

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



ÜRETİM HATLARI İÇİN MAKİNE ÖĞRENİMİ TEMELLİ
KESTİRİMCİ BAKIM MODELİ GELİŞTİRME

19011037 – Zeynep ACAR

19011059 – Betül ÇELİK

BİLGİSAYAR PROJESİ

Danışman
Doç. Dr. Serkan AYVAZ

Haziran, 2023

TEŞEKKÜR

Bu çalışmanın yürütülmesi sırasında desteğini esirgemeyen, kendisine danıştığımız her alanda bize yardımcı olan, kıymetli vaktini bize ayırmaktan çekinmeyen değerli danışmanımız Doç.Dr. Serkan AYVAZ hocamıza teşekkür ediyor ve şükranlarımızı sunuyoruz.

Zeynep ACAR

Betül ÇELİK

İÇİNDEKİLER

KISALTMA LİSTESİ	v
ŞEKİL LİSTESİ	vi
TABLO LİSTESİ	vii
ÖZET	viii
ABSTRACT	ix
1 GİRİŞ	1
2 ÖN İNCELEME	3
2.1 Kalan Ömür Tahmini İçin Derin Öğrenme Yöntemiyle Geliştirilmiş Sistem	3
2.2 Kestirimci Bakım İçin İstatiksel Yaklaşım	3
2.3 Yapay Zeka Teknikleri İle Sistemin Önleyici Bakımı	4
3 FİZİBİLİTE	5
3.1 Teknik Fizibilite	5
3.1.1 Yazılım Fizibilitesi	5
3.1.2 Donanım Fizibilitesi	6
3.2 Zaman Fizibilitesi	6
3.3 Yasal Fizibilite	7
3.4 Ekonomik Fizibilite	7
4 SİSTEM ANALİZİ	8
4.1 Veri Seti	8
4.2 Korelasyon Matrisi	9
5 SİSTEM TASARIMI	11
5.1 Özellik Seçimi	11
5.1.1 Temel Bileşen Analizi (PCA)	12
5.2 ML Algoritmaları	12
5.2.1 Random Forest	12

5.2.2	Karar Ağacı (Decision Tree)	13
5.2.3	Gradyan Güçlendirme (Gradient Boosting)	14
5.2.4	XGBoost Algoritması	15
5.2.5	MLP Algoritması	16
5.3	Eğitim ve Test Setleri	16
6	UYGULAMA	18
7	Deneyisel Sonuçlar	19
7.1	Veri Kümesi İşlemleri	19
7.2	Random Forest İle Seçilen Özelliklere Uygulanan Modeller	21
7.2.1	Model Sonuçları	21
8	Performans Analizi	24
8.0.1	MAE	24
8.0.2	R^2	24
8.0.3	MAPE	25
8.0.4	MSE	25
8.0.5	RMSE	25
9	Sonuç	26
	Referanslar	28
	Özgeçmiş	30

KISALTMA LİSTESİ

AI	Artificial Intelligence
ARIMA	Autoregressive Integrated Moving Average
BSD	Berkeley Software Distribution
C-MAPPS	Commercial Modular Aero-Propulsion System Simulation
CNN	Convolutional Neural Networks
IoT	Internet of Things
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MLP	Multi Layer Perceptron
MSE	Mean Squared Error
NASA	National Aeronautics and Space Administration
PCA	Principal Component Analysis
PdM	Predictive Maintenance
R^2	R-Squared
RMSE	Root Mean Squared Error
RNN	Reccurent Neural Network
RUL	Remaining Useful Lifetime
SVR	Support Vector Regression
XGBoost	Extreme Gradient Boosting

ŞEKİL LİSTESİ

Şekil 3.1	GANTT Çizelgesi	6
Şekil 4.1	Kestirimci Bakım Modeli Mimarisi	9
Şekil 4.2	Korelasyon Tablosu	10
Şekil 5.1	Sistem İşleyiş Süreci	11
Şekil 5.2	Random Forest Algoritması	12
Şekil 5.3	Karar Ağacı Yapısı	13
Şekil 5.4	Gradyan Güçlendirme Algoritması	14
Şekil 5.5	XGBoost Algoritması	15
Şekil 5.6	MLP Algoritması	16
Şekil 7.1	Özelliklerin PC'lere Varyans Katkıları	20
Şekil 7.2	Random Forest Algoritmasınana Göre Özellik Önemleri	21
Şekil 7.3	Algoritma Sonuç Karşılaştırmaları	23

TABLO LİSTESİ

Tablo 3.1	Donanım Bilgileri	6
Tablo 7.1	Random Forest ile En İyi Sonuçlar	21
Tablo 7.2	En İyi Parametreler	22

ÜRETİM HATLARI İÇİN MAKİNE ÖĞRENİMİ TEMELLİ KESTİRİMCİ BAKIM MODELİ GELİŞTİRME

Zeynep ACAR

Betül ÇELİK

Bilgisayar Mühendisliği Bölümü

Bilgisayar Projesi

Danışman: Doç. Dr. Serkan AYVAZ

Günümüz dünyasındaki dijitalleşme hayatımızda büyük rol almaktadır. Firmalar da bu gelişime ayak uydurup daha verimli sonuçlar elde etmeye çalışmaktadır. Üretim firmaları için cihazların bakımlarının ne zaman yapılması gerektiğinin önceden tahmin edilmesi büyük önem taşır. Bu doğrultuda firmalar gereksinimleri kolaylaştıran akıllı çözümlere ihtiyaç duyar. Bu çözümler yapay zeka ve makine öğrenimi sayesinde gerçekleştirilebilir.

Bu çalışmada gerçek zamanlı üretilen sensör verileri kullanılarak üretim hatları için makine öğrenmesi temelli kestirimci bakım modeli geliştirilmiştir. Makine öğrenimi algoritmaları kullanılarak arıza gerçekleşmeden önce ne zaman gerçekleşeceğine dair bir tahminde bulunulması amaçlanmıştır. Sonuç olarak önceden gerekli önlemlerin alınarak üretim hattının duraksamdan işleyişe devam etmesine katkı sağlanmıştır.

Boyut indirgeme ve özellik seçimi uygulanarak makine öğrenimi modelleri bu veriler üzerinde eğitilmiştir. Model sonuçları çeşitli metrikler kullanılarak karşılaştırılmıştır. Elde edilen değerlendirmeler sonucunda bir ensemble algoritması olan Random Forest ve boosting yöntemi olan XGBoost algoritmalarının yüksek performans sergilediği ortaya çıkmıştır.

Anahtar Kelimeler: Makine öğrenimi algoritmaları, önleyici bakım sistemi, özellik seçimi, boyut indirgeme, regresyon

DEVELOPING A PREDICTIVE MAINTENANCE MODEL FOR PRODUCTION LINES IN MANUFACTURING BASED ON MACHINE LEARNING

Zeynep ACAR

Betül ÇELİK

Department of Computer Engineering
Computer Project

Advisor: Assoc. Prof. Dr. Serkan AYVAZ

Digitalization in today's world plays a major role in our lives. Companies are trying to keep up with this development and achieve more efficient results. For manufacturing companies, it is of great importance to predict when the maintenance of the devices should be done. In this direction, companies need smart solutions that facilitate requirements. These solutions can be realized thanks to artificial intelligence and machine learning.

In this study, a machine learning-based predictive maintenance model was developed for production lines by using real-time sensor data. By using machine learning algorithms, it is aimed to make an estimation of when the failure will occur before it happens. As a result, the necessary precautions were taken in advance and it contributed to the continuation of the production line without stopping.

Machine learning models are trained on this data by applying dimension reduction and feature selection. Model results were compared using various metrics. As a result of the evaluations, it has been revealed that Random Forest, which is an ensemble algorithm, and XGBoost, which is a boosting method, exhibit high performance.

Keywords: Machine learning, predictive maintenance system, feature selection, dimension reduction, regression

Bugünün dünyasında yaşanan dijital gelişim Artificial Intelligence (AI) ve Big Data'nın günlük hayatımızı şekillendirmesini sağlar ve iş süreçlerinin verimliliğinin birçok alanda arttırılmasını sağlar. Dördüncü sanayi devriminin temelini IoT ve AI gibi örnek verilebilecek teknolojik ve bilimsel gelişmeler oluşturmaktadır. Bu dönemde AI ve Big Data karmaşık sorunlara çözümler sunarak modern insanların yaşama ve çalışma şeklini değiştirmektedir. Bu değişimle beraber firmalar bulundukları sektördeki piyasaların yoğun rekabeti sebebiyle sürekli olarak performanslarını arttırma ihtiyacı duymaktadır. Bu doğrultuda dönemin teknolojik gelişmelerine adapte olabilen firmalar bu rekabetçi ortamda bir adım öne geçmektedir.

Günümüzde veriler farklı kaynaklardan çeşitli yöntemlerle toplanmaktadır. Bu durum toplanan veriyi daha karmaşık bir yapıya dönüştürerek veri analizini zorlaştırmaktadır. Analiz sonuçları; gerçek zamanlı operasyonların yönetimi, risklerin ve hataların tespiti gibi önemli konularda daha iyi kararların alınması için kullanılır. IoT cihazları üretim sistemleri arasındaki bağlantı ve veri alışverişinin yapılmasını sağlar.

Üretim ortamı ekipmanların düzgün çalışmasına bağlıdır. Herhangi bir bileşende veya alt sistemde meydana gelen bir arıza tüm üretim hattının durmasına ve firmanın mali zarara uğramasına sebebiyet verebilir. Tüm hataların önceden tespit edilmesi mümkün olmasa da duraksamaya sebep olan hataların büyük bir bölümü öngörülebilir. Böylece hatanın giderilmesi üretimde duraksama ve olası bir mali kayıp olmadan gerçekleştirilebilir[1].

Bakım maliyetlerinin azaltılması ve operasyonların sürdürülebilir bir şekilde yürütülebilmesi için kestirimci bakım (PdM Predictive Maintenance) yaygın hale gelmiştir. Kestirimci bakımın ana hedefi önleyici bakımın gerçekleştirilmesi için arızanın oluşma zamanını önceden tahmin etmektir.

Veri odaklı olarak çalışan yapay zeka uygulamaları, IoT cihazlarından toplanan verileri kullanarak önleyici bakımın tespit edilmesinde yardımcı olabilir. Bu çalışmanın

amacı, üretim hatlarında olası arızaları gerçekçi bir şekilde tahmin eden ve bunu makine öğrenimi yöntemleri kullanarak gerçekleştiren önleyici bir bakım sistemi gerçekleştirmektir. Bu probleme en uygun modeli elde etmek için bir firmadan alınan gerçek dünya veri seti kullanılarak birden fazla algoritma detaylı bir şekilde incelenmiş ve karşılaştırılmıştır

Bu raporda öncelikle literatürdeki diğer benzer konularda yapılan çalışmalar incelenmiştir. Fizibilite bölümünde proje aşamalarının nasıl olacağı ve proje kapsamında nasıl bir şekilde ilerleneceği anlatılmıştır. Sistem analizinde projenin hedefleri ve gereksinimleri tanımlanmıştır. Sistem tasarımında yöntemler tanıtılmıştır. Uygulanan modellerin sonuçları ve nasıl değerlendirildiği deneysel sonuçlar ve performans analizi bölümlerinde açıklanmıştır.

2.1 Kalan Ömür Tahmini İçin Derin Öğrenme Yöntemiyle Geliştirilmiş Sistem

Zhang ve diğerleri tarafından gerçekleştirilen "Deep Learning for Improved System Remaining Life Prediction" isimli bu çalışmada, derin öğrenme temelli bir yaklaşım izlenerek üretim sistem performansı ve bozulma tahmini için bir model geliştirilmiştir. Sistem durumunun bozulmasını takip etmek için özellikle varyasyon desenlerini ortaya çıkarma yeteneğine sahip derin bir mimari olan Long Short-Term Memory (LSTM) ağı kullanılmıştır. Çalışmanın değerlendirilmesi için NASA'nın C-MAPSS motor veri seti kullanılmıştır. Sonuçlar incelendiğinde SVR, derin CNN ve derin RNN gibi diğer yaygın kullanılan makine öğrenimi tekniklerine kıyasla derin LSTM ağının daha başarılı çalıştığı sonucuna varılmıştır. Gelecekteki çalışmalarda derin LSTM'nin diğer makine öğrenimi tekniklerine göre genel kullanılabilirliği araştırarak ve daha verimli bir ağ parametre ayarlama yöntemi geliştirilecektir[2].

2.2 Kestirimci Bakım İçin İstatiksel Yaklaşım

Francis ve diğerleri tarafından yapılan "ARIMA Model based Real Time Trend Analysis for Predictive Maintenance" isimli bu çalışmada gerçek zamanlı sensör değerlerini toplayarak ve üreticilere eş zamanlı olarak sunup trenin sağlıklı durumunun sürekli kontrol edilebilmesi ve böylece trenin çalışmasının iyileştirilmesi hedeflenmiştir. Hataların nerede ve ne zaman meydana geldiğinin belirlenmesine yardımcı olması için trend analizi yapılmıştır. Trend analizi ve tahmin işlemleri için istatistiksel bir analiz yöntemi olan ARIMA Modeli kullanılmıştır. Ancak ARIMA metodları makine öğrenimi metodlarıyla karşılaştırıldığında onlar kadar başarılı ve esnek bir yapıya sahip olmadıkları görülmüştür[3].

2.3 Yapay Zeka Teknikleri İle Sistemin Önleyici Bakımı

Lee ve diğerleri tarafından yapılan "Predictive Maintenance of Machine Tool System Using Artificial Intelligence Techniques Applied to Machine Condition Data" isimli bu çalışmada veriye dayalı bir modelleme yaklaşımı kullanılmıştır. Yapay zeka temelli algoritmaların tasarımı ve optimizasyonu sunularak iki kritik makine aracının izlenmesi sağlanmıştır. Bu iki sistem elemanı, kesici takım ve mili motorudur. Algoritmalar, takım aşınması ve bearing hatalarını izlemek için uygulanmıştır. Bir AI modelinin tanıma yeteneklerini arttırmak için büyük veri setinden anlamlı özellikler seçilmelidir. Deney verilerinden çıkarılan özellikler üzerinde uygulanan algoritmaların araçların bozulma durumunu değerlendirmek için etkili bir şekilde çalıştığı gözlemlenmiştir[4].

Bu bölümde projenin fizibilitesine odaklanılmaktadır. Projenin gerçekleştirilebilirliği hakkında bilgi sahibi olmak için fizibilite analizi yapılmıştır. İlk olarak iki ana başlıktan oluşan teknik fizibiliteye odaklanıp projemizi gerçekleştirdiğimiz uygulama geliştirme ortamlarına, programlama dili ve kütüphanelerine ve gerekli donanımlara değinilmiştir. Ardından proje sürecinin analizi ve gerekli adımlar ele alınmıştır. Son olarak da ekonomik ve yasal fizibilite incelemesi yapılmıştır.

3.1 Teknik Fizibilite

Teknik fizibilite yazılım ve donanım olmak üzere iki farklı grupta incelenir.

3.1.1 Yazılım Fizibilitesi

Bu projede veri seti üzerinde analiz çalışması yaptığımız için programlama dili olarak Python seçilmiş ve buna en uygun geliştirme ortamı olarak da daha kullanışlı bir arayüz sağlamasından dolayı Jupyter Notebook kullanılmıştır.

Python kullanma sebeplerimiz:

- Python dilinin veri bilimi ve makine öğrenimi için kütüphane çeşitliliği sunması
- Kolay öğrenilebilir ve uygulanabilir olması
- Açık kaynak kodlu olması

Jupyter Notebook kullanma sebeplerimiz:

- Kod bloklarının ayrı olarak çalıştırılmasına olanak sağlaması
- Görsel materyallerin kolayca görüntülenebilmesi

- Projede kullanmış olduğumuz iki farklı işletim sistemine de uygun olması

Veri bilimi ve makine öğrenimi için pandas, numpy, matplotlib ve sklearn kütüphanelerinden faydalanılmıştır.

3.1.2 Donanım Fizibilitesi

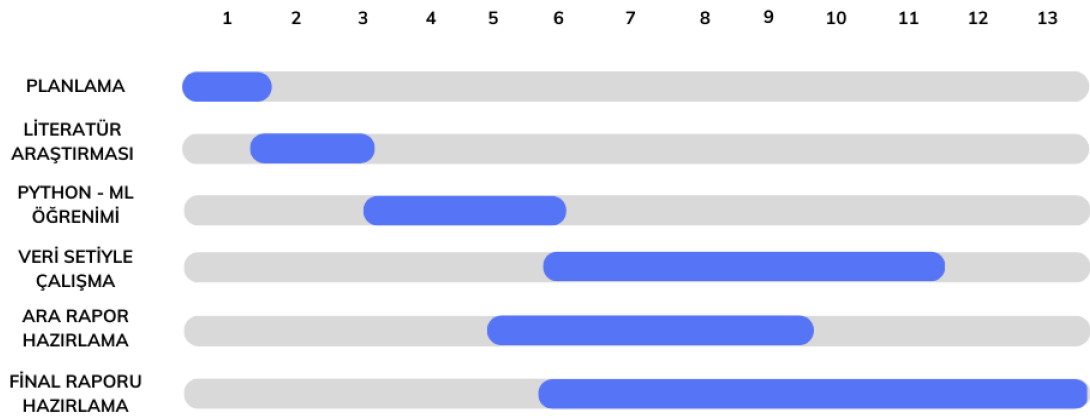
Projeyi gerçekleştirirken ek bir donanım kaynağına ihtiyaç duyulmamıştır. Projeyi gerçekleştirilen kişilere ait bilgisayarlar kullanılmıştır.

Tablo 3.1 Donanım Bilgileri

	1.Bilgisayar	2.Bilgisayar
RAM	16	16
Depolama	512 SSD	512 SSD
İşlemci	i7-10750H	i5
Ekran Kartı	NVIDIA GeForce GTX 1650 Ti	Intel Iris Plus Graphics 1536 MB
İşetim Sistemi	Windows	macOS

3.2 Zaman Fizibilitesi

13 haftalık projenin Gantt diyagramı Şekil 3.1’de gösterilmiştir.



Şekil 3.1 GANTT Çizelgesi

3.3 Yasal Fizibilite

Açık kaynaklı kütüphaneler kullanılmıştır. İlgili firmadan veri seti üzerinde çalışmak için gerekli izin alınmıştır. Kullandığımız kütüphanelere ait lisanslar belirtilmiştir.

NumPy - BSD lisansı

Pandas - BSD-3-Clause lisansı

Scikit-learn (sklearn) - BSD-3-Clause lisansı

Matplotlib - Matplotlib lisansı

3.4 Ekonomik Fizibilite

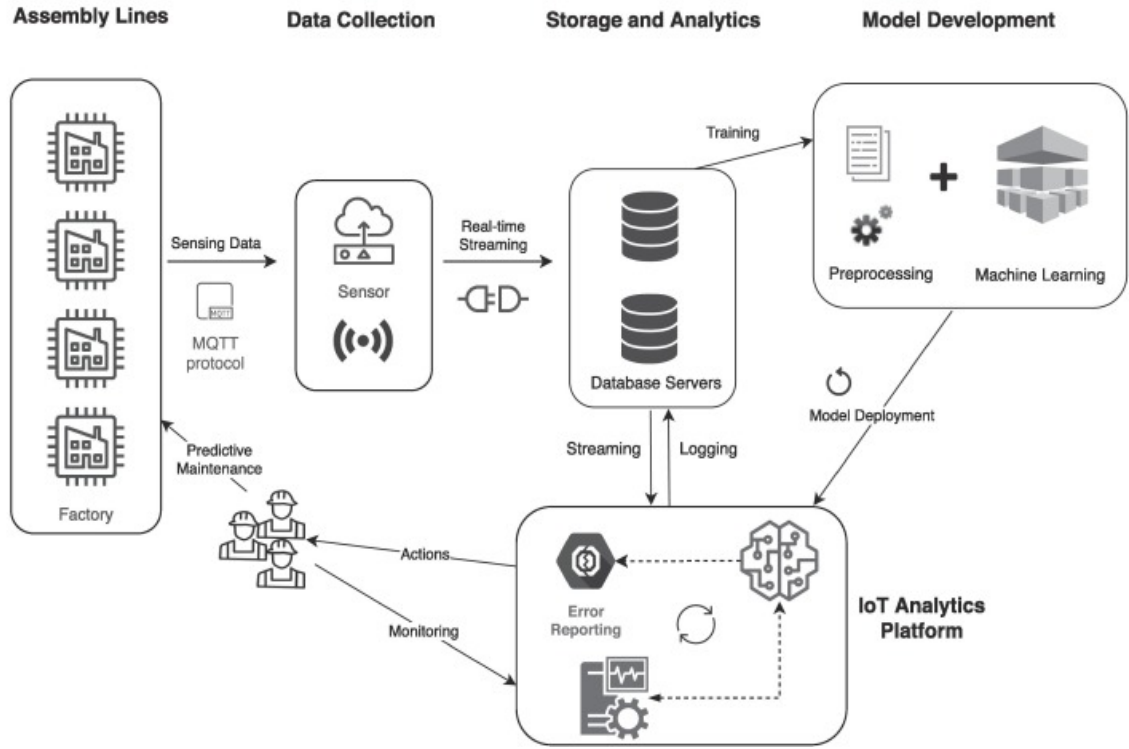
Bu bir araştırma çalışması olduğu için uygulamaya dönüşmeyecektir ve bu sebeple proje çıktısının ekonomik bir getirisi bulunmamaktadır. Ayrıca proje için de şahsi bilgisayarlarımızı kullandığımızdan herhangi bir gider söz konusu değildir.

4 SİSTEM ANALİZİ

Fabrikaya yerleştirilen sensörlerden sürekli ve anında veri akışı kullanılarak karar verme süreçlerinde kullanmak amacıyla ölçeklenebilir olması bir IoT uygulaması için en temel adımdır. IoT sisteminin harici sistemlerle entegre edilebilecek bir çerçeveye sahip olması gerekmektedir. Entegre veri akışı tahmin modeli ile birlikte bir PdM sistem arızalarını tespit edebilir, belirli kurallar uygulayarak alarm oluşturabilir, üretim sistemlerinde komutlar çalıştırabilir ve gerçek zamanlı uyarı mesajları gönderebilir. Projemizde bu yöntemlerle sensörlerden toplanan veriler üzerinde çalışma yapılmıştır.

4.1 Veri Seti

Kullanılan veri seti 26,431 satır ve 19 özellikten oluşmaktadır. Tüm özellikler sayısal değerlerden oluşmaktadır. Veri setinde önceden normalizasyon ve standardizasyon işlemleri gerçekleştirilmiştir. Bu nedenle veri setinde eksik, hatalı ve tutarsız değerler bulunmamaktadır. Projenin amacı bu özelliklerin 18 tanesini kullanarak bir tahmin sonucu oluşturmak ve bunu gerçek değerlerle karşılaştırıp analiz etmektir. Kestirimci bakım modelinin doğası gereği veri seti dengesiz bir yapıdadır ancak regresyon için kullanılacağından ve sınıflandırılma yapılmayacağından bu durum bir sorun teşkil etmemektedir. Dengeli hale getirmek için ekstra bir çalışma yapılmamıştır.

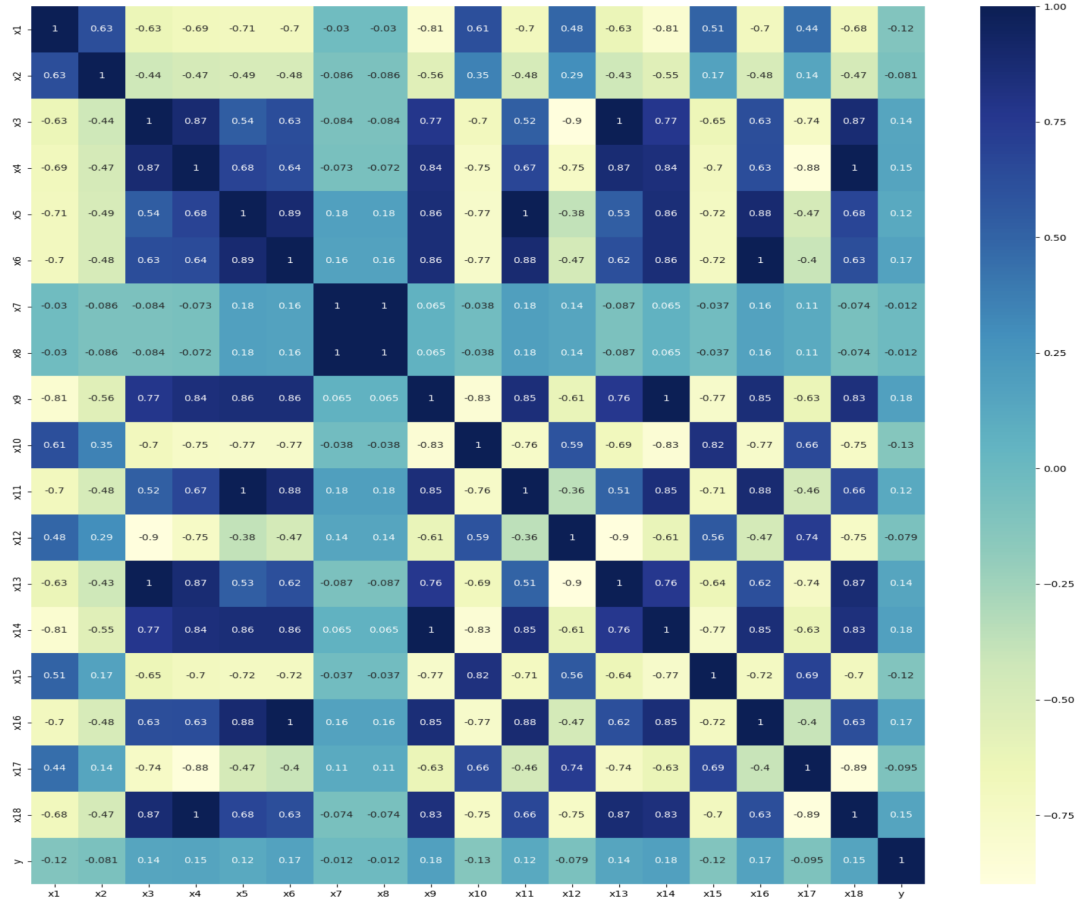


Şekil 4.1 Kestirimci Bakım Modeli Mimarisi

[1]

4.2 Korelasyon Matrisi

Veri keşfi bu çalışmanın en önemli parçalarından biridir. Korelasyon analizi farklı özellikler arasındaki ilişkinin yönünü (pozitif veya negatif) ve gücünü gösterir. Bu projede korelasyon analizi sensörlerden toplanan veriler ile arıza oluşmasından önce kalan ömür arasındaki ilişkileri incelemek için kullanılmış ve modelleme görevinde dikkate alınmıştır.



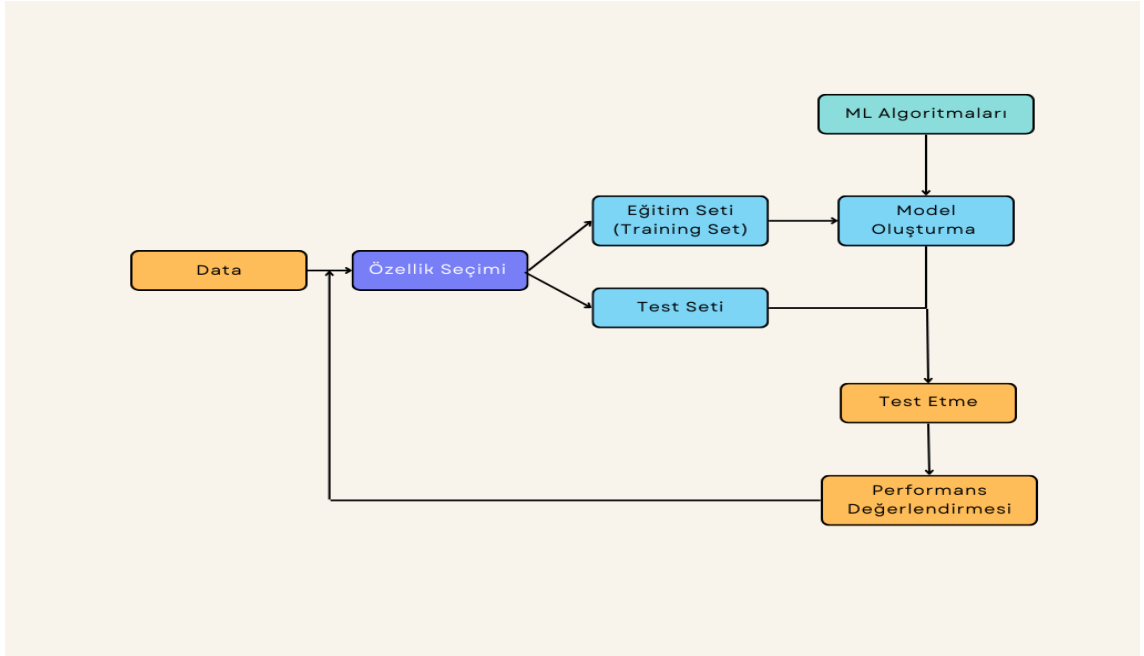
Şekil 4.2 Korelasyon Tablosu

Şekil 4.2’de veri setindeki tüm özellikler arasındaki ilişki korelasyon tablosu kullanılarak görselleştirilmiştir. Bazı özellikler arasında yüksek korelasyon olduğu gözlemlenmiştir. Örneğin korelasyon tablosuna bakıldığında x3, x4, x5, x6, x13 ve x14 özelliklerinin yüksek korelasyona sahip olduğu gözlemlenir. Yüksek korelasyona sahip özellikler daha küçük bir formda temsil edilebileceğinden özellik seçimi aşamasında boyut indirgeme için göz önünde bulundurulmuştur.

5

SİSTEM TASARIMI

Şekil 5.1’de proje geliştirme sürecinde sistemin nasıl ilerleyeceği şematik olarak gösterilmiştir. Bu bölümde özellik seçimine ve model oluşturma aşamalarına dair yöntemler tanıtılacaktır.



Şekil 5.1 Sistem İşleyiş Süreci

5.1 Özellik Seçimi

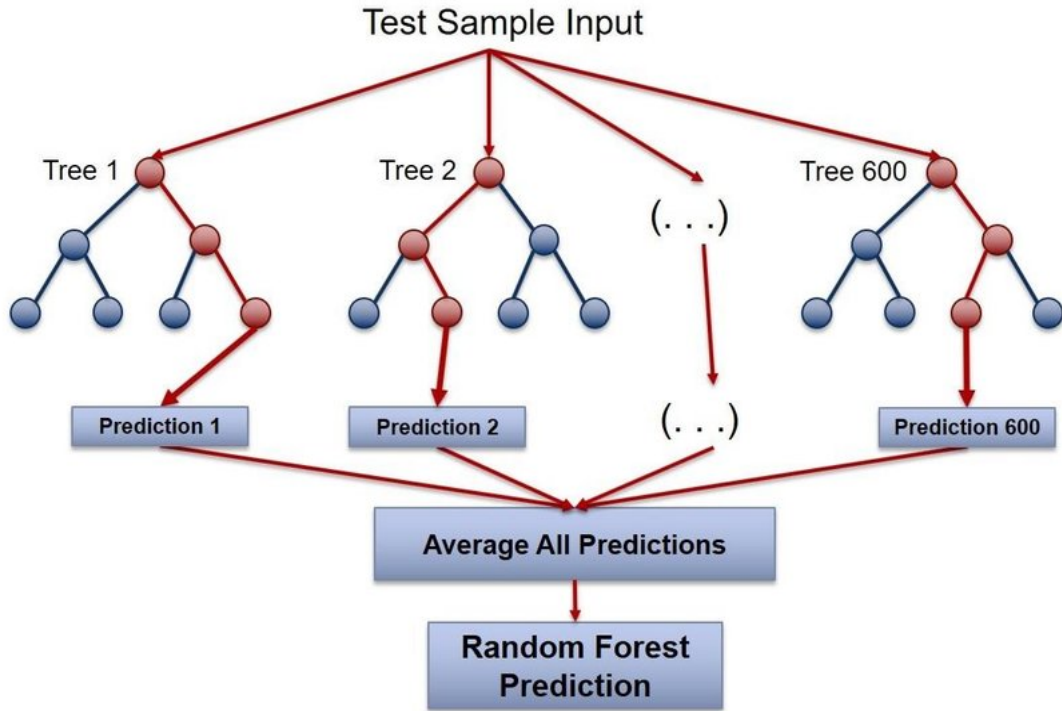
Özellik seçimi yüksek boyutlu büyük verilerin başarılı bir şekilde modellenmesi için makine öğrenimi modellerinde büyük bir role sahiptir. Aynı zamanda boyutluluk laneti (curse of dimensionality) olarak bilinen yüksek boyutlu problem, sıklıkla aşırı öğrenme (overfitting), çoklu bağımlılık (multicollinearity) gibi sorunlara neden olur. Verilerdeki olası ML modeli tahmin hatalarının azaltılması, boyut indirgeme yöntemlerinin dikkatli bir şekilde uygulanması ile gerçekleştirilir.

5.1.1 Temel Bileşen Analizi (PCA)

Temel bileşen analizi veri bilimi ve makine öğrenimi gibi çeşitli alanlarda çok değişkenli veri analizi için yaygın olarak kullanılan doğrusal bir boyut indirgeme tekniğidir. PCA ilişkili değişkenlerden oluşan bir kümenin, daha az sayıda temel bileşene dönüştürülmesi için kullanılır. Bunu yaparken de mümkün olan en çok sayıda varyansın korunmasını sağlar. PCA boyut azaltma işlemini gerçekleştirirken ilk olarak eğitim verilerinin kovaryans matrisini oluşturup özdeğer problemini hesaplar. Ardından en büyük özdeğerlere(eigenvalue) karşılık gelen özvektörleri(eigenvectors) temel faktörler olarak ele alır[5].

5.2 ML Algoritmaları

5.2.1 Random Forest

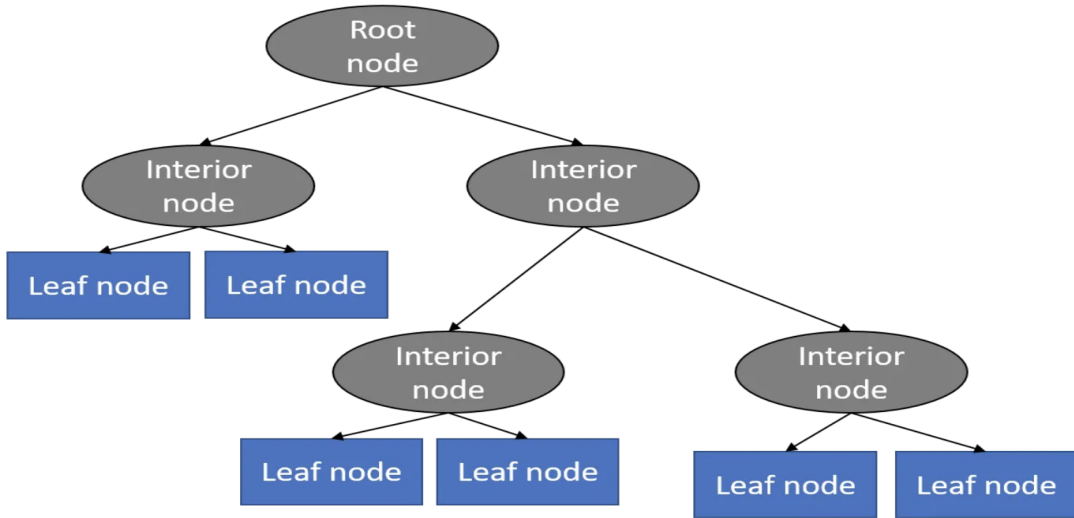


Şekil 5.2 Random Forest Algoritması

[6]

Random forest algoritması hem özellik seçimi hem de test/training veri seti ayrımı yapmak için kullanılan regresyon tabanlı bir algoritmadır. Denetimli öğrenme yöntemidir çünkü hem özellikler hem de tahmin etmek istediğimiz hedefler mevcuttur. Veri setinden rastgele örneklem alarak ve özellikleri rastgele seçilerek oluşturulmuş karar ağaçlarından oluşur. Random Forest'ın aşırı öğrenme yapması engellenerek daha iyi bir genelleme performansı sergilemesi rastgele örnekleme yöntemi ile sağlanır.

5.2.2 Karar Ağacı (Decision Tree)



Şekil 5.3 Karar Ağacı Yapısı

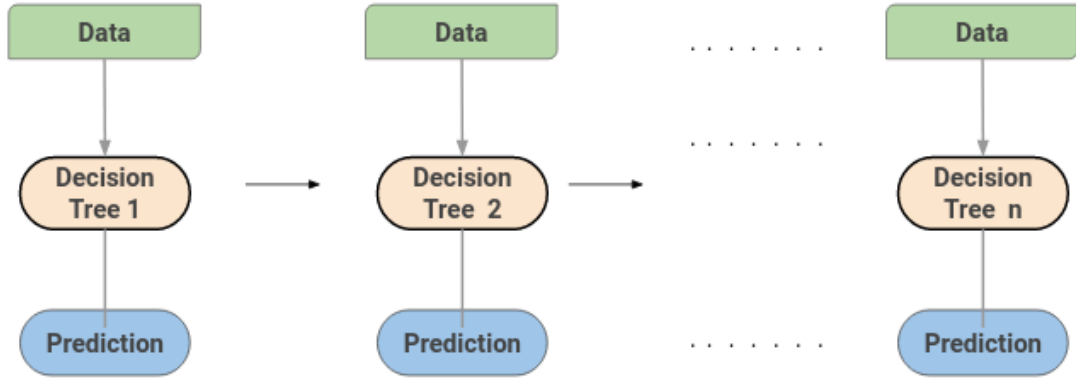
[7]

Karar Ağacı, denetimli öğrenim için en yaygın ve pratik yaklaşımlardan biridir. Bu yöntem, sınıflandırma problemlerinde pratikte daha yaygın olarak kullanılmasına rağmen, hem regresyon hem de sınıflandırma görevlerini çözmek için kullanılabilir.

Karar ağacı üç tür düğümden oluşan bir ağaç yapısıdır. Kök düğümü tüm örnekleri temsil eden ilk düğümdür ve daha fazla düğüme bölünebilir. İç düğümler girdi veri örüntülerine yönelik testler, yaprak düğümleri de bu örüntülerin kategori olarak alınabileceği bir ağaç yapısıdır. Bu testler doğru çıktıyı elde etmek için ağaç üzerinden filtrelendir ve girdi örüntüsüne göre çıktı üretilir[8].

İlk olarak en önemli özellik seçilir, seçilen özelliğe göre veri kümesini iki veya daha fazla alt kümeye bölünür. Her bir alt küme verilerini ayrı ayrı ele alarak karar ağacını oluşturur. Bu işlem alt kümelerde bir noktada tamamlanana kadar tekrarlanır.

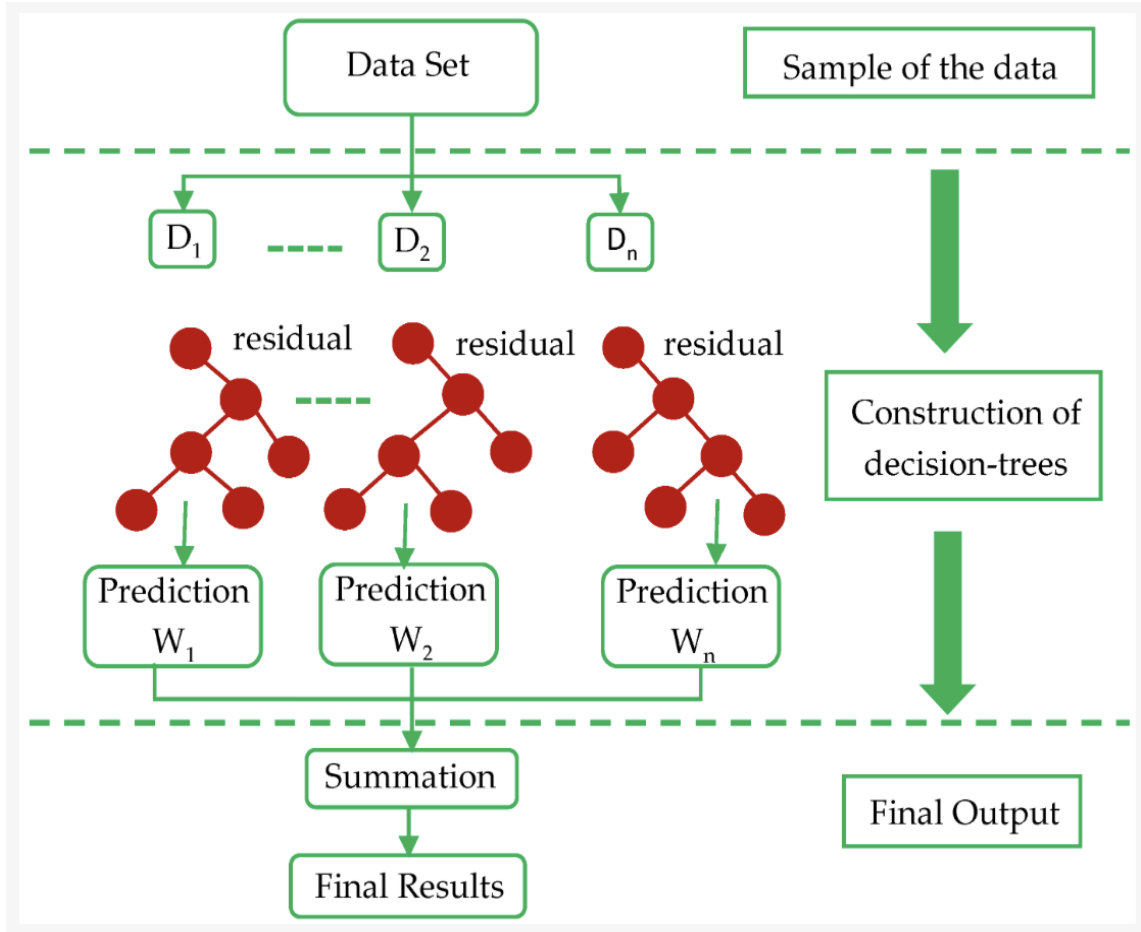
5.2.3 Gradyan Güçlendirme (Gradient Boosting)



Şekil 5.4 Gradyan Güçlendirme Algoritması
[9]

Gradyan güçlendirme birkaç zayıf öğreniciyi güçlü öğrenicilere dönüştüren güçlü bir arttırma (boosting) algoritmasıdır. Algoritma başlangıç adımı olarak maksimum derinliğe sahip bir karar ağacı ile model oluşturur. Ardından artık hata denilen tahmin sonuçları ve gerçek değerler arasındaki farkı hesaplar. Her yeni model bir önceki modelin tahminlerine göre hesaplanan artık hataların üzerinde eğitilmesi sonucu oluşur. Böylece artık hatanın minimuma inmesi amaçlanır.

5.2.4 XGBoost Algoritması



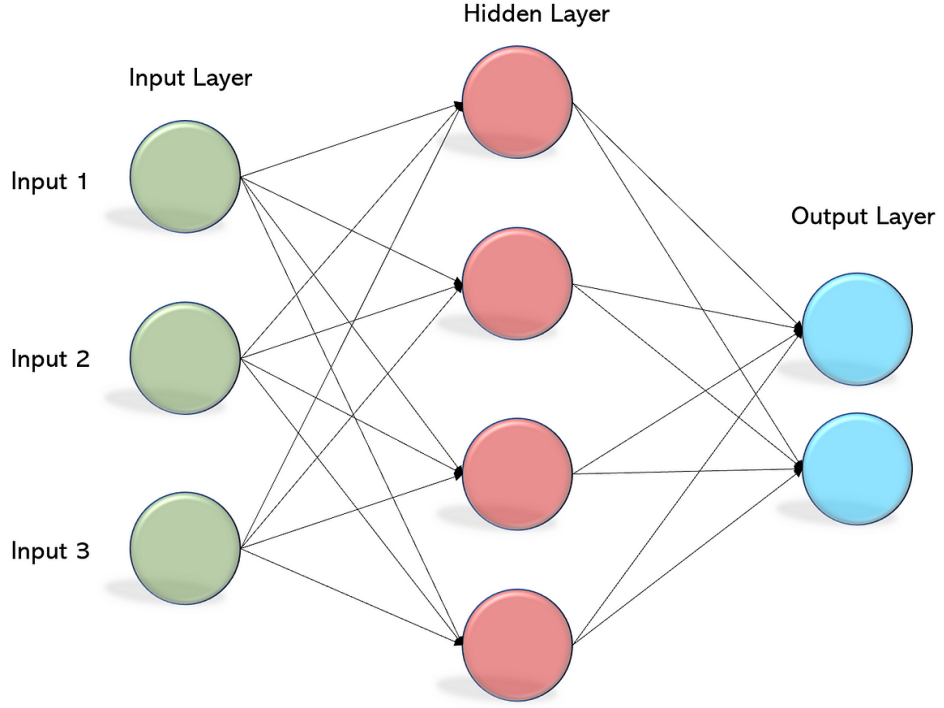
Şekil 5.5 XGBoost Algoritması

[10]

Son derece ölçeklenebilir olması için tasarlanmış gradyan arttırımına dayalı bir karar ağacı topluluğudur. Gradyan arttırımında olduğu gibi, XGBoost da amaç fonksiyonunun bir eklemeli genişlemesini, kayıp fonksiyonunu minimize ederek oluşturur. XGBoost yalnızca karar ağaçlarını temel sınıflandırıcı olarak kullanırken, ağaçların karmaşıklığını kontrol etmek için artık hatanın bir varyasyonu kullanılır. Bu algoritmada, karar ağaçları sıralı şekilde oluşturulur[11].

XGBoost'ta ağırlıkların önemli bir rolü vardır. Tüm bağımsız özelliklere ağırlıklar atanır ve ardından sonuçları tahmin eden karar ağacına aktarılır. Ağaç tarafından yanlış tahmin edilen bir değişken olduğunda bu değişkenin ağırlığı arttırılır ve ağırlığı arttırılmış değişkenler ikinci karar ağacına aktarılır. Bu tahmincilerin bir araya getirilmesiyle daha güçlü ve daha hassas bir model oluşturulması sağlanır. XGBoost, regresyon, sınıflandırma, sıralama ve kullanıcı tanımlı tahmin problemleri üzerinde çalışabilir.

5.2.5 MLP Algoritması



Şekil 5.6 MLP Algoritması

[12]

MLP ileri beslemeli yapıya sahip yapay bir sinir ağıdır. Gizli katman , giriş katmanı ve çıkış katmanı olarak bilinen 3 ana bölümden oluşur . Giriş katmanı , dışarıdan modele gelen verileri alıp aktivasyon fonksiyonu kullanarak verileri doğrudan gizli katmana iletir. Gizli katmana gelen veriler işlenerek daha karmaşık özelliklerin öğrenilmesi sağlanır. Gizli katmandan gelen veriler çıkış katmanında tahmin değerleri üretilmek için kullanılır. MLP öğrenme için back propagation algoritmasını kullanır. Sonuç çıktısı ve beklenen değer arasında bir sapma olduğunda bağlantı ağırlıklarını ayarlayarak bu işlemi gerçekleştirir[13]. Ancak MLP Algoritması'nın yavaş yakınsama, arama sırasında kararsızlık gibi dezavantajları bulunmaktadır. Eğitilmesi zor olduğu için büyük veri setleri üzerinde çalışması uzun sürmektedir.

5.3 Eğitim ve Test Setleri

Makine öğrenimini uygulanırken model için kullanılacak veriler iki farklı kümeye ayrılır. Modelin eğitimini gerçekleştirmek için ayrılan alt küme eğitim verisi adını alırken performans değerlendirmesi yapmak için oluşturulan alt küme test kümesi olarak adlandırılır. Tahminleri oluşturmak için modelin eğitilmesi amacıyla eğitim kümesi ML algoritmalarına aktarılır. Algoritmanın performansını değerlendirmek için

yapılan tahmin sonucuyla test kümesi kullanılarak modelin yeni veya görülmemiş veri kümeleriyle bir genelleme yapılır. Projemizde veri setini en yaygın kullanılan şekilde 80 eğitim ve 20 test seti olarak ayırmak için scikit-learn kütüphanesine ait train-test-split fonksiyonu kullanılmıştır. Yapılan ayırma işlemi sonucunda eğitim seti 21144 satırdan , test seti 5287 satırdan oluşmuştur.

6 UYGULAMA

Bu çalışma genel olarak makine öğrenimi ve veri analizine yönelik bir çalışmadır. Analizlerin nasıl gerçekleştiği ve sonuçlara dair bilgiler deneysel sonuçlar ve performan analizi bölümlerinde açıklanacağından dolayı bu bölüm boş bırakılmıştır.

7

Deneysel Sonular

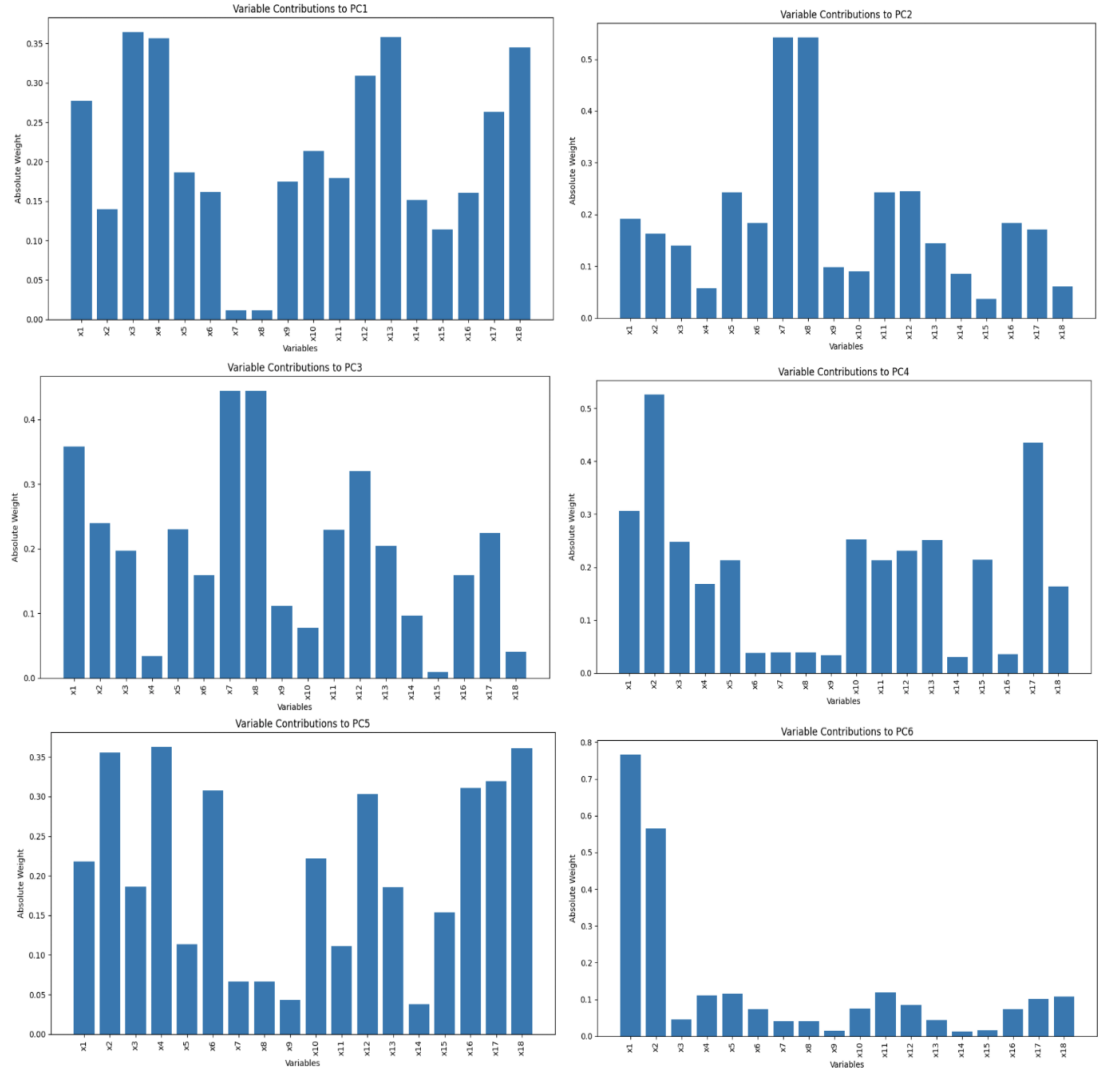
Bu blmde, makine ğrenmesi algoritmalarını eğitirken kullanılacak olan veri setindeki zellikleri belirlemek iin uygulanan ařamalar anlatılacak ve bu verilere uygulanan modellerin sonuları karřılařtırılacaktır. Kullanılan modellerdeki parametre deęerleri random search algoritması ile en yksek performans gsteren parametrelerin hangileri olduęu belirlenmiřtir.

Bu alıřmada RUL tahmini bir regresyon problemi olarak ele alındıęı iin regresyon problemlerine uygun makine ğrenme algoritmaları seilmiřtir.

7.1 Veri Kmesi İřlemleri

Yksek boyutlu uzayları fazla kayıp vermeden daha kk boyutlara indirgemek iin sıklıkla PCA algoritmasına bařvurulur. Biz de alıřmamızda elimizdeki yksek boyutlu veriyi indirgemek iin bu yntemi kullandık. PCA sonularına gre veri kmesindeki varyansın %95'i 6 temel bileřen ile temsil edilebilmektedir.

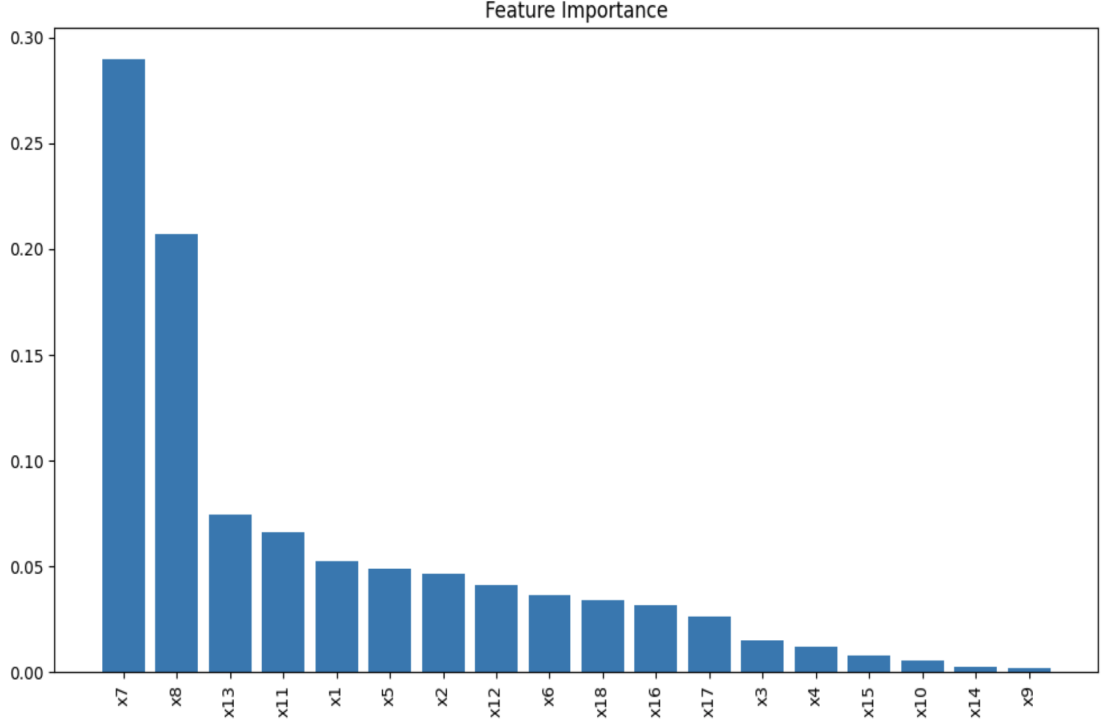
řekil 7.1'de veri seti ierisindeki zelliklerin PCA sonucu oluřan 6 farklı temel bileřene olan varyans katkılarının nasıl bir daęılım izledięi gsterilmektedir. Grafikler incelendięinde farklı temel bileřenler iin daha ok varyans katkısı olan zellikler deęiřiklik gsterse de genel olarak x1, x2, x7, x8 ve x12'nin daha baskın olduęu gzlemlenmiřtir.



Şekil 7.1 Özelliklerin PC'lere Varyans Katkıları

Ardından hangi özelliklerin daha yüksek öneme sahip olduğunu bulmak amacıyla genel olarak regresyon problemlerinde iyi bir tahmin performansı sergileyen ve kolay yorumlanabilen Random Forest Regressor algoritmasını tercih ettik.

Algoritma sonucu Şekil 7.2'de gösterilmektedir. Bu grafiğe göre tüm özellikler arasından en önemli ilk 9 özellik olan x1, x2, x5, x6, x7, x8, x11, x12, x13'ü seçtik. PCA sonuçları ile karşılaştırıldığında PCA sonuçlarında temel bileşenlere yüksek varyans katkısı gösteren özellikler de bu 9 özellik arasında yer almaktadır.



Şekil 7.2 Random Forest Algoritmasına Göre Özellik Önemleri

7.2 Random Forest İle Seçilen Özelliklere Uygulanan Modeller

Bu bölümde veri setinde belirlenen özellikler Random Forest, Gradient Boosting, XGBoosting, Decision Tree ve MLP Regressor modellerinin eğitilmesi için kullanılmıştır. Ve Random Search sonucu modeller için en iyi performans gösteren parametreler belirlenmiştir.

7.2.1 Model Sonuçları

Random Forest uygulanarak seçilen özellikler kullanılarak elde edilen sonuçlar aşağıdaki tabloda verilmiştir.

Tablo 7.1 Random Forest ile En İyi Sonuçlar

Algoritmalar	R^2	MSE	RMSE	MAE	MAPE
Random Forest	0.98	21552.27	146.80	53.20	3.26
XGBoost	0.98	23808.40	154.29	59.17	3.28
Gradient Boosting	0.97	31541.93	177.60	91.85	4.74
Decision Tree	0.72	336914.71	580.44	372.84	20.66
MLP Regressor	0.60	492209.00	701.57	531.32	27.35

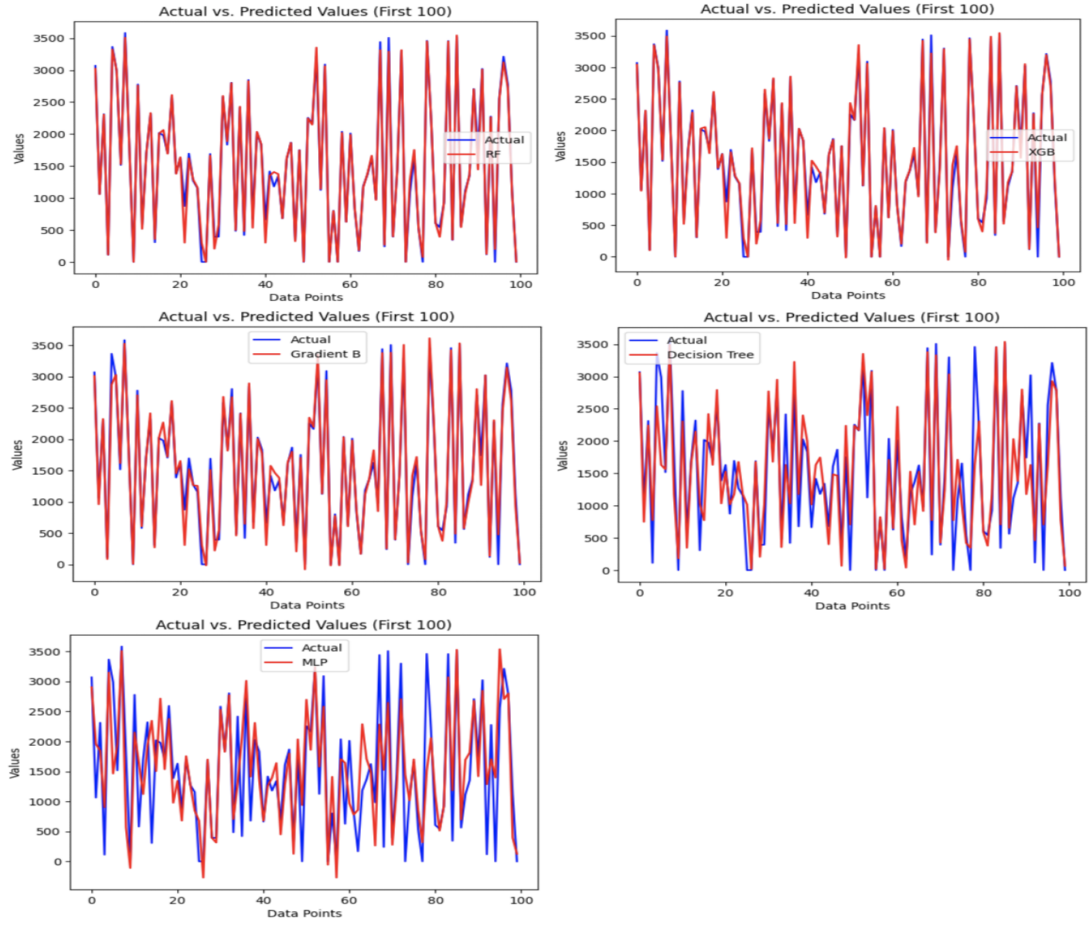
Tablo 7.1'e bakıldığında değerlendirmeler sonucunda 0.60 R^2 değeri ile Decision Tree ve 0.72 R^2 değeriyle de MLP algoritmalarının diğer algoritmalara göre daha başarısız bir performans sergilediği görülmektedir. En iyi sonuçların 0.98 R^2 değeri ile Random Forest ve XGBoost algoritmalarına ait olduğu görülmektedir. MAPE değerleri karşılaştırıldığında Random Forest algoritması küçük bir farkla da olsa daha iyi bir sonuç vermiştir.

Her bir model için yüksek performans gösteren sonuçları elde etmek için kullanılmış parametreler de Tablo 7.2'de yer almaktadır.

Tablo 7.2 En İyi Parametreler

Algoritma	Parametreler
Random Forest	n_estimators: 78, min_samples_split: 6, min_samples_leaf: 1, max_features: 'sqrt', max_depth: 30, 'bootstrap: False
XGBoost	subsample: 0.8, n_estimators: 500, max_depth: 15, learning_rate: 0.1, colsample_bytree: 0.8
Gradient Boosting	subsample: 1.0, n_estimators: 200, min_samples_split: 4, min_samples_leaf: 1, max_depth: 7, learning_rate: 0.1
Decision Tree	max_depth: 9, min_samples_leaf: 4, min_samples_split: 10, splitter: 'best'
MLP Regressor	hidden_layer_sizes=100, activation='relu', solver='lbfgs', random_state=35, max_iter=5000

Uygulanan modeller sonucunda ortaya çıkan tahmin sonuçlarının gerçek değerler ile karşılaştırılması Şekil 7.3'te görselleştirilmiştir. MLP Regressor ve Decision Tree grafiklerindeki tahmin sonuçları ve gerçek değerler arasındaki farkların daha büyük olması Tablo 7.1'e bakıldığında beklenen bir durumdur. Diğer model sonuçlarının da tablodaki değerlere uyumlu olduğu görülmektedir.



Şekil 7.3 Algoritma Sonuç Karşılaştırmaları

8 Performans Analizi

Makine öğrenimi algoritmalarında eğitilen modelin veri seti üzerindeki performansını ölçmek için çeşitli değerlendirme metrikleri bulunur. Regresyon problemlerinde bu amaçla yaygın olarak ortalama mutlak hata (MAE), ortalama kare hata (MSE), kök ortalama kare hata (RMSE), R-kare (R^2) ve ortalama mutlak yüzde hata (MAPE) metrikleri kullanılır. Bu çalışma da bir regresyon problemini ele alındığı için bu metrikler kullanılmıştır.

8.0.1 MAE

Elde edilen tahmin sonuçları ile gerçek sonuçlar arasındaki ortalama büyüklüğü mutlak değer olarak ölçer. 8.1'deki formülde \hat{y}_i tahmin sonucu oluşan RUL değerini ifade ederken y_i ise gerçek sonucu ifade eder. MAE değerinin azalması modelin daha iyi bir performans gösterdiğinin ifadesidir.

$$MAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n} \quad (8.1)$$

8.0.2 R^2

R-kare bir modelin toplam varyansının ne kadar açıklanabildiğini gösteren istatistiksel bir ölçüdür. Modelin verilere ne kadar iyi bir şekilde uyum sağlayabildiğini gösterir. 0 ile 1 arasında bir değer alır. R-kare değerinin 1'e yaklaşması tahmin sonuçlarının gerçek sonuçlara yakın olduğunu yani modelin veriye uygun olduğunu gösterir. Formüldeki \bar{y} ortalama gerçek değeri ifade eder.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (8.2)$$

8.0.3 MAPE

Elde edilen tahmin sonuçları ile gerçek sonuçlar arasındaki ortalama mutlak değeri (MAE) yüzdelik olarak ölçer.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100 \quad (8.3)$$

8.0.4 MSE

Ortalama kare hatası tahmin edilen sonuçların gerçek sonuçlardan farkının karesini hesaplayarak mutlak bir sayı döndürür ve aykırı değerlere karşı daha duyarlıdır. Sadece mükemmel bir model sonucunda 0 değerini alabilir onun dışında pozitif bir değere sahiptir. 0 değerini alması pratikte karşılaşılan bir durum değildir.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (8.4)$$

8.0.5 RMSE

MSE sonucunun karekökü alınarak hesaplanır ve bu şekilde modelin yorumlanmasını daha kolay bir hale getirilmesini sağlar. MSE değerinin bazı durumlarda kolayca karşılaştırılamayacak kadar büyük olması sebebiyle daha çok RMSE tercih edilir.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (8.5)$$

Bu projede üretim hatları için makine öğrenimi temelli kestirimci bakım modeli geliştirmek hedeflenmiştir. Proje sürecinde gerekli geliştirmeleri ve analizleri yapmak için özel bir firmadan alınan sensör verileri kullanılmıştır.

Başlangıçta veri seti analiz edilip herhangi bir kayıp değer veya aykırı değer olmadığı tespit edilmiştir. Ayrıca veri setine normalizasyon işlemleri önceden yapıldığı için bu amaçla başka bir işlem yapılmasına gerek duyulmamıştır. Sonrasında veri setinin korelasyon matrisi oluşturulmuş ve hangi özelliklerin birbiriyle yüksek korelasyona sahip olduğu incelenmiştir.

Analizler yapıldıktan sonra veri seti boyutunun indirgenmesi için yöntemler kullanılmıştır. Öncelikle temel bileşen analizi ile %95 varyansın 6 temel bileşen ile ifade edilebileceği belirlenmiş ve hangi özelliklerin bu varyansa daha çok katkı sağladığına dair bilgi sahibi olunmuştur. Bu işlemi takiben özellik seçimi için Random Forest algoritması uygulanmıştır. Model sonucuna göre belirlenen en önemli 9 özellik seçilmiştir.

Seçilen özellikler kullanılarak veri seti eğitim ve test olmak üzere iki kümeye ayrılmıştır. Sonrasında 5 farklı makine öğrenimi algoritması verileri eğitmek için uygulanmıştır. Modellerde kullanılacak parametrelerin en optimum değerlerini belirlemek için Random Search ve Grid Search algoritmaları değerlendirilmiştir. Grid Search algoritmasının çalışmasının çok uzun sürmesi sebebiyle Random Search algoritmasının uygulanmasına karar verilmiştir.

Uygulanan modeller çıktı olarak arıza öncesi kalan tahmini kullanılabilir zamanı vermektedir. Bu sonuçların performanslarını değerlendirmek ve karşılaştırmak için regresyon problemlerinde kullanılan R^2 , MAE, MAPE, MSE, RMSE metrikleri kullanılmıştır. Değerlendirme sonuçlarına bakıldığında MLP ve Decision Tree algoritmalarının başarı oranı oldukça düşükken Random Forest ve XGBoost algoritmalarının daha yüksek bir başarı elde ettiği gözlemlenmiştir.

Sonuç olarak arıza öncesi kalan tahmini kullanılabilir zamanı belirlemek için 0.98 R^2 başarısı ile Random Forest ve XGBoost algoritmalarının daha iyi olduğu belirlenmiştir. Bu çalışmada yalnızca tek bir firmaya ait veriler kullanılmış, geliştirilen modelin performansının daha iyi değerlendirilebilmesi için farklı veri setleri üzerinde çalışmalar yapılabilir. Geliştirilen model, kullanılan veri seti üzerinde oldukça iyi bir performans sergilese de daha farklı algoritmalar da eğitilerek sonuçlar karşılaştırılabilir.

- [1] S. Ayvaz and K. Alpay, "Predictive maintenance system for production lines in manufacturing: A machine learning approach using iot data in real-time," *Expert Systems with Applications*, vol. 173, p. 114598, 2021, ISSN: 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2021.114598>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417421000397>.
- [2] J. Zhang, P. Wang, R. Yan, and R. X. Gao, "Deep learning for improved system remaining life prediction," *Procedia CIRP*, vol. 72, pp. 1033–1038, 2018, 51st CIRP Conference on Manufacturing Systems, ISSN: 2212-8271. DOI: <https://doi.org/10.1016/j.procir.2018.03.262>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2212827118304335>.
- [3] F. Francis and M. Mohan, "Arima model based real time trend analysis for predictive maintenance," in *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2019, pp. 735–739. DOI: 10.1109/ICECA.2019.8822191.
- [4] W. J. Lee, H. Wu, H. Yun, H. Kim, M. B. Jun, and J. W. Sutherland, "Predictive maintenance of machine tool systems using artificial intelligence techniques applied to machine condition data," *Procedia CIRP*, vol. 80, pp. 506–511, 2019, 26th CIRP Conference on Life Cycle Engineering (LCE) Purdue University, West Lafayette, IN, USA May 7-9, 2019, ISSN: 2212-8271. DOI: <https://doi.org/10.1016/j.procir.2018.12.019>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2212827118312988>.
- [5] C. Yang, L. Wang, and J. Feng, "On feature extraction via kernels," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 2, pp. 553–557, 2008. DOI: 10.1109/TSMCB.2007.913604.
- [6] "Evaluation and comparison of machine learning techniques for rapid qsts simulations - scientific figure on researchgate." Görsel. (2023), [Online]. Available: https://www.researchgate.net/figure/Random-Forest-visualization_fig11_326560291 (visited on 04/27/2023).
- [7] "Decision trees." Görsel. (2023), [Online]. Available: <https://python-course.eu/machine-learning/decision-trees-in-python.php> (visited on 04/27/2023).
- [8] A. Navada, A. N. Ansari, S. Patil, and B. A. Sonkamble, "Overview of use of decision tree algorithms in machine learning," in *2011 IEEE Control and System Graduate Research Colloquium*, 2011, pp. 37–42. DOI: 10.1109/ICSGRC.2011.5991826.

- [9] “Gradient boosted trees for regression.” Görsel. (2023), [Online]. Available: https://www.analyticsvidhya.com/blog/2016/02/complete-guide-parameter-tuning-gradient-boosting-gbm-python/?utm_source=blog&utm_medium=4-boosting-algorithms-machine-learning (visited on 04/27/2023).
- [10] “Structure of xgboost algorithm.” Görsel. (2023), [Online]. Available: <https://www.mdpi.com/2076-3417/12/4/2126> (visited on 04/27/2023).
- [11] C. Bentéjac, A. Csörgő, and G. Martinez-Muñoz, “A comparative analysis of gradient boosting algorithms,” *Artificial Intelligence Review*, vol. 54, pp. 1937–1967, 2021.
- [12] “Mlp algorithm.” Görsel. (2023), [Online]. Available: <https://becominghuman.ai/multi-layer-perceptron-mlp-models-on-real-world-banking-data-f6dd3d7e998f> (visited on 04/27/2023).
- [13] I. R. Widiyari, L. E. Nugroho, and Widyawan, “Deep learning multilayer perceptron (mlp) for flood prediction model using wireless sensor network based hydrology time series data mining,” in *2017 International Conference on Innovative and Creative Information Technology (ICITech)*, 2017, pp. 1–5. DOI: 10.1109/INNOCIT.2017.8319150.

BİRİNCİ ÜYE

İsim-Soyisim: Zeynep ACAR
Doğum Tarihi ve Yeri: 31.03.2001, İzmir
E-mail: zeynep.acar1@std.yildiz.edu.tr
Telefon: 0536 920 29 49
Staj Tecrübeleri: UNILOGISTIX Şirketi Yazılım Departmanı

İKİNCİ ÜYE

İsim-Soyisim: Betül ÇELİK
Doğum Tarihi ve Yeri: 13.02.2001, Denizli
E-mail: betul.celik2@std.yildiz.edu.tr
Telefon: 0539 240 65 84
Staj Tecrübeleri: UNILOGISTIX Şirketi Yazılım Departmanı

Proje Sistem Bilgileri

Sistem ve Yazılım: Windows/MacOS İşletim Sistemi, Python
Gerekli RAM: 4GB
Gerekli Disk: 5GB