T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ TEKNİKLERİ İLE HABER KAYNAKLARI KULLANILARAK KRİPTO PARA - BİTCOİN TAHMİNİ

Hatice EKENEK

YÜKSEK LİSANS

Bilgisayar Mühendisliği Anabilim Dalı

Ocak-2025
KONYA

TEZ KABUL VE ONAYI

Hatice Ekenek tarafından hazırlanan "Makine Öğrenmesi Teknikleri İle Haber Kaynakları Kullanılarak Kripto Para - Bitcoin Tahmini" adlı tez çalışması 14/01/2025 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS olarak kabul edilmiştir.

Jüri Üyeleri	**İmza**
Başkan Prof. Dr. Halife KODAZ (Konya Teknik Üniversitesi)	……..
Danışman Dr. Öğr. Üyesi Ali SAĞLAM (Konya Teknik Üniversitesi)	……..
Üye Dr. Öğr. Üyesi Ahmet ÖZKIŞ (Necmettin Erbakan Üniversitesi)	……..

Yukarıdaki sonucu onaylarım.

Prof. Dr. Mevlüt UYAN
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza

Hatice EKENEK

Tarih:14.01.2025

ÖZET

YÜKSEK LİSANS

MAKİNE ÖĞRENMESİ TEKNİKLERİ İLE HABER KAYNAKLARI KULLANILARAK KRİPTO PARA - BİTCOİN TAHMİNİ

Hatice EKENEK

Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Ali SAĞLAM

2025, 92 Sayfa

Jüri
Dr. Öğr. Üyesi Ali SAĞLAM
Prof. Dr. Halife KODAZ
Dr. Öğr. Üyesi Ahmet ÖZKIŞ

Gelişen günümüz şartlarında teknolojinin ve internetin gelişimi, çeşitli bilimsel gelişmelerin ortaya çıkması, insanların yaşamında önemli yeri olan parayı da etkilemiştir. Para, dijital dünyada sanal para birimi olan Bitcoin için farklı bir birim olarak yerini almıştır.

Günümüzde yaygın olarak bilinen sanal paralardan birisi olan Bitcoin' in herhangi bir kuruma, devlete bağlı olmaması, transferinin gerçek paralara göre masrafsız, kolay ve hızlı olması gün geçtikçe dünyanın her yerinde Bitcoin'e olan ilgiyi artırmaktadır.

Tez çalışması kapsamında, makine öğrenmesi teknikleri kullanılarak kripto para birimi olan Bitcoin'in fiyat tahmini incelenmektedir. Çalışma bununla birlikte, Bitcoin fiyat tahminine etki eden haber kaynaklarını, sosyal medya verilerini ve piyasa duyarlılığını göz önünde bulundurarak, bu unsurların makine öğrenmesi modelleri ile senkronizasyonunu ele almaktadır.

https://www.kaggle.com internet sitesinden alınan 2021-2022 yılları arasındaki tweetler ve Yahoo Finance apisi ile alınan 2021-2022 yılları arasındaki Bitcoin açılış, kapanış, en yüksek değer, en düşük değer ve hacim değerleri ile iki yıllık veri setleri oluşturulmuştur. Python programlama dili ile makine öğrenmesinde tweetlerin duygu durumları doğal dil işleme aracı olan TextBlob kullanılarak değerlendirilmiş ve bu duygu durumları bitcoin fiyat giriş verileri ile birleştirilerek, makine öğrenmesi yöntemlerinden XGBoost, Rastgele Orman, LSTM ve Doğrusal Regresyon ile bir sonraki günün açılış fiyatı tahmin edilmiştir. MAPE= 0.052624 hata oranı oldukça düşük olduğu görülmüştür. R^2 = 0.999999, 1' e yakın bir değer bulunarak performans metriklerinde doğruluğunun yüksek olduğu bulunmuştur. Çalışma sonucunda Doğrusal Regresyon algoritmasının daha başarılı sonuçlar verdiği gözlemlenmiştir. Veri setinde olmayan gerçek değerler ile test edilen uygulamanın gerçek değerlere yakın sonuçlar verdiği ortaya çıkmıştır.

Anahtar Kelimeler: Dijital, Sanal Para, Bitcoin, Makine Öğrenmesi, Veri seti

ABSTRACT

MS THESIS

CRYPTO CURRENCY - BITCOIN PREDICTION USING MACHINE LEARNING TECHNIQUES AND NEWS SOURCES

Hatice EKENEK

Konya Technical University
Institute of Graduate Studies
Department of Computer Engineer

Advisor: Asst. Prof. Dr. Ali SAĞLAM

2025, 92 Pages

Jury
Advisor Asst. Prof. Dr. Ali SAĞLAM
Prof.Dr. Halife KODAZ
Asst. Prof. Dr. Ahmet ÖZKIŞ

In today's developing conditions, the development of technology and the internet, the emergence of various scientific developments have also affected money, which has an important place in people's lives. Money has taken its place as a different unit in the digital world for Bitcoin, a virtual currency.

Bitcoin, which is one of the widely known virtual currencies today, is not affiliated to any institution or state, and its transfer is costless, easy and fast compared to real money, which increases the interest in Bitcoin all over the world day by day.

This thesis examines the price prediction of Bitcoin, a cryptocurrency, using machine learning techniques. The study also considers news sources, social media data and market sentiment that affect Bitcoin price prediction and synchronizes these factors with machine learning models.

Two-year datasets were created with tweets between 2021 and 2022 from https://www.kaggle.com and Bitcoin opening, closing, high, low and volume values between 2021 and 2022 from Yahoo Finance API. In machine learning with Python programming language, the sentiment states of tweets were evaluated using TextBlob, a natural language processing tool, and these sentiment states were combined with bitcoin price input data and the opening price of the next day was predicted with machine learning methods XGBoost, Random Forest, LSTM and Linear Regression. MAPE= 0.052624 error rate was found to be quite low. $R^2 = 0.999999$, a value close to 1, was found to have high accuracy in performance metrics. As a result of the study, it was observed that the Linear Regression algorithm gave more successful results. The application tested with real values that were not in the data set gave results close to the real values.

Keywords: Digital, Virtual Currency, Bitcoin, Machine Learning, Data Set

ÖNSÖZ

Çalışmam esnasında benden yardımlarını esirgemeyen, tüm süreç boyunca yönlendirmelerde bulunan ve akademik gelişimime önemli katkılar sağlayan değerli hocam Dr. Öğr. Üyesi Ali SAĞLAM' a, tez önerilmesi konusunda yenilikçi fikirleri ile gelişimime katkı sağlayan rahmetli Doç. Dr. Üyesi Barış KOÇER'e ve tüm çalışmam esnasında fedakârlıklarını esirgemeyen, eşime, çocuklarıma ve arkadaşım meslektaşım Dr. Öğr. Üyesi Sümeyra Büşra ŞENGÜL'e teşekkür etmeyi bir borç bilirim.

Hatice EKENEK
KONYA-2025

İÇİNDEKİLER

ÖZET ..... **iv**

ABSTRACT ..... **v**

ÖNSÖZ ..... **vi**

İÇİNDEKİLER ..... **vii**

SİMGELER VE KISALTMALAR ..... **ix**

1. GİRİŞ **1**

2. KAYNAK ARAŞTIRMASI **3**

3. KRİPTO PARA-BİTCOİN **8**

3.1. Kripto Para – Bitcoin' in Diğer Paralardan Farkları 8
3.1.1. Nitelik ve şekil 8
3.1.2. İhraç ve yönetim 9
3.1.3. Değer temeli 9
3.1.4. Yasal statü 9
3.1.5. Güvenlik ve işlem kuralları 9
3.1.6. Ulaşılabilirlik ve kapsayıcılık 9
3.1.7. Değişkenlik 10
3.1.8. Şeffaflık ve Anonimlik 10
3.2. Kripto Para- Bitcoin 10

4. MAKİNE ÖĞRENMESİ **13**

4.1. Terminoloji 13
4.2. Makine Öğrenmesi Türleri 16
4.2.1. Gözetimli öğrenme 16
4.2.2. Gözetimsiz öğrenme 19
4.2.3. Yarı gözetimli öğrenme 19
4.2.4. Takviyeli Öğrenme 20
4.3. Yaygın Olarak Kullanılan Makine Öğrenmesi Algoritmaları 21
4.3.1. XGBoost Algoritması 22
4.3.2. Karar Ağaçları (Decision Trees) 24
4.3.3. Naive Bayes Algoritması 26
4.3.5. Lojistik Regresyon (Logistic Regression) 27
4.3.6. Destek Vektör Makineleri (Support Vector Machines - SVM) 31
4.3.7. Long–Short-Term Memory (LSTM) Algoritmaları 33
4.3.8. Doğrusal Diskriminant Analizi (Linear Discriminant Anlysis) Algoritmaları 36
4.3.9. En Yakın Komşular(K-Nearest Neighbors) 37
4.3.10. Kümeleme Algoritmaları 39
4.3.11. Doğrusal Regresyon(Linear Regrassion) Algoritmaları 42
4.3.12. Rasgele Orman (Random Forest) Algoritması 45

4.4. NLP (Doğal Dil İşleme) Nedir? 45
4.4.1. NLP Seviyeleri 46
4.4.1. NLP Görevleri 46
4.5. Performans Ölçümleri 47
4.5.1. MAPE (Mean Absolute Percentage Error - Ortalama Mutlak Yüzde Hatası) 48
4.5.2. MAE (Mean Absolute Error - Ortalama Mutlak Hata) 48
4.5.3. MSE (Mean Squared Error - Ortalama Kare Hata) 48
4.5.4. R^2 (R-Squared - Determinasyon Katsayısı) 48

5. MATERYAL VE YÖNTEM 50

5.1. Materyal 50
5.1.1. Veri Setleri 50
5.1.2. Yazılım ve Araçlar 54
5.2. Yöntem 54
5.2.1. Veri Toplama 54
5.2.2. Veri Ön İşleme 54
5.2.3. Model Geliştirme 55
5.2.4. Model Değerlendirme 55

6. ARAŞTIRMA SONUÇLARI VE TARTIŞMA 56

6.1. Uygulama 56
6.1.1. Ön İşleme 56
6.1.2. Analiz ve Sınıflandırma 57
6.1.2. Makine Öğrenmesi Modelleri ile Fiyat Tahmini Sonuçları 57

7. SONUÇLAR VE ÖNERİLER 88

7.1. Sonuçlar 88
7.2. Öneriler 88

KAYNAKLAR 89

SİMGELER VE KISALTMALAR

Simgeler

β : Beta
λ : Lamda
∑ : Toplama

Kısaltmalar

ARMA	: AutoRegressive Moving Average (OtoRegresif Hareketli Ortalama)
BTC	: Bitcoin
CV	: Cross Validation
DBSCAN	: Density-Based Spatial Clustering of Applications with Noise
DT	: Decision Tree (Karar Ağacı)
DVM	: Destek Vektör Makinesi
GARCH	: Generalized AutoRegressive Conditional Heteroskedasticity
GMM	: Gaussian Mixture Model (Gaussian Karışım Modeli)
IRLS	: Iteratively Reweighted Least Squares
k-NN	: k-Nearest Neighbors (En Yakın Komşular Algoritması)
LDA	: Linear Discriminant Analysis (Doğrusal Ayırma Analizi)
LSTM	: Long Short-Term Memory (Uzun Kısa Süreli Bellek)
MAPE	: Mean Absolute Percentage Error (Ortalama Mutlak Yüzde Hatası)
MDS	: Multidimensional Scaling (Çok Boyutlu Ölçekleme)
ML	: Machine Learning (Makine Öğrenmesi)
NLP	: Natural Language Processing (Doğal Dil İşleme)
MSE	: Mean Squared Error (Ortalama Kare Hatası)
NER	: Named Entity Recognition (Adlandırılmış Varlık Tanıma)
PCA	: Principal Component Analysis (Temel Bileşen Analizi)
RF	: Rastgele Orman (Random Forest)
RMSE	: Root Mean Square Error (Karekök Ortalama Kare Hatası)
RNN	: Recurrent Neural Network (Tekrarlayan Sinir Ağı)
SVM	: Support Vector Machine (Destek Vektör Makinesi)
XGB	: XGBoost/eXtreme Gradient Boosting
UMAP	: Uniform Manifold Approximation and Projection

1. GİRİŞ

Bitcoin, 2008'deki ortaya çıkışından bu yana geleneksel finansal sistemlere meydan okuyan merkezi olmayan bir dijital para birimi olarak karşımıza çıkmaktadır (Yermack, 2024). Blok zinciri teknolojisi üzerine tasarlanan Bitcoin, işlemler için güvenli ve açık bir platform sunarak dünyanın dört bir yanındaki yatırımcıların büyük ilgisini çekmiştir. Bununla birlikte, Bitcoin fiyatı oldukça uçucu olup, onu hem potansiyel olarak kazançlı hem de daha riskli bir yatırım haline getirmektedir. Bu değişkenlik, Bitcoin fiyat hareketlerini anlamak ve tahmin etmek isteyen araştırmacılar ve ticaret ile uğraşan kişiler için zorluklar yaratmaktadır.

Son yıllarda, makine öğrenimi teknolojileri, karmaşık kalıpları ve eğilimleri analiz etme yetenekleri nedeniyle Bitcoin fiyat tahmini için giderek daha fazla kullanılmaktadır (Reddy ve diğ., 2022). Bu yöntemler, geçmiş verilerden öğrenerek ve piyasa dinamiklerindeki değişikliklere uyum sağlayarak Bitcoin fiyatının geleceğindeki yönü hakkında içgörü sunmayı amaçlamaktadır.

Kripto para piyasaları son yıllarda dünyanın dört bir köşesindeki yatırımcıların büyük ilgisini çeken, dinamik ve çok hızlı büyüyen finansal alanlardan birisi haline gelmiştir (Reddy ve diğ., 2022). Bitcoin son zamanlarda bu piyasadaki en yaygın ve popüler para birimlerinden birisi olarak görülmektedir. Ancak Bitcoin'in fiyatı diğer küresel finansal varlıklara nazaran spekülasyonlara daha açıktır (Yermack, 2024). Bu durum kripto para piyasasını tahmin etmeyi güçleştirmekle kalmıyor, aynı zamanda piyasanın yapısına ilişkin belirsizliklere de yol açmaktadır. Bu gibi belirsizlikler, yatırımcılar ve finansal analistler için isabetli tahmin yapmayı önemli hale getirmektedir.

Piyasadaki arz-talep dengesinin yanı sıra, Bitcoin fiyatları çeşitli haber kaynaklarından gelen bilgilerden de büyük ölçüde etkilenmektedir. Kripto para piyasalarındaki olumlu/olumsuz haberler ve tweetler fiyatlar üzerinde önemli bir etkiye neden olmaktadır (Huynh, 2023). Bu nedenle, Bitcoin fiyatlarındaki değişimlerin daha doğru bir şekilde tahmin edilebilmesi için haber kaynaklarının etkisi dikkate alınmış olmalıdır.

Tez kapsamında, sosyal medyada paylaşılan haber ve tweetlerin Bitcoin fiyatlarının tahminine etkisi incelendi ve bu tahminleri makine öğrenmesi tekniklerini kullanarak gerçekleştirildi. Ayrıca haber kaynaklarının Bitcoin fiyatları üzerindeki etkisini analiz ederek, farklı makine öğrenmesi tekniklerinin fiyat tahminlerindeki

başarı oranları ve hata oranları kıyaslanarak, ön işlemler daha nitelikli hale getirerek model performansının iyileştirmeleri yapıldı. Farklı makine öğrenmesi yöntemlerinin kripto para piyasasında uygulanabilirliği değerlendirilerek en doğru tahmin sonuçlarını verebilecek modellerin belirlenmesi sağlandı. Böylece Bitcoin fiyat tahminine yönelik daha güvenilir ve veri odaklı yaklaşımlar geliştirildi.

Araştırmada kullanılan veri setleri iki ana kaynaktan oluşmaktadır: Bitcoin ile ilgili haber verileri ve tarih bazında fiyat verileri olmaktadır. Bunlar haber verileri çeşitli finansal haber sitelerinden ve sosyal medya platformlarından toplanan ve doğal dil işleme (Natural Language Programming - NLP) teknikleri ile analiz edilerek nötr, pozitif veya negatif içerik olarak sınıflandırıldı. Bitcoin'in geçmiş fiyat verileri büyük kripto para borsalarından toplandı. Toplanan veriler ön işleme tabi tutularak analiz ve modelleme aşamaları için hazır hale getirildi.

Tez çalışması kapsamında, veriler üzerinde farklı makine öğrenimi teknikleri uygulanmış ve modellerin tahmin performansı çeşitli metrikler aracılığıyla değerlendirilmiştir. Bu analizler sonucunda Bitcoin fiyatlarını tahmin etmede en yüksek performansı sağlayan model ya da modeller belirlenmiş ve gelecekteki çalışmalara katkı sağlamak üzere veri analizinde kullanılmıştır.

Çalışmanın bir sonraki bölümünde, Bitcoin fiyat tahminine ilişkin mevcut literatürü gözden geçirerek bu alandaki boşluklar ve zorluklar vurgulanmaktadır.

Üçüncü ve dördüncü bölümlerde, çalışmanın içerisinde kullanılan veri kaynağı olan Bitcoin ve veriyi modelleyen algoritmaları kapsayan makine öğrenmesi tekniklerinden bahsedilmiştir.

Beşinci bölümde, veri seti toplama kaynaklarına, veri seti içerikleri, veriye uygulanılan işlemler ile bu veriyi doğal dil işleme yöntemleri ile nasıl işlediğimize dair bilgiler ve kullanılan algoritmalar verilmiştir.

Altıncı bölümde, uygulama ve uygulamanın çıktılarına değinilmektedir.

Son bölümde ise tez kapsamında elde edinilen sonuçlara dair yorumlar ve öneriler bulunmaktadır.

2. KAYNAK ARAŞTIRMASI

Öğretici öğrenme yöntemleri, temel makine öğrenimi yöntemlerini alana özgü etiketler ile eğiterek bir sınıflandırma modeli oluşturmaktadır. Pang ve diğ., makine öğrenimi tekniklerini kullanarak duygu sınıflandırmasının otomasyonunu ele almışlardır (Pang ve diğ., 2002). Naive Bayes, Maksimum Entropi ve Destek Vektör Makineleri gibi makine öğrenimi algoritmaları, metinlerin duygusal durumunu (olumlu veya olumsuz) sınıflandırmak için kullanmışlardır. Veri kümesi olarak film değerlendirmeleri seçilmiş ve otomatik olarak olumlu veya olumsuz olarak etiketlendirilmiştir. Sınıflandırma için unigramların (tek kelime), bigramların (iki kelime), kelime frekanslarının ve sıfatların varlığı gibi farklı özellikler incelenmiş ve "iyi değil" gibi olumsuz ifadeleri tanımlamak için bağlamsal etiketlendirme kullanılmıştır. En başarılı sonuçlar Destek Vektör Makineleri ile elde edilirken, unigramların varlığı genel olarak en etkili özellik olarak ortaya çıkmıştır. Bu yöntemlerde, insan tarafından oluşturulan kelime listelerinin sağladığı %64'lük doğruluk oranı %82,9'a kadar çıkarılmıştır. Duygu sınıflandırmasındaki en büyük zorluk, metinlerdeki duyguların içeriksel olarak ifade edilebilmesidir. Özellikle, bir metinde olumlu ve olumsuz ifadelerin bir arada bulunması veya tonun değişmesi (örneğin, başlangıçta olumlu başlayıp sonunda olumsuza dönmek) makine öğrenimi modellerini zora sokabilmektedir. Ayrıca, duygu sınıflandırmasının, genellikle kelime bazında ayırt edilmesi kolay olan konu sınıflandırmasından daha karmaşık olduğu görülmüştür. Bu bağlamda, daha yüksek doğruluk oranlarına ulaşmak için metinlerdeki anlatı yapılarını ve içeriksel ilişkilerini göz önünde bulunduran yöntemlere ihtiyaç duyulduğu vurgulanmaktadır (Pang ve diğ., 2002).

Daha sonraki yıllarda Pang ve Lee (2004) çalışmalarında, metinlerin (olumlu veya olumsuz) ifade ettiği duygusal ifadelerin sınıflandırılması için yeni bir makine öğrenimi yöntemi önermişlerdir. İlk olarak, belgelerdeki yalnızca öznel (duygusal) cümlelerin seçilmesi için bir özellik belirleyici kullanılmış ve daha sonra seçilen bu bölümler bir duygu sınıflandırıcısına girdi olarak verilmiştir. Özellik belirleme için grafik tabanlı "minimum kesme" yöntemleri kullanılmaktadır, böylece cümleler arasındaki bağlam ve benzerlik ilişkileri etkili bir şekilde değerlendirilmektedir. Naive Bayes ve Destek Vektör Makineleri (SVM) gibi geleneksel sınıflandırıcılarla yapılan kıyaslamalarda, önerilen yöntem öznel kısımların %40'ına kadarını ortadan kaldırmasına rağmen sınıflandırma başarısını (%82,8'den %86,4'e kadar) iyileştirmiştir.

Whitelaw ve diğ., film eleştirilerinde duygu sınıflandırmasını iyileştirmek için klasik "kelime torbası" yaklaşımını genişleten yenilikçi bir yöntem sunmuşlardır. Çalışma, metinlerin duygusal içeriğini daha iyi temsil etmek için "Değerlendirme Grupları" adı verilen bir takım özellikler sunmuştur. Bu gruplar, metindeki olumlu veya olumsuz ifadeleri, yoğunluklarını ve bağlamlarını göz önünde bulundurarak sınıflandırma işlemine dahil etmeyi amaçlamaktadır. Araştırma, bu özellikleri kullanan makine öğrenimi modellerinin standart "kelime çantası" tabanlı yöntemlerden daha yüksek doğruluk oranlarına ulaştığını göstermektedir (Whitelaw ve diğ., 2005).

He ve Zhou (2011) çalışmalarında, etiketli verilerin sınırlı olduğu durumlarda duygu analizinde model performansını iyileştirmek için kendi kendine eğitimin uygulanması için bir yöntem geliştirmişlerdir. Kendi kendine öğrenme, modelin ilk olarak küçük bir etiketlenmiş veri kümesiyle eğitildiği ve daha sonra etiketsiz veriler üzerinde tahmin yaparak ve bu tahminleri etiketlenmiş verilere ekleyerek genişletildiği bir eğitim sürecini kullanmaktadır. Bu sayede etiketli verilere olan bağımlılık azalır ve model etiketsiz verilerden öğrendikçe doğruluk seviyesi daha iyi duruma gelmektedir. Çalışmalarında, kendi kendine öğrenmenin duygu analizindeki performansı, özellikle etiketli verinin kısıtlı olduğu durumlarda test edilmektedir. Bulunan sonuçlar, kendi kendine eğitimin duygu analizindeki model doğrululuğunu önemli ölçüde iyileştirebileceğini göstermektedir. Bununla birlikte, model tarafından tahmin edilen etiketlerin doğruluğunun sürekli olarak gözlemlenmesi ve yanlış etiketlemelerin model üzerinde negatif etkilerinin bulunmamasına dikkat edilmesi gerektiği vurgulanmaktadır. Genel olarak, kendi kendine öğrenme, duygu analizi gibi doğal dil işleme uygulamalarında etkin bir şekilde kullanılabilen ve verimliliği artırabilen bir yöntem olduğu belirtilmektedir.

Asif ve diğ., çalışmalarında öğrenci performansının sadece notlar kullanılarak erken bir aşamada makul bir doğruluk oranıyla tahmin edilebildiğini göstermektedir. Naive Bayes, K- En Yakın Komşu (k- Nearest Neighbor – k-NN) ve Rastgele Orman (RO) gibi sınıflandırıcılar iyi tahmin sonuçları vermektedir. Ayrıca, iyi veya kötü öğrenci performansının göstergesi olabilecek belirli dersleri belirlemek için karar ağaçları uygulanmıştır. Çalışmada ayrıca öğrencilerin akademik performanslarının dört yıl boyunca nasıl ilerlediğini incelemek için kümeleme teknikleri kullanılmıştır. Çalışmalarında sonuçlar, öğrencilerin dört yıl boyunca genellikle benzer not örüntüleri sergilediğini ve iki ana gruba ayrıldığını (yüksek performanslı öğrenciler ve düşük performanslı öğrenciler) göstermektedir. Çalışma, bu bulguların erken müdahale ve

destek stratejileri geliştirmek için nasıl kullanılabileceğiyle ilgili bazı önerilerde bulunmaktadır. Elde edilen sonuçlar incelendiğinde en iyi performans gösteren sınıflandırıcının %83,65 doğru sınıflandırma yüzdesi ile Naive Bayes algoritması tarafından yapıldığı görülmektedir. Bu algoritmaya en yakın sonucu veren algoritma ise %74,04 ile En Yakın Komşu algoritmasıdır (Asif ve diğ., 2017).

Shahiri ve Husain, öğrencilerin performanslarını tahmin etmek için kullanılan benzer bir çalışmada veri madenciliği teknikleri konusunda genel bir bakış sağlamak için sistematik bir çalışma yapmışlardır. Çalışmanın amacı, eğitsel veri madenciliği tekniklerinden yararlanarak öğrencilerin akademik performanslarının tahmin edilmesi üzerine bir alan incelemesi yapmaktır. Çalışma, farklı yöntemleri değerlendirmekte ve hangi algoritmaların öğrenci performansını tahmin etmede daha başarılı olduğunu analiz etmektedir. Değerlendirilen yöntemlerin arasında Karar Ağaçları, Naive Bayes, K-En Yakın Komşu (KNN), Destek Vektör Makineleri (DVM) ve Yapay Sinir Ağları gibi algoritmalar yer almaktadır. Çalışma, bu yöntemlerin hangi tür veri setlerinde (örneğin akademik notlar, bunlarla ilgili demografik bilgiler veya diğer faktörler) daha başarılı olduğunu incelemişlerdir. Yöntemlerin başarısı incelendiğinde, her algoritmanın belirli veri setlerine ve uygulama durumlarına göre birbirinden farklı sonuç verdikleri görülmektedir. Örneğin, Karar Ağaçları ve Naive Bayes küçük ve yapılandırılmış veri setlerinde iyi sonuç verirken, Yapay Sinir Ağları daha kompleks ve büyük veri kümelerinde daha etkili sonuç vermiştir. Destek Vektör Makineleri genellikle daha yüksek doğruluk oranları ile ön plana çıkmaktadır. Ancak yöntemin seçimi ve başarısı, kullanılan veri tipi, özellik geliştirme süreçleri ve model inceleme kriterlerine bağlı olarak değişiklik göstermektedir. Çalışma, veri madenciliği tekniklerinin eğitim alanında uygulanma konusundaki potansiyeli ve etkili bir tahmin modeli geliştirilmesine yönelik gelecek araştırma alanlarını vurgulamaktadır (Shahiri ve Husain, 2015).

Zheshi ve diğ. (2020), çalışmalarında iki farklı yaklaşım ile veriler üzerinde tahmin algoritmalarını kullanmışlardır. Günlük Bitcoin fiyatları gibi düşük frekanslı ve yüksek boyutlu özelliklere sahip veriler için istatistiksel yöntemler daha başarılı bulunmuştur. Özellikle Lojistik Regresyon (LR) ve Lineer Diskriminant Analizi (LDA) algoritmaları, bu tür verilerde %66 ve %63.9 gibi yüksek doğruluk oranları elde etmelerini sağlamıştır. 5 dakikalık aralıklı Bitcoin fiyatları gibi yüksek frekanslı ve düşük boyutlu özelliklere sahip veriler için ise makine öğrenimi modelleri daha başarılı sonuçlar vermiştir. Uzun Kısa Dönemli Bellek (LSTM) algoritması, bu tür verilerde %67.2 gibi en yüksek doğruluk oranına ulaşmıştır. Diğer makine öğrenimi modelleri,

Rastgele Orman, XGBoost, Kuadratik Diskriminant Analizi ve Destek Vektör Makinesi, istatistiksel yöntemlerden daha iyi performans göstermiş, ancak LSTM kadar başarılı olamamıştır.

Chen ve diğ. (2020), Bitcoin fiyatlarını tahmin etmek için SVM ve LSTM modellerinin birleştirildiği hibrid bir model geliştirmiştir. Model, geçmiş fiyat verilerini ve teknik analiz göstergelerini kullanarak tahminler yapmıştır. Hibrid modelin, yalnızca LSTM ya da yalnızca SVM modellerine göre daha yüksek doğruluk sağladığı ve Bitcoin'in gelecekteki fiyat hareketlerini daha iyi tahmin ettiği bulunmuştur.

Chen (2023) , çalışmasında Bitcoin fiyat tahmini için zaman serileri ve makine öğrenimi yöntemlerini ele almaktadır. Rastgele orman regresyonu ve LSTM aracılığıyla bir sonraki gün Bitcoin fiyatı için yüksek tahmin doğruluğuna sahip bir algoritma modeli elde etmek ve hangi değişkenlerin Bitcoin fiyatı üzerinde etkisi olduğunu açıklamaya çalışmıştır. Diebold-Mariano testi ile rastgele orman regresyonunun tahmin doğruluğunun LSTM' ninkinden önemli ölçüde daha iyi olduğu kanıtlanamasa da, rastgele orman regresyonunun tahmin hataları RMSE ve MAPE, LSTM' ninkinden daha iyi olduğu sonucuna ulaşmıştır.

Cortez ve diğ. (2020), LSTM rastgele orman regresyonunu ele almışlardır. LSTM modelinin aşırı öğrenme etkisini azaltmak için her katman arasına Dropout katmanları eklemenin önemli olduğunu vurgulayarak farklı vazgeçme katsayıları tercihlerini değerlendirmişlerdir. Tanımlayıcı değişkenlerin tercihi konusunda, birçok çalışmada kullanılan genel makroekonomik değişkenlerin yanı sıra, Bitcoin blok zincirinin ana değişkenleri (kullanıcılar, madenciler ve borsalar) ve teknik göstergelere de yer verilmektedir. Gecikmeli parametrelerin Bitcoin fiyatı üzerindeki belirleyici etkisini incelemiş ve gecikme sayısı arttıkça rastgele orman regresyonunun MAPE değerinin de arttığı sonucuna ulaşmışlardır. Kripto para birimleri ve sabit para birimlerindeki piyasa likiditesinin tahminleri, iki geleneksel zaman serisi yöntemi olan bağımsız hareketli ortalama (ARMA) ve genelleştirilmiş bağımsız koşullu değişen varyans (GARCH) ile k-en yakın komşu (k-NN) yaklaşımı adı verilen bir makine öğrenimi algoritması arasında karşılaştırmışlardır. Piyasa değişkenliği, belirli bir tarih arasında üç kripto para biriminin (Bitcoin, Ethereum ve Ripple) ve farklı para birimlerinin alış-satış farklarının logaritmik oranları olarak ölçülmüştür. k-NN yaklaşımının, piyasa mikro yapısının karmaşıklığı sebebiyle kısa vadede bir kripto para biriminin piyasa değişkenliğini tespit etmek için ARMA ve GARCH modellerinden daha avantajlı olduğu görülmüştür. Geleneksel zaman serisi modelleri göz önüne

alındığında, ARMA modellerinin gelişmiş piyasalardaki fiili para birimlerinin likiditesini öngörmede iyi performans gösterdiği, GARCH modellerinin ise yükselen piyasalardaki fiili para birimleri için aynı şekilde performans gösterdiği görülmüştür. Bununla birlikte, sonuçlar KNN yönteminin kripto para birimleri ve fiili para birimlerinin alış-satış farklarının logaritmik oranlarını ARMA ve GARCH yöntemlerinden daha iyi tahmin edebildiğini göstermektedir.

Mudassir ve diğ. (2020), Bitcoin fiyatlarını tahmin etmek için makine öğrenimi ve derin öğrenme tekniklerinden yararlanmışlardır. Araştırmacılar ertesi gün, yedinci gün ve doksanıncı gün için tahminler oluşturmak amacıyla sınıflandırma ve regresyon yöntemleri geliştirmişlerdir. Geliştirilen modellerin gerçekleştirilebilir ve yüksek performans göstermekte olduğu, sınıflandırma modellerinin ertesi gün tahmini için %65'e varan doğruluk puanı ve yedinci ile doksanıncı gün tahmininin %62 ile %64 arasında doğruluk oranı elde ettiği saptanmıştır. Günlük fiyat tahminindeki hata yüzdesi %1,44 gibi oldukça düşük bir seviyede iken, yedi ila doksan günlük süreler için bu oran %2,88 ila %4,10 arasında değişmiştir.

Tripathi ve diğ. (2024), çalışmalarında Bitcoin fiyatlarındaki değişiklikleri tahmin etmede tutarlı bir şekilde en yüksek doğruluğu gösteren modeli belirlemek için çeşitli makine öğrenimi modellerini kullanmışlar ve karşılaştırmışlardır. Sonuç olarak, LSTM modeli, çalışmalarında Bitcoin fiyat tahmini için Lojistik Regresyondan daha başarılı bulunmuştur. LSTM' nin düşük hata değerleri ve yüksek doğruluk puanları, karmaşık zaman serisi verilerini analiz etme ve gelecekteki Bitcoin fiyatlarını tahmin etme becerisini göstermiştir.

3. KRİPTO PARA-BİTCOİN

Bitcoin, dünyanın herhangi bir yerinden başka bir yerine transfer yapılabilmesine imkan tanıyan ve hiç bir hükümet ya da kuruma bağlı olmayan bir para sistemidir. Dijital dünya içerisinde şifrelenerek saklandığı için kripto para veya sanal para olarak adlandırılmaktadır.

Mali Eylem Gücü, sanal parayı "…Sanal para, dijital olarak alınıp satılabilen ve bir değişim aracı ve/veya bir hesap birimi; ve/veya bir değer saklama aracı olarak işlev gören, ancak herhangi bir yargı alanında yasal ihale statüsüne (yani, bir alacaklıya teklif edildiğinde geçerli ve yasal bir ödeme teklifi) sahip olmayan dijital bir değer ifadesidir. Herhangi bir yargı alanı tarafından ihraç edilmez veya garanti edilmez ve yukarıdaki işlevleri yalnızca sanal para birimi kullanıcıları topluluğu içindeki anlaşma ile yerine getirir." olarak tanımlamaktadır (FATF, 2014).

Bitcoin, 2008 yılında Satoshi Nakamoto takma ismini kullanan bilinmeyen bir kişi veya grup tarafından yayınlanan bir makaleyle yaratılan açık kaynak kodlamaya dayanan uluslararası bir para birimidir. Aynı zamanda Bitcoin, dijital para ekosisteminin esasını oluşturan öğelerin bir kombinasyonudur. Bu sistemdeki katılımcılar arasında alışverişi sağlamak amacıyla kullanılmaktadır. Açık kaynak kodlu olduğu için herkes bu yazılımı kullanabilmekte ve destekleyebilmektedir. Bitcoin'in üretimi için belirli bir kurum yoktur. Hiç bir devlete ya da otoriteye bağlı değildir. Bu nedenle herhangi bir kontrol bulunmamaktadır (Nakamoto, 2008).

3.1. Kripto Para – Bitcoin' in Diğer Paralardan Farkları

Kripto paranın diğer paralardan farklı olarak tanımlanabilen bir takım temel özellikleri bulunmaktadır.

3.1.1. Nitelik ve şekil

Sadece elektronik ortamda var olan dijital değerlerdir.

Merkezi olmayan (Nakamoto, 2008), blok zincirine veya dağıtılmış defter teknolojisine (DLT) dayanmaktadır.

Örnekler arasında Bitcoin, Ethereum ve Litecoin bulunmaktadır.

3.1.2. İhraç ve yönetim

Genellikle madencilik veya pay alma gibi merkezi olmayan süreçlerle oluşturulmaktadır.

Yönetimi çoğunlukla kod ve mutabakat mekanizmaları (örn. iş kanıtı, hisse kanıtı) ile yönetilmektedir.

3.1.3. Değer temeli

Değer genellikle piyasadaki arz ve talebe göre belirlenmektedir.

Bitcoin gibi bazı kripto para birimleri sabit bir arz sınırı (21 milyon BTC) ile deflasyonist yapıdadır (Nakamoto, 2008).

3.1.4. Yasal statü

Evrensel olarak yasal ödeme aracı olarak tanınmamaktadır.

Düzenleyici statü yargı yetkisine göre değişmektedir (yasaklı, kısıtlı veya izinli).

Yasal belirsizlik ve gelişen düzenleyici çerçeveler bunların benimsenmesini etkileyebilmektedir.

3.1.5. Güvenlik ve işlem kuralları

İşlemler kriptografik yöntemlerle güvence altına alınmaktadır ve değişmez blok zinciri defterlerine işlenmektedir.

Sahte anonimdir, bir dereceye kadar kişisel gizlilik sağlamaktadır.

Çeşitli siber saldırılara, dolandırıcılıklara ve özel anahtarların yanlış yönetimine karşı korunmasızdır.

3.1.6. Ulaşılabilirlik ve kapsayıcılık

İnternet bağlantısı ve dijital cüzdanı olan herkes için erişim mümkündür.

Özellikle sağlam bankacılık sistemleri bulunmayan bölgeler için finansal katılımı teşvik etmektedir.

3.1.7. Değişkenlik

Spekülatif ticaret ve içsel destek eksikliği sebebiyle yüksek fiyat dalgalanması yaşanmaktadır.

Değer kısa dönemler içinde hızla dalgalanabilmektedir.

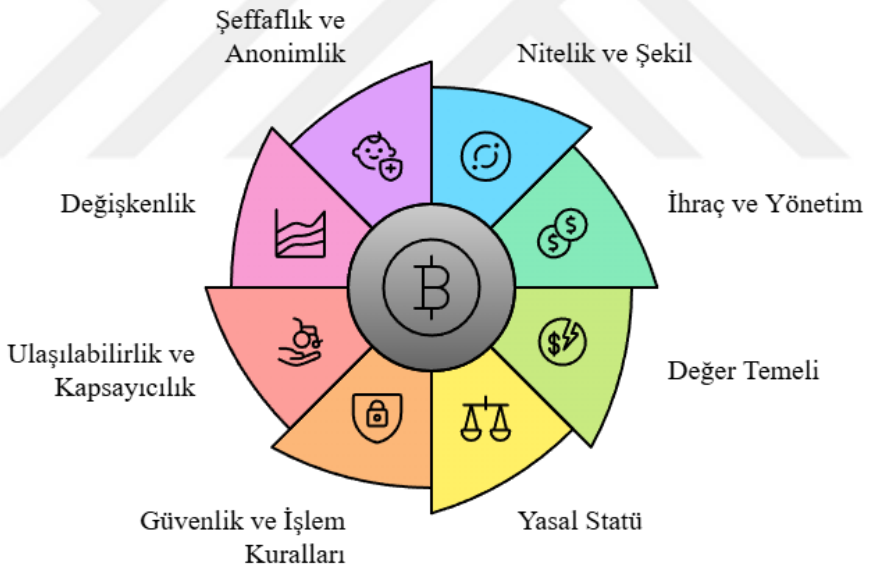
3.1.8. Şeffaflık ve Anonimlik

Şeffaftır: Tüm işlemler blok zincirinde halka açık bir şekilde kaydedilmektedir.

Sahte anonim: Cüzdan adreslerinin kişisel kimliklerle doğrudan bir bağlantısı yoktur.

Bazı gizlilik odaklı kripto paralar gelişmiş anonimlik sunmaktadır.

Şekil 3.1'de, tüm özelliklerin başlıklarına yer verilmektedir.



Şekil 3.1. Kripto Para- Bitcoinin Diğer Paralardan Farkları

3.2. Kripto Para- Bitcoin

Bitcoinler, birbirine "eş" iki veya daha fazla istemci arasında haberleşmenin sağlanması, veri paylaşımı yapılmasını sağlamak amacıyla kullanılan bir ağ protokolü

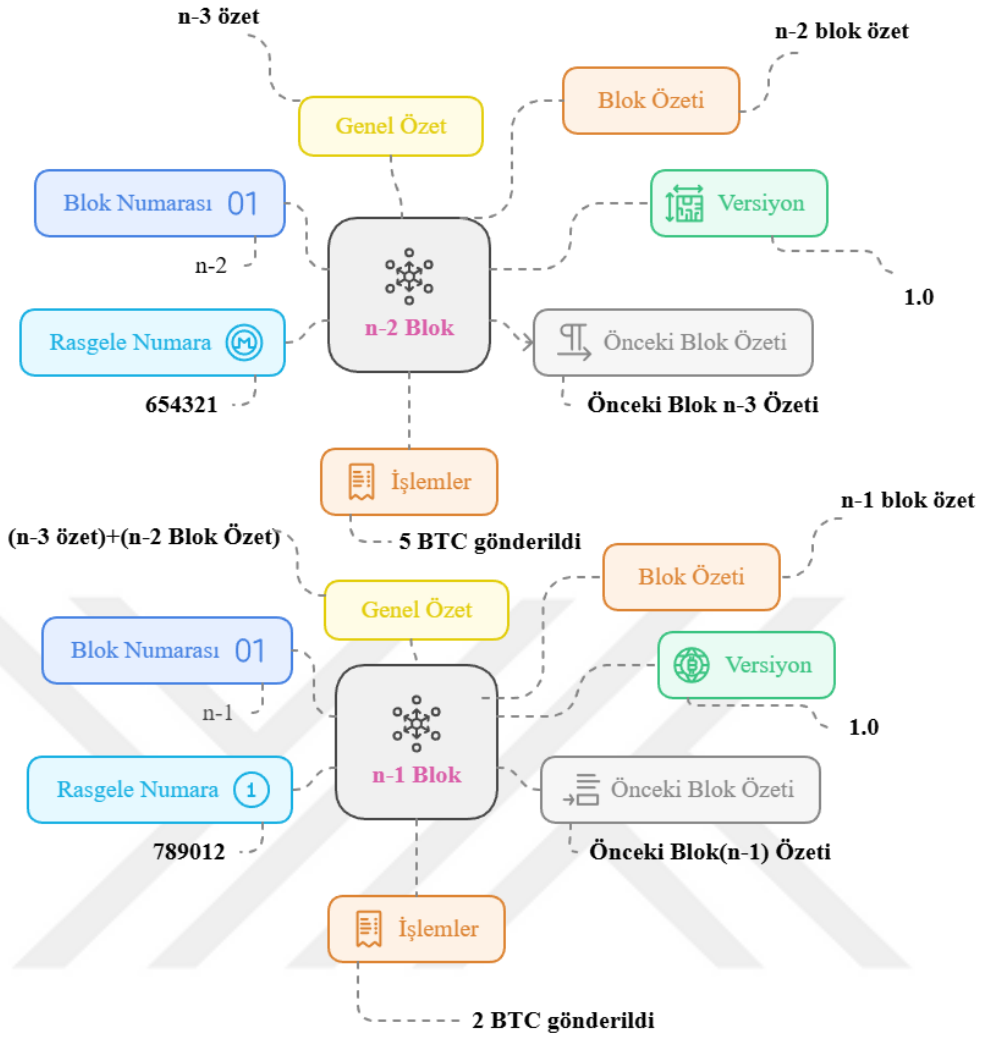
olan P2P ağı üzerinde işlem görmektedirler. Şekil 3.2'de, ağın işlem adımları görülmektedir.

Şekil 3.2. P2P ağ örneği

Tamamen sanal ortamda üretilen bitcoini kullanabilmek ve saklamak için bitcoin cüzdanı oluşturulmaktadır. Cüzdanın güvenliği ise Blockchain (Blok Zinciri) adı verilen yöntemle sağlanmaktadır. Blok zinciri yönteminde bloklar yapılan işlemleri ve bir önceki bloğun adresini tutarlar (Karaarslan ve Akbaş, 2017). Ağ içerisinde yapılan tüm işlemler bloklara kaydedilir. Kaydedilen işlemler silinemez ve değiştirilemez. Ağa dahil (Schober ve Vetter, 2021) olan kullanıcılar, tüm bilgilere erişebilmektedirler.

Blok zinciri yapıları, her bir bloğun nasıl yapılandırıldığını ve bloklar arasındaki bağlantıları göstermektedir. Her blok, önceki bloğun özetini tutarak zincirin güvenliğini ve bütünlüğünü sağlamaktadır.

Şekil 3.3, blok zincir yapısını göstermektedir. Blok zincir üzerinde tutulan bilgiler gösterilmektedir. Önceki blok(n-1) nolu bloktan bir işlem yapıldığında, diğer blok (n-2) üzerindeki akışı açıklanmaktadır.



Şekil 3.3. Blok zinciri yapısı

4. MAKİNE ÖĞRENMESİ

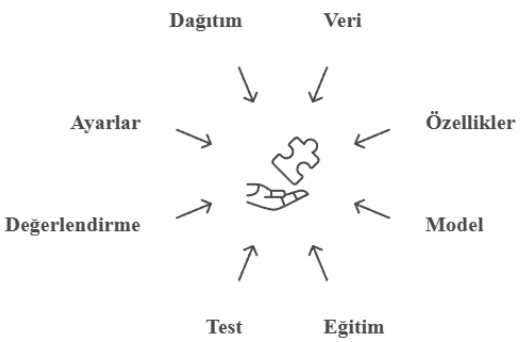
Makine Öğrenmesi (Machine Learning - ML)**,** matematiksel ve istatistiksel yöntemler kullanarak mevcut verilerden çıkarımlar yapan, bu çıkarımlarla bilinmeyene dair tahminlerde bulunan yöntemler bütünüdür.

Makine öğrenimi, özellikle, verilerdeki örüntüleri otomatik olarak saptayabilen ve sonrasında ortaya koyduğu örüntüleri gelecek verileri tahmin etmek ya da belirsizlik durumunda başka tür karar alma işlemlerini (nasıl daha fazla veri toplanacağını planlamak gibi) gerçekleştirebilmek için kullanan bir dizi yöntemler olarak tanımlanmaktadır (Murphy, 2012). Makine öğrenimi metotları, özel görevleri yerine getirmek üzere açıkça programlanmak yerine, daha fazla veri işleyerek, değişimlere uyum sağladıkça ve insan müdahalesi olmaksızın örüntüleri ortaya çıkardıkça kendi performanslarını geliştirmektedir.

4.1. Terminoloji

Makine öğrenmesi için temel bazı terimler bulunmaktadır. Öğrenme, deneyimi uzmanlığa ya da bilgiye dönüştürme sürecidir. Bir öğrenme algoritmasının girdisi, deneyimi temsil eden eğitim verileridir ve çıktı, genellikle bir görevi yerine getirebilen başka bir bilgisayar programı biçimini alan uzmanlıklardır (Shalev-Shwartz ve Ben-David, 2014).

Şekil 4.1'de, tüm bileşenler özetlenmektedir.



4.1. Makine öğrenmesinin bileşenleri

Makine öğrenimi algoritmasının çalıştırılmasının sonucunda, yeni bir rakam görüntüsü *x*'i girdi olarak alan ve hedef veri ile aynı şekilde kodlanmış bir çıktı veri *y* üreten bir fonksiyon *y(x)* olarak ifade edilebilmektedir. Buradaki *y(x)* fonksiyonunun kesin biçimi, öğrenme aşaması olarak da bilinen eğitim sürecinde, eğitim verileri temel alınarak belirlenmektedir. Model eğitildikten sonra, bir test kümesi oluşturduğu varsayılan yeni rakam görüntülerinin kimliğini belirleyebilmektedir. Eğitim için kullanılanlardan farklı olan yeni örnekleri doğru bir şekilde sınıflandırabilme yeteneği olarak bilinmektedir. Pratik uygulamalarda, girdi vektörlerinin değişkenliği, eğitim verilerinin tüm olası girdi vektörlerinin yalnızca küçük bir bölümünü oluşturabileceği şekilde olacağından, genelleme model tanımada merkezi bir hedef olmaktadır (Bishop, 2006).

Temel olarak makine öğrenmesi terimleri aşağıdaki gibidir:

Veri: Makine öğrenmesi, modelin öğrenmesi için gereken girdidir. Veriler genellikle yapılandırılmış (tablolar gibi) veya yapılandırılmamış (metin, resim gibi) olabilmektedir.

Model: Öğrenme sürecinin gerçekleştiği matematiksel veya algoritmik yapılar olarak adlandırılmaktadır. Modeller, verilerden öğrenerek belirli bir görevi yerine getirebilmektedir.

Öğrenme Algoritmaları: Verilerden bilgi çıkaran yöntemler olarak isimlendirilmektedir. Bu algoritmalar, modelin nasıl güncelleneceğini ve verilerden nasıl öğrenileceğini belirlemektedir.

Gözlemler: Öğrenmek ya da değerlendirmek için kullanılan her bir veri parçası olarak düşünülmektedir.

Örn: her bir e-posta bir gözlem olarak adlandırılmaktadır.

Özellikler: Bir gözlemi temsil eden genelde sayısal verilerden oluşmaktadır. Öğrenme sürecinde tahmin edilmek istenen değerler bütünü olarak düşünülmektedir.

Örn: E-posta'nın uzunluğu, tarihi, bazı kelimelerin varlığı, tez kapsamında işlenecek olan bitcoin fiyatlarında günlük olarak düşünülen açılış fiyat bilgisi, kapanış fiyat bilgisi örnek olarak verilebilmektedir.

Etiketler: Gözlemlere atfedilen kategorilerdir. Örn: spam, spam-değil.

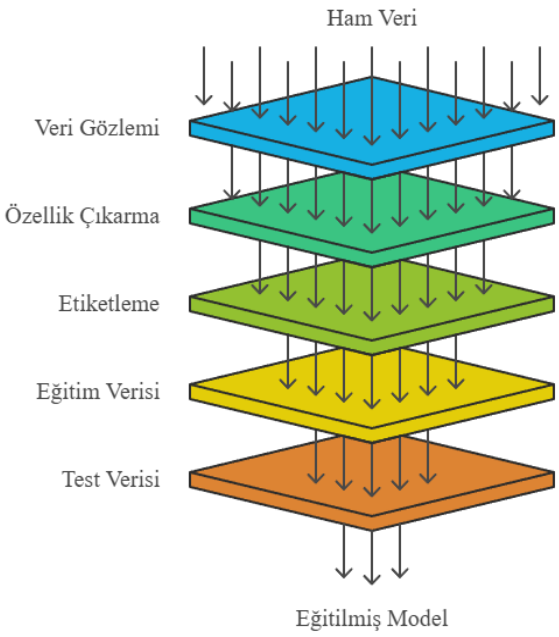
Eğitim Verisi: Algoritmanın öğrenmesi için sunulan gözlemler dizisi olarak adlandırılmaktadır. Algoritma bu veriye bakarak çıkarımlarda bulunur ve sonucunda model kurmaktadır.

Test Verisi: Algoritmanın şekillendirdiği modelin ne kadar gerçeğe yakın olduğunu test etmek için kullanılan veri seti olarak belirtilmektedir. Eğitim esnasında kullanılmaktadır.

Doğrulama Verisi: Veri çok ise, bir dizi modeli veya karmaşıklık parametreleri için bir dizi değerle belirli bir modeli eğitmek için mevcut verilerin bir kısmı kullanılır ve daha sonra bunları bazen doğrulama seti olarak adlandırılan bağımsız veriler üzerinde karşılaştırılır ve en iyi tahmin performansına sahip olanı seçilmektedir.

Çapraz Doğrulama(Cross-Validation - CV) : Örnekle açıklamak gerekirse, S = 4 durumu için açıklanan S-kat çapraz doğrulama tekniği, mevcut verilerin alınmasını ve S gruba bölünmesini kapsamaktadır (en basit durumda bunlar eşit büyüklüktedir). Gruplardan S - 1 tanesi daha sonra bir dizi modeli eğitmek için kullanılır ve bu modeller daha sonra kalan grup üzerinde tekrar değerlendirmeye tabi tutulmaktadır. Bu prosedür daha sonra hariç tutulan grup için tüm S olası seçenekler için tekrarlanmaktadır. Birçok uygulamada eğitim ve test için veri temini kısıtlı olabilir ve iyi modeller oluşturmak için mevcut verilerin mümkün olduğunca çoğunun eğitime yönelik olarak kullanılması istenmektedir. Bununla birlikte, doğrulama kümesi küçükse, tahmin performansının nispeten gürültülü bir tahmini elde edilir. Bu ikilem için bir çözüm çapraz doğrulama yöntemi kullanmaktadır.

Şekil 4.2'de, makine öğrenmesinin veriyi işleme süreci gösterilmektedir.



4.2. Makine öğrenmesi veri işleme süreci

4.2. Makine Öğrenmesi Türleri

Makine öğrenimi alanı, farklı öğrenme görevleri türleriyle ilgilenen çeşitli alt alanlara ayrılmıştır (Shalev-Shwartz ve Ben-David, 2014).

ML' nin türleri:

- Gözetimli öğrenme (Supervised Learning)
- Gözetimsiz öğrenim (UnSupervised Learning)
- Yarı gözetimli öğrenme (Semi-supervised learning)
- Takviyeli öğrenme (Reinforcement Learning)

4.2.1. Gözetimli öğrenme

Gözetimli öğrenme, etiketlenmiş eğitim verisinden bir fonksiyonun çıkarımıdır. Etiketler, algoritmaya gözlemlerin nasıl etiketlenmesi gerektiğini öğretmektedir. Örneğin içinde "para kazan" ifadesi geçiyorsa, spam demelisin gibi yol göstermelerde bulunmaktadır.

Gözetimli öğrenme yöntemlerinin düzgün bir şekilde çalışabilmesi için, eğitim için kullanılan veri setinin yeterince büyük olması gerekmektedir (Onan, 2015).

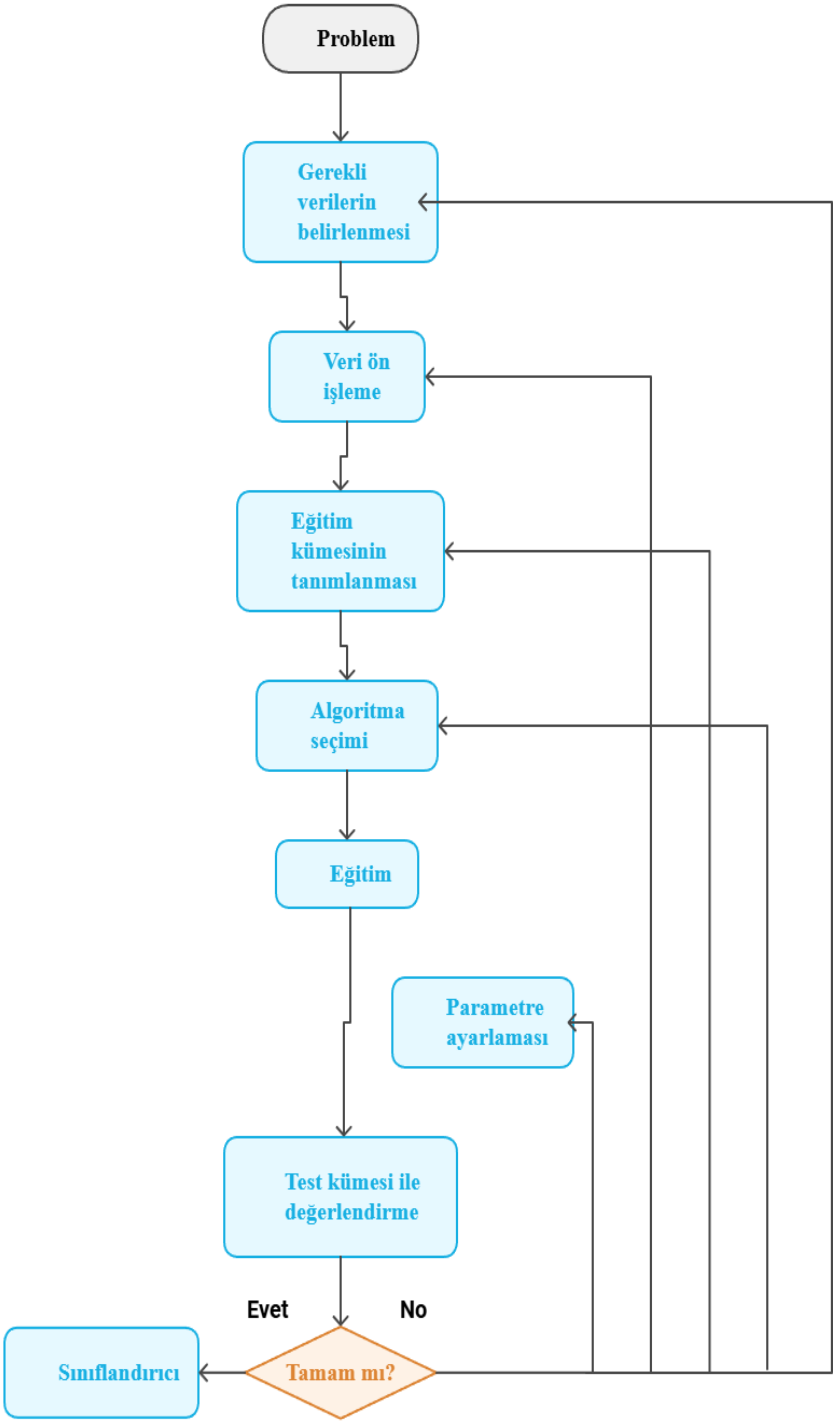
Gözetimli öğrenme, Sınıflandırma ve Regresyon olmak üzere iki başlıkta toplanmaktadır.

4.2.1.1. Sınıflandırma

Sınıflandırma, bir verinin belirli kategorilere veya sınıflara ayrılmasını amaçlayan bir denetimli öğrenme yöntemi olarak isimlendirilmektedir. Çıktı değişkenleri genellikle kategori veya ayrık bir yapıya sahiptir. Var/Yok, Spam/ Spam Değil, Hastalık türlerini belirleme gibi örnekler verilebilmektedir.

Denetimli öğrenme görevinin standart bir formülasyonu sınıflandırma problemidir: Öğrenenin, fonksiyonun birkaç girdi çıktı örneğine bakarak bir vektörü birkaç sınıftan birine eşleştiren bir fonksiyonu öğrenmesi gerekmektedir. Tümevarım yöntemi ile makine öğrenimi, eğitim setindeki örneklerden bir takım kurallar öğrenme ya da daha genel bir tanımla, yeni örneklerden genelleştirme yapmak için yararlanılabilecek bir sınıflandırıcı oluşturma süreci olarak tanımlanmaktadır. Denetimli

makine öğrenimini gerçek dünyadaki bir problem üzerinde uygulama aşamalarını Şekil 4.3'te açıklanmaktadır (Osisanwo ve diğ., 2017).



Şekil 4.3. Denetimli makine öğrenmesi veri işleme süreci (Osisanwo ve diğ., 2017)

Şekil 4.3'te, denetimli makine öğrenmesinin öğrenme ve işleme süreci görülmektedir. Bir problemin ele alınmasında ilk verilerin belirlenmesindeki başlangıç sürecinden örneğin sınıflandırılana kadar geçen aşamalar detaylı bir şekilde gösterilmektedir.

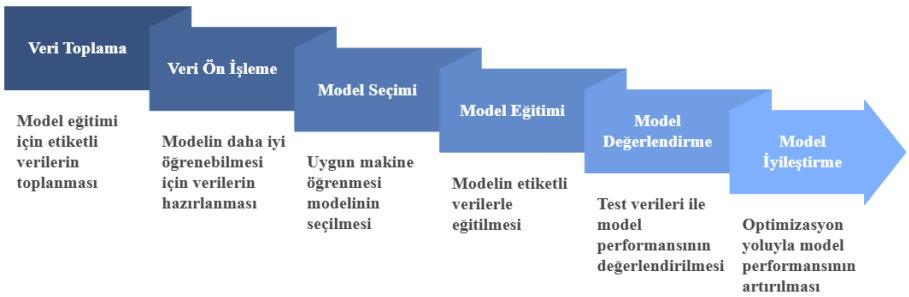
Aşağıdaki örnekler de olduğu gibi verileri sınıflandırma işlemi yapılabilmektedir.

- Bir e-postayı istenmeyen posta veya istenmeyen posta değil olarak sınıflandırma.
- Bir müşterinin kredi başvurusunun onaylanabilir veya reddedilebilir olarak sınıflandırılması.
- Bir görüntünün içinde bitki mi yoksa hayvan mı olduğunu belirleme.

4.2.1.2. Regresyon (Regression)

Regresyon, giriş verilerini kullanarak süreklilik gösteren değerleri tahmin eden temel bir denetimli öğrenme türüdür. Örneğin, sağlık hizmetlerinde, tıbbi maliyetleri tahmin etmek için regresyon uygulanabilmektedir. Girdi verileri ilaç fiyatları, gerekli tıbbi ekipman ve personel giderlerini içerirken, çıktı toplam tedavi maliyeti olmaktadır. Bu modellerin girdi ve çıktı verileriyle eğitilmesi yoluyla, yeni girdiler için toplam tedavi maliyeti tahminleri yapılabilmektedir (Kufel ve diğ., 2023).

Regresyon, girdiler ile devamlı bir çıktı değişkeni arasındaki ilişkiyi incelemeyi amaçlamaktadır. Çıktı değişkeni genellikle numerik bir değerdir (örneğin ev fiyatı, sıcaklık, gelir miktarı). Sonuç olarak gözetimli öğrenme süreci Şekil 4.4'te özetlenmektedir.



Şekil 4.4. Denetimli makine öğrenme süreci

Aşağıdaki örnekler de olduğu gibi verileri regresyon işlemi yapılabilmektedir.

- Bir konutun fiyatının tahmin edilmesi.
- Belirli bir günde hava sıcaklığının ne olacağının tahmin edilmesi.
- Bir kişinin yıllık eğitim ücretinin tahmin edilmesi.

4.2.2. Gözetimsiz öğrenme

Gözetimsiz makine öğrenmesi, etiketlenmemiş verileri kullanarak veri setlerindeki kalıpları ve yapıları keşfetmesine olanak tanıyan bir makine öğrenmesi türü olarak tanımlanmaktadır. Gözetimli öğrenmede algoritmalar, girdi verileri ile çıktı verileri arasındaki ilişkiyi öğrenmek için etiketli verileri kullanırken gözetimsiz öğrenmede etiketli veriye ihtiyaç bulunmamaktadır (Kufel ve diğ., 2023). Bunun yerine, verilerdeki benzerliklere, farklılıklara ve diğer örüntülere dayanarak verileri gruplandırmaktadır veya yapılandırmaktadır (Bishop, 2006).

Denetimsiz öğrenme, öğrenme sisteminin önceden var olan herhangi bir etiket veya özellik olmadan örüntüleri tespit etmesi gerektiğinde gerçekleşmektedir. Bu nedenle, eğitim verileri yalnızca ortak özellikleri paylaşan element grupları (kümeleme adıyla bilinmektedir) veya çok boyutlu bir alandan daha düşük bir alana yansıtılan veri görünümleri (boyut azaltma adıyla bilinmektedir) gibi önemli yapısal bilgileri bulmayı amaçlayan x parametrelerinden oluşmaktadır (Bishop 2006).

Kümeleme: Gözlemleri homojen bölgelere ayırmaktadır.

Boyut Azaltımı: Boyutsallığın azaltılması, denetimsiz öğrenmenin kullanabildiği bir diğer yöntemdir. Bir veri kümesinde bulunan özelliklerin sınırlandırılması veya farklı niteliklerin özelliklerini kaybetmeyecek biçimde birleştirilmesi işlemlerine "boyutsallık azaltma" adı verilmektedir (Tufail ve diğ., 2023).

4.2.3. Yarı gözetimli öğrenme

Yarı gözetimli öğrenme, hem etiketli hem de etiketsiz veriler üzerinde çalıştığından, daha önce ele alınan denetimli ve denetimsiz yöntemlerin bir melezlemesi olarak tanımlanabilmektedir (Sarker ve diğ., 2020).

Denetimli duygu sınıflandırma yaklaşımları, eğitim kümesi yeterince büyük olduğunda genellikle iyi performans gösterir. Ancak, test verisi dağılımının eğitim verisi dağılımından önemli ölçüde farklı olduğu zamanlarda, eğitilen sınıflandırıcıların

genellikle tatmin edici bir performans gösteremediği bilinmektedir. Bu sebeple, yeni bir etki alanıyla her karşılaşıldığında sınıflandırıcının yeniden eğitilmesi için veri toplanması ve etiketlenmesi gerekir.

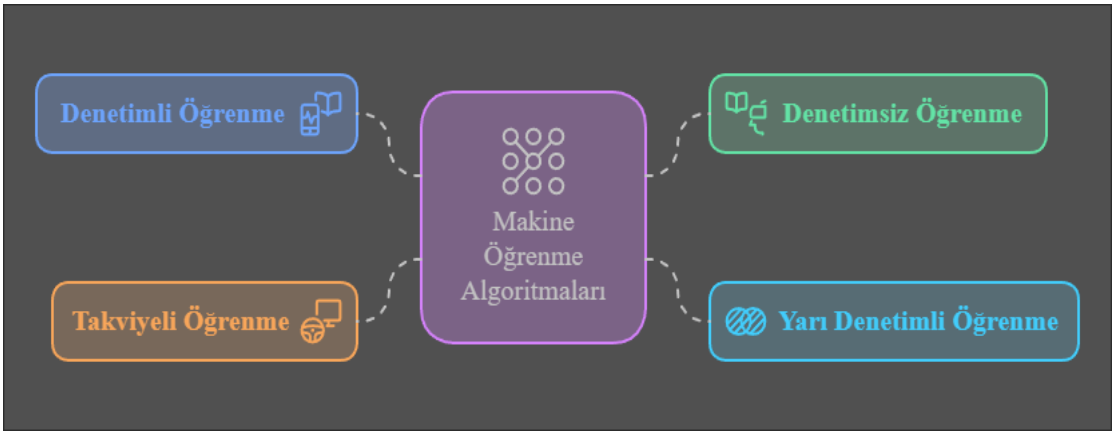
Duygu sınıflandırma görevinde denetimli yaklaşımların karşılaştığı etiketleme maliyeti sorununa karşılık sınıflandırıcı eğitimi için büyük miktarda etiketlenmemiş veriden ve az miktarda etiketlenmiş veriden yararlanan yarı denetimli yöntemlerin araştırılmasına yönelik ilgi giderek artmaktadır (Lin, 2011). Bu da maliyet sorunu olduğu durumlar için alternatif çözüm olarak yarı gözetimli öğrenmeye yönelimi düşündürmektedir.

4.2.4. Takviyeli Öğrenme

Takviyeli öğrenimi, öğrenmelerden hiçbirine benzemez, çünkü etiketlenmiş veya etiketlenmemiş veri kümeleri yoktur.

Bazı uygulamalarda, sistemin çıktısı bir dizi işlemdir. Böyle bir durumda, tek bir işlem önemli değildir; önemli olan, hedefe ulaşmak için doğru işlemler dizisi olan ilkedir. Herhangi bir ara durumda en iyi işlem diye bir şey yoktur; bir işlem iyi bir politikanın parçasıysa iyi olmaktadır. Böyle bir durumda, makine öğrenimi programı politikaların iyiliğini değerlendirebilmeli ve bir uygulama oluşturabilmek için geçmişteki doğru uygulama örneklerinden faydalanabilmelidir. Bu tür öğrenme yöntemlerine pekiştirmeli öğrenme algoritmaları denilmektedir (Alpaydın, 2010).

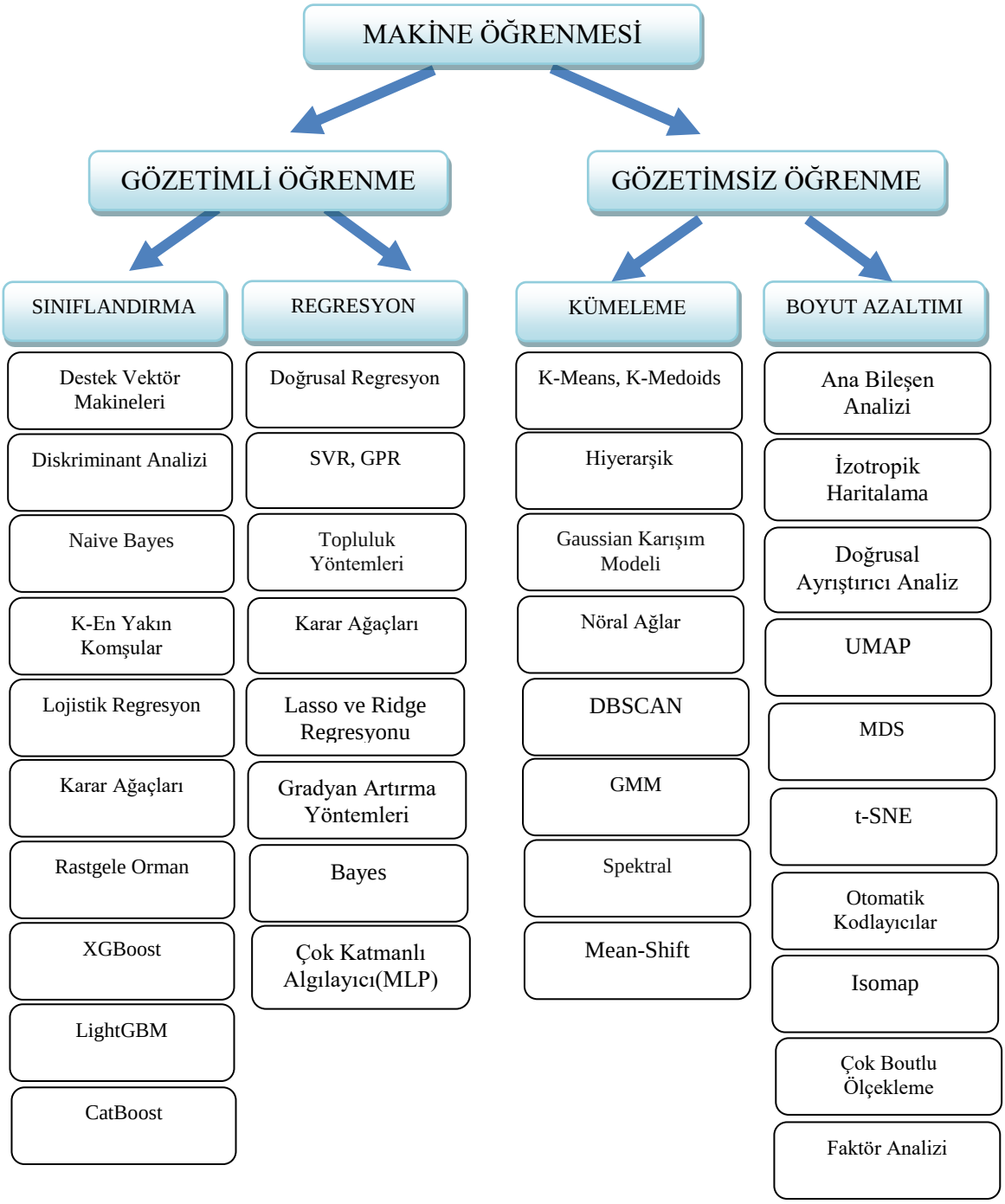
Şekil 4.5'te, algoritma türleri özet şekilde verilmektedir.



Şekil 4.5. Makine Öğrenme algoritmaları

4.3. Yaygın Olarak Kullanılan Makine Öğrenmesi Algoritmaları

Sınıflandırma ve regresyon gözetimli makine öğrenmesi algoritmaları ve kümeleme ve boyut azaltımı gözetimsiz algoritmalarının türleri örnek olarak Şekil 4.6' da, gözetimli ve gözetimsiz makine öğrenmesi algoritmaları detaylı olarak ele alınmaktadır (Sarker, 2021).



Şekil 4.6. Gözetimli ve gözetimsiz makine öğrenme algoritmaları

4.3.1. XGBoost Algoritması

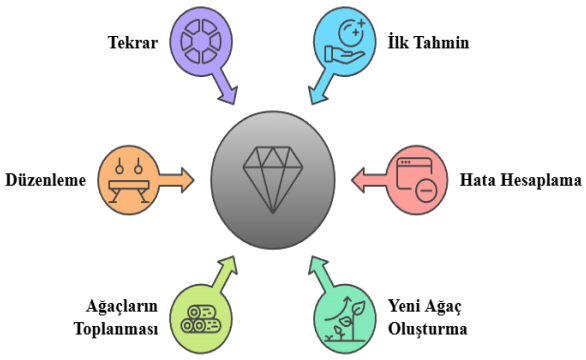
XGBoost, gradyan artırmaya dayanan etkili ve bir o kadar da güçlü bir sınıflandırma ve regresyon algoritmasıdır(Chen ve Guestrin, 2016).

Gradient boosting, genellikle karar ağaçları kullanarak her tekrarlamada oluşan hataları en düşük seviyeye indirmeye çalışmaktadır. XGBoost, özellikle çok büyük veri setlerinde büyük doğruluk ve hız sağlar, bu sebeple Kaggle gibi veri bilimi alanında düzenlenen yarışmalarda oldukça yaygın olarak kullanılmaktadır (Chen ve Guestrin, 2016).

4.2.1.1. XGBoost algoritmasının çalışma prensibi

XGBoost, zayıf öğrenicilerin kuvvetli bir öğreniciye dönüşümlü olarak birleştirildiği boosting algoritmaları ailesine girmektedir.

XGBoost, temel sınıflandırıcı olarak karar ağaçlarını kullanan Gradient Boosting sistemine dayanmaktadır. XGBoost, gradyan artırma gibi, bir kayıp fonksiyonunu en aza indirerek hedef fonksiyonunun toplam uzantısını oluşturmaktadır. Gradyan artırmaya benzer şekilde, XGBoost birbirini izleyen iterasyonlarda bir dizi ağaç oluşturmaktadır. Her iterasyonda algoritma, mevcut modelin hatalarını tahmin eden yeni bir ağaç oluşturur. Hatalar, gerçek değerler ile modelin tahminleri arasındaki fark demektir. Yeni ağaç, modelin doğruluğunu artırmak için önceki ağaçla birleştirilmektedir (Bentejac ve diğ., 2019). Şekil 4.7'de, XGBoost algoritmasının model eğitim aşamalarının özeti bulunmaktadır.



Şekil 4.7. XGBoost model eğitimi aşamaları

4.2.1.1. XGBoost Algoritmasının matematiksel temeli

XGBoost'un temel matematiksel yapısı gradyan yükseltme işlemlerinin optimizasyonuna dayanmaktadır. Bu, modelin doğruluğunu her iterasyonda kayıp fonksiyonunun bir türevini alarak geliştirmeyi amaçlamaktadır. XGBoost iki temel unsurdan oluşur; kayıp fonksiyonu ve düzenlilik terimi işlemi.

Kayıp Fonksiyonu: Modelin hatalarını genellikle karesel hatalar (regresyon) veya log kaybı (sınıflandırma) gibi fonksiyonlar kullanarak ölçmektedir.

Düzenlileştirme: Modelin karmaşık yapısını düşürmeye yarayan bir kavramdır. Bu, modelin fazla uyum sağlamasını engellemeye yardımcı olmaktadır (Chen ve Guestrin, 2016). Kayıp fonksiyonu formülleri Denklem 4.1 ve 4.2'deki gibidir (Bentejac ve diğ., 2019):

$$L(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \quad (4.1)$$

Burada:

$l(y_i, \hat{y}_i)$: Gerçek değer y_i ile tahmin $\hat{y}_i$ arasındaki kayıp fonksiyonu.

$\Omega(f_k)$: Ağaç f_k' 'nin regülerleşme terimi.

$$\Omega(f) = \gamma T + 1/2\lambda\|w\|^2 \quad (4.2)$$

Burada:

γ, yaprak sayısı için ceza terimidir.

λ, yaprak ağırlıkları için ceza terimidir.

T, ağacın yaprak sayısıdır.

w, yaprak ağırlıklarıdır.

4.2.1.3. XGBoost algoritmasının avantajları

XGBoost algoritması için yüksek verimlilik, aşırı uyum önleme, esneklik gibi özellikler (Chen ve Guestrin, 2016) önemli bir artı sağlamaktadır.

Bu özellikler aşağıdaki gibidir:

- Yüksek Verimlilik: Özellikle geniş veri kümelerinde güçlü hız ve doğruluk sağlamaktadır.

- Aşırı Uyumu Önleme: Düzenli hale getirme tekniklerini uygulayarak aşırı uyumu engellemektedir.
- Esneklik: Hem sınıflandırma hem de regresyon problemleri için elverişlidir.
- Eksik Verileri İşleme: XGBoost, eksik verileri etkili bir şekilde işleyebilen yerleşik bir mekanizmaya sahiptir (Bentejac ve diğ., 2019).

XGBoost'un birçok parametresi vardır ve bu parametreler algoritmanın performansını önemli ölçüde etkilemektedir.

En önemli parametrelerden bazıları şunlardır (Bentejac ve diğ., 2019):

learning_rate: Öğrenme oranı

max_depth: Ağaçların maksimum derinliği

min_child_weight: Bir yaprağın bölünmesi için gereken minimum örnek ağırlığı.

gamma: Bölünme için gereken minimum kayıp azaltma miktarı

subsample: Her ağaç için kullanılan verilerin alt örnekleme oranı

colsample_bytree: Her ağaç için kullanılan özelliklerin alt örnekleme oranı

colsample_bylevel: Her seviye için özellik alt kümesi oranı.

4.3.2. Karar Ağaçları (Decision Trees)

Denetimli öğrenme algoritmaları sınıfına ait olan Karar Ağacı (DT) algoritması daha çok sınıflandırma problemlerinin çözümlenmesinde kullanılmakla beraber, her iki durumda da gerek sınıflandırma gerekse regresyon çalışmalarında tercih edilebilmektedir.

Karar ağaçlarında dalların yapılarını gösteren iç düğümler, algoritma tarafından verilen kararı temsil eden veri kümesi ve her bir sonucu temsil eden yaprak düğümlerden oluşmaktadır. Birincisi karar vermek için kullanılan ve çeşitli dallara sahip olan karar düğümü; ikincisi ise karar düğümlerinin çıktısı olan ve başka dalları olmayan yaprak düğümü olmak üzere iki düğüm bulunmaktadır (Bansal ve diğ., 2022). Karar ağaçları algoritması aşağıdaki gibi elementlere sahiptir (Bansal ve diğ., 2022) :

- **Kök Düğüm:** Karar Ağacının ilk kısmı, tüm veri setinin bölünmeye başlayarak çeşitli olası kümelere bölündüğü düğüm olarak adlandırılmaktadır.
- **Yaprak Düğüm:** Ağaçların daha fazla bölünmesinin artık mümkün olmadığı son nihai sonuç noktasıdır.

- **Bölme:** Ana düğümün, belirtilen sınırlamalar çerçevesinde alt düğümlere ayrılması işlemini kapsamaktadır.
- **Alt Ağaç:** Bir ana düğümün bir alt ağaca veya dala ayrılmasına neden olan işlemdir.
- **Budama:** Bu, en iyi sonuçları elde etmek için Karar Ağacının gereksiz dallarının ortadan kaldırılması işlemini içermektedir.
- **Alt ve Ana düğüm:** Ana düğüm aynı zamanda ana düğüm olarak da adlandırılırken, geri kalan düğümler basitçe alt düğüm olarak adlandırılmaktadır.

4.3.2.1. Karar ağacı algoritmasının matematiksel temeli

Karar ağacı algoritması, bölme işlemi için çeşitli ölçütler kullanılmaktadır. En yaygın ölçütler arasında bilgi kazanımı ve Gini indeksi bulunmaktadır. Bilgi kazanımı, bir bölmenin veri kümesindeki belirsizliği ne kadar azalttığını ölçer. Gini indeksi, bir karar ağacı algoritmasının oluşturulması aşamasında kullanılan safsızlığı veya saflığı ölçmektedir (Bansal ve diğ., 2022). Bilgi kazanımı ve gini indexi formülleri Denklem 4.3 ve Denklem 4.4'deki gibi tanımlanmaktadır(Sarker, 2021). Burada $p(x_i)$, her sınıfın olasılığını ifade etmektedir.

$$H(x) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i) \quad (4.3)$$

$$\text{Gini}(E) = 1 - \sum_{i=1}^{c} p_i^2 \quad (4.4)$$

4.3.2.1. Karar ağacı öğrenme algoritmalarının avantajları

Karar ağacı algoritmasının avantajları aşağıdaki gibidir:

- Anlaşılma kolaylığı: Karar ağaçları, verilerin nasıl sınıflandırıldığına dair basit ve anlaşılır bir temsil sağlamaktadır.
- Veri ön işleme gerekliliği: Karar ağaçları, verileri normalleştirme veya boyutlandırma gibi ön hazırlama işlemleri gerektirmemektedir.
- Hem sayısal hem de kategorik verileri işleme: Karar ağaçları, çeşitli veri türlerini kullanabilen esnek bir yaklaşım sunmaktadır (Bansal ve diğ., 2022).

- Hız: Karar ağaçları çoğunlukla diğer algoritmalara göre daha hızlı eğitilmektedir (Lin ve diğ., 2020).

4.3.3. Naive Bayes Algoritması

Naive Bayes, verilen veri kümesinin frekans ve değer birleşimini hesaplayarak bir takım ihtimal hesaplamaları yapan basit bir olasılıksal sınıflayıcıyı ifade etmektedir (Peling ve diğ., 2017).

4.3.3.1. Naive bayes algoritmasının matematiksel temeli

Naive Bayes formülü Denklem 4.5'deki gibidir (Wongkar ve Angdresey, 2019):

$$P(Cj \mid x) = \frac{p(x|Cj)P(Cj)}{p(x)} = \frac{p(x|Cj)P(Cj)}{\sum_k p(x|Ck)P(Ck)} \quad (4.5)$$

$p(x \mid Cj)$: Sınıf j'den bir örneğin x olma olasılığı

$P(Cj)$: Sınıf j'nin ilk olasılığı

$p(x)$: Herhangi bir örneğin x olma olasılığı

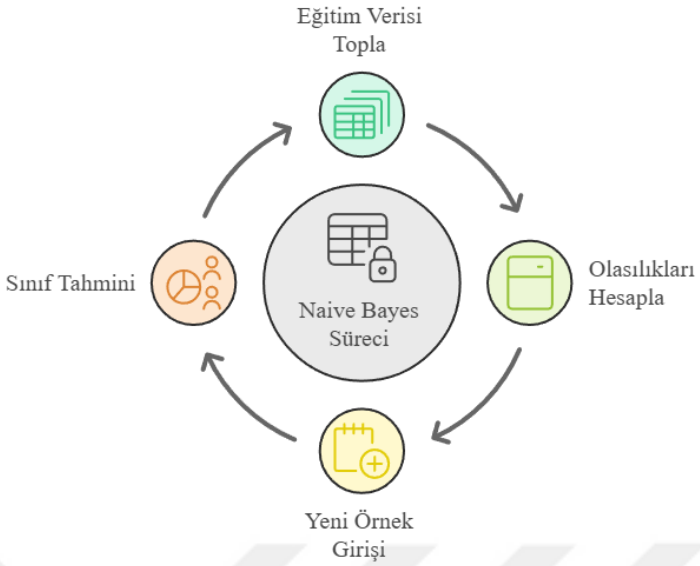
$P(Cj \mid x)$: x olan bir örneğin sınıf j'den olma olasılığı (son olasılık)

4.3.3.2. Naive bayes algoritmasının avantajları

Naive Bayes algoritmasının güçlü yönleri olarak vurgulanmaktadır(Peling ve diğ., 2017):

- **Basitlik:** Anlaşılması ve uygulanması kolaydır.
- **Hız:** Eğitim ve sınıflandırma aşamaları hızlı gerçekleşmektedir.
- **Yüksek Boyutlu Verilerde Etkisi:** Çok fazla sayıda özelliğe sahip veri kümelerinde iyi performans gösterir.
- **Eksik Verilerle Başa Çıkma Durumu:** Eksik verileri işleyebilmektedir.
- **Metin Sınıflandırmada Başarısı:** Spam Filtreleme, Duygu Analizi ve Konu Sınıflandırması gibi metin tabanlı uygulamalarda yaygın olarak kullanılmaktadır.

Şekil 4.8'de, Naive Bayes algoritmasının sınıflandırma döngüsü bulunmaktadır.



Şekil 4.8. Naive bayes sınıflandırma döngüsü

4.3.5. Lojistik Regresyon (Logistic Regression)

Lojistik regresyon, bir veya daha fazla sayıda bağımsız değişkenin ikili bir bağımlı değişken üzerindeki ilişkisini tahmin etmek için kullanılmaktadır (Schober ve Vetter, 2021).

4.3.5.1. Lojistik regresyon türleri

Lojistik regresyon türleri aşağıdaki gibidir:

- İkili (Binary) Lojistik Regresyon: İki sınıf arasında sınıflandırma yapılır.
- Çok sınıflı (Multinomial) Lojistik Regresyon: Birden fazla sınıfın olduğu durumlar için uygulanır (Boateng ve Abaye, 2019).
- Ordinal Lojistik Regresyon: Sınıfların sıralı olduğu durumlarda kullanılmaktadır.

4.3.5.2. Lojistik regresyon algoritmasının matematiksel temeli

Lojistik Regresyonun matematiksel modeli doğrusal regresyonun bir türevi gibi görünmesine rağmen, doğrusal tahmin çıktıları doğrudan sınıflandırma için

kullanılmamaktadır. Onun yerine, doğrusal modelin sonucu bir sigmoid fonksiyonu (veya lojistik fonksiyonu) ile dönüştürülmektedir.

Sigmoid Fonksiyonu (Lojistik Fonksiyon): Lojistik regresyon, genellikle sınıflandırma problemleri için kullanılan istatistiksel bir yöntemdir (Sarker, 2021). Bu model, girdi parametrelerinin doğrusal bir birleşimi alınır ve bu birleşim sigmoid adı verilen doğrusal olmayan bir fonksiyon aracılığıyla bir olasılığa dönüştürmektedir. Bu olasılık, giriş değişkeninin belirli bir sınıfa ait olma ihtimalini temsil etmektedir.

Sigmoid fonksiyonu, çıktı değerlerini 0 ile 1 arasında sınırlayan bir fonksiyondur ve Denklem 4.6'daki şekilde tanımlanmaktadır (Bishop, 2006):

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{4.6}$$

Burada z, doğrusal modelin çıktısıdır ve Denklem 4.7'deki şekilde ifade edilmektedir:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \tag{4.7}$$

Bu doğrusal birleşim, bağımsız değişenlerin (girdi verilerinin) katsayılarıyla (parametreleriyle) çarpılmasını ve bir sabit terim β_0 ′ yu içerir.

Sigmoid fonksiyonu, modelin tahmin ettiği olasılığı Denklem 4.8'deki gibi vermektedir:

$$P(y = 1 \mid X) = \sigma(z) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\cdots+\beta_n x_n)}} \tag{4.8}$$

Burada:

$P(y = 1 \mid X), y = 1$ olma olasılığıdır (örneğin, "evet" veya "spam" gibi).

$\beta_0, \beta_1, \ldots, \beta_n$ modelin parametreleridir ve öğrenilmesi gereken katsayılardır.

$X = (x_1, x_2, \ldots, x_n)$, özellikler veya bağımsız değişkenlerdir.

Lojistik regresyon, genellikle maksimum likelihood (ML) yöntemi ile parametrelerini öğrenmektedir. Maksimum likelihood, gözlemlerin elde edilme olasılığını maximum düzeye çıkarmayı amaçlamaktadır.

Likelihood Fonksiyonu, daha iyi istatistiksel özelliklere sahip olduğu için β'yı tahmin etmek için en çok olasılık yöntemi olarak tercih edilmektedir (Joshi ve Dhakal, 2021).

Lojistik regresyon için likelihood fonksiyonu, sınıf etiketleri $y_i \in$ (y1,y2,..) ve modelin tahminleri $\hat{y}_i$ kullanılarak Denklem 4.9'daki gibi tanımlanmaktadır:

$$\mathcal{L}(\beta) = \sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \quad (4.9)$$

Burada $\hat{y}_i = \sigma(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_n x_{in})$ modelin tahmin ettiği olasılıktır. Log-likelihood fonksiyonu, modelin doğruluğunu değerlendirmek için kullanılan ana fonksiyondur ve parametreler $(\beta_0, \beta_1, \ldots, \beta_n)$ bu fonksiyonu en üst düzeye ulaştıracak şekilde optimize edilmektedir.

Kayıp Fonksiyonu: Genellikle log kaybı veya ikili çapraz entropi olarak adlandırılan lojistik regresyonun maliyet fonksiyonu (kayıp fonksiyonu) Denklem 4.10'daki gibi ifade edilmektedir:

$$J(\beta) = -\frac{1}{N}\sum^{N} [y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \quad (4.10)$$

Bu fonksiyon, tahmin edilen olasılık ile gerçek etiketler arasındaki farkı ölçmektedir.

Gradyan İnişi (Gradient Descent): Lojistik regresyon, model parametrelerini öğrenmek için gradient descent (gradyan inişi) algoritmasını kullanılmaktadır. Bu algoritma, cost fonksiyonunun türevini alarak, parametrelerin değerlerini optimize etmektedir.

Her bir parametre için gradyan Denklem 4.11'deki gibi hesaplanmaktadır:

$$\frac{\partial J(\beta)}{\partial \beta_j} = \frac{1}{N}\sum_{i=1}^{N} (\hat{y}_i - y_i)x_{ij} \quad (4.11)$$

Lojistik regresyonla ilgili bazı önemli noktalar vurgulanmaktadır (Bishop, 2006):

- Genelleştirilmiş Doğrusal Modeller: Lojistik regresyon, genelleştirilmiş doğrusal modeller grubunun bir parçasıdır. Genelleştirilmiş doğrusal modellerde, sonuç değişkeninin olası değeri, girdi değişkenlerinin

doğrusal bir birleşiminin doğrusal olmayan bir fonksiyonu ile modellenmektedir.

- Olasılıksal Ayırıcı Modeller: Bu tür modeller, direkt girdi bilgisi verilen çıktı değişkeninin şartlı olasılık dağılımlarını modellemeyi amaçlamaktadır.
- Maksimum Olabilirlik Tahmini: Lojistik regresyon modelinin parametreleri en yüksek benzerlik (maximum likelihood) tahmin metodu kullanılarak belirlenmektedir. Bu yöntemde model parametreleri, gözlemlenen verilerin olasılığını maksimuma çıkaracak şekilde ayarlanmaktadır.
- Çapraz Entropi Hata Fonksiyonu: Lojistik regresyon modelinin eğitim sürecinde çapraz entropi hata fonksiyonu kullanılmaktadır. Bu hata fonksiyonu, model tarafından tahmin edilen olasılıklar ile gerçek etiketler arasındaki farklılığı ölçmektedir.
- Yinelemeli Ağırlıklı En Küçük Kareler Algoritması (IRLS): Bir Lojistik Regresyon modelinin değişkenlerini bulmak için en yaygın kullanılan optimizasyon yöntemlerinden biri IRLS'dir. Bu algoritma, çapraz entropi hata fonksiyonunu minimize etmek amacıyla ağırlıklandırılmış en az kareler problemini iteratif olarak çözer.
- Çok Sınıflı Lojistik Regresyon: Lojistik regresyon modeli, softmax fonksiyonu ile çoklu sınıf sınıflandırma problemlerine uyarlanabilmektedir. Softmax fonksiyonu, girdi değişkenlerine ait doğrusal kombinasyonları, her bir sınıf için bir ihtimal değeri üreten bir olasılık dağılımına dönüştürmektedir.

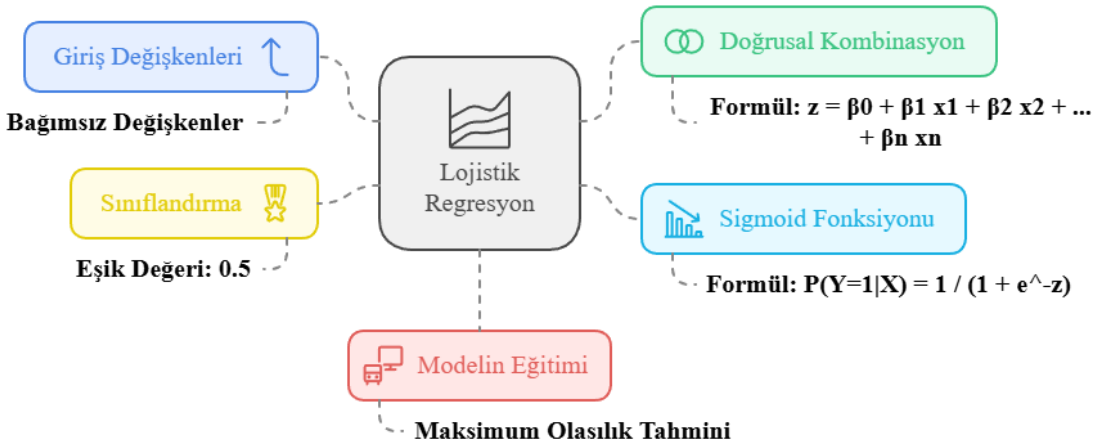
4.3.5.2. Lojistik regresyonun avantajları

Basit ve Yorumlanabilir: Lojistik regresyon sonuçların yorumlanmasını kolaylaştırmaktadır.

Hızlı Eğitim: Diğer kompleks algoritmalardan daha hızlı olup küçük veri setlerinde iyi sonuç vermektedir.

Çoklu Bağımsız Değişkenlerle Çalışabilirliği: Lojistik regresyon birden fazla birbirinden bağımsız değişkenlerle de çalışabilmektedir (Schober ve Vetter, 2021).

Şekil 4.9'da, lojistik regresyon özeti bulunmaktadır.



Şekil 4.9. Lojistik regresyon özeti

4.3.6. Destek Vektör Makineleri (Support Vector Machines - SVM)

Destek Vektör Makineleri (DVM), makine öğrenmesinde yaygın olarak kullanılan güçlü bir sınıflandırma ve regresyon algoritması olarak kullanılmaktadır (Kok ve diğ., 2021). DVM, özelikle sınıflandırma sorunlarında gösterdiği üstün performansla bilinmektedir.

Destek Vektör Makineleri, veri noktalarını birbirinden farklı sınıflara en iyi biçimde ayıracak hiper düzlemi bulmaya çalışmaktadır. Bir hiper düzlem, n boyutlu bir uzayda verileri ayıran bir yüzeydir. DVM, iki sınıf arasındaki farkı(marjı) maksimize ederek bu hiper düzlemi oluşturmaktadır. Fark (marj), sınıflandırma yüzeyi ile en yakın sınıf veri noktaları arasındaki uzaklıktır. Bu en yakındaki veri noktaları destek vektörleri olarak adlandırılır ve hiper düzlemin pozisyonunun belirlenmesinde kritik bir role sahiptir.

DVM doğrusal olarak ayrılabilen veriler için çalışabilmektedir, ancak aynı zamanda çekirdek fonksiyonları yardımıyla doğrusal olmayan verileri de sınıflandırabilmektedir (Leong ve diğ., 2020). Çekirdek işlevleri verileri daha büyük boyutlu bir uzaya dönüştürüp doğrusal olarak ayrıştırılabilir hale getirmektedir. En yaygın kernel fonksiyonları (Leong ve diğ., 2020):

- Doğrusal Kernel: Veriler doğrusal olarak ayrılabilir olduklarında kullanılmaktadır.
- Polinom Kernel: Daha kompleks veri kümeleri için uygundur.

- RBF (Radyal Taban Fonksiyonu) Çekirdeği: En çok kullanılan kernel işlevlerinden biridir ve genellikle daha kompleks, doğrusal olmayan sınıflandırma problemlerinde tercih edilmektedir.

4.3.6.1. DVM algoritmasının avantajları

DVM' nin çok sayıda avantajı bulunmaktadır (Kok ve diğ., 2021). Bunlardan bazıları paragraflar halinde aşağıda belirtilmektedir.

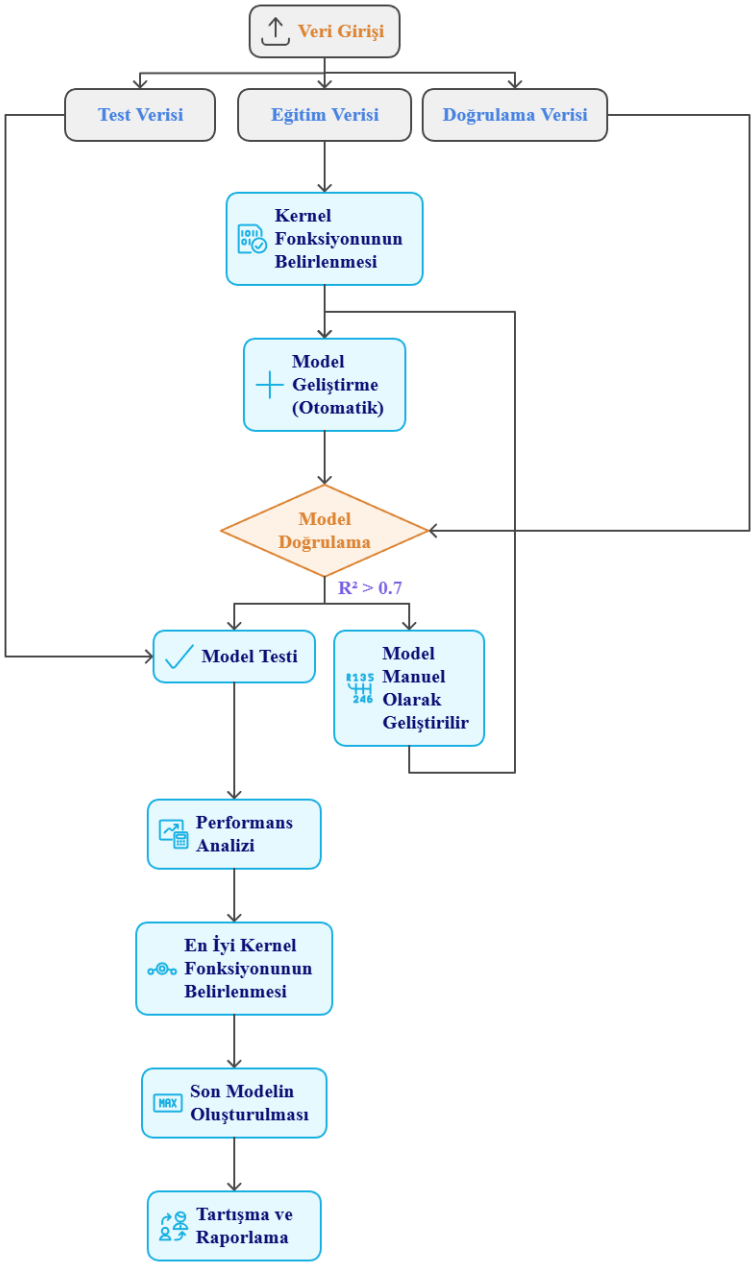
- Yüksek Boyutlu Verilerde Verimlilik: DVM yüksek boyutlu verilerde dahi iyi performans göstermektedir. Bu, birçok özelliğe sahip verilerle çalışırken önemli bir avantaj sağlamaktadır.
- Karmaşık Karar Sınırlarını Modelleme: Kernel fonksiyonları ile DVM doğrusal olmayan karar sonuçlarını modelleyebilmektedir. Bu da kompleks veri setlerinde daha doğru sınıflandırmaların yapılmasını sağlamaktadır.
- Genelleme Yeteneği: DVM, büyük marjlar üreterek genelleştirebilme yeteneğini artırmakta ve yeni verileri daha doğru şekilde sınıflandırabilmektedir.
- Küçük Örnek Büyüklüklerinde Etkililik: DVM, diğer bazı makine öğrenimi yöntemlerine göre daha küçük örnek boyutlarında da etkin olabilmektedir.
- Gürültülü Verilere Karşı Sağlamlık: DVM nispeten gürültülü verilere karşı da sağlam yapıdadır.

4.3.6.2. DVM algoritmasının dezavantajları

DVM' nin bazı dezavantajları da vardır (Kok ve diğ., 2021):

- Hiperparametre Ayarlama: DVM'nin başarısı, çekirdek fonksiyonunun türü ve C (ceza parametresi) türü gibi hiper parametrelerde değişikliklere yol açabilmektedir. Bu aşırı parametrelerin en uygun olan değerlerini bulmak için deneme yanılma gerekebilmektedir.
- Büyük Veri Kümelerinde Hesaplama Maliyeti: DVM yönteminin büyük veri kümeleri üzerinde eğitilmesi işlem maliyeti açısından yüksek olabilmektedir.

Şekil 4.10'da, SVM ' nin çalışma akış şekli görülmektedir (Leong ve diğ., 2020).



Şekil 4.10. SVM akış şeması

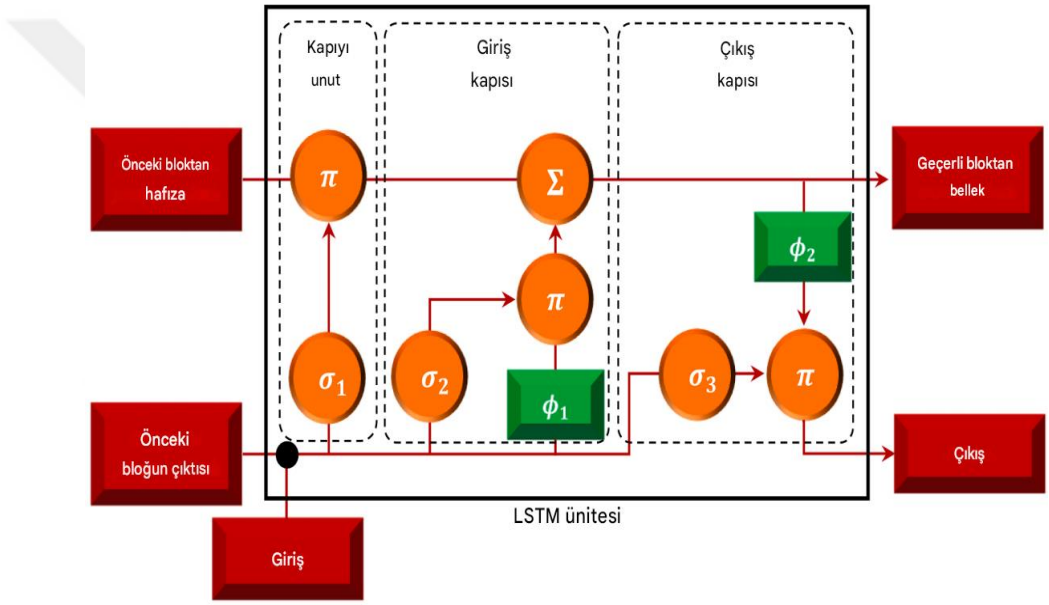
4.3.7. Long–Short-Term Memory (LSTM) Algoritmaları

LSTM, zaman serisi, ses veya metin gibi birbirini izleyen verileri işleyebilmek için geliştirilmiş bir yapay sinir ağı yöntemidir. Belirli bir zaman adımındaki çıktının

önceki zaman adımlarından gelen verilere bağlı olarak değiştiği uzun vadeli bağımlılıklara yönelik verileri işlemede özellikle kullanılabilmektedir. LSTM ağları, bellek hücreleri, giriş kapıları, çıkış kapıları ve unutma kapıları kullanmak suretiyle bu bilgileri daha uzun bir zaman dilimi süresince hatırlayabilmektedir (Gülmez, 2023).

LSTM, her zaman katmanındaki bellek hücrelerinin durumunu güncelleyerek çalışmaktadır. Güncelleme süreci giriş kapısı, unutma kapısı ve çıkış kapısı ile kontrol edilmektedir (Wu ve diğ., 2023).

Şekil 4.11'de, LSTM algoritmasının tekrarlama modülü detaylı olarak gösterilmektedir.

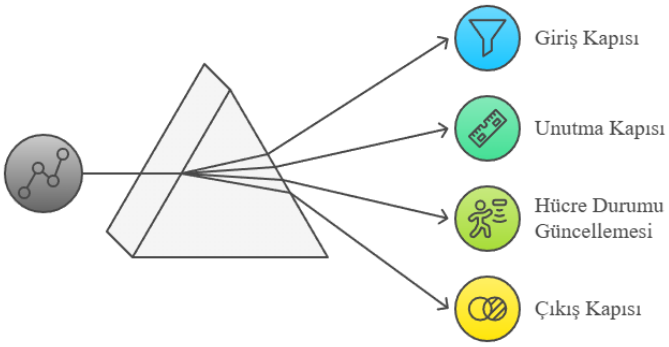


Şekil 4.11. LSTM tekrarlama modülü (Mahmoodzadeh ve diğ., 2022)

4.3.7.1. LSTM algoritmasının matematiksel temeli

Unutma kapısı, mevcut zaman ve bir önceki zamandaki çıktılar kullanılarak, hücre durumunun çıktısı sigmoid fonksiyonu aracılığıyla sigmoid fonksiyonunun çıktısı ile çarpılmaktadır. Sigmoid fonksiyonu 0 değerini üretirse, bu bilginin bir kısmının unutulması gerekmektedir, yoksa bilginin bir kısmının birleşik durumda iletilmesine devam edilmektedir (Wu ve diğ., 2023).

Şekil 4.12'de, LSTM mimarisini gösterilmektedir.



Şekil 4.12. LSTM' nin mimarisi

$$z_t = \delta\left(E_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{4.12}$$

Denklem 4.12'de, z_t mevcut çıkış değeri, E_f mevcut çıkışın ağırlığı, b_f mevcut çıkışın yanlılığı ve h_{t-1} bir önceki katmanın çıkış değeri olarak verilmektedir.

Giriş Kapısı, yeni bilginin ne kadarlık kısmının bellek hücresine gireceğini kontrol etmektedir.

$$j_t = \delta(E_i \cdot [h_{t-1}] + b_i) \tag{4.13}$$

$$\tilde{B}_t = \tan h(E_B \cdot [h_{t-1}, x_t] + b_B) \tag{4.14}$$

Denklem 4.13 ve 4.14'te, j_t ve $\tilde{B}_t$, iki parçayı temsil eden giriş kapısındaki mevcut çıkış değerlerini temsil etmektedir. Denklem 4.14'te hücre durumu hesaplaması gösterilmektedir. Hücre Durumu, Bellek hücresindeki bilgileri depolamaktadır (Gülmez, 2023).

$t-1$ zamanındaki, h_{t-1} ve x_t girişleri başka bir doğrusal dönüşüm ve sigmoid yani giriş kapısı ve jt çıkışı ile etkinleştirilmektedir. Aynı zamanda, h_{t-1} ve x_t başka bir doğrusal dönüşümle yani tan h etkinleştirildikten sonra, bir ara sonuç elde etmek için j_t ile çarpılır. Sonuç olarak B̃t' yi elde etmek için önceki adımın ara sonucuna eklenmektedir. Çıkış kapısı, bellek hücresindeki bilginin ne kadarının çıktı olarak kullanılacağını kontrol etmektedir.

$$p_t = \delta\left(W_p[h_{t-1}, x_t] + b_p\right) \tag{4.15}$$

$$h_t = p_t * \tan h(B_t) \tag{4.16}$$

Denklem 4.15'te, $t-1$ zamanındaki h_{t-1} ve x_t girişleri başka bir doğrusal dönüşüm + sigmoid ile etkinleştirildikten sonra (buna çıkış kapısı denir), p_t, p_t çıkışı h_t elde etmek için tanh yoluyla Bt ile çarpılır.

Denklem 4.16'da, Gizli durum; LSTM biriminin çıktısı, tahminler yapmak için kullanılır veya bilgileri bir sonraki LSTM birimine aktarmakta olduğunu gösterir (Gülmez, 2023).

4.3.8. Doğrusal Diskriminant Analizi (Linear Discriminant Anlysis) Algoritmaları

Doğrusal Diskriminant Analizi (LDA) hem özellik çıkarma hem de boyut azaltma için kullanılan iyi bilinen yöntemlerden biri olup, verileri daha düşük boyutlu bir vektör uzayına yansıtmaktadır (Balakrishnama ve Ganapathiraju, 1998).

LDA'nın işleyebilmesi için veri setindeki her bir sınıfın ortalamalarının ve varyansının hesaplanması gerekmektedir. Bu ortalamalar ve varyanslar, verileri daha küçük boyutlu bir uzaya yansıtmakta ve sınıflar arasındaki farkı en üst seviyeye taşıyacak doğrusal bir ayrıştırıcı oluşturmaktadır (Ye ve diğ., 2004).

4.3.8.1. LDA'nın Kullanımı ve Kademeleri

LDA' nın kullanımı ve kademeleri aşağıdaki gibidir:

- Ortalama Vektörlerin Hesaplanması: Her sınıf için ortalama vektörler hesaplanır.
- Dağılım Matrisinin Oluşturulması: Her bir sınıfın dağılım matrisleri hesaplanarak toplam dağılım matrisi elde edilmektedir.
- Doğrusal Diskriminant Bulma: En büyük ayrıştırmayı gerçekleştirecek diskriminant düzlemi (vektörü) belirlenmektedir.
- Veri Dönüşümü: Veriler bu ayrıştırıcıya göre dönüştürülmekte ve sınıflandırılmaktadır.

4.3.8.2. LDA algoritmasının avantajları

LDA algoritmasının, basitlik, verimlilik, yüksek boyutlu verilerle çalışabilme ve çok sınıflı problemleri destekleme gibi avantajları bulunmaktadır.

LDA' nın avantajları aşağıdaki gibidir:

- Basitlik ve Verimlilik: LDA hesaplama açısından oldukça basit ve verimli bir yöntemdir (Ye ve diğ., 2004).
- Yüksek Boyutlu Verilerle Başa Çıkabilme Yeteneği: LDA, yüksek boyutlardaki veriler ile etkili bir biçimde başa çıkabilmektedir, bu da onu görüntü işleme ve yüz tanıma gibi uygulamalar için elverişli kılmaktadır (Ye ve diğ., 2004).
- Çok Sınıflı Problemleri Destekleme: LDA, hem ikili sınıflandırma problemlerini hem de çok sınıflı sorunları çözebilmektedir (Balakrishnama ve Ganapathiraju, 1998).

4.3.8.3. LDA algoritmasınınnın dezavantajları

LDA algoritmasının, avantajlarının yanı sıra dezavantajları da bulunmaktadır. LDN' nın dezavantajları aşağıdaki gibidir:

- Tekillik Problemi: Dağılım matrislerinin tekil olması halinde LDA başarısız olabilmektedir. Bu sorun PCA + LDA gibi yöntemlerle ele alınabilmektedir (Ye ve diğ., 2004).
- Doğrusal Olarak Ayrıştırılamayan Verilerle Başa Çıkma Zorluğu: LDA, lineer olarak ayrıştırılamayan veriler ile mücadele edebilmektedir. Bu durumda, doğrusal olmayan boyut azaltma yöntemlerinin uygulanması gerekebilmektedir.

4.3.9. En Yakın Komşular(K-Nearest Neighbors)

En Yakın Komşular (KNN) algoritması hem sınıflandırma hem de regresyon problemlerinde kullanılan temel bir makine öğrenmesi tekniğidir(Mladenova ve Valova, 2023). Denetimli öğrenme yöntemlerinden biridir ve veriyi sınıflandırmak veya tahmin etmek için yeni bir veri noktasının k en yakın komşularını kullanmaktadır. K-NN algoritması genelde etiketli verilere dayanan sınıf veya değer tahmini için tercih edilmektedir ve basitliği ile bilinmektedir.

k-NN algoritması üç ana aşamadan oluşmaktadır (Mladenova ve Valova, 2023):

Mesafe Hesaplama: Bu aşamada bilinmeyene ait veri noktasının eğitim veri setindeki her bir veri noktasına olan uzaklığı hesaplanmaktadır. En çok kullanılan

uzaklık ölçüsü Öklid uzaklığı olmakla birlikte Manhattan uzaklığı, Minkowski uzaklığı ve Hamming uzaklığı gibi diğer uzaklık ölçüleri de kullanılabilmektedir.

En Yakın Komşuların Seçimi: Kaynağı bilinmeyen veri noktasına en yakın 'k' komşu seçilmektedir. Burada 'k' kullanıcı tanımlı bir aşırı parametredir.

Sınıflandırma veya Regresyon: Sınıflandırma sorunlarında, bilinmeyen veri noktasının sınıfına en yakın 'k' komşusunun en çok rastlanan sınıf değeri (çoğunluk oylaması) atanmaktadır. Regresyon problemlerinde, bilinmeyen veri noktasının değeri en yakın 'k' komşusunun değerlerinin ortalamasından hesaplanmaktadır. Öklidyen ve Manhattan mesafeleri Denklem 4.17 ve Denklem 4.18'de verildiği gibidir (Strauss ve Von Maltitz, 2017):

Öklidyen Mesafesi: İki veri noktası arasındaki düz çizgi mesafesi.

$$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{4.17}$$

Manhattan Mesafesi: Yatay ve dikey mesafelerin toplamı.

$$d = \sum_{i=1}^{n} |x_i - y_i| \tag{4.18}$$

Minkowski Mesafesi: Minkowski mesafe ölçüm değeri Denklem 4.19' daki gibi tanımlanmaktadır (Mailagaha Kumbure ve Luukka, 2022) :

$$d_{Md}(X_i, X_j) = \left(\sum_{t=1}^{m} \left|x_t^i - x_t^j\right|^p\right)^{1/p} \text{ for } p \geq 1 \tag{4.19}$$

4.3.9.1. k-NN algoritmasının avantajları

k-NN algoritmasının, basitlik, hızlı sınıflandırma gibi avantajları bulunmaktadır.

k-NN algoritmasının avantajları aşağıdaki gibidir:

- Basitlik ve anlaşılabilirlik: k-NN algoritması oldukça kolay bir şekilde anlaşılabilmektedir ve kullanılabilmektedir.
- Parametrik olmayan yapı: Verilerin doğrusal veya doğrusal olmayan olmasına aldırmaksızın çalışmaktadır.

- Yeni verileri hemen sınıflandırabilir: Model eğitim istemediği için yeni veri geldiğinde hızlı bir şekilde sınıflandırılabilmektedir.

4.3.9.2. k-NN algoritmasının dezavantajları

k-NN algoritmasının avantajlarının yanı sıra dezavantajları da bulunmaktadır. k-NN algoritmasının dezavantajları aşağıdaki gibidir:

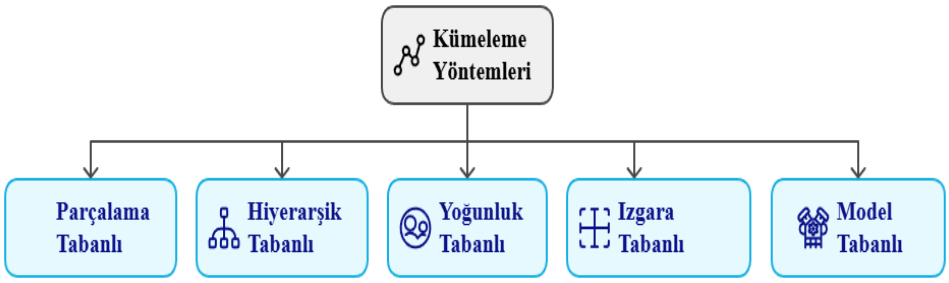
- Yavaş çalışma: KNN, bütün veri noktalarına uzaklığı hesaplamak gerektiğinden büyük veri kümelerinde yavaş çalışabilmektedir.
- Boyut arttıkça performans düşer: Çok boyutlu veri setlerinde performans kaybı yaşanabilmektedir.
- Özellik ölçeklendirme: Değişik birimlerde özellikler içindeki mesafeleri doğru bir şekilde hesaplamak için verileri ölçeklendirmek gerekmektedir.

4.3.10. Kümeleme Algoritmaları

Kümeleme algoritmaları, veri noktaları arasındaki benzerliği veya mesafeyi ölçmek için çeşitli ölçütler kullanarak, etiketlenmemiş veri kümeleri üzerinde çalışmaktadırlar (Rodriguez ve diğ., 2019).

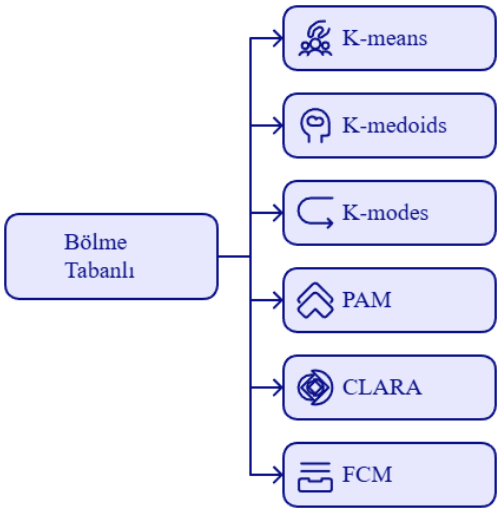
Özetle kümeleme algoritmaları, veri noktalarını benzerliklerine göre gruplar halinde düzenlemek için kullanılan bir yöntem olarak tanımlanır. Bu gruplara küme adı verilir ve her küme birbirine benzer veri noktalarından oluşturmaktadır.

Kümeleme algoritmalarına yönelik özet, Şekil 4.13'te gösterilmektedir (Sajana ve diğ., 2016).



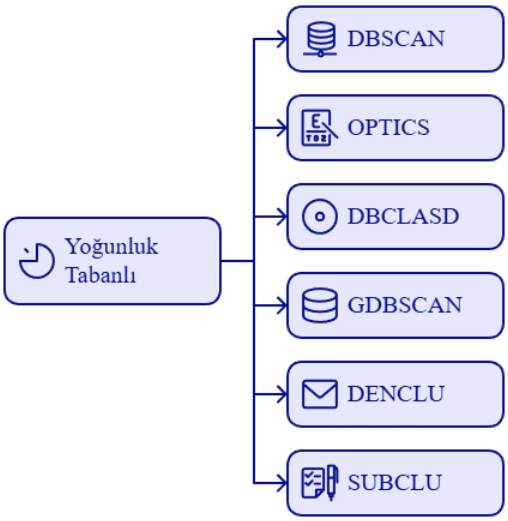
Şekil 4.13. Kümeleme algoritmaları

Şekil 4.14, Kümeleme yöntemlerinin bölme tabanlı algoritmalarını göstermektedir.

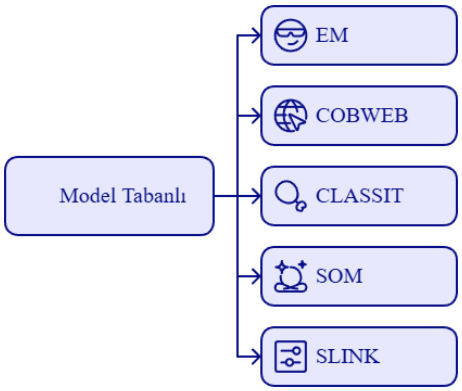


Şekil 4.14. Bölme tabanlı kümeleme algoritmaları

Şekil 4.15'te, kümeleme yöntemlerinin yoğunluk tabanlı algoritmaları ve Şekil 4.16'da, kümeleme yöntemlerinin model tabanlı algoritmalarını gösterilmektedir.

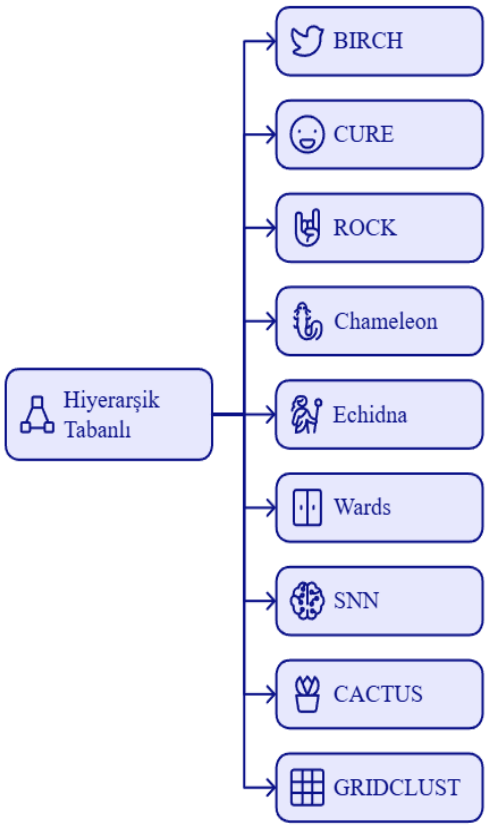


Şekil 4.15. Yoğunluk tabanlı kümeleme algoritmaları



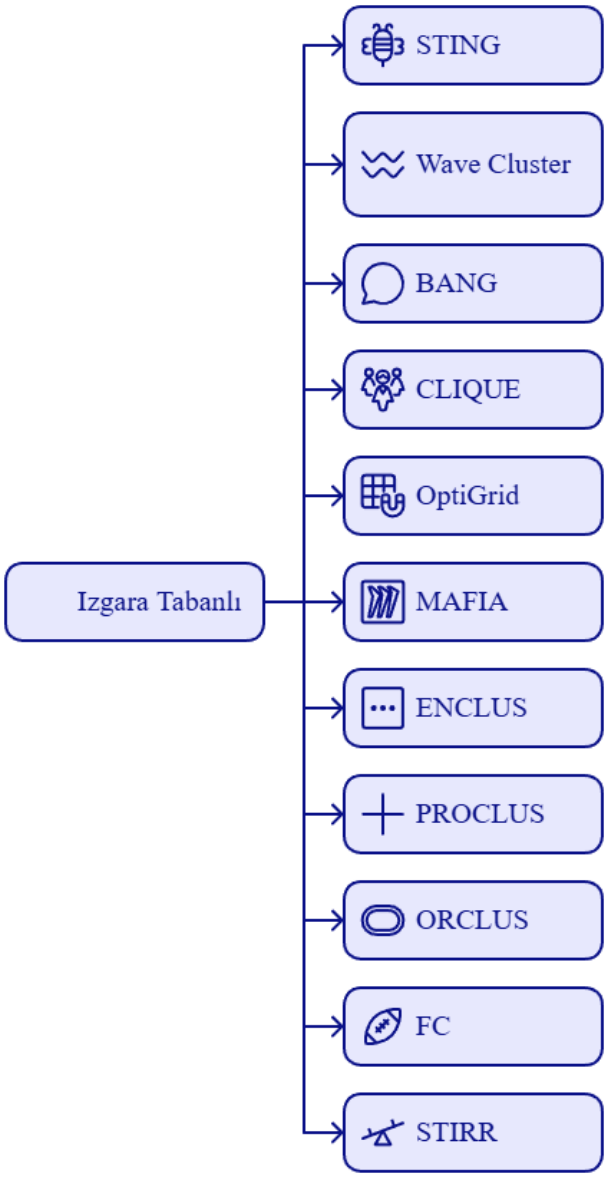
Şekil 4.16. Model tabanlı kümeleme algoritmaları

Şekil 4.17'de, kümeleme yöntemlerinin hiyerarşik algoritmaları gösterilmektedir.



Şekil 4.17. Hiyerarşik tabanlı kümeleme algoritmaları

Şekil 4.18'de, kümeleme yöntemlerinin ızgara tabanlı algoritmaları gösterilmektedir.



Şekil 4.18. Izgara tabanlı kümeleme algoritmaları

4.3.11. Doğrusal Regresyon(Linear Regrassion) Algoritmaları

Girdi değişkenlerinin D boyutundaki x vektörünün değeri göz önüne alındığı zaman bir veya daha fazla sayıda sürekli hedef değişkenlerinin t değerini tahmin edebilmeyi amaçlayan modele Lineer Regresyon denir (Bishop, 2006). Bağımlı

değişkeni bulmak için oluşturulan model içerisinde girdi olarak tek bağımsız parametre kullanılması durumunda tekli regresyon, çok sayıda bağımsız parametreler kullanılması sonucunda ise çoklu regresyon analizinden söz edilmektedir (Şevgin, 2023).

4.3.11.1. Lineer regresyon algoritmasının türleri ve matematiksel temeli

Basit Doğrusal Regresyon: Tek bir bağımsız değişken ile bağımlı değişken arasındaki ilişkiyi modellemek için kullanılır. Denklem 4.20'de, formülü gösterilmektedir.

$$Y = \beta_0 + \beta_1 X + \epsilon \tag{4.20}$$

Çoklu Doğrusal Regresyon: Birden fazla bağımsız değişkenin bağımlı değişken üzerindeki etkisini incelemek için kullanılır. Denklem 4.21'de, formülü gösterilmektedir.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \epsilon \tag{4.21}$$

Denklem 4.20 ve Denklem 4.21'de:

Y : Bağımlı değişken (hedef)

X : Bağımsız değişkenler (özellikler)

β_0 : Y kesişim noktası (intercept)

$\beta_0, \beta_1, \beta_2, \ldots, \beta_n$: Katsayılar (coefficients)

ϵ: Hata terimi (residuals)

Doğrusal regresyonda kullanılan en küçük karesel hata kayıp fonksiyonu Denklem 4.22'de gösterildiği gibi ifade edilmektedir.

$$LSE = \frac{1}{2}\left(f_\theta\left(x^{(i)} - y^{(i)}\right)^2\right. \tag{4.22}$$

Burada $f_\theta\left(x^{(i)}\right)$, belirli bir nokta için tahmini değeri göstermektedir ve $y^{(i)}$, gerçek değeri temsil etmektedir. Maliyet fonksiyonu, veri setindeki tüm veri noktalarının tüm LSE' lerinin toplamı olup, Denklem 4.23'deki gibi ifade edilmektedir; burada n, veri noktası sayısını temsil etmektedir.

$$J = \frac{1}{2n}\sum_{i=1}^{n} \left(f_\theta\left(x^{(i)} - y^{(i)}\right)^2\right. \quad (4.23)$$

4.3.11.2. Lineer regresyonun avantajları

Doğrusal regresyon kullanmanın en önemli faydası, eğitilmesinin çok kısa sürede gerçekleşmesi ve uygulanması çok kolay bir model olmasıdır. Bununla birlikte, doğrusal regresyonun eğitim süresi diğer modellerden önemli ölçüde daha düşük olmaktadır (Tufail ve diğ., 2023).

Lineer regresyon algoritmasının avantajları aşağıdaki gibidir:

- Basitlik ve Kolay Uygulanabilirlik: Doğrusal regresyonun kavranması ve uygulanabilirliği son derece kolaydır çünkü oldukça anlaşılır bir matematiksel yapıya dayanmaktadır.
- Hızlı Hesaplama: Eğitim ve öngörü hızlı olduğundan büyük veri setleriyle çalışırken bir avantaj sağlamaktadır.
- Yorumlanabilirlik: Katsayılar, birbirinden bağımsız olarak değişkenlerin hedef değişken üzerindeki etkisini kolaylıkla yorumlamayı sağlamaktadır.
- Düşük Aşırı Uyum Riski: Model yapısı oldukça yalın olduğu için, özellikle az sayıda özelliğe sahip olunduğunda aşırı uyum riski genellikle düşük olmaktadır.

4.3.11.3. Lineer regresyonun dezavantajları

DVM' nin avantajlarının yanı sıra bazı dezavantajları da bulunmaktadır.

- Doğrusallık Varsayımı: Doğrusal regresyon bağımlı ve bağımsız değişkenlerin arasında doğrusal bir ilişkiye sahip olduğunu varsaymaktadır. Bu varsayım karşılanmadığında model performansı düşmektedir.
- Aykırı Değerlere Duyarlılık: Aykırı değerlerin katsayı tahminlerini ve bu sayede modelin hassasiyetini negatif yönde etkileyebilmektedir.
- Çoklu Doğrusallık Sorunu: Bağımsız değişkenler arasındaki ilişkinin yüksek olması halinde katsayılar istikrarsız hale gelebilmekte ve modelin güvenilirliği düşebilmektedir.

- Genelleme Problemleri: Karmaşık ilişkileri ve doğrusal olmayan modelleri yansıtamamaktadır. Bu gibi hallerde doğrusal model olmayan yöntemler daha iyi sonuç vermektedir.

4.3.12. Rasgele Orman (Random Forest) Algoritması

Rastgele orman, birçok sınıflandırıcı oluşturan ve bunların sonuçlarını birleştiren "toplu öğrenme" metotlarından biri olarak tanımlanmaktadır. Rastgele ormanlar, sınıflandırma ve regresyon ağaçlarının torbalama tekniğine ek bir rastgelelik katmanı eklenmesiyle tanımlanmıştır. Rastgele ormanlar, standart ağaçlardaki her bir düğümü tüm değişkenler arasında en iyi bölünmeyi kullanarak bölmenin tersine, her bir düğümü o düğümde rastgele seçilmiş bir alt kümedeki en iyi tahmin ediciyi kullanarak bölmek suretiyle oluşturulmaktadır (Liaw ve diğ., 2022).

Rastgele orman, çoklu regresyon ağaçlarının bir topluluk versiyonudur. Avantajı, yüksek açıklanabilirliktir, ancak tahmin edilen sonuçlar eğitim örnekleriyle sınırlı kalmaktadır. Regresyon ağacının prensibi, belirli bir değişkenin bir göstergesini kullanarak ana grubu alt gruplara bölmek olup sınıflandırma, aşağıdaki Denklem 4.24'de gösterildiği gibi, her grubun karesel hatalar toplamının ortalamasını en küçük yapmak üzerine temellendirilmiştir(Chen, 2023).

$$\frac{1}{n_1}\sum_{i=1}^{n_1}\left(y_i - \overline{y_{(1:n_1)}}\right) + \frac{1}{n_2-n_1}\sum_{j=n_1+1}^{n_2}\left(y_j - \overline{y_{(n_1+1:n_2)}}\right) \rightarrow \min \qquad (4.24)$$

4.4. NLP (Doğal Dil İşleme) Nedir?

Doğal Dil İşleme (NLP), bilgisayarların kullanıcı tarafından kullanılan dili anlayabilmesini, yorumlayabilmesini ve üretebilmesini mümkün kılan bir yapay zekâ alanı olarak tanımlanmaktadır. NLP, kişilerin hayatlarında kullanmakta oldukları dili (Türkçe, İngilizce, Fransızca vb.) işleme tabi tutarak bilgisayarların metin ve konuşmaları anlayıp reaksiyon göstermelerini sağlamaktadır (Chopra ve diğ., 2013).

NLP (Doğal Dil İşleme), insan dilindeki farklı boyutları (gramer, anlam, bağlam) inceleyerek dilin karmaşık yapısını anlamaya ve dil temelli görevleri gerçekleştirmeye çalışmaktadır. Bu süreç, metin veya ses verilerini anlayıp özetlemek, metin tabanlı sınıflandırmalar yapabilmek, dildeki duygu durumlarını analiz etmek,

öngörülerde bulunmak ve çeviri yapabilmek gibi birçok farklı kullanım alanını kapsamaktadır.

4.4.1. NLP Seviyeleri

NLP' nin temel amacı insanlar ve bilgisayarlar arasında etkili bir iletişim kurmaktır. Bu amaçla NLP, dilin farklı seviyelerini analiz etmekte ve işlemektedir. Bu seviyeler aşağıdaki gibidir (Liddy 2001):

- **Fonoloji:** Konuşma seslerinin anlamlandırılmasını ele almaktadır. Ses dosyalarını analiz ederek bir dil modeline göre yorumlamaktadır.
- **Morfoloji:** Kelimelerin en küçük anlamlı birimleri olan biçim öbeklerinin yapısını incelemektedir.
- **Sözcüksel:** Tek tek sözcüklerin ne anlama geldiğini yorumlamaktadır. Kelime türlerini (isim, fiil, sıfat, vb.) tespit ederek anlamlarını ifade edebilmektedir.
- **Sözdizimsel:** Cümlelerin dil bilgisel özelliklerini inceler ve kelimeler üzerindeki yapısal bağlılıkları ortaya çıkarmada kullanmaktadır.
- **Anlamsal:** Sözcüklerin, cümlelerin ve metinlerin anlamını inceler. Kelimelerin değişik anlamlarını birbirinden ayırıp cümleler arasında anlam ilişkilerini belirler.
- **Söylem:** Cümlelerden daha uzun metin birimlerinin niteliklerini analiz etmektedir. Tüm metnin anlamının yapılandırılması için tümceler arsındaki ilişkileri analiz edebilecektir.
- **Edimbilim:** Bazı durumlarda dilin nasıl anlamlı kullanıldığını incelemektedir.

  Metnin içeriğinden öte bağlam kullanarak anlam çıkarmaya çalışmaktadır.

4.4.1. NLP Görevleri

NLP’ nin başlıca görevleri aşağıda açıklanmaktadır.

Dil Modelleri: Dil modelleri, metinlerde kelimelerin ve tümcelerin nasıl sıralanacağını öğrenen istatistiksel modellerdendir. N-gram dil modelleri, önceki n-1

kelimeye bağlı olarak bir kelimenin bir metinde geçme olasılığını hesaplar. Bu modeller metin oluşturma, yazım denetimi ve makine çevirisi gibi görevlerde kullanılır (Jurafsky, 2018).

Metin Sınıflandırma: Metin verisini belirli kategorilere ayırmaktadır.

Duygu Analizi: Kullanıcı yorumlarındaki veya sosyal medya yayınlarındaki duygu durumunu belirlemektedir.

Duygu analizi, bir yazının arkasındaki hissi veya tutumu tanımlama sürecidir. Kaynaklarda duygu analizi için kullanılan sözcük yerleştirmeleri ve sözlükler gibi yöntemler açıklanmaktadır.

Kelime yerleştirmeleri kelimeleri çok yönlü vektörler olarak temsil etmektedir ve birbirine benzer anlamlara sahip sözcükler vektör uzayında birbirine yakın yerleştirilmektedir.

Adlandırılmış Varlık Tanıma (NER): Metindeki kişiler, yerler, kuruluşlar vb. gibi isimlendirilmiş varlıkları tanır.

Makine Çevirisi: Metnin bir dilden başka bir dile çevrilmesidir. Makine çevirisi, bir dili otomatik olarak diğerine çevirme sürecidir.

Kaynaklarınız, kodlayıcı-kod çözücü mimarisi gibi makine çevirilerinde uygulanan sinir ağı modellerini açıklamaktadır. Bu modeller hedef dildeki kaynak dili bir vektör gösterimine kodlar ve ardından hedef dilde bir çeviri oluşturur.

Metin Özetleme: Büyük metinleri anlamlı bir şekilde özetleyerek çözümler.

Soru ve Cevap Sistemleri: Kullanıcı tarafından doğal dilde sorulan sorulara uygun yanıtlar sağlar.

Sinir Ağları: Sinir ağları, NLP'de gittikçe daha yaygın hale gelen çok gelişmiş makine öğrenme modellerindendir. Kaynaklar, tekrarlayan sinir ağları (RNN'ler), uzun kısa süreli bellek (LSTM) ağları ve transformatöfr ağları gibi farklı sinir ağı mimarilerini tanımlamaktadır.

Bu yöntemler, metin verilerindeki karmaşık kalıpları öğrenmekte ve çeşitli NLP uygulamalarında en gelişmiş performansı elde etmekte kullanılır.

4.5. Performans Ölçümleri

Modellerin performansını değerlendirmek için birden fazla ölçüm kriteri bulunmaktadır (De Myttenaere ve diğ., 2016).

4.5.1. MAPE (Mean Absolute Percentage Error - Ortalama Mutlak Yüzde Hatası)

MAPE, bir modelin tahmin edilen değerler ile gerçek değerleri arasındaki yüzdesel hata oranının ortalamasını ölçen bir performans metriğidir.

Denklem 4.25'de, MPAE formülü gösterilmektedir.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n} \left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \quad (4.25)$$

y_i: Gerçek değer

$\hat{y}_i$: Tahmin edilen değer

n: Toplam veri sayısı

4.5.2. MAE (Mean Absolute Error - Ortalama Mutlak Hata)

MAE, tahmin edilen değerler $\hat{y}_i$ ile gerçek değerler y_i arasındaki farkın mutlak değerlerinin ortalamasıdır.

Denklem 4.26'da, MAE formülü gösterilmektedir.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (4.26)$$

4.5.3. MSE (Mean Squared Error - Ortalama Kare Hata)

MSE, tahmin edilen değerler $\hat{y}_i$ ile gerçek değerler y_i arasındaki farkların karelerinin ortalamasıdır. Denklem 4.27'de, MSE formülü gösterilmektedir.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad (4.27)$$

4.5.4. R^2 (R-Squared - Determinasyon Katsayısı)

R^2, bağımlı değişkenin (örneğin hedef değişken y) toplam sapmasının ne kadarının bağımsız değişkenler (örneğin modelin girdileri) tarafından açıklandığını göstermektedir. Denklem 4.28'de, R^2 formülü gösterilmektedir.

$$R^2 = 1 - \frac{\text{Model Hatasi (RSS)}}{\text{Toplam Varyans (TSS)}} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (4.28)$$

MAE, MSE, ve MAPE düşük olduğunda modelin performansı daha iyidir.

R^2 yüksek olduğunda model veri ile uyumlu görünür.

Çizelge 4.1'de, performans hedefleri nasıl değerlendirmeli konusu açıklanmıştır.

Çizelge 4.1. Performans hedefleri

Metrik	Performans Hedefi
MAE	**Düşük**
MSE	**Düşük**
R^2	**Yüksek** (1'e yakın)
MAPE	**Düşük**

Bu metrikler birlikte değerlendirilmelidir, çünkü her biri farklı bir perspektiften modelin doğruluğunu ölçer. Bu şekilde tüm metrikler değerlendirildiğinde Doğrusal Regresyon en iyi başarıya ulaşmıştır.

5. MATERYAL VE YÖNTEM

Bu bölümde tez çalışmasında materyal olarak kullanılan bitcoin fiyat verileri, haber verileri ve yazılım araçları hakkında bilgi verildikten sonra yöntem olarak kullanılan veri toplama, veri ön işleme, model geliştirmede tercih edilen modellerden ve modellerin değerlendirilmesinden bahsedilmiştir.

5.1. Materyal

Meteryal olarak kullanılan veri setleri tweeter, haber verileri ve bitcoin fiyatlarının içeriklerinden oluşmaktadır.

5.1.1. Veri Setleri

5.1.1.1. Bitcoin fiyat verileri

Bitcoin'in tarihsel fiyat bilgileri, Yahoo Finance Api aracılığı ile geçmişe yönelik eğitim seti için veriler toplanmıştır. Bu veriler, günlük açılış, kapanış fiyatları, en düşük ve en yüksek değeri, hacmi gibi önemli değişkenleri içerir. Veri seti, eğitim ve test amacıyla 22/01/2021 - 09/09/2021 dönem aralığı ve 22/01/2021- 09/01/2023 dönem aralığı için oluşturulmuştur. Ayrıca 01/01/2014 - 01/01/2024 tarih aralığı için de büyük veri seti ile de tahmin işlemlerini denemek için veri seti oluşturulmuştur. Çizelge 5.1' de, Bitcoin fiyat bilgileri gibi özelliklerin olduğu veri seti örneği bulunmaktadır.

Çizelge 5.1. Bitcoin fiyat bilgileri (USD)

Tarih	Kapanış	En Yüksek	En Düşük	Açılış	Hacim
2021-01-22	33005.76171875	33811.8515625	28953.373046875	30817.625	77207272511
2021-01-23	32067.642578125	33360.9765625	31493.16015625	32985.7578125	48354737975
2021-01-24	32289.37890625	32944.0078125	31106.685546875	32064.376953125	48643830599
2021-01-25	32366.392578125	34802.7421875	32087.787109375	32285.798828125	59897054838
2021-01-26	32569.849609375	32794.55078125	31030.265625	32358.61328125	60255421470

5.1.1.2. Haber verileri

Bitcoin ile ilgili haber makaleleri, ve sosyal medya platformlarından (örneğin, Twitter) elde edilmiştir. Tweeter apilerinden geçmişe yönelik veri çekme işlemleri ücretli olduğundan, eğitim için ‘https://www.kaggle.com/datasets’ web sitesi adresinden Bitcoin ile ilgili tweet içeren ‘bitcoin-tweets-2021.csv’ ve Bitcoin ile ilgili haberleri içeren ’bitcoin_news_sentiments_2024.csv’ isimli dosyalar temin edilmiştir. ‘bitcoin-tweets-2021.csv’ isimli dosyada yaklaşık 16.000.000 veri bulunmaktadır. Bu veri setinden yaklaşık 10.000.000 milyon satır okundu ve ön işlem sonrası 9.554.304 adete indirilmiştir. Ek olarak aralığı artırmak için ‘bitcoin-tweets-2021.csv’, ‘bitcoin-tweets-2022.csv’ ve ‘Bitcoin_tweets.csv’ isimli dosyalardan sırasıyla 15630735, 7277322, 4693145 adet veri okunmuştur. Ayrıca test için ’bitcoin_news_sentiments_2024.csv’ isimli dosyadan da 1070 adet veri alınmıştır. Farklı veri setleri kaynaklarından alınan metinler NLP teknikleri kullanılarak analiz edilmiş ve duygu durumları “pozitif”, “negatif” ve “nötr” olarak belirlenmiştir.

Metinlerin sınıflandırılması sonrası, günlük olarak maksimum içeriğe sahip sınıflandırma türü, gün bazında segmentasyon girdisini oluşturacak şekilde set edildi (Örneğin; 01/01/2022 tarihi için sınıflandırma işlemleri yapılmış veri setine göre segmentasyon verisi ‘Positive’ olarak işaretlenmiştir). Günlük bazda belirlenen segmentasyon verisi, fiyat veri seti ile birleştirileceğinden eğitim ve test amacıyla, aynı dönem aralığı olan 22/01/2021 - 09/09/2021 ve 22/01/2021 - 09/01/2023 aralıkları için oluşturulmuştur. Ayrıca uzun dönem eğitim için segmentasyon verisi olmadan 01/01/2014 ve 01/01/2024 tarihleri arasındaki bitcoin fiyatları çekilmiştir.

Çizelge 5.2’ de, günlük segmentasyon örnek veri seti gösterilmiştir. Çizelge 5.3, tweetlerin içeriği örnek niteliğinde gösterilmektedir. Çizelge 5.4’te, Bitcoin tweet metinlerini içeren veri seti örneği ve Çizelge 5.5’te, haber veri setinin içeriği örnek olarak verilmiştir.

Çizelge 5.2. Günlük segmentasyon örnek veri seti

Tarih	Segmentasyon
2021-01-22	Negative(Negatif)
2021-01-23	Positive(Pozitif)
2021-01-24	Neatral(Nötr)

Çizelge 5.3. Tweet içerikleri

Tweet ile İlgili Genel Bilgiler	
Id	Tweet'in benzersiz kimliği.
Text	Tweet'in içeriği.
created_at	Tweet'in oluşturulma zamanı.
Lang	Tweet'in dili (örneğin, 'en' veya 'tr').
Source	Tweet'in gönderildiği platform (örneğin, 'Twitter for iPhone').
possibly_sensitive	Tweet'in hassas içerik içerip içermediğini belirten bayrak (isteğe bağlı).
Tweet Etkileşim Verileri	
retweet_count	Retweet sayısı.
favorite_count	Beğeni (like) sayısı.
reply_count	Yanıt sayısı
quote_count	Alıntı tweet sayısı.
Kullanıcı Bilgileri	
User	Tweet'i atan kullanıcıya ait bilgiler
user.id	Kullanıcının benzersiz kimliği.
user.screen_name	Kullanıcının kullanıcı adı (örneğin, "@username").
user.name	Kullanıcının tam adı.
user.verified	Kullanıcının doğrulanmış olup olmadığı.
user.followers_count	Kullanıcının takipçi sayısı.
user.friends_count	Kullanıcının takip ettiği kişi sayısı.
user.location	Kullanıcının lokasyon bilgisi (eğer paylaşılmışsa).
Medya ve Ek İçerikler	
Entities	Tweet'teki medya, hashtagler, ve bağlantılar:
Hashtags	Kullanılan hashtaglerin listesi.
url	Tweet içinde geçen URL'ler.

Çizelge 5.4. Örnek tweet veri seti

Datetime	Date	Text
2021-01-01 23:59:35+00:00	2021-01-01	5 Reasons to invest in Bitcoin in 2021 https://t.co/G30BXV9Mcg",5 reasons invest bitcoin 2021
2021-01-01 23:53:03+00:00	2021-01-01	Bitcoin Cash: The Road to Mass Adoption (New BCH Documentary) https://t.co/NmW5fd5uY2 #bitcoincash #bch
2021-01-01 23:52:56+00:00	2021-01-01	#Fundstrat #Analyst #Tom #Lee #Reveals #Bitcoin #Price #Prediction for 2021 - https://t.co/4GKgzoj1u8 #Fundstrat #analyst and executive #Tom #Lee thinks Bitcoinâ€™s incredible bull run is just be...

Çizelge 5.5. Örnek haber veri seti

Date(Tarih)	Text(Tweet içeriği)
2024-05-08	Bitcoin is struggling to break through the 26-day Exponential Moving Average (EMA). The overall market sentiment remains bearish. The trading volume is neutral with a slight downward trend. Liquidation of long positions has slowed the momentum in perpetual markets. Bitcoin enthusiasts and traders are waiting to see if Bitcoin can break through $65,000 resistance level.
2024-05-07	Shiba Inu (SHIB) and Bitcoin (BTC) are showing signs of a price rebound. Lucie has urged the cryptocurrency community to closely monitor the prices of the two digital currencies. Shiba inu has increased by 8% over a seven-day period and Bitcoin is still above $64,000.
2024-05-06	Bitcoin has crossed the 65,000 USDT benchmark and is now trading at $65,232.9 USDT, with a 2.28% increase in 24 hours. Binance Market Data shows Bitcoin is trading on May 06, 2024,10:07 (UTC+0).

5.1.2. Yazılım ve Araçlar

Python Programlama Dili: Veri analizi, makine öğrenmesi ve doğal dil işleme uygulamaları için Python kullanılmıştır. Python kütüphaneleri, ‘pandas’ dosya işlemleri, ‘xgboost’, ‘sklearn’, ‘keras’, ‘keras.models’, ‘keras.layers’, ‘sklearn.linear_model’, ‘sklearn.preprocessing’, ‘keras.callbacks’ makine öğrenmesi algoritmalarının veri setini eğitip modeli sonuçlandırması, ‘yfinance’ bitcoin günlük fiyat ve diğer özelliklerine erişmek için, ‘textblob’, ‘nltk’ tweet metinlerinin duygu durumunu belirlemek için, ‘seaborn’ ve ‘matplotlib’ da görselleştirme için kullanılmıştır.

5.2. Yöntem

Yöntem olarak veri toplama, veri ön işleme, model geliştirme ve modelin değerlendirilmesi aşamaları bulunmaktadır.

5.2.1. Veri Toplama

Bitcoin fiyat verileri, kripto para borsalarının API' leri aracılığıyla otomatik olarak toplanmaktadır. Haber verileri toplanırken, tweeter gibi sosyal medya API' leri ile geliştirici hesabı açarak sınırlı sayıda güncel haberler toplanmıştır, eğitim ve test için ‘https://www.kaggle.com/datasets’ web sitesi kullanılmıştır.

5.2.2. Veri Ön İşleme

Veri setleri modellerde kullanılmadan önce ön işlemlerden geçirilir.

Temizleme: Toplanan verilerdeki eksik ve hatalı veriler tespit edilerek temizlenmiştir. Özellikle, boş hücreler ve tutarsız veriler giderilmiştir.

Duygu Analizi: Haber verileri, duygu analizi yöntemleriyle (VADER, TextBlob) işlenmiş ve her haberin duygu durumu pozitif, negatif ve nötr olacak şekilde belirlenmiştir.

Dönüştürme: Fiyat verileri zaman serisi formatında düzenlenmiştir ve haber metinleri ise doğal dil işleme teknikleri kullanılarak işlenmiştir. Fiyat ve haber metinleri hibrit bir şekilde giriş verisi oluşturacak şekilde hazır hale getirlmiştir.

5.2.3. Model Geliştirme

Makine Öğrenmesi algoritmaları kullanılarak farklı senaryolar oluşturulup modeller geliştirilmiştir.

5.2.3.1. Çalışmada kullanılan modeller

Tez kapsamında aşağıdaki algoritmalar kullanılmıştır.

- Regresyon Modelleri: XGBoost, Doğrusal regresyon.
- Sınıflandırma Modelleri: Rastgele orman.
- Zaman Serisi Modelleri: LSTM (Uzun Kısa Süreli Bellek) modelleri.

Her model, farklı özellik setleri kullanılarak eğitilmiştir ve model parametreleri, çapraz doğrulama yöntemleri ile optimize edilmiştir.

5.2.4. Model Değerlendirme

Modellerin performansı, MAE, MSE, RMSE, R-kare ve MAPE fonksiyonu gibi metriklerle değerlendirilmiştir. Hatalara göre başarı oranları tespit edilmiştir.

Elde edilen tahmin sonuçları, gerçek fiyatlarla karşılaştırılarak analiz edilmiştir.

6. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu bölümde yazmış olduğum uygulamaya dair tüm işlemler detaylı olarak açıklanmaktadır. Uygulamada kullandığımız ön işleme yöntemleri, veri setinde yapılan analiz ve sınıflandırma açıklanmaktadır. Ek olarak uygulamada performansta maksimum başarı oranı elde edilmek için farklı senaryolar hazırlanmıştır.

6.1. Uygulama

Bitcoin fiyatlarının tahmini için, verilerin analizi ve sınıflandırılmasıyla ilgili python programlama dili kullanılarak uygulama geliştirilmiştir. Uygulamanın amacı yaygın olarak kullanılan twitter aracında paylaşılan tweetleri inceleyerek sınıflandırma yapıp, haber sitelerindeki günlük belirlediğimiz fiyat parametreleri ile giriş olarak alınarak Bitcoin için, bir sonraki gün açılış fiyatı ile ilgili tahmin yapabilmektir. Bu amacı gerçekleştirebilmek için olumlu, olumsuz ve nötr olarak etiketlenmiş tweetler ile haber sitesi verileri ve günlük Bitcoin fiyat verileri kullanılmıştır.

6.1.1. Ön İşleme

Twitter API kullanılarak, sistemde kaydedilmiş olan anahtar kelimelere göre sorgulama yapılarak elde edilen tweetler dosyaya kaydedilmiştir. Haber kaynakları apileri kullanılarak aynı şekilde haber verileri de dosyaya kaydedilmiştir. Ancak ücretli apiler olduğu için veri çekerken limitler bulunmaktadır. Eğitim ve test için ‘https://www.kaggle.com/datasets’ sitesi üzerinden indirilen veri setleri üzerinde ön işlemler yapılarak verinin daha okunabilir hale gelmesi sağlanmıştır.

Bitcoin hakkındaki sosyal medya ve haber kaynaklarındaki veriler İngilizce olarak tercih edilmiştir.

Bu kısımda tweet ve haber metinlerinin içerikleri, sınıflandırma aşamasından önce çeşitli işlemlerden geçirilmiştir.

Öncelikle metin içinde boş ve tekrar eden satırlar kaldırılmıştır. Url, kullanıcı etiketleri(@kullanıcıadı) ve hashtag ifadelerde geçen, # işaretleri kaldırılmıştır.

Harfler, rakamlar ve boşluklar dışındaki her şey temizlenmiştir.

Fazla boşuklar temizlenmiştir.

Tweetler ve haber metinleri içerisindeki özel karakterler, sayılar ve noktalama işaretleri kaldırılmıştır.

Tweetler ve haber metinleri küçük harfli hale dönüştürülmüştür.

Tweet alınırken kullanılan anahtar kelimeler, olumlu ya da olumsuz anlam belirtmeyen stopworld olarak tanımlanmış etkisiz kelimeler tweet içeriğinden çıkarılmıştır Tekrarlı tweetler temizlenmiştir. Aynı işlemler haber sitelerinden alınan anlık haberler için de uygulanmıştır.

6.1.2. Analiz ve Sınıflandırma

Elde edilen nihai tweetler ve haber verileri NLP teknikleri kullanılarak, duygu durumuna göre textblob ve vader kütüphaneleri kullanılarak nötr (neatural), olumlu(positive) ve olumsuz(negative) olarak sınıflandırılmıştır. Sınıflandırılmış test verileri ana model olarak seçilen algoritmalar ile eğitilmiştir.

Sınıflandırılan test verileri Bitcoin fiyat verileri ile hibrit bir hale getirilmiştir ve giriş özellikleri belirlenmiştir. Nihai veri setimizin giriş değerleri aşağıdaki gibi belirlenmiştir:

- Tarih
- Kapanış
- En yüksek fiyat
- En düşük fiyat
- Açılış fiyatı
- Hacim
- 5 günlük ortalama değer
- Volatility
- Segmentasyon

6.1.2. Makine Öğrenmesi Modelleri ile Fiyat Tahmini Sonuçları

Sınıflandırma işleminden sonra, literatür taramasında ve incelenen örnek çalışmalarda başarılı sonuçlar veren XGBoost, Doğrusal Regresyon (Linear Regression), Rasgele Orman(Random Forest – RF) ve LSTM uygulaması modülün ana algoritmaları olarak seçilmiştir.

Tahmini yapmak için gün bazında, .csv uzantılı dosyada saklanan son yayınlanan bitcoin hakkındaki tweetler için bir sınıflandırma yapılmıştır.

Eğitim setindeki tweetler baz alınarak, günlük olarak gruplanan tweetlerin duygu durumu, bitcoin fiyatlarının gün içerisindeki açılış fiyatı, kapanış fiyatı, en yüksek fiyatı, hacmi ve en düşük fiyatına ek olarak volatiliy ve 5 günlük ortalama değer de eklenerek, özellik girişleri olarak hibrit bir model veri seti oluşturulmuştur.

Daha sonra oluşturduğumuz veri seti, ana model olarak seçtiğimiz makine öğrenmesi teknikleri ile eğitilerek, sonraki gün için bitcoin fiyat tahmini yapılmıştır.

Bitcoin fiyatlarının günlük değişimine göre sonuca en yakın tahmini çıkarmak için eğitim verileri ile farklı parametreler ve farklı doğrulama yöntemi işlemleri kullanılarak test senaryoları uygulanmıştır. Farklı senaryolar ile her birinde farklı sonuçlar bulunmuştur.

En düşük RMSE'ye sahip model, Doğrusal Regresyon(Linear Regression) olarak bulunmuştur. Diğer performans kriterlerinde de tutarlı bir şekilde değerler aldığı görülmüştür.

Çizelge 6.1'de, sonuçlar analiz edilerek en iyi model için performans metrikleri gösterilmiştir.

Çizelge 6.1. LR Modellerin performans karşılaştırması sonucu

MAE	11.741562
MSE	1932.187206
RMSE	43.956651
R^2	0.999993
MAPE	0.153803
Başarı Oranı	99.846197

Farklı makine öğrenmesi teknikleri denenerek modellerin değerlendirme parametreleri ile tüm teknikler kıyaslanmıştır ve doğru sonuca en yakın algoritma olarak, R^2 değeri 1' e en yakın 0.9999993 olarak LR algoritması olarak tespit edilmiştir. Hata oranı en düşük algoritma denenmiş tüm senaryolarda, Doğrusal Regresyon modeli olarak belirlenmiştir.

6.1.2.1. Çalışmada kullanılan test senaryoları

En başarılı koşulların tespit edebilmesi için farklı senaryolarda testler yapılmıştır.

Senaryo 1: Tüm algoritmalar için, python kütüphanelerinde modellerin varsayılan parametreleri kullanılarak, 9 aylık veriler ile rasgele olarak %80 eğitim ve %20 test oranında veri bölünmesi yapılmıştır. Twitter verilerinin duygu durumları tahmin edilmiştir ve Bitcoin fiyatlarının farklı giriş verileri ile hibrit bir veri seti oluşturularak, test veri setindeki satırların sonraki gün için açılış fiyatı tahmin edilmiştir. Senaryo 1 için kullanılan algoritmaların, varsayılan parametreleri aşağıdaki gibi belirlenmiştir.

Senaryo 1 için, Doğrusal Regresyon (Linear Regression) varsayılan parametreleri:

- **fit_intercept:** Modelin bir sabit terim hesaplayıp hesaplamayacağını belirler (varsayılan: True).
- **normalize:** Girdilerin normalize edilip edilmeyeceğini belirler (artık varsayılan değildir, bunun yerine ön işleme kullanılır).

Senaryo 1 için, Random Forest (Rastgele Orman) parametreleri:

- **n_estimators:** Kullanılacak karar ağacı sayısı (Varsayılan: 100).
- **max_depth:** Her bir ağacın maksimum derinliği(Varsayılan= Hiçbiri).
- **min_samples_split:** Bir düğümün bölünmesi için gereken minimum örnek sayısı (Varsayılan: 2).
- **min_samples_leaf:** Yaprak düğümde bulunması gereken minimum örnek sayısı (Varsayılan: 1).
- **random_state:** Modelin her çalıştırmada aynı sonucu vermesi için sabit bir rastgelelik sağlayan değer (Varsayılan: Hiçbiri).

Senaryo 1 için, XGBoost (Extreme Gradient Boosting) parametreleri:

- **n_estimators:** Kullanılacak temel ağaçların sayısı (Varsayılan: 100).
- **learning_rate:** Her iterasyonda yapılacak güncellemelerin adım boyutu (Varsayılan: 0.3).
- **max_depth:** Karar ağaçlarının maksimum derinliği (Varsayılan: 6).
- **subsample:** Modelin her iterasyonda kullandığı örneklerin oranı (Varsayılan: 1.0).

- **random_state:** Rastgelelik kontrolü sağlayan değer (Varsayılan: Hiçbiri).

Senaryo 1 için, LSTM (Long Short-Term Memory) varsayılan parametreleri:

- **Units:** Hücre sayısıdır. Bu, LSTM katmanının 50 boyutlu bir çıktı üreteceği anlamına gelmektedir (Varsayılan Değer: 50).
- **Activation:** Hafıza kapısındaki aktivasyon fonksiyonu(Varsayılan Değer: 'tanh').
- **recurrent_activation:** Tekrarlı hücrelerin kapı aktivasyonu için kullanılan fonksiyon(Varsayılan Değer: 'sigmoid').
- **use_bias:** Katman ağırlıklarına bir bias (sapma) eklenip eklenmeyeceğini belirleyen değer.
  (Varsayılan Değer: True)
- **bias_initializer:** Bias vektörünün başlangıç değerlerini belirleyen değer. (Varsayılan Değer: 'zeros')
- **return_sequences:** Zaman adımlarının çıktılarını döndürüp döndürmeyeceğini belirleyen değer. Varsayılan olarak, yalnızca son zaman adımındaki çıktıyı döndürür.
  (Varsayılan Değer: False)
- **return_state:** **S**adece çıktıyı döndürecek değer. Eğer True olarak ayarlanırsa, hücre durumu ve gizli durum da döndürülür (Varsayılan Değer: False).
- **go_backwards:** Zaman adımları sırayla mı, yoksa ters sırayla mı işleneceğini belirleyen değer (Varsayılan Değer: false).
- **Stateful:** Bir partideki (batch) durumu bir sonrakine aktarıp aktarmayacağını belirleyen değer. Eğer True ise, aynı durum bir sonraki eğitim örneğine aktarılır (Varsayılan Değer: False).
- **Dropout:** Girdilerde bırakma işlemi uygulanmasını belirleyen değer. Varsayılan değeri 0.0' dır
- **recurrent_dropout:** Tekrarlı bağlantılar üzerinde bırakma işlemi uygulayan değer. Varsayılan değeri 0.0’ dır.

Çizelge 6.2’de, modellerin performans karşılaştırması görülmektedir. Çizelge 6.3’te, en iyi tahminlerin yapıldığı tarihler, tahmin değerleri ve gerçek değer ile tahmin değerleri arasındaki farklar, Çizelge 6.4’te, en kötü tahminlerin yapıldığı tarihler, tahmin değerleri ve gerçek değer ile tahmin değerleri arasındaki farklar verilmiştir.

Çizelge 6.2. Modellerin performans karşılaştırması

Model	MAE	MSE	RMSE	R^2	MAPE
Linear Regression	190.095975	5.417420e+04	232.753523	**0.999343**	0.455565
XGBoost	362.516364	2.654746e+05	515.242259	0.996779	0.850396
Random Forest	316.636008	2.944209e+05	542.605657	0.996427	0.803510
LSTM	1643.370724	4.025152e+06	2006.278214	0.951156	3.585600

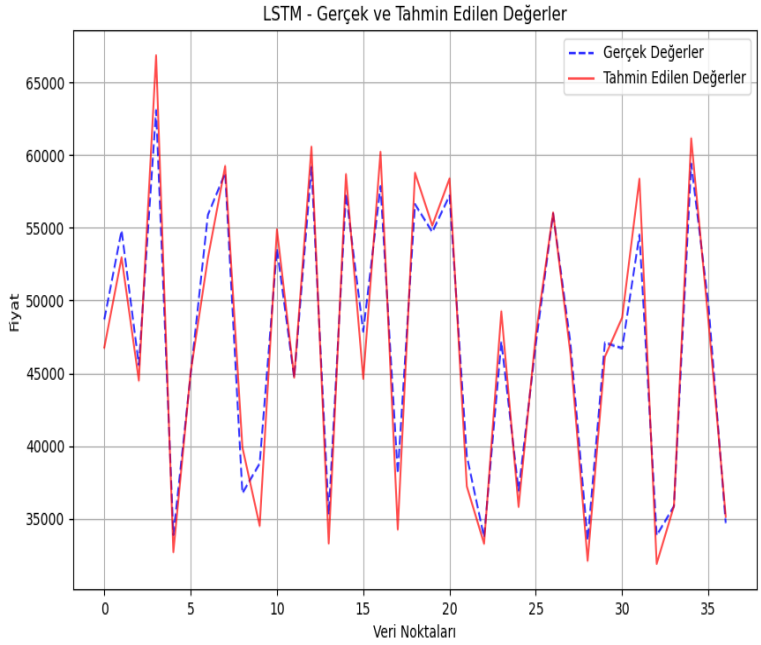
Çizelge 6.3. En iyi tahminler (USD)

Tarih	Gerçek D.	RF Tahmin	XGB Tahmin	LR Tahmin	LSTM Tahmin
2021-07-11	44898.710938	44946.260664 47.549726	44876.976562 **21.734376**	44941.596592 42.885654	45050.105469 151.394531
2021-08-28	55974.941406	55942.737734 **32.203672**	55861.960938 112.980468	55846.720272 128.221134	56036.433594 61.492188
2021-08-24	33723.507812	33735.174805 **11.666993**	33582.035156 141.472656	34028.080641 304.572829	33290.468750 433.039062
2021-08-27	47024.339844	47129.255898 104.916054	47068.277344 **43.937500**	47137.263430 112.923586	47658.976562 634.636718
2021-09-04	35854.527344	36408.848047 554.320703	35730.750000 123.777344	36127.356309 272.828965	35873.730469 **19.203125**

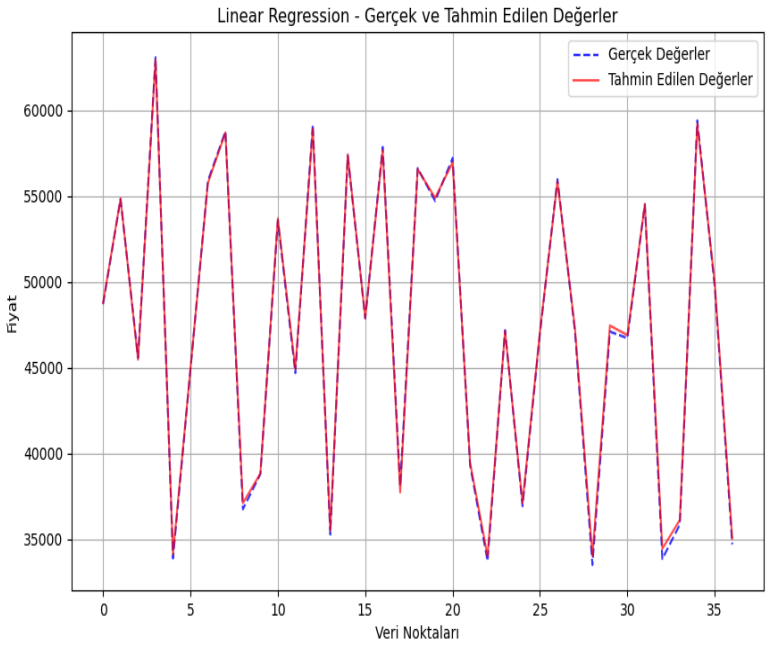
Çizelge 6.4. En kötü tahminler (USD)

Tarih	Gerçek D.	RF Tahmin	XGB Tahmin	LR Tahmin	LSTM Tahmin
2021-07-14	36753.667969	39084.986563 2331.318594	38510.523438 1756.855469	37107.386270 **353.718301**	39906.437500 3152.769531
2021-08-11	38795.781250	37785.667617 1010.113633	37781.777344 1014.003906	38828.993604 **33.212354**	34514.375000 4281.406250
2021-08-19	38099.476562	37260.571094 838.905468	37258.347656 841.128906	37737.393397 **362.083165**	34269.144531 3830.332031
2021-09-02	54511.660156	54646.892266 135.232110	54557.046875 45.386719	54522.272163 **10.612007**	58354.769531 843.109375
2021-07-09	63075.195312	63069.140430 **6.054882**	63184.238281 109.042969	62928.064159 147.131153	66843.140625 3767.945313

Şekil 6.1 ve Şekil 6.2'de, en iyi ve en kötü tahminleri yapan algoritmaların gerçek değer ve tahmin değeri üzerinden grafiksel gösterimleri bulunmaktadır.

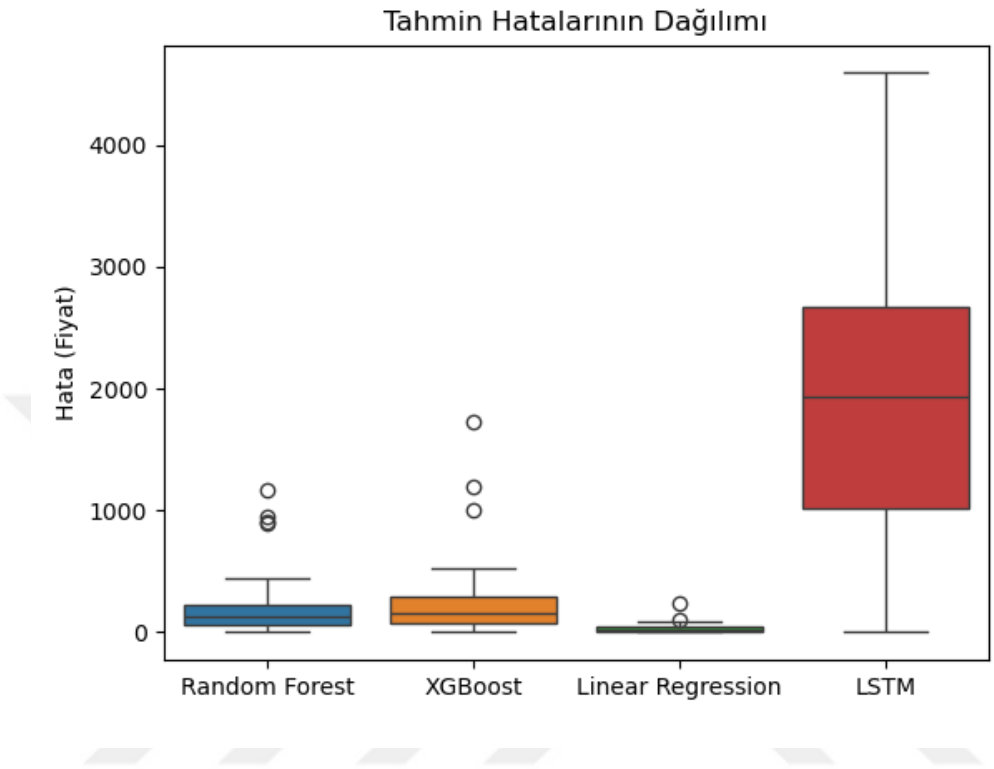


Şekil 6.1. En kötü performans sergileyen algoritma değerleri



Şekil 6.2. En iyi tahmin yapan modelin gerçek ve tahmin değerleri

Şekil 6.3'te, eğitilen algoritmalardaki hataların grafiksel gösterimi verilmiştir.



Şekil 6.3. Hataların dağılımı

Senaryo 2: Tüm algoritmalar, performansı artıracak şekilde farklı parametreler ile düzenlenerek 9 aylık veriler için twitter verilerinin duygu durumları günlük olarak belirlenmiştir ve Bitcoin fiyatlarının farklı giriş verileri ile hibrit bir veri seti oluşturularak sonraki gün açılış fiyatı tahmin edilmiştir. Veriler rasgele olarak %70 eğitim, kalan %30 verinin yarısı doğrulama veri seti, diğer yarısı ise test veri seti olarak belirlenmiştir.

Senaryo 2'de belirtilen tüm işlemler, 22/01/2021- 09/01/2023 aralığı için yaklaşık iki yıllık dönem için de oluşturulmuştur.

Algoritmaların performansını artırmak için parametrelerin varsayılan değerlerinden farklı değerler ile uygulama yapılmıştır.

Senaryo 2 için, Doğrusal Regresyon(Linear Regression) parametreler:

- Veri ölçeklendirme (standardizasyon) işlemi uygulandı.

- RMSE (Root Mean Squared Error) metriği ile doğrulama yapıldı.

Senaryo 2 için, Random Forest(Rastgele Orman Regresyonu) parametreleri:

- **n_estimators:** Kullanılan karar ağacı sayısı.
  Değerler: [100, 200, 300]
- **max_depth:** Her bir ağacın maksimum derinliği.
  Değerler: [10, 15, None]
- **min_samples_split:** Dallanma işlemi için gerekli olan minimum örnek sayısı.
  Değerler: [2, 5, 10]

Negatif ortalama kare hatası (neg_mean_squared_error) kullanılarak GridSearchCV'de değerlendirilmiştir ve uygun parametre modülü seçilmesi sağlanmıştır. Sonrasında performans metrikleri ile doğrulama ve test verileri üzerindeki performansı ölçülmüştür.

Senaryo 2 için, XGBoost (Extreme Gradient Boosting) parametreleri:

Modelin performansını artırmak için GridSearchCV ile aşağıdaki hiperparametreler optimize edilmiştir:

- **n_estimators:** Kullanılan zayıf karar ağacı sayısı.
  Değerler: [100, 200, 500]
- **learning_rate:** Öğrenme oranı, modelin her bir adımda ne kadar öğrenmesi gerektiğini belirleyen değer.
  Değerler: [0.01, 0.05, 0.1]
- **max_depth:** Karar ağaçlarının maksimum derinliği.
  Değerler: [3, 5, 10]
- **subsample:** Modelin eğitimde kullanacağı veri oranı.
  Değerler: [0.7, 0.8, 1.0]

Senaryo 2 için LSTM (Long Short-Term Memory) parametreleri:

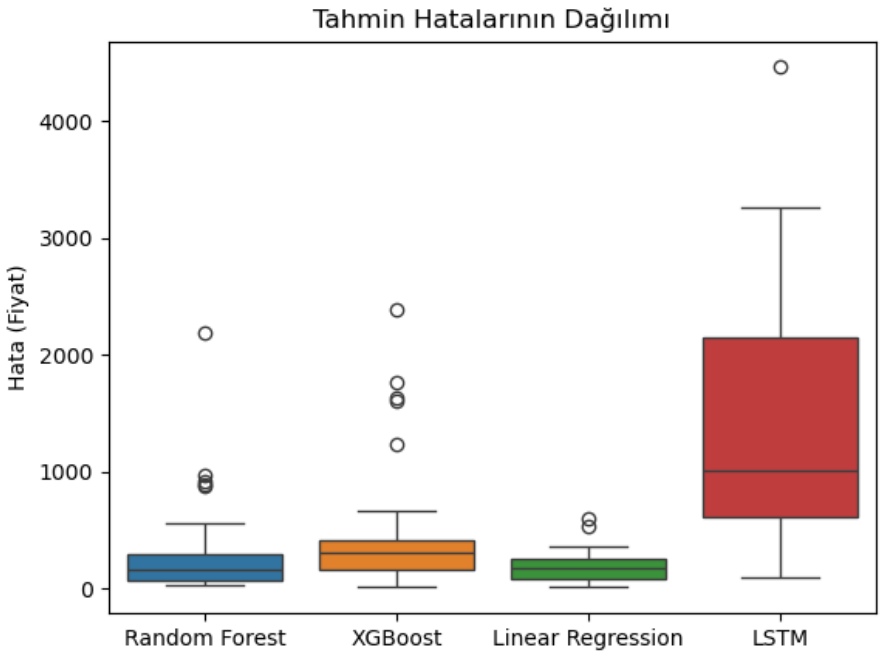
LSTM iki katman olarak ayarlanmıştır.

İlk LSTM katmanı: 50 hücre ve return_sequences=True (sonuçların bir sonraki LSTM katmanına aktarılması için).

İkinci LSTM katmanı: 50 hücre.

- **Dropout:** Her LSTM katmanının ardından %20 oranında Dropout kullanılarak aşırı öğrenmenin önlenmesi sağlanmıştır.
- **Dense:** Çıkış katmanı, tek bir çıktı üretecek şekilde ayarlanmıştır.
- **Optimizer (optimizasyon algoritması):** 'adam' algoritması kullanılmıştır.
- **Loss (kayıp fonksiyonu):** Mean Squared Error (MSE) kullanılmıştır.
- **Early Stopping:** Eğitim sırasında doğrulama kaybı (val_loss) 10 dönem boyunca iyileşmezse modelin eğitimi erken durdurulup, en iyi ağırlıklar geri yüklenmiştir.

Şekil 6.4'te, değerlendirilen algoritmalardaki hataların grafiksel gösterimi verilmiştir.



Şekil 6.4. Hataların dağılımı

Çizelge 6.5'te, modellerin performans karşılaştırması görülmektedir.

Çizelge 6.6'da, en iyi tahminlerin yapıldığı tarihler, tahmin değerleri ve gerçek değer ile tahmin değerleri arasındaki farklar verilmiştir.

Çizelge 6.7'de, en kötü tahminlerin yapıldığı tarihler, değeri ve gerçek değer ile tahmin değerleri arasındaki farklar gösterilmiştir.

Çizelge 6.5. Modellerin performans karşılaştırması

Model	MAE	MSE	RMSE	R^2	MAPE	Başarı Oranı(%)
RF	185.737501	5.874954e+04	242.383046	0.999204	0.408849	99.591151
XGBoost	272.050521	1.757269e+05	419.197905	0.997620	0.575572	99.424428
LR	25.015319	9.362960e+02	30.598955	0.999987	0.052624	**99.947376**
LSTM	1704.576172	3.979385e+06	1994.839689	0.946095	3.756405	96.243595

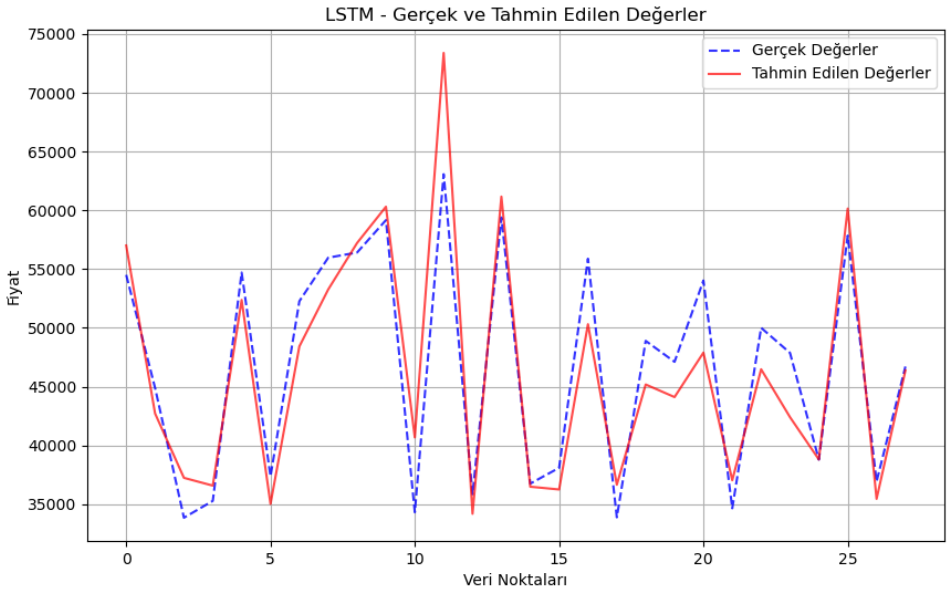
Çizelge 6.6. En iyi tahminler (USD)

Tarih	Gerçek D.	RF Tahmin	XGB Tahmin	LR Tahmin	LSTM Tahmin
2021-09-07	46716.636719	46997.241914 280.605195	46986.925781 270.289062	46910.847181 **194.210462**	46382.144531 334.492188
2021-08-19	56413.953125	56328.751667 85.201458	56366.863281 **47.089844**	56053.613271 360.339854	57226.929688 812.976563
2021-08-14	35284.343750	35477.423763 193.080013	35188.167969 **96.175781**	35593.933009 309.589259	36585.281250 1300.937500
2021-08-20	59171.933594	58977.492891 194.440703	58912.980469 258.953125	58819.916817 352.016777	60311.250000 **139.316406**
2021-08-24	59397.410156	59336.009284 61.400872	59385.832031 **11.578125**	59093.468398 303.941758	61170.707031 1773.296875

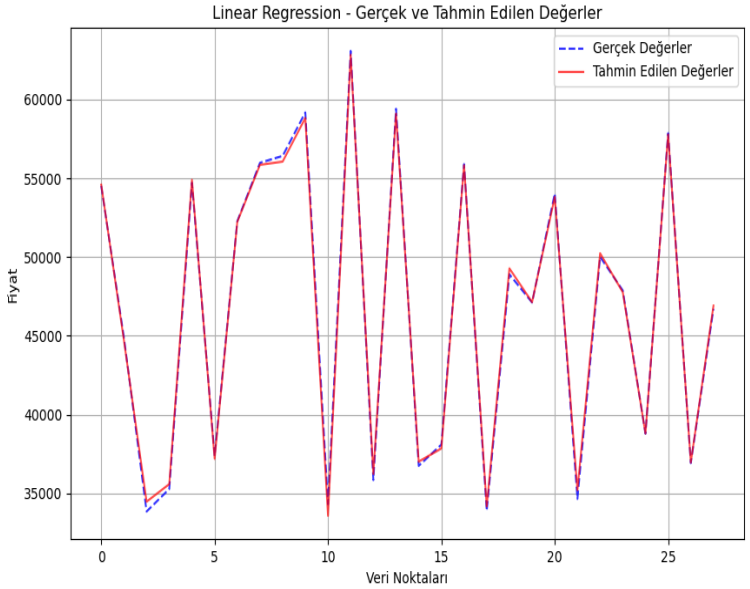
Çizelge 6.7. En kötü tahminler (USD)

Tarih	Gerçek D.	RF Tahmin	XGB Tahmin	LR Tahmin	LSTM Tahmin
2021-08-22	63075.195312	62685.697878 389.497434	63039.820312 **35.375000**	62828.336054 246.859258	73393.945312 10318.750000
2021-08-21	34318.671875	34292.306204 **26.365671**	33544.945312 773.726563	33587.747501 730.924374	40686.984375 6368.312500
2021-08-31	54030.304688	54183.976003 **153.671315**	53447.757812 582.546876	53807.299205 223.005483	47897.753906 6132.550782
2021-08-25	36753.667969	39226.757161 2473.089192	39912.722656 3159.054687	37019.003368 265.335399	36491.597656 **262.070313**
2021-08-27	55887.335938	55689.564297 197.771641	55607.414062 279.921876	55816.582609 **70.753329**	50317.070312 5570.265626

Şekil 6.5'te ve Şekil 6.6'da, en iyi tahminleri ve en kötü tahminleri yapan algoritmanın gerçek değer ve tahmin değeri üzerinden grafiksel gösterimi verilmiştir.



Şekil 6.5. En kötü tahmini yapan algoritmanın gerçek değer ve tahmini



Şekil 6.6. En iyi tahmin yapan algoritmanın gerçek değer ve tahmini

Aynı senaryo parametreleri ile, 22/01/2021- 09/01/2023 dönem aralığı için de denemeler yapılmıştır. En başarılı sonuçlara Doğrusal Regresyon ile erişilmiştir. Çizelge 6.8'de, Doğrusal Regresyon performans kriter sonuçları gösterilmektedir.

Çizelge 6.8. Doğrusal regresyon performans kriterleri

Performans Kriterleri	Doğrusal Regresyon Sonuçları
MAE	165.534894
MSE	85737.217043
RMSE	292.809182
R^2	0.999591
MAPE	0.581474

Senaryo 3: Tüm algoritmalar varsayılan parametreler ile düzenlenerek 22/01/2021 - 09/09/2021 dönem aralığı için, 9 aylık veriler ile twitter verilerinin duygu durumunu belirten segmentasyon giriş verisi olmadan, Bitcoin fiyatlarının farklı giriş verileri bir veri seti oluşturularak sonraki gün açılış fiyatı tahmin edilmiştir.

Parametreler Senaryo 1'deki gibi ayarlanmıştır. Veri seti rasgele %80 eğitim, %20 test olacak şekilde bölünmüştür.

Çizelge 6.9'da, modellerin performans karşılaştırması verilmiştir.

Çizelge 6.10'da, en iyi tahminlerin yapıldığı tarihler ve tahmin değerleri ile gerçek değer ve tahmin değerleri arasındaki farklar verilmiştir.

Çizelge 6.11'de en kötü tahminlerin yapıldığı tarihler, değerleri ve gerçek değer ile tahmin değerleri arasındaki farkları verilmiştir

Çizelge 6.9. Modellerin performans karşılaştırması

Model	MAE	MSE	RMSE	R^2	MAPE
LR	32.296263	2.508743e+03	50.087355	0.999969	0.076753
XGBoost	207.572448	1.139644e+05	337.586167	0.998602	0.488471
RF	255.551519	1.680295e+05	409.913956	0.997939	0.600481
LSTM	1293.765017	2.692129e+06	1640.770737	0.966972	2.948587

Çizelge 6.10. En iyi tahminler (USD)

Tarih	Gerçek D.	RF Tahmin	XGB Tahmin	LR Tahmin	LSTM Tahmin
2021-08-21	31533.884766	31472.993184 60.891582	31530.402344 3.482422	31535.531604 **1.646838**	31491.138672 42.746094
2021-08-09	57343.371094	57446.965977 103.594883	57339.167969 4.203125	57303.921896 39.449198	57374.976562 **1.605468**
2021-08-05	44898.710938	44983.815000 85.104062	44747.984375 150.726563	44916.937348 **18.226410**	45163.609375 264.898437
2021-08-13	33509.078125	33730.550039 221.471914	33766.949219 257.871094	33519.451608 **10.373483**	33435.566406 73.511719
2021-08-07	56068.566406	56013.167109 55.399297	56234.230469 165.664063	56095.832612 **27.266206**	55749.433594 319.132812

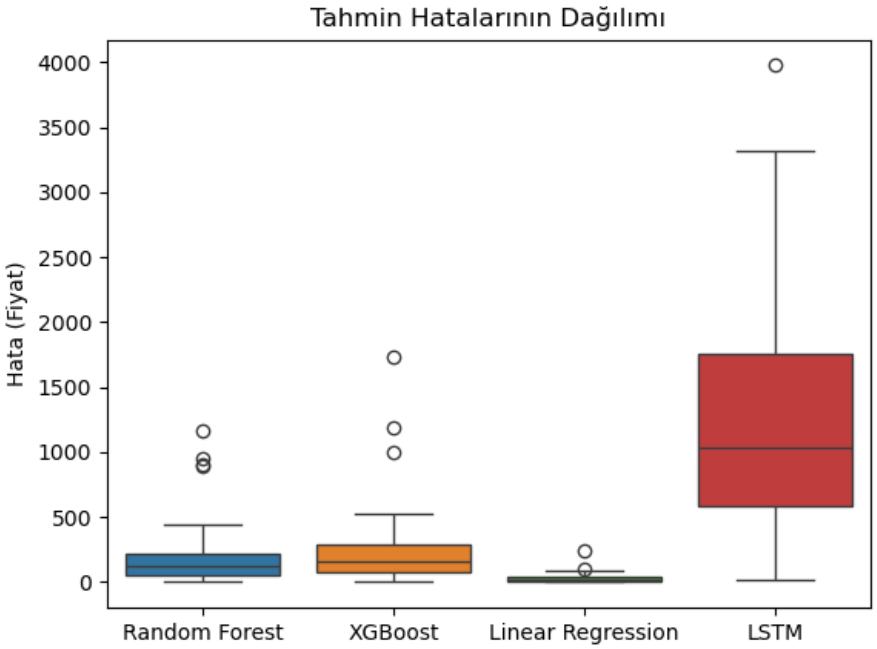
.

Çizelge 6.11. En kötü tahminler (USD)

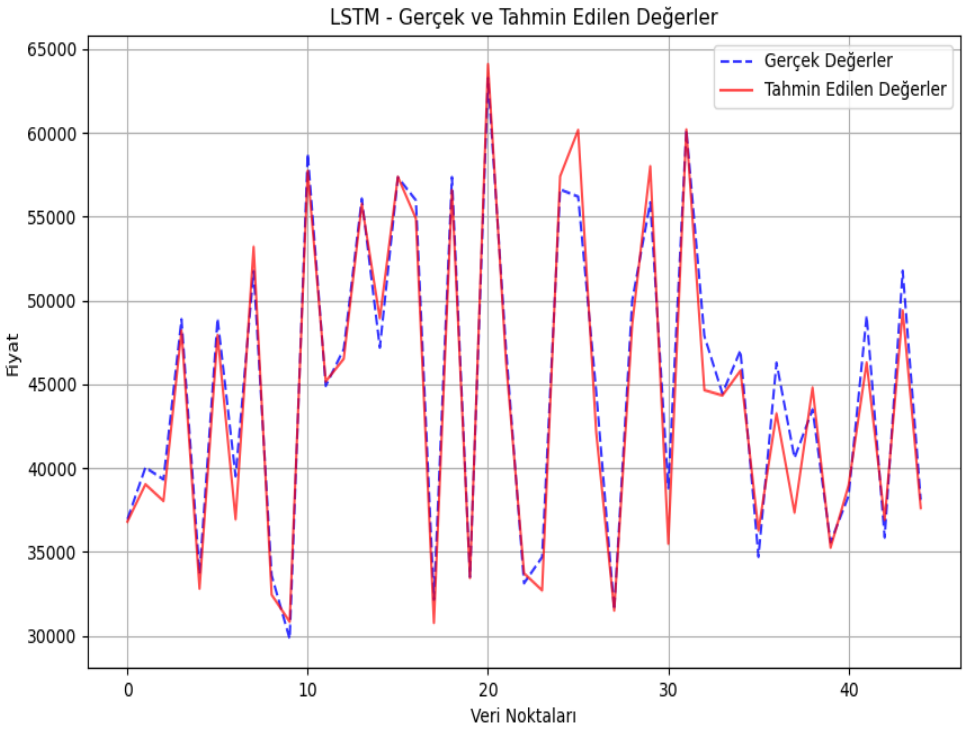
Tarih	Gerçek D.	RF Tahmin	XGB Tahmin	LR Tahmin	LSTM Tahmin
2021-08-20	44574.437500	43621.396953 953.040547	42844.917969 1729.519531	44553.019098 **21.418402**	42138.496094 2435.941406
2021-08-19	56191.585938	56119.890898 71.695040	55904.496094 287.089844	56211.998007 **20.412069**	60166.199219 3974.613281
2021-08-24	38795.781250	38523.683555 272.097695	38406.414062 389.367188	38752.064047 **43.717203**	35481.429688 3314.351562
2021-08-26	47877.035156	47630.775000 246.260156	47719.394531 157.640625	47924.961920 **47.926764**	44647.585938 3229.449218
2021-08-31	40596.949219	40541.647031 55.302188	40574.535156 **22.414063**	40835.280522 238.331303	37337.652344 3259.296875

Şekil 6.7’de, değerlendirilen algoritmalardaki hataların dağılımının grafiksel gösterimleri sunulmuştur.

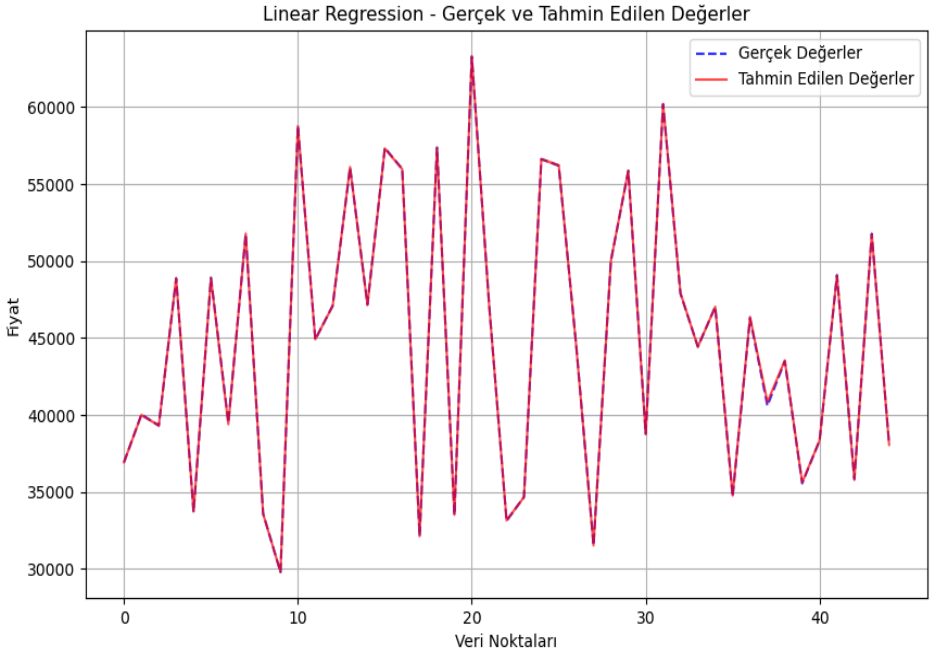
Şekil 6.8’de, en iyi tahminleri yapan algoritmaların gerçek değer ve tahmin değeri üzerinden grafiksel gösterimleri ve Şekil 6.9’da, en kötü tahminleri yapan algoritmaların gerçek değer ve tahmin değeri üzerinden grafiksel gösterimleri verilmiştir.



Şekil 6.7. Hataların dağılımı



Şekil 6.8. En Kötü Tahmin Yapan Algoritmanın Gerçek Değer Ve Tahmini



Şekil 6.9. En İyi Tahmin Yapan Algoritmanın Gerçek Değer Ve Tahmini

Senaryo 4: Tüm algoritmalar performansı artıracak şekilde parametreler ile düzenlenerek 9 aylık veriler ile twitter verilerinin duygu durumunu belirten segmentasyon giriş verisi olmadan, Bitcoin fiyatlarının farklı giriş verileri bir veri seti oluşturularak sonraki gün açılış fiyatı tahmin edilmiştir. Parametreler Senaryo 2' deki gibi ayarlanmıştır. Veri seti rasgele %80 eğitim, %20 test olacak şekilde ayrılmıştır.

Çizelge 6.12'de, modellerin performans karşılaştırması verilmiştir.

Çizelge 6.12. Modellerin Performans Karşılaştırması

Model	MAE	MSE	RMSE	R^2	MAPE
Linear Regression	32.296263	2.508743e+03	50.087355	0.999969	0.076753
XGBoost	198.894618	1.074221e+05	327.753050	0.998682	0.493172
Random Forest	207.572448	1.139644e+05	337.586167	0.998602	0.488471
LSTM	2212.647396	6.823013e+06	2612.089700	0.916292	5.218857

Çizelge 6.13'te, en iyi tahminlerin yapıldığı tarihler, tahmin değerleri ve gerçek değerler ile tahmin değerleri arasındaki farklar verilmiştir.

Çizelge 6.13. En İyi Tahminler

Tarih	Gerçek D.	RF Tahmin	XGB Tahmin	LR Tahmin	LSTM Tahmin
2021-08-18	56620.273438	56619.857773 **0.415665**	56493.718750 126.554688	56612.863864 7.409574	56548.601562 71.671876
2021-08-29	34700.363281	34573.301523 127.061758	34678.855469 **21.507812**	34773.967104 73.603823	34517.675781 182.687500
2021-08-01	51739.808594	51907.326836 167.518242	51825.218750 85.410156	51782.056761 **42.248167**	52080.906250 341.097656
2021-08-08	47180.464844	46960.218125 220.246719	47057.847656 122.617188	47137.128829 **43.336015**	47597.218750 416.753906
2021-09-05	35854.527344	35788.362266 66.165078	35659.152344 195.375000	35791.124662 **63.402682**	35167.515625 687.011719

.

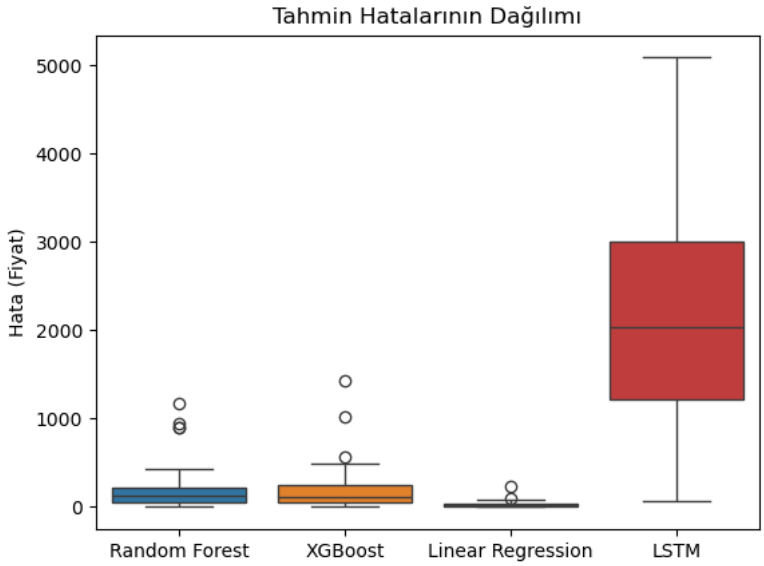
Çizelge 6.14'te, en kötü tahminlerin yapıldığı tarihler, tahmin değerleri ve gerçek değerler ile tahmin değerleri arasındaki farklar verilmiştir.

.

Çizelge 6.14. En Kötü Tahminler

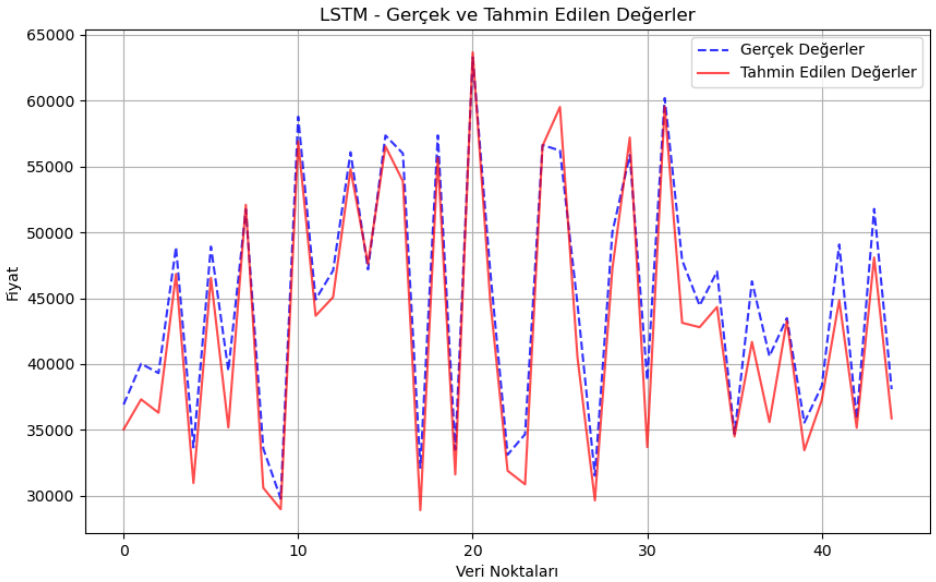
Tarih	Gerçek D.	RF Tahmin	XGB Tahmin	LR Tahmin	LSTM Tahmin
2021-08-20	44574.437500	43621.396953 953.040547	43556.675781 1017.761719	44553.019098 **21.418402**	40532.390625 4042.046875
2021-08-24	38795.781250	38523.683555 272.097695	38491.371094 304.410156	38752.064047 **43.717203**	33703.558594 5092.222656
2021-08-31	40596.949219	40541.647031 **55.302188**	40293.332031 303.617188	40835.280522 238.331303	35598.878906 4998.070313
2021-07-31	39503.187500	39065.997148 437.190352	39235.988281 267.199219	39404.015459 99.**172041**	35192.558594 4310.628906
2021-08-26	47877.035156	47630.775000 246.260156	47818.988281 58.046875	47924.961920 **47.926764**	43128.859375 4748.175781

Şekil 6.10'da, hataların dağılımının grafiksel gösterimleri bulunmaktadır.

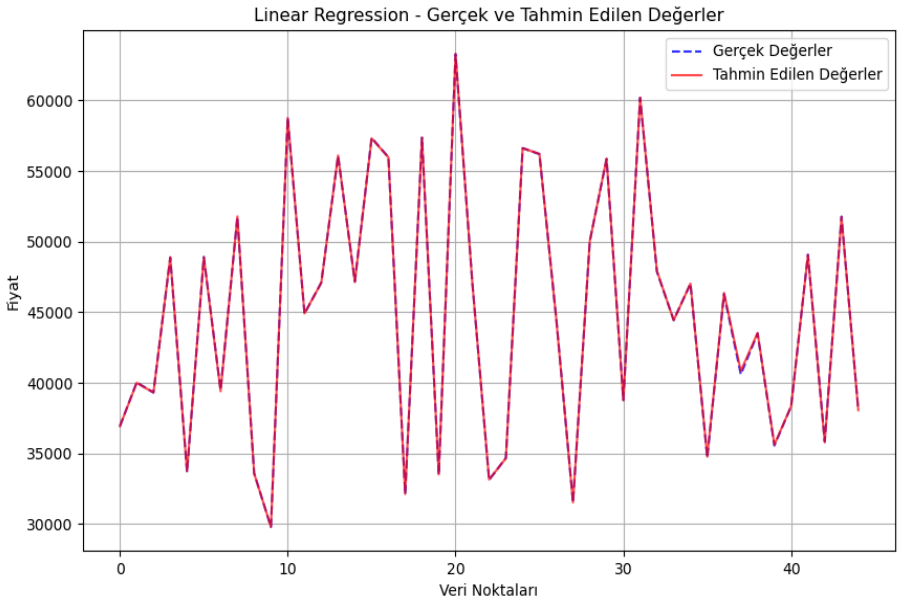


Şekil 6.10. Hataların dağılımı

Şekil 6.11 ve Şekil 6.12'de, en kötü ve iyi performans ile kötü performans gösteren algoritma fiyat tahminini gösterecek sonuç değerlendirmeleri verilmiştir.



Şekil 6.11. En Kötü Tahmin Yapan Algoritmanın Gerçek Değer Ve Tahmini



Şekil 6.12. En iyi tahmin yapan algoritmanın gerçek değer ve tahmini

Senaryo 5: Tüm algoritmalar performansı artıracak şekilde parametreler ile düzenlenerek 01/01/2014 - 01/01/2024 arasındaki 10 yıllık veriler ile twitter verilerinin duygu durumunu belirten segmentasyon giriş verisi olmadan, Bitcoin fiyatlarının farklı giriş verileri ile bir veri seti oluşturularak sonraki gün açılış fiyatı tahmini yapılmıştır.

Parametreler Senaryo 2' deki gibi ayarlanmıştır. Veri seti rasgele %80 eğitim, %20 test olacak şekilde ayrılmıştır.

Çizelge 6.15'te, modellerin performans karşılaştırması verilmiştir.

Çizelge 6.15. Modellerin performans karşılaştırması

Model	MAE	MSE	RMSE	R^2	MAPE
LR	9.018968	3.870389e+03	62.212450	0.999992	0.098674
XGBoost	34.313612	4.041477e+04	201.034241	0.999913	0.240854
RF	111.821933	6.091535e+04	246.810345	0.999868	2.510509
LSTM	1311.787999	2.807336e+06	1675.510653	0.993935	109.627943

Çizelge 6.16'da, en iyi tahminlerin yapıldığı tarihler, tahmin değerleri ve gerçek değerler ile tahmin değerleri arasındaki farklar verilmiştir.

Çizelge 6.16. En iyi tahminler (USD)

Tarih	Gerçek D.	RF Tahmin	XGB Tahmin	LR Tahmin	LSTM Tahmin
2024-03-21	61554.921875	61555.399102 **0.477232**	61562.156250 7.234380	61518.754063 36.167807	61558.535156 3.613286
2022-12-12	21819.005859	21806.769219 12.236631	21767.921875 51.083975	21825.008523 6.002673	21820.570312 **1.564462**
2023-08-08	16836.472656	16833.851699 2.620951	16836.451172 **0.021478**	16834.885305 1.587345	16757.037109 79.435541
2024-05-31	16796.976562	16808.810977 11.834417	16836.451172 39.474612	16796.524442 **0.452118**	16746.580078 50.396482
2024-09-29	5066.577637	5083.245142 16.667505	5134.577637 68.000000	5066.888778 **0.311141**	5085.776367 19.198730

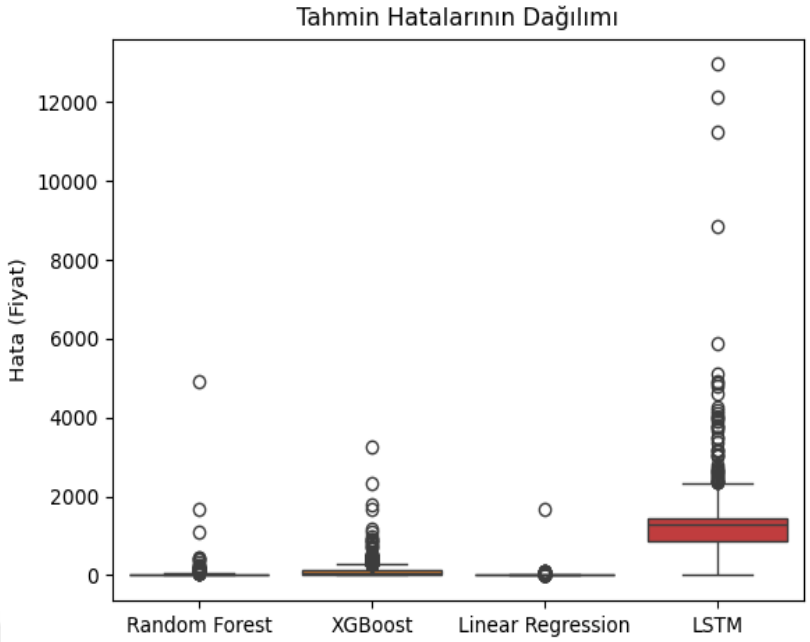
.

Çizelge 6.17'de, en kötü tahminlerin yapıldığı tarihler, tahmin değerleri ve gerçek değerler ile tahmin değerleri arasındaki farklar verilmiştir.

Çizelge 6.17. En kötü tahminler (USD)

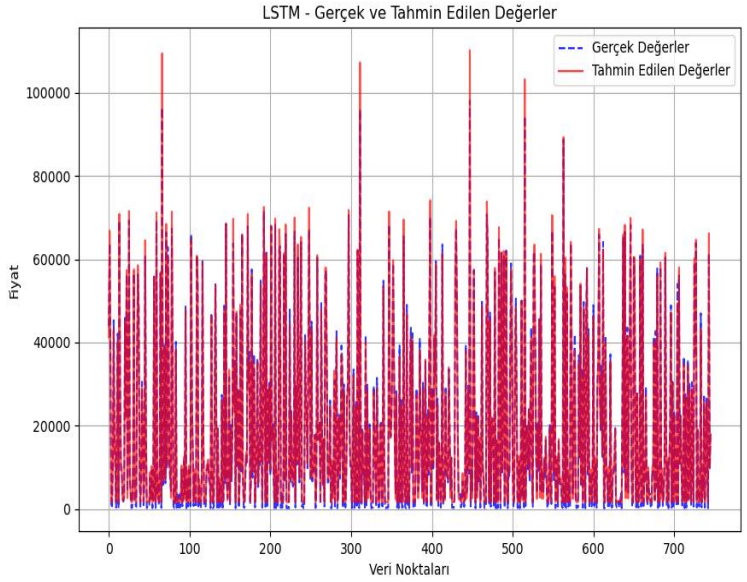
Tarih	Gerçek D.	RF Tahmin	XGB Tahmin	LR Tahmin	LSTM Tahmin
2023-01-26	96461.335938	96712.653594 251.317656	94112.242188 2349.093750	96443.129903 **18.206035**	109423.859375 12962.523437
2024-04-19	94334.640625	93225.320938 1109.319687	97601.820312 3267.179687	94336.453593 **1.812968**	103186.843750 8852.203125
2023-09-28	95988.531250	95994.163203 5.631953	97780.851562 1792.320312	95982.976841 **5.554409**	107228.695312 11240.164062
2024-02-11	98033.445312	97618.183281 415.262031	97743.468750 289.976562	98000.280965 **33.164347**	110150.578125 12117.132813
2023-04-24	63776.050781	63799.740430 23.689649	64662.222656 886.171875	63796.672164 20.621383	69666.500000 5890.449219

Şekil 6.13'te, değerlendirilen algoritmaların hatalarının dağılımının grafiksel gösterimleri bulunmaktadır.

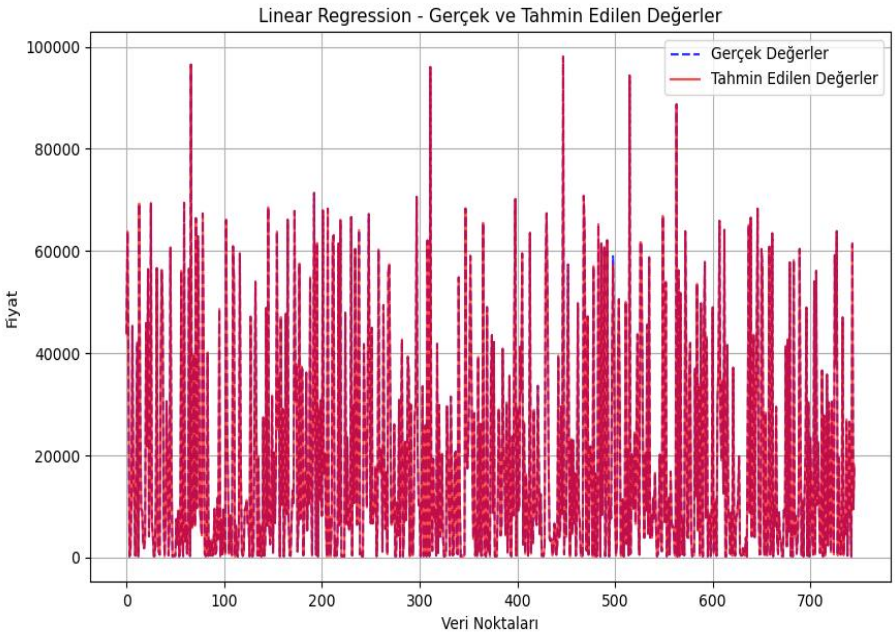


Şekil 6.13. Hataların Dağılımı

Şekil 6.14 ve Şekil 6.15'te, iyi performans ile kötü performans gösteren algoritma fiyat tahminini gösterecek sonuçları verilmiştir.



Şekil 6.14. En kötü tahmin yapan algoritmanın gerçek değer ve tahmini



Şekil 6.15. En iyi tahmin yapan algoritmanın gerçek değer ve tahmini

Senaryo 6: Tüm algoritmalar performansı artıracak şekilde parametreler ile düzenlenerek, 01/01/2014 – 01/01/2024 arasındaki 10 yıllık veriler ile twitter verilerinin duygu durumunu belirten segmentasyon giriş verisi olmadan, Bitcoin fiyatlarının farklı giriş verileri ile bir veri seti oluşturularak sonraki gün açılış fiyatı tahmini yapılmıştır.

Parametreler aşağıdaki şekilde ayarlanmıştır. Veri seti rasgele %70 eğitim, %30 test ve doğrulama olacak yarı yarıya bölündü. Bu senaryoda hem doğrulama verisi hem de çapraz doğrulama kullanılmıştır.

LR Modeli için kullanılan parametreler:

fit_intercept: Modelin y-intercept (kesim noktası) öğrenip öğrenmeyeceğini belirleyen değer (Varsayılan: True).

normalize: Özelliklerin otomatik olarak normalize edilip edilmeyeceğini belirleyen değer (Varsayılan: False).

n_jobs: Model eğitiminde kullanılacak işlemci çekirdek sayısı (Varsayılan: None).

Şekil 6.16'da LR Modeli için kullanılan hiperparametreler gösterilmektedir.

```
# GridSearchCV parametreleri
param_grid = {
    'fit_intercept': [True, False],
    'n_jobs': [-1,3, None],
    'positive': [True, False]
}
lr_grid_search = GridSearchCV(
    LinearRegression(),
    param_grid,
    cv=5,                                  # 5 katlı doğrulama
    scoring='neg_mean_squared_error'       # Negatif ortalama kare hatası
)
```

Şekil.6.16. LR Modeli parametreleri

Hiperparametre (param_grid) kullanılarak tüm olası kombinasyonlar oluşturulmuştur.

Parametrelere Göre İşlem Adımları:

12 hiperparametre kombinasyonu belirlenmiştir. Kombinasyon sayısı, her hiperparametre değerinin çarpımı ile hesaplanmıştır. Burada fit_intercept için 2 değer, n_jobs için 3 değer, positive için 2 değer kullanılmıştır.

Toplam Kombinasyon Sayısı = 2×3×2 = 12

Çapraz Doğrulama (CV) ile Eğitim: Her kombinasyon için 5 katlı çapraz doğrulama(CV) yapılmıştır.

Veri kümesi, her bir hiperparametre kombinasyonu için 5 farklı eğitim/doğrulama alt kümesine ayrılmıştır. Her bir alt küme için model eğitilmiştir ve doğrulama skoru hesaplanmıştır.

Her Kombinasyon için Eğitim Sayısı: Eğitim Sayısı (Iterasyon) olarak toplam 60 iterasyon gerçekleşmiştir. Çapraz doğrulama kat sayısı 5, Toplam kombinasyon sayısı 12 bulunur.

Toplam Eğitim Sayısı = Kombinasyon Sayısı × CV Kat Sayısı

Toplam Eğiitim Sayısı = 12×5 = 60

Negatif Ortalama Kare Hatası (MSE) Skoru Hesaplama: Performans metriği olarak negatif ortalama kare hata (MSE; Skor Metriği) kullanılmıştır. En düşük MSE skoru en iyi model olarak seçilmiştir.

Her bir hiperparametre kombinasyonu için 5 katlı çapraz doğrulama yapılır ve bu 5 doğrulama skorunun ortalaması alınarak genel bir skor hesaplanması yapılmıştır.

En İyi Modelin Seçilmesi: Çapraz doğrulama sonucu, en iyi hiperparametre kombinasyonunu belirlenmiştir.

Seçilen kombinasyon, modelin, 'best_params_' özelliğinde saklanmaktadır. En iyi model ise, 'best_estimator_' özelliğinde bulunmaktadır.

RF Modeli için parametreler:

Şekil 6.17'de, LR Modeli için kullanılan hiperparametreler verilmiştir.

```
# Random Forest için GridSearchCV Parametreleri
rf_param_grid = {
    'n_estimators': [100, 200, 300],        # Ağaç sayısı
    'max_depth': [10, 15, None],            # Ağaç derinliği
    'min_samples_split': [2, 5, 10]         # Numune bölme sayısı
}

# GridSearchCV ile Random Forest Modeli Eğitme
rf_grid_search = GridSearchCV(
    RandomForestRegressor(random_state=42), # Rastgele durum parametresi
    rf_param_grid,                          # Parametre grid'i
    cv=5,                                   # 5 katlı doğrulama
    scoring='neg_mean_squared_error'        # Negatif ortalama kare hatası
)
```

Şekil 6.17. RF Modeli hiperparametreleri

Parametrelere Göre İşlem Adımları:

Parametrelere göre RF modelinin kombinasyonları aşağıdaki şekilde hesaplanmıştır.

n_estimators için 3 değer, max_depth için 3 değer, min_samples_split için 3 değer.

Toplam Kombinasyon: 3 x 3 x 3 = 27

Çapraz Doğrulama (cv=5): Her bir kombinasyon için model, veriyi 5 parçaya bölmektedir ve 5 farklı eğitim/doğrulama işlemi gerçekleştirilmiştir. Böylece her kombinasyon 5 kez değerlendirilmiştir.

Toplam Eğitim Sayısı:

27 kombinasyon × 5 çapraz doğrulama katı = 135 model eğitimi.

XGBoost Modeli için parametreler:

Şekil 6.18'de, XGBoost Modeli için kullanılan hiperparametreler gösterilmektedir.

```python
# XGBoost için GridSearchCV Parametreleri
xgb_param_grid = {
    'n_estimators': [100, 200, 500],              # Ağaç sayısı
    'learning_rate': [0.01, 0.05, 0.1],           # Öğrenme oranı
    'max_depth': [3, 5, 10],                      # Ağaç derinliği
    'subsample': [0.7, 0.8, 1.0]                  # Alt numunelerin yüzdesi
}

# GridSearchCV ile XGBoost Modeli Eğitme
xgb_grid_search = GridSearchCV(
    XGBRegressor(random_state=42),                # Rastgele durum parametresi
    xgb_param_grid,                               # Parametre grid'i
    cv=5,                                         # 5 katlı doğrulama
    scoring='neg_mean_squared_error'              # Negatif ortalama kare hatası
)
```

Şekil 6.18. XGBoost Modeli hiperparametreleri

Parametrelere Göre İşlem Adımları:

Hiperparametre Kombinasyonlarının Sayısı: Her hiperparametrenin aldığı değerler: n_estimators: 3 değer, learning_rate: 3 değer, max_depth: 3 değer, subsample: 3 değer

Toplam Kombinasyon Sayısı = 3×3×3×3 = 81

Toplam 81 farklı hiperparametre kombinasyonu denenmiştir.

Çapraz Doğrulama Kat Sayısı: Çapraz doğrulama için veri kümesi 5 alt kümeye bölünücek şekilde ayarlanmıştır.

Her hiperparametre kombinasyonu için 5 farklı model eğitilir ve doğrulama yapılmıştır.

Toplam Eğitim (Iterasyon) Sayısı:

81 kombinasyon × 5 çapraz doğrulama katı = 405 model eğitimi.

Toplamda 405 iterasyon gerçekleşmiştir.

Performans metriği olarak negatif ortalama kare hata (neg_mean_squared_error) kullanılmıştır. En düşük hata veren kombinasyon seçilmiştir.

LSTM Modeli parametreleri

Şekil 6.19'da, LSTM Modeli için kullanılan hiperparametreler gösterilmektedir.

Sequential**:** Sequential sınıfı**,** katmanları sıralı bir şekilde tanımlanmıştır. Bu modelde, sırasıyla bir LSTM katmanı, bir dropout katmanı, bir başka LSTM katmanı, tekrar bir dropout katmanı ve son olarak bir tam bağlı (dense) katmandan oluşturulmuştur.

```
# LSTM Modeli
lstm_model = Sequential([
    LSTM(50, activation='relu', return_sequences=True, input_shape=(X_train_lstm.shape[1], 1)),
    Dropout(0.2),
    LSTM(50, activation='relu'),
    Dropout(0.2),
    Dense(1)
])

lstm_model.compile(optimizer='adam', loss='mse')
early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)

# Modeli Eğitme
history = lstm_model.fit(X_train_lstm, y_train,
                         epochs=50, batch_size=32,
                         validation_data=(X_test_lstm, y_test),
                         callbacks=[early_stopping])
```

Şekil 6.19. LSTM Modeli hiperparametreleri

Parametrelere Göre İşlem Adımları:

Model Yapısının Tanımlanması: LSTM modelinin katmanları tanımlanmıştır. Sequential ile 50 birimlik bir LSTM katmanı belirlenmiştir. Aktivasyon fonksiyonu olarak ReLU (Rectified Linear Unit) seçilmiştir, return_sequences=True ile de bir sonraki katmana sıralı çıktılar sağlayacak şekilde ayarlanmıştır (örneğin, başka bir LSTM katmanına bağlanmak için).

LSTM(50):

Bu LSTM katmanında 50 adet nöron bulunacak şekilde ayarlama yapılmıştır.

activation='relu':

LSTM hücreleri içindeki aktivasyon fonksiyonunu belirtilmiştir.

relu, negatif değerleri sıfıra sabitler ve pozitif değerleri geçiş yapmaktadır.

return_sequences=True:

Bir sonraki katman tekrar bir LSTM olduğu için, bu parametre True olarak ayarlanmıştır.

input_shape=(X_train_lstm.shape[1], 1):

Modelin giriş boyutunu belirtilmiştir.

X_train_lstm.shape[1]: Zaman serisi uzunluğu (Bir veri noktasının kaç zaman adımı içerdiği).

1: Her zaman adımındaki özellik sayısı (Bir zaman adımında yalnızca fiyat bilgisi olduğundan 1 olarak belirlendi).

Dropout(0.2): Dropout(0.2), her eğitim iterasyonunda ağırlıkların %20'sini rastgele sıfırlayacak şekilde ayarlanmıştır. Modelin fazla ezber yapmasını engellenmiştir.

LSTM(50, activation='relu'): İkinci bir LSTM katmanı belirlenmiştir. return_sequences=False (varsayılan) olarak ayarlanmıştır. Bu LSTM katmanı son katmandır ve yalnızca son zaman adımındaki çıktıyı döndürecek şekilde ayarlanmıştır. Diğer parametreler ilk LSTM katmanı ile aynı olacak şekilde belirlenmiştir.

Dropout(0.2): İkinci Dropout katmanı, önceki LSTM katmanının overfitting yapmasını önlemek için uygulanmıştır.

Dense(1): Bu, modelin çıkış katmanıdır. Bu katman yalnızca bir nöron içerecek şekilde ayarlanmıştır.

Çıkış, modelin tahmin ettiği değer olarak belirlenmiştir (bir sonraki zaman adımındaki fiyat tahmini).

Modelin Derlenmesi: Model derlenmiştir. Erken durdurma ayarlanmıştır (patience=10).

optimizer='adam': Adam optimizasyon algoritması kullanılmıştır. Bu, öğrenme oranını dinamik olarak ayarlayan bir optimizasyon yöntemidir.

loss='mse'; Kayıp fonksiyonu olarak Ortalama Kare Hatası (MSE) kullanılmıştır.

early_stopping ile erken durdurma ayarları belirlenmiştir.

monitor='val_loss' (Eğitim sırasında doğrulama kaybı izlenir), patience=10 (Doğrulama kaybı 10 ardışık dönem boyunca iyileşmezse eğitim durdurulacak şekilde ayarlayan değer), restore_best_weights=True(Eğitim sonunda doğrulama setindeki en iyi ağırlıklar geri yüklenecek şekilde belirleyen değer) olarak ayarlanmıştır.

Modelin Eğitilmesi: Maksimum 50 dönem boyunca model eğitilmiştir. Ancak erken durdurma etkin olduğu için eğitim daha erken bitmiştir.

batch_size=32: Eğitim sırasında veri seti, 32 örneklik alt gruplara bölünerek, model bu alt gruplar üzerinde güncellenmiştir.

validation_data=(X_test_lstm, y_test): Eğitimden bağımsız olarak doğrulama seti (validation data) kullanılmıştır. Her dönemin sonunda val_loss hesaplanmıştır.

Iterasyon Sayısının Hesaplanması:

1 Epoch İçerisindeki Iterasyon Sayısı

Bir epoch, tüm eğitim veri setinin modelden geçirilmesi işlemidir. Eğitim veri seti batch_size boyutundaki alt gruplara bölünmektedir.

Eğitim veri setindeki örnek sayısı: 3650

Iterasyon Sayısı (Bir Epoch için) = 3650/32 =114

Toplam Epoch Sayısı: Maksimum epochs=50 olarak verilmiş, ancak erken durdurma (patience=10) etkin olduğundan, 10 epoch boyunca doğrulama kaybında iyileşme olmadığı eğitim daha erken durdurulduğu görülmüştür.

Eğitim, 27 epoch'ta durduğu için, iterasyon sayısı:

Toplam Iterasyon Sayısı = 27×114= 3078 iterasyon

Toplamda 3078 iterasyon olarak çalıştığı hesaplanmıştır.

Çizelge 6.18'de, modellerin validasyon için performans karşılaştırması verilmiştir. Çizelge 6.19'da, modellerin test için performans karşılaştırması verilmiştir.

Çizelge 6.20'de, en iyi tahminlerin yapıldığı tarihler, tahmin değerleri ve gerçek değerler ile tahmin değerleri arasındaki farklar verilmiştir.Çizelge 6.21'de, en kötü tahminlerin yapıldığı tarihler, tahmin değerleri ve gerçek değerler ile tahmin değerleri arasındaki farklar verilmiştir. Çizelge 6.22'de başarı oranları verilmiştir.

Çizelge 6.18. Modellerin validasyon performans karşılaştırması

Model	MAE	MSE	RMSE	R^2	MAPE
RF	28.759749	6066.667081	77.888812	0.999977	0.28575
XGBoost	76.69196	32607.684945	180.575981	0.999875	0.752909
LR	8.939073	321.715293	17.936424	0.999999	0.150425
LSTM	338.225885	350333.55177	591.889814	0.99866	9.32268

Çizelge 6.19. Modellerin test performans karşılaştırması

Model	MAE	MSE	RMSE	R^2	MAPE
RF	38.567789	11791.330668	108.587894	0.999959	0.258656
XGBoost	99.370928	88516.681065	297.517531	0.999692	0.734241
LR	11.741562	1932.187206	43.956651	0.999993	0.153803
LSTM	365.749138	435689.780783	660.068012	0.998482	12.366600

Çizelge 6.20. En iyi tahminler (USD)

Tarih	Gerçek D.	RF Tahmin	XGB Tahmin	LR Tahmin	LSTM Tahmin
2022-08-10	449.687988	449.367738 **0.320250**	447.407318 2.280670	448.344877 1.343111	450.373047 0.685059
2023-03-23	446.062012	446.047829 **0.014183**	444.679169 1.382843	444.894859 1.167153	450.411316 4.349304
2023-07-24	455.756012	455.617801 **0.138211**	449.810028 5.945984	454.538283 1.217729	455.297272 0.458740
2023-02-24	444.290985	444.084202 **0.206783**	439.569580 4.721405	443.056781 1.234204	448.493835 4.202850
2023-12-16	465.208008	463.906885 1.301123	464.181091 1.026917	464.284739 **0.923269**	454.493652 10.714356

Çizelge 6.21. En kötü tahminler (USD)

Tarih	Gerçek D.	RF Tahmin	XGB Tahmin	LR Tahmin	LSTM Tahmin
2023-02-06	66953.335938	65996.287480 957.048458	62285.980469 4667.355469	66986.511362 **33.175424**	68721.828125 1768.492187
2022-08-09	30817.625000	30948.509990 130.884990	31657.626953 840.001953	30854.545540 **36.920540**	34853.550781 4035.925781
2023-10-03	53252.164062	53685.897324 -433.733262	54000.984375 748.820313	53364.815953 **112.651891**	56250.890625 2998.726563
2023-07-16	22487.986328	22761.794102 273.807774	22696.095703 208.109375	22523.201177 **35.214849**	26224.916016 3736.929688
2023-06-16	48835.085938	49107.229434 272.143496	48786.011719 40049.074219	48794.533486 **40.552452**	69666.500000 20831.414062

Çizelge 6.22. Modellerin Başarı Oranları

Model	Başarı Oranı
LR Test Ortalama başarı oranı (%)	**99.84619663056965**
RF Test Ortalama başarı oranı (%)	99.74134394928248
RF Test Ortalama başarı oranı (%)	99.26575874940758
LSTM Test Ortalama başarı oranı (%)	87.63339967403444

Şekil 6.20'de yapılan işlemde, tahmin edilen değer ile gerçek değer arasındaki farkın, gerçek değere oranını yüzdelik olarak hesaplanmıştır.

```
# # Test Verisi Üzerinde Tahmin
lstm_test_preds = lstm_model.predict(X_test_lstm).flatten()

# Sadece değerleri numpy array olarak almak
y_test_values = y_test.values

# Yüzdelik fark hesabı
percentage_error_test = np.abs((lstm_test_preds - y_test_values) / y_test_values) * 100
success_rate_test = 100 - percentage_error_test

# Ortalama başarı oranı
average_success_rate_test = np.mean(success_rate_test)
```

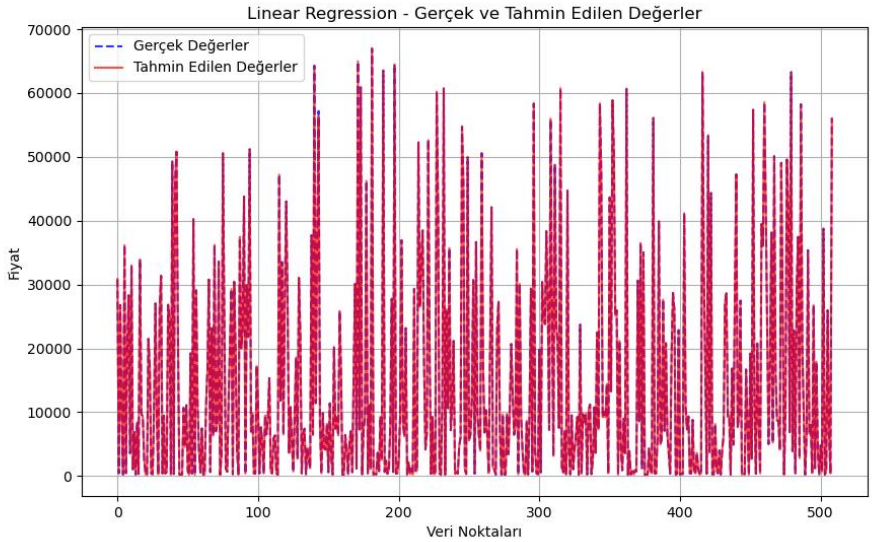
Şekil 6.20. LSTM Modeli başarı oranı

Mutlak değer alınır, böylece negatif hatalar pozitif hale getirilmiştir.

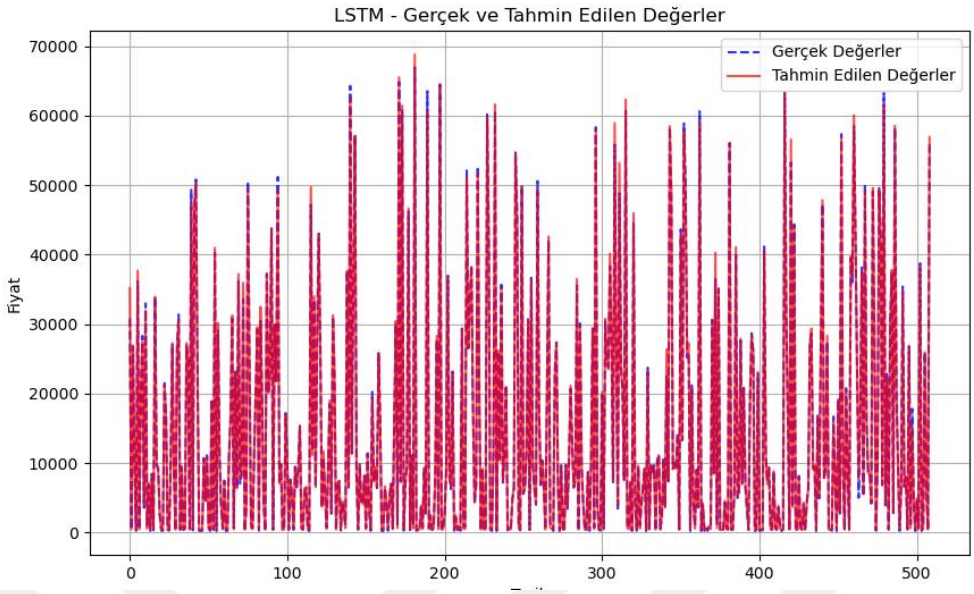
Başarı Oranı; 100 − Yüzdelik Hata ile bulunmuştur.

Test setindeki her bir veri noktası için hesaplanan başarı oranlarının ortalaması alınır. Bu hesaplama da, test setindeki genel başarı oranını vermektedir.

Şekil 6.21' de, en iyi performans gösteren algoritma fiyat tahminini gösterecek sonuçları verilmiştir. Şekil 6.22' de, en kötü performans gösteren algoritma fiyat tahminini gösterecek sonuçları verilmiştir.



Şekil 6.21. En iyi tahmin yapan algoritmanın gerçek değer ve tahmini



Şekil 6.22. En kötü tahmin yapan algoritmanın gerçek değer ve tahmini

Tüm senaryolarda en iyi sonucu veren Doğrusal Regresyon olduğu, RF ve XGBoost modelleri LR modelini takip ettiği görülmüştür.

Bir önceki senaryolarda en kötü sonucu veren LSTM modeli için, büyük veri seti kullanıldığında, işlemler sonucunda LSTM modelinin de performansının arttığı gözlemlenmiştir.

Çizelge 6.23'te, 9 aylık veriler ile denenmiş senaryolar için en başarılı olarak bulunan Doğrusal Regresyon algoritmasının performans sonuçları verilmiştir.

Çizelge 6.23. Doğrusal Regresyon aylık performans değerlendirmesi

Model	MAE	MSE	RMSE	R^2	MAPE
Segmentli Varsayılan Parametre - 9 Aylık Data	190.095975	54174.202496	232.753523	0.999343	0.455565
Segmentli Gelişmiş Parametre - 9 Aylık Data	25.015319	936.296039	30.598955	0.999987	0.052624
Segmentsiz Varsayılan Parametre - 9 Aylık Data	32.296263	2508.743171	50.087355	0.999969	0.076753
Segmentsiz Gelişmiş Parametre 9 Aylık Data	8.956590	622.537293	24.950697	**0.999999**	0.098373

Çizelge 6.24'te, 10 yıllık veriler ile denenmiş senaryolar için en başarılı olarak bulunan Doğrusal Regresyon algoritmasının performans sonuçları gösterilmektedir.

Çizelge 6.24. Doğrusal Regresyon 10 yıllık senaryolardaki performans değerlendirmesi

Segmentsiz Gelişmiş Parametre - 10 Yıllık Data	9.018968	3870.388939	62.212450	0.999992	0.098674
Doğrulama Verisi - Segmentsiz Gelişmiş Parametre - 10 Yıllık Data	8.939073	321.715293	17.936424	**0.999999**	0.150425
Test Verisi - Segmentli Gelişmiş Parametre - 10 Yıllık Data	11.741562	1932.187206	43.956651	0.999993	0.153803

Çizelge 6.25'te, 10 yıllık ve 4 yıllık veriler ile denenmiş senaryolar için CV kullanılarak, en başarılı olarak bulunan Doğrusal Regresyon algoritmasının performans sonuçları gösterilmektedir.

Çizelge 6.25. Doğrusal regresyon CV ile senaryolardaki performans değerlendirmesi

Test Verisi - Segmentsiz Gelişmiş Parametre - 10 Yıllık Data	11.741562	1932.187206	43.956651	0.999993	0.153803
Test Verisi - Segmentsiz Gelişmiş Parametre - 4 Yıllık Data	17.886950	1323.309194	36.377317	**0.999994**	0.064372

7. SONUÇLAR VE ÖNERİLER

7.1. Sonuçlar

Tez kapsamında Bitcoin fiyatı için, makine öğrenmesi yöntemlerinden XGBoost, Doğrusal Regresyon (Lineer Regression), Random Forest ve LSTM ile tahmin yapılmıştır. Farklı zaman dilimlerinden oluşan veri setleri ile en başarılı doğruluk oranlarına on yıllık verilerle ulaşılmıştır. Yöntem olarak Doğrusal Regresyon (Lineer Regression) algoritmasının başarılı olduğu görülmüştür. Gerçek değerler incelendiğinde gerçeğe yakın değerlerin tahmin edildiği ortaya çıkmıştır. Gerçek değerler ile yapılan sonuçlar incelendiğinde mutlak hata yüzdelerinin en düşük on yıllık verilerde olduğu gözlemlenmiştir. Çalışmanın bütün aşamalarında farklı senaryolar ile, akademik bilgileri genişletmeye katkı sağlamak için denemeler yapılmıştır. Sonuçlar yatırım tavsiyesi niteliğinde değildir.

7.2. Öneriler

Veri seti içeriğinde, segmentasyon verilerine ek olarak tweet veya haber kaynaklarında çekilen metinlerin ekstra özellikleri de dahil edilerek giriş verileri artırılarak denemelere devam edilebilir. Zaman maliyeti gibi bir durum olmadığı durumlarda haber metinleri de daha geniş aralıklarda incelenerek algoritmaların doğruluğuna etkisi incelenebilir. Makine öğrenmesinden farklı olarak derin öğrenme ile tahminler genişletilebilir. Duygu durumu tespitinde metinler için gerçek zamanlı haber veya tweeter apisinin gerçek zamanlı verileri kullanılarak, ertesi gün tahminleri için çalışmalar canlı ortama taşınarak, gerçek zamanda çalışmalara devam edilebilir.

KAYNAKLAR

Alpaydın, E., 2010, Introduction to Machine Learning, *London,England*, p.

Asif, R., Merceron, A., Ali, S. A. ve Haider, N. G., 2017, Analyzing undergraduate students' performance using educational data mining, *Computers & education*, 113, 177-194.

Balakrishnama, S. ve Ganapathiraju, A., 1998, Linear discriminant analysis-a brief tutorial, *Institute for Signal and information Processing*, 18 (1998), 1-8.

Bansal, M., Goyal, A. ve Choudhary, A., 2022, A comparative analysis of K-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning, *Decision Analytics Journal*, 3, 100071.

Bentejac, C., Csörgo, A. ve Martínez-Mu, G., 2019, A Comparative Analysis of XGBoost.

Bishop, C. M., 2006, Pattern Recognition and Machine Learning, *U.K*, p.

Boateng, E. Y. ve Abaye, D. A., 2019, A Review of the Logistic Regression Model with Emphasis on Medical Research, *Journal of Data Analysis and Information Processing*, 190-207.

Chen, J., 2023, Analysis of bitcoin price prediction using machine learning, *Journal of Risk and Financial Management*, 16 (1), 51.

Chen, T. ve Guestrin, C., 2016, XGBoost: A Scalable Tree Boosting System, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 785 - 794.

Chen, Z., Li, C. ve Sun, W., 2020, Bitcoin price prediction using machine learning: An approach to sample dimension engineering, *Journal of Computational and Applied Mathematics*, 365, 112395.

Chopra, A., Prashar, A. ve Sain, C., 2013, Natural language processing, *International journal of technology enhancements and emerging engineering research*, 1 (4), 131-134.

Cortez, K., Rodríguez-García, M. d. P. ve Mongrut, S., 2020, Exchange market liquidity prediction with the K-nearest neighbor approach: Crypto vs. fiat currencies, *Mathematics*, 9 (1), 56.

Gülmez, B., 2023, Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm, *Expert Systems with Applications*, 227, 120346.

He, Y. ve Zhou, D., 2011, Self-training from labeled features for sentiment analysis, *Information Processing & Management*, 47 (4), 606-616.

Huynh, T. L. D., 2023, When Elon Musk changes his tone, does bitcoin adjust its tune?, *Computational Economics*, 62 (2), 639-661.

Joshi, R. D. ve Dhakal, C. K., 2021, Predicting type 2 diabetes using logistic regression and machine learning approaches, *International journal of environmental research and public health*, 18 (14), 7346.

Karaarslan, E. ve Akbaş, M. F., 2017, blokzinciri tabanli siber güvenlik sistemleri, *Uluslararası Bilgi Güvenliği Mühendisliği Dergisi*, 3 (2), 16-21.

Kok, Z. H., Shariff, A. R. M., Alfatni, M. S. M. ve Khairunniza-Bejo, S., 2021, Support vector machine in precision agriculture: a review, *Computers and Electronics in Agriculture*, 191, 106546.

Kufel, J., Bargieł-Łączek, K., Kocot, S., Koźlik, M., Bartnikowska, W., Janik, M., Czogalik, Ł., Dudek, P., Magiera, M. ve Lis, A., 2023, What is machine learning, artificial neural networks and deep learning?—Examples of practical applications in medicine, *Diagnostics*, 13 (15), 2582.

Liaw, A. ve Wiener, M., 2002, Classification and regression by randomForest. R news.

Leong, W., Kelani, R. ve Ahmad, Z., 2020, Prediction of air pollution index (API) using support vector machine (SVM), *Journal of Environmental Chemical Engineering*, 8 (3), 103208.

Liddy , E. D., 2001, Natural Language Processing.

Lin, J., Zhong, C., Hu, D., Rudin, C. ve Seltzer, M., 2020, Generalized and scalable optimal sparse decision trees, *International Conference on Machine Learning*, 6150-6160.

Mahmoodzadeh, A., Nejati, H. R., Mohammadi, M., Ibrahim, H. H., Rashidi, S. ve Rashid, T. A., 2022, Forecasting tunnel boring machine penetration rate using LSTM deep neural network optimized by grey wolf optimization algorithm, *Expert Systems with Applications*, 209, 118303.

Mailagaha Kumbure, M. ve Luukka, P., 2022, A generalized fuzzy k-nearest neighbor regression model based on Minkowski distance, *Granular Computing*, 7 (3), 657-671.

Mladenova, T. ve Valova, I., 2023, Classification with K-Nearest Neighbors Algorithm: Comparative Analysis between the Manual and Automatic Methods for K-Selection, *International Journal of Advanced Computer Science and Applications*, 14 (4).

Murphy, K. P., 2012, Machine Learning A Probabilistic Perspective, *London, England*, p.

Nakamoto, S., 2008, Satoshi Nakamoto.

Osisanwo, F., Akinsola, J., Awodele, O., Hinmikaiye, J., Olakanmi, O. ve Akinjobi, J., 2017, Supervised machine learning algorithms: classification and comparison, *International Journal of Computer Trends and Technology (IJCTT)*, 48 (3), 128-138.

Pang, B., Lee, L. ve Vaithyanathan, S., 2002, Thumbs up? Sentiment classification using machine learning techniques, *arXiv preprint cs/0205070*.

Pang, B. ve Lee, L., 2004, A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts, *arXiv preprint cs/0409058*.

Peling, I. B. A., Arnawan, I. N., Arthawan, I. P. A. ve Janardana, I. G. N., 2017, Implementation of Data Mining To Predict Period of Students Study Using Naive Bayes Algorithm, *Int. J. Eng. Emerg. Technol*, 2 (1), 53.

Reddy, N. A., Sai, B. S. K. ve Chary, S. S., 2022, A Novel Prediction Model for Cryptocurrency Trend Analysis Based on Time Series Data by Using Machine Learning Techniques.

Rodriguez, M. Z., Comin, C. H., Casanova, D., Bruno, O. M., Amancio, D. R., Costa, L. d. F. ve Rodrigues, F. A., 2019, Clustering algorithms: A comparative approach, *PloS one*, 14 (1), e0210236.

Sajana, T., Rani, C. S. ve Narayana, K., 2016, A survey on clustering techniques for big data mining, *Indian journal of Science and Technology*, 9 (3), 1-12.

Sarker, I. H., Kayes, A., Badsha, S., Alqahtani, H., Watters, P. ve Ng, A., 2020, Cybersecurity data science: an overview from machine learning perspective, *Journal of Big data*, 7, 1-29.

Sarker, I. H., 2021, Machine learning: Algorithms, real-world applications and research directions, *SN computer science*, 2 (3), 160.

Schober, P. ve Vetter, T. R., 2021, Logistic regression in medical research, *Anesthesia & Analgesia*, 132 (2), 365-366.

Shahiri, A. M. ve Husain, W., 2015, A review on predicting student's performance using data mining techniques, *Procedia Computer Science*, 72, 414-422.

Shalev-Shwartz, S. ve Ben-David, S., 2014, Understanding Machine Learning:, Cambridge University Press, p.

Strauss, T. ve Von Maltitz, M. J., 2017, Generalising Ward’s method for use with Manhattan distances, *PloS one*, 12 (1), e0168288.

Şevgin, F., 2023, Gayrimenkul Piyasasında Regresyon Yöntemleri ile Veri Madenciliği, Muş Alparslan Üniversitesi, Mühendislik Mimarlık Fakültesi Dergisi, 4(2), 20-28.

Tripathi, A. M., Shukla, S. ve Chaudhary, N., 2024, Estimation and Prediction of Bitcoin by Applying Machine Learning, *2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT)*, 191-196.

Tufail, S., Riggs, H., Tariq, M. ve Sarwat, A. I., 2023, Advancements and challenges in machine learning: A comprehensive review of models, libraries, applications, and algorithms, *Electronics*, 12 (8), 1789.

Whitelaw, C., Garg, N. ve Argamon, S., 2005, Using appraisal groups for sentiment analysis, *Proceedings of the 14th ACM international conference on Information and knowledge management*, 625-631.

Wongkar, M. ve Angdresey, A., 2019, Sentiment analysis using Naive Bayes Algorithm of the data crawler: Twitter, *2019 Fourth International Conference on Informatics and Computing (ICIC)*, 1-5.

Wu, J. M.-T., Li, Z., Herencsar, N., Vo, B. ve Lin, J. C.-W., 2023, A graph-based CNN-LSTM stock price prediction algorithm with leading indicators, *Multimedia Systems*, 29 (3), 1751-1770.

Ye, J., Janardan, R. ve Li, Q., 2004, Two-dimensional linear discriminant analysis, *Advances in neural information processing systems*, 17.

Yermack, D., 2024, Is Bitcoin a real currency? An economic appraisal, In: Handbook of digital currency, Eds: Elsevier, p. 29-40.