Using machine learning to predict football match outcomes based on clubs' transfer spendings

by

Varun Kale

(URN: 6475451)

Data Science for Health BSc (Hons)

Department of Computer Science

University of Surrey

Declaration of Originality

I confirm that the submitted work is my own work and that I have clearly identified and fully acknowledged all material that is entitled to be attributed to others (whether published or unpublished) using the referencing system set out in the programme handbook. I agree that the University may submit my work to means of checking this, such as the plagiarism detection service Turnitin® UK. I confirm that I understand that assessed work that has been shown to have been plagiarised will be penalised.

Acknowledgements

I would like to thank my Supervisor Dr Mark Manulis for his advice, guidance and regular sessions throughout the duration of the project as well as the Computer Science department staff and teachers for their constant support over the last few years. It has been of great value to me and has provided me the skills and resources necessary to complete this project.

I also want to thank everyone that I worked with during my 13-Month Industrial Placement at Amgen Inc. The programming skills and software engineering principles that I developed at Amgen Inc., have been a relevant input during this project and will serve me even after my University degree. It has helped me increase my confidence, ability in using technologies used in this report and helped me gain soft skills that are transferable across industries.

I must also thank my friends and family for their support and constant encouragement during my entire University degree, it has been a major motivational factor for me.

Abstract

As one of the most popular sports in the world, football has been analysed and followed by football enthusiasts across the globe. Prediction of outcomes in sports has been an area of significant investments across various industry segments. Football prediction has been a topic that has received great attention over the years, especially in the top divisions of respective countries. Currently a wide range of historical data is involved including key statistics such as total shots in a match, shots on target, betting odds for that specific game and expected goal statistics to be used for data analysis. Researchers and organisations have tried to predict the outcome of football matches using these historical datasets and many have fallen short in their research. Football tends to be one of the hardest sports to predict as the factors such as shots and possession could all point towards one team, but the variable that counts, goals, could be favoured towards the other team, making all trends redundant.

The main objective of this report is to test different machine learning techniques to predict the outcome of football matches using features including home team, away team, shots, yellow/red cards and other in game data. This report will assess and evaluate the performance of the machine learning models against other techniques that have been previously used by researchers. The expenditure and income (transfer data) of the teams will be taken into account when creating the machine learning models in addition to the match specific information. A model with the transfer data of the teams will also be compared to a model that does not consider the transfer information.

For this project, two main machine learning models were created and evaluated. The first model was the Deep Neural Network. Neural networks aim to emulate the way the human brain works in terms of the stimulus created through neural connections and they have a complex topology and can have multiple hidden layers and factors such as weights, biases, synapses and activation functions. A Random Forest model is the second model used in this project. A random forest is used to combine many decision trees together into a single model to get a higher accuracy and better results from the rules generated from the decision trees.

The resulting models will be evaluated and used to predict the results of a Premier League football match and display this using an R Shiny based application. This app enables the users to select a home and away team along with inputting each team's expenditure and income on their transfers based spending to output a prediction of the result as well as probability of a win, loss or a draw for each team.

Table of Contents

Declaration of Originality 2
Acknowledgements 3
Abstract 4
List of Figures 7
List of Tables 8
List of Equations 9
1. Introduction 10
1.1 Problem Background 10
1.2 Project Motivation 11
1.3 Project Aims and Objectives 12
1.4 Structure of the Report 13
2. Literature Review 14
2.1 Football Rules and Transfer Market Background 14
2.1.1 Football Rules 14
2.1.2 Premier League 14
2.1.3 Transfer Market 15
2.2 Introduction to Machine Learning Techniques 17
2.2.1 Supervised Learning 17
2.2.2 Random Forest 17
2.2.3 Neural Networks and Deep Learning 18
2.3 Machine Learning in Sports Prediction 22
2.3.2 Neural Network 22
2.3.1 Random Forest 23
2.4 Football Prediction Using Statistical Approaches 24
2.5 Football Prediction using Machine Learning 25
2.5.1 General Machine Learning Techniques 25
2.5.2 Random Forest 26
2.5.3 Neural Network 26
2.5.4 Previous Literature Comparing NN and RF 27
2.6 Football Prediction Including Transfer Data 29
3 Data Analysis 30
3.1 Data Collection 30
3.1.1 Historical Results Data 30
3.1.2 Transfer Data 30
3.1.3 Transfer Inflation Data 31
3.2 Data Exploration 32
3.3 Combining the Datasets 37
3.4 Key Features of the Dataset 38
3.4.1 Removing Redundant Fields 38
3.4.2 Feature Engineering 38
3.5 Data Pre-Processing 39
3.5.1 Random Forest Preparation Steps 39
3.5.2 Neural Network Preparation Steps 40

3.6 Splitting of Training and Testing Data 42
4 Model Choices *43*
4.1 Grid Search 43
4.2 Neural Network 44
4.3 Random Forest 46
5 Evaluation of Models *48*
5.1 Evaluation Metrics Used 48
5.2 With Transfer Data Models 51
5.2 Without Transfer Data 53
6 Football Prediction Application *56*
6.1 Technical Choices for Application 56
6.1 Shiny Application and R Libraries Used 58
6.2 Functional Requirements 59
6.3 Non-Functional Requirements 63
6.4 Deployment of Application 64
6.5 Football Prediction Application Screenshots 66
6.4.1 Team Page 66
6.5.2 Historical Results 67
6.5.3 Transfers Page 68
6.5.4 Prediction Page – With Transfer Data 69
6.5.5 Prediction Page – Without Transfer Data 70
6.5.6 Prediction Page – Loading Page 71
7 Testing *72*
7.1 Functional Requirements Testing 72
7.2 Non-Functional Requirements Testing 73
8 Project Evaluation *74*
8.1 Comparison of Other Models 74
8.2 Critical Discussion 76
8.2.1 The Project Objectives 76
8.2.2 Models and Previous Literature 78
8.3 Strengths and Weaknesses of This Project 79
8.4 Academic Contributions 80
8.5 Resources Used 81
9 Statement of Ethics *82*
10 Conclusion *83*
List of References *84*
Appendix *88*

List of Figures

Figure 1 Premier League Titles by Club 15
Figure 2 Random Forest Example Structure [13] 18
Figure 3 Example of a Deep Neural Network [15] 19
Figure 4 Example of Backpropagation in Neural Networks [16] 19
Figure 5 Gradient Descent Step Sizes [18] 20
Figure 6 Example Data used for Random Forest Model, Yezus [32] 26
Figure 7 Example of a Transfer Converted to Todays Money [39] 31
Figure 8 Factor to Convert Transfers to Todays Money 32
Figure 9 Expenditure and Net Spend of Teams 33
Figure 10 West Ham and Liverpool Net Income Comparison 35
Figure 11 Raw Data Example with Missing Data 35
Figure 12 Correlation Plot of Results Data 36
Figure 13 Occurrences of Home or Away Wins or Draws 36
Figure 14 Model Data Once 3 Datasets Were Combined 37
Figure 15 Ordinal, Discreet and Symbolic Data Split 39
Figure 16 One-Hot Encoding Example 40
Figure 17 Example 5-Fold Stratified Cross Validation [41] 42
Figure 18 Trade-Off Between Overfitting and Underfitting 43
Figure 19 Accuracy Measurements When maxnodes Was Changed 46
Figure 20 Accuracy When the Number of Trees Was Changed 47
Figure 21 True Positive Example 48
Figure 22 False Positive Example 49
Figure 23 With Transfer Accuracy Comparison 51
Figure 24 Recall for With Transfer Model 51
Figure 25 Precision for With Transfer Models 52
Figure 26 Accuracy for Without Transfer Models 53
Figure 27 Precision and Recall of Without Transfer Models 54
Figure 28 RMSE Score Comparison of Without Transfer Model 55
Figure 29 RStudio IDE template 57
Figure 30 Example of Token Necessary to Link Application to shinyapps.io 64
Figure 31 Shinyapps.io Admin Home Page for the Football Prediction 65
Figure 32 Team Page Screenshot 66
Figure 33 Historical Results Tab Screenshot 67
Figure 34 Transfers Page Screenshot 68
Figure 35 Prediction Page for With Transfer Model 69
Figure 36 Prediction Page Screenshot for Without Transfer Model 70
Figure 37 Prediction Page Screenshot - Loading Screen 71
Figure 38 Comparison of Model Accuracies Across Papers 74
Figure 39 Comparison of F1-Scores Across Papers 75

List of Tables

Table 1 Advantages and Disadvantages of Random Forest 18
Table 2 Advantages and Disadvantages of NN 21
Table 3 Historical Results Raw Data 30
Table 4 Transfer Information Raw Data 31
Table 5 Transfer Inflation Raw Data 31
Table 6 Parameters for NN With Transfer Model 44
Table 7 Parameters for DNN Without Transfer Model 44
Table 8 Description of Packages Used in This Project 58
Table 9 Functional Requirements 62
Table 10 Non-Functional Requirements 63

List of Equations

Equation 1 Linear Regression 17
Equation 2 Standardisation 20
Equation 3 Normalisation used by Davoodi and Khanteymoori [21] 22
Equation 4 Poisson Model used by Dixon and Coles [2] 24
Equation 5 Naïve Bayes Classifier used by Dobravec [29] 25
Equation 6 Consistency Feature Calculation by Esme and Kiran [31] 26
Equation 7 Data Normalisation 41
Equation 8 Precision Formula 49
Equation 9 Recall Formula 49
Equation 10 F1-Score 49
Equation 11 RMSE Formula 50
Equation 12 MAE Formula 50
Equation 13 R2 Equation 50

1. Introduction

1.1 Problem Background

As one of the most popular sports in the world, football has been analysed and followed by a large number of audiences globally. The recent 2018 FIFA World Cup Final held in Russia had an average live audience viewership of 517 million people with more than 1.1 billion people tuning in over its 90-minute period [1]. Football prediction has been a topic that has received great attention over the years, especially in the top divisions of the respective countries. Currently a wide range of historical data is involved including key stats such as total shots in a match, shots on target, betting odds for that specific game and expected goal statistics that is used for data analysis. The large amount of additional data that is now readily accessible has allowed researchers and data science establishments to enhance their ability in the field of predicting football results, leading to overall higher accuracies in their predictions.

The quality of data that is currently available for research is very rich in terms of, the details of each team's expenditure and income, how many players have been bought and how many were sold each year, and similar triggers for these buy/sell decisions.

Each year teams are spending progressively more on players transfers. For example, in 2000 the world record fee for Hernan Crespo to move to Lazio, was £35,000,000. Now the world record fee is over five and a half times that much, at £198,000,000. The average transfer fees for Europe's top 5 leagues have increased every year since 2013/14 [2]. There have been many reasons for the increase in the money spent by teams throughout the years, including inflation of the club's countries' respective currencies, direct investment by current or new owners and the increase of income from TV rights. Recently UEFA financial fair play has been introduced to prevent clubs spending beyond what they have and remove any "financial doping". Clubs cannot spend more than €5 million more than they make a year or €45 million if the clubs' investors were willing to cover the costs [3].

Many researchers have attempted to use statistical analysis and machine learning in order to predict the outcome of football matches, however, very few have included the expenditure and income of teams in their models and predictions, due to the complexity of the transfer market as explained in section 2.1.3. The exploration of how the net income of each team affects their performance and the inflation of prices must be taken into account. This project aims to create an application to predict the outcome of football matches taking into account both historical results data and the transfer data.

1.2 Project Motivation

This project has been chosen as it combines my interest in Artificial Intelligence and Machine Learning with the software development skills that I learnt during my professional training year. During my placement period I worked with R programming language and therefore, I have decided to create a R Shiny application for my project. The modules taught in my final year will allow me to expand my knowledge, especially the Practical Business Analytics, where we were taught the concepts of machine learning using R including data cleaning, pre-processing, modelling and evaluations. My interest and knowledge in machine learning techniques has increased vastly over the duration of my University degree and I have been able to put my knowledge and learning into action during this final year project.

My interest in football was a big motivation for me and researching about what researchers have previously done using football and other sports data created a drive for me. Being able to learn and analyse the statistical and machine learning techniques previously used and tested and being able to build on them to include transfer data, which I follow regularly, provided inspiration for me to complete this project.

1.3 Project Aims and Objectives

The overall aim of this project is to investigate the machine learning techniques that can be used for predicting the outcome of football results and how the expenditure and income of teams can affect these predictions. Different models will be created and compared; one will contain the transfer data for each of the teams and one will exclude the expenditure and income statistics.

The objectives of this project are:

Step 1: Gather data for historical football results and teams transfer data

- To generate the machine learning models, historical data and teams transfer data will need to be obtained. Collecting the data will be the first step before the model can be created.

Step 2: Research the methods used and models that have been created in predicting the outcome of sports games

- This will show the methods that have been tried and tested previously by researchers including the advantages and limitations of the models created.

Step 3: Research on how the expenditure of football teams can affect their performance

- To gain an understanding of how transfers affect teams and how increased inflation has changed the way football clubs spend their money.

Step 4: Explore, clean and manipulate the data to contain only the relevant features

- Manipulate the datasets to contain the relevant information that will be used for the prediction of football matches.

Step 5: Generate the model to predict the outcome (Win/Draw/Loss) of football matches in the English Premier League

- After gathering the datasets and manipulating them, a model will be developed which is capable of predicting the full-time result of a premier league game.

Step 6: Create an R Shiny application to allow users to get the results of the predictions

- A R Shiny application that will allow users to view historical data, information about each team and make prediction on the outcome of football matches depending on a variety of features including the expenditure and income of the relevant teams.

1.4 Structure of the Report

This report's structure shows the different chapters in the report and a brief description of what can be found in each section.

Chapter 1 - Introduction

This section contains aims of the project along with the motivations which describes why I chose to make this project. There is also a success criteria and details of the structure of the report.

Chapter 2 - Literature Review

The literature review contains my findings of previous attempts to use statistics and/or machine learning techniques to predict the outcomes of football matches. Brief explanations of the existing implementations are also provided.

Chapter 3 - Data Analysis

This section contains information on the raw data collected for the historical football results and historical transfer data. Data exploration and cleaning will take place in this section to gain a better understanding of the data to be used. The data manipulation techniques will be discussed as well as how the key features were determined.

Chapter 4 - Neural Network and Random Forest Models

This section details the creation of both the Neural Network and the Random Forest, how its structure was determined and different hyperparameters of the models.

Chapter 5 – Results and Testing of Neural Network and Random Forest Model

This section compares the results of both models and their relative metrics, such as accuracy. It will also document the shortcomings and highlight the benefits of each of the models.

Chapter 6 - Football Prediction App Created

This section discusses the implementation of the model created into an application for a user and the performance of the application. This section also describes the functional requirements and non-functional requirements of the application.

Chapter 7 - Testing

This section outlines the steps taken when testing the application, including system testing and whether the requirements have been fulfilled.

Chapter 8 - Project Evaluation

This section contains the reflection on the project, with any parts that could have been improved and anything that could have been done differently if the project was to be completed again.

Chapter 9 - Statement of Ethics

Reflection on the ethical and legal issues for the application and the research for the project will be discussed in this section.

Chapter 10 - Final Conclusion

This section gives the final remarks on the project and reflection upon its completion.

2. Literature Review

This section will include some of the existing methods and relevant literature that have been used in the process of predicting sporting results. This literature review will summarise the existing solutions and how this project can improve on these designs. It will also discuss the machine learning techniques used when predicting the outcome of sporting events and discuss the strategies used when creating the model for this project.

2.1 Football Rules and Transfer Market Background

2.1.1 Football Rules

Modern football is a sport that was first played around the middle of the 19th century. It is the most watched sport in the world with an estimated global following of around 4 billion. The 2018 World Cup that was hosted in Russia had the third most estimated audience for any televised broadcast event after the 2012 London and 2016 Rio De Janeiro Summer Olympic Games [4].

The simplified rules of the game are:

1. The sport is played on a field with the dimensions of around 90-120 meters in length by 45-90 meters in depth and for a duration of 90 minutes consisting of two 45-minute halves.
2. There are two teams consisting of 11 players each, of which, 10 are outfield players and 1 is a goalkeeper. Each team is allowed up to a maximum of 3 substitutes per match.
3. The objective is to put the ball into the opposition's goal/target and when this is done the team gets a point/goal
4. The team with the most points/goals at the end of the match is the winner. If both teams have the same number of points/goals, then the match ends a draw.

Relevant Match Events:

1. Shots - Occurs when the team hits the ball in order to attempt to score a goal
2. Shots on Target - Happens when the team takes a shot, and it is successfully directed towards the goal and it goes in the goal or the goalkeeper prevents the ball from entering the goal.
3. Yellow/Red Cards - awarded when the referee deems foul play has occurred. Yellow cards are given for smaller fouls and red cards are given for serious offences and the player with the red card must leave the pitch and the team play with a player less.

2.1.2 Premier League

The English Premier League was formed in 1992 runs from the period of August to May every year. It is the most popular football league in the world with the league boasting 3.9 billion viewers across the 2018/19 season. It is the top league in England [5]. Manchester United have won the most Premier Leagues (13) and there have been 7 different winners over its duration as shown in Figure 1.

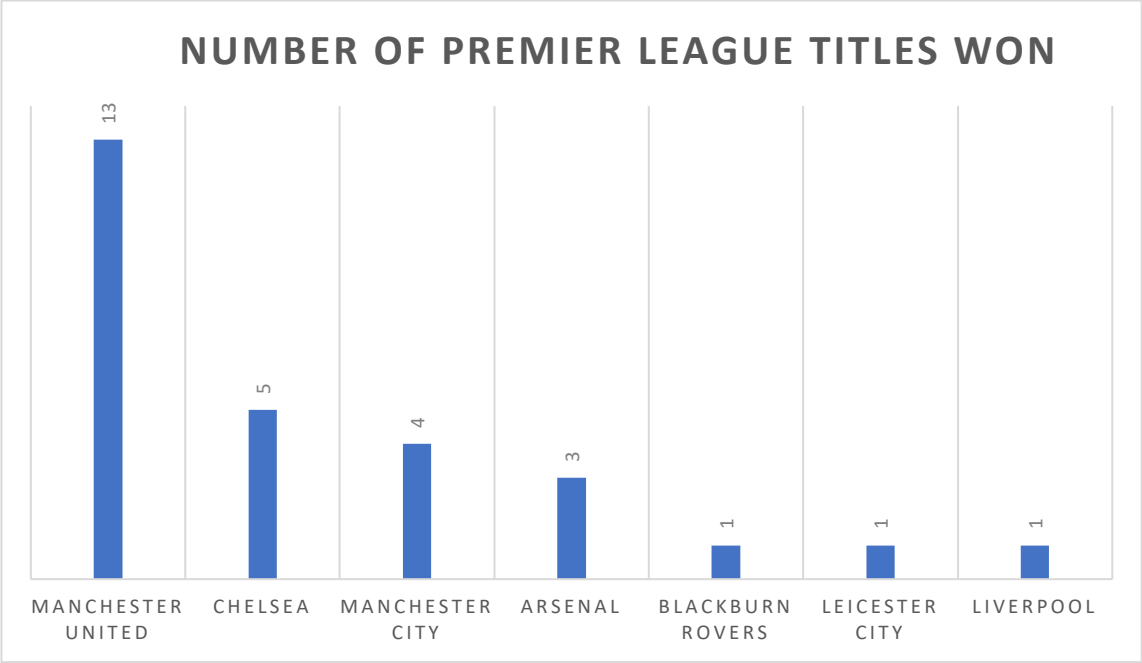


Figure 1 Premier League Titles by Club

A brief set of rules for the English Premier League are discussed below [6]:

1. The league consists of 20 teams. Each team plays each other at their home stadium and at their opposition's away stadium. Therefore, each team plays 38 games, and the season consists of 380 games.
2. 3 points is awarded to the team that wins the match, 1 point for each team if the result is a draw and 0 for the team that loses.
3. The team with the most points after the 380 matches has been played are crowned champions and the 3 teams with the least points are relegated to the English second division, also known as the Championship. The 3 places are replaced by the top 3 teams from the Championship for the next season.
4. If the points achieved are the same at the end of the season the team with the final position is determined by the goal difference of the teams.

2.1.3 Transfer Market

The transfer window is a period that occurs twice a year all around the world where teams can exchange players for a fee. This allows for teams to improve or sell their players to make a profit. In simple terms a transfer occurs when a player who is under a contract at a club moves to another club. Each player signs a contract with a team which specifies how long the player is theirs for. The buying club usually pays the selling team a compensation fee for having to break the contract, this is known as a transfer fee. There are three main types of scenarios that can trigger a transfer [7]:

1. An interested club approaches a current players club.
2. A current players team approaches another team because they want to sell the player.
3. A player is out of contract and is searching for a new club.

When a player has more years left on their contract, they tend to be sold for more money than they would if they had a small number of years left. There are many factors that determine the price of a player.

- Squad status - if they are an important member of the team, they may sell for more money.
- Age - if they are young, they may sell for more money as they can perform at the level for more years than an older player.

- Performance - A player performing at their peak is likely to sell for more.
- League Factor - A player playing in a higher division will sell for more.
- Position - Attacking players tend to sell for more money than defensive players, however this trend is starting to even out.
- Contract Length - The remaining years on the players contract tends to play a major role in the amount the player sells for.
- Competitor Team - if the team they are selling to a rival team they may ask the buying club to pay a higher fee for the player than they would for a team from a different country.

A team may loan a player for a temporary period of time if both clubs agree, and this tends to be a period of 6-12 months before the player returns to their parent club. As trends show, the richer clubs in the country tend to spend more money on transfers and would be able to afford players of a higher calibre. These clubs would also be able to attract these high-quality players due to the large contracts they pay the players. Therefore, the richer clubs tend to perform better in the Premier League. However, in 2011 the Union of European Football Associations (UEFA) introduced Financial Fair Play (FFP) Regulations meaning that clubs cannot spend more than the income they make and must balance their books over a three-year period. This was introduced in order to prevent clubs from spending more than they earn and not paying their debts. The punishments can include banning from tournaments or transfer market period bans for example Italian team AC Milan were handed a one-year ban from playing in European tournaments for breaching FFP regulations [8]. The introduction of FFP regulations has prevented club owners injecting massive sums of money into the club in order to gain advantage in the transfer market. An example of the pre-FFP financial advantage is when Chelsea was taken over by Roman Abramovich in 2003. His £700 million was the largest sum paid to a club by a single investor in football history [9]. In his first season alone, he spent £121.3 million on transfers, which was considerably more than the other Premier League clubs. This financial advantage had a direct link to success at Chelsea, before Abramovich's takeover, they had won 9 trophies in their history and since the takeover they have now won 16 additional trophies in the short time and have established themselves as a European powerhouse. Roman Abramovich's large investment has had a major role in Chelsea's recent success. The introduction of FFP has been seen as beneficial to prevent rich owners from injecting money into the clubs and spending masses on transfer fees, such as the case with Roman Abramovich and Chelsea.

2.2 Introduction to Machine Learning Techniques

This section will give an overview of popular machine learning techniques used for classification problems.

2.2.1 Supervised Learning

Supervised Learning involves the mapping between a set of input variables and an output variable and then using this mapping to predict the outputs of unseen data [10]. The output of the model is already fed into the system so the machine knows the output and the model created can form patterns to predict the outcome of new data. Supervised learning can be seen as two different problems, classification and regression. Classification problems involve labelling classes for a given example of input data. For example, if given some data, predicting if an email is spam or not. The two classes here are “spam” and “not spam’. Regression problems are used to predict a continuous output from the values of the predictor variables. Linear regression is typically used for prediction on continuous variables [11] such as predicting house prices given, population, schools in the area etc. An example can be seen in Equation 1.

$$y = b_0 + b_1x_1 + b_2x_2 + \ldots + b_zx_z + \epsilon$$

Equation 1 Linear Regression

Where:

- y is the observed value, e.g., house prices
- b_0 is the intercept
- x_m are the input variables, e.g., population, schools in the area
- ϵ is the residual error rate.
- b_m are the relative weights that are to be found

Logistic regression is used for classification problems where all the examples from the data are assigned to *K* different classes. Predicting football results can be seen as both a regression problem, if predicting the full-time score (e.g., 3-1 2-0, 0-1), and classification if predicting the outcome of the match, with the classes Win, Draw and Lose. This project aims to find the outcome of the football match and therefore, can be seen as a classification problem.

2.2.2 Random Forest

Random Forest is an ensemble learning method that is used for classification and regression and is used to create a collection of decision trees. For the classification problems each tree can be seen as a base classifier and this is done by majority voting where each decision tree votes for the class label that it predicts and the class with the most votes is used as the output class [12]. A Random Forest algorithm tends to give the highest accuracy among the classification methods available and is also useful for handling large datasets with a number of variables. A Random Forest classifier is known to work well with non-linear data and prevents overfitting, as well as being able to deal with outliers effectively. The disadvantage of a Random Forest is that it tends to be biased towards the attributes that contains more levels, which can result in unreliable variable importance scores.

Figure 2 shows how a simplified example of a Random Forest classifier where *B* decision trees have been combined and voting (for classification) or averaging (for regression) takes place for determining the output class of the forest. This project involves classification, therefore voting takes place.

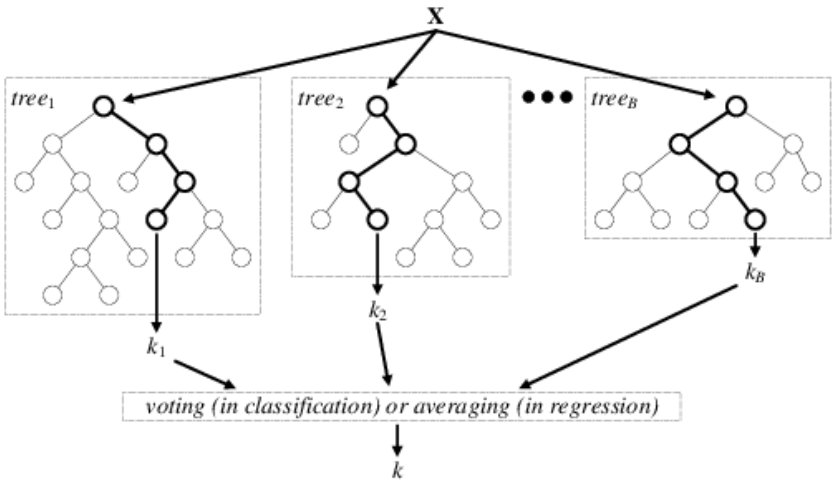


Figure 2 Random Forest Example Structure [13]

The advantages and disadvantages of a Random Forest can be seen in Table 1:

Advantages	**Disadvantages**
Good at solving regression and classification models.	Large number of tress can make the algorithm slow and not appropriate for real-time predictions.
Returns variable importance for each feature in the dataset.	Requires a lot of computational power.
Aggregates multiple decision trees to avoid overfitting and error.	Modelers have little control on how the model works as it is a predictive method but not a descriptive one.
It is tolerant to null and missing values.	
Does not require normalisation of data in the data preparation stage.	

Table 1 Advantages and Disadvantages of Random Forest

2.2.3 Neural Networks and Deep Learning

Neural networks, which are also known as artificial neural networks has the purpose to attempt to imitate the network of neurons in the human brain so that the machine will be able to learn the same way and make key decisions. Deep learning uses feature learning, to map input features to an output. The process is done in multiple layers known as hidden layers and with each layer containing multiple neurons [14]. Each neuron has the aim to learn the relationships between the other input neurons and the output. In a Deep Neural Network, feature data is passed from the neurons in one layer to the next layer during an iterative process.

Figure 3 illustrates a simple example of a Deep Neural Network where $\mathcal{X}$ is input variable and the probability of the class ω_c is returned from the network [15].

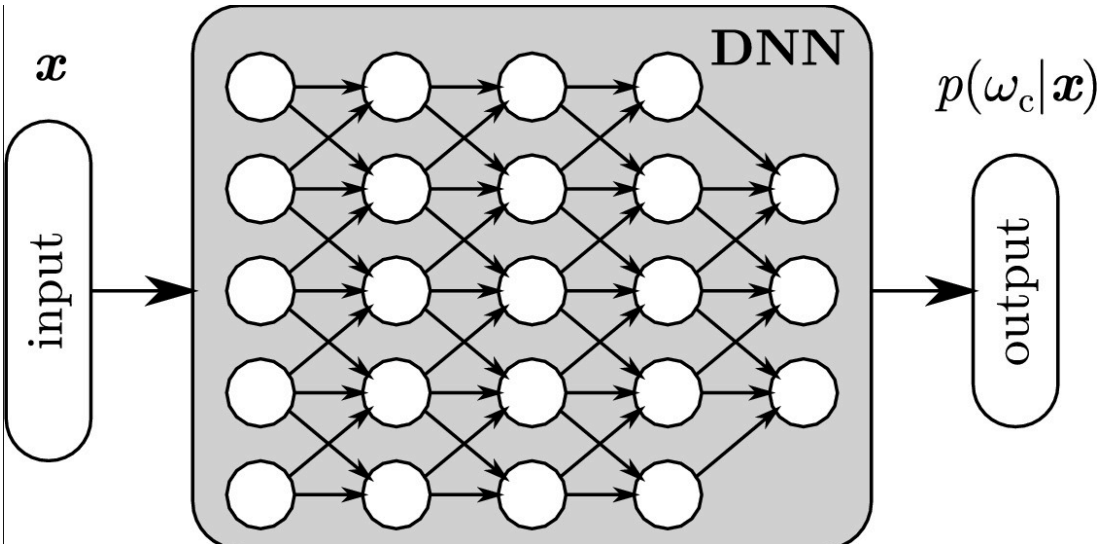


Figure 3 Example of a Deep Neural Network [15]

In a deep neural network, there are 6 mains steps [16].

1) **Initialisation** – the initial weights are set to each neuron in the network
2) **Forward Propagation** – the initial outputs for the network are calculated using the input data
3) **Error Function** – an error function is used to calculate the difference between the actual and desired output.
4) **Backpropagation** – weights are adjusted to bring the error value down to a minimum
5) **Weight Change** – the weights are changed to its optimal values as a result of the back propagation stage
6) **Iterate until convergence** – at each iteration, the weights are adjusted due to the gradient descent force and the error reaches a global loss minimum.

The mapping of the input to output is done by the learned weights of each neuron using a technique called backpropagation [17]. An error function is used to find the error for the overall network, this is the difference between the actual value and estimated error of a vector. The error is propagated backwards towards the input, this step can take a long time until the error is minimised.

Backpropagation example

This is a simplified example of backpropagation calculation from [16].

Consider the network:

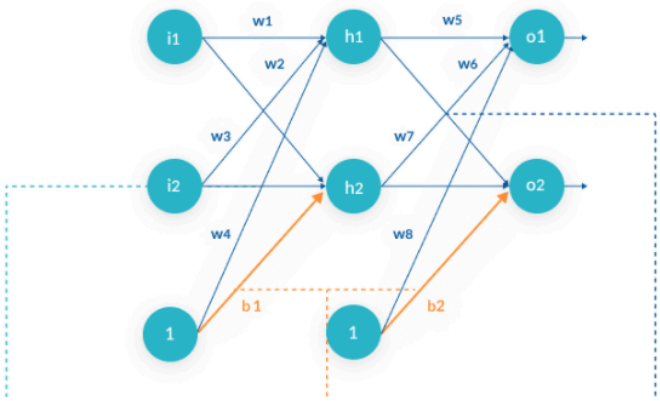


Figure 4 Example of Backpropagation in Neural Networks [16]

The example from Figure 4 has 2 input neurons, two hidden neurons and two output neurons.

The forward pass is done by multiplying the weights from the first layer (w1, w2, w3, w4) and applying the activation function for the hidden neurons h1 and h2. The output is the result of h1 and h2 multiplied by the second layer weights (w5, w6, w7, w8). The error function, e.g., Mean Squared Error (MSE) function is used to calculate the difference between the output and the desired output.

If the error from output o1 is 0.24 and 0.35 from output o2. The MSE(o1) is 1/2 $(0.24)^2$ and MSE(o2) is ½ $(0.35)^2$ and the total error is MSE(o1) + MSE(o2)
Backpropagation is the next step and calculates the partial derivative from the error function to the neuron with the specific weight. For example, hidden neuron h2 with weight w7 affects the output of o1. Therefore, the model calculates three derivatives:

- The total error compared to o1
- Output o1 compared with input of o1
- The input of o1 compared with hidden neuron h2 with w7

Using the Leibniz Chain Rule the error is minimised and the weight for the neuron is updated. The process is repeated until errors reach a global minimum.

During the backpropagation stage, gradient descent takes place, and its goal is to take steps to reach an optimal solution and aims to reach the minimum of a function. This is done by differentiating the error function. Problems may occur if the learning rate is too large too small. If there is a large learning rate, the gradient decent algorithm may miss the minimum and move away from it. If the step sizes are too small, the algorithm may become slow and take plenty of time to reach the minimum.

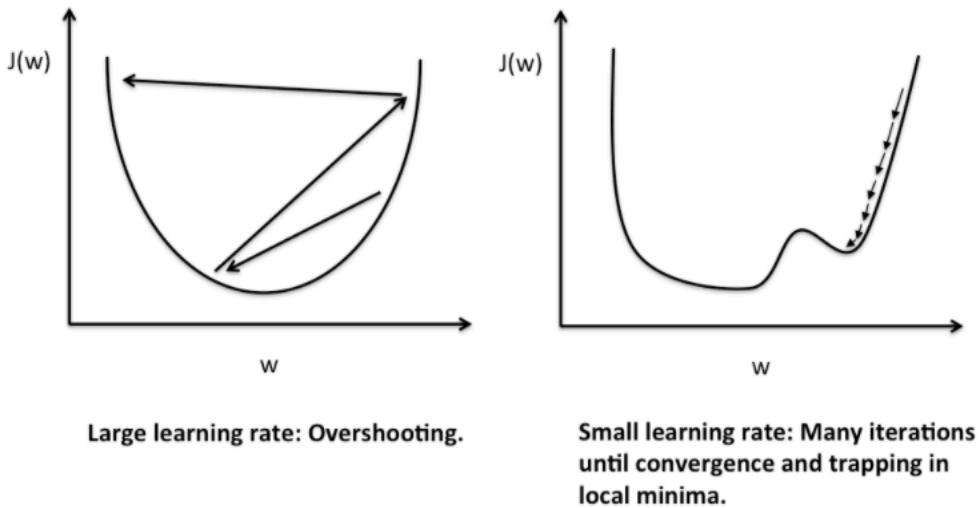


Figure 5 Gradient Descent Step Sizes [18]

Feature scaling can help to find the appropriate learning rate and can be done using standardisation as shown in Equation 2

$$x_j, std = \frac{x_j - \mu_j}{\sigma_j}$$

Equation 2 Standardisation

Where: μ_j is the mean of the input feature x_j and σ_j is the standard deviation of the feature.

Table 2 shows the advantages and disadvantages of the Neural Network

Advantages	**Disadvantages**
Can be used to build complex models.	Requires long training times
Results tend to be more successful than other machine learning models when complex models are created.	Tuning of hyperparameters such as neurons at each level, number of layers, activations functions can be difficult and complex.
AI model based on the way the human brain learns.	What is learned from the model may not be easy to understand.
Technique can be used in many different situations and applications such as image recognition.	Requires normalisation of data.
Scalable for large volumes of data.	

Table 2 Advantages and Disadvantages of NN

2.3 Machine Learning in Sports Prediction

Sports prediction has been explored by many researchers over the years and many techniques have been used in the process. This section of the literature review will explore the methods used for the predicting of sporting events, besides football, using machine learning techniques, namely, random forest and deep neural networks.

2.3.2 Neural Network

One of the initial studies for predicting the outcome of the National Football League (NFL) was completed by Purucker [19] in 1996. He used the matches of the 1994 NFL season to predict the outcome of 14 matches in week 16 and compared the use of supervised and unsupervised learning. Five input vector values for each team were used, these being yards gained, rushing yards gained, turnover margin, possession time margin and victories. Purucker used unsupervised learning to attempt to separate the winning teams from the losing team and a 10x10 neuron grid was used to implement a Self-Organising Map (SOM) network, which allowed the separation of strong and weak teams. A back propagation network was then utilised and was trained using targeted input vectors which contained the “exemplar” teams. These entailed the four best performing teams and the four worst performing teams in the NFL. The use of back propagation method correctly predicted 11 of the 14 results from week 16 and the SOM network only predicted 8 out of 14 accurately. The backpropagation used supervised methods and had more information than the SOM network, meaning that the results did not come as a surprise. The major limitation of this study was the small number of features that were used.

Purucker’s model was extended by Kahn [20] in 2003. From the 2003 season the data from weeks 1-14 were used which consisted of 208 NFL matches. The features used here were, total yardage differential, rushing yardage differential, time of possession differential, turnover differential and home or away factor. Weeks 14 and 15 were used a testing set and the weeks 1-13 were used as the training set. Kahn achieved a 75% success rate in the predictions for weeks 14 and 15. The model was tested against the findings of the ESPN sports casters who predicted and average of 72% across weeks 14 and 15 meaning that Khan achieved 3% more than the sports casters.

Neural networks have also been used to predict non-team sporting events. In 2010, Davoodi and Khanteymoori [21] used the data from 100 races to predict to predict the result of horse races using artificial neural networks. A different neural network was created for each horse and 8 features were used for each network. These were horse weight, type of race, horse’s trainer, horse’s jockey, number of horses in the race, race distance, track condition and weather. The features were normalised using Equation 3 to ensure all the data was within the same range where $\mathcal{X}_i$ is the feature.

$$\mathcal{X}_i = \frac{\mathcal{X}i - mean(\mathcal{X}i)}{var(\mathcal{X}i)}$$

Equation 3 Normalisation used by Davoodi and Khanteymoori [21]

Multilayer feedforward network was used as it minimised the Mean Square Error (MSE) and the network consisted of 4 layers. These were an input layer, two hidden layers (5 neurons in the first hidden layer and seven in the second) and an output layer where each output neuron corresponds to an output signal. 5 different algorithms were compared, these being Gradient Descent Backpropagation, Gradient Descent with Momentum, Quasi-Newton BFGS, Levenberg-Marquadt and Conjugate Gradient Descent. These were compared in multiple areas such as first and last position correctly predicted, 2 and more horses in actual results

predicted correctly and more. It was concluded that the backpropagation algorithms had the highest accuracy at 77%, however required a much larger training time.

2.3.1 Random Forest

A random forest model was used by Pretorius and Parry [22] in 2016, which was used to predict the results of the 2015 Rugby World Cup heled England and Wales. The model created was a classification model with two classes, whether the home team won or not. For the training set there were 379 historical matches and 21 matches for the testing set. Two different random forests were used. The first was a method called Forest-RI, which builds random forests from a random input selection and the trees are grown to their maximum size. The second method was oblique random forests, which uses a random linear combination of the input features. The number of trees was set to 200 and the size of the subset of randomly selected variables at each of the node splits was set to 1 after a grid search to determine the best values. The Forest-RI model had a faster training time and a lower test error. Out of the 48 matches in the 2015 World Cup, Forest-RI model correctly predicted 43 compared to the 41 correctly predicted by the OddsPortal and SuperBru companies. The limitation of the model is that there was a small sample size of only 48 matches, and the training data was data that was not from a World Cup. The World Cup would add pressure to the teams, and they would play their best available teams compared to a friendly where they may want to give lesser-known or less-experienced players a chance.

A model for predicting the win probability of NFL matches was also completed Lock and Newton [23]. The fixtures from the 2001 to 2012 seasons were used as the training data. The models had ten different predictor variables, including the time out remaining for each team, the total points scored and the yards to go for a first down. The randomForest R package was used to create the random forest and a cross validation strategy was implemented to find the best hyper-parameters. Based on the misclassification rates, Lock and Newton decided that the best node size was 200 and the number of predictors at each split was set to 2. The models showed that the most important variables on average across the 4 quarters of an NFL game were difference in score between the two teams and Las Vegas pre-game point spread. As this a regression model where probability is predicted, accuracy was measured using the Mean Squared Error (MSE) for each quarter and there was an average MSE of 0.157. Another way of measuring accuracy was to put the win probabilities in bins calculating the proportion of win in each of the bins. There was high correlation between the wins and the win probability at the centre of each bin showing that the estimates were successful.

2.4 Football Prediction Using Statistical Approaches

Football prediction has been attempted for a long period of time. Before the evolution of machine learning techniques, many researchers attempted to use statistical approaches in these predictions.

In 1982, Maher [24] investigated the Poisson Distribution with parameters representing the attacking and defensive strengths of each of the teams and used the distribution to predict the mean number of goals that the teams scored. An independent Poisson model for scores was adopted. A model as such was used

- If team x is playing against team y, the score of the match is (i_{xy}, j_{xy})
- Where i_{xy} is Poisson with a mean $a_x b_y$ such that a_x is the strength of team x's attack at home and b_y is the weakness of team y's defence away from home.

This model was enhanced in 1997 by Dixon and Coles [25] where the Poisson Distribution was used to predict the probabilities of different scores in addition to the full-time result. The model created is based on a Poisson regression model, the paper suggests the model created by Maher underestimated the likelihood of low scoring matches (1-0, 0-0, 0-1 and 1-1). For this an extended function was introduced to counter the lack of low scoring probabilities as shown in Equation 4.

$$(X_{i,j} = x, Y_{i,j} = y) = \tau_{\lambda,\mu}(x) \frac{e^{-\lambda}\lambda^x}{x!} \frac{e^{-\mu}\mu^y}{y!}$$

$$\text{Where: } \lambda = \alpha_i \beta_j \gamma \ and \ \mu = \alpha_j \beta_i$$

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

Equation 4 Poisson Model used by Dixon and Coles [2]

The Dixon Coles model also used a time decay factor, so that matches played closer to the time has more influence on the prediction than the games that are further away in time, which have less influence on the estimates. The attack and defensive rating were decided by using a Maximum Likelihood estimation, which calculates the likelihood of the data with a set of parameters and then finds the best combination of parameters that gives the best results. This model created is still seen a successful and is used by many researchers when comparing their models to previous literature.

In 2000 Kuypers [26] used information from the betting market to predict football results in and explore the efficiency of fixed odds betting in England. The model created had three main decision points, the bookmaker decides the odds, the putters decide on which result to bet on and nature decides the outcome of the game. Kuypers model also had three different possible outcomes, home win, away win or draw. Kuypers predicted the outcome of the matches (home win, away win, draw) rather than the scores as it made the modelling more simplified as less features are required.

2.5 Football Prediction using Machine Learning

This section will discuss the newer football prediction research that has been conducted using the advances of machine learning techniques.

2.5.1 General Machine Learning Techniques

In 2016, Adam [27] created a generalised linear model to predict the result of football matches in the European Championships tournament. The model was trained using matches from the previous year using a gradient descent. The researcher used two different categories of features, team-based and player-based. The team-based statistics included their European rank and their ELO rank (based on skills of players) and the player-based data contained the goals and transfer value of the players at the time. The downside of the team rank data is that the rankings change over the year and the rank before each match was not in the data, only the rank of the team just before the tournament started was used. Another limitation was the lack of features available.

Tavakol [28] was able to create a model that implemented more features to predict the results for the same tournament he was able to add the players previous matches. He also added a unique feature that listed the players from a country with the ranking of the club they play for to give the teams with players from a higher ranked club an advantage in the model. The model resulted in an average log loss error value of 1.3187. However, where there was historical data for countries that have played more than two games in the tournament previously the loss value reduces to 0.9680, showing that the more historical data available for each team, the more accurate the predictions are likely to be.

Dobravec [29] created a Naïve Bayes Classifier to predict the outcome of the 2014 FIFA World Cup held in Brazil. The researcher built a model that obtained features from a matrix factorisation process. This was used due to evidence in the field of multimedia recommender systems showing high accuracy when predicting user satisfaction [30]. The model is based on the number of goals scored by a team and their opposition in historical matches. The model for the matrix factorisation used is shown in Equation 5.

$$g_{T,O} = \mu + b_T + b_O + q_O^T p_T$$

Equation 5 Naïve Bayes Classifier used by Dobravec [29]

Where $g_{T,O}$ is the outcome of the match between *T* and *O*, b_T and b_O is the biases for each of the teams calculated so that $\mu + b_T$ represents the average goals scored of the team and $\mu + b_O$ is the average goals conceded by the opposition team. And $p\ and\ q$ represent the latent feature vectors. Stochastic gradient descent was used through the training data and the weights were adjusted using a MSE function. The accuracy of the model for the Naïve Bayes classifier was 50%, however the prediction for a win or loss was much more accurate than for a draw. This could be due to the lack of data used for the training, only 64 matches were used and very few features were used.

In 2018 Esme and Kiran [31] used data from the Turkish football league to predict the outcome of football matches based on odds set by bookmakers and using a k-Nearest Neighbour model. The dataset comprised of the matches between the 2010/11 and 2015/16 seasons, giving the researchers a large dataset at their disposal. The second half of the 2015/16 season was used for testing and the remaining data was used to train the model. 17 features were used for the model, with 10 based on betting odds and 7 based on statistically engineered features, including the cost of the team and consistency of the teams betting odds. The calculation in Equation 6 shows how the consistency feature is calculated feature by the

Euclidean distance between the probability of the betting odds and the frequency of the betting odds.

$$\|P_{odds} - P_{freq}\| = \sqrt{\sum_{i=1}^{n} (P_{odds_i} - P_{freq_i})^2}$$

Equation 6 Consistency Feature Calculation by Esme and Kiran [31]

The model created had an 57.52% accuracy on predicting the full-time result (win, lose, draw) which was better than the 55.56% predicted by the bookmakers. The model also took double chance into account, this is where the person betting can predict two of the three outcomes (e.g., the bettor bets that the result will be a loss or draw and not win). Here the model created had an accuracy of 86.27% compared to the 78.43% of the bookmaker's predictions. The KNN model was seen as successful due to the feature engineering that had been completed.

2.5.2 Random Forest

Random forests are a common technique that have previously been used to predict the outcome of football results. An example is in 2014, when Yezus [32] used multiple machine-learning methods, including Random Forest's on English Premier League data. The researcher split the features into two sets, static and dynamic features. The static features included the form, concentration and motivation of each of the teams. The dynamic features consisted of the goal difference at the time, the score difference and the history of the teams.

1. name1	2. name2	3. form1	4. form2	5. result	6. concentration1
name1	name2	form1	form2	result	concentration1
Манчестер Юнайтед	Манчестер Сити	1.0	0.6	0.0	1.0
Манчестер Юнайтед	Челси	0.6	0.7	0.0	1.0
Манчестер Юнайтед	Арсенал	0.8	0.5	1.0	1.0
Манчестер Юнайтед	Тоттенхэм Хотспур	0.8	0.6	0.0	1.0
Манчестер Юнайтед	Эвертон	0.9	0.7	1.0	1.0
Манчестер Юнайтед	Ливерпуль	0.9	0.6	1.0	1.0
Манчестер Юнайтед	Вест Бромвич Альбион	0.9	0.5	1.0	1.0
Манчестер Юнайтед	Суонси Сити	0.6	0.5	1.0	1.0
Манчестер Юнайтед	Вест Хэм Юнайтед	0.8	0.4	1.0	0.2

7. concentration2	8. goal_diff	9. score_diff	10. history	11. motivation1	12. motivation2
concentration2	goal_diff	score_diff	history	motivation1	motivation2
0.8	0.5757575757575758	0.6388888888888888	0.5	1.0	1.0
0.8	0.5704225352112676	0.6666666666666666	0.25	1.0	1.0
0.2	0.5625	0.6578947368421053	1.0	0.0	0.0
1.0	0.6333333333333333	0.6666666666666666	1.0	0.0	0.0
1.0	0.6727272727272727	0.7222222222222222	0.25	0.7692307692307692	0.9487179487179488
1.0	0.6764705882352942	0.7763157894736843	1.0	1.0	1.0
1.0	0.6829268292682926	0.6805555555555556	1.0	0.8771929824561403	0.9649122807017544
1.0	0.8	0.825	0.75	1.0	1.0
0.8	0.8493150684931507	0.8359375	0.5	0.0	0.0

Figure 6 Example Data used for Random Forest Model, Yezus [32]

Figure 6 shows an example of the dataset used for the Random Forest model. The model resulted in a 0.634 precision. Note that the first two columns are in Russian as the team as the team names were translated into Russian.

2.5.3 Neural Network

S.Mohamed Arabzad et al. in 2014 [33] used Artificial Neural Networks to predict the outcome of one week's worth of Iran Pro League football matches from the 2013-14 season using a training set of 2068 matches from previous seasons. The features for the model included Teams, condition of teams in recent weeks, condition of teams in the league and quality of previous opponents. The label for this model was the result of the match result. The model was a multi-layer perceptron comprising of 10 inputs and 2 hidden layers with 20 neurons at each layer and the network had a learning rate of 0.05. The network resulted in a high

accuracy and correctly predicted 5 teams that would finish the season in a top 6 position. However, the paper discusses that the limitation of the models, these include:

- The impact of other tournaments, resulting in players being rested.
- The club's investment (this will be a feature in this project) and average ages of the players
- The weather conditions
- The distance between matches, the referee and injured players.

2.5.4 Previous Literature Comparing NN and RF

There have been a few papers that have used both Neural Networks and Random Forests. In 2011 Hucaljuk and Rakipović [34] compared multiple machine-learning classification models for their accuracy and F1 score when predicting Champions League results. The features selected were the current form of the teams based of the last 6 matches, the outcome of the previous meeting between the two teams, the current rank of the teams, the number of players injured, and the average goals scored and conceded. The Neural Network was trained using backpropagation and consisted of 5 hidden layers. Out of the 6 models created, it was concluded that the Artificial Neural Network performed best for accuracy, with an average of 59.8% across 3 cross validation folds compared to the Random Forest's 51.7%. The Neural Network also had the best F1 score with an average of 0.604 compared to the Random Forest's 0.51. The overfitting of the trees in the Random Forest was predicted to be the downfall of the model and the reasoning behind its lower accuracy due to the relatively small training dataset of 96 matches.

In 2015, Tax and Joustra [35] used the following machine learning techniques to predict the results from the Dutch Eredivisie league:

- Naïve Bayes
- Logit Boost
- Neural Network (Multilayer Perceptron)
- Random Forest
- CHIRP
- FURIA
- DTNB
- Hyper Pipes

They compared a model containing only betting odds with hybrid models of historical matches and betting data and models that consisted of only historical data. The historical match features included performance in the season, the goals scored in previous encounters with the opposition team, the amount of national team players on the team and many more. The models had a total of 51 features when the betting odds of different companies were excluded. Tax and Joustra trained their models using a 10-fold cross validation experiment.

The Neural Network achieved the highest accuracy when predicting the results with an accuracy of 54.702% when only the historical match data was used, however this was marginally lower than the bookmaker's odds, which was 55.279%. The Random Forest was trained using 10, 50, 250 and 500 trees, and achieved an accuracy of 53.738% when. This performed better than the betting odds which was 53.351% and also resulted in a better accuracy than the hybrid model that was created.

In 2019, Pugsee & Pattawong [36] used the Random Forest Classifier to predict the outcome of some matches from the 2017-18 season. The training data included 1140 matches across the 2014-15, 2015-16 and 2016-17 seasons and a 10-fold cross validation technique was used along with a confusion matrix to evaluate the classification performance. Accuracy,

Precision and Recall were the three methods of measuring success in the model created. The averages were 68.71%, 48%, 47.53% for the accuracy, precision and recall respectively. When compared to a multi-layer perceptron model created, the random forest classifier performed better in all three measurements. The Neural Network performed better when predicting draws, however the forest was more accurate when it came to home and away wins.

The research conducted has been the basis for deciding which machine learning techniques to use for this Premier League prediction task. This project will compare the results of using Neural Networks and Random Forest to predict Premier League results. The model accuracies will also be compared when including the expenditure and income of each of the teams and discussing how including the transfer information will affect the model's evaluation metrics.

2.6 Football Prediction Including Transfer Data

The expenditure and income of teams during a season is not something that has been explored to a large extent, with the majority of researchers choosing to focus on features such as in game data and betting odds. The money teams spend on transfers can have an effect on performance as it may allow them to invest in high calibre players.

In 2018 Sujatha, Godhavari and Bhavani [37] created an Artificial Neural Network that was compared to the Football Result Expert System (FRES). The features for this model were the home teams' advantage, the league rank, the teams game statistics (e.g., wins, goals scored), the market value of the team, their current UEFA coefficient and the amount of money the team had spent on transfers. The model did not take into account the yellow and red cards that the team accumulated in the matches, which this project will consider. A backpropagation network was created with a single hidden layer containing 87 hidden nodes. The model also used logarithmic and exponential (augmented) neurons which were for improve the training of the network by modifying the weights. The model was evaluated by comparing the Root Mean Squared Error (RMSE) score of the model to the FRES error. The researchers concluded from their results that their model was better at predicting the goals than FRES, however the RMSE value was higher when predicting the winner. The limitation of this model was the lack of data used to test the models created. Only the matches from two teams were used and using more testing data would have resulted in better conclusions.

This project aims to explore the effect of using team's expenditure and income on predicting Premier League results and determine and evaluate how much these changes to the data affect the accuracies and other validation metrics.

3 Data Analysis

This section will contain all the information about the raw data collected, the location of the datasets and a data dictionary for each of the datasets. It will also explain the methodology used to prepare the datasets, including finding the key features and the pre-processing steps.

3.1 Data Collection

3.1.1 Historical Results Data

The first stage was the data collection from its various sources. For the historical data, a dataset from Kaggle was used. The dataset contained the English Premier League results from 1993-94 season to the 2019-20 season [38]. Some of the data was disregarded and this will be explained later in the report. Table 3 shows a data dictionary for the raw dataset.

Column	**Data Type**	**Explanation**
Season	Character	The season that the game took place
DateTime	Character	The Date and Time that the game took place
HomeTeam	Character	Name of the team playing at their home ground
AwayTeam	Character	Name of the team playing away from their home ground
FTHG	Numeric	The goals scored by the home team at the end of the match
FTAG	Numeric	The goals scored by the away team
FTR	Character	The full-time result (W, D, L)
HTHG	Numeric	Number of goals scored at half-time by the home team
HTAG	Numeric	Number of goals scored at half-time by the away team
HTR	Character	The half-time result (H, D, L)
Referee	Character	The name of the referee for the match
HS	Numeric	The number of shots taken by the home team
AS	Numeric	The number of shots taken by the away team
HST	Numeric	The number of shots on target by the home team
AST	Numeric	The number of shots on target by the away team
HC	Numeric	Number of corners obtained by the home team
AC	Numeric	Number of corners obtained by the away team
HF	Numeric	Number of fouls made by the home team
AF	Numeric	Number of fouls made by the away team
HY	Numeric	Number of yellow cards accumulated by the home team
AY	Numeric	Number of yellow cards accumulated by the away team
HR	Numeric	Number of red cards accumulated by the home team
AR	Numeric	Number of red cards accumulated by the away team

Table 3 Historical Results Raw Data

The raw dataset contained 10,659 rows and 23 columns. Each row in the dataset was equivalent to one match, therefore there was 10,659 different matches to base the models of.

3.1.2 Transfer Data

For the expenditure and income of the teams in the English Premier League, transfer data was exported from *Transfermarkt.com* [39] using the Table Capture Google Chrome add on.

This data was exported to Microsoft Excel and then saved as a CSV file to be used for the machine learning models.

The dataset created contained the following columns shown in Table 4

Column	Data Type	Explanation
Season	Character	The season the transfer information is relating to
Team	Character	The Team the transfer information is relating to
Expenditure	Character	The spending of the team on transfers
Arrivals	Numeric	The number of players that have been bought in the season
Income	Character	The income of the team from transfers
Departures	Numeric	The number of players that have left the club
Balance	Character	The expenditure – income for that season.

Table 4 Transfer Information Raw Data

The dataset once all the seasons were merged together contained 400 rows and 7 columns. This was for the 20 teams in each of the 20 seasons that the models were based on.

3.1.3 Transfer Inflation Data

The transfer inflation data was sourced from a table on *TotallyMoney* [40] that compared transfers to how much they would cost today. A table was made by dividing the columns Todays Money by Original Fee. For example, Figure 6 shows an example of a transfer from the 2000-01 season. To work out the factor needed to multiply the 2000-01 data the column *Today's Money* is divided by *Original Fee*. For this example, the factor is 59,703,398 / 23,400,000 = 2.5514.

Name	Country	Position	Age	Season	League	From	To	Original Fee	Today's Money
Rio Ferdinand	+	Centre-Back	22	2000-2001	Premier League	West Ham	Leeds United	£23,400,000	£59,703,398

Figure 7 Example of a Transfer Converted to Todays Money [39]

This gave us the factor that was needed to convert the expenditure and income of each team from each season into today's money. The dataset created has two columns as shown in Table 5.

Column	Data Type	Explanation
Season	Character	The season that the transfer information is related to
Inflation Rate	Numeric	The factor that the money needs to be multiplied by to be converted to today's money

Table 5 Transfer Inflation Raw Data

The dataset contained 20 rows and 2 columns. The 20 rows were for each season that the model would be created for.

3.2 Data Exploration

The transfer inflation dataset was used to convert the money clubs spent in previous year to the today's money. The general trend from Figure 8 shows that the further back in seasons we go, the more we have to multiply the transfer data by.

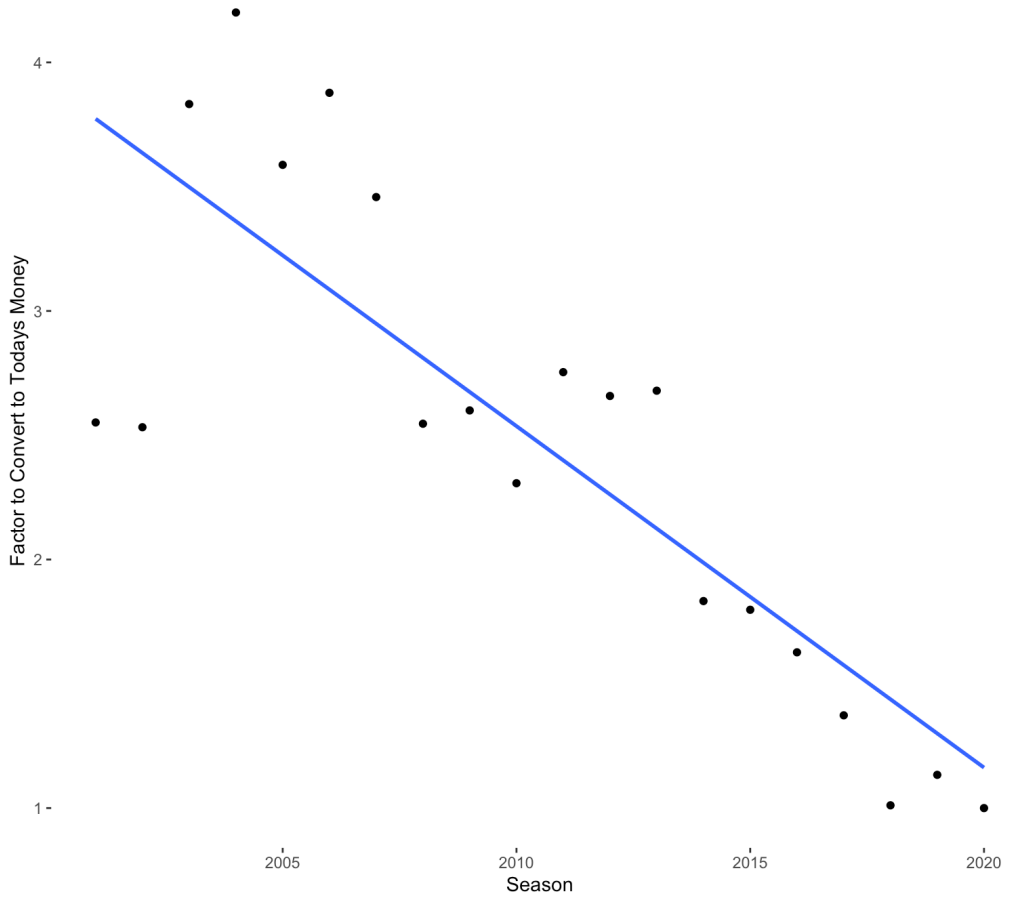


Figure 8 Factor to Convert Transfers to Todays Money

The season in which the transfer data is furthest away from today's money is 2003-04 and the closest season is 2017-18. This is largely due to the most expensive transfer in the world taking place in that season; Neymar's transfer to Paris Saint-Germain from FC Barcelona for the large sum of £199,800,000.

If inflation is not taken into consideration, there are clear correlations that can be seen between expenditure and the net spend of clubs in the English Premier League. This is largely due to teams spending more and clubs having more money available to spend on transfers every season. Figure 9 shows two graphs that show that over the years the expenditure of each the teams has generally increased, and the net income of each team has decreased showing that the teams spending has increased every year.

Teams are generating more income in different ways that do not include income from transfers. The increased TV revenue broadcasting rights is increasing all the time, the tournament prize money has increased almost every year and the sponsorships that clubs are getting from companies such as Adidas or Nike allows the teams to generate a large amount of revenue.

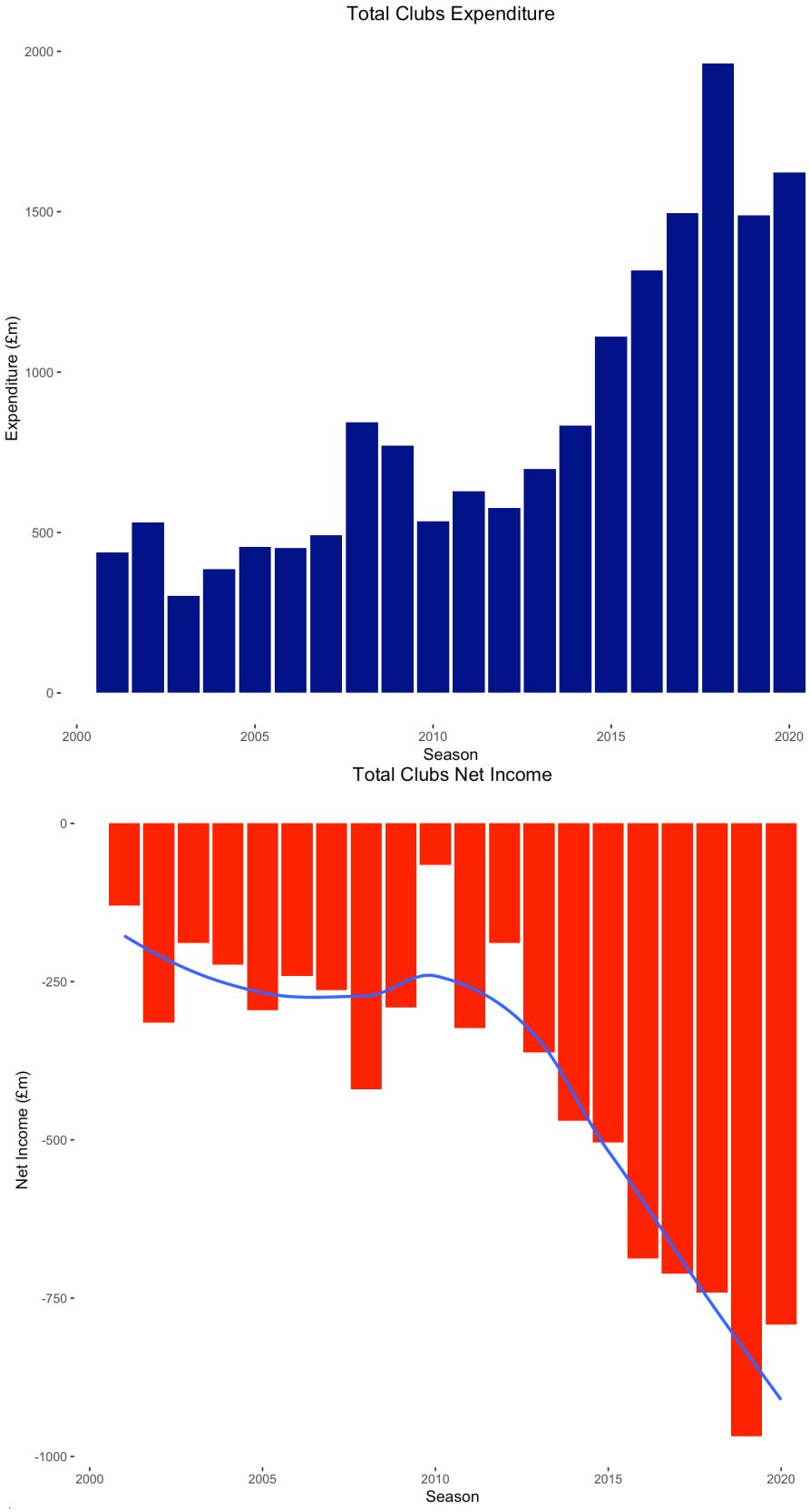


Figure 9 Expenditure and Net Spend of Teams

The trends shown in Figure 9 are a general trend for the English Premier League over the years. However, these trends may vary from team to team. As can be seen in Figure 10, the

general trend for West Ham is that they have in general spent more on transfers every year and this is due to the team being able to securely stay in the Premier League every year without fearing relegation. The higher place finishes have given West Ham more prize money every year and has allowed them to spend more on improving their squads each year.

A different situation also shown in Figure 10 is Liverpool's net income spend over the years. Their lack of qualifying for the Champions League (requires being placed in the top 4 positions in the Premier League) had seen their revenue drop by a large amount. This was shown in the trends where they spent less on transfers each season and even made a profit on transfers in 2017 and 2018, eventually qualifying for the Champions league in 2017 for only the second time in the last eight years. Their success in the 2018 and 2019 Champions League (reaching the final in 2018 and winning the competition in 2019) and their second-place finish in the Premier League enabled the owners to spend more in the transfer window. this is shown in Figure 10, where Liverpool's net spends to be £127 million. This was also partly helped to the sale of Phillipe Coutinho in 2018 to Barcelona for £130.5 million.

The two graphs in Figure 10 show that different circumstances can lead to clubs spending different amounts of money each year, however Figure 9 shows us that the trends in the league point towards teams spending more each season.

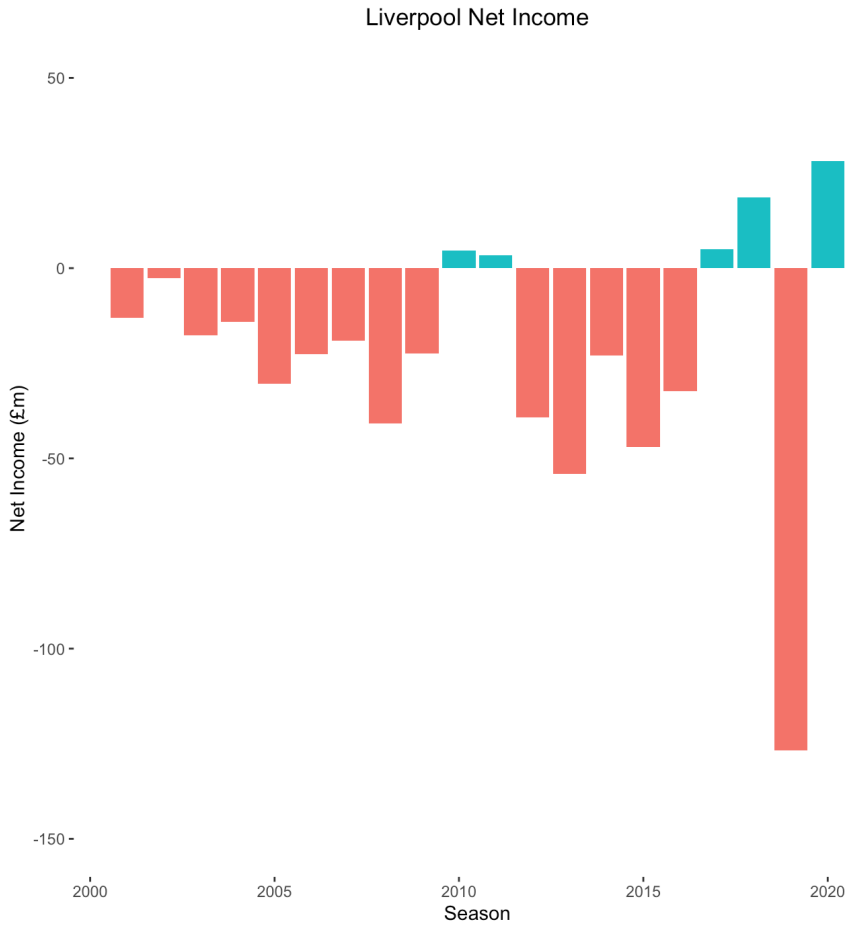


Figure 10 West Ham and Liverpool Net Income Comparison

For the historical results dataset, the only data available for the matches that took place from the 1993/1994 season to the 1999/2000 season were full-time goals as show in Figure 11.

Season	DateTime	HomeTeam	AwayTeam	FTHG	FTAG	FTR	HTHG	HTAG	HTR	Referee	HS	AS	HST	AST	HC	AC	HF	AF	HY
1993-94	1993-08-14T00:00:00Z	Arsenal	Coventry	0	3	A	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1993-94	1993-08-14T00:00:00Z	Aston Villa	QPR	4	1	H	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1993-94	1993-08-14T00:00:00Z	Chelsea	Blackburn	1	2	A	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1993-94	1993-08-14T00:00:00Z	Liverpool	Sheffield Weds	2	0	H	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1993-94	1993-08-14T00:00:00Z	Man City	Leeds	1	1	D	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1993-94	1993-08-14T00:00:00Z	Newcastle	Tottenham	0	1	A	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1993-94	1993-08-14T00:00:00Z	Oldham	Ipswich	0	3	A	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1993-94	1993-08-14T00:00:00Z	Sheffield United	Swindon	3	1	H	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Figure 11 Raw Data Example with Missing Data

The missing columns from the dataset are some of the key features that were used in the model and would have an effect on the results being produced. Therefore, it was decided that the data used for this project would be the matches beginning from the 2000/01 season to the 2019/20 season and the matches with the missing data would not be considered for the model.

Figure 12 shows the correlation plot between the features in the data set to determine what features are most likely to determine the outcome of a football match when the teams are not considered. From the figure it can be seen that the shots and corners are highly correlated, this was expected as most corners in a football match are won through shots on the goal or

from attempted crosses. For the full-time result, we can see that the features that are mostly correlated to this are home and away shots and shots on target which was expected. The red cards for both teams also seemed to have an effect on the full-time result, which can be explained by the fact that playing with less players would the team a disadvantage and hence the opposition team have more chance of winning the game. The fouls, yellow cards and corners for both teams were the least correlated to the full-time result and showing that players on the pitch and shots the team take are key when predicting the outcome of a football match.

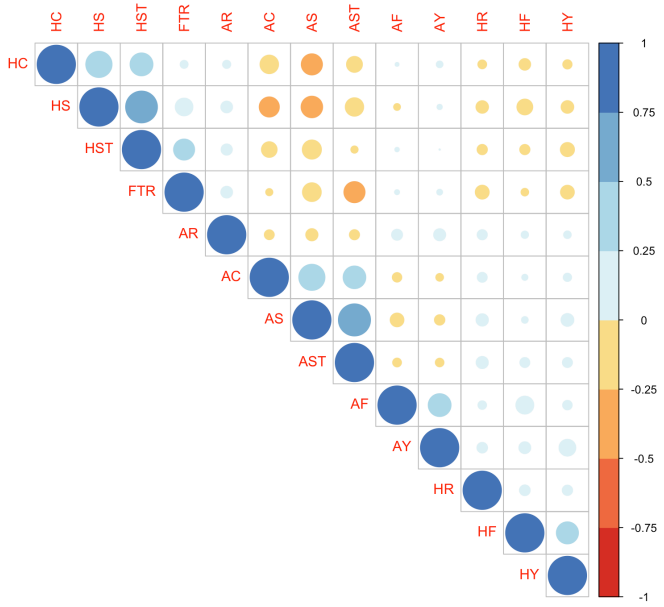


Figure 12 Correlation Plot of Results Data

Figure 13 shows the number of occurrences of each of the results from the historical dataset. The number home wins are much more than the number of away wins or draws, showing that the home team is an important feature as they have a higher chance of winning. This could be down to playing in familiar conditions and in front of their own fans. 42.9% of the games resulted in a home win and this also shows us that the dataset used is an unbalanced dataset due to the nature of home advantage.

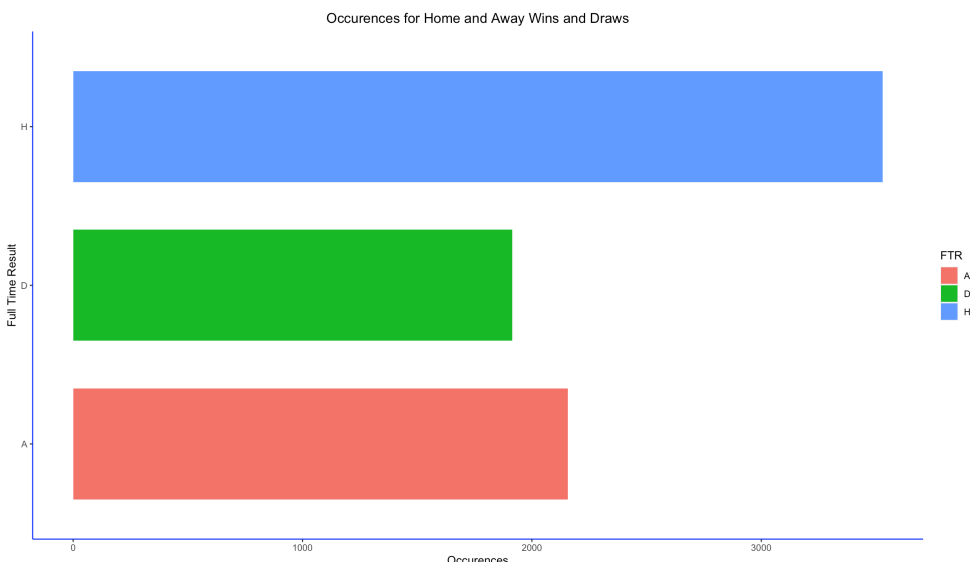


Figure 13 Occurrences of Home or Away Wins or Draws

3.3 Combining the Datasets

The input for the models would require one dataset to be provided and this was done by combining the three datasets. For this step, multiple functions were created to allow the merging of the datasets.

- *fix_transfer_names()* – the names of the teams were different in the historical dataset and the transfers dataset. This function adjusted the names in the transfer dataset, so they were identical. For example, *Arsenal FC* in the transfer dataset and *Arsenal* in the historical dataset.

- *adjust_transfers_money()* – used the inflation dataset and the transfers dataset to combine them by multiplying the expenditure and income of each team for each season by the increase factor.

- *results_and_transfer_combined()* – combines the transfer dataset that has been adjusted for inflation to the historical dataset. The feature from the transfer data that had been kept was the balance (Net Income) of each of teams to be used in the machine learning models.

Once the datasets were combined there was a total of 7600 games with 25 features. An example of the combined dataset can be seen in Figure 14. This dataset was then pre-processed to be ready for the machine learning models.

Season	HomeTeam	DateTime	AwayTeam	FTHG	FTAG	FTR	HTHG	HTAG	HTR	Referee
2020	Liverpool	2019-08-09	Norwich	4	1	H	4	0	H	M Oliver
2020	Bournemouth	2019-08-10	Sheffield United	1	1	D	0	0	D	K Friend
2020	Burnley	2019-08-10	Southampton	3	0	H	0	0	D	G Scott
2020	Crystal Palace	2019-08-10	Everton	0	0	D	0	0	D	J Moss
2020	Tottenham	2019-08-10	Aston Villa	3	1	H	0	1	A	C Kavanagh
2020	Watford	2019-08-10	Brighton	0	3	A	0	1	A	C Pawson
2020	West Ham	2019-08-10	Man City	0	5	A	0	1	A	M Dean
2020	Leicester	2019-08-11	Wolves	0	0	D	0	0	D	A Marriner
2020	Man United	2019-08-11	Chelsea	4	0	H	1	0	H	A Taylor
2020	Newcastle	2019-08-11	Arsenal	0	1	A	0	0	D	M Atkinson
2020	Arsenal	2019-08-17	Burnley	2	1	H	1	1	D	M Dean
2020	Aston Villa	2019-08-17	Bournemouth	1	2	A	0	2	A	M Atkinson
2020	Brighton	2019-08-17	West Ham	1	1	D	0	0	D	A Taylor
2020	Everton	2019-08-17	Watford	1	0	H	1	0	H	L Mason
2020	Man City	2019-08-17	Tottenham	2	2	D	2	1	H	M Oliver
2020	Norwich	2019-08-17	Newcastle	3	1	H	1	0	H	S Attwell

HS	AS	HST	AST	HC	AC	HF	AF	HY	AY	HR	AR	HomeNetSpend	AwayNetSpend	Pos
15	12	7	5	11	2	9	9	0	2	0	0	28080000	-5950000	1
13	8	3	3	3	4	10	19	2	1	0	0	-19760000	-64040000	18
10	11	4	3	2	7	6	12	0	0	0	0	-9270000	-28170000	9
6	10	2	3	6	2	16	14	2	1	0	1	43000000	-36450000	14
31	7	7	4	14	0	13	9	1	0	0	0	-75600000	-140490000	6
11	5	3	3	5	2	15	11	0	1	0	0	-20340000	-59800000	19
5	14	3	9	1	1	6	13	2	2	0	0	-56000000	-79670000	16
15	8	1	2	12	3	3	13	0	2	0	0	-14220000	-93330000	5
11	18	5	7	3	5	15	13	3	4	0	0	-131040000	101040000	4
9	8	2	2	5	3	12	7	1	3	0	0	-33530000	-96080000	13
16	18	9	5	10	7	13	11	2	1	0	0	-96080000	-9270000	8
22	12	7	4	10	5	10	13	0	2	0	0	-140490000	-19760000	17
16	8	4	3	8	6	11	10	0	2	0	0	-59800000	-56000000	15
12	8	2	2	4	7	11	11	2	3	0	0	-36450000	-20340000	12
30	3	10	2	13	2	14	4	1	0	0	0	-79670000	-75600000	2
15	10	8	3	7	5	9	11	1	3	0	0	-5950000	-33530000	20

Figure 14 Model Data Once 3 Datasets Were Combined

3.4 Key Features of the Dataset

3.4.1 Removing Redundant Fields

The dataset contained key features that would have a major effect on the model, however there were some features that were not required and would affect the model.

- FTHG – The full-time home goals were not required as they would have a bias on predicting the results alongside with the away goals.
- FTAG – For the same reason as the FTHG, this feature was removed.
- HTHG – The half time goals were removed as the goals are the main feature that is required when predicting the outcome of a football match and not needed in the model.
- HTAG – This was removed for the same reason as the HTHG feature.
- HTR – The half time result was removed from the dataset as the result and goals was required to be included in the model.
- Referee – There were 153 different referees in the dataset, and some would have only refereed a small number of games, and some may have retired, therefore it was decided that this feature would not be included in the machine learning models.

3.4.2 Feature Engineering

A feature was added to the dataset which had the position of the team at the end of the season the data was associated with. This was done by working out the final league table using the points system (3 for a win, 1 for a draw and 0 for a loss). The position feature was added as it would include a bias towards the better teams that would usually perform better in the Premier League and would have a positive effect on the models created.

3.5 Data Pre-Processing

The data pre-processing is an important step that is required before the models can be built. This is done to make the datasets into a useable format for the training and testing phases. Data pre-processing has multiple stages that need to be considered when preparing the data for models. These include the data aggregation and determination of the field attributes, the normalisation of the data, dimensionality reduction, feature selection and construction and checking for missing data and outlier values.

The Random Forest and Neural Network required to be pre-processed in different ways in order to optimise the model. Therefore, it was decided to create two different R scripts, *data_preprocessing_rf.R* and *data_preprocessing_NN.R*, to prepare the data for its respective models.

3.5.1 Random Forest Preparation Steps

This section will outline the steps taken to prepare the data for the Random Forest model.

1. The first step was to combine the historical results, transfer and transfer inflation datasets as discussed in section 3.3.

2. The next stage was to remove the data from the 1993/94 to 1999/00 seasons due to the data only containing the full-time results of each match and no relevant features.

3. The position of the team in the Premier League column was added to the dataset.

4. The redundant columns were removed from the data that were described in section 3.4.

5. If the model to be created was to not include the transfer data, the net spend columns of the home and away team were removed. Otherwise, this step was skipped.

6. The next step was to split up the numeric and symbolic fields. This was done so that the numeric fields could be split into ordinal or discreet fields. Figure 15 shows an example of the data split with the three different fields

```
 [1] "ORDINAL"  "SYMBOLIC" "SYMBOLIC" "DISCREET" "SYMBOLIC" "ORDINAL"  "ORDINAL"  "ORDINAL"  "ORDINAL"  "ORDINAL"
[11] "ORDINAL"  "ORDINAL"  "ORDINAL"  "ORDINAL"  "ORDINAL"  "DISCREET" "DISCREET" "ORDINAL"  "ORDINAL"  "ORDINAL"
```

Figure 15 Ordinal, Discreet and Symbolic Data Split

7. The discreet data was tested for outliers and this was replaced with the mean where relevant.

8. The discreet and ordinals dataset was then combined, and this was tested for fields that had a variance of 0, meaning that the values in a column were all the same. This was not the case for any of the fields

9. Then, the columns with high correlation were removed and the symbolic dataset was combined and checked to ensure that there was no missing data in the new dataset. The combined datasets were then ready to be used in the Random Forest modelling.

3.5.2 Neural Network Preparation Steps

The steps for pre-processing the data were similar to the random forest pre-processing stage, however there were two major changes. These were that symbolic fields are not used in Neural Networks and needed to be converted to numeric. The other main stage was that data was normalised between 0 and 1 at the end of the pre-processing stage so that all the data was within the same range. Normalising the data allows the model to learn the optimal parameters faster for each of the features and un-normalised data can increase the loss of the network. Random Forests do not require scaling of the data as each tree is splitting the features up and not comparing the features values to each other. If the features in a Random Forest are normalised the results will be the same.

The steps of the Neural Network data preparation are shown below.

1. The first step was to combine the historical results, transfer and transfer inflation datasets as discussed in section 3.3.

2. The next stage was to remove the data from the 1993/94 to 1999/00 seasons due to the data only containing the full-time results of each match and no relevant features.

3. The position of the team in the Premier League column was added to the dataset.

4. The redundant columns were removed from the data that were described in section 3.4.

5. If the model to be created was to not include the transfer data, the net spend columns of the home and away team were removed. Otherwise, this step was skipped.

6. The next step was to split up the numeric and symbolic fields. This was done so that the numeric fields could be split into ordinal or discreet fields.

7. The symbolic fields were converted to numeric fields using One-Hot Encoding (OHE). This gave every symbolic field a different option. Figure 16 shows an example of the new fields created by the OHE. The row where *HomeTeamChelsea* is 1, would be where the *HomeTeam* column in the original data set would be equal to *Chelsea* and all other columns in the new dataset beginning with "*HomeTeam*" would be 0.

HomeTeamCharlton	HomeTeamChelsea	HomeTeamCoventry
0	0	0
1	0	0
0	1	0
0	0	1

Figure 16 One-Hot Encoding Example

8. The discreet data was tested for outliers and this was replaced with the mean where relevant.

9. The discreet and ordinals dataset was then combined, and this was tested for fields that had a variance of 0, meaning that the values in a column were all the same. This was not the case for any of the fields

10. Then, the columns with high correlation were removed and the completed dataset thus far was checked to ensure that there was no missing data in the new dataset.

11. The final stage was the normalisation of the data. For this a function was created that used an equation to standardise the data as show in Equation 7, where x is the feature and x' is the normalised feature with a value between 0 and 1.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Equation 7 Data Normalisation

12. The pre-processed dataset was then ready to be used for the Neural Network modelling.

3.6 Splitting of Training and Testing Data

For the models, training data was required to train the machine learning models. Testing data was also required in order to test the model and evaluate its results. This was done using a method called stratified cross validation in the R script *kfold_functions.R*. Using a holdout method and splitting the dataset, for example 80:20 (80% of the data on training and 20% on the testing of the model), would result in conclusions that are not convincing. Stratified cross validation involves splitting the dataset into k folds so that each fold has the same proportion of data. The steps involved can be seen below.

1. The dataset is mixed up randomly and each row is assigned a number from 1 to k with k being the number of folds to use
2. Data is split up into k groups
3. For each of the groups, take a partition as the testing data.
4. The rest of the data is used for the training of the model.
5. Run each of the models and evaluate the model using the testing dataset. Do this for each of the folds.
6. The final performance of the model is calculated as the mean of the k evaluations

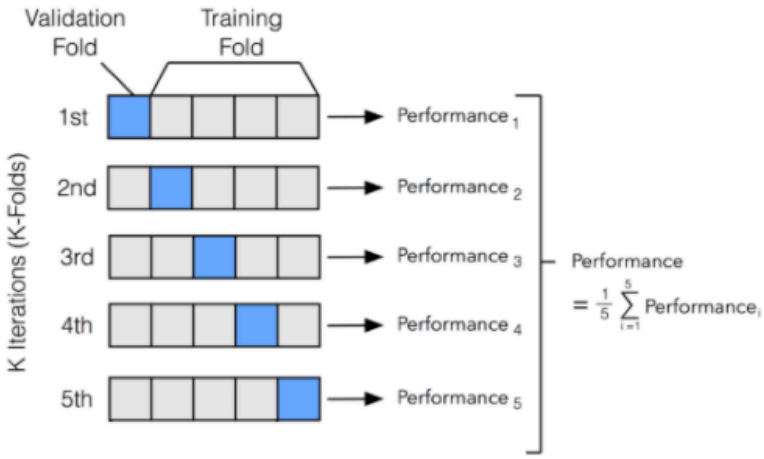


Figure 17 Example 5-Fold Stratified Cross Validation [41]

Figure 17 shows an example of a 5-Fold stratified cross-validation and shows the validation folds being different for each of the iterations and the overall performance is measured by averaging each of the evaluation measures from the 5 models.

The R script, *kfold_functions.R,* contained 3 functions. One function was for assigning the k value to each row of the dataset, one for creating the training and testing data from each of the data split in the k folds. The final function was used for training each of the models on the k folds and averaging out the performance metrics to return a list of the overall evaluations for the models. It was decided that 8 folds would be used for creating the machine learning models, as it was a divisor of the training data and this would mean that each fold would contain the same amount of testing data. If too large a k value was chosen, there would be the risk of overfitting the model.

4 Model Choices

This section will outline the models created for this project. Both machine learning techniques had two different models created depending on if the transfer data was to be used in the model or if it was to be excluded. This resulted in four models in total being created.

Once the cleaning and manipulating of the datasets was completed, the machine learning models could be created. This section will discuss how the models' parameters were decided and the techniques used to create, save and use the created models.

The aims of these models were to create models that would result in a high accuracy but would not overfit the data. If the training data was going to overfit the data, the accuracy would be close to 100%, however, the results from the testing would be poor and the model would not generalise well. Another problem is underfitting, where the model cannot generalise to new data and the model performs poorly on the training data. Underfitting also means that the models that have been created cannot find a relationship between the input variables and the output variables. Igea [42] discusses the importance of finding a model that generalises the data the best and find the best trade-off between overfitting and underfitting as shown in Figure 18.

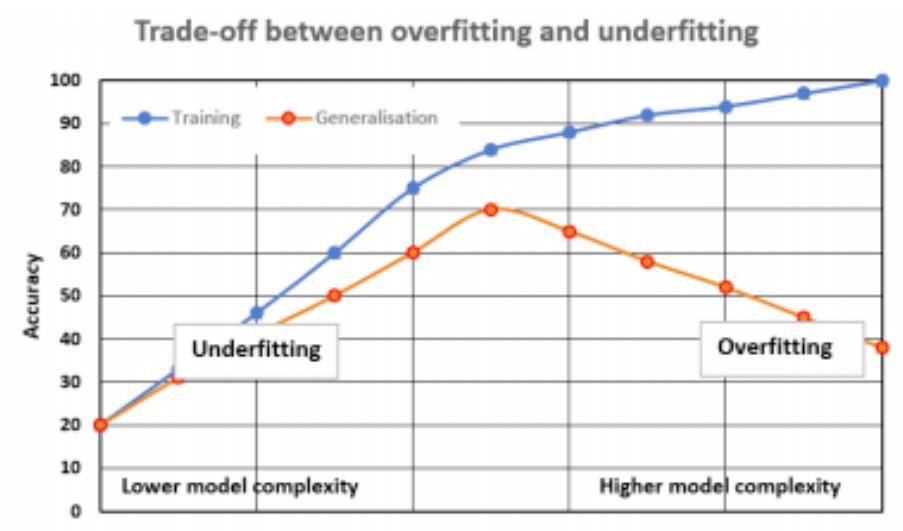


Figure 18 Trade-Off Between Overfitting and Underfitting [42]

The following sections describe the taken when creating the Neural Networks and Random Forest models. These models are created to prevent overfitting and underfitting and to try and achieve the best results for the football prediction. The choice of using the Neural Network and Random Forest models were based of the previous research that had been conducted and determining which models had the highest chance of success from the research.

4.1 Grid Search

Grid search is a method when creating machine learning models that can be used for determining the optimal hyperparameters for the models. It is a way of hyperparameter tuning that creates combinations of models in a specified grid. This is done by the programmer defining the range of values for each of the relevant hyperparameters and the model will run with different combinations of the hyperparameters. This results in multiple models being created and the best model is chosen depending on the user preference. The downside of using grid search is the long time and high computation [43] that it uses, however it was

chosen for this project as it was a method of finding the best hyperparameters and would result in the best possible models created.

4.2 Neural Network

The first model was the Neural Network, for this the h2o [44] *R* package was used for the training and testing. The h2o package was developed and is maintained by h2o.ai. The library is used for parallelised implementations of many supervised and unsupervised models. It is a package that is used by over 18,000 organisations, including banks, insurance and healthcare companies, and is used by over 80,000 data scientists. The h2o package requires that there is an up-to-date version of Java running on the machine. h2o specialises at in-memory processing to support large datasets which enables efficiency without the sacrifice of accuracy. When using the library, it must be initialised first to check that the local h2o package is the same version as the h2o running on the server. The deep learning models created by the h2o package are based on a feed forward network that is trained with stochastic gradient descent using back propagation [45].

In order to find the best parameters for the model a grid search was used. The h2o library has a function that allows the ease of performing a grid search. Grid search was used to determine the optimal hyperparameters by creating multiple models. The training and testing datasets must be converted to h2o objects prior to being used in the model and the feature variables split from the label. The model was trained with four different types of activation functions, *Rectifier, RectifierWithDropout, Maxout and MaxoutWithDropout.* Different combinations of hidden layers and L1 and L2 regularisation. With the combination of hyperparameters, a total of 216 models were created each time and the best model was selected by using the model with the lowest error rate on the training data. The parameters chosen for the model that include the expenditure and income of each of the teams are shown in Table 6 and the parameters for the without transfer data models are shown in Table 7.

K-Folds	Activation	Hidden Layers	Epochs	L1	L2
1	Maxout	5 5	100	N/A	0.001
2	Maxout	5 5	100	N/A	0.001
3	Maxout	5 5	100	0.001	0.001
4	Maxout	5 5	100	0.001	0.0001
5	Maxout	5 5	100	0.0001	N/A
6	Maxout	5 5	100	0.001	0.0001
7	Maxout	5 5	100	0.001	0.001
8	Maxout	5 5	100	0.001	0.0001

Table 6 Parameters for NN With Transfer Model

K-Folds	Activation	Hidden Layers	Epochs	L1	L2
1	Maxout	5 5	100	0.0001	N/A
2	Maxout	5 5	100	0.001	0.001
3	Maxout	5 5	100	0.001	N/A
4	Maxout	5 5	100	0.001	0.001
5	Maxout	5 5	100	0.001	0.001
6	Maxout	5 5	100	0.0001	0.0001
7	Maxout	5	50	N/A	N/A
8	Maxout	5 5	100	N/A	0.0001

Table 7 Parameters for DNN Without Transfer Model

This step was done for each of the 8 folds and the best model that was determined at each iteration was used to evaluate the testing data. It can be seen that the Maxout activation function was used for all the models and 2 hidden layers and 100 epochs were the most common. The result of this was 8 models that were optimised for the lowest error. The evaluation metrics of each of the networks were averaged to get the overall metrics of the Deep Neural Networks created to be compared with the Random Forest Models.

The creation, running of the grid search and saving of the models to be used was done in the R Script *myOwnNeuralNetwork.R*. The saved models are the ones that are to be used in the R Shiny user interface. The R scripts contained four main functions as shown below:

- *runNeuralNetwork*() – This was used to run the grid search and save the best model from the grid search and loading the models if necessary.
- *getConfusionMatrix*() – This was used to get the confusion matrix from the models and the relevant values including True Positive, False Positive, Precision, Recall and F1-Score, which are all explained in Chapter 5.
- *evaluateNeuralNetwork()* – This was used to get the extra metrics from the data such as accuracy, R^2 and error rates, which are also explained in Chapter 5.
- *getMetricsofNN()* – This function, takes the raw datasets, pre-processes the data and runs the models. It returns the metrics from the functions above.

4.3 Random Forest

For the Random Forests models created, a type of grid search was also conducted to determine the best parameters for the models. A function was created to determine the best parameters, these were *ntree*, *mtry* and *maxnodes.*

- ntree – the number of decision trees to grow in the forest.
- mtry – the number of variables that are sampled at each tree split.
- maxnodes – the maximum number of terminal nodes in the forest.

For determining the optimal hyperparameters, the *caret* R package was used and for training the model on the 8 folds the *randomForest* library was used.

The first stage was setting the train control to specify the number of parameters and folds to set. As 8 folds were used for this the number was set to 8 with the method set to cross validation. The output from this is passed as an argument when determining the hyperparameters.

An initial random forest was run using a grid of *mtry* values from 1 to 20 and *ntree* at 300. From this it was concluded that the best value for the number of variables sampled at each of the tree splits was 10. When searching for the best maxnodes, a grid with the maxnodes from 5 to 15 was made and trained using the optimal mtry value found in the previous step. The models were looped around, and the results were summarised to get the accuracy and kappa value. Figure 19 shows the accuracy of each of the models when the maxnodes value was changed.

```
Models: 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
Number of resamples: 8

Accuracy
        Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
5  0.5187970 0.5272879 0.5338346 0.5364636 0.5433736 0.5639098    0
6  0.5172932 0.5312030 0.5383380 0.5383439 0.5477444 0.5563910    0
7  0.5187970 0.5323308 0.5432637 0.5385327 0.5471280 0.5488722    0
8  0.5218045 0.5394737 0.5447685 0.5432308 0.5512955 0.5548872    0
9  0.5278195 0.5345865 0.5462746 0.5443592 0.5535511 0.5593985    0
10 0.5323308 0.5372180 0.5496242 0.5449248 0.5507519 0.5533835    0
11 0.5278195 0.5372180 0.5470265 0.5441715 0.5520479 0.5548872    0
12 0.5278195 0.5372180 0.5485325 0.5447360 0.5527603 0.5548872    0
13 0.5278195 0.5390977 0.5455238 0.5445469 0.5533835 0.5570571    0
14 0.5248120 0.5379699 0.5484590 0.5441730 0.5502440 0.5548872    0
15 0.5278195 0.5447368 0.5515423 0.5477433 0.5533835 0.5570571    0
```

Figure 19 Accuracy Measurements When maxnodes Was Changed

From Figure 19 it was determined that the optimal maxnodes was 5 and that was used for the Random Forest.

Now that the maxnodes and ntree optimal values had been determined, the next step was to get the best number of trees for the model. This was done in a similar way as determining the maxnodes. A grid of a number of trees was created with the maxnodes and mtry value that were found to be the best for the data and the results were summarised to get the best number of trees. This was found to be 1000 as show in Figure 20.

```
Models: 250, 300, 350, 400, 450, 500, 550, 600, 800, 1000, 2000
Number of resamples: 8

Accuracy
          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
250  0.5187970 0.5300493 0.5353383 0.5394726 0.5545113 0.5578947    0
300  0.5172932 0.5308023 0.5353383 0.5400370 0.5567669 0.5578947    0
350  0.5157895 0.5315519 0.5353383 0.5392846 0.5541353 0.5578947    0
400  0.5157895 0.5317765 0.5349364 0.5392840 0.5530075 0.5609023    0
450  0.5172932 0.5319250 0.5368421 0.5392838 0.5500000 0.5593985    0
500  0.5157895 0.5311754 0.5383459 0.5396603 0.5522556 0.5578947    0
550  0.5157895 0.5319284 0.5390977 0.5396608 0.5503759 0.5578947    0
600  0.5142857 0.5319284 0.5375940 0.5390969 0.5511278 0.5578947    0
800  0.5157895 0.5338618 0.5398496 0.5411660 0.5522556 0.5593985    0
1000 0.5157895 0.5326814 0.5398496 0.5415411 0.5533835 0.5624060    0
2000 0.5157895 0.5356867 0.5368421 0.5417291 0.5556391 0.5609023    0
```

Figure 20 Accuracy When the Number of Trees Was Changed

The hyperparameters were decided for the model and then for each of the 8 folds, the Random Forest was created with a different testing dataset each time. For creating the models, randomForest R package was used. This implements Breiman [46] and Cutler's Random Forest algorithm using random inputs to solve classification and regression problems. The randomForest package is also to retrieve the importance of each of the input to the model to determine which features are affecting the model the most.

Similarly, to the Neural Network models the Random Forest was created in a R script called *myOwnRandomForest.R*. In this script, the grid search was conducted, the best model was chosen and the saved for use. The functions here were:

- *createForest*() – This function was used for creating the Random Forest models using the hyperparameters derived from the grid search.
- *createAndEvaluateForest*() – This function is used to load or save the models returned from the *createForest*() function; it also calculates the confusion matrix.
- *getTreeClassifications*() – This returned the predicted class probabilities from the random forest and testing dataset.
- *getRandomForestMetrics()* – This function returned the metrics of the Random Forest model including the Precision, Recall, Accuracy, F1 Score and more which is discussed in Chapter 5.

5 Evaluation of Models

5.1 Evaluation Metrics Used

For evaluating the Deep Neural Network and Random Forest models created, certain metrics were used that will be explained below.

1) Accuracy – how often the number of correctly predicted classes (win, draw and loss) in the dataset is predicted correctly. For example, if 6 games were correctly predicted out 10 games, the accuracy would be 6/10 = 0.6. Other metrics besides accuracy were used as the dataset did not have the same number of wins, draws and losses, therefore it can be seen as an imbalanced dataset. For this reason, the metrics below were also used.

2) True Positive (TP) – Number of occurrences where the label was correctly predicted.

3) False Positive (FP) – Number of occurrences where the label is classified as wrong

An example of a confusion matrix can be seen in Figure 20 and 21, showing how the TP and FP are calculated. Figure 21 calculates the TP for all the values for all the animals, this the value where the actual and predicted for each of the animals intersect.

		ACTUAL		
		Cat	**Dog**	**Fish**
PREDICTED	**Cat**	5	6	5
	Dog	3	10	1
	Fish	7	4	5

Figure 21 True Positive Example

Figure 22 shows how the false positive values are created. This can be calculated in two different ways, number of cases – TP or adding up the values where the animals do not intersect. It can be seen as the number of occurrences where the wrong class is predicted.

		ACTUAL		
		Cat	Dog	Fish
PREDICTED	Cat	5	6	5
	Dog	3	10	1
	Fish	7	4	5

Figure 22 False Positive Example

4) Precision – Calculates what proportion of the predicted positive class data was actually positive. Precision is important when we require the correct label to be produced. For this project, this is important as it results in the correct outcome of the football match being achieved. The formula is shown in Equation 8.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Equation 8 Precision Formula

5) Recall – Calculates what proportion of the positive data was actually predicted as positive by the classifier. Recall is important when getting the value correct for a certain class is important. For example, a security scanner searching for illegal objects, such as knives, on entrance to a building. It has to get every dangerous item identified. However, it is acceptable if someone is checked even if they do not have an illegal object.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

Equation 9 Recall Formula

6) F1 Score – Measures a balance between Precision and Recall as shown in equation 1. A F1 score of 1 means that the model is perfect, and this only occurs if the precision and recall values are both at 100%. If either precision or recall is 0, then the F1 score will be 0.

$$F1 = 2\frac{Precision * Recall}{Precision + Recall}$$

Equation 10 F1-Score

7) RMSE – The Root Means Square Error is a way of measuring the error of a machine learning model. It is calculated using the formula shown in Equation 11, where y_i is the actual value and x_i is the predicted value. The RMSE value can be seen as a method of evaluating the models for usefulness or accuracy. RMSE uses the Euclidean distance between the predicted results and the actual results for its method.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{n}}$$

Equation 11 RMSE Formula

8) MAE – The Mean Absolute Error is the mean of the difference between the predicted and actual value and is used to assess the quality of the machine learning model. Its formula is shown in equation 12 where y_i is the actual value and x_i is the predicted value. The main difference between the MAE and the RMSE is that the RMSE squares the errors before they are averaged and therefore, gives a high weighting to the larger errors.

$$MAE = \frac{\sum_{i=1}^{n} abs(y_i - x_i)}{n}$$

Equation 12 MAE Formula

9) R2 – This can be explained as the percentage of variation in the data. It can also be interpreted as a measure of how close the data points are fitted to the regression line. Its formula can be seen in Equation 13 where RSS is the sum of squares of residuals and TSS is the total sum of squares. A low R2 measurement means that the model does not fit the data very well.

$$R^2 = 1 - \frac{RSS}{TSS}$$

$$RSS = \sum_{i=1}^{n} ((y_i - x_i))^2$$

$$TSS = \sum_{i=1}^{n} (y_i - \overline{y_i})^2$$

Equation 13 R2 Equation

5.2 With Transfer Data Models

The accuracy results for both machine learning models are shown in figure 23. It can be seen that the Deep Neural Network marginally performed better in this case by 0.6%.

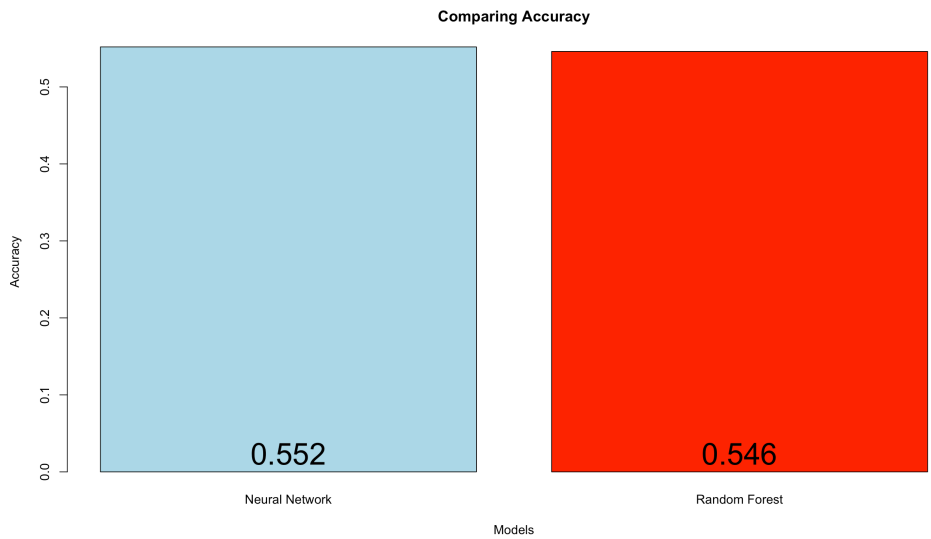


Figure 23 With Transfer Accuracy Comparison

On average, The Neural Network had a higher True Positive number than the Random forest on average over the 8 folds, at 524 compared to 518 meaning that the network was better at predicting the correct class (home win, draw, away win). In contrast, the Random Forest had a higher average False Positive value at 431 compared to the Neural Networks 425, which also shows that predicted the wrong label on more occurrences. The Deep Neural Network model also performed better than the Random Forest in precision and recall as show in Figures 24 and 25. This shows that the Neural Network was better at correctly classifying the classes and led to overall better results than the Random Forest.

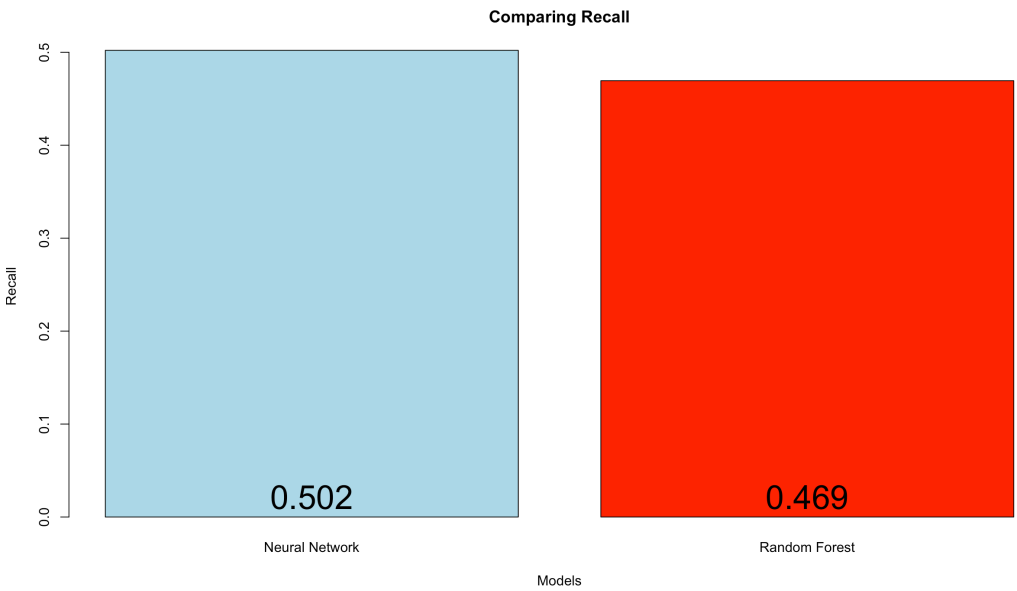


Figure 24 Recall for With Transfer Model

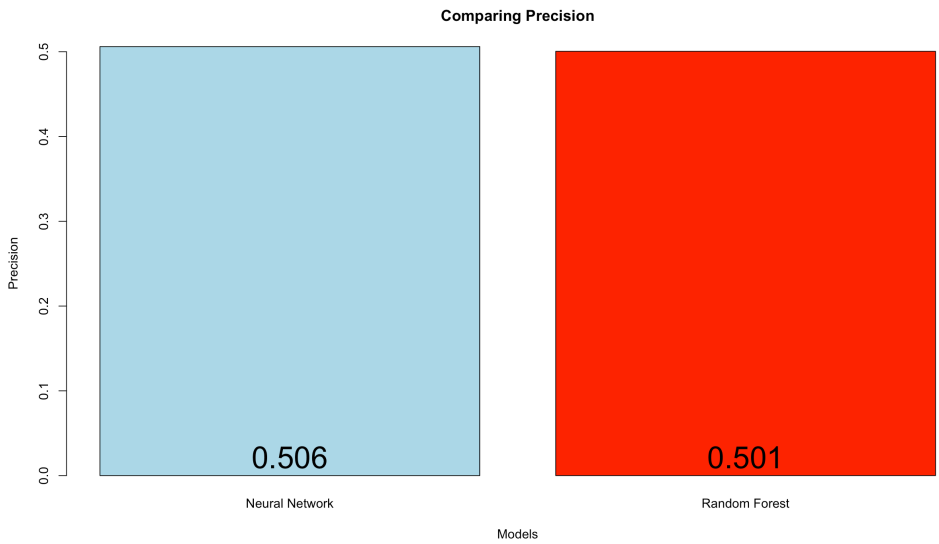


Figure 25 Precision for With Transfer Models

The models also showed us that the F1 score was higher for the Deep Neural Network at 0.566 compared to the 0.486 of the Random Forest, showing that the Neural Network was closer to being the perfect classifier. From the metrics above, it was concluded that the Deep Neural Network model created was better at correctly classifying the outcome of the Premier League Football match using the data that included the expenditure and income of the teams once the transfers were adjusted to “todays money”. The R2 value was also higher for the Neural Network at 0.184 compared to the 0.126 of the Random Forest, showing that the Deep Neural Network model fit the data better out of the 2 classifiers.

For the MAE and the RMSE of each of the models, the Neural Network had a lower value for both. The lower MAE value of the Neural Network showed that the Random Forest model’s predictions had more errors on average across the 8 folds. To find out if the model was underfitting the data, the RMSE of the training data and the testing data was calculated. If the RMSE of the testing data was greater than the RMSE of the training data then the model would be overfitting the data. If the RMSE of the testing data is less than the training data then the model would be underfitting the data. For the Deep Neural Network, the RMSE for train was 0.526 and for the test data was 0.588 showing that the data was slightly overfitting the data. The small difference between the two RMSE values shows that the model was not overfitting the data too much and can be concluded as successful. For the Random Forest Model, the RMSE value was 0.501 for the training data and 0.509 for the testing data. There was very small margin between the testing and training RMSE values showing that the Random Forest Model was also successful.

Overall, the Deep Neural Network performed better than the Random Forest model when the transfer data of the teams were considered. Therefore, it was decided that the Deep Neural Network will be used for the football prediction application for the with transfer model.

5.2 Without Transfer Data

When comparing the models that did not include the expenditure and income of each of the teams' transfers, the accuracy for both models was marginally better than the models that considered transfer data. As shown in Figure 27, the accuracies were very similar and other metrics would be key to deciding which model was best when transfer data was excluded.

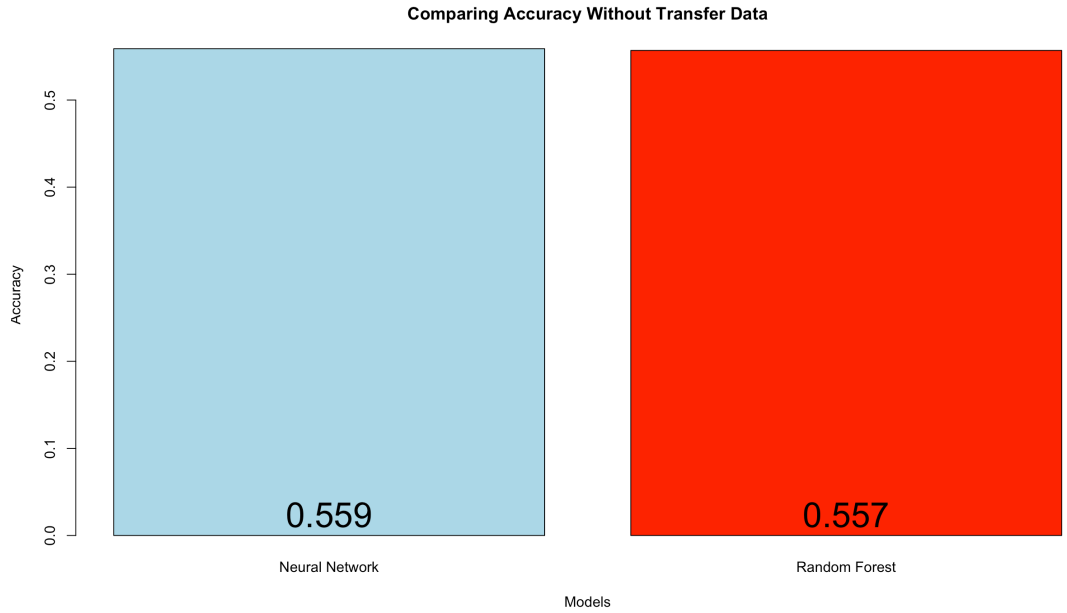


Figure 26 Accuracy for Without Transfer Models

The True Positive value was higher on average for the Neural Network over the 8 folds at 531 compared to the 529 for the Random Forest. This shows that the classifier was slightly better at predicting the correct class of result. Both classifiers were performed better in this compared to the models that included the transfer data. The False Positive value for the Neural Network was also lower than the Random Forest at 420 compared to 419 on average across the 8 folds. Once again, both models performed better compared to the models that included transfer data. As the True Positive and False Positive values were very close for both Random Forest and Neural Network, it would be unfair to use these metrics to conclude which model performed better. The Recall, Precision, F1-Score and error metrics will be used when deciding if the Deep Neural Network or Random Forest model would be used for the football predictions application.

Following the trend of the accuracy, True Positive and False Positive, the Precision of the Neural Network and the Random Forest were similar with the Neural Network having a marginally higher score as shown in Figure 26. The recall value for the Deep Neural Network was also higher but by a larger margin 5.01% increase as Figure 27 shows. This showed that both machine learning performed similarly when calculating the ratio between the True Positives and the all the positives, however, the Neural Network had a higher proportion of correctly predicting the True Positives in the testing dataset across the 8 folds. In comparison to the models that contained the transfer data, both models performed better for Precision and Recall scores for the without transfer data models. This followed the trend that the model tended to perform better when the transfer data was excluded. This could be due to the transfer values ranging from £0 to £100's of millions of pounds leading to a varied and potentially skewed dataset even after outliers were replaced.

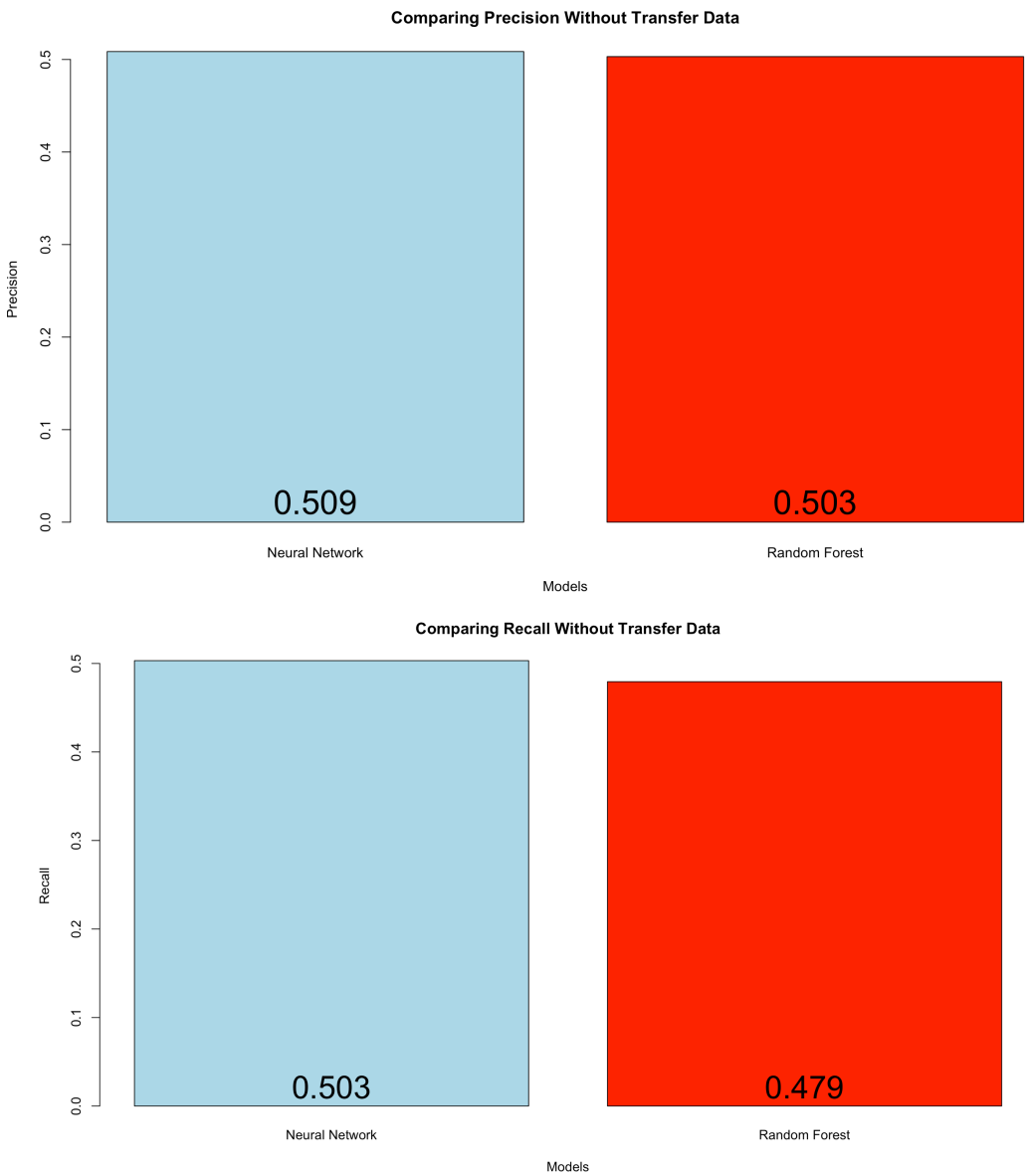


Figure 27 Precision and Recall of Without Transfer Models

The F1 Score had a slightly bigger gap between the Neural Network and the Random Forest with the Neural Network having a score of 0.579 and the Random Forest's score was 0.489. There was a 9% difference between the results showing that the Deep Neural Network model has a better balance between of Precision and Recall. The with transfers model had a lower F1 Score for both classifiers, again showing that the models overall performance improves when the transfer data is used. Teams' expenditure and income can have a random element to it; therefore, the full-time result and the Premier League teams net spends features may not be fully correlated.

Across the 8 folds of each model the RMSE value for the Neural Network was on average larger than the Random Forest model as shown in Figure 28. When compared to the RMSE values on the training set, the RMSE for the Neural Network was 5.2% more for the testing dataset than the training dataset, showing that the model had overfit the data. For the Random Forest model, the training dataset had a RMSE value of 0.496, which was 0.5% less than the RSME for the testing dataset, proving that the model only slightly underfit the data but fit the data better than the Deep Neural network Model. In contrast to this the MAE value for the

Forest was 0.32 compared to the 0.29 for the Neural Network, showing that the Random Forest had a larger difference of error when predicting the outcome.

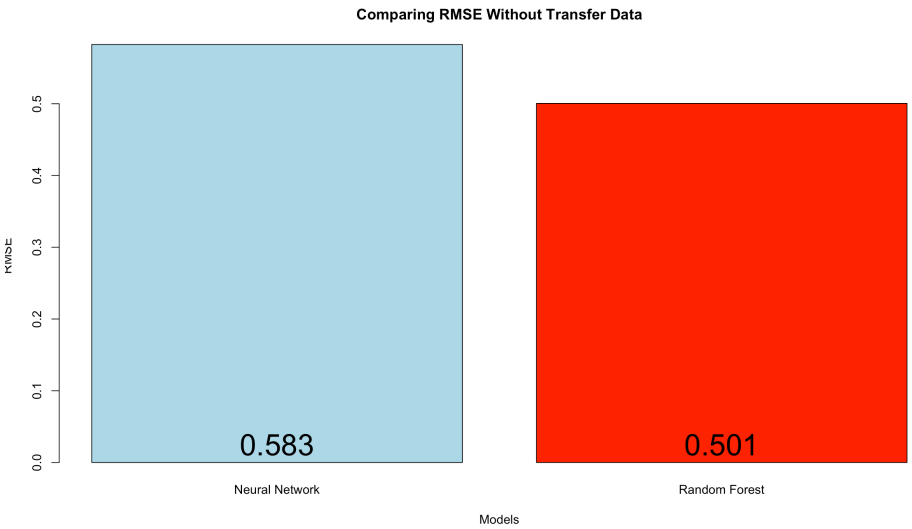


Figure 28 RMSE Score Comparison of Without Transfer Model

The R2 value of the Neural Network was 0.194 in comparison to the 0.147 returned by the Random Forest. Although both values returned were low, the Random Forest's value was closer to 0, meaning that the Neural Network model fit the data better. Compared to the models created that included the transfer data, the R2 scores were higher for both of the without transfer data models. This once again proved that the model's performance was better overall when the transfer data was excluded.

From the evidence above the Accuracy, Precision and Recall were all very similar for both models. The RMSE, showed that the Neural Network slightly overfit the data but this was only by small margin and the difference between the train RMSE and the test RMSE was not large. Due to the F1 Score showing that the Deep Neural Network had a better balance of Precision and Recall and the R2 score providing evidence that the model fit the data better, it was decided that the Deep Neural Network would be used in the application to predict the outcome football matches when transfer data was excluded.

6 Football Prediction Application

6.1 Technical Choices for Application

The programming language R was used for this project. This was chosen for its ease and because of my experience in using the language during my industrial placement. R and Python are the main languages that are used in creating machine learning models and R allows for fast prototyping. R is comprised of multiple large libraries which can help improve performance and is useful for data exploration and visualisation with only a few lines of code.

An Integrated Development Environment called RStudio [47] was used for the development of this project, from the data analysis to the machine learning models and for the user interface. It was initially developed by JJ Allaire and a team of R experts. RStudio can be ran on Mac OS, Linux or Windows and is free to use for all members. It is split into four main panes, which are shown in their default arrangement in Figure 29, however these can be rearranged as the user wishes. The first pane (top left) is the window where the user to view their files and data. It allows the user to view the files where their functions or code is created and can also display data in a spreadsheet-like visualisation. The second pane (top left) is where the objects that are loaded into the working environment are shown to the user. All datasets, functions, and objects that are available are displayed here. The user can click on the variable/object in the window to get more information on it. The history tab shows the history of the R code typed.

The third pane (bottom left) shows the console and allows R programmers to enter R code in the command line. Here the code that the user wants to run is completed here, and the output of the code is usually displayed here. The fourth pane (bottom right) is where the files are displayed, the user can rename a file/folder, create a folder, delete a file/folder, and multiple other tasks. This pane is especially important for project maintenance of the project files and folders. The plots are also displayed here as a result of the code in the Plots tab. The Packages tab shows the packages that are currently loaded in the project and allows the user to install and update packages as necessary.

RStudio is extremely user friendly and provides help for the users. The user can type a ”?” before a function name and a description of the function and its arguments will be displayed in the help tab. When the user is using a function that is from a package that is loaded in the project, a hint will show up for the user to display what arguments can be used. These features make RStudio an ideal IDE for this project as it allows for useful project management and maintenance whilst being ideal and easy to use.

The R version used for this project was 4.0.3 which was the nickname “Bunny-Wunnies Freak Out” and RStudio IDE version 1.3.1093.

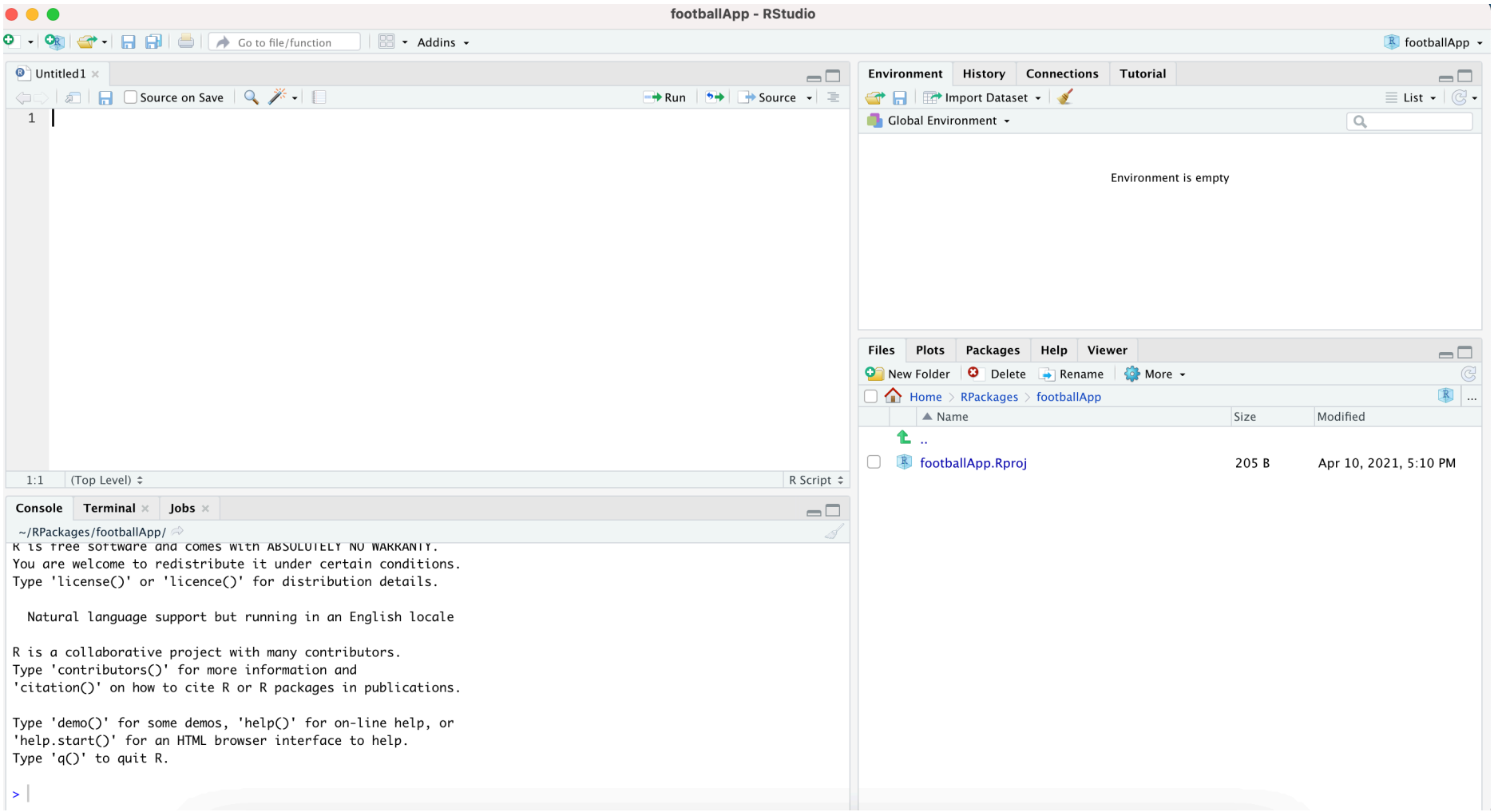


Figure 29 RStudio Integrated Development Environment Template

6.1 Shiny Application and R Libraries Used

To allow for access to viewing the predictions formed by the machine learning models, an application was developed. The final application that was created using the *R* package called *Shiny*. This allowed for a user interface to be formed and would allow the user to input data and the outcome (home win, away win, draw) of the head-to-head match would be displayed. *Shiny* was chosen as it provided it allows any *R* user to build web applications without the ease of needing another program and does not require the need for advanced web development knowledge. It can be used for many different varieties of tasks, for example, sharing insights through visualisations, prototyping and day-to-day tasks. *Shiny* allows a web application to be deployed and shared as a URL, so that it appears as a regular web page. Table 8 shows the *R* libraires used in the creating and evaluation of the models and the web application created for this project.

Library Name	Version	Package Description
shiny	1.5.0	For creating web application interface.
shinydashboard	0.7.1	Allows the creation of dashboards using shiny.
shinyjs	2.0.0	Allows improvement in user experience in shiny application
leaflet	2.0.4.1	Creates interactive maps for the user interface
rsconnect	0.8.17	Used to deploy the shiny application to the web
stats	4.0.3	Contains statistical functions.
ggplot2	3.3.2	Creates graphs for data and results visualisation.
outliers	0.14	Contains functions for detecting and removing outliers.
caret	6.0-86	Used for classification and regression training and used in this project for finding optimal Random Forest parameters.
dplyr	1.0.2	Used for data manipulation such as, merging, filtering, grouping and sorting datasets and mutating columns to the datasets.
pacman	0.5.1	Package manager, used for installing and loading R libraries.
corrplot	0.84	Used to find and display correlation between fields in the dataset.
randomForest	4.6-14	Used to create, train, test and evaluate Random Forest model
H2O	3.33.0.53	Advanced Neural Network package [43]

Table 8 Description of Packages Used in This Project

The application developed had multiple tabs.

1. The team page, this page will allow the user to select a team to view and display their Premier League record and a description sourced from Wikipedia. A map of the team's stadium is also displayed along with their stadium name and capacity.
2. The historical page, for viewing historical tables of the English Premier League over the years and additional records such as titles, top 4's, relegated, and overall records.
3. The transfers tab, this displays the transfer spending of the teams over the seasons for expenditure, income and transfer balance (income – expenditure).
4. The prediction tab, this has two tabs, a with transfer tab and without transfer tab. The with transfer tab allows the user to input the teams and transfer spending and income. The without transfer tab only allows the user to select the teams. The model metrics including accuracy, F1-Score and R^2 value.

The RShiny application was created using two R Scripts, *ui.R* and *server.R*. The first file is used for the user interface definition of the application, the layout of the application and each of its pages are done here. The server file is used for the server logic of the Shiny web application, the functionality is completed here which leads to the outputs being displayed in the user interface.

Table 9 and 10 show the functional and non-functional requirements for the web application. The functional requirements define the function of the web application and describe what is expected to be outputted by the project. Tests will be created for each of the functional requirements and will be given either a pass or fail depending on if the tests fulfil the requirements. Non-functional requirements are used to judge the performance of the application rather than the expected outputs of the system. These will also be judged with a pass or fail with a description for the reasoning of how the test result was derived.

6.2 Functional Requirements

Req ID	Requirement Description
	Prediction Page
R1.1.1	The system shall allow the user to choose if the predictions are done with transfer data or without transfer data.
R1.1.1.1	The systems shall allow the user to select the following if With Transfer Data is selected from R1.1.1. • The home team for the prediction • The away team for the prediction • The home team expenditure from a slider • The away team expenditure from a slider • The home team income from a slider • The away team income from a slider
R1.1.1.2	The systems shall allow the user to select the following if Without Transfer Data is selected from R1.1.1. • The home team for the prediction • The away team for the prediction
R1.1.1.3	The system shall allow the user to press the predict button to run the prediction of the football match.
R1.1.1.4	The system shall display the following, depending on R1.1.1: • Model Accuracy • Model Precision • Model Recall • Model F1-Score • More R^2 • Model RMSE
R1.1.2	The systems shall display a loading message following the user click in R1.1.1.3.
	Historical Data Page

R1.2.1	The system shall allow the user to select a Season and display a table containing the following: • Team – Teams playing in the Premier League that specific season • Season – The season the table is related to • Points – Number of points the team obtained that season • Wins – Number of wins the team obtained that season • Draws – Number of draws the team obtained that season • Losses – Number of losses the team obtained that season
R1.2.2	The systems shall allow the user to select a range of Seasons from 2001 to 2020 and shall allow the user to select an option from the following: • Years in Premier League • Titles • Top 4 • Relegated • Overall
R.1.2.2.1	The system shall display the following if "Years in Premier League" was chosen in R1.2.2: • Team – The team the row data relates to. • Tournaments – The number of seasons the team has played. • Matches – The number of matches the team has played. • First – The first season the team played in Premier League. • Latest – The last season the team played in Premier League. • Highest – The highest place finish for the team. • Lowest – The lowest place finish for the team.
R1.2.2.2	The system shall display the following if "Titles" was chosen in R1.2.2: • Team – The team the row data relates to • Titles – The number of first place finishes the team has obtained • Seasons – The different seasons where the team has finished first
R1.2.2.3	The system shall display the following if "Top 4" was chosen in R1.2.2: • Team – The team the row data relates to • Top 4 – The number of times the team has finished the season in 1st, 2nd, 3rd or 4th place • Seasons – The different seasons where the team has finished in the top 4
R1.2.2.4	The system shall display the following if "Relegated" was chosen in R1.2.2: • Team – The team the row data relates to • Relegated – The number of times the team has finished the season in 18th, 19th or 20th place • Seasons – The different seasons where the team has been relegated.
R1.2.2.5	The system shall display the following if "Overall" was chosen in R1.2.2:

	• Team – The team the row data relates to • Points – The total number of points obtained • Wins – The total number of wins over the seasons • Draws – The total number of draws over the seasons • Loses – The total number of losses over the seasons.
	Team Page
R1.3.1	The system shall allow the user to select a team from a selection of 43 different teams.
R1.3.2	The system shall display a scatter chart displaying the position of the team selected over the seasons that the selected team has played in the Premier League.
R1.3.2.1	The system shall display a line where position is 4, to show the times when the selected team has finished in the positions higher or equal to 4th.
R1.3.2.2	The system shall display a line where position is 18, to show the times when the selected team has finished in the positions lower or equal to 18th.
R1.3.3	The system shall show the Premier League record for the team selected, including the following: • Tournaments – Number of seasons the team has played in the Premier League • Titles – Number of seasons the team has finished in 1st place • Top 4's – Number of times the team has finished 1st, 2nd, 3rd or 4th. • Times Relegated – Number of times the team has finished 18th, 19th or 20th. • Wins – Total wins over all the seasons for the selected team • Draws – Total draws over all the seasons for the selected team • Losses – Total losses over all the seasons for the selected team • Goals – Total goals scored over all the seasons by the selected team
R1.3.4	The system shall display a brief overview of the team selected along with a link to the search page on Wikipedia.
R1.3.5	The system shall display a map showing the location of the selected teams stadium.
R1.5.3.1	The system shall show the user the stadium name and capacity if the user hovers over the marker displayed in the map from 1.3.5.
R1.3.6	The systems shall display a table showing the selected teams overall Premier League record including the following: • Pos – The final position • Season – The season the row data relates to • Points – The final points • W – Wins • D – Draws • L – Losses • GS – Goals Scored • GC – Goals Conceded • GD – Goal Difference (GS-GC)
	Transfers Page
R1.4.1	The system shall allow the user to select the following to filter the transfer data:

	• The team • The range of seasons • If the transfer data is adjusted for inflation to convert the data to “todays money”
R.1.4.1.1	The system shall display the following using R1.4.1: • The average expenditure per season • The average player arrivals per season • The average income per season • The average player departures per season
R.1.4.1.2	The system shall display Transfer Spending Stats following R.1.4.1, which shall contain the following: • Team – The team the transfer statistics is related to • Expenditure – The total expenditure • Income – The total income • Balance – The total balance (Income – Expenditure) • Seasons – The number of seasons the transfer data is relating to • Arrivals – The number of arrivals • Departures – The number of departures • Data – If the transfer data is adjusted to “todays money” or not.
1.4.2	The system shall allow the user to select one of the following graphs to display: • Balance • Income • Expenditure
1.4.2.1	The system shall display a graph showing the balance (income – expenditure) of the team with positive values shown in green and negative values shown in red if Balance is selected from 1.4.2.
1.4.2.2	The system shall display a graph showing the Income of the team with positive values shown in green and negative values shown in red if Income is selected from 1.4.2.
1.4.2.3	The system shall display a graph showing the expenditure of the team with positive values shown in green and negative values shown in red if Expenditure is selected from 1.4.2.

Table 9 Functional Requirements

6.3 Non-Functional Requirements

Area	Req ID	Requirement
Performance	R2.1	Web Application should load within 15 seconds.
Usability	R2.2	Web application should be accessible through a web browser
Usability	R2.3	Web application should be designed in a user-friendly way.
Availability	R2.4	The system shall be available for use 24 hours a day, 7 days a week.
Performance	R2.5	Predictions should be complete within 5 seconds.
Security	R2.6	The user should not be able to change the code or download the data from the application.

Table 10 Non-Functional Requirements

6.4 Deployment of Application

The Shiny application was deployed and hosted using Shinyapps.io [48], which is platform as a service. To do this, a development environment was required, which was RStudio and the *rsconnect* package, which is used to deploy applications using shinyapps.io. The next stage was to create a shinyapps.io account where a dashboard is displayed, and the owner can view the application details here. Figure 31 shows the application page on the dashboard shows the overview of the shiny application including the URL, the status (Running, Sleeping or Archived), The date deployed and updated, the current running instances and the application usage. On the dashboard the metrics are also available to see including the connections, memory usage, worker processes, CPU usage and network usage. Other instances that can be viewed on the creator's dashboard are the logs and the ability to archive or delete the application. A token generated from shinyapps.io dashboard was using the code shown in Figure 30 this connects the RStudio development environment to the users details so that they can create and update the application and deploy it to shinyapps.io.

```
shinyapps::setAccountInfo(name='mytest',
                          token='                                ',
                          secret='<SECRET>')
```

Figure 30 Example of Token Necessary to Link Application to shinyapps.io

It was decided to use this platform for the user interface as it would allow the user easy access whenever they wanted to use the predictions. It may be seen as cumbersome to make the user open the package in RStudio and then open the application from the specific file. This would mean that the user only needs access to an internet browser and not need to have the RStudio environment and then install the necessary libraries on their system.

The relevant files that contained the functions necessary for the Shiny application were included in the bundle when deploying the apps. This included the functions that loaded the relevant libraries, created the tables and visualisations for the Shiny app and any functions that are necessary for working of the application. The three datasets, historical, transfer and transfer inflation, were also included along with the machine learning models for with transfer and without transfer data.

One of the struggles when deploying the Shiny application was the h2o package version. The version used for the models was not supported by shinyapps.io and kept failing when attempting to deploy the application. To fix this, the h2o library version was downgraded to the latest version supported by CRAN and changed the model versions to match the ones used in the Shiny application.

To access the application the user can type in:
https://varun-kale.shinyapps.io/FootballPredictionApp/
into their browser search bar. From this the user can view the different pages and run the predictions as often as the user wants to.

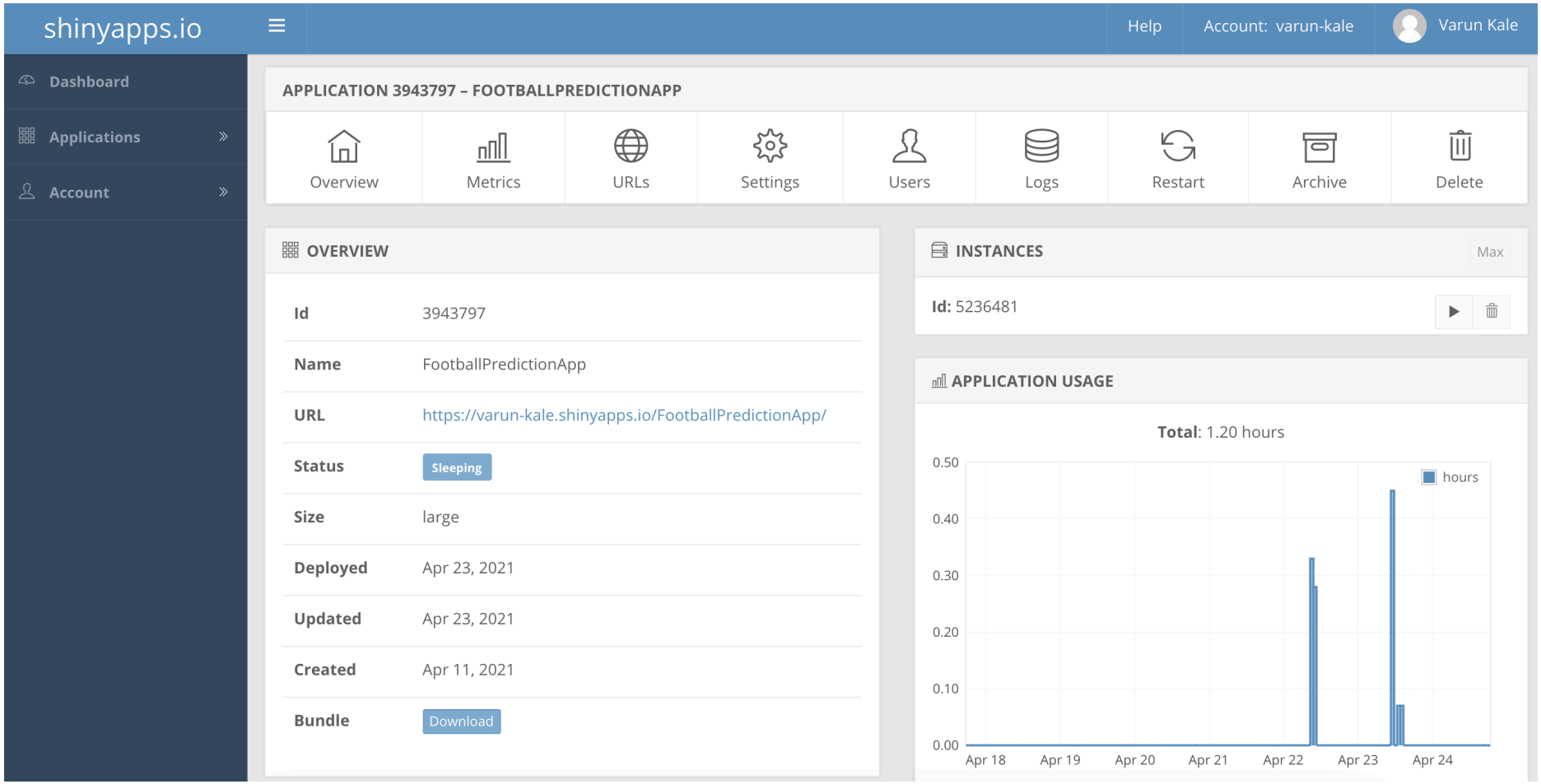
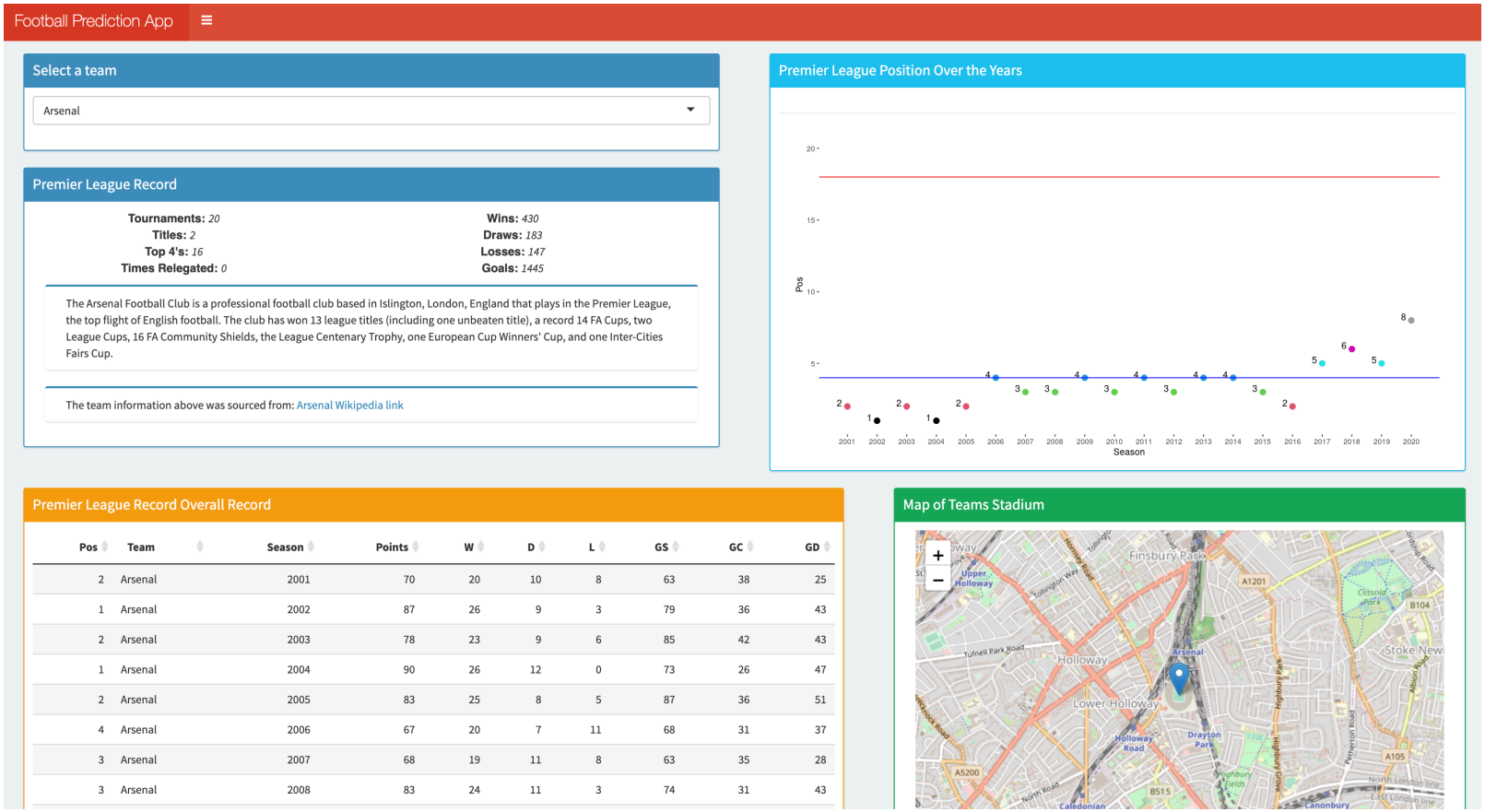


Figure 31 Shinyapps.io Admin Home Page for the Football Prediction

6.5 Football Prediction Application Screenshots

6.4.1 Team Page



Pos	Team	Season	Points	W	D	L	GS	GC	GD
2	Arsenal	2001	70	20	10	8	63	38	25
1	Arsenal	2002	87	26	9	3	79	36	43
2	Arsenal	2003	78	23	9	6	85	42	43
1	Arsenal	2004	90	26	12	0	73	26	47
2	Arsenal	2005	83	25	8	5	87	36	51
4	Arsenal	2006	67	20	7	11	68	31	37
3	Arsenal	2007	68	19	11	8	63	35	28
3	Arsenal	2008	83	24	11	3	74	31	43

Figure 32 Team Page Screenshot

Figure 32 shows the team page and the different sections, the user can select a team and it will display the team's historical data, a plot of their historical positions (top right) and a map showing where their stadium is located (bottom right).

6.5.2 Historical Results

Football Prediction App

Yearly Tournament Leaderboard

2001

	Team	Season	Points	W	D	L
1	Man United	2001	80	24	8	6
2	Arsenal	2001	70	20	10	8
3	Liverpool	2001	69	20	9	9
4	Leeds	2001	68	20	8	10
5	Ipswich	2001	66	20	6	12
6	Chelsea	2001	61	17	10	11
7	Sunderland	2001	57	15	12	11
8	Aston Villa	2001	54	13	15	10
9	Charlton	2001	52	14	10	14
10	Southampton	2001	52	14	10	14
11	Newcastle	2001	51	14	9	15
12	Tottenham	2001	49	13	10	15
13	Leicester	2001	48	14	6	18

Historical Records

2001 to 2020 Years in Premier League

	Team	Tournaments	Matches	First	Latest	Highest	Lowest
1	Arsenal	20	760	2001	2020	1	8
2	Chelsea	20	760	2001	2020	1	10
3	Everton	20	760	2001	2020	4	17
4	Liverpool	20	760	2001	2020	1	9
5	Man United	20	760	2001	2020	1	7
6	Tottenham	20	760	2001	2020	2	14
7	Man City	19	722	2001	2020	1	19
8	Newcastle	18	684	2001	2020	3	18
9	Aston Villa	17	646	2001	2020	5	20
10	West Ham	17	646	2001	2020	7	20
11	Fulham	14	532	2002	2019	7	19
12	Sunderland	14	532	2001	2017	7	20
13	Southampton	13	494	2001	2020	6	20

Figure 33 Historical Results Tab Screenshot

Figure 33 shows the historical results tab where the user can show the overall Premier League table for a specific season. On the table on the right the user can select a range of years and an option of either the Years in Premier League, Titles, Top 4, Relegated or Overall table to be displayed.

6.5.3 Transfers Page

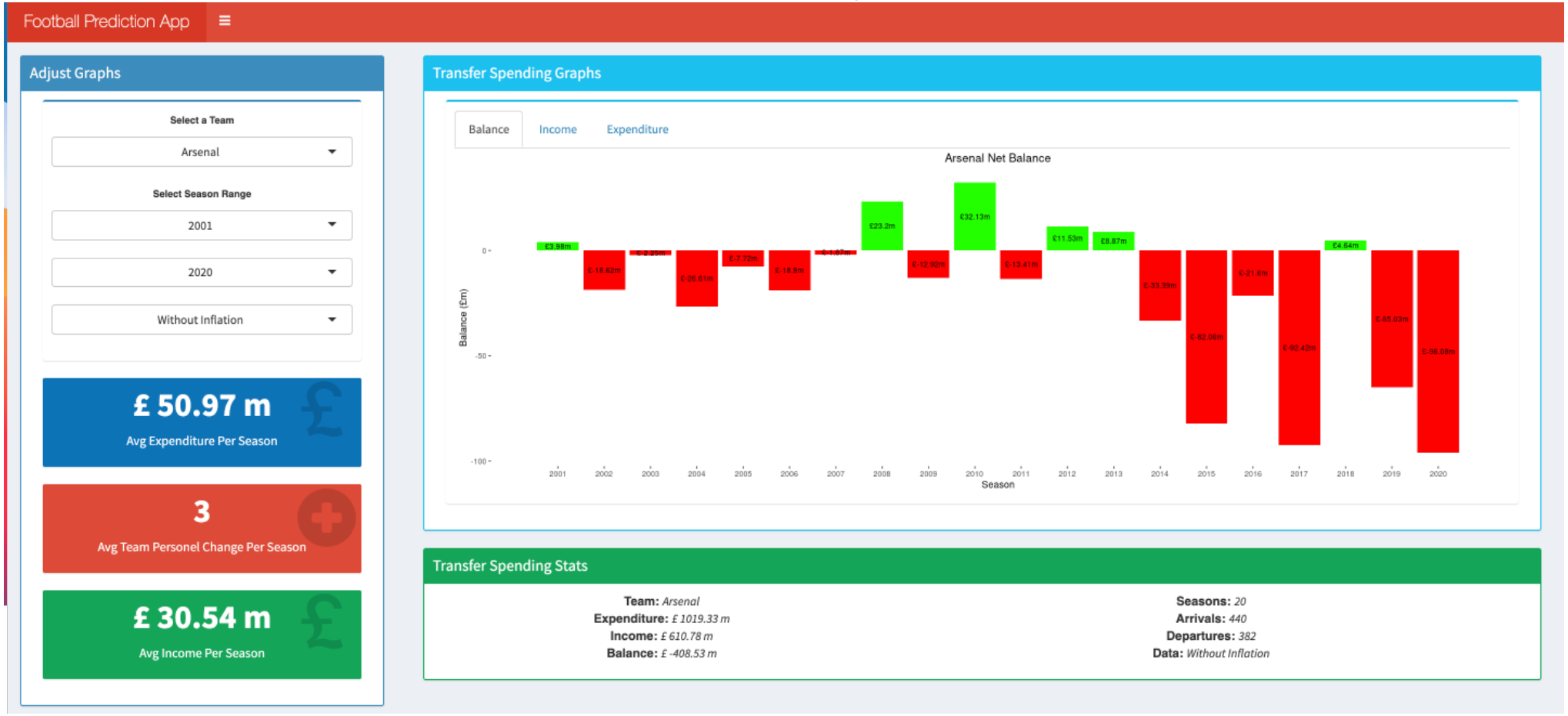


Figure 34 Transfers Page Screenshot

Figure 34 shows the transfer page from the application, the user can select a team, a range of seasons and decide if they want the data to show for inflation or without inflation. The average expenditure, arrival, income and departures are shown on the left and the transfer spending stats are shown in the bottom of the page. The user can choose which of the three graphs to show from Balance, Income and Expenditure with values less than £0 shown in red and values greater than £0 shown in green.

6.5.4 Prediction Page – With Transfer Data

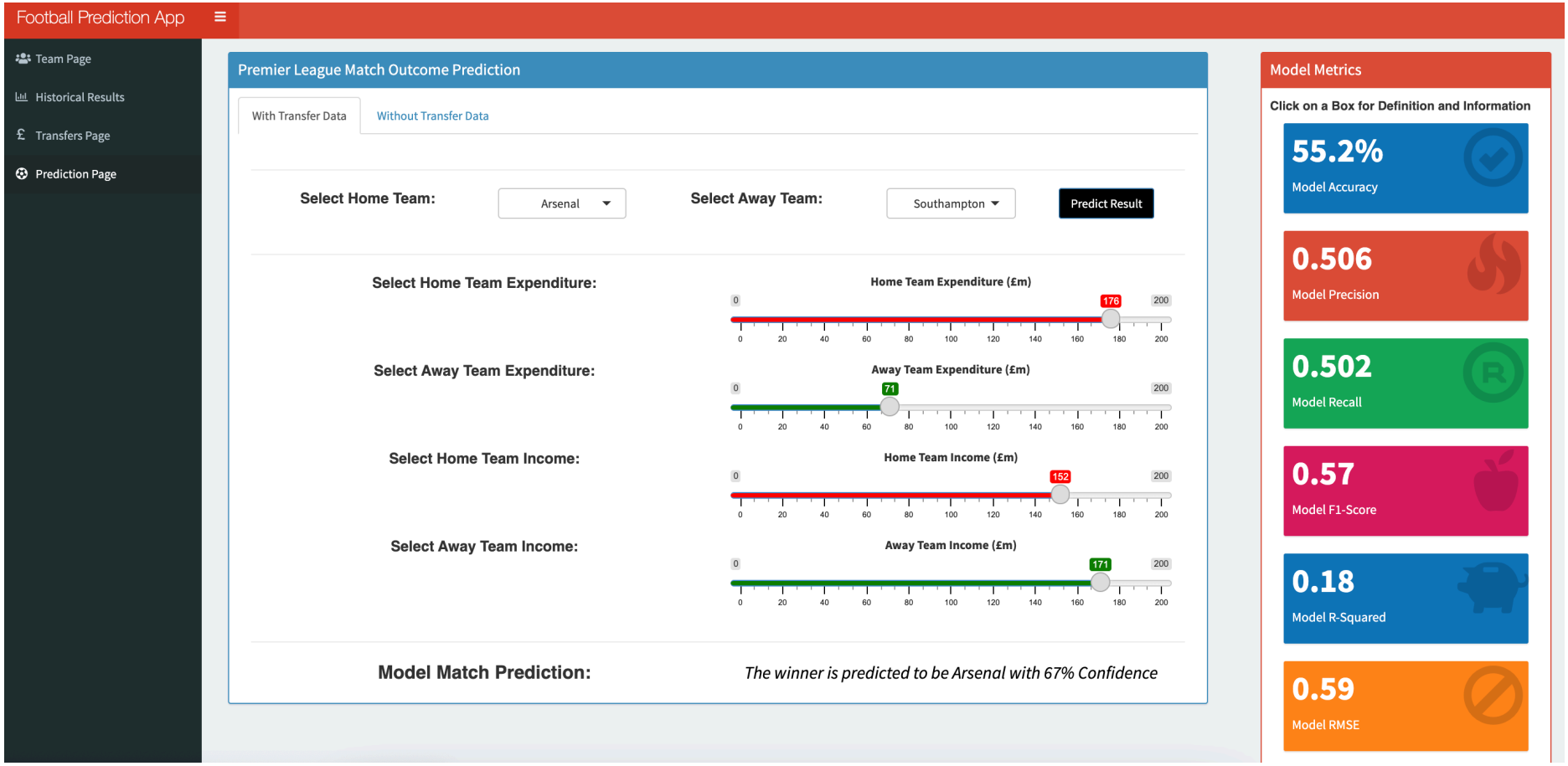


Figure 35 Prediction Page for With Transfer Model

Figure 35 shows the page where the user is able to select the teams and use the slider to set the amount spent and earnt by each of the teams from transfers and this will result in a prediction. Model metrics for a with transfer model can be seen on the right. The user can click on any of the metrics to see what they mean.

6.5.5 Prediction Page – Without Transfer Data

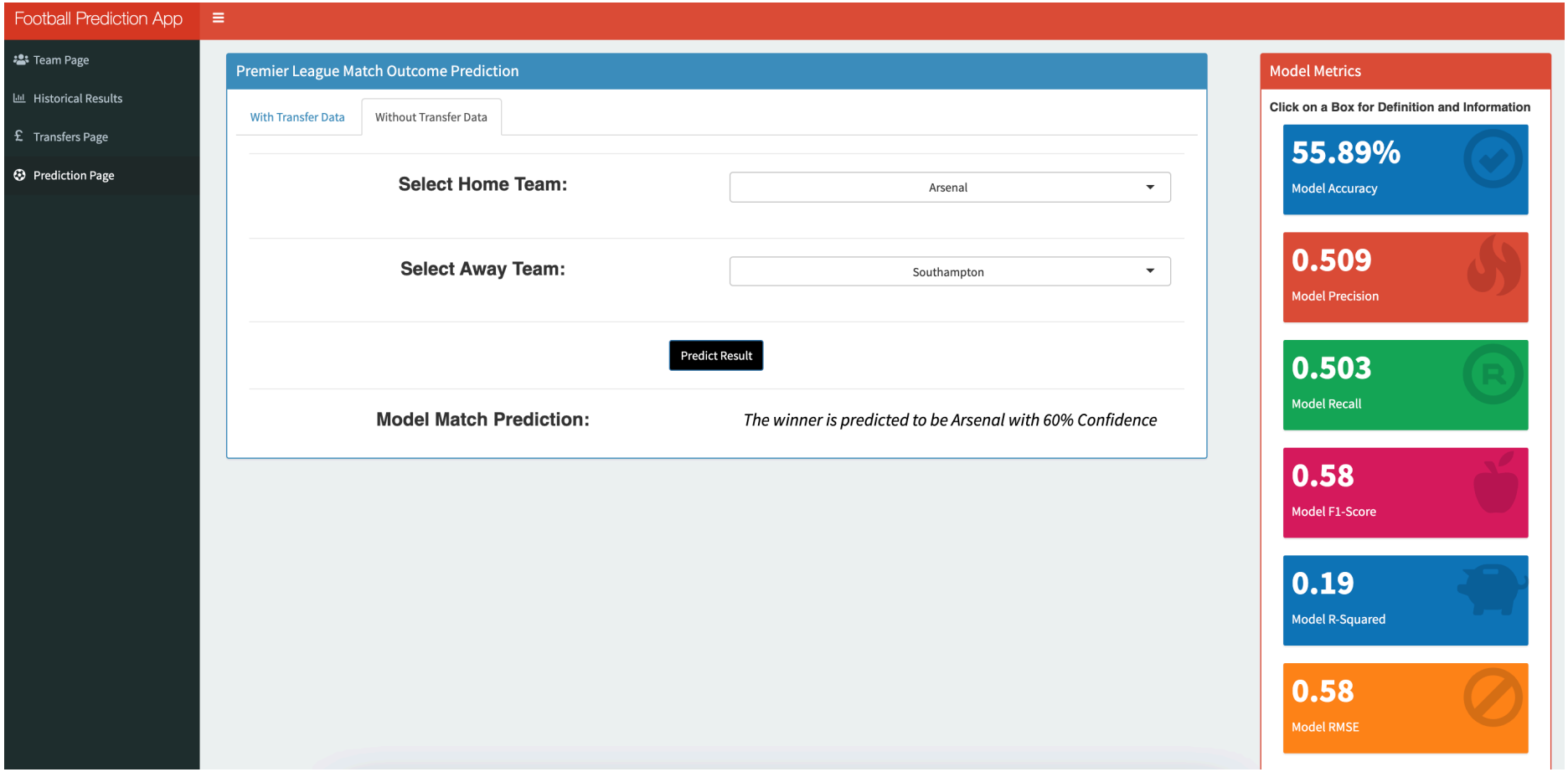


Figure 36 Prediction Page Screenshot for Without Transfer Model

Figure 36 shows the prediction page for the without transfer model. The user only selects the home team and away team here and presses the predict button to get a prediction. The model metrics are shown on the right.

6.5.6 Prediction Page – Loading Page

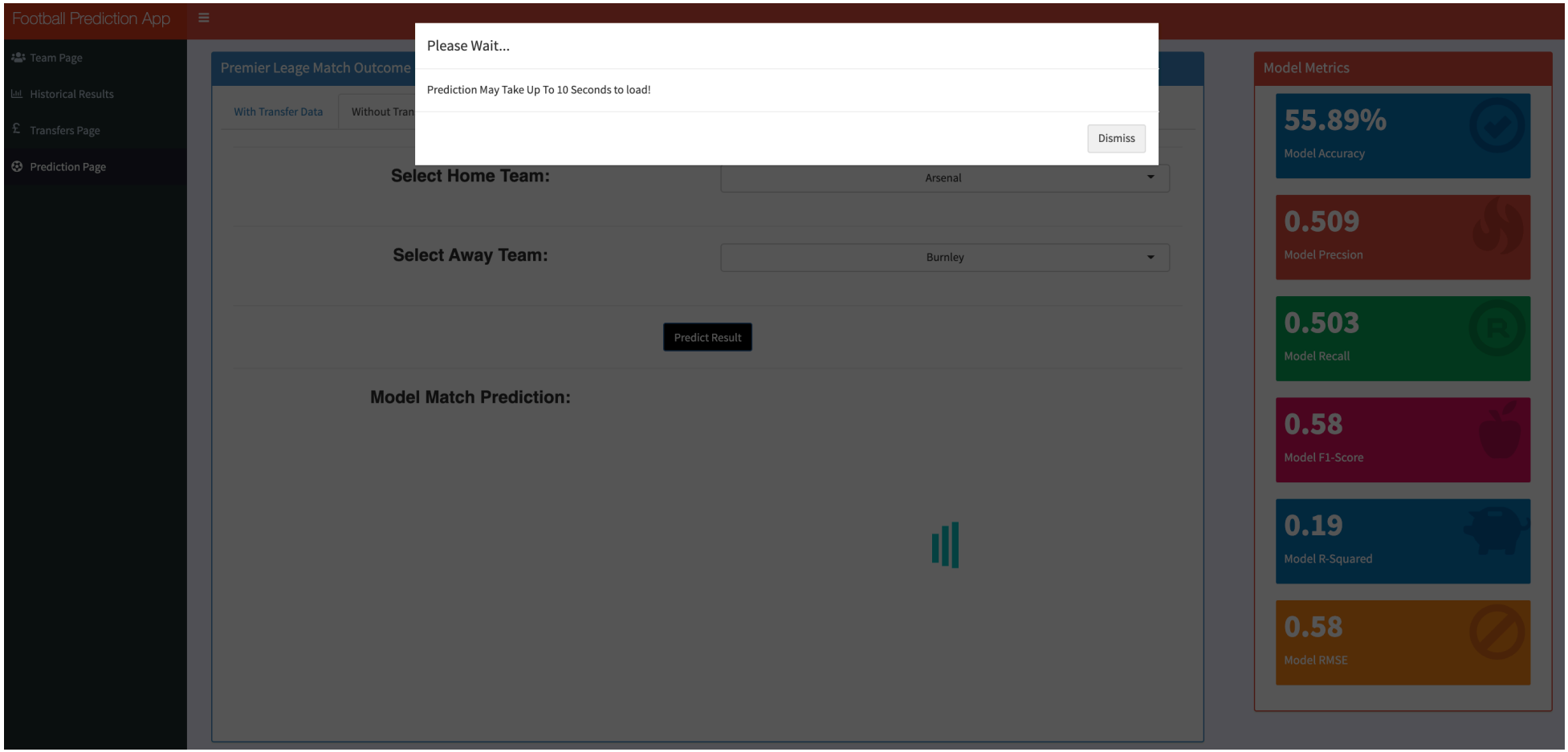


Figure 37 Prediction Page Screenshot - Loading Screen

Figure 37 shows the loading page when the user presses the predict button. The message will disappear when the prediction loads or if the user presses the dismiss button. A spinner is also displayed where the prediction is to be displayed.

7 Testing

Testing is an important part of any project as it can be used to evaluate the success of the overall project or application. The machine learning algorithms have previously been tested and given evaluation metrics as discussed in chapters 3, 4 and 5, therefore the data analysis and the testing of the models themselves are not included in this section. The user interface and application are the main outputs of this project and ensuring that there is full functionality is the main outcome to be tested for this project

7.1 Functional Requirements Testing

The testing for this project was done by testing against the functional and non-functional requirements that were in the application section. Each requirement will be briefly mentioned and given a simple pass or fail with each requirement.

- R1.1.1 User can choose with transfer or without transfer prediction – PASS
- R1.1.1.1 User can select inputs for with transfer model – PASS
- R1.1.1.2 User can select inputs for without transfer model – PASS
- R1.1.1.3 User can press predict – PASS
- R1.1.1.4 Metrics shall be displayed – PASS
- R1.1.2 Loading message shall be displayed – PASS
- R1.2.1 User can select year for historical table – PASS
- R1.2.2 User can select range of historical tables to display – PASS
- R1.2.2.1 Years in Premier League table displays - PASS
- R1.2.2.2 Titles table displays – PASS
- R1.2.2.3 Top 4 table displays – PASS
- R1.2.2.4 Relegated table displays – PASS
- R1.2.2.5 Overall table displays – PASS
- R1.3.1 User can select a team – PASS
- R1.3.2 Scatter chart for positions is displayed – PASS
- R1.3.2.1 4th position line is displayed – PASS
- R1.3.2.2 18th position line is displayed – PASS
- R1.3.3 – Premier League record is shown – PASS
- R1.3.4 – Team description is displayed – PASS
- R1.3.5 – Team map is displayed – PASS
- R1.3.5.1 – Stadium name and capacity is shown – PASS
- R1.3.6 – Overall Premier League record is shown – PASS
- R1.4.1 – User can filter transfer data – PASS
- R1.4.1.1 – Average Transfer information is displayed – PASS
- R1.4.1.2 – Transfer spending stats is displayed – PASS
- R1.4.2 – User can select transfer graph to show – PASS

- R1.4.2.1 – Balance graph displays – PASS
- R1.4.2.2 – Income graph displays – PASS
- R1.4.2.3 – Expenditure graph shows – PASS

7.2 Non-Functional Requirements Testing

Each of the non-functional requirements for the application were tested using the final user interface. The following shows how they were tested and if they had passed or failed the tests.

- R2.1 – Web application should load within 15 seconds – PASS.

This was tested by loading the web application and timing how long it took to open. Out of the ten attempts, nine of them resulted in them being faster than the 15 seconds, the one that failed was due to slow internet connection

- R2.2 – Web application should be accessible through a web browser – PASS

This was done through deploying the Shiny application allowing the user to access it through a web URL

- R2.3 – Web application should be designed in a user-friendly way

This was done by improving the initial application with feedback from the project supervisor to discuss ways that user experience can be enhanced.

- R2.4 – System shall be available for use 24 hours a day, 7 days a week

The deployed application is available for use whenever the user requires to use it.

- R2.5 – Predictions should be complete within 5 seconds

Initially the models were taking longer than ten seconds to load and after changing the way the models were loaded, they took under 5 seconds to run the predictions. From test of 10 predictions, they all ran in less than 5 seconds.

- R2.6 – The user should not be able to change the code or download the data from the application.

The web application does not allow access to the code that is used to run the Shiny application and only the admin can make updates. The data is not accessible to the user to ensure the integrity of the predictions and data in the application.

8 Project Evaluation

8.1 Comparison of Other Models

When comparing the results from this project to the results of researchers, the most evaluation metric used by the authors was accuracy and in some cases, precision and recall. Therefore, for the comparison of results, accuracy and F1-Score were used.

Figure 38 shows the accuracy of a few of the researchers compared to the models created in the project to predict the outcome of English Premier League matches in which one model includes the expenditure and income of the teams and one model excludes this data.

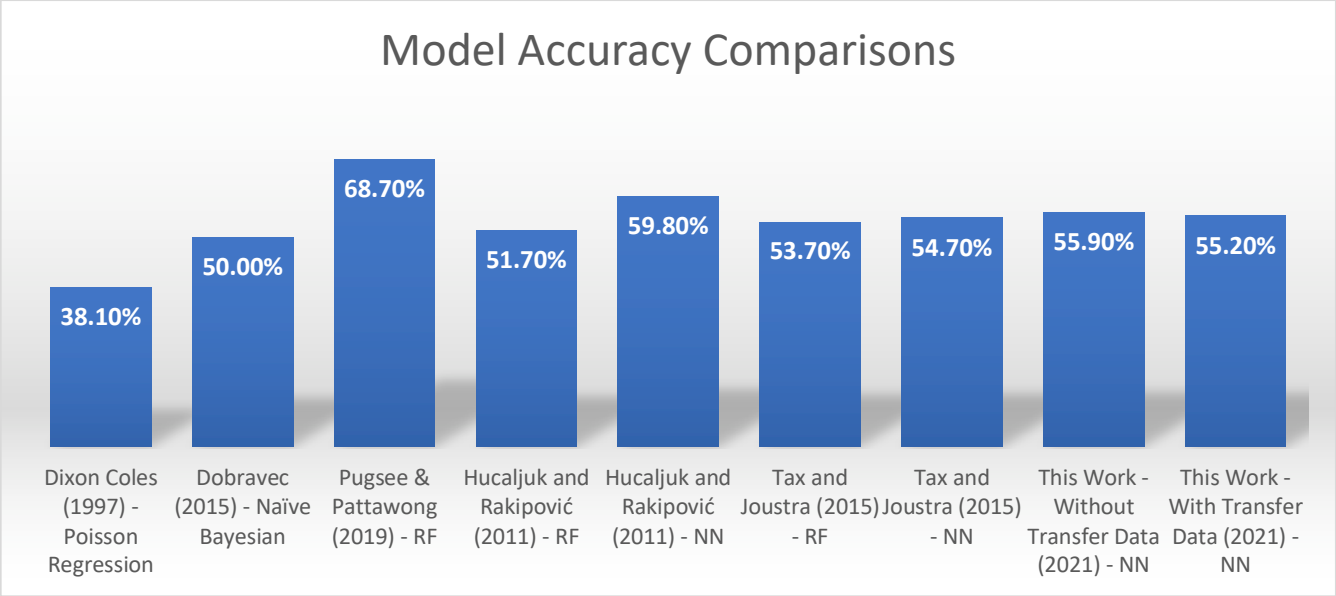


Figure 38 Comparison of Model Accuracies Across Papers

The accuracies tended to be in between 50-60% with the exception of Pugsee and Pattawong [36], who seemed to get high accuracies. This could be due to the data used in the model, the model took into account the referee. As the data used for the model was using three seasons as opposed to the twenty seasons of data that this project used. This would mean that the referees in the first season that Pugsee and Pattawong used be less likely to be changed, making the feature useful in the predictions as some team's results may be better when a specific referee is in charge of the match. The authors also used the previous five matches of each teams to create a "form" feature that would make the most relevant match important in addition to having home and away team possession data to help in the predictions.

Hucaljuk and Rakipović [34] achieved higher results in their predictions for the Neural Network classifier and lower accuracy in their Random Forest Model. Their model used the form of the team, their position in the table at the time of the match taking place and the number of injured players in the home and away team in addition to the historical results. The overall accuracy of Hucaljuk and Rakipović's model are similar to the ones achieved by this project.

As Figure 28 shows, the Poisson Regression model created by Dixon and Coles had a much lower accuracy than any of the models created using machine learning. The model for this model had a 55.2% accuracy including transfer data and 55.2% accuracy excluding transfer data compared to Dixon and Coles 38.1%. This shows that the machine learning models have an advantage in football prediction compared to using statistical methods only. Dixon

and Coles also predicted the probability of scores of the matches rather than the outcome of the games (home win, draw, away win), which could be a reason for the lower accuracy.

The overall low accuracy (less than 70%) shows the complexity that is involved when predicting football results. A team could beat the four best teams in the league and then the next week lose to the team placed bottom of the table. This could be due to many reasons including complacency, injuries, resting players or lack of motivation. From Figure 38, the average accuracy for the models was 54.2%, showing that both models created in this project were slightly above average and can be viewed as successful.

Figure 39 shows the F1-Score of the models created by researchers where available compared to the models created in this project.

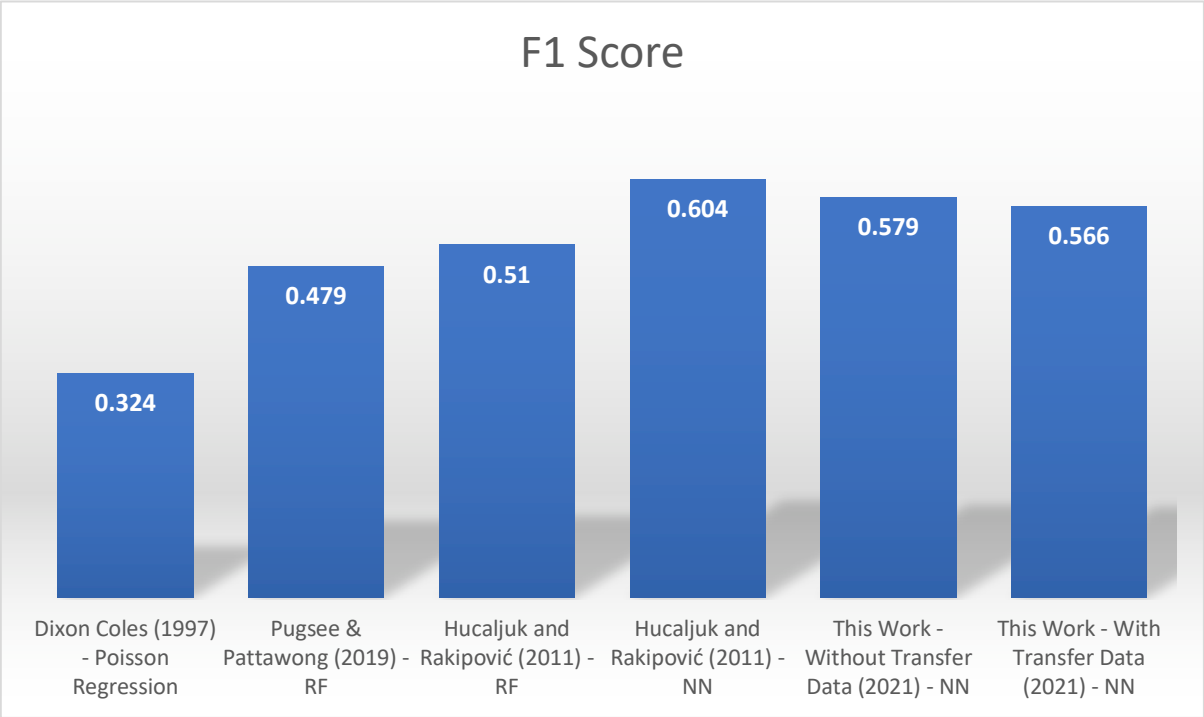


Figure 39 Comparison of F1-Scores Across Papers

It can be seen that only Hucaljuk and Rakipović's Neural Network only had a better F1-Score than the models created in this paper. Pugsee and Pattawong had a much lower F1-Score, even though the accuracy of the model was lower than the ones created in this paper. This shows that the models without transfer data and without transfer data models had a better balance of precision and recall. However, for the prediction of football results, the accuracy is vital as the main outcome, the Pugsee and Pattawong Random Forest is better for the predictions. If the overall model balance is the objective, it is clear that the models created for this paper are better balanced in terms of the precision and recall scores.

8.2 Critical Discussion

This report and project contain efforts to predict the outcome of Premier League results using multi-classification machine learning models, namely, Neural Networks and Random Forest. Four models were created, two that included the expenditure and income of the teams. It has been shown in section 8.1 that the models compared well with other papers, performing better than the majority of studies used as a benchmark.

8.2.1 The Project Objectives

This paper shows that the objectives of the project have been explored and successfully met as shown below:

1. Gather data for historical football results and teams transfer methods:

   The first stage in this process was deciding what data was necessary for this model. The historical data was the main dataset as it provided the basics for the models, such as the previous results and encounters between teams. As this project was created to evaluate the impact of transfers on the model outcomes and predictions, a transfer data set was necessary for this project. To take into account for the fluctuating prices of transfers and value of players with the same quality increasing over the years, it was decided that a transfer inflation dataset would also be used.

   Data collection was done by collecting the data from the relevant sources, transfer inflation data was also collected to transform the data to today's monetary value. The data was collected in different ways, the historical results was the simplest as it was a publicly available data set sourced from Kaggle. The transfer data collection was a lengthier process, each seasons transfer data was collected from transfermarkt.com using the Table Capture Google Chrome add on and each of the seasons were merged together to from a single dataset. In the future, finding a different, faster way to source the data could be looked at.

2. Research the methods used and models that have been created in predicting the outcome of sports games

   The aim for this objective was discover what previous literature was available that had been completed for similar projects. This would be used to show what was missing from previous research and what machine learning models were the best to use for the football prediction.

   This was done by researching other authors papers and exploring the different methods they have used. This was split into multiple sections, football rules and transfer market background, machine learning techniques background, machine learning in sports prediction, football prediction using statistical approaches, football prediction using machine learning and football prediction using transfer data. All these were evaluated, and the strengths and weaknesses compared. These were used as a benchmark against this project. The focus of this section was to explain the background of the project and find papers that were similar to this project and evaluate the research and this section successfully explain the required information.

3. Research on how expenditure of football teams can affect their performance

   The aim of this was to determine the factors that affect the transfers of football teams and how they have an impact on the results of the matches played. This section was also

written to give a brief overview of how the transfer window works and how and why teams choose to purchase other teams players.

This section was completed in the literature review section and discussed the reasons that teams spend more on transfers, factors that determine the price of players and guidelines that teams must follow when buying or selling their players, such as Financial Fair Play.

4. Explore, clean and manipulate the data to contain only the relevant features

   The purpose of this section was to explore the datasets and determine what features were necessary and what would add bias to the models. The data exploration involved determining what factors had the main effect on the predictions. The transfer data was also explored to see if the team's final position over the years affected the net spend of the team the next season. This stage also required removing data from the 1993/94 season to the 1999/00 season as they contained a lot of missing data which would cause complication when creating the models. Another task was adjusting the names from the transfer data set to the historical dataset, as they were sourced from different places they had slightly different variation of the team's name. For example, in the transfer dataset the team's name was Arsenal FC, whereas in the historical dataset the team was referred to as Arsenal. A key was created to match the team names. This stage also contained feature generation such as creating the position feature which was the teams overall placed finish at the end of that season.

   This stage was successfully completed; features were removed such as referee and the number of arrivals and departures from the transfer data. The removal of these features was discussed and justified in the data exploration and pre-processing. Full-time and half-time goals were also removed as they would add a significance bias to the results. One of the main challenges in the data manipulation and was merging the data together, this was done by season and team and required some of the transfer data to be repeated. The processing of the dataset also required different steps to take place. The Random Forest model accepted symbolic data so converting all the data to a numerical format was not necessary. The Random Forest model also did not require the scaling of the data between 0 and 1. The different models were also pre-processed differently depending on if the transfer data was required. Overall, the data manipulation was successful and allowed for a more accurate machine learning model.

5. Generate the model to predict the outcome (Win/Draw/Loss) of football matches in the English Premier League

   This was implemented by creating four models, two using transfer data and two excluding the transfer data for random forest and neural network. The outputs gave us a class of win, draw or loss with a probability of each class being achieved. The challenge here was deciding which machine learning models to use. After conducting the research on previous literature and projects, it was concluded that Random Forest and Neural Networks were the best options to use.

   The first part of the creating the models was finding a way to split the training and testing data. After conducting some research, it was decided that using k-fold cross validation was the best way to optimise the models. It ensured that each row of the combined datasets had the chance to be used as the training and testing data to ensure a fair test.

   The next step was to decide what R libraries to use for the training of the models. For the Random Forest, the grid search to find the best parameters were conducted using the caret library and the RandomForest library was used for training the model with the derived

parameters. This was done as the RandomForest library allowed me to easily choose the hyperparameters and it had been previously used during my university degree.

For the Neural Network, the h2o library was chosen for its ease and efficient when trying to create Deep Neural Network models. It also allowed the models to be saved as a MOJO (Model Object, Optimised) which resulted in much faster predictions and allowed the h2o models to work with any version of h2o. When the models were saved as h2o models, the predictions took up to 12 seconds to complete compared to 4 seconds when using MOJO objects.

To improve the models, the grid search could have been done with more parameter options, however, this was not done in this project as it would have taken significantly longer to train the models due to the grid search being ran on each of the k-folds.

6. Create an RShiny application to allow users to get the results of the predictions

   Once the machine learning models had been created, the next stage was creating a way for the user to select a match and view the predictions for the result whenever they wished. This involved researching which R packages were necessary to be used when creating the application. The main objective was to create a user interface that would be easy to use and visually appealing to enhance the user experience. In order to achieve this, multiple attempts and drafts to create the application were done, experimenting with the layout, design and colours.

   This stage was completed and deployed successfully as a web page. It contained 4 pages, a team page showing each of the team's individual historical results with a map locating its stadium. The historical page showed the historical data including final tables, winners, teams relegated and top 4 finishes. The transfer page displayed the transfer statistics for each of the teams, showing how much they had spent and received over the seasons. The final page was the prediction page where the user selects the teams and the spending's of the teams and the application displays the likelihood of the result with probability of that result occurring.

   The challenge when creating the application was the lack of knowledge of using the Shiny library and features. This was self-taught using YouTube videos and online tutorials. Many different versions of the application were created in order to fulfil the requirements. The initial model lacked colour and then the shinydashboard library was used to add detail and design to the application. Following feedback from my supervisor, the application was altered slightly in order to enhance user experience and make the application more user friendly.

8.2.2 Models and Previous Literature

The results from the metrics of the models show us that the excluding the transfer data gave us a slightly higher accuracy that when we used it. One of the main reasons for this is the wide range of transfer spending data compared to other statistics. Transfer data can vary from anything including -£200m to £200m showing that there is a wide range of values it can take. For example, goals scored ranged from 0 to 9 and only had 10 distinct values. When scaling the data between 0 and 1 the transfer data would have taken on multiple unique values adding to the complexity of the model and making it more difficult to predict the result. Instead of using the raw transfer numbers as they were, in the future the transfer data can be put into bins of data. For example, £0-£10m, £10m-£20 …£190m-£200m, meaning that there would be less distinct values making it easier for the model to understand.

When compared to other researchers' papers, this paper came out successful. Only Hucaljuk and Rakipović [34] achieved better results across the multiple metrics for their models. This

was concluded that the results occurred because Hucaljuk and Rakipović used additional useful features such as the form of the team including their position before the game started and the number of injuries the team currently had.

Whilst this model did not outperform all of the literature reviewed in this report in the validation metrics used, it is believed that the further recommendation and improvements discussed later will help achieve the goal of beating the benchmark papers.

8.3 Strengths and Weaknesses of This Project

Overall, this project had multiple strengths and weaknesses. The weaknesses are ones that would be worked on if this project was to be repeated.

I believe that this project achieved all of the aims and objectives that it set out to do. The project successfully compared machine learning, namely Random Forest and Neural Networks to predict the outcome of Premier League results and analysed how the expenditure and income of the teams affected the predictions. The machine learning models were successfully implemented, and a user interface was created to allow the users to select the teams and the other relevant information to view the predicted result with a confidence measure.

The results of the machine learning models were successfully measured against a set of chosen metrics for this project and compared against previous literature containing different methods for predicting football results. The models created performed better than most of the papers reviewed and where it did not perform better an explanation was given. This project was able to fulfil all the functional and non-functional requirements and provided extra functionality to enhance the user experience. This project has been able to compare multiple machine learning models and a range of hyperparameters for the model.

There are many directions for future extensions of this project in order to improve the predictions and overall outcome given more time and resources. The first thing that could have been done is using improved data. I believe the models could have been improved using more historical data. These can include is the expected goals, this is a statistic that tells us how many goals the team was expected to score during the match using multiple calculations involving the players position on the pitch and the reputation of the person attempting to score the goal. Simple statistics such as possession, number of players injured, number of passes could have helped improve the accuracy of the results. Other statistics could have been sourced to get the average ratings of team using EA Sports player and team ratings.

Additional data could be collected such as the teams' commitments to other cups, for example the top 4 teams play in the UEFA Champions League which could mean that they rest their main players in the Premier League games having an effect on the result. Creating a feature to show the games played by the teams outside the Premier League is on to be considered for future extensions of the project. Another improvement for the data collection could be collecting data from other teams from different countries. For example, the historical results from different leagues, such as the German Bundesliga or the Italian Serie A could be used to expand the application and project. This would increase the target audience as followers of the German and Italian league may see benefit in using the web page. One of the issues with the extending the application with more data would be increasing the features used in the machine learning models or having to create additional models which would have a substantial time increase.

Creating a form feature would have also helped the predictions, this could be done by taking the last 5 games for each of the teams and creating a feature to show how well they were performing. This could be extended to home and away form to see how well the relevant team

has be performing at home in recent fixtures. Another potential future feature that could be used can be the betting odds set by bookmakers. Bookmakers set odds for each game using complex bespoke statistical models, which allow them to make profit. These set odds could be analysed to see how they would compare to actual results of the football matches.

Social media such as twitter is a place where fans voice their opinions and can use their platforms to give an idea on how they think their team may do during the match day. Exploring the tweets could be researched further and Natural Language Processing can be used to perform a sentiment analysis to create an overall evaluation if the people using the platform are positive or negative about the team for that certain game. There are a couple of difficulties that are associated with extracting tweets to determine if they are positive or negative. Fans tweets are all opinion based and some may not be based on facts and statistics. Although this can be countered by the fact that statistics do not contain all the necessary information for the match. The team chosen will have a major impact on the result and using fans tweets may help with the predictions. Kampakis and Adamides [49] used three months of tweets as well as historical data to predict Premier League games and determined that they do have a positive impact on the models. A difficulty would also be creating a model that can be used to extract tweets from twitter. An approach could be using hashtags to get the relevant tweets. This is something that can and will be explored in the future in studies and projects.

To improve the machine learning models, the gird search that was ran to determine the optimal parameters, could be ran with a larger set of options. There is no guarantee that the optimal hyperparameters would be changed, but it would ensure that all the options had been checked. Running a gird search with more options would take a substantially longer time and if more time was available, this could be examined. Other machine learning models could be implemented and compared to the Random Forest and Neural Network to gain a substantial comparison of a wide variety of models.

For the application additional data analysis pages can be added allowing the user to see the trends between the historical results and the transfer data. This was not added as the main objective was to allow the user to get predictions of the Premier League results adding more pages would take the focus away from this. In the team page, the team stadium map could be optimised to allow the user to find direction to the ground by entering their current location or view different types of maps such as terrain, street view or showing public transport routes. Another option to enhance the application could have been the ability for the user to select additional features when selecting a match. For example, as discussed above, if a team's form feature was added to the models, the user would be able to select how the team had recently been performing and output the predictions based of the choices set.

8.4 Academic Contributions

During my BSc Data Science for Health degree at the University of Surrey, there have been many modules that have helped me build up to producing this project. The following models are the ones relevant to this project.

COM1027 – Programming Fundamentals
This module taught me the basics of coding, such as functions, if else statements with relevant labs and exercises.

COM2018 – Software Engineering
This taught me the fundamentals of the Software Development Life Cycle including the functional and non-functional requirements and testing. The steps used for developing the application was based of the knowledge that I gained during this module.

COM2034 – Information Retrieval

This module involved collection of data, understanding the different data types and the fundamental techniques used on online applications for information retrieval. It was also my first experience in using the R programming language.

COM2028 – Artificial Intelligence
This was my first introduction to machine learning and the different techniques that can be used. I was able to create neural networks here and understand the basis behind the complex networks.

Level P – Industrial Placement at Amgen
The year in industry allowed me to develop my R programming skills, understand the libraries and become proficient at data analysis using R. I was also able to work on software development side which helped me in this project including writing and coding to a set of requirements and testing effectively against the requirements.

COM3018 – Practical Business Analytics
I learnt how to create machine learning models including Random Forest and Neural Networks using R programming language. This module involved completing a data analysis task following the steps similar in this project. The first step was data collection, then exploring and manipulating the data. Then machine learning models were created and evaluated using metrics such as accuracy and confusion matrixes. This project followed a similar process with the major differences being previous research being conducted a user interface being crated for displaying the machine learning models.

8.5 Resources Used

There were many resources used for this project, each being used in the different stages when necessary**.**

- **Project Supervisor** – Regular meetings were held as arranged to discuss my progress and get feedback on where I can improve and steps to take. The guidance given allowed for an improvement in the structure of the report and the use of the application.

- **University Library –** This was used for research papers for the literature review

- **Internet –** This was used for relevant resources such as papers relevant to the report and it was used for and programming or machine learning questions that could have been searched online.

- **Laptop –** This was the main device used for the building of the application. The machine learning models were also run on the device.

9 Statement of Ethics

This section contains the statement of ethics for the project and will reflect on the contributions to society and human well-being. This includes the legal, ethical, social and professional considerations of the entire project. Working in an ethical and legal manner is important as it ensures that no harm is done to the public, the data used is not confidential and used in a legal manner. This section will also discuss the relevant data protection principles ensuring that sensitive data must be stored and processed fairly & lawfully, not kept longer than necessary and processed securely. This section will also revie eth BSC Code of Conduct.

I have ensured that this report and project contributes to society and human well-being, avoids harm to others, is honest and trustworthy and is fair and takes action to not discriminate. In addition, this project honours property rights including copyright and patent, gives proper credit for intellectual property and respects the privacy of others. This project does not involve the use of active participation of human subjects and therefore informed consent forms were not necessary.

The three data sets used for this project was from sourced in three different ways from three different sources. The first was the historical results which was sourced from Kaggle which contains 1000's of publicly available datasets. The dataset created for the transfer data of each of the teams, including the expenditure and income was sourced from a publicly available website and Table Capture Google Chrome add on was used to create the tables from the data. The final dataset was the transfer inflation dataset, the data used for this is also publicly available and was created using calculations on the website. The data for all three datasets used were publicly available, free to use and referenced as necessary in the references section. The data collection and usage were all conducted in an ethical and legal matter. All the data for this project was stored in the RStudio server and it not downloadable from the application user so that the data integrity remains, and it cannot be changed to produce false information.

The user of the application may use the Premier League predictions for betting purposes; however, this application was not created to promote gambling and is to be used just for the interest of the user. The application accuracy has been shown in the prediction page and it clearly shows that the predictions will not be 100% accurate. Therefore, the user is at their own risk if they decide to use the predictions for gambling purposes. This application was created to compare machine learning techniques that can be used for predicting the outcome of Premier League matches and how the expenditure and income of the teams can affect these predictions. Its aim was to compare the metrics of the models and allow the user to view the results from this model.

This completion of this project complies with the relevant data protection acts and copyright and patents acts. The source of the datasets is clearly stated along with the use for each of the datasets. The model metrics are also clearly shown for the user of the application to see and therefore does not give the user the thought that the predictions are 100% accurate. Overall, the project was conducted in an ethical and legal manner.

The SAGE form for this project can be seen in the Appendix.

10 Conclusion

Using historical football data is something that has become increasingly popular over the years, with it advancing from simple Poisson distributions to complex machine learning models. Due the complicated and random nature of the sport, predicting the results has always been a difficult task for researchers.

This project has shown that machine learning techniques can be used to predict the outcome of Premier League football matches and has used the expenditure and income of the teams when creating the models. Neural Networks and Random Forest models were created and evaluated against each other, where it was concluded that Neural Network performed better the Random Forest for both the with transfer data and without transfer data models and were hence used in the user interface for displaying the match prediction. The models were benchmarked against previous attempts at predicting football results be researchers and authors and this project performed well in comparison to the other literature.

To address the expenditure and income of the teams, these were included in the machine learning models and compared to models that excluded the data. The results showed that excluding the transfer data performed marginally better than when they were included, and this was concluded that the complexity and wide range of the transfer data was a cause for this. Methods were discussed as to how the transfer data could be used differently in the model, including creating bins for a certain number of ranges and these may have allowed for high accuracies. Additional transfer features are ones that can be explored such as the price the teams bought the players at and contract lengths of the players.

The produced application shows a user interface that explores the data using visualisations and allows for the prediction of the Premier League matches. The project shows that the different machine learning models result in different outcomes and that the features used make a difference in the outputs and accuracy of the results. Football prediction is a topic that will continue to be explored by a large number of researchers and using the team's expenditure and income in different ways is a subject that should be experimented with to try and achieve higher success rates.

List of References

[1] Richter, F., (2021). Infographic: Super Bowl Pales in Comparison to the Biggest Game in Soccer. [online] Statista Infographics. Available at: https://www.statista.com/chart/16875/super-bowl-viewership-vs-world-cup-final/#:~:text=According%20to%20FIFA%2C%20the%202018,in%20over%20its%2090%20minutes [Accessed 12 March 2021].

[2] Alty, R., (2019). “Comparison of World Record Transfer Fees in Football Post-2000”. Leicester Undergraduate Mathematical Journal, Vol. 1 No. 1, pp.1-10.

[3] Preuss, H., Haugen, K.K. and Schubert, M. (2014), “UEFA financial fair play: the curse of regulation”, European Journal of Sport Studies, Vol. 2 No. 1, pp. 33-51.

[4] En.wikipedia.org. List of most-watched television broadcasts. [online] Available at: https://en.wikipedia.org/wiki/List_of_most-watched_television_broadcasts [Accessed 12 March 2021].

[5] Sportingindex.com. Premier League Viewing Figures. [online] Available at: <https://www.sportingindex.com/spread-betting-blog/premier-league-viewing-figures> [Accessed 12 March 2021].

[6] Premierleague.com. Premier League Competition Format & History | Premier League. [online] Available at: <https://www.premierleague.com/premier-league-explained> [Accessed 12 March 2021].

[7] Gulbrandsen, A. M. and Gulbrandsen, C. M. (2011), “Valuation of football players: a complete pricing framework”, Master's thesis, Vol. 1 No. 1, pp. 1-117.

[8] BBC Sport. AC Milan banned from Europe for a year. [online] Available at: <https://www.bbc.co.uk/sport/football/48801681> [Accessed 12 March 2021].

[9] Wilson, R., Plumey, D. and Ramchandani, G., (2013). “The relationship between ownership structure and club performance in the English Premier League.” Sport, Business and Management: An International Journal, Vol. 3 No. 1, pp. 19-36.

[10] Roth, P. and Bischof, H., (2008). “Machine Learning Techniques for Multimedia”. Cognitive Technologies,. Vol. 1 No.1, pp. 21-49.

[11] Li, J., (2019). “Regression and Classification in Supervised Learning”. ICCBD 2019: Proceedings of the 2nd International Conference on Computing and Big Data, Vol. 1, No. 1 pp. 99-104.

[12] Fawagreh, K. et al. (2014) “Random forests: from early developments to recent advancements”. Systems science & control engineering. Vol. 2 No. 1, pp. 602–609.

[13] Verikas, A. et al. (2016) “Electromyographic Patterns during Golf Swing: Activation Sequence Profiling and Prediction of Shot Effectiveness.” Sensors (Basel, Switzerland). Vol. 16 No. 4, pp. 592–.

[14] Georgevici, A. I. & Terblanche, M. (2019) “Neural networks and deep learning: a brief introduction”. Intensive care medicine. Vol. 45 No. 5, pp. 712–714.

[15]Montavon, G. et al. (2018) “Methods for interpreting and understanding deep neural networks”. Digital signal processing. Vol. 73 No.1, pp. 1–15.

[16]MissingLink.ai. Backpropagation in Neural Networks: Process, Example & Code - MissingLink.ai. [online] Available at: <https://missinglink.ai/guides/neural-network-concepts/backpropagation-neural-networks-process-examples-code-minus-math/> [Accessed 12 March 2021].

[17]Rumelhart, D., Hinton, G. and Williams, R., (1986). “Learning representations by back-propagating errors”. Nature, Vol. 323 No. 6088, pp. 533-536.

[18]Dr. Sebastian Raschka. Single-Layer Neural Networks and Gradient Descent. [online] Available at: https://sebastianraschka.com/Articles/2015_singlelayer_neurons.html [Accessed 12 March 2021].

[19]Purucker, M., (1996) “Neural network quarterbacking”. IEEE potentials. Vol. 15 No. 3, pp. 9–15.

[20]Kahn, J., (2003). “Neural Network Prediction of NFL Football Games”. World Wide Web Electronic Publication, pp. 9-15.

[21]Davoodi, E. and Khanteymoori, A., (2010). “Horse Racing Prediction Using Artificial Neural Networks”. Proceedings of the 11th WSEAS International Conference on Neural Networks and 11th WSEAS International Conference on Evolutionary Computing and 11th WSEAS International Conference on Fuzzy Systems, pp.155-160.

[22]Pretorius, A. and Parry, D., (2016). “Human Decision Making and Artificial Intelligence”. Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists on - SAICSIT '16, Vol. 32, pp. 1-10.

[23]Lock, D. & Nettleton, D. (2014) “Using random forests to estimate win probability before each play of an NFL game”. Iowa State University Digital Repository, Vol. 10 No. 2, pp. 197-205.

[24]Maher, M. J. (1982), “Modelling Association Football Scores”. Statistica Neerlandica, Vol. 36, pp. 109–118.

[25]Mark J. Dixon & Stuart G. Coles (1997) “Modelling Association Football Scores and Inefficiencies in the Football Betting Market”. Journal of the Royal Statistical Society. Series C (Applied Statistics). Vol. 46 No. 2, pp. 265–280.

[26]Kuypers, T. (2000) “Information and efficiency: an empirical study of a fixed odds betting market”. Applied economics. Vol. 32 No. 11, pp. 1353–1363

[27]Adam, A., (2016). “Generalised linear model for football matches prediction”. CEUR Workshop Proceedings. Vol. 1842 , pp. 1-7.

[28]Tavakol, M., Zafartavanaelmi, H., and Brefeld, U. (2016). “Feature extraction and aggregation for predicting the Euro 2016”. Machine Learning and Data Mining for Sports Analytics Workshop, pp.1-7.

[29]Dobravec, S., (2015). “Predicting sports results using latent features: A case study”. 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1267-1272.

[30]Y. Koren, R. Bell, C. Volinsky, (2009). “Matrix factorization techniques for recommender systems,”. IEEE 2009. Vol. 42, No. 8, pp. 30-37,

[31]Esme, E. and Kiran, M., (2018). “Prediction of Football Match Outcomes Based on Bookmaker Odds by Using k-Nearest Neighbor Algorithm”. International Journal of Machine Learning and Computing. Vol. 8 No. 1, pp. 26-32.

[32]Yezus, Albina., (2014). “Predicting outcome of soccer matches using machine learning.” pp. 1-12.

[33]Arabzad, S. Mohammad & Araghi, Me & Soheil, Sadi-Nezhad & Ghofrani, Nooshin. (2014). “Football Match Results Prediction Using Artificial Neural Networks; The Case of Iran Pro League”. International Journal of Applied Research on Industrial Engineering. Vol 1, pp. 159-179.

[34]Hucaljuk, J. & Rakipovic, A. (2011) “Predicting football scores using machine learning techniques”, 2011 Proceedings of the 34th International Convention MIPRO. 2011 IEEE. Vol. 34, pp. 1623–1627.

[35]Tax, Niek & Joustra, Yme. (2015). “Predicting the Dutch Football Competition Using Public Data: A Machine Learning Approach”. Transactions on Knowledge and Data Engineering. Vol. 10, pp 1-13.

[36]Pugsee, P. and Pattawong, P., (2019). “Football Match Result Prediction Using the Random Forest Classifier”. Proceedings of the 2nd International Conference on Big Data Technologies - ICBDT2019. Vol. 2, pp. 154-158.

[37]Sujatha, K., Godhavari, T. and Bhavani, N., (2018). “Football Match Statistics Prediction using Artificial Neural Networks”. International Journal of Mathematical and Computational Methods, Vol 3, pp 1-8.

[38]Kaggle.com. English Premier League Results. [online] Available at: https://www.kaggle.com/irkaal/english-premier-league-results [Accessed 12 March 2021].

[39]Transfermarkt.co.uk. Premier League - Transfer income and expenditures. [online] Available at: https://www.transfermarkt.co.uk/premier-league/einnahmenausgaben/wettbewerb/GB1/ids/a/sa//saison_id/2019/saison_id_bis/2019/nat/0/pos//w_s//intern/0 [Accessed 12 March 2021].

[40]Totallymoney.com. The Transfer Index - TotallyMoney. [online] Available at: https://www.totallymoney.com/content/transfer-index/data/ [Accessed 12 March 2021].

[41]Ethen8181.github.io. model_selection. [online] Available at: http://ethen8181.github.io/machine-learning/model_selection/model_selection.html [Accessed 12 March 2021].

[42]Igea, A., (2019). “Machine Learning Algorithms for Sports Prediction”. Applied Computing and Informatics. Vol. 15 No. 1, pp. 27-33

[43]Badriyah, T., Santoso, D. and Syarif, I., (2019). “Deep learning algorithm for data classification with hyperparameter optimization method”. Journal of Physics: Conference Series, Vol. 1193, pp 1-9.

[44]H2o.ai. 2021. [online] Available at: <https://www.h2o.ai/wp-content/uploads/2018/01/RBooklet.pdf> [Accessed 12 March 2021].

[45]Docs.h2o.ai. (2021). Deep Learning (Neural Networks) — H2O 3.32.0.4 documentation. [online] Available at: http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/deep-learning.html [Accessed 12 March 2021].

[46]Breiman, L., (2001). “Random Forest”. Machine Learning, Vol. 45 No. 1, pp. 5-32.

[47] Rstudio.com. (2021). RStudio | Open source & professional software for data science teams. [online] Available at: <https://www.rstudio.com/> [Accessed 27 April 2021].

[48]Shiny.rstudio.com. (2021). Shiny - Getting started with shinyapps.io. [online] Available at: <https://shiny.rstudio.com/articles/shinyapps.html> [Accessed 25 April 2021].

[49]Kampakis, Stylianos & Adamides, Andreas. (2014). “Using Twitter to predict football outcomes”. arXiv. pp 1-10.

Appendix

SAGE-HDR

Response ID	Completion date
640816-640807-77983791	11 May 2021, 18:15 (BST)

1	**Applicant Name**	Varun Kale
1.a	**University of Surrey email address**	vk00205@surrey.ac.uk
1.b	**Level of research**	Undergraduate
1.b.i	**Please enter your University of Surrey supervisor's name. If you have more than one supervisor, enter the details of the individual who will check this submission.**	Mark Manulis
1.b.ii	**Please enter your supervisor's University of Surrey email address. If you have more than one supervisor, enter the details of the supervisor who will check this submission.**	m.manulis@surrey.ac.uk
1.c	**School or Department**	Computer Science
1.d	**Faculty**	FEPS - Faculty of Engineering and Physical Sciences Sciences

2	**Project title**	Using machine learning to predict football match outcomes based on clubs' transfer spendings

3	**Please enter a brief summary of your project and its methodology in 250 words. Please include information such as your research method/s, sample, where your research will be conducted and an overview of the aims and objectives of your research.**	Creating machine learning models, namely Neural Networks and Random Forest to predict the outcome of Premier League Football Matches. The datasets used are a historical results one, a transfer dataset and a transfer inflation dataset. The overall aim of this project is to investigate the machine learning techniques that can be used for predicting the outcome of football results and how the expenditure and income of teams can affect these predictions. Different models will be created and compared; one will contain the transfer data for each of the teams and one will exclude the expenditure and income statistics.

4	**Are you making an amendment to a project with a current University of Surrey favourable ethical opinion in place?**	NO

5	**Does your research involve any animals, animal data or animal derived tissue, including cell lines?**	NO

7	**Does your project involve any of the following: human participants (including human data and/or any human tissue*); or is your project linked to engineering and/or the physical sciences?**	NO

8	**Will you be accessing any organisations, facilities or areas that may require prior permission? This includes organisations such as schools (Headteacher authorisation), care homes (manager permission), military facilities etc. If you are unsure, please contact RIGO.**	NO

9	**Will you be working with any collaborators or third parties to deliver any aspect of the research project?**	NO

10	**Is your project a service evaluation or an audit?**	NO

11	**Does your funder, collaborator or other stakeholder require a mandatory ethics review to take place at the University of Surrey?**	NO

12	**Are you undertaking security-sensitive research, as defined in the text above?**	NO

30	**Declarations**	• I confirm that I have read the University's Code on Good Research Practice and ethics policy and all relevant professional and regulatory guidelines applicable to my research and that I will conduct my research in accordance with these. • I confirm that I have provided accurate and complete information regarding my research project • I understand that a false declaration or providing misleading information will be considered potential research misconduct resulting in a formal investigation and subsequent disciplinary proceedings liable for reporting to external bodies • I understand that if my answers to this form have indicated that I must submit an ethics and governance application, that I will NOT commence my research until a Favourable Ethical Opinion is issued and governance checks are cleared. If I do so, this will be considered research misconduct and

		result in a formal investigation and subsequent disciplinary proceedings liable for reporting to external bodies. • I understand that if I have selected any options on the higher, medium or lower risk criteria then I MUST submit an ethics and governance application (EGA) for review before conducting any research. If I have NOT selected any of the higher, medium or lower risk criteria, I understand I can proceed with my research without review and acknowledge that my SAGE answers and research project will be subject to audit and inspection by the RIGO team at a later date to check compliance

31	**If I am conducting research as a student:**	• I confirm that I have discussed my responses to the questions on this form with my supervisor to ensure they are correct. • I confirm that if I am handling any information that can identify people, such as names, email addresses or audio/video recordings and images, I will adhere to the security requirements set out in the relevant Data protection Policy