FINAL REPORT

Predicting Professional Football Match Outcomes with Artificial Intelligence

Abstract

'Sports have been part of society for millenniums, the earliest known being Wrestling dating back 15,300 years [1]. The first interest and value of guessing the outcome of a match is thought to be the Greeks over 2000 years ago, who would place wagers on the gladiator games. Predicting outcomes in sports is now used in various industries such as, sports pundits on television discussing who will win a particular game as a form of entertainment for a viewer, sports fans giving their opinion on who they think will win, or most notably the sports betting industry which is worth 203 billion US dollars as of 2020 and expected to grow by another 8 billion by 2025 [2]. There is a clear want and business value in predicting outcomes, and therefore investment into technology and techniques to achieve this.'

Team BAWKOS

University of Surrey

Jamie Wiggins, Joseph Kutler, Nkem Ogosi, Manraj Sidhu, Syed Ali Raza, Stefano Bussandri

Contents

1 Introduction

1.1 Project Mission

Predicting the outcome of a football match (or result of any sporting event) is a complex and challenging task due to the large variety of factors and randomness involved. Project BAWKOS strives to create and compare models and algorithms that predict professional English Premier League football match outcomes (win, loss, or draw) using artificial intelligence. The project aims to build models that are as accurate as possible, and at minimum better than a random or informed guess.

1.2 Background

1.2.1 Overview

Sports have been part of society for millenniums, the earliest known being Wrestling dating back 15,300 years [1]. The first interest and value of guessing the outcome of a match is thought to be the Greeks over 2000 years ago, who would place wagers on the gladiator games. Predicting outcomes in sports is now used in various industries such as, sports pundits on television discussing who will win a particular game as a form of entertainment for a viewer, sports fans giving their opinion on who they think will win, or most notably the sports betting industry which is worth 203 billion US dollars as of 2020 and expected to grow by another 8 billion by 2025 [2]. There is a clear want and business value in predicting outcomes, and therefore investment into technology and techniques to achieve this.

The use of Artificial intelligence (AI) in the sporting industry has grown substantially in the last decade and is not slowing down, with an expected CAGR (compound annual growth rate) of 28.72% in the period 2021-2026 [3]. This is in due to the complexity of patterns in data that can be understood compared to more traditional methods like linear regression. And against a human making a prediction there is no comparison, a human simply is not able to comprehend or process the number of features and size of data that can be assessed by AI based models. However, predicting any sporting result is notoriously complex to achieve, even with the use of AI.

It is widely accepted that getting 100% prediction accuracy is impossible due to the unpredictability that exists within sport, more so in football as it has three outcomes, win, loss, draw, versus the traditional two, win/loss. The greatest example in modern times of unpredictability is when Leicester won the league in 2016 with odds of 5000/1 [4]. This project aims to evaluate various AI based techniques in predicting the outcome of English Premier League football [5] games and achieve an accuracy as high as possible.

Whilst getting 100% is considered impossible, the challenge remains, how high an accuracy can be achieved?

1.2.2 Machine Learning Techniques

There are two key approaches that make up the field of machine learning: Supervised and Unsupervised Learning which in turn solve three unique problem types. These are Classification, Regression and Clustering problems.

1.2.2.1 Supervised Learning

Supervised learning utilises labelled datasets to train algorithms that can classify data or deploy a regression prediction. As input data is fed into the model it adjusts its weights until the model has

been fitted appropriately. This can be done using a variety of methods, the most common being the holdout method and k-Fold Cross Validation. Supervised learning uses a training dataset to teach models to learn over time about how to obtain the desired output whereupon the accuracy is measured through a loss function, adjusting until the error has been sufficiently minimised. Once the model has been adequately trained, a testing dataset is executed upon so that it can be seen how a model performs when faced with unseen data.

1.2.2.2 Unsupervised Learning

Unsupervised learning uses machine learning algorithms to analyse and cluster unlabelled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. Its ability to discover similarities and differences in information make it the ideal solution for exploratory data analysis and image recognition problems.

1.2.2.3 Regression

Regression is used to understand the relationship between dependent and independent variables. This consists of an algorithm making use of complex mathematical methods to predict a continuous outcome based on the value of one or more predictors. Examples of where regression algorithms include Multiple Logistic Regression (MLR), Linear Regression and polynomial regression.

1.2.2.4 Classification

Classification uses an algorithm to accurately assign (classify) test data into specific categories, sometimes called classes. Classification algorithms recognise specific entities within a dataset and attempts to draw conclusions as to how these entities should be labelled or classified. Common examples of classification machine learning models are Support Vector Machines (SVM), Random Forests and k-Nearest Neighbour.

1.2.2.5 Clustering

Clustering is a data mining technique which groups unlabelled data based on their similarities or differences. Clustering algorithms are used to process raw, unclassified data objects into groups represented by structures or patterns in the information.

1.3 Research

1.3.1 Predictions in Football

Machine learning models can be used to predict the outcomes of football matches in various ways, and there are many examples of this. The best method is still debated and dependent on the data and features used. Most data sets contain date played, home team, away team, number of goals scored by each team in the game, outcome of the game, and a small variance of other stats such as, shots on target, total shots, and yellow and red cards for each team, but this is not even close to what is accessible in the private sector as touched upon in section 1.3.4 Data in Football and means publicly available examples are somewhat limited.

Football is one of the more difficult sports to predict. Firstly, there is three outcomes which takes the chance of a guess from 33.3% to 50% with sports that have only two outcomes (win or loss). Additionally, there is a large amount of unpredictability, just because a team is statistically better on paper does not mean they are guaranteed to win the game. There have been many examples of this over the years like when Manchester United, a top premier league team lost 4-0 to MK Dons of the third tier of English football [6]. It is also low scoring which compared to sports such as

basketball which is more predictable because of its scoring system which sees fairly high numbers [7].

There are two ways to predict the outcome of a game, predicting the score of each team, or predict one of the three possible outcomes. This lends football predictions well to a regression based and classification type problem respectively.

Bookmakers are notoriously taciturn about revealing how accurate their predictions are, as well as the fact that a financial incentive is considered. This means that companies may alter odds slightly to favour the businesses revenue. Each of the following three companies in the table below produce a weekly pre-game match prediction blog post where they attempt to predict the outcome of each game.

To work out an estimate for successful predictions by bookmakers the result with the lowest odds was interpreted as their prediction of the outcome. This prediction was compared to the actual outcome for each match, and an average accuracy score was computed for each game week based on how many predictions were correct. This was done for the first ten game weeks of the English Premier League season.

Bookmaker	Percentage of successful predictions
Ladbrokes - One of the largest high street sports betting and gambling company [8]	43%
Paddy Power - Major telephone and online sports betting service [9]	34%
Yahoo! Sports - Major sports news and gambling website [10]	40%

Football fans and experts will often give their views on a game and their prediction of what they think may happen. The chance of a guess here is 33.3% per outcome since there are three possibilities, win, loss, and draw. However, people who have an informed knowledge of the sport, particularly experts, are likely to have a better chance than this as they can assess other information and statistics better than having no knowledge and randomly picking an outcome. For example, former professional footballer, manager, and current Sky Sports pundit Paul Merson was able to make forecasts with a success rate of 58% [11]. However, this can then lead to bias and results in the prediction becoming skewed.

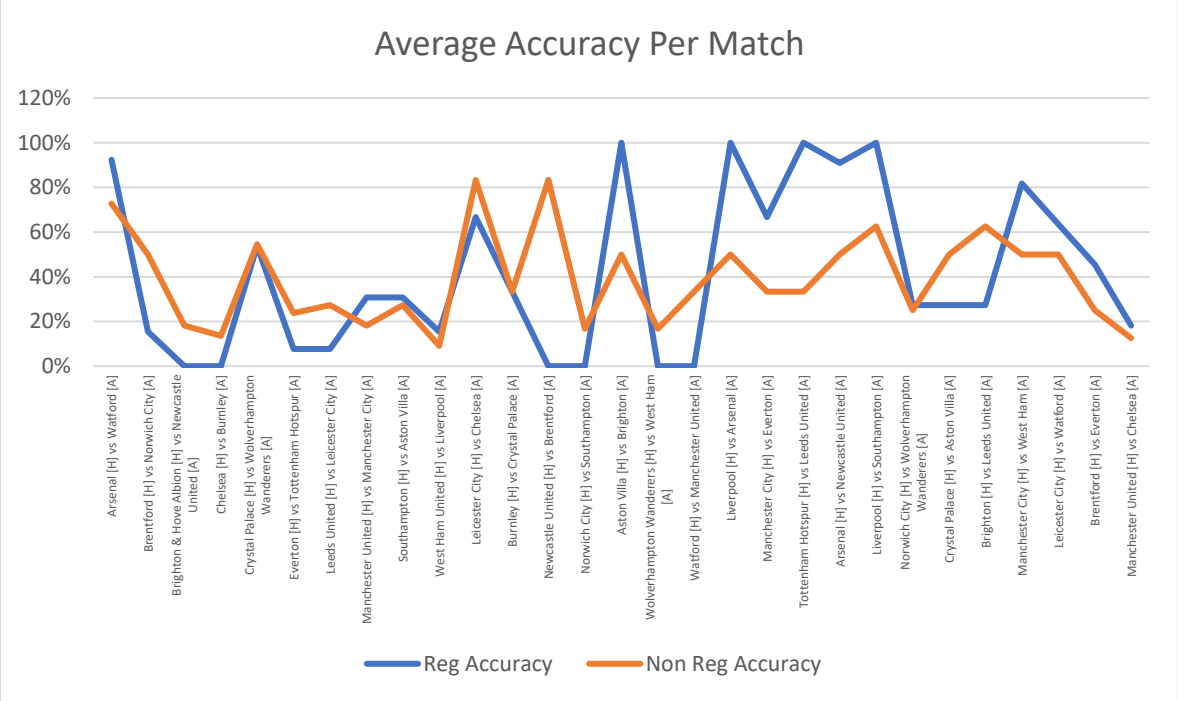


Figure 1 Avg Acc per match

From the above it can be seen that in general both regular and non-regular viewers of football primarily focus on the "bigger" team when selecting a winner. Although the regular viewer may also take other footballing factors into account. From the Brighton vs Manchester United and Chelsea vs Burnley games it can be seen that whereas some of the non-regular viewers predicted the result due to their unbiased opinions. No regular viewers were able to predict the result correctly. Due to the additional factors taken in however it can also be seen that when a regular viewer makes a prediction the accuracy of that prediction can have a large range due to the vast unpredictability of football as a sport.

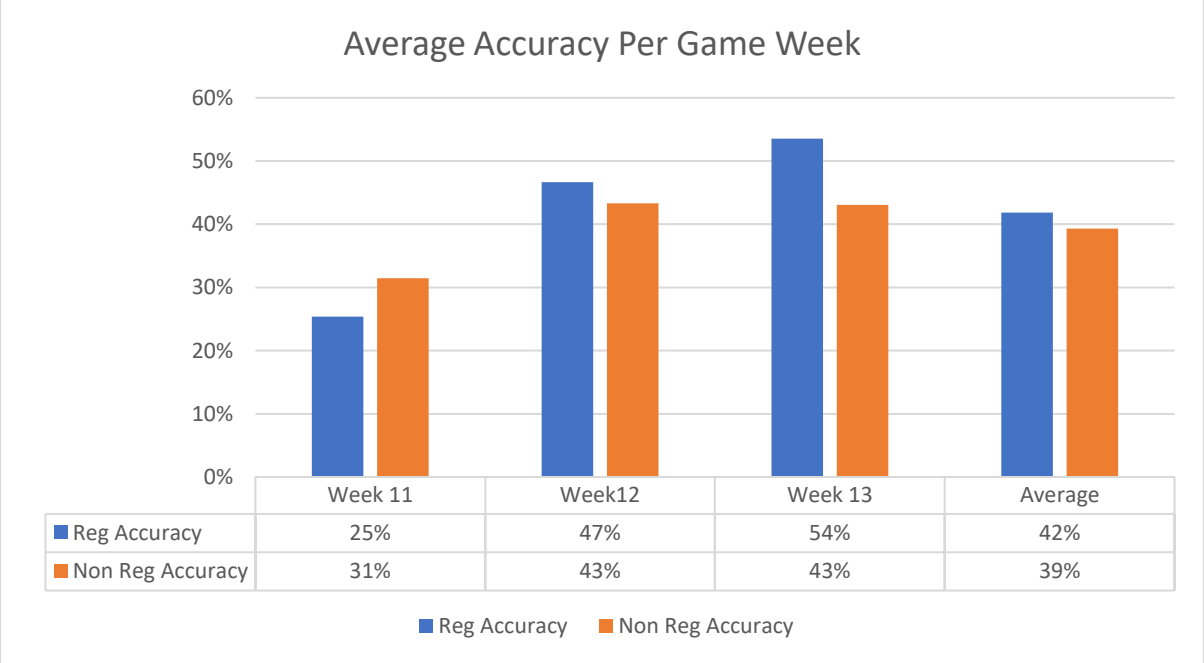


Figure 2 Average Accuracy Per Game Week

The result also show that regular viewers were generally more accurate at predicting the outcomes of games. Although the Regular Viewers accuracy was lower for week 11, this can be attributed to the large number of games ending with widely unexpected results. However, although there is a slightly better accuracy the difference is not significant.

1.3.2 Rating Systems

Rating systems are systems that allow comparisons of quantitative measures to determine the superior side. This can be for teams or players. Ratings are numerical representations of weighted strength of teams dependent on the features/stats used and can provide the basis for a prediction.

1.3.2.1 ELO

The ELO rating system is a mathematical equation used to work out the relative skill levels of two players. The original purpose was to rank two chess players when first introduced by Arpad Elo, a Hungarian-American physics professor in the 1960's. It has since been adapted to be used in a wide range of disciplines such as sports and e-sports (electronic sports). The World Football ELO ratings organisation makes uses of the formula to rank all international football sides [12].

ELO Formula:

- $R_n = R_O + P$
- Where: $P = KG(W - W_e)$
- And where:
    - R_n = The new team rating
    - R_O = The old team rating
    - K = Weight index regarding the importance of the match
    - G = A number from the index of goal differences
    - W = The result of the match
    - W_e = The expected result
    - P = Points change (Rounded to the nearest integer before updating the team rating)

1.3.2.2 Power Ranking

Power ranking is where statistics are considered and assigned point values which are total to give a final power ranking. This can then be used to compare two teams, with the greater value being better. The stats used and points allocated per stat is up to the creators of the model to determine. It will often have a weighting to reflect its importance or impact. A well-documented example of power rankings is used by sky sports where they assign players points based on how they perform each week [13]. This same methodology can be applied to teams.

1.3.2.3 Goal Superiority

Goal superiority is a simplistic measure of team dominance. It takes the goal difference (goals scored minus goals conceded of a given team) over the course of a few matches and compares this with the other team to determine a match rating. Football-data provide an example of this where the teams rating is taken from the last 6 games. Tottenham have a rating of -3 and so do Leeds, making the match rating 0. With enough historical data this can then be transformed into a prediction via method such a Poisson distribution to work out the probable outcome [14].

1.3.3 Versions of Expected Goals (xG) models

Expected goals is a metric which determines the quality of a chance by calculating the probability that the chance created will be scored from that position on the pitch during that stage in the

match. The expected goals value at the end of the game is a cumulative sum of these probabilities worked out for each respective team. There is different way of calculating xG, below are three of the most used:

1.3.3.1 Method 1

Official method provided via the analyst [15]

Expected goals takes the following variables into consideration when predicting the likeliness of scoring:

- Distance from the goal
- Angle to the goal
- One-on-one
- Big chance
- Body part
- Type of assist
- Pattern of play

All these factors are recognised and are data inputs ran using logistic regression – 1 demonstrating that it is a guaranteed goal and 0 suggesting that it is impossible to score. The analyst use data from OPTA to provide the prior and historical observations for the chances and with that data use the statistical method to predict a data valued based from those observations.

Set pieces are given their own values and modelled independently as mentioned on TheAnalyst penalties have an overall conversion rate 0.79 xG and this methodology is applied to freekicks and headed chances from set pieces as opposed to open play.

Logistic Regression:

$$\Phi(z) = \frac{1}{1 + e^{-z}}$$

1.3.3.2 Method 2

Fantasy football fix has provided the equation for the simplest xG model. This model treats every shot equally rather than accounting for the quality of shot or the passage of play, hence their xG model is very simple as is at follows:

Shots * 0.1 [16]

The percentage of goals scored from a given shot in the premier league is 10%, the multiple for the xG is given 0.1

Since every shot is treated equally a shot from 40 yards and a shot from 6 yards are considered "equal" in this simplified xG model. If a team continued to take shots from 40 yards out, it is likely that their shots on goal would be inflated and if a team were to take 25 shots at goal their expected goal value would be 2.5. This would be unrepresentative of the game as if we took the probability of someone scoring from 40 yards it is less than 2% [17].

1.3.3.3 Method 3

Prosoccer.eu [18] uses a team's defensive recorded and offensive record as well as taking into account the league's averages for the goals conceded and scored to create a formula which produces expected goals. The example given is using La Liga Santander (Spanish Primera Division).

- Average goals scored in the league:

    - **(total scored goals) / (total number of matches)**
- Average goals scored for a team:
    - **(total scored goals for the team) / (total number of matches played by team)**
- Average goals conceded in the league:
    - **(total scored conceded) / (total number of matches)**
- Average goals conceded for a team:
    - **(total conceded goals for the team) / (total number of matches played by team)**
- The defensive rating:
    - **Average goals conceded for a team / Average goals conceded in the league**
- The offensive rating:
    - **Average goals scored for a team / Average goals scored in the league**
- For the home team the equation is:
    - **(Average league GF) * (Home team attack power at home) * (Away team defence power away)**
- For the away team the equation is:
    - **(Average league GA) * (Away team attack power away) * (Home team defence power at home)**

Despite providing numerical values for the teams expected Goals, this method fails to take into consideration individual players and how expected goals changes throughout a match's duration.

1.3.4 Data in Football

Football has become increasing data driven and has many examples such as, top level clubs now employing entire departments for data analytics to gauge performance or who to buy in the transfer market, pundits and football fans using them to discuss and analysis matches, and betting companies to create odds. This has resulted in a wide range of quantitative statistics being used for top level sport, high level stats such as goals scored per team, or possession, down to a lower granularity such as the angle of a shot that resulted in a goal, or the number and direction of passes completed. But does the availability to a greater range of statistics help improve the prediction of the outcome? And how can one stat be compared to another, and which ones are relevant and have an impact on the outcome?

Teams have different styles, take Stoke City versus Arsenal on 19th August 2017 [19]. Statistically the raw stats of the game are more likely to favour Arsenal on paper, however, despite Arsenal averaging 1.95 goals a game that season verse 0.92, Stoke won the game 1-0. This only considers one stat, but it shows how stats cannot be taken at face value and that stats cannot account for everything. There is an element of randomness and unpredictability, which is why achieving a highly accurate prediction is problematic. Raw stat adjustments are made and used to weight values to make stats between two teams more comparable rather than compare apples vs oranges, it becomes apples vs apples – a fairer comparison. Whilst quantitative stats can be manipulated to help improve the accuracy of models there will still be a factor of unpredictability that may not ever be possible to be truly accounted for. A way of further minimising this could be to use qualitative data.

Non-numeric data can have a significant influence on the trajectory of a match outcome. Some examples factors could be:

- The referee in charge of a match – they are human, and all have their own view on events within a game, for example, one referee may view a foul as a red card offense, whilst another may consider the same offense as only a yellow.

- Players personal life – if a player has personal troubles off the pitch, these issues could affect their performance levels, they may play worse than they would be expected to.
- The effect of weather on players – some player may adjust or not to differences in climates. This is more significant when traveling to different continents which have considerable variance in climates. Players who do not adjust well are likely to have lower performance levels, which could impact the ability of the team.
- Momentum and club atmosphere – teams that are in good form and have a positive mindset and feel-good factor around the club are likely to perform better than they would averagely and the same goes for the inverse.

The range of non-numeric data available is vast and tricky to determine and convert to a numeric value. Complex prediction models will use data that is gathered through bots or more sophisticated AI techniques to analysis the relevant information that can be found online, such as, from a player's twitter page or news reports to attempt to mitigate these factors.

The examples of data used includes team and player data. Including player data allows for greater analysis of individuals and their impact on the team. A basic instance of this is when a club plays a weaker team in a cup game to rest the strongest starting team for a tougher match – often to preserve fitness and energy – the players in the weaker line-up are unlikely to meet the standards set by the best starting line-up and the stats of the team will be affected, such as expected goals which would likely decrease. Finding the balance of a player's impact on the team will vary and change player by player which increases the difficulty in determining the weighting and assessment of these stats within the team.

Many of the above are good examples of stats used in industry to increase the level of accuracy, however, a lot of the more detailed breakdown of stats data is not publicly available and may limit the accuracy of the prediction. See section 3.1.2 Chosen Dataset – Football-Data for further discussion around the data and its availability.

1.3.5 Football Prediction Markets

Football is an incredibly popular sport with an average live audience of 517 million viewers tuning into watch the 2018 World Cup Final, with more than 1.1 billion people tuning in over the course of the game [20].

This case is by no means an outlier, and a steady increase in viewership has been identified since the first televised broadcast of the World Cup in 1954. This far outperforms the viewing numbers for any other major sporting event, even overtaking the Olympics since 2018 [21].

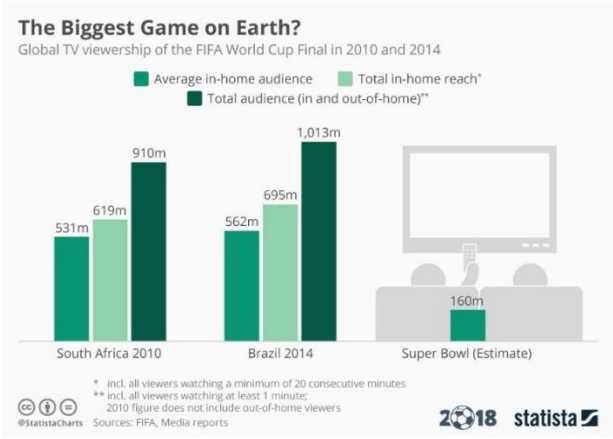


Figure 3 Biggest Game on Earth [60]

Additionally, view counts for football highlight videos posted onto social media sites such as YouTube often break into the millions within the first few days of upload, demonstrating that not only televised football matches can bring in an incredible number of viewers, but can obtain these viewing figures on a consistent basis.

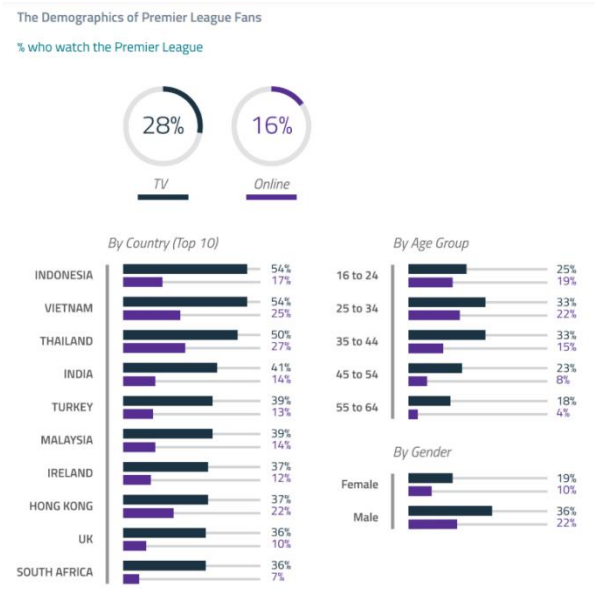


Figure 4 Demographics of the EPL [22]

As of 2021 the EPL is the top league in Europe [22] which likely plays a factor in its worldwide viewer ship and popularity. The EPL is equally popular across all age groups between the ages of 16 and 64, with 16–24-year-olds leading online viewing of EPL football matches. Additionally, twice as many males actively watch the EPL than females [23].

Partaking in sports betting requires individuals to be a minimum of 18 years old. This results in a potential customer base of over 4 million, as 58% of all males in the UK aged between 18 and 34 years old watch the EPL on a regularly. Furthermore, over half of all bets placed on sports in the UK are on predicting the outcome of EPL football matches [23], meaning that around 1.8 million people are repeatedly placing bets.

Examples of markets where football predictions are used:

Industry	Overview
Sports Betting	Football predictions are used within the sports betting industry to inform the odds that are produced by companies. These odds are often optimised to financially benefit the bookmaker but also give consumers an opportunity to financially benefit themselves.
Sports Punditry	Sports punditry is a media field that is diversifying more so now than ever before. The transition from mainstream media onto video-sharing platforms such as YouTube, TikTok and Instagram has meant that the content produced by sports broadcasters has had to undergo a drastic change to stay relevant. Predictions help to inform pundits when creating content for these platforms such as video debates, prediction-based shows, and in-depth conversations. This provides entertainment for consumers bringing in more viewers to both mainstream television channels and online media outlets. Also, sports-based discussion shows bring in hundreds of thousands of views daily, which creates a turnover of over $1,000,000 per month from YouTube alone [24].
Professional Sports Teams	Teams can use football results prediction algorithms to adjust how they prepare for upcoming matches to optimise their chance of winning. This could be changing their training regimen, altering the team line up or employing different in-game tactics. There is a huge number of data analytics roles being fulfilled in professional sports teams to create personalised matchups and other strategies for every match the team plays. The most successful clubs In the EPL routinely employ more than 15 people solely to fulfil data analytics roles [25]. This introduction of roles dedicated to the field of data analytics within a sports club demonstrates how integral this field is in the modern era of sports.
Football Fans	Almost every single football fan has attempted to predict the outcome of a match, either casually in conversation or by attempting to profit from their

	prediction by placing a bet with a bookmaker. Fans avidly partake in online fantasy football games, and 75% of players demonstrate an interest in the data analytics aspect of football to better make decisions about their line up for the upcoming games and improve their position within the leader boards.

There is a wide range of applications for predicting football match outcomes, with many aspects of people's lives being influenced by the field of sports data analysis and predictions. From entire jobs centred around successfully predicting events within football, to the way people spend their free time, sports data analytics is all around us.

2 The Data

2.1 Selecting a Dataset

2.1.1 Dataset Comparison

Category	Comments
Minimum required features	This is the minimum required features to be able to construct a reasonable model. They are as follows: • Home and away team • Full time result (win, loss, draw) • Home and away goals • Home and away total shots • Home and away cards (yellow and reds) • Home and away fouls conceded (or committed as this is just the inverse)
Size of the Data	The data set must include a reasonable number of season and every single game from each season. A single season will consist of 380 games (rows of data). The minimum requirement is all games from at least 5 consecutive seasons. This will provide at least 1900 rows of data and provide a good historical trend in relation to football.
CSV File	This determine if additional steps are required to access the data in R. R requires file types of csv, so if the data is not natively available in this state, then it will require additional work to get it in the correct format.
Extended Features	The data includes all the minimum required features and more. These features are more detailed stats about a game. This could include, the number of corners, set piece goals, possession and so on.
Match Events	The data includes stats based on time throughout a match. For example, it may have goal times, possession of the game at 10-minute intervals or other in game-based events.
Player Stats	The data includes player stats in relation to a game. Individual players and their personal stats within a game, such as goals scored by them, passes completed, number of tackles and so on.
Available for free	Is the data publicly available with no cost attached?

Data Set	Category Included						
	Minimum required features	Size of the Data	CSV File	Extended Features	Match Events	Player Stats	Available for free
SerpAPI [26]	✓	✓	✗	✓	✗	✓	✓
StatsPerform – Opta [27]	✓	✓	✗	✓	✓	✓	✗
Football-Data [28]	✓	✓	✓	✓	✗	✗	✓
Kaggle [29]	✓	✓	✗	✓	✓	✓	✓

Data Set	Size of Data	Advantages	Limitations
SerpAPI	Varying (Depending on query)	Wide range of data, containing extended features and player statistics. Gives information about a large variety of different sports, which could be beneficial for future development and out-of-scope targets.	Results provided in .json so will require additional work to collate information as .csv. Limited number of request can be made via the API by incurring costs.
StatsPerform – Opta	30,000+ matches covered, 1 billion + tracked data points, 3'900 global competitions covered	Provides by far the most in depth, range, and variety of data of all the data sets. Includes qualitative and quantitative data on players and teams.	Access to data requires payment. Large number of data items may extend cleaning and pre-processing data time required.
Football-Data	7980 x 45 (380 x 45 per file)	Includes most of the desired fields by default and contains the correct information to derive the remaining fields. Data is provided in a .csv format which is the desired format.	Contains many unnecessary fields that would need to be removed during data cleaning. No player or match stats available, so less in-depth compared to other data sets.
Kaggle [30]	25979 x 115	Includes many of the most important data points that will be used by our models. It contains +25,000 matches, and +10,000 players across 11 European competitions from 2008 to 2016. Includes team data such as line-ups and formations, which	More complex set up than other data sets. The data is in database format and links matches to match events and player and team performance-based statistics. This would increase the required pre-processing and data

		provides additional information that may impact outcomes.	cleaning portion of the project. Data only contains a short time span in relation to the other data sets (8 years).

2.1.2 Chosen Dataset – Football-Data

Opta data by far provides the most in-depth collection of data and is to a professional industry standard, but as it has a large cost it is not feasible to use for this project.

Kaggle provides an excellent range of data for a publicly available data set, containing in game statistic as well as player and team data. This would allow for complex stat analysis and construction. For example, it would be possible to work out xG by its official method (see section 1.3.3.1 Method 1) and calculate xG by game time due to the inclusion of match events. This information is beneficial as xG is likely to fluctuate during a game, such as when a team is 3-0 with 10 minutes left in the match their expected goals will decrease as they are no longer pushing to score with as much intensity. However, the setup is likely to increase the amount of time and resources required for set up and due to the short nature of the project other more simplistic data sets may be better suited for this project. It also has shortest range of historical data which could affect the ability to find long term trends in the data.

SerpAPI provided via Google use API requests to web scrape data, returning in a JSON format. This gives flexibility in only selecting the fields required saving a step in the data cleaning process, but the data is potentially unreliable. As it is web scraping it relies on what is available online and can result in incomplete data. In conjunction, the number of API request are limited (200 a month) before it incurs charges, making it not feasible within budget of the project.

Football-data provides each season in a csv format ready for quantitative analysis. This mean more emphasis can be placed on the pre-processing and model creation. Although, the data is not as rich as some of the other data sets in this list it does provide sufficient data to meet the requirements for the project. It also has just over 20 years' worth of data which will allow for analysis of trends over a large period of time. Moreover, it would be possible to web scrape the additional information in relation to players, teams, or match events, but this is unlikely to be as reliable as the data from Opta for example.

2.1.3 Dataset Overview

Name	Type	Comments
Division	Categorical, string	Level of the league. E0 – English Premier League E1 – English Championship and so on.
Date	Categorical, dateTime	Date of the game
Home Team	Categorical, string	Team playing at home
Away Team	Categorical, string	Team playing away from home
FTHG	Ordinal, integer	Full time home goals scored
FTAG	Ordinal, integer	Full time away goals scored
FTR	Categorical, string	Full time result of the game H = Home win A = Away win

		D = Draw
HTHG	Ordinal, integer	Half time home goals scored
HTAG	Ordinal, integer	Half time away goals scored
HTR	Categorical, string	Half Time Result H = Home win A = Away win D = Draw
Attendance	Ordinal, integer	Number of people who went to watch the game
Referee	Categorical, string	The referee in charge of the match
HS	Ordinal, integer	Home shots total
AS	Ordinal, integer	Away shots total
HST	Ordinal, integer	Home shots on target total
AST	Ordinal, integer	Away shots on target total
HHW	Ordinal, integer	Home team hit the woodwork total
AHW	Ordinal, integer	Away team hit the woodwork total
HC	Ordinal, integer	Home corners total
AC	Ordinal, integer	Away corners total
HF	Ordinal, integer	Home fouls committed total
AF	Ordinal, integer	Away fouls committed total
HO	Ordinal, integer	Home offsides total
AO	Ordinal, integer	Away offsides total
HY	Ordinal, integer	Home yellow cards total
AY	Ordinal, integer	Away yellow cards total
HR	Ordinal, integer	Home red cards total
AR	Ordinal, integer	Away red cards total
HBP	Ordinal, integer	Home booking points Yellow = 10 points Red = 25 points
ABP	Ordinal, integer	Away booking points Yellow = 10 points Red = 25 points
52 columns related to: “1X2 (match) betting odds data”	Ordinal, float	Betting odds-based column values
15 columns related to: “total goals betting odds”	Ordinal, float	Betting odds-based column values
22 columns related to: “Asian handicap betting odds”	Ordinal, float	Betting odds-based column values

Official key can be found here: https://www.football-data.co.uk/notes.txt

2.2 Handling the Data

2.2.1 Cleaning the Data

The following steps were taken to clean the data in readiness for pre-processing:

- All csv files combined into one data frame ordered by date and added additional column SZN to denote season.

- Checked for null values as some data was only available in select seasons. To fairly analysis across season they needed to have the same data. For example, attendance was removed as it was not available for every season.
- Removed columns that were not required see table below:

Name to Remove	Comments
Division	All teams are in the same division (division 1) and so this field provides no additional information and, if included, would add noise.
Referee	The referee provides a level of subjectivity that is out of scope for our project to consider currently. While it may be useful to include this field upon future development to inform trends where a particular referee may make an impact upon the result, given the time constraints of the project this is not something that our proposed solution will be considering at this time.
Attendance	Has null values in some seasons and will affected comparison.
All 52 columns related to: "1X2 (match) betting odds data"	The models that will be used are not concerned about data related to match betting odds, and not removing this data could blemish the important fields and make data processing a more complex task.
All 15 columns related to: "total goals betting odds"	The models that will be used are not concerned about data related to total goals betting odds, and not removing this data could blemish the important fields and make data processing a more complex task.
All 22 columns related to: "Asian handicap betting odds"	The models that will be used are not concerned about data related to Asian handicap betting odds, and not removing this data could blemish the important fields and make data processing a more complex task.

2.2.2 Pre-processing the Data

The following outlines the steps taken to pre-processing the data:

- Categorical fields were assigned numeric values and new features were constructed. Some of the calculated columns were required for calculation for final feature sets only, and hence removed when no longer required. The following fields that fall into these categories are as follows:

Name	Type	Comment	Calculation Column
HP	Integer	Home Points – this is the number of points a home team gains. Based on the FTR column, H = 3 points, A = 0 points, and D = 1 point.	Yes
AP	Integer	Same as HP but inverse.	Yes
FTR	Integer	Full time result. Three categories home, away, and drawn: D = 1, A = 2, and H = 3	No
TP (H & A) (Ovr, H, & A)	Integer	Total points were defined for both home and away teams and in terms of overall (sum of home and away), home points, and away points at a given date.	Yes
TGS (H & A) (Ovr, H, & A)	Integer	Total goals were defined for both home and away teams and in terms of overall (sum of home and	Yes

		away), home points, and away points at a given date.	
TGD (H & A) (Ovr, H, & A)	Integer	Total goal difference was defined for both home and away teams and in terms of overall (sum of home and away), home points, and away points at a given date.	Yes
TGC (H & A) (Ovr, H, & A)	Integer	Total goals conceded was defined for both home and away teams and in terms of overall (sum of home and away), home points, and away points at a given date.	Yes
POS (H & A)	Integer	Position in the league at a given date for a home and away team.	Yes
N_Home_Game	Integer	Number of home games home team has played	Yes
N_Away_Game	Integer	Number of away games an away team has played	Yes
GW (H & A)	Integer	Game week per home and away team. Due to fixture rescheduling game week is not always the same for two teams, hence requiring both an away and home game week field.	Yes
SZN	Integer	Season, the year in which the game occurred.	Yes
Game_Num	Integer	Unique game identifier.	Yes
TS (H & A)	Integer	Total shots of team up to the date of the game.	Yes
TST (H & A)	Integer	Total shots on target for a team up to the date of the game.	Yes
GS_LG_Avg (Ovr, H, & A)	Float	Goals scored league average at the date of the game.	Yes
GC_LG_Avg (Ovr, H, & A)	Float	Goals conceded league average at the date of the game.	Yes
GD_LG_Avg (Ovr, H, & A)	Float	Goals difference league average at the date of the game.	Yes
FRM_VO (H & A)	Integer	Form verses the game opposition. This was calculated using match rating by collating information of past fixtures that were between these two teams.	Yes
ovr_form (H & A)	Integer	Overall form verses the opposition. This uses match rating based of off a team's last 5 games.	Yes
home_form & away_form	Integer	Home and away specific form. For example, the home teams home form will be the match rating of the teams last 5 home games.	Yes
PD (H & A)	Integer	Points difference. This is the difference in points at that date between the two teams.	Yes
POSD (H & A)	Integer	Position difference. This is the difference in position in the league table at that date between the two teams.	Yes
P90 (H & A) (Over, H, & A)	Float	Points per 90 minutes. This is the average number of points a team has achieved. It is also split overall points, home points and away points. The points total here will be the total for each respective value up to the date of the game.	Yes
GS90 (H & A) (Over, H, & A)	Float	Goals scored per 90 minutes. This is the average number of goals scored by a team. It is also split	Yes

		overall goals scored, home goals scored, and away goals scored. The goals scored total here will be the total for each respective value up to the date of the game.	
GC90 (H & A) (Over, H, & A)	Float	Goals conceded per 90 minutes. This is the average number of goals conceded by a team. It is also split overall goals conceded, home goals conceded, and away goals conceded. The goals conceded total here will be the total for each respective value up to the date of the game.	Yes
ATT (H & A) (Over, H, & A)	Float	Attacking rating. This is rating of a team's attack, calculate as outlined in section 1.3.3.1 Method 3.	Yes
DEF (H & A) (Over, H, & A)	Float	Defensive rating. This is rating of a team's attack, calculate as outlined in section 1.3.3.1 Method 3.	Yes
xG (H & A) (Over, H, & A)	Float	Expected goals. This is calculated as outlined in section 1.3.3.1 Method 3 and provides a value for the number of goals a team should be expected to score.	Yes
xGC (H & A) (Over, H, & A)	Float	Expected conceded goals. This is calculated as outlined in section 1.3.3.1 Method 3 and provides a value for the number of goals a team should be expected to concede.	Yes
xGNS (H & A) (Over, H, & A)	Float	Expected goals non-shot-based stats. This determines the effect of non-shot-based stats on the number of goals expected to be scored by a team.	Yes
ELO (H & A)	Float	ELO rating. This is the ELO score of a given team, accounting for weighting of xG values, and strength of opposition**.** See section 1.3.2.1 ELO and 4.1.1 ELO for more information on how this is derived.	No
OR (H & R)	Float	Offensive rating is a team's strength of offense home, or away specifically. Calculated using the ELO formula.	Yes
DR (H & R)	Float	Defensive rating is a team's strength of defence home, or away specifically. Calculated using the ELO formula.	Yes

(H & A) denotes home and away values. (Ovr, H, & A) denotes overall (home + away), home, and away

- The information provided in the dataset included game related data only. There were no stats relating to how teams in each game were doing relative to each other outside of the match itself. Home and away values are also explored individually, and this is because some teams perform better at home than away or vice versa. By including league table values and form of teams this accounted for this. Note this is explored further in later sections.
    - League Table: a league table was constructed by date groups. So, for a given season a vector of all rows with the same date group would be selected. This would then be used to update the league table and then the values of the columns for the main data frame were the features are later selected from.

By selecting a single date group at a time this meant for each game the value, such as position of each team, were identical to what they would have been in that moment of time.
        - League tables were constructed as overall (combined home and away), and home and away tables only – these consisted of home and away game results respectively. This information is later used to weight various formulas based on a team's specific home and away strength.
    - Form:
        - Form is calculated as match rating seen in section 1. This is a simple, but effective measure to gauge a team's current progress. This is split into overall, home, and away. Similarly, to league home and away values, these are used to determine a team's strength at home or away.
    - Form versus opposition:
        - Form verses a particular opposition is calculated. This is to portray a team's ability against a particular opponent, as some teams have 'bogey' teams that they historical do not perform well against. This is perhaps beginning to lean into quantifying qualitative data as this is based on psychology around a game.
- Adjustments:
    - Not all teams are equal and play in the same way. Some may play a possession-based game, others may play long ball. There are numerous versions of different tactics and styles. Stats are adjusted to make two teams more comparable, rather than compare apples vs oranges, it becomes apples vs apples – a fair comparison. Take goals scored, should Chelsea – the European Champions – scoring a goal against Burnley – a relegation threatened team – verse a goal against Manchester City – reigning EPL champions – have the same value? Manchester City would be considered a tougher opponent and have a stronger defence than Burnley, this can be quickly seen as Burnley on average so far this season have conceded 1.17 more goals per game than Manchester City. It is more difficult for Chelsea to score against Manchester City than Burnley and this can be reflected via the weighting. Weighting and adjustments can allow for better comparisons than taking face value but is still not guaranteed. Take Burnley's 1-1 draw with Chelsea away from home, Chelsea was favoured, but Burnley dug in and held on for a point. So, quantitative stats can be manipulated to help improve the accuracy of models but there will still be a factor of unpredictability that can never truly be accounted for. A way of further minimising this could be to use qualitative data.
- Z-Scale:
    - Z-scale (standard score) is used to calculate the probability of stats occurring within the normal distribution. This accounts for the league average allows fairer more accurate comparisons of data between teams.
- Weighting xG and NSxG:
    - xG and NSxG will be weighted in relation to goals scored. This is weighted to adjust the influence on each one in relation to the actual value. The formula used is as follows and more details can be seen section 1.3.3.1 Method 3:

$$\mathrm{HG} = \mathrm{pxG} + (1\text{-}\mathrm{p})\mathrm{NSxG}\ /2$$

derives to: $\mathrm{P} = 2\mathrm{HG} - \mathrm{NS}\ /\ \mathrm{xG}\ \text{-}\ \mathrm{NSxG}$

where:

- HG: home goals
- xG: Expected goals
- NSxG: Non-shot-based expected goals
- p: weight

- Weighting ELO:
  - The ELO rating will also be weighted. This is to account for a team's specific strength at home or away. This is important as some teams are good when at home and not as good when playing away and vice versa. By weighting the ELO value this can be accounted for.

    The weighted formula is as follows. To see how these fits into the ELO rating formula see section 4.1.1 ELO:

$$(HOR\_H[k] - ADR\_A[k]) + (HDR\_H[k] - AOR\_A[k])$$

    Where:
    - HOR: home offensive rating
    - ADR: away defensive rating
    - HDR: home defensive rating
    - AOR: away offensive rating
- Removing first season:
  - The first season is removed as the first season will have zero values for form between two previous teams as there in no data available before this point.
- Removing first 5 games:
  - The first 5 games for each team for a giving season are removed. This is because the form calculation works of off the previous 5 games, and therefore cannot allow for a fair comparison, and every team first game will also have a zero value.

2.3 Data Analysis

2.3.1 Data Exploration & Explanation

After cleaning the data an initial assessment was done. Firstly, the following values, which may have been useful to explore, but due to zero values amongst various seasons was removed. This is because there was no way to provide a fair value that would make a useful comparison, this included values such as, offsides, attendance, bookings (red and yellow cards), corners, and fouls for and against. The trade off here was with the size of the data, over half of the data set did not contain a full set of values for these fields and having long term trends of historical data was considered to be more important, especially given there was opportunity to derive further fields.

Using domain knowledge and research the next sensible step was to calculate points, goals scored, conceded, and difference as well as the position of a team at a given

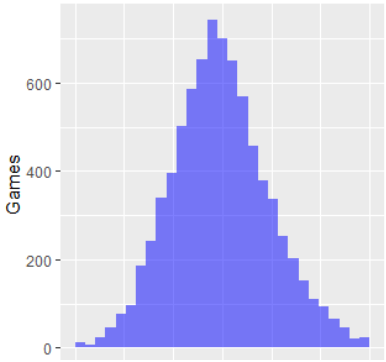


Figure 6 Overall Home team form

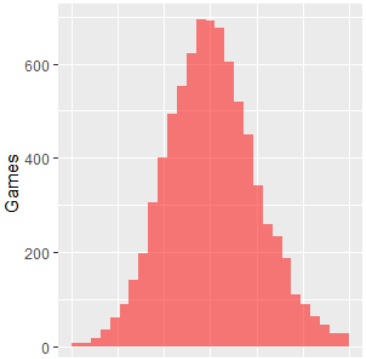


Figure 5 Overall Away Team Form

game. Essentially creating a league table to what it would have been at the point in time the game in the data was due to take place. This allows for analysis of a team's performance overall in a league, and how it can affect a game. For example, you would expect the team higher in the league to beat the team lower down in general, although this is not guaranteed. You can also see how form plays a factor. Form was calculated as match rating (goal difference of each team from their last 5 games added together) and you can see in figures 5 and 6 the distribution of form across games. This is as expected as the greatest number of games are at the median value which would make up most of the teams in the league, as only the stronger teams are likely to earn higher scores as this means that they are on a good run of form (scoring goals and winning games consecutively) and weaker teams on the opposite end of the scale.

The next step was to plot a scatter matrix which can be seen in the appendix to get guidance were trends may lie in the data from a quick overview. This then informed the direction that was taken with further manipulation of the data to generate more features. At this point based on research from section 1 the focus shifted to the relationship between goals and expected goals. Figure 7 and figure 8 show there is a fairly strong correlation between expected goals and number of goals scored. This is to be expected. However, there is a widespread in the data which is due to the way xG has been calculated. Firstly, the number of goals can never be guaranteed and is an estimate, and secondly the formula used is not the most accurate calculation. So, with a more in-depth data set you would expect to see a smaller spread of results.

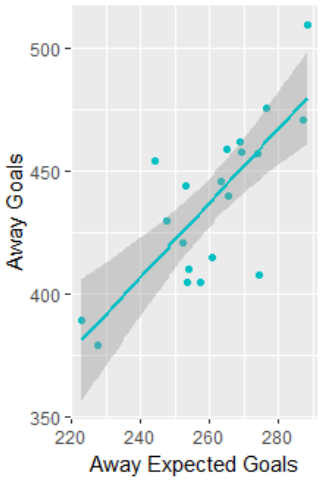


Figure 8 xG vs Actual (Away)

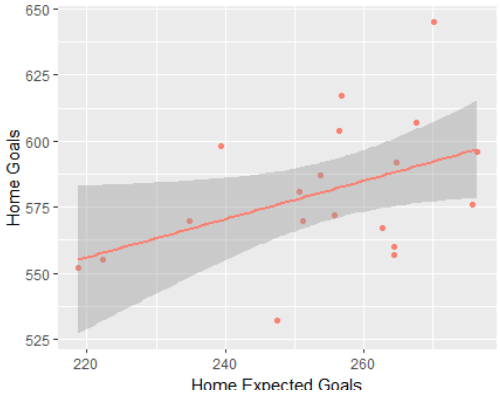


Figure 7 xG vs Actual (Home)

NSxG's feature selection is used to show the relation between goals and non-shot-based stats and account for these. Through domain knowledge it can be determined that non-shot-based stats will have an influence. For example, a team that has a lot of offsides suggest that perhaps they are a poor attacking team as they are caught out by the offside trap of the defensive team, who would win a free kick in this action, and hence the attacking team would lose the opportunity to score on that attack. But how much do these stats influence the goals scored? To decide on the features used initially plots were done for various feature to identify trends as well as referring to the scatter matrix. An example of this is in figure 9 you can see a shallow correlation between home goals and total away corners in a game. Whilst this seems strange it does make sense. When an attacking team has a corner, they will have players committed forwards leaving them susceptible to the counterattack. This graph shows that expected goals may have a slight

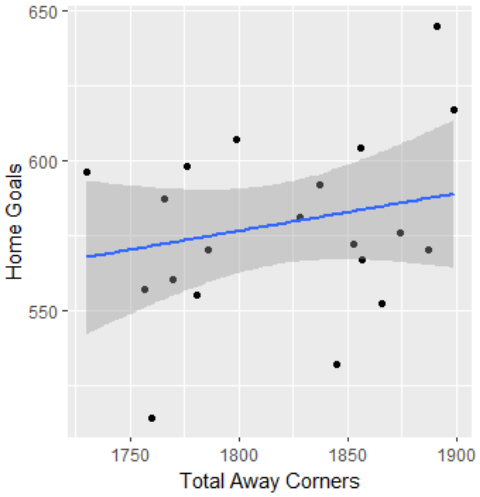


Figure 9 Home Goals vs Away Corners

increase when a team is defending a corner. Moreover, non-shot-based are unlikely to have a strong correlation due to not being directly related in football, however some will still have an influence and it is these ones that will be used in the weighting for NSxG. The selected features were, full time result, goals per 90, corner count, fouls conceded, team form.

Furthermore, qualitative data converted to numeric and player data would have been good and likely further improved the accuracy. Access to data such as referees for a given game would allow for accounting for referee bias against particular teams.

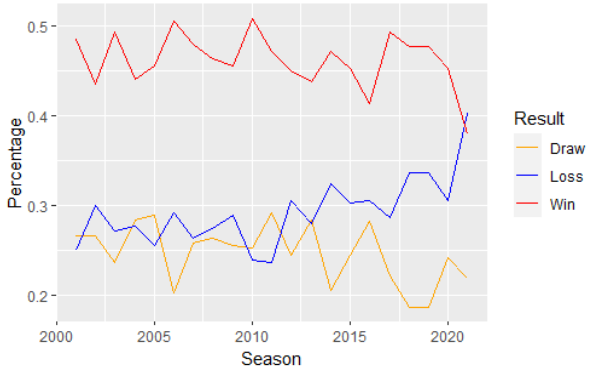


Figure 11 Percentages of Win, Loss, & Draws

Teams are more likely to be stronger at home. This is clearly portrayed by ~49% of games resulting in a home win. Figure 11 emphasises this with only a single season having away wins above home, which was during the pandemic. This shows the effect and influence fans have on a game, as the one time they are not in the stadium teams in general performed better away from home. However, some teams are stronger away from home outside of the 2021 season which may be down to their play style, such as being a counter attacking team. Therefore, to account for a teams' strength an offensive and defensive rating feature was created, which forms the basis of the weighting for the final ELO score of each team respectively.

Figure 10 Number of goals home and away per season

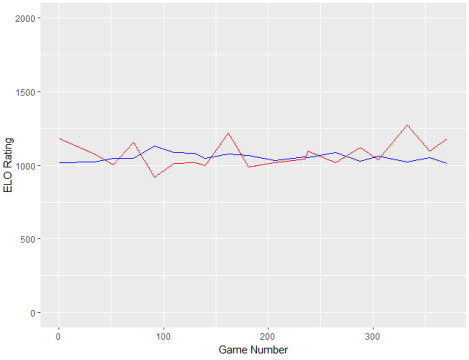


Figure 12 ELO ratings for Charlton vs Oppo

The following figure, figure 12 shows the ELO score of Charlton (red) vs their opponents for the 2001 season (blue). Charlton finished in 9th place (mid-table) and it is visible in the graph that the lines are crossing between each other but staying fairly close together, never going exceedingly high or low and each being above the other about half the time. This fits a mid-table team as they are likely to win as many as they lose. You would expect to see the red line higher than the blue for most the season for a team that finishes nearer the summit of the league as they would be expected to win most of their games.

3 Models

3.1 Algorithm Choices

3.1.1 Elo

In section 1 different ratings were discussed to rank teams. This project will make use of ELO ratings that rate teams using expected goals. Often, wins or points are used in the ELO calculation, but by using xG it provides a more detailed account of a game, since a team that loses a game should not necessarily lose points or too many points. This is because the team may be

considerably weaker than the opponent and say only lose 1-0, and this team should not lose as many points as if you take the same game and the team loses 4-0. However, ELO is considered a simple ranking method compared to others but given the time constraints and domain knowledge of business analytics this made sense for a project of this size.

The final ELO calculation is as follows:

- Home (HOR) and away (AOR) offensive ratings:

  OR = POR + (FTG - (xG + (xGNS/2)))

  Where:

  - OR = offensive rating of a team
  - POR = previous offensive rating of a team
  - FTG = full time goal count
  - xG = expected goals for a team
  - xGNS = expected goals non-shot-based for a team
- Home (HDR) and away (ADR) defensive ratings:

  DR = PDR + (FTGC - (xGC + (xGCNS/2)))

  Where:

  - DR = defensive rating of a team
  - PDR = previous defensive rating of a team
  - FTGC = full time goal conceded count
  - xGC = expected goals conceded for a team
  - xGCNS = expected goals conceded non-shot-based for a team
- ELO home rating and ELO away rating:
  - Weight is used to determine a team's strength this is calculated via the following:

    Weight = MP * ((HOR - ADR) + (HDR - AOR))

    Above is calculated from home teams' perspective, away is just the opposite

    Where:

    MP = match importance. This is set to 50. International matches are given the value of 60 by ELO ratings.net [31].

  ELO_H = ELO_H + (FTHG - Weight_H * (xG_OVR_H + (xGNS_OVR_H/2)))

  ELO_A = ELO_A + (FTHG - Weight_A * (xG_OVR_A + (xGNS_OVR_A/2)))

  Where:

  H denotes home team and A denotes away team

  - ELO = is the ELO rating of a given team
  - FTHG /FTAG = full time goals by home team/away team
  - Weight = weight of team at home or away and match importance
  - xG_OVR = overall expected goals (team average per game)
  - xGNS_OVR = overall non-shot-based expected goals (team average per game)

3.1.2 xG & xGNS

Expected goals (xG) is calculated via the method in section 1.3.3.1 Method 3. The data available in the data set does not allow for a more complex xG calculation and choosing the data set (Kaggle) which would allow this would take the schedule of the project beyond the deadline and is hence

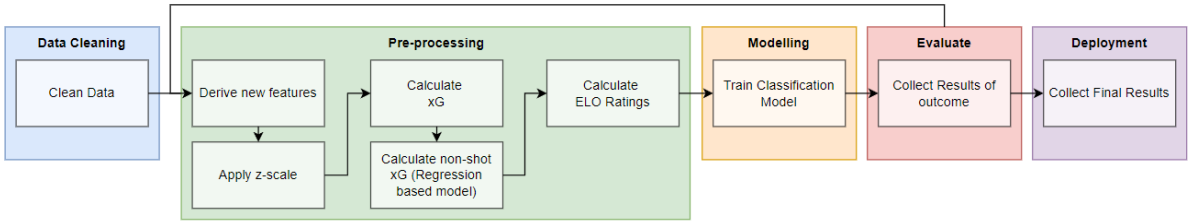
not feasible. Moreover, this method is more optimal than simply multiplying total shots by 0.1 as it accounts for a team's performance more accurately.

Expected goals non-shot-based is used to take into consider other stats that are not to do with shooting. The purpose behind this is the idea that other stats will have an influence on xG. For example, if a team has more possession of the ball, or more players, they are more likely to score in most case, so their xG would increase. Therefore, to calculate this value the following was done:

- A set of features was decided upon (outlined in the data section of the document).
- These features were fed into a regression-based model (KNN, NN, and RF).
- The objective is actual goals scored. This means the models will produce a value for xGNS which will show how the non-shot-based stats impact goals scored.
- Once the values were obtained, they were fed back into the pre-processing script to continue the processing the data.

3.2 Design Flow

The following is an overview of the main aspects of the design flow:



3.3 Model Choices

3.3.1 Model Comparison

Comparison matrix of the 5 proposed models, colour represents the strength for the following values.

Key:

Score (out of 5)	Colour
1	
2	
3	
4	
5	

Comparison Matrix of Model Choices:

Model	Well supported in R	Ease of implementation	Compatible with dataset	Suitable for project scenario	Typically produces strong results
SVM					
KNN					
MLR					
NN					

RF					

Based on the matrix above the project will compare three different ML models. This should provide a good comparison of different ML based techniques. The chosen models were deemed feasible in the time available and best suited as options for this project. The chosen models to be implemented are KNN, Random Forest and a Neural Network. See sections 5.4.1.1 KNN Design, 5.4.2.1 Random Forest Design, and 5.4.3.1 Neural Network Design for overview about how each of these models work.

3.4 Models

3.4.1 KNN

3.4.1.1 KNN Design

KNN is a supervised learning algorithm that it used in this project to classify and label data into a given category. It can also be used for regression-based tasks by using feature similarity to predict the values of the unseen data, this is based on how similar the points are within a set of data. KNN is a lazy learning method that means the target is based off the labels found in the training set. Therefore, approximations are susceptible to fitting to the structure found in the current dataset.

Mattheus Kempa conducted a similar project using a KNN model on Brazilian League Data [32]. Kempa's model was used on the data randomly with a train test split of 70:30, and he was able to successfully predict 45.65% of the matches.

Euclidean Distance is used to work out the similarity between feature vectors and where the points will be classed based on the training dataset. The new datapoints are given a value based on the nearest datapoints around them. The classifier suits the problem of predicting win, loss, draw as these can be categorised into three labels. There is the potential for label imbalance here as on average 49% of Premier League matches end in home wins, meaning 49% of the labels will result in home wins.

Euclidean Distance:

$$\sqrt{\sum_{i=1}^{n}(xi - yi)^2}$$

KNN regression was used as part of pre-processing to produce results for non-shot-based expected goals (xGNS). xGNS are determined from non-shot-based features referenced in section 3.1.2 xG & xGNS. This value is then used in further pre-processing calculations outlined in earlier sections to ultimately inform the ELO rating, which the model will use to predict. A higher K value leads to a reduction in the effect of noise within the data, this results in predictions becoming underfitted. This is less useful when predicting against to our training set, to avoid this we used the square root method – take the square root of the number of samples in the training set.

3.4.1.2 KNN Implementation

During the pre-processing stage KNN regression is used to provide the NSxG values from the non-shot xG features. The xGNS is calculated for the whole dataset the partition value for p is 1. The feature set contains all values for FTHG, and train set handles all the values found in the feature set (non-shot-based metrics) besides FTHG. For the K value the square root method is utilised to

```
#getting predicted values for when k = sqrt of the training dataset
pred_values <- knn(train=trained_data, test=test_data, cl=train_labels, k=7)

#confusion matrix
confusionMatrix(table(pred_values ,test_labels))
```

```
knn_reg <- function(dataset, hometeam) {
  set.seed(length(dataset))

  if (hometeam == TRUE)
  { indexes = createDataPartition(dataset$FTHG, p = 1.00, list = F)
    train = dataset[indexes,]

    train_x = train[, -1] #train the data excludes the column
    train_y = train[, 1] #train the data for the FTHG
    tsamples_count <- NROW(train_x)
    kroot_training <- sqrt(tsamples_count) #square root number of observations

    knnmodel = knnreg(train_x, train_y, k = kroot_training) #runs the knnreg method to train and test
    pred_y = predict(knnmodel, data.frame(train_x)) # a list of the expected goals

    output <- data.frame(matrix(unlist(pred_y), nrow = length(pred_y), byrow = TRUE)) #output is a dat
```

give the square root of all observations. When the regression is run the results provide a prediction for the xGNS. These values are collated for the entire data set and added to the data set for continuation of the pre-processing as discussed in section 3.1.2 xG & xGNS.

When predicting the outcome of a match a classifier is used, this is because the outcome of the games is categorised discretely as follows: Home win, Draw and Away win. KNN requires normalisation as shown in the figure below. This is necessary to avoid the features dominating and influencing the algorithm when classifying the FTR (full time result), as it uses the Euclidean distance between points within their feature sets.

```
normalize <- function(x) {
  return((x - min(x)) / (max(x) - min(x))) }
```

```
#randomly splitting data into test and training data with 80:20 split
sample <- sample(1:nrow(normalised_dataset),size=nrow(normalised_dataset)*0.8,replace = FALSE) #random selection of training data 80%
trained_data <- normalised_dataset[sample,]
test_data <- normalised_dataset[-sample,]
```

Applying the hold-out method, 80% of the original data is held for the training portion of the method and the rest is used for the testing section. The labels from the data are obtained from both sections of the data. This is done to evaluate how effective the KNN algorithm is on predicting the outcome/results of the matches.

Based off the science direct article [33], it demonstrates the general notion of using the square root method is an agreed principle/standard value for K when testing large datasets. In this case the k value is the square root of the number of training sample observations. After running the KNN method with the parameters of the training set, testing set, the classifications of the training set (the training labels) and the number of neighbours in this case square root (number of training sample observations). This is then applied to the unseen data to see how the model performs.

To identify the effectiveness of the KNN algorithm, the confusion matrix method is used on a table of predicted values in conjunction to the test labels to compare how well and accurate the algorithm was.

Predicted values	Actual values		
	0	1	3
0	132	19	43
1	35	192	68
3	220	137	470

The square root method despite providing a result of 0.6033 with respect to accuracy there may be other k values that may provide a better result – to test this hypothesis a for loop was added at the end of the code to loop through the values from 1 to sqrt(number in the training set) to see how a small k and a significantly larger k may influence the outcome of the KNN algorithm.

Once identifying the optimal K, the results of recall for each class the pattern remained the same, away wins still fall short against Home wins and draws but the tightness in accuracy is much less spread out amongst the other classes.

	Home Win	Draw	Away Win
K Value			
Square root number of observation	80.9	55.17	34.11
7	72.81	60.06	54.01

```
# plotting the accuracy of different k values  showing the sqrt of each for a clearer image
i=1
k_plot=1
limit = sqrt(NROW(trained_data))
for (i in 1:limit){
  knn_output <-  knn(train=trained_data, test=test_data, cl=train_labels, k=i)
  k_plot[i] <- 100 * sum(test_labels == knn_output)/NROW(test_labels)
  k=i
  cat(k,'=',k_plot[i],'\n')       # print the accuracy percentage
}
plot(k_plot, type="b", xlab="K-Value",ylab="Accuracy")
```

3.4.2 Random Forest

3.4.2.1 Random Forest Design

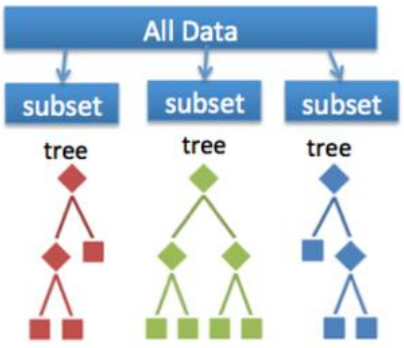


Figure 13 Random Forest Diagram [63]

Random Forest is a flexible machine learning algorithm which can be used for classification and regression tasks. Random Forest works by building a network of decision trees that are trained in conjunction with one another as an ensemble to increase the overall result. The entire provided dataset is split into subsets with each subset being passed through an individual decision tree using a random subset of features. Each tree then makes a prediction based upon the input data and the total network of trees then vote on what they believe to be the most likely outcome. This vote acts as the overall prediction made by the Random Forest model. Due to the large number of uncorrelated trees operating as a committee, this model will generally outperform many of the individual constituent models. This occurs because of the trees protecting each other from their individual errors.

While there have been many forays into how best to utilise the flexibility of Random Forest models for prediction problems, a study conducted by Nicholas Utikal has relevance to the project as it shares similarities. The main comparison between Project BAWKOS and Utiakl's study is the implementation of a Random Forest to predict home and away goals scored by each team in a matchup, to determine whether a Home Win, Draw or Away Win occurs. Utikal was able to achieve an average prediction accuracy of 51% [34] when using a Random Forest model and has provided inspiration and motivation for enabling this facet of the project to be a success.

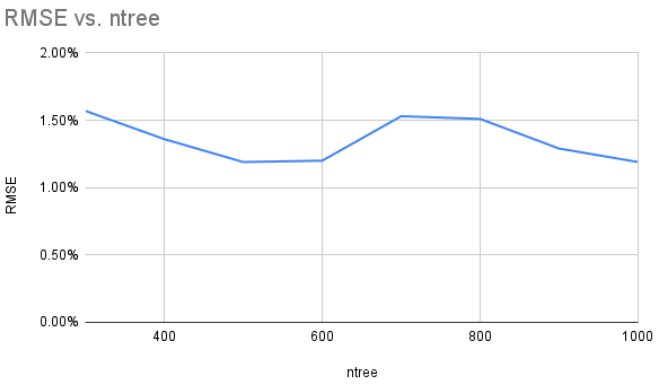


Figure 14 RMSE vs nTree

A Random Forest model has been used in project BAWKOS by taking an input of the home and away team datasets in two unique trees, which results in two unique forests being created. All fields will be used to inform the final classifier field of 'home_xG_ns' and 'away_xG_ns' at each decision tree split where the model will assess the most likely number of goals based on non-shot-based metrics. The result outputted from these two Random Forests will be used to inform the ELO rating for each team in a matchup. This will be analysed and interpreted to settle upon an outcome of a match being either a Home Win, Draw or Away Win.

Experimentation will be conducted upon a variety of Random Forest variables such as the number of variables sampled as candidates at each split (*mtry*), the number of trees to grow (*ntree*) or whether to include proximity. This experimentation will be conducted until an optimal fit for the model has been found, and the Root Mean Square Error (RMSE) and Out-of-Bag (OOB) error rates have been reduced to a reasonable degree.

The experimentation was conducted with values for *ntree* ranging from 300 to 1000. A definitive result was settled upon whereupon the optimal number of trees was around 500 as it resulted in the lowest RMSE error rate while still obtaining the same degree of success as Random Forest models which incorporated more decision trees. It is considered good practice to set *ntree* to an odd number because, in the case that there is an exact 50/50 split in the voting, there will be one final tree that will be able to make a majority decision. Due to this factor, *ntree* was set to 501.

The optimal value for the input variable *mtry* was also identified thanks to the checks that occur when utilising k-fold cross validation on the dataset. The following table was produced while fine tuning the model, which gave evidence that setting *mtry* to 2 was the optimum decision as it resulted in the lowest value for RMSE, a more influential signifier of error for Random Forests than RSquared and MAE.

mtry	RMSE	RSquared	MAE
2	1.08	0.097	0.84
5	1.09	0.093	0.85
6	1.09	0.093	0.85

3.4.2.2 Random Forest Implementation

The following code snippets contain the key lines of code that generate, manipulate, and optimise the Random Forest models:

- The first integral line seen when creating the Random Forest model is 'set.seed(15)' which sets the seed of the pseudo-random so that the same result can be reproduced.

```
set.seed(15)
```

- The line next important line is responsible for calibrating the k-Fold settings within the 'trainControl()' function The 'method' parameter is set to be equal to 'cv' which signifies that the training will be executed using a cross-validation approach. The 'number' has been set to 5 so that the program operates using 5 folds. This number was decided upon as it enforces an 80:20 split between training and testing data, while also balancing a high degree of prediction accuracy with excellent time-efficiency.

```
train_ctrl <- trainControl(method = "cv", number = 5, savePredictions=TRUE)
```

- The following line utilises the 'train()' function which is used to determine the method that is used. The first parameter has 'FTHG~.' passed to it to indicate that the training phase will be training the model to inform an expected value for Full Time Home Goals based upon the values found in all subsequent fields in the training dataset. The 'method' parameter is set to 'rf' to signify that the program wishes to use a Random Forest model and the 'trControl' parameter will be set to equal 'train_ctrl' which specifies that the model will be using the k-Fold approach described in the 'train_ctrl' variable. All these variables combine to train the Random Forest regression model using a 5-fold cross validation approach to finding FTHG using a Random Model approach.

```
rf_regression<- train(FTHG~., data = train, method = "rf", trControl=train_ctrl)
```

- This line computes the actual prediction based upon the training information inputted into the 'rf_regression' variable from the previous line, and the training dataset provided initially. This prediction is then stored in a dataframe called 'p1' to signify the first prediction. This is a generic multi-purpose prediction method provided by R.

```
p1<-predict(rf_regression, train)
```

- The following lines produce the console output that can be seen below. The code here demonstrates a comparison between six random predicted values for the expected goals and the corresponding actual number of goals that was scored. To ascertain an integer value for the number of expected goals, the prediction made and stored in 'p1' will be rounded up or down to the nearest integer. As can be seen from figure 5, an accuracy of 4/6 matches had the number of goals scored by the home team correctly predicted.

```
print("Home Training dataset prediction")
print(head(p1))
print("Actual home training data info")
print(head(train$FTHG))
```

```
[1] "Home Training dataset prediction"
       30      5783      6798      5913      5014      4773
1.1867667 0.8491822 0.9421393 2.2204988 1.1951056 0.2829105
[1] "Actual home training data info"
[1] 1 0 1 3 1 0
```

- Similar to the previous use of the 'predict' function, the Random Forest regression model will compute a prediction for the number of goals scored by the team using the testing dataset which, at this point, is unseen data. One small difference between using the

'predict' function on testing data compared to training data is that the type is required to be specified. In this case the type is 'raw' data.

```
p2<-predict(rf_regression, test, type="raw")
```

- Based off the prediction made and stored in the dataframe 'p2', the following console output is produced below. Again, the code demonstrates a comparison between six randomly selected predicted values for the expected goals and the corresponding actual number of goals that was scored. In this run of the program, it can be seen below that the prediction was correct 4/6 times.

```
print("Away Testing dataset prediction")
print(head(p2))
print("Actual away testing data info")
print(head(test$FTAG))
```

```
[1] "Away Testing dataset prediction"
      788      2885      6855      5557      2004      4739
1.3078989 1.9263156 0.9731711 1.3654170 0.8179085 1.5037101
[1] "Actual away testing data info"
[1] 1 2 1 1 0 1
```

This results in the expected number of goals for both the home and away teams being returned and stored in an appropriate dataframe. This function is executed four separate times, each time with a slightly altered version of the primary dataset to inform values for:

- The overall expected number of home goals scored from non-shot-based metrics: 'home_xG_ns_ovr'
- The expected number of home goals scored from non-shot-based metrics: 'home_xG_ns'
- The overall expected number of away goals scored from non-shot-based metrics: away_xG_ns_ovr
- The expected number of away goals scored from non-shot-based metrics: away_xG_ns

Once all values have been computed, the result is analysed and inputted into a weighting formula which is used to compute the ELO rating for both teams. See Section 4.2 for a detailed explanation as to how this is worked out.

Considering the ELO rating of both teams and incorporating a variety of other factors (which are further detailed in Section 4.3), a final prediction is made to determine whether a matchup results in a Home Win (H), Draw (D), or Away Win (A). This final prediction is stored in a new field, 'FTR',

meaning Full Time Result, which will subsequently be re-entered back into the original Random Forest, this time running a classification model rather than a regression model.

The key line of code that enabled a classification model to be generated was utilising the 'randomForest()' function. The following parameters have been defined:

- FTR~.
  - Defines that the Random Forest model will be using all fields found in the dataset to inform a prediction for the FTR field.
- data = train
  - Contains the dataset consisting of the prediction of a match outcome, and the ELO rating for both the home and away teams.
- mtry = 2
  - Specifies that two variables will be tried at each decision tree split. This value was decided according to the explanation found in Section 4.5.2.1.
- ntree = 501
  - Specifies that the Random Forest will be built using a total of 501 trees. This value was decided according to the explanation found in Section 4.5.2.1.

```
rf_class<-randomForest::randomForest(FTR~., data=train, mtry=2, ntree=501, importance=TRUE, proximity=TRUE)
```

This ultimately culminated in the following confusion matrix being created when evaluating how well the classification model performs on the training data:

```
               Type of random forest: classification
                     Number of trees: 501
No. of variables tried at each split: 2

        OOB estimate of  error rate: 55.67%
Confusion matrix:
    A   D    H class.error
A 727 372  775   0.6120598
D 454 317  809   0.7993671
H 627 517 1786   0.3904437
```

3.4.3 Neural Network (NN)

3.4.3.1 Neural Network Design

Neural Networks work by taking a value for each feature and then adjusting it using weights and a bias to get the final output. It uses each neutron to compute a weighted sum of its inputs and adjusts these weights throughout the training to predict the output. It is a suitable model for predicting a football game as it will use its hidden layers to try and make the predictions more accurate than just linear regression which will just use the inputs and output.

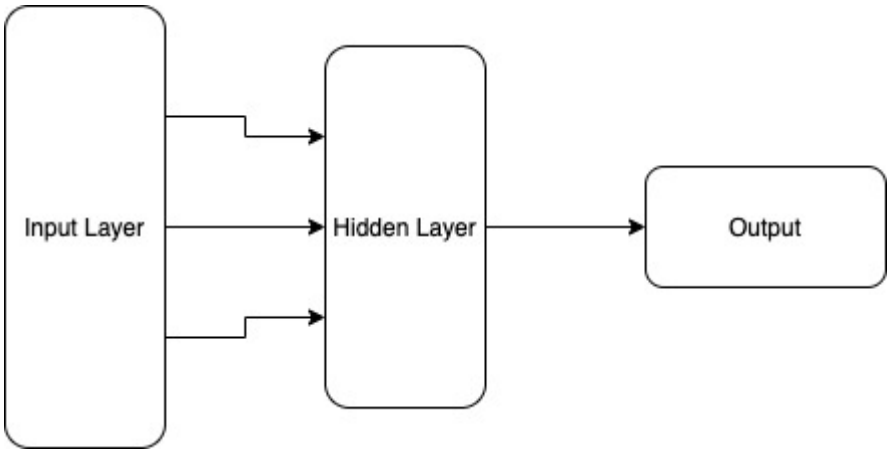


Figure 15, How data flows between the layers in a Neural Network

A paper from 2018 uses a backwards propagation neural network to predict the scores of two teams [35]. The model that they created had an average RMSE when predicting the scores of 1.079. This is different to the neural network used in the project as the model for the project uses forward

propagation which means that the model works out the weights by going from the input layer to the output. Which is different as backwards propagation works by going from the output layer to the input layer.

3.4.3.2 Neural Network Implementation

The Neural Network is used to calculate xGNS during the pre-processing and to calculate what the result of a game will be after all the pre-processing is completed. It is used to calculate the xGNS_ovr_H, xGNS_H, xGNS_ovr_A and xGNS_A. When the results of the neural network are examined, there is shown to be a variance of ±0.02 for the accuracy, ±0.15 for the RMSE and ±0.12 for the MAE. The values in table below shows that the away team models fitted better and were more accurate than the home team models.

	xGNS_ovr_H	xGNS_H	xGNS_ovr_A	xGNS_A
Accuracy	0.3259	0.3177	0.3411	0.3348
RMSE	1.27	1.29	1.12	1.13
MAE	0.93	0.96	0.82	0.83

The model for calculating xGNS is a function that takes in the formula used for the neural network, the data and the side of the team as shown in Figure 6. The formula used is dependent on what side is being calculated as it is the label and all the features, an example of this is *"FTHG~."*. The formula is first the label so in the example it is *FTHG* and followed by *"~features"*. However, a "." Can also be used to if all the other fields are the features instead of typing them all out. The data passed through is unique for all four xGNS calculated. The side argument is used for measuring how well the model ran as there is a difference in the values used so the confusion matrix needs the correct parameters passed through with the right number of levels as shown in Figure 7. The parameters that could be controlled in the neural network where the stepmax, hidden, act.fct and linear.output.

```
nn_weighting <- function(f, data, side) {
  nn <- neuralnet(f,
                  data=data,
                  stepmax=1e7,
                  hidden=2,
                  act.fct="logistic",
                  linear.output = TRUE)

  #plot(nn)
  predict = neuralnet::compute(nn, data)
```

- stepmax = This was set to "1e7" due to the model crashing if it was the default of 1 and the model needing a lot of steps to figure out the weights needed.
- hidden = This was chosen to be two as the data is not that complex and does not have many features.
- act.fct = This is "logistic" as it creates a logistic function to smooth the weights. Also, the other selection of "tanh" did not work as the model would crash.

- Linear.output = This was set to TRUE to make sure that the output would not be scaled between [0,1].

```
if (side == "H") {
  matrix <- confusionMatrix(factor(round(predict$net.result,digits = 0),
                                   levels=c("0", "1", "2", "3", "4", "5", "6",
                                            "7", "8", "9")),
                                   as.factor(data$FTHG))
  print(matrix)
  # Workout the errors for the model
  rmse = RMSE(round(predict$net.result,digits = 0), data$FTHG)
  mae = MAE(round(predict$net.result,digits = 0), data$FTHG)
  print(rmse)
  print(mae)
```

Once the pre-processing is completed the data is used in a neural network to work out the result of a game based off the ELO score of either side. From the final list of data the last 380 games were taken to be as unseen data and the rest was setup to train and run the neural network. It was first tested without cross validation using a similar approach to the way the xGNS model is created. After this cross validation was added to fold through the data which will allow for a better fit of the model to the data.

	No Folds	Folds
Accuracy	0.3974	0.2211
RMSE	2.08	1.40
MAE	1.59	1.17
MSE	4.32	1.97
Rsquared	NA	NA

The data in the table below shows that the neural network with cross validation has worse accuracy than the neural network without cross validation. Despite being lower accuracy the model with cross validation has a better RMSE, MAE and MSE. This is likely due to the cross-validating over-fitting the model so it will work well with the training and testing data but when tested against the unseen data it performs badly.

```
##### Neural Network Model Fitting #####
cv.error <- NULL
k <- 10
maxs <- apply(data, 2, max)
mins <- apply(data, 2, min)
scaled <- as.data.frame(scale(data, center = mins, scale = maxs - mins))

pbar <- create_progress_bar('text')
pbar$init(k)
for(i in 1:k){
  index <- sample(1:nrow(data),round(0.9*nrow(data)))
  train.cv <- scaled[index,]
  test.cv <- scaled[-index,]
  nn <- neuralnet(FTR~.,data=train.cv,hidden=1,stepmax=1e7,act.fct = "logistic",linear.output=TRUE)
  pr.nn <- compute(nn,test.cv[,1:3])
  pr.nn <- pr.nn$net.result*(max(data$FTR)-min(data$FTR))+min(data$FTR)
  test.cv.r <- (test.cv$FTR)*(max(data$FTR)-min(data$FTR))+min(data$FTR)
  cv.error[i] <- sum((test.cv.r - pr.nn)^2)/nrow(test.cv)
  pbar$step()
}

#### Neural Network Model no fitting ####
nn = neuralnet(FTR~.,
               data=data,
               hidden=1,
               stepmax=1e7,
               act.fct = "logistic",
               linear.output = TRUE
               )
```

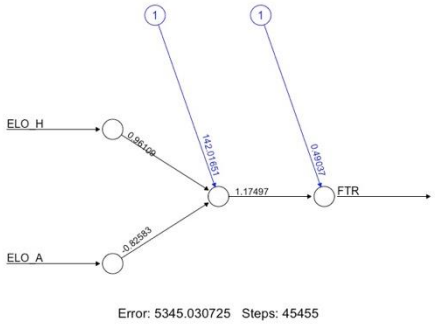


Figure 16, Neural Network diagram showing the weights and biases for the model without cross validation

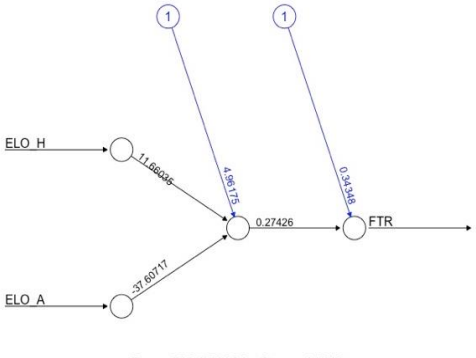


Figure 17 Neural Network diagram showing the weights and biases for the model with cross validation

3.5 Evaluation Methods

Type	Description	Formula
MAE	Mean absolute error is the measure of errors in an observation. As understood by its name it is the mean of the difference between the predicted value and the true value.	$MAE = \frac{\sum_{i=1}^{n} \lvert y_i - x_i \rvert}{n}$ MAE = Mean Absolute Error y_i = prediction x_i = true value N = number of data points
R squared	R squared is the variation that from the dependent variable that is predictable from the independent variable.	$R^2 = 1 - \frac{RSS}{TSS}$ R^2 = R squared RSS = sum of squares of residuals TSS = total sum of squares
MSE	Mean squared error is the average squared difference between the predicted value and the true value.	$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2$ MSE = Mean Squared Error n = number of data points x_i = true value y_i = prediction
RMSE	Root mean squared error is used to measure the differences between the values predicted by the model and the real values.	$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - y_i)^2}{N}}$ RMSE = Rroot Mean Squared Error N = number of non-missing data points x_i = true value y_i = prediction
Accuracy	Accuracy is the percentage of how many correct points the model predicted.	$Accuracy = \frac{\sum_{i=1}^{n} Y_i}{n}$ Y = Correctly predicted values n = number of data points

Precision	Precision is the overall accuracy of the model	$Precision = \frac{TP}{(TP + FP)}$ TP = True Positives FP = False Positives
Recall	Recall is the accuracy of a specific class.	$Recall = \frac{TP}{(TP + FN)}$ TP = True Positives FP = False Negatives

Method Name	Accuracy	Description
Random Guess	33.3%	Given that a football match can only ever end in a Home Win, Draw or Away Win, selecting one of these three results at random creates an average success accuracy of 1/3 being achieved.
Informed Prediction	42%	In section 1 research was done to determine a regular watcher of football verse a non-regular watcher of football and how good at predicting outcome each were. A regular viewer came out slightly on top, which is to be expected as they will have domain knowledge and be able to make more informed judgments.
Expert Prediction	58%	Veteran football pundit Paul Merson was able to successfully predict the outcome of an EPL football match with an accuracy of 58% over the course of an entire season.
Bookmakers	*Approx.* 43%	The highest performing bookmaker was able to successfully predict the outcome of EPL matches with an accuracy of 43%. Other bookmakers produced a similar level of precision.
Open league comp	51.9%	Open league provide public football data similar to which is used within this project. Each year they run a competition to see who can get the most accurate results. In 2018 the winners achieved 51.9% [36].

- The chance of randomly guessing the outcome of an EPL football match (primary benchmark)
    - This was chosen as our primary benchmark because if our solution is unable to consistently predict the outcome of football matches at a higher rate than that of simply randomly guessing, our solution will be considered obsolete. Thus, for this not to be the case, a minimum benchmark of 33.3% of all predictions being correct must be defined and achieved by the solution.
- Informed prediction – data via Google Forms survey filled out by a small sample of colleagues (secondary benchmark)
    - The results obtained from computing the average success rate from predictions made via a survey will be used as a secondary benchmark to give a human comparator for our AI-driven models to be compared against. Given that the survey was filled out by a mix of those who do and do not regularly watch football, it can be considered that the models should be able to outperform this

secondary benchmark, especially given the wealth of historical knowledge that our models have been trained upon.
    - Being able to outperform humans who have a varying amount of footballing knowledge is a strong signifier of success as it would show that the models are more reliable at making correct football predictions than the average person.
- The average success rate of mainstream bookmakers (stretch goal)
    - Bookmakers deploy particularly complex algorithms that allow for them to make more accurate predictions than the average person. However, given the unpredictability of modern football, this accuracy rarely exceeds a consistent rate of greater than 50% of predictions made being successful. Defining these values as a stretch goal for the solution to outperform would not only give evidence that the project can be evaluated as a resounding success, but it is also a rather viable target to aim for.

Each model will be compared to each of these values to see how the models created as part of this project compare. Additionally, the process used within the project and other aspects will be evaluated, such as the use of the CRISP-DM approach as outlines in the project plan.

4 Results

Predicting the outcome of a football match (or result of any sporting event) is a complex and challenging task due to the large variety of factors and randomness involved. Project BAWKOS strives to create and compare models and algorithms that predict professional English Premier League football match outcomes (win, loss, or draw) using artificial intelligence. The project aims to build models that are as accurate as possible, and at minimum better than a random or informed guess.

4.1 Assessment of Models

4.1.1 KNN

When reflecting on the implementation of the KNN model, the classifier had an accuracy of 0.6391 with the optimal K value. The testing set represented 20% of the original dataset which was unused for the training of the model this was done to test the effectiveness of the model. For each given class which represents the following – home win, draw and away the recall accuracy can be seen in the table and bar chart below.

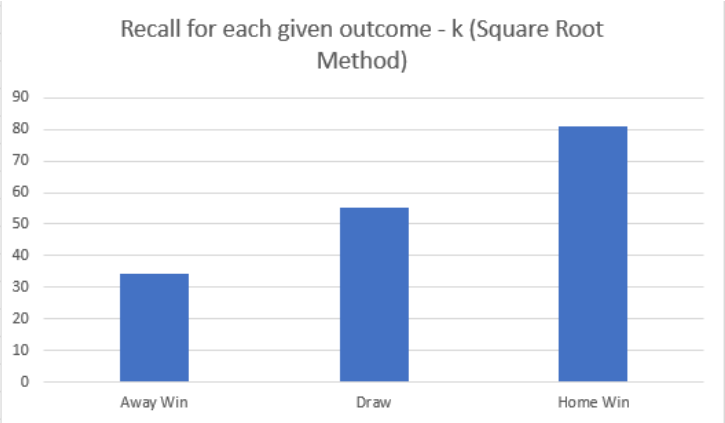


Figure 18 Recall for given Outcome

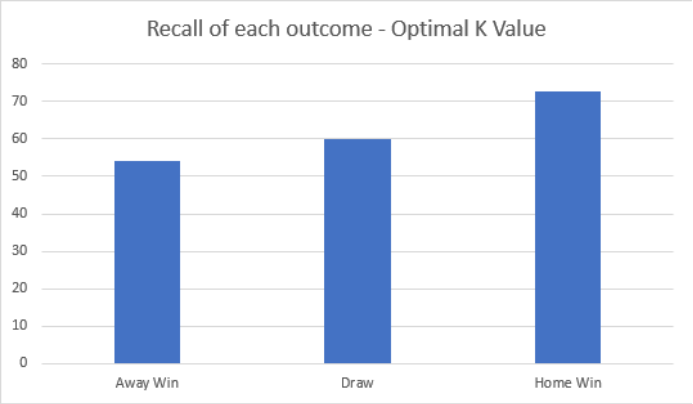


Figure 19 Recall of each outcome

The recall rate for each class varied, although predicting a home win had the highest rate of recall with a 0.8090. In isolation, this performed much better than the predetermined objectives we had set out for the project and when compared to the other models it has performed better than all bookmakers' models and went far and above the predictions from analysts such as Paul Merson when it came to predicting outcomes. As shown in the confusion matrix below, it is evident KNN does not predict very home losses correctly, since there are such fine differences between the ELO ratings that are determined, the neighbouring values may fall into either a win or draw. The overall model had an accuracy of 0.6391 this far exceeded the expectations for the project and achieves the stretch goals that were set.

For the model it would prove useful to see if the more variables added during the pre-processing phase as the expected goals from this section influences the ELO rating. KNN is very good as it gives a quick calculation time and high accuracy however the accuracy for the number of losses is evident that the quality of the ELO ratings could be improved if we used more inputs to determine the expected goals more concretely – this may have resulted in the classifier being able to predict more losses then presently constructed. Football being a sport which is very unpredictable it is clear to see so many teams which on the face of them having a high ELO rating in comparison to another team may not defeat the inferior team (with respect to ELO) as this fails to consider the line-ups, team news and other factors which have not been taking into consideration during the construction of the ratings. This is perhaps why there is a very minimal number of draws predicted by the algorithm due to the nature of most matches despite the randomness the teams with the better ELO would win.

Predicted values	Actual values		
	0	1	3
0	132	19	43
1	35	192	68
3	220	137	470

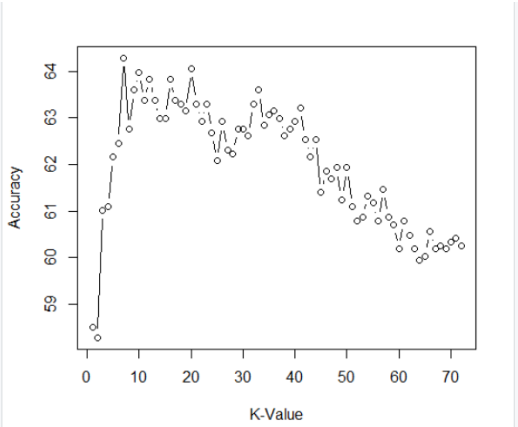


Figure 20 Accuracy vs K-value

Furthermore, based upon inspection it is telling that the value of k using the square root method was not necessarily the best k value to provide the best accuracy percentage. The best k value was 7 and this is shown on the plot above – the accuracy was 0.6391. The values of recall had been impacted by the change made in the K value this resulted in the following values for Home win, Draw and Away win. The bar charts show that with the optimal K the percentage of accuracy is much closer than before the percentage of away wins jumps up from 34.11% to 54.01%, the draws increase slightly from 55.17 to 60.06% however the percentage of home wins predicted drops from 80.9% to 72.81%. These values when mapped to Project Bawkos aims and

objectives beat all the bookmaker's and individuals guessing the results, Paul Merson though does compete with the away win and draw statistics if sticking to the 58% of games predicted.

Lastly, comparing the evaluation methods the optimal k performs better than the k value using the square root method – the predicted data from the model with the k value of 7 was marginally better with respective to the Mean Absolute Error, however the Mean Square Error dropped by 0.38. The r squared values are very close and this demonstrates they both have weak correlation and that there is no clear correlation.

With these statistics, it is evident that the project mission statement and goals have been able to be achieved by the KNN algorithm. It is also implied that the quality of data was very good as the KNN algorithm was able to find near neighbours and cultivate a high accuracy percentage.

4.1.2 Random Forest

Following the success of the implementation of the Random Forest model, unseen testing data was introduced to the model once it had been adequately trained. Similar to the correctness of how well the model worked with the training data, when making predictions on unseen testing data, a similar prediction accuracy was achieved. The unseen testing data consisted of a 20% section of the primary dataset that was not used during the training phase of the Random Forest model. The results of the testing phase of the model on unseen data are as follows:

		Actual Results		
		Away Win	Draw	Home Win
Predictions	Away Win	369	28	2
	Draw	11	288	11
	Home Win	0	11	596

When looking at the recall rate for each class, predicting a Home Win is the metric which has the highest rate of recall, at an astounding 97.87%. Not only is this far better than all the benchmarks that were defined early in the project lifecycle, but the model performs better than veteran sports pundit Paul Merson at predicting the outcome. Despite the fact that Draws are predicted correctly only 88%, upon closer inspection of the data this slightly lower accuracy begins to make more sense. Instead of predicting a draw when the two teams in a matchup have a similar ELO rating, the model opted to predict a win to the team which had the slightly higher ELO rating, even if there was only a marginal difference. Given that one of the drawbacks of a Random Forest is that there is very little interaction or flexibility, once the model 'learnt' to do this, it did not sway from 'voting' in this manner when the scenario arose. The model was able to successfully predict an Away Win with a

	MSE	MAE	RSquared
Sqrt(number of training samples)	2.455927	1.12614	0.09571423
K value of 7	2.075228	1.052432	0.1282794

recall rate of 97%, thus achieving a higher precision than both benchmark percentages for this specific metric. The stretch goal wasn't quite able to be achieved as the precision of the model fell slightly short of that of the two highest performing bookmakers.

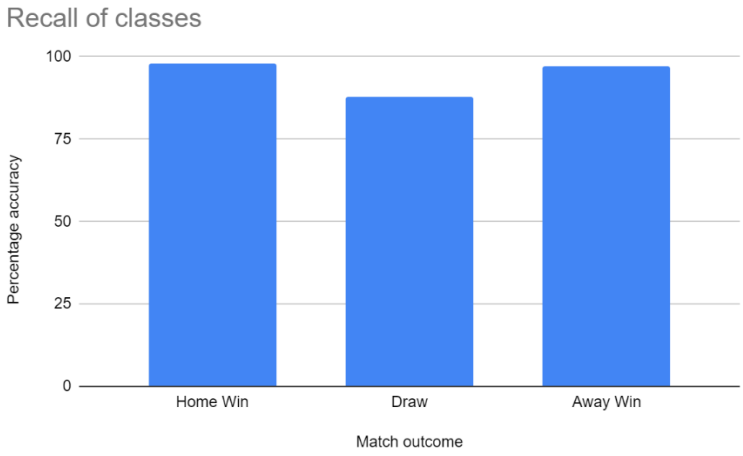


Figure 21 Recall of Classes

When observing the model's predictions on the unseen data, it can be seen that the precision for the Random Forest model is 95.21%. This clearly demonstrates that using a Random Forest to make predictions about the outcome of EPL football matches far outperforms both the primary and secondary benchmarks. This is to be expected as the model has access to a great deal of historical with which it can make informed decisions better than any individual is able to. Additionally, the model has been able to not only compete with the accuracy of the highest prediction accuracy from the three previously mentioned bookmakers but is able to outshine these as well. Given the innate randomness that occurs within football, it is not surprising to see that the model is not able to achieve a higher degree of accuracy than what was reached. Nonetheless, it is pleasing to see that the implementation of this model can be recognised as a resounding success.

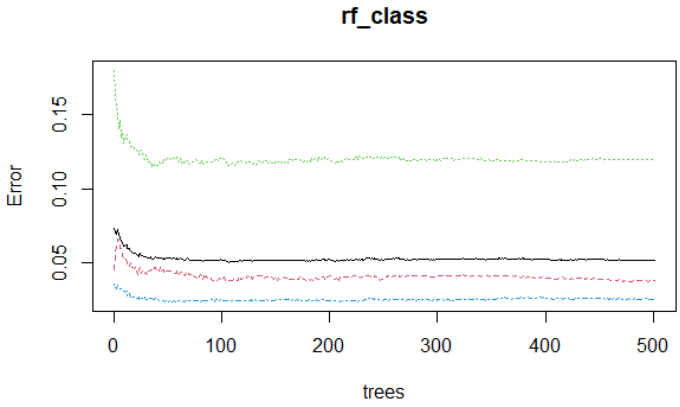


Figure 22 Error vs Trees

The Random Forest model can be improved by ensuring that the decision tree committee does not vote to traverse the incorrect path Specific to Project BAWKOS' solution, the Random Forest model consistently voted for the incorrect outcome by awarding a win to the team that has a marginally higher ELO rating rather than predicting the outcome of a match to be a draw. This led to the solution obtaining a recall accuracy of barely over 20% when obtaining True Positives for Draws. This could be enhanced by delving into the intricacies of how a Random Forest operates and making purposeful alterations to high level nodes within the forest to ensure that the votes are cast on the optimal route.

4.1.3 NN

The neural network was implemented twice in the project, once during the pre-processing and once for the results of unseen games. During the pre-processing the neural network's average accuracy was 33% which means that the model can predict the number of goals that side is going to score 33% of the time using stats that aren't shots. As well as having an average MAE of 0.89 means that on average it is a ±0.88 number of goals away from the actual number of goals. When looking at the data for the prediction of xGNS its shows that the model fit better for the away team data and in turn had a higher accuracy.

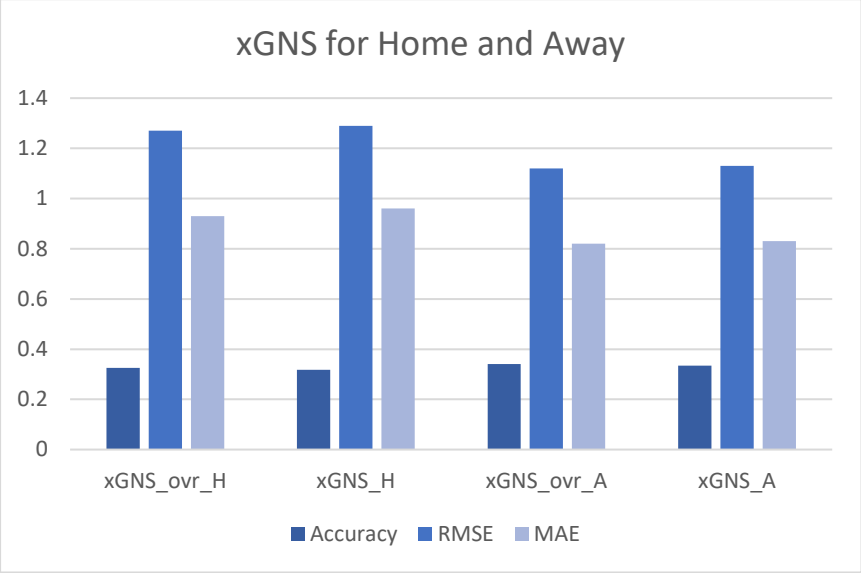


Figure 23 xGNS for Home and Away

Predictions		Actual Results		
		Away Win	Draw	Home Win
	Away Win	13	16	8
	Draw	0	0	0
	Home Win	132	68	143

The neural network predictions for the final unseen data without cross validation showed that the model predicted only 'Away wins' and 'Home wins'. This is most likely due to the model having a weighting that favoured home teams as 'Home wins' made up the largest part of the data. This could be improved in the future by providing more data or extra fields to compute the prediction.

As stated as in section 4.4.3.2 the neural network without cross validation had a higher accuracy than the one with cross validation due to it no being overfitted. This means that for any testing against the training dataset it will be very accurate but once it is tested against unseen data the biases set for the model are swayed in favour of the training data.

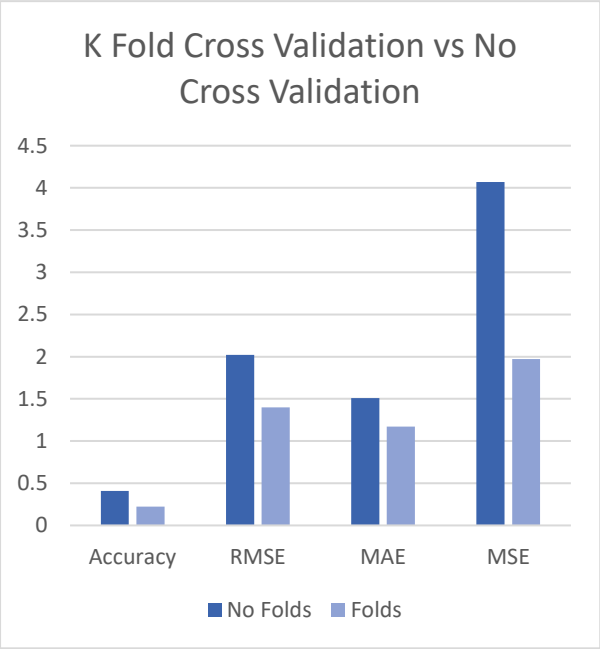


Figure 24 K Fold Cross Validation vs No Cross Validation

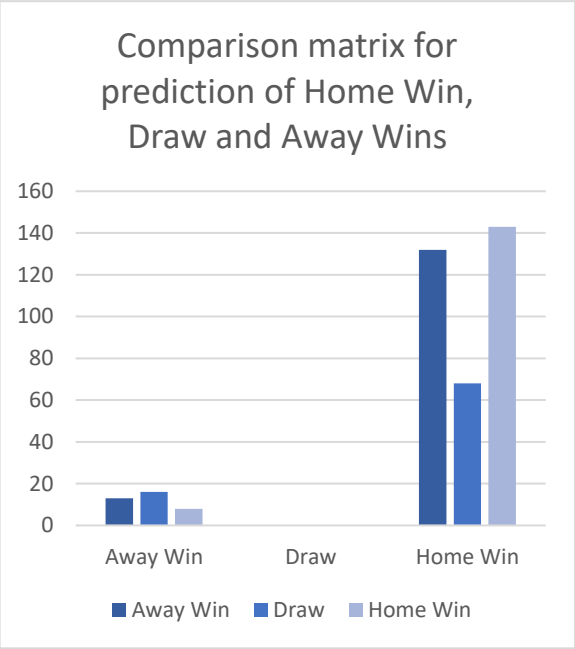


Figure 25 Comparison matrix for prediction of Home Win, Draw and Away Wins

5 Discussion

To determine which model performs best, the following results have been worked out regarding the precision of the model overall, as well as how accurately the recall rates are for each individual class:

	Precision of Model	Recall accuracy of Home Wins	Recall accuracy of Draws	Recall accuracy of Away Wins
KNN	68%	80.09%	55.17%	34.11%
Random Forest	95.21%	97.87%	88%	97%
Neural Network (Cross Validation)	22.11%	0%	100%	0%
Neural Network (No Cross Validation)	41.05%	94.70%	0%	8.97%

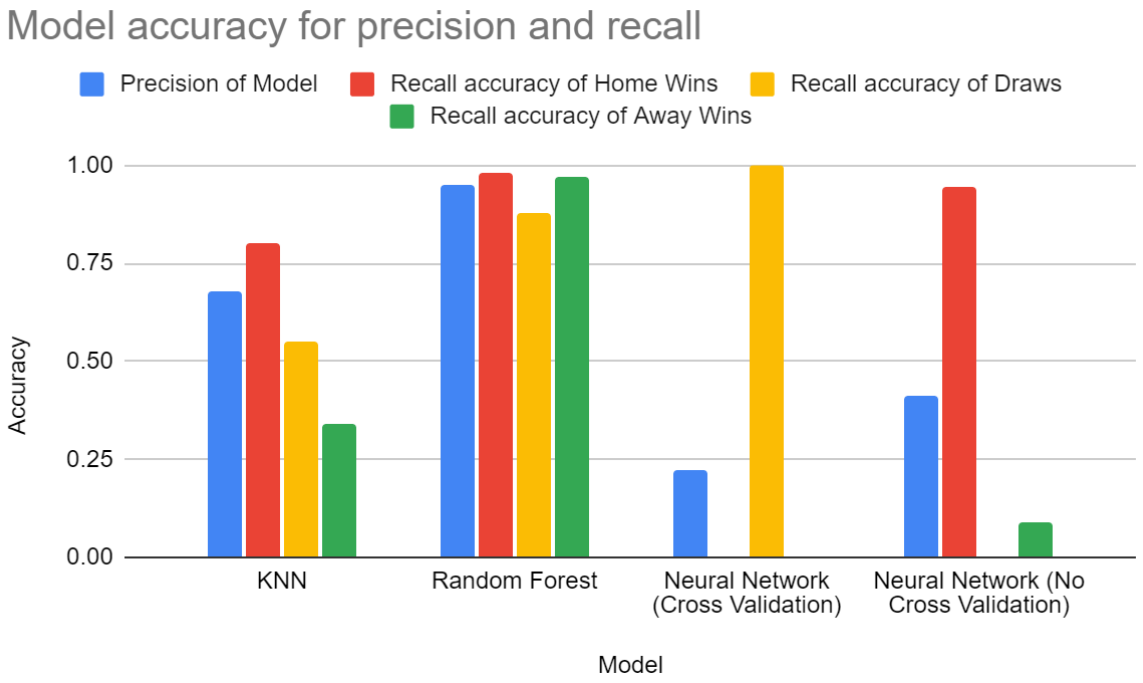


Figure 26 Model Accuracy for Precision and Recall

As the effectiveness of a machine learning model is not characterised solely by how well it is able to perform, but also the degree at which the model is able to minimise the error produced, the following table has been defined to represent how good each model was when it came to limiting error rates:

Model Type	RMSE Error rate	MAE Error Rate	MSE	RSquared Error rate	Model specific error rate
KNN (square root method)	N/A	1.12614	2.455927	0.09571423	N/A
KNN (optimal K)	N/A	1.052432	2.075228	0.1282794	N/A
Random Forest	1.17	0.92	NA	0.079	OOB Error: 5.28%
Neural Network (Cross Validation)	1.40	1.17	1.97	NA	522.77
Neural Network (No Cross Validation)	2.02	1.51	4.07	0.007	5345.03

When focusing solely on the precision of each solution it becomes apparent that all three machine learning models can be deemed a success. All models perform with a precision above the primary benchmark and give competitive rates when compared to the secondary benchmark as well as

against the accuracy of mainstream bookmakers. However, upon inspection of the recall accuracy for each individual class, the results become a little more varied in their degree of success.

The most notable feature is that the Neural Network model is not able to recognise any situations in which a match ends in a draw when the model is trained using the holdout method but, bizarrely, the NN model also becomes unable to identify any situations in which a match ends in a Home Win or Away Win when the model is trained using the k-Fold Cross Validation approach. Despite these shortcomings, a precision rate of 41.05% is obtained for an NN that is trained using the holdout method, while unfortunately a precision of only 22.11% is obtained for an NN that is trained using k-Fold Cross Validation. The fact that an NN (using Cross Validation) has a precision rate of below a random guess, it can be stated that this approach will be considered inviable.

Comparing the KNN and Random Forest models against the evaluation methods, it becomes clear that both significantly outperform the outlined metrics and can automatically be deemed a resounding success. Furthermore, when observing the recall rate for both models, all classes (with the exception of KNNs prediction of Away Wins) within these models perform at a substantially higher level than all the outlined evaluation metrics as defined in Section 4.5 of the document.

However, when it comes to settling upon an outrightly optimal model to select for predicting the outcome of football matches, a clear winner can be easily identified. While the precision and recall accuracy that is produced by a KNN certainly is respectable, it pales in comparison to the level of success that has been achieved by the Random Forest model. Being able to obtain an average successful prediction accuracy of 95.21% in a notoriously unpredictable subject area is certainly a feat worth recognising. Not only was the Random Forest model able to make consistently accurate predictions, but it was able to do so while having the minimal error margins of all models as well, thus proving that the implementation of a Random Forest is not only effective but also robust.

6 Conclusion

6.1 Summary

The project vision that was defined at the very beginning of the project plan has most certainly been met by all models, as all three of KNN, Random Forest and Neural Network are able to make predictions at a higher rate than that of a random guess, the primary benchmark. Not only this, but all the models have a precision rate that is competitive to that of the industry standards set by mainstream bookmakers, which was a stretch goal of Project BAWKOS. Furthermore, the KNN and Random Forest Models greatly outperformed our expectations by obtaining a precision of 68% and 95.21% respectively, with accuracies reaching as high as 97.87% for a Random Forest correctly predicting a win for the Home Team.

6.2 Solution Improvements

Given the time constraints of the project, there are many improvements that could be made to the project, albeit many of these improvements relate specifically to what is contained within the dataset and the pre-processing phase of the implementation. The clearest improvement that could be made is the inclusion and interpretation of qualitative data, many of which have a significant impact on the outcome of a football match. Data points such as the referee in charge of a match, personal circumstances surrounding a player and the behaviour patterns of an individual player all play a notable role in determining the outcome of a match.

Should this project have had a budget, perhaps having access to a paid dataset such as 'Opta' which provides coverage of qualitative fields may have further improved the final solution. Additionally, the quality and type of qualitative data found within the dataset could have been improved by collating multiple datasets into one. Having information regarding factors such as the geographical location and type of passes that were made in the game could have meant that the models perform with significant improvements. As many dangerous scoring positions do not result in a shot, having passes data would allow the model to better estimate the number of expected goals a team should have scored in a game. This in turn would have resulted in a more accurate ELO rating and subsequently meant that a more well-informed classification algorithm could have been deployed, giving more accurate results. While the team is very pleased with the precision of each model, it is foreseeable that, with more time spent fine-tuning the intricacies of the models, the existing solution could yet be enhanced further if such an intensive time constraint was not imposed.

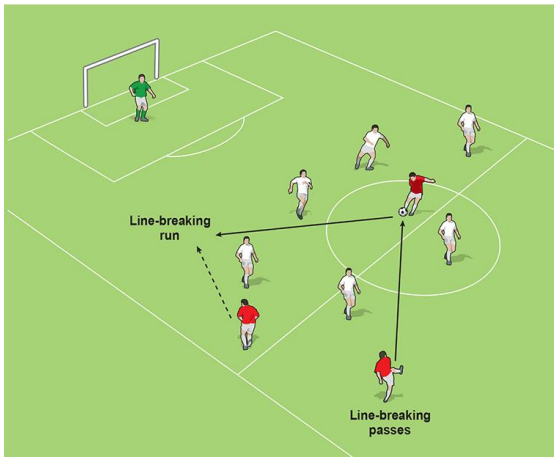


Figure 27 Passing Types [64]

6.3 Maintenance and Future Development

Due to the robust nature that the project was developed under, there is no requirement for continued maintenance once the final solution has been distributed. There are many directions that the project could be taken with more time and resources. A particularly interesting experiment to perform with the project could be to run a Monte Carlo simulation thousands of times on a single EPL season and observing which team has the highest probability of winning the league title, according to the models. This is made possible as Monte Carlo simulations are typically used to generate predictions for competitions that are in progress or are yet to begin. Given that the expected goals model has been defined and the current EPL season is underway, this could be an interesting route to take Project BAWKOS in the future.

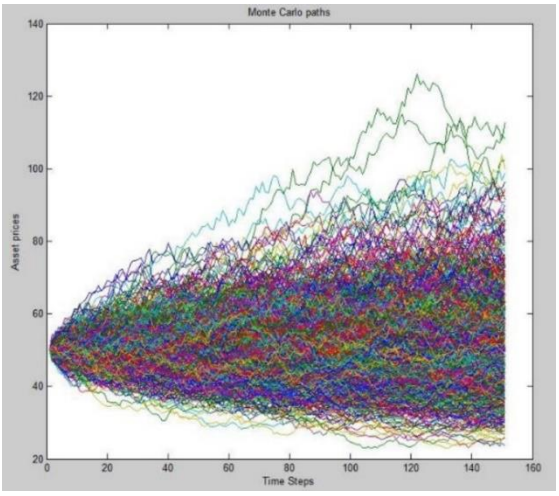


Figure 28 Monte Carlo Example [65]

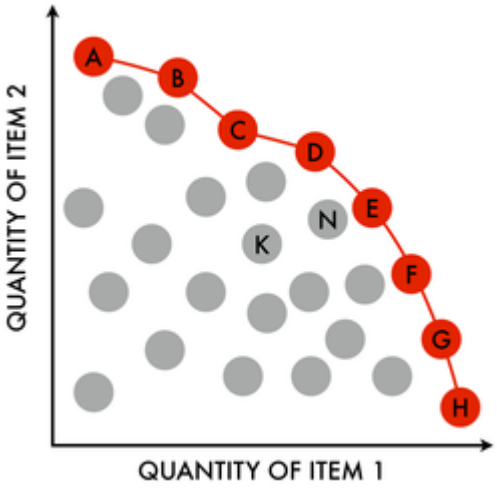


Figure 29 Patero Fronts [66]

Considering Patero efficiency into the final solution to create a Patero front that can allow the solution to make optimal predictions is a future development that could be interesting to explore. Patero efficiency is a situation where no individual or preference criterion can be better off without making at least one individual criterion worse. In the case of Project BAWKOS, each

option is first assessed, under multiple criteria, and then a subset of options is identified with the property that no other option can categorically outperform the specified option. Once this set has been identified a Pareto front can be deployed in order to restrict attention to the set of the most efficient choices of model for each of Home Win, Draw and Away Win. This effectively would combine the best aspect of each model to optimise the solution.

Another such direction to take could be dedicating time to enhancing the classification section of the solution to be able to make team, manager, and player specific profiles. Further developing Project BAWKOS in this direction may mean that the solution becomes viable from a business perspective, with the high level of personalisation giving potential for professional sports teams who may be interested in accessing this data-driven solution, to purchase access to the project as a service. Another expansion for this project has already been briefly touched upon where incorporating betting analysis into the solution could be considered a useful feature. The extended solution could analyse betting odds and validate whether any of the models could recommend good value bets and profitable betting strategies that generate long-term profit.

7 Bibliography

[1] "8 Oldest Sports in the World," Oldest, [Online]. Available: https://www.oldest.org/sports/sports/. [Accessed 10 November 2021].

[2] "key-data-global-sports-betting-industry," [Online]. Available: https://www.statista.com/statistics/1154681/key-data-global-sports-betting-industry/. [Accessed 29 11 2021].

[3] "artificial-intelligence-market-in-sports," [Online]. Available: https://www.mordorintelligence.com/industry-reports/artificial-intelligence-market-in-sports. [Accessed 29 11 2021].

[4] "premier-league-2015-16-how-the-odds-changed-as-leicester-claimed-the-title," [Online]. Available: https://www.skysports.com/football/news/11712/10261535/premier-league-2015-16-how-the-odds-changed-as-leicester-claimed-the-title. [Accessed 29 11 2021].

[5] "Home Page," [Online]. Available: https://www.premierleague.com/. [Accessed 29 11 2021].

[6] "28847955," [Online]. Available: https://www.bbc.co.uk/sport/football/28847955. [Accessed 29 11 2021].

[7] "predictability-sports-disciplines," [Online]. Available: https://www.bettingwell.com/sports-betting-guide/interesting-bookmaker-facts/predictability-sports-disciplines. [Accessed 29 11 2021].

[8] "Home page," [Online]. Available: https://sports.ladbrokes.com/. [Accessed 29 11 2021].

[9] "bet," [Online]. Available: https://www.paddypower.com/bet. [Accessed 29 11 2021].

[10] "home page," [Online]. Available: https://sports.yahoo.com/sportsbook/. [Accessed 29 11 2021].

[11] "could-paul-mersons-premier-league-predictions-bring-you-profit," [Online]. Available: https://www.skysports.com/football/news/15205/10974184/could-paul-mersons-premier-league-predictions-bring-you-profit. [Accessed 29 11 2021].

[1 2] "Home page," [Online]. Available: https://www.eloratings.net/. [Accessed 29 11 2021].

[1 3] "how-are-the-sky-sports-power-rankings-calculated," [Online]. Available: https://www.skysports.com/football/news/19024/9994064/how-are-the-sky-sports-power-rankings-calculated. [Accessed 29 11 2021].

[1 4] "ratings.pdf," [Online]. Available: https://www.football-data.co.uk/ratings.pdf. [Accessed 29 11 2021].

[1 5] "what-are-expected-goals-xg/," [Online]. Available: https://theanalyst.com/eu/2021/07/what-are-expected-goals-xg/. [Accessed 29 11 2021].

[1 6] "how-we-calculate-expected-goals-xg/," [Online]. Available: https://www.fantasyfootballfix.com/blog-index/how-we-calculate-expected-goals-xg/. [Accessed 29 11 2021].

[1 7] "shot-matrix-i-shot-location-and-expected-goals," [Online]. Available: https://cartilagefreecaptain.sbnation.com/2013/11/13/5098186/shot-matrix-i-shot-location-and-expected-goals. [Accessed 29 11 2021].

[1 8] "attack-defence-rating-using-scored-and-conceded-goals/," [Online]. Available: https://www.prosoccer.eu/betting-theory/attack-defence-rating-using-scored-and-conceded-goals/. [Accessed 29 11 2021].

[1 9] "Home page," [Online]. Available: https://www.google.com/search?q=Stoke+City+versus+Arsenal+on+19th+August+2017&oq=Stoke+City+versus+Arsenal+on+19th+August+2017&aqs=chrome..69i57.289j0j4&sourceid=chrome&ie=UTF-8#sie=m;/g/11ggb1bdd3;2;/m/02_tc;dt;fp;1;;. [Accessed 29 11 2021].

[2 0] "super-bowl-viewership-vs-world-cup-," [Online]. Available: https://www.statista.com/chart/16875/super-bowl-viewership-vs-world-cup-. [Accessed 29 11 2021].

[2 1] "fifa-world-cup-olympics-biggest-sports-world-21997," [Online]. Available: https://thebridge.in/tokyo-2020/fifa-world-cup-olympics-biggest-sports-world-21997. [Accessed 29 11 2021].

[2 2] "UEFA Club Coefficents," [Online]. Available: https://www.uefa.com/nationalassociations/uefarankings/country/#/yr/2022. [Accessed 29 11 2021].

[2 3] "file-2593818997-," [Online]. Available: https://insight.gwi.com/hs-fs/hub/304927/file-2593818997-. [Accessed 29 11 2021].

[2 4] "SkySports YouTube," [Online]. Available: http://www.insideworldfootball.com/2019/10/30/sky-sports-earns-1m-month-youtube-liverpool-platforms-leading-club/#:~:text=Inside%20World%20Football-,Sky%20Sports%20earns%20%241m%20a%20month%20from,Liverpool%20are%20platform's%20leading%20club&text=Octobe. [Accessed 29 11 2021].

[2 5] "business-56164159," [Online]. Available: https://www.bbc.co.uk/news/business-56164159. [Accessed 29 11 2021].

[2 6] "serpapi.com/?gclid=Cj0KCQiA7oyNBhDiARIsADtGRZaZjYgU14bg7XJfpNG6yLXS5XC89Vo0GPLR3Zqg5_Z5n6LGv71GVwgaAmLxEALw_wcB," [Online]. Available: https://serpapi.com/?gclid=Cj0KCQiA7oyNBhDiARIsADtGRZaZjYgU14bg7XJfpNG6yLXS5XC89Vo0GPLR3Zqg5_Z5n6LGv71GVwgaAmLxEALw_wcB. [Accessed 29 11 2021].

[2 7] "Opta-football," [Online]. Available: https://www.statsperform.com/opta-football/. [Accessed 29 11 2021].

[2 8] "football-data.co.uk/," [Online]. Available: https://football-data.co.uk/. [Accessed 29 11 2021].

[29] "International-football-results-from-1872-to-2017," [Online]. Available: https://www.kaggle.com/martj42/international-football-results-from-1872-to-2017. [Accessed 29 11 2021].

[30] "What is Machine Learning Definition Types Applications and Examples," Potentiaco, [Online]. Available: https://www.potentiaco.com/what-is-machine-learning-definition-types-applications-and-examples/. [Accessed 12 November 2021].

[31] "about," [Online]. Available: https://www.eloratings.net/about. [Accessed 29 11 2021].

[32] "@matheuskempa," [Online]. Available: https://medium.com/@matheuskempa. [Accessed 29 11 2021].

[33] "k-nearest-neighbor," [Online]. Available: https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/k-nearest-neighbor#:~:text=K%2DNearest%20Neighbors%20(KNN)%20is%20a%20standard%20machine%2D,large%2Dscale%20data%20mining%20efforts.&text=The%20number%20K%20is%20typically,40. [Accessed 29 11 2021].

[34] "predict-football-results-with-random-forest-c3e6f6e2ee58," [Online]. Available: https://medium.com/@nicholasutikal/predict-football-results-with-random-forest-c3e6f6e2ee58 . [Accessed 29 11 2021].

[64] "play-the-final-pass/," [Online]. Available: https://www.soccercoachweekly.net/newsletters/play-the-final-pass/. [Accessed 29 11 2021].

[65] "an-overview-of-monte-carlo-methods-675384eb1694," [Online]. Available: https://towardsdatascience.com/an-overview-of-monte-carlo-methods-675384eb1694. [Accessed 29 11 2021].

[66] "Pareto_front," [Online]. Available: https://en.wikipedia.org/wiki/Pareto_front. [Accessed 29 11 2021].

8 Figures

9 Appendix

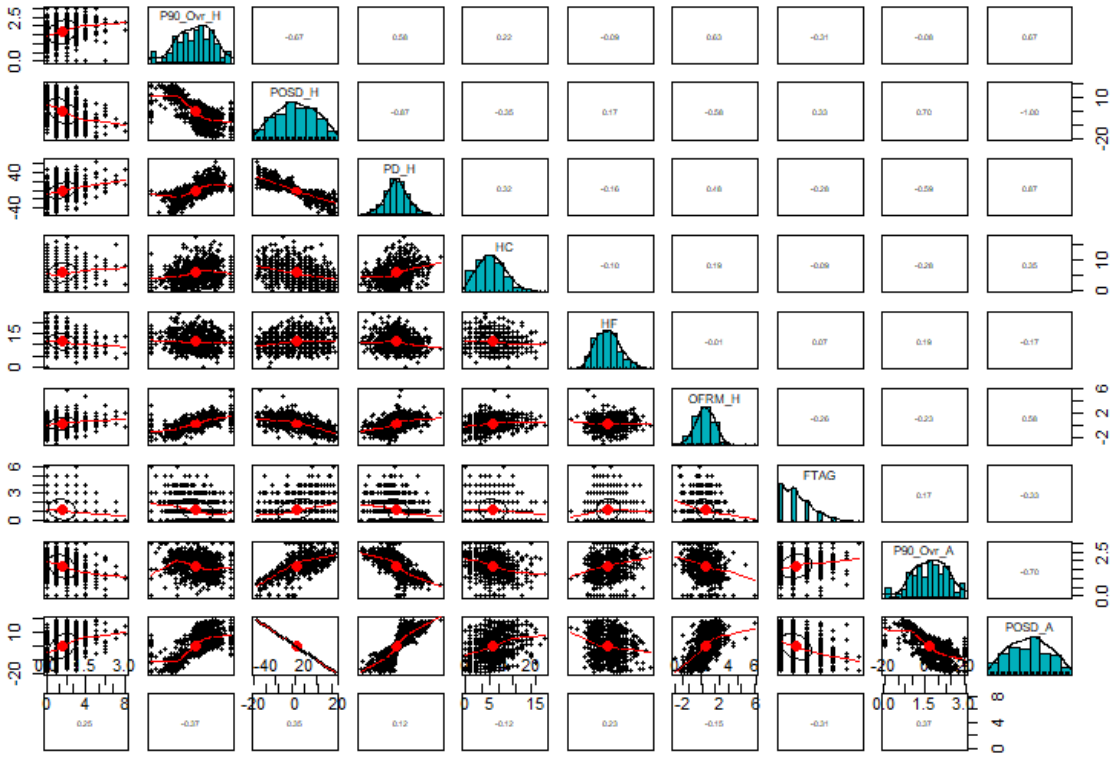


Figure 30 Scatter Matrix