# Football Forecasting - Can the Bookies be Beaten?

## Abstract:

This paper concerns the prediction of football matches, primarily in the Premier League, via predicting the final score lines. This is then expanded into other leagues. Three main methods are utilised- a generalised linear model using Elo ratings, a Poisson probability model for goals, and an advanced neural network. All three are explored in depth, with the first of the three developing a broader and more general view of predictions and the latter two exploring a more refined and tailored approach for predicting specific fixtures with a higher degree of accuracy. The models are evaluated and found to be highly useful in comparison with bookmaker's probabilities. As such, astute use of these models may allow for money to be made via betting- the main aim of having a good model.

## Introduction:

Betting companies, such as Bet365, make billions of pounds a year through offering odds that will entice customers to place bets, but put the bookies in pole position to win. Central to this is a forecasting model; if one can accurately predict future football scores (via a probability of a 1-0 home win, say) then they may offer odds which are slightly above this in order to tip the balance into their favour. This paper looks at building such a model, from a rather simple initial template, before building up to a more complex model. The data available are the statistics from all Premier League games from the beginning of the 93/94 season until 29/01/2020. Early season data just contains statistics of final result, and later seasons expand out greatly with more in-game statistics, such as corners, fouls and cards, as well as betting odds for the game in question which one can use as a reference. Performance is tested via application to real life games, as well as using PLL as a measure of success. A versatile model ought not to be league-specific, and so the model is also tested on other leagues for its accuracy.

## Home Advantage:

A key concept in football and many other sports, is the home advantage. There is a consensus that teams will perform better on a pitch that they constantly play on with the encouragement of the majority of the stadium (barring exceptional circumstances such as the game being played behind closed doors). The theory behind exactly why home teams perform better is not necessarily our concern, the crucial point is that the home advantage will give an underlying boost to the home team (whether that makes a good team perform incredibly or an awful team perform less poorly). This will, in turn, influence all of our statistical models.

Firstly, consider the most general case and aim to answer the question: “do home teams really perform better than away teams?” in a statistical framework. Consider the null hypothesis:

$$H_0 : Home\ teams\ win\ as\ much\ as\ away\ teams$$

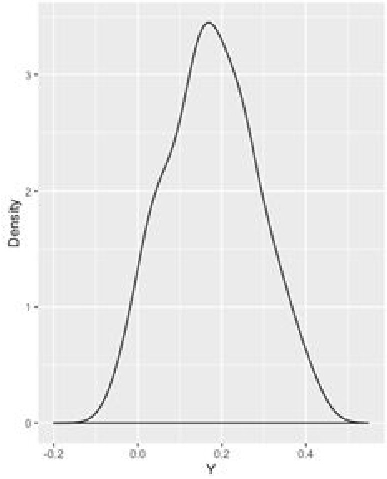
$$if\ \overline{H} = proportion\ of\ games\ resulting\ in\ home\ wins$$

$$and\ \overline{A} = proportion\ of\ games\ with\ away\ wins\ :$$

$$H_0 : \overline{H} = \overline{A}$$

$$H_0 : \overline{Y} = \overline{H} - \overline{A} = 0$$

Clearly each line is equivalent here. In order to test for the significance of $\overline{Y}$ , it is first required that this variable is normally distributed in some sense. This condition is satisfied via sampling 92 groups of 52 games, and taking an observation  from each group. These  tend to a normal distribution as the number of groups tends to infinity, by the central limit theorem. Whilst the variables being summed are not i.i.d, the Y’s will tend to a normal distribution as the number of groups tend to infinity, by a generalised version of the central limit theorem.



It is observed that the distribution is centered around a mean of $\overline{Y_{obs}}$ =0.1767206. One can thus carry out a test statistic with T given as below. The t distribution is used due to estimation of variance of Y, but a large number of groups ensures a small departure from normality anyway.

$$T = \overline{(Y_{obs}} - \overline{Y})/s\sqrt{92} \sim t_{91}$$

One calculate the sample standard deviation to be 0.1070097, and , under $H_0$ , $\overline{Y} = 0$ . Our statistic is T=15.8401 and $H_0$ is rejected comfortably at even the 1% level.

Although it may seem obvious to some that there exists a home advantage in football, quantifying the significance of this advantage through a test statistic is crucial in setting the foundations of this report. The exact meaning of this finding is as follows: consider a game between Team A (home) and Team B (away). A priori, the probability of a home win is undeniably higher than the probability of an away win, at least until one finds the true identities of Team A and Team B (and/or some useful information about them).

A home advantage has now been  established. In fact, from the years 2006-2019, 46.34% of games resulted in a home win, 25% a draw and 28.66% an away win, which is yet more support to our claim. This forms a basis for the models that follow.

## **Model Evaluation**

It is vital to ensure that our model predicts well, and there are some ways to test this (primarily by testing our models against the bookmaker's). One such measure is the mean square error of our probabilities to the bookmaker's implied probabilities. If this is small then this is an indication of success. However, one would also like a way to test our models without the assumption that the bookmaker's are entirely correct. As such, the PLL measure is introduced, which does not rely on the bookmakers' models, defined as:

$$PLL = \sum_{k=1}^{N} \delta_k^H log(p_k^H) + \delta_k^D log(p_k^D) + \delta_k^A log(p_k^A)$$

$$where\ \delta_k^H = 1\ if\ home\ team\ wins, \delta_k^H = 0\ otherwise$$

$$\delta_k^D = 1\ if\ teams\ draw, \delta_k^D = 0\ otherwise$$

$$\delta_k^A = 1\ if\ away\ team\ wins, \delta_k^A = 0\ otherwise$$

$$p_k^H = P(Home\ Win)$$

$$p_k^D = P(Draw)$$

$$p_k^A = P(Away\ Win)$$

This result is always non-positive since it is the log of an argument in [0,1]. As such, one wishes to maximise this via making it least negative as possible. A PLL near 0 indicates the model has predicted the event with high probability, when the outcome of that game is realised. This still allows for comparison to bookmaker's however, since one can merely compare PLL ratings. For data of varied lengths one can take mean PLL as a measure instead. Since our dataset concludes such betting odds, one can immediately realise the following performances of respective betting companies:

| Company | Bet365 | Betway | Stan James | Interwetten | Ladbrokes | Sportingbet |
|---|---|---|---|---|---|---|
| Mean PLL | -0.9554 | -0.9574 | -0.9611 | -0.9585 | -0.9630 | -0.9644 |

The results are very similar between the bookmakers-perhaps they use largely similar models to make their predictions! From this, one can see a model with mean PLL of -1 would be of similar quality to the bookmakers', and models that  approach values greater than -0.95 over large testing datasets will actually predict better than bookmakers. This allows us a "measure of success" versus the bookmakers.

**Methods:**

# 1.Generalised Linear Model

In addition to the aforementioned home advantage, one may wonder what other factors may influence the probability of each result. An obvious one would be the teams involved. But with this comes complications: Manchester City of late have been one of the most dominant teams in Premier League history but were mediocre at best before their financial takeover in 2008. So there could be an interaction between the team playing and the time period that the game takes place. In addition, there may be a situation where Team A and Team B have met 20 times previously, and Team A have won 19 and drew 1. In this case, there is possibly an interaction between the two teams. A more serious flaw is that teams that enter the league for the first time ever (recent examples are Brighton and Bournemouth) have such little data available that, initially, it is hard to gauge how they will perform. A useful way to avoid some of these issues is to use Elo rating.

## Elo Rating

Elo rating is a metric used to quantify the relative skill of players or teams in games, which leads to a power ranking of the competitors. Originally applied to games like chess, there is research into applying such theory to football. The inclusion of Elo ratings to this report are a result of a few useful qualities:

1. A model which depends only on Elo ratings and not historical data from specific teams' results should, in theory, be easily generalised to other leagues (one of the key themes of this report). All that is necessary is one season's data for training and continuous recordings from this point (regardless of season or league). For example, if one wants to use such model to predict the 2019/20 Bundesliga season, one requires full and continuous Bundesliga data from at least 2018/2019- gaps in data is not consistent with the discrete differential equation involved in the Elo rating method. One must also update Elo ratings throughout the predicted season so that every prediction has up-to-date comparisons.
2. A consideration mentioned at the beginning of this section was that certain teams (like Manchester City) have performed very differently across the years. Since Elo ratings alone do not take into account the team in question, only the difference in recent skill between two teams, the issue of historical data being used for current games is alleviated.

Now that some justification behind the use of Elo rating in this report has been given, here is how it is calculated. First of all, there are many adaptations of Elo theory but, in our case, a system implemented by Stuart Lacy (research fellow at the University of York) that always has a 1500 mean rating will be considered. This is a result of the system being zero-sum. That is, if a winning team gains a certain magnitude of rating, the losing team loses the same magnitude of rating. One is concerned with the difference in Elo rating between the home team and away team:

$$dr = Elo_H - Elo_A$$

A common assumption is that a team's performance is a random variable centred on its rating. The underlying distribution of the performance is often taken to be the logistic distribution. The reason behind this is that upsets in football are fairly prevalent and the logistic distribution has larger tails than the normal (for example). Though the Elo ratings of different teams may be centred about different means, due to the zero-sum nature of Elo ratings the average difference in Elo between teams is zero. Hence, one can assume that $dr$ is also logistic but centred about zero. One might therefore assume that a higher value of $dr$ would indicate a higher chance that the home team wins. In fact, this is the case. Using the logistic curve, an expected outcome is calculated by the CDF:

$$E = \frac{1}{1+10^{-\frac{dr}{400}}}$$

Clearly, $E = 1$ would indicate a home win (when $dr$ is large and positive), $E = 1/2$ would indicate a draw ( $dr$ close to zero) and $E = 0$ would indicate an away win ( $dr$ is large and negative). One thing to note is the parameter 400. This scaling constant has previously been found to work well on football Elo ratings and means that $dr \sim Logistic(0, 400)$.

Previously, the home advantage has been proven to exist. To this end, should one really expect two teams of equal skill to draw? One could certainly claim that in this event, the home team should be more likely to win. In fact, Lacy adapts the model above by including a home advantage, labelled HA:

$$dr = Elo_H - Elo_A + HA$$

HA is calculated by looking at historical data. Assume again that, on average, both teams are of equal rating. Then $dr = HA$ , so:

$$E = \frac{1}{1+10^{-\frac{HA}{400}}}$$

$$\Rightarrow HA =- 400log_{10}(\frac{1}{E} - 1)$$

Using all of our data, one finds that $E = 0.5884$. Then $HA \approx 62$ (this is a slight deviation from the value Lacy used). The update of Elo rating is governed by the equation:

$$Elo_H^{'} = Elo_H + KG(O - E)$$

Where K and G are variables to be discussed, the dash represents the updated Elo rating and $O$ is the observed outcome: $O = 1$ if the home team wins, $O = 1/2$ if teams draw and $O = 0$ if the away team wins. Consequently, if a team ‘punches above their weight’ and $O$ is much larger than E, their Elo rating will increase more than if they beat a team far inferior.

The simpler parameter, K, is arbitrarily chosen in most models as 20. The effect of this parameter is to scale the importance of each result. For tournaments with fewer games, K may be large but for European leagues, teams play many games so K can be relatively small.

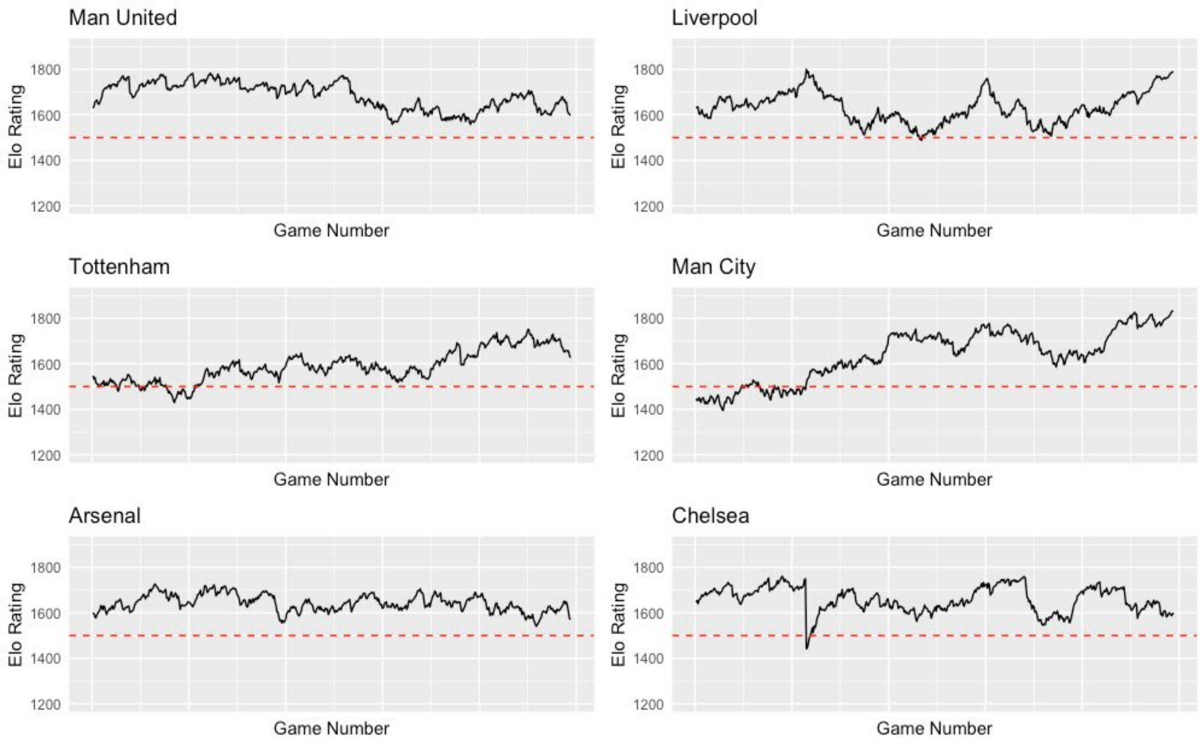
The more difficult parameter is G, which is dependent on the margin of victory (MOV). Sparing most of the details, Lacy’s formula for G contains a few key ideas. G will be affected more when the MOV goes from 1 to 2 than it will be from 7 to 8 (for example). That is, if G were continuous, $\frac{d^2G}{dMOV^2} < 0$; the increase of G becomes less as MOV increases. This decreases the importance of huge victories- winning by 8 goals is hardly more convincing than winning by 7. In addition, there will be a penalty added to G when $dr$ is large. When one team is far better than another, the effect of large MOV will be less than if the teams are similar in ability. The value of G will depend on MOV like so:

$$G = 1,\ if\, MOV \leq 1$$
$$G = log_2(1.7MOV)\frac{2}{2+0.001dr},\ if\, MOV > 1$$

There is one last issue with this model. Unlike other sports (like basketball), every year three teams with the lowest points will be relegated from the league and three teams from the Championship will be promoted (2 automatically and 1 through play-off games). So the teams in the league change annually and our model must find a way to deal with this. Since the teams relegated from the Premier League and the teams promoted from the Championship bridge the gap between the two leagues, the most informed assumption would be that they are of similar quality. Of course this assumption can be false in some cases, but it is the best simple assumption to make and their Elo rating will be corrected within a matter of games regardless. Justification for this is that there is only one occasion since 2006 where no promoted teams were relegated in the next season. Most of the time, at least one of the teams promoted face immediate relegation in the following season. To ensure that the mean of 1500 rating is consistent throughout, one can assign the average Elo of the three relegated teams to the three promoted teams at the start of the following season.

In addition, due to factors like end of season form and transfers, the gap in ratings will be pulled towards the mean using $Elo^{'} = 0.8Elo + 0.2 \times 1500$. Hence, the hierarchy of teams is maintained but the gap in quality is reduced to account for the uncertainty at the start of the season, and to indicate a “refresh”- a new season beginning. At the end of each season, the ratings will be pulled towards the mean and subsequently the promotion/relegation theory previously mentioned will be applied.

Having applied this measure to our Premier League data (after first training it on the 2005/06 season), this leaves a dataset containing results from 2006-2019, along with the Elo ratings of the home and away teams involved. To give this theory some colour, an example of Elo over time can be visualised by comparing the ‘Big 6’ over time:



The dotted red line represents the league average Elo. It is encouraging to see that Manchester City and Liverpool have the best Elo ratings in recent years which definitely reflects reality; Manchester City being the current champions, and Liverpool the champions elect.

## Building The Model

With three outcomes, a home win, a draw and an away win, one requires a multinomial output with away win acting as a reference level. The generalised linear model with home and away Elo as the only covariates gives equations of the form:

$$ln\left(\frac{P(Home\ Win)}{P(Away\ Win)}\right) = a_0 + a_1 Elo_H + a_2 Elo_A + \varepsilon_a$$
$$ln\left(\frac{P(Draw)}{P(Away\ Win)}\right) = b_0 + b_1 Elo_H + b_2 Elo_A + \varepsilon_b$$

Using data from the 2006-2019 seasons, there are 4940 games. Fortunately, our model doesn't rely on the time that data is recorded or the teams involved. Our model is only dependent on Elo ratings before a game instance occurs. One can therefore split training and testing data in any manner one likes. For ease, ones shall take the first 70% of data (3500 games) to train and the remaining 30% to test. This gives estimations to the parameters of our predictions as:

$$\frac{P(Home\ Win)}{P(Away\ Win)} = exp\,(0.3732401 + 0.00674583 Elo_H - 0.00659926 Elo_A)$$
$$\frac{P(Draw)}{P(Away\ Win)} = exp\,(1.1073225 + 0.00280756 Elo_H - 0.00349202 Elo_A)$$
$$Also,\ one\ imposes\ P(Home\ Win) + P(Draw) + P(Away\ Win) = 1$$

Thus, there are three equations with three unknowns, which is solvable provided one knows the Elo ratings of each team.

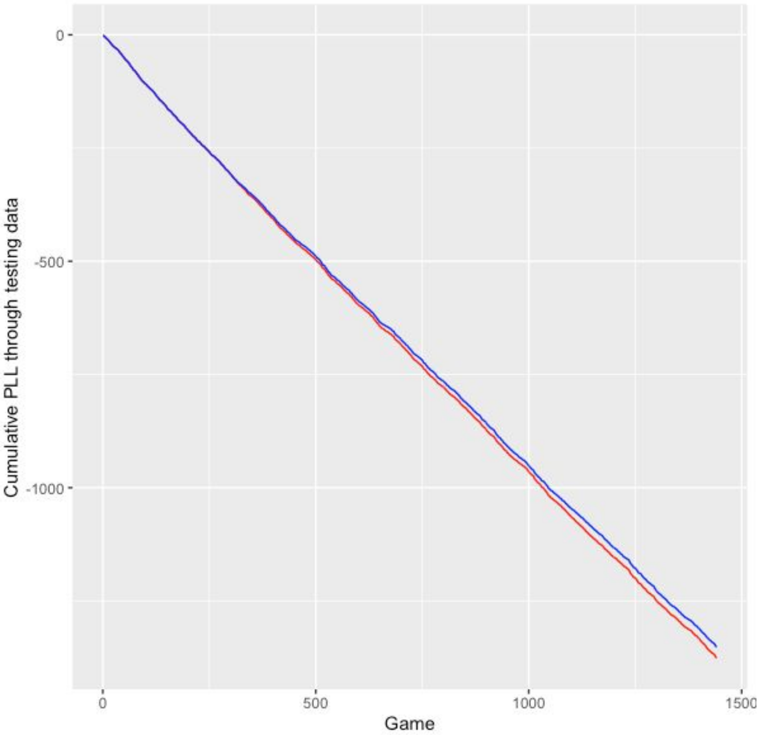
Despite exploring other variables to include in this model, there are several reasons for leaving this one untouched. The first idea was a variable for each team. There is little doubt that the inclusion of a team covariate would improve the PLL for certain games in a season. However, one arrives at the same conclusion as one did at the beginning of this section; a model that depends on a team variable across many seasons will only be good for some teams. Many teams have limited data points available. Some have no data at all- if one wanted to predict the 2017-18 season from any of the previous years' data, one couldn't possibly calculate a parameter for Brighton, who entered the league for the Premier League for the first time that season. Moreover, even if one could calculate a parameter for each team, it would have to be weighted so that recent games have more of an effect. And even if one could implement such a weighting system, how much more of the variability of the response variable would be explained by a team variable, that Elo hasn't explained already?

In an attempt to create more abstract variables, one could try and create variables that represent the 'form' of each team at each game. If one defines form as the accumulation of each game metrics (goals, goals against, shots, shots against, red cards etc) across the last five games for each team, would this be useful in our model? In short, none of the form metrics are classed as significant in the generalised linear model, once the Elo ratings have been accounted for (possibly because Elo rating is a measure of from anyway). Regardless, for each form metric to be calculated, one must wait around fifty fixtures in a season for each team to have played five games each, in the first place. So even if it had an effect on the model, it reduces our dataset and means our model won't make predictions at the start of the season which is something one would want a general model to do.

In order to keep the model general and versatile, only the Elo ratings are used to predict games. This allows us to predict every game of each season for many different leagues. Later parts of the report will look into more specific predictions.

## Elo Model Evaluation

Applying this model to the testing dataset gives a mean PLL value of -0.9552. Clearly this model performs similarly to the bookmakers', at least under the PLL measure. Using the same training data, Bet365 (who historically perform the best under the PLL measurement), achieved a mean PLL of around -0.941. One can visualise the evolution of PLL in the Elo model, coloured red, and the implied bookmaker's probability, coloured blue, across the training dataset:



From the picture, this model approaches a similar degree of accuracy in its predictions as the betting companies do. This is a very positive result- this model depends only on Elo rating and nothing else, whereas huge companies like Bet365 with many employees will be able to adapt and tailor their odds to each and every game. They will undoubtedly have more data available, as well as information like transfers and appointments of new managers which can have an immediate effect on the outcome of games, something Elo rating doesn't necessarily react as quickly to, as discussed later.

So, whilst this model hasn't beaten the bookmakers in our predictions, there are a few positive results:

1. It performs significantly better than simple models; it has a mean PLL 11.2% better than the home advantage alone.
2. It isn't considerably worse than the best of the bookmakers with a mean PLL rating of only 1.5% lower.
3. This model is highly generalisable to other leagues, one of the aims of the report. It is fairly low maintenance, requiring only the teams involved and the final score as data.

To elaborate on point 3 further, one can test this theory on the 2018/19 Serie A season and achieve a mean PLL of -0.9745 compared to Bet365's -0.9448. To an extent, our hypothesis was true: our model still does a good job of predicting (around 3% worse PLL than the bookmakers) compared to simple models like the home advantage alone. However, it must be noted that it performs worse compared to bookmakers on the Serie A than on the Premier League. Perhaps this is because the model is trained on Premier League fixtures and assumed the home advantage is the same in both leagues. Furthermore, parameters in the model may have been influenced by the 'predictability' of the league it was tested on. If one league is more predictable - teams of higher quality beat teams of lower quality more frequently - perhaps the Elo rating would influence the predictions more.

If one wants to have a better insight into the probability of each result happening, the Elo model is useful. It provides far better insight than blindly expecting the home team to win. In this regard, and because no football predictions can be made with 100% accuracy, it does a good job of answering the question posed by the report.

The model is extremely flexible. It can be used to predict any game of any season pretty well. However, it must be noted that it performs marginally better on the Premier League because this is the data the model was trained on. So there are some limits on the use of the model (though it was never claimed that this model was highly predictive in the first place). One could train the data on the league one wishes to predict, but this removes the intrinsic convenience associated with using such model.

To conclude this section, the use of the Elo GLM is far more informative than simple models and extremely versatile in its use. However, it fails to reach quite the same quality in its predictions as the bookmakers do, especially when predicting results of other leagues. The idea of this model is not to be finely tuned to each and every game, but to be useful in fulfilling one of the aims of this report: to predict any football match, regardless of league. But what if one requires the kind of accuracy in predictions to beat the bookies? Betting syndicates surely wouldn't place all of their money into a general model like this one. In order to make money from the fine margins in the betting markets, our model must be highly specific to each game. One must use models that aren't necessarily as easy to apply but are able to exploit weaknesses in bookmakers' predictions. In the next section, models of this nature are explored.

# 2.Poisson and Dixon Coles Models

In general, the idea of the Poisson regression model here is to use the maximum likelihood to formulate parameters to characterise each teams' level of offensive and defensive skill and hence to predict the probabilities of each game result.

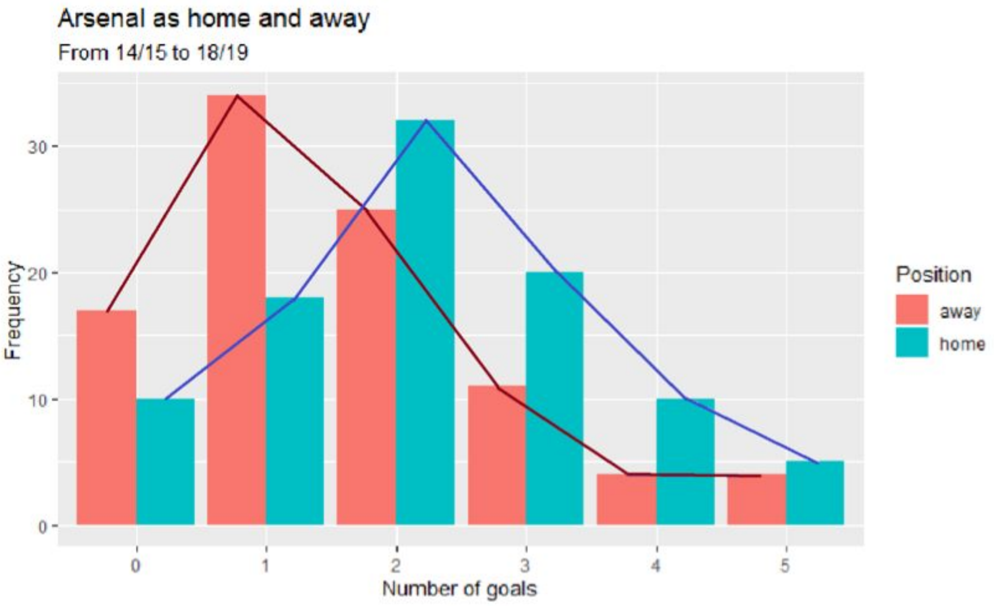
## Poisson Distribution

Before building the model of Poisson regression, it's necessary to assume that the number of goals for each team follow independent Poisson distributions. In addition to the plethora of academia using such distribution, this assumption is further justified by the following idea: in theory, one of the key concepts in football is the possession of the ball. During each possession, a team will try to attack and score against the defence of the opposition, with a small probability of success (scoring) of $p$. Since the number of possessions for each team is ideally very large and, under the assumption that $p$ is constant and each attack is independent, the number of goals follow the binomial distribution which approximates the Poisson distribution according to the Poisson limit theorem.

Poisson limit theorem:

$$\lim_{n\to\infty} \binom{n}{k} p^k (1-p)^{n-k} = e^{-\lambda} \frac{\lambda^k}{k!}$$

If this isn't enough, the Poisson distribution can also be shown by an example from the data. Here is a plot of the number of goals Arsenal score, broken down into home and away games. Clearly, the goals generally follow the Poisson distribution:



## Poisson Regression Model

Before showing the model, the following assumptions are made:

1. The number of goals for each team follow a Poisson distribution
2. The number of goals between each team and their opponent is independent
3. Each score-line is independent from match to match

Here is the model:

$$P(X_{ij} = x, Y_{ij} = y|\lambda, \mu) = \frac{\lambda^x e^{-\lambda}}{x!}\frac{\mu^y e^{-\mu}}{y!} \qquad (1)$$

$X$ and $Y$ are the random variables for the number of goals scored in the game where team $i$ plays against team $j$.

From the assumptions, these two random variables are independent so that the joint probability of the home team scoring  goals and the away team scoring  goals is the product of the two independent probabilities.

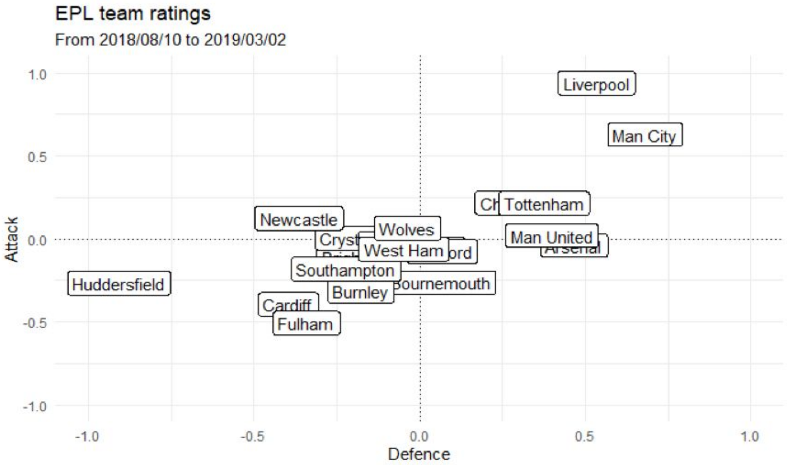
There are several ways to evaluate the expected value of parameters $\lambda$ and $\mu$. One way is to express it by the attacking ability of one team, the defensive ability of the other team and the existence of the home advantage, which has been proved to affect the result of the game.

Here it is:

$$\lambda = exp(\alpha_i - \beta_j + \gamma)$$

$$\mu = exp(\alpha_j - \beta_i)$$

Where $\alpha_m$ , and $\beta_m$ represents the attacking and defensive ability of the team $m$ respectively and $\gamma$ corresponds to the effect of the aforementioned home advantage. To visualise the values of $\alpha$ and $\beta$ for each team, here is a plot for the 2018/19 season:



After substituting the expected values into the pmf(1) and taking the Log, the expression turns into:

$$logP(x, y|\lambda, \mu) =-(\alpha_i - \beta_j + \gamma) - (\alpha_j - \beta_i) + x(\alpha_i - \beta_j + \gamma) \ + y(\alpha_j - \beta_i) - log(x!) - log(y!)$$

If $i$ is a game instance and $k$ games are observed in total, then the log-likelihood of these k games would be:

$$l = \sum_{i=1}^{k} logP(x_i, y_i|\lambda_i, \mu_i)$$

To avoid multiple combinations of parameters which may produce the same model, this constraint is added:

$$\sum_i \alpha_i = 0, \ \sum_i \beta_i = 0$$

After using the optimisation method to calculate the maximum log-likelihood, the value of each of the parameters could be obtained. The probability of each result (home win/draw/away win) can be forecasted like so:

$$P(Home\ win) = \sum_{x=1}^{\infty} \sum_{y=0}^{x-1} P(x, y|\lambda, \mu)$$

$$P(Draw) = \sum_{x=1}^{\infty} P(x, x|\lambda, \mu)$$

$$P(Away\ win) = \sum_{y=1}^{\infty} \sum_{x=0}^{y-1} P(x, y|\lambda, \mu)$$

## Dixon-Coles Model

Different from the assumption in the Poisson regression model, the Dixon-Coles model assumes that the score-lines 0-0, 1-0, 0-1 and 1-1 are not independent. Therefore, one probability modified function is to be added to the Poisson regression model.

$$\tau(x, y, \mu, \lambda, \rho) = \begin{cases} 1 - \lambda\mu\rho & x = 0, y = 0 \\ 1 + \lambda\rho & x = 0, y = 1 \\ 1 + \mu\rho & x = 1, y = 0 \\ 1 - \rho & x = 1, y = 1 \\ 1 & otherwise \end{cases}$$

and the Dixon-Coles model is:

$$P(X_{ij} = x, Y_{ij} = y|\lambda, \mu, p) = \tau(x, y, \mu, \lambda, p)\frac{\lambda^x e^{-\lambda}}{x!}\frac{\mu^y e^{-\mu}}{y!}$$

The modified function is the only difference between the simple Poisson model and the Dixon-Coles Model, which means the ways to calculate the parameters are generally the same (by MLE).

### Time weighting

During building the model for game forecasting, the more recent results should be more significant as they better reflect the current level of the team. Therefore, the weight function is introduced and used here:

$$\emptyset(t) = \begin{cases} \exp(-\xi t) & t \leq t_0 \\ 0 & t > t_0 \end{cases}$$

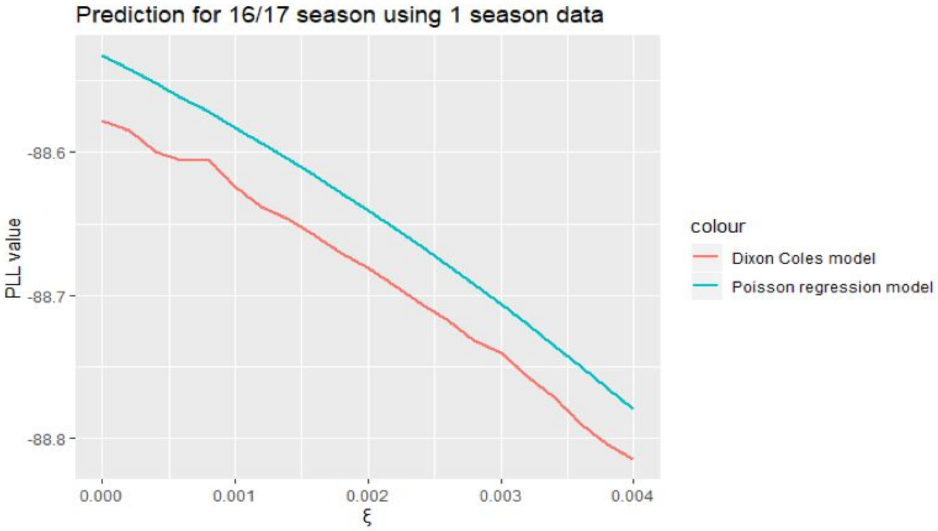
Where $\xi$ is a constant and $t_0$ is a time at which our prediction is being made. This ensures that, at each prediction, older data has less of an effect than more recent data on our model. The former log-likelihood for Poisson regression turns into:

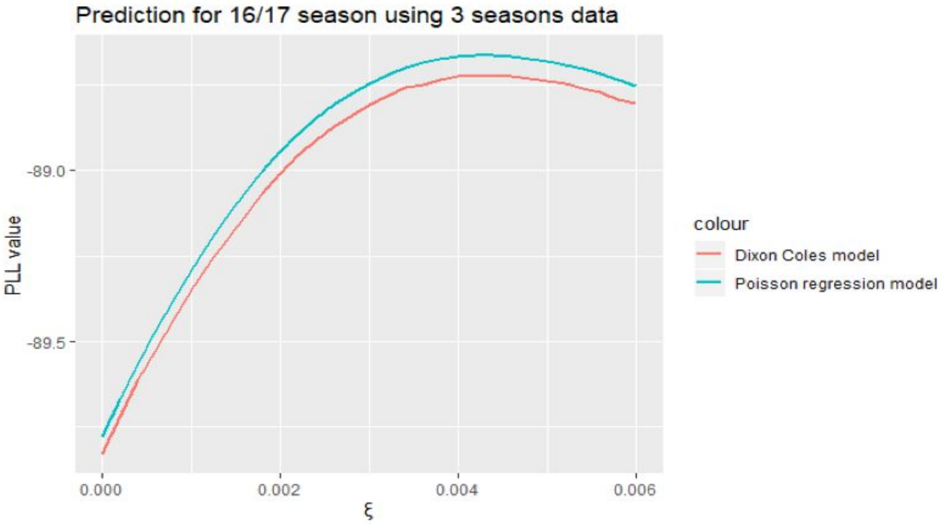
$$l = \sum_{i=1}^{k} \emptyset(t_i) log P(x_i, y_i|\lambda_i, \mu_i)$$

Therefore, for each change in the value of $\xi$, the value of the parameters in the model will be different by calculating the MLE. Following this, a different mean PLL for each model can be calculated. The $\xi$ with the highest mean PLL may represent that such model could predict the results best and that should be used for the foregoing forecasting.

In addition, it could be seen that the weight function depends on $t_0$. However, $t_0$ is just a point, not an interval, which means the weight function needs to be shifted after each prediction when forecasting more than one date. In our case, when calculating the optimal value of $\xi$, one must first mention how the characteristics of the dataset is changed for every prediction. After a group of fixtures is forecasted on a given date, $t_0$ will be shifted for the next predictions and the (now realised) fixtures that one first predicted will be added to the dataset and will have a strong weighting on our model. The older data will therefore be shifted further back and will have less weighting on our model.

Here are plots of the mean PLL for predicting the last 100 games in the 16/17 season using the different $\xi$ given by the Poisson regression model and the Dixon-Coles model. The first plot is using 1 season's data and the second uses 3.





It can be seen that, in order to maximise the mean PLL, both Dixon Coles model and Poisson regression model could predict best without adding time weight when using 1 season's data to build the model. However, when using the 3 seasons data to build the model, $\xi$ of weight function should be set as 0.0044 for the best predictions.

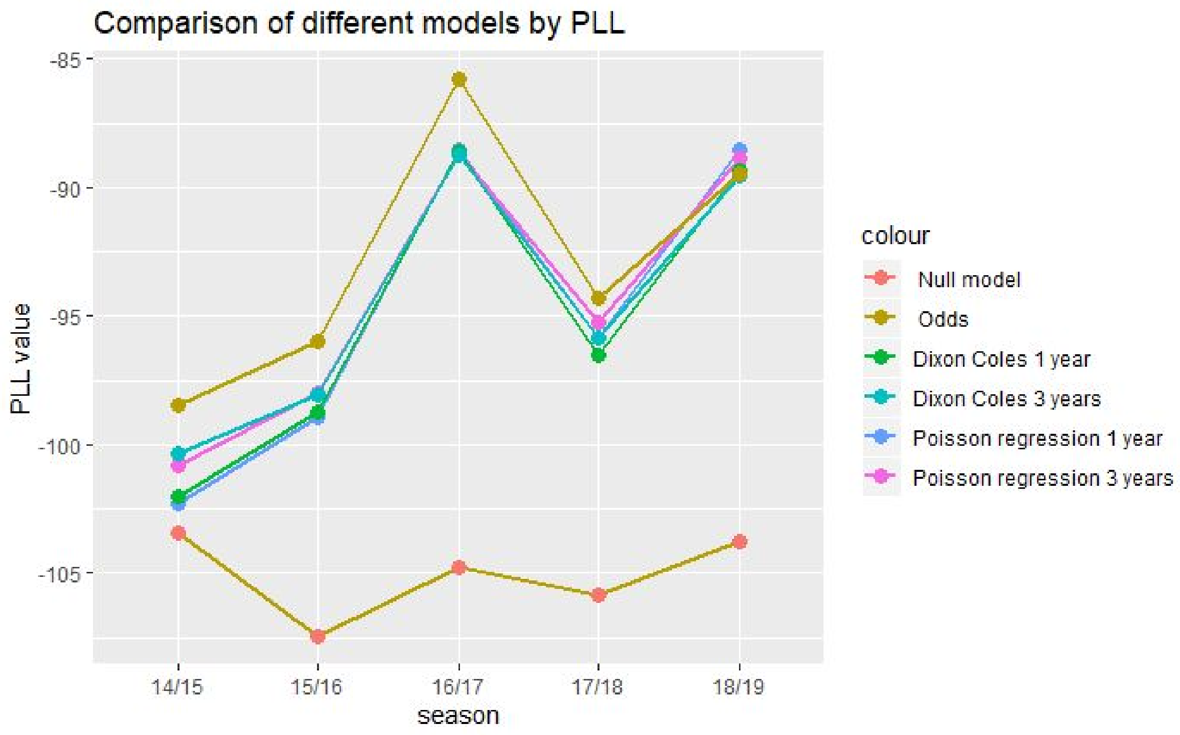
## Predicting the Results

To predict the last 100 games in 5 seasons, Dixon Coles model and Poisson regression model using 1 season and 3 seasons data are built and the mean PLL is calculated to evaluate the different models. In addition, the PLL of the implied probability given by bookmakers' odds is also calculated as a reference.

To justify predicting only the last 100 games of each season, it can be observed in football that there is an underlying uncertainty of results at the start of the season due to obvious changes in personnel at certain clubs over the transfer window (new signings and managers), newly promoted teams of unknown quality and perhaps, more conjecturally, a 'new season mentality'; results at the beginning of the season can be erratic, with bad teams of the previous season starting with the optimism of a clean slate and good teams potentially losing their end of season form.

In this section of the report, one is concerned only with predicting games with the highest possible accuracy. Perhaps for the previous part of the season, one would use the Elo generalised linear model since this, to some capacity, combats the problem associated with such uncertainty with the help of a few assumptions, with the drawback of some accuracy.

Below is the comparison plot of each model:



From the plot, it could be seen that all the models are able to predict much better than the null model (a model which always assigns $P(Home\ win) = 0.4634,\ P(Draw) = 0.25,\ P(Away\ win) = 0.2866$) and the predicting ability between each model is very similar. There was not a model that could always obviously predict better than the others. Although the predictability of these models could not catch up with the odds by bookmakers, it can achieve a very similar degree of accuracy. In addition, it is evident that there is an undeniable similarity between the shapes of all of our models and the bookmakers', that the null model doesn't exhibit. To this end, it is highly probable that the bookmakers' models are also informed by some slight deviation from the Poisson model, like our four. At the very least, even if this model predicted far worse than the bookmakers', the qualitative behaviour of the graphs is so similar that it, to some extent, has shed light on the sort of model that bookmakers have used.

This is encouraging because, whilst our models perform worse for all but the last season, this type of model is clearly on the same track as the bookmakers'. Resources tell us that many bookmakers' models are dependent on both their model (which was just theorised to be similar to the Poisson model), but also the group betting behaviour of thousands of customers. Hence, as well as having a dataset which may be more complex than ours, they are also able to make adaptations to their predictions based on the consensus of the betters. This would explain why their predictions, in general, are marginally better.

The table below shows the mean PLL of these 4 kinds of models, which would be helpful to compare with the models generated by the methods of different types.

| | Dixon Coles model | Poisson regression model |
|---|---|---|
| 1 year data | -0.9504066 | -0.9481608 |
| 3 years data | -0.9439942 | -0.9424600 |

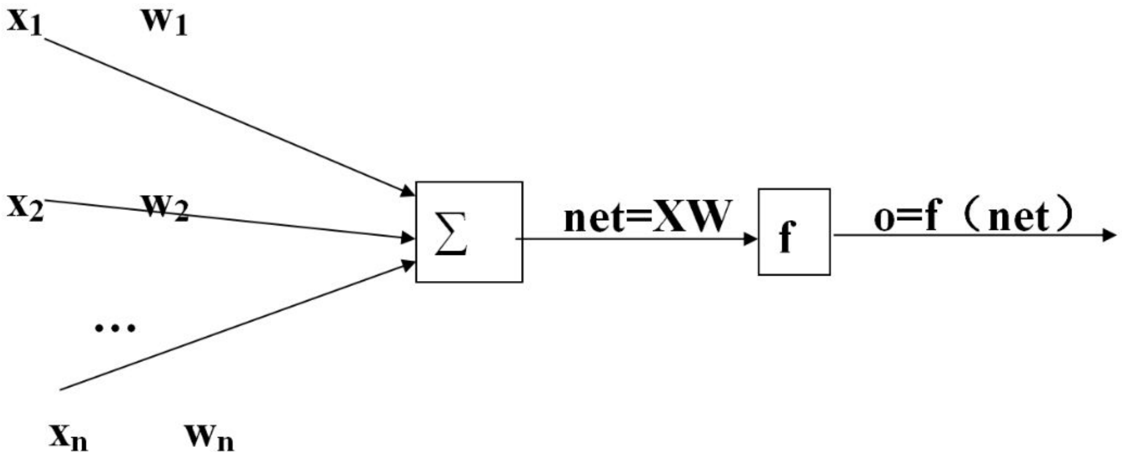
# 3.BP ANN

Artificial neural networks (ANN) are a simulation of human neural networks. More directly, it is a mathematical model, which can be achieved by computer or electric circuit. It has been found to be an effective way to research Artificial intelligence (AI).

Simpson (1987) claimed ANN is a non-linear directed graph. The graph contains some sides with weight which can be changed, and the ANN is able to find the pattern from incomplete or unknown input. Artificial neurons are the basic unit of ANN, which is a simulation of biological neurons. It has six characteristics:

1. Connect the neurons
2. The strength of connection between neurons determines the strength of signal transmission
3. The connection strength between neurons can be changed with training
4. Signals can be stimulating or inhibiting
5. The cumulative effect of signals received by a neuron determines its state
6. Each neuron can have a threshold

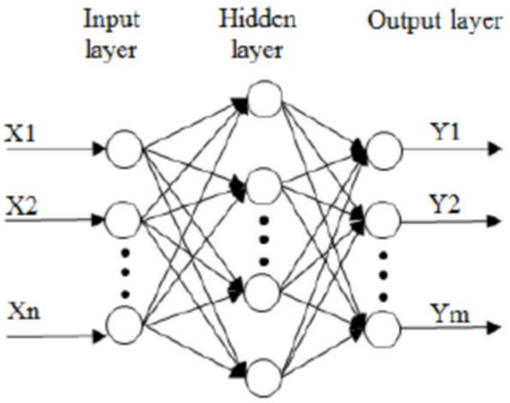
If one has inputs, $X = (x_1, x_2, ..., x_n)$ and weights, $W = (w_1, w_2, ..., w_n)$, the net input is given by $net = \sum_i x_i w_i$ and passed through an activation function $o = f(net)$. This process can be visualised below:



Werbos (1974) raised the back-propagation learning algorithm for multilayer feedforward networks which was developed by Rumelhart in 1986 (Rumelhart et al. 1986).

Back-propagation artificial neural network (BP ANN) often contains one input layer, one or more hidden layers and one output layer. There are a certain quantity of neurons in every layer and the neurons are connected by a certain weight to the next layer's neurons. Once these neurons work together, the network can solve problems. Each neuron accumulates the signals from the previous layer, and deals with the signal by an activation function (often a sigmoid function).

$$o = f(net) = \frac{1}{1+e^{-net}}$$



Basic characteristics:

1. Network contains $L$ layers

2. Existence of Connection Matrix: $W^1, W^2, ..., W^L$

3. Sample set $S = \{(x_1, y_1), (x_2, y_2), ..., (x_s, y_s)\}$

4. Neurons in layer k is $H_k$

According to the samples $(x_i, y_i)$ in the sample set, calculate the actual output $O_i$ and calculate the error between the predicted and observed output, then adjust the weights $(W_1, W_2, ..., W_L)$ and repeat this procedure until $E = \sum_p E_P \leq \varepsilon$ where $E_P = \frac{1}{2}\sum_{j=1}^{m}(y_{pj} - O_{pj})^2$ .

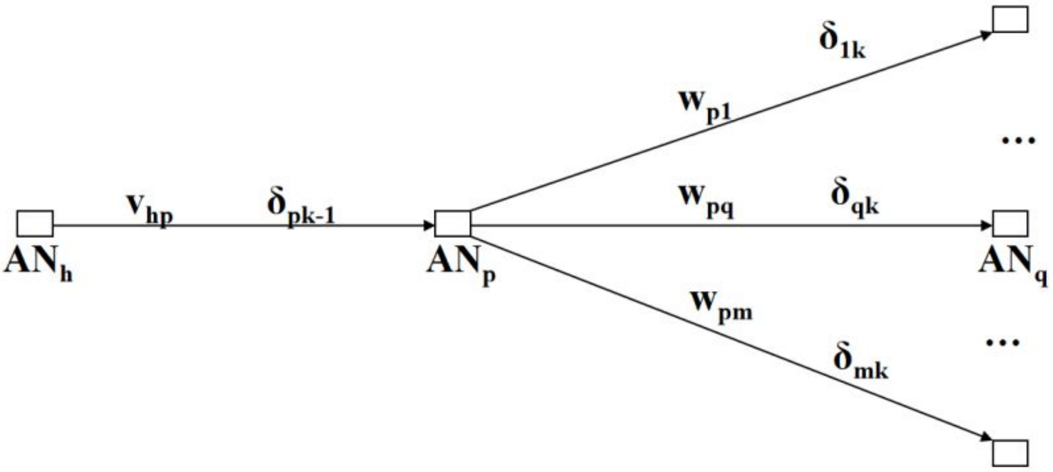
The adjustment of connecting weights of $pth\ W$ can be found by using the steepest descent method (Rumelhart et al. 1986) as below:

$$\Delta_p W_{ji} = \alpha \delta_{pj} O_{pi}$$

Where α is the learning ratio, which can control the change of step. In addition, the adjustment of output and hidden layers is given by:

$$\Delta w_{pq} = \alpha \delta_q o_p = af'(net_q)(y_q - o_q)o_p = \alpha o_q(1 - o_q)(y_q - o_q)o_p$$

$$\Delta V_{hp} = \alpha o_{pk-1}(1 - o_{pk-1})(w_{p1}\delta_{1k} + w_{p2}\delta_{2k} + \cdots w_{pm}\delta_{2m})o_{hk-2}$$



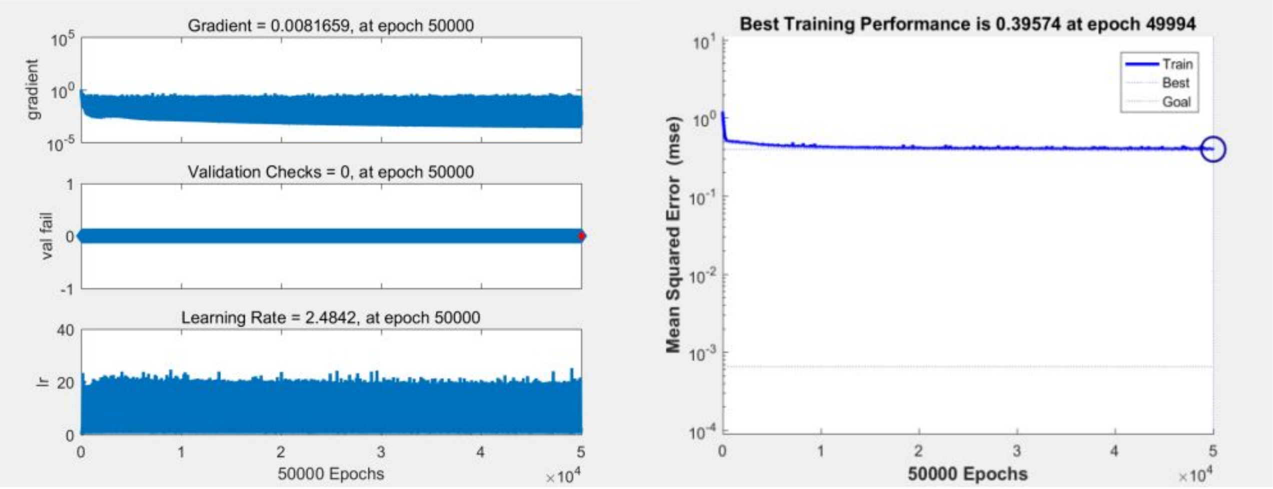
After searching the effect of every covariate, the data from Premier League in 17/18 (380 matches) is seen as the training data. If one finds the cumulative statistics of the first 280 matches in the season, one finds that the significant inputs that affect the results of the game are the average of [Home shots (HST), Home corners (HC), Home red cards (HR), Away shots (AST), Away corners (AC), Away red cards (AR)]. For example, if Arsenal are the home team, calculate the average HST, HC, HR, If Liverpool are the away team, calculate the average AST,AC,AR. More clearly:

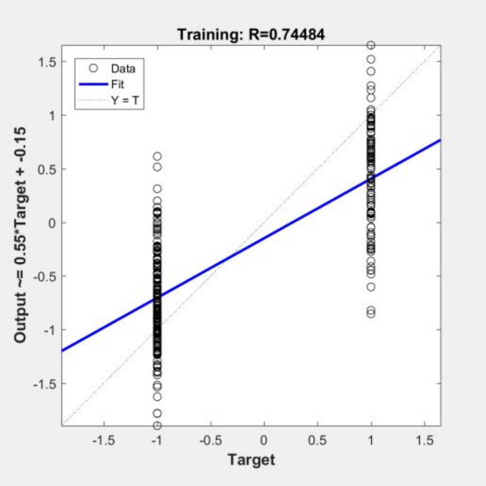
- Input:[Arsenal[HST],Arsenal[HC],Arsenal[HR],Liverpool[AST],Liverpool[AC],Liverpool[AR]]
- Output: if home win: [1,0,0], if draw: [0,1,0], if away win: [0,0,1].

For testing, this model selects the data of Premier League in 18/19 (380 matches). the first 280 matches are used to train the model, and the last 100 matches test (like the previous Poisson model). In this network, the number of neurons in the hidden layer and output layer is 6 and 3 respectively connected by log-sigmoid function and 'purelin' function, and using 'traingdx' as the training function for gradient descent. Using Matlab, some parameters used are shown below:

- trainParam.show=2000
- trainParam.Lr=0.01
- trainParam.epochs=50000
- trainParam.goal=0.65*10^(-3)

The figures below show some training details of the model like performance over time, training states and regression:





When the output is found, employ the softmax function to transform it to a probability of home win, a draw or an away win. The standard (unit) softmax function that takes $\sigma : R^k \rightarrow R^k$ is defined by the formula:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for} \quad i = 1, \ldots K \quad \text{and} \quad z = (z_1, \ldots z_K) \in R^K$$

After analysing the result, the BP ANN predicts 57% of the last 100 matches of the 18/19 season and has a mean PLL=-1.012. as the output shown in the appendix.

# Conclusion:

In conclusion, each model introduced works well under different circumstances. If one requires a decent and convenient approximation to the probabilities of each outcome, but doesn't necessarily care much about the accuracy of such prediction, the Elo model is the most logical to use. Elo rating in football is intuitive and easy to understand, so this model is more widely accessible to readers than the Poisson and neural network model. However, if one is more concerned with making money from the margins of error in bookmakers' predictions, it is far more useful to use other ideas.

The Poisson model is also intuitive in the way it works. Intrinsically, football is concerned with scoring the most goals. Attacking with quality and defending with strength is essential in achieving this. Hence, the inclusion of a Poisson model with parameters that depend on attacking and defensive ability makes a lot of sense. Though it may not have the flexibility of the Elo model, nor does it predict results at the start of the season, it generally predicts the results more accurately. It also goes some way to explain the type of models that bookmakers may use.

Finally, the model based on neural networks is less intuitive; the parameters that it includes in its predictions is less obvious. One need not know why the covariates are selected, only that it has an effect on our result. Though it doesn't necessarily perform better on the single season testing data than the Poisson model, it has lots of room for improvement by continuously learning throughout time, adapting its parameters on-the-go. Although our results for the neural network are not as good as the Poisson model (for now at least), its inclusion in this report is highly necessary because it represents the type of high level prediction methods that bookmaking companies might employ. We may not have found the best model using this method, but it is certainly likely that neural networks are the key to unlocking similar, or even better, predictions than the bookmakers. With more education on this area of mathematics, one may have provided better predictions. If one were to use this method in the future, it may have a lot of potential.

# Discussion and Limitations:

## Teams in transition:

Whilst the data used in the formation of the model are extensive they are not exhaustive; there exist other factors which may affect the outcome of a game, which may not be simple to incorporate into a model. An example of this (with varied severity) is player suspensions and injuries; many teams are highly dependent on a couple of key players, whose absence could negatively affect a team's chances. For instance, a striker who is a team's main goal contributor will likely be missed. This is not particularly simple to consider when modelling, but may be considered by betting companies, and thus may explain why bookies appear to do better in some cases.

Similarly, it is often the case that a team playing particularly badly will look to change their manager. Whilst our model will not anticipate or identify this, "new manager bounce" is very often seen at a club, whereby a turnaround in form occurs when a new manager takes over. This may be another factor that is neglected, but whose consideration ought to be able to improve our model. These game-specific factors may sometimes allow the bookies an edge.

### Quality of Data

One should also consider the data available. If this approach were used on a team who were newly promoted, there would be little-to-no data to make use of yet, and similarly with newly relegated teams. This drawback is not particularly easy to negotiate, but one that would only serve as an issue for a very early period of a season, until enough data are gathered.

A more detailed dataset would also be surely beneficial. In recent years, new metrics have been developed that would provide much more information that one could extract and use in our modelling. Advanced statistics like expected goals, passes allowed per defensive action and many more could provide useful information about football matches that can bring us closer to the bookmakers' predictions.

## References

- https://stuartlacy.co.uk/2017/08/31/implementing-an-elo-rating-system-for-european-football/
- Werbos P (1974) Beyond regression: new tools for prediction and analysis in the behavioral sciences. PhD Thesis, Department of Applied Mathematics, Harvard University, Cambridge
- Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. In: Rumelhart DE, McClelland JL (eds) Parallel distributed processing: exploration in the microstructure of cognition. MIT Press, Cambridge, pp 318–362

- https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0198668#pone.0198668.s002
- https://opisthokonta.net/?p=890https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-9574.1982.tb00782.x
- M. J. Dixon and S. G. Coles, Modelling association football scores and inefficiencies in the football betting market, J. R. Stat. Soc.—Wiley Online Library, 46 ( 2) (1997), 265– 280.

# Appendix

## Poisson Code

(Part of the codes in Poisson model to show the authenticity of the data)

~~~~
```{r}
install.packages("devtools")
devtools::install_github("opisthokonta/goalmodel")
```
~~~~

~~~~
```{r}
library(goalmodel)
library(dplyr) # Useful for data manipulation.
```
~~~~

~~~~
```{r}
edit<-function(data){data<-data[,c(2:7)]
names(data)<-c("Date","home","visitor","hgoal","vgoal","result")
data[order(data$Date,decreasing=TRUE),]
return(data)}
```
```{r}
edit_time<-function(data){
  data$Date<-as.character((data$Date))
  if (nchar(data$Date[1])==8){
data$Date<-paste("20",substr(data$Date,7,8),substr(data$Date,3,6),substr(data$Date,1,
2),sep = "")
  }

else{data$Date<-paste(substr(data$Date,7,10),substr(data$Date,3,6),substr(data$Date,
1,2),sep = "")}
return(data)
~~~~

~~~
}
```
```{r}
xi<-function(data,predict){

sum_xi=0
for (i in 1:dim(data)[1]){
if (data[i,]$result=="H"){
sum_xi=sum_xi+log(predict$p1[i])}
else if (data[i,]$result=="A"){
sum_xi=sum_xi+log(predict$p2[i])}
else{sum_xi=sum_xi+log(predict$pd[i])}
}
return(sum_xi)}
```

```{r}
data1213<-edit_time(edit(read.csv("C:\\Users\\lidai\\Desktop\\Football\\1213.csv")))
data1314<-edit_time(edit(read.csv("C:\\Users\\lidai\\Desktop\\Football\\1314.csv")))
data1415<-edit_time(edit(read.csv("C:\\Users\\lidai\\Desktop\\Football\\1415.csv")))[1:380,]
data1516<-edit_time(edit(read.csv("C:\\Users\\lidai\\Desktop\\Football\\1516.csv")))
data1617<-edit_time(edit(read.csv("C:\\Users\\lidai\\Desktop\\Football\\1617.csv")))
data1718<-edit_time(edit(read.csv("C:\\Users\\lidai\\Desktop\\Football\\1718.csv")))
data1819<-edit_time(edit(read.csv("C:\\Users\\lidai\\Desktop\\Football\\1819.csv")))
#data<-rbind(data1415,data1516,data1617)
#data<-data1617
data1415_n<-edit_time(read.csv("C:\\Users\\lidai\\Desktop\\Football\\1415.csv"))[1:380,]
data1516_n<-edit_time(read.csv("C:\\Users\\lidai\\Desktop\\Football\\1516.csv"))
data1617_n<-edit_time(read.csv("C:\\Users\\lidai\\Desktop\\Football\\1617.csv"))
data1718_n<-edit_time(read.csv("C:\\Users\\lidai\\Desktop\\Football\\1718.csv"))
data1819_n<-edit_time(read.csv("C:\\Users\\lidai\\Desktop\\Football\\1819.csv"))
#View(data)

```



```{r}
#input data
k<-100
s<-1



sum_xi<-0
~~~

~~~
data<-rbind(data1617,data1718,data1819)
train_data<-data[1:(dim(data)[1]-k),]
test_data<-data[(dim(data)[1]-k+1):dim(data)[1],]

while (!(is.null(test_data))){
my_weights <- weights_dc(train_data$Date, xi=0.0044)

gm_res_dc <- goalmodel(goals1 = train_data$hgoal, goals2 = train_data$vgoal,
                     team1 = train_data$home, team2=train_data$visitor,dc=TRUE,
          weights = my_weights)

if (dim(test_data)[1]>s){
  predict<-predict_result(gm_res_dc, team1=test_data[1:s,]$home,
team2=test_data[1:s,]$visitor, return_df = TRUE)
sum_xi<-sum_xi+xi(test_data[1:s,],predict)
train_data<-rbind(train_data,test_data[1:s,])
test_data<-test_data[(s+1):dim(test_data)[1],]}

else{
   predict<-predict_result(gm_res_dc, team1=test_data$home, team2=test_data$visitor,
return_df = TRUE)
sum_xi<-sum_xi+xi(test_data,predict)
train_data<-rbind(train_data,test_data)
test_data<-c()}
}
sum_xi


```
~~~

## Elo Code

```
centraliser <- function(s1.end.elo){
  s1.centralised <- s1.end.elo
  for (i in 1:20){
    new.elo = s1.end.elo[i,2]*0.8 + mean(s1.end.elo$Elo)*0.2
    s1.centralised[i,2] = new.elo
  }
  s1.centralised
}

##############################################################################
##############################################################################

promotion.finder <- function(s1.end.elo,new.season){
```

```
  s2.team.vector <- NULL
  k=0
  for (i in 1:380){
    if (is.element(new.season[i,3],s2.team.vector) == FALSE){
      k=k+1
      s2.team.vector[k]<- as.character(new.season[i,3])
    }
  }
  promoted <- as.character(setdiff(s2.team.vector,as.vector(s1.end.elo$TeamName)))
}

##############################################################################
##############################################################################

relegation.finder <- function(s1.end.elo,new.season){

  s2.team.vector <- NULL
  k=0
  for (i in 1:380){
    if (is.element(new.season[i,3],s2.team.vector) == FALSE){
      k=k+1
      s2.team.vector[k]<- as.character(new.season[i,3])
    }
  }
  relegated <- as.character(setdiff(as.vector(s1.end.elo$TeamName),s2.team.vector))
}

##############################################################################
##############################################################################

next.season.elo <- function(relegated,promoted,centralised.Elo){

  sorted.old.elos <- centralised.Elo[order(centralised.Elo$Elo),]
  mean.bottom.3 = (sorted.old.elos[1,2]+sorted.old.elos[2,2]+sorted.old.elos[3,2])/3

  s2.elos <- centralised.Elo
  pre.team.names <- as.vector(s2.elos$TeamName)
  Elo <- as.vector(s2.elos$Elo)
  k=1

  for (i in 1:20){
    if (is.element(pre.team.names[i],relegated)==TRUE){
      pre.team.names[i] = promoted[k]
      Elo[i] = mean.bottom.3
      k=k+1
    }
  }

  TeamName <- pre.team.names
  new.season.elo <- data.frame(TeamName,Elo)
}

######################################################################
######################################################################

elo.visualised <- function(full.06_19.elo,Team){

  useful.elo <- full.06_19.elo[,c(2,3,4,69,70)]
  team.elo.vector <- NULL
  date.vector <- NULL
  k=1

  for (i in 1:4940){
    if (useful.elo[i,2]==Team){
      team.elo.vector[k]=useful.elo[i,4]
      date.vector[k]=k
      k=k+1
    }

    else if (useful.elo[i,3]==Team){
      team.elo.vector[k]=useful.elo[i,5]
      date.vector[k]=k
      k=k+1
    }

  }

  plot.df <- data.frame(team.elo.vector,date.vector)
  plot <- ggplot(plot.df,aes(date.vector,team.elo.vector))+geom_line()
  plot <- plot + labs(title = Team , x = "Game Number", y = "Elo Rating")
  plot <- plot + theme(axis.text.x = element_blank(),axis.ticks = element_blank()) +
ylim(1200,1900)
  plot <- plot + geom_hline(yintercept=1500, linetype="dashed", color = "red")
}

plotMU <- elo.visualised(full.06_19.elo,"Man United")
plotMC <- elo.visualised(full.06_19.elo,"Man City")
```

```
plotL <- elo.visualised(full.06_19.elo,"Liverpool")
plotT <- elo.visualised(full.06_19.elo,"Tottenham")
plotA <- elo.visualised(full.06_19.elo,"Arsenal")
plotC <- elo.visualised(full.06_19.elo,"Chelsea")

ggarrange(plotMU,plotL,plotT,plotMC,plotA,plotC,ncol = 2, nrow = 3)
```