

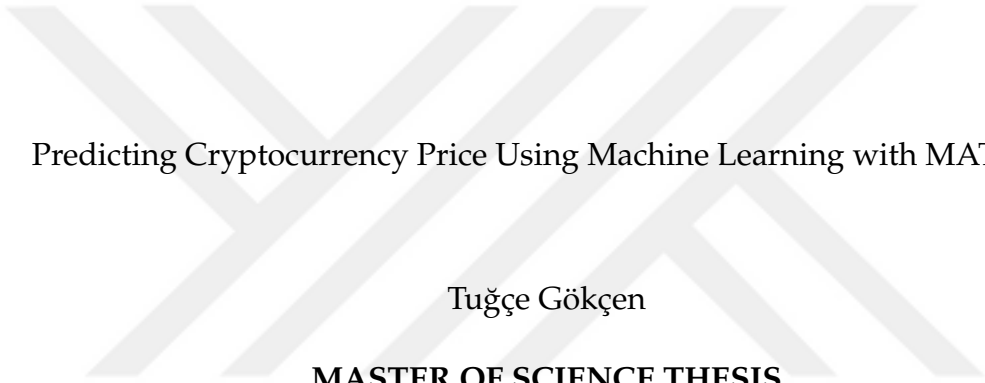
MATLAB ile Makine Öğrenmesi Kullanarak Kripto Para Fiyatının Tahmin Edilmesi

Tuğçe Gökçen

YÜKSEK LİSANS TEZİ

Matematik ve Bilgisayar Bilimleri Anabilim Dalı

Kasım 2023



Predicting Cryptocurrency Price Using Machine Learning with MATLAB

Tuğçe Gökçen

MASTER OF SCIENCE THESIS

Department of Mathematics and Computer Science

November 2023

MATLAB ile Makine Öğrenmesi Kullanarak Kripto Para Fiyatının Tahmin Edilmesi

Tuğçe Gökçen

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Eğitim ve Öğretim Yönetmeliği Uyarınca
Matematik ve Bilgisayar Bilimleri Anabilim Dalı
Bilgisayar Bilimleri Bilim Dalında
YÜKSEK LİSANS TEZİ
olarak hazırlanmıştır

Danışman: Prof. Dr. Alper Odabaş

Kasım 2023

ONAY

Matematik ve Bilgisayar Bilimleri Anabilim Dalı Bilgisayar Bilimleri Bilim Dalı Yüksek Lisans öğrencisi **Tuğçe Gökçen**'in YÜKSEK LİSANS tezi olarak hazırladığı "**MATLAB ile Makine Öğrenmesi Kullanarak Kripto Para Fiyatının Tahmin Edilmesi**" başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğin ilgili maddeleri uyarınca değerlendirilerek oybirliği ile kabul edilmiştir.

Danışman : Prof. Dr. Alper Odabaş

İkinci Danışman :

Yüksek Lisans Tez Savunma Jürisi:

Üye: Prof. Dr. Alper Odabaş

Üye: Doç. Dr. Ahmet Faruk Aslan

Üye: Dr. Öğr. Üyesi Mesut Aydemir

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun tarih ve sayılı kararıyla onaylanmıştır.

Prof.Dr. Fatma TÜMSEK
Enstitü Müdürü

ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Prof. Dr. Alper Odabaş danışmanlığında hazırlamış olduğum **“MATLAB ile Makine Öğrenmesi Kullanarak Kripto Para Fiyatının Tahmin Edilmesi”** başlıklı Yüksek Lisans tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel etik ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik ilke ve kurallara göre sunduğumu beyan ederim.

29/12/2023

Tuğçe Gökçen

ÖZET

Kripto para birimleri, merkezi parasal otoritelerin etkisi olmadan para transfer etmek için kriptografi kullanan merkezi olmayan dijital varlıklardır. Kripto para borsası gibi hızlı değişim gösteren borsalarda çoklu paralel anlık fiyat değişiminin izlenmesi insan için çok yorucu bir süreçtir. Buna çözüm olarak sürekli gözlem yaparak hızlı ve objektif kararlar verebilen bilgisayar yazılımları insanın yerini alabilir. Bu çalışmada kripto para sistemi içerisinde en yüksek hacime sahip olan Bitcoin(BTC) kapanış fiyatı analiz edilmiştir. Gaussian Process Regression (GPR) modelinin ve SMOTE yönteminin kullanıldığı çalışmada veri seti olarak, BTC'ye ait 25/07/2010 - 05/06/2022 arası döneme ait veriler kullanılmıştır. Giriş parametresi olarak açılış fiyatı, en yüksek-en düşük fiyat, hacim, dolar endeksi ve teknik analizde kullanılan bazı indikatörler ile çalışılmıştır. Eğitim ve test verilerinin ayrılmasında kfold yöntemi izlenmiştir. SMOTE uygulanan modellerde ortalama MAPE değeri 1887 ve ortamala R^2 değeri 0.99977 olarak bulunmuştur. Ayrıca veriden açılış fiyatı, en yüksek-en düşük fiyat çıkarılarak GPR modeli ve SMOTE yöntemine uygulanan GPR model fonksiyonları karşılaştırılmıştır.

Anahtar Kelimeler: Kripto para, GPR, SMOTE, Bitcoin

SUMMARY

Cryptography is used by cryptocurrencies to shift money without the intervention of centralized financial institutions. They are decentralized digital assets. On rapidly changing exchanges like those for crypto currencies, it is a tremendously taxing procedure for people to keep track of many simultaneous instantaneous price changes. As a solution to this, computer software that can make fast and objective decisions by constantly observing can replace humans. In this study, the closing price of Bitcoin (BTC), which has the highest volume in the crypto money system, is analyzed. In the study, in which the Gaussian Process Regression (GPR) model and the SMOTE method were used, data belonging to BTC for the period between 25/07/2010 and 05/06/2022 were used as the data set. Opening price, highest-lowest price, volume, dollar index and some indicators used in technical analysis were used as input parameters. The kfold method was followed in the separation of training and test data. The mean MAPE value was found to be 1887, and the mean R2 value was found to be 0.99977 in the models with SMOTE. In addition, the GPR model and the GPR model functions that were applied to the SMOTE method were compared by excluding the opening price, which was the price that was highest-lowest, from the data. It was carried out to determine which model performed better.

Keywords : Cryptocurrency, GPR, SMOTE, Bitcoin.

TEŞEKKÜR

Beni bu çalışmaya sevkeden ve yöneten, çalışma boyunca değerli yardımlarını esirgemeyen, Hocam, sayın **Prof. Dr. Alper Odabaş'a** ve aileme sonsuz saygı ve teşekkürlerimi sunarım.



İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	vi
SUMMARY	vii
TEŞEKKÜR	viii
İÇİNDEKİLER	ix
ŞEKİLLER DİZİNİ	xi
ÇİZELGELER DİZİNİ	xii
SİMGELER VE KISALTMALAR DİZİNİ	xiii
SİMGELER VE KISALTMALAR DİZİNİ	xiv
1. GİRİŞ VE AMAÇ	1
2. LİTERATÜR ARAŞTIRMASI	3
2.1. Kripto Para	3
2.1.1. Blockchain	3
2.1.2. Bitcoin	5
2.2. Kripto paraları etkileyen faktörler	6
2.2.1. Teknik analiz ve indikatörler	7
3. MAKİNE ÖĞRENMESİ	9
3.1. Gözetimli Öğrenme	11
3.1.1. Regresyon	11
3.1.2. Sınıflandırma	21
3.2. Gözetimsiz Öğrenme	27
3.2.1. Kümeleme	28
3.3. Yarı Denetimli Algoritmalar	33
3.4. Takviyeli Öğrenme	33
3.5. İlgili Çalışmalar	34
3.6. Makine Öğrenmesi ve MATLAB	40
3.6.1. Sistematiik Örneklem Teoremi	49
3.6.2. K-Katlamalı Çapraz Doğrulama (Kfold Cross Validation): . . .	51

3.6.3. Dengesiz veri setleri ve SMOTE	52
3.6.4. Regresyon	57
3.6.5. Sınıflandırma	67
3.6.6. Kümeleme	78
4. ARAŞTIRMA METODOLOJİSİ	85
4.1. Veri Seti	85
4.2. Modelin Oluşturulması ve Eğitilmesi	87
4.3. Değerlendirme Metrikleri	89
4.4. Model Sonuçları	89
5. SONUÇ VE ÖNERİLER	94
KAYNAKLAR DİZİNİ	95

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
2.1 BTC ve Diğer Para Birimleri Hacim Grafiği	5
2.2 Fincan Kulp Formasyonu	7
3.1 Yapay Zeka	9
3.2 Makine Öğrenmesi	10
3.3 Makine Öğrenmesi	10
3.4 k-Means Algoritması	29
3.5 NCA	46
3.6 k-Fold	51
3.7 Oversampling ve Undersampling	53
3.8 Genetik algoritma çalışma prensibi	56
3.9 Lineer Regresyon	60
3.10 Logistik Regresyon eğitim	62
3.11 Logistik Regresyon test	63
3.12 Çoklu Doğrusal Regresyon	64
3.13 Karışıklık Matrisi	70
3.14 Roc Eğrisi	71
3.15 Naive Bayes	76
3.16 Naive Bayes Karışıklık Matrisi	76
3.17 Naive Bayes Gözlemleri	78
3.18 Calinski Harabasz	79
3.19 Davies Bouldin	79
3.20 Silhouette	80
3.21 Kmeans	81
3.22 Kmeans	82
3.23 Kmeans	83
3.24 Kmeans Karışıklık Matrisi	84
4.1 Akış Şeması, eğitim süreci adımları	88
4.2 GPR grafiği	90
4.3 GPR-SMOTE grafiği	91
4.4 eksGPR grafiği	92
4.5 eksGPR-SMOTE grafiği	93

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
4.2 GPR Performans Sonuçları	89
4.3 GPR-SMOTE Performans Sonuçları	91
4.4 eksGPR Performans Sonuçları	92
4.5 eksGPR-SMOTE Performans Sonuçları	93



SİMGELER VE KISALTMALAR DİZİNİ

Simgeler

X

Y

β_0

β_1

B

E

ϵ_i

$m(x)$

$K(x, x')$

β

σ^2

SE

N, n

P

Kısaltmalar

ANN

AdjRsqr

BTC

BB

CCI

ETH

EFI

EMA

FN

FP

GPR, GP

GRU

Kfold

Açıklama

Giriş (Bağımlı) değişkeni

Çıkış (Bağımsız) değişkeni

Doğrunun y eksenini kestiği nokta

Eğim

Katsayılar vektörü

Rastgele hata vektörü

Hata terimlerinin toplamı

ortalama

Kovaryans fonksiyonu

Çekirdek işlevinin temel işlem katsayıları

Gürültü varyansı

Tahminin standart hatası

Örneklem

Bağımsız değişkenlerin sayısı

Açıklama

Yapay Sinir Ağı

Düzeltilmiş R^2

Bitcoin

Bollinger Band

Emtia Kanal Endeksi

Ethereum

Elders Forse Index

Exponential Moving Avarage

False Negative

False Positive

Gaussian Process Regression

Gatet Recurrent Units

Katlamalı Çapraz Doğrulama

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

KNN

LSTM

MACD

MA

MSE

MAPE

MAE

MFI

NCA

PCA

RSI

RMSE

R

 R^2

ROC

SVM

SMOTE

SH

TP

TN

USD

WMA

XRP

Açıklama

K-Nearest Neighbors

Uzun Kısa Süreli Bellek

Moving Average Convergence Divergence

Hareketli Ortalama

Ortalama Karesel Hata

Ortalama Mutlak Yüzde Hata

Ortalama Mutlak Hata

Para akışı endeksi göstergesi

Komşuluk bileşeni analizi

Temel bileşen analizi

Göreceli Güç Endeksi

Kök ortalama karesel hatası

Korelasyon analizi

Açıklayıcılık korelasyon analizi

Değişim hız göstergesi

Destek vektör makinesi

Synthetic Minority Over-Sampling Technique

Standart hata

True positive

True Negative

Amerikan doları

Ağırlıklı hareketli ortalama

Ripple

1. GİRİŞ VE AMAÇ

2009'da ortaya çıkan ve dünya çapında popüler hale gelen dijital para birimleri, Türkiye'deki birçok yatırımcı tarafından ilgi gördü. Ancak kripto paraların Türkiye'de bilinirlik kazanması ve yaygınlaşması, 2017'de ivme kazandı. Bitcoin, 2013'te birim fiyatı yaklaşık 100 ABD Doları iken, Kasım 2017'de 10.000 ABD Dolarına çıkmış ve aynı yıl Aralık ayında 20.000 ABD Dolarını geçmiştir. Kripto para sektörünün bu hızlı yükselişi, birçok yatırımcının ilgisini çektiği gibi, genel kamuoyunda da büyük bir merak uyandırdı. Bu durum, Türkiye'de kripto paraların tanınabilirliğini önemli ölçüde artırmıştır.

Kovid-19 salgınının ekonomik etkilerini en aza indirmek adına, merkez bankaları faiz oranlarını düşürerek 2020'den 2021'e kadar kripto para piyasalarında rekor seviyede fon girişleri sağlamıştır. Faiz oranlarının düşürülmesi küresel çapta enflasyona yol açarken, ABD Merkez Bankası'nın 2021 sonunda daha sıkı bir duruş sergilemesi, tüm piyasalarda risk iştahının azalmasına ve dolara olan talebin artmasına neden olmuştur. 2022'de başlayan Rusya-Ukrayna savaşı, tedarik zinciri sorunlarına ve karşılıklı uygulanan yaptırımlara bağlı olarak üretim ve enerji maliyetlerini artırmış, küresel piyasalarda risk algısını yükseltmiştir. Yıl sonlarına doğru, FTX gibi kripto para piyasasının önde gelen borsalarından birinin iflası, sektörü olumsuz etkilemiştir. Bitcoin, 2022'ye 46,217 dolardan başlamış, ancak tüm bu gelişmelerle birlikte 15,588 dolara kadar gerileyerek Kasım 2020'den bu yana en düşük seviyeyi görmüştür. Yıl sonunda ise yüzde 64'lük bir değer kaybıyla 16,542 dolardan kapanmıştır.

Pek çok kripto para birimi vardır: Bitcoin (BTC), Ethereum, Binance Coin, Tether. En yüksek hacime sahip olan BTC 'dir. Bu nedenle, bu çalışmada BTC kullanılmıştır. Makine öğrenmesi, matematiksel ve istatistiksel teknikleri kullanarak mevcut verilerden öğrenme kapasitesine sahip olan, bu öğrenmeleri kullanarak bilinmeyen durumlar hakkında tahminlerde bulunan ve bu süreçte modelleme ve algoritmaları kullanan yapay zekanın bir alt dalıdır (Yergök ve Acı, 2019). Bu nedenle, BTC fiyatını tahmin etmek için makine öğrenmesi teknikleri uygulanmıştır.

Bu tez çalışmasının amacı, BTC'nin günlük verileri baz alınarak çeşitli yöntemlerle tahmin edilebilen bir makine öğrenme modeli geliştirmektir. Bu

amaçla, Gaussian Process Regression (GPR) yöntemini kullanırken kfold ve smote ile çeşitli varyasyonlar test edilmiştir. Bu yöntemler MATLAB ortamında yapılmıştır.

Bu tezin geri kalanı şu şekilde yapılandırılmıştır. 2. Bölüm, bu tezle ilgili olarak, ilk olarak kriptoloji, kripto para, blockchain, kripto paraları etkileyen faktörler, bitcoin, teknik analiz ve indikatörleri konularına genel bir bakışla giriş bilgileri sağlar. İkinci kısımda, makine öğrenmesi ve türleri, veri ön işleme konularına giriş yapılarak ilgili algoritmalar açıklanmıştır. Son olarak, MATLAB ortamında yapılan farklı örneklerle kavramlar pekiştirilmiştir.



2. LİTERATÜR ARAŞTIRMASI

Bu bölüm, kripto para, makine öğrenimi ve kripto para tahmininde makine öğreniminin kullanımına ilişkin çalışmalar ile konuyla ilgili ön bilgileri açıklamaktadır.

2.1 Kripto Para

Kripto para, güvenlik açısından kriptoloji bilimini kullanan, dijital ve sanal bir para birimidir ve matematik temelli şifrelenmişlerdir. Banka kartları, sanal kartlar veya sanal ortamda yapılan her işlemde bu paraları kullanılır. 2009 yılı itibariyle hayatımıza giren kripto para olgusu, temelde "blockchain" teknolojisine dayanmaktadır - ki bu ifade Türkçe'de blokzincir olarak adlandırılır. Kripto paralarla ilgili tüm işlemler, bu yenilikçi teknolojinin sağladığı blokzincir sistemine dayalı olarak gerçekleşmektedir. Kripto paraların hızla küresel bir çıkış açmasındaki temel etkenler, merkezi bir otoriteye bağlı olmamaları ve anonim olma özellikleri olarak öne çıkmaktadır. Bu özellikler, kripto paraların dünya genelinde yaygın bir kabul görmesine ve birçok kişi tarafından alternatif bir para birimi olarak benimsenmesine zemin hazırlamıştır. Gün geçtikçe, kripto paraların popülerliği artmakta ve dijital varlıkların çeşitli kullanım alanları genişlemektedir.

Kriptoloji, bilgilerin (metin, rakam veya herhangi bir mesaj) belirli bir algoritma tarafından şifrelenmesini, güvenli bir kanal üzerinden alıcıya iletilmesini ve ardından şifreleme sisteminin çözülmesi yoluyla verilerin orijinal haline dönmesini içeren bir şifreleme bilimidir. Bu disiplin, bilgi güvenliği, iletişim gizliliği ve veri bütünlüğü gibi konulara odaklanarak bilgilerin korunmasını amaçlar. Kriptoloji, modern iletişim sistemlerinin temelini oluştururken, siber güvenlik alanında da önemli bir rol oynamaktadır.

2.1.1 Blockchain

Blokzincir, internet üzerinde toplanan bilgilerin veya verilerin açık bir kayıt defteri olarak nitelendirilebilir. Bu sistem, temelde şifreli bir verinin zincir halkalarıyla birbirine bağlanması ve değiştirilemez bloklar halinde birleştirilmesine dayanmaktadır. Örneğin, bir kitaplıkta bulunan her kitap bir bloğu temsil eder. Her blok, kitaplıkta bulunan diğer bloklara bağlı olarak bir önceki bloğun içeriğini referans alır. Bu bağlantılar, bir zincir oluşturur, bu yüzden

bu teknolojiye Blokzincir denir. Bir finansal işlemi gözönüne alırsak blokzincirde bir blok oluşturur. Bu blok, işlemin gerçekleştiği tarih, zaman, gönderenin kimliği, alıcının kimliği ve işlem miktarını içerebilir. Her yeni işlem, yeni bir blok oluşturur ve bu bloklar birbirine bağlanır. Blokzincir üzerine eklenen verilerin değiştirilmesi veya silinmesi neredeyse imkânsızdır; çünkü bir zincirdeki veride değişiklik yapabilmek için söz konusu zincirin en başındaki bloğa kadar gidilmesi gerekmektedir, ki bu pratikte imkânsızdır. Bu özelliği sayesinde blokzincir, verilerin güvenli bir şekilde depolanmasını sağlayan bir sistemdir.

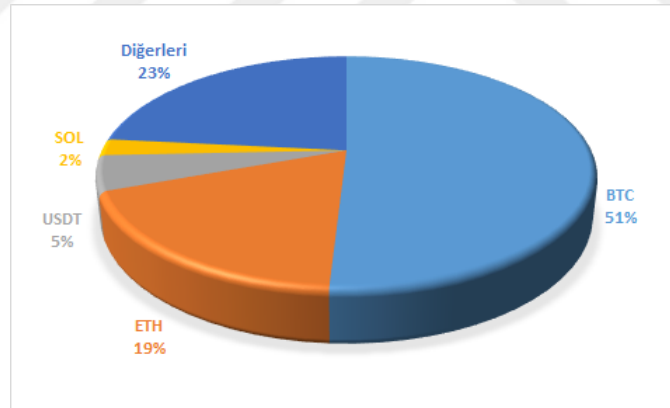
Blokzincir teknolojisinin en önemli özelliği, merkezi olmayan bir doğrulama sistemine sahip olmasıdır. Bu özellik, blokzincirin dijital dönüşüm süreçlerinde etkili bir şekilde kullanılmasına olanak tanır. Blokzincir, gayrimenkul, araç ve değerli varlıkların kanıt ve kaydından, doğum, evlilik ve ölüm belgelerinin tutulmasına, seçimlerin gerçekleştirilmesinden akıllı kontratların yönetilmesine kadar birçok farklı alanda uygulanabilen değerli bir teknoloji olarak görülmektedir. Aynı zamanda blokzincir teknolojisi, bireysel kullanıcılara benzeri görülmemiş bir kontrol imkânı sunarak dijital kimlik yönetiminde önemli bir rol oynamaktadır. Küresel bir açık hesap defteri olan blokzincir, sadece kripto paraların üretiminde değil, aynı zamanda saklama, yönetme ve depolama gibi birçok işlem için kullanılmaktadır.

Blokzinciri, dağıtılmış hesaplama, olasılık ve istatistik gibi çeşitli matematiksel kavramların entegre olduğu bir teknolojidir. Ağın genel güvenliğinin sağlanması gibi işlevler, matematiksel modeller ve algoritmalar kullanılarak gerçekleştirilir. Blokzincirinde hash fonksiyonları, verilerin benzersiz bir şekilde tanımlanmasını sağlamak için kullanılır. Hash fonksiyonları, belirli bir veri setini alır ve sabit uzunlukta bir çıktı üretir. Genellikle benzersiz olan bu hash değeri her blokta bulunur ve bir önceki bloğun hash değeri ile birlikte hash fonksiyonundan geçirilir. Böylece bir bloğun içinde herhangi bir değişiklik yapıldığında, o bloğun hash değeri değişir. Bu, bir bloğun içeriğindeki değişikliklerin tüm blok zincirindeki blokları etkileyeceği anlamına gelir. Dağıtılmış defter teknolojisi, bir ağın birden çok noktasında bulunan bir veritabanı sistemidir ve matematiksel olarak karmaşık algoritmaları içerir. Örneğin, blokzinciri işlemlerinin doğrulanması, blokların madenciler tarafından oluşturulması ve ağın genel güvenliğinin sağlanması gibi işlevler, matematiksel modeller ve algoritmalar kullanılarak gerçekleştirilir. Bir bloktaki işlemin doğrulanması için gereken zamanın istatistiksel dağılımı gibi konular blokzincirin performansını ve güvenilirliğini artırır.

Kripto para ve blokzincir teknolojisi, hemen hemen her sektörde geniş bir kullanım potansiyeline sahiptir. Özellikle Bitcoin, Ethereum gibi birçok kripto para biriminin geliştirilmesine olanak tanıyan blokzincir teknolojisi, finansal sistemin ötesinde birçok alanda etkili olmuştur.

2.1.2 Bitcoin

Bitcoin, 2008’de bilinmeyen bir kişi veya grup olan Satoshi Nakamoto tarafından geliştirilen bir kripto para birimidir. İlk defa 2009’da açık kaynaklı bir yazılım olarak piyasaya sürüldü ve o tarihten bu yana aktif olarak kullanılmaktadır. Bitcoin, merkezi olmayan bir dijital para birimidir ve herhangi bir merkez bankasına veya tek bir yöneticiye bağlı değildir. Aracılara ihtiyaç duyulmadan, kullanıcıdan kullanıcıya doğrudan transfer edilebilir. Bitcoin, geleneksel finansal sistemin dışında, blokzincir teknolojisi üzerine inşa edilmiş bir kripto paradır. 2009’da ilk kez ortaya çıkan BTC, kripto paraların öncülerinden biri olarak, yeni bir dijital para çağının kapılarını aralamıştır. Bitcoin, dünya genelinde 40’tan fazla borsada işlem görmekte olup, en değerli kripto para birimi unvanını taşımaktadır (McNally vd. 2018)



Şekil 2.1 BTC ve Diğer Para Birimleri Hacim Grafiği

Bitcoin ve diğer kripto para birimleri hacmi, piyasa değeri ve dominansı, kripto para piyasasının genel sağlığını ve yatırımcı ilgisini ölçmek için kullanılan temel metriklerdir. Bitcoin hacmi, belirli bir zaman diliminde gerçekleşen tüm Bitcoin işlemlerinin toplam değerini ifade eder ve yüksek hacim, piyasadaki likiditenin güçlü olduğunu gösterebilir. Kripto para birimi hacmi ise tüm kripto paraların toplam işlem değerini kapsar, bu da genel kripto piyasasının büyüklüğünü ve aktivitesini ölçer. Piyasa değeri, bir kripto paranın dolaşımdaki arzının mevcut fiyatına göre hesaplanan toplam değerini ifade eder. Bitcoin

dominansı ise tüm kripto paraların piyasa değeri içindeki Bitcoin'in payını gösterir. **Şekil 2.1** de de verilen grafik 25.02.2024 tarihindeki coinmarketcap sitesinden elde edilen dominans verileriyle görselleştirdiğimiz BTC ve diğer kripto para birimlerinin yüzdelik grafiğini göstermektedir. Bu metrikler, yatırımcılar ve analistler için kripto para piyasasının genel durumu, yatırımcı ilgisi ve olası fiyat hareketlerini değerlendirmek için önemli birer gösterge olarak kullanılır.

2.2 Kripto paraları etkileyen faktörler

Bitcoin, değerini tek bir merkez bankası ya da kurum belirlemez. Bitcoin'in değeri, tamamen arz ve talep faktörleriyle, yani insanların ödemeye razı olduğu fiyatlarla şekillenir. Bu dijital varlığın fiyatı, piyasada talebin arttığı durumlarda yükselirken, talebin düştüğü zamanlarda ise düşüş gösterir.

Bitcoin'in maksimum arzı, belirlenmiş bir üst sınır olan 21 milyon BTC ile sınırlıdır. Bu durum, Bitcoin'in benzersiz bir kısıtlama mekanizması sunarak değerini korumasına ve madencilik süreciyle sürdürülebilir bir şekilde üretilmesine olanak tanır.

Talep faktörü, küresel ekonomik gelişmelerden hisse senedi ve tahvil piyasalarındaki hareketliliğe kadar geniş bir yelpazede değişkenlik gösterir. Ayrıca, küresel ticaret savaşları ve diğer uluslararası olaylar da Bitcoin'in fiyatını etkileyebilecek önemli faktörler arasında yer alır. Bu nedenle, Bitcoin yatırımcıları ve uzmanları, dijital varlığın fiyatındaki değişimleri anlamak ve tahmin etmek için geniş bir ekonomik bağlamı dikkate almak zorundadır.

Kripto para piyasasını etkileyen bir dizi faktör bulunmaktadır. Bu faktörler arasında üretim maliyeti, rekabetçi kripto paraların sayısı, düzenlemeler, medya ve haberler önemli rol oynar. Her biri piyasada dalgalanmalara neden olabilir ve kripto para birimlerinin değerini etkileyebilir.

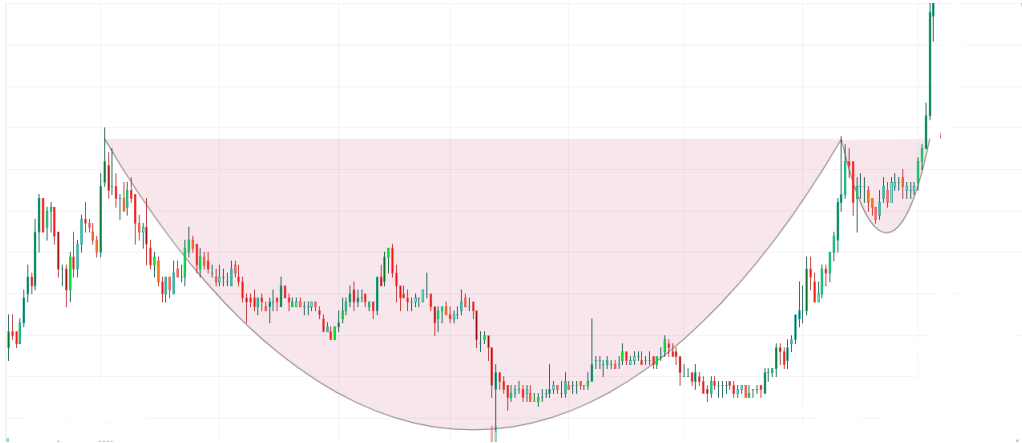
Şahin, 2020 çalışmasında BTC fiyatını etkileyen faktörler olarak altın, ABD doları ve finansal baskı endeksi ve jeopolitik risk endeksi etkileri incelenmiştir. Çok Değişkenli Adaptif Regresyon Uzanımları (Multivariate Adaptive Regression Splines; MARS) metodu olan bir regresyon yöntemi kullanmıştır. Analiz sonuçlarına göre, Bitcoin fiyatının ABD Doları'ndaki değişikliklere oldukça hassas bir şekilde tepki verdiği gözlemlenmiştir. Özellikle Altın fiyatlarının belirli bir seviyeyi aştığı durumlarda, Bitcoin fiyatının üzerinde etkili olabildiği tespit

edilmiştir. Bu durum, Altın fiyatlarının yükseldiği zamanlarda küresel risklerin ve finansal baskının arttığı dönemlere işaret etmektedir. Finansal baskı endeksi ile Bitcoin fiyatı arasındaki ilişki incelendiğinde ise, finansal baskının yükseldiği zamanlarda Bitcoin fiyatının belirlenmesinde potansiyel bir belirleyici faktör olabileceği sonucuna ulaşılmıştır.

Genel anlamda, Bitcoin fiyatının temelinde arz ve talep faktörlerinin etkili olduğu bilinse de, bu faktörleri etkileyen unsurların detaylı bir şekilde incelenmesinin kritik olduğu ortaya çıkmaktadır. Özellikle haber manipülasyonları ve bu haberlerin kaynağı olan bölgesel çatışmalar, terör eylemleri, ticaret savaşları ve dijital devrim gibi faktörlerin, Bitcoin ve diğer kripto para fiyatlarını dolaylı bir şekilde etkilediği sonucuna varılmıştır. Bu bağlamda, kripto para piyasalarındaki dinamikleri anlamak ve fiyat hareketlerini öngörebilmek için geniş bir perspektife ihtiyaç duyulduğu vurgulanmaktadır.

2.2.1 Teknik analiz ve indikatörler

Bitcoin fiyat grafiğinde, genellikle mum grafikleri veya çizgi grafikleri kullanılır. Mum grafikleri, belirli bir zaman diliminde açılış, kapanış, en yüksek ve en düşük fiyatları gösteren bir dizi mumdan oluşur. Yeşil (boğa) mumlar genellikle fiyatın arttığını, kırmızı (ayı) mumlar ise fiyatın düştüğünü gösterir. Bitcoin grafiğini okurken, destek ve direnç seviyeleri belirlemek önemlidir. Destek seviyeleri, fiyatın düşerken ancak aşağı yönlü bir hareketi durdurduğu veya yavaşlattığı seviyelerdir. Direnç seviyeleri ise fiyatın yükselirken ancak yukarı yönlü bir hareketi durdurduğu veya yavaşlattığı seviyelerdir. Bu seviyeler, geçmiş fiyat hareketleri ve grafik analizleri kullanılarak belirlenebilir.



Şekil 2.2 Fincan Kulp Formasyonu

Fincan ve kulp formasyonu, bir düşüş trendinin sona erdiğini ve yükseliş trendinin başlayabileceğini gösteren bir grafik modelidir. Fincan, bir varlığın değerinin düştüğü ve ardından bir fincan şeklinde yükseldiği bir süreci temsil eder. Kulp ise fincanın oluşumunu tamamladıktan sonra bir miktar geri çekilme gösteren kısa bir düşüşü temsil eder. Bu formasyon genellikle yükselişin başlangıcını işaret edebilir. Ters fincan ise tam tersidir. Yani, bir varlığın değeri yükseldikten sonra bir fincan şeklinde düştüğü ve ardından bir kulp olduğu bir süreci temsil eder. Bu formasyon, düşüş trendinin sona erdiğini ve yükseliş trendinin başlayabileceğini gösterebilir. Bu grafik analiz araçlarına ek olarak, teknik göstergeler de kullanılır.

Teknik analiz, fiyat, hacim gibi piyasa göstergelerinden yola çıkarak geçmiş dönem piyasa hareketlerinin analiz edilmesi ve bu analiz sonuçlarına dayanarak gelecekteki fiyat hareketlerinin tahmin edilmesi amacını taşır. Bu yöntemde, çeşitli grafik oluşumları ve istatistiksel yöntemler kullanılarak piyasa trendleri ve olası fiyat değişiklikleri analiz edilir.

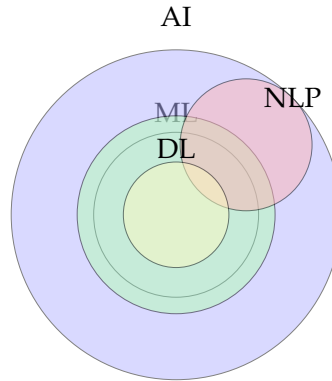
Teknik analizin temel prensibi, geçmişte meydana gelen fiyat hareketlerinin, gelecekte de benzer koşullar altında benzer şekilde tekrarlanabileceği varsayımına dayanır. Grafikler üzerinde oluşan çeşitli desenler ve formasyonlar, yatırımcılara piyasanın gelecekteki potansiyel hareketleri hakkında fikir verme amacı taşır. Bu desenlerin ve formasyonların doğru bir şekilde tanımlanması ve yorumlanmasıyla, yatırımcılar alım-satım kararlarına daha bilinçli bir şekilde yönlenebilirler.

Teknik analizde kullanılan göstergeler arasında hareketli ortalamalar, RSI (Relative Strength Index), MACD (Moving Average Convergence Divergence) gibi çeşitli matematiksel formüllere dayalı göstergeler bulunmaktadır. Bu göstergeler, fiyat hareketleri ve ticaret hacmi gibi verilere dayalı olarak analizler sunarak yatırımcılara piyasanın gücü, trendi ve potansiyel dönüş noktaları hakkında bilgi sağlar.

Teknik analiz, özellikle kısa vadeli yatırımcılar arasında popülerdir ve finansal piyasalarda alım-satım stratejileri oluşturmak için sıkça kullanılır. Ancak, bu analiz yöntemi her zaman kesin sonuçlar vermeyebilir ve yatırımcıların dikkatli bir şekilde diğer faktörleri de göz önünde bulundurması gerekmektedir.

3. MAKİNE ÖĞRENMESİ

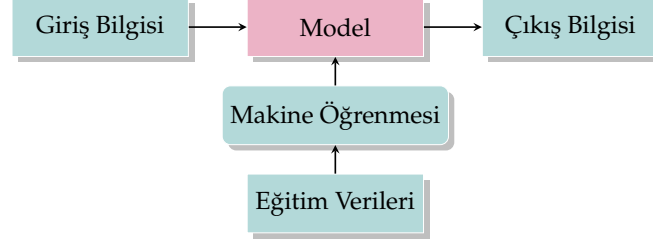
Yapay zeka, bilgisayar sistemlerinin insan benzeri zeka yeteneklerini geliştirmeyi amaçlayan bir alanı ifade eder. Bu geniş kapsamlı disiplin, birkaç önemli alt dalı içerir. Yapay zekanın bir alt dalı olan makine öğrenmesi, verileri analiz ederek kendi kendine öğrenme yeteneği geliştirerek bilgisayar sistemlerinin deneyimlerle gelişmesine odaklanır. Derin Öğrenme (DL), yapay sinir ağları üzerine odaklanarak karmaşık görevleri gerçekleştirebilen bir ML dalıdır. Yapay zekâ, bilgisayarların bazı görevleri gerçekleştirebilmesi için akıllı denilen yöntemleri kullanabilmesini ifade eder ve kapsam olarak makine öğrenmesinden daha geniş bir alana sahiptir (Gultepe, 2019). Doğal Dil İşleme (NLP), bilgisayarların dil anlama, yorumlama ve üretme yeteneğini içerir, bu da metin madde analizi, dil çevirisi ve konuşma tanıma gibi görevleri kapsar. Büyük Dil Modelleri (LLM), geniş veri setleri üzerinde eğitilen, dilin kapsamlı bir şekilde anlaşılması ve üretilmesini hedefleyen büyük dil modelleridir. Bu modeller, dilin öğrenilmesi ve anlaşılmasında derin öğrenme tekniklerini kullanır. Bu alt dallar, yapay zeka teknolojilerinin gelişimi ve uygulanabilirliği açısından önemli ilerlemelere öncülük ederek birçok sektörde çeşitli uygulamalara olanak tanımaktadır.



Şekil 3.1 Yapay Zeka

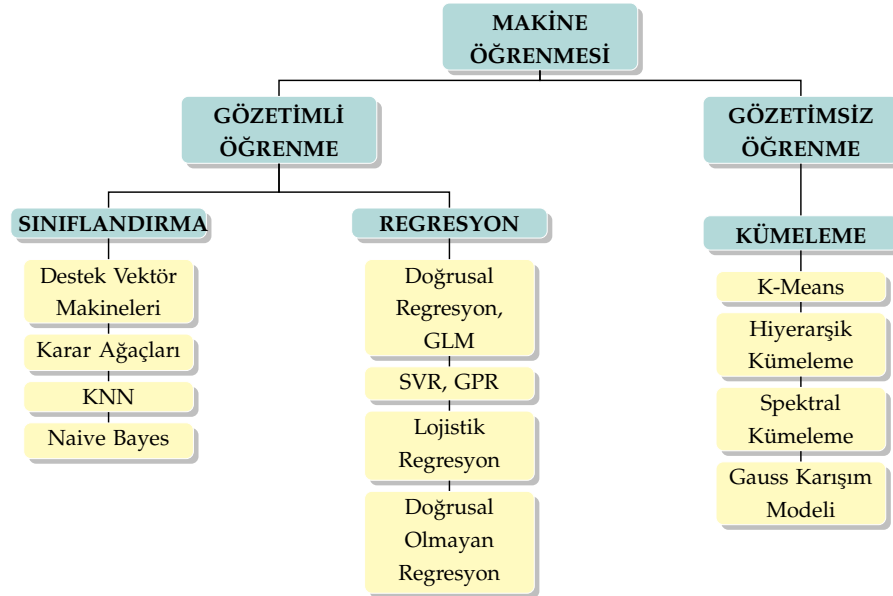
Makine öğrenmesi, bilgisayarların belirli bir programlama ile değil, elde edilen veriler üzerinden öğrenerek bir model oluşturur. Yani, bu yaklaşım, makinelerin deneyimlerden öğrenmesine ve bu öğrenmeyi kullanarak gelecekteki görevleri daha etkili bir şekilde yerine getirmesine odaklanır. Bu süreç, verilerden desenleri çıkarma, kararlar alma yeteneği gibi öğrenme faaliyetlerini içerir.

Makine öğrenmesi ve yapay zekâ, günümüzde birçok sektörde kullanılan ve hızla gelişen teknolojilerdir. Bu alanlardaki ilerlemeler, bilgisayar sistemlerinin daha karmaşık görevleri yerine getirebilme yeteneklerini artırmaktadır.



Şekil 3.2 Makine Öğrenmesi

Makine öğrenmesi, bir modelleme tekniği olarak bilinir ve genellikle belge, ses, görüntü gibi çeşitli bilgi türlerini içeren verilerle çalışır. Model, makine öğreniminin sonuç ürünüdür ve öğrenilen bilgilerin temsilini sağlar. Geleneksel analitik modellerin yetersiz kaldığı durumlarda sorunları çözmek için geliştirilen makine öğrenimi, genellikle matematiksel denklemler ve genel yasalara uymayan durumlarda eğitim verilerini kullanarak bir model elde etme tekniklerini içerir. Bu sayede, belirli bir görevi yerine getirmek üzere verilerden öğrenilen bilgileri içeren bir model oluşturulabilir.



Şekil 3.3 Makine Öğrenmesi

Makine öğrenmesi algoritmaları genellikle dört ana kategoriye ayrılır, her biri farklı bir amaç taşır. *Gözetimli öğrenme*, verilerin kapsamını ölçeklendirmek ve buna dayalı tahminlerde bulunmak için kullanılırken, *Gözetimsiz algoritmalar* verileri anlamlı bir şekilde düzenlemek ve filtrelemek amacıyla kullanılır.

3.1 Gözetimli Öğrenme

Gözetimli öğrenme, yapay zeka alanında sıkça kullanılan bir öğrenme yöntemidir ve genellikle bir eğitim dataseti içerir. Bu datasetinde her bir örnek için giriş (X) ve çıkış (Y) değerleri bulunur. Bu değerler arasında bir ilişki olduğu varsayılarak, sisteme öğrenilmek istenen kavramla ilgili elde edilen gözlemler sunulur. Gözetimli öğrenme yönetiminde, sistemin öğrenmesi için bu eğitim kümesi kullanılır.

Eğitim kümesinde her bir örnek için istenen çıkış değerleri verildiğinden, sistemin giriş ve çıkış arasında bir bağlantı kurabilmesi için bu verileri kullanır. Bu bağlantı, gelecekte karşılaşılabilecek X girişlerinin karşılık geldiği Y değerlerini tahmin etmek için kullanılabilir.

Gözetimli öğrenme, genellikle *regresyon* ve *sınıflandırma* olmak üzere iki ana kategoriye ayrılır. Regresyonda, giriş değerleri belirli bir sürekli fonksiyona uygulanarak çıkış değerleri tahmin edilmeye çalışılır. Sınıflandırmada ise giriş değerleri belirli kategorilere haritalanarak her bir gözleme bir sınıf atanır. Bu şekilde, gözetimli öğrenme çeşitli uygulama alanlarında kullanılarak gelecekteki bilinmeyen verilere dair tahminlerde bulunabilir.

3.1.1 Regresyon

Regresyon, iki ya da daha fazla değişken arasındaki doğrusal ilişkinin fonksiyonel şeklini ifade etmekle kalmaz, aynı zamanda bir değişkenin değeri bilindiğinde diğer değişkenin tahmin edilmesine olanak tanır. Bu yöntem, bağımlı değişken ile bağımsız değişken arasındaki ilişkiyi bir doğru denklemi olarak gösterir ve bu denklem aracılığıyla tahminler yapılmasına olanak sağlar.

Regresyon analizi, verilerin x-y koordinat düzleminde dağıldığını düşünerek başlar. Veriler arasındaki uzaklıklar dikkate alınarak, tüm verileri kapsayan veya en uygun şekilde uyan bir doğru çizme amacı güdülür. Bu doğru, regresyon analizinde kullanılan lineer bir modeli temsil eder. Eğitim verilerinin düzlem üzerindeki dağılımını matematiksel bir modele benzetmek ve bu modele

en uygun olan regresyon türünü bulmak, regresyon analizinin temel hedeflerindendir.

Bu süreç, veriler arasındaki ilişkiyi temsil eden bir doğru oluşturmayı amaçlar ve bu doğru, bağımlı değişkenin değerini bağımsız değişkenlerle ilişkilendirerek gelecekteki değerleri tahmin etmek için kullanılır. Lineer regresyon, bu tür bir modelleme için en sık kullanılan yöntemlerden biridir ve iki değişken arasındaki doğrusal ilişkiyi ifade eder.

Doğrusal regresyon (lineer regresyon)

Lineer regresyon, bir veya daha fazla bağımsız değişkenle bağımlı bir değişken arasındaki ilişkiyi modellemek için kullanılan bir istatistiksel yöntemdir. Doğrusal regresyon, bir değişkenin diğerini nasıl etkilediğini ve bu etkileşimin sonucunun olumlu ya da olumsuz olup olmadığını değerlendirmek için kullanılır. Bu yöntem, değişkenler arasındaki ilişkiyi açıklamak ve gelecekteki değerleri tahmin etme amacını taşır. Matematiksel bir model aracılığıyla değişkenler arasındaki ilişkiyi anlamaya çalışır, böylece bu ilişkinin altında yatan dinamikleri kavramamıza yardımcı olur. Bu bağımlı(y) ve bağımsız(x) değişkenler arasında doğrusal ilişkiyi doğrusal regresyon belirler. Doğrusal Regresyon,

$$y = \beta_0 + \beta_1 x + \epsilon \quad (3.1)$$

şeklinde bir çizgi olarak temsil edilir ve tahmin edilen çıktının sürekli olduğu ve sabit bir eğime sahip olduğu durumlarda uygulanır. ϵ , doğrusal regresyonun hata terimini temsil eder ve x ve y arasındaki değişkenliği açıklamak için kullanılır. β_0 , doğrunun y eksenini kestiği noktayı temsil ederken β_1 eğimi temsil eder (Özen vd. 2021).

Lojistik regresyon

Lojistik regresyon algoritması, çoğunlukla ikili sınıflandırma problemlerinde kullanılan bir yöntemdir, burada genellikle iki değerden biri sonuca ulaşır. Örneğin, geçti veya kaldı, doğru veya yanlış gibi durumlar, bu tür problemleri kapsar. Bu algoritma, bağımlı değişkenin, iki kategoriden birine girmesi olasılığını tahmin etme ihtiyacı için özellikle uygun bir seçenektir (Sevli, 2019). Bu model, lojistik fonksiyonu (sigmoid fonksiyonu) kullanarak veri seti

üzerinde bir olasılık dağılımı tahmin eder. Lojistik regresyonun matematiksel formülü aşağıdaki gibi tanımlanır.

$$P(Y = 1) = \frac{1}{1 + e^{-(b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n)}} \quad (3.2)$$

Burada $P(Y=1)$, bağımlı değişkenin 1 olma olasılığını temsil eder. e , Euler sayısı; b_0 , doğrusal regresyon sabiti; $b_1, b_2, \dots, b_n$ modelin katsayıları ve $x_1, x_2, \dots, x_n$ bağımsız değişkenlerdir. Model, bağımsız değişkenlerin ağırlıklı toplamını sigmoid fonksiyonuna sokarak 0 ile 1 arasında bir olasılık değeri üretir. Bu olasılık değeri, belirli bir gözlemin 1 olma olasılığını ifade eder. Genellikle bir eşik değeri belirlenir (örneğin, 0.5), ve bu eşik değeri üzerindeki olasılıklar 1 olarak sınıflandırılırken, altındakiler 0 olarak sınıflandırılır.

Bu algoritma, özellikle tıp, finans, pazarlama gibi birçok alanda, iki kategorili sınıflandırma problemleri üzerinde başarıyla uygulanabilir.

Çoklu doğrusal (multilinear) regresyon

Çoklu regresyon analizi, bir bağımlı değişkenin değerini, bir dizi bağımsız değişkenin değerlerine dayanarak tahmin etme amacını taşır. Bu analiz, bağımsız değişkenlerin bağımlı değişken üzerindeki etkilerini nicel olarak ölçebilir ve bu etkilerin doğrusal bir model içinde nasıl birleştiğini inceleyebilir. Bu, özellikle karmaşık ilişkileri anlamak ve gelecekteki değerleri tahmin etmek için kullanışlıdır. Bu modelin matematiksel formülü aşağıdaki gibi tanımlanır.

$$Y = XB + E \quad (3.3)$$

- Y bağımlı değişken gözlem vektörü
- X bağımsız değişkenler gözlem matris
- B katsayılar vektörü
- E rasgele hata vektörü

Verilere çoklu regresyon uygulanabilmesi için, bağımsız değişkenler arasında çoklu bağımlılık (multicollinearity) olmaması gerekir (Kayaalp vd. n.d.).

Doğrusal olmayan regresyon(nonlinear)

Doğrusal olmayan regresyon, istatistik biliminde gözlemlenen verilerin bir veya birden fazla bağımsız değişkenin doğrusal olmayan bir bileşiği olarak tanımlanır. Bu metod, doğrusal regresyonun sınırlamalarını aşmak ve daha karmaşık ilişkileri modellemek için kullanılır. Bu bağlamda, regresyon modeli, doğrusal olmayan bir fonksiyonla ifade edilerek, bağımlı ve bağımsız değişkenler arasındaki ilişkiyi anlamaya yönelik daha esnek bir yaklaşım sunar.

Doğrusal olmayan regresyon modellerinin formu genelde doğrusal regresyon modelleri gibidir.

$$Y_i = f(X_i, \gamma) + \varepsilon_i \quad (3.4)$$

denklemini doğrusal olmayan regresyon modelinin genel formunu gösterir. Burada Y_i gözlemi, doğrusal olmayan $f(X_i, \gamma)$ fonksiyonu ve ε_i hata terimlerinin toplamıdır. Hata terimlerinin doğrusal regresyon modelindeki gibi, beklenen değerlerinin sıfır, sabit varyanslı ve korelasyonsuz olduğu varsayılır. $f(X_i, \gamma)$ fonksiyonu ise, Y bağımlı değişkeninin, X bağımsız değişkenlerinin bir fonksiyonu olduğunu gösterir (Altunkaynak ve Esin, 2004).

Doğrusal olmayan regresyon modelinin, doğrusal regresyon modelinden en önemli ayrımı; regresyon parametre sayılarının modeldeki bağımsız X değişkenlerinin sayısı ile doğrudan bağlantılı olmamasıdır. Geleneksel doğrusal regresyon modellerinde, modelde $p - 1$ bağımsız X değişkeni bulunduğunda, buna bağlı olarak “ p ” adet regresyon katsayısı bulunurdu. Ancak doğrusal olmayan regresyon modellerinde bu ilişki geçerli değildir. Bu durumda, doğrusal olmayan regresyon modellerinde X bağımsız değişkenlerinin sayısı q olarak ifade edilirken, fonksiyondaki regresyon parametreleri sayısı p olarak adlandırılır. Bu esnek yapı, modelin karmaşıklığını ve bağımsız değişkenlerin etkilerini daha geniş bir perspektiften anlamamıza olanak tanır.

Ridge regresyon

Regresyon modeli karmaşık hale geldiğinde, ridge regresyonu; modelin katsayılarının boyutunu düzeltmeye yardımcı olan bir yöntemdir. Ridge regresyonu, özellikle çoklu bağımsız değişkenlerin olduğu durumlarda aşırı

uydurmayı (overfitting) önlemeye ve modelin daha genelleştirilebilir olmasını sağlamaya yönelik olarak kullanılır. Ridge regresyonu, katsayıları sıfıra yaklaştırarak ve böylece modelin ağırlıklarını dengeleyerek, modelin aşırı uydurmayı kontrol etmeye çalışır. Bu sayede, modelin gerçek dünya verilerine daha iyi uymasını ve daha güvenilir tahminler yapmasını sağlar.

Lasso regresyon

Lasso regresyon, değişkenleri seçmek ve düzenlemek için kullanılan bir regresyon türüdür. Bu yöntem, özellikle yüksek boyutlu veri setlerinde aşırı uydurmayı önlemek ve modelin daha basit ve anlamlı olmasını sağlamak amacıyla kullanılır. Lasso regresyonu, katsayıları sıfıra indirerek gereksiz veya etkisiz olan değişkenleri modelden çıkarmaya odaklanır. Bu sayede, modelin daha az karmaşık olması ve daha iyi genelleme yapabilmesi hedeflenir. Lasso regresyonu, ridge regresyonu gibi aşırı uydurmayı kontrol etmeye yönelik bir regresyon yöntemidir ancak lasso, katsayıları tamamen sıfıra çekebilme özelliğiyle farklılık gösterir.

Gaussian process regression

Gaussian süreç modeli, çoğunlukla olasılıksal ve parametrik olmayan doğrusal olmayan regresyon problemlerini çözmek için tercih edilen bir başarılı makine öğrenme yaklaşımıdır (Aci ve Dogansoy, 2022; K. Liu vd. 2019). Gaussian süreç regresyonu (GPR), bir Gaussian süreç modelini temsil eder ve bu model, belirli bir dağılımın sınırlı sayıda rastgele değişkenin toplamını içerir. GPR, parametrik olmayan bir regresyon modeli olarak öne çıkar ve özellikle doğrusal olmayan regresyon problemlerini çözmek için etkili bir araç olarak kullanılır. Bu model, belirli bir noktadaki beklenen değeri ve tahmini belirsizliği veriye dayalı olarak sağlar, bu da onu çeşitli uygulamalarda kullanışlı hale getirir (K. Liu vd. 2019). Gaussian süreçleri, istatistiksel modelleme, çoklu hedefli regresyon ve yüksek boyutlu haritalama analizi için kullanılan önemli bir araçtır (N. Zhang vd. 2018).

GPR, sınırlı veri setlerinde dahi etkili sonuçlar üretebilir ve tahminlerle birlikte belirsizlik ölçümlerini sağlama yeteneğine sahiptir. Farklı kovaryans fonksiyonlarının kullanılması, GPR ile en doğru seçeneğin belirlenmesine olanak tanır (Heo ve Zavala, 2012). Gaussian süreç (GP), prior fonksiyonlarının Gauss dağılımında ifade edilmesi ilkesine dayanan bir modelleme yöntemidir. Bu yöntem, sürekli bir giriş uzayında belirli bir şekilde dağılmış olan fonksiyonları tanımlamak için kullanılır. Gaussian süreç, veri etiketlerini içeren bir eğitim seti ile

eğitildiğinde, test vektörlerinin etiketlerini tahmin etmek üzere kullanılabilir. Bu süreç, tahminlerini yaparken sistematik ve rassal varyasyonları içerebilir. Yani, her tahmin bir belirsizlikle birlikte gelir ve bu belirsizlik, test verilerinin sürekli bir fonksiyonun farklı bölgelerine düşme olasılıklarını yansıtır. Gaussian süreçler, özellikle regresyon ve sınıflandırma problemlerinde kullanılan başarılı bir modelleme yaklaşımıdır. GP fonksiyonlar üzerine olasılık dağılımını tanımlar ve aşağıda verildiği gibi ifade edilebilmektedir.

$$f(x) \sim GP(m(x), K(x, x')) \quad (3.5)$$

$m(x)$ ortalama, $K(x, x')$ ise kovaryans fonksiyonu olup aşağıdaki gibi ifade edilmektedir.

$$m(x) = E[f(x)] \quad (3.6)$$

$$K(x, x') = E[(f(x) - m(x))(f(x') - m(x'))^T] \quad (3.7)$$

Kovaryans fonksiyonunda genellikle tercih edilen SE , aşağıda verilmiştir.

$$K_{SE}(x, x') = \theta_f^2 \exp \left(-\frac{1}{\theta_f^2} \|x - x'\|^2 \right) \quad (3.8)$$

Kovaryans fonksiyonunda belirtilen θ_f ve θ_l sırası ile x-ölçekleme (genlik) ve y-ölçekleme (uzunluk ölçeği) olarak tanımlanmaktadır.

Kovaryans ve ortalama, değerlendirilecek girdi bileşenlerin X bütün sınırlı kom- binasyonlarını değerlendirilmesini sağlamaktadır. Kovaryans matrisi aşağıda belirtildiği gibi olmaktadır.

$$K := k((x_1, \dots, x_n), (x_1, \dots, x_n)) = \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \dots & k(x_n, x_n) \end{bmatrix} \quad (3.9)$$

Kovaryans fonksiyonu $K(x, x')$ genellikle bir dizi çekirdek parametresi veya hiperparametresi, θ ile parametreleştirilir. $K(x, x')$ θ 'ye bağımlılığı açıkça

göstermek için genellikle $(K(x, x')|\theta)$ olarak yazılır (Eberhard ve Geissbuhler, 2000). Burada GPR fonksiyonunun dört çekirdek fonksiyonu (kernel functions) aşağıdaki gibi incelenmektedir:

$$\begin{aligned}
 \textit{Exponentiel} & \rightarrow k(x_i, x_j|\theta) = \sigma_f^2 \exp\left(-\frac{r}{\sigma_l}\right) \\
 \textit{SquaredExponentiel} & \rightarrow k(x_i, x_j|\theta) = \sigma_f^2 \exp\left(-\frac{(x_i - x_j)^T(x_i - x_j)}{2\sigma_l^2}\right) \\
 \textit{Matern5/2} & \rightarrow k(x_i, x_j|\theta) = \sigma_f^2 \left(1 + \frac{\sqrt{5}r}{\sigma_l} + \frac{5r^2}{3\sigma_l^2}\right) \exp\left(-\frac{\sqrt{5}r}{\sigma_l}\right) \\
 \textit{RationalQuadratic} & \rightarrow k(x_i, x_j|\theta) = \sigma_f^2 \left(1 + \frac{r^2}{2\alpha\sigma_l^2}\right)^{-\alpha}
 \end{aligned}$$

burada σ_f sinyal standart sapması, σ_l karakteristik uzunluk, $r = \sqrt{(x_i - x_j)^T(x_i - x_j)}$ ve α pozitif değerli ölçek karışım parametresidir. Benzer şekilde, aşağıdaki gibi dört temel fonksiyon vardır:

$$\begin{aligned}
 \textit{None} & \rightarrow H = [] \text{ (empty matrix)} \\
 \textit{Constant} & \rightarrow H = I_{n \times 1} \\
 \textit{Linear} & \rightarrow H = [1, X] \\
 \textit{PureQuadratic} & \rightarrow H = [1, X, X^2]
 \end{aligned}$$

$$X^2 = \begin{bmatrix} x_{11}^2 & x_{12}^2 & \dots & x_{1d}^2 \\ x_{21}^2 & x_{22}^2 & \dots & x_{2d}^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1}^2 & x_{n2}^2 & \dots & x_{nd}^2 \end{bmatrix} \quad (3.10)$$

Önerilen GPR modellerinin performansı, korelasyon katsayısı (R), ortalama karekök hatası (RMSE) ve ortalama mutlak hata (MAE) kullanılarak değerlendirilir (Pat vd. 2022).

Performans değerlendirme kriterleri

Tanım 1 (MAPE) MAPE (Mean Absolute Percentage Error) regresyon ve zaman serileri modellerinde gerçek ve tahmin edilen değerler arasındaki hata yüzdesinin ortalamasını ölçen bir performans ölçütüdür. Ancak, MAPE'nin sıfır içeren gerçek değerlerle kullanılması durumunda sorunlar ortaya çıkabilir, çünkü sıfır ile bölme işlemi hataya neden olabilir. Ayrıca, çok düşük tahmin değerleri için yüzde hatası %100'ü aşamaz, ancak yüksek tahmin değerleri için bir üst sınır yoktur. MAPE, tahminleyicilerin doğruluğunu

karşılaştırmak için kullanıldığında, özellikle çok düşük tahmin değerleri olduğunda bir önyargı yaratabilir. Bu durum, MAPE'nin sistematik olarak düşük tahminlere yönelme eğiliminde olması nedeniyle ortaya çıkar. Bu sorunu çözmek için, tahmin değerlerinin gerçek değerlere oranını hesaplayan bir doğruluk ölçütü kullanmak faydalı olabilir. Bu yaklaşım, tahminlerin geometrik ortalama açısından yorumlanabilir ve daha dengeli bir doğruluk ölçümü sunabilir.

$$MAPE = \frac{100}{N} \sum \frac{|e_j|}{|A_j|} \quad (3.11)$$

Burada, N, toplam gözlem sayısını temsil eder. e_j , j. gözlemdeki gerçek değer ile tahmin değeri arasındaki hata (fark) olarak tanımlanır. A_j , j. gözlemdeki gerçek değeri temsil eder. Formülde, her bir gözlem için gerçek değer ile tahmin değeri arasındaki hata yüzdesi hesaplanır. Bu hata yüzdesi, mutlak değer alındıktan sonra gerçek değer üzerine bölünür. Son olarak, tüm gözlemlerin hata yüzdesi ortalaması alınır ve 100 ile çarpılır.

Tanım 2 (RMSE) RMSE (Root Mean Square Error) makine öğrenmesi modellerinin performansını değerlendirmek için yaygın olarak kullanılan bir metrik, tahminleyicinin tahmin ettiği değerler ile gerçek değerleri arasındaki uzaklığı ölçen karekök ortalama karesel hata (RMSE) metriğidir.

$$RMSE = \sqrt{MSE} \quad (3.12)$$

Tahmin hataları, regresyon hattının veri noktalarından ne kadar uzakta olduğunun bir göstergesi iken, RMSE bu kalıntıların ne kadar yayıldığını ölçer. RMSE değeri 0 ile ∞ arasında değişebilir. Negatif yönelimli puanlar, yani daha düşük değerlere sahip tahminleyiciler, daha iyi performans sergiler. RMSE değerinin sıfır olması, modelin hiç hata yapmadığı anlamına gelir.

Tanım 3 (MAE) MAE Ortalama Mutlak Hata (Mean Absolute Error), gerçek ve tahmin edilen değerler arasındaki mutlak hataların ortalamasını ölçen bir performans ölçütüdür. Her gerçek değer ile veriye en iyi uyan çizgi arasındaki ortalama dikey mesafeyi ifade eder. Aynı zamanda, her veri noktası ile en iyi uyan çizgi arasındaki ortalama yatay mesafeyi de kapsar.

$$MAE = \frac{1}{n} \sum |e_j| \quad (3.13)$$

MAE değeri, 0 ile ∞ arasında değişebilir ve negatif yönelimli puanlar, yani daha düşük değerlere sahip tahminleyiciler, daha iyi performans sergilediği durumları temsil eder. MAE değeri, kolay yorumlanabilir olması nedeniyle regresyon ve zaman serisi problemlerinde sıkça tercih edilmektedir.

Tanım 4 (SH) *Standart hata (SH), bir istatistiksel örneklem veya bir tahminin belirsizliğini ölçen bir metriktir. Makine öğrenmesi performans değerlendirme bağlamında, standart hata, özellikle modelin tahminlerinin ne kadar güvenilir olduğunu değerlendirmek için kullanılır.*

Standart hata, örneklem veya tahminin gerçek değerden ne kadar sapma gösterdiğini ifade eden bir ölçüdür. Daha düşük bir standart hata, tahminin gerçek değere daha yakın olduğunu, daha yüksek bir standart hata ise tahminin belirsizlik içerdiğini gösterir. Makine öğrenmesinde, genellikle regresyon modellerinin tahmin güvenilirliğini değerlendirmek için kullanılır. Standart hata, bir modelin öğrenme sürecindeki hata miktarını ölçer ve modelin gelecekteki tahminlerinin ne kadar güvenilir olabileceği konusunda bir fikir verir. Standart hata genellikle hata terimiyle ilgili olup, özellikle regresyon modellerinde kullanılan hata terimi standart sapma ile ilişkilidir. Daha düşük standart hata, modelin verilere daha iyi uyması ve gelecekteki tahminlerin daha güvenilir olması anlamına gelir. Bu nedenle, standart hata, bir makine öğrenimi modelinin performansını değerlendirirken, modelin tahminlerinin güvenilirliği hakkında önemli bilgiler sağlar.

Tanım 5 *Korelasyon analizi, değişkenler arasındaki ilişkinin yönünü, derecesini ve önemini belirlemek amacıyla kullanılan bir istatistiksel yöntemidir. Bu analiz, değişkenler arasındaki ilişkinin yönünü ve gücünü gösteren korelasyon katsayısını kullanır. Korelasyon katsayısı, küçük harf "r" ile ifade edilir ve -1 ile +1 arasında değerler alır.*

Eğer r değeri -1'e yakınsa, değişkenler arasında negatif bir ilişki olduğunu gösterir. +1'e yakın değerler ise pozitif bir ilişkiyi ifade eder. Sıfıra yakın r değerleri ise iki değişken arasında bir ilişki olmadığını gösterir. Korelasyon analizi, genellikle Pearson ve Spearman olmak üzere birkaç türde yapılabilir.

Pearson korelasyon analizi, değişkenlerin oransal veya aralıklı ölçekle elde edildiği ve normal dağılıma uygunluk gösterdiği durumlarda kullanılır. Ancak, normal dağılıma uymayan veriler için Spearman korelasyon analizi tercih edilebilir. Bu analiz, oransal veya aralıklı ölçekle elde edilmiş değişkenler arasındaki ilişkiyi değerlendirmek için kullanılır. Sıralı ölçekle elde edilen değişkenler için de Spearman korelasyon analizi uygulanabilir.

Tanım 6 (R^2) Açıklayıcılık Korelasyon Katsayısı (R^2), bir modelin bağımsız değişkenlerin içerdiği toplam varyasyonun yüzde kaçını açıkladığını ölçen bir performans ölçütüdür.

$$R^2 = 1 - \frac{\text{AcıklananVaryasyon}}{\text{ToplamVaryasyon}} \quad (3.14)$$

"AcıklananVaryasyon", model tarafından tahmin edilen değerlerin bağımsız değişkenlerin içindeki varyasyonu ifade eder. "ToplamVaryasyon", bağımsız değişkenlerin genel toplam varyasyonunu temsil eder. R^2 , aynı zamanda belirleme katsayısı veya çoklu regresyon için çoklu belirleme katsayısı olarak adlandırılır. Daha sade bir ifadeyle, R-kare, doğrusal regresyon modellerinin uygunluğunu ölçen bir metriktir. Aynı veri seti için, daha yüksek R^2 değerleri, gözlemlenen veriler ile uyumlu olan tahmin edilen değerler arasında daha küçük farkları ifade eder.

Regresyon analizi, bir bağımlı değişken ile bir veya daha fazla bağımsız değişken arasındaki ilişkiyi modellemek için kullanılan bir istatistiksel yöntemdir. R^2 , bu modelin uygunluğunu ölçen bir istatistik olarak bilinir. Ancak, modeldeki değişikliklerle birlikte R^2 'nin değeri nasıl etkilendiği önemli bir konudur. Yeni bir değişken eklemek, genellikle regresyon modelinin R^2 değerini düşürmez, aksine artırır. Ancak, bu durum her zaman modelin iyileştiği anlamına gelmez.

Tanım 7 (Düzeltilmiş R^2) R^2 'ye dayalı seçim yapmak, modelin neden-sonuç ilişkilerini eksik anlamasına ve yanıltıcı sonuçlara neden olabilir. Daha güvenilir bir değerlendirme için düzeltilmiş R^2 (Adjusted R Square) kullanılır. Düzeltilmiş R^2 modelin karmaşıklığını ve aşırı uyumunu dikkate alarak R^2 'nin düzeltilmiş bir versiyonudur. Düzeltilmiş R^2 (AdjRsqr) formülü aşağıdaki gibidir.

$$\text{AdjRsqr} = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1} \quad (3.15)$$

N: Örneklem büyüklüğü

p: Bağımsız değişkenlerin sayısıdır.

Bu bağlamda, her yeni değişkenin R^2 değerini dikkatlice değerlendirmek ve düzeltilmiş R^2 'ye odaklanmak, regresyon modellerinin daha sağlam ve anlamlı olmasını sağlayabilir. Bu sayede, eklenen değişkenin modelin gerçek performansını yansıtıp yansıtmadığı daha iyi değerlendirilebilir.

Tanım 8 (MSE) *MSE, Ortalama kare hata (Mean Squared Error) (MSE), bir regresyon eğrisinin belirli bir veri nokta kümesine ne kadar yakın olduğunu ölçen bir metriktir.*

$$MSE = \frac{1}{n} \sum e_j^2 \quad (3.16)$$

MSE, her bir tahminin gerçek değerden ne kadar uzak olduğunu ölçer, bu hataları kare alır, ortalar ve sonuç olarak pozitif bir değer elde eder.

MSE'nin makine öğrenmesi modellerinde yaygın olarak kullanılmasının nedenlerinden biri, her zaman pozitif olması ve daha küçük MSE değerlerinin daha iyi performansla ilişkilendirilmesidir. Yani, MSE değeri sıfıra yakın olan bir tahminleyici, veriye daha yakın ve daha doğru tahminler yapma eğilimindedir. MSE, modelin tahminlerinin gerçek değerlere olan yakınlığını nicel olarak değerlendirmek için etkili bir araçtır.

3.1.2 Sınıflandırma

Sınıflandırma, makine öğrenmesi içinde önemli bir konsepttir ve genellikle öğeleri önceden belirlenmiş kategorilere ayırmak için kullanılan bir denetimli öğrenme algoritması olarak kabul edilir.

Sınıflandırma algoritmaları, öğrenme süreci sırasında eğitim veri setinde bulunan öğelerin özelliklerini kullanarak bu öğelerin belirli sınıflara ait olma olasılıklarını öğrenir. Daha sonra, bu öğrenilen modeli kullanarak yeni, görülmemiş öğelerin hangi sınıfa ait olabileceğini tahmin eder.

Farklı sınıflandırma algoritmalarından bazıları şunlardır;

- KNN ((K-Nearest Neighbors)

- Karar Ağaçları (Decision Trees)
- Naive Bayes
- SVM (Support Vector Machine)

KNN (K-en yakın komşular)

KNN (k-Nearest Neighbors), eğitim veri kümelerini kullanarak veri setlerindeki en yakın k veri noktasını bulan bir algoritmadır. KNN, regresyon ve sınıflandırma problemleri için kullanılabilen, denetimli bir makine öğrenmesi algoritmasıdır. Özellikle örüntü tanıma uygulamalarında yaygın olarak kullanılır. Algoritma, veri noktaları arasındaki mesafeleri hesaplar ve belirli bir sorgu noktasına en yakın k sayıda veri noktasını seçer. Sınıflandırma durumunda, bu yakın noktaların sınıfları dikkate alınarak sorgu noktasının sınıfı belirlenir. Regresyon durumunda ise bu noktaların değerleri kullanılarak sorgu noktasının tahmini değeri elde edilir.

KNN algoritması genellikle Öklid uzaklığına bakarak en yakın noktaları belirler. K değeri, bu algoritmanın başarısını etkileyen önemli bir parametredir. Küçük K değerleri modelin gürültüye duyarlı olmasına neden olabilirken, büyük K değerleri sınıflar arasındaki sınırları belirsizleştirebilir (Sevli, 2019).

Bu algoritmanın çeşitli uygulama alanları vardır, bunlar arasında parmak izi algılama, kredi notu tahmini, borsa analizi, kara para aklama analizi, iflas tahmini ve döviz kuru tahmini gibi gerçek hayat problemleri bulunmaktadır. KNN'nin kullanıldığı bu alanlarda, özellikle benzer özelliklere sahip veri noktalarının birbirine yakın olma eğiliminde olduğu durumlar söz konusu olduğunda etkili sonuçlar elde edilebilir.

Karar ağaçları

Karar Ağacı algoritması, gözetimli makine öğrenmesi türünde bir algoritmadır ve regresyon ile sınıflandırma problemlerini çözmek için kullanılır. Temel amacı, gözlemlerden sonuçları işlemek ve anlamak için bir karar ağacı oluşturmaktır. Karar ağacı algoritmasının temel yapısında kök, dal, düğüm ve yapraklar bulunur. Ağaç yapısı, her bir özneliğin bir düğümle ilişkilendirildiği şekilde oluşturulur. Kök ve düğümler arasında dallar, her bir düğümden diğer düğüme geçişi temsil eder.

Karar ağacı algoritması, eğitim verilerinden en uygun özniteliğin kök olarak seçildiği ve işlemin her dal için tekrarlandığı yukarıdan aşağıya bir yaklaşım benimser. Yani, ağaç oluşturma süreci, verilerin özniteliklere göre bölünmesi ve bu bölünmelerin en iyi sonuçları elde etmek için optimize edilmesi üzerine kuruludur.

Her düğüm, bir özniteliği temsil eder ve bu özniteliklerin değerlerine göre dallara ayrılır. Bu işlem, ağaç yapısının her seviyesinde tekrarlanır ve sonunda nihai yaprak düğümlerine ulaşılır (Esmer vd. 2020). Her bir yaprak düğümü, bir sınıf veya regresyon tahminiyle ilişkilendirilmiştir. Karar ağaçları, modelin kararları ve öğrenme süreci açısından yüksek bir açıklanabilirlik sağlar. Ayrıca, regresyon ve sınıflandırma problemlerinde etkili sonuçlar üretebilirler. Karar ağaçları genellikle şu alanlarda kullanılır:

- Bilgi yönetimi platformları oluşturma
- Uçuş seçme (seyahat)
- Oteller için yüksek doluluk tarihlerinin tahmin edilmesi
- Müşterilere ürün önerme
- “Tahminleri tahmin etme” ve çeşitli alanlardaki olasılıkları belirleme

Naive bayes

Koşullu olasılık kuralını temel alan bu algoritma, belirli bir ögenin belirli bir kategoriye ait olma olasılığını hesaplamak için Bayes’in teoremini kullanır. Sınıf değişkeni göz önüne alındığında, özellik çiftleri arasında koşullu bağımsızlık varsayımıyla çalışır, yani bir olayın gerçekleştiği durumda başka bir olayın olma olasılığını belirler. Genellikle istenmeyen mesajları filtrelemede, Netflix gibi öneri sistemlerinde, teknolojiye, politika veya sporla ilgili bir haber makalelerini sınıflandırmada, sosyal medyada duygu analizinde, yüz tanıma uygulamalarında kullanılır.

SVM

SVM, özellikle sınıflandırma problemlerini çözmek için kullanılan bir gözetimli makine öğrenme algoritmasıdır. Bu algoritma, veri noktalarını sınıflandırmak veya belirli bir değeri tahmin etmek amacıyla kullanılabilir.

SVM'nin temel amacı, veri noktalarını iki sınıf arasında bir hiperdüzlemle bölmektir.

SVM, sınıflandırma sürecinde kullanılan hiperdüzlemi belirlerken, bu düzleme en yakın olan veri noktalarına odaklanır. Bu en yakın noktalara "destek vektörleri" denir. SVM, destek vektörlerini kullanarak, veri noktalarını sınıflandırmak veya regresyon analizi yapmak için modeller oluşturur (Sevli, 2019). SVM'nin çekirdek fonksiyonları sayesinde, veriler düşük boyutlu uzaylarda karmaşık şekillerde de sınıflandırılabilir. SVM'nin esnek yapısı, özellikle non-lineer sınıflandırma problemlerini ele almak için kullanılmasını sağlar. Bu nedenle, SVM, birçok uygulama alanında kullanılan güçlü bir algoritma olarak öne çıkar.

Topluluk (Ensemble) algoritması

Topluluk algoritmaları, iki veya daha fazla makine öğrenmesi algoritmasının tahminlerini birleştirerek daha kesin sonuçlar elde etmeyi amaçlayan yöntemlerdir. Genellikle sınıflandırma ve regresyon problemlerinde kullanılır. Bu algoritmaların bir yöntemi olan oylama, farklı modellerin tahminlerini bir araya getirir ve genellikle en çok oy alan sınıfı seçerek sınıflandırma sonuçlarını birleştirir. Bu, çeşitli modellerin avantajlarını birleştirerek daha güvenilir tahminler elde etmeye yardımcı olabilir. Oylama genellikle regresyon sırasında sınıflandırma ve ortalama alma sürecinde kullanılır (Tuğ Karoğlu ve Okut, 2020).

Başka bir yaklaşım ise sonuçların ortalamasını almak veya ağırlıklı bir ortalamayı kullanmaktır. Bu, regresyon problemlerinde özellikle etkilidir çünkü farklı modellerin tahminlerini bir araya getirerek daha hassas ve güvenilir bir tahmin sağlayabilir.

Topluluk algoritmaları, farklı hataları dengeleyerek genel performansı artırabilir ve model çeşitliliğini artırarak özellikle tek bir modelin zayıf olduğu durumlarda faydalı olabilir. Örneğin; veri setindeki beş doktorun hasta için çıkardığı sonuçlardan üç doktorun hasta teşhisi koyması ve iki doktorun sağlıklı teşhisi koyması sonucunda ensemble karar ağaçları algoritmasının çıkardığı sonuç hasta etiketi olur. Bu algoritmaların başlıca üç türü şunlardır: Bagging, Boosting ve Stacking.

Bagging algoritmaları, u yöntemde, aynı öğrenme algoritması kullanılarak farklı eğitim veri kümeleri oluşturulur. Her veri seti bootstrap örnekleme ile seçilir

(veri noktalarının tekrarlı örnekleme yapılması). Ardından, bu farklı veri kümeleri üzerinde aynı model eğitilir ve sonuçlar birleştirilir (Tuğ Karoğlu ve Okut, 2020). En yaygın bagging algoritması, Random Forest'tir.

Boosting algoritmaları Boosting, zayıf öğrenen modelleri bir araya getirerek güçlü bir model oluşturma stratejisidir. Her bir model, önceki modellerin hatalarını düzeltmeye odaklanır. Adım adım güçlenen modellerle bir sonuca ulaşılır (Tuğ Karoğlu ve Okut, 2020).

Stacking algoritmalarının, Stacking, bir dizi farklı modelin tahminlerini bir araya getirme yöntemidir. Temelde, birinci seviyedeki modellerin tahminleri, ikinci bir modelin girişi olarak kullanılır ve bu ikinci seviye model, bu tahminleri birleştirir. Stacking, farklı model türlerinin birbirini tamamlayıcı özelliklerini kullanarak daha güçlü bir model elde etmeyi amaçlar.

Bu topluluk algoritmaları, model çeşitliliğini artırarak genelleme yeteneklerini iyileştirebilir ve tek bir modele kıyasla daha iyi performans sağlayabilirler. Hibrit sonuçlar elde edilebilir.

Performans değerlendirme kriterleri

Karışıklık matrisi (confusion matrix), bir sınıflandırma modelinin performansını değerlendirmek için kullanılan bir tablodur. Karışıklık matrisi, gerçek sınıfları ve tahmin edilen sınıfları içerir. Genellikle iki sınıf için kullanılır ve dört temel değeri içerir: true positive (TP), false negative (FN), false positive (FP), ve true negative (TN). True Positive (TP), modelin doğru bir şekilde pozitif sınıfları tahmin ettiği durumdur. Yani, gerçekte pozitif olan bir örneği pozitif olarak sınıflandırmasıdır. False Negative (FN), modelin gerçekte pozitif olan bir örneği negatif olarak sınıflandırdığı durumdur. Yani, bir hata yaparak pozitif örneği kaçırmasıdır. False Positive (FP), modelin gerçekte negatif olan bir örneği pozitif olarak sınıflandırdığı durumdur. Yani, bir hata yaparak negatif örneği yanlışlıkla pozitif olarak sınıflandırmasıdır. True Negative (TN), modelin doğru bir şekilde negatif sınıfları tahmin ettiği durum. Yani, gerçekte negatif olan bir örneği doğru bir şekilde negatif olarak sınıflandırmasıdır. Bu terimlerle karışıklık matrisi aşağıdaki gibidir.

$$\begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$$

Bu matris, sınıflandırma modelinin performansını değerlendirmek için kullanılan birçok metrik ve oranın hesaplanmasına olanak tanır. Örnek olarak, doğruluk oranı (accuracy), duyarlılık (sensitivity), özgüllük (specificity), F1 skoru gibi metrikler bu matristen türetilebilir. Bu metrikler, modelin hangi sınıfları ne kadar doğru tahmin ettiğini ve hangi hataları yaptığını anlamak için kullanılır.

Doğruluk Oranı, genel performans hakkında bilgi verir. 0 ile 100 arasında değer alır.

$$\text{Doğruluk Oranı} = \frac{TP+TN}{(FP+TN)+(TP+FN)}$$

Duyarlılık, modelin tahmin ettiği sonuçları yüzde kaç oranında doğru bulduğunu gösterir. 0-1 arasında değer alır.

$$\text{Duyarlılık} = \frac{TP}{TP+FN}$$

Özgüllük, bir sınıflandırma modelinin negatif sınıfları doğru bir şekilde tanıma yeteneğini ifade eden bir değerdir. 0-1 arasında değer alır.

$$\text{Özgüllük} = \frac{TN}{TN+FP}$$

F- ölçümü, duyarlılık ve özgüllük değerlerini birleştiren bir performans ölçüsüdür. Özellikle dengesiz sınıflandırma problemlerinde, yani sınıflar arasında örnek sayıları eşit olmayan durumlarda kullanılır. 0-1 arasında değer alır.

$$\text{F-Ölçümü} = \frac{2 \times \text{Sen} \times \text{Spe}}{\text{Sen} + \text{Spe}}$$

Burada *Sen*, duyarlılık ve *Spe*, özgüllük değerleridir.

Auc Katsayısı: ROC altında kalan eğri alanıdır. ROC eğrisi altındaki alan (AUC), bir parametrenin iki grubun ne kadar iyi ayırt edilebildiğinin bir ölçüsüdür. 0-1 arasında değer alır.

ROC Eğrisi: ROC (Receiver Operating Characteristic) eğrisi, özgüllük raporu oluşturarak bir parametrenin performansını değerlendirmeyi sağlar. ROC eğrisi, gerçek pozitif oranı (hassasiyet) ile yanlış pozitif oranı (100-özgüllük) arasındaki ilişkiyi gösterir. Bu eğri, bir parametrenin farklı kesme noktalarında performansını görselleştirmek için kullanılır. Her nokta, belirli bir karar eşliğine karşılık gelen bir duyarlılık/özgüllük çiftini temsil eder.

Örnek 1 Diyabet hastalarının 260 veriden oluşan veri setinde hasta yada sağlıklı olma durumlarını karışıklık matrisinde inceledik. İki popülasyon arasında ayırım yapmak için belirlenen her olası kesme noktası veya kriter değeri için, doğru pozitif (TP = True Positive) durumlar, hastalığın doğru bir şekilde saptandığı durumları ifade eder. Ancak bu durumda, hastalığa sahip bazı bireyler yanlış negatif olarak sınıflandırılacaktır (FN = False Negative). Diğer yandan, hastalığı olmayan bireyler arasında doğru negatif (TN = True Negative) durumlar vardır, yani hastalığın olmadığı doğru bir şekilde tespit edilen durumlar. Ancak, hastalığa sahip olmayan bazı bireyler yanlış pozitif olarak sınıflandırılacaktır (FP = False Positive). Bu terimler, bir sınıflandırma modelinin performansını değerlendirmede yaygın olarak kullanılır. Bu, özellikle tıbbi teşhisler, güvenlik uygulamaları ve diğer sınıflandırma problemleri ve modelin doğruluğunu anlamak için önemli bir rol oynar. Bu veri seti üzerinden yapay zekanın ürettiği sonuçlar ile çapraz tablo oluşturulması aşağıdaki gibi örneklendirilmiştir.

		Gerçek Durumlar		
		TP	TN	Toplam
Tahmin Değerleri	Pozitif	150 (TP)	5 (FP)	155 (TP+FP)
	Negatif	10 (FN)	95 (TN)	105 (FN+TN)
Total		160	100	

Burada yapay zeka hasta olan yani pozitif olan değerlerin 150 sini doğru, 10 tanesini yanlış bulmuştur. Aynı şekilde sağlıklıların 95 ini doğru ve 5 tanesini yanlış bulmuştur. Bu değerlerin doğruluk oranı, duyarlılık, özgüllük ve F ölçümü hesaplandığında sırasıyla %94, %93, %95, %93 dür.

3.2 Gözetimsiz Öğrenme

Gözetimsiz makine öğrenmesi algoritmaları, eğitim verilerinin sınıflandırılmadığı veya etiketlenmediği durumlarda devreye girer. Bu algoritmalar, sistemlerin etiketlenmemiş verilerdeki potansiyel yapıları anlamak ve ortaya çıkarmak için bir yetenek geliştirmesini sağlar.

Bu yaklaşım, sistemlerin doğru çıktıları önceden bilmemesi durumunda işlev gösterir. Veri setlerindeki desenleri keşfetmeye yönelik bir strateji benimseyen algoritmalar, doğru çıktıyı bulamadıklarında veriler üzerinde keşifler yaparlar. Etiketlenmemiş verilerden çıkarımlar yaparak ve veri kümelerini inceleyerek, sistem gizli yapıları anlamak ve ortaya çıkarmak için çaba harcar.

Gözetimsiz öğrenme, veri içindeki doğal yapıları keşfetme ve bu yapıları anlama konusunda güçlü bir araçtır. Bu algoritmalar genellikle kümeleme, boyut azaltma ve desen tanıma gibi görevlerde etkilidir.

3.2.1 Kümeleme

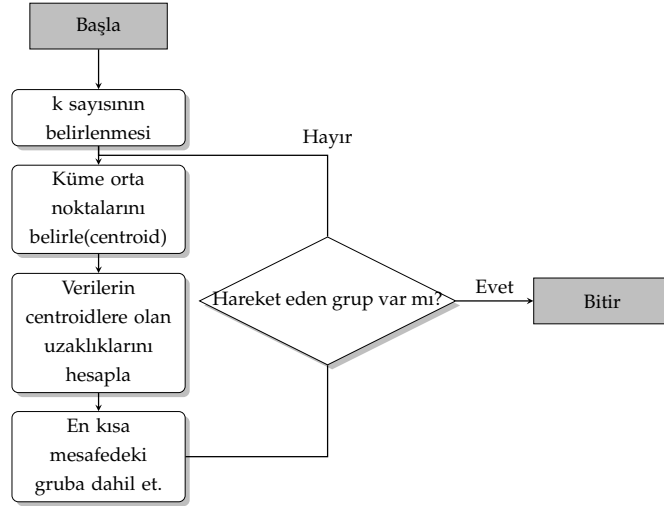
Kümeleme algoritmaları, gözetimsiz algoritmalar grubuna ait olup veri noktalarını gruptandırmak için kullanılırlar. Aynı küme içinde bulunan noktalar, genellikle farklı kümelerdeki noktalardan daha çok birbirine benzemektedir. Bu algoritmalar, benzer özelliklere sahip örnekleri tespit ederek veri setindeki doğal gruplamaları ortaya çıkarmak amacıyla kullanılır (Atalay ve Çelik, 2017). Kümeleme algoritmaları genel olarak parçalayarak kümeleme ve hiyerarşik kümeleme olarak ikiye ayrılır.

Parçalayarak kümeleme, verilerin belirli sayıda gruplara ayrılmasıdır. Merkez tabanlı kümeleme algoritması, verileri başlangıç koşullarına ve aykırı değerlere göre gruplara ayıran bir yöntemdir. K-means, en yaygın olarak kullanılan merkez tabanlı kümeleme algoritmalarından biridir. Yoğunluk tabanlı kümelemede ise algoritma yüksek yoğunluklu bölgeleri tanımlayarak ve rastgele şekillerde dağılmış kümeler oluşturarak çalışır. Ek olarak dağıtım tabanlı kümeleme algoritması, verilerin olasılık dağılımlarından türediğini varsayar ve ardından verileri bu dağılımın farklı sürümlerine göre gruplara ayırır.

Hiyerarşik Kümeleme (Hierarchical Clustering) hiyerarşik veri kümelerinden oluşan bir ağaç oluşturur. Bu ağaç, veri setindeki benzer örnekleri birleştirerek alt kümeleme düzeylerini gösterir. Bu yöntemde, küme sayısı belirli bir ağaç seviyesindeki bir kesme işlemi ile kontrol edilebilir. Alt alta sıralanmış hiyerarşik kümeler yukarıdan aşağıya doğru alt kümeler oluşturur. Dendrogramlar yardımıyla bu yaklaşım gerçekleştirilir.

k-Means Algoritması:

Kümelemede en çok kullanılan algoritma k-means algoritmasıdır. Parçalayarak kümelere ayırarak merkez noktası üzerinden kümeleme yapar. K-means algoritmasında veri setini kaç tane kümeye ayırmamız gerektiği söylenir. Bunun için k'yı belirlemenin yollarından biri Elbow yöntemidir. Her veri en yakın orta noktaya sahip bir kümeye dahil olur. k-Means algoritmasının çalışma şekli aşağıdaki gibidir:



Şekil 3.4 k-Means Algoritması

Spektral kümeleme yöntemi:

K-means algoritması ile aynı amaca ulaşmak için ancak grafik tabanlı bir yaklaşımla kullanılır. Her veri noktasını bir grafik düğümü olarak alır ve kümeleme problemini çözmek için grafik bölümlenmeyi kullanır. Verileri bir grafik şeklinde elde ettikten sonra matrisin her sütununda i ve j düğümleri arasındaki kenarın ağırlığını girerek bir benzerlik matrisi oluşturabilir. Bu bir simetrik matristir.

Bazı veri yapıları k-means ile kümelenmeyebilir. Bu yüzden Spektral kümeleme gibi grafik kümeleme yaklaşımları, veri noktalarını doğrudan yerel veri alanlarında kümelemez. Bunun yerine (i,j) ile benzerlik matrisi oluşturur.

Gauss karışım modeli:

Gauss bir olasılık yöntemidir. Kümeleri elips şeklinde kullanarak daha esnek modeller oluşturur. K-means algoritmasının iki dezavantajı vardır. Küme yapısı esnek değildir ve verileri kümelediğinde bir daire gibi düşünürsek kesişen daireler üzerindeki verileri gruplaması zayıftır. Diğer dezavantajı ise olasılıksal bir model olmamasıdır.

Kümelemede Performans Değerlendirme Kümeleme performansının değerlendirilmesi amacıyla geliştirilen benzerlik indeksleri, birçok araştırmacı tarafından incelenmiştir ve bu alanda çeşitli ölçütler önerilmiştir. Aslanyürek ve

Mesut, 2021 tarafından yapılan çalışma da benzerlik indekslerine odaklanmıştır. Kümeleme performansını ölçmek için kullanılan yöntemler genellikle dahili ve harici yöntemler olarak iki ana grupta ele alınır. Dahili yöntemler, küme içindeki benzerlikleri değerlendirirken, harici yöntemler kümeleme sonuçlarının dış referanslarla olan uygunluğunu değerlendirir. Bu iki grup, kümeleme algoritmalarının etkinliğini değerlendirmek ve karşılaştırmak için kullanılan temel ölçütleri sağlar.

Dahili Yöntemler

Gerçek küme etiketlerine ihtiyaç duymadan kümeleme performansını değerlendiren ölçütleri içerir. Bu kapsamda, Silhouette İndeksi önemli bir metrik olarak öne çıkar.

Silhouette İndeksi, kümelenen verilerin ait olduğu kümedeki uygunluğunu ölçen bir ölçüdür. Her veri noktasının kendi kümesi içindeki benzerlik düzeyini değerlendirir ve bunu, aynı kümedeki diğer veri noktalarına olan ortalama uzaklık ile en yakın diğer kümedeki veri noktalarına olan ortalama uzaklık arasındaki farkın normalizasyonu ile hesaplar. Bu sayede, kümeleme algoritmalarının etkinliği daha ölçülebilir hale gelir. Silhouette indeksi aşağıdaki eşitlik ile hesaplanır.

$$S = \frac{(b - a)}{\max(a, b)} \quad (3.17)$$

İndeksin hesaplanmasında, 'a' değeri, bir örneğin aynı kümedeki diğer tüm noktalara olan ortalama mesafeyi temsil ederken; 'b' değeri, bir örneğin en yakın olduğu başka bir kümedeki diğer tüm noktalara olan ortalama mesafeyi belirtir. Bu değerlerin kullanılmasıyla hesaplanan Silhouette İndeksi, -1 ile 1 arasında bir sayı üretir. Elde edilen bu 's' değeri, 1'e yaklaştıkça kümelemenin daha iyi olduğunu gösterir, çünkü bu durumda örnekler kendi kümesindeki diğer noktalara daha yakın ve diğer kümelerle daha uzak olmaktadır.

Calinski-Harabasz İndeksi, bir veri setinin k adet kümeye bölünmüş halinin kümeleme geçerliliğini değerlendirmek için kullanılan bir ölçüdür. Bu indeks, küme merkezlerinin etkisi altındaki değişkenliklerin küme içindeki değişkenliklere oranını değerlendirerek kümeleme kalitesini ölçer. Hesaplamak için kullanılan formül, küme merkezlerinin oluşturduğu toplam karelerin, küme içindeki

değişkenliklerin toplamına oranını alırken, veri noktalarının sayısı ve küme sayısını da dikkate alır. Hesaplamak için kullanılan formül aşağıda verilmiştir.

$$ch = \frac{tr(B_k) \times (n_E - k)}{tr(W_k) \times (k - 1)} \quad (3.18)$$

$tr(B_k)$: küme merkezlerinin ortalamasından kaynaklanan toplam kareler toplamı,

$tr(W_k)$: küme içi değişkenliğin toplamı,

n_E : veri noktalarının toplam sayısı,

k : küme sayısı

Elde edilen Calinski-Harabasz İndeksi'nin yüksek olması, daha iyi bir kümeleme geçerliliğini ifade eder.

Davies-Bouldin İndeksi, bir veri setindeki verilerin kendi küme merkezlerine olan uzaklıklarını minimize etmeyi ve aynı zamanda küme merkezleri arasındaki uzaklıkları maksimize etmeyi amaçlayan bir ölçüdür. Bu yöntem, kümeleme geçerliliğini değerlendirmek için kullanılır. İndeks, her kümenin merkezine olan uzaklıkların minimum olmasını ve farklı kümeler arasındaki uzaklıkların maksimum olmasını hedefler. Hesaplamak için kullanılan eşitlik, bu iki uzaklığı dikkate alarak kümeleme geçerliliğini ölçer. Kümeleme geçerliliğini ölçmek için aşağıdaki eşitlik kullanılır.

$$db = \frac{1}{k} \sum_{i=1}^k R_i \quad (3.19)$$

Her bir kümenin (i) diğer kümelerle (j) arasındaki maksimum karşılaştırma oranını belirlemek için, i. kümenin diğer kümelerle olan ilişkisini ifade eden R_{ij} küme indeksi aşağıdaki formülle hesaplanır.

olmak üzere i . ve diğer kümeler arasındaki maksimum karşılaştırma oranı her bir küme için R_{ij} ile gösterilen küme indeksi aşağıdaki eşitlik ile hesaplanır.

$$R_{ij} = \frac{S_i + S_j}{d_{ij}} \quad (3.20)$$

$i = 1, 2, \dots, k$ ve $j = 1, 2, \dots, k$

d_{ij} : kümelerde bulunan merkezler arasındaki mesafe

S_i ve S_j : küme gözlemlerinin bulunduğu kümenin merkezlerine olan ortalama mesafe

Küçük db değerleri, daha iyi ve anlamlı bir kümeleme yapılan bir modelin işaretidir.

Harici yöntemler

Düzeltilmiş Rand İndeksi (ARI), kümeleme algoritmalarının performansını ölçmede sıkça kullanılan bir metrik olarak karşımıza çıkar. Bu indeks, iki küme arasındaki benzerliği değerlendirmek amacıyla geliştirilmiş bir versiyondur. Geleneksel Rand İndeksi, bir küme içindeki verileri karşılaştırırken sınırlamalara sahiptir ve küme sayısının artmasıyla ortaya çıkan sorunları çözmekte yetersiz kalabilir. ARI, bu tür zorlukları aşmak ve daha güvenilir sonuçlar elde etmek için tasarlanmıştır.

ARI'nin ürettiği değerler 0 ile 1 arasında yer alır; 1'e yaklaşan değerler, gerçek sınıflar ile kümeleme sonuçlarının yüksek bir uyum içinde olduğunu gösterir. Bu metrik, gerçek sınıfları C ve kümeleme sonucu oluşan sınıfları K temsil eder. Dolayısıyla, ARI, kümeleme algoritmalarının doğruluğunu değerlendirirken küme sayısının etkisini düzelterek daha kapsamlı bir değerlendirme sunar.

$$RI = \frac{a + b}{C_2^{n_{\text{örnekler}}}} \quad (3.21)$$

Burada, 'a', gerçek sınıflar (C) ve kümeleme sonuçları (K) arasında aynı kümede bulunan örnek sayısını ifade ederken; 'b', farklı kümelerde yer alan örneklerin sayısını temsil etmektedir.

E[RI], beklenen Rand İndeksi örnek sayısını simgeler ve aşağıdaki matematiksel ifade ile hesaplanır.

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]} \quad (3.22)$$

Rand İndeksi (RI), modelin doğruluğunu ölçerken Ayarlanmış Rand İndeksi (ARI), rastgele eşleşmelerin beklendiği durumu dikkate alarak daha düzeltilmiş halini verir. Mirkin İndeksi (MI), iki farklı sınıflandırma yöntemi arasındaki uyumsuzluğu ölçerken, Hubert İndeksi (HI), Mirkin İndeksi ve Rand İndeksi arasındaki farkı ifade eder. Bu metrikler genellikle karışıklık matrisi temel alınarak hesaplanır ve değerlendirme sürecinde modelin performansını daha kapsamlı bir şekilde anlamak için kullanılır.

Doğruluk, Kesinlik, Duyarlılık ve F1 Skoru, özellikle doküman sınıflandırma gibi alanlarda sıkça kullanılan metriklerdir. Ancak, kümeleme durumlarında doğrudan uygulanmaları zor olabilir. Kümeleme bağlamında, Silhouette İndeksi, Calinski-Harabasz İndeksi ve Davies-Bouldin İndeksi gibi spesifik metrikler, kümeleme algoritmalarının performansını daha uygun bir şekilde değerlendirebilir. Bu metrikler, kümeleme özelinde benzerlik, dağılım ve homojenlik gibi kriterlere odaklanarak daha anlamlı sonuçlar elde etmeye yardımcı olabilir.

3.3 Yarı Denetimli Algoritmalar

Yarı gözetimli makine öğrenmesi, hem denetlenen hem de denetlenmeyen algoritmaların özelliklerini birleştirerek ortaya çıkan bir yaklaşımı temsil eder. Bu algoritmalar, eğitim verilerinin tamamının etiketlenmemiş olabileceği veya algoritma başlatılırken tüm kuralların sağlanmadığı durumları kapsar.

Yarı gözetimli makine öğrenmesi algoritmaları, eğitim sürecinde etiketli ve etiketlenmemiş verileri bir arada kullanır. Sınırlı sayıda etiketli veri ve geniş bir etiketlenmemiş veri kümesi içeren sistemler, öğrenme doğruluğunu önemli ölçüde artırabilir. Bu yöntem, özellikle etiketleme sürecinin maliyetli veya zaman alıcı olduğu durumlarda avantajlı olabilir.

3.4 Takviyeli Öğrenme

Takviyeli öğrenme, sistemin kendi başına kararlar alabilen bir yapıya sahip olmasını ve hedefine ulaşmak için gerekli yetenekleri öğrenmesini vurgular. Genellikle, bu yöntemler belirsiz veya dinamik ortamlarda kullanılır ve uygulama

alanları arasında robotik, oyun programlama, hastalık teşhisi ve üretim otomasyonu yer alır.

Makine öğrenmesi algoritmalarının birçoğu, keşfetme adı verilen bir süreç kullanarak çalışır. Bu süreçte, sistem çevresiyle etkileşime girer, çeşitli eylemler gerçekleştirir, bu eylemlerin sonuçlarını gözlemler ve bu gözlemleri kullanarak gelecekteki adımlarını belirler. Bu şekilde, algoritma zaman içinde gelişir ve optimum stratejiyi bulana kadar bu döngü devam eder. Bu yöntem, makinelerin belirli bir bağlamda en etkili davranışı otomatik olarak öğrenmelerine izin verir.

3.5 İlgili Çalışmalar

Literatürde Bitcoin ile alakalı yapılan çalışmalar fiyat tahmini ve kriptoloji alanında yoğunlaşmıştır.

Bitcoin fiyatının zaman içinde doğrusal olmayan bir eğilimi olmuştur. Bitcoin'leri dikkate alan ve değişen değişkenlerle eğriye uyan bir modelle gelecekteki Bitcoin fiyatını tahmin etmek için doğrusal olmayan modellerin daha iyi performans göstermesine neden olan birçok varsayım vardır. Zaj vd. 2022 çalışmasında, Bitcoin fiyatını, Grey model, Yapay Sinir Ağının Geri Yayılımı modeli ve Gri Sinir Ağının Entegre modeli olarak bilinen üç doğrusal olmayan modeli kullanarak etmeyi amaçlayarak 2014-2018 yıllarına ait beş yıllık bitcoin fiyat verilerini kullanmışlardır. Bitcoin fiyatının günlük ve aylık olarak tahminini üç modeli kullanarak göstermişlerdir. Günlük fiyat tahmin sonuçları, Geri Yayılım Yapay Sinir Ağı modeli, Gray modeli ve entegre modele kıyasla %5,6 en düşük mutlak hata oranına sahip olduğu belirlenmiştir. Bitcoin aylık fiyat tahmin sonuçlarında ise Grey sinir ağı entegre modelinin yüzde 10'dan daha azına eşit olan hata yüzdesi en az mutlak ortalamaya sahiptir. Bu nedenle daha iyi bir performansı vardır. Doğrusal olmayan modeller Bitcoin'in fiyatını tahmin ederken daha başarılı sonuçlar vermiştir. Belirtilen sonuçlara göre bitcoin'in günlük fiyatını tahmin ederken, sinir ağı modelinin en az MAPE ve RMSE'ye sahip olduğu görülmektedir.

BTC fiyatında çok fazla dalgalanmalar olur ve bu da tahmin etmeyi zorlaştırır. Shankhdhar vd. 2021 çalışmasında, BTC fiyatının tahmin edilmesi için Multivariate Linear Regression, Theil-Sen Regression, Huber Regression gibi farklı makine öğrenme modelleri ile LSTM, GRU gibi derin öğrenme algoritmalarının kullanılmasıyla daha hızlı ve doğru bir model bulma hedeflenmiştir. Çalışmanın

veri seti, 6 Ağustos 2016'dan 14 Eylül 2020'ye kadar olan ABD Doları cinsinden 1501 veri noktası içerir. BTC verileri kapanış fiyatı , açılış fiyatı, en yüksek-en düşük fiyat özelliklerinden oluşmaktadır. Veriler çok büyük olduğundan veri setlerini MongoDB'de (Big-Data Tool) tutmuşlardır. Bitcoin fiyatının değeri bir eşik değere ulaştığında kullanıcıyı uyaran bir uyarı sistemi oluşturmak için IOT sistemini kullanmışlardır. Araştırmanın sonuçlarına göre, tüm modellerin yaklaşık olarak benzer doğruluk seviyelerine sahip olduğu, ancak doğrusal regresyon modelinin en iyi yürütme süresine sahip olduğu tespit edilmiştir.

Livieris vd. 2021 tarafından yürütülen bu çalışma, en yüksek piyasa değerine sahip üç kripto para birimi olan Bitcoin (BTC), Ethereum (ETH) ve Ripple (XRP) üzerine odaklanarak, üç yıllık art arda gelen verileri kullanarak kapsamlı bir ampirik araştırma sunmaktadır. Bu çalışmada kullanılan veriler, USD cinsinden BTC, ETH ve XRP'nin 1 Ocak 2017'den 31 Ekim 2020'ye kadar olan günlük geçmiş verilerini içermektedir. Değerlendirme amacıyla, kripto para birimi verileri eğitim seti, doğrulama seti ve test seti olarak ayrılmıştır. Eğitim seti (1153 veri noktası), doğrulama seti (94 veri noktası) ve test seti (152 veri noktası) olarak ayrılmıştır. Kripto para birimi fiyatının ve hareketinin tahmini için MICDL olarak adlandırılan çok girişli bir mimariye dayalı derin bir sinir ağı modeli önerilmiştir. Önerilen tahmin modeli, her bir kripto para birimi bilgisinin başlangıçta ayrı ayrı kullanılması ve işlenmesi için bunları bağımsız girdiler olarak kullanmıştır. Önerilen modelin karışık kripto para birimi verilerini verimli bir şekilde kullanma yeteneğine sahip olduğu, aşırı öğrenme (overfitting) sorununu azalttığı ve bağlantılı derin sinir ağlarına kıyasla hesaplama maliyetini azalttığı sonucuna varmıştır.

Phaladisailoed ve Numnonda, 2018 tarafından gerçekleştirilen çalışma, Bitcoin fiyatlarını tahmin etmek için çeşitli makine öğrenme algoritmaları arasında en verimli ve en yüksek doğrulukla çalışan modelin belirlenmesi amaçlanmıştır. 1 Ocak 2012'den 8 Ocak 2018'e kadar bitstamp adlı Bitcoin borsa web sitesindeki 1 dakikalık aralıklı ticaret verileri kullanılarak, scikit-learn ve Keras kitaplıklarıyla çeşitli regresyon modelleri test edilmiştir. Scikit-Learn kütüphanesi ile karşılaştırmak için Theil-Sen regresyonu ve Huber regresyonu seçilmiştir. Derin öğrenme tabanlı regresyon modelleri olan LSTM ve GRU modellerini oluşturmak için keras kütüphanesini kullanmışlardır. Veri setinde close (kapanış fiyatı), open (açılış fiyatı), high (gün içindeki en yüksek fiyat), low (gün içindeki en düşük fiyat), weighted price (ortalama bitcoin fiyatı), volume_(BTC) (BTC'de toplam fiyat hacmi), volume_(Currency) (USD'deki toplam fiyat hacmi), timestamp (veri kaydedilen süre) özelliklerini kullanmışlardır.

Sonuçlar, derin öğrenme temelli regresyon modellerinin: GRU ve LSTM'nin Theil-Sen regresyonu ve Huber regresyonundan daha iyi sonuç verdiğini göstermektedir. En iyi sonuçlar GRU için, ortalama kare hatanın (MSE) 0.00002 kadar düşük ve R-Karenin (R^2) %99.2 kadar yüksek olduğunu göstermiştir. Fakat, Huber regresyon kullanımının hesaplanan süresi LSTM ve GRU'dan çok daha az olarak hesaplanmışlardır. Buna ek olarak, seçilen özellikler sosyal medyadan gelen tepkiler, politikalar ve yasalar gibi çeşitli faktörler nedeniyle bitcoin fiyatlarını tahmin etmek için yeterli olmayabilir sonucuna varmışlar. Bu nedenle, tüm modellerin en iyi sonuçlarına göre, veri kümeleri her zaman güncellenmiş veriler toplanması gerektiği sonucuna varmışlardır.

Madan vd. 2015 tarafından yürütülen bu çalışma, iki aşamalı araştırma yolu izlemişlerdir. İlk aşamada bitcoin fiyatını çevreleyen optimal özellikler hakkında bilgi edinirken bitcoin pazarındaki günlük eğilimleri anlamayı ve daha iyi tanımlamayı amaçlamışlardır. Makine öğrenmesi yöntemleri ile BTC fiyatı ile ilişkili 25 farklı özelliği kullanarak günlük verilerden oluşan beş yıllık BTC fiyat verisi üzerinde çalışmışlardır. Bu özellikler arasından kendilerinin uyumlu bulduğu 16 özelliği kullanarak BTC verileri üzerinde üç farklı binomial sınıflandırma algoritması kullanmışlardır. Bu algoritmalar, İki durumlu lojistik regresyon(binomial GLM), destek vektör makinesi ve rastgele orman algoritmalarıdır. Elde edilen algoritmanın başarımının %98.7 olduğunu ifade etmişlerdir. İkinci aşamada, Bitcoin fiyat tahminlerini çeşitli hassasiyet düzeylerinde ve gürültüye göre değerlendirmek amacıyla 10 dakikalık ve 10 saniyelik zaman dilimlerinde kaldıraç verileri kullanılmıştır. Elde edilen modelin, 10 dakikalık zaman aralıklarında gelecekteki fiyat değişimi tahmininde %50-55 arasında bir başarı oranına sahip olduğu belirtilmiştir. Rastgele orman modelinin GLM'ye kıyasla daha iyi bir doğruluk oranı verdiğini gözlemlemişlerdir.

Borsanın temel prensibi, matematiksel işlemlere, grafiklere ve belirli indikatörlere dayanan teknik analizdir. Yatırımcılar, bu analiz yöntemlerinin ürettiği tahmin sonuçlarına güvenerek işlemlerini gerçekleştirirler. Teknik analiz, geçmiş fiyat hareketleri ve ticaret hacimleri üzerinden gelecekteki fiyat eğilimlerini tahmin etmeye odaklanan bir strateji olarak, finans piyasalarında önemli bir yer tutmaktadır. Arslan ve Kirci, 2021 çalışmasında, makine öğrenmesi kullanılarak geçmiş yıllara ait verilerle bir sistem eğitilmiş ve bu sistem, gelecekteki Bitcoin verilerini analiz ederek borsa hareketlerinin momentumunu görselleştirmeyi amaçlamıştır. Elde edilen analiz sonuçlarına dayanarak, kullanıcıya al ve sat sinyalleri üretmek hedeflenmiştir. Bitcoin borsasını ele almışlardır. Bitcoin'in günlük grafikleri üzerinden en yüksek, en düşük, hacim ve arz talep verileri

kullanılarak al-sat sinyalleri üretmek amacıyla doğrusal regresyon yöntemi kullanılmıştır. Giriş verilerinde RSI ve MACD indikatörlerini kullanmıştır. Yapılan test sonuçları, geliştirilen sistemin %95,5 doğruluk oranına sahip olduğunu göstermiştir. Ancak, çalışma sonrasında yapılan incelemelerde, bir sosyal medya paylaşımının ardından Bitcoin'in değerinin hızla tahmin edilen değer aralığının dışına çıktığı belirlenmiştir. Bu inceleme, diğer küresel borsalarda görülen manipülasyon hareketlerinin genellikle %5 ila %10 civarında değişim gösterirken, Bitcoin üzerinde bu manipülasyon hareketlerinin %30'luk bir değişim oranının üzerine çıkabileceğini öne sürmektedir. Bu durum, sistemin etkinliği ve Bitcoin'in hassasiyeti açısından önemli bir değerlendirme sunmaktadır.

Yazarın önerisi doğrultusunda, gelecekteki çalışmalar için RSI ve MACD gibi daha fazla teknik analiz yönteminin eklenmesi düşünülebilir. Bunun yanı sıra, medya ve tanınmış kişilerin spekülasyonlarına karşı duyarlı bir şekilde tepki verebilen bir botun geliştirilmesi ve ayrıca projenin kapsamının sadece Bitcoin ile sınırlı kalmayarak farklı borsalarda daha güvenli hisse senetleri gibi varlıkları içerecek şekilde genişletilebileceği belirtilmiştir. Bu öneriler, sistemdeki analitik kapasitenin ve çeşitliliğin artırılmasına yönelik önemli adımları içermektedir.

Wu vd. 2022 çalışmasında, S_I_LSTM olarak adlandırılabilen, birden çok veri kaynağını ve yatırımcı duyarlılığını içeren bir hisse senedi fiyatı tahmin yöntemi uygulanmıştır. Veriler, hisse senedi geçmiş verilerini, teknik göstergeleri ve hisse senedi gönderileri ve finans haberleri gibi geleneksel olmayan veri kaynaklarını içerir. Geleneksel olmayan veriler için, yatırımcıların duyarlılık endeksini hesaplayabilen sinir ağına dayalı duyarlılık analizi yöntemini kullanmışlardır. Hisse piyasası olarak China Shanghai A-share piyasasını tahmin etmek için uzun kısa süreli bellek ağını kullanmışlardır. Deneyler, tahmin edilen hisse senedi kapanış fiyatının gerçek kapanış fiyatına tek veri kaynağından daha yakın olduğunu ve ortalama mutlak hatanın geleneksel yöntemlerden daha iyi olan 2,386835'e ulaşabileceğini göstermiştir.

H. Liu, 2021 çalışmasında, temel bileşen uzun kısa süreli bellek (LSTM) modeline dayalı hisse senedi tahmini üzerine araştırma yapılmıştır. LSTM hisse senedi tahmin modelini kurmak için teknik yöntemler kullanılırken, birçok girdi veri değişkeni, veri bilgilerinin üst üste binmesi ve aykırı değerlerin eğitim ve diğer faktörler üzerindeki büyük etkisi nedeniyle, geleneksel yöntemler genellikle zayıf genellemeye ve zayıf tahmin etkisine yol açar. Bu tür sorunlar ışığında, temel verilerin boyutunu azaltmak için temel bileşenler analizini (PCA) kullanmayı ve ardından stokla ilgili teknik göstergeler KDJ ve MACD'yi girdi verileri olarak

birleştirmeyi ve stok karakteristik modeline göre ayarlamadan sonra tahmin yapmayı amaçlamıştır.

LSTM, zaman serilerinde nispeten uzun aralıklı ve gecikmeli önemli olayları işlemek ve tahmin etmek için uygun bir tür zaman özyinelemeli sinir ağı olan uzun süreli ve kısa süreli hafızalı yapay sinir ağıdır. Temel bileşen analizi (PCA), uzamsal koordinat döndürme yoluyla örnek veri yapısını değiştirmeden, birbiriyle ilişkili bir dizi orijinal veriyi az sayıda ilişkisiz doğrusal kombinasyonlara dönüştürmektir. Orijinal LSTM modeli, stok özellikleri için geliştirilmiş SLSTM modeli ve ana bileşene dayalı PCA-S-LSTM modeli sırasıyla eğitilmiştir. Deneysel sonuçlar, PCA-S-LSTM modelinin yalnızca ortalama tahmin hatasını azaltmakla kalmayıp aynı zamanda çalışma süresini de büyük ölçüde azalttığını ve tahmin kararlılığını iyileştirdiğini göstermektedir.

Kilimci vd. 2021 çalışmasında, günlük ticaret olarak bilinen kısa vadeli veya sık alım satım işlemlerinde BTC dolar kurunun son fiyatını tahmin etmeyi amaçlamışlardır. Maksimum, minimum ve ortalama fiyatlar gibi istatistiksel göstergelerin yanı sıra Bollinger bandı (BB), saat bazlı hareketli ortalama (MA), Göreceli Güç Endeksi (RSI) gibi teknik göstergeler de bir özellik seti olarak değerlendirmişlerdir. Bitcoin'in fiyatını tahmin etmek için farklı derin ve hibrit derin öğrenme metodolojileri olan evrişimli sinir ağları (CNN'ler), uzun kısa süreli bellek ağları (LSTM'ler), evrişimli uzun kısa süreli bellek ağları (ConvLSTM), CNN Uzun Kısa Süreli Bellek Ağı (CNN-LSTM) kullanmışlardır. Deney sonuçları bitcoin fiyatını tahmin etmede ConvLSTM diğerlerinden daha iyi performans göstermiştir.

Sun ve Glabadanidis, 2022 çalışmasında, teknik göstergelerin Çin hisse senedi risk primi üzerinde önemli tahmin gücüne sahip olduğunu bulmuşlardır. Haftalık ve aylık göstergelere baktıklarında aylık düzeydeki teknik göstergelerin, Çin'in aylık hisse senedi risk primini tahmin etmede makroekonomik değişkenlere tamamlayıcı bilgiler sağladığını bulmuşlardır. Haftalık teknik göstergelerinin risk faktörlerini ve ayrıca piyasa kapitalizasyonu, getiri oynaklığı ve piyasa oranlarına göre sıralanan portföyleri azaltma konusunda güçlü tahmin gücüne sahip olduğunu da bulmuşlardır.

Dvoryatkina ve Golovin, 2022 çalışmasında, endüstriyel finansal araçların piyasasını tahmin etmişlerdir. Tahmin etmede analiz ve risk değerlendirmesi için yapay sinir ağlarının mimarisi olarak en güçlü tahmin modellerinden biri olan uzun süreli kısa süreli bellek hücresi LSTM'ye sahip tekrarlayan bir ağ seçilmiştir. 42 teknik analiz göstergesi kullanılmıştır. Deneyler sonucunda finansal aracın

maliyet göstergeleri için tahminin doğruluğuna ilişkin tahminler, oluşturulan sinir ağı modelinin yüksek verimliliğini göstermiştir.

Lee vd. 2022 çalışmasında, hisse senedi fiyatlarının tahmin edilmesinde BİLSTM ve teknik göstergeleri kullanmışlardır. Hisse senedi fiyatı tahmini için LSTM-dikkat zaman serisi modeline teknik analizleri uygulama konseptini öneren ilk araştırmadır. TPE:0050 nin geriye dönük testinde, deneysel sonuçlar %42,74 ile en yüksek yatırım getirisine ulaşmıştır. Bu çalışmada kullanılan teknik göstergeler ağırlıklı olarak hareketli ortalamalardır. Yazar, daha farklı teknik gösterge analizi türleri dahil edilebileceğini önermiştir.

Mohapatra vd. 2022 çalışmasında, Hindistan bankalarının hisse senedi getirilerini kullanarak teknik göstergelerin etkisini görmek amacıyla tahmin etmek için farklı makine öğrenme modelleri (Rastgele Ormana ek olarak XGBoost, Gradient Boosting ve AdaBoost) kullanmışlardır. Sonuçlar, XGBoost algoritmasının dört topluluk modeli arasında en iyi performansı gösterdiğini göstermektedir. Mutlak hatanın ortalaması ve ortalama karekök hatasının ortalaması %3-5 civarında olduğunu göstermektedir. Çeşitli teknik göstergeleri kullanan makine öğrenmesi teknolojilerinin tahmin gücünün yüksek olduğu sonucuna varmışlardır.

Liang ve Unwin, 2022 çalışmasında, günlük COVID-19 vakalarını mumlar açısından yeniden yorumlayarak, pandeminin seyri üzerinde tahmin gücü elde etmek için en popüler borsa teknik göstergelerinden MACD, RSI ve mum çubuğu analizlerinin niceliksel bir değerlendirmesini sağlayarak, bunların hem borsa verileri hem de COVID-19 verileri için tahminlerde bulunmadaki istatistiksel önemini vurgulamışlardır.

Erfanian vd. 2022 çalışmasında makine öğrenme yöntemlerini uygulayarak hem makroekonomik hem de mikroekonomik teoriler bağlamında BTC fiyat tahmini yapmışlardır. Bu makalede, ekonomik teorilere dayanan makroekonomik, mikroekonomik, teknik ve blockchain göstergelerinin tahmin edip etmediğini araştırmak için sıradan en küçük kareler (OLS), topluluk öğrenme, destek vektör regresyonu (SVR) ve çok katmanlı algılayıcı (MLP) dahil olmak üzere karşılaştırmalı yaklaşımlar uygulamışlardır. Bulgular, bazı teknik göstergelerin önemli kısa vadeli BTC fiyat tahminleri olduğuna işaret ediyor ve bu nedenle teknik analizin geçerliliğini doğruluyor. Genel bulgular, SVR'nin diğer makine öğrenmesi modellerinden ve geleneksel modellerden üstün olduğunu göstermektedir.

Yang vd. 2022 çalışmasında, geleneksel endekslere, gelişmekte olan göstergelere, emtialara ve altının tarihsel fiyat zaman serilerine göre altın fiyatlarını tahmin etmek için bir dizi makine öğrenme modelini araştırmaktadır. Altın fiyatını tahmin eden modelleri oluşturmak için üç makine öğrenme algoritması, Yapay Sinir Ağı (ANN), Uzun Kısa Süreli Bellek (LSTM) ve Destek Vektörü Regresyonu (SVR) uygulamışlardır. Bu araştırma için veri seti, ABD’de iki ana endeks (S&P 500 ve DJI), iki popüler kripto para birimi (BTC ve ETH), iki meta (gümüş ve ham petrol) içeren bir zaman serisidir. SVR modelinin diğer iki modelden daha iyi performans gösterdiği gözlemlenmiş ve sonucun kripto para birimlerine ait ek verilerin her üç model üzerinde de olumlu bir etkiye sahip olduğunu göstermişlerdir.

3.6 Makine Öğrenmesi ve MATLAB

MATLAB, büyük verileri işlemek için araçlar ve işlevlerin yanı sıra makine öğrenmesini erişilebilir kılmak için uygulamalarla, veri analitiğinize makine öğrenmesini uygulamak için ideal bir ortamdır.

Makine öğrenmesi temelde birkaç adımdan oluşur. Bu adımlar verilerin toplanması, veri ön işleme, özellik çıkarımıdır.

Makine öğrenmesine bağlı olarak kullanılacak veri bu adımda toplanır. Kullanılan algoritma, elimizdeki verinin özelliklerini analiz ederek öğrenir ve kendisine verilen bir sonraki veriyi tahmin eder. Bu yüzden bu aşamada ne kadar fazla ve doğru veri topladığımız çok önemlidir.

Makine öğrenmesi modellerini oluşturma sürecindeki kritik adımlardan biri, veri ön işleme aşamasıdır. Bu aşamada, veriyi temizleme, dönüştürme ve modele uygun hale getirme işlemleri gerçekleştirilir. Genellikle farklı kaynaklardan gelen veriler düzensiz bir şekilde bulunabilir, bu da modellerin tahmin performansını etkileyebilir. Bu nedenle, makine öğrenimi modellerini eğitmek, değerlendirmek ve kullanmadan önce ham verilerin düzenlenmesi önemlidir.

Algoritmaların operasyonel işleyişi genellikle karmaşık olabilir ve çoğu algoritmanın neden çalıştığına dair kesin teoriler bulunmayabilir. Bu nedenle, tahmin temelli modelleme projeleri genellikle deneysel bir yaklaşım gerektirir.

Günümüzde, makine öğrenmesi algoritmalarının otomatikleşmesi düşünüldüğünde, projeler arasındaki ana fark genellikle kullanılan veri setleridir.

Veri kalitesi, veri yönetiminin önemli bir sorunudur, çünkü kirli veriler genellikle yanıltıcı analiz sonuçlarına ve dolayısıyla yanlış iş kararlarına neden olabilir. Bu nedenle, projelerin büyük bir kısmı genellikle veri toplama, doğrulama, temizleme, görselleştirme, dönüştürme gibi veri işleme adımlarına odaklanır.

Veri setinin boyutsallığı, içerdiği giriş değişkenlerinin veya özelliklerin sayısını belirtir. Çoğu zaman, fazla özellik modelleme sürecini karmaşık hale getirebilir. Bu nedenle, veri görselleştirmesi için yüksek boyutsallık istatistikleri ve boyutsal azaltma teknikleri kullanılır. Ancak, bu teknikler sadece veri setini basitleştirmekle kalmaz, aynı zamanda sınıflandırma veya regresyon gibi uygulamalı makine öğreniminde daha etkili hale getirmek için de kullanılabilir.

Boyutsal küçültme yöntemleri, rastgele değişkenlerin sayısını azaltarak çalışır. Bu, istatistik, makine öğrenimi ve bilgi teorisi alanlarında yaygın olarak uygulanan bir tekniktir. Örneğin, sınıflandırma problemlerinde genellikle çok sayıda faktör bulunur. Bu faktörler genellikle özellikler olarak adlandırılan değişkenlerdir. Özellik sayısı arttıkça, eğitim setini görselleştirmek ve üzerinde çalışmak zorlaşır. Boyutsal küçültme algoritmaları, bu gereksiz özellikleri ortadan kaldırarak veriyi sadeleştirir ve daha anlamlı hale getirir, böylece daha etkili sonuçlar elde edilir. Bu yaklaşımların türleri başlıca; *özellik seçimi* (feature selection) ve *özellik çıkarma* (feature extraction) bileşenleri olarak ikiye ayrılmaktadır.

Makine öğrenmesi algoritmalarının performansı, çok sayıda giriş değişkeni ile olumsuz bir şekilde etkilenebilir. Veriyi temsil eden her bir sütunun bir boyutu ve veri satırlarının bu uzaydaki noktaları temsil ettiği düşünülürse, çok sayıda boyuta sahip bir özellik uzayı, geniş bir hacme sahip olacaktır. Ancak, bu geniş alandaki veri noktaları genellikle küçük ve temsil gücü düşük olacaktır, bu da veri setinin etkili bir şekilde temsil edilmediği anlamına gelir. Bu durum, bir veri setinin geometrik bir bakış açısıyla yararlı bir açıklama sunar.

Özellik seçimi (Feature Selection) yaklaşımları, giriş değişkenlerinin bir alt kümesini belirlemeyi amaçlar; bu değişkenler aynı zamanda özellikler veya öznitelikler olarak adlandırılır. Bu amaçla kullanılan başlıca üç yaklaşım şunlardır:

Filtre metodu (Filter method): Bilgi kazancını temel alır. Bu yöntemde, veri özellikleri, önceden belirlenmiş bir kriter kullanılarak filtrelendirir. Veri setinin ön işleme aşamasında, belirli özellikleri çıkarma veya seçme işlemi uygulanır.

Sarma metodu (Wrapper method): Doğruluk odaklı yönlendirilmiş bir arama prensibine dayanır. Farklı özellik küme kombinasyonları denenir ve her kombinasyonun performansı değerlendirilir. En etkili performans gösteren özellik kümesi seçilir.

Gömülü metodu (Embedded method): Seçilen özellikler, model tahminlerine dayanarak eklenir veya çıkarılır. Bu yaklaşım, veri analizi sürecinde regresyon veya sınıflandırma gibi modellerin, azaltmanın gerçekleştirildiği alanda, orijinal veri alanından daha doğru bir şekilde çalışmasını hedefler.

Bu yaklaşımların kullanılmasındaki temel amaçlar, model eğitim sürelerini kısaltarak hesaplama maliyetini düşürmek ve modelleri basitleştirerek araştırmacıların ve kullanıcıların anlamasını kolaylaştırmaktır. Ayrıca boyutsal karmaşıklık azaltarak modelin genelleştirilebilirliğini artırmak ve varyansı kontrol altında tutarak modelin performansını iyileştirmekte bu yaklaşımların amaçları arasındadır.

MATLAB’da Özellik Seçimi Fonksiyonları

fscchi2	chi-kare testleri kullanılarak sınıflandırma için tek
fscmrnr	değişkenli özellik sıralaması Minimum fazlalık maksimum alaka düzeyi (MRMR)
fscnca	algoritmasını kullanarak sınıflandırma için özellikleri sıralama
fscnca	Sınıflandırma için komşu bileşen analizini kullanarak
fscnca	özellik seçimi
fsrftest	F testleri kullanılarak regresyon için tek
	değişkenli
	özellik sıralaması

fsrmrmr	Minimum fazlalık maksimum alaka düzeyi (MRMR) algoritmasını kullanarak regresyon için sıralama özellikleri
fsrnca	Regresyon için komşu bileşen analizini kullanarak özellik seçimi
fsulaplacian	Laplacian puanlarını kullanarak gözetimsiz öğrenme için sıralama özellikleri
partialDependence	Kısmi bağımlılığı hesapla
plotPartialDependence	Kısmi bağımlılık grafiği (PDP) ve bireysel koşullu beklenti (ICE) grafikleri oluşturma
predictorImportance	Sınıflandırma ağacı için tahmin edici önemi tahminleri
relieff	ReliefF veya RReliefF algoritması kullanan tahmin edicilerin önem sıralaması
sequentialfs	Özel kriter kullanarak sıralı özellik seçimi
stepwiselm	Adım adım regresyon gerçekleştirin
stepwiseglm	Adım adım regresyon ile genelleştirilmiş doğrusal regresyon modeli oluşturma

Özellik çıkarma (Feature extraction), makine öğrenmesinde, örüntü tanıma ve görüntü işleme konularında özellik çıkarma süreci, mevcut veri kümesinden türetilmiş ve özgün bilgi içeren özelliklerin oluşturulmasıyla başlar. Bu aşama, sonraki öğrenme ve genelleme adımlarını kolaylaştırmayı hedefler.

Eğer bir algoritmanın karşılaştığı giriş verileri çok büyük ve gereksiz gibi görünüyorsa, bu verilerin daha yönetilebilir bir hale getirilmesi adına azaltılmış bir küme oluşturulabilir. Bu azaltılmış kümedeki seçilen özelliklerin, giriş verilerinden önemli bilgiler içermesi beklenir. Böylece, istenen görev, başlangıç verileri yerine bu daha özgün ve etkili gösterim kullanılarak gerçekleştirilebilir.

Boyutsal küçültmede kullanılan başlıca yöntemler Temel Bileşen Analizi (Principal Component Analysis), Doğrusal Diskriminant Analizi (Linear Discriminant Analysis) ve Doğrusal Ayrımcılık Analizi (Generalized Discriminant Analysis) dir.

Özellik çıkarmanın uygulamaları

Temel Bileşen Analizi (Principal Component Analysis): Verilerin düşük boyutlu bir temsili oluşturmak için kovaryans matrisi üzerinden öz vektörleri ve öz değerleri kullanır. Büyük öz değerlere sahip öz vektörler, verinin temsili sağlayan ana bileşenleri belirler.

Negatif Olmayan Matris Yaklaşdırması (Non-negative Matrix Factorization): Negatif olmayan iki matrisin çarpımı ile veriyi temsil eden negatif olmayan bir matris elde eder. Özellikle negatif olmayan sinyallerin olduğu alanlarda etkili bir boyutsal küçültme yöntemidir.

Çekirdek Temel Bileşen Analizi (Kernel PCA): Temel bileşen analizini çekirdek yöntemlerle genişleterek doğrusal olmayan yapıları çözer. Çekirdek kullanarak, doğrusal olmayan ilişkileri daha iyi ifade edebilir.

Grafik Tabanlı Çekirdek Temel Bileşen Analizi (Graph-based Kernel PCA): Manifold öğrenme tekniklerini içerir, verilerin yerel özelliklerini koruyan bir maliyet fonksiyonu kullanarak düşük boyutlu bir temsili oluşturur.

Doğrusal Diskriminant Analizi (Linear Discriminant Analysis): Sınıflar arası ayrımı artırmak için özellikleri doğrusal bir kombinasyon ile ifade eder. Veri setindeki sınıflar arasındaki farkı maksimize etmeyi amaçlar.

Doğrusal Ayrımcılık Analizi (Generalized Discriminant Analysis): Çekirdek fonksiyon operatörlerini kullanarak doğrusal olmayan ayırıcı analizlere odaklanır. Giriş vektörlerini düşük boyutlu bir alana eşlemek için kullanılır.

Otomatik Kodlayıcı (Autoencoder): Doğrusal olmayan boyut küçültme işlevlerini ve kodlamalarını öğrenmek için kullanılır. Kodlamadan orijinal veriye geri dönmeyi öğrenerek veriyi sıkıştırır ve daha az boyutlu bir temsili elde eder.

T-dağıtımlı Stokastik Komşu Gömme (t-SNE): Yüksek boyutlu veri kümelerini görselleştirmek için kullanılan doğrusal olmayan bir tekniktir. Benzer özelliklere sahip veri noktalarını yakın bir uzaya yerleştirir.

Düzgün Manifold Yaklaşımı ve İzduşümü (Uniform Manifold Approximation and Projection): Doğrusal olmayan boyutsal küçültme yöntemidir. Verilerin lokal bağlantılarını ve eşit dağılımını dikkate alarak düşük boyutlu bir temsili oluşturur.

Boyutsal küçültmenin avantajı, gerekli zamanı ve depolama alanını azaltarak işlemleri hızlandırır. Çoklu eşlenimsizliğin (multi-collinearity) azalması, makine öğrenme modelinin parametrelerinin daha anlamlı bir şekilde yorumlanmasını sağlar. Ayrıca düşük boyutlarda görselleştirme imkanı sunarak karmaşık veri yapılarını daha anlaşılır hale getirir ve model performansını artırır.

Boyutsal küçültmenin dezavantajı, veri kaybına neden olabilir. PCA, değişkenler arasında bazen istenmeyen doğrusal korelasyonlar bulma eğilimindedir. Ortalama ve kovaryansın veri setini yeterince tanımlayamadığı durumlarda başarılı olmayabilir. Ayrıca hangi ana bileşenlerin tutulması gerektiği pratikte belirsiz olabilir.

Örnek 2 *İyonosfer atmosferdeki elektromanyetik dalgaları yansıtan katmandır. Lonosphere veri setinde hedefler iyonosferdeki serbest elektronlardır. "İyi" radar dönüşleri, iyonosferdeki bir tür yapının kanıtını gösterenlerdir. "Kötü" getiriler, olmayanlardır; yani sinyalleri iyonosferden geçer. Veri setinde etiketler kötü radar dönüşleri için 'b', iyi radar dönüşleri için 'g' kullanılmıştır.*

NCA, komşu bileşen analizi, kullanılarak özellik seçimi MATLAB kodları ve çalıştırıldığında elde edilen sonuç aşağıda verilmiştir.

```
1 close all; clear; clc;
2 cd ../Data
3 load('ionosphere.mat')
4 cd ../ornek_8
5 mdl = fscnca(X,Y)
6 figure()
7 plot(mdl.FeatureWeights,'ro')
8 xlabel('Feature Index')
9 ylabel('Feature Weight')
10 grid on
11 figure
12 plot(mdl.FitInfo.Iteration,mdl.FitInfo.Objective,'ro-')
13 grid on
14 xlabel('Iteration Number')
15 ylabel('Objective')
16 mdl.ModelParameters.Standardize
17 XTest = X(1:100,:);
18 XTrain = X(101:end,:);
```

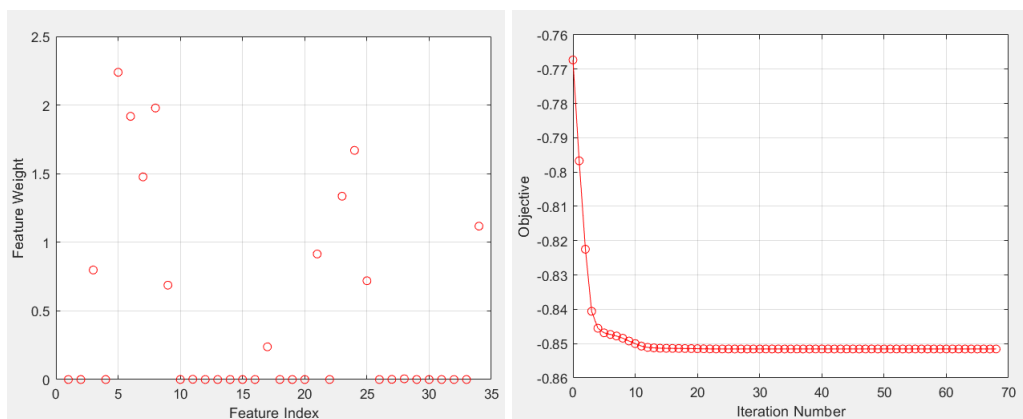
```
1 >> mdl = fscnca(X,Y)
2 mdl =
```

```

3 FeatureSelectionNCAClassification
4 NumObservations: 351
5 ModelParameters: [1x1 struct]
6 Lambda: 0.0028
7 FitMethod: 'exact'
8 Solver: 'lbfgs'
9 GradientTolerance: 1.0000e-06
10 IterationLimit: 1000
11 PassLimit: 5
12 InitialLearningRate: []
13 Verbose: 0
14 InitialFeatureWeights: [34x1 double]
15 FeatureWeights: [34x1 double]
16 FitInfo: [1x1 struct]
17 Mu: []
18 Sigma: []
19 X: [351x34 double]
20 Y: {351x1 cell}
21 W: [351x1 double]
22 ClassNames: {2x1 cell}
23 Properties, Methods
24 >> mdl.ModelParameters.Standardize
25 ans =
26 logical
27 0

```

Yukarıdaki komutların grafiksel yorumuda aşağıdaki gibidir.



Şekil 3.5 NCA

PCA, temel bileşen analizidir. Boyutsal küçültme için ana doğrusal tekniklerden biridir. İyonosfer veri seti kullanılarak verilerin ilk 100 ü eğitim ve

kalan kısmı test için ayrılmıştır. Temel bileşen analizinde pca fonksiyonu kullanılmıştır. Özellikler X tablosunda ve etiketler Y tablosundadır. PCA 21 özellik seçti ve bu özellikler kullanılarak modelin eğitim ve test bulgularına göre tahmin yapılmıştır. MATLAB kodları ve çalıştırıldığında elde edilen sonuç aşağıda verilmiştir.

```

1 YTest = Y(1:100);
2 YTrain = Y(101:end);
3 [coeff,scoreTrain,~,~,explained,mu] = pca(XTrain);
4 explained
5 idx = find(cumsum(explained)>95,1)
6 scoreTrain95 = scoreTrain(:,1:idx);
7 mdl = fitctree(scoreTrain95,YTrain);
8 scoreTest95 = (XTest-mu)*coeff(:,1:idx);
9 YTest_predicted = predict(mdl,scoreTest95);

```

```

1  >> explained
2  explained =
3  36.1778
4  12.3137
5  8.7629
6  7.1800
7  4.1942
8  3.2902
9  2.8312
10 2.4970
11 2.3688
12 2.1179
13 1.9137
14 1.6768
15 1.5424
16 1.5300
17 1.3914
18 1.2055
19 1.0866
20 0.9676
21 0.9161
22 0.7583
23 0.7161
24 0.6578
25 0.6308
26 0.5526
27 0.4990
28 0.4281
29 0.3741
30 0.3245
31 0.2890

```

```

32 0.2699
33 0.2166
34 0.1761
35 0.1432
36 0.0000
37 >> idx = find(cumsum(explained)>95,1)
38 %ilk iki bileşeni kullanarak bir sınıflandırma ağacı eğitilmesi:
39 scoreTrain95 = scoreTrain(:,1:idx);
40 mdl = fitctree(scoreTrain95,YTrain)
41 idx =
42 21
43 mdl =
44 ClassificationTree
45 ResponseName: 'Y'
46 CategoricalPredictors: []
47 ClassNames: {'b' 'g'}
48 ScoreTransform: 'none'
49 NumObservations: 251

```

Toplanan veri seti, ham formundan işlenebilir bir hale getirildikten sonra, öğrenme modeli için kullanılmak üzere %80 ve değerlendirme amacıyla ise %20 oranında ayrılmalıdır. Bu veri bölünmesinin temel amacı, öğrenme algoritmasında kullanılan verinin aynı set içinde değerlendirme için kullanılması durumunda, elde edilen sonuçların adil olmayan ve gerçeği yansıtmayan sonuçlar üretebilme olasılığını en aza indirmektir.

Veriyi tercihe ve uygunluğuna göre ayrılmak istenilen yüzde değişebilir. Aşağıda eğitim için verinin %80 ni kalan %20 si de testte kullanılmak üzere ayrılmıştır. Elimizde 100 veri varsa bunun 80 ni eğitim için kalan 20 si test içindir. Tabloda Özellikler kolonu giriş verileridir. Örneğin diyabet hastalarının hasta veya sağlıklı olma durumu tahmin edilmek istensin. Burada diyabet hastalarının birtakım *özellikleri* kan şekeri, yaş cinsiyet gibi, hasta yada sağlıklı olma durumu ise *etikettir*.

Eğitim Matrisi(%80)					
	Özellikler(Giriş)(X)			Etiket (Çıkış)(Y)	Yapay Zeka(Çıkış)
Özellik Numarası	1	2	...	100	
1	Eğitim			Eğitimc	Eğitims
2					
...					
80					

Test Matrisi(%20)						
	Özellikler(Giriş)				Etiket (Çıkış)	Yapay Zeka(Çıkış)
Özellik Numarası	1	2	...	100		
1						
2						
...						
20						

MATLAB üzerinde Giriş verilerine X , Gerçek çıkış verileri Y olarak kabul edilir. Veri tahmininde bulunurken yapay zekanın ilk olarak eğitilmesi gerekir. 100 verinin 80 ni ile eğitime giren yapay zeka daha sonra eğitim için kullanılan verilerle tahmin eder. Tahmin sonuçlarını MATLAB da `Eğitims` değişkenine atılır. Test için kullanılan verilerin tablodaki gerçek değerleri MATLAB da `Testc` dir. Test verilerinin yapay zeka tarafından tahmin değerleri MATLAB da `Tests` değişkenidir.

3.6.1 Sistemik Örneklem Teoremi

MATLAB da eğitim ve test verilerinin ayırma yöntemlerinden birisi *Sistemik Örneklem Teoremi*dir. Rastgele belirlenen başlangıç noktasından itibaren her n . üye örneklem içerisine dahil edilir. Örneğin elimizdeki veri üzerinden 1,6,11,.. numaralı kişi özellikleri sistemik örneklem yöntemi kullanılarak eğitim ve test verileri oluşur. Sistemik örneklem, maliyet ve zaman bakımından daha avantajlıdır. Algoritmanın aşamaları adım adım aşağıda verilmiştir.

- Örneklemdeki eleman sayısının evrendeki eleman sayısına oranı (k) hesaplanır.
- Daha sonra evrendeki elemanlar sıraya dizilir ve bu orana göre sıra numarası verilir.
- Verilen sıra numarasına göre başlangıçtan itibaren her $1/k$ 'ninci eleman örneklem alınır.
- Örneğin 1000 birimden oluşan bir evrenden 100 birimlik bir örneklem oluşturulmak isteniyorsa; ($k = 1000/100 = 10$) sabit değeri bulunur.

- Tamamen rastgele olarak 1 ile 10 arasında birinci örnek için bir rakam belirlenir.
- Bu rakama 10 eklenerek ikinci örnek ve son bulunan rakama da yine 10 eklenerek üçüncü örnek bulunur.
- İşlem bu şekilde son örneği seçene kadar devam eder.
- İlk belirlenen rakamı 3 kabul edersek 3-13- 23- 33- 43- 53 ve nihayet 993 numaralı birimler örnek olarak seçilirler.

Aşağıda MATLAB üzerinde sistematik örnekleme teoremi kodları verilmiştir.

```

1 function [egitim,egitimc,test,testc] = Orneklem(data,ornekyuzde)
2 N = size(data,1);
3 n = floor(N*ornekyuzde/100) ;
4 d = round(N/n) ;
5 a = 1 ;
6 s = size(data,2) ;
7 for i=1:n
8     B=(a+d*(i-1));
9     if (B>N)
10         C=B-N;
11     else
12         C=B;
13     end
14     test1(i,:) = data(C,:) ;
15 end
16 test = test1(:,1:s-1) ;
17 testc = test1(:,s) ;
18 egitim1 = data ;
19 for i=1:n
20     silme(i) = (a+d*(i-1)) ;
21 end
22 silme=sort(silme,'descend');
23 for i=1:n
24     k = silme(i) ;
25     egitim1(k,:) = [] ;
26 end
27 egitim = egitim1(:,1:s-1) ;
28 egitimc = egitim1(:,s) ;
29 end

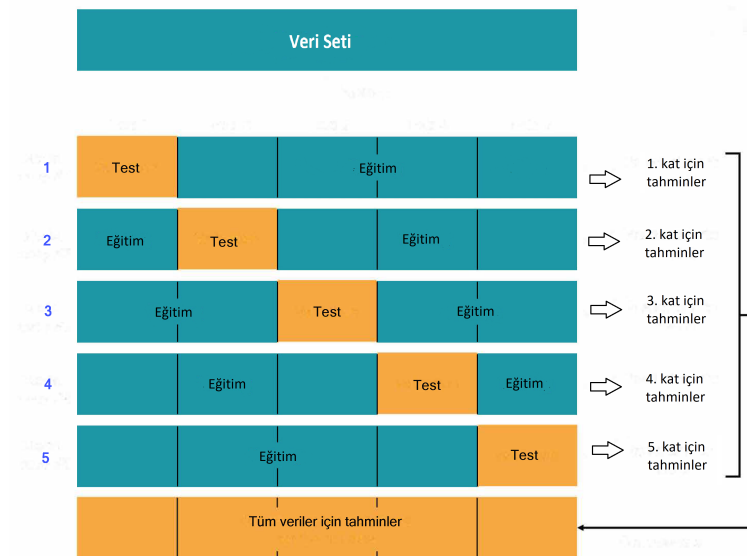
```

Sistemantik örnekleme kodları MATLAB da fonksiyonlarda oluşturulmuştur. Bu fonksiyonun MATLAB üzerinde eğitim ve test matrisi olarak çağırılması aşağıda verilmiştir.

```
1 close all; clear; clc;
2 cd ../Data
3 load('Data1.mat')
4 cd ../ornek_1
5 disp('Verilerin Dagilimi')
6 tabulate(Data1(:,end))
7 yuzde=80;
8 [egitim,egitimc,test,testc] = Orneklem(Data1,yuzde);
9 disp('Egitim Veri Dagilimi')
10 tabulate(egitimc)
11 disp('Test Veri Dagilimi')
12 tabulate(testc)
13 save Egitim_Test
```

3.6.2 K-Katlamalı Çapraz Doğrulama (Kfold Cross Validation):

Aşırı öğrenme, bir algoritmanın eğitim verilerine aşırı derecede uyum sağlaması ve bu verileri ezberleyerek, yalnızca eğitim setinde iyi performans göstermesi durumunu ifade eder. Ancak, aynı modelin test verileri üzerinde uygulanması sonuçların beklenenden düşük olmasına neden olabilir. Bu olguyu engellemek ve güvenilir bir model oluşturmak için, verinin tüm parçalarının eğitim ve test verisinde yer alması önemlidir.



Şekil 3.6 k-Fold

K-faktörlü çapraz doğrulama yöntemi, veriyi eğitim ve test verisi olarak bölerek modelin genel performansını değerlendirmek için kullanılır. Veri seti k eşit parçaya ayrılır, her bir parça sırayla test verisi olarak kullanılırken diğer parçalar eğitim verisi olarak hizmet eder. Bu işlem k defa tekrarlanır ve her seferinde farklı bir parça test verisi olarak seçilir. Bu yöntem, modelin farklı veri kesimlerindeki performansını değerlendirerek aşırı öğrenme durumlarını daha etkili bir şekilde belirler. Yöntemin görselleştirilmesi **Şekil 3.6** de verilmiştir. K değeri genellikle 3 ya da 5 olarak seçilmektedir. Bu değer 10 ya da 15 de seçilebilir ancak bu oldukça pahalı bir hesaplama ve zaman kaybına sebep olacaktır. Verilerin kfold ile 5 alt kümeye ayrılmış MATLAB komutları aşağıda verilmiştir.

```
1 close all; clear; clc;
2 cd ../Data
3 load('Data3.mat')
4 cd ../ornek_6
5 disp('Verilerin Dagilimi')
6 tabulate(Data3(:,end))
7 data_son=categorical(Data3(:,end));
8 pt=cvpartition(data_son,"KFold",5)
9 save Egitim_Test2
```

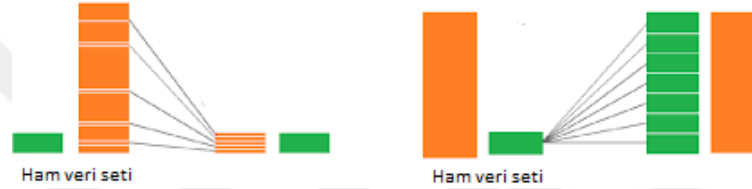
3.6.3 Dengesiz veri setleri ve SMOTE

1990'larda makine öğrenmesi ve veri madenciliğine ilişkin daha fazla veri ve uygulama yaygınlaşmaya başladıkça, önemli ölçüde çarpık sınıf dağılımlarına sahip verilerle uğraşırken istenen sınıflandırma doğruluğunun nasıl elde edileceği sorunu ortaya çıktı (He ve Garcia, 2009; López vd. 2013; Provost, 2000; Zewdu, 1998). Dengesiz veri setleri, sınıf dengesizliği adı verilen bir sorunu beraberinde getirir ve bu durum, sınıflandırma algoritmalarıyla uğraşan veri bilimi projelerinde dikkat edilmesi gereken bir konudur. Genellikle sınıflandırma algoritmaları, eğitim setlerinin dengeli olduğunu varsayar, ancak gerçek hayatta bu durum sıkça göz ardı edilir. Sınıf dengesizliği, bir sınıfın çok az örnekle temsil edilirken, diğer sınıfın ise çok sayıda örnekle temsil edildiği durumları içerir. Bu durum, sınıflandırma algoritmaları için ciddi sorunlara yol açabilir, çünkü model, genellikle az örneklenmiş sınıfın etiket bilgisi eksik olduğu için yeterince eğitilemez ve bu durum hatalı tahminlere neden olabilir. Yöntemin görselleştirilmesi **Şekil 3.7** da verilmiştir.

Bu sorunun üstesinden gelmek için geliştirilen stratejilerden biri, veriyi yeniden örnekleyerek sınıf dağılımlarını dengelemektir. Ancak bu stratejilerin kendine özgü avantajları ve dezavantajları bulunmaktadır.

Eksik örnekleme, çoğunluk sınıfına ait örnekleri çıkararak veri setini dengelemeyi amaçlar. Ancak, bu yöntemin bir dezavantajı, gözlem sayısının az olduğu durumlarda projeye katkı sağlayabilecek gözlemleri de veri setinden çıkarmasıdır. Ayrıca, az sayıda gözlem olduğunda örnek uzayın rastgeleliği zarar görebilir, bu da dengesizlik sorununu çözmek yerine yeni sorunlar oluşturabilir.

Aşırı örnekleme, azınlık sınıfına ait örnekleri çoğaltarak eşit sınıf dağılımları elde etmeyi amaçlar. Ancak, bu yöntemlerin çoğu azınlık sınıfının örneklerini kopyalayarak çalıştığı için aşırı öğrenme olasılığı artabilir. Ayrıca, büyük ve yüksek düzeyde dengesiz veri kümeleri için aşırı örnekleme, hesaplama açısından maliyetli olabilir. Bu nedenle, dengesiz veri setleriyle çalışırken doğru stratejiyi seçmek önemlidir.



Şekil 3.7 Oversampling ve Undersampling

Eksik ve aşırı örnekleme ek olarak, “gelişmiş örnekleme teknikleri” olarak adlandırılan, dağılımların yeniden dengelenmesinde keşifsel yöntemler kullanan başka örnekleme yöntemleri de vardır. Bunlardan biri tezimde kullandığım sentetik azınlık aşırı örnekleme (SMOTE) yöntemidir. SMOTE (Synthetic Minority Over-Sampling Technique), sentetik veri üretilmesini sağlayan bir aşırı örnekleme sürecidir (Chawla vd. 2002). Veri bilimi projelerinde en sık kullanılan yöntemlerden biridir. Yöntemin ana fikri, azınlık sınıfının örnekleri arasında belirli işlemler yaparak yeni azınlık sınıfı örnekleri yaratmaktır.

SMOTE, sınıf dengesizlikleri olan veri kümeleri ile sentetik verilerin üretilmesini içeren bir aşırı örnekleme sürecidir. SMOTE’un ana fikri, azınlık sınıfının çeşitli örnekleri arasında enterpolasyon yaparak azınlık sınıflarının yeni örnekler oluşturmaktır (Pat vd. 2022). Ayrıca SMOTE, regresyon görevlerini yerine getirmek için de kullanılabilir (Y. Zhang ve Xu, 2020).

SMOTE algoritması uygulanırken ilk olarak, bir azınlık sınıfı seçilir ve ardından K ’nin varsayılan olarak 5’e ayarlandığı K -en yakın komşuları elde edilir.

Tüm K komşuları azınlık örnekleri olmalıdır. Daha sonra enterpolasyon yoluyla yeni örnekleri hesaplamak için bu K komşu örneğinden biri rastgele seçilir. Enterpolasyon, incelenmekte olan azınlık sınıfı örneği (E_i) ile alınan seçilen komşu arasındaki fark hesaplanarak gerçekleştirilir. Bu fark, 0 ile 1 arasında düzgün bir şekilde çizilen rastgele bir sayı (δ) ile çarpılır ve elde edilen örnek, dikkate alınan azınlık sınıfına eklenir (Pat vd. 2022). Aslında, bu prosedür gözlemleri çoğaltmaz, ancak enterpolasyon, özellikler arasındaki “doğru parçası” boyunca rastgele bir noktanın seçilmesine neden olur (So ve Boucher, 2021).

$$E_{yeni} = E_i + (E_i - E_j) \times \delta \quad (3.23)$$

Gereken aşırı örnekleme miktarına bağlı olarak, en yakın k komşudan komşular rastgele seçilir. Bu yöntem, aşırı öğrenme sorununu engellemeye yönelik bir çözüm sunar ve sınıflandırma performansını artırabilir.

Örnek 3 SMOTE tekniğinin MATLAB kodları aşağıda verilmiştir.

```
1 function [X,C,Xn,Cn] = smote(X, N, k, options)
2 if isfield(options,'Class')
3 C = options.Class;
4 else
5 C = ones(size(X,1),1);
6 end
7 uC = unique(C);
8 nC = groupcounts(C);
9 if isempty(N)
10 if numel(uC)<2
11 error('Class vector must contain at least 2 classes to balance.');

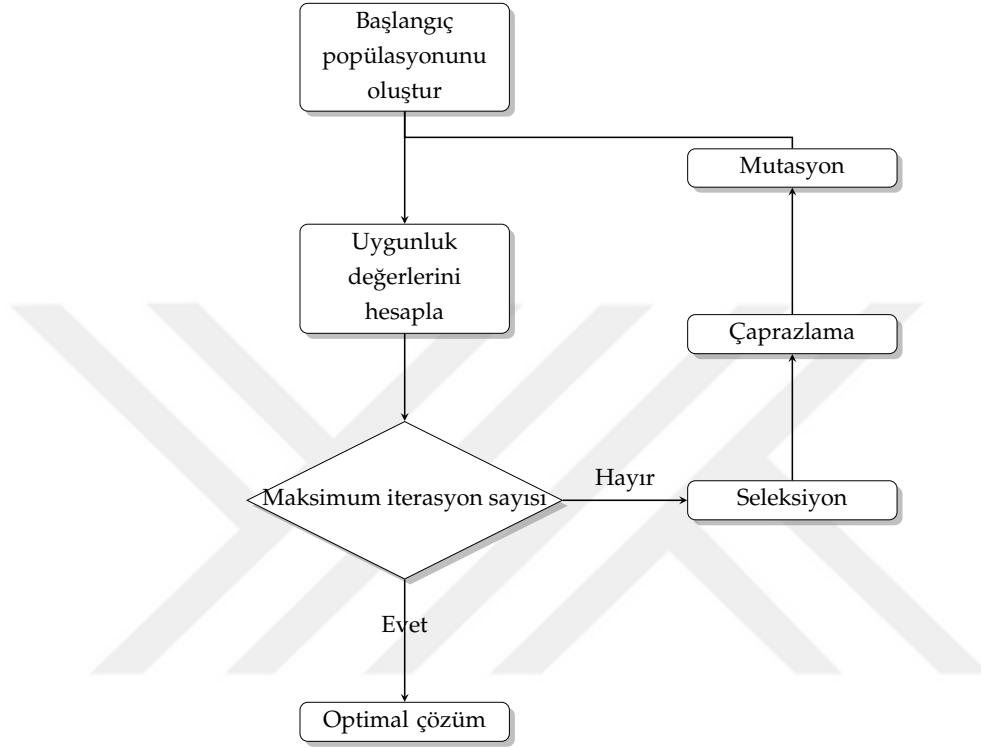
```

```

23 end
24 if isfield(options,'SynthObs')
25 J = options.SynthObs;
26 else
27 J = nan(size(X,1),1);
28 for ii=1:numel(uC)
29 iC = find(C==uC(ii));
30 P = randperm(nC(ii),round(rem(N(ii),1)*nC(ii)));
31 J(iC) = ones(nC(ii),1)*floor(N(ii));
32 J(iC(P)) = J(iC(P))+1;
33 end
34 end
35 Xn = [];
36 Cn = [];
37 for ii=1:numel(uC)
38 iC = C==uC(ii);
39 if sum(J(iC))>0
40 Xnn = simpleSMOTE(X(iC,:),J(iC),k(ii));
41 Xn = [Xn;Xnn];
42 Cn = [Cn;repmat(uC(ii),size(Xnn,1),1)];
43 end
44 end
45 X = [X;Xn];
46 C = [C;Cn];
47 end
48 function Xn = simpleSMOTE(X,J,k)
49 [idx, ~] = knnsearch(X,X,'k',k+1);
50 Xn = nan(sum(J),size(X,2));
51 for ii=1:numel(J)
52 P = randperm(k,J(ii))+1;
53 for jj=1:J(ii)
54 x = X(idx(ii,1),:);
55 xk = X(idx(ii,P(jj)),:);
56 Xn(sum(J(1:ii-1))+jj,:) = (xk-x)*rand+x;
57 end
58 end
59 end
60 function mustBeSameSize(a,b)
61 if ~isequal(size(a,1),size(b,1))
62 error('Must have the same number of elements as number of observations (
    rows) in X.');
```

Genetik algoritma

Genetik algoritma (GA), oldukça paralel bir şekilde çalışan matematiksel bir yöntemdir. Her biri karşılık gelen uygunluk değerine sahip bir grup bireyi alır ve yeni nesil olarak bilinen yeni bir popülasyon oluşturmak için üreme, mutasyon ve çaprazlama dahil olmak üzere çeşitli operatörler uygular (Arqub ve Abo-Hammour, 2014). Genetik Algoritma çalışma prensibi aşağıdaki gibidir.



Şekil 3.8 Genetik algoritma çalışma prensibi

Algoritma, bir toplum adını taşıyan ve kromozomlar aracılığıyla temsil edilen bir çözüm kümesi ile başlar. Toplumdaki çözümler, yeni toplumların oluşturulmasında kullanılmaktadır. Bu süreç, yeni toplumun öncekilerden daha etkili olacağı umuduyla gerçekleştirilir. Yeni çözümler, uygunluklarına göre seçilir ve daha uygun olanlar, tekrar üretilmek üzere daha fazla fırsata sahip olur. Bu süreç, belirli bir durumun karşılanmasına kadar tekrar edilir, örneğin belirli bir toplum sayısı veya en iyi çözümün gelişimi gibi.

Hyperparameter optimizasyonu; makine öğrenmesi süreçlerinde en uygun modeli bulmamıza yardımcı olur. Örnek olarak, sınıfların sayısını belirlemede veya veri setinin hangi oranda test-train olarak bölünebileceğinin kararında kullanılır. Kısaca en iyi modelin bilgilerini bulmak için kullanılır.

3.6.4 Regresyon

Bu bölümde MATLAB kullanılarak bazı regresyon türleri için hesaplama örnekleri verilecektir. Eğitim için ayrılan özelliklerimizin bulunduğu matris `egitim`, test için ise `test` olarak adlandırılmıştır. Özelliklerimize karşılık gelen etiketlerimiz ise `egitimc` ve `testc` dir. Bu örneklerde elde edilen modelin başarısını değerlendirmek için Sayfa 17 Bölüm 3.1.1 da verilen metrikler kullanılmıştır. Bu metrikleri hesaplayabilmek için aşağıdaki MATLAB fonksiyonu kullanılmaktadır.

MATLAB üzerinde performans değerlendirme

```

1 function PDK = Performans(testc,tests)
2 L=size(testc);
3 if L(1)==1
4     testc=testc';
5     tests=tests';
6 end
7 %% Hata Orani - MAPE - Mean Absoulute Percentage Error
8 % t - Target - Gercek Deger
9 % y - Cikis - Simulasyon Degeri
10 n=size(testc,1);
11 Toplam =0;
12 for i=1:n
13     t=testc(i,1);
14     y=tests(i,1);
15     Toplam_Ara=abs((t-y)/t)*100;
16     Toplam = Toplam + Toplam_Ara;
17 end
18 clearvars t y
19 MAPE=Toplam/n;
20 %% RMSE - Root Mean Square Error
21 RMSE_Toplam=0;
22 for i=1:n
23     t=testc(i,1);
24     y=tests(i,1);
25     RMSE_Ara=(t-y)^2;
26     RMSE_Toplam = RMSE_Toplam + RMSE_Ara;
27 end
28 clearvars t y
29 RMSE=sqrt(RMSE_Toplam/n);
30 %% MAD - Mean Absolute Difference
31 MAD_Toplam=0;
32 for i=1:n
33     t=testc(i,1);
34     y=tests(i,1);

```

```

35     MAD_Ara=abs(t-y);
36     MAD_Toplam = MAD_Toplam + MAD_Ara;
37 end
38 clearvars t y
39 MAD=(MAD_Toplam/n);
40 %% ST - Standart Hata
41 ST_Toplam=0;
42 for i=1:n
43     t=testc(i,1);
44     y=tests(i,1);
45     ST_Ara=(t-y)^2;
46     ST_Toplam = ST_Toplam + ST_Ara;
47 end
48 clearvars t y
49 SH=sqrt(ST_Toplam/(n-2));
50 %% R - Spearman Korelasyon Katsayisi
51 t=testc;
52 y=tests;
53 [R,P] = corr(t,y,'Type','Spearman');
54 clearvars t y
55 %% R^2 - Aciklayicilik Korelasyon Katsayisi
56 % R^2
57 %% MSE - Mean Square Error
58 MSE_Toplam=0;
59 for i=1:n
60     t=testc(i,1);
61     y=tests(i,1);
62     MSE_Ara=(t-y)^2;
63     MSE_Toplam = MSE_Toplam + MSE_Ara;
64 end
65 clearvars t y
66 MSE=MSE_Toplam/n;
67 %% Performans Degerlendirme Kriterleri
68 PDK=[MAPE RMSE MAD SH R R^2 MSE]';

```

Örnek 4 Veri seti, bir öğrencinin ders çalışmaya ayırdığı zamanın sınav sonuçlarına etkisini içerir. Model, öğrencinin çalışma süresine göre alabileceği puanın tahmin edilmesi üzerinedir. Toplam 500 verinin olduğu bu veri setinde giriş değişkeni çalışma süresi ve etiketimiz sınav sonuçları olarak belirlenmiştir. Bu veri seti üzerinde lineer regresyon modeli kurmak için `fitlm` metodu kullanacağız.

```

1 close all; clear; clc;
2 load('Egitim_Test.mat')
3 mdl=fitlm(egitim,egitimc)

```

```

4 egitims = predict(mdl, egitim);
5 E_D_C=[egitimc, egitims];
6 tests = predict(mdl, test);
7 T_D_C=[testc, tests];
8 plot(mdl)

```

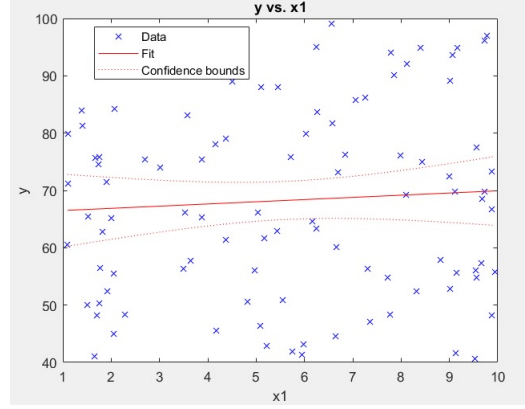
Eğitim ve test kümelerini parçalamak için Sayfa 49 Bölüm 3.6.1 de verilen sistematik örnekleme yöntemi ile verilerin %80 i eğitim, kalanı test için ayrılmıştır. Modelin fonksiyonu aşağıda gösterildiği gibidir ve bu fonksiyon üzerinden model eğitilmiştir. Model, predict fonksiyonu ile yanıtların tahmini yapılmıştır. Modelin başarısını test etmek için Estimated Coefficients değerleri incelenebilir. Bu değerlerin açıklamaları Sayfa 60 Bölüm 3.6.4 dedir. R^2 ve AdjRsquared değerlerine baktığımızda verilerin regresyon eğrisine uzak kaldığı ve modelin bu veriler üzerinde iyi bir sonuç çıkarmadığı görülmüştür.

```

1 mdl =
2 Linear regression model:
3 y ~ 1 + x1
4 Estimated Coefficients:
5           Estimate      SE      tStat      pValue
6           _____  _____  _____  _____
7
8 (Intercept)    66.114      3.7149     17.797    1.8272e-32
9 x1             0.38228     0.58738     0.65083     0.51668
10
11 Number of observations: 100, Error degrees of freedom: 98
12 Root Mean Squared Error: 16.8
13 R-squared: 0.0043, Adjusted R-Squared: -0.00586
14 F-statistic vs. constant model: 0.424, p-value = 0.517

```

Yukarıdaki komutların plot fonksiyonuyla görselleştirilmesi aşağıdaki gibidir.



Şekil 3.9 Lineer Regresyon

Bazı Model Fonksiyonları

Katsayı Tahminleri:

Coefficients : Katsayı değerlerini verir.

Estimate — Tahmini katsayı değeri

SE — Tahminin standart hatası

tStat — t -katsayının sıfır olduğu bir test için istatistiği

pValue — t istatistiği için p değeri

Komutun MATLAB üzerindeki çıktısı aşağıda verilmiştir.

```

1  >> mdl.Coefficients
2  ans=
3      2x4 table
4
5              Estimate      SE      tStat      pValue
6              _____      _____      _____      _____
7
8  (Intercept)      38.662      1.5641      24.719      3.4296e-75
9  x1      -0.00223893      0.0064858      -0.36839      0.71284

```

Özet İstatistik

DFE: Modeli eğitirken yaptığı hata sayısını verir.

Komutun MATLAB üzerindeki çıktısı aşağıda verilmiştir.

```
1 >> mdl.DFE
2 ans =
3 309
```

Diagnotics: Her satır ve sütun için satır teşhisi.

Giriş Verileri

NumObservations: Satır sayısını verir.

PredictorNames: Tahmin edicilerin adını verir.

ResponseNames: Yanıt değişkeninin adını verir.

Nesne Fonksiyonları

predict: Modelin yanıtlarını tahmin eder.

plot: Doğrusal regresyon modelinin dağılım grafiği veya eklenen değişken grafiğidir.

Örnek 5 Veri seti, uyku ve günlük alışkanlıklarla ilgili çok çeşitli değişkenleri kapsayan 400 satır ve 10 sütundan oluşur. Cinsiyet, yaş, uyku süresi, uyku kalitesi, fiziksel aktivite düzeyi, stres düzeyleri, BMI kategorisi (zayıf, normal, fazla kilolu), nabız, günlük adımlar, uyku bozukluklarının varlığı veya yokluğu parametrelerini içerir. Bu örnekte uyku ve günlük alışkanlıkların gün içindeki stres düzeylerine etkisi tahmin edilecektir. Stres düzeyleri 1-10 arasındaki değerlerden oluşmaktadır. Bu veri seti üzerinde lojistik regresyon modeli kurmak için *fitglm* modeli kullanacağız.

```
1 close all; clear; clc;
2 load('Egitim_Test4.mat')
3 mdl=fitglm(egitim,egitimc)
4 egitims = predict(mdl,egitim);
5 tests = predict(mdl,test);
6 plot(egitims)
7 plot(tests)
```

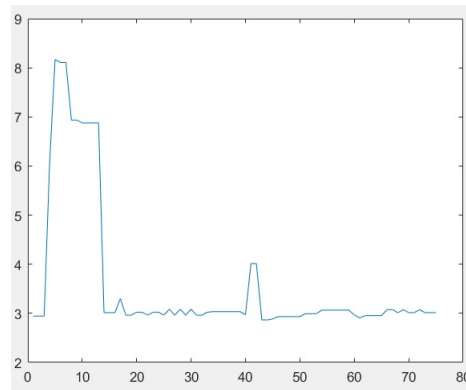
Eğitim ve test kümelerini parçalamak için Sayfa 49 Bölüm 3.6.1 de verilen sistematik örnekleme yöntemi ile verilerin %80 i eğitim, kalanı test için ayrılmıştır. Modelin fonksiyonu aşağıda gösterildiği gibidir ve bu fonksiyon üzerinden model eğitilmiştir. Model, predict fonksiyonu ile yanıtların tahmini yapılmıştır. Modelin başarısını test etmek için Estimated Coefficients değerleri incelenebilir.

```

1 mdl =
2
3 Generalized linear regression model:
4 y ~ 1 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9
5 Distribution = Normal
6 Estimated Coefficients:
7
8           Estimate          SE      tStat      pValue
9           _____          _____
10 (Intercept)              0              0          NaN          NaN
11 x1             -28.421         3.4344     -8.2754     9.4686e-12
12 x2              0.0099812      0.011961     0.83448     0.40707
13 x3             -0.60283        0.11422    -5.2776     1.613e-06
14 x4              1.1422         0.2008      5.688      3.3049e-07
15 x5             -0.10231        0.0096654   -10.585     8.8396e-16
16 x6             -0.36261        0.04211     -8.611      2.4074e-12
17 x7              0.80688        0.067946    11.875     6.1716e-18
18 x8              0.0010936      0.00011047   9.8995     1.3295e-14
19 x9              0.10814        0.027725     3.9005     0.0002302
20
21
22 75 observations, 66 error degrees of freedom
23 Estimated Dispersion: 0.00613
24 F-statistic vs. constant model: 3.18e+03, p-value = 3.12e-82

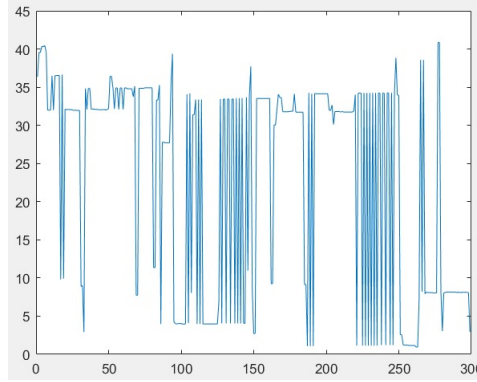
```

Eğitim sonuçlarının görselleştirilmesi aşağıda verilmiştir.



Şekil 3.10 Logistik Regresyon eğitim

Test sonuçlarının görselleştirilmesi aşağıda verilmiştir.



Şekil 3.11 Logistik Regresyon test

Örnek 6 Veri seti bir şirketteki çalışanların maaşları hakkında 375 veriden oluşan bilgi içermektedir. Her satırda farklı bir çalışan temsil ediliyor ve sütunlarda yaş, cinsiyet, eğitim düzeyi, deneyim yılı özelliklerimizi oluştururken maaşlar etiket değişkenini oluşturuyor. Bu örnekte çalışanların bahsedilen özelliklerinin maaşlarına etkisi tahmin edilecektir. Bu veri seti üzerinde çoklu doğrusal regresyon modeli kurmak için `fitlm` modeli kullanacağız. Lineer regresyondan farkı özelliklerin birden fazla olmasıdır.

```
1 close all; clear; clc;
2 load('Egitim_Test.mat')
3 X = egitim(:, :);
4 Y = egitimc(:, 1);
5 mdl=fitlm(X,Y)
6 plot(mdl)
```

Eğitim ve test kümelerini parçalamak için sistematik örnekleme yöntemi ile verilerin %80 i eğitim, kalanı test için ayrılmıştır. Modelin fonksiyonu aşağıda gösterildiği gibidir ve bu fonksiyon üzerinden model eğitilmiştir. Model, `predict` fonksiyonu ile yanıtların tahmini yapılmıştır. Modelin başarısını test etmek için Estimated Coefficients değerleri incelenebilir. R^2 ve AdjRsq değerlerine baktığımızda %98 başarılı bir model olduğu görülmüştür. Veriler regresyon eğrisi boyunca uyumludur.

```
1 mdl =
2 Linear regression model:
3 y ~ 1 + x1 + x2 + x3 + x4
4 Estimated Coefficients:
5
```

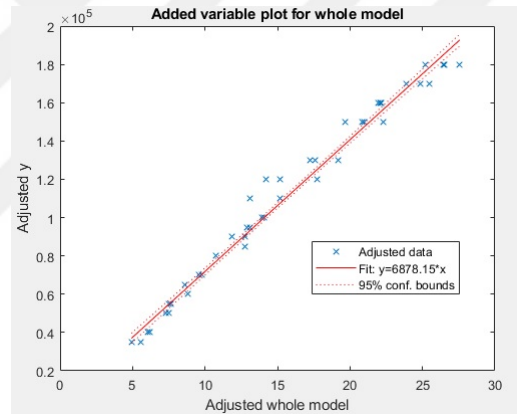
	Estimate	SE	tStat	pValue
--	----------	----	-------	--------

```

6
7
8 (Intercept)      3105      19796      0.15685      0.87582
9 x1              1057.6     712.51      1.4843      0.14221
10 x2             -1363.4     1597      -0.85373     0.39617
11 x3              301.01    1564.9      0.19235     0.84803
12 x4              6651.4     689.41      9.6479     1.7353e-14
13
14
15 Number of observations: 75, Error degrees of freedom: 70
16 Root Mean Squared Error: 6.37e+03
17 R-squared: 0.983, Adjusted R-Squared: 0.982
18 F-statistic vs. constant model: 1.02e+03, p-value = 3.03e-61

```

Yukarıdaki komutların plot fonksiyonuyla görselleştirilmesi aşağıdaki gibidir.



Şekil 3.12 Çoklu Doğrusal Regresyon

Örnek 7 Veri seti, 992 veriden oluşan bir işletmenin belirli bir ürünü hakkında bilgi içerir. Veri setinde ürünü tanıtan radyo reklamlarının sayısı özelliğimizi oluştururken ürünün haftalık satış sayısı etiket değişkenini oluşturuyor. İşletmenin reklam harcamalarının satışlarına etkisi tahmin edilecektir. Bu veri seti üzerinde doğrusal olmayan regresyon modeli kurmak için `fitnlm` modelini kullanacağız.

```

1 close all; clear; clc;
2 load('Egitim_Test3.mat')
3 X = egitim(:, :);
4 Y = egitimc(:, 1);
5 beta0 = [0, 0, 0];
6 modelfun = @(beta, x) beta(1) + beta(2) * x + beta(3) * x.^2;

```

```
7 mdl = fitnlm (X', Y, modelfun, beta0)
```

Eğitim ve test kümelerini parçalamak için sistematik örnekleme yöntemi ile verilerin %80 i eğitim, kalanı test için ayrılmıştır. Modelin fonksiyonu aşağıda gösterildiği gibidir ve bu fonksiyon üzerinden model eğitilmiştir. β_0 başlangıç değerini ifade eder ve ikinci dereceden polinom için 3 başlangıç değeri tanımladık. `modelfun` fonksiyonunda doğrusal olmayan regresyon modeli tanımladık ve `fitnlm` modeliyle eğittik. Modelin başarısını test etmek için `Estimated Coefficients` değerleri incelenebilir. R^2 ve `AdjRsqr` değerlerine baktığımızda oldukça düşük olduğundan modelin veri setindeki değişkenliği açıklamada zayıf olduğunu gösterir.

```
1 mdl =
2 Nonlinear regression model:
3 y ~ beta1 + beta2*x + beta3*x^2
4 Estimated Coefficients:
5           Estimate          SE          tStat          pValue
6           _____          _____          _____          _____
7
8 beta1      1.9383e+05          18594          10.424          1.5944e-20
9 beta2       -28.361           28.842          -0.98333          0.32666
10 beta3      0.0062286          0.0092042          0.67672          0.49938
11
12
13 Number of observations: 199, Error degrees of freedom: 196
14 Root Mean Squared Error: 8.64e+04
15 R-Squared: 0.0116, Adjusted R-Squared 0.00155
16 F-statistic vs. constant model: 1.15, p-value = 0.318
```

Örnek 8 Veri seti, 414 veriden oluşan gayrimenkul fiyatlarını ve bunlarla ilişkili özelliklerini içerir. Özellikler işlem tarihi, konut yaşı, en yakın MRT istasyonuna uzaklık, market sayısı, enlem, boylam ve etiketimiz konut fiyatlarını içerir. Bu örnekte gayrimenkulün bulunduğu konum ve çevresindeki özelliklere göre konut fiyatları tahmin edilecektir. Bu veri seti üzerinde GPR modeli kurmak için `fitrgp` modelini kullanacağız. MATLAB İstatistik ve makine öğrenmesi Araç Kutusu'nun "fitrgp" işlevi, GPR modelini eğitirken, verilerden elde edilen çekirdek işlevinin temel işlev katsayılarını β , gürültü varyansını σ^2 ve hiperparametrelerini θ tahmin eder (Mathworks, 2016).

```
1 cd ../Data
2 load('Data8.mat')
```

```

3 cd ../ornek_5
4 rng(0);Data8 = smote(Data8, 1, 5, 'Class');
5 X =Data8(:,1:6);
6 Y = Data8(:,end);

```

Önce, Sayfa 52 Bölüm 3.6.3 de verilen SMOTE tekniğini uygulayarak verileri iki katına çıkardık. 828 verideki özelliklerimizi X değişkeni, etiketimizi Y değişkeni olarak isimlendirdik.

```

1 hpartition = cvpartition(size(X,1),'Kfold',5);
2 sonuc = table();
3 for i=1:hpartition.NumTestSets
4 trainId = training(hpartition,i);
5 testId = test(hpartition,i);
6 X_train = X(trainId,:);
7 X_test = X(testId,:);
8 y_train = Y(trainId,:);
9 y_test = Y(testId,:);
10 rng(0);model = fitrgp(X_train,y_train,'KernelFunction','Exponential',...
11 'BasisFunction','pureQuadratic','FitMethod','sr','PredictMethod','exact
    ','Standardize',0);
12 t = model.predict(X_test);
13 p = size(X,2);
14 sonuc(i,:) = reg_basari_metrikleri(y_test,t,p);
15 end
16 disp(sonuc)
17 M = varfun(@mean, sonuc, 'InputVariables', @isnumeric);
18 disp(M)

```

GPR modelini kullanarak bu veri seti üzerinde regresyon performansını değerlendirmek için 5 katlı çapraz doğrulama(kfold) yöntemini uyguladık. Her iterasyonda eğitim ve test veri setleri oluşturuldu. GPR modeli bu eğitim verileri üzerinde eğitilip ardından test verileri kullanılarak tahminleme yapıldı. Her iterasyonda hesaplanan regresyon performans metriklerini, 'sonuc' tablosuna ekledik.

1 Ortalama R	:	0.90395
2 Ortalama MAE	:	3.3547
3 Ortalama MSE	:	33.028
4 Ortalama RMSE	:	5.6146
5 Ortalama R-Kare	:	0.81189
6 Ortalama AdjRsq	:	0.80478
7 Ortalama MAPE	:	9.8801

Yukarıda verilen performans metriklerinin sonuçlarına bakıldığında modelin başarılı bir regreyon modeli olduğu görülmüştür.

3.6.5 Sınıflandırma

Bu bölümde MATLAB kullanılarak bazı sınıflandırma algoritmaları için örnekler verilecektir. Eğitim için ayrılan özelliklerimizin bulunduğu matris `egitim`, test için ise `test` olarak adlandırılmıştır. Özelliklerimize karşılık gelen etiketlerimiz ise `egitimc` ve `testc` dir. Bu örneklerde elde edilen sınıflandırma algoritmalarının başarısını değerlendirmek için Sayfa 25 Bölüm 3.1.2 de verilen metrikler kullanılmalıdır. Bu metrikleri hesaplayabilmek için aşağıdaki MATLAB fonksiyonları kullanılır.

MATLAB üzerinde performans değerlendirme

Confusion Matrix (Karışıklık Matrisi): Gerçek ve tahmin edilen sınıfların sayılarını gösterir. Modelin performansını daha ayrıntılı bir şekilde değerlendirmek ve hata türlerini belirlemek için kullanılır. TP, TN, FP, ve FN değerlerini içerir.

```
1 C = confusionmat(egitimc, egitims);
2 hassasiyet=diag(C) ./ sum(C,1)';
3 recall=diag(C) ./ sum(C,2);
4 f1_score=2*hassasiyet.*recall./(hassasiyet+recall);
```

Classperf: Sınıflandırma performansını değerlendirmek için MATLAB'da kullanılan bir sınıflandırma performans nesnesi oluşturur. Özellikle, sınıflandırma performansını değerlendirmek ve bu değerleri kaydetmek için kullanılır. Hassasiyet(Sensitivity), Özgüllük(Specificity), Kesinlik(Precision) gibi performans metriklerini hesaplamak için kullanılır.

```
1 cp= classperf(egitimc, egitims);
2 dogruluk=cp.CorrectRate;
3 disp(['Dogruluk:', num2str(dogruluk)]);
4 duyarlilik=cp.Sensitivity;
5 disp(['Duyarlilik:', num2str(duyarlilik)]);
```

Confusionchart: Karmaşıklık matrisini ve modelin performansını daha açıklayıcı bir şekilde görselleştirmek amacıyla kullanılır.

```
1 figure;
2 confusionchart(C, {'Class -1', 'Class 1'}, 'Title', 'Confusion Matrix')
```

Perfcurve: ROC eğrisi oluşturmak ve AUC değerini hesaplamak için kullanılır. Sınıflandırma modelinin duyarlılık (sensitivity) ve özgüllük (specificity) gibi performans ölçümlerini görsel olarak değerlendirmek için kullanılır. AUC değeri, modelin sınıfları ayırt etme yeteneğini ölçer.

```
1 [~,scores]=predict(mdl,egitim);
2 labels = (egitimc + 1) / 2;
3 [X_ROC, Y_ROC, ~, AUC] = perfcurve(labels, scores(:, 2), 1);
4 figure;
5 plot(X_ROC, Y_ROC);
6 xlabel('False Positive Rate');
7 ylabel('True Positive Rate');
8 title(['ROC Curve (AUC = ' num2str(AUC) ')']);
9 grid on;
```

Örnek 9 Kişilerin diyabet hastası olma durumlarını belirlemek amacıyla 520 veriden oluşan 16 özellik ve kişinin diyabet hastalığı durumunu gösteren veri setini kullandık. Bu 16 özelliği kişilerin hasta olma durumlarını tahmin etmek için kullandık. Bu veri seti üzerinde SVM modeli kurmak için `fitcsvm` modelini kullanacağız.

```
1 close all; clear; clc;
2 cd ../Data
3 load('Data.mat')
4 cd ../ornek_1
5 disp('Verilerin Dagilimi')
6 tabulate(Data(:,end))
7 yuzde=80;
8 [egitim,egitimc,test,testc] = Orneklem(Data,yuzde);
9 disp('Egitim Veri Dagilimi')
10 tabulate(egitimc)
11 disp('Test Veri Dagilimi')
12 tabulate(testc)
13 save Egitim_Test
```

Eğitim ve test kümelerini parçalamak için sistematik örnekleme yöntemi ile verilerin %80 i eğitim, kalanı test için ayrılmıştır. Verilerin dağılımı aşağıda verilmiştir.

```
1 Verilerin Dagilimi
2 Value      Count      Percent
3 1          320        61.54%
4 2          200        38.46%
5 Egitim Veri Dagilimi
```

```

6 Value      Count      Percent
7 1          56         53.85%
8 2          48         46.15%
9 Test Veri Dagilimi
10 Value      Count      Percent
11 1          264        63.46%
12 2          152        36.54%

```

```

1 load('Egitim_Test.mat')
2 mdl=fitcsvm(egitim,egitimc)
3 egitims = predict(mdl,egitim);
4 tests = predict(mdl,test);

```

Modeli `fitcsvm` ile eğittik ve SVM sınıflandırmasının özellikleri aşağıda verilmiştir.

```

1 ClassificationSVM
2 ResponseName      : 'Y'
3 CategoricalPredictors : []
4 ClassNames        : [1 2]
5 ScoreTransform     : 'none'
6 NumObservations    : 104
7 Alpha              : [28*1 double]
8 Bias                : -8.0919
9 KernelParameters   : [1*1 struct]
10 BoxConstraints     : [104*1 double]
11 ConvergenceInfo    : [1*1 struct]
12 IsSupportVector    : [104*1 logical]
13 Solver             : 'SMO'

```

Modelin performansını değerlendirmek için Sayfa 67 Bölüm 3.6.5 de verilen hazır MATLAB fonksiyonları kullanılmıştır. Karışıklık matrisi sonuçları ve bu matrisi kullanarak yazılan hassasiyet, duyarlılık ve f1 skoru sonuçları aşağıdaki gibidir. Hassasiyet, duyarlılık ve f1 skoru sonuçlarına baktığımızda 1'e yakın değerler görülmüştür ve modelin pozitif olarak tahmin ettiği durumların çoğunlukla doğru olduğunu gösterir. `Classperf` kullanılarak doğruluk ve duyarlılık değerleri hesaplanmıştır ve sonuçlar 1'e yakın sonuç vermiştir.

```

1 C =
2 52      4
3 1       47
4 hassasiyet=
5 0.9811
6 0.9215
7 recall=

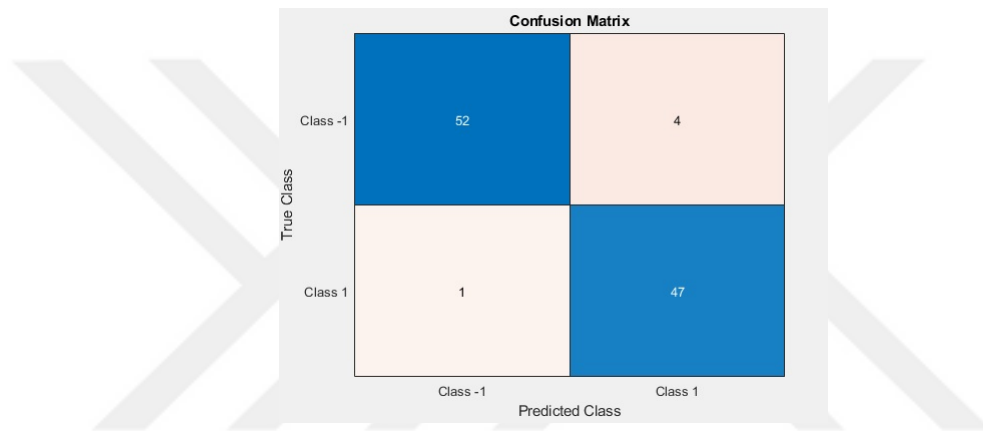
```

```

8 0.9285
9 0.9791
10 f1-skore=
11 0.9541
12 0.9494
13 Dogruluk=0.95192
14 Duyarlilik=0.92857

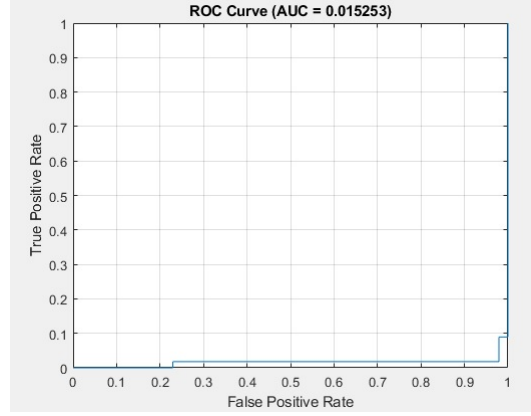
```

`confusionchart` fonksiyonu ile karışıklık matrisi **Şekil 3.13** de görselleştirilmiştir.



Şekil 3.13 Karışıklık Matrisi

`perfcurve` fonksiyonu ile Roc eğrisi **Şekil 3.14** de görselleştirilmiştir. Ancak, 0.015 gibi çok düşük bir AUC değeri, modelin performansının çok zayıf olduğunu ve sınıfları ayırt etmede zorlandığını gösterir. Sonuç olarak, diğer metriklerle birlikte değerlendirildiğinde modelin pozitif tahminlerde yüksek hassasiyet gösterdiği ancak sınıfları ayırt etmede genel olarak zayıf bir performansa sahip olduğu görülmüştür.



Şekil 3.14 Roc Eğrisi

Örnek 10 Veri seti, bir öğrencinin ders çalışmaya ayırdığı zamanın ve bunun sonucunda aldığı sınav sonuçlarının dersten geçip/kalma durumuna etkisini içerir. Model, öğrencinin çalışma süresine ve aldığı puana göre dersten geçip/kalma durumunun tahmin edilmesini üzerinedir. Toplam 500 verinin olduğu bu veri setinde giriş değişkeni çalışma süresi ve sınav sonuçları iken etiketimiz dersten geçme/kalma durumudur. Bu veri seti üzerinde karar ağaçları sınıflandırması kurmak için *fitctree* modelini kullanacağız.

```
1 close all; clear; clc;
2 cd ../Data
3 load('Data.mat')
4 cd ../ornek_1
5 disp('Verilerin Dagilimi')
6 tabulate(Data(:,end))
7 yuzde=80;
8 [egitim,egitimc,test,testc] = Orneklem(Data,yuzde);
9 disp('Egitim Veri Dagilimi')
10 tabulate(egitimc)
11 disp('Test Veri Dagilimi')
12 tabulate(testc)
13 save Egitim_Test
```

Eğitim ve test kümelerini parçalamak için sistematik örnekleme yöntemi ile verilerin %80 i eğitim, kalanı test için ayrılmıştır.

```
1 close all; clear; clc;
2 load('Egitim_Test.mat')
3 mdl=fitctree(egitim,egitimc)
4 view(mdl,'Mode','graph')
```

```
5 egitims = predict(mdl,egitim);
6 tests = predict(mdl,test);
```

Modelin performansını değerlendirmek için Sayfa 67 Bölüm 3.6.5 de verilen hazır MATLAB fonksiyonları kullanılmıştır. Burada bahsedilen bütün metriklerin değeri 1 çıkmıştır. Bu değer, modelin gerçek pozitifleri ve negatifleri mükemmel bir şekilde tahmin ettiğini göstermektedir. Bu durum genellikle nadir görülür. Veri dengesine ve aşırı uyum sorununa dikkat etmek gerekir.

Bu örnekte k-fold yöntemide kullanılmıştır. MATLAB üzerindeki kodları aşağıda verilmiştir.

```
1 close all; clear; clc;
2 cd ../Data
3 load('Data3.mat')
4 cd ../ornek_6
5 disp('Verilerin Dagilimi')
6 tabulate(Data3(:,end))
7 data_son=categorical(Data3(:,end));
8 pt=cvpartition(data_son,"Kfold",5)
9 save Egitim_Test2
```

İlk olarak veriler kategorileştirilip cvpartition komutuyla 5 alt kümeye ayrılmıştır. Kodun çıktısı aşağıda verilmiştir.

```
1 pt =
2 K-fold cross validation partition
3 NumObservations: 500
4 NumTestSets: 5
5 TrainSize: 400 400 400 400 400
6 TestSize: 100 100 100 100 100
```

Verimizdeki özelliklerimizi X değişkeni, etiketimizide Y değişkeni olarak isimlendirdik. Bir fitctree modeli eğittikten sonra kfold ile oluşturduğumuz herbir alt kümeyi değerlendirdik. Bunun için doğruluk değerlerine baktık.

```
1 load('Egitim_Test2.mat')
2 X=Data3(:,1:end-1);
3 Y=Data3(:,end);
4 treeCVModel = fitctree(X,Y);
5 accuracyList = zeros(1, 5);
6 for fold = 1:5
7 trainIdx = training(pt, fold);
```

```

8 testIdx = test(pt, fold);
9 X_train = X(trainIdx, :);
10 Y_train = Y(trainIdx);
11 X_test = X(testIdx, :);
12 Y_test = Y(testIdx);
13 treeModel = fitctree(X_train, Y_train);
14 predictions = predict(treeModel, X_test);
15 accuracy = sum(predictions == Y_test) / numel(Y_test);
16 accuracyList(fold) = accuracy;
17 fprintf('Fold %d Accuracy: %.4f\n', fold, accuracy);
18 end

```

Her bir katmandan elde edilen doğruluk değerleri 1'e yakın olduğundan dolayı modelimizin eğitim verilerine çok iyi uyum sağladığı görülmüştür.

```

1 Fold 1 Dogruluk: 1.0000
2 Fold 2 Dogruluk: 0.9900
3 Fold 3 Dogruluk: 1.0000
4 Fold 4 Dogruluk: 1.0000
5 Fold 5 Dogruluk: 1.0000

```

Örnek 11 Veri seti, 760 veriden oluşan elma meyvenin çeşitli özellikleri hakkında bilgi içerir ve bu meyvenin özelliklerine ilişkin bilgiler sağlar. Özelliklerimiz meyve boyutu, ağırlığı, tatlılığı, gevrekliği, sululuğu, olgunluğu, asitliği ve etiketimiz meyve kalitesidir. Model, verilen meyve özelliklerine göre kalitesinin iyi yada kötü olma durumunun tahmin edilmesi üzerinedir. Bu veri seti üzerinde KNN sınıflandırması kurmak için `fitcknn` modelini kullanacağız.

```

1 load('Egitim_Test.mat')
2 mdl=fitcknn(egitim,egitimc)
3 egitims = predict(mdl,egitim);
4 tests = predict(mdl,test);

```

Eğitim ve test kümelerini parçalamak için sistematik örnekleme yöntemi ile verilerin %80 i eğitim, kalanı test için ayrılmıştır. Modelin performansını değerlendirmek için Sayfa 67 Bölüm 3.6.5 de verilen hazır MATLAB fonksiyonları kullanılmıştır. Burada bahsedilen AUC değeri '0' diğer bütün metriklerin değeri '1' çıkmıştır. Diğer metriklerin 1 olması modelin çok iyi performans gösterdiğini gösteriyor ancak AUC değerinin 0 olması, ROC eğrisinin altındaki alanın hiç olmadığını veya çok düşük olduğunu gösterir. Bu durum modelin rastgele tahmin yapmasından kaynaklanabilir.

```

1 C =
2 68    0
3 0     84
4 hassasiyet=1
5 recall=    1
6 f1-skore=   1
7 Dogruluk=   1
8 Duyarlilik=0

```

Örnek 12 Bu örnekte veri seti olarak KNN sınıflandırmasında kullanılan elma meyvesinin verileri kullanılmıştır. Özellik ve etiket aynı seçilmiştir. Bu veri seti üzerinde Hibrit sınıflandırması kurmak için üç farklı algoritmanın Karar Ağaçları , SVM ve KNN model fonksiyonlarının birleşmesiyle oluşturulmuştur.

```

1 load('Egitim_Test.mat')
2 CTree=fitctree(egitim,egitimc);
3 SVMs=fitcsvm(egitim,egitimc);
4 kNN=fitcknn(egitim,egitimc);
5 Egitims_CTree=predict(CTree,egitim);
6 Egitims_SVMs=predict(SVMs,egitim);
7 Egitims_kNN=predict(kNN,egitim);
8 Cevaplar = [Egitims_CTree Egitims_SVMs Egitims_kNN];
9 Cevaplar(:,4)=round(mean(Cevaplar,2));
10 Egitims_Hibrit = Cevaplar(:,4);
11 Tests_CTree=predict(CTree,test);
12 Tests_SVMs=predict(SVMs,test);
13 Tests_kNN=predict(kNN,test);
14 Cevaplar_T = [Tests_CTree Tests_SVMs Tests_kNN];
15 Cevaplar_T(:,4)=round(mean(Cevaplar_T,2));
16 Tests_Hibrit = Cevaplar(:,4);

```

Eğitim ve test kümelerini parçalamak için sistematik örnekleme yöntemi ile verilerin %80 i eğitim, kalanı test için ayrılmıştır. Modelin performansını değerlendirmek için Sayfa 67 Bölüm 3.6.5 de verilen hazır MATLAB fonksiyonları kullanılmıştır. Karışıklık matrisi sonuçları ve bu matrisi kullanarak yazılan hassasiyet, duyarlılık ve f1 skoru sonuçları aşağıdaki gibidir. Hassasiyet, duyarlılık ve f1 skoru sonuçlarına baktığımızda 1'e yakın değerler görülmüştür ve modelin pozitif olarak tahmin ettiği durumların çoğunlukla doğru olduğunu gösterir.

```

1 C =
2 64    4
3 4     80

```

```

4 hassasiyet= 0.94118
5 f1-skore   = 0.94118
6 Dogruluk   = 0.94737
7 Duyarlilik= 0.94118

```

Örnek 13 *Veri seti, İris çiçeğinin yaprak özelliklerini ve üç farklı özellikte türünü içerir. Bu türler Versicolor, Virginica ve Setosa. Bu verilerde çiçeğin yaprak genişliği ve uzunluğuna bakılarak türü etiketlenmiştir. Bu veri seti üzerinde Naive Bayes sınıflandırması kurmak için fitcnb modelini kullanacağız.*

```

1 close all; clear; clc;
2 cd ../Data
3 load('iris1.mat')
4 cd ../ornek_7
5 X=iris1(:,1:2);
6 Y=setosa1;
7 tabulate(Y)
8 figure;
9 gscatter(X(:,1), X(:,2), setosa1,'rgb','osd');
10 xlabel('Sepal length');
11 ylabel('Sepal width');
12 rng('default')
13 data_son=categorical(setosa1);
14 pt=cvpartition(data_son,"Kfold",5);
15 order = unique(Y)
16 func = @(Xtrain,ytrain,Xtest,ytest) confusionmat(ytest, ...
17 classify(Xtest,Xtrain,ytrain),'Order',order);
18 confMat = crossval(func,X,Y,'Partition',pt);
19 cvMat = reshape(sum(confMat),3,3)
20 confusionchart(cvMat,order)
21 save Egitim_Test3

```

Veri setinin sınıflarını görsel olarak inceledik ve sınıflandırma modelinin performansını k-fold ile değerlendirdik. Verilerin dağılımı ve karışıklık matrisi aşağıdaki gibidir.

```

1 Verilerin Dagilimi
2 Value      Count      Percent
3 setosa      50         33.33%
4 versicolor  50         33.33%
5 virginica   50         33.33%
6 pt =
7 K-fold cross validation partition

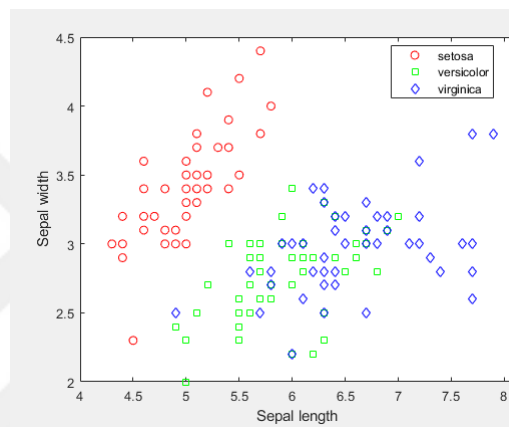
```

```

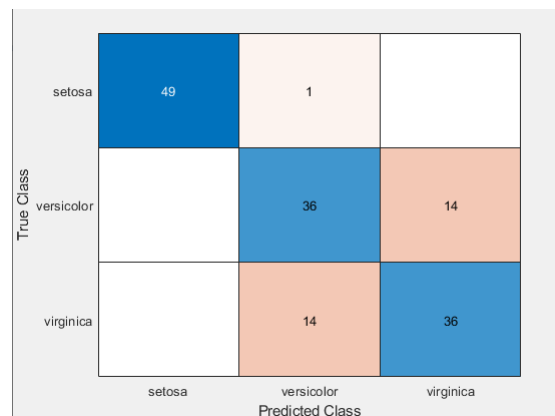
8 NumObservations: 150
9 NumTestSets: 5
10 TrainSize: 120 120 120 120 120
11 TestSize: 30 30 30 30 30
12 cvMat =
13 49      1      0
14 0      36     14
15 0      14     36

```

Sınıfların dağılımı aşağıdaki gibi görselleştirilmiştir.



Şekil 3.15 Naive Bayes



Şekil 3.16 Naive Bayes Karışıklık Matrisi

Örnekte veriler kategorileştirilip `cvpartition` komutuyla 5 alt kümeye ayrılmıştır. Daha sonra karışıklık matrisi oluşturularak durumları görülmüştür. Ardından Naive Bayes modeli ile eğitilmiştir. Eğitim ve test seti gözlemlerini

Şekil 3.17 te görselleştirdik ve k-fold ile eğitilen bir Naive Bayes modelinin hata oranını hesaplamak için kfoldLoss fonksiyonunu kullandık.

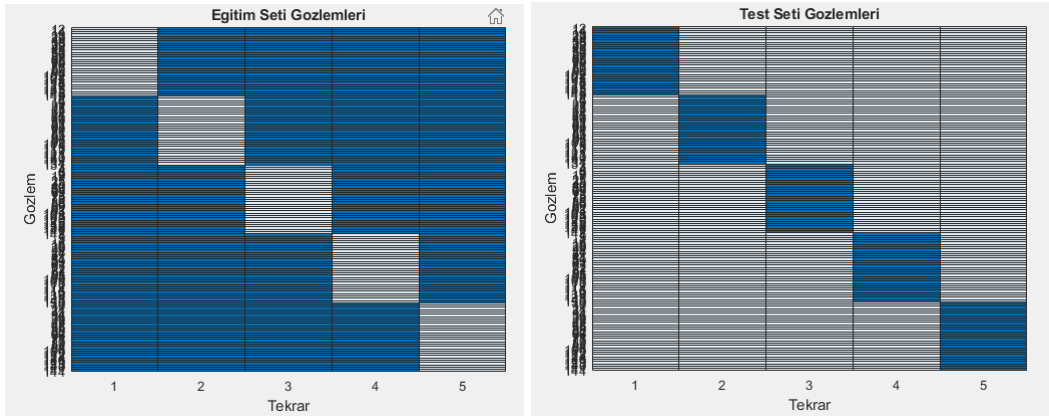
```

1 load('Egitim_Test3.mat')
2 model = fitcnb(X,data_son,'CVPartition',pt)
3 test1=test(pt,1);
4 test2=test(pt,2);
5 test3=test(pt,3);
6 test4=test(pt,4);
7 test5=test(pt,5);
8 veril = [test1,test2,test3,test4,test5];
9 h = heatmap(double(veril),'ColorbarVisible','off');
10 sorty(h,{'1','2','3','4','5'},'descend')
11 xlabel('Tekrar')
12 ylabel('Gozlem')
13 title('Test Seti Gozlemleri')
14 egitim1=training(pt,1);
15 egitim2=training(pt,2);
16 egitim3=training(pt,3);
17 egitim4=training(pt,4);
18 egitim5=training(pt,5);
19 veri2 = [egitim1,egitim2,egitim3,egitim4,egitim5];
20 h = heatmap(double(veri2),'ColorbarVisible','off');
21 sorty(h,{'1','2','3','4','5'},'ascend')
22 xlabel('Tekrar')
23 ylabel('Gozlem')
24 title('Egitim Seti Gozlemleri')
25 treeRate = kfoldLoss(model)
26 tahmin=kfoldPredict(model);

```

Modelin performansını değerlendirmek için Sayfa 67 Bölüm 3.6.5 de verilen hazır MATLAB fonksiyonları kullanılmıştır. Karışıklık matrisi sonuçları ve bu matrisi kullanarak yazılan hassasiyet, duyarlılık ve f1 skoru sonuçları aşağıdaki gibidir. Hassasiyet, duyarlılık ve f1 skoru ile birlikte hata oranı 0.2267 değeri sonuçlarına baktığımızda performansın iyi olduğu görülmüştür.

1 treeRate	: 0.2267
2 Dogruluk (Accuracy)	: 0.7733
3 Hassasiyet (Precision)	: 0.6429
4 Duyarlilik (Recall)	: 0.7200
5 F1-Skoru	: 0.6792



Şekil 3.17 Naive Bayes Gözlemleri

3.6.6 Kümeleme

Bu bölümde MATLAB kullanılarak bazı kümeleme algoritmaları için hesaplama örnekleri verilecektir. Bu örneklerde elde edilen kümeleme algoritmalarının başarısını değerlendirmek için Sayfa 29 Bölüm 3.2.1 de verilen metrikler kullanılmalıdır. Bu metrikleri hesaplayabilmek için aşağıdaki MATLAB fonksiyonları kullanılır.

MATLAB üzerinde kümeleme performans değerlendirme

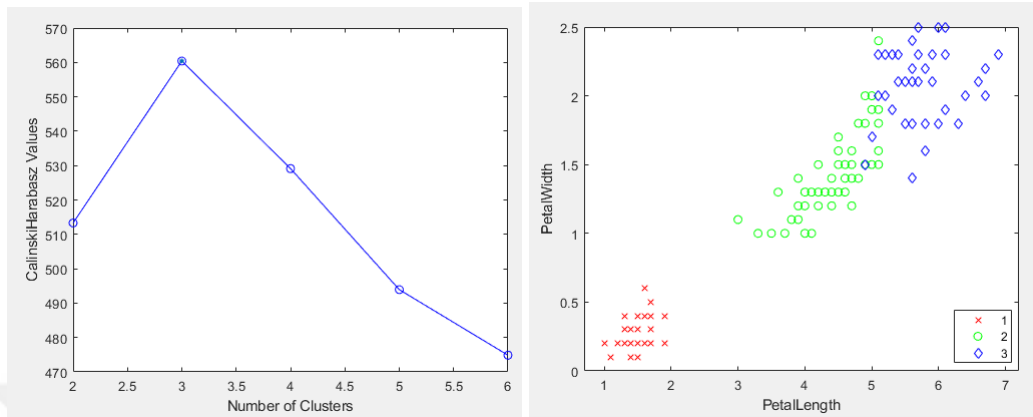
Dahili yöntemlerden Silhouette, Calinski-Harabasz ve Davies-Bouldin indeksleri kullanılmıştır.

```
1 close all; clear; clc;
2 cd ../Data
3 load('iris1.mat')
4 cd ../Performans_Degerlendirme
5 X=iris12;
6 Y=setosa1;
7 tabulate(Y)
8 %Calinski-Harabasz
9 rng('default')

1 eva= evalclusters(X,'kmeans','CalinskiHarabasz','KList',2:6)
2 plot(eva)
3 PetalLength = X(:,3);
4 PetalWidth = X(:,4);
5 clusters = eva.OptimalY;
6 gscatter(PetalLength,PetalWidth,clusters,[],'xod');
7 %Davies-Bouldin
8 eva = evalclusters(X,"kmeans","DaviesBouldin","KList",2:6)
```

```
9 plot(eva)
```

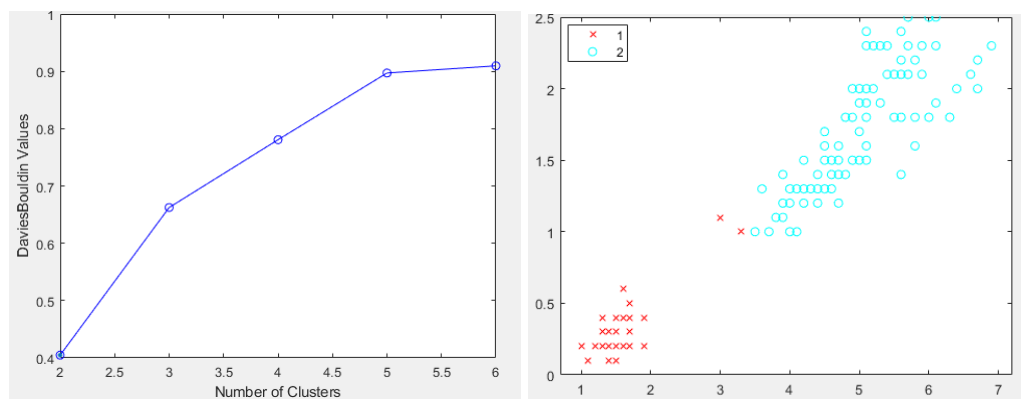
Calinski-Harabasz grafiksel yorumu aşağıdaki gibidir.



Şekil 3.18 Calinski Harabasz

```
1 clusters = eva.OptimalY;
2 gscatter(X(:,3),X(:,4),clusters,[],"xod")
3 %Silhouette Evaluation
4 eva = evalclusters(X,"kmeans","silhouette","KList",2:6)
5 plot(eva)
6 clusters = eva.OptimalY;
```

Davies-Bouldin grafiksel yorumu aşağıdaki gibidir.



Şekil 3.19 Davies Bouldin

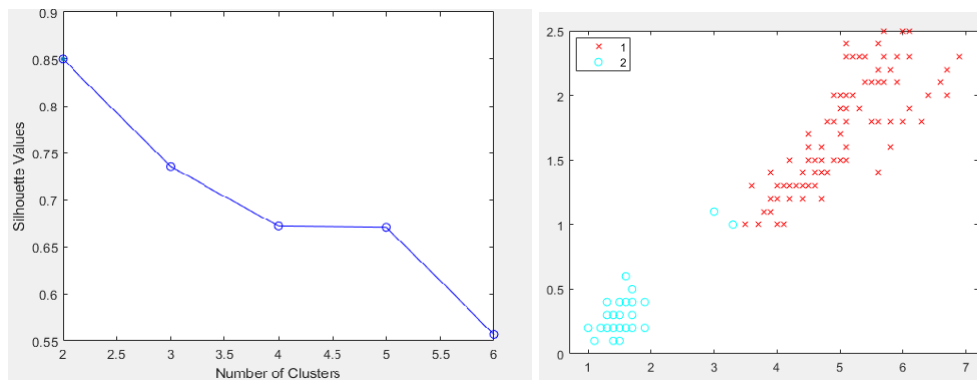
```
1 gscatter(X(:,3),X(:,4),clusters,[],"xod")
2 %% silhouette
3 [idx3,C,sumdist3] = kmeans(X,3,'Distance','cityblock', ...
```

```

4     'Display','final');
5 [silh3,h] = silhouette(X,idx3,'cityblock');
6 xlabel('Silhouette Value')
7 ylabel('Cluster')
8 idx4 = kmeans(X,4,'Distance','cityblock','Display','iter');
9 [silh4,h] = silhouette(X,idx4,'cityblock');
10 xlabel('Silhouette Value')
11 ylabel('Cluster')
12 cluster3 = mean(silh3)
13 cluster4 = mean(silh4)
14 idx5 = kmeans(X,5,'Distance','cityblock','Display','final');
15 [silh5,h] = silhouette(X,idx5,'cityblock');
16 xlabel('Silhouette Value')
17 ylabel('Cluster')
18 mean(silh5)
19 c1=setosa1;
20 k=3;
21 out=tclust(X,k,0.1,100,'msg',0);
22 c2=out.idx;
23 [ARI,RI,MI,HI]=RandIndexFS(c1,c2);
24 disp("ARI index")
25 disp(ARI)
26 [tbl,~,~,labels]=crosstab(c1,c2);
27 TBL=array2table(tbl,'RowNames',labels(1:3,1),'VariableNames',labels(1:k
    +1,2));
28 disp(TBL)
29 idx=kmeans(X,3,'emptyaction','singleton','Replicates',5);
30 [Acc,rand_index,match]=AccMeasure(categorical(Y),idx)
31 save kMeans_kumeleme

```

Silhouette grafiksel yorumu aşağıdaki gibidir.



Şekil 3.20 Silhouette

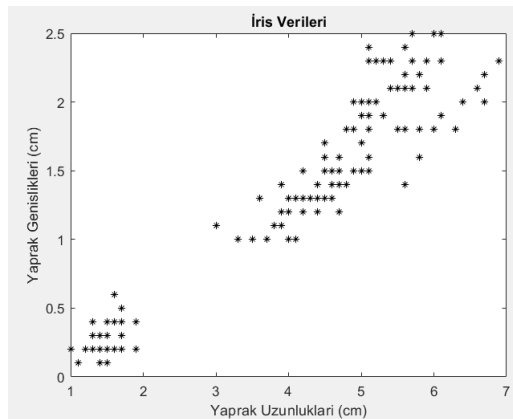
Örnek 14 Veri seti, Sayfa 75 Bölüm 13 deki örnekte verilen İris çiçeği verisi kullanılmıştır. Bu veri seti üzerinde kümeleme algoritması kurmak için *kmeans* modelini kullanacağız.

```
1 close all; clear; clc;
2 cd ../Data
3 load('iris1.mat')
4 cd ../ornek_1
5 X=iris12(:,3:4);
6 Y=setosa;
7 disp('Verilerin Dagilimi')
8 tabulate(Y)
9 figure;
10 plot(X(:,1),X(:,2),'k*','MarkerSize',5);
11 title 'iris Verileri';
12 xlabel 'Yaprak Uzunluklari (cm)';
13 ylabel 'Yaprak Genislikleri (cm)';
14 rng(1);
```

Seçilen özelliklerin sınıfları nasıl ayırdığını gözlemleyebilmek için bir grafik oluşturduk.

	Value	Count	Percent
1	setosa	50	33.33%
2	versicolor	50	33.33%
3	virginica	50	33.33%

Yukarıdaki komutların grafiksel yorumu Şekil 3.21 deki gibidir.



Şekil 3.21 Kmeans

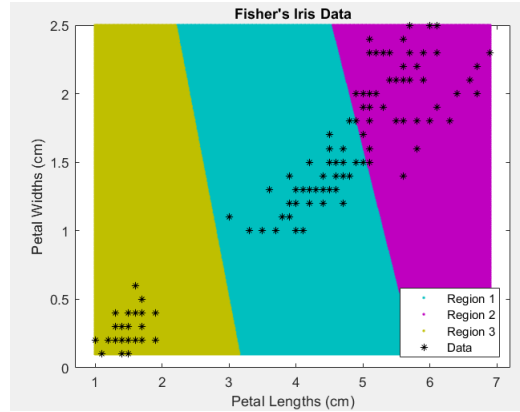
İris veri setinde 3 tür iris çiçeği etiketlendiği için $k = 3$ alınarak kümeleme yapılmıştır. Daha sonra meshgrid kullanılarak veri setinin uzayının daha düzenli bir şekilde temsil edilmesini sağladık. Kmeans fonksiyonu tekrar çağrılarak bu düzenli uzayda kümeler belirledik (idx2Region) ve oluşturulan kümeler ile veri setini grafikte görselleştirdik.

```

1 [idx,C] = kmeans(X,3);
2 x1 = min(X(:,1)):0.01:max(X(:,1));
3 x2 = min(X(:,2)):0.01:max(X(:,2));
4 [x1G,x2G] = meshgrid(x1,x2);
5 XGrid = [x1G(:),x2G(:)];
6 idx2Region = kmeans(XGrid,3,'MaxIter',1,'Start',C);
7 figure;
8 gscatter(XGrid(:,1),XGrid(:,2),idx2Region,...
9         [0,0.75,0.75;0.75,0,0.75;0.75,0.75,0],'..');
10 hold on;
11 plot(X(:,1),X(:,2),'k*','MarkerSize',5);
12 title 'Fisher''s Iris Data';
13 xlabel 'Petal Lengths (cm)';
14 ylabel 'Petal Widths (cm)';
15 legend('Region 1','Region 2','Region 3','Data','Location','SouthEast');
16 hold off;

```

Yukarıdaki komutların grafiksel yorumu Şekil 3.22 deki gibidir.



Şekil 3.22 Kmeans

İkinci bir kümeleme işlemi yapılır. İki küme oluşturulur ve görselleştirilir. Küme merkezleri (C) ve küme üyeleri renkli olarak grafiğe eklenir.

```

1 opts = statset('Display','final');
2 [idx,C] = kmeans(X,2,'Distance','cityblock',...
3     'Replicates',5,'Options',opts);

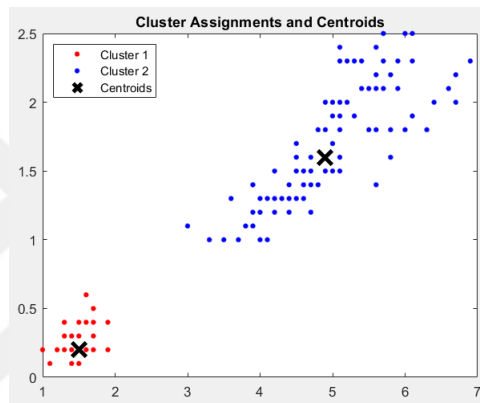
```

```

4 figure;
5 plot(X(idx==1,1),X(idx==1,2),'r.','MarkerSize',12)
6 hold on
7 plot(X(idx==2,1),X(idx==2,2),'b.','MarkerSize',12)
8 plot(C(:,1),C(:,2),'kx',...
9      'MarkerSize',15,'LineWidth',3)
10 legend('Cluster 1','Cluster 2','Centroids',...
11        'Location','NW')
12 title 'Cluster Assignments and Centroids'
13 hold off

```

Yukarıdaki komutların grafiksel yorumu Şekil 3.23 gibidir.



Şekil 3.23 Kmeans

Veriler kategorileştirilip `cvpartition` komutuyla 5 alt kümeye ayrılmıştır. Daha sonra karışıklık matrisi yani çapraz tablo oluşturularak durumları görülmüştür.

```

1 rng('default')
2 data_son=categorical(setosa1);
3 pt=cvpartition(data_son,"KFold",5);

1 pt =
2 K-fold cross validation partition
3 NumObservations: 150
4 NumTestSets: 5
5 TrainSize: 120 120 120 120 120
6 TestSize: 30 30 30 30 30

```

Aşağıda karışıklık matrisinin kodları verilmiştir.

```

1 save kMeans_kumeleme

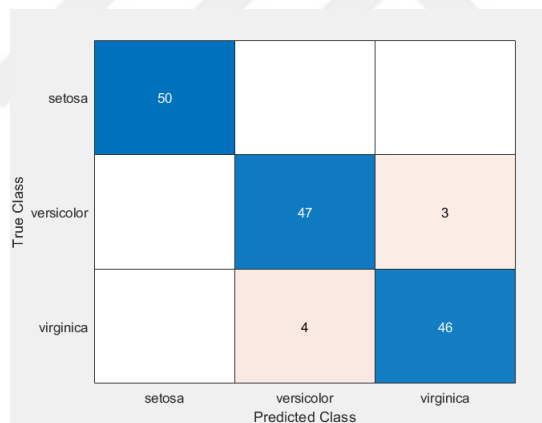
```

Karışıklık matrisi ile verilerin durumuna bakılır.

```
1 order = unique(Y)
2 func = @(Xtrain,ytrain,Xtest,ytest) confusionmat(ytest, ...
3         classify(Xtest,Xtrain,ytrain),'Order',order);
4 confMat = crossval(func,X,Y,'Partition',pt);
5 cvMat = reshape(sum(confMat),3,3)
6 confusionchart(cvMat,order)
```

```
1 order =
2 3x1 categorical array
3 setosa
4 versicolor
5 virginica
6 cvMat =
7 50      0      0
8 0       47     3
9 0       4     46
```

Yukarıdaki komutların grafiksel yorumu **Şekil 3.24** gibidir.



Şekil 3.24 Kmeans Karışıklık Matrisi

İris veri seti üzerinde yapılan kümeleme performans değerlendirmesine göre, Calinski-Harabasz ölçütü en iyi küme sayısını 3 olarak belirlerken, Davies-Bouldin indeksi en iyi küme sayısını 2 olarak belirlemiştir. Silhouette değerlendirmesi de en iyi küme sayısını 2 olarak göstermiştir. Silhouette analizi sonuçlarına göre, küme sayısı 3 için ortalama Silhouette değeri 0.4902, küme sayısı 4 için ise 0.4199 olarak bulunmuştur. Adjusted Rand Index (ARI) indeksi ise 0.7824 olarak hesaplanmıştır, bu değer kümeleme sonuçlarının gerçek sınıflandırmaya oldukça yakın olduğunu göstermektedir.

4. ARAŞTIRMA METODOLOJİSİ

4.1 Veri Seti

Bu tez çalışmasında kullandığımız veriler, BTC'nin 25/07/2010'dan 05/06/2022'e kadar olan günlük USD cinsinden değerinin yanısıra bazı ek parametreler kullanılmıştır. Veriler TradingView web sitesinden elde edilmiştir. Bu verilere ek olarak BTC fiyatı üzerinde etkili olduğunu düşündüğümüz dolar endeksi değerlerindeki ilgili günler için bularak verisetine ekledik. Böylece çalışmada giriş parametresi olarak açılış fiyatı, en yüksek-en düşük fiyat, hacim, dolar endeksi ve teknik analizde kullanılan bazı indikatörler ile kullanılmıştır. Veri setinde yer alan bu özellikler şu şekildedir;

Parametre	Açıklama
Open	Bitcoinin açılış fiyatıdır.
High	Bitcoinin gün içindeki en yüksek fiyatıdır.
Low	Bitcoinin gün içindeki en düşük fiyatıdır.
MA	Moving Average (MA), bir varlığın belirli bir zaman dilimindeki kapanış fiyatlarının ortalamasını hesaplamak için kullanılan bir göstergedir.
Volume	Günlük işlem hacmidir.
RSI	Göreceli güç endeksi, Welles Wilder Jr. Tarafından 1978'de geliştirilen bir finansal analiz aracıdır, ve hisse senetleri üzerindeki eğilimleri değerlendirmek için kullanılır. 14 günlük dönemdeki fiyat değişimlerine dayanarak, bir varlığın aşırı alım ya da aşırı satım durumlarını belirlemeye yardımcı olur. RSI, yatırımcılara alım-satım kararları konusunda rehberlik etmek için kullanılabilecek değerler üretir. Genellikle 30 ve 70 seviyeleri, aşırı satım ve aşırı alım bölgelerini belirlemek için kullanılır. Örneğin, 70'in üzerinde bir RSI değeri, bir varlığın potansiyel olarak aşırı alım durumunda olduğunu gösterebilir, bu da satış fırsatlarını düşündürtebilir. Diğer yandan, 30'un altındaki bir değer, aşırı satış durumunu işaret edebilir ve alım fırsatlarına işaret edebilir.

MACD	Hareketli Ortalamaların Yakınsaması/Uzaklaşması (MACD), Gerald Appel tarafından 1960'lı yıllarda geliştirilmiş bir teknik analiz indikatörüdür ve finansal piyasalarda yaygın olarak kullanılan en popüler göstergelerden biridir.
Bollinger Bands (Basis, Upper, Lower)	Bollinger Bantları, teknik analizde sıkça kullanılan bir volatilité bandıdır ve 1980 yılında John Bollinger tarafından geliştirilmiştir. Hareketli ortalamaların üstüne ve altına yerleştirilen bu bantlar, fiyat hareketlerinin göreceli yüksekliğini veya düşüklüğünü belirlemek amacıyla kullanılır. Volatilité, standart sapma tarafından ölçülür ve bu değer, fiyat volatilitésindeki değişikliklere bağlı olarak genişleyip daralabilir. Bollinger Bantları, genellikle fiyat hareketlerinin %88-89'unu kapsadığı düşünülen bir aralığı temsil eder. Bu nedenle, fiyatların bantların dışına çıkması olağandışı olarak değerlendirilir. Üst banda yaklaşan fiyatlar genellikle göreceli olarak yüksek olarak kabul edilirken, alt banda yaklaşan fiyatlar göreceli olarak düşük olarak yorumlanır. 2011 yılında Bollinger Bantları, John Bollinger adına patentlenmiştir.
WMA	Ağırlıklı hareketli ortalama (WMA) olarak da bilinen hareketli ortalama ağırlıklı göstergesi, trend yönünü belirlemenize veya onaylamanıza yardımcı olabilecek bir teknik analiz aracıdır. Bu gösterge, mevcut fiyatı önceki fiyatlarla karşılaştırır. En son fiyatlara daha fazla ağırlık verir ve daha eski veri noktalarına daha az ağırlık verir, bu da verilerin öneminin doğrusal olarak değiştiğini gösterir. Böyle bir ağırlıklandırma sisteminin arkasındaki fikir, son fiyatların fiyat hareketlerini tahmin etmek için daha güvenilir olmasıdır. WMA, veri setindeki her sayının önceden belirlenmiş bir ağırlıkla çarpılması ve elde edilen değerlerin toplanmasıyla elde edilir.
ROC	Değişim hızı göstergesi (ROC) bir momentum osilatörüdür. Dönemler arası fiyat değişimini yüzde olarak hesaplar. ROC mevcut fiyatı alır ve bunu "n" dönem (kullanıcı tanımlı) önceki fiyatla karşılaştırır. Hesaplanan değer daha sonra çizilir ve bir sıfır çizgisinin üstünde ve altında dalgalanır.

Coppock Eğrisi	Coppock Eğrisi, bir borsa endeksindeki büyük gerileme ve yükselişleri tanımak için kullanılan uzun vadeli bir fiyat momentum göstergesidir. Endeks için 14 aylık değişim oranı ile 11 aylık değişim oranının toplamının 10 aylık ağırlıklı hareketli ortalaması olarak hesaplanır.
CCI	Emtia Kanalı Endeksi (CCI), bir yatırım aracının aşırı alım veya aşırı satım durumlarına ulaşp ulaşmadığını belirlemek amacıyla kullanılan bir momentum tabanlı osilatördür.
EFI	Elder's Force Index (EFI), fiyat ve hacmi kullanarak bir fiyat hareketinin arkasındaki gücü ölçer. Gösterge, potansiyel geri dönüşleri ve fiyat düzeltmelerini belirlemek için de kullanılabilir. EFI, Sıfır Çizgisinin üstünde ve altında, pozitif ve negatif değerler arasında dalgalanan bir osilatördür.
EMA	Exponential Moving Avarage (EMA), herhangi bir hisse senedindeki veya döviz paritelerindeki fiyat değişimlerini gözlemlemek için kullanılır.
MFI	Para Akışı Endeksi göstergesi (MFI), teknik analizde alım satım baskısını ölçmek için kullanılan bir araçtır. Bu, hem fiyat hem de hacmi analiz ederek yapılır.
Dolar Endeksi	ABD dolar endeksinin sembolü ve kısaltması DXY şeklinde yazılır. DXY Euro, Japon Yen'i, İngiliz Pound'u, Kanada Doları, İsviçre Frangı ve İsveç Kronu'ndan oluşan 6 farklı ülkenin 6 farklı para birimi karşısındaki değeri göstermektedir.
Close	Bitcoin kapanış fiyatıdır.

Veri seti üzerinde yapılan ön analizlerde tez kapsamında Gaussian process regression (GPR) algoritmasının daha başarılı sonuçlar vereceği gözlenmiştir. Bu kapsamda oluşturulacak olan GPR modelleri dört farklı kernel fonksiyonu ve basis fonksiyonları altında detaylı olarak değerlendirilecektir.

4.2 Modelin Oluşturulması ve Eğitilmesi

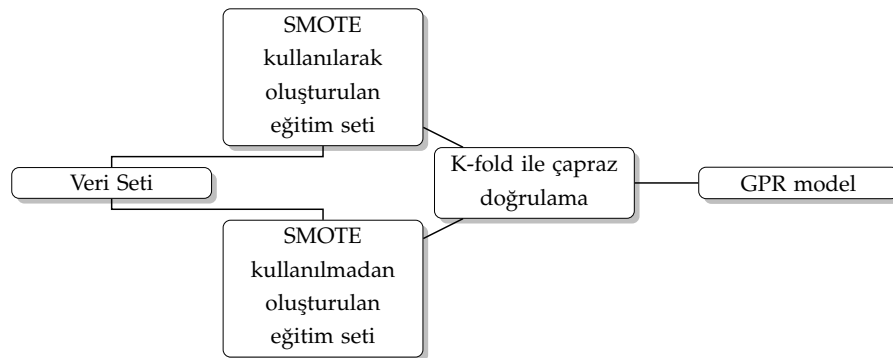
Oluşturmuş olduğumuz veri setinde 16 bağımsız değişken kullanarak BTC'nin kapanış fiyatını tahmin etmek hedeflenmektedir. Bazı bağımsız

değişkenlerdeki eksik veriler düzenlendikten sonra veri setindeki toplam örnek sayısı 4320'dir. Bu veri seti kullanılarak 16 bağımsız değişkenle BTC kapanış fiyatını tahmin etmeye çalıştığımız modele Model1 ismi verilmiştir. Daha sonra veri setindeki bağımsız değişkenlerden BTC açılış fiyatı, en yüksek ve en düşük BTC değeri çıkartıldıktan sonra 13 bağımsız değişkenle BTC kapanış fiyatının tahmin etmeye çalıştığımız modele Model3 ismi verilmiştir.

Tez kapsamında sentetik verilerin model başarısına etkisi gözlenmek istendiğinden SMOTE yöntemi kullanılarak 4320 olan veri noktası sayısı 8640'a çıkartılmıştır. Bu yeni veri seti kullanılarak 16 bağımsız değişkenle BTC kapanış fiyatını tahmin etmeye çalıştığımız modele Model2 ismi verilmiştir. Son olarak SMOTE yöntemiyle üretilen sentetik verileri içeren veri setinden yine BTC açılış fiyatı, en yüksek ve en düşük BTC değerinin çıkarılarak BTC kapanış fiyatının tahmin etmeye çalıştığımız modele Model4 ismi verilmiştir.

Verileri modele beslemeden önce, verilerin bir kısmı eğitim bir kısmı test olarak ayrılması gerekir. Oluşturacağımız modeller için test ve eğitim kümelerinin seçiminin başarı oranını etkilemesinin önüne geçmek için her modelde k-Fold yöntemi $k=5$ olarak uygulanmıştır. Böylelikle her seferinde veriler %80-%20 oranında eğitim ve test olarak kullanılarak tüm veriler eğitim ve test süreçlerine dahil edilmiştir.

Modelin eğitim sürecinde seçilen GPR modelinin dört çekirdek fonksiyonları ile birlikte dört temel fonksiyonları kullanıldı. Oluşturulan 4 modelin eğitim süreci **Şekil 4.1** de görselleştirilmiştir.



Şekil 4.1 Akış Şeması, eğitim süreci adımları

4.3 Değerlendirme Metrikleri

Değerlendirme süreci için doğruluk derecesi (R), ortalama mutlak hata (MAE), ortalama kare hatası (MSE), kök ortalama kare hatası (RMSE), R-kare (R^2), ortalama mutlak hata (MAPE) ile birlikte modeldeki her bir ek bağımsız değişkenin katkısını gözönünde bulunduran düzeltilmiş R-kare (AdjRsqr) metrikleri kullanılmıştır.

Bu metrikler, her modelin performansını daha iyi analiz etmemizi sağlar. Regresyon modelleri sonuçlarına göre MAPE ve RMSE değeri, gerçek değerler ile tahmin edilen değerler arasında son derece küçük sapmaları gösteren değerlerdir (Chen ve Wang, 2007). MAPE tahmini; çıktı değerlerinin doğruluğunun değerlendirilmesi ve ölçülmesi için tahmini hata değerlerinin ölçülmesinde yaygın olarak kullanılır. MAPE; hata değerlerini yüzde olarak gösterir. MAPE değerleri % 10'un altında olan tahmin modelleri "yüksek doğruluk" derecesine, % 10 ile % 20 arasında olan modeller ise doğru tahminler %20 ile %50 arasında olan modellerin "kabul edilebilir" ve %50'nin üzerinde olan modelleri ise "yanlış ve hatalı" olarak sınıflandırmaktadır Metin, 2021. AdjRsqr değeri regresyon analizinde modelin performansını değerlendirmek için kullandığımız istatistiksel bir ölçüttür. 1'e ne kadar yakınsa modelin performansı o kadar iyidir.

4.4 Model Sonuçları

Model1 sonuçları

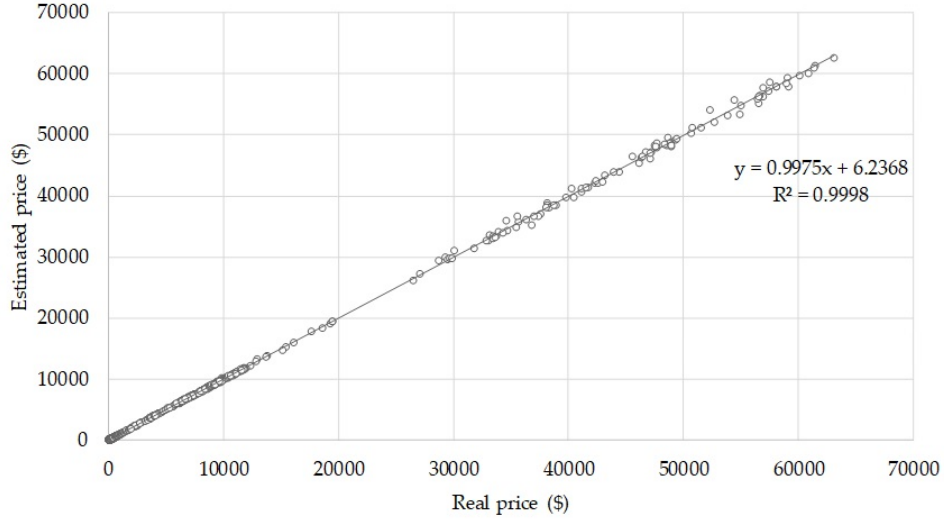
Model1 de Table 2 de verilen kriterler dikkate alınarak GPR modeli oluşturulmuştur.

Model1 den elde edilen sonuçlar **Çizelge 4.2** de verilmiştir.

Kernel	Basis Function	R	MAE	MSE	RMSE	R^2	AdjRsqr	MAPE
Exponential	Constant	0.96427	1204	1.51E+07	3883.4	0.92881	0.9273	124.81
Exponential	None	0.96453	1203.8	1.52E+07	3865.7	0.92918	0.92767	124.94
Exponential	Linear	0.99988	91.557	50655	224.91	0.99976	0.99976	1017.4
Exponential	Pure Quadratic	0.99989	95.885	49086	221.42	0.99977	0.99977	1887
Squared Exponential	Constant	0.8256	468.27	2.56E+08	9823	-0.24188	-0.26833	670.07
Matern 5/2	Constant	0.99966	96.524	1.4748e+05	355.92	0.99932	0.99931	690.04
Rational Quadratic	Constant	0.87404	398.36	3.9722e+08	9170.4	-0.79373	-0.83194	976.93
Squaredexponential	Optimize Hyperparameters	0.99987	133.12	57713	239.42	0.99973	0.99972	3450.3

Çizelge 4.2 GPR Performans Sonuçları

Model1 sonuçların grafiği Şekil 4.2 deki gibi elde edilir.



Şekil 4.2 GPR grafiği

Çizelge 4.2 deki AdjRsqr sonuçları dikkate alınarak exponential kernel fonksiyonu ve pure quadratic basis fonksiyonu modeli 0.99977 başarı oranıyla diğer parametrelere göre daha iyi sonuç verdiği görülmüştür. En yüksek performansı veren GPR modelinin tahmin edilen ve gerçek değerlerin korelasyon grafiği Şekil 4.2 deki gibidir. Şekil 4.2 de gösterilen tahmin değerlerinin doğrusal bir grafik göstermesi modelin performansının yüksek olduğunu göstermektedir. Grafikteki y değeri doğrunun denklemini, R^2 ise korelasyon katsayısının karesini göstermektedir. Hangi parametrenin BTC fiyatını güçlü bir şekilde etkilediğini belirlemek için verilere PCA analizi uygulandı ve en yüksek parametrenin açık fiyat olduğu görüldü. Ayrıca verilere optimizasyon uygulandı. Bunun için fitrgp modeli için MATLAB'ın squaredexponential kernel fonksiyonu ve OptimizeHyperparameters kullanılmıştır. AdjRsqr sonucu 0,99972 olarak bulundu. Genetik algoritma literatürdeki çeşitli çalışmalarda benzer şekilde uygulanmıştır (Abo-Hammour, Alsmadi, vd. 2013; Abo-Hammour, Arqub, vd. 2014; Arqub ve Abo-Hammour, 2014; Arqub, Abo-Hammour, vd. 2012). Bu yöntem Model2, Model3 ve Model4'e uygulanamadı.

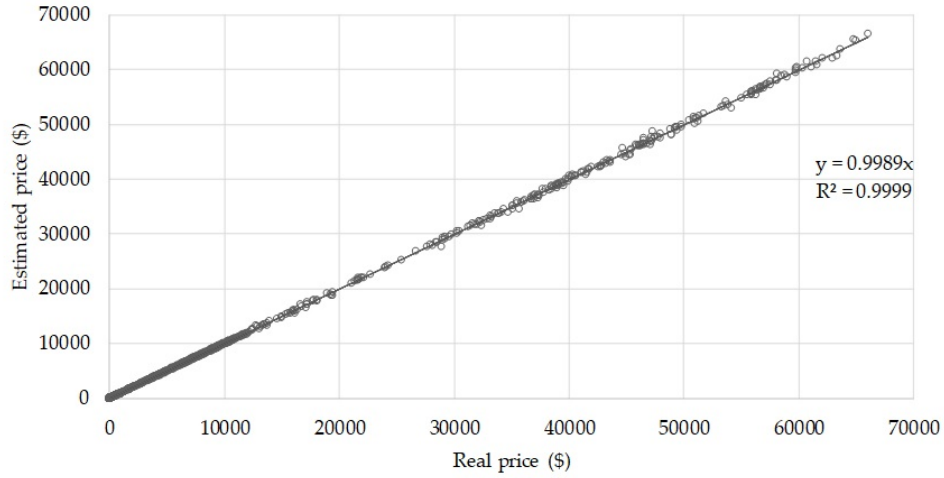
Model2 sonuçları

Model1 den farklı olmak üzere Model2 de elde ettiğimiz verileri sentetik olarak SMOTE yöntemiyle biraz daha arttırdık ve performansını görmek istedik. Model2 den elde edilen sonuçlar **Çizelge 4.3** de verilmiştir.

Kernel	Basis Function	R	MAE	MSE	RMSE	R ²	AdjRsqr	MAPE
Exponential	Constant	0.99037	493.05	3.8697e+06	1965.2	0.98048	0.98027	49.876
Exponential	None	0.99044	490.43	3.845e+06	1958.7	0.9806	0.9804	48.799
Exponential	Linear	0.99991	73.268	34146	183.34	0.99983	0.99983	731.53
Exponential	Pure Quadratic	0.99993	68.307	26951	162.7	0.99986	0.99986	1160.8
Squared Exponential	Constant	0.97825	125.14	1.1231e+07	1791.3	0.94434	0.94375	429.06
Matern 5/2	Constant	0.99984	76.369	63255	236.74	0.99968	0.99968	768.61
Rational Quadratic	Constant	0.97832	125.78	1.1181e+07	1795.2	0.94458	0.944	452.88

Çizelge 4.3 GPR-SMOTE Performans Sonuçları

Model2 sonuçların grafiği **Şekil 4.3** deki gibi elde edilir.



Şekil 4.3 GPR-SMOTE grafiği

Çizelge 4.3 deki AdjRsqr sonuçları dikkate alınarak exponential kernel fonksiyonu ve pure quadratic basis fonksiyonu modeli 0.99986 başarı oranıyla diğer parametrelerden daha iyi sonuç verdiği görülmüştür. En yüksek performansı veren GPR modelin grafiği **Şekil 4.3** deki gibidir. Grafiğin doğrusal olması modelin yüksek performansı olduğunu göstermektedir.

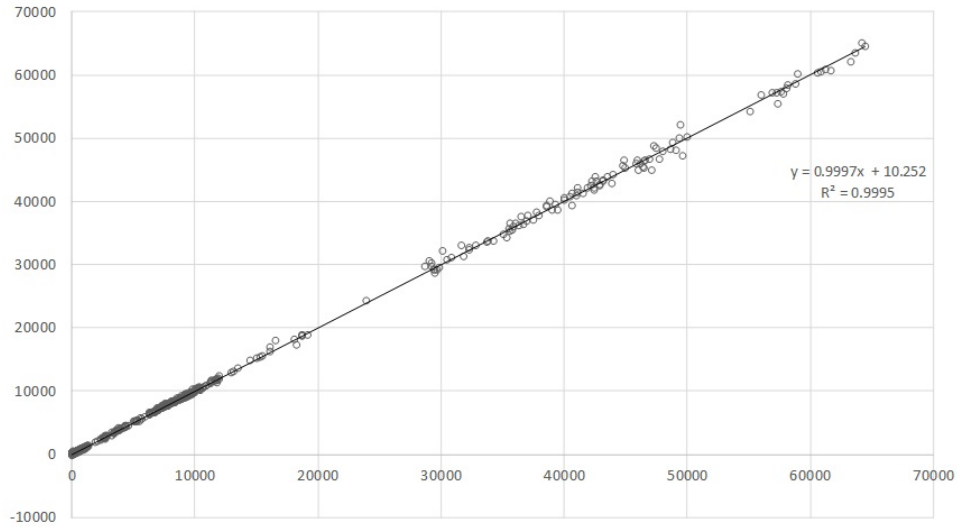
Model3 sonuçları

Model3 ün, Model1 ve Model2 den farkı, elde ettiğimiz verilerden a çıkış fiyatı, en düşük-en yüksek fiyat verilerini çıkardığımızda GPR modelinin performansını g örnek istedik. Model3 den elde edilen sonuçlar **Çizelge 4.4** de verilmiştir.

Kernel	Basis Function	R	MAE	MSE	RMSE	R ²	AdjRs _q	MAPE
Exponential	Constant	0.95582	1389.6	1.8649e+07	4281	0.91301	0.91148	155.16
Exponential	None	0.95854	1374.6	1.7673e+07	4179.5	0.91742	0.91596	155.56
Exponential	Linear	0.99973	140.05	1.1677e+05	340.2	0.99946	0.99945	1329.7
Exponential	Pure Quadratic	0.99976	141.8	1.0264e+05	319.69	0.99952	0.99951	2673.4
Squared Exponential	Constant	0.90436	369.09	5.8318e+07	6367.1	0.72923	0.72944	1085.2
Matern 5/2	Constant	0.96735	205.42	1.7183e+07	2474.7	0.92226	0.92089	931.83
Rational Quadratic	Constant	0.66497	1873.1	4.43e+09	41918	-19.594	-19.958	1243.9

Çizelge 4.4 eksGPR Performans Sonuçları

Model3 sonuçların grafiği **Şekil 4.4** deki gibi elde edilir.



Şekil 4.4 eksGPR grafiği

Çizelge 4.4 teki AdjRs_q sonuçları dikkate alındığında exponential kernel fonksiyonu ve pure quadratic basis fonksiyonu modeli 0.99951 başarı oranıyla diğer parametrelerden daha iyi sonuç verdiği görülmüştür. En iyi performans veren GPR modelinin grafiği **Şekil 4.4** teki gibidir. Korelasyon grafiği doğrusal bir grafik olduğ undan model performansının iyi olduğu grafik üzerinde görülmektedir.

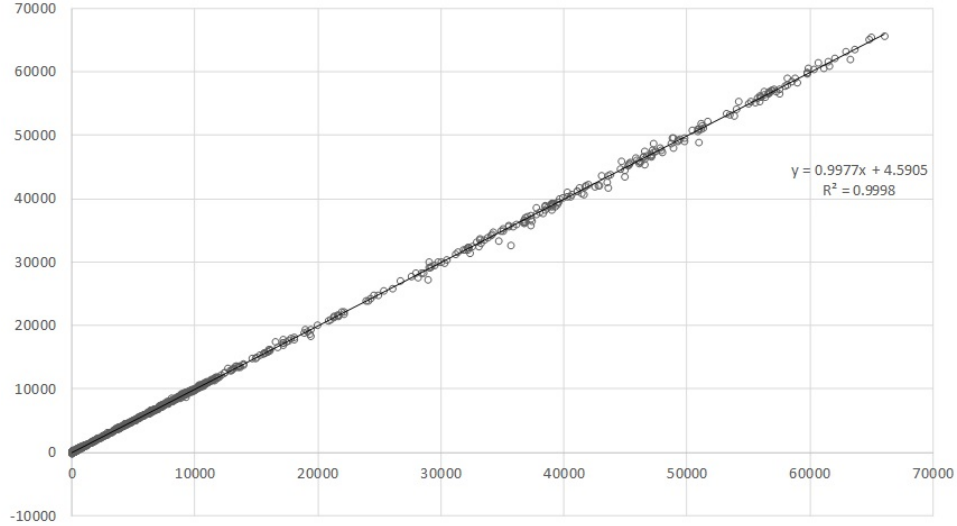
Model4 sonuçları

Model4, Model3 ten farklı olarak Model3 te kullandığımız verilere sentetik olarak SMOTE yöntemiyle biraz arttırdık ve performansını görmek istedik. Model4 den elde edilen sonuçlar **Çizelge 4.5** de verilmiştir.

Kernel	Basis Function	R	MAE	MSE	RMSE	R ²	AdjRsqr	MAPE
Exponential	Constant	0.98857	567.21	4.5819e+06	2137.7	0.97683	0.97662	61.1
Exponential	None	0.98864	565.85	4.5616e+06	2132.7	0.97693	0.97672	60.729
Exponential	Linear	0.99989	82.871	45725	213.51	0.99977	0.99977	1081.7
Exponential	Pure Quadratic	0.99988	88.144	47998	217.42	0.99976	0.99976	1902.7
Squared Exponential	Constant	0.91486	201.34	9.4034e+07	4516.9	0.53018	0.52606	913.84
Matern 5/2	Constant	0.9778	141.54	1.1194e+07	1683.5	0.94407	0.94358	1177
Rational Quadratic	Constant	0.91562	206.95	9.1081e+07	4591.7	0.54489	0.5409	1033.9

Çizelge 4.5 eksGPR-SMOTE Performans Sonuçları

Model4 sonuçlarının grafiği **Şekil 4.5** deki gibi elde edilir.



Şekil 4.5 eksGPR-SMOTE grafiği

Çizelge 4.5 teki AdjRsqr sonuçları dikkate alındığında exponential kernel fonksiyonu ve linear basis fonksiyonu modeli 0.99977 başarı oranıyla diğer parametrelerden daha iyi sonuç verdiği görülmüştür. En iyi performans veren GPR modelinin grafiği **Şekil 4.5** teki gibidir.

5. SONUÇ VE ÖNERİLER

Çalışmamızda kripto paralardan BTC üzerinde inceleme yapılmıştır. Kripto paraların geliştirilmesi için altında yatan dağınık yapılar ve indikatörler ile ilgili kavramlar hakkında bilgilere yer verilmiştir.

Çalışmanın temel amacı bir önceki verilerden faydalanılarak kripto para fiyatları için bir tahmin modeli oluşturmaktır. Bu kapsamda BTC kripto para birimine ait 25/07/2010- 05/06/2022 tarihleri arasındaki fiyatlar kullanılarak tahmin işlemleri için Gaussian Process Regression (GPR) modeli ve SMOTE yöntemi uygulanarak GPR modeli kullanılmıştır. Kfold ile 5 alt kümeye ayrılarak eğitilmiş ve test edilmiştir. Eğitim sürecinde, yukarıda açıklandığı gibi, en iyi modeli oluşturmak için Model1, Model2, Model3 ve Model4 olmak üzere dört farklı model yapımız ve kullanılacak GPR modelinin kernel ve basis fonksiyonları vardı. En iyi modeli seçmek için öncelikle her bir fonksiyonu eğitim veri seti ile eğitmemiz, ardından değerlendirme metrikleriyle en iyi fonksiyonu belirlememiz ve son olarak modeli test veri seti ile değerlendirmemiz gerekir. Eğitim sürecini tamamladıktan sonra Model1, Model2, Model3 ve Model4 teki sonuçlara baktığımızda Model1, Model2 ve Model3 te exponential kernel fonksiyonu ve pure quadratic basis fonksiyonu modeli, Model4 te exponential kernel fonksiyonu ve linear basis fonksiyonu modelinin en iyi performansı verdiği görülmüştür. Tüm modellerin en iyi performansa sahip parametrelerinin AdjRsquared değerleri birbirine yakın iyi bir performans gösteriyor. Değerlendirme metriklerinden AdjRsquared değeri baz alınarak modeller arasından Model2 0.99986 başarı oranıyla en iyi performansa sahip olduğu görülmüştür. Elde edilen performans kriterlerinden MAPE, RMSE, MAE, AdjRsquared, R, R^2 , MSE incelendiğinde SMOTE kullanılan modelin mevcut veriler üzerinde başarılı olduğu gözlenmiştir.

KAYNAKLAR DİZİNİ

- Abo-Hammour, Z., Alsmadi, O., Momani, S., & Arqub, O. A. (2013). A Genetic Algorithm Approach for Prediction of Linear Dynamical Systems. *Mathematical Problems in Engineering*. <https://doi.org/10.1155/2013/831657>
- Abo-Hammour, Z., Arqub, O. A., Momani, S., & Shawagfeh, N. (2014). Optimization Solution of Troesch's and Bratu's Problems of Ordinary Type Using Novel Continuous Genetic Algorithm. *Discrete Dynamics in Nature and Society*. <https://doi.org/10.1155/2014/401696>
- Aci, M., & Dogansoy, G. A. (2022). Demand forecasting for e-retail sector using machine learning and deep learning methods. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 37(3), 1325–1339. <https://doi.org/10.17341/gazimmfd.944081>
- Altunkaynak, B., & Esin, A. (2004). *The Genetic Algorithm Method for Parameter Estimation in Nonlinear Regression* (tech. rep. No. 2).
- Arqub, O. A., & Abo-Hammour, Z. (2014). Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm. *Information Sciences*, 279, 396–415. <https://doi.org/10.1016/j.ins.2014.03.128>
- Arqub, O. A., Abo-Hammour, Z., Momani, S., & Shawagfeh, N. (2012). Solving Singular Two-Point Boundary Value Problems Using Continuous Genetic Algorithm. *Abstract and Applied Analysis*, 2012, 1–25. <https://doi.org/10.1155/2012/205391>
- Arslan, M. E., & Kirci, P. (2021). Makine Öğrenmesi İle Borsa Analizi. *European Journal of Science and Technology*, (28), 1117–1120. <https://doi.org/10.31590/ejosat.1012785>
- Aslanyürek, M., & Mesut, A. (2021). Kümeleme Performansını Ölçmek için Yeni Bir Yöntem ve Metin Kümeleme için Değerlendirmesi. *European Journal of Science and Technology*. <https://doi.org/10.31590/ejosat.932938>

KAYNAKLAR DİZİNİ

- Atalay, M., & Çelik, E. (2017). Büyük Veri Analizinde Yapay Zekâ ve Makine Öğrenmesi Uygulamaları - Artificial Intelligence and Machine Learning Applications in Big Data Analysis. *Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 155–172. <https://doi.org/10.20875/makusobed.309727>
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE : Synthetic Minority Over-sampling Technique. *16*, 321–357.
- Chen, K. Y., & Wang, C. H. (2007). Support vector regression with genetic algorithms in forecasting tourism demand. *Tourism Management*, 28(1), 215–226. <https://doi.org/10.1016/j.tourman.2005.12.018>
- Dvoryatkina, S., & Golovin, D. (2022). Neural network technologies for analysis and risk assessment in forecasting the market of industrial financial instruments. *Journal of Physics: Conference Series*, 2176(1). <https://doi.org/10.1088/1742-6596/2176/1/012091>
- Eberhard, J., & Geissbuhler, V. (2000). *Konservative und operative therapie bei harninkontinenz, deszensus und urogenital-beschwerden* (Vol. 7).
- Erfanian, S., Zhou, Y., Razzaq, A., Abbas, A., Safeer, A. A., & Li, T. (2022). Predicting Bitcoin (BTC) Price in the Context of Economic Theories: A Machine Learning Approach. *Entropy*, 24(10), 1–29. <https://doi.org/10.3390/e24101487>
- Esmer, S., Uçar, M. K., Çil, İ., & Bozkurt, M. R. (2020). Parkinson Hastalığı Teşhisi İçin Makine Öğrenmesi Tabanlı Yeni Bir Yöntem. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 1877–1893. <https://doi.org/10.29130/dubited.688223>
- Gultepe, Y. (2019). A comparative assessment on air pollution estimation by machine learning algorithms. *European Journal of Science and Technology*, 16, 8–15.
- He, H., & Garcia, E. A. (2009). Learning from Imbalanced Data. *21*(9), 1263–1284.

KAYNAKLAR DİZİNİ

- Heo, Y., & Zavala, V. M. (2012). Gaussian process modeling for measurement and verification of building energy savings. *Energy and Buildings*, 53, 7–18. <https://doi.org/10.1016/j.enbuild.2012.06.024>
- Kayaalp, G. T., Güney, M. Ç., & Cebeci, Z. (n.d.). *Çoklu Doğrusal Regresyon Modelinde Değişken Seçiminin Zootekniye Uygulanışı* (tech. rep.).
- Kilimci, H., Yildirim, M., & Kilimci, Z. H. (2021). The Prediction of Short-Term Bitcoin Dollar Rate (BTC/USDT) using Deep and Hybrid Deep Learning Techniques. *ISMSIT 2021 - 5th International Symposium on Multidisciplinary Studies and Innovative Technologies, Proceedings*, 633–637. <https://doi.org/10.1109/ISMSIT52890.2021.9604741>
- Lee, M. C., Chang, J. W., Yeh, S. C., Chia, T. L., Liao, J. S., & Chen, X. M. (2022). Applying attention-based BiLSTM and technical indicators in the design and performance analysis of stock trading strategies. *Neural Computing and Applications*, 34(16), 13267–13279. <https://doi.org/10.1007/s00521-021-06828-4>
- Liang, Y., & Unwin, J. (2022). COVID-19 forecasts via stock market indicators. *Scientific Reports*, 12(1), 1–13. <https://doi.org/10.1038/s41598-022-15897-x>
- Liu, H. (2021). A research on stock forecasting based on principal component LSTM model. *2021 IEEE International Conference on Advances in Electrical Engineering and Computer Applications, AEECA 2021*, 684–688. <https://doi.org/10.1109/AEECA52519.2021.9574128>
- Liu, K., Hu, X., Wei, Z., Li, Y., & Jiang, Y. (2019). Modified Gaussian Process Regression Models for Cyclic Capacity Prediction of Lithium-Ion Batteries. *IEEE Transactions on Transportation Electrification*, 5(4), 1225–1236. <https://doi.org/10.1109/TTE.2019.2944802>
- Livieris, I. E., Kiriakidou, N., Stavroyiannis, S., & Pintelas, P. (2021). An Advanced CNN-LSTM Model for Cryptocurrency Forecasting. <https://doi.org/10.3390/electronics>

KAYNAKLAR DİZİNİ

- López, V., Fernández, A., García, S., Palade, V., & Herrera, F. (2013). An insight into classification with imbalanced data : Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250, 113–141. <https://doi.org/10.1016/j.ins.2013.07.007>
- Madan, I., Saluja, S., Zhao, A., Madan, I., Saluja, S., & Zhao, A. (2015). Automated Bitcoin trading via machine learning algorithms. Available via DIALOG. *The Weizmann Institute of Science*, 1–5.
- Mathworks, C. (2016). Statistics and Machine Learning Toolbox™ User ' s Guide R 2016 a.
- McNally, S., Roche, J., & Caton, S. (2018). Predicting the Price of Bitcoin Using Machine Learning. *Proceedings - 26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2018*, 339–343. <https://doi.org/10.1109/PDP2018.2018.00060>
- Metin, S. (2021). Kripto Para Fiyatlarının Regresyon Analizi Yöntemleri ile Tahmini: Bitcoin, Ethereum ve Ripple. 2. Uluslararası Sosyal Bilimler ve İnovasyon Kongresi.
- Mohapatra, S., Mukherjee, R., Roy, A., Sengupta, A., & Puniyani, A. (2022). Can Ensemble Machine Learning Methods Predict Stock Returns for Indian Banks Using Technical Indicators? *Journal of Risk and Financial Management*, 15(8). <https://doi.org/10.3390/jrfm15080350>
- Özen, N. S., Saraç, S., & Koyuncu, M. (2021). COVID-19 Vakalarının Makine Öğrenmesi Algoritmaları ile Tahmini: Amerika Birleşik Devletleri Örneği. *European Journal of Science and Technology*. <https://doi.org/10.31590/ejosat.855113>
- Pat, S., Çelik, Ö., Odabaş, A., & Korkmaz, Ş. (2022). Optik Optical properties of Nb 2 O 5 doped ZnO nanocomposite thin film deposited by thermionic vacuum arc. 258(March). <https://doi.org/10.1016/j.ijleo.2022.168928>

KAYNAKLAR DİZİNİ

- Phaladisailoed, T., & Numnonda, T. (2018). Machine learning models comparison for bitcoin price prediction. *Proceedings of 2018 10th International Conference on Information Technology and Electrical Engineering: Smart Technology for Better Society, ICITEE 2018*, 506–511. <https://doi.org/10.1109/ICITEED.2018.8534911>
- Provost, F. (2000). Machine Learning from Imbalanced Data Sets 101 Extended Abstract.
- Sevli, O. (2019). Göğüs Kanseri Teşhisinde Farklı Makine Öğrenmesi Tekniklerinin Performans Karşılaştırması. *European Journal of Science and Technology*, 176–185. <https://doi.org/10.31590/ejosat.553549>
- Shankhdhar, A., Singh, A. K., Naugraiya, S., & Saini, P. K. (2021). Bitcoin Price Alert and Prediction System using various Models. *IOP Conference Series: Materials Science and Engineering*, 1131(1), 012009. <https://doi.org/10.1088/1757-899x/1131/1/012009>
- So, B., & Boucher, J.-p. (2021). Synthetic Dataset Generation of Driver Telematics. (2015), 1–19.
- Sun, M., & Glabadanidis, P. (2022). Can technical indicators predict the Chinese equity risk premium? *International Review of Finance*, 22(1), 114–142. <https://doi.org/10.1111/irfi.12344>
- Şahin, E. E. (2020). Determination of Factors Affecting Bitcoin Price By Mars Method. 4(1), 171–184.
- Tuğ Karoğlu, T., & Okut, H. (2020). Classification of the Placement Success in the Undergraduate Placement Examination According to Decision Trees with Bagging and Boosting Methods. *Cumhuriyet Science Journal*, 41(1), 93–105. <https://doi.org/10.17776/csj.544639>

KAYNAKLAR DİZİNİ

- Wu, S., Liu, Y., Zou, Z., & Weng, T. H. (2022). S-I-LSTM: stock price prediction based on multiple data sources and sentiment analysis. *Connection Science*, 34(1), 44–62. <https://doi.org/10.1080/09540091.2021.1940101>
- Yang, J., De Montigny, D., & Treleaven, P. (2022). ANN, LSTM, and SVR for Gold Price Forecasting. *2022 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics, CIFEr 2022 - Proceedings*. <https://doi.org/10.1109/CIFEr52523.2022.9776141>
- Yergök, D., & Acı, M. (2019). Toplu Yemek Üretiminde Günlük Talep Tahmini için Alternatif Bir Yaklaşım: Öğrenci Regresyon. *European Journal of Science and Technology*, 64–73. <https://doi.org/10.31590/ejosat.636368>
- Zaj, M. M., Samavi, M. E., & Koosha, E. (2022). Measurement of Bitcoin Daily and Monthly Price Prediction Error Using Grey Model, Back Propagation Artificial Neural Network and Integrated model of Grey Neural Network. *Adv. Math. Fin. App*, 2022(3), 535–553. <https://doi.org/10.22034/AMFA.2020.1881110.1315>
- Zewdu, T. (1998). Prediction of HIV Status in Addis Ababa using Data Mining Technology.
- Zhang, N., Xiong, J., Zhong, J., & Leatham, K. (2018). Gaussian process regression method for classification for high-dimensional data with limited samples. *8th International Conference on Information Science and Technology, ICIST 2018*, 358–363. <https://doi.org/10.1109/ICIST.2018.8426077>
- Zhang, Y., & Xu, X. (2020). Yttrium barium copper oxide superconducting transition temperature modeling through gaussian process regression. *Computational Materials Science*, 179, 109583. <https://doi.org/10.1016/j.commatsci.2020.109583>