Illinois Institute of Technology

Statistical Learning Project

MATH 569

Soccer Match Probability Prediction

Shashank Parameswaran

sparameswaran@hawk.iit.edu

Prof. Lulu Kang

April 30, 2022

1 Abstract

In this project, machine learning algorithms are used to predict probabilities of the outcome of soccer matches. The dataset comes from a Kaggle competition called 'Football Match Probability Prediction' and it contains historical matches played by different soccer teams from all over the world. Specifically, the dataset contains information on the past 10 matches of each game for both the teams involved and the outcome of the current match is predicted based on this. The task is a multiclass classification problem where we need to predict if the home team wins, draws or loses the game. Different models have been used such as Logistic regression, Random Forests, XGBoost, LDA and Neural Nets. The best error rate was observed in the Logistic regression model (0.507).

2 Introduction

Soccer is one of the most popular sports worldwide. Over half of the world's population tune in to watch the World Cup each year. According to FIFA's recent survey, there are 265 million players actively involved in soccer which is 4% of the world's population. Each nation has its own professional league system that would have one Premier/Super league where the elite players participate. The leagues do not restrict players from playing in any one country. A player from one nation can play in any league from any other nation. If you look at the English Premier League, the highest level in the English football system, 65% of the players are international players.

Analytics has become a key part of any kind of sport and soccer is no exception. Ball related data such as passes, shots and turnovers give us key information on how a team or a player is performing. Optical tracking can be used to pinpoint the position of the players on the pitch in relation to the ball and opposition. Ever since teams started using a metric called Expected Goals (xG) which measures the quality of a player's shot, the average distance of shots from goal has come down. Many teams have changed their way of attacking play because this metric indicated how shooting from a long range is counterproductive. Predicting soccer match results is very much dependent on the team's form. This report uses data from 10 historical matches with the power of well-known algorithms to predict the outcome of a soccer match. Its real-world applications include gambling, coaching, and journalism.

3 Dataset

3.1 Data Origin

The dataset has been obtained from a Kaggle competition called 'Football Match Probability Prediction'. This dataset is available for anyone who enrols in the competition.

Link to the dataset: https://www.kaggle.com/competitions/football-match-probability-prediction/data.

The training dataset that is used contains the results of 104,461 games that are played worldwide between the time period of December 2019 and May 2021. The dataset contains descriptive features and historical features. There are a total of 188 features in the dataset and 180 of the 188 features contain historical data. These 180 features represent the match date, game played at home/away, Cup match yes/no, goals scored, opponent goals scored, team history rating, team opponent's history rating, team coach and team league ID – each of 10 features representing 10 games for home team and away team (Opponent team). The response variable basically tells us whether the game ended in a win, draw or lose for the home team and it takes values of 'Home', 'Draw' and 'Away' to represent it respectively.

3.2 Data Cleaning

In terms of data cleaning, data points with at least 5 null values in their historical games were dropped, i.e., we used the 'historical match date' column to identify data points that had null values. For other null values, zero was used as a replacement wherever applicable.

3.3 Challenges

A major challenge faced in this dataset is the randomness of data. The response variable is split by 43% wins, 32% losses and 25% draws. Using entropy as a measure for randomness, we get a value of 1.04. An entropy value close to one indicates more randomness. This makes the classification process difficult, and they tend to ignore a class. Moreover, since soccer is a game that requires synergy among the players, a weaker team can beat a stronger team if they are in better sync. This means upsets are common. A strong team not winning (draw or loss) can also be called an upset. Hence the results are more difficult to predict and the models have higher bias for this dataset.

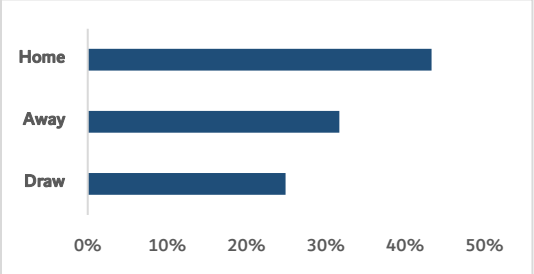

Figure 1: % split of the response variable

4 Feature Engineering

4.1 Approach

Based on the available features, two approaches have been used. Both approaches are implemented for every model and the best result has been shown. The first approach is to use features representing combined information from previous games. The historical features are combined into one field. For example, if we have 'home_team_history_goal_1', 'home_team_history_goal_2' and so on, we sum up all the columns to get 'home_team_history_goals'. When applied to all the historical variables, this reduces the feature set to 28 features.

The second approach uses the information of each historical game as a feature. You can think of it as one-hot-encoding of information from 10 games. This is how it is originally available in the dataset and we add more features to it such as goal difference, streak, etc. For this approach, we have a feature set of 347 features.

4.2 Feature Extraction

New features were generated like home team wins at home, home team wins in cup matches, home team draws at home, winning streak, home team wins when rated higher, coach change and days between historical match date and current match date. All of these metrics were obtained for the away team as well.

4.3 Feature Selection

Removing highly correlated variables using Pearson's correlation was tested. Features like home team goals and home team wins were correlated. However, removing these variables did not improve the accuracy of the model. We did not explore too many methods for Feature Selection due to the higher bias in our models.

4.4 Oversampling and Scaling

Since the response variable had a class imbalance of 43:32:25, oversampling methods like random oversampling and Synthetic Minority Oversampling Technique (SMOTE) were tested. Random oversampling just increases the size of the training set by repetition of the original samples. SMOTE creates new artificial training samples based on the training set. These sampling methods did not improve the accuracy. Similarly, scaling was also performed without much change in the accuracy of the models. The features were standardized by centering and scaling to unit variance.

5 Model Analysis and Results

We have a 3-class classification in our hands. A variety of models are used such as Logistic Regression, LDA, Random Forests, Naïve Bayes and XGBoost. We will also explore models based on neural networks. Models are tested on the modified dataset with 28 features as well as the original dataset with 188 features to see which fits well. Dataset was split into 80% training set and 20% testing set. Since our dataset is time based, we have split the training and test sets based on time, i.e., we considered the first 80% of the data in chronological order to be the training set.

5.1 Logistic Regression

Logistic regression is one of the most popular techniques for solving classification problems. It is easy to implement and interpret. It can also interpret model coefficients as indicators of feature importance. Elastic net optimization was used and the optimal value of alpha was 0.1. We got a training error rate of 51% and testing error rate of 50.7% using the logistic regression model which was the least amongst all the models.

Training Error	Test Error
0.51	0.507

5.2 Random Forest

Random Forests is an ensemble learning method that uses multiple decision trees. The class that is most outputted by these individual decision trees is used as the predicted class. It can handle all types of features and requires very little pre-processing. Hyperparameter tuning was performed to get optimal number of trees, criteria for information gain, depth and minimum samples in the leaf node. The model had a training error rate of 51% and testing error rate of 51.4%. Random Forests is generally used to counter low bias high variance situations but, in this case, it's the opposite.

Training Error	Test Error
0.51	0.514

5.3 XGBoost

We tried bagging using Random Forests. Extreme Gradient Boosting or XGBoost is a boosting algorithm where several optimization techniques are combined to get results within a short span of time. The objective is to minimize the loss function of the model by adding weak learners using gradient descent. Hyperparameter tuning was performed to get optimal gradient boosted trees, learning rate and depth. The model had a training error rate of 54.6% and testing error rate of 51.1%.

Training Error	Test Error
0.546	0.511

5.4 Neural Networks

Neural networks are designed to recognise patterns and it could help in improving the high bias that we have for this problem. The dataset used for this technique was the original dataset with 188 features as the model has the capacity to establish complex relationships between these features. A Multilayer Perceptron model was used with 5 layers. We added a 20% dropout after the first layer. The activation function used was 'relu' for intermediate layers. The model was compiled using 'sparse_categorical_crossentropy' loss and 'adam' optimiser. The model had a training error rate of 51.1% and testing error rate of 50.8%.

Training Error	Test Error
0.511	0.508

5.5 LDA & QDA

Linear Discriminant Analysis is generally used as a dimensionality reduction technique. It focusses on projecting features in higher dimension space to a lower dimension. For prediction, it uses the Bayes theorem to get the probabilities and the class with the higher probability is considered as the output response. Just like logistic regression, scaling is necessary for this model, but scaling did not have much impact to the accuracy and recall scores. The LDA model had a training and test error rate of 51% and 50.8% respectively. The Quadratic Discriminant Analysis (QDA) model was also tested and it had higher training and test error rates of 51% and 50.8% respectively.

LDA Training Error	LDA Test Error
0.510	0.508

5.6 Naïve Bayes

We also tried the Multinomial Naïve Bayes algorithm which assumes independence among the features. Sometimes, the simplicity of the model can handle noisy data. It is also robust to missing data. The model had a training error rate of 53.2% and testing error rate of 52.9%.

Training Error	Test Error
0.532	0.529

5.7 Ensemble

Finally, an ensemble of multiple models was also tested. The predicted probabilities of Logit, Random Forests, XGB, Naïve Bayes and LDA were used. Maximum and average probabilities were tested for different combinations of models but it did not improve the results. The error rates increased drastically with the models compounding its errors.

6 Discussion

The Test error rates are the lowest for Logistic Regression, LDA and Neural Networks. Naïve Bayes has the highest Test error rates which could be due to giving more importance to Draws. We tried giving more weight to 'Draw' for other models but it resulted in higher recall loss for Wins and Losses.

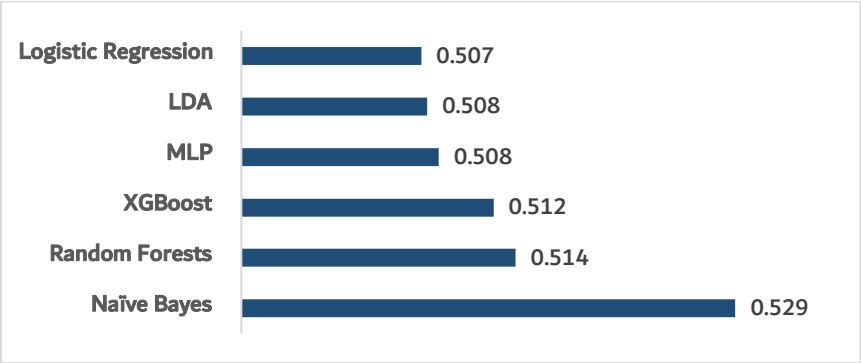

Figure 2: Test Error Rates

The confusion matrix of the best performing model, i.e., the logistic regression model, shows 81% recall when predicting home team wins. For home team losses, it has a prediction recall of 45%. We notice that 'Draw' is mostly ignored by the model. This is quite interesting as 'Draw' is the starting state for any match. This is one of the challenges of predicting soccer match outcomes compared to other forms of sports. In fact, this trend was observed for all the models as they drastically underpredicted draws. Naïve Bayes was slightly better off in this aspect. Naïve Bayes has better accuracy in predicting away team wins. The other models were very similar to the Logistic Regression model in terms of predicting the response variable.

		Logistic Regression Predicted		
		Away	**Draw**	**Home**
Actual	**Away**	2,701	50	3,210
	Draw	1,478	43	3,479
	Home	1,498	35	6,730

		Naïve Bayes Predicted		
		Away	**Draw**	**Home**
Actual	**Away**	3,405	387	2,169
	Draw	2,150	366	2,484
	Home	2,472	497	5,294

Interestingly, removing cup matches improved the accuracy of the model. This is possible because cup matches are played between teams from different leagues and their form may not necessarily be indicative of their strength. Also, top clubs tend to not prioritize some cup matches and they don't play their main team. This could indicate possible noise that these matches add to our model's prediction and we decided to remove data points that were related to cup matches.

Prediction was also evaluated by converting the problem into binary classification. We tried to see if we could predict Wins vs No Wins. This resulted in a much better test error rate of 0.364 for Logistic Regression. The recall was 80% for Wins and 43% for Loses. As we can see here, changing it into binary classification did not improve the recall in any way. This further indicates that the data does not have enough identifiable patterns to separate the Draw class.

		Binary Case: Logistic Regression Predicted	
		Home Win Y	**Home Win N**
Actual	**Home Win Y**	8,666	2,295
	Home Win N	4,715	3,548

7 Conclusion

Since this dataset is a part of Kaggle competition, we could compare the average log loss of the predicted probabilities with the Bookmakers. The loss metric obtained from logistic regression model was 1.013, which was not too far from the loss metric of the Bookmakers which is 0.973. There are several findings which indicate that beating the Bookmaker's odds is very difficult. The Basic model here could be predicting all games as the 'Home' team winning, since that is the highest response observed. This would give us an accuracy of 43%. Our best model is able to predict 6% better than this basic model. One reason for Logistic regression performing as well as complex models such as neural networks could be the data not having good patterns and a linear approach could be the simplest one that works here.

There are a few things that could be done to improve prediction accuracy here. Time as a dimension has not been considered here. Teams' strengths change overtime as it integrates changes in team composition, match tactics, injuries and weaknesses. Using time to give more importance to recent historical matches could help in improving accuracy. Additional data such as injuries/suspensions to key players, key player purchases and player specific form could further improve the model's accuracy. A state-space model such as the Kalman filter could be used by incorporating and giving more weight to the initial state, i.e., 'Draw'.

8 References

[1] Contributors to Wikimedia projects. “Entropy (Information Theory) - Wikipedia.” Wikipedia, the Free Encyclopedia, Wikimedia Foundation, Inc., 9 July 2001, http://en.wikipedia.org/wiki/Entropy_(information_theory).

[2] “Football Match Probability Prediction | Kaggle.” Kaggle: Your Machine Learning and Data Science Community, https://www.kaggle.com/competitions/football-match-probability-prediction/overview.

[3] “Football Prediction Performance: How to Calculate Hit-Ratio and Log-Loss | by Octosport.Io | Geek Culture | Medium.” Medium, Geek Culture, 25 Nov. 2021, https://medium.com/geekculture/football-prediction-performance-how-to-calculate-hit-ratio-and-log-loss-1e5e22310497.

[4] Goddard, John, and Ioannis Asimakopoulos. “Forecasting Football Results and the Efficiency of Fixed-Odds Betting.” Journal of Forecasting, no. 1, Wiley, Jan. 2004, pp. 51–66. Crossref, doi:10.1002/for.877.

[5] Harper, Justin. “Data Experts Are Becoming Football’s Best Signings - BBC News.” BBC News, BBC News, 5 Mar. 2021, http://www.bbc.com/news/business-56164159.

[6] Hastie, Trevor, et al. The Elements of Statistical Learning. Springer Science & Business Media, 2013.

[7] Louzada, Francisco, et al. “Predicting Match Outcomes in the English Premier League: Which Will Be the Final Rank?” Journal of Data Science, no. 2, School of Statistics, Renmin University of China, Mar. 2021, pp. 235–54. Crossref, doi:10.6339/jds.201404_12(2).0002.

[8] octosport.io. “How to Compute Football Implied Probabilities From Bookmakers Odds | by Octosport.Io | Geek Culture | Medium.” Medium, Geek Culture, 30 Mar. 2021, https://medium.com/geekculture/how-to-compute-football-implied-probabilities-from-bookmakers-odds-bbb33ccf7c1d.

[9] “Wikipedia:WikiProject Football/Fully Professional Leagues - Wikipedia.” Wikipedia, the Free Encyclopedia, http://en.wikipedia.org/wiki/Wikipedia:WikiProject_Football/Fully_professional_leagues.

9 Appendix

9.1 Dataset Summary

Total number of matches	104,461
Total number of leagues	834
Total number of non-cup leagues	623
Average number of teams per league	18
Average Home Goals per match	1.52
Average Away Goals per match	1.24
Average Historical home goals per match	1.37
Average Historical Away goals per match	1.33

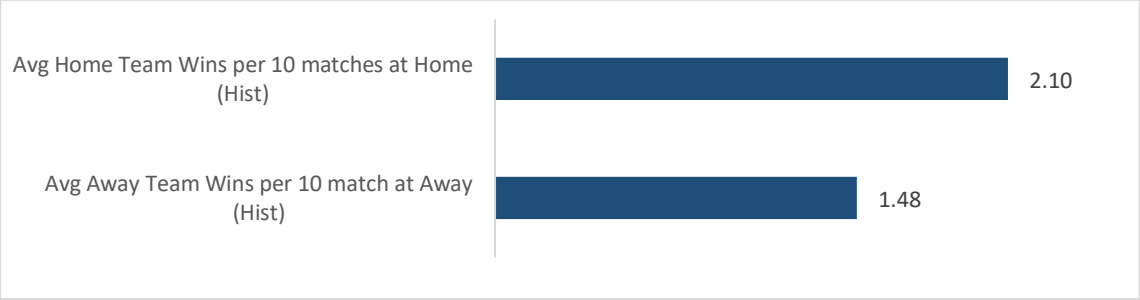

Figure 3: Win statistics showing home team advantage

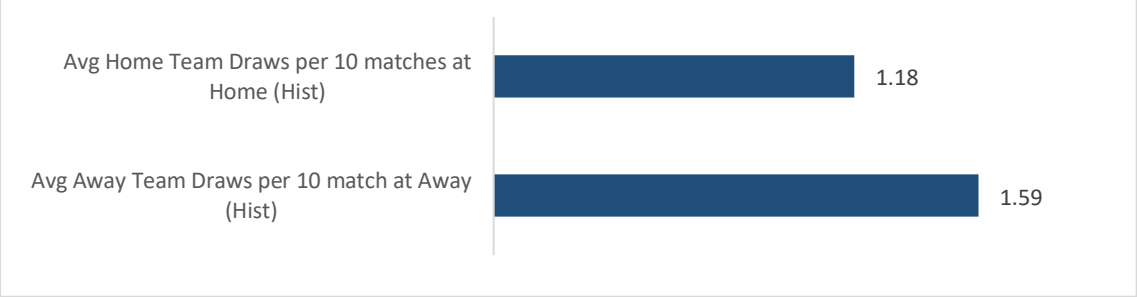

Figure 4: Draw statistics for Home/Away teams

9.2 Correlation matrix of the features (after conversion of the historical features):

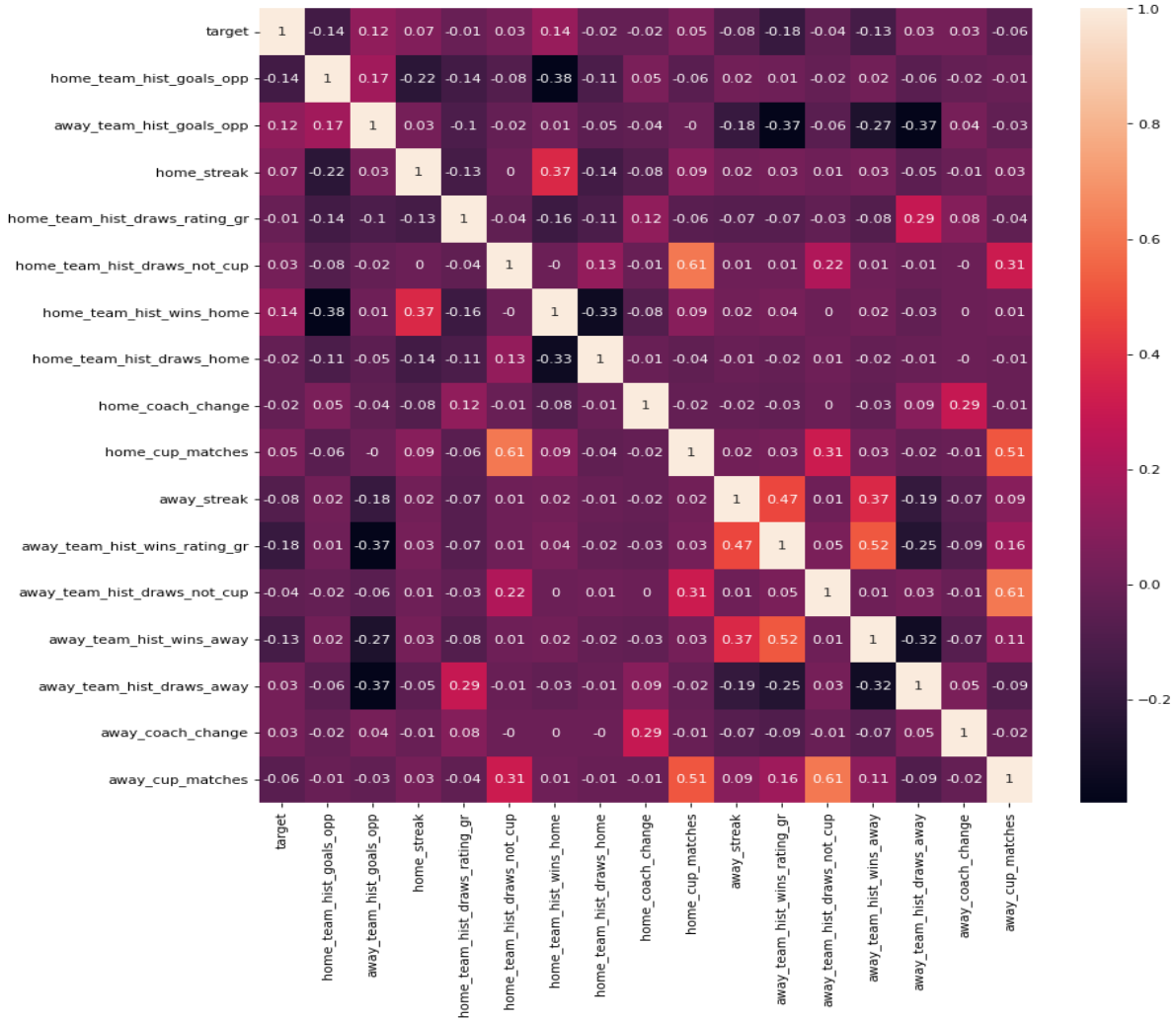

Figure 5: Correlation matrix

9.3 Log Loss metric calculation:

Here we try to measure how far the probability is from 1 when the result is true. A model that tells us the away team wins with 75% probability is better than a model that tells us away team wins with 40% probability. The value of the log-loss is between 0 and minus infinity.

$$LogLoss = \frac{1}{N}\sum_{i=1}^{N}(\log(p_{Win})\,\mathbf{1_{Win}} + (\log(p_{Draw})\,\mathbf{1_{Draw}} + \ (\log(p_{Lose})\,\mathbf{1_{Lose}}))$$