ST309 Group Project Report: Predicting Football Results*

[redacted] (50% contribution)
London School of Economics

[redacted] (50% contribution)
London School of Economics

2019

February

*Thank you to Cheng Qian and Dr Qiwei Yao for support in writing this paper.

1 Introduction

The global gambling market gross gaming yield [1] is estimated to be $495 billion in 2019 (Statistica 2019), while betting turnover in the 2018 FIFA World Cup was estimated to be €136Bn (FIFA 2018). From the bookmakers' point of view, it is essential that they have accurate models to predict the probability of certain results such that they can make profit, and not be beaten by customers. The opportunity for data analytics lies within the fact that the odds often reflect the market rather than the true probability - for example, most people may think Manchester United will win their next match, and hence the odds reflect this, but that often may not reflect the true probability of them winning. Furthermore, there is a wealth of data collected on every football game in all the top leagues in Europe, from who played, to the prior results of each team; data that we can use to investigate to what extent the result can be predicted, if at all.

However, results are often hard to predict. Leicester won the Barclays Premier League in 2016, yet at the start of the season, the odds of them doing so was 5000/1 at many bookmakers. Similarly, Greece won the European Championships in 2004 being priced at 150/1 (Keogh 2016), and the FA cup frequently has low-ranked team knocking out top tier teams. Predicting results from football games is thus an important, profitable, but difficult challenge.

Prior literature on this topic is interesting, since a wide array of approaches are taken. One approach is to use ELO ratings in a logit framework (Hvattum and Arntzen 2010). Fivethirtyeight, a statistics website, predicts results using Poisson models to predict the number of goals scored by each time, and thus indirectly predicting the result (Five Thirty Eight 2019). Many papers have looked to use Bayesian network techniques (for example, Joseph, Fenton, and Neil 2006) . While some, citing the randomness of the actual number of goals scored as a potential issue, use the expected number of goals scored instead (Herbinet 2018).

We have picked this topic since it is an area of interest to both of us, so believe our knowledge should aid us in analysing the problem. The goal of our study is to identify key variables that can be used in a regression to predict the result of a football match, using the novelty of having video game ratings of the players to see if these can also be used as a predictor. Such variables are to include the form of each team, their current points tally, as well as their performance in the previous season. Once we have identified the key variables, the aim is to

[1] This is the money that the bookmakers keep after paying out on bets made by customers that won (i.e. the stakes made by customers minus the winnings paid to them).

split the data into training and testing data, and then fit our model on the former and test it on the latter. We will do this for each English Premier League season between 2008/2009 season and 2015/2016 season. To give an idea of how successful we can expect to be in doing so, it is apt to consider previous attempts to predict football results. In the 2012/2013 Premier League season, bookmakers successfully predicted (home win, away win, or draw) 52.36% of results (BBC Sport 2013). Of course, the extent and success of our analysis may not reach the depth of the bookmakers', so we should not be disappointed if we do not see such a success rate.

2 Description of Data and Methodology

We obtain the data from Kaggle[2]. The data file type is SQLite, which required use of the R package 'RSQLite' to be able to access the data. The data itself is very rich, containing the results of football games from 2008 to 2016 in 11 European leagues. The focus of this project is on the English Premier League. Details for each game include basics such as the number of goals scored, but also the starting 11 for each team, the way in which the goal was scored and the betting odds that top bookmakers offered for that game. The qualitative variables (e.g. how the goal was scored) and some other variables (e.g. possession) copy over poorly using the RSQLite package so were not used in the analysis.

In addition to game data, the dataset has further layers with details on player and team ratings from the sports video game 'FIFA'. This adds an additional dimension to the analysis, as we can look to see whether or not these ratings can predict real outcomes. In addition, we created the following variables:

- Average FIFA player rating of the starting 11
- Whether or not there was a 'superstar' (a top-10 rated player in the Premier League for the respective season, according to FIFA ratings) in the starting 11
- Form of the team based on the past 5 games
- League position in previous season
- Current number of each team's points as of *before* the match

We first created dummy values for home win, away win or draw which was done by comparing the number of goals scored and depending on who scored the most, assigning 1 to the respective

[2]https://www.kaggle.com/hugomathien/soccer

variable. For example, if the home team scored more than the away team then the dummy value for home win would be 1, and for the other two dummies the value would be 0. One of the predictive variables used is the average player rating of the starting 11 according to FIFA's player ratings; this was more challenging to obtain. The player ratings were stored along with other variables in a dataframe called *Player_Attributes*. For each season there are multiple ratings available for each player, since the ratings in FIFA are updated at times throughout the season to reflect their performances. We created a dataframe which has the ratings for each player at the start of each season. Then, for each game, the season ratings for each player were obtained. The ratings for the starting 11 were then averaged and stored in the column corresponding to the average player rating for the home team. The same was done for the away team. Splicing was the main method used to do this.

To create the star player variable, we used the FIFAIndex player rating database (FIFAIndex 2019) to find for each season the top 10 rated players in the English Premier League. We then obtained the player ID number for each of them. Using the which function, we found the index position in which these players started and for those index positions assigned a '1' to the star player column to indicate this. Again, this was done for both the home and away teams.

The assigning of the previous league position was only possible by manually inputting each team's previous position for each season, using the English Premier League's online archive to find historic Premier League tables (Premier League 2019).

Retrieving the form from our dataset was difficult, as it is a moving average of the points tally of each team's chronological matches. Therefore, the window of data considered must be shifted with each calculation, adjusting for what stage of the season that particular match is at. The main challenge was creating a vector of the points achieved, for each team. Initial attempts included adding the home and away points vectors, but this did not work as each vector was of length 19, rather than 38, and it did not account for the ordering of the matches. After experimenting, we developed the following solution:

1. For team i, retrieve the points won by the home team in each of their 38 matches and store this as an array.

2. The points won by team i at home in this array are correct (19 elements).

3. For the games in which team i were away, if the element of the array is 3 this means that

team i lost (the home team won) so we change this value to 0.

4. For the games in which team i were away, if the element of the array is 0, that means team i in fact won (i.e. the home team lost, and team i were the away team), so we change this value to 3.

5. Once we have our ordered array of points, we calculate a moving average (with the rollapplyr function), and then assign each of these to *home_team_form* if team i is at home, and similarly for away.

Note that *home_points* becomes a locally defined variable within the function, representing the points of team i, both home and away.

We ran the model at this stage and were unimpressed with the level of predictions. We realised that by adding the current points tally we may be able to improve the accuracy of the model by capturing the more general long term form of the teams. To do this, we included a variable for the current points tally for each team, as at the start of each match. To do this, we began by creating the points vector, as described in steps (1) - (4) above. Once we had this vector, we calculated the current points of the team via a cumulative sum. However, this represented the points tally of each team *after* that match, rather than before (e.g. for Match 1, the cumulative sum would be the points won from that match, not the amount of points they had prior to it, i.e. 0). Therefore, we shifted the cumulative sum across by 1, meaning the 2nd - 38th entries of each team's current points were correct, and then assigned the first entry to be equal to 0. This was then split into the current points of the home and away team.

Throughout the process of creating new variables to use as predictors, to avoid for-loops we used a combination of defining functions and *lapply* to speed up the process and improve computational efficiency.

The dataset is mostly clean, however FIFA player ratings are not available for every player in the starting 11 of the real life games. While a player may have data available for the 2009 season they may not have it for the 2010 season. This means that the average player rating when they start in the 2010 season has to be taken without them. However, star players all had their ratings available which means that there were no significant outliers that were omitted so we believe the average ratings calculated without missing players to be generally accurate.

The data analysis overall was designed to predict the result of football games; that is, whether the home team or away team won, or the match resulted in a draw. To do this, we first created the aforementioned predictive variables that we thought would have an effect on the result. We created three different forms of training data. First, for each season, we split the data 50-50 into training and testing data, using a random sample (what we will refer to as the 'random approach'). Table 1 shows the resulting regressions for the 2012/2013 season. The table, along with other tables in this paper, were generated with Stargazer (Hlavac 2018). For the second approach we split the data in half, with the first half of the season corresponding to the training data, which we used to predict the second half of the season (what we will refer to as the 'half-season split'). The final approach was to use continually updated training data (what we will refer to as the 'rolling approach'). In each progressive round of fixtures, the training window expanded to encompass all previous rounds of fixtures. For example, for the 8th round of fixtures, the training data would be the previous 7 rounds. For the very first round of fixtures, for which there is no previous data to go from, only previous league position and whether the teams had a 'superstar' player were used as a predictor for the match outcomes. For the latter two versions, they were done only on the 2012/2013 season. This was done because we have the bookmakers success rate to compare the performance against, and these two approaches reflect more closely how predictions would be made by bookmakers - they both make predictions based on training data exclusively from the past.

For each of our possible outcomes (home win, away win, draw) we fit a classification tree to our data, and then pruned it accordingly to discover which variables were deemed significant. With these variables, we fit a logistic regression using our training data, and then used this to predict the outcomes of the matches within our testing data, similar to the methodology used in the exercises (for example Exercise 4 Question 4). As our regressions produced 3 probabilities for each of the possible outcomes, our actual prediction was the result corresponding to the maximum of these probabilities.

Before we proceed to analyse our results, it is apt to note that the pruned trees sometimes stated the 'best' amount of terminal nodes to be 1 (this was especially the case for draws, since they are difficult to predict). In this case, the variables *home_team_form*, *away_team_form*, *home_prev_pos*, *away_prev_pos*, *home_current_points* and *away_current_points* were used in the respective logistic regression. The choice of these variables encompasses those which were frequently used in our other regressions, and thus seemed to play a big part in determining our

Table 1: Logistic regression for 2012/2013 Season

	Dependent variable:		
	Home win	Away win	Draw
	(1)	(2)	(3)
Home team form		0.033	0.200
		(0.336)	(0.369)
Away team form		-0.814^{**}	0.821^{**}
		(0.369)	(0.399)
Home previous position	-0.122^{***}	0.006	0.092^{**}
	(0.030)	(0.039)	(0.042)
Home current points	0.108^{***}	-0.076^{*}	0.045
	(0.028)	(0.045)	(0.046)
Away previous position	0.0003	-0.051^{**}	0.001
	(0.010)	(0.025)	(0.025)
Away current points		0.048^{**}	0.003
		(0.022)	(0.022)
Constant	-0.230	1.016	-4.171^{***}
	(0.546)	(1.082)	(1.282)
Observations	190	164	164
Log Likelihood	-111.182	-94.074	-86.463
Akaike Inf. Crit.	230.363	202.149	186.926

Note: *p<0.1; **p<0.05; ***p<0.01

predictions.

3 Results Interpretation and Evaluation

A summary of our results for when the training data was a random sample can be found in Table 2.

Table 2: Correct Predictions for Each Season and Accuracy (Random Method)

Season	Games Predicted	Home Win	Away Win	Draw	Accuracy
2008/2009	166	60	24	4	53.01
2009/2010	166	73	8	1	49.40
2010/2011	166	70	11	9	54.22
2011/2012	162	59	22	3	51.85
2012/2013	166	54	23	7	50.60
2013/2014	144	53	30	1	58.33
2014/2015	166	64	23	2	53.61
2015/2016	149	44	19	6	46.31

At a first glance, our results make intuitive sense. That is, home wins are by far the easiest to predict, with draws causing the most difficulty. Furthermore, we can be pleased with the level of accuracy achieved by our model, ranging from 46.31% to 58.33%. As previously mentioned, in the 2012/2013 season, for example, the bookmakers successfully predicted 52.36% of results, whereas we achieved success of 50.60% for that particular season. So, given the relatively extensive resources the bookmakers have, as well as the level of detail they go into, our model performs well. The total, across-season average accuracy rate of our predictions is 52.14%.

Now we know that our model is accurately predicting results as a whole, it is an interesting exercise to analyse what results we are predicting relative to the actual results that occur. A summary of this can be found in Figure 1, where figures represent the cumulative amounts across all seasons.

Note that the 'actual' results above are exclusive of those that we did not predict, which we will

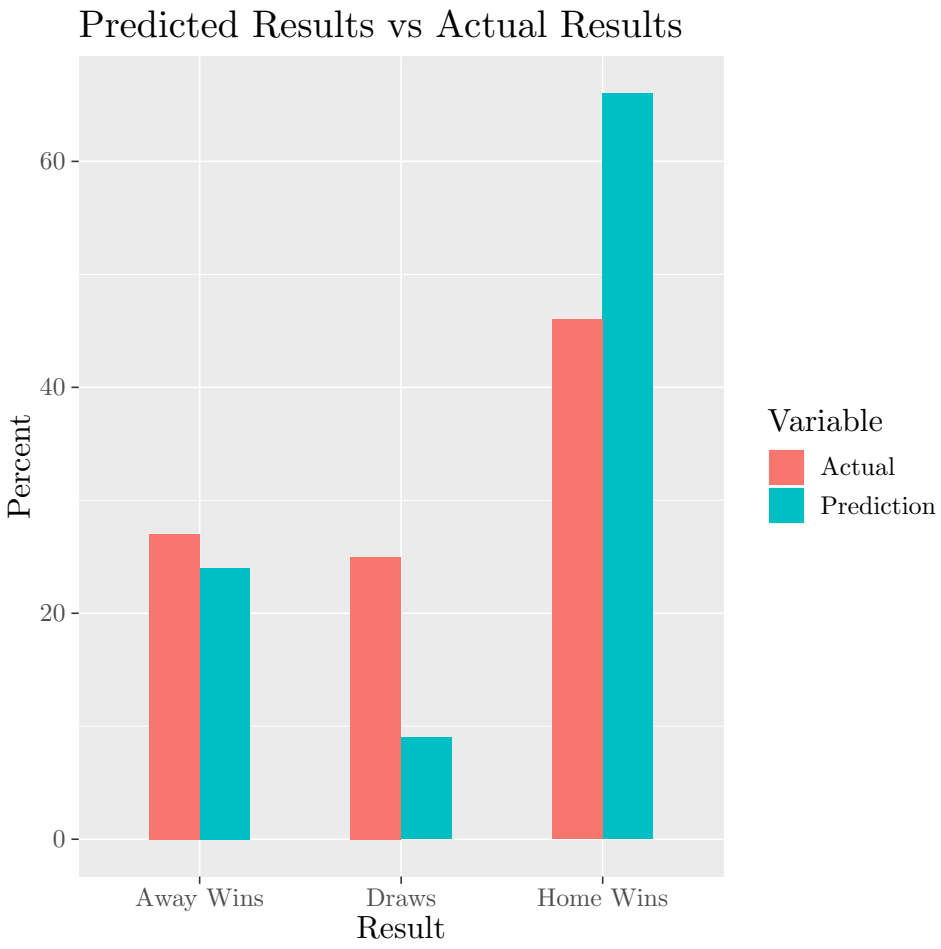

Figure 1: Predicted Results vs Actual Results (average for all seasons using random approach)

discuss later. We see that our model overestimates the amount of home wins, and underestimates the amount of draws. Meanwhile, the proportion of away wins predicted is very similar to those that actually occurred.

Although our results are impressive, it is important to evaluate our model further. A nuance underlying our accuracy rate is that none of our predictions are for the first 5 rounds of matches of each season. This is because our form variable is a moving average of the team's points tally from their previous 5 matches, and thus the first 5 form variables for each team are always N/A. So, rather than predicting 190 matches per season, we in fact predict a maximum of 166 (within the testing data we have 24 matches with form N/A, and since we set a seed, this value does not vary season-by-season). Therefore, we are not predicting the matches that are arguably the hardest to predict - those at the beginning of each season, where we are yet to see how newly signed players will perform and how well the team is likely to play that season etc., which is exacerbated if the team is newly promoted. For this reason, the difference between our accuracy rate, and that of the bookmakers, is likely to be slightly wider than seen at first glance. Sometimes we predict fewer than 166 matches per season, which is caused by a small number of N/As occurring in other variables; for example, the *average_player_rating* variable is sometimes N/A, due to the lack of a FIFA rating for a player within our dataset. Note that in order to see how many predictions were made for the season, we calculate $190 - sum(is.na(prob.draw))$, or the same but for *prob.home.win* or *prob.away.win*. When the probability is N/A, our model predicts neither a home win, draw, or away win, i.e. makes no prediction at all.

The FIFA ratings of a team's players was rarely deemed a useful variable by our model. Although surprising, this does make sense, for several reasons. Firstly, young players' FIFA ratings are often incredibly low, which may not reflect their true performance and ability. Also, the ratings we have used are as at the *beginning* of each season, and thus are constant throughout the season. So, a player at the start of the season who has a low FIFA rating, will see his rating stay the same - the ratings do not continuously change match-by-match to reflect true performance levels. Finally, the FIFA ratings of teams within the Premier League tend to be relatively close to each other; unlike if a Premier League team was playing a League 2 team. The indicator variable for whether each team possessed a 'superstar' player based on FIFA ratings, however, was used in several of our regressions as a predictor variable.

Other points to consider include the randomness of our simulations. Of course, changing the seed

will change our results slightly, but the large sample size means this is not a major issue. Perhaps a more worrying concern is the possibility of multicollinearity. For example, *home_current_points* is likely to be correlated with *home_team_form*, variables which, at times, were included in the same regression. The extent of any possible multicollinearity is unknown however; if it is only moderate, then it may not be problematic, though it will be more so if the correlation is severe.

It is a useful exercise to compare how the performance of the model changed when the training data changed, which Table 3 summarises. For the half-season split, the accuracy rate is similar to when using the random approach, for the 2012/2013 season, at 50.00%. The small difference between this rate and that of our random approach shows our model does not greatly suffer when using the games in chronological order. We can see that when using the rolling approach, the performance is worse, correctly predicting only 47.1% of results. It is apt to note that for our rolling data approach, we didn't fit a classification tree for each round of fixtures due to time constraints. Instead, we used explanatory variables that were used frequently in prior regressions, such as form. For the first round of matches, we could not use the form and current points variables. Whilst for the first five rounds of matches, we could not use form.

Table 3: Accuracy for Each Training Data Approach with Bookmaker Comparison (2012/2013 Season)

	Random	Half-season Split	Rolling	Bookmakers
Performance (%)	50.60	50.00	47.11	52.36

The difference in performance of the rolling approach will be mainly down to attempting to predict the first few games of the season, where data on past performance is non-existent/very low. This means that the model will perform poorly at the start of the season, and hence bring down its overall performance. We expect the model to have a much better performance later on in the season because of using much larger training data than in the other cases, but this potentially improved performance is not large enough to compensate for the poor performance at the beginning of the season, so overall the performance is lower than the other methods. Indeed, this is what we see in our model's predictions. When predicting the first 5 rounds of fixtures, the accuracy rate was just 32%, compared to 49.39% for the remaining 33 rounds of fixtures. Comparing the difference in performance of the 'rolling' training data model to

the bookmakers' performance is perhaps the most revealing in assessing how well our model performs. The overall performance of 52.36% is greater than the 47.1% that we achieved in a similar approach, shedding light on the fact that the issue of predicting results is much more complex than what our model can extract.

4 Conclusion

We have been able to predict the results of football games to a relatively high standard, with an average accuracy rate of 52.14% (using the random draw training data approach), not too dissimilar from that of bookmakers. Given bookmakers' extensive resources, and the amount of time and analysis they devote to predicting results, our results are impressive. However, the most challenging games to predict (those at the start of the season) are omitted from our predictions. Our analysis offers a lot to learn, regarding the prediction of football results. As expected, it is clear that factors such as the form of each team and their current points tally, aid us greatly in attempting to predict the result. On the other hand, players' FIFA ratings is of less use, though is still a helpful indicator in some cases. Although, it is important to remember the possibility of severe multicollinearity and the effect it would have on the reliability of our results.

To test how well our model could be used in the real-world setting, we attempted 2 variations of training data that did not include data in the future. Firstly, our half-season split approach, which produced similar results to when we randomised the split into training and testing sets. Then, we proceeded to predict each round of matches, with all of those that preceded it, i.e. our rolling approach. This produced a slightly lower accuracy rate, as although it carries the advantage of having more training data to predict later games, the earlier games have less training data and are thus predicted with a lower accuracy.

In order to strengthen and improve our model, there are several advancements we could implement. We could seek to alter the functional form specification to account for nonlinearities, with the addition of quadratic terms. Also, we could include interaction terms in attempt to capture heterogeneous effects within our model. Perhaps the most substantial improvement we could make, is the creation of additional variables. For example, a form variable that places more weight on recent results than more distant ones (perhaps via an Exponentially Weighted Moving Average model). Another possible example is a variable for the time elapsed since the teams last played. If one team's last match was 3 days ago, and their opponent's was 2 weeks

ago, this fitness aspect may have an effect on the result.

A further, albeit minor, improvement that we could make, is fixing the small amount of N/A values that we have (apart from those within the form variable, which are unavoidable). For example, some of the N/A values are caused by our data not having the FIFA rating for certain players, perhaps we could seek to manually enter these ourselves. The amount of N/As is relatively small though, and therefore does not discredit our results to a great extent.

References

BBC Sport (2013). *How good are Lawro's predictions?* (Accessed on 08/02/2019). URL: https://www.bbc.co.uk/sport/football/22596125.

FIFA (2018). *136Bn Betting Turnover and no suspicious betting behaviour at Russia 2018.* (Accessed on 01/02/2019). URL: https://www.fifa.com/worldcup/news/136bn-betting-turnover-and-no-suspicious-betting-behaviour-at-russia-2018.

FIFAIndex (2019). *Player Stats Database.* (Accessed on 02/02/2019). URL: https://www.fifaindex.com/players.

Five Thirty Eight (2019). *How Our Club Soccer Predictions Work.* (Accessed on 11/02/2019). URL: https://fivethirtyeight.com/methodology/how-our-club-soccer-predictions-work/.

Herbinet, Corentin (2018). *Predicting Football Results Using Machine Learning Techniques.* (Accessed on 11/02/2019). URL: 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-profressional-football-matches.pdf.

Hlavac, Marek (2018). *stargazer: Well-Formatted Regression and Summary Statistics Tables. R package version 5.2.2.* (Accessed on 10/02/2019). URL: https://CRAN.R-project.org/package=stargazer.

Hvattum, Lars Magnus and Halvard Arntzen (2010). "Using ELO ratings for match result prediction in association football". eng. In: *International Journal of Forecasting* 26.3. (Accessed on 01/02/2019), pp. 460–470. ISSN: 0169-2070.

Joseph, A., N.E. Fenton, and M. Neil (2006). "Predicting football results using Bayesian nets and other machine learning techniques". eng. In: *Knowledge-Based Systems* 19.7. (Accessed on 11/02/2019), pp. 544–553. ISSN: 0950-7051.

Keogh, Frank (2016). *Leicester City: Fan hopes to win £25,000 after 5,000-1 bet.* (Accessed on 01/02/2019). URL: https://www.bbc.co.uk/sport/football/35520842.

Premier League (2019). *Premier League Table, Form Guide & Season Archive.* (Accessed on 01/02/2019). URL: https://www.premierleague.com/tables.

Statistica (2019). *Global gambling market gross gaming yield GGY from 2001 to 2019 in billion U.S. dollars.* (Accessed on 01/02/2019). URL: https://www.statista.com/statistics/253416/global-gambling-market-gross-win.