INTRODUCTION TO DATA SCIENCE

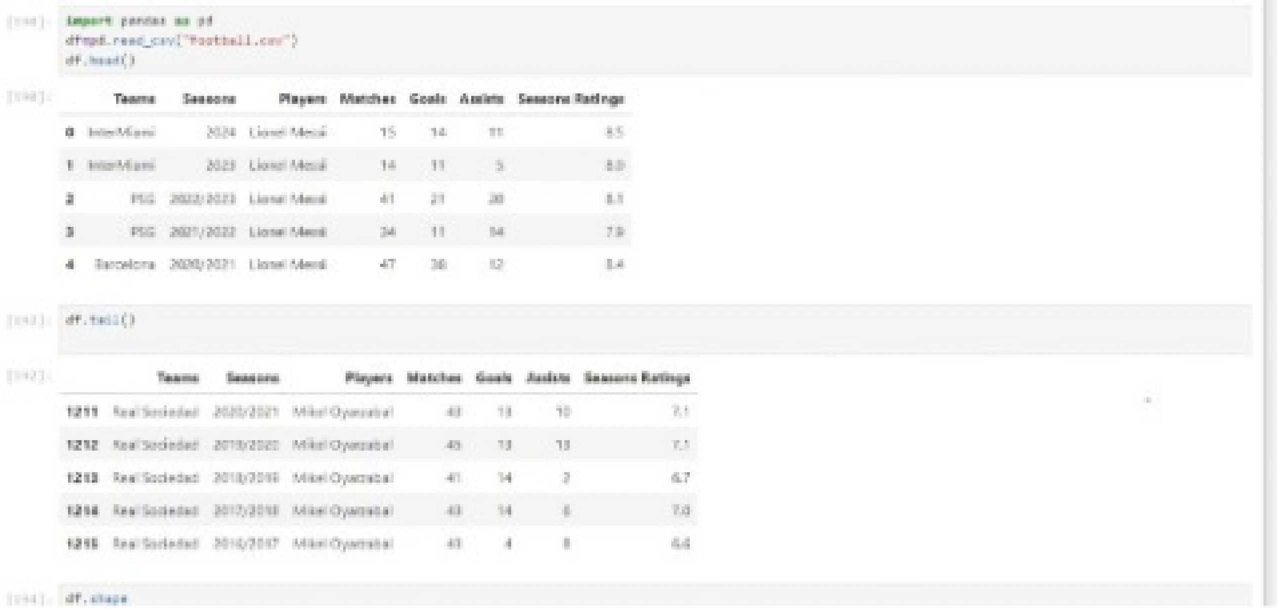
FOOTBALL PLAYERS 2015-2024

project overview

Title: Football Player Data Cleaning and Analysis (2015-2024

Objective: Clean and preprocess football player data to enable insightful analysis and visualization.

let's read the data head and tail values



```python
import pandas as pd
df=pd.read_csv("football.csv")
df.head()
```

	Teams	Seasons	Players	Matches	Goals	Assists	Seasons Ratings
0	InterMiami	2024	Lionel Messi	15	14	11	8.5
1	InterMiami	2023	Lionel Messi	14	11	5	8.0
2	PSG	2022/2023	Lionel Messi	41	21	20	8.1
3	PSG	2021/2022	Lionel Messi	34	11	14	7.9
4	Barcelona	2020/2021	Lionel Messi	47	38	12	8.4

```python
df.tail()
```

	Teams	Seasons	Players	Matches	Goals	Assists	Seasons Ratings
1211	Real Sociedad	2020/2021	Mikel Oyarzabal	43	13	10	7.1
1212	Real Sociedad	2019/2020	Mikel Oyarzabal	45	13	13	7.1
1213	Real Sociedad	2018/2019	Mikel Oyarzabal	41	14	2	6.7
1214	Real Sociedad	2017/2018	Mikel Oyarzabal	43	14	6	7.0
1215	Real Sociedad	2016/2017	Mikel Oyarzabal	43	4	8	6.6

```python
df.shape
```

memory usage of the dataset

how to reduce memory size?

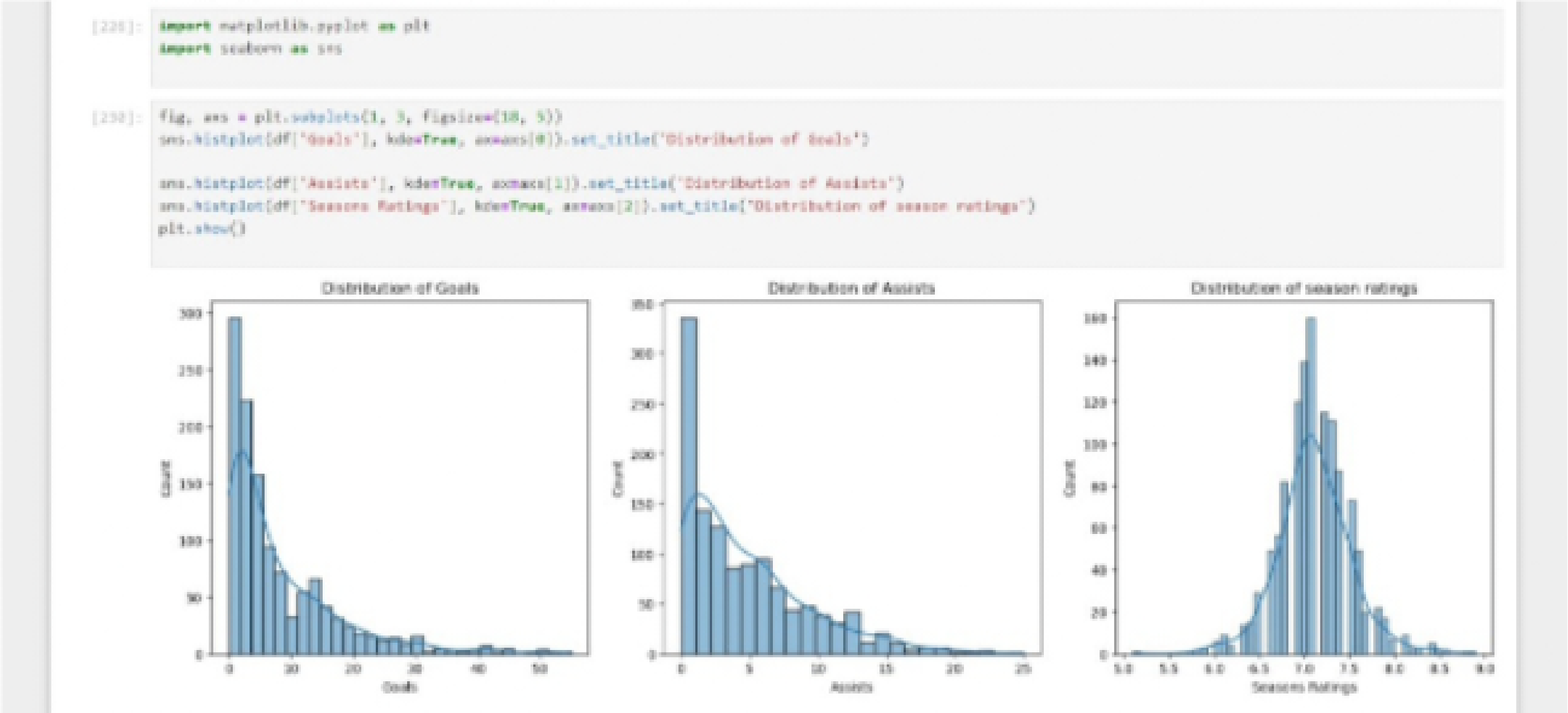
```
[220]: import pandas as pd
       df=pd.read_csv("football.csv",usecols=req_cols)
       df.info(memory_usage="deep")
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1216 entries, 0 to 1215
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Teams            1216 non-null   object
 1   Seasons          1216 non-null   object
 2   Players          1216 non-null   object
 3   Matches          1216 non-null   int64
 4   Goals            1216 non-null   int64
 5   Assists          1216 non-null   int64
 6   Seasons Ratings  1216 non-null   float64
dtypes: float64(1), int64(3), object(3)
memory usage: 249.2 KB
```

after reducing the memory size

```
[224]: df.info(memory_usage="deep")

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1216 entries, 0 to 1215
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Teams            1216 non-null   object
 1   Seasons          1216 non-null   object
 2   Players          1216 non-null   object
 3   Matches          1216 non-null   int32
 4   Goals            1216 non-null   int32
 5   Assists          1216 non-null   int32
 6   Seasons Ratings  1216 non-null   float32
dtypes: float32(1), int32(3), object(3)
memory usage: 230.2 KB
```

histogram



import matplotlib.pyplot as plt
import seaborn as sns
plt.show()
Distribution of Goals
Distribution of Assists
Distribution of season ratings
Count
Goals
Assists
Seasons Ratings

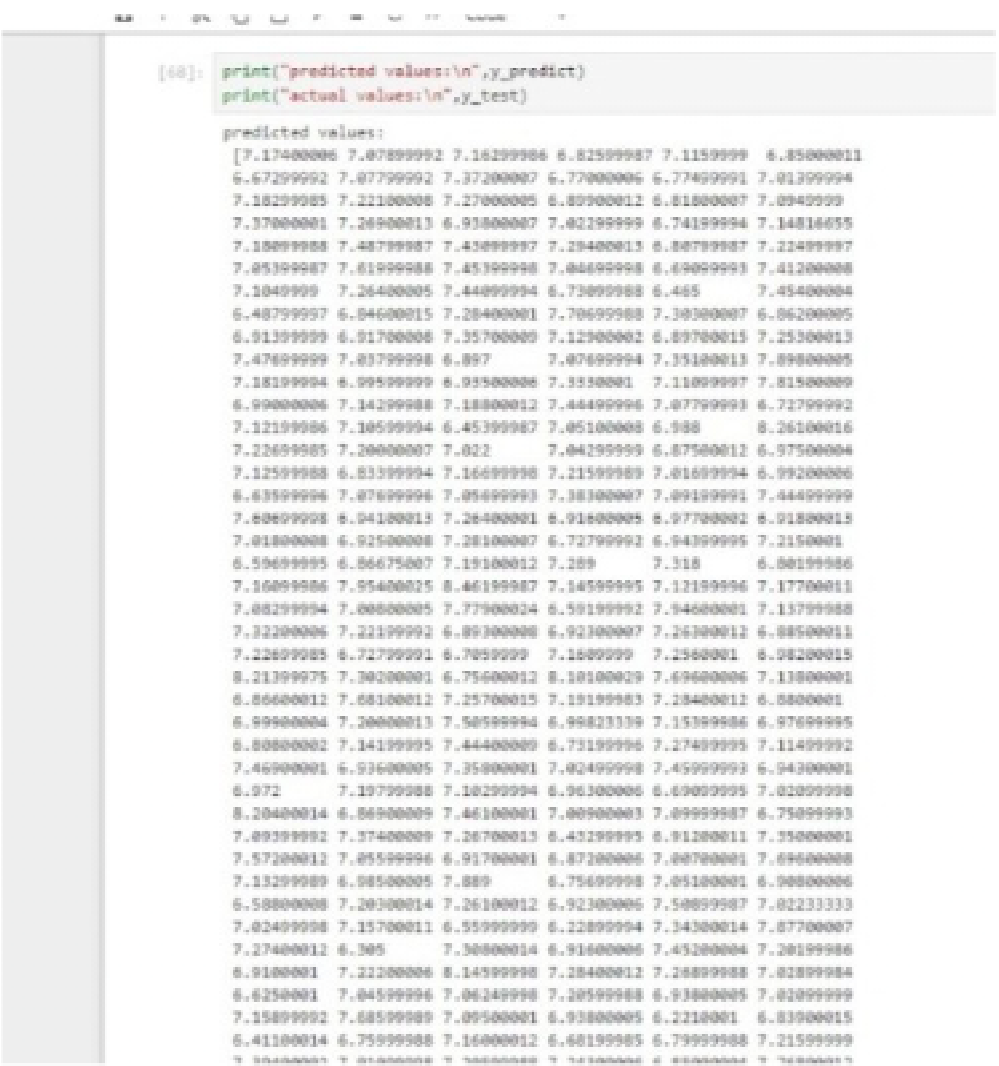
accuracy and prediction

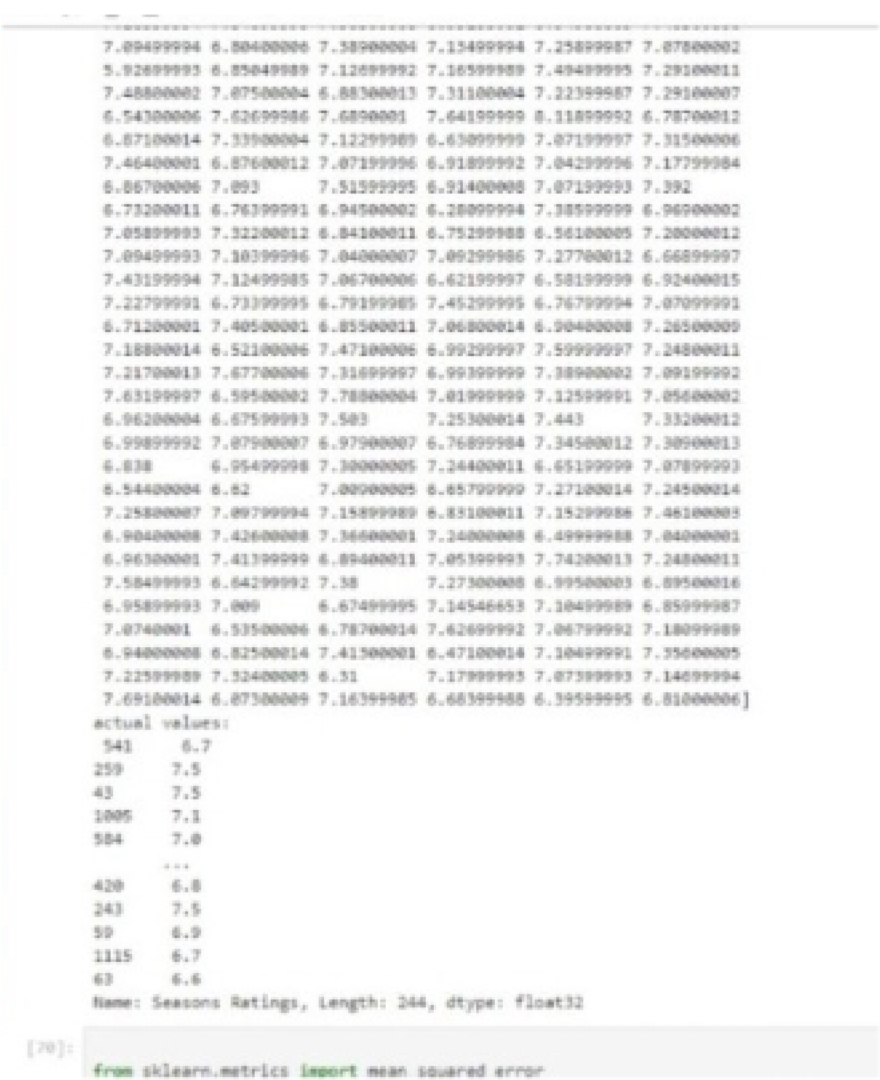
```
[31]: df=df.drop(columns=['Seasons'])

[  ]: X = df.drop(columns=['Seasons Ratings'])
      y = df['Seasons Ratings']

[37]: from sklearn.model_selection import train_test_split
      X_train,X_test,y_train,y_test = train_test_split(X,y, random_state=42,test_size=0.2)
      X_train,y_train

[37]: (                 Teams           Players  Matches  Goals  Assists
       432             Seville      Jules Kounde       44      3        1
       277             Chelsea        Kai Havertz      47     14        5
       721            Man City     Manuel Akanji       48      4        0
       991     Atletico Madrid     Thomas Partey       24      1        4
       678     Atletico Madrid  Rodrigo De Paul        48      4        1
       ...                 ...               ...      ...    ...      ...
       1044          Wolfsburg   Victor Osimhen        13      0        0
       1095           Juventus     Adrien Rabiot       45      0        2
       1150           Brighton        Lewis Dunk       59      1        1
       860   Linares Deportivo      Fermin Lopez       40     12        0
       1126           Brighton        Lewis Dunk       31      1        0

       [972 rows x 5 columns],
       432     7.3
       277     7.1
       721     7.2
       991     6.7
       678     7.1
              ...
       1044    5.8
```

```
[282]: y_predict=model.predict(X_train)

[283]: model.score(X_train,y_train)*100

[283]: 92.51097108806892
```

accuracy is 92%

prediction


print("predicted values:\n",y_predict)
print("actual values:\n",y_test)
predicted values:


actual values:
541 6.7
259 7.5
43 7.5
1005 7.1
584 7.0
...
420 6.8
243 7.5
59 6.9
1115 6.7
63 6.6
Name: Seasons Ratings, Length: 244, dtype: float32
from sklearn.metrics import mean_squared_error

time of the dataset

[384]:

```
%%time
import pandas as pd
df=pd.read_csv("football.csv")
df
```

```
CPU times: total: 31.2 ms
Wall time: 12.1 ms
```

[384]:

	Teams	Seasons	Players	Matches	Goals	Assists	Seasons Ratings
0	InterMiami	2024	Lionel Messi	15	14	11	8.5
1	InterMiami	2023	Lionel Messi	14	11	5	8.0
2	PSG	2022/2023	Lionel Messi	41	21	20	8.1
3	PSG	2021/2022	Lionel Messi	34	11	14	7.9
4	Barcelona	2020/2021	Lionel Messi	47	38	12	8.4
...	...	...	...	...	...	...	...
1211	Real Sociedad	2020/2021	Mikel Oyarzabal	43	13	10	7.1
1212	Real Sociedad	2019/2020	Mikel Oyarzabal	45	13	13	7.1
1213	Real Sociedad	2018/2019	Mikel Oyarzabal	41	14	2	6.7
1214	Real Sociedad	2017/2018	Mikel Oyarzabal	43	14	6	7.0
1215	Real Sociedad	2016/2017	Mikel Oyarzabal	43	4	8	6.6

after reducing time

[88]:

```
%%time
import pandas as pd
df=pd.read_csv("football.csv",usecols=req_cols,iterator=True,chunksize=90)
df
```

```
CPU times: total: 0 ns
Wall time: 5.56 ms
```

outliers

```
[232]: plt.figure(figsize=(10, 6))
       sns.scatterplot(data=df, x='Goals', y='Seasons Ratings', hue='Seasons')
       plt.title('Goals vs Seasons Ratings')
       plt.xlabel('Goals')
       plt.ylabel('Seasons Ratings')
       plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
       plt.show()
```

THANK YOU

team-04
K.Harshitha(23091A3243)
M.Lakshmidevi(23091A3265)
BT.Bhavesh(23091A3221)
Nandini(FDH)