An investigation into a predictive data model to forecast football match outcomes.

Jim Owen

PG Cert Applied Data Science Project

26/08/2021

Table of Contents

Abstract

For my PG Certificate in Applied Data Science project, I investigate whether it is possible to apply statistical analysis and data modelling methods to predict the outcome of a football (soccer) match. This report outlines the theory underpinning this project and discusses the investigation and development of the solution.

Introduction

My motivation for this project comes from a keen interest in sports analytics, and how data can be used to gain a deeper understanding about what is happening within a game. I wanted to explore if ‘Expected goal’ data, often used to help analyze past performances, could also be projected forward and used to predict future outcomes. As with any uncertain outcome, it is extremely difficult to determine in absolute terms what will happen within a game, so my objective with this project was to use predictive modelling techniques to try and discover the probability of each potential outcome. I hoped to be able to compare the probabilities produced by my own model to the odds available at sports bookmakers, in hope of gaining a ‘competitive edge’ by finding bets with a positive expected value.

Literature review and research

Making absolute predictions on a football match is extremely difficult, as Dyte and Clarke (2000) explain: “Although soccer fans might want outcomes with a high degree of certainty, natural variation in the scores makes such a task impossible.” However, it’s possible to estimate the probability of each outcome using the Poisson distribution (Mayer 1982, Karlis and Ntzoufras 2003). The Poisson distribution is a discrete probability distribution that describes the probability of the number of events (goals) within a specific time period (one game), when we know how many times this event would normally occur. This means that if we have an accurate ‘Predicted Goals’, a figure representing how many goals we expect a team to score in the game considered, we can use the Poisson distribution to calculate the probability that a team will score 1 goal, 2 goals, etc. For example, the table below illustrates how likely each number of goals is when the team’s ‘Predicted Goals’ is 1.43:

Table 1: Probability of different goal quantities when ‘Predicted Goals’ is 1.43

Number of Goals	0	1	2	3	4
Probability	0.23	0.34	0.24	0.11	0.04

Similarly, if the opposing team’s ‘Predicted Goals’ was 0.92:

Table 2: Probability of different goal quantities when ‘Predicted Goals’ is 0.92

Number of Goals	0	1	2	3	4
Probability	0.39	0.36	0.17	0.05	0.01

It is then relatively straightforward to matrix the two probability distributions to work out the probability of each score line. We can therefore calculate the probability of each result by adding together the probabilities of the score lines which satisfy those criteria:

Table 3: Probability of results

Home Win	Draw	Away Win

0.488	0.267	0.243

The challenge, therefore, when creating a model which predicts the probability of the outcomes, is to accurately assess the information available to create a ‘Predicted goals’ for each team.

According to (Mayer, 1982), the number of goals we would expect each team to score depends on three factors. The first is the attacking ability of the team trying to score. Greater attacking ability increases the number of goals we would expect them to achieve during a match. The second factor is the defensive ability of the team trying to prevent them from scoring. The greater the ability of the defending team, the more difficult it will be for the attacking team to score. Finally, the number of goals will also depend on where the game is played, as the team which is playing at home tends to have an advantage, due to familiarity of conditions (Pollard 2008) and officiating bias (Boyko, Boyko and Boyko 2007).

These three factors can be added into a simple formula to create a ‘predicted goals’ for the home team:

$$\begin{aligned}&Home\ Team\ Predicted\ Goals\\&\qquad = (Home\ Team\ Attacking\ Ability/Away\ Team\ Defensive\ Ability)\\&\qquad * Average\ Home\ Goals\ Scored\end{aligned}$$

Conversely, the number of goals we would expect the away team to score would be as follows:

$$\begin{aligned}&Away\ Team\ Predicted\ Goals\\&\qquad = (Away\ Team\ Attacking\ Ability/Home\ Team\ Defensive\ Ability)\\&\qquad * Average\ Away\ Goals\ Scored\end{aligned}$$

The average number of goals scored in the top European leagues, for both home and away, is 1.49 and 1.13, respectively. We can therefore substitute these into our equation above. Next, we must find a way to express, in numerical terms, the relative attacking and defensive strength of each team.

Lee (1997) discusses a simple way in which this can be done, using goal data from the previous season. In this model, the abilities of the teams are estimated by comparing the number of goals that the team scored and conceded during the previous season and comparing it to the mean for the league. For example, to calculate the attacking strength of the home team, the mean number of goals a team has scored at home during the previous season is taken and divided by 1.49, the mean number of goals scored by home teams. Similarly, the number of goals the away team has conceded is divided by the mean number of goals conceded by away teams, which is also 1.49. So, for a strong attacking team, scoring an average of 2 goals per game, and a weak team which conceded an average of 1.8 goals per game, the formula would be as follows:

Home Team Predicted Goals = (2/1.49) * (1.8/1.49) * 1.49 = 2.41

Note that, in this formula, a weaker defensive team will have conceded more goals, leading to a higher defense strength value. This means that we must multiply the attacking and defensive strength together, rather than dividing, but the result is identical to the formula above – teams which conceded more goals in the past will be projected to concede more goals in the match. Using this formula, we calculate a ‘Predicted goals’ of 2.41 the home team. We can repeat this to calculate the away team’s ‘Predicted goals’, and therefore the probability of each outcome for the match, using the Poisson distribution described above.

However, there are some clear flaws with this approach, which mean that the attacking and defensive abilities of the teams are not assessed as accurately as they might be. Firstly, taking the data from the entire previous season could be unrepresentative, as such a long sample may not take account of short or medium-term changes in a team’s form. By adjusting the window used to assess performance, it may be possible to improve the accuracy of the model.

In addition, using the goals scored/conceded as an expression of a team's ability is problematic. As (Petty 2018) identifies, "soccer is a low-scoring game and goals are a rare event. As a result, pure goals data can sometimes be misleading." Goals data alone makes no reference to the quality of the chances created within a match. For example, a team may have multiple shots and create several good scoring chances, but only finish with one goal. The other team may score a fortunate goal with their only shot of the game. If goals alone are considered, the teams will have performed identically, with an equal chance of scoring one goal in future games, an assessment most football fans would disagree with. (Skinner and Freeman 2009) find significant evidence that the best team does not always win the game. As they point out, this is exacerbated by the fact that "the scores which most frequently arise correspond to relatively high probabilities of a misleading outcome." My aim, therefore, is to build upon Lee's simple version of the model by finding a more robust and detailed way to assess a team's relative attacking and defensive performances.

The data I wish to use to make this assessment is 'Expected Goals' data, available on a number of sports websites, such as https://understat.com. 'Expected Goals' (xG) are designed to show how many goals a team 'should' have scored in a game, based on the scoring chances they created. xG is seen as "a way of assigning a 'quality' value to every attempt... The higher the xG - with 1 being the maximum - the more likelihood of the opportunity being taken." (Stanton, 2017) This 'quality' value is based on the historical success rate of similar attempts. For example, a shot which was taken far from goal at a difficult angle would be assigned a low value, as the data shows that success rates from such positions are low. However, a shot closer to the goal would be assigned a higher quality value, as the historical success rates are much higher. A team's xG for the game is the sum of all these quality-adjusted attempts made within the match.

Since xG accounts for the quality of scoring chances created, we would therefore expect a team's expected goals performance to be more repeatable over time than the actual number of goals, as it is less susceptible to biases due to a lucky or unlucky event. According to (Caley, 2017), there is indeed significant evidence that Expected Goals are a better predictor of future results than goals alone. xG can therefore be used to assess a team’s past performances in a more detailed manner. By replacing the average number of goals scored over the previous season with an xG average, we may be able to generate a more accurate assessment of the team’s relative ability, allowing us to set a more precise ‘Predicted Goals’ for each team.

If the team’s chances of winning can be estimated accurately, it may be possible to gain a competitive advantage in the sports betting market, by finding ‘value’ bets. A ‘value bet’ is a bet for which the expected return is greater than the amount waged, which, as (Petty, 2018) identifies, “occurs when probability of the event occurring is higher than that implied by the bookmaker’s odds”. For example, if a sports bookmaker was offering odds of 3/1 on a team, but the ‘true’ odds of the team winning were only 2/1, betting on that team would have a positive expected value. If bets with a positive expected value are made consistently, a gambler can expect to make a positive return over time. A successful football betting model will therefore illuminate profitable betting opportunities by estimating the ‘true’ odds of an outcome as accurately as possible, allowing a gambler to select bets offered by the bookmakers which have a positive expectation.

Although it may seem unlikely that a model such as this could out-predict multinational bookmakers, the model needn’t be a perfect reflection of outcomes in order to find value bets. Indeed, (Goddard and Asimakopoulos, 2004) find that bookmakers are not fully ‘efficient’ actors - meaning that they do not incorporate all available information into the odds they set. Despite the uncertain industry in which they operate, bookmakers aim to have equal liabilities for every outcome of an event, which guarantees a profit in line with their margin. In order to achieve this, bookmakers typically reduce the odds on the outcome on which they expect to have a substantial amount of money wagered, in order to reduce the relative attractiveness of the bet and therefore the total liability. Similarly, they increase the odds on outcomes which are likely to have too little money wagered, in order to increase the attractiveness of that bet to punters and ensure that their liabilities remain balanced. This means that the odds offered often drift away from the bookmaker's assessment of true probability, towards what the ‘average’ punter believes the outcome is likely to be. This means that any predictive model which can predict the outcomes of a match better than the general public may be able to find profitable bets where the public has overestimated/ underestimated a team's chance of success.

Data Cleaning and transformation

To investigate whether it is possible to build such a model, I examine a dataset with around 14,000 games, containing the expected goals achieved and conceded for both teams, as well as the actual result. In addition, I use a separate dataset containing the odds which were offered for those same games.

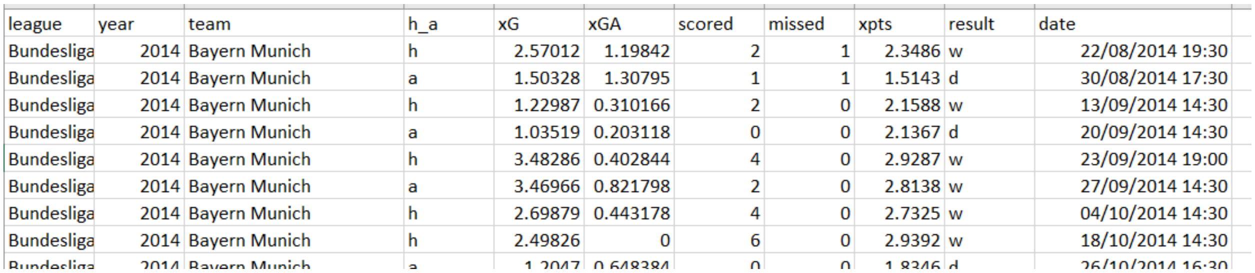
The first required step in this project is to clean and transform the data into a format which can be used for analysis. Both datasets were originally CSV files, however, they were in a heterogeneous format, which required extensive transformation before it could be used. I decided to use the Python Pandas library to transform the data, as this allows for fast computations and easy transformations. Before beginning the transformation, I also had to drop the columns I was not using in my analysis, as both datasets contained a lot of data not relevant to this project.

In order to create a 'Predicted Goals' and track the predictions of the model, I needed to know the following information for each game:

- The Home and Away teams
- The Date on which the game was played
- The xG achieved during the match by each team
- The number of actual goals and result of the game
- The attacking and defensive strengths for each team (for each version of the model)
- The odds offered by the bookmakers for that match.

As I was using the Pandas Python Library, all this data for the game needed to be on the same row. This presented two key challenges. The first was that, in the original CSV file, the data was presented for only one team at a time, as shown below:

Figure 1: Sample of Expected Goal Data CSV



league	year	team	h_a	xG	xGA	scored	missed	xpts	result	date
Bundesliga	2014	Bayern Munich	h	2.57012	1.19842	2	1	2.3486	w	22/08/2014 19:30
Bundesliga	2014	Bayern Munich	a	1.50328	1.30795	1	1	1.5143	d	30/08/2014 17:30
Bundesliga	2014	Bayern Munich	h	1.22987	0.310166	2	0	2.1588	w	13/09/2014 14:30
Bundesliga	2014	Bayern Munich	a	1.03519	0.203118	0	0	2.1367	d	20/09/2014 14:30
Bundesliga	2014	Bayern Munich	h	3.48286	0.402844	4	0	2.9287	w	23/09/2014 19:00
Bundesliga	2014	Bayern Munich	a	3.46966	0.821798	2	0	2.8138	w	27/09/2014 14:30
Bundesliga	2014	Bayern Munich	h	2.69879	0.443178	4	0	2.7325	w	04/10/2014 14:30
Bundesliga	2014	Bayern Munich	h	2.49826	0	6	0	2.9392	w	18/10/2014 14:30
Bundesliga	2014	Bayern Munich	a	1.2047	0.648384	0	0	1.8346	d	26/10/2014 16:30

Data in this format is of little use, as much of the data listed above is missing, including data for the opposing team. There is however, a home and away column (‘h_a’), which was populated with either an ‘h’, indicating that the game was played at home, or an ‘a’, indicating the game was played away. My solution for this was to split up the data into home and away and create a unique index for each row using the league and xG columns. With this unique index, I was able to merge the data so that the home and away teams appeared on the same row.

The next step was to track each team's performance over time, to allow for an assessment of a team’s attacking and defensive strength as discussed above. At this stage, it was not clear what the most effective assessment of a team’s performances would be. Although I knew that it would be taking an ‘average’ of the team’s xG performance within a certain window, it was not clear how many games this window should be. I believed that there would be a trade-off here, as a larger window of games would mean that the sample would be less biased by an easier/harder run of fixtures. On the other hand, a longer sample might not be sensitive enough to changes in the team’s form. These two contrasting forces implied a ‘maximization’ point which might be found through testing. In addition, I was unsure if the mean or the median was the best metric to use. Since I was unsure on the best way to proceed, I decided to create several metrics. I decided on an upper limit of 19 games, which is one total season. Using more than this number of games is unhelpful, as a team which was promoted/relegated could go through a whole season before the minimum criteria was met. Since the model relies on the measurement of relative performance, I first needed to divide each xG figure for the home team by the average home xG achieved and the away xG by the average away xG achieved.

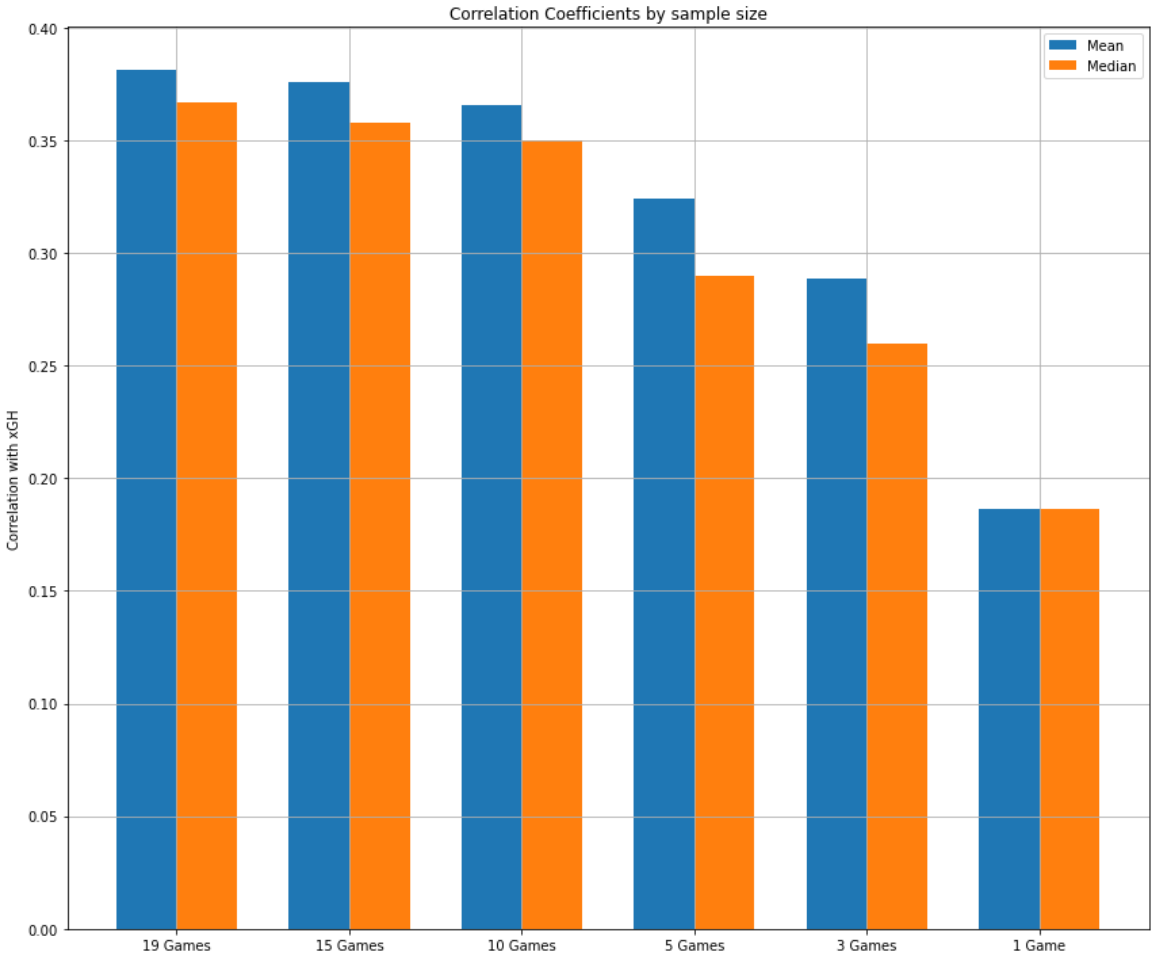
The second key challenge to overcome was the issue created by the time-series nature of the data. For a data model to be of any use, it must be able to make predictions based on data which was available before the game. We cannot, therefore, use the data generated from the match itself, as this would bias our results. This meant that computing a simple rolling average would be of no use, as it would include data from the game we are looking to predict. To solve this challenge, I first had to create an index for each game, to make sure that the games for each team could be tackled in chronological order. I then had to create a separate data frame to create the metrics. I took the mean and median for each team across the entire dataset, grouping by team name and applying lambda functions to track the mean and median relative performance over time. I repeated this process for the Home xG for, home xG against, away xG for, away xG against. I tracked both the mean and median of the relative performance values for the previous 19 games, 15 games, 10 games, 5 games, 3 games and 1 game. These were to be my metrics for assessing the team's relative strength. Once this was done, I updated all the indices, and merged them back into the original dataframe, using a unique index of home team + away team + counter, to ensure it was merged in such a way that the data from time t-1 was available at time t.

A further key challenge was to manipulate the odds dataset to allow it to be joined into the main dataframe. To do this, extensive transformation of the data was required. The initial CSV file contained over 30 columns, listing the odds available at different bookmakers, as well redundant information such as the number of corners and cards. I dropped the redundant columns, leaving only the home team, away team and date, as well as the odds of the game. When analysing the value counts of the dataset, it was clear that I needed to use the ‘Bet365’ odds, as this was the most complete dataset (10854 non-null records). I then had to create a function to re-name the teams, as the odds dataset used different naming conventions. To allow for the merge into the main dataframe, I again created a unique index using the date, home team and away team, which was common to both datasets. This ensured that the correct odds were attached to the correct game.

Initial Analysis of Metrics

As it was impractical to create and assess predictive models for all the potential relative strength metrics, I decided to use a simpler method to make an initial comparison. For this, I assessed the correlation between each of the metrics and the actual xG in the game itself. The stronger and more repeatable metrics will, in theory, be more closely correlated with the values they are looking to predict. Judging by the correlations, it became clear that the mean values outperformed the median values in every case:

Figure 2: Correlation coefficients with xGH by sample size



For this reason, I decided to drop the median as an assessment of a team's performance and focus on the mean. It was also clear that there was an interesting relationship between the number of games used and the correlation - although the accuracy increased with games added, there seemed to be diminishing marginal returns. However, it was also clear that the 1-game and 3-game means were so poorly correlated that they were of little use. To move forward, I decided to continue with the mean relative performances from the previous 19, 15, 10 and 5 games as my descriptions of the relative strength of the team.

Analyzing the usefulness of xG

Before using xG data to predict the outcome of games, I first had to check that it has descriptive power and is repeatable over time. I tested this in three different ways.

The first was to check for a simple correlation between the xG (both for and against) and actual goals across the whole dataframe. This returned a statistically significant coefficient of 0.64. This means that, as expected, the xG explains a large part of the number of goals in a match. However, it is also clear that there is a fair amount of natural variation, as a team can over/under perform vs their xG within a match itself. This is to be expected given a single football match is a small sample size.

To check if the correlation coefficient increased with sample size, I created a separate dataframe to take the average xG and actual goals (both for and against), grouped by team and year. This allowed me to assess the correlation between xG and Goals for each team over the course of the season:

Figure 3: Home xG season average (for) vs Home goal season average (for)

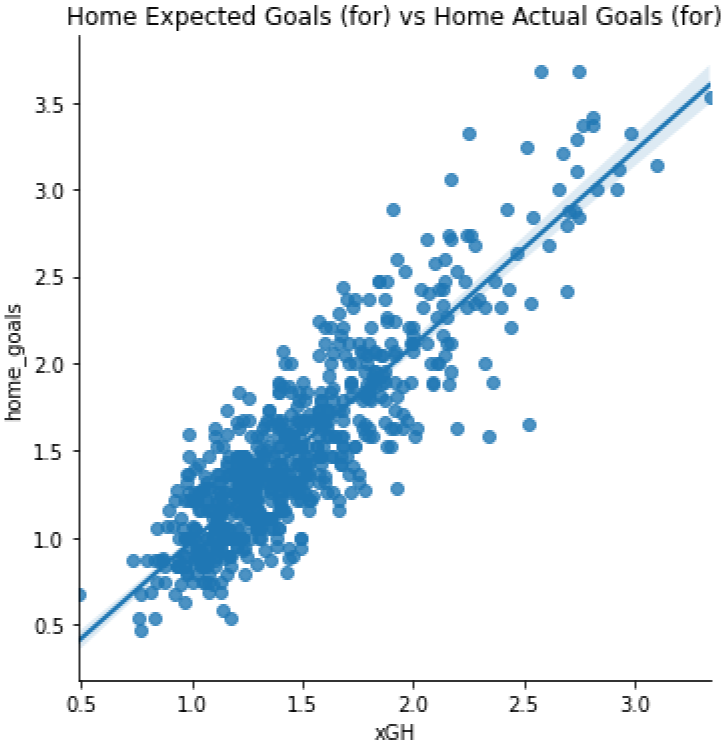


Figure 4: Home xG season average (against) vs Home goal average (against)

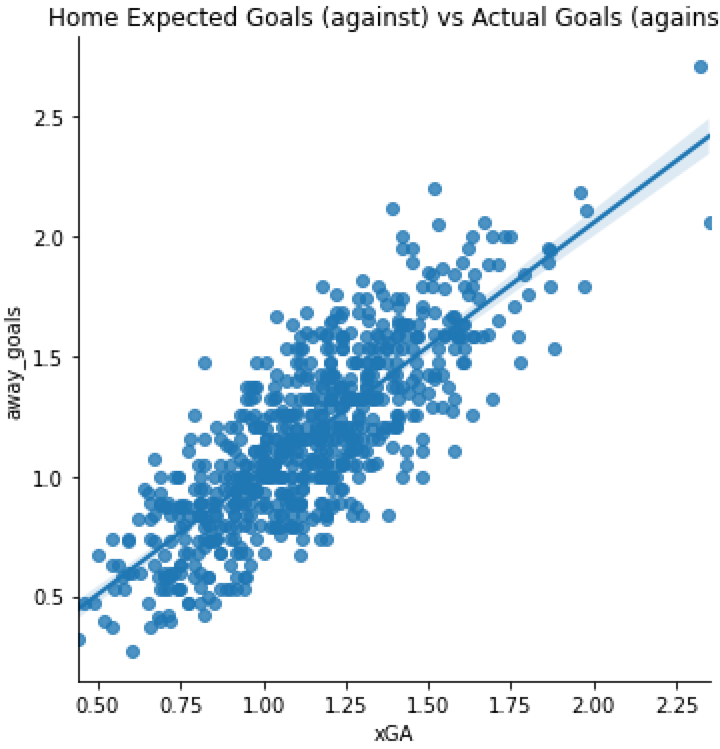


Figure 5: Away xG season average (for) vs Away goal average (for)

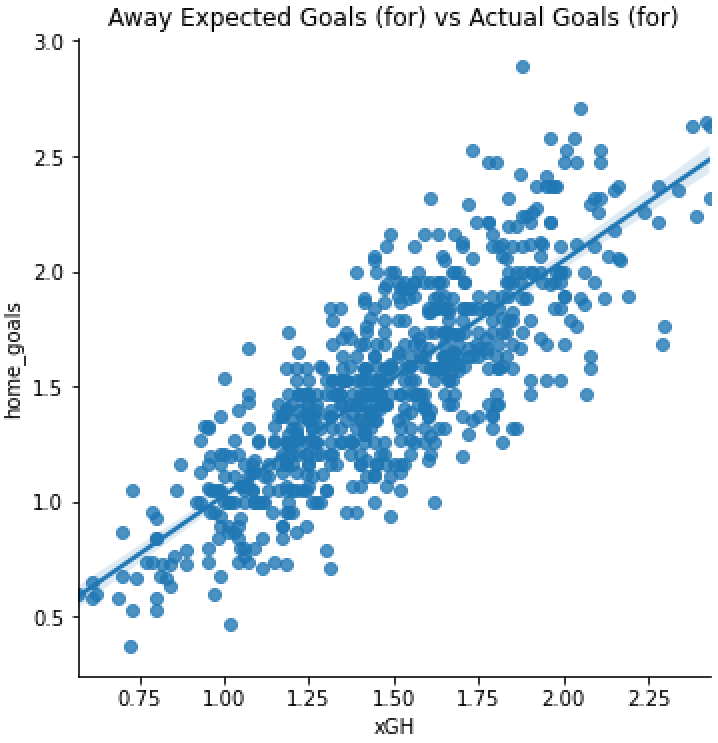
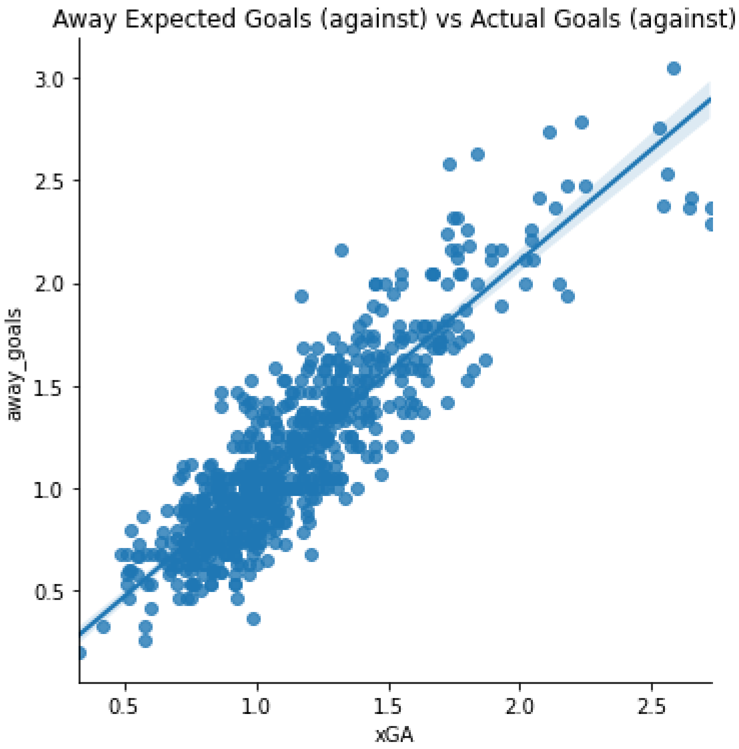


Figure 6: Away xG season average (against) vs Away goal average (against)



As we can see above, all four of the metrics studied are closely related, with only a few outliers. When checking the correlation coefficients, these were as follows:

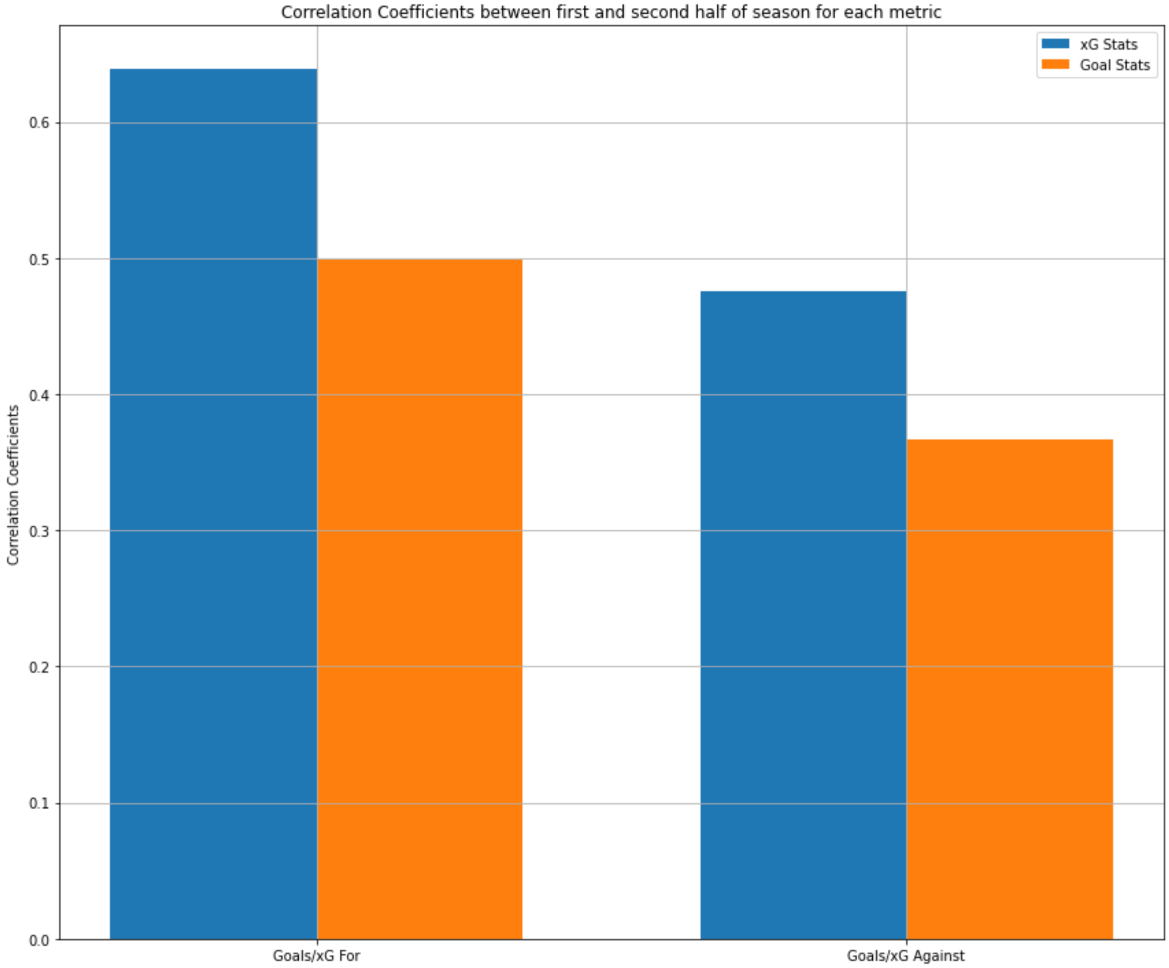
Table 4: Correlation between actual goals and xG (season averages)

Season average used:	Correlation Coefficient:
xGH and goals scored (home team)	0.87
xGA and goals conceded (home team)	0.80
xGA and goals scored (away team)	0.88
xGH and goals conceded (away team)	0.81

As we can see, in the long run, the number of goals a team scores will be similar to its xG values. Overall, it seems clear that a model which attempts to predict xG in the upcoming game will be able to also predict results over the long run, although there may be some fluctuation in the short term.

Finally, I assessed the 'repeatability' of xG by comparing performances across the season. To do this, I split the dataset in half, using the game counters as the conditional variable. I then took an average for both halves of the season, for both the xG data and the goals data (for and against). I then compared the two to see if a team's performance in the first half of the season predicted its performance in the second half. To assess the performance of goals vs xG, I took the correlations between the first and second half of the season and plotted them on the following chart:

Figure 7: Correlation coefficient between performances in first half and second half of season



As we can see, a team’s xG from the first half of the season is a better predictor of its performance in the second half of the season than the goal stats alone, both for and against. This is clear evidence that xG is more repeatable over time, an encouraging sign for our model. It also notable that this chart, as well as the correlation coefficients in table 4, imply that defensive performances are less repeatable than attacking performances.

From these tests, it’s clear that analyzing xG is a useful way to assess performance and predict future outcomes. It is also clear that xG can be used as a ‘target variable’, since goals and xG are closely related in the long run.

Creating a ‘Predicted Goals’ using machine learning

One way to create a ‘Predicted Goals’ figure is to substitute the relative performance means into the Lee model discussed above. Rather than use the goal average from the previous season, we can substitute in the 5, 10, 15 and 19 game relative performance means instead. Taking the 15-game mean as an example, the formula for Predicted goals would be as follows:

$$\begin{aligned} &\mathit{Home\ Team\ Predicted\ Goals} \\ &\quad = \mathit{Home\ Team\ Relative\ Attacking\ Performance\ mean}\ (15\ \mathit{game}) \\ &\quad * \mathit{Away\ Team\ Relative\ defensive\ performance\ mean}\ (15\ \mathit{game}) \\ &\quad * \mathit{Average\ Home\ Goals\ Scored} \end{aligned}$$

And for the away team predicted goals:

$$\begin{aligned} &\mathit{Away\ Team\ Predicted\ Goals} \\ &\quad = \mathit{Away\ Team\ Relative\ Attacking\ Performance\ mean}\ (15\ \mathit{game}) \\ &\quad * \mathit{Home\ Team\ Relative\ defensive\ performance\ mean}\ (15\ \mathit{game}) \\ &\quad * \mathit{Average\ Away\ Goals\ Scored} \end{aligned}$$

Although this formula is a useful approximation, it implicitly gives equal weight to the attacking and defensive strengths, with one simply multiplied by the other. As identified above, the defensive performances may not be as repeatable attacking performances, so an equal weighting may be misleading. To make an initial assessment of accuracy, I used the Pearson correlation coefficient to assess the correlation between the Home Team Predicted Goals (for each model version) and the Actual xG from the game itself. As discussed above, the significance of the relationship between goals and xG allows me to use xG as the ‘target’ variable - the variable we are trying to predict. Pearson correlation gives a correlation coefficient, but it also shows us the statistical significance of that correlation. If the statistical significance is above 0.05, it is not statistically significant at the 5% margin. The results were as follows, for the home team xG:

Table 5: Correlation coefficients between different metrics and xGH, including statistical significance

Metric	Correlation Coefficient with xGH	Statistically significant at 5%?
19 Game Mean	0.41	Yes
15 Game Mean	0.40	Yes
10 Game Mean	0.37	Yes
5 Game Mean	0.33	No

This correlation is encouraging, however, it may be possible to improve on the accuracy of the 'Predicted Goals' by treating xG as a target variable and using Linear Regression to assess the relationship between this variable and the metrics we have gathered. Taking the coefficients from this model, we can then build our own formula to estimate the 'Predicted Goals' for the match. Using the 15 game mean as an example once again, I set up the following feature and target variables:

Feature variables: [xGH]
Target variables: [Home Attacking Strength (15 game mean), Away Defensive Strength (15 game mean)]

I then used Linear regression to understand the relationship between the variables. Linear Regression determines a linear function between the feature and target variables that best describes the relationship between them. To do this, a 'line of best fit' is plotted which minimizes the total distance between the line and the observed data points. The output of the linear regression is therefore an expression for xG in the following format:

$$xG = b + b1x1 + b2x2$$

Where *bo* is the intercept term, (*x1*) is the Home Attacking Strength (15 game mean) and (*x2*) is the Home Defensive Strength. (*b1*) and (*b2*) represent the coefficients for each of these two variables. As this equation expresses the relationship with xG, we can therefore use it to create an expression for 'Predicted Goals':

$$
\begin{aligned}
&Predicted\ Goals\ (\ Home) \\
&\qquad = b\ +\ b1(Home\ Team\ xG\ For(15\ Game))\ +\ b2(Away\ Team\ xG\ against\ (15\ game))
\end{aligned}
$$

The estimates for the 15-game mean for the home team were as follows:

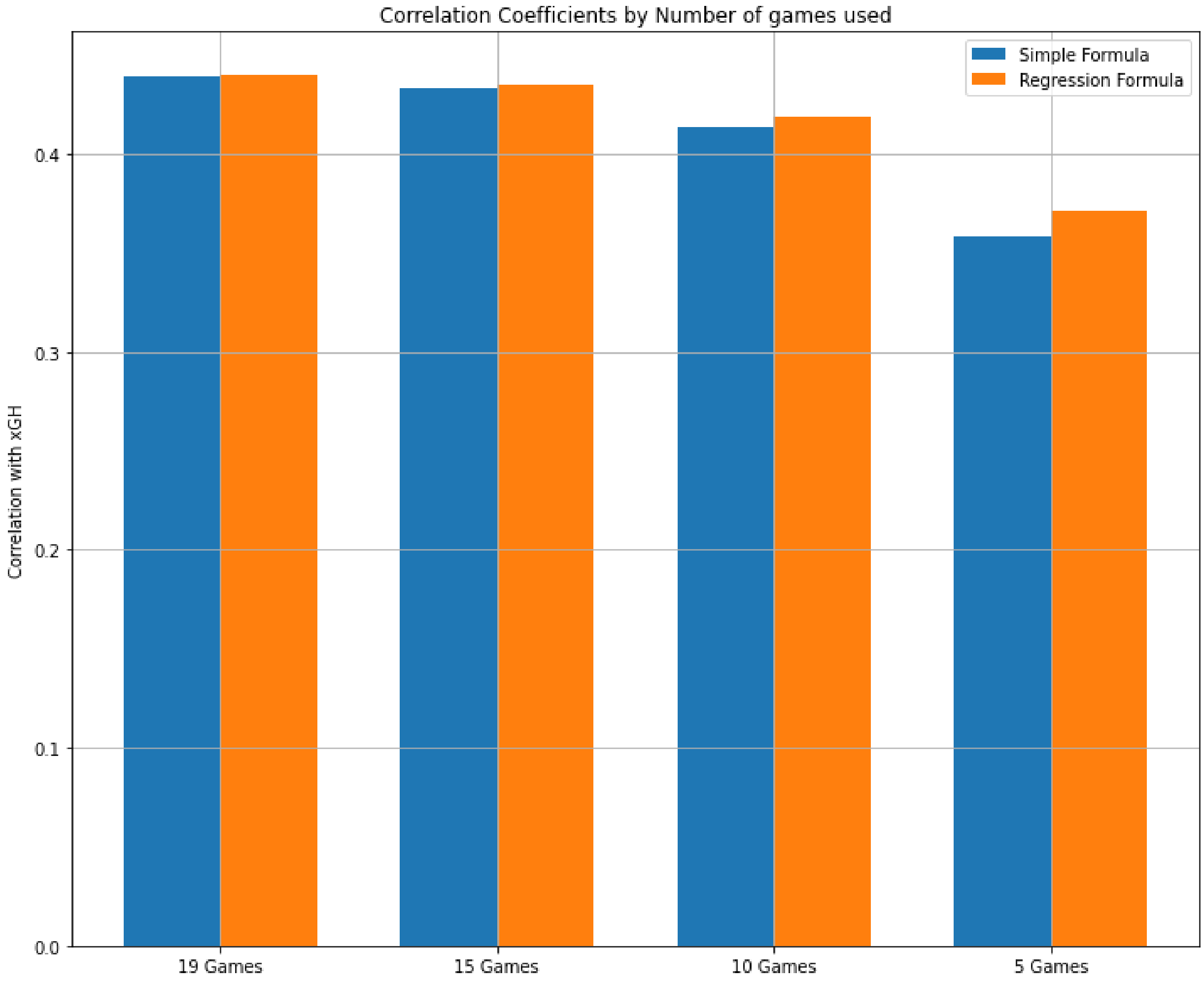
Predicted xGH (for 15 Game mean): -0.41 + 1.07*(home_scored_15_mean) + 0.82*(away_conceded_15_mean)

I reproduced these estimates for all 4 of the means, home and away. This allowed me to create a new dataframe column representing the predicted goals as per the regression parameters, example for the 15-game mean below:

```
df['home_goal_expectancy_15_reg'] = (-0.41) + (1.07*(df['home_scored_15_mean'])) + (0.82*(df['away_conceded_15_mean']))
```

For each of the means, I assessed the correlation between the 'Predicted Goals' by the regression-based model and the xG:

Figure 8: Correlation coefficient with xG for different metrics



As can be seen on the graph, the correlation coefficients are roughly equivalent for the ‘simple’ formula and the regression-based formula for both the 19 and 15 game means. However, the regression formula out-performs the simple formula on the 10 and 5 game mean metrics.

Predicting outcomes using the KNN and Logistic Regression models

The KNN model (K nearest Neighbors) and Logistic Regression machine learning models are classification algorithms, used to make predictions when the target data is discrete (categorical).

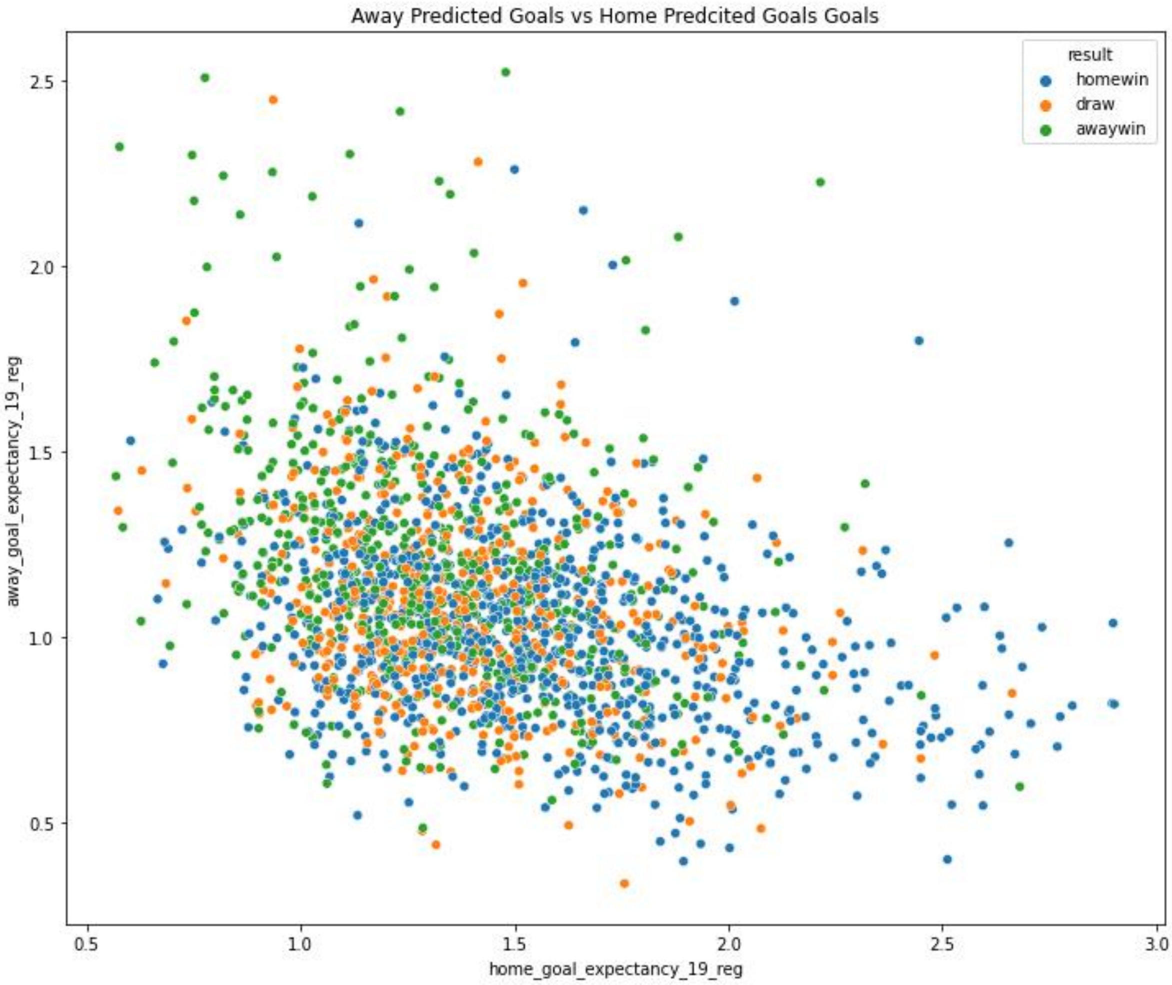
The KNN algorithm decides how each data point should be categorized by looking at the neighboring data points (meaning data points which have similar values for the feature variables). If most of the neighbors belong to a certain category, it is likely that the unknown data point will be in the same category as well. For each new data point, the model will assess the closest *K* number of neighbors and assign it to a category based on which category occurs most frequently.

The KNN model can be used on this dataset to predict whether a game will be a home win, away win or draw, using our 'Predicted Goals' values as target data to make the prediction. In this dataset, we use the following feature and target data. For the predicted goals, I used regression-based 19-game mean, as this held the highest correlation coefficient:

Target Data: Result (i.e. Home win, Away win or Draw)
Feature Data: Home Predicted Goals, Away Predicted Goals.

To visualize and see if there was any separation, I first plotted the variables on the graph:

Figure 9: Away Predicted Goals vs Home Predicted Goals (for sample of 1000 games)

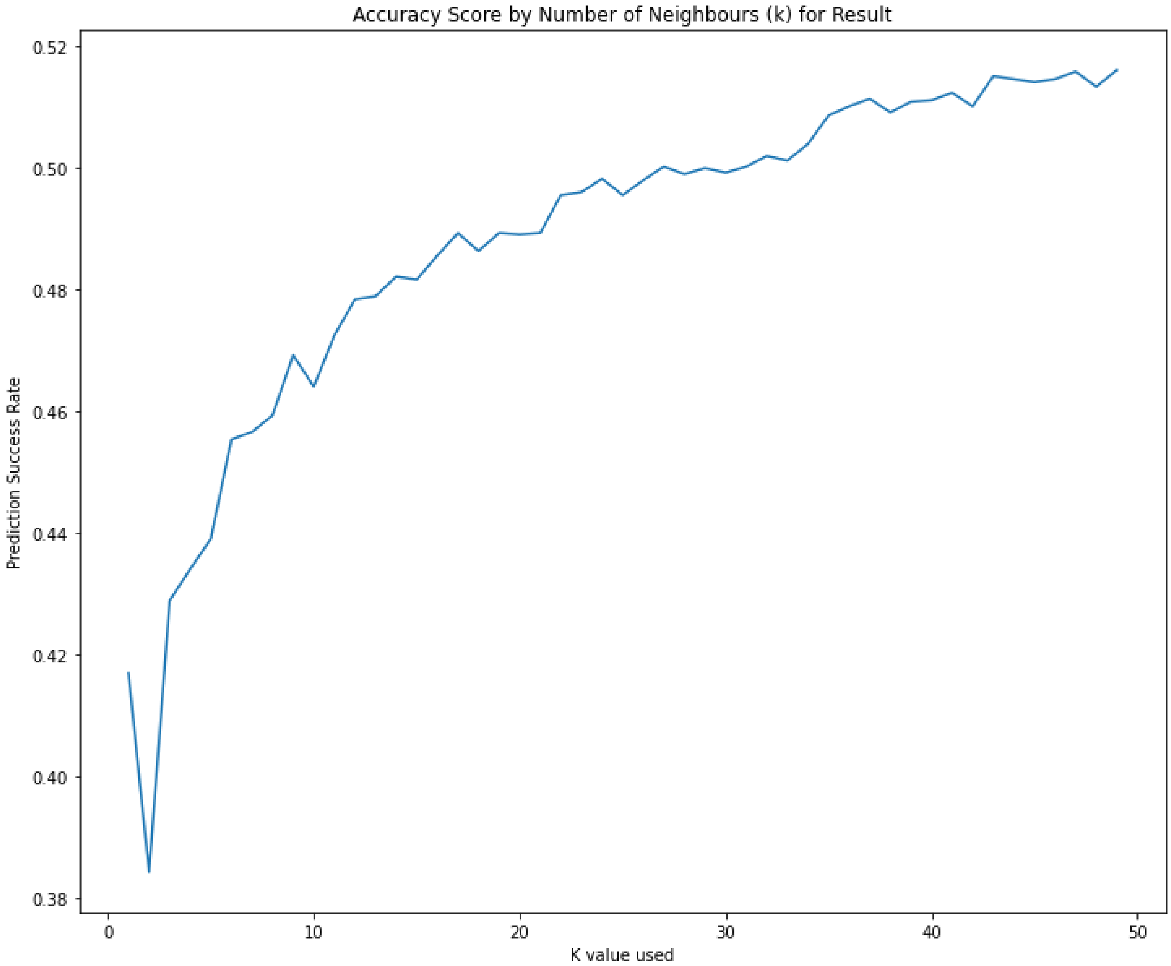


We would expect this figure to show separation based on the colour (the result). This means we would expect to see games in which the home ‘Predicted Goals’ was higher and the away ‘Predicted Goals’ was lower (the bottom right hand of the graph) to result in a home win (coloured blue). We would also expect games in which the home ‘Predicted Goals’ was lower, and the away predicted goals was higher (the top left of the graph) to end in an away win. If the ‘Predicted goals’ were roughly equal (the center of the chart), we might expect the teams to draw. As we can see, there is some separation, particularly at the extremes, but there is also a good deal of overlap.

To test the predictions made by the model, I used the train-test split method, which divides the data up into two groups, the training data and the test data. The model learns from the training data to understand the relationships between the feature and target variables. It then applies this relationship to the test data, allowing us to assess the accuracy of the predictions.

When assessing the outcome, the result is underwhelming. Even when a high K value is used, the success rate of the predictions is only around 50%. The graph plotting the success rate against the K number used can be seen below:

Figure 10: Accuracy score vs (K) values for result-based model.

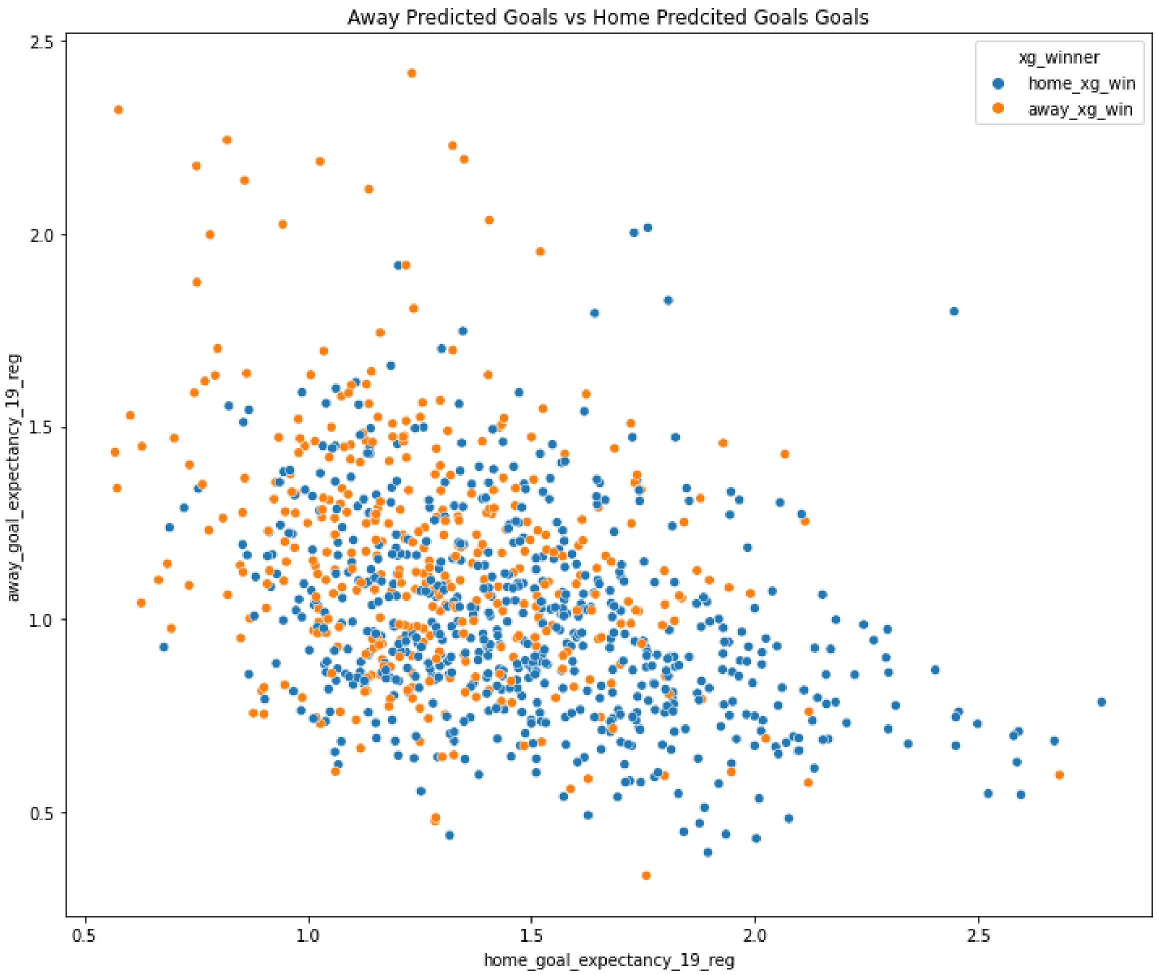


To try and improve the success rate of predictions, I decided to use the Logistic Regression model. This model aims to isolate the impact of each feature variable on the target variable, computing the difference that changes in the feature variables have on the probability that the target variable will be a particular category. Once again, I fit the model then used a train-test split to evaluate the results. This model correctly predicted the result around 52% of the time, very similar to the KNN model.

This is unconvincing, as someone who simply chose an outcome at random would expect to be correct around 33% of the time. This can perhaps be explained by the high variance which occurs within a football match. As the number of goals scored within a football match is very low, the result does not always reflect what has happened in the game.

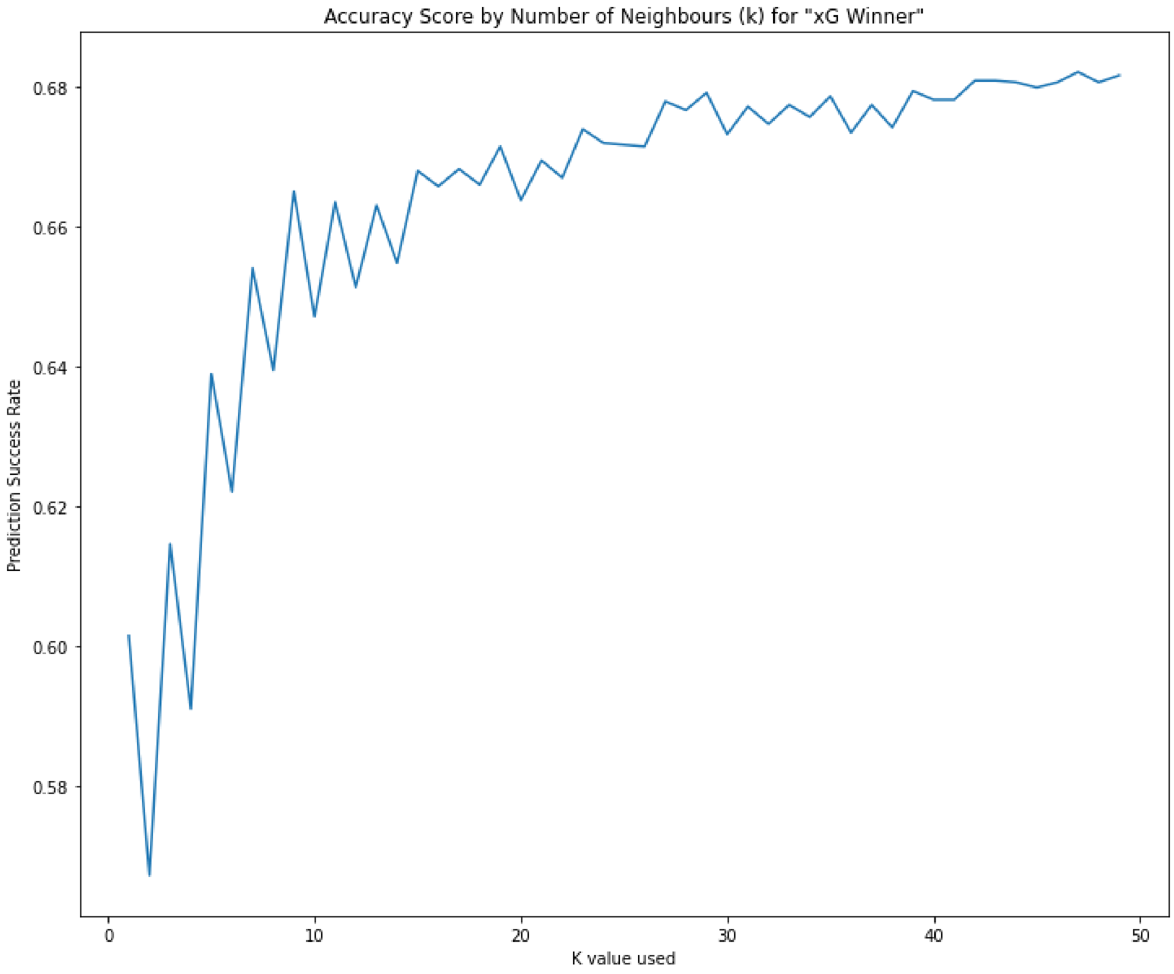
To try and account for this ‘noise’, I decided to create a different version of the model using different target data. To do this, I created a column to represent which team had the higher expected goals in the match, ignoring the actual result. In theory, this would account for some of the fluctuations discussed above, and therefore provide a more stable target variable to predict. Once again, I visualized this on a graph to see the separation:

Figure 11: Away predicted goals vs Home predicted goals for ‘xG Winner’ version of model



As seen on the graph, this target variable produces more separation between the points based on colour. Use of this target variable did indeed improve the performance of the model, raising the success rate to around 68%:

Figure 12: Accuracy score vs (K) values for 'xG Winner' based model.



Applying the Logistic Regression model to this new model specification produces a success rate of 68%, again very similar to the KNN model. Unfortunately, despite the higher success rate, this revised model is likely to be of little use to a gambler, who can only wager on the result itself.

Furthermore, the KNN would always be of limited use, even if the prediction rate was higher. In a game of high variance, a model which predicts the outcome in absolute terms is unlikely to be successful in the long run. As we highlighted, rather than predicting which team is the most likely to win, a successful betting model will find the teams whose chances of winning are not reflected by its odds. Since the KNN model's predictions make no reference to the odds at all, but rather selects just one result, it is unlikely that following the predictions of the model will return a profit in the long run, particularly since the teams it chooses are likely to be favorites with the bookmakers anyway. The KNN model is a much better fit where the categorical data you are looking to predict is a fixed category, such as with the famous ‘Iris’ dataset.

Assessing the Poisson Distribution

Before I used the Poisson distribution to determine the outcome of the game, I first needed to assess whether it accurately represents the data. To test this, I used the Poisson distribution to assess the probability of each outcome when the ‘Predicted Goals’ was set to 1.48 and 1.13, the mean goals scored for both the home and away teams in the dataset. If the Poisson distribution is an accurate representation of the data, then we would expect the probability of each goal outcome to be equal to the percentage that this outcome occurs within the dataset. For example, we know that the home team scores exactly one goal around 32% of the time across the entire dataset. We would therefore expect the Poisson distribution to produce a probability of 0.32 when the home team’s ‘predicted goals’ is set to the average value. When plotting the predicted odds against the observed outcomes, the Poisson distribution looks a good match for the data:

Figure 13: Average Poisson probabilities plotted against observed frequencies for home goals

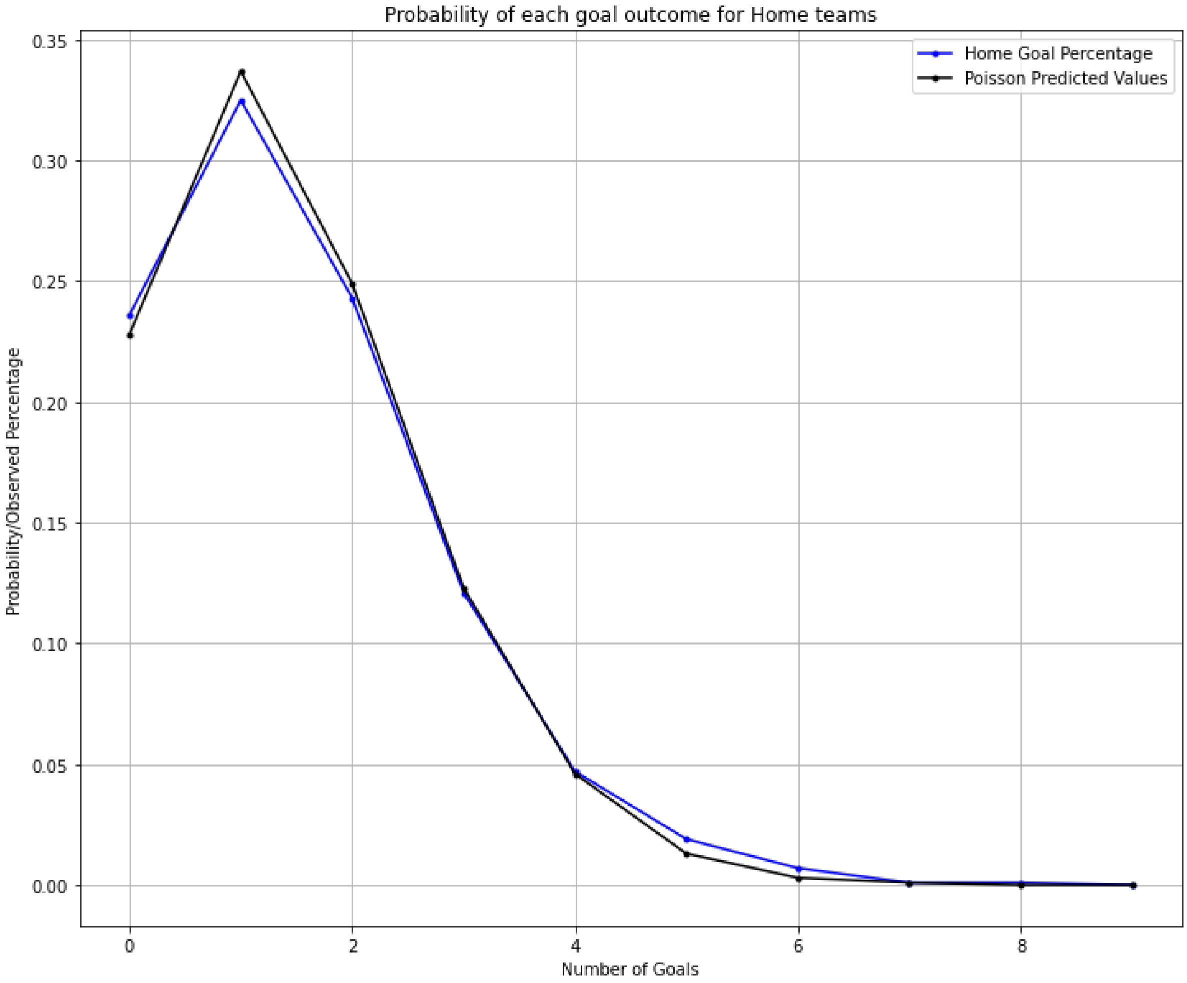
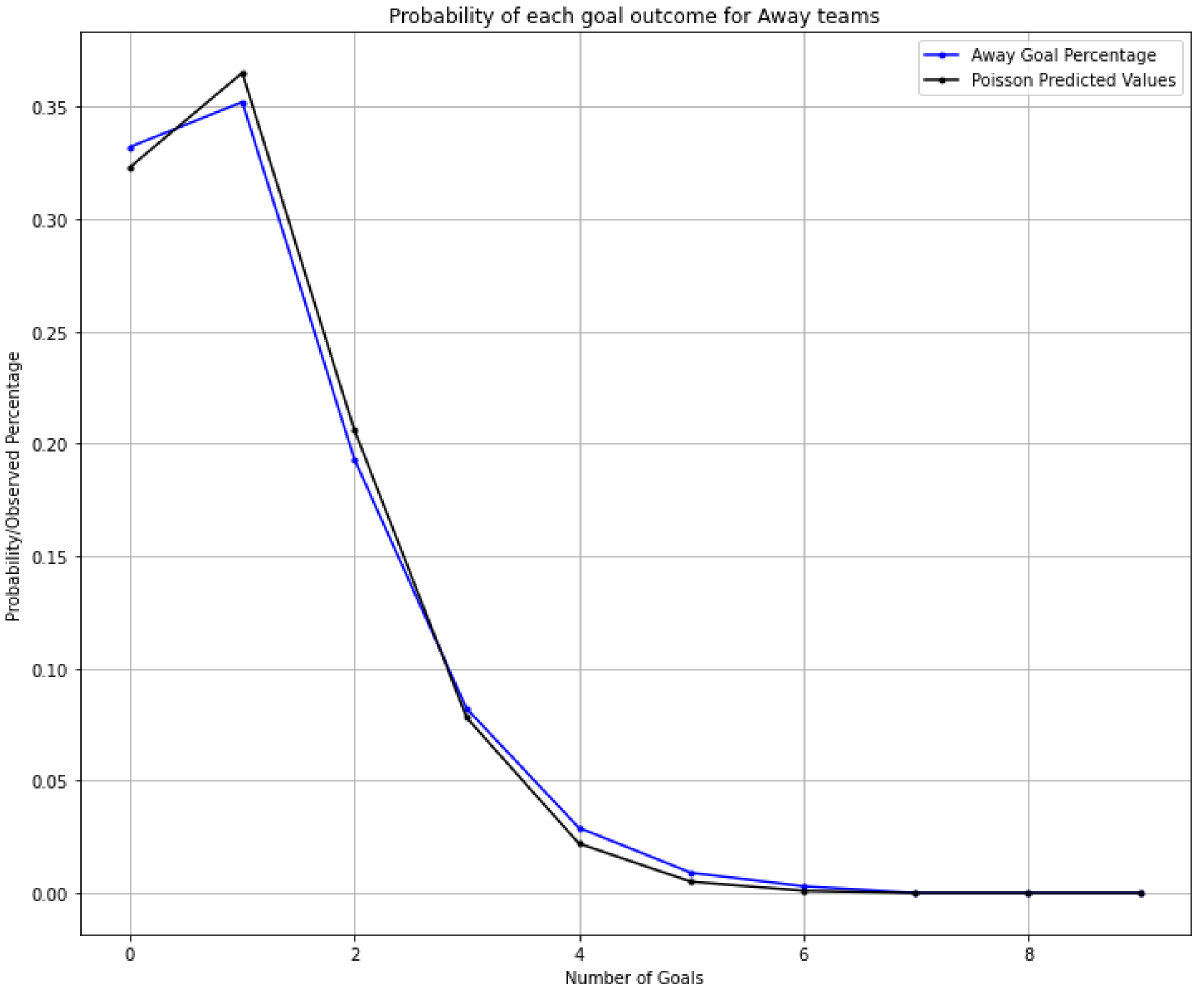


Figure 14: Average Poisson probabilities plotted against observed frequencies for away goals



As we can see, the data is a close match in all cases, an encouraging result.

Testing the model's predictions using the Poisson distribution

I decided to set the odds for each game using all 8 of the metrics, so that I could track the performance of each of them. I inputted each goal expectation into the Poisson formula, to understand the odds associated with each outcome:

```
df['home0_15'] = Poisson.pmf(0, df['home_goal_expectancy_15'])
```

The code above creates a new column which tells us the probability of the home team scoring 0 goals, given the predicted goals produced by the 15 game average. I repeated this process to understand what the probability was of each number of goals scored, by each team. I then matrixed the two distributions to work out the probability of each result. This allowed me to attach the odds produced by all 8 of the metrics to each game. Assessing the correlation of odds produced by each version of the model vs the odds offered by the bookmakers yielded the following results:

Table 6: Correlation coefficients between the odds produced by each version of the model and the bookmaker's odds

Metric	Correlation Coefficient vs Bookmaker odds	Statistically significant at 5%?
Simple 19 Game Mean	0.82	Yes
Simple 15 Game Mean	0.79	Yes
Simple 10 Game Mean	0.72	Yes
Simple 5 Game Mean	0.39	Yes
Regression-based 19 Game Mean	0.81	Yes
Regression-based 15 Game Mean	0.79	Yes
Regression-based 10 Game Mean	0.73	Yes
Regression-based 5 Game Mean	0.60	Yes

Interestingly, the pattern of the correlation with the bookmaker odds seems to mirror that of the xG itself, observed in figure 8. Once again, we see that the correlation is strongest with the 19-game metrics, although the increase in correlation achieved by increasing sample size is once again subject to diminishing returns.

I then created a function to select the bet which has the highest implied value, according to the odds produced by each version of the model. For each outcome (home win, away win or draw), I divided the bookmaker's odds by those produced by the model. If the resulting value was positive, this implied that the bookmaker's odds were higher than the estimated ‘true’ odds of that outcome, and the bet therefore had a positive expectation. If there were two outcomes which yielded a positive value, the one which was highest was selected. In each case, a threshold was built in, where odds offered by the bookmakers had to be higher than 1.05x the odds offered by the model, to account for the bookmaker’s margin (the amount by which the odds are reduced to guarantee the bookmaker a profit). Finally, if there were no selections for that threshold, ‘no bet’ was returned by the function. For each row in the database, the function selected a value bet for each version of the model. The frequency of each bet is shown below:

Table 7: Frequency of each bet type recommended by each model

Metric	‘No Bet’ Count	Home Bet Count	Away Bet Count	Draw Bet Count
Simple 19 Game Mean	5613	3474	2301	902
Simple 15 Game Mean	5519	3422	2406	943
Simple 10 Game Mean	5124	3393	2657	1098
Simple 5 Game Mean	3697	3658	3316	1619
Regression - Based 19 Game Mean	5016	3970	3040	264
Regression - Based 15 Game Mean	4983	3953	3089	265
Regression - Based 10 Game Mean	4697	4059	3346	188
Regression - Based 5 Game Mean	2716	3026	958	5590

The frequency of the bets does seem to cast aspersions on the regression-based 5 game mean, which seems to over-predict a draw significantly. Once again, this might be because 5-games window is a small sample size.

Once I had the bets which the model selected, I compared these bets with the actual outcome which occurred for the match, creating a new column to track the results. If the recommended bet proved to lose, the function returned -1, to represent the loss of a stake when betting. If the recommended bet was a winner, the column was populated with the value the bet would have returned, subtracting 1 to account for the stake. Once again, I did this for all eight versions of the model.

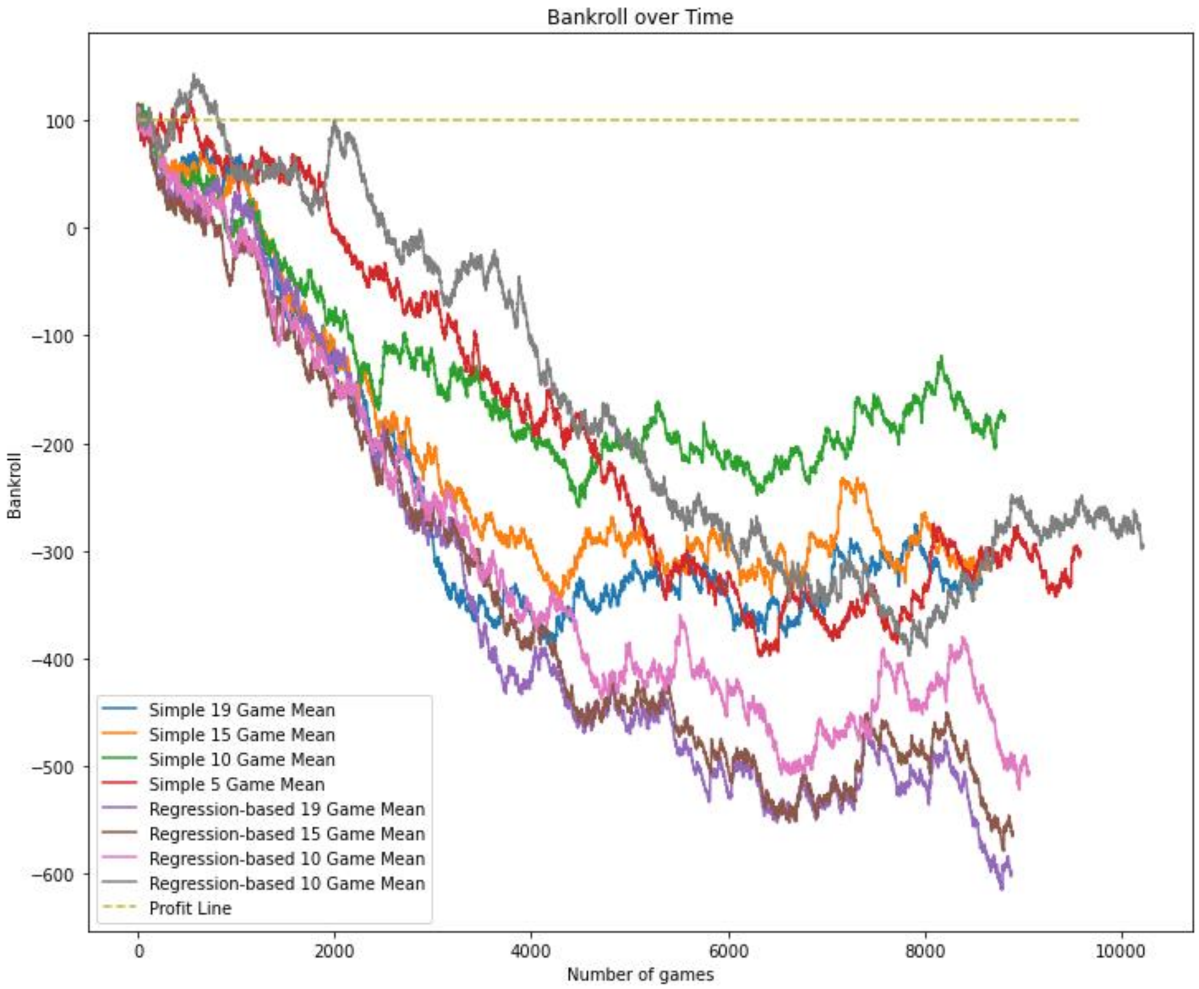
Once this was done, I was able to compute an average expected return for each of the model types. By dropping the rows which have a ‘nobet’ and adding all the remaining values in the ‘results’ column to a list, I was able to determine the mean value. This represents the expected return for each version of the model, or the amount of money we would expect to receive back when placing a £1 bet. Calculating these yielded the following:

Table 8: Expected values for each version of the model

Model used	Expected Return (% of stake)
Simple 19 Game Mean	95.08 %
Simple 15 Game Mean	95.21 %
Simple 10 Game Mean	96.89 %
Simple 5 Game Mean	95.81 %
Regression-based 19 Game Mean	92.12%
Regression-based 15 Game Mean	92.54%
Regression-based 10 Game Mean	93.30%
Regression-based 5 Game Mean	96.12%

As can be seen above, all versions of the model have a net negative expected value, meaning they do not have a competitive edge vs the bookmakers. However, they are not far from the profit mark of 100. Given the average bookmakers margin is around 5 - 8 %, the performance is reasonably strong. Interestingly, it is the simple 10 game mean which is the most profitable, despite not being the metric with the highest correlation with bookmaker odds. However, correlation with the bookmaker's odds does not necessarily imply that the model will produce value. Indeed, it is only through deviation from the bookmaker’s odds which can produce value at all - if all of the odds produced by the model were identical to the bookmakers odds, there would be no bets recommended. It may be that the 10-game period is a short enough window to account for short term changes to a team’s form, finding value despite not being so strongly correlated. To visualize what would have happened if the recommendations had been followed, I plotted the bankroll achieved over time for each version.

Figure 15: Simulated bankroll when following the recommendations over time



As we can see from the above, all the models lose money over time, however, it is the ‘simple’ 10 game mean which loses the least.

To try and improve the performance of the model, I set the value threshold for making a bet to 2, rather than 1.05. This means that the criteria for making the bet was much higher. The intention was to filter out the marginal bets, leaving only the bets which the model believed were of premium value. Unsurprisingly, his had the effect of significantly reducing the number of bets the model recommended:

Table 9: Frequency of each bet type for the adjusted version of the model

Metric	‘No Bet’ Count	Home Bet Count	Away Bet Count	Draw Bet Count
Simple 19 Game Mean (Adjusted)	12013	147	115	15
Simple 15 Game Mean (Adjusted)	11972	166	134	18
Simple 10 Game Mean (Adjusted)	11856	199	217	18
Simple 5 Game Mean (Adjusted)	11445	325	471	49
Regression-based 19 Game Mean (Adjusted)	11448	453	373	16
Regression-based 15 Game Mean (Adjusted)	11358	493	423	16
Regression-based 10 Game Mean (Adjusted)	11134	614	533	9
Regression-based 5 Game Mean (Adjusted)	11288	579	225	198

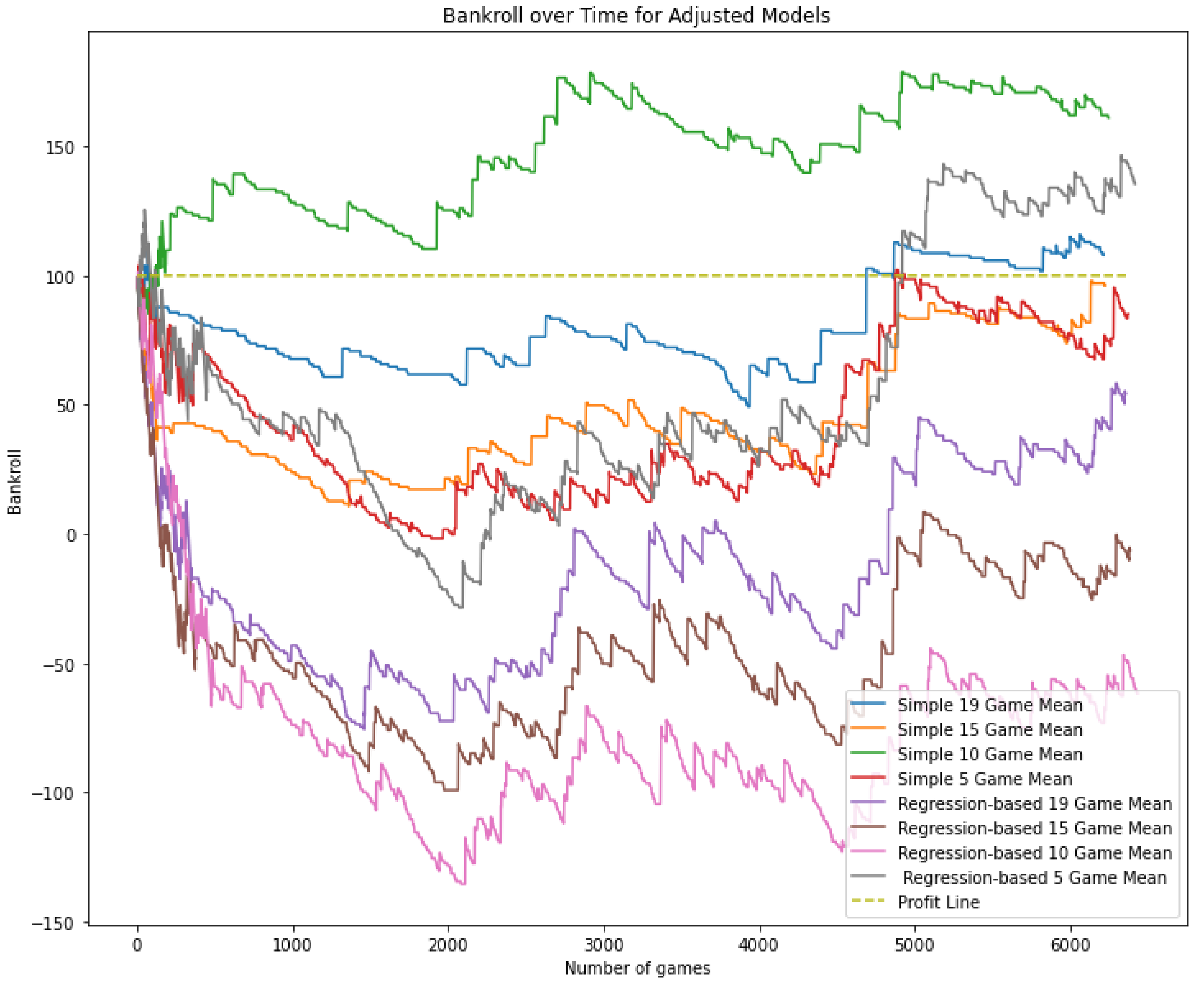
Despite the lowered value count, the expected values of the model were significantly improved:

Table 10: Expected value for the adjusted version of the models

Metric	Expected Return (% of stake)
Simple 19 Game Mean (Adjusted)	100.13 %
Simple 15 Game Mean (Adjusted)	99.97 %
Simple 10 Game Mean (Adjusted)	101.3 %
Simple 5 Game Mean (Adjusted)	99.75 %
Regression-based 19 Game Mean (Adjusted)	99.28
Regression-based 15 Game Mean (Adjusted)	98.34
Regression-based 10 Game Mean (Adjusted)	97.49
Regression-based 5 Game Mean (Adjusted)	100.55

As you can see, three of the models have positive expected values, meaning that a gambler who followed the recommendations could expect to make a modest profit over time. Once again, it is the simple 10 game mean which provides the highest expected value, implying that this might be the number of games which optimizes the trade-off between a shorter and longer sample size. Once again, I plotted the results for all the revised models over time, including a profit line:

Figure 16: Simulated bankroll when following the recommendations of the adjusted models over time



As we can see from the graph, a gambler who followed the predictions of the adjusted simple 10-game mean, adjusted simple 19-game mean or adjusted regression-based 5-game mean would have made a profit by following the predictions. However, it is the adjusted simple 10-game mean which comes out on top.

When examining the results of the adjusted simple 10 game mean further, it becomes clear that the model seems to find value by selecting ‘underdog’ teams, as the average odds which the model recommends are 7.25, a significant underdog in a football match.

Looking at the bets recommended, it seems as if the model often recommends betting against high-reputation teams. The table below shows the teams that model successfully recommended betting against more than once:

Table 11: Teams who the 10- game model successfully recommended betting against more than once, with frequencies.

Team	**Number of times the model successfully recommended betting against them:**
Barcelona	5
Real Madrid	5
Chelsea	5
Manchester United	3
Paris Saint Germain	3
Inter	2
Atletico Madrid	2
Wolfsburg	2
Bayer Leverkusen	2
Napoli	2
Arsenal	2
Manchester City	2
Tottenham	2
Monaco	2

All the teams above are historically successful clubs with large reputations, easily recognizable to casual football fans. On this evidence, it seems that the model has a competitive edge where the betting public are biassing the team with which they are familiar, leading to a profitable betting opportunity from betting against them.

Weaknesses of the model and areas of improvement

Although having a model which can produce a profit over the long run is an exciting outcome, there are some areas in which the model could be improved. These broadly fall into the two categories, the first being the way the model is constructed, the second being missing factors from the model.

Although the Poisson distribution seems to be a good representation of the distribution of football results, there are reasons to think that it might not be optimum. A key assumption of the Poisson distribution is that the events are independent, and that the probability of an event occurring does not change based on how often it has happened previously. However, this is unlikely to be the case in a football match, as teams adjust their strategies based on the goals already scored. For example, a team who has gone behind is likely to try a more attacking style in order to try and recover the deficit, meaning that goal events are not completely independent of each other. For this reason, an adjusted Poisson distribution which accounts for the interdependence of goals may improve the profitability of the model.

In addition, this version of the model treats the home and away teams as separate entities, meaning a team’s performance at home will not affect their away ratings, and vice versa. Although many teams have contrasting home and away records, it is extremely unlikely that they are completely independent of one another. Although it would be possible to build a model in which the away performance affects the home performance, the impact of home performances on away performances is likely to be heterogeneous between clubs, making it hard to predict its impact. Another alternative would be for the model to simply ignore the distinction between home and away, but this would make it difficult to index the games in a way which allowed for tracking over time.

One further weakness of the model are the unconsidered factors not included in its assessments. Although xG is a repeatable metric in general, there are exogenous factors which could mean that a team's recent performances are unlikely to be repeated in the next game. If these factors are not considered, then the odds given by the model may be unrepresentative of their true chances. For example, the model is unable to account for an injured player. Losing a high-profile player can significantly impact a team's attacking or defensive ability, but, since the model is unaware of the injury, it will make predictions as if the injury had never occurred. This could lead to an overestimate of the team's chances of winning. Having more detailed data, which accounts for the contributions of individual players may help this, but it is not publicly available. Another factor which is unconsidered in the model is the tactical matchups between teams. Each team has its own tactical style, with its own strengths and weaknesses. If a team employs a tactic which the opponent struggles to adapt to, there are reasons to think that the opposing team may not be able to repeat their recent attacking or defensive performances during the match. The final unconsidered factor is the recovery time between matches. If a team has a shorter or longer recovery time than usual, it is likely that performance will be affected. Assessing the impact of recovery on performance could be an area for further research, but is not possible with this dataset, which only includes domestic league games.

When analyzing the results, it is also notable that all the models with the higher value threshold have a similar expected return, all around 99% of stake. It is therefore possible that the positive expected returns produced by three of the models are just down to natural variance, and they would not produce the same results over a larger sample size. This theory is backed up by the fact that the models have remarkably similar trajectories when plotted on the graph. Unfortunately, there is no way to test this using the existing data, indeed, since the true odds are never known to us, it would take an enormous sample size to rule out luck completely. In the end, this comes down to a value judgement, however a profit over a 10,000-point dataset would be good enough for most gamblers.

There are also reasons to think that the model could produce even greater returns when applied to out of sample data. The bookmaker's odds used in this project are from Bet 365 at the time of kickoff, as these were the most complete. If this model was used in the real world, the gambler would have the luxury of ‘shopping around’ for odds on other websites, as well as the ability to bet at other times. This may allow them to find higher odds, increasing the expected value of the bets.

Conclusion

This project explored the use of data analysis techniques to predict the outcome of a football match. The data used was expected goals data, in combination with data on the odds offered historically by bookmakers. This report finds that expected goals is a repeatable and detailed descriptive metric, with predictive power when used to forecast results. This report describes a model built using expected goals to predict match outcomes in absolute terms, with limited success. The model is adapted to produce the ‘true’ odds of an outcome occurring, illuminating profitable betting opportunities. Simulating the results across the dataset, some of the models do produce a positive expected return, with reasons to think that these positive returns could continue on out of sample data.

References

Alan J. Lee (1997) Modeling Scores in the Premier League: Is Manchester United Really the Best?, CHANCE, 10:1, 15-19, DOI: 10.1080/09332480.1997.10554791

Caley, Michael (19 October 2015). "Premier League Projections and New Expected Goals". https://cartilagefreecaptain.sbnation.com/2015/10/19/9295905/premier-league-projections-and-new-expected-goals. (Accessed 10 June 2021).

D Dyte & S R Clarke (2000) A ratings based Poisson model for World Cup soccer simulation, Journal of the Operational Research Society, 51:8, 993-998, DOI: 10.1057/ palgrave.jors.2600997

Dixon, Mark J., and Michael E. Robinson. “A Birth Process Model for Association Football Matches.” Journal of the Royal Statistical Society. Series D (The Statistician), vol. 47, no. 3, 1998, pp. 523–538. JSTOR, www.jstor.org/stable/2988632. Accessed 10 Aug. 2021.

Dixon, Mark J., and Stuart G. Coles. “Modelling Association Football Scores and Inefficiencies in the Football Betting Market.” <i>Journal of the Royal Statistical Society. Series C (Applied Statistics)</i>, vol. 46, no. 2, 1997, pp. 265–280. <i>JSTOR</i>, www.jstor.org/stable/2986290. Accessed 10 Aug. 2021.

Football-Data.co.uk Dataset (for Odds data): https://www.football-data.co.uk/englandm.php

G. K. Skinner & G. H. Freeman (2009) Soccer matches as experiments: how often does the ‘best’ team win?, Journal of Applied Statistics, 36:10, 1087-1095, DOI: 10.1080/02664760802715922

Goddard, J. and Asimakopoulos, I. (2004), Forecasting football results and the efficiency of fixed-odds betting. J. Forecast., 23: 51-66. https://doi.org/10.1002/for.877

Kaggle.com Dataset (for ExG data): https://www.kaggle.com/slehkyi/extended-football-stats-for-european-leagues-xg

Karlis, Dimitris, and Ioannis Ntzoufras. “Analysis of Sports Data by Using Bivariate Poisson Models.” Journal of the Royal Statistical Society. Series D (The Statistician), vol. 52, no. 3, 2003, pp. 381–393. JSTOR, www.jstor.org/stable/4128211. Accessed 10 June 2021.

Maher MJ (1982). Modelling association football scores. Statistica Neerlandica 36: 109-118

Petty, L. (2017) “What is expected goals? Expected goals explained” Available at: https://www.pinnacle.com/en/betting-articles/Soccer/expected-goals-explained/B8Q2HGJ7XMJRZ58C (Accessed: 10/06/2021).

Petty, L. (2018) “What is a value bet?”. Available at: https://www.pinnacle.com/en/betting-articles/educational/what-is-a-value-bet/MDFJ8GATUVW3M3MW (Accessed: 10/06/2021).

Richard Pollard (1986) Home advantage in soccer: A retrospective analysis, Journal of Sports Sciences, 4:3, 237-248, DOI: 10.1080/02640418608732122

Ryan H. Boyko, Adam R. Boyko & Mark G. Boyko (2007) Referee bias contributes to home advantage in English Premiership football, Journal of Sports Sciences, 25:11, 1185-1194, DOI: 10.1080/02640410601038576

Scarf, Phil. (2006). Modelling the outcomes of association football matches. 48th Annual Conference of the Operational Research Society 2006, OR48. 59-72.

Stanton, J. (2017) “Premier League: 'Expected goals' tells us whether a player really should have scored.” Available at: https://www.bbc.co.uk/sport/football/40699431 (Accessed: 10/06/2021).

Vlastakis, N., Dotsis, G. and Markellos, R.N. (2009), How efficient is the European football betting market? Evidence from arbitrage and trading strategies. J. Forecast., 28: 426-444. https://doi.org/10.1002/for.1085