



# SCOUTERS

FIFA

PLAYER RECOMMENDATION SYSTEM

# Contents



- Master of Plan
- INTRODUCTION
  - FIFA
  - FIFA Career Mode
  - Decision of Problem
  - Significance of Problem
  - Aim of Project
  - Scope of Project
  - Methodology
- INPUT DATA ANALYSIS
- DEVELOPMENT OF DECISION MODEL
- DEVELOPMENT OF DSS



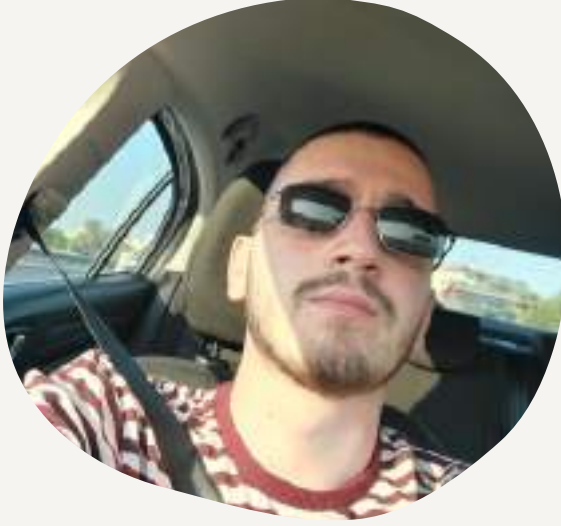
Scouters

# THE TEAM

3



BUĞRA  
KANDEMİR



YAVUZ SELİM  
KABAKCI



ENES YAPICI



UMUT DURAN



MEHMET ALİ  
ŞAHİN



YUNUS MALİK  
YAVUZ

# Road Map to Scouters

4

**3.10/10.10**

Team Formation

**21.10/28.10**

Project Proposal Report

**30.10/7.11**

Data collection

**30.10/7.11**

Literature Review

**7.11/14.11**

Data Analysis



# Road Map to Scouters

5

14.11/21.11

Mid-Report

17.11/08.12

Development of the Model

8.12/13.12

Coding Interfaces

13.12/16.12

Tests for verification

11.12/18.12

Final Report Presentation







# INTRODUCTION







# FIFA

- Fifa is a game series developed by EA Sports, which has released a new game every year since 1994.
- FIFA 2023 has sold more than 10 million copies so far.



# FY23 Q1 Financial Results Expectency

	Twelve Months Ending March 31, 2023			
	GAAP-Based Financial Data*			
	GAAP Guidance Range	Acquisition- related expenses	Change in deferred net revenue (online- enabled games)	Stock-based compensation
(in \$ millions)				
Total net revenue	7,600 to 7,800	-	300	-
Cost of revenue	2,020 to 2,065	(110)	-	(5)
Operating expense	4,200 to 4,315	(140)	-	(595)
Income before provision for income taxes	1,321 to 1,358	250	300	600
Net income	793 to 815			
Number of shares used in computation:				
Diluted shares	284			

\* The mid-point of the range has been used for purposes of presenting the reconciling items





# Decision Environment

- Career mode





# FIFA CAREER MODE

- FIFA has different game modes. Besides the online mode, the most preferred mode is the career mode.







# Career Mode Display

CENTRAL

SQUAD

TRANSFERS

OFFICE 3

SEASON

ADVANCE 🔔 ⚓

➡ JUL 2021  
Deadline AUG 31

Fri	Sat	Sun	Mon	Tue
9	10	11	12	13
⚓	☁	🏆 FRIENDLY	⚡	⚓

NOTIFICATIONS

📧

Chief Executive  
Vestergaard Transfer Offer

STANDINGS

🎆

		PLD	PTS
1	🏆 SL Benfica	2	4
2	🏆 Leicester City	2	4
3	🏆 Atlético Mineiro	2	1
4	🏆 Tigres	2	1

WEEKLY SCHEDULE

📅

DEVELOPMENT

Help players grow to reach their full potential

👤 ⬆

EA SPORTS

NEWS

Manager Out to Impress In Pre-Season Debut.

⏪ (R) ⏩



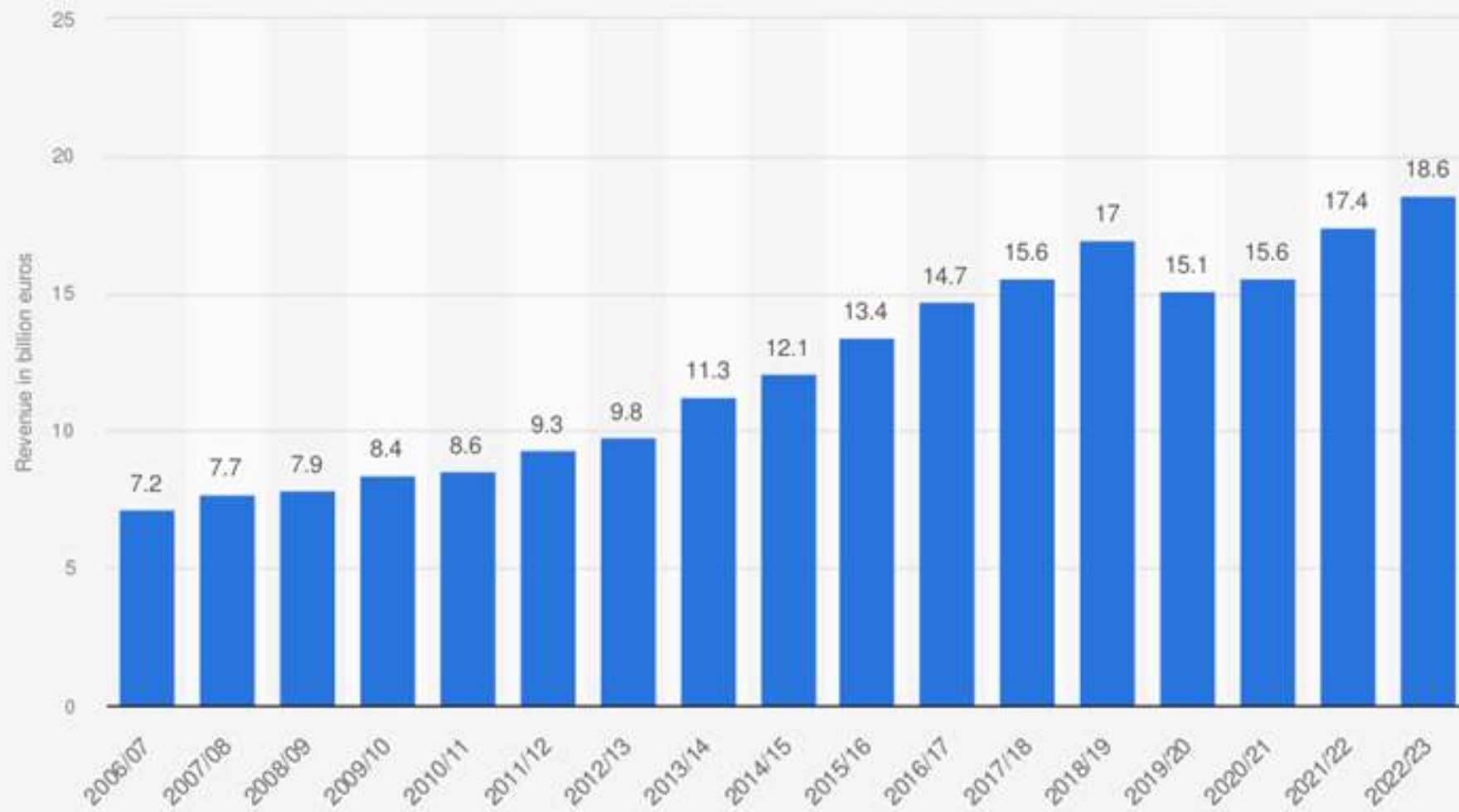


# Significance of Problem

- FIFA has been increasing the number of active players for 28 years and continues to break both sales and playing time records every year.
- In real life, football has become an entertainment industry that left the title of weekend fun time for people and became one of the pioneers of entertainment sector.



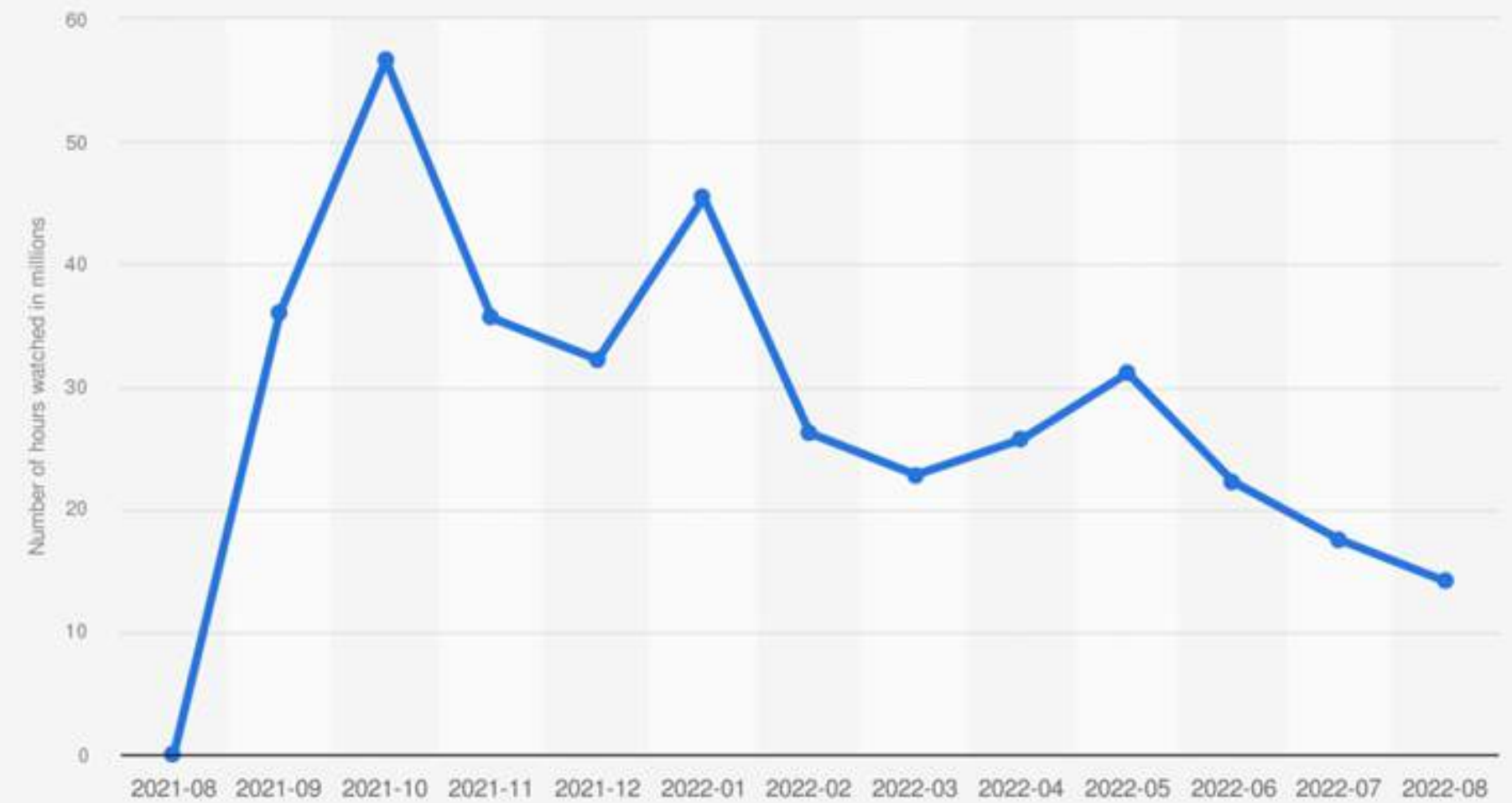
Revenue of the Big Five soccer leagues in Europe from 2006/07 to 2020/21, with a forecast to 2022/23 (in billion euros)



Source  
Deloitte  
© Statista 2022

Additional Information:  
Europe; Deloitte; 2006/07 to 2020/21

Number of hours watched of FIFA 22 on Twitch worldwide from August 2021 to August 2022 (in millions)



Source  
TwitchTracker  
© Statista 2022

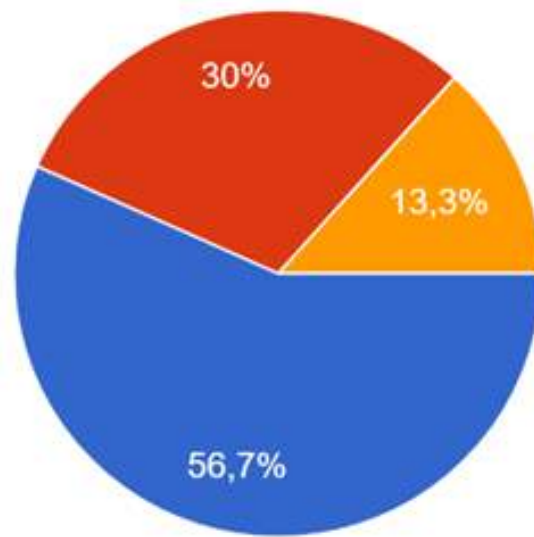
Additional Information:  
Worldwide; August 2021 to August 2022



# Online Surveys

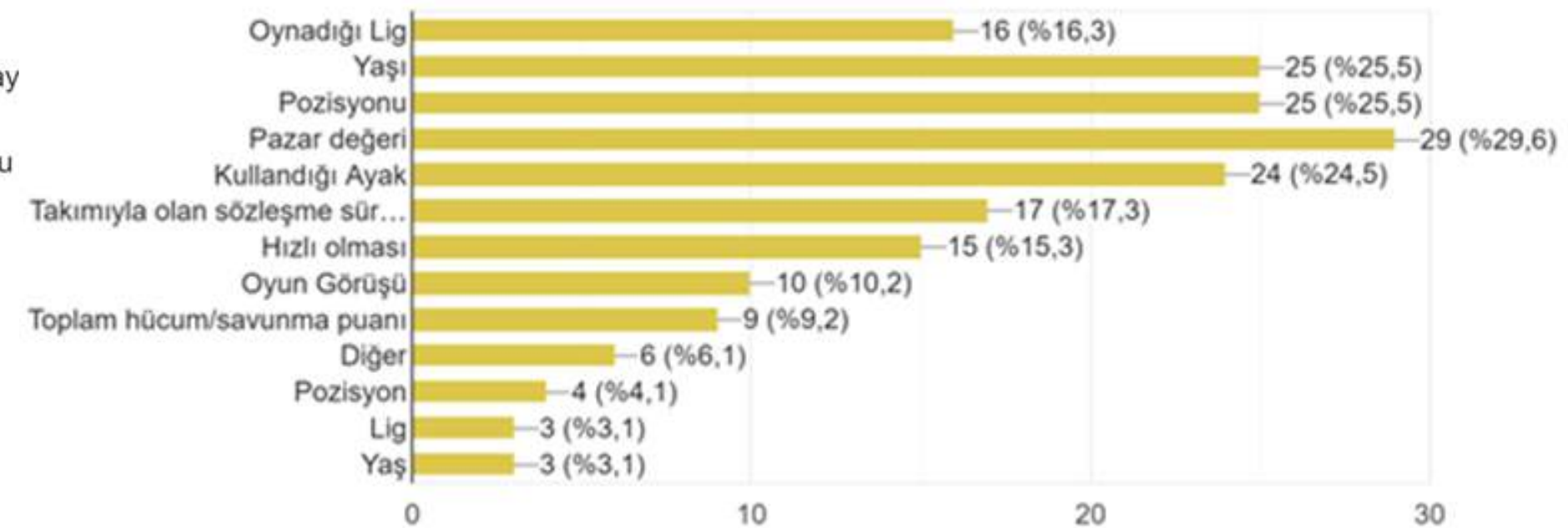
We sent survey to our 90 friends who play FIFA 21 & FIFA 22 & FIFA 23 career mode, after we saw their answers to our survey we understood that this is not just our problem but everyone has a some kind of an issue with finding the right transfer alternatives. As you can see in survey, players want to fill the vacant position with similar skills to the player who played last.

Takımınızdan kilit bir oyuncu ayrıldığında  
90 yanıt



- Yerine aynı profilde bir oyuncuyu almay tercih ederim
- Yeni bir oyun planına uygun bir oyuncu alırım
- Altyapıdan oyuncu yerleştiririm

Transfer yaparken seçtiğiniz oyuncuların hangi özellikleri sizin için öncelikli  
98 yanıt







# Interview 1

**Q:** We are working on an app that will recommend football players for FIFA career mode players. Would you find such an application useful?

**A:** Sometimes it can be difficult to find your dream player. Such an application can be useful if it is easy to use.

**Q:** Any tips you can suggest? What are your biggest problems during the transfer period?

**A:** When a player wants to leave while I have a steady squad during the transfer period, I have to sell him. If this player is a key player, there is a gap in the team that needs to be filled. A recommendation system where I can find players with a profile similar to the player who left my team and fill the gap in my team would be very useful for me.



Ex-Pro - Eren Kara





# Interview 2

**Q:** As I understand it, for example, when the manager tells you that he wants a player like Messi and for some reason it is not possible to transfer Messi to the team, you should suggest a player similar to Messi to the manager. What features do you think provide this similarity? Which parameters should we use?

**A:** It's a question that needs some thought. First of all, of course, the position has to fit. Afterwards, the similarity of player value is also a critical issue in my opinion.

...

The foot preferred by the player is also very important in determining the profile. For example, if a right winger is right-footed, he will usually become a sprinter, but if he is left-footed, he will be a more technical and passing player. This makes the preferred foot an important criterion. Apart from that, it is very important that which team the player we will transfer is currently playing in order to fit our team.



Scout - Muzaffer Çaylı





# Aim of Project

- According to the results of our surveys, a player playing in career mode makes 5 transfers on average in the summer transfer period, while this figure is around 2 in the winter transfer period. While there are an average of 7 transfers per year, there is no player search option other than player filtering in the game. Our aim with this project is to develop an algorithm that can suggest the most suitable players to the gamer by using Machine Learning and to enable FIFA gamers to use this algorithm.







# Scope of Project

- Our FIFA 23 player recommendation system, Scouters, will recommend players suitable for FIFA gamers' teams they manage in career mode. Our system will ask the player to enter a player that matches the profile they have in mind. It will then present to the manager a list of the players in the system who are most similar in profile and characteristics to the player in the manager's mind.





# Methodology

- To create our FIFA player recommendation system, we will first analyze for column selection by Principal Component Analysis (PCA) and distributed stochastic neighbor placement (t-SNE) methods.
- As SCOUTERS, our primary goal is for FIFA players to reach the most suitable player for their team when transferring. For this, it is important to analyze the columns in our dataset in the best way and to get the most appropriate result.



# INPUT DATA ANALYSIS







```

for x in range(totalPlayer):
    name = inlineTables[x].find('a').text.strip()
    positionBase = inlineTables[x].find_all('tr')
    position = positionBase[1].find('td').text.strip()
    pathYas = '//*[@id="yw1"]/table/tbody/tr[{}]/td[3]'.format(x+1)
    pathNumber = '//*[@id="yw1"]/table/tbody/tr[{}]/td[1]/div'.format(x+1)
    pathMarketValue = '//*[@id="yw1"]/table/tbody/tr[{}]/td[6]'.format(x+1)
    footPath = '//*[@id="yw1"]/table/tbody/tr[{}]/td[6]'.format(x+1)
    age = dom.xpath(pathYas)[0].text
    marketValue = dom.xpath(pathMarketValue)[0].text
    foot = dom2.xpath(pathFoot)[0].text
    PlayersList.append(name)
    AgeList.append(age)
    ValuesList.append(marketValue)
    PositionsList.append(position)
    FootList.append(foot)
bayern= pd.DataFrame({"Players":PlayersList,"Position":PositionsList, "Age":AgeList,"Values":ValuesList,"Foot":FootList

```

```
In [17]: bundesliga = bayern.append([dortmund,leipzig,leverkusen,frankfurt,borussia,wolfsburg,freiburg,hoffenheim,stuttgart,unic
```

```
In [21]: bundesliga['Leauge'] = pd.Series(["Bundesliga" for x in range(len(bundesliga.index))])
```

```
In [23]: bundesliga
```

```
Out[23]:
```

	Players	Position	Age	Values	Foot	Leauge	Contract Year
0	Manuel Neuer	Goalkeeper	36	€12.00m	right	Bundesliga	2
1	Sven Ulreich	Goalkeeper	34	€900k	right	Bundesliga	4
2	Johannes Schenk	Goalkeeper	19	€300k		Bundesliga	1
3	Matthijs de Ligt	Centre-Back	23	€70.00m	right	Bundesliga	4
4	Dayot Upamecano	Centre-Back	24	€50.00m	right	Bundesliga	2
...	...	...	...	...	...	...	...
506	Tarsis Bonga	Right Winger	25	€300k	right	Bundesliga	5
507	Philipp Hofmann	Centre-Forward	29	€2.50m	left	Bundesliga	1
508	Lys Mousset	Centre-Forward	26	€1.50m	right	Bundesliga	2
509	Simon Zoller	Centre-Forward	31	€1.10m	right	Bundesliga	5
510	Silvère Ganvoula	Centre-Forward	26	€750k	right	Bundesliga	3

```
In [24]: bundesliga.to_csv('bundesliga.csv')
```

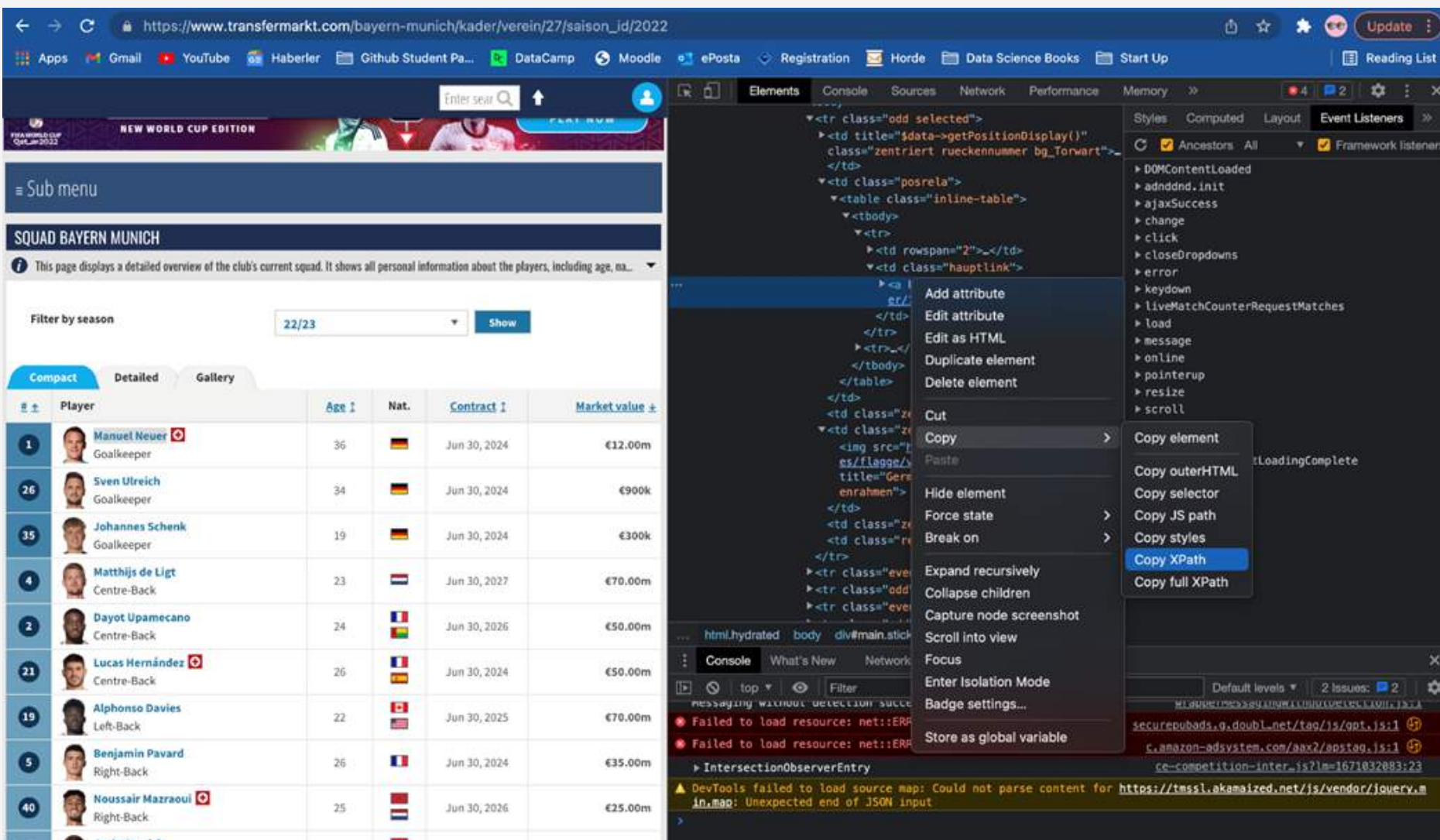


# The Bayern Munich Dataframe

```
##bayern
from bs4 import BeautifulSoup
from lxml import etree
import requests
import pandas as pd
URL = "https://www.transfermarkt.com/bayern-munich/kader/verein/27/saison_id/2022"
HEADERS = ({'User-Agent':
            'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 \
            (KHTML, like Gecko) Chrome/44.0.2403.157 Safari/537.36',\
            'Accept-Language': 'en-US, en;q=0.5'})
webpage = requests.get(URL, headers=HEADERS)
soup = BeautifulSoup(webpage.content, "html.parser")
dom = etree.HTML(str(soup))
#Detay sayfasından güçlü ayak çekiliyor.
URL2 = "https://www.transfermarkt.com/bayern-munich/kader/verein/27/saison_id/2022/plus/1"
webpage2 = requests.get(URL2, headers=HEADERS)
soup2 = BeautifulSoup(webpage2.content, "html.parser")
dom2 = etree.HTML(str(soup2))
odds= soup.findAll('tr',attrs={'class':'odd'})
evens= soup.findAll('tr',attrs={'class':'even'})
inlineTables = soup.findAll('table',attrs={'class':'inline-table'})
totalPlayer = len(odds)+len(evens)
PlayersList = []
AgeList=[]
NationList=[]
ValuesList = []
PositionsList=[]
FootList = []
```

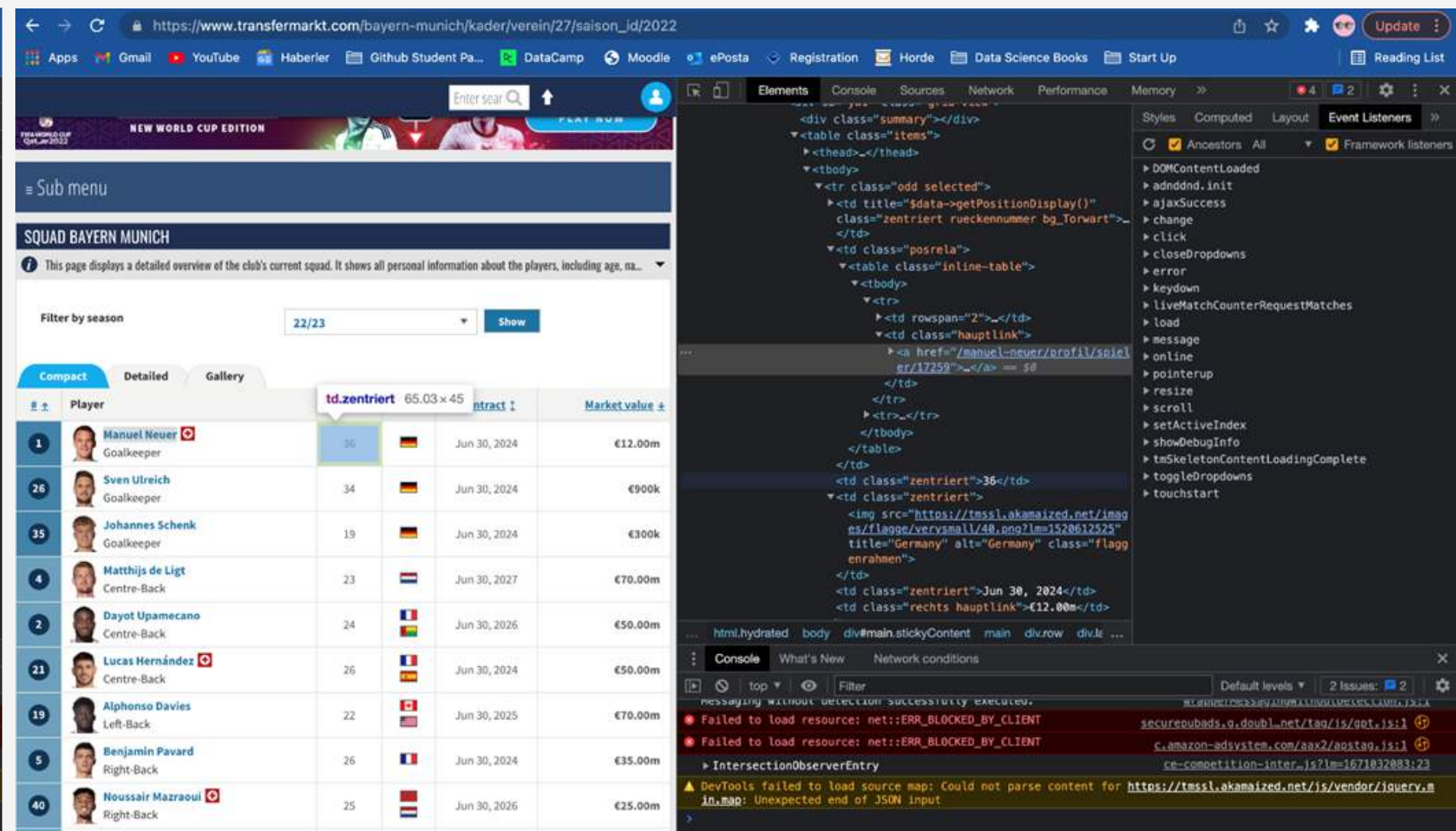


# Web Scrapping



The screenshot shows the Bayern Munich squad page on transfermarkt.com. The DevTools console is open, and the 'Elements' panel shows the HTML structure of the squad table. A context menu is open over a table cell, and the 'Copy XPath' option is highlighted. The console shows several error messages related to failed resource loading.

#	Player	Age	Nat.	Contract	Market value
1	Manuel Neuer	36	Germany	Jun 30, 2024	€12.00m
26	Sven Ulreich	34	Germany	Jun 30, 2024	€900k
35	Johannes Schenk	19	Germany	Jun 30, 2024	€300k
4	Matthijs de Ligt	23	Netherlands	Jun 30, 2027	€70.00m
2	Dayot Upamecano	24	France	Jun 30, 2026	€50.00m
21	Lucas Hernández	26	France	Jun 30, 2024	€50.00m
19	Alphonso Davies	22	Canada	Jun 30, 2025	€70.00m
5	Benjamin Pavard	26	France	Jun 30, 2024	€35.00m
40	Noussair Mazraoui	25	Germany	Jun 30, 2026	€25.00m



This screenshot is identical to the one on the left, showing the Bayern Munich squad page with DevTools open and the 'Copy XPath' option highlighted in the context menu. The console shows several error messages related to failed resource loading.

#	Player	Age	Nat.	Contract	Market value
1	Manuel Neuer	36	Germany	Jun 30, 2024	€12.00m
26	Sven Ulreich	34	Germany	Jun 30, 2024	€900k
35	Johannes Schenk	19	Germany	Jun 30, 2024	€300k
4	Matthijs de Ligt	23	Netherlands	Jun 30, 2027	€70.00m
2	Dayot Upamecano	24	France	Jun 30, 2026	€50.00m
21	Lucas Hernández	26	France	Jun 30, 2024	€50.00m
19	Alphonso Davies	22	Canada	Jun 30, 2025	€70.00m
5	Benjamin Pavard	26	France	Jun 30, 2024	€35.00m
40	Noussair Mazraoui	25	Germany	Jun 30, 2026	€25.00m





# Dataset Cleaning

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

In [2]: allPlayer = pd.concat(
    map(pd.read_csv, ['laliga.csv', 'bundesliga.csv', 'premierLeague.csv', 'league1.csv', 'serieA.csv']), ignore_index=True

In [3]: allPlayer.head(10)

Out[3]:
```

	Unnamed: 0	Players	Position	Age	Values	Foot	Leauge	Contract Year
0	0	Thibaut Courtois	Goalkeeper	30	€60.00m	left	La Liga	5
1	1	Andriy Lunin	Goalkeeper	23	€4.00m	right	La Liga	3
2	2	Éder Militão	Centre-Back	24	€60.00m	right	La Liga	4
3	3	David Alaba	Centre-Back	30	€55.00m	left	La Liga	1
4	4	Antonio Rüdiger	Centre-Back	29	€40.00m	right	La Liga	5
5	5	Nacho Fernández	Centre-Back	32	€5.00m	right	La Liga	2
6	6	Jesús Vallejo	Centre-Back	25	€2.50m	right	La Liga	5
7	7	Ferland Mendy	Left-Back	27	€40.00m	left	La Liga	4
8	8	Daniel Carvajal	Right-Back	30	€18.00m	right	La Liga	1
9	9	Álvaro Odriozola	Right-Back	27	€6.00m	right	La Liga	2

```
In [4]: allPlayer = allPlayer.drop_duplicates()

In [5]: allPlayer= allPlayer.drop(columns= [ 'Unnamed: 0'])

In [11]: allPlayer.rename(columns = {'Players':'Player'}, inplace = True)
```

```
In [12]: allPlayer

Out[12]:
```

	Player	Position	Age	Values	Foot	Leauge	Contract Year
0	Thibaut Courtois	Goalkeeper	30	€60.00m	left	La Liga	5
1	Andriy Lunin	Goalkeeper	23	€4.00m	right	La Liga	3
2	Éder Militão	Centre-Back	24	€60.00m	right	La Liga	4
3	David Alaba	Centre-Back	30	€55.00m	left	La Liga	1
4	Antonio Rüdiger	Centre-Back	29	€40.00m	right	La Liga	5
...	...	...	...	...	...	...	...
2540	Mehdi Lérís	Right Winger	24	€2.20m	right	Serie A	3
2541	Manolo Gabbiadini	Centre-Forward	31	€3.00m	left	Serie A	3
2542	Manuel De Luca	Centre-Forward	24	€2.50m	both	Serie A	2
2543	Francesco Caputo	Centre-Forward	35	€1.50m	right	Serie A	1
2544	Fabio Quagliarella	Centre-Forward	39	€1.00m	right	Serie A	5

2545 rows x 7 columns

```
In [13]: allPlayer.to_csv('allPlayer.csv')
```

```
In [ ]:
```

# Filtering the Goalkeepers

```
In [11]: df_filtering
```

```
Out[11]:
```

	index	Player	Age	Values	Foot	Contract Year	Nation	Pos	Squad	Comp	...	8- Crs	8- Int	8- TklW	8- PKwon	8- PKcon	8- OG	8- Recov	8- Won	8- Lost	8- Won%
0	2	Éder Militão	24	€60.00m	right	4	BRA	DF	Real Madrid	La Liga	...	2	12	10	0.0	1.0	0	58.0	21.0	17.0	55.3
1	3	David Alaba	30	€55.00m	left	1	AUT	DF	Real Madrid	La Liga	...	16	11	5	0.0	0.0	0	61.0	4.0	3.0	57.1
2	4	Antonio Rüdiger	29	€40.00m	right	5	GER	DF	Real Madrid	La Liga	...	2	4	3	0.0	0.0	0	49.0	12.0	4.0	75.0
3	6	Ferland Mendy	27	€40.00m	left	4	FRA	DF	Real Madrid	La Liga	...	6	10	6	0.0	0.0	0	47.0	6.0	4.0	60.0
4	7	Aurélien Tchouaméni	22	€80.00m	right	2	FRA	MF	Real Madrid	La Liga	...	5	21	15	0.0	0.0	0	54.0	21.0	11.0	65.6
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1411	2072	Valerio Verre	28	€1.80m	both	2	ITA	MF	Sampdoria	Serie A	...	20	8	5	0.0	0.0	0	42.0	7.0	8.0	46.7
1412	2074	Mehdi Lérís	24	€2.20m	right	3	ALG	MFDF	Sampdoria	Serie A	...	25	9	23	0.0	0.0	0	52.0	22.0	13.0	62.9
1413	2075	Manolo Gabbiadini	31	€3.00m	left	3	ITA	MFFW	Sampdoria	Serie A	...	13	1	0	0.0	0.0	0	14.0	15.0	23.0	39.5
1414	2077	Francesco Caputo	35	€1.50m	right	1	ITA	FW	Sampdoria	Serie A	...	3	1	1	0.0	0.0	0	18.0	6.0	21.0	22.2
1415	2078	Fabio Quagliarella	39	€1.00m	right	5	ITA	FW	Sampdoria	Serie A	...	7	0	1	0.0	0.0	0	8.0	5.0	16.0	23.8

1416 rows x 153 columns

```
In [13]: df_filtering = df_last[df_last['90s']>=3]
```

```
In [14]: df_gk = df_filtering[df_filtering['Pos'] == 'GK'].reset_index()
```

```
In [15]: df_gk = df_gk.fillna(0)
```

```
In [16]: gk = [ ]
for idx in range(len(df_gk)):
    gk.append(df_gk['Player'][idx] + '({})'.format(df_gk['Squad'][idx]))
gk_ID = dict(zip(gk, np.arange(len(gk))))
```

```
In [18]: df_gk
```

```
Out[18]:
```

	index	Player	Age	Values	Foot	Contract Year	Nation	Pos	Squad	Comp	...	8- Crs	8- Int	8- TklW	8- PKwon	8- PKcon	8- OG	8- Recov	8- Won	8- Lost	8- Won%
0	0	Thibaut Courtois	30	€60.00m	left	5	BEL	GK	Real Madrid	La Liga	...	0	0	0	0.0	0.0	0	15.0	2.0	0.0	100.0
1	1	Andriy Lunin	23	€4.00m	right	3	UKR	GK	Real Madrid	La Liga	...	0	0	0	0.0	0.0	0	4.0	0.0	0.0	0.0
2	20	Marc-André ter Stegen	30	€30.00m	right	5	GER	GK	Barcelona	La Liga	...	0	1	0	0.0	0.0	0	22.0	3.0	0.0	100.0
3	43	Jan Oblak	29	€40.00m	right	5	SVN	GK	Atlético Madrid	La Liga	...	0	0	0	0.0	0.0	0	12.0	1.0	0.0	100.0
4	63	Álex Remiro	27	€25.00m	right	2	ESP	GK	Real Sociedad	La Liga	...	0	0	0	0.0	0.0	0	17.0	4.0	0.0	100.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
111	1962	Michele Di Gregorio	25	€4.50m	right	5	ITA	GK	Monza	Serie A	...	0	3	0	0.0	0.0	0	13.0	2.0	1.0	66.7
112	1990	Luigi Sepe	31	€1.50m	left	4	ITA	GK	Salernitana	Serie A	...	0	0	0	0.0	0.0	0	24.0	3.0	0.0	100.0
113	2013	Guglielmo Vicario	26	€12.00m	right	2	ITA	GK	Empoli	Serie A	...	0	0	0	0.0	0.0	0	14.0	5.0	1.0	83.3
114	2038	Marco Carnesecchi	22	€8.00m	right	4	ITA	GK	Cremonese	Serie A	...	0	0	0	0.0	0.0	0	2.0	0.0	0.0	0.0
115	2061	Emil Audero	25	€6.00m	right	1	ITA	GK	Sampdoria	Serie A	...	0	0	0	0.0	0.0	1	20.0	3.0	0.0	100.0

116 rows x 153 columns





## Checking for Null Value

# allPlayer

```
In [2]: import pandas as pd
import numpy as np
```

```
In [3]: fbref = pd.read_csv('fbreflast.csv')
```

```
In [4]: transfer = pd.read_csv('transferlast.csv')
```

```
In [7]: ## We see that There samle columns with same value in both table. So we decide to drop Age in Transfer and League
transfer = transfer.drop(columns=['Leauge', 'Position'])
fbref = fbref.drop(columns='Age')
```

```
In [8]: df = pd.merge(transfer, fbref, on='Player', how='inner')
```

```
In [12]: df = df_filtering.columns
df
```

```
Out[12]: Index(['index', 'Player', 'Age', 'Values', 'Foot', 'Contract Year', 'Nation',
               'Pos', 'Squad', 'Comp',
               ...
               '8-Crs', '8-Int', '8-TklW', '8-PKwon', '8-PKcon', '8-OG', '8-Recov',
               '8-Won', '8-Lost', '8-Won%'],
              dtype='object', length=153)
```

```
In [12]: df_filtering = df_filtering.fillna(0)
```

```
In [13]: df_filtering.isnull().sum()
```

```
Out[13]: index      0
Player      0
Age          0
Values      0
Foot        0
            ..
8-OG         0
8-Recov      0
8-Won        0
8-Lost       0
8-Won%       0
Length: 153, dtype: int64
```





# Extracting Data From Fbref.com

Big 5 European Leagues History

2022-2023 Big 5 European Leagues Overview

Player Stats ▲

Squad Stats ▼

Nationalities

Squad & Player Wages

Player Stats

[Standard Stats](#) [Goalkeeping](#) [Advanced Goalkeeping](#) [Shooting](#) [Passing](#) [Pass Types](#) [Goal and Shot Creation](#) [Defensive Actions](#) [Possession](#) [Playing Time](#) [Miscellaneous Stats](#)

League Summary

+ Premier League

Most Goals:

[Erling Haaland](#) (Manchester City) - 18

Most Assists:

[Kevin De Bruyne](#) (Manchester City) - 9

Most Clean Sheets:

[Aaron Ramsdale](#) (Arsenal),  
[Nick Pope](#) (Newcastle United) - 7

🇮🇹 Serie A

Most Goals:

[Victor Osimhen](#) (Napoli) - 9

Most Assists:

[Sergej Milinković-Savić](#) (Lazio) - 7

Most Clean Sheets:

[Ivan Provedel](#) (Lazio) - 9

🇪🇸 La Liga

Most Goals:

[Robert Lewandowski](#) (Barcelona) - 13

Most Assists:

[Mikel Merino](#) (Real Sociedad) - 6

Most Clean Sheets:

[Marc-André ter Stegen](#) (Barcelona) - 11

🇫🇷 Ligue 1

Most Goals:

[Kylian Mbappé](#) (Paris Saint-Germain) - 12

Most Assists:

[Lionel Messi](#) (Paris Saint-Germain) - 10

Most Clean Sheets:

[Gianluigi Donnarumma](#) (Paris Saint-Germain) - 9

🇩🇪 Bundesliga

Most Goals:

[Christopher Nkunku](#) (RB Leipzig) - 12

Most Assists:

[Randal Kolo Muani](#) (Eintracht Frankfurt) - 9

Most Clean Sheets:

[Mark Flekken](#) (Freiburg) - 8

Apps

Gmail

YouTube

Haberler

Github Student Pa...

DataCamp

Moodle

ePosta

Registration

Horde

Data Science Books

Start Up

Reading List

welcome, football fans from around the world! Use FBref in: [Portuguese](#) • [English](#) • [Français](#) • [Italiano](#) • [Dansk](#) • [Español](#)

Sports Reference®

Baseball

Football (college)

Basketball (college)

Hockey

Soccer

Blog

Stathead®

Questions or Comments?

Login

Enter Person, Team, Section, etc.

Search

Players

Clubs

Competitions

Countries

Matches <sup>47</sup>

Languages

Mailing List

Full Site Menu Below ▾

Announcing a Major Coverage Expansion for 19 Competitions.[Get the full details of our biggest update yet on our blog.](#) Sign up for our World Cup mailing list [here](#).

2022-2023 Big 5 European Leagues Stats

[« Previous Season](#)

Big 5 European Leagues History

2022-2023 Big 5 European Leagues Overview

Player Stats ▼

Squad Stats ▼

Nationalities

Squad & Player Wages

On this page:

[League Summary](#) [Big 5 Table](#) [Leaders](#) [Nationalities](#) [Full Site Menu](#)

League Summary

+ Premier League

Most Goals:

[Erling Haaland](#) (Manchester City) - 18

Most Assists:

[Kevin De Bruyne](#) (Manchester City) - 9

🇮🇹 Serie A

Most Goals:

[Victor Osimhen](#) (Napoli) - 9

Most Assists:

[Sergej Milinković-Savić](#) (Lazio) - 7

Most Clean Sheets:

[Ivan Provedel](#) (Lazio) - 9

🇪🇸 La Liga

Most Goals:

[Robert Lewandowski](#) (Barcelona) - 13

Most Assists:

[Mikel Merino](#) (Real Sociedad) - 6

Most Clean Sheets:

[Marc-André ter Stegen](#)

🇫🇷 Ligue 1

Most Goals:

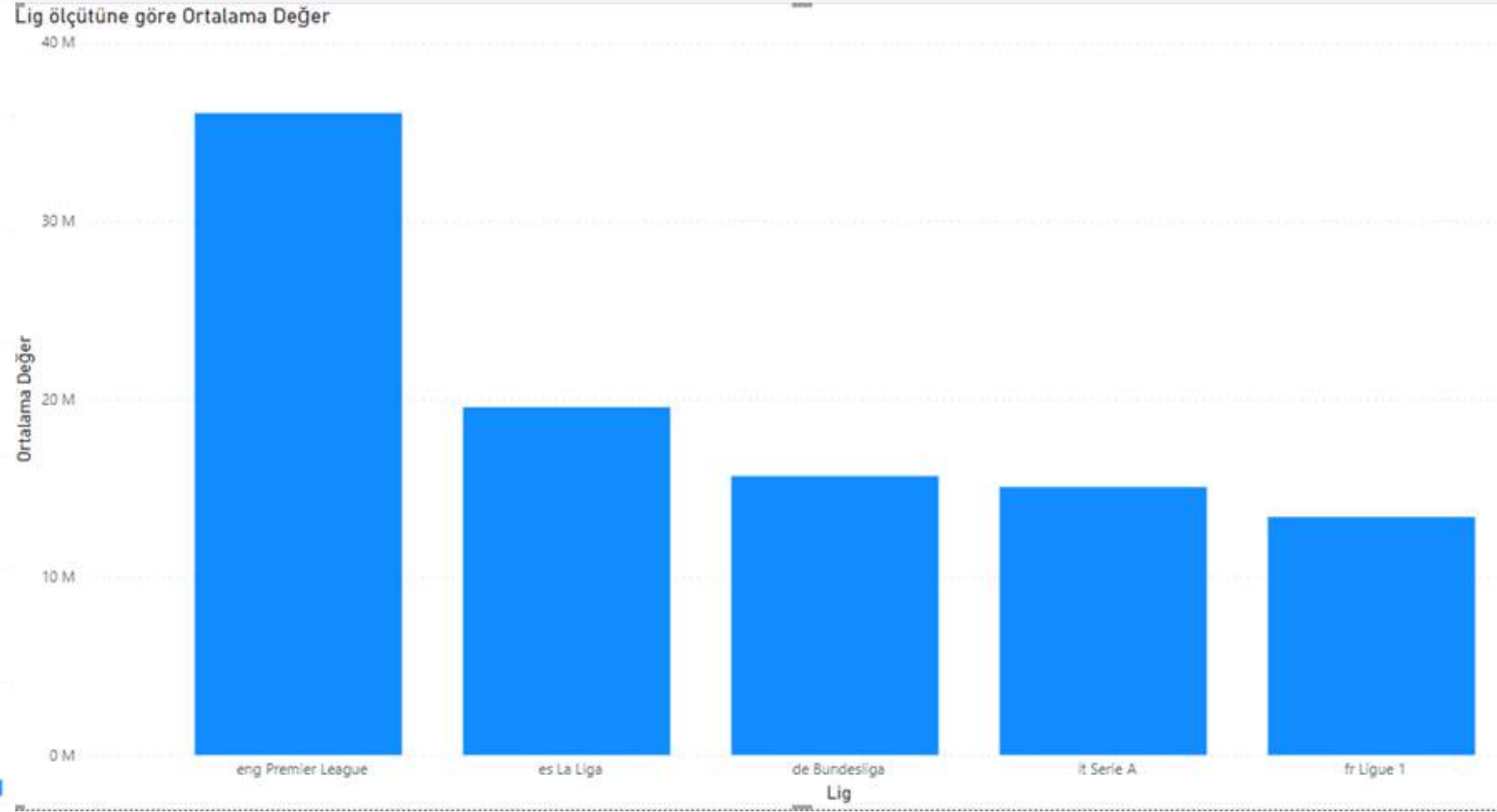
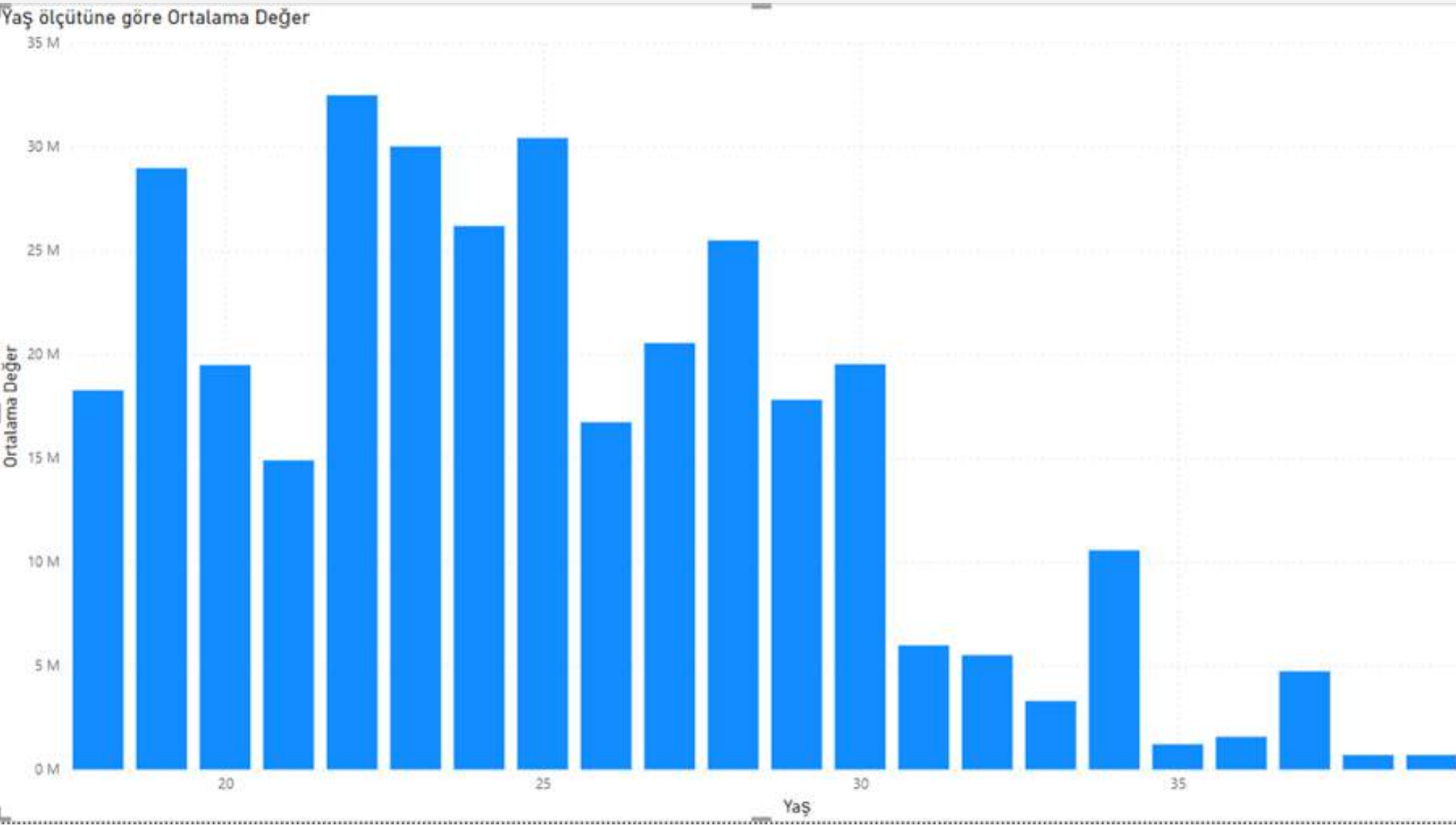
[Kylian Mbappé](#) (Paris Saint-Germain) - 12

Most Assists:

[Lionel Messi](#) (Paris Saint-Germain) - 10

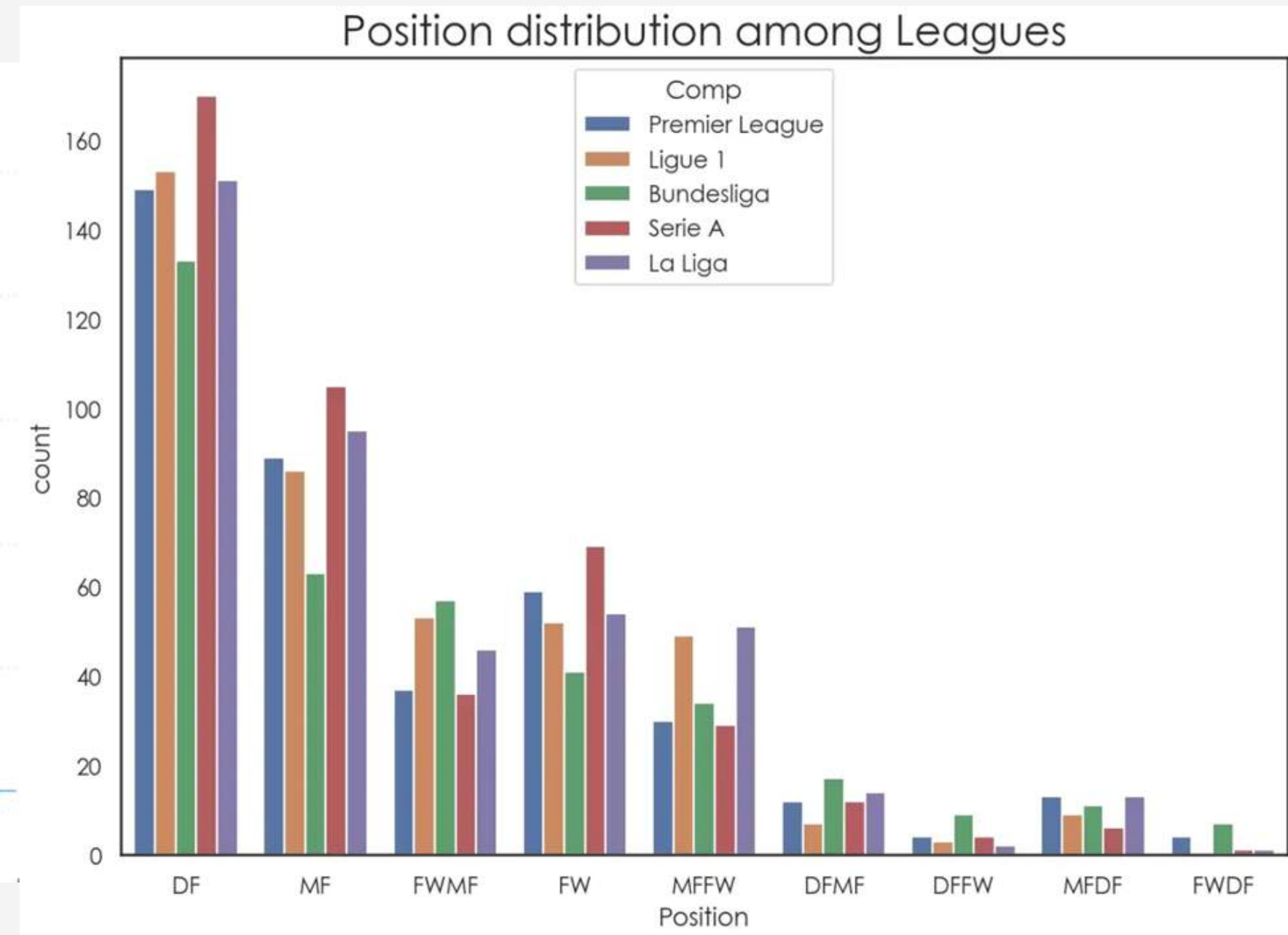
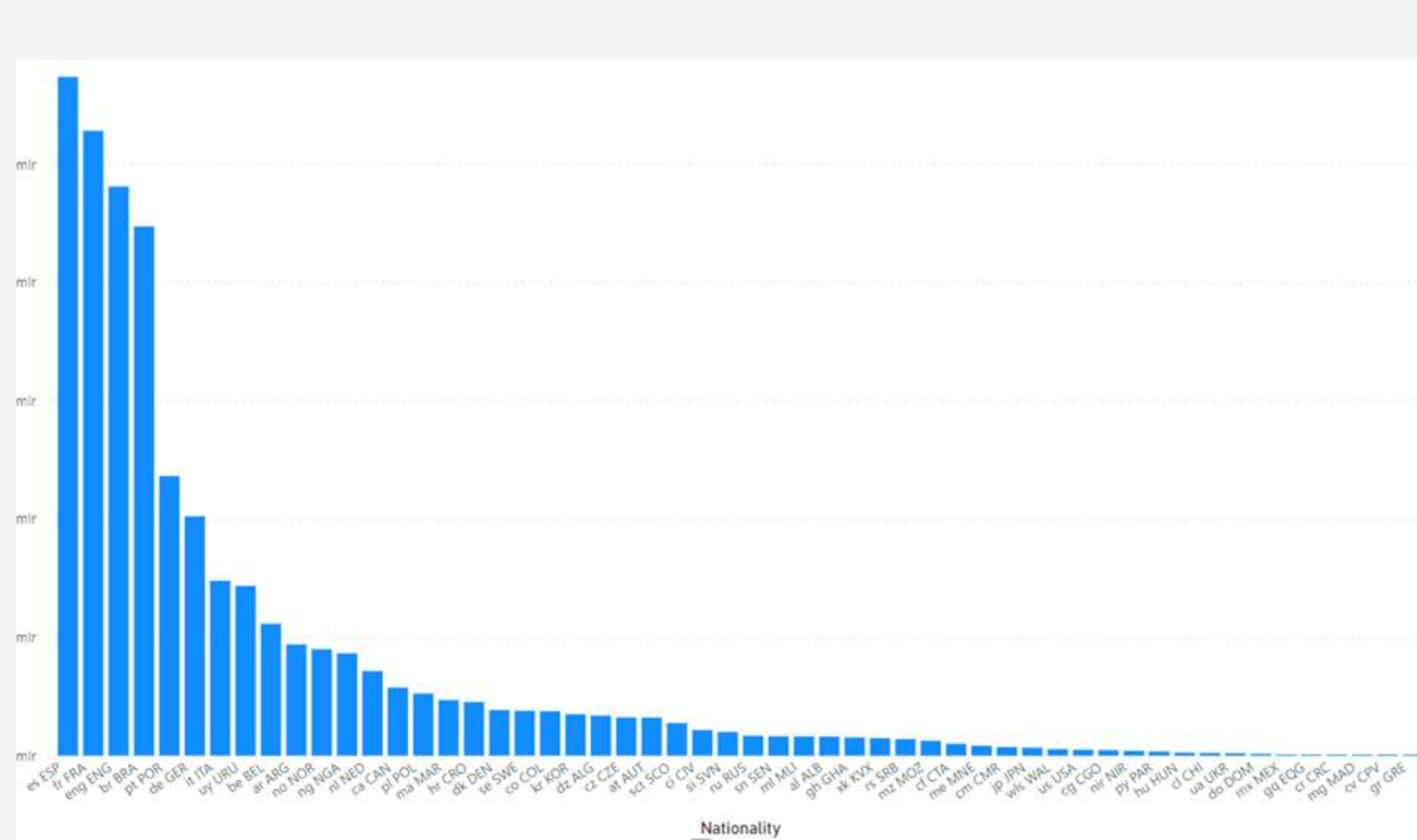


# Data Visualization & Analysis





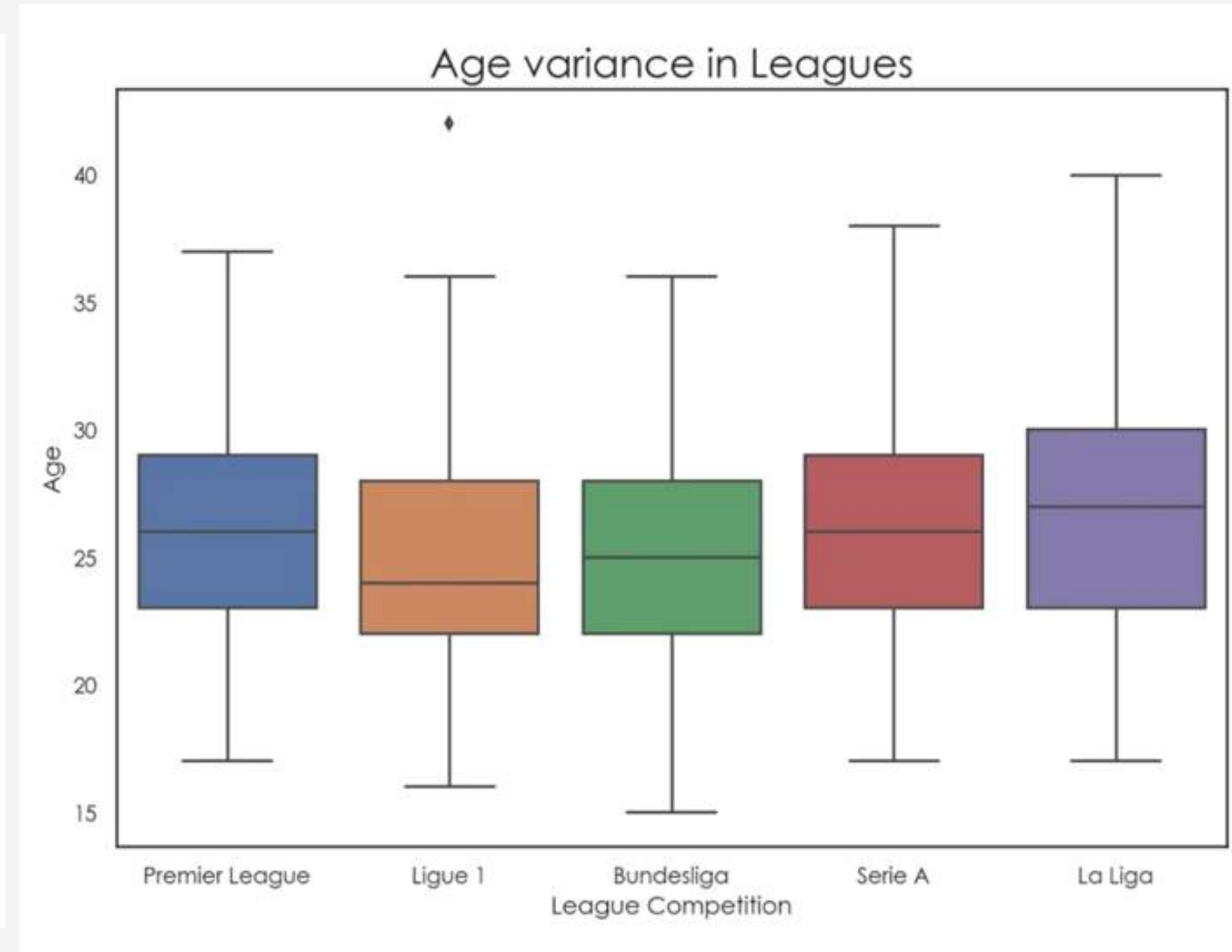
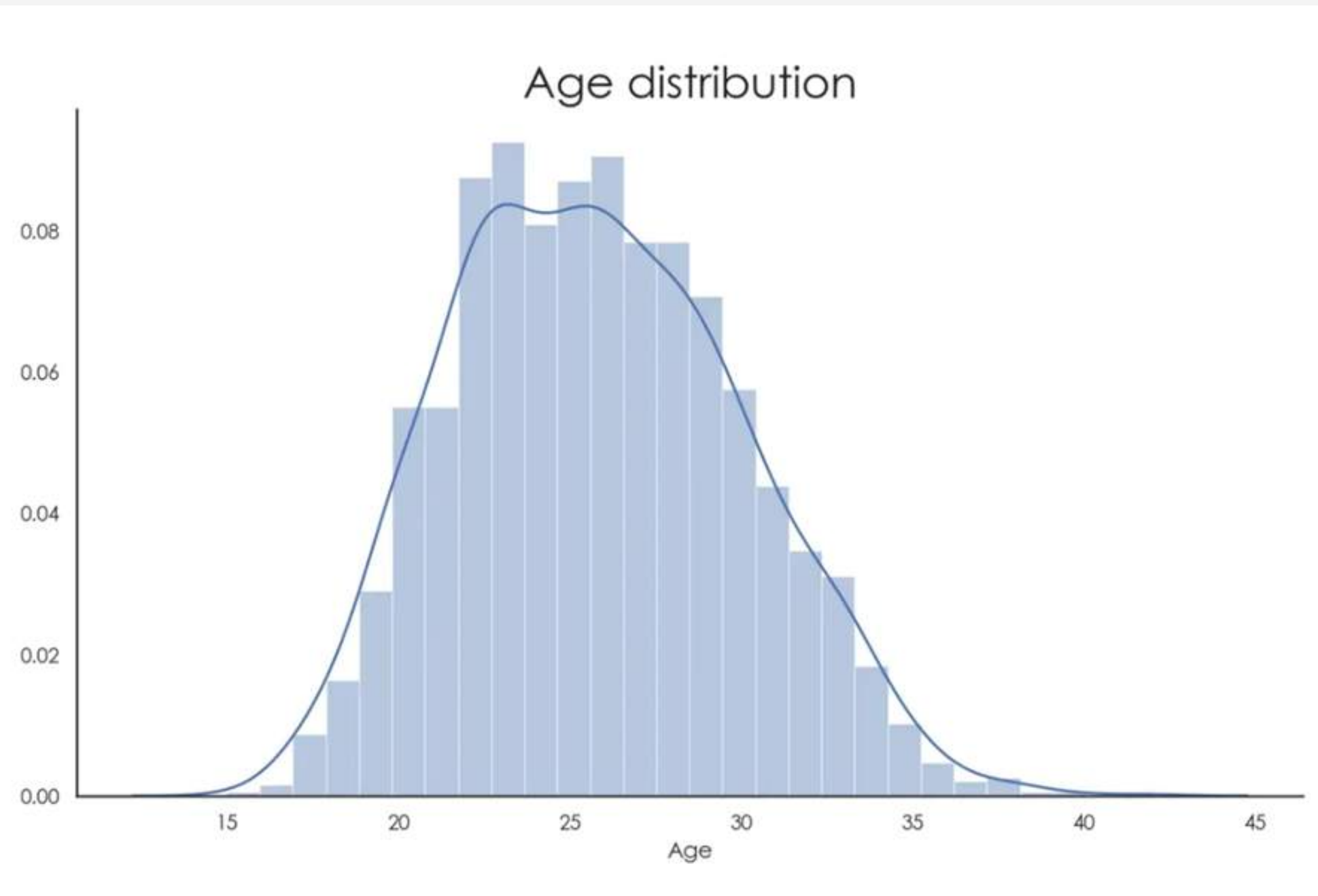
# Data Visualization & Analysis







# Data Visualization & Analysis





# DEVELOPMENT OF DECISION MODEL







## t-SNE, PCA

```
from sklearn import decomposition
from sklearn.preprocessing import StandardScaler

# standardizing the data
data = StandardScaler().fit_transform(stats)

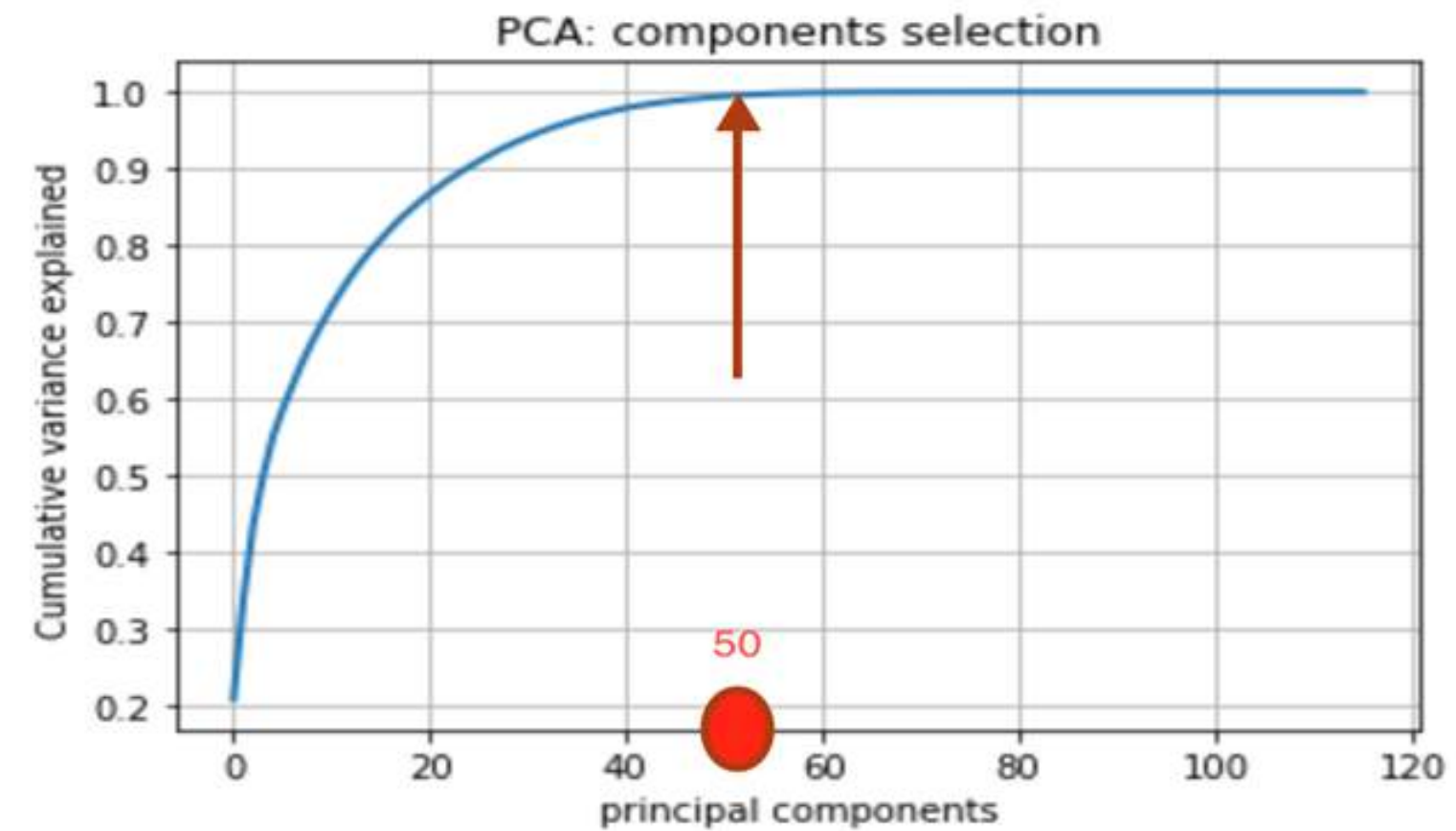
# setting up and running PCA
pca = decomposition.PCA()
pca.n_components = 116
pca_data = pca.fit_transform(data)

# % variance explained per components
percentage_var_explained = pca.explained_variance_ / np.sum(pca.explained_variance_);

# cumulative variance explained
cum_var_explained = np.cumsum(ppercentage_var_explained)

plt.figure(1, figsize=(6, 4))
plt.plot(cum_var_explained, linewidth=2)
plt.axis('tight')
plt.grid()
plt.xlabel('principal components')
plt.ylabel('Cumulative variance explained')
plt.title('PCA: components selection')
```

Out[21]: Text(0.5, 1.0, 'PCA: components selection')





# t-SNE, PCA

```
from scipy.spatial import distance
import numpy as np
import pickle
from tqdm import tqdm

# fetch the player vector
def getStats(name):
    idx = gk_ID[name]
    return stats[idx, :]

# fetch cosine similarity between two player vectors
def similarity(gk1, gk2):
    return 1 - distance.cosine(getStats(gk1), getStats(gk2))

# normalize an array to a scale of 0 to 100
def normalize(array):
    return np.array([round(num, 2) for num in (array - min(array))*100/(max(array)-min(array))])

# player-similarities hash table
engine = {}
for query in tqdm(gk):
    metric = []
    for player in gk:
        value = similarity(query, player)
        metric.append(value)
    metric = normalize(metric)
    # adding normalized similarity values to the 'player name' key
    engine[query] = metric

# saving hash table/dict as a pickle
with open(r'data\gk_engine.pickle', 'wb') as file:
    pickle.dump(engine, file)
```

100%|██████████| 116/116 [00:00<00:00, 305.47it/s]

```
from sklearn.preprocessing import StandardScaler
from sklearn.manifold import TSNE
import seaborn as sns
import matplotlib as plt
import matplotlib.pyplot as plt
stats = df_gk.select_dtypes(include=['int64', 'float64'])
```

In [23]: stats = pca\_data[:, :80]

```
In [24]: from scipy.spatial import distance
import numpy as np
import pickle
from tqdm import tqdm
# fetch the player vector
def getStats(name):
    idx = player_ID[name]
    return stats[idx, :]

# fetch cosine similarity between two player vectors
def similarity(player1, player2):
    return 1 - distance.cosine(getStats(player1), getStats(player2))

# normalize an array to a scale of 0 to 100
def normalize(array):
    return np.array([round(num, 2) for num in (array - min(array))*100/(max(array)-min(array))])

# player-similarities hash table
engine = {}
for query in tqdm(players):
    metric = []
    for player in players:
        value = similarity(query, player)
        metric.append(value)
    metric = normalize(metric)
    # adding normalized similarity values to the 'player name' key
    engine[query] = metric

# saving hash table/dict as a pickle
with open(r'playengine.pickle', 'wb') as file:
    pickle.dump(engine, file)
```

100%|██████████| 1416/1416 [00:55<00:00, 25.54it/s]





## t-SNE, PCA

```
In [22]: from sklearn import decomposition
from sklearn.preprocessing import StandardScaler

# standardizing the data
data = StandardScaler().fit_transform(stats)

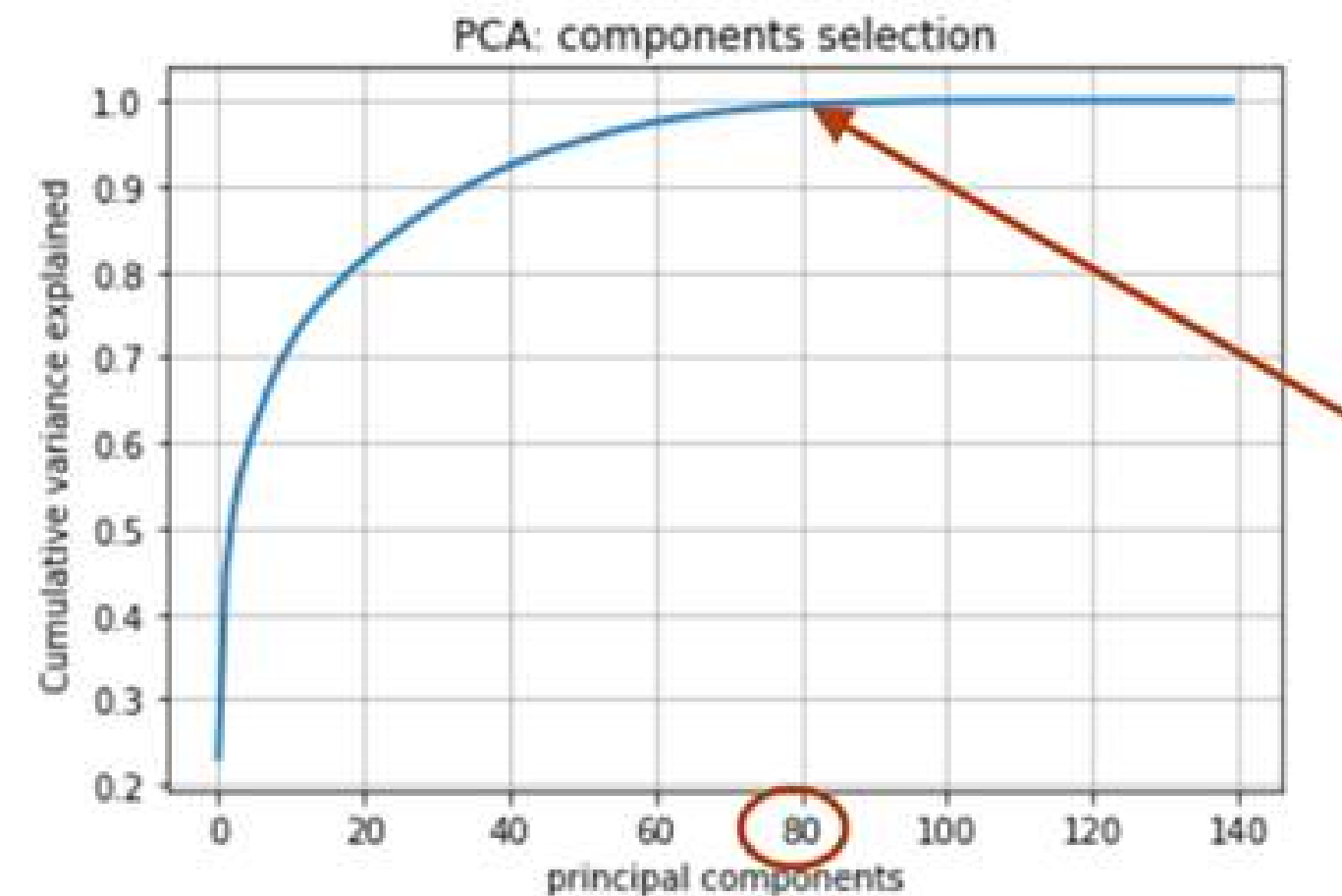
# setting up and running PCA
pca = decomposition.PCA()
pca.n_components = 140
pca_data = pca.fit_transform(data)

# % variance explained per components
percentage_var_explained = pca.explained_variance_ / np.sum(pca.explained_variance_);

# cumulative variance explained
cum_var_explained = np.cumsum(ppercentage_var_explained)

plt.figure(1, figsize=(6, 4))
plt.plot(cum_var_explained, linewidth=2)
plt.axis('tight')
plt.grid()
plt.xlabel('principal components')
plt.ylabel('Cumulative variance explained')
plt.title('PCA: components selection')
```

```
Out[22]: Text(0.5, 1.0, 'PCA: components selection')
```





## t-SNE, PCA

```
In [21]: from sklearn.preprocessing import StandardScaler
from sklearn.manifold import TSNE
import seaborn as sns
import matplotlib as plt
import matplotlib.pyplot as plt

# selecting only numerical metrics
stats = df_filtering.select_dtypes(include=['int64', 'float64'])
# position types
labels = df_filtering['Pos']

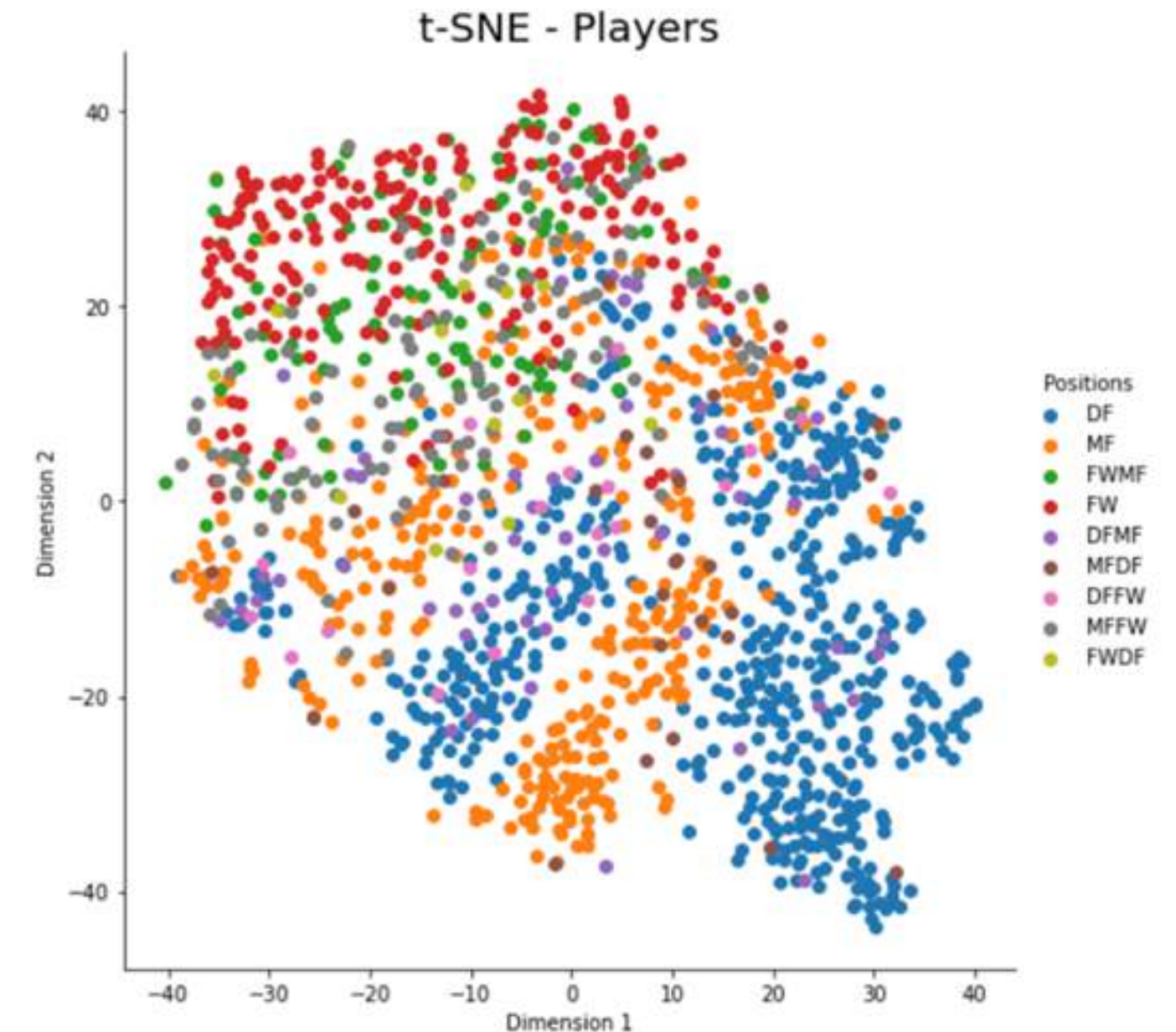
# standardizing the data
data = StandardScaler().fit_transform(stats)

# configuring tSNE params
model = TSNE(n_components=2, perplexity=30, random_state=0)
tsne_data = model.fit_transform(data)

# creating a new df to plot the result data
tsne_data = np.vstack((tsne_data.T, labels)).T
tsne_df = pd.DataFrame(data=tsne_data, columns=("Dimension 1", "Dimension 2", "Positions"))

# plotting the result of tSNE
ax = sns.FacetGrid(tsne_df, hue="Positions", size=7)\
    .map(plt.scatter, 'Dimension 1', 'Dimension 2').add_legend()
plt.title('t-SNE - Players', size=20)
```

Out[21]: Text(0.5, 1.0, 't-SNE - Players')







# DEVELOPMENT OF THE DSS



# **FIFA 23**

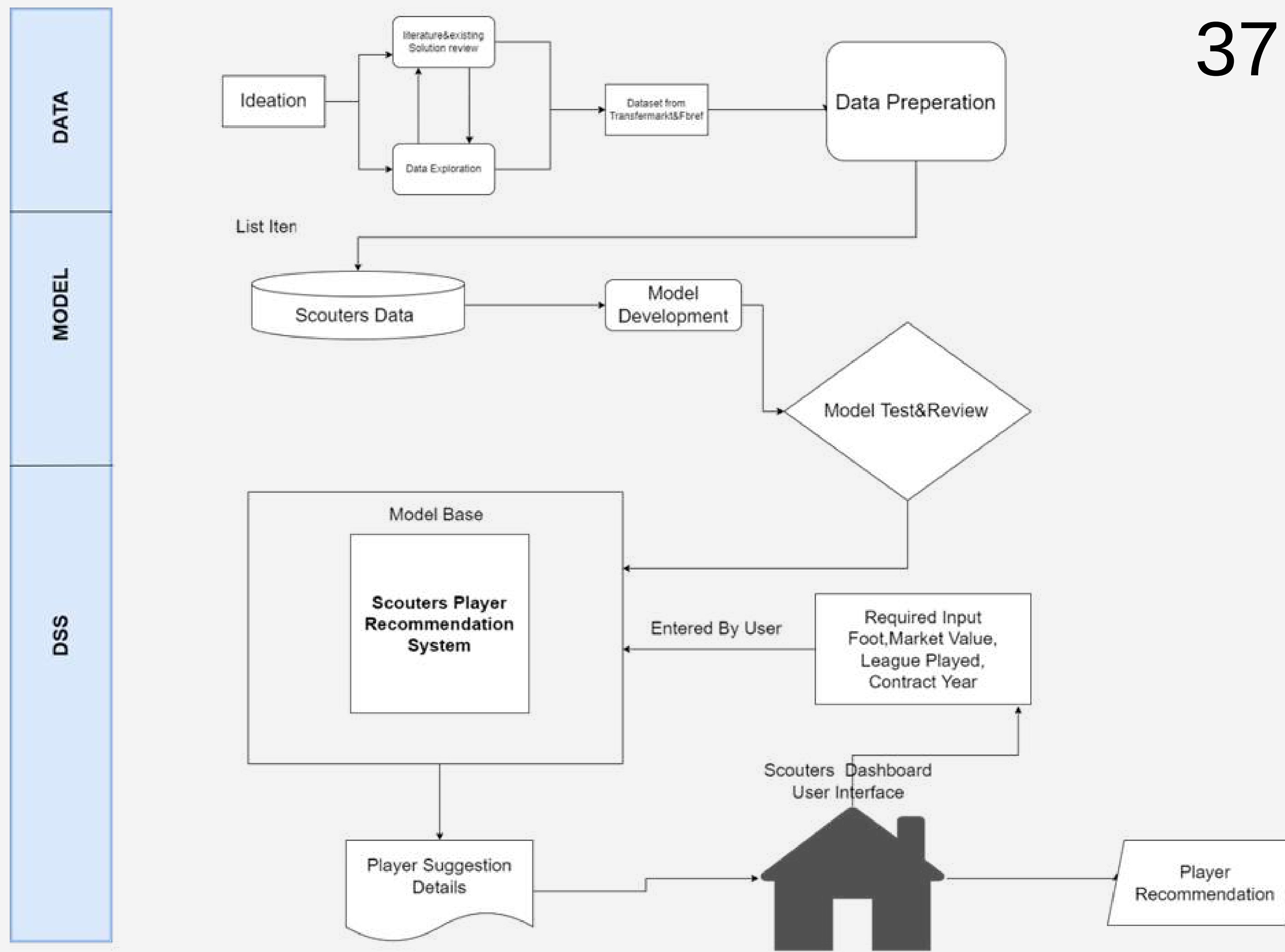


## **LEGACY EDITION**





# Architecture of DSS





# Datasets of the DSS

38

## Players

```
In [11]: df_filtering
```

```
Out[11]:
```

	index	Player	Age	Values	Foot	Contract Year	Nation	Pos	Squad	Comp	...	8-Crs	8-Int	8-TklW	8-PKwon	8-PKcon	8-OG	8-Recov	8-Won	8-Lost	8-Won%
0	2	Éder Militão	24	€80.00m	right	4	BRA	DF	Real Madrid	La Liga	...	2	12	10	0.0	1.0	0	58.0	21.0	17.0	55.3
1	3	David Alaba	30	€55.00m	left	1	AUT	DF	Real Madrid	La Liga	...	16	11	5	0.0	0.0	0	61.0	4.0	3.0	57.1
2	4	Antonio Rüdiger	29	€40.00m	right	5	GER	DF	Real Madrid	La Liga	...	2	4	3	0.0	0.0	0	49.0	12.0	4.0	75.0
3	6	Ferland Mendy	27	€40.00m	left	4	FRA	DF	Real Madrid	La Liga	...	6	10	6	0.0	0.0	0	47.0	6.0	4.0	60.0
4	7	Aurélien Tchouaméni	22	€80.00m	right	2	FRA	MF	Real Madrid	La Liga	...	5	21	15	0.0	0.0	0	54.0	21.0	11.0	65.6
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1411	2072	Valerio Verze	28	€1.80m	both	2	ITA	MF	Sampdoria	Serie A	...	20	8	5	0.0	0.0	0	42.0	7.0	8.0	46.7
1412	2074	Mehdi Lérís	24	€2.20m	right	3	ALG	MFDF	Sampdoria	Serie A	...	25	9	23	0.0	0.0	0	52.0	22.0	13.0	62.9
1413	2075	Manolo Gabbiadini	31	€3.00m	left	3	ITA	MFFW	Sampdoria	Serie A	...	13	1	0	0.0	0.0	0	14.0	15.0	23.0	39.5
1414	2077	Francesco Caputo	35	€1.50m	right	1	ITA	FW	Sampdoria	Serie A	...	3	1	1	0.0	0.0	0	18.0	6.0	21.0	22.2
1415	2078	Fabio Quagliarella	39	€1.00m	right	5	ITA	FW	Sampdoria	Serie A	...	7	0	1	0.0	0.0	0	8.0	5.0	16.0	23.8

1416 rows x 153 columns

## Goalkeepers

```
In [18]: df_gk
```

```
Out[18]:
```

	index	Player	Age	Values	Foot	Contract Year	Nation	Pos	Squad	Comp	...	8-Crs	8-Int	8-TklW	8-PKwon	8-PKcon	8-OG	8-Recov	8-Won	8-Lost	8-Won%
0	0	Thibaut Courtois	30	€80.00m	left	5	BEL	GK	Real Madrid	La Liga	...	0	0	0	0.0	0.0	0	15.0	2.0	0.0	100.0
1	1	Andriy Lunin	23	€4.00m	right	3	UKR	GK	Real Madrid	La Liga	...	0	0	0	0.0	0.0	0	4.0	0.0	0.0	0.0
2	20	Marc-André ter Stegen	30	€30.00m	right	5	GER	GK	Barcelona	La Liga	...	0	1	0	0.0	0.0	0	22.0	3.0	0.0	100.0
3	43	Jan Oblak	29	€40.00m	right	5	SVN	GK	Atlético Madrid	La Liga	...	0	0	0	0.0	0.0	0	12.0	1.0	0.0	100.0
4	63	Alex Remiro	27	€25.00m	right	2	ESP	GK	Real Sociedad	La Liga	...	0	0	0	0.0	0.0	0	17.0	4.0	0.0	100.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
111	1962	Michela Di Gregorio	25	€4.50m	right	5	ITA	GK	Monza	Serie A	...	0	3	0	0.0	0.0	0	13.0	2.0	1.0	66.7
112	1990	Luigi Sepe	31	€1.50m	left	4	ITA	GK	Salermitana	Serie A	...	0	0	0	0.0	0.0	0	24.0	3.0	0.0	100.0
113	2013	Guglielmo Vicario	26	€12.00m	right	2	ITA	GK	Empoli	Serie A	...	0	0	0	0.0	0.0	0	14.0	5.0	1.0	83.3
114	2038	Marco Carnesecchi	22	€8.00m	right	4	ITA	GK	Cremonese	Serie A	...	0	0	0	0.0	0.0	0	2.0	0.0	0.0	0.0
115	2061	Emil Audero	25	€6.00m	right	1	ITA	GK	Sampdoria	Serie A	...	0	0	0	0.0	0.0	1	20.0	3.0	0.0	100.0

116 rows x 153 columns



## Home

## db

Home.py — scouts

```
Home.py x
Home.py > read_info
1  import pandas as pd
2  import streamlit as st
3  import pickle
4  from pathlib import Path
5  import streamlit_authenticator as stauth
6  import yaml
7  from PIL import Image
8
9  st.set_page_config(page_title="Home Page", page_icon=":soccer:", layout="wide")
10
11  @st.cache
12  def read_info(path):
13      |   return Path(path).read_text(encoding='utf8')
14
15
16  st.title("Best Scouters find Best Players ")
17  image1 = Image.open('loginn.jpeg')
18  st.image(image1,channels="RGB", output_format="auto")
19  st.markdown(read_info('scouters.md'), unsafe_allow_html=True)
20  st.sidebar.header("Menu")
21  image = Image.open('scouterr.jpeg')
22  image2= Image.open('sid.jpeg')
23  st.sidebar.image(image2,channels="RGB", output_format="auto")
24  st.sidebar.image(image,channels="RGB", output_format="auto",width=336)
```

```
# DB Management
import sqlite3
conn = sqlite3.connect('data.db')
c = conn.cursor()
# DB Functions
def create_usertable():
    c.execute('CREATE TABLE IF NOT EXISTS userstable(username TEXT,password TEXT)')

def add_userdata(username,password):
    c.execute('INSERT INTO userstable(username,password) VALUES (?,?)',(username,password))
    conn.commit()

def login_user(username,password):
    c.execute('SELECT * FROM userstable WHERE username =? AND password = ?',(username,password))
    data = c.fetchall()
    return data

def view_all_users():
    c.execute('SELECT * FROM userstable')
    data = c.fetchall()
    return data
```

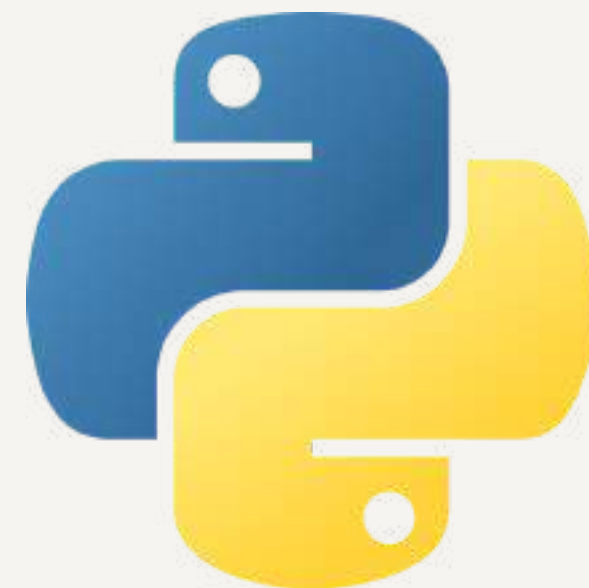




# Technical Features



Streamlit



seaborn

matplotlib



# USER INTERFACES

