Prediction of FIFA World Cup and Wages of FIFA Players.

Author –

- **Riya Dwarka Lachuriya**
- **Srivalli Mandadi**
- **Haseeb Mohammed**
- **Zeel Nimesh Patel**

Abstract

Predicting Market Value of football players:
This project explored different Machine Learning (ML) techniques to predict and study the market value of professional football players based on their characteristics and football attributes and compare it with their actual transfer value, we explored several model design ideas and evaluated their performances against benchmark techniques.

In this project, we built a predictive model using machine learning regression algorithms to predict the market value of professional football players. The performance of the model compares favorably to other ML algorithms model and traditional techniques.

The predictive model was also used to identify under and overvalued players relative to the market. From these conclusions, we can conclude that the prediction of players market value using machine learning methods can help market values manipulations, as large disparities between actual and predicted results were observed in the project.

Prediction of the winner of an international matches Prediction results are "Win / Lose / Draw" or "goal difference" and in the last we will be predicting the overall winner of FIFA world cup 2022.

Introduction

FIFA (Fédération International de Football Association), meaning International Association Football Federation) Headquartered in Zürich, Switzerland, its membership now comprises 211 national associations. FIFA publishes its results according to International Financial Reporting Standards. A report in London's The Sunday Times in June 2014 said the members of the committee had their salaries doubled from $100,000 to $200,000 during the year, revenue in 2014 was $2096 million. The report also said leaked documents had indicated $4.4 million in secret bonuses had been paid to the committee members following the 2010 FIFA World Cup in South Africa, also in 2010 the revenue of FIFA was $1291 million.

Besides its worldwide institutions, there are six confederations recognized by FIFA which oversee the game in the different continents and regions of the world. In total, FIFA recognizes 211 national associations and their associated men's national teams as well as 129 women's national teams; see the list of national football teams and their respective country codes. The number of FIFA member associations is higher than the number of UN member states as FIFA has admitted associations from 23 non-sovereign entities as members in their own right, such as the four Home Nations within the United Kingdom and the two special administrative regions of China: Hong Kong and Macau. The FIFA Men's World Rankings are updated monthly and rank each team based on their performance in international competitions, qualifiers, and friendly matches. There is also a world ranking for women's football, updated four times a year.

- As per article the player gets the chance for earning the bonus in every four years while representing for their country in world cup, the main source of income for them can't be this. A) The payout will be distributed by the national federation to team who had participated. B) Every country has their different policies to distribute the amount to the players. C) Salary is typical is out as per the skill level. D) They get paid the number of games they have played and being taken participated in. E) Winning bonus is as per
  1) Eliminated in group stage: $9 million
  2) Eliminated in round of 16: $13 million
  3) Eliminated in quarterfinals: $17 million
  4) Fourth Place: $25 million
  5) Third Place: $27 million
  6) Runner Up: $30 million
  7) Champion: $42 million
- The 2022 world cup is considered as most expensive in history, ten times more than the previous world cup which held in 2018, it's estimates cost is $220 billion right now.
- In 2014 world cup was won by Germany and in 2018 it was taken by France.

In this paper we have work on two different data that are related as the first one is FIFA player and another is FIFA team, In FIFA players we are predicting what are the wages of the players and we choose ten players for that and in FIFA team we are predicting which team is going to win the world cup 2022 and there are "n" numbers of teams which we are going with for prediction.

Related Work/Literature Review

For Players

- In the research of Kaggle for players they have come up with data as per the countries not by the overall players as per that the two prediction we have of them are from Argentina and Portugal and the prediction they came up with is: -
  1. Predicted wage of a player with Argentina nationality is: 8022.46 Euro/ week
     Mean wage of players with Argentina nationality is: 8360.83 Euro/ week
  2. Predicted wage of a player with Portugal nationality is: 14373.79 Euro/ week
     Mean wage of players with Portugal nationality is: 15323.04 Euro/ week

- In [1] that tell us about length of data set and here we find out the how many attracter, defender, goal keeper and mid fielder, this four data set are created to check the amount of missing value, which come up with that attribute value for GK that is goal keeper player data is missing altogether , so we drop the indexes and we are now left with the data which is 15148 observation.

- In [2] we have done distribution of overall rating of player; this follows a normal distribution with the mean value of 66. And only 25% have high rating i.e., more than 71 that can be understood why there are less team who have an overall strong squad.

- In [3] the top ten teams by total wages are displayed, which now make sense why the biggest world cup is here, it's clearly seen that the two Spanish clubs FC Barcelona and Real Madrid CF take up the top spot and there more in this count of top spot which is six English, one German and one Italian club in this list.

- In [4] we went with by playing position, the average salaries are sorted, and it is coming up as expected to be as per now that the most salaries are drawn by attacking and the come up with the other two which is midfielder and defenders.

- Then we have done the visualization by using tree visualization for DF(Defender) Model in which we have come up with the standing tackle, sliding tackle and dribbling and then at last we have two values that are final weekly wage and weekly wage.

- Now we have the visualization by using tree visualization for MF(Midfielder), came up same as defender just the smaller version of that.

- And at last, we came with the last tree visualization for AT(Attackers)

FOR TEAM

- In the research of Kaggle for teams they have perform in python and they are performing on point base and ranking and used random tree forest is used here, and for conclusion it's end without clarifying the conclusion and what the prediction can be made.
- There are very less different between the player and team
- Here I have use numbers for players data or pictures and for team will go with alphabetical orders
- [a] analyzing the available data so here the attribute is that home team, away team, home score, away score, tournament, city, country and neutral
- So here home team is the for example my team is from Boston, and we have blue color cloth so if we played in Boston my team will be considered as home team and I will going to have the plus point or advantage that blue color will be our and no other team can have it so whom so ever is coming from other than Boston will be considered as away team.
- Then we have went with the outliers finding in numerical data
- [b] We have added the new column name as win statues
- [c] Type of matches taken places in this the friendly match is the one which is not considered it's just for fun, just to know their strength and weakness and where they stand.
- Then we went with the experiment that was to; finding out the impact hosting a major tournament helps a country's chances in winning the matches. [d]
- Went with the home ground and awake ground result to get more into that
- Search which team is most successful in both the conditions that are awake and home.
- At last, we went with the running match of FIFA.

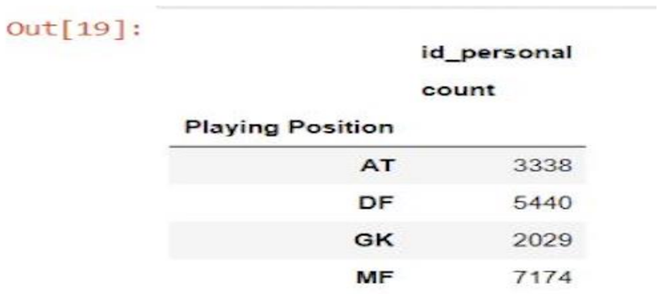
Methodology

For Players

1. The data here is available in three different tables
   - Attribute (for playing spot there are different attribute)
   - Personal (having personal information along with their wages and values)
   - Position (overall rating from different position)

2. We have used python (Jupyter notebook) in this we have import the data then we must clean the data and preparate it with all the rows and column we want:
   - Preferred position has multiple position separated by the space
   - Columns of value and wage are modified into proper format, as the value is in million and wages are in thousand, hence multiplied them with 1000000 and 1000 as shown in [5]
   - [1] is the player available in each category, which shows us what are the attribute missing and then remove the extra data set with NAN value

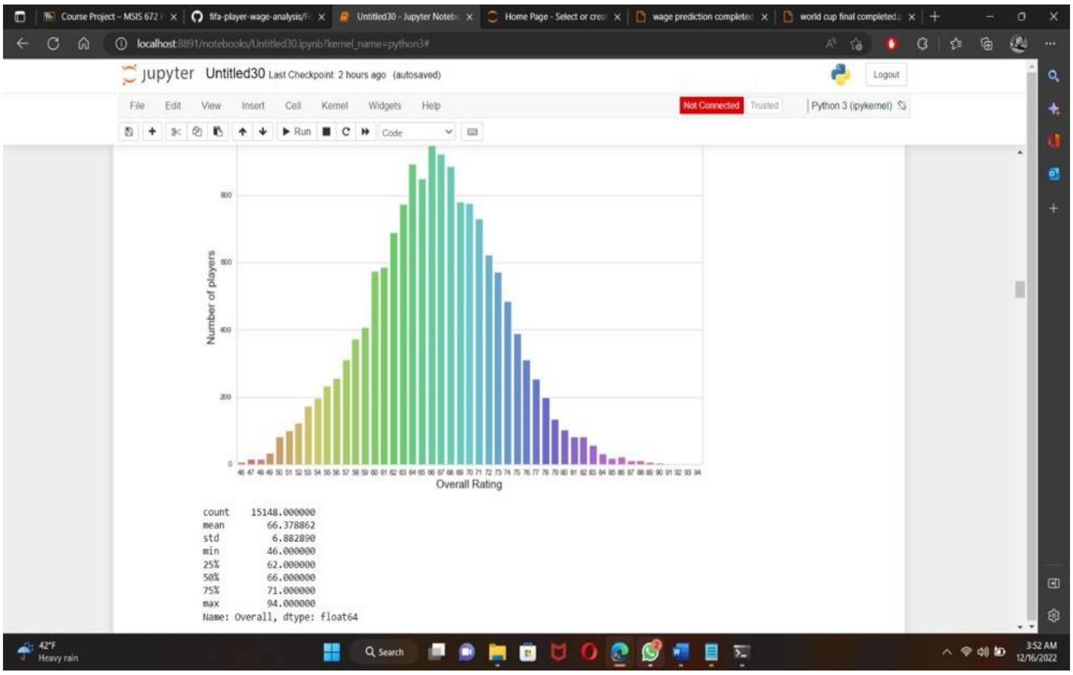
3. Exploratory Data Analysis
    - For further finding will check the distribution of some variable and try to infer some finding [6] is the one for age distribution which we have done first and got to know most of the player age range is lie between 20-30 value
    - Overall rating distribution of players [2]
    - Then we have used box plot for wage distribution, there are few players with high rating is clear that their wages will skew the below distribution.[7]
4. Variable relationship
    - As we saw in [6] we got to know the age of player are mostly in peak of 25 or early 30, so we did here age vs overall distribution which give us the conclusion that the most wage is also collected by same age range [8]
5. Club Wage Analysis
    - [3] we have seen the top 10 team and top two spot was taken by Spanish club, then we sort the list on the bases of average salaries before it was total salaries of players and came up with that most of them are from attacking attribute [4]
6. Predictive modeling
    - Built it with wage as the target/ response variable with all the variable from attribute table are predictors.
    - Position can’t act as variable as it is dependent on attribute variable.
    - Data was split into training and testing sets in 70:30 ratio respectively and built the model on training set and evaluated on testing set.
7. Linear Regression
    - Model built extremely high mean squared error with very low r squared value, (r square value explain the how much target variable is explained by predictors) which is only 32% over here.
    - Mean square error is 418756793.28
    - Coefficient of determination is 0.32
8. Variable selection through RFE
    - Recursive feature elimination to remove certain variable and improve model, however the improvement is not quite significant with 29 variable and still has a very low accuracy of 39.68%
    - Optimum number of features: 29
    - Score with 29 features :0.396846
9. Decision Tree
    - We have made decision tree for all the three-attribute using random forest defenders, attackers, midfielders.[9][11][10]



Out[19]:

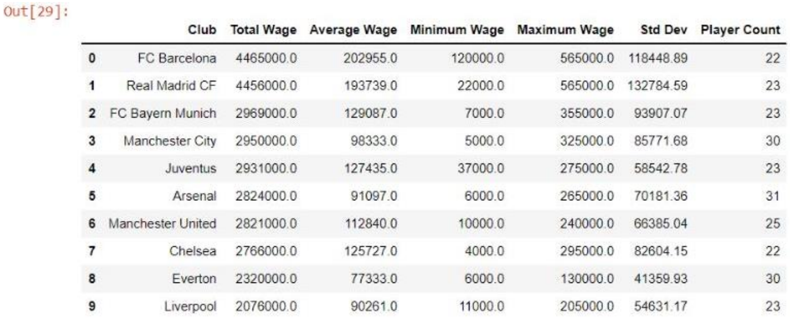
Playing Position	id_personal count
AT	3338
DF	5440
GK	2029
MF	7174

Figure 1



```
count    15148.000000
mean        66.378862
std          6.882890
min         46.000000
25%         62.000000
50%         66.000000
75%         71.000000
max         94.000000
Name: Overall, dtype: float64
```

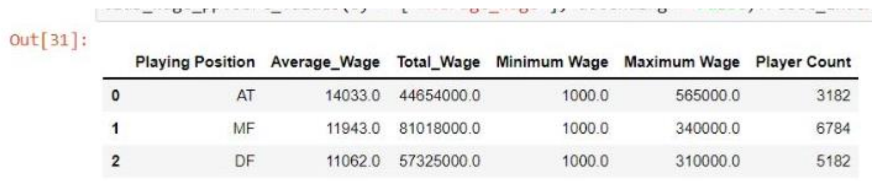
Figure 2



Out[29]:

	Club	Total Wage	Average Wage	Minimum Wage	Maximum Wage	Std Dev	Player Count
0	FC Barcelona	4465000.0	202955.0	120000.0	565000.0	118448.89	22
1	Real Madrid CF	4456000.0	193739.0	22000.0	565000.0	132784.59	23
2	FC Bayern Munich	2969000.0	129087.0	7000.0	355000.0	93907.07	23
3	Manchester City	2950000.0	98333.0	5000.0	325000.0	85771.68	30
4	Juventus	2931000.0	127435.0	37000.0	275000.0	58542.78	23
5	Arsenal	2824000.0	91097.0	6000.0	265000.0	70181.36	31
6	Manchester United	2821000.0	112840.0	10000.0	240000.0	66385.04	25
7	Chelsea	2766000.0	125727.0	4000.0	295000.0	82604.15	22
8	Everton	2320000.0	77333.0	6000.0	130000.0	41359.93	30
9	Liverpool	2076000.0	90261.0	11000.0	205000.0	54631.17	23

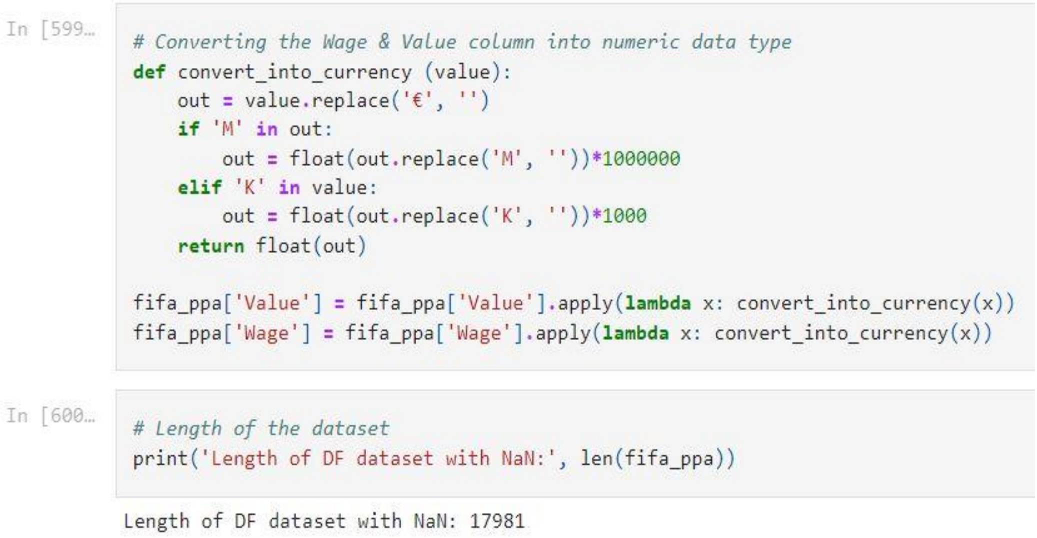
Figure 3



Out[31]:

	Playing Position	Average_Wage	Total_Wage	Minimum Wage	Maximum Wage	Player Count
0	AT	14033.0	44654000.0	1000.0	565000.0	3182
1	MF	11943.0	81018000.0	1000.0	340000.0	6784
2	DF	11062.0	57325000.0	1000.0	310000.0	5182

Figure 4

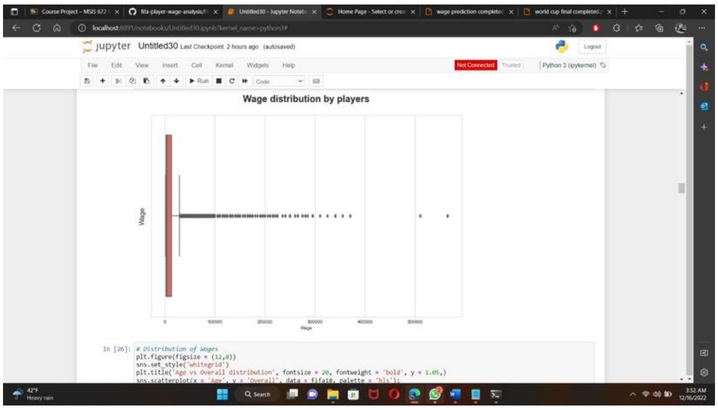


```
In [599…
# Converting the Wage & Value column into numeric data type
def convert_into_currency (value):
    out = value.replace('€', '')
    if 'M' in out:
        out = float(out.replace('M', ''))*1000000
    elif 'K' in value:
        out = float(out.replace('K', ''))*1000
    return float(out)

fifa_ppa['Value'] = fifa_ppa['Value'].apply(lambda x: convert_into_currency(x))
fifa_ppa['Wage'] = fifa_ppa['Wage'].apply(lambda x: convert_into_currency(x))
```

```
In [600…
# Length of the dataset
print('Length of DF dataset with NaN:', len(fifa_ppa))
```

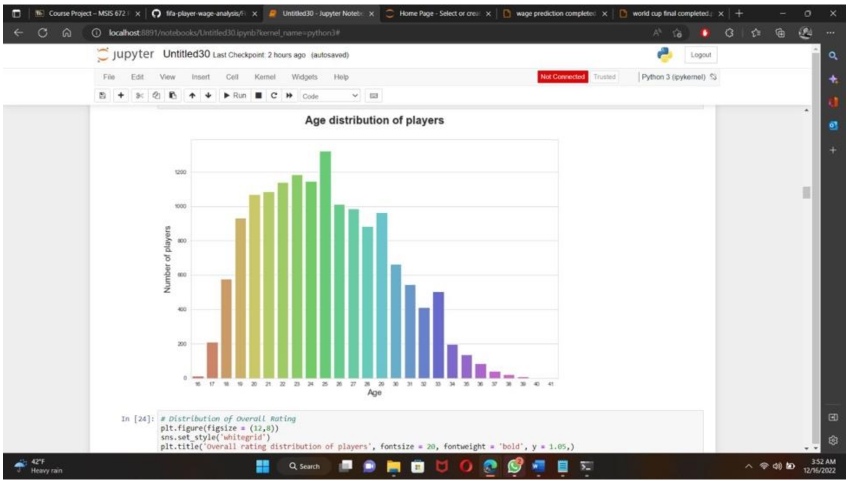
```
Length of DF dataset with NaN: 17981
```

Figure 5


jupyter Untitled30 Last Checkpoint: 2 hours ago (autosaved)
Not Connected
Trusted
Python 3 (ipykernel)
Wage distribution by players
Wage
0
100000
200000
300000
400000
500000
Wage

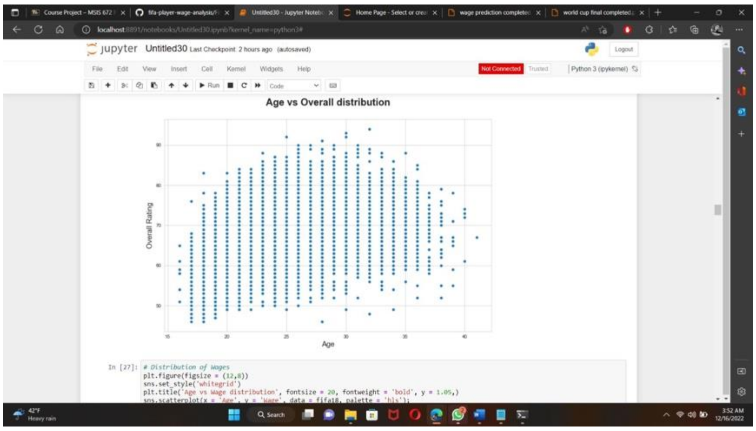
```
In [26]: # Distribution of Wages
         plt.figure(figsize = (12,8))
         sns.set_style('whitegrid')
         plt.title('Age vs Overall distribution', fontsize = 20, fontweight = 'bold', y = 1.05,)
         sns.scatterplot(x = 'Age', y = 'Overall', data = fifa18, palette = 'hls');
```

Figure 7



```
In [24]: # Distribution of Overall Rating
         plt.figure(figsize = (12,8))
         sns.set_style('whitegrid')
         plt.title('Overall rating distribution of players', fontsize = 20, fontweight = 'bold', y = 1.05,)
```

Figure 6


jupyter Untitled30 Last Checkpoint: 2 hours ago (autosaved)
Not Connected
Trusted
Python 3 (ipykernel)
Age vs Overall distribution
Overall Rating
50
60
70
80
90
15
20
25
30
35
40
Age

```
In [27]: # Distribution of Wages
         plt.figure(figsize = (12,8))
         sns.set_style('whitegrid')
         plt.title('Age vs Wage distribution', fontsize = 20, fontweight = 'bold', y = 1.05,)
         sns.scatterplot(x = 'Age', y = 'Wage', data = fifa18, palette = 'hls');
```

Figure8

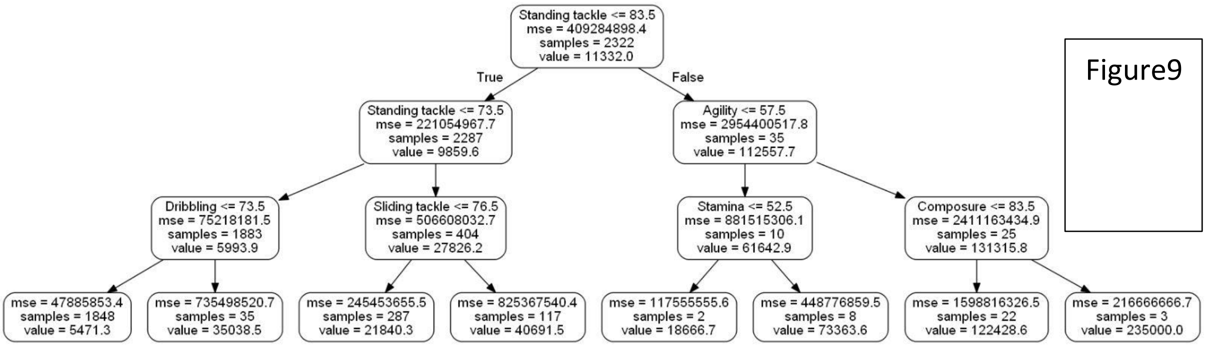


Figure9

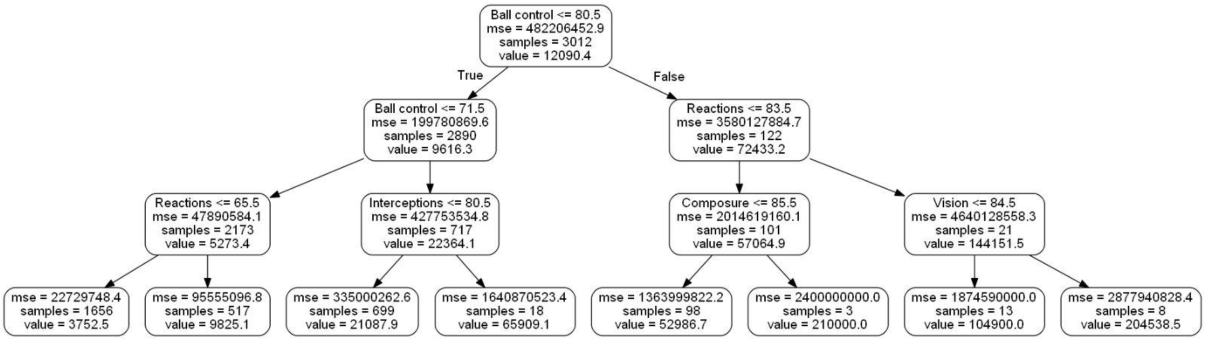


Figure 10

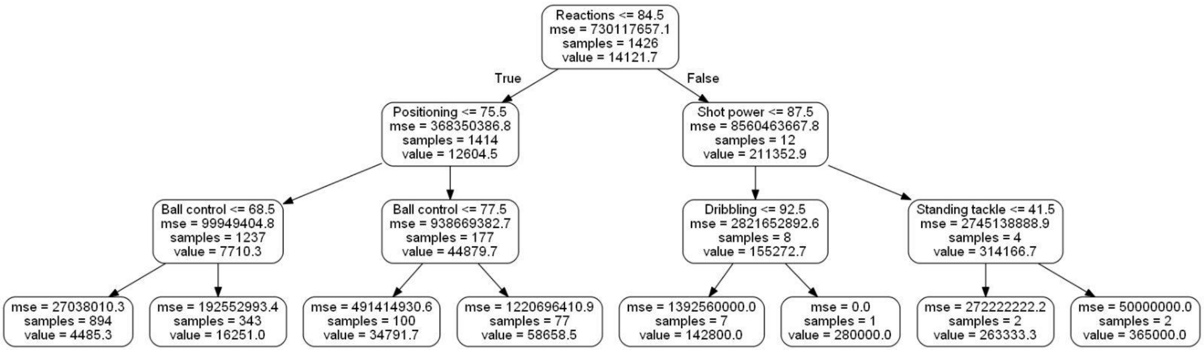


Figure 11

Conclusion

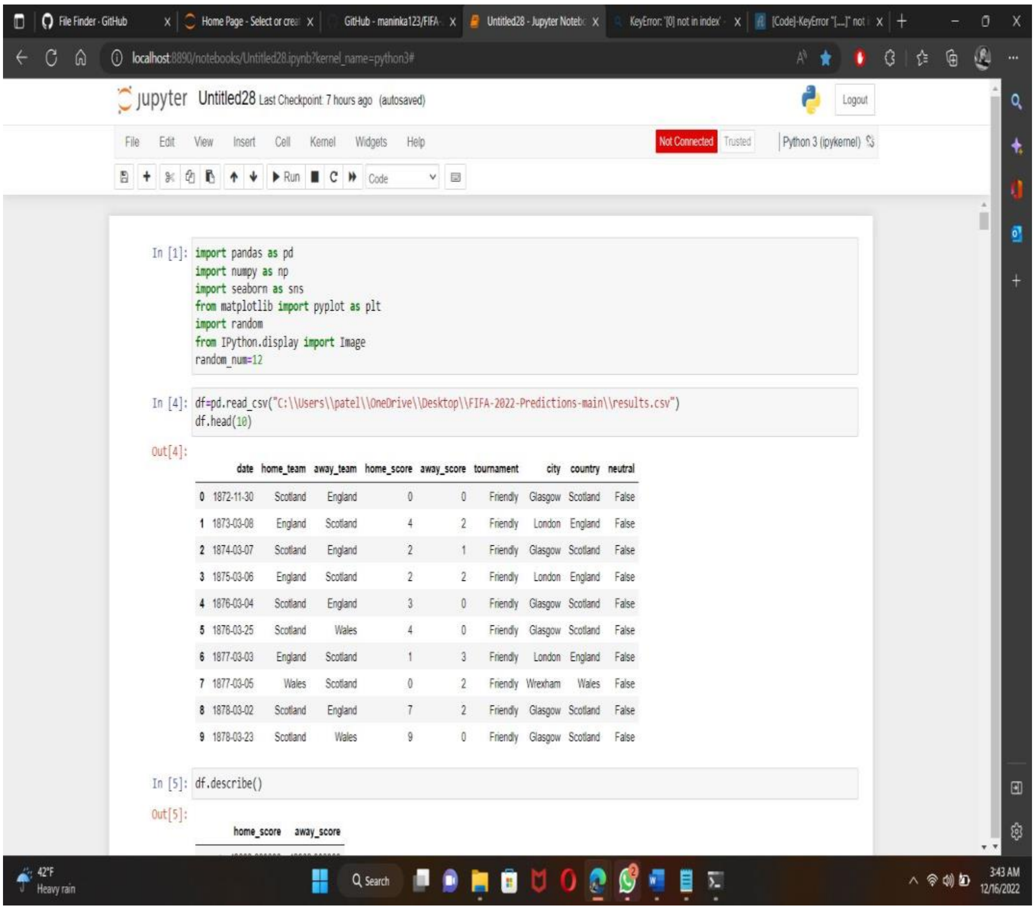
1) The elements mentioned above are the most significant ones affecting player salaries significantly. They are tackles, reactions, composure, and quick passing skills for defenders. They frequently must perform the unsavory task of attacking the opponent's player to prevent them from scoring goals. A bad decision might result in a player being suspended, which would be detrimental to the club, or, even worse, could gravely hurt the player. Therefore, teams are seeking for athletes that can perform their jobs with a sense of composure. To prevent penalties and make life simpler for their goalkeepers, they must move fluidly. Along with all other attributes, players in defending position should place more emphasis on the elements.

2) The job of a midfielder is difficult. They need to generate opportunities for their attacking partners while also taking a step back occasionally to help their defenders. As can be seen from the above statistics, midfielders must possess exceptional ball control skills. Following that are appropriate short passes, responses to circumstances, field location, and a clear understanding of inventiveness to score goals. With these criteria, the clubs are willing to be more flexible with their spending on midfielders.

3) Attacking players receive higher salary than other players, as was highlighted during the EDA. Due to the fact that they receive the most attention on the field, they must excel in their position. Because of this, Ronaldo and Messi get the highest salaries in the current generation. In order to receive the ball with quick reflexes, attacking players must be exceptional in their placement. To end with a goal, this must conclude with a flawless finish. In order to dribble past the defenders, players must have perfect ball control throughout. In addition to all other areas, the attacking players can devote more time in training to bolstering their strength in these ones.

For Team

- Analyzing the data
- Here we went with analyzing the data and having the look over the home team and away team how in this both the case team perform.
- Then we have calculated what country have played what number of matches till the date how many matches they have played till now.
- Then we have also calculated what team played in both the situation and how they did.
- We have move forward with searching for most successful team in both conditions separately
- And last we went with the running match
    1) Here we made a copy of main data frame for future use.
    2) Went with the machine learning model to make new data set by adding year, country in which it takes place, team1 and team2 score of teams 1 and team 2 (that is team1 against team 2 and by what number they win or lose)
    3) Made a heat map to get correlation of each column
    4) We tried to make a prediction [e]
    5) Made confusion matrix and then plotting matrix I heatmap[f]
    6) Classified both the team matrix differently
    7) And made sample prediction output [g]
    8) Run the model and initialize basic parameters[h], group stage match after that group stage results
    9) Divided rounds in two parts, came up with quarter final match
    10) After that what team will move forward in semifinals and came to the finals and at last finals[i]
    11) Overall, we made a model first and put all the data of previous matches and took all the possible cases for that and tried once the model then apply all the terms and conditions of running match which covers all the data of previous matches and gave the results as per that.



```
In [1]: import pandas as pd
        import numpy as np
        import seaborn as sns
        from matplotlib import pyplot as plt
        import random
        from IPython.display import Image
        random_num=12
```

```
In [4]: df=pd.read_csv("C:\\Users\\patel\\OneDrive\\Desktop\\FIFA-2022-Predictions-main\\results.csv")
        df.head(10)
```

Out[4]:

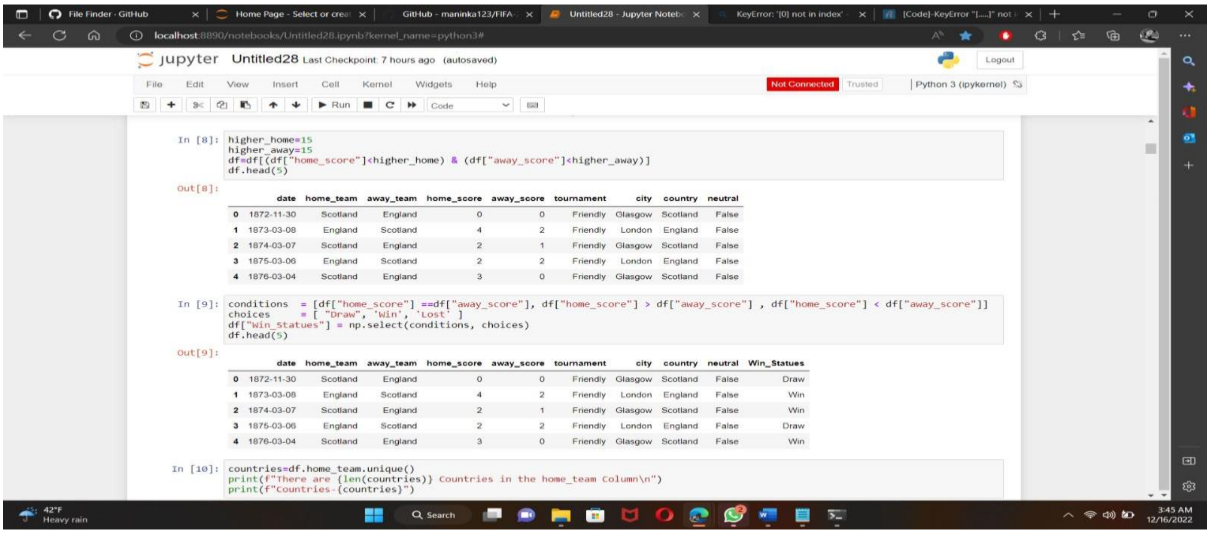
	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral
0	1872-11-30	Scotland	England	0	0	Friendly	Glasgow	Scotland	False
1	1873-03-08	England	Scotland	4	2	Friendly	London	England	False
2	1874-03-07	Scotland	England	2	1	Friendly	Glasgow	Scotland	False
3	1875-03-06	England	Scotland	2	2	Friendly	London	England	False
4	1876-03-04	Scotland	England	3	0	Friendly	Glasgow	Scotland	False
5	1876-03-25	Scotland	Wales	4	0	Friendly	Glasgow	Scotland	False
6	1877-03-03	England	Scotland	1	3	Friendly	London	England	False
7	1877-03-05	Wales	Scotland	0	2	Friendly	Wrexham	Wales	False
8	1878-03-02	Scotland	England	7	2	Friendly	Glasgow	Scotland	False
9	1878-03-23	Scotland	Wales	9	0	Friendly	Glasgow	Scotland	False

```
In [5]: df.describe()
```

Out[5]:

	home_score	away_score

Figure a



```
In [8]: higher_home=15
        higher_away=15
        df=df[(df["home_score"]<higher_home) & (df["away_score"]<higher_away)]
        df.head(5)
```

Out[8]:

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral
0	1872-11-30	Scotland	England	0	0	Friendly	Glasgow	Scotland	False
1	1873-03-08	England	Scotland	4	2	Friendly	London	England	False
2	1874-03-07	Scotland	England	2	1	Friendly	Glasgow	Scotland	False
3	1875-03-06	England	Scotland	2	2	Friendly	London	England	False
4	1876-03-04	Scotland	England	3	0	Friendly	Glasgow	Scotland	False

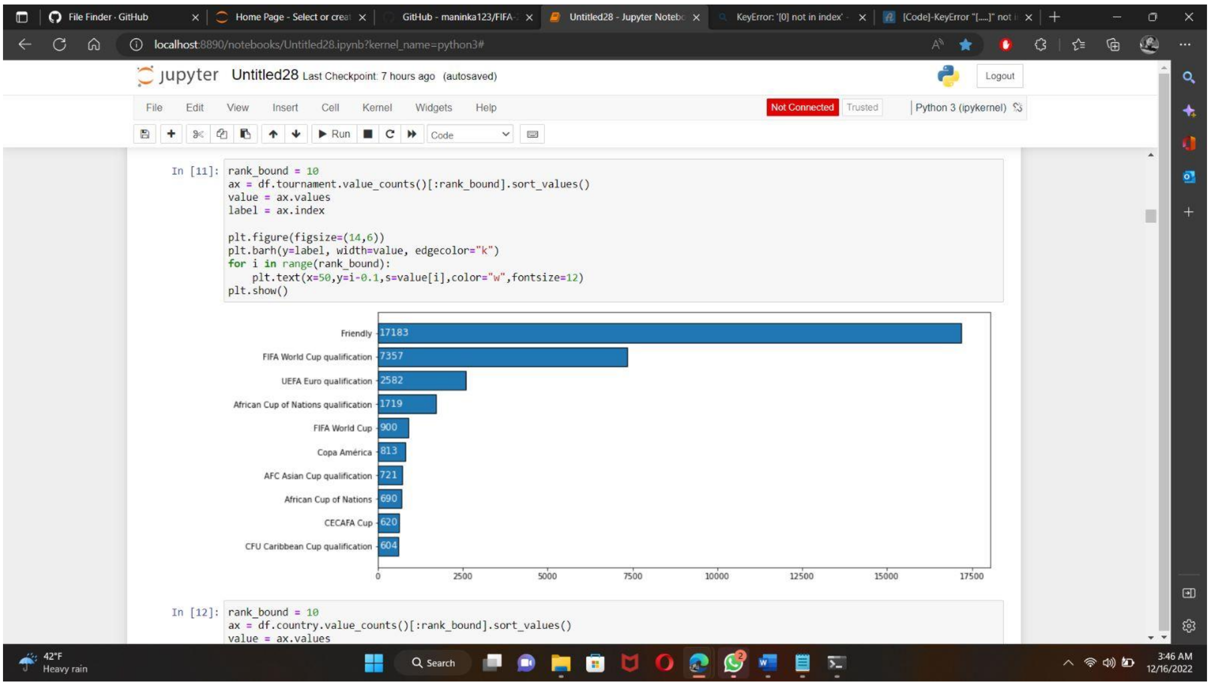
```
In [9]: conditions  = [df["home_score"] ==df["away_score"], df["home_score"] > df["away_score"] , df["home_score"] < df["away_score"]]
        choices     = [ "Draw", 'Win', 'Lost' ]
        df["Win_Statues"] = np.select(conditions, choices)
        df.head(5)
```

Out[9]:

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	Win_Statues
0	1872-11-30	Scotland	England	0	0	Friendly	Glasgow	Scotland	False	Draw
1	1873-03-08	England	Scotland	4	2	Friendly	London	England	False	Win
2	1874-03-07	Scotland	England	2	1	Friendly	Glasgow	Scotland	False	Win
3	1875-03-06	England	Scotland	2	2	Friendly	London	England	False	Draw
4	1876-03-04	Scotland	England	3	0	Friendly	Glasgow	Scotland	False	Win

```
In [10]: countries=df.home_team.unique()
         print(f"There are {len(countries)} Countries in the home_team Column\n")
         print(f"Countries-{countries}")
```

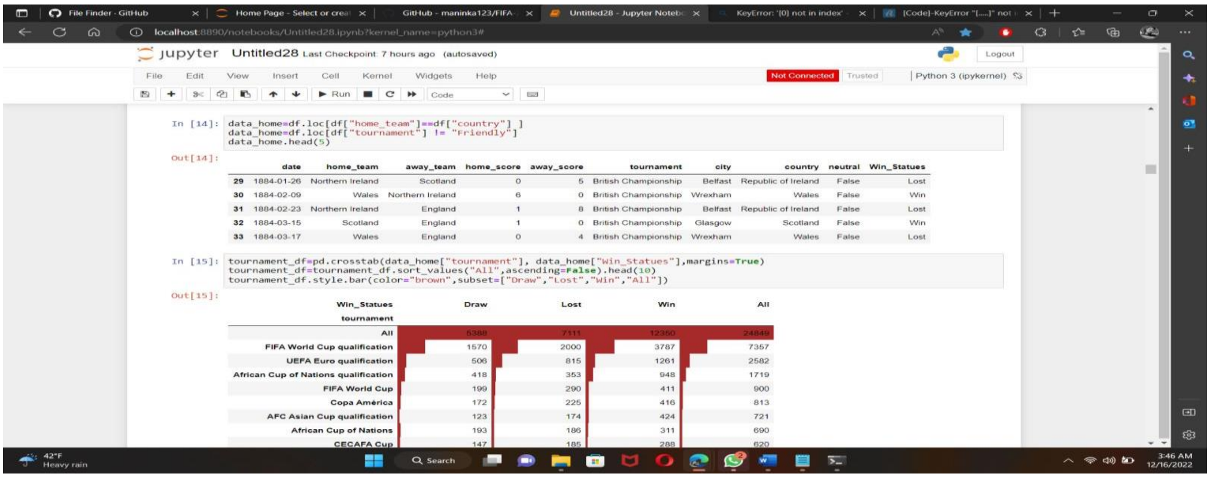
Figure b



```
In [11]: rank_bound = 10
         ax = df.tournament.value_counts()[:rank_bound].sort_values()
         value = ax.values
         label = ax.index

         plt.figure(figsize=(14,6))
         plt.barh(y=label, width=value, edgecolor="k")
         for i in range(rank_bound):
             plt.text(x=50,y=i-0.1,s=value[i],color="w",fontsize=12)
         plt.show()
```

```
In [12]: rank_bound = 10
         ax = df.country.value_counts()[:rank_bound].sort_values()
         value = ax.values
```

Figure c



```
In [14]: data_home=df.loc[df["home_team"]==df["country"] ]
         data_home=df.loc[df["tournament"] != "Friendly"]
         data_home.head(5)
```

Out[14]:

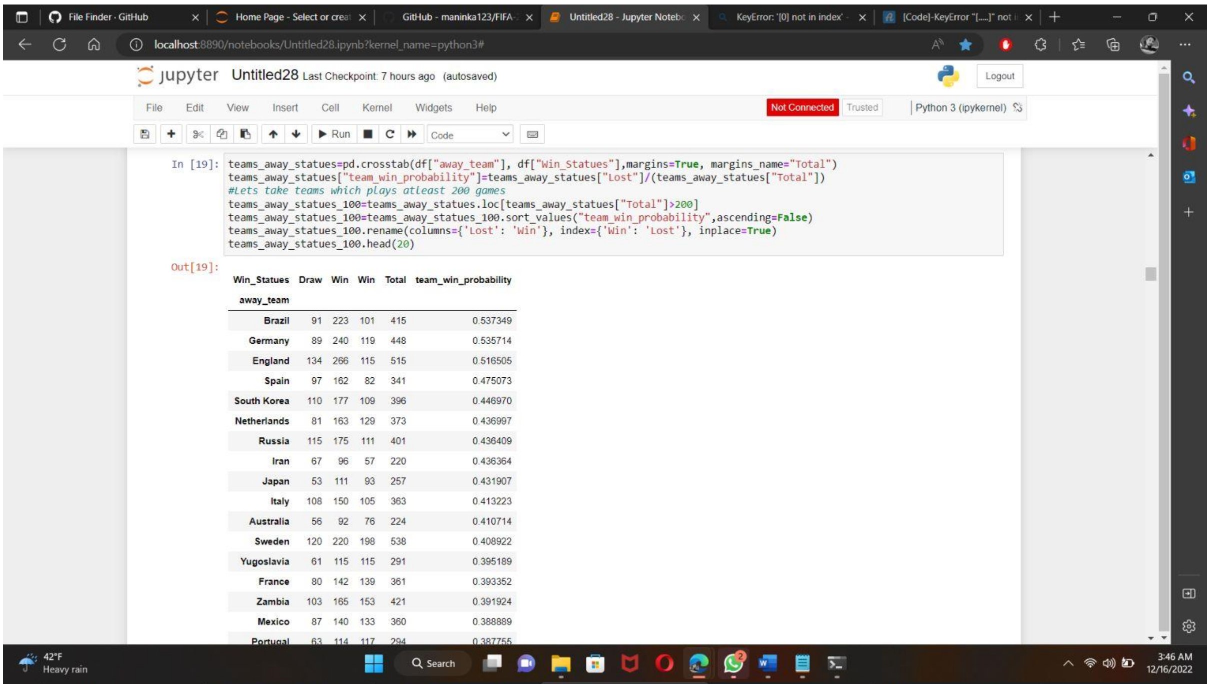
	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	Win_Statues
29	1884-01-26	Northern Ireland	Scotland	0	5	British Championship	Belfast	Republic of Ireland	False	Lost
30	1884-02-09	Wales	Northern Ireland	6	0	British Championship	Wrexham	Wales	False	Win
31	1884-02-23	Northern Ireland	England	1	8	British Championship	Belfast	Republic of Ireland	False	Lost
32	1884-03-15	Scotland	England	1	0	British Championship	Glasgow	Scotland	False	Win
33	1884-03-17	Wales	England	0	4	British Championship	Wrexham	Wales	False	Lost

```
In [15]: tournament_df=pd.crosstab(data_home["tournament"], data_home["Win_Statues"],margins=True)
         tournament_df=tournament_df.sort_values("All",ascending=False).head(10)
         tournament_df.style.bar(color="brown",subset=["Draw","Lost","Win","All"])
```

Out[15]:

Win_Statues	Draw	Lost	Win	All
tournament				
All	5388	7111	12350	24849
FIFA World Cup qualification	1570	2000	3787	7357
UEFA Euro qualification	506	815	1261	2582
African Cup of Nations qualification	418	353	948	1719
FIFA World Cup	199	290	411	900
Copa América	172	225	416	813
AFC Asian Cup qualification	123	174	424	721
African Cup of Nations	193	186	311	690
CECAFA Cup	147	185	288	620

Figure d

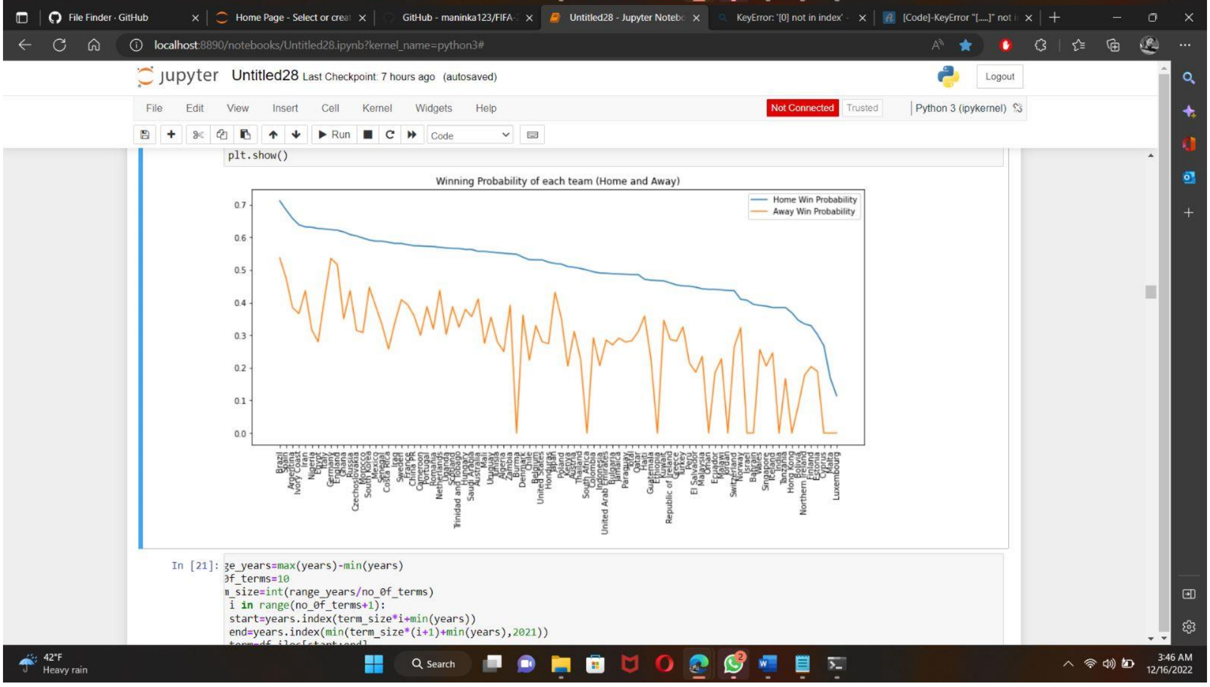

File Finder · GitHub
Home Page - Select or creat
GitHub - maninka123/FIFA-
Untitled28 - Jupyter Notebo
KeyError: '[0] not in index'
[Code]-KeyError "[....]" not i
localhost:8890/notebooks/Untitled28.ipynb?kernel_name=python3#
jupyter
Untitled28 Last Checkpoint: 7 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Connected
Trusted
Python 3 (ipykernel)
Run
Code
42°F
Heavy rain
Search
3:46 AM
12/16/2022

In [19]:

```
teams_away_statues=pd.crosstab(df["away_team"], df["Win_Statues"],margins=True, margins_name="Total")
teams_away_statues["team_win_probability"]=teams_away_statues["Lost"]/(teams_away_statues["Total"])
#Lets take teams which plays atleast 200 games
teams_away_statues_100=teams_away_statues.loc[teams_away_statues["Total"]>200]
teams_away_statues_100=teams_away_statues_100.sort_values("team_win_probability",ascending=False)
teams_away_statues_100.rename(columns={'Lost': 'Win'}, index={'Win': 'Lost'}, inplace=True)
teams_away_statues_100.head(20)
```

Out[19]:

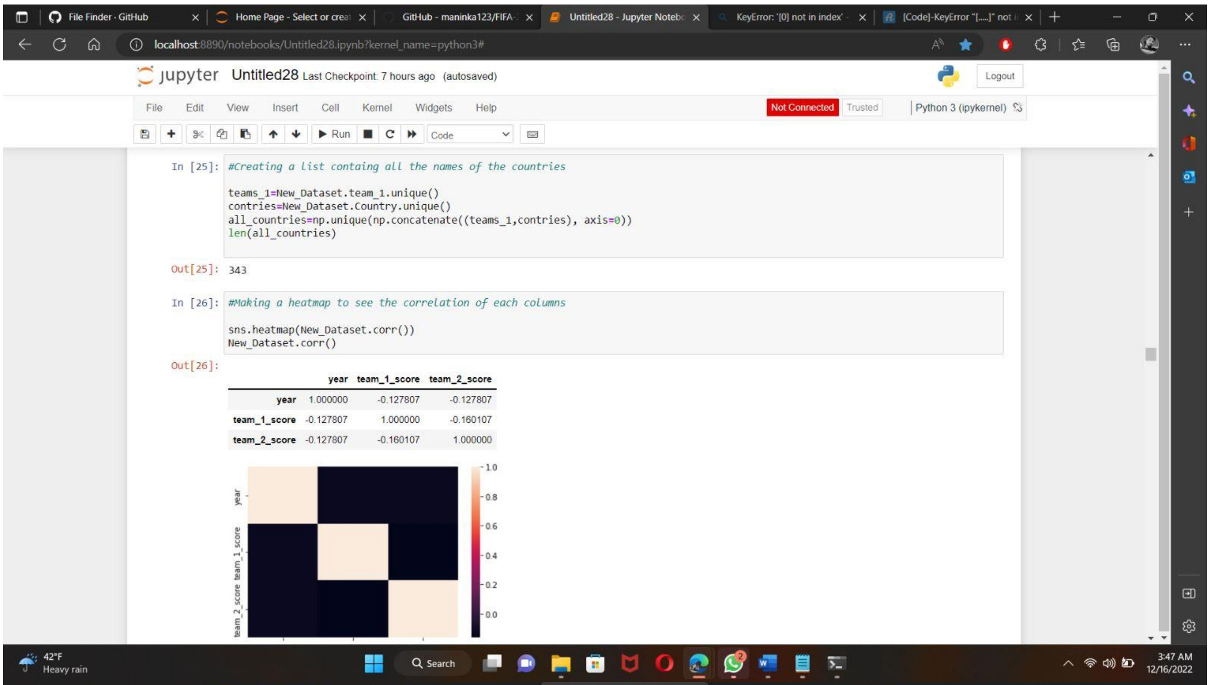
Win_Statues	Draw	Win	Win	Total	team_win_probability
away_team					
Brazil	91	223	101	415	0.537349
Germany	89	240	119	448	0.535714
England	134	266	115	515	0.516505
Spain	97	162	82	341	0.475073
South Korea	110	177	109	396	0.446970
Netherlands	81	163	129	373	0.436997
Russia	115	175	111	401	0.436409
Iran	67	96	57	220	0.436364
Japan	53	111	93	257	0.431907
Italy	108	150	105	363	0.413223
Australia	56	92	76	224	0.410714
Sweden	120	220	198	538	0.408922
Yugoslavia	61	115	115	291	0.395189
France	80	142	139	361	0.393352
Zambia	103	165	153	421	0.391924
Mexico	87	140	133	360	0.388889
Portugal	63	114	117	294	0.387755



```
plt.show()
```

In [21]:

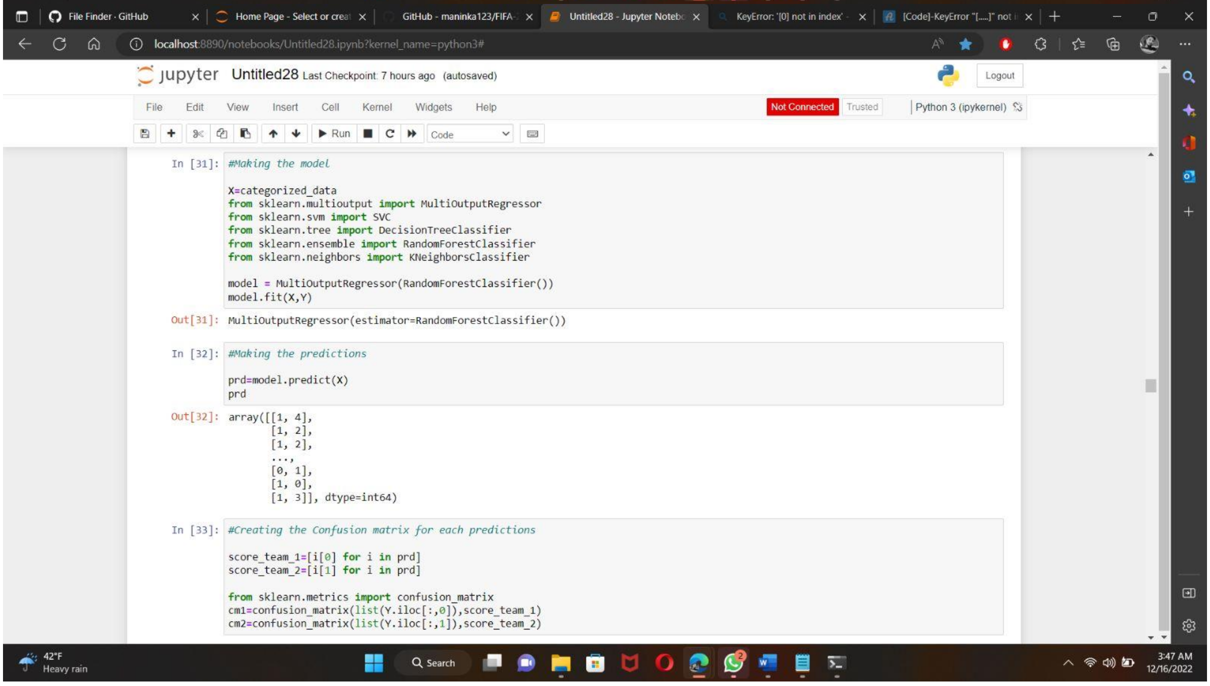
```
ge_years=max(years)-min(years)
0f_terms=10
n_size=int(range_years/no_0f_terms)
 i in range(no_0f_terms+1):
 start=years.index(term_size*i+min(years))
 end=years.index(min(term_size*(i+1)+min(years),2021))
```


File Finder · GitHub
Home Page - Select or creat
GitHub - maninka123/FIFA-
Untitled28 - Jupyter Notebo
KeyError: '[0] not in index'
[Code]-KeyError "[....]" not i
localhost:8890/notebooks/Untitled28.ipynb?kernel_name=python3#
jupyter Untitled28 Last Checkpoint: 7 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Connected
Trusted
Python 3 (ipykernel)
Run
Code
42°F
Heavy rain
Search
3:47 AM
12/16/2022

```
In [25]: #Creating a list containg all the names of the countries

         teams_1=New_Dataset.team_1.unique()
         contries=New_Dataset.Country.unique()
         all_countries=np.unique(np.concatenate((teams_1,contries), axis=0))
         len(all_countries)

Out[25]: 343

In [26]: #Making a heatmap to see the correlation of each columns

         sns.heatmap(New_Dataset.corr())
         New_Dataset.corr()

Out[26]:
```

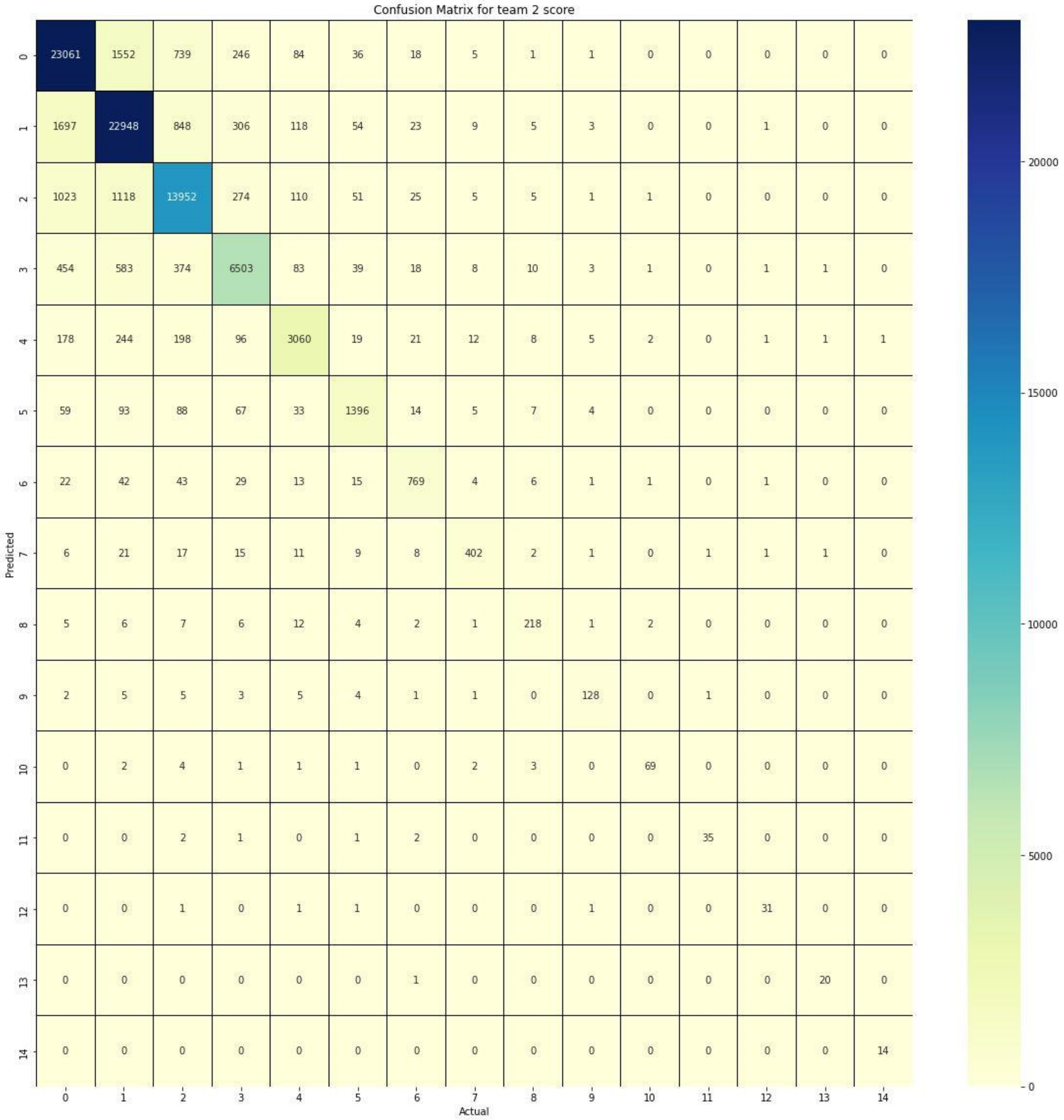
	year	team_1_score	team_2_score
year	1.000000	-0.127807	-0.127807
team_1_score	-0.127807	1.000000	-0.160107
team_2_score	-0.127807	-0.160107	1.000000

year
team_1_score
team_2_score
1.0
0.8
0.6
0.4
0.2
0.0


File Finder · GitHub
Home Page - Select or creat
GitHub - maninka123/FIFA-
Untitled28 - Jupyter Notebo
KeyError: '[0] not in index'
[Code]-KeyError "[....]" not i
localhost:8890/notebooks/Untitled28.ipynb?kernel_name=python3#
jupyter Untitled28 Last Checkpoint: 7 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Connected
Trusted
Python 3 (ipykernel)
Run
Code
42°F
Heavy rain
Search
3:47 AM
12/16/2022

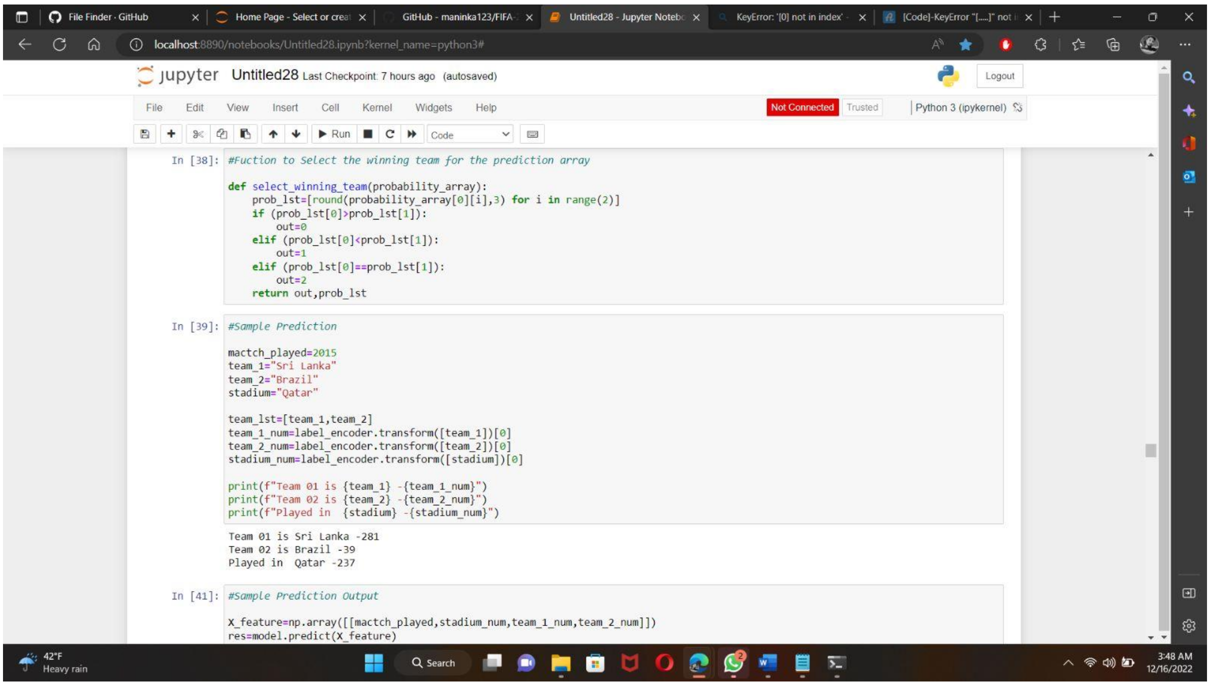
```
In [31]: #Making the model

         X=categorized_data
         from sklearn.multioutput import MultiOutputRegressor
         from sklearn.svm import SVC
         from sklearn.tree import DecisionTreeClassifier
         from sklearn.ensemble import RandomForestClassifier
         from sklearn.neighbors import KNeighborsClassifier

         model = MultiOutputRegressor(RandomForestClassifier())
         model.fit(X,Y)

Out[31]: MultiOutputRegressor(estimator=RandomForestClassifier())

In [32]: #Making the predictions

         prd=model.predict(X)
         prd

Out[32]: array([[1, 4],
                [1, 2],
                [1, 2],
                ...,
                [0, 1],
                [1, 0],
                [1, 3]], dtype=int64)

In [33]: #Creating the Confusion matrix for each predictions

         score_team_1=[i[0] for i in prd]
         score_team_2=[i[1] for i in prd]

         from sklearn.metrics import confusion_matrix
         cm1=confusion_matrix(list(Y.iloc[:,0]),score_team_1)
         cm2=confusion_matrix(list(Y.iloc[:,1]),score_team_2)
```

Figure e



Predicted \ Actual	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	23061	1552	739	246	84	36	18	5	1	1	0	0	0	0	0
1	1697	22948	848	306	118	54	23	9	5	3	0	0	1	0	0
2	1023	1118	13952	274	110	51	25	5	5	1	1	0	0	0	0
3	454	583	374	6503	83	39	18	8	10	3	1	0	1	1	0
4	178	244	198	96	3060	19	21	12	8	5	2	0	1	1	1
5	59	93	88	67	33	1396	14	5	7	4	0	0	0	0	0
6	22	42	43	29	13	15	769	4	6	1	1	0	1	0	0
7	6	21	17	15	11	9	8	402	2	1	0	1	1	1	0
8	5	6	7	6	12	4	2	1	218	1	2	0	0	0	0
9	2	5	5	3	5	4	1	1	0	128	0	1	0	0	0
10	0	2	4	1	1	1	0	2	3	0	69	0	0	0	0
11	0	0	2	1	0	1	2	0	0	0	0	35	0	0	0
12	0	0	1	0	1	1	0	0	0	1	0	0	31	0	0
13	0	0	0	0	0	0	1	0	0	0	0	0	0	20	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14

Figure f



localhost:8890/notebooks/Untitled28.ipynb?kernel_name=python3#
jupyter Untitled28 Last Checkpoint: 7 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Connected
Trusted
Python 3 (ipykernel)
Run
Code
42°F Heavy rain
Search
3:48 AM 12/16/2022

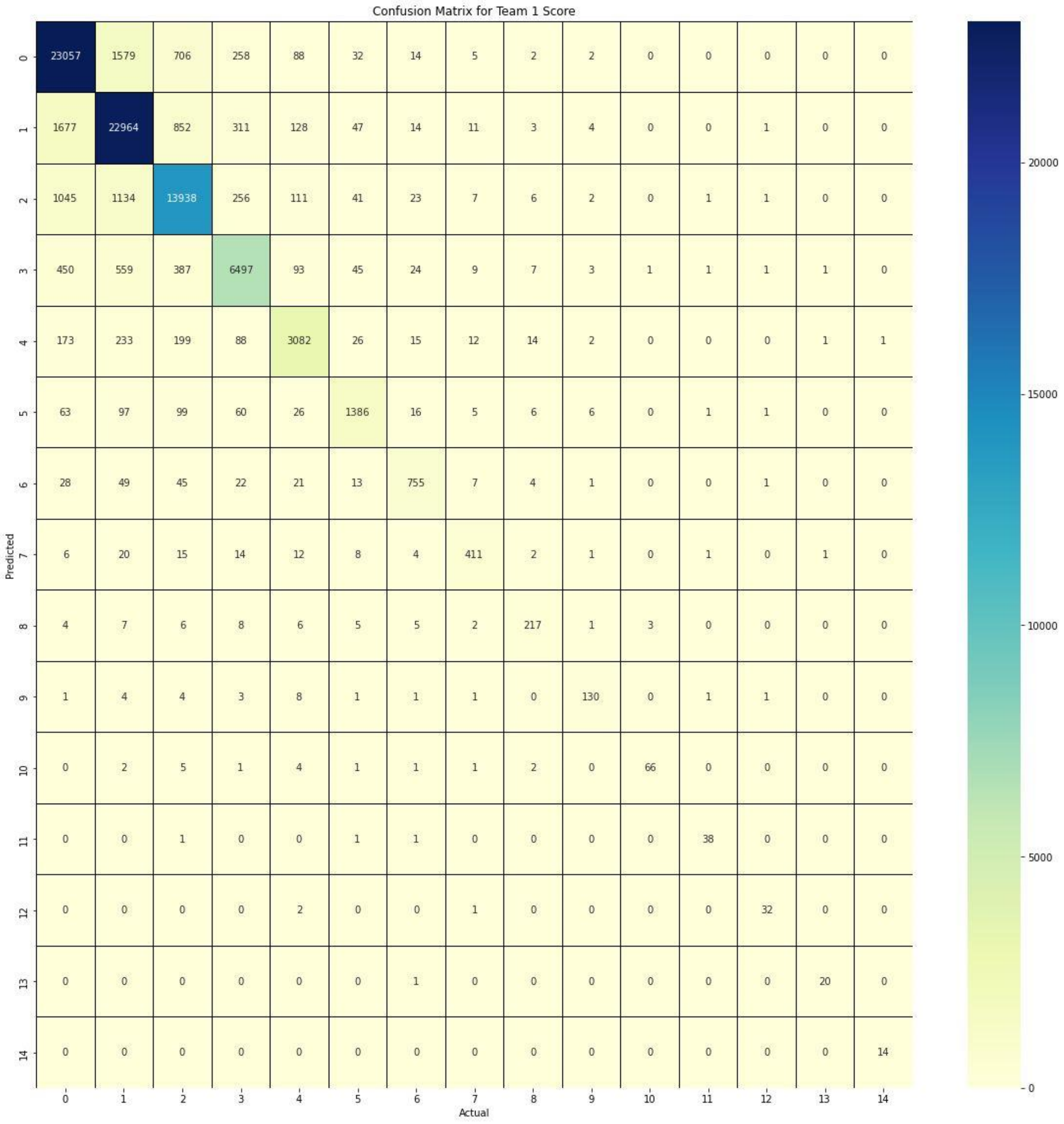
```
In [38]: #Fuction to Select the winning team for the prediction array

         def select_winning_team(probability_array):
             prob_lst=[round(probability_array[0][i],3) for i in range(2)]
             if (prob_lst[0]>prob_lst[1]):
                 out=0
             elif (prob_lst[0]<prob_lst[1]):
                 out=1
             elif (prob_lst[0]==prob_lst[1]):
                 out=2
             return out,prob_lst
```

```
In [39]: #Sample Prediction

         mactch_played=2015
         team_1="Sri Lanka"
         team_2="Brazil"
         stadium="Qatar"

         team_lst=[team_1,team_2]
         team_1_num=label_encoder.transform([team_1])[0]
         team_2_num=label_encoder.transform([team_2])[0]
         stadium_num=label_encoder.transform([stadium])[0]

         print(f"Team 01 is {team_1} -{team_1_num}")
         print(f"Team 02 is {team_2} -{team_2_num}")
         print(f"Played in  {stadium} -{stadium_num}")
```

```
Team 01 is Sri Lanka -281
Team 02 is Brazil -39
Played in  Qatar -237
```

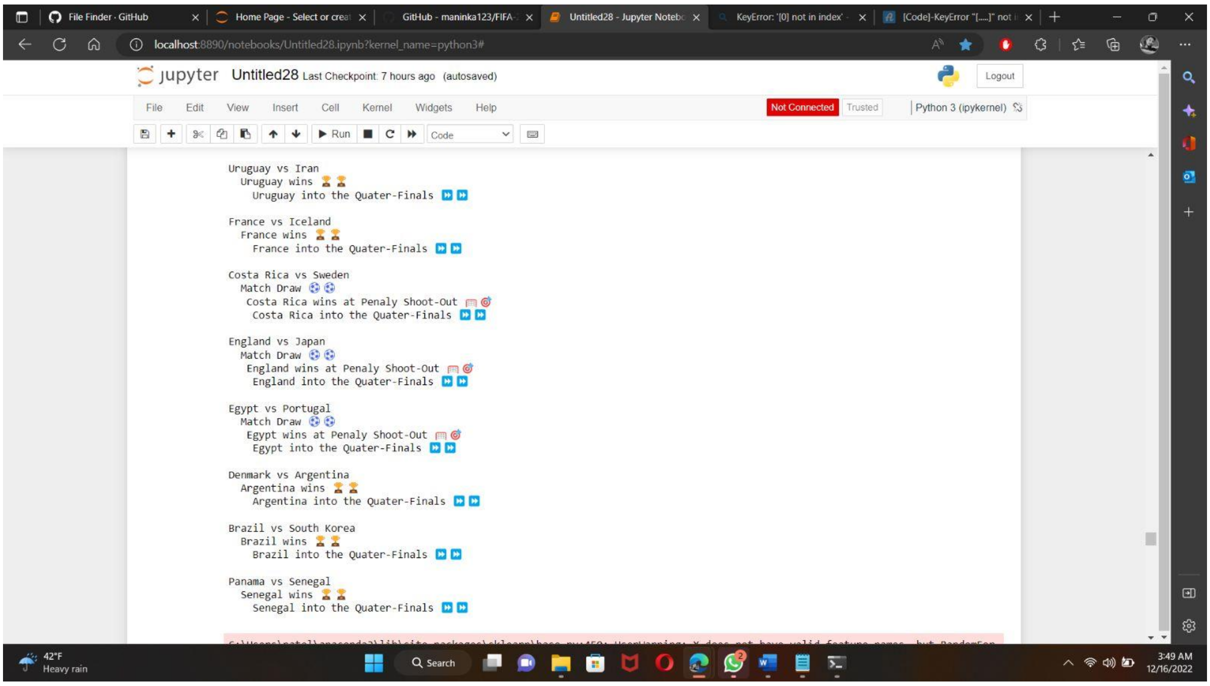
```
In [41]: #Sample Prediction Output

         X_feature=np.array([[mactch_played,stadium_num,team_1_num,team_2_num]])
         res=model.predict(X_feature)
```

Figure g



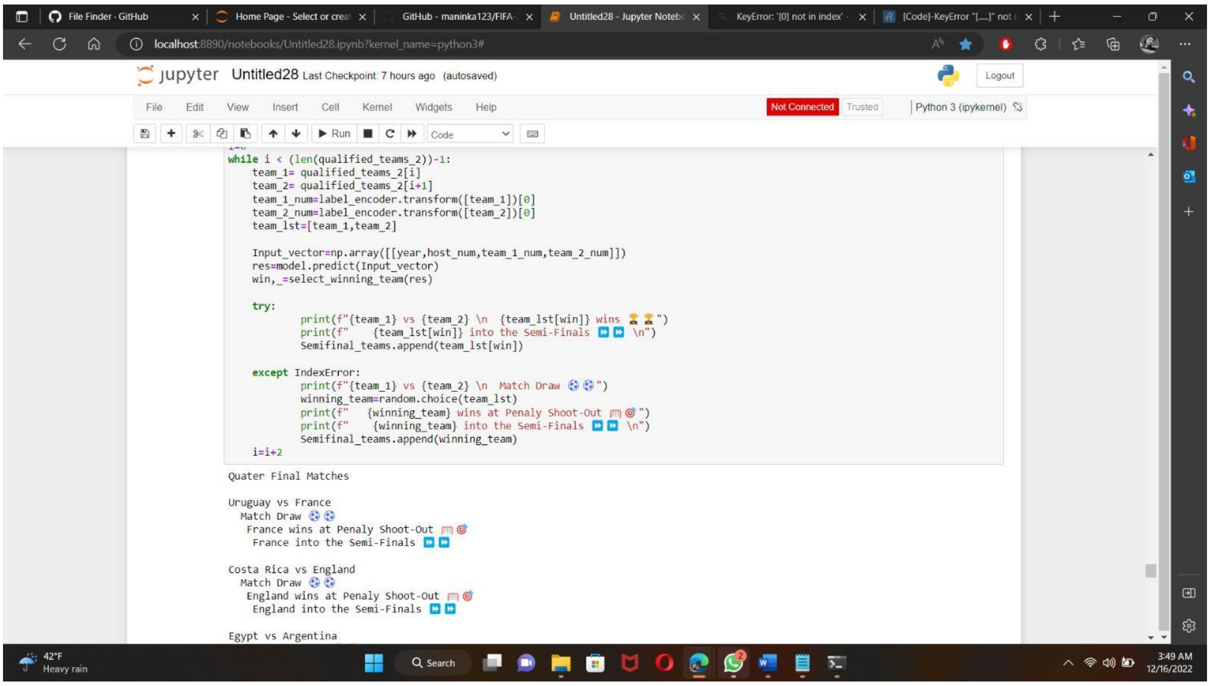
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	23057	1579	706	258	88	32	14	5	2	2	0	0	0	0	0
1	1677	22964	852	311	128	47	14	11	3	4	0	0	1	0	0
2	1045	1134	13938	256	111	41	23	7	6	2	0	1	1	0	0
3	450	559	387	6497	93	45	24	9	7	3	1	1	1	1	0
4	173	233	199	88	3082	26	15	12	14	2	0	0	0	1	1
5	63	97	99	60	26	1386	16	5	6	6	0	1	1	0	0
6	28	49	45	22	21	13	755	7	4	1	0	0	1	0	0
7	6	20	15	14	12	8	4	411	2	1	0	1	0	1	0
8	4	7	6	8	6	5	5	2	217	1	3	0	0	0	0
9	1	4	4	3	8	1	1	1	0	130	0	1	1	0	0
10	0	2	5	1	4	1	1	1	2	0	66	0	0	0	0
11	0	0	1	0	0	1	1	0	0	0	0	38	0	0	0
12	0	0	0	0	2	0	0	1	0	0	0	0	32	0	0
13	0	0	0	0	0	0	1	0	0	0	0	0	0	20	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14

Figure f(a)



```
Uruguay vs Iran
  Uruguay wins 🏆🏆
    Uruguay into the Quater-Finals ⏩⏩

France vs Iceland
  France wins 🏆🏆
    France into the Quater-Finals ⏩⏩

Costa Rica vs Sweden
  Match Draw ⚽⚽
   Costa Rica wins at Penaly Shoot-Out 🥅🎯
    Costa Rica into the Quater-Finals ⏩⏩

England vs Japan
  Match Draw ⚽⚽
   England wins at Penaly Shoot-Out 🥅🎯
    England into the Quater-Finals ⏩⏩

Egypt vs Portugal
  Match Draw ⚽⚽
   Egypt wins at Penaly Shoot-Out 🥅🎯
    Egypt into the Quater-Finals ⏩⏩

Denmark vs Argentina
  Argentina wins 🏆🏆
    Argentina into the Quater-Finals ⏩⏩

Brazil vs South Korea
  Brazil wins 🏆🏆
    Brazil into the Quater-Finals ⏩⏩

Panama vs Senegal
  Senegal wins 🏆🏆
    Senegal into the Quater-Finals ⏩⏩
```

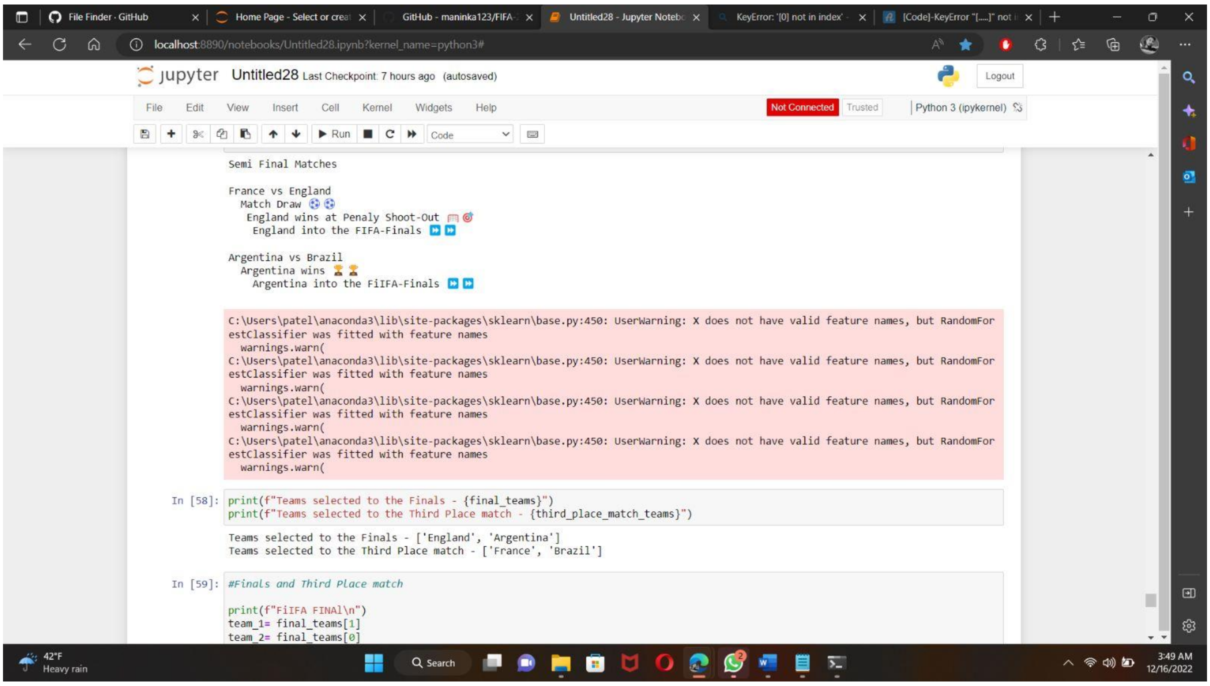
Figure i



```python
while i < (len(qualified_teams_2))-1:
    team_1= qualified_teams_2[i]
    team_2= qualified_teams_2[i+1]
    team_1_num=label_encoder.transform([team_1])[0]
    team_2_num=label_encoder.transform([team_2])[0]
    team_lst=[team_1,team_2]

    Input_vector=np.array([[year,host_num,team_1_num,team_2_num]])
    res=model.predict(Input_vector)
    win,_=select_winning_team(res)

    try:
            print(f"{team_1} vs {team_2} \n  {team_lst[win]} wins 🏆🏆")
            print(f"    {team_lst[win]} into the Semi-Finals ⏩⏩ \n")
            Semifinal_teams.append(team_lst[win])

    except IndexError:
            print(f"{team_1} vs {team_2} \n  Match Draw ⚽⚽")
            winning_team=random.choice(team_lst)
            print(f"   {winning_team} wins at Penaly Shoot-Out 🥅🎯")
            print(f"    {winning_team} into the Semi-Finals ⏩⏩ \n")
            Semifinal_teams.append(winning_team)
    i=i+2
```

```
Quater Final Matches

Uruguay vs France
  Match Draw ⚽⚽
   France wins at Penaly Shoot-Out 🥅🎯
    France into the Semi-Finals ⏩⏩

Costa Rica vs England
  Match Draw ⚽⚽
   England wins at Penaly Shoot-Out 🥅🎯
    England into the Semi-Finals ⏩⏩

Egypt vs Argentina
```

Figure i(a)


File Finder · GitHub
Home Page - Select or creat
GitHub - maninka123/FIFA-
Untitled28 - Jupyter Notebo
KeyError: '[0] not in index'
[Code]-KeyError "[...]" not i
localhost:8890/notebooks/Untitled28.ipynb?kernel_name=python3#
jupyter
Untitled28 Last Checkpoint: 7 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Connected
Trusted
Python 3 (ipykernel)
Run
Code
42°F
Heavy rain
Search
3:49 AM
12/16/2022

```
Semi Final Matches

France vs England
  Match Draw ⚽⚽
   England wins at Penaly Shoot-Out 🥅🎯
    England into the FIFA-Finals ⏩⏩

Argentina vs Brazil
  Argentina wins 🏆🏆
    Argentina into the FiIFA-Finals ⏩⏩
```

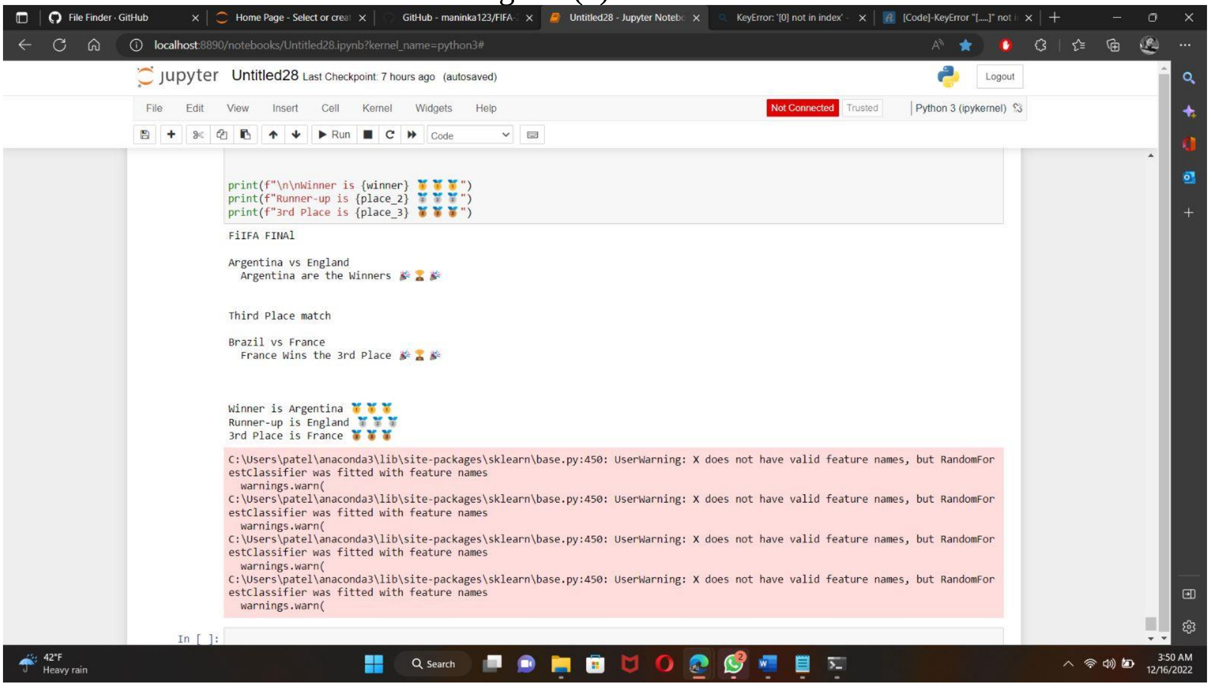
```
C:\Users\patel\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but RandomFor
estClassifier was fitted with feature names
  warnings.warn(
C:\Users\patel\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but RandomFor
estClassifier was fitted with feature names
  warnings.warn(
C:\Users\patel\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but RandomFor
estClassifier was fitted with feature names
  warnings.warn(
C:\Users\patel\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but RandomFor
estClassifier was fitted with feature names
  warnings.warn(
```

```
In [58]: print(f"Teams selected to the Finals - {final_teams}")
         print(f"Teams selected to the Third Place match - {third_place_match_teams}")
```

```
Teams selected to the Finals - ['England', 'Argentina']
Teams selected to the Third Place match - ['France', 'Brazil']
```

```
In [59]: #Finals and Third Place match

         print(f"FiIFA FINAl\n")
         team_1= final_teams[1]
         team_2= final_teams[0]
```

Figure i(b)


File Finder · GitHub
Home Page - Select or creat
GitHub - maninka123/FIFA-
Untitled28 - Jupyter Notebo
KeyError: '[0] not in index'
[Code]-KeyError "[...]" not i
localhost:8890/notebooks/Untitled28.ipynb?kernel_name=python3#
jupyter
Untitled28 Last Checkpoint: 7 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Connected
Trusted
Python 3 (ipykernel)
Run
Code
42°F
Heavy rain
Search
3:50 AM
12/16/2022

```
print(f"\n\nWinner is {winner} 🥇🥇🥇")
print(f"Runner-up is {place_2} 🥈🥈🥈")
print(f"3rd Place is {place_3} 🥉🥉🥉")
```

```
FiIFA FINAl

Argentina vs England
  Argentina are the Winners 🎉🏆🎉


Third Place match

Brazil vs France
  France Wins the 3rd Place 🎉🏆🎉



Winner is Argentina 🥇🥇🥇
Runner-up is England 🥈🥈🥈
3rd Place is France 🥉🥉🥉
```

```
C:\Users\patel\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but RandomFor
estClassifier was fitted with feature names
  warnings.warn(
C:\Users\patel\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but RandomFor
estClassifier was fitted with feature names
  warnings.warn(
C:\Users\patel\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but RandomFor
estClassifier was fitted with feature names
  warnings.warn(
C:\Users\patel\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but RandomFor
estClassifier was fitted with feature names
  warnings.warn(
```

```
In [ ]:
```

Figure i(c)

Conclusion

As we were trying to find the out who can win the match of FIFA world cup 2022 which is taking place in Qatar and they have their final match on 12/18/2022 as per our prediction till now we are in the right direction the final is in between Argentina and France, we predicted Argentina to win the game and France to be the runner up. Hope we get the right prediction.

REFERANCES:

https://www.rookieroad.com/fifa-world-cup/rules-7024950/