Predict soccer match outcomes based on player's season stat with Machine Learning

Machine Learning, Fall 2020

Hanyang University

SeongCheol Kim

2020178469

Abstract

Recently, the application of machine learning to various fields is increasing. The sports field is also one of them. I want to predict the outcome of the football, the most famous sport in the world. Most of the previous studies predicted the outcome of the game through the data of the game itself. However, through this project, I am trying to predict the outcome of the match based on the player's season stats. For this, a total of six machine learning methods were used: Decision Tree, Gradient Boost, Ada Boost, Random Forest, Support Vector Machine, and Multi-Layer Perceptron, and the game results were predicted with an accuracy of 40% to 55%. Through this, it was suggested that the match result can be predicted based on the player's season stat.

1. Introduction

Soccer is the most popular sport in the world. The Super Bowl, which is said to have a large number of viewers, is also about 100 million, but the World Cup final is over 1.1 billion.

For this project I chose The English Premier League. Because which is the most-watched professional soccer league on the planet, with an estimated viewer figure of 12 million people per game. EPL consists of 20 teams in one season, with a total of 380 games played, each team plays every other team twice, once at home and once away.

In this project, I want to predict the outcomes of soccer matches based on player's season stats using machine learning algorithms. For this, I used match outcome data and individual player stat data of 5 seasons. The five seasons selected are from 2015-16 to 2019-20. 90 features to be used in the model were created through pre-processing of match result data and player stat data, and data of a total of 1900 matches were used for the project.

The predicting features will be fed as inputs to Machine Learning classifier algorithms such as Decision Tree (DTs), Gradient Boosting (GB), Ada Boosting (AB), Support Vector Machine (SVM), Random Forest (RF) and Multi-Layer Perceptron (MLP). The prediction is in one of three classes for each game, with respective to the home team: win, draw, or loss. To improve model performance, I implement various techniques such as Recursive Feature Elimination (RFE), Wrapper Method (WM) and Univariate Selection for features selection and cross-validation for model evaluation.

This paper proceeds as follows: In section 2 the literature related to prediction of football outcomes, section 3 describes the proposed method in depth, section 4 shows experimental results, section 5 discussion, and finally shows the conclusions of section 6 and future work.

2. Literature review

(Zaveri, N. et al, 2018) predicted the results of the Spanish La Liga game in season 5 based on the final score, the starting 11 players, the substitutes, and the names of probable goal scorers. They predicted Logistic Regression 63.94%, Random Forest 61.53%, ANN 63.1%, Linear SVM 58.25, Naïve Bayes 58.63% with accuracy. [1]

(Groll, A. et al, 2019) predicted scores of soccer matches by applying combine two different ranking methods together with several other predictors in a joint random forest approach. To do this, they used data from the previous two FIFA Women's World Cups 2011 and 2015, and finally, based on the resulting estimates, the FIFA 2019 Women's World Cup was iteratively simulated and showed the odds of winning for all teams. [2]

In (Capobianco, G. et al, 2019), a new feature set (related to the match and to players) for model a soccer match was proposed. In addition, using data from the Italian Serie A League for the 2017-2018 season, it showed a precision of 0.857 and a recall of 0.750 in match predictions via Random Forest. [3]

(Baboota, R., & Kaur, H., 2019) build a generalized prediction model to predict the outcome of the English Premier League. Using a gradient boosting model, it was 0.2156 in the Rank Probability Score (RPS) metric over the 6-to-38-week game weeks of the English Premier League in the 2014-15 and 2015-16 seasons. [4]

In (Liti, C., Piccialli, V., & Sciandrone, M., 2017), to solve the difficulty of predicting soccer game outcomes due to the randomness of data and the presence of complex interaction factors, the imbalance problem with less Away wins was adjusted. By reducing the number of features as much as possible, the test accuracy of Naive Bayes was 0.4340, LibSVM C was 0.3920, LibSVM ν was 0.3640, and RBFClassifier was 0.4360. [5]

3. Methodology

3.1 The dataset

This section outlines the three types of data you can use to model and predict the English Premier League.

The first data set is the data on the game result, and includes the season, game date and time, home team name, away team name, home team score, away team score, and game result. The second data set is the stats of individual players in the five major European leagues, including player name, team name, league, games, minutes played, goal, non-penalty goal, assist, expected goal, expected assist, non-penalty expected goal, expected goal per 90 minutes, expected assist per 90 minutes, non-penalty expected goal per 90 minutes, position, shots, key passes, yellow card, red card, expected buildup and expected goal chain. From this dataset, only data of EPL players were selected and used. The last dataset is EPL's individual player stats: Season, Name, Position, Appearances, Clean sheets, Goals conceded, Tackles, Tackle success %, Last man tackles, Blocked shots, Interceptions, Clearances, Headed Clearance, Clearances off line, Recoveries, Duels won, Duels lost, Successful 50/50s, Aerial battles won, Aerial battles lost, Own goals, Errors leading to goals, Assists, Passes, Passes per match, Big chances created, Crosses, Cross accuracy %, Through balls, Accurate long balls, Yellow cards, Red cards, Fouls, Offsides, Goals, Headed goals, Goals with right foot, Goals with left foot, Hit woodwork, Goals per match, Penalties scored, Freekicks scored, Shots, Shots on target, Shooting accuracy %, Includes Big chances missed, Saves, Penalties saved, Punches, High Claims, Catches, Sweeper clearances, Throw outs, and Goal Kicks.

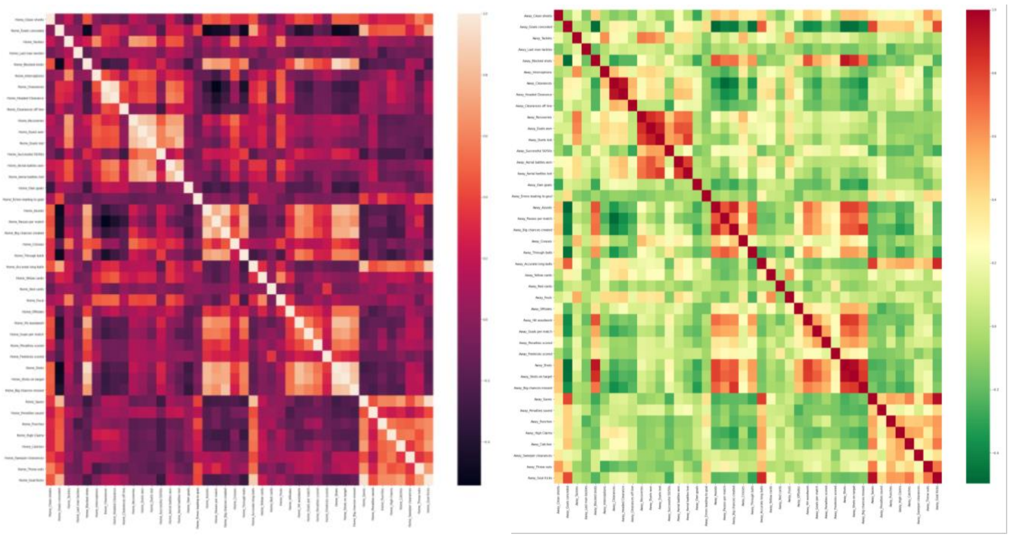
3.2 Preprocess

As I said before, I predict the outcome based on the player's stats. To do that, I had to get the team's expected goal or the team’s performance value in some way. But In the match result data, I only use information about each team's goals and which team won. So, there is no information on the lineup and formation. So, to solve this problem, I looked for the most used formation in the EPL in the last 5 seasons. They were 4-3-3, 4-2-3-1, 3-4-3, etc. Simply put, the value of each position of the team was obtained by weighting the value of each player stats by the probability of participation, and based on that, the value of the team was obtained by multiplying the number by position.

Briefly explaining the process for this, I first divided the cumulative data of each player's data by the number of games played and converted it into an average stat per game. Also, during this process, data that can directly show the game result such as goals and assists were deleted. After that, the value of each team's position was calculated, which was multiplied by other stats using the probability of each player's appearance per game and the average playing time per game as weights. Then, based on the formation information, the team's performance value was calculated. For example, if I use a formation like 4-3-3 in my analysis, the team's value will be the goalkeeper's value multiplied by 1, the defender's value multiplied by 4, the midfielder's value multiplied by 3, and the attacker's value multiplied by 3 and then summed up.

Finally, these calculated values were combined with the data for each match. So, 90 data functions to be used for the analysis.

3.3 EDA



(a) Home Team stat correlation (b) Away Team stat correlation

Figure 1.

To fit our model appropriately, I must consider an X matrix of values that has no information regarding our target variables. For this I did correlation analysis like Figure 1. So, during EDA I dropped the features that might provide the model information on y, i.e., 'Season', 'HomeTeam', 'AwayTeam', 'Home_score', and 'Away_score' Etc. along with our target.

In addition, it was checked whether the match result to be used as the target of the model was skewed to one side. The result is the same as figure 2, where 1 means the home team wins, 0 means a draw, -1 means the away team wins. The home team has some advantage, but the data imbalace is not serious, so this was not modified.

Finally, I analyzed the importance of other features to the target, and the results were interesting. The reason is that, as expected, 'Shots on target', which has the greatest influence on scoring, appeared as a very important feature, but the importance of 'penalties saved', which is often considered crucial for winning or losing a game, was the lowest.

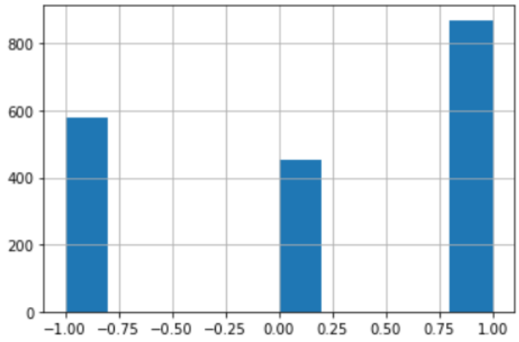


Figure 2.

3.4 Feature selection

Because of the randomness of data and the existence of complex interacting factors, the prediction of soccer match results could be accepted as a hard three-class classification. Through data preprocessing, unnecessary data such as game date and time and stats which is direct to game outcome were dropped, but still 74 features remain. If all of these are used, there is a possibility that the model will overfit, and the time is too long, so we reduced the number through feature selection. Feature selection methods used include as Recursive Feature Elimination (RFE), Wrapper Method (WM) and Univariate Selection. The features selected through each method are shown in Table 1. Univariate Selection was not possible with normalized data because data must be positive.

Mathod	Selected features with normalization	Selected features without normalization
Recursive Feature Elimination	'Home_Passes per match', 'Home_Big chances created', 'Away_Clean sheets', 'Away_Shots', 'Away_Shots on target'	'Home_Passes per match', 'Home_Big chances created', 'Away_Big chances created', 'Away_Shots', 'Away_Shots on target'
Wrapper Method	'Home_Clean sheets', 'Home_Aerial battles lost', 'Home_Passes per match', 'Home_Big chances created', 'Home_Through balls', 'Home_Hit woodwork', 'Home_Shots on target', 'Away_Clean sheets', 'Away_Blocked shots', 'Away_Clearances', 'Away_Headed Clearance', 'Away_Passes per match', 'Away_Big chances created', 'Away_Through balls', 'Away_Red cards', 'Away_Hit woodwork', 'Away_Shots', 'Away_Shots on target', 'Away_Big chances missed'	'Home_Clean sheets', 'Home_Blocked shots', 'Home_Passes per match', 'Home_Big chances created', 'Home_Fouls', 'Home_Hit woodwork', 'Home_Shots', 'Away_Clean sheets', 'Away_Blocked shots', 'Away_Clearances', 'Away_Successful 50/50s', 'Away_Aerial battles lost', 'Away_Errors leading to goal', 'Away_Passes per match', 'Away_Big chances created', 'Away_Through balls', 'Away_Red cards', 'Away_Shots', 'Away_Shots on target', 'Away_Big chances missed'
Univariate Selection		'Home_Passes per match', 'Home_Shots', 'Home_Throw outs', 'Away_Shots', 'Away_Shots on target'

Table 1

4. Results

Classifier	Feature selection	Accuracy	AUC
Decision Tree	Recursive Feature Elimination	0.44 (+/- 0.09)	0.57 (+/- 0.07)
	Wrapper Method	0.42 (+/- 0.08)	0.56 (+/- 0.06)
Gradient Boost	Recursive Feature Elimination	0.50 (+/- 0.08)	0.65 (+/- 0.07)
	Wrapper Method	0.48 (+/- 0.10)	0.63 (+/- 0.10)
Ada Boost	Recursive Feature Elimination	0.54 (+/- 0.05)	0.66 (+/- 0.05)
	Wrapper Method	0.55 (+/- 0.07)	0.65 (+/- 0.07)
Random Forest	Recursive Feature Elimination	0.51 (+/- 0.06)	0.55 (+/- 0.05)
	Wrapper Method	0.51 (+/- 0.09)	0.64 (+/- 0.08)
Support Vector Machine with linear kernel	Recursive Feature Elimination	0.55 (+/- 0.05)	0.65 (+/- 0.07)
	Wrapper Method	0.55 (+/- 0.07)	0.66 (+/- 0.05)
Multi-Layer Perceptron	Recursive Feature Elimination	0.54 (+/- 0.04)	0.68 (+/- 0.06)
	Wrapper Method	0.53 (+/- 0.04)	0.66 (+/- 0.05)

Table 2: Results with feature normalization (CV=10)

Classifier	Feature selection	Accuracy	AUC
Decision Tree	Recursive Feature Elimination	0.43 (+/- 0.08)	0.56 (+/- 0.07)
	Wrapper Method	0.43 (+/- 0.10)	0.55 (+/- 0.06)
	Univariate Selection	0.42 (+/- 0.06)	0.56 (+/- 0.04)
Gradient Boost	Recursive Feature Elimination	0.49 (+/- 0.10)	0.64 (+/- 0.10)
	Wrapper Method	0.49 (+/- 0.07)	0.64 (+/- 0.07)
	Univariate Selection	0.48 (+/- 0.08)	0.64 (+/- 0.08)
Ada Boost	Recursive Feature Elimination	0.55 (+/- 0.04)	0.65 (+/- 0.05)
	Wrapper Method	0.55 (+/- 0.05)	0.65 (+/- 0.05)
	Univariate Selection	0.55 (+/- 0.05)	0.65 (+/- 0.06)
Random Forest	Recursive Feature Elimination	0.51 (+/- 0.08)	0.64 (+/- 0.09)
	Wrapper Method	0.51 (+/- 0.09)	0.64 (+/- 0.08)

	Univariate Selection	0.49 (+/- 0.08)	0.64 (+/- 0.07)
Support Vector Machine with linear kernel	Recursive Feature Elimination	0.55 (+/- 0.06)	0.67 (+/- 0.06)
	Wrapper Method	0.55 (+/- 0.06)	0.67 (+/- 0.06)
	Univariate Selection	0.54 (+/- 0.05)	0.66 (+/- 0.05)
Multi-Layer Perceptron	Recursive Feature Elimination	0.54 (+/- 0.05)	0.68 (+/- 0.06)
	Wrapper Method	0.54 (+/- 0.07)	0.66 (+/- 0.07)
	Univariate Selection	0.53 (+/- 0.07)	0.65 (+/- 0.07)

Table 3: Results without feature normalization (CV=10)

I used a total of 6 classifiers in this project, which are Decision Tree, Gradient Boost, Ada Boost, Random Forest, Support Vector Machine, and Multi-Layer Perceptron. And the experimental results based on the 4-3-3 formation and 10-cross validation were the best. So here I only describe and show the results for it.

Since this problem is a 3-class classification, the baseline was set to an accuracy of 33%. However, looking at Tables 2 and 3, all classifiers in any method show higher accuracy. Among them, Ada boost classifier and SVM show the best performance. However, in the case of SVM, there were cases where the runtime exceeded 2500 seconds. Accordingly, it can be determined that the Ada boost classifier is the most suitable regardless of feature normalization.

5. Discussion and Future work

With this project, I have shown that it is possible to predict the outcome of a match by calculating the team's performance value with the players' individual season stats instead of using details about the team that played the match.

What I want to do based on this project in the future is to calculate the expected goal and conceded of the team based on the player’s stat and method I used and predict the outcome based on team’s expected goal and conceded. This requires additional data, such as GPS data that records players' movements during the game, club's expenditure, and salary. Or I would like to apply this model to other soccer leagues to verify the model.

References

[1] Zaveri, N., Shah, U., Tiwari, S., Shinde, P., & Kumar, T. L. (2018). Prediction of football match score and decision making process. *International Journal on Recent and Innovation Trends in Computing and Communication*, *6*(2), 162-165.

[2] Groll, A., Ley, C., Schauberger, G., Van Eetvelde, H., & Zeileis, A. (2019). Hybrid Machine Learning Forecasts for the FIFA Women's World Cup 2019. *arXiv preprint arXiv:1906.01131*.

[3] Capobianco, G., Di Giacomo, U., Mercaldo, F., Nardone, V., & Santone, A. (2019). Can machine learning predict soccer match results?. In *ICAART (2)* (pp. 458-465).

[4] Baboota, R., & Kaur, H. (2019). Predictive analysis and modelling football results using machine learning approach for English Premier League. *International Journal of Forecasting*, *35*(2), 741-755.

[5] Liti, C., Piccialli, V., & Sciandrone, M. (2017). Predicting soccer match outcome using machine learning algorithms. In *Proceedings of MathSport International 2017 Conference* (p. 229).