



**T.C.
HALIÇ ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
MÜHENDİSLİK ANABİLİM/ANASANAT DALI
BİLGİSAYAR MÜHENDİSLİĞİ TEZLİ YÜKSEK LİSANS PROGRAMI**

**TÜRKİYE'DE COVID-19 HASTALIĞININ
MAKİNE ÖĞRENMESİ YÖNTEMİYLE ANALİZİ VE TAHMİNİ**

YÜKSEK LİSANS TEZİ

**Hazırlayan
ONUR KIZILARSLAN**

**Tez Danışmanı
Dr. Öğr. Üyesi FİGEN ÖZEN**

**İSTANBUL
Haziran 2022**



**T.C.
HALIÇ ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
MÜHENDİSLİK ANABİLİM/ANASANAT DALI
BİLGİSAYAR MÜHENDİSLİĞİ TEZLİ YÜKSEK LİSANS PROGRAMI**

**TÜRKİYE'DE COVID-19 HASTALIĞININ
MAKİNE ÖĞRENMESİ YÖNTEMİYLE ANALİZİ VE TAHMİNİ**

YÜKSEK LİSANS TEZİ

**Hazırlayan
ONUR KIZILARSLAN**

**Tez Danışmanı
Dr. Öğr. Üyesi FİGEN ÖZEN**

**İSTANBUL
Haziran 2022**

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ MÜDÜRLÜĞÜ'NE

Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans Programı Öğrencisi Onur Kızıllarslan tarafından hazırlanan ***“Türkiye’de Covid-19 Hastalığının Makine Öğrenmesi Yöntemleriyle Analizi ve Tahmini”*** konulu çalışması jürimizce Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Savunma Tarihi: 29/06/2022

Jüri Üyesinin Ünvanı, Adı, Soyadı ve Kurumu:

İmzası

Jüri Üyesi : Dr. Öğr. Üye. Figen Özen
Haliç Üniversitesi

Jüri Üyesi : Dr. Öğr. Üye. Sait Erkovan
Haliç Üniversitesi

Jüri Üyesi : Dr. Öğr. Üye. Zeynep Turgut Akgün
İstanbul Medeniyet Üniversitesi

Bu tez yukarıdaki jüri üyeleri tarafından uygun görülmüş ve Enstitü Yönetim Kurulu’nun kararıyla kabul edilmiştir.

(Enstitü Müdürünün Ünvanı, Adı, Soyadı)
Müdür

COVID19_TEZ_ONUR

ORJİNALLİK RAPORU

% **12**

BENZERLİK ENDEKSİ

% **11**

İNTERNET KAYNAKLARI

% **2**

YAYINLAR

% **6**

ÖĞRENCİ ÖDEVLERİ

BİRİNCİL KAYNAKLAR

1

Submitted to The Scientific & Technological
Research Council of Turkey (TUBITAK)
Öğrenci Ödevi

% **2**

2

Turkiye.Ai
İnternet Kaynağı

% **1**

3

akademi40.org
İnternet Kaynağı

% **1**

4

Submitted to Fırat Üniversitesi
Öğrenci Ödevi

% **1**

5

acikbilim.yok.gov.tr
İnternet Kaynağı

% **1**

6

www.safakkayikci.com
İnternet Kaynağı

% **1**

7

Submitted to Kocaeli Üniversitesi
Öğrenci Ödevi

<% **1**

8

acikerisim.sakarya.edu.tr
İnternet Kaynağı

<% **1**

9

Submitted to Istanbul Aydin University
Öğrenci Ödevi

<% **1**

TEZ ETİK BEYANI

Yüksek Lisans Tezi olarak sunduğum “Türkiye’de Covid-19 Hastalığının Makine Öğrenmesi Yöntemiyle Anazlizi Ve Tahmini” başlıklı bu çalışmayı baştan sona kadar danışmanım Figen Özen’in sorumluluğunda tamamladığımı, verileri/örnekleri kendim topladığımı, deneyleri/analizleri ilgili laboratuvarlarda yaptığımı/yaptırdığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim.

Onur KIZILARSLAN

ÖNSÖZ

Bu çalışma boyunca yardım ve katkılarıyla bana destek olan tez hocam Sayın Dr. Öğretim Üyesi Figen ÖZEN'e, bu süreçte hep yanımda olarak çalışmamı destekleyen ve bugüne kadar verdikleri her türlü yardımları ile bu tezi yazmamda en büyük destekçim olan aileme teşekkürlerimi borç bilirim.

Haziran, 2022

Onur KIZILARSLAN

İÇİNDEKİLER

Sayfa No.

TEZ ETİK BEYANI	i
ÖNSÖZ.....	ii
İÇİNDEKİLER	iii
KISALTMALAR	vi
SEMBOLLER	viii
TABLO LİSTESİ	ix
ŞEKİL LİSTESİ.....	x
ÖZET.....	xii
ABSTRACT	xiii
1. GİRİŞ.....	1
2. GENEL BİLGİLER	3
2.1. Salgın Hastalıklar.....	3
2.2. Tarihteki Salgın Hastalıklar	4
2.3. Koronavirüsler	4
2.4. Covid-19 Nasıl Bulaşır?	4
2.5. Kimler Daha Fazla Risk Altında?.....	5
2.6. Hastalıktan En Çok Etkilenen Kişiler	5
3. GEREÇ VE YÖNTEM	6
3.1. Yapay Zekâ Nedir?	6
3.2. Makine Öğrenmesi.....	7
3.3. Derin Öğrenme	8
3.3.1. Derin Öğrenmenin Kullanıldığı Alanlar	8
3.4. Yapay Sinir Ağları	8
3.4.1 Yapay Sinir Ağlarında Öğrenme.....	13
3.4.2 Yapay Sinir Ağlarında Öğrenme Kuralları	13
3.5. Yapay Sinir Ağı Mimarileri.....	14
3.5.1. İleri Yayılım Algoritmaları	17

3.5.2. Geri Yayılım Algoritmaları.....	17
3.6. Yapay Zekâ.....	18
3.6.1 Yapay Zekânın Tarihsel Süreçleri.....	18
3.7. Veri	22
3.7.1. Veri Saklama Yönetimi.....	22
3.7.2. Veri Betimleme	22
3.7.3. Veri Görselleştirme	23
3.7.4. Veri Madenciliği Nedir?	23
3.7.5. Veri Madenciliğinin Kullanım ve Uygulama Alanları	24
3.7.6. Veri Madenciliği Süreçleri.....	25
3.7.6.1 Problemin Tanımlanması	26
3.7.6.2. Verilerin Hazırlanması.....	26
3.7.6.3 Toplama (Collection):	26
3.7.6.4. Değer Biçme (Assesment):	27
3.7.6.5. Birleştirme ve Temizleme (Consolidation and Cleaning):	27
3.7.6.6. Seçim (Selection):	27
3.7.6.7. Dönüştürme (Transformation):	27
3.7.7. Veri Madenciliği Modelleri	27
3.7.8. Veri Analizi	29
3.7.9. Veri Madenciliğinde Karşılaşılan Sorunlar	30
3.8 Yapay Zeka Çalışmalarında Kullanılan Yazılımlar	30
3.8.1. Python Programlama Dili	32
3.8.2. Python Kütüphaneleri	33
3.8.2.1. NumPy Kütüphanesi	33
3.8.2.2. Pandas Kütüphanesi	34
3.8.2.3. Matplotlib Kütüphanesi.....	34
3.8.2.4. SciPy Kütüphanesi	34
3.8.2.5. Tensorflow Kütüphanesi	35
3.8.2.6. Keras Kütüphanesi	35
3.8.2.7. Scikit-Learn Kütüphanesi	35
3.8.2.8. Bokeh Kütüphanesi.....	35
3.8.2.9. Jupyter Notebook	35
3.8.2.10. GeoPandas.....	36
3.9. Evrişimsel Sinir Ağları Kullanılarak Derin Bilgisayarlı Görü.....	36

3.10. Evrişimsel Katmanlar	36
3.11. Biriktirme Katmanlar.....	37
3.12. CNN Mimarileri.....	37
3.13. X-Ray Görüntülerinden Tahmininde Bulunan Evrişimsel Sinir Ağı	38
3.14. Epidemiyolojide Kompartımanlı Modeller.....	43
3.15. SIR ve SEIR Komartıman Modelleri.....	43
3.16. SIR ve SEIR Komartıman Modelleri Arasındaki Benzerlikler	44
3.17. SIR ve SEIR Komartıman Modelleri Arasındaki Benzerlikler	44
3.18. Yayıp Sinir Ağı Modelinin Geliştirilmesi.....	44
3.19. Tek Katmanlı Ağır Hasta Sayısı	45
3.20. Tek Katmanlı Toplam Vaka Sayısı.....	54
3.21. Tek Katmanlı Toplam İyileşen Sayısı.....	56
3.22. Tek Katmanlı Toplam Vefat Sayısı	58
3.23. Üç Katmanlı Ağır Hasta Sayısı.....	59
3.24. Üç Katmanlı Toplam İyileşen Sayısı	61
3.25. Üç Katmanlı Toplam Test Sayısı.....	63
3.26. Üç Katmanlı Toplam Vaka Sayısı	64
4. BULGULAR.....	67
5. TARTIŞMA	68
6. SONUÇLAR	70
7. ÖNERİLER.....	71
KAYNAKLAR	72
ÖZGEÇMİŞ.....	74

KISALTMALAR

CNN	: Evrişimsel Sinir Ağları
NLP	: Doğal Dil İşleme
NLU	: Doğal Dil Anlama
RNN	: Yinelenen (Tekrarlayan) Sinir Ağları
SVM	: Destek Vektör Makineleri
RL	: Pekiştirmeli Öğrenme
ML	: Makine Öğrenmesi
DBN	: Derin İnanç Ağları
RBM	: Kısıtlanmış Boltzmann Makineleri
MAE	: Ortalama Mutlak Hata
MAD	: Ortalama Mutlak Sapma
MSE	: Ortalama Kare Hata
RMSE	: Kök Ortalama Kare Hatası
MAPE	: Ortalama Mutlak Yüzde Hata
sMAPE	: Simetrik Ortalama Mutlak Yüzde Hatası
MASE	: Ortalama Mutlak Ölçekli Hata
MNB	: Ortalama Normalleştirilmiş Bias
MdAE	: Ortanca Mutlak Hata
MaxAE	: Maksimum Mutlak Hata
MARE	: Ortalama Mutlak Bağlı Hata
MdAPE	: Ortanca Mutlak Yüzde Hata
RAE	: Göreli Mutlak Hata
MRAE	: Ortalama Göreli Mutlak Hata
GMAE	: Geometrik Ortalama Mutlak Hata
SAD	: Mutlak Farkların Toplamı
GMRAE	: Geometrik Ortalama Bağlı Mutlak Hata
MdRAE	: Medyan Göreli Mutlak Hata
sMdAPE	: Simetrik Medyan Mutlak Yüzde Hatası

SSE	: Kare Hata Toplamı
RSE	: Göreli Kare Hatası
RRSE	: Kök Göreli Kare Hatası
GRMSE	: Geometrik Kök Ortalama Kare Hatası
MSPE	: Ortalama Kare Yüzde Hatası
MdSPE	: Medyan Kare Yüzde Hatası
RMSPE	: Kök Ortalama Kare Yüzde Hatası
RMdSPE	: Kök Medyan Kare Yüzde Hatası



SEMBOLLER

x_n : Kullanıcı tarafından sisteme girilen verilerdir.

w_n : Nörona gelen giriş bilgisinin, nöronun çıkışına hangi oranda etki edeceğini ifade etmektedir.

$h(.)$: Aktivasyon fonksiyonudur.



TABLO LİSTESİ

Sayfa No.

Tablo 3.1. Yapay Sinir Ağlarında Kullanılan Toplam Fonksiyonları	11
Tablo 3.2. Yapay Sinir Ağlarında Kullanılan Aktivasyon Fonksiyonları	12
Tablo 3.3. Toplam Test Sayısı, Toplam Vaka Sayısı, Toplam Vefat Sayısı ve Toplam İyileşen Hasta Sayısı Verileri	51
Tablo 3.4. Ağır Hasta Sayısı Verileri	52
Tablo 3.5. MAE – MSE – RMSE – MAPE Verileri Tablosu	54
Tablo 3.6. MAE – MSE – RMSE – MAPE Verileri Tablosu	55
Tablo 3.7. MAE – MSE – RMSE – MAPE Verileri Tablosu	57
Tablo 3.8. MAE – MSE – RMSE – MAPE Verileri Tablosu	59
Tablo 3.9. MAE – MSE – RMSE – MAPE Verileri Tablosu	61
Tablo 3.10. MAE – MSE – RMSE – MAPE Verileri Tablosu	62
Tablo 3.11. MAE – MSE – RMSE – MAPE Verileri Tablosu	64
Tablo 3.12. MAE – MSE – RMSE – MAPE Verileri Tablosu	66

ŞEKİL LİSTESİ

Sayfa No.

Şekil 2.1. LSTM hücresi	3
Şekil 3.1. Bir Sinir Hücresinin Biyolojik Gösterimi.....	9
Şekil 3.2. Bir Sinir Hücresinin Matematiksel Modeli.....	10
Şekil 3.3. Basit bir yapay sinir ağı modeli	14
Şekil 3.4. Üç katmanlı model.....	15
Şekil 3.5. Aktivasyon fonksiyonu kullanmadan gizli katman kullanarak modelin çıktısını hesaplama	16
Şekil 3.6. Veri Madenciliği Süreçleri	26
Şekil 3.7. Veri Madenciliği Modelleri	28
Şekil 3.8. Covid19 Pandemisinden Etkilenen Hastaların Akciğer Röntgenleri.....	39
Şekil 3.9. Eğitim ve Doğrulama Kayıplarının Grafiği.....	41
Şekil 3.10. Eğitim ve Doğruluk Grafiği.....	41
Şekil 3.11. Eğitim ve Doğrulama Başarımı Grafiği.....	42
Şekil 3.12. Eğitim Kaybı ve Doğrulama Kaybı Grafiği.....	42
Şekil 3.13. Corona Virüse yakalanmış ve yakalanmamış iki farklı kişinin akciğer röntgen görüntüsü.....	43
Şekil 3.14. Toplam Test Sayısı Grafiği.....	45
Şekil 3.15. Toplam Vaka Sayısı Grafiği	46
Şekil 3.16. Toplam Vefat Sayısı Grafiği.....	46
Şekil 3.17. Hastalarda Zature Oranı Grafiği	47
Şekil 3.18. Ağır Hasta Sayısı Grafiği.....	47
Şekil 3.19. Toplam İyileşen Hasta Sayısı Grafiği.....	48
Şekil 3.20. Bugünkü Vaka Sayısı Grafiği	48
Şekil 3.21. Bugünkü Test Sayısı Grafiği	49
Şekil 3.22. Bugünkü Vefat Sayısı Grafiği	49
Şekil 3.23. Bugünkü İyileşen Sayısı Grafiği.....	50
Şekil 3.24. İşı Haritası Grafiği	50
Şekil 3.25. Tek Katmanlı Gerçek Ağır Hasta Sayısı ve Tahmini Ağır Hasta Sayısı Grafiği	53
Şekil 3.26. Tek Katmanlı Ağır Hasta Sayısı Grafiği	53

Şekil 3.27. Tek Katmanlı Gerçek Toplam Vaka Sayısı ve Tahmini Vaka Sayısı Grafiği	54
Şekil 3.28. Tek Katmanlı Toplam Vaka Sayısı Grafiği	55
Şekil 3.29. Tek Katmanlı Gerçek Toplam İyileşen Sayısı ve Tahmini İyileşen Sayısı Grafiği.....	56
Şekil 3.30. Tek Katmanlı Toplam İyileşen Hasta Sayısı Grafiği.....	57
Şekil 3.31. Tek Katmanlı Gerçek Toplam Vefat Sayısı ve Tahmini Vefat Sayısı Grafiği	58
Şekil 3.32. Tek Katmanlı Toplam Vefat Sayısı Grafiği.....	58
Şekil 3.33. Üç Katmanlı Gerçek Ağır Hasta Sayısı ve Tahmini Ağır Hasta Sayısı Grafiği	60
Şekil 3.34. Üç Katmanlı Ağır Hasta Sayısı Grafiği	60
Şekil 3.35. Üç Katmanlı Gerçek Toplam İyileşen Sayısı ve Tahmini İyileşen Sayısı Grafiği	61
Şekil 3.36. Üç Katmanlı Toplam İyileşen Hasta Sayısı Grafiği	62
Şekil 3.37. Üç Katmanlı Gerçek Toplam Test Sayısı ve Tahmini Test Sayısı Grafiği	63
Şekil 3.38. Üç Katmanlı Toplam Test Sayısı.....	63
Şekil 3.39. Üç Katmanlı Gerçek Toplam Vaka Sayısı ve Tahmini Vaka Sayısı Grafiği	65
Şekil 3.40. Üç Katmanlı Toplam Vaka Sayısı	65

ÖZET

TÜRKİYE'DE COVID-19 HASTALIĞININ MAKİNE ÖĞRENMESİ YÖNTEMİYLE ANALİZİ VE TAHMİNİ

2019 Yılında Çin'in Wuhan kentinden başlayarak tüm dünyaya yayılan ve pandemi halini alan Corona Virüs bütün ülkelere büyük zararlar vermiştir. Covid19 tüm dünyada milyonlarca insana bulaşarak salgın halinde ölümlere neden olmuştur. Ülkeler gerek sosyal gerekse ekonomik açıdan büyük zararlar görmüştür. Her ülke kendi içinde birçok önlemler almıştır ve bazı önleyici faaliyetlerde bulunmuştur. Bu önlemlerin çoğu teknoloji destekli çalışmalardır. Bu çalışmada öncelikle veri ve bilgi arasındaki fark anlatılmıştır. Çünkü bir makine öğrenimi projesi üzerinde çalışmak isteniliyorsa ilk önce veri ile bilgi arasındaki farkın tam olarak bilinmesi gerekmektedir. Daha sonra veri madenciliği konusunda bilgi verilmiştir. Bu çalışmada T.C. Sağlık Bakanlığının yayınlamış olduğu veriler, veri madenciliği tekniğine uygun bir şekilde elde edilmiştir. Veriler yapay sinir ağı projesinde kullanılabilmesi için bir ön işlemden geçirilmiştir. Daha sonra çalışmamıza uygun yöntemler incelenmiştir.

Dünyada makine öğrenimi projelerinde kullanılmak üzere geliştirilmiş ücretli ve ücretsiz birçok yazılım bulunmaktadır. Bu çalışmada tüm dünyada en çok kullanılan, hızlı ve güvenilir sonuçlar veren Python yazılım dili kullanıldı. Python dili tek başına proje geliştirmekte yeterli olmadığı için kütüphanelere ihtiyaç duymaktayız. Makine öğrenimi projeleri için geliştirilen bu kütüphaneler bizlere hız kazandırmaktadır.

Ülkelerin sağlık yönetimi gelecekte oluşacak hasta ve vaka sayılarını tespit ederek sağlık sistemini korumayı istemektedir. Bunun için ülkeler yapay zekadan destek alma yoluna gitmiştir ve birçok yapay sinir ağı projesi geliştirmiştir. Makine öğreniminden faydalanılarak erken tanı sistemini hayata geçiren birçok ülke bu çalışmalarında başarılı olmuştur. Bu çalışmada yapay sinir ağlarından faydalanılarak T.C. Sağlık Bakanlığının vermiş olduğu verisetleri ışığında makine öğrenmesi yöntemiyle verisetlerinin analizi yapılmıştır. Türkiye'deki vakaların sayısını görebilmek için tek katmanlı ve üç katmanlı bir yapay sinir ağı algoritması kullanılmıştır. Çalışma içerisinde kullanılacak veriler T.C. Sağlık Bakanlığının resmi internet sitesinden çekilerek düzenlenmiş ve .csv uzantılı bir şekilde internet bulut hesabı üzerine yüklenmiştir. Bütün çalışma boyunca buradan veriler çekilerek işlenmiştir. Makine öğrenimi algoritmasında gizli katmanların sayıları değiştirilerek elde edilen sonuçlar incelenmiştir ortaya çıkan değerler tablolar halinde sunulmuştur. İlk önce vaka sayıları tek katmanlı bir yapay sinir ağında daha sonrada aynı veriler üç gizli katmanlı yapay sinir ağında incelenmiştir. Çıkan değerler tablolarda verilmiş olup aynı zamanda değişimler grafikler ile sunulmuştur.

Anahtar Kelimeler : *Büyük Veri, Covid19, Makine Öğrenimi, Pandemi, Python, Yapay Zeka.*

ABSTRACT

PREDICTION AND MODELLING OF COVID-19 DISEASE IN TURKEY BY MACHINE LEARNING

Starting from the city of Wuhan in China in 2019, the Corona Virus, which spread all over the world and became a pandemic, caused great damage. Covid-19 has infected millions of people all over the world, causing deaths in an outbreak. Countries have suffered great losses both socially and economically. Each country has taken many precautions within itself and has taken some preventive measures which are mainly technology based. In this study, firstly, the difference between data and information is explained. In order to work on a machine learning project, the difference between data and information must be acknowledged. Following that, the information about data mining is provided. In this study, the data published by the Ministry of Health of the Republic of Turkey were obtained in accordance with the data mining technique. The data has been preprocessed so that it can be used in the artificial neural network project. Then, the methods suitable for our study were examined.

There are a variety of paid and free software developed to be used in machine learning projects in the world. In this study, Python software language, which is the most used all over the world and delivers fast and precise results, was used. Since the Python software language alone is not enough to develop projects, libraries are needed to expedite the process.

The health management of the countries aims to prevent damage to the health system by determining the number of patients and cases that will occur in the future. In order to enhance this, countries have used artificial intelligence and have developed many artificial neural network projects. Many countries that have implemented the early diagnosis system by making use of machine learning, have been successful in these studies.

Analysis of the datasets by using the artificial neural networks and the datasets provided by the Ministry of Health of the Republic of Turkey has been done by using the machine learning method.

While creating the study content, a single-layer and three-layer artificial neural network algorithm was used to see the number of cases in Turkey. The data to be used in the study were taken from the official website of the Ministry of Health of the Republic of Turkey, arranged and uploaded to the internet cloud account with a .csv extension.

During the whole study, data was extracted and processed from there. The results obtained by changing the number of hidden layers in the machine learning algorithm are examined and the resulting values are presented in tables. First, the case numbers were analyzed in a single-layer artificial neural network, and then the same data were examined in a three-layer artificial neural network. The resulting values are given in the tables and the changes are demonstrated with graphics.

Anahtar Kelimeler: *Artificial Intelligence, Big Data, Covid19, Machine Learning, Pandemic, Python.*

1. GİRİŞ

Covid-19 ilk olarak 2019 yılında Çin'in Wuhan kentinde ortaya çıkan ve pandemi şeklini alarak tüm dünyaya yayılan bir tek sarmallı RNA virüstür. Virüsün ilk önce Çin'in Wuhan eyaletindeki balık pazarında görüldüğü ve hızlı bir şekilde tüm dünyaya yayıldığı bilinmektedir. Dünya sağlık örgütü tarafından pandemi olarak ilan edilmesi ile tüm ülkeler en üst seviyede alarmla geçerek tedbirler almaya başlamıştır. Bu tedbirler kapsamında evlerden gerekli olmadıkça sokağa çıkmalar ilk önce yasaklandı. Daha sonra kontrollü olarak iş yerleri ve kamuya açık olan bazı alanlar kapandı. Devletler, vatandaşlarına bulaştan korunmak için bilgi paylaşımlarında bulundu. Birçok ülkede koruyucu maskeler dağıtıldı ve dezenfektan malzemelerin üretimi ve dağıtımına ağırlık verildi. İnsandan insana bulaşabilen ve birçok varyantı olan koronavirüs birçok ülkede çok fazla can kaybına neden oldu. Corona Virüs, hayvandan hayvana, hayvandan insana ve insandan insana bulaşabilen bir yapıya sahiptir. Bu virüsün MERS (Orta Doğu Solunum Sendromu) ve SARS (Şiddetli Akut Solunum Sendromu) gibi hastalıklara neden olduğu bilinmektedir. Covid-19 virüsüne, SAR-CoV-2 virüsü neden olmaktadır (URL-3).

Covid-19 gençler ve çocuklardan çok yaşlılarda daha fazla etki göstermiştir. Pandemi sürecinde hükümetler, insanlara evlerinden çıkmamaları konusunda tavsiyelerde ve yasaklamalarda bulunmuştur. Salgının artmaması için evde oturan ve bu nedenle spor ve hareketleri azalan ve yeme alışkanlıklarının değişmesi neticesinde çoğu kişide artan kilo sorunları oluşmuştur. Pandemi sürecinde çoğu kişide mental, biyolojik ve fiziksel değişiklikler insanların yaşam kalitelerini olumsuz yönde etkilemiştir (URL-3).

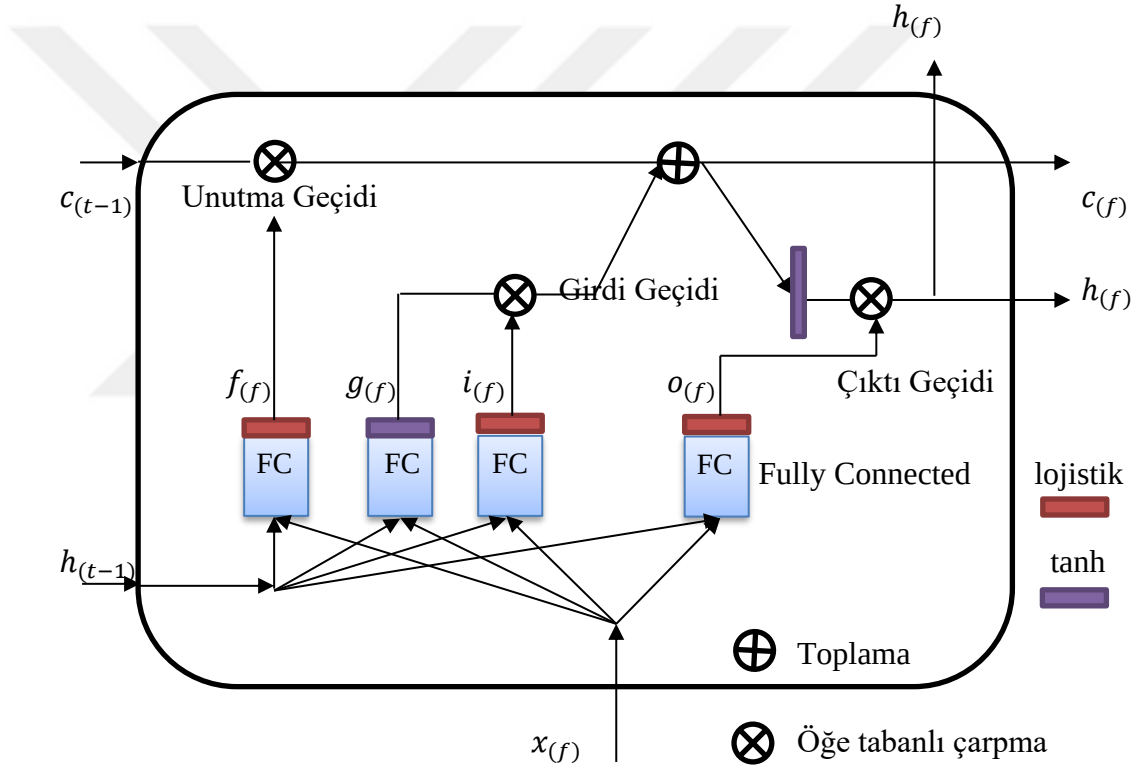
Evlerinden çıkamayan çalışanlar bu süreçte internet bağlantılarını kullanarak uzaktan çalışma sistemine geçmiştir. Birçok firma uzaktan çalışma sistemini firmalarına adapte etmiştir. Gerek iş yaşamındaki ihtiyaçları nedeniyle gerekse evde uzun süreli çalışmaları nedeniyle insanlarda internet bağımlılığı artmıştır. Okulların kapalı olması nedeniyle eğitim hayatı uzaktan eğitim ile gerçekleşmiştir. Eğitimin devam etmesi için uzaktan eğitime geçilmesi ile internete erişemeyen öğrenciler göz

önünde bulundurularak devlet televizyon kanallarından ders yayınları verilmeye başlanmıştır.



2. GENEL BİLGİLER

1997 yılında Sepp Hochreiter ve Jürgen Schmidhuber, uzun ömürlü kısa dönem bellek anlamına gelen LSTM algoritmasını geliştirmiştir. Daha sonraları LSTM üzerinde yapılan çalışmalar neticesinde bu konuda ilerlemeler sağlanmıştır. Bazı araştırmalarda ise Keras'ta SimpleRNN katmanının yerine LSTM kullanılmaktadır. GRU hücresi ise LSTM'in popüler bir çeşididir (Géron, 2021).



Şekil 2.1. LSTM hücresi

Kaynak: Géron (2021)

2.1. Salgın Hastalıklar

İnsanlığın ilk yıllarından itibaren salgın hastalıklar çok fazla can almıştır. Salgın ile mücadele çok zor süreçleri içinde barındırmaktadır. Özellikle modern tıptan önce insanlar salgın hastalıklarla mücadele konusunda çok yetersizdiler. Salgın hastalıklar zaman içerisinde farklı varyantlar ile çoğu zaman tamame yok olmadan

etkisini sürdürmeyi başarmıştır. Bu süreçte gelişen tıp çalışmaları sayesinde bulunan aşı ve diğer tıbbi yaklaşımlar, salgınları önceki dönemlere göre daha hızlı kontrol edilebilir kılmıştır.

2.2. Tarihteki Salgın Hastalıklar

Yeni tip koronavirüsün birçok farklı belirtisi görülebilmektedir. Ancak en çok görülen belirtileri yüksek ateş, nefes darlığı, öksürük ve nefes darlığıdır. İleri seviye vakalarda ise zatürre, ağır solunum yetmezliği, böbrek yetmezliği ve ölümler görülmektedir (URL-3).

2.3. Koronavirüsler

2019 yılında Çin'in Wuhan eyaletinde görülen ve tüm dünyaya yayılan koronavirüs pandemi şeklini alarak tüm dünya ülkelerinde krizlere neden olmuştur. İnsanlarda koronavirüsler 7 farklı tipte görülmektedir. Bunlar:

- HCoV-229E
- HCoV-OC43
- SARS-CoV
- HCoV-NL63
- HKUI
- MERS-CoV
- Covid-19

Koronavirüsler bulaştığı vakada solunum yolu enfeksiyonlarına neden olan patojenlerdir (Yıldırım, 2020).

2.4. Covid-19 Nasıl Bulaşır?

Yeni tip koronavirüsün birçok farklı belirtisi görülebilmektedir. Ancak en çok görülen belirtileri yüksek ateş, nefes darlığı, öksürük ve nefes darlığıdır. İleri seviye vakalarda ise zatürre, ağır solunum yetmezliği, böbrek yetmezliği ve ölümler görülmektedir. Yeni tip koronavirüs hastalığa yakalanan vakaların öksürmeleri, aksırmaları ile ortama saçılan damlacıkların diğer insanlar tarafından solunması ile bulaşır. Hastalığa yakalanan vakalar tarafından ortama saçılan tükürük damlacıklarına temas edenler ellerini yeterli seviyede yıkamadan yüz, göz, burun veya ağzına temas

etmesiyle de virüs bulaşmaktadır. Bu sebeple ellerin iyi bir şekilde sabunlu su ile yıkanması gerekmektedir (URL-3).

2.5. Kimler Daha Fazla Risk Altında?

Yeni tip koronavirüs olan Covid-19 enfeksiyonunda bazı kesimlerin daha fazla risk altında olduğu görülmüştür (URL-3). Edinilen veriler ışığında ;

- Hastalığa yakalananların %80'i hastalığı diğer vakalara göre daha hafif geçirmektedir.
- Hastalığa yakalananların %20'si hastane koşullarında tedavi edilmektedir.
- Hastalık en fazla 60 yaş ve üzerindeki vakaları etkilemektedir.

2.6. Hastalıktan En Çok Etkilenen Kişiler

Yeni tip koronavirüs olan Covid-19 enfeksiyonuna yakalanan vakaların arasında hastalıktan en çok etkilenenler şunlardır (URL-3):

- 60 yaş üstündeki vakalar
- Ciddi kronik rahatsızlıklara sahip olan vakalar
- Kalp hastalığına sahip olan vakalar
- Hipertansiyon hastalığına sahip olan vakalar
- Diyabet hastalığına sahip olan vakalar
- Kronik solunum yolu hastalığına sahip olan vakalar
- Kanseri gibi ağır hastalıklara sahip olan vakalar
- Sürekli hastanede görev alan sağlık çalışanları

3. GEREÇ VE YÖNTEM

2019 yılında tüm dünyaya yayılan corona virüsü pandemi halini almıştır. Bu sorun ülkelerin altından kalkmakta zorlandığı bir durum olmuştur. Hastanelerde ihtiyaç duyulan yatak sayıları ve oluşabilecek vaka sayıları için birçok yöntem araştırılmıştır. Bu çalışmada kullanılan vaka sayıları verileri de T.C. Sağlık Bakanlığının yayınlamış olduğu verilerden alınmıştır. Çalışmada Python yazılım dili ve kütüphanelerinden faydalanılmıştır.

3.1. Yapay Zekâ Nedir?

Yapay zeka bilimi, 1950’li yıllarda ortaya atılmış ve “bilgisayarlar düşünebilir mi?” sorusuna cevap ararken gelişmeye başlamıştır. Yapay zeka, insanların günlük hayatta yerine getirdiği düşünsel faaliyetleri, otonom hale getirilebilmesi için çalışır. Yapay zeka kendi içersinde derin öğrenme ve makine öğrenmeyi barındırmaktadır. Bu yüzden yapay zekayı anlayabilmek için makine öğrenmesi ve derin öğrenmenin de ne olduğunu öğrenmekte fayda vardır. Önceleri insanlar yapay zekanın kodlama ile mümkün olabileceğine inanmaktaydılar. Bu ilk başlarda birçok projede iyi sonuçlar vermiştir. Buna “Sembolik Yapay Zeka” denilmektedir. Sembolik Yapay Zeka 1950 - 1980 yıllar arasında oldukça popüler olmuştur. İlerleyen yıllarda ise farklı projelerde (görüntü işleme, ses tanıma, metin işleme vb.) yapay zekanın kullanımı gerektiğinde Sembolik Yapay Zekanın çok başarılı olmadığı görülmüştür. Sembolik Yapay Zekanın daha karmaşık, ileri seviye projeler için artık yeterli olmadığı anlaşılnca “Makine Öğrenmesi” fikri ortaya atılmıştır (Chollet, 2021).

Veri madenciliğinde en önemli iki konu istatistik ve makine öğrenimidir. Bunlar veri madenciliğinin temel taşlarıdır. Yapay zeka çalışmalarında karar ağaçları, yapay sinir ağları, destek vektörleri, en çok kullanılan yöntemlerdir. Yapay zeka çalışmaları insana özgü düşünme, bilgiyi saklama, verileri yönetme ve algoritmalar geliştirme yeteneğini araştırmaktadır. Yapay zeka, 1956 yılında Dartmouth College’de ismi telafuz edilmiş ve üzerinde uzun yıllar çalışmalar yapılmıştır (Akpınar, 2014).

3.2. Makine Öğrenmesi

Mekanik bir cihaz olan analitik makine 1830 – 1840 yıllarda Charles Babage tarafından tasarlanmış ve kullanılmıştır. Günümüzde kullanılan bilgisayarlardan farklı olarak analitik makine kendisinden istenilen matematiksel işlemleri yerine getirmek için bazı mekanik işlemler yapmaktaydı (Akpınar, 2014).

Arkadaşı ve destekçisi olan Lady Ada Lovelace bir konuşmasında analitik makine için “Analitik makine bizim bildiğimiz ve yapabildiğimiz şeyleri yapmak için bizlere yardımcı olmaktadır” demiştir. Yapay zekanın öncülerinden kabul edilen Alan Turing, 1950 yılında yayınladığı “Computing Machinery and Intelligence (Hesaplama Makinesi ve Zeka)” adlı makalesinde Lady Ada Lovelace tarafından ortaya atılan görüşe itiraz etmiştir. Alan Turing makalesinde Ada Lovelace’in söylediklerinin aksine kendilerinin bunu yapabildiklerini söylemiştir (Chollet, 2021).

Veri madenciliğinde bilgisayarları öğretmek amaçlanmaktadır. 1990’lı yıllardan beri birçok elektronik posta servisi sunan sağlayıcılar gereksiz posta algılamada veri madenciliğinden faydalanmaktadır. Bu veri madenciliği için bir örnektir. Bunun dışında optik karakter tanıma gibi uygulamalar da makine öğrenmesine örnek verilebilir. Birçok insanın makine öğrenmesi teriminden kendi kendine öğrenen ileri seviyede ve güvenilir olmayan tehlikeli robotlar algılasa da bu aslında doğru değildir (Géron, 2021).

Yapay zeka aslında bilgisayarların kendi kendilerine öğrenmesi mümkün mü? Bilgisayarlar veriye bakarak kuralları kendileri öğrenebilir mi? Sorularına cevap aramaktır. Makine öğrenmesi klasik yaklaşıma göre çalışmazlar. Makine öğrenmesinin en temelinde sistemin eğitilmesi yer almaktadır. Klasik programlamada veriler ve kurallar sisteme girdi olarak verilir ve cevapların üretilmesi sağlanır. Makine öğrenmesinde ise veriler ve cevaplar sisteme verilir ve kuralların üretilmesi sağlanır. Makine öğrenmesinde ihtiyaç duyulan konular:

- Veri Giriş Noktaları: Metinler, ses dosyaları, görüntü kayıtları, resimler
- Beklenen Çıktılar: Görüntünün kime ait olduğu kedi, köpek, insan vb.
- Algoritmanın Başarımının Ölçülmesi: Beklenen çıktı ile gerçekleşen çıktı arasındaki farkın ölçülmesi (Chollet, 2021).

3.3. Derin Öğrenme

Derin öğrenme, makine öğrenmesinin bir alt alanıdır. Derin öğrenmede çoklu katmanlarda veriler işlenir. Burada derin kelimesinden kast edilen derinlik değil birden fazla katman içermesidir. Projedeki katman sayısı modelin derinliğini oluşturmaktadır. Buna katmanlı gösterim öğrenimi veya hiyerarşik gösterim öğrenme de denilebilir. Derin öğrenme projelerindeki katman sayısı yüzlerce katmana ulaşabilmektedir. Makine öğrenmesinde ise bu sayısı birkaç katmandan oluşmaktadır. Bu sebeple makine öğrenmesine sığ öğrenme de denilebilmektedir (Chollet, 2021).

Derin öğrenme iş zekası çalışmalarında tercih edilmektedir. İşletmeler pazar koşullarının değişmesi ve değişen ihtiyaçların önceden belirlenmesi için iş zekası çalışmalarından büyük oranda destek almaktadır. İş zekası 90'lı yıllarda Gartner Group tarafından dile getirilmiştir (Akpınar, 2014).

3.3.1. Derin Öğrenmenin Kullanıldığı Alanlar

Derin öğrenme yöntemi çok eskilere dayansada popüler olması ve çok fazla kullanılmaya başlanması 2010 yılından sonra olmuştur. Makine öğrenmesinin yeterli olmadığı ve derin öğrenmeden faydalanılan alanlar olmuştur. Bunlar (Chollet, 2021):

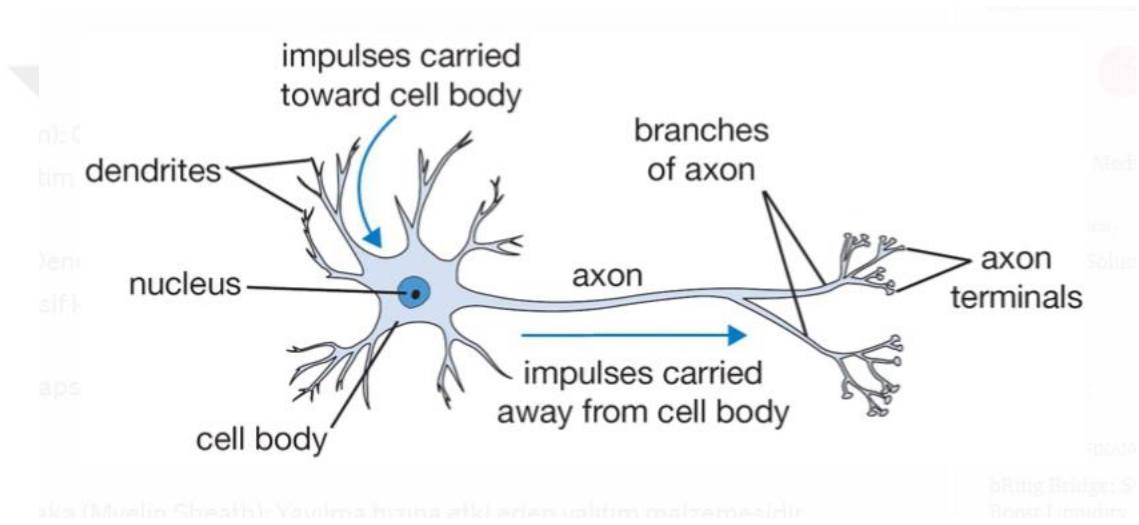
- Görüntü tanıma sistemleri
- Ses tanıma sistemleri
- El yazısı tanımlama sistemleri
- Tercüme yapan sistemler
- Metin işleme sistemleri
- Metin okuma sistemleri
- Google Now ve Amazon Alexa gibi dijital sistemler
- Otonom araç sistemleri
- Dijital reklamlar için hedef kitle analizi yapan Google Baidu ve Bing gibi sistemler
- Go gibi dijital oyunlar

3.4. Yapay Sinir Ağları

İnsan beyninden esinlenerek ortaya çıkan Yapay Sinir Ağları bir bilgi işlem teknolojisi ve birbirlerine bağlanan ve her biri kendine ait belleğe sahip işlem elemanlarından oluşmaktadır. Yapay sinir ağları biyolojik sinir sisteminden

esinlenilmiştir. Biyolojik sinir ağlarında olan nöron ve sinapslar yapay sinir ağlarında da mevcuttur. İnsanlar yeni bilgiler öğrenirken eğitimden faydalanır. Bu yapay sinir ağlarında da geçerlidir. Yapay sinir ağları, birbirine bağlanmış birçok nörondan oluşan matematiksel sistemlerdir (Yıldırım, 2020).

1940'lı yıllara kadar uzanan sinir ağları çalışması sinir ağları ile yapay sinir ağlarındaki benzerlikleri araştırmaktadır. İnsan beyninin temel yapısında akson ve dendritler bulunmaktadır. Sinir ağlarında nöronlar topladıkları sinyalleri akson ile başka nöronlara iletmektedir. Bu yapı yapay sinir ağlarının sistemini oluşturmaktadır (Arslan, 2020).



Şekil 3. 1. Bir Sinir Hücresinin Biyolojik Gösterimi

Kaynak: (URL-2)

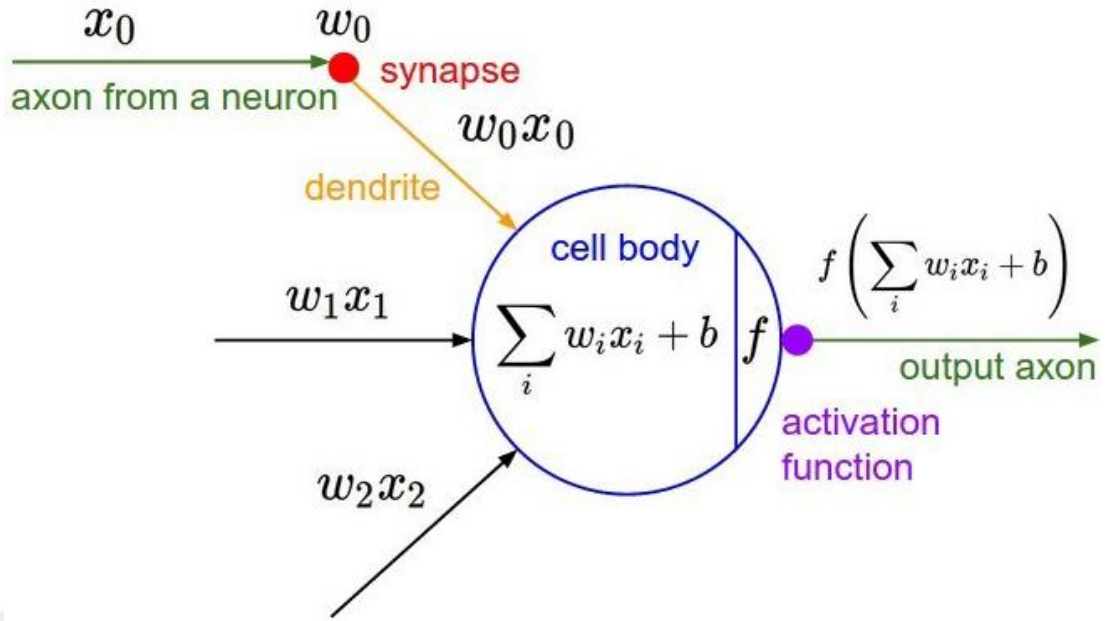
Axon (Axo): Sistemin çıkışı ve çıkış darbelerinin üretilmesi görevini üstlenmektedir.

Dentritler (Dendrites): Sistemin girişi olan bu kollar diğer hücrelerden gelen işaretleri toplar.

Sinaps (Synapse): Aksonlar ile dentritlerin bağlantısını gerçekleştirir.

Miyelin Tabaka (Myelin Sheath): Yalıtım görevi görür ve yayılma hızına etki eder.

Çekirdek (Nucleus): İşaretleri, periyodik olarak Akson boyunca yeniden üretir. İnsandaki bir sinir hücresinin matematiksel modeli ise şu şekildedir:



Şekil 3. 2. Bir Sinir Hücresinin Matematiksel Modeli

Kaynak: (URL-2)

İnsan yaşamında öğrenme süreci beynin gelişimini etkilemektedir. Öğrenilen her yeni bilgide sinaptik bağlantılar kendisini günceller ve yeni bağlantılar oluşturur. Yapay sinir ağlarının temel yapı taşı sinir hücreleridir. Bu hücre 5 temel unsurdan oluşmaktadır. Bunlar; giriş, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıkıştır. Hücrelerin çok sayıda girişi bulunmaktadır. Bu girişlerin her biri ağırlıklarla çarpılarak toplama fonksiyonuna iletilir. Elde edilen bilgi, aktivasyon fonksiyonunda işlenerek nöron çıkışına iletilir. Çıkan sonuç ise o nöronun değerini verir. Peki bu 5 temel unsurlar nelerdir:

Girişler ($x_1, x_2, x_3, \dots, x_n$): Kullanıcı tarafından sisteme girilen verilerdir.

Ağırlıklar ($w_1, w_2, w_3, \dots, w_n$): Nörona gelen giriş bilgisinin, nöronun çıkışına hangi oranda etki edeceğini ifade etmektedir.

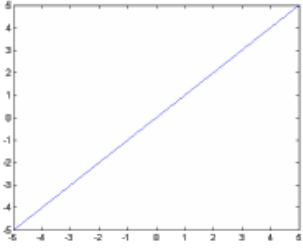
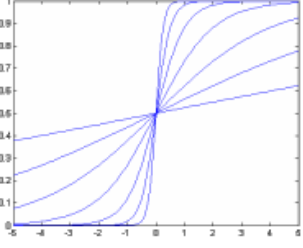
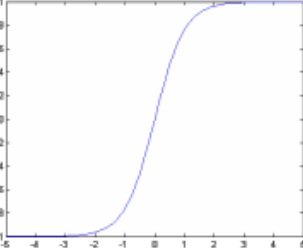
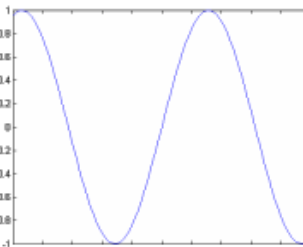
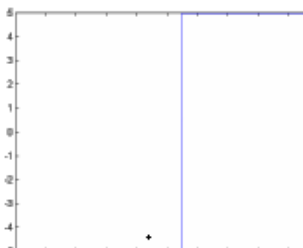
Toplama Fonksiyonu: Hücresinin net girdisini hesaplamak için kullanılan ve en çok tercih edilen fonksiyondur (Yıldırım, 2020).

Tablo 3. 1. Yapay Sinir Ağlarında Kullanılan Toplam Fonksiyonları

FONKSİYON ADI	DENKLEM
Ağırlıklı Toplam	$NET = \sum x_i.w_i$
Çarpım	$NET = \prod x_i.w_i$
Maksimum	$NET = MAX(x_i.w_i)$
Minimum	$NET = MIN(x_i.w_i)$
Artımlı Toplam	$NET = NET_{k-1} + \sum x_i.w_i$

Kaynak: Yıldırım, Ö. (2020)

Tablo 3. 2. Yapay Sinir Ağlarında Kullanılan Aktivasyon Fonksiyonları

Fonksiyon Adı	Fonksiyon Grafiği	Matematiksel Model
Lineer		$F(NE\text{T})= NE\text{T}$
Sigmoid		$F(NE\text{T})= NE\text{T}$
Hiperbolik Tanjant		$F(NE\text{T})= NE\text{T}$
Sinüs		$F(NE\text{T})= NE\text{T}$
Basamak		$F(NE\text{T})= NE\text{T}$

Kaynak: Yıldırım, Ö. (2020)

Geri besleme durumlarında, hücre kendi çıktı değerini giriş olarak kullanabildiği gibi başka hücreye de aktarabilir (Yıldırım, 2020).

3.4.1 Yapay Sinir Ağlarında Öğrenme

Yapay sinir ağlarında öğrenme süreci insan beynindeki öğrenme sürecine benzemektedir. Ancak biraz daha farklı öğrenme süreci devam etmektedir. Bu süreçte öğrenme süreci bilgisayar tememlleri ile işlenmektedir. İnsanların beyninde 10^{11} adet sinir hücresi bulunmaktadır. Bu nöronlar sinir sisteminin temelini oluşturmaktadır. Nöronlar dört bölümden oluşmaktadır. Bunlar; hücre gövdesi, sinaps, akson ve dendritlerdir (Arslan, 2020).

Organik beyin fonksiyonlarını tamamen benzetemediğimiz için insan beynine yakın yapılar modellenmeye çalışılmıştır. Bunun içinde birkaç farklı yapı kullanılmaktadır. Yapay sinir ağları ile öğrenme sürecinde bilgiler örnek veri girdileri ile sağlanır. Yapay sinir ağının bu girdiler ve çıktılar arasındaki ilişkisel ağ yapısını öğrenmesi hedeflenmektedir. Öğrenme sürecini tamamlayan yapay sinir ağı modeli artık verilen yeni veriler için doğruluğu yüksek çıkışlar verebilecek tecrübeyi kazanmış olur. Yapay sinir ağlarının öğrenme sürecinde 4 temel öğrenme yöntemi bulunmaktadır (Yıldırım, 2020).

Bunlar:

1. Öğretmenli (Supervised) Öğrenme
2. Destekleyicili (Reinforcement) Öğrenme
3. Öğretmensiz (Unsupervised) Öğrenme
4. Karma Öğrenme

3.4.2 Yapay Sinir Ağlarında Öğrenme Kuralları

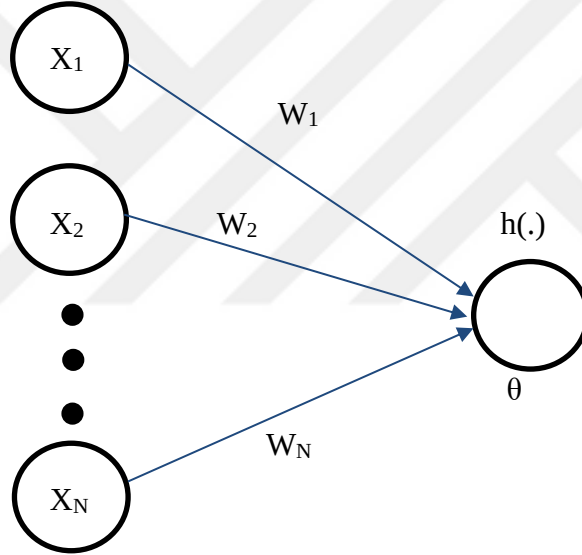
Yapay sinir ağlarında öğrenme sürecinde 4 temel öğrenme kuralı bulunmaktadır. Yapay sinir ağlarındaki öğrenme kuralları; Hebb, Hobfield, Delta ve Kohonen kurallarıdır. Tüm kuralların temelini 1949 yılında geliştirilen **Hebb** kuralı oluşturmaktadır. Bu öğrenme kuralına göre; matematiksel işaretleri aynı ve ardı ardına gelen hücrelerin bağlantıları güçlendirilmelidir. Bu hücrelerin bağlantılarının güçlenmesi için iki hücre arasındaki ağırlık değerinin artırılması gereklidir. Eğer hücrelerin matematiksel işaretleri aynı değilse bu sefer de ağırlıkları azaltılmalıdır (Yıldırım, 2020).

3.5. Yapay Sinir Ağı Mimarileri

Yapay sinir ağıları, beynin çalışmasını taklit etme fikri ile ortaya çıkmıştır. Bilinen ilk yapay sinir ağı olan klasik algılayıcı (perceptron – McCulloch-Pitts) modelinde bir nöron, diğer nöronlardan bilgileri toplar ve belirli bir fonksiyona göre işleyip bir çıktı üretir. En yaygın yapay sinir ağı mimarilerinden bazıları şunlardır:

1. Perceptron
2. Adaline
3. Çok Katmanlı Algılayıcı
4. Radyal Tabanlı Fonksiyon
5. Eleman Ağı
6. Hopfield Ağı

Bunun örnek görüntüsü şu şekildedir.



Şekil 3. 3. Basit bir yapay sinir ağı modeli

Kaynak: Arslan, İ. (2020)

Yukarıdaki şekilde görülen sol taraftaki oklar dendritler, sağ tarafta görülen oklar ise aksondur. $h(.)$ fonksiyonu aktivasyon fonksiyonudur. Yukarıdaki şeklin matematiksel ifadesini yazılırsa,

$$\text{net} = \sum_{i=1}^N W_i X_i - \theta \text{ ve } y = h(\text{net}) \quad (2.1)$$

1980’li yıllarda çok popüler olan yapay sinir ağıları ilerleyen yıllarda yüksek seviyede işlem gücüne ve donanımlara sahip bilgisayarlara gerek duyduğu için ilgiyi kaybetmiştir. Ancak son yıllarda gerek grafik kartlarındaki gelişmeler gerekse diğer

donanımlardaki yüksek teknolojik ürünlerin hızla gelişmesi ile tekrar eski popülerliğine kavuşmuştur. Günümüzde görüntü tanıma, ses tanıma, poz tanıma, metin tanıma, otonom sistemler için gerekli olan işlem gücü eskiden olmadığı için bu konulardaki gelişmeler duraklama devri yaşamıştır. Ancak günümüzdeki teknolojik gelişmeler yapay zekanın özellikle otonom sistemlerde kullanımını kolaylaştırmıştır (Arslan, 2020).

McCulloch ve Pitts yayınlamış oldukları makalelerinde istenilen herhangi bir mantıksal önermeyi hesaplayabilen basit bir sinir hücresi modelini önerdiler. Bu önermeleri ileride yapay sinir hücresi olarak anılacaktır. Önerdikleri bu sinir hücresinin bir veya daha çok ikili girdi ve çıktısı vardır (Géron, 2021).

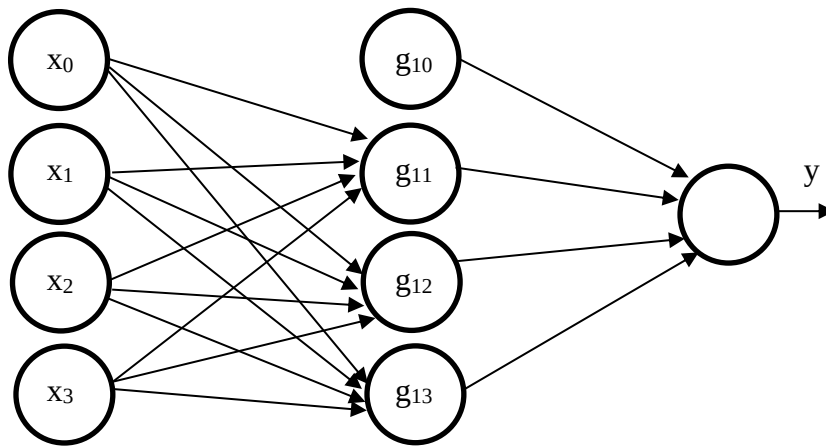
Gerçek hayattaki uygulamalarda değişken sayısı bizim en çok önemseydiğimiz konulardan birisidir. Örnek vermek gerekirse bir örnek proje için projemizi tanımlayacak 50 adet özellik belirleyelim. Birinci dereceden bağlantı modelimiz için 50 adet bağımsız değişken tanımlamamız gerekecek (Arslan, 2020).

İkinci derecede bir model kurmak istersek,

$$x_1, x_2, x_3, \dots, x_{50}, x_1^2, x_1, x_2, x_1, x_3, \dots, x_1, x_{50}, x_{22}, x_2 x_3, \dots, x_2 x_{50}, \dots, x_{50}^2 \quad (2.2)$$

Yukarıdaki denklemi kurmamız gerekecektir. Bu modelde ise 1.325 adet bağımsız değişken ile çalışmak durumunda kalacağız. Değişken sayısı 100 olduğunda ise bu bağımsız değişken sayısı 5.150 adete çıkmaktadır. Genelde biz gerçek hayattaki yapay öğrenme problemlerinde çok fazla değişken sayısı ile çalışmaktayız. Bu da yüksek seviyeli bilgisayarlara ihtiyaç duyulması anlamına gelmektedir (Arslan, 2020).

Örnek bir yapay sinir ağı modelinde,



Şekil 3. 4. Üç katmanlı model

Kaynak: Arslan, İ. (2020)

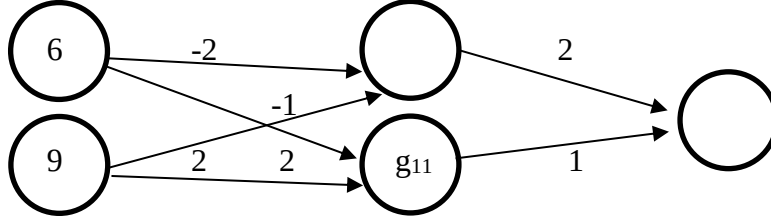
$$g_{11} = h (w_{101} x_0 + w_{111} x_1 + w_{121} x_2 + w_{131} x_3) \quad (2.3)$$

$$g_{12} = h (w_{102} x_0 + w_{112} x_1 + w_{122} x_2 + w_{132} x_3) \quad (2.4)$$

$$g_{13} = h (w_{103} x_0 + w_{113} x_1 + w_{123} x_2 + w_{133} x_3) \quad (2.5)$$

$$y = h (w_{201} g_{10} + w_{211} g_{11} + w_{221} g_{12} + w_{231} g_{13}) \quad (2.6)$$

Örnek bir model ile bu çalışmayı yaparsak,



Şekil 3. 5. Aktivasyon fonksiyonu kullanmadan gizli katman kullanarak modelin çıktısını hesaplama

Kaynak: Arslan, İ. (2020)

Aktivasyon fonksiyonları, yapay sinir ağlarında büyük öneme sahiptir. Aktivasyon fonksiyonlarını kullanmamızdaki amaç; yapay sinir ağlarına, doğrusal olmayan fiziksel dünyadaki özellikleri aktarmaktır. Aktivasyon fonksiyonunun uygulanmadığı çıkış sinyali, basit bir doğrusal fonksiyondur. Basit doğrusal fonksiyonlarda tek dereceli polinomlardır. Aktivasyon fonksiyonunu kullanmadığımız yapay sinir ağları basit ve sınırlı seviyede öğrenme yeteneğine sahiptir. Ancak biz fotoğraf, video, ses, metin gibi fiziksel dünyadaki veriler ile çalışacaksak eğer burada aktivasyon fonksiyonlarına ihtiyaç duymaktayız (Arslan, 2020).

Doğrusal olmayan aktivasyon fonksiyonları, yapay sinir ağlarının daha iyi öğrenmesini sağlamaktadır. Yapay sinir ağlarında kullandığımız aktivasyon fonksiyonları:

- Doğrusal Fonksiyon
- Basamak Fonksiyonu
- Sigmoid Fonksiyon
- Hiperbolik Tanjat Fonksiyonu
- ReLU Fonksiyonu
- Leaky Fonksiyonu
- Swish Fonksiyonu

Bu aktivasyon fonksiyonlarından en çok kullandığımız fonksiyonlardan birisi **Sigmoid Fonksiyondur**. Her fonksiyonun kendine ait avantaj ve dezavantajı bulunmaktadır. Her sinir ağı için aynı aktivasyon fonksiyonunu kullanmak yanlış olur. Yapay sinir ağlarında en çok kullanılan fonksiyonlardan olan Sigmoid fonksiyonu S harfine benzer bir yapıya sahiptir. 0 ile 1 arasında değer alan Sigmoid fonksiyonu çok sayıda işlemi yapay sinir ağı şeklinde gösterebiliriz (Arslan, 2020).

3.5.1. İleri Yayılım Algoritmaları

Yapay sinir ağlarından ilk katman gözlemlenebilir ve yorumlanabilir girdi katmanıdır. Son katman ise yine gözlemlenebilir ve yorumlanabilir çıktı katmanı olarak isimlendirilir. Ortadaki katmanlar ise gizli katmanlardır. Gizli teriminden kasıt bu katmanların gözlemlenebilir ve yorumlanabilir olmamasıdır.

İleri yayılım algoritmasında girdi katmanındaki veriler, çıktı katmanına kadar ileri yönlü bir hesaplama dahil olur. İleri yayılım algoritması ile kesin değil tahmini değerler elde edilir. Bu elde edilen tahmin sonuçlar %100 doğrulukta değildir. Tahmin değerlerin %100 doğru çıkmaması nedeniyle gerçek değerler ile arasında bir fark olacaktır. Amacımız bu hata farkının olabildiğince küçük olmasıdır (Arslan, 2020).

3.5.2. Geri Yayılım Algoritmaları

Yapay sinir ağlarında optimum sonuca ulaşmayı, gerçeğe en yakın çıktıyı almak hedeflenmektedir. Bunun için de en iyi katsayıların bulunması amaçlanmaktadır. Peki gerçeğe en yakın çıktıyı aldığımızı nasıl anlayacağız? Burada gerçeğe en yakın olduğumuzu bize söyleyen fonksiyon, kayıp fonksiyondur. Amacımız en düşük kayıp değerine ulaşmaktır. Ara katmanları olmayan çalışmalarda birkaç yineleme ile en iyi ağırlıkları bulabilmekteyiz. Ancak gerçek hayat problemlerinde ise çok fazla sayıda ara katmanlara sahip projeler ile çalışmaktayız.

1986 yılında David Rumelhart, Geoffrey Hinton ve Ronald Williams geri yayılım üzerine geliştirdikleri algoritma ile uzun yıllar boyunca MLP'lerin eğitilmesinde yaşanan sorunlar için çözüm üretmişlerdir. Autodiff, gradyanların otomatik hesaplanmasıdır. Geri yayılım algoritmasında da bundan faydalanılmaktadır ve buna geri modda autodiff denilmektedir (Géron, 2021).

Her katmandaki ağırlıkları güncellemek için geri yayılımdan faydalanmaktayız. İleri yayılım algoritmalarında, girdi verileri ile ağırlık ve aktivasyon fonksiyonlarını kullanarak sonuç değerlerini hesaplayabilmekteyiz. Geri yayılım algoritmasında ise sonuçtan yola çıkarak hata değerlerini ve eğimleri

kullanılarak geri dönük olarak işlem yapılır. İleri yayılım algoritmasında hata ve tahmin değerleri hesaplanmaktadır, geri yayılım algoritmasında ise kayıp fonksiyonunun her ağırlıktaki eğim hesaplanmaktadır (Arslan, 2020).

3.6. Yapay Zekâ

3.6.1 Yapay Zekânın Tarihsel Süreçleri

Yapay zeka, derin öğrenme, makine öğrenmesini kendi içinde barındırmaktadır. Yapay zekanın gelişiminde birçok farklı projeler hayata geçmiştir. Yapay zeka, derin öğrenme ve makine öğrenmesinin tarihsel gelişimine baktığımız zaman yıllar içerisinde birçok aşamadan geçtiğini görmekteyiz. Bu tarihsel süreçlere baktığımız zaman şöyle bir kronolojik sıralama yapılır (URL-1, URL-14).

- 1830 – 1840 yıllarında Charles Babbage tarafından Analitik Makine tasarlandı.
- 1943 yılında “A Logical Calculus of Ideas Immanent in Nervous Activity” – Warren S. McCulloch & Walter H. Pitts yayınlandı.
- 1950’li yıllarda sembolik yapay zeka başladı.
- 1950 yılında Alan Turing düşünen makineler yaratma hakkındaki düşüncelerini paylaştığı makalesini (Computing Machinery and Intelligence) yayınladı.
- 1950 yılında Isaac Asimov “Ben Robot” isimli bilim kurgu romanını yayınladı.
- 1951 yılında Christopher Strachey, Manchester Üniversitesi’nin Ferranti Mark 1 makinesini kullanarak dama programı geliştirdi. Aynı yılda Dietrich Prinz ise bir satranç programı yazdı.
- 1952 yılında Arthur Samuel makine öğrenimini icat etti. Makine öğrenimi tabirini çalışmalarında kullandı. IBM’deki Poughkeepsie Laboratuvarında ilk bilgisayar öğrenimi programlarını ortaya çıkardı. Dama oynaması için geliştirilen yazılım kendisini geliştirmesi için tasarlanmıştı. Bu yapılan otomatik öğrenme sistemi geliştirilmiş ilk makine öğreniminin örneği olmuştur.
- 1955 yılında John McCarthy yapay zeka terimini kullanmıştır ve 1956 yılında Dartmouth konferansında bunu dile getirmiştir.
- 1956 – 1974 yılları arasında yapay zeka altın çağını yaşamıştır.
- 1957 yılında psikolog olan Rosenblatt, çalıştığı Cornell Havacılık Laboratuvarı için “Perceptron: Algılayan ve Tanıyan bir Otomaton (The Perceptron: A

Perceiving and Recognizing Automaton)” başlıklı yazısını yayınladı. Bu çalışması ile Derin Sinir Ağlarının (DNN) kurucusu olarak tanındı.

- 1959 yılında nörofizyolojist ve Nobel ödüllü David H. Hubel ve Torsten Wiesel birincil görme korteksinde iki tip hücre olduğunu keşfetti. Bu çalışmaları derin öğrenme üzerinde büyük etkisi olmuştur.
- 1960 yılında Virginia Teknik Enstitüsü’nde hava-uzay ve okyanus mühendisliği profesörü Henry J. Kelly “Optimal Uçuş Rotaları için İrtifa Kuramı” makalesini yayınladı. Kontrol Kuramı konusunda geliştirdiği fikirler AI ve ANNs alanlarında uygulanmaya başladı.
- 1962 yılında profesyonel anlamda endüstriyel robot şirketi olan Unimation kuruldu.
- 1965 yılında matematikçi Alexey Ivakhnenko ve V.G. Lapa ilk çalışan derin öğrenme ağlarını yarattılar. Çalışmaları sonunda Veri İşlemede Grup Metodu’nu geliştirdiler ve bunu sinir ağlarında uyguladılar. Ivakhnenko 1971’de GMDH’yi kullanarak 8 katmanlı bir derin ağ yarattı ve Alpha bilgisayar tanımlama sisteminde öğrenme sürecini başarılı bir şekilde uygulayabilmiştir.
- 1968 yılında Arthur C. Clarke, “2001: Bir Uzay Destanı” isiminde bilim kurgu romanında “HAL 9000” isimli akıllı makineyi ana düşman karakter olarak sundu.
- 1970 – 1980 yıllarında yapay zekaya yatırımlar azalmış birçok yatırımcı bu alandan çekilmiştir. Buna yapay zekanın kışı denilmiştir.
- 1974 yılında Moravec’s Paradox oraya atılmıştır.
- 1978 yılında “Savaş Yıldızı Galactica” bilim-kurgu dizisi savaşçı robotlar olan “Cylon”ları tanıttı.
- 1979 – 1980 yılları arasında Fukushima görsel kalıpları nasıl tanıyacağını öğrenen yapay sinir ağı olan Neocognitron’u yarattı. Bu sistem örüntü tanıma, öneri sistemlerinde ve doğal dil işlemede kullanıldı.
- 1980 – 1987 yıllarında yapay zekaya yönelik yatırımlar arttı ve yatırım patlaması yaşanmıştır.
- 1980 yılında Amerika Stanford’da ilk yapay zeka konferansı Amerikan Yapay Zeka Derneği (AAAI) tarafından düzenlendi.

- 1980’li yıllarda sembolik yapay zeka uzman sistemlerde kullanılmaya başlandı.
- 1982 yılında Hopfield kendi adını taşıyan tekrarlayan sinir ağlarını geliştirdi.
- 1984 yılında “Terminatör” filminde dünyayı kıyamete sürükleyen bir yapay zeka sistemi olan “SKYNET” tanıttı.
- 1985 yılında hesaplamalı nörobilimci Terry Sejnowski NETtalk’ı yarattı.
- 1986 yılında D.E. Rumelhart & G.E. Hinton & R.J. Williams “Learning internal representations by error propagation” (Geri-yayılımlı Hatalarla Öğrenme Temsilleri) makalesini yayınladı. Bilim insanları bu makalede geri yayılım sistemini detaylı olarak anlattılar.
- 1987 yılında bilim-kurgu dizisi olan “Uzay Yolu: Yeni Nesil” kendi bilinci olan Android Yüzbaşı DATA’yı tanıttı.
- 1987 – 1993 yıllarında endüstride ilgisi azalan yapay zeka ikinci kışını yaşadı.
- 1989 yılında Yann LeCun evrişimli sinir ağlarını, geri yayılım kuramıyla birleştirdi. Geliştirdiği sistemle elle yazılan çekleri ve posta kodlarını okumak için kullanıldı.
- 1989 yılında Christopher Watkins, “Geciken Ödüllerden Öğrenmek (Learning from Delayed Rewards)” adındaki makalesinde Q-Öğrenme kavramını tanıttı.
- 1992 yılında Gerald Tesauro tavla oynamayı öğrenen yapay sinir ağına sahip TD-Gammon’u geliştirdi.
- 1993 yılında Jürgen Schmidhuber, 1000’den fazla katmana sahip derin öğrenmeyi başarıyla tamamladı. Bu çok katmanlı sinir ağları yönteminde büyük bir adım olarak anıldı.
- 1995 yılında “Support Vector Networks” günümüzdeki modeli Corinna Cortes & Vladimir Vapnik tarafından geliştirildi.
- 1997 yılında tarafından geliştirilen Deep Blue isimli yazılım Garry Kasparov’u satrançta yendi.
- 1997 yılında Jürgen Schmidhuber ve Sepp Hochreiter, Uzun Kısa-Süreli Hafıza (LSTM) sistemi bulundu.
- 1998 yılında Yann LeCun “Gradyan Tabanlı Öğrenimin Belge Tanımaya Uygulanması (Gradient-based Learning Applied to Document Recognition)” makalesini yayınladı. Bu sayede derin öğrenme alanında gelişmelere bir yenisi daha eklenmiş oldu.

- 2001 yılında Steven Spielberg “AI: Artificial Intelligence” filminde robot bir çocuğu başrolde oynattı.
- 2006 yılında Deep Neural Network (Deep Learning) Ruslan Salakhutdinov & Geoffrey Hinton çalışmalar yaptı.
- 2009 yılında Stanford Üniversitesinden Fei-Fei Li ImageNet’i geliştirdi.
- 2011 yılında Apple Siri geliştirildi.
- 2011 yılında IBM firması tarafından geliştirilen Watson yazılımı, televizyon yarışması olan “Jeopardy”de yarışma şampiyonları Rutter ve Jennings’i yendi.
- 2011 – 2012 yılları arasında Alex Krizhevsky, kendi adını taşıyan AlexNet’i geliştirdi. Kendi yarattığı proje olan evrişimli sinir ağı ile makine öğrenimi ve derin öğrenmede birçok projede başarıyla kullanılmıştır.
- 2012 yılında binlerce bilgisayara yayılmış sinir ağı ile “Kedi Deneyi” yapıldı.
- 2013 yılında Alex Krizhevsky, Ilya Sutskever ve Geoff Hinton, Large Scale Visual Recognition Challenge (LSVRC) görüntü işleme yarışmasında birinci oldu. Bu yarışmada yapay zeka çalışmaları için grafik işlemcisini kullanarak yapay zekada GPU kullanımı başlamış oldu.
- 2013 yılında “HER” isimli filmde oyuncunun yapay zeka destekli bir sisteme olan aşkı konu alınmıştır.
- 2014 yılında Facebook tarafından DeepFace geliştirildi. DeepFace %97.35 isabetle tanıma gerçekleştirdi.
- 2014 yılında Ian Goodfellow’un liderliğinde “Üretken Çekişmeli Ağlar (Generative Adversarial Networks) GAN” tanıtıldı.
- 2014 yılında yayınlanan “Ex Machine” filminde “Alan Turing” tarafından geliştirilen “Turing Testi”ni konu alarak yapay zeka alanında birçok konuyu dile getirmiştir.
- 2014 yılında “Amazon Alexa” geliştirildi.
- 2014 yılında Ian Goodfellow tarafından Generative Adversarial Networks – GAN (Çekişmeli Üretici Ağlar) geliştirildi. Bu sayede sahte içerikler ilk kez üretilmiş oldu ve üretilmenin önü açılmış oldu.
- 2015 yılında “Future of Life Institute” Yapay Zeka Güvenliği Konferansını düzenledi.
- 2016 yılında Google DeepMind’ın, AlphaGo’su karşılaştığı Lee Sedol’u 4-1 yendi.

- 2016 yılında Microsoft Tay adındaki Chatbotu Twittter’da kullanıcılar tarafından yanlış eğitildiği için 24 saat sonra kapatmak zorunda kaldı.
- 2017 yılında Future of Life Institute, Amerika Kaliforniya’da Asimolar Konferans Alanında “Faydalı Yapay Zeka” konferansını düzenledi.
- 2017 yılında Transformer Networks, (Dönüştürücü Ağlar) adında yeni bir sinir ağı geliştirildi.
- 2018 yılında Google tarafından “Dönüştürücü Ağ Tabanlı” dil işleme sistemi BERT’i ortaya çıkardı. AlphaZero: Deepmind AlphaZero kendi kendisiyle satranç oynadı.
- 2019 yılında 1.5 Milyar parametrelili GPT-2, OpenAI tarafından yayınlandı.
- 2020 yılında GPT-3 ile 175 Milyar parameter kullandı. Bu sayede Alphafold 50 yıl boyunca üzerinde çalışılan protein katlanma problemini çözerken yapay zeka kullandı.
- 2021 yılında OpenAI tarafından DALL-E çalışması yayınlandı (URL-1, URL-14).

3.7. Veri

3.7.1. Veri Saklama Yönetimi

Verilerin saklanması önceki yıllarda oldukça pahalı bir süreçti. Yapay zekanın kışı denilen dönemde de bu veri saklama maliyetleri, veri madenciliği ve derin öğrenmenin gelişmesini etkilemiştir. Gelişen teknolojiler sürekli veri saklama yöntemlerini araştırmaktadır. Zamanla büyüyen veriler, Navigasyonel SQL, İlişkisel, İlişkisel Sonrası olmak üzere üç farklı devrede incelenecek veri tabanı teknolojilerini geliştirmiştir (Akpınar, 2014).

3.7.2. Veri Betimleme

Veri betimlemesi verinin tanımlanması için yapılan ilk aşamadaki veri analizidir. Eğer pozitif bir sonuç daha sivri, negative bir sonuç daha düz bir dağılım gösteriyorsa buna basıklık denilmektedir. Pozitif sonuçlar aritmetik ortalamadan daha büyük olduğu yönde ve negafı sonuçlar ise değerlerin aritmetik ortalamadan daha küçük olduğu yönde ise çarpıklık oluşmaktadır (Akpınar, 2014).

3.7.3. Veri Görselleştirme

Veriden anlamlı bilgi çıkarma sürecindeki en önemli süreçlerden birisi de verileri görselleştirmektir. Veriler hakkında edinilen bilgilerin karşı tarafa daha anlaşılır bir şekilde aktarılabilmesi için verileri görselleştirmemiz gerekmektedir. Basit şekilde çizilen bir grafik karşı tarafa sayfalarca açıklama yapmaktan çok daha kısa sürede bilgi verebilmektedir. Python yazılımında da veri görselleştirmede çok fazla kullanılan kütüphaneler mevcuttur. Bunlardan en çok kullanılanı ise matplotlib'dir (Arslan, 2020).

Veri görselleştirme sürecinde en çok tercih edilen grafikler şunlardır (Akpınar, 2014):

- Çizgi, Sütun Grafiği
- Histogram
- Box – Whisker Çizimi
- Serpilme Çizimleri
- Serpilme Matris Çizimi
- Survey Plot
- Paralel Koordinat Çizimi
- Yüzey, Eşyükselti, Gofret Grafikleri
- Chernoff Yüzeyleri
- Coğrafi Görselleştirme
- Nokta Gösterimi

3.7.4. Veri Madenciliği Nedir?

Veri madenciliği her geçen gün önemi artan bir çalışma alanıdır. Veri madenciliği Yapay Sinir Ağları, Yapay Öğrenme, İstatistik, Veri Görselleştirme, Veritabanı Sistemleri ile birlikte çalışmaktadır. Veri madenciliği, tıp, uzay bilimleri, alışveriş sistemleri, otonom sistemler, sivil savunma, meteoroloji, jeoloji ve daha birçok bilim dalına hizmet vermektedir. En basit açıklama ile veri madenciliği; depolanan bilgiler içinden değerli ve anlamlı bilgileri bulmaktır. Veri madenciliği için birçok araştırmacı farklı tanımlamalar getirse de temelde ortak görüş faydalı bilgiye ulaşmaktır. Veri madenciliği çalışmaları, veritabanları ile birlikte çalışmaktadır. Veri madenciliğinde kullanılan yöntemlerin büyük çoğunluğu istatistik ve makine öğrenmesi ile ilişkilidir. Teknolojinin gelişmesi ile veri depolama sistemleri gelişmiştir. Bu gelişme de depolanan verilerin daha da büyümesi ile sonuçlanmaktadır.

Her yıl toplanan verilerin daha da büyümesi, bu verilerden değerli ve anlamlı bilgilerin çıkarılması ihtiyacını ortaya çıkartmıştır (İçöz, 2021).

3.7.5. Veri Madenciliğinin Kullanım ve Uygulama Alanları

Veri madenciliği gün geçtikçe önemi hızla artan bir çalışma alanıdır. Birçok sektör veri madenciliğinden faydalanmaktadır. Özellikle tıp, savunma sanayi, mühendislik araştırmaları, uzay araştırmaları, yer altı sismik çalışmalar, iletişim – telekomünikasyon, bankacılık, eğitim, bankacılık, elektronik ticaret, sigortacılık, supermarket ve diğer alışveriş kurumları, borsalar, madencilik, makine – metalurji, emlak, güvenlik gibi sektörlerde veri madenciliği çok fazla kullanılmaktadır. Veri madenciliğinin kullanımı her geçen gün artmaktadır ve sektör gözetmeksizin her sektörde kullanılmaktadır. Veri madenciliğinin en çok kullanıldığı sektörler ve uygulama alanları ise şu şekildedir:

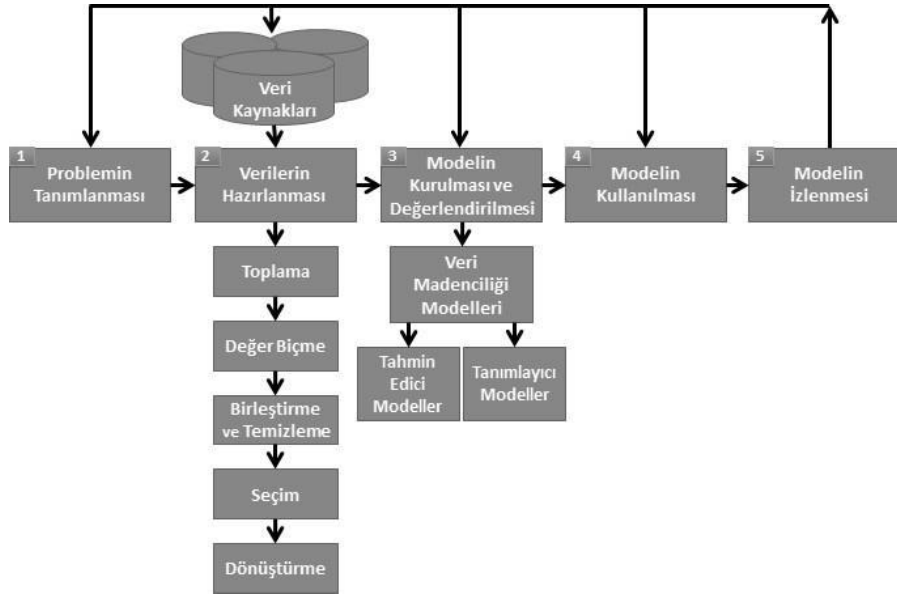
1. Pazarlama Sektörü
 - a. Müşterilerin satın alma örüntülerinin belirlenmesi
 - b. Müşterilerin demografik özellikleri arasında bağlantı bulma
 - c. Posta kampanyası oluşturma
 - d. SMS kampanyası oluşturma
 - e. Elektronik Posta kampanyası oluşturma
 - f. Sadakat uygulamalarını geliştirme
 - g. Müşteri davranışı tahminleri
 - h. Mevcut müşterinin korunması
 - i. Yeni müşteri kazanımı
 - j. Pazar sepeti uygulamaları
 - k. Müşteri ilişkileri yönetimi
 - l. Müşteri değerlendirme
 - m. Satış tahmini
 - n. Satın alma tahmini
 - o. Çapraz satış
2. Risk Yönetimi
3. Dolandırıcılık
 - a. Kredi kartı dolandırıcılığı
 - b. İnternet işlemleri
 - c. E-nakit dolandırıcılığı

- d. Sigorta dolandırıcılığı
- e. Kara – Para aklama dolandırıcılığı
- f. Sahte para dolandırıcılığı
- g. Bilgisayar ağlarına grime
- h. Telefon dolandırıcılığı
- i. Üyelik abonelik dolandırıcılığı
- 4. İşaret İşleme
- 5. Biyoloji
 - a. DNA Analizi
- 6. Tıp
 - a. Hastalıkların önceden tahmin edilmesi
 - b. Kalıtsal hastalıkların tahmini
- 7. Askeri Savunma Sistemleri
 - a. Dost – Düşman sistemleri tanıma

3.7.6. Veri Madenciliği Süreçleri

Veri madenciliği, veriden anlamlı bilgiler çıkarmaktadır. Peki bu hangi süreçler nelerdir (İçöz, 2021):

1. Problemin Tanımlanması
2. Verilerin Hazırlanması
3. Modelin Kurulması
4. Modelin Kullanılması
5. Modelin İzlenmesi



Şekil 3. 6. Veri Madenciliği Süreçleri

Kaynak: İçöz, E. (2021)

3.7.6.1 Problemin Tanımlanması

Veri madenciliğinin ilk adımı olarak problemin tanımlanması, problemin çözümü için gerekliliklerin net bir şekilde belirlenmesidir.

3.7.6.2. Verilerin Hazırlanması

Verilerin hazırlanma süreci, kullanılacak verilerin tespitini ve bunların kullanılmak için uygun hale getirilmesini içerir. Veri hazırlanma süreci iş yoğunluğunun yarısına yakın bir süreci kapsamaktadır. Veri hazırlama sürecinde karşılaşılan problemler, sık sık başa dönme ve verilerin tekrardan düzenlenmesini gerektirmektedir. Veri hazırlama “toplama”, “değer biçme”, “birleştirme ve temizleme”, “seçim”, “dönüştürme” süreçlerini kapsamaktadır (İçöz, 2021).

3.7.6.3 Toplama (Collection):

Bu süreç veri veri madenciliğini olumlu ve olumsuz yönde çok fazla etkilemektedir. Veri kaynaklarını seçerken çok dikkatli olunması gerekmektedir. Yanlış kaynaklardan alınan veriler analiz sonuçlarını olumsuz yönde etkileyecektir. Bunun yanı sıra eksik veriler veya gereğinden fazla veri de verilerin işlenmesi sürecinde çok fazla zaman kaybına neden olacaktır (İçöz, 2021).

Toplanmış verilerin çok büyük hacimlerde olması ve çok farklı kaynaklardan toplanması beraberinde o verinin parazitli, eksik ve tutarsız olmasına neden olmaktadır (Akpınar, 2014).

3.7.6.4. Değer Biçme (Assesment):

Toplanan verilerin birbirleri ile olan uyumları çok önemlidir. Birbirleri ile uyumsuz olan verilerden iyi sonuçlar elde etmek her zaman mümkün olmamaktadır. Toplanan verilerin aynı kodlamalara sahip olup olmaması, verilerin hangi koşullarda, nerede ve nasıl toplandığı da bu aşamada oldukça önemlidir (İçöz, 2021).

3.7.6.5. Birleştirme ve Temizleme (Consolidation and Cleaning):

Verilerin ön işleme sürecinde veri entegrasyonu, veri temizlemesi, veri dönüştürmesi ve veri indirgemesi işlemleri yapılmaktadır. Verilerin eksikliklerinin giderilmesi, veride bulunan parazitlerin giderilmesi ve tutarsızlıkların ortadan kaldırılması da veri temizlemesi işlemidir (Akpınar, 2014).

3.7.6.6. Seçim (Selection):

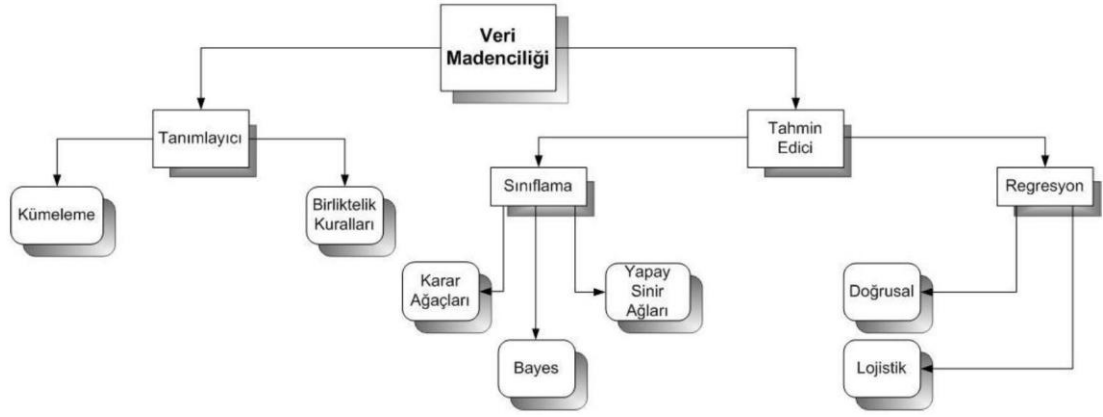
Çalışılacak projeye göre verilerin seçilmesi işlemidir.

3.7.6.7. Dönüştürme (Transformation):

Verilerin, veri madenciliğine uygun formatta kodlanması veya gösterim şeklinin dönüştürülmesidir.

3.7.7. Veri Madenciliği Modelleri

Veri madenciliğinde temel olarak iki model bulunmaktadır. Bunlar; tahmin edici (predictive) ve tanımlayıcı (descriptive) 'dir. Bunları kısaca açıklamak gerekirse tahmin edici modelde, sonuçları bilinen verilerden yola çıkarak modelin geliştirildiği ve bu modelden yararlanılarak sonuçları bilinmeyen veri kümeleri için sonuçların tahmin edildiği modellerdir. Tanımlayıcı ise karar vermeye rehberlikde kullanılan verilerin yapılarının tanımlanması şeklindeki modeldir (İçöz, 2021).



Şekil 3. 7. Veri Madenciliği Modelleri

Kaynak: İçöz, E. (2020)

Veri madenciliğinde birçok işlem için algoritmalar geliştirilmiştir. Dolandırıcılık tahmininden, hedef müşteri analizine kadar bu algoritmalarından faydalanılmaktadır. Bu algoritmaların bazıları sadece sınıflandırma, kümeleme gibi algoritmalar ile çalışırken bazıları ise birden fazla konuda kullanılmaktadır. Veri madenciliği temelde üç ayrı modele sahiptir. Bunları kısaca şu şekilde listelemek mümkündür: (Silahtaroglu, 2016)

1. Sınıflandırma Modeli
 - a. İstatistiğe Dayalı Algoritmalar
 - i. Bayesyen Sınıflandırma
 - ii. Regresyon
 - iii. CHAID
 - b. Mesafeye Dayalı Algoritmalar
 - i. En Yakın Komşu
 - ii. En Küçük Mesafe Sınıflandırıcısı
 - c. Karar Ağaçları
 - i. CART
 - ii. ID3
 - iii. C4.5
 - iv. Sprint
 - d. Genetik Algoritmalar
 - e. Yapay Sinir Ağları
2. Kümeleme Modeli

- a. Hiyerarşik Yönelimler
 - i.SLINK Algoritması
 - ii.CURE Algoritması
 - iii.CHAMELEON Algoritması
 - iv.BIRCH Algoritması
 - v.CLUCDUH Algoritması
 - b. Bölümlemeli Yötemler
 - i.K-Ortalama Algoritması
 - ii.PAM Algoritması
 - iii.CLARA Algoritması
 - iv.CLARANS Algoritması
 - c. Yoğunluğa Dayalı Algoritmalar
 - i.DBSCAN Algoritması
 - ii.OPTICS Algoritması
 - iii.DENCLUE Algoritması
 - d. Grid Temelli Algoritmalar
 - i.STING Algoritması
 - ii.Dalga Kümeleme
 - iii.CLIQUE Algoritması
 - e. Genetik Algoritmalar
 - f. Yapay Sinir Ağları
3. Bağlantı Analizi Modeli
- a. Apriori Algoritması
 - b. SETM Algoritması
 - c. Apriori TID Algoritması
 - d. GRI Algoritması

3.7.8. Veri Analizi

Veri madenciliği sürecinde eldeki verileri kullanarak gelecek tahminlerinde bulunmaktayız. Bu süreçte bilinmesi gereken bazı terimler vardır. Bunlar:

a. Tahmin (estimation): Yeterli verilere sahip olunmadığı halde yinede bu verileri kullanarak belirli amaçlar için tahminlerde bulunma sürecidir.

b. Gelceği Tahmin (forecasting): İleri bir tarih için bir değişkenin değerini tahmin etmektir. Bu süreçte de var olan verinin kalitesine göre değişkenin değeri, yaklaşık olarak hesaplanır.

c. Önsöylem (prediction): Latince kökenli bir kelime olan prediction, önceden söylemek demektir. Önsöylem, herhangi bir bilgiye ve tecrübeye sahip olunmadan gelecekte olacak olaylar hakkında fikir yürütmektir.

d. İş Zekâsı (Business Intelligence): Şirketlerin elde ettikleri verileri kullanarak doğru pazarın bulunması amacıyla yaptıkları pazar analizi çalışmaları, hedef müşteri analizi çalışmaları gibi etkin stratejiler yürütmek için yaptıkları enformasyonun analizini gerçekleştiren tüm süreçleri içermektedir. Bu süreçte firmalar ihtiyaç duydukları tüm yönetim araçlarının kullanımını, verinin analizini, verilerin toplanması ve saklanması, verilerin düzenlenmesini ve görselleştirmesini yapabilmektedirler. İlk defa 90'lı yıllarda Gartner Group tarafından ortaya atılmıştır (Akpınar, 2014).

3.7.9. Veri Madenciliğinde Karşılaşılan Sorunlar

Veri madenciliği projelerinde kullanılacak veriler, veritabanından çekilmektedir. Veritabanındaki verilerin eksiksiz veriler içermesi daha hızlı ve sorunsuz bir çalışma ortaya koyacaktır. Verilerin eksik, hatalı ve hantal bir şekilde depolanması, sistemin krize girmesine neden olacak ve veri madenciliği sürecini zora sokacaktır (İçöz 2021).

Veri madenciliğinde en sık karşılaşılan problemler şunlardır:

- Veritabanı boyutunun çok büyük olması
- Veritabanına girilen hatalı veriler
- Değersiz veriler
- Eksik veriler
- Gereksiz veriler
- Çok hızlı değişen veri tabanları.
- Belirsizliklik

3.8 Yapay Zeka Çalışmalarında Kullanılan Yazılımlar

Yapay zeka çalışmalarında kullanılan birçok bilgisayar yazılımı bulunmaktadır. Derin öğrenme çalışmalarında büyük veriler ile çalışmaktadır. Büyük verilerin

depolanması, ön işlemlerinin yapılması ve verilerin analizlerinin yapılması gibi çalışmalarda işlem gücü yüksek bilgisayarlara ve gerekli özelliklere ve kütüphanelere sahip bilgisayar yazılımlarına ihtiyaç duyulmaktadır. İşlenecek verilerin boyutuna göre bilgisayar donanımları da o seviyede güçlü olması beklenmektedir. Yapay zeka çalışmalarında genellikle işlemci gücü yüksek olduğu için iş istasyonu bilgisayarlardan faydalanılmaktadır. Ancak yapay zeka çalışmalarında sadece işlemci gücü yüksek bilgisayarlar yeterli olmuyor bunun için de birçok yazılım üretilmiştir. Bu yazılımların bir kısmı ücretsiz iken bir kısmı da ücretli ve lisanslıdır. Lisanslı yazılımların arkasında profesyonel şirketler bulunurken ücretsiz yazılımların arkasında bu şekilde şirketler bulunmamaktadır. Ücretsiz yazılımların büyük çoğunluğu açık kaynak kodlu ve tüm dünyadan yazılımcıların geliştirmesine açıktır. Açık kaynak kodlu yazılımların ve lisanslı yani ücretli yazılımların avantaj ve dezavantajları bulunmaktadır.

Ücretli yazılımların avantajı, yazılımın arkasında kullanıcılarına profesyonel destek veren bir şirketin olmasıdır. Ayrıca ücretli yazılımlar uzun yıllar boyunca güncelleme ve yeni özellikler kazandığı için kullanıcıların sürekli yeni yazılımlar kullanmasına gerek yoktur. Ücretli yazılımların genellikle kullanımı kolaydır. Genellikle bu tür yazılımları geliştiren şirketler yazılımın kullanımını anlatan kitaplar ve eğitim videoları yayınlamaktadır. Ücretli yazılımların dezavantajı ise genellikle yüksek fiyatlara satılması ve yıllık sözleşmeler içerdiği için aylık veya yıllık olarak lisans ücretlerinin yenilenmesidir. Lisanslar yenilenmediği takdirde yazılımın kullanımı durmakta ve önceden yapılmış çalışmalar kullanılamamaktadır.

Ücretsiz yazılımların avantajı, tüm dünyadan yazılım geliştiricilerinin katkısına açık ve kullanım tamamen ücretsiz olmasıdır. Büyük projelerde dahi ücretsiz açık kaynak kodlu yazılımların kullanıldığı görülmektedir. Ücretsiz yazılımların avantajları arasında birçok kullanıcının bilgi paylaşımı yaptığı tartışma forumlarının ve platformların bulunmasıdır. Kullanıcılar burada bir birleri ile bilgi paylaşımında bulunmaktadırlar. Ücretsiz yazılımların dezavantajı ise bu yazılımlar için gerekli kütüphaneleri geliştiren yazılımcılar, gönüllülük esasına göre çalıştıkları için ilerleyen dönemlerde bu topluluklar dağılabilmektedir. Tekrar aynı projeyi yürütecek gönüllü ve profesyonel yazılımcıların bir araya gelmesi zaman almaktadır. Bu da yazılımı kullanarak proje geliştirmekte olan kullanıcıları zor durumlara sokmaktadır.

Yapay Zeka, Veri Madenciliği, Veri Analizi projelerinde en çok kullanılan yazılımlar şunlardır:

- Python – Jupyter Notebook
- Google Developers Codelabs
- Knime
- Weka
- BigQuery
- JetBrains Datalore
- Causal AI
- MATLAB
- SAS
- Apache Spark
- BigML
- D3.js
- Excel
- ggplot2
- Tableau
- NLTK

Tez çalışmasında Python yazılım dili ve Jupyter Notebook yazılımından faydalanılmıştır. Peki bu yazılımlar nedir ve hangi kütüphaneleri kullanmaktadır?

3.8.1. Python Programlama Dili

Python programlama dili 1989 – 1990 yıllarında Guido van Rossum tarafından geliştirilmiştir. Python programlama dili, nesne yönelimli, yorumlamalı, yüksek seviyeli bir programlama dilidir. Python şu an günümüzde kullanılan tüm işletim sistemlerinde kullanılabilmektedir. Python, kolay kullanımı ile çabuk öğrenilebilen bir programlama dilidir. Bu sebeple diğer programlama dillerine göre çok daha fazla tercih edilmektedir. Python diğer programlama dillerine göre çok daha fazla

kütüphaneye sahiptir. Ücretsiz ve açık kaynak kodlu olarak piyasaya sürülen Python birçok yazılımcı tarafından desteklenmektedir. Tüm dünyada en çok kullanılan ve yüksek kullanıcı trafiğine sahip internet siteleri olan Youtube, Dropbox, Instagram Python ve kütüphanelerinden destek almaktadır. Python'ın çok fazla sayıda destekçisinin olması, her platformda çalışması, çok fazla kütüphanesinin bulunması, kullanımının kolaylığı onu birçok sektörde kullanılabilir hale getirmiştir (Arslan, 2020).

Veri bilimi, veri madenciliği, veri ambarı yönetimi, yapay zeka, derin öğrenme gibi yüksek kapasiteli projelerde Python programlama dili kullanılmaktadır. Burada en çok tercih edilen kütüphaneler ise şunlardır:

- Numpy
- Pandas
- Matplotlib
- SciPy
- Tensorflow
- Keras
- Scikit Learn
- Bokeh

3.8.2. Python Kütüphaneleri

Python yazılımı gibi kütüphaneleri de ücretsiz ve açık kaynak kodlu olarak piyasaya sürülmüştür. Burada yapacağımız projeye göre kütüphaneleri çekip kullanmamız gerekmektedir. Python kütüphaneleri bilimsel projeler için çok hızlı ve tutarlı sonuçlar vermektedir.

3.8.2.1. NumPy Kütüphanesi

NumPy yani Numeric Python. NumPy paketi, Python'da matematiksel işlemler yapabilmek için geliştirilmiş bir kütüphanedir. Numpy, büyük boyutta dizileri ve matrisleri desteklemetedir. Bu diziler ve matrislerde işlemler yapan bir kitaplıktır. NumPy, Python ile entegra çalışmakta ve hızlı dönüşler vermektedir (URL-11).

3.8.2.2. Pandas Kütüphanesi

Pandas kütüphanesi Python ile entegre çalışmaktadır. Pandas, zaman etiketli serileri ve sayısal taboları işlemek için bir veri yapısı oluşturur ve bu şekilde çeşitli işlemler bu veri yapısı üzerinde çalıştırılır. Pandas veri filtrelemeleri yapabilmektedir. Bunun dışında veri setleri üzerinde oldukça güçlü özellikleri vardır. Bunlardan bazıları şu şekildedir (URL-12):

- Veri filtrelemesi yapılabilir
- İstatistiksel fonksiyonlar oluşturur
- Çok boyutlu verilerden daha az boyutlu veriler oluşturur.
- Verilerde ayırma – uygulama – birleştirme işlemleri yapabilmektedir.
- Veri sütunlarına ekleme çıkartma yapılabilir.
- Veri setlerinde esnek çalışma imkanı vermektedir.
- Veri iskeletleri ile çalışabilmektedir.
- Farklı formattan verileri içer aktarabilmektedir.
- Veri temizleme işlemi yapabilmektedir.

3.8.2.3. Matplotlib Kütüphanesi

Matplotlib, Python programlama dili ve Numpy için yazılmış bir çizim kütüphanesidir. Tkinter, wxPython, Qt veya GTK gibi genel amaçlı GUI araç setlerini kullanarak grafiklerin uygulamalara gömülmesi için nesne yönelimli bir API sağlar. Matplotlib diğer kütüphaneler gibi ücretsizdir. Yazılım geliştiricilerin katkısına açıktır (URL-5).

3.8.2.4. SciPy Kütüphanesi

SciPy, Python programlama dili için geliştirilmiş bilimsel hesaplama ve teknik bilgi işlem için kullanılan bir kütüphanedir. SciPy ücretsiz ve açık kaynak kodludur. SciPy, bilimsel ve mühendislik alanında ihtiyaç duyulan işlemlerde yani optimizasyon, doğrusal cebir, entergrasyon, enterpolasyon, özel fonksiyonlar, FFT, sinyal ve görüntü işleme, ODE çözücülerde kullanılır (URL-7).

3.8.2.5. Tensorflow Kütüphanesi

Tensorflow, Python programlama dili ile geliştirilmiş makine öğreniminde kullanılan ücretsiz ve açık kaynak kodlu bir kütüphanedir. Kütüphanenin uzman olduğu konu derin sinir ağlarının eğitimi ve çıkarımıdır. Tensorflow, veri akışına ve türevlenebilir programlamaya dayalı sembolik bir matematiksel kitapıdır. Google gibi büyük işletmeler tarafından kullanılmaktadır (URL-13).

3.8.2.6. Keras Kütüphanesi

Keras, Python programlama dili ile geliştirilmiş ücretsiz ve açık kaynak kodlu sinir ağı kütüphanesidir. Keras diğer yazılımlar (Tensorflow, Microsoft Cognitive Toolkit, R, Theano, PlaidML) ile birlikte çalışabilir. Hızlı bir yapıya sahip olan Keras, derin sinir ağları projelerinde kullanılmaktadır. ONEIROS (Açık Uçlu Nöro-Elektronik Akıllı Robot İşletim Sistemi) için geliştirilmiştir. Keras, Google tarafından desteklenmektedir (URL-10).

3.8.2.7. Scikit-Learn Kütüphanesi

Scikit-Learn, Python programlama dili için geliştirilmiş ücretsiz ve açık kaynak kodlu makine öğrenimi kütüphanesidir. Makine öğreniminde kullanılan Destek Vektör Makineleri, Rastgele Ormanlar, Gradyan Artırma, K-Ortalamaları, DBScan gibi sınıflandırma, regresyon ve kümeleme algoritmalarını bünyesinde bulundurmaktadır. Python'ın diğer NumPy ve SciPy kütüphaneleri ile birlikte kullanılabilir. Büyük ölçüde Python dili ile yazılmıştır (URL-6).

3.8.2.8. Bokeh Kütüphanesi

Bokeh, Python programlama dili için geliştirilen esnek, etkileşimli, paylaşılabılır, verimli, ücretsiz ve açık kaynak kodlu bir kütüphanedir. Bokeh'de gelişmiş veya özelleştirilmiş servis taleplerini desteklemek için her zaman özel Javascript ekleyebilirsiniz. Bokeh, modern web tarayıcıları için etkileşimli görselleştirmeler oluşturmaya yönelik bir Python kütüphanesidir. Basit grafiklerden akış veri kümelerine sahip karmaşık panolara kadar birbirinden güzel grafikler oluşturmanıza yardımcı olur. Bokeh ile kendiniz JavaScript yazmadan JavaScript destekli görselleştirmeler oluşturabilirsiniz (URL-4).

3.8.2.9. Jupyter Notebook

Jupyter Notebook, Python kodlarını ve kütüphaneleri çalıştırabileceğimiz, notlar alabileceğimiz ücretsiz bir web uygulamasıdır. Jupyter Notebook ile

uygulamalar geliřtirmek çok hızlıdır. Bilgisayarda CPU kullanarak Jupyter Notebook ile işlem yapmak çok uzun süreler almaktadır. Ancak GPU ile işlem yapmak daha hızlıdır. Bu ancak her kullanıcının bilgisayarında bulunan bir özellik değildir. O yüzden büyük veriler ile çalışacaklar için uzak bağlantı ile internet üzerinden ücretli hizmet veren firmalarda bulunmaktadır (Chollet, 2021).

3.8.2.10. GeoPandas

Python'da çok kullanılan Pandas nesnelere coğrafi veri desteği sağlamak için kullanılan açık kaynak kodlu ve ücretsiz bir kütüphanedir. GeoPandas nesneleri geometri nesneleri üzerinde hareket edebilir ve geometrik işlemler yapmanıza olanak sağlar (URL-8).

3.9. Evriřimsel Sinir Ağları Kullanılarak Derin Bilgisayarlı Görü

1980'li yıllardan beri kullanılmakta olan Evriřimsel Sinir Ağları (CNNs: Convolutional Neural Networks) beynin görsel korteksiyle ilgili çalışmalarda ortaya çıkmıştır. Son yıllarda gelişen teknoloji ile yüksek hesaplama gücüne sahip bilgisayarların da yardımıyla CNN'lerin kullanımı artmıştır. Görüntü tanıma teknoloji alanında otoparklardan, insansız araçlara, tıp alanından askeri çalışmalara kadar birçok alanda kullanılmaktadır. CNN teknolojisinin gelişmesiyle sadece görüntü tanıma değil ses tanıma ve doğal dil işleme gibi diğer görevlerde de başarıyla kullanılmaktadır. 1958 – 1959 yıllarında David H. Hubel ve Torsten Wiesel kediler üzerinde yaptıkları çalışmalar ile görsel korteks üzerinde önemli bilgiler edinmiştir. Yaptıkları çalışmalarda görsel korteksteki pek çok sinir hücresinin küçük yerel alıcısına sahip olduklarını keşfettiler. Görsel kortek ile ilgili çalışmalar neocognitron'a da ilham kaynağı olmuştur (Géron, 2021).

3.10. Evriřimsel Katmanlar

Yapay zeka çalışmalarda adını sıkça duyduğumuz CNN'in en önemli yapısı evriřimsel katmandır. Birinci katmandaki sinir hücreleri alıcı alanlardaki piksellere bağlı olup girdi görüntüdeki piksellere bağlı değildir. Ancak ikinci katmandaki sinir hücreleri ilk katmandaki sinir hücrelerine bağlıdır. CNN teknolojisinin bu kadar iyi olmasının altında yatan neden CNN teknolojisinin hiyerarşik yapısının gerçek dünya görüntülerinde yaygındır. İki katmanında aynı boyutlara sahip olması için girdilerin etrafına sıfırlar eklenebilir. Buna Sıfır Doldurma (Zero Padding) denilir ve çok sıklıkla

başvurulan bir yöntemdir. Diğer bir yöntemse alıcı alanları arasında boşluklar bırakmaktır. Bu sayede büyük bir girdi katmanını çok daha küçük bir katmana bağlayabiliriz. Bu yöntem sayesinde hesaplamada yaşadığımız karmaşıklık azalacaktır (Géron, 2021).

3.11. Biriktirme Katmanlar

Yapay zeka çalışmalarında kullanılan katmanlardan bir diğeri de Biriktirme Katmanlarıdır. Biriktirme katmanlarındaki her sinir hücresi bir önceki katmandaki küçük dikdörtgensel bir alıcı alanda bulunan sınırlı sayıdaki sinir hücrelerinin çıktılarına bağlıdır bu yapısı ile biriktirme katmanları, evrimsel katmanlara benzemektedir. Biriktirme katmanlarında sinir hücrelerinin ağırlı bulunmamaktadır. Ancak biriktirme katmanlarının büyüklüğünü, adım aralığını, doldurma tipini tanımlamamız gerekmektedir. Biriktirme katmanlarında En Büyükleri Biriktirme Katmanı (Max Pooling Layer) ve Ortalamaları Biriktirme Katmanı (Average Pooling Layer) gibi toplulaştırma fonksiyonu kullanarak girdileri bir araya getirmektedir. CNN projelerinde en çok kullanılan Tensorflow uygulamasında biriktirme katmanını kullanmak oldukça kolay ve basittir. Keras uygulaması derinlemesine en büyükleri biriktirme katmanı bulundurmadağı halde Tensorflow alt düzeyde bunu içerir (Géron, 2021).

3.12. CNN Mimarileri

Yapay zeka çalışmalarında kullandığımız CNN projelerinde, CNN mimarilerinin çalışma mantığında şöyle bir durum vardır. CNN Mimarileri ilk önce bir birlerini takip eden ReLU katmanı sonra biriktirme katmanı daha sonra birkaç evrimsel katmanı daha sonra ReLU ve daha sonra tekrar biriktirme katmanını bir biri peşinde sıralar. Bu yapısı sayesinde ağ boyunca görüntünün ilerledikçe küçüldüğü ve derinleştigi görülür. CNN mimarilerinin gelişmesi için her yıl çeşitli yarışmalar düzenlenmektedir. Bu yarışmalar sayesinde yeni algoritmalar ve mimariler bu teknolojiye kazandırılır. Bu yarışmalardan en çok bilinenlerden birisi de ImageNet (<http://image-net.org>) yarışmasıdır. ImageNet hiyerarşinin her düğümünün yüzlerce hatta binlerce görüntüyle gösterildiği bir görüntü veritabanı projesidir. Bu yönde yapılan çalışmalar bilgisayar görüşü ve derin öğrenme araştırmalarının gelişmesinde çok fazla etkisi olmuştur. Bu veritabanında sunulan veriler, ticari olmadığı sürece

araştırmacılara ücretsiz sunulmaktadır. ImageNet yarışmalarında geliştirilen algortiamalar sayesinde hata oranı altı yılda %26'dan %2.3'e düşürülmüştür. Bu yarışmalarda geliştirilen algoritmalar yeni CNN Mimarileri olarak karşımıza çıkmaktadır (Géron, 2021).

En çok bilinen ve kullanılan CNN Mimarileri şunlardır:

- LeNet-5
- AlexNet
- GoogLeNet
- VGGNet
- ResNet
- Xception
- SENet

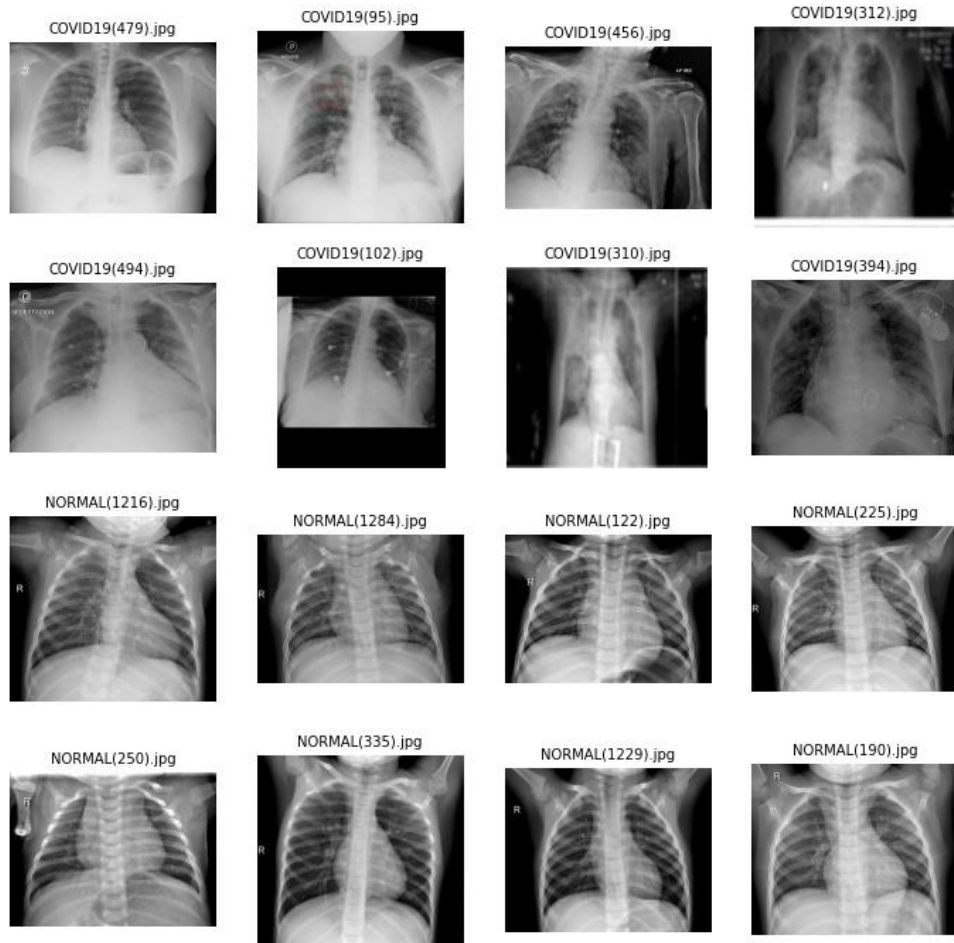
3.13. X-Ray Görüntülerinden Tahmininde Bulunan Evrişimsel Sinir Ağı

Bu çalışmada School of Disruptive Innovation tarafından kullanıma sunulan röntgen görüntülerini kullanarak bir kişinin koronavirüs enfeksiyonuna yakalanıp yakalanmadığını tahmin etmeye çalışacağız. Röntgen görüntülerini geliştirilen Evrişimsel Sinir Ağı (CNN) modelimiz ile tahmin etmeye çalışacağız.

CNN teknolojisi ile bugüne kadar birçok uygulama geliştirilmiştir. Tıp alanında birçok hastalık (Cilt kanseri tespiti, Normal pnömoni tespiti, Beyin kusuru tespiti, Retina kusuru tespiti vb.) CNN ile tahmin edilmeye çalışılmaktadır. İlk önce gerekli kütüphaneler yüklenir (URL-9).

Gerekli kütüphaneleri yükledikten sonra verisetlerinin bulunduğu internet adresini giriyoruz Burada github üzerinden açmış olduğumuz bir hesap üzerinden verileri çekilmektedir. Veri biliminde kullanılmak üzere birçok veriseti gerek GitHub gerekse Kaggle üzerinden araştırmacılara sunulmaktadır. Daha sonra eğitim ve test verisetlerinin olduğu klasör tanımlanır. Sisteme yüklediğimiz verisetleri kullanılarak elde edilen sonuçlar ekrana yazdırılır (URL-9).

Çıkan görüntü şu şekilde olacaktır.



Şekil 3. 8. Covid19 Pandemisinden Etkilenen Hastaların Akciğer Röntgenleri

Kaynak: URL-9

Daha sonra eğitim, doğrulama ve test grupları oluşturulur. Burada ImageDataGenerator üretici kullanarak dizinden resimlerin okunması sağlanır. Bunun için ilk önce gerekli kütüphaneyi sistemimize yüklenir. Eğitim, doğrulama ve test grupları için kullandığımız üreteçler tanımlanır. Burada tüm resimler 1/255 ölçeğine getirilir. Daha sonra CNN Modeli oluşturulur. Burada ilk önce gerekli kütüphaneler sisteme yüklenir (URL-9).

Kütüphaneler yüklendikten sonra evrişim katmanı eklenir. Filtreler, filtrelerin boyutları, aktivasyon fonksiyonu burada belirlenir. Pooling katmanı olarak En Büyükleri Biriktirme Katmanını (MaxPoolingLayer) kullanılmaktadır. Ardından Dropout katmanı eklenir. Ardından ReLU aktivasyon fonksiyonunu kullanan bir başka evrişim katmanı daha eklenir. Daha sonra tekrar pooling katmanı daha eklenir. Daha sonra tekrardan Dropout katmanı eklenir. Daha sonra resmi 1 boyutlu diziye düzleştirilir. Düğüm sayısı ve ReLU aktivasyon fonksiyonu olan bir katman daha

eklenir. Daha sonra Dropout katmanı eklenir. Daha sonra Sigmoid aktivasyon fonksiyonuna sahip bir katman daha eklenir ve bitirilir. Elde edilen çıktı şu şekilde olacaktır (URL-9).

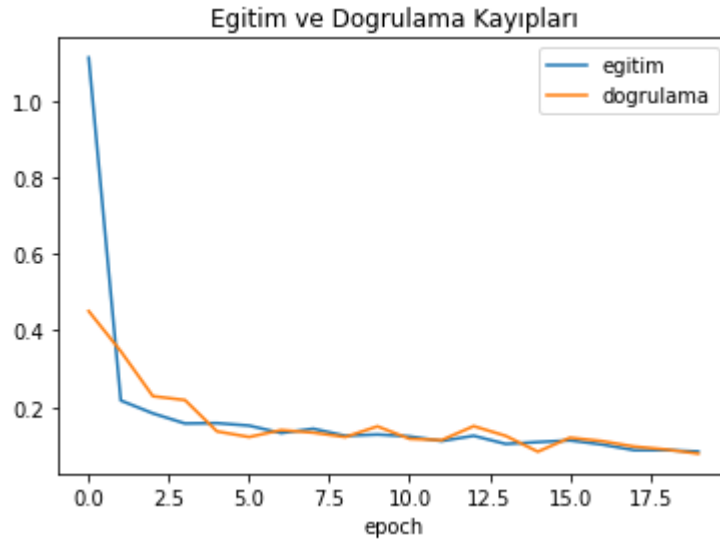
Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 150, 150, 32)	2432
max_pooling2d (MaxPooling2D)	(None, 75, 75, 32)	0
dropout (Dropout)	(None, 75, 75, 32)	0
conv2d_1 (Conv2D)	(None, 75, 75, 64)	51264
max_pooling2d_1 (MaxPooling2D)	(None, 37, 37, 64)	0
dropout_1 (Dropout)	(None, 37, 37, 64)	0
flatten (Flatten)	(None, 87616)	0
dense (Dense)	(None, 256)	22429952
dropout_2 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 1)	257
Total params: 22,483,905		
Trainable params: 22,483,905		
Non-trainable params: 0		

Şimdi sıra modelimizin eğitilmesi sürecinde. Burada modeli eğitirken yine ilk önce gerekli kütüphaneler yüklenir ve daha sonra model derlenir (URL-9).

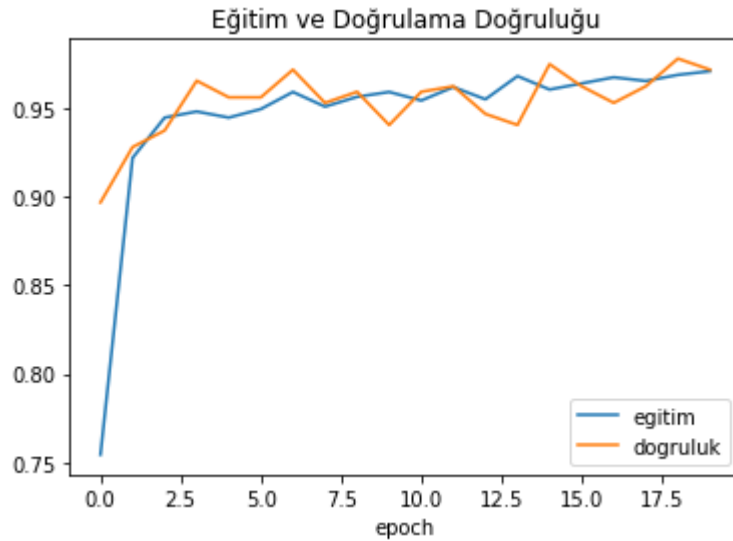
Daha sonra model eğitilir. Burada Epoch yani Eğitim Tur Sayısı 20 olarak belirlenir. Modelimizi eğitirken, verisetimizdeki tüm verler aynı anda eğitime dahil olmaz. Parçalar halinde eğitime dahil olurlar. İlk önce ilk parça eğitime dahil edilir daha sonra modelin başarımı test edilir ve başarıma göre geri yayılım ile ağırlıklar güncellenir. Daha sonra yeni eğitim kümesi ile model tekrar eğitilip ağırlıklar tekrar güncellenir. Bu eğitim adımlarının her birine epoch yani eğitim tur sayısı denir. Epoch sayısının fazla olması iyi olacağı anlamına gelmemektedir. Epoch fazla olması ezberlemeye yol açabilir. Az olmasıda istenilen sonuca ulaştırmayabilir. Bu yüzden

optimum sayıda epoch sayısının girilmesi önemlidir. Doğrulama ve eğitim kaybı sonuçları ekrana çizdirilir (URL-9).



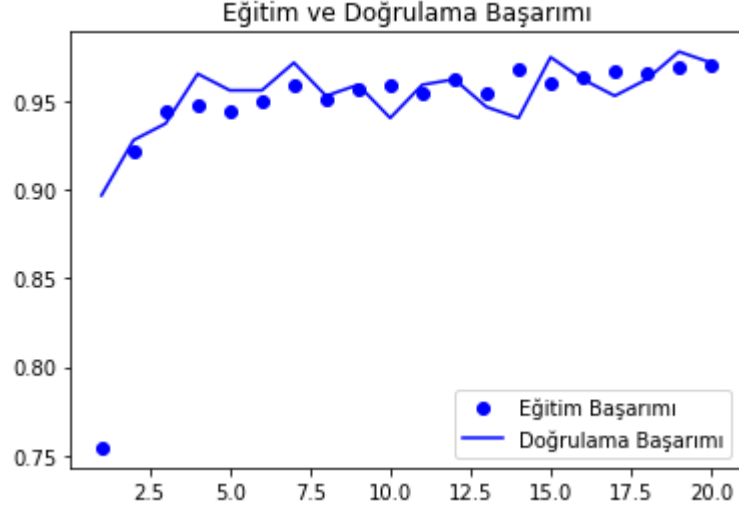
Şekil 3. 9. Eğitim ve Doğrulama Kayıplarının Grafiği

Eğitim ve doğrulamanın ekrana çizdirilmesinden sonra bir de eğitim ve doğrulamanın kesinliği ekrana çizdirilir.

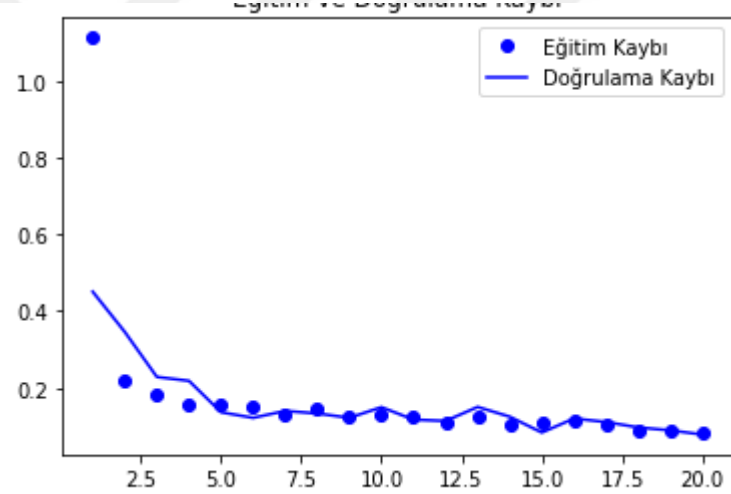


Şekil 3. 10. Eğitim ve Doğruluk Grafiği

Yapmış olduğumuz çalışma neticesinde %90 üzerinde doğrulama başarımlarını yakaladığı görülmektedir. Bunu ekrana çizdirmek için ilk önce gerekli kütüphane yüklenir.



Şekil 3. 11. Eğitim ve Doğrulama Başarımı Grafiği



Şekil 3. 12. Eğitim Kaybı ve Doğrulama Kaybı Grafiği

Burada başarımlar ve kayıplar daha iyi görülmektedir. Şimdi sıra bir hastanın COVID19'dan etkilenip etkilenmediğini tahmin etmek için modelimize kendi elimizde bulunan röntgen görüntüleri yüklenecek ve ondan bu görüntüleri inceleyerek tahminde bulunması istenecektir. Burada tahminde bulunurken bu aşamaya gelene kadar sisteme yaptığımız eğitimden faydalanacak. Bunun için ilk önce gerekli kütüphaneler yüklenir.

Daha sonra bilgisayardan bir görüntü dosyasını yükleyebilmek için gerekli kodlar yazdırılır. Burada yükleyeceğimiz görüntüler eğittiğimiz CNN modelimiz tarafından incelenecek ve bize bir tahmin sonucu verecektir.



ctest.jpg



ntest.jpg

Şekil 3. 13. Corona Virüse yakalanmış ve yakalanmamış iki farklı kişinin akciğer röntgen görüntüsü

Sisteme normal bir insanın akciğer röntgeni verildiğinde sistemin bize vermiş olduğu sonuç şu şekilde olmaktadır. ntest.jpg adlı röntgen görüntüsü normal bir insanın akciğer görüntüsü ctest.jpg ise covid hastası bir insanın akciğer görüntüsüdür.

3.14. Epidemiyolojide Kompartımanlı Modeller

Epidemiyolojide katman modeller 20. Yüzyılın başlarında bulaşın dinamiğini araştıran modellerdir. Çoğu salgın hastalıkta başarılı şekilde uygulanmıştır. 2019 yılında dünyayı saran Covid19 pandemisi içinde kullanılmış ve başarılı sonuçlar alınmıştır. Bu sistemin çalışma mantığında toplumu kapalı bir kompartıman olarak ele alır. Daha sonra bunları kendi içinde parçalara ayırarak bölümler ve isimlendirir. Burada S (hasta olma potansiyeli yüksek olan insanların toplam sayısıdır) I (enfekte ve/veya bulaştırma potansiyeli yüksek olan insanların toplam sayısıdır) R (iyileşmiş insanların toplam sayıları) şeklinde kısaltmalar ile ifade edilmektedir (Ateşli ve ark., 2021).

3.15. SIR ve SEIR Komartıman Modelleri

Epidemiyolojide en çok kullanılan modellerden ilki SIR Modelidir. Bu modelde salgın hastalık bölgesi incelenirken hastalığa yakalanmış vedirek bulaştııcı konumunda olan yani E kompartımanı olmayan modeldir. SIR ve SEIR Modellerin dışında SEIR-Q, SEIR- QD, SEIR – AHQ olmak üzere üç farklı model daha bulunmaktadır (Ateşli ve ark., 2021).

3.16. SIR ve SEIR Komartıman Modelleri Arasındaki Benzerlikler

Epidemiyolojide en çok kullanılan modeller olan SIR ve SEIR arasında çok önemli benzerlikler bulunmaktadır (Ateşli ve ark., 2021). Bunlar:

- İki modelde epidemiyolojide çok kullanılmaktadır.
- Bu modeller üreme sayısı ile aynı değerlere sahiptirler.
- Her iki modelde kabakulak, kızamık gibi enfeksiyonlarda kullanılır.

3.17. SIR ve SEIR Komartıman Modelleri Arasındaki Benzerlikler

- Epidemiyolojide ikisinin kullanım alanları E'den dolayı farklıdır.
- Enfekte olmuş ancak daha bulaştırıcı olmayan modeller için SIR kullanılır.

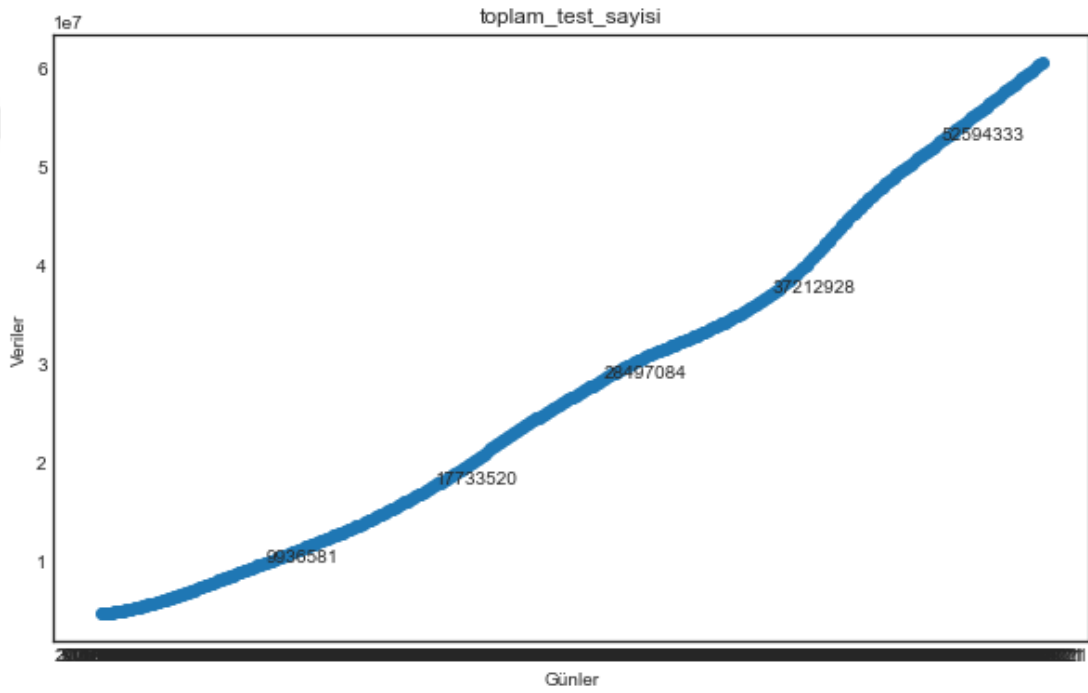
3.18. Yapay Sinir Ağı Modelinin Geliştirilmesi

Yapay sinir ağı çalışmamızda T.C Sağlık Bakanlığının yayınladığı verilerden yararlanılmıştır. T.C. Sağlık Bakanlığının yayınladığı veriler internet sitesi üzerinde .csv uzantılı bir dosya şeklinde saklanmaktadır. Verilerimiz bu sistemle çekilmektedir. Çalışmamız ilk önce tek katmanlı sonra da üç katmanlı olacak şekilde işlenmektedir. Üzerinde çalıştığımız veriler şunlardır:

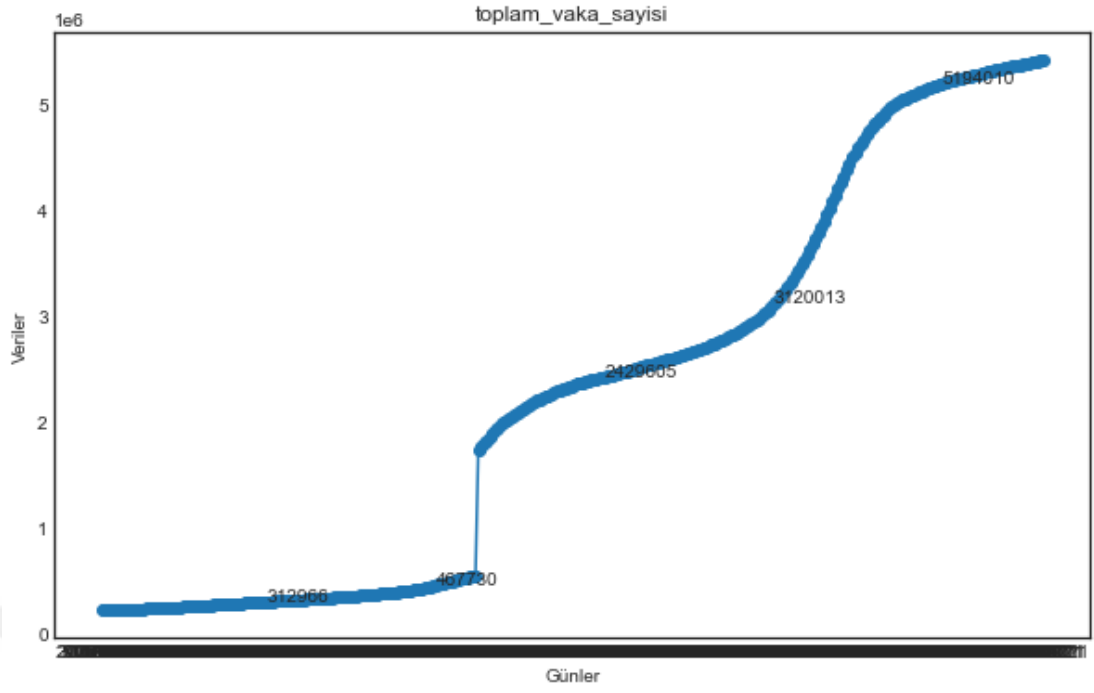
1. Tek Katmanlı Ağır Hasta Sayısı
2. Tek Katmanlı Toplam İyileşen Sayısı
3. Tek Katmanlı Toplam Vaka Sayısı
4. Tek Katmanlı Toplam Vefat Sayısı
5. Üç Katmanlı Ağır Hasta Sayısı
6. Üç Katmanlı Toplam İyileşen Sayısı
7. Üç Katmanlı Toplam Vaka Sayısı
8. Üç Katmanlı Toplam Vefat Sayısı

3.19. Tek Katmanlı Ağır Hasta Sayısı

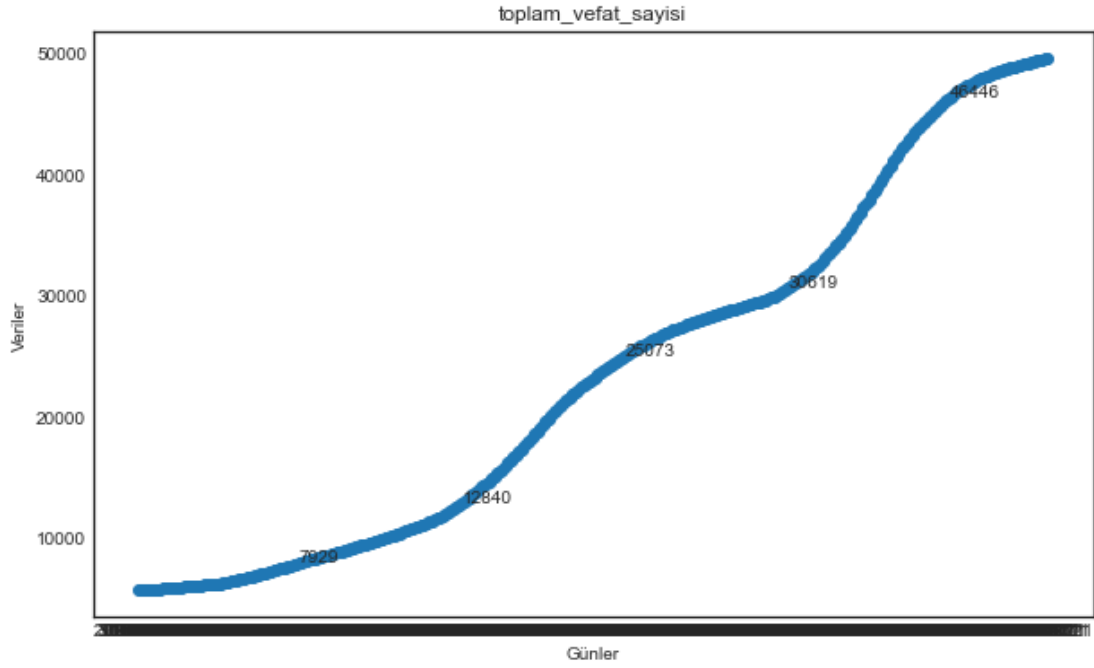
Tek katmanlı ağır hasta sayımız için ön hazırlıklar yapılmaktadır. Bellirli bir yere kadar tüm çalışmalar aynı şekilde gitmektedir. İlk önce gerekli kütüphaneler yüklenmektedir. T.C. Sağlık Bakanlığından alınan veriler yüklediğimiz verisetleri internet sitesine .csv uzantılı şekliyle yüklenir. Daha sonra yapay sinir ağı modelimize bu veriler çekilir. Ekrana veriler alınır ve veriler listelenir. Daha sonra veriler grafik halinde listelenir. Bunun için fonksiyon yazdırılır.



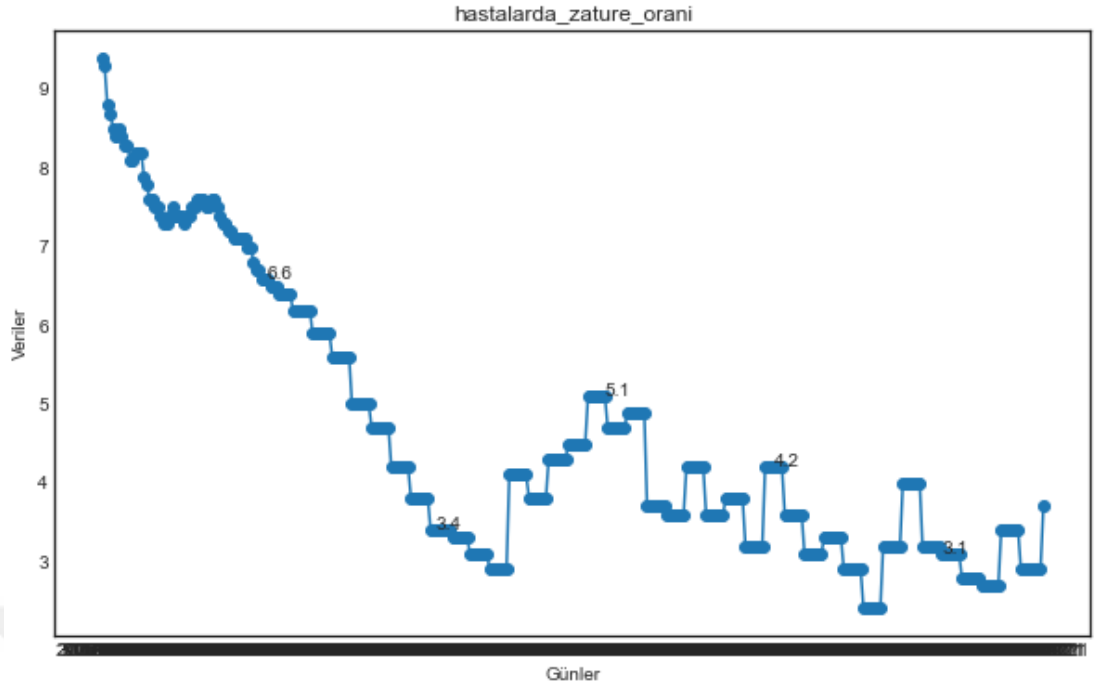
Şekil 3. 14. Toplam Test Sayısı Grafiği



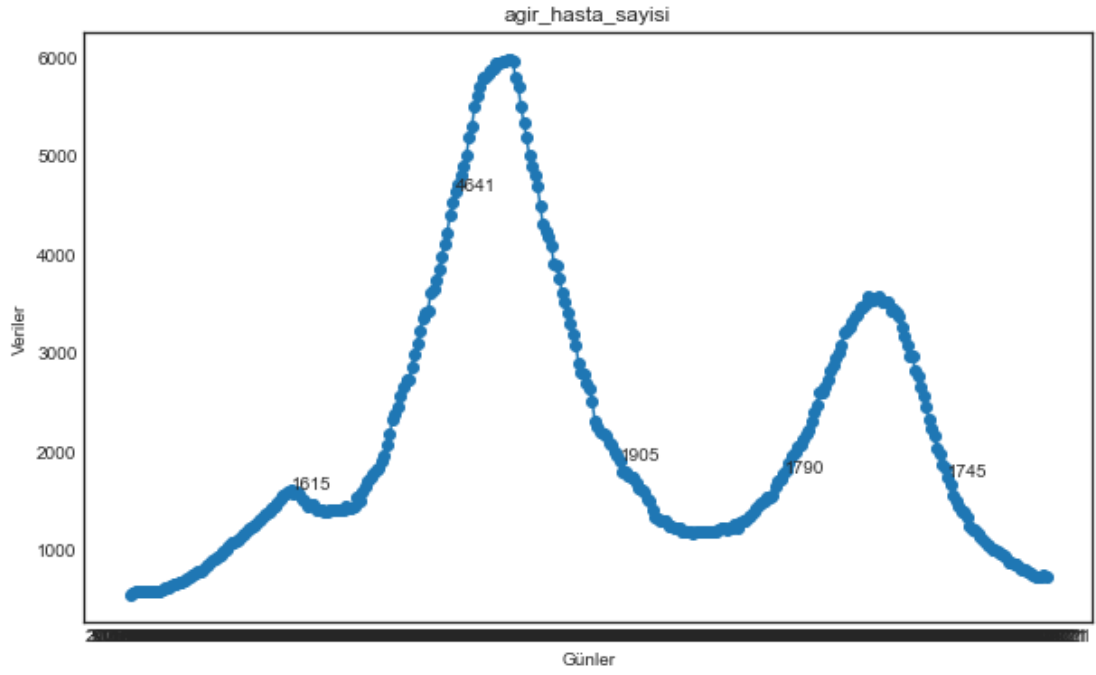
Şekil 3. 15. Toplam Vaka Sayısı Grafiği



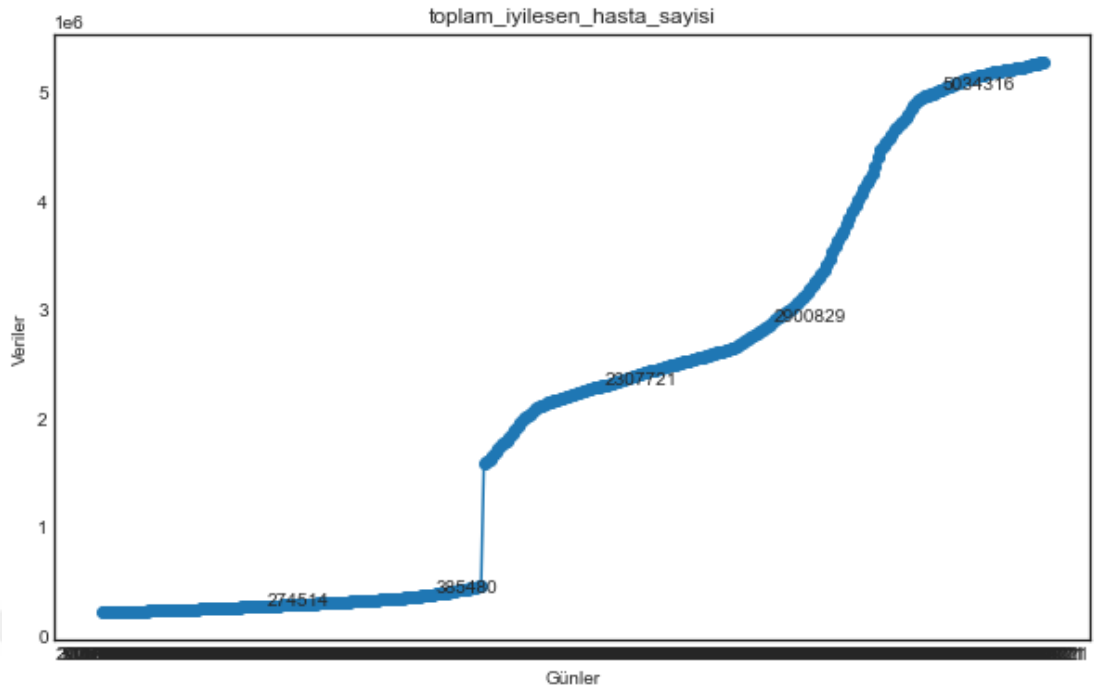
Şekil 3. 16. Toplam Vefat Sayısı Grafiği



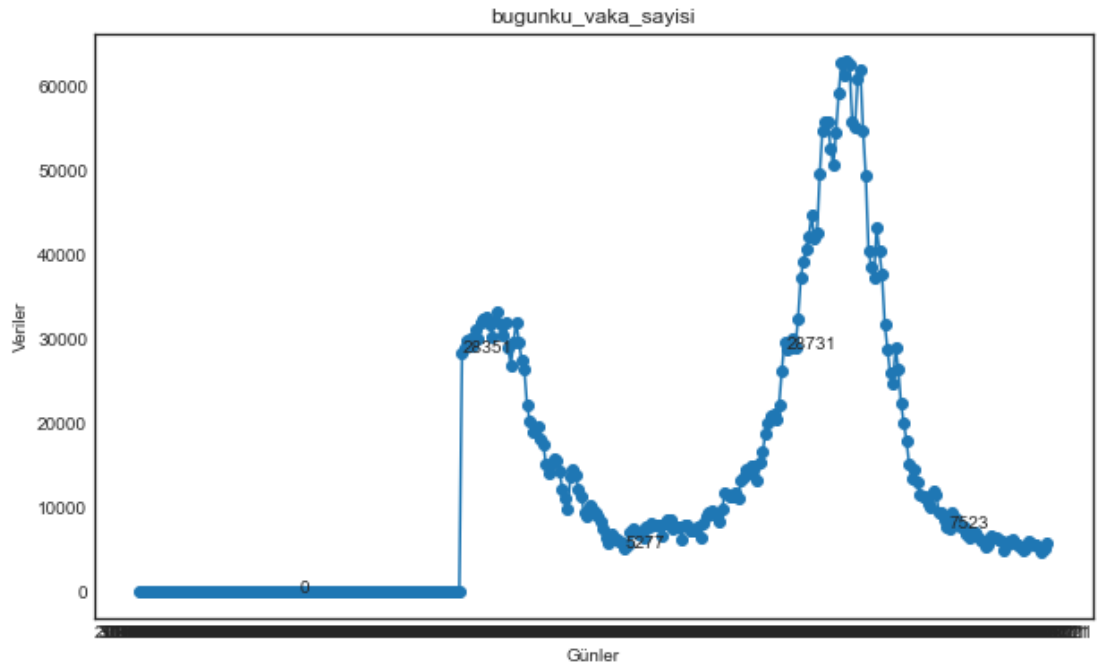
Şekil 3 17. Hastalarda Zature Oranı Grafiği



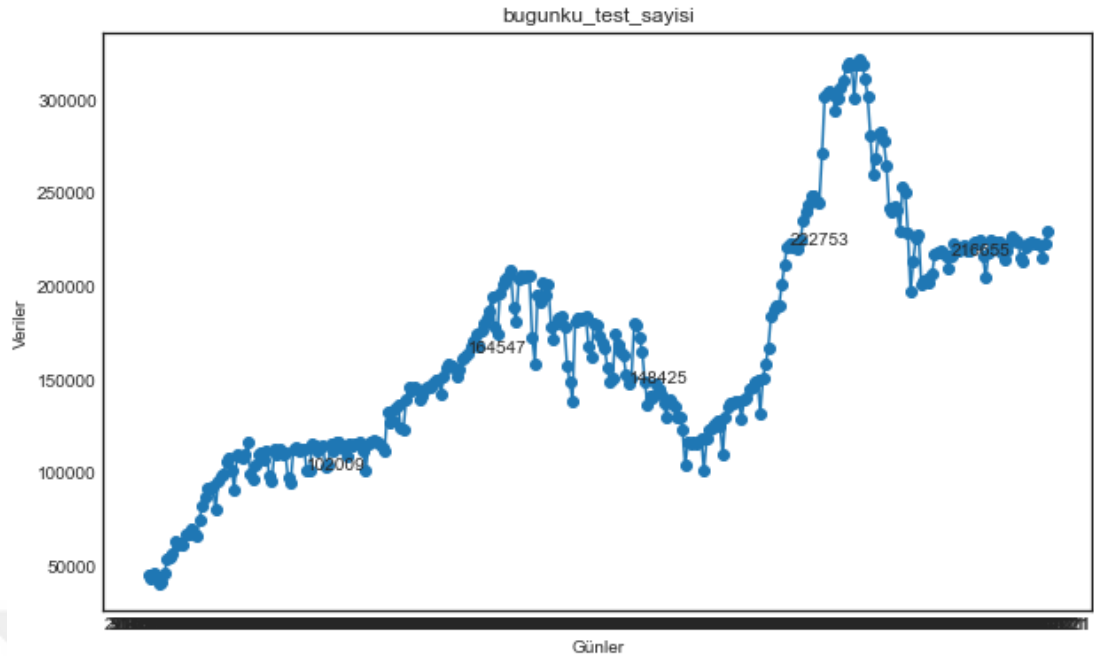
Şekil 3. 18. Ağır Hasta Sayısı Grafiği



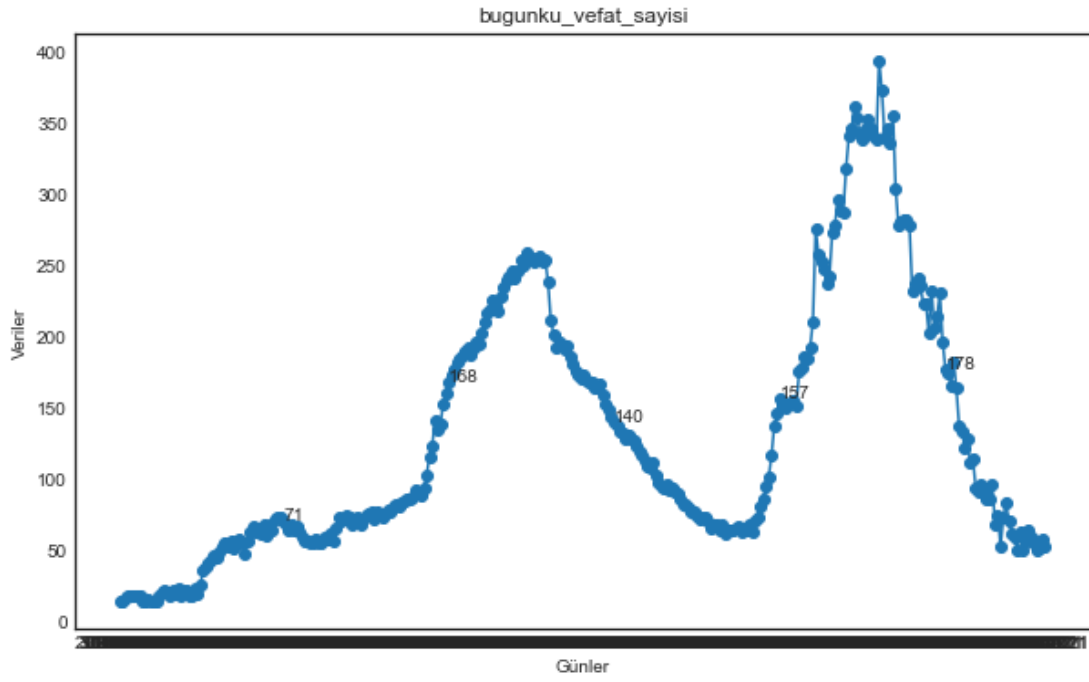
Şekil 3. 19. Toplam İyileşen Hasta Sayısı Grafiği



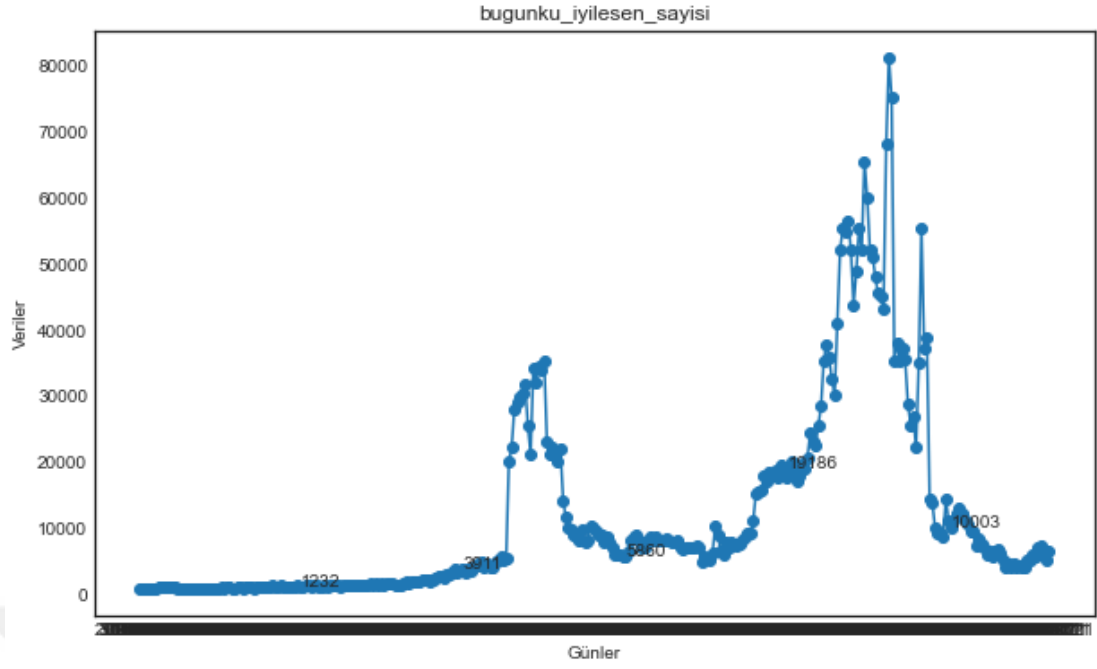
Şekil 3 20. Bugünkü Vaka Sayısı Grafiği



Şekil 3. 21. Bugünkü Test Sayısı Grafiği

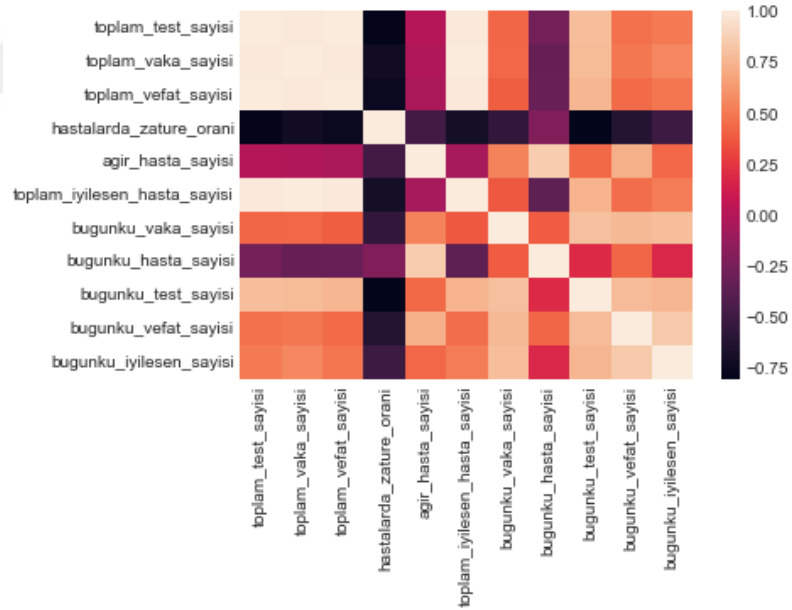


Şekil 3. 22. Bugünkü Vefat Sayısı Grafiği



Şekil 3. 23. Bugünkü İyileşen Sayısı Grafiği

Korelasyon sonuçları listeledikten sonra ısı haritası ekrana yazdırılır.



Şekil 3. 24. Isı Haritası Grafiği

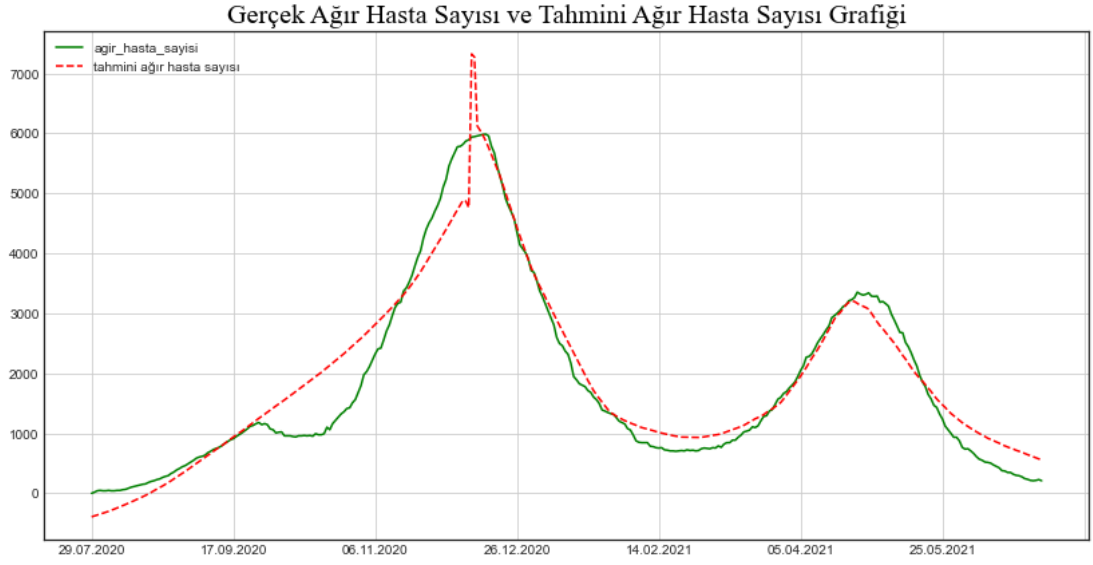
Isı haritası ekrana yazdırıldıktan sonra “Toplam Test Sayısı”, “Toplam Vaka Sayısı”, “Toplam Vefat Sayısı” ve “Toplam İyileşen Hasta Sayısı” verileri ekrana yazdırılır.

Tablo 3. 3. Toplam Test Sayısı, Toplam Vaka Sayısı, Toplam Vefat Sayısı ve Toplam İyileşen Hasta Sayısı Verileri

Tarih	Toplam Test Sayısı	Toplam Vaka Sayısı	Toplam Vefat Sayısı	Toplam İyileşen Hasta Sayısı
29.07.2020	0.000000	0.000000	0.000000	0.000000
30.07.2020	0.000774	0.000186	0.000341	0.000194
31.07.2020	0.001607	0.000375	0.000727	0.000390
01.08.2020	0.002410	0.000567	0.001158	0.000583
02.08.2020	0.003130	0.000757	0.001567	0.000776
03.08.2020	0.003870	0.000949	0.001999	0.000973
04.08.2020	0.004698	0.001158	0.002408	0.001169
05.08.2020	0.005662	0.001385	0.002839	0.001369
06.08.2020	0.006638	0.001607	0.003157	0.001574
07.08.2020	0.007654	0.001835	0.003498	0.001777

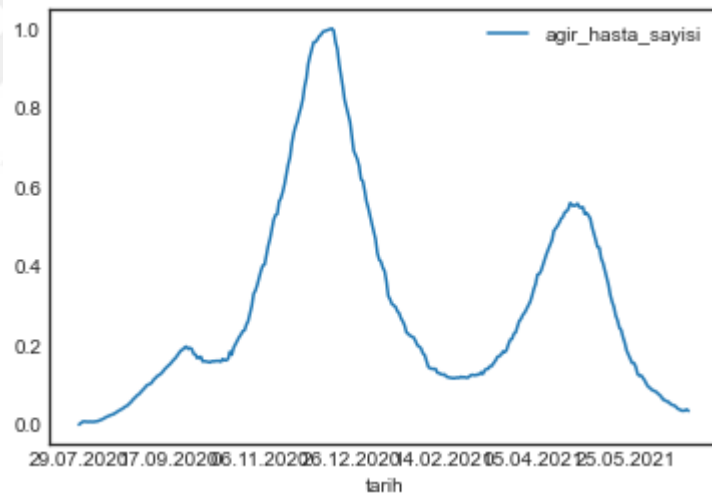
Tablo 3. 4. Ağır Hasta Sayısı Verileri

Tarih	Ağır Hasta Sayısı
29.07.2020	0.000000
30.07.2020	0.003489
31.07.2020	0.007345
01.08.2020	0.008079
02.08.2020	0.007345
03.08.2020	0.006978
04.08.2020	0.007528
05.08.2020	0.007345
06.08.2020	0.006978
07.08.2020	0.008079
08.08.2020	0.008263
09.08.2020	0.009916
10.08.2020	0.011201
11.08.2020	0.013772
12.08.2020	0.016526



Şekil 3. 25. Tek Katmanlı Gerçek Ağır Hasta Sayısı ve Tahmini Ağır Hasta Sayısı Grafiği

Sadece ağır hasta sayısı ekrana çizdirilir.



Şekil 3. 26. Tek Katmanlı Ağır Hasta Sayısı Grafiği

Ortalama mutlak hata bulunur.

MAE: 0.049453

Ortalama kare hata bulunur.

MSE: 0.004666

Kök ortalama kare hatası bulunur.

RMSE: 0.068305

Ortalama mutlak yüzde hata bulunur.

MAPE: 4.945261600613645

Tablo 3. 5. MAE – MSE – RMSE – MAPE Verileri Tablosu

MAE	MSE	RMSE	MAPE
0.049453	0.004666	0.068305	4.945261600613645

3.20. Tek Katmanlı Toplam Vaka Sayısı

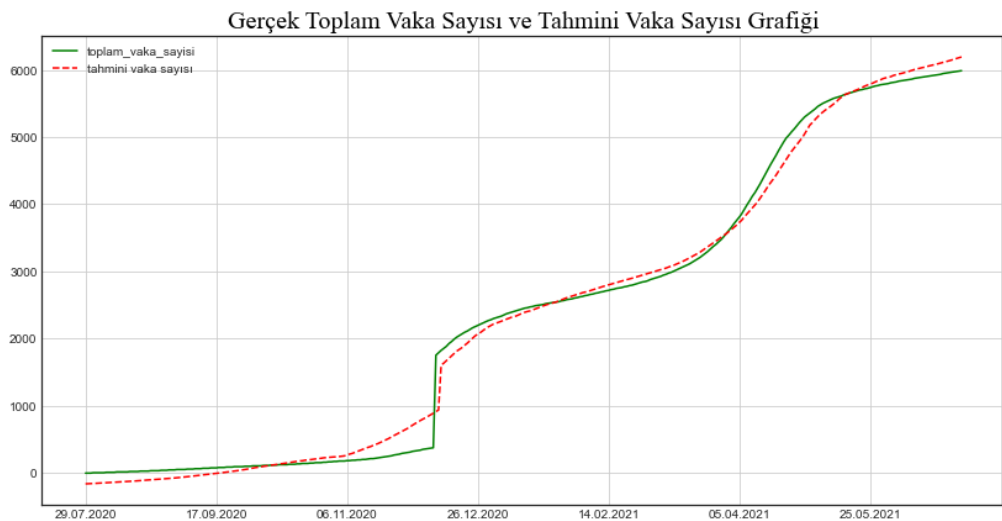
Tek katmanlı ağır hasta sayımızı bulurken kullandığımız kodların ilk kısmı ile aynı şekilde ilerlemektedir. Daha sonra listelediğimiz toplam vaka sayısı ekrana yazdırılır. Daha sonra tek katmanlı ağır hasta sayımızdaki gizli katman ve iterasyon değerlerimizde olduğu gibi aynı şekilde toplam vaka sayısını belirler.

Ortalama Hata Karesi bulunur.

Mean_squared_error (y.values, y_predicted) : 0.0007717970146197763

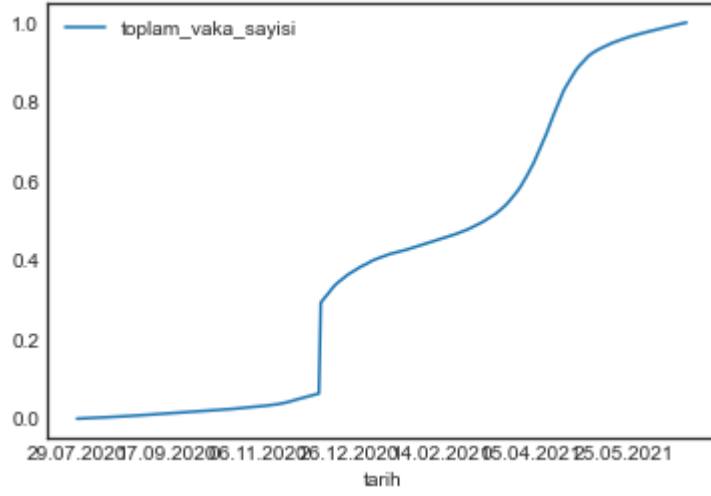
Belirleme katsayısı bulunur.

R2 score (y.values, y_predicted) : 0.9939252827157549



Şekil 3. 27. Tek Katmanlı Gerçek Toplam Vaka Sayısı ve Tahmini Vaka Sayısı Grafiği

Sadece toplam vaka sayısının olduğu grafik ekrana çizdirilir.



Şekil 3. 28. Tek Katmanlı Toplam Vaka Sayısı Grafiği

Ortalama mutlak hata belirlenir.

MAE: 0.021106

Ortalama kare hata belirlenir.

MSE: 0.000772

Kök ortalama kare hata belirlenir.

RMSE: 0.027781

Ortalama mutlak yüzde hata belirlenir.

MAPE: 2.110574189155364

Tablo 3. 6. MAE – MSE – RMSE – MAPE Verileri Tablosu

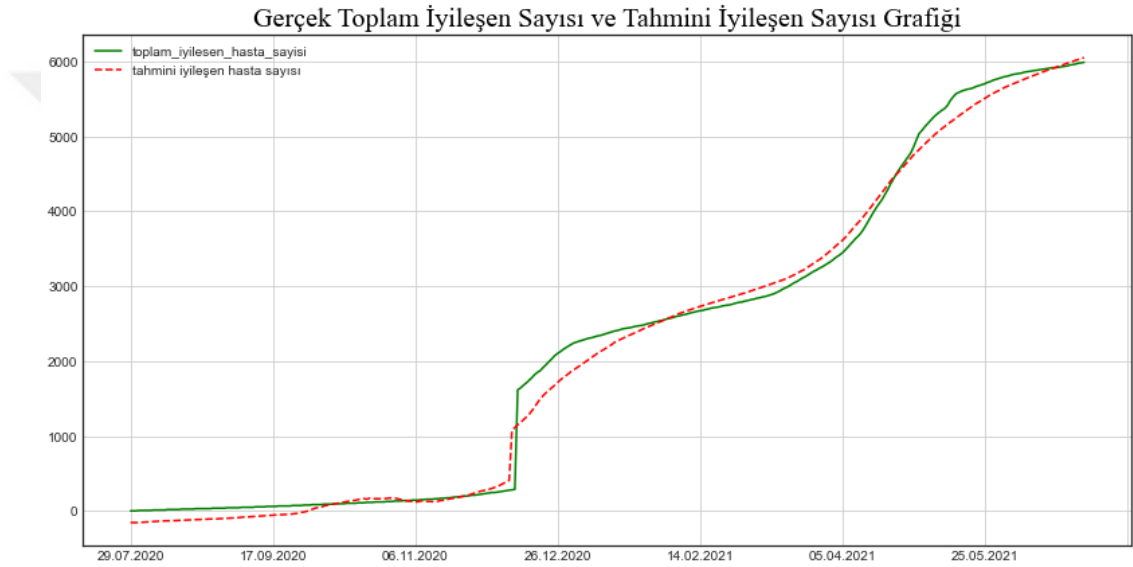
MAE	MSE	RMSE	MAPE
0.021106	0.000772	0.027781	2.110574189155364

3.21. Tek Katmanlı Toplam İyileşen Sayısı

Tek katmanlı ağır hasta sayımızı bulurken kullandığımız kodların ilk kısmı ile aynı şekilde ilerlemektedir. Daha sonra listelediğimiz toplam iyileşen sayısını ekrana yazdırılır. Daha sonra tek katmanlı ağır hasta sayımızdaki gizli katman ve iterasyon değerlerimizde olduğu gibi aynı şekilde toplam iyileşen sayısı belirlenir.

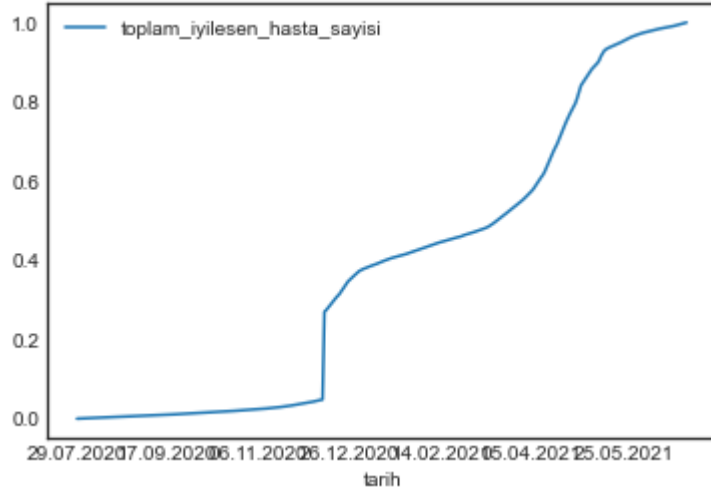
Ortalama Hata Karesi 0.0008307680251631879 olarak bulunur.

Belirleme katsayısı 0.9931771722145035 olarak bulunur.



Şekil 3. 29. Tek Katmanlı Gerçek Toplam İyileşen Sayısı ve Tahmini İyileşen Sayısı Grafiği

Sadece toplam iyileşen sayısının olduğu grafik ekrana çizdirilir.



Şekil 3. 30. Tek Katmanlı Toplam İyileşen Hasta Sayısı Grafiği

Ortalama mutlak hata bulunur.

MAE: 0.021551

Ortalama kare hata belirlenir.

MSE: 0.000831

Kök ortalama kare hata bulunur

RMSE: 0.028823

Ortalama mutlak yüzde hata bulunur.

MAPE: 2.155108298684423

Tablo 3. 7. MAE – MSE – RMSE – MAPE Verileri Tablosu

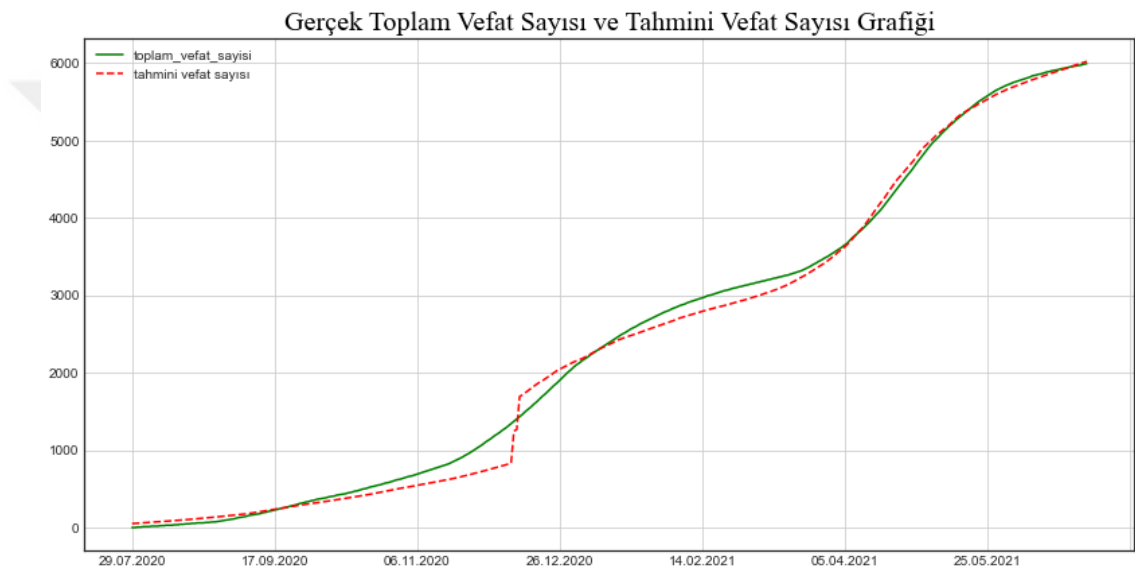
MAE	MSE	RMSE	MAPE
0.021551	0.000831	0.028823	2.155108298684423

3.22. Tek Katmanlı Toplam Vefat Sayısı

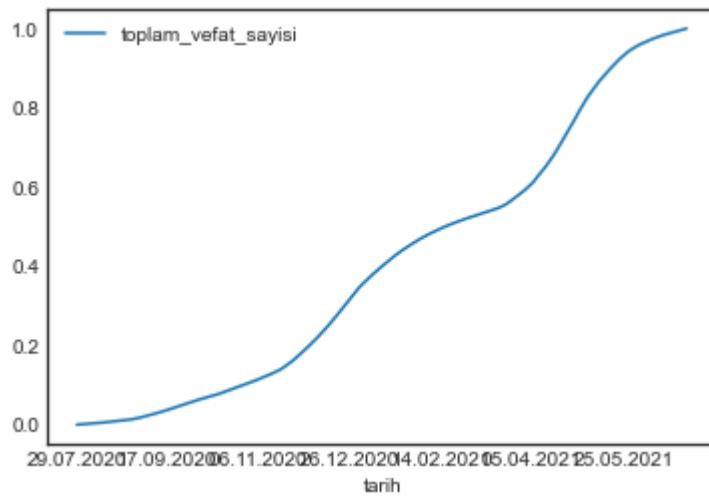
Tek katmanlı ağır hasta sayısı bulunurken kullanılan kodların ilk kısmı ile aynı şekilde ilerlenmektedir. Daha sonra listelenen toplam vefat sayısı ekrana yazdırılır. Daha sonra tek katmanlı ağır hasta sayısı gizli katman ve iterasyon değerleri olduğu gibi aynı şekilde toplam vefat sayısı belirlenir.

Ortalama Hata Karesi 0.0005163407207756335 olarak bulunur.

Belirleme katsayısı 0.9951224752534823 olarak bulunur.



Şekil 3. 31. Tek Katmanlı Gerçek Toplam Vefat Sayısı ve Tahmini Vefat Sayısı Grafiği



Şekil 3. 32. Tek Katmanlı Toplam Vefat Sayısı Grafiği

Ortalama mutlak hata bulunur.

MAE: 0.016827

Ortalama kare hata belirlenir.

MSE: 0.000516

Kök ortalama kare hata belirlenir.

RMSE: 0.022723

Ortalama mutlak yüzde hata belirlenir.

MAPE: 1.682703906085101

Tablo 3. 8. MAE – MSE – RMSE – MAPE Verileri Tablosu

MAE	MSE	RMSE	MAPE
0.016827	0.000516	0.022723	1.682703906085101

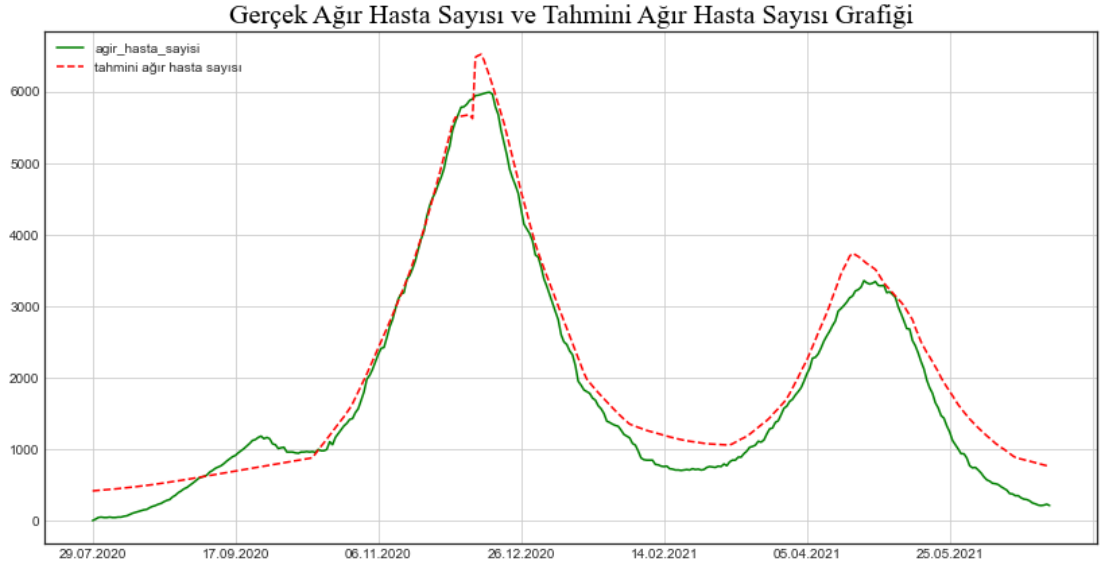
3.23. Üç Katmanlı Ağır Hasta Sayısı

Tek katmanlı ağır hasta sayımızı bulurken kullanılan kodların ilk kısmı ile aynı şekilde ilerlenir. Daha sonra MLPRegressor metodunda gizli katman sayısı **75,75,75** olarak belirlenir ve üç katman kullanılacağı sisteme işlenir.

Ortalama Hata Karesi 0.0028489745759749173 olarak bulunur.

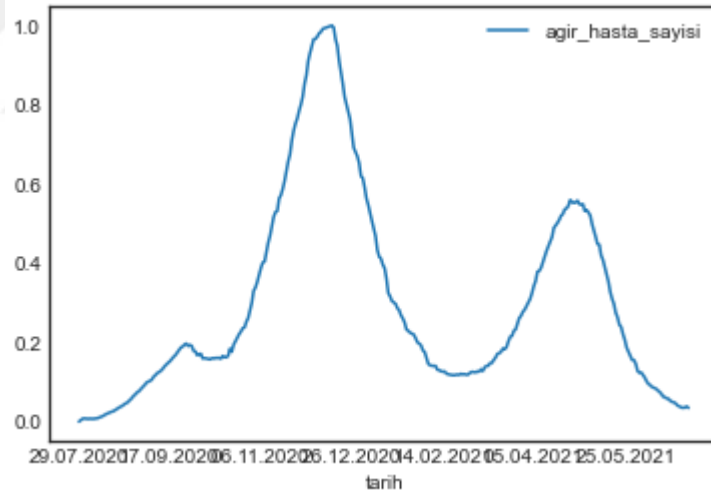
Belirleme katsayısı 0.9574152416400753 olarak bulunur.

Gerçek ağır hasta sayısı ve tahmini ağır hasta sayısı grafiği ekrana yazdırılır.



Şekil 3. 33. Üç Katmanlı Gerçek Ağır Hasta Sayısı ve Tahmini Ağır Hasta Sayısı Grafiği

Sadece ağır hasta sayısı ekrana çizdirilir.



Şekil 3. 34. Üç Katmanlı Ağır Hasta Sayısı Grafiği

Ortalama mutlak hata bulunur.

MAE: 0.045550

Ortalama kare hata bulunur.

MSE: 0.002849

Kök ortalama kare hata bulunur.

RMSE: 0.053376

Ortalama mutlak yüzde hata bulunur.

MAPE: 4.554983986350733

Tablo 3. 9. MAE – MSE – RMSE – MAPE Verileri Tablosu

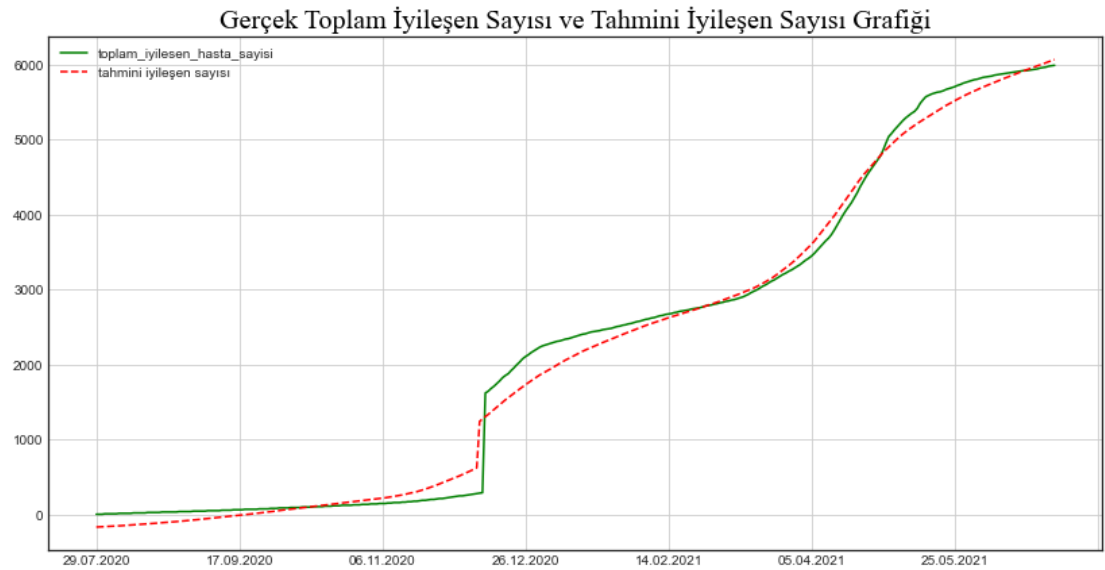
MAE	MSE	RMSE	MAPE
0.045550	0.002849	0.053376	4.554983986350733

3.24. Üç Katmanlı Toplam İyileşen Sayısı

Tek katmanlı toplam iyileşen sayısı bulunacağımız zaman kullanılan kodların ilk kısmı ile aynı şekilde ilerlenir. Daha sonra MLPRegressor metodunda gizli katman sayısı 75,75,75 olarak belirleyerek üç katman kullanılacağı sisteme işlenir.

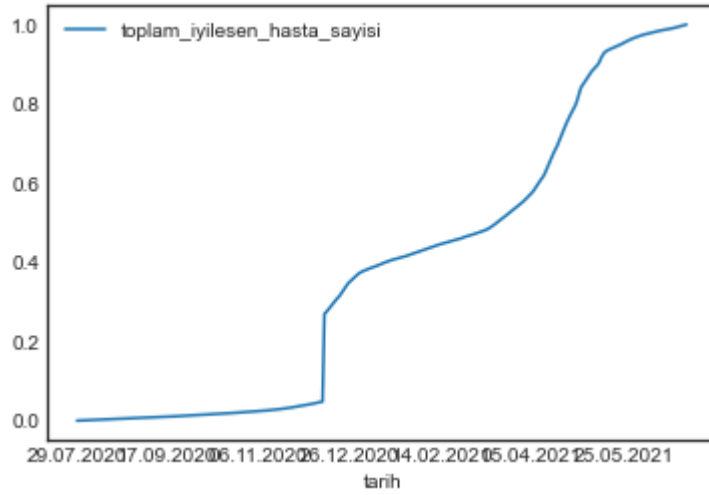
Ortalama Hata Karesi 0.000873865133037908 olarak bulunur.

Belirleme katsayısı 0.992823229674377 olarak bulunur.



Şekil 3. 35. Üç Katmanlı Gerçek Toplam İyileşen Sayısı ve Tahmini İyileşen Sayısı Grafiği

Sadece toplam iyileşen hasta sayısı ekrana çizdirilir.



Şekil 3. 36. Üç Katmanlı Toplam İyileşen Hasta Sayısı Grafiği

Ortalama mutlak hata bulunur.

MAE: 0.022336

Ortalama kare hata bulunur.

MSE: 0.000874

Kök ortalama kare hata bulunur.

RMSE: 0.029561

Ortalama mutlak yüzde hata bulunur.

MAPE: 2.2335681667801026

Tablo 3. 10. MAE – MSE – RMSE – MAPE Verileri Tablosu

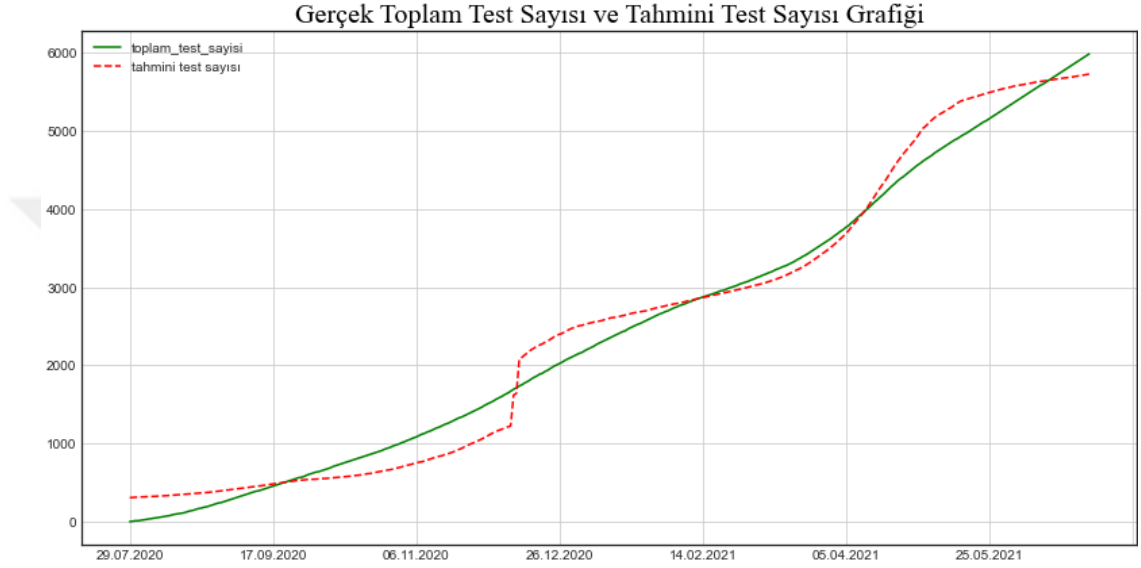
MAE	MSE	RMSE	MAPE
0.022336	0.000874	0.029561	2.2335681667801026

3.25. Üç Katmanlı Toplam Test Sayısı

Tek katmanlı toplam test sayısı belirlenirken kullanılan kodların ilk kısmı ile aynı şekilde ilerlenir. Daha sonra MLPRegressor metodunda gizli katman sayısı 75,75,75 olarak belirleyerek üç katman kullanılacağı sisteme işlenir.

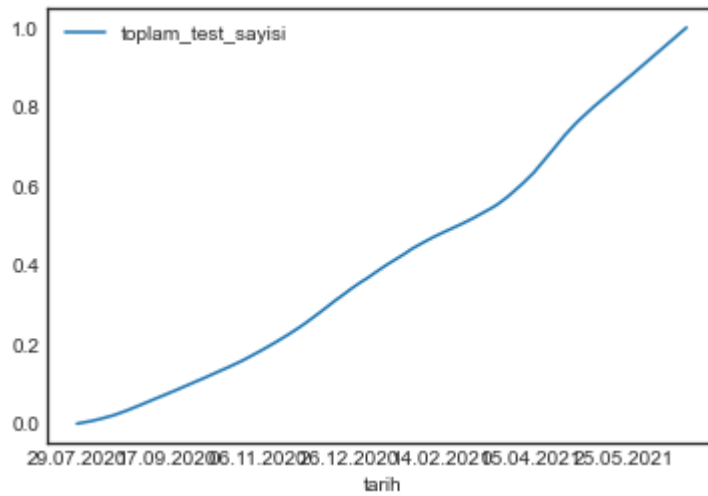
Ortalama Hata Karesi 0.0017358341802037537 olarak bulunur.

Belirleme katsayısı 0.9802551145455027 olarak bulunur.



Şekil 3. 37. Üç Katmanlı Gerçek Toplam Test Sayısı ve Tahmini Test Sayısı Grafiği

Sadece toplam test sayısı ekrana çizdirilir.



Şekil 3. 38. Üç Katmanlı Toplam Test Sayısı

Ortalama mutlak hata bulunur.

MAE: 0.034958

Ortalama kare hata bulunur.

MSE: 0.001736

Kök ortalama kare hata bulunur.

RMSE: 0.041663

Ortalama mutlak yüzde hata bulunur.

MAPE: 3.495794259879534

Tablo 3. 11. MAE – MSE – RMSE – MAPE Verileri Tablosu

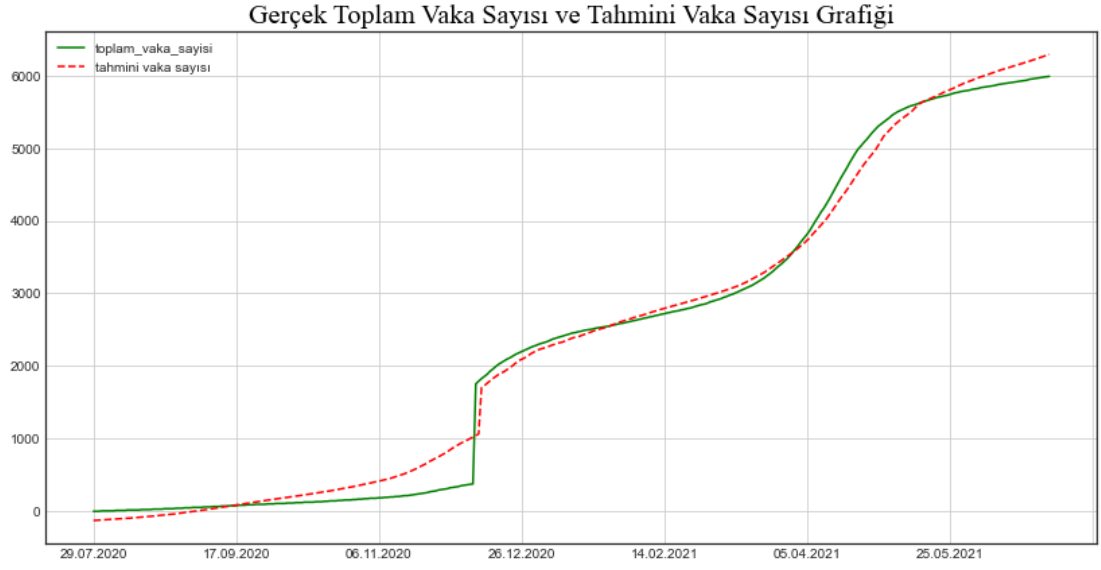
MAE	MSE	RMSE	MAPE
0.034958	0.001736	0.041663	3.495794259879534

3.26. Üç Katmanlı Toplam Vaka Sayısı

Tek katmanlı toplam vaka sayısı hesaplanırken kullanılan kodların ilk kısmı ile aynı şekilde ilerlenir. Daha sonra MLPRegressor metodunda gizli katman sayısı 75,75,75 olarak belirlenir ve üç katman kullanılacağı sisteme işlenir.

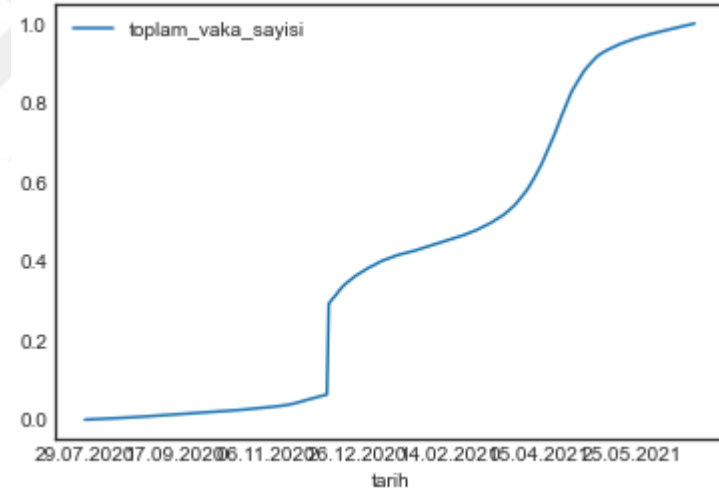
Ortalama Hata Karesini 0.0010539403090523514 olarak bulunur.

Belirleme katsayısı 0.9917045683117639 olarak bulunur.



Şekil 3. 39. Üç Katmanlı Gerçek Toplam Vaka Sayısı ve Tahmini Vaka Sayısı Grafiği

Sadece toplam vaka sayısı ekrana çizdirilir.



Şekil 3. 40. Üç Katmanlı Toplam Vaka Sayısı

Ortalama mutlak hata belirlenir.

MAE: 0.023984

Ortalama kare hata belirlenir.

MSE: 0.001054

Kök ortalama kare hata belirlenir.

RMSE: 0.032464

Ortalama mutlak yüzde hata belirlenir.

MAPE: 2.3983737261111955

Tablo 3. 12. MAE – MSE – RMSE – MAPE Verileri Tablosu

MAE	MSE	RMSE	MAPE
0.023984	0.001054	0.032464	2.3983737261111955

4. BULGULAR

Bu çalışmada python yazılım dili ve kütüphaneleri kullanılmıştır. Evrişimsel sinir ağı çalışmasında yapmış olduğumuz çalışma neticesinde %90 üzerinde doğrulama başarımını yakaladığı görülmektedir. Şekil 3.11.'de görüldüğü üzere eğitim ve doğrulama kayıpları görülmüştür. Yapmış olduğumuz çalışma neticesinde %90 üzerinde doğrulama başarımını yakaladığımız görülmektedir. LSTM çalışmalarında ise ne kadar çok veri ile çalışılırsa o kadar daha iyi sonuçlar alındığı görülmüştür.

5. TARTIŞMA

Bu çalışmada tüm dünyayı etkisi altına alan ve pandemi halini alan Covid19 salgını makine öğrenimi yöntemiyle incelenmiştir. Corona virüs 2019 yılında Çin'in Wuhan eyaletinde bir balık pazarında ortaya çıkmıştır. Balık pazarında ortaya çıkan ve tüm dünyaya yayılarak pandemi halini alan covid daha sonra sürekli başkalaşım geçirerek yeni varyantlar üretmiştir.

Tüm ülkelerin sosyo-ekonomik açıdan ağır darbeler aldığı ve milyonlarca insanın ölümüne neden olan corona virus pandemisi halen daha bitmemiş ve yeni ölümlere neden olmaktadır. Bu pandeminin yarattığı ağır ekonomik darbeler birçok ülkenin sağlık sistemini çökertmiştir. Yeni varyantların ortaya çıkması nedeniyle ülkeler her geçen gün yeni vakalar ile mücadele etmekte ve tam olarak ne zaman biteceği de ön görülememektedir. Ülkelerin günlük vaka sayılarının artışını tahmin etmekte zorlanmaları günlük vaka artışını bilmemeleri sağlık sistemini daha fazla zora sokmuştur.

Birçok ülke yapay zeka sistemlerinden destek alarak gelecekte oluşacak vakaların sayısını tahmin etmeye çalışmaktadır. Makine öğrenimi, yapay sinir ağları gibi sistemleri kendi sistemlerine entegre ederek başarılı bir şekilde kullanan ülkeler bu salgını daha kontrol edilebilir hale getirmiştir.

Bu deneysel çalışmada Evrişimli Sinir Ağı yani CNN ve Katmanlı Sinir Ağları kullanılmıştır. Gizli katmanların sayısının değişmesinin sonuçları karşılaştırılmıştır. Bu çalışmada tek gizli katman ve üç gizli katmana sahip modeller ile çalışıldığında sonuçların farklı olacağı görülmüştür. CNN Modelinde virüsün etkilediği hastaların ve hasta olmayanların akciğer röntgen sonuçları sisteme öğretilmiştir ve test için bazı örnek röntgenler verilmiştir. Sistemin %95 üzerinde doğruluğunda sonuç verdiği görülmüştür. Tek Gizli Katmanlı ve Üç Gizli Katmanlı Yapay Sinir Ağları ile yapılan çalışmada ise gizli katman sayısının artmasının sonuçlarda etkisi olacağı görülmüştür.

Tüm dünyada yapılan makine öğrenimi çalışmalarında verisetlerinin eksik olması başarılı sonuçlar almakta zorluk çıkarmaktadır. Yapay sinir ağları kullanılarak

bir proje geliştirilmek isteniliyorsa eğer verisetlerinin eksiksiz olarak hazırlanması gerekmektedir. Yapay sinir ağlarında kullanılan verilerin boyutu olabildiğince fazla olması gerekmektedir. Büyük veriler ile çalışıldığı takdirde doğru sonuçlar alınmaktadır.

Covid19 pandemisi dünyanın son zamanlarda gördüğü en ağır salgını olmuştur. Ancak birçok ülke zamanında verileri doğru ve eksiksiz yayınlamadı. Eksik olan veriler bu teknolojiye faydalanmayı engelledi. Dünyanın ilk kez karşılaştığı bir salgın olduğu için önceki yıllara ait biriken büyük veri de bulunmamaktadır. Bu da yine yapay sinir ağlarını salgının ilk günlerinde kullanarak başarılı tahminler almayı engellemiştir. Verilerin analizinden de anlaşıldığı üzere verilerin fazla olması sonuçların doğruluk payının yüksek olacağı görülmüştür.

6. SONUÇLAR

Makine öğrenmesi modeli kullanılırken çok fazla veri ile çalışılmazsa sonuçlar doğruluktan uzaklaşabilir. Covid-19 pandemisi de tüm dünyanın yeni karşılaştığı bir hastalık olduğundan ve geçmişten gelen toplanmış veriler olmadığı için pandeminin ilk günlerinde vaka tahminlerinde sıkıntılar tüm ülkelerde görülmüştür. Görüntü işleme ile yapılan çalışmada %95 üzerinde doğrulukla sonuçları elde edebildiğimiz sonucuna ulaşılmıştır.

Dünya genelinde birçok ülke yapay zeka çalışmalarından destek almaktadır. Yapay zeka çalışmalarında ise en çok tercih edilen yazılım dillerinden biri olan Python birçok kütüphaneye sahiptir. Bu çalışmada da python ve kütüphanelerinden yararlanılmıştır. Python yazılımının makine öğrenmesi çalışmalarında çok hızlı sonuçlar verdiği görülmüştür.

7. ÖNERİLER

Bu çalışmada tüm dünyayı etkisi altına alan ve 2 yılı aşkın bir süredir etkisini sürdüren ve halen daha ölümlere neden olan covid19 pandemisinin vaka verileri ışığında makine öğrenmesi üzerinde bir çalışma yapılmıştır. CNN çalışmasında verilerin yeterli olmasıyla tahminin doğruluk oranının %95'in üzerinde olduğu görülmüştür. Tek gizli katmanlı ve üç gizli katmanlı yapay sinir ağlarında katman sayısı, epoch değeri ve diğer verilerin değişmesinin sonuçlardaki etkisi görülmüştür.

Bu çalışmada CNN ile röntgen sonuçları üzerinden ve .csv uzantılı verisetleri üzerinden çalışmalar yürütülmüştür. Veri setlerinin yetersiz olması tüm dünyadaki ülkelerde sıkıntılar yaratmaktadır. Büyük verilerin doğruluk payı yüksek sonuçlar verdiği yapılan CNN çalışmasında da görülmüştür.

Pandeminin ilk günlerinde vaka sayıları yeni olduğu için ülkelerin yeterli veri seti oluşmamıştı. Bu yüzden pandeminin ilk günlerinde yapılmaya çalışılan makine öğrenimi çalışmasında istenilen tahminlerde doğruluk payları düşük çıkmaktadır. Ancak verilerin çoğalması ve ülkelerin vakalara ait verileri şeffaf ve düzenli bir şekilde paylaşmaya başlamasıyla makine öğrenimi çalışmalarındaki tahminlerde doğruluk payları arttığı görülmektedir. Bu yüzden ne kadar büyük veriler ile çalışılırsa o oranda sağlıklı sonuçlar elde edildiği görülmüştür.

KAYNAKLAR

Akpınar, H. (2014). *Data Veri Madenciliği Veri Analizi*. 1. Baskı. Papatya Yayıncılık Eğitim Bilgisayar Sis. San. Ve Tic. A.Ş.

Arslan, İ. (2020). *Python ile Veri Bilimi*. 3. Baskı. Pusula 20 Teknoloji ve Yayıncılık A.Ş.

Ateşli, B., ESEN, O., SÜTLÜ, S., (2021). Epidemiyolojideki Kompartman Modellerinin Eşlenmiş Hamilton Analizi. Gebze Teknik Üniversitesi, Matematik Bölümü, *Int. J. Adv. Eng. Pure Sci.* 2021, 33(2): 265-276, DOI: 10.7240/jeps.796442

Chollet, F. (2021). *Python ile Derin Öğrenme*. 2. Baskı. Buzdağı Yayınevi Tuva Yazılım Eğitim Bilişim Yapımcılık Yayıncılık Hiz. Tic. A.Ş.

Géron, A. (2021). *Scikit-Learn, Keras ve TensorFlow ile Uygulamalı Makine Öğrenmesi*. 1. Baskı. Buzdağı Yayınevi Tuva Yazılım Eğitim Bilişim Yapımcılık Yayıncılık Hiz. Tic. A.Ş.

İçöz, E. (2021). *Covid-19 Pandemi Sürecinde Milli Eğitim Bakanı'nin Twitter Mesajlarının Metin Madenciliği Yöntemiyle İncelenmesi*, Yüksek Lisans Tezi, Akdeniz Üniversitesi

Silahtaroglu, G. (2016). *Veri Madenciliği Kavram ve Algoritmaları*. 3. Baskı. Papatya Yayıncılık Eğitim Bilgisayar Sis. San. Ve Tic. A.Ş.

Yıldırım, Ö. (2020, Ağustos). *Derin Öğrenme Yöntemleri Kullanılarak Covid-19 Vaka Tahmini*, Yüksek Lisans Tezi, Karabük Üniversitesi

URL-1 <https://akademi40.org/makine-ogrenimi-tarihi>, Erişim tarihi: 05.04.2022.

URL-2 <https://ayyucekizrak.medium.com>, Erişim tarihi: 05.04.2022.

URL-3 <https://covid19.saglik.gov.tr/TR-66300/covid-19-nedir-.html>, Erişim tarihi: 05.04.2022.

URL-4 <https://docs.bokeh.org/en/latest>, Erişim tarihi: 05.04.2022.

URL-5 <https://en.wikipedia.org/wiki/Matplotlib>, Erişim tarihi: 05.04.2022.

URL-6 <https://en.wikipedia.org/wiki/Scikit-learn>, Erişim tarihi: 05.04.2022.

URL-7 <https://en.wikipedia.org/wiki/SciPy>, Erişim tarihi: 05.04.2022.

URL-8 <https://geopandas.org/en/stable/about.html>, Erişim tarihi: 05.04.2022.

URL-9 <https://github.com/education454>, Eriřim tarihi: 05.04.2022.

URL-10 <https://tr.wikipedia.org/wiki/Keras>, Eriřim tarihi: 05.04.2022.

URL-11 <https://tr.wikipedia.org/wiki/NumPy>, Eriřim tarihi: 05.04.2022.

URL-12 <https://tr.wikipedia.org/wiki/Pandas>, Eriřim tarihi: 05.04.2022.

URL-13 <https://tr.wikipedia.org/wiki/TensorFlow>, Eriřim tarihi: 05.04.2022.

URL-14 <https://turkiye.ai/kaynaklar/yapay-zeka-zaman-cizelgesi>, Eriřim tarihi: 05.04.2022.



ÖZGEÇMİŞ

Ad-Soyad : Onur KIZILARSLAN

ÖĞRENİM DURUMU:

- **Ön Lisans** : 2003, Süleyman Demirel Üniversitesi, Burdur Meslek Yüksekokulu, Makine, Resim ve Konstrüksiyon Bölümü
- **Lisans** : 2009, Sakarya Üniversitesi, Mühendislik Fakültesi, Makine Mühendisliği Bölümü
- **Yüksek Lisans** : 2016, Trakya Üniversitesi, Fen Bilimleri Enstitüsü, Mühendislik Anabilim Dalı, Makine Mühendisliği Bölümü

MESLEKİ DENEYİM VE ÖDÜLLER:

2003 yılından bu yana çalışma hayatında yer almaktayım. 2014 yılında kurmuş olduğum Facadium Mühendislik adındaki mühendislik firmam ile Türkiye'deki kobilere ve devlet kurumlarına mühendislik alanında danışmanlık ve eğitim hizmetleri vermekteyim. Türk Patent Enstitüsüne kendi yapmış olduğum çalışmalarımın marka, patent ve faydalı model başvuruları bulunmaktadır. Mühendislik ve teknolojiyi hayatımın her alanında kullanmaya ve bu konuda sürekli kendimi geliştirmeye çalışmaktayım. KOSGEB tarafından desteklenen ar-ge projem ile denizlere ve çevreye atılan plastik atıkların 3 boyutlu yazıcıların hammadesine dönüştüren masaüstü filament ekstrüzyon makinesi başarıyla tamamladım.