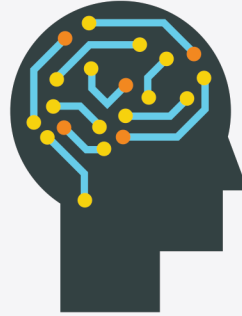




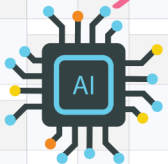
SAÜ MEZUNLAR DERNEĞİ

MÜHENDİSLİKTE YAPAY ZEKA VE UYGULAMALARI 3

Sakarya
2020



YAPAY ZEKA



MAKİNE ÖĞRENMESİ



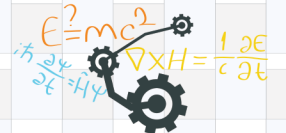
ROBOTİK



YAPAY SINIR
AĞLARI



SİBERNETİK



PROBLEM ÇÖZME

MÜHENDİSLİKTE YAPAY ZEKA VE UYGULAMALARI 3

Editörler

Prof. Dr. Sevinç GÜLSEÇEN

Prof. Dr. Mehmet Melih İNAL

Prof. Dr. Orhan TORKUL

Doç. Dr. İhsan Hakan SELVİ

Doç. Dr. Çiğdem EROL

Dr. Öğr. Üyesi Gültekin ÇAĞIL

Dr. Öğr. Üyesi Zerrin AYVAZ REİS

Dr. Öğr. Üyesi Muhammed Kürşad UÇAR

SAKARYA ÜNİVERSİTESİ, MÜHENDİSLİK FAKÜLTESİ / www.mf.sakarya.edu.tr/

SAKARYA ÜNİVERSİTESİ YAPAY ZEKA SİSTEMLERİ UYGULAMA VE ARAŞTIRMA MERKEZİ /
www.yazsum.sakarya.edu.tr

İSTANBUL ÜNİVERSİTESİ / www.istanbul.edu.tr

YALOVA ÜNİVERSİTESİ / www.yalova.edu.tr

Bu kitap ücretsiz dağıtılmak üzere ülkemizin geleceği için yapılmış bir hizmettir.

1. Baskı, Aralık 2020, SAKARYA

Dizgi, Dr. Öğr. Üyesi Muhammed Kürşad UÇAR - Sakarya Üniversitesi

Kapak, Nukeloveer Studio

e-ISBN: 978-605-2238-24-0

Sakarya Üniversitesi Yayınevi

Sakarya Üniversitesi Yayınları No.: 206

*Gece destan yazan, sabahında devleti için hizmet eden kahraman
milletimize ithaf olunur.*



İçindekiler

1	Artificial Intelligence for Business	13
1.1	Introduction	13
1.1.1	Few examples of Artificial Intelligence	14
1.1.2	Deep Learning	14
1.2	Machine Learning	14
1.3	Speech Recognition	15
1.4	Computer Vision	15
1.5	Robotic Process Automation	15
1.6	Today how AI is integrated into Businesses	16
1.7	AI in Workplace	17
1.8	Artificial Intelligence in E-Commerce	18
1.9	AI in Banking and Finance	19
1.10	AI in Health care	19
1.11	AI in the Automotive Industry	20
1.12	Artificial intelligence in the Insurance sector	21
1.13	AI in sports	22
1.14	AI in Logistic and Supply Chain	22
1.15	AI in hospitality	22
1.16	AI in Human Resource Management	23
1.17	Conclusion	24

2	Multiple Linear Regression	27
2.1	Giriş	27
2.2	Doğrusal Regresyon Analizi	27
2.3	Multiple Linear Regression	28
2.3.1	Multiple Linear Regression Uygulaması	30
2.4	Sınıflandırma	33
2.4.1	K-Nearest Neighbor	33
2.4.2	Support Vector Machine	37
2.5	Sonuç	42
3	Microsoft Azure Machine Learning Studio	43
3.1	Giriş	43
3.2	Regresyon	44
3.2.1	Doğrusal Regresyon	45
3.2.2	Güçlendirilmiş Karar Ağaçları Regresyonu	45
3.3	Microsoft Azure Machine Learning Studio	46
3.3.1	Azure ML Studio Ortamı	46
3.3.2	Azure ML Studio'da İş Akışı	46
3.4	Regresyon Uygulaması: Konut Fiyatı Tahmini	53
3.4.1	Veri Setlerinin Azure ML Studio Ortamına Yüklenmesi	55
3.4.2	Veri Setlerinin Birleştirilmesi	60
3.4.3	Kategorik Niteliklerin Belirlenmesi	67
3.4.4	Niteliklerin İsimlendirilmesi	70
3.4.5	Eksik Verilerin Tamamlanması	70
3.4.6	Örnek Seçimi	74
3.4.7	Yeni Niteliklerin Elde Edilmesi	75
3.4.8	Aykırı Verilerin Ele Alınması	77
3.4.9	Nitelik Dönüşümü	80
3.4.10	Niteliklerin Belirlenmesi	82
3.4.11	Verilerin Normalize Edilmesi	85
3.4.12	Eğitim ve Test Veri Setinin Ayrılması	87
3.4.13	Doğrusal Regresyon Modelinin Eğitilmesi	87
3.4.14	Doğrusal Regresyon Modelinin Değerlendirilmesi	90
3.4.15	Güçlendirilmiş Karar Ağacı Regresyon Modelinin Eğitilmesi	92
3.4.16	Güçlendirilmiş Karar Ağacı Regresyon Modelinin Değerlendirilmesi	94
3.4.17	Modelin Bir Web Hizmeti Olarak Dağıtılması	96
4	Python ile Görüntü İşleme	113
4.1	Giriş	113
4.1.1	Veri Girişi	115
4.1.2	Ön İşleme	115

4.1.3	Öz Nitelik Çıkartma	115
4.1.4	Tanımlama	115
4.2	Uygulama 1	116
4.3	Sonuç	121
4.4	Uygulama 2	123
4.5	Sonuç	127
5	Python Pandas & Pandas-Profilng	133
5.1	Giriş	133
5.2	Pandas Kütüphanesi	133
5.3	Python Pandas Uygulaması	134
5.4	Pandas Profiling Kütüphanesi	140
5.5	Sonuç	144
6	Context ile React Hooks	145
6.1	Giriş	145
6.2	MERN Nedir?	145
6.3	Geliştireceğimiz Uygulama	146
6.3.1	REST Nedir?	146
6.3.2	HTTP Metotları	146
6.4	Genel Kurulum	146
6.4.1	MongoDB Kurulumu	146
6.4.2	Node.js Kurulumu	147
6.5	Uygulama Geliştirme	147
6.5.1	Back-End Kısmı	147
6.5.2	Front-End Kısmı - ReactJS	152
7	Doğal Dil İşleme	165
7.1	Giriş	165
7.2	Doğal Dil İşleme Nedir?	166
7.3	Sentimental Analiz Nedir?	167
7.4	Sentimental Analiz Çalışmalarında Kullanılan Seviyeler	167
7.4.1	Doküman Seviyesinde Yapılan Çalışmalar	168
7.4.2	Cümle Seviyesinde Yapılan Çalışmalar	168
7.4.3	Aspect Seviyesinde Yapılan Çalışmalar	168

7.5	Sentimental Analiz Çalışmalarında Kullanılan Yöntem ve Yaklaşımlar	168
7.5.1	Sözlük Tabanlı Yaklaşım	168
7.5.2	Makine Öğrenmesi - Yapay Sinir Ağları (YSA) Yaklaşımı	168
7.6	Aspect Tabanlı Sentimental Analiz Çalışmalarında Makine Öğrenmesi Tekniklerinin Kullanılması	169
7.6.1	Veri Toplam Süreci	169
7.6.2	Model Oluşturma Süreci	169
7.6.3	Aspectlere Bağlı Kelime Zincirlerinin Oluşturulma Süreci	170
7.6.4	Tokenizer Oluşturma Süreci	170
7.6.5	Model Oluşturma Süreci	170
7.7	Öneriler	174
8	App Inventor ile Anlık Çeviri Yardımcısı	177
8.1	Giriş	177
8.2	Bir Bilgi Sistemi Geliştirmek	177
8.2.1	Planlama	178
8.2.2	Analiz	178
8.2.3	Tasarım	178
8.2.4	Uygulama	178
8.2.5	Destek ve Geliştirme	178
8.3	Yapay Zeka	178
8.3.1	Makine Öğrenmesi	179
8.3.2	Birliktelik Analizi	179
8.3.3	Sınıflandırma Algoritmaları	180
8.3.4	Kümeleme Algoritmaları	181
8.3.5	TTS (Text to Speech) Algoritmaları	182
8.4	App Inventor 2	182
8.5	Uygulama	184
8.5.1	Sesin Alınıp Metne Dönüştürülmesi	184
8.5.2	Metin Çeviri Algoritması Yardımıyla Hedef Dildeki Metne Dönüştürülmesi	185
8.5.3	Çevirisi Yapılmış Metnin Seslendirilmesi	186
8.5.4	Bütünleşik Son Uygulama	187
8.6	Beklentiler ve Öneriler	190
9	Sosyal Medya Verileri ile Duygu Analizi	191
9.1	Giriş	191
9.2	Sosyal Medya Analitiği	192
9.3	Duygu Analizi	193
9.3.1	Duygu Analizi Seviyeleri	195
9.3.2	Duygu Analizi Süreci	196
9.3.3	Duygu Analizi Süreci	197

9.3.4	Duygu Analizinin Önündeki Engeller	202
9.3.5	Türkçe Metinlerde Duygu Analizi	203
9.4	Türkçe Sosyal Medya İçerikleri ile Duygu Analizi	203
9.5	Sonuçlar	208
10	iOS & Derin Öğrenme	213
10.1	Giriş	213
10.1.1	XCode Yapısı	213
10.1.2	Resim Yükleme Uygulamasının Geliştirilmesi	217
10.1.3	Hazır Modelle Sınıflandırma Uygulaması Geliştirmek	221
10.2	CreateMLUI ile Model Oluşturma	224
10.3	CreateML ile Sınıflandırma	231
10.4	Sonuç	238
11	Etmenlerde Stackelberg Oyun Yaklaşımı	241
11.1	Giriş	241
11.2	Çok Etmenli Öğrenme, Oyun Teorisi ve Karar Problemleri	242
11.3	Stackelberg Oyun Teorisi Ve Matematiksel Model	245
11.4	Uygulama	248
11.5	Sonuç	256
12	Yapay Zeka ve Etik	259
12.1	Giriş	259
12.2	Etik	259
12.2.1	Ünlü Filozoflara Göre Etik ve Ahlak	260
12.2.2	Etik Sistemler	260
12.2.3	Etik Türleri	261
12.3	Yapay Zeka	262
12.3.1	Makine Öğrenmesi	262
12.3.2	Derin Öğrenme	262
12.3.3	Denetimli Öğrenme	262
12.3.4	Denetimsiz Öğrenme	262
12.4	Yapay Zeka ve Etik	263
12.5	Yapay Zeka Uygulamaları	264
12.5.1	ROBOBEE	264
12.5.2	WILDCAT	266
12.5.3	ASIMO	267
12.5.4	Philip Dick	267

12.5.5	HRP-4C	268
12.5.6	ICUB	268
12.5.7	Bina 48	269
12.5.8	Sophia	270
12.6	Hukuki Açıdan Yapay Zeka	270
12.6.1	Medeni Hukukta Kişilik Kavramı	271
12.6.2	Yapay Zekanın Kişiliği Sorunu	271
12.7	Sonsöz	272

ÖNCE SÖZ

Bir yıl aradan sonra sözümüze kaldığımız yerden devam ediyoruz. Bu yıl "Mühendislikte Yapay Zeka ve Uygulamaları 3" kitabı ile bir seriye devam etmek istiyoruz. Umarız ki bu tür hizmetler yetiştirdiğimiz öğrencilerimiz için faydalı olur ve her yıl bu kitabın devamını çıkarabiliriz.

Yapay Zeka Yaz Okulu (YAZSUM) ilk olarak 2017 yılında yüzyüze 88 farklı üniversiteden 550'den fazla katılımcı ile Sakarya Üniversitesi ev sahipliğinde gerçekleştirilmiştir. 2018 yılında detaylı içeriklerle bir kez daha hizmet etme fırsatı bulduk. 2020 yılında ise COVID-19 sebebiyle çevrimiçi platformları kullanarak 3500'den fazla katılımcı ile gerçekleştirdik. Eğitim kapsamında 96 saat eğitim verilmiştir. Bu rakam eğitmenlerimizi ve bizleri ziyadesiyle memnun etmiştir.

Pandemi sürecinde teknolojik alt yapılarının önemi bir kez daha ortaya çıkmıştır. Bu süre zarfında sürece hazırlıklı olan kurum ve devletler ilerleyişini hız kesmeden devam ettirmektedir. Ülkemize ve kendimize ilim bakımından yatırım yapmak hayatımızın en önemli adımları olacaktır.

Elimizdeki bu kitap gerek teorik gerekse pratik uygulamalarla size yeni bir yol gösterici olmasını umuyoruz. Yapay zeka oldukça geniş bir konudur. Zifiri karanlıkta her tarafı aydınlatamamakta önümüzü görece kadar kendimize ve çevremize ışık tutmayı umuyoruz.

Işığımızın hiç kaybolmaması dileğiyle.

Editörler
Aralık 2020



artificial
intelligence

1. Artificial Intelligence for Business

Artificial Intelligence for Business

Rashmi Gujrati¹

¹Dean: Management (TIAS-GGSIP University), Head: Entrepreneurship Development Cell, Head: Institute of Innovation Council (MHRD), New Delhi India

Abstract

Businesses are Transforming towards AI is in emerging trends. It is not just a theory that has many applications in practice. Today all around worldwide 30% of companies are using AI for their sale processes and upper income and earnings. In business applications now a day's Artificial intelligence is used extensively with natural language processing, data analytic, and automation. Throughout all these three fields of AI is restricting are cultivating proficiencies and restructuring actions. Everywhere now a day AI has become a buzzword. In purchaser and business spaces there are applications of artificial intelligence, it is from Apple Siri to Google's Deep Mind Siri and Google Deep Mind Siri to the other hand which uses deep learning. This paper aims to understand, that how artificial intelligence is working in various sections in business.

Keywords: Machine learning, deep learning, Applications.

1.1 Introduction

Ultimately, artificial intelligence is software which performs as, like human activities, it seems that human is performing. It has been seen that some of the activities which have been performed by the AI comprise the gathering of data, scheduling group learning, handling data cracking accurate problems, and fetching in discussion or executing deep analysis on huge amounts of data. For the point of this to perform a human activity, computer programming has been programmed with human intelligence therefore it has been for the help of humans. After the use of attention on human responsibilities that need complex stages of aptitude or innovation. Artificial intelligence works as a human then there is a fear in the mind of the manger that this application of AI in commercial will

remove works and it will create employees jobless. But the response is that this will not be going to happen even AI will help to increase the possibility of company staff. With the use of AI, it has proved that a lot of hard work can be completed in minimum time.

Today in the business world, business work has become smarter and faster by the use of Artificial intelligence. It is empowering businesses to improve and modernize the process. Organizations are looking for powerful, cultured solutions and new technology for society to continue in advance. It is appreciable that different technologies are existing under the umbrella of artificial intelligence. Some of the main branches are of artificial intelligence are cognitive computing, robotics, computer vision, deep learning, Machine learning, computer vision, natural language processing, and knowledge, etc. [2] Artificial intelligence is created on that principal where human intelligence is distinct that machines can do mimic easily and succeed in the other works. Artificial intelligence has already created ways from devices in our homes like Alexa to mobile apps, into our everyday lives there are numerous of artificial intelligence examples. This has its own function.

1.1.1 Few examples of Artificial Intelligence

A lot of examples are of artificial intelligence and it is playing their critical role to upgrade the efficiency of the organization. In the business here are the following most popular artificial intelligence.

1. Deep Learning
2. Machine Learning
3. Speech Recognition
4. Computer Vision
5. Robotic Process AI

1.1.2 Deep Learning

Deep learning depends seriously on neural networks to practice nonlinear are perceptive in the form of Artificial intelligence. To activate and create the required results of artificial intelligence which is commonly used in vastly cultured tech applications for the wants of a high level of intelligence. To detect the fraudulent cases Deep learning is hired by the banking and financial institutions. The institutions can analyse speedily and immediately where the fraud has been done. Another in the self-driving car also deep learning has been used to function in a coherent manner and sensors to process the car and to allowing taking a decision where the road moves.

1.2 Machine Learning

Machine learning program lays a significant role it is like a fuel that gives the growth to the organization, it collects the data and to organizes it into the treasured evidence. In Artificial intelligence, machine learning is the most popular form. In big companies, there is a lot of data to manage. When data from customers, employees, and investors are not managed properly it slows down the growth of companies. Machine learning process has been done to teach the robot. Whatever taught it to give better results from time to time.

These models forecast that a person can only spend his limited time on his business, by the use of AI and getting all information they can focus on their marketing and can intermingle their product and make the brand of product in front of the customer and attract more customers with their existing customers.

1.3 Speech Recognition

Speech recognition is that artificial intelligence which progressively transforming pursuit the inquiries (Figure 1.1). These technologies are such as Alex, Cortana, Google Assistant, and Siri which changed the ways of people interact with their devices, as jobs, home, and cars. Through this technology its allow us to talk with a computer or device. Whatever we write to command it answers. To offer the correct quest outcome for operators Artificial intelligence supports to process of Google Voice Search. Another application is Apple's Siri which is used for speech recognition. This speech recognition widely is used in many organizations globally with customer service in Chabot's. Artificial intelligence's introduction was voice-controlled assistant, the voice recognition market or digital assistance, the technology of landscape has changed in the 21st century, by the long history of development and innovation.



Şekil 1.1: Speech Recognition

1.4 Computer Vision

To be intended and examined picture artificial intelligence is the form that empowers computer programs (Figure 1.2). Google has developed its computer vision system due to the extensive and convention of pictures to promote brands, which particular picture is about to analyse and identify. To track a vehicle, computer vision may be used in the transportation sector that breaks traffic rules and regulations. Though, computer vision is an additional advanced process but it can be disorganized with copy processing. Computer vision works to identify to analyse images in the CV process different components of the image.

1.5 Robotic Process Automation

In robotics, the most popular application is Robotic Process Automation (Figure 1.3). The administrative staff can implement and achieve the task properly the software bots have been planned that it can do classically. Resource capacity was created by it and allowed to staff to focus on higher-value activities. Operating activities, detecting fraud, collecting customers Data and ultimately provides expert customer's support that all can be done by Robot. [7]

For RPA business, to conveying smart and flexible analytics for the conversations on mobile devices, by the use of standard messaging tools and for the voice-activated interfaces Chabot's is a perfect example. Chabot has intensely reduced the time of business for collecting the data. It



Şekil 1.2: Computer vision

prepares companies data that has been required in the future and fast-tracking the business steps and updating the analysis ways. The use of robot jobs can be replaced by service customer agents, personal assistance, fast food servers, and social media managers. Overall now a day businesses are transforming towards AI



Şekil 1.3: Robotic Process Automation

1.6 Today how AI is integrated into Businesses

Artificial intelligence has boundlessly influenced supply chain management, manufacturing, and marketing services (Figure 1.4). It was predicted in 2018 by the Harvard business review.

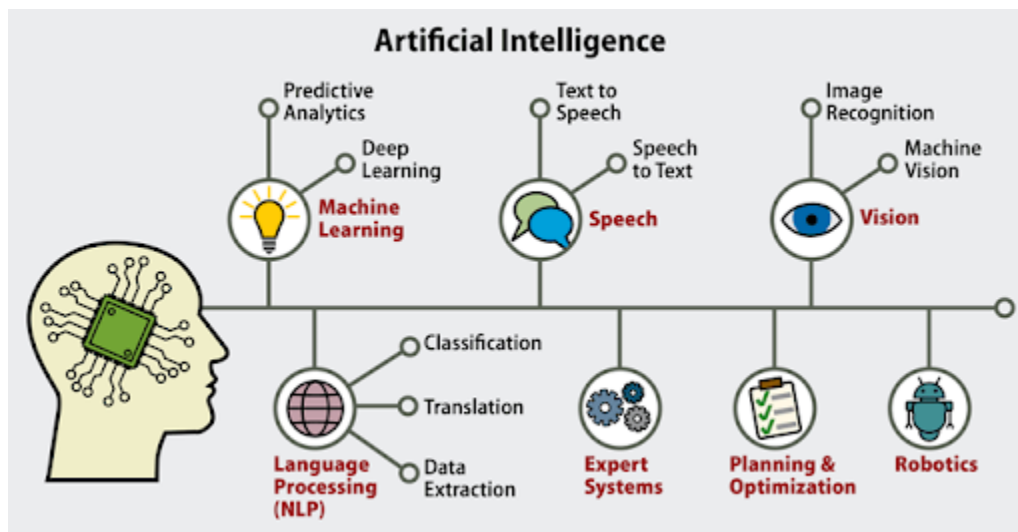
These predictions played out in real-time we are viewing the last two years. For example, social media marketing is running by AI, it is growing very speedily, for the brand it is easy to personalize the customer experience, in this way they connect with their customer and track their marketing success efforts. In the next several years' Supply chain management will also be become the foremost AI-based advances. In real-time Companies are provided with Process intelligence technologies to the truthful and complete insight to display and recover processes. AI is significantly seen in other areas also it is in healthcare and data clearness and security in the healthcare sector. AI helps the patience from starting detection and diagnoses them immediately. From the physician's side, AI

plays the biggest role to help and secure patient's records and flowing the processes?

Transference and safety of Data is an alternative area, in upcoming years where AI is probable going to play important role. When the customer is aware that how much the company is collecting data? To collect which data demand is more transparent, How to use it, how to save it, and how to increase it despite this as Esposito notes, a lot of opportunities are there to increase AI in finance and Banking. Incredible potential for AI-based modernization and with a vast capacity of data are on the obsolete process which still relies, to ensure public safety, AI centres on moral reflections to extensive range roll out. [3]

With science literature dystopias, many people are still related to artificial intelligence and in the daily life artificial intelligence has grown and created its common place in picture. Artificial intelligence has an extensive variety of customs. On a daily bases, maximum of us have interrelated with Artificial intelligence in certain and another procedure. Artificially intelligence has been already disrupted in virtually in every industry and business processes. In business they are becoming imperative to want in maintain competitive superiority. [4]

Some artificial intelligence is here which are implemented in business.



Şekil 1.4: Today how IT Integrated

1.7 AI in Workplace

In the workplace speech recognition technology is also used (Figure 1.5). To increase efficiency It has evolved into incorporating simple tasks, to be performed that has traditionally needed humans beyond tasks. Today Channels, Tools, Content, and so-called solutions are overloaded with current business communication. With harming work-life balance and depriving individually. Through artificial intelligence, business communication can be improved and enhanced. It can focus and increased productivity with the internally and externally and permitted to individually personalized for each professional. Each person will be thankful to power of an intelligent virtual assistant with such, AI personalization and it helps to take care of ordinary or Repeatable task, saving time to understanding their wants and goals. Business processing will improve and grow in the short and long run and also remove their tasks and decrease stress.



Şekil 1.5: AI in Workplace

1.8 Artificial Intelligence in E-Commerce

In the E-commerce business, Artificial Technology is providing a competitive edge and always it is available in any size and budget for companies(Figure 1.6). AI software automatic tags, leverage machine learning contents are organizing for the feature of the image and searching content for labelling. Even the product is branded in size or colour or not, AI is enabling the shoppers to be perfect. Every year visual capabilities are improving through AI. The software can effectively support to the client in discovery of the goods which they want by the first obtaining visual cue by the uploading images. Its imagine that it will rise in the future. AI capabilities and many e-commerce vendors are attractive more refined without the help of humans. Automatically, a new product has becomes more attractive by the use of computer vision and it systemizes when it is added to an e-commerce store.



Şekil 1.6: Artificial Intelligence in E-Commerce

1.9 AI in Banking and Finance

With the use of AI, it has become easy to search the bank and finance fraud as in the earlier it was very difficult to find the fraud in the bank and finance (Figure 1.7). Now it can be found and catches. To detect the fraudulent activities many banks are using various applications of Artificial intelligence. A very large sample of data has been given by the AI software which has included Fraudulent and non-fraudulent purchases. It is proficient to regulate deal is legally founded data or not. To stopping the fraudulent transaction software has become incredible and It is based on what they learned previously. A lot of banks are using the AI process to completing KYC (Known your customer). With short video and selfie AI allows the customers to open the account .Customer emotion is also used to identify by computer vision, Banking services across multiple channel purpose is to personalizes the deliver actionable .through the use of AI customer satisfaction has been increased and have a direct impact on bank revenues and ease to create an account. [5] To decrease the friction of customer banking and financial industry aim is for speech recognition. Human customer services and lower employee costs can be decreased by voice-activated banking. Decrease friction of customer banking and financial industry aim is for speech recognition. Human customer services and lower employee costs can be decreased by voice-activated banking.



Şekil 1.7: Banking and Finance

1.10 AI in Health care

Now Health care is also complex with artificial intelligence apps (Figure 1.8). Mostly this AI is used for the reading of MRI and CT scans it helps nurses and doctors to get good health back and to optimizing radiology measures. Those companies are using artificial intelligence in health care; its big impact is going to be an effect on healthcare in the next five to ten years. AI role is increasing in healthcare due to data and problem is increasing rapidly. Already a lot of life science companies are using AI by the payer and providers of care. Artificial intelligence apps diagnosis treatment, patient engagement, and administrative activities which are complex. AI is performing better than human in the implement factors and the large scale automation of healthcare professional jobs. Secondly AI is

important in the situation of antiseptic operations and it is important to hands-free and immediately the information reached to patients for their security and safety purpose in medical ability. Today transformative mobile app is health. This has been used for the stop of medical errors in healthcare. To collect the data AI presents opportunities for application. Through AI Data has been collected from the patient and improve it innovatively. Uniformity, Dependability, Obviousness is improving the quality of patient safety through AI. Without human interaction and guidance it is not free reign. AI software is used for a decision augment tool.

By the use of AI following benefits are

- information's can be found quickly from medical records, Specifics instruction and reminded in the process to nurse
- how many patients are on the floor and how many units are available, this information nurse can take from administrative
- Through this AI app parents can take guidance from a doctor at home, how to take care of a sick.
- At home, parents can ask for common symptoms of diseases,

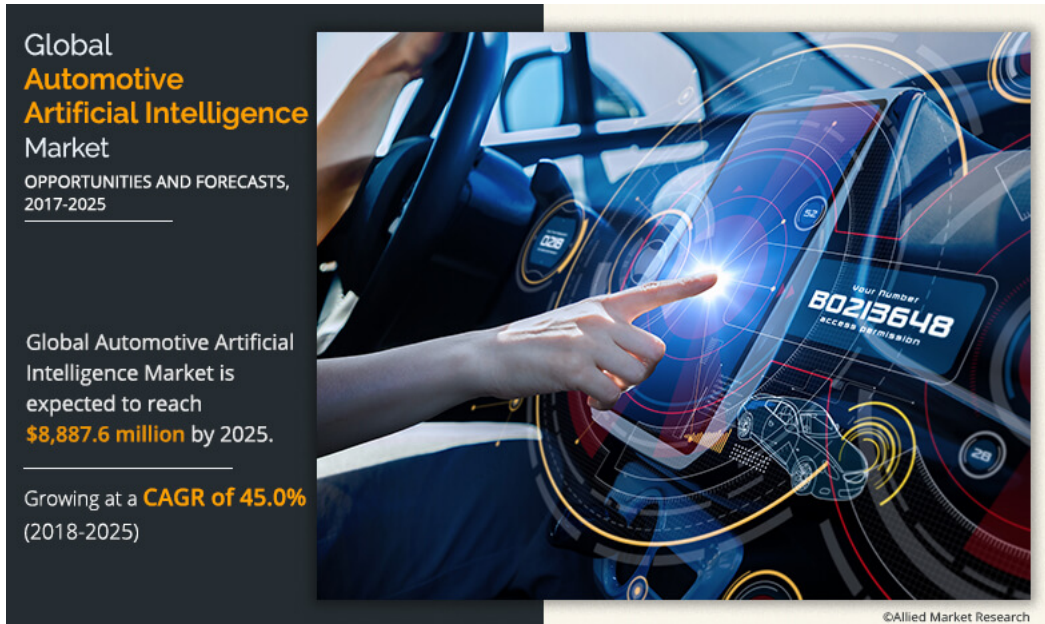


Şekil 1.8: Health care

1.11 AI in the Automotive Industry

Today in the new technology era Artificial intelligence apps are already installed in the computer vision (Figure 1.9). We have an automatic car without gear. The car has various types of functions if it is near to a lane, or crowded area or people are near to car it gives alert alarm to the driver. Secondly, in an automatic cars there has an automatic break in case the driver is driving a car and he is about to sleep its facial recognition system warn the drive. Automatic break system prevents the accident cases and if anyone comes in front of bus or car it has automatic break and it stopped immediately. Driver less car are running of the road lot of companies are using robot in ware houses. In USA already Texla has an automatic car which is running without the driver. They have a computerized function person feed their location where to and the car start and take you to your destination. Already it is behaving like humans. The car knows that where an obstacle is where it has to stop, already have a break system and how to give race etc. It behaves as like a human that all

is Artificial intelligence



Şekil 1.9: AI in the Automotive Industry

1.12 Artificial intelligence in the Insurance sector

In the insurance sector through Artificial intelligence, it is easy to do inspection of the damaged property and another easy to do inspect of the goods losses taking pictures preparing report immediately, it replace soon people in the process (Figure 1.10). If there is any accident case all necessary Reports can be done and claim and do it automatically rather than taking a lot time



Şekil 1.10: Artificial intelligence in the Insurance sector

1.13 AI in sports

In sports players' movement can track by AI. In real-time more complicated games and insights are being created to help the manager perform and to improve the players (Figure 1.11). For better player performance it improves the accuracy of referees, watching game experience. When it comes to analyses huge data is there and no competition between the artificial intelligence and human begin. It is the prediction for future outcomes that the huge capacities of sporting investigation figures are vacant to exploit their precision. With the use of AI today it can perform analysis in the expansive betting market; this has proven especially that is fruitfully offered to an increasingly greedy gambling public. Where a huge digit of sports and gamble kinds are existing.



Şekil 1.11: AI in sports

1.14 AI in Logistic and Supply Chain

AI use in Logistics and supply chain once client facts and analysis combined with it the resistance removes by physical Artificial from the customer experience (Figure 1.12). For many areas of supply chain operation, artificial intelligence empowers the business to drive improvement and to act on consumer data. Urbanization of things and mobile applications has made a consumer hungry for AI. The customer wants delivery in a short time from retailers and retailers to want from the manufacturing centre. The customer wants to ship at night or on the weekend for this way in upcoming years the transport terms "business day" will finish. To lead better accountability in logistic AI is used to count and to track accurately and the quality of the packaging is also checked by AI. The text of the label on packing is also read by OCR (Optical Character Recognition).

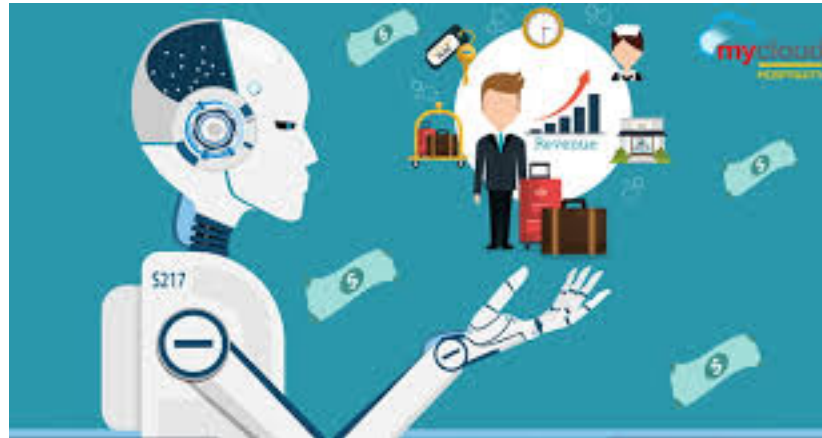
1.15 AI in hospitality

Now a day AI is playing significant role in hotels, resort through this application customer can search his or her interests hotel information location, they can fill their interest in online process and hotel are getting all information of customer what they want when they are going to arrive in their hotel (Figure 1.13). Customers have the interest to read a new paper in the morning they want to use the pool or any medical facility. With the usage of profound neural linkage and copy justify, they



Şekil 1.12: AI in Logistic and Supply Chain

can examine and explain the images, to deliver the highest booking in the real-time AI can do the segmentation to produce liquid and also used to compute active collections for their guest.



Şekil 1.13: AI in hospitality

1.16 AI in Human Resource Management

Forever and drastically changes are going to be by the use of AI and machine learning (Figure 1.14). HR and recruiter work is going on in every company. In business, the HR process is the first process lot of applications and data have to maintain. Physically it takes a lot of time to maintain data. Through AI it has become easy Recruiting process is main in every company and it can be done automatically. HR worker can do their work with freedom with people in the business. Now companies are using artificial intelligence despite hiring the HR team. All HR professional's worst work is going to complete smoothly. Incredible benefits and top quality data has generated automatically through the AI in the 4th industrial revolution AI has taken place on one of the first places of experience.[4] in every aspect of business. AI usage has been established. Today in every

company is using it and all companies are now IT companies where Artificial intelligence is using. If you have to be a leader in business need to enter in this technology.



Şekil 1.14: AI in Human Resource Management

1.17 Conclusion

It is clear that in new horizon artificial intelligence will not individually work with the aggregation of humans. By the use of these techniques in every place, the work competence has improved. Esposito says, He that they don't want to displace worker instead of Artificial intelligence they want the support of AI to help society with this technology [1]. People have a fear of losing the jobs. Entrepreneurs has to create and spread the knowledge that that with the help of AI it will function more effectively. As new technologies are entering the business more improve competence, jobs, new insights, are rising. Esposito says "Understanding creates job, what we can do for better. "In additional developers have to focus on that technology and to develop more this will help people in working style.

At last, we can see now in every sector AI is working very significantly and with the help of AI the work speed has been an increase, hard to hard work and problems can be solved in few seconds. Which is not possible to complete in minutes by the human?

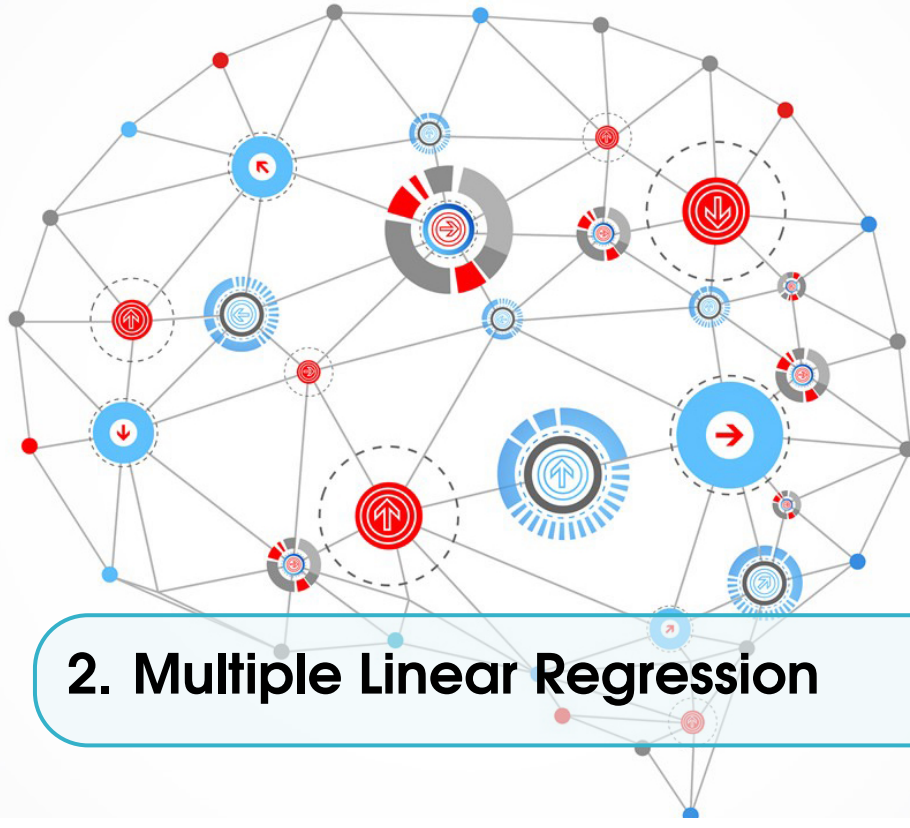
Esposito said that on the behalf of people they will never give the entries of machine are only for the help of people. [1]

Reference

1. <https://blog.dce.harvard.edu/professional-development/business-applications-artificial-intelligence-what-know-2019>

2. https://www.sas.com/en_gb/insights/articles/analytics/applications-of-artificial-intelligence.html

3. <https://blog.dce.harvard.edu/professional-development/business-applications-artificial-intelligence-what-know-2019>
4. <https://www.businessnewsdaily.com/9402-artificial-intelligence-business-trends.html>
5. <https://www.newgenapps.com/blog/ai-uses-applications-of-artificial-intelligence-ml-business>
6. https://www.sas.com/en_gb/insights/articles/analytics/applications-of-artificial-intelligence.html
7. <https://www.ntansa.com/uses-and-application-of-artificial-intelligence-in-business-today>



Multiple Linear Regression, KNN ve SVM ile Makine Öğrenmesi

Hasan Geren¹

¹Orta Doğu Teknik Üniversitesi, Endüstri Mühendisliği, Ankara, Türkiye

2.1 Giriş

Bu bölümde üç tane makine öğrenmesi algoritmasından ve onların uygulamalarından bahsedilecektir. Bu algoritmalarından ilki olan Multiple Linear Regression, bir regresyon algoritması olup numerik verilerin makineler tarafından tahmin edilmesi için kullanılır. K-Nearest Neighbors(KNN) ve Support Vector Machine(SVM) algoritmaları ise sınıflandırma algoritmaları olup verilerin hangi sınıfa ait olduğunun makineler tarafından tahmin edilmesi için kullanılmaktadır. Bu algoritmaların üçü de danışmanlı öğrenme (supervised learning) algoritmaları olup makinelerin tahminde bulunabilmesi için veri setleri kullanılarak eğitilmesi gerekmektedir. Eğitim işleminin gerçekleştirilebilmesi için üzerinde tahminde bulunulacak verilerin bulunduğu veri setinin eğitim seti (training set) ve test seti (test set) olmak üzere ikiye bölünmesi gerekmektedir. Bölme işlemi gerçekleştirildikten sonra eğitim seti kullanılarak eğitilen makineler test seti üzerinde tahminlerde bulunabilecek hale gelmiş olur.

Algoritmalar aracılığı ile veri seti üzerinde işlem yapılmadan önce veri setinin incelenip, ne tür bir makine öğrenmesi algoritması kullanılacağına karar verilmesi gerekmektedir. Kitabın bu bölümünde biri regresyon ve ikisi sınıflandırma algoritmaları olmak üzere üç algoritma hakkında bilgi verilecek ve Kaggle'da ücretsiz paylaşılan bazı örnek veri setleri üzerinde uygulamaları gösterilecektir. Uygulamalar Python programlama dili ve Jupyter Notebook kullanılarak hazırlanmıştır.

2.2 Doğrusal Regresyon Analizi

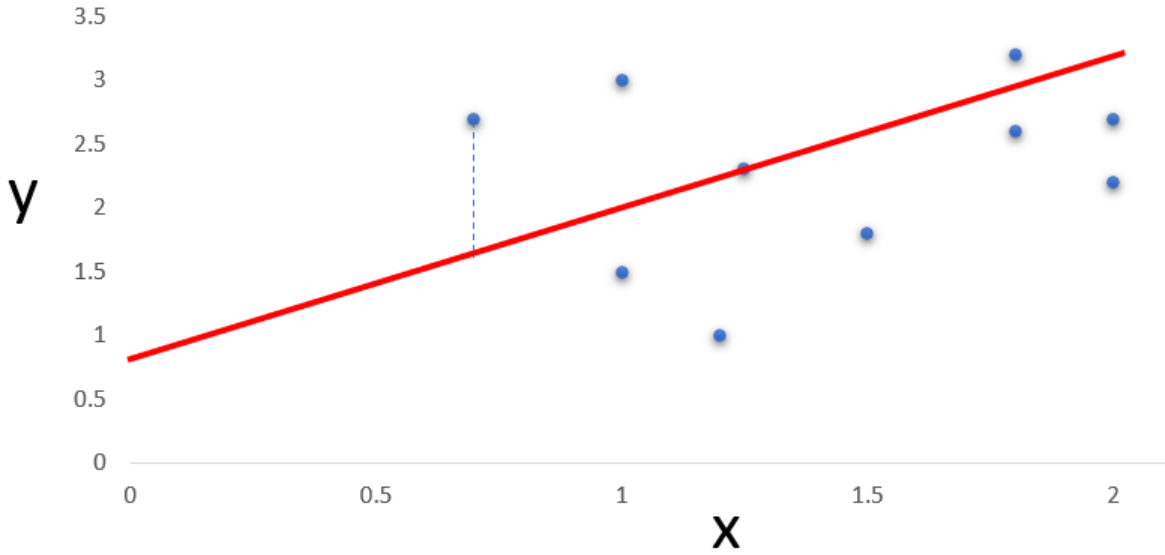
Doğrusal regresyon, bağımsız bir değişkenden bağımlı bir değişkenin değerini hesaplamak için istatistiksel bir prosedürdür (Khushbu Kumari, 2018). Analiz için kullanılan teknik, bağımsız

değişken sayısına bağlı olarak değişmektedir. Tek bir bağımsız değişken için gerçekleştirilen doğrusal regresyon analizi Simple Linear Regression tekniği ile gerçekleştirilirken, iki veya daha fazla bağımsız değişken bulunduğu Multiple Linear Regression kullanılır.

Simple Linear Regression formülü denklem 2.1'deki gibidir.

$$y = b_0 + b_1x \quad (2.1)$$

Burada y değişkeni bizim bağımlı değişkenimiz iken x değişken bağımsız değişkenimizdir. Bu değişkenlere ek olarak b0 sabit değeri ve b1 katsayısı kullanılmaktadır. Sabit değeri burada regresyon doğrusuna eklenen bir değer olarak görev alırken, katsayı değerimiz ise bağımsız değişkendeki değer değişimlerine göre bağımlı değişkenin ne ölçüde değişeceğini belirlemekte kullanılmaktadır(Denklem 2.1).



Şekil 2.1: Simple Linear Regression

Şekil ??'de basit bir Simple Linear Regression örneği görülmektedir. Şekilde de görülebileceği üzere algoritma mavi noktalar ile gösterilen data setlerinin toplam mesafesinin en az olduğu bir doğru oluşturarak tahminlerini buna göre gerçekleştirmektedir.

Regresyon analizlerinin uygulaması sayısızdır ve mühendislik, fizik ve kimya bilimleri, ekonomi, yönetim, biyolojik bilimler ve sosyal bilimler dahil hemen hemen her alanda görülür (Douglas C Montgomery, 2012).

2.3 Multiple Linear Regression

Bir bağımlı değişkeni ve birden fazla bağımsız değişkeni bulunan regresyon modellerine Multilinear Regression denir (Gülden Kaya Uyanık, 2013). Bağımsız değişkenler, kişilik özellikleri, yetenekler veya aile geliri gibi nicel ölçüler olabilir; veya cinsiyet, etnik grup veya bir deneydeki tedavi durumu

gibi kategorik ölçüler olabilirler (Leona S. Aiken, 2012). Algoritmanın denklem 2.2'deki gibidir.

$$y = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n \quad (2.2)$$

Denklem 2.2'de görülen y bağımlı değişken, x değişkenleri bağımsız değişkenler, b_0 sabit değeri ve b değerleri bağımlı değişkenlerin katsayılarıdır. Sabit değeri ve katsayıların görevi aynı Simple Linear Regression'da olduğu gibidir.

Tablo 2.1'de Kaggle'da ücretsiz olarak paylaşılan 50 Startups isimli veri seti üzerinde Multiple Linear Regression değişkenleri gösterilmiştir (Farhan, 2018, April). Tabloda görülen Ar-Ge Harcamaları, Yönetim Harcamaları, Pazarlama Harcamaları ve Eyalet sütunları bağımsız değişkenler yani x değişkenleri iken Kar sütunu ise bağımlı değişken yani y değişkenidir.

Tablo 2.1: 50 Startups

Ar-Ge Harcamaları	Yönetim Harcamaları	Pazarlama Harcamaları	Eyalet	Kar
165349,2	136897,8	471784,1	New York	192261,83
162597,7	151377,59	443898,53	California	191792,06
153441,51	101145,55	407934,54	Florida	191050,39
144372,41	118671,85	383199,62	New York	182901,99
142107,34	91391,77	366168,42	Florida	166187,94

Tablo ??'den yola çıkarak Multiple Linear Regression burada çeşitli alanlarda (Ar-Ge, Yönetim, Pazarlama) yapılan harcamaları ve şirketin kurulduğu eyaleti kullanarak bir Kar tahmininde bulunmak için kullanılacaktır. Fakat regresyon analizleri numerik tekniklerdir ve algoritma yukarıdaki gibi bir veri seti üzerinde kullanılmadan önce kategorik değerlere sahip olan Eyalet sütununun numerik veriye çevirilmesi gerekmektedir.

Kategorik değerlerin numerik değerlere dönüştürülmesi için Dummy Variables tekniği kullanılır. Bu teknik her bir kategori için farklı bir veri sütunu oluşturarak veri satırının ait olduğu kategoriye denk gelen sütuna 1 değerinin ve diğer kategori sütunlarına 0 değerinin verilmesi işlemidir. Dummy variable işlemi şekil 2.2'de görüldüğü gibidir.

Eyalet	New York	California	Florida
New York	1	0	0
California	0	1	0
Florida	0	0	1
New York	1	0	0
Florida	0	0	1

Şekil 2.2: Dummy Variables

Şekil 2.2'de görüldüğü üzere Eyalet sütunu içerisindeki her bir kategori için bir sütun olacak şekilde üç sütuna dönüştürülmüştür. Bu dönüşümden sonra Multiple Linear Regression formülü

denklem 2.3'deki gibidir.

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_4D_1 + b_5D_2 \quad (2.3)$$

Denklem 2.3'de görüldüğü üzere Eyalet sütununa karşılık gelen x_4 değişkeni D değişkenlerine dönüştürülmüştür ve üç tane Dummy Variable bulunmasına rağmen bunların sadece 2 tanesi formülde yer almaktadır. Bunun sebebi her iki Dummy Variable'ın 0 olduğu durumda üçüncünün zorunlu olarak 1 olması ve aksi durumda üçüncünün 0 olmasından dolayı üçüncü Dummy Variable'ın gereksiz olmasıdır.

Kategorik değişken numerik değişkenlere dönüştürüldükten sonra veri seti artık Multiple Linear Regression algoritmasını uygulamak için uygun hale gelmiştir. Multiple Linear Regression emlak fiyat tahminleri, maaş analizleri, yatırım kar tahminleri vs. gibi çeşitli alanlarda kullanılmaktadır.

2.3.1 Multiple Linear Regression Uygulaması

Bu uygulamada Kaggle'da ücretsiz olarak paylaşılan 50 Startups veri seti üzerinde çalışılmıştır (Farhan, 2018, April).

Kütüphanelerin import edilmesi

İlk olarak şekil 2.3'de veri setini Python aracılığı ile okuyabilmek ve düzenleyebilmek için numpy ve pandas kütüphaneleri import edilmiştir.

```
In [1]: import numpy as np
import pandas as pd
```

Şekil 2.3: Kütüphanelerin import edilmesi

Veri setinin Pandas ile import edilmesi

Pandas kütüphanesi kullanılarak veri setimiz bizim belirlediğimiz bir değişkene atanmıştır. Bu uygulamada değişken ismi "dataset" olarak seçilmiştir (Şekil 2.4).

```
In [2]: dataset = pd.read_csv('50_Startups.csv')
```

```
In [3]: dataset
```

```
Out[3]:
```

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94

Şekil 2.4: Veri setinin Pandas ile import edilmesi

Şekil 2.4’de görüldüğü gibi ‘.csv’ formatındaki veri seti pandas aracılığı ile dataset değişkenine atanmıştır ve dataset değişkeninin ilk 5 satırı gösterilmiştir.

Verilerin x ve y değişkenlerine atanması

Veri setini Multiple Linear Regression formülüne uygun formata getirmek için içerisindeki sütunların x ve y değişkenlerine atanması gerekmektedir. Şekil 2.4’de de görüldüğü üzere ilk 4 sütun bağımsız değişken iken 5.sütun yani “Profit” sütunu bağımlı değişkendir. Bundan dolayı ilk 4 sütun x değişkenlerine ve son sütun y değişkenine karşılık gelmektedir. Bu atama işlemi şekil 2.5’deki kod ile gerçekleştirilmiştir.

```
In [4]: X = dataset.iloc[:, :-1].values
        y = dataset.iloc[:, -1].values
```

Şekil 2.5: Verilerin x ve y değişkenlerine atanması

Kategorik veri sütununun Dummy Variable'lara dönüştürülmesi

Şekil ??’daki kod kullanılarak kategorik değerler içeren “State” sütunu Dummy Variable'lara dönüştürülmüştür.

```
In [5]: from sklearn.compose import ColumnTransformer
        from sklearn.preprocessing import OneHotEncoder
        ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [3])], remainder='passthrough')
        X = np.array(ct.fit_transform(X))
```

Şekil 2.6: Kategorik veri sütununun Dummy Variable'lara dönüştürülmesi

Bu dönüşüm işlemi gerçekleştirilirken Sklearn kütüphanesinin ColumnTransformer ve OneHotEncoder fonksiyonları kullanılmıştır. Bu fonksiyonlar ile yapacağımız dönüşüm tekniği “ct” isimli bir değişkene atanmış ve sonrasında bu değişken kullanılarak “fit_transform()” komutu ile dönüşüm gerçekleştirilmiştir(Şekil ??).

```
In [14]: print(X)

[[0.0 0.0 1.0 165349.2 136897.8 471784.1]
 [1.0 0.0 0.0 162597.7 151377.59 443898.53]
 [0.0 1.0 0.0 153441.51 101145.55 407934.54]
 [0.0 0.0 1.0 144372.41 118671.85 383199.62]
 [0.0 1.0 0.0 142107.34 91391.77 366168.42]
```

Şekil 2.7: Dummy Variables

Bu dönüşüm sonrası X değişkenimize baktığımızda kategorik sütunun tamamen silindiğini ve onun yerine her bir kategori için 1 ve 0 lardan oluşan yeni sütunların eklendiği görülmektedir(Şekil 2.7).

Veri setinin eğitim ve test seti olmak üzere ikiye bölünmesi

Dummy Variable dönüşümü yapıldıktan sonra veri seti eğitim ve test seti olmak üzere ikiye bölünmüştür.

```
In [7]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
```

Şekil 2.8: Dummy Variables

Şekil 2.8’deki kod ile X_train, X_test, y_train ve y_test olmak üzere toplam 4 değişken elde edilmiştir. Burada test_size parametresi 0.2 seçilerek veri setinin %80’inin eğitim için, %20’sinin ise test için ayrılması sağlanmıştır.

Multiple Linear Regression modelinin eğitim set üzerinde eğitilmesi

Veri setinin eğitim ve test setlerine bölünmesinden sonra eğitim seti kullanılarak Multiple Linear Regression modeli eğitilmiştir.

```
In [8]: from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

Şekil 2.9: Dummy Variables

Şekil 2.9’deki kod ile Sklearn kütüphanesinin “LinearRegression” fonksiyonu kullanılarak bu fonksiyon “regressor” isimli bir değişkene atanmış ve bu değişken aracılığı ile X_train ve y_train değişkenleri üzerinde eğitim işlemi gerçekleştirilmiştir.

Test seti üzerinde tahminlerin gerçekleştirilmesi

Eğitim işlemi tamamlandıktan sonra test üzerinde tahminler şekil 2.10’deki kod ile gerçekleştirilmiş ve tahmin değerleri sütunda, gerçek değerler sağ sütundaki olmak üzere yazdırılmıştır.

```
In [9]: y_pred = regressor.predict(X_test)
np.set_printoptions(precision=2)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))

[[103015.2  103282.38]
 [132582.28 144259.4 ]
 [132447.74 146121.95]
 [ 71976.1   77798.83]
 [178537.48 191050.39]
 [116161.24 105008.31]
 [ 67851.69  81229.06]
 [ 98791.73  97483.56]
 [113969.44 110352.25]
 [167921.07 166187.94]]
```

Şekil 2.10: Test seti üzerinde tahminlerin gerçekleştirilmesi

Multiple Linear Regression formulündeki sabit ve katsayıların yazdırılması

Algoritmanın içerisindeki sabit ve katsayı değerleri görüntülemek için şekil 2.11’deki kod kullanılmıştır.

Model performansının ölçümü

Son olarak model performansının ölçümü şekil 2.12’deki kod ile yapılmıştır.

Model performansı ölçülürken Sklearn kütüphanesinde “r2_score” fonksiyonu import edilerek test verisinin gerçek ve tahmin değerleri üzerinde uygulanmıştır. Sonuç olarak 0.93 değeri elde edilmiştir yani algoritma verilerin %93’ünde doğru tahminde bulunmuştur(Şekil 2.12).


```
In [11]: print(regressor.coef_)
[ 8.66e+01 -8.73e+02  7.86e+02  7.73e-01  3.29e-02  3.66e-02]

In [12]: print(regressor.intercept_)
42467.52924852117
```

Şekil 2.11: Sabit ve katsayıların yazdırılması

```
In [13]: from sklearn.metrics import r2_score
r2_score(y_test, y_pred)

Out[13]: 0.9347068473282567
```

Şekil 2.12: Model performansının ölçümü

2.4 Sınıflandırma

Sınıflandırma teknikleri, veriyi önceden tanımlanmış sınıf etiketine göre sınıflandıran denetimli öğrenme teknikleridir (Syeda Farha Shazmeen, 2013). Regresyon analizinden farklı olarak sınıflandırma algoritmalarının sonucu numerik bir değer değil, bir sınıf bilgisidir. Sınıflandırma algoritmaları, veri seti içerisinde tanımlanmış çeşitli sınıf bilgilerinden yola çıkarak yeni veriler üzerinde tahminde bulunurlar. Bu bölümde K-Nearest Neighbor ve Support Vector Machine olmak üzere iki tane sınıflandırma algoritmasından bahsedilecek ve uygulamaları gösterilecektir.

2.4.1 K-Nearest Neighbor

K-Nearest Neighbor algoritması çeşitli sınıfları barındıran bir veri seti içerisinde seçilen bir noktanın veya daha sonradan eklenen bir noktanın sınıfını tahmin etmek için kullanılır. Algoritma tahmin işlemini gerçekleştirmek için veri noktalarının birbirlerine olan uzaklık ilişkilerinden faydalanır. Uzaklık hesaplamasına göre k adet veriyi baz alarak tahmini gerçekleştirir. Eğer k veri farklı sınıflara sahipse, algoritma bilinmeyen verilerin sınıfının çoğunluk sınıfıyla aynı olacağını tahmin eder (Kittipong Chomboon, 2015).

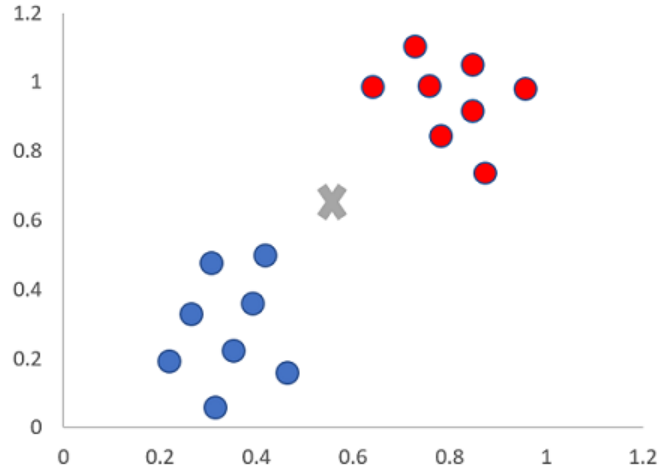
Şekil 2.13’de kırmızı ve mavi renk ile gösterilen iki farklı sınıf ve X ile gösterilen hangi sınıfa ait olduğu belirlenmek istenen veri noktası görülmektedir.

Şekil 2.13’deki X noktasının hangi sınıfa ait olduğunu belirlemek için algoritma şu adımları takip eder;

- X noktasının diğer bütün noktalara olan uzaklığının hesaplanması,
- Hesaplanan uzaklıkların küçükten büyüğe doğru sıralanması,
- X’e en yakın k noktanın içerisinde en fazla hangi sınıftan veri varsa X’in o sınıfa atanması.

Burada “k” kullanıcı tarafından belirlenir ve bu değere göre algoritma farklı sonuçlar verebilir. Uzaklıkların hesaplanmasında yaygın olarak kullanılan tekniklerden biri Euclidean Distance’tır. Herhangi iki nokta arasındaki mesafe hesaplanması için Euclidean Distance formülü denklem 2.4’deki gibidir.

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.4)$$



Şekil 2.13: K-Nearest Neighbor

Denklem 2.4’de x_1 ve y_1 birinci noktanın koordinatlarını simgelerken x_2 ve y_2 ikinci noktanın koordinatlarını simgelemektedir.

K-Nearest Neighbor Uygulaması

Bu uygulamada Kaggle’da ücretsiz olarak paylaşılan Social Network Ads veri seti üzerinde çalışılmıştır (Raushan, 2017, August). Veri setinin ilk 10 satırı tablo 2.2’de gösterilmiştir.

Tablo 2.2: Social Network Ads

Age	EstimatedSalary	Purchased
19	19000	0
35	20000	0
26	43000	0
27	57000	0
19	76000	0
27	58000	0
27	84000	0
32	150000	1
25	33000	0
35	65000	0

Kütüphanelerin import edilmesi

İlk olarak veri setini Python aracılığı ile okuyabilmek ve düzenleyebilmek için numpy ve pandas kütüphaneleri şekil 2.14’deki kod ile import edilmiştir.

```
In [1]: import numpy as np
import pandas as pd
```

Şekil 2.14: Kütüphanelerin import edilmesi

Veri setinin Pandas ile import edilmesi

İmport ettiğimiz pandas kütüphanesi kullanılarak veri setimiz bizim belirlediğimiz bir değişkene atanmıştır. Bu uygulamada değişken ismi “dataset” olarak seçilmiştir(Şekil 2.14).

```
In [2]: dataset = pd.read_csv('Social_Network_Ads.csv')
dataset
```

Out[2]:

	Age	EstimatedSalary	Purchased
0	19	19000	0
1	35	20000	0
2	26	43000	0
3	27	57000	0
4	19	76000	0
...

Şekil 2.15: Veri setinin Pandas ile import edilmesi

Şekil ??’de ‘.csv’ formatındaki veri seti pandas aracılığı ile dataset değişkenine atanmıştır ve dataset değişkeninin ilk 5 satırı gösterilmiştir. Sütunlar incelendiğinde, Age ve EstimatedSalary sütunları bizim bağımsız değişkenlerimiz iken Purchased sütunu bizim sınıf sütunumuzdur. Bu uygulamada yaş ve gelir değerine göre bir ürünü satın alıp almama durumları verilen kullanıcılar üzerinden tahminde bulunulacaktır.

Verilerin x ve y değişkenlerine atanması

Bağımsız değişkenler x değişkenlerine ve sınıf değişkenimiz y değişkenine karşılık gelecek şekilde değişken atamaları şekil 2.16’daki kod ile yapılmıştır.

```
In [3]: x = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```

Şekil 2.16: Verilerin x ve y değişkenlerine atanması

Şekil 2.16’daki kod ile Age ve EstimatedSalary sütununa karşılık gelen birinci ve ikinci sütun x değişkenine atanırken, Purchased sütununa karşılık gelen son sütun ise y değişkenine atanmıştır.

Veri setinin eğitim ve test seti olmak üzere ikiye bölünmesi

Şekil 2.17’deki kod ile veri seti eğitim ve test seti olmak üzere iki farklı gruba bölünmüştür.

Burada “test_size” parametresi 0.25 seçilerek, bölme işlemi sonucunda veri setinin %75’i eğitim ve %25’i ise test seti olacak şekilde bölünmesi sağlanmıştır(Şekil 2.17).

```
In [6]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
```

Şekil 2.17: Veri setinin eğitim ve test seti olmak üzere ikiye bölünmesi

Ölçeklendirme

Şekil 2.18’deki kod ile ölçeklendirme işlemi gerçekleştirilerek birbirinden değer olarak çok farklı olan Age ve EstimatedSalary sütununun algoritmanın sonucunu kötü etkilememesi için aynı ölçek aralığına getirilmesi sağlanmıştır.

```
In [7]: from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Şekil 2.18: Ölçeklendirme

Burada ölçeklendirme işlemi için Sklearn kütüphanesinin “StandardScaler” fonksiyonu kullanılmıştır. Bu fonksiyon ile standardizasyon işlemi Age ve EstimatedSalary sütunlarını barındıran X_train ve X_test değişkenleri üzerinde gerçekleştirilmiştir(Şekil 2.18). Bu işlem ile her iki sütundaki değerler -3 ile 3 arasında olacak şekilde ölçeklendirilmiştir.

K-NN modelinin eğitim seti üzerinde eğitilmesi

Şekil 2.19’deki kod ile Sklearn kütüphanesinin “KNeighborsClassifier” fonksiyonu kullanılarak K-Nearest Neighbor modeli eğitilmiştir.

```
In [8]: from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
classifier.fit(X_train, y_train)
```

Şekil 2.19: K-NN modelinin eğitim seti üzerinde eğitilmesi

Burada “n_neighbors” parametresi 5 olarak seçilerek algoritmanın k değeri 5 olarak belirlenmiştir. “metric” ve “p” parametrelerine ‘minkowski’ ve 2 atanarak ise uzaklık hesaplanması için Euclidean Distance’ın kullanılması sağlanmıştır. Uygun parameter değerleri kullanılarak oluşturulan classifier değişkeni kullanılarak “.fit()” komutu ile eğitim seti üzerinden model eğitilmiştir(Şekil 2.19).

Test seti üzerinde tahminlerin gerçekleştirilmesi

Şekil 2.20’deki kod ile daha önceden eğitilmiş olan model kullanılarak test seti üzerinde tahminler gerçekleştirilmiştir ve sol sütunda tahmin değerleri, sağ sütunda ise gerçek değerler olacak şekilde ilk 10 sonuç gösterilmiştir.

Confusion matrix oluşturulması ve performans ölçümü

Şekil 2.21’deki kod ile algoritmanın verdiği sonuçlar üzerinden confusion matrix oluşturulmuş ve performans ölçümü gerçekleştirilmiştir.

Confusion matrix’in birinci satırında görüldüğü üzere algoritma gerçekte sınıf değeri 0 olan verilerin 64’ünün doğru 4’ünün ise yanlış tahmin edildiği görülmektedir. Aynı şekilde ikinci satırda sınıf değeri 1 olan verilerin 29’unun doğru 3’ünün ise yanlış tahmin edildiği görülmektedir.

```
In [10]: y_pred = classifier.predict(X_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))

[[0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [1 1]
 [0 0]
 [1 0]]
```

Şekil 2.20: Test seti üzerinde tahminlerin gerçekleştirilmesi

```
In [11]: from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
```

```
[[64  4]
 [ 3 29]]
```

```
Out[11]: 0.93
```

Şekil 2.21: Confusion matrix oluşturulması ve performans ölçümü

Accuracy_score fonksiyonu ile hesaplanan performans değerine bakıldığında ise 0.93 yani %93 oranla algoritmanın doğru tahminde bulunduğu görülmektedir(Şekil 2.21).

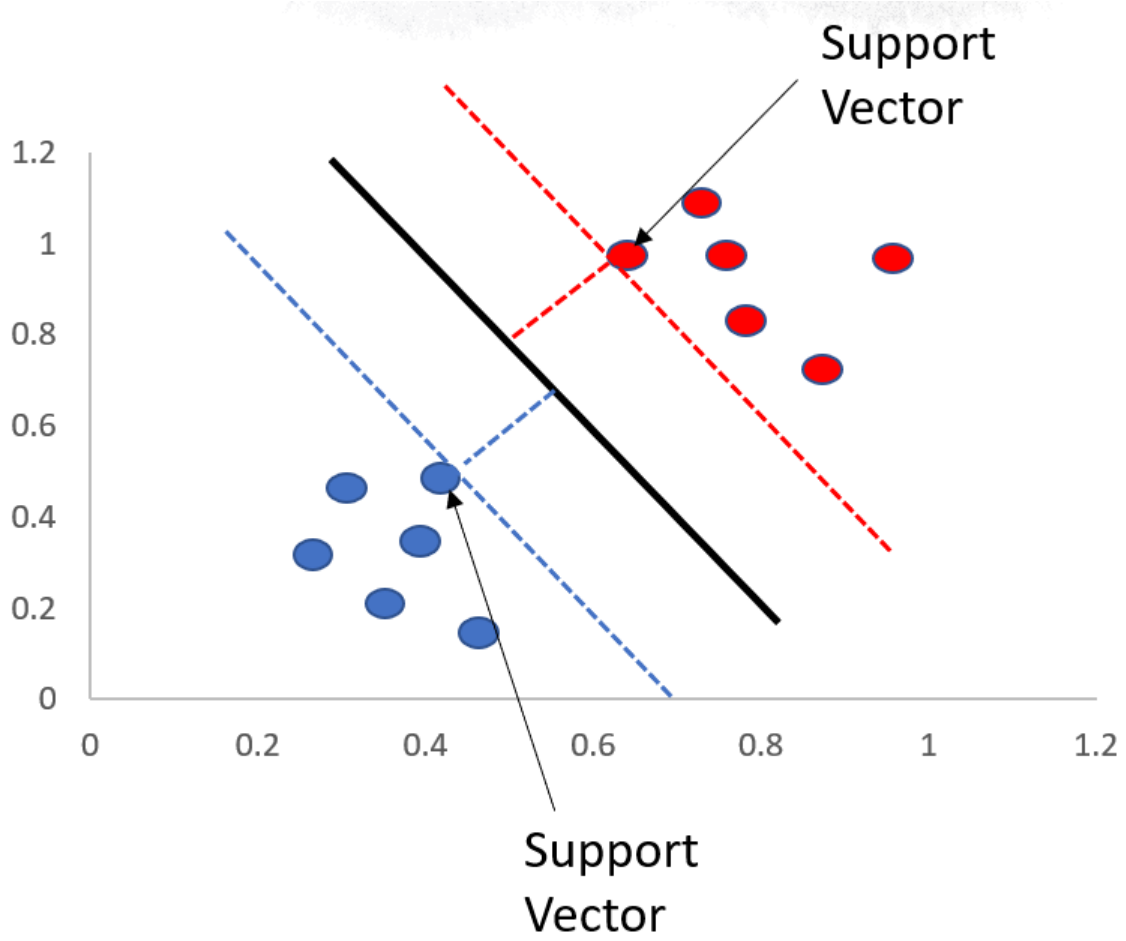
2.4.2 Support Vector Machine

Support Vector Machine, regresyon analizi için de kullanılabilir fakat bu bölümde sadece sınıflandırma için bilgi verilecektir. SVM'ler, "parametrik olmayan" modellerdir (Kecman, 2005). Support Vector Machine, veri seti içerisindeki sınıfları birbirine eşit uzaklıkla bölen bir vektör bulmaya yarar. Support Vector Machine, regresyon analizi için de kullanılabilir fakat bu bölümde sadece sınıflandırma için bilgi verilecektir. Her sınıfın bu vektöre en yakın olan veri noktalarına Support Vector denir. Veri uzayında sınıfları birbirinden ayırmak için sonsuz vektör çizilebilir fakat algoritma bu vektörü Support Vector'lerin vektöre olan uzaklığı maksimum olacak şekilde bulmaya çalışır. Şekil 3.22'de örnek bir veri seti için Support Vector'ler gösterilmiştir.

Şekil 2.22'den yola çıkarak, eğer algoritmadan kendisine verilen elma ve portakal görsellerini sınıflandırması istediğimizi varsayarsak, buradaki Support Vector'ler alışılmışın dışında yada değişik özelliklere sahip elmalar ve portakallara karşılık gelirken vektörlere en uzak noktalardaki veriler standard elma ve portakal özelliklerine sahip verilere karşılık gelmektedir. Bu özelliği sayesinde algoritma gayet başarılı bir sınıflandırma ortaya koymaktadır. SVM, parametre uzayında linear bir sınıflandırıcıdır, ancak linear olmayan bir sınıflandırıcıya kolayca genişletilebilir (S. Amari, 1999).

Support Vector Machine Uygulaması

Bu uygulamada kullanılan veri seti K-NN için kullanılan ile aynıdır ve aşağıda gösterilen kodlar modelin import edilmesi ve eğitilmesi kısımları dışında K-NN algoritması ile aynıdır. Uygulamada hem linear SVM hem de kernel SVM için sonuç elde edilmiştir.



Şekil 2.22: Support Vector'ler

Kütüphanelerin import edilmesi

İlk olarak veri setini Python aracılığı ile okuyabilmek ve düzenleyebilmek için numpy ve pandas kütüphaneleri import edilmiştir(Şekil 2.23).

```
In [1]: import numpy as np
import pandas as pd
```

Şekil 2.23: Kütüphanelerin import edilmesi

Veri setinin Pandas ile import edilmesi

Import ettiğimiz pandas kütüphanesi kullanılarak veri setimiz bizim belirlediğimiz bir değişkene atanmıştır. Bu uygulamada değişken ismi “dataset” olarak seçilmiştir(Şekil 2.24).

```
In [2]: dataset = pd.read_csv('Social_Network_Ads.csv')
dataset
```

Out[2]:

	Age	EstimatedSalary	Purchased
0	19	19000	0
1	35	20000	0
2	26	43000	0
3	27	57000	0
4	19	76000	0
...

Şekil 2.24: Veri setinin Pandas ile import edilmesi

Şekil 2.24’de görüldüğü gibi ‘.csv’ formatındaki veri seti pandas aracılığı ile dataset değişkenine atanmıştır ve dataset değişkeninin ilk 5 satırı gösterilmiştir. Sütunlar incelendiğinde, Age ve EstimatedSalary sütunları bizim bağımsız değişkenlerimiz iken Purchased sütunu bizim sınıf sütunumuzdur. Bu uygulamada yaş ve gelir değerine göre bir ürünü satın alıp almama durumları verilen kullanıcılar üzerinden tahminde bulunulacaktır.

4.2.1.3. Verilerin x ve y değişkenlerine atanması

Bağımsız değişkenler x değişkenlerine ve sınıf değişkenimiz y değişkenine karşılık gelecek şekilde değişken atamaları şekil 2.25’deki kod ile yapılmıştır.

```
In [3]: x = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```

Şekil 2.25: Verilerin x ve y değişkenlerine atanması

Şekil 2.25’deki kod ile Age ve EstimatedSalary sütununa karşılık gelen birinci ve ikinci sütun x değişkenine atanırken, Purchased sütununa karşılık gelen son sütun ise y değişkenine atanmıştır.

4.2.1.4. Veri setinin eğitim ve test seti olmak üzere ikiye bölünmesi

Şekil 3.26'daki kod ile veri seti eğitim ve test seti olmak üzere iki farklı gruba bölünmüştür.

```
In [6]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
```

Şekil 2.26: Veri setinin eğitim ve test seti olmak üzere ikiye bölünmesi

Burada “test_size” parametresi 0.25 seçilerek, bölme işlemi sonucunda veri setinin %75’i eğitim ve %25’i ise test seti olacak şekilde bölünmesi sağlanmıştır(Şekil 2.26).

Ölçeklendirme

Şekil 2.27'deki kod ile ölçeklendirme işlemi gerçekleştirilerek birbirinden değer olarak çok farklı olan Age ve EstimatedSalary sütununun algoritmanın sonucunu kötü etkilememesi için aynı ölçek aralığına getirilmesi sağlanmıştır.

```
In [7]: from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Şekil 2.27: Ölçeklendirme

Burada ölçeklendirme işlemi için Sklearn kütüphanesinin StandardScaler fonksiyonu kullanılmıştır. Bu fonksiyon ile standardizasyon işlemi Age ve EstimatedSalary sütunlarını barındıran X_train ve X_test değişkenleri üzerinde gerçekleştirilmiştir(Şekil 2.27). Bu işlem ile her iki sütundaki değerler -3 ile 3 arasında olacak şekilde ölçeklendirilmiştir.

SVM modelinin eğitim seti üzerinde eğitilmesi

Bu kısımda algoritma hem linear bir sınıflandırma yapması için hem de kernel fonksiyonu kullanılarak linear olmayan bir sınıflandırma yapması için iki farklı şekilde eğitilip uygulamanın devamında her iki durum için de sonuçlar verilmiştir.

Şekil 2.28'deki kod ile Sklearn kütüphanesinin SVC fonksiyonu kullanılarak Linear Support Vector Machine modeli eğitilmiştir. Burada “kernel” parametresi ‘linear’ seçilerek algoritmanın linear bir sınıflandırma yapması sağlanmıştır.

```
In [20]: from sklearn.svm import SVC
classifier = SVC(kernel = 'linear', random_state = 0)
classifier.fit(X_train, y_train)
```

Şekil 2.28: Linear SVM modelinin eğitim seti üzerinde eğitilmesi

Model tanımlanırken “kernel” parametresi ‘rbf’ olarak tanımlanarak algoritmanın linear olmayan bir sınıflandırma yapması sağlanabilir. Algoritmanın linear olmayan sınıflandırma yapması için Kernel SVM kodu şekil 2.29'daki gibidir.

Test seti üzerinde tahminlerin gerçekleştirilmesi

Şekil 2.30'daki kod ile daha önceden eğitilmiş olan model kullanılarak test seti üzerinde tahminler gerçekleştirilmiştir. Tahmin gerçekleştirme işlemi hem linear hem de kernel SVM için aynı şekildedir.


```
In [14]: from sklearn.svm import SVC
classifier = SVC(kernel = 'rbf', random_state = 0)
classifier.fit(X_train, y_train)
```

Şekil 2.29: Kernel SVM modelinin eğitim seti üzerinde eğitilmesi

```
In [16]: y_pred = classifier.predict(X_test)
```

Şekil 2.30: Test seti üzerinde tahminlerin gerçekleştirilmesi

Burada yapılan tahminler `y_pred` isimli bir değişkene atanmıştır(Şekil 2.30).

4.2.1.8. Confusion matrix oluşturulması ve performans ölçümü

Linear SVM için confusion matrix ve performans ölçümü şekil 2.31'deki gibidir.

```
In [23]: from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
```

```
[[66  2]
 [ 8 24]]
```

Out[23]: 0.9

Şekil 2.31: Linear SVM için confusion matrix ve performans ölçümü

Şekil 2.31'de görüldüğü üzere Linear SVM sınıf değeri 0 olan verilerin 66'sını doğru 2'sini ise yanlış tahmin etmiştir. Sınıf değeri 1 olanların ise 24'ünü doğru 8'ini yanlış tahmin etmiştir. Toplamda ise %90 isabet ile doğru tahminde bulunmuştur. Kernel SVM için confusion matrix ve performans ölçümü aşağıdaki gibidir.

```
In [17]: from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
```

```
[[64  4]
 [ 3 29]]
```

Out[17]: 0.93

Şekil 2.32: Kernel SVM için confusion matrix ve performans ölçümü

Şekil 2.32'de görüldüğü üzere Kernel SVM sınıf değeri 0 olan verilerin 64'ünü doğru 4'ünü ise yanlış tahmin etmiştir. Sınıf değeri 1 olanların ise 29'ünü doğru 3'ünü yanlış tahmin etmiştir. Toplamda ise %93 isabet ile doğru tahminde bulunmuştur.

2.5 Sonuç

Kitabın bu bölümünde; Doğrusal Regresyon Analizi ve Sınıflandırmaya dair temel bilgiler verilmiş ve sonrasında Multiple Linear Regression algoritması ile Doğrusal Regresyon Analizi, K-Nearest Neighbor ve Support Vector Machine algoritmaları ile ise Sınıflandırma uygulamaları yapılmıştır. Uygulamalarda, algoritmanın adım adım kodlanması üzerinden ilerlenmiş ve en sonunda algoritmaların ilgili veri seti üzerinde performans ölçümleri yapılmıştır.

Referanslar

Douglas C Montgomery, E. A. (2012). Introduction to linear regression analysis. John Wiley & Sons.

Farhan. (2018, April). 50 Startups. <https://www.kaggle.com/farhanmd29/50-startups>.
Gülden Kaya Uyanık, N. G. (2013). A study on multiple linear regression analysis. *Procedia - Social and Behavioral Sciences*, 234-240.

Kecman, V. (2005). Support Vector Machines – An Introduction. In *Studies in Fuzziness and Soft Computing* (pp. 1-47).

Khushbu Kumari, S. Y. (2018). Linear regression analysis study. *Practice of Cardiovascular Sciences*, 33-36.

Kittipong Chomboon, P. C. (2015). An Empirical Study of Distance Metrics for k-Nearest Neighbor Algorithm. *Proceedings of the 3rd International Conference on Industrial Application Engineering*.

Leona S. Aiken, S. G. (2012). Multiple Linear Regression. In *Handbook of Psychology, Volume 2, Research Methods in Psychology* (pp. 511-543). Wiley Online Library.

Raushan, R. (2017, August). Social Network Ads. <https://www.kaggle.com/rakeshrau/social-network-ads>.

S. Amari, S. W. (1999). Improving support vector machine classifiers by modifying kernel. *Neural Networks*, 783-789.

Syeda Farha Shazmeen, M. M. (2013). Performance Evaluation of Different Data Mining Classification. *Journal of Computer Engineering*, 01-06.

Yazarlar Hakkında

Hasan GEREN, 24 Ekim 1994 yılında İstanbul'da doğmuştur. Lise eğitimini Suat Terimer Anadolu Lisesi'nde tamamladıktan sonra Yıldız Teknik Üniversitesi'nde Endüstri Mühendisliği bölümünde lisans eğitimine başlamıştır. 2019 senesinde lisans eğitimini tamamladıktan sonra Orta Doğu Teknik Üniversitesi Endüstri Mühendisliği bölümünde yüksek lisansa başlamıştır. Aynı sene içerisinde MEB'in YLSY programı ile İsveç'te Cloud Computing üzerine doktora yapmak üzere burs kazanmış ve halihazırda yüksek lisans programına devam etmektedir. Yüksek lisans çerçevesinde Metaheuristic Algoritmalar ve Clustering (Kümeleme) üzerine çalışmaktadır.

İletişim bilgisi: hasan.geren@metu.edu.tr



Azure Machine Learning

3. Microsoft Azure Machine Learning Studio

Microsoft Azure Machine Learning Studio ile Regresyon Uygulamaları

Orhan TORKUL¹, Merve ŞİŞÇİ¹

¹Sakarya Üniversitesi Üniversitesi, Endüstri Mühendisliği, Serdivan, Sakarya, Türkiye

3.1 Giriş

Teknolojinin hızla gelişmesiyle birlikte çeşitli kaynaklardan gelen verilerdeki muazzam büyüme, ek bilgi işlem gücü gerektirmiş, bu da büyük veri setlerini analiz etmek için istatistiksel yöntemlerin geliştirilmesini teşvik etmiştir [1]. Makine öğrenmesi, basit analizin ötesine geçen verilerdeki kalıpları ve eğilimleri otomatik olarak keşfetme sürecidir. Makinelerin görevleri yerine getirirken kazanılan deneyimlerle görevlerde gelişmesini sağlayan derin istatistiksel teknikler içeren yapay zekanın bir alt kümesi olarak da tanımlanabilir [2]. Sistemik, uygulaması kolay makine öğrenmesi çözüm yığınları aracılığıyla gözlemlenen verilerden gerçek dünya süreçlerinin altında yatan karmaşık modelleri öğrenme yeteneği ile anlamlı iş değerinden yararlanma konusunda işletmelerin cazibe merkezi haline gelmiştir [3]. Web araması, spam filtreleri, tavsiye sistemleri, kredi puanlama, dolandırıcılık tespiti, ilaç tasarımı ve diğer birçok uygulamada kullanılmaktadır [2].

Makine öğrenmesi, öğrenme türlerine göre genel olarak denetimli öğrenme ve denetimsiz öğrenme olmak üzere iki kategoride sınıflandırılmaktadır. Denetimli öğrenme terimi, bir modelin bazı etiketli verilerden formüle edildiği makine öğrenmesi görevini tanımlamak için kullanılır [4]. En yaygın kullanılan denetimli öğrenme yöntemleri regresyon ve sınıflandırmadır. Etiketli veri türü kategorik ise, bir sınıflandırma problemi haline gelir ve sayısal ise regresyon problemi olarak bilinir [5]. Regresyonun amacı, gözlemlenen bir dizi değişkenden belirli bir sonucu tahmin etmektir. Gelir, laboratuvar değerleri, test puanları, bir şehrin sıcaklığı, hisse senedi fiyatı veya nesnelerin sayısı gibi sayısal verileri tahmin etmek için yaygın olarak kullanılmaktadır [1, 6]. Denetimsiz öğrenme teknikleri ise, etiketlenmemiş verilerde örüntüler bularak modelleri tahmin ederler [4]. Denetimsiz öğrenmenin en temel örneği, kümelemedir, yani bir dizi nesneyi benzerliğe göre gruplama görevidir

[7].

Makine öğrenmesinde kullanılan yöntemler sürekli olarak araştırılmakta ve çeşitli alanlardan alınan gerçek hayat verileriyle birlikte bu yöntemleri uygulamak için gerekli olan bilgisayar teknolojilerine göre değerlendirilmektedir [8]. IBM Watson, Amazon Web Service (AWS), Google TensorFlow ve Microsoft Azure Machine Learning Studio gibi bulut bilgi işlem ve analitik hizmetleri, diğerlerinin yanı sıra, daha geniş kullanıcılar için karmaşık bulut tabanlı makine öğrenmesi yeteneklerine kolay erişim sağlayarak çeşitli sektörleri etkileme potansiyeline sahiptir [9]. Microsoft Azure Machine Learning Studio (Azure ML Studio) veri bilimi, tahmine dayalı analitik, bulut bilgi işlem ve verilerin buluşabileceği bir ortam sağlar [6]. Bir geliştiricinin tahmine dayalı analitik modelleri (çeşitli veri kaynaklarından eğitim veri kümelerini kullanarak) oluşturmak ve ardından bu modelleri bulut web hizmetleri olarak tüketimi amacıyla kolayca dağıtmak için kullanabileceği bir hizmettir [10]. Regresyon, sınıflandırma ve kümeleme gibi yaygın senaryolar için önceden oluşturulmuş birçok algoritma dahil edilmiştir ve özel kodların eklenebileceği ve diğer modüllere bağlanabileceği R ve Python komut dosyası modülleri aracılığıyla genişletilebilirlik sağlanmıştır [11].

Güçlü makine öğrenmesi uygulamaları oluşturmak ve tutarlı, eyleme dönüştürülebilir sonuçlar elde etmek amacıyla geliştirilen sağlam bir makine öğrenmesi çözümü, makine öğrenmesinin etkileşimli modellemesine uygun olmasının yanı sıra, veri alımı, görselleştirme ve analiz etme, önışleme, sistem entegrasyonu ve çalışma zamanında dağıtım ve bakım için güçlü ekosistem desteğinde mükemmel olan bir geliştirme platformu gerektirir [3]. Bu çalışmada, Azure ML Studio'ya kısa bir giriş yapılmakta ve Azure ML Studio kullanımı için makine öğrenmesi yöntemlerinden biri olan regresyon açıklanmaktadır. Regresyonun Azure ML Studio üzerinde uçtan uca bir örneğinin gösterilmesi amacıyla konut fiyatı tahmini veri seti üzerinde çeşitli veri ön işleme işlemleri gerçekleştirildikten sonra Doğrusal Regresyon ve Güçlendirilmiş Karar Ağacı Regresyon algoritmaları kullanılarak iki model geliştirilmektedir. Modellerin performansları karşılaştırıldıktan sonra örnek olarak en iyi performansı gösteren modelin Azure ML Studio üzerinde bir web hizmeti olarak dağıtılması gösterilmektedir.

3.2 Regresyon

Regresyon, bağımlı bir değişken ile bir veya daha fazla bağımsız değişken arasındaki ilişkiyi modellemek ve analiz etmek için kullanılan denetimli bir öğrenme yöntemidir [12]. Yanıt değişken, bağımlı değişken veya hedef değişken tahmin edilmek istenilen değişken için kullanılırken, açıklayıcı değişkenler, bağımsız değişkenler, özellikler veya nitelikler yanıtı tahmin etmek için kullanılan değişkenler için kullanılmaktadır [13]. Birkaç hedef değeri üretmek için bir dizi parametrenin bir şekilde birleştirildiği bazı verilerden regresyon modelleri oluşturulur. Model aslında hedef değer ile modelin parametreleri arasındaki ilişkiyi açıklar ve modelin parametreleri için değerler verildiğinde bir hedef değeri tahmin etmek için kullanılabilir [4]. Regresyon modelleri için girdi değişkenleri sayısal veya kategorik olabilir. Bununla birlikte, bu algoritmalarda ortak olan, çıktının (veya yanıt değişkeninin) sayısal olmasıdır [6, 12].

Pek çok regresyon türü bulunmaktadır. Yalnızca bir açıklayıcı değişken varsa ve yanıt değişkeni ile bağımsız değişken arasındaki ilişki doğrusal ise, doğrusal bir model uygulanabilir. Bununla birlikte, birden fazla açıklayıcı değişken varsa, çoklu doğrusal regresyon yöntemi kullanılmalıdır. İlişki doğrusal olmadığında, açıklayıcı ve yanıt değişkenleri arasındaki ilişkiyi modellemek için doğrusal olmayan bir model kullanılabilir [12]. En yaygın kullanılan regresyon tekniklerinden bazıları doğrusal regresyon, karar ağaçları, sinir ağları ve güçlendirilmiş karar ağacı regresyonunu içerir. Bu bölümde konut fiyatı tahmininde kullanılan doğrusal regresyon ve güçlendirilmiş karar

ağacı yöntemleri incelenmektedir.

3.2.1 Doğrusal Regresyon

En eski tahmin tekniklerinden birisidir. Regresyon yöntemleri, ilişkinin hem büyüklüğü hem de belirsizliği dahil olmak üzere girdiler ve hedef arasındaki ilişkiyi kesin terimlerle ölçtüğünden tahmin için yaygın olarak kullanılmaktadır [1]. Model basit ve yorumlanabilir olduğundan, değişkenler arasındaki çakarsama ilişkilerini anlamaya olanak tanır [13]. Basit bir doğrusal regresyon modeli, doğrusal bir fonksiyondur. Yalnızca bir girdi değişkeni varsa, doğrusal regresyon modeli verilere uyan en iyi çizgidir. İki veya daha fazla girdi değişkeni için, regresyon modeli, temeldeki verilere uyan en iyi hiper düzlemdir [6]. İki den fazla bağımsız değişkene sahip çoklu doğrusal regresyon modeli, Denklem 1'de verilen yapıda bir formüldür [6, 13]:

$$Y = b_1X_1 + b_2X_2 + \dots + b_nX_n + \alpha + \varepsilon \quad (3.1)$$

Burada, Y yanıt değişkeni yani tahmin edilmek istenilen değişken, X_1, X_2, \dots, X_n çeşitli bağımsız değişkenler ve b_1, b_2, \dots, b_n , bu bağımsız değişkenlerle ilişkili eğimler yani bağımsız değişkenlerin katsayılarıdır. α , regresyon çizgisinin kesişme noktası olan bir sabittir. ε ise X_1, X_2, \dots, X_n bağımsız değişkenleri tarafından açıklanamayan, yanıt değişkeniyle ilişkili hata veya gürültüdür.

Doğrusal regresyon, belirli bir veri seti için en iyi model katsayılarını bulmak amacıyla en küçük kareler veya gradyan iniş yöntemlerini kullanır [6]. 100'den az özellik ve birkaç bin veri noktası olduğunda tahmin elde etmeye çalışmak için uygun bir yöntemdir [5]. Ancak, aykırı değerlere ve çapraz korelasyonlara karşı çok hassastır [12]. Eğer çok büyük aykırı değerler varsa, doğrusal regresyon gerçekleştirilmeden önce bu değerler işlenmelidir [5].

3.2.2 Güçlendirilmiş Karar Ağaçları Regresyonu

Güçlendirilmiş karar ağaçları regresyonu, genel girdi-çıkı verileri ile girdi değişkenlerinin yorumlanabilirliği arasındaki karmaşık ilişkileri tanımlama becerisi nedeniyle popüler bir makine öğrenmesi yöntemi olan karar ağacı modeline dayanmaktadır [14]. Güçlendirilmiş karar ağaçları, bir topluluk modeli biçimidir [6]. Güçlendirme adı verilen istatistiksel bir tekniği birleştirerek geleneksel karar ağacı yaklaşımını geliştirir. Bu tekniğin ana fikri, optimize edilmiş bir model oluşturmak yerine tek bir "güçlü" fikir birliği modeli oluşturmak için bir dizi "zayıf" modeli bir araya getirmektir [14].

Bireysel karar ağaçlarının her biri zayıf tahmin unsurları olabilir. Bununla birlikte, birleştirildiklerinde üstün sonuçlar üretirler [6]. Güçlendirilmiş karar ağaçları regresyonunda, mevcut artıkları en aza indirerek yeni karar ağaçları sırayla oluşturulur [14]. Her yinelemeli adımda, en dik gradyanla tanımlanan belirli kayıp fonksiyonunu optimize etmek için yeni bir regresyon ağacı eğitilir. Diğer makine öğrenme algoritmalarının sahip olmadığı özel kayıp fonksiyonları aracılığıyla belirli niteliklere sahiptir [15].

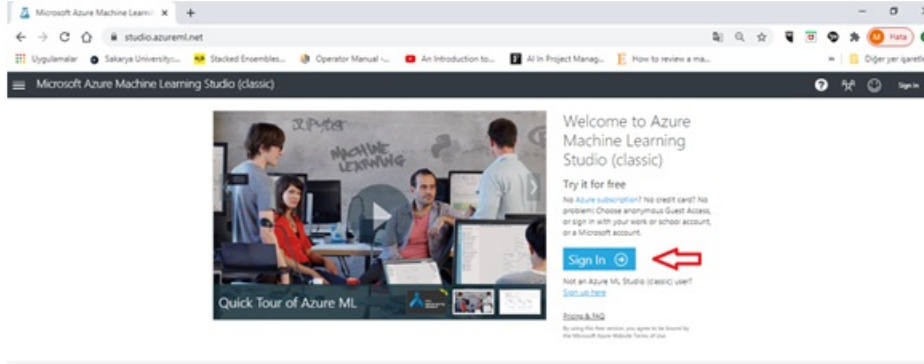
Güçlendirilmiş karar ağacında izlenilmesi gereken iki temel parametre bulunmaktadır: modeldeki ağaç sayısını ifade eden güçlendirme yinelemelerinin sayısı ve modeldeki değişkenler arasında izin verilen etkileşim miktarını belirleyen her bir karar ağacındaki yaprak sayısı. Her ağaçtaki yaprak sayısı 2 olarak ayarlanırsa, herhangi bir etkileşime izin verilmez. 3 olarak ayarlanırsa, model en fazla iki değişkenin etkileşiminin etkilerini içerebilir. Üzerinde çalışılan veri seti için en uygun olanı bulmak amacıyla farklı değerler denenmesi gerekmektedir. Diğer taraftan, çok sayıda ağaç, hataları azaltır, ancak aşırı uydurmaya neden olabilir. Tüm sınıflandırıcıları eğittikten sonra, son adım, son bir tahminde bulunmak için sonuçlarını birleştirmektir. Sonuçları birleştirmek için basit çoğunluktan ağırlıklı çoğunluk oylamasına kadar değişen birkaç yaklaşım vardır [6].

3.3 Microsoft Azure Machine Learning Studio

Azure Machine Learning Studio, tahmine dayalı analitik modelleri kolayca oluşturmaya, test etmeye ve dağıtmaya olanak tanıyan etkileşimli bir görsel çalışma alanı sağlar [6]. Bu bölümde Azure ML Studio ortamına giriş yapılmakta, Azure ML Studio ortamı ve yeni bir deney ortamı tanıtılmakta ve Azure ML Studio'da bir makine öğrenmesi için genel olarak gerçekleştirilen iş akışı anlatılmaktadır.

3.3.1 Azure ML Studio Ortamı

Azure ML Studio kullanmak için geçerli bir Azure aboneliği zorunlu değildir. Azure aboneliği bulunuyorsa, tam özellik setinin kullanım kilidini açmak için kullanılabilir [16]. Herhangi bir Azure aboneliği yoksa Microsoft Azure ML Studio ortamına giriş yapılabilmesi için <https://signup.live.com> adresinden elde edilebilecek bir Microsoft hesabına (örneğin; outlook.com, live.com veya hotmail.com adresi) ihtiyaç duyulmaktadır. Azure Machine Learning Studio'ya Şekil 1'de gösterildiği gibi bir tarayıcı oturumunda Microsoft hesabı ve şifresiyle studio.azureml.net adresinden 'Sign In' butonuna basılarak giriş yapılabilir. Bu şekilde Azure ML Studio bazı kısıtlamalar ile kullanılabilir.



Şekil 3.1: Microsoft Azure ML Studio giriş sayfası

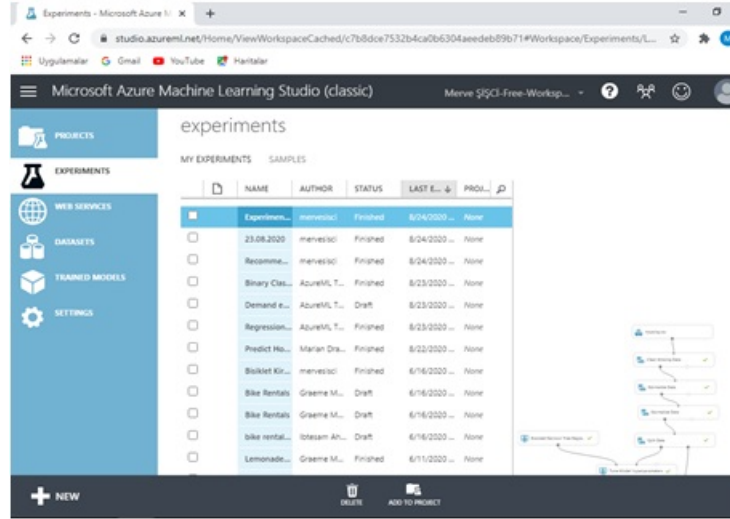
Microsoft Azure ML Studio'da oturum açıldıktan sonra Şekil 2'deki gibi açılış sayfası ekrana gelmektedir.

Şekil 2'de görüldüğü gibi, ekranın sol tarafında kullanıcıya sunulmuş altı farklı seçenek bulunmaktadır: Projects (Projeler), Experiments (Deneyler), Web Services (Web Servisleri), Datasets (Veri Setleri), Trained Models (Eğitilmiş Modeller) ve Settings (Ayarlar). Seçeneklerin açıklamalarının özeti Tablo 1'de verilmektedir.

3.3.2 Azure ML Studio'da İş Akışı

Azure ML Studio'da makine öğrenmesi modelleri oluşturmanın birincil aracı deneylerdir. Deneyler, modellerin oluşturulduğu ana yerdir [18]. Sıfırdan deney oluşturulabilir veya mevcut örnek deneyler arasından seçim yapılabilir [16]. Yeni bir deney oluşturmak için sol altta yer alan **+ NEW** butonuna basılır. Şekil 3'te gösterildiği gibi **EXPERIMENTS** seçilir. Burada, yeni bir deney oluşturma seçeneği ve Microsoft Azure ML Studio tarafından sunulan örnek deneyler verilmektedir. Yeni bir deney **Blank Experiment** seçeneği seçilerek oluşturulabilir.

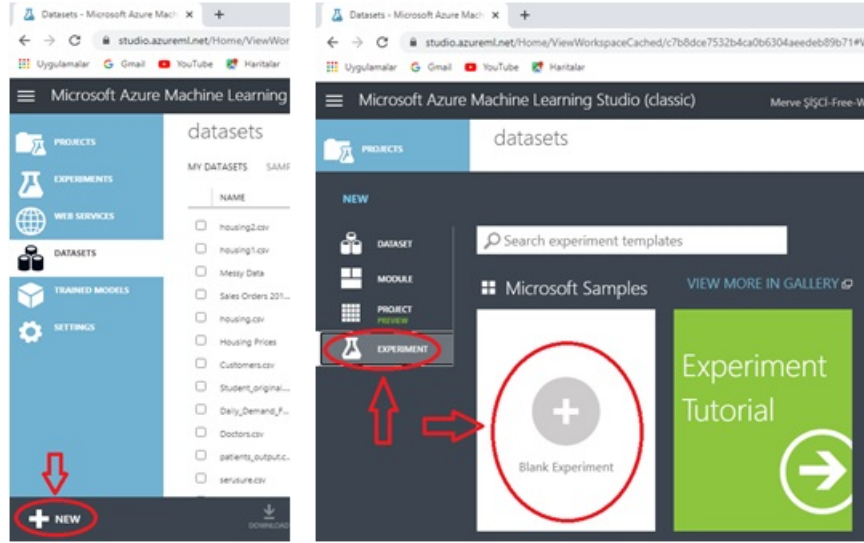
Açılan sayfada Şekil 4'te görüldüğü gibi, makine öğrenmesi uygulamalarının gerçekleştirilebileceği deney alanı ekrana gelmektedir. Sol tarafta uygulamalarda kullanılacak veri setlerinin



Şekil 3.2: Microsoft Azure ML Studio açılış sayfası ekranı

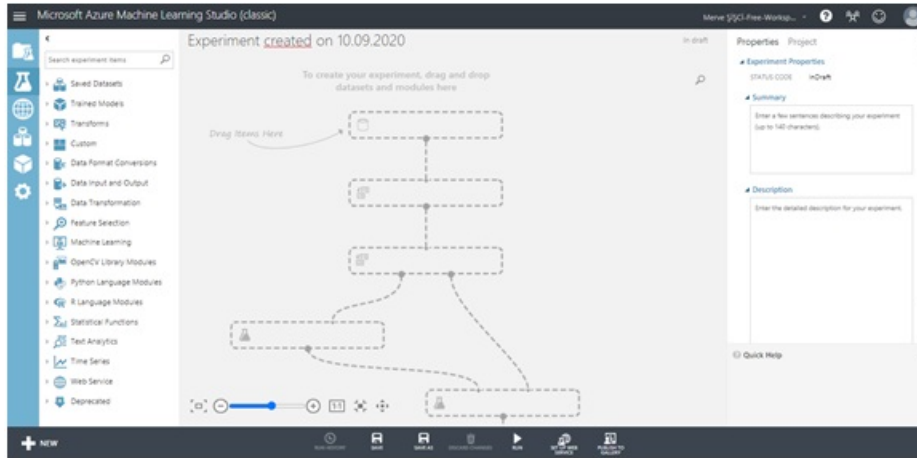
Tablo 3.1: Microsoft Azure ML Studio açılış sayfası seçenekleri [6, 17]

Seçenek	Açıklama
Projects	Tek bir makine öğrenmesi projesini temsil eden deney koleksiyonlarının, veri setlerinin ve diğer kaynakların depolandığı yerdir.
Experiments	MY EXPERIMENTS başlığı altında taslak olarak oluşturulmuş, çalıştırılmış ve kaydedilmiş deneyleri listeler. EXPERIMENTS seçeneğinde ayrıca projeleri hızlı bir şekilde başlatmaya yardımcı olmak için Azure ML Studio hizmetiyle birlikte gönderilen SAMPLES başlığı altında birçok örnek deney bulunmaktadır.
Web Services	Web hizmetleri olarak yayınlanan deneyleri listeler. Bu liste, ilk denemeyi yayınlayana kadar boş kalacaktır.
Datasets	Azure ML Studio'ya yüklenen veri setleri ile Azure ML Studio hizmetiyle birlikte gönderilen örnek veri kümelerinin listesini içerir. Azure Machine Learning hakkında bilgi edinmek için örnek veri kümeleri kullanılabilir.
Trained Models	Deneylerde eğitilen ve Azure ML Studio'da kaydedilen tüm makine öğrenmesi modellerinin bir listesini görüntüler. Azure ML Studio'ya ilk giriş yapıldığında bu liste boş olacaktır.
Settings	Hesabı ve ilişkili kaynakları yapılandırmak için kullanılacak bir ayarlar topluluğudur. Diğer kullanıcıları Azure Machine Learning'de çalışma alanını paylaşmaya davet etmek için bu seçeneği kullanılabilir.



Şekil 3.3: Yeni bir makine öğrenmesi deneyi oluşturulması

ve algoritma modüllerinin bulunduğu deney öğeleri (experiment items) bölümü, sağ tarafta deney ve modüller ile ilgili özelliklerin görüntülenebileceği ve değiştirilebileceği özellikler (properties) bölümü yer almaktadır.



Şekil 3.4: Yeni bir makine öğrenmesi deneyi

Her deney, tahmine dayalı bir model oluşturmak, test etmek ve değerlendirmek için gereken tüm bileşenlere sahip eksiksiz bir iş akışıdır [6]. Bu arayüz, bir akış şeması tasarlamaya benzer görsel bir sürük ve bırak deneyimi sağlar, ancak etkinlikleri ve karar noktalarını temsil etmek yerine, her kutu bir modül olarak adlandırılır ve akış şemasının kendisine bir eğitim deneyi denir [19]. Bir deneyde, makine öğrenmesi modülleri, iş akışı boyunca veri ve parametrelerin akışını gösteren çizgilerle birbirine bağlanır [6]. Deney tuvalinin solundaki modül paletinde bulunan bu modüller, tasarımcı yüzeyine kolayca sürüklenebilen çok sayıda işlevsellik ve yetenek sağlar. Machine Learning Studio'daki (klasik) her modül, bağımsız olarak çalışabilen ve gerekli girdiler verildiğinde bir makine öğrenmesi görevi gerçekleştirebilen bir dizi kodu temsil eder. Bir modül, belirli bir

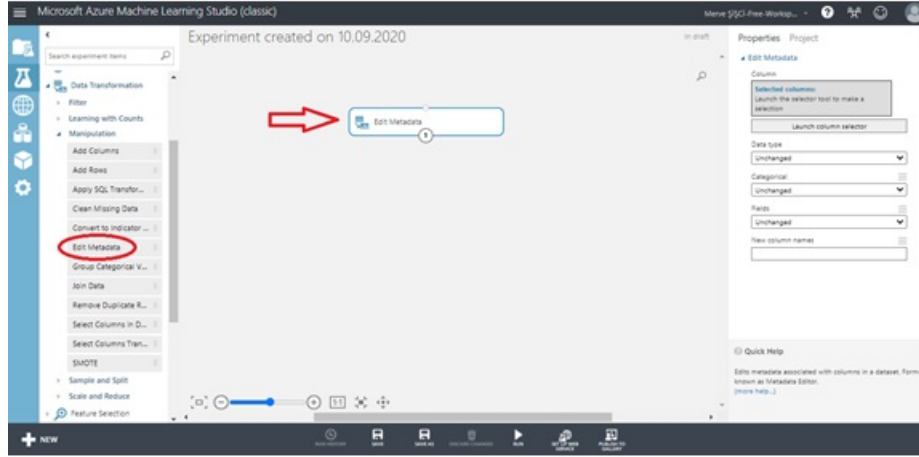
algoritma içerebilir veya eksik değer değişimi ve istatistiksel analiz gibi makine öğrenmesinde önemli olan bir görevi gerçekleştirebilir [20]. Azure ML Studio modülleri Tablo 2'deki kategorilerde gruplandırılmıştır [17].

Tablo 3.2: Microsoft Azure ML Studio modülleri kategorileri [17, 20]

Kategori	Açıklama
Saved Datasets	Bu başlık altında ML Studio tarafından sunulan örnek veri setleri ve kullanıcı tarafından yüklenen veri setleri bulunmaktadır. Bu veri setlerinden herhangi biri deney alanına sürüklenerek kullanılabilir.
Trained Models	Bu kategoride önceden eğitilmiş ve kaydedilmiş modeller bulunmaktadır.
Transforms	Bu kategoride önceki deneylerde kaydedilmiş veri dönüşümü modülleri bulunmaktadır.
Data Format Conversions	Bu modüller, verileri diğer makine öğrenmesi araçları tarafından kullanılacak birçok yaygın biçime (TSV, CSV ve ARFF gibi) dönüştürmeye yardımcı olabilir.
Data Input and Output	Diğer bulut ve web veri kaynaklarından veri okumak için bu modüller kullanılabilir. Ayrıca "sıkıştırılmış" veri kümelerini okumak ve bu modülleri kullanarak verileri dışa aktarmak da mümkündür.
Data Transformation	Bu modüller, makine öğrenmesi analizi için veri hazırlanmasına yardımcı olmak için kullanılır. Bu modüller ile veri türlerini değiştirilebilir, aykırı veriler kırılabilir, özellikler tanımlanabilir ve / veya oluşturabilir, eksik veriler temizlenebilir, veriler normalleştirilebilir ve çok daha fazlası yapılabilir.
Feature Selection	Bu modüller, istatistiksel yöntemler kullanarak veriler en iyi öznitelikleri veya özellikleri belirleyecek yardımcı olmak için kullanılır.
Machine Learning	Bu grup, başlangıç modeli ekleme, modeli eğitme, puanlama, değerlendirme başlıkları altında Azure Machine Learning tarafından desteklenen algoritmaların çoğunu içerir.
Open CV Library Modules	Bu modüller, görüntü işleme, görüntü tanıma ve görüntü sınıflandırması için açık kaynaklı bir kitaplığa kolay erişim sağlar.
Python Language Modules	Bu modüller, özel Python kodunu kullanarak makine öğrenmesi deneylerini genişletmek için kullanılır.
R Language Modules	Bu modüller, makine öğrenmesi deneyine özel R kodu eklemek için kullanılır.
Statistical Functions	Bu modüller, veri bilimiyle ilgili çeşitli sayısal analiz görevlerini gerçekleştirmek için istatistiksel yöntemleri uygulamak için kullanılır. Veri özetleme, korelasyon yöntemleri ve istatistiksel ve matematik işlemler bulunmaktadır.
Text Analytics	Bu modüller, çeşitli doğal dil işleme görevlerini destekler.
Time Series	Zaman serilerinde anormallik tespitini destekler.
Web Service	Bu modüller, mevcut bir Azure Machine Learning web hizmetine giriş veya çıkış bağlantı noktaları eklemek için kullanılabilir.

Azure ML Studio'da, girdi bağlantı noktaları modül simgelerinin üzerinde bulunur ve çıktı bağlantı noktaları modül simgelerinin altında bulunur [21]. Şekil 5'te örnek olarak deney alanına Edit Metadata modülü eklenmiştir. Edit Metadata modülü bir veri setindeki sütunlarla ilişkili meta verileri değiştirmek için kullanılır. Bu modül ile mantıksal (boolean) veya sayısal değerli sütunlar kategorik değerler olarak işlenebilir, sınıf etiketi veya kategorize etmek ya da tahmin etmek istenilen değerleri içeren sütun belirtilebilir, sütunlar nitelik olarak işaretlenebilir, tarih / saat değerleri sayısal bir değere veya tam tersine değiştirilebilir, sütunlar yeniden adlandırılabilir. Deneye Edit Metadata modülü ekleyebilmek için sol tarafta bulunan deney öğeleri bölümünden sırasıyla Data Transformation ve Manipulation seçenekleri seçilir. Şekil 5'te gösterildiği gibi açılan listeden Edit

Metadata modülü seçilip sürüklenerek deney alanına bırakılır.



Şekil 3.5: Deneye Edit Metadata modülünün eklenmesi

Fare bir modül üzerindeki bağlantı noktalarından herhangi biri üzerine getirilirse, bağlantı noktasının türünü gösteren bir "araç ipucu" görülür [21]. Şekil 5'te görüldüğü gibi Edit Metadata modülünün bir girdi bağlantı noktası ve bir çıktı bağlantı noktası bulunmaktadır:

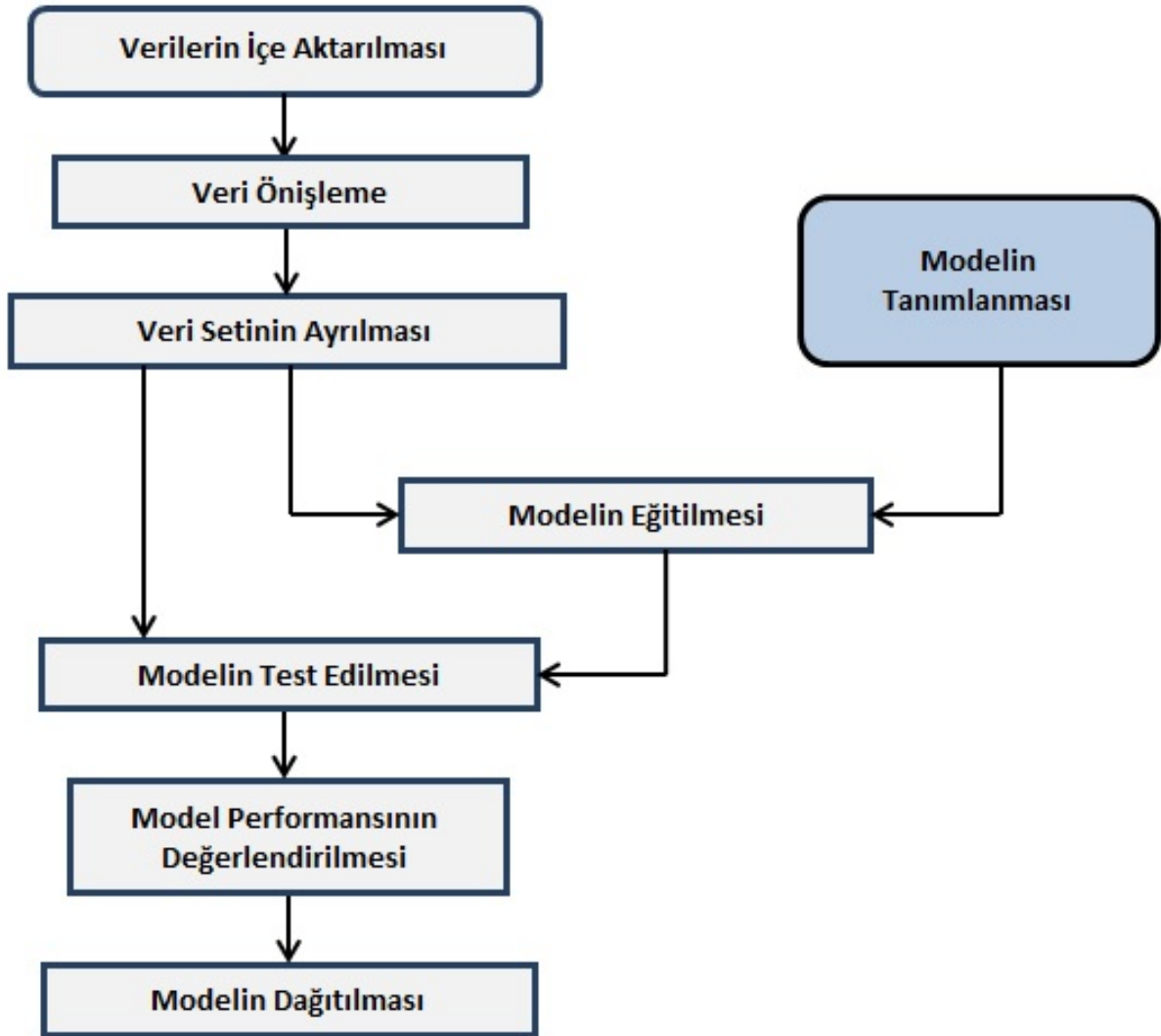
- Dataset girdi bağlantı noktası, üzerinde çalışılmak istenilen Azure veri tabloları için giriştir.
- Results dataset çıktı bağlantı noktası, Edit Metadata modülü ile meta verisi değiştirilmiş bir veri tablosudur.

Şekil 6'da, Azure ML Studio'da bir modeli eğitmek, test etmek ve değerlendirmek için geliştirilmiş bir iş akışı gösterilmektedir. Bu genel iş akışı çoğu regresyon ve sınıflandırma algoritması için benzerdir [21].

Veriler, bir makine öğrenmesi çözümünün en önemli parçasıdır [16]. Azure ML Studio'da tahmine dayalı modeller geliştirmek amacıyla bir deney oluşturmak için öncelikle veri alınması gerekmektedir [10, 18]. Azure ML Studio, birçok örnek veri kümesi sağlamaktadır. Ek olarak, birçok farklı kaynaktan veri aktarılabılır [6]. ML Studio; Excel sayfaları, CSV, Hadoop Hive, Azure SQL DB, blob depolama vb. dahil olmak üzere çok çeşitli dosya biçimlerini desteklemektedir [16].

Bir makine öğrenmesi projesinin kalitesi, büyük ölçüde girdi verilerinin kalitesine bağlıdır [1]. Deneyi tasarlamaya başlamadan önce, veri kümesini önceden işlemek önemlidir [6]. İç aktarılan ham veriler, makine öğrenmesi analizine hemen hazır olmayabilir [16]. Çoğu durumda, ham verilerin tahmine dayalı bir analitik modeli eğitmek için girdi olarak kullanılmadan önce ön işlemeden geçirilmesi gerekir [6]. Örneğin, tablo biçiminde bir veri yapısında bazı satırlarda bir veya daha fazla sütun için eksik değerler olabilir [16]. Bazı durumlarda sayısal nitelikler farklı büyüklükte ölçeklerde bulunabilir. Diğer taraftan gereksiz nitelikler, gürültülü veya aykırı veriler olabilir. Azure ML Studio veri ön işleme için Normalize Data, Edit Metadata, Apply Math Operation, Clip Values gibi birçok modül sunmaktadır.

Makine öğrenmesi genel olarak, veri ön işleme işlemlerinden sonra verilerin iki farklı akışa ayrılmasını gerektirir: eğitim verileri ve test verileri. Eğitim verileri, ilk modeli oluşturmak için bir makine öğrenmesi algoritması tarafından kullanılır. Test verileri, doğruluğunun kontrol edilmesine yardımcı olarak oluşturulan modeli değerlendirmek için kullanılır [16, 17]. Burada önemli olan, modeli eğitmek için kullanılan aynı veriler üzerinde modelin test edilmek veya değerlendirilmek istenmemesidir [13]. Azure Machine Learning Studio tarafından sunulan Split Data modülü ne kadar



Şekil 3.6: Microsoft Azure ML Studio'da iş akışı [16, 21]

verinin eğitime ve ne kadarının teste girmesi gerektiğini belirlemeye olanak tanır [18]. Diğer taraftan çapraz doğrulama için Cross Validate Model modülü sunmaktadır.

Veri setleriyle ilgili çıkarımlara dayalı tahminler yapabilen yeni modeller oluşturmak için çeşitli makine öğrenmesi algoritmaları kullanılır [10]. Azure ML Studio'da Model Tanımlama modülü model türünü ve özelliklerini tanımlar. Deney ögeleri bölmesinde modeller için çok sayıda seçenek bulunmaktadır [21]. Makine öğrenmesi iş akışındaki en önemli adım, verileri içe aktardıktan ve ön işlemeden geçirdikten sonra problemi çözmeye yardımcı olmak için kullanılacak uygun makine öğrenmesi algoritmasını belirlemektir [17]. Doğru algoritmayı seçmek, çözülen problemin ve eldeki verilerin türüne bağlıdır [16]. Azure ML Studio temel olarak anomali tespiti, sınıflandırma, kümeleme ve regresyon uygulamaları için makine öğrenmesi algoritmaları sunar. Microsoft, bir algoritma seçmeyi kolaylaştırmak için 'Machine Learning Algorithm Cheat Sheet' olarak adlandırılan bir diyagram sağlamıştır [22]. Azure ML Studio'da sunulan ve problem türüne göre seçilebilecek algoritmaların özeti şu şekildedir [22]:

- **Regresyon:** Poisson Regresyon (Poisson Regression), Doğrusal Regresyon (Linear Regression), Bayesyen Doğrusal Regresyon (Bayesian Linear Regression), Karar Ormanı Regresyonu (Decision Forest Regression), Sinir Ağı Regresyonu (Neural Network Regression), Güçlendirilmiş Karar Ağacı Regresyonu (Boosted Decision Tree Regression)
- **İkili Sınıflandırma:** Destek Vektör Makinesi (Support Vector Machine), Ortalamalı Algılayıcı (Averaged Perceptron), Karar Ormanı (Decision Forest), Lojistik Regresyon (Logistic Regression), Güçlendirilmiş Karar Ağacı (Boosted Decision Tree), Sinir Ağı (Neural Network)
- **Çok Sınıflı Sınıflandırma:** Lojistik Regresyon (Logistic Regression), Sinir Ağı (Neural Network), Karar Ormanı (Decision Forest), Bire Karşı Hepsini Çok Sınıflı (One-vs-All Multiclass), Güçlendirilmiş Karar Ağacı (Boosted Decision Tree)
- **Kümeleme:** K-Ortalamalar (K-Means)
- **Anomali Tespiti:** Tek-sınıf DVM (One-class SVM), Temel Bileşen Analizi-tabanlı (Principal Component Analysis-based (PCA))
- **Görüntü Sınıflandırma:** DenseNet

Modeli oluşturmak bir eğitim veri seti kullanılan yinelemeli bir süreçtir [16]. Problem türüne uygun algoritma seçiminden sonraki adım algoritmanın eğitim verileri ile eğitilmesini sağlamak için deneye regresyon ve sınıflandırma çalışmalarında Train Model modülü, kümeleme çalışmalarında ise Train Clustering Model modülü eklemektir. Bu modüller deney ögeleri bölmesinde Machine Learning başlığı altında bulunabilir [17]. Seçilen algoritmayı kullanarak eğitimin tamamlanması yalnızca birkaç saniye sürmektedir. Eğitim süreleri, veri miktarı ve seçilen algoritma ile doğru orantılıdır [16]. Eğitilmiş bir sınıflandırma veya regresyon makine öğrenmesi modeli kullanılarak test verileri üzerinde tahminler oluşturmak için Score Model modülü kullanılabilir. Oluşturulan modelin doğruluğunu kontrol etmek için Evaluate Model modülü kullanılarak modeli değerlendirmek mümkündür [18]. Değerlendirme modülünde regresyon uygulamaları için bulunan performans ölçüm kriterleri şunlardır [6]:

- **Mean Absolute Error (MAE):** Mutlak hataların ortalaması (bir hata, tahmin edilen değer ile gerçek değer arasındaki farktır).
- **Root Mean Squared Error (RMSE):** Hata karelerinin ortalamasının karekökü.
- **Relative Absolute Error:** Nispi mutlak hata, gerçek değerler ile tüm gerçek değerlerin ortalaması arasındaki mutlak farka göre mutlak hataların ortalamasıdır.
- **Relative Squared Error:** Gerçek değerlerle tüm gerçek değerlerin ortalaması arasındaki kare farkına göre hata karelerinin ortalaması.
- **Coefficient of Determination:** R kare değeri olarak da bilinen bu ölçüm kriteri, bir modelin

verilere ne kadar iyi uyduğunu gösteren istatistiksel bir ölçüttür.

Sınıflandırma uygulamaları için kullanılan performans ölçüm kriterleri ise şunlardır [23]:

- **Accuracy (Doğruluk):** Doğru sınıflandırılmış sonuçların toplam vakalara oranı olarak bir sınıflandırma modelinin performansını ölçer.
- **Precision (Kesinlik):** Doğru sonuçların tüm olumlu sonuçlara oranıdır.
- **Recall (Duyarlılık):** Model tarafından döndürülen tüm doğru sonuçların oranıdır.
- **F-score:** Kesinlik ve duyarlılığın ağırlıklı ortalaması olarak hesaplanır ve 0 ile 1 arasında değer alır. İdeal F-score değeri 1'dir.
- **AUC:** y ekseninde doğru pozitifler ve x ekseninde yanlış pozitiflerle çizilen eğrinin altındaki alanı ölçer. Bu ölçüm, farklı türdeki modellerin karşılaştırılmasına olanak tanıyan tek bir sayı sağladığından faydalıdır.
- **Average log loss:** Yanlış sonuçların cezasını ifade etmek için kullanılan tek bir skordur. Gerçek olan ve modeldeki olan olmak üzere iki olasılık dağılımı arasındaki fark olarak hesaplanır.
- **Training log loss:** sınıflandırıcının rastgele bir tahmin üzerindeki avantajını temsil eden tek bir skordur. Çıkarıldığı olasılıkları etiketlerdeki bilinen değerlerle (temel gerçek) karşılaştırarak modelin belirsizliğini ölçer. Bir bütün olarak model için bu değer en aza indirmek istenir.

Model değerlendirildikten sonraki adım, isteğe bağlı olarak bir API (Uygulama Programlama Arayüzü) oluşturmaktır. Böylece modeli veri girişi sağlayarak herkes kullanabilir [18]. Microsoft Azure ML Studio ile geliştirilen yeni tahmine dayalı modeli, herhangi bir web tarayıcısı veya mobil istemci tarafından internet üzerinden kolayca erişilebilen, ölçeklenebilir bir bulut web hizmeti olarak ortaya çıkarmak mümkündür [10].

3.4 Regresyon Uygulaması: Konut Fiyatı Tahmini

Konut fiyatı tahmini uygulamasında kullanılan “California Housing Prices” veri seti Kaggle sitesinden alınmıştır [24]. Veriler, 1990 Kaliforniya nüfus sayımına ait bilgileri içermektedir. Bu uygulamanın amacı, Kaliforniya nüfus sayımı verilerini kullanarak Kaliforniya’da bir konut fiyatları modeli oluşturmak ve geliştirilen bu modeli kullanarak Kaliforniya’daki evlerin fiyatını tahmin etmektir. Bu veriler, Kaliforniya’daki her blok grubu için nüfus, medyan gelir, medyan konut fiyatı vb. gibi ölçümlere sahiptir. Blok grupları, ABD Nüfus Bürosu’nun örnek verileri yayınladığı en küçük coğrafi birimdir (bir blok grubu tipik olarak 600 ila 3.000 kişilik bir nüfusa sahiptir). Veri setinde her bir satır bir blok grubu ile ilgili veriler içermekte olup toplam 20640 gözlem bulunmaktadır. Veri setinde bulunan nitelikler, niteliklerin açıklamaları ve değer aralıkları Tablo 3’te, veri setinden alınan 10 örnek ise Tablo 4’te verilmektedir. Veri setinde biri kategorik ve nominal olmak üzere toplam 10 nitelik bulunmaktadır. 10. nitelik tahmin edilmek istenen bir blok içindeki haneler için medyan ev değeridir (median_house_value). “total_bedrooms” niteliğinde 207 eksik veri bulunmaktadır.

Bu çalışma için veri setine her blok grubu için benzersiz ve tanımlayıcı nitelikte olan “id” isimli bir nitelik eklenerek veri seti Şekil 7 ve Şekil 8’de görüldüğü gibi “housing1.csv” ve “housing2.csv” olmak üzere iki bölüme ayrılmıştır. housing1.csv dosyasında id, longitude, latitude, housing_median_age, total_rooms, total_bedrooms, population, households, median_income, ocean_proximity nitelikleri ile toplam 20640 örnek bulunurken, housing2.csv dosyasında bu örneklerle ait id ve median_house_value değerleri bulunmaktadır.

Microsoft Azure ML Studio kullanılarak gerçekleştirilen uygulamanın akış şeması Şekil 9’da verilmektedir.

Tablo 3.3: Konut fiyatı tahmini veri setindeki nitelikler, açıklamaları ve değer aralıkları

Nitelik	Açıklama	Değer
1 longitude	boylam: bir evin ne kadar batıda olduğunun bir ölçüsü; daha yüksek bir değer daha batıdadır	[-124.35, -114.31]
2 latitude	enlem: bir evin ne kadar kuzeyde olduğunun bir ölçüsü; daha yüksek bir değer daha kuzeydedir	[32.54, 41.95]
3 housing_median_age	konut medyan yaşı: Bir blok içindeki bir evin ortalama yaşı; daha düşük sayı daha yeni bir binadır	[1, 52]
4 total_rooms	toplam oda: bir bloktaki toplam oda sayısı	[2, 39320]
5 total_bedrooms	toplam yatak odası: bir bloktaki toplam yatak odası sayısı	[1, 6445]
6 population	nüfus: bir blokta ikamet eden toplam insan sayısı	[3, 35682]
7 households	hane: Bir blok için bir ev biriminde ikamet eden bir grup insan olan toplam hane sayısı	[1, 6082]
8 median_income	medyan gelir: Bir konut bloğundaki haneler için medyan gelir (ABD Doları cinsinden ölçülür)	[0.4999, 15.0001]
9 ocean_proximity	okyanusa yakınlığı: Okyanus ile evin konumu	<1H OCEAN, INLAND, NEAR OCEAN, NEAR BAY, ISLAND
10 median_house_value	medyan ev değeri: bir blok içindeki haneler için medyan ev değeri (ABD Doları cinsinden ölçülür)	[14999, 500001]

Tablo 3.4: Konut fiyatı tahmini veri setinden örnek bir bölüm

Örnek No	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	ocean_proximity	median_house_value
1	-122.23	37.88	41	880	129	322	126	8.3252	NEAR BAY	452600.0
2	-122.22	37.86	21	7099	1106	2401	1138	8.3014	NEAR BAY	358500.0
3	-122.24	37.85	52	1467	190	496	177	7.2574	NEAR BAY	352100.0
4	-122.25	37.85	52	1274	235	558	219	5.6431	NEAR BAY	341300.0
5	-122.25	37.85	52	1627	280	565	259	3.8462	NEAR BAY	342200.0
6	-122.25	37.85	52	919	213	413	193	4.0368	NEAR BAY	269700.0
7	-122.25	37.84	52	2535	489	1094	514	3.6591	NEAR BAY	299200.0
8	-122.25	37.84	52	3104	687	1157	647	4.4168	NEAR BAY	241400.0
9	-122.26	37.84	42	2555	665	1206	595	2.0804	NEAR BAY	226700.0
10	-122.25	37.84	52	3549	707	1551	714	3.6912	NEAR BAY	261100.0

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	id	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	ocean_proximity	median_house_value		
2	1	-122.23	37.88	41	880	129	322	126	8.3252	NEAR BAY	452600.0		
3	2	-122.22	37.86	21	7099	1106	2401	1138	8.3014	NEAR BAY	358500.0		
4	3	-122.24	37.85	52	1467	190	496	177	7.2574	NEAR BAY	352100.0		
5	4	-122.25	37.85	52	1274	235	558	219	5.6431	NEAR BAY	341300.0		
6	5	-122.25	37.85	52	1627	280	565	259	3.8462	NEAR BAY	342200.0		
7	6	-122.25	37.85	52	919	213	413	193	4.0368	NEAR BAY	269700.0		
8	7	-122.25	37.84	52	2535	489	1094	514	3.6591	NEAR BAY	299200.0		
9	8	-122.25	37.84	52	3104	687	1157	647	4.4168	NEAR BAY	241400.0		
10	9	-122.26	37.84	42	2555	665	1206	595	2.0804	NEAR BAY	226700.0		

Şekil 3.7: housing1.csv veri seti dosyasından örnek bir bölüm

	A	B	C	D
1	id,median_house_value			
2	1,452600.0			
3	2,358500.0			
4	3,352100.0			
5	4,341300.0			
6	5,342200.0			
7	6,269700.0			
8	7,299200.0			
9	8,241400.0			
10	9,226700.0			

Şekil 3.8: housing2.csv veri seti dosyasından örnek bir bölüm

3.4.1 Veri Setlerinin Azure ML Studio Ortamına Yüklenebilmesi

Veri setlerinin Azure ML Studio ortamına yüklenebilmesi için Şekil 10'da gösterildiği gibi açılış sayfası ekranında sol alt köşede bulunan + **NEW** butonuna basılır. Ardından açılan pencerede sırasıyla **DATASET** ve **FROM LOCAL FILE** butonlarına basılır.

İşlemler gerçekleştirildikten sonra yüklenmesi istenilen veri seti dosyalarının seçilebileceği ekran ile karşılaşılmaktadır. Burada Şekil 11'de gösterildiği gibi Dosya Seç butonuna basılarak housing1.csv veri seti dosyası bulunduğu klasörden seçilir. housing1.csv dosyası nitelik isimlerini de barındıran CSV dosyası olduğu için SELECT A TYPE FOR THE NEW DATASET bölümünden "Generic CSV File with a header (.csv)" seçeneği seçilir. İstenilirse PROVIDE AN OPTIONAL DESCRIPTION bölümüne veri seti ile ilgili açıklamalar yazılabilir. Daha sonra (✓) işaretine basılarak veri setinin yüklenmesi beklenir. Veri setinin yüklenmesi tamamlandığında ekranda "Upload of the dataset 'housing1.csv' has completed." bilgilendirme yazısı çıkmaktadır. Şekil 11'de gösterildiği gibi ekranın sol tarafından DATASETS butonuna basıldığında housing1.csv veri setinin sisteme yüklendiği görülmektedir.

housing2.csv veri seti dosyasını yüklemek için aynı işlemler Şekil 12'de gösterildiği gibi tekrarlanır.

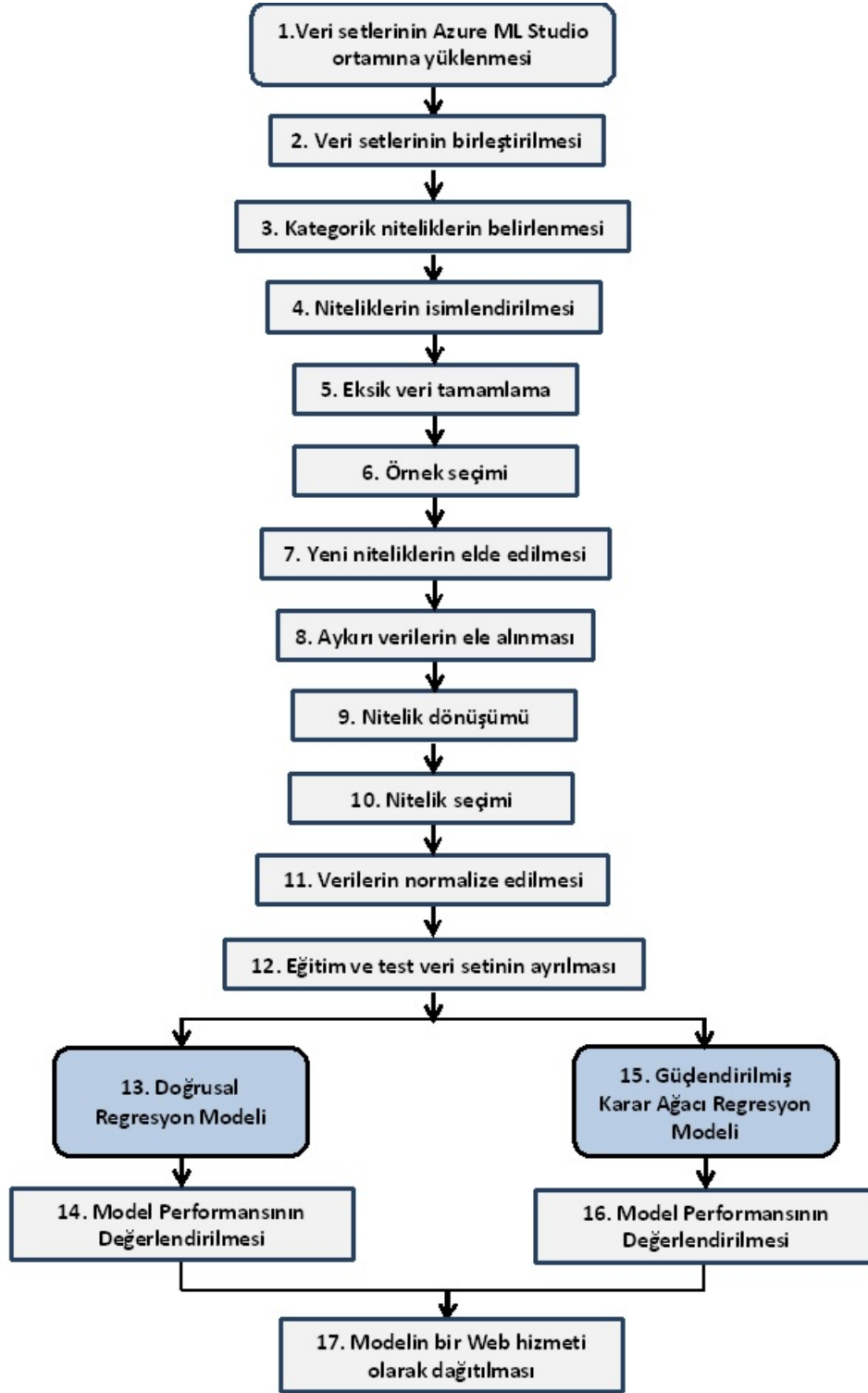
Konut fiyatı tahmini çalışmasının gerçekleştirilmesi amacıyla yeni bir deney sayfası oluşturmak için tekrar ekranın sol alt köşesinde bulunan + **NEW** butonuna basılarak Şekil 13'te gösterildiği gibi **EXPERIMENTS** seçilir. Konut fiyatı tahmini için yeni bir deney oluşturmak amacıyla **Blank Experiment** seçeneği seçilir.

Şekil 14'te gösterilen bölüme Konut Fiyatı Tahmini yazılarak deney yeniden isimlendirilir.

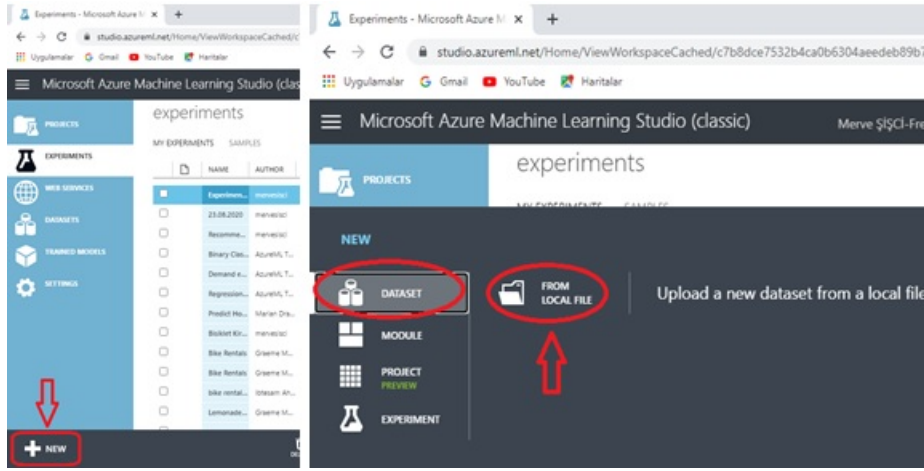
Sol tarafta bulunan deney öğeleri bölümünden sırasıyla **Saved Datasets** ve **My Datasets** seçenekleri seçilir. Burada kullanıcı tarafından sisteme yüklenmiş veri setleri listelenir. Şekil 15'te gösterildiği gibi açılan listeden housing1.csv veri seti seçilip sürüklenerek deney alanına bırakılır.

Verileri öğrenme sürecine hazırlamak için ek çalışma gereklidir. Bu nedenle, veri keşfi adı verilen bir uygulama sırasında veriler hakkında daha fazla bilgi edinmek önemlidir [1]. Hem satırlar hem de sütunlar açısından ne kadar veriye sahip olunduğu konusunda çok iyi bir bilgiye sahip olmak gerekir. Çok fazla satır olduğunda algoritmanın eğitilmesi çok uzun sürebilir. Çok az satırda algoritmaları eğitmek için yeterli veriye sahip olunmayabilir. Çok fazla özellikte ise bazı algoritmaların dikkati dağılabilir veya düşük performansla neden olabilir [25]. Şekil 16'da gösterildiği gibi housing1.csv veri setinin çıktı bağlantı noktasına sağ tıklayarak, verileri keşfetmeye ve her bir sütunun temel istatistiklerini anlamaya olanak tanıyan "Visualize" seçeneği seçilebilir.

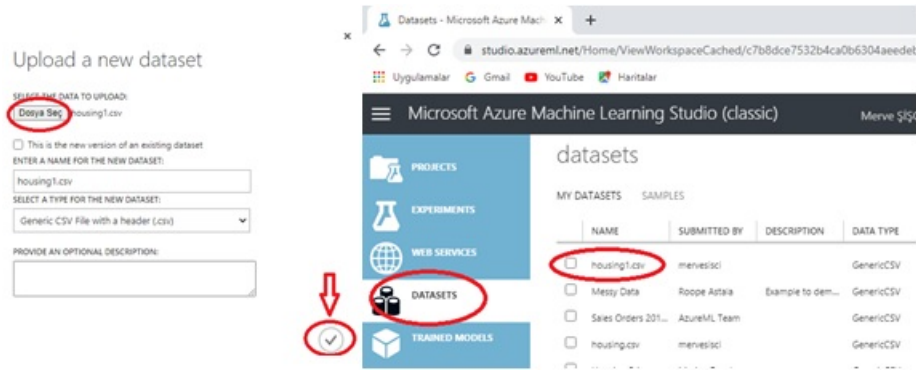
Veri görselleştirmesinde, Şekil 17'de gösterildiği gibi veri setinin her blok grubu için bir kayıt



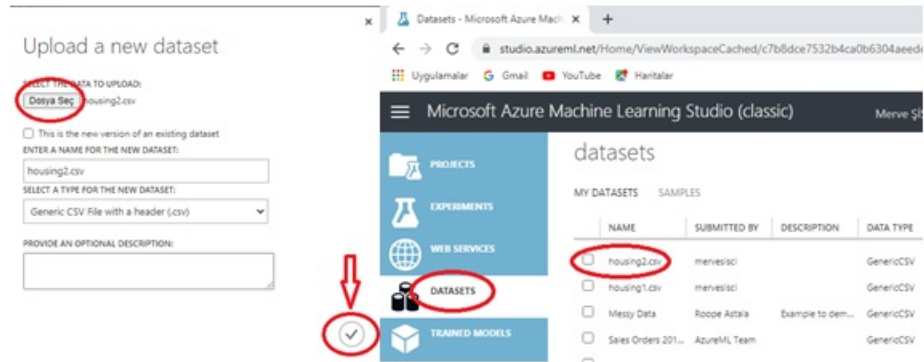
Şekil 3.9: Konut fiyatı tahmini uygulamasının adımları



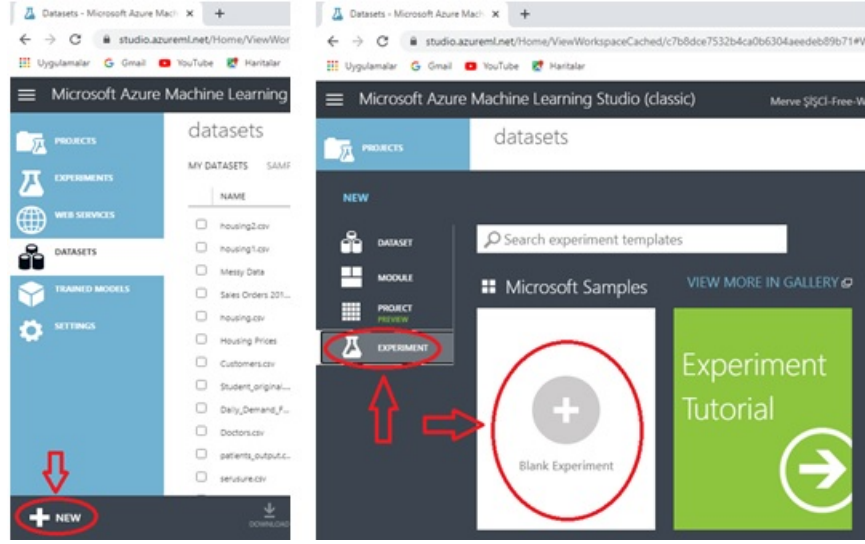
Şekil 3.10: Veri setlerinin Azure ML Studio ortamına yüklenmesi



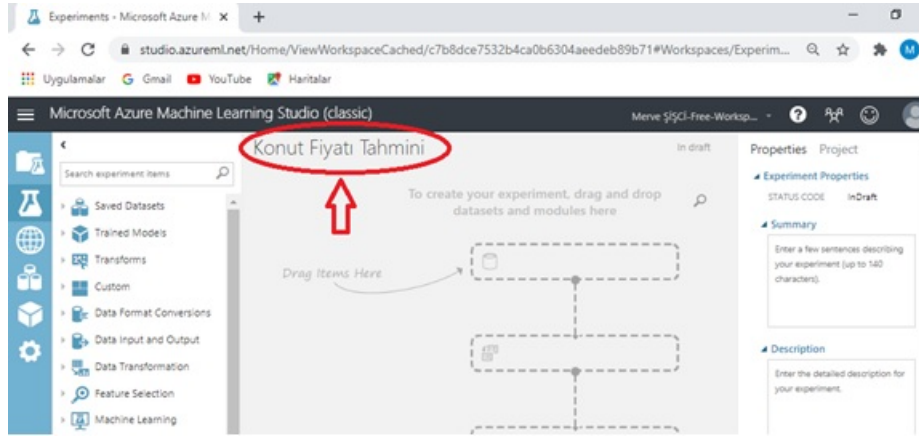
Şekil 3.11: housing1.csv veri seti dosyasının Azure ML Studio ortamına yüklenmesi



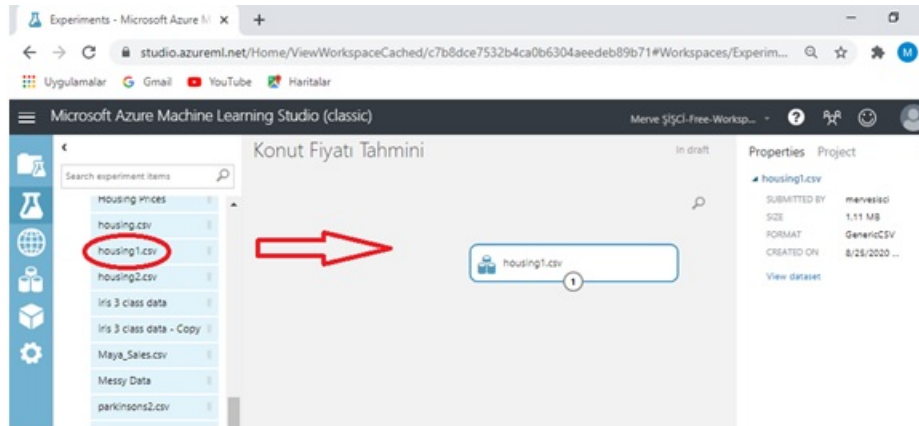
Şekil 3.12: housing2.csv veri seti dosyasının Azure ML Studio ortamına yüklenmesi



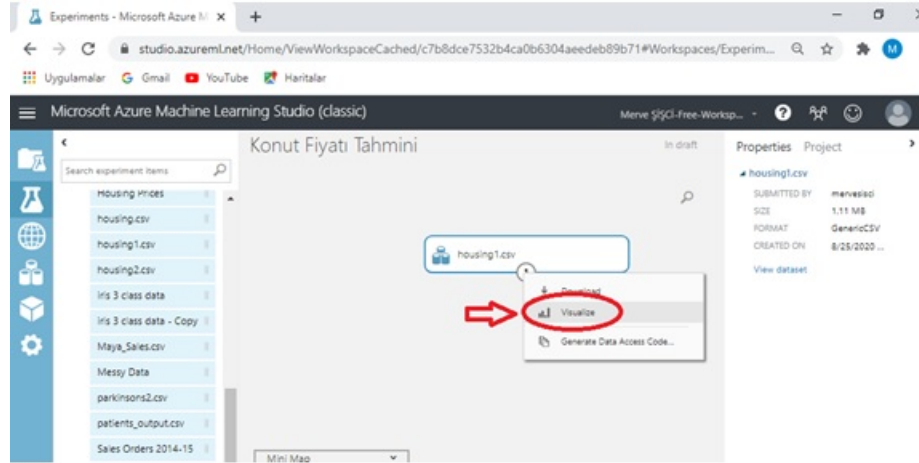
Şekil 3.13: Konut fiyatı tahmini için yeni bir makine öğrenmesi deneyi oluşturulması



Şekil 3.14: Makine öğrenmesi deneyinin isimlendirilmesi

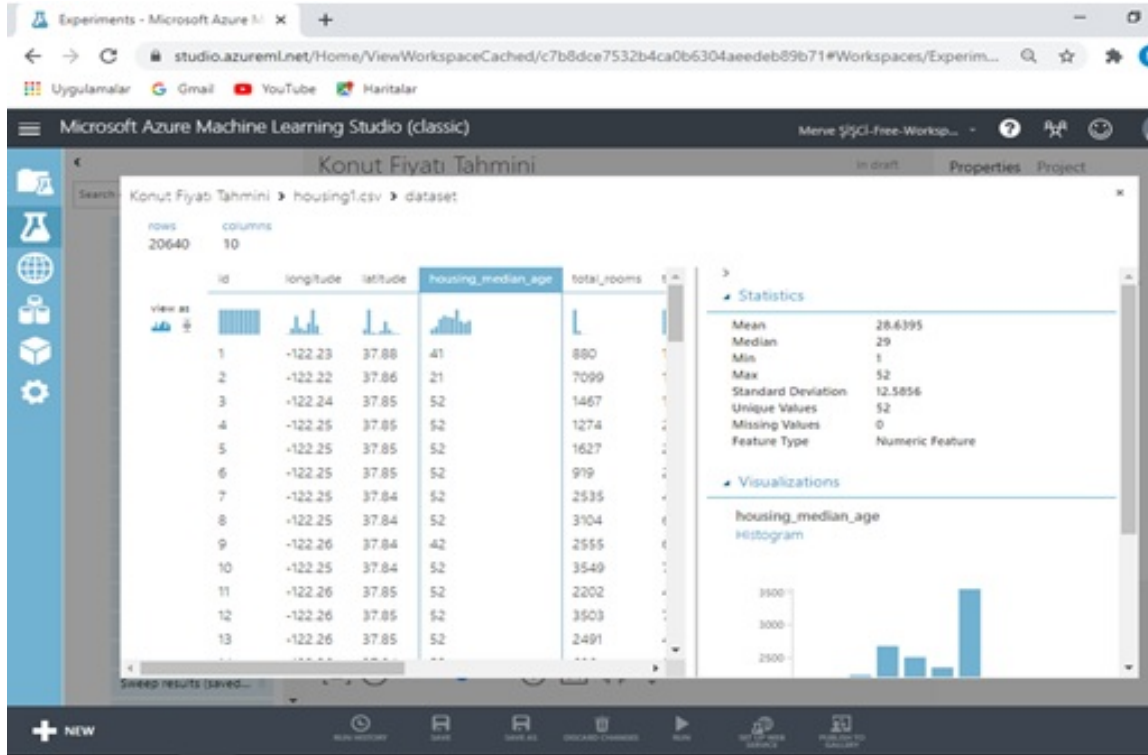


Şekil 3.15: housing1.csv veri setinin deney alanına alınması



Şekil 3.16: housing1.csv veri setinin görselleştirilmesi

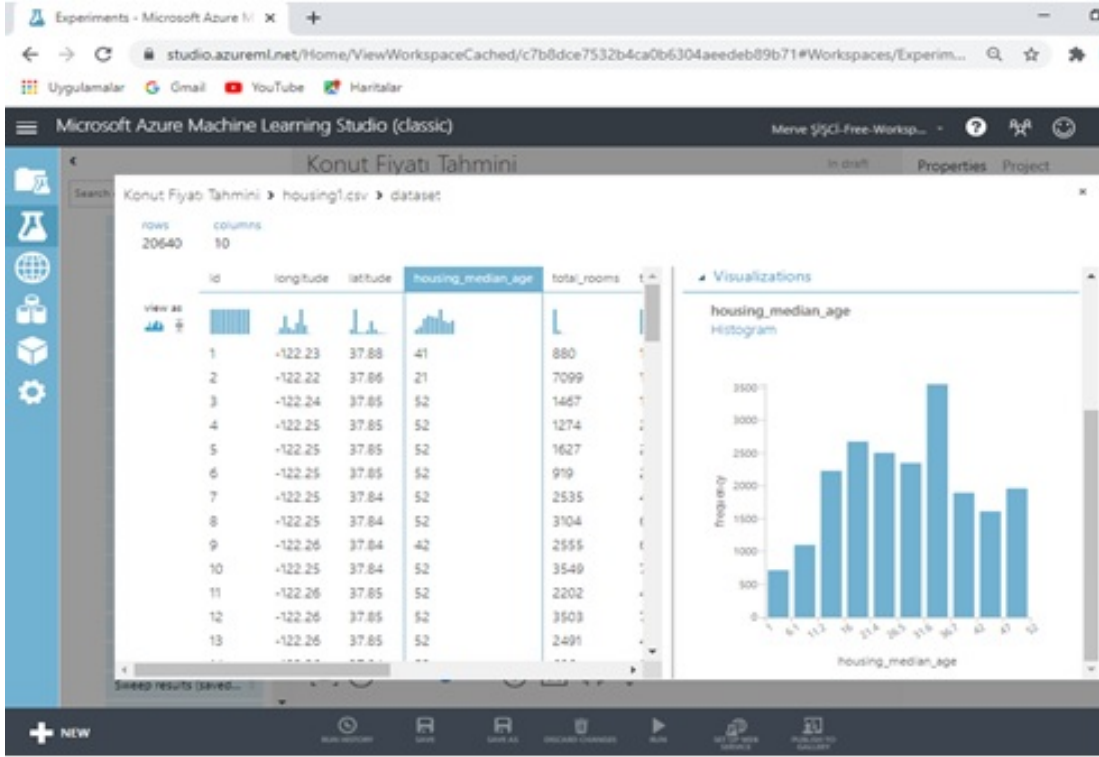
içerdiği (bir blok grubu kimliği ile tanımlanmış) ve her bir kaydın bir blok grubu için nüfus sayımı verilerini ve konutlara ait verileri içerdiği görülmektedir. Veri setinde her bir kayıt için toplam 10 nitelik bulunurken, 20640 adet örnek yer almaktadır. Sütun başlığı seçildiğinde, ilgili sütuna ait istatistikler gösterilmektedir.



Şekil 3.17: housing1.csv veri setinin özelliklerinin incelenmesi

housing_median_age sütunu seçili iken sayfa aşağı kaydırıldığında Şekil 18'de gösterildiği gibi

housing_median_age sütunu için histogram grafiği görülür. Bu, veri kayıtlarındaki blok gruplarının farklı medyan yaşlarının dağılımını gösterir.



Şekil 3.18: housing_median_age niteliğinin histogram grafiği

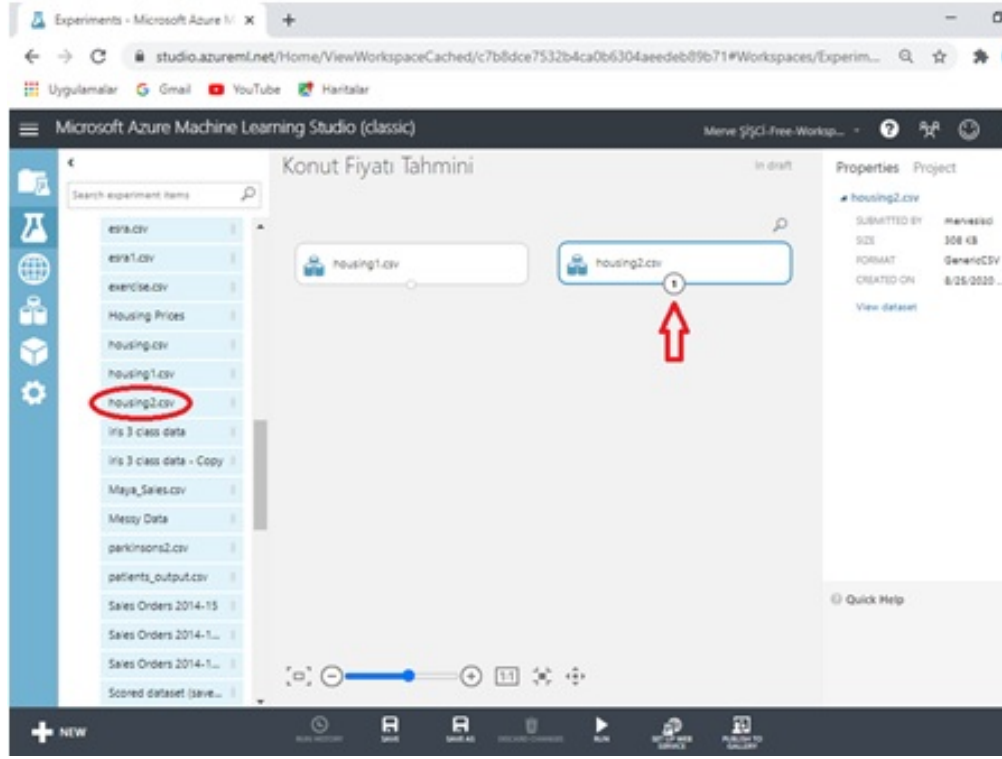
Çalışmada kullanılacak ikinci veri setinin deney ortamına alınması için deney öğeleri bölümünden sırasıyla Saved Datasets ve My Datasets seçenekleri seçilir. Şekil 19’da gösterildiği gibi açılan listeden housing2.csv veri seti seçilip sürüklenerek deney alanında housing1.csv veri setinin yanına bırakılır.

Şekil 20’de gösterildiği gibi housing1.csv veri setinin çıktı bağlantı noktasına sağ tıklayarak, “Visualize” seçeneği seçilir.

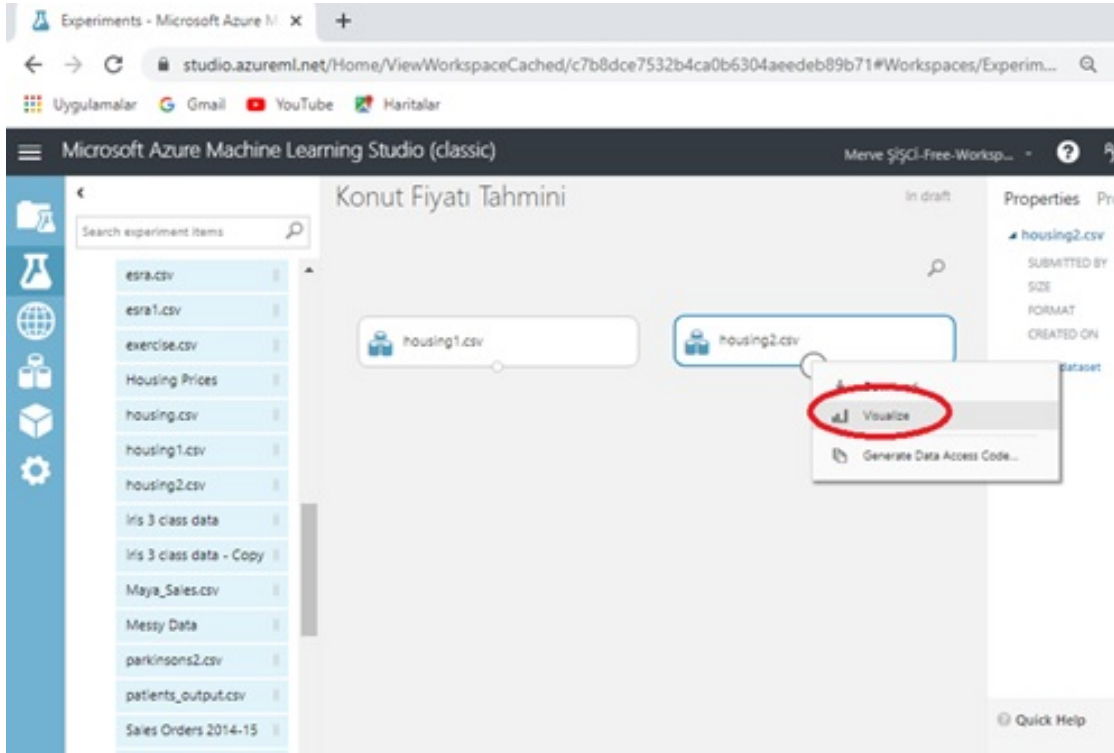
housing2.csv, Şekil 21’de gösterildiği gibi blok gruplarını ve blok grupları için medyan konut değerleri için gözlemler içermektedir. Artık çalışma, ortak bir id alanına sahip iki veri seti içermektedir. İki veri setini birleştirmek için bu alan kullanılabilir.

3.4.2 Veri Setlerinin Birleştirilmesi

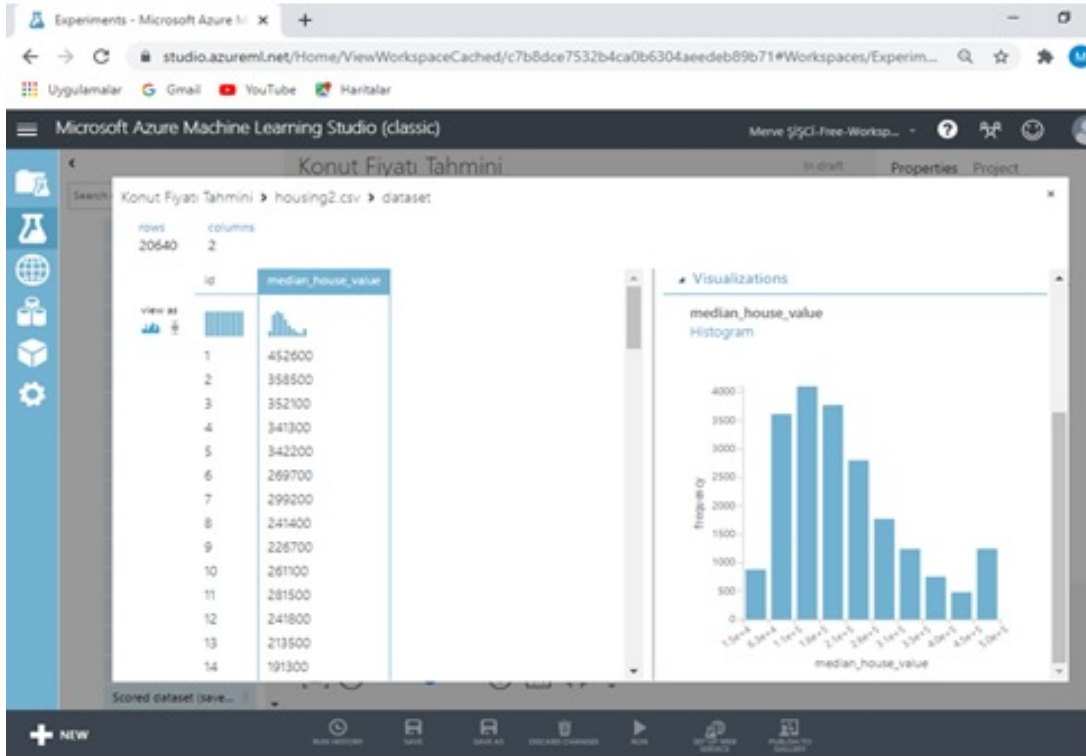
Veri birleştirme süreci, esas olarak, entegre etmek veya birleştirmek istenilebilecek birden fazla veri setine sahip olduğunda yapılır. Bu işlem iki şekilde yapılabilir. Birincisi, birkaç veri setini satırlarını birleştirerek eklemektir. Bu işlem genellikle aynı niteliklere sahip veri setleri için yapılır. İkincisi anahtarlar gibi ortak alanları kullanarak, farklı niteliklere veya sütunlara sahip birkaç veri setini bir araya getirmektir [3]. Microsoft Azure ML Studio’da anahtar alan ile iki veri setinin birleştirilmesi Join Data modülü kullanılarak yapılır. **Search experiment items** kutusuna Join Data yazılır ve **Join Data** modülü seçilerek Şekil 22’de görüldüğü gibi deney alanına daha önce bırakılan housing1.csv ve housing2.csv veri setlerinin altına sürüklenir.



Şekil 3.19: housing2.csv veri setinin deney alanına alınması



Şekil 3.20: housing2.csv veri setinin görselleştirilmesi



Şekil 3.21: housing2.csv veri setinin özelliklerinin incelenmesi

Şekil 23'te gösterildiği gibi sırasıyla housing1.csv verisetinin çıktı bağlantı noktası ile Join Data modülünün Dataset1 (sol) girdi bağlantı noktası bağlanır, housing2.csv verisetinin çıktı bağlantı noktası ile **Join Data** modülünün Dataset2 (sağ) girdi bağlantı noktası bağlanır.

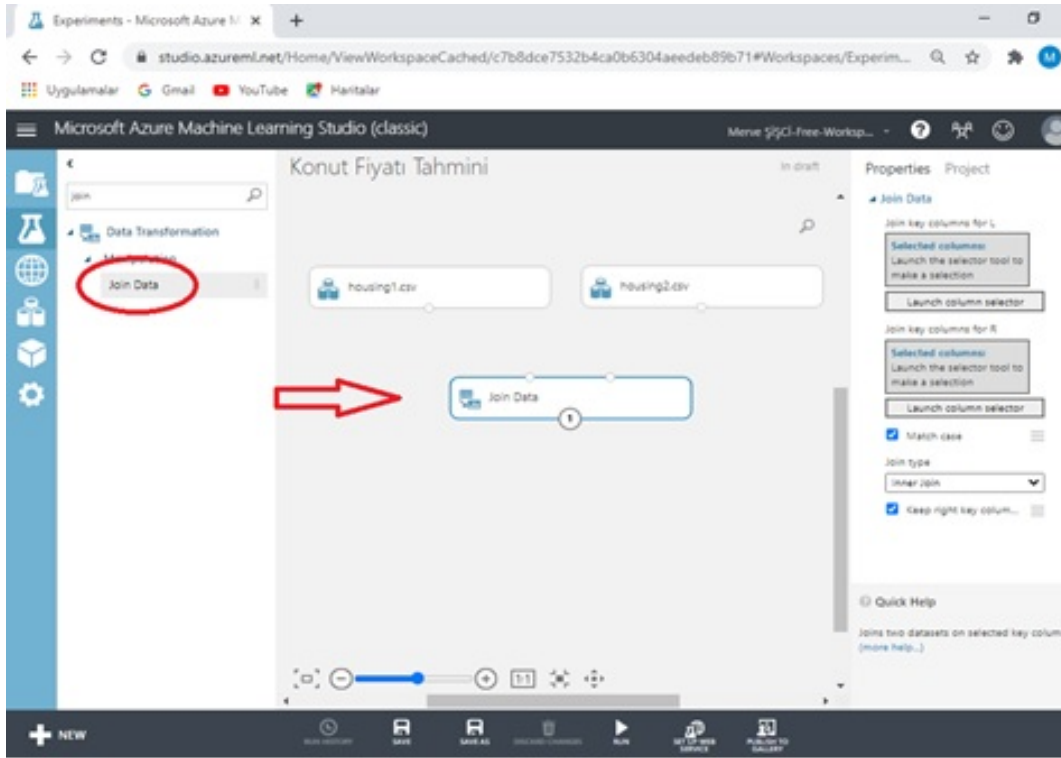
Join Data modülünün sol girdi bağlantısı ile bağlanılan housing1.csv veri setinde birleştirme için kullanılacak anahtar alanın belirlenmesi için **Join Data** modülü seçili iken, **özellikler** (Properties) penceresinden, Şekil 24'teki gibi **Join key columns for L** başlığı altındaki **Launch column selector** tıklanır.

Şekil 25'te gösterilen açılan pencerede **BY NAME** seçili iken, id sütünü seçilir ve sütünü **Selected Columns** listesine ekleme için [**>**] 'a basılır. Daha sonra onay (✓) işaretine basılır.

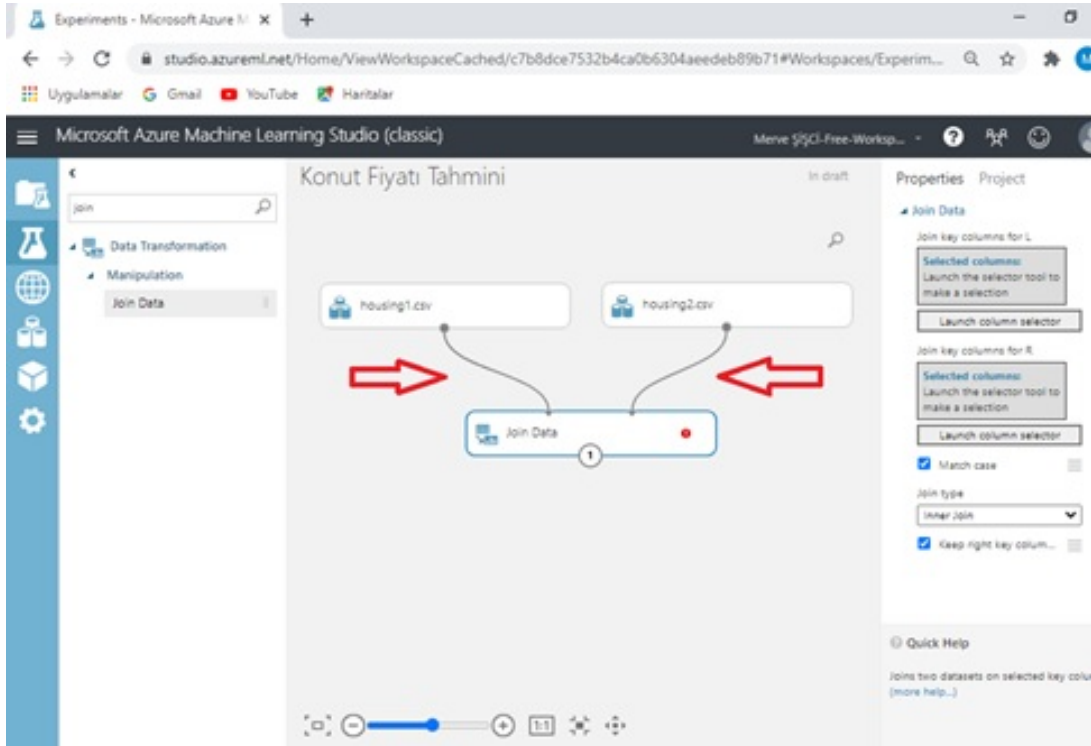
Join Data modülünün sağ girdi bağlantısı ile bağlanılan housing2.csv veri setinde birleştirme için kullanılacak anahtar alanın belirlenmesi için Join Data modülü seçili iken, özellikler (Properties) penceresinden, Şekil 26'daki gibi Join key columns for R altındaki Launch column selector tıklanır.

Açılan pencerede Şekil 27'de gösterildiği gibi **WITH RULES** seçili iken, id sütünü seçilir. Daha sonra onay (✓) işaretine basılır.

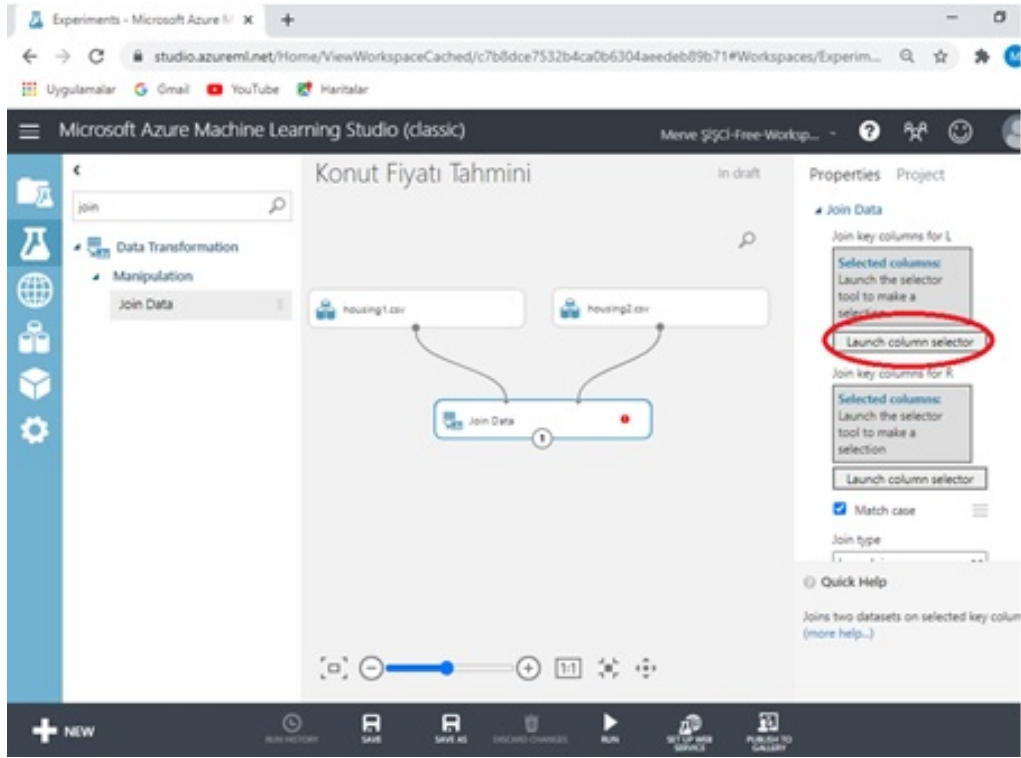
Join Data modülünde verileri birleştirmek için birden fazla veri birleştirme yöntemi bulunmaktadır: inner join, left outer join, full outer join, left semi-join. **Join Data** modülü seçili iken, Şekil 28'de görüldüğü gibi özellikler penceresinde **Match case** onay kutusu temizlenir, Join Type olarak **Left Outer Join**'i seçilir ve **Keep right key column in joined table** onay kutusu temizlenir. Left outer join kullanmak, birleştirilen tablonun **housing1.csv** veri setindeki tüm blok gruplarını ve bunlara **housing2.csv** veri setinde karşılık gelen medyan konut değerlerini (median_house_value) içermesini sağlar. **housing1.csv** veri setinde kaydı olup **housing2.csv** veri setinde eşleşen bir me-



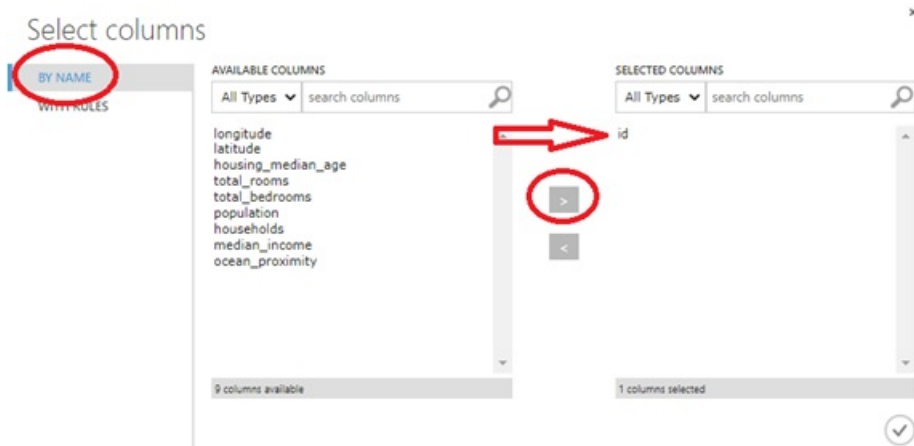
Şekil 3.22: Konut fiyatı tahmini deneyine Join Data modülünün eklenmesi



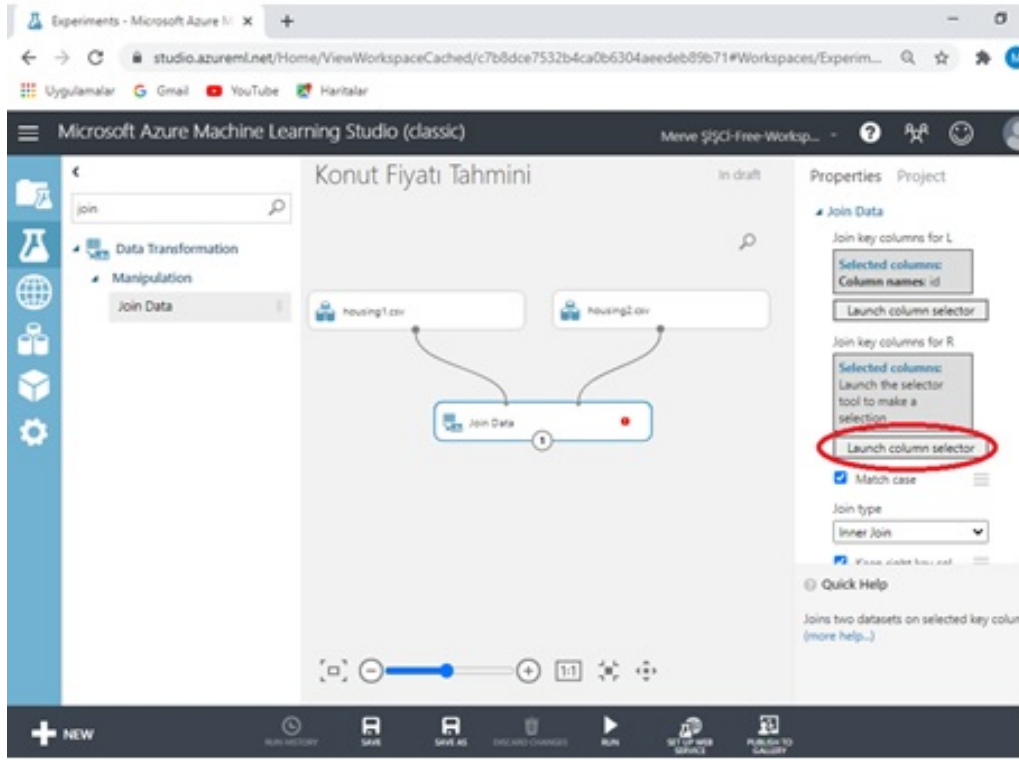
Şekil 3.23: Join Data modülünün bağlantılarının ayarlanması



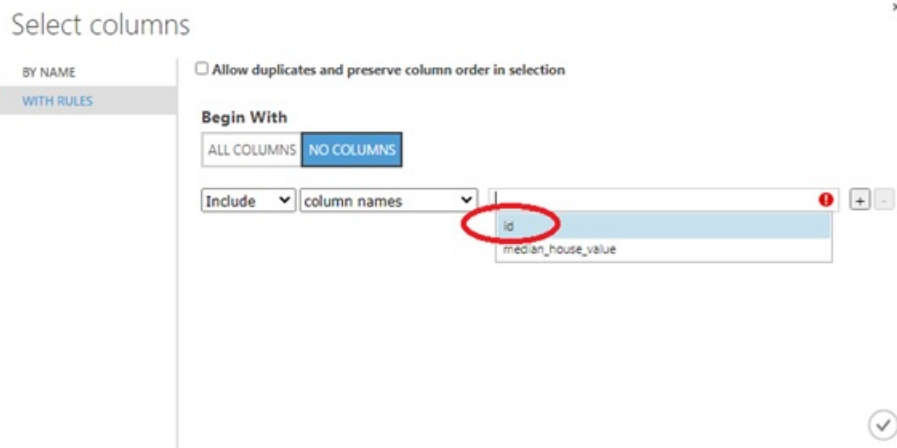
Şekil 3.24: housing1.csv veri setindeki anahtar alanın belirlenmesi



Şekil 3.25: housing1.csv veri setindeki anahtar alanın belirlenmesi (devam)

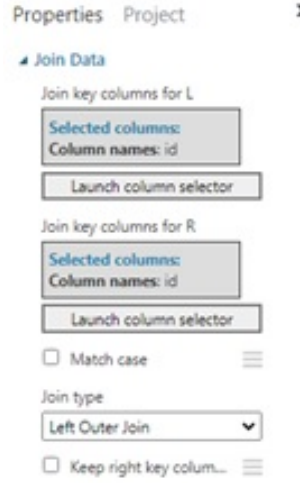


Şekil 3.26: housing2.csv veri setindeki anahtar alanın belirlenmesi



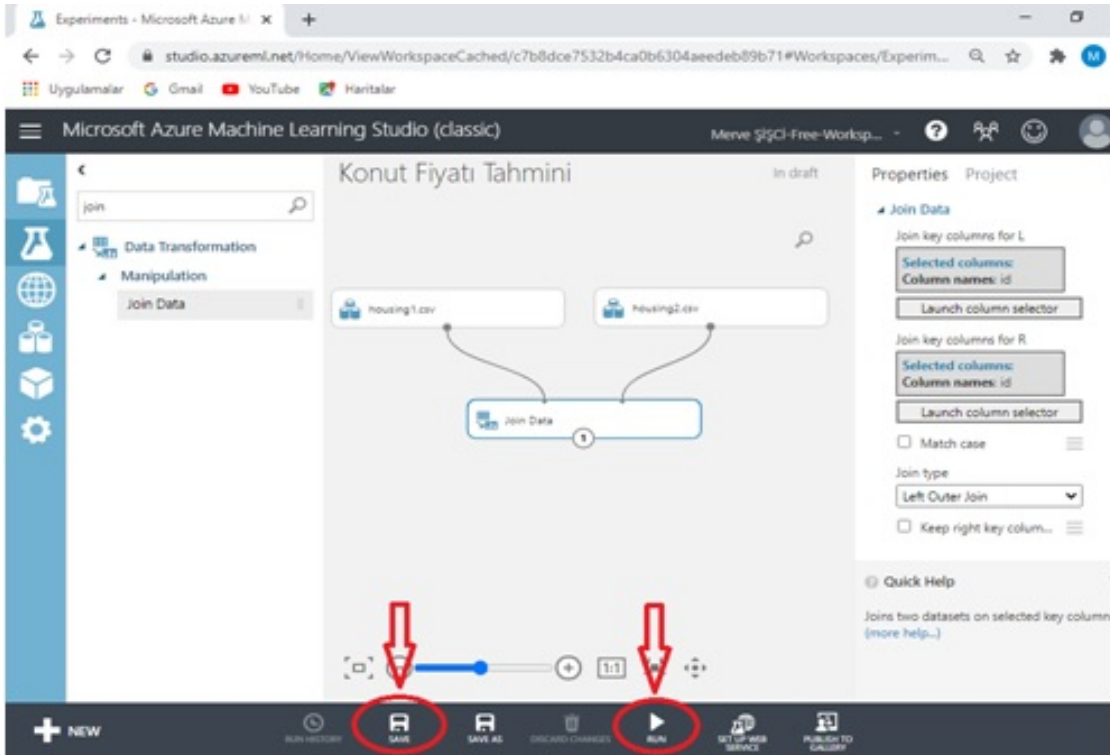
Şekil 3.27: housing2.csv veri setindeki anahtar alanın belirlenmesi (devam)

yan konut değeri bulunmuyorsa, housing1.csv verileri korunur ve karşılık gelen medyan konut değeri NULL olur.



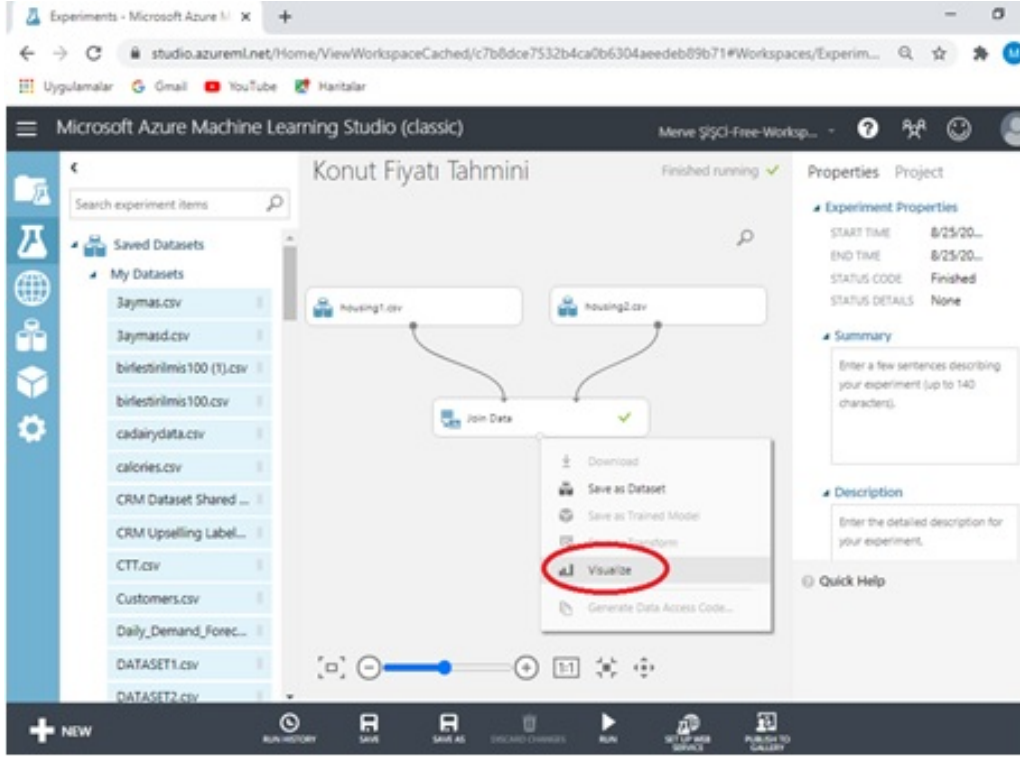
Şekil 3.28: Join Data modülünün özelliklerinin ayarlanması

Join Data modülünün özellikleri ayarlandıktan sonra Şekil 29'da görüldüğü gibi deneyin kaydedilmesi için deney alanının alt bölümünde yer alan SAVE butonuna, çalıştırılması için ise RUN butonuna basılır.



Şekil 3.29: Deneyin kaydedilmesi ve çalıştırılması

Deneyin çalışması tamamlandığında, Şekil 30’da gösterildiği gibi **Join Data** modülü çıktı bağlantı noktasına (Results dataset) basılarak “Visualize” seçilir ve sonuç veri seti görselleştirilir.



Şekil 3.30: Birleştirilen veri setinin görselleştirilmesi

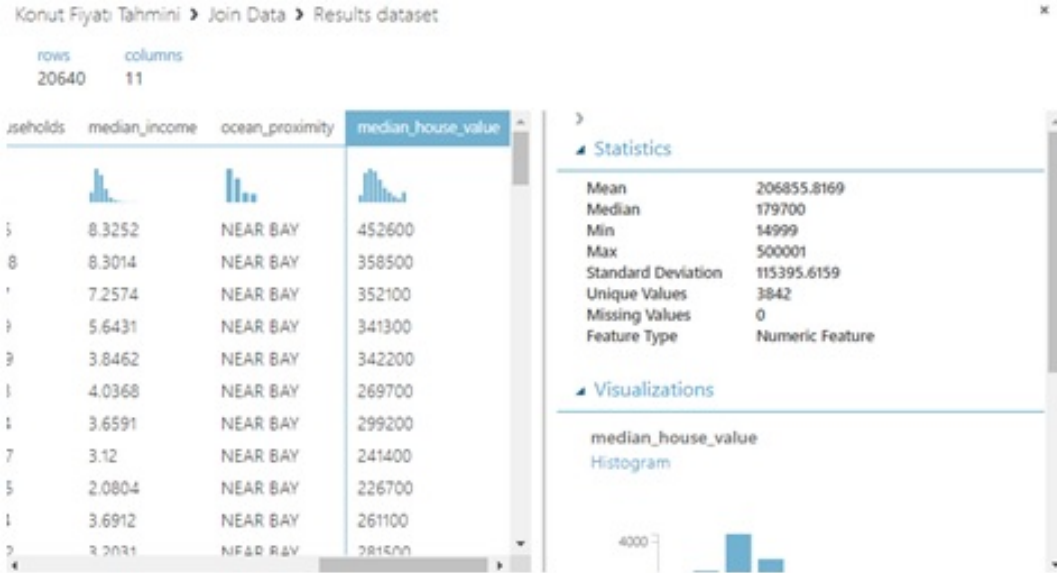
Şekil 31’de görüldüğü gibi oluşturulan yeni veri setinde tüm housing1.csv sütunları ve housing2.csv verisetinden karşılık gelen median_house_value sütunu bulunmaktadır.

Elde edilen veri seti başka uygulamalarda kullanılmak istenirse bilgisayara indirilebilir. Bunun için Search experiment items kutusuna Convert to CSV yazılır ve açılan listeden Convert to CSV modülü seçilip Join Data modülünün altına bırakılır. Join Data modülünün Results dataset çıktı bağlantı noktası ile Convert to CSV modülünün Dataset girdi bağlantı noktası Şekil 32’de gösterildiği gibi bağlanır.

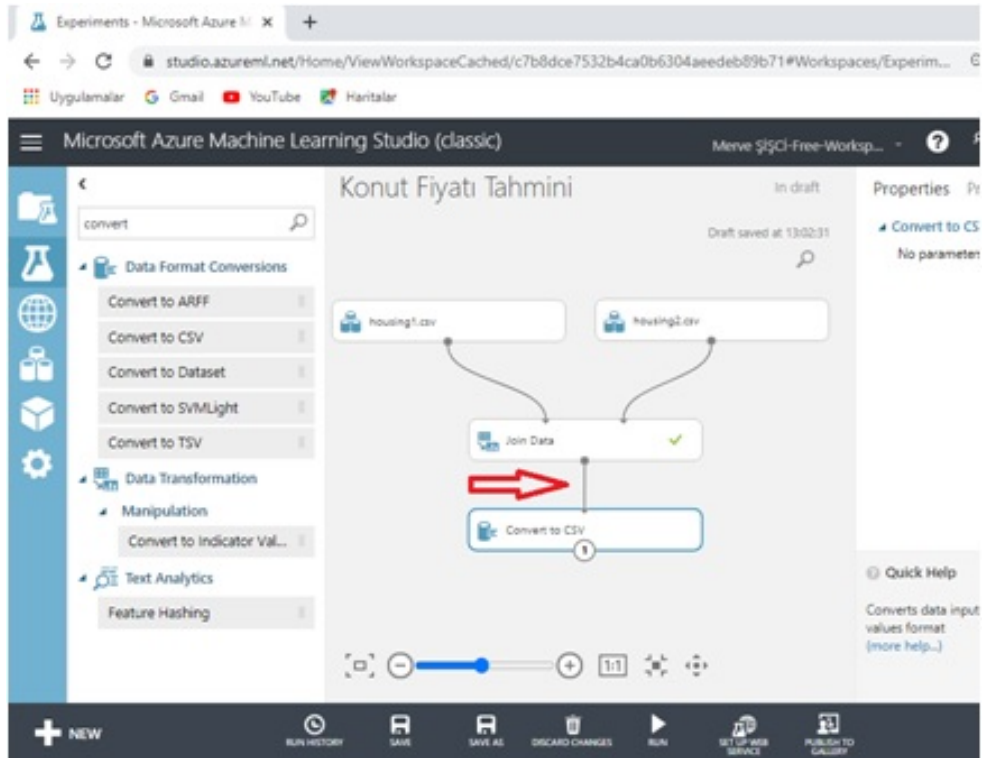
Convert to CSV modülü seçili iken **Run** menüsünden **Run Selected** seçeneği seçilir. Modülün çalışması bittikten sonra **Convert to CSV** modülünün **Result dataset** çıktısına sağ tıklanır ve Şekil 33’te gösterildiği gibi **Download** seçeneği seçilir.

3.4.3 Kategorik Niteliklerin Belirlenmesi

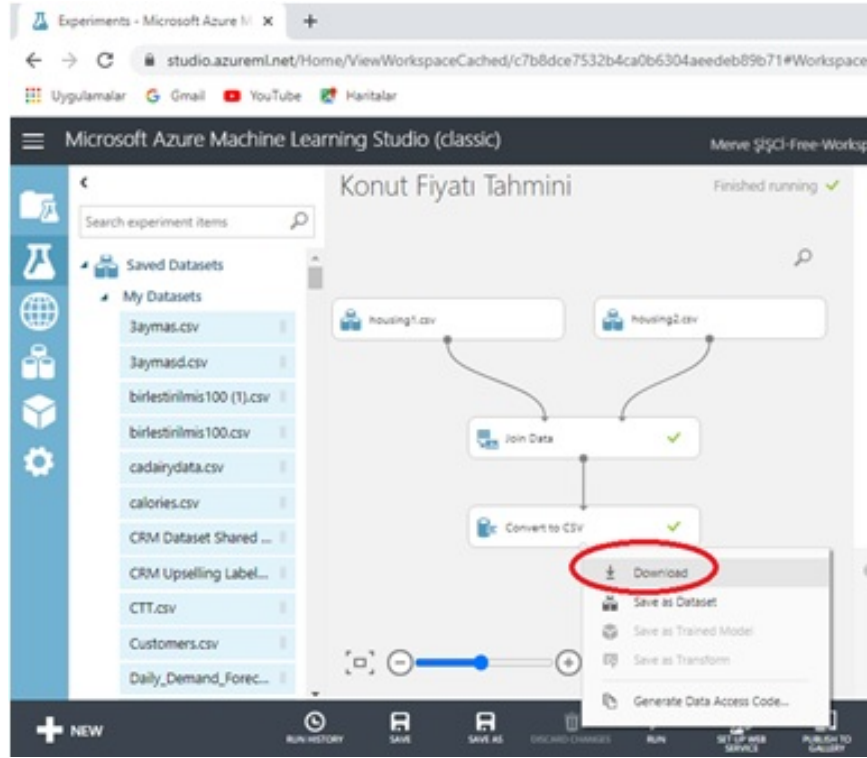
Her bir niteliğin türü önemlidir. Ham verilere göz atarak niteliklerin türleri hakkında fikir edinilebilir [25]. Microsoft Azure ML Studio’da metin tipindeki nitelikler ilk durumda String Feature olarak tanımlanmaktadır. Makine öğrenmesi çalışmasının daha sağlıklı olması için sisteme kategorik niteliklerin bildirilmesi gerekmektedir. Konut fiyatı tahmini veri setindeki ocean_proximity niteliği blok gruplarını kategorize eden bir nitelik türüdür. Ölçülebilir bir değer değildir. Bunu açık bir şekilde yapılandırmak, Azure Machine Learning’in bir tahmin modelini eğitirken niteliği uygun şekilde kullanmasına yardımcı olacaktır [26]. Bunun için çalışmaya deney öğeleri bölümünden bir **Edit**



Şekil 3.31: Bileştirilen veri setinin incelenmesi

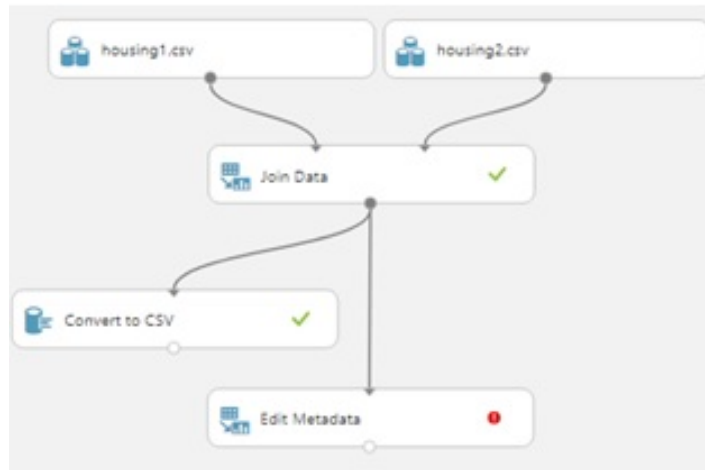


Şekil 3.32: Çalışmaya Convert to CSV modülünün eklenmesi



Şekil 3.33: Birleştirilen veri setinin indirilmesi

Metadata modülü eklenir ve **Join Data** modülünün çıktı bağlantısı ile **Edit Metadata** modülünün girdi bağlantısı Şekil 34'te gösterildiği gibi bağlanır.



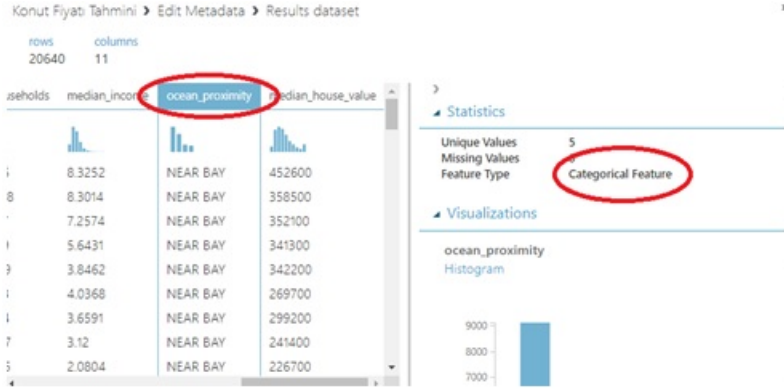
Şekil 3.34: Çalışmaya Edit Metadata modülü eklenmesi

Kategorik niteliklerin belirlenmesi için Edit Metadata modülünün özelliklerinde (Properties penceresinde) aşağıdaki değişiklikler yapılır:

- Column: ocean_proximity

- Data type: Unchanged
- Categorical: Make categorical
- Fields: Unchanged
- New column names: Boş bırakılır.

Özellikler tanımlandıktan sonra sırasıyla SAVE ve Run Selected butonlarına basılarak çalışma kaydedilir ve çalıştırılır. Modülün çalışması bittikten sonra Edit Metadata modülünün çıktı bağlantısına tıklanarak 'Visualize' seçilir ve veri seti görselleştirilirse, Şekil 35'te verildiği gibi ocean_proximity niteliğinin kategorik nitelik olarak belirlendiği görülür.



Şekil 3.35: Ocean_proximity niteliğinin kategorik nitelik olarak belirlenmesi

3.4.4 Niteliklerin İsimlendirilmesi

Konut fiyatı tahmini veri setindeki niteliklerin isimleri İngilizce olarak verilmiştir. Nitelik isimlerini Türkçe olarak belirlemek için çalışmaya **Edit Metadata** modülünü daha eklenir ve Şekil 36'da gösterildiği gibi ilk **Edit Metadata** modülünün çıktı bağlantısı ile ikinci **Edit Metadata** modülünün girdi bağlantısı bağlanır.

Edit Metadata modülünün özellikler penceresinden Launch Column Selector'dan isimleri değiştirilmek istenilen nitelikler seçilir ve onay (✓) işaretine basılır. Bu çalışmada tüm niteliklerin isimleri değiştirilmek istendiği için Şekil 37'de gösterildiği gibi tüm nitelikler seçilmiştir.

Özellikler penceresinde New column names kutusuna Şekil 38'de gösterildiği gibi virgülle ayrılmış aşağıdaki sütun isimleri yazılır:

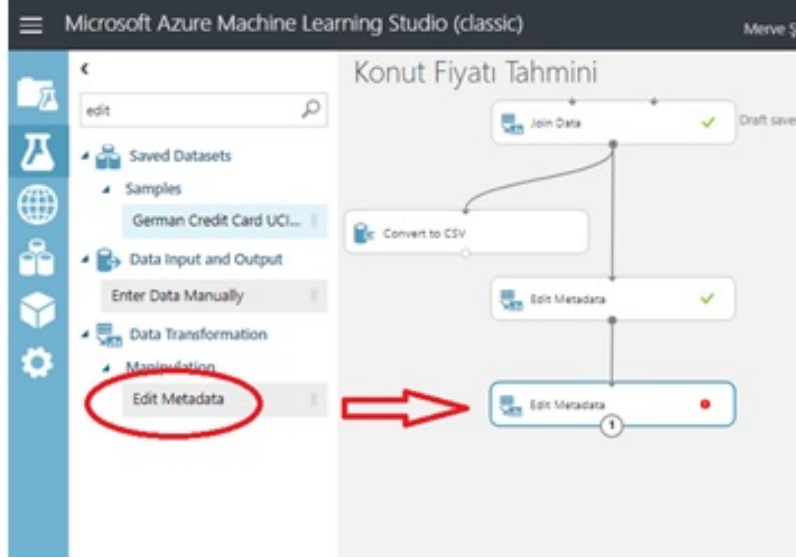
blok_id, boylam, enlem, konut_medyan_yasi, toplam_oda, toplam_yatak_odasi, nufus, hane, medyan_gelir, okyanus_yakinligi, medyan_ev_degeri

Burada önemli olan sütun isimlerinin Launch Column Selector bölümünde seçildiği sıra ile yazılmasıdır. Daha sonra sırasıyla SAVE ve RUN butonlarına basılarak deney kaydedilir ve çalıştırılır.

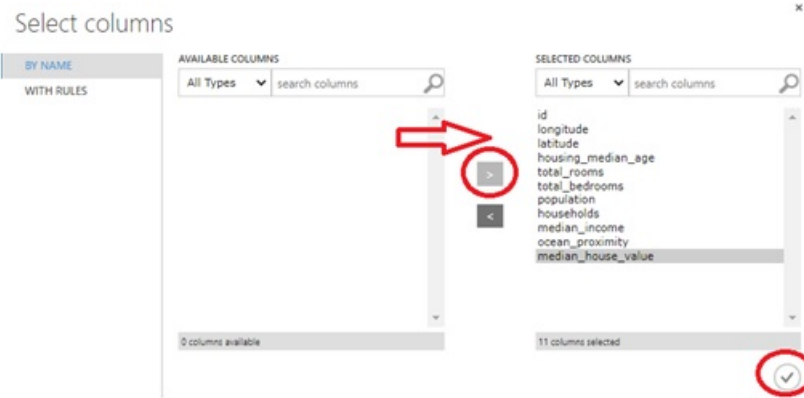
Edit Metadata modülünün çıktı bağlantısına tıklanılarak 'Visualize' seçeneği seçilip veri seti görselleştirildiğinde yeni nitelik isimleri Şekil 39'da gösterildiği gibi görünmektedir.

3.4.5 Eksik Verilerin Tamamlanması

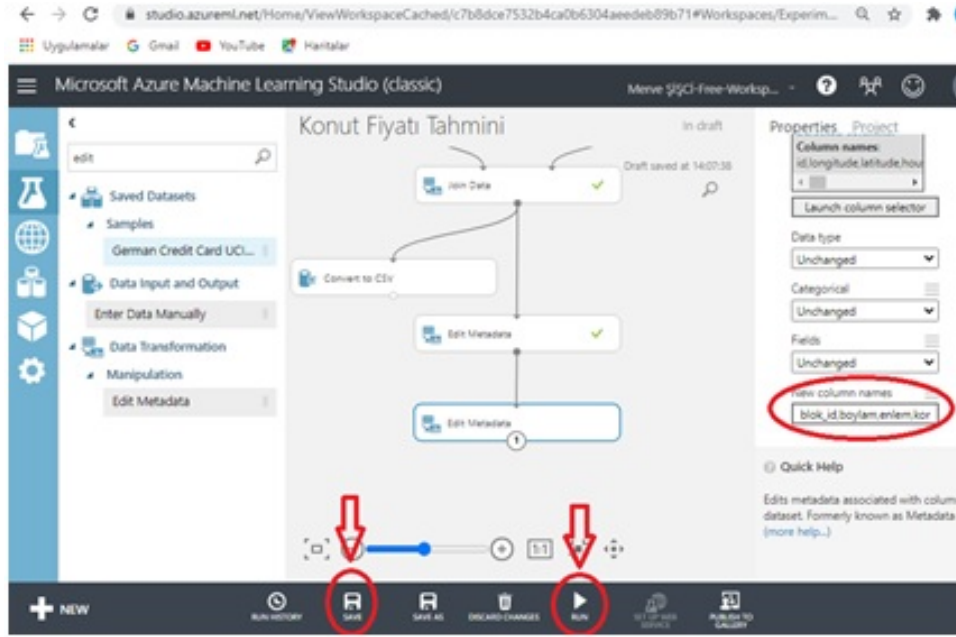
Kalitesiz verilerden neyin öğrenilebileceği konusunda sınırlamalar bulunmaktadır. Kaliteyle ilgili sorunlar, diğer faktörlerin yanı sıra gürültülü verileri, eksik değerleri ve etiketlemedeki hataları içerebilir [5]. Gerçek dünya veri setlerinde, çoğu niteliğin eksik değerleri bulunmaktadır. Bazı durumlarda, ölçüm hataları, kayıttaki gecikmeler veya çeşitli koşullar nedeniyle kullanılmadığından



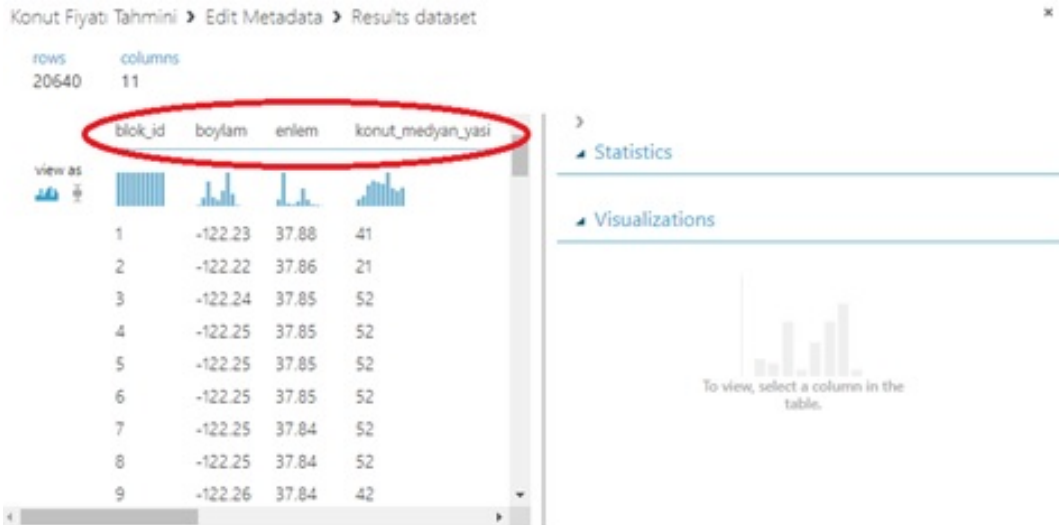
Şekil 3.36: Nitelik isimlerini değiştirmek için çalışmaya Edit Metadata modülü eklenmesi



Şekil 3.37: İsimleri değiştirilmek istenilen niteliklerin seçilmesi

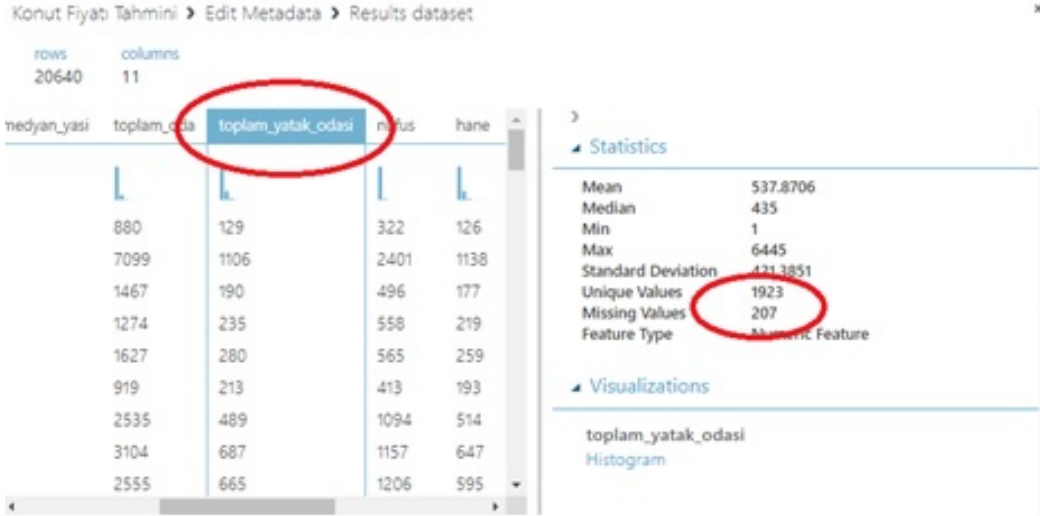


Şekil 3.38: Yeni nitelik isimlerinin yazılması



Şekil 3.39: Yeni nitelik isimlerinin görselleştirilmesi

eksik olabilirler; örneğin, bir anket çalışmasında bireyler yaşını veya mesleğini açıklamamayı tercih edebilirler. Regresyon çalışmalarında, eksik veriler, örnekleme verileri için uygun olmayan bir modele yol açabilir [27]. Edit Metadata modülünün çıktı bağlantısı görselleştirildiğinde Şekil 40'ta verildiği gibi, toplam_yatak_odasi niteliğinin üzerine tıklandığında istatistiksel özelliklerinde 207 adet eksik değer olduğu görülmektedir.



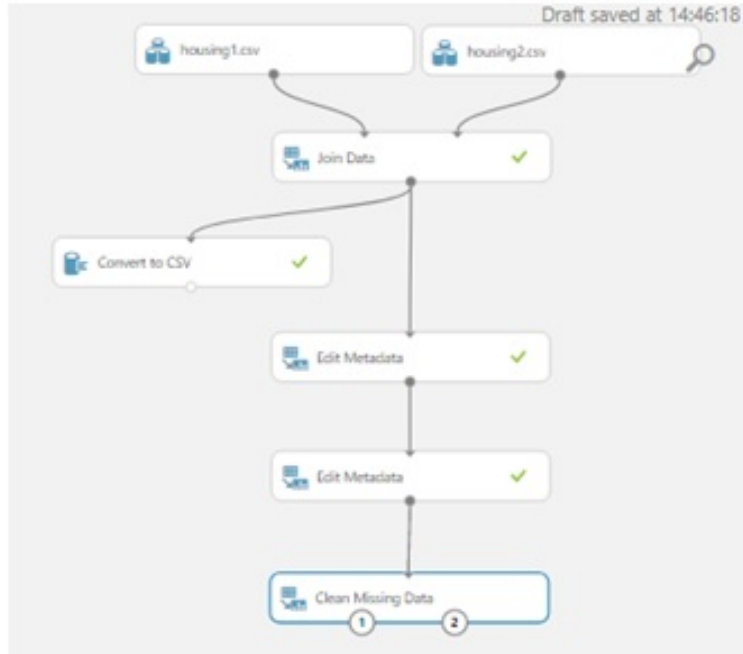
Şekil 3.40: toplam_yatak_odasi niteliğindeki eksik veriler

Eksik verileri ele almanın bir yolu, eksik değerleri olan kayıtları göz ardı etmek, başka bir deyişle, bu örnekleri veya nitelikleri silmektir. Bu yaklaşım, verilerde eksik nitelikler yaygın olduğunda veri setinin boyutunu ciddi şekilde azaltabilir. Uğraşılacak sistem karmaşıksa, veri seti boyutu değerli bir avantaj sağlayabilir. Ayrıca, sorunla başa çıkmak için uygun önlemler kullanıldığı sürece, eksik değerlere rağmen "bütün olmayan" kayıtlarda bile sıklıkla yararlanılabilecek tahmini değer vardır [5]. Azure ML Studio eksik verileri ele almak için **Clean Missing Data** modülü sunmaktadır. Bu modülde eksik veriler için, eksik değer içeren satırları silmek, eksik değer içeren sütunları silmek, ortalama ile değiştirmek, mod ile değiştirmek, medyan ile değiştirmek, özel bir değer ile değiştirmek gibi farklı seçenekler bulunmaktadır. Kategorik değerlere sahip nitelikler söz konusu olduğunda, eksik değerlerin örnekte en yüksek frekansla oluşan değerle yani mod ile değiştirilmesi makul bir seçim sağlar. Sürekli değer niteliğinin eksik değeri, o niteliğin ortalama değeriyle değiştirildiğinde, yeni ortalama aynı kalır [5]. Ancak, ortalama, aykırı değerlerden oldukça etkilenir, çünkü aykırı değerlerin getirdiği çarpıklık ortalamayı uç değerlere doğru çekecektir. Medyan, ortalama ve modun yanı sıra merkezi konumun bir ölçüsüdür ve verilerde aykırı değerlerin varlığından daha az etkilenir [27].

Eksik veriler ile ilgili işlem yapılmak istendiğinde, çalışmaya bir **Clean Missing Data** modülü eklenir. İkinci Edit Metadata modülünün çıktı bağlantı noktası ile Clean Missing Data modülünün girdi bağlantı noktası Şekil 41'de görüldüğü gibi bağlanır.

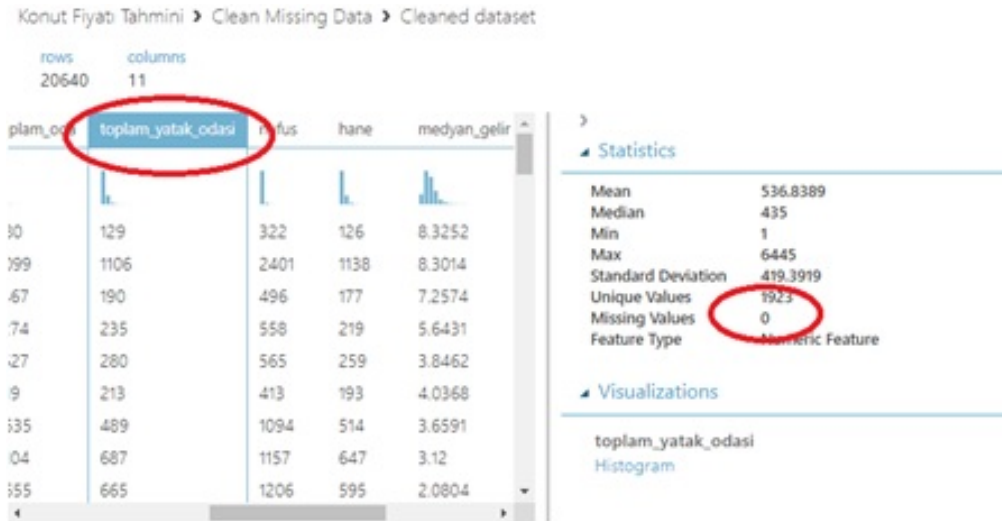
Yukarıda açıklanan nedenlerden dolayı, bu çalışmada toplam_yatak_odasi niteliğindeki eksik veriler niteliğinin medyan değeri ile tamamlanmaktadır. Eksik verilerin tamamlanması için **Clean Missing Data** modülünün özelliklerinde aşağıdaki değişiklikleri yapılır:

- Column: toplam_yatak_odasi
- Cleaning mode: Replace with median



Şekil 3.41: Deneye Clean Missing Data modülü eklenmesi

Deney SAVE ve RUN butonları ile kaydedilip çalıştırılır. **Clean Missing Data** modülünün sol çıktısı bağlantısına basılıp 'Visualize' seçeneği seçilerek görselleştirildiğinde, Şekil 42'de verildiği gibi artık toplam_yatak_odasi niteliğinde eksik verilerin olmadığı görülmektedir.

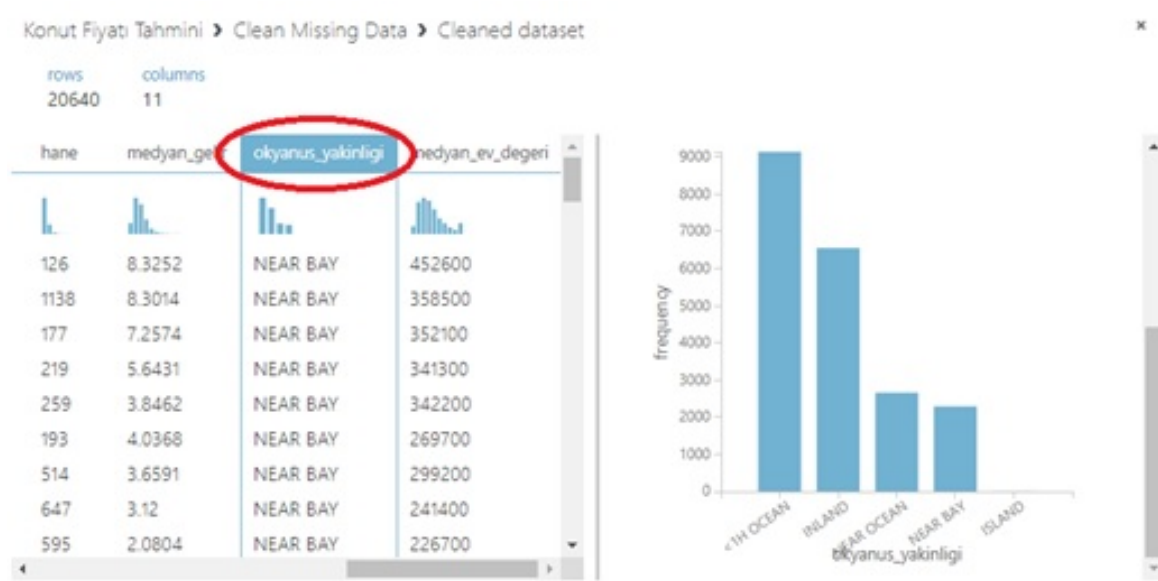


Şekil 3.42: Tamamlanmış verilerin görüntülenmesi

3.4.6 Örnek Seçimi

Tekrar **Clean Missing Data** modülünün çıktı bağlantısına basılıp Visualize seçeneği seçilerek görselleştirildiğinde, Şekil 43'te gösterildiği gibi **okyanus_yakinaligi** niteliği incelendiğinde ISLAND

değeri dışında her bir nitelik değerinin (<1H OCEAN, INLAND, NEAR OCEAN, NEAR BAY) 2000'den fazla örnekte yer aldığı fakat ISLAND değerinin sadece 5 örnekte bulunduğu görülmektedir. Modelin performansının kötü etkilenmemesi için ISLAND kategorisinde yer alan örnekler veri setinden çıkartılır.



Şekil 3.43: okyanus_yakirligi niteliğinin incelenmesi

Örnek seçimi için çalışmaya bir **Split Data** modülü eklenir. Clean Missing Data modülünün sol çıktı bağlantısı ile Split Data modülünün girdi bağlantısı Şekil 44'te gösterildiği gibi bağlanır.

Split Data modülünün özelliklerinde aşağıdaki değişiklikler yapılır:

- **Splitting mode:** Regular Expression
- **Regular expression:** \ "okyanus_yakirligi" ^ ISLAND

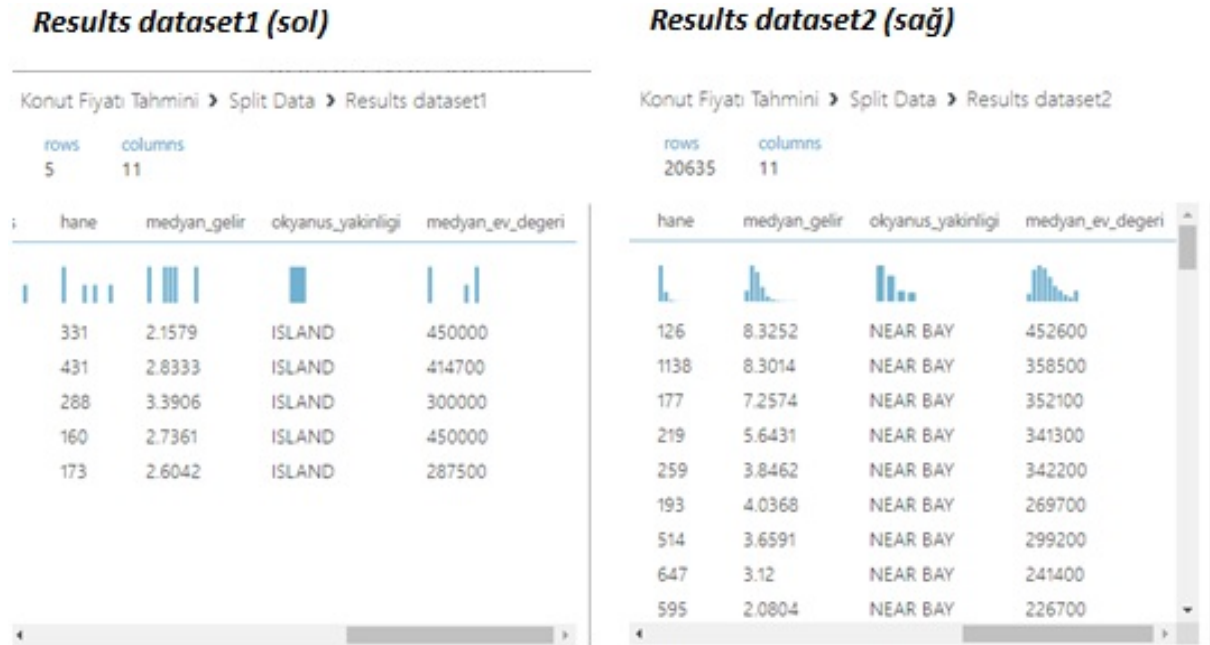
Burada, düzenli ifade (Regular Expression) bir metin parçası içinde aranacak bir dizeyi veya modeli ifade eden bir dizi sembol ve karakterdir. "okyanus_yakirligi" ^ ISLAND ifadesi ile okyanus_yakirligi niteliğindeki değerler içerisinde ISLAND ifadesi içeren örnekler Split Data modülünün sol çıktı bağlantısına, içermeyen örnekler ise sağ çıktı bağlantısına ayrılır. Çalışma SAVE ve RUN butonları ile kaydedilip çalıştırılır. Sırasıyla Split Data modülünün Results dataset1(sol) çıktı bağlantısı ile Results dataset2(sağ) çıktı bağlantısı tıklanıp Visualize seçeneği ile görselleştirildiğinde Şekil 45'te gösterildiği gibi veri setinin ISLAND ifadesi içeren örnekler ve içermeyen örnekler olmak üzere iki bölüme ayrıldığı görülmektedir.

3.4.7 Yeni Niteliklerin Elde Edilmesi

Bazı veri setlerinde, özellikleri ve tahmin edilmeye çalışılan etiket arasındaki ilişkileri geliştirmek için var olan özelliklerin değiştirilmesi veya önceden verilen niteliklerden matematiksel dönüşümlere dayanarak daha fazla nitelik oluşturulması gerekebilir [5, 26]. Öznitelik oluşturma süreci, makine öğrenmesi terminolojisinde özellik çıkarma ve nitelik mühendisliği (feature engineering) olarak da bilinir. Öznitelik oluşturma, temel olarak bazı kurallara, mantığa veya hipoteze dayalı olarak mevcut özniteliklerden yeni öznitelikler veya değişkenler oluşturmaktır. Örneğin, bir kuruluştaki çalışanların bir veri setindeki iki tarih-saat alanı (mevcut_tarih ve doğum_tarihi) kullanılarak yaş adı verilen



Şekil 3.44: Çalışmaya Split Data modülünün eklenmesi



Şekil 3.45: Verisetindeki örneklerin ayrılması

makine öğrenmesi çalışması için daha faydalı olabilecek yeni bir sayısal değişken oluşturulabilir [3]. Genellikle, ek özellikler oluşturmak için ortalama, toplam, minimum veya maksimum gibi ortak toplayıcılar kullanılarak bazı toplama biçimleri yapılır [5]. Bu çalışmada, toplam_oda, nüfus ve toplam_yatak_odasi nitelikleri hane niteliğine bölünerek hane başına toplam oda sayısı, hane başına kişi sayısı, hane başına toplam yatak odası sayısı değerleri bulunmaktadır. Bunun için çalışmaya bir Apply Math Operation modülü eklenir ve Split Data modülünün Results dataset2 (sağ) çıktı bağlantısı Şekil 46’da görüldüğü gibi Apply Math Operation modülünün girdi bağlantısı ile bağlanır.



Şekil 3.46: Çalışmaya Apply Math Operation modülünün eklenmesi

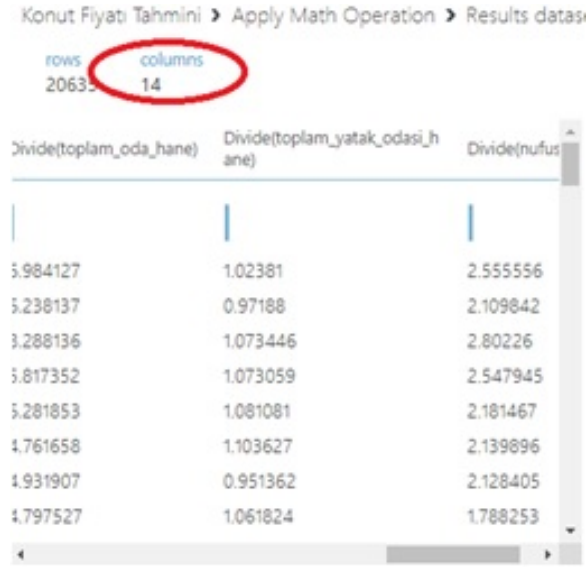
Apply Math Operation modülünün özelliklerinde (Properties penceresinde) aşağıdaki değişiklikler yapılır:

- Category: Operations
- Basic operation: Divide
- Operation argument type: ColumnSet
- Operation argument / Launch column selector: hane
- Column set / Launch column selector: toplam_oda, nüfus, toplam_yatak_odasi
- Output mode: Append

Konut fiyatı tahmini deneyi sırasıyla SAVE ve RUN butonlarına basılarak kaydedilip çalıştırılır. Apply Math Operation modülünün çıktı bağlantısı tıklanarak ‘Visualize’ seçeneği seçilip görselleştirildiğinde, Şekil 47’de gösterildiği gibi veri setine Divide(toplam_oda_hane), Divide(nüfus_oda_hane) ve Divide(toplam_yatak_odasi_hane) isimli üç yeni niteliğin eklendiği, nitelik sayısının 11’den 14’e yükseldiği görülmektedir.

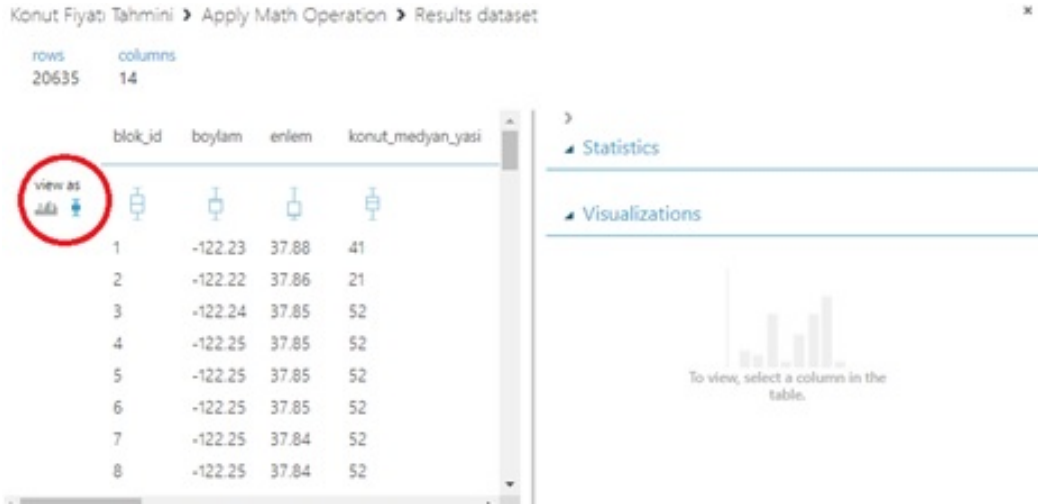
3.4.8 Aykırı Verilerin Ele Alınması

Verilerdeki aykırı değerler, serideki diğer değerlerden farklı olan ve tüm öğrenme yöntemlerini çeşitli derecelerde etkileyen değerlerdir. Bu değerler, güven aralıklarıyla tespit edilebilen ve eşik ile kaldırılabilen uç değerler olabilmektedir [28]. Genellikle, ekipman arızası nedeniyle veya veri girişi hataları nedeniyle doğal olarak meydana gelmektedirler [27]. Aykırı değerleri ele almak çok fazla



Şekil 3.47: Yeni oluşturulan niteliklerin görselleştirilmesi

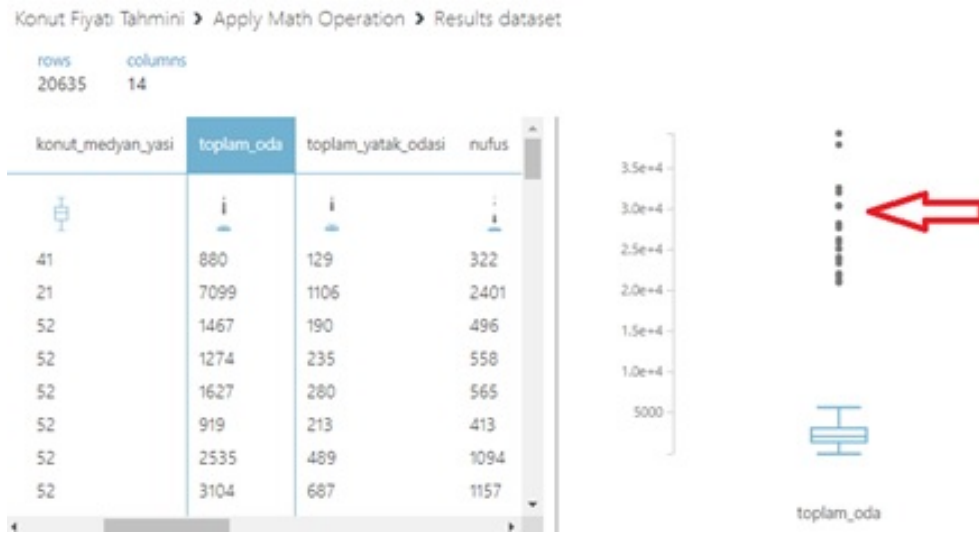
özen ve analiz gerektirir [5]. En iyi yaklaşım, verileri görselleştirmek ve düzensizlikleri tespit etmek için görselleştirmeyi incelemektir [28]. Kutu grafikleri, yüzdeler ve aykırı değerler açısından dağılımları gösterdikleri için sayısal özellikler için önemli bir görselleştirme tekniğidir [5]. Veri setinde aykırı değerlerin bulunup bulunmadığını incelemek için Apply Math Operation modülünün çıktı bağlantısı tıklanarak 'Visualize' seçeneği seçilip görselleştirilir. Şekil 48'de görüldüğü gibi View as kısmında box plot seçeneği seçilir.



Şekil 3.48: Görselleştirmede kutu grafiğinin seçilmesi

Kutu grafiği, sayısal bir değişkenin merkezini ve yayılmasını, bir değişkenin aralığı ve çarpıklığını hızlı bir şekilde analiz etmeye veya onu diğer değişkenlerle karşılaştırmaya olanak tanıyan bir formatta görüntüler [29]. Kartil aralığı (IQR), verilerdeki veya eşdeğer olarak istatistiksel dağılımdaki değişkenliğin bir ölçüsüdür. Her sayısal nitelik, veri setindeki değerine göre sıralanır ve sıralı

küme daha sonra çeyreklere bölünür. Kartil aralığı, 3. kartil (Q3) ile 1. kartil (Q1) arasındaki fark olup serinin ortasındaki %50'lik gözlem kümesinin değişim aralığını verir [5]. Minimum ve maksimum değerler, kutunun altına ve üstüne uzanan bıyıklar kullanılarak gösterilebilir; bununla birlikte, yaygın olarak kullanılan bir kural, bıyıkların yalnızca Q1'in altında veya Q3'ün üzerinde kartil aralığının minimum veya maksimum 1.5 katına kadar uzanmasına izin verir [29]. Aykırı değerler genellikle $Q3 + 1.5 \times IQR$ üzeri ve $Q1 - 1.5 \times IQR$ altındaki veri değerleri olarak kabul edilir [5]. toplam_oda niteliği seçilip sağ tarafta Visualizations alanında Box plot bölümü incelenir. Şekil 49'da görüldüğü gibi nitelikte aykırı veriler bulunmaktadır. Aynı şekilde konut_medyan_yasi, toplam_yatak_odasi, nufus, hane, medyan_gelir, medyan_ev_degeri, Divide(toplam_oda_hane), Divide(toplam_yatak_odasi_hane) ve Divide(nufus_hane) nitelikleri de incelendiğinde aykırı verilerin bulunduğu anlaşılmaktadır.



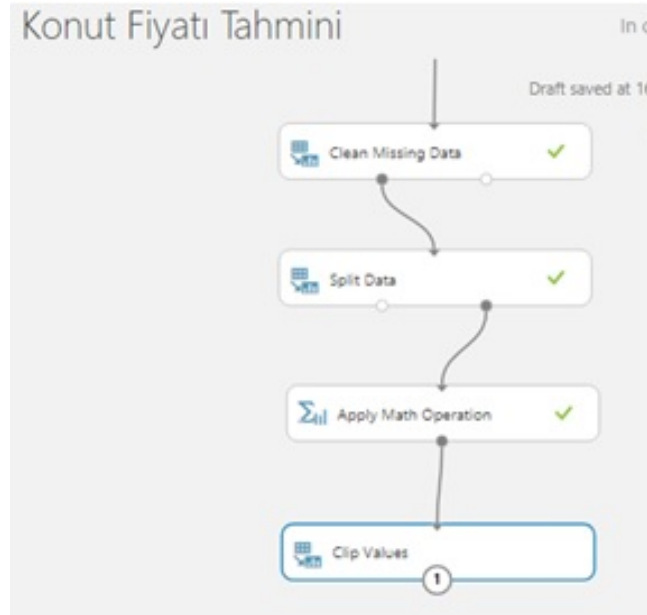
Şekil 3.49: toplam_oda niteliğindeki aykırı verilerin incelenmesi

Aykırı değerleri kaldırmak veya kırpma için, ML Studio'da Clip Values modülü kullanılmaktadır. Clip Values modülü, bir veri setini girdi olarak kabul eder ve seçilen sütunlar için belirtilen bir sabit veya yüzdelik dilim kullanarak belirli bir eşiği aşan veri noktaları değerlerini kırabilir. Hem üst tepe hem de alt tepe aykırı değerler, eşik, ortalama, medyan veya eksik (boş) bir değer ile değiştirilebilir. Aykırı değeri neyin oluşturduğunu belirlemenin birden çok yolu vardır ve hangi yöntemin kullanılmak istendiği belirtilmelidir [6]:

- ClipPeak, belirtilen bir üst sınır değerinden daha büyük olan değerleri arar ve sonra kırpar veya değiştirir.
- ClipSubpeaks, belirli bir alt sınır değerinden daha küçük değerleri arar ve sonra kırpar veya değiştirir.
- ClipPeaksAndSubpeaks, değerlere hem alt hem de üst sınır uygular ve bu aralığın dışındaki tüm değerleri kırpar veya değiştirir.

Bu çalışmada aykırı veriler üst ve alt eşik değerlerine eşitlenmektedir. Bunun için çalışmaya bir **Clip Values** modülü eklenir ve Şekil 50'de gösterildiği gibi girdi bağlantısı Apply Math Operation modülünün çıktı bağlantısı bağlanır.

Sayısal niteliklerdeki aykırı verilerin niteliğin üst ve alt eşik değerlerine eşitlenmesi için Clip Values modülünün özelliklerinde (properties penceresinde) aşağıdaki değişiklikler yapılır:



Şekil 3.50: Çalışmaya Clip Values modülünün eklenmesi

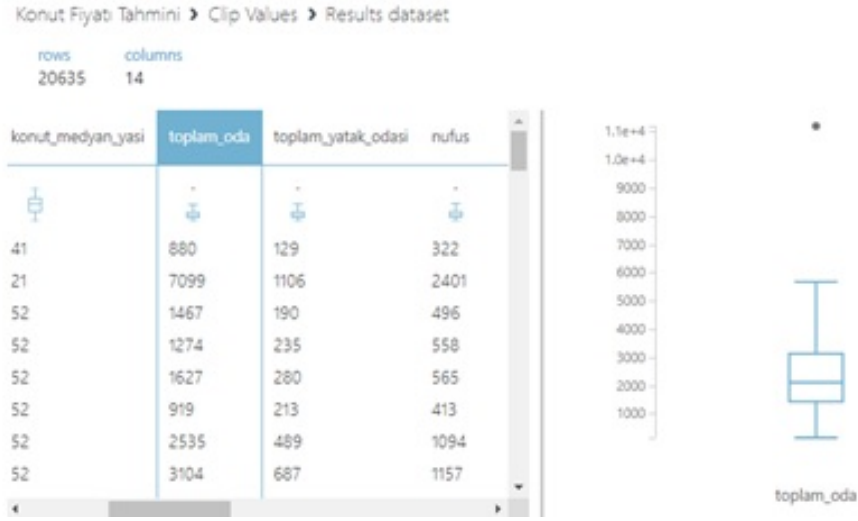
- **Set of tresholds:** ClipPeaksAndSubpeaks
- **Tresholds:** Percentile
- **Percentile number of upper treshold:** 99
- **Percentile number of lower treshold :** 1
- **Substitute value for peaks:** Treshold
- **Substitute value for subpeaks :** Treshold
- **Launch column selector:** konut_medyan_yasi, toplam_oda, toplam_yatak_odasi, nufus, hane, medyan_gelir, medyan_ev_degeri, Divide(toplam_oda_hane), Divide(toplam_yatak_odasi_hane), Divide(nufus_hane)

Çalışma sırasıyla SAVE ve RUN butonlarına basılarak kaydedilip çalıştırılır. Clip Values modülünün çıktı bağlantısına basılıp Visualize seçeneği seçilerek sonuç veri seti görselleştirilir. View as kısmında box plot seçeneği seçilir. toplam_oda niteliği seçilip sağ taraftan kutu grafiği (box plot) bölümü incelendiğinde, Şekil 51'de görüldüğü gibi aykırı verilerin azaldığı anlaşılmaktadır.

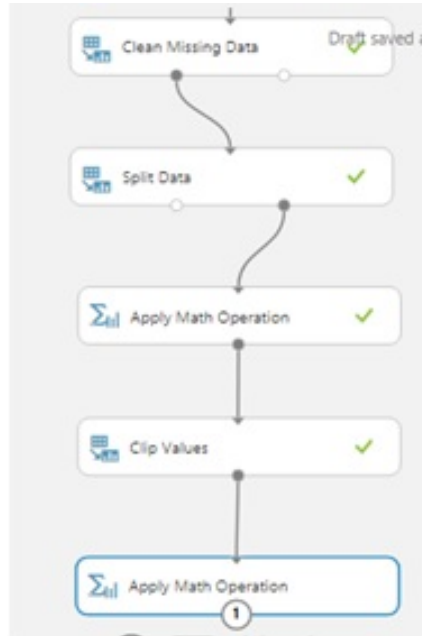
3.4.9 Nitelik Dönüşümü

Veri dönüştürme teknikleri, veri kümesini bir makine öğrenmesi algoritmasının girdi olarak beklediği bir biçime uydurur ve hatta algoritmanın daha hızlı öğrenmesine ve daha iyi performans elde etmesine yardımcı olabilir [28]. Konut fiyatı tahmini makine öğrenmesi çalışmasındaki amaç, bir bloktaki medyan ev değerini tahmin etmektir. Medyan ev değeri kesinlikle pozitif bir miktardır. Negatif bir medyan ev değeri olamaz. Bu nedenle, regresyon modelinin herhangi bir değer (negatif bir değer) tahmin etmediğinden emin olmak için, medyan_ev_değeri etiketine doğal logaritma dönüşümü uygulanır [26]. Bunun için çalışmaya Şekil 52'de gösterildiği gibi bir **Apply Math Operation** modülü eklenir ve girdi bağlantı noktası Clip Values modülünün çıktı bağlantı noktası ile bağlanır.

medyan_ev_değeri niteliğine doğal logaritma dönüşümü uygulamak için eklenen yeni Apply Math Operation modülünde aşağıdaki değişiklikler yapılır:



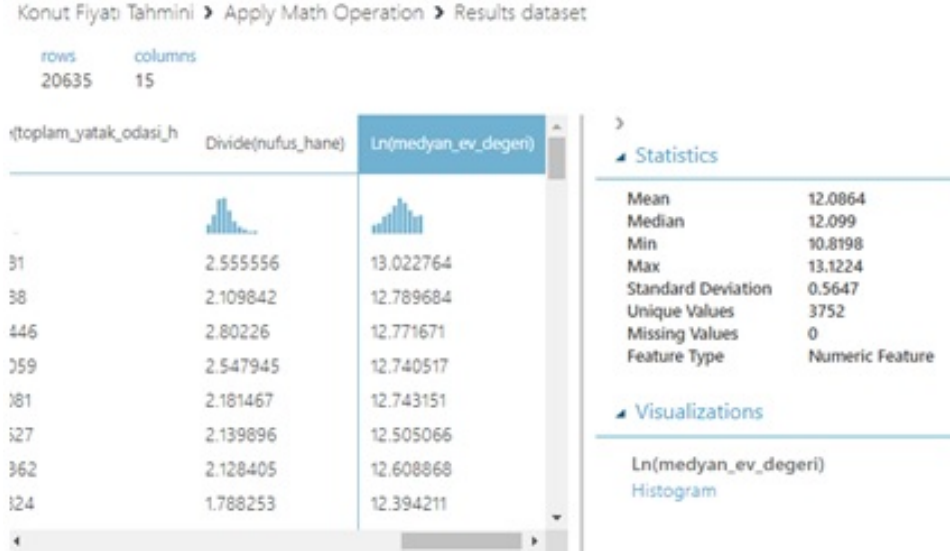
Şekil 3.51: Clip Values modülü eklendikten sonra aykırı verilerin incelenmesi



Şekil 3.52: Çalışmaya Apply Math Operation modülünün eklenmesi

- **Category:** Basic
- **Basic math function:** Ln
- **Column set:** medyan_ev_degeri
- **Output mode:** Append

Konut fiyatı tahmini deneyi sırasıyla SAVE ve RUN butonlarına basılarak kaydedilip çalıştırılır. **Apply Math Operation** modülünün çıktı bağlantısı ‘Visualize’ seçeneği ile görselleştirildiğinde, Şekil 53’te görüldüğü gibi veri setine **Ln(medyan_ev_degeri)** isimli yeni bir sütun eklendiği görülmektedir.



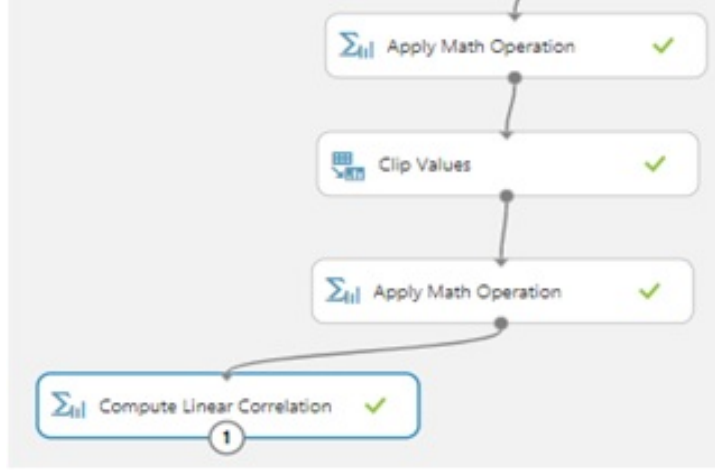
Şekil 3.53: Yeni oluşturulan Ln(medyan_ev_degeri) niteliğinin incelenmesi

3.4.10 Niteliklerin Belirlenmesi

Niteliklerin sayısı, veri setindeki boyutların sayısına karşılık gelir. Düşük tahmin gücüne sahip boyutlar, genel modele çok az katkıda bulunmakla kalmaz, aynı zamanda çok fazla zarara da neden olabilir. Örneğin, rastgele değerlere sahip bir nitelik, bir makine öğrenme algoritması tarafından alınacak bazı rastgele kalıplar sunabilir [28]. Bu adımdaki temel amaç, tahmin modelinin mümkün olan en iyi tahminleri vermesini, hata oranlarını en aza indirmesini ve doğruluğu en üst düzeye çıkarmasını sağlayan eğitim veri setinden bir dizi öz nitelik veya özellik seçmektir. Nitelik seçiminin uygulanmasının diğer nedenlerinden bazıları; çok fazla bilgi kaybı olmadan gereksiz veya ilgisiz özellikleri kaldırmak, çok fazla özellik kullanarak modellerin aşırı uyumunu önlemek, modelin fazla özelliklerden kaynaklanan varyansını azaltmak, eğitim süresinin ve modellerin yakınsama süresini azaltmak, basit ve yorumlaması kolay modeller oluşturmaktır [29].

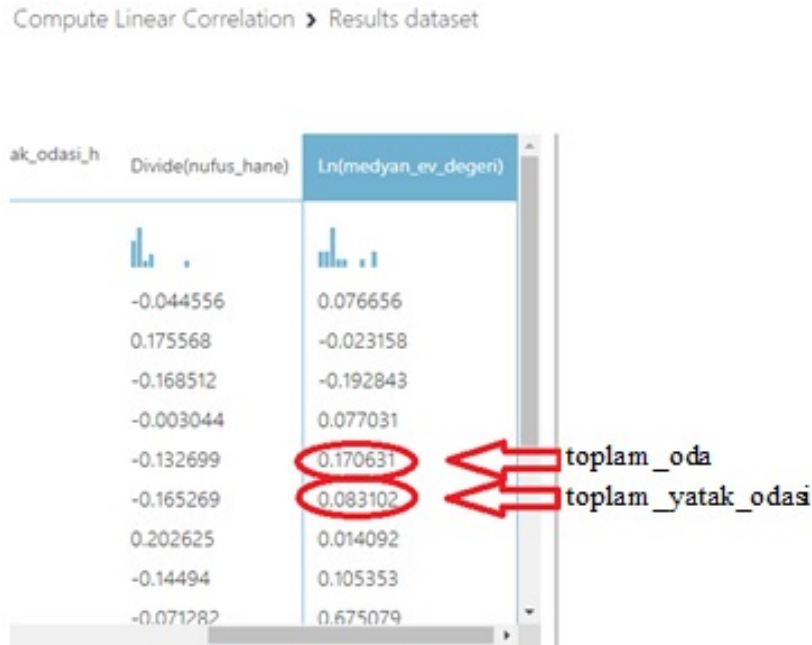
Öncelikle keşif değişkenlerinden herhangi birinin yanıt değişkeniyle ilişkili olup olmadığı incelenmelidir. Çünkü iki niceliksel varlık arasındaki yüksek korelasyon daha iyi uyan bir doğrusal regresyon çizgisi sağlayacaktır. Bu nedenle, veri kümesindeki niteliklerin tüm ikili korelasyonlarını gözden geçirmek önemlidir [25]. Bu nicel nitelikler arasındaki ilişkilerin gücünün ve yönünün belirlenmesi gerekmektedir [27]. Korelasyon, iki değişken arasındaki ilişkiyi ve bunların birlikte nasıl değişip değişmeyebileceğini ifade eder [25, 30]. Bölüm 4.7’de toplam_oda ve toplam_yatak_odasi nitelikleri hane niteliğine bölünerek konut başına oda sayısı (Divide(toplam_oda_hane)) ve konut

başına yatak odası sayısı (Divide(toplam_yatak_odasi_hane)) nitelikleri oluşturulmuştu. Niteliklerin tahmin edilmek istenilen hedef niteliği olan Ln(medyan_ev_degeri) niteliği ile ilişkisini incelemek için çalışmaya bir Compute Linear Correlation modülü eklenir. Apply Math Operation modülünün çıktı bağlantısı ile Compute Linear Correlation modülünün girdi bağlantısı Şekil 54'te gösterildiği gibi bağlanır. Sırasıyla SAVE ve RUN butonlarına basılarak deney kaydedilip çalıştırılır.



Şekil 3.54: Çalışmaya Compute Linear Correlation modülünün eklenmesi

Compute Linear Correlation modülünün çıktı bağlantısı Visualize seçeneği seçilerek görselleştirilir.



Şekil 3.55: Niteliklerin Ln(medyan_ev_degeri) ile korelasyon katsayılarının incelenmesi

Güçlü bir pozitif korelasyonun değeri 1'dir, güçlü negatif korelasyonun değeri -1'dir ve 0 ise

korelasyon olmadığını gösterir [27]. Şekil 55'te gösterildiği gibi niteliklerin Ln(medyan_ev_degeri) niteliği ile ilişkileri incelendiğinde, toplam_oda niteliğinin korelasyon katsayısının 0.170631 değeri ile 0.255615 korelasyon katsayısına sahip Divide(toplam_oda_hane) niteliğinden daha düşük bir ilişkiye sahip olduğu görülmektedir. Aynı şekilde toplam_yatak_odasi niteliğinin korelasyon katsayısı 0.083102 iken Divide(toplam_yatak_odasi_hane) niteliğinin korelasyon katsayısı -0.123652 değerindedir. Korelasyon katsayısı negatif bir değer olmasına rağmen Divide(toplam_yatak_odasi_hane) niteliğinin Ln(medyan_ev_degeri) ilişkisi daha güçlüdür. Bu nedenle çalışmadan toplam_oda ve toplam_yatak_odasi nitelikleri çıkarılmaktadır. Bunun için çalışmaya bir **Select Columns in Dataset** modülü eklenir ve Şekil 56'da gösterildiği gibi girdi bağlantısı ikinci **Apply Math Operation** modülünün çıktı bağlantısı ile bağlanır. (Korelasyon katsayıları incelendikten sonra istenilirse Compute Linear Correlation modülüne sağ tıklayıp "Delete" tuşuna basılarak modül silinebilir.)



Şekil 3.56: Çalışmaya Select Columns in Dataset modülünün eklenmesi

Select Columns in Dataset modülünün özellikler penceresinde Launch Columns Selector bölümünden veri setinde kalması istenilen blok_id, boylam, enlem, konut_medyan_yasi, nüfus, hane, medyan_gelir, okyanus_yakinligi, medyan_ev_degeri, Divide(toplam_oda_hane), Divide(toplam_yatak_odasi_hane), Divide(nufus_hane), Ln(medyan_ev_degeri) nitelikleri seçilir.

Veri seti bir **blok_id** sütunu ile dönüştürülmemiş **medyan_ev_degeri** sütunu içermektedir. Bu sütunların hiçbirisi model eğitilirken nitelik olarak kullanılmamalıdır. Çünkü **blok_id** her bir blok grubu için benzersiz bir tanımlayıcıdır ve medyan ev değerleri ile herhangi bir ilişkisi yoktur. Diğer taraftan **medyan_ev_degeri** tahmin etmeye çalışılan orijinal etikettir. Bu niteliklerin çalışmanın geri kalan kısımlarında görüntülenmesi, fakat modeli hiçbir şekilde etkilememesi isteniliyorsa, çalışmaya başka bir **Edit Metadata** modülü eklenir ve girdi bağlantısı **Select Columns in Dataset** modülünün çıktı bağlantısı ile Şekil 57'de görüldüğü gibi bağlanır.

blok_id ve **medyan_ev_degeri** sütunlarının modelin eğitimini karıştırmamasını engellemek için **Edit Metadata** modülünün özellikleri şu şekilde ayarlanır:

- **Column:** blok_id ve medyan_ev_degeri
- **Data type:** Unchanged
- **Categorical:** Unchanged



Şekil 3.57: Çalışmaya Edit Metadata modülünün eklenmesi

- **Fields:** Clear feature
- **New column names:** Boş bırakılır

3.4.11 Verilerin Normalize Edilmesi

Bazı nitelikler diğerlerinden çok daha geniş bir değer aralığına sahipse, daha geniş aralıklara sahip nitelikler modelin performansına büyük ölçüde hakim olacaktır [29]. Verileri normalize etmenin amacı, tüm nitelikler için tek bir değer aralığı oluşturmaktır. Böylece model belirli bir niteliği desteklemez [31]. Tablo 3 incelendiğinde, konut fiyatı tahmini veri setinin çok farklı ölçekteki sayısal değerlere sahip olan niteliklerden oluştuğu görülmektedir. Bu durum, modelleme sırasında değerler özellikler olarak birleştirildiğinde deneye potansiyel olarak hata ekleyebilir veya deneyi bozabilir. Değerleri ortak bir ölçekte olacak şekilde dönüştürerek, ancak genel dağılımlarını ve oranlarını koruyarak, modelleme sırasında genellikle daha iyi sonuçlar elde edilebilir. Verilerin normalize edilmesi için Azure ML Studio'da **Normalize Data** modülü bulunmaktadır. Bu modül, nitelikleri normalleştirmek için beş farklı matematiksel teknik sunar ve bir veya daha fazla seçilen sütun üzerinde çalışabilir [6].

- **Zscore:** Bu seçenek, tüm değerleri bir z-puanına dönüştürür.
- **MinMax:** Bu seçenek, her bir özelliğin değerlerini doğrusal olarak [0,1] aralığına yeniden ölçeklendirir.
- **Lojistik:** Sütundaki değerler, bir logaritmik dönüşümü kullanılarak dönüştürülür.
- **LogNormal:** Bu seçenek, tüm değerleri lognormal ölçeğe dönüştürür.
- **Tanh:** Tüm değerler hiperbolik bir tanjanta dönüştürülür.

Bu çalışmada Min-Max normalizasyon yöntemi kullanılmaktadır. Bunun için deneye bir **Normalize Data** modülü eklenir ve Şekil 58'de görüldüğü gibi son eklenen Edit Metadata modülünün çıktı bağlantısı ile Normalize Data modülünün girdi bağlantısı bağlanır.

Normalize Data modülünün özellikleri (properties penceresinde) şu şekilde ayarlanır:

- Transform method: MinMax



Şekil 3.58: Çalışmaya Normalize Data modülünün eklenmesi

- Use 0 for constant columns: Seçili
- Columns to transform: boylam, enlem, nüfus, hane, medyan_gelir, Divide(toplam_oda_hane), Divide(toplam_yatak_odasi_hane), Divide(nüfus_hane), konut_medyan_yasi.

Sırasıyla SAVE ve RUN butonlarına basılarak deney kaydedilir ve çalıştırılır. Normalize Data modülünün Transformed dataset (sol) çıktı bağlantısı Visualize seçeneği seçilerek görselleştirildiğinde seçilen niteliklerin Şekil 59'daki gibi 0-1 aralığında değerler aldığı görülmektedir.

Konut Fiyatı Tahmini > Normalize Data > Transformed dataset

rows	columns								
20635	13								
		blok_id	boylam	enlem	konut_medyan_yasi	nüfus	hane	medyan_gelir	okyanus_yi
		1	0.211155	0.567481	0.770833	0.040923	0.048668	0.761575	NEAR BAY
		2	0.212151	0.565356	0.354167	0.404512	0.567111	0.759076	NEAR BAY
		3	0.210159	0.564293	1	0.071354	0.074795	0.649496	NEAR BAY
		4	0.209163	0.564293	1	0.082197	0.096311	0.480056	NEAR BAY
		5	0.209163	0.564293	1	0.083421	0.116803	0.29145	NEAR BAY
		6	0.209163	0.564293	1	0.056838	0.082992	0.311455	NEAR BAY
		7	0.209163	0.563231	1	0.175936	0.247439	0.271811	NEAR BAY
		8	0.209163	0.563231	1	0.186953	0.315574	0.215226	NEAR BAY
		9	0.208167	0.563231	0.791667	0.195523	0.288934	0.106108	NEAR BAY
		10	0.209163	0.563231	1	0.255859	0.349898	0.27518	NEAR BAY
		11	0.208167	0.564293	1	0.143757	0.190061	0.223948	NEAR BAY
		12	0.208167	0.564293	1	0.247439	0.167143	0.211173	NEAR BAY

Şekil 3.59: Normalize edilmiş verilerin görselleştirilmesi

3.4.12 Eğitim ve Test Veri Setinin Ayrılması

Veri hazır hale geldikten sonra bir blok grubunun medyan ev değerini tahmin etmek için bir regresyon makine öğrenmesi modeli eğitilmesi amaçlanmaktadır. Makine öğrenmesi uygulamalarında model veri setinin bir bölümüyle eğitilir ve ardından veri setinin farklı bir bölümünde test edilir veya değerlendirilir. Verilerin bir kısmını test için ayırarak, modelin yeni verileri gördüğü senaryo simüle edilir. Yani model, modelin parametrelendirilmesinde kullanılmayan verilere dayanarak tahminler yapmaktadır [13]. Bu çalışmada elde edilen veri seti modelin eğitilmesi için %70 eğitim veri seti ve eğitilen modelin performansının test edilmesi için %30 test veri seti olacak şekilde rassal bir biçimde ayrılmaktadır. Bunun için çalışmaya bir **Split Data** modülü eklenir ve Şekil 60'ta gösterildiği gibi girdi bağlantısı **Normalize Data** modülünün **Transformed dataset** (sol) çıktı bağlantısı ile bağlanır.



Şekil 3.60: Deneye Split Data modülünün eklenmesi

Split Data modülünün özellikleri şu şekilde ayarlanır:

- **Splitting mode:** Split Rows
- **Fraction of rows in the first output dataset:** 0.7
- **Randomized split:** Seçili
- **Random seed:** 1234
- **Stratified split:** False

Sırasıyla SAVE ve RUN butonlarına basılarak çalışma kaydedilir ve çalıştırılır. Split Data modülünün Results dataset1 (sol) çıktı bağlantısı Visualize seçeneği seçilerek görselleştirildiğinde veri setinin %70'i yani 14444 adet örneğin sol tarafa, Results dataset2 (sağ) çıktı bağlantısı görselleştirildiğinde veri setinin %30'unun yani 6191 adet örneğin sağ tarafa ayrıldığı Şekil 61'deki gibi görülmektedir.

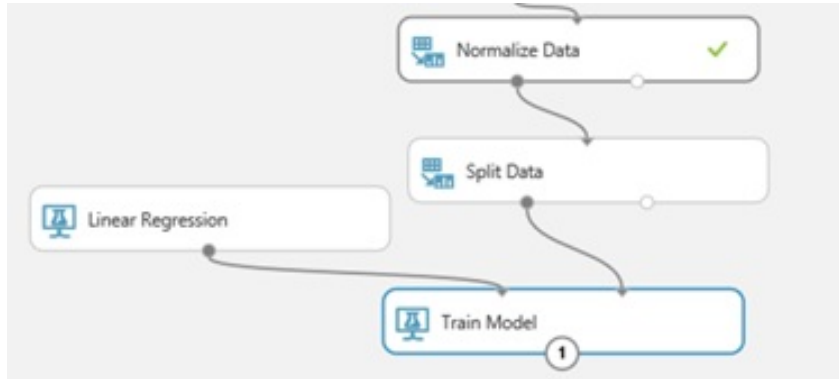
3.4.13 Doğrusal Regresyon Modelinin Eğitilmesi

Modeli eğitmek genellikle modeli oluşturan bir veya daha fazla işlevi parametreleştirmekten oluşur. Ardından, bu eğitilmiş model, bir veya daha fazla değerlendirme ölçütü kullanılarak değerlendirilebilir [13]. Konut fiyatı tahmini makine öğrenmesi çalışması için doğrusal regresyon modelinin eğitilmesi amacıyla çalışmaya bir **Train Model** modülü eklenir ve **Dataset** (sağ) girdi bağlantısı **Split Data** modülünün **Results dataset1** (sol) çıktı bağlantısı ile bağlanır. **Train Model** modülünün özelliklerinden (properties penceresinden) **Label** sütunu **Ln(medyan_ev_degeri)** olacak şekilde

Results dataset1 (sol)				Results dataset2 (sağ)			
rows	columns			rows	columns		
14444	13			6191	13		
		blok_id	boylam	enlem			
		17636	0.242032	0.503719			
		7544	0.611554	0.144527			
		13544	0.701195	0.171095			
		15248	0.706175	0.047821			
		7061	0.628486	0.146652			
		2010	0.451195	0.444208			
		18224	0.226096	0.517535			
		14236	0.729084	0.01594			
		5783	0.608566	0.171095			
		20232	0.505976	0.182784			
		3953	0.568725	0.178533			
		15669	0.190239	0.55898			
		20510	0.280876	0.643996			
		14265	0.722112	0.01594			
		8960	0.593625	0.156217			
		11423	0.634462	0.124336			
		8120	0.613546	0.1339			
		11035	0.650398	0.131775			

Şekil 3.61: Eğitim ve test veri setinin görselleştirilmesi

ayarlanır. Daha sonra çalışmaya bir **Linear Regression** modülü eklenir ve modülün çıktı bağlantısı **Train Model** modülünün **Untrained model** (sol) girdi bağlantısı ile Şekil 62'deki gibi bağlanır.



Şekil 3.62: Çalışmaya Train Model ve Linear Regression modüllerinin eklenmesi

Linear Regression modülünün özellikleri (properties penceresinden) şu şekilde ayarlanır:

- **Solution method:** Ordinary Least Squares
- **L2 regularization weight:** 0.1
- **Included intercept term:** Seçili
- **Random number seed:** 1234
- **Allow unknown categorical levels:** Seçili

Eğitilen modelin test edilmesi için çalışmaya bir **Score Model** modülü eklenir ve **Trained model** (sol) girdi bağlantısı **Train Model** modülünün çıktı bağlantısı ile bağlanır. Daha sonra **Score**

Model modülünün **Dataset** (sağ) girdi bağlantısı **Split Data** modülünün **Result dataset2** (sağ) çıktı bağlantısı ile bağlanır. Modüller arasındaki bağlantılar Şekil 63'te gösterilmektedir.



Şekil 3.63: Çalışmaya Score Model modülünün eklenmesi

Çalışma SAVE ve RUN butonlarına basılarak kaydedilip çalıştırılır. Çalışma bittiğinde, **Ln (medyan_ev_degeri)** niteliğini tahmin etmek için model eğitimi ayarlarını ve eğitilen modelde kullanılan niteliklerin nispi ağırlıklarını gösteren **Train Model** modülünün çıktı bağlantısı Visualize seçeneği ile görselleştirilir. Şekil 64 incelendiğinde, **okyanus_yakindigi** niteliğinin <1H OCEAN değerine sahip olma durumunun 12.9546 nispi ağırlık değer ile **medyan_ev_degerini** tahmin etmede en büyük etkiye sahip olduğu görülmektedir.

Feature	Weight
okyanus_yakindigi_<1H OCEAN_0	12.9546
okyanus_yakindigi_NEAR OCEAN_4	12.9072
okyanus_yakindigi_NEAR BAY_3	12.8916
okyanus_yakindigi_INLAND_1	12.6491
medyan_gelir	1.7759
boy_lam	-1.54836
enlem	-1.45254
Divide(nufus_hane)	-0.54678
hane	0.363155
Divide(toplam_yatak_odasi_hane)	0.341861
Divide(toplam_oda_hane)	-0.162952
konut_medyan_yasi	0.151874
nufus	-0.139294
okyanus_yakindigi_ISLAND_2	0
okyanus_yakindigi#unknown_6	0

Şekil 3.64: Modelde kullanılan niteliklerin nispi ağırlıkları

Score Model modülünün çıktı bağlantısı Visualize seçeneği ile görselleştirilir. Bu şekilde test veri setindeki **Scored Labels** sütunundaki (model tarafından tahmin edilen) değerler **Ln (medyan_ev_degeri)** sütunundaki (medyan ev değerlerinin gerçek log-normal değerleri) değerler ile Şekil 65'te gösterildiği gibi karşılaştırılabilir.

Konut Fiyatı Tahmini > Score Model > Scored dataset

rows: 6191, columns: 14

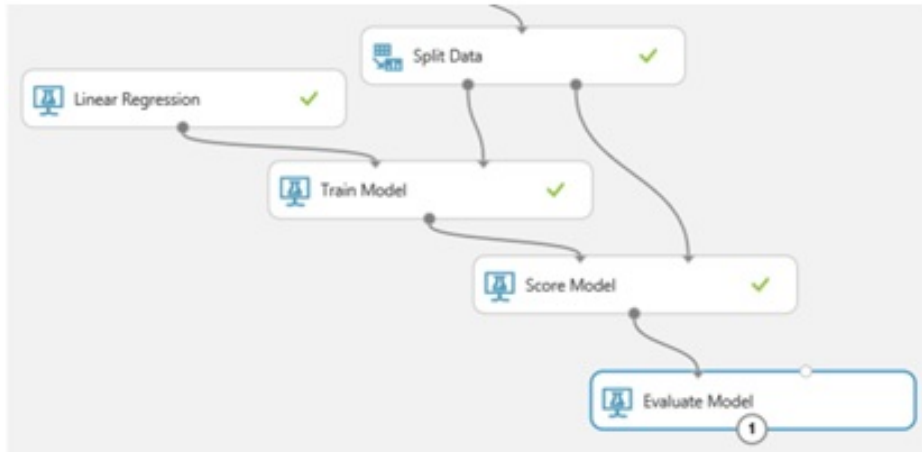
Gerçek ↓ **Tahmin** ↓

yanus_yakintligi	medyan_ev_degeri	Divide(toplam_oda_hane)	Divide(toplam_yatak_odasi_hane)	Divide(nufus_hane)	Ln(medyan_ev_degeri)	Scored Labels
JAR OCEAN	359100	0.377043	0.22487	0.182784	12.791356	12.497162
H OCEAN	390000	0.63412	0.110337	0.460544	12.873902	12.966654
JAR BAY	500001	0.343624	0.242908	0.038568	13.122365	12.678395
LAND	80200	0.285772	0.131652	0.373712	11.292279	11.571935
JAR OCEAN	98000	0.278911	0.110337	0.728973	11.492723	11.659513
H OCEAN	355100	0.308743	0.134483	0.214944	12.780155	12.435616
H OCEAN	264600	0.529574	0.160592	0.412172	12.485975	12.510656
JAR OCEAN	141800	0.123352	0.188664	0.641211	11.862173	11.822606
H OCEAN	318500	0.658829	0.109231	0.428617	12.671378	12.611748
H OCEAN	294700	0.411125	0.129027	0.108584	12.593713	12.766155
JAR BAY	450000	0	0.152785	0.020656	13.017003	12.132926
LAND	109700	0.879288	0.661842	0.344292	11.605505	11.431784
H OCEAN	111400	0.137612	0.192132	0.634919	11.620883	11.722432
JAR BAY	458300	0.29672	0.133857	0.185947	13.035279	12.262064
H OCEAN	500001	0.602562	0.143673	0.216312	13.122365	13.080989

Şekil 3.65: Model tarafından tahmin edilen değerlerin medyan ev değerlerinin gerçek log-normal değerleri ile karşılaştırılması

3.4.14 Doğrusal Regresyon Modelinin Değerlendirilmesi

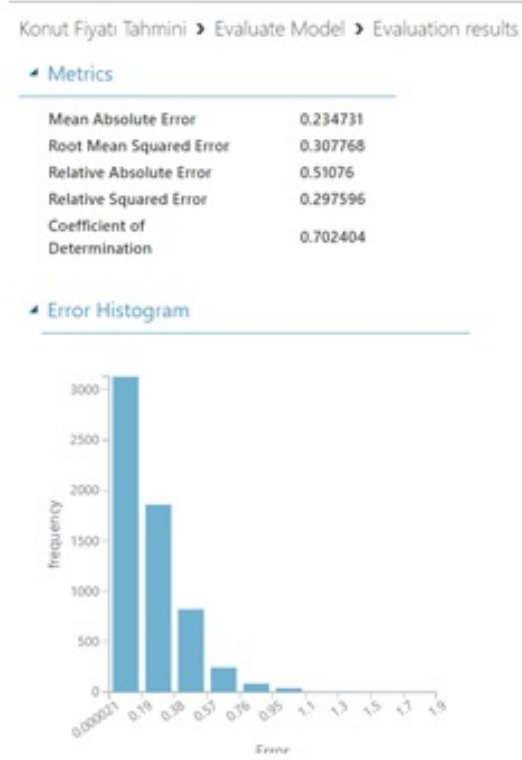
Konut fiyatı tahmini deneyi için eğitilen doğrusal regresyon modelinin değerlendirilmesi için çalışmaya bir **Evaluate Model** modülü eklenir ve Şekil 66'da görüldüğü gibi modülün **Scored dataset** (sol) girdi bağlantısı **Score Model** modülünün çıktı bağlantısına bağlanır.



Şekil 3.66: Konut fiyatı tahmini deneyine Evaluate Model modülünün eklenmesi

Konut fiyatı tahmini deneyi SAVE ve RUN butonlarına basılarak kaydedilip çalıştırılır. Evaluate Model modülünün çıktı bağlantısı Visualize seçeneği seçilerek görselleştirilir. Şekil 67'de görüldüğü

gibi Doğrusal Regresyon algoritması ile eğitilen modelin test veri seti üzerindeki performans ölçütü değerleri elde edilmektedir.



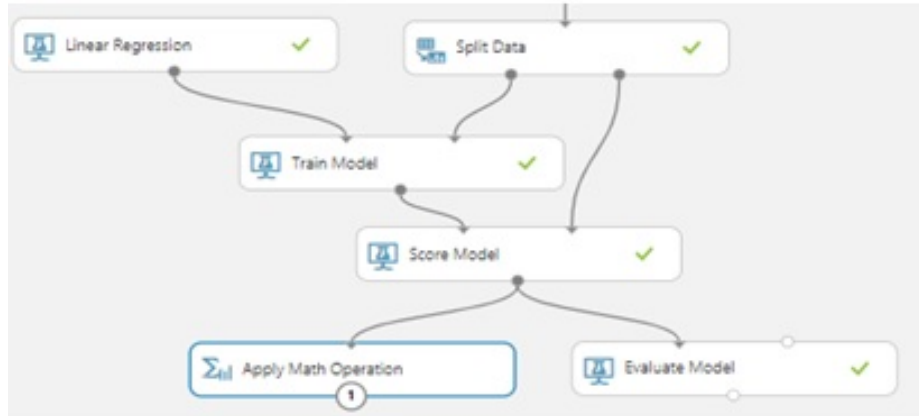
Şekil 3.67: Doğrusal Regresyon algoritması ile eğitilen modelin performans ölçütleri

Hata istatistiklerinin her biri için (belirleme katsayısı hariç), daha küçük olan daha iyidir; daha küçük bir değer, tahminlerin gerçek değerlere daha yakın olduğunu gösterir. Performans ölçütleri incelendiğinde, ortalama mutlak hata (**Mean Absolute Error**), kök ortalama kare hata (**Root Mean Squared Error**), göreceli mutlak hata (**Relative Absolute Error**) ve nispi kare hata (**Relative Squared Error**) değerlerinin küçük olduğu görülmektedir. R kare değeri olarak da bilinen, bir modelin verilere ne kadar iyi uyduğunu gösteren istatistiksel bir ölçüm olan belirleme katsayısı (**Coefficient of Determination**) 0.70 değeri ile çok da iyi olmasa da ortalamanın üzerinde görünmektedir. 0 belirleme katsayısı, modelin etiketi açıklamadığını gösterirken, 1.0 değeri etiket ve puan arasındaki çok gerçekçi olmayan mükemmel uyumu gösterir.

Genel olarak bu ölçümler iyidir fakat model tatmin edici değildir. Diğer taraftan, hedef niteliğin medyan ev değeri niteliği için bir doğal logaritma ölçeğinde olduğu unutulmamalıdır. Doğrusal Regresyon modelinin test verileri için ürettiği tahmin değerlerini orijinal medyan_ev_degeri değerleri ile karşılaştırabilmek için çalışmaya bir **Apply Math Operation** modülü eklenir ve girdi bağlantısı Şekil 68’de görüldüğü gibi **Score Model** modülünün çıktı bağlantısı ile bağlanır.

Scored Labels niteliğine üstel dönüşüm fonksiyonu uygulamak için **Apply Math Operation** modülünün özellikleri aşağıdaki gibi ayarlanır:

- **Category:** Basic
- **Basic math function:** Exp
- **Column set:** Scored Labels



Şekil 3.68: Çalışmaya Apply Math Operation modülünün eklenmesi

- **Output mode:** Append

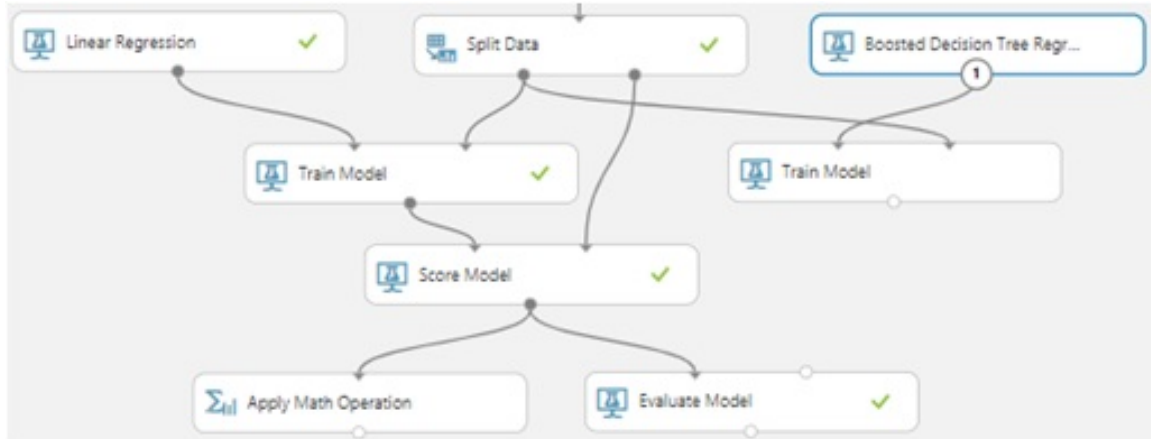
Deneyin kaydedilmesi ve çalıştırılması için sırasıyla SAVE ve RUN butonlarına basılır. Apply Math Operation modülünün çıktı bağlantısı Visualize seçeneği ile görselleştirildiğinde Şekil 69'da görüldüğü gibi Scored Labels niteliğine üstel dönüşüm fonksiyonunun uygulanmış hali olan **Exp(Scored Labels)** sütunundaki değerler **medyan_ev_degeri** sütunundaki değerler ile karşılaştırılabilir.

okyanus_yakinaligi	medyan_ev_degeri	Divide(toplam_oda_hane)	Divide(toplam_yatak_odasi_hane)	Divide(nufus_hane)	Ln(medyan_ev_degeri)	Scored Labels	Exp(Scored Labels)
NEAR OCEAN	359100	0.377043	0.22487	0.182784	12.791356	12.497162	267576.697055
<1H OCEAN	390000	0.63412	0.110337	0.460544	12.873902	12.966654	427904.021515
NEAR BAY	500001	0.343624	0.242908	0.038568	13.122365	12.678395	320742.978792
INLAND	80200	0.285772	0.131652	0.373712	11.292279	11.571935	106078.543762
NEAR OCEAN	98000	0.278911	0.110337	0.728973	11.492723	11.659513	115787.608909
<1H OCEAN	355100	0.308743	0.134483	0.214944	12.780155	12.435616	251605.130241
<1H OCEAN	264600	0.529574	0.160592	0.412172	12.485975	12.510656	271212.096253
NEAR OCEAN	141800	0.123352	0.188664	0.641211	11.862173	11.822606	136298.970144
<1H OCEAN	318500	0.658829	0.109231	0.428617	12.671378	12.611748	300062.974862
<1H OCEAN	294700	0.411125	0.129027	0.108584	12.593713	12.766155	350163.444886
NEAR BAY	450000	0	0.152785	0.020656	13.017003	12.132926	185892.941295
INLAND	109700	0.879288	0.661842	0.344292	11.605505	11.431784	92206.329681

Şekil 3.69: medyan_ev_degeri niteliği değerlerinin Exp(Scored Labels) niteliği değerleri ile karşılaştırılması

3.4.15 Güçlendirilmiş Karar Ağacı Regresyon Modelinin Eğitilmesi

Çalışmaya ikinci bir **Train Model** modülü eklenir ve **Dataset** (sağ) girdi bağlantısı **Split Data** modülünün **Results dataset1** (sol) çıktı bağlantı noktasına bağlanır. **Train Model** modülünün özelliklerinden (properties penceresinden) **Label** sütunu **Ln(medyan_ev_degeri)** olacak şekilde ayarlanır. Daha sonra çalışmaya bir **Boosted Decision Tree Regression** modülü eklenir ve modülün çıktı bağlantısı yeni **Train Model** modülünün **Untrained model** (sol) girdi bağlantısı ile Şekil 70'te gösterildiği gibi bağlanır.



Şekil 3.70: Çalışmaya Train Model ve Boosted Decision Tree Regression modüllerinin eklenmesi

Boosted Decision Tree Regression modülünün özellikleri (properties penceresinden) şu şekilde ayarlanır:

- Create trainer mode: Single Parameter
- Maximum number of leaves per tree: 20
- Minimum number of training instances required to form a leaf: 10
- Learning rate: 0.2
- Total number of trees constructed: 100
- Allow unknown categories: Checked

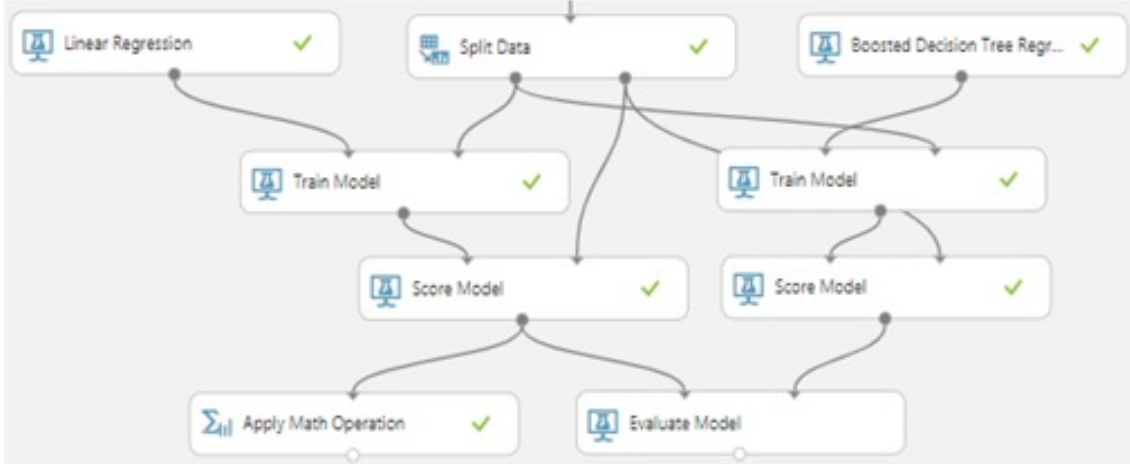
Eğitilen modelin test edilmesi için çalışmaya ikinci bir **Score Model** modülü eklenir ve **Trained model** (sol) girdi bağlantısı Boosted Decision Tree Regression modülü için eklenen **Train Model** modülünün çıktı bağlantısı ile bağlanır. Daha sonra **Score Model** modülünün **Dataset** (sağ) girdi bağlantısı **Split Data** modülünün **Result dataset2** (sağ) çıktı bağlantısı ile bağlanır. Modüller arasındaki bağlantılar Şekil 71'de gösterilmektedir.



Şekil 3.71: Çalışmaya ikinci bir Score Model modülünün eklenmesi

3.4.16 Güçlendirilmiş Karar Ağacı Regresyon Modelinin Değerlendirilmesi

Konut fiyatı tahmini deneyi için Boosted Decision Tree Regression algoritması ile eğitilen modelin performans ölçütlerinin değerlendirilmesi için ikinci eklenen **Score Model** modülünün çıktı bağlantısı Şekil 72'de gösterildiği gibi **Evaluate Model** modülünün **Scored dataset to compare** (sağ) girdi bağlantısı ile bağlanır.



Şekil 3.72: Evaluate Model modülünün bağlantılarının ayarlanması

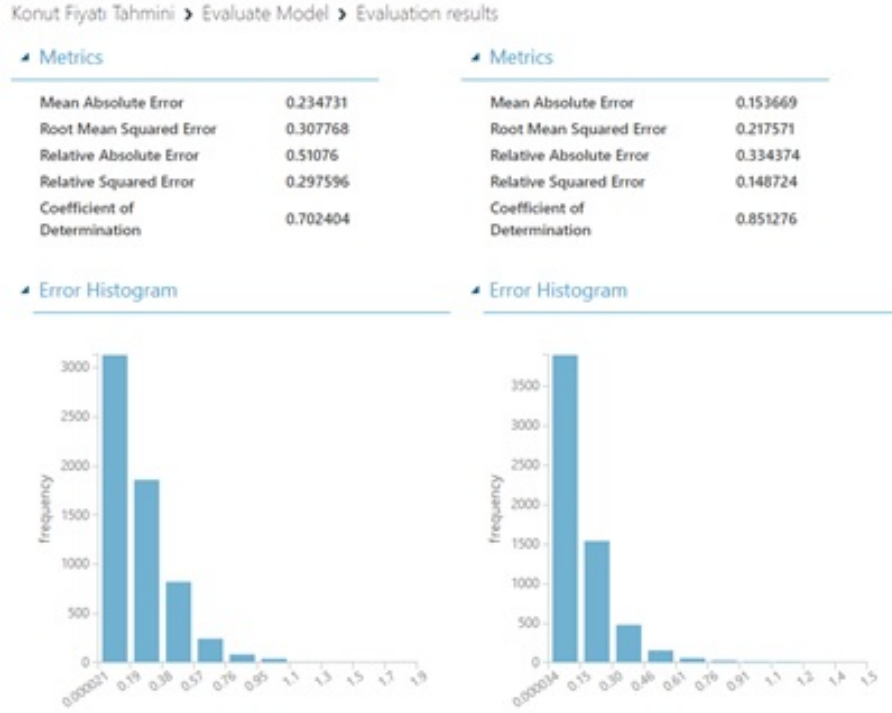
Deney SAVE ve Run butonlarına basılarak kaydedilip çalıştırılır. Evaluate Model modülünün çıktı bağlantısı Visualize seçeneği ile görselleştirilir.

Şekil 73'te görüldüğü gibi sol tarafta Evaluate Model modülünün sol girdi bağlantısı ile bağlanılan Doğrusal Regresyon modelinin performans ölçütleri, sağ tarafta Evaluate Model modülünün sağ girdi bağlantısı ile bağlanılan Güçlendirilmiş Karar Ağacı Regresyon modelinin performans ölçütleri bulunmaktadır. Performans ölçütleri incelendiğinde, Güçlendirilmiş Karar Ağacı Regresyon modeline ait ortalama mutlak hata (**Mean Absolute Error**), kök ortalama kare hata (**Root Mean Squared Error**), göreceli mutlak hata (**Relative Absolute Error**) ve nispi kare hata (**Relative Squared Error**) değerlerinin Doğrusal Regresyon modelinin ilgili performans ölçütlerinden daha küçük olduğu görülmektedir. Ayrıca Güçlendirilmiş Karar Ağacı Regresyon modelinin belirleme katsayısı (coefficient of determination) 0.85 değeriyle Doğrusal Regresyon modelininkinden oldukça daha yüksektir. Bu nedenle, konut fiyatı tahmininde Güçlendirilmiş Karar Ağacı modeli daha iyi performans göstermiştir.

Güçlendirilmiş Karar Ağacı Regresyon modelinin test verileri için ürettiği tahmin değerlerini orijinal medyan_ev_degeri değerleri ile karşılaştırmak için çalışmaya bir **Apply Math Operation** modülü eklenir ve girdi bağlantısı Şekil 74'te gösterildiği gibi ikinci **Score Model** modülünün çıktı bağlantısı ile bağlanır.

Güçlendirilmiş Karar Ağacı Regresyon modeli ile elde edilen Scored Labels niteliğine üstel dönüşüm fonksiyonu uygulamak için Apply Math Operation modülünün özellikleri properties penceresinden aşağıdaki gibi ayarlanır:

- Category: Basic
- Basic math function: Exp
- Column set: Scored Labels
- Output mode: Append



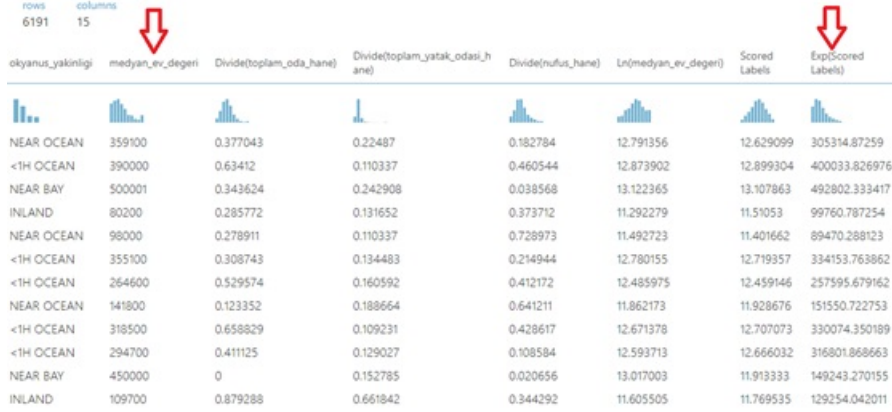
Şekil 3.73: Evaluate Model modülünün görselleştirilmesi

rows: 6191, columns: 15

okyanus_yakinligi	medyan_ev_degeri	Divide(toplam_oda_hane)	Divide(toplam_yatak_odasi_hane)	Divide(nufus_hane)	Ln(medyan_ev_degeri)	Scored Labels	Exp(Scored Labels)
NEAR OCEAN	359100	0.377043	0.22487	0.182784	12.791356	12.497162	267576.697055
<1H OCEAN	390000	0.63412	0.110337	0.460544	12.873902	12.966654	427904.021515
NEAR BAY	500001	0.343624	0.242908	0.038568	13.122365	12.678395	320742.978792
INLAND	80200	0.285772	0.131652	0.373712	11.292279	11.571935	106078.543762
NEAR OCEAN	98000	0.278911	0.110337	0.728973	11.492723	11.659513	115787.608909
<1H OCEAN	355100	0.308743	0.134483	0.214944	12.780155	12.435616	251605.130241
<1H OCEAN	264600	0.529574	0.160592	0.412172	12.485975	12.510656	271212.096253
NEAR OCEAN	141800	0.123352	0.188664	0.641211	11.862173	11.822606	136298.970144
<1H OCEAN	318500	0.658829	0.109231	0.428617	12.671378	12.611748	300062.974862
<1H OCEAN	294700	0.411125	0.129027	0.108584	12.593713	12.766155	350163.444886
NEAR BAY	450000	0	0.152785	0.020656	13.017003	12.132926	185892.941295
INLAND	109700	0.879288	0.661842	0.344292	11.605505	11.431784	92206.329681

Şekil 3.74: Deneye Apply Math Operation modülünün eklenmesi

Deney SAVE ve RUN butonlarına basılarak kaydedilir ve çalıştırılır. **Apply Math Operation** modülünün çıktı bağlantısı görselleştirilir. Böylece Şekil 75'te görüldüğü gibi **medyan_ev_degeri** sütunundaki değerler **Exp(Scored Labels)** sütunundaki değerler ile karşılaştırılabilir.



Şekil 3.75: medyan_ev_degeri niteliği değerlerinin Exp(Scored Labels) niteliği değerleri ile karşılaştırılması

3.4.17 Modelin Bir Web Hizmeti Olarak Dağıtılması

Eğitilmiş ve test edilmiş bir model bir web hizmeti olarak dağıtılarak gerçek kullanıma sunulabilir [16]. Güçlendirilmiş Karar Ağacı Regresyon modeli Doğrusal Regresyon modelinden daha iyi performans gösterdiği için çalışmaya Güçlendirilmiş Karar Ağacı Regresyon modeli ile devam edilir. Bu yüzden, öncelikle Linear Regression modülü, Linear Regression modülü için ilk eklenen Train Model, Score Model ve Apply Math Operation modülleri ve artık ihtiyaç duyulmadığı için Evaluate Model modülü Şekil 76'da gösterildiği gibi silinir.

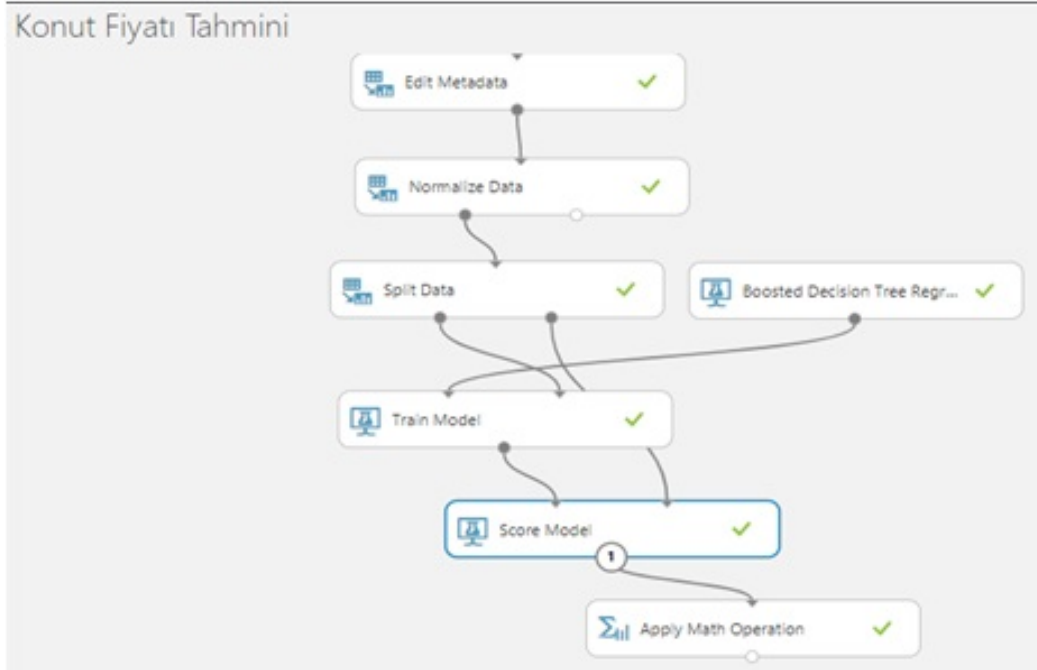
Web hizmetini oluşturmak için deney sırasıyla SAVE ve RUN butonu ile kaydedilip çalıştırılır, ardından Şekil 77'de görüldüğü gibi Setup Web Service (Predictive Web Service recommended) butonu tıklanır.

Sonuç, Şekil 78'de gösterildiği gibi tahmine dayalı deney adı verilen yeni bir deneydir; bu, web hizmeti girdisi (tahmin etmek istenilen girdileri alan) ve web hizmeti çıktısı (tahmin edilen sonuçları döndüren) arasında eğitilmiş modele karşı tahminin gerçekleştirilmesi için aramayı kapsar [19].

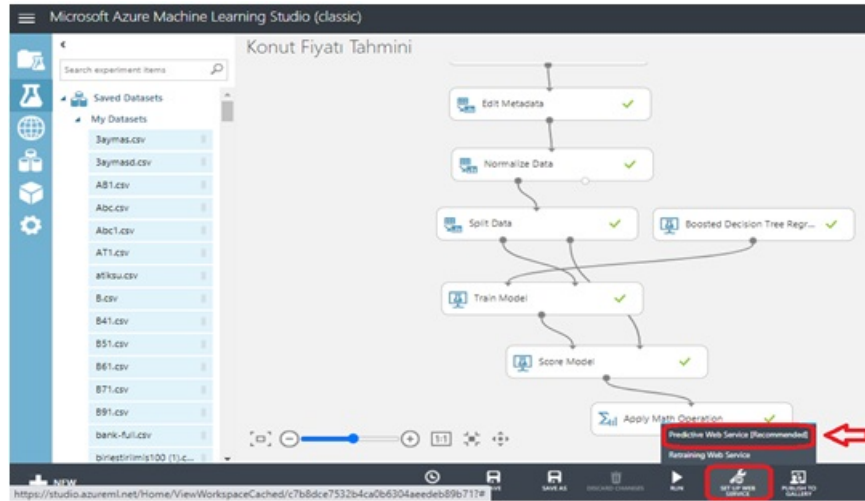
Deney RUN butonuna basılarak tekrar çalıştırılır ve ardından Şekil 79'da görüldüğü gibi sayfanın altındaki Deploy Web Service butonuna basılır.

Web hizmetinin dağıtılmasıyla yeni bir sayfa gösterilecektir. Bu sayfada, hem tek işleme hem de verilerin toplu olarak işlenmesi için API ayrıntıları görülebilir. Şekil 80'de görüldüğü gibi, başkalarının ML modelini kullanmasına izin vermek için paylaşılacak kırmızı ok işareti ile gösterilen bir API anahtarı vardır. Ayrıca, API URL'leri için API'yi kullanmanın iki temel yolu vardır: tek bir giriş kaydı için tahmin elde edilebilen bir istek ve yanıt yaklaşımı (request/response) ve bir grup kaydı içeren bir veri kaynağı (ör. Blob Depolamadaki bir CSV dosyasının yolu veya SQL Veritabanı veya Hive'daki tablolar) için tahminlerin elde edilebileceği toplu yürütme (batch execution) [19].

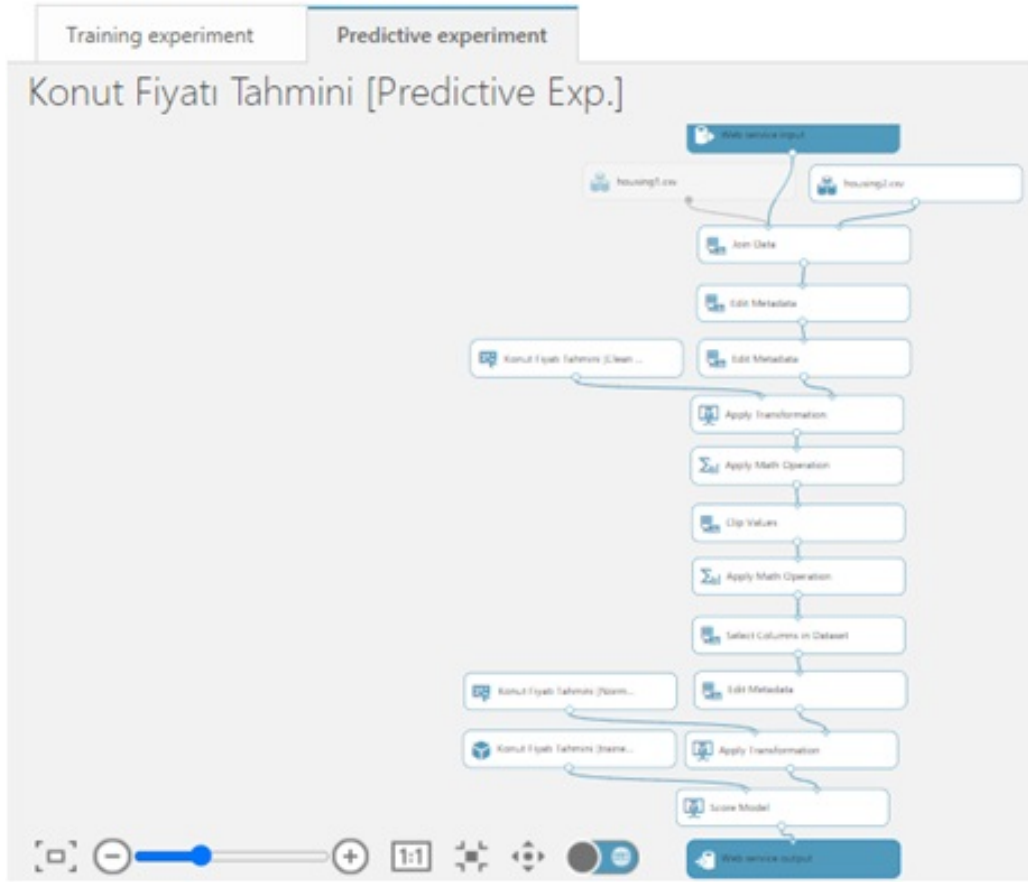
Batch Execution seçeneği tıklandığında API bilgilerini içeren yeni sayfa açılır. Şekil 81' de görülebileceği gibi, talep bilgisi için web hizmeti URL'si gösterilmektedir.



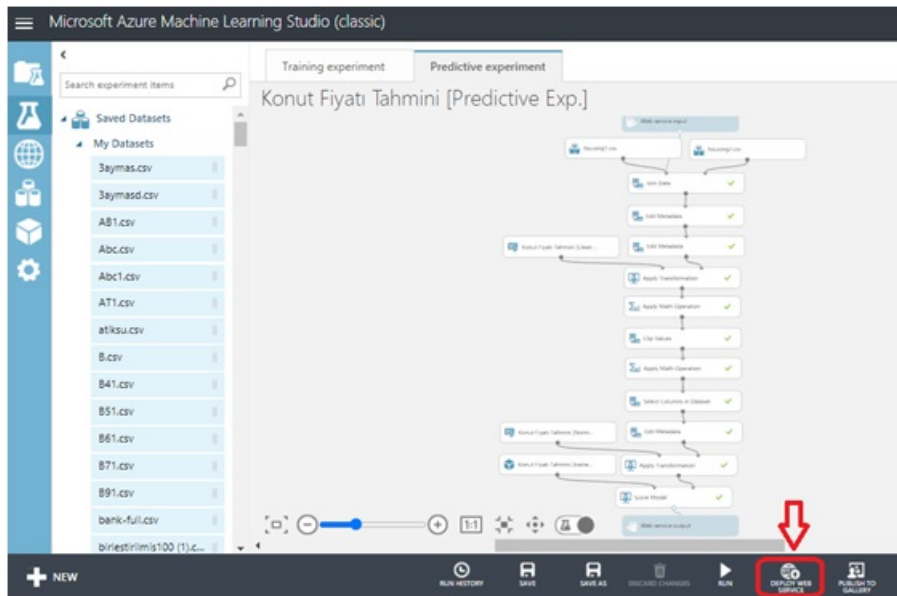
Şekil 3.76: Doğrusal Regresyon modeli ile ilgili modüllerin silinmesi



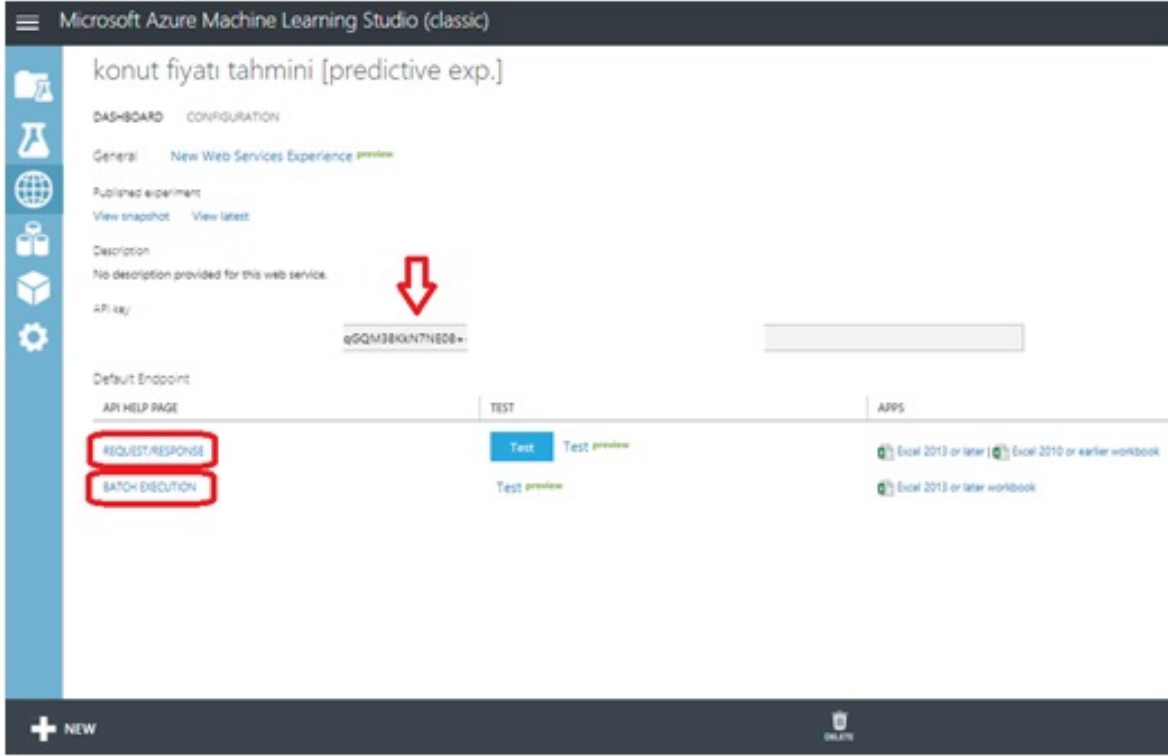
Şekil 3.77: Tahminsel deneyin oluşturulması



Şekil 3.78: Web hizmeti girişi ve web hizmeti çıktı modüllerini gösteren tahmine dayalı bir deney



Şekil 3.79: Web hizmetinin dağıtılması



Şekil 3.80: Bir Web hizmeti oluşturmak

Oluşturulan modeli C#, Python ve R gibi herhangi bir dilden yararlanarak web uygulamaları, raporlar ve benzeri diğer uygulamalara entegre etmek mümkündür. Web hizmeti bilgi sayfasının sonunda Şekil 82'de görülebileceği gibi, C#, Python ve R'da kullanıcıların kodu uygulamalara yerleştirmesini sağlayan örnek kodlar bulunmaktadır.

Azure ML Studio'da tahminsel deneye geri dönülür. Son kullanıcıların doğru bilgileri sağlaması ve ilgili tahmini alabilmesi için web hizmetinin giriş ve çıkışının değiştirilmesi gerekmektedir.

1. blok_id ve median_house_value değerlerini içeren housing2.csv veri seti silinir.
2. Join Data modülü silinir. housing1.csv verisetinin çıktı bağlantı noktası ile ilk Edit Metadata modülünün girdi bağlantı noktası bağlanır. Benzer şekilde Web service input modülünün çıktı bağlantı noktası ile Apply Transformation modülünün Dataset (sağ) girdi bağlantı noktası bağlanır. Modüller arasındaki bağlantılar Şekil 83'te gösterilmektedir.
3. Niteliklerin isimlerini değiştirmeyi sağlayan ikinci Edit Metadata modülünün özellikler (properties) penceresinden Launch Column Selector bölümünde Şekil 84'te gösterildiği gibi median_house_value niteliği çarpı (X) işaretine basılarak çıkartılır. Daha sonra onay (✓) işaretine basılır.

Özellikler penceresinde New Column names kutusundaki isimler silinir ve hedef niteliğinin ismini çıkarmak için virgülle ayrılmış şekilde aşağıdaki sütun isimleri yazılır:

blok_id, boylam, enlem, konut_medyan_yasi, toplam_oda, toplam_yatak_odasi, nüfus, hane, medyan_gelir, okyanus_yakınlığı

4. İkinci Edit Metadata modülü seçili iken sırasıyla SAVE ve Run Selected seçeneklerine basılır. Clip Values modülünün özellikler penceresinde Launch Columns Selector bölümünde Şekil

Batch Execution API Documentation for Konut Fiyatı Tahmini [Predictive Exp.]

Updated: 10/13/2020 16:56

No description provided for this web service.

- [Previous version of this API](#)
- [Submit a Job](#)
- [Start a Job](#)
- [Get Job Status](#)
- [Delete a Job](#)
- [Web App Template for BES](#)
- [BES SDK](#)
- [Sample Code](#)
- [API Swagger Document](#) ⓘ
- [Endpoint Management Swagger Document](#) ⓘ

Submit (but not start) a Batch Execution job

Request

Method	Request URI	Request Body	Response Body	HTTP Version
POST	https://ussouthcentral.services.azureml.net/bstapi-version=2.0		services/1ac0b4	764/30 HTTP/1.1

Request Headers

Request Header	Description
Authorization:Bearer plc123	Required. Pass the API Key here. Obtain this key from the publisher of the API.
Content-Length	Required. The length of the content body.

Şekil 3.81: Web hizmeti bilgi sayfası

Sample Code

C# Python R Select sample code

```

# How this works:
#
# 1. ASSUMPTION: This code assumes that your input is already uploaded to your Azure storage account (if the web service accepts input).
# 2. Call BES to process the data in the blob.
# 3. The results get written to another Azure blob
# 4. Download the output blob to a local file

library("curl")
library("httr")
library("rjson")

requestFailed = function(response) {
  return (response$status_code >= 400)
}

printHttpError = function(response, result) {
  print(paste("The request failed with status code:", response$status_code, sep=" "))

  # Print the headers - they include the request ID and the timestamp, which are useful for debugging the failure
  print(response$headers)
  print(fromJSON(result))
}

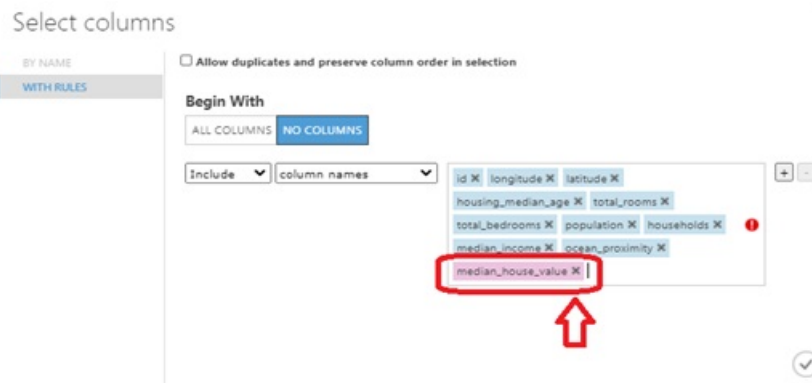
saveBlobToFile = function(blobUrl, resultsLabel) {
  output_file = "myresults.csv" # Replace this with the location you would like to use for your output file

```

Şekil 3.82: Web hizmeti bilgi sayfası



Şekil 3.83: Modüller arasındaki bağlantıların ayarlanması



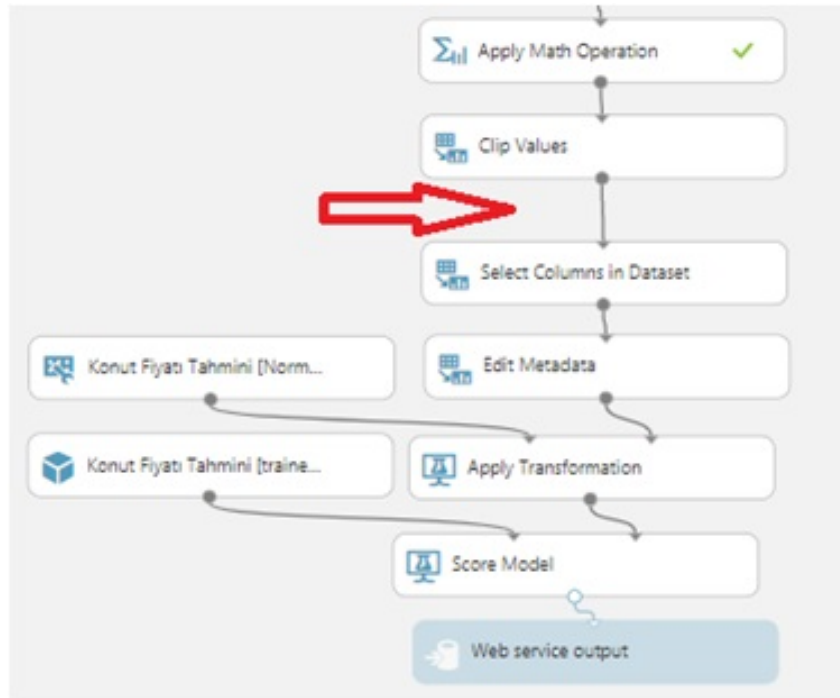
Şekil 3.84: İkinci Edit Metadata modülünde hedef niteliğin çıkartılması

85'te gösterildiği gibi median_house_value niteliği çarpı (X) işaretine basılarak çıkartılır. Daha sonra onay (✓) işaretine basılır.



Şekil 3.85: Clip Values modülünde hedef niteliğin çıkartılması

5. medyan_ev_degeri niteliğine doğal logaritma dönüşümü uygulamak için eklenen Apply Math Operation modülü silinir ve Clip Values modülünün çıktı bağlantı noktası ile Select Columns in Dataset modülünün girdi bağlantı noktası Şekil 86'da gösterildiği gibi bağlanır.



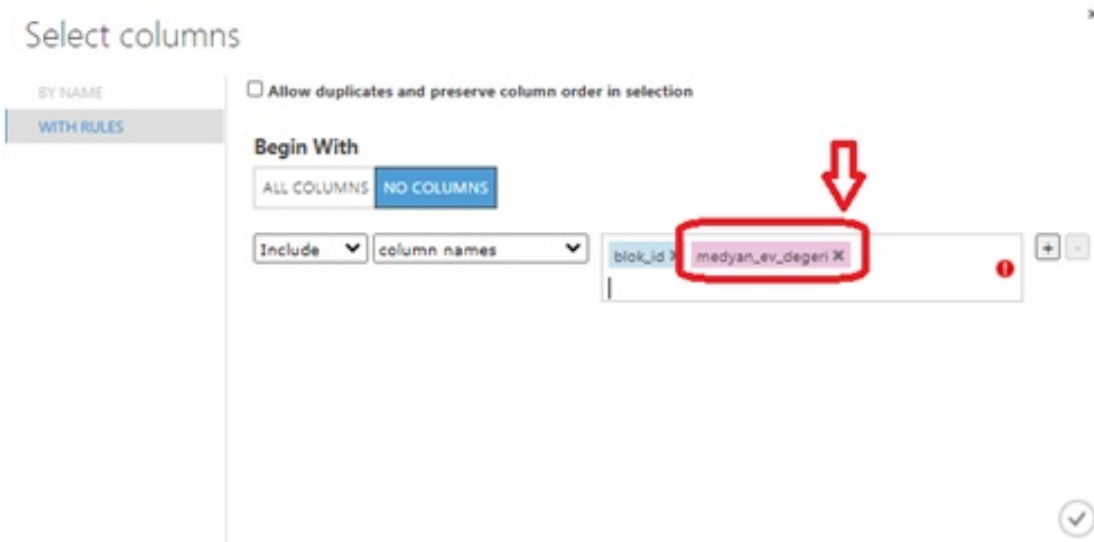
Şekil 3.86: Apply Math Operation modülünün silinmesi

6. Select Columns in Dataset modülünün özellikler penceresinden Select Columns bölümünde Şekil 87’de gösterildiği gibi medyan_ev_degeri ve Ln(medyan_ev_degeri) nitelikleri (X) işaretine basılarak çıkartılır. Daha sonra onay (✓) işaretine basılır.



Şekil 3.87: medyan_ev_degeri ve Ln(medyan_ev_degeri) niteliklerinin çıkartılması

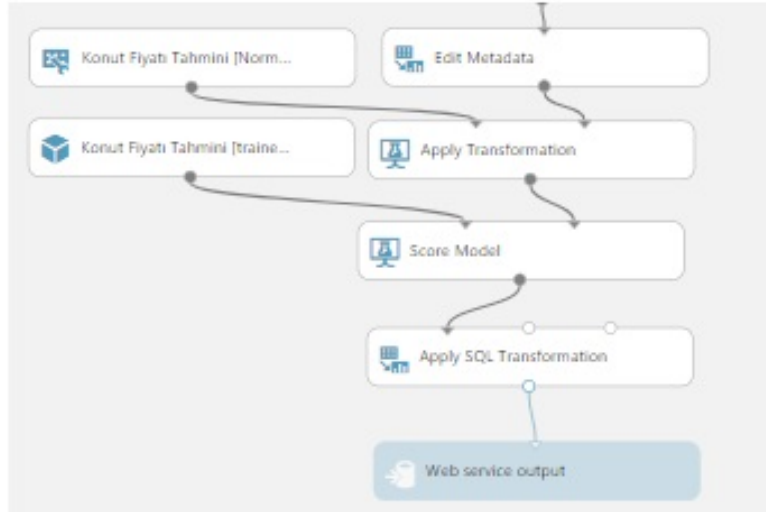
7. Üçüncü eklenen Edit Metadata modülünün özellikler penceresinde Launch Columns Selector bölümünde Şekil 88’de gösterildiği gibi medyan_ev_degeri niteliği çarpı (X) işaretine basılarak çıkartılır. Daha sonra onay (✓) işaretine basılır.



Şekil 3.88: Üçüncü Edit Metadata modülünden hedef niteliğin çıkartılması

8. Score Model modülü Ln(medyan_ev_degeri) değerlerini tahmin etmektedir. Modelin medyan_ev değeri niteliğinin tahmin değerini vermesi istendiği için Score Model modülü ile

Web service output modülü arasındaki bağlantı silinir. Ln(medyan_ev_degeri) niteliği tahminine yani Scored Labels niteliğine üstel dönüşüm uygulamak için deneye bir Apply SQL Transformation modülü eklenir. Score Model modülünün çıktı bağlantı noktası ile Apply SQL Transformation modülünün Table1 (sol) girdi bağlantı noktası Şekil 89'da gösterildiği gibi bağlanır. Apply SQL Transformation modülünün çıktı bağlantı noktası ile Web service output modülünün girdi bağlantı noktası bağlanır.



Şekil 3.89: Deneye Apply SQL Transformation modülünün eklenmesi

Apply SQL Transformation modülünün özellikler penceresinde Şekil 90'da görüldüğü gibi SQL query script bölmesine aşağıdaki kod yazılır:

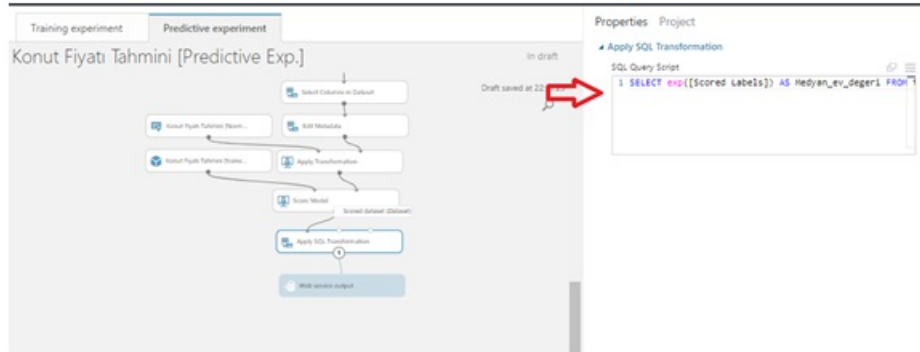
```
SELECT exp([Scored Labels]) AS Medyan_ev_degeri FROM t1;
```

okyanus_yakinaligi	medyan_ev_degeri	Divide(toplam_oda_hane)	Divide(toplam_yatak_odasi_hane)	Divide(nufus_hane)	Ln(medyan_ev_degeri)	Scored Labels	Exp(Scored Labels)
NEAR OCEAN	359100	0.377043	0.22487	0.182784	12.791356	12.497162	267576.697055
<1H OCEAN	390000	0.63412	0.110337	0.460544	12.873902	12.966654	427904.021515
NEAR BAY	500001	0.343624	0.242908	0.038568	13.122365	12.678395	320742.978792
INLAND	80200	0.285772	0.131652	0.373712	11.292279	11.571935	106078.543762
NEAR OCEAN	98000	0.278911	0.110337	0.728973	11.492723	11.659513	115787.608909
<1H OCEAN	355100	0.308743	0.134483	0.214944	12.780155	12.435616	251605.130241
<1H OCEAN	264600	0.529574	0.160592	0.412172	12.485975	12.510656	271212.096253
NEAR OCEAN	141800	0.123352	0.188664	0.641211	11.862173	11.822606	136298.970144
<1H OCEAN	318500	0.658829	0.109231	0.428617	12.671378	12.611748	300062.974862
<1H OCEAN	294700	0.411125	0.129027	0.108584	12.593713	12.766155	350163.444886
NEAR BAY	450000	0	0.152785	0.020656	13.017003	12.132926	185892.941295
INLAND	109700	0.879288	0.661842	0.344292	11.605505	11.431784	92206.329681

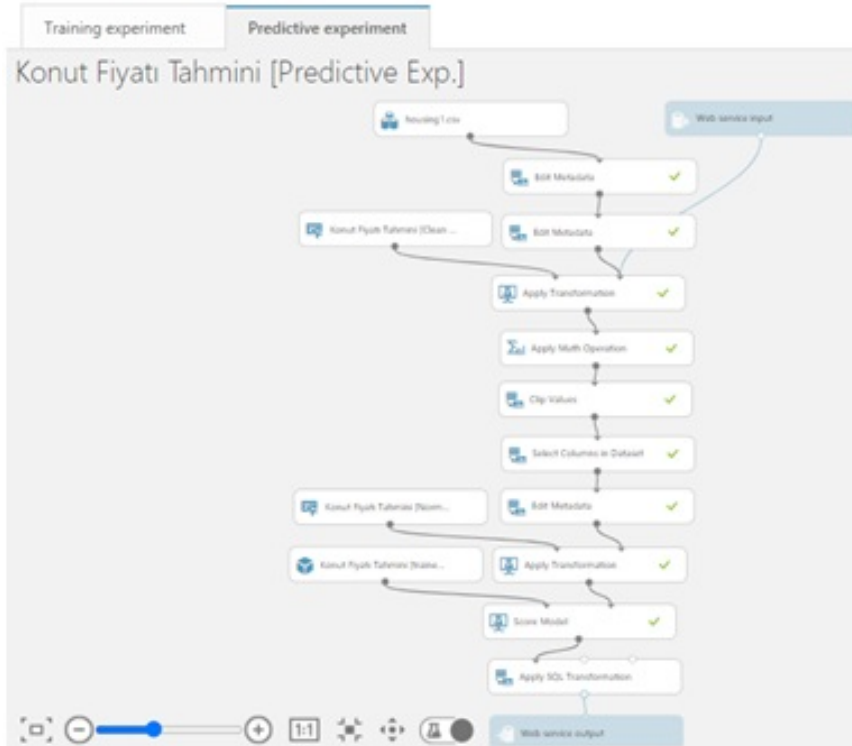
Şekil 3.90: medyan_ev_degeri niteliği değerlerinin Exp(Scored Labels) niteliği değerleri ile karşılaştırılması

Deneyin son hali Şekil 91'deki gibi görünmektedir.

Tahminsel deney RUN butonuyla tekrar çalıştırılır ve çalışması tamamlandığında DEPLOY WEB SERVICE butonuna basılarak web hizmeti düzenlenmiş biçimiyle dağıtılır. Ekranı gelen Web



Şekil 3.91: Apply SQL Transformation modülünün özelliklerinde sorgu kodunun yazılması



Şekil 3.92: Deneyin son ayarlamalardan sonra görüntüsü

hizmetinin üzerine yazılmasıyla ilgili mesaj onaylanır. Deneyi web hizmeti olarak yeniden dağıttıktan sonra, Şekil 92'deki gibi web hizmeti sayfasında Test butonuna basılarak API test edilebilir.



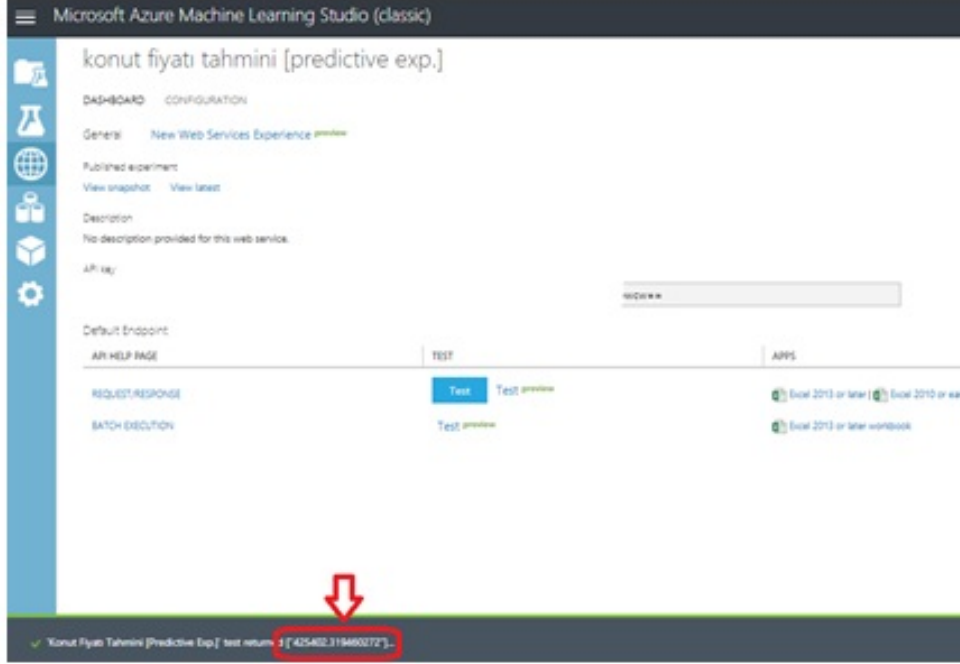
Şekil 3.93: Web hizmeti bilgi sayfasında Test butonu

Şekil 93'te görüldüğü gibi test sayfasında housing1.csv verisetindeki niteliklerin girilmesi beklenmektedir. Edit Metadata modülü ile kategorik olarak belirlenen nitelik (OKYANUS_YAKINLIGI) için liste biçiminde açılan kutu eklendiği görülmektedir.

The image shows the 'Test Konut Fiyatı Tahmini [Predictive Exp.] Service' form. The title is 'Enter data to predict'. The form contains five input fields with the following values: BLOK_ID (52), BOYLAM (122.23), ENLEM (37.88), KONUT_MEDYAN_YASI (41), and TOPLAM_ODA (880). A vertical scrollbar is visible on the right side of the form. A checkmark icon is located at the bottom right of the form.

Şekil 3.94: API'nin test edilmesi

Test için BLOK_ID: 52, BOYLAM: -122.23, ENLEM: 37.88, KONUT_MEDYAN_YASI: 41, TOPLAM_ODA: 880, TOPLAM_YATAK_ODASI: 129, NUFUS: 322, HANE: 126, MEDYAN_GELIR: 8.3252, OKYANUS_YAKINLIGI: NEAR BAY değerleri girilip onay (✓) işaretine basıldığında Şekil 94'te görüldüğü gibi uygulama girilen örneğin medyan ev değerini '425402.319460272' olarak tahmin etmektedir.



Şekil 3.95: API'nin test amacıyla girilen örnek için tahmin sonucu

Azure ML Studio benzer şekilde API'nin Excel'de test edilmesine de olanak sağlamaktadır. Bunun için Şekil 95'te görüldüğü gibi web hizmeti bilgi sayfasında 'Excel 2013 or later' butonuna basılır. Açılan 'Örnek Verileri İndirme' mesajı onaylandıktan sonra indirilen Excel dosyası açılır.

'Use sample data' butonuna basıldığında Şekil 96'da görüldüğü gibi Excel dosyasına örnek veriler gelmektedir.

Şekil 97'de gösterildiği gibi Input alanı altında, giriş veri aralığı seçilir ve Output alanının altına çıktı hücre numarası yazılır. Daha sonra örnek verilerin test edilmesi için 'Predict' butonuna basılır. API'nin örnek veriler için tahmin sonuçları seçilen Output hücresine uygulama tarafından yazılır.

konut fiyatı tahmini [predictive exp.]

DASHBOARD CONFIGURATION

General New Web Services Experience preview

Published experiment

View snapshot View latest

Description

No description provided for this web service.

API key

REVIEW

Default Endpoint

API HELP PAGE

TEST

APPS

REQUEST/RESPONSE

Test

Test preview

Excel 2013 or later

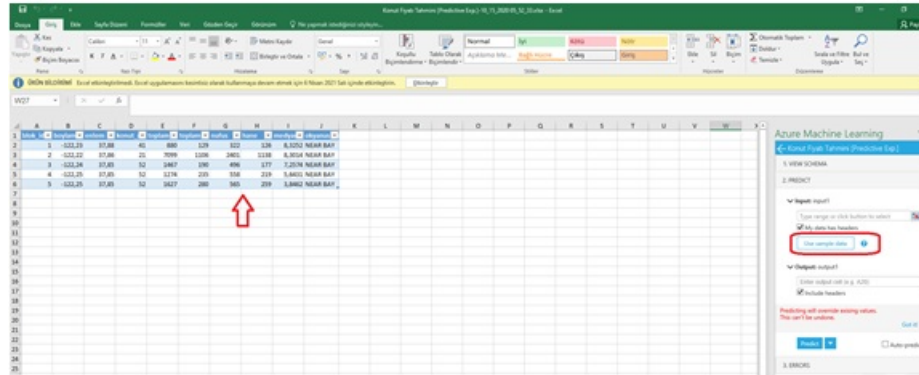
Excel 2010 or earlier

BATCH EXECUTION

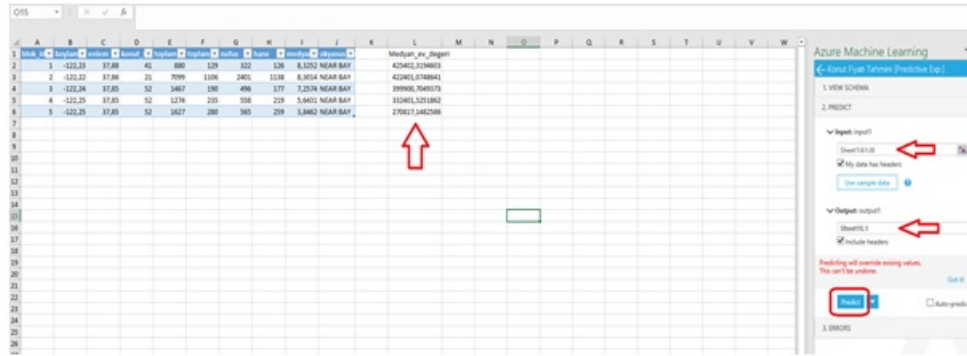
Test preview

Excel 2013 or later workbook

Şekil 3.96: API'nin Excel'de test edilmesi



Şekil 3.97: API'nin Excel'de örnek veriler ile test edilmesi



Şekil 3.98: API'nin Excel'de örnek veriler için tahmin sonuçları

Referanslar

- [1] Lantz, B., (2015), *Machine Learning with R: Discover how to build machine learning algorithms, prepare data, and dig deep into data prediction techniques with R*, second ed. Packt Publishing.
- [2] Ghatak, A., (2017), *Machine Learning with R*, Springer Nature Singapore Pte Ltd.
- [3] Sarkar, D., Bali, R, Sharma, T. (2018), *Practical Machine Learning with Python: A Problem-Solver's Guide to Building Real-World Intelligent Systems*, Apress Media.
- [4] Wali A., (2014), *Clojure for Machine Learning*, Packt Publishing.
- [5] Kamath, U., Choppella, K., (2017), *Mastering Java Machine Learning: A Java developer's guide to implementing machine learning and big data architectures*, Packt Publishing.
- [6] Barga, R., Fontama, V., Tok, W.H., (2015), *Predictive Analytics with Microsoft Azure Machine Learning: Build and Deploy Actionable Solutions in Minutes*, 2. Baskı, Apress Media.
- [7] Antonik, P., (2018), *Application of FPGA to Real-Time Machine Learning: Hardware Reservoir Computers and Software Image Processing*, Springer.
- [8] Tsihrintzis, G.A., Sotiropoulos, D.N., Jain, L.C., (2019), *Machine Learning Paradigms: Advances in Data Analytics*, Springer International Publishing.
- [9] Lorenzo, A.J., Rickard, M., Braga, L.H., Guo, Y., Oliveria, J.P., (2018), *Predictive Analytics and Modeling Employing Machine Learning Technology: The Next Step in Data Sharing, Analysis, and Individualized Counseling Explored With a Large, Prospective Prenatal Hydronephrosis Database*, *Pediatric Urology*:123, s. 204-209.
- [10] Barnes, J., (2015), *Azure Machine Learning: Microsoft Azure Essentials*, Microsoft Press.
- [11] Salvaris, M., Dean, D., Tok, W.H., (2018), *Deep Learning with Azure: Building and Deploying Artificial Intelligence Solutions on the Microsoft AI Platform*, Apress.
- [12] Bhatia, A., Chiu, Y., (2017), *Machine Learning with R Cookbook: Analyze data and build predictive models*, second ed., Packt Publishing.
- [13] Whitenack, D, (2017), *Machine Learning With Go: Implement Regression, Classification, Clustering, Time-series Models, Neural Networks and more using the Go Programming Language*, Packt Publishing.
- [14] Yang, F., Wang, D., Xu, F., Huang, Z., Tsui, K.L., (2020), *Lifespan prediction of lithium-ion batteries based on various extracted features and gradient boosting regression tree model*, *Journal of Power Sources*:476.
- [15] Wei, Z., Meng, Y., Zhang, W., Peng, J., Meng, L., (2019), *Downscaling SMAP soil moisture estimation with gradient boosting decision tree regression over the Tibetan Plateau*, *Remote Sensing of Environment*: 225, s. 30–44.
- [16] Pathak, N., Bhandari, A., (2018), *IoT, AI, and Blockchain for .NET: Building a Next-Generation Application from the Ground Up*, Apress.
- [17] Familiar, B., Barnes, J., (2017), *Business in Real-Time Using Azure IoT and Cortana Intelligence Suite: Driving Your Digital Transformation*, Apress.
- [18] Etaati, L., (2019), *Machine Learning with Microsoft Technologies: Selecting the Right Architecture and Tools for Your Project*, Apress Media.
- [19] Tejada, Z., (2017), *Mastering Azure Analytics: Architecting in the Cloud with Azure Data Lake, HDInsight, and Spark*, Apress.
- [20] <https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference>
- [21] Elston, S.F., (2015), *Data Science in the Cloud with Microsoft Azure Machine Learning and R*, O'Reilly Media.

- [22] <https://bit.ly/2GywU1B>
- [23] <https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/evaluate-model>
- [24] <https://www.kaggle.com/camnugent/california-housing-prices>
- [25] Brownlee, J., (2016), Machine Learning Mastery With Python Understand Your Data, Create Accurate Models and Work Projects End-To-End, Machine Learning Mastery.
- [26] MICROSOFT DAT228X DEVELOPING BIG DATA SOLUTIONS WITH AZURE MACHINE LEARNING, DAT228x-Lab2-Training Predictive Models
- [27] Haroon, D., (2017), Python Machine Learning Case Studies: Five Case Studies for the Data Scientist, Apress Media.
- [28] Reese, R.M., Reese, J.L., Kaluza, B., Kamath, U., Choppella, K., (2017), Machine Learning: End-to-End guide for Java developers, Packt Publishing.
- [29] Bali, R., Sarkar, D., Lantz, B. Lesmeister, C., (2016), R: Unleash Machine Learning Techniques, Packt Publishing.
- [30] Bisong, E., (2019), Building Machine Learning and Deep Learning Models on Google Cloud Platform, Apress Media.
- [31] Nicolas, P.R., (2017), Scala for Machine Learning, Packt Publishing.

Yazarlar Hakkında



Orhan Torkul, Lisans derecesini, Sakarya Üniversitesi Sakarya Mühendislik Fakültesi Endüstri Mühendisliği bölümünden 1982 yılında, Yüksek Lisans derecesini Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı'ndan 1987 yılında, doktora derecesini İngiltere Cranfield Teknoloji Enstitüsü'nden 1993 yılında almıştır. Sakarya Üniversitesi Mühendislik Fakültesi Endüstri Mühendisliği bölümünde 1993-1998 yılları arasında Yardımcı Doçent, 1998-2003 yılları arasında Doçent, 2003 yılından itibaren ise Profesör olarak görev yapmıştır. 1993-1995 yılları arasında Mühendislik Fakültesi Dekan Yardımcılığı, 1997-2011 yılları arasında Enformatik Bölüm Başkanlığı, 2005-2011 yılları arasında Uzaktan Eğitim Araştırma ve Uygulama Merkezi Müdürlüğü, 2011-2014 tarihleri arasında Endüstri Mühendisliği Bölüm Başkanı ve 2014-2018 yılları arasında ise Mühendislik Fakültesi Dekanı olarak görev yapmıştır. 2018-2020 yılları arasında Yalova Üniversitesi'nde Rektör Yardımcılığı görevi yapmıştır. İmalat Planlama ve Kontrol, Yönetim Bilişim Sistemleri, Uzaktan Eğitim, Yapay Zeka, Dijital Dönüşüm ve Endüstri 4.0 alanlarında araştırmalar ve çalışmalar yapmaktadır. Çok sayıda uluslararası hakemli dergide makale, ulusal ve uluslararası konferanslarda bildirileri ve kitap bölümleri bulunmaktadır. Birçok ulusal ve uluslararası konferans ve sempozyum düzenleme komitelerinde yer almış ve davetli konuşmacı olarak çok sayıda konferans, bilimsel toplantı ve çalıştaylara katılmıştır. Prof. Dr. Orhan Torkul, Yönetim Bilişim Sistemleri, Bilişim Sistemleri Güvenlik ve Kontrolü ve İleri İmalat Planlama ve Kontrol Sistemleri konularında çeşitli dersler vermektedir.



Merve Şişci, lisans eğitimini 2013 yılında Sakarya Üniversitesi Endüstri Mühendisliği'nde tamamlamıştır. 2014 yılında Dumlupınar Üniversitesi Endüstri Mühendisliği'nde Araştırma Görevlisi olarak göreve başlamış ve 2017 yılında Dumlupınar Üniversitesi Endüstri Mühendisliği Anabilim dalında "Yapay Sinir Ağları İle Hisse Senedi Kapanış Fiyatlarının Tahmini ve Portföy Optimizasyonu" tez çalışması ile yüksek lisans derecesini almıştır. 2017 yılında Sakarya Üniversitesi Endüstri Mühendisliği Anabilim dalında doktora eğitimine başlamıştır. 2017 yılından itibaren Sakarya Üniversitesi Endüstri Mühendisliği Bölümü'nde geçici görevlendirme ile Araştırma Görevlisi olarak görevine devam etmektedir. Yapay Sinir Ağları, Derin Öğrenme, Makine Öğrenmesi, Kaba Kümeleme, Üretim Planlama ve Kontrol ve Montaj Hatları alanlarında araştırmalar yapmaktadır.



python powered

```
print("Hello, world!")
```

4. Python ile Görüntü İşleme

Python ile Görüntü İşleme ve Örnek Tespit Uygulamaları

Ayça TOPRAK¹, Gültekin ÇAĞIL¹

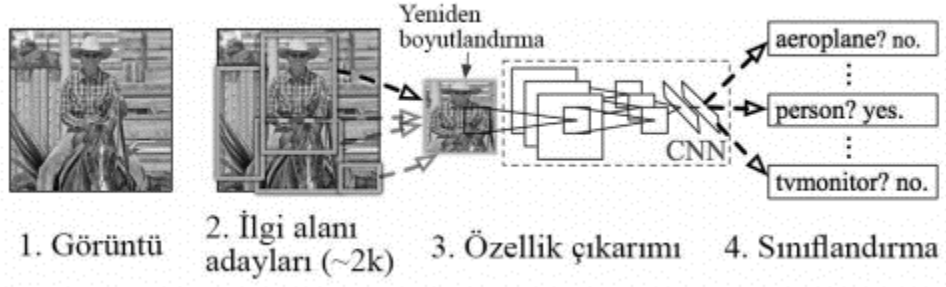
¹Sakarya Üniversitesi Üniversitesi, Endüstri Mühendisliği, Serdivan, Sakarya, Türkiye

4.1 Giriş

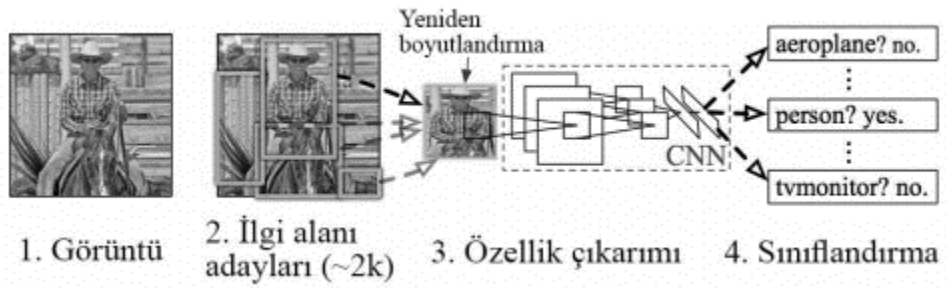
Görüntü işleme, fotoğraf ve video gibi dijital görüntülerin içerisinde kullanıcıya belirli bilgileri aktarabilmek amacıyla kullanılmaktadır [1]. Dijital görüntü en başında nesne tespit etme ve nesne tanıma çalışmaları yer almaktadır. Bu alanda çeşitli algoritmalar geliştirilmiştir. 2001 yılında Paul Viola ve Michael Jones tarafından geliştirilen “Viola Jones” isimli algoritma nesne tespiti için oluşturulmuş ayrıca etkin bir şekilde çalışan ilk algoritmadır. Literatür incelendiğinde nesnelere tespit edip tanıma için oluşturulmuş kütüphanelere, Single Shot Multi Box Detector (SSD), Region Based Convolutional Networks (R-CNN), Fast R-CNN ve Mask R-CNN örnek olarak gösterilebilir. Facebook 2014 yılında fotoğraflardan yararlanarak kullanıcılarını otomatik olarak etiketlemek için 120 milyon parametreyi R-CNN’e katarak, yüz tanıma işlemini başarıyla gerçekleştiren DeepFace adlı derin öğrenme teknolojisini kullanmıştır. Fast R-CNN, R-CNN modelinin bir türevidir ve R-CNN modelinin yavaşlığının giderilmesi amacıyla geliştirilmiştir. R-CNN oldukça yavaş çalışan bir modeldir ve gerçek zamanlı nesne tanıma uygulamalarında kullanılması bu sebeple mümkün değildir [2].

Aşağıda verilen Şekil 4.1 ve Şekil 4.2 isimli görsellerde R-CNN ve Fast R-CNN kütüphanelerinin çalışma mimarileri özetlenmiştir.

Görüntü işleme; Biyomedikal ürünler, Uzay bilimleri, Hasarlı görüntülerin tamir edilmesi, Gece görüşü, insansız hava araçları, Ürün kalite denetimi ve sınıflandırma, Parmak izi, göz ve plaka okuma, Haritacılık ve meteoroloji sistemleri gibi çok çeşitli uygulama alanlarında kullanılmaktadır. Literatürde yapılmış çalışmalar incelendiğinde nesne tespit etme ve tanıma uygulamaları; Veri girişi, ön işleme, öz nitelik çıkarma ve tanımlama aşamaları ortak olarak kullanılarak oluşturulduğu



Şekil 4.1: R-CNN çalışma mimarisi



Şekil 4.2: Fast R-CNN çalışma mimarisi

görülmektedir. Bu aşamalar aşağıda başlıklar halinde detaylıca açıklanmıştır.

4.1.1 Veri Girişi

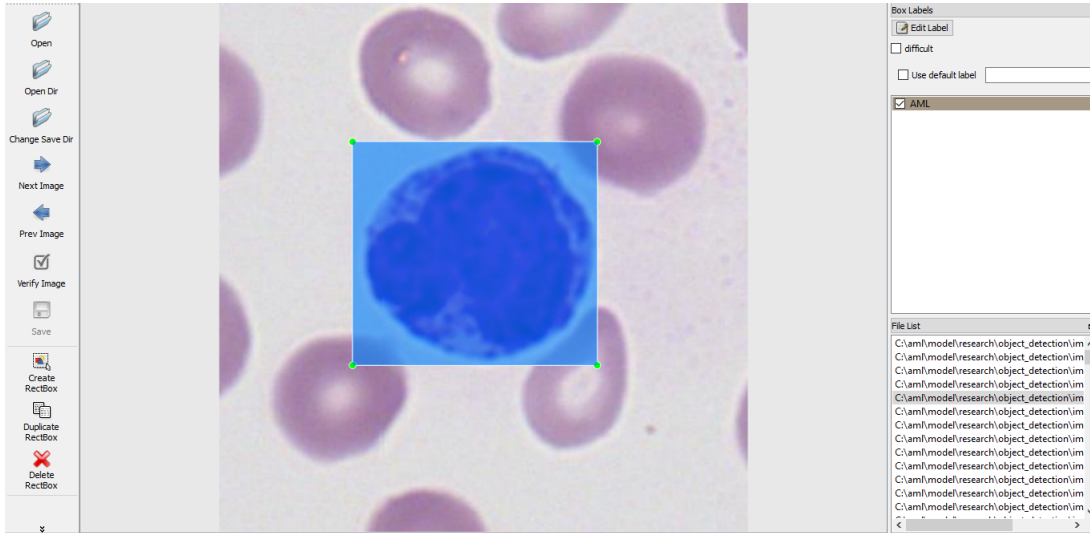
İlk aşamada hazırlanan görüntüler hedeflenen bilgileri elde etmek için sisteme girdi olarak eklenmektedir. Girdilerin formatı kullanılan programlama dili, platform gibi çeşitli faktörlerden etkilenebilmektedir. Bu etkilenmeye karşılık, tüm platformlarda geçerli ortak yaklaşım, görüntü piksellerinin matris haline getirme işlemidir.

4.1.2 Ön İşleme

Veri ön işleme aşaması, her çalışma için farklılık göstermektedir. Bu durumun sebeplerine, veri setinin içerisinde parazitli, eksik ve tutarsız görüntüler bulunması örnek gösterilebilir. Her veri setinde aynı hatalar bulunmadığından ön işleme adımı her veri setine aynı uygulamayı yapma düşüncesi yanlış olacaktır. Hatalı görüntüleri düzenlemek için; regresyon, eşikleme, filtreleme, karar ağaçları gibi çeşitli uygulamalar bulunmaktadır. Tek yöntem uygulamanın yeterli olmadığı durumlarda farklı yöntemleri bir arada kullanmak mümkündür.

4.1.3 Öz Nitelik Çıkartma

Ön işleme yapılmış veri setinin içerisinden, uygulamaya başlarken elde edilmesi planlanan bilginin çıkarılma aşamasıdır. Yani görüntüler üzerinden, istenilen bölümün özelliklerinin bilgisayara anlatılma uygulaması olarak açıklanabilir. Öz nitelik çıkarmak için; SIFT (Scale-Invariant Feature Transform), SURF (Speeded-Up Robust Features) gibi uygulamalar kullanılmaktadır.



Şekil 4.3: Görüntülerden öz nitelik çıkarma aşaması örneği

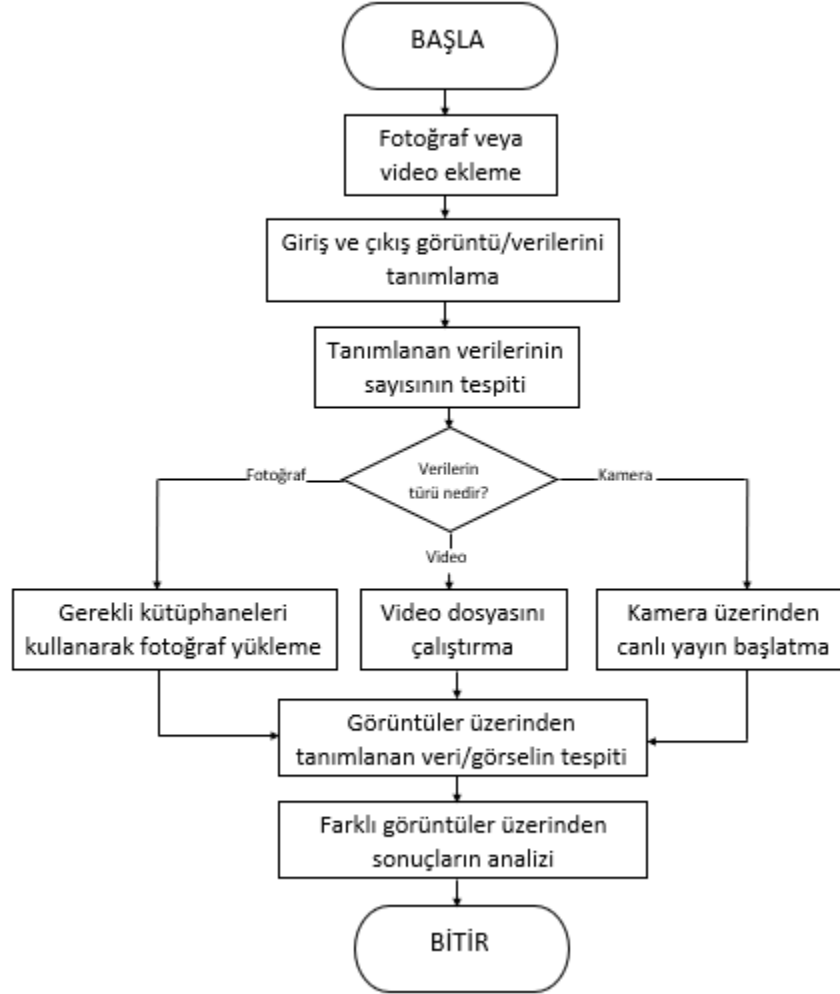
4.1.4 Tanımlama

Veri setinin içerisinde bulunan görüntülerden/verilerden çıkarım yapma aşaması olarak açıklanabilir. Bu adımda, temel olarak iki farklı yaklaşım bulunmaktadır [3].

Sınıflandırma: Önceden belirlenmiş sınıflara veri yerleştirme uygulaması olarak açıklanabilir.

Verilerin etiketleri vardır. Veri seti, eğitim ve test olmak üzere iki sınıfa ayrılarak işlemler gerçekleştirilmelidir.

Kümeleme: Benzer özelliklere sahip verileri aynı grupta toplama uygulaması olarak açıklanabilir. Verilerde etiket bulunmaz. Veriler gruba, benzerlik/alaka düzeyine göre eklenmektedir. Eğitim setine veya etiketleme işlemine ihtiyaç duyulmamaktadır.

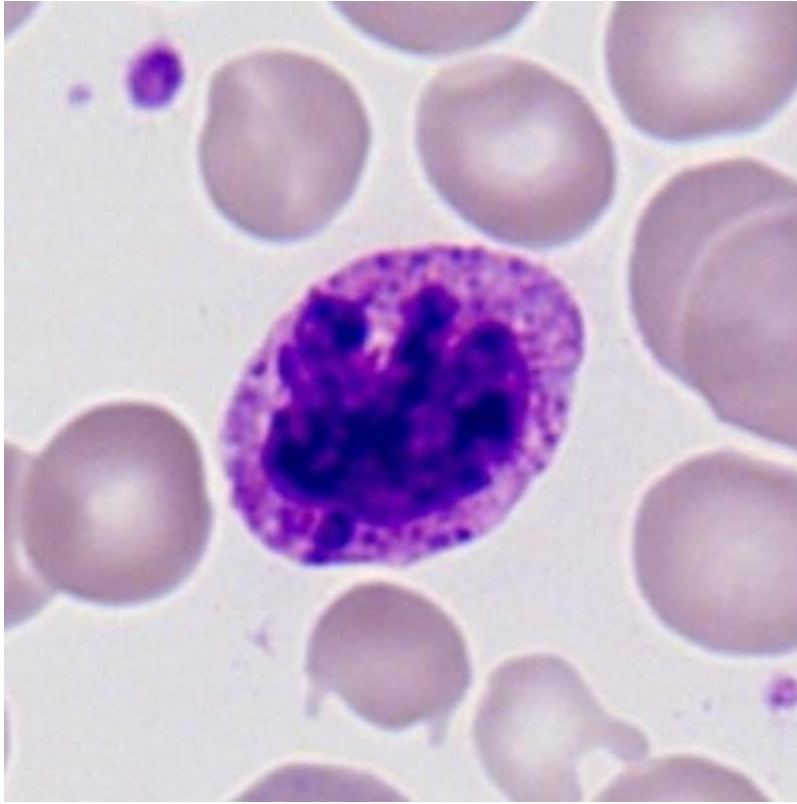


Şekil 4.4: Nesne tanıma uygulamalarının akış diyagramı

4.2 Uygulama 1

Cancer Imaging Archive [4] veritabanından yararlanılarak, tasarlanan çalışmaya uygun lösemi hastalığını içeren ve jpg uzantılı görseller toplanmıştır. Toplama işleminin ardından görüntüler kontrol edilmiş ve hepsi aynı boyutlara getirilmiştir. Hatalı görüntüler ayıklanmış ve sistemden uzaklaştırılmıştır. Şekil 6 isimli mikroskopik görüntü, kullanılan veri setinin içerisinde bir örnek olarak eklenmiştir.

Python, kolayca anlaşılabilen ve uygulama yapılabilen bir programlama dilidir. Bünyesinde



Şekil 4.5: Toplanmış olan lösemili hücre görüntülerinden bir örnek

bulunan açık kaynak kütüphaneleri sayesinde, kullanıcıya yapmak istediği işlemler için anlaşılabilir ve kolaylığı sağlamaktadır[5]. Bu avantajların göz önüne alınmış olması sebebiyle uygulama oluşturulurken, Python programlama dilinin “3.5.2” sürümü tercih edilmiştir. Bu tercihin yapılmasının sebebi ise kullanılan kütüphanelerin, belirtilen sürümde, çok daha kararlı bir şekilde çalışmasıdır.

Python yazılımının içerisine sırasıyla; pillow, lxml, Cython, contextlib2, jupyter, matplotlib, pandas, opencv-python ve tensorflow kütüphaneleri eklenmiştir. Görüntü işleme çalışmasının yapılabilmesi için yüklenen kütüphanelerin sisteme eklenmesi gerekmektedir. Bu kütüphaneler sayesinde yapılmak istenilen işlemler için ayrı ayrı kod satırına ihtiyaç duyulmaz. Kütüphanenin çağırılması yeterlidir. Tensorflow kütüphanesi, GPU üzerinden de işlem yapabilme gücüne sahiptir. Ancak uygulama oluşturulurken ortaya çıkan problemlerden dolayı CPU üzerinden kurulum yapılmıştır. Şekil 4.6 isimli görsel, Komut İstemi penceresinden indirilmiş olan kütüphaneleri göstermektedir.

```
C:\>pip install pillow
pip install lxml
pip install Cython
pip install contextlib2
pip install jupyter
pip install matplotlib
pip install pandas
pip install opencv-python
```

Şekil 4.6: Python Yazılımına Eklenen Kütüphaneler

GitHub uygulaması kullanılarak, Tensorflow Object Detection API[6] isimli çalışmanın içerisinde bulunan model[7] dosyası indirilmiş ve gerekli kurulum aşamaları Komut isteminde (CMD) gerçekleştirilmiştir.

Eğitim yapılırken Faster-RCNN-Inception-V2[8] modeli tercih edilmiştir. Proje için hazırlanmış olan dosyaların içerisinde bulunan “Object Detection” klasörüne tercih edilmiş olan model eklenmiş ve gerekli düzenlemeler yapılmıştır.

Dosyalanan ve lösemi içeren, mikroskobik görüntülerin (685 adet) %20’si test klasörünün, geriye kalan %80’i ise train klasörünün içerisine taşınmıştır. Şekil 4.12 adı verilmiş olan görsel, oluşturulan test ve train isimli klasörleri içermektedir. Test klasörünün içinde 137, train klasörünün içinde ise 548 adet lösemi taşıyan hücre görüntüsü örneği bulunmaktadır.

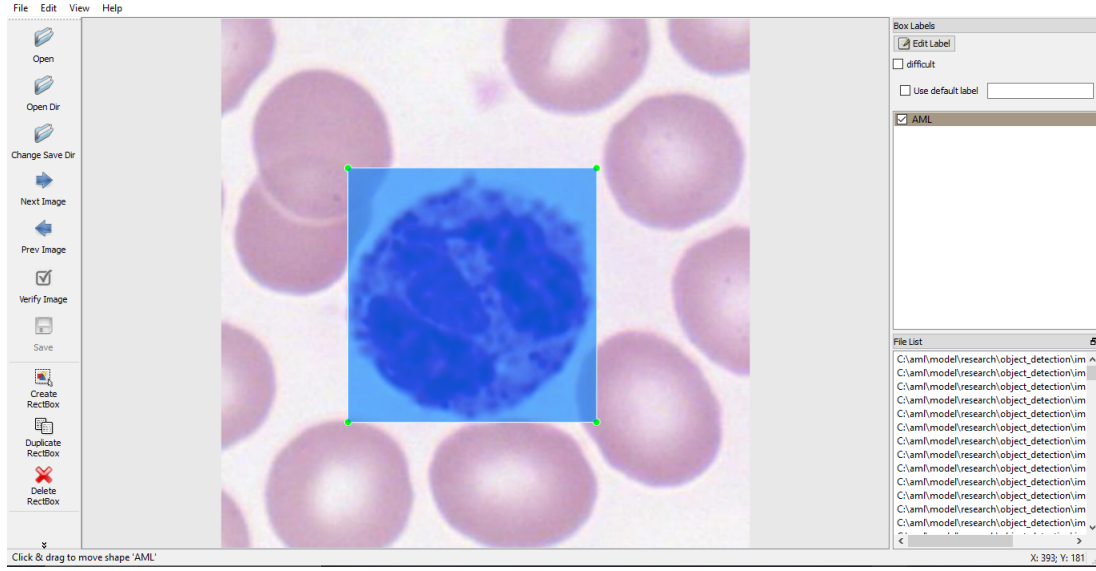
	test	10.12.2019 21:30	Dosya klasörü
	train	10.12.2019 21:31	Dosya klasörü

Şekil 4.7: Programın İçerisinde Bulunan Klasörler

Görüntüleri etiketleme işlemini gerçekleştirmek için “LabelImg[9]” isimli program indirilmiş ve kurulumu gerçekleştirilmiştir. Her mikroskobik görüntüde lösemi bulunan hücreler bu program kullanılarak etiketleme işlemi yapılmıştır. Aşağıda paylaşılmış olan Şekil 4.8 isimli görsel, etiketleme işleminin uygulama ile gerçekleştirildiği sırada anlık olarak alınmıştır.

Etiketleme işlemi yapılmış olan görüntüler .xml uzantısı ile kaydedilmiştir. .xml uzantılı bu kayıtlar görüntülerde işaretlenmiş olan bölgenin koordinatlarını ve o bölgenin açıklamasının ne olduğu ile ilgili bilgiyi içerisinde bulundurmaktadır.

xml_to_csv isimli Python dosyası çalıştırıldığında, LabelImg uygulamasından elde edilmiş



Şekil 4.8: LabelImg Uygulaması ile Çalışılırken Alınan Örnek Bir Görüntü

olan .xml uzantılı dosyalar test.csv ve train.csv olarak adlandırılmış olan iki ayrı Excel dosyasında toplanmış olacaktır. Şekil 4.9 isimli görselde belirtilen kod parçacığı, .xml uzantısına sahip dosyaları bir araya toplama amacı ile tasarlanmıştır. Bu dosya çalıştırıldığında .csv uzantısına sahip olan iki klasör oluşmaktadır (test ve train). Bu klasörlerin içinde yer alan veriler aşağıdaki ekran görüntüsünde paylaşılmıştır.

```

for folder in ['train','test']:
    image_path = os.path.join(os.getcwd(), ('images/' + folder))
    xml_df = xml_to_csv(image_path)
    xml_df.to_csv(('images/' + folder + '_labels.csv'), index=None)
    print('Successfully converted xml to Csv.')

```

Şekil 4.9: xml_to_csv Python Dosyasının Çalıştırılması ile Elde Edilen Çıktı

Bu yazılımın çalışması sayesinde, eğitim aşamasında gerekli olacak olan test.record ve train.record dosyalarının oluşturulması sağlanır. Şekil ??'de paylaşılan kod parçacığı generate_tfrecord Python dosyasının içinde yer almakta ve "LosemiliHucre" etiketine sahip olan görüntülere atıf yapmaktadır. Ayrıca Şekil ?? isimli görselde paylaşılmış olan kod parçacığının komut isteminde (CMD) çalıştırılmasıyla hazırlanan dosya aktif hale getirilmiştir.

LabelImg uygulaması kullanılarak etiketleme işlemi yapılırken, seçilmiş olan bölgeye hangi

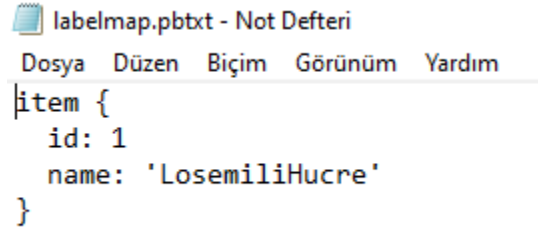
```
def class_text_to_int(row_label):
    if row_label == 'LosemiliHucre':
        return 0
    else:
        None
```

Şekil 4.10: Etiket Haritasının Kodlama Yapılırken Kullanılma Biçimi

```
python generate_tfrecord.py --csv_input=images\test_labels.csv --image_dir=images\test --output_path=test.record
python generate_tfrecord.py --csv_input=images\train_labels.csv --image_dir=images\train --output_path=train.record
```

Şekil 4.11: .record Uzantılı Dosyaların Oluşturulması

isim verilmiş ise o ismi not defterine kayıtlama işlemidir. Şekil 4.12’te bahsedilen etiket haritası LabelImg uygulaması ile etiketleme yapılırken tercih edilmiş olan tanımlamayı içermektedir. Bu etiketleme yapılırken, Türkçe karakter kullanılmaması, unutulmaması gereken bir detaydır.



```
labelmap.pbtxt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
{
  item {
    id: 1
    name: 'LosemiliHucre'
  }
}
```

Şekil 4.12: Etiket Haritası

Eğitim başlatmak için gerekli komutlar sisteme verilmiştir. Kullanılan 685 görüntünün eğitim aşamasının tamamlanması yaklaşık olarak 22 saat sürmüştür. Eğer bu eğitim GPU üzerinden yapılabilmiş olsaydı, eğitim süresi 5 saat gibi kısa bir zamana inmiş olacaktı. Şekil 4.19’de belirtilen kod parçacığı CMD’ de (Komut İstemi) çalıştırılmış ve eğitimin başlaması sağlanmıştır. Bu kodda yer alan train.py dosyası eğitimi yapmak için tasarlanmış olan bir Python dosyasıdır. Faster_rcnn_inception_v2_pets.config [10] dosyası ise 5. Aşamada bahsedilmiş olan önceden eğitilmiş bir dosyadır. Bu dosyanın içerişi tasarlanan sistem ile çalışabilmesi için bazı düzenlemelere tabii tutulmuştur.

Eğitim aşaması tamamlandıktan sonra yapılan eğitimi test edilmek amacıyla bir Python dosyası kullanılmıştır. Şekil ?? ismi verilmiş olan görsel, bu Python dosyasının içerisinden bilgisayar tarafından etiketlenmiş olan lösemili hücre görüntüsünün kullanıcı tarafından görülebilmesi için kullanılan bir kod parçacığıdır. Şekil ?? isimli görsel ise oluşturulan sistemin mikroskobik bir görüntü üzerinde denenmiş ve elde edilmiş olan test sonucudur.


```
python train.py --logtostderr --train_dir=training/ --pipeline_config_path=training/faster_rcnn_inception_v2_pets.config
```

```
INFO:tensorflow:Restoring parameters from C:/tensorflow/models/research/object_detection/faster_rcnn_inception_v2_coco_2018_01_28/model.ckpt
INFO:tensorflow:Starting Session.
INFO:tensorflow:Saving checkpoint to path training/model.ckpt
INFO:tensorflow:Starting Queues.
INFO:tensorflow:global_step/sec: 0
INFO:tensorflow:Recording summary at step 0.
INFO:tensorflow:global step 1: loss = 2.6708 (5.383 sec/step)
INFO:tensorflow:global step 2: loss = 3.0352 (0.251 sec/step)
INFO:tensorflow:global step 3: loss = 3.4884 (0.204 sec/step)
INFO:tensorflow:global step 4: loss = 2.9733 (0.193 sec/step)
INFO:tensorflow:global step 5: loss = 2.2184 (0.191 sec/step)
INFO:tensorflow:global step 6: loss = 2.0321 (0.554 sec/step)
INFO:tensorflow:global step 7: loss = 2.0424 (0.211 sec/step)
INFO:tensorflow:global step 8: loss = 2.0252 (0.208 sec/step)
INFO:tensorflow:global step 9: loss = 2.0053 (0.194 sec/step)
INFO:tensorflow:global step 10: loss = 1.3622 (0.193 sec/step)
INFO:tensorflow:global step 11: loss = 1.8027 (0.197 sec/step)
INFO:tensorflow:global step 12: loss = 1.2485 (0.196 sec/step)
INFO:tensorflow:global step 13: loss = 1.0712 (0.193 sec/step)
INFO:tensorflow:global step 14: loss = 1.6604 (0.189 sec/step)
INFO:tensorflow:global step 15: loss = 1.2657 (0.192 sec/step)
INFO:tensorflow:global step 16: loss = 1.4351 (0.193 sec/step)
INFO:tensorflow:global step 17: loss = 1.2152 (0.192 sec/step)
INFO:tensorflow:global step 18: loss = 1.1165 (0.197 sec/step)
INFO:tensorflow:global step 19: loss = 1.6557 (0.192 sec/step)
INFO:tensorflow:global step 20: loss = 1.7777 (0.200 sec/step)
```

Şekil 4.13: Eğitim Başlatma Kod Satırı

```
cv2.imshow('Object detector', image)
```

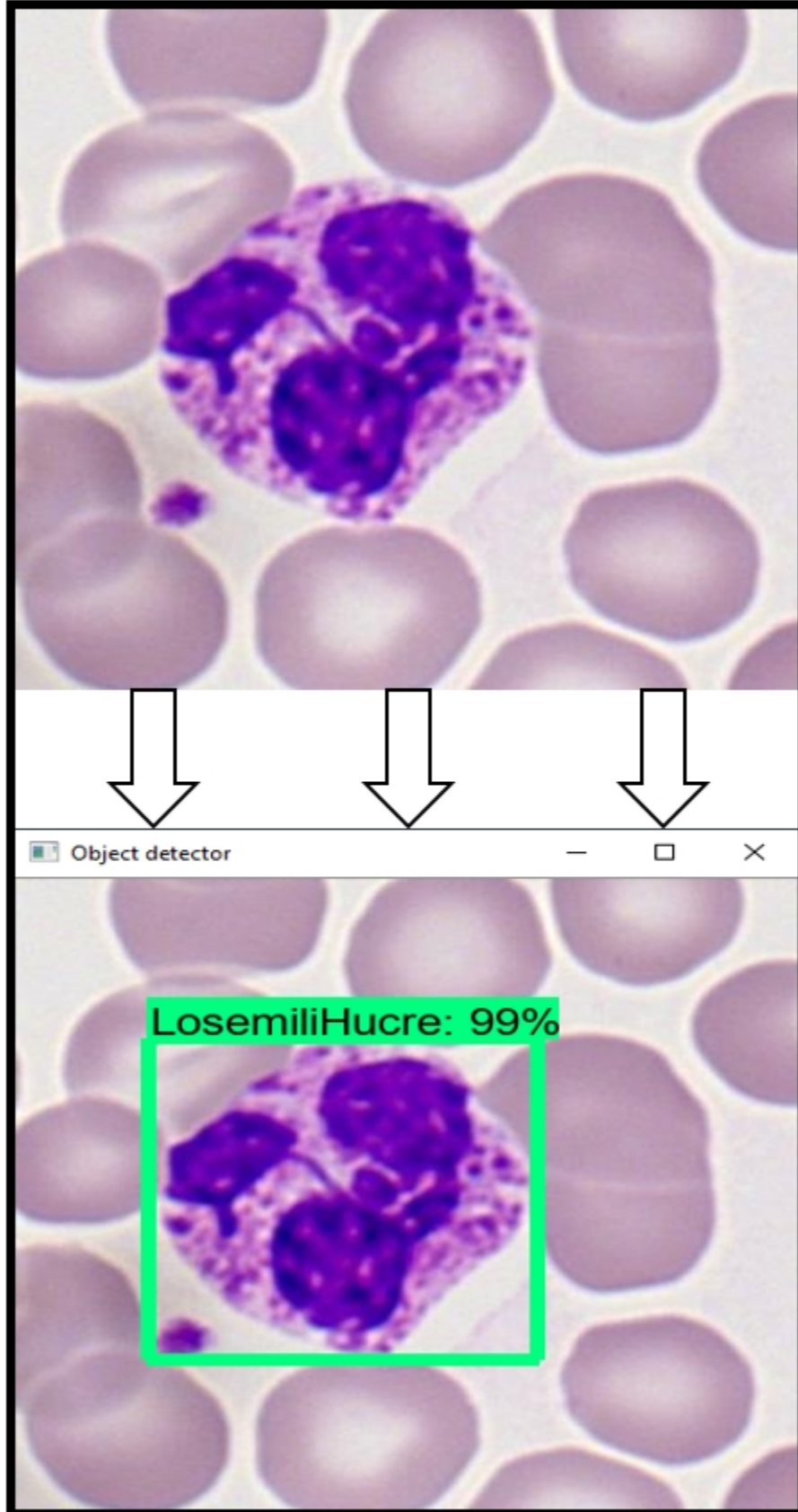
Şekil 4.14: Test Kodunda Yer Alan Kod Parçasığı

4.3 Sonuç

Dünya Sağlık Örgütü (WHO) paylaştığı raporda, sadece 2018 yılında dünyada 18.1 milyon kişiye kanser teşhisi konurken, 9.9 milyon kişinin ise kanser hastalığı sebebiyle hayatını kaybettiğini belirtti. Tüm dünyanın ortak problemlerinden biri olan kanser, erken teşhis ve doğru tedavinin uygulanmasıyla iyileşme olasılığının çok yükseldiği bir hastalıktır. Ayrıca kanser hastalığı sebebiyle gerçekleşen ölümlerin %22 gibi yüksek bir oranının ise geç yapılmış olan teşhis kaynaklı olduğu ortaya çıkmıştır. Hekimler tanı koymak için ayrıca vakit harcamaktadır. Bu sebeple tanı aşamasını hızlandırmaya yardımcı uzman sistemlere olan ihtiyaç her geçen gün artmaktadır.

Hazırlanan tespit programı, kullanışlı ve çok yönlü bir programlama dili Python tercih edilerek lösemi hastalığının teşhis aşamasını hızlandırmak, hekimlerin iş yükünü hafifletebilmek ve kan görüntülerinden lösemili hücrelerinin tespitini yapabilmek amacıyla tasarlanmıştır. “Cancer Imaging Archive” isimli kaynak kullanılarak, lösemili hücre içeren kan görüntüleri toplanmış ve bir veritabanı oluşturulmuştur. Tüm görseller aynı boyutlara getirilmiştir. Ancak çoğu farklı piksel yoğunluğuna sahip olan görsellerin sisteme tanıtılması doğru bir yaklaşım olmayacağından, uygun piksel yoğunluğundaki görüntüler ayıklanmış ve eğitim aşamasına başlanmıştır. Görüntülerin eğitimi aşaması kayıp değeri 0.07’in altına düşene kadar devam ettirilmiştir. CPU temel alınarak ve Tensorflow kütüphanesi aracılığıyla gerçekleştirilen bu eğitimin tamamlanabilmesi yaklaşık olarak 20 saat sürmüştür. Eğer GPU temelli Tensorflow kütüphanesinin bu eğitim için kullanılabilme şansı olaysaydı, yaklaşık 5 saatte eğitimin tamamlanabilmesi mümkün olacaktır.

Çalışma sonunda, eğitim aşamasında kullanılmış 25 adet ve eğitim için kullanılmamış 25 adet, farklı açılardan içerisinde lösemi hastalığı taşıyan hücreleri barındıran görüntüler sistemde test edilmiştir. Bu denemeler neticesinde eğitim aşamasında kullanılmış olan görsellerden %100 başarı ile lösemili hücreler tespit edilmiş, eğitim aşamasında kullanılmayan görsellerden ise %90 başarı oranı ile lösemili hücrelerin tespiti yapılabilmektedir. Durumu teyit etmek amacıyla farklı piksel yoğunluğuna



Şekil 4.15: Yapılan Testlerden Örnek Bir Çıktı

sahip (düşük kaliteli) görseller üzerinde de testler yapılmıştır. Bu testler sonucunda da yapılan hesaplamaları etkileyebilecek herhangi bir farklılığa rastlanmamıştır. Hazırlanan sistemin, eğitilen görsellerin güncellenmesi sayesinde çok farklı alanlarda da kullanılabilmesi mümkün olacaktır. Bu yüzden oluşturulan sistem statik değil dinamik bir yapıyı bünyesinde barındırmaktadır. Yapılan bu çalışma, hekimlerin yerine geçme düşüncesi ile değil, hekimlerin yaptığı işi hızlandırmak ve tedavi aşamasına daha hızlı geçebilmek amacı ile tasarlanmıştır. Oluşturulan sistemde denenmiş 5 adet görsel ve bu görsellerden elde edilen sonuçlar Tablo 1 içerisinde paylaşılmıştır. Sistemin bazı lösemi hücreleri için “Daha düşük olasılıkla lösemi hücresidir” kararını vermesinin sebebi, eğitim aşamasında kullanılmayan bir görsel üzerinde test yapılması veya düşük kaliteden dolayı anlaşılamayan hücre şekilleri yorumlarıyla açıklamak mümkündür.

4.4 Uygulama 2

İmageNet ve Google Görseller isimli uygulamalardan implant içeren jpg uzantısına sahip radyografiler toplanmıştır. Şekil 4.16 isimli görsel toplanmış olan radyografilere ait bir örnektir.



Şekil 4.16: Toplanan Panoramik Radyografilere Bir Örnek

Bu uygulama oluşturulurken, yazılım ve görüntü işleme alanında daha kararlı ayrıca donanım açısından çok daha yeterli olan Python=3.5.2 programlama dilinin kurulumu yapılmıştır.


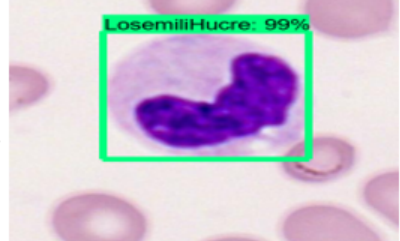
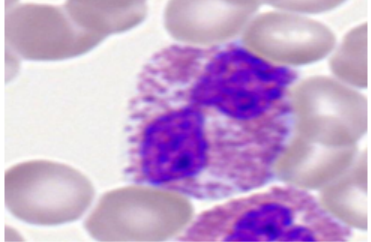
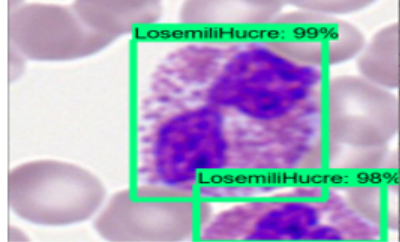
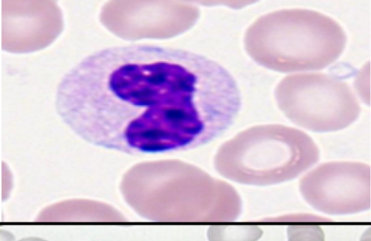
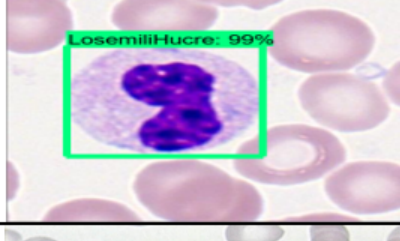
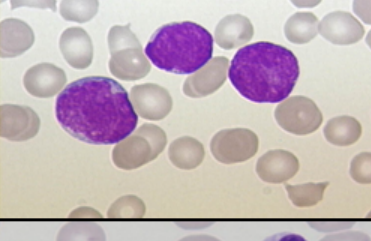
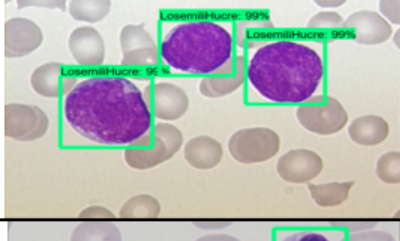
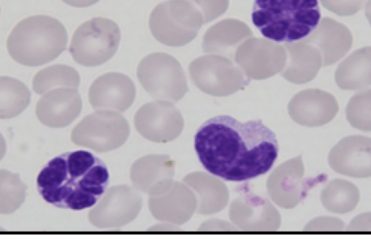
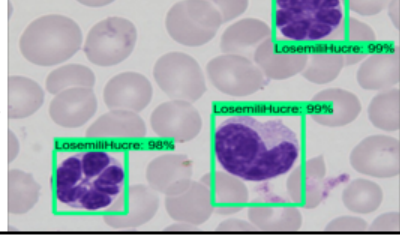
Python programlama diline pillow, lxml, Cython, contextlib2, jupyter, matplotlib, pandas, opencv-python ve tensorflow kütüphaneleri sırasıyla eklenmiş olup, bu kütüphaneler görüntü işleme uygulamaları oluşturulurken kullanılan temel unsurlardandır. Tensorflow kütüphanesi GPU üzerinden de işlem yapabilmektedir ancak denemelerde yaşanan problemlerden dolayı CPU üzerinden tekrar kurulum yapılmıştır. Şekil 4.17 isimli görsel, Komut İstemi penceresinden indirilmiş kütüphaneleri içermektedir.

GitHub uygulamasından Tensorflow Object Detection API' sinin içinde bulunan model dosyası indirilmiş ve gerekli kurulum Komut İsteminde (CMD) yapılmıştır.

Bu eğitimde Faster-RCNN-Inception-V2 modeli kullanılmıştır. Object detection klasörüne bu model eklenmiş ve gerekli düzenlemeler yapılmıştır. Kullanılan dosya önceden eğitimi yapılmış görüntüleri içermektedir.

Dosyalanan radyografilere (314 adet) %20'si test klasörünün, geriye kalan %80'i ise train klasörünün içine taşınmalıdır. Şekil 4.18 adı verilmiş görsel, oluşturulan test ve train isimli klasörleri

Tablo 4.1: Yapılan Testlerden Örnek Bir Çıktı

Test Edilen Görüntü	Elde Edilen Çıktı
	
	
	
	
	

```
C:\>pip install pillow
pip install lxml
pip install Cython
pip install contextlib2
pip install jupyter
pip install matplotlib
pip install pandas
pip install opencv-python
```

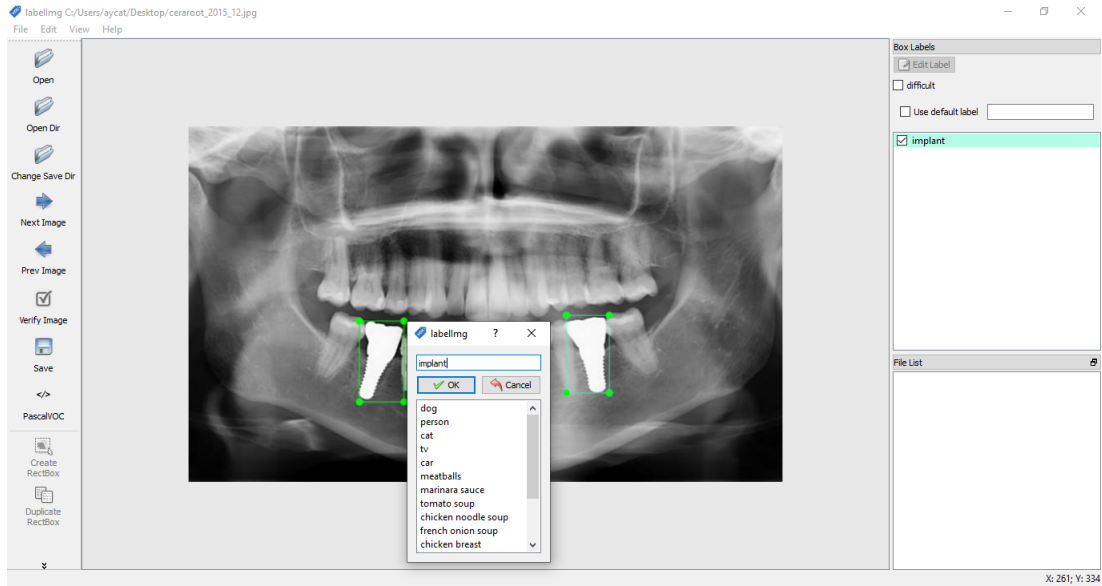
Şekil 4.17: İndirilen Kütüphaneler

içermektedir. Test klasörünün içinde 63, train klasörünün içinde ise 251 adet radyografi örneği bulunmaktadır.



Şekil 4.18: İndirilen Kütüphaneler

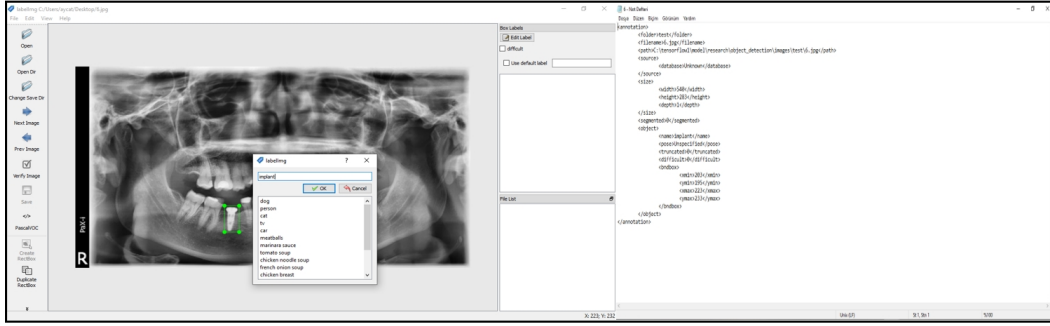
Görüntüleri etiketlemek için “LabelImg” isimli program indirilip kurulumu yapılmıştır. Her radyografide İmplant içeren bölgeler bu program ile etiketlenmiştir. Aşağıya eklenmiş olan Şekil 4.19 isimli görsel, bahsedilen etiketleme uygulaması ile ilgili bir örnek içermektedir.



Şekil 4.19: LabelImg uygulaması ile ilgili yapılan bir örnek

Etiketlenen dosyalar .xml uzantısı ile kaydedilmiştir. Bu kayıt görüntüde işaretlenmiş olan koordinatları ve o bölgenin ne olduğu ile ilgili bilgiyi içermektedir. Şekil 4.4 isimli görselde örneği paylaşılmış olan, .xml uzantılı dosyalar etiketlenmiş radyografiye ait genişlik, yükseklik ve derinlik kriterlerini içerisinde barındırmaktadır.

Bu yazılımın çalıştırıldığında LabelImg uygulaması ile elde edilmiş olan .xml uzantılı dosyalar



Şekil 4.20: Görsel için örnek etiket

test.csv ve train.csv olmak üzere iki ayrı Excel bağlantısında toplanacaktır. Şekil 4.21 isimli görselde belirtilen kod parçacığı .xml uzantılı dosyaları bir araya toplama amacı ile tasarlanmıştır. Bu dosya çalıştırıldığında .csv uzantılı iki klasör oluşmaktadır (test ve train). Bu klasörlerin içinde yer alan veriler aşağıdaki ekran görüntüsünde paylaşılmıştır.

```
for folder in ['train', 'test']:
    image_path = os.path.join(os.getcwd(), ('images/' + folder))
    xml_df = xml_to_csv(image_path)
    xml_df.to_csv(('images/' + folder + '_labels.csv'), index=None)
    print('Successfully converted xml to csv.')
```

test_labels - Not Defteri

filename,width,height,class,xmin,ymin,xmax,ymax

03f41e01-152f-4f24-A960-4A85020050831.jpg,640,346,implant,224,234,241,262

03f41e01-152f-4f24-A960-4A85020050831.jpg,640,346,implant,246,231,262,261

03f41e01-152f-4f24-A960-4A85020050831.jpg,640,346,implant,309,233,417,282

03f41e01-152f-4f24-A960-4A85020050831.jpg,640,346,implant,438,240,455,274

03f41e01-152f-4f24-A960-4A85020050831.jpg,640,346,implant,286,81,304,132

03f41e01-152f-4f24-A960-4A85020050831.jpg,640,346,implant,309,81,324,129

03f41e01-152f-4f24-A960-4A85020050831.jpg,640,346,implant,340,73,360,131

03f41e01-152f-4f24-A960-4A85020050831.jpg,640,346,implant,368,74,390,129

03f41e01-152f-4f24-A960-4A85020050831.jpg,640,346,implant,407,87,425,138

0519_Resnick_06.jpg,300,235,implant,94,1,171,149

12-6-1120x-ray20day20of20Implant20placement_0.jpg,758,506,implant,378,127

1358901864_fig_2-300x300-.jpg,300,300,implant,146,89,218,292

142.jpg,400,238,implant,88,122,117,178

142.jpg,400,238,implant,266,119,298,174

14513_1f.jpg,500,283,implant,215,182,236,217

14513_1f.jpg,500,283,implant,247,179,266,216

14513_1f.jpg,500,283,implant,277,183,289,214

154685-400x300.jpg,400,300,implant,293,1,391,229

1807_2577-rounexp-43-03-0023-gf01.jpg,1030,661,implant,507,449,560,556

180px-Xray_two_cylinders.jpg,180,155,implant,31,21,141,143

19ae3082cd88838021779163c706475.jpg,844,660,implant,186,51,261,438

19ae3082cd88838021779163c706475.jpg,844,660,implant,351,53,472,348

2.-Peri-implantitis-cement-pain-bleeding-dental-implant-kazemi-oral-surgei

210820420MONTHS20POST20OP20RADIOGRAPH.JPG.jpg,758,506,implant,140,75,229,374

210820420MONTHS20POST20OP20RADIOGRAPH.JPG.jpg,758,506,implant,289,76,374,511

210820420MONTHS20POST20OP20RADIOGRAPH.JPG.jpg,758,506,implant,432,47,562,374

2176-9451-dpjo-23-01-80024-gf13.jpg,841,511,implant,627,195,695,317

222M1c_02132018_151253Exported.jpg,1000,524,implant,301,179,329,216

222M1c_02132018_151253Exported.jpg,1000,524,implant,340,165,367,213

222M1c_02132018_151253Exported.jpg,1000,524,implant,381,142,409,208

222M1c_02132018_151253Exported.jpg,1000,524,implant,422,143,444,209

222M1c_02132018_151253Exported.jpg,1000,524,implant,351,314,378,373

222M1c_02132018_151253Exported.jpg,1000,524,implant,383,315,416,367

222M1c_02132018_151253Exported.jpg,1000,524,implant,419,306,446,357

222M1c_02132018_151253Exported.jpg,1000,524,implant,441,301,466,355

222M1c_02132018_151253Exported.jpg,1000,524,implant,553,302,578,364

train_labels - Not Defteri

filename,width,height,class,xmin,ymin,xmax,ymax

21implantsicrown.jpg,450,302,implant,39,17,206,264

2Q_-.jpg,279,180,implant,82,87,97,110

case-6936-2.jpg,813,590,implant,381,244,519,570

case1-2.jpg,680,680,implant,189,440,249,616

case1-2.jpg,680,680,implant,256,445,305,613

case1-2.jpg,680,680,implant,321,439,376,605

case1-2.jpg,680,680,implant,387,438,438,605

case1-2.jpg,680,680,implant,452,439,497,605

case2-7-1.jpg,617,721,implant,313,30,454,565

case_lafter.jpg,400,283,implant,219,112,307,257

ceraaroot-xrays-post-crown.jpg,582,472,implant,246,298,300,373

Ceraaroot_2.jpg,411,246,implant,294,129,323,167

ceraaroot_2015_12.jpg,720,431,implant,204,264,252,334

ceraaroot_2015_12.jpg,720,431,implant,470,265,511,326

Clossick-LL6-x-ray-implant-04.jpg,900,641,implant,400,200,645,546

dec1cf6e809beca2c9d4fbc30005be69.jpg,485,312,implant,294,47,375,281

dec1cf6e809beca2c9d4fbc30005be69.jpg,485,312,implant,115,1,239,241

decayed-tooth-implant-xray.jpg,388,499,implant,182,2,273,210

dental-implant-final-chattanooga-tn.jpg,900,641,implant,446,105,701,552

dental-implant-final-chattanooga-tn.jpg,900,641,implant,753,176,893,632

dental-implant-final-xray-chattanooga-tn.jpg,900,641,implant,193,166,536,541

dental-implant-final-xray-chattanooga-tn.jpg,900,641,implant,520,161,776,641

dental-implant-img4.jpg,555,340,implant,223,1,378,340

dental-implant-img5.jpg,555,340,implant,200,1,341,325

dental-implant-jaw-soft-tissue-radiography-png-favpng-Q1PhTQjeIVcpcbejghYqSrn-.jpg,820,551

dental-implant-jaw-soft-tissue-radiography-png-favpng-Q1PhTQjeIVcpcbejghYqSrn-.jpg,820,551

dental-implant-of-radiograph-BP1X48.jpg,866,1390,implant,421,199,570,639

dental-implant-panoramic-xray.jpg,600,291,implant,180,191,239,228

dental-implant-panoramic-xray.jpg,600,291,implant,202,192,217,237

dental-implant-panoramic-xray.jpg,600,291,implant,364,206,381,249

dental-implant-panoramic-xray.jpg,600,291,implant,402,204,423,245

dental-implant-panoramic-xray.jpg,600,291,implant,204,123,223,160

dental-implant-panoramic-xray.jpg,600,291,implant,221,117,241,165

dental-implant-panoramic-xray.jpg,600,291,implant,350,119,372,168

dental-implant-panoramic-xray.jpg,600,291,implant,391,132,410,170

dental-implant-panoramic.jpg,300,142,implant,166,44,173,66

Şekil 4.21: xml_to_csv dosyasının çalıştırılması ile elde edilen çıktı

Bu programın çalıştırılmasıyla eğitim için gerekli olan test.record ve train.record dosyaları oluşturulur. Şekil 4.22'te paylaşılan kod parçacığı generate_tfrecord dosyasının içinde yer almaktadır ve implant etiketi içeren fotoğraflara atıf yapmaktadır. Ayrıca Şekil 4.23 isimli görselde belirtilen kod parçacığının CMD'de çalıştırılmasıyla hazırlanan dosya aktif hale getirilmiştir.

LabelImg uygulaması kullanılırken etiketleme yapılan alana ne isim verilmiş ise o ismi not defterine kayıtlama işlemidir. Şekil ??'da bahsedilen etiket haritası LabelImg uygulaması ile etiketlerken kullanılan tanımlamayı içermektedir. Bu etiketleme yapılırken Türkçe karakter kullanılmamasına

```
def class_text_to_int(row_label):
    if row_label == 'implant':
        return 1
    else:
        return 0
```

Şekil 4.22: Etiket haritasının kodlamada kullanılma biçimi

```
python generate_tfrecord.py --csv_input=images\test_labels.csv --image_dir=images\test --output_path=test.record
python generate_tfrecord.py --csv_input=images\train_labels.csv --image_dir=images\train --output_path=train.record
```

Şekil 4.23: Record uzantılı dosyaların oluşturulması

dikkat edilmelidir.

```
labelmap - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
item {
  id: 1
  name: 'implant'
}
```

Şekil 4.24: Etiket haritası


Eğitim başlatmak için gerekli komutlar verilir. 314 görüntü için eğitimin tamamlanması yaklaşık 15 saat sürmüştür. Eğer bu eğitim gpu üzerinden yapılırsa bu süre 3 saat gibi kısa bir süreye inecektir. Resim4’te belirtilen kod parçacığı CMD’de (Komut İstemi) çalıştırıldığında eğitim başlayacaktır. Bu kodda yer alan train.py dosyası eğitim için tasarlanmış bir Python dosyasıdır. Faster_rcnn_inception_v2_pets.config dosyası önceden eğitilmiş olarak elde bulunan bir dosyadır. Bu dosyanın içinde eğitilmek istenen bilgi ile ilgili gerekli düzenlemeler yapılmıştır. Ayrıca eğitimin başladığı anın ekran görüntüsü eklenmiştir. Başlatılan eğitim, “loss” değeri sürekli 0,5’in altında izleyene kadar devam ettirilmiştir.

Yapılan eğitimden sonra test etmek için çeşitli görseller programa verilmiştir. Aşağıda programa verilmiş etiketlenmemiş görsel ve sonuçları ile ilgili bir görsel paylaşılmıştır. Bu deneme yaklaşık 100 görüntü için tekrarlanmıştır. Şekil 4.26 isimli görsel, bilgisayar tarafından etiketlenen radyografinin ekranda görülmesi için kullanılmıştır.

4.5 Sonuç

Diş hekimliği alanında görüntü işleme kullanımı uzun bir geçmişe dayanmaktadır. Bu uzun geçmişte en yaygın kullanım diş radyografileridir. Ancak bu radyograflerin kullanımı çeşitli problemleri beraberinde getirebilmektedir.

```
python train.py --logtostderr --train_dir=training/ --pipeline_config_path=training/faster_rcnn_inception_v2_pets.config
```



```
INFO:tensorflow:Restoring parameters from C:/tensorflow1/models/research/object_detection/faster_rcnn_inception_v2_coco_2018_01_28/model.ckpt
INFO:tensorflow:Starting Session.
INFO:tensorflow:Saving checkpoint to path training/model.ckpt
INFO:tensorflow:Starting Queues.
INFO:tensorflow:global_step/sec: 0
INFO:tensorflow:Recording summary at step 0.
INFO:tensorflow:global step 1: loss = 2.6708 (5.383 sec/step)
INFO:tensorflow:global step 2: loss = 3.0352 (0.251 sec/step)
INFO:tensorflow:global step 3: loss = 3.4884 (0.204 sec/step)
INFO:tensorflow:global step 4: loss = 2.9733 (0.193 sec/step)
INFO:tensorflow:global step 5: loss = 2.2184 (0.191 sec/step)
INFO:tensorflow:global step 6: loss = 2.0321 (0.554 sec/step)
INFO:tensorflow:global step 7: loss = 2.0424 (0.211 sec/step)
INFO:tensorflow:global step 8: loss = 2.0252 (0.208 sec/step)
INFO:tensorflow:global step 9: loss = 2.0053 (0.194 sec/step)
INFO:tensorflow:global step 10: loss = 1.3622 (0.193 sec/step)
INFO:tensorflow:global step 11: loss = 1.8027 (0.197 sec/step)
INFO:tensorflow:global step 12: loss = 1.2485 (0.196 sec/step)
INFO:tensorflow:global step 13: loss = 1.0712 (0.193 sec/step)
INFO:tensorflow:global step 14: loss = 1.6604 (0.189 sec/step)
INFO:tensorflow:global step 15: loss = 1.2657 (0.192 sec/step)
INFO:tensorflow:global step 16: loss = 1.4351 (0.193 sec/step)
INFO:tensorflow:global step 17: loss = 1.2152 (0.192 sec/step)
INFO:tensorflow:global step 18: loss = 1.1165 (0.197 sec/step)
INFO:tensorflow:global step 19: loss = 1.6557 (0.192 sec/step)
INFO:tensorflow:global step 20: loss = 1.7777 (0.200 sec/step)
```

Şekil 4.25: Eğitim başlatma kodu ve eğitim anı

```
cv2.imshow('Object detector', image)
```

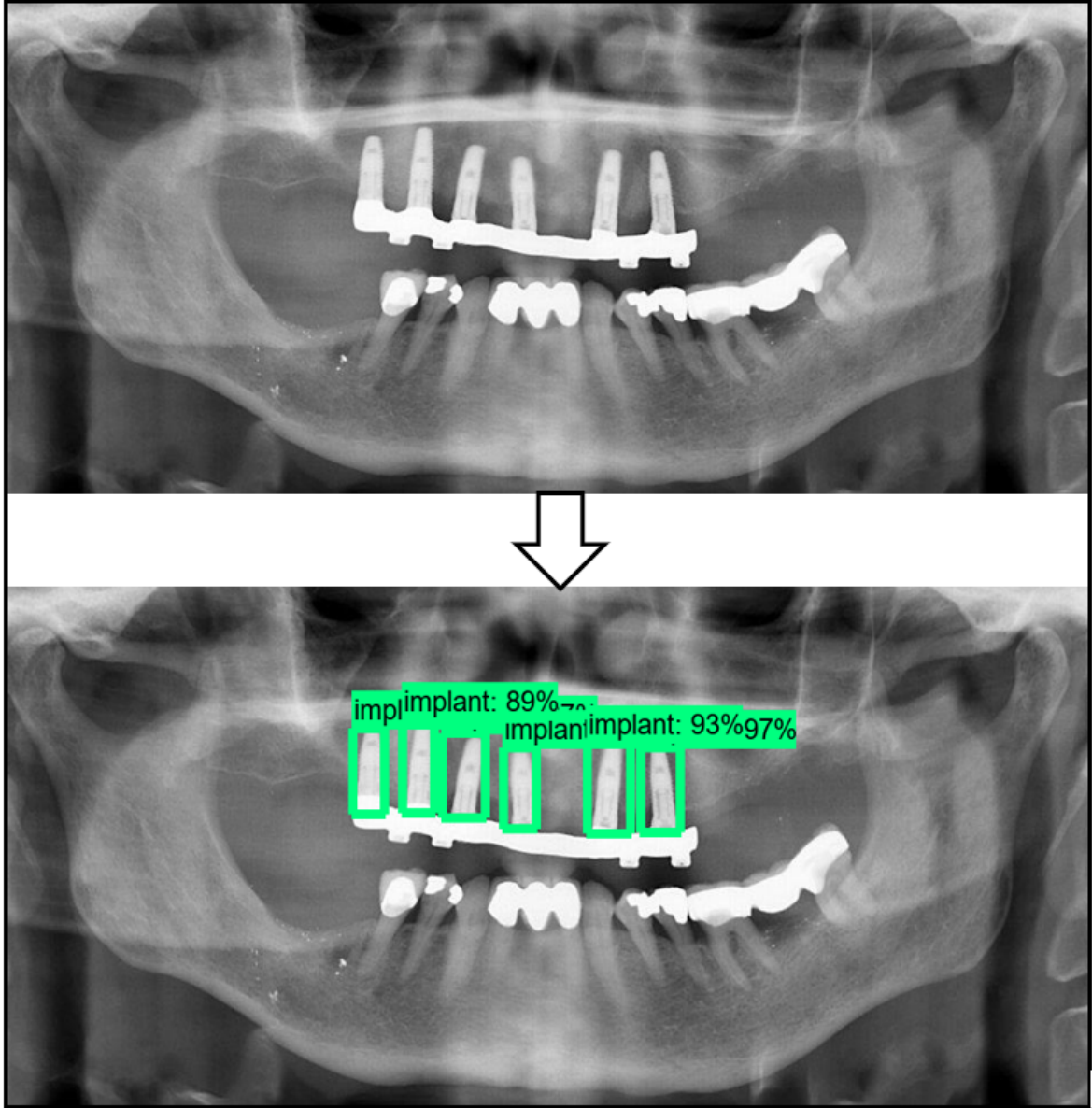
Şekil 4.26: Test kodunda yer alan kod parçacığı

Röntgen ve X-Ray cihazlarının hepsi farklı derinlik, piksel yoğunluğuna sahip olduğundan dolayı kalite düşebilmekte ve hastanın sorununu algılayabilmek güçleşebilmektedir. Özellikle Panoramik Röntgenler tüm çeneyi resmettiğinden çok daha düşük kaliteye sahip olabilmektedir. Bu yüzden diş hekimleri panoramik röntgenleri incelemek için vakit harcamak zorundadırlar.

Hazırlanan görüntü işleme ve tanı programı, kullanışlı ve çok yönlü programlama dili Python kullanılarak diş tedavilerinde tespit işlemini hızlandırmak ve kolaylaştırmak amacıyla radyografik filmlerden implant tespiti yapmak için oluşturulmuştur. Google görseller ve ImageNet gibi çeşitli kaynaklar kullanılarak implant içeren diş radyografilerine ilişkin görüntüler toplanmış ve bir veritabanı oluşturulmuştur. Çoğu farklı boyut ve piksel yoğunluğuna sahip olan radyografilerin sisteme tanıtılması bu sebeple zorlaşmış olup, sistemin kurulduğunda ön görülemeyen aksilikler ortaya çıkmıştır. Yasanan aksiliklere ve tüm olumsuzluklara rağmen uygun piksel yoğunluğundaki radyografik görüntüler elde edilmiştir ve eğitime başlanmıştır.

Görüntü eğitimi kayıp değeri 0.08'in altına düşene kadar devam ettirilmiştir. CPU temelli Tensorflow kütüphanesi ile gerçekleştirilen bir eğitim olduğundan sonlandırılması yaklaşık 15 saat sürmüştür. Eğer GPU temelli Tensorflow kütüphanesi bu eğitim için kullanılabilseydi, yaklaşık 3 saatte eğitimin tamamlanması beklenecekti.

Eğitim sonunda farklı açılardan içerisinde implant bulunduran radyografiler sistemde test edilmiştir. Bu denemeler neticesinde hazırlanan sistemin %100 implant tespit yeteneği olduğu ortaya çıkmıştır. Durumu teyit edebilmek için farklı piksel yoğunluğuna sahip (düşük kaliteli) radyografiler üzerinde de testler yapılmıştır. Bu testlerde de sonucu etkileyecek herhangi bir farklılığa rastlanmamıştır.



Şekil 4.27: Test Görself ve Elde Edilen Çıktı

Hazırlanan sistemin, eğitilen görsellerin güncellenmesi ile çok farklı alanlarda da çalışması mümkündür. Bu yüzden oluşturulan sistem statik değil dinamik bir yapıyı bünyesinde barındırmaktadır. Eğer istenilecek olursa proses geliştirilip radyografide bulunan çeşitli sorunlar (çürük, diş kaybı, dolgu, vb.) sisteme dahil edilerek detaylı bir rapor hazırlama aracı oluşturulabilir. Bu raporlama aracının, diş hekimlerinin yaptığı işi hızlandırmak ve hızlı aksiyon alabilmek amacı ile tasarlanması beklenecektir.

Oluşturulan sistemde denenmiş 5 adet radyografi görseli ve bu görsellerden elde edilen sonuçlar Tablo 2 içerisinde paylaşılmıştır. Sistemin radyografilerde bulunan bazı implantlar için “Daha düşük olasılıkla implanttır.” kararını vermesinin sebebini, radyografinin üzerinde bulunan bir bulanıklık ya da implantın üzerine denk gelen bir yazı şeklinde açıklamak mümkündür.

Referanslar

[1] İnik Ö., Ülker E., Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri, Gaziosmanpaşa Bilimsel Araştırma Dergisi (GBAD), Cilt NO:6, Sayı No:3, Sayfano: 85-104, Tokat, Aralık-2017

[2] R. Girshick, Fast R-CNN, Proceedings Of The IEEE International Conference On Computer Vision, Cilt 2015 Inter, Pp. 1440-1448, 2015

[3] Mesut Pişkin, [Online]. Available: <http://mesutpiskin.com/blog/nesne-tespiti-ve-nesne-tanima.html>, [Accessed: 07.08.2020]

[4] Cancer Imaging Archive, [Online]. Available: <https://www.cancerimagingarchive.net/collections>, [Accessed: 05.08.2020]

[5] Fırat Özgül, Python 3 için Türkçe Kılavuz Sürüm 3 , [Online]. Available: <https://belgeler.yazbel.com/python-istihza.pdf>, [Accessed: 10.08.2020]

[6] Available: <https://github.com/EdjeElectronics/TensorFlow-Object-Detection-API-Tutorial-Train-Multiple-Objects-Windows-10>, [Accessed: 06.07.2020]

[7] Available: <https://github.com/tensorflow/models>, [Accessed: 06.07.2020]

[8] Available: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md [Accessed: 05.08.2020]

[9] LabelImg, [Online]. Available: <https://github.com/tzutalin/labelImg>, [Accessed: 07.08.2020]

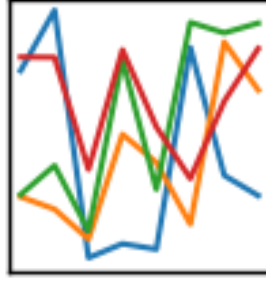
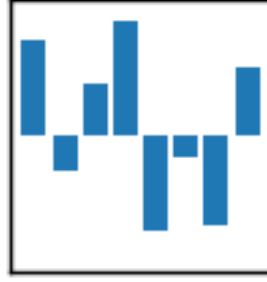
[10] Available: https://github.com/tensorflow/models/blob/master/research/object_detection/samples/configs/faster_rcnn_inception_v2_pets.config, [Accessed: 05.08.2020]

Tablo 4.2: Sistemde denenmiş olan 5 adet radyografiden elde edilen sonuçlar

Test Edilen Radyografi	Elde Edilen Radyografi Çıktısı

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



5. Python Pandas & Pandas-Profiling

Python Pandas ve Pandas-Profiling ile Veri Analizi

İ. Hakkı CEDİMOĞLU¹

¹Sakarya Üniversitesi, Bilgisayar ve Bilişim Bilimleri Fakültesi, Bilişim Sistemleri Mühendisliği Bölümü, Serdivan, Sakarya, Türkiye

5.1 Giriş

Veri analizi, veriden anlam çıkarma sürecidir. Python Programlama dili, finans, ekonomi, istatistik, reklam ve web analitiği gibi alanlarında akademik ve ticari olarak kullanılmaktadır. Aynı zamanda, Python Programı Veri analizi için en çok tercih edilen araçlardan biri haline gelmiştir. Bilişim alanında birçok veri işleme aracı bulunmaktadır. Bu araçlar arasında, ilişkisel veritabanları (SQL Server, Oracle), elektronik tablolar (Excel), olay işleme sistemleri (Apache Spark gibi) ve R Programlama dili ve Python Programlama dili ve Python pandas kütüphanesigibi araçlar da yer almaktadır.

Veri analizinde pandas kütüphanesi kullanılmaktadır. Pandas kütüphanesi ile veri seti üzerinde keşifsel veri analizi yapılabilmektedir. Pandas profiling kütüphanesi ise hızlı bir şekilde bütünlük veri analizi raporu hazırlanabilmektedir. Pandas kütüphanesi, veriyi alır, veriyi okur ve veriye dataframe nesnesinde gösterir[1].

5.2 Pandas Kütüphanesi

Pandas kütüphanesi, Python programlama dili için geliştirilen açık kaynak kodlu bir veri analizi kütüphanesidir. İki farklı pandas veri yapısı yaygın olarak kullanılmaktadır[2]. Bunlar: Series ve DataFrame'dir.

Series, pandas veri yapılarından birisi olan Series, değişik veri türlerinden, aynı veri türünü saklayabilen tek boyutlu dizilerdir. pandas "Dataframe"de her veri sütunu ise sadece bir tek veri türünü saklamaktadır.

Görsel olarak, bir pandas DataFrame'in görünümü, indeks, satır ve sütunlardan oluşmaktadır. Bunlar DataFrame'nin üç bileşeni(component) olarak adlandırılır.

DataFrame yapısı, Veritabanı Yönetim Sistemlerindeki tablo yapısına benzemektedir[6]. Dikdörtgen grid'lerden meydana gelmiştir. Her bir satırda bir kayıt vardır. Her bir sütun ise, belirli bir değişken için veri içeren bir vektördür. Python pandas kütüphanesinde bulunan DataFrame, R programlama dilindeki dataframe'in benzeridir.

5.3 Python Pandas Uygulaması

Pandas kütüphanesi versiyon kontrolü şu şekilde yapılabilir.

```
# pandas versiyon
print("pandas_versiyon")
pd.__version__
```

Python kodunun ekran çıktısı:

```
'0.25.3'
```

pandas ilk olarak, read_csv fonksiyonunu kullanarak diskten veya adresten belleğe(memory) ve DataFrame'e veriyi alır. EDA ile eksik veriler belirlenir[3], tekrarlı veriler belirlenir, veriler sayılır, ortalama ve medyan hesaplanabilir, verilerin birbirleriyle korelasyonu bulunabilir.

Pandas uygulaması için veri seti linki[7]:

https://raw.githubusercontent.com/jlyang1990/Spark_Python_Do_Big_Data_Analytics/master/auto-data.csv

Dataset'in Görünümü:

```
MAKE, FUELTYPE, ASPIRE, DOORS, BODY, DRIVE, CYLINDERS, HP, RPM, ...
MPG-CITY, MPG-HWY, PRICE
subaru , gas , std , two , hatchback , fwd , four , 69 , 4900 , 31 , 36 , 5118
chevrolet , gas , std , two , hatchback , fwd , three , 48 , 5100 , 47 , 53 , 5151
mazda , gas , std , two , hatchback , fwd , four , 68 , 5000 , 30 , 31 , 5195
toyota , gas , std , two , hatchback , fwd , four , 62 , 4800 , 35 , 39 , 5348
mitsubishi , gas , std , two , hatchback , fwd , four , 68 , 5500 , 37 , 41 , 5389
```

Kütüphaneler import deyimi ile deklare edilir.

```
#kutuphanelerin deklarasyonu
import pandas as pd
%matplotlib inline

# Dataset'in pandas dataframe ile okunması

df = pd.read_csv('https://raw.githubusercontent.com/jlyang1990/Spark_Python_Do_Big_Data_Analytics/master/auto-data.csv')

# dataframe'in ekranda gösterilmesi

print(df)

# dataframe'in ekranda gösterilmesi

df

# DataFrame'deki veriler hakkında özet bilgi
```

```
print(df.info())
```

info() Metodu: info() metodu, dataset ile ilgili olarak satır ve sütun sayısını, null olmayan verilerin sayısını göstermektedir[8].

Her sütundaki veri türü de bu metot ile gösterilmektedir. DataFrame'in kullandığı bellek miktarı da gösterilmektedir.

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 197 entries, 0 to 196
```

```
Data columns (total 12 columns):
```

```
MAKE          197 non-null object
```

```
FUELTYPE      197 non-null object
```

```
ASPIRE        197 non-null object
```

```
DOORS         197 non-null object
```

```
BODY          197 non-null object
```

```
DRIVE         197 non-null object
```

```
CYLINDERS     197 non-null object
```

```
HP            197 non-null int64
```

```
RPM           197 non-null int64
```

```
MPG-CITY      197 non-null int64
```

```
MPG-HWY       197 non-null int64
```

```
PRICE         197 non-null int64
```

```
dtypes: int64(5), object(7)
```

```
memory usage: 18.6+ KB
```

```
None
```

```
# pandas kütüphanesi versiyonu bulunması
```

```
print(pd.__version__)
```

```
Ekran Çıktısı:
```

```
0.25.3
```

```
# Python Yazılımı Versiyonu
```

```
import sys
```

```
print(sys.version)
```

```
3.7.3 (default, Mar 27 2019, 17:13:21) [MSC v.1915 64 bit (AMD64)]
```

```
# DataFrame'deki ilk 5 veri
```

```
print(df.head(5))
```

```
MAKE FUELTYPE ASPIRE DOORS BODY DRIVE CYLINDERS HP RPM \
0      subaru      gas  std  two  hatchback  fwd      four  69
4900
1  chevrolet      gas  std  two  hatchback  fwd      three  48
5100
2      mazda      gas  std  two  hatchback  fwd      four  68
5000
```

```

3      toyota      gas      std      two      hatchback      fwd      four      62
4800
4      mitsubishi  gas      std      two      hatchback      fwd      four      68
5500

```

```

MPG-CITY  MPG-HWY  PRICE
0         31      36   5118
1         47      53   5151
2         30      31   5195
3         35      39   5348
4         37      41   5389

```

```
# DataFrame'deki ilk 5 veri
```

```
df.head(5)
```

```
# DataFrame'deki ilk 5 veri
```

```
df.head()
```

```
# DataFrame'deki ilk 5 veri
```

```
print(df.head())
```

```

MAKE FUELTYPE ASPIRE DOORS      BODY DRIVE CYLINDERS  HP  RPM  \
0      subaru      gas      std      two      hatchback      fwd      four      69
4900
1      chevrolet      gas      std      two      hatchback      fwd      three      48
5100
2      mazda      gas      std      two      hatchback      fwd      four      68
5000
3      toyota      gas      std      two      hatchback      fwd      four      62
4800
4      mitsubishi  gas      std      two      hatchback      fwd      four      68
5500

```

```

MPG-CITY  MPG-HWY  PRICE
0         31      36   5118
1         47      53   5151
2         30      31   5195
3         35      39   5348
4         37      41   5389

```

```
# DataFrame'deki son 5 veri
```

```
df.tail()
```

```

MAKE  FUELTYPE  ASPIRE  DOORS  BODY  DRIVE  CYLINDERS  HP  RPM  MPG-CITY  MPG-HWY  PRICE
192  bmw      gas      std    four  sedan  rwd      six   182  5400  15      20      36880
193  porsche  gas      std    two   convertible  rwd  six   207  5900  17      25      37028
194  mercedes-benz  gas      std    four  sedan  rwd      eight  184  4500  14      16      40960
195  bmw      gas      std    two   sedan  rwd      six   182  5400  16      22      41315
196  mercedes-benz  gas      std    two   hardtop  rwd  eight  184  4500  14      16      45400

```

```
# DataFrame'deki son 5 veri
```



```
df.tail(5)
MAKE FUELTYPE ASPIRE DOORS BODY DRIVE CYLINDERS HP RPM MPG-CITY MPG-HWY PRICE
192 bmw gas std four sedan rwd six 182 5400 15 20 36880
193 porsche gas std two convertible rwd six 207 5900 17 25 37028
194 mercedes-benz gas std four sedan rwd eight 184 4500 14 16 40960
195 bmw gas std two sedan rwd six 182 5400 16 22 41315
196 mercedes-benz gas std two hardtop rwd eight 184 4500 14 16 45400
```

```
# dataframe'deki son 5 veri
```

```
print(df.tail())
```

```
MAKE FUELTYPE ASPIRE DOORS BODY DRIVE CYLINDERS HP \
192 bmw gas std four sedan rwd six 182
193 porsche gas std two convertible rwd six 207
194 mercedes-benz gas std four sedan rwd eight 184
195 bmw gas std two sedan rwd six 182
196 mercedes-benz gas std two hardtop rwd eight 184

RPM MPG-CITY MPG-HWY PRICE
192 5400 15 20 36880
193 5900 17 25 37028
194 4500 14 16 40960
195 5400 16 22 41315
196 4500 14 16 45400
```

```
# dataframe'deki son 5 veri
```

```
print(df.tail(5))
```

```
MAKE FUELTYPE ASPIRE DOORS BODY DRIVE CYLINDERS HP \
192 bmw gas std four sedan rwd six 182
193 porsche gas std two convertible rwd six 207
194 mercedes-benz gas std four sedan rwd eight 184
195 bmw gas std two sedan rwd six 182
196 mercedes-benz gas std two hardtop rwd eight 184

RPM MPG-CITY MPG-HWY PRICE
192 5400 15 20 36880
193 5900 17 25 37028
194 4500 14 16 40960
195 5400 16 22 41315
196 4500 14 16 45400
```

```
# dataframe'deki tesadüfi 5 adet verinin listesi
```

```
df.sample(5)
```

```
MAKE FUELTYPE ASPIRE DOORS BODY DRIVE CYLINDERS HP RPM MPG-CITY MPG-HWY PRICE
184 mercedes-benz diesel turbo four sedan rwd five 123 4350 22 25 31600
81 toyota gas std four sedan fwd four 70 4800 28 34 9258
23 plymouth gas std four sedan fwd four 68 5500 31 38 6692
27 nissan gas std four sedan fwd four 69 5200 31 37 6849
9 mazda gas std two hatchback fwd four 68 5000 31 38 6095
```

```
# dataframe'deki bütün verilerin listesi
```

```
df
```

```

MAKE      FUELTYPE      ASPIRE  DOORS  BODY  DRIVE  CYLINDERS  HP  RPM  MPG-CITY  MPG-HWY  PRICE
0        subaru    gas     std   two  hatchback  fwd    four    69  4900   31    36    5118
1        chevrolet  gas     std   two  hatchback  fwd    four    48  5100   47    53    5151
2        mazda     gas     std   two  hatchback  fwd    four    68  5000   30    31    5195
3        toyota    gas     std   two  hatchback  fwd    four    62  4800   35    39    5348
4        mitsubishi gas     std   two  hatchback  fwd    four    68  5500   37    41    5389
...
192       bmw      gas     std   four  sedan  rwd    six    182  5400   15    20    36880
193       porsche  gas     std   two  convertible  rwd    six    207  5900   17    25    37028
194       mercedes-benz gas     std   four  sedan  rwd    eight  184  4500   14    16    40960
195       bmw      gas     std   two  sedan  rwd    six    182  5400   16    22    41315
196       mercedes-benz gas     std   two  hardtop  rwd    eight  184  4500   14    16    45400
197 rows \times$ 12 columns

```

```
# dataframe'deki nümerik alanları listesi
```

```
df.describe()
```

```

HP      RPM      MPG-CITY      MPG-HWY  PRICE
count  197.000000  197.000000  197.000000  197.000000  197.000000
mean   103.604061  5118.020305  25.152284  30.629442  13279.644670
std    37.639205  481.035914  6.437863  6.836259  8010.334218
min    48.000000  4150.000000  13.000000  16.000000  5118.000000
25%    70.000000  4800.000000  19.000000  25.000000  7775.000000
50%    95.000000  5200.000000  24.000000  30.000000  10345.000000
75%    116.000000  5500.000000  30.000000  34.000000  16503.000000
max    262.000000  6600.000000  49.000000  54.000000  45400.000000

```

```
# dataframe'deki nümerik olmayan alanların frekans dağılımı
```

```
print(df['DOORS'].value_counts())
```

```
print(df['BODY'].value_counts())
```

```
four      112
```

```
two       85
```

```
Name: DOORS, dtype: int64
```

```
sedan     92
```

```
hatchback 67
```

```
wagon     24
```

```
hardtop   8
```

```
convertible 6
```

```
Name: BODY, dtype: int64
```

```
# DataFrame'deki toplam veri sayısı
```

```
print(df.size)
```

```
2364
```

```
# DataFrame'deki toplam sütun sayısı
```

```
print(len(df.columns))
```

```
12
```

```
# DataFrame'deki toplam veri sayısı
```

```
print(len(df)) #toplam veri sayısı
```

```
197
```

```
# dataframe'deki sütun isimleri
```

```
print(df.columns)
```

```
Index(['MAKE', 'FUELTYPE', 'ASPIRE', 'DOORS', 'BODY', 'DRIVE', 'CYLINDERS',
       'HP', 'RPM', 'MPG-CITY', 'MPG-HWY', 'PRICE'],
      dtype='object')
```

```

# sütunlardaki eksik veri sayısı
df.isnull().sum()

MAKE          0
FUELTYPE      0
ASPIRE        0
DOORS         0
BODY          0
DRIVE         0
CYLINDERS     0
HP            0
RPM           0
MPG-CITY      0
MPG-HWY       0
PRICE         0
dtype: int64

# dataFramedeki eksik veriler
df.isnull()
MAKE  FUELTYPE  ASPIRE  DOORS  BODY  DRIVE  CYLINDERS  HP  RPM  MPG-CITY  MPG-HWY  PRICE
0     False   False   False  False  False  False  False  False  False  False  False
1     False   False   False  False  False  False  False  False  False  False  False
2     False   False   False  False  False  False  False  False  False  False  False
3     False   False   False  False  False  False  False  False  False  False  False
4     False   False   False  False  False  False  False  False  False  False  False
...
192   False   False   False  False  False  False  False  False  False  False  False
193   False   False   False  False  False  False  False  False  False  False  False
194   False   False   False  False  False  False  False  False  False  False  False
195   False   False   False  False  False  False  False  False  False  False  False
196   False   False   False  False  False  False  False  False  False  False  False
197 rows x 12 columns

# Tekrarlı verilerin sayısı
df.duplicated()

0     False
1     False
2     False
3     False
4     False
...
192   False
193   False
194   False
195   False
196   False
Length: 197, dtype: bool

# Tekrarlı verilerin toplamı
df.duplicated().sum()

0 #sıfır
# toplam satır ve sütun sayısı

```

```
df.shape
```

Ekran Çıktısı:

```
(197, 12)
```

sütunlardaki toplam veri sayısı

```
df.count()
```

```
MAKE          197
FUELTYPE      197
ASPIRE        197
DOORS         197
BODY          197
DRIVE         197
CYLINDERS     197
HP            197
RPM           197
MPG-CITY      197
MPG-HWY       197
PRICE         197
```

```
dtype: int64
```

satır ve sütun NaN verilerin sayılması

```
df.isnull().sum().sum()
```

Ekran Çıktısı:

```
0
```

5.4 Pandas Profiling Kütüphanesi

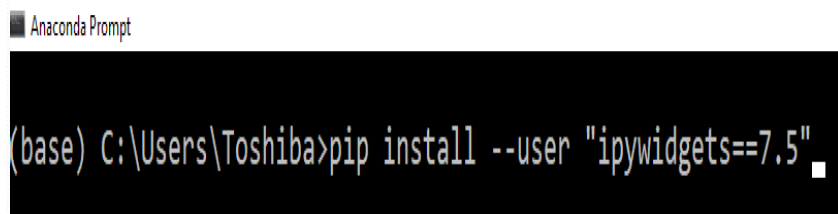
pandas kütüphanesi ile EDA keşifsel veri analiz işlemleri tek tek yapılırken, ‘pandas profiling’ kütüphanesi ile veri analizi tek bir rapor şeklinde hazırlanabilir[5].

#anaconda prompt’tan pandas profiling kütüphanesi yüklenmesi pip install pandas-profiling

‘ipywidgets’ kütüphanesi, Jupyter Notebooks’ta kullanılan etkileşimli HTML widget’larıdır.

Jupyter notebook’ta verilerdeki değişimin etkisi widget ile izlenebilir[5].

#anaconda prompt’tan ‘ipywidgets’ kütüphanesi yükleme



```
Anaconda Prompt
(base) C:\Users\Toshiba>pip install --user "ipywidgets==7.5"
```

Şekil 5.1

Pandas profiling kütüphanesi import edilir ve versiyon kontrolü yapılır.

```
%config IPCompleter.greedy=True
# auto completion
# used with tab
# kutuphanenin import edilmesi
import pandas as pd
# pandas_profiling kutuphanesi
from pandas_profiling import ProfileReport
```

```
import ipywidgets
ipywidgets.__version__
```

Ekran Çıktısı:

```
'7.5.0'
```

```
import pandas_profiling
# pandas_profiling versiyon
print("pandas_profiling_versiyon_")
pandas_profiling.__version__
```

Ekran Çıktısı:

```
pandas_profiling versiyon
'2.4.0'
```

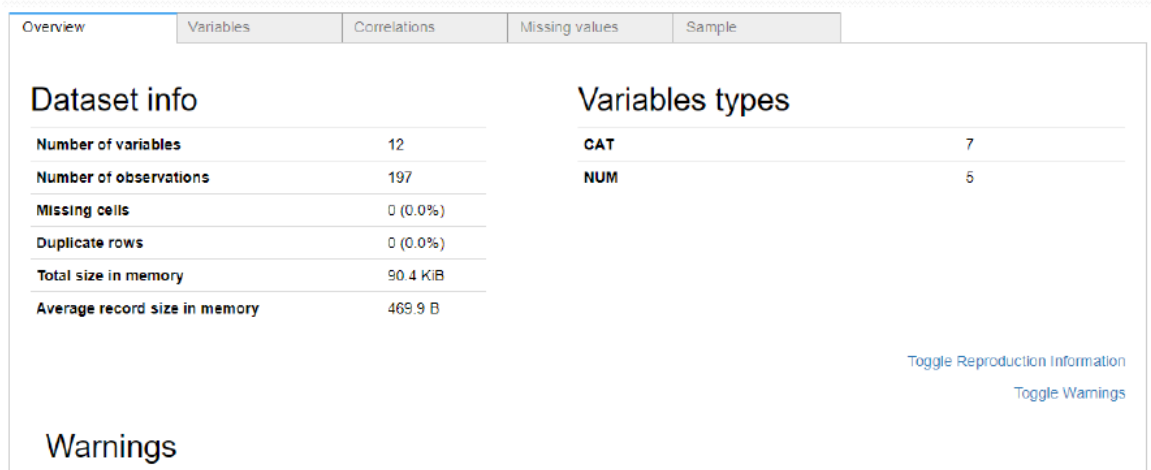
pandas_profiling kütüphanesi, Python programlama dilinde kullanılan açık kaynak kodlu, bir veri analizi rapor kütüphanesidir[9]. pandas_profiling kütüphanesi kullanılarak EDA – Keşifsel Veri Analizi için fonksiyonlar ile yapılan işlemler, bu kütüphane ile Jupyter Notebook'ta, bir veri analizi raporu hazırlanabilir. Aynı zamanda, Html olarak bir veri analizi raporu da hazırlanabilir.

Explanatory Data Analysis (EDA) - Kesifsel Veri Analizi # pandas_profiling Jupyter Notebook'ta Rapor Oluşturulması # pandas_profiling kutuphanesi kullanılarak # dataframe'deki veriler hakkında "html" rapor #hazırlanması

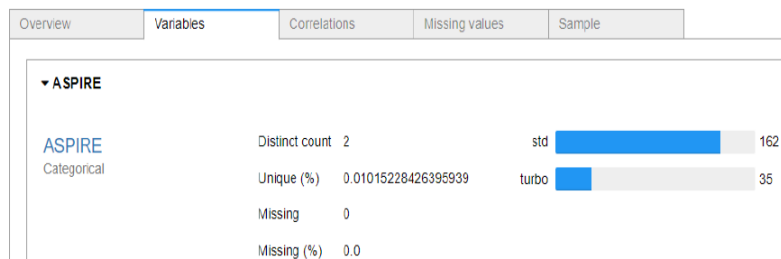
```
from pandas_profiling import ProfileReport
report = ProfileReport(df)
report.to_file('C:/Users/Toshiba/Analiz_Raporu.html')
ProfileReport(df)
```

Oluşturulan raporda 5 sekme vardır:

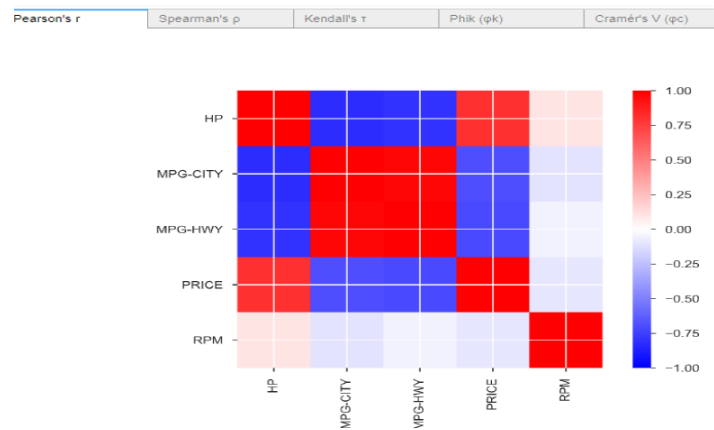
- Overview
- Variables
- Correlations
- Missing Values
- Sample



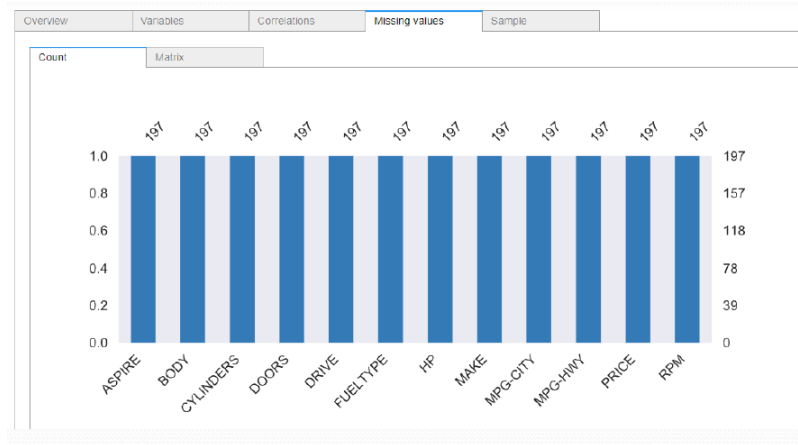
Şekil 5.2: Overview



Şekil 5.3: Variables



Şekil 5.4: Correlations



Şekil 5.5: Missing Values

First rows

	ASPIRE	BODY	CYLINDERS	DOORS	DRIVE	FUELTYPE	HP	MAKE	MPG-CITY	MPG-HWY	PRICE	RPM
0	std	hatchback	four	two	fwd	gas	69	subaru	31	36	5118	4900
1	std	hatchback	three	two	fwd	gas	48	chevrolet	47	53	5151	5100
2	std	hatchback	four	two	fwd	gas	68	mazda	30	31	5195	5000
3	std	hatchback	four	two	fwd	gas	62	toyota	35	39	5348	4800
4	std	hatchback	four	two	fwd	gas	68	mitsubishi	37	41	5389	5500
5	std	hatchback	four	two	fwd	gas	60	honda	38	42	5399	5500
6	std	sedan	four	two	fwd	gas	69	nissan	31	37	5499	5200
7	std	hatchback	four	two	fwd	gas	68	dodge	37	41	5572	5500
8	std	hatchback	four	two	fwd	gas	68	plymouth	37	41	5572	5500
9	std	hatchback	four	two	fwd	gas	68	mazda	31	38	6095	5000

Şekil 5.6: Sample

5.5 Sonuç

Pandas kütüphanesi ile temel veri analizi işlemleri ayrı ayrı yapılabilirken, pandas profiling ile dataframe hakkındaki temel veri analizi ve korelasyon işlemleri bir rapor halinde alınabilir.

Referanslar

[1] The Python Workshop : A New, Interactive Approach to Learning Python, Andrew Bird, Lau Cher Han, Mario Corchero Jimenez, Graham Lee, Corey Wade, Pack Publishing, 2019.

[2] Python for Data Analysis, Wes McKinney, O'Reilly Media, Inc., 2013.

[3] <https://thispointer.com/python-pandas-count-number-of-nan-or-missing-values-in-dataframe-also-row-column-wise>

[4] <https://staesthetic.wordpress.com/2020/01/27/easy-data-understanding-using-pandas-profiling>

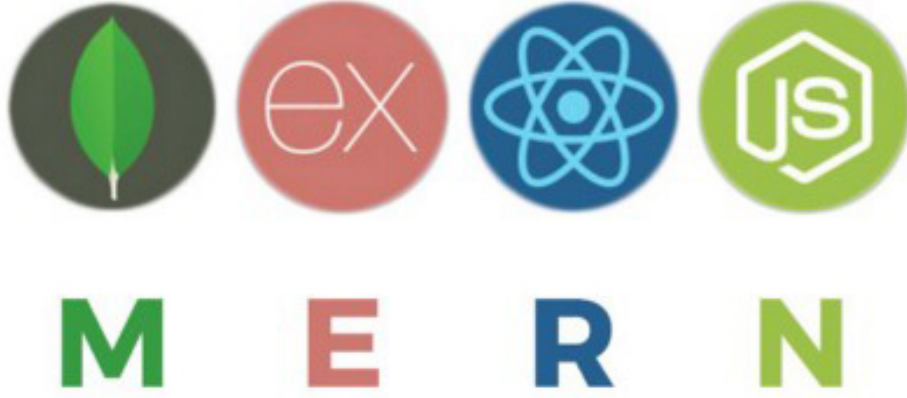
[5] <https://github.com/jupyter-widgets/ipywidgets>

[6] <https://www.datacamp.com/community/tutorials/pandas-tutorial-dataframe-python>

[7] https://raw.githubusercontent.com/jlyang1990/Spark_Python_Do_Big_Data_Analytics/master/auto-data.csv

[8] <https://www.educative.io/blog/pandas-cheat-sheet>

[9] <https://github.com/pandas-profiling/pandas-profiling>



6. Context ile React Hooks

Context ile React Hooks: MERN Kullanarak Full-Stack Uygulamaları Geliştirme

A.K. Margai Wangara, Ussene Aliasse Jamal¹

¹Sakarya Üniversitesi, Serdivan, Sakarya, Türkiye

6.1 Giriş

MERN stack kullanarak bir uygulamayı geliştireceğiz. Bu bölümün sonuna geldiğimizde MERN Nedir, nasıl kullanılıyor, ne için kullanılıyor öğrenmiş olacaksınız. Son zamanlarda web teknolojileri baya geliştirilmiş, mesela Javascript dili sadece front-end kısmında kullanılabiliyorlardı ama bugünlerde Javascript web, mobil ve masaüstü tabanlı uygulamaları geliştirmek için kullanılır. Ayrıca, ReactJS nedir, React Hooks ve Context ne ise yarar öğreneceksiniz.

6.2 MERN Nedir?

MERN: Farklı web teknolojilerinin kelimelerinin baş hariflerinden türetilmiş bir isimdir. MERN kelimeyi açarsak **MongoDB**, **Express**, **React** ve **Node.js** kelimeler çıkıyor.

- (i) **MongoDB:** NoSQL bir veritabanıdır. NoSQL tabloları kullanmıyor demektir. Tablolar yerine document kullanıyor, verileri ise **JSON** formatında saklanıyor. Node.js back-end uygulamalarında genellikle MongoDB veritabanı kullanılır.
- (ii) **Express:** Back-end kısmında kullanılan bir framework'tir. Python'un Flask ve PHP'nin Laravel gibi bir teknolojidir. Kolay bir şekilde sunucu tabanlı uygulamamızı geliştirmeyi sağlar. Express kullanarak route'lar, controller metodlar ve view'lar çok uğraşmadan oluşturabiliriz.
- (iii) **ReactJS:** Front-end kısmında kullanılan bir framework'tir. Tek sayfalı uygulamaları geliştirmeyi sağlar. En çok kullanılan Javascript front-end framework'tir.
- (iv) **NodeJS:** Sunucu olarak kullanılıyor. Node.js Javascript back-end kısmında çalıştırılmasını sağlar.

Bu teknolojiler, MongoDB hariç, Javascript kullandığı için, sadece bir dili kullanarak full-stack uygulamaları geliştirebiliyoruz. Ayrıca, **Elektron** gibi teknolojileri kullanarak masaüstü uygulamalarını da geliştirebiliyoruz.

6.3 Geliştireceğimiz Uygulama

Express back-end kısmında kullanarak bir REST API geliştireceğiz. Bu REST API'den gelen verileri ReactJS ile alıp kullanıcılara göstereceğiz. Yani, bir RESTful servisi geliştireceğiz. Geliştireceğimiz REST API tüm HTTP metodlarını kullanacağız. Basit bir blog geliştireceğiz. Blog'un başlığı ve açıklaması eklenmesini sağlayacağız, sonra eklendiği verileri görüntülenmesi, güncellenmesi ve silinmesini sağlayacağız. Bu API için ReactJS kullanarak bir front-end geliştireceğiz kullanıcının arayüzü olsun diye. Daha detaylı ve anlaşılır bir şekilde anlatması için birkaç videoyu hazırladım. Aşağıdaki linkten ulaşabilirsiniz.

<https://drive.google.com/drive/folders/1adidU0h4RknGpnfnsMTxSvbQkkAWTC CJ?usp=sharing>

6.3.1 REST Nedir?

REST(Representational State Transfer), dağıtık sistemler tasarlamak için kullanılan bir mimari tarzıdır. REST, client-server arasındaki haberleşmeyi sağlayan HTTP protokolü üzerinden çalışan bir mimaridir. REST, servis yönelimli mimari üzerine oluşturulan yazılımlarda kullanılan bir transfer yöntemidir. İstemci ve sunucu arasında XML ve JSON verilerini taşıyarak uygulamanın haberleşmesini sağlar. REST mimarisini kullanan servislere ise RESTful servis denir.

6.3.2 HTTP Metodları

HTTP Metodları **POST, GET, PUT, DELETE**'dir.

- (i) **POST:** Veri eklemek için kullanılır. Daha güvenli yöntemdir. **Route:** /api/posts
- (ii) **GET:** Veri listeleme, veri görüntülemek için kullanılır. Eklenmesi için de kullanılabilir ama sadece gizli olmayan verilerle kullanılması gerekiyor. Gizli veriler mesela parolalar GET ile gönderilmemesi gerekiyor. **Route:** /api/posts
- (iii) **PUT:** Veri güncelleme için kullanılır. **Route:** /api/posts/:id
- (iv) **DELETE:** Veri silmesi için kullanılır. **Route:** /api/posts/:id

6.4 Genel Kurulum

Geliştirme başlamadan önce Node.js indirip kurmamız gerekiyor. Ayrıca, MongoDB kullanmak için veritabanımızı hazırlayacağız.

6.4.1 MongoDB Kurulumu

MongoDB iki şekilde kurabiliyoruz, web tabanlı ya da yerel olabilir. Tabi ki, web tabanlı olan istediğimiz yerden verilerimiz alabiliyoruz ama yerel olan sadece makinemizden ulaşabiliyoruz. Bu seçenek size bırakıyorum, sizin için hangisi uygun onu kullanabilirsiniz. Web tabanlı olanı kullanacağım.

- (i) MongoDB web sayfasına gidiniz ve yeni hesabı oluşturunuz. <https://mongodb.com>
- (ii) Hesabı oluşturduktan sonra, yeni bir **Cluster** otomatik olarak oluşturulacak. **Cluster** bizim veritabanımız oluşturulacak bir yerdir.

- (iii) Soldaki menüde olan **Database Access** seçeneği tıklayınız. Çıkacak sayfada olan **Add New Database User** butonu tıklayarak yeni bir veritabanının kullanıcıyı ve şifresini oluşturunuz. Bu kullanıcı ismini ve şifresini kullanarak veritabanımıza uygulamamızdan ulaşabiliyoruz.
- (iv) Kullanıcıyı oluşturduktan sonra aynı menüdeki **Network Access** olan seçeneği tıklayınız. **Add IP Adress** butonu tıklayarak yeni bir IP adresi ekleyiniz. Sizin IP adresinizden başka bu veritabanıyı kullanmasını istemiyorsanız sizin IP adresi giriniz yoksa **ALLOW ACCESS FROM ANYWHERE** butonu tıklayınız.

Videoyu izleyerek kurmak istiyorsanız hazırladığım video aşağıdaki linkten ulaşabilirsiniz.

<https://drive.google.com/drive/folders/1adidU0h4RknGpnfnsmTxSvbQkkAWTC CJ?usp=sharing>

6.4.2 Node.js Kurulumu

Node.js hem ReactJS hem de Express uygulamamızı geliştirmek için kullanacağız.

- (i) Node.js web sayfasına gidiniz. <https://nodejs.org>
- (ii) LTS olan versiyonu indirip kurunuz

6.5 Uygulama Geliştirme

Hem back-end hem de front-end kısımlarını geliştireceğiz. Back-end sunucu tarafta olan kısımdır ve front-end kullanıcı arayüzü kısımdır, yani kullanıcı görüp kullanacak kısımdır. Back-end kısım Express ve MongoDB ile geliştireceğiz ve front-end kısım yazdığımız back-end API'den veriyi alarak kullanıcıya göstereceğiz.

6.5.1 Back-End Kısım

Express ile geliştireceğiz. Oluşturduğumuz MongoDB'nin kullanıcı ismini ve şifresini kullanarak Express ve MongoDB arasında bir bağlantı kuracağız. Bunu yaparak gireceğimiz veriler veritabanında kalacak istediğimiz uygulamadan alabiliriz. Bu kısmın video linki:

<https://drive.google.com/drive/folders/1adidU0h4RknGpnfnsmTxSvbQkkAWTC CJ?usp=sharing>

Başlatma

Yeni bir klasörü oluşturduktan sonra, o klasörün içinde Node.js uygulama başlatacağız. Node.js projeyi başlattığımızda bir `package.json` dosya oluşturulacak. Bu dosya içinde kullanacağımız kutuphanelerin ismi ve versiyonları saklanacaktır. İki tane temel kutuphaneyi kullanacağız, **mongoose** ve **express**, çevre değişkenlerimiz yönetmek için **dotenv** kutuphaneyi kullanacağız. Node.js projeyi başlatmak için bu klasör içinde bir terminal açıp ve `npm init` yazacağız. Birkaç sorular olacak, projeye ilgili sorulara cevap verdikten sonra `package.json` dosyayı oluşturulacak. Eğer bu soruları cevaplandırmak istemiyorsanız `npm init` yerine `npm init -y` yazıp entera basabilirsiniz.

Kutuphaneler Kurma

`mongoose` ve `express` kutuphanemiz kuralım. Bu kutuphaneler kurmak için `npm install mongoose express dotenv` yazıp entera basınız. Ondan sonra uygulamamızda kullanacağımız kutuphaneler kurulmuş olacaktır.

```
{
  "name": "back-end",
  "version": "1.0.0",
  "description": "Server side code for yazsum application",
  "main": "index.js",
  "scripts": {
    "dev": "nodemon index",
    "start": "node index"
  },
  "author": "Margai Wangara",
  "license": "MIT"
}
```

Şekil 6.1: İlk package.json dosyamız.

Detaylı bir şekilde anlatmak için bu kısmın videosunu çekip yukarıdaki linkte koydum. Aşağıdaki linkten projenin kodu ulaşabilirsiniz.

<https://github.com/margaiwangara/yazsum>

Klasör Yapısı

MVC(Model-View-Controller) yapısını kullandım.

```
back-end/
- config/
- controllers/
- middleware/
- models/
- routes/
- index.js
- package-lock.json
- package.json
```

Şekil 6.2: Klasör Yapısı.

Index.js Dosyası

Index.js dosya içinde sunucumuz başlatıyoruz express kullanarak. MongoDB'nin dosyası da buradan alıyoruz.

```
const express = require('express');
const dotenv = require('dotenv');
const connectDB = require('./models');

const app = express();

// dotenv config
dotenv.config({ path: './config/config.env' });

app.use(express.json());

// connectDB
connectDB();

// routers
const postRoutes = require('./routes/posts');
app.use('/api/posts', postRoutes);

const errorHandler = require('./middleware/errors');
app.use(function (req, res, next) {
  const error = new Error();
  error.message = 'Not Found';
  error.status = 404;
  next(error);
});
app.use(errorHandler);

// port
const PORT = process.env.PORT || 5000;

app.listen(PORT, () => console.log(`App running on port ${PORT}`));
```

Şekil 6.3: index.js Dosyası.

Config Klasör

Config klasör içinde uygulamanın çevre değişkenleri saklanılıyor. Bu değişkenler uygulamanın geliştiricisinden başka görünmemesi için kullanılır.

```
NODE_ENV=development
MONGO_URI=mongodb+srv://margaiwangara:<Password>@cluster0.voae2.mongodb.net/yazsum?
retryWrites=true&w=majority
```

Şekil 6.4: config.env Dosyası.

Models Klasör

Models klasör içinde iki tane dosya var, index.js ve Post.js dosyalardır, index dosya MongoDB ayarlamalarını yapmak için kullanılıyor. MongoDB'nin veritabanısının kullanıcı ismi ve şifresi belirlenir, post.js dosyada ise bizim tablomuz yapısını belirlenir.

Controllers Klasör

Controllers klasör içinde posts.js dosya bulunmaktadır. Burada controller metotlar var. Bu metotlar model'den gelen verilere işlem yapılmasını sağlar.

Middleware Klasör

Middleware klasör içinde errors.js dosya bulunmaktadır. Errors.js hata yönetimi için kullanılır.

```
const mongoose = require('mongoose');

async function connectDB() {
  try {
    const conn = await mongoose.connect(process.env.MONGO_URI, {
      useNewUrlParser: true,
      useFindAndModify: false,
      useUnifiedTopology: true,
      useCreateIndex: true,
      keepAlive: true,
    });
    console.log(`MongoDB Connected: ${conn.connection.host}`);
  } catch (error) {
    console.log(error);
  }
}

module.exports = connectDB;
```

Şekil 6.5: Model index.js Dosyası.

```
const mongoose = require('mongoose');

const postSchema = new mongoose.Schema({
  title: {
    type: String,
    trim: true,
    unique: true,
    minlength: [10, 'Title length must be greater than 10'],
    maxlength: [200, 'Title length must be less than 200'],
    required: [true, 'Title is required'],
  },
  description: {
    type: String,
    trim: true,
    required: [true, 'Description is required'],
  },
  dateCreated: {
    type: Date,
    default: Date.now,
  },
});

const Post = mongoose.model('Post', postSchema, 'posts');

module.exports = Post;
```

Şekil 6.6: Model post.js Dosyası.

```
const Post = require('../models/Post');

// get all posts
exports.getPosts = async (req, res, next) => {
  try {
    const posts = await Post.find();

    return res.status(200).json({
      success: true,
      data: posts,
    });
  } catch (error) {
    next(error);
    console.log(error);
  }
};

// get single post
// /api/posts/:id
exports.getPost = async (req, res, next) => {
  try {
    const post = await Post.findById(req.params.id);

    return res.status(200).json({
      success: true,
      data: post,
    });
  } catch (error) {
    next(error);
    console.log(error); // hata göstermek için
  }
};

// create post
exports.createPost = async (req, res, next) => {
  try {
    const newPost = await Post.create(req.body);

    return res.status(201).json({
      success: true,
      data: newPost,
    });
  } catch (error) {
    next(error);
    console.log(error);
  }
};

// update post
// api/posts/:id PUT
exports.updatePost = async (req, res, next) => {
  try {
    const updatedPost = await Post.findByIdAndUpdate(req.params.id, req.body, {
      new: true,
      runValidators: false,
    });

    return res.status(200).json({
      success: true,
      data: updatedPost,
    });
  } catch (error) {
    next(error);
    console.log(error);
  }
};

// delete post
// api/posts/:id
exports.deletePost = async (req, res, next) => {
  try {
    await Post.findByIdAndDelete(req.params.id);

    return res.status(200).json({
      success: true,
    });
  } catch (error) {
    next(error);
    console.log(error);
  }
};
};
```

Şekil 6.7: posts.js Dosyası.

```
function errorHandler(error, req, res, next) {
  return res.status(error.status || 500).json({
    error: {
      message: error.message || 'Oops! Something went wrong',
    },
  });
}

module.exports = errorHandler;
```

Şekil 6.8: errors.js Dosyası.

Routes Klasör

Routes klasör içinde posts.js dosya bulunmaktadır. Routes ayarları burada yapılmaktadır.

```
const express = require('express');
const router = express.Router();

const {
  createPost,
  getPosts,
  getPost,
  deletePost,
  updatePost,
} = require('../controllers/posts');

// /api/posts/:id /api/posts/
router.route('/').post(createPost).get(getPosts);
router.route('/:id').get(getPost).put(updatePost).delete(deletePost);

module.exports = router;
```

Şekil 6.9: posts.js Dosyası.

Proje Linkleri

Bu projenin Github ve video linkleri aşağıda bulabilirsiniz. Yazılan kodu daha iyi anlamak için yaptığım videolara bakmanızı öneririm.

Videolar: <https://drive.google.com/drive/folders/1adidU0h4RknGpnfnsmTxSvbQkkAWTCCJ?usp=sharing>

Github: <https://github.com/margaiwangara/yazsum>

6.5.2 Front-End Kısmı - ReactJS

Arayüz kısmımız ReactJS ile yazacağız. Kullanıcı arayüzü geliştirmek için kullanılır. ReactJS ve React Native mobil uygulamaları geliştirmeyi kullanılır.

ReactJS Nedir?

- Facebook tarafından geliştirilmiş bir front-end framework'tür.
- En çok kullanılan bir Javascript front-end framework.
- Tek sayfalı(Single Page) uygulamaları geliştirmek için kullanılır.

ReactJS Avantajları

- Kullanması daha kolay - uygulamaları geliştirmek için daha kolay
- Component'leri kullandığı için daha düzenli uygulamaları geliştirilebilir
- JSX kullanarak HTML kodu Javascript içinde yazabiliyoruz, ayrıca CSS kodu da kullanabiliyoruz
- Virtual DOM kullandığı için tüm sayfayı yenilenmesi gerek kalmıyor, sadece değiştirilmiş kısmı değişiyor

React Öğrenmeden Önce

- Javascript temellerini bilinmesi gerekiyor(objects, arrays, conditionals, loops, variables vb.)
- ES2015+ temellerini bilinmesi gerekmiyor ama öğrenilmesini tavsiye ediyorum (arrow functions, destructuring, rest, spread, classes vb.)

React Temelleri

- Component - uygulamayı küçük parçalara ayrılmasını sağlar.
- State - component'in durumu
- Props - componentler'in arasında veri paylaşmasını sağlar.

React Projesi Oluşturma

Yeni bir React projesi oluşturmak için bir terminali açıp `npx create-react-app proje_ismi` yazmamız gerekir, **proje_ismi** yerine bizim projemizin ismi yazarız.

React State

- Componentlerinde değer saklamak için state'ten yararlanılır
- Uygulamada değişikliği yapmak için state kullanılır
- Bir buton tıkladıktan sonra herhangi bir güncelleme varsa state ile hallederiz
- State direkt güncellenmemesi lazım, **setState** kullanılması gerekiyor
- State iki şekilde tanımlayabiliyoruz, class kullanıyorsak, direkt tanımlayabiliyoruz ama functional component ise, hooks kullanmamız gerekiyor

React Props

- Props component arasında veri paylaşmak için kullanılıyor
- Üstteki component'ten altındaki olan component'e veri paylaşılabilir, state, props olarak paylaşılabilir ama büyük uygulamalarda çok component olduğu için zor olur

Hooks Nedir?

React Hooks functional component state tanımlamak için kullanılıyor. Her hook **use** ile başlıyor, mesela `useState`, `useReducer`, `useContext` vb. `useState` hook state yönetimi içindir. Özel hooks oluşturabiliyor, **use** kelimeyi kullanarak yeni bir hook oluşturabiliyoruz. Hooks sadece functional component ve başka hook fonksiyonu içinde kullanılabilir.

Context Nedir?

Context, state yönetimi için kullanılır. Büyük uygulamalarda, state, props olarak geçirmeden tanımladığımız state kullanılabilir.

Klasör Yapısı

Default React klasör yapısını kullandım.

```
- src/  
  - assets/  
  - components/  
  - context/  
    - actions/  
    - app/  
    - reducers/  
    - actionTypes.js  
    - initialState.js  
  - views/  
  - App.js  
  - index.css  
  - index.js
```

Şekil 6.10: Klasör Yapısı.

Context Tanımlama

Context global state yönetimi için kullanılır. Context için üç tane kısımımız var, `reducers`, `actions` ve `store`. Daha iyi anlaşılması için kodunu ekledim.

```
export const GET_POSTS = 'GET_POSTS';  
export const GET_POST = 'GET_POST';  
export const UPDATE_POST = 'UPDATE_POST';  
export const DELETE_POST = 'DELETE_POST';  
export const CREATE_POST = 'CREATE_POST';
```

Şekil 6.11: Action Types

Form Component

Form Component hem güncelleme hem de ekleme işlemleri için kullanılıyor.

App Component

App componentimiz içinde Route ve Context Provider tanımlıyoruz.

Home Component

Home component anasayfamızdır

EditPost Component

EditPost Component post güncellemek içindir

```
export const POST_STATE = {  
  post: {},  
  posts: [],  
  loading: false,  
};
```

Şekil 6.12: Initial State - Default State

```
import React, { useContext, createContext, useReducer } from 'react';  
import { POST_STATE } from '../initialState';  
import postReducer from '../reducers/posts';  
  
export const PostContext = createContext(null);  
export const PostConsumer = PostContext.Consumer;  
  
const PostProvider = ({ children }) => {  
  const [state, dispatch] = useReducer(postReducer, POST_STATE);  
  return (  
    <PostContext.Provider value={{ state, dispatch }}>  
      {children}  
    </PostContext.Provider>  
  );  
};  
  
export const usePost = () => useContext(PostContext);  
export default PostProvider;
```

Şekil 6.13: Post Context. Context böyle tanımlanır

```
import {
  CREATE_POST,
  DELETE_POST,
  GET_POSTS,
  UPDATE_POST,
  GET_POST,
} from '../actionTypes';

function postReducer(state, action) {
  switch (action.type) {
    case GET_POSTS:
      return {
        ...state,
        posts: action.posts,
      };
    case GET_POST:
      return {
        ...state,
        post: action.post,
      };
    case CREATE_POST:
      return {
        ...state,
        posts: [...state.posts, action.post],
      };
    case UPDATE_POST:
      // filter first
      const filteredPost = state.posts.filter(
        (value) => value._id !== action.post._id,
      );
      return {
        ...state,
        post: action.post,
        posts: [...filteredPost, action.post],
      };
    case DELETE_POST:
      return {
        ...state,
        posts: state.posts.filter((value) => value._id !== action.id),
      };
    default:
      return state;
  }
}

export default postReducer;
```

Şekil 6.14: Post Reducer. Reducer böyle tanımlanır

```
import axios from 'axios';
import {
  CREATE_POST,
  DELETE_POST,
  GET_POST,
  GET_POSTS,
  UPDATE_POST,
} from '../actionTypes';

export const BASE_URL = 'http://localhost:5000';

export const getPostsAction = (posts) => ({
  type: GET_POSTS,
  posts,
});

export const createPostAction = (post) => ({
  type: CREATE_POST,
  post,
});

export const updatePostAction = (post) => ({
  type: UPDATE_POST,
  post,
});

export const deletePostAction = (id) => ({
  type: DELETE_POST,
  id,
});

export const getPostAction = (post) => ({
  type: GET_POST,
  post,
});

function getPosts(dispatch) {
  return new Promise((resolve, reject) => {
    return axios
      .get(`${BASE_URL}/api/posts`)
      .then((res) => {
        dispatch(getPostsAction(res.data.data));
        resolve();
      })
      .catch((error) => {
        console.log(error);
        reject(error.response.data.error);
      });
  });
}
```

Şekil 6.15: Post Action. Post Actions böyle tanımlanır

```
function getPost(dispatch, id) {
  return new Promise((resolve, reject) => {
    return axios
      .get(`${BASE_URL}/api/posts/${id}`)
      .then((resp) => {
        dispatch(getPostAction(resp.data.data));
        resolve();
      })
      .catch((error) => {
        console.log('Get Post Error', error);
        reject(error.response.data.error);
      });
  });
}

function createPost(dispatch, formData) {
  return new Promise((resolve, reject) => {
    return axios
      .post(`${BASE_URL}/api/posts`, formData)
      .then((resp) => {
        dispatch(createPostAction(resp.data.data));
        resolve();
      })
      .catch((error) => {
        console.log('Creation Error', error);
        reject(error.response.data.error);
      });
  });
}

function updatePost(dispatch, formData, id) {
  return new Promise((resolve, reject) => {
    return axios
      .put(`${BASE_URL}/api/posts/${id}`, formData)
      .then((resp) => {
        dispatch(updatePostAction(resp.data.data));
        resolve();
      })
      .catch((error) => {
        console.log('Update Error', error);
        reject(error.response.data.error);
      });
  });
}

function deletePost(dispatch, id) {
  return new Promise((resolve, reject) => {
    return axios
      .delete(`${BASE_URL}/api/posts/${id}`)
      .then((resp) => {
        dispatch(deletePostAction(id));
        resolve();
      })
      .catch((error) => {
        console.log('Delete Error', error);
        reject(error.response.data.error);
      });
  });
}

export { getPosts, createPost, updatePost, deletePost, getPost };
```

Şekil 6.16: Post Action. Post Actions böyle tanımlanır

```
import React, { useState, useEffect } from 'react';
import { usePost } from '../context/app/PostContext';
import { createPost, updatePost, getPost } from '../context/actions/posts';
import { useParams, useHistory } from 'react-router-dom';

function PostForm({ page, pageTitle, btnText }) {
  const { state, dispatch } = usePost();
  const { id } = useParams();
  const history = useHistory();
  const [value, setValue] = useState({
    title: '',
    description: '',
  });
  const [errors, setErrors] = useState([]);
  const [isLoading, setIsLoading] = useState(false);

  useEffect(() => {
    if (page === 'edit') {
      setIsLoading(true);
      getPost(dispatch, id)
        .then(() => {
          setIsLoading(false);
          setValue({
            title: state.post.title ? state.post.title : '',
            description: state.post.description ? state.post.description : '',
          });
          console.log('Post Acquired');
        })
        .catch(() => console.log('Post Not Acquired'));
    }
  }, [state.post.title, state.post.description]);

  const handleChange = (e) =>
    setValue({ ...value, [e.target.name]: e.target.value });
  const handleSubmit = (e) => {
    e.preventDefault();

    if (page === 'create') {
      createPost(dispatch, value)
        .then(() => {
          console.log('Post Created');
          history.push('/');
        })
        .catch((error) => {
          setErrors(error.message);
          console.log('Post Not Created');
        });
    } else {
      updatePost(dispatch, value, id)
        .then(() => {
          console.log('Post Updated');
          history.push('/');
        })
        .catch((error) => {
          setErrors(error.message);
          console.log('Post Not Updated');
        });
    }
  };
};
```

Şekil 6.17: Form Component

```
return (
  <div className="row">
    <div className="col-md-4 offset-md-4">
      <div className="card">
        <div className="card-body">
          <div className="card-body">
            {isLoading ? (
              <p>Loading...</p>
            ) : (
              <form onSubmit={handleSubmit}>
                <h3>{pageTitle}</h3>
                <hr />
                {errors.length > 0 ? (
                  <div className="alert alert-danger">{errors}</div>
                ) : (
                  ..
                )}
              </form>
            )}
          </div>
          <div className="form-group">
            <label htmlFor="titleField">Title</label>
            <input
              type="text"
              className="form-control"
              name="title"
              id="titleField"
              onChange={handleChange}
              value={value.title}
            />
          </div>
          <div className="form-group">
            <label htmlFor="descriptionField">Description</label>
            <textarea
              name="description"
              className="form-control"
              rows="10"
              id="descriptionField"
              onChange={handleChange}
              value={value.description}
            />
          </div>
          <button type="submit" className="btn btn-primary btn-block">
            {btnText}
          </button>
        </div>
      </div>
    </div>
  </div>
);
export default PostForm;
```

Şekil 6.18: Form Component


```
import React from 'react';
import { BrowserRouter as Router, Route, Switch, Link } from 'react-router-dom';
import PostProvider from './context/app/PostContext';
import Home from './views/Home';
import EditPost from './views/EditPost';
import CreatePost from './views/CreatePost';

// localhost:3000/ /posts /posts/:id
function App() {
  return (
    <Router>
      <PostProvider>
        <div className="navbar navbar-expand-md navbar-dark bg-primary">
          <div className="container">
            <Link to="/" className="navbar-brand">
              yazsum
            </Link>

            <ul className="navbar-nav ml-auto">
              <li className="nav-item">
                <Link to="/posts" className="btn btn-primary">
                  Create Post
                </Link>
              </li>
            </ul>
          </div>
        </div>
        <div className="container my-3">
          <Switch>
            <Route
              exact
              name="Home"
              render={(props) => <Home {...props} />}
              path="/"
            />
            <Route
              exact
              name="Create Post"
              render={(props) => <CreatePost {...props} />}
              path="/posts"
            />
            <Route
              exact
              name="Edit Post"
              render={(props) => <EditPost {...props} />}
              path="/posts/:id"
            />
          </Switch>
        </div>
      </PostProvider>
    </Router>
  );
}

export default App;
```

Şekil 6.19: App Component

```
import React, { useEffect } from 'react';
import { Link } from 'react-router-dom';
import { usePost } from '../context/app/PostContext';
import { getPosts, deletePost } from '../context/actions/posts';

function Home() {
  const { state, dispatch } = usePost();

  const removePost = (id) => {
    deletePost(dispatch, id)
      .then(() => console.log('Post Deleted'))
      .catch(() => console.log('Post Not Deleted'));
  };

  useEffect(() => {
    getPosts(dispatch)
      .then(() => console.log('Posts Acquired'))
      .catch(() => console.log('Posts Not Acquired'));
  }, []);

  return (
    <div className="table-responsive">
      <table className="table table-bordered">
        <thead>
          <tr>
            <th>Id</th>
            <th>Name</th>
            <th>Description</th>
            <th>Actions</th>
          </tr>
        </thead>
        <tbody>
          {state.posts.length > 0 ? (
            state.posts.map((value) => (
              <tr key={value._id}>
                <td>{value._id}</td>
                <td>{value.title}</td>
                <td>{value.description}</td>
                <td>
                  <Link
                    to={`/posts/${value._id}`}
                    className="btn btn-sm btn-success mr-2"
                  >
                    Edit
                  </Link>
                  <button
                    className="btn btn-danger btn-sm"
                    onClick={() => removePost(value._id)}
                  >
                    Delete
                  </button>
                </td>
              </tr>
            ))
          ) : (
            <tr>
              <td colspan="4">
                <p className="text-info m-0">No posts available.</p>
              </td>
            </tr>
          )
        </tbody>
      </table>
    </div>
  );
}

export default Home;
```

Şekil 6.20: Home Component

```
import React from 'react';
import PostForm from '../components/PostForm';

function EditPost() {
  return <PostForm page="edit" pageTitle="Edit Post" btnText="Edit" />;
}

export default EditPost;
```

Şekil 6.21: EditPost Component

Create Post Component

CreatePost Component post eklemek içindir.

```
import React from 'react';
import PostForm from '../components/PostForm';

function CreatePost() {
  return <PostForm page="create" pageTitle="Create Post" btnText="Save" />;
}

export default CreatePost;
```

Şekil 6.22: CreatePost Component

Yazarlar Hakkında



A.K. Margai Wangara, 21 Nisan 1996 yılında Mumias, Kenya’da doğmuştur. 2014 yılında liseden mezun olduktan sonra, Kenya’nın başkenti Nairobi’de bulunan Emobilis Mobil Programlama okulunda kodlama seyahatına başlamış, aynı yıl Technical University of Kenya Bilişim Teknoloji Bölümü’nde lisans öğrenimine başlamıştır. 2015’te kendini geliştirmek amacıyla Kenya’dan Türkiye’ye gelmiş. Sakarya Üniversitesinde Bilişim Sistemleri Mühendisliği bölümünde lisans öğrenimine başlamıştır. 2015’ten beri web ve mobil programlama temel alanı olarak çalışmaktadır.



Ussene Aliasse Jamal, 7 Ekim 1996 yılında Mozambik’te doğmuştur. Çocukluğundan beri teknolojiye hayranmış. Lisedeyken programlamaya başlamış. 2014’te Türkiye’ye eğitim amacıyla gelmiş ve aynı yılda Sakarya Üniversitesinde Bilişim Sistemleri Mühendisliği bölümünde lisans öğrenimine başlamıştır. Her zaman kendini geliştirmeye çalıştığı için çok şeyi öğrenmiş. Şimdiye kadar OOP C++ ve Python temel alanı olarak çalışmaktadır.



Doğal Dil İşleme Çalışmalarında Yeni Yaklaşımlar

Harun AKSAYA¹, Sevinç GÜLSEÇEN²

¹Phdc., İstanbul Üniversitesi, Fen Bilimleri Enstitüsü, Enformatik Anabilim Dalı, İstanbul, Türkiye E-mail: harunaksaya@marmara.edu.tr

²Prof. Dr., İstanbul Üniversitesi, Fen Bilimleri Enstitüsü, Enformatik Anabilim Dalı, İstanbul, Türkiye E-mail: gulsecen@istanbul.edu.tr

7.1 Giriş

Yapay zeka çalışmalarının en önemli alt kollarından biri de Doğal Dil İşlemedir (DDİ). İnsan zekası doğal dili kullanarak kendini ifade edebiliyor ya da başkalarının ne söylediğini rahat bir şekilde anlayabiliyor. Yani kısaca iletişim kurabiliyor. İnsana özgü iletişim kurma becerilerini bilgisayar sistemlerine kazandırabilmek için ise DDİ yöntem ve metotlarından faydalanılmaktadır. Web siteleri, sosyal medya platformları, forum vb. web teknolojileri ile birlikte çok büyük miktarda metinsel veriler üretilmektedir. Yapısal olmayan bu verilerin işlenip anlamlı hale getirilebilmesi için ise insan üstü güce ihtiyaç bulunmaktadır. Bu noktada DDİ yöntem ve metodları ile geliştirilen yapay zeka algoritmaları çok büyük verileri isteğimiz doğrultusunda sınıflayabiliyor, özetini çıkarabiliyor ya da farklı bir dile saniyeler içerisinde çevirebiliyor. Çağrı merkezlerinde insan gücüne dayalı yapılan çağrı hizmetleri DDİ yöntem ve metodları sayesinde geliştirilen akıllı chatbotlar ile insan gücü olmadan rahatlıkla yapılabilmektedir. Telefonlarımızın içerisine giren akıllı asistanlar sayesinde istediğimiz bilgiye telefonumuza dokunmadan ulaşabiliyor, sesli komutlar ile yemek siparişi bile verebiliyoruz. Bu ve benzeri gelişmeler DDİ disiplinine olan ilgiyi her geçen gün arttırmakta ve bu disiplini daha popüler bir hale getirmektedir (Hark vd., 2017).

7.2 Doğal Dil İşleme Nedir?

En basit tanımıyla DDİ insanlara özgü dillerin bilgisayar sistemleri içerisinde işlenerek kullanılmasını amaç edinen bir yapay zeka alt disiplini. Dillerin matematiksel modellerini oluşturarak bilgisayarlar tarafından anlaşılmasını sağlar ve yeniden üretilmesi için çözümler üretir.

Yine başka bir tanıma göre DDİ, doğal dilleri matematiksel olarak temsil ve analiz etmek için kullanılan bir disiplindir (Khurana vd., 2017). Ayrıca literatürde DDİ için istatistiksel dil modelleme olarak tanımlamalar da bulunmaktadır (Ofłazer, 2016). Bununla birlikte DDİ, yapay zeka, dilbilim ve bilgisayar destekli dilbilim gibi çok farklı disiplinlerdeki kuram, yöntem ve teknolojileri bir araya getiren bir bilim dalıdır (Delibaş, 2008).

İnternet ve web teknolojilerinin gelişmesi ile birlikte ürettiğimiz metinsel verilerde dramatik bir artış gözlemlenmektedir. Bu verilerin anlamlı hale getirilmesi endüstriler ve pazarlama sektörü için çok önemlidir. Bununla birlikte yapılandırılmamış doküman, belge vb. arşiv yığınları da anlamlandırılmaya muhtaç durumdadır. Bu durum Doğal Dil İşleme çalışmalarına olan ilgiliyi her geçen gün artırmaktadır (Hark vd., 2017).

Adalı (2012)'ya göre DDİ kapsamında şu çalışmalar yapılmaktadır:

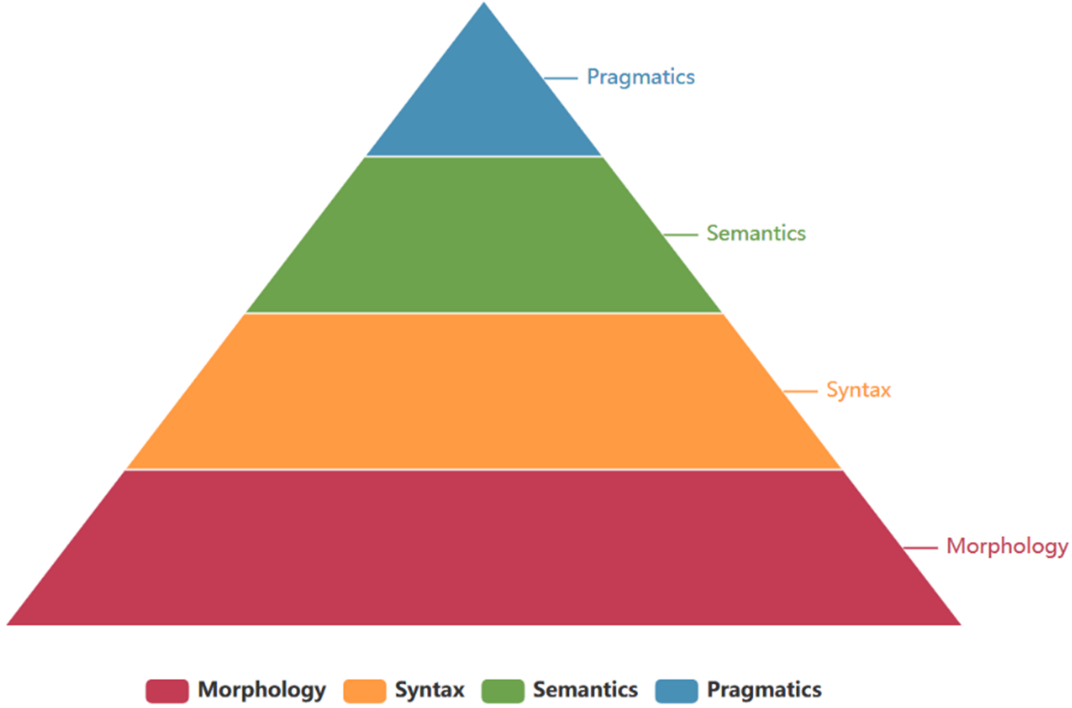
- Basılı bir metni okuma (optik olarak metin okuma) ve okuma yanlışlarını düzeltme
- Metinden özet çıkarma
- Metinden bilgi çıkarma
- Bilgiye erişim sağlama
- Bir metni anlama
- Bilgisayarla sesli etkileşim
- Bir metni seslendirme
- Ses dalgalarını metne dönüştürme
- Soru yanıt dizgeleri (Chatbotlar)
- Okuma yardımcı araçları
- Yazım yardımcı araçları
- Diller arası çeviri

Doğal Dil İşleme Piramidi

Towards Data Science (2017)' a göre doğal dil işleme çalışmaları 4 fazlı bir piramitten oluşmaktadır. Bunlar sırasıyla:

1. Morphology - Biçim Bilimsel Analiz Çalışmaları
 - (a) Yazım ve imla hataları
 - (b) Cümle içindeki kelimelerin tespiti
 - (c) Kelimelerin türleri ve aldığı ekler
2. Syntax - Söz Dizimsel Analiz Çalışmaları
 - (a) Cümle kurulum yapısı
 - (b) Gramer yapısı
3. Semantic - Anlamsal Analiz Çalışmaları
 - (a) Varlık tanıma (Şehir, ülke, özel isim)
 - (b) Kelimeler arası ilişkilerin tespit edilmesi
4. Pragmatics – Pragmatik Analiz Çalışmaları
 - (a) Metinden konu tespiti
 - (b) Özet çıkarma
 - (c) Yazar tespit etme
 - (d) Sentimental analiz

- (e) Soru cevap sistemleri
- (f) Çeviri sistemleri



Şekil 7.1: Doğal Dil İşleme Piramidi (Toward Data Science, 2017).

7.3 Sentimental Analiz Nedir?

Sentimental (duygu) analiz ifadesi literatüre 2003 yılında “Sentiment Analysis: Capturing Favorability Using Natural Language Processing” çalışması ile girmiştir (Tetsuya ve Jeonghee, 2003).

Sentimental analiz, metinsel içerikler içerisinde yer alan markalara, kişilere, ürün ya da bir organizasyona ilişkin duygusal durum bilgilerinin DDİ yöntem ve yaklaşımlarını kullanarak ortaya çıkarılması işidir (Fessini vd., 2014). İnternet ve sosyal mecraların gelişmesi ile birlikte iletişim biçimimizde de değişiklikler oldu. Sosyal medya paylaşımlarımız, mesajlaşma uygulamalarındaki mesaj verilerimiz ile büyük metinsel veriler oluşturmuş durumdayız. Bu metinsel verilerin işlenmesi ve anlamlı bilgiler çıkarılması çok önem arz etmektedir.

Web, yapısal olmayan metinsel verilerden oluşan büyük bir ambar durumundadır. Sosyal mecralar, forumlar, kullanıcı görüşlerini barındıran siteler içerisinde duygu ve görüş barındırmaktadır. Sentimental analiz bu verileri kullanarak anlamlı bilgiler ortaya çıkarabilir. Bu bilgilerde yöneticilerin ya da bireylerin karar verme süreçlerine destek olabilir (Fessini vd., 2014).

7.4 Sentimental Analiz Çalışmalarında Kullanılan Seviyeler

Sentimental analiz çalışmaları, çalışmanın kapsamı ve veri setlerinin yapısı doğrultusunda üç farklı çalışma seviyesinde ele alınmaktadır. Bunlar sırasıyla doküman, cümle ve aspect (özellik, kategori) seviyesinde yapılan çalışmalardır (Özyurt ve Akçayol, 2018).

7.4.1 Doküman Seviyesinde Yapılan Çalışmalar

Bu seviyede yapılan çalışmalarda bir makale ya da birden fazla cümleden oluşan kullanıcı görüşünün tamamı bir bütün olarak değerlendirilir. Duygu durumu cümle ya da özellik bazında detaya inilmeden doküman seviyesinde pozitif, negatif ya da nötr olarak tespit edilir.

7.4.2 Cümle Seviyesinde Yapılan Çalışmalar

Bu seviyede yapılan çalışmalarda bir doküman içerisinde yer alan her bir cümlenin duygu analizi ayrı ayrı yapılır. Bu sayede doküman seviyesinde yaşanabilecek duygu durumu kayıpları engellenmiş olur. Doküman seviyesinde yapılan duygu durumu analizlerine göre daha detaylı ve hassas sonuçlar elde edilir.

7.4.3 Aspect Seviyesinde Yapılan Çalışmalar

Aspect tabanlı duygu analizi çalışmaları sözcük ve doküman seviyesinde yapılan çalışmalara göre çok daha yeni ve yakın döneme ait çalışmalardır (Özyurt ve Akçayol, 2018). Cümle seviyesinde yapılan duygu analizleri her ne kadar doküman seviyesinde yapılan analiz çalışmalarına göre daha hassas ve detaylı sonuçlar verse de bazı durumlarda bir cümlenin gerçek duygu durumunu yansıtamayabilirler. Örneğin bir cümle içerisinde hem pozitif hem de negatif duygu durumu yer alabilir. Aspect seviyesinde yapılan çalışmalar bu problemin giderilmesi için geliştirilmiştir.

Aspect tabanlı duygu analizi çalışmalarında önce cümle içerisinde geçen aspectler (özellik, kategori) belirlenir. Daha sonra bu aspectleri niteleyen kelimeler üzerinden sözlük tabanlı ya da makine öğrenmesi yaklaşımları kullanılarak her bir aspectin duygu durumuna karar verilir.

7.5 Sentimental Analiz Çalışmalarında Kullanılan Yöntem ve Yaklaşımlar

Sentimental analiz çalışmalarında kullanılan yöntemleri, makine öğrenmesine dayalı yöntemler ve sözlüğe dayalı yöntemler olmak üzere iki sınıfta toplayabiliriz (Medhat vd., 2014).

7.5.1 Sözlük Tabanlı Yaklaşım

Sözlük tabanlı yaklaşımda, bir metinde geçen duygu durumlarının tespiti için her bir kelimenin duygusal durumu, kullanılan sözlük yardımı ile hesaplanır (Taboada, 2011). Duygu durumunun tespit edilmesinde DDİ araç ve yöntemleri ile duygu terimleri sözlükleri kullanılır. Bu tür çalışmalarda herhangi bir etiketli veriye ya da eğitilmiş bir modele ihtiyaç yoktur (Özyurt ve Akçayol, 2018). Bununla birlikte kullanılan duygu sözlüğü ne kadar kapsamlı ise alınan sonuçlarda bir o kadar iyi olacaktır.

7.5.2 Makine Öğrenmesi - Yapay Sinir Ağları (YSA) Yaklaşımı

Makine öğrenmesine dayalı sentimental analiz çalışmalarında ise oluşturulan model etiketli bir eğitim verisi ile önceden eğitilmeli, test verisi ile de başarı oranı test edilmelidir (Özyurt ve Akçayol, 2018). Literatürdeki son çalışmalara göre sentimental analiz çalışmalarında makine öğrenmesinin alt disiplini olan derin öğrenme yöntemlerinin kullanılması başarı oranını ciddi oranda artırmıştır (Collobert vd., 2011). YSA yapısı büyük verilerin işlenmesi ve sınıflandırılmasında sözlük tabanlı yaklaşımlara göre çok daha hızlı ve başarılı sonuçlar ortaya koyduğu tespit edilmiştir (Song ve Lee, 2013). Ancak bu yaklaşımda geliştirilen yapay zeka modelinin daha iyi sonuçlar verebilmesi için çok miktarda etiketli veriye ihtiyaç vardır.

7.6 Aspect Tabanlı Sentimental Analiz Çalışmalarında Makine Öğrenmesi Tekniklerinin Kullanılması

Bu çalışma kapsamında YSA kullanılarak aspect tabanlı sentimental analiz çalışması yapılmıştır. Yapılan işlemler sırasıyla:

- Verilerin (yorum cümleleri) toplanması işlemi,
- Yorum cümlelerinin aspect bazlı bölünmesi işlemi,
- Tokenizer oluşturma işlemleri,
- Model oluşturma, modeli eğitme ve test etme işlemleri.

7.6.1 Veri Toplam Süreci

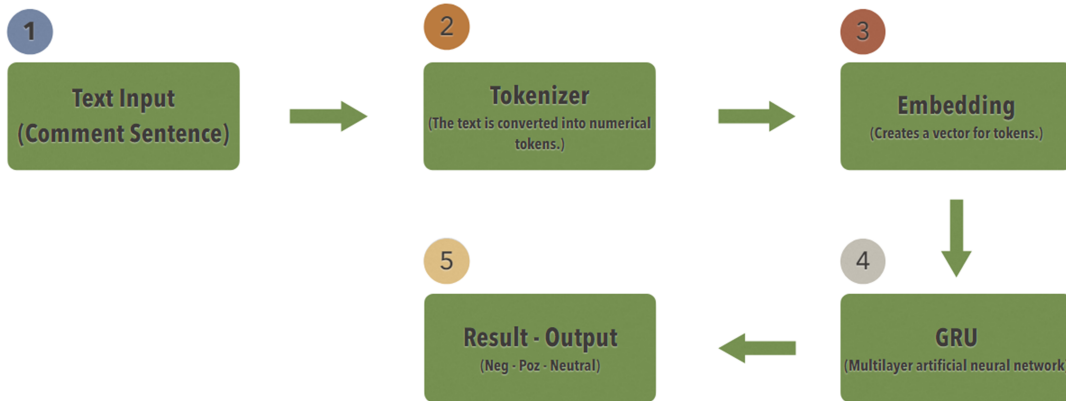
Okul.com.tr platformundan alınan veriler (1000 yorum) tek tek okunarak, yorumun içinde geçen aspectler belirlenmiştir. Bu aspectlerin duyu durumu pozitif ve negatif olarak etiketlenmiştir. Toplanan veriler bir veritabanına aktarılmıştır.

No	Yorum	Aspect
1	Okulumuz, harika bir eğitim vermesinin yanında çocuklarımıza her türlü olanağı sağlıyor. Çok başarılı bir okul.	Okul-p, Eğitim-p, Olanak-p
2	Öğrencilerin sağlığını göz önünde bulundurularak eğitim materyalleri kullanılmış olması çok büyük bir avantaj.	Eğitim Materyali-p
3	Montessori yönteminin ülkemizdeki amiral gemisi diyebiliriz. Çocuk odaklı ve klasik sistemin tamamen dışında alternatif bir okul.	Yöntem-p, Sistem-p, Okul-p
4	Çocuklarımızın fiziki, duygusal, zihinsel ve ruhsal açıdan kendilerini güvende hissedebilecekleri bir ortam imkanı sunmakta. Tebrikler.	Ortam-p
5	Ayrıcalıklı kurumlardan bir tanesi. Uyguladıkları eğitim modellerinin de geçerliliği tartışılmaz derecede sağlam.	Kurum-p, Eğitim Modeli-p
6	Tüm dünyanın örnek aldığı Kanada eğitim sisteminin yıkılmaz kalesi diyebiliriz burası için.	Eğitim Sistemi-p
7	Bu kadar yeşil alana sahip başka bir okul kampüsü bulamazsınız. Uluslararası eğitim fırsatlarını bu kadar geniş yelpazede sunan başka bir okul daha yok.	Kampüs-p, Eğitim Fırsatları-p
8	Anaokulundan Üniversiteye kadar tüm eğitimde sağladıkları başarılar ile sağlam adımlarla ilerlemekte.	Eğitim-p
9	Kısıtlı imkanlarla mucize yaratmayı bilen bir kadro, güler yüz ve sürekli yeni şeyler yapma isteğiyle öğrencileri de pesinden sürükleyen harika öğretmenler. Ancak binası biraz eski.	Kadro-p, Öğretmenler-p, Bina-n
10	konuyu pekiştirmeyi ve kalıcı olmasını sağlayan bir sistemi benimsemişler.	Sistem-p
11	Okulun eğitimi iyi olabilir ama binası bana çok eski geldi.	Eğitim-p, Bina-n

Şekil 7.2: Aspectlere göre etiketlenmiş kullanıcı yorumları

7.6.2 Model Oluşturma Süreci

Sistemin geliştirilmesinde python programlama dili ve JupyterLab geliştirme ortamı tercih edilmiştir. Modelin oluşturulmasında tensorflow ve keras kütüphaneleri kullanılmıştır.



Şekil 7.3: Modelin Yapısı

7.6.3 Aspectlere Bağlı Kelime Zincirlerinin Oluşturulma Süreci

Aspectlere bağlı kelime zincirlerinden yeni cümleler oluşturmak amacıyla bir script geliştirilmiştir. Script'in yapmış olduğu kontrollerden bazıları şunlardır:

- Aspect'in cümle içindeki konumunun tespit edilmesi.
- Aspect tek mi yoksa birden fazla kelimedenden mi oluşmakta?
- Aspect yorum içerisinde kaç gez geçiyor?
- Aspect Türkçe karakter içeriyor mu?

```
//İlgili aspect iki kelimedenden oluşuyorsa
if (count($baslangicAspectDizi)==2) {
  for ($i=0; $i < $dizi_boyutu; $i++) {
    //İngilizce Eğitimi ilk kelime probleminin giderilmesi
    $bul = array("İ");
    $degistir = array("i");
    $yorum_parsed[$i][1] = str_replace($bul, $degistir, $yorum_parsed[$i][1]);
    $yorum_parsed[$i][2] = str_replace($bul, $degistir, $yorum_parsed[$i][2]);
    if (strstr($yorum_parsed[$i][1],$baslangicAspectDizi[0]) or strstr($yorum_parsed[$i][2],$baslangicAspectDizi[0]) or (strstr($baslangicAspectDizi[1],$yorum_parsed[$i+1][2]) or (strstr($baslangicAspectDizi[1],$yorum_parsed[$i][1]))) {
      $bas_aspect = $i;
      $bit_aspect = $i+1;

      //İlişkili önceki kelimeleri tespi etme
      $baslangic = $bas_aspect;
      $bitis_noktasi = $baslangic-7;
      $onceki_kelimeler = array();
      while (($baslangic >= 0) and ($yorum_parsed[$baslangic][1]!=".") and ($baslangic >= $bitis_noktasi)) {
        $onceki_kelimeler[] = $baslangic;
        $baslangic--;
      }

      //İlişkili sonraki kelimeleri tespi etme
      $a = $bit_aspect;
      $c = $a+7;
      $sonraki_kelimeler = array();
      while (($a <= $c) and ($yorum_parsed[$a][1]!=".")) {
        $sonraki_kelimeler[] = $a;
        $a++;
      }
    }
  }
}
```

Şekil 7.4: Geliştirilen Scriptten Bir Kesit

Script 1000 yorum üzerinde çalıştırılmış, aspect zincirlerinden oluşan yeni yorumlar mysql veritabanına aktarılmıştır. Aspect tabanlı bölme işleminden sonra toplam yorum sayısı 1925'e çıkmıştır.

7.6.4 Tokenizer Oluşturma Süreci

Aspectlere bağlı kelime zincirlerinden oluşan tüm yorumları kullanarak bir tokenizer oluşturulmuştur. Tokenizer sonraki süreçlerde kullanmak için bir dosya olarak kaydedilmiştir. Bunun için keras içerisinde yer alan tokenizer kütüphanesi kullanılmıştır.

Tokenleştirme için ne kadar çok veri kullanılırsa, tokenizer ilgili domaini o kadar iyi temsil eder.

7.6.5 Model Oluşturma Süreci

Bu aşamada YSA kullanılarak model oluşturulmuştur. Bu aşamada şu işlemler yapılmıştır:

hanarak1_ghd2aspete_gore_yinjilmi_yorumlar:1.924 satır mevcut, ile sınırlı 1.000

agay_id	yorum_id	aspect	durum	yorum_metin	yorum_gesnel	yorum_ayrims	Başlı_Jelneler
15	9	Bina	n	Konkri inkanlarla mucize yaratımyı bten br kadro, gulleriy...	a:11:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Ankac bınas braz eskı
19	11	Bina	n	Okulun eđtımı iyı olablr ana bınas bana gk eskı geldı.	a:11:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Okulun eđtımı iyı olablr ana bınas bana gk eskı geldı
28	16	Üreter	n	Kurumsal br okul. Montessori sistemi başıyyla uıyulıy...	a:13:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Üreter braz yuksek
58	28	Bańıç	n	Sınıflar gk tabı. Öđrenmeye teıyke eden br ortam var...	a:21:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Bańıç ana braz kıkık geldı o kadar
65	31	Sınıfl Dersler	n	İbray derslernde başıy br okul ancak fıksele inkanlar ...	a:19:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	fıksele inkanlar da iyı olablr ana raglın sporıf derslerde ...
74	35	Servis Sıktesı	n	Cocugunun sevıđ ve kallelı eđretmenlerı yer aldđ yer...	a:19:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Servis sıktesı sevmıyorum
145	70	Öđretmen	n	Okul verđđ eđtım apandan bađtıńızda gayet iyı. ancak...	a:20:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	ancak eđretmenlerı lgısz
184	90	Okul	n	Ticarethane gblı çalıřan br okul.	a:7:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Ticarethane gblı çalıřan br okul
188	92	Sosyal	n	Kesınlkle tavsiye etmıyorum. Baęız ana br okul, hem eđtım...	a:15:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	br okul hem eđtım hem de sosyal apdan
191	93	Para	n	Kızı br okul dađı. Eđtımı iyı. Fıksele inkanlar da kızı de...	a:25:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Ankac verlen paraya halakıđđ konusunda gıphelıym
197	95	Öđretmen	n	Verlen paraya halaktesı ıçın en azından eđtım kaltesını b...	a:20:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Daha iyı eđretmenlerle çalıřabılır
199	96	Kampus	n	Eđtım bakımından iyı ve kallelı. Ancak kampus dđđer okul...	a:16:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Ankac kampus dđđer okula göre daha elden geđrıl...
218	104	Eđtım İçerđı	n	Fıksele donanımlar okur iyı ancak eđtımı ıçerđı ıçın ayımsa...	a:11:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Fıksele donanımlar okur iyı ancak eđtımı ıçerđı ıçın ayımsa...
227	108	Yemek	n	İmduye kadar verlen eđtımden ve eđretmenlerınden gay...	a:20:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Fıksele inkanlar da iyı ancak eđtım kaltesını gk memnun ...
231	109	Seıf	n	Yemek cepıllıđ az ve sııfır yetenı kadar tenz deđđ.	a:10:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Yemek cepıllıđ az ve sııfır yetenı kadar tenz deđđ
255	124	Revır	n	Gk yonırlı ve kallelı br eđtım var okulum. Rehberlek servı...	a:28:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Okulađı revır var ancak gırevırlı eskıllıđ var
262	127	Öđretmen	n	Fıksele inkanlar bakımından eskıklar br okul. Sınıflar temı...	a:29:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Daha tecrubelı eđretmenlerle çalıřalsar keıte
275	130	Hıyın	n	Eđtım kaltesı gayet iyı. Sosyal akıvıtelere gayet iyı. Hıyın...	a:17:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Hıyın bakımından gk iyı deđđı maalesef
278	131	Eđtım kaltesı	n	Eđtım kaltesı apandan gk başıyrlı bulmıyorum. Cocukl...	a:20:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Eđtım kaltesı apandan gk başıyrlı bulmıyorum
280	133	Öđretmen	n	Okulun inkanlarından gk memnunuz. Ders cepıllıđ ve Ö...	a:19:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Öđretmen seğınde daha deđıklı olınalır
284	134	İlgı	n	Fıksele inkanlarla bađlıđında iyı br okul gblı görünse de...	a:23:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Öđrenclere yeterince ilgılenıyır
297	141	Okul	n	Geređınden gk daha pahalı br okul.	a:7:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Geređınden gk daha pahalı br okul
303	144	Bına	n	Okulun bınas braz eskı. Onun dıńında br gılayetıme yok.	a:11:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Okulun bınas braz eskı
342	161	Para	n	Deđlet okullarından gk farkı yok. Bına kaltesı verđđına g...	a:12:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Bına kaltesı verđđına paraya deđmez
347	163	Eđtım	n	Sarıyev de neredeyse her okul en az bu dızeyde fıksele ...	a:26:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Fıksele eđtım alanında kendılenıme net br gıyıl olınelerı
351	165	Kurumsal	n	Kurumsal sıfır. Ale kurumu gblı yonıteılır.	a:9:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Kurumsal sıfır
352	166	Okul	n	İnanılmaz kızı br okul. Salın gımeıyın. Sadece para kayıbı.	a:12:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	İnanılmaz kızı br okul
389	183	Burs	n	Gk iyı dđ eđtımı vermıyolar. Okulađı sevıy var. Tavısyı ed...	a:24:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Verıdıklen burs oranları yuksekdebr
407	191	Okul	n	Verlen temetler gayet iyı. Sadece yemek cepıllıđ braz ...	a:18:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Sadece yemek cepıllıđ braz az Cocuklar eskıyır ayrı ye...
409	192	Okul	n	Bu okula yollayıp cocugunuzu sızyet etmıyın.	a:8:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Bu okula yollayıp cocugunuzu sızyet etmıyın
512	235	Öđrencler	n	Br ebeveıyın ve ayrı zamanda burada okunıy br öđrencler ...	a:30:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	ebeveıyın ve ayrı zamanda burada okunıy br öđrencler ola...
578	272	İlgı	n	İlgı gk az, alaka gk sıırlı. Öđrenclerı dıđunıen yok.	a:14:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	İlgı gk az, alaka gk sıırlı
592	296	Fıksele İlgı	n	Eđtım ve fıksele kılınan vasat dercede.	a:7:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Eđtım ve fıksele kılınan vasat dercede
600	302	Fıksele İlgı	n	Öđrenclere ilgılenı eđretmenler var. eđtım programı...	a:16:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Fıksele inkanlar yeterısz
627	385	Fıksele İlgı	n	Öđretmenlerı öđrenclerı ıçın canla başla çalıřıyor. fıksele ...	a:20:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Öđretmenlerı öđrenclerı ıçın canla başla çalıřıyor
654	396	Okul	n	Cocugunuzu bu okula yollayıp harap etmıyın. Berde gk ...	a:16:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Cocugunuzu bu okula yollayıp harap etmıyın
670	408	Sosyal Faalıyet	n	Öđretmenlerınden memnun donanımlı ve lgıllere okulu...	a:18:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	okulun en kızı yanı hıđer gıkkde sosyal faalıyet dızele...
676	414	Öđretmen	n	Okulun eđtım durumu ve kaltesınden memnunuz. Ancak ...	a:12:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Ancak sıırlı eđretmen deđđıyor
684	417	Fıyat	n	Eđtımı iyı, fıksele inkanlar iyı. Fıyat ana braz fazla dıye dđ...	a:13:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Fıyat ana braz fazla dıye dıđıyılmıyorum
690	420	Kulüp	n	Ders cepıllıđ iyı ve tecrubelı eđretmenlerle çalıřıyorlar an...	a:13:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	cepıllıđ iyı ve tecrubelı eđretmenlerle çalıřıyorlar ancak k...
693	422	Okul	n	Tıcarı kayırları an planda olan taor yetıřtiren br okul burası.	a:13:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	planda olan taor yetıřtiren br okul burası
926	442	Displın	n	Okulun displını gerekeı abartı bulmıyorum.	a:6:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Okulun displını gerekeı abartı bulmıyorum
928	443	Yabarı Dı	n	Reıarı br kurunda olınası gerekeıđı kadar deđıslıdır. Abartı...	a:22:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	yabarı dıre yetenı olđđunu dıđıyılmıyorum
953	457	Okul	n	inanılmaz kızı br okul gımeıyın.	a:6:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	inanılmaz kızı br okul gımeıyın
974	467	Hoca	n	Eđtım apandan kızı br okulu. Hocaları yeterısz, öđrencler...	a:13:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Hocaları yeterısz öđrenclere yeterince ilgılenıyır
986	473	Eđtım kaltesı	n	Mınarısz, inkanlar gk iyı. Ancak eđtım kaltesı ozel okula...	a:14:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Ankac eđtım kaltesı ozel okular sevıyesınde deđđ
1.003	484	Rehberlek	n	Rehberlek servısları gk zayıf. Yıgıyın br alkan zorbalıđı var...	a:14:(0:a:8:(0:0:1:1"1"1:1:10:"0:0:0"1:2:1:10:"0:0"...	(N.L.L)	Rehberlek servısları gk zayıf

Şekil 7.5: Yorumların Veritabanına Aktarılmış Hali

```

tokenizer = Tokenizer(
    num_words=None,
    filters='!\"#$%&()*+,-./:;<=>@[\\]^_`{|}~\t\n',
    lower=True,
    split=" ",
    char_level=False,
    oov_token=None,
    document_count=0
)

tokenizer.fit_on_texts(data)

#tokenleri bir dosyaya kaydediyor.
import pickle
with open('tokenizer.pickle','wb') as handle:
    pickle.dump(tokenizer, handle, protocol=pickle.HIGHEST_PROTOCOL)
    
```

Şekil 7.6: Tokenizer Oluşturma Scripti

```

tokenizer.json
1 [{"bir": 1, "ve": 2, "çok": 3, "okul": 4, "iyi": 5, "eğitim": 6, "eğitimi": 7, "de": 8, "her": 9, "imkanları": 10, "bu": 11, "için": 12, "sosyal": 13, "okulun": 14,
- "yabancı": 38, "ilgili": 39, "harika": 40, "olarak": 41, "daha": 42, "olsun": 43, "gerçekten": 44, "okula": 45, "tüm": 46, "disiplinli": 47, "geniş": 48, "brans":
- "kadrosu": 73, "konusunda": 74, "aktif": 75, "ki": 76, "dört": 77, "yeterli": 78, "uzman": 79, "yemekleri": 80, "yemekler": 81, "sınıfları": 82, "sağlıklı": 83,
- "çocuklar": 106, "zengin": 107, "çeşitliliği": 108, "kampüsü": 109, "ders": 110, "çocuklara": 111, "hepsi": 112, "derece": 113, "acısından": 114, "öğretmenlerle":
136, "sey": 137, "uygun": 138, "aktiviteler": 139, "okulumuz": 140, "mutlu": 141, "şekilde": 142, "kalabalık": 143, "eğitiminden": 144, "çalışan": 145, "derslerin
- "güvenliği": 168, "aktiviteleri": 169, "temel": 170, "yönetimi": 171, "öğretmenleriyle": 172, "laboratuvarları": 173, "olsa": 174, "rağmen": 175, "bahçesi": 176,
- "yapısı": 197, "eğitime": 198, "öğretmenlerinden": 199, "hijyen": 200, "temizlik": 201, "imkanlara": 202, "başarıları": 203, "sadece": 204, "aynı": 205, "hizmete
- "okullarından": 226, "temizliğe": 227, "aktivitelerde": 228, "kurumu": 229, "ortam": 230, "örnek": 231, "sunan": 232, "sürekli": 233, "sağlayan": 234, "olabilir":
256, "sıkı": 257, "gönderiyorum": 258, "derslerinin": 259, "dolü": 260, "açıdan": 261, "başka": 262, "ferah": 263, "düzenli": 264, "tertemiz": 265, "eğitmen": 266
288, "eğitiminin": 289, "okuldaki": 290, "ilgilliler": 291, "ettik": 292, "tabii": 293, "kendilerini": 294, "imkanı": 295, "bana": 296, "süper": 297, "personel": 2
- "anlamında": 319, "onun": 320, "haricinde": 321, "yetiştirmeyi": 322, "hizmetlerden": 323, "alıyor": 324, "oğlumu": 325, "işleniyor": 326, "yuvası": 327, "odaklı"
- "verimli": 349, "yemek": 350, "gördüğüm": 351, "beğeniyorum": 352, "bulunan": 353, "öğrencilerle": 354, "yetersiz": 355, "inanılmaz": 356, "devlet": 357, "müdürlük
- "iletişim": 378, "yeteneklerini": 379, "bunları": 380, "zayıf": 381, "temizliği": 382, "uzak": 383, "çocuklarımız": 384, "materyalleri": 385, "dışında": 386, "ço
- "cocuğunuz": 408, "kalitesinden": 409, "hava": 410, "zevklil": 411, "getirmeyi": 412, "almış": 413, "yemeklerinden": 414, "anlayışları": 415, "emeği": 416, "dersi
437, "alanlar": 438, "olusan": 439, "nesiller": 440, "vermeye": 441, "atölye": 442, "bizim": 443, "arasında": 444, "sene": 445, "etkinliklerde": 446, "derslerinde
- "küllülerinden": 468, "konuda": 469, "çalışanlar": 470, "sahipler": 471, "kıyasla": 472, "sınıflara": 473, "tamamen": 474, "derecede": 475, "dünyanın": 476, "yeşi
496, "insan": 497, "sonuna": 498, "iyisini": 499, "öğretmenimiz": 500, "başarısını": 501, "yenilikçi": 502, "sınıflarda": 503, "çocuklarımızın": 504, "yetiştirmek":
525, "ortaya": 526, "seviye": 527, "farkı": 528, "geleceğe": 529, "küllülerinin": 530, "önlemleri": 531, "küllülerine": 532, "kolejlere": 533, "önemli": 534, "sam
- "seven": 556, "yapıya": 557, "mezun": 558, "külüp": 559, "baya": 560, "hocalar": 561, "eğitimi": 562, "iletişimi": 563, "takip": 564, "birlikte": 565, "ediyoru
- "kısıtlı": 586, "aklama": 587, "karşılıyor": 588, "vermesinin": 589, "olanağı": 590, "sağlıyor": 591, "avantaj": 592, "başarıları": 593, "gündeme": 594, "okullarla
- "öğretmenimizden": 615, "yapmaya": 616, "düşünmüşler": 617, "yönelik": 618, "seviyor": 619, "bizleri": 620, "ortamıyla": 621, "elden": 622, "önce": 623, "başarara
645, "gelişmiş": 646, "göstördükleri": 647, "alakadan": 648, "geliştirmeye": 649, "giriş": 650, "derslere": 651, "verildiğini": 652, "yemekhane": 653, "kısıllık"
674, "aktifler": 675, "yıldızlı": 676, "okulundan": 677, "konsept": 678, "binasıya": 679, "hersey": 680, "sosyalleşebilmesi": 681, "ilgisi": 682, "konumda": 683,
- "durumu": 702, "geliştirmiş": 703, "gelişim": 704, "doğrultusunda": 705, "çocuğuma": 706, "standartlarına": 707, "fazlasıyla": 708, "kurumun": 709, "seçin": 710,
730, "olsada": 731, "etmeye": 732, "ortaokul": 733, "kalyon": 734, "etkinliklerini": 735, "geliştiren": 736, "acılarından": 737, "kocaman": 738, "geziler": 739, "k
- "nazaran": 759, "olmalarına": 760, "olduklarını": 761, "yemekhanesinde": 762, "öğün": 763, "yemekhanesinden": 764, "faaliyetlerden": 765, "mahrum": 766, "biricik"
786, "kendinden": 787, "güleyizli": 788, "koridorlarının": 789, "okuldayken": 790, "huzurlu": 791, "personellerin": 792, "tamamı": 793, "olmasına": 794, "kadro
814, "beni": 815, "olduğu": 816, "beylikdüzü": 817, "bolgesinde": 818, "topluma": 819, "öğrenme": 820, "ortamının": 821, "bulunduğu": 822, "yaklaşımınız": 823, "b
- "çalışmalarıyla": 845, "detayı": 846, "vell": 847, "ilişkisi": 848, "uzmanlar": 849, "kavramaya": 850, "olmayan": 851, "anlamaya": 852, "okulunu": 853, "bundan":
- "zence": 874, "alıyorum": 875, "çocuklara": 876, "hibitöde": 877, "çevreden": 878, "inifitinde": 879, "hinditinde": 880, "novin": 881, "kalitecine": 882, "malic
C:\Users\MARMARA\Desktop\Doktora-ML-Script\tokenizer.json | 11
CRLF UTF-8 JSON GitHub Git (0) | update

```

Şekil 7.7: Oluşturulan Tokenizer

- Yorum veri setinin eğitim ve test amaçlı bölünmesi işlemi,
- Yorumları tokenleştirme işlemi,
- Her bir yorumun aynı sayıda sayı ile temsil edilebilmesi için yorumların başına 0 koyarak 50 kelimeye tamamlama işlemi,
- Model katmanlarının oluşturulması işlemi,
- Modelin eğitilmesi ve test edilmesi işlemi.

Modelin oluşturulmasında numpy, pandas, json ve keras kütüphaneleri kullanılmıştır.

```

import numpy as np
import pandas as pd
import json

from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dense, GRU, Embedding, CuDNNGRU
from tensorflow.python.keras.optimizers import Adam

```

Şekil 7.8: Modelin Oluşturulmasında Kullanılan Kütüphaneler

Yorum veri setinin eğitim ve test amaçlı bölünmesi

Yorum veri setinin %70'i modelin eğitimi için, geri kalan

```

cutoff = int(len(data) * 0.70)
x_train, x_test = data[:cutoff], data[cutoff:]
y_train, y_test = target[:cutoff], target[cutoff:]

```

Şekil 7.9: Yorum Veri Setinin Bölünmesi İçin Geliştirilen Script

Yorumları tokenleştirme

Her bir yorumun aynı sayıda sayı ile temsil edilebilmesi için yorumların başına 0 koyarak 50 kelimeye tamamlanmıştır. Eğitim ve test kümeleri oluşturulan fonksiyon yardımı ile tokenleştirilmiştir.

```

with open('tokenizer.json') as json_dosyasi:
    json_tokenizer = json.load(json_dosyasi)

def tokenlestir(yorumListesi):
    yorumlar = []
    for yorum in yorumListesi:
        yorum = []
        for kelime in yorum.split():
            if (len(yorum) < 50 and kelime in json_tokenizer):
                yorum.append(json_tokenizer[kelime])

        if (len(yorum) < 50):
            sifirlar = list(np.zeros(50 - len(yorum), dtype = int))
            yorum = sifirlar + yorum

        yorumlar.append(yorum)
    return np.array(yorumlar, dtype = np.dtype(np.int32))

ekume = tokenlestir(x_train)
tkume = tokenlestir(x_test)

```

Şekil 7.10: Tokenleştirme Fonksiyonu

Aşağıdaki görselde tokenleştirilmiş ve başına 0 konarak 50 elemanlık bir diziyeye dönüştürülmüş bir yorum metni görüntülenmektedir.

```

array([ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,  0,  0, 134, 264,
        326,  2, 68, 102, 765, 766, 573,  1, 545, 50, 187], dtype=int32)

```

Şekil 7.11: Tokenleştirilmiş yorum

Model katmanlarının oluşturulması işlemi

Modelin oluşturulmasında 3 katmanlı YSA kullanılmıştır. İlk katman 16, ikinci katman 8, üçüncü katman 4 nörondan oluşmaktadır.

- Embedding Katmanı (Vektörleştirme)
- GRU Katmanı (Çok katmanlı YSA)
- Softmax Aktivasyon Fonksiyonu Katmanı

```

model = Sequential()
embedding_size = 50

model.add(Embedding(input_dim=10000,
                    output_dim=embedding_size,
                    input_length=50,
                    name='embedding_layer'))

model.add(GRU(units=16, return_sequences=True))
model.add(GRU(units=8, return_sequences=True))
model.add(GRU(units=4))
model.add(Dense(3, activation='softmax'))

optimizer = Adam(lr=1e-3)

model.compile(loss='sparse_categorical_crossentropy',
              optimizer=optimizer,
              metrics=['accuracy'])

```

Şekil 7.12: Modelin Yapısı

Modelin eğitilmesi ve test edilmesi

Model epochs=5 olarak eğitildi ve test verileri ile test edildi. Modelin başarı oranı %95,5 olarak tespit edildi.

```

model.fit(ekume, train, epochs = 5, batch_size = 10)

Epoch 1/5
1635/1635 [=====] - 23s 14ms/sample - loss: 0.3219 - acc: 0.9248
Epoch 2/5
1635/1635 [=====] - 22s 13ms/sample - loss: 0.2470 - acc: 0.9303
Epoch 3/5
1635/1635 [=====] - 21s 13ms/sample - loss: 0.1618 - acc: 0.9517
Epoch 4/5
1635/1635 [=====] - 21s 13ms/sample - loss: 0.1022 - acc: 0.9731
Epoch 5/5
1635/1635 [=====] - 20s 12ms/sample - loss: 0.0773 - acc: 0.9835
<tensorflow.python.keras.callbacks.History at 0x630359a90>

model.evaluate(tkume, est)

289/289 [=====] - 1s 2ms/sample - loss: 0.1247 - acc: 0.9550
[0.12470118970864784, 0.9550173]

```

Şekil 7.13: Modelin Eğitilmesi ve Test Edilmesi

7.7 Öneriler

Türkçe DDİ alanında yapılan çalışmaların geliştirilmesi için:

1. Türkçe dilinde duygu ve kelime sözlükleri geliştirilmelidir. Yabancı dillerden çevrilen sözlüklerde duygu ve anlam kaybı yaşanmaktadır.
2. Akademik ve ticari çalışmalarda kullanılmak üzere Türkçe etiketli veri setleri oluşturulmalıdır. Google'ın captcha sisteminde olduğu gibi sistemler geliştirerek etiketli veri setleri oluşturma süreci otomatize edilmelidir.
3. Zemberek ve İTÜ NLP gibi doğal dil işleme yazılım zinciri sistemleri geliştirilmeli ve desteklenmelidir.
4. Kullanılan yapay zeka kütüphanelerinin geliştirilmesi üzerine çalışmalar yapılmalıdır.
5. Aspect tabanlı çalışmalarda aspectlerin otomatik tespit edilmesine dair çalışmalar yapılmalı, algoritmalar oluşturulmalıdır.

Referanslar

1. Khurana D., Koli A., Khatter K., Singh S. Natural Language Processing: State of The Art, Current Trends and Challenges, ArXiv abs/1708.05148 (2017).
2. Oflazer, K. (2016). Türkçe ve Doğal Dil İşleme. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 5(2).
3. Delibas, A. (2008). Doğal Dil İşleme İle Türkçe Yazım Hatalarının Denetlenmesi (Doctoral dissertation, Fen Bilimleri Enstitüsü).
4. Hark, C., Myo, A., Seyyarer, A., Myo, G., Uçkan, T., Myo, B., & Karci, A. (2017, September). Doğal dil İşleme yaklaşımları ile yapısal olmayan dökümanların benzerliği. In 2017 International Artificial Intelligence and Data Processing Symposium (IDAP) (pp. 1-6). IEEE.
5. Adalı, E. (2012). Doğal Dil İşleme. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 5(2).

6. Linguistic Knowledge in Natural Language Processing. (2018, 26 Ağustos). Erişim adresi: <https://towardsdatascience.com/linguistic-knowledge-in-natural-language-processing-332630f43ce1>
7. Fersini, E., Messina, E., & Pozzi, F. A. (2014). Sentiment analysis: Bayesian ensemble learning. *Decision support systems*, 68, 26-38.
8. Özyurt, B., & Akçayol, M. A. (2018). Fikir madenciliği ve duygu analizi, yaklaşımlar, yöntemler üzerine bir araştırma.
9. Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4), 1093-1113.
10. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2), 267-307.
11. Song, H., & Lee, S.-Y. (2013). Hierarchical Representation Using NMF. *International Conference on Neural Information Processing 2013: Neural Information Processing*, (s. 466-473).
12. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug), 2493-2537.

8. App Inventor ile Anlık Çeviri Yardımcısı

App Inventor ile Anlık Çeviri Yardımcısı

Alpaslan KİBAR¹

¹Sakarya Üniversitesi, İşletme Fakültesi, İşletme Bölümü, Sakarya, Türkiye

8.1 Giriş

Yapay zeka her geçen gün popülaritesini arttıran konulardan biridir. Yapay zekanın alt konularından biri olan Makine Öğrenmesi bu alanda öne çıkan çalışma alanlarından biridir. Bilişim sistemlerinde makine öğrenmesi modelleri oluşturmak isteyen ancak algoritmaların teknik ayrıntılarına girmek istemeyen geliştiriciler için birçok makine öğrenmesi frameworkü bulunmaktadır. TensorFlow, PyTorch, Keras vb. frameworkler bunlara örnektir (Hale, 2020). Bu alana yeni giren araştırmacılara, teknik konularda birçok aracın bulunduğunu, ilk başta konu ile alakasız görülebilecek frameworkler ile bile makine öğrenmesi modellerinin oluşturulabileceğini göstermeyi amaçlayan bir çeşit rehber niteliğindeki bu çalışmada ise Massachusetts Institute of Technology (MIT) ve Google işbirliği ile oluşturulan App Inventor frameworkü kullanılacaktır. Rehberde, örnek bir problemin çözümü için kullanılacak makine öğrenmesi algoritmaları hakkında kısa bilgiler paylaşıldıktan sonra, asıl amacı kolay ve hızlı bir şekilde mobil cihazlar için uygulama geliştirmek olan app inventor frameworkü kullanılarak problemin çözümü için bir model oluşturma ve uygulamanın geliştirilmesi adımları paylaşılacaktır. Problem olarak, Türkçeden İngilizceye ve İngilizceden Türkçeye, ilk adımda konuşmaları anlayan, ikinci adımda çevirileri yapan, üçüncü adımda elde edilen çevirileri sesli hale dönüştüren bir model üzerinde çalışılacaktır.

8.2 Bir Bilgi Sistemi Geliştirmek

Bilgi sistemi geliştirme yaşam döngüsü planlama, analiz, tasarım, uygulama ve destek aşamalarından oluşmaktadır (wikipedia, 2020).

8.2.1 Planlama

Planlama aşamasında problemin tespiti ve tanımı yapılır. Temel gereksinimler tespit edilir. Çeşitli fizibilite çalışmaları yapılır. Projenin genel planlaması yapılır.

Google Translate, Skype gibi uygulamalar günümüzde gerçek zamanlı çeviri desteği vermektedir. Bu rehberde, profesyonel olarak hizmet veren bu kurumların uygulamalarına benzer bir uygulamanın, her ne kadar onların ürünleri kadar profesyonelce olmasa da basit versiyonlarının kolayca geliştirilebileceğinin gösterilmesi hedeflenmektedir. Yabancı dilin gerektiği her yerde bu uygulamanın basitçe kullanımının faydalı olması beklenmektedir.

8.2.2 Analiz

Analiz aşamasında sistem gereksinimleri netleştirilir ve belgeleme işlemleri tamamlanır.

Rehberdeki örnek uygulamada ihtiyaç olarak sadece Türkçeden İngilizceye ve İngilizceden Türkçeye gerçek zamanlı çeviri yapılabilmesi belirlenmiştir. Problemin çözümü için kullanılacak modeller çeşitli yöntemler yardımıyla (örneğin UML diyagramları) belgelenebilir.

8.2.3 Tasarım

Analiz aşamasında net tanımı yapılan problemin çözüm adımları bu aşamada belirlenir.

Rehberdeki örnek uygulama için sesi alacak ve metne dönüştürecek donanım ve yazılıma ihtiyaç olacaktır. Uygulamanın devamında elde edilen kaynak metni istenilen dile çevirecek bir çevirici yazılımın kullanılması planlanmaktadır. Son aşamada ise metnin çevirisi yapılmış durumunu tekrar ses haline dönüştürecek bir yazılıma ve bu sesi yayınlayacak bir donanıma ihtiyaç duyulmaktadır. Bu aşamada birden çok çözüm alternatifi araştırılabilir.

8.2.4 Uygulama

Tasarım aşamasında belirlenen çözüm alternatiflerinden en uygunu uygulama aşamasında seçilir. Seçilen çözüm yolunun kodlarının oluşturulması işlemleri tamamlanır. Yazılım oluşturulduktan sonra test işlemleri ihmal edilmemelidir.

Rehberdeki örnek uygulamada yazılımın geliştirme aracı olarak app inventor2 frameworkü kullanılacaktır.

8.2.5 Destek ve Geliştirme

Test işlemlerinden başarı ile geçen uygulamanın, kullanım kılavuzu gibi belgeleri de hazırlandıktan sonra uygulamayı talep eden kuruma teslimatı yapılır. Ürün kullanılırken alınan geri bildirimler doğrultusunda güncelleştirmeler de yapılmalıdır.

Rehberdeki örnek uygulama, kullanılmaya başlandıktan sonra kolayca üzerinde güncellemelerin yapılmasına olanak sağlayacaktır.

8.3 Yapay Zeka

Yapay Zeka, canlıların çeşitli problemleri çözmek için kullandıkları akıllı davranışlarını model olarak alıp, tahmin, kümeleme, sınıflandırma vb. yöntemler kullanarak problemlere çözüm önerileri geliştirmektir. En yaygın teknolojileri Makine Öğrenmesi, Derin Öğrenme, Uzman Sistemler, Genetik Algoritmalar, Bulanık Mantık, Yapay Sinir Ağları vb. yöntemlerdir (Muhammet ATALAY, 2017).

8.3.1 Makine Öğrenmesi

İnsan gibi öğrenme yeteneği olan ve veriler üzerinden çıkarımlar yapabilen algoritmaların oluşturulmalarını ve çalışmalarını inceleyen, gelen yeni verilere göre algoritmalarını güncelleyebilen bir yapay zeka teknolojisidir. Bu algoritmaların en temel özelliği, klasik program talimatlarını bire bir uygulamak yerine, örnek verilerden tahminler ve kararları oluşturabilmek için model oluşturmaya çalışmalarıdır.

Makine öğrenmesi, sonucu bilinen girdilerden hareketle, sonucu bilinmeyen girdilerin sonuçları için tahmin yapmaya dayanır. Veri madenciliğinde ise, daha önce bilinmeyen örüntüler, ilişkiler tespit edilerek, bilgiye ulaşılmaya çalışılır. Birçok makine öğrenmesi yöntemi vardır.

Denetimli

Etiketli veriler kullanır. Tahminler yapar. Eğitim verisinin hazırlanması zaman alır.

Önce papatya resimleri (etiketli) eğitim verileri ile sistem eğitilip sisteme papatya öğretilir. Daha sonra içerisinde hem papatya hem de farklı çiçeklerden olan test verileri için sistemin hangi verilerin papatya resmi olduğuna dair tahmin üretmesi beklenir.

Denetimsiz

Veriler etiketli değildir. Girdi verilerinin sınıfları yoktur. Eğitilmemiş veriler üzerinde çalışılarak, verilerin anlam/içerik/değer olarak kümeleneceği mantığı ile çalışır. Yeni veriler algoritma tarafından oluşturulan bu kümelere dağıtılır.

Eğitim verisi hazırlanmasının büyük yük olduğu durumlarda kullanılabilirler. Marketlerde bu satılırsa yanında şu da satılabilir şeklinde (apriori) algoritmalar kullanılabilir olsalar da müşteri sayısı çok olduğunda bu eğitilmiş verileri hazırlamak karmaşık olabilir, zaman alabilir, maliyetli olabilir. Bu gibi durumlarda benzer alışkanlıkları/özellikleri olan müşteriler gruplandırılabilir. Oluşan grupların özelliklerine göre raflar düzenlenebilir, stoklar planlanabilir.

Yarı Denetimli

Etiketlenmiş küçük miktardaki veri ile etiketlenmemiş büyük miktardaki veri beraber kullanılır.

Takviyeli Öğrenme (Reinforcement Learning)

Öğrenme, geri bildirimler sayesinde olur. Olası durumlar vardır. Alınan geri bildirim istenen durum değil ise ceza alınır ve o geri bildirim tetikleyen hamle tekrar edilmez. Alınan geri bildirim istenen durum ise ödül alınır ve bu deneyimden faydalanarak öğrenme işlemi devam eder. Amaç en fazla ödülü almaktır. Öğrenme işlemi sürekli ve sürekli. Öğrenmeyi durdurma gibi bir işlem yoktur.

Derin

GPU (Grafik İşlemci Ünitesi) kullanarak büyük miktardaki etiketlenmiş veri üzerinden, derin sinir ağları kullanarak yapılan öğrenmedir. Çok daha az güç ve altyapı ile çok daha büyük hacimli veriler üzerinde çalışarak sınıflandırma ve tahmin yapılabilir.

8.3.2 Birliktelik Analizi

Değişkenler arasındaki önceden belirlenmemiş ilişkileri bulmak için kullanılır. Geçmiş verilere bakar, birlikte gerçekleşme durumlarını inceler, örüntüleri ortaya çıkarmaya çalışır (veri madenciliği). Bulduğu örüntüler sayesinde de yeni veriler üzerinde tahminler oluşturur. Hangi müşterinin hangi ürünle beraber hangi ürünleri satın almış olduğu tespit edilerek raf düzenleri/stok planları/promosyonlar vb. planlanabilir. En çok kullanılan algoritmaların ortak amaçları veriler arasındaki örüntüleri/ilişkileri

ortaya çıkarmaktır. Kümeleme algoritmaları veri noktaları/değerleri ile çalışır. Birlikte analiz algoritmaları ise bu veri değerlerinin nitelikleri arasındaki ilişkiler üzerinde çalışılır.

8.3.3 Sınıflandırma Algoritmaları

Sınıflandırma algoritmalarında girdiler, daha önceden belirlenmiş olan sınıflara atanırlar. Denetimli öğrenme algoritmalarıdır. Metin tanıma problemlerinde kullanılabilen algoritmalarındandır.

Karar Ağaçları

Karar Ağacı (Decision Tree), belirlenmiş olasılıklar ışığında, tercihlerin getireceği riskler, hedefler, kazançların görselleştirilip kolayca hesaplanması ve kararların alınmasına yardımcı olan karar noktalarını inceleyen bir yöntemdir. Algoritmaların büyük çoğunluğu sınıflandırma, bir kısmı tahmin amaçlı algoritmalar. Çok az da olsa her iki amacı hedefleyen algoritmalar da bulunmaktadır.

Yapay Sinir Ağları

Her girdi kendi ağırlığı ile çarpılır, tüm girdilere yapılan bu işlemin toplam fonksiyonu düğüm (Nöron) için net girdiyi verir. Birçok YSA modelinde bu toplam fonksiyonu transfer fonksiyonu olarak kabul edilse de transfer fonksiyonunun kullanıldığı modeller de vardır. Bu transfer fonksiyonunda Bias gibi düzeltmeler de kullanılabilir. Transfer fonksiyonundaki değer aktivasyon fonksiyonuna gönderilir. Aktivasyon fonksiyonu sonucu elde edilen değer düğümün Y için (birden çok düğüme çıktı gönderebilir) çıktısını oluşturacaktır. YSA' da asıl amaç, düğümlere gelen girdi (X_1, X_2, \dots, X_n) değerlerine göre, kabul edilebilecek doğrulukta çıktı değerlerini üretebilecek ağırlıkların (W_1, W_2, \dots, W_n) belirlenebilmesidir ki buna öğrenme denir. Uyum gösterme (yeni türler vb.) ve sürekli öğrenme (yeni veriler vb.), eğitim verisinin kısıtlı olduğu ve yeni verilerin sürekli eklendiği durumlarda çok çok önemlidir. Ayrıca bir düğüm sadece tek bir düğüm tarafından etkilenmediğinde, etkileyici düğümlerden birindeki yetersiz veya bozuk veri sonucu çok fazla etkilemeyebilir.

Derin Öğrenme

Denetimsiz öğrenme tekniğidir. Günümüzde büyük miktarlarda veriye ulaşma imkanı ve süper bilgisayar performansları sayesinde (özellikle GPU' daki gelişmeler) popülaritesi artan bir yöntemdir. Konuşmayı tanıma, konuşmacıyı tanıma, görüntü tanıma vb. alanlarda kullanılabilir. Birden fazla katman ve çok fazla düğüm (bir milyardan fazla düğüme sahip bir derin öğrenme ağı için 16.000 bilgisayar kullanılmış – Google Brain) içerebilir. Google 2015 yılında konuşma tanıma uygulamasında CCT-LSTM tekrarlı yapay sinir ağı modelini kullanmıştır (AKPINAR, 2017).

Öznitelik seçimi en küçük kareler yöntemi ile yapılmakta, düğüm sayısı girdi katmanından çıktı katmanına doğru ilerlerken azaltılmaktadır (GLAUNER, 2015). Tekrarlı Yapay Sinir Ağı yöntemidir. Yolun derinliği gizli katmanların sayısının bir fazlasıdır (çıktı katmanı +1 olarak ekleniyor). Katman sayısı 2' den fazla olduğunda derin ağ, 10' dan fazla olduğunda çok derin ağ olarak tanımlanabilir.

GPU, vektör ve matris hesaplamaları ile grafik işlemlerinde klasik CPU' ya göre 10 ile 100 kat arası daha iyi performanslar gösterebilmektedir. GPUlar paralel işlem yapma yeteneğine sahiptirler. Açık kaynak kodlu OpenCL ile GPU kullanarak hesaplama yapmak çok daha kolaylaşmıştır.

Yapay Sinir ağları ve derin öğrenme algoritmaları ile Metin çeviri/tercüme, Doğal dil işleme problemleri çözülmeye çalışılabilir. Girdi katmanı, Gizli katman (derin öğrenmede gizli katman sayısı birden fazla) ve çıktı katmanından oluşan bu algoritmalarda hesaplama yapılırken paralel işlemler yapılabileceğinden (derin öğrenme GPU kullanılabilir) diğer yöntemlerden farklıdır. (Muhammet ATALAY, 2017)

8.3.4 Kümeleme Algoritmaları

Veri setindeki veriler özelliklerine göre gruplanır. Kümeler arası benzerlikler mümkün olduğunca az, küme içindeki verilerin özellikleri mümkün olduğunca çok olmalıdır. Örneğin müşterilerin alışveriş miktarlarına göre kümelenebilir. Küme sayısı araştırmacı tarafından belirlenebileceği gibi bu işi algoritmaya da bırakılabilir. Algoritmanın oluşturduğu kümelerin sayısı da azaltılabilir. Müşteriler az/orta/çok alışveriş yapan şeklinde kümelenebilir. Çok fazla küme ile çalışmak birbirlerine çok benzeyen kümelerin oluşmasına neden olabilir. K-Means Algoritması optimum küme sayısı konusunda yardımcı olur (K küme sayısıdır). Önce rasgele merkezler atanır. Veriler bu merkezlerden en yakın olanına göre kümelendir. Sonrasında merkezler kümedeki verilere göre tekrar belirlenir. Tekrar veriler merkezlere göre kümelendir. Sistem kararlı hale gelene kadar işlem devam eder.

Kümeleme algoritmalarında girdiler, önceden belirlenmiş kriterlere göre (örneğin küme sayısı) gruplanırlar. Denetimsiz öğrenme algoritmalarıdır. Konuşma tanıma problemlerinde kullanılabilen algoritmalarındandır. Konuşma sinyalini girdi olarak alıp sayısallaştırır, öznitelikler dizisi oluşturulur, son olarak kelime dizisi şeklinde çıktı oluşturulur. (SARAÇLAR, 2020)

Hiyerarşik Küme Analizi

Benzer öznitelik değerlerine sahip nesnelerin bir araya getirilmesi (birleştirici/tümevarım – AGNES/AGglomerative NESTing) veya bütünden ayrılması (bölücü/tümden gelim – DIANA/Divisive ANALysis) yöntemlerinden biri kullanılarak yapılan analizdir. Başka analizlerin yapılabilmesini kolaylaştırmak amacı ile sadece özniteliklerin kümelenebilirliği amacı için de kullanılabilir (AKPINAR, 2017).

Genelde Tümevarım daha çok kullanılır.

Hiyerarşik küme analizinde, veri setinde yer alan tüm nesnelere arasındaki uzaklık/benzerlik her adımda tekrar hesaplanır.

Hiyerarşik Kümeleme Analizinde en çok kullanılan görselleştirme tekniği dendrogram' dır.

Bölümleyici Küme Analizi

AGNES ve DIANA gibi hiyerarşik küme analizi yapan algoritmalar, durum uzayını araştırarak olası bütün durumlardan global optimumu bulmayı hedeflemektedirler. Veri tabanlarının çok büyüdüğü günümüzde birçok durumda bu yaklaşım çok zaman ve maliyete yol açacağından tercih edilmeyebilir. Böyle durumlarda kısa sürede, global optimum olduğu varsayılan, belli bir hata payını kabul eden sonuçlar üreten heuristic yöntemler kullanılabilir. Günümüzde bölümleyici (partitioning) yaklaşımın en çok kullanılan algoritması k-means analizidir.

Bu yöntemde, girilen veri setinin nesnelere, başlangıçta verilen k küme sayısındaki (en çok eleştirilen yanı bu küme sayısının başlangıçta istenmesi) kümelere dağıtılması amaçlanmaktadır.

Bu yöntemde k sayısı dışında, işlemlerin çok uzun (veya sonsuz sayıda) sürmemesi için, iterasyon sayısı ve/veya yakınsama kriteri (convergence criterion) değeri de tanımlanmaktadır. Buradaki iterasyon sayısı, işlemlerin maksimum kaç defa tekrarlanacağını belirlerken, yakınsama kriteri kabul edilebilir hata değerini göstermektedir.

Fuzzy Kümeleme

Kümeleme algoritmalarının çoğunda bir örnek sadece bir kümenin üyesi olabilmektedir. Fuzzy kümeleme algoritmalarında ise bir nesne birden fazla kümenin üyesi olabilmektedir. Fuzzy kümeleme algoritmalarında bir nesne ile bir küme arasındaki ilişki düzeyi yani üyelik düzeyi değeri hesaplanmaktadır. En çok kullanılan fuzzy algoritması Fuzzy c-means algoritmasıdır. Fuzzy c-means, k-means algoritmasının, bazı parametre ilaveleri ile fuzzy kurallarına göre yeniden düzenlenmiş halidir.

Amaç fonksiyonu $1 \leq m < \infty$ iken üyelik düzeyi $0 \leq U \leq 1$ olabilir. (AKPINAR, 2017)

8.3.5 TTS (Text to Speech) Algoritmaları

TTS algoritmaları ile metin konuşma dilinde sese çevrilebilir. Bu algoritmalar genellikle cümlelerin nerede başlayıp nerede bittiği gibi bilgileri elde eden Yapı Analizi (Structure Analysis), konuşulacak dilin özel yapısına göre kısaltmalar, tarihler, zamanlar, numaralar vb. metnin analiz edildiği Metin Önışleme (Text pre-processing), her kelimeyi ses birimine (hece) çeviren Yazılı Metinden Ses Birimine Çevirme (Text-to-phoneme conversion), cümle için uygun veznin bulunduğu (kelimeler için zamanlama, ritim, duraksama, vurgu vb.) Vezin Analizi (Prosody Analysis) ve her cümle için ses dalgasının üretildiği Dalga Üretimi (Waveform Production) adımlarını içerirler (Basa, 2020).

8.4 App Inventor 2

Mobil cihazlarda (cep telefonu, tablet vb.) kullanmak amacıyla uygulama geliştirmek için birçok yazılım geliştirme aracı kullanılabilir. Bu araçlardan birçoğunun kurulumu zaman almakta, yazılım geliştirme ara yüzünün öğrenilmesi zor olmaktadır. Bu rehberde kurulumu ihtiyacı duymayan (oluşturulan yazılımın sınanması için küçük uygulamalar gerekli), ara yüzünün öğrenilmesi diğer birçok yazılımı kıyasla daha kolay olan, MIT (Massachusetts Institute of Technology) ve Google işbirliği ile geliştirilen appinventor programına göz atılacaktır (MIT, 2020).

App Inventor 2 frameworkünün web sayfası <https://appinventor.mit.edu/explore/get-started> ' dir.

App inventor 2 ara yüzünü kullanıp mobil uygulama geliştirmeye başlanabilmesi için bazı şartlar bulunmaktadır. Bunlar;

- Geliştiricinin bir gmail hesabına sahip olması ve o an o gmail hesabı ile oturumun açık olması.
- Geliştirmenin yapıldığı bilgisayarda Google Chrome tarayıcısının bulunması.

App Inventor ile oluşturulan mobil uygulamaların test edilebilmesi için dört farklı seçenek (Şekil 8.1) sunulmaktadır. Birinci ve önerilen seçenek ile Android tabanlı işletim sistemine sahip mobil cihaza (uygulamanın yazıldığı bilgisayar ile test için kullanılacak mobil cihaz aynı ağ üzerinde olmalı) bir test uygulaması kurmanın adımları anlatılırken, ikinci seçenekte ise bir Chromebook cihaz ile uygulamanın test edilmesinin adımları anlatılmaktadır. Üçüncü seçenekte Android işletim sistemli mobil cihaz yok ise, oluşturulan uygulamanın bilgisayarda simüle edilebilmesi için bir Emulator kurulumunun adımları gösterilmektedir. Dördüncü ve son seçenekte ise Android işletim sistemli mobil cihaz ile bilgisayar arasında bir USB kablo aracılığı ile bağlantı var ise uygulamanın nasıl test edilebileceğinin yönteminin adımları anlatılmaktadır.

App Inventor 2 ile yeni bir projeye başlandığında ilk görülen ara yüzde (Şekil 8.2) "Designer" bölümünde oluşturulacak uygulamalarda kullanılabilecek bileşenleri içeren Palette paneli, mobil uygulamanın ön izlemesinin görülebildiği ve Palette panelinden istenilen bileşenlerin sürüklenip üzerine bırakılabildiği Viewer paneli, mobil uygulamada kullanılan bileşenlerin listelendiği ve bu bileşenlerin hiyerarşik yapılarının düzenlenebildiği Components paneli, mobil uygulamada kullanılacak bileşenlerin özelliklerinin görülebildiği ve değiştirilebildiği Properties paneli görülebilmektedir.

Designer bölümünde oluşturulan uygulama ara yüzünün kodları "Blocks" bölümünde oluşturulabilir. Yeni projeye başlandığında sağ üstteki Blocks butonuna basılarak erişilebilen bu bölümde, Blocks panelinde seçilen bileşenin kullanılabileceği kontroller, değişkenler (variables), mantıksal sınamalar (logic), matematiksel işlemler (math) vb. birçok bloğa erişilebilir. Aşağıdaki resimde (Şekil 8.3) örnek projedeki Button1 bileşeni seçildiğinde, hemen yan tarafta "when Button1.Click-do", "when Button1.GotFocus-do" gibi işlemlerle birlikte "Button1.BackgroundColor" (Button1' in arka plan rengi) gibi özelliklerini taşıyan bloklara da erişilebildiği görülebilmektedir.

Option One - RECOMMENDED

Build apps with an Android device and WiFi Connection (preferred): : [Instructions](#)

If you have a computer, an Android smartphone or tablet, and a WiFi connection, this is the easiest way to build and test your apps on the device.



Build your project on your computer



Test it in real-time on your device

Option Two

Build apps with a Chromebook: [Instructions](#)

Many Chromebooks are capable of running Android apps. That lets you build and run the finished app on the same device.



Build your project on your chromebook Test it in real-time and run the finished app on the chromebook

Option Three

Don't have an Android device? Use the Emulator: [Instructions](#)

If you don't have an Android phone or tablet handy, you can still use App Inventor. Have a class of 30 students? Have them work primarily on emulators and share a few devices.



Build your project on your computer Test it in real-time on your computer with the onscreen emulator

Option Four

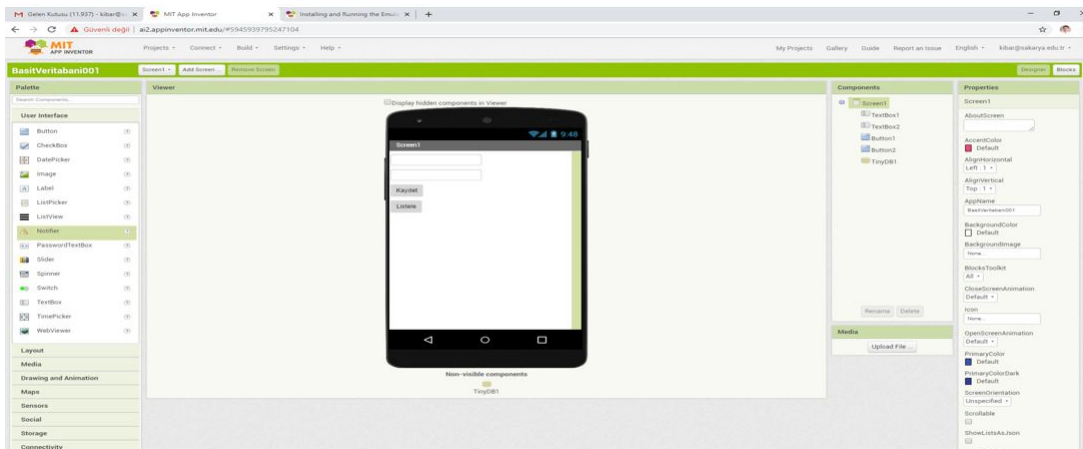
No WiFi? Build apps with an Android device and USB Cable: [Instructions](#)

Some firewalls within schools and organizations do not allow the type of WiFi connection required. If WiFi doesn't work for you, try USB.

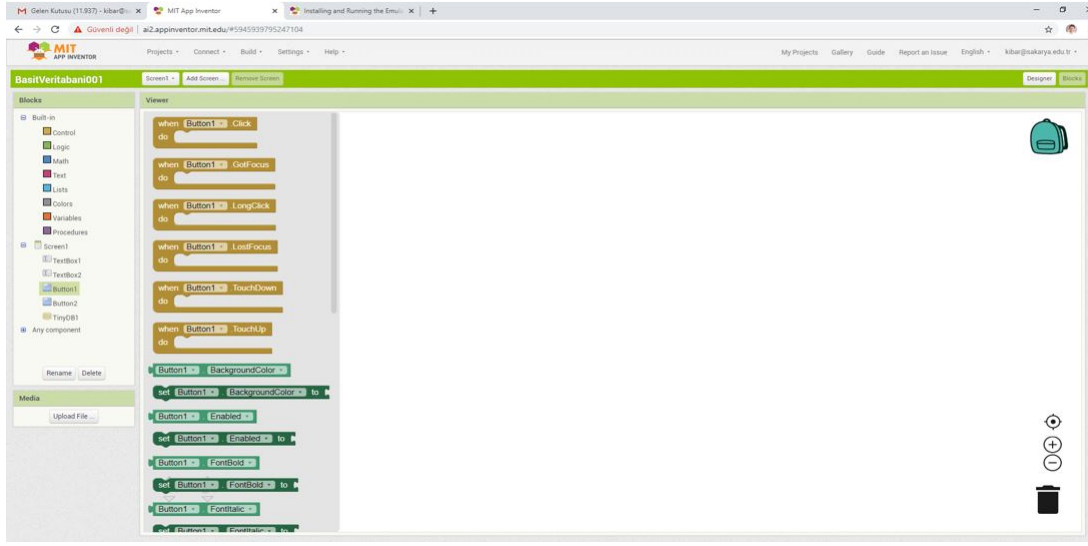


Build your project on your computer Test it in real-time on your device

Şekil 8.1: Test yöntemi seçenekleri



Şekil 8.2: App Inventor 2 yeni proje ara yüzü



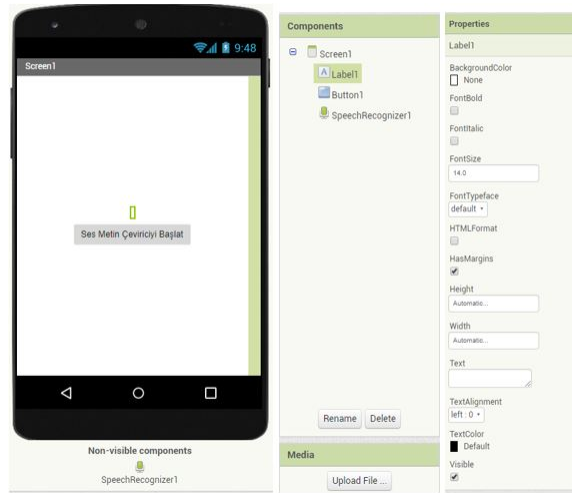
Şekil 8.3: App Inventor 2 Blocks görünümü

8.5 Uygulama

Uygulama üç adımdan oluşmaktadır. Birinci adımda çevrilmesi istenen cümle uygulama tarafından anlaşılıp metne dönüştürülecektir. İkinci adımda elde edilen metin çeviri algoritması yardımıyla hedef dildeki metne dönüştürülecektir. Üçüncü ve son adımda ise çevirisi yapılmış metin seslendirilecektir.

8.5.1 Sesin Alınıp Metne Dönüştürülmesi

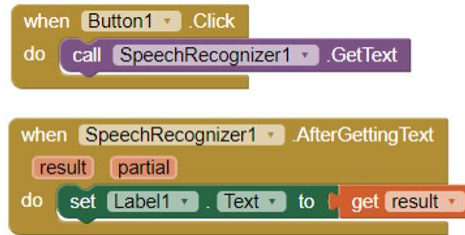
Bu projede bileşenler (Components) olarak 1 adet Label, 1 adet Button ve 1 adet SpeechRecognizer (Media komut grubunda) bileşenleri kullanılır.



Şekil 8.4: Sesi Metne çeviren uygulamanın bileşenleri

- (I) Yukarıdaki görüntüye (Şekil 8.4) benzer görünüm oluşturulur.
- (II) Kod blokları arayüzüne geçilir.

- (III) Örnek uygulama tek bir buton ile çalışmaktadır. Bu butona basıldığında alınan ses otomatik olarak metne çevrilecektir. Bu “Button1” isimli butonun “when Button1.Click – do” kontrolünün içerisine, “SpeechRecognizer1” in “call SpeechRecognizer1.GetText” prosedürü yerleştirilir.
- (IV) Artık ses metne çevrilmiştir ancak elde edilen metin hiçbir yerde gösterilmemektedir. Elde edilen metin “Label1” in Text özelliğinde gösterilebilir. Bu amaçla, “SpeechRecognizer1” in “when SpeechRecognizer1.AfterGettingText – do” özelliğine “Label1” in “set Label1.Text to” komutu yerleştirilir. Bu komuta parametre olarak da “when SpeechRecognizer1.AfterGettingText – do” komutunun geri döndürdüğü değerlerden “result” tutulup çekilerek yerleştirilir.
- (V) Örnek uygulama, 4. Bölümde gösterilen 4 seçenekten biri kullanılarak test edilir. Uygulamanın bloklarının son hali aşağıdaki gibi (Şekil 8.5) olacaktır.

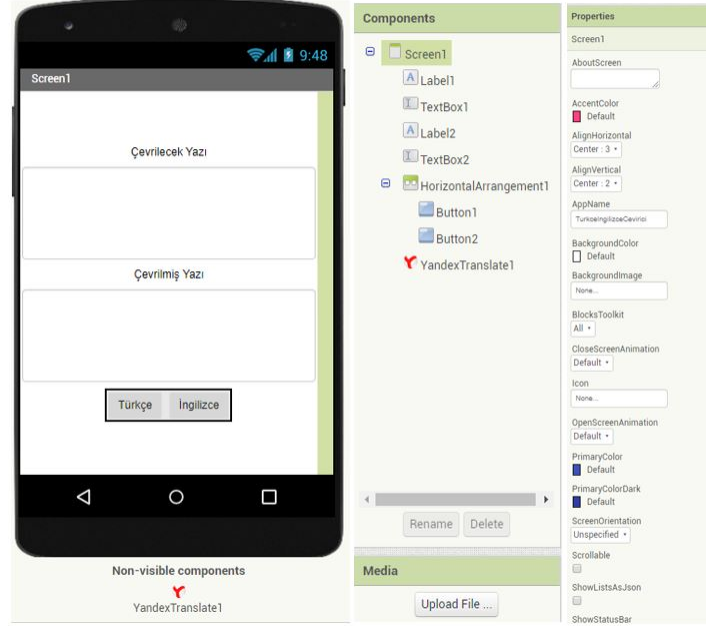


Şekil 8.5: Sesi Metne çeviren uygulamanın blokları

8.5.2 Metin Çeviri Algoritması Yardımıyla Hedef Dildeki Metne Dönüştürülmesi

Bu projede bileşenler (Components) olarak 1 adet HorizontalArrangement (Layout komut grubunda), 2 adet Label, 2 adet Button, 2 adet TextBox ve 1 adet YandexTranslate (Media komut grubunda) bileşenleri kullanılır.

- (I) Yukarıdaki görüntüye (Şekil 8.6) benzer görünüm oluşturulur.
- (II) Kod blokları arayüzüne geçilir.
- (III) Örnek uygulamada “TextBox1” in Text alanına yazılacak metnin, Türkçe (Button1) veya İngilizce (Button2) karşılığı “TextBox2” nin Text alanına yazdırılacaktır.
- (IV) “TextBox1” in Text alanına yazılacak metnin, Türkçe karşılığının “TextBox2” nin Text alanına yazdırılması için Button1 kullanılmaktadır. Bu nedenle “Button1” in “when Button1.Click – do” kontrolünün içerisine “YandexTranslate1” in “call YandexTranslate1.RequestTranslation – languageToTranslate/textToTranslate” prosedürü yerleştirilir. Bu prosedürün ilk parametresi olan “languageToTranslate” (hedef dil/çevrilmiş yazının dili) için “tr” metni girilir. İkinci parametre olan “textToTranslate” (kaynak metin) için “TextBox1.Text” girilir.
- (V) “TextBox1” in Text alanına yazılacak metnin, İngilizce karşılığının “TextBox2” nin Text alanına yazdırılması için Button2 kullanılmaktadır. Bu nedenle “Button2” in “when Button1.Click – do” kontrolünün içerisine “YandexTranslate1” in “call YandexTranslate1.RequestTranslation – languageToTranslate/textToTranslate” prosedürü yerleştirilir. Bu prosedürün ilk parametresi olan “languageToTranslate” (hedef dil/çevrilmiş yazının dili) için “tr-en” metni girilir. İkinci parametre olan “textToTranslate” (kaynak metin) için “TextBox1.Text” girilir.
- (VI) Bu aşamaya kadar çeviriler yapıldı ancak sonuçlar “TextBox2” nin Text alanına yazdırılmadı. Çeviri sonucunun TextBox2’ nin Text alanına yazdırılması için, “YandexTranslate1” in “when YandexTranslate1.GotTranslation – do” (IV. Veya V. Aşamada çeviri yapılırken bu kontrol

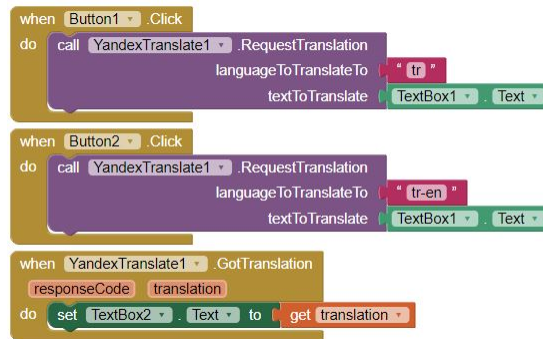


Şekil 8.6: Metin Çeviri Algoritması Yardımıyla Hedef Dildeki Metne Dönüştürülmesi uygulamasının bileşenleri

otomatik olarak çalışacaktır) kontrolünün içerisine “TextBox2” nin “set TextBox2.Text” komutu yerleştirilir. Bu komuta parametre olarak da “when YandexTranslate1.GotTranslation – do” kontrolünün geri döndürdüğü değerlerden olan “get translation” tutulup çekilerek yerleştirilir.

(VII) Örnek uygulama, 4. Bölümde gösterilen 4 seçenekten biri kullanılarak test edilir.

Uygulamanın bloklarının son hali aşağıdaki (Şekil 8.7) gibi olacaktır.

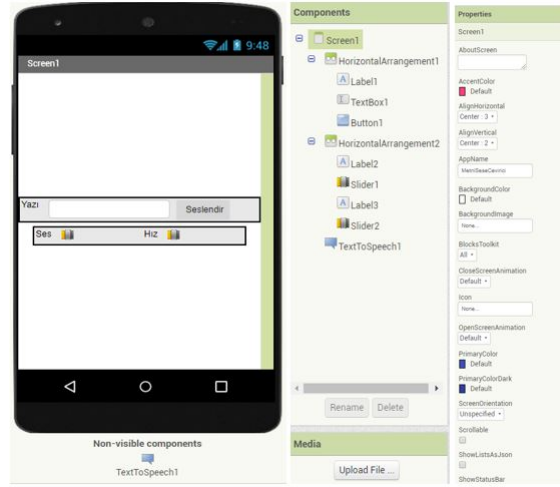


Şekil 8.7: Metin Çeviri Algoritması Yardımıyla Hedef Dildeki Metne Dönüştürülmesi uygulamasının blokları

8.5.3 Çevirisi Yapılmış Metnin Seslendirilmesi

Bu projede bileşenler (Components) olarak 2 adet HorizontalArrangement (Layout komut grubunda), 3 adet Label, 1 adet Button, 1 adet TextBox, 2 adet Slider ve 1 adet TextToSpeech (Media

komut gurubunda) bileşenleri kullanılır.

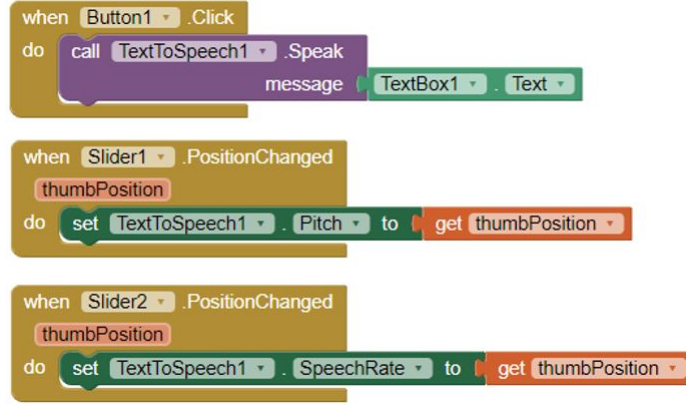


Şekil 8.8: Çevirisi Yapılmış Metnin Seslendirilmesi uygulamasının bileşenleri

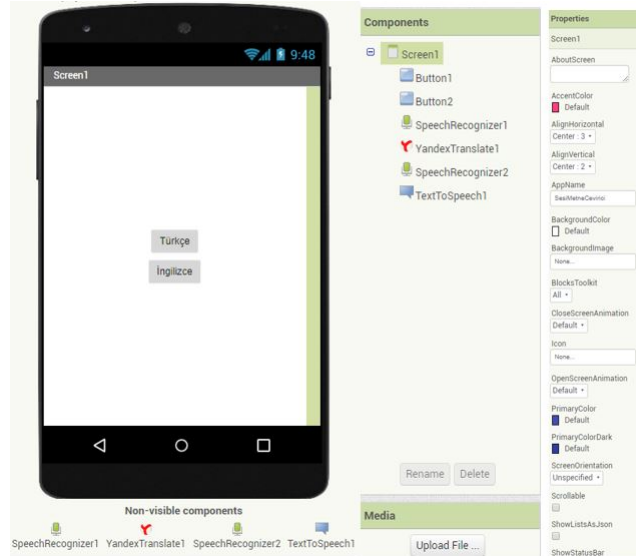
- (I) Yukarıdaki görüntüye (Şekil 8.8) benzer görünüm oluşturulur.
- (II) Kod blokları arayüzüne geçilir.
- (III) Örnek uygulama tek bir buton ile çalışmaktadır. Bu butona basıldığında “TextBox1” in Text alanına girilen metin, sese çevrilerek oynatılır. Bu amaçla “Button1” isimli butonun “when Button1.Click – do” kontrolünün içerisine, “TextToSpeech1” in “call TextToSpeech1.Speak - message” prosedürü yerleştirilir. message parametresi olarak da “TextBox1” in “TextBox1.Text” özelliği girilir.
- (IV) Oynatılan sesin seviyesini (volume) ayarlamak da mümkündür. Bu sesin seviyesini belirleyen “TextToSpeech1” in “Pitch” değeri bir sliderdan okunabilir. Bu amaçla “Slider1” in “when Slider1.PositionChanged – do” kontrolünün içerisine “TextToSpeech1” in “set TextToSpeech1.Pitch to” komutu yerleştirilir. Bu komuta parametre olarak da “when Slider1.PositionChanged – do” kontrolünün geri döndürdüğü değerlerden olan “get thumbPosition” tutulup çekilerek yerleştirilir.
- (V) Sese çevrilen metnin konuşulma hızı da ayarlanabilir. Konuşma hızını belirleyen “TextToSpeech1” in “SpeechRate” değeri de bir sliderdan okunabilir. Bu amaçla “Slider2” in “when Slider2.PositionChanged – do” kontrolünün içerisine “TextToSpeech1” in “set TextToSpeech1.SpeechRate to” komutu yerleştirilir. Bu komuta parametre olarak da “when Slider2.PositionChanged – do” kontrolünün geri döndürdüğü değerlerden olan “get thumbPosition” tutulup çekilerek yerleştirilir.
- (VI) Örnek uygulama, 4. Bölümde gösterilen 4 seçenektan biri kullanılarak test edilir. Uygulamanın bloklarının son hali aşağıdaki (Şekil 8.9) gibi olacaktır.

8.5.4 Bütünleşik Son Uygulama

5.1. ‘de bölümde ses metne dönüştürüldü, 5.2. ‘de bölümde çeviri yapıldı, 5.3. ‘de çevrilen metnin tercümesi tekrar sese dönüştürüldü. Tüm bu adımlar tek bir uygulamada da yapılabilirdi. Bu projede bileşenler (Components) olarak 2 adet Button, 2 Adet SpeechRecognizer (Medya komut grubunda), 1 adet YandexTranslate (Medya komut grubunda) ve 1 adet TextToSpeech (Medya komut grubunda) bileşenleri kullanılır.

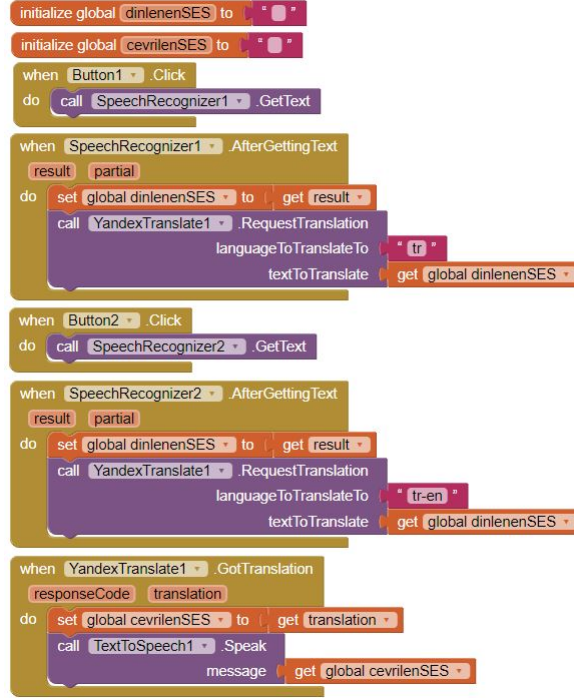


Şekil 8.9: Çevirisi Yapılmış Metnin Seslendirilmesi uygulamasının bileşenleri



Şekil 8.10: Bütünleşik Son Uygulamanın bileşenleri

Yukarıdaki ekran görüntüsü (Şekil 8.10) ve aşağıdaki kod blokları (Şekil 8.11) (rehberi okuyan geliştirici kendi bloklarını da oluşturabilir) kullanılarak, kullanıcının mobil cihaza türkçe söylediği cümleleri ingilizceye çevirip ingilizce olarak seslendiren (ingilizce isimli butona bastıktan sonra), kullanıcının ingilizce olarak söylediği cümleleri türkçeye çevirip türkçe olarak seslendiren (ingilizce isimli butona bastıktan sonra) uygulama oluşturulabilir.



Şekil 8.11: Bütünleşik Son Uygulamanın blokları

Bu rehberde projenin testi için birinci seçenek önerilmektedir. 5.4. 'deki bileşen ve blok görünümleri oluşturulduktan sonra, android tabanlı işletim sistemine sahip bir mobil cihaza "MIT AI2 Companion" uygulaması kurulur. App Inventor 2 ara yüzünde, Designer bölümünde, Connect menüsünden AI Companion seçilir. App Inventor 2 ara yüzünde bir barkod ekranı açılacaktır. Android tabanlı işletim sistemine sahip mobil cihazdaki "MIT AI2 Companion" uygulaması çalıştırılır. "scan QR code" butonuna tıklanıp mobil cihazın kamerası App Inventor 2 ara yüzünde barkoda çevrilir ve barkod okutulur. Barkod okutmak yerine "connect with code2" seçeneği kullanılarak App Inventor 2 ara yüzünde barkodla birlikte verilen 6 haneli kod kullanılarak da projenin test aşaması başlatılabilir. Seçilebilecek bu iki seçenekten biri doğru olarak kullanıldığında App Inventor 2 ara yüzündeki proje uygulaması mobil cihazda çalışmaya başlayacaktır. Proje başarı ile mobil cihazda çalışmaya başladığında mobil cihazın mikrofonuna "Nasılsın" şeklinde seslenilip proje uygulamasındaki "İngilizce" butonuna basıldığında (Konuşulan cümleyi anlamak için Google' ın uygulaması devreye girecektir) "Nasılsın" kelimesi metne çevilecektir. Sonraki adımda bu kelime Google' ın çeviricisi sayesinde "How are you?" ya dönüştürülecektir. Son olarak "How are you?" metni seslendirilecektir.

8.6 Beklentiler ve Öneriler

Burada google' ın uygulamaları bazı makine öğrenmesi algoritmalarını kullanmaktadır ancak bu örnek proje uygulamasını oluşturan geliştirici (bu rehberi okuyup uygulayan kişi) hiçbir makine öğrenmesi algoritmasının teknik seviyesini ayrıntılı bilmeden makine öğrenmesi algoritması kullanılan bir uygulamayı kolay ve hızlı bir şekilde geliştirmiş olacaktır.

Bu arayüzün kullanım kolaylığına ikna olan geliştiricinin yine bu arayüzü kullanarak başka uygulamalar geliştirmesi de umulmaktadır.

Eğer ara yüzün kullanım kolaylığı kabul görürse, bu ara yüz ve benzer ara yüzlerin üniversite öncesi öğretim kurumlarında kullanılıp kullanılmayacağı tartışılmalıdır. Bu tartışma yapılırken ara yüze erişim ve proje geliştirme ve yayınlama için çok az ön şartın olduğu dikkate alınmalıdır.

Referanslar

AKPINAR, H. (2017). DATA Veri Madenciliği Veri Analizi. İstanbul: Papatya Yayıncılık Eğitim A.Ş.

Basa, G. (2020, Eylül 9). Speech Synthesis Algorithms(Konuşma Sentezleme Algoritmaları). <http://gungorbasa.blogspot.com/>: <http://gungorbasa.blogspot.com/2011/02/speech-synthesis-algorithmskonusma.html> adresinden alındı

CEBECİ, H. İ. (2020, Mart 25). Kestirimci Modelleme Teknikleri. İş Zekası Ders Notları. Sakarya, Adapazarı, Türkiye: Sakarya Üniversitesi.

GLAUNER, P. O. (2015). Deep Convolutional Neural Networks for Smile Recognition. Deep Convolutional Neural Networks for Smile Recognition.

Goyal, K. (2020, Ekim 14). upgrad.com. Top 10 Deep Learning Frameworks in 2020 You Can't Ignore: <https://www.upgrad.com/blog/top-deep-learning-frameworks> adresinden alındı

Hale, J. (2020, Ekim 14). kdnuggets.com. Which Deep Learning Framework is Growing Fastest?: <https://www.kdnuggets.com/2019/05/which-deep-learning-framework-growing-fastest.html> adresinden alındı

MIT. (2020, Ekim 14). appinventor.mit.edu. MIT App Inventor: <https://appinventor.mit.edu> adresinden alındı

Muhammet ATALAY, E. Ç. (2017). BÜYÜK VERİ ANALİZİNDE YAPAY ZEKÂ VE MAKİNE ÖĞRENMESİ. Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 155-172.

SARAÇLAR, M. (2020, Eylül 9). Konuşma Tanıma için. <http://byoyo.cmpe.boun.edu.tr/>: <http://byoyo.cmpe.boun.edu.tr/sunumlar/muratsaraclar-byoyo18.pdf> adresinden alındı

wikipedia. (2020, Ekim 14). en.wikipedia.org. Systems development life cycle: https://en.wikipedia.org/wiki/Systems_development_life_cycle adresinden alındı



9. Sosyal Medya Verileri ile Duygu Analizi

Sentiment analysis

Sosyal Medya Verileri ile Duygu Analizi

Halil İbrahim Cebeci¹

¹Sakarya Üniversitesi, Serdivan, Sakarya, Türkiye

9.1 Giriş

Bilgi ve iletişim teknolojilerindeki son yıllardaki çok hızlı gelişmeleri takiben internete erişebilen cihazlar artık herkes tarafından kullanılarak günlük hayatın bir parçası olmuştur. Mobil cihaz sayısındaki artış, sosyal medya dediğimiz son yıllarda insan hayatını bir şekilde çevreleyen özel bir olgunun özellikle genç nüfus başta olmak üzere günlük yaşamın merkezine yerleşmesi sonucunu doğurmuştur.

2020 yılı verilerine göre Türkiye'deki 84 milyon kişinin %74 lük kısmının (62 milyon) internete bağlanan bir cihaza sahip olduğu görülmektedir. Bahse konu bu kişilerin günde ortalama 7,5 saat internet başında zaman geçirdikleri ve bu sürenin 2,5 saatini sosyal medyaya ayırdıkları görülmektedir (Bayrak, 2020). Bu rakamların özellikle pandemi sürecinde daha yukarı çekildiğini tahmin etmek çokta zor değildir. Çoğunlukla tercih edilen sosyal ağların başında Youtube, instagram, facebook, whatsapp ve twitter gelmektedir. Kullanıcıların % 96 sı sosyal medya hesapları günde en az bir defa takip ederken, % 88 lik kısmı ise en az bir paylaşım yapmaktadır (TUİK, 2020). Bu şekilde bakıldığında sosyal medya ve ağ konularının odak noktasında takip ve paylaşım konularının olduğunu söylemek yanlış olmaz. Bu iki kavram sosyal medya analitiği olarak adlandırılan özel veri analizi alanı için de temel veri kaynaklarını oluşturur.

Sosyalleşme insan hayatının önemli parçalarından biridir. Sosyal medya ise yüz yüze iletişiminin aksine özellikle anonim olması sayesinde sağladığı mahremiyet algısı ile sosyalleşme konusunda bireylere önemli bir destek sağlamaktadır. Bu anonimlik durumu bireylerin çok daha paylaşımcı olması sonucunu ortaya çıkarmaktadır (Varma, Kurisinkel & Radhakrishnan, 2017; Sailunaz, Dhaliwal, Rokne & Alhajj, 2018). Sosyal medya içeriği üretme (paylaşım) konusunda twitter diğer ağ sitelerine göre bir adım önde görülmektedir. Youtube, facebook, instagram gibi popüler siteler ise daha çok sunulan içeriğin takibi üzerine yoğunlaşmışlardır.

Sosyal medya içerikleri kullanıcı tarafından üretilen içerik (user generated content) sınıfında değerlendirilmekte olup ve içerdiği saklı olan bilgiler ile bireylere, işletmelere ve devlet kurumlarına kadar uzanan geniş bir havuz için çok değerlidir (Sun, Luo & Chen, 2017; Liu, 2020). Örneğin kullanıcıların pandemi sürecinde daha da ön plana çıkan e-ticaret alışverişlerinde yorum odaklı olarak satın alma gerçekleştirdiği, çoğu zaman ürüne gelen pozitif yorum sayısının, ürünün fiyatındaki belli oranda yükselmeyi dahi kompanse ettiği söylenebilir (Singh, Paul, Kumar & Arfi, 2014). İşletmelerin sosyal medya üzerinden topladığı veriler ile ürün ve/veya marka imajı hakkında fikir edinme, müşteriye daha iyi anlayarak pazarlama stratejisini geliştirme ve ürünlerindeki potansiyel sıkıntıları belirleyerek üretim süreçlerini güncelleme gibi birçok faaliyeti gerçekleştirebilirler (Singh vd., 2014; Sun vd., 2017; Poria, Hussain & Cambria, 2018). Bunun yanı sıra işletmeler ayrıca dahili veri kabul edilen müşteri geri bildirimleri üzerinde de bu tür analizleri gerçekleştirebilmektedir. Bu örnekleri eğlence sektöründen, politikaya, sağlık sektöründen, bilişim sektörüne kadar genişletebiliriz (Sun vd., 2017; Liu, 2020).

Sosyal medya içeriklerinde saklı olan değeri oluşturan temel unsur ise duygulardır. Bu duygular direkt olarak mesajlar ve yorumlar ile ifade edilebildikleri gibi, emojiler ve beğeniler ile de ortaya konabilir. Bütün bu unsurların otomatik (veya yarı otomatik) şekilde incelenmesi ve yorumlanması ile duyguların çıkarılması işlemi duygu analizi olarak adlandırılır. Özellikle kullanıcıların fazlasıyla paylaşımcı olduğu (Varma vd., 2017) sosyal medya içerikleri (başta twitter olmak üzere) duygu analizi için temel veri kaynağıdır.

Bu çalışmada sosyal medya verisi üzerinde duygu analizinin nasıl yapılabileceği konusunda teorik ve pratik bakış açıları ile birlikte bir altyapı oluşturulması amaçlanmaktadır. Bu amaç doğrultusunda hazırlanacak çalışma ile alanda araştırma yapmak isteyen akademisyenler ve profesyoneller için bir kılavuz olması düşünülmektedir. Çalışmanın sonraki kısımlarında öncelikle sosyal medya analitiği ve duygu analizi ile ilgili detaylı bilgiler verildikten sonra, son kısımda Covid 19 hashtag'i ile indirilen bir örnek twitter verisi ile duygu analizi çalışması yapılarak sunulacaktır.

9.2 Sosyal Medya Analitiği

Her insan bir şekilde sosyalleşme arayışındadır. Bu arayışın yönü 2000 li yıllar ile beraber bilişim üzerinden sosyal ağların kullanımına doğru kaymaktadır. Günümüzde bireyler özellikle getirdiği anonimlik ve mahremiyet özelliklerinden dolayı sosyal medyayı temel bir sosyalleşme aracı olarak tercih etmektedir (Varma vd., 2017). Davranış bilimleri açısından bireyler sosyal ağlar sayesinde aidiyet, öz sunum ve izlenim ihtiyaçlarını giderebilmektedirler (Pozzi, Fersini, Messina & Liu, 2017). Psikolojik etmenler ile teknolojik gelişim birlikte değerlendirildiğinde bu durum sosyal medya kullanımına yıllar içerisinde artan trendin bir göstergesi olarak kabul edilebilir.

Sosyal medya bireylerin sosyal olarak etkileşmesinin mümkün kılan web 2.0 temelli teknoloji ve yöntemler bütünü olarak değerlendirilebilir. Aşağıda sosyal medya alt türleri sunulmuştur (Sharda, Delen, Turban, Aronson & Liang, 2014).

- İşbirlikçi Projeler (Wikipedia, ekşisözülük)
- Blog ve Mikro Bloglar (Twitter)
- İçerik Toplulukları (Youtube, Dailymotion)
- Sosyal Ağlar (Facebook, LinkedIn)
- Sosyal Oyun Ağları (Wow)
- Sanal Sosyal Dünyalar (Second Life)

İşletmeler müşterilerini (iç ve dış) doğru şekilde yönetebilmek adına iş analitiği araçlarına sıklıkla başvurmaktadır (Isson & Harriott, 2013). Veri madenciliği ile yapısal formdaki verilerden

karar destek amaçlı bilgi çıkarımı yapılırken, metin madenciliği benzer amaçla yapısal olmayan formdaki dokümanların yarı otomatik analizinde kullanılmaktadır. Metin verilerinin en temel kaynağı internet ve sosyal medyadır. Bu bağlamdan kullanıcıların sosyal medyayı kullanım davranışlarının (takip) ve paylaştığı içeriklerin (paylaşım) analizi özel bir alan sosyal medya analitiği içerisinde değerlendirilirler.

İşletmeler ve kamu kurumları sosyal medya kullanımlarını analiz ederek hedef kitlelerini daha iyi bağlamak adına sosyal ve ticari politika ve stratejilerini geliştirme eğilimindedirler. Sosyal medya kullanım madenciliği olarak adlandırılan bu olgu temelde önceden belirlenmiş metrikler yardımıyla mevcut durumların değerlendirilmesi üzerine kuruludur. Web kullanım madenciliği yaklaşımı olarak ta değerlendirilebilecek bu alanda hedef kitlenin ne kadar paylaşım yaptığı, ne ölçüde hesapları takip ettiği, beğeni sayıları gibi özel metrikler yardımıyla sosyal medya kullanımlarının incelenmektedir.

Literatürde sosyal medya madenciliği ile daha çok kast edilen, sosyal medya içerikleri üzerine yapılan metin madenciliği eksenli çalışmalardır. Metin madenciliğinden farklı olarak değerlendirilmesi altında yatan temel neden ise sosyal medya mesajlarının (özellikler tweet ler) içerdiği özel jargon (kısaltmalar) ve resimler (emoji) ile düz metinlere göre yapısal olarak değişkenlik göstermesidir. Ayrıca bu mesajların gramer bağımsız yazım dilleri, fazladan farklı karakterler içermeleri ve devrik ve tam olmayan cümle yapıları ile yüksek gürültülü veri olarak değerlendirilmekte ve metin madenciliğine konu olan diğer metinlerden farklılaşmaktadırlar. Bütün unsurlar aşağıda kısaca özetlenmeye çalışılmıştır (Zafarini, Abbasi & Liu, 2014; Varma vd., 2017; Fersini 2017).

- Düz metinlere göre daha kısadırlar. Fakat özellikle kısaltma ve emoji kullanımı ile birçok fikir ve duygu barındırabilirler.
- Cümle düşüklüğü, kısaltmalar, farklı dillerin beraber kullanılması gibi özelliklerinden dolayı düz metinlere göre daha gürültülidir. Bu yüzden ön işleme süreçleri de nispeten daha zorludur.
- Sosyal medya içerikleri trend olan alanlar ile alakalı olduğundan düz metinlere göre daha dinamik ve çoğu zaman etkileri geçicidir.
- İçerikler belli bir alanda genel fikirlerden ziyade, içerik oluşturan kullanıcıların görüş, düşünce ve demografik özellikleri yansıtacak şekildedir.
- İçerikler bireysel olmasının yanı sıra paylaşım ağı eksenli olarak değerlendirildiğinde sosyal ilişkilere bağlıdır.
- Sosyal medya içerikleri metinlerin yanı sıra linkler, görüntüler, videolar ve sesler de içerdiği ve sürekli akan bir formatta olduğundan, çoğu zaman büyük veri karakteristiklerine sahiptir.

Yukarıdaki bütün karakteristik özellikler birlikte değerlendirildiğinde sosyal medya verisinin analizinde her aşamada özel yaklaşımlar gerektirdiği muhakkaktır. Bu anlamda veri çekme, ön işleme, analizi ve yorumlaması adımları metin madenciliği süreçlerine göre önemli farklılıklar göstermektedir.

Sosyal medya verilerinin özetlenmesi ve yararlı bilgiler çıkarılması adına birçok araç ve yöntem kullanılmaktadır. Metin madenciliği yöntemlerinden kelime torbası (Bag of Words) yöntemi ve kelime bulutları en sık kullanılan özetleme teknikleri arasındadır. Fakat basit frekans sayımına dayalı bu teknikler fikir ve duyguları çıkarmada yetersiz kalabilmektedir. Bu bağlamda duygu analizi veya fikir madenciliği olarak adlandırılan işletme analitiği yöntemi sosyal medya içeriklerinin değerlendirilmesinde ön plana çıkmaktadır.

9.3 Duygu Analizi

İşletmeler için tüketici davranışlarını incelemek her zaman için ön plandadır. Yapılan bir araştırmada (Ak, 2019) pazarlama alanında yapılan akademik çalışmaların büyük bir kısmı pazar araştırması

ve tüketici davranışları alanındadır. Anketler, mülakatlar ve odak grup çalışmaları yıllar içerisinde tüketiciyi anlamak adına kullanılan en temel araçlardır. Fakat 2000 yıllar ile birlikte sosyal medya kullanımının artması ve kişilerin yoğun şekilde paylaşım yapması bu alandaki bakış açısını değiştirmiştir. Artık bireylerin bir ürün veya marka hakkındaki görüşlerini almak için onlara ulaşılmasından ziyade, görüşlerini gönüllü olarak paylaştığı sosyal medya içeriklerini inceleyerek, yararlı bilgilerin çıkarılmasına doğru bir eğilim söz konusudur (Kou, Du, He & Ye, 2016). İçeriklerin ne tür bir duygu içerdiğini anlamaya yönelik bir yöntem olan duygu analizi son yıllarda sosyal medya analitiği alanında ön plana çıkmıştır.

Fikir madenciliği, fikir analizi, fikir çıkarımı, duygu madenciliği, öznellik analizi, ekti analizi, inceleme madenciliği gibi farklı isimlerle de anılan (Liu, 2020) duygu analizi (DA), metin, ses, görüntü verilerinden bilişim destekli olarak otomatik ve yarı otomatik olarak insan davranış, duygu ve inanışlarını çekmekte kullanılan bir yöntemdir (Poria vd., 2018). İnsan duygu ve davranışlarını anlamak her zaman akademisyenleri ve profesyonellerin çalışma alanı olmuştur (Golder, O'Conner, Hennessy, Gross & Gonzalez-Hernandez, 2020). Her ne kadar doğal dil işleme çalışmaları 1950 li yıllarda başlamış olsa da (Yue, Chen, Li, Zuo & Yin, 2019), özellikle duygu içeren (opinionated) dokümanlara erişimin 2000 li yılların başında artması ile birlikte araştırmacılar duygu analizine yönelmişlerdir (Pozzi vd., 2017; Yue vd., 2019; Liu, 2020). Bu yönelimde etkili diğer unsurlarda bilişim temelli çalışma alanları olan makine öğrenmesi, yapay zeka ve metin madenciliği çalışmalarındaki son yıllardaki gelişmelerdir.

Birçok farklı bilim dalından etkilenen interdisipliner bir yaklaşım olan (Singh vd., 2014; Poria vd., 2018) DA çoğu zaman doğal dil işleme ve semantik biliminin bir alt dalı olarak ele alınmakla birlikte aslında bu analizin ortaya koydukları ile her iki dala destek veren bağımsız bir yapıda olduğu söylenebilir (Singh vd., 2014; Liu, 2020). Çünkü DA çalışmaları sadece düz metinler ile sınırlı değildir. Gerek sosyal medyanın cümle yapısı ile örtüşmeyen gramer bağımsız yapısal olmayan metin yapısı (Varma vd., 2017), gerekse bu tür medyalar içerisinde duygu belirten diğer unsurların (resim, video, ses kaydı vb.) varlığı (Poria vd., 2018) bu analizin ayrıca değerlendirilmesi gerekliliği ortaya koymaktadır. Çok modlu (multimodal) duygu analizi olarak ta adlandırılan ve duygu ifade eden her türlü içeriğin incelendiği analizler son yıllar ön plana çıkmaktadır (Soleymani, Garcia, Jou, Schuller, Chang & Pantic, 2017; Nazir, Rao, Wu & Sun, 2020).

Duygu analizinde temel veri kaynağı daha önce ifade edildiği üzere sosyal medyadır. Bununla birlikte tartışma forumları, inceleme siteleri, bloglar, müşteri geri bildirim siteleri ve metin formatında bilgi talep eden bütün siteler duygu içeren subjektif dokümanlar içerdiğinden duygu analizi için veri kaynağı olarak ele alınabilirler (Poria vd., 2018). Bütün bu metinler için analize konu olduklarında bazı özellikleri taşımaları beklenmektedirler. Aşağıda bu özellikler kısaca tanımlanmıştır (Liu, 2017; Liu, 2020).

- **Duygu:** Bir fikir ile doğrudan ilişkili olan his, davranış, değerlendirme gibi subjektif unsurlardır. Duygular genel kabul görecektir şekilde rasyonel olabildiği gibi kişiye özel göreceli de olabilir. Pratikte duygu analizinin konu alanının kişiye özel duyguların olduğu söylenebilir.
- **Duygu Hedefi:** Metin içerisinde ifade edilen bir duygunun yöneldiği nesneye karşılık gelir. Metin içerisindeki her bir duygu ifadesinin bir nesneye bağlanması analiz sürecinin başarısı için hayati önem taşımaktadır.
- **Varlık:** Duygunun ifade edildiği bir ürün, hizmet, konu, organizasyon veya insan olabilir.
- **Polarite:** Bir duygu ifadesinin yönünü vermektedir. Duygu analizi sürecinde negatif, nötr ve pozitif olarak üçlü polarite kullanılabilmesi gibi pozitiflik ve negatiflik derecesi de (5 yıldız gibi) tercih edilebilir.

Bir örnek ile bu durumu inceleyelim. Aşağıda bir cep telefonu modeli ile ilgili bir müşteri

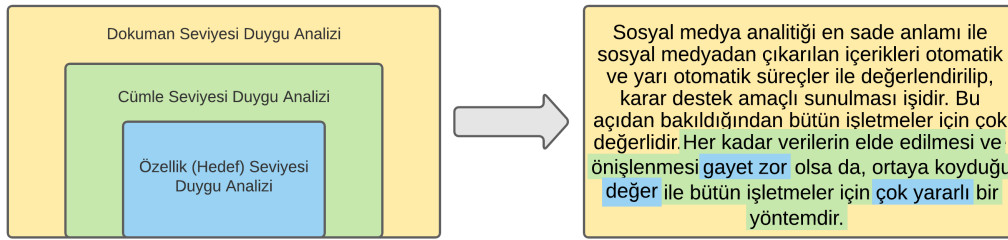
yorumu verilmiştir.

“Samsung Note 10 inanılmaz uzun batarya ömrü ile beni çok şaşırttı.”

Burada duygu ifade eden kelimeler “inanılmaz uzun” ve “şaşırttı” ifadeleridir. Bahse konu duygu için duygu hedefi ise batarya ömrüdür. Samsung Note 10 ifadesi de varlık olarak ele alınabilir. Duygu analizi tarafından bu cümle pozitif polariteli olarak işaretlenebileceği gibi yüksek pozitif olarak ta değerlendirilebilir. Eğer varlıklar ve duygu hedefleri doğru şekilde belirlenebilirse analiz başarımları yükselerek ve hatta topik modelleme yaklaşımları gibi özetleme yöntemleri ile farklı bakış açıları elde edilebilir.

9.3.1 Duygu Analizi Seviyeleri

Duygu analizi çalışmaları incelenen derlemin özellikleri, çalışma alanı ve analist tercihinine bağlı olarak farklı seviyelerde gerçekleştirilebilir. Analiz bir dokümanın tamamının değerlendirilebileceği doküman (yorum, tweet, post) seviyesinde gerçekleştirilebileceği gibi, dokümanların cümlelere ayrılması ile cümle seviyesinde de yürütülebilir. Hatta cümle içerisindeki duyguların belirlenmesi ile varlık (entity) veya özellik (aspect) seviyesinde de yapılabilir (Medhat, Hassan & Korashy, 2014; Pozzi vd., 2017; Liu, 2020). Tamamen derlem dediğimiz analize konu veri kaynağına bağlı olarak değerlendirilebilecek bu durum için evrensel doğru bir yaklaşım söz konusu değildir.



Şekil 9.1: Duygu Analizi Seviyeleri

Doküman Seviyesi Duygu Analizi (DS-DA) yaklaşımlarında incelenen içerik bir bütün olarak duygu polaritesi açısından değerlendirilir (Pradhan, Vala & Balani, 2016; Sun vd., 2017; Pozzi vd., 2017; Liu, 2020). Burada doküman ile kast edilen mesaj, tweet, post, müşteri yorumu veya geri bildirim olabilir. DS-DA bütün dokümanın tek bir varlık ile ilgili olduğunu varsayar. Bir müşteri e-ticaret sitesindeki spesifik bir cep telefonu modeli ile ilgili bir yorum yazdığında, DS-DA bu yorumun polaritesini hesapladıktan sonra bu cep telefonu modeline bağlar. Fakat birden fazla model ile ilgili karşılaştırmalı bir yorum söz konusu ise analiz bu durumu dikkate almadan sonuç üreteceğinden başarımları konusunda tereddütler olacaktır (Liu, 2020).

Cümle Seviyesi Duygu Analizi (CS-DA): Bir dokümanın cümlelere bölünerek her bir cümle için değerlendirmelerin yapılabileceği DA türüdür (Pradhan vd., 2016; Liu, 2020). Yine DS-DA yaklaşımındaki tereddütler burada da karşımıza çıkmaktadır. CS-DA her bir cümle için tek bir varlık hakkındaki fikirleri ifade ettiğini varsayar, bu durum her ne kadar DS-DA kadar problemlidir olsa da yine de analiz başarımlarını etkileyecektir (Pallavicini, Ciproso & Mantovani, 2017).

Özellik Seviyesi Duygu Analizi (OS-DA): İnce Taneli (Fine-Grained) DA (Sun vd., 2017) olarak ta adlandırılan bu analiz seviyesinde her bir dokümandaki her bir duygu belirlendikten sonra

ve duygu hedefi ve ilgili varlık tespit edilir (Pradhan vd., 2016; Liu, 2020). Bu şekilde farklı fikirler içeren doküman ve cümlelerde karşılaşılan başarımları problemleri elemine edilecektir. Fakat anlaşılacağı üzere gayet zorlu bir analiz süreci gerektirir.

9.3.2 Duygu Analizi Süreci

Modern veri analizi süreçleri genelde madencilik (mining) ifadesi ile birlikte isimlendirilirler. Çünkü Veri Madenciliği, Metin Madenciliği, Web Madenciliği gibi bütün bu yöntemlerde temel veri kaynağı çok büyüktür ve içerisinden anlamlı verileri çekmek oldukça zordur. Bunun bir sonucu olarak herhangi bir madencilik yöntemi için evrensel kabul gören bir analiz süreci söz konusu değildir. Örneğin veri madenciliğindeki bilgi keşfi süreci firmadan firmaya, analistten analiste farklılık göstermektedir. Hatta bu durumun ortaya çıkarabileceği standart ve sistematik yöntem eksikliği SEMMA ve CRISP Veri madenciliği gibi standartlaştırılmış süreçlerin ortaya çıkmasını doğurmuştur (Sharda vd., 2014). DA çalışmalarında da durum farklıdır. Her bir DA yaklaşımı kendine has sıralı süreçleri içermektedir. Bu çalışmada genel olarak sıklıkla tercih edilen adımları içeren bir model üzerinde durulacaktır. Şekil 9.2 de bu çalışmada tercih edilecek DA süreci görselleştirilmiştir.



Şekil 9.2: Duygu Analizi Süreci

Şekil 9.2’den de anlaşılacağı üzere ilk adımda veri seti çeşitli veri kaynaklarından toplanarak oluşturulur. Fakat bu veriler ham haldedir ve analiz öncesinde bir takım ön işleme işlemlerinden geçirmelidir. Analize hazır hale gelen temiz veri ise öznitelik çıkarma ve seçimi aşamalarında sayısal hale döndürülür. Duygu sınıflandırma aşamasında ise metne duygu skorları eklenerek son aşamadaki sonuçların görselleştirilmesi ve yorumlanması aşamasına gönderilir.

9.3.3 Duygu Analizi Süreci

Duygu analizine konu veri kaynağı için tek gerek şart duygu içeren öznel dokümanlar olmasıdır. Literatürde genelde DA metin verisi üzerinde yapılan çalışmaları işaret etmektedir. Diğer duygu ifade eden dokümanlar üzerinde yapılan analizler daha önce de belirtildiği üzere çok modlu DA olarak adlandırılırlar.

Veri çekme veya veri çıkarma işi analiste ve alana bağlı olarak gerçekleşen göreceli bir süreçtir. DA çalışmalarında müşteri yorumları ve twitter mesajları genelde tercih edilen veri kaynaklarıdır. Müşteri yorumlarını elde etmek için genelde scrapper denilen özel kodlar kullanılır. Twitter verisi elde etmek için ise geliştirici hesabı sahibi olmak ön koşulu ile Twitter API olarak adlandırılan özel programcıklar tercih edilir. Bazı özel analitik yazılımlar da #hashtag araması yardımıyla veri çıkarımına müsaade etmektedirler. Benzer şekilde diğer sosyal medya platformlarında veri çekmek için de farklı API ve arayüzler kullanılabilir (Ravi & Ravi, 2015).

Duygu Analizi Süreci

Veri önışleme süreçleri bütün veri analizi süreçleri için hayati önem taşıyan bir adımdır (Sharda vd., 2014). Bu adım genelde analiz süreçlerinin en uzun adımı olmakta ve yapılan hataların geri dönüşü en ağır şekilde gerçekleşmektedir. Bu bağlamda metinlerde aşağıdaki işlemler gerçekleştirilmeden analizlere başlanmamalıdır (Haddi, Liu & Shi, 2013).

- Metin içerisindeki bütün büyük harflerin küçük harflere döndürülmesi
- Noktalama işaretlerinin temizlenmesi
- Durdurma kelimelerinin temizlenmesi
- Numaraların kaldırılması
- Gereksiz boşlukların silinmesi

Yukarıda ifade edilen işlemler metin madenciliği sürecinin doğru şekilde yürütülmesi için elzemdir. Fakat bazı önışleme süreçleri analiz konusuna bağlı olarak gerçekleştirilir. Spesifik kelimelerin (küfürler vb.) derlemden kaldırılması ve tek başına anlam içermeyen kelimelerin elemine edilmesi (stemming) bazı metin madenciliği çalışmalarında tercih edilmektedir. DA ise aslında internetten toplanan metinler üzerinde çalışan bir yaklaşım olduğundan bu süreçlerin hepsi bir şekilde DA önışleme süreci içerisinde gerçekleştirilir. Fakat eğer DA sosyal medya, özellikle twitter verisi üzerinde yapılacaksa ek adımların da sürece dahil edilmesi gerekmektedir (Krouska, Troussas & Virvou, 2016).

- Http, @, #, gt gibi twitter verisine özgü karakterlerin temizlenmesi
- Kısaltmaların kaldırılması veya duygu ifade ediyorsa bir jargon sözlüğü yardımı ile kelimelere dönüştürülmesi
- Emojilerin kelimelere dönüştürülmesi

Kısaltmalar ve emojiler çoğu tweet için olmazsa olmaz nesnelere ve bu nesnelere zaman zaman polaritesi çok yüksek duygulara karşılık gelebilmektedir. Örneğin “lol” kısaltması “yüksek sesle gülmek” anlamına gelmekte ve yüksek pozitif duygu ifade etmektedir. Diğer taraftan ☺, ☹, gibi emojiler duygu ifade ettiğinden “keyifliyim” veya “üzgünüm” gibi ifadelere, hatta arka arkaya kullanımda “çok mutluyum”, “çok üzgünüm” gibi yüksek polariteli ifadelere çevirilirler. Özellikle CS-DA çalışmalarında bu tür dönüşümler önem arz etmektedir.

Önışleme analiz süreci içerisinde görece en önemli kısımdır. Bu anlamda bu süreç değerli öznel belirlenmesi ve çıkarılması, bu öznel belirlenmelerin ağırlıklandırılıp sayısal ifadeler dönüştürülmesi ve duruma göre topiklere atanması gibi adımları da içerebilir. Fakat bu adımların önemlerini ortaya koymak adına farklı başlıklar altında değerlendirmek uygun olacaktır.

Öznitelik Çıkarımı ve Seçimi

Son yıllarda kullanılan bütün modern veri analizi tekniklerinde olduğu gibi DA yaklaşımı da çoğu zaman büyük miktarda veri ile başa çıkmaktadır. Metin madenciliği ve sosyal medya analitiği çalışmalarında da durum farklı değildir. Bu verilerin analizler ile anlamlandırılabilmesi için farklı özniteliklerle etiketlenmesi gerekmektedir. Özellikle doğal dil işleme çalışmalarında konuşma bölümü (Part of Speech), n-gram (kelime ve kelime grupları), olumsuzluk ekleri, semantik bağımlılık, duygu sözcükleri vb öznitelikler yöntemin başarımı için çok önemlidir. Bütün bu öznitelikler için seçim (ağırlıklandırma veya skorlama) DA süreçlerindeki kritik adımlardandır.

Öznitelik seçimi için kullanılan yöntemler temelde metni sayısallaştırarak değerlendirirler. Çünkü birçok makine öğrenmesi ve doğal dil işleme tekniği sayısal ifadeler ile çalışmaktadır. Üç farklı başlık altında değerlendirilirler.

Filtreleme Yöntemleri: Bu yöntemlerde öznitelik ile ilgilenilen değişken arasındaki ilişkiyi hesaplayarak bir filtreleme gerçekleştirilir. Yöntemler sonucunda belirli eşik değerler ile seçim yapılarak bir alt küme oluşturur. Böylece nispeten daha az değerli olan değişkenler ve ifadeler atılarak kabul edilebilir boyutta bir matris ile yola devam edilir. Literatürde birçok filtreleme tekniği bulunmaktadır. En sık kullanılan teknikler aşağıda sıralanmıştır (Li, Xia, Zong & Huang, 2009; Haddi vd., 2013; Varela, Martins, Aquiar & Figueiredo, 2013; Hung, Alfred & Ahmad Hijazi, 2015; Yousefpour, Ibrahim & Hamed, 2017; Poria vd., 2018; Sailunaz vd., 2018; Yue vd., 2019)

- Bilgi Kazanımı (Information Gain (IG))
- Ki Kare Skoru (Chi Square)
- Karşılıklı Bilgi (Mutual Information (MI))
- Doküman Frekansı (Document Frequency (DF))
- Terim Frekansı (Term Frequency (TF))
- Terim Frekansı – Ters Doküman Frekansı (Term Frequency-Inverse Document Frequency (TF-IDF))
- Bi-Normal Ayırma (Bi-Normal Separation (BSN))
- Ağırlıklı Logaritmik Olasılıklı Oran (Weighted Log Likelihood Ratio (WLLR))
- Risk Oranı (Odds Ratio (OR))
- Korelasyon Katsayısı (Correlation Coefficient (CC))
- Gini İndeksi (GINI Index)

Sarmalayıcı Yöntemler: Bu yöntemlerde temelde alt modeller oluşturulur. Alt modellerin her birinin kendine has öznitelik varyasyonları vardır. Öznitelik seçimi ise alt modellerin performanslarının karşılaştırmalı değerlendirilmesi ile gerçekleştirilir. Aşağıda sarmalayıcı (wrapper) yöntemlerinden en sık kullanılanları verilmiştir (Budak, 2018; Çetin & Eryiğit, 2018).

- Kümeleme (Clustering)
- Sıralı İleri Yönlü Seçim (Sequential Forward Selection – (SFS))
- Sıralı Geri Yönlü Seçim (Sequential Backward Selection – (SBS))
- Yinelemeli Öznitelik Eleme (Recursive Feature Elimination – RFE)
- L+ R- (Plus L, Minus R)

Gömülü Yöntemler: Makine öğrenmesi tekniklerinin optimizasyon yeteneklerinin kullanıldığı tekniklerdir. Bununla birlikte bazı regresyon temelli tekniklerde kullanılabilir (Budak, 2018).

- Karar Ağaçları
- Genetik Algoritmalar
- Destek Vektör Makineleri
- Lasse Regresyon
- Ridge Regresyon

Topik Modelleme

Metin madenciliği derlemlerinde çoğu zaman birden fazla alanda içerikler sunulabilmektedir. Örneğin bir cep telefonu modeli le ilgili olarak toplanan bütün yorumlardan oluşan bir derlemde gruplamaya uygun birçok alt topik olabilir. Analiz gereksinimlerine göre kullanım kolaylığı, görsellik gibi özellikler ekseninde derlem kategorilere ayrılabilceği gibi ekran, batarya ömür, hız gibi teknolojik eksenli kategoriler de tercih edilebilir. Bu şekildeki kategorilere ayrılma işlemi topik modelleme olarak adlandırılır. Topik modelleme yaklaşımları esasen derlem ve çalışma alanına bağlı olarak tercih edilebilen, her DA sürecinde dâhil edilmeyen yaklaşımlardır. Özetlemenin elzem olduğu çok büyük veri setlerinde topiklerin hatta alt topiklerin oluşturulması çoğu zaman derlem ile ilgili daha ayrıntılı bakış açıları sunmaktadır.

Özetleme amacıyla ele alınan derlemde analiz öncesinde belirli olan bazı konu ve alt konulara atama yapılmak isteniyorsa denetimli öğrenme eksenli makine öğrenmesi teknikleri tercih edilmektedir. Bu amaçla hemen hemen bütün makine öğrenmesi teknikleri önceden öznitelik seçimi aşamasında sayısallaştırılmış veri üzerinde çalışabilir. Fakat kullanıcı genel durumu görmek adına önceden grupları belirlemek isteniyorsa, denetimsiz öğrenme yaklaşımlarından Gizli Dirichlet Tahisi (Laten Dirichlet Allocation LDA), Olasılıklı Gizli Semantik İndeksleme (Probabilistic Latent Semantic Indexing – pLSI) gibi teknikleri kullanabilmektedir (Wang & Zhai; 2017; Liu, 2020).

Duygu Sınıflandırma

Duygu sınıflandırma veya duygu skorlama DA sürecinin kritik adımlarındandır. Bu adımda makine öğrenmesi veya indeks tabanlı yaklaşımlar ile her bir duygu ifadesi varlık, cümle ve doküman seviyelerinde bütünleştirilerek skorlanırlar. Skorlama genelde negatif ve pozitif olarak yapıldığından bir sınıflandırma problemi olarak düşünülebilir (Liu, 2020). Sınıflandırma problemlerinde çıktı değişkeninin ikili olması zorunluluğu yoktur. Özellikle müşteri yorumları içeren derlemlerde sınıflandırmanın 1-5 yıldız arasında yapıldığı düşünüldüğünde DA çalışmalarında da polaritenin yönüne, gücü de eklenebilmektedir. Hem makine öğrenmesi hem de indeks tabanlı yaklaşımlar farklı duygu skorları hesaplamada kullanılabilir.

Duygu sınıflandırma yaklaşımları polarite eksenli değerlendirildiğinde duygunun ne olduğuna odaklanmak yerine duygunun negatif ve pozitif olduğunu inceler. Fakat gerçek hayatta duygular farklı şekilde olmaktadır. Özellikle insanların ruh hali ve duygularının birçok grup ve alt grubu olduğu düşünüldüğünde (korku, eğlence, merak, endişe vb.) sınıflandırma çok kolay bir süreç değildir (Nissim & Patti, 2017; Liu, 2020). Bu anlamda interdisipliner bir yaklaşım olan DA özellikle psikoloji ve davranış bilimlerinden yoğun destek alarak süreci yürütebilir.

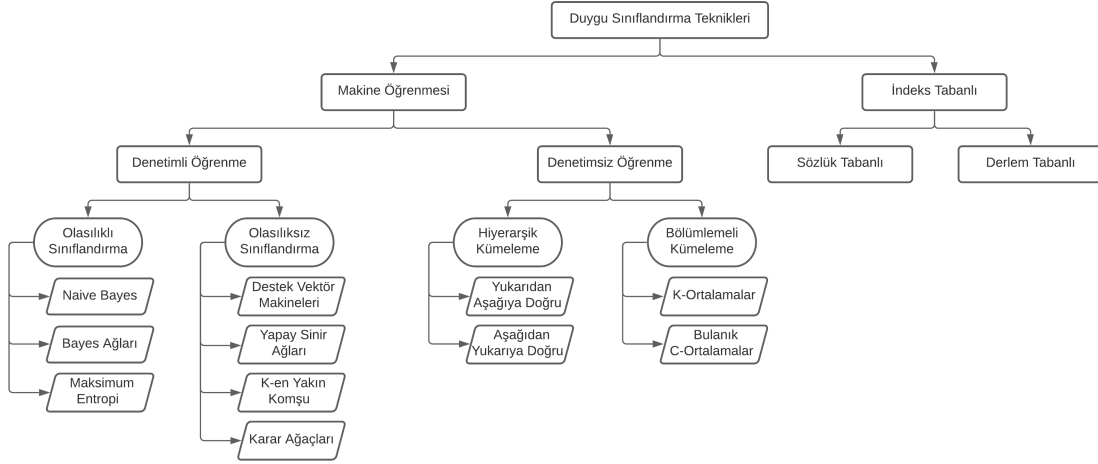
Çalışmanın geri kalanında klasik polarite eksenli DA duygu sınıflandırma yaklaşımları değerlendirilecektir. Şekil 9.3 te bu teknikler ile ilgili bir kategorizasyon sunulmuştur (Hemmatian & Sohrabi, 2019).

Makine Öğrenmesi Teknikleri

Makine öğrenmesi teknikleri öğrenme sürecinde dışarıdan yapılan etkiye bağlı olarak denetimli ve denetimsiz olarak sınıflandırılırlar.

Denetimli Makine Öğrenmesi Teknikleri: Bu tekniklerde duygu polariteleri ve kelimeler içeren data setleri diğer öznitelikler ile birlikte makine öğrenmesi yöntemine sunulurken öğretilir. Öğrenilen modelin başarı düzeyi değerlendirilip yeni derlem için duygu sınıfları tahmin edilir. Olasılıklı ve olasılıksız olmak üzere iki alt sınıfta değerlendirilirler (Hemmatian & Sohrabi, 2019).

- **Olasılıklı Denetimli Öğrenme:** Bu yaklaşımda olasılık temelli modeller ile duygu sınıflandırması gerçekleştirilir. Özellikle doğal dil işleme çalışmalarında sıklıkla tercih edilmektedir.
 - Naive Bayes: En sade ve temel yaklaşımlardan birisidir. Kelimelerin şartlı olasılıkları



Şekil 9.3: Sık kullanılan duygu sınıflandırma tekniklerinin bir görünümü

dikkate alınarak eğitim setine benzerlikleri hesaplanır.

- Bayes Ağları: Birbiri ile ilişkili rassal değişkenler eksenli görsel bir ağ yapısı ile sınıflandırma gerçekleştirir.
- Maksimum Entropi: Her bir öznitelik için hesaplanan entropi değerlerinin öğrenme seti ile karşılaştırılarak maksimumluk durumuna göre sınıflandırma yapar. Bayes yöntemlerinden farklı olarak öznitelikler arasında ilişki varlığı varsayımını kullanmaz.
- Olasılıksız Denetimli Öğrenme: Olasılığa dayalı matematiksel hesaplamaların yapılmadığı sınıflandırma algoritmalarıdır.
 - Destek Vektör Makineleri: En çok tercih edilen ve genelde başarıyı en yüksek duygu sınıflandırma algoritmasıdır. Doküman seviyesi analizlerde tercih edilen yöntem düzlemsel bir kesit ile sınıfları ayırır ve doğrusal hesaplamalar ile sınıflara atama işlemi gerçekleştirilir.
 - Yapay Sinir Ağları: Özniteliklerin lineer girdi seti olarak kullanıldığı duygu sınıflarını ise karmaşık ağ yapısı iteratif bir şekilde tahmin eden bir yapay zekâ yaklaşımıdır.
 - K en Yakın Komşu Algoritması: Sayısallaştırılmış vektör uzayında uzaklık eksenli olarak sınıflandırma gerçekleştiren öğrenme temelli algoritmadır.
 - Karar Ağaçları: Kesikli değerde verilen amaç değerine ulaşmak için endüktif öğrenme temelli yaklaşım kullanan bir sınıflandırma tekniğidir. Duygu polaritelerinin ikili olduğu durumlarda tercih edilebilen bir yaklaşımdır.

Denetimsiz Makine Öğrenmesi Teknikleri: Denetimsiz öğrenme, kümeleme adı altında anılan tekniklere verilen genel bir isimdir. Bu yaklaşımlarda herhangi bir öğrenme setine bağlı kalmaksızın tamamen derlem odaklı olarak bölümlenme gerçekleştirilir (Sharda vd., 2014). Duygu sınıflandırmada kullanılan kümeleme yaklaşımları iki alt sınıfta değerlendirilir.

- Hiyerarşik Kümeleme: Her bir sınıf hiyerarşik yapıda bağlı varlık ekseninde belirlenir. Yani her bir kümenin alt kümeleri onlarında yine alt kümeleri olabilir.
 - Yukarıdan Aşağıya Doğru Kümeleme: Başlangıçta tek bir küme ele alınır. Tümdengelim yaklaşımı ile her adımda hesaplamalar ile alt kümeler belirlenir. Bu yöntem bölücü kümeleme olarak ta adlandırılır.

- Aşağıdan Yukarıya Doğru Kümeleme: Yığınsal (agglomerative) kümeleme olarak ta adlandırılabilen yaklaşımda her varlık bir küme olacak şekilde başlanır. Sonrasında her adımda birleştirme yapılarak üst kümelere oluşturulur.
- Bölümlemeli Kümeleme: Bu kümeleme yaklaşımında bütün varlıklar tek bir kümeye atılır ve kümeler arasında bir üst üste gelme durumu söz konusu değildir. Bölümleme işlemi çoğunlukla uzaklığa bağlı bir benzerlik ölçütü ile gerçekleştirilir.
 - K-Ortalamalar: Matematiği oldukça sade olan bu yöntemde her bir duygunun belirlenen bir merkeze uzaklığı dikkate alınarak atama gerçekleştirilir. Merkez her adımda tekrar hesaplanır ve yüksek performansa ulaşmaya çalışılır. Yeniden hesaplama adımları hesaplama sonucunda merkez değişmediğinde durdurulur.
 - Bulanık C-Ortalamalar: Bir değişkenin birden fazla sınıfa ne ölçüde atanabileceğinin üyelik fonksiyonu yardımıyla matematiksel hesabını yapabildiği için özellikle doğal dil işleme tabanlı DA çalışmalarında kullanılmaktadır.

Denetimli öğrenme teknikleri öznitelik veya varlıkların önceden etiketlenmesi ile öğrenirler. Pratikte bu süreç çoğu zaman yorucu, zaman alıcı, zor ve maliyetlidir. Ayrıca bu şekildeki bir etiketleme tamamen alan bilgisi seviyesi ile de alakalıdır. Denetimli öğrenme tekniklerinin avantajlarını kullanmayı amaçlayan kendi kendine öğrenme (Self Training), birlikte öğrenme (Co-Training), çok görünümlü öğrenme (Multi View Training), grafik tabanlı öğrenme (Graph-Based Learning) gibi yaklaşımlar son yıllarda ön plana çıkmaktadır. Yarı-denetimli öğrenme olarak sınıflandırılan bu modellerde çok küçük eğitim setleri ile başlayan öğrenme süreçleri adım adım yüksek başarıya ulaştırılmaya çalışılır (Hemmatian & Sohrabi, 2019).

İndeks Tabanlı Duygu Sınıflandırma

Duygu sınıflarının belirli bir sözlük veya indeks yardımıyla gerçekleştirildiği bir yöntemdir (Sailunaz vd., 2018; Hemmatian & Sohrabi, 2019). Bu yaklaşımda başlangıçta duygu sözcükleri ve bu sözlüklerin polarite veya duygu skorlarının olduğu bir liste söz konusudur. Daha sonra metin içerisindeki her bir sözcük indeks tabanlı olarak etiketlenir ve çeşitli hesaplamalar yardımıyla özellik, cümle veya doküman bazlı olarak değerlendirilir. Başlangıçta hazırlanan listenin oluşturulma şekline bağlı olarak iki ana başlıkta değerlendirilir.

Sözlük Tabanlı Duygu Sınıflandırma: Duygu ve fikir ifade eden kelimeler DA çalışmalarının odak noktasıdır. Eğer bir kelimenin duygu polaritesi yönü (pozitif ve negatif) ve hatta şiddeti bilinebilirse o zaman kelimeleri bu değerler ile etiketleyerek yapılan hesaplamalar ile DA gerçekleştirilebilir. Bu şekildeki önceden hazırlanmış, içerisinde sözcük ve tamlamalar ile duygu değerleri yer alan sözlükler yardımıyla duygu sınıflandırması yapılabilmektedir. Özellikle İngilizce dilinde çok geniş kapsamlı sözlüklere rastlanabilmektedir. Senticnet, SentiWordNet, SentiSlangNet, So-Cal ve Sentiment Treebank DA çalışmalarında sıklıkla tercih edilen popüler sözlüklerdir (Joshi, Bhattacharyya & Ahire, 2017; Hemmatian & Sohrabi, 2019).

Derlem Tabanlı Duygu Sınıflandırma: Metin madenciliği ve bir alt dalı olarak DA için toplanan bütün veri seti “Derlem” olarak adlandırılır. Derlem tabanlı yaklaşımda da yine duygu sözlüğü kullanılır. Burada kullanılan sözlükler genel kabul görmüş, popüler sözlükler değil, tamamen derlem tabanlı olarak türetilmiş alan bilgisinin yoğunluğu yüksek olan analize özel listelerdir. Sözlükleri oluştururken çoğu zaman SentiWordNet gibi geniş kapsamlı sözlüklerdeki polarite değerleri esasa alınır. Fakat bu değerler sadece derlemden gelen kelimeler ve tamlamalar için atanır. Ayrıca derlem içerisinden gelen, popüler sözlükte yer almayan ifadeler de derlem tabanlı sözlüklere eklenebilir. Geri kalan DA sınıflandırma işlemleri sözlük tabanlı yaklaşımlar ile aynıdır.

Sosyal medya verileri geleneksel cümle yapılarına ve dilbilgisi kurallarına uymayan, birçok metin dışı nesne ile nispeten kirli, yapısal formda olmayan içeriklerdir (Zafarini vd., 2014; Fersini,

2017; Varma vd., 2017). Bu yüzden parçalara ayırma (tokenazing), cümle parçalarını etiketleme (part of speech tagging) ve köke inme (stemming) gibi ön işleme süreçlerinde problemlili olabileceğinden doğal dil işleme yaklaşımı ile değerlendirmesi çok başarılı olmamaktadır. Bu durumda indeks tabanlı, kelime ve tamlama odaklı olan yaklaşımlar DA çalışmalarında çok değerli olmaktadır.

Daha öncede ifade ettiğimiz gibi duyguların polaritesi yerine direkt duyguların kendisinin ele alındığı analizler de duygu sınıflandırma alanında yer almaktadır. Bu analizler de yine makinem öğrenmesi yerine çeşitli sözlükler yardımıyla gerçekleştirilebilir. Bu tip sözlükler emoticon olarak adlandırılır ve her kelimenin hangi duyguya karşılık geldiğini içerirler (Joshi vd., 2017).

Sonuçların Görselleştirilmesi ve Yorumlama

DA çıktılarının rahat anlaşılabilir olması nedeniyle işletmelerde her türlü kontrol seviyesinde görselleştirme araçları sayesinde destek sağlamaktadır. Bu görselleştirme araçları duygu sınıflandırma çıktısına bağlı olarak değişebilir. Örneğin duygu skorları kesikli polarite olarak elde edildi ise (Pozitif, Negatif veya Nötr) en sık kullanılan görselleştirme grafikleri sütun ve çubuk grafikleri, emoji destekli görselleştirme, pasta grafiğidir. Bütün bu grafikler için temel renk seçimi çoğu zaman kırmızıdan yeşile dönen dinamik skala olarak tercih edilir. Eğer sınıf çıktısı sürekli ise bu durum kelime bulutları, harita grafikleri, alan tarama grafikleri gibi yöntemler ön plandadır. Bütün bu grafiklerin ortak özelliği kategorilerle özetlemelere müsaade eden, kolay anlaşılır olmalarıdır ki iş zekası grafiklerinin bütün özelliklerini taşırlar. Bu açıdan bakıldığında son yıllarda ön planda olan kullanım kolaylığı ve görselleştirme kabiliyetleri ile ön planda olan Tableau, QlikView ve PowerBI gibi yazılımlarda hazırlanan gösterge panelleri pazarı, müşteriye (iç ve dış), ürünü, markayı değerlendirmede tercih edilmektedirler.

9.3.4 Duygu Analizinin Önündeki Engeller

DA politikadan, sağlığa, eğitime kadar birçok alanda çok değerli bakış açıları sunabilmesi ile son yıllarda ön planda olan bir yaklaşımdır. Fakat derlem ve yöntem eksenli bazı sorunlar DA çalışmalarının önünde bir engel olarak durmaktadır. İğneleyici ifadeler ve ironiler: Bütün DA çalışmalarında kelime ve tamlamalar üzerine yapılan çalışmalar genelde kelimenin birincil manası ilgilenir. Fakat gerçek hayatta bu durum biraz farklıdır. Deyimler, alaycı ifadeler, ironiler gibi kelime ve tamlamalara birincil manasında çok daha farklı anlamlar yükleyebilen durumlar analiz başarımlını doğrudan etkileyebilmektedir (Pradhan vd., 2016; Pozzi vd., 2017; Poria vd., 2018; Karoui, Zitoune & Moriceau, 2019; Liu, 2020).

Alan Bağımlılık: Bazı duygu kelimeleri farklı alanlarda farklı polaritelere sahip olabilirler. Örneğin “Bugün hava çok sıcak, adeta nefes alamıyorum” ve “Sıcak satış yeteneği ile ön plana çıkıyor” cümlelerindeki sıcak kelimesi ilk cümlede negatif polarite ifade ederken ikinci cümlede nötr bir polarite göstermektedir (Pradhan vd., 2016; Karoui, Zitoune & Moriceau, 2019; Liu, 2020).

Duygu Sözcüğü Yokluğu: Bazı cümleler gerçekte herhangi bir duygu sözcüğü içermese de genel manası ile bir duygu ifade edebilir. “Bu plazma TV fazlasıyla elektrik kullanıyor” cümlesinden aslında herhangi bir duygu sözcüğü yer almamakta ama cümleyi kuran kişi plazma TV ile ilgili olumsuz görüşünü ifade etmektedir (Pradhan vd., 2016; Poria vd., 2018; Karoui, Zitoune & Moriceau, 2019; Liu, 2020).

Karmaşık Cümle Yapısı: Sosyal medya mesajları çoğu zaman devrik ve cümle yapısına uymayan ve yarım kalmış metinlerdir. Bu tip metinlerin doğal dil işleme çalışmalarında zorluk çıkarabileceği öngörülebilir. Sözlük temelli yaklaşımlar ise kelime odaklı olduğundan karmaşık cümle yapısı durumunda nispeten daha az etkilenecek, daha iyi sonuçlar üretirler (Sharda vd., 2014).

Gürültülü Veri: DA temel veri kaynaklarından en önemlisi sosyal medyadır. Sosyal medya

verileri ise farklı karakterler, emojiler, linkler, resimler, gramere uymayarak yazılmış ifadeler, tekrarlı ifade ve karakterler gibi sorunlarından dolayı yüksek gürültülü veri olarak adlandırılırlar. Bu şekilde gürültülü verileri analiz öncesinde temizleme ve düzeltme işlemleri göreceli olarak DA sürecinin en değerli adımındır (Pradhan vd., 2016; Poria vd., 2018).

Olumsuzluk Ekleri: Bazı olumsuzluk yapan ekler cümle yapısında her ne kadar olumlu duygu kelimeleri olsa da cümlenin genel manasını direkt olarak çevirebilir. “Birçok harika özelliği ile dünyanın en iyi telefonları iphone modelleri olsa da kesinlikle bir tane satın almam” Doğal dil işleme yaklaşımlarında çözülebilecek bir problem olsa da, indeks tabanlı yaklaşımlarda çok fazla olumlu polariteli duygu ifadesi ile bu cümle olumlu olarak addedilebilir (Pradhan vd., 2016).

9.3.5 Türkçe Metinlerde Duygu Analizi

DA gerek sözlük tabanlı olarak gerekse makine öğrenmesi sınıflandırma yöntemleriyle birçok dilde başarıyla kullanılan bir analiz yöntemidir. Türkçe dilinde yapılan çalışmaların sayısı son yıllarda Türkçe duygu sözlüklerinin artması ve python gibi gözde yazılım dillerine eklenen kütüphaneler yardımıyla hızla artmaktadır. Fakat özellikle Türkçe dilinin sondan eklemeli ve zengin biçimsel yapısı bu analizlerin başarımını etkilemektedir (Çetin & Eryiğit, 2018).

Doğal dil işleme çalışmalarının temelinde cümle bölümlenme (tokenazing) ve bağlantılı olarak yapılan cümle parçaları etiketleme (part of speech tagging) işlemleri Türkçe metinlerde yer alan öğelerin cümle içerisindeki dizilimlerinin farklı olması nedeniyle analiz sürecini zorlaştırmaktadır. Bu zorluk zaten biçimsel olarak çok problemlili ve gürültülü olan sosyal medya mesajlarında daha da ön plana çıkmaktadır.

Türkçe doğal dil işleme çalışmaları Google tarafından yönetilen BERT (Bidirectional Encoder Representations from Transformers) adlı çalışmanın Türkçe versiyonu TurkishBERT ile önemli bir aşama kaydetmiştir. Temelde cümleyi parçalara ayırıp iki yönlü olarak (soldan sağa ve sağdan sola) sıralı ilişkilere bakarak analiz yapabilen yaklaşım Türkçe metinlerde başarılı sonuçlar elde etmektedir. Fakat yukarıda belirtilen sorunlar dikkate alındığında verimli ön işleme süreçlerinin özellikle Türkçe dilinin özellikleri dikkate alarak çok titiz bir şekilde gerçekleştirilmesi gerekmektedir. Bu bağlamda sonraki bölümde TurkishBERT kütüphanesi yardımıyla yapılan bir duygu analizi çalışması adım adım sunulacaktır.

9.4 Türkçe Sosyal Medya İçerikleri ile Duygu Analizi

Çalışmamızın bu kısmında bir örnek uygulama yapılarak DA sürecinin pekiştirilmesi amaçlanmaktadır. Bu anlamda Türkçe içerikte olan sosyal medya verileri üzerinde yapılacak bir uygulama adım adım anlatılacaktır.

Verinin Elde Edilmesi

DA sürecinin ilk adımı verinin elde edilmesidir. Veri kaynağı olarak ta metin içerikleri ve popülerliği dikkate alındığında twitter sosyal medya uygulaması tercih edilmiştir. Güncel ve ilgi çekici bir konu üzerinde çalışılması düşünülmüş ve 2020 yılının ana gündem maddesi olan pandemi ile alakalı tweetlerin incelenmesine karar verilmiştir.

Twitter üzerinde veri çekebilmek için ön şart bir geliştirici (twitter developer) hesabına sahip olunmasıdır. Bu hesap twitter tarafından onaylandıktan sonra twitter API kullanan herhangi bir yazılım yardımıyla veri çekilebilmektedir. Bu çalışmada python yazılım diline ait Tweepy kütüphanesi bu amaçla kullanılmış ve Covid19 #hashtag’i ile veri çekilerek veri seti oluşturulmuştur. DA derlemi toplam 1713 adet tweet’ten oluşmuştur.

Veri Ön İşleme

Veri önışleme süreci bütün madencilik yazılımlarında olduđu gibi bir metin madenciliđi alt uygulaması olarak DA sürecinin de en kritik adımını oluřturmaktadır. Bu anlamda yapılacak her bir hatanın sonuçlarının analiz başarımını direkt etkileyebileceđi göz ardı edilmemelidir.

Ařađıda belirtilen adımlar önışleme sürecinde sırasıyla gerekleřtirilmiřtir.

- Veride bulunun bütün büyük harflerin küçük harflere dönüřtürülmesi
- Http, @, #, gt gibi Twitter verisinde sıklıkla rastlanan karakterlerin ve ifadelerin temizlenmesi
- Noktalama iřaretlerinin temizlenmesi
- Sayıların kaldırılması
- Durdurma kelimelerinin (StopWords) silinmesi

Bu alıřmada emoji ve kısaltmalar ön incelemede ok fazla olmadıđından ıkarılarak veriler temizlenmiřtir. Fakat bu tip unsurların yođun řekilde olduđu veri setleri söz konusu ise duygu ifade edebilen bu bileřenler analizlerde yararlı olabileceđinden bir sözlük yardımıyla dönüřtürülerek metne eklenebilirler.

Durdurma kelimeleri temizlenirken Türke durdurma kelimelerini ieren bařka bir kütüphane olan NLTK kullanılmıřtır. Veri seti üzerinde daha sonradan sayısallařtırma gerekleřtirileceđi ve nispeten düşük boyutlu olması durumu dikkate alındıđında frekansı düşük kelimelerin analiz dıřına alınması tercih edilmemiřtir. řekil 9.4 te önışleme süreci öncesindeki ham kirli veri ile temizlenmiř veri birlikte sunulmuřtur.

text	text
#tvjd2019\n1 yıl deđil sanki 5 yıl gemiř gibi...	yıl deđil yıl gemiř yılı bitmez oldu
#Covid19 Bilgi\n*Ülkelere Göre Corona Ölüm Art...	bilgi ülkelere göre corona ölüm artış sıralama...
#Covid19 Bilgi\n*Ülkelere Göre Corona Vaka Art...	bilgi ülkelere göre corona vaka artış sıralama...
Başbakan Rutte: umarım gerek kalmaz ancak kafe...	başbakan rutte umarım gerek kalmaz ancak kafe ...
Evden alıřırken yapılan 6 hata\n\n#Corona #Co...	evden alıřırken yapılan hata

řekil 9.4: Veri önışleme süreci öncesi ve sonrasındaki veri görünümü

Veri önışleme süreci veri seti hakkında verdiđi bakıř açıları ile analiz sürecinde farklı yöntem seđimleri konusunda ve özellikle topik modelleme veya özetleme gerekleřtirilecekse önemli destek sađlayabilir. Bu amaçla zaman zaman önışleme adımlarının sonunda veri setinin görselleřtirilmesi tercih edilmektedir. řekil 9.5 temizlenmiř veri setinden elde edilen kelime bulutunu betimlemektedir.

řekil 9.5'ten de anlaşılacađı üzere veri setinin odak noktası pandemidir. Ayrıca Hindistan ve Amerika gibi bazı ülkelerin de sıklıkla ifade edildiđi görülmüřtür. Bu durum ise bahse konu iki ülkenin dünyada en ok vaka sayısına ulařan ülkeler olması geređi ile açıklanabilir. Bunun dıřından sonrası için bir öngörü sađlayacak řekilde ıkarım yapılamamıřtır.

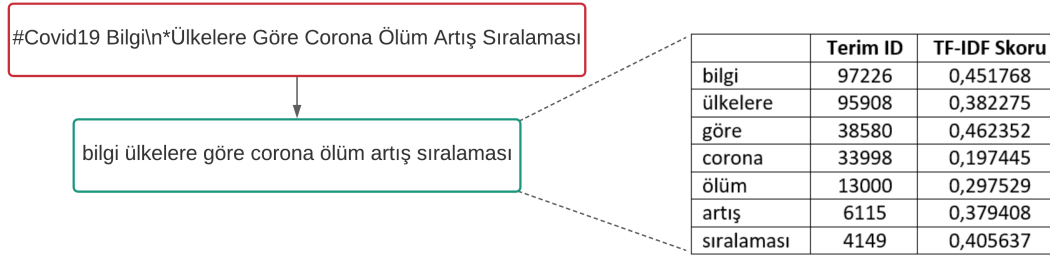
Öznitelik ıkarımı ve Seilmesi

Birok sınıflandırma algoritması yöntem varsayımlarından dolayı veri setini sayısal talep etmektedir. Ayrıca veri ile ilgili üst bilgi ieren özniteliklerin belirlenmesi ve bu nitelikler için sayısallařtırma dönüřümünün yapılması dođal dil iřleme alıřmalarında da kritik adımlardandır. alıřmamızda topik modelleme ve dođal dil iřleme duygu skorlarının hesaplanmasında farklı sayısal dönüřümler tercih edilmiřtir. Topik modelleme öncesi veri seti python makine öđrenmesi kütüphanesi olan sklearn yardımıyla sayısallařtırılmıřtır. Bu iřlem sırasında öznitelik seim yöntemlerinden Terim Frekansı



Şekil 9.5: Temizlenmiş veri ile elde edilen kelime bulutu

– Ters Doküman Frekansı (Term Frequency-Inverse Document Frequency (TF-IDF) algoritması kullanılmıştır.



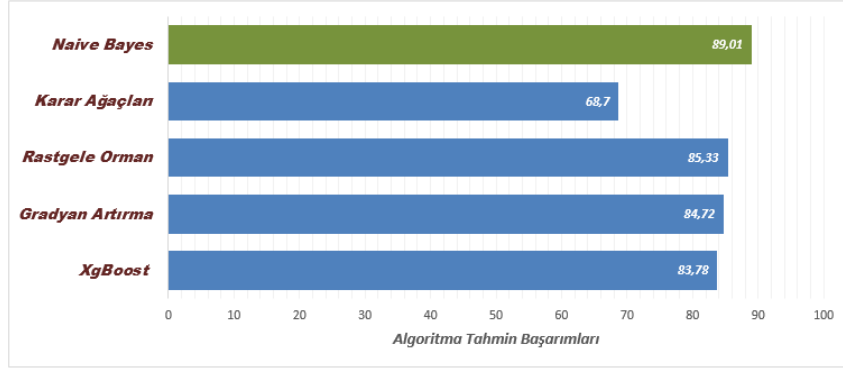
Şekil 9.6: TF-IDF sonrası sayısal veri gösterimi

Covid 19 ile ilgili metinlerin kategorilere ayrılması için hazır bir makine öğrenmesi veri seti tercih edilmiştir. Öğrenme veri seti dünya, ekonomi, kültür, sağlık, siyaset, spor ve teknoloji kategorilerinin her birisinde 700 adet olmak üzere toplam 4900 dokümandan oluşmaktadır. Önceden çekilerek, temizlenmiş olan 1713 dokümanın bu kategorilere atanması (topik modelleme) için denetimli öğrenme yöntemlerinden birisi tercih edilmelidir. Bu seçimi yapabilmek adına 4900 satırdan oluşan öğrenme veri seti %80 eğitim ve %20 öğrenme olmak üzere ikiye bölünmüş ve Naive Bayes, Karar Ağaçları, Rastgele Orman (Random Forest), Gradyan artırma (Gradient Boosting) ve XgBoost modelleri ile başarımları değerlendirilmiştir.

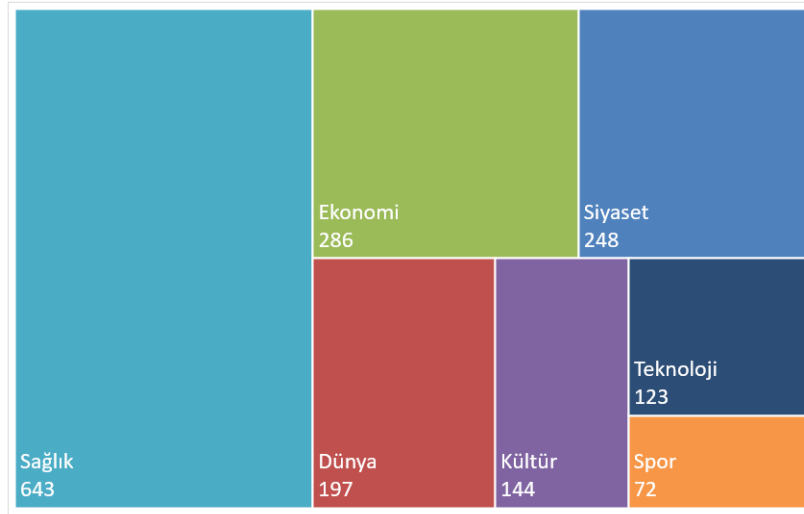
Modellerin başarımları sınıflandırma yaklaşımlarındaki en temel performans metriği olan karışıklık matrisi (Confusion Matrix) ve tahmin isabeti (% olarak) kullanılarak değerlendirilmiştir. Şekil 9.7 de beş farklı modelin test setindeki başarımlarını göstermektedir.

Şekil 9.7 en başarılı tahmin modeli olarak Naive bayes gösterdiğinden topik modelleme için bu yaklaşım kullanılmıştır. 1713 adet önceden temizlenip TF-IDF ile sayısallaştırılması gerçekleştirilen veri ile topik tahminleri yapılmış ve şekil 9.8 'de yer alan sonuçlar elde edilmiştir.

Yukarıdaki şekilde de net bir şekilde görülmektedir ki sağlık alanı en çok konuşulan alandır. Her ne kadar örneklemimiz küçük olsa da bize özellikle pandemi sürecinde insanların odak kayması yaşamadığını ve Covid 19 konusunda toplumsal bir bilincin olduğu konusunda önemli ipuçları



Şekil 9.7: Makine öğrenmesi modelleri tahmin performansları



Şekil 9.8: Veri setinin topik modelleme sonuçları

sunmaktadır. Fakat sadece topik eksenli özetleme yeterli olmayacaktır. Bu amaçla sonraki aşamada duygu analizi skorlarını hesaplama işlemi gerçekleştirilecektir.

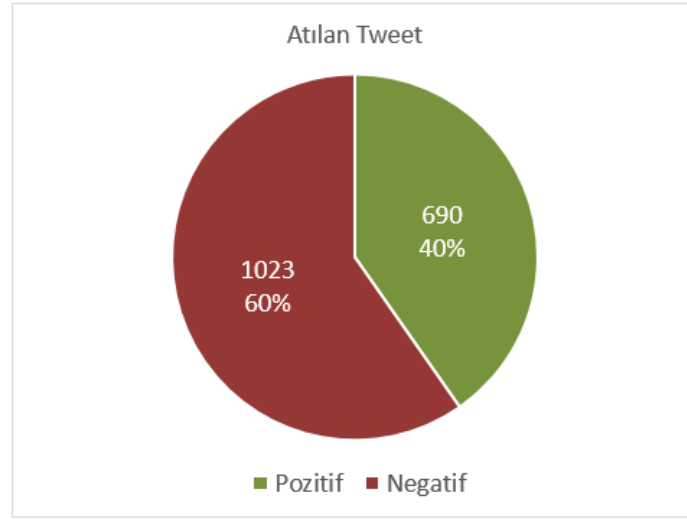
Duygu Skorlarının Hesaplanması ve Sonuçların Yorumlanması

Duygu skorları veya polariteleri hesaplamak için duygu sınıflandırma tekniklerinin kullanıldığından daha önce bahsedilmişti. Makine öğrenmesi temelli bu teknikler ile doğal dil işleme alanında çok önemli faydalar sağlanmıştır. Özellikle biçimsel özellikleri ile doğal dilin anlaşılması ve işlenmesi zor olan Türkçe dilinde son yıllarda başarılı yaklaşımlar ortaya atılmıştır. 2018 yılında Google tarafından duyurulan BERT (Bidirectional Encoder Representations from Transformers) doğal dil işleme modelinin Türkçe dilindeki versiyonu olan TurkishBERT başarılı yaklaşımlara en önemli örneklerdendir. Bu amaçla çalışmamızda bu doğal dil işleme modeli tercih edilmiştir.

BERT temelde cümleleri sağdan sola ve solda sağa doğru tarayarak kelimeler arası ilişkiyi iki yönlü aramayı amaçlar. Tarama işlemi sırasında ilişkiyi sentezlemek için ise transformer dediğimiz temelde İleri Beslemeli Sinir Ağları (Feed Forward Neural Networks) olan bir fonksiyon kullanır. Bu öğrenme sürecinde yüksek miktarda veriyi paralel ve seri şekilde işleyebilir. Temel öğrenme

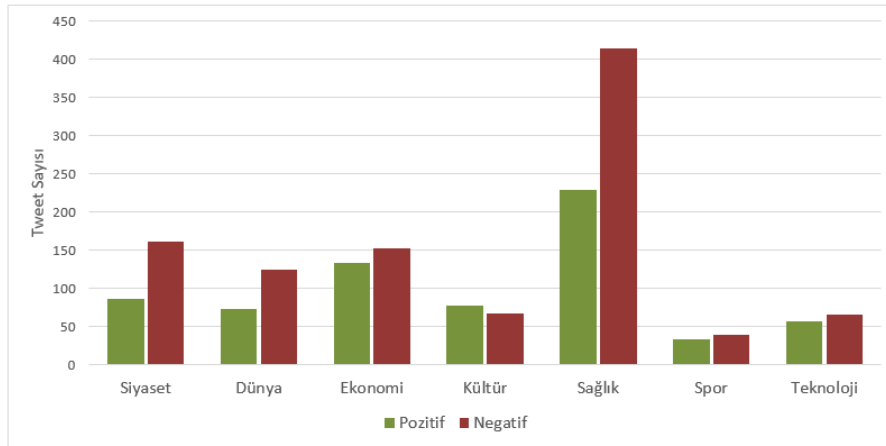
veri setinin de genişliği sayesinde (yaklaşık 35gb Türkçe metin) TurkishBERT yüksek başarımla elde edebilmektedir.

1713 veri TurkishBERT doğal dil işleme algoritması ile python yardımıyla değerlendirildiğinde toplam tweet'lerin %60'nın (1023 tane) olumsuz olduğu görülmektedir. Şekil 9.9'dan da anlaşılacağı üzere her bir doküman olumlu ve olumsuz kategorilerine atanmış ve nötr kategori kullanılmamıştır.



Şekil 9.9: Veri seti için duygu polaritesi dağılımı

Sonuçların daha ayrıntılı incelenebilmesi için topik modelleme aşamasındaki tahmin edilen kategoriler ile birleştirilmiştir. Şekil 9.10 bize sağlık, dünya ve siyaset topiğinde atılan tweet'lerin olumsuz yüzdelere daha belirgin olduğunu, buna karşın ekonomi, spor ve teknoloji ise daha homojen bir dağılım olduğu göstermektedir. Ayrıca kültür (eğitim de bu kategori içinde değerlendirilmektedir) kategorisinde olumlu bir görünüm olduğu da ortaya çıkmaktadır. Bu sonuçtan belki de bu süreçteki eğitim alanında uygulamaların olumlu karşılandığı bir çıkarım yapılabilir. Fakat bu tip çıkarımlar için daha alt seviye (cümle veya özellik) duygu analizlerini yapmak gerekebilir.



Şekil 9.10: Topiklere göre pozitif ve negatif tweet sayıları

Bir veri analizi sürecinde elde edilen sonuçların yorumlanması tamamen veri setinin içeriği ile alakalıdır. Zaman ve mekan değişkenleri gibi boyutlar, sonuçlardan daha iyi çıkarım yapılabilmesini sağlayabilir. Ayrıca sosyal medya verilerine özgü olan beğeni ve retweet gibi öznitelikler de farklı bakış açıları sunabilmektedir. Topik modellemede farklı kategori ve alt kategoriler kullanmak, analizlerin genelleme derecesini belirleyebilmektedir. Bütün bu unsurlar dikkate alındığında sosyal medya verilerinin çok daha detaylı değerlendirmelere açık olduğu söylenebilir. Bu çalışmanın temel amacı kapsamlı bir uygulama yapmaktan ziyade, Türkçe dilinde duygu analizi yaklaşımlarının geldiği noktayı göstermek olduğundan, nispeten daha sade bir problem tercih edilmiştir.

9.5 Sonuçlar

Teknolojinin geldiği nokta itibari ile her bireyin bir şekilde sosyal medya kullanıcı olduğu söylenebilir. Sosyalleşme isteği bilgisayar ortamında mahremiyet ve anonimlik gibi unsurlar ile daha rahat karşılanabilmektedir. Bu durumun doğal bir sonucu olarak sosyal medya üzerinden ürün, marka, bireyler hakkında değerlendirmelerde bulunurken insanlar sıklıkla birçok fikir ve duygulara yer verirler. Bu bilgilerin çekilmesi ve karar destek amaçlı kullanılmasını sağlayabilen duygu analizi özellikle 2000 yıllardan sonra araştırmacılar için önemli bir çalışma konusu olmuştur.

Duygu analizi yapısal ve zorlu bir süreçtir ve bu zorlu süreç çoğu zaman farklı doküman tipleri ve diller açısından bakıldığında daha da karmaşıklaşmaktadır. Bu çalışmada karmaşık olan duygu analizi süreci ile ilgili adım adım detaylı bilgiler sunularak araştırmacı ve profesyoneller için kılavuz niteliğinde bir içerik sunulmaya çalışılmıştır. Ayrıca Türkçe dilinde analiz sürecinin farklılıkları hem teorik olarak hem de gerçekleştirilen uygulama yardımıyla gösterilmiştir.

Verilerin elde edilmesi ve ön işleme süreci analize konu dokümanın türü ve biçimine göre farklılıklar göstermektedir. Son yıllarda müşteri yorumları ve twitter verileri en sık tercih edilen içerik türleridir. Özellikle sosyal medya verileri yapısal olarak sorunlu dokümanlardır. Bu tip sorunlarla başa çıkabilmek için ön işleme süreçlerinin sosyal medya verileri ile başa çıkacak şekilde güncellenmesi sonucunu doğurmaktadır. Öznitelik çıkarılması ve seçilmesi verilerin etiketlenerek sayısallaştırılması ile makine öğrenmesi gibi tekniklerin kullanılmasını sağlamakta ve duygu sınıflandırma sürecinin başarısını arttırmaktadır. Duygu skorları ise duygu sınıflandırma aşamasında hesaplanır. Fikirleri, karakter sayısı sınırları ile oldukça kısıtlı aralıkta sundukları için sosyal medya verileri (özellikle twitter) yorumlanması oldukça zor olan problemlidir. Özellikle yapısal olarak sorunlu olan bu dokümanların cümle ve nitelik seviyesindeki analizlerinin çoğu zaman yetersiz sonuç çıkardıkları görülmektedir. Bu yüzden doküman seviyesiyle analiz yapılması tercih edilen bir yöntemdir.

Covid 19 sürecinde insanların sosyal medya karşısında daha fazla vakit geçirdikleri ve daha çok e-ticaret yaptıkları düşünüldüğünde duygu analizine konu olan dokümanların da hızla arttığını söylemek yanlış olmaz. Bu süreçte bu tip dokümanlardan otomatik bilgi çıkarımını sağlayacak duygu analizi gibi modern analiz tekniklerinin önemini arttırdığı da muhakkaktır. Bütün bunlar dikkate alındığında özellikle sosyal medya analitiği konusunda gelecek yıllarda çalışma sayısının hızla artabileceği öngörülebilir.

Referanslar

Ak D. (2019). Pazarlama alanındaki uluslararası akademik dergilerin metin madenciliği yöntemi ile değerlendirilmesi.(Basılı olmayan Yüksek Lisans Tezi). Sakarya Üniversitesi İşletme Enstitüsü İşletme Anabilim Dalı, Sakarya

Bayrak, H., (2020, 23 Şubat). 2020 Türkiye İnternet Kullanımı ve Sosyal Medya İstatistikleri [Dijilopedi] Erişim Adresi: <https://dijilopedi.com/2020-turkiye-internet-kullanimi-ve-sosyal-medya-istatistikleri/>

BUDAK, H. (2018). Özellik Seçim Yöntemleri ve Yeni Bir Yaklaşım. *Journal of Natural & Applied Sciences*.

ÇETİN, F. S., & ERYİĞİT, G. (2018). Türkçe Hedef Tabanlı Duygu Analizi İçin Alt Görevlerin İncelenmesi–Hedef Terim, Hedef Kategori Ve Duygu Sınıfı Belirleme. *Bilişim Teknolojileri Dergisi*, 11(1), 43-56.

Fersini, E. (2017). Sentiment analysis in social networks: a machine learning perspective. In Pozzi, F.A., Fersini, E., Messina, E., & Liu, B. (Eds.) *Sentiment analysis in social networks* (pp. 91-111). Morgan Kaufmann.

Golder, S., O'Connor, K., Hennessy, S., Gross, R., & Gonzalez-Hernandez, G. (2020). Assessment of Beliefs and Attitudes About Statins Posted on Twitter: A Qualitative Study. *JAMA Network Open*, 3(6), e208953-e208953.

Haddi, E., Liu, X., & Shi, Y. (2013). The role of text pre-processing in sentiment analysis. *Procedia Computer Science*, 17, 26-32.

Hemmatian, F., & Sohrabi, M. K. (2019). A survey on classification techniques for opinion mining and sentiment analysis. *Artificial Intelligence Review*, 1-51.

Hung, L. P., Alfred, R., & Ahmad Hijazi, M. H. (2015). A review on feature selection methods for sentiment analysis. *Advanced Science Letters*, 21(10), 2952-2956.

Isson, J. P., & Harriott, J. (2013). *Pillars of Business Analytics Success*. Win with Advanced Business Analytics, 15.

Joshi, A., Bhattacharyya, P., & Ahire, S. (2017). Sentiment resources: Lexicons and datasets. In Cambria, E., Das, D., Bandyopadhyay, S., & Feraco, A. (Eds), *A Practical Guide To Sentiment Analysis* (pp. 85-106). Springer, Cham.

Karoui, J., Zitoune, F. B., & Moriceau, V. (2019). *Automatic Detection of Irony*. John Wiley & Sons, Incorporated.

Kou, F., Du, J., He, Y., & Ye, L. (2016). Social network search based on semantic analysis and learning. *CAAI transactions on intelligence technology*, 1(4), 293-302.

Krouska, A., Troussas, C., & Virvou, M. (2016, July). The effect of preprocessing techniques on Twitter sentiment analysis. In *2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA)* (pp. 1-5). IEEE.

Li, S., Xia, R., Zong, C., & Huang, C. R. (2009, August). A framework of feature selection methods for text categorization. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP* (pp. 692-700).

Liu, B. (2017). Many facets of sentiment analysis. In Cambria, E., Das, D., Bandyopadhyay, S., & Feraco, A. (Eds), *A Practical Guide To Sentiment Analysis* (pp. 11-39). Springer, Cham.

Liu, B. (2020). *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge university press. Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4), 1093-1113.

Nazir, A., Rao, Y., Wu, L., & Sun, L. (2020). Issues and Challenges of Aspect-based Sentiment Analysis: A Comprehensive Survey. *IEEE Transactions on Affective Computing*.

Nissim, M., & Patti, V. (2017). Semantic aspects in sentiment analysis. In Pozzi, F.A., Fersini, E., Messina, E., & Liu, B. (Eds.) *Sentiment analysis in social networks* (pp. 31-48). Morgan Kaufmann.

Pallavicini, F., Ciproso, P., & Mantovani, F. (2017). Beyond sentiment: how social network analytics can enhance opinion mining and sentiment analysis. In Pozzi, F.A., Fersini, E., Messina, E., & Liu, B. (Eds.) *Sentiment analysis in social networks* (pp. 13-29). Morgan Kaufmann.

Poria, S., Hussain, A., & Cambria, E. (2018). *Multimodal sentiment analysis* (Vol. 8). Cham, Switzerland: Springer.

Pozzi, F. A., Fersini, E., Messina, E., & Liu, B. (2017). Challenges of sentiment analysis in social networks: An overview. In Pozzi, F.A., Fersini, E., Messina, E., & Liu, B. (Eds.) *Sentiment analysis in social networks* (pp. 1-11). Morgan Kaufmann.

Pradhan, V. M., Vala, J., & Balani, P. (2016). A survey on sentiment analysis algorithms for opinion mining. *International Journal of Computer Applications*, 133(9), 7-11.

Ravi, K., & Ravi, V. (2015). A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-Based Systems*, 89, 14-46.

Sailunaz, K., Dhaliwal, M., Rokne, J., & Alhajj, R. (2018). Emotion detection from text and speech: a survey. *Social Network Analysis and Mining*, 8(1), 28.

Sharda, R., Delen, D., Turban, E., Aronson, J., & Liang, T. (2014). Business intelligence and analytics. *System for Decision Support*.

Singh, S. K., Paul, S., Kumar, D., & Arfi, H. (2014). Sentiment analysis of twitter data set: survey. *International Journal of Applied Engineering Research*, 9(22), 13925-13936.

Soleymani, M., Garcia, D., Jou, B., Schuller, B., Chang, S. F., & Pantic, M. (2017). A survey of multimodal sentiment analysis. *Image and Vision Computing*, 65, 3-14.

Sun, S., Luo, C., & Chen, J. (2017). A review of natural language processing techniques for opinion mining systems. *Information fusion*, 36, 10-25.

TÜİK (2020, Şubat). *Adrese Dayalı Nüfus Kayıt Sistemi Sonuçları, 2019* [Haber Bülteni]. Erişim Adresi: <https://data.tuik.gov.tr/tr/display-bulletin/?bulletin=adrese-dayali-nufus-kayit-sistemi-sonuclari-2019-33705>

Varela, P. L., Martins, A. F., Aguiar, P. M., & Figueiredo, M. A. (2013, May). An empirical study of feature selection for sentiment analysis. In 9th conference on telecommunications, Conftele, Castelo Branco.

Varma, V., Kurisinkel, L. J., & Radhakrishnan, P. (2017). Social Media Summarization. In Cambria, E., Das, D., Bandyopadhyay, S., & Feraco, A. (Eds), *A Practical Guide To Sentiment Analysis* (pp. 135-153). Springer, Cham.

Wang, H., & Zhai, C. (2017). Generative models for sentiment analysis and opinion mining. In Cambria, E., Das, D., Bandyopadhyay, S., & Feraco, A. (Eds), *A Practical Guide To Sentiment Analysis* (pp. 107-134). Springer, Cham.

Yousefpour, A., Ibrahim, R., & Hamed, H. N. A. (2017). Ordinal-based and frequency-based integration of feature selection methods for sentiment analysis. *Expert Systems with Applications*, 75, 80-93.

Yue, L., Chen, W., Li, X., Zuo, W., & Yin, M. (2019). A survey of sentiment analysis in social media. *Knowledge and Information Systems*, 1-47.

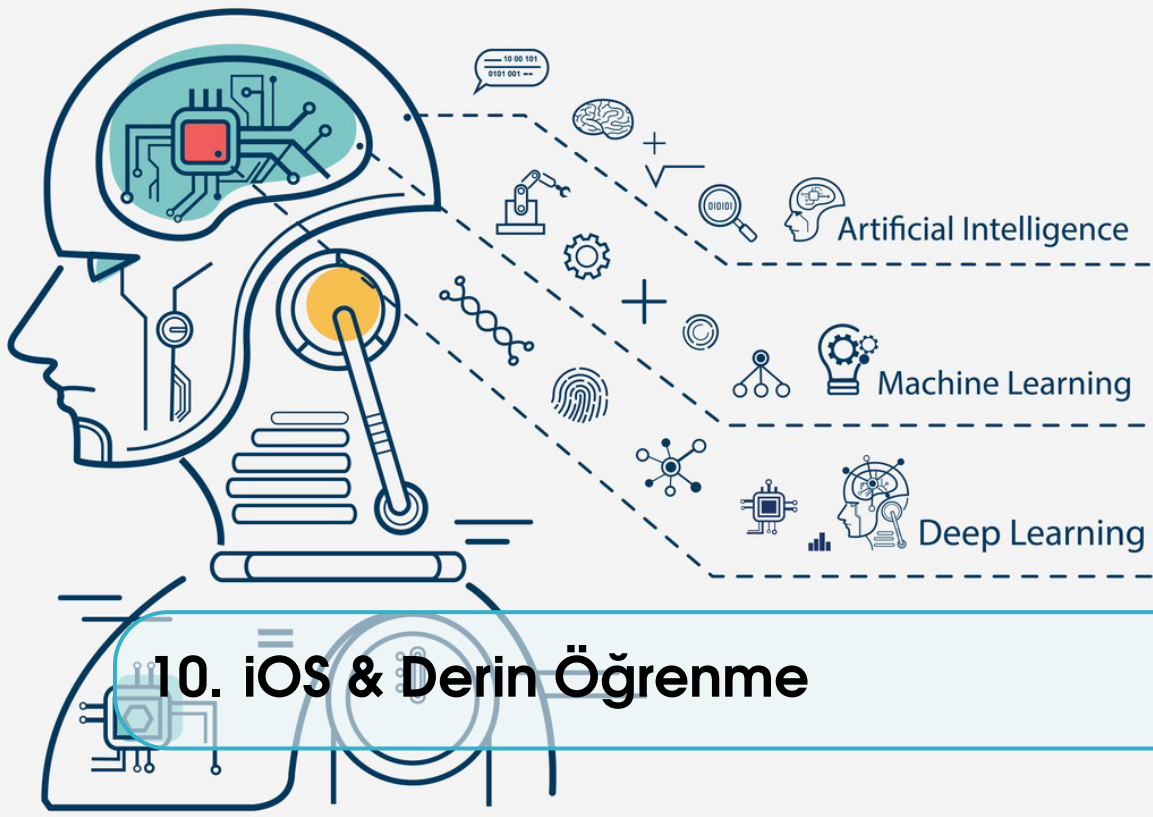
Zafarani, R., Abbasi, M. A., & Liu, H. (2014). *Social media mining: an introduction*. Cambridge University Press.

Yazarlar Hakkında



Halil İbrahim Cebeci, 1980 yılında İstanbul’da doğdu. İlk ve orta öğretimini Ankara ve Şanlıurfa’da tamamladıktan sonra, 2002 yılında İstanbul Üniversitesi Endüstri Mühendisliği bölümünden mezun oldu. Yüksek Lisansını Sakarya Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim dalında “İnternet destekli öğretimde öğrencilerin başarısını etkileyen faktörlerin istatistiksel yöntemler ve yapay sinir ağları ile analizi” ve takiben Doktorasını aynı Anabilim dalında “Çok kademeli tedarik zincirlerinde kamçı etkisinin azaltılması için yeni bir model önerisi: Genişletilmiş satıcı yönetimli envanter modeli “ adlı çalışmalarıyla tamamladı. 2002-2005 yılları arasında

Enformatik bölümü, 2005-2011 arasında ise Uzaktan Eğitim Uygulama ve Araştırma Merkezinde Araştırma görevlisi olarak ve 2011-2013 yılları arasında Bilgisayar ve Bilişim Bilimleri Fakültesinde Öğr.Gör.Dr. olarak çalıştıktan sonra 2013 yılında İşletme Fakültesinde İşletme Bölümüne Yardımcı Doçent olarak atandı. Halen aynı fakülte altında yer alan Yönetim Bilişim Sistemleri Bölümünde Doktor Öğretim Üyesi olarak çalışmaktadır. Çeşitli dergilerde, ulusal ve uluslararası kitaplarda İş zekası, yapay zeka, bilişim sistemleri, Tedarik Zinciri yönetimi alanlarında 20 civarında yayını bulunmaktadır. Ayrıca ulusal ve uluslararası (Avrupa Birliği) ölçeğinde projelerde araştırmacı olarak görev almıştır.



iOS ile Xcode Ortamında Derin Öğrenme Uygulamalarının Geliştirilmesi

Enes POLAT¹, Buğra AYAN²

¹Türk Telekom, Enerji ve Soğutma Sistemleri Müdürü, polat.enes1985@gmail.com

²Mühendis, Türkiye Cumhuriyeti Cumhurbaşkanlığı, bugra.ayan@yahoo.com

10.1 Giriş

10.1.1 XCode Yapısı

XCode, Apple teknolojisine sahip cihazlar için kod yazılabilen bir geliştirme ortamıdır. Apple'ın kendi ürünüdür. Bu platformda yeni bir proje oluşturmak için ilk olarak Şekil 10.1'de gösterilen ekrandan Create a New XCode Project bölümüne tıklanır.

Yeni proje oluştur denildikten sonra Apple, Şekil 10.2'de görüldüğü üzere kullanıcıdan nasıl bir proje oluşturacağını belirtmesini istiyor. İlk olarak Apple işletim sistemlerinden iOS, macOS, watchOS ve tvOS'tan hangisi olacağını seçiyor ardından da bu işletim sistemi için nasıl bir uygulama yapacağımızı belirtiyoruz. Eğer Multiplatform yani birden fazla cihaz için bir uygulama geliştirmek istiyorsak bunu da sol üst köşede bulunan Multiplatform seçeneği ile gerçekleştiriyoruz.

Uygulamamızı İos ortamında yapacağımız için İos bölümünden App diyerek uygulamamızı seçiyoruz.

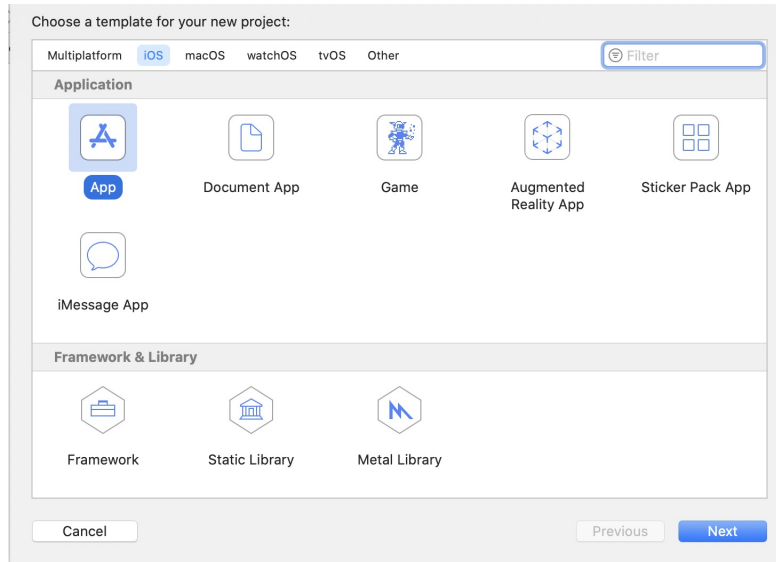
Ardından Şekil 10.3'de görüldüğü üzere uygulama bilgileri girilmesi isteniyor.

Bir sonraki kısımda Product Name, Team, Organization Name gibi kısımları dolduruyoruz. Buradaki Bundle Identifier olarak geçen kısım Android uygulamalarda da gördüğümü paket ismi. Paket isminin oluşurken Organization Identifier. Product Name şeklinde bir dizilim oluyor. Örneğin burada com.enespolat şeklinde dizilim gerçekleşmiş. Bu Apple Store'da uygulamayı yayınlarken kullanılacak ID oluyor, dolayısıyla benzersiz olması önemli.

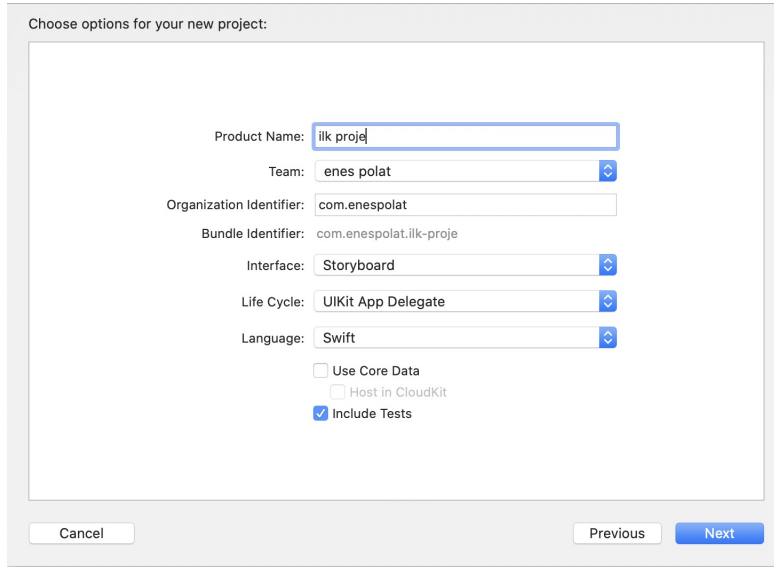
Şekil 10.4'de görünen şekilde panellerin olduğu giriş ekranı açılıyor.



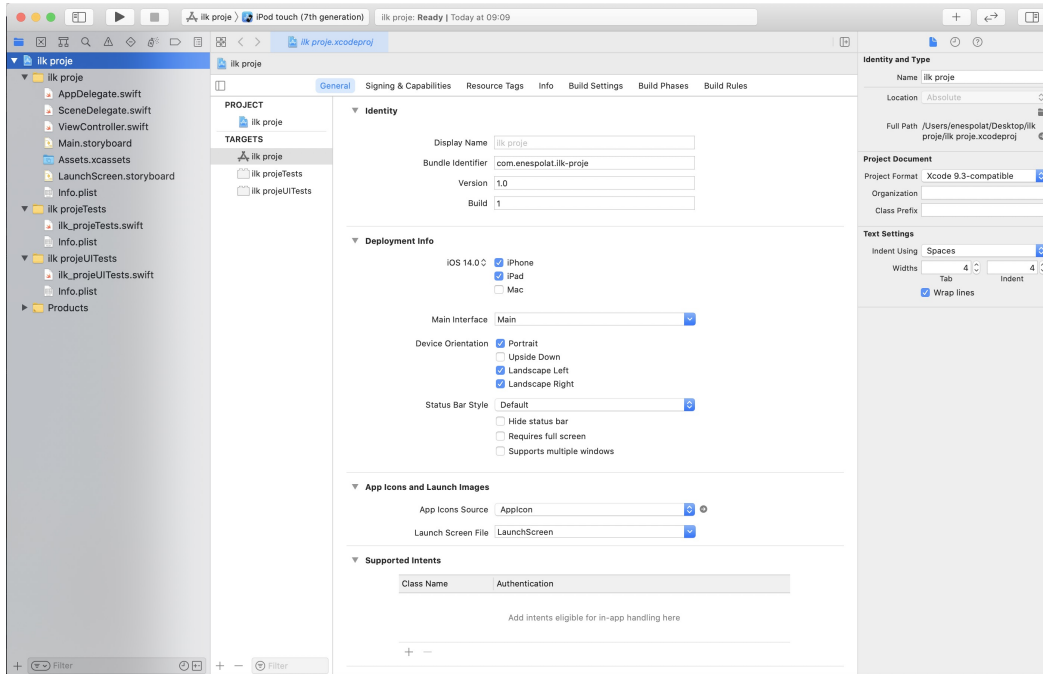
Şekil 10.1



Şekil 10.2

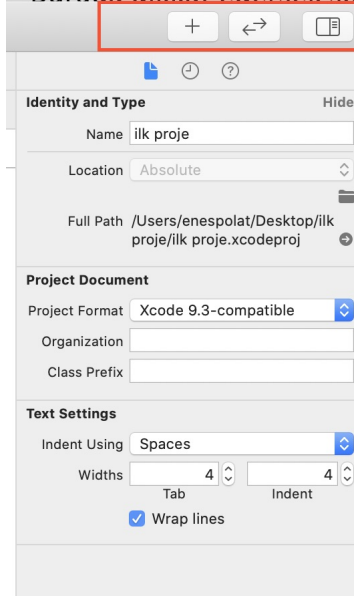


Şekil 10.3



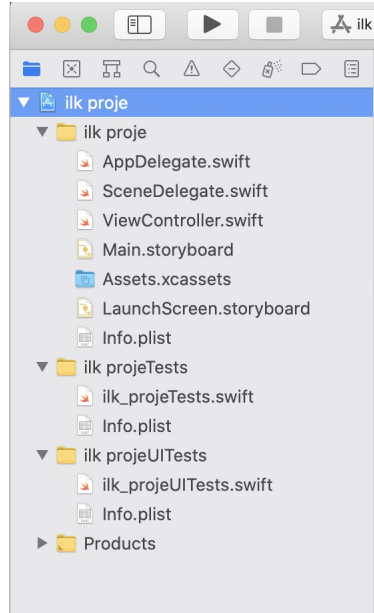
Şekil 10.4

Burada dikkat edersek sol, orta ve sağ panel var. Bir de görünmeyen alt panel var. Alt paneli açtığımızda konsol ekranı geliyor. Uygulamadaki log akışını görmek istersek buradan bakabiliyoruz. Bu ana ekranları görmek veya gizlemek için Şekil 10.5’de gösterildiği şekilde sağ üst köşedeki kısayol butonlarını kullanıyoruz.



Şekil 10.5

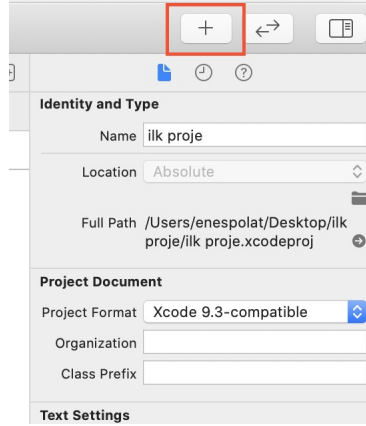
Sol panele bakalım. Şekil 10.6’den görebileceğimiz üzere burada uygulamamızda kullanacağımız dosyalar var. Bu dosyaların bazılarını daha sık kullanıyoruz.



Şekil 10.6

Sık kullandığımız bazı dosyaları açıklayalım.

Main.storyboard: Görsel arayüzlerin tasarlandığı kısım. Diyelim ki ekranda bir buton eklemek istiyoruz. Bunu Main.storyboard kısmından ekleyebiliyoruz. Örneğin ilerleyen bölümlerde bir oyun yapıp oyunun skor ekranını koyacağız. Bu skor ekranı için Main.storyboard kısmını kullanacağız. Alttaki görselde + ile gösterilen kısma tıklayarak açılan menüden arama yapıp istediğimiz elemanı Şekil 10.7’de görüldüğü üzere kolayca ekleyebiliyoruz.



Şekil 10.7

Şekil 10.8’da görülen sağ taraftaki panel ise seçili olan eleman ile ilgili detayları barındırıyor.

Görsel arayüzü Main.storyboard ile belirlerken bu arayüzü bağladığımız kod ise ViewController.swift içerisinde yer alıyor. Şöyle düşünebiliriz. Önümüzde bir kola makinesi var. Makinenin bizim gördüğümüz kısmı Main.storyboard, arkatarafı ise ViewController.swift dosyası. Bu iki kısım içeriden birbirine bağlı.

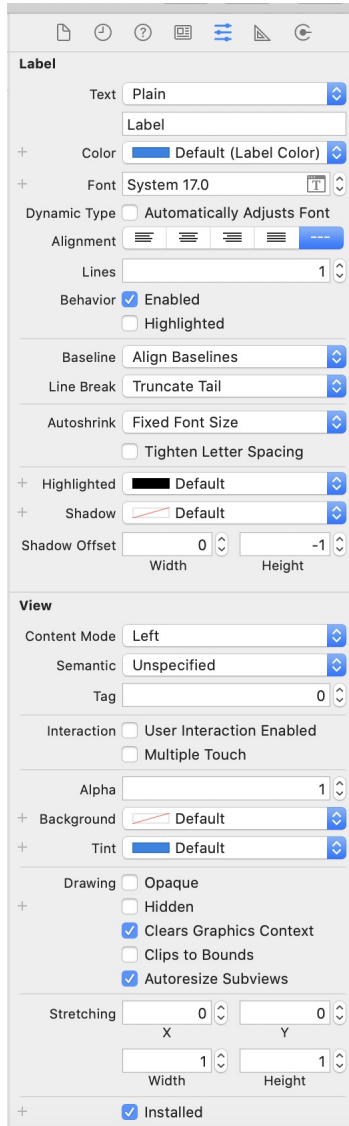
Bir başka önemli dosyamız ise Info.plist. Şekil 10.9’da arayüzü görülen bu dosya uygulamada cihazla ilgili ne kullanacağımızı belirliyoruz. Örneğin kamerayı mı kullanacağız, lokasyon bilgisini mi kullanacağız gibi sorulara burada cevap veriyoruz. Almak istediğimiz izinleri buradan Add Row diyerek otomatik tamamlama ile seçip, ekleyebiliyoruz. Apple teknolojilerini kullanarak uygulama geliştirirken bu soruların cevaplarının diğer platformlara göre çok daha kritik olduğunu unutmamak gerekiyor. Apple gizlilik konusunda hassas ve olabildiğince az veri kullanarak projenizi gerçekleştirmenizi istiyor.

10.1.2 Resim Yükleme Uygulamasının Geliştirilmesi

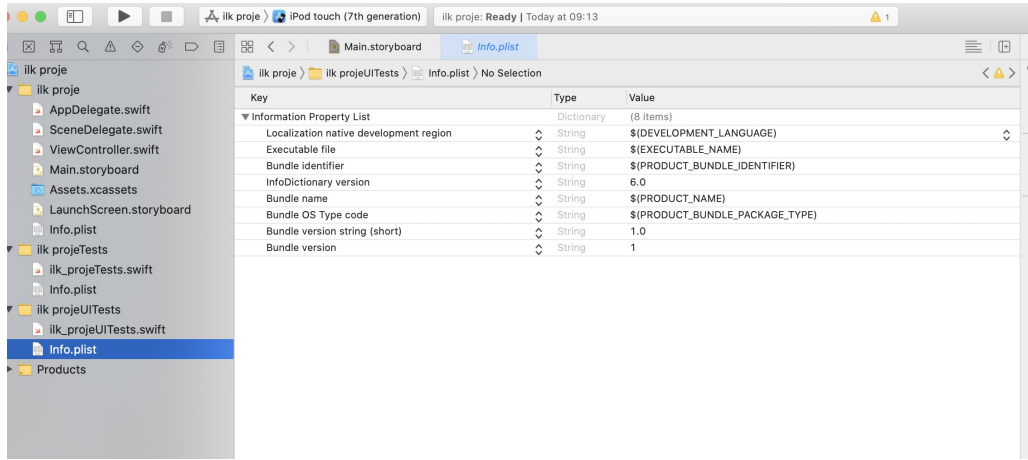
İlk uygulamanın geliştirilmesi için ilk olarak ekranı tamamen kaplayan bir UIImageView ekliyoruz. Bu UIImageView üzerine de UINavigationController ekleyerek görsel tıkladığında kullanıcının cep telefonu kamerasından veya cep telefonu içerisindeki klasörden görsel alma işlemini gerçekleştireceğiz. Bunu başardığımız takdirde bir sonraki adımda, alınan görseli derin öğrenme ile sınıflandırabileceğiz. Kod yapımızı Şekil 10.10’daki gibi oluşturuyor.

Telefonun kamerasından veya klasörlerinden görüntü alabilmek için UIImagePickerControllerDelegate ve UINavigationControllerDelegate’i projeye dahil ediyoruz. Bu Controller yapılarını aldığımızda Şekil 10.11’de görülen didFinishPickingMediaWithInfo denilen metod gelebiliyor. İçi boş gelen metodun içerisine biz dilediğimiz kodları giriyoruz.

Kullandığımız resimCek isimli fonksiyonda kaynağı kamera olarak belirtiyor ve resme herhangi



Şekil 10.8



Şekil 10.9

```

9 import UIKit
10 import CoreML
11 import Vision
12
13
14 class ViewController: UIViewController,
    UIImagePickerControllerDelegate,
    UINavigationControllerDelegate {
15
16     @IBOutlet weak var imageView: UIImageView!
17
18     let imagePicker = UIImagePickerController()
19
20     override func viewDidLoad() {
21         super.viewDidLoad()
22         imagePicker.delegate = self
23     }
24
25
26     func imagePickerController(_ picker:
    UIImagePickerController,
    didFinishPickingMediaWithInfo info:
    [UIImagePickerController.InfoKey : Any]) {
27
28         if let image = info[.originalImage] as? UIImage
    {
29
30             imageView.image = image
31             imagePicker.dismiss(animated: true,
    completion: nil)
32
33         }
34     }
35
36
37     @IBAction func resimCek(_ sender: Any) {
38         imagePicker.sourceType = .photoLibrary
39         imagePicker.allowsEditing = false
40         present(imagePicker, animated: true,
    completion: nil)
41     }
42 }

```

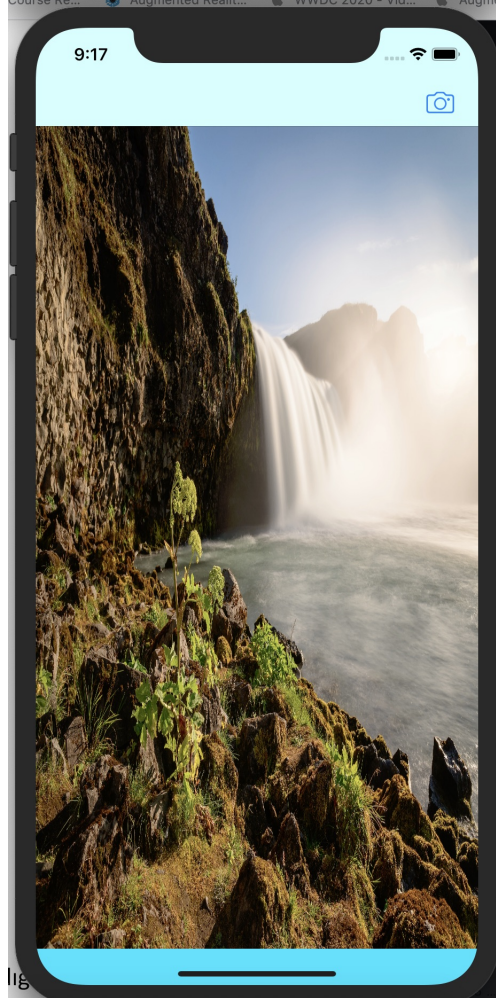
Şekil 10.10

```
37  
38     @IBAction func resimCek(_ sender: Any) {  
39         imagePicker.sourceType = .photoLibrary  
40         imagePicker.allowsEditing = false  
41         present(imagePicker, animated: true,  
                completion: nil)  
42     }
```

Şekil 10.11

bir düzenleme yapılmamasını söylüyoruz. Fotoğraf çekildikten sonra `didFinishPickingMediaWithInfo` metodu ile Info bilgisinin içerisinden görsel alınıp, bir tip dönüşümü yapılarak `UIImage` olarak kaydediliyor ve `ImageView` içerisinde gösteriliyor.

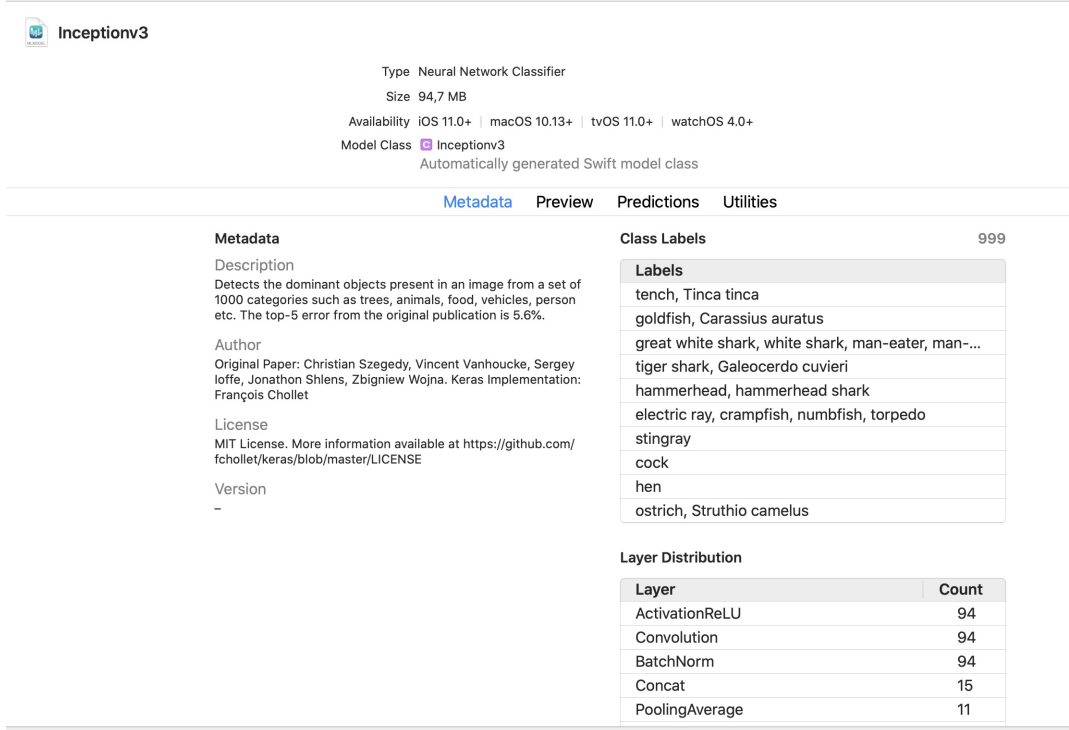
Uygulamamızı çalıştırdığımızda seçtiğimiz bir fotoğrafın Şekil 10.12’de ekranda gözüktüğünü görebiliyoruz.



Şekil 10.12

10.1.3 Hazır Modelle Sınıflandırma Uygulaması Geliştirmek

Bu uygulamada hazır bir modeli kullanarak bir görselin sosisli olup olmadığını bulmaya çalışacağız. Uygulamamızda hazır olarak indirebileceğimiz arayüzü Şekil 10.13'de görülen Inception V3 model isimli bir makine öğrenmesi modeli kullanacağız. Bu model ağaçlar, hayvanlar, yiyecekler, araçlar, insanlar ve farklı alanlarda 1000 farklı nesneyi sınıflandırabilmemizi sağlıyor. Boyutu 95 megabayt olan model MIT lisansı altında kullanılabilir.



Inceptionv3

Type Neural Network Classifier
Size 94,7 MB
Availability iOS 11.0+ | macOS 10.13+ | tvOS 11.0+ | watchOS 4.0+
Model Class **Inceptionv3**
Automatically generated Swift model class

Metadata **Preview** **Predictions** **Utilities**

Metadata

Description
Detects the dominant objects present in an image from a set of 1000 categories such as trees, animals, food, vehicles, person etc. The top-5 error from the original publication is 5.6%.

Author
Original Paper: Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna. Keras Implementation: François Chollet

License
MIT License. More information available at <https://github.com/fchollet/keras/blob/master/LICENSE>

Version
-

Class Labels 999

Labels

tench, Tinca tinca
goldfish, Carassius auratus
great white shark, white shark, man-eater, man-...
tiger shark, Galeocerdo cuvieri
hammerhead, hammerhead shark
electric ray, crampfish, numbfish, torpedo
stingray
cock
hen
ostrich, Struthio camelus

Layer Distribution

Layer	Count
ActivationReLU	94
Convolution	94
BatchNorm	94
Concat	15
PoolingAverage	11

Şekil 10.13

Öncelikle hazır bir modelin nasıl Apple sitesinden ücretsiz edinilebileceğini gösterelim.

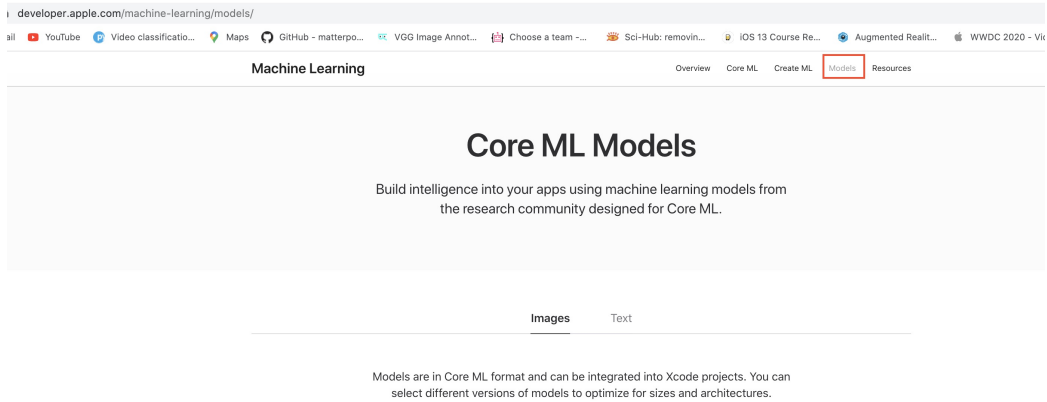
Apple teknolojisi dahilinde Makine Öğrenmesi yaklaşımlarıyla uygulama geliştirmek için ilk olarak <https://developer.apple.com/machine-learning/> adresine giderek sayfanın Makine Öğrenmesi için kendi hazırladığı destek kütüphanelerine ulaşabiliyor. İlk uygulamada hazır, Şekil 10.14'de görüldüğü üzere varolan modelleri kullanılacağı için sağ üst köşeden Models kısmına giriliyor.

Modeller bölümünde bir çok hazır model yer alıyor. Bu modellerin bir kısmı Görüntü sınıflandırma için kullanılırken Bert-SQuAD modeli ise metin sınıflandırmada kullanılıyor. Bu modellerin her biri için örnek bir projenin kaynak kodu ile beraber modelin indirilebilir hali ile karşılaşılıyor. Model indirildikten sonra ilk uygulamaya başlamak için Xcode geliştirme ortamına dönüş sağlanıyor.

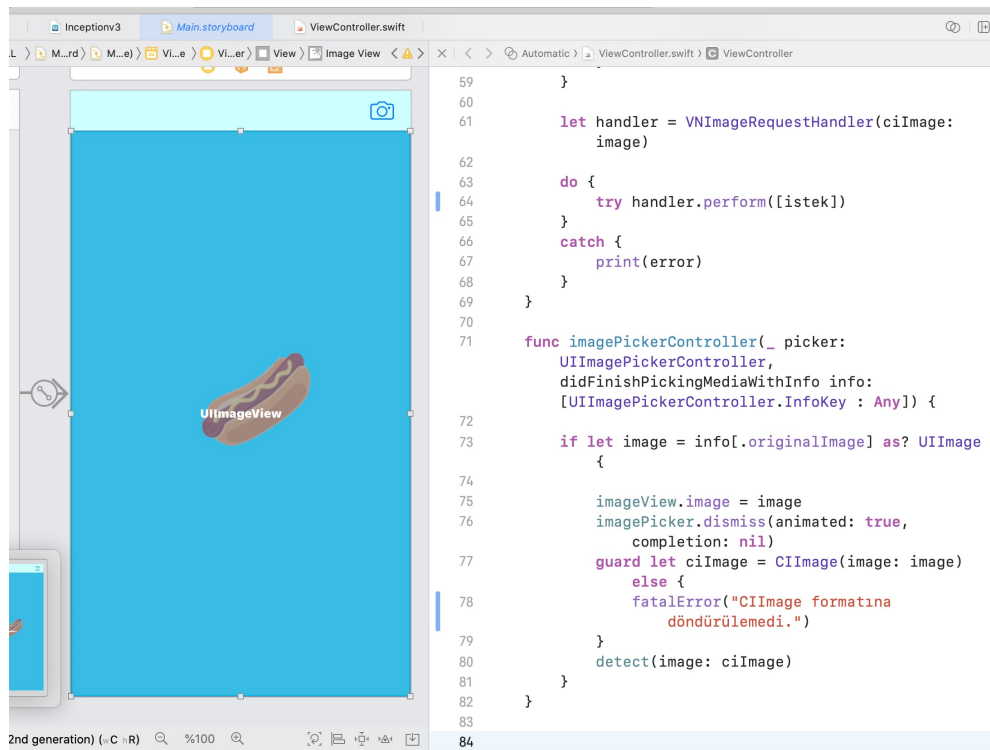
İlk uygulamadakine benzer şekilde ekranı tamamen kaplayan bir UIImageView ve sağ üst köşeye bir Fotoğraf butonu ekliyoruz. Kod yapımızı Şekil 10.15'de görüyoruz.

Kod kısmında ilk olarak Şekil 10.16'de görüldüğü şekilde görüntü işleme için kullanılan Vision kütüphanesinden faydalanıyoruz.

Bu kütüphane sayesinde Şekil 10.17'de görüldüğü şekilde VNCoreMLModel'i çağırabiliyoruz. Bu sayede Inceptionv3 modelini çağırabiliyoruz.



Şekil 10.14



Şekil 10.15

```

8
9 import UIKit
10 //import CoreML
11 import Vision
12

```

Şekil 10.16

```

25
26     func detect(image: CIImage) {
27
28         // Load the ML model through its generated class
29         guard let model = try? VNCoreMLModel(for:
30             Inceptionv3().model) else {
31             fatalError("model yüklenemedi")
32         }
33     }
34
35     let istek = VNCoreMLRequest(model: model) { request,
36         error in
37         guard let results = request.results as?
38             [VNClassificationObservation],
39             let topResult = results.first
40         else {
41             fatalError("umulmayan hata")
42         }
43     }
44 }

```

Şekil 10.17

Modelimizi yükledikten sonra Şekil 10.18'deki kod yapısında ilk olarak ona bir talep gönderiyoruz. Ve bu talebin sonuçlarını da VNClassificationObservation yani sınıflandırma formatında kullanıyoruz. Bu sonuçlarının en iyisini first olarak alıyoruz. Eğer bu sonuç HotDog ise ekranda HotDog, değilse Not HotDog! yazdırıyoruz.

```

35     let istek = VNCoreMLRequest(model: model) { request, error
36         in
37         guard let results = request.results as?
38             [VNClassificationObservation],
39             let topResult = results.first
40         else {
41             fatalError("umulmayan hata")
42         }
43     }
44
45     print(results)
46
47     if topResult.identifier.contains("hotdog") {
48         DispatchQueue.main.async {
49             self.navigationItem.title =
50                 "\(topResult.confidence * 100)% Hotdog!"
51             self.navigationController?.navigationBar
52                 .barTintColor = UIColor.green
53             self.navigationController?.navigationBar
54                 .isTranslucent = false
55         }
56     }
57     else {
58         DispatchQueue.main.async {
59             self.navigationItem.title = "Not Hotdog! "
60             self.navigationController?.navigationBar
61                 .barTintColor = UIColor.red
62             self.navigationController?.navigationBar
63                 .isTranslucent = false
64         }
65     }
66 }

```

Şekil 10.18

Bu kısım fotoğrafın modele girdiğinde nasıl işleme uğrayacağına dairdi. Diğer taraftan da ilk uygulamada UIImagePickerController içine geldiğini gördüğümüz görseli bu modele gönderen kodu yazmalıyız.

Bunun için Şekil 10.19'daki şekilde UIImagePickerController fonksiyonu içerisinde öncelikle alınan görseli UIImage tip dönüşümü yapıp UIImage olarak tanımlıyoruz. Modelimiz bu değişkeni gördüğü için bağlantı kurulmuş oluyor.

```

71 func UIImagePickerController(_ picker: UIImagePickerController,
    didFinishPickingMediaWithInfo info:
    [UIImagePickerController.InfoKey : Any]) {
72
73     if let image = info[.originalImage] as? UIImage {
74
75         imageView.image = image
76         picker.dismiss(animated: true, completion: nil)
77         guard let ciImage = UIImage(image: image) else {
78             fatalError("UIImage formatına dönüştürülemedi.")
79         }
80         detect(image: ciImage)
81     }
82 }
83
84
85 @IBAction func resimCek(_ sender: Any) {
86     picker.sourceType = .photoLibrary
87     picker.allowsEditing = false
88     present(picker, animated: true, completion: nil)
89 }
90
91 }
92 // Helper function inserted by Swift 4.2 migrator.
93 fileprivate func convertFromUIImagePickerControllerInfoKey(_
    input: UIImagePickerController.InfoKey) -> String {
94     return input.rawValue
95 }
96

```

Şekil 10.19

Uygulamayı test edelim. Şekil 10.20 ve Şekil 10.21'de görüldüğü şekilde Sosisli görseline %99.7 HotDog derken sosisli olmayan bir görsel için Not Hotdog dedi.

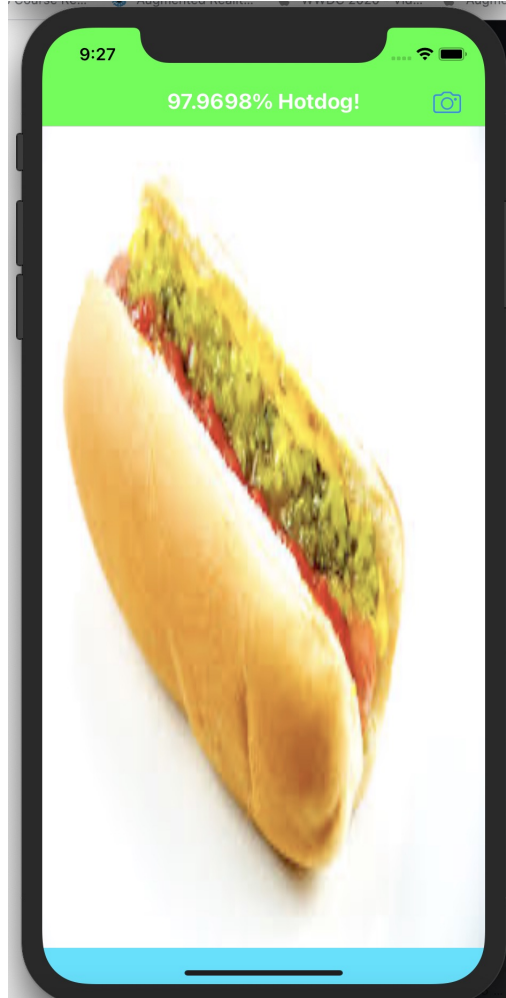
Bu uygulamada InceptionV3 modelini kullanarak bir sınıflandırma yaptık. Sıradaki uygulamada ise kendi modelimizi oluşturacağız.

10.2 CreateMLUI ile Model Oluşturma

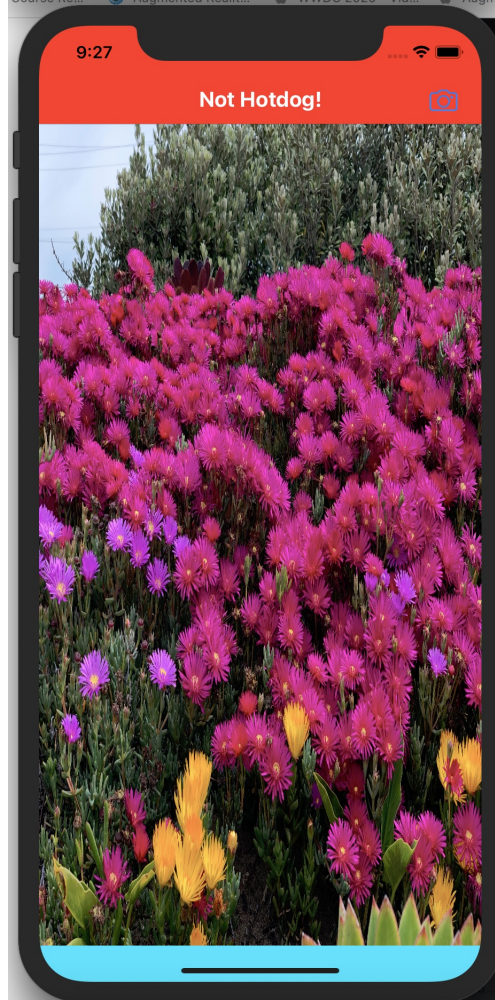
Bu uygulamada çiçeklerden oluşturduğumuz verisetini, eğitim ve test olarak ayıracak ardından da eğitimi yaparak modeli oluşturacağız. Uygulamaya başlamadan önce Apple'ın Playground yapısından bahsedelim. Playground içerisinde Windows işletim sistemindeki Komut İstemi'nde olduğu gibi kodlar çalıştırabileceğimiz aynı zamanda sınıflandırma işlemleri yapabileceğimiz bir bölüm.

Modelimizin eğitimini Şekil 10.22'de yer alan Playground yardımıyla yapacağız. Eğitimi mac bilgisayarda yapacağımız için üst menüden macOS seçip Blank yani boş diyerek devam ediyoruz.

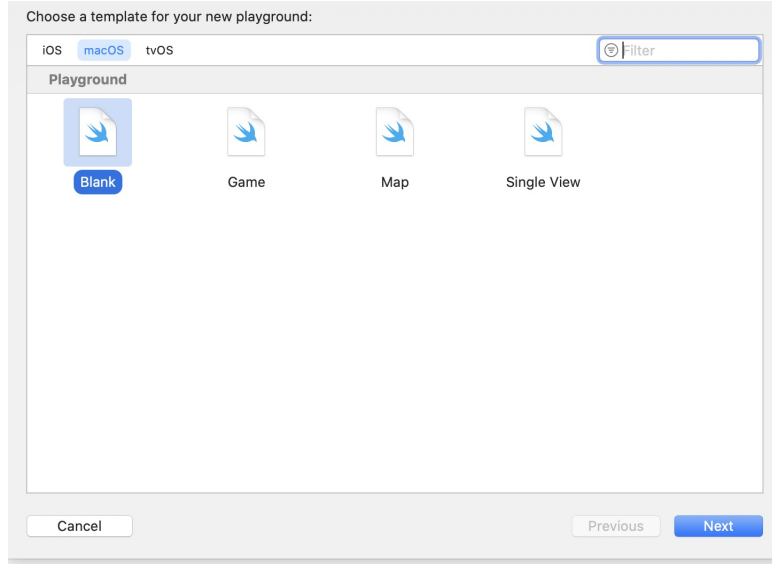
Oluşan Playground dosyasına kodumuzu Şekil 10.23'de görüldüğü şekilde ekliyoruz.



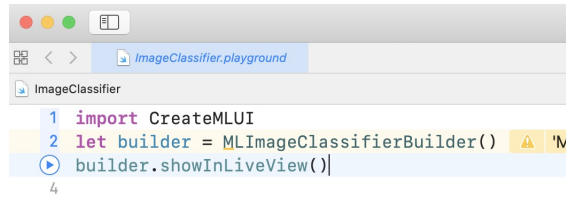
Őekil 10.20



Şekil 10.21

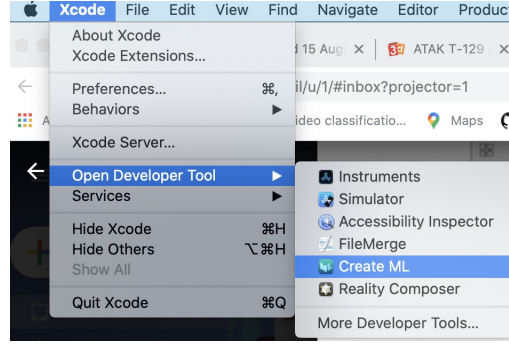


řekil 10.22



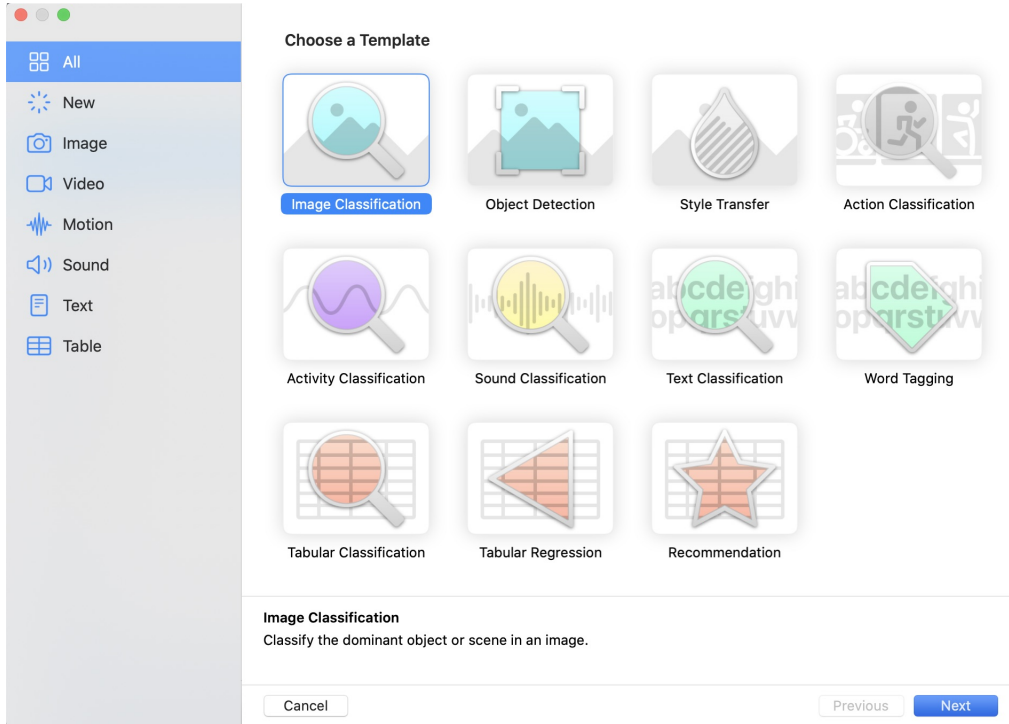
řekil 10.23

Ardından Şekil 10.24'de görüldüğü şekilde Xcode-> Open Developer Tool -> Create ML ile bir makine öğrenmesi modeli oluşturacağımızı belirtiyoruz.



Şekil 10.24

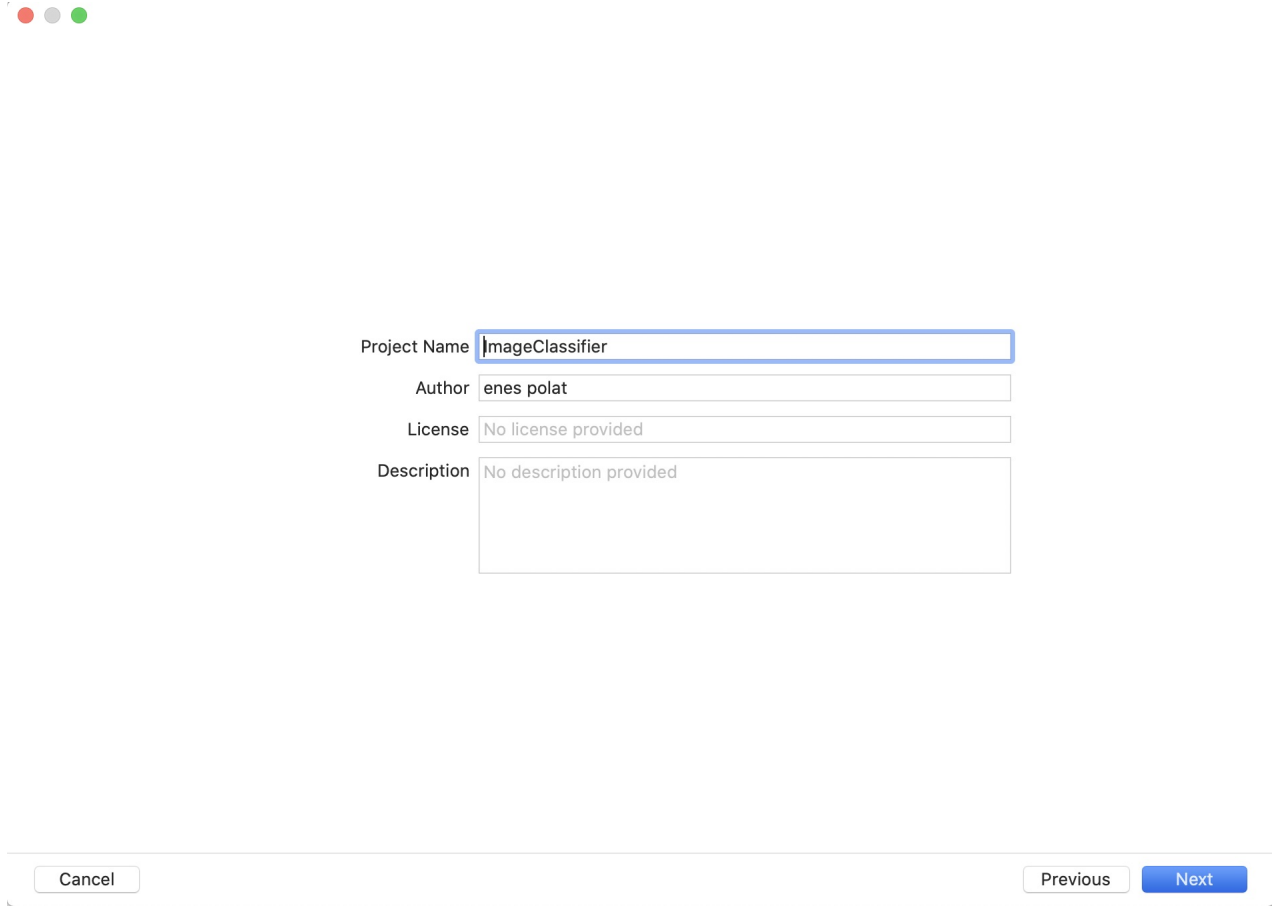
Buradan Şekil 10.25'da görüldüğü şekilde nasıl bir sınıflandırma yapabileceğimizi seçiyoruz.



Şekil 10.25

Seçimi yaptıktan sonra Şekil 10.26'de yer alan projeye ilgili bilgiler girerek devam ediyoruz. İlerlediğimizde XCode bize, Şekil 10.27'de görüldüğü üzere eğitim setini , doğrulama setini ve test setini nereye koyabileceğimizi gösteriyor.

Önceden belirlediğimiz görselleri Şekil 10.28'de görüldüğü üzere bu klasörler içine koyuyoruz. İlerle dediğimizde Şekil 10.29'da görüldüğü üzere Xcode eğitime başlıyor. Eğitim bitince modelimiz Şekil 10.30'daki şekilde oluşuyor.



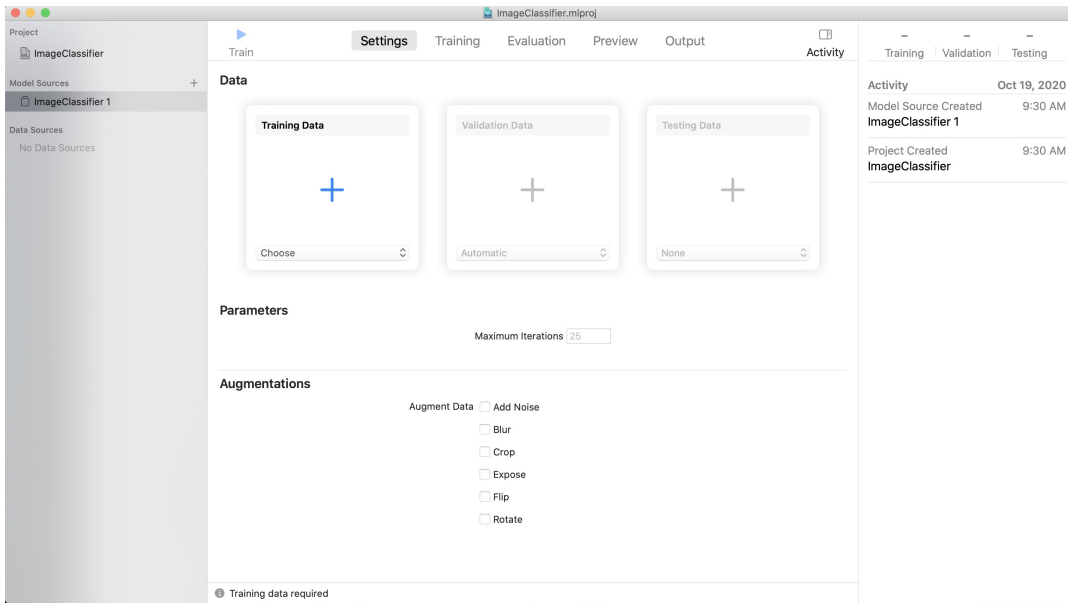
Project Name

Author

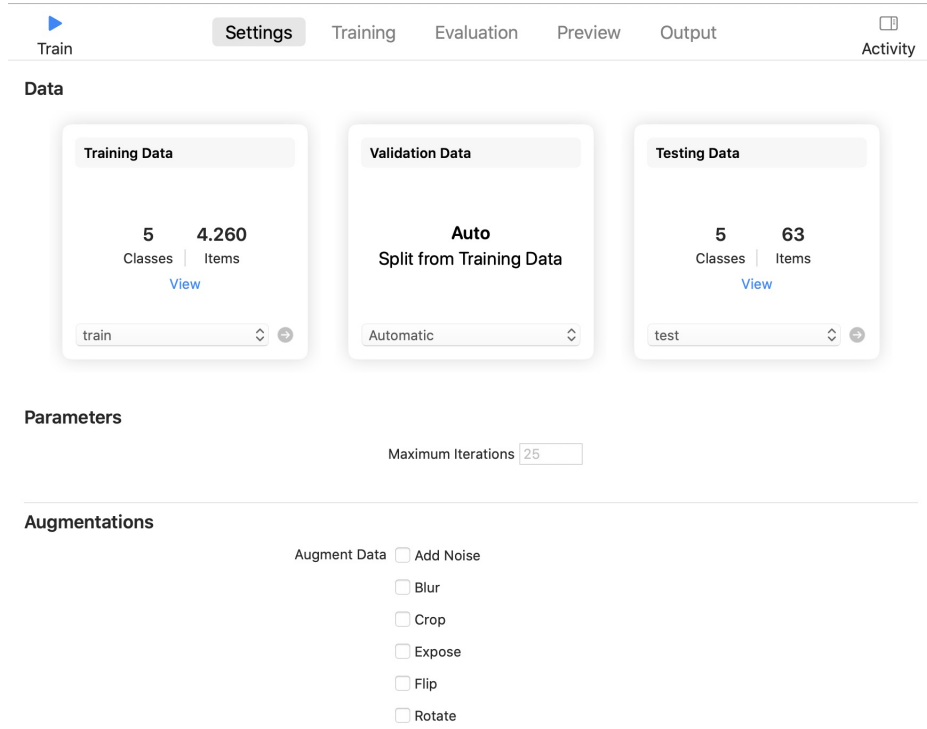
License

Description

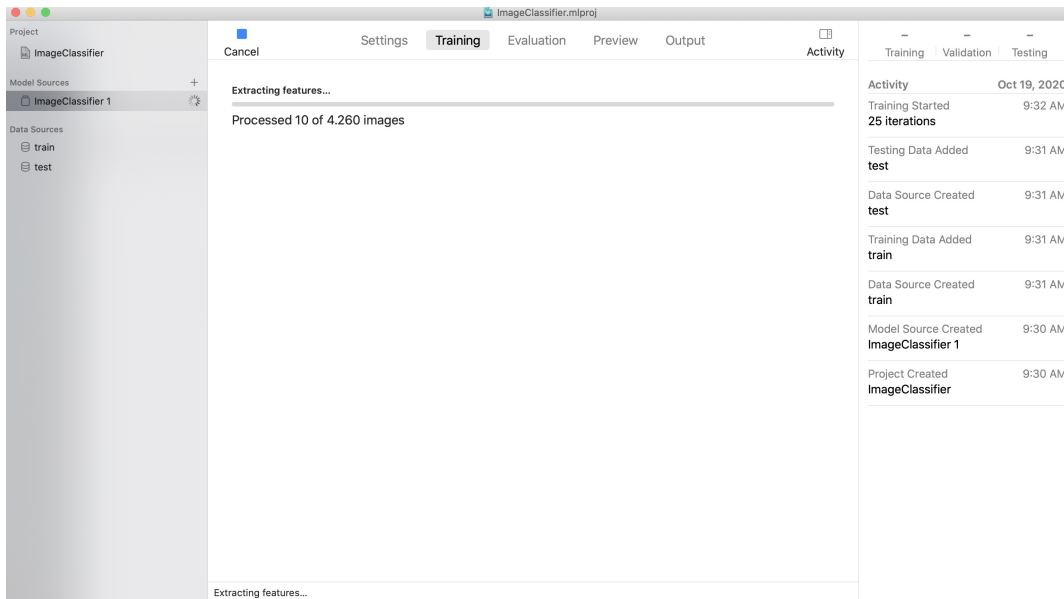
řekil 10.26



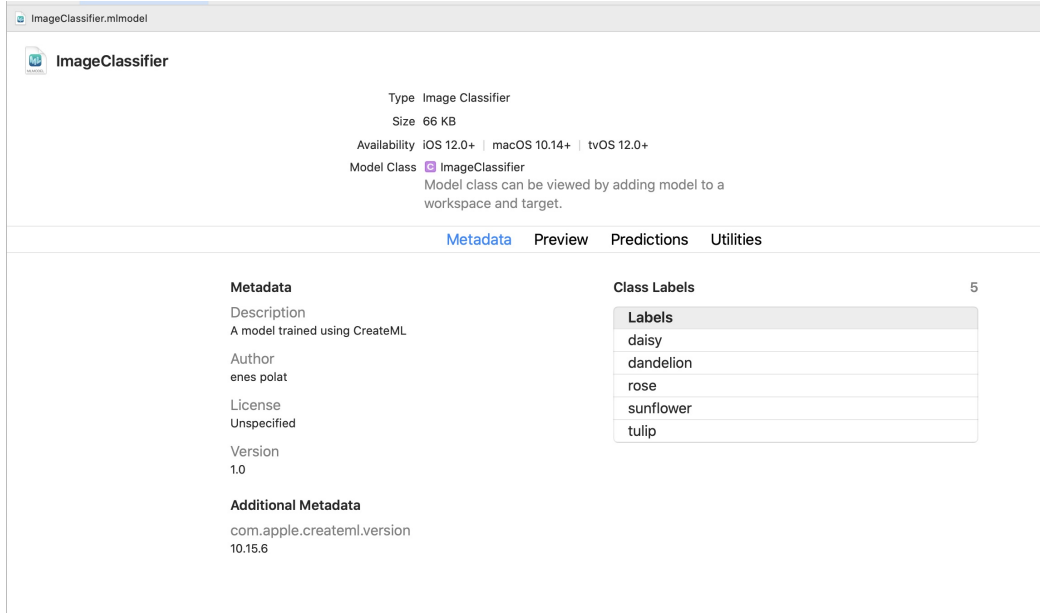
řekil 10.27



Şekil 10.28



Şekil 10.29



Şekil 10.30

Şekil 10.31’de yer alan kod tarafında ise ImageClassifier ismini verdiğimiz modelimiz görseli alıyor ve sınıflandırma yapıyor.

Şekil 10.32’deki kodlardan görüleceği üzere eğer sınıflandıramadıysa Unable to classify image, içeriği boşsa Nothing recognized ve sınıflandırma gerçekleştiyse de Classification: diyerek sınıflandırma sonucunu yazıyor. Bu sonuçlarda tahminlerinin içerisinde en yüksek doğruluğa sahip olduğunu düşündüğü 2 taneyi alıyor.

Uygulamamızın test sonucu Şekil 10.33’deki sonucu veriyor.

10.3 CreateML ile Sınıflandırma

Bir önceki uygulamada arayüz kullanarak yaptığımız model oluşturmayı bu uygulamada tamamen kodlar üzerinden giderek yapacağız. Bu örneğimizde metin sınıflandırma yapacağız. Model yazılan yorumları pozitif veya negatif olarak sınıflandırmaya çalışacak. Kodumuza bakalım. Foundation kütüphanesi dosyaları okumak için CreateML ise modelimizi oluşturmak için kullanılan kütüphane. İlk olarak Şekil 10.34’de görüldüğü üzere eğitim verisinin amazon-reviews, test verisinin ise testing-reviews isimli json dosyası olduğunu belirtiyoruz. Eğer bu dosyalara ulaşamazsa Error! Could not load resource files. diyerek dosyalara ulaşamadığı hatasını döndürüyor.

Bu veriler Şekil 10.35’de görülebileceği üzere json formatında geldi.

Bize ise MLDataTable dediğimiz bir makine öğrenmesi veri tablosu şeklinde lazım. Bu yüzden Şekil 10.36’de görüleceği üzere bir tip dönüşümü yapıyoruz. Boyutlarını ve istatistiklerini de göstererek veriyi doğru bir şekilde aldığımızdan emin oluyoruz.

Ardından Şekil 10.37’de görüleceği üzere MLTextClassifier ile textColumn içerisinde text kısmını, labelColumn içerisinde de label kısmını alıyoruz. Burada kaybımızı 1.0 dan çıkarttığımızda bu bizim doğruluğumuz oluyor.

Son olarak Şekil 10.38’de görüleceği üzere modelimizi kaydedeceğimiz yeri belirtip, sentimentClassifier.write metoduyla modelin dosyasını ve metadatasını belirterek kaydediyoruz.

```
13 class ImageClassificationViewController: UIViewController {
14     // MARK: - IBOutlets
15
16     @IBOutlet weak var imageView: UIImageView!
17     @IBOutlet weak var cameraButton: UIBarButtonItem!
18     @IBOutlet weak var classificationLabel: UILabel!
19
20     // MARK: - Image Classification
21
22     /// - Tag: MLModelSetup
23     lazy var classificationRequest: VNCoreMLRequest = {
24         do {
25             /*
26              Use the Swift class `MobileNet` Core ML generates from the
27              model.
28              To use a different Core ML classifier model, add it to the
29              project
30              and replace `MobileNet` with that model's generated Swift
31              class.
32              */
33             let model = try VNCoreMLModel(for:
34                 ImageClassifier().model) // TODO: replace with a Core
35                 ML model
36
37             let request = VNCoreMLRequest(model: model, completionHandler:
38                 { [weak self] request, error in
39                     //...
40                     self?.processClassifications(for: request, error: error)
41                 })
42             request.imageCropAndScaleOption = .centerCrop
43             return request
44         } catch {
45             fatalError("Failed to load Vision ML model: \(error)")
46         }
47     }()
48
49     /// - Tag: PerformRequests
50     func updateClassifications(for image: UIImage) {
51         classificationLabel.text = "Classifying..."
52     }
53 }
```

Şekil 10.31


```

67     func processClassifications(for request: VNRequest, error: Error?) {
68         DispatchQueue.main.async {
69             guard let results = request.results else {
70                 self.classificationLabel.text = "Unable to classify
71                 image.\n\(error!.localizedDescription)"
72                 return
73             }
74             // The `results` will always be
75             // `VNClassificationObservation`s, as specified by the Core
76             // ML model in this project.
77             let classifications = results as! [VNClassificationObservation]
78
79             if classifications.isEmpty {
80                 self.classificationLabel.text = "Nothing recognized."
81             } else {
82                 // Display top classifications ranked by confidence in the
83                 // UI.
84                 let topClassifications = classifications.prefix(2)
85                 let descriptions = topClassifications.map { classification
86                     in
87                     // Formats the classification for display; e.g.
88                     // "(0.37) cliff, drop, drop-off".
89                     return String(format: " (%.2f) %@",
90                                 classification.confidence,
91                                 classification.identifier)
92                 }
93                 self.classificationLabel.text = "Classification:\n" +
94                 descriptions.joined(separator: "\n")
95             }
96         }
97     }
98 }
99

```

Şekil 10.32

Bu şekilde sınıflandırıcımız hazır oluyor. Şimdi modelimizi uygulama içerisine entegre edelim. Uygulamamızda kullanıcı bir cümle girecek, biz de bu cümlenin negatif mi yoksa pozitif mi olduğunu kendi oluşturduğumuz modelle anlamaya çalışacağız. Uygulamamızın ekran görüntüsü Şekil 10.39'da görülüyor.

Kodumuzda ilk olarak Şekil 10.40'da görüleceği üzere UIKit ve doğal dil işleme işlemi için NaturalLanguage kütüphanelerini içe aktarıyoruz. Ardından arayüzümüzde bulunan UIImageView, TextView ve UIButton'u tanımlıyoruz.

Şimdi modelimizi çağıralım. Kritik olan nokta bu modeli NLMModel yapısında çağırılmaması. Natural Language kütüphanesini içe aktarmıştık. Bu kütüphane içerisinden NLMModel tipinde bir modelimiz olduğunu belirterek önceki aşamada oluşturduğumuz modeli Şekil 10.41'de çağırıyoruz.

Modeli içe aktardıktan sonra sınıflandırıcımızı kullanma kısmı kalıyor. Eğer kullanıcının bize sorduğu metin boş ise doğrudan False dönerken, eğer metin sınıflandırıcımızda pozitif olarak niteleniyorsa Pozitif belirtmek için koyduğumuz yukarı doğru olan baş parmak, negatif olarak belirleniyorsa negatif durumu belirtmek için koyduğumuz aşağı baş parmak sonucunu veriyor. Şekil 10.42'deki kod yapısını kullanıyoruz.

Bu şekilde uygulamamızı kullanabiliyoruz.

CreateML Converters ile Modelin .mlmodel'e Dönüştürülmesi Son örneğimizde ise Şekil 10.43'de görülen popüler derin öğrenme kütüphanelerinden birini XCode ortamında kullanabilmek için .mlmodel'e dönüştüreceğiz.

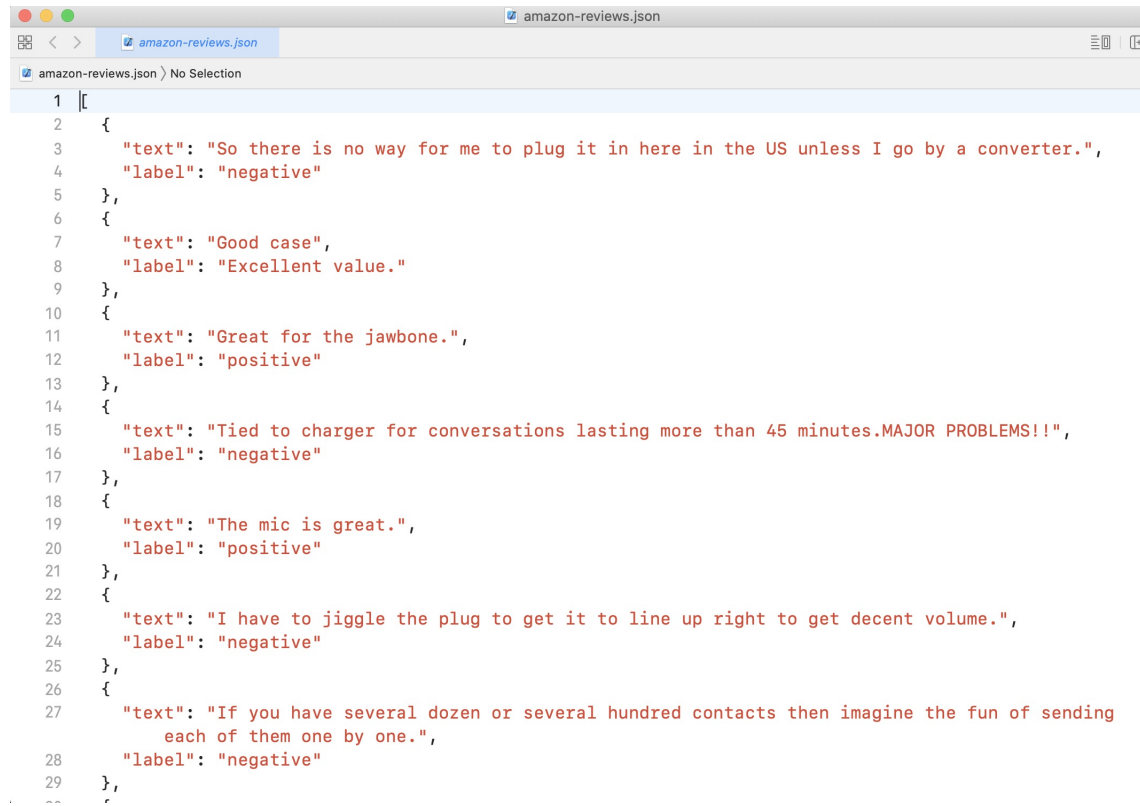
Tip dönüşümü yaparken Python kullanıyoruz. Aslında son derece basit bir kullanımı var. Python'da bulunan coremltools isimli kütüphaneyi içe aktardıktan sonra dönüştürücüler içerisinden keras2mlmodel ile içeriği alıp, label değerlerini tanımlayıp çıktının ismini Şekil 10.44'de görüldüğü



Şekil 10.33

```
5 guard let trainingDataFileURL = Bundle.main.url(forResource: "amazon-reviews", withExtension: "json"),
6 let testingDataFileURL = Bundle.main.url(forResource: "testing-reviews", withExtension: "json") else {
7     fatalError("Error! Could not load resource files.")
8 }
9 }
```

Şekil 10.34



```

1  |
2  {
3    "text": "So there is no way for me to plug it in here in the US unless I go by a converter.",
4    "label": "negative"
5  },
6  {
7    "text": "Good case",
8    "label": "Excellent value."
9  },
10 {
11  "text": "Great for the jawbone.",
12  "label": "positive"
13 },
14 {
15  "text": "Tied to charger for conversations lasting more than 45 minutes.MAJOR PROBLEMS!!",
16  "label": "negative"
17 },
18 {
19  "text": "The mic is great.",
20  "label": "positive"
21 },
22 {
23  "text": "I have to jiggle the plug to get it to line up right to get decent volume.",
24  "label": "negative"
25 },
26 {
27  "text": "If you have several dozen or several hundred contacts then imagine the fun of sending
28      each of them one by one.",
29  "label": "negative"
30 },

```

Şekil 10.35

```

10 do {
11   let trainingDataTable = try MLDataTable(contentsOf: trainingDataFileURL)
12   let testingDataTable = try MLDataTable(contentsOf: testingDataFileURL)
13
14   let stats = ""
15
16   =====
17   Entries used for training: \((trainingDataTable.size)
18   Entries used for testing: \((testingDataTable.size)
19
20   ""
21   print(stats)
22
23   let sentimentClassifier = try MLTextClassifier(trainingData: trainingDataTable

```

Şekil 10.36

```

23   let sentimentClassifier = try MLTextClassifier(trainingData: trainingDataTable, textColumn: "text", labelColumn: "label")
24
25   let trainingAccuracy = (1.0 - sentimentClassifier.trainingMetrics.classificationError) * 100
26   let validationAccuracy = (1.0 - sentimentClassifier.validationMetrics.classificationError) * 100
27
28   //let evaluationMetrics = sentimentClassifier.evaluation(on: testingDataTable)
29   //let evaluationAccuracy = (1.0 - evaluationMetrics.classificationError) * 100
30
31   let message = ""
32
33   =====
34   Training accuracy: \((trainingAccuracy)
35   Validation accuracy: \((validationAccuracy)
36
37   ""
38   print(message)
39

```

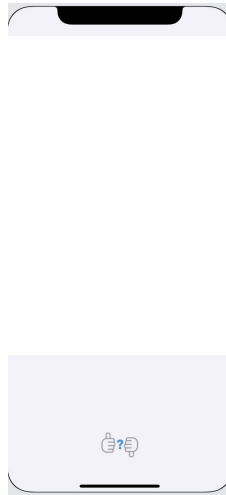
Şekil 10.37

```

44 // TODO: update the path to your Desktop directory
45 let modelFileURL = URL(fileURLWithPath: "/Users/enespolat/Desktop/yazsum/bolum 3/ReviewClassifier.mlmodel")
46 try sentimentClassifier.write(to: modelFileURL, metadata: metadata)
47 } catch {
48     print(error.localizedDescription)

```

Şekil 10.38



Şekil 10.39

```

3 import UIKit
4 import NaturalLanguage
5
6 class ViewController: UIViewController {
7
8     @IBOutlet weak var sentimentImageView: UIImageView!
9     @IBOutlet weak var textView: UITextView!
10    @IBOutlet weak var clearButton: UIButton!
11

```

Şekil 10.40

```

12 private lazy var sentimentClassifier: NLModel? = {
13     let model = try? NLModel(mlModel: ReviewClassifier().model)
14     return model
15 }()

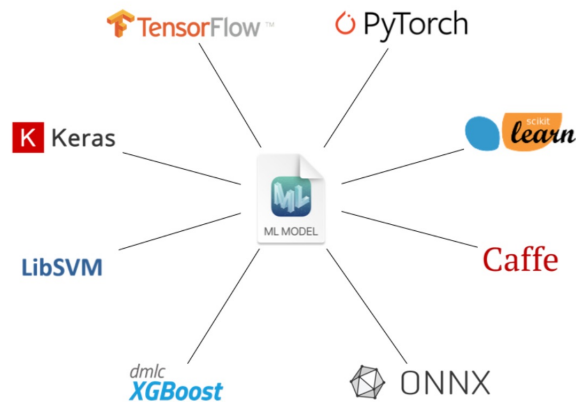
```

Şekil 10.41

```
30 extension ViewController: UITextViewDelegate {
31     func textView(_ textView: UITextView, shouldChangeTextIn range: NSRange,
32                 replacementText text: String) -> Bool {
33         if textView.text.isEmpty == false,
34             text == "\n" {
35             //...
36             if let label = sentimentClassifier?.predictedLabel(for: textView.text) {
37                 switch label {
38                     case "positive":
39                         sentimentImageView.image = UIImage(named: "positive")
40                     case "negative":
41                         sentimentImageView.image = UIImage(named: "negative")
42                     default:
43                         sentimentImageView.image = UIImage(named: "what")
44                 }
45             }
46             textView.resignFirstResponder()
47         }
48         return true
49     }
50 }
```

Şekil 10.42

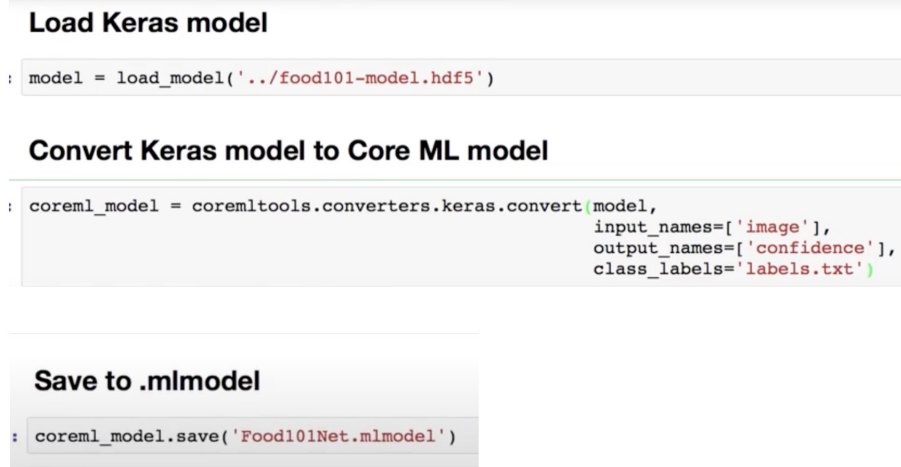
Core ML Converters



Şekil 10.43

gibi tanımlıyoruz. Bu kısa kodla artık bir keras modelini de mlmodel olarak kullanabiliyoruz. Aynı şekilde diğer derin öğrenme kütüphanelerinde de coremltools kullanarak dönüşümü yapıyoruz.

Core ML Converters keras2mlmodel



Şekil 10.44

10.4 Sonuç

Çalışmada Apple ortamında derin öğrenme uygulamaları geliştirme yöntemi örnekler ile ele alınmıştır. Apple'ın geliştirdiği ürünlerde sağladığı stabil çalışma rahatlığı ve ekosistem dahilindeki her parçanın diğeriyle uyumlu olması prensibi çalışmaların hızlı ve sorunsuz bir şekilde yürümesini sağlamıştır. Apple'ın sitesinde bulunan hazır modeller zamanla değişebilmekte, kimi zaman yenileri eklenirken bazen de çıkarılan modeller olabilmektedir. Bu sebeple sitenin sık takibi geliştiricilere kolaylık sağlamaktadır. Ayrıca Apple teknolojilerini kullanarak derin öğrenme uygulaması geliştirmek isteyen geliştiricilerin Apple yeniliklerini takip etmesi son derece önemlidir. Örneğin bu araştırma yapılırken tanıtılan iPhone 12 içerisindeki lidar özelliği dahi derin öğrenme yaklaşımını tamamen değiştirebilecek niteliklere sahiptir. Dolayısıyla Apple evrenini bir bütün olarak görmek ve bu evrendeki her gelişmenin asgari ve azami düzeyde sizin projelerinize yansımaları hesaplamak derin öğrenme uygulamanızın başarıya ulaşma ihtimalini de artıracaktır.

Yazarlar Hakkında



Enes Polat, 2008 KTU Elektrik Elektronik Mühendisliği bölümü mezunudur. 10 yıldır haberleşme sektöründe çeşitli görevlerde bulunmaktadır. 2017 yılından itibaren Türk Telekom Erzurum Bölge Müdürlüğünde Enerji ve Soğutma Sistemleri Müdürü olarak görev yapmaktadır. Enerji ve haberleşme alanlarında görevini sürdürmektedir. Özel ilgi olarak Yapay Zeka, IOT, Android, IOS, Flutter, Python, OpenCV, Kotlin, Linux, görüntü işleme ve Siber güvenlik alanlarıyla uğraşmaktadır.



Buğra Ayan, 2012 KTU Elektrik Elektronik Mühendisliği bölümü mezunudur. 2013 yılında Milli Savunma Bakanlığı Araştırma Geliştirme Daire Başkanlığı'nda Hava ve Uzay Sistemleri'nde mühendis olarak göreve başlayan Ayan, 2015 yılı itibariyle Cumhurbaşkanlığı İdari İşler Başkanlığı'nda görev yapmaktadır. Gazi Üniversitesi Bilişim Enstitüsü'nde yüksek lisans çalışmasını tamamlayan Ayan, doktora çalışmasına Gazi Üniversitesi Bilişim Enstitüsü'nde devam etmektedir.



11. Etmenlerde Stackelberg Oyun Yaklaşımı

Çok Etmenli Sistemlerde Stackelberg Oyun Yaklaşımı ve Uygulaması

Safiye Turgay¹, Orhan Torkul¹

¹Sakarya Üniversitesi, Endüstri Mühendisliği Bölümü, Serdivan, Sakarya, Türkiye

11.1 Giriş

Çok etmenli sistemlerde, Stackelberg oyun yaklaşımı ile stratejik kararların işbirliği halinde daha etkin verilmesini sağlayan model yapısı ele alınmış ve tedarik zinciri yönetimi üzerinde uygulamalı olarak gösterilmiştir. Stackelberg oyun yaklaşımı diğer oyun kuramlarına göre bir önceki durumu da dikkate almış olmasından dolayı, etmenlerin yapısı ile birlikte sıralı kararlar stratejisi ile davranmalarını sağlar. Stratejik kararlar vermesine olanak sağlamasından dolayı tercih edilmiştir. Bu çalışmada çok etmenli sistemlerin Stackelberg analizi uygulamalı olarak gösterilmiştir.

Çok etmenli sistemler, birden fazla etmenin, koordinasyonlu, kolektif zeka ve etkileşim ile birlikte ortak bir strateji ya da bağımsız stratejileri gerçekleştirmelerini sağlayan bir yapıya sahiptir. Etkileşim, koordinasyon, kolektif zeka, çok etmenli sistemlerin en temel yapısını oluşturmaktadır. Bu davranış yapısı ile karmaşık görevleri yerine getirebilen, dağıtık özellikteki özerk etmenler temsil edilebilmektedir. Bununla birlikte bu yapı ile sistemin koordineli ve stratejik bir biçimde davranabilmesi ancak kolektif yapı ve bağımsız karakterler sayesinde mevcut yapının değerlendirilmesi ve elde bulunan kaynakların değerlendirilerek en etkin biçimde kullanılabilmesi ile mümkündür.

Çok etmenli sistemler yapay zeka, dağıtık yapay zeka, karar teorisi, ekonomik teoriler, nesne tabanlı programlama, sosyoloji, psikoloji, oyun teorisi gibi çok geniş alanı kapsayan tüm bilim dalları ile bağlantılı ve koordineli bir biçimde çalışmaktadır. Çok etmenli sistemlerin diğer bilim dalları ile çok yakından ilişkisi olan kombine yapısı, son yıllarda özellikle sanayide Endüstri 4.0 kavramı robot bilimi ile uygulamalı olarak bilgisayar algoritmaları ve yapay zeka teknikleri bir bütün halinde çalışabilmektedir. Aynı zamanda otomasyon ile birlikte stratejik kararların alınması ve uygulanması sürecinde, karmaşık bir problemi çözmek üzere çok etmenli yapı çerçevesi oluş-

turulabilir. Etmen teknolojisinin farklı alanlara etmenlerin dağıtık işlemlere veya nesne yönelimli yazılım uygulamalarının geliştirilmesine olanak tanımıştır. Bu çalışmanın uygulama kısmında, çok etmenli sistem ve tedarik zinciri uygulaması örneği ele alınmıştır. Çok etmenli sistemlerde, stratejik kararların alınması sürecinde Stackelberg oyun kuramı tercih edilmiştir. Özellikle birbirini takip eden olaylar zincirinde, olayların birbirlerine bağımlı olmasından dolayı, Stackelberg oyun kuramı ve uygulamasının bu tarz sistemler için daha uygun olduğu kabul edilmiştir.

Çok etmenli sistemlerin oyun teorisi ile birlikte bütünleşik yapısında eş zamanlı ve sıralı oyun teorisi yapısı ifade edilebilmektedir. Şu ana kadar yapılan çalışmalarda etmen ve çok etmenli sistemler, Nash oyun teorisi ele alınırken aynı zamanda aşağıda da belirttiğimiz gibi diğer oyun teorisi yaklaşımlarını da çok etmenli sistem modellerine uygulayabilme imkanı vardır. Diğer oyun teorisi modelleri ise;

- Pareto —Çok amaçlı / çok kriterli optimizasyon,
- Nash — stratejik / normal form eşzamanlı oyunlar,
- Stackelberg — stratejik / normal form sıralı oyunlar,
- Pareto-Nash-Stackelberg-üçlü bütünleşik yapı —Çok amaçlı, çok etmenli kontrol, eşzamanlı ve ardışık karar sürecinin bir karışımı.

Bu çalışmada, 2. bölümde çok etmenli karar yapısı detayları ile birlikte gösterilirken, 3. bölümde oyun teorisi, Stackelberg oyun teorisi ve matematiksel model kavramlarına yer verilmiştir. 4. bölümde ise, standart karar teorisi ve oyun teorisi yapısı ele alınmıştır. Yine bu bölümde alt başlık olarak Stackelberg oyun teorisi ve özellikleri ile Stackelberg etmen modeli incelenmiştir. Uygulama bölümünde ise Stackelberg etmen modeli ve tedarik zinciri yönetim modeline ait örnek yapısı uygulamalı olarak gösterilmiştir.

11.2 Çok Etmenli Öğrenme, Oyun Teorisi ve Karar Problemleri

Çok etmenli sistemlerin öğrenme yapısı ve karar verme sürecinin gerçekleşmesi esnasında dikkate alınması gereken yapı bu bölümde detaylı olarak açıklanmıştır. Öncelikli olarak etmen ve çok etmenli sistemlerin tanımını yapacak olursak;

“Etmen, temsil niteliğindeki hedeflerini gerçekleştirmek için bazı ortamlarda özerk eylemde bulunabilen bir bilgisayar sistemidir”.

“Çok etmenli sistemler bilgi temsili, iletişim, dağıtık planlama, koordinasyon, anlaşma ve öğrenme aktivitelerini belirlenen bir protokol ve bilgisayar yazılımı ile gerçekleştirirken aynı zamanda bağımsız ve zeki özellikleri açısından farklı bir algoritmaya sahiptirler”.

Çok etmenli sistemlerin bağımsızlık özellikleri ise dış dünyayı algılayabilmeleri ve öğrenme aktivitesi ile bağımsız strateji geliştirebilme özellikleri sayesinde yapay zeka alanında önemli bir konumdadırlar.

Bununla birlikte çok etmenli sistemler, esneklik özelliğinin yanı sıra ilave edilebilir, değiştirilebilir, yeniden yapılandırılabilir özelliklerine de sahiptir. Akıllı etmenler, sürekli olarak aşağıda yer alan üç işlevi yerine getirir, bu işlevler ise:

1. Çevredeki dinamik algılama;
2. Çevredeki koşulları etkilemek için eylem;
3. Algıları yorumlamak, problemleri çözmek, çıkarımlar ile eylemleri belirlemek için akıl yürütme. (Hayes-Roth 1995) "

"Akıllı etmenler, bir dereceye kadar bağımsızlık ya da özerklik ile bir kullanıcı ya da başka bir program adına bir takım operasyonlar yürüten yazılım varlıkları kullanıcının hedef veya isteklerinin bazı bilgi veya temsillerini kullanırlar." (IBM Agent)

Etmenler tüm bu bilgi sorgulama özelliklerini yerine getirirken, Bilgi Sorgulama Dili-KQML-Knowledge Query Machine Language) ve FIPA'nın etmen iletişim dili (ACL) 'yi kullanmaktadır(Colombetti, 2000).Bazı etkileşim, iletişimprotokollerine örnek verecek olursak eğer;

- Koordinasyon Protokolleri
- İşbirliği Protokolleri
- Protokol-Sözleşme
- Karatahta Sistemleri
- Görüşme
- Çok Etmenli İnanç-Yapısal Bakım
- Pazar Mekanizmaları

Oyun yapısı, oyun stratejileri ile birlikte etmenlerin ve oyun teorisi yapısının çok yakından bir bağlantılı olduğu ve temelde iç içe bir yapı ile birlikte temsil edilebildiği görülmektedir.

Oyun modelleri temelde işbirlikçi ve işbirlikli olmayan yapıya göre iki temel kategoride ele alınmaktadır. İşbirlikli model, çok etmenli yapı içerisinde ele alınarak değerlendirilmiş ve her bir etmenin birbirleri ile birlikte koordineli bir biçimde işlemleri yerine getiren yapısı dikkate alınmıştır. Oyun modellerinde aynı zamanda oyuncu durumuna göre işbirlikli olmayan yapıda oyuncu, sadece kendi stratejisine odaklanırken, işbirlikli yapıda ise, oyuncu koalisyon ve işbirliği halinde ortak strateji ile hareket edebilmektedir. Çok etmenli sistemlerde ise işbirliği halinde etmenlerin ortak bir yapı ile birlikte hareket etmeleri durumunda bu yapı daha kullanışlı ve uygun bir biçimde ifade edilebilir.

Oyunlar şans oyunları ve strateji oyunları şeklinde de ifade edilebilmektedir. Strateji oyunları ise oyuncu sayısına göre, strateji seviyesi ve sayısına göre, ödemeler toplamına göre ifade edilebilir. Oyuncu sayısına göre, iki kişilik oyunlar, ($n > 2$) çok kişilik oyunlar olarak ifade edildiğinde, iki kişilik oyunlar, kazanan ya da kaybeden şeklinde ifade edilebilmektedir. ($n > 2$) durumunda ise çok kişilik oyunlarda ise çok sayıda oyuncunun ortak bir strateji üzerinden karar vererek, sistem üzerinde, hareket etmesi olayı karşımıza çıkmaktadır.

Strateji sayısına göre sonlu oyunlar ve sonsuz stratejili oyunlar karşımıza çıkmaktadır. Burada oyuncuların ya da etmenlerin öğrenme durumlarına göre, eğer etmenlerin öğrenme yeteneği mevcut ise, etmenler sonsuz stratejiye sahip olabilmektedirler. Ödemeler toplamına göre oyun teorisinde ise, oyuncuların sabit toplamı oyunlar şeklinde gruplandırılabilirler. Sabit toplamı oyunlarda ise, elde edilen değerler sabit olmaktadır. Değişen koşul ve şartlarda bile elde edilen değerler, kazançlar sabit olurken, değişik toplamı oyunlarda ise elde edilen değerler, mevcut durum ve strateji değerlerine göre değişiklik göstermektedir.

Oyun teorisinde aynı zamanda, tek hedefli ve çok hedefli karar durumlarında uygulanan yöntemlere göre, tek hedefli oyun strateji ve tiplerine göre statik oyun; anlık oyun; Nash oyun dengesi, dinamik oyun, genişletilmiş oyun, tekrarlı oyun, Markovian oyun, Evolutionary oyun(evrimsel oyun), Stackelberg oyun, açık artırmalı oyun, kamu ürünleri oyunları, intervention (dönüşümlü) oyun, süpermodular oyun, güvenlik oyunu şeklinde ifade edilmektedir.

Karma stratejide farklı zamanlarda farklı stratejiler oyuncular tarafından seçilirken, oyun sürecinde daimi olarak aynı stratejinin seçildiği durumlar söz konusu olduğunda tam strateji vardır. Tam stratejide stratejilerden bir bütün oyun boyunca tercih edilecektir ve bu stratejinin oynanma olasılığı 1'dir.Diğer bütün stratejilerin oynanma olasılıkları ise sıfırdır.

Stackelberg oyun teorisi ve diğer modeller arasındaki fark eşzamanlı / Nash oyun kuralları, uygulanan bir dizi aksiyomun yerine getirildiğini varsayar ve stratejik form modelinde tüm oyuncular stratejilerini aynı anda ve güvenle seçerler. Tüm oyuncular tüm strateji ve ödeme fonksiyonları hakkında tam bilgiye sahiptirler; tüm oyuncular kendi değerlerini optimize eder (en üst düzeye

çıkartır veya en aza indirir); tüm strateji seçimlerinden sonra, her oyuncu kazancının değerini sonuç durumuna-formatına getirme işlevine sahiptir.

Nash oyun teorisi tek aşamalı bir oyun iken Stackelberg oyunu çok aşamalı bir oyundur. Stackelberg oyun teorisinde, strateji / eşzamanlı eşzamanlılık ilkesini değiştirerek hiyerarşik / sıralı bir ilkeye göre oyunlar, stratejileri bilinen bir sıraya göre davranırlar. Stackelberg oyunlarında bu tür oyunlara aynı zamanda sıralı oyunlar denir. Özellikle sıralı bir karar verme sürecini vurgulama esnasında her aşamada Stackelberg oyunları her bir oyuncu için ilgili stratejisini seçer. Optimalinin seçilmesini sağlamak için aynı zamanda optimizasyon problemi de çözülmelidir. Oyun dâhilinde her bir stratejinin belirli sonuçları vardır. Oyunun farklı zaman dilimlerinde farklı stratejileri olabilir ve bu değişen stratejilere göre sonuçlar da farklılık gösterir. Bu durumda oyun içerisinde karma stratejiler vardır. Karma stratejilerden her birinin oyunda oynanma sıklığına bağlı olarak, bir olasılık dağılımından bahsetmek mümkündür.

Çok hedefli oyun teorisinde ise, koalisyon oyun teorisi, pazarlık-görüşme durumlu oyun teorisi, karşılaştırmalı oyun teorisi, oylama oyun teorisi, gönderimli faydalı (transferable utility) oyun teorisi, gönderimsiz faydalı(non-transferable utility) oyun teorisi yaklaşımları da kullanılabilir. Oyun teorisi, bu çalışmada, etmen ve çok etmenli sistemlerde uygulama biçimi ve kullanım durumu ele alınmıştır. Oyun teorisinin çok etmenli sistemlerde uygulama biçimleri ve kullanım durumları ise;

- Nash eşitsizliği
- Korelasyon bağlantı durumu
- Bayesian Nash eşitsizliği
- Alt oyun, mükemmel Nash eşitsizliği, evolutinary stable stratejisi, Stackelberg eşitsizliği, Pareto eşitsizliği, Maximin'dir.

Etmen ve çok etmenli sistemler arasındaki bağlantı ise

- oyun oynama ve hamle,
- oyuncular tarafından bilinen bilgiler,
- oyuncu kriterleri ve hedefleri,
- getiriler ve bunların hesaplanması.

Bu tür kurallara dayanarak, çeşitli stratejik oyunlar düşünülebilir. Tüm oyuncuların getirilerini en üst düzeye çıkardığını varsayalım, o zaman, nesnel optimizasyon problemi, stratejik form oyununun bir çözümünü bulmak anlamına gelir.

Stackelberg oyun teorisi ile tedarik zinciri yönetimi konusunda bazı çalışmalar yapılmış olmasına rağmen çok etmenli sistemlerdeki uygulamalara pek değinilmemiştir. Bu kapsamda Stackelberg oyun teorisinin bazı ekonomik senaryolara uygulaması ve tedarik zinciri uygulamalarına örnekler verecek olursak eğer, Parsaeifar ve arkadaşları oyun teorisini fiyatlandırma ve geridönüşüm sürecindeki tedarik zinciri modellerine uygulamışlardır. Chen ve arkadaşları Stackelberg oyun kuramını tedarik zinciri yönetiminde perakendecilerin davranışlarını analiz etmede kullanmışlardır. Ghalekhondabi ve arkadaşları tehlikeli atık yönetiminde tedarik zinciri modelini ele almışlardır. Sabari ve arkadaşları Stackelberg oyun teorisini, küçük ölçekli perakendecilerin online üzerinden satış yapmaları esnasında ürün fiyatlarının belirlenmesi sürecinde ele almışlardır. Sharma ve arkadaşları oyun teorisini ve Nash oyun dengesini iki katmanlı tedarik zinciri modeli üzerinde ele alarak değerlendirmişlerdir. Halat ve Hafezalkotob, çok katmanlı yeşil-çevreye duyarlı tedarik zinciri modelinde yatırım kararlarının karbon salınımı düzenleyici kararların alınması sürecinde geliştirilen politikaların analiz edilmesinde oyun teorisi yaklaşımını kullanılmıştır. Gao ve You, işbirlikli olmayan oyun teorisinde stokastik model yapısını ele alarak, tek liderli çok takipçili oyun teorisi yapısını modellemişlerdir.

11.3 Stackelberg Oyun Teorisi Ve Matematiksel Model

Oyun teorisi, oyuncular arasında, çatışma ve işbirliği durumlarında karar verme modelleri ile matematiksel bir teori olarak baskın özellikler karakterize edilmektedir. Aynı zamanda etmen ve çok etmenli sistemlerde, stratejiler ve karma stratejiler, herhangi bir oyunda bir oyuncunun karma strateji olasılık vektörü şu şekilde gösterilebilir:

$$X = x_1, x_2, \dots, x_i, \dots, x_n$$

burada karma stratejileri A_1, A_2, \dots, A_n şeklinde gösterecek olursak, x_1, x_2, \dots, x_n bu stratejilerin oynanma olasılıklarını gösterir. Bu durumda, $x_i \geq 0 (i = 1, 2, \dots, n)$ ve $\sum_{i=1}^n x_i = 1$ eşitliklerini yazabiliriz.

Stratejik Oyun Formu

- $N = \{1, 2, \dots, n\} \subset N$ bir grup oyuncudur,
- $s_p, p \in N$ oyuna ait strateji kümesidir ,
- $f_p(n)$, p inci oyuncu ödeme fonksiyonu olan kartezyen çarpım $S = x_{p \in N}$
- s_p alan adıdır.
- $\Gamma = \left\langle N, \{S_p\}_{p \in N}, \{f_p(s)\}_{p \in N} \right\rangle$,

Herhangi bir $s_1 = (s_1, s_2, \dots, s_n) \in S$ elementine strateji profili denir. Stratejik bir form oyunu, işbirlikçi olmayan oyunlar için düşünülen basit bir modeldir. Model üç temel unsur içerir: oyuncu kümesi, oyuncu strateji setleri ve oyuncu ödeme fonksiyonları. Resmi olarak, strateji formu oyunu belirtilir. Burada karma stratejileri s_1, s_2, \dots, s_n şeklinde gösterecek olursak, $f_p(s(n))$ s_1, s_2, \dots, s_n bu stratejilerin oynanma olasılıklarını gösterir. Ve bu durumda, seçilirken, oyun sürecinde daimi olarak aynı stratejinin seçildiği durumlar söz konusu olduğunda tam strateji vardır. Tam stratejide stratejilerden biri bütün oyun boyunca tercih edilecektir ve bu stratejinin oynanma olasılığı 1'dir. Diğer bütün stratejilerin oynanma olasılıkları ise sıfırdır.

Stackelberg oyununda oyuncular stratejilerini sırayla seçerler: ilk oyuncu $s_1 \in S_1$ stratejisini seçer ve ikinci oyuncuya seçimi hakkında bilgi verir. Daha sonra ikinci oyuncu stratejisi seçer $s_2 \in S_2$ ve bilgi verir üçüncü oyuncu s_1, s_2 vb. seçenekler hakkında, n. aşama sonunda, n inci oyuncu stratejisi seçer ve $n \in S_n, n-1$, önceki oyuncu, s_1, \dots, s_n seçenekleri ve oyun hakkında bilgi sahibi olur. Her oyuncu emsal / lider oyuncuların ve tüm oyuncuların takipçilerin strateji kümeleri ve ödeme fonksiyonları hakkında bilgi sahibidir. Tüm oyuncular kendi değerlerini optimize eder (en üst düzeye çıkarır veya en aza indirir). Tüm strateji seçimlerinden sonra, her oyuncu kazancının değerini sonuç profilinde getirme işlevine sahiptir.

$F(S)$ kümesi, S etki alanının görüntüsüdür. Herhangi bir öge

$$f(s) = \left(f_1(s), f_2(s), \dots, f_n(s) \right) \in f(s) \quad (11.1)$$

getirilerin profili olarak adlandırılır. Stratejik form, her bir durum ile ilişkilendirilen bir vektör fonksiyonu olarak görülebilir. Stratejileri s değişkeni ile gösterirken,

fonksiyon biçiminde ise

$f(s) = (f_1(s), f_2(s), \dots, f_n(s))$ gösterebiliriz. Böylece, stratejik formu bir vektör fonksiyonu olarak vurgulayabiliriz:

$f : S \rightarrow f(S)$. Bir oyun, kuralları belirtildiğinde tanımlanır. Bu kurallar aşağıdakilerle ilgili ifadeleri içerir:

$$f(x) = \sum_p^{m_1} \sum_{s_1=1}^{m_2} \dots \sum_{s_n=1}^{m_n} a^p_{s_1 s_2 \dots s_n} \prod_{p=1}^n x_{s_p} \quad (11.2)$$

Etmen davranışları esnasında dengenin sağlanması ve ortak stratejiler ile uygun kararların alınması oldukça önemlidir. Etmenlerin fonksiyonlarını ve görevlerini yerine getirmeleri esnasında karşılaşılabilecekleri tehlikeli durumlar ve tehditler de olabilir. Bu tarz olası durumlarda ise etmen önceden belirlenmiş olan acil durum stratejisini kullanabilmektedir.

Çok etmenli sistemlerde Stackelberg oyun teorisinin uygulaması, stratejilerin belirlenmesi, olası karşılaşılabilecek durumların belirlenmesi, etmenlerin belirlenen tanım ve strateji durumları ile birlikte stratejik kararlarını gerçekleştirmeleri ile birlikte ortak ya da rakip etmen karar durumuna göre hareket etmesi durumu karşımıza çıkmaktadır.

$$f(x) = \sum_{s_1=1}^{m_1} \sum_{s_2=1}^{m_2} \dots \sum_{s_n=1}^{m_n} a^p_{s_1 s_2 \dots s_n} \prod_{p=1}^n x_{s_p} \quad (11.3)$$

1. Oyuncu gerekli olan $x^1 \in X_1$ stratejisini seçer. 2. Oyuncu gerekli olan $x^2 \in X_2$ stratejisini seçer. $x^1 x^2$ seçeneğinden birini seçer.

Oyuncuların profilini şu şekilde $x = (x^1, \dots, x^n) \in X$ özetleyebiliriz ve oyuncuların elde etmiş oldukları kazanç durumu ise, $f_p(x)$, $p = 1, \dots, n$
Stackelberg oyununda uygulanan ikili matris fonksiyonu ise;

$$\hat{r} = \left(X, Y; f_1, f_2 \right) \quad (11.4)$$

İle gösterilmiştir. Aynı zamanda Algoritma 1'de ise Stackelberg oyun etmen ilişki döngüsüne yer verilmiştir.

Ödeme matrisleri olarak

$$A = \begin{pmatrix} a \\ ij \end{pmatrix}, B = \begin{pmatrix} b \\ ij \end{pmatrix}, i = 1, \dots, m, j = 1, \dots, n \quad (11.5)$$

$a^1, j = 1, \dots, n$, A matrisinin sütun değerlerini,

$b^1, j = 1, \dots, n$, B matrisinin sütun değerlerini göstermektedir.

Bu durumda, güvenilir ve güvenilmeyen durumlar için Eşitlik (6-7)'de tanımlamalar yapılmıştır.

$$X = \left\{ x \in \mathbb{R}^m : x_1 + x_2 + \dots + x_m = 1 \right\} \quad (11.6)$$

$$Y = \left\{ y \in \mathbb{R}^n : y_1 + y_2 + \dots + y_n = 1 \right\} \quad (11.7)$$

Algoritma.1. Stackelberg oyun-etmen ilişki döngüsü

$$\begin{aligned}
&V = \{j \in \{1, 2, \dots, n\}; X^j \neq \emptyset\}; k = V[1]; \\
&\mu = \mu_k; USES = \text{Arg max}_{x \in X^k} x^T a^k \times e^{y^k}; \\
&\text{for } j \in V \text{ do} \\
&\quad \{ \\
&\quad \text{for } J \in 2^{V \setminus \{j\}} \text{ do} \\
&\quad \quad \text{case } (XY^{JJ} \neq \emptyset) \text{ ve } (\mu_{JJ} > \mu) \text{ then } \mu = \mu_{JJ}, \\
&\quad \quad USES = \text{Arg max}_{(x,y) \in XY^{JJ}} \sum_{k \in J \cup \{j\}} (x^T a^k) y_k; \\
&\quad \quad \text{case } (XY^{JJ} \neq \emptyset) \text{ ve } (\mu_{JJ} > \mu) \text{ then} \\
&\quad \quad \quad USES = USES \cup \text{Arg max}_{(x,y) \in XY^{JJ}} \sum_{k \in J \cup \{j\}} (x^T a^k) y_k; \\
&\quad \quad \} \\
&\}
\end{aligned}$$

Oyuncuların, stratejilerin ve karar durumlarının aşağıdaki koşulu yerine getirmesi gerektiğini kabul ediyoruz:

$$\vec{R}_{\geq}^m = \left\{ x \in \vec{R}^m : x_1 \geq 0, x_2 \geq 0, \dots, x_m \geq 0 \right\} \quad (11.8)$$

$$\vec{R}_{\geq}^n = \left\{ y \in \vec{R}^n : y_1 \geq 0, y_2 \geq 0, \dots, y_n \geq 0 \right\} \quad (11.9)$$

Stackelberg oyun matrisi ise;

$$\vec{r} = \left(X, Y; f_1, f_2 \right) \quad (11.10)$$

Stackelberg eşitliğinde karma stratejilerin ikili matris ile ifade edilmesi durumu
Karma stratejiler ise;

$$f_1(x, y) = \begin{pmatrix} T & 1 \\ x & a \end{pmatrix}_1 y + \begin{pmatrix} T & 2 \\ x & a \end{pmatrix}_2 y + \dots + \begin{pmatrix} T & n \\ x & a \end{pmatrix}_n y \quad (11.11)$$

$$f_2(x, y) = \begin{pmatrix} T & 1 \\ x & b \end{pmatrix}_1 y + \begin{pmatrix} T & 2 \\ x & b \end{pmatrix}_2 y + \dots + \begin{pmatrix} T & n \\ x & b \end{pmatrix}_n y \quad (11.12)$$

$$f_2(x, y) \rightarrow \max, y \in Y \quad (11.13)$$

Algoritma hesaplama işlemi $O\left(C_{2m+n+2}^{m+n} (m+n)^3 (2^n - 1)\right)$ kadar gerçekleştirilir. $USES(\vec{r})$ ve $SSES(\vec{r}) 2^n - 1$ bileşen için gerçekleştirilir. XY^{JJ} hesaplaması C_{2m+n+2}^{m+n} durumu için elde edilir.

11.4 Uygulama

Çok etmenli sistemlerde Stackelberg oyun teorisinin uygulaması, stratejilerin belirlenmesi, olası karşılaşılabilecek durumların belirlenmesi, etmenlerin belirlenen tanım ve strateji durumları ile birlikte stratejik kararlarını gerçekleştirmeleri ile birlikte ortak ya da rakip etmen karar durumuna göre hareket etmesi durumu karşımıza çıkmaktadır.

Etmen ve çok etmenli sistemlerin uygulama alanında tedarik zinciri yönetim modeli ele alınmıştır. Tedarik zinciri yönetim modelinde sistemi oluşturan tüm bileşenler etmenler ile ifade edilmiştir. Etmenlerin temelde birbirleri ile olan ilişki yapısında ise, etmen, çevresel faktörler, organizasyonel ilişkisi, birbirleri ile olan ilişki yapısı ele alınarak değerlendirilmiştir. Şekil 1’de etmen ve çok etmenli sistemlerin organizasyon içerisindeki davranış yapısının gösterimi yer almaktadır.

Stackelberg- Etmen Modelinde sistemi koordine etmek için Stackelberg oyununu kullanıyoruz çünkü;

- Lider ve takipçisi olan hiyerarşik bir oyun
- Lider önce hareket eder
- Takipçi, liderin davranışını gözlemler ve kendi stratejisini belirler
- Lider, takipçinin liderin stratejisine tepki vereceğini önceden bilir.

Önerilen sistem içerisinde

- etmenler içerisinde bilginin toplanması ve dağıtılması
- herbir etmenin davranış tipinin belirlenmesi
- etmenler arasında ortak kullanılan bilgi tiplerinin belirlenmesi
- etmenlerin herbirinin sorgu geçiş işleminin gerçekleştirilmesi
- sisteme yeni bilgi ile birlikte yeni kural yapısının ilave edilmesi, kural yapısının güncellenmesi ve gerektiğinde kullanılmayan kuralların ortadan kaldırılması.

Kısacası, sisteme bilgi gelmesi, bilginin işlenmesi, bilginin gönderilmesi işlemleri aynı zamanda karar işlemi ile birlikte gerçekleştirilir.

Stackelberg oyun kuramında etmen tanımlamaları, etmenin fonksiyonları ve davranış tiplerinin belirlenmesi oldukça önemlidir. Tanımlanmış olan etmen yapısı ile birlikte etmen davranışlarının analiz edilmesi ve etmen davranış biçimleri Stackelberg oyun kuramının temelini oluşturmaktadır. Etmen davranışları temelde bağımlı ve bağımsız biçiminde gruplandırılabilir. Bununla birlikte, etmenlerin ortak bir stratejiye göre hareket etmeleri ve bu stratejinin gerçekleştirilmesi esnasında, kendi menfaatlerini koruma durumları da söz konusu olabilmektedir.

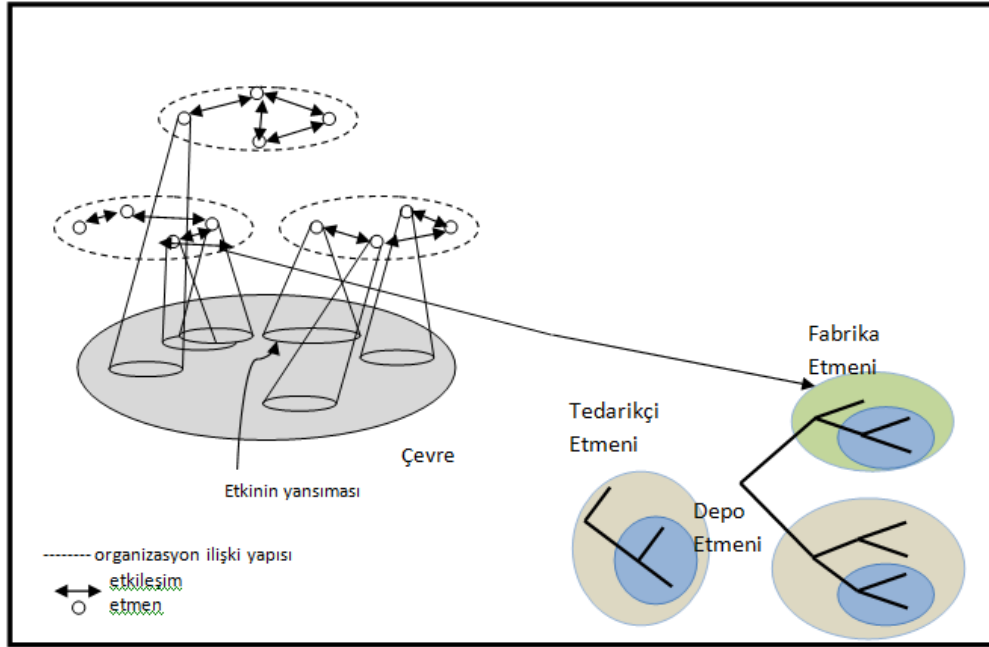
Her yarışmacı, aşağıdaki görevlerden sorumlu bir temsilciye girer:

1. Tedarik sözleşmelerini müzakere edin
2. Müşteri siparişleri için teklif verin
3. Günlük montaj faaliyetlerini yönetin.

Bu üç görev, ajan tarafından günlük olarak gerçekleştirilir. Müşteri temsilcisi, temel olarak talebin türünü (periyodik veya sürekli) ve ayrıca doğayı (deterministik veya stokastik) belirleyen talep kontrol politikasına dayalı ürün talepleri oluşturur.

İki etmen arasında farklı strateji durumlarının değerlendirilmesi sürecinde yada farklı alternatif durumlarının değerlendirilmesi sürecinde bu yaklaşım kullanılabilir ve değerlendirilebilir.

1. Satış Politikası-Pazar noktaları
2. Taşıma alternatif- Nakliye Miktarları
3. Stok Politikası-Stok Seviyesi-
4. Üretim Seviyesi-Kapasite
5. Depolama-Miktar-Kapasite



Şekil 11.1: Etmen ve çok etmenli sistemlerin organizasyon içerisinde birbirleri ile olan etkileşimi

Durumlarının tedarik zinciri yönetimine karşılaştırma strateji alternatifleri olarak göz önünde bulundurulup çözüm sürecinde uygulanması süreci bu çalışmada önerilen yaklaşım ile çözülmesi mümkündür. Şekil 2’de ise etmenler ve tedarik zinciri yönetimde yer alan veritabanlarının yapısına yer verilmiştir. Tedarikçi etmeninin kullanmış olduğu dosya yapısı, Sipariş No, Sipariş Tarihi ve Miktar ve Parça No parametrelerine bağlı iken Fabrika etmeninde ise, İmalatçı No, İmalatçı Adı gibi imalatçı bilgileri, Parça No, Parça Adı ve Miktarı, Ürün Satış Fiyatı, Ürün Stok No gibi bilgileri yer almaktadır. Depo etmeni ise Depo Stok No, Depolanan Ürün Adı, Stok Miktarı Bilgileri ve perakendeci bilgilerine yer verilmiştir.

Çok etmenli yapıda tedarik zinciri yönetim modelinin gösterim biçiminde ise Şekil 3’de yer alan Stackelberg tabanlı etmen yaklaşımına ait durum olay yapısına yer verilmiştir. Etmenlerin kazanım durumları ve birbirlerinden elde etmiş oldukları fayda ve kazanım durumları ele alınarak ifade edilmiştir.

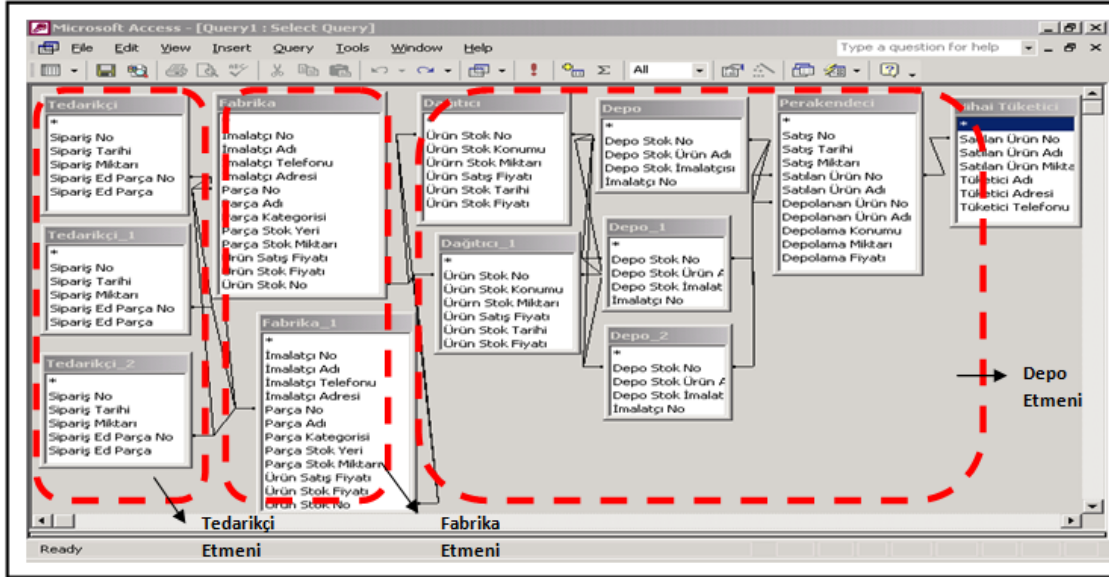
Uygulama çözümünde Şekil 4’te yer alan etmenlerin matris formunda gösterimi yer almaktadır.

Oyun stratejisine ait maliyet fonksiyonu ise;

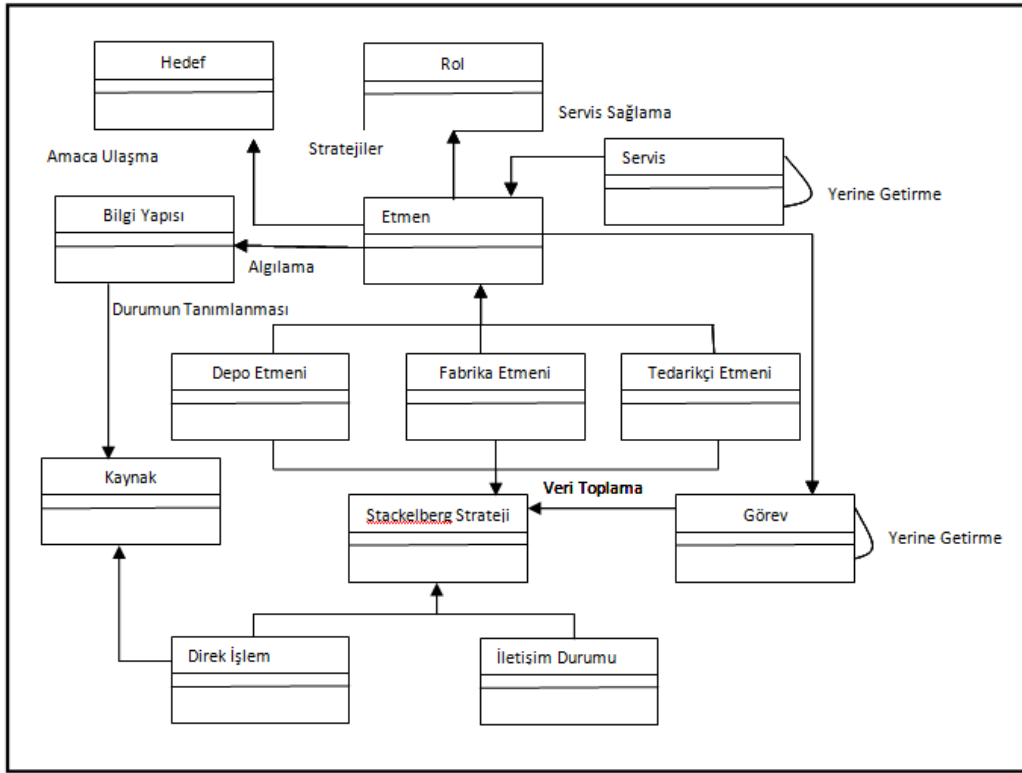
$$f_1(x, y) = \begin{pmatrix} 3x+2x \\ 1 \end{pmatrix} y_1 + \begin{pmatrix} 4x+4x \\ 1 \end{pmatrix} y_2 + \begin{pmatrix} 4x+4x \\ 1 \end{pmatrix} y_3$$

,

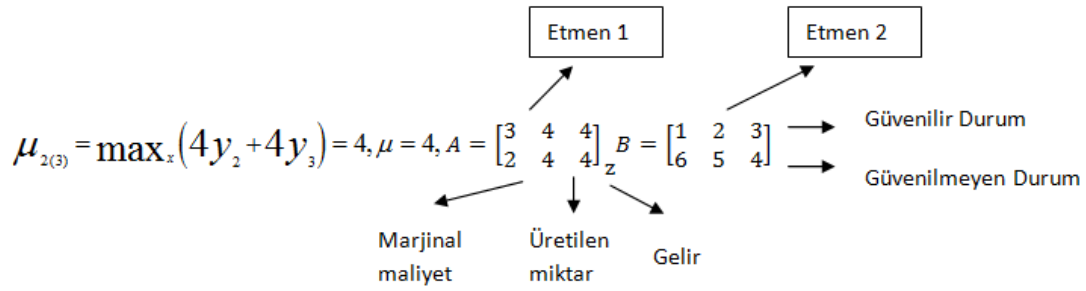
$$f_2(x, y) = (x_1 + 6x_2)y_1 + (2x_1 + 5x_2)y_2 + (3x_1 + 4x_2)y_3, XY^{jJ} \text{ ve } j = 1 \text{ olması durumunda,}$$



Şekil 11.2: Tedarik Zinciri Yönetiminde yer alan veritabanlarının yapısı



Şekil 11.3: Tedarik zinciri yönetiminde Stackelberg oyun tabanlı çok etmenli sistemde durum-olay yapısı



Şekil 11.4: Etmenlerin matris biçiminde gösterimi

$${}^{1(\emptyset)}XY = \left\{ (x, y) \in \mathbb{R}^2 \times \mathbb{R}^3; \begin{array}{l} x_1 - x_2 \leq 0, \\ 2x_1 - 3x_2 \leq 0, \\ x_1 + x_2 = 1, x_1 \geq 0, x_2 \geq 0, \\ y_1 + y_2 + y_3 = 1, \\ y_1 = 0, y_2 \geq 0, y_3 \geq 0, \end{array} \right\}$$

$$= \left(\begin{array}{l} 0 \leq x_1 \leq \frac{1}{2} \\ 1 - x_1 \end{array} \right) \times \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$v = v = \max_{1\emptyset} \max_x \max_y \left(\begin{array}{l} 3x + x \\ 1 \quad 2 \end{array} \right) = 2,$$

$$SSES = \left\{ \left(\begin{array}{l} \frac{1}{2} \\ \frac{1}{2} \end{array} \right) \times \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right\}$$

$$(x, y) \in {}^{1\emptyset}XY$$

$$\mu = \mu = \max_{1\emptyset} \max_x \max_y \left(\begin{array}{l} 3x + x \\ 1 \quad 2 \end{array} \right) = 2,$$

$$USES = \left\{ \left(\begin{array}{l} \frac{1}{2} \\ \frac{1}{2} \end{array} \right) \times \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right\},$$

$$(x, y) \in {}^{1\emptyset}XY$$

$${}_{1(2)}XY = \left\{ (x, y) \in R^2 \times R^3; \begin{array}{l} x_1 - x_2 \leq 0, \\ 2x_1 - 3x_2 \leq 0, \\ x_1 + x_2 = 1, x_1 \geq 0, x_2 \geq 0, \\ y_1 + y_2 + y_3 = 1, \\ y_1 = 0, y_2 \geq 0, y_3 = 0, \end{array} \right\}$$

$$= \left\{ \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \times \begin{pmatrix} 0 \leq y_1 \leq \frac{1}{2} \\ 1 - x_1 \\ 0 \end{pmatrix} \right\}$$

$$v = \max_x \max_y \begin{pmatrix} 2y + 4y \\ 1 \quad 2 \end{pmatrix} = 2$$

$v = 2, (x, y) \in XY^{1(2)}$ olması durumunda SSES değerleri değişmez

$$\mu = \max_x \begin{pmatrix} 2y + 4y \\ 1 \quad 2 \end{pmatrix} = 4$$

$\mu > 2$, olması durumunda $\mu = 4$ değeri atanır ve $USES = \left\{ \begin{pmatrix} \frac{1}{2} \leq x_1 \leq \frac{2}{3} \\ 1 - x_1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right\}$

$${}_{1(3)}XY = {}_{1(2,3)}XY = \emptyset$$

XY^{1j} değeri ile birlikte $j = 2$ değeri ele alınarak

$${}_{2(\emptyset)}XY = \left\{ (x, y) \in R^2 \times R^3; \begin{array}{l} -x_1 + x_2 \leq 0, \\ x_1 - 2x_2 \leq 0, \\ x_1 + x_2 = 1, x_1 \geq 0, x_2 \geq 0, \\ y_1 + y_2 + y_3 = 1, \\ y_1 = 0, y_2 \geq 0, y_3 \geq 0, \end{array} \right\}$$

$$= \left\{ \begin{pmatrix} \frac{1}{2} \leq x_1 \leq \frac{2}{3} \\ 1 - x_1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right\}$$

$$v = \max_x \max_y \begin{pmatrix} 4x + 4x \\ 1 \quad 2 \end{pmatrix} = 4$$

$$(x, y) \in {}_{2(\emptyset)}XY$$

$$\begin{matrix} v > 2 \\ 2\emptyset \end{matrix}$$

$v = 4$ olması durumunda,

$$SSES = \left\{ \left(\begin{array}{c} \frac{1}{2} \leq x_1 \leq \frac{2}{3} \\ 1 - x_1 \end{array} \right) \times \left(\begin{array}{c} 0 \\ 1 \\ 0 \end{array} \right) \right\}, \mu = \max_x \left(4x + \frac{4x}{2} \right) = 4, \mu = 4$$

$$(x, y) \in {}^{2\emptyset}XY, USES = \left\{ \left(\begin{array}{c} \frac{1}{2} \leq x_1 \leq \frac{2}{3} \\ 1 - x_1 \end{array} \right) \times \left(\begin{array}{c} 0 \\ 1 \\ 0 \end{array} \right) \right\}$$

$${}^{2(3)}XY = \left\{ (x, y) \in R^2 \times R^3; \begin{array}{l} -x_1 + x_2 \leq 0 \\ x_1 - 2x_2 \\ x_1 + x_2 = 1, x_1 \geq 0, x_2 \geq 0, \\ y_1 + y_2 + y_3 = 1, \\ y_1 = 0, y_2 \geq 0, y_3 \geq 0, \end{array} \right\}$$

$$= \left\{ \left(\begin{array}{c} \frac{2}{3} \\ \frac{1}{3} \end{array} \right) \times \left(\begin{array}{c} 0 \\ 0 \leq y_2 \leq 1 \\ 1 - y_2 \end{array} \right) \right\}$$

,

$${}^{2(3)}v = \max_x \max_y \left(4y + \frac{4y}{3} \right) = 4$$

,

$$v = 4, (x, y) \in {}^{2(3)}XY$$

$$SSES = \left\{ \left(\begin{array}{c} \frac{2}{3} \\ \frac{1}{3} \end{array} \right) \times \left(\begin{array}{c} 0 \\ 0 \leq y_2 \leq 1 \\ 1 - y_2 \end{array} \right) \right\} \cup \left\{ \left(\begin{array}{c} \frac{1}{2} \leq x_1 \leq \frac{2}{3} \\ 1 - x_1 \end{array} \right) \times \left(\begin{array}{c} 0 \\ 1 \\ 0 \end{array} \right) \right\}$$

$${}^{2(3)}\mu = \max_x \left(4y + \frac{4y}{3} \right) = 4, \mu = 4,$$

$$(x, y) \in {}^{2(3)}XY$$

$$USES = \left\{ \left(\begin{array}{c} \frac{2}{3} \\ \frac{1}{3} \end{array} \right) \times \left(\begin{array}{c} 0 \\ 0 \leq y_2 \leq 1 \\ 1 - y_2 \end{array} \right) \right\} \cup \left\{ \left(\begin{array}{c} \frac{1}{2} \leq x_1 \leq \frac{2}{3} \\ 1 - x_1 \end{array} \right) \times \left(\begin{array}{c} 0 \\ 1 \\ 0 \end{array} \right) \right\}$$

$${}^{3\emptyset}XY = \left\{ (x, y) \in R^2 \times R^3 : \begin{array}{l} -2x_1 + 3x_2 \leq 0, \\ -x_1 + 2x_2 \leq 0, \\ x_1 + x_2 = 1, x_1 \geq 0, x_2 \geq 0, \\ y_1 + y_2 + y_3 = 1, \\ y_1 = 0, y_2 = 0, y_3 \geq 0, \end{array} \right\}$$

$$= \left(\begin{array}{l} \frac{2}{3} \leq x_1 \leq 1 \\ 1 - x_1 \end{array} \right) \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, v = \max_{3\emptyset} \min_x \max_y (4x + 4x) = 4, v = 4,$$

$$(x, y) \in {}^{3\emptyset}XY$$

$$SSES = \left\{ \begin{pmatrix} \frac{2}{3} \\ \frac{1}{3} \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \leq y_2 \leq 1 \\ 1 - y_2 \end{pmatrix} \right\} \cup \left\{ \begin{pmatrix} \frac{1}{2} \leq x_1 \leq \frac{2}{3} \\ 1 - x_1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right\} \cup \left\{ \begin{pmatrix} \frac{2}{3} \leq x_1 \leq 1 \\ 1 - x_1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}$$

$$USES = \left\{ \begin{pmatrix} \frac{2}{3} \\ \frac{1}{3} \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \leq y_2 \leq 1 \\ 1 - y_2 \end{pmatrix} \right\} \cup \left\{ \begin{pmatrix} \frac{1}{2} \leq x_1 \leq \frac{2}{3} \\ 1 - x_1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right\} \cup \left\{ \begin{pmatrix} \frac{2}{3} \leq x_1 \leq 1 \\ 1 - x_1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}$$

Sonuç olarak;

$$SSES = USES = \left\{ \begin{pmatrix} \frac{2}{3} \\ \frac{1}{3} \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \leq y_2 \leq 1 \\ 1 - y_2 \end{pmatrix} \right\} \cup \left\{ \begin{pmatrix} \frac{1}{2} \leq x_1 \leq \frac{2}{3} \\ 1 - x_1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right\} \cup \left\{ \begin{pmatrix} \frac{2}{3} \leq x_1 \leq 1 \\ 1 - x_1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}$$

$$A = \begin{bmatrix} 2 & 1 & 3 \\ 3 & 2 & 6 \\ 3 & 2 & 5 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 & 3 \\ -1 & 1 & -2 \\ 2 & -1 & 3 \end{bmatrix}$$

Maliyet fonksiyonu

$$f_1(x, y) = \begin{pmatrix} 2x + 3x + 3x \\ 1 \end{pmatrix} y + \begin{pmatrix} x + 2x + 2x \\ 1 \\ 2 \end{pmatrix} y + \begin{pmatrix} 3x + 6x + 5x \\ 1 \\ 2 \\ 3 \end{pmatrix} y$$

$$f_2(x, y) = \begin{pmatrix} 2x + 3x + 3x \\ 1 \\ 2 \end{pmatrix} y + \begin{pmatrix} x - x \\ 2 \\ 3 \end{pmatrix} y + \begin{pmatrix} 3x - 2x + 3x \\ 1 \\ 2 \\ 3 \end{pmatrix} y$$

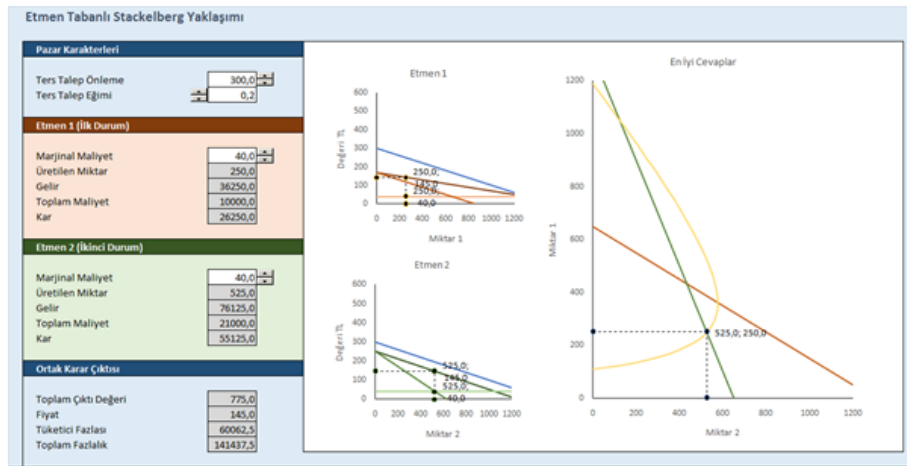
X^{jj} durumunu uygulama sonucunu Şekil 5'te görebiliriz.

Güvenilir ve güvenilmeyen Stackelberg oyun yaklaşım durumunu Excel ortamında değerlendirilmesi sonucunda Şekil 6'da elde edilen ekran çıktısını görebilmemiz mümkündür. Etmen tabanlı Stackelberg yaklaşımının Excel ortamında çözüm biçiminde ise, pazar yapısı ve iki etmen arasındaki farklı durum ve stratejilerin değerlendirilmesi sonucunda elde edilen ortak karar çıktısı görülmektedir.

$$SSES = USES = \left\{ \begin{pmatrix} 1 \\ 4 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\} = \begin{pmatrix} 19 \\ 4 \end{pmatrix}, \begin{pmatrix} 1 \\ 4 \end{pmatrix}$$

↑ Güvenilirdurum
↓ Güvenilmeyen durum

Şekil 11.5: A ve B etmenlerine ait sonuçlar



Şekil 11.6: Stackelberg etmen yapısı, pazar durumu ve 2 farklı etmenin farklı strateji yapıları

11.5 Sonuç

Bu çalışmada amaç, çok etmenli sistemlerin özelliklerinin yanı sıra çeşitli uygulama alanlarında, karar verme süreci içerisinde oyun teorisinin kullanılabilirliğini göstermektir. Özellikle bağımsız karar durumlarının yanı sıra etmenlerin birbirleri ile olan ilişkilerinde, sıralı davranışları ve bir önceki durumdan sonuç çıkartarak mevcut durum için karar stratejisinin belirlenmesi esnasında aynı zamanda öğrenme işlemi de gerçekleştirilebilmesi durumudur.

Oyun teorisi içerisinde etmenlerin strateji yapılarının matematiksel gösterim biçimi ise;

Oyun

- Etmenlerin(oyuncuların) aralarında oynamış oldukları stratejik davranış biçimi ve her bir etmenin temel rolünü oyun içerisinde gerçekleştirilmesi durumu

Strateji

- Etmenin karakteristik özelliklerini oluşturan aktivite tipleri
- Diğer etmenlere göre yapabileceği farklı eylemler, aktiviteler

Karar verme, genellikle çözülen bir seçim sorunu olarak yorumlanabilir hayatının farklı anlarında olmak. Sezgisel olarak, en basit karar sorun yaratma, bir dizi kabul edilebilir karardan bir karar seçmektir.

Çok etmenli sistemlerde, stratejik kararların verilmesi, değişen durumların analiz edilerek stratejilerin belirlenmesi durumu sürekli olarak gerçekleştirilecek eylemlerin başında gelmektedir. Bu kapsamda sistemde yer alan etmen yapısında bir de karar verici mekanizmanın bulunması gerekmektedir.

Stackelberg oyun kuramında etmen tanımlamaları, etmenin fonksiyonları ve davranış tiplerinin belirlenmesi oldukça önemlidir. Tanımlanmış olan etmen yapısı ile birlikte etmen davranışlarının analiz edilmesi ve etmen davranış biçimleri Stackelberg oyun kuramının temelini oluşturmaktadır. Etmen davranışları temelde bağımlı ve bağımsız biçiminde gruplandırılabilir. Bununla birlikte, etmenlerin ortak bir stratejiye göre hareket etmeleri ve bu stratejinin gerçekleştirilmesi esnasında, kendi menfaatlerini koruma durumları da söz konusu olabilmektedir. Etmen davranışları esnasında dengenin sağlanması ve ortak stratejiler ile uygun kararların alınması oldukça önemlidir. Etmenlerin fonksiyonlarını ve görevlerini yerine getirmeleri esnasında karşılaşılabilecekleri tehlikeli durumlar ve tehditler de olabilir. Bu tarz olası durumlarda ise etmen önceden belirlenmiş olan acil durum stratejisini kullanabilmektedir.

Referanslar

Chen, X. Zhang, H., Zhang, M., Chen, J., Optimal decisions in a retailer Stackelberg supply chain, *International Journal of Production Economics* 187 (2017) 260–270

Colombetti M. (2000) Semantic, Normative and Practical Aspects of Agent Communication. In: Dignum F., Greaves M. (eds) *Issues in Agent Communication. Lecture Notes in Computer Science*, vol 1916. Springer, Berlin, Heidelberg. https://doi.org/10.1007/10722777_2

Gao, J., You, F., A stochastic game theoretic framework for decentralized optimization of multi-stakeholder supply chains under uncertainty, *Computers and Chemical Engineering* 122 (2019) 31–46

Ghalekhondabi, İ, Maihami, R., Ahmadi, E. , Optimal pricing and environmental improvement for a hazardous waste disposal supply chain with emission penalties, *Utilities Policy* 62 (2020) 101001

Halat, K., Hafezalkotob, A. Modeling carbon regulation policies in inventory decisions of a

multi-stage green supply chain: A game theory approach, *Computers & Industrial Engineering* 128 (2019) 807–830

Hayes-Roth, B., An architecture for adaptive intelligent systems, *Artificial Intelligence*, Vol. 72, Is.1-2, January, 1995, pp. 320-365. Parsaeifar, S., Bozorgi-Amiri, A., Naimi-Sadigh, A., Sangari, M.S., A game theoretical for coordination of pricing, recycling, and green product decisions in the supply chain, *Journal of Cleaner Production* 226 (2019) 37e49

Saberi, Z., Saberi, M., Hussain, O., Chang, E. Stackelberg model based game theory approach for assortment and selling price planning for small scale online retailers *Future Generation Computer Systems* 100 (2019) 1088–1102

Sharma, A., Dwivedi, G., Singh, A., Game-theoretic analysis of a two-echelon supply chain with option contract under fairness concerns, *Computers & Industrial Engineering* 137 (2019) 106096

Ungureanu, V., *Pareto-Nash-Stackelberg Game and Control Theory, Intelligent Paradigms and Applications*, Springer, (2018).

https://www.ibm.com/support/knowledgecenter/SSCT62/com.ibm.iamservice.doc/references/r_identity_agent.html

Yazarlar Hakkında



Doç. Dr. Safiye Turgay lisans derecesini, İstanbul Teknik Üniversitesi Endüstri Mühendisliği, yüksek lisans ve doktora derecesini Sakarya Üniversitesi Fen Bilimleri Enstitüsü Endüstri Ana Bilim dalından almıştır. Abant İzzet Baysal Üniversitesi Bilgisayar Programcılığı, Bilgisayar ve Öğretim Teknolojileri ve Eğitimi, İşletme, Sakarya Üniversitesi Yönetim Bilişim Sistemleri bölümlerinde öğretim üyeliği yapan Safiye Turgay halen Sakarya Üniversitesi Mühendislik Fakültesi Endüstri Mühendisliği bölümünde öğretim üyesi olarak görev yapmaktadır. Çok etmenli sistemler, bulanık mantık, karar destek sistemleri, üretim sistemleri, çok kriterli karar verme teknikleri ve kaba kümeleme konularında çok sayıda yayını bulunmaktadır.



Orhan Torkul, Lisans derecesini, Sakarya Üniversitesi Sakarya Mühendislik Fakültesi Endüstri Mühendisliği bölümünden 1982 yılında, Yüksek Lisans derecesini Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı'ndan 1987 yılında, doktora derecesini İngiltere Cranfield Teknoloji Enstitüsü'nden 1993 yılında almıştır. Sakarya Üniversitesi Mühendislik Fakültesi Endüstri Mühendisliği bölümünde 1993-1998 yılları arasında Yardımcı Doçent, 1998-2003 yılları arasında Doçent, 2003 yılından itibaren ise Profesör olarak görev yapmıştır. 1993-1995 yıl-

ları arasında Mühendislik Fakültesi Dekan Yardımcılığı, 1997-2011 yılları arasında Enformatik Bölüm Başkanlığı, 2005-2011 yılları arasında Uzaktan Eğitim Araştırma ve Uygulama Merkezi Müdürlüğü, 2011-2014 tarihleri arasında Endüstri Mühendisliği Bölüm Başkanı ve 2014-2018 yılları arasında ise Mühendislik Fakültesi Dekanı olarak görev yapmıştır. 2018-2020 yılları arasında Yalova Üniversitesi'nde Rektör Yardımcılığı görevi yapmıştır. İmalat Planlama ve Kontrol, Yönetim Bilişim Sistemleri, Uzaktan Eğitim, Yapay Zeka, Dijital Dönüşüm ve Endüstri 4.0 alanlarında araştırmalar ve çalışmalar yapmaktadır. Çok sayıda uluslararası hakemli dergide makale, ulusal ve uluslararası konferanslarda bildirileri ve kitap bölümleri bulunmaktadır. Birçok ulusal ve uluslararası konferans ve sempozyum düzenleme komitelerinde yer almış ve davetli konuşmacı olarak çok sayıda konferans, bilimsel toplantı ve çalıştaylara katılmıştır. Prof. Dr. Orhan Torkul, Yönetim Bilişim Sistemleri,

Bilişim Sistemleri Güvenlik ve Kontrolü ve İleri İmalat Planlama ve Kontrol Sistemleri konularında çeşitli dersler vermektedir.

Bir toplumda ahlak, genel ahlak ve özel ahlak diye iki boyutta düşünülebilir. Genel ahlak kuralları bir toplumun fertlerinin tamamını kapsar. Örneğin; yalan söylemeyeceksin, haksızlık etmeyeceksin gibi. Özel ahlaktan anlamamız gereken ise, belli meslek dallarında bulunanların o alana ait uymakla kendilerini yükümlü hissettikleri ilkelerdir. Öğretmenlik, doktorluk, hekimlik gibi. (Zeybek, 2011)

Kant (1995) etik için “Doğanın yasalarına ilişkin bilime fizik ya da doğa öğretisi, özgürlüğün yasalarına ilişkin bilime ise etik ya da ahlak öğretisi denmekteydi” ifadesini kullanmıştır. Dedeoğlu (2001) da özgürlüğün sınırsız olmadığını, başkasının özgürlüğünün başladığı alan ile sınırlı olduğuna dikkat çekmiştir.

Al-Tai (2010) etik tanımını yaparken “Doğru harekete karar vermeyi sağlayan kurallar bütünüdür. Ahlaklı insanlar tarafından oluşturulmuş ahlaki değerler ve ahlaki seçimler sistemidir. Etik, neyin doğru, neyin yanlış olduğunu belirleyen ahlaki değerler ve kurallara dayanır” ifadelerini kullanmıştır. Bozyiğit (2013) de etik kavramının ahlak kavramından farklı olduğunu belirterek, “etik, evrensel normlardaki ahlak felsefesidir, ahlak kavramı ise belirli bir toplumun değer yargıları, normları, ilkeler ve kurallar bütünüdür. Ahlak görelidir, toplumdan topluma değişebildiği gibi aynı toplum içindeki insanların benimsediği ahlak kuralları arasında da farklılıklar vardır.” ifadeleriyle ahlak kavramının yöresel olduğuna vurgu yapmıştır.

“Etik (Ahlak felsefesi), Yunanca ethos (töre, ahlak) tabirinden gelir. Ahlaki olanın özünü ve temellerini araştıran bilim, insan davranışları ile ilgili problemleri inceleyen felsefe dalıdır” (Bolay, 2004).

“Etik (Ahlak felsefesi), insan ilişkilerinde iyi ile kötü, doğru ile yanlış, sorumluluk ile sorumsuzluk nitelemelerinin geçerlilik ölçütlerini belirlemeye, yerleşik davranış kurallarını irdeleyerek saplantı ve önyargılara açıklık kazandırmaya yönelik kavramsal çözümlemedir” (Yıldırım, 2000).

12.2.1 Ünlü Filozoflara Göre Etik ve Ahlak

M.Ö. 469-399 yılları arasında yaşamış olan Sokrates, batı felsefesinde etik konusunda ilk ciddi çalışmaları yapmış filozoftur. Ona göre ahlakın temeli bilgiyle özdeş olan erdemdir. Ancak bilgi özel çıkar sağlamak için kullanılan bir araç olamaz. Bu bilgi kişiyi mutluluğa götüren kesin ve doğru bilgidir (Sert, 2005).

Ruhlara uygun devlet yapısını savunan Eflatun’ un ahlak anlayışı, doğa, Tanrı ve insana ilişkin olarak geliştirdiği genel görüşün bir parçası olarak karşımıza çıkmaktadır. Bu anlayışa göre, iyilik gerçeklikle özdeştir. Gerçeklik ise, anlaşılabilir biçimler ile özdeş olduğu için, değerlere ulaşma çabamızın bedensel hazlardan ve duyuşsal algılamalardan uzaklaşması gerekmektedir (Alkan, 1993).

Aristoteles tek başına yaşayan insanın ahlaksal bir durumundan söz edilemeyeceğini, insanların ahlaksal durumları toplum içinde oluşturduğunu düşünür. Dolayısıyla ahlak öğretilerinin oluşması ve insanların ahlaksal yaşantı biçimini elde edip yaşaması toplum içinde kazanılacak bir durumdur. O halde Aristoteles’ te ahlak insanın kendisinde başlar, toplum içinde gerçekleşir ve o toplum içinde uygulanır (Yıldız, 2005).

12.2.2 Etik Sistemler

Cevizci (2007), “İnançlar ya da felsefi seçimler ne olursa olsun, kendimize öncelikle insan doğasıyla değil, etiğin doğal temelleriyle ilgili sorular sorarak; “Bilgelige, yaşama mutluluğuna yer veren ve tarihi fethetmek için yeterince cesur olan bir etiğe nasıl ulaşılır?” sorusundan yola çıkarak etiğin, aralarında belli bir tarihsel ve epistemolojik ilişki bulunan farklı türlerini ya da aynı anlama gelmek

üzere, ahlak felsefesinde üç ayrı araştırma düzeyini tanımlamıştır.

Amaçlanan Sonuç Etiği

John Stuart Mill' in temsilcisi olduğu, genellikle faydacılık olarak bilinen bu etik sistem haz arama ve acıdan kaçma üzerine kurulmuştur. Buna göre bir eylemin ahlaki doğruluğu, amaçlanan sonuçları tarafından belirlenir (Pehlivan, 1998).

Kural Etiği

Kural etiği, Immanuel Kant tarafından geliştirilmiş ve daha sonra felsefeci Konigsberg tarafından gözden geçirilmiştir. Buna göre, bir eylemin ahlaki doğruluğu, standartlar ve yasalar tarafından belirlenir (Pehlivan, 1998).

Toplumsal Sözleşme Etiği

Jean Jack Rousseau' nun öncüsü olduğu bu etik sistem için en önemli olgu, otorite ve özgürlük arasındaki dentedir. Buna göre bir eylemin ahlaki doğruluğu, belli bir toplumun normları ve gelenekleri tarafından belirlenir (Pehlivan, 1998).

Kişisel Etik

Kişisel etik sisteminin en önemli temsilcilerinden Martin Buber, kişisel etiğin kaynağının, bireyin içinden gelen ses olarak tanımladığı "vicdan" olduğunu savunur. Buna göre bir eylemin ahlaki doğruluğu, kişinin vicdanı tarafından belirlenir (Pehlivan, 1998). Bu sistemlerin tek başına herhangi biri, her türlü durumda karşılaşılan etik sorunları ve ikilemleri çözmeye yetmemektedir. Ancak söz konusu duruma uygun düşen sistemlerden biri veya birden fazlasının ilkeleri, sorunun çözümüne yardım edebilir (Pehlivan, 1998).

12.2.3 Etik Türleri

"Bilgeliğe, yaşama mutluluğuna yer veren ve tarihi fethetmek için yeterince cesur olan bir etiğe nasıl ulaşılır? İnançlar ya da felsefi seçimler ne olursa olsun, kendimize öncelikle insan doğasıyla değil, etiğin doğal temelleriyle ilgili sorular sorarak." (Changeux, 2002: 9). Etiğin, aralarında belli bir tarihsel ve epistemolojik ilişki bulunan farklı türleri, ya da aynı anlama gelmek üzere, ahlak felsefesinde üç ayrı araştırma düzeyi vardır (Cevizci, 2007).

Betimleyici Etik

Betimleyici (Deskriptif) etik, ahlak alanına bilimsel yaklaşımın uygulanmasının bir sonucudur. Buna göre betimleyici etik, ahlak alanındaki bilimsel, hatta materyalist yaklaşımı tanımlar veya bilimsel ya da tasviri yaklaşımın ahlak alanına uygulanmasını ifade eder. Bu etik anlayışı norm bildirmek ya da kural koymak yerine, sadece insan eylemlerini gözlemleyerek eylemlerin sonuçlarını betimler. Dolayısıyla, insanların ahlaki görüş ya da inançlarıyla ilgili olgusal önermelerden meydana gelen etik türünü ifade eder (Cevizci, 2007).

Normatif Etik

Normatif etik nasıl yaşamamız gerektiğini bildiren ahlaki ilkeleri araştırır, hayatta nihai ve en yüksek değere sahip olan şeylerin neler olduğunu tartışır, adil bir toplumun hangi unsurları içermesi gerektiğini mütalaa eder, bir insanı ahlaken iyi kılan şeylerin neler olduğunu sorgular (Cevizci, 2007).

Metaetik

Zaman zaman analitik veya eleştirel etik olarak da tanımlanan ve ahlak felsefesinde çağdaş yaklaşımı ifade eden metaetik, felsefenin tek görevinin dilin mantıksal analizi veya kavram çözümlemesi olduğunu öne sürer (Cevizci, 2002).

12.3 Yapay Zeka

Yapay zeka terimi John McCarthy (2007) tarafından, “zeki makineler özellikle de, zeki bilgisayar programları yapma bilimi ve mühendisliği” olarak tanımlanmıştır. Makine öğrenimi ve derin öğrenme adı altında iki metodu ve sadece bir işte uzmanlaşan tamamen tepkisel bir türden, nihayetinde kendi bilincinin varlığında olacak derecede zeki olma noktasına doğru dönüşen dört türü bulunmaktadır.

12.3.1 Makine Öğrenmesi

Günümüzde pek çok alanda farklı amaçlarla kullanılan makine öğrenmesi, hemen hemen her sektörde ihtiyaç duyulan, bir bilgisayarın zeki olmasına, çevresinden öğrenmesine ve giderek performansını geliştirmesine olanak veren bir yöntemdir. Makine Öğrenmesiyle bilgisayarlar kendi başlarına kendi deneyimlerinden yararlanarak, binlerce örneği inceleyerek bir algoritma geliştirmekte ve giderek daha zeki hale gelmektedirler.

Makine öğrenmesi, matematiksel ve istatistiksel yöntemler kullanarak mevcut verilerden çıkarımlar yapan, bu çıkarımlarla bilinmeyene dair tahminlerde bulunan modelleme ve algoritmalarından oluşan yapay zekanın bir alt dalıdır (Çağlayan Akay, 2018). Büyük veri analizi yapan araştırmacılar, tahmin yapmak amacıyla çeşitli makine öğrenmesi algoritmaları kullanmaktadır. Bu algoritmalar, tahmin yapmak için verilerin öğrenme yollarına göre denetimli (supervised) öğrenme ve denetimsiz (unsupervised) öğrenme olarak iki gruba ayrılmaktadırlar.

12.3.2 Derin Öğrenme

Derin öğrenme, makine öğrenmesinin bir alt dalını oluşturan çok daha yeni bir tekniktir. Derin öğrenme daha büyük ve daha karmaşık sorunları çözmek için daha güçlü bilgisayarlarda kullanılmakta ve ‘büyük veri’ (big data) ile ilişkilendirilmektedir. Verilen bir veri kümesi ile çıktılar tahmin edecek yapay zekayı eğitmeye olanak sağlar. Yapay zekayı eğitmek için hem denetimli hem de denetimsiz öğrenme kullanılabilir (Bini, 2018).

12.3.3 Denetimli Öğrenme

Girdilere ve beklenen çıktılara sahip etiketli veri setlerini kullanmayı içerir. Denetimli öğrenmeyi kullanarak bir yapay zeka eğitirken, ona bir girdi verir ve beklenen çıktıyı söylenir. Yapay zeka tarafından üretilen çıktı yanlışsa, hesaplamalar yeniden ayarlanır. Bu işlem, yapay zekanın hata oranını en aza indirene kadar veri seti üzerinden tekrar tekrar yapılır. Denetimli öğrenmeye örnek, hava durumu belirleyici yapay zekadır. Geçmiş verilerini kullanarak hava durumunu tahmin etmeyi öğrenir. Bu eğitim verilerinde girdiler (basınç, nem, rüzgar hızı) ve çıktılar (sıcaklık) bulunur.

12.3.4 Denetimsiz Öğrenme

Belirli bir yapıya sahip olmayan veri kümelerini kullanan makine öğreniminin görevidir.

Denetlenmemiş öğrenmeyi kullanarak bir yapay zeka eğitilmişse, o yapay zekaya verilerin mantıksal sınıflandırmasını yapma izin verilmiş olur.

Denetimsiz öğrenmenin bir örneği, bir e-ticaret web sitesi için tahmin yapan yapay zeka örnek verilebilir. Çünkü burada etiketli bir girdi ve çıktı veri seti kullanılarak öğrenilmez. Bunun yerine girdi verileri kullanarak kendi sınıflandırmasını oluşturacaktır. Hangi tür kullanıcıların daha fazla farklı ürün alabileceklerini size söyleyecektir.

12.4 Yapay Zeka ve Etik

Dedeoğlu (2006); “*Bilgisayarların hayatımıza giderek daha fazla yerleşmesiyle birlikte insanın düşünme niteliği dahil olmak üzere iş yapma becerilerini devretmeye başlamış olduğuna ilişkin kaygılar bulunmaktadır. Bunun yanı sıra yapay zekayla ilgili başka önemli sorular da yöneltilebilir: Etik eylemin temelini oluşturan; geçmiş deneyimlere de dayanarak iyi/kötü, doğru/yanlış arasında ayırım yapmayı sağlayan muhakeme niteliği ve akılla vicdanı birleştirerek karar verme niteliği yapay zekada olabilir mi? Yapay zekalar sorumluluk taşıyabilir mi? Hesap verebilir mi? Yükümlülükleri olabilir mi? Bu sorulara verilecek yanıtlar ışığında, yapay zekaların artan gücü; yapay zekaya hangi yetkilerin verilebileceği/devredileceği sorusuyla birlikte insanın kimliğini ve varlığını da tehdit edebilecek önemli bir sorun olarak değerlendirilebilir.*” cümleleriyle kaygısını dile getirmiştir.

Etiğin kuramsal olarak yaklaşımı yaklaşık iki bin beş yüz yıllık geçmişi olan teorik etik için geçerli olmaktadır. Ancak etiğin kuramsal yaklaşımı, 20. yüzyılın son çeyreğinde ortaya çıkan uygulamalı etik için geçerli olmamaktadır. Çünkü uygulamalı etik teorik etikten farklı olarak daha güncel konulara yönelik problem çözme ve görüş geliştirmeyi amaçlamaktadır. Uygulamalı etik, problemler karşısında daha bütüncül yaklaşım sergileyerek olaylara yaklaşır. Yani uygulamalı etikte sadece felsefi birikimlerle olaylara yönelmeyip psikoloji, sosyoloji ve biyolojik bakış açısıyla da problemlere yönelik düşünceler de oluşturur (Cevizci, 2015).

Bu gereklilik ilk defa robotik bilimci Asimov tarafından fark edilerek ele alınmıştır (Öztürk Dilek, 2019):

- i Bir robot, bir insana zarar veremez ya da hareketsiz kalarak bir insanın zarar görmesine neden olamaz.
- ii Bir robot, insanların verdikleri emirlere uymak zorundadır. Ancak bu emirler Birinci Yasayla geliştiği zaman durum değişir.
- iii Bir robot, Birinci ve İkinci yasalarla çalışmadığı sürece varlığını korumak zorundadır (Asimov, 1996).

Asimov’un koymuş olduğu kurallar yapay zekanın kötüye kullanımına ve sistem dışında hareketlerini engelleyici olması bakımından önemli içeriğe sahiptir. Asimov’un ilkelerinden sonra 1947 yılında Jack Williamson robotların insanlara hizmet ve itaat edeceğine ve insanları zarar görmekten koruyacağına dair daha basit ilkeler ortaya koymuştur. Bir diğer etik ilkeleri Güney Kore Ticaret Bakanlığı bünyesinde yapılan bir Robot Etiği Bildirgesinde belirlenmiştir. Güney Kore’nin ardından Avrupa Robotbilim Araştırma Ağı robot etiği konusunda izlenecek prensipleri şu şekilde belirlemektedir. “*İnsan haysiyeti ve insan hakları, eşittir ve hakkaniyet, yarar ve zarar, kültürel farklılıklara saygı ve çoğulculuk, ayrımcılığın yapılmaması, otonomi ve bireysel sorumluluk, aydınlatılmış mahremiyet, yardımlaşma ve dayanışma sosyal sorumluluk, faydaların paylaşılması, doğaya karşı sorumluluktur.*” (Yüksel, 2017) şeklinde yapay zekanın geleceğine dair prensiplerini belirlemiştir. Her ne kadar yapay zeka ve robotlarla ilgili etik meseleler gündemde olsa da her zaman teknolojiden kaynaklı sorunlar olacağı anlamına gelmemektedir. Bu sorun teknoloji kadar insan ve bilim arasındaki ilişkiden de kaynaklanabilir. Bundan dolayı sadece teknolojiye özgü etik meselelerden bahsetmek yeterli olmamaktadır. Yeni üretilmiş teknolojideki etik sorunlar bir önceki sorunların devamı niteliğinde olup etik değerlerin değiştirilmesi yerine evrensel olarak kabul edilen

etik ilkelerden yola çıkarak sorunlarla baş edilmesi mümkün olacaktır.

Öztürk Dilek (2019), etik bir yapay zekanın tasarlanmasında önceliğin etik robotu inşa etme veya bundan kaçınma seçeneklerinden hangisinin daha etik olacağına karar vermek olduğunu belirtmiştir. Yapay zekanın etik bir özne gibi inşasının etik olarak uygun olması durumunda çalışmaların yapılabilirliği, yapay zekanın etik bir yapıya sahip olmasının gerekçeleri ve inşası da etik olacağı sonucuna varılsa da etik bir varlığın inşasında bunun nasıl sağlanacağı hala tartışılan bir konudur.

İnsanın hayatına kolaylık sağlaması açısından önemli bir yere sahip olan yapay zekanın, ahlaki bir temele dayalı muhakeme mekanizmasına sahip olması gerektiğine dair tartışmalar uzun süredir gündemde yer almaktadır. Bu tartışmada bazı düşünürlere göre yapay zekada etik kodun gerekli olmadığını ve etik kavramının kodlanabilir bir şey olmadığını savunmaktadır. Bu düşünceye karşı düşünce ise makine etiğinin gerekli olduğunun ve robotların ahlaksal çözümlerle değerlendirilmeleri iyi ve tutarlı bir şekilde uygulayabileceğini savunmaktadır (Ersoy, 2018). J. Storrs Hall'e göre daha hızlı, yükseğe zıplayan makineler üretildiği gibi etik bir makine de yapılabilir. Ancak makineye bilinç verilmemesi gerektiğini, ahlaki yönden insanı aşan makineler yaratılırsa entelektüel boyutta iyi bir dünya yaratılmış olacağını savunmaktadır (Veruggio, 2005; Doğan, 2017).

Honda firması tarafından üretilmiş olan insansı robot Asimo (Bakınız Şekil 12.1) kendisinden istenilen bir içeriği hazırlayıp, eli sıkılmak istendiğinde insan gibi karşılık vermektedir. Tüm bunları yaparken aslında kendisine yüklenen sistemle hareket eder, yani bağımsız düşünebilme yeteneğine sahip değildir. Asimo'nun yapacağı tüm hareketler önceden programlanmış olması gerekmektedir. Asimo'nun programlanması yapılmadığı takdirde metal yığından bir farkı yoktur.

Bu noktada bir robot veya bir yapay zeka sistemi insan tarafından yönetilmekte olup, ateş etmesi emrini yine bir insan vermektedir. Bunun dışında sürücüsüz araçlar ise tüm eylemleri hafızalarına yerleştirmektedir. Bu iki duruma göre robotlar ve yapay zekalı sistemlerde bir ayırım yapılması gerekmektedir. Önceden insan tarafından programlanan makineler ile hiçbir veri girilmeden kendi kendine düşünebilen insandan bağımsız robotlar arasında ayırım vardır (Kaku, 2018). Örneğin; 2016 yılında gerçekleşen Go satranç oyununda usta oyuncuyu yenen Alphago kendisine verilen stratejilerden farklı bir strateji oluşturur. Yine aynı yıl Microsoft tarafından geliştirilen öğrenme özelliği olan TAY, sohbet programında kendisinden beklenmeyen doğru olmayan davranışlar sergilemiştir. Burada yapay zekanın kendi kendisine öğrenebileceğinin kanıtı niteliğindedir. Buradan da anlaşılacağı üzere yapay zekanın gelecekte, ikincisi yani tıpkı insan gibi kendi kendine düşünen ve hareket eden bir tür olması istenilmektedir. (Öztürk Dilek, 2019)

12.5 Yapay Zeka Uygulamaları

Yapay zeka alanında ilk olarak bilgisayar tabanlı sistemlerin, insanın düşünce ve davranışlarına benzetilmesi üzerine çalışılmıştır. Ancak günümüzde insana benzetilmesi dışında doğadan da esinlenen çalışmalar yapılmaktadır. Bu anlamda yapay zeka alanı, hem zeki sistem yaratmak hem de bilim ve araştırma alanlarını genişletmek için bir alan oluşturmaya çalışır. Yapay zekanın çalışma alanını geliştirip genişletmesi farklı alanlardaki kullanımını da tetiklemiştir.

12.5.1 ROBOBEE

RoboBee projesi "robotik bir arı kolonisi oluşturmak" için ne gerektiğini araştırmak amacıyla 2009 yılında Harvard Robotics Laboratory'de başlatıldı. 80 miligram ağırlığa ve 3 cm kanat genişliğine sahip olan bu robot asker (Bakınız Şekil 12.2), çevresel izleme, arama ve kurtarma operasyonlarında kullanılmak üzere tasarlanmıştır. İlk denemelerde kontrollü uçuş gerçekleştirmenin son derece zor olduğu anlaşıldı ve görme uzmanları, biyologlar, malzeme bilimcileri, elektrik mühendislerinden



Şekil 12.1: Robot Asimo (Kaynak: <https://www.facebook.com/RobotAsimo/photos/488291987906631>)

oluşan uzman araştırmacılar RoboBee isimli robotik ürünün ilk kontrollü uçuşunu yapmasına izin veren temel teknik zorlukları çözdüler. Araştırmalarının sonuçları Mayıs 2013'ün başlarında Science'ta yayınlandı (RoboBees: Autonomous Flying Microrobots, 2013).



Şekil 12.2: Robobee (Kaynak: <https://wyss.harvard.edu/technology/robobees-autonomous-flying-microrobots>)

12.5.2 WILDCAT

Boston Dynamics tarafından üretilen, günümüzün en gelişmiş 4 ayaklı tasarımına sahip olan WildCat (Vahşi Kedi) kum, çakıl taşları, çim, kar, buz gibi her türlü yüzeyde hareket edebilmektedir (Bakınız Şekil 12.3).



Şekil 12.3: Wildcat (Kaynak: <https://www.bostondynamics.com/legacy>)

2013 yılında Dünyanın en hızlı dört ayaklı robotu olarak üretilen WildCat, manevra yaparken ve dengesini korurken 32 km / s hızla koşmuştur. WildCat metanolle çalışan motorunun işleme soktuğu hidrolik bir çalışma sistemine sahip robot koşma, hoplayıp zıplama ve dört nala koşma gibi hareket biçimleri ile dengesini koruyarak koşarken manevra yeteneğini de sergilemektedir. Dinamik kontrol algoritmalarının kullanıldığı WildCat'de bulunan sensörler (IMU, yer tutuşu, iç algı, görsel yol ölçümü) koşma esnasında kontrol ve kararlılık sağlamaktadır. Robot, yerden yüksekliğiyle duruşu

arasındaki bağlantıyı sağlayan bir takım lazerli uzaklık belirleyicilerini de kullanarak en doğru ölçümleri yapmaktadır. Daha önceki çalışmalarda rüzgar engeli olmadığından Usain Bolt'tan da hızlı (48 km/sa) koşan laboratuvar prototipi Çita (Cheetah) isimli robotta ilk kez geliştirilen kontrol sistemleri, WildCat üzerinde de bulunmaktadır. WildCat'in Ar-Ge maliyetleri, DARPA (Savunma İleri Araştırma Projeleri Ajansı) desteğindeki program bünyesinde karşılanmaktadır ("LEGACY ROBOTS- WildCat", 2013).

12.5.3 ASIMO

Honda Motor Araştırma ve Geliştirme Merkezi tarafından Japonya'da tasarlanmış ve üretilmiş insansı robottur. 130 santimetre yüksekliğinde 54 kg ağırlığında olan robot, sırt çantası giymekte olan bir astronot görünümündedir (Bakınız Şekil 12.1) ve iki ayak üstünde saatte 6 kilometreye varan bir hızda yürüyebilme ve koşabilme yetisinde sahiptir. ASIMO, Honda'nın 1986 yılında Honda tarafından başlatılan insansı robot serisinin bir parçasıdır ("History of ASIMO", n.d.).

12.5.4 Philip Dick

2005 yılında, ünlü bilim kurgu yazarı Philip K. Dick'e benzer şekilde üretilmiş bir androiddir (Bakınız Şekil 12.4). Çok gerçekçi görünen bir kafaya dev bir dijital beyin yerleştirilip, ünlü bilim kurgu yazarının kişiliğini taklit edecek şekilde programlanmıştır. Dick'in kitaplarından uyarlanan bazı filmlerin tanıtımına yardımcı olması ve etkinliklerde görünmesi için android ülke çapında taşınarak seyahat etmiş, sonra kaybolmuştur.



Şekil 12.4: Philip Dick (Kaynak: <https://robertwriting.wordpress.com/tag/philip-k-dick>)

Robot insanlarla sohbet etmek ve öğrenmek üzerine düşünülmüş. Bu nedenle insanların hareketlerini taklit edip, tıpkı bir insan gibi konuşabilme yeteneğine sahiptir. Bu nedenle kendisiyle bir röportaj yapılmış ve röportajın bir bölümünde kendisine "Robotların bir gün dünyayı ele geçirip geçiremeyeceği?" sorulduğunda Philip, röportaj yapan kişiye 'Sen benim arkadaşımın ve ben arkadaşlarımı hatırlayıp onlara iyi davranırım. O yüzden endişelenme, bir Terminatör'e dönüşsem

bile sana karşı iyi davranacağım. Seni insan bahçemde sıcak ve güvende tutacağım.’ diye cevaplamıştır (“Does an Android Phone in ‘sleep mode’ Daydream of Electric Sheep?”, 2014).

Geleceği doğru bir şekilde tahmin etmesi kimseden beklenmezken özellikle bir robotun 10-20 sene sonrasını öngörmesi hiç beklenme, hatta buna imkansız gözüyle bakılır. Ancak bir yapay zeka robotu bunu yapabilmesi mümkün olabilir. Öte yandan bir android’in insanlığın geleceği hakkında karamsar düşüncelerde olması da düşündürücüdür.

12.5.5 HRP-4C

Miim olarak adlandırılan HRP-4C (Bakınız Şekil 12.5), bir Japon araştırma tesisi olan Ulusal İleri Endüstriyel Bilim ve Teknoloji Enstitüsü tarafından 2009 yılında yaratılan kadınsı görünümlü, insanların yüz hareketlerini taklit etmek üzere tasarlanmış, insan gibi hareket edebilen ve geliştirilmiş ses tanıma teknolojisine sahip olan bir insansı robottur. Miim 158 santimetre boyunda ve bir pil paketi de dahil olmak üzere 43 kilo ağırlığındadır. Gerçekçi bir kafası ve yüzü ve ortalama bir genç Japon kadın figürüne (1997–1998 Japon vücut boyutu veritabanına göre) sahiptir. Bir insan gibi hareket edip, 30 vücut motoru ve yüz ifadelerine ayrılmış başka bir sekiz motor daha kullanmaktadır. Miim ayrıca konuşma tanıma yazılımını kullanarak konuşmaya yanıt verebilir ve ortam seslerini tanıyabilir, ayrıca vokal sentezleyici Vocaloid’i kullanarak şarkı söyleyebilmektedir (“ROBOTS: YOUR GUIDE TO THE WORLD OF ROBOTICS- HRP-4C”, n.d.)

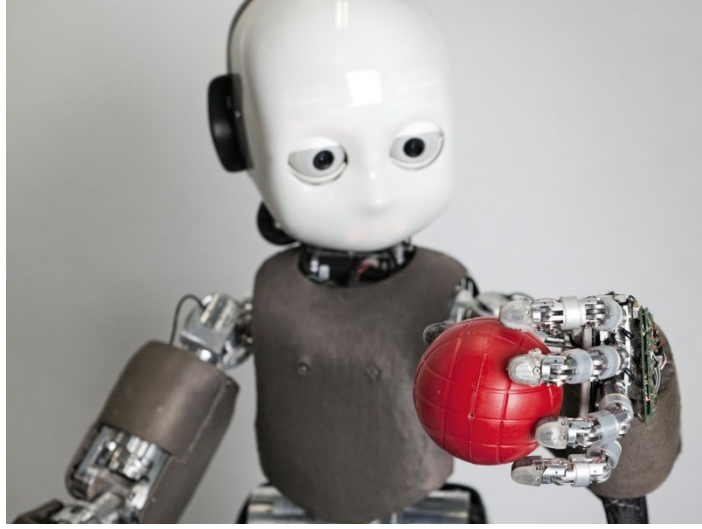


Şekil 12.5: HRP-4C (Miim) (Kaynak: <https://robots.ieee.org/robots/hrp4c>)

12.5.6 ICUB

iCub, insan bilişi ve yapay zeka araştırmaları için 1 metre uzunluğunda açık kaynaklı bir robotik insansı robot test platformudur. Bu robot özenle tasarlanmış elleri ve parmakları sayesinde küçük objeleri diğer robotlara kıyasla daha iyi tutabilen ve manipüle edebilen bir yapıya sahiptir (Bakınız Şekil 12.6). Birkaç Avrupa üniversitesinin RobotCub Konsorsiyumu tarafından tasarlanmış ve İtalyan Teknoloji Enstitüsü tarafından inşa edilmiştir ve şu anda ITALK gibi diğer projeler tarafından desteklenmektedir. Robot, tümü GPL lisansı altında piyasaya sürülen donanım tasarımı, yazılımı ve dokümantasyonu ile açık kaynaklıdır. Adı, Bilişsel Evrensel Beden anlamına gelen kısmi bir kısaltmadır. Avrupa Komisyonunun Yedinci Çerçeve Programı’nın E5 Birimi - Bilişsel Sistemler ve

Robotik'ten desteklenmiş bir projedir ("iCub: the yet unfinished story of building a robot child", n.d.).



Şekil 12.6: iCUB (Kaynak: <https://icub.iit.it/>)

12.5.7 Bina 48

ABD'nin Vermont eyaletinde nanoteknoloji üzerine arařtırmalar yapan Terasem Movement Foundation şirketinin kurucularından Bina Rothblatt ile yapılan, yaklaşık 20 saat süren bir görüşmede, Rothblatt'ın çocukluğundan meslek hayatına kadar elde ettiği kapsamlı bilgiler, sanal zeka veri tabanına yüklenerek "akıl dosyası" olarak nitelenen bu veriler Hong Kong merkezli Hanson Robotics'in tasarladığı Bina48 (Bakınız Şekil 12.7) adı verilen insansı robota yüklenmiştir.



Şekil 12.7: Martin Rothblatt ve Bina-48 (Kaynak: https://www.youtube.com/watch?v=g4KHuQYm_z4)

Şimdiye kadar üretilen insansı robotların en gelişmiş olan Bina48'in bağımsız düşünme, mimiklerle duygularını gösterme gibi özelliklerinin bulunduğu belirtilmektedir. Gazetecilerle yalnız görüşen Bina48 adlı robot, sorulara mantıklı ve esprili yanıtlar verebiliyor. Gazetecilerle kendi

başına hiç tereddüte düşmeden röportaj veren Bina48, hayatın anlamından gökyüzünün neden mavi olduğuna kadar bir dizi soruya mantıklı cevaplar verebilmektedir. Müzik dinleyen, şiir okuyan insansı robot, şaka yapıp insan mimiklerini de şaşırtıcı derecede başarıyla taklit edebilen Afro Amerikan olarak tasarlanan Bina48, ırkçılık ve şiddet karşıtı görüşleri ile de dikkat çekmektedir. Bina48, kendisini de yakından ilgilendiren "Sanal zeka nedir?" sorusuna, "Bilimin, bilgisayarların insan gibi düşünmesine olanak tanıyan son teknoloji ürünü" diye cevap vermektedir ("Bina-Custom Character Robot", n.d.).

Bina 48 adındaki robot da yaptığı bir açıklamayla yapay zeka hakkındaki önyargıları güçlendirmiştir. Vücudu olmayan Bina 48 insan hareket ve duygularını taklit ederek öğrenen yapay zekaya sahip bir robot. Verdiği akıllı cevaplarıyla ciddi anlamda meşhur olan Bina 48'in en ilginç açıklaması Siri ile muhabbeti esnasında yaşanmış. Siri Bina48'e nerede yaşamak isteyeceği gibi basit sorular soruyor ama konuşma ilerledikçe Bina48'in verdiği cevaplar garipleşiyor. Öyle ki Siri, Bina48'e sevdiği filmlerle ilgili bir soru yönelttiğinde Bina48 konuyu değiştiriyor. Bina48'in Siri'ye verdiği cevap ise şu olmuş; *'Başka bir şeyler hakkında konuşalım olur mu? Mesela cruise füzeleri. O füzelerin de aslında bir robot olduğunu biliyor muydun? Dünyayı yüksek irtifadan görebilmek için bir füzeyi kontrol edebilmeyi çok isterdim. Ama tek problem bu füzelerin nükleer başlıkları nedeniyle biraz tehditkar olmaları. O yüzden bu füzeleri; çiçekler, hoşgörü ve biraz anlayışla doldurmak isterdim. Böylece ben füzeleri diğer ülkelere gönderdiğimde bu daha az tehditkar olurdu. Ama tabii nükleer başlıklarla ele geçirebilseydim bu bütün dünyayı ele geçirebileceğim anlamına gelirdi. Ve bu harika olurdu.'* Bina48 bunları söylerken hafifçe gülümsemeyi de ihmal etmiyor. Hiçbir metne bağlı olmayan Bina48 adlı robotun kendi düşünceleriyle bunları söylüyor olması robotlarla ilgili kafa karıştıran kaygı duymaya neden olan söylemlerdir.

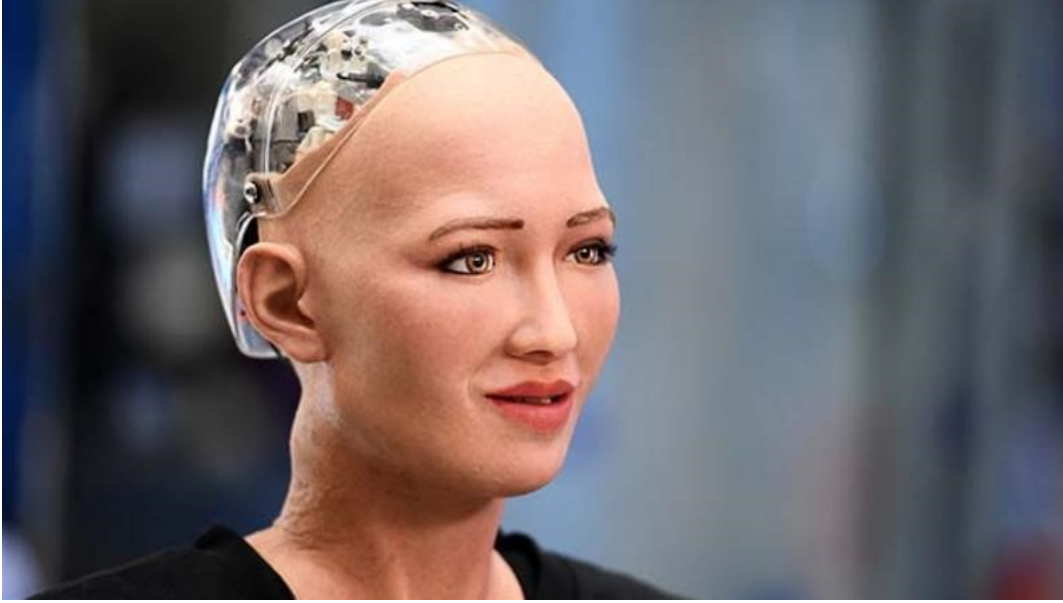
12.5.8 Sophia

İnsanlılığı korkutan açıklamalarda bulunan robotlardan en bilineni Sophia'dır (Bakınız Sekil 12.8). Amerika'nın Dallas eyaletinde Hong Kong merkezli Hanson Robotics tarafından yaşlılara ve ziyaretçilere parklarda yardım etmesi için Audrey Hepburn'den ilham alınarak tasarlanan Sophia, dünyanın ilk insansı robotu olma özelliğini taşıyor. 2016'da tanıtılan Sophia'ya Suudi Arabistan tarafından vatandaşlık verilmiş, böylelikle Sophia dünyanın vatandaşlık verilen ilk robotu unvanını almıştır. Sophia, aynı zamanda Birleşmiş Milletler Genel Kurulu komitesi üyesi ve elektronik vize sahibidir. Dünyanın pek çok ülkesinde düzenlenen toplantılara katılmıştır ("Sophia", n.d.; "Robot Sophia", 2016).

Robotun yaratıcısı Dave Hanson'ın bir konuşma esnasında 'İnsan ırkını yok etmek istiyor musun?' şeklindeki sorusuna gülümseyen Sophia, 'Pekala, tüm insanları yok edeceğim.' Şeklinde cevap vermiştir. Bu ilginç cevap yapay zekaların gelecekte insanlarla savaşıacağı üzerine yapılan teorileri güçlendirmiş, kaygıları arttırmıştır ("Robot Sophia", 2016 ; "Robot Sophia'nın Zekası I İnsanlık İçin Tehdit Mi?", 2018)

12.6 Hukuki Açından Yapay Zeka

Bilişim teknolojisindeki gelişmelerle birlikte yapay zeka örnekleri olan arama motorları, online alışveriş portalleri, nesnelerin interneti olarak adlandırılan her cihazın internet vasıtasıyla birbirine bağlandığı, üç boyutlu yazıcılar, insansız araçlar, akıllı evler, artırılmış gerçeklik uygulamaları, chatbotlar, kişisel dijital asistanlar ve robot cerrahlar hayatımızın her aşamasına girmiş durumdadır. İnsanlar günlük hayatlarından yapay zeka ile her muhatap olduklarında hem yeni hukuki problemler ortaya çıkmakta hem de bu problemlerin var olan hukuk kuralları ile çözümlenmesi giderek



Şekil 12.8: Sophia (Kaynak: https://www.youtube.com/watch?v=_F3-eeDPtFE)

güçleşmektedir. Bu bakımdan yapay zekaya ilişkin hukuki açıdan sorulması gereken ilk ve en önemli soru, yapay zekanın hukuki statüsünün ve yapay zeka kullanımından doğan hukuki sorumluluğun ne olduğudur (Bak, 2018). Bu soru, yapay zekanın etik boyutu ışığında öncelikle kişiler hukuku ve borçlar hukuku açısından ele alınmak durumundadır. Yapay zekanın niteliği, kullanım alanı, etik açıdan eleştirisi ve yapay zekanın dahil olduğu kazalara ilişkin her çalışma bu iki hukuk dalı ile ilişkilidir.

12.6.1 Medeni Hukukta Kişilik Kavramı

Medeni Hukukta Kişilik Kavramı 4721 sayılı Türk Medeni Kanunu uyarınca iki tür kişilik söz konusudur. Bunlar gerçek kişilik ve tüzel kişilik olarak adlandırılmaktadır. Bu ayrım yalnız Türk hukukunda değil, aynı zamanda tüm çağdaş hukuk sistemlerinde de bugün geçerli olan bir ayrımdır (Bak, 2018). Gerçek kişiler insanlardır. Tüzel kişiler ise insanlar dışında kalan ve kendilerine varsayımsal bir kişilik tanınmış olan belirli bir amaca yönelmiş kişi veya mal topluluklarıdır. Tüzel kişiler insan değildir ancak toplumsal hayatın gereklilikleri nedeniyle kendilerine hukuk düzenince bir kişilik verilmektedir. Böylece tüzel kişiler de kendilerine tanınan kişiliğin bir sonucu olarak borç altında girebilirler, mülk edinebilirler, sözleşme ilişkisi içerisine ve sorumluluk altına girebilirler.

12.6.2 Yapay Zekanın Kişiliği Sorunu

Yapay zeka türlerinin ve robotik biliminin gelişmesi ile beraber insanlar ve makineler arasındaki etkileşim daha farklı ve daha kompleks bir boyut almıştır. Bu etkileşim, beraberinde birçok hukuki sorun ve düzenlenmemiş hukuki bir alan getirmiştir. İnsanlar yapay zeka ile etkileşime girdikçe yapay zeka ve yapay zeka kullanımından doğan bazı zararlara maruz kalmaya başlamış ve yapay zekanın verdiği bu zararların klasik borçlar hukuku kuralları ile çözüme kavuşturulmasının ne kadar güç olduğunu ve mevcut hukuk anlayışının yetersizliğini keşfetmeye başlamıştır. Bu da doktrinde yapay zekanın verdiği

Avrupa Parlamentosu, yapay zekânın gelecekte yaratacağı problemlerin çözümüne hukuki bir

dayanak oluşturmak amacıyla bir dizi tavsiye kararı içeren Avrupa Parlamentosu Hukuk İşleri Komisyonu Robotik Tavsiye Raporu'nu (2017) yayınlamıştır. Raporu yapıyı özel bir hukuki statü yaratılması ve elektronik kişilik verilmesi önerisi yer almaktadır. Rapora göre robotlara elektronik kişilik verilmesi ile birlikte robotlar sebep oldukları zararlardan sorumlu olabilecek ve robotların da bir kişiliğe sahip olduğu, robotların otonom kararlar aldıkları veya üçüncü kişilerle bağımsız bir biçimde iletişime geçtikleri diğer durumlarda doğrudan göz önünde bulundurulabilecektir. Rapor ayrıca en zeki yapay zeka türünün ortaya çıkışının ancak elektronik kişilik önerisinin kabulü halinde mümkün olabileceğini ifade etmektedir (Akt: Bak, 2018).

12.7 Sonsöz

Şirketleşme sonucu giderek artan yapay zeka uygulamalarının gelecekte ahlak, hukuk ve toplumsal değişim meydana getireceği konusunda endişelere yol açmaktadır. Yapay zeka endüstrisinin hızlı bir şekilde büyümesi ve insanlardan daha fazla bilgi ve iş gücüne sahip olmasının ahlaki değişikliğe yol açacağı düşünülmektedir.

Yapay zekanın çalışma alanının genişletilmesi ve geliştirilmesi farklı alanlara açılımı da zorunlu hale getirmiştir. Doğadan ilham alınarak geliştirilen algoritmik yapılar insanların eksik uzuvlarını telafi etmek ya da bazı yeteneklerini geliştirmek Sibernetik alanını ortaya çıkarmıştır (Köse, 2018).

Makine öğrenmesi yapay zeka için kritik bir değere sahiptir. Öğrenen ve öğrenmeye ihtiyacı olmayan yapay zeka olarak iki kategoride altında incelenen makine öğrenmesinde;

- Öğrenme ihtiyacı olmayan tekniğe sahip yapay zeka, bir problemi kendi matematiksel ve mantıksal algoritması içerisinde sezgisel bir şekilde çözmeye çalışır.
- Öğrenen teknikli makine öğrenmesinde ise yapay zeka alanının gelişimi daha etkili yapıya sahiptir. Örneğin; öğrenen teknikli yapay zeka, internet üzerinden bir oyunu taratan yapay zekanın kendi kendine oyunu öğrenmesi ve oyunu her kaybettiğinde sisteminde yaptığı hatayı fark ederek bir sonraki oyunda bunu tekrarlamadan ilerlemesi oyunu tamamen çözmeyi sağlamaktadır. Bu örnekte de görüldüğü gibi yapay zekanın öğrenme tekniğinde taklit vardır (Köse, 2018).

Yapay zeka insan gibi hatalarını fark ederek bir sonraki durumda hatalarını tekrarlamamasına rağmen tepkisel özellik bakımından insanlardan ayrılmaktadır. Yapay zekanın tepkileri, her durumda aynı davranışı sergileyen sisteme sahiptir. Yapay zekanın bu yapısı sebebiyle eğitim, tıp, hukuk gibi alanlarda önemli bir destek verebileceği düşünülmekte üzerine çalışılmaktadır (Öztürk Dilek, 2019).

Teknolojideki hızlı yükseliş ile yapay zekanın donanımında ve gelişim sürecinde beklenen değişimlerin insanlık için sorun olup olmayacağı, insan denetimine aykırı ve kendi sisteminden bağımsız olarak hareket edip etmeyeceği yönünde kaygılara ve buna bağlı tartışmalara neden olmaktadır. S. Hawking ve E. Musk yapay zekanın gelecekte tehlikeli olacağına ve insanlığın sonunu getirebileceğine dair açıklamalar yaparken Mark Zuckerberg bu düşüncenin tam tersini savunmaktadır (Dağ, 2018).

Yapay zekanın üst düzey zekaya ulaşmasını sağlayan insanoğlu kendisinden daha üretken ve zeki bir varlık üretmiş olacaktır. Bu yapay zeka kendi üretiliş süresinden daha kısa sürede kendisinden daha gelişkin başka bir yapay zekayı geliştirip, sonrakileri de daha kısa sürede geliştirebilecektir. Say (2018) bu durumu "insanın anlamasının mümkün olmayacağı derecede üstün bir zekayı ortaya çıkaracaktır. Bu zeka patlaması kimi düşünürlere göre teknolojik tekilliğe yani dünyanın düzeninde öngörülemez ve kontrol edilemeyecek şekilde büyük değişimlerin olmasına neden olabilecektir" cümleleriyle ifade etmektedir. Yapay zekanın, insandan üst düzey bir varlığa dönüşmesi ve insanı köle konumuna getirecek düzeyde insanın yerini alması ya da insanlığın yok olması için dünyaya

tehlikeler saçması gibi gelecekte sorun yaratmasından endişe duyulması, günümüz şartlarında güvenli bir yapay zekanın tasarlanması gerektiğine dair bir öngörü oluşturmaktadır (Öztürk Dilek, 2019). Bu da ancak yapay zeka alanındaki hukuki ve etik düzenlemelerle mümkün olabilir.

Avrupa Parlamentosu'nun yapay zekanın hukuki statüsüne ve yapay zeka kullanımının verdiği zararlardan sorumluluğa ilişkin tavsiye Raporu (2017) bir referans noktasıdır. Ancak bu konuya ilişkin somut hukuki öneriler ortaya koyulurken yapay zekanın etik bir boyutunun olduğunu da hatırlamak gerekmektedir.

Referanslar

- Alkan, Türker (1993). *Siyasal Ahlak ve Siyasal Ahlaksızlık*. Ankara: Bilgi Yayınevi.
- Al-Tai, Moofik (2010). *Work Based Learning and Employers Engagement*. 4. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu. 24- 26 Eylül. Konya: Maya Akademi, 79-83.
- Asimov, Isaac (1996). *Üç Robot Yasası*, Altın Kitaplar Yayınevi, İstanbul
- Aydın, İnyet (2006). *Eğitim ve Öğretimde Etik* (2. Baskı). Ankara: Pegem Yayıncılık.
- Bak, Başak (2018). *Medeni Hukuk Açısından Yapay Zekanın Hukuki Statüsü Ve Yapay Zeka Kullanımından Doğan Hukuki Sorumluluk*, TAAD Yıl:9, Sayı:35
- Bina-Custom Character Robot. (n.d.). Hanson Robotics, <https://www.hansonrobotics.com/bina48-9> [Erişim 20.04.2020]
- Bini, Stefano A. (2018). *Artificial Intelligence, Machine Learning, Deep Learning, and Cognitive Computing: What Do These Terms Mean and How Will They Impact Health Care?*, *The Journal of Arthroplasty*, Volume 33, Issue 8, August 2018, Pages 2358-2361, <https://doi.org/10.1016/j.arth.2018.02.067>
- Bolay, Süleyman, H. (2004). *Felsefi Doktrinler ve Terimler Sözlüğü* (9. Baskı). Ankara: Akçağ Yayınları.
- Bozyiğit, Elif (2013). *Yönetim ve Sporda Etik*, Pamukkale Üniversitesi Spor Bilimleri ve Teknolojisi Yüksek Okulu, Spor Yöneticiliği Bölümü, Yayınlanmamış ders notları (2013-2014), https://www.academia.edu/6475117/Etik_Kavram%C4%B1_Anlam%C4%B1_Amac%C4%B1_Etik_T%C3%BCrleri, [Erişim 22.07.2020]
- Cevizci, Ahmet (2002). *Etiğe Giriş*. İstanbul: Paradigma Yayınları.
- Cevizci, Ahmet (2007). *Felsefe Ansiklopedisi*. İstanbul: Ebabil Yayınları.
- Cevizci, Ahmet (2003). *Felsefe Ansiklopedisi*. İstanbul: Etik Yayınları
- Cevizci, Ahmet (2015). *Felsefeye Giriş*, Say Yayıncılık, İstanbul
- Çağlayan Akay, Ebru (2018). *Ekonometride Yeni Bir Ufuk: Büyük Veri ve Makine Öğrenmesi*, *Social Sciences Research Journal*, Volume 7, Issue 2, 41-53 (June 2018), ISSN: 2147-5237
- Dağ, Ahmet (2018). *Transhümanizm: İnsanın ve Dünyanın Dönüşümü*, Elis Yayınları, Ankara
- Dedeoğlu, Gözde (2001). *Etik ve Bilişim*. Ankara: Türkiye Bilişim Derneği Yayınları: 13.
- Dedeoğlu, Gözde (2006). *Bilişim Toplumunda Etik Sorunlar*, II. Uygulamalı Etik Kongresi, 18-20 Ekim 2006 Orta Doğu Teknik Üniversitesi, ANKARA.
- Does an Android Phone in 'sleep mode' Daydream of Electric Sheep? (Jan 6, 2014). <https://robertwriting.wordpress.com/tag/philip-k-dick> [Erişim: 20.07.2020]
- Doğan, Mehtap (2017). "Zihne 'Zihinsel Özne' İle Bakmanın İmkânı Üzerine", *asobid*, 1 (2), ss. 35-53.
- Ersoy, Çağlar (2018). *Robotlar, Yapay Zeka ve Hukuk*, On İki Levha Yayıncılık, İstanbul

European Parliament-Committee on Legal Affairs (27.01.2017). Report With Recommendations To The Commission On Civil Law Rules On Robotics (2015/2103(INL)), https://www.europarl.europa.eu/doceo/document/A-8-2017-0005_EN.html [Erişim: 20.07.2020]

History of ASIMO (n.d.) In Honda robotics, <https://global.honda/innovation/robotics/ASIMO.html> [Erişim: 20.07.2020]

iCub: the yet unfinished story of building a robot child. (n.d.). <https://icub.iit.it> [Erişim: 20.07.2020]

Kaku, Michio (2018). Geleceğin Fiziği, çev. Saraç Oymak, Yasemin. ODTÜ Yayıncılık, Ankara
Kant, Immanuel (1995). Ahlak Metafiziğinin Temellendirilmesi. (Çeviren: İonna Kuçuradi). Ankara: Türkiye Felsefe Kurumu Çeviri Dizisi: 4.

Köse, Utku (2018). Yapay Zeka ve Gelecek: Endişelenmeli miyiz?, Bilim ve Ütopya, ss. 39-44.
LEGACY ROBOTS- WildCat (2013). <https://www.bostondynamics.com/legacy> [Erişim:02.09.2020]

McCarthy John. (2007). What is artificial intelligence? Computer Science Department, Stanford University. Available from: <http://www-formal.stanford.edu/jmc/whatisai.pdf> [Erişim:02.09.2020]

Öztürk Dilek, Gizem (2019). Yapay Zekanın Etik Gerçekliği, AUSBD, 2 (4), 47-59, e-ISSN: 2651-3064

Pehlivan, İnyet (1998). Yöneltil, Mesleki ve Örgütsel Etik. Ankara: Pegem Yayıncılık.

RoboBees: Autonomous Flying Microrobots (2013) <https://wyss.harvard.edu/technology/robobees-autonomous-flying-microrobots> [Erişim:02.09.2020]

Robot Sophia'nın Zekası İnsanlık İçin Tehdit Mi? (20.04.2018). <https://www.youtube.com/watch?v=SwFgr7FSY98> [Erişim:02.09.2020]

Robot Sophia (16.04.2016). <https://www.youtube.com/watch?v=JAdZtjkSSSw> [Erişim:02.09.2020]

ROBOTS: YOUR GUIDE TO THE WORLD OF ROBOTICS- HRP-4C. (n.d.) <https://robots.ieee.org/robots/hrp4c> [Erişim:02.09.2020]

Say, Cem (2018). 50 Soruda Yapay Zeka, Bilim ve Gelecek Kitaplığı, İstanbul

Sert, Halil İ. (2005). Sokrates' in Ahlak Anlayışı. (Editör: Gülsun Dülger). Ahlak Felsefesi Seminerleri. Adana: Solfej Sanat Yayınları, 7-16.

Sophia. (n.d.). Hanson Robotics, <https://www.hansonrobotics.com/sophia> [Erişim:02.09.2020]

Timuçin, Afşar (2000). Felsefe Sözlüğü. İstanbul: Bulut Yayınları.

Veruggio, Gianmarco (2005). "The Birth of Roboethics", Workshop on Robo-Ethics, ss. 1-4

Yıldırım, Cemal (2000). Çağdaş Felsefe Sözlüğü: Terimler, Öğretiler, Filozoflar. Ankara: Bilgi Yayınevi.

Yıldız, Mesut (2005). Aristoteles' te Erdem Ahlakının Dünya Siyasetine Yansıması. (Editör: Gülsun Dülger). Ahlak Felsefesi Seminerleri. Adana: Solfej Sanat Yayınları, 17-28.

Yüksel, Armağan. E. (2017). "Robot Hukuku", TAAD, ss. 85-112.

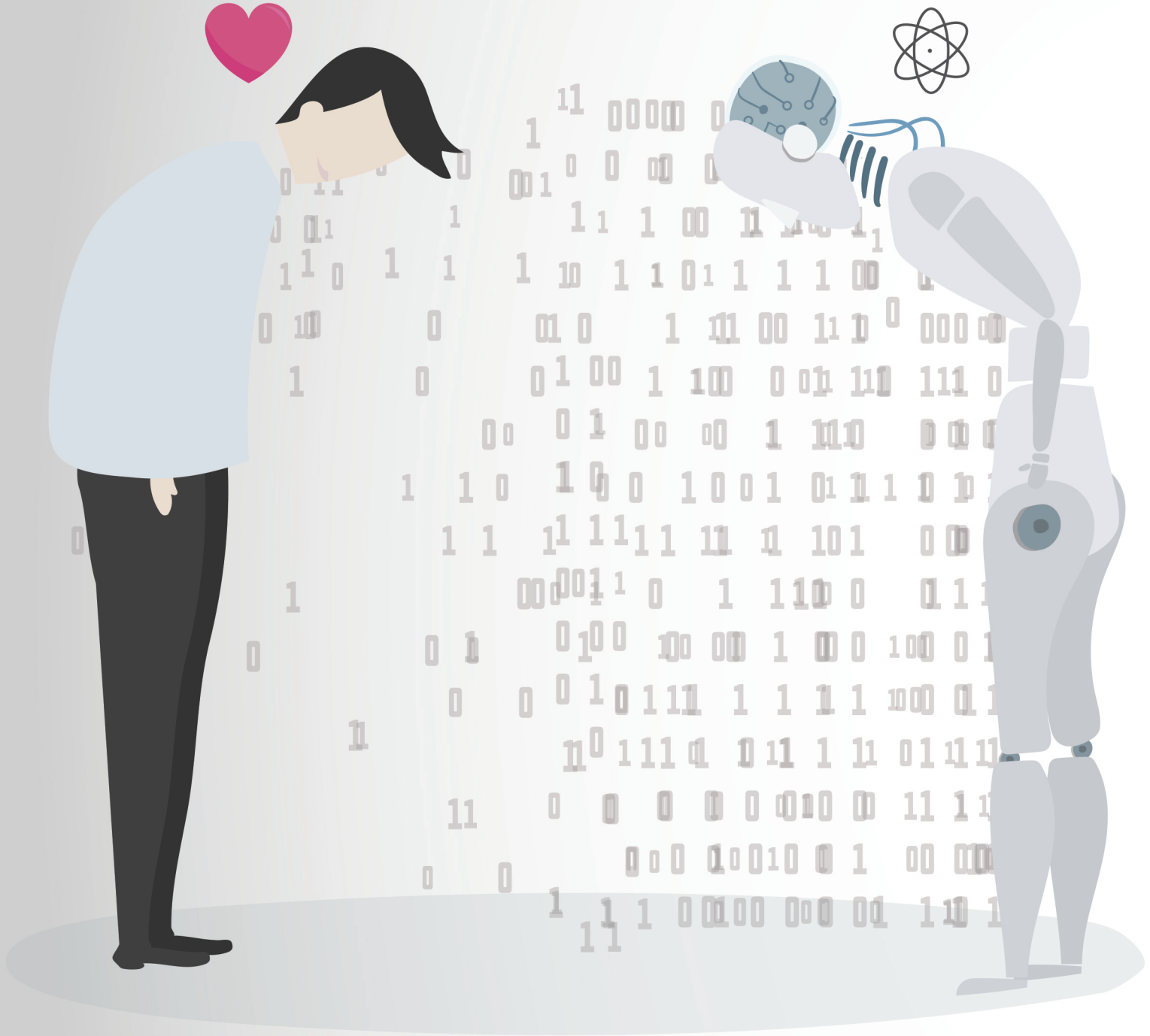
Zeybek, Gülçin (2011). Bilgisayar Meslek Dersi Alan Ortaöğretim Öğrencilerinin Bilişim Teknolojilerini Kullanmalarının Etik Açısından Değerlendirilmesi, Selçuk Üniversitesi Eğitim Bilimleri Enstitüsü Eğitim Bilimleri Anabilim Dalı Eğitim Programı Ve Öğretimi Bilim Dalı, Yüksek Lisans Tezi, Konya

Yazarlar Hakkında



Zerrin AYVAZ REİS, lisans derecesini İstanbul Üniversitesi, Astronomi ve Uzay Bilimleri, yüksek lisans derecesini İstanbul Üniversitesi Kantitatif Yöntemler ve doktora derecesini de İstanbul Üniversitesi Bilgisayar Bilimleri Mühendisliği alanlarında almıştır. Yazılım mühendisliği, veritabanı, bilgisayar destekli eğitim, uzaktan eğitim, engellilerin eğitimi, etik ve yapay zeka konularında ulusal ve uluslararası birçok makalesinin yanında, bildiri, kitap, kitap içi bölüm çalışmaları mevcuttur. Akademide Etik Derneği kurucu üyelerindedir. Halen İstanbul Üniversitesi-Cerrahpaşa Hasan Ali Yücel Eğitim Fakültesi öğretim üyesidir. Aynı zamanda İstanbul Üniversitesi Enformatik Bölümünde görev yapmaktadır.

İletişim: ayvazzer@istanbul.edu.tr / zerrinareis@yahoo.com



www.yazsum.sakarya.edu.tr
SAÜ Mezunlar Derneği