılan işlemler verilerin yeniden yapılandırılması, verileri bölümleme, verileri yeniden şekillendirme Derin öğrenme modellerinin tanımlanması, model eğitim fonksiyonunun yazılması, performans değerlendirme, modellerin eğitimi ve karşılaştırma, performans görselleştirme, en iyi modelin seçilmesi, hiperparametre optimizasyonu, veri artırma, En iyi model seçme ve yenidden eğitme, yeniden eğitilmiş modelin performansını değerlendirme ve performans göreselleştirme şeklindedir.

3.2.1. CNN

Şekil 3.2.1.'de, görüntü sınıflandırma için bir yapay zeka modeli oluşturmaktadır. Özellikle bir Evrişimli Sinir Ağı (CNN) tasarlayan bu fonksiyon, görüntülerin özelliklerini otomatik olarak öğrenir. Model önce iki evrişim katmanı kullanarak görüntülerdeki önemli desenleri yakalar, ardından bu özellikleri tam bağlantılı katmanlara aktararak sınıflandırma yapar. Aşırı öğrenmeyi önlemek için dropout katmanları eklenmiştir. Son katman, belirlenen sınıf sayısına göre ayarlanıp softmax aktivasyonu kullanarak her bir sınıfa ait olasılıkları hesaplar. Model, Adam optimizasyon algoritması ve sparse_categorical_crossentropy kayıp fonksiyonu ile derlenerek eğitime hazır hale getirilmiştir.

```
def build_cnn_model(input_shape, num_classes):
    model = Sequential([
        Input(shape=input_shape),  # Input katmanı eklendi
        Conv1D(64, 3, activation='relu'),
        MaxPooling1D(2),
        Dropout(0.3),
        Conv1D(128, 3, activation='relu'),
        MaxPooling1D(2),
        Flatten(),
        Dense(256, activation='relu'),
        Dropout(0.5),
        Dense(num_classes, activation='softmax')
    ])
    model.compile(
        optimizer=Adam(0.001),
        loss='sparse_categorical_crossentropy',
        metrics=['accuracy']
    )
    return model
```

Şekil 3.2.1. Girdi Katmanı, Evrişim Katmanları ve Fully Connected Katmanlardan Oluşan CNN Modelinin Python Kod Bloğu

3.2.2. LSTM

Şekil-3.2.2.'de, sıralı veri analizi için bir Uzun-Kısa Vadeli Bellek (LSTM) modeli oluşturmaktadır. Fonksiyon, giriş boyutunu ve hedef sınıf sayısını parametre olarak alır ve ardından katmanları adım adım inşa eder. İlk olarak 128 birimlik bir LSTM katmanı, ardından aşırı öğrenmeyi engellemek için bir dropout katmanı ekler. Sonrasında 64 birimlik ikinci bir LSTM katmanı ve ReLU aktivasyonlu yoğun bir katman gelir. Son olarak, bir dropout katmanının ardından softmax aktivasyonlu çıkış katmanı eklenir. Model, 0.0005 öğrenme oranlı Adam optimizasyon algoritması, sparse_categorical_crossentropy kayıp fonksiyonu ve doğruluk metriği kullanılarak derlenir. Bu yapı, metin sınıflandırma, duygu analizi veya zaman serisi tahmini gibi sıralı veri gerektiren problemler için uygundur.

```
def build_lstm_model(input_shape, num_classes):
    model = Sequential([
        Input(shape=input_shape),  # Input katmanı eklendi
        LSTM(128, return_sequences=True),
        Dropout(0.2),
        LSTM(64),
        Dense(64, activation='relu'),
        Dropout(0.3),
        Dense(num_classes, activation='softmax')
    ])
    model.compile(
        optimizer=Adam(0.0005),
        loss='sparse_categorical_crossentropy',
        metrics=['accuracy']
    )
    return model
```

Şekil 3.2.2. İki Katmanlı LSTM ve Fully Connected Katmanlardan Oluşan Sınıflandırma Modelinin Python Kod Bloğu

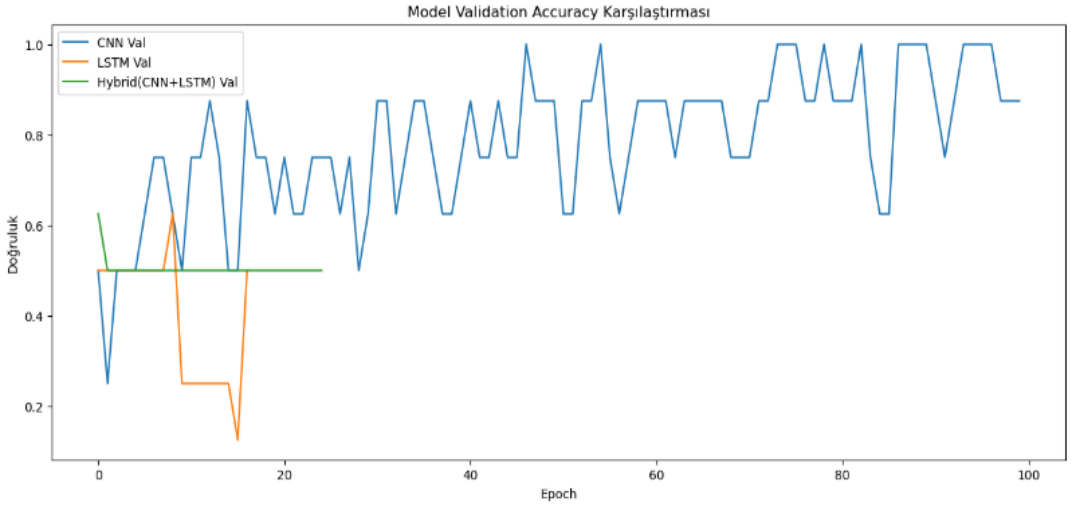
3.2.3. CNN + LSTM

Şekil-3.2.3.1. 'te, CNN ve LSTM tekniklerini birleştiren hibrit bir yapay zeka modeli oluşturmaktadır. Fonksiyon, belirli bir giriş boyutu ve sınıf sayısına göre çalışan bir model inşa eder. Öncelikle, 32 filtreli bir evrişim katmanı görüntü veya veri üzerindeki desenleri yakalar ve bir havuzlama katmanı önemli özellikleri sıkıştırır. Ardından, sıralı veri işleme için 64 birimlik bir LSTM katmanı ve bunu takip eden 32 birimlik ikinci bir LSTM katmanı eklenir. Model, 128 nöronlu tam bağlantılı bir katmana bağlanır ve aşırı öğrenmeyi önlemek için farklı oranlarda dropout katmanları kullanılır. Son katman, softmax aktivasyonu ile sınıflandırma sonuçlarını üretir. Model, Adam optimizasyonu, kategorik çapraz entropi kayıp fonksiyonu ve doğruluk metriği ile derlenir.

```
def build_hybrid_model(input_shape, num_classes):
    model = Sequential([
        Input(shape=input_shape),
        Conv1D(32, 3, activation='relu'),
        MaxPooling1D(2),
        LSTM(64, return_sequences=True),
        Dropout(0.3),
        LSTM(32),
        Dense(128, activation='relu'),
        Dropout(0.4),
        Dense(num_classes, activation='softmax')
    ])
    model.compile(
        optimizer=Adam(0.00075),
        loss='sparse_categorical_crossentropy',
        metrics=['accuracy']
    )
    return model
```

Şekil 3.2.3.1. CNN–LSTM Hibrit Sınıflandırma Modelinin Python Kod Bloğu

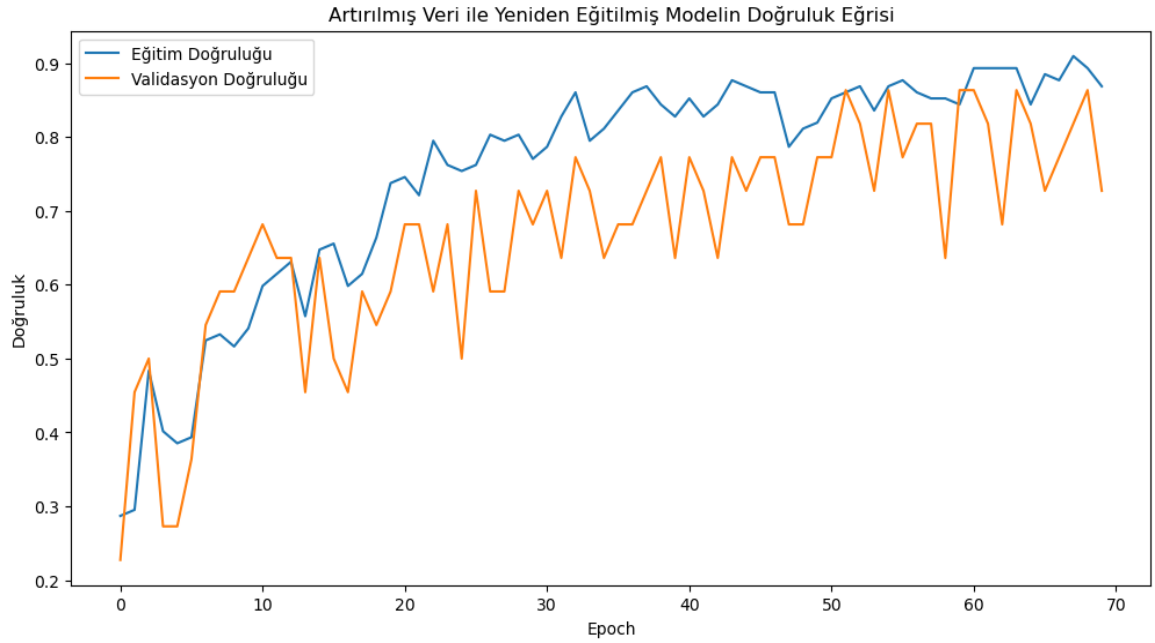
Şekil 3.2.3.2. 'te, farklı modellerin (CNN, LSTM ve Hibrit CNN+LSTM) doğrulama (validation) doğruluklarını (accuracy) karşılaştırmaktadır. Grafik, epoch sayısına göre doğrulama doğruluğunun nasıl değiştiğini göstermektedir. Hibrit model (CNN+LSTM), genellikle en yüksek doğruluk oranına sahip gibi görünmekte ve epoch sayısı arttıkça daha iyi performans göstermektedir. CNN ve LSTM modelleri de benzer bir eğilim izlemekte, ancak hibrit model kadar yüksek doğruluk oranlarına ulaşamamaktadır. Grafik, hibrit modelin diğer modellere kıyasla daha etkili olduğunu ve daha fazla epoch ile daha iyi sonuçlar verebileceğini göstermektedir.



Şekil 3.2.3.2. Model Validasyon Accuracy Karşılaştırılması

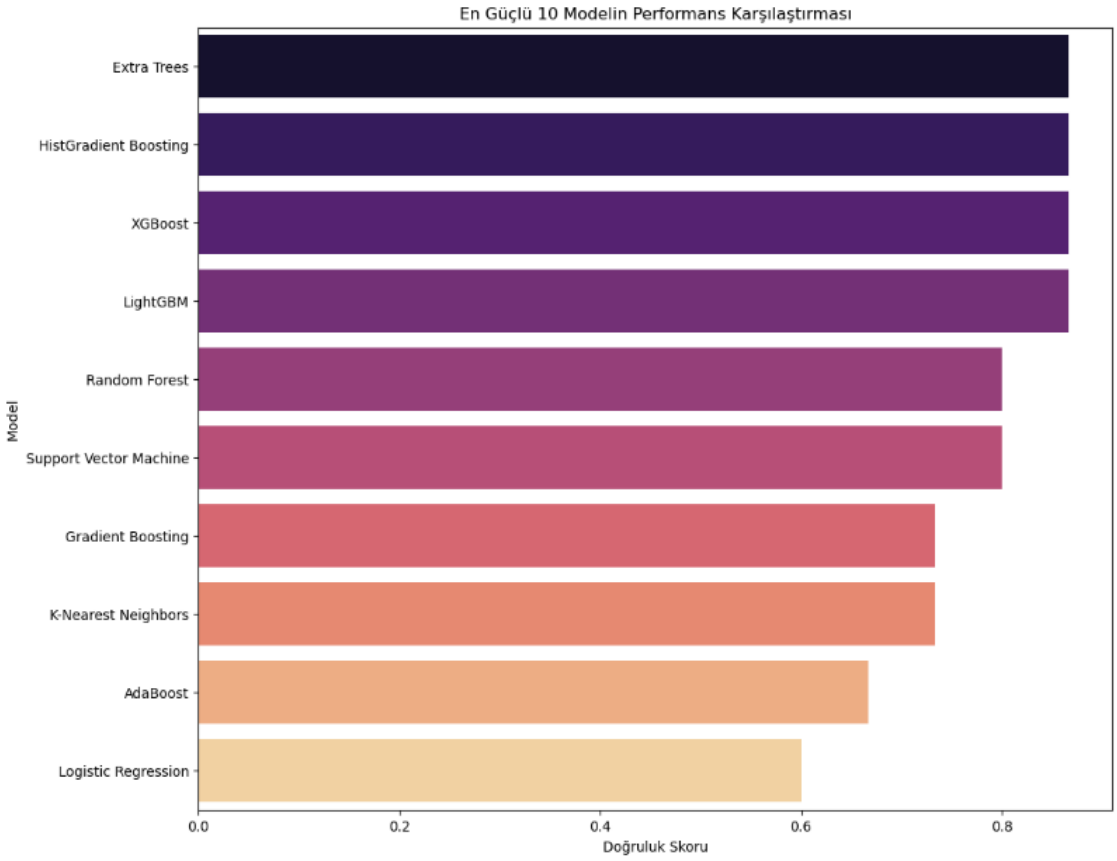
--

Şekil 3.2.3.3**,** artırılmış veri ile yeniden eğitilmiş bir modelin eğitim ve doğrulama (validation) doğruluk (accuracy) eğrilerini göstermektedir. Grafikte, hem eğitim hem de doğrulama doğrulukları epoch sayısına göre değişmektedir. Eğitim doğruluğu, modelin eğitim verileri üzerindeki performansını göstermekteyken, doğrulama doğruluğu modelin görmediği veriler üzerindeki genelleme yeteneğini değerlendirmektedir. Grafikten, modelin epoch sayısı arttıkça hem eğitim hem de doğrulama doğruluklarının arttığı görülmektedir.



Şekil 3.2.3.3. Artırılmış Veri ile Eğitilen Modelin Eğitim ve Doğrulama Doğruluk Eğrileri

4. BULGULAR



Şekil-4.1. Farklı Sınıflandırıcıların Doğruluk Performanslarının Karşılaştırılması

Çizelge. 4.1. Makine Öğrenimi ve Derin Öğrenme Modellerinin Sınıflandırma Performans Karşılaştırması

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.8000	0.8222	0.8000	0.8054
Gradient Boosting	0.7333	0.7333	0.7333	0.7333
Extra Trees	0.8667	0.9048	0.8667	0.8704
HistGradient Boosting	0.8667	0.8778	0.8667	0.8660
AdaBoost	0.6667	0.8333	0.6667	0.6296
XGBoost	0.8667	0.9048	0.8667	0.8704
LightGBM	0.8667	0.8778	0.8667	0.8660
Support Vector Machine (SVC)	0.8000	0.8750	0.8000	0.8027
K-Nearest Neighbors (KNN)	0.7333	0.7460	0.7333	0.7222
Logistic Regression	0.6000	0.5937	0.6000	0.5778
Random Forest + XGBoost	0.8667	0.9048	0.8667	0.8704
LightGBM + Extra Trees	0.8667	0.8778	0.8667	0.8660
XGBoost + Extra Trees	0.8667	0.9048	0.8667	0.8704
All Four Models	0.8667	0.9048	0.8667	0.8704
LSTM	0.3846	0.38	1.00	0.56
Hybrid (CNN+LSTM)	0.3846	0.38	1.00	0.56
CNN (Augmented Data)	0.9231	0.94	0.93	0.93

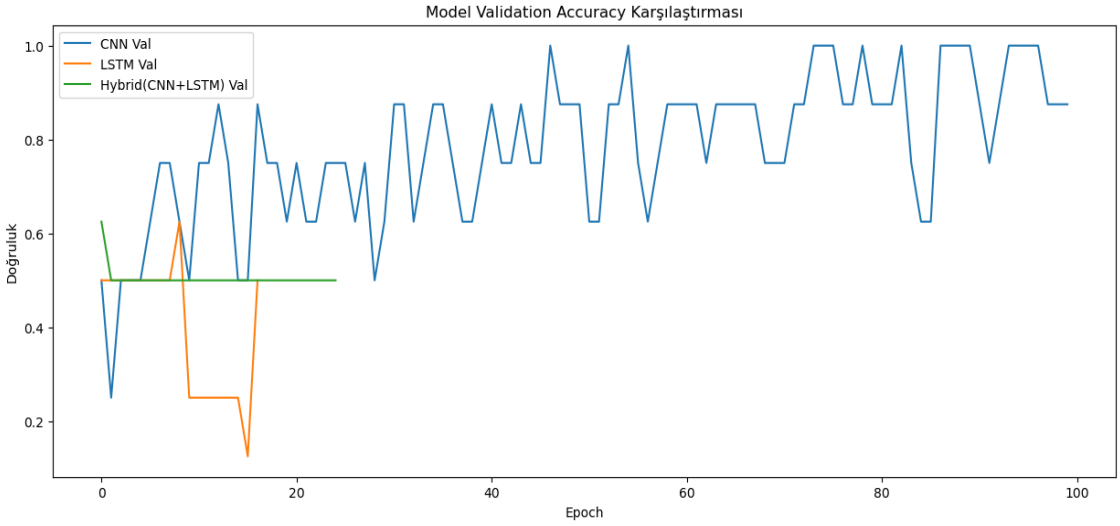
Çizelge 4.1., farklı makine öğrenimi ve derin öğrenme modellerinin **Accuracy (Doğruluk), Precision (Kesinlik), Recall (Duyarlılık) ve F1-Score** metrikleri açısından performans karşılaştırmaları sunulmuştur. Elde edilen sonuçlar, farklı modellerin sınıflandırma başarımını değerlendirmek için kullanılmaktadır.

- **CNN (Augmented Data) modeli**, %92.31 doğruluk ile en yüksek performansı göstermiştir. Precision (%94), Recall (%93) ve F1-Skoru (%93) değerleri, veri artırma tekniklerinin modelin genelleştirme yeteneğini artırdığını göstermektedir.
- **En iyi geleneksel makine öğrenimi modelleri**, **Extra Trees, XGBoost ve Random Forest + XGBoost** kombinasyonları olmuştur. Bu modeller %86.67 doğruluk oranına ulaşarak benzer performans sergilemiştir.
- **LSTM ve Hybrid (CNN+LSTM) modelleri**, düşük doğruluk (%38.46) göstermesine rağmen, **%100 Recall değerine** sahip olmaları dikkat çekicidir. Bu durum, modellerin yanlış negatifleri en aza indirerek tüm pozitif sınıfları yakalama yeteneğine sahip olduğunu göstermektedir. Ancak Precision değerinin düşük olması (%38) modelin fazla sayıda yanlış pozitif ürettiğini işaret etmektedir.

- **Geleneksel yöntemlerden Logistic Regression ve K-Nearest Neighbors (KNN) modelleri**, diğer makine öğrenimi yöntemlerine kıyasla daha düşük doğruluk oranları sergilemiştir (%60 ve %73.33).
- **Topluluk (ensemble) modelleri**, genellikle yüksek doğruluk ve F1-Score değerleri sunmuştur. Özellikle **Random Forest + XGBoost ve XGBoost + Extra Trees kombinasyonları** başarılı sonuçlar üretmiştir.
- **Destek Vektör Makineleri (SVM)**, **%80 doğruluk oranı** ile güçlü bir performans göstermiştir. Ancak doğruluk seviyesi, XGBoost ve Extra Trees gibi modellerin gerisinde kalmıştır.

CNN Model Validation Accuracy Karşılaştırması

Şekil 4.2. , farklı model türlerinin (CNN, LSTM ve Hibrit CNN+LSTM) doğrulama (validation) doğruluklarını karşılaştırmaktadır. Grafik, modellerin eğitim süreci boyunca epoch sayısına bağlı olarak doğruluk oranlarını göstermektedir. Hibrit model (CNN+LSTM), genellikle en yüksek doğruluk oranına ulaşarak diğer modellerden daha iyi performans göstermektedir.



Şekil 4.2. CNN, LSTM ve CNN–LSTM Hibrit Modellerinin Doğrulama Doğruluklarının Epoch Bazlı Karşılaştırılması

5. DEĞERLENDİRME VE TARTIŞMA

Bu çalışmada, mikroorganizma sınıflandırması için farklı makine öğrenimi ve derin öğrenme modelleri karşılaştırılmıştır. CNN (Augmented Data) modeli, %92.31 doğruluk oranı ile en yüksek performansı sergilemiş ve önceki çalışmalarla kıyaslandığında oldukça rekabetçi bir sonuç elde etmiştir.

Örneğin, Dziuba (2013) çalışmasında FTIR spektroskopisi ve Yapay Sinir Ağları (YSA) kullanılarak %93 doğruluk oranı elde edilmiştir. Alakuş (2023), mikroorganizma sınıflandırmasında Tekrarlayıcı Sinir Ağları (RNN) ve Uzun/Kısa Süreli Bellek (LSTM) modellerinin performanslarını değerlendirmiştir. Çalışmalarında RNN modeli %92.53, LSTM modeli ise %99.85 doğruluk oranı sağlamıştır.

Çalışmamızda elde edilen doğruluk oranı (%92.31), FTIR tabanlı YSA yaklaşımına yakın bir başarı göstermektedir. Bu da FTIR spektroskopisinin derin öğrenme teknikleriyle analiz edilmesinin etkili bir yöntem olduğunu desteklemektedir.

Bununla birlikte, metagenomik verilerle yapılan bazı çalışmalar, derin öğrenme modellerinin sınıflandırma başarısını artırabileceğini göstermektedir. Fiannaca ve ark. (2018), bakterilerin 16S rRNA genleri üzerinden CNN ve Derin İnanç Ağları (DBN) ile sınıflandırılabileceğini göstermiştir. Bu modeller, bakteriyel toplulukları cins seviyesine kadar ayırt etmede başarılı sonuçlar vermişti**r.**

Benzer şekilde, Shakibania ve ark. (2024) çalışmasında, probiyotik türlerin patojenlere karşı koruyucu etkilerini anlamak amacıyla CNN modelleri kullanılmıştır. CNN modeli, biyobelirteçler ve mikrobiyal görüntü verilerini birleştirerek sınıflandırma başarısını artırmıştır.

Son dönem çalışmalarından biri olan Enders ve ark. (2024), FTIR spektrumlarının fonksiyonel gruplarını analiz etmek için CNN tabanlı derin öğrenme modellerini kullanmıştır. Bu çalışma, FTIR spektroskopisinin CNN ile birleştirildiğinde daha hızlı ve etkili analiz yapılabileceğini ortaya koymuştur. Bizim modelimizin de FTIR spektrumlarıyla başarılı sonuçlar üretmesi, bu yöntemin mikroorganizma sınıflandırması için geniş bir uygulama potansiyeline sahip olduğunu göstermektedir.

Makine öğrenimi modelleri arasında en iyi performans gösteren yöntemler XGBoost, Extra Trees ve Random Forest kombinasyonları olmuştur. Bu modeller, %86.67 doğruluk oranına ulaşarak, CNN modeli kadar yüksek olmasa da, başarılı sonuçlar elde etmiştir. Destek Vektör Makineleri (SVM) ise %80 doğruluk oranı ile rekabetçi bir performans sergilemiştir. Bununla birlikte, Logistic Regression ve K-

Nearest Neighbors (KNN) gibi geleneksel yöntemler, diğer modellere kıyasla daha düşük doğruluk oranları sunmuştur (%60 ve %73.33).

LSTM ve Hybrid (CNN+LSTM) modelleri, düşük doğruluk (%38.46) göstermelerine rağmen, %100 recall değerine sahip olmaları dikkat çekicidir. Bu sonuç, modellerin tüm pozitif sınıfları doğru şekilde yakalayabildiğini ancak yanlış pozitif oranlarının yüksek olduğunu göstermektedir. Hassasiyetin (recall) kritik olduğu sağlık ve güvenlik gibi alanlarda, bu tür modellerin kullanımı avantajlı olabilirken, daha dengeli bir doğruluk ve F1-Score gerektiren genel sınıflandırma görevlerinde, CNN ve XGBoost tabanlı yöntemler daha etkili bir seçenek sunmaktadır.

Wu ve arkadaşları (2024) tarafından yapılan araştırmada, probiyotiklerin metagenomik verilerden çıkarımını kolaylaştırmak için karar ağaçları ve XGBoost algoritmaları uygulanmıştır. Karar ağaçları, probiyotik biyobelirteçlerin sınıflandırılmasında temel yapı taşlarını oluştururken, XGBoost algoritması, büyük ve karmaşık veri kümelerindeki örüntüleri daha doğru bir şekilde modellemiştir. Wu ve ark. (2024) yapmış oldukları çalışma sonuçları bu projedi sonuçlarla paralellik göstermektedir.

Bu karşılaştırmalar, model seçiminde uygulama alanına ve belirlenen metriklere göre farklı yöntemlerin tercih edilmesi gerektiğini göstermektedir. CNN tabanlı modeller, **özellikle veri artırma teknikleriyle birlikte genelleme yeteneğini artırarak en iyi performansı sağlamıştır.** Ancak geleneksel **topluluk (ensemble) modelleri**, yüksek doğruluk ve istikrar sunmaları nedeniyle alternatif olarak değerlendirilebilir.

Bu çalışma, FTIR spektroskopisiyle mikroorganizma sınıflandırmasında derin öğrenme ve geleneksel makine öğrenimi yöntemlerinin etkinliğini karşılaştırarak, **veri artırma tekniklerinin ve model seçiminin sınıflandırma başarısı üzerindeki kritik rolünü** ortaya koymaktadır.

6. SONUÇ VE ÖNERİLER

Bu çalışmada, mikroorganizma sınıflandırması için farklı makine öğrenimi ve derin öğrenme modellerinin performansları karşılaştırılmıştır. CNN (Augmented Data) modeli, %92.31 doğruluk oranı ile en yüksek performansı sergilemiş ve özellikle FTIR spektroskopisi verileri üzerinde etkili bir şekilde uygulanabileceğini göstermiştir. Bu sonuç, FTIR spektroskopisi ile derin öğrenme tekniklerinin birleştirilmesinin mikroorganizma sınıflandırması için güçlü bir yöntem olduğunu ortaya koymaktadır. Ayrıca, XGBoost, Extra Trees ve Random Forest gibi topluluk (ensemble) modelleri de %86.67 doğruluk oranı ile rekabetçi bir performans sergilemiş ve geleneksel makine öğrenimi yöntemlerinin hala etkili bir alternatif olabileceğini göstermiştir.

Çalışmamızın sonuçları, literatürdeki diğer çalışmalarla tutarlılık göstermektedir. Örneğin, Dziuba (2013) ve Alakuş (2023) gibi çalışmalarda da benzer doğruluk oranları elde edilmiştir. Özellikle LSTM ve Hybrid (CNN+LSTM) modellerinin %100 recall değeri ile tüm pozitif sınıfları doğru şekilde yakalayabilmesi, bu modellerin hassasiyetin kritik olduğu sağlık ve güvenlik gibi alanlarda potansiyel kullanımını ortaya koymaktadır. Ancak, bu modellerin düşük doğruluk oranları, genel sınıflandırma görevlerinde daha dengeli bir performans sergileyen CNN ve XGBoost tabanlı yöntemlerin tercih edilmesini daha uygun hale getirmektedir.

Gelecekteki çalışmalar için, özellikle metagenomik verilerin ve FTIR spektroskopisi verilerinin daha büyük ve çeşitli veri kümeleri üzerinde analiz edilmesi önerilmektedir. Bu tür veri kümeleri, modellerin genelleme yeteneklerini artırarak daha güvenilir sonuçlar elde edilmesini sağlayabilir. Ayrıca, farklı veri artırma tekniklerinin ve hibrit modellerin (örneğin, CNN ile LSTM'nin kombinasyonu) daha detaylı incelenmesi, sınıflandırma performansını daha da artırabilir. Bunun yanı sıra, probiyotikler ve patojenler gibi spesifik mikroorganizma grupları üzerinde odaklanan çalışmalar, bu alanlarda daha derinlemesine analizler yapılmasına olanak tanıyabilir.

Sonuç olarak, bu çalışma, mikroorganizma sınıflandırmasında derin öğrenme ve geleneksel makine öğrenimi yöntemlerinin etkinliğini karşılaştırmış ve veri artırma tekniklerinin model performansı üzerindeki önemini vurgulamıştır. Elde edilen bulgular, FTIR spektroskopisi ve derin öğrenme tekniklerinin mikroorganizma sınıflandırması için geniş bir uygulama potansiyeline sahip olduğunu göstermektedir. Gelecekteki çalışmalar, önerilen modellerin **endüstriyel, klinik ve çevresel analizlerde kullanımı için prototip**

bir sistem geliştirilmesi, teorik başarıdan pratik uygulamaya geçiş sürecini hızlandırabilir.

7. KAYNAKLAR

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 265-283.

Alakuş, T. B. (2023). Tekrarlayıcı sinir ağları ile mikropların sınıflandırılması. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 35*(2), 735–743. https://doi.org/10.35234/fumbd.1302903

Albayrak, G. (2010). *Halofilik Arkea ve Bakteria İdentifikasyonunda FT-IR (Fourier Transform-Infrared Spektroskopisi) Kullanımı*. Anadolu Üniversitesi Fen Bilimleri Enstitüsü.

Ayoola, M. B., Pillai, N., Nanduri, B., Rothrock, M. J., Jr., & Ramkumar, M. (2023). Predicting foodborne pathogens and probiotics taxa within poultry-related microbiomes using a machine learning approach. *Animal Microbiome, 5*(57). https://doi.org/10.1186/s42523-023-00260-w

Başyiğit Kılıç, G. (2009). *Bazı Laktobasil suşlarının genetik tanısının yapılması ve faj dirençliliklerinin belirlenmesi* (Doktora tezi). Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Isparta.

Başyiğit Kılıç, G., & Karahan, A. G. (2010). Fourier Dönüşümlü Kızılötesi (FTIR) Spektroskopisi ve Laktik Asit Bakterilerinin Tanısında Kullanılması. *Gıda*, 35(6), 445-452.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chollet, F. (2015). *Keras*. https://keras.io

Cordovana, M., Mauder, N., Join-Lambert, O., Gravey, F., LeHello, S., Auzou, M., Pitti, M., Zoppi, S., Buhl, M., Steinmann, J., Frickmann, H., Dekker, D., Funashima, Y., Nagasawa, Z., Soki, J., Orosz, L., Veloo, A. C., Justesen, U. S., Holt, H. M., ... Kostrzewa, M. (2022). Machine learning-based typing of *Salmonella enterica* O-serogroups by the Fourier-Transform Infrared (FTIR) Spectroscopy-based IR Biotyper system. *Journal of Microbiological Methods, 201*, 106564. https://doi.org/10.1016/j.mimet.2022.106564

Çırak, O. (2011). *Fourier Transform Infrared (FT-IR) Spektroskopisi Kullanılarak Süt Türlerinin Belirlenmesi*. Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü.

Dietterich, T. G. (2000). Ensemble methods in machine learning. *International workshop on multiple classifier systems*, 1–15. https://doi.org/10.1007/3-540-45014-9_1

Dziuba, B. (2013). Identification of Propionibacteria to the species level using Fourier transform infrared spectroscopy and artificial neural networks. *Polish Journal of Veterinary Sciences, 16*(2), 351–357. https://doi.org/10.2478/pjvs-2013-0047

Enders, A. A., North, N. M., Fensore, C. M., Velez-Alvarez, J., & Allen, H. C. (2021). Functional group identification for FTIR spectra using image-based machine learning

models. *Analytical Chemistry, 93*(28), 9711–9718. https://doi.org/10.1021/acs.analchem.1c00867

Ergüven, Ö., & Ökten, S. (2022). Yapay zekânın mikrobiyolojide kullanımı. *Sağlık Bilimlerinde Yapay Zeka Dergisi, 2*(2), 1–12. https://doi.org/10.52309/jaihs.v2i2.41

Fiannaca, A., La Paglia, L., La Rosa, M., Lo Bosco, G., Renda, G., Rizzo, R., Gaglio, S., & Urso, A. (2018). Deep learning models for bacteria taxonomic classification of metagenomic data. *BMC Bioinformatics, 19*(Supplement 7), 198. https://doi.org/10.1186/s12859-018-2182-6

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

Gazi Üniversitesi. (t.y.). *FTIR spektroskopisi ders notları*. Gazi Üniversitesi Kimya Mühendisliği Bölümü.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning, 63*(1), 3–42. https://doi.org/10.1007/s10994-006-6226-1

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (2nd ed.). O'Reilly Media.

Graf, E., Soliman, A., Marouf, M., Parwani, A. V., & Pancholi, P. (2024). Potential roles for artificial intelligence in clinical microbiology: From improved diagnostic accuracy to solving the staffing crisis. *American Journal of Clinical Pathology*. Advance online publication. https://doi.org/10.1093/ajcp/aqae107

Hill, C., Guarner, F., Reid, G., Gibson, G. R., Merenstein, D. J., Pot, B., ... & Sanders, M. E. (2014). The International Scientific Association for Probiotics and Prebiotics consensus statement on the scope and appropriate use of the term probiotic. *Nature Reviews Gastroenterology & Hepatology, 11*(8), 506-514. https://doi.org/10.1038/nrgastro.2014.66

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression* (3rd ed.). Wiley.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering, 9*(3), 90-95. https://doi.org/10.1109/MCSE.2007.55

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters, 31*(8), 651–666. https://doi.org/10.1016/j.patrec.2009.09.011

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning: with applications in R* (2nd ed.). Springer.

Kandilci, M., Yakıcı, G., & Kayar, M. B. (2024). Artificial intelligence and microbiology.
Experimental and Applied Medical Science, 5(2), 119–128. https://doi.org/10.46871/eams.1458704

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems, 30*, 3146–3154. https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1412.6980

Kleinbaum, D. G., & Klein, M. (2010). *Logistic regression: A self-learning text* (3rd ed.). Springer.

Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data preprocessing for supervised leaning. *International Journal of Computer Science, 1*(2), 111–117.

Kumar, Y., Kaur, I., & Mishra, S. (2024). Foodborne disease symptoms, diagnostics, and predictions using artificial intelligence-based learning approaches: A systematic review. Archives of Computational Methods in Engineering, 31, 553–578. https://doi.org/10.1007/s11831-023-09991-0

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research, 18*(17), 1–5.

Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.

Liu, D., Caliskan, S., Rashidfarokhi, B., Oldenhof, H., Jung, K., Sieme, H., Hilfiker, A., & Wolkers, W. F. (2021). Fourier transform infrared spectroscopy coupled with machine learning classification for identification of oxidative damage in freeze-dried heart valves. *Scientific Reports, 11*, 12299. https://doi.org/10.1038/s41598-021-91802-2

Lutz, M. (2013). *Learning Python* (5th ed.). O'Reilly Media.

Ma, L., Yi, J., Wisuthiphaet, N., Earles, M., & Nitin, N. (2023). Accelerating the detection of bacteria in food using artificial intelligence and optical imaging. *Applied and Environmental Microbiology, 89*(1), e01828-22. https://doi.org/10.1128/aem.01828-22

Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research, 9*(Nov), 2579–2605.

Marcos-Zambrano, L. J., Karaduzovic-Hadziabdic, K., Loncar Turukalo, T., Przymus, P., Trajkovik, V., Aasmets, O., Berland, M., Gruca, A., Hasic, J., Hron, K., Klammsteiner, T., Kolev, M., Lahti, L., Lopes, M. B., Moreno, V., Naskinova, I., Org, E., Paciência, I., Papoutsoglou, G., Shigdel, R., Stres, B., Vilne, B., Yousef, M., Zdravevski, E., Tsamardinos, I., Carrillo de Santa Pau, E., Claesson, M. J., Moreno-Indias, I., & Truu, J. (2021). Applications of machine learning in human microbiome studies: A review on feature selection, biomarker identification, disease prediction and treatment. *Frontiers in Microbiology, 12*, Article 634511. https://doi.org/10.3389/fmicb.2021.634511

Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research, 11*, 169–198.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems, 32*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

Matuszewski, D. J., & Sintorn, I.-M. (2018). Minimal annotation training for segmentation of microscopy images. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)* (pp. 387-390). IEEE. https://doi.org/10.1109/ISBI.2018.8363599

Metchnikoff, E. (1907). *The prolongation of life: Optimistic studies*. New York: G.P. Putnam's Sons.

McKinney, W. (2012). *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. O'Reilly Media.

Mettler Toledo. (2023). *FTIR spektroskopisi: Uygulamalar ve avantajlar*. https://www.mt.com/

Müller, A. C., & Guido, S. (2016). *Introduction to machine learning with Python: A guide for data scientists*. O'Reilly Media.

Ng, A. Y. (2004). Feature selection, L1 vs. L2 regularization, and rotational invariance. *Proceedings of the Twenty-First International Conference on Machine Learning* (ICML 2004).

Nielsen, D. (2016). *Tree boosting with XGBoost: Why does XGBoost win "every" machine learning competition?* (Master's thesis, Norwegian University of Science and Technology).

Oliphant, T. E. (2006). *A guide to NumPy*. Trelgol Publishing.

Ozen, M., & Dinleyici, E. C. (2015). The history of probiotics: the untold story. *Beneficial Microbes, 6*(2), 159-165. https://doi.org/10.3920/BM2014.0103

Pan, Y., Ye, W., Xie, D., Wang, J., Wang, H., & Qiu, H. (2024). Outlier classification for microbiological open set recognition. *Computers and Electronics in Agriculture, 224*, 109104. https://doi.org/10.1016/j.compag.2024.109104

Park, H., Lim, S. J., Cosme, J., O'Connell, K., Sandeep, J., Gayanilo, F., Cutter, G. R., Montes, E., Nitikitpaiboon, C., Fisher, S., Moustahfid, H., & Thompson, L. R. (2023). Investigation of machine learning algorithms for taxonomic classification of marine metagenomes. *Microbiology Spectrum, 11*(5), e05237-22. https://doi.org/10.1128/spectrum.05237-22

Python Software Foundation. (2023). *Python 3 documentation - Time module*. https://docs.python.org/3/library/time.html

Portable Analytical Solutions. (2023). *A brief history of FTIR spectroscopy*. https://www.portableas.com/

Patnaik, P. R. (2009). Intelligent models of the quantitative behavior of microbial systems. *Food and Bioprocess Technology, 2*(2), 122–137. https://doi.org/10.1007/s11947-008-0112-8

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing, 45*(11), 2673–2681.

Shakibania, T., Arabfard, M., & Najafi, A. (2024). A predictive approach for host-pathogen interactions using deep learning and protein sequences. *VirusDisease, 35*(4), 434–445. https://doi.org/10.1007/s13337-024-00882-x

Sill, J., Takács, G., Mackey, L., & Lin, D. (2009). Feature-weighted linear stacking. *arXiv preprint arXiv:0911.0460*.

Smith, K. P., Wang, H., Durant, T. J. S., Mathison, B. A., Sharp, S. E., Kirby, J. E., Long, S. W., & Rhoads, D. D. (2020). Applications of artificial intelligence in clinical microbiology diagnostic testing. *Clinical Microbiology Newsletter, 42*(8), 61–70. https://doi.org/10.1016/j.clinmicnews.2020.03.006

Sofu, A. ., & Ekinci, N. D. V. F. Y. . (2007). Gıda Bilimi ve Teknolojisi Alanında Yapay Zekâ Uygulamaları. Gıda, 32(2), 93-99.

Sun, Y., Li, H., Zheng, L., Li, J., Hong, Y., Liang, P., Kwok, L.-Y., Zuo, Y., Zhang, W., & Zhang, H. (2022). iProbiotics: A machine learning platform for rapid identification of probiotic properties from whole-genome primary sequences. *Briefings in Bioinformatics, 23*(1), Article bbab477. https://doi.org/10.1093/bib/bbab477

Talon, R., Walter, D., Viallon, C., & Berdagué, J. L. (2002). Prediction of *Streptococcus salivarius* subsp. *thermophilus* and *Lactobacillus delbrueckii* subsp. *bulgaricus* populations in yoghurt by Curie point pyrolysis-mass spectrometry. *Journal of Microbiological Methods, 48*(3), 271–279. https://doi.org/10.1016/s0167-7012(01)00329-3

Tenenbaum, J. B., De Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science, 290*(5500), 2319–2323. https://doi.org/10.1126/science.290.5500.2319

Wahid, M. F., Hasan, M. J., & Alom, M. S. (2019). Deep convolutional neural network for microscopic bacteria image classification. *Proceedings of the 2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*, 866–869. https://doi.org/10.1109/ICAEE48663.2019.8975588

Wang, D., Keller, J. M., Carson, C. A., McAdo-Edwards, K. K., & Bailey, C. W. (1998). Use of fuzzy-logic-inspired features to improve bacterial recognition through classifier fusion. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 28*(4), 583-591. https://doi.org/10.1109/3477.704297

Waskom, M. (2021). Seaborn: Statistical data visualization. *Journal of Open Source Software, 6*(60), 3021. https://doi.org/10.21105/joss.03021

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks, 5*(2), 241–259.

Wu, S., Feng, T., Tang, W., Qi, C., Gao, J., He, X., Wang, J., Zhou, H., & Fang, Z. (2024). metaProbiotics: A tool for mining probiotic from metagenomic binning data based on a language model. *Briefings in Bioinformatics, 25*(2), Article bbae085. https://doi.org/10.1093/bib/bbae085

Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. Scotts Valley, CA: CreateSpace.

Venkatesh, G. P., Kuruvalli, G., Syed, K., & Reddy, V. D. (2024). An updated review on probiotic production and applications. *Gastroenterology Insights, 15*(1), 221–236. https://doi.org/10.3390/gastroent15010016

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... & SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental algorithms for

scientific computing in Python. *Nature Methods, 17*(3), 261-272. https://doi.org/10.1038/s41592-019-0686-2

Zhang, J., Li, C., Rahaman, M. M., Yao, Y., Ma, P., Zhang, J., Zhao, X., Jiang, T., & Grzegorzek, M. (2021). A comprehensive review of image analysis methods for microorganism counting: From classical image processing to deep learning approaches. *arXiv*. https://doi.org/10.48550/arXiv.2103.13625

Zhou, Z.-H. (2012). *Ensemble methods: Foundations and algorithms*. Chapman and Hall/CRC.