

T.C. GALATASARAY ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

FUTBOLCU MARKET DEĞERİ TAHMİNLEME

(PREDICTING MARKET VALUE OF FOOTBALL PLAYER)

DÖNEM PROJESİ

Oğuzhan ZENGİN

Anabilim Dalı : MATEMATİK

Program : VERİ BİLİMİ

Danışmanı : Dr. GÜNCE KEZİBAN ORMAN

ARALIK 2022

ÖNSÖZ

Bu çalışma futbolcuların market değerlerini tahminleme üzerine gerçekleştirilmiştir. Veri seti bir futbol oyunundan alınmıştır, 10000 futbolcunun verisi üzerinden çalışma gerçekleştirilmiştir. Bireysel veriler makine öğrenmesi yardımıyla, python programında tahminlenmiştir.

Proje süresince yardımları ve desteğiyle, projenin tamamlanmasına yardımcı olan sayın hocam Günce Keziban ORMAN'a teşekkürlerimi sunarım.

Oğuzhan ZENGİN

12/2022

İÇİNDEKİLER

ÖNSÖZ.....	i
ŞEKİLLER LİSTESİ.....	iii
ÖZET.....	iv
ABSTRACT.....	v
1-GİRİŞ.....	1
2-UYGULAMA.....	2
2.1-VERİ ÇEKME.....	2
2.2-VERİ TEMİZLEME.....	4
2.3-VERİ ÖN İŞLEME.....	6
2.4-VERİ ANALİZİ.....	8
3-TAHMİNLEME.....	16
4-SONUÇ.....	23
KAYNAKÇA.....	24

ŞEKİLLER LİSTESİ

- Fig.1: SOFIFA web arayüzü
- Fig.2: Kütüphaneler
- Fig.3: URL ile veri çekme kodları
- Fig.4: Çekilen verinin ilk hali
- Fig.5: Veri temizleme kodları 1
- Fig.6: Veri temizleme kodları 2
- Fig.7: Sütun isimlendirme
- Fig.8: Verinin temizlenmiş hali
- Fig.9: Veri setini ön işleme
- Fig.10: Veri tiplerini değiştirme
- Fig.11: Veri setinin sadece sayısal değer barındıran hali
- Fig.12: Korelasyon matrisi
- Fig.13: Korelasyon ısı haritası
- Fig.14: Korelasyon ısı haritası 2
- Fig.15: OLS regresyon çıktısı
- Fig.16: Market deüeri ve potansiyel ilişkisi
- Fig.17: Tüm verinin istatistiksel özeti
- Fig.18: Mevkilerin pasta dilimi dağılımı
- Fig.19: İstatistiksel inceleme
- Fig.20: En yüksek oyun gücüne sahip 10 oyuncu
- Fig.21: En yüksek potansiyele sahip 10 oyuncu
- Fig.22: En yüksek market değerine sahip 10 oyuncu
- Fig.23: Gradyan ağacı güçlendirme
- Fig.24: Tahminleme modeli 1
- Fig.25: Tahminleme modeli 2
- Fig.26: Tahmin, gerçek değerler ve aralarındaki fark
- Fig.27: Tahmin ve gerçek değerlerin histogramı
- Fig.28: Tahmin ve gerçek değerler (kırmızı: gerçek, mavi: tahmin)
- Fig.29: İstatistiksel tahmin sonuçları

ÖZET

Futbolcuların market değerleri, diğer bir deyişle bonservislerinin belirlenmesi uzun bir süredir tartışılan bir konudur. Gerçekleşen transfer ücretleriyle, öngörülen ücretler arasında zaman zaman büyük farklar oluşabiliyor. Bu ücretin belirlenmesinde mevcut takımıyla kalan sözleşme süresi, yaşı, sakatlık geçmişi gibi birçok parametre vardır. Bu çalışmada verilerin bir futbol oyunundan alınmasının sebebi, oyunun verilerinin sürekli güncellenmesi ve gerçek hayatı belirlemeyen bireysel verilerin bu oyunda bulunmasıdır. Futbolcuların market değerini etkileyen her parametrenin istatistiksel olarak ele alınmadığını göstermiş olacağız. İstatistiklerle belirlenen verileri kullanmamamızın sebebi ise istatistiklerin faklı liglerdeki mücadele düzeyini, çok gerçekçi olmayan bir yöntemle ayırmasıdır. Neden bu oyunun verilerini kullandığımız sorusuna gelirse, yapılan çalışmalarda yapay zekâ ile maç sonucu tahmin etme ve turnuva şampiyonu tahminlemelerinde çok başarılı olmasıdır. Sırasıyla 2010, 2014 ve 2018 dünya kupası şampiyonları bu verilerle doğru tahmin edilmiştir. Veri setinin bir diğer önemli özelliği sürekli güncellenmesi, mevcut form durumuna göre şekillenmesidir. SOFIFA sitesinden FIFA 22 oyununun son verisi olan, 18 Ağustos 2022 verileriyle çalışma gerçekleştirılmıştır. Web sitesinden çekilen veri temizlendikten sonra çalışmamız için ön işleme ile kullanılabilir hale getirilmiştir. Yapılan analizler sonucu bir tahminleme modeli kurulmuştur. Kurulan model yüksek oranda başarı sağlamıştır.

ABSTRACT

Determining the market values, in other words, the testimonials of the players, is a subject that has been discussed for a long time. From time to time, there may be large differences between the actual transfer fees and the anticipated fees. In determining this fee, there are many parameters such as the remaining contract period with the current team, age, injury history. The reason for taking the data from a football game in this study is that the data of the game is constantly updated and the individual data that are not determined in real life are found in this game. We will show that every parameter that affects the market value of football players cannot be considered statistically. The reason why we did not use the data determined by statistics is that the statistics separate the level of struggle in different leagues with a very unrealistic method. If we come to the question of why we use the data of this game, it is very successful in predicting match results and tournament champions with artificial intelligence in studies. The 2010, 2014 and 2018 world cup champions, respectively, were estimated correctly with these data. Another important feature of the data set is that it is constantly updated and shaped according to the current form state. The study was carried out with the data of August 18, 2022, which is the last data of the FIFA 22 game from the SOFIFA site. After the data retrieved from the website has been cleaned, it has been made available for our study with preprocessing. As a result of the analyzes made, a forecasting model was established. The established model has achieved a high degree of success.

1) GİRİŞ

Dünyanın en yaygın sporlarından biri olan futbolda transfer ücretlerinin neye göre belirlendiği yıllarda beri tartışma konusu olmuştur. Yapılan transferlerin çok fazla değişkeni vardır. Bu çalışmanın amacı, bu değişkenlerin belirlenmesi üzerinden veri analizi yapıp, makine öğrenmesiyle bir tahminleme modeli kurmaktır. Günümüzde market değerinin belirlenmesinde devletlerin vergi sisteminden, birbirleriyle olan tutumundan, klüp başkanlarının arasındaki bireysel ilişkilere kadar ince detaylar rol oynayabiliyor. Bu çalışmada göz önüne alacağımız veriler böyle görülemeyen detayları barındırmamaktadır. Veri seti 9901 oyuncunun 62 çeşit bireysel verisinden oluşmaktadır.

Sırasıyla; isim soyisim, yaş, oyun gücü, potansiyel, takım, boy, kilo, tercih ettiği ayak, en iyi oyun gücü, en iyi olduğu mevki, gelişime yatkınlık, market değeri, hücuma yatkınlık, çalım, bitiricilik, hava topu hakimiyeti, kısa pas yeteneği, topa yere değmeden vurma yeteneği, bireysel beceri, top sürme, falsolu vurma yeteneği, frikik vurma becerisi, uzun pas yeteneği, top kontrolü, hareketlilik, hızlanma, sürat, çeviklik, reaksiyon, denge, fiziksel güç, şut gücü, zıplama, dayanıklılık, kuvvet, uzun mesafeli vuruş yeteneği, zihinsel güç, saldırganlık, rakibi durdurma, konumlanma, vizyon, penaltı kullanma yeteneği, sakinlik, savunma yeteneği, rakibi bireysel markaja alma, ayakta müdahale, yerden kayarak müdahale, kalecilik becerileri, kalecinin topa uçması, kalecinin el yeteneği, kalecinin degaj yapması, kalecinin rakibe göre konumlanması, kaleci refleksleri, tüm istatistiklerin toplamı, kendi mevkiinin verilerinin toplamı, uluslararası itibar, hareket hızı, şut gücü veya kalecinin topa hakimiyeti, pas kabiliyeti veya kalecinin vuruşu, koşarak çalımlama veya kalecinin refleksleri, savunma hızı, fizik veya kalecinin konumlanması.

Bu verilerden sayısal olanlar 0 ile 100 arasında değişim göstermektedir. Oyuncuların performans verileri belirlenirken, oynadığı ligin zorluk derecesi göz önüne alınmaktadır. Bu sayede gerçekçi bir karşılaştırma ve rekabet ortamı oluşturulmaktadır.

1) UYGULAMA

2.1) VERİ ÇEKME

The screenshot shows the SOFIFA website interface. At the top, there's a navigation bar with links for PLAYERS, TEAMS, SQUADS, SHORTLISTS, and DISCUSSIONS. On the right side of the header are buttons for SIGN IN, language selection (English), and a search bar labeled "Search Player ...". Below the header, the page title is "Players" and the date is "AUG 18, 2022". A toolbar below the title includes buttons for Trending, Added, Updated, Free, On Loan, Removed, Customized, Create Player, and Calculator. There's also a search input field. To the left of the main content area, there's a sidebar with filters for "COLUMNS SELECTED", "BASKET", and "SEARCH". The search section includes dropdowns for Name, All Players, Continents, Nationality / Region, Leagues, and Teams. It also has sliders for Age (15 to 45), Overall Rating (0 to 99), and Potential (0 to 99). The main content area displays a table of player statistics. The columns include NAME, AGE, OVA, POT, TEAM & CONTRACT, HEIGHT, WEIGHT, FOOT, BOV, BP, GROWTH, VALUE, and ATTACK. The table lists several players: R. Lewandowski (FC Bayern München), L. Messi (Paris Saint Germain), K. Mbappé (Paris Saint Germain), M. Salah (Liverpool), K. De Bruyne (Manchester City), and K. Benzema (Real Madrid). Each player entry includes their profile picture, position, and detailed stats like height, weight, and foot.

Fig.1: SOFIFA web arayüzü

Verilerini çektiğimiz sitenin arayüzü bu şekildedir. Python kütüphanelerinden yardım alarak veri çekme işlemi gerçekleştirildi.

```
import pandas as pd
import numpy as np
from bs4 import BeautifulSoup
import requests
import string
import re
import matplotlib.pyplot as plt
%matplotlib inline

from scipy.stats import stats
import seaborn as sns
from scipy.stats import f_oneway
import math
from sklearn.linear_model import LinearRegression
pd.plotting.register_matplotlib_converters()
from sklearn.linear_model import LogisticRegression
from sklearn.datasets import make_blobs

import xgboost as xgb
from xgboost import XGBRegressor
from sklearn.model_selection import train_test_split, GridSearchCV,cross_val_score
from sklearn.metrics import mean_squared_error, r2_score
from sklearn import model_selection
import statsmodels.api as sm

pip install beautifulsoup4
Requirement already satisfied: beautifulsoup4 in /opt/anaconda3/lib/python3.8/site-packages (4.9.3)
Requirement already satisfied: soupsieve>1.2 in /opt/anaconda3/lib/python3.8/site-packages (from beautifulsoup4) (2
.2.1)
Note: you may need to restart the kernel to use updated packages.

pip install xgboost
```

Fig.2: Kütüphaneler

```

tablo=[]
i=0
while i < 10000:
    url="https://sofifa.com/?showCol%5B0%5D=ae&showCol%5B1%5D=hi&showCol%5B2%5D=wi&showCol%5B3%5D=ls&showCol%5B4%5D=ls&showCol%5B5%5D=ls&showCol%5B6%5D=ls&showCol%5B7%5D=ls&showCol%5B8%5D=ls&showCol%5B9%5D=ls"
    istek=requests.get(url)
    html=istek.text
    soup=BeautifulSoup(html,"lxml")
    #lxml hafif ve hızlı bir kütüphane olduğu için kullanıldı.
    rows=soup.find_all('tr')
    #html inceleyince "tr"-->"satır" ile "td"-->"sütun" arasında veriler

    for tr in rows:
        td = tr.find_all('td')
        td_str=str(td)
        clean_td = (re.sub(re.compile('<.*?>'),'',td_str))
        #Düzenli ifadeler (Regular Expressions) kütüphanesi ile,
        #verinin içindeki html kodları kaldırılmıştır.
        #'.*?' ifadesi ile aradaki(<...>) tüm değerler silindi.
        tablo.append(clean_td)
        result = pd.DataFrame(tablo)
    i+=60
    #her sayfada 60 oyuncu bulunmaktadır.

result

```

Fig.3: URL ile veri çekme kodları

0	
0	
1	\n, \nR. Lewandowski ST, 32, 92, 92, \n\n\n...
2	\n, \nL. Messi RW ST CF, 34, 92, 92, \n\n\n...
3	\n, \nK. Mbappé ST LW, 22, 91, 95, \n\n\nPa...
4	\n, \nM. Salah RW, 29, 91, 91, \n\n\nLiverp...
...	...
10182	\n, \nJ. Wießmeier RWB RM RB, 28, 65, 65, \n\...
10183	\n, \nD. Chima Chukwu ST, 30, 65, 65, \n\n\...
10184	\n, \nN. Madsen CDM CM, 21, 65, 75, \n\n\nS...
10185	\n, \nL. Montsma CB, 23, 65, 73, \n\n\nLinc...
10186	\n, \nE. Mero RM RW, 22, 65, 71, \n\n\nDelf...
10187 rows × 1 columns	

Fig.4: Çekilen verinin ilk hali

Şekil 4'te görüldüğü üzere çektiğimiz veri tek sütun halinde ve bilgilerin arasında istemediğimiz işaret ve yazılar bukunmakta. Veri temizleme aşamasında veriyi kullanılabilir hale getireceğiz.

2.2) VERİ TEMİZLEME

Çektiğimiz veride gereksiz semboller ve işaretleri çıkarıp veriyi 62 sütuna ayıracagız. İlk yapacağımız işlem veriden “\n” çıkarmak olacaktır. Sonrasında virgül ile ayrılmış tek satır olan verilerimizi sütunlara ayıracagız. Bu işlemlerden sonra elimizde olan veride bazı sütunlarda tahminlemede işimize yaramayacak verileri sütunlardan çıkarıyoruz. Kiralık oyuncularda bir sütun fazla bulunmaktadır, bu sorunu çözmek için sadece kiralık oyuncularda sütunları kaydırma işlemi yapacağız. Her bir verinin başında ve sonunda boşluk bulunmakta, bunu ortadan kaldıracağız. Sütunlara özellik isimlerini verip sonrasında verisi yanlış çekilen 11 oyuncu veriden çıkarılır. Veriyi kullanma aşamasına geçmeden önce verimizin dtype’ı yani veri tipi obje olarak görülmektedir, veri ile çalışma yapabilmek için bunu düzeltmemiz gerekmektedir. Bu işlemler sırasıyla şöyle gerçekleşmiştir;

```
i=0
for i in range(0,10187):
    result[0][i]=result[0][i].replace("\n","")
i+=1

result.drop_duplicates(subset=None, keep="first", inplace=True)

pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', 150)

for i in range(0,1):
    result = [result[0][i].split(",") for result[0][i] in result[0]]
    i+=1

result=pd.DataFrame(result)

result.drop(0,axis=0,inplace=True)
result.drop(0,axis=1,inplace=True)

for i in range(1,9913):
    result[1][i] = result[1][i].replace("CB","",).replace("LW","",).replace("LM","",).replace("RM","",).replace("LB","",)
    result[5][i] = result[5][i].replace("Jun 1","",).replace("Jun 30","",)
    result[6][i] = result[6][i].replace("2022 On Loan","",).replace("2024 On Loan","",).replace("2023 On Loan","",)
    result[5][i] = result[5][i].replace("1","",).replace("2","",).replace("3","",).replace("4","",).replace("5","",).replace("6","",)
    i+=1

for x in range(1,9912):
    if (result[63][x]) is not None:
        for i in range(6,63):
            result[i][x] = str(result[i+1][x])
            i+=1
        x+=1
    else :
        True
```

Fig.5: Veri temizleme kodları 1

```

result.drop(63,axis=1,inplace=True)
result.drop(0,axis=0,inplace=True)
result.drop(0,axis=1,inplace=True)

for i in range(1,9913):
    result[1][i] = result[1][i].replace("CB","",).replace("LW","",).replace("LM","",).replace("RM","",).replace("LB","",)
    result[5][i] = result[5][i].replace("Jun 1","",).replace("Jun 30","",)
    result[6][i] = result[6][i].replace("2022 On Loan","",).replace("2024 On Loan","",).replace("2023 On Loan","",)
    result[5][i] = result[5][i].replace("1","",).replace("2","",).replace("3","",).replace("4","",).replace("5","",).rep
    i+=1

for x in range(1,63):
    for i in range(1,9913):
        result[x][i] = result[x][i].strip()
        i+=1
    x+=1

```

Fig.6: Veri temizleme kodları 2

```

result.drop(64,axis=1,inplace=True)
result=result.set_axis(['name',
'age',
'overall',
'potential',
'team',
'height',
'weight',
'foot',
'best_overall',
'best_position',
'growth',
'value',
'attacking',
'crossing',
'finishing',
'heading_accuracy',
'short_passing',
'velleys',
'skill',
'dribbling',
'curve',
'freekick_accuracy',
'long_passing',
'ball_control',
'movement',
'acceleration',
'sprint_speed',
'agility',
'reactions',
'balance',
'power',
'shot_power',

```

Fig.7: Sütun isimlendirme

		name	age	overall	potential	team	height	weight	foot	best_overall	best_position	growth	value	attacking	crossing	finishing	he
1	R. Lewandowski	32	92	92	FC Bayern München	185cm	81kg	Right	92	ST	0	€119.5M	430	71	95		
2	L. Messi	34	92	92	Paris Saint Germain	169cm	67kg	Left	93	CAM	0	€69.5M	425	85	91		
3	K. Mbappé	22	91	95	Paris Saint Germain	182cm	73kg	Right	92	ST	4	€194M	411	78	93		
4	M. Salah	29	91	91	Liverpool	175cm	71kg	Left	91	RW	0	€129M	402	81	93		
5	K. De Bruyne	30	91	91	Manchester City	181cm	70kg	Right	91	CM	0	€125.5M	408	94	83		
...	
9908	J. Wießmeier B	28	65	65	Ried	171cm	71kg	Right	65	RWB	0	€650K	297	63	58		
9909	D. Chima Chukwu	30	65	65	Jamshedpur	180cm	74kg	Right	65	ST	0	€700K	288	40	64		
9910	N. Madsen	21	65	75	SC Heerenveen	193cm	76kg	Right	66	CDM	10	€1.5M	266	52	48		
9911	L. Montsma	23	65	73	Lincoln City	191cm	84kg	Right	67	CB	8	€1.5M	232	38	33		
9912	E. Mero	22	65	71	Delfin	173cm	78kg	Right	66	RM	6	€1.1M	287	61	55		

Fig.8: Verinin temizlenmiş hali

Veri setini istediğimiz formata getirdik ama analiz yapmak için birkaç değişikliğe ihtiyacımız var örneğin, kilo verisi üzerinde çalışabilmek için “kg”yi veri setinden çıkarmamız gerekmektedir.

2.3) VERİ ÖN İŞLEME

Veri setinde istatistiksel ve matematiksel işlemler yapmak için bazı değişikliklere ihtiyacımız var. Kuracağımız makine öğrenmesi modeli için bazı ön çalışmalar gerçekleştireceğiz. Sayısal verilerin yanındaki “€”, “kg”, “cm” gibi sözel verileri kaldıracağız. Veri tipinin python kütüphanelerinde çalışması için sayısal verilerin “integer” veya “float” yani tam sayı veya ondalıklı sayı formatına getirmemiz gerekiyor.

```
result["value"] = result["value"].str.replace("€", "")
result["value"] = result["value"].str.replace("M", "")
result.loc[result["value"].str.contains("K"), "value"] = result["value"].str.split("K").str[0].astype(float)/1000
result["height"] = result["height"].str.replace("cm", "")
result["weight"] = result["weight"].str.replace("kg", "")
result["physical_positioning"] = result["physical_positioning"].str.replace("]", "")
```

Fig.9: Veri setini ön işleme

result.info()				result.info()			
<class 'pandas.core.frame.DataFrame'>				<class 'pandas.core.frame.DataFrame'>			
Int64Index: 9901 entries, 1 to 9912				Int64Index: 9901 entries, 1 to 9912			
Data columns (total 62 columns):							
#	Column	Non-Null Count	Dtype	#	Column	Non-Null Count	Dtype
0	name	9901	non-null	0	name	9901	non-null
1	age	9901	non-null	1	age	9901	non-null
2	overall	9901	non-null	2	overall	9901	non-null
3	potential	9901	non-null	3	potential	9901	non-null
4	team	9901	non-null	4	team	9901	non-null
5	height	9901	non-null	5	height	9901	non-null
6	weight	9901	non-null	6	weight	9901	non-null
7	foot	9901	non-null	7	foot	9901	non-null
8	best_overall	9901	non-null	8	best_overall	9901	non-null
9	best_position	9901	non-null	9	best_position	9901	non-null
10	growth	9901	non-null	10	growth	9901	non-null
11	value	9901	non-null	11	value	9901	non-null
12	attacking	9901	non-null	12	attacking	9901	non-null
13	crossing	9901	non-null	13	crossing	9901	non-null
14	finishing	9901	non-null	14	finishing	9901	non-null
15	heading_accuracy	9901	non-null	15	heading_accuracy	9901	non-null
16	short_passing	9901	non-null	16	short_passing	9901	non-null
17	volleys	9901	non-null	17	volleys	9901	non-null
18	skill	9901	non-null	18	skill	9901	non-null
19	dribbling	9901	non-null	19	dribbling	9901	non-null
20	curve	9901	non-null	20	curve	9901	non-null
21	freekick_accuracy	9901	non-null	21	freekick_accuracy	9901	non-null
22	long_passing	9901	non-null	22	long_passing	9901	non-null
23	ball_control	9901	non-null				
24	movement	9901	non-null				
25	acceleration	9901	non-null				
26	sprint_speed	9901	non-null				

Fig.10: Veri tiplerini değiştirme

	age	overall	potential	height	weight	best_overall	growth	value	attacking	crossing	finishing	heading_accuracy	s
1	32	92	92	185	81	92	0	119.50	430	71	95	90	
2	34	92	92	169	67	93	0	69.50	425	85	91	70	
3	22	91	95	182	73	92	4	194.00	411	78	93	72	
4	29	91	91	175	71	91	0	129.00	402	81	93	59	
5	30	91	91	181	70	91	0	125.50	408	94	83	55	
...	
9908	28	65	65	171	71	65	0	0.65	297	63	58	58	
9909	30	65	65	180	74	65	0	0.70	288	40	64	70	
9910	21	65	75	193	76	66	10	1.50	266	52	48	57	
9911	23	65	73	191	84	67	8	1.50	232	38	33	64	
9912	22	65	71	173	78	66	6	1.10	287	61	55	55	

Fig.11: Veri setinin sadece sayısal değer barındıran hali

Bu işlemlerden sonra verimizi analiz etmeye hazırlız. Analiz edip, inceledikten sonra daha doğru bir model kurmak için çalışmalar ve denemeler yapacağız.

2.4) VERİ ANALİZİ

Günümüzde veri hakkında yeterli bilgiye sahip olunmadan yapılan çalışmalarla başarı oranı bir hayli düşüktür. Bunun için veri analizi yapılan çalışmalarla önemli bir faktör oluşturmaktadır. Bu aşamada seçilecek algoritmaların, kullanılacak formüllerin, hangi kütüphanenin daha faydalı olacağının ve verilerin arasındaki ilişkilerin kuvvetliliğinin belirlenmesini sağlayacağımız. Dağılım tabloları, korelasyon ilişkileri, ikili ilişki hipotez testleri gibi bazı analizler gerçekleştiriyoruz. Bu analizler sonucunda kuracağımız makine öğrenmesi modelinin doğru ve güvenilir bir sonuç vermesini sağlayacağımız.

	age	overall	potential	height	weight	best_overall	growth	value	attacking	crossing	finishing	heading_acc	
age	1.000000	0.112014	-0.473882	0.073204	0.197860	-0.012216	-0.810044	-0.107981	-0.041255	-0.058154	-0.056619	0.00	
overall	0.112014	1.000000	0.722461	0.032710	0.051146	0.975361	-0.158136	0.730703	0.253902	0.204660	0.194552	0.14	
potential	-0.473882	0.722461	1.000000	0.024116	-0.051973	0.790089	0.568466	0.638700	0.164275	0.130630	0.132796	0.07	
height	0.073204	0.032710	0.024116	1.000000	0.770817	0.026025	-0.004481	0.011465	-0.399187	-0.546307	-0.410544	0.06	
weight	0.197860	0.051146	-0.051973	0.770817	1.000000	0.025700	-0.135082	0.001036	-0.348976	-0.493315	-0.348867	0.05	
best_overall	-0.012216	0.975361	0.790089	0.026025	0.025700	1.000000	-0.032236	0.738508	0.292399	0.216953	0.231457	0.18	
growth	-0.810044	-0.158136	0.568466	-0.004481	-0.135082	-0.032236	1.000000	0.042680	-0.067512	-0.056969	-0.041849	-0.06	
value	-0.107981	0.730703	0.638700	0.011465	0.001036	0.738508	0.042680	1.000000	0.230346	0.186483	0.196161	0.11	
attacking	-0.041255	0.253902	0.164275	-0.399187	-0.348976	0.292399	-0.067512	0.230346	1.000000	0.842068	0.887830	0.67	
crossing	-0.058154	0.204660	0.130630	-0.546307	-0.493315	0.216953	-0.056969	0.186483	0.842068	1.000000	0.674556	0.40	
finishing	-0.056619	0.194552	0.132796	-0.410544	-0.348867	0.231457	-0.041849	0.196161	0.887830	0.674556	1.000000	0.41	
heading_accuracy	0.007337	0.143723	0.074335	0.059363	0.055441	0.181167	-0.064856	0.117118	0.678907	0.400881	0.414691	1.00	
short_passing	-0.089766	0.315963	0.258219	-0.384960	-0.359069	0.366551	-0.007196	0.277662	0.877619	0.781346	0.664096	0.63	
volleys	0.012744	0.228016	0.121058	-0.374952	-0.310303	0.256378	-0.098430	0.203965	0.887641	0.670513	0.893232	0.43	
skill	-0.061130	0.264416	0.189043	-0.508397	-0.455488	0.301563	-0.044650	0.237731	0.927593	0.882127	0.808754	0.50	
dribbling	-0.171909	0.217597	0.214883	-0.515374	-0.470663	0.266415	0.047960	0.220668	0.920521	0.856770	0.830136	0.51	
curve	-0.012891	0.242629	0.146925	-0.497568	-0.437829	0.261264	-0.078878	0.209513	0.842202	0.839288	0.772621	0.35	
freekick_accuracy	0.057842	0.186440	0.060174	-0.471006	-0.393018	0.200718	-0.135908	0.158559	0.774576	0.743992	0.734087	0.31	
long_passing	-0.017624	0.309472	0.219202	-0.351982	-0.335483	0.347443	-0.055193	0.253686	0.718465	0.716916	0.496196	0.48	
ball_control	-0.137377	0.261401	0.237253	-0.438057	-0.407532	0.314420	0.027786	0.248911	0.932904	0.823510	0.782968	0.63	

Fig.12: Korelasyon matrisi

Bir korelasyon matrisi, değişkenler arasındaki korelasyon katsayılarını gösteren bir tablodur. Tablodaki her hücre, iki değişken arasındaki ilişkiyi gösterir. Bir korelasyon matrisi, daha gelişmiş bir analize girdi olarak ve gelişmiş analizler için bir teşhis olarak verileri özetlemek için kullanılır.

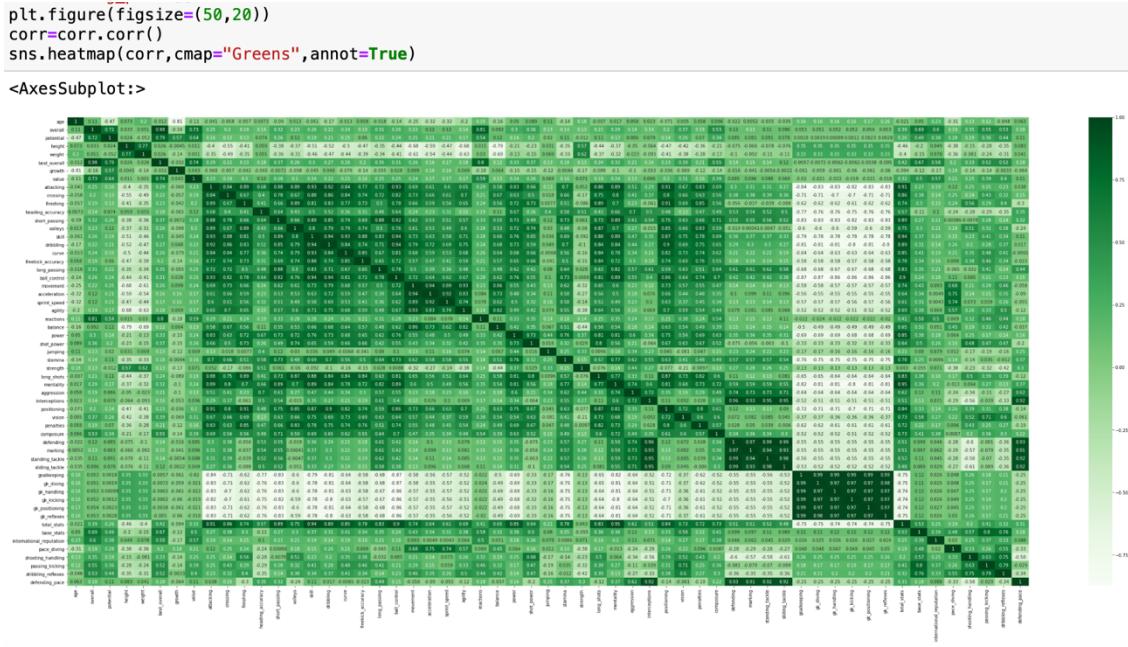


Fig.13: Korelasyon ısı haritası

Tüm parametrelerin birbiriyle olan koreleasyon katsayısını açık tondan koyu tona güçlendigini gösteren harita.

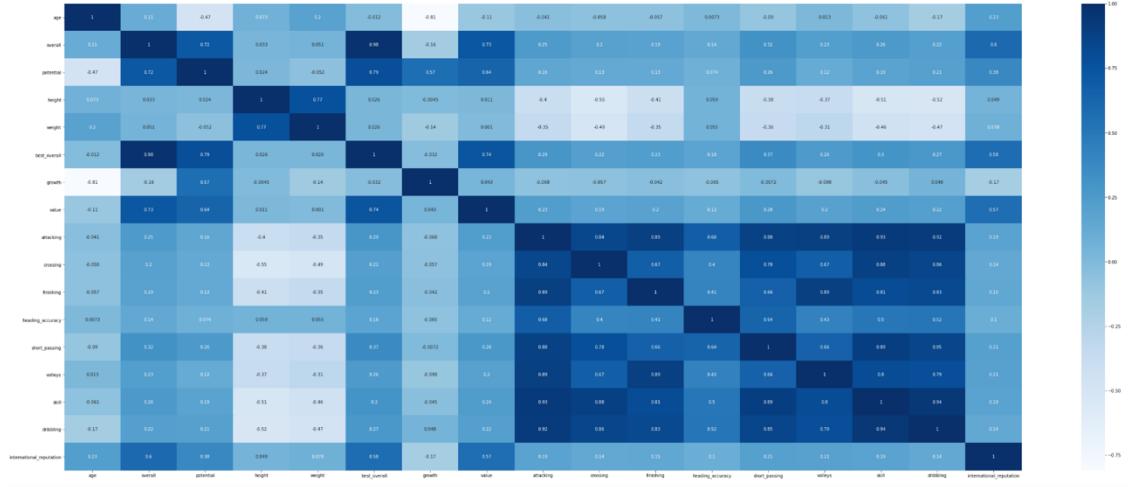


Fig.14: Korelasyon ısı haritası 2

Korelasyon matrisinde oyuncuların market değeriyle en yüksek olumlu veya olumsuz ilişkiye sahip olan yaş, oyun gücü, potansiyel, boy, kilo, en iyi oyun gücü, gelişime yatkınlık, market değeri, atak gücü, çalım yeteneği, bitiricilik, hava topu hakimiyeti, kısa pas, vole, beceri, depar ve uluslararası itibar verilerinin olduğu korelasyon matrisidir. Market değerini en fazla etkileyen etkenin uluslararası itibar olduğunu saptadık.

OLS Regression Results									
Dep. Variable:	y	R-squared (uncentered):	0.971						
Model:	OLS	Adj. R-squared (uncentered):	0.971						
Method:	Least Squares	F-statistic:	7093.						
Date:	Fri, 09 Dec 2022	Prob (F-statistic):	0.00						
Time:	17:36:32	Log-Likelihood:	-18642.						
No. Observations:	9901	AIC:	3.738e+04						
Df Residuals:	9854	BIC:	3.772e+04						
Df Model:	47								
Covariance Type:	nonrobust								
	coef	std err	t	P> t	[0.025	0.975]			
age	-0.8028	0.007	-114.846	0.000	-0.816	-0.789			
overall	0.9266	0.015	62.885	0.000	0.898	0.956			
potential	0.3855	0.009	44.495	0.000	0.368	0.402			
height	-0.3575	0.003	-137.872	0.000	-0.363	-0.352			
weight	0.0132	0.004	3.252	0.001	0.005	0.021			
body_fat	0.0720	0.002	3.212	0.001	0.117	0.020			

Fig.15: OLS regresyon çıktıları

Doğrusal regresyon, bir dizi bağımsız ve bağımlı değişken arasındaki ilişkiyi analiz etmek için basit ama güçlü bir araçtır. OLS tarafından ortaya konan çeşitli istatistikleri analiz etmek önemli bir adımdır. İstatistikte model seçimi bir sanattır. Bu sanatın anlamlı hale gelmesinde pek çok faktör dikkate alınmaktadır. Her bir istatistiğe tekereker bakmak ve seçim yaparken anlamlılığı göz önünde bulundurmak gereklidir. OLS regresyon çıktıları bize bazı verilerin olumlu ilişkisinin veya olumsuz ilişkisinin anlamlı olup, olmadığını söylemektedir. Nitekim bazı özelliklerin yüksek korelasyon ilişkisine sahipken anlamlı çıkmaması veya mantık olarak olumlu etkileyebilecek bir özelliğin olumsuz çıkması gibi durumlarla karşılaşmıştık. Bu parametreleri kullanmamayı tercih edeceğiz.

```
plt.scatter(statistical_data["value"], statistical_data["potential"])
plt.show()
```

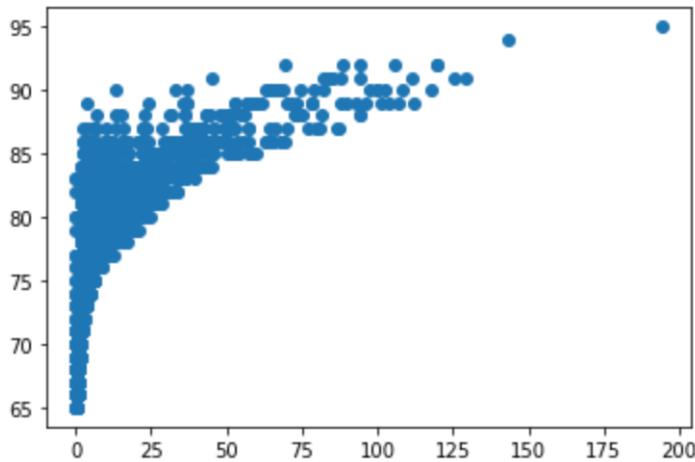


Fig.16: Market deüeri ve potansiyel ilişkisi

0-25 milyon euro bandında bir yığılma gözlemliyoruz, bu bandın dışına çıktıktan sonra daha anlamlı bir ilişki oluşmaktadır.

result.describe()											
	age	overall	potential	height	weight	best_overall	growth	value	attacking	crossing	finishing
count	9901.000000	9901.000000	9901.000000	9901.000000	9901.000000	9901.000000	9901.000000	9901.000000	9901.000000	9901.000000	9901.000000
mean	26.536612	70.861428	73.616806	181.441572	75.755075	71.640137	2.755378	4.902518	274.503585	55.025048	51.005151
std	4.238186	4.360149	5.233074	6.918074	7.116300	4.397575	3.664318	10.127508	72.094008	18.032725	19.626962
min	16.000000	65.000000	65.000000	156.000000	54.000000	65.000000	0.000000	0.000000	43.000000	6.000000	2.000000
25%	23.000000	67.000000	69.000000	176.000000	71.000000	68.000000	0.000000	1.200000	254.000000	46.000000	37.000000
50%	26.000000	70.000000	73.000000	182.000000	75.000000	71.000000	1.000000	1.900000	292.000000	61.000000	56.000000
75%	29.000000	73.000000	77.000000	186.000000	80.000000	74.000000	5.000000	3.600000	319.000000	67.000000	67.000000
max	43.000000	92.000000	95.000000	205.000000	103.000000	93.000000	21.000000	194.000000	434.000000	94.000000	95.000000

Fig.17: Tüm verinin istatistiksel özeti

Bu çıktı yardımıyla 9901 oyuncunun yaş ortalamasının 26.53, en yaşlı oyuncunun 43, en genç oyuncunun 16 olması gibi bütün sütunların bilgilerinin özetiyle ulaşabiliyoruz.

```
result.groupby(['best_position']).count().plot(kind='pie', y='name', figsize=(15,15))
```

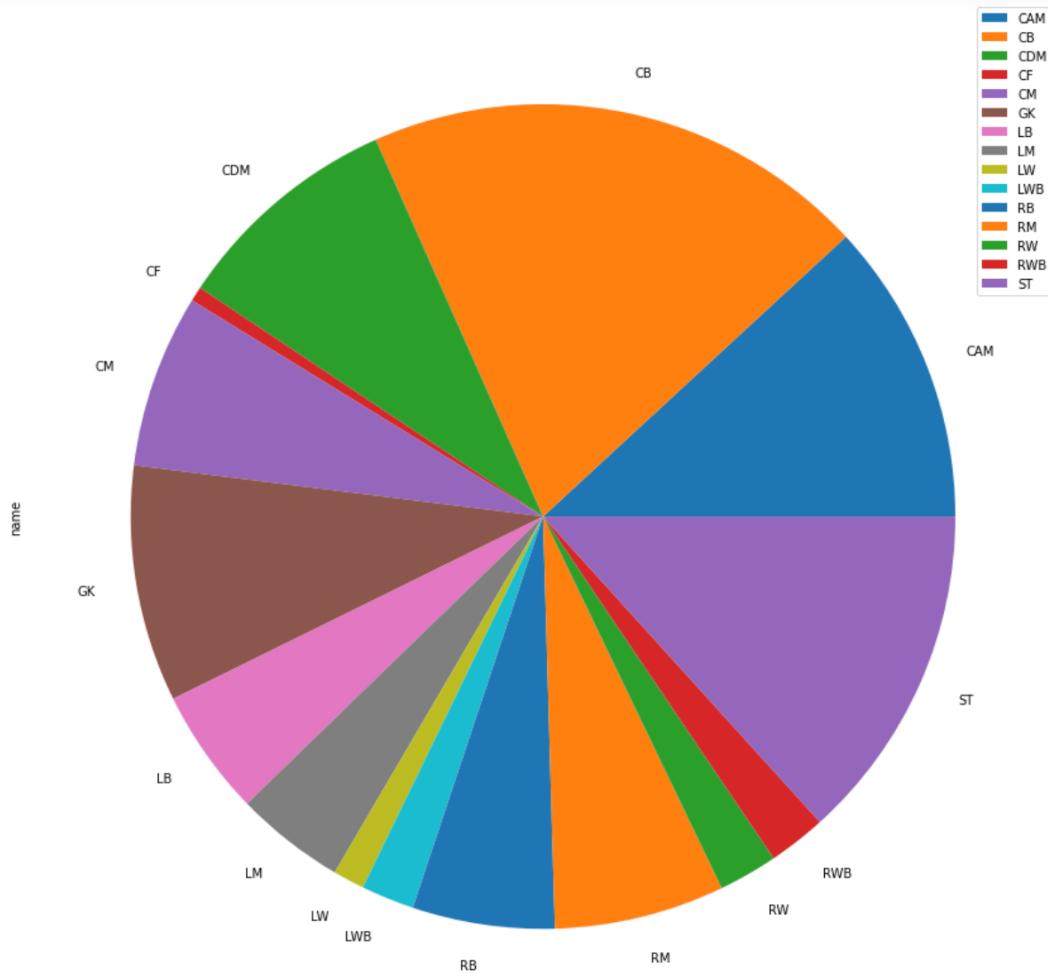


Fig.18: Mevkilerin pasta dilimi dağılımı

Oyuncuların oynadıkları mevkilerin dağılımı bu şekildedir. Ofansif orta saha (CAM), stoper (CB), ön libero (CDM), forvet (CF), merkez orta saha (CM), kaleci (GK), sol bek (LB), sol orta saha (LM), sol kanat (LW), hücumcu sol bek (LWB), sağ bek (RB), sağ orta saha (RM), sağ kanat (RW), hücumcu sağ bek (RWB) ve santrafor (ST). Sayısal verilerin haricinde bulunan az sayıdaki sözel verilerimizi inceledik. Bu verilerden bir anlamlılık çıkarmaya çalıştık.

```

ttest,pval=stats.ttest_ind(result[result["foot"]=="Right"]["value"],
                           result[result["foot"]=="Left"]["value"])
if pval<0.05:
    print("H0'i reddedebiliriz.")
else :
    print("H0'i reddedemeyiz.")
print("p value: ",pval)

H0'i reddedemeyiz.
p value:  0.38716025585882197

ttest,pval=stats.ttest_ind(result[result["best_position"]=="ST"]["value"],
                           result[result["best_position"]=="CAM"]["value"])
if pval<0.05:
    print("H0'i reddedebiliriz.")
else :
    print("H0'i reddedemeyiz.")
print("p value: ",pval)

H0'i reddedebiliriz.
p value:  0.0005853670703830149

fvalue,pvalue =stats.f_oneway(result[result["best_position"]=="ST"]["value"],
                               result[result["best_position"]=="CAM"]["value"])

if pvalue<0.05:
    print("H0'i reddedebiliriz.")
else :
    print("H0'i reddedemeyiz.")
print("F-VALUE: ",fvalue," P-VALUE: ",pvalue)

H0'i reddedebiliriz.
F-VALUE:  11.852461561650573   P-VALUE:  0.0005853670703838166

```

Fig.19: İstatistiksel inceleme

Geçekleştirilen hipotez testlerinde amaç sözel verilerin market değeri ile ilişkisini saptamaktı. Örnek olarak verilen bu üç testin ilkinde oyuncuların kullanmayı tercih ettiği ayakların market değeriyle alakasını test ettik t-test ile gerçekleştirdiğimiz test olumsuz çıktı, yani tercih edilen ayak market değerini etkilemez hipotezimizi reddedemedik. İkinci ve üçüncüörnekte ise en iyi olduğu pozisyon ile market değeri ilişkisi test edildi. Hem “t-test” hem de “f-test” yöntemleriyle test ettiğimiz santrafor pozisyonunda oynamak ile ofansif orta saha olarak oynamak market değerlerini etkilemez hipotezimizi reddediliyoruz. Bunun anlamı santrafor mevkii market değerini artıran bir unsur oluşturmaktadır.

		name	age	overall	potential	team	height	weight	foot	best_overall	best_position	growth	value
1	R.	Lewandowski	32	92	92	FC Bayern München	185cm	81kg	Right	92	ST	0	€119.5M
2	L.	Messi	34	92	92	Paris Saint Germain	169cm	67kg	Left	93	CAM	0	€69.5M
3	K.	Mbappé	22	91	95	Paris Saint Germain	182cm	73kg	Right	92	ST	4	€194M
4	M.	Salah	29	91	91	Liverpool	175cm	71kg	Left	91	RW	0	€129M
5	K.	De Bruyne	30	91	91	Manchester City	181cm	70kg	Right	91	CM	0	€125.5M
6	K.	Benzema	33	91	91	Real Madrid	185cm	81kg	Right	91	CF	0	€84M
7	Cristiano	Ronaldo	36	91	91	Manchester United	187cm	83kg	Right	91	ST	0	€45M
8	N.	Kanté	30	90	90	Chelsea	168cm	70kg	Right	90	CDM	0	€100M
9	V.	van Dijk	29	90	90	Liverpool	193cm	92kg	Right	90	CB	0	€100M
10	Neymar Jr		29	90	90	Paris Saint Germain	175cm	68kg	Right	90	LW	0	€117.5M

Fig.20: En yüksek oyun gücüne sahip 10 oyuncu

		name	age	overall	potential	team	height	weight	foot	best_overall	best_position	growth	value
3	K.	Mbappé	22	91	95	Paris Saint Germain	182cm	73kg	Right	92	ST	4	€194M
22	E.	Haaland	20	88	94	Borussia Dortmund	194cm	94kg	Left	90	ST	6	€143.5M
1	R.	Lewandowski	32	92	92	FC Bayern München	185cm	81kg	Right	92	ST	0	€119.5M
94	Pedri		18	84	92	FC Barcelona	174cm	61kg	Right	86	CM	8	€88.5M
2	L.	Messi	34	92	92	Paris Saint Germain	169cm	67kg	Left	93	CAM	0	€69.5M
97	P. Foden		21	84	92	Manchester City	171cm	69kg	Left	87	CAM	8	€94.5M
33	F. de Jong		24	87	92	FC Barcelona	180cm	74kg	Right	89	CM	5	€119.5M
23	G.	Donnarumma	22	88	92	Paris Saint Germain	196cm	90kg	Right	88	GK	4	€106M
14	Ederson		27	89	91	Manchester City	188cm	86kg	Left	89	GK	2	€94M
99	K. Havertz		22	84	91	Chelsea	189cm	82kg	Left	86	CAM	7	€85.5M

Fig.21: En yüksek potansiyele sahip 10 oyuncu

		name	age	overall	potential	team	height	weight	foot	best_overall	best_position	growth	value
3	K. Mbappé	22	91	95	Paris Saint Germain	182	73	Right	92	ST	4	194.0	
22	E. Haaland	20	88	94	Borussia Dortmund	194	94	Left	90	ST	6	143.5	
4	M. Salah	29	91	91	Liverpool	175	71	Left	91	RW	0	129.0	
5	K. De Bruyne	30	91	91	Manchester City	181	70	Right	91	CM	0	125.5	
1	R. Lewandowski	32	92	92	FC Bayern München	185	81	Right	92	ST	0	119.5	
33	F. de Jong	24	87	92	FC Barcelona	180	74	Right	89	CM	5	119.5	
10	Neymar Jr	29	90	90	Paris Saint Germain	175	68	Right	90	LW	0	117.5	
16	H. Kane	27	89	89	Tottenham Hotspur	188	89	Right	89	ST	0	112.0	
21	Rúben Dias	24	88	91	Manchester City	187	82	Right	90	CB	3	111.5	
13	J. Kimmich	26	89	90	FC Bayern München	177	75	Right	89	CDM	1	108.0	

Fig.22: En yüksek market değerine sahip 10 oyuncu

Bu üç tabloda gördüğümüz sonuçlar yaptığımız hipotez testlerini doğrular nitelikte. Santrafor mevkinin etkin olup, tercih edilen ayağın etkisiz olduğunu ilk 10 oyuncudan çıkarabiliriz. Analiz işlemleri bittikten sonra modelimizi kurup bu modelin tahminleme işlemini gerçekleştireceğiz.

3) TAHMİNLEME

Tahminleme yanş tahmine dayalı modelleme, elimizde bulunan veri kümelerinden tercih ettiğimiz boyutta örnekleri analiz ederek tahmin etmek için kullanılan matematiksel bir yöntemdir. Çok büyük veri setleri ile çalıştığımızda gelecek tahmini yapmak için tüm veriyi kullanmak zaman ve yeterli ekipman anlamında çok külfetli olabiliyor. Bu zorlu süreci kolaylaştıran makine öğrenmesi, doğru parametrelerle bize çok daha kısa bir sürede gerçeğe yakın sonuçlar bulmamızı sağlıyor. Bu projede tahminleme için küçük veri setleriyle birçok algoritma denenmiştir. En verimli ve en hızlı sonuç veren algoritma XGBoost algoritması olmuştur.

XGBoost, gradyan artırılmış ağaçlar algoritmasının bir açık kaynak uygulamasıdır. Gradient boosting, bir dizi daha basit, daha zayıf modelin tahminlerini birleştirerek bir hedef değişkeni doğru bir şekilde tahmin etmeye çalışan denetimli bir öğrenme algoritmasıdır. Regresyon için gradyan artırmayı kullanırken, zayıf öğrenenler regresyon ağaçlarıdır ve her regresyon ağıacı, sürekli bir puan içeren yapraklarından birine bir girdi veri noktası eşler. XGBoost, dışbükey bir kayıp işlevini (öngörülen ve hedef çıktılar arasındaki farka dayalı olarak) ve model karmaşıklığı için bir ceza terimini (başka bir deyişle, regresyon ağıacı işlevleri) birleştiren düzenlileştirilmiş (L1 ve L2) bir amaç işlevini en aza indirir. Eğitim, son tahmini yapmak için daha sonra önceki ağaçlarla birleştirilen önceki ağaçların artıklarını veya hatalarını tahmin eden yeni ağaçlar ekleyerek yinelemeli olarak ilerler. Yeni modeller eklerken kaybı en aza indirmek için bir gradyan iniş algoritması kullanıldığından buna gradyan artırma denir.

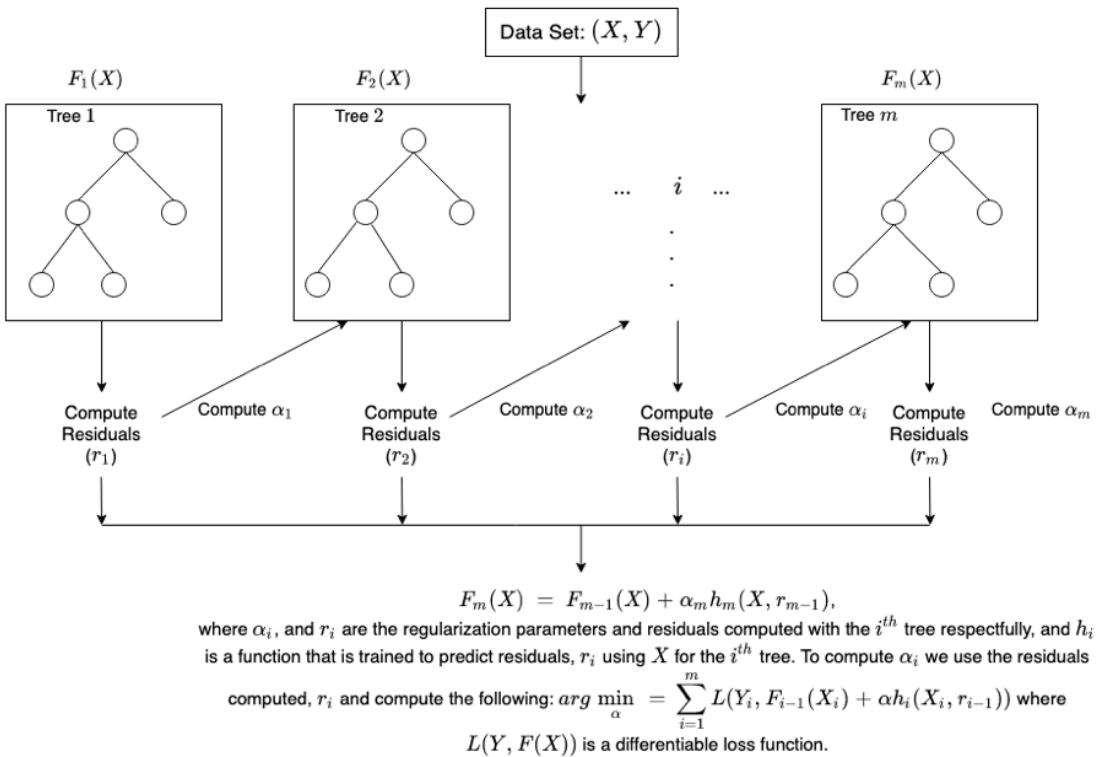


Fig.23: Gradyan ağacı güçlendirme

```

X = statistical_data[["age","overall","potential","height","growth","reactions","international_reputation"]]
y = statistical_data[["value"]]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size = 0.2, random_state = 100)

xgb = XGBRegressor()

parametre = {"colsample_bytree": [0.3, 0.5, 0.7],
             "learning_rate": [0.01, 0.05, 0.09],
             "max_depth": [2, 3, 4, 5, 6],
             "n_estimators": [100, 500, 1000, 3000]}

grid = GridSearchCV(xgb, parametre, cv = 5, n_jobs = -1, verbose = 2)

grid.fit(X_train, y_train)
  
```

Fig.24: Tahminleme modeli 1

```

grid.fit(X_train, y_train)

Fitting 5 folds for each of 180 candidates, totalling 900 fits

GridSearchCV(cv=5,
             estimator=XGBRegressor(base_score=None, booster=None,
                                    callbacks=None, colsample_bylevel=None,
                                    colsample_bynode=None,
                                    colsample_bytree=None,
                                    early_stopping_rounds=None,
                                    enable_categorical=False, eval_metric=None,
                                    feature_types=None, gamma=None, gpu_id=None,
                                    grow_policy=None, importance_type=None,
                                    interaction_constraints=None,
                                    learning_rate=None, m...
                                    max_cat_to_onehot=None, max_delta_step=None,
                                    max_depth=None, max_leaves=None,
                                    min_child_weight=None, missing=nan,
                                    monotone_constraints=None, n_estimators=100,
                                    n_jobs=None, num_parallel_tree=None,
                                    predictor=None, random_state=None, ...),
             n_jobs=-1,
             param_grid={'colsample_bytree': [0.3, 0.5, 0.7],
                         'learning_rate': [0.01, 0.05, 0.09],
                         'max_depth': [2, 3, 4, 5, 6],
                         'n_estimators': [100, 500, 1000, 3000]},
             verbose=2)

```

```
grid.best_params_
```

```
{'colsample_bytree': 0.7,
 'learning_rate': 0.05,
 'max_depth': 3,
 'n_estimators': 1000}
```

```
parametre2 = XGBRegressor(colsample_bytree = 0.7,
                           learning_rate = 0.05,
                           max_depth = 3,
                           n_estimators = 1000)
```

```
prediction_model = parametre2.fit(X_train, y_train)
```

Fig.25: Tahminleme modeli 2

Tahminleme modelimizi, veri setinin yüzde 20'lik bir kısmıyla oluşturacağız. Market değerini tahminlemek için analiz ettiğimiz ve yüksek bağlantı düzeyine sahip olan 7 parametre kullanıldı. Algoritma için parametre seçimi gerçekleştirildi, birden fazla parametre değerleri tek tek denenerek en yüksek verimliliğe sahip parametreler saptandı. Bu parametrelerin kısaca ne işe yaradığını özetlemek gerekirse;

“Colsample_bytree”, her ağaç oluştururken sütunların alt örnek oranıdır. Alt örnekleme, oluşturulan her ağaç için bir kez gerçekleşir.

“Learning_rate”, öğrenme oranıdır. Kayıp gradyan inişine göre ağıumızın ağırlıklarındaki ayarlamayı tanımlayan böyle bir hiper parametredir. Optimal ağırlıklara doğru ne kadar hızlı veya yavaş ilerleyeceğimizi belirler.

“Max_depth”, maksimum derinlik; bu parametre her ağaçın maksimum derinliğini belirtir.

“N_estimators”, maksimum oylamayı veya tahmin ortalamalarını almadan önce oluşturmak istediğiniz ağaç sayısıdır. Daha fazla sayıda ağaç size daha iyi performans sağlar ancak kodunuza yavaşlatır.

Bulduğumuz parametre değerlerini modelimize oturtup sonuçları gözlemleye geçiş yapıyoruz.

	predicted	real	difference
0	2.150711	2.0	0.150711
1	0.664790	0.7	-0.035210
2	0.963495	1.0	-0.036505
3	32.545177	35.5	-2.954823
4	3.162606	2.8	0.362606
...
1976	20.641132	19.0	1.641132
1977	1.670390	1.7	-0.029610
1978	1.355993	1.2	0.155993
1979	1.863774	1.7	0.163774
1980	1.033360	1.2	-0.166640

1981 rows × 3 columns

Fig.26: Tahmin, gerçek değerler ve aralarındaki fark

```
z[["predicted","real"]].plot(kind='bar',figsize=(7,7))
plt.show()
```

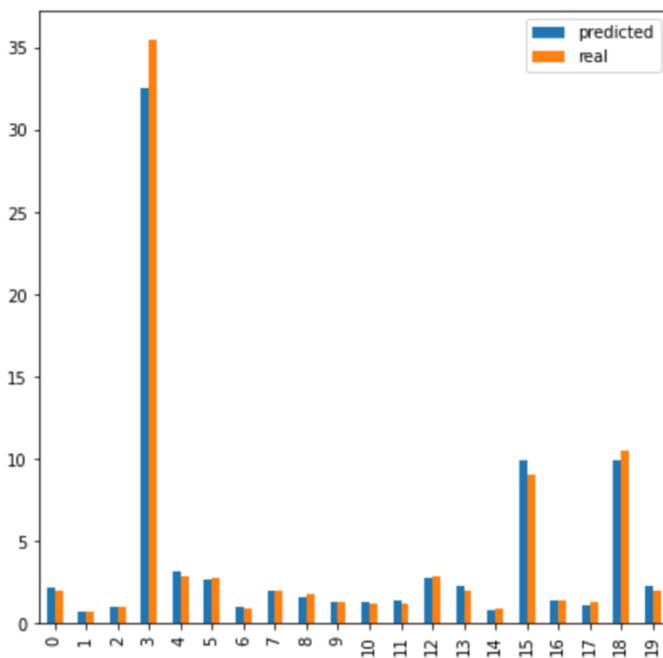


Fig.27: Tahmin ve gerçek değerlerin histogramı

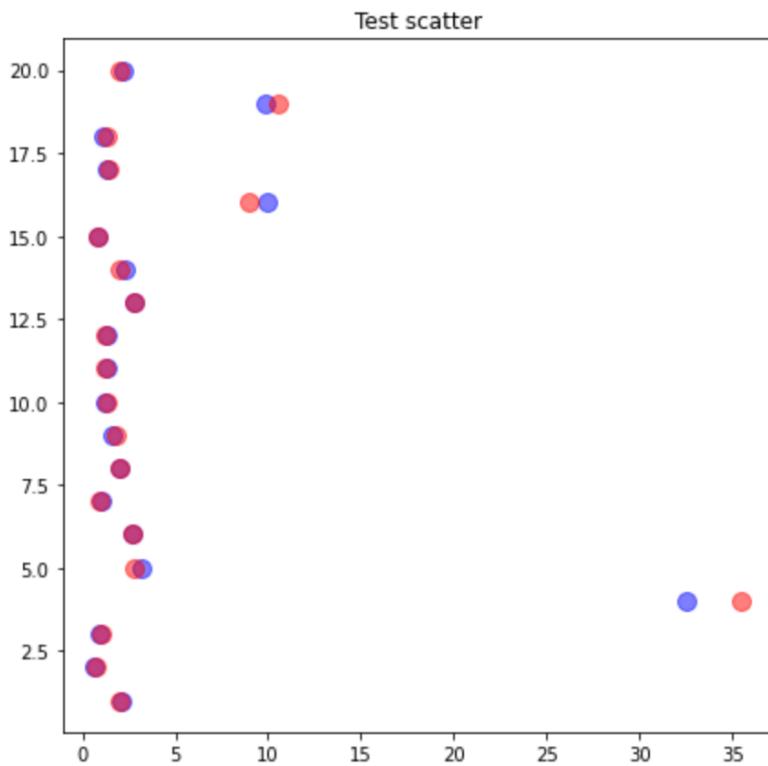


Fig.28: Tahmin ve gerçek değerler (kırmızı: gerçek, mavi: tahmin)

İlk yaptığımız gözlemlere göre tahmin sonuçlarımız başarılı gözükmekte. Sonuçların başarılı olup olmadığını ölçmek için istatistiksel gözlemler gerekmekte, aşağıda modelimizin verimliliğini anlayabilmek için incelemeye bulunacağız.

```

print("TEST SCORE: ", prediction_model.score(X_test, y_test))
print("TRAIN SCORE: ", prediction_model.score(X_train, y_train))
print('Mean Absolute Error:', mean_absolute_error(y_test, res))
print('Mean Squared Error:', mean_squared_error(y_test, res))
print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, res)))
print("R^2: ", r2_score(y_test, res))

```

TEST SCORE: 0.9926224454300299
TRAIN SCORE: 0.9955605443657698
Mean Absolute Error: 0.3598646954799438
Mean Squared Error: 0.8273760612417723
Root Mean Squared Error: 0.9096021444795369
R^2: 0.9926224454300299

Fig.29: İstatistiksel tahmin sonuçları

“Train score” modelin eğitim verilerine nasıl genelleştirildiği veya uydurulduğu.

Model, çok fazla varyansa sahip bir veriye çok iyi uyuyorsa, bu aşırı uyuma neden olur. Bu, Test Puanında kötü sonuca neden olur. Çünkü model, eğitim verilerine uyması için çok eğildi ve çok zayıf bir şekilde genellendi. Yani, genelleme amaçtır. 0-1 arasında olup 1’e yakın olması iyi bir sonuçtur.

“Test score” modelimizin hazır olduğu zamandır. Bu adımdan önce bu veri setinde değişiklik yapmadık. Yani, bu gerçek gerçek verimizi temsil ediyor. 0-1 arasında olup 1’e yakın olması iyi bir sonuçtur.

“MAE” Büyük hataları cezalandırmadığı için MSE'ye kıyasla aykırı değerlere karşı çok hassas değildir. Genellikle performans sürekli değişken veriler üzerinde ölçüldüğünde kullanılır. Ağırlıklı bireysel farkların eşit olarak ortalamasını alan doğrusal bir değer verir. Değer ne kadar düşükse, modelin performansı o kadar iyidir.

“MSE” En sık kullanılan metriklerden biridir, ancak tek bir kötü tahmin, tüm modelin tahmin yeteneklerini mahvedecekse, yani veri kümesi çok fazla gürültü içerdiginde, en az yararlıdır. Veri kümesi aykırı değerler veya beklenmeyen değerler (çok yüksek veya çok düşük değerler) içerdiginde en kullanışlıdır.

“RMSE”de hataların ortalaması alınmadan önce kareleri alınır. Bu temel olarak, RMSE'nin daha büyük hatalara daha yüksek bir ağırlık atadığı anlamına gelir. Bu, büyük hatalar olduğunda RMSE'nin çok daha yararlı olduğunu ve modelin performansını büyük ölçüde etkilediğini gösterir. Bu metrikte de değer ne kadar düşükse modelin performansı o kadar iyidir.

“ R^2 ”, modelin uyumunu açıklar 0-1 arasında olup 1’e yakın olması modelin uyumunun iyi olduğunu gösterir.

Bu sonuçlara göre modelimiz çok başarılı bir şekilde çalışmıştır. İlk yaptığımız grafik incelemesinde olduğu gibi istatistiksel sonuçlar da bize modelin başarılı olduğunu kanıtladı.

4) SONUÇ

Özetle veri setimizi link üzerinden çekme, temizleme, ön işleme, analiz etme ve son olarak modellereyerek inceledik ve bir tahminleme yaptık. Yaptığımız tahminleme büyük oranda başarı sağladı. Tahminleme modellerinin amacı kısa sürede ve yüksek bilişim gereksinimlerine ihtiyaç duymadan gerçeğe yakın sonuç bulmaya yarar. Zamandan ve gereksinimden tasarruf etmemizi sağlar ve geleceğe dair çıkarımlar yapmamızı kolaylaştırır. Birçok alanda kullanılan bu yöntem, büyük verilerin incelenmesinde ve analiz edilmesinde bizlere yardımcı olur. FIFA 22 oyununda 20000'den fazla futbolcu, 700'den fazla takım bulunmaktadır. Her oyuncunun bireysel olarak 62 çeşit verisi bulunmaktadır. Basit olarak 1240000 verinin bulunduğu bu veri setinde yaptığımız tahminler yaklaşık %99 oranında başarı sağlamıştır. Toplam veriye kıyasla aldığımız %20'lik örneklem yaklaşık %99 benzerlik sağlayıp, sonuçların başarı oranını anlamlı kılmaktadır. Veri setinin market değeri adına en önemli parametresi olan uluslararası itibar, gerçek hayatı da bonservis belirlenmesinde çok önemli bir faktördür. Futbolda elde edilen reklam gelirleri, tüm gelirlerin büyük bir kısmını oluşturmaktadır. Bu yüzden uluslararası itibar ödenen bonservis bedelinin klüpler adına geri kazanılmasında önemli bir faktör oluşturur. Bu da oyunun verilerinin gerçek hayatı oldukça alakalı ve uyumlu olduğunu göstermektedir.

KAYNAKÇA

[1] Veri seti

<https://sofifa.com/?showCol%5B0%5D=ae&showCol%5B1%5D=hi&showCol%5B2%5D=wi&showCol%5B3%5D=pf&showCol%5B4%5D=oa&showCol%5B5%5D=pt&showCol%5B6%5D=bo&showCol%5B7%5D=bp&showCol%5B8%5D=gu&showCol%5B9%5D=vl&showCol%5B10%5D=cr&showCol%5B11%5D=fi&showCol%5B12%5D=he&showCol%5B13%5D=sh&showCol%5B14%5D=vo&showCol%5B15%5D=dr&showCol%5B16%5D=cu&showCol%5B17%5D=fr&showCol%5B18%5D=lo&showCol%5B19%5D=bl&showCol%5B20%5D=ac&showCol%5B21%5D=sp&showCol%5B22%5D=ag&showCol%5B23%5D=re&showCol%5B24%5D=ba&showCol%5B25%5D=so&showCol%5B26%5D=ju&showCol%5B27%5D=st&showCol%5B28%5D=sr&showCol%5B29%5D=ln&showCol%5B30%5D=ar&showCol%5B31%5D=in&showCol%5B32%5D=po&showCol%5B33%5D=vi&showCol%5B34%5D=pe&showCol%5B35%5D=cm&showCol%5B36%5D=ma&showCol%5B37%5D=sa&showCol%5B38%5D=sl&showCol%5B39%5D=tg&showCol%5B40%5D=gd&showCol%5B41%5D=gh&showCol%5B42%5D=gc&showCol%5B43%5D=gp&showCol%5B44%5D=gr&showCol%5B45%5D=ir&showCol%5B46%5D=pac&showCol%5B47%5D=sho&showCol%5B48%5D=pas&showCol%5B49%5D=dri&showCol%5B50%5D=def&showCol%5B51%5D=phy&showCol%5B52%5D=ta&showCol%5B53%5D=ts&showCol%5B54%5D=to&showCol%5B55%5D=tp&showCol%5B56%5D=te&showCol%5B57%5D=td&showCol%5B58%5D=tt&showCol%5B59%5D=bs&r=220069&set=true>

[2] Futbolcuların market değerinde önemli olan nedir?

<https://www.colossusbets.com/blog/market-value-of-football-players/>

[3] Bir oyuncuya nasıl değer biçilir?

<https://theathletic.com/3085749/2022/01/27/premier-league-how-do-you-value-a-player/>

[4] Korelasyon matrisi nedir?

<https://www.displayr.com/what-is-a-correlation-matrix/>

[5] OLS sonuçlarını nasıl yorumlamalıyım?

<https://jyotiyadav99111.medium.com/statistics-how-should-i-interpret-results-of-ols-3bde1ebeec01>

[6] XGBoost nasıl çalışır?

<https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost-HowItWorks.html>

[7] Model Performansını Değerlendirmek-Metrikler

<https://medium.com/deep-learning-turkiye/model-performansini-değerlendirmek-metrikler-cb6568705b1>