

Project Report

# European Soccer Database

---

Johnpaul Kosisochukwu Nwagwu

25th March, 2021

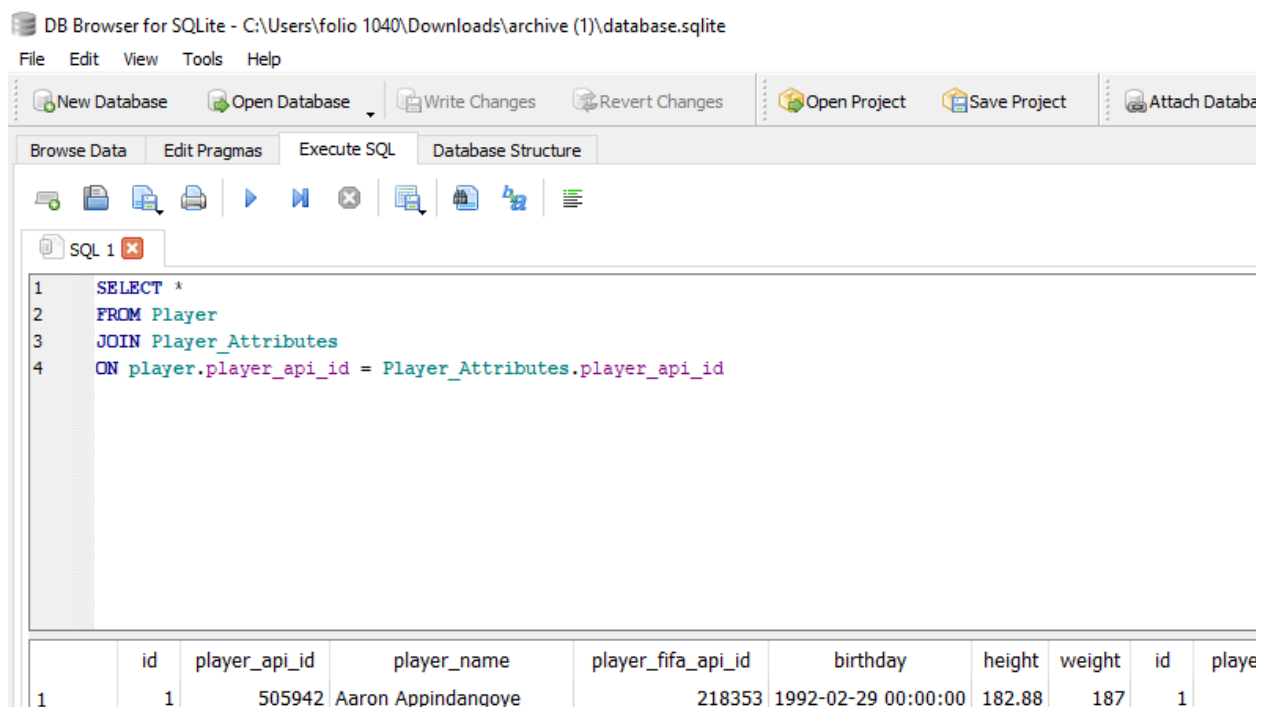
## Dataset

The European Soccer Dataset was gotten from Kaggle. It contains data about players, teams, soccer matches played and individual player attributes. It is a huge dataset with 7 tables containing multiple columns and entries and is typically suited for a Data Analysis and Machine Learning Project.

There are tons of questions that could be asked with this Dataset from varying angles and degrees.

For my analysis, I focused on the Player Table and Player Attributes Table. Since the Data is stored in .sqlite on Kaggle. I installed SQLliteserver on my computer and joined the two tables using SQL.


### SQL CODE IN DB BROWSER FOR SQLite



The screenshot shows the DB Browser for SQLite application. The title bar indicates the file path: C:\Users\folio 1040\Downloads\archive (1)\database.sqlite. The menu bar includes File, Edit, View, Tools, and Help. The toolbar contains icons for New Database, Open Database, Write Changes, Revert Changes, Open Project, Save Project, and Attach Database. The main window has tabs for Browse Data, Edit Pragmas, Execute SQL, and Database Structure. The Execute SQL tab is active, showing a SQL query in a text area. Below the query, the results are displayed in a table with 10 columns: id, player\_api\_id, player\_name, player\_fifa\_api\_id, birthday, height, weight, id, and player. The first row of data shows a player with id 1, player\_api\_id 505942, player\_name Aaron Appindangoye, player\_fifa\_api\_id 218353, birthday 1992-02-29 00:00:00, height 182.88, weight 187, and player 1.

```
1 SELECT *
2 FROM Player
3 JOIN Player_Attributes
4 ON player.player_api_id = Player_Attributes.player_api_id
```

	id	player_api_id	player_name	player_fifa_api_id	birthday	height	weight	id	player
1	1	505942	Aaron Appindangoye	218353	1992-02-29 00:00:00	182.88	187	1	1



The player table contains 7 columns: id, player\_api\_id, player\_name, player\_fifa\_api\_id, birthday, height and weight

The player\_Attributes table contains 42 columns: id, player\_fifa\_api\_id, player\_api\_id, date, overall\_rating, potential, preferred\_foot, attacking\_work\_rate, defensive\_work\_rate, crossing, finishing, heading\_accuracy, short\_passing, volleys, dribbling, curve, free\_kick\_accuracy, long\_passing, ball\_control, acceleration, sprint\_speed, agility, reactions, balance, shot\_power, jumping, stamina, strength, long\_shots, aggression, interceptions, positioning, vision, penalties, marking, standing\_tackle, sliding\_tackle, gk\_diving, gk\_handling, gk\_kicking, gk\_positioning, gk\_reflexes.

The tables are combined in SQL and downloaded. Then loaded to Jupyter Notebook



## Questions Posed

First, I answer simple questions about our data.

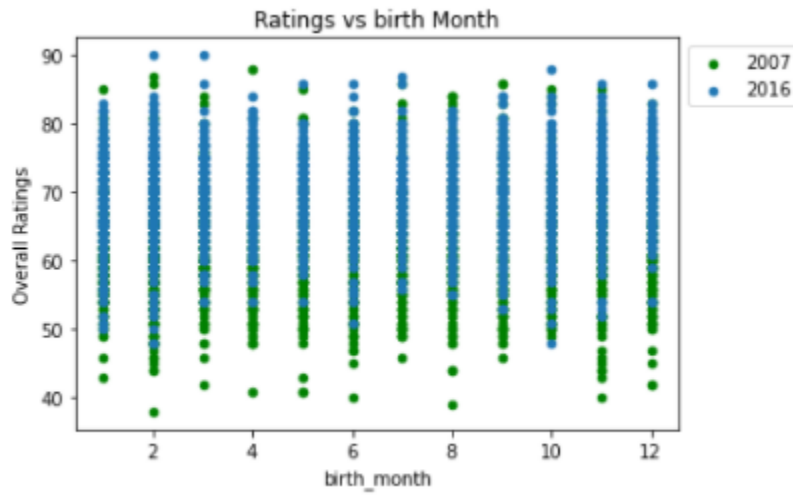
- 1) The tallest player and height
- 2) The shortest player and height
- 3) The most improved player rating from 2007 to 2016
- 4) The player with the most potential in 2016.
- 5) Distribution of left and right-footed players
- 6) Do players born in a specific month or year seem to have a better overall rating?
- 7) Do left-footed players have better potential?

## Steps that were taken to investigate the Dataset

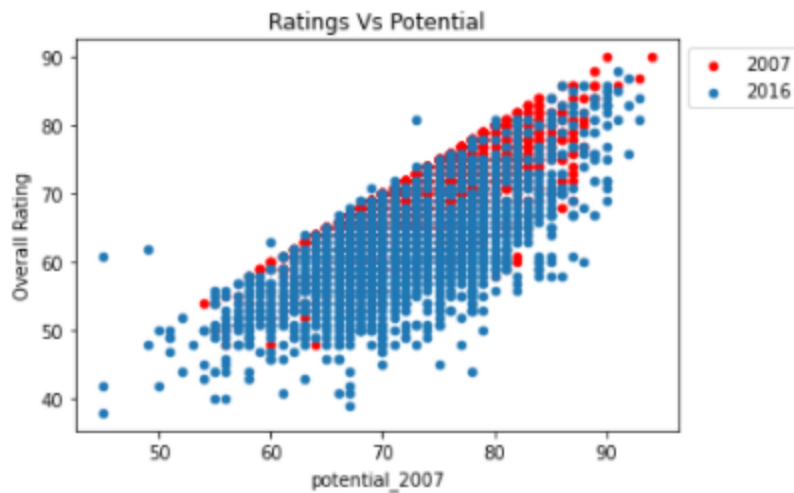
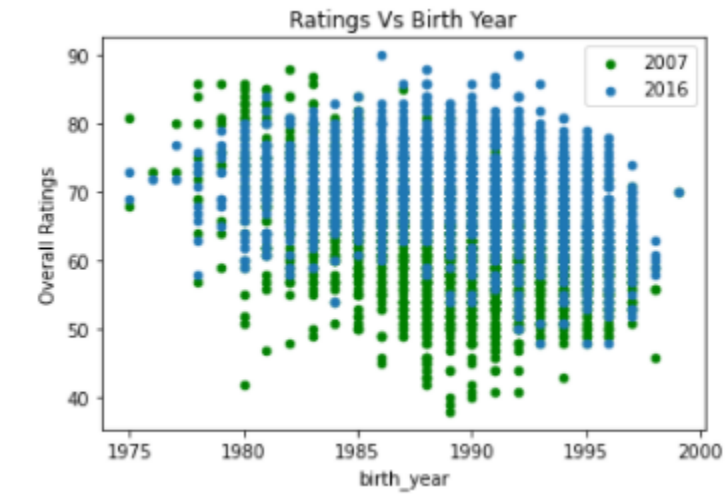
### DATA WRANGLING

- 1) First to make our analysis simpler from start. We will drop columns that are not part of our analysis.
- 2) Convert date column to date time format and then extract years and months and drop original columns
- 3) Convert birthday column to date time format and then extract years and months and drop original columns
- 4) Check count of unique values for each year. Our interest is years 2016 and 2007
- 5) Select data for years 2016 and 2007 and pass to new\_dataFrames
- 6) Inspecting the data. We notice that there can be more than one overall rating per player due to the rating being awarded at different months. We further inspect the date\_month column for the two years to find a basis for our analysis
- 7) Since month 2, February is common for both years. We select data for month 2 to make our analysis easier and more direct.
- 8) Merge datasets with common attributes.
- 9) The value\_counts() method gives us the idea that missing values are present since the row counts are unequal. We check for the number of missing values and drop them
- 10) Rename columns, drop date\_month column and change data types. We don't need date month columns anymore since all the data is for February i.e 2
- 11) Check for duplicates and remove them

## Exploratory Data Analysis And Results



There appears to be no correlation between overall ratings and birth month



From the graph of overall ratings against birth year, There appears to be no obvious directional movement in the graph and so therefore no obvious correlation between ratings and birth year

We notice that from the graph of Potential against Ratings. An increase in Ratings is associated with an increase in Potential. We can say that they are both correlated

\_\_\_\_\_





## Conclusion

The tallest player is Vanja Milinkovic-Savic with a height of about 203

The shortest player is Bakari Kone with a height of about 162

The most improved player rating from 2007 to 2016 is Lamine Kone

The player with the most potential in 2016 is Neymar

The distribution of left and right-footed players is unbalanced. There are 1529 right footed players and 392 left footed players

Birth month and Year is uncorrelated with Overall Ratings

Potential is Correlated with Overall Ratings

Left footed players have better potential on Average

Other Areas to Explore

- 1) Relationship between overall\_rating and potential for left-footed players?
- 2) Which player attributes contribute more to the overall rating?
- 3) Does player weight affect their overall rating?

And so on. The data is really broad and so there are a lot of things to explore. I'd love to do an ML project on this going forward

Limitations to Data

- 1) Data doesn't reflect all the professional soccer players at any year as there are some missing players and information