IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Volume-6, Issue-4 E-ISSN: 2347-2693

Prolego: A Data Science Approach to Predict the Outcome of a Football Match

Sourabh Swain[1*], Shriya Mishra[2]

[1]SAP Labs, India
[2]SAP Labs, India

Available online at: www.ijcseonline.org

Received: 08/Mar/2018, Revised: 15/Mar/2018, Accepted: 26/Mar/2018, Published: 30/Apr/2018

Abstract- Prolego aims to predict results of Premier League football matches accurately by applying machine learning techniques to historical data. The historical data consists of rows where each row consists of several statistics for both the Home Team and the Away Team. The historical data is generated using web scraping libraries such as Selenium and BeautifulSoup. Based on the scraped data, data cleaning and feature engineering is done to generate several features of a football match like Shots, Shots On Target, Possession, Tackles, Corners, Ratting etc. Finally, the features are represented in a vector format and fed as inputs to different Machine Learning classifier algorithms like Multinomial Logistic Regression, SVM, Gradient Boosting Classifier and DecisionTreeClassifier. After the classification, accuracy is measured by calculating percentage of correct predictions and percentage of correct draw predictions. Error analysis is performed using techniques like Region under Curve to tune hyperparameters and identify the features which are more prominent/useful in accurately predicting the results.

Keywords- Prolego, Dataset, Collection

1. Introduction

Football is the most popular sport on the planet. Part of its popularity can be attributed to its highly unpredictable nature. Compared to other sports, a game of football can change in a matter of minutes and sometimes, even seconds. Among all football leagues around the world, the premier league is considered to be the most competitive and unpredictable league. Leicester City winning the league title in 2015-16 with odds of 5000 1 against them highlights the highly unpredictable nature of the league. In this project, we try to predict the results of different matches in the Premier League by generating historical data, performing data analysis, feature engineering and finally evaluating the performance of different machine learning models.

2. Related Work

Many previous works have attempted to predict the results of premier league matches before. Ben Ulmer and Matthew Fernandez of Stanford University [1] used game day data and current team performance achieving error rates of linear classifier (:48), Random Forest (:50), and SVM (:50). Another work that we went through was by Timmaraju et al [2]. They were able to incorporate features such as corner kicks and shots attempted, which is why they were able to obtain an accuracy of 60% using a RBF-SVM. Joseph et al used another approach to the problem. They used Bayesian Nets to predict the results of matches played by Tottenham Hotspur during 1995 1997. Their results show huge variations in accuracy (38% 59%). However, their approach provides a different insight into feature selection. Finally, we took further inspiration from the famous Kaggle [4] competition called March Madness[4] where different approached like converting the dataset into feature vectors and then evaluation of different machine learning models are performed.

3. Keywords

- **Corner** - a place kick taken by the attacking side from a corner of the field after the ball has been sent over the byline by a defender.
- **Free Kick** - A free kick is an action used in several codes of football to restart play with the kicking of a ball into the field of play.
- **Home team**- a sports team playing at Home (sports), which has the home advantage

- **Tackle** - try to take the ball from (an opponent) by intercepting them.
- **Pass** - kick to another player of one's own side.
- **Clearance** - a kick that sends the ball away from one's goal.
- **Offside** - of a player in some sports) occupying a position on the field where playing the ball or puck is not allowed, especially (in soccer) in the attacking half ahead of the ball and having fewer than two defenders nearer the goal line at the moment the ball is played.

4. The Data Set

Data forms the most integral part for any Machine Learning Algorithm. It is the data, that trains the Algorithm to make predictions. The more accurate data is fed, the more accurate are the results. Hence, collecting and preparing a data set forms a crucial part. For our project, we decided to build the dataset over the period of 2013-14 season to 2016-17 season. We wanted to incorporate features like possession, corners, tackles, o sides, clearances etc. We also wanted to take into account the ratings of the teams playing as usually better players lead to better performances. The reason for choosing 2013-14 season was due to the availability of all the statistics like shots taken, corners, possession on the social Premier League website for every game from that season onwards. Thus, having a dataset which consisted of features that could describe a match was the primary motivation of including the features that have been incorporated. We also introduced a feature of home team factor as it is usually seen that the home team has an advantage in the premier league.

5. Methodology

Here are the steps that we followed while collecting and preparing the data.

5.1. Feature Selection

In order to generate the data set, the first thing we had to decide upon is what features do we need for our problem i.e. which all features are relevant to the problem of accurately predicting the result of a football match. Statistics such as possession, corners, shots attempted, tackles come to the mind as these are important aspects of a match. For obtaining all such statistics relevant to a particular match, we decided to scrape the data from the official Premier League website [5] as it contained an authentic and detailed summary for every match. Also, the ability of the players is another thing which has a huge influence on the outcome of the match. For this, we decided to build a feature called team rating for each team. We decided to use the ratings of the popular game, FIFA by EA Sports. In order to calculate the team rating for a particular team, we fetched the ratings of each player who played in the particular match and then calculated the effective rating of the player in that match using the Minutes played by an individual player (Mi) Total minutes in a match ( ) and the Player Rating ( ). We calculated the Effective Rating ( ) of a player as follows:

$$= \quad \times$$

Using the individual obtained from (1), the total

Team Rating ( ) was calculated as:

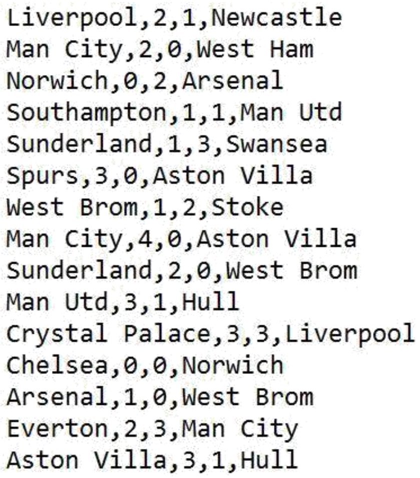
$$=\sum$$

Here, is the number of players of a team.
For fetching the EA Sports FIFA ratings of each player, we used the FIFA Index website [6]. This way we could calculate the effective team ratings for all the teams for a given match.

5.2. Data Collection

This step involves in collecting data from various sources that is relevant to the problem statement. We adopted a method known as Scraping. Data scrapping or web scrapping is a method that is used to collect data from web pages into a readable form for e.g. in the form of a spread sheet or notepad etc., for further data analysis.

To collect data, we started off with BeautifulSoup4, a python framework which is used for pulling data out of HTML or XML les. This gave us incomplete data as BeautifulSoup4 is a library that extracts data from static web pages (i.e., pages that are purely designed from HTML and XML), and the website we were sur ng was a dynamic one. Hence, the data that we collected was incomplete. We then used another framework Selenium to extract information from the dynamic pages. Selenium is a framework that creates a version of the web browser which is controlled by python. Thus, allowing us to extract information from dynamic pages.



```
Liverpool,2,1,Newcastle
Man City,2,0,West Ham
Norwich,0,2,Arsenal
Southampton,1,1,Man Utd
Sunderland,1,3,Swansea
Spurs,3,0,Aston Villa
West Brom,1,2,Stoke
Man City,4,0,Aston Villa
Sunderland,2,0,West Brom
Man Utd,3,1,Hull
Crystal Palace,3,3,Liverpool
Chelsea,0,0,Norwich
Arsenal,1,0,West Brom
Everton,2,3,Man City
Aston Villa,3,1,Hull
```

Figure 1: Data obtained after Scraping

5.3. Data Preprocessing

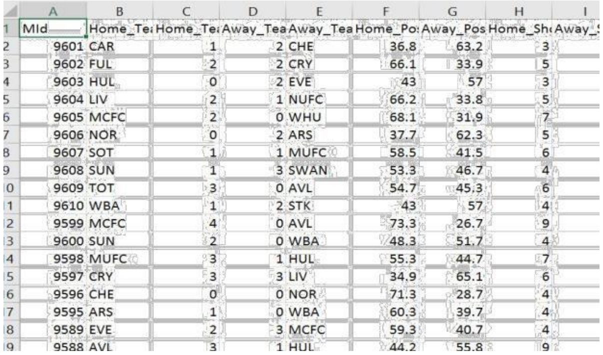
Preprocessing is basically arranging the data in a format in which the data can be visualized. As you can see in the Figure 1, the data that we had obtained after scraping, may not be in a format that could be fed to the Machine Learning Algorithms. Hence, the data has to be organized to make it easier to work with. We proceeded with data processing in the following order.

5.3.1. Data Formatting

After scrapping, the data present with us was haphazard. It was not in a format that could be accepted by a Machine Learning Algorithm (as, all machine learning Algorithms accept data in a vector format). We chose to arrange the data, obtained, in a
.csv file to visualize the data in a better way. Organizing that data in a .csv file helped us to arrange and segregate the data into rows and columns based on the features selected by us.

5.3.2. Data Cleaning

This is the step in which we add or remove data based on the problem requirement. After the data was arranged in a .csv file we came across several records that were redundant and some important records that were incomplete. Such situations had to be specially handled. In our case, as we had extracted data from one web page, we found that the rating of a few players was missing. To handle this situation, we had to plug out data from other web pages to fit into the incomplete records.



	A	B	C	D	E	F	G	H	I
1	MId	Home_Te	Home_Te	Away_Tea	Away_Tea	Home_Po	Away_Pos	Home_Sh	Away_
2	9601	CAR	1	2	CHE	36.8	63.2	3	
3	9602	FUL	2	2	CRY	66.1	33.9	5	
4	9603	HUL	0	2	EVE	43	57	3	
5	9604	LIV	2	1	NUFC	66.2	33.8	5	
6	9605	MCFC	2	0	WHU	68.1	31.9	7	
7	9606	NOR	0	2	ARS	37.7	62.3	5	
8	9607	SOT	1	1	MUFC	58.5	41.5	6	
9	9608	SUN	1	3	SWAN	53.3	46.7	4	
10	9609	TOT	3	0	AVL	54.7	45.3	6	
11	9610	WBA	1	2	STK	43	57	4	
12	9599	MCFC	4	0	AVL	73.3	26.7	9	
13	9600	SUN	2	0	WBA	48.3	51.7	4	
14	9598	MUFC	3	1	HUL	55.3	44.7	7	
15	9597	CRY	3	3	LIV	34.9	65.1	6	
16	9596	CHE	0	0	NOR	71.3	28.7	4	
17	9595	ARS	1	0	WBA	60.3	39.7	4	
18	9589	EVE	2	3	MCFC	59.3	40.7	4	
19	9588	AVL	3	1	HUL	44.2	55.8	9	

Figure 2: Data arranged in a .csv format

5.3.3. Sampling

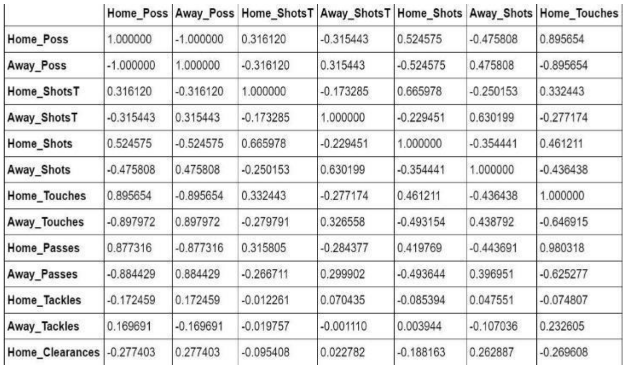
Sometimes, there might be situations where we have a data set with large number of features. Large data sets might sometimes slow down the performance of the algorithms to which the data is fed. There might be cases where the features selected by us are related to each other. In this case, it is sensible to drop some as it can improve the performance significantly.

On analyzing the data obtained, we noticed that there were several features that were similar. To confirm our intuition, we analyzed features by comparing two at a time and visualizing it using plots. Comparing two features at a time got tedious as we had 28 features. This left us with 378 possibilities to be compared and comparing them manually was next to impossible. Moreover, we ran into few errors while making our analysis. We had to search for other methods to help us draw conclusions.

On further research, we found a technique to analyze features automatically. This could be done by the Correlation Matrix. This matrix finds the correlation between every feature pairwise. In other words, it compares every feature with every other feature present in a data set. There are many Correlation Matrices out of which we chose the Pearson's Correlation Matrix.

The Pearson's Correlation quantifies the degree to which a relation can be established between two variables. The correlation compares all the features and gives an output from a scale of 1 to 1. When the output is 1, it implies that the increase in one variable leads to the increase in the other. An output of 1 indicates that, the increase in one variable leads to the decrease in the other. An output of 0 indicates that the variables are independent.

At first, we got several numbers in a matrix format ranging from 1 to 1, as shown in figure 3. It was difficult for us to decipher the correlation. Using Pandas, the matrix could be highlighted with colors. This helped us to visualize the high and low correlations in a better way. As one can see in figure 4, as the shade for a cell becomes darker it indicates an inverse correlation and as the cells become lighter, it indicates that the two variables are strongly correlated.



	Home_Poss	Away_Poss	Home_ShotsT	Away_ShotsT	Home_Shots	Away_Shots	Home_Touches
Home_Poss	1.000000	-1.000000	0.316120	-0.315443	0.524575	-0.475808	0.895654
Away_Poss	-1.000000	1.000000	-0.316120	0.315443	-0.524575	0.475808	-0.895654
Home_ShotsT	0.316120	-0.316120	1.000000	-0.173285	0.665978	-0.250153	0.332443
Away_ShotsT	-0.315443	0.315443	-0.173285	1.000000	-0.229451	0.630199	-0.277174
Home_Shots	0.524575	-0.524575	0.665978	-0.229451	1.000000	-0.354441	0.461211
Away_Shots	-0.475808	0.475808	-0.250153	0.630199	-0.354441	1.000000	-0.436438
Home_Touches	0.895654	-0.895654	0.332443	-0.277174	0.461211	-0.436438	1.000000
Away_Touches	-0.897972	0.897972	-0.279791	0.326558	-0.493154	0.438792	-0.646915
Home_Passes	0.877316	-0.877316	0.315805	-0.284377	0.419769	-0.443691	0.980318
Away_Passes	-0.884429	0.884429	-0.266711	0.299902	-0.493644	0.396951	-0.625277
Home_Tackles	-0.172459	0.172459	-0.012261	0.070435	-0.085394	0.047551	-0.074807
Away_Tackles	0.169691	-0.169691	-0.019757	-0.001110	0.003944	-0.107036	0.232605
Home_Clearances	-0.277403	0.277403	-0.095408	0.022782	-0.188163	0.262887	-0.269608

Figure 3:Pearson correlation matrix for the feature set

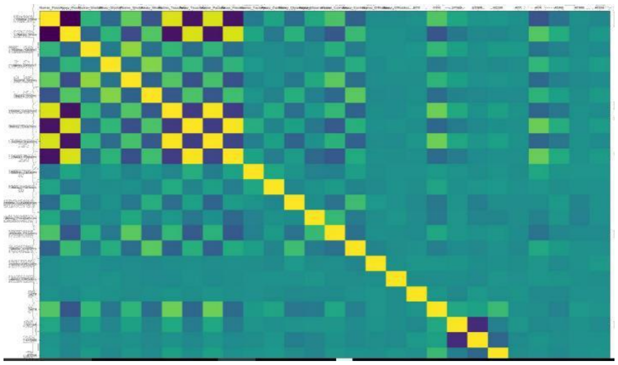


Figure 4: Color Visualization of the matrix shown in Figure 3

5.4. Feature Engineering

This is the step in which we modify features set, of the raw data, to represent the data, in a better format, to the predicting models. At first, we chose to decompose the feature named as Total Rating into Midfield Rating, Attack Rating and defense Rating as in a football match, the Total Rating may not always reflect the true nature of how the game is played. There have been instances where a team with very high Defense Rating has won against teams which had Total Rating more than them. Thus, breaking the total team rating into 3 sub categories helped us gain more insight into a particular match.

5.5. Data Transformation

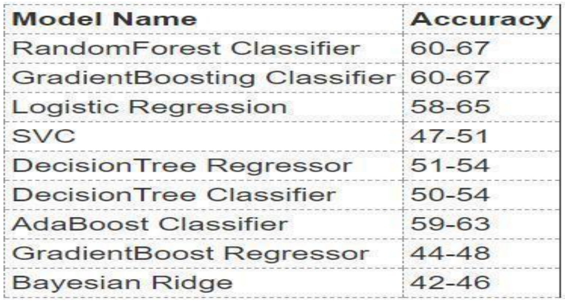
The next step is important as it involves in transforming the dataset into feature vectors, where each team of a match is represented using a feature vector. The feature vector consists of all 28 features, having a dimension of 1x28 suitable to be fed to the Machine Learning Algorithms. We also had two fields named Home Team and Away Team, to distinguish between the two, we assigned 1 to the Home Team and -1 to the Away Team.

6. Results and Evaluation

We implemented several machine learning algorithms with our data set and we got the best accuracy with the Random Forest Classifier and the Gradient Boosting Classifier. The following are the results obtained:

6.1. Results with the Random Forest Classifier

We predicted the matches of the 2016-17 season using 11 different machine learning models. After evaluating and tuning these 11 models, we found that Gradient Boosting Classifier and Random Forest Classifier gave us the best results, with accuracy ranging between 60% - 67% for both of the models. The models along with their accuracies can be found in the figure 5.



Model Name	Accuracy
RandomForest Classifier	60-67
GradientBoosting Classifier	60-67
Logistic Regression	58-65
SVC	47-51
DecisionTree Regressor	51-54
DecisionTree Classifier	50-54
AdaBoost Classifier	59-63
GradientBoost Regressor	44-48
Bayesian Ridge	42-46

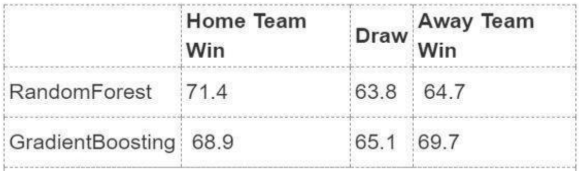
Figure 5:Results of the output for various Machine Learning Algorithms

Due to higher accuracy of GradientBoosting Classifer and Random-Forest Classifer, we focus on these two models for further evaluation. One common problem faced in other works was poor accuracy in predicting draws.

Since our dataset had features which were specific to a particular match, we found that our accuracy in correctly predicting win, draw or lose was roughly equal. The accuracies for home Team Win, Draw and Away Team Win can be found in the tables below for both

© 2018, IJCSE All Rights Reserved

the models. Since our dataset had features which were specific to a particular match, we found that our accuracy in correctly predicting win, draw or lose was roughly equal. The accuracies for home Team Win, Draw and Away Team win can be found in the tables below for both the models.



	Home Team Win	Draw	Away Team Win
RandomForest	71.4	63.8	64.7
GradientBoosting	68.9	65.1	69.7

Figure 6: Results for the matches given by the GradientBoosting and RandomForest Classifier

Further, we also were interested to know which features had the highest contribution or weightage while predicting the results for Random Forest Classifier, we found out that Shots on Target was the most important feature followed by Clearances, while for GradientBoosting Classifier, Team Rating was the most important feature followed by Clearances.

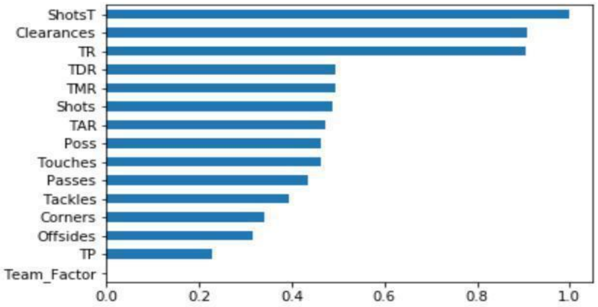


Figure 7:Feature importance for the Random Forest Classifier

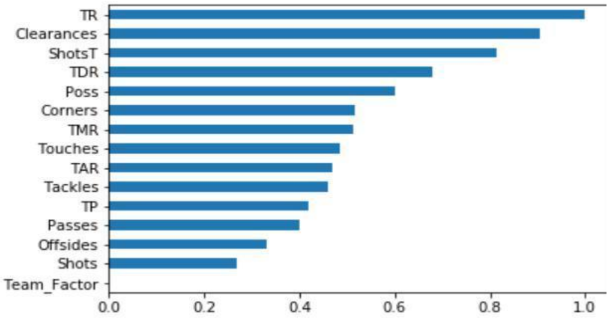


Figure 8: Feature importance for the Gradient Boosting Classifier

.

7. Future Work

Our model performs comparatively well in predicting draws for the time period chosen. However, it remains to be seen how it would perform over a longer time period. The accuracy obtained was for a relatively short period of time (4 years). Also, the accuracy of the models can be further improved by having more accurate data in terms of past statistics between the two playing teams. Other aspects which can be taken into account are player form and playing styles, which could have a significant impact on the accuracy of the models.

8. References

[1]. Ben Ulmer and Matthew Fernandez, Predicting Soccer Match Results in the English Premier League. (http://cs229.stanford.edu/proj2014/Ben%20Ulmer, %20Matt%20Fernandez,%20Predicting%20Soccer% 20Results% 20in%20the%20English%20Premier%20League.pdf)

[2]. A. S. Timmaraju, A. Palnitkar,& V. Khanna, Game ON! Predicting English Premier League Match Outcomes, 2013. (http://cs229.stanford.edu/proj2013/TimmarajuPalnitkarKhanna-GameON!PredictionOfEPLMatchOutcomes.pdf)

[3]. Kaggle March Machine Learning Mania https://www.kaggle.com/c/march- machine-learning-mania-2017

[4]. Adit Deshpande, Applying Machine Learning to March Madness - Applying Machine Learning To March Madness (https://adeshpande3.github.io/Applying-Machine-Learning-to-March-Madness)

[5]. Premier League website - https://www.premierleague.com/

[6]. EA Sports FIFA Rating - https://www. faindex.com