Data Analysis and Visualisation of European Soccer Dataset

Abstract

European Soccer has been one of the most followed sport in the world, which hosts a lot of leagues and has been home for many great players the world as ever seen, analyzing such a rich dataset to find some insight and visualize the results.
For this assignment, I am exploring the European Soccer dataset [1] from Kaggle. I am trying to answer the following questions.

- Comparison of different attributes of Lionel Messi and Cristiano Ronaldo in order to conclude who is superior?
- Finding out which leagues are most popular in Europe, top teams in each league, top players and their best attributes?

To visualize these questions, I have used an interactive radar plot [2][3][4] to show attributes of each player and a hover function to get most of the information from the graph and to visualize question 2, I have used a Zoomable Sunburst graph[8] to create a parent-child relationship between different attributes so the user can drill down into data to inspect more. I have used Python and Jupyter Notebook for the purpose of data cleaning and visualizations.

1. Dataset

The dataset used for this assignment is European Soccer Dataset from kaggle[1], the format of the dataset is an SQLite Database which consists of 7 tables, size of the whole data set is 300 MB, few of the tables are as follows:

Dataset 1: Player
- Number of rows: 11100
- Number of Columns: 7

Dataset 2: Player_Attributes
- Number of rows: 184000
- Number of Columns: 42

Dataset 3: Match
- Number of rows: 26000
- Number of Columns: 115

Player table consists of various attributes related to each player and a unique ID which is linked to other tables. A few of the attributes are player_api_id, player_name, birthday, height, weight, etc.

Player_Attributes table consists of about 42 attributes (Potential, Crossing, Short pass etc.) and 18400 rows, which rates each player to those attributes.

Team and Team_Attributes tables consist of 2000 rows and 30 columns combined which gives information about each team and their mentality during the game. A few of the attributes are defencePressure, PlaySpeed, PlayPositioning, DefenderLine etc.

Match table is the main table where all the data is linked to, it has 26000 rows and 155 columns (match_api_id, home_team_api_id, away_team_api_id, home_team_goal, away_team_goal, etc.) and

describes about each player involved in that particular game and his respective country, team, season and league he belongs to and common link between them is the ID field.

League and Country tables consist of various league played in Europe and all the teams taking part in those leagues. Attributes are name, id, country_id - to link country and league table

In aspects of big data, the European soccer dataset contains volume and variety.

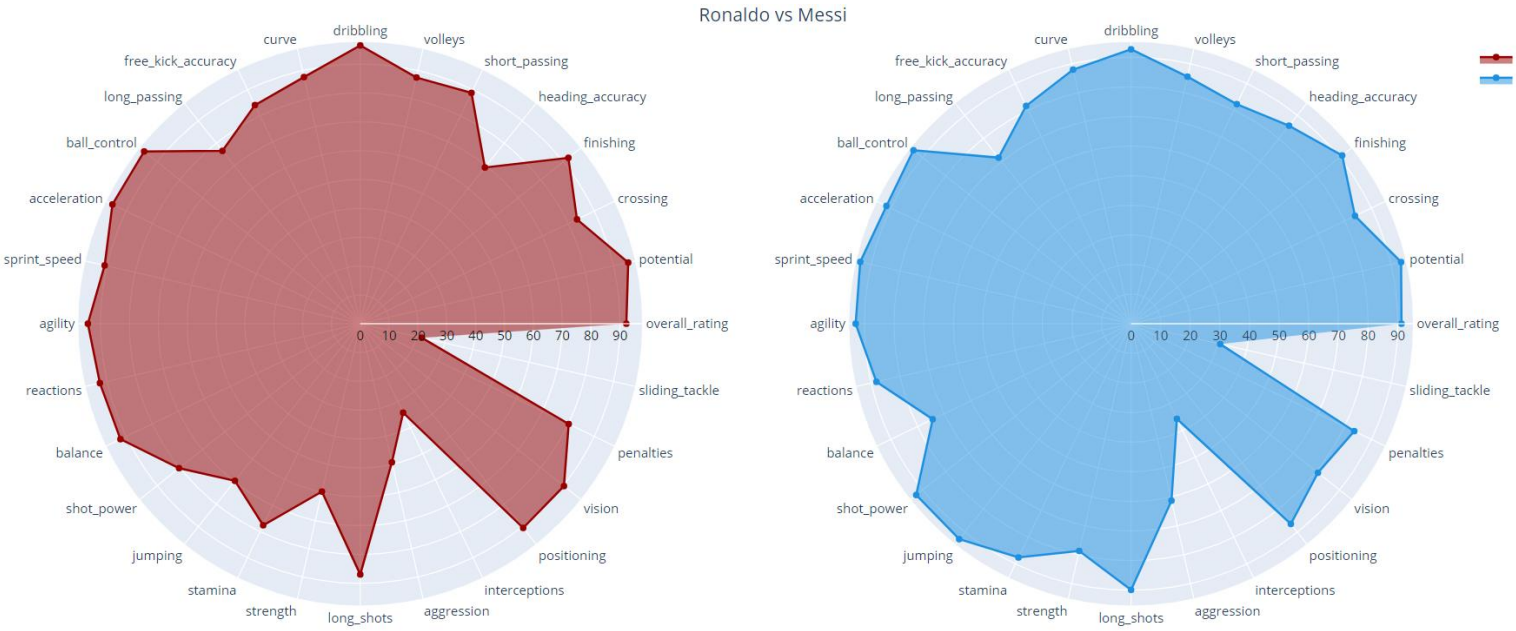
2. Data Exploration, Processing, Cleaning and/or Integration

In order to produce visualization with respect to the questions, I had to read the data from the SQLite file and create multiple data frames to store the data. In order to answer my first question, I had to merge multiple data sources (**Player and Player_Attributes, League and match**) to generate "**MergedPlayer.csv**" and "**data.csv**" files. In order to do that I had to perform a full outer join on "**player_api_id**" and "**country_id**" attribute between datasets respectively and once the dataset was merged I had to clean the dataset and fill out the missing values as well as drop few columns which were not needed for my analysis.
As there were numerous occurrence for each Player in different matches, I had to sort the name field and **club all the rows** of each individual player and recalculate the attribute using their **mean value**, furthermore, I had to drop more columns as we are comparing skills of players hence their height, weight, BMI, age and few more attributes which are not related to player skills were filtered.
For the second question, **country and league data** had to be merged on "**id**" field using full outer join as well as **league and match data** had to be merged on "**country_id**" to create a bigger table containing all the details of matches played between teams, players and there attributes, leagues in which each team participated and a parent-child relationship had to be developed from this values.
Once the data was merged and cleaned, I had achieved a subset from the original dataset containing attributes that had to be visualized

3. Visualization

Visualization 1:

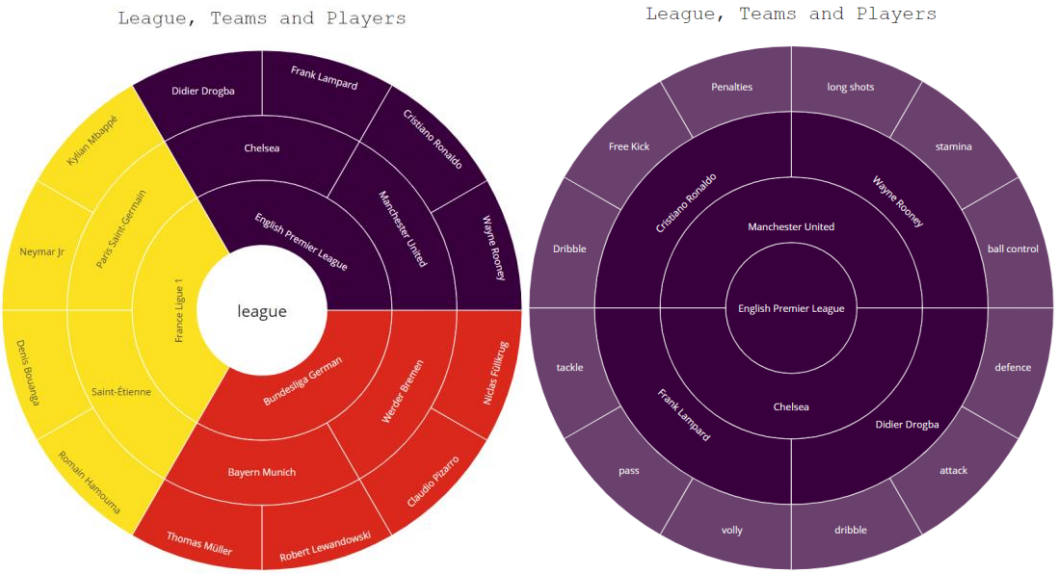
For the first visualization, I am showing a comparison between two players namely Ronaldo and Messi, comparison is done between numerous attributes which contribute to skill of the player, for example, dribbling ability, short pass or long pass ability, free kick, penalties and so on, in order to do this I had to refactor the original data. Each player participated in many matches hence all their appearance had to be taken into account, therefore I had to merge rows in these case, by grouping their names and had to merge their attribute values as well, following this I had to drop few columns as well which didn't fit well into the skill spectrum by plotting a correlation matrix for example, height, weight and date of birth of a player, after all the cleaning I ended up with a data-frame which had players and a mean values of the merged attribute values.



For the visualization, I used a radar chart [4] because I had multi-variant data for each player and I had to showcase each attribute to create a comparison between both the players, the polar radius gives the respective value for that particular attribute with a step count of 10 and its dynamically scalable, polar angle gives various attributes to which the values are mapped.
I chose this color specifically because it represents the player's national colors.
Few major Python libraries and tools used for this visualization are numpy, pandas, IPython, Matplotlib, Plotly, cufflinks, and Jupyter notebook
To add interaction to the graphs I have used iplot() function from plotly to generate the plots, this allowed me to add features like hover function which gives value when the user hovers the mouse over any point on the graph, axis and attributes can be rotated around according to user needs and a zoom-in feature as well, to dig deep into the visualization

Visualization 2:

For the second visualization, I am showing top 3 leagues in Europe depending on the win percentage and using the same technique to find top 2 teams in each league further, I found out top players in those teams by the number of home and away goals they have scored and their top attributes with respect to values.



For this visualization I needed to show granularity and parent-child relationship between each level so I chose zoomable sunburst graph [8], this lets the user drive through the data and focus on granularity of the data, colors which I have chosen are the base colors of each league, which makes it easier for the user to recognize.
Libraries used for this is plotly offline and cufflinks, in order to prepare the data for this graph I had to find total matches played by each team in both home and away occasions and find the win percentage for each team which helped me rank the teams and leagues, once this was done I filtered the data to find top 3 leagues and top 2 teams in each league, to find top players in a team I used the data frame generated from the first plot and filter it with respect to teams. I then built a parent-child relationship between each attribute in the sunburst plot.
To add interaction to the plot, I used iplot() function, which allows hover over function and when the user clicks on any segment in the plot it bursts open or zooms in and makes that sector as a parent and lets the user drill down even more into data. I had to restrict the depth of the plot to 3 to keep it clean.
Burst interaction is specifically useful in this case as it lets the user focus on individual aspects but still keeping rest of the data in hand.
Part of cleaning and analysis was done in R studio for this plot, I used plotly in python to bring this visualization to life and standard libraries like numpy, pandas, SQLite were used as well.

4. Conclusion

After exploration and cleaning of data, I have answered the questions posed in the abstract as follows,

- **Comparison of different attributes of Lionel Messi and Cristiano Ronaldo in order to conclude who is superior?**

  Analysis: from the plot we can observe that the top two player's Lionel Messi and Cristiano Ronaldo attributes have been visualized using a radar plot, which allows direct comparison between different attributes of each player. Just by looking at the plot we can infer that the heading accuracy of Messi is better than Ronaldo, Ronaldo has better balance than Messi, Messi has more shot power, jumping and stamina when compared to Ronaldo, both players play in forward position which explains why their interception and sliding tackle attributes are very low.

  By using tool tip to find values, we can see that Messi plays more aggressively than Ronaldo, Ronaldo has better chances of completing a long pass than Messi by 5%, dribbling ability of both the players are exceptional but Ronaldo has a slight edge of about 4% than Messi. The best skillset of Messi is ball control whereas in case of Ronaldo it's his dribbling ability, by looking at the stats we can conclude that Cristiano Ronaldo is a better play than Lionel Messi

  Certain aspects that could be improved are, the initial and final points are not connected which seems a bit odd, but the plotly library was plotting the points as a scatter plot and since the points were already plotted, I couldn't get it connected.

  I wanted both the graphs in a single plot where they overlap but plotly uses a layer trace to add points on one another when both the attributes of player was plotted, hover function was jinxed ie if the underlying plot was covered by the above plot, hover function was not able to pick the value which was underneath.

- **Finding out which leagues are most popular in Europe, top teams in each league, top players and their best attributes?**

  Analysis: from the plot, we can observe that it shows information about the top 3 leagues i.e. French League, English Premier League and German League which are the major sectors in the plot. A pie chart or a Donut chart could have used to convey this information more precisely but instead, I have chosen a sunburst chart [7] [8].
  which divides this sectors into sub-sectors to embed more data, 2nd level of the sunburst plot gives information about the top 2 teams in each league and the 3rd level in the plot tells us about the top players in each team, by using the interacting elements of the graph, we can drill down into each sector by clicking on them which reveals the last level which contains top attributes for each player

  Certain aspects could have been improved such as embedding values to each sector instead of dividing them into equal sectors but the data was not sufficient to map values to each sub-sector hence implementing was a challenge.

  Few effects such as fading in and out while moving between different levels and overall smoothness of the transition could have been improved, as this graph is a concept from D3.js Zoomable Sunburst [8] which is natively developed using SVG elements which provide this smooth transitions.

References

[1] European Soccer Database, Available Online: https://www.kaggle.com/hugomathien/soccer

[2] Radar Chart in D3.js Blog, Available Online: https://blockbuilder.org/Ananda90/8269def4e60b17d57d358b2e8219f62d

[3] Radar Chart Redesign in D3.js, Available Online: http://bl.ocks.org/nbremer/21746a9668ffdf6d8242 http://bl.ocks.org/tpreusse/2bc99d74a461b8c0acb1

[4] Radar Charts in Python, Available online: https://plot.ly/python/radar-chart/

[5] D3- Data Driven Documents, Available Online: https://d3js.org/

[6] Plotly Python Open Source Graphing Library, Available Online https://plot.ly/python/

[7] Sunburst Charts in Python, Available online https://plot.ly/python/sunburst-charts/

[8] Zoomable Sunburst, Available online https://observablehq.com/@d3/zoomable-sunburst