# MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE KALP HASTALIĞI TAHMİNİ

Güneş Gürsoy
201402101

**YÜKSEK LİSANS TEZİ**

Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı
Danışman: Prof. Dr. Asaf Varol

İstanbul
T.C. Maltepe Üniversitesi
Lisansüstü Eğitim Enstitüsü
Nisan, 2022

# MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE KALP HASTALIĞI TAHMİNİ

Güneş Gürsoy
201402101
ORCID: 0000-0003-3716-0334

**YÜKSEK LİSANS TEZİ**
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı
Danışman: Prof. Dr. Asaf Varol

İstanbul
T.C. Maltepe Üniversitesi
Lisansüstü Eğitim Enstitüsü
Nisan, 2022



# JÜRİ VE ENSTİTÜ ONAYI

Bu belge, Yükseköğretim Kurulu tarafından 19.01.2021 tarihli *"Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge"* ile bildirilen 6689 Sayılı Kişisel Verilerin Korunması Kanunu kapsamında gizlenmiştir.

# ETİK İLKE VE KURALLARA UYUM BEYANI

Bu belge, Yükseköğretim Kurulu tarafından 19.01.2021 tarihli *"Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge"* ile bildirilen 6689 Sayılı Kişisel Verilerin Korunması Kanunu kapsamında gizlenmiştir.

# TEŞEKKÜR

Tez çalışmamda verdikleri destekler için danışman hocam Prof. Dr. Asaf Varol'a ve Dr. Öğr. Üyesi Erdal Güvenoğlu'na, eğitim hayatımın sürekliliğine teşvik eden başta anneannem Ayşe Öğüt olmak üzere tüm aileme, manevi desteklerini esirgemeyen arkadaşlarım Doç. Dr. Egemen Avcu ve Arş. Gör. Dr. Yasemin Yıldıran Avcu'ya teşekkür ederim.

Güneş Gürsoy

Nisan, 2022

# ÖZ

## MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE KALP HASTALIĞI TAHMİNİ

Güneş Gürsoy
Yüksek Lisans Tezi
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı
Danışman: Prof. Dr. Asaf Varol
Maltepe Üniversitesi Lisansüstü Eğitim Enstitüsü, 2022

Son yıllarda hızlı bilgisayarların hayatımıza girmesi, geliştirilen algoritmalar ile yapay zeka teknikleri hızla ilerleyerek hemen her sektörde olduğu gibi tıp alanında da önemli gelişmelere yol açmaktadır. Tıp alanında kalp hastalıkların teşhisinde yapay zekanın alt dallarından olan makine öğrenmesi sıklıkla kullanılmaktadır. Tez çalışmasında makine öğrenmesinde bir sınıflandırma problemi olarak kalp hastalığına ait bir veri seti ile hastaların kalp hastalığının olup olmadığı hakkında bir tahminde bulunulmuştur. Mevcut veri setini denetimli öğrenme türünün altındaki sınıflandırma algoritmaları ile eğiterek her algoritma ile ayrı model oluşturulmuş daha sonra bu modeller ile mevcut test veri seti üzerinden tahminlerde bulunulmuştur. Uygulamada MATLAB programı ve sınıflandırma algoritmalarından lineer diskriminant analizi, k-en yakın komşu, lojistik regresyon, karar ağacı, destek vektör makineleri, Naive Bayes, toplu öğrenme ve sinir ağları algoritmaları kullanılmıştır. Alınan sonuçlarda test veri setinin doğruluk değerleri değerlendirildiğinde en yüksek doğruluk oranlarını lojistik regresyon ve toplu öğrenme algoritmaları %88.9 olarak vermiştir. Çalışmada tahminlere ait hata matrisleri ve ROC eğrilerine de yer verilmiş, algoritmaların doğruluk oranları tablolar halinde gösterilmiştir.

**Anahtar Sözcükler:** Makine Öğrenmesi, Kalp Hastalığı Tahmini, Denetimli Öğrenme Algoritmaları.

# ABSTRACT

## PREDICTION OF HEART DISEASE WITH MACHINE LEARNING ALGORITHMS

Güneş Gürsoy
Master Thesis
Graduate School
Computer Engineering with Thesis
Thesis Advisor: Prof. Dr. Asaf Varol
Maltepe University Graduate School, 2022

In recent years, fast computers have entered our lives, improved algorithms and artificial intelligence techniques have progressed rapidly. Significant developments have been experienced in the field of medicine as in almost every sector. Machine learning, which is one of the sub-branches of artificial intelligence, is frequently used to diagnose heart diseases in medical field. In the thesis study, a dataset of heart disease (heart-statlog) was used to predict whether patients have heart disease or not. By training the existing data set with the classification algorithms under the supervised learning type, a separate model was created with each algorithm, and then predictions were made on the existing test data set with these models. MATLAB program and classification algorithms linear discriminant analysis, k-nearest neighbor, logistic regression, decision trees, support vector machines, Naive Bayes, ensemble learning and neural networks algorithms were used to analyse the data. When the accuracy values of the test data set were evaluated, the logistic regression and ensemble learning algorithms gave the highest accuracy rates as 88.9%. In the study, the confusion matrices and ROC curves of the predictions are also included, and the accuracy rates of the algorithms are given in tables.

**Keywords:** Machine Learning, Prediction Of Heart Disease, Supervised Learning Algorithms.

# İÇİNDEKİLER

JÜRİ VE ENSTİTÜ ONAYI ..... ii

ETİK İLKE VE KURALLARA UYUM BEYANI ..... iii

TEŞEKKÜR ..... iv

ÖZ ..... v

ABSTRACT ..... vi

İÇİNDEKİLER ..... vii

TABLOLAR LİSTESİ ..... x

ŞEKİLLER LİSTESİ ..... xi

KISALTMALAR LİSTESİ ..... xiii

ÖZGEÇMİŞ ..... xiv

BÖLÜM 1. GİRİŞ ..... 1

1.1. Literatür Araştırması ..... 2

1.2. Kalp Hastalıkları ..... 6

1.3. Kalp Hastalığı Türleri ..... 6

1.4. Kardiyovasküler Hastalıklar İçin Risk Faktörleri ..... 7

BÖLÜM 2. YAPAY ZEKA ..... 8

2.1. Makine Öğrenmesi ..... 9

2.2. Makine Öğrenmesinin Metotları ..... 10

2.2.1. Denetimli Öğrenme ..... 11

2.2.1.1. Lineer Diskriminant Analizi ..... 11

2.2.1.2. K-En Yakın Komşu Algoritması ..... 13

2.2.1.3. Lojistik Regresyon ..... 15

2.2.1.4. Karar Ağacı ..... 16

2.2.1.5. Destek Vektör Makineleri ..... 18

2.2.1.5.1. Keskin Kenar Payı (Hard Margin) ..... 18

2.2.1.5.2. Esnek Kenar Payı (Soft Margin) ..... 20

2.2.1.5.3. Çekirdekler ..... 20

2.2.1.6. Naive Bayes Sınıflandırıcısı ..... 22

2.2.1.6.1. Naive Bayes Sınıflandırıcısının Formülü ..... 22
2.2.1.7. Toplu Öğrenme Algoritmaları ..... 22
2.2.1.7.1. Torbalama (Bagging) ..... 23
2.2.1.7.2. Yükseltme (Boosting) ..... 23
2.2.1.7.3. LogitBoost ..... 23
2.2.1.8. Sinir Ağları ..... 25
2.2.2. Denetimsiz Öğrenme ..... 27
2.2.3. Yarı Denetimli Öğrenme ..... 27
2.2.4. Takviyeli Öğrenme ..... 27
BÖLÜM 3. YÖNTEM ..... 28
3.1. Veriler ve Toplanması ..... 28
3.2. Makine Öğrenmesinde Sınıflandırma Performans Ölçütleri ..... 29
3.3. Çapraz Doğrulama ..... 29
3.4. Performans Ölçütleri ..... 30
3.4.1. Doğruluk (Accuracy) ..... 31
3.4.2. Kesinlik (Presicion) ..... 31
3.4.3. Duyarlılık (Recall) ..... 31
3.4.5. ROC Eğrisi ..... 31
3.5. Verilerin Çözümlenmesi ve Yorumlanması ..... 32
BÖLÜM 4. BULGULAR VE YORUMLAR ..... 33
4.1. Bulgular ..... 33
4.1.1. Lineer Diskriminant Analizi ..... 33
4.1.2. K-En Yakın Komşu Algoritması ..... 35
4.1.3. Lojistik Regresyon ..... 36
4.1.4. Karar Ağacı ..... 38
4.1.5. Destek Vektör Makineleri ..... 39
4.1.6. Naive Bayes Algoritması ..... 41
4.1.7. Toplu Öğrenme Algoritmaları ..... 42
4.1.8. Sinir Ağları ..... 44
4.2. Yorumlar ..... 46
BÖLÜM 5. SONUÇ ..... 48

KAYNAKÇA ..... 49

# TABLOLAR LİSTESİ

**Tablo 2.1** - KNN Algoritmasının Avantajları ve Dezavantajları ..... 14
**Tablo 2.2** - Karar Ağacı Algoritmaları..... 17
**Tablo 3.1** - Öznitelik Tablosu ..... 28
**Tablo 4.1** - LDA Algoritması Test Veri Seti Doğruluk Oranları..... 33
**Tablo 4.2** - KNN Algoritması Test Veri Seti Doğruluk Oranları ..... 35
**Tablo 4.3** - LR Algoritması Test Veri Seti Doğruluk Oranları..... 36
**Tablo 4.4** - KA Algoritması Test Veri Seti Doğruluk Oranları ..... 38
**Tablo 4.5** - DVM Test Veri Seti Doğruluk Oranları..... 39
**Tablo 4.6** - NB Algoritması Test Veri Seti Doğruluk Oranları..... 41
**Tablo 4.7** - TÖ Algoritması Test Veri Seti Doğruluk Oranı..... 42
**Tablo 4.8** - SA Algoritması Doğruluk Oranları ..... 44
**Tablo 4.9** - Algoritmaların Eğitim Sonuçları..... 45
**Tablo 4.10** - Algoritmaların Test Veri Seti Doğruluk Oranları.....46

# ŞEKİLLER LİSTESİ

**Şekil 2.1** - Makine Öğrenmesi Metotları ..... 11
**Şekil 2.2** - Lineer Diskriminant Analizi ..... 12
**Şekil 2.3** - KNN Algoritması Genel Yapısı ..... 13
**Şekil 2.4** - Hamming Mesafesi Örneği ..... 14
**Şekil 2.5** - Sigmoid Fonksiyonu ..... 15
**Şekil 2.6** - LR Genel Gösterimi ..... 16
**Şekil 2.7** - Karar Ağacı Algoritması Genel Yapısı ..... 17
**Şekil 2.8** - Destek Vektör Makineleri ..... 19
**Şekil 2.9** - DVM Esnek Kenar Payı ..... 20
**Şekil 2.10** - DVM Sınıfların Doğrusal Olarak Ayrılması ..... 21
**Şekil 2.11** - DVM Doğrusal Olarak Ayrılamayan Sınıflar ..... 21
**Şekil 2.12** - DVM Çekirdekler Yardımı ile Sınıfların Ayrılması ..... 21
**Şekil 2.13** - Toplu Öğrenme Algoritmaları Genel Yapısı ..... 25
**Şekil 2.14** - Sinir Ağları Genel Yapısı ..... 26
**Şekil 2.15** - Yapay Nöronların Çalışma Prensibi ..... 26
**Şekil 3.1** - Çapraz Doğrulama Örneği ..... 30
**Şekil 3.2** - Hata Matrisi Genel Yapısı ..... 30
**Şekil 4.1** - LDA Algoritması Test Veri Seti Hata Matrisi ..... 34
**Şekil 4.2** - LDA Algoritması Test Veri Seti ROC Eğrisi ..... 34
**Şekil 4.3** - KNN Algoritması Test Veri Seti Hata Matrisi ..... 35
**Şekil 4.4** - KNN Algoritması Test Veri Seti ROC Eğrisi ..... 36
**Şekil 4.5** - LR Algoritması Test Veri Seti Hata Matrisi ..... 37
**Şekil 4.6** - LR Algoritması Test Veri Seti ROC Eğrisi ..... 37
**Şekil 4.7** - KA Algoritması Test Veri Seti Hata Matrisi ..... 38
**Şekil 4.8** - KA Algoritması Test Veri Seti ROC Eğrisi ..... 39
**Şekil 4.9** - DVM Algoritması Test Veri Seti Hata Matrisi ..... 40
**Şekil 4.10** - DVM Algoritması Test Veri Seti ROC Eğrisi ..... 40
**Şekil 4.11** - NB Algoritması Test Veri Seti Hata Matrisi ..... 42
**Şekil 4.12** - NB Algoritması Test Veri Seti ROC Eğrisi ..... 42
**Şekil 4.13** - TÖ Algoritması Test Veri Seti Hata Eğrisi ..... 43

**Şekil 4.14** - TÖ Algoritması Test Veri Seti ROC Eğrisi ..... 44
**Şekil 4.15** - SA Algoritması Test Veri Seti Hata Matrisi ..... 45
**Şekil 4.16** - SA Algoritması Test Veri Seti ROC Eğrisi ..... 45

# KISALTMALAR LİSTESİ

**ROC** : Receiver Operating Characteristic

**AUC** : Area Under The Curve

**k** : K-En Yakın Komşu algoritması komşu sayısı

**LDA** : Lineer Diskriminant Analizi

**KNN** : K-En Yakın Komşu

**LR** : Lojistik Regresyon

**KA** : Karar Ağacı

**DVM** : Destek Vektör Makineleri

**NB** : Naive Bayes

**TÖ** : Toplu Öğrenme

**SA** : Sinir Ağları

# ÖZGEÇMİŞ

**Güneş Gürsoy**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Eğitim**

| *Derece* | *Yıl* | *Üniversite, Enstitü, Anabilim Dalı* |
|---|---|---|
| Ls. | 2020 | Hoca Ahmet Yesevi Uluslararası Türk Kazak Üniversitesi, Mühendislik Fakültesi, Yönetim Bilişim Sistemleri |
| Lise | 1996 | İzmit Lisesi |

**Mesleki Birlik/Dernek Üyelikleri**

| *Yıl* | *Kurum* |
|---|---|
| 2021 - | Üye: Association of Software and Cyber Security, Turkey |

**Yayınlar ve Diğer Bilimsel/Sanatsal Faaliyetler**

2021 - G. Gürsoy and A. Varol, "Prediction of Arrhythmia with Machine Learning Algorithms," *2021 9th International Symposium on Digital Forensics and Security (ISDFS)*, 2021, pp. 1-5, doi: 10.1109/ISDFS52919.2021.9486383.

2022 - G. Gürsoy and A. Varol, "Risks of Digital Transformation: Review of Machine Learning Algorithms in Credit Card Fraud Detection," *2021 2nd International Informatics and Software Engineering Conference (IISEC)*, 2021, pp. 1-6, doi: 10.1109/IISEC54230.2021.9672354.

# BÖLÜM 1. GİRİŞ

Teknolojinin getirdiği olanaklar her sektörde kullanılmaya başlamış, işletmelerin iş yapış şekilleri de her alanda değişmiştir. Sağlık alanında da yaşanan gelişmelerin, üretilen teknolojilerin yanında en önemli konu hastalıkların erken teşhisidir. Erken teşhisler insanların karşılaşacakları tehlikelerle yüz yüze gelmelerini önleyebilir, hayatlarını kurtarabilir ve sağlık masraflarını da önemli ölçüde azaltabilir. Dünya Sağlık Örgütü'nün yaptığı araştırmalara göre dünya üzerinde meydana gelen ölüm sebeplerinin başında kalp hastalıkları gelmektedir. Kalp hastalığı sebebi ile hayatını kaybeden insan sayısının 2019 yılında 17.9 milyon olduğu tahmin edilmektedir. Dünya Sağlık Örgütü bu sayının araştırmanın yapıldığı zamanda dünya üzerinde hayatını kaybeden insanların %32'sine denk geldiğini belirtmektedir. Bu oranın bu kadar ciddi olması teşhis yöntemlerinin de önemini göstermektedir. Günümüzde birçok teşhis yöntemi olmasına rağmen teknolojinin gelişmesi ile geçmiş yıllarda gelişme imkânı bulma da zorlanan yapay zeka alanları inanılmaz bir şekilde hızlanarak çözümler üretmeye başlamıştır. Yapay zekanın alt dallarından biri olan makine öğrenmesi algoritmaları mevcut veri setleri ile eğitilerek tahminlerde bulunulmasını sağlayabilmektedir. Makine öğrenmesi algoritmaları ile oluşturulan modellerden yüksek doğruluk oranına sahip algoritmalar yeni hasta kayıtlarında teşhisler için kullanılabilir.

Tez çalışmasının amacı insanların kalp hastalıkları ile ilgili mevcut bir veri seti üzerinden makine öğrenmesinde denetimli öğrenme türünün sınıflandırma algoritmalarını kullanarak hastaların kalp hastalığına yakalanma riskleri hakkında tahminler yapmaktır.

Tez çalışmasında uygulamada kullanılan veri seti önceki yıllara ait güncel olmayan bir veri setidir ancak bu veri seti algoritmalarla model oluşturmaya ve tahmin sonuçları elde edilmesine engel değildir. Sağlık kuruluşlarından güncel veri setlerine ulaşmak bu araştırmanın başlıca sınırlıklarındandır.

## 1.1. Literatür Araştırması

Literatürde makine öğrenmesi algoritmaları kullanılan birçok medikal araştırma yapılmıştır.

M. M. Ali ve arkadaşları hazırladıkları çalışmalarında veri setinden kalp hastalığı riskini değerlendirmek için makine öğrenmesi türlerinden olan denetimli öğrenme algoritmalarından lojistik regresyon, adaboosM1, çoklu katman algılayıcı(MLP), k-en yakın komşu algoritması, karar ağacı, rastgele orman algoritması ile performanslarını karşılaştırmışlar ve sonuç olarak k-en yakın komşu algoritması, karar ağacı ve rastgele orman algoritması'nın en başarılı sonucu verdiğini bulmuşlardır [1].

K. A. Jothi ve arkadaşları kalp hastalığı risk düzeyini tahmin etmek için 303 veri örneği ve yaş, cinsiyet, göğüs ağrısı, kan basıncı gibi 14 parametre içeren kalp hastalığı veri setinin %80'ini eğitim seti olarak ve %20 'sini test veri seti olarak belirlemişlerdir. Önerdikleri tahmin modelinde makine öğrenmesi algoritmalarından k-en yakın komşu algoritması ve karar ağacı algoritmasını kullanarak k-en yakın komşu algoritması ile %67 doğruluk oranı ve karar ağacı algoritması ile %81 doğruluk oranlarına ulaşmışlardır [2].

M. Diwakar ve diğerleri makine öğrenmesi algoritmaları ve görüntü birleştirme kullanarak yaptıkları çalışmada yapay sinir ağlarının kalp hastalığında iyi sonuç verdiğini gözlemlemişlerdir [3].

A. Kondababu ve arkadaşları R Studio Rattle Tool ile UCI Machine Learning Repository 'den aldıkları Cleveland veri setini kullanmışlardır. Toplam 303 hasta kayıdı ve hastalara ait yaş, cinsiyet, kolesterol gibi 14 niteliği makine öğrenmesi algoritmalarını kullanarak doğruluk oranlarını karşılaştırmışlardır. Naive Bayes %75.8, lineer model %85.1, lojistik regresyon %82.9, karar ağacı %85, rastgele orman algoritması %86.1, destek vektör makineleri %86.1 ve HRFLM (Hybrid Random Forest with Linear Model) %88.4 oranlarını bulmuşlardır. En yüksek doğruluk oranını rastgele orman ve lineer modelin birleşimi olan HRFLM algoritması vermiştir [4].

R. Hagan ve arkadaşları kalp hastalığını tahmini için hazırladıkları çalışmalarında University of California Irvine (UCI) Machine Learning Repository 'den aldıkları kardiyak aritmi veri setini ve Kaggle'dan aldıkları kalp hastalığı veri setini kullanmışlardır. Kalp hastalığını tahmininde makine öğrenmesi algoritmalarından destek vektör makineleri, çok katmanlı algılama sinir ağları ve toplu öğrenme algoritmalarını

kullanmışlardır. UCI veri setinde destek vektör makineleri ile %92 oranında doğruluk bulmuşlar, destek vektör makineleri kullanarak Kaggle veri seti için bu oranı %72 olarak bulmuşlardır. Çok katmanlı algılama sinir ağları kullanarak UCI veri seti için %74 doğruluk oranına ulaşmışlar ve Kaggle veri seti için bu oranı %71 olarak bulmuşlardır. Toplu öğrenme yöntemleri ile UCI veri setinde %90'nın üzerinde doğruluk oranına ulaşmışlardır ve Extra Tree modeli ile %96 doğruluk oranına ulaşmışlardır. Sonuç olarak kullandıkları modeller içinde karar ağaçlarının en az maliyet ile en iyi doğruluğu verdiklerini kanıtlamışlardır [5].

V. Shorewala yaptığı çalışmada kalp hastalığı tahmini için Kaggle'dan aldığı 70 bin hastanın kaydı bulunan koroner kalp hastalığı veri seti kullanmıştır. Çalışmasında makine öğrenmesi algoritmalarından olan lojistik regresyon, destek vektör makineleri, k-en yakın komşu algoritması, karar ağacı, Naive Bayes ve sinir ağı algoritmaları kullanmış ve bu teknikleri toplu öğrenme algoritmalarından olan torbalama (Bagging), arttırma (Boosting) ve istifleme (Stacking) teknikleri ile karşılaştırmıştır. Shorewala yaptığı çalışmada rastgele orman sınıflandırıcısının %74.4 doğruluk oranına ve karar ağacı algoritmasını %74.8 doğruluk oranına ulaştığını bulmuştur [6].

S. P. Patro ve diğerleri hazırladıkları kalp hastalığını tahmin etme çalışmalarında UCI Machine Repository'den edindikleri kalp hastalığı veri setini %80 eğitim veri seti ve %20 test veri seti olarak kullanmışlardır. Çalışmalarında makine öğrenmesinin denetimli öğrenme algoritmalarından Naive Bayes, B.O. destek vektör makineleri (Bayesian Optimized Support Vector Machines), k-en yakın komşu algoritması ve S.S.O. sinir ağı (Salp Swarm Optimized Neural Network) kullanmışlardır ve en yüksek doğruluk oranını %93.3 oranı ile B.O. destek vektör makineleri (Bayesian Optimized Support Vector Machines) vermiştir [7].

D. Shah ve diğerleri UCI Machine Learning Repository'den aldıkları Cleveland veri setinde kalp hastalığı tahmininde 303 hasta kaydı ve hastalara ait 14 özniteliği kullanmışlardır. Analizlerinde WEKA yazılımını ve makine öğrenmesi türlerinden olan denetimli öğrenme algoritmalarından Naive Bayes, karar ağacı, k-en yakın komşu algoritması ve rastgele orman algoritmalarını tahminde kullanmışlar ve en yüksek doğruluk sonucunu k=7 değeri ile k-en yakın komşu algoritmasında bulmuşlardır [8].

A. Rajdhan ve diğerleri kalp hastalığı tahmini için yaptıkları çalışmada Cleveland veri setini UCI Machine Learning Repository 'den almışlardır. Tahmin çalışmalarında Naive Bayes, karar ağacı, lojistik regresyon ve rastgele orman algoritmalarını kullanmışlardır. Yapılan çalışma sonucunda Naive Bayes %85.25, karar ağacı %81.97, lojistik regresyon %85.25 doğruluk oranları bulunmuş ve en yüksek sonunu rastgele orman algoritması %90.16 oranında verdiğini bulmuşlardır [9].

M.Furqan ve diğerleri kalp hastalığı tahmini için yaptıkları çalışmada veri setini the University of California, Irvine (UCI) 'den almışlardır. Tahmin modeli için makine öğrenmesi algoritmalarından lojistik regresyon sınıflandırıcısı, k-en yakın komşu sınıflandırıcısı ve rastgele orman sınıflandırıcısı kullanmışlardır. Gerçekleştirdikleri analizde en yüksek doğruluk oranını lojistik regresyon sınıflandırıcısı vermiştir [10].

R.Katarya ve S. K. Meena UCI Repository'den aldıkları veri setini kullanarak kalp hastalığı tahmininde bulunmuşlardır. Veri setindeki 76 öznitelikten 14'ünü kullanmışlardır. Makine öğrenmesi algoritmalarından lojistik regresyon, Naive Bayes, destek vektör makineleri, k-en yakın komşu algortiması, karar ağacı, rastgele orman algoritması, yapay sinir ağları, derin sinir ağı ve çok katmanlı algılayıcı algoritmalarını kullanarak karşılaştırmalı analiz yapmışlardır. Sonuçların doğruluk değerlerini değerlendirmek için kök ortalama kare hata, ortalama mutlak hata, kesinlik ve duyarlılık tekniklerini kullanmışlardır. Bu analizlerinin sonucunda lojistik regresyon %93.40, destek vektör makineleri %92.30 ve yapay sinir ağları %92.30 olarak birbirine yakın sonuçlar vermesine rağmen rastgele orman algoritması %95.60 doğruluk oranı ile en iyi sonucu vermiştir [11].

R.Bharti ve diğerleri kalp hastalığı tahmini için UCI'den aldıkları Cleveland, Hungary, Switzerland ve Long Beach V isimli veri setlerini içeren 1988 yılına ait Public Health Dataset veri setini ve bu veri setindeki 76 öznitelikten 14 özniteliği kullanmışlardır. Çalışmalarında makine öğrenmesi algoritmalarını kullanmışlar ve buldukları sonuçlarda lojistik regresyon %83.3, k-en yakın komşu algoritması %84.8, destek vektör makineleri %83.2, rastgele orman algoritması %80.3, karar ağacı %82.3, derin öğrenme algoritması %94.2 doğruluk oranlarını vermiştir. En başarılı sonucu derin öğrenme tekniği (%94.2 ) vermiştir [12].

A. K. Dwivedi makine öğrenmesi algoritmaları ile kalp hastalığı tahmininde bulunmuştur. UCI'den aldığı StatLog Heart Disease isimli veri setini ile algoritmaların performanslarını değerlendirmek için ROC (Receiver Operative Characteristic - Alıcı İşletim Karakteristiği) eğrisini ve kalibrasyon grafiklerini kullanmıştır. Çalışmasında kullandığı algoritmalardan yapay sinir ağı %84, destek vektör makineleri %82, lojistik regresyon %85, k-en yakın komşu algoritması %80, ağaç sınıflandırıcı algoritması %77 ve Naive Bayes %83 sonucunu vermiştir. Algoritmalar içinde en yüksek doğruluk oranını lojistik regresyon %85 oranı ile vermiştir [13].

E. Maini ve diğerleri yayınladıkları çalışmalarında kalp hastalığı tahmini için güney Hindistan'da bir hastaneden içinde 1670 kayıt bulunan anonimleştirilmiş bir veri seti seçmişler bu veri setinin %70'ini (1169 kayıt) eğitim veri seti ve %30'unu (569 kayıt) test verisi olarak kullanmışlardır. Python programlama dilini kullanarak yaptıkları tahmin çalışmalarında makine öğrenmesi algoritmalarından k-en yakın komşu algoritması, Naive Bayes, lojistik regresyon, adaboost ve rastgele orman algoritmalarını kullanmışlardır. Bu algoritmalar içinde en yüksek doğruluk oranını rastgele orman algoritması %93.8 oranı ile vermiştir [14].

R. R Sanni ve diğerleri kalp hastalığı ile ilgili veri setinden aldıkları 377.650 kayıdı kalp hastalığı tahmini için kullanmışlardır. Yöntem olarak Python programlama dilini ve makine öğrenmesinde denetimli öğrenme algoritmalarından karar ağacı, lojistik regresyon, k-en yakın komşu ve rastgele orman algoritmalarını kullanmışlardır. Çalışmalarının sonucunda en yüksek doğruluk oranını %85.33 oranı ile karar ağacı vermiştir [15].

S. Nikhar ve diğerleri kalp hastalığı tahmini için 303 kayıt ve 76 öznitelikten oluşan Cleveland Heart Disease veri setini kullanmışlardır. Öznitelik sayısını 19 adete indirmişler, veri setini eşit olarak eğitim seti ve veri seti olarak ayırmışlardır. Analizlerinde Naive Bayes ve karar ağacı algoritmalarını karşılaştırmışlar ve karar ağacının daha iyi sonuç verdiğini gözlemlemişlerdir [16].

S.Barik ve diğerleri kalp hastalığı tahmini çalışmalarında UCI Repository'den temin ettikleri 303 hasta kaydı ve her kayda ait 14 öznitelik içeren Cleveland veri setini kullanmışlardır. Çalışmada Python programlama dili ve makine öğrenmesi algoritmalarından k-en yakın komşu algoritması, karar ağacı ve rastgele orman

algoritmalarını kullanmışlar ve en yüksek doğruluk oranını k-en yakın algoritması k=12 değeri ile % 85.066 olarak bulmuşlardır [17].

A. Sridhar ve diğerleri kalp hastalığı tahmini çalışmasında Kaggle'dan aldıkları veri setini kullanmışlardır. Çalışmalarında Python programlama dilini ve makine öğrenmesi algoritmalarından Naive Bayes ve karar ağacı algoritmalarını seçmişler ve analizleri sonucunda karar ağacının daha iyi sonuç verdiğini gözlemlemişlerdir [18].

H. Jindal ve diğerleri makine öğrenimi algoritmaları kullanarak kalp hastalığı tahmini isimli çalışmalarında kalp hastalığı tahmini için UCI'den temin ettikleri kalp hastalığı veri setini ve algoritma olarak k-en yakın komşu algoritması, lojistik regresyon ve rastgele orman algoritmalarını kullanmışlardır. Çalışmalarının sonucunda k-en yakın komşu algoritması ve lojistik regresyon %88.5 doğruluk oranı ile en iyi sonucu verdiğini gözlemlemişlerdir [19].

Mythili T. ve diğerleri UCI Repository'den temin ettikleri Cleveland Heart Disease veri setini kullanarak yaptıkları tahmin çalışmasında doğru bir tahmine ulaşmak için makine öğrenmesi algoritmalarından destek vektör makineleri, karar ağacı ve lojistik regresyon algoritmalarından oluşan bir çerçeve önermektedirler [20].

## 1.2. Kalp Hastalıkları

Teknolojinin yaşantımızın içine girmesi ile insanların yaşam tarzları büyük ölçüde değişikliğe uğradı. Teknolojik yeniliklerin avantajlarının yanında hareketsizlik, stres, beslenme alışkanlıklarının değişmesi gibi dezavantajlar sağlık açısından insanları olumsuz yönde etkilemektedir [21].

## 1.3. Kalp Hastalığı Türleri

Dünya Sağlık Örgütü (WHO) kalp hastalıklarını aşağıdaki şekilde gruplandırmış ve tanımlamıştır [21].

- **Koroner Kalp Hastalığı** : Kalp kasını besleyen kan damalarının neden olduğu kalp hastalığıdır.
- **Serebrovasküler Kalp Hastalığı** : Beyni besleyen kan damarlarının neden olduğu kalp hastalığıdır.

- **Periferik Arter Hastalığı** : Kolları ve bacakları besleyen kan damarlarının hastalığıdır.
- **Romatizmal Kalp Hastalığı** : Streptokok bakterilerinin neden olduğu kalp hastalığıdır.
- **Konjenital Kalp Hastalığı** : Doğumdan itibaren gelen, kalbin normal gelişimin ve işleyişinin etkilendiği kalp hastalığıdır.
- **Derin Ven Trombozu (DVT) ve Pulmoner Emboli** : Bacak damarlarından kalbe ve akciğerlere giden kan pıhtılarının neden olduğu hastalıktır.

## 1.4. Kardiyovasküler Hastalıklar İçin Risk Faktörleri

Tütün kullanımı, sağlıksız beslenme, fiziksel hareketsizlik, alkol kullanımı, obezite, diyabet ve hipertansiyon bireylerin kardiyovasküler hastalıklara maruz kalmaları için risk faktörleri oluşturmaktadır [21].

# BÖLÜM 2. YAPAY ZEKA

Yapay zeka bir bilgisayar bilimi olarak 1950'lerin ortalarında ortaya çıkmış [22] ve günümüze kadar bu kavramı oluşturan görüntü işleme, yapar sinir ağları, makine öğrenmesi, doğal dil işleme, genetik algoritmalar, robotbilim, uzman sistemler ve bulanık mantık gibi farklı metotlar üzerinde çalışmalar geliştirilerek devam etmiştir. Amerika'da 1956 yılında Dartmouth Kolejinde yapılan Dartmouth Konferansı'nda "Yapay Zeka" terimi ilk defa John McCarthy tarafından önerilmiştir [23]. Yapay zeka tarihinin İngiliz matematikçi Alan Mathison Turing'in 1950 yılında Mind dergisinde yayınlanan "Computing Machinery and Intelligence" adlı makalesinde bahsettiği "Makineler Düşünebilir mi?" (Can Machine Think ? ) sorusu ile başladığı kabul edilir [24]. John McCarthy de "What is Artificial Intelligence? " adlı makalesinde yapay zeka üzerine çalışmalar yapan kişiler arasında ilk olarak Alan Turing'in olabileceğinden bahsetmiş ve yapay zekanın makineler yapmak yerine programlanmış bilgisayarlar ile araştırılabileceğine karar veren ilk kişinin Alan Turing'in olabileceğini belirtmiştir[25].

Alan Turing "Computing Machinery and Intelligence" adlı makalesinde "Taklit Oyunu" (Imitation Game) ismini verdiği oyunda üç oyuncudan bahseder[24]. Bu oyunculardan birisi erkek (A), diğeri kadın (B) ve sorgulayıcı (C) erkek ya da kadın olarak tanımlanmıştır. Sorgulayıcı diğer ikisinin önünde, onları göremeyeceği bir odada bulunur. Sorgulayıcının sorduğu sorulara karşı cevap şeklinde ses tonlarının belirleyici olmaması için cevaplar yazılı olarak verilerek sorgulayıcı cevabın kime (A ya da B) ait olduğunu tahmin etmeye çalışır. Oyunda A'nın amacı sorgulayıcıyı (C) yanıltmak, B'nin amacı sorgulayıcıya yardım etmektir. Alan Turing'in sorusu bu oyunda örneğin A kişisinin yerini bir makine alırsa ne olacağı, "Makineler Düşünebilir mi ?" sorusudur [24]. Sorgulayıcı karşısındakilerin hangisinin makine hangisinin kadın ya da erkek olduğunu bilebilir mi? Makinelerin düşünmesi üzerine ortaya atılan bu fikir yapay zeka çalışmalarının başlangıcı olarak görülür.

Literatürde birçok yapay zeka tanımı yapılmıştır. Kesin bir tanımı olmamasına rağmen yapay zeka hakkında yapılan tanımların bir kısmına aşağıda yer verilmiştir.

Yapay zeka canlılara özgü özellikleri taklit edebilen, geçmiş bilgiler üzerinden öğrenebilen, bu bilgileri yorumlayabilen ve sonuçlar üretebilen bilgisayar sistemleri olarak tanımlanabilir. Yapay zeka programlama dillerinden biri olan LISP'i geliştiren, zaman paylaşım sistemi ile bugünkü sunucu ve bulut bilişimin temellerini atan ve yapay zeka terimini ilk olarak öneren John McCarthy 2004 yılında yayınlanan "What is Artificial Intelligence?" adlı makalesinde yapay zekayı "Akıllı makineler, özellikle akıllı bilgisayar programları yapma bilimi ve mühendisliğidir." olarak tanımlamaktadır [25]. John McCarthy ile yapay zekanın öncülerinden olan Marvin Lee Minsk'ye göre yapay zeka insanlar tarafından yapılabilen ve zeka gerektiren şeyleri makinelere yaptırma bilimidir [26]. Yapay zeka çeşitli durumlarda nasıl etkili ve güvenli bir şekilde hareket edebileceğini hesaplayabilen akıllı makineler oluşturmakla ilgilidir [27].

Yukarıda bahsedildiği gibi literatürde birçok farklı yapay zeka tanımı yapılmıştır. Yapay zekanın zeki varlıkların davranışlarını taklit ederek yardımcı olması, problemlere karşı çözümler üretmesi, insana hizmet edecek uygulamalarda yer alması amaçlanmaktadır. Teknolojinin ilerlemesi ile paralel bir şekilde gelişen yapay zeka metotları birçok alanda insanların hayatlarına kolaylıklar sağlamaktadır. Günlük iş hayatında yardımcı olan sanal asistanlar, sürücüsüz araçlar, insanların tercihlerine göre (alışveriş, müzik, film vb.) önerilerin sunulması, kronik hastalıkların tedavi edilmesi, iklim değişikliği ile mücadele, siber güvenlik tehditlerin tahmini gibi zorlu problemlerde yapay zeka yardımcı olmaktadır [28]. Ayrıca yapay zeka uygulamaları ticaret, finans, üretim, sağlık hizmetleri, ulaşım, eğitim, eğlence ve sosyal etkileşimler gibi konularda toplumları artan bir hızda etkilemeye devam etmektedir. Yapay zeka tek başına bir bilim dalı olarak değil bilgisayar bilimleri, matematik, fizik, biyoloji, felsefe, psikoloji, dil bilgisi, fizyoloji, elektronik ve güzel sanatlar gibi birçok alan ile ilişkilidir. Belirtilen bu disiplinler görüntü işleme, yapar sinir ağları, makine öğrenmesi, doğal dil işleme, genetik algoritmalar, robotbilim, uzman sistemler ve bulanık mantık gibi yapay zekanın alt dallarında kullanılmaktadır.

## 2.1. Makine Öğrenmesi

Makine öğrenmesi, bir sistemin açık bir şekilde programlama gerektirmeden mevcut veriden öğrenmesini sağlayan yapay zekanın alt dallarından biridir [29]. Makine

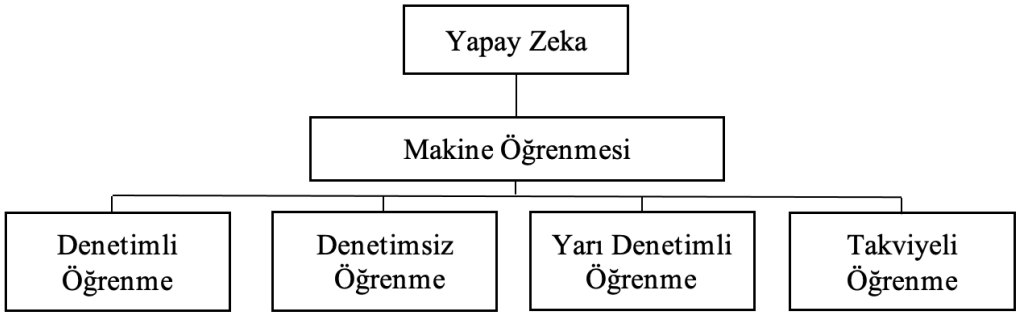
öğrenmesinde veri analiz edilerek gelecekteki sonuçlar için tahminde bulunması sağlanır [30]. Girdi olarak verilen veriye bir takım matematiksel formülleri içeren algoritmalar uygulanarak istenen çıktının tahmin edilmesi amaç edinilir [31].

Makine Öğrenmesi terimi ilk olarak bilgisayar oyunlarının ve yapay zeka alanının öncülerinden olan Arthur Lee Samuel tarafından kullanılmış [31] ve kendisi bu yaklaşımını 1959 yılında "IBM Journal of Research and Development" dergisinde yayınlanan "Some Studies in Machine Learning Using the Game of Checkers" isimli makalesinde açıklamıştır [29]. Bu makalede makine öğrenmesi prosedürünü dama oyununun bazı özellikleri üzerinden araştırmış, programın oyunun kurallarının verildiğinde kısa sürede öğrendiğini ve bilgisayarın oyunu yazan programcıdan daha iyi oynadığını doğrulamıştır [32]. Ayrıca bu deneyde doğrulanan ilkelerin başka durumlarda da uygulanabileceğini belirtmiştir [32].

Makine öğrenmesi hastalık teşhisi, petrol endüstrisi, biyoloji, tarım, güç sistemleri ve birçok alanda sınıflandırma, regresyon ve kümeleme problemlerinde kullanılmaktadır. Ayrıca günümüzde gittikçe popülerleşen dijital dönüşüm alanında da makine öğrenmesi işletmelerin müşterilerine değer katan işler yapmasını sağlamaktadır.

## 2.2. Makine Öğrenmesinin Metotları

Makine öğrenmesinde genel olarak kullanılan dört farklı metot vardır. Bunlar Denetimli Öğrenme (Supervised Learning), Denetimsiz Öğrenme (Unsupervised Learning), Yarı-Denetimli Öğrenme (Semi-Supervised Learning) ve Takviyeli Öğrenme (Reinforcement Learning)'dir. Tez çalışmasında denetimli öğrenmenin sınıflandırma algoritmaları ile veri seti üzerinden tahmin çalışmaları yapılmış ve doğruluk oranları gösterilmiştir.




Şekil 2.1 - Makine Öğrenmesi Metotları

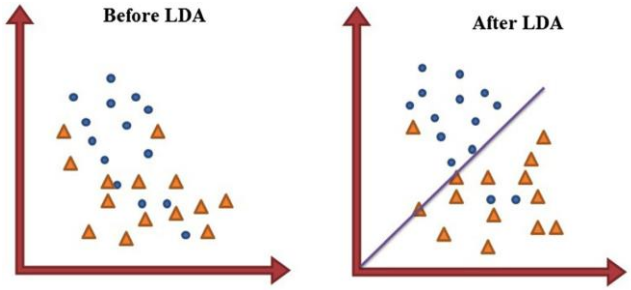
### 2.2.1. Denetimli Öğrenme

Denetimli öğrenme, algoritmaların uygulanması için etiketli eğitim veri setlerini kullanır [1]. Veri setlerindeki özniteliklere ait girdi değerleri ile hedef değişkenine ait öznitelikler arasındaki ilişkiyi keşfetmeye çalışan denetimli öğrenmenin sınıflandırma ve regresyon olmak üzere iki modeli vardır [33]. Sınıflandırma modeli, girdileri önceden belirli olan sınıflardan birine eşleyen bir sınıflandırıcı oluşturur [33]. Tez çalışmasında denetimli öğrenmenin sınıflandırma algoritmalarından lineer diskriminant analizi, k-en yakın komşu algoritması, lojistik regresyon, karar ağacı, destek vektör makineleri, Naive Bayes, toplu öğrenme ve sinir ağları algoritmaları kullanılmıştır.

Bu bölümde tez çalışmasında uygulamada kullanılan algoritmalar hakkında bilgilere yer verilmiştir.

#### 2.2.1.1. Lineer Diskriminant Analizi

Lineer diskriminant analizi veri setindeki öznitelikleri daha düşük boyutlu bir uzaya dönüştüren ve maksimum sınıf ayrılabilirliğini garanti eden bir algoritmadır [34]. Lineer diskriminant analizi aykırı (uç değerler) değerlere karşı duyarlılık gösterir [35]. Lineer diskriminant analizinde sonuç değişkeni bir kategoriye veya bir gruba aittir [36]. Aşağıda şekil 2.2'de görüldüğü gibi lineer diskriminant analizi uygulamasının öncesinde iki sınıf verisi dağınık bir şekilde görülmektedir. Lineer diskriminant analizi ile sınıflar içi mesafe en düşüğe, sınıflar arası mesafe en yükseğe çıkarılarak iki sınıf birbirinden ayrılmaktadır.




**Şekil 2.2** - Lineer Diskriminant Analizi [37]

Dağılım matrislerinin ölçümünde sınıflar arası dağılım ($S_b$), sınıf içi dağılım matrisi ($S_w$) ve toplam dağılım matrisi ($S_t$) dağılımları kullanılır [37].

Burada A bir küme olsun

$$A = [a_1, \ldots, a_n] = [A_1, A_2, \ldots, A_r] \in R^{mxn},$$

$i$ sınıfındaki ($1 \leq i \leq r$) değerler $A_i \in R^{mxn}$ blok matrisi olarak gösterilir ve $N_i$, $i$ sınıfındaki veri öğelerinin dizin kümesidir.

Sınıflar hakkındaki dağılımlar aşağıdaki şekilde ifade edilmektedir.

$$S_b = \sum_{i=1}^{r} n_i (c_i - c)(c_i - c)^{\mathrm{T}}, \tag{2.1}$$

$$S_w = \sum_{i=1}^{r} \sum_{j \in N_i} (a_j - c_i)(a_j - c_i)^{\mathrm{T}}, \tag{2.2}$$

$$S_w = \sum_{j=1}^{r} \sum_{j \in N_i} (a_j - c)(a_j - c_i)^{\mathrm{T}}, \tag{2.3}$$

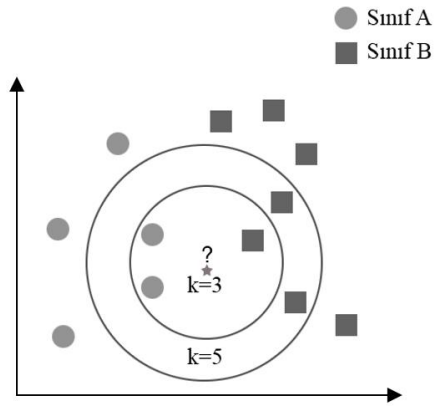
$c_i = (1/n_i) \sum_{j \in N_i} a_j \quad ve \quad c = (1/n) \sum_{j=1} a_j$ sınıfları sırasıyla sınıf merkezleri ve küresel merkezleri gösterir [37].

Veri biliminde öznitelik seçimi ve boyut indirgeme modellerin doğruluğu artırmak için uygulanan önemli tekniklerdir. Bu yüzden lineer diskriminant analizi sıklıkla tercih edilmektedir [38].

### 2.2.1.2. K-En Yakın Komşu Algoritması

K-en yakın komşu (KNN) algoritması denetimli öğrenme metodunun bir sınıflandırma algoritmasıdır [39]. KNN algoritması veri kümesinin tamamını model olarak kullanan ve diğer öğrenme algoritmaların aksine tüm eğitim veri setini bellekte saklayan örnek tabanlı bir algoritmadır [31]. Sınıflandırma aşamasında komşu sayısını belirlemek için bir *k* değeri ve bilinmeyen veri ile bilinen veriler arasındaki mesafeyi hesaplamak için bir mesafe metriği seçilir [40]. Bir veri noktasının sınıfını tanımlamak için kaç adet komşunun dikkate alınacağı *k* 'ye değer verilerek belirlenir [41].

KNN algoritmasının genel olarak eğitim, test ve sınıflandırmadan oluşan üç aşaması vardır [40]. Eğitim aşamasında önce eğitim veri seti öznitelikleri ve hedef değişken kullanılarak model oluşturulur. Sonraki adımda oluşturulmuş bu modelin doğruluğunu test etmek için bir test veri seti ve hedef değişken kullanılır. Daha sonra KNN algoritması yeni verinin etiket tahmininde bulunmak için verilen *k* sayısı kadar yakınındaki komşulara bakarak örnekleri bulur ve çoğunlukta olan etiketi çıkarır [31]. Elde edilen sonucun yetersiz olması durumunda modeli geliştirmek için *k* komşu sayısı ve mesafe metriği değiştirilebilir.




Şekil 2.3 - KNN Algoritması Genel Yapısı

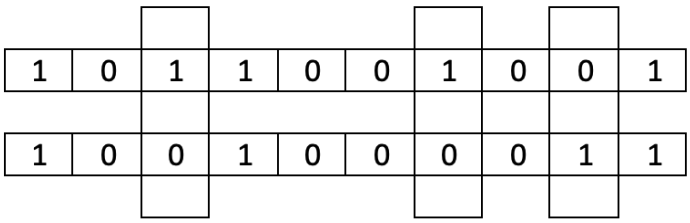
K-en yakın komşu algoritmasında mesafe metrikleri olarak City Block, Chebyshev, Correlation, Cosine, Euclidean, Hamming, Jaccard, Mahalanobis,

Minkowski (Cubic), Spearman metrikleri kullanılmaktadır [42]. Tez çalışmasında her algoritma hiperparametreler ile optimize edilmiş, KNN algoritması kullanılarak ulaşılan en yüksek doğruluk oranında mesafe metriği Hamming seçilmiştir. Hamming mesafesi aynı boyuttaki iki dizinin ne kadar farklı olduğunu ölçmek için kullanılır [43].

Hamming mesafesi formülü aşağıdaki şekilde ifade edilmektedir [44].

$$H = {}^{1}/_{N} \sum_{J=1}^{N} x_j (XOR)\, y_j \tag{2.4}$$

Örnek : Hamming Mesafesi = 3




Şekil 2.4 - Hamming Mesafesi Örneği [25]

**Tablo 2.1** - KNN Algoritmasının Avantajları ve Dezavantajları

| **Avantajları** | **Dezavantajları** |
|---|---|
| 1. Eğitim çok hızlıdır.<br>2. Öğrenmesi kolaydır.<br>3. Gürültülü eğitim verilerine karşı dirençlidir.<br>4. Eğitim verisi büyükse etkilidir. | 1. Hesaplama karmaşıklığı<br>2. Bellek sınırlamasının olması<br>3. Algoritma olarak yavaş çalışması |

KNN algoritmasının kullanım alanları içinde görüntü tanıma, metin sınıflandırma, veri madenciliği gibi alanlar yer almaktadır [40].

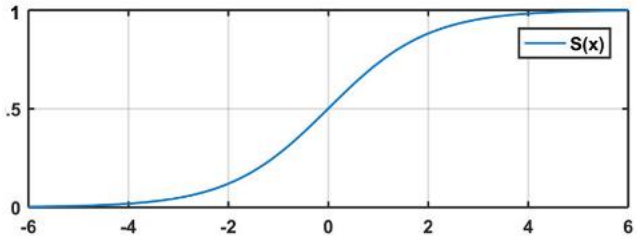
### 2.2.1.3. Lojistik Regresyon

Lojistik regresyon, istatistikte iki sınıflı değişkenlerin sınıflandırılması için kullanılır [45]. Lojistik regresyon birden fazla bağımsız değişken ile kategorik bir bağımlı değişken arasındaki ilişkiyi bir eğri ile göstererek gerçekleşecek olan durumun olasılığını tahmin eder [36]. Bağımsız değişkenler kategorik ve sürekli değer olabilir ancak bağımlı değişken kategorik olmak zorundadır [46]. Lojistik regresyon olayların gerçekleşme zamanlarını değil, bir olayın karar verme durumunda yani başarılı-başarısız, evet-hayır, hasta-sağlıklı gibi ikili sonuçlara odaklandığı modeller için uygundur ve lojistik regresyon tarafından bu kategorik değerlere "Evet" ise 1 ve "Hayır" ise 0 ifadesi olacak şekilde bir katsayı uygulanır [46]. Lojistik regresyon bir veya birden fazla değişkene bağlı ikili (binary) yanıt olasılıklarını bulmak için genel olarak sigmoid fonksiyonunu kullanır ve bu fonksiyon ile en uygun parametreleri bulur [47].

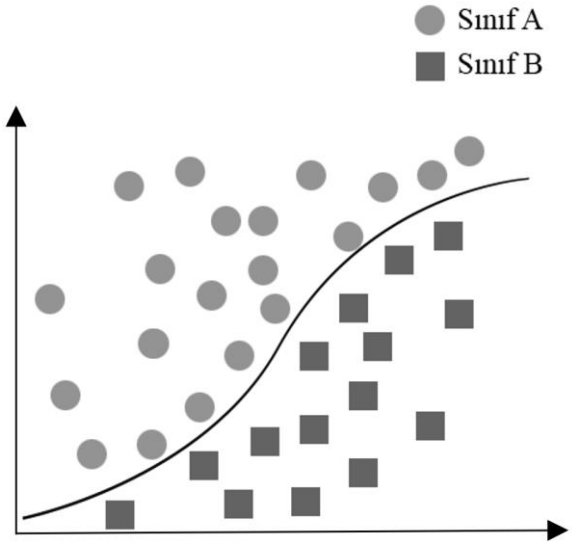
Sigmoid işlevi (σ) ve sigmoid fonksiyon girdisi (x) aşağıda şekil 2.5'te gösterilmektedir [47].

$$\sigma(x) = \frac{1}{(1 + l^{-x})} \tag{2.5}$$

x ∈ R and S(x) ∈ (0, 1)




Şekil 2.5 - Sigmoid Fonksiyonu [29]




Şekil 2.6 - LR Genel Gösterimi

Fizik, ekonomi, siyaset bilimleri gibi alanlarda da kullanılmasına rağmen lojistik regresyon tıp alanında yoğun bir şekilde kullanılmaktadır [46]. Lojistik regresyonun dezavantajlarından birisi büyük ölçekli uygulamalarda yavaş performans göstermesidir [45].

#### 2.2.1.4. Karar Ağacı

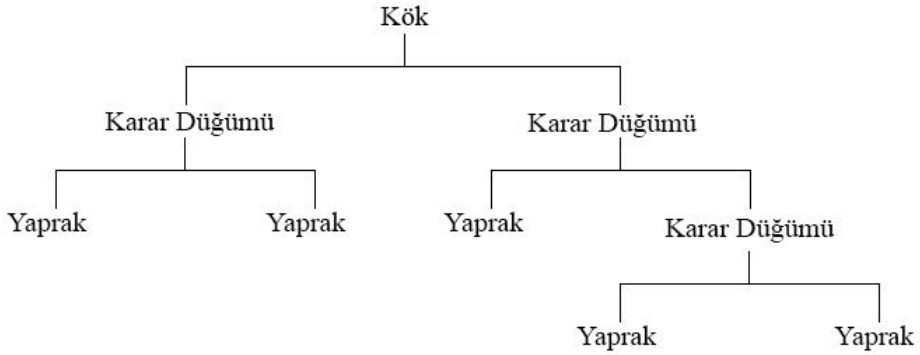
Karar ağacı algoritmaları kategorik ve sayısal bilgileri işleyebilen denetimli öğrenme algoritmalarındandır [2]. Karar ağacındaki her düğüm öznitelik değerine göre iki veya daha fazla alt bölüme ayrılır [33]. Her dal düğümdeki testin bir sonucunu ve her yaprak düğümü sınıf etiketini temsil eder [48]. Karar ağacının amacı en düşük genelleme hatası ile optimal karar ağacını bulmaktır [49].

Karar ağacı şekil 2.7'de görüldüğü gibi, seçimleri ve sonuçları bir ağaç şeklinde gösteren grafiksel bir yapıya sahiptir. Grafikteki düğümler bir seçimi ve grafiğin kenarları da koşulları gösterir [50]. Karar ağacı modeli genel olarak kök, iç düğüm (ya da karar düğümü) ve yaprak bölümlerinden oluşur [2].

**Kök düğüm :** Başlangıç düğümü veri setindeki tüm örnekleri içeren düğümdür [31].

**İç Düğüm(Karar Düğümü) :** Kısıtlamaları ve koşulları kontrol etmek için kullanılır [2].

**Yaprak düğüm :** Sınıflandırma sonucunun saklandığı düğümdür [2].




**Şekil 2.7** - Karar Ağacı Algoritması Genel Yapısı

Karar Ağacı sınıflandırıcısının altında birçok algoritma vardır. Bu algoritmaların en sık kullanılanları ID3, C4.5 ve CART algoritmalarıdır [30].

**ID3 ( Iterative Dichotomiser 3) :** ID3 algoritması sadece kategorik veriler ile kullanılır ve gürültülü verilere karşı dirençsizdir [51].

**C4.5 :** ID3 algoritmasının iyileştirilmiş halidir ve sürekli (sayısal) ve kategorik veriler ile kullanılabilmektedir [51]. C4.5 algoritmasında ID3 algoritmasına ek olarak aşağıdaki özellikler yer alır.

- Aşırı uyum problemini çözmek için aşağıdan yukarı doğru "budama" olarak bilinen bir teknik uygulanır [31].
- Sürekli(sayısal) ve kategorik öznitelikler ile çalışabilir [31].
- Eksik öznitelik değerlerinin "?" olarak işaretlenmesine izin verir [51].

**CART (Classification and Regression Trees) :** İkili ağaçlar geliştiren CART algoritması veri setindeki aykırı değerlerden etkilenmemekte ve hem kategorik hem sürekli (sayısal) veriler ile kullanılabilmektedir.

**Tablo 2.2** - Karar Ağacı Algoritmaları

| Karar Ağacı Algoritması | Veri Tipleri | Veri Bölme Methodu | Sınıflandırma Metriği |
|---|---|---|---|
| ID3 | Kategorik | Kısıtlama yok | Entropi ve Bilgi Kazancı (Information Gain) |
| C4.5 | Kategorik ve Sayısal | Kısıtlama yok | Kazanç Değeri (Gain Ratio) |

| | | | |
|---|---|---|---|
| CART | Kategorik ve Sayısal | İkili (Binary) Bölme | Gini Index |

### 2.2.1.5. Destek Vektör Makineleri

Destek vektör makinelerinin arkasında sınıfların aralarındaki kenar boşluğunu maksimize eden bir yüzeyle ayırması ana fikri vardır [52]. Destek vektör makinelerinde girdiler doğrusal olmayan bir şekilde çok yüksek boyutlu öznitelik uzayına eşleştirilir ve burada doğrusal bir karar yüzeyi oluşturulur [53]. Karar yüzeyindeki sınıfları birbirinden ayıran doğruya karar sınırı denir [31]. Karar sınırının belirlenmesine yardımcı olan ve iki sınıfın birbirine en yakın noktaları üzerinden sınırlar geçmektedir. Bu sınırlar karar sınırına eşit mesafededir ve üzerindeki vektörlere de destek vektörleri denir. Destek vektör makineleri doğrusal olarak ayrılan ve doğrusal olarak ayrılamayan sınıflar için kullanılmaktadır. Doğrusal olarak ayrılan sınıflar keskin kenar payı (hard margin) ve esnek kenar payı (soft margin) yöntemleri ile ve doğrusal olarak ayrılamayan sınıflar ise çekirdekler yardımı ile sınıflandırılmaktadır.

#### 2.2.1.5.1. Keskin Kenar Payı (Hard Margin)

Bir sınıfa ait vektörlerin diğer sınıfa karışmayacak şekilde sınırlar ile ayrılması durumu keskin kenar payı (hard margin) veya maksimum marj sınıflandırıcısı (maximal margin classifier) olarak adlandırılır. Bu teknikte sınıfların birbirine en yakın vektörleri üzerinden bir sınır geçmektedir ve vektörler diğer sınıfın alanına girmemektedir. Destek vektörleri arasındaki kenar boşluğunun en uzak olması amaç edinilmektedir. Şekil 2.8'de görüldüğü gibi karar sınırını destek vektörlerinin uzaklıkları belirlemektedir. Kenar sınırı iki sınıfa eşit mesafededir.

Sınıflandırma problemlerindeki genel problem elimizdeki bilinmeyen verinin hangi sınıfa ait olduğunu bulmak onu bir sınıfa atamaktır ve bu problem aşağıdaki şekilde ifade edilmektedir [54].

$$(x_1,y_1),\ldots, (x_m\ y_m) \in X \times \{+1, -1\}$$

X : boş olmayan bir küme,

$x_i$ : Girdi (örnek veya gözlem de denilmektedir.)

$y_i$ : Hedef (etiket veya çıktı denilmektedir.)

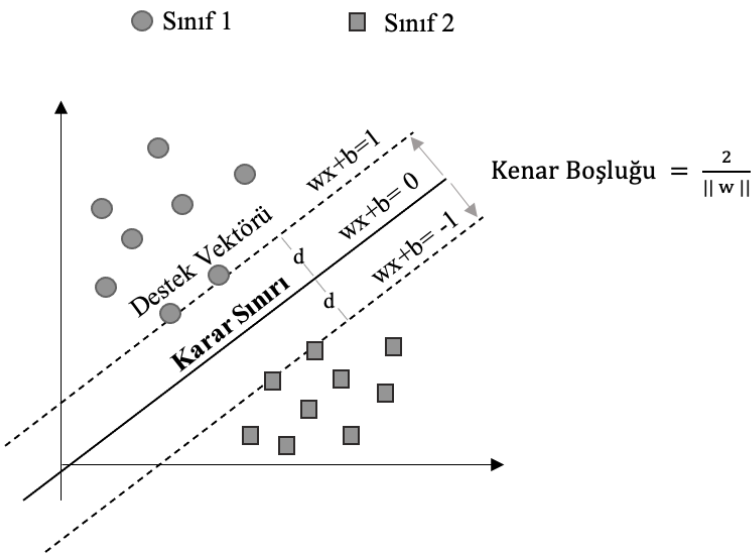
Destek vektör makineleri algoritmalarında etiketlerin sayısal ifadelere, pozitif (+1) ve negatif (-1) değerlere sahip olması gerekir [53]. Bu durum, ikili örüntü tanıma (binary pattern recognition) veya ikili sınıflandırma (binary classification) olarak adlandırılır.

$y_i = \{+1,-1\}$ tahmin kullanacağımız etiketlerin karşılıkları olarak çıktılarımızdır.

Aşağıdaki düzlemde belirtilen Sınıf 1 ($y_i = +1$) ve Sınıf 2 ($y_i = -1$) olarak denklemi;

$$wx + b \geq 1 \quad , \quad y_i = +1 \tag{2.6}$$

$$wx + b \leq 1 \quad , \quad y_i = +1 \tag{2.7}$$




Şekil 2.8 - Destek Vektör Makineleri

Yukarıda şekil 2.8'de hiper düzlemin (kenar sınırı) denklemi wx+b=0 olarak verilir.

w : Ağırlık, eğim.

x : Girdi

d : Mesafe

b : Sapma oranıdır. Bias veya kesme olarak da adlandırılmaktadır.

### 2.2.1.5.2. Esnek Kenar Payı (Soft Margin)

Verinin gürültülü olması yani bir sınıfın bazı vektörlerinin diğer sınıf alanında olması durumuna esnek kenar payı (soft margin) denir.

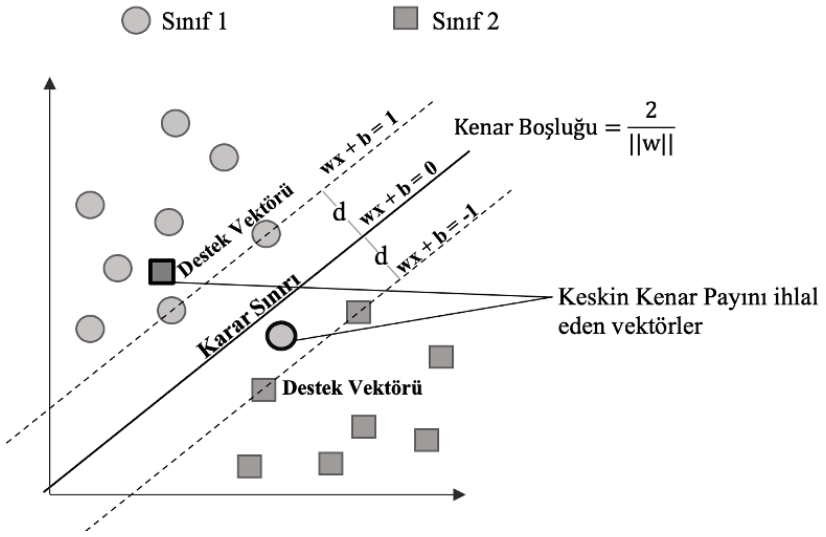
$(x_i,y_i)$ ve $y_i= 1$ için

$$y_i(wx+b) \geq 1 - \varepsilon , \quad \varepsilon = 0 \tag{2.8}$$

$$y_i(wx+b) \geq 1 - \varepsilon , \quad 0 < \varepsilon <1 \tag{2.9}$$

$$y_i(wx+b) \geq 1 - \varepsilon , \quad \varepsilon \geq 1 \tag{2.10}$$

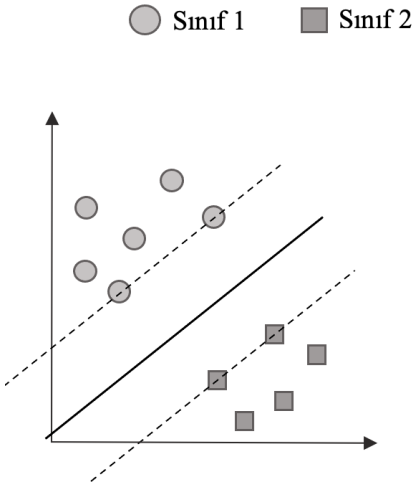
Burada ε ile esneklik payı ekleniyor. Bu doğruların düzlem üzerinde hareket payını göstermektedir.



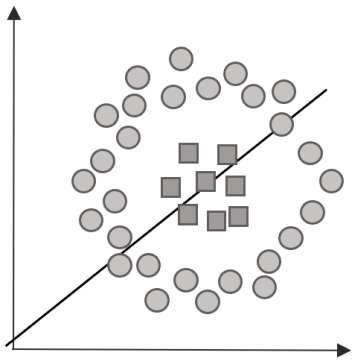

Şekil 2.9 - DVM Esnek Kenar Payı

### 2.2.1.5.3. Çekirdekler

Gürültülü veriyi doğrusal olarak ayırmak her zaman mümkün olmamaktadır [55]. Örneğin aşağıda şekil 2.10'da sınıflar birbirinden kesin olarak ayrılabilmektedir. Ancak şekil 2.11'de bu durum söz konusu değildir. Bu yüzden sınıfların doğrusal olarak ayrılamadığı durumlarda çekirdekler yardımı ile sınıflar birbirinden ayrılabilmektedir.

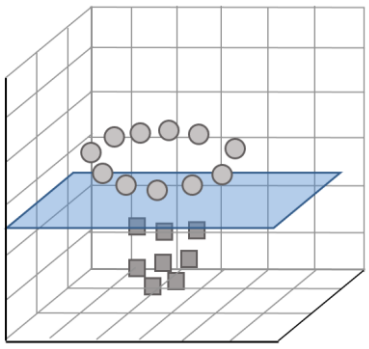



**Şekil 2.10** - DVM Sınıfların Doğrusal Olarak Ayrılması



**Şekil 2.11** - DVM Doğrusal Olarak Ayrılamayan Sınıflar

Doğrusal olarak ayrılamayan sınıflar destek vektör makinelerinde aşağıda gösterildiği gibi çekirdekler yardımı ile düzlem oluşturularak ayrılır.



**Şekil 2.12** - DVM Çekirdekler Yardımı ile Sınıfların Ayrılması

Destek vektör makineleri yüz tanıma [56], kurumsal kredi notlarının tahmini [57], zararlı yazılımların tahmini [58], Elektroensefalografi (EEG) [59] gibi bir çok konuyu içeren mühendislik, tıp, finans, işletme gibi bir çok alanda kullanılmaktadır.

### 2.2.1.6. Naive Bayes Sınıflandırıcısı

Naive Bayes sınıflandırıcısı, gerçekleştiği bilinen bir olaya dayalı olarak yeni bir olayın meydana gelme olasılığını veren [60] karmaşık verileri işleyebilen, doğrusal olmayan [8], Bayes teoremine dayanan olasılıksal [61] bir sınıflandırma algoritmasıdır [8]. Bu algoritmada veri kümesindeki bir sınıfa ait özelliğin diğer özelliklerden bağımsız olduğu varsayılır [11]. Bu durum algoritmanın düşük performans göstermesine sebep olmaktadır [62]. Bunun yanında diğer algoritmaların karmaşıklığına ve yavaşlığına göre Naive Bayes algoritması hızlı ve kolay uygulanabilen bir algoritmadır [62]. Naive Bayes algoritması genellikle diğer sınıflandırıcılardan daha az parametre tahmin ettiği için aşırı öğrenmeye karşı daha fazla dirençlidir [63].

#### 2.2.1.6.1. Naive Bayes Sınıflandırıcısının Formülü

$A_1$, $A_2$, $A_3$, ..., $A_n$ aynı veri kümesi içindeki bir dizi ve B gerçekleştiği bilinen bir olay ise, bu olayın $A_j$ olayı nedeniyle meydana gelme olasılığı veren formül aşağıdaki şekilde ifade edilmektedir.

$$P(A_j/B) = \frac{P(B \cap A_j)}{P(B)} = \frac{P(B \cap A_j)}{P(A_1).P\left(\frac{B}{A_1}\right)+P(A_2).P\left(\frac{B}{A_2}\right)+\cdots+P(A_n).P\left(\frac{B}{A_n}\right)} = \frac{P(B \cap Aj)}{\sum_{j=1}^{n} P(A_j).P\left(\frac{B}{A_j}\right)} \quad (2.11)$$

Naive Bayes algoritması sağlık alanında kalp hastalıkların teşhisi [64], medya alanında sahte haberlerin tespiti [65], metin sınıflandırma [66], görüntü tanıma [67], istenmeyen postaların filtrelenmesi [68] gibi çeşitli uygulamalarda sıklıkla kullanılmaktadır.

### 2.2.1.7. Toplu Öğrenme Algoritmaları

Toplu öğrenme algoritmaları en başta veriyi birden fazla bağımsız öğreniciye (alt modellere) bölerek ve daha sonra birleştirerek modelin sonuç performansının iyileştirilmesini sağlar [69]. Çok sayıda uygulama performansı arttırmak, aşırı uyumu

engellemek ve hataları azaltmak için toplu öğrenme algoritmalarını kullanmaktadır [70]. Toplu öğrenme yöntemi içinde toplu üretmek için sık kullanılan iki algoritma yükseltme (boosting) ve torbalama (bagging) algoritmalarıdır [71]. Bu tekniklere aşağıda kısaca değinilmiştir. Tez çalışmasında toplu öğrenme yöntemleri içinde en iyi performansı yükseltme (boosting) metotlarından olan LogitBoost göstermiştir.

**2.2.1.7.1. Torbalama (Bagging):** Torbalama, eğitim setini rastgele öğrenicilere ayırır [69] ve ağırlıklı ortalama gibi bazı ortalama yöntemleri kullanarak sonunda bu tahminleri birleştirilir [72].

**2.2.1.7.2. Yükseltme (Boosting):** Bu algoritma zayıf sınıflandırıcıları hiyerarşik olarak eğiterek bir problemdeki zor kısımları öğrenmelerini sağlar ve böylece daha güçlü bir öğrenici meydana gelir [73].

**2.2.1.7.3. LogitBoost:** Bu algoritma, binom log olasığını kullanarak modelin kayıp fonksiyonunu lineer olarak değiştirir ve bu nedenle veri setindeki aykırı değerlere ve gürültülü veriye karşı az duyarlılık gösterir [74]. LogitBoost aşırı uyuma karşı oldukça dayanıklı bir algoritma olmasına rağmen gereğinden fazla yükseltme (boosting) yapıldığında tahminlerde aşırı uyum gerçekleşebilir [75].

LogitBoost algoritması aşağıdaki adımlardan oluşur [74]. N sayıda örnekli eğitim veri seti iki sınıfa bölünmüş olsun. Buradaki iki y ∈ {-1, +1} olarak tanımlanır.

y = 1 ifadesi olumlu (evet / var)
y = -1 ifadesi olumsuz (hayır / yok) olarak tanımlanmıştır.

Eğitim verisi kümesi $\{(x_1, y_1), ..., (x_i,y_i),\dots (x_n,y_n)\}$ olsun,

1. Girdi veri seti $N = \{(x_1, y_1), \dots, (x_i, y_i), \dots, (x_n, y_n)\}$ olmak üzere

   burada $x_i$ özellik vektörü, $y_n$ ise hedef sınıf

   $x_i \in X$ ve $y_i \in Y = \{-1,+1\}$

   K , Girdi yenileme sayısı.

2. Ağırlıklar $w_i = 1/N$, i=1,2,...,N;

   $F(x)=0$ ve olasılık tahmini $P(x_i)=1/2$

3. Yineleme k=1,2,...,K:

a. Ağırlıkların ve çalışma yanıtının hesaplanması

$$w_i = p(x_i)(1 - p(x_i)) \tag{2.12}$$

$$z_i = \frac{y_i - p(x_i)}{p(x_i)(1 - p(x_i))} \tag{2.13}$$

b. $z_i$'nin en küçük kareler regresyonu ile ağırlıklandırılmış fk(x) fonksiyonu $x_i$ ye uygulanır.

c.

$$F(x) \leftarrow F(x) + \frac{1}{2} fk(x) \tag{2.14}$$

ve buradan

$$p(x) \leftarrow \frac{e^{F(x)}}{e^{F(x)} + e^{-F(x)}} \tag{2.15}$$

bulunur.

d. Sınıflandırıcı Çıktısı

$$sign[F(x)] = sign\left[\sum_{k=1}^{K} fk(x)\right] \tag{2.16}$$

bu durumda $sign[F(x)]$ fonksiyonu iki olasılığa sahiptir.

$$sign[F(x)] = \text{sign\ [k=1Kfk(x)]} sign[F(x)] = \begin{cases} 1, if\ F(x) < 0 \\ -1, if\ F(x) \geq 0 \end{cases} \tag{2.17}$$

Temel toplu öğrenmesi formülü aşağıdaki gibi ifade edilmektedir. [76].

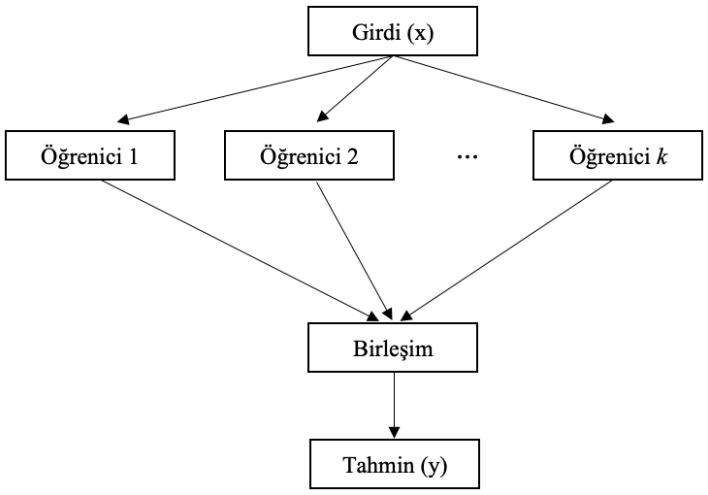
$\varphi(x_i) = G(f_1(x_i), f_2(x_i), \ldots, f_k(x_i)) = y_i$, (2.18)

x = Girdi

y = Tahmin

G = Toplam fonksiyonu

$f_1, f_2, \ldots, f_k$ = Temel öğreniciler.

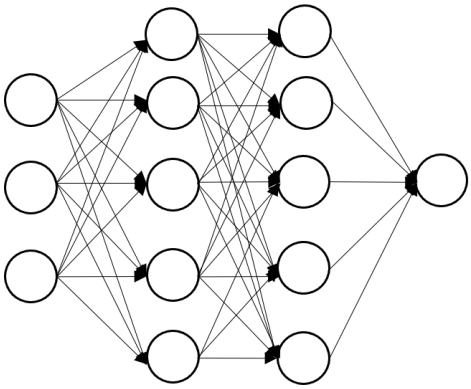



**Şekil 2.13** - Toplu Öğrenme Algoritmaları Genel Yapısı

Toplu öğrenme algoritmaları duygu analizi [76], metin sınıflandırma [70], kredi notu değerlendirmesi [69], KOAH ve Akciğer Kanseri tespiti [77], rüzgar enerjisi üretim tahmini [78] gibi bir çok farklı konuda sıklıkla kullanılmaktadır.

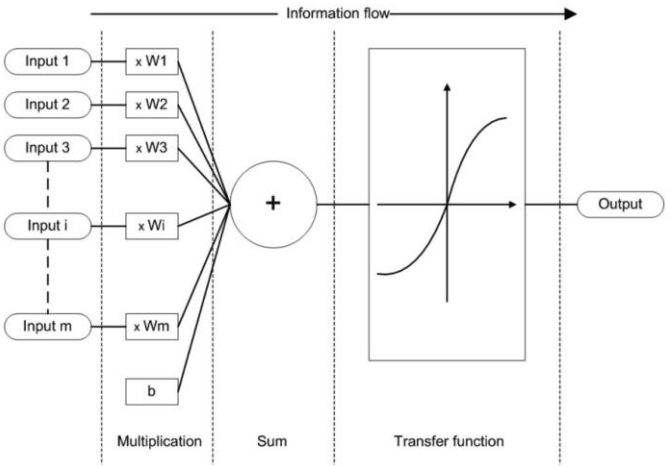
### 2.2.1.8. Sinir Ağları

Sinir ağları (ya da Yapay Sinir Ağı, YSA), insan beyninin bilgiyi işleme yeteneğini örnek alan, örneklerden öğrenen ve problemleri çözmek için birbirine bağlı çok sayıda bağlantı noktasından oluşan matematiksel modellerdir [79]. Yapay sinir ağları sadece algoritma değil , bir donanım parçası da olabilir [80]. İnsan beyni gibi paralel olarak işlem yapabilen, eksik, gürültülü ve verinin net olmadığı durumlarda çalışabilen yapay sinir ağları denetimli ve denetimsiz öğrenme yöntemlerinde kullanılmaktadır [81]. Sinir ağlarında bir çok öğrenme kuralı olmasına rağmen tahmin ve sınıflandırma problemlerinde yüksek performans gösteren denetimli öğrenme kurallarından olan geri yayılım (Back Propagation) algoritması en sık kullanılan öğrenme yöntemidir [82]. Geri yayılım yönteminde algoritma verilen örneklerden öğrenir, değişkenler arasındaki matematiksel ilişkilere yer verilmez [83].



Şekil 2.14 - Sinir Ağları Genel Yapısı

Yapay sinir ağları yapay nöronlardan oluşur ve şekil 2.15'te görüldüğü gibi girdi değerleri (input) ayrı ayrı ağırlık değerleri (W) ile çarpılır (multiplication), daha sonra çarpılan bu değerler ve gerçek değer ile tahmin değeri arasında ki uzaklığı gösteren yanlılık (bias) değerleri toplanır (sum) ve toplama işleminden sonra elde edilen değerler etkinleştirme (activation ya da transfer function) fonksiyonundan geçerek bir çıktı oluşturur [84].




**Şekil 2.15** - Yapay Nöronların Çalışma Prensibi [84]

Tahmin doğruluğu katman sayısı ve aktivasyon fonksiyonlarına bağlı olan sinir ağlarında aktivasyon fonksiyonları girdiler ve çıktılar arasındaki lineer olmayan ve karışık eşleşmelerin öğrenilmesinde oldukça önemlidir [80]. Binary Step Function, Linear , Sigmoid, Tanh, ReLU, Leaky ReLU, Parametrized ReLU, Exponential Linear Unit, Swish, SoftMax gibi bir çok aktivasyon fonksiyonu içinde sigmoid fonksiyonu sinir

ağlarında en sık kullanılan, değerleri 0 ile 1 aralığına dönüştüren aktivasyon fonksiyonudur [80].

Yapay sinir ağları ilaç araştırmalarında [82], biyomedikal uygulamalarda [84], bilgisayar ağlarının güvenliğinde [85], hava yolları sektöründe [86] ve finans, yönetim, tıp, üretim, pazarlama, ulaşım gibi bir çok alanda kullanılmaktadır [87].

### 2.2.2. Denetimsiz Öğrenme

Denetimsiz öğrenme etiketsiz veri setleri ile çalışır. Öncelikle veriler öznitelik gruplarına ayrılır daha sonra verilerin sınıflandırma işlemi yapılır [29]. Denetimsiz öğrenme etiketlenmemiş yani sınıflandırılmamış, bir kategoriye ait olmayan veriler arasındaki ilişkileri öğrenerek algoritmalar yardımı ile gruplara ayırır [88].

### 2.2.3. Yarı Denetimli Öğrenme

Denetimli öğrenme ile aynı amacı olan yarı denetimli öğrenme etiketli veri setlerini ve genellikle sayıca daha çok olan etiketsiz veri setlerini kullanarak daha iyi bir model geliştirmeyi amaçlar [31].

### 2.2.4. Takviyeli Öğrenme

Takviyeli öğrenme, bir sistemin kendi etkileşimlerinin sonuçlarını gözlemleyip önceki deneyimlerinden öğrenmesini sağlar [79]. Bu algoritma denetimli öğrenme gibi örnek veri seti ile değil, deneme yanılma yöntemi ile öğrenen, verinin analiz edilmesi sonucunda aldığı bildirimler ile kullanıcıları en doğru sonuca yönlendiren davranışsal bir öğrenme yöntemidir [29]. Öğrenici, aldığı geri bildirimler ile yöntemini ödüllendirme (iyi) ve cezalandırma (kötü) olarak derecelendirerek bir denge durumu oluşana kadar parametrelerini ayarlar [89]. Yapay zeka takviyeli öğrenme (ya da pekiştirmeli öğrenme) ile derin öğrenmenin birleşiminden meydana gelmektedir [79].

# BÖLÜM 3. YÖNTEM

Tez çalışmasında MathWorks firmasına ait MATLAB programı ile Makine öğrenmesi yöntemlerinden denetimli öğrenmenin sınıflandırma algoritmaları kullanılmıştır. Denetimli öğrenme etiketli veri setlerini kullanır ve genel olarak sınıflandırma ve regresyon problemlerinde kullanılır. Uygulamada kullanılacak veri seti 216 kayıt ve 13 öznitelik içeren eğitim veri seti ve 54 kayıt ve 13 öznitelik içeren test veri seti olarak ikiye ayrılmıştır. Eğitim veri seti ile MATLAB programında çapraz doğrulama (Cross-Validation) özelliği seçilerek lineer diskriminant analizi, k-en yakın komşu, lojistik regresyon, karar ağacı, destek vektör makineleri, Naive Bayes, toplu öğrenme ve sinir ağları algoritmaları kullanılarak model oluşturulmuş ve daha sonra test veri seti kullanılarak algoritmaların doğruluk oranları bulunmuştur.

## 3.1. Veriler ve Toplanması

Tez çalışmasında UCI ( University of California, Irvine) Machine Learning Repository web sitesinden "Statlog (Heart) Data Set" [90] isimli veri seti kullanılmıştır. Veri seti toplamda 270 kayıt ve 13 özniteliğe sahiptir. Veri setinde kullanılan öznitelikler ve açıklamaları Tablo 3.1'de verilmiştir.

**Tablo 3.1** - Öznitelik Tablosu

| **Sıra No** | **Öznitelik** | **Değer** |
|---|---|---|
| 1 | Yaş (Age) | Gerçek Sayı |
| 2 | Cinsiyet (Sex) | İkili Sayı |
| 3 | Göğüs Ağrısı Tipi ( Chest Pain Type (4 Values) ) | Nominal Sayı |
| 4 | Dinlenme Kan Basıncı (Resting Blood Pressure) | Gerçek Sayı |
| 5 | Serum Kolesterolü - Mg/DI (Serum Cholesterol In Mg/Dl) | Gerçek Sayı |
| 6 | Açlık Kan Şekeri > 120 Mg/Dl (Fasting Blood Sugar > 120 Mg/Dl ) | İkili Sayı |
| 7 | Dinlenme Elektrokardiyografik Sonuçları (Değerler 0,1,2) (Resting Electrocardiographic Results (Values 0,1,2) ) | Nominal Sayı |
| 8 | Ölçülen Maksimum Nabız (Maximum Heart Rate Achieved) | Gerçek Sayı |
| 9 | Egzersize Bağlı Angina (Exercise Induced Angina) | İkili Sayı |

| | | |
|---|---|---|
| 10 | Oldpeak = Dinlenmeye Göre Egzersizin Neden Olduğu ST Depresyonu (ST Depression Induced By Exercise Relative To Rest) | Gerçek Sayı |
| 11 | Zirve Egzersiz ST Segmentinin Eğimi (The Slope Of The Peak Exercise ST Segment) | Sıralı Sayı |
| 12 | Flourosopy ile Renklendirilen Damar Sayısı (0-3) (Number Of Major Vessels (0-3) Colored By Flourosopy) | Gerçek Sayı |
| 13 | Thal: 3 = Normal; 6 = Sabit Hata (Fixed Defect); 7 = Tersine Çevrilebilir Hata (Reversable Defect) | Nominal Sayı |

Tahmin Değişkeni
0 : Kalp Hastalığı Yok , 1 : Kalp Hastalığı Var

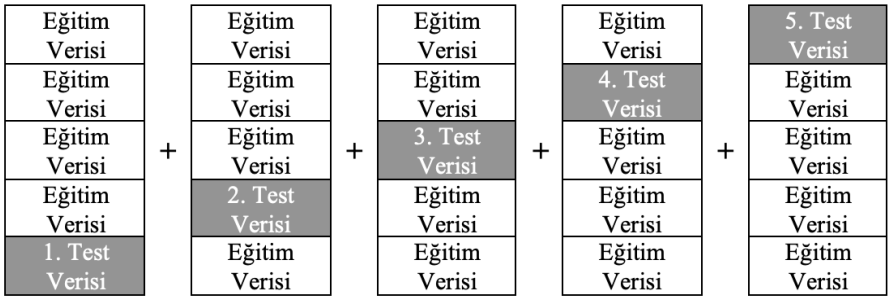
## 3.2. Makine Öğrenmesinde Sınıflandırma Performans Ölçütleri

Eğitim veri seti ile oluşturulan modellerin performansını göstermek için test veri setleri kullanılmaktadır. Ancak bazen algoritmalar oluşturulan modeldeki veriyi ezberleyerek yüksek doğruluk oranı göstermektedir. Bu duruma aşırı öğrenme ya da aşırı uyum denilmektedir. Bu durumda test veri seti doğruluk oranı, eğitim veri setinin doğruluk oranına göre düşük çıkmaktadır. Aşırı öğrenmeyi (ya da aşırı uyum) engellemek için bu tez çalışmasında çapraz doğrulama (Cross-Validation veya k-fold Cross Validation) tekniğinden yararlanılmıştır. Ayrıca kullanılan algoritmaların performansları farklı performans ölçütleri ile gösterilmiştir. Bu konular ile ilgili açıklama ve formüllere aşağıda yer verilmiştir.

## 3.3. Çapraz Doğrulama

Çapraz doğrulama (Cross-Validation veya k-fold Cross-Validation ) ile veri seti algoritmaların hesaplamalarına uyması için alt bölümlere ayrılır [91]. Bölünen alt kümeler eğitim ve doğrulama (test) olarak iki alt kümeye ayrıldıktan sonra, eğitim kısmı ile eğitilen model, doğrulama alt kümesi ile test edilir [91]. Veri seti belirlenen *k* değeri kadar parçaya bölünür ve her bölüm yine *k* değeri kadar çalıştırılır. Bunun sonunda test kısımlarının doğruluk oranlarının toplamı *k* değerine bölünür ve sonuca ulaşılır.

Örnek: Çapraz Doğrulama (kfold) değeri = 5 ise



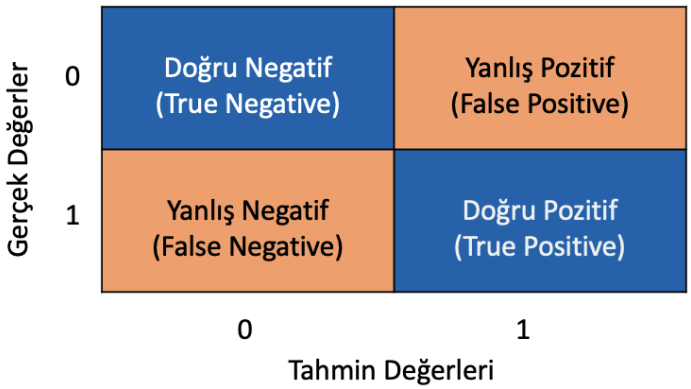

Şekil 3.16 - Çapraz Doğrulama Örneği

D.O = Doğruluk Oranı
V.S =Veri Seti

$$\text{Değerlendirme} = \frac{1.V.S.D.O \; + \; 2.V.S.D.O. \; + 3.V.S.D.O. \; + 4.V.S.D.O. + 5.V.S.D.O.}{5} \qquad (3.1)$$

## 3.4. Performans Ölçütleri

Sınıflandırma problemlerinde performans ölçütlerinin hesaplamalarında kullanılan gerçek ve tahmin değerleri hata (karışıklık) matrisi göz önünde bulundurularak hesaplanır. Hata matrisindeki alanların anlamları ve tablosu aşağıda verilmiştir.




Şekil 3.17 - Hata Matrisi Genel Yapısı

**DN (Doğru Negatif) :** Gerçek değerlerde Negatif olarak sınıflandırılan (0) değerlere Negatif sınıf tahmini yapılarak sonucun doğru tahmin edildiğini gösterir.

**DP (Doğru Pozitif) :** Gerçek değerlerde Pozitif olarak sınıflandırılan (1) değerlere Pozitif sınıf tahmini yapılarak doğru tahmin edildiğini gösterir.

**YP (Yanlış Pozitif) :** Gerçek değerlerde Negatif olarak sınıflandırılan (0) değerlere Pozitif sınıf tahmini yapılarak yanlış tahmin edildiğini gösterir.

**YN (Yanlış Negatif) :** Gerçek değerlerde Pozitif olarak sınıflandırılan (1) değerlere Negatif sınıf tahmini yapılarak yanlış tahmin edildiğini gösterir.

**3.4.1. Doğruluk (Accuracy) :** Veri setinde tahmin ettiğimiz değerlerin doğruluk oranını gösterir. Aşağıdaki şekilde hesaplanır:

$$\text{Doğruluk} = \frac{\text{DP} + \text{DN}}{\text{DP} + \text{DN} + \text{YP} + \text{YN}} \tag{3.2}$$

**3.4.2. Kesinlik (Presicion) :** Tahmin alanında sınıflandırılan pozitif örnekler arasında ne kadarının doğru sınıflandırıldığını gösterir. Aşağıdaki şekilde hesaplanır:

$$\text{Kesinlik} = \frac{\text{DP}}{\text{DP} + \text{YP}} \tag{3.3}$$

**3.4.3. Duyarlılık (Recall) :** Doğru olarak tahmin edilen pozitif değerlerin gerçek değerlerdeki tüm pozitif sınıfa (doğru pozitif ve yanlış negatif) olan oranını gösterir. Duyarlılık oranının düşük çıkması yanlışların fazla olduğunu doğruların az olduğunu, duyarlılık oranının yüksek çıkması doğruların fazla yanlışların az olduğunu gösterir.

$$\text{Duyarlılık} = \frac{\text{DP}}{\text{DP} + \text{YN}} \tag{3.4}$$

**3.4.4. F1 Skor (F1 Score) :** F1 Skor kesinlik ve duyarlılığın harmonik ortalamasını verir. Aşağıdaki şekilde hesaplanır:

$$\text{Doğruluk} = 2 * \frac{\text{Kesinlik x Duyarlılık}}{\text{Kesinlik} + \text{Duyarlılık}} \tag{3.5}$$

**3.4.5. ROC Eğrisi** : ROC eğrisi kesinlik ve duyarlılıkta olduğu gibi dengeli olmayan veri setlerinde performans değerlendirmesi için önemlidir. ROC eğrisi modelin

performansını değerlendirmek için kullanılır [31]. ROC eğrisinin altındaki alanın büyüklüğü doğruluk oranının yüksekliğini gösterir [31]. ROC eğrisi y ekseninde 1 değerine yaklaştıkça ve x ekseninde 0 değerine yaklaştıkça doğruluk oranı artmaktadır. ROC eğrisinde x eksenini yanlış pozitif oran (YPO) ve y eksenini de gerçek pozitif oran oluşturmaktadır.

$$\text{Özgüllük} = \frac{\text{DN}}{\text{DN} + \text{YP}} \tag{3.6}$$

$$\text{Gerçek Pozitif Oran} = \frac{\text{DP}}{\text{DP} + \text{YN}} \tag{3.7}$$

$$\text{Yanlış Pozitif Oran} = 1 - \text{Özgüllük} = \frac{\text{DP}}{\text{DP} + \text{YN}} \tag{3.8}$$

## 3.5. Verilerin Çözümlenmesi ve Yorumlanması

Kullanılan veri seti csv (comma-separated values) dosya formatında etiketli veri setidir. Eğitim veri seti ve test veri setlerini doğrulamada denetimli öğrenmenin sınıflandırma algoritmalarından

- Lineer Diskriminant Analizi (Linear Discriminant Analysis)
- K-En Yakın Komşu Algoritması (K-Nearest Neighbor Algorithm)
- Lojistik Regresyon (Logistics Regression)
- Karar Ağacı (Decision Tree)
- Destek Vektör Makineleri (Support Vector Machines)
- Naive Bayes Teoremi (Naive Bayes Theorem)
- Toplu Öğrenme Algoritmaları (Ensemble Learning Algorithms)
- Sinir Ağları (Neural Networks)

algoritmaları kullanılıp, sonuçlar karşılaştırılmıştır. Uygulamada MathWorks firması tarafından geliştirilen MATLAB programı kullanılmıştır.

# BÖLÜM 4. BULGULAR VE YORUMLAR

Bu bölümde, önce araştırmanın amaçlarını yansıtan başlıklar halinde bulgulara ve sonrasında bulguların bütün halinde anlamlandırılmaya çalışıldığı yorumlara yer verilmiştir.

## 4.1.Bulgular

Bu bölümde, eğitim veri seti ile oluşturulan modele seçilen algoritmaların test veri setine uygulanması ile elde edilen doğruluk, kesinlik, duyarlılık, F1 skor ve ROC AUC doğruluk oranlarına, hata matrislerine, ROC eğrisi grafiklerine ve yorumlara yer verilmiştir. Tüm algoritmaların eğitim veri setleri ile model oluşturulmasında çapraz doğrulama (Cross-Validation) k=5 değeri kullanılmıştır.

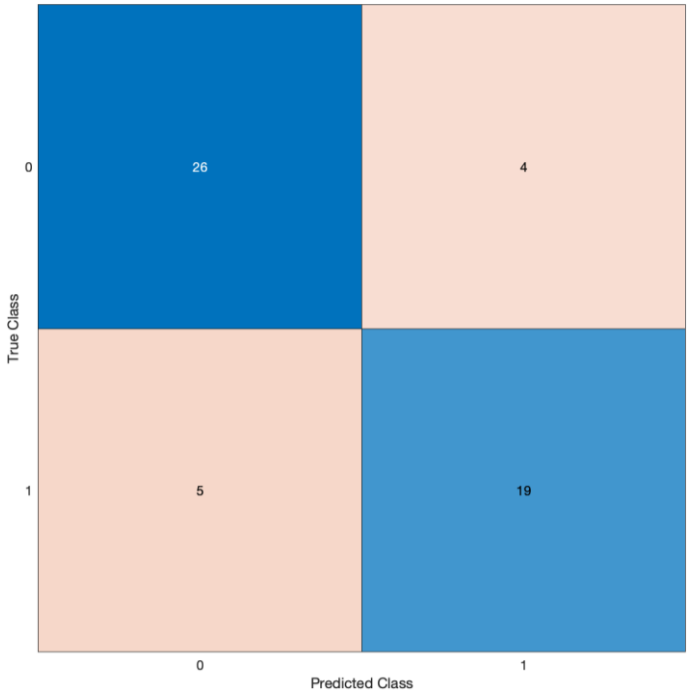
### 4.1.1. Lineer Diskriminant Analizi

Tez çalışmasındaki test veri setine lineer diskriminant analizi algoritması uygulanmış ve kalp hastalığı olmayan 26 kişinin ve kalp hastalığı olan 19 kişi için yapılan tahmin doğru çıkmıştır. Geri kalan kalp hastalığı bulunmayan 4 kişi için ve kalp hastalığı bulunan 5 kişi için yanlış tahmin sonuçları elde edilmiştir. Lineer diskriminant analizi uygulanan modelde diskriminant türü olarak diagonal quadratic seçilerek ve 30 tekrar ile en yüksek test veri seti doğruluk oranı %83.3 olarak, ROC Eğrisi Altındaki Alan (AUC) değeri %93 olarak bulunmuştur.

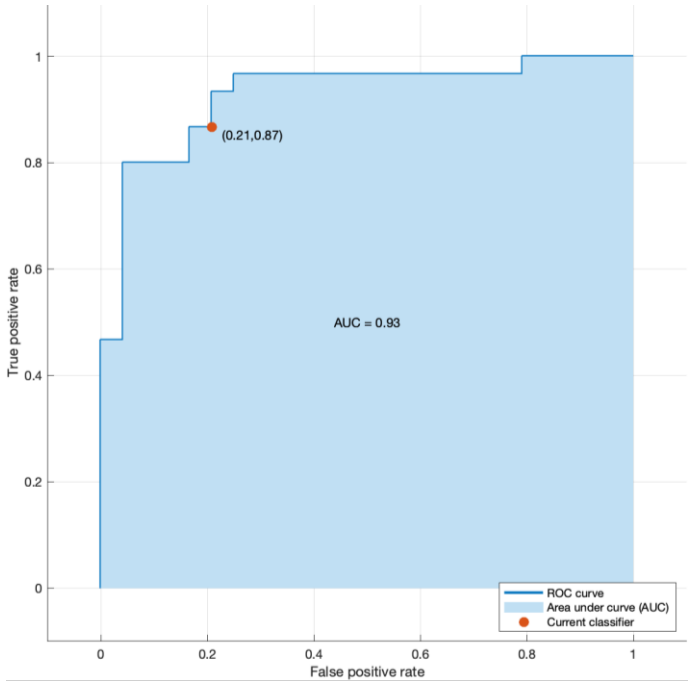
Eğitim Veri Seti Sonuçları
Doğruluk (Değerlendirme) : %85.6
Tahmin Hızı : ~6500 gözlem/saniye
Eğitim Süresi : 68.21 saniye

**Tablo 4.1** - LDA Algoritması Test Veri Seti Doğruluk Oranları

| Model | Doğruluk | Kesinlik | Duyarlılık | F1 Skor | AUC |
|---|---|---|---|---|---|
| LDA | %83.3 | %82.6 | %79.1 | %80.8 | %93 |




**Şekil 18.1** - LDA Algoritması Test Veri Seti Hata Matrisi




**Şekil 4.2** - LDA Algoritması Test Veri Seti ROC Eğrisi

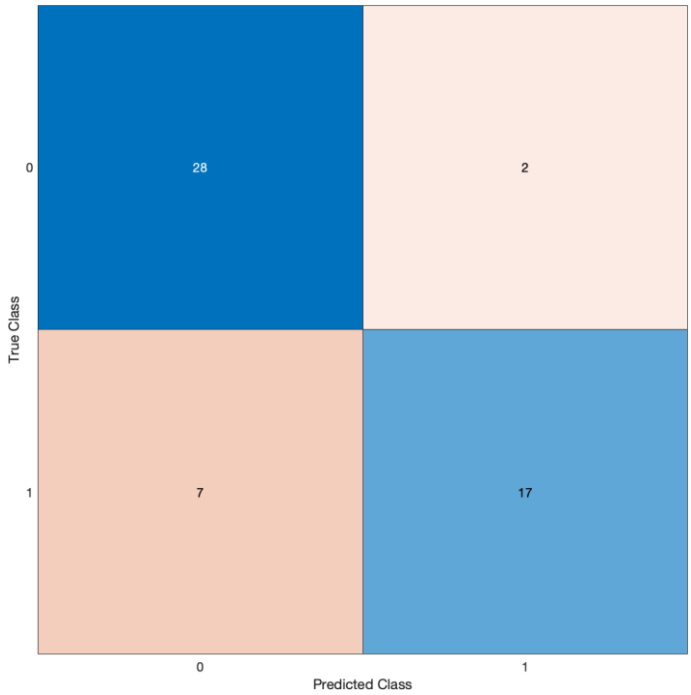
### 4.1.2. K-En Yakın Komşu Algoritması

Şekil 4.3'te gösterilen test veri setine ait hata matrisinde 54 hastaya ait değerler bulunmaktadır. Bu değerlerden kalp hastalığı olmayan 28 kişi ve kalp hastalığı olan 17 kişi doğru tahmin edilmiş, kalp hastalığı olmayan 2 kişi ve kalp hastalığı olan 7 kişi için yanlış tahminde bulunulmuştur. KNN algoritması k=108 değeri ve 30 tekrar uygulanan modelde test veri setine ait doğruluk oranı % 83.3 olarak ve ROC Eğrisi Altındaki Alan (AUC) değeri %92 olarak bulunmuştur.

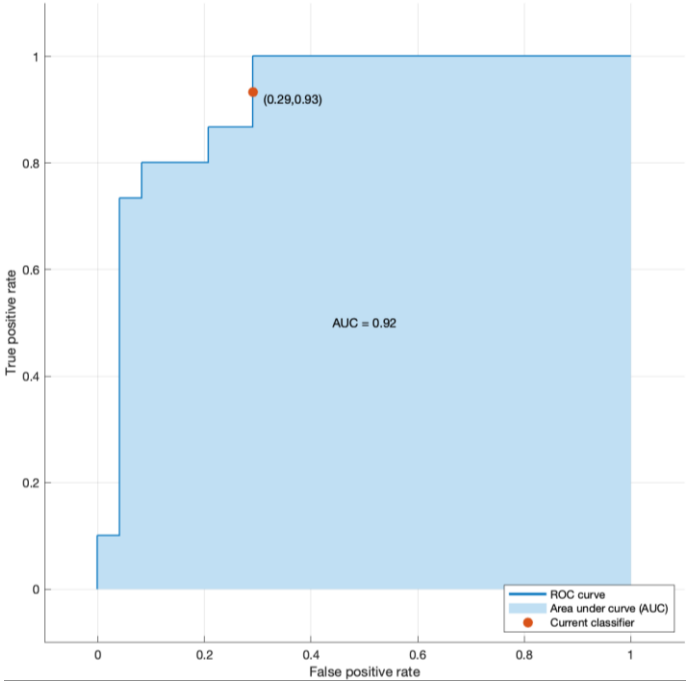
Eğitim Veri Seti Sonuçları
Doğruluk (Değerlendirme) : %85.2
Tahmin Hızı : ~4400 gözlem/saniye
Eğitim Süresi : 61.735 saniye

**Tablo 4.2** - KNN Algoritması Test Veri Seti Doğruluk Oranları

| Model | Doğruluk | Kesinlik | Duyarlılık | F1 Skor | AUC |
|---|---|---|---|---|---|
| KNN | %83.3 | %89.5 | %70.8 | 79% | %92 |




**Şekil 4.19** - KNN Algoritması Test Veri Seti Hata Matrisi




**Şekil 4.20** - KNN Algoritması Test Veri Seti ROC Eğrisi

### 4.1.3. Lojistik Regresyon

Lojistik regresyon algoritması uygulanan test veri setinde 54 hasta için tahminde bulunulmuştur. Kalp hastalığı bulunmayan 28 kişi için ve kalp hastalığı bulunan 20 kişi için doğru tahmin yapılmıştır. Kalp hastalığı olmayan 2 kişi için ve kalp hastalığı bulunan 4 kişi için yanlış tahminde bulunulmuştur. Lojistik regresyon algoritması uygulanan modelde test veri setinin doğruluk oranı %88.9 olarak ve ROC Eğrisi Altındaki Alan (AUC) değeri %91 olarak bulunmuştur.

Eğitim Veri Seti Sonuçları

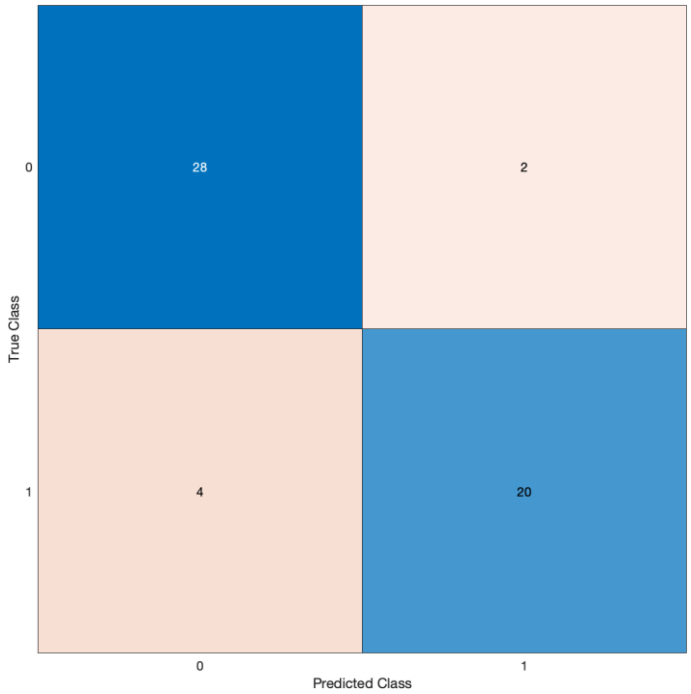
Doğruluk (Değerlendirme) : %83.3

Tahmin Hızı : ~2300 gözlem/saniye

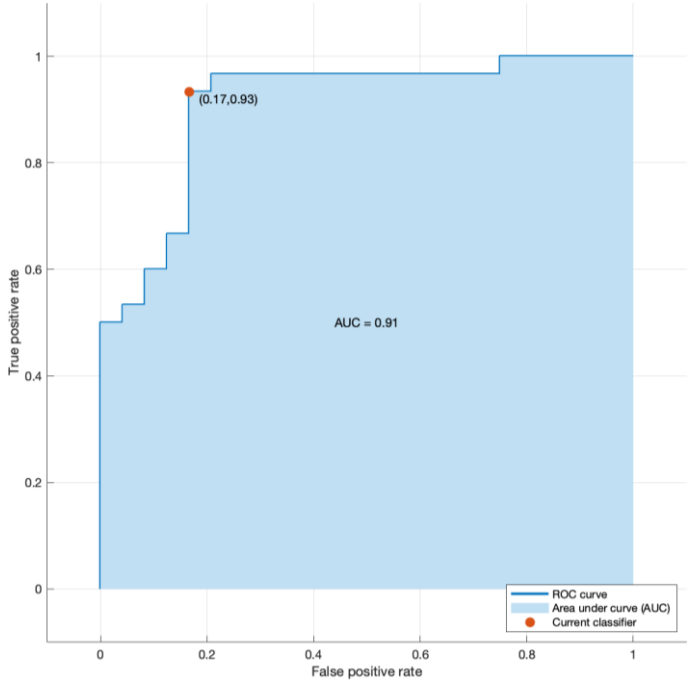
Eğitim Süresi : 4.6194 saniye

**Tablo 4.3** - LR Algoritması Test Veri Seti Doğruluk Oranları

| Model | Doğruluk | Kesinlik | Duyarlılık | F1 Skor | AUC |
|---|---|---|---|---|---|
| LR | %88.9 | %91 | %83.3 | %86.7 | %91 |




**Şekil 4.21** - LR Algoritması Test Veri Seti Hata Matrisi




**Şekil 4.22** - LR Algoritması Test Veri Seti ROC Eğrisi

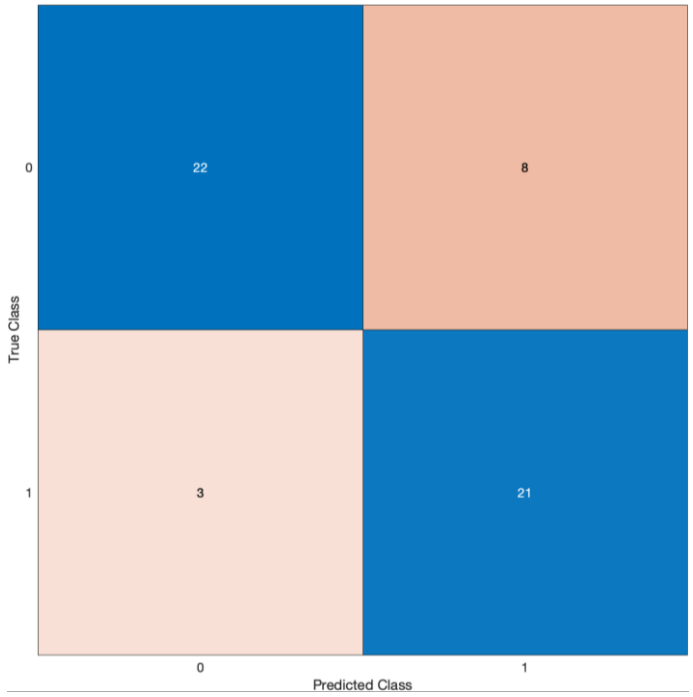
### 4.1.4. Karar Ağacı

Şekil 4.7'de gösterilen 54 hastaya ait hata matrisinde kalp hastalığı olmayan 22 kişi ve kalp hastalığı bulunan 21 kişi için doğru tahminde bulunulmuştur. Geri kalan kalp hastalığı bulunmayan 8 kişi ve kalp hastalığı bulunan 3 kişi için yanlış tahminde bulunulmuştur. Test veri setine karar ağacı algoritması 30 tekrar ve maksimum bölme sayısı 4 olarak uygulanmış, test veri setinin doğruluk oranı %79.6 olarak ve ROC Eğrisi Altındaki Alan (AUC) değeri %81 olarak bulunmuştur.

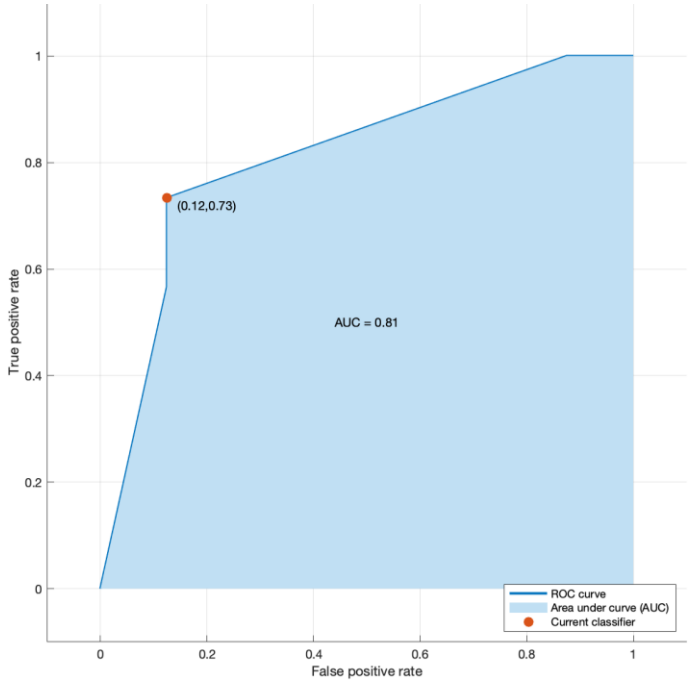
Eğitim Veri Seti Sonuçları
Doğruluk (Değerlendirme) : %77.8
Tahmin Hızı : ~10000 gözlem/saniye
Eğitim Süresi : 40.639 saniye

**Tablo 4.4** - KA Algoritması Test Veri Seti Doğruluk Oranları

| Model | Doğruluk | Kesinlik | Duyarlılık | F1 Skor | AUC |
|---|---|---|---|---|---|
| KA | %79.6 | %72.4 | %87.5 | %79.2 | %81 |




**Şekil 4.23** - KA Algoritması Test Veri Seti Hata Matrisi




**Şekil 4.24** - KA Algoritması Test Veri Seti ROC Eğrisi

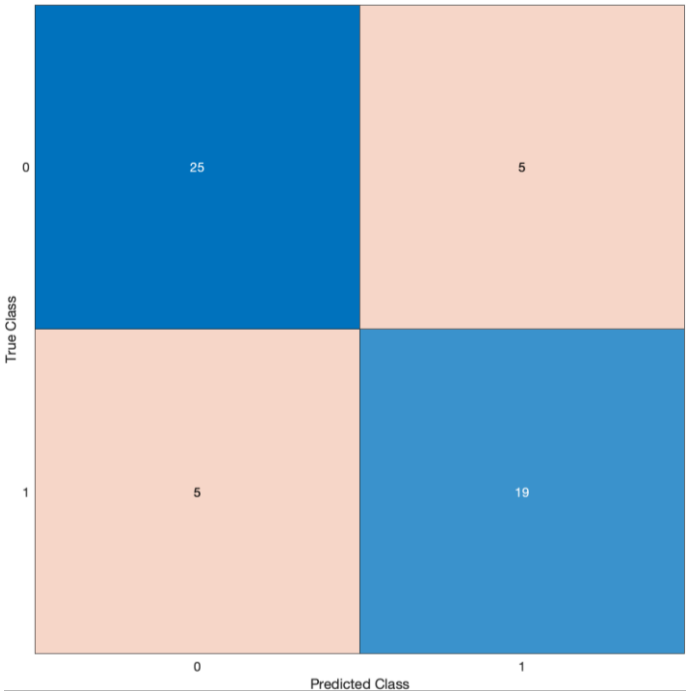
### 4.1.5. Destek Vektör Makineleri

Destek vektör makineleri algoritması seçilerek oluşturulan model ile test edilen ve 54 hasta kaydı içeren test veri setinde kalp hastalığı bulunmayan 25 kişi ve kalp hastalığı bulunan 19 kişi için doğru tahmin yapılmıştır. Kalp hastalığı bulunmayan 5 kişi ve kalp hastalığı bulunan diğer 5 kişi için tahminler yanlış çıkmıştır. Destek vektör makineleri algoritması uygulanan test veri setinde kernel fonksiyonu olarak lineer model ve 30 tekrar ile test veri seti doğruluk oranı % 81.5 olarak ve ROC Eğrisi Altındaki Alan (AUC) değeri %89 olarak bulunmuştur.

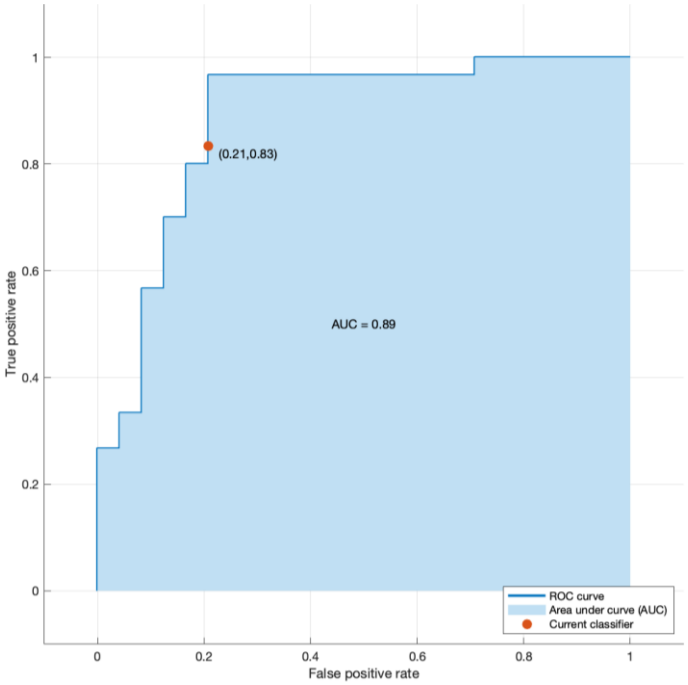
Eğitim Veri Seti Sonuçları
Doğruluk (Değerlendirme) : %84.7
Tahmin Hızı : ~9800 gözlem/saniye
Eğitim Süresi : 177.13saniye

**Tablo 4.5** - DVM Test Veri Seti Doğruluk Oranları

| Model | Doğruluk | Kesinlik | Duyarlılık | F1 Skor | AUC |
|---|---|---|---|---|---|
| DVM | %81.5 | %79.1 | %79.1 | %79.1 | %89 |




**Şekil 4.25** - DVM Algoritması Test Veri Seti Hata Matrisi




**Şekil 4.26** - DVM Algoritması Test Veri Seti ROC Eğrisi

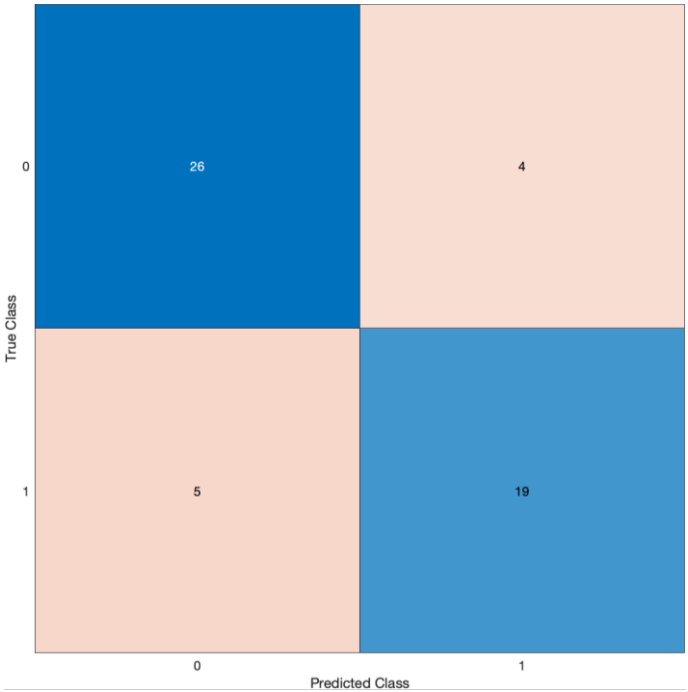
### 4.1.6. Naive Bayes Algoritması

Naive Bayes algoritması uygulanarak oluşturulan model ile toplamda 54 hastaya ait test veri setinde kalp hastalığı olmayanlardan 26 kişi için ve kalp hastalığı olan 19 kişi için doğru tahmin yapılmıştır. Kalp hastalığı olmayan 4 kişi ve kalp hastalığı olan 5 kişi için yanlış tahminde bulunulmuş ve en yüksek test veri seti doğruluk oranı kernel türü gaussian seçilerek ve 30 tekrar uygulanarak %83.3 olarak ve ROC Eğrisi Altındaki Alan (AUC) değeri %93 olarak bulunmuştur.

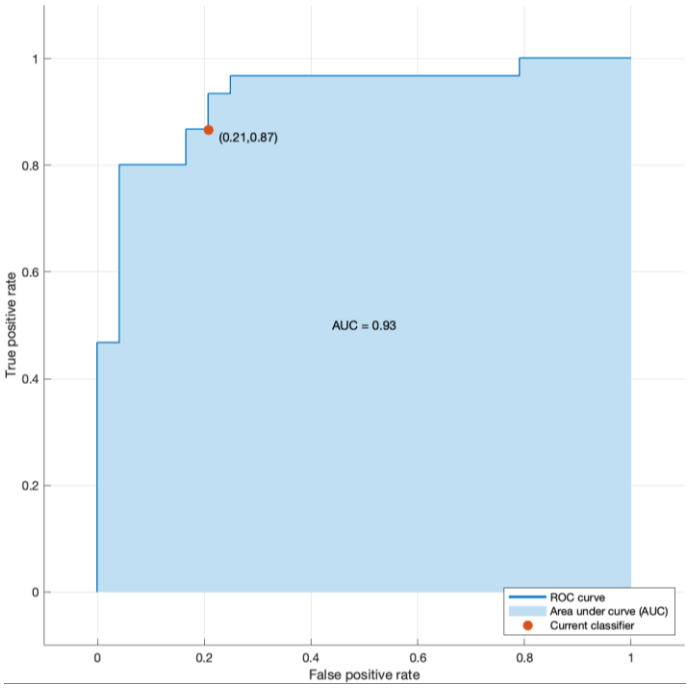
Eğitim Veri Seti Sonuçları
Doğruluk (Değerlendirme) : %85.6
Tahmin Hızı : ~6900 gözlem/saniye
Eğitim Süresi : 99.574 saniye

**Tablo 4.6** - NB Algoritması Test Veri Seti Doğruluk Oranları

| Model | Doğruluk | Kesinlik | Duyarlılık | F1 Skor | AUC |
|---|---|---|---|---|---|
| NB | %83.3 | %82.6 | %79.1 | %80.8 | %93 |




**Şekil 4.27** - NB Algoritması Test Veri Seti Hata Matrisi




**Şekil 4.28** - NB Algoritması Test Veri Seti ROC Eğrisi

### 4.1.7. Toplu Öğrenme Algoritmaları

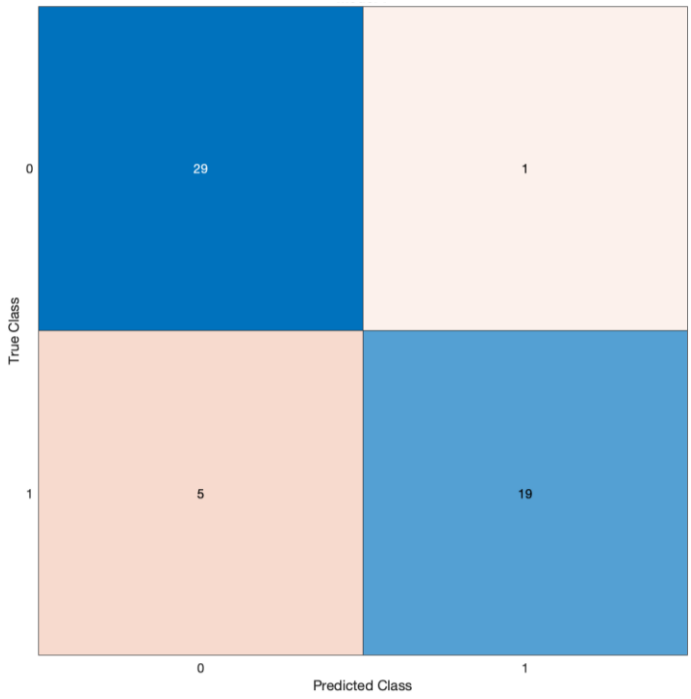
Şekil 4.13'te gösterilen hata matrisinde 29 kişinin kalp hastalığının bulunmadığı ve 19 kişinin kalp hastalığına sahip olması doğru tahmin edilmiştir. 1 kişinin kalp

hastalığına sahip olmaması ve 5 kişinin kalp hastalığının bulunması yanlış tahmin edilmiştir. Toplu öğrenme algoritmaları test verisine LogitBoost öğrenme metodu kullanılarak 30 tekrar uygulanmış ve test veri setinin doğruluk oranı %88.9 olarak ve ROC Eğrisi Altındaki Alan (AUC) değeri %91 olarak bulunmuştur.

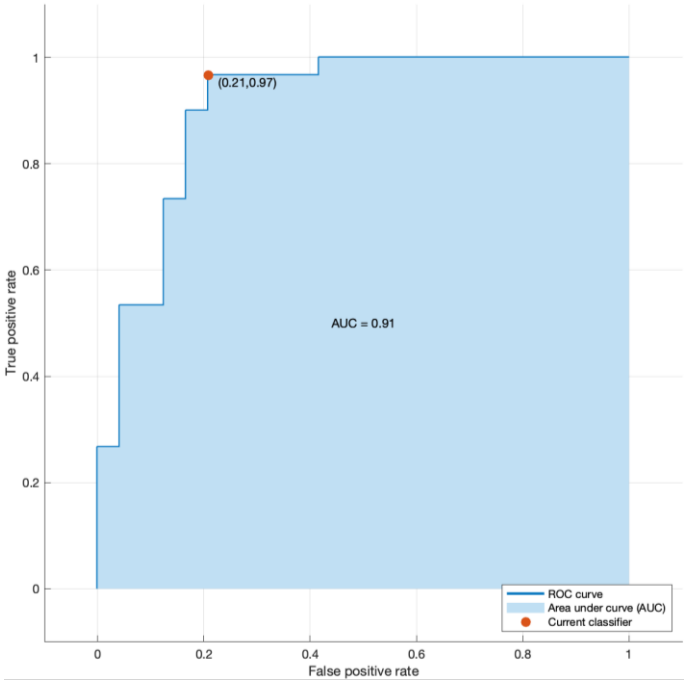
Eğitim Veri Seti Sonuçları
Doğruluk (Değerlendirme) : %83.8
Tahmin Hızı : ~1500 gözlem/saniye
Eğitim Süresi : 174.12 saniye

**Tablo 4.7** - TÖ Algoritması Test Veri Seti Doğruluk Oranı

| Model | Doğruluk | Kesinlik | Duyarlılık | F1 Skor | AUC |
|---|---|---|---|---|---|
| TÖ | %88.9 | %95 | %79.1 | %86.4 | %91 |




**Şekil 4.29** - TÖ Algoritması Test Veri Seti Hata Eğrisi




**Şekil 4.30** - TÖ Algoritması Test Veri Seti ROC Eğrisi

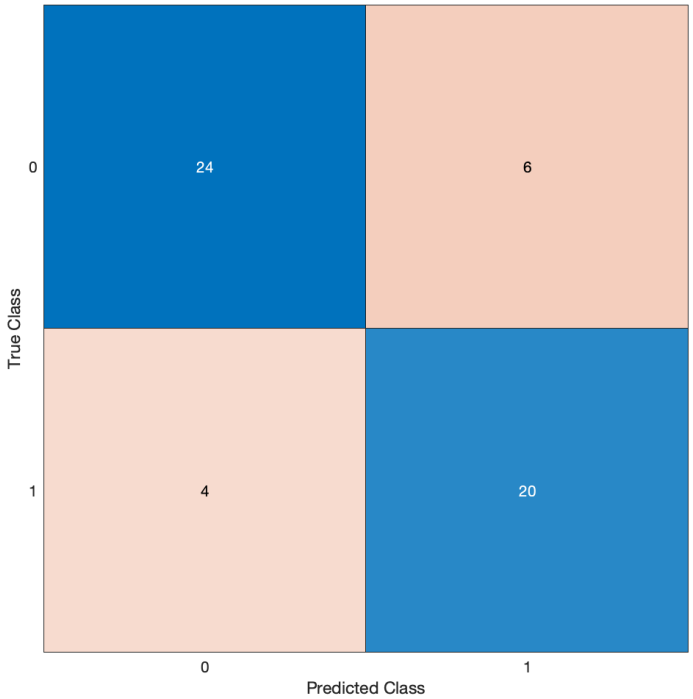
### 4.1.8. Sinir Ağları

Sinir ağları algoritmasının uygulandığı test verisinde kalp hastalığı bulunmayan 24 kişi için ve kalp hastalığı bulunan 20 kişi için doğru tahmin yapılmıştır. Kalp hastalığı bulunmayan 6 kişi ve kalp hastalığı bulunan 4 kişi için yanlış tahminde bulunulmuştur. 100 katmanlı sinir ağları algoritması ile 1000 tekrar uygulanan modelde test veri setinin doğruluk oranı %81.5 olarak ve ROC Eğrisi Altındaki Alan (AUC) %82 olarak bulunmuştur.

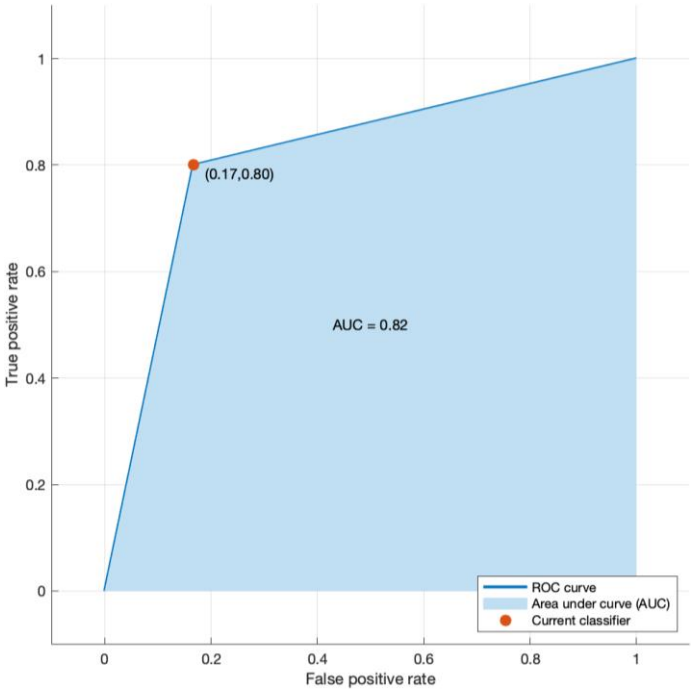
Eğitim Veri Seti Sonuçları
Doğruluk (Değerlendirme) : %74.5
Tahmin Hızı : ~4700 gözlem/saniye
Eğitim Süresi : 3.0539 saniye

**Tablo 4.8** - SA Algoritması Doğruluk Oranları

| Model | Doğruluk | Kesinlik | Duyarlılık | F1 Skor | AUC |
|---|---|---|---|---|---|
| SA | %81.5 | %76.9 | %83.3 | %80 | %82 |




**Şekil 4.31** - SA Algoritması Test Veri Seti Hata Matrisi




**Şekil 4.32** - SA Algoritması Test Veri Seti ROC Eğrisi

## 4.2. Yorumlar

Tez çalışmasında kullanılan algoritmalar ile eğitim veri seti kullanılarak modeller oluşturulmuş ve daha sonra bu model ile tahminde bulunmak için test veri seti kullanılmıştır. Tablo 4.9'da gösterildiği gibi eğitim veri setinin değerlendirmesinde en yüksek doğruluk oranlarını, birbirine yakın tahmin hızları ile aynı doğruluk oranı (%85.6) ile lineer diskriminant analizi ve Naive Bayes sınıflandırma algoritmaları olarak vermiştir. Ancak bu algoritmalar test veri setinde daha düşük doğruluk oranı (%83.3) vermişlerdir.

**Tablo 4.9** - Algoritmaların Eğitim Sonuçları

| Model | Doğruluk(Değerlendirme) | Tahmin Hızı (gözlem/saniye) | Eğitim Süresi (saniye ) |
|---|---|---|---|
| LDA | **%85.6** | ~6500 | 68.21 |
| KA | %77.8 | ~10000 | 40.63 |
| LR | %83.3 | ~2300 | 4.61 |
| NB | **%85.6** | ~6900 | 99.57 |
| DVM | %84.7 | ~9800 | 177.13 |
| KNN | %85.2 | ~4400 | 61.73 |
| TÖ | %83.8 | ~1500 | 174.2 |
| SA | %74.5 | ~4700 | 3.05 |

Aşağıda Tablo 4.10'da gösterildiği gibi test verisinden elde edilen tahminlerde en yüksek doğruluk oranlarını %88.9 oranı ile toplu öğrenme ve lojistik regresyon algoritmaları vermiştir ancak modellerin doğruluk değerlendirmelerinde sadece doğruluk oranları yeterli olmadığı için ROC Eğrisinin altındaki alanın (AUC) da değerlendirilmesi gerekir. Doğruluk değerleri "Doğru" tahmin edilen değerlerin (doğru pozitif ve yanlış pozitif) tüm değerlere (doğru pozitif, yanlış pozitif, doğru negatif ve yanlış negatif) bölümünü gösterir. ROC Eğrisinin Altındaki Alan (AUC) modellerin doğruluğunda daha kesin sonuçlar vermektedir. Algoritmaların ROC Eğrisi Altındaki Alan (AUC) doğruluk oranlarında lineer diskriminant analizi (%93), k-en yakın komşu (%92), lojistik regresyon (%91) ve toplu öğrenme algoritmaları (%91) oranları ile birbirine çok yakın sonuçlar vermiştir. Bu durumda ikinci bir ölçüt olarak algoritmaların doğruluk oranlarına bakıldığında en yüksek doğruluk oranlarını %88.9 olarak lojistik regresyon ve toplu öğrenme algoritmalarının verdiği gözlemlenmiştir.

**Tablo 4.10** - Algoritmaların Test Veri Seti Doğruluk Oranları

| **Model** | **Doğruluk** | **Kesinlik** | **Duyarlılık** | **F1 Skor** | **AUC** |
|---|---|---|---|---|---|
| LDA | %83.3 | %82.6 | %79.1 | %80.8 | %93 |
| KNN | %83.3 | %89.5 | %70.8 | %79.0 | %92 |
| LR | **%88.9** | %91 | %83.3 | %86.7 | %91 |
| KA | %79.6 | %72.4 | %87.5 | %79.2 | %81 |
| DVM | %81.5 | %79.1 | %79.1 | %79.1 | %89 |
| NB | %83.3 | %82.6 | %79.1 | %80.8 | %93 |
| TÖ | **%88.9** | %95 | %79.1 | %86.4 | %91 |
| SA | %81.5 | %76.9 | %83.3 | %80 | %82 |

# BÖLÜM 5. SONUÇ

Tez çalışmasında sınıflandırma problemi için mevcut veri seti eğitim ve test veri seti olarak ikiye ayrılmış ve makine öğrenmesi metodunda denetimli öğrenmenin sınıflandırma algoritmalarından lineer diskriminant analizi, k-en yakın komşu algoritması, lojistik regresyon, karar ağacı, destek vektör makineleri, Naive Bayes, toplu öğrenme ve sinir ağları algoritmaları ile tahmin çalışmasında bulunulmuş, algoritmaların doğruluk, kesinlik, duyarlılık, F1 skor ve ROC Eğrisi Altındaki Alan (AUC) değerlerine yer verilmiştir. Yapılan araştırmada ROC Eğrisi Altındaki Alan (AUC) değerleri birden fazla algoritmada birbirine çok yakın çıkmıştır. Bu yüzden ikinci bir ölçüt olarak test veri setinin doğruluk değerleri göz önünde bulundurulmuştur. Buna göre lojistik regresyon ve toplu öğrenme algoritmaları en yüksek sonucu vermiştir. Toplu öğrenme algoritmalarının performansının yüksek olduğu bilinmektedir ancak çalışmalarda sadece bir algoritma ile değil diğer algoritmalar da denenerek gözlem yapılmalı ve sonuca ulaşılmalıdır.

Kalp hastalığı ile ilgili veri setine dayanılarak yapılan çalışmada elde edilen modeller ile yeni hasta verilerine ait veri setleri için doğruluk tahminlerinde bulunulabilir. Kalp hastalıkları üzerinde çalışan uzmanlar yeni hasta verileri ile alınan doğruluk tahminlerine kendi yorumlarını da katarak hastalık teşhisi için tahminde bulunabilirler.

# KAYNAKÇA


[1] M. M. Ali, B. K. Paul, K. Ahmed, F. M. Bui, J. M. W. Quinn, and M. A. Moni, "Heart disease prediction using supervised machine learning algorithms: Performance analysis and comparison," *Comput. Biol. Med.*, vol. 136, no. May, p. 104672, 2021, doi: 10.1016/j.compbiomed.2021.104672.

[2] K. Arul Jothi, S. Subburam, V. Umadevi, and K. Hemavathy, "Heart disease prediction system using machine learning," *Mater. Today Proc.*, pp. 1–3, 2021, ISSN 2214-7853, doi: 10.1016/j.matpr.2020.12.901.

[3] M. Diwakar, A. Tripathi, K. Joshi, M. Memoria, P. Singh, and N. Kumar, "Latest trends on heart disease prediction using machine learning and image fusion," *Mater. Today Proc.*, vol. 37, no. Part 2, pp. 3213–3218, 2020, doi: 10.1016/j.matpr.2020.09.078.

[4] A. Kondababu, V. Siddhartha, B. B. Kumar, and B. Penumutchi, "A comparative study on machine learning based heart disease prediction," *Mater. Today Proc.*, pp. 1–5, 2021, ISSN 2214-7853, doi: 10.1016/j.matpr.2021.01.475.

[5] R. Hagan, C. J. Gillan, and F. Mallett, "Comparison of machine learning methods for the classification of cardiovascular disease," *Informatics Med. Unlocked*, vol. 24, no. May, p. 100606, 2021, doi: 10.1016/j.imu.2021.100606.

[6] V. Shorewala, "Early detection of coronary heart disease using ensemble techniques," *Informatics Med. Unlocked*, no. June, p. 100655, 2021, doi: 10.1016/j.imu.2021.100655.

[7] S. P. Patro, G. S. Nayak, and N. Padhy, "Heart disease prediction by using novel optimization algorithm: A supervised learning prospective," *Informatics Med. Unlocked*, vol. 26, p. 100696, 2021, doi: 10.1016/j.imu.2021.100696.

[8] V. Sharma, S. Yadav, and M. Gupta, "Heart Disease Prediction using Machine Learning Techniques," *Proc. - IEEE 2020 2nd Int. Conf. Adv. Comput. Commun. Control Networking, ICACCCN 2020*, vol. 1, no. 6, pp. 177–181, 2020, doi: 10.1109/ICACCCN51052.2020.9362842.

[9] J. Patel, A. A. Khaked, J. Patel, and J. Patel, "Heart Disease Prediction Using Machine Learning," *Lect. Notes Networks Syst.*, vol. 203 LNNS, pp. 653–665, 2021, doi: 10.1007/978-981-16-0733-2_46.

[10] J. Patel, A. A. Khaked, J. Patel, and J. Patel, "Heart Disease Prediction Using Machine Learning," *Lect. Notes Networks Syst.*, vol. 203 LNNS, no. Inccst 20, pp. 653–665, 2021, doi: 10.1007/978-981-16-0733-2_46.

[11] R. Katarya and S. K. Meena, "Machine Learning Techniques for Heart Disease Prediction: A Comparative Study and Analysis," *Health Technol. (Berl).*, vol. 11, no. 1, pp. 87–97, 2021, doi: 10.1007/s12553-020-00505-7.

[12] R. Bharti, A. Khamparia, M. Shabaz, G. Dhiman, S. Pande, and P. Singh, "Prediction of Heart Disease Using a Combination of Machine Learning and Deep Learning," *Comput. Intell. Neurosci.*, vol. 2021, 2021, doi: 10.1155/2021/8387680.

[13] A. K. Dwivedi, "Performance evaluation of different machine learning techniques for prediction of heart disease," *Neural Comput. Appl.*, vol. 29, no. 10, pp. 685–693, 2018, doi: 10.1007/s00521-016-2604-1.

[14] E. Maini, B. Venkateswarlu, B. Maini, and D. Marwaha, "Machine learning–based heart disease prediction system for Indian population: An exploratory study done in South India," *Med. J. Armed Forces India*, vol. 77, no. 3, pp. 302–311, 2021, doi: 10.1016/j.mjafi.2020.10.013.

[15] R. R. Sanni and H. S. Guruprasad, "Analysis of performance metrics of heart failured patients using Python and machine learning algorithms," *Glob. Transitions Proc.*, vol. 2, no. 2, pp. 233–237, 2021, doi: 10.1016/j.gltp.2021.08.028.

[16] P. Kalpana, S. Shiyam Vignesh, L. M. P. Surya, and V. Vishnu Prasad, "Prediction of Heart Disease Using Machine Learning," *J. Phys. Conf. Ser.*, vol. 1916, no. 1, 2021, doi: 10.1088/1742-6596/1916/1/012022.

[17] R. C. Mallela, R. L. Bhavani, and B. Ankayarkanni, *Disease Prediction Using Machine Learning Techniques*. Springer Singapore, 2021.

[18] I. Journal, "IRJET- Predicting Heart Disease using Machine Learning Algorithm."

[19] H. Jindal, S. Agrawal, R. Khera, R. Jain, and P. Nagrath, "Heart disease prediction using machine learning algorithms," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1022, no. 1, 2021, doi: 10.1088/1757-899X/1022/1/012072.

[20] M. T., D. Mukherji, N. Padalia, and A. Naidu, "A Heart Disease Prediction Model using SVM-Decision Trees-Logistic Regression (SDL)," *Int. J. Comput. Appl.*, vol. 68, no. 16, pp. 11–15, 2013, doi: 10.5120/11662-7250.

[21] C. H. Macphee and A. Daugherty, "Cardiovascular diseases," *Drug Discov. Today Ther. Strateg.*, vol. 5, no. 1, pp. 1–3, 2008, doi: 10.1016/j.ddstr.2008.11.002.

[22] D. T. Pham and P. T. N. Pham, "Artificial intelligence in engineering," *Int. J. Mach. Tools Manuf.*, vol. 39, no. 6, pp. 937–949, 1999, doi: 10.1016/S0890-6955(98)00076-5.

[23] C. Zhang and Y. Lu, "Study on artificial intelligence: The state of the art and future prospects," *J. Ind. Inf. Integr.*, vol. 23, no. May, p. 100224, 2021, doi: 10.1016/j.jii.2021.100224.

[24] P. H. Winston, "On Computing Machinery and Intelligence," *Bost. Stud. Philos. Hist. Sci.*, vol. 324, pp. 265–278, 2017, doi: 10.1007/978-3-319-53280-6_11.

[25] A. Entwistle, “What Is Artificial Intelligence?,” *Eng. Mater. Des.*, vol. 32, no. 3, pp. 1–14, 1988, doi: 10.7551/mitpress/12518.003.0004.

[26] T. Rademacher, “Artificial intelligence and law enforcement,” *Regul. Artif. Intell.*, vol. 35, no. 4, pp. 225–254, 2019, doi: 10.1007/978-3-030-32361-5_10.

[27] S. R. and P. Norvig, *Artificial Intelligence A Modern Approach 4th Ed.*, vol. 53, no. 9. 2019.

[28] European Commission, “Artificial Intelligence for Europe - Communication,” *Brussels, 25.4.2018 COM(2018) 237 Final*, 2018, [Online]. Available: https://ec.europa.eu/transparency/regdoc/rep/1/2018/EN/COM-2018-237-F1-EN-MAIN-PART-1.PDF.

[29] D. Kirsch and J. Hurwitz, *Machine Learning by Judith Hurwitz and*. 2018.

[30] H. Liu and B. Lang, “Machine learning and deep learning methods for intrusion detection systems: A survey,” *Appl. Sci.*, vol. 9, no. 20, pp. 2881–2886, 2019, doi: 10.3390/app9204396.

[31] A. H. Jackson, “Machine learning,” *Expert Syst.*, vol. 5, no. 2, pp. 132–150, 1988, doi: 10.1111/j.1468-0394.1988.tb00341.

[32] A. L. Samuel, “Some Studies in Machine Learning,” *IBM J. Res. Dev.*, vol. 3, no. 3, pp. 210–229, 1959, [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5392560.

[33] A. Fattah, M. M., P. S., and T. F., “A Decision Tree Classification Model for University Admission System,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 3, no. 10, pp. 17–21, 2012, doi: 10.14569/ijacsa.2012.031003.

[34] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien, “Linear discriminant analysis: A detailed tutorial,” *AI Commun.*, vol. 30, no. 2, pp. 169–190, 2017, doi: 10.3233/AIC-170729.

[35] T. Bahraini, S. M. Hosseini, M. Ghasempour, and H. S. Yazdi, “Density-oriented linear discriminant analysis[Formula presented],” *Expert Syst. Appl.*, vol. 187, no. March 2021, p. 115946, 2022, doi: 10.1016/j.eswa.2021.115946.

[36] H. A. Park, “An introduction to logistic regression: From basic concepts to interpretation with particular attention to nursing domain,” *J. Korean Acad. Nurs.*, vol. 43, no. 2, pp. 154–164, 2013, doi: 10.4040/jkan.2013.43.2.154.

[37] C. H. Park and H. Park, “A comparison of generalized linear discriminant analysis algorithms,” *Pattern Recognit.*, vol. 41, no. 3, pp. 1083–1097, 2008, doi: 10.1016/j.patcog.2007.07.022.

[38] A. Sharma and K. K. Paliwal, “A deterministic approach to regularized linear discriminant analysis,” *Neurocomputing*, vol. 151, no. P1, pp. 207–214, 2015, doi: 10.1016/j.neucom.2014.09.051.

[39] R. Goel, “Heart Disease Prediction Using Various Algorithms of Machine Learning,” *SSRN Electron. J.*, 2021, doi: 10.2139/ssrn.3884968.

[40] Ankur Rohilla, “Comparative Analysis of Various Classification Algorithms in the Case of Fraud Detection,” *Int. J. Eng. Res.*, vol. V6, no. 09, pp. 118–122, 2017, doi: 10.17577/ijertv6is090047.

[41] N. Bhatia and Vandana, “Survey of Nearest Neighbor Techniques,” vol. 8, no. 2, pp. 302–305, 2010, [Online]. Available: http://arxiv.org/abs/1007.0085.

[42] K. Chomboon, P. Chujai, P. Teerarassammee, K. Kerdprasop, and N. Kerdprasop, “An Empirical Study of Distance Metrics for k-Nearest Neighbor Algorithm,” no. January 2015, pp. 280–285, 2015, doi: 10.12792/iciae2015.051.

[43] L. S. Zaremba and W. H. Smoleński, “Optimal portfolio choice under a liability constraint,” *Ann. Oper. Res.*, vol. 97, no. 1–4, pp. 131–141, 2000, doi: 10.1023/A.

[44] F. R. Ishengoma, “Authentication System for Smart Homes Based on ARM7TDMI-S and IRIS-Fingerprint Recognition Technologies,” *Int. J. Inf. Eng. Electron. Bus.*, vol. 6, no. 6, pp. 64–69, 2014, doi: 10.5815/ijieeb.2014.06.08.

[45] S. Kost, O. Rheinbach, and H. Schaeben, “Using logistic regression model selection towards interpretable machine learning in mineral prospectivity modeling,” *Geochemistry*, no. September, p. 125826, 2021, doi: 10.1016/j.chemer.2021.125826.

[46] E. Y. Boateng and D. A. Abaye, “A Review of the Logistic Regression Model with Emphasis on Medical Research,” *J. Data Anal. Inf. Process.*, vol. 07, no. 04, pp. 190–207, 2019, doi: 10.4236/jdaip.2019.74012.

[47] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, “Credit card fraud detection using machine learning techniques: A comparative analysis,” *Proc. IEEE Int. Conf. Comput. Netw. Informatics, ICCNI 2017*, vol. 2017-Janua, pp. 1–9, 2017, doi: 10.1109/ICCNI.2017.8123782.

[48] H. Sharma and S. Kumar, “A Survey on Decision Tree Algorithms of Classification in Data Mining,” *Int. J. Sci. Res.*, vol. 5, no. 4, pp. 2094–2097, 2016, doi: 10.21275/v5i4.nov162954.

[49] S. Cycles, “Chapter 9 Chapter 9,” *Cycle*, vol. 1897, no. Figure 1, pp. 44–45, 1989, doi: 10.1007/0-387-25465-X.

[50] D. Tanouz, R. R. Subramanian, D. Eswar, G. V. P. Reddy, A. R. Kumar, and C. H. V. N. M. Praneeth, “Credit card fraud detection using machine learning,” *Proc. - 5th Int. Conf. Intell. Comput. Control Syst. ICICCS 2021*, pp. 967–972, 2021, doi: 10.1109/ICICCS51141.2021.9432308.

[51] A. Priyam, R. Gupta, A. Rathee, and S. Srivastava, “Comparative Analysis of Decision Tree Classification Algorithms,” pp. 334–337, 2013.

[52] E. Osuna, R. Freund, and F. Girosi, “Support Vector Machines : Training and Applications,” *Massachusetts Inst. Technol.*, vol. 9217041, no. 1602, 1997.

[53] L. Saitta, “Support-Vector Networks,” vol. 297, pp. 273–297, 1995.

[54] D. Mezghani, S. Boujelbene, and N. Ellouze, “Evaluation of SVM Kernels and Conventional Machine Learning Algorithms for Speaker Identification,” *J. Hybrid Inf. Technol.*, vol. 3, no. 3, p. 12, 2010.

[55] N. Cristianini and J. Shawe-Taylor, Eds., *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

[56] J. Wei, Z. Jian-Qi, and Z. Xiang, “Face recognition method based on support vector machine and particle swarm optimization,” *Expert Syst. Appl.*, vol. 38, no. 4, pp. 4390–4393, 2011, doi: 10.1016/j.eswa.2010.09.108.

[57] P. Golbayani, I. Florescu, and R. Chatterjee, “A comparative study of forecasting corporate credit ratings using neural networks, support vector machines, and decision trees,” *North Am. J. Econ. Financ.*, vol. 54, no. April, p. 101251, 2020, doi: 10.1016/j.najef.2020.101251.

[58] M. Wadkar, F. Di Troia, and M. Stamp, “Detecting malware evolution using support vector machines,” *Expert Syst. Appl.*, vol. 143, p. 113022, 2020, doi: 10.1016/j.eswa.2019.113022.

[59] M. C. Guerrero, J. S. Parada, and H. E. Espitia, “EEG signal analysis using classification techniques: Logistic regression, artificial neural networks, support vector machines, and convolutional neural networks,” *Heliyon*, vol. 7, no. 6, p. e07258, 2021, doi: 10.1016/j.heliyon.2021.e07258.

[60] G. Gursoy and A. Varol, “Prediction of Arrhythmia with Machine Learning Algorithms,” *9th Int. Symp. Digit. Forensics Secur. ISDFS 2021*, pp. 1–5, 2021, doi: 10.1109/ISDFS52919.2021.9486383.

[61] K. Vembandasamy, R. Sasipriya, and E. Deepa, “Heart Diseases Detection Using Naive Bayes Algorithm,” *Int. J. Innov. Sci. Eng. Technol.*, vol. 2, no. 9, pp. 441–444, 2015.

[62] B. Seref and E. Bostanci, “Performance of Naïve and Complement Naïve Bayes Algorithms Based on Accuracy, Precision and Recall Performance Evaluation Criterions,” *Int. J. Comput. Acad. Res.*, vol. 8, no. 5, pp. 75–92, 2019, [Online]. Available: http://www.meacse.org/ijcar.

[63] R. Blanquero, E. Carrizosa, P. Ramírez-Cobo, and M. R. Sillero-Denamiel, “Variable selection for Naïve Bayes classification,” *Comput. Oper. Res.*, vol. 135, p. 105456, 2021, doi: 10.1016/j.cor.2021.105456.

[64] G. Parthiban, A. S.K.Srivatsa, and A. Rajesh, “Diagnosis of Heart Disease for Diabetic Patients using Naive Bayes Method,” *Int. J. Comput. Appl.*, vol. 24, no.

3, pp. 7–11, 2011, doi: 10.5120/2933-3887.

[65] M. Singh, M. Wasim Bhatt, H. S. Bedi, and U. Mishra, “Performance of bernoulli’s naive bayes classifier in the detection of fake news,” *Mater. Today Proc.*, 2020, ISSN 2214-7853, doi: 10.1016/j.matpr.2020.10.896.

[66] Y. Zou, Henan-Ligong-Daxue Jiaozuo, ISCSCT (3 2010.08.14-15 Jiaozuo), and International Symposium Computer Science and Computational Technology (3 2010.08.14-15 Jiaozuo), *Proceedings : 14-15 Aug. 2010, Jiaozuo, China*, no. Iscsct. 2010.

[67] J. Yao and Y. Ye, “The effect of image recognition traffic prediction method under deep learning and naive Bayes algorithm on freeway traffic safety,” *Image Vis. Comput.*, vol. 103, p. 103971, 2020, doi: 10.1016/j.imavis.2020.103971.

[68] N. F. Rusland, N. Wahid, S. Kasim, and H. Hafit, “Analysis of Naïve Bayes Algorithm for Email Spam Filtering across Multiple Datasets,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 226, no. 1, 2017, doi: 10.1088/1757-899X/226/1/012091.

[69] Y. Li and W. Chen, “A comparative performance assessment of ensemble learning for credit scoring,” *Mathematics*, vol. 8, no. 10, pp. 1–19, 2020, doi: 10.3390/math8101756.

[70] A. Mohammed and R. Kora, “An Effective Ensemble Deep Learning Framework for Text Classification,” *J. King Saud Univ. - Comput. Inf. Sci.*,2021, ISSN 1319-1578, doi: 10.1016/j.jksuci.2021.11.001.

[71] R. Maclin, “Popular Ensemble Methods : An Empirical Study Popular Ensemble Methods : An Empirical Study,” *J. Artif. Intell. Res.*, vol. 11, no. July, pp. 169–198, 2016.

[72] P. D. A and S. Homayouni, “Bagging and Boosting Ensemble Classifiers for Classification of Comparative Evaluation,” 2021.

[73] G. Zhang and B. Fang, “LogitBoost classifier for discriminating thermophilic and mesophilic proteins,” *J. Biotechnol.*, vol. 127, no. 3, pp. 417–424, 2007, doi: 10.1016/j.jbiotec.2006.07.020.

[74] M. H. Kamarudin, C. Maple, T. Watson, and N. S. Safa, “A LogitBoost-Based Algorithm for Detecting Known and Unknown Web Attacks,” *IEEE Access*, vol. 5, no. November, pp. 26190–26200, 2017, doi: 10.1109/ACCESS.2017.2766844.

[75] J. E. Gentle, W. K. Härdle, and Y. Mori, *Handbook of computational statistics: Concepts and methods: Second Edition*, no. January 2012. 2012.

[76] J. Kazmaier and J. H. van Vuuren, “The power of ensemble learning in sentiment analysis,” *Expert Syst. Appl.*, vol. 187, no. June 2021, p. 115819, 2022, doi: 10.1016/j.eswa.2021.115819.

[77] B. V.A., M. Subramoniam, and L. Mathew, “Detection of COPD and Lung Cancer

with electronic nose using ensemble learning methods," *Clin. Chim. Acta*, vol. 523, no. October, pp. 231–238, 2021, doi: 10.1016/j.cca.2021.10.005.

[78] M. H. D. M. Ribeiro, R. G. da Silva, S. R. Moreno, V. C. Mariani, and L. dos S. Coelho, "Efficient bootstrap stacking ensemble learning model applied to wind power generation forecasting," *Int. J. Electr. Power Energy Syst.*, vol. 136, no. October 2021, p. 107712, 2022, doi: 10.1016/j.ijepes.2021.107712.

[79] M. S. Wanker, "Research Paper on Basic of Artificial Neural Network," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 2, no. 1, pp. 96–100, 2014.

[80] S. Sharma, S. Sharma, and A. Anidhya, "Understanding Activation Functions in Neural Networks," *Int. J. Eng. Appl. Sci. Technol.*, vol. 4, no. 12, pp. 310–316, 2020.

[81] S. Harvey and R. Harvey, "An introduction to artificial intelligence," *Appita J.*, vol. 51, no. 1, 1998, doi: 10.2514/6.1994-294.

[82] S. Agatonovic-Kustrin and R. Beresford, "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research," *J. Pharm. Biomed. Anal.*, vol. 22, no. 5, pp. 717–727, 2000, doi: 10.1016/S0731-7085(99)00272-1.

[83] A. T. C. Goh, "Back-propagation neural networks for modeling complex systems," *Artif. Intell. Eng.*, vol. 9, no. 3, pp. 143–151, 1995, doi: 10.1016/0954-1810(94)00011-S.

[84] K. Suzuki, *Artificial Neural Networks - Methodological Advances and Biomedical Applications*. InTech, 2011.

[85] K. Shihab, "A Backpropagation Neural Network for Computer Network Security," *J. Comput. Sci.*, vol. 2, no. 9, pp. 710–715, 2006, doi: 10.3844/jcssp.2006.710.715.

[86] S. Choi and Y. J. Kim, "Artificial neural network models for airport capacity prediction," *J. Air Transp. Manag.*, vol. 97, no. September, p. 102146, 2021, doi: 10.1016/j.jairtraman.2021.102146.

[87] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. E. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, p. e00938, 2018, doi: 10.1016/j.heliyon.2018.e00938.

[88] B. Rajoub, "Supervised and unsupervised learning," *Biomed. Signal Process. Artif. Intell. Healthc.*, no. January, pp. 51–89, 2020, doi: 10.1016/b978-0-12-818946-7.00003-2.

[89] A. D. Dongare, R. R. Kharde, and A. D. Kachare, "Introduction to Artificial Neural Network ( ANN ) Methods," *Int. J. Eng. Innov. Technol.*, vol. 2, no. 1, pp. 189–194, 2012, [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1082.1323&rep=rep1&type=pdf.

[90] C. L. Blake and C. J. Merz, “Statlog (Heart) Data Set” 1998. UCI repository of machine learning databases. University of California,Irvine. https://archive.ics.uci.edu/ml/datasets/statlog+(heart).

[91] M. T. Rouabah, A. Tounsi, and N. E. Belaloui, “Genetic algorithm with cross validation-based epidemic model and application to early diffusion of COVID-19 in Algeria,” *arXiv*, p. e01050, 2020, doi: 10.1016/j.sciaf.2021.e01050.