
BEAT THE BOOKIE

A PREPRINT

Group Name: GROUP E
Department of Computer Science
University College London
London, WC1E 6BT

December 22, 2021

1 Introduction

Football is one of the most popular sports in the world. It also occupies the largest betting market in the United Kingdom[1]. As a result, predicting the match result has become a trending topic. Many researchers have used data collected from historical matches, such as goals, to forecast future outcomes.

In this report, we used the data from the English Premier League(EPL) and other customized helpful data sources to build a model that predicts the match results in January 2022. The model takes the home team and away team as the inputs and returns the value of full time result (FTR) in H (home win), D (draw) and A (away win). We aimed to beat the 53 % accuracy with our model.

To achieve this goal, new features, which involve ratings of the teams and their average goals, were designed based on the provided data. We trained several multiclass classification algorithm models with batch supervised learning and compared their performance in two ways. The results showed that the Random Forest model outperforms other models on the test set. Then, we searched for the optimal hyperparameters of the Random Forest model using k-fold cross-validation. This prevented overfitting and improved the general performance of the model. In the end, the model predicted 56.7 % of the results correctly. The accuracy varied slightly when the data in the test set was changed on different runs.

2 Data Transformation and Exploration

The original training data listed the 22 features of 4940 matches from August 2008 to May 2021. These features include the date, the name of the home team and away team, the result of the match, each team's number of goals at half time and full time, the number of shots on goal and target, the fouls committed, the corners, and the warnings and penalties received by each team.

2.1 Outlier

An outlier is an absolutely unusual data point. We didn't discover any abnormal data after reviewing all the football match data, so we omitted the step of removing outliers.

2.2 Missing values(NaN)

There is no missing value in the data. Nonetheless, it is necessary to drop NaN rows while importing the datasets from the CSV files.

2.3 New data

We added the latest data from [2] to the original dataset in the same format.

2.4 Getting new features

2.4.1 Motivation

Although the data given in the training CSV file is correlated to the FTR, it contains features that will not be given for the final prediction. So new features that can be determined before the match are beneficial.

2.4.2 Expected features

We wanted to create new features that indicate a team's general performance and relative strength against opponent teams. Thus, we checked each team's win/lose/draw rate and looked at the cumulated FTR results when the same team faced each other multiple times (the first three columns of Figure 2). We then used the mean of some of the given features to show each team's average performance and some rating/ranking system to represent their relative strength. Two tables containing the relationship between average goal amounts and the average Win/Draw/Lose rate according to our decisions were laid out. And they show the long term and recent relations separately. Using these two tables, we plot the data correlations once again in Figure 1.

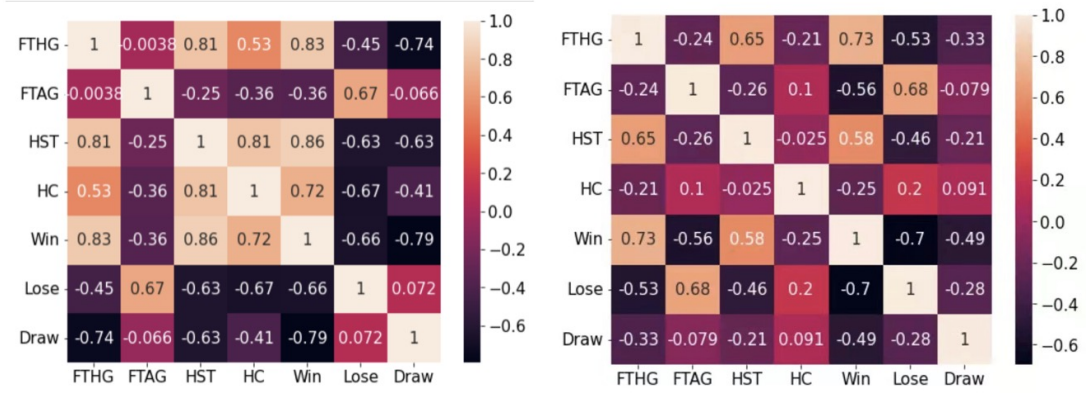


Figure 1: Correlation of the average of the given features and FTR, left long term, right recent

The results show that the long term data is more correlated to the FTR value. Thus, we will calculate our new features regarding the long term data.

2.4.3 Missing team

We listed the past encounters of the teams that will meet in January 2022. The historical matches with couples on their respective home and away fields are shown in figure 2. It was found that Brentford was a relatively new team, as it did not appear in the original EPL data. By adding the latest data, we insert 15 match results of it. However, it only competed with Liverpool once on Brentford's field. The lack of data regarding this team will be coped with in the next section.

Out[20]:

		A	D	H	Average home goals	Average away goals
HomeTeam	AwayTeam					
Aston Villa	Man United	6	4	0	1.232	1.596
West Ham	Leeds	0	0	1	1.475	1.500
Norwich	Everton	1	3	1	1.154	1.141
Brighton	Crystal Palace	2	1	1	1.083	1.131
Wolves	Southampton	0	2	1	1.163	1.100
Liverpool	Brentford	0	0	0	2.117	1.375
Tottenham	Arsenal	1	4	8	1.797	1.629
Man City	Chelsea	4	1	8	2.520	1.637
Newcastle	Watford	2	1	1	1.385	1.000
Burnley	Leicester	2	1	3	1.079	1.447

Figure 2: The historical matches with Pairs

3 Methodology Overview

3.1 New features

3.1.1 Rating-relative strength

The first critical area of our machine learning module is the rating method for each team. In the beginning, we use the traditional technique that only concerns the full-time results of matches to rate the strength of teams at each game[4].

The equation is shown below:

$$rating = \frac{1 + wins}{2 + matches} \quad (1)$$

However, equation 1 is not reliable as the rating outcome of each match is calculated individually, so it only gives us the team performance at one game without considering the general performance of one team during the whole season. Fortunately, the primary commonly used rating system called 'Elo rating' can tackle the problem in equation one to calculate the relative strength between teams.

The Elo rating system was initially invented to calculate the relative skills of chess players. However, it is also expanded to calculate the relative strength between football teams by updating the ranking value of each football team after every game. In this system, a higher ranking value indicates more advanced skills.

Each group's following nine ratings are introduced based on different game events in our approach. The initial ratings of each team are set to 2000, and they are stored and updated after each match.

1. Outcome rating
2. Home offensive rating
3. Away offensive rating
4. Home defensive rating
5. Away defensive rating
6. General offensive rating
7. General defensive rating
8. General Home rating
9. General defensive rating

It is better to understand the equation of the expected score (ES) before explaining the theory and formula of the rating system because the methods used to calculate the expected score are the same, except for some variations in coefficients. For the two opponent teams A and B, if the rating of team A is R_A and the rating of team B is R_B , then the expected score of team A is:

$$ES_A = \frac{1}{1 + 10^{(R_B - R_A)/400}} \quad (2)$$

And the expected score of team B is calculated by a similar equation:

$$ES_B = \frac{1}{1 + 10^{(R_A - R_B)/400}} \quad (3)$$

The relationship between ES_A and ES_B is:

$$ES_A + ES_B = 1 \quad (4)$$

The ES of a football team is the predicted outcome of one match and has a range from 0 to 1. A higher ES value indicates a greater chance of winning. If ES exactly equals 0.5, then it means a draw. The denominator of the ES equation shows that the exponent of 10 is the division of rating difference between the two teams and 400. The creator of Elo proposed this value because he wanted the more skilled chess player to have an expected score around 0.75 when two players have a 200 ranking points difference. Therefore, in the Elo rating system, the result of one football match is predicted by considering the ranking points difference or the relative skill between two teams. After some experimentation, we set a suitable value of 100 instead of 400[7]. The ES equations of the following nine ratings are the same, except having different divisor values. The first Elo rating is the outcome rating that is determined by the full-time results (FTR) of all the past games, and the equation used to update the outcome rating is shown below:

$$R'_A = R_A + K_1(S_A - E_A) \quad (5)$$

In this equation, R'_A and R_A mean the rating value after and before one match. S_A indicates the actual tournament score of the game. When the team wins the game, S_A equals 1. Conversely, S_A will become 0 if the team loses the game. Then, if the game's result is a draw, S_A equals 0.5. The most critical parameter in the rating updating equation is the K-factor because it controls the maximum adjustment of the rating for one match. An increase in the value of K raises the sensitivity of rating value to the outcome of matches and vice versa. Therefore, a significant value of K is used for a team new to the rating pool until it completes enough games.

On the other hand, a small value of K is used for a team that is well known or has very high strength as the rating of this team should be very close to the actual performance; thus, the result of one game should not affect the rating a lot. After updating the outcome rating, the new rating list will be an excellent reference to show the relative strength

between teams. The new rating value is used to calculate the expected score of the next game. However, the prediction is only built on the result of games. In that case, we cannot know the main reason for improving a team's chances of winning. Sometimes, the result of a game can be heavily affected by some in-game match events, which leads us to use inappropriate data, thus generating prediction errors. That is why rating 2 to 5 is created. Rating 2 and 3 are based on the team's shot attempt in a game and indicate the team's performance at the home court and the away court, respectively. Therefore, the Corsi method is implemented to calculate the relative shot on goals between teams. The Corsi percentage equation of a football team playing at home field is shown below:

$$\frac{HS}{HS + AS} \quad (6)$$

where HS is the total number of shots on goal by the home team and AS is the total number of shots on goal by the away team. Therefore, new rating updating equations of Home offensive and away offensive rating becomes:

$$R'_A = R_A + K_2 \frac{HS}{HS + AS} \quad (7)$$

$$R'_A = R_A + K_3 \frac{AS}{HS + AS} \quad (8)$$

Equation 6 is used for home offensive rating and Equation 7 is used for away offensive rating. The Corsi method inspired us to create a Elo rating list for football teams based on the number of shots blocked. Therefore, the rating updating equation of home defensive rating and away defensive rating is created:

$$R'_A = R_A + K_4 \frac{AST - FTAG}{(AST - FTAG) + (HST - FTHG)} \quad (9)$$

$$R'_A = R_A + K_5 \frac{HST - FTHG}{(AST - FTAG) + (HST - FTHG)} \quad (10)$$

Where AST means the total number of shots on target by the away team, $FTAG$ means the goals scored by the away team at full time, HST means the total number of shots on target by the home team and $FTHG$ means the goals scored by the home team at full time. However, we cannot calculate the ES of teams using rating 2,3,4,5 as the ES calculation requires the corresponding rating of two teams. Thus, four general ratings are created from rating 2 to 5 using a linear combination of two corresponding ratings with a suitable coefficient to each term. The relation between each rating are listed below:

1. Home offensive rating and away offensive rating form general offensive rating
2. Home defensive rating and away defensive rating form general defensive rating
3. Home offensive rating and home defensive rating form general home rating
4. Away offensive rating and away defensive ratings form general away rating

Each general rating concerns the performance of teams at different courts, the overall shot attempt and the defensive ability. Finally, the ES equation can be used to calculate the expected results of opponent teams in one match according to the different game events in previous matches.

3.1.2 Average-general performance

We took 5 features from the given data to construct their averages per match[5]. And the features are shown blow:

1. the number of matches played by the teams
2. average goals per match
3. average concedes per match
4. average margin of victory per match
5. average corners per match

These features are generated for both the home team and the away team for each match.

3.1.3 Putting up the new features table

We want our final trained model to predict match results without any prior knowledge of the original given features. The model needs to elaborate on the data calculated before the match to satisfy this condition. To get this result, we have to train the model using the same way. In detail, for each match X in the given data, we calculate all the new features we designed for both teams before match X and the result of match X. In this way, the model meets the same scenario whenever a new line of data is fed into the model. The first ten matches of each team were omitted before putting the table together to balance the numbers of new features. For the sake of Brentford, we would repeat the process without dropping each team's first ten matches. So we can get a rating of how new teams correlate with the results. Then the new table will be used for the training process to predict the Brentford match only.

Out[29]:

	home_team_matches	average home goals	average home concedes	home average margin of victory	home average corners	home rating	home_elo	home_off	home_def	HES	FTR	away_team_matches	average away goals	c
227	11	1.636	1.000	0.636	8.091	0.615	2025.0	2045.0	2053.0	0.629389	D	11	1.182	
249	12	1.500	0.917	0.583	8.500	0.571	2017.0	2048.0	2055.0	0.738109	D	12	1.000	
259	13	1.385	0.846	0.538	8.462	0.533	2013.0	2051.0	2057.0	0.607661	D	13	0.231	
279	14	1.286	0.786	0.500	8.643	0.500	2013.0	2053.0	2059.0	0.729110	H	14	1.214	
299	15	1.467	0.733	0.733	8.533	0.529	2020.0	2055.0	2061.0	0.710540	H	15	1.000	
...
5003	22	1.409	1.273	0.136	6.045	0.375	1845.0	2051.0	2058.0	0.666139	H	98	0.969	
5024	23	1.391	1.217	0.174	6.043	0.400	1838.0	2054.0	2059.0	0.471249	D	118	1.017	
5048	24	1.375	1.208	0.167	6.042	0.385	1841.0	2056.0	2060.0	0.431359	D	138	1.449	
5070	25	1.360	1.200	0.160	6.200	0.370	1834.0	2058.0	2061.0	0.442688	H	158	1.146	
5105	27	1.370	1.185	0.185	6.185	0.379	1832.0	2062.0	2062.0	0.303871	A	255	1.620	

4485 rows × 21 columns

Figure 3: Modified date table

3.1.4 Variations of training data

The new feature table involves the data for the home team and away team separately. They are then concatenated together by match index. This approach, therefore, gives us a variety of ways to concatenate the tables. The data is merged horizontally for the previously shown table. We also had the data added vertically, with a new column stating whether the data is for the home team or away team. In addition to that, the two sets were trained separately and had their results combined. In the end, we will choose one of the best options according to our performance measure.

3.2 Dimensionality Reduction

3.2.1 Algorithms

Three different dimensionality reduction techniques were used to shrink the size of the features and make the data more correlated. Firstly, we fit the Linear Discriminant Analysis(LDA) to the training data to get two components out. Secondly, LDA is also used, but it's on the separated data sets to produce 1 component each. Lastly, we use Principal component analysis on the training data without FTR values. All dimensionality reduction algorithms are from the scikit-learn package.



Figure 4: Correlation map for LDA on full table

3.2.2 Correlations

Using the heatmap function from seaborn library, correlations of the above prepared tables and the FTR result can be plotted. Comparing the correlation results, we discarded the PCA approach.

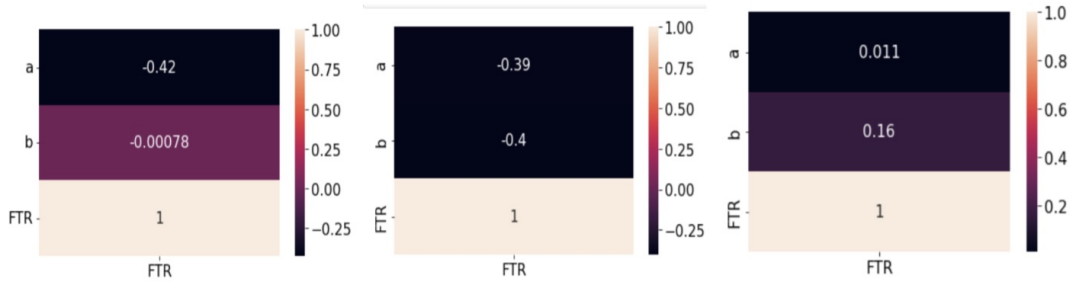


Figure 5: The heatmap of LDA reduced data, separately reduced data and PCA reduced data

4 Model Training and Validation

4.1 Training

4.1.1 Train test split

As the total data size is relatively small, using a larger test set will increase the possibility of overfitting and reduce the general features learned by the model. On the other hand, a smaller test set will make the evaluation of the model unstable and unreliable. Therefore, we randomly split the total data into 90% training and 10% test sets. We used “random_state” equals 18 to make the result reproducible.

4.1.2 data

Five sets of similar data are fed into the models, all of which are calculated from step 3 - Methodology. To restate, these are:

1. The new features data from 3.1.3
2. Two separately produced data by home team and away team
3. A vertically appended data from b.
4. An LDA reduced set using a.
5. An LDA reduced set using b.

These data sets will be rescaled and sent to the models.

It is essential to scale the training data before feeding them to the model since the features may not have the same range. The features may have different orders of magnitude. As a result, the features with higher orders of magnitude may be more “prior” than the others, “control” the model and stop the model from learning from other variables. The learning algorithm will have better performance and higher stability when the features have the same range. The standard scalar removes the mean and scales to unit variance. The standard score of a sample x is calculated as:

$$Z = \frac{x - u}{s} \quad (11)$$

where u stands for the mean of the training data/ zero if `with_mean=false`, and s represents the standard deviation of the training samples/ one if `with_std=false`[3].

In this task, the standard scalar is applied to scale the data to ensure the data to have the same orders since the standard scalar provides a reasonable data range. In addition, the outliers are not necessarily required to be considered.

4.1.3 Models

In total, we tried 11 different models on our five sets, along with scikit-learn’s Voting Classifier to train the models.

```
classifiers = [
    KNeighborsClassifier(3),
    SVC(kernel="linear", C=0.005),
    SVC(C=0.05),
    DecisionTreeClassifier(max_depth=5),
    RandomForestClassifier(max_depth=5, n_estimators=10, max_features=1),
    MLPClassifier(alpha=1, max_iter=1000),
    AdaBoostClassifier(),
    GaussianNB(),
    LogisticRegression(solver='lbfgs', max_iter=250),
    QDA(),
    LinearDiscriminantAnalysis()
]
```

Figure 6: Eleven different classifier

4.1.4 Performance measure and evaluation

We used a combination of two rating systems to measure the model performance.

Percentage This calculates the ratio between the number of cases that the model gets right and the total number of cases.

Performance If the actual result is a home win, the prediction will wrong, regardless of whether it is a draw or an away win. Therefore, we need a system that counts the “distance” of each prediction to the actual one. The performance system is introduced: it has a total score of 100 and gets smaller when the prediction is farther from the actual result. Our performance measure equals the product of the two ratings. The optimized model will have the highest performance measure.

Based on this, the most models got scores around 50% and performance around 66, outputting the overall performance measures of 33.0. The highest score is obtained from the random forest model with a performance measure of 38.48, with the linear support vector machine getting the second-best, with a measure of 37.32.

4.2 Cross-validation - hyperparameter tuning

After running 11 models on the 5 data sets for several iterations, the Random Forest model on the LDA reduced set often has the highest performance measure. We attempted to tune the model to improve its performance on the testing set and prevent overfitting on the training set.

4.2.1 Random search

By utilizing the randomized search algorithm from the scikit-learn library, we fed in an extensive range of possible parameters for Random Forest. K-fold cross-validation was integrated into the algorithm. The number of folds k was set to 10 for our setting. The random search gave a shrunk range of the optimal model parameter of our selected model.

4.2.2 Grid search

Based on the outcome of the random search, Grid search was used to divide the shrunk parameter range into finer sections and test each of them out using cross-validation. This time, a k of 5 was used taking runtime into consideration.

Grid search outputs the optimal parameters for the model.

5 Results

By using the parameters tested by grid search, we trained the model on the full training set. Then we got a result of 0.55 for percentage and 67.96875 for performance. This is often higher than what we would get from using the previously eliminated approaches and models.

```
Out[62]: ('Best RF', 0.5567928730512249, 67.96875)
```

Figure 7: Result

Note: For the repeated train process with a table without any matches dropped, we found the optimal model was a linear support vector machine.

6 Final Predictions on Test Set

By using the best model, Random Forest, we obtained the results shown in Figure 8. However, just as we stated in section 3, the prediction of the Brentford match can be conservative. The result from the un-dropped table with the respective best model then replaces the result for that match. The EPL-test.csv is overwritten by the prediction table.


	Date	Home Team	Away Team	FTR			Date	Home Team	Away Team	FTR
0	2022-01-15	Aston Villa	Man United	A		0	2022-01-15	Aston Villa	Man United	A
1	2022-01-15	West Ham	Leeds	H		1	2022-01-15	West Ham	Leeds	H
2	2022-01-15	Norwich	Everton	A		2	2022-01-15	Norwich	Everton	A
3	2022-01-15	Brighton	Crystal Palace	H		3	2022-01-15	Brighton	Crystal Palace	H
4	2022-01-15	Wolves	Southampton	H		4	2022-01-15	Wolves	Southampton	H
5	2022-01-15	Liverpool	Brentford	D		5	2022-01-15	Liverpool	Brentford	H
6	2022-01-15	Tottenham	Arsenal	A		6	2022-01-15	Tottenham	Arsenal	A
7	2022-01-15	Man City	Chelsea	A		7	2022-01-15	Man City	Chelsea	A
8	2022-01-15	Newcastle	Watford	H		8	2022-01-15	Newcastle	Watford	H
9	2022-01-15	Burnley	Leicester	A		9	2022-01-15	Burnley	Leicester	A

Figure 8: Final predictions

7 Conclusion

In this report, we calculated the strength of each team based on their previous matches and used this rating to predict the results in future competitions. Random Forest achieved an accuracy of 55.6% on the test set after we applied 10-fold cross-validation and grid search to fine-tune it. We used this model to predict the result of the January match, except for Brentford team, which was treated separately.

The future work can use more features that are not shown in our dataset to build the rating system. The prediction could be more precise if we consider different parameters of each player in a team which can measure their performance such as hit rate, turnover and number of steals per game. For example, it may be possible to evaluate the personal strength of each member in a team with a more detailed match report. We can predict the offensive ability of a player using the number of balls passed by them and estimate the defensive ability of a player based on the number of balls stolen by them. The performance of a team under a certain manager/coach, weather, and the team's performance relative to each other can be some more factors to be taken into consideration.

References

- [1] G. Walker, "What are the most popular sports to bet on in the UK?", BUSINESSFIRST, 2020. [Online]. Available: <https://www.businessfirstonline.co.uk/articles/what-are-the-most-popular-sports-to-bet-on-in-the-uk/>. [Accessed: 21- Dec- 2021].
- [2] "Football Betting | Football Results | Free Bets | Betting Odds", Football-data.co.uk, 2021. [Online]. Available: <http://www.football-data.co.uk>. [Accessed: 21- Dec- 2021].
- [3] "Compare the effect of different scalers on data with outliers": https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html
- [4] The Main Idea behind the Colley Method [Online]. Available: <https://www.dcs.bbk.ac.uk/~ale/dsta/2020-21/dsta-3/lm-ch3-colley.pdf>
- [5] Regression to the Mean and Football Wagers [Online]. Available: <http://economics-files.pomona.edu/GarySmith/papers/footballOdds/footballOdds.html>
- [6] Predicting Football Results Using Machine Learning Techniques[Online]. Available:<https://www.imperial.ac.uk/media/imperial-college/faculty-of-engineering/computing/public/1718-ug-projects/Corentin-Herbinet-Using-Machine-Learning-techniques-to-predict-the-outcome-of-professional-football-matches.pdf>
- [7] Elo rating system[Online]. Available:https://en.wikipedia.org/wiki/Elo_rating_system