

Predicting European Soccer Matches

Sahil Goyal

Machine Learning Engineer Nanodegree Capstone

Abstract

Numerous tactics, twenty two players on the field, hearts and passions of the people around the world, a billion dollar industry, one beautiful game: Football. Football is by far the most popular and interesting (a personal opinion) that is out there. Through this capstone project, I present and propose my findings to predict the result of a football match, using a dataset I found on Kaggle that contains data for European Football Matches. I also present my results from a mini-project I did to cluster players into the position on the football field they will be most suited for.

Keywords: Football, Machine Learning, Feature Engineering

I. Introduction

Football (or soccer) is the biggest global sport and is a fast-growing multi-billion dollar industry; with an estimate of 27 billion dollars in terms of annual revenue for the football clubs [1].

Prediction of soccer matches is a tough problem. Predicting the exact scoreline is a near impossible task. On the field often defies on paper. So much so, in the 2011-12 season of the English Premier League (EPL), it was not known who the champions would be until the last five minutes of a nine month long season [3]. In the 2015-16 season of EPL, at the beginning of the season it was more likely that Kim Kardashian would become the US president, than the eventual team Leicester winning the title [4].

With more and more money pouring into the sport, the betting industry for predicting the outcome of matches is worth a billions pounds every year[2]. The ridiculous amount of money can be summed in the chart below, that shows the amount of money clubs have paid to secure a football player's services (I being a mere mortal find myself counting the number of zeros in these huge sums of money).

The game involves emotional factors such as the passion of the crowd, how pumped the players are that day, players' personal lives; factors which are beyond measure. In this project, I wish to tackle the problem using factors we can indeed measure.



Figure 1: The most expensive player in the world with time

I.I. Personal Motivation

I would divide my motivation to do this project into three reasons:

- **My interest and domain knowledge** in this field. I follow European soccer with passion and interest. I would like to use my domain knowledge to come up with questions interesting to me, and help me solve those questions.
- **Feature selection and generation**, the biggest take away for me from this nanodegree course is the importance of feature selection, and finding correlations between features. The project on customer segments gave me a chance to apply some techniques such as PCA to reduce dimensionality; and I would like to apply them to this problem, in conjunction with my domain knowledge.
- **Visualizing results and insights**, I wish to draw some insights to the game of soccer through this project, and want to use some visualizations to analyze the results from my analysis.

I.II. Problem Statement

I try to find answers to the following problems through this project:

- The **holy grail** is predicting the outcome of a football match, as mentioned in the kaggle link for the dataset I intend to use [5].
- The **clustering** of players into their positions on the field; such as midfielder, goalkeeper, midfielder and attacker; based on the FIFA skill data given in the database.

Metrics to evaluate

- For the match outcome prediction is the accuracy of predicting results. More details are given in the section II.II on the base naive predictor.
- For the players clustering project VI, I pick some famous players that I know the type and their position on the fields of; and analyze if I cluster them properly.

I.III. Datasets and Inputs

The dataset I will be using for this task is the European soccer database over at Kaggle [5]. The database contains data from 25,000 matches; 10,000 players for the years 2008-2016, spread over 11 European countries. The matches are defined well in terms of data, with information about fouls, shots on target, possession etc. The dataset contains players' skills and ratings, as evaluated by the makers of the popular video game *FIFA*.

I.IV. Overview of time spent

Before diving in detail over the solution and the steps taken, I would like to given an overview of the time spent in the various steps of the solution the problem.

Task	Percentage of total time spent
Data Loading and Familiarity	10%
Feature Generation using domain knowledge	50%
Model Fitting and Optimization	30%
Results, Visualizations and Discussion	10%

Table 1: Overview of time spent on various steps of the project

II. Exploring the data

II.I. Reading the data

The data is in the form of a SQLite database. I used python's *sqlite3* module to load and read the database. The database contains the following tables:

- Table with attributes and related data
 - ***Match*** contains the data for the matches from the database with information such as the country it was played in, between the teams it was played, the season and the date it was played, the players belonging to the various teams and of course the result of the match, among other data.
 - ***Player_Attributes*** contains the data for the players; and data from FIFA such as the player's overall rating, players skills such as shot accuracy, passing, defending, tackle skills etc.
 - ***Team_Attributes*** contains the data for the teams'; such as speed of the build play, defense pressure, passing in the build up play etc.
- Metadata and lookup tables such as ***Player, Team, Country, League*** which are mostly lookup tables mapping things such as player id, team id, league id, country id into their respective names.

Some of the pre-processing to get the relevant data is mentioned here. The reason for this pre-processing will be clearer in the section on Feature generation [III](#).

- **Extracting relevant columns for Matches** The matches table contains a lot of data, that I considered not to be highly relevant; such as betting data; player formations etc. The columns that were selected were the player ids from both teams, the team ids, the season it was played in.
- **Extracting relevant columns for Players** To classify players into their relevant skills, columns corresponding to different skills were selected. I also selected the overall rating of each player.

- **Generating the output labels** The output label (the match result) was generated using the columns *home_team_goal* and *away_team_goal*; which give us the goals scored by the home and the away team respectively. Output label for a match was generated to have one of these three possible values:

1. *HOME_WIN* The label when the home team has score more goals than the away team.
2. *DRAW* When both teams have scored the same number of goals.
3. *AWAY_WIN* When the away team has scored more goals than the home team.

Therefore, this problem is a **multi class classification** problem.

II.II. Defining the base classifier: The home advantage

Now that we have our labels defined, let us define the baseline classifier. Home advantage is real, as mentioned in the Kaggle link [5], and [6]. I analyze the average home advantage across all matches, and also analyze it by country.

I found that constantly predicting a Home Win will get you right about 46% of the time. Predicting an Away Win will get you right about 29% of the time. Figure 2 shows the home advantage across different countries. As evident, every league has a significant advantage for the team that is playing at their home stadium.

On the Kaggle page, the author of the dataset mentions he gets it right with an accuracy of about 53%; this is another reference to compare my model's performance against.

II.III. Players' ratings histogram and visualization

A histogram visualization to view player distributions in terms of ratings will be helpful. I will be using this to gain intuition for features such as 'players_in_top_X_percent'. Sometimes a single **star player** can win a match for their team [7]. Sometimes, a single mistake can cost a team the match. These moments of brilliance, and the grief of mistakes, is what makes football so unpredictable and beautiful. Figure 3 shows a histogram of player distribution by ratings.

We will be focusing on the players that are at the extreme ends of this histogram; more to be discussed in the section on Features [III](#).

name	percentage_home_win
Belgium	46.875000
England	45.723684
France	44.703947
Germany	45.220588
Italy	46.635731
Netherlands	47.834967
Poland	45.312500
Portugal	44.249513
Scotland	41.666667
Spain	48.848684
Switzerland	45.710267

Figure 2: Home advantage across different European leagues

II.IV. Handling Bad Data

The Kaggle page does say that some input values are missing, and indeed it does. I dropped the matches where the values are missing (*NaN*). For this I use the *dropna* method in pandas [8]. I certainly do not want to blindly check for validity of all columns in the match; a lot of it is information I don't think is necessary for prediction (bettting data, corners for example); atleast for the problem of determining match results. Therefore I define the columns *important* to me. Luckily, even if I drop the data if *any* of the player ratings is absent, I do not lose a lot of data; dropping 4605 matches out of a total of 25000 matches.

III. Feature generation

This is the part of the project which was the most fun, and I got to use my **Domain Knowledge**. Must admit I was a bit nervous about generating features (what if they are not relevant and give me bad results!); more on the results in the section on model fitting [IV](#). I divide my features into three main categories, and shine light on the rationale and the thinking behind generating these features.

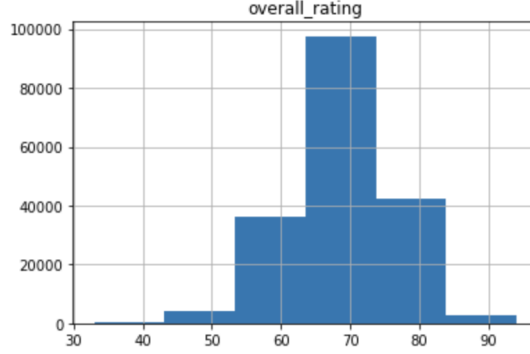


Figure 3: Distribution of players by their ratings

III.I. Players' based features

These are the features based on the players' attributes (skills) and FIFA ratings. Each match contains the information about the line up of both the home and the away team; listing the eleven players that played for each team. These player ids' can then be used to extract information from the players' attributes table and used to generate relevant features such as:

- Ratings for the teams involved in the match

On the Kaggle page, I saw that some contributors took the average rating of the players; defined for both home and away[9]. I chose a different approach here; instead of taking average across all players; based on my domain knowledge I split players based on the position in their teams.

I divided the players into four types- *attacker*, *defender*, *midfielder* and *goalkeeper*; based on a naive algorithm to classify players' based on their skills. The four skills I selected were '*finishing*', '*sliding_tackle*', '*gk_reflexes*', '*short_passing*' from the players' table to classify players into their skills; based on a simple algorithm: if the max skill is *finishing*, then the player is an attacker; if the max skill is *sliding_tackle*; the player is a defender, and so on.

The validation of this algorithm was done for famous players that I know the type of; and it works pity good. This is also the subject of my analysis for clustering players VI.

Based on this naive *clustering* of players; I got four features for every team: the average ratings for their players in attack, defense, midfielder and goalkeeping: Note that goalkeeping is a single player; albeit very important [10], so I kept this as a separate feature. Note that to get the rating of a player, I get his closest rating to the match date, as every player has yearly entries for his rating.

Datatype These eight features ($4*2$); are positive float values in the range 0-100.

- Exceptionally good (and bad players)

It is often seen that a football match is decided by a single player's moment of brilliance, or a player's costly mistake. I visualized the players' ratings in Figure 3; based on this distribution I defined a top player in a team as any player with a rating greater than 80; this corresponds to roughly the top 1% of the players in the histogram. I similarly defined the bottom players as the ones with rating below 50; this corresponds to roughly the bottom 1% of the players in the histogram.

For both teams, the number of top and bottom players was calculated.

Datatype These four features ($2*2$); are positive integer values in the range 0-11.

III.II. Form Stats features

A team's current form in this season, its historic home record, its away record, head to head with the team its playing against affect players' confidence; and have a significance in that it tells us the trend of a *stronger club between the two*. A club that wins against another 90% of the time is likely to win again. Using the data given, I generated the following features in terms of percentages:

- Home Team
 - All time home record - The percentage of home matches that the home team wins historically.
 - This season home record - The percentage of home matches that the home team has won this season.
- Away Team

- All time away record - The percentage of away matches that the away team wins historically.
- This season home record - The percentage of away matches that the away team has won this season.
- Away record at this ground- This is a feature I thought of through following football. If historically a club has won at a particular ground many times in the past, it's fans and players are pumped up; causing *upsets*. These features tells us the percentage of away matches that the away team has won at this particular home team.

- Head To Head

- Head to head draws- The percentage of head to head matches between the two teams that have resulted in draws.
- Head to head home wins- The percentage of head to head matches between the two teams that have resulted in the home team winning.
- Head to head home loss- The percentage of head to head matches between the two teams that have resulted in the home team losing/away team winning.

An example for the home team win rate of *FC Barcelona*, one of the world's most famous clubs is given in Figure 4.

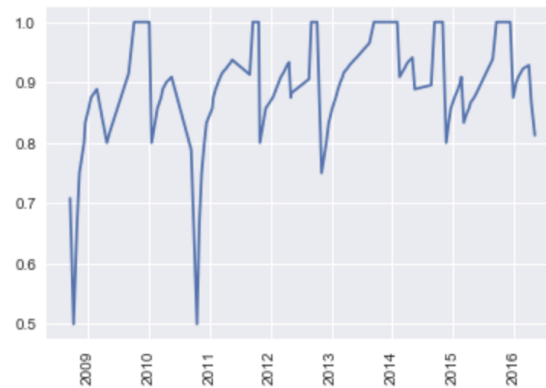


Figure 4: FC Barcelona's home win rate with time (2008-2016)

Datatype These nine features are float percentages in the range 0-100.

III.III. Form Guide Features

I defined the form guide for a team as an aggregation of its last five results. So if a team has won the last two matches, lost the two before that and won the one before those; its form guide will be **WWLLD**. A win corresponds to the letter W, loss to the letter L and draw to the letter D. Note that there are a total of 3^5 possible values for this variable. I calculated this for both teams for the current season. **Datatype** These two features are categorical variables.

Filling missing values

You might have observed it, but some matches might not have enough data to calculate the features I suggested. For example: how should I calculate the head to head for the two teams if they have never played before? How should I get the form guide for a team, if this is only the 2nd match of the season? (I need five matches to define the form guide)

I thought about using sklearn's *Imputer* function; however then decided to implement my own code to do so; in order to fill those *NaN* values more in accordance with my domain knowledge. For example:

- Head to head wins/draw/loss : Take all head to head matches, and replace np.nan with the mean of all non nan values. It is possible that some teams have NEVER played before, in which case I will just randomly assign a value of 0.33 to each of the three events (equal probability of happening).
- Away team's win rate at this ground: Find all samples where this team played away at this home team's ground; and take average. If this team has never played here; just take the average of its all time away record. Note that this is rare, but possible.
- Form guide: Replace the NaN values with the most common form guide for that team during that season.

Lambdas and Partial

In terms of generating these features, the dataset is huge (25000 matches); I educated myself on the use of *pandas'* *lambda* and *partial* functions to process the dataset match by match. This was a fun exercise, and as a programmer, I loved being able to do this after an initial steep learning curve.

IV. Model fitting and optimization

IV.I. Bad Performance on draws and response to it

V. Results and visualizations

VI. Mini Project: Players Clustering

VII. Future Work

VIII. Conclusion, some personal musings and thank you Udacity

IX. Links to code and notebooks

X. Acknowledgment

Mention notebooks used, the udacity visuals code

XI. References

1. [Call for Papers for a Special Issue in the Springer Journal](#)
2. [Football betting - the global gambling industry worth billions](#)
3. [City beat QPR in the last five minutes](#)
4. [Things likelier than Leicester winning the league](#)
5. [European Soccer Database](#)
6. [How much does home advantage matter in soccer](#)
7. [Messi single-handedly guides Argentina to the World Cup](#)
8. [Pandas Drop NaN](#)
9. [Match outcome prediction in football](#)
10. [David De Gea wins the match by himself](#)
11. [sklearn preprocessing Imputer](#)