2019

Using Supervised Learning to Predict English Premier League Match Results From Starting Line-up Player Data

Runzuo Yang

Technological University Dublin

Using supervised learning to predict English Premier League match results from starting line-up player data

Student Name

Runzuo Yang

A dissertation submitted in partial fulfilment of the requirements of
Dublin Institute of Technology for the degree of
MSc. in Computing (Data Analytics)

2019

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Data Analytics), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the test of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Dublin Institute of Technology and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed: ___________Runzuo Yang______________________

Date: 03/01/2019

ABSTRACT

Soccer is one of the most popular sports around the world. Many people, whether they are a fan of a soccer team, a player of online soccer games or even the professional coach of a soccer team, will attempt to use some relevant data to predict the result of a match. Many of these kinds of prediction models are built based on data from the match itself, such as the overall number of shots, yellow or red cards, fouls committed, etc. of the home and away teams. However, this research attempted to predict soccer game results (win, draw or loss) based on data from players in the starting line-up during the first 12 weeks of the 2018-2019 season of the English Premier League. It covered their ICT index, influence, creativity, threat and BPS index, cost and selection by using supervised Machine Learning techniques, namely Random Forest, Naïve Bayes, K-Nearest Neighbour and Support Vector Machine. As a result of the research, it was determined that Random Forest was the best classifier in this project. Influence, creativity, threat, BPS index and selection were the most suitable features in this model, achieving an accuracy level of approximately 80%. On this basis, apart from predicting the results, this model can also provide strategies for coaches, fans and online soccer game players regarding which kinds of features and positions of players have an essential influence on the final result, thus affecting how they assign starting line-up.

Key words: *Random Forest, Naïve Bayes, K-Nearest Neighbour, Support Vector Machine, Feature Selection, Players' data.*

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere thanks to my supervisor Prof. Sarah Jane Delany. She is the Head of Postgraduate Studies & Research, School of Computer Science that has a lot of work and meetings to deal with every day, but she still arranges a meeting for me every week to guide me carry out the dissertation and patiently answer my any questions. Let me learn how to logically and critically consider issues and solve problems, this thinking method not only helps my current dissertation writing but also affects me in the future study and work. Genuine appreciation from my heart to my supervisor Prof. Sarah Jane Delany.

Besides, I am also grateful to my initial supervisor Mr. Brian Leahy. Although he cannot be my supervisor because of his significant increased workload, he still gave me a lot of inspiration at the beginning of the dissertation.

At the same time, I am thankful for Dr. Luca Longo who is the dissertation coordinator directed me how to write the dissertation scientifically and rigorously, Mr. Trevor Conway who helped me fix the mistakes in English proofreading. And all others DIT tutors and classmates during my postgraduate semesters, they taught me the knowledge of data analysis and helped me understand this domain better.

Finally, I am going to express my gratitude to my family and friends for comforting and encouraging me to overcome difficulties when I met problems in study and life.

Thanks to all the people who helped me.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1. INTRODUCTION

1.1 Background

Soccer is one of the most popular sports around the world with a significant number of fans (Razali, Mustapha, Yatim & Ab Aziz, 2017), therefore, predicting the result of each match is an attractive and exciting thing for audiences to speculate the competitions and bet respective team. On the other hand, predicting the actual outcomes of soccer games can also give a series of practical suggestions for the football club to improve their matches strategies, and has insight into their rivals. The earliest human team activity with the ball occurred in ancient Mesoamerican cultures over 3000 years, and the original precursors of soccer game took place in ancient China between the 3rd and 2nd century BC[1]. However, the beginning of the modern soccer was in England in 1863[2]. Nowadays, the English Premier League is the top level of English soccer organization and one of the most powerful leagues in the international soccer field. Each team in this league has the same chance to win the final championship. A top team may occasionally fail to win a match when it competes against a weak team. Different teams will arrange diverse starting line-up against various types of opponents. This research will focus on the statistics of players in the starting line-up in each match over a certain period, and it will use these features to build a series of models, predicting match results by using supervised Machine Learning techniques.

There are many algorithms in Machine Learning that have been applied to the prediction of sports especially soccer competitions. Bayesian Network is an appropriate method to build and develop predicting models in soccer games (Zhao & Xie, 2015). As a consequence of the dataset, it has a series of sophisticated features with quite small sample sizes, handling missing values and avoiding overfitting issues (Uusitalo, L. 2007). Naïve Bayes which is another approach based on the Bayesian theorem that is commonly used in classification cases, and it can calculate the distribution of each class in target variable (Hai, M., Zhang, Y., & Zhang, Y. 2017). Moreover, the K-nearest neighbours' algorithm, Random forest, LogitBoost, and Artificial neural networks also applied to achieve and compare the accuracy of prediction results in soccer games (Hucaljuk, J. & Rakipović, A. 2011). In order to discuss the performance of feature selection and variables' correlation coefficient, regression models can also be taken

[1] https://www.footballhistory.org/

[2] https://www.fifa.com/about-fifa/who-we-are/the-game/

advantage of predicting Australian football which is similar to British soccer (Jelinek, Kelarev, Robinson, Stranieri & Cornforth, 2014).

1.2 Research Project

This research aims to predict matches results based on data from players in the starting line-up of the English Premier League by using supervised Machine Learning techniques. As a result of that, the research question as follows:

"Can people use the data of players in starting line-up to predict soccer game results by using supervised Machine Learning techniques?"

The data for this the research was collected from the first 12 match weeks of the 2018/2019 season of the English Premier League, involving a total of 120 games (10 matches each week). Information regarding home and away teams' names, starting line-up lists and final results were acquired from https://www.premierleague.com/results. The personal data of the players was obtained from https://fantasy.premierleague.com/. The research will choose the classifier with the highest prediction accuracy to build the training model from Random Forest, Support Vector Machine, Naïve Bayes and Nearest Neighbour. Subsequently, it will use this classifier for feature selection to select the most appropriate features. Eventually, the model will evaluate and analyse the prediction results and summarise the importance of various features of players in different positions. As a result, the final model can provide suggestions and solutions for online players or fans, and even professional coaches.

1.3 Scope and Limitations

This research was completed based on data from the first 12 weeks of the Premier League 2018-2019 season. Due to the limited size of the dataset, it can only be used for the current season; this model cannot be applied to other seasons. Furthermore, this project mainly used the starting line-up, so the performance of substitute players and the changing dynamics of starting players' positions were not reflected. To test the feasibility and stability of this project, an extra experiment was applied to predict the results from the 13th to the 15th weeks, based on the previous 12 weeks, because the season is not over yet, and the 13th to 15th weeks included the latest matches at the time of writing.

1.4 Document Outline

This research is organised as follows.

Chapter 2 discusses, compares and summarises various literature which used different Machine Learning techniques to predict the results of soccer games, determining their advantages and disadvantages. It also explains the methodologies of the Machine Learning techniques, along with the feature selection used in this project.

Chapter 3 concentrates on the general structure of this prediction system, regarding which kinds of techniques should be used and the methodologies adopted for each step, from data selection to model evaluation.

Chapter 4 describes each model used for each experiment in this project in a step-by-step fashion, interpreting their modelling, parameter-fixing and predicting processes. Evaluating and analysing the results and findings from this series of experiments.

Chapter 5 aims to give a general review of the entire project. It discusses the final result and conclusion, highlighting the limitations and problems relating to this project, and it gives suggestions for future work in the effort to improve research in this field.

2. LITERATURE REVIEW AND RELATED WORK

In this chapter, several fields of literature will be listed and be discussed, covering Machine Learning algorithms, datasets, features, etc., to compare and conclude their study results or findings. This will be done to find suggestions and inspirations from previous work to improve the prediction accuracy of soccer matches' results for this project. In addition, this chapter will be divided into two parts. Part one is from **Section 2.1** to **Section 2.2** which focuses on the methodology of Machine Learning algorithms, feature selection. This part describes the principles of Machine Learning algorithms and related work and how they will be used in this project. Part two is from **Section 2.3** to **Section 2.4** which reviews how other researchers apply Machine Learning techniques and select or collect data sets for prediction in the field of sports, especially in soccer games.

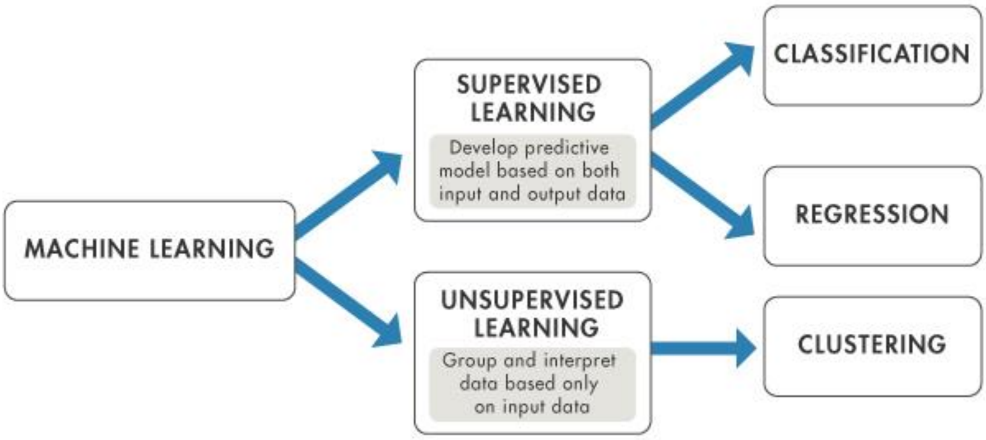

MACHINE LEARNING
SUPERVISED LEARNING
Develop predictive model based on both input and output data
UNSUPERVISED LEARNING
Group and interpret data based only on input data
CLASSIFICATION
REGRESSION
CLUSTERING

Fig 2.1 Branches of Machine Learning

As **Fig 2.1** shows, Machine Learning is classified as supervised and unsupervised learning. The sports industry always belongs to the former one, because people can acquire both input and output information frequently. As a result, people may develop an available model to attempt to predict the outcome for the next time. Besides, this project will use supervisory classification to predict results because of the existence of a categorical target variable. The basic processes are as **Fig 2.2** illustrates:

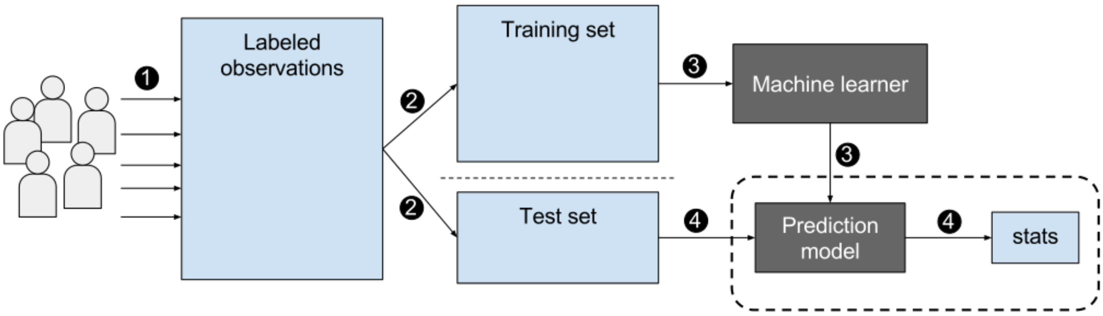


Fig 2.2 Processes of Supervised Machine Learning

Firstly, the Machine Learning model acquires labelled observations from the raw data source and then adjusts the quality of them and after that the system divides them into the training set and test set selecting suitable Machine Learning algorithms to build the prediction model based on the training set. The label of observations should be removed in the test set. Finally, the experiment will predict the results by using the training model, comparing the result with the label-removed test set and calculating prediction accuracy afterward.

After a brief explanation of the workflow of supervised Machine Learning, the next **Section 2.1** (from **Section 2.1.1** to **Section 2.1.4)** is going to describe four kinds of favourite Machine Learning algorithms which frequently appear in the literature to be discussed, and a Machine Learning feature selection in **Section 2.2** which is an essential step in the entire Machine Learning process.

2.1 Machine Learning – Algorithms

The following subsections are going to introduce four supervised Machine Learning algorithms which will be applied in Chapter 4.

2.1.1 Random Forest

Random forest is an ensemble Machine Learning method for classification by using the decision tree as a basic learner device to build bagging and further introduces random attributes in the training process of the decision tree. The algorithm procedures are:

(1) Assume that there is a data set $D = \{x_{i1}, x_{i2}, \ldots, x_{in}, y_i\}\,(i \in [1, m])$ that has N number of features, the samples under bootstrap sample $(m * n)^{m*n}$ can generate sampling space

(2) Building a base learner (decision tree): sampling each one like

$d_j = \{x_{i1}, x_{i2}, \ldots, x_{ik}, y_i\}\ (i \in [1, m])$ (k should be less than m) to generate the decision tree and record each result of it as $h_j(x)$

(3) Training T times let $H(x) = max \sum_{t=1}^{T} \phi(h_j(x) = y)$ in the formula that

$\phi(x)$ which is a kind of decision algorithm (includes absolute majority voting, plurality voting, and weighted voting, etc.)

In (1) and (2) steps, the input samples in each tree will not include the whole sample, and each decision tree is established in a completely split manner, so that one leaf node of the decision tree cannot continue to split, or all the samples in the same class are directed to the same classification. Both of them ensure the randomness of sampling which does not need to prune the branches and can avoid the problem of overfitting problem.

Random forest does not need to adjust too many parameters compared with other machine semester algorithms. Baboota & Kaur (2018) summarized that the most parameters that needed to be optimized for their random forest model were the number of trees to build, the splitting criterion to consider, the maximum depth of each tree and the minimum sample split. Ulmer & Fernandez (2014) tuned the number of estimators and minimum sample required to split an internal node by using the grid search, acquiring the third lowest error rates in their models' comparison.

2.1.2 Support Vector Machine

Baboota & Kaur (2018) provided a brief explanation for non-linear classification in the Support Vector Machine. SVMs have an excellent performance in dealing with high-dimensional feature spaces, because, through some pre-selected non-linear mapping (kernels trick), this transforms it into a linear separable dimension in a high-dimensional space, and constructs an optimal classification hyperplane in this high-dimensional space. The formula of non-linear SVM is as follows **Equation 2.1**:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i y_i \langle \emptyset(\boldsymbol{x_i}), \emptyset(\boldsymbol{x}) \rangle + b$$

Equation 2.1

The mapping of φ from the input space (X) to a specific feature space (F) means that the establishment of a non-linear learner is divided into two steps: The first step is transforming the data into a feature space F using a nonlinear map. Then the second one is using the linear learner classification in the feature space.

Kernel is a way to calculate the inner product directly in the feature $< \phi(x_i), \phi(x) >$ space and combine the above two steps to build a linear learner. In a word, Kernel is like a function K, for all $x, z \in X$, it should be satisfied with $k(x,z) =< \phi(x_i), \phi(x) >$ and φ (·) has the same meaning as φ in the non-linear SVM. Calling a method of replacing an inner product with a kernel function is named kernel trick. Baboota and Kaur selected the radial basis kernel and the linear kernel in their project tuned the two main hyper-parameters for SVM models which are C and Gamma. The C is the cost of misclassification on the training data, the lower value represents a smooth decision surface. In contrast, the higher value shows that the model needs more cases to support vectors to classify all training cases correctly (Ancona, Cicirelli, Branca & Distante, 2001). Consequently, a suitable C should keep a balance between under-fitting and over-fitting. Gramma comes from the Gaussian radial basis function: a higher Gamma will lead to a small variance and a high bias that reflects the support vector's lack of extensive influence.

This project is going to develop a comparison function in R language which is used to select the least error rate of four common kinds of kernels function. Except for the two types which were mentioned in the previous $k(x,y) = \left(ax^T y + c\right)^d$,there are Polynomial kernel and Sigmoid kernel as well, the Polynomial kernel is non-fixed kernels, and it is ideal for normalizing all training data, which calculate formula as $k(x,y) = \tanh\left(ax^T + c\right)$ follows: generally, it is not appropriate to choose a high dimension. The most suitable dimension needs to be selected by cross-validation. The Sigmoid kernel function is calculated as follows: which is a common S-type function derived from neural networks and now heavily used for deep learning. When there is a kernel trick, the support vector machine implements a multi-layer perceptron neural network, applying the SVM method, the number of hidden layer nodes (which determines the structure of the neural network), and the weight of the hidden layer nodes to the input nodes. Values are automatically determined during the design (training) process. Moreover, the theoretical basis of the support vector machine determines that it finally obtains the optimal global value rather than the local minimum, and also guarantees its good generalization ability for unknown samples without over-learning. Apart from that, as the precondition of modelling, the dataset should be normalized so that all of the value should fall in between 0 to 1.

So far, this part has discussed how to deal with independent variables in SVM, and this paragraph will interpret the dependent variable in this dataset. The SVM algorithm was initially designed for the binary classification problem. When dealing with multiple types of issues, it is necessary to construct a suitable multi-class classifier. At present, there are two main methods for creating SVM multi-class classifiers. (1) The direct method, directly modifies the objective function, merge the parameter solutions of multiple classification surfaces into one optimization problem, and realize multi-class classification by solving the optimization problem "one-time". This method seems simple, but its computational complexity is relatively high, and it is difficult to implement. It is only suitable for small problems. (2) The indirect method mainly performs the construction of a multi-classifier by combining a plurality of two classifiers, and the conventional techniques are one against one and one against all. In the training step of one against one, the samples of a particular category are classified into one class, and the other remaining samples are classified into another class, so that the samples of the K categories construct K SVMs. When classifying, the unknown sample is classified as the one with the most substantial classification function value. However, this method has a drawback because the training set is 1: M that always produce quite obviously biased data in the other classifier with remaining samples. The second approach is to design an SVM between any two types of samples, so K samples need to develop K(K-1)/2 SVMs, when an unknown sample is classified, the category with the most votes last is the category of the target. In this project, the target variable has three kinds of classifications (W, L, and D). The process by using one against all is: In the beginning assume that W=L=D=0. Selecting W and L to build the model, if W has a higher prediction accuracy, W win the comparison and W=W+1, otherwise, L=L+1. Selecting W and D to build the model, if W has a higher prediction accuracy, A win the comparison and W=W+1, otherwise, D=D+1. Selecting D and L to build the model, if D has a higher prediction accuracy, D wins the comparison and D=D+1, otherwise, L=L+1. The final result is the Max(W, L, D). Because there are only three categories in the project, it only builds 3(3-1)/2 = 3 SVMs, in the end, one against all will be applied in these multiple SVM models.

2.1.3 Naïve Bayes

Naive Bayes classification uses a probabilistic classifier to predict maximum likelihood-based results. The model assumes that all variables in the dataset used to predict the target value are independent. The classification model is based on the assumption that the value of a feature in the dataset does not depend on the values of other features in the dataset. It focuses on the dependent variable and then thinks over the probability of the given value that independent variables have, determining which one has the highest probability. The dependent variable will fall in that classification (Hijmans & Bhulai, 2017).

The basic theory of the Naive Bayesian classification is as follows:

(1) Firstly, selecting a known classification of items to be classified as training samples C and assume that the category set of the sample is represented as $C = (y_1, y_2, \dots, y_m)$ Sample C has n discrete features, expressed as: $X = (x_1, x_2, \dots, x_n)$, any x_j is the feature attribute of the sample.

(2) Secondly, calculating separately $P(y_1|x_1x_2\dots x_n)$, $P(y_2|x_1x_2\dots x_n)$,..., $P(y_m|x_1x_2\dots x_n)$;

If any individual C in the training C sample is satisfied with the following $P(y_k|x_1x_2\dots x_n) = max\left\{P(y_1|x_1x_2\dots x_n), P(y_2|x_1x_2\dots x_n), \dots, P(y_m|x_1x_2\dots x_n)\right\}$ formula:

it can be regarded as $c \in y_k$ and then according to the Bayes' theorem **Equation 2.2**:

$$P(y_k|x_1x_2\dots x_n) = \frac{P(x_1x_2\dots x_n|y_k)P(y_k)}{P(x_1x_2\dots x_n)}$$ **Equation 2.2**

to calculate the conditional probability of each category of the sample. Because the denominator is the same value for all categories, it can be omitted. Whichever, one has the highest numerator that is the suitable class for the target variable. Although the assumption that "all features are independent of each other" is unlikely to be right in reality, it can significantly simplify the calculations, and studies have shown that the accuracy of the classification results has little effect.

2.1.4 K-Nearest Neighbour

Hucaljuk & Rakipović (2011) illustrated that the K-nearest neighbour is the representative algorithm of lazy classifiers which classifies a new example by finding the k nearest neighbours in the space of features such as the Euclidean distance measurement from others examples in the existing learning set. After that, based on the learning set, to make a vote determines the classification of the unknown case. The concept as described is to find the appropriate K value and how to select the calculation methods are essential steps in the K-nearest neighbour.

2.2 Machine Learning – Feature Selection

The goal of feature selection is to maximize the extraction of features from raw data for use by Machine Learning algorithms and models. According to the form of feature selection, there are three popular methods nowadays:

Filter: The Filter method, which scores each feature according to divergence or correlation, sets the threshold or the number of thresholds to be selected, and selects features.

Wrapper: A wrapper method that selects several features at a time, or excludes several features, based on an objective function (usually a predictive effect score).

Embedded: The embedding method, which first uses some Machine Learning algorithms and models to train, obtains the weight coefficients of each feature and selects features according to the coefficients from the large value to the small value. It is similar to the Filter method, but it is trained to determine the pros and cons of the feature.

When comparing the first two methods, the filtering method does not consider the effect of the feature on the learner when selecting features, but the wrapped selection is more flexible. Parcels are usually “tailor-made” feature subsets for learners based on predictive performance scores. Compared with filtering methods, learners can perform better. The disadvantage is that the computational overhead is often higher.

2.3 Machine Learning in sports

Machine Learning algorithms are more and more widely and frequently used in competitive sports. They use historical records, as well as live real-time game data, to build models that predict what might happen in the future. Some studies have proved that the application of Machine Learning in sports has established a systematic approach and has achieved quite good results and experience. The following sections are the guided review of these kinds of literature.

2.3.1 Algorithms Comparisons

In this section, the literature review will mainly focus on the Machine Learning algorithms used in several sports, but the majority materials are only relevant to soccer matches and describe their dataset and features as well.

Soccer matches are one of the most popular branches of sport prediction. There are other kinds of ball games which are similar to soccer that are referred to in this research. Delen, Cogdell & Kasap (2012) predicted the NCAA bowl outcomes by using artificial neural networks, decision trees and support vector machines based on eight seasons' data and 36 variables. Continuous variables are inputted from the home team's perspective by calculating and using the different values between home and away teams. Their scenario is building and comparing the direct classification and regression-based classification models which output both wins and losses. Finally, decision trees which represent the direct classification method got better than an 85% prediction accuracy that defeated the other one. Leung & Joseph, (2014) attempted to predict the same target as Delen, Cogdell and Kasap, however, they explored the results of similar level teams, using their data to predict the result of this team with the same opponent instead of comparing these two teams directly. Leung and Joseph made the model compared with the exciting models from previous researches they referenced and even got 97.14% accuracy, this high accuracy may be because of a potential multicollinearity problem during the feature selection step, considering more reliable reference models could also be adopted in further work.

Returning to the soccer domain Min, Kim, Choe, Eom & (Bob) McKay (2008) proposed a Football Result Expert System (FRES) which predicted the soccer result based on a multiple framework composed by Bayesian networks and a rule-based reasoner. Owramipur, Eskandarian & Mozneb (2013) applied a similar construction to predict the result of Spanish League match involving Barcelona as well. The features classification is different from Min, Kim, Choe, Eom & (Bob) McKay (2008) in that their variables are divided into psychological data such as weather, historical records, psychological condition, etc. and non-psychological data such as the average age of players, average goals in each match, average matches each week, etc. The FRES can give a somewhat steady and reliable output between two clubs which was seldom encountered in previous matches. The authors mentioned that the Bayesian models are good at merging uncertain human knowledge and probative knowledge. However, their construction of the knowledge is inclined to a subject activity. It cannot be applied without expert knowledge, and this is a common limitation in a knowledge-based system. FRES by from coach's perspective, organizing Bayesian networks for offense, defence, possession and fatigue strategies, generating optimum discrete values from each section and passing to the rule-based reasoner to adjust the initial output and carry out entire appropriate strategies. Bayesian networks and other technologies can also operate the parameter learning to tune each position's output automatically, combine it with the rule-based reasoner to establish the fundamental team knowledge probably producing a more reasonable strategy for the football team.

Hijmans & Bhulai (2017) worked on predicting Dutch football by using Machine Learning classifiers along with random forest, Naïve Bayes and the k-nearest neighbour models. Their work had an interesting result, finding that the tactics of the team coach do not have much effect on the final result of a match. The dataset is composed of three types of matches which are friendly, qualification and tournament, with the details of individual players adopted in it as well. In the random forest models, the authors applied the generalized boost method, which can synthetically use weak predictors and generate a series of constraints for each node of decision trees to control the random outputs or overfitting issues, testing different nodes to find out the best fit tree. The prerequisites for Naïve Bayes are that one variable will not depend on other variables in the same dataset. The case will be applied to whichever class has the highest probability. For instance, the following **Equation 2.3** is acquired from the model:

$$Win = P(win)*P(age|win)*P(attrackers|win)*P(home|win) / normalization\ constant$$

Equation 2.3

In this research, the authors chose to use the maximum or minimum value replacing the outliers during the data cleaning stage instead of using the average value, as this might influence the final accuracy and methods selection of experiments.

The Bayesian networks models are frequently mentioned in research studies. Joseph, Fenton & Neil (2006) used and compared expert Bayesian networks, MC4 (which identifies factors with the most significant effect on the match result), k-nearest neighbour, Naïve Bayes and Data-Driven Bayesian (which entirely learns from the dataset) to predict soccer results. They focused on Tottenham Hotspur Football Club and used 1995 to 1997 season's data to constitute the dataset which is the first time to develop the expert Bayesian networks in the English Premier League, so it was an uncommon opportunity for making a comparison between expert Bayesian networks models and others Machine Learning models directly. It is a distinctive point, but it also means the dataset is probably quite old and rare at the same time. The expert BN picked some key pieces of information from several core players as the fundamental parameters. Others used more players' information (position, attendance, and performance) as a result of the experiment, with the complete two seasons as the dataset KNN got the best performance when disjoint training and test data, the expert BN won the competition. However, the biggest disadvantaged of the expert BN model is that players might change their position or even change their football club in their career. Hence it cannot provide a sustainable use for a long time. Besides, expanding data from other teams in the league could help to construct more symmetrical models for Bayesian networks to strengthen their accuracy and stability (their results are in the range of 38% to 59% so far).

96% is one of the highest average prediction accuracies found in one research study by (Martins et al., 2017). Their dataset was collected from different soccer leagues with different seasons, such as England, Spain, and Brazil, from 2010 to 2015. They introduced a polynomial classifier which used polynomial algorithms to expand input data in an advanced dimension and separate analysed classes and output as nonlinear data. Making a comparison between the support vector machine, Naïve Bayes and decision trees, it costs more time than others to dispose of dimensionality problems with multiple features, due to the complicated procedures; consequently, this system is not suitable to be used in real-time prediction at present.

2.3.2 Feature selection

Feature selection is the primary and initial step for a Machine Learning model which affects the decision of project objectives and the quality of models' results. As mentioned in the previous **section 2.3.1**, Joseph, Fenton & Neil (2006) used players' data from the Tottenham Hotspur football club, but other researchers prefer to select the data from matches themselves. However, all the statistics during the matches are made by each player. This project and the following researchers that will now be discussed chose a different aspect of the dataset and features based on information of players to explore more interesting or useful performances.

Pariath, Shah, Surve & Mittal (2018) considered their system from the perspective of coaches and team management, estimated and generated a performance value for one soccer player from his value budget, competitiveness, position and skills in his individual career. As a result of that, they scrapped data which included 21280 players with 36 attributes from the grassroots level of players in India from the 2017 version of EA sports. The overall performance accuracy can reach 84.34%, and market value prediction accuracy is around 91% under the linear regression model. During the modelling step, Pariath, Shah, Surve & Mittal tried to separate players in a different position (Forward, Midfielder, Defender and Goalkeeper) which provided a balanced exploration for players in their proper and individual standard.

Some researchers mainly concentrated on English Premier League research. Their datasets ranged approximately from 2006-2007 to 2012-2013 seasons. Bush, Barnes, Archer, Hogg & Bradley (2015) investigated specific position evolution of players and generated relevant parameters, to evaluate match performance, their project can also simulate the view of coach to arrange the squads. The Genetic Programming system which produced a series of GP-generated functions according to different parameters, weights, and settings, followed a majority voting method that combined superior quality functions to get better prediction (Cui, Li, Woodward & Parkes, 2013). Archer, Hogg & Bradley (2015) combined 43 GP-generated functions and got an average predicting accuracy around of 75% eventually which had a more excellent performance than ANN's result.

McHale & Relton (2018) identified the key players in soccer teams by using network analysis and pass difficulty. They acquired professional datasets from Prozone which is a company dedicates sports data and related technologies. The dataset includes 380 matches with the tracking data of players covering passes, tackles, dribbles, and shots, etc. in 2012 to 2013 season for English Premier League. The pass difficulty is defined as the probability of the successful pass by using a weighting scheme which can identify who has the highest threat to get a score for the attacking team. McHale & Relton also examined more positions than others researchers. The goalkeeper, central back, full back, wide midfield, central midfield and attacker in their model which is much closer to the actual matches. However, this research mainly considered the total distance and the total number of sprints by each team as the target which cannot represent diverse strategies in the English Premier League, and different scores during the match will lead and change their running distance and sprints in the particular period.

Sarangi & Unlu (2010) conducted a similar study to McHale & Relton's (2018), where they collected data from the UEFA Euro 2008 Tournament and explored how the players' contribution affected their team and their salaries, constructed a team network analysis based on individual actions and interactions between players. Passing and receiving are important indexes in helping to calculate the team's intercentrality measure which is regarded as the final team performance index. As a result, the key player in a team is always the person who keeps the most frequent interaction with teammates, not the person who has the maximum kicks at goal. Sarangi & Unlu did not divide players into their specific position as it was not suitable for the exploration of the team line-up during a match. Using defensive data like tackling and dribbling data will more scientific to assess the strong probability of interaction between players.

2.4 Future Development

There are also some new and unique ideas that can be applied in future design and improvement for this project which have been successful modelling cases as described in the following paragraphs.

Lu, Chen, Little & He (2018) carried out their project from a different kind of dataset. They extracted information from images and videos of different sports which consist of soccer, basketball, ice hockey. They used the convolutional neural network (CNN) to classify and predict team memberships for both teams or specific positions in the line-up on the field, which can be developed in the future and more profound experiments for this graduation project.

On the match field, not only are there 22 players from the home and away teams but also have professional referees. Most researchers will not select the data from the statistic of referees to predict matches results, which might be a new field for further exploration. Weston, Castagna, Impellizzeri, Rampinini & Abt (2007) proposed three variables: 1) total distance covered, 2) high-intensity running distance whose running speed more than 5.5m/s and 3) average distance from infringements will influence the physical performance of referees and tactical strategies on behalf of the referees, even the tempo of match.   Weston, Bird, Helsen, Nevill & Castagna (2006) also illustrated the second half match time would change the standard of judgment and intensity of competition. All of these factors have the potential to decide the matches' result.

This chapter described in detail the technical principles involved in this research and the application of Machine Learning in sports competitions, especially soccer games in the literature review. The following chapter will introduce the design and process details of the prediction model.

3. DESIGN AND METHODOLOGY

In the third chapter, the research will discuss the general process of how to use players' data in the starting line-up to predict the result in English Premier League. The model describes details of original data, for instance, the reason why the project selected the dataset, how to pre-process different features in the entire dataset, and explain the meaning of each feature.

3.1 Design Outline

Both Chapters 3 and 4 will develop their work according to the design outline as following **Fig 3.1**:

(1) Data collection - Describe the data source and features meaning from perspectives of both matches and players in Chapter 3.

(2) Data construction - Explain the structure of datasets (from 1 to 4 features for players) and demonstrate the data segmentation in Chapter 3.

(3) Cross validation – Interpret its operating principle in Chapter 3. Use this method to divide the training set and test set in order to get the most reliable and stable model in Chapter 4.

(4) Algorithms comparison - Illustrate individual parameters for tuning for each algorithm and select the best classifier for feature selection in Chapter 4.

(5) Feature selection - Describe the methodology of feature selection in Chapter 3. Select the excellent features which can generate the highest prediction accuracy in Chapter 4.

(6) Final Model - After determining the final model, two kinds of analysis include matches and players will be provided for people as a reference.

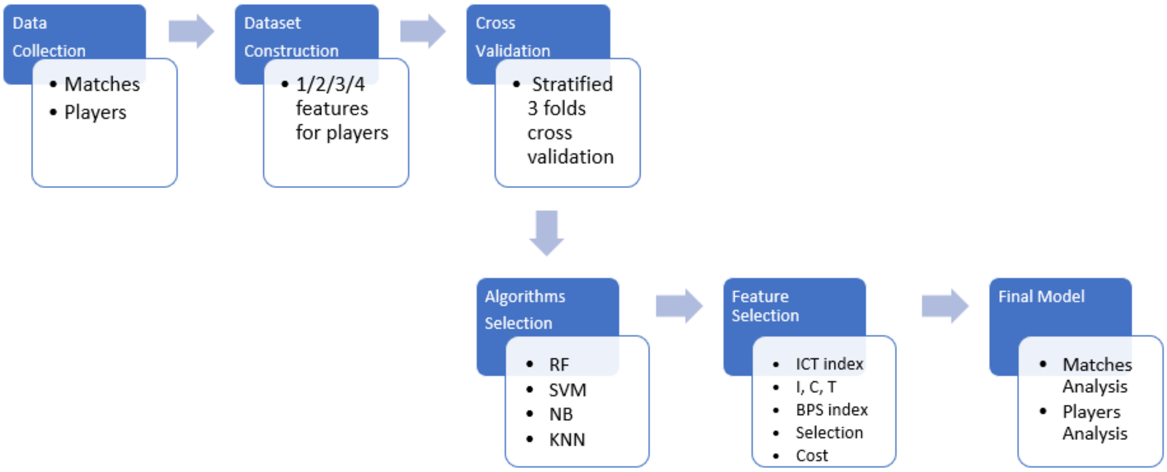
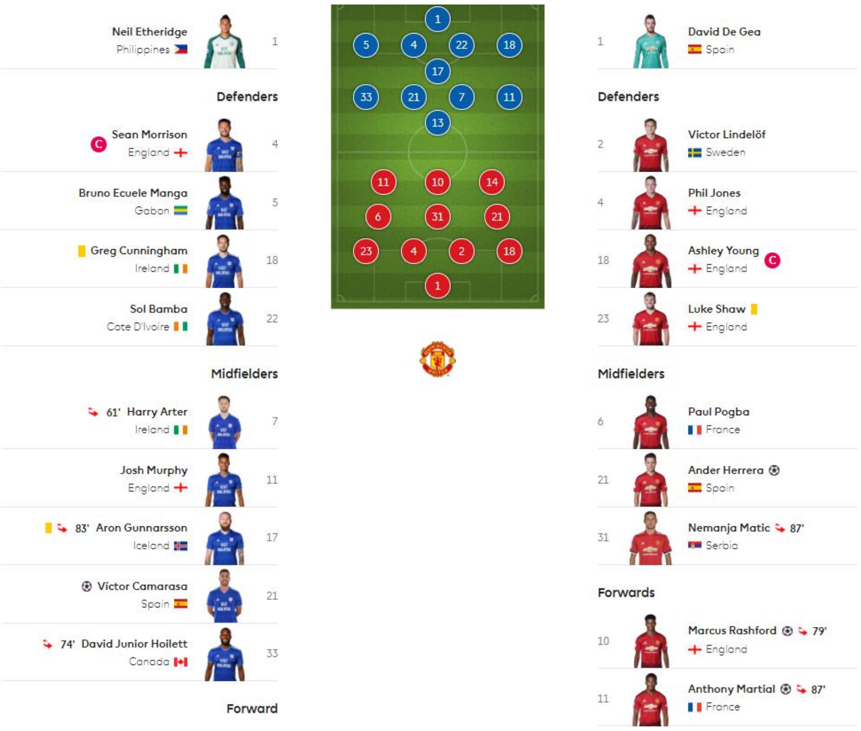


Fig 3.1 The workflow of entire project

3.2 Original Data Collection

The entire dataset consists of two sources which are the introduction of each match and the statistics of players in the starting line-up each match. The first source is a CSV format document for match results in the English Premier League which are collected from: https://www.premierleague.com/results. This is the official website of the English Premier League which provides the latest and specific statistics for each match in each week. There are 20 teams in the English Premier League and 10 matches for each week. As a result of that, this project decided to import the last 12 weeks' matches (totalling 120 matches) of the English Premier League in the 2018/2019 season when this dissertation start to write on October,2018. The entire data will be imported in the further experiments until this season is completed. The other reason why this project only uses 120 matches for the current season is that the soccer market keeps a high dynamic situation. A team cannot ensure that one player will still be playing for the same football club after the transfer period in the summer or winter and so the tactical arrangements for starting line-up might be quite distinctive under different coaches and seasons. This project extracts 120 matches with their match weeks, names of home teams and away teams, match results, goals of home teams and away teams. Besides, the starting line-up list of the match is also provided in the sub-link of this link, click on the result of each match to acquire the specific information which the following **Fig 3.2** demonstrates.



Neil Etheridge 1
Philippines

Defenders

Sean Morrison 4
England

Bruno Ecuele Manga 5
Gabon

Greg Cunningham 18
Ireland

Sol Bamba 22
Cote D'Ivoire

Midfielders

61' Harry Arter 7
Ireland

Josh Murphy 11
England

83' Aron Gunnarsson 17
Iceland

Víctor Camarasa 21
Spain

74' David Junior Hoilett 33
Canada

Forward

1 David De Gea
Spain

Defenders

2 Victor Lindelöf
Sweden

4 Phil Jones
England

18 Ashley Young
England

23 Luke Shaw
England

Midfielders

6 Paul Pogba
France

21 Ander Herrera
Spain

31 Nemanja Matic 87'
Serbia

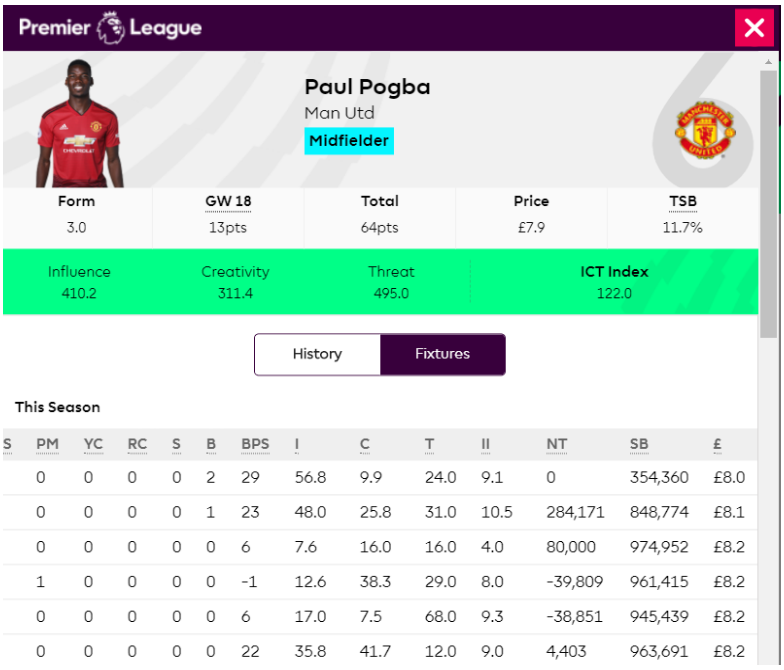
Forwards

10 Marcus Rashford 79'
England

11 Anthony Martial 87'
France

Fig 3.2 An example of Starting Line-up in English Premier League

The second source is the statistics of players in the starting line-up which is typed manually into an XLSX format document which is built based on https://fantasy.premierleague.com/, the following **Fig 3.3** is an example of the data source of a player. The statistical table involves the player's name, his position and football club and relevant features in each match of this season.



Paul Pogba
Man Utd
Midfielder

Form	GW 18	Total	Price	TSB
3.0	13pts	64pts	£7.9	11.7%

Influence	Creativity	Threat	ICT Index
410.2	311.4	495.0	122.0

This Season

S	PM	YC	RC	S	B	BPS	I	C	T	II	NT	SB	£
	0	0	0	0	2	29	56.8	9.9	24.0	9.1	0	354,360	£8.0
	0	0	0	0	1	23	48.0	25.8	31.0	10.5	284,171	848,774	£8.1
	0	0	0	0	0	6	7.6	16.0	16.0	4.0	80,000	974,952	£8.2
	1	0	0	0	0	-1	12.6	38.3	29.0	8.0	-39,809	961,415	£8.2
	0	0	0	0	0	6	17.0	7.5	68.0	9.3	-38,851	945,439	£8.2
	0	0	0	0	0	22	35.8	41.7	12.0	9.0	4,403	963,691	£8.2

Fig 3.3 An example of a player's statistics in the English Premier League

There are 20 features for each player in each match as shown in **Table 3.1**:

Name	Interpretation	Name	Interpretation	Name	Interpretation
MP	Minutes played	PM	Penalties missed	C	Creativity
GS	Goals scored	YC	Yellow cards	T	Threat
A	Assists	RC	Red cards	II	ICT Index
CS	Clean sheets	S	Saves	NT	Net Transfers
GC	Goals conceded	B	Bonus	SB	Selected by
OG	Own goals	BPS	Bonus Points System	£(Cost)	0.1million pounds as unit
PS	Penalties saved	I	Influence		

Table 3.1 All features in official website

All of them are provided by EA sport for the English Premier League. Unlike collecting statistics from a professional data company (McHale & Relton ,2018), the authoritative open source website seldom provides statistics such as the number or success rate of passing, tacking, shooting and cross, etc. for each player in each match. It only updates the overall data cumulative from the first match to the latest match, this is a developmental issue in further experiments.

3.3 Data Representation

All the series of datasets used in this project are built manually, according to different purposes in the following experiments. There will be various features selected for players. This section is going to describe the data representation in each possible line-up for players.

From the first portion of the data source, this project decided to use the **Week**, **Matches**, **Home Team** and **Away Team**, and **HomeResult:**

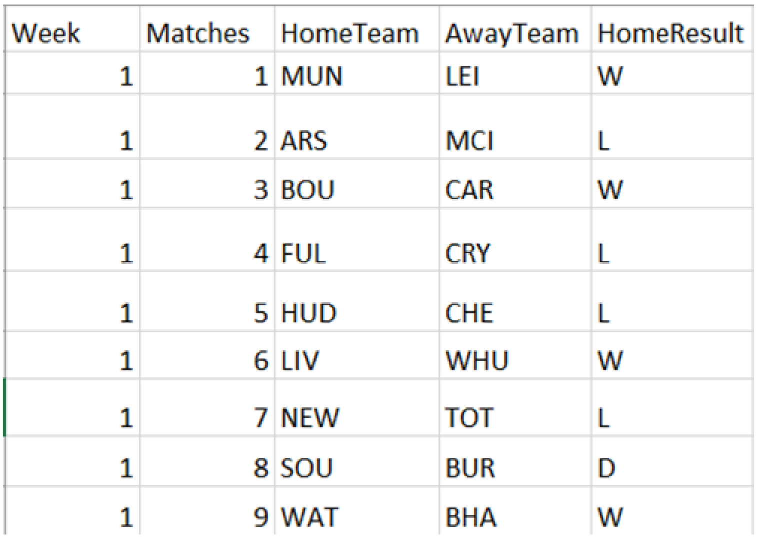
Week represents the number of match week. There are 12 weeks' data used in this project.

Matches represent the number of matches in each week, which is in the range of 1 to 10 in each week (10 matches each week in English Premier League).

Home Team and **Away Team** represent the name of both Home and Away team using their abbreviation.

HomeResult represents the final result of the full-time matches which is the dependent variable in this project, and this variable is from the perspective of the home team. There are three classes of results, namely W (means Win), L (means Loss) and D (means Draw).

Fig 3.4 is an example where according to these features as the reference, the data from the second source will be more clearly and quickly checked and inputted in this dataset.



Week	Matches	HomeTeam	AwayTeam	HomeResult
1	1	MUN	LEI	W
1	2	ARS	MCI	L
1	3	BOU	CAR	W
1	4	FUL	CRY	L
1	5	HUD	CHE	L
1	6	LIV	WHU	W
1	7	NEW	TOT	L
1	8	SOU	BUR	D
1	9	WAT	BHA	W

Fig 3.4 Features selected from the first source

There are 22 players and four kinds of positions in the starting line-up for both the home and away team in a soccer game. The following seven possible line-ups are arranged during these 120 matches:

(1) **1-3-4-3** represents 1 Goalkeeper, 3 Defenders, 4 Midfielders and 3 Forwards

(2) **1-3-5-2** represents 1 Goalkeeper, 3 Defenders, 5 Midfielders and 2 Forwards

(3) **1-4-3-3** represents 1 Goalkeeper, 4 Defenders, 3 Midfielders and 3 Forwards

(4) **1-4-4-2** represents 1 Goalkeeper, 4 Defenders, 4 Midfielders and 2 Forwards

(5) **1-4-5-1** represents 1 Goalkeeper, 4 Defenders, 4 Midfielders and 2 Forwards

(6) **1-5-3-2** represents 1 Goalkeeper, 5 Defenders, 3 Midfielders and 2 Forwards

(7) **1-5-4-1** represents 1 Goalkeeper, 5 Defenders, 4 Midfielders and 1 Forwards

In order to satisfy all of the possible tactical line-ups selected in matches, the dataset provides 28 places for players that assigns maximum places for each position in a team which means one place for Goalkeeper, five places for Defenders, five places for Midfielders and three places for Forwards. If the starting line-up does not have a full number of players in one position, the empty place will be substituted by using number zero, for instance, the team assigns four defenders in the starting line-up, so the fifth place for the defenders is given as zero. **Fig 3.5** shows the basic features and general structure of the dataset. In the next series of experiments, a different value will be filled in under these columns with a specific postfix, for instance, if the experiment uses ICT index for each player, the columns' name will be HG_ICT, HD1_ICT, HD2_ICT and so on.

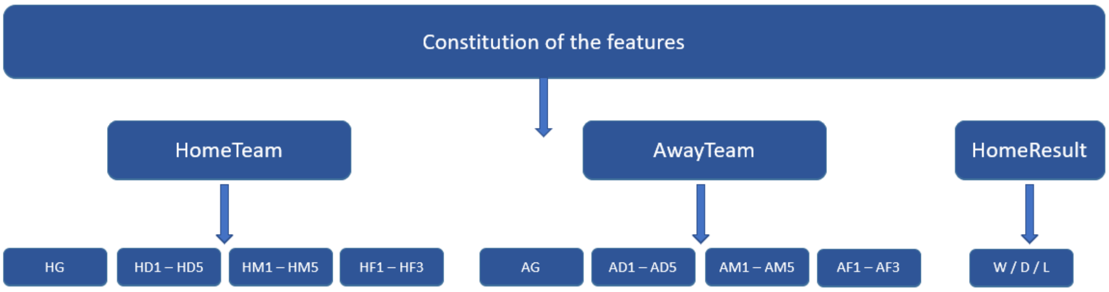


Fig 3.5 Basic constitution of the features in the dataset

The following details are the interpretation for each abbreviation feature in **Fig 3.5**:

HG represents the Home team goalkeeper

HD1 to **HD5** represent the No.1 Home team defender to the No.5 Home team defender

HM1 to **HM5** represent the No.1 Home team midfielder to the No.5 Home team midfielder

HF1 to **HF3** represent the No.1 Home team forward to the No.3 Home team forward

AG represents the Away team goalkeeper

AD1 to **AD5** represent the No.1 Away team defender to the No.5 Away team defender

AM1 to **AM5** represent the No.1 Away team midfielder to the No.5 Away team midfielder

AF1 to **AF3** represent the No.1 Away team forward to the No.3 Away team forward

Since columns of this data set in this project are named as the player's position instead of the player's name, the data in the same position of each match does not necessarily come from the same player, and it does not make sense to enter the players’ data randomly. To solve this problem, this project will use the ICT index to rank the players for each position. For instance, HD1 represents the player has the highest ICT index as a defender in the Home Team and HD5 represents the lowest ICT index as a defender in the Home Team.

The column names appearing in the following paragraphs and figures are created and explained based on this standard. For instance, HD1_BPS represents the Home Team defender’s BPS index whose ICT index is the highest among the five Home Team defenders. AM3_Selection represents the Away Team midfielder’s Selection value whose ICT index is the third highest in the five Away Team midfielders, and so on.

After building the general structure of the dataset combined with features in the first portion and the second portion, the next step is to select features for players from the second portion. Soccer is different from rugby, American football or basketball, etc. As it does not have a high-score round and the total goals are usually under 10 during one match, therefore, goals scored, assists, yellow or red cards might not be suitable to be the variables in this project, because their standard deviation will tend to be zero like a list of constants. In this case, BPS (Bonus Points System), I (Influence), C (Creativity), T (Threat), II (ICT Index), SB (Selected by), £ (Value, 0.1 million pounds as unit) are selected from the original variables scope to be used in the following experiments on account of their diverse continuous value and less missing value. Apart from that, the BPS, I, C, T, and ICT Index are acquired from a perspective of official statistics which can help coaches to arrange their tactics. SB influences the selection from the perspective of fans and online soccer players, namely the players who are the most popular players in the team. Value reflects the tendency of the soccer market which can provide suggestions for the manager on how to keep running the soccer club.

The ICT index is a soccer statistical index to offer insight to help soccer fans and online soccer players. Even professional staff make selections and formulate strategies for further matches. This index combines the values of Influence, Creativity and Threat values and calculate the individual weight for these three factors and then generates one specific value to be the ICT index from the Fantasy Premier league. The first component of ICT index is Influence, which is a measurement to evaluate the impact degree made by one player in a single match or even throughout the entire season. This reflects who the core player is to lead and unite the team, bring a positive influence no matter where his position may be or whether he scores a goal or assists his teammates during the match. The second component is Creativity, which is an index to assess a player who produces goal-scoring opportunities for others. It has a similar function to the assists index but considers more on frequency and the quality of passing and crossing on pitch location. The third component is Threat which examines the possibility of a player scoring goals, or his 'threat'. The index is also generated based on a player's shooting action and location during the match.

	BPS		BPS
Playing 1 to 60 minutes	3	Scoring the goal that wins a match	3
Playing over 60 minutes	6	70 to 79% pass completion (at least 30 passes attempted)	2
Goalkeepers and defenders scoring a goal	12	80 to 89% pass completion (at least 30 passes attempted)	4
Midfielders scoring a goal	18	90%+ pass completion (at least 30 passes attempted)	6
Forwards scoring a goal	24	Conceding a penalty	-3
Assists	9	Missing a penalty	-6
Goalkeepers and defenders keeping a clean sheet	12	Yellow card	-3
Saving a penalty	15	Red card	-9
Save	2	Own goal	-6
Successful open play cross	1	Missing a big chance	-3
Creating a big chance (a chance where the receiving player should score)	3	Making an error which leads to a goal	-3
For every 2 clearances, blocks and interceptions (total)	1	Making an error which leads to an attempt at goal	-1
For every 3 recoveries	1	Being tackled	-1
Key pass	1	Conceding a foul	-1
Successful tackle (the total of successful tackles minus unsuccessful tackles)	2	Being caught offside	-1
Successful dribble	1	Shot off target	-1

Table 3.2 Interpretation of BPS index statistics

The BPS index is the abbreviation for the Bonus Points System which captures players' actions on the pitch, according to their specific performance to mark their scores based on the following grade standard:

Compared with the ICT index, the BPS index is more objective, because it is created from a series of concrete statistics from the match. However, Influence, Creativity, and Threat are more likely to assess one player from a general summary, which is more subjective to provide a reference probability.

Selection is the variable which illustrates the number of times a player is selected by online soccer players. The last variable applied in this dataset is Cost, which represents the weekly value of a player in the soccer market and it is given in units of 0.1 million pounds.

3.3.1 One feature for players

In the following feature selection, the models will use one feature to construct the dataset using the ICT index, Influence, Creativity and Threat, BPS index, Cost and Selection, respectively.

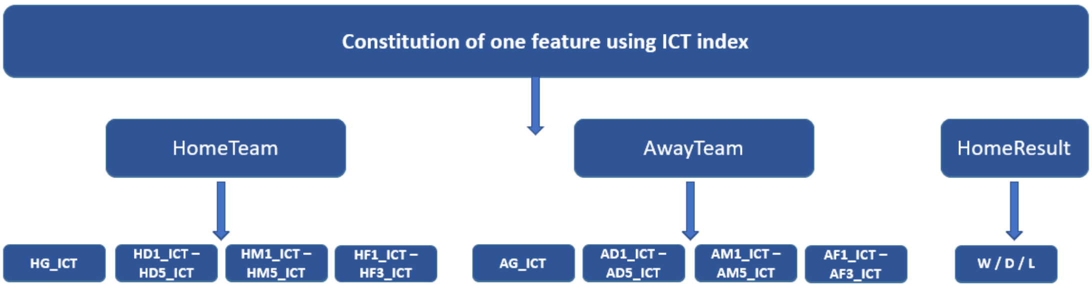


Fig 3.6 Constitution of one feature using ICT index

In the fourth chapter, the ICT index feature, as shown in **Fig 3.6** above, will be used to make the comparison between the algorithms in experiment No.1 and the baseline selection in experiment No.2. The data size is 29 columns and 120 rows (12 weeks' data):

28 independent variables for ICT index: from HG_ICT to AF3_ICT

1 dependent variable: HomeResult

Fig 3.7 on page 36 illustrates the constitution of using Influence, Creativity, and Threat, which features regard as " one feature" called ICTS because they are separate editions of the ICT index, and none of three features will be removed in the experiments so that they can be used to do the baseline selection with the ICT index in the second experiment in the fourth chapter, and then the winner is the new baseline to carry out feature selection in experiment No.3. In this dataset, there are 85 columns and 120 rows (12 weeks' data):

84 independent variables:

28 columns for I (Influence): from HG_I to AF3_I

28 columns for C (Creativity): from HG_C to AF3_C

28 columns for T (Threat): from HG_T to AF3_T

1 dependent variable: HomeResult

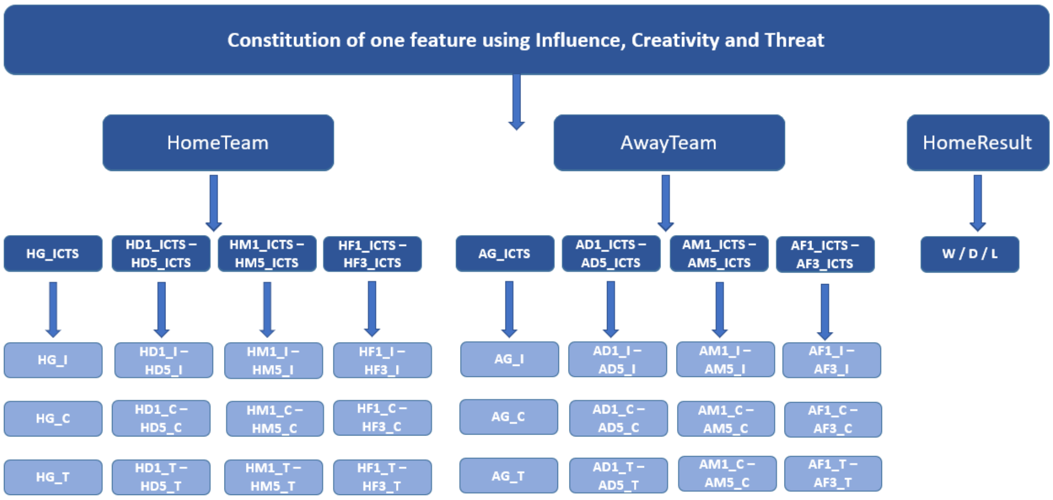


Fig 3.7 Constitution of one feature using Influence, Creativity and Threat

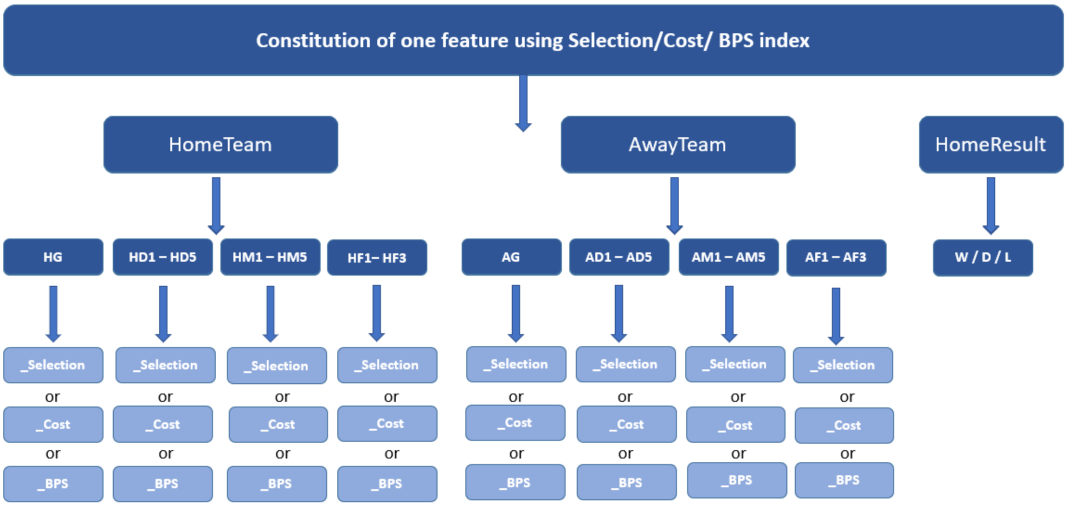


Fig 3.8 Constitution of one feature using Selection or Cost or BPS index

As **Fig 3.8** above illustrates, the structure of the dataset with a feature Selection or Cost or BPS index. In their dataset, all of them have 29 columns and 120 rows (12 weeks' data):

28 independent variables for Cost: from HG_Cost to AF3_Cost

Or

28 independent variables for Selection: from HG_Selection to AF3_Selection

Or

28 independent variables for BPS index: from HG_BPS to AF3_BPS

Common 1 dependent variable: HomeResult

3.3.2 Two features for players

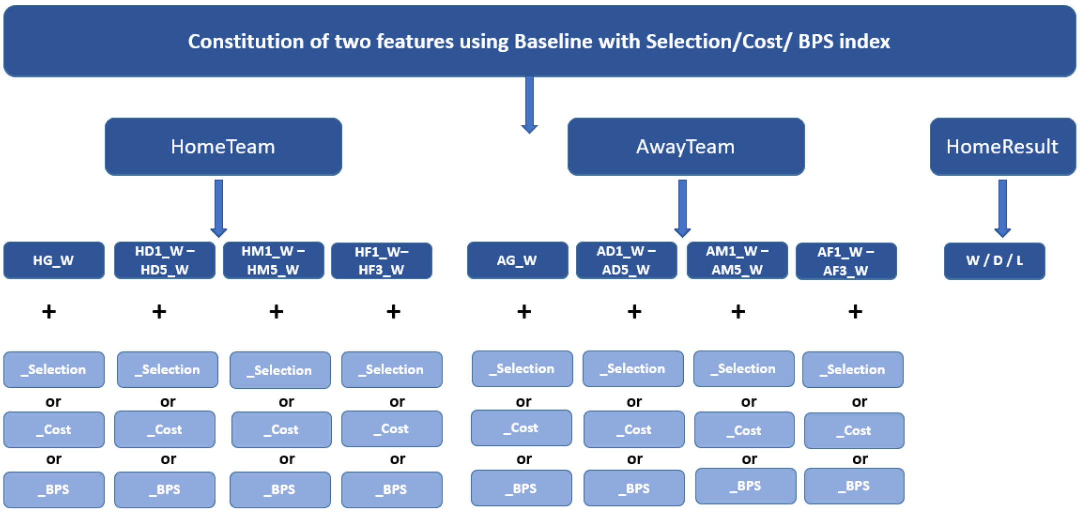


Fig 3.9 Constitution of two features using Baseline with Selection or Cost or BPS index

Fig 3.9 demonstrates the structure of the dataset with baseline feature and Selection or Cost or BPS index to do the feature selection in experiment No.2. The baseline feature is the higher prediction accuracy between the ICT index and Influence, Creativity and Threat which is also the meaning of suffix "**w**" in **Fig 3.9**. As a result, there will be three datasets during the feature selection. If the ICT index is the winner of the baseline feature comparison, the dataset will have 57 columns and 120 rows in total:

56 independent variables:

28 variables for ICT index: from HG_ICT to AF3_ICT

28 variables for Selection: from HG_Selection to AF3_Selection

Or

28 variables for Cost: from HG_Cost to AF3_Cost

Or

28 variables for BPS index: from HG_BPS to AF3_BPS

1 independent variable: HomeResult

If the Influence, Creativity, and Threat are the winners of the baseline feature comparison, the dataset will have 85 columns and 120 rows in total:

84 independent variables:

28 variables for I, 28 variables for C, 28 variables for T with 28 variables for Selection or Cost or BPS index

1 independent variable: HomeResult

3.3.3 Three features for players

The structure of datasets with three features for players is similar to the above datasets in **Section 3.2.2**, but the number of columns that there will be 85 columns and 120 rows. If the ICT index is the baseline feature in this project, 85 columns will include 28 columns for the ICT index, 28 columns for the highest prediction accuracy feature in **Section 3.2.2**(which is the result of feature selection in the first layer that will be interpreted in the **Section 3.5**), 28 columns for the remaining two features respectively, and 1 for independent variable – HomeResult. If Influence, Creativity, and Threat is the new baseline feature in this project, which means there will be 113 columns in the dataset.

3.3.4 Four features for players

The structure of datasets in **Section 3.2.4** is composed of baseline feature, Selection, Cost and BPS index. If the baseline feature is ICT index there will be 113 columns and 120 rows in total, otherwise, Influence, Creativity and Threat is the baseline feature that there will be 169 columns and 120 rows in total.

3.4 Data Segmentation

3.4.1 Cross-validation

Cross-validation is a statistical analysis method used to verify the performance of the classifier. The basic idea is to group the original data, one part as the training set and the other part as the verification set. The device is trained to use the verification set to test the trained model as a performance indicator for evaluating the classifier to divide the complete data set into a training set and a test set. The following points must be observed:

(1) Only the training set can be used in the training process of the model. The test set must be used to evaluate the merits of the model after the model is completed.

(2) The number of samples in the training set must be sufficient, generally at least 50% of the total number of samples.

(3) The two sets of subsets must be evenly sampled from the complete set.

K-folds cross-validation is one of the most common methods used in cross-validation. The original data is divided into K groups (generally equal), each subset data is separately verified, and the remaining K-1 subset data is used as a training set so that K models are obtained, and the K models are used finally. The average of the classification accuracy of the verification set is used as the performance indicator of the classifier under this K-CV. K-CV can effectively avoid over-learning and under-learning, and the results obtained are more persuasive than those from manual data partitioning.

In this case, to ensure each fold ’s data can be divided exactly by 120 matches and keep enough data in each subset. The model selects 3 folds cross-validation in this project, as **Fig 3.10 on page 40** presents that each partition will be the testing data to predict the matches result. The overall predicting outcomes will assemble 3 times results from these.

Apart from that, this project is going to develop a stratified random sampling base on the 3 folds cross-validation. As **Fig 3.11 on page 40** there are approximately 30 matches in Draw (25% of all), 40 matches in Loss (33.3% of all) and 50 matches in Win (41.7% of all). Stratified random sampling means each fold contains roughly the same ratio of the three types of HomeResult which can avoid the imbalance problem which might appear during data splitting steps, ensure each partition has a good performance in representing the entire data.

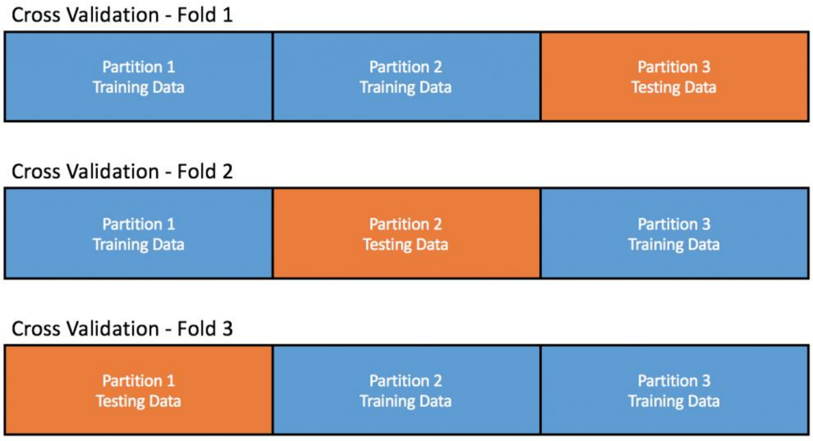

Cross Validation - Fold 1
Partition 1 Training Data
Partition 2 Training Data
Partition 3 Testing Data
Cross Validation - Fold 2
Partition 1 Training Data
Partition 2 Testing Data
Partition 3 Training Data
Cross Validation - Fold 3
Partition 1 Testing Data
Partition 2 Training Data
Partition 3 Training Data

Fig 3.10 the principle of 3 folds cross validation

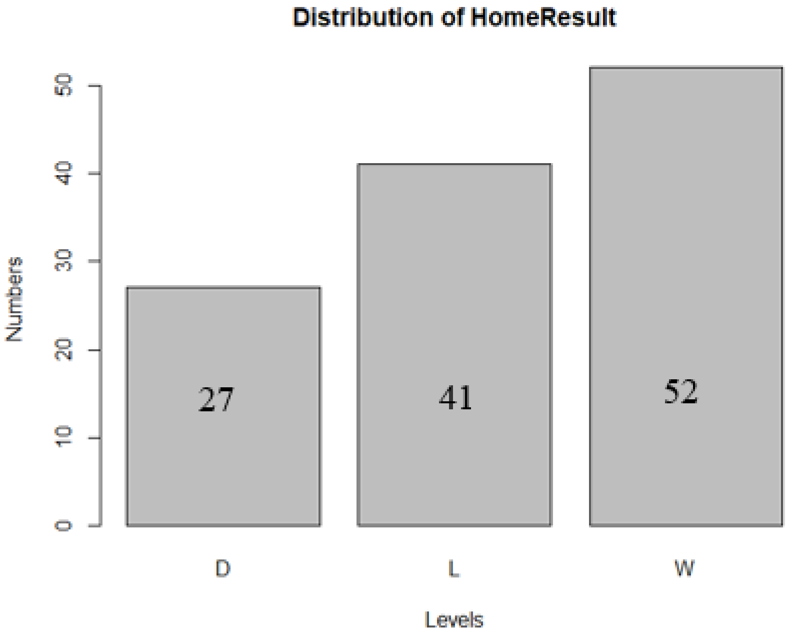


Fig 3.11 Distribution of target variable *HomeResult*

After the complete data are divided, as shown in **Fig 3.12,** the model will be executed twice to satisfy the 3 folds cross-validation in each of the following experiments. Firstly, it divides the entire dataset into three parts, and takes two of them as training sets, where the remaining one is the test set. Secondly, it continues to perform stratified 3 folds-cross validation in the training set, using these three parts to adjust the parameters of the training model.

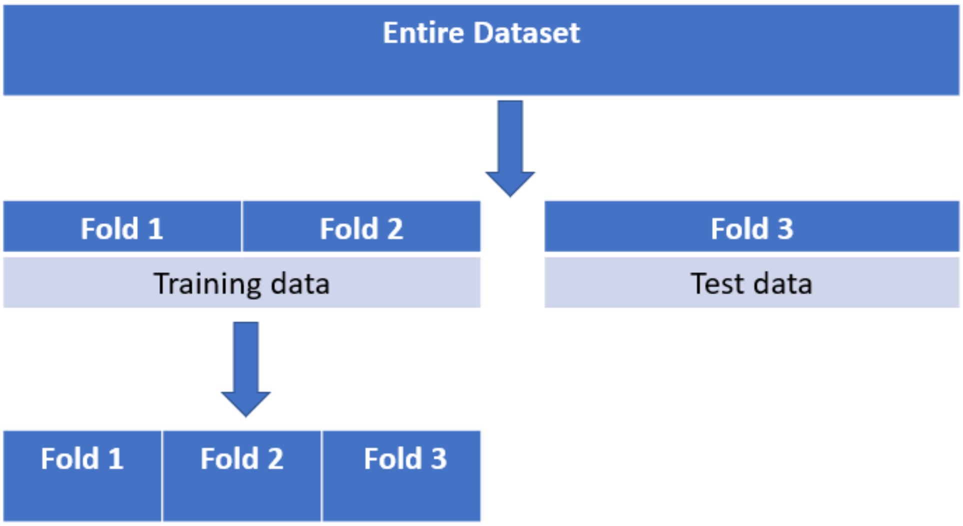


Fig 3.12 The processes of stratified 3 folds cross validation

3.5 Machine Learning Algorithms

This project will compare four algorithms which are Random Forest, Support Vector Machine, Naïve Bayes and K-Nearest Neighbour so as to choose the best classifier to create the final model. These algorithms are frequently used in classification models, as discussed in the literature review. **Section 4.1** describes how to build models with them and adjust their respective parameters. Afterwards, the most accurate classifiers will be used in order to conduct the feature selection.

It should be noted that the results of **Section 4.1** are used as the model classifier in **Section 4.5**.

3.6 Feature Selection

Section 3.3 has described the dataset which will be used in the feature selection, consequently, this section is going to describe comprehensively how the process of feature selection works.

The Wrapper method finds a subset of all feature subsets, which enables subsequent learning algorithms to achieve a higher performance. In the feature selection phase, the wrapper can be viewed as a search method plus learning algorithm. As is demonstrated in **Fig 3.13 on Page 42**, the model compares prediction accuracy by using ICT index and its separate version in order to select the starting point of feature selection in the first step. In the second step, the model adds each feature (which are Selection, Cost and BPS index) based on the feature which gets a higher accuracy in the previous step to make the prediction again so as to select the highest accuracy group to be the next baseline features in the first layer. It is necessary to repeat the second step until the prediction accuracy does not increase or stay constant.

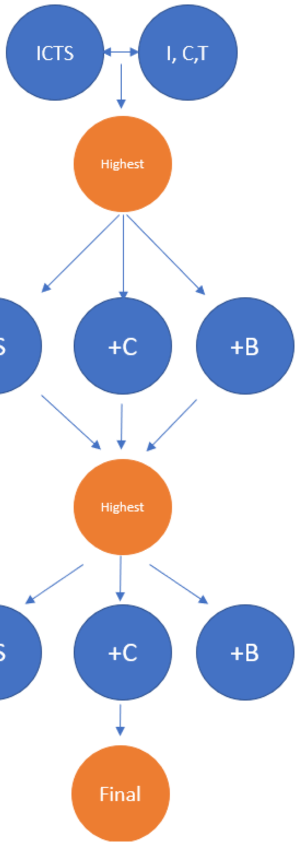


Fig 3.13 The diagram of feature selection

3.7 Evaluation

This section will focus on the confusion matrix which is one of the most effective and commonly used evaluation methods to estimate the predictive result. The following **Section 3.7.1** interprets the confusion matrix and points out the relevant indicators used in this research.

3.7.1 Confusion Matrix

In Machine Learning, especially in statistical classification, the confusion matrix is also called the ‘error matrix’. The confusion matrix compares the prediction results with the real results in a matrix of binary classifications as shown in **Table 3.3** below.

Predicted Values	Actual Values		
		Positive (1)	Negative (0)
	Positive (1)	True Positive	False Positive
	Negative (0)	False Negative	True Negative

Table 3.3 Interpretation of Confusion Matrix

True Positive (TP): The real category of the sample is a positive example, and the results predicted by the model are also positive examples.

True Negative (TN): The real category of the sample is a negative example, and the model makes its prediction as a negative example as well.

False Positive (FP): The real category of the sample is a negative example, but the model makes its prediction a positive example.

False Negative (FN): The true category of the sample is a positive example, but the model makes its prediction as a negative example.

There are several indexes which must always be calculated as the evaluation criteria in the confusion matrix which help users to understand and analyse the results.

Accuracy = (TP+TN)/TP+FP+TN+FN

Precision = accuracy = precision = TP / (TP + FP): The proportion of samples that truly positive in the sample predicted by the model is positive

Recovery rate = recall rate = recall = TP/(TP+FN): the ratio of samples correctly predicted by the model is positive in the total number of positive samples.

Generally speaking, **Accuracy** and **Recall** values are used in the next series of models in Chapter 4. **Accuracy** represents the overall accuracy of a model, which is the most direct criterion for algorithms comparison and feature selection to select the best performing model. **Recall** is used to compare the prediction of each class in the target variable to find out how many cases in the actual situation are correctly predicted.

There is also a reference indicator called the ‘kappa coefficient’, in extreme situations, different models may have the same prediction accuracy. Because of this, the researcher can further refer to this indicator to make the final decision. The kappa coefficient is a method used to evaluate consistency in statistics. It can be used to evaluate the accuracy of a multi-class model. The value of this coefficient is [-1, 1]. But usually kappa falls between 0 and 1 and can be divided into five groups to indicate the consistency of different levels: 0.0-0.20 very low consistency, 0.21-0.40 general consistency, 0.41-0.60 medium consistency, and 0.61 -0.80 being substantial consistency. It is almost perfect at 0.81-1.0.

Chapter 3 has already described the methodologies and criteria for data collection to the evaluation of model results in this research. The fourth chapter will follow and implement the design outline of this chapter, obtain predictions and discuss and analyse them.

4. RESULTS, EVALUATION AND DISCUSSION

In this chapter, the following paragraphs are going to describe particular processes of modelling and predicting results based on theories of four Machine Learning algorithms and classifiers in **RStudio** which is a professional platform for the R statistical computing environment, as described in the previous chapter, and then use feature selection to select further, useful features in order to improve the accuracy of the prediction.

4.1 Experiment 1 is used to do the algorithms comparison so as to select the most suitable classifiers for the following experiments which can be applied as the unique classifier to run the feature selection.

4.2 Experiment 2 is to select the most appropriate data set size, compare the data of one team (which has a total of 240 rows) and the data of two teams (which has a total of 120 rows).

4.3 Experiment 3 is to perform feature selection, and select which features of the players can get the highest prediction accuracy. By determining the starting feature in Experiment 4, the next step is to develop three rounds of feature selection through the wrapper approach.

4.3.1 Experiment 4 is the first work of Experiment 3, which selects higher predicted values from the ICT index and its separate versions – Influence, Creativity, and Threat as the baseline feature in feature selection.

4.1 Experiment 1 - Algorithms Comparison

Experiment 1 is the algorithms comparison which aims to select the best classifier from these four: Random Forest, Support Vector Machine, Naïve Bayes and K-Nearest Neighbour. In order to unify the modelling standards, all four of these classifiers are going to use 29 columns and 120 rows of data (12 weeks data) simultaneously with the ICT index which was discussed in **Section 3.3.1**. In addition, according to the class of the target variable (***HomeResult***), there are 27 rows for Draw which occupied 22.5% of all, 41 rows for Loss which occupied approximately 34.17% of all and finally 52 rows for Win which occupied approximately 43.33% of all. As a result, a stratified 3 folds cross-validation is used to divide training sets and test sets as follows.

Category sets	Folds	D **(22.5%)**	L **(34.17%)**	W **(43.33%)**
Training 1	Fold 1 & 2	18	27	35
Test 2	Fold 3	9	14	17
Training 2	Fold 1 & 3	18	27	35
Test 2	Fold 2	9	14	17
Training 3	Fold 2 & 3	18	28	34
Test 3	Fold 1	9	13	18

Table 4.1 Data distribution and Data segmentation

Table 4.1 shows that fold 1 has 9 for Draw, 14 for Loss and 17 for Win, fold 2 also has 9 for Draw, 14 for Loss and 17 for Win, and fold 3 has 9 for Draw, 13 for Loss and 18 for Win.

The following **Sections, 4.1.1** to **Section 4.1.4,** will describe the process and results of applying the four algorithms separately. The results are summarized in **Section 4.1.5** and discussed and analysed to select the most appropriate classifier.

4.1.1 Random Forest

The details of modelling, fixing of parameters and determining the results from predictions in Random Forest algorithm will be illustrated in **Section 4.1.1**.

Random forest model building and parameter modification are performed in the *randomForest* package which is specifically designed to perform regression or classification through random forest algorithms. As described above, stratified 3 folds cross-validation is used to divide the data set. The first step is to build three training models with default parameters as the baseline model. The second step is to adjust the two main and significant parameters of mtry and ntree of the three training sets, mtry represents the number of variables selected for random sampling in the split node for the binary classification tree and ntree is the number of trees to generate in the random forest. Choosing the right mtry and ntree can stabilize the error rate of the training model and avoid problems with unnecessarily inefficiency and over-fitting. The final parameter selection is determined to predict the test sets by comparing the prediction accuracy of the default parameters with the prediction accuracy of the adjusted parameters in the training sets. The next stage is to combine the prediction results of the three test sets to obtain the overall prediction accuracy of the random forest.

The following **Table 4.2** illustrates the prediction accuracy of the three test sets divided by stratified cross-validation and the final parameters of the mtry and ntree they used.

Test set	mtry	ntree	Prediction accuracy
Fold 1	6	500	62.5%
Fold 2	5	500	57.5%
Fold 3	2	500	70%

Table 4.2 Prediction accuracy and parameters of each test set in RF

In summary, by adjusting the most suitable parameters of the three models based on the Random Forest algorithm, the optimal forest prediction results are obtained. The following table gathered 120 test data and demonstrated the process of gathering statistics, and the results are shown in **Table 4.3.** The overall prediction accuracy uses correct predictive value of Class D, L and W divided by the overall number of prediction data, and the Recall can reflect the ratio of correct predictions in the actual situation in each class:

RF overall prediction accuracy= (9+25+42)/120 =76/120≈63.3%

The Recall of class D=9/ (9+4+14) =9/27≈33.33%

The Recall of class L=25/ (5+25+11) =25/41≈60.98%

The Recall of class W=42/ (4+6+42) =42/52≈80.77%

	Reference		
Prediction	D	L	W
D	9	5	4
L	4	25	6
W	14	11	42

Table 4.3 Overall confusion matrix for RF with ICT index

4.1.2 Support Vector Machine

This section is a description of the Support Vector Machine modelling process and predictions. The general implementation is similar to Random Forest in using default parameters to compare with adjusted parameters in training sets, and select the suitable results to predict the test sets. SVM uses the *e1071* package in the process of modelling and adjusting parameters. In the *e1071* package, there is a *svm* function that can carry out multiple classifications with more than two classes with the ***class. type***= 'one against one' or 'one against all' approach. Both approaches are to divide the data into binary models for comparison and get the appropriate class by the voting scheme (review **Section 2.1.2**) and, further, by using each ***kernel*** and ***type*** in turn to find the combination with the least amount of errors as the final option for these two parameters. In the *e1071* package, the *tune.svm* function can help to optimize ***cost*** and ***gamma*** which effects have been interpreted in **Section 2.1.2** that ***cost*** determines the generalization ability of the model, and ***gamma*** will affect the number of support vectors to affect the speed of training and prediction. The following **Table 4.4** demonstrates the final parameters that should be used to predict the test sets.

Test set	Cost	Gamma	kernel	type	class. type	Prediction accuracy
Fold 1	4	1	nu-classification	polynomial	'one against one'	60.0%
Fold 2	4	0.5	nu-classification	polynomial	'one against one'	52.5%
Fold 3	8	0.5	nu-classification	radial	'one against one'	62.5%

Table 4.4 Prediction accuracy and parameters of each test set in SVM

One thing that needs to be emphasized in the table is the selection of class.type, although for 'one against one' its disadvantage is that the number of binary classifiers required for construction and testing is usually much more than 'one against all' and the total training times and test times are relatively slow. In this data set, there are only three classes for the target variable, while the number of models that need to be built in both methods is the same. But 'one against one' can keep the training set more balanced than the other and that is the reason why this experiment selected it.

In selecting the most appropriate training models to predict the result the following **Table 4.5** details the classes distribution of the target variable.

SVM overall prediction accuracy = (10+19+41)/120=70/120≈58.33%

The Recall of class D=10/ (6+3+18) =10/27≈37.04%

The Recall of class W=41/ (6+5+41) =41/52≈78.85%

The Recall of class L=19/ (6+19+16) =19/41≈46.34%

	Reference		
Prediction	D	L	W
D	10	10	6
L	6	19	5
W	11	12	41

Table 4.5 Overall confusion matrix for SVM with ICT index

4.1.3 Naïve Bayes

In this section, the project is going to build 3 training models in Naïve Bayes algorithm in order to continue the algorithms comparison with the other three algorithms. This experiment keeps the same processes as **Section 4.1.1** and **Section 4.1.2** to find out the most appropriate parameters for prediction.

In **Table 4.6** there are three kinds of parameters that the Naïve Bayes models should be tuned in the train function with method= 'nb' in ***caret*** package. The first one is ***usekernel*** which allows the user to apply both a kernel density estimate and a Gaussian density estimate for continuous variables. The second one is ***adjust*** which allows user to adjust the bandwidth of the kernel density, as larger numbers represent a more flexible density estimate in the model. The last one is ***fL*** which can help the user to modify the Laplace smoother.

Test sets	fL	usekernel	adjust	Prediction accuracy
Fold1	0	True	2	56.10%
Fold2	0	True	5	61.54%
Fold3	1	True	3	60.00%

Table 4.6 Prediction accuracy and parameters of each test set in NB

In summary, by adjusting a series of appropriate parameters of the three models based on Naïve Bayes algorithm, the following table shows the 120-test data collected and demonstrates the process of gathering statistics, as the result of in **Table 4.7**, the overall prediction accuracy uses the correct predictive value of Class D, L and W divided by the overall number of prediction data:

NB overall prediction accuracy = (8+26+37)/120 =71/120≈59.16%

The Recall of class D=8/ (8+9+10) =8/27≈29.63%

The Recall of class L=26/ (5+26+10) =26/41≈63.41%

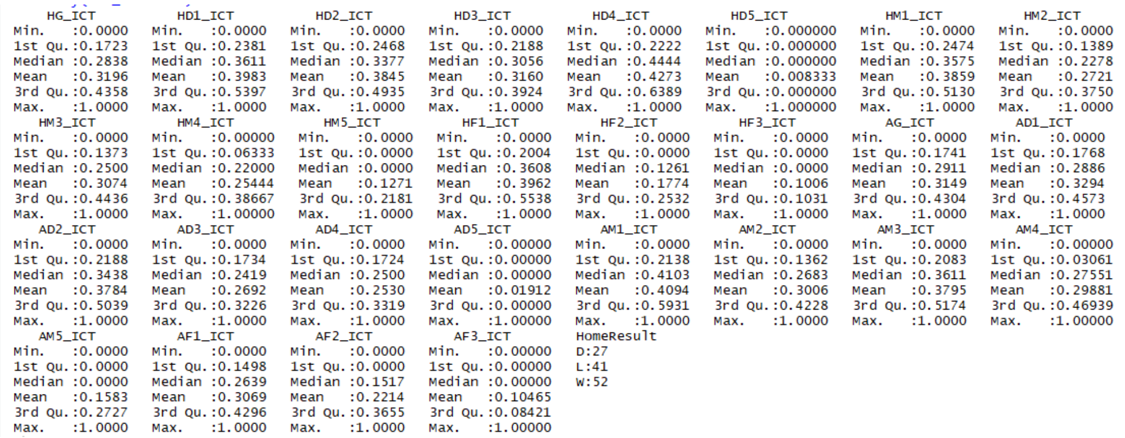
The Recall of class W=37/ (10+5+37) =36/52≈69.23%

	Reference		
Prediction	D	L	W
D	8	5	10
L	9	26	5
W	10	10	37

Table 4.7 Overall confusion matrix for NB with ICT index

4.1.4 K-Nearest Neighbour

The following series of models with K-Nearest Neighbour aims to make algorithms comparison with the other three algorithms. KNN still uses the same data set as above three algorithms, however, because of the requirement of K-Nearest Neighbour, the data should be normalized which would be used in the following models that transferred them between 0 and 1 as the **Fig 4.1** above shows.



```
    HG_ICT            HD1_ICT            HD2_ICT            HD3_ICT            HD4_ICT            HD5_ICT              HM1_ICT            HM2_ICT
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.000000   Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.1723   1st Qu.:0.2381   1st Qu.:0.2468   1st Qu.:0.2188   1st Qu.:0.2222   1st Qu.:0.000000   1st Qu.:0.2474   1st Qu.:0.1389
 Median :0.2838   Median :0.3611   Median :0.3377   Median :0.3056   Median :0.4444   Median :0.000000   Median :0.3575   Median :0.2278
 Mean   :0.3196   Mean   :0.3983   Mean   :0.3845   Mean   :0.3160   Mean   :0.4273   Mean   :0.008333   Mean   :0.3859   Mean   :0.2721
 3rd Qu.:0.4358   3rd Qu.:0.5397   3rd Qu.:0.4935   3rd Qu.:0.3924   3rd Qu.:0.6389   3rd Qu.:0.000000   3rd Qu.:0.5130   3rd Qu.:0.3750
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.000000   Max.   :1.0000   Max.   :1.0000
    HM3_ICT            HM4_ICT             HM5_ICT            HF1_ICT            HF2_ICT            HF3_ICT            AG_ICT             AD1_ICT
 Min.   :0.0000   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.1373   1st Qu.:0.06333   1st Qu.:0.0000   1st Qu.:0.2004   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.1741   1st Qu.:0.1768
 Median :0.2500   Median :0.22000   Median :0.0000   Median :0.3608   Median :0.1261   Median :0.0000   Median :0.2911   Median :0.2886
 Mean   :0.3074   Mean   :0.25444   Mean   :0.1271   Mean   :0.3962   Mean   :0.1774   Mean   :0.1006   Mean   :0.3149   Mean   :0.3294
 3rd Qu.:0.4436   3rd Qu.:0.38667   3rd Qu.:0.2181   3rd Qu.:0.5538   3rd Qu.:0.2532   3rd Qu.:0.1031   3rd Qu.:0.4304   3rd Qu.:0.4573
 Max.   :1.0000   Max.   :1.00000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
    AD2_ICT            AD3_ICT            AD4_ICT            AD5_ICT             AM1_ICT            AM2_ICT            AM3_ICT            AM4_ICT
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
 1st Qu.:0.2188   1st Qu.:0.1734   1st Qu.:0.1724   1st Qu.:0.00000   1st Qu.:0.2138   1st Qu.:0.1362   1st Qu.:0.2083   1st Qu.:0.03061
 Median :0.3438   Median :0.2419   Median :0.2500   Median :0.00000   Median :0.4103   Median :0.2683   Median :0.3611   Median :0.27551
 Mean   :0.3784   Mean   :0.2692   Mean   :0.2530   Mean   :0.01912   Mean   :0.4094   Mean   :0.3006   Mean   :0.3795   Mean   :0.29881
 3rd Qu.:0.5039   3rd Qu.:0.3226   3rd Qu.:0.3319   3rd Qu.:0.00000   3rd Qu.:0.5931   3rd Qu.:0.4228   3rd Qu.:0.5174   3rd Qu.:0.46939
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
    AM5_ICT            AF1_ICT            AF2_ICT            AF3_ICT          HomeResult
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   D:27
 1st Qu.:0.0000   1st Qu.:0.1498   1st Qu.:0.0000   1st Qu.:0.00000   L:41
 Median :0.0000   Median :0.2639   Median :0.1517   Median :0.00000   W:52
 Mean   :0.1583   Mean   :0.3069   Mean   :0.2214   Mean   :0.10465
 3rd Qu.:0.2727   3rd Qu.:0.4296   3rd Qu.:0.3655   3rd Qu.:0.08421
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
```

Fig 4.1 Summary of normalized data in KNN

The most important parameter in the KNN model that needs to be adjusted in this experiment is the K value. The KNN finds the k training samples whose training set is closest to the test sample and then predicts the species based on the information of the K training samples. ***Class*** package has a ***knn*** function which is used to carry out the KNN algorithm and select the suitable k value for three training models to acquire the highest prediction accuracy on three test sets. The K value selection shown in **Table 4.8** below is the best result after a stratified 3 folds cross-validation in the training model is conducted.

Test set	K	Prediction accuracy
Fold 1	6	67.5%
Fold 2	13	52.5%
Fold 3	9	52.5%

Table 4.8 Prediction accuracy and parameters of each test set in NB

In summary, combined with the results of three experiments by using K-Nearest Neighbour, **Table 4.9** illustrates a general outline of the overall performance and the recall of each class:

KNN overall prediction accuracy = (9+22+38)/ 120 =69/120=57.50%

The Recall of class D=9/ (9+10+8) =9/27≈33.33%

The Recall of class L=22/ (9+22+10) =22/41≈53.66%

The Recall of class W=38/ (9+5+38) =38/52≈73.08%

	Reference		
Prediction	D	L	W
D	9	9	9
L	10	22	5
W	8	10	38

Table 4.9 Overall confusion matrix for KNN with ICT index

4.1.5 Algorithms Conclusion

In summary, with the overall highest average prediction accuracy and each class prediction accuracy, as illustrated **Table 4.10 below**, **Random Forest (63.30%)** won the first place with a slight advantage in these four algorithms, which means that **Random Forest** will be selected to conduct feature selection in the following experiments.

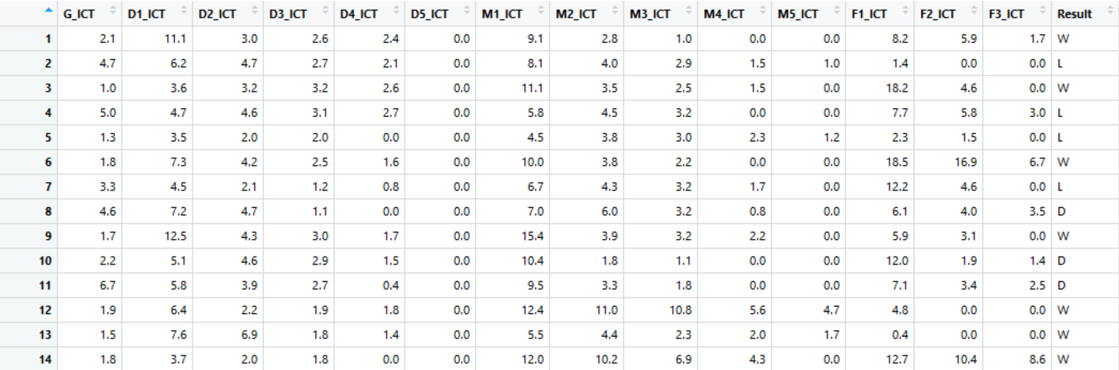
Compared with the prediction accuracy of each class, RF (80.77% in class W) beats RF (80.77% in class W) in class W, in the remaining classes, NB (63.41% in class L) has the highest accuracy in class L and SVM (37.04% in class D) gets the best result in class W. Although the SVM has the best performance in predicting class D that is the most challenging predictions in the match, it has the lowest prediction accuracy in class L at the same time. This is because the mechanism of the SVM itself is a binary selection. For the target variable of the multi-level classification in this data, it is necessary to compare each class one by one to obtain the result, so it probably generates more error predictions than others in this stage of the process. The KNN algorithm is the last one in this comparison. Since the KNN algorithm does not require the data distribution, its prediction accuracy in each class is at an intermediate level. No one class has the best prediction result, and also no class has the worst prediction effect in comparison. As a typical representative of "lazy learning", it has no visible training process, but it is predicted by majority voting (this is similar to multi-level classified SVM), so the ability to discover relationships between features is limited. On the other hand, Random Forest deals with the problem of unbalanced sample classification, which can balance errors and is more robust dealing with errors and outliers, therefore, it may have a better performance in the draw or the shock match in the following predictions.

Classifiers	Overall prediction accuracy	Prediction accuracy of Class W	Prediction accuracy of Class L	Prediction accuracy of Class D
RF	63.30%	80.77%	60.98%	33.33%
SVM	58.33%	78.85%	46.34%	37.04%
NB	59.16%	69.23%	63.41%	29.63%
KNN	57.50%	73.08%	53.66%	33.33%

Table 4.10 Prediction accuracy of entire and each class for all classifiers

4.2 Experiment 2 - Data Size Selection

Experiment 2 is a preparatory experiment in order to develop feature selection scientifically and comprehensively which uses one team's data to construct the dataset so as to make the other baseline format selection. The data size in experiment 4 has 15 columns and 240 rows (12 weeks' data), and the experiment 4 was performed before experiment 3, therefore the model is still using the ICT index as the default feature. The following **Fig 4.2** demonstrates specific information about one team's dataset. There are 14 independent numeric variables and 1-factor dependent variable.



	G_ICT	D1_ICT	D2_ICT	D3_ICT	D4_ICT	D5_ICT	M1_ICT	M2_ICT	M3_ICT	M4_ICT	M5_ICT	F1_ICT	F2_ICT	F3_ICT	Result
1	2.1	11.1	3.0	2.6	2.4	0.0	9.1	2.8	1.0	0.0	0.0	8.2	5.9	1.7	W
2	4.7	6.2	4.7	2.7	2.1	0.0	8.1	4.0	2.9	1.5	1.0	1.4	0.0	0.0	L
3	1.0	3.6	3.2	3.2	2.6	0.0	11.1	3.5	2.5	1.5	0.0	18.2	4.6	0.0	W
4	5.0	4.7	4.6	3.1	2.7	0.0	5.8	4.5	3.2	0.0	0.0	7.7	5.8	3.0	L
5	1.3	3.5	2.0	2.0	0.0	0.0	4.5	3.8	3.0	2.3	1.2	2.3	1.5	0.0	L
6	1.8	7.3	4.2	2.5	1.6	0.0	10.0	3.8	2.2	0.0	0.0	18.5	16.9	6.7	W
7	3.3	4.5	2.1	1.2	0.8	0.0	6.7	4.3	3.2	1.7	0.0	12.2	4.6	0.0	L
8	4.6	7.2	4.7	1.1	0.0	0.0	7.0	6.0	3.2	0.8	0.0	6.1	4.0	3.5	D
9	1.7	12.5	4.3	3.0	1.7	0.0	15.4	3.9	3.2	2.2	0.0	5.9	3.1	0.0	W
10	2.2	5.1	4.6	2.9	1.5	0.0	10.4	1.8	1.1	0.0	0.0	12.0	1.9	1.4	D
11	6.7	5.8	3.9	2.7	0.4	0.0	9.5	3.3	1.8	0.0	0.0	7.1	3.4	2.5	D
12	1.9	6.4	2.2	1.9	1.8	0.0	12.4	11.0	10.8	5.6	4.7	4.8	0.0	0.0	W
13	1.5	7.6	6.9	1.8	1.4	0.0	5.5	4.4	2.3	2.0	1.7	0.4	0.0	0.0	W
14	1.8	3.7	2.0	1.8	0.0	0.0	12.0	10.2	6.9	4.3	0.0	12.7	10.4	8.6	W

Fig 4.2 The dataset of one team's data

Table 4.11 illustrates the distribution of the entire data from the classes and data splitting using stratified 3 folds cross-validation where each fold has 36 for Draw, 62 for Loss and 62 for Win. There are 22.5% data which belong to class D, 38.75% data that belong to class L, and the other 38.75% data that belong to class W.

Category sets	Folds	D **(22.5%)**	L **(38.75%)**	W **(38.75%)**
Training	Fold 1 and 2	36	62	62
Test	Fold 3	18	31	31
Training	Fold 1 and 3	36	62	62
Test	Fold 2	18	31	31
Training	Fold 2 and 3	36	62	62
Test	Fold 1	18	31	31

Table 4.11 The result of stratified 3 folds cross validation in one team

Using the same Machine Learning processes as in experiment 2, **Table 4.12** shows the following predicting outcomes:

RF with ICT index in one team=

9+58+69/ (9+58+69+27+18+10+25+7+17=136/240≈56.67%

	Reference		
Prediction	D	L	W
D	9	10	7
L	27	58	17
W	18	25	69

Table 4.12 Overall confusion matrix of one team's data

In summary, as **Table 4.13** below demonstrates, one team's data got 56.67% prediction accuracy which is lower than 63.30% in two teams' data as columns. This result is more convincing to illustrate that the choice of two teams of data is a better decision to build the models in this project.

Comparison	Columns number	Rows number	Overall Prediction accuracy
One team's data in a row	29	240	56.67%
Two teams' data in a row	29	120	63.30%

Table 4.13 Comparison of results of one team's data and two teams' data

4.3 Experiment 3 - Feature Selection

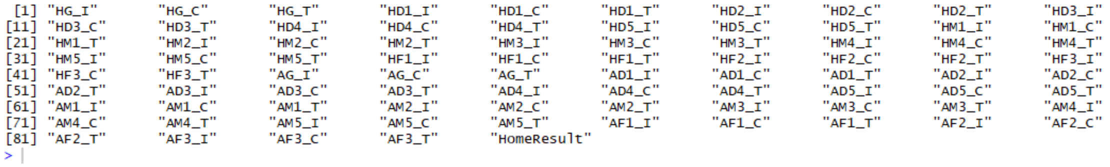
After doing the algorithms comparison analysis from **Section 4.1.1** to **Section 4.1.4**, Random Forest is the optimum choice for the following experiments to carry out the feature selection.

In this section, the project is going to compare the ICT index with its separate indexes in order to ensure the baseline feature which was mentioned in the previous chapter where the ICT or I, C and T indexes are one of the most characteristic and significant features in Fantasy Premier League. As a result of that, the project adapted the wrapper approach for it that let the higher prediction accuracy of ICT index and I, C and T to be the starting node, adding the BPS index, Selection and Cost, respectively in the first round to acquire the highest accuracy, and repeat this step until the accuracy stops improving. Furthermore, the following models use the method of adjusting parameters consistent with **Section 4.1.1**, build models in the ***randomForest*** package with the ***randomForest*** function, and find the most appropriate modelling parameters in the training set through stratified 3 folds cross-validation to predict the test set.

4.3.1 Experiment 4 – Baseline Feature Selection

ICT index versus Influence, Creativity and Threat

This experiment aims to compare the prediction accuracy of the ICT index with its separate indexes. The best result between them is going to be the starting node in the following feature selection. There are 120 rows and 29 columns of data in the ICT index which has been illustrated in the **Section 4.1.1**, as the comparative group, there are 120 rows and 85 columns' data of which 85 columns include 28 columns for Influence, 28 columns for Creativity, 28 columns for Threat, while the last one is the target variable as **Fig 4.3** below shows.



```
 [1] "HG_I"     "HG_C"     "HG_T"     "HD1_I"    "HD1_C"    "HD1_T"    "HD2_I"    "HD2_C"    "HD2_T"    "HD3_I"
[11] "HD3_C"    "HD3_T"    "HD4_I"    "HD4_C"    "HD4_T"    "HD5_I"    "HD5_C"    "HD5_T"    "HM1_I"    "HM1_C"
[21] "HM1_T"    "HM2_I"    "HM2_C"    "HM2_T"    "HM3_I"    "HM3_C"    "HM3_T"    "HM4_I"    "HM4_C"    "HM4_T"
[31] "HM5_I"    "HM5_C"    "HM5_T"    "HF1_I"    "HF1_C"    "HF1_T"    "HF2_I"    "HF2_C"    "HF2_T"    "HF3_I"
[41] "HF3_C"    "HF3_T"    "AG_I"     "AG_C"     "AG_T"     "AD1_I"    "AD1_C"    "AD1_T"    "AD2_I"    "AD2_C"
[51] "AD2_T"    "AD3_I"    "AD3_C"    "AD3_T"    "AD4_I"    "AD4_C"    "AD4_T"    "AD5_I"    "AD5_C"    "AD5_T"
[61] "AM1_I"    "AM1_C"    "AM1_T"    "AM2_I"    "AM2_C"    "AM2_T"    "AM3_I"    "AM3_C"    "AM3_T"    "AM4_I"
[71] "AM4_C"    "AM4_T"    "AM5_I"    "AM5_C"    "AM5_T"    "AF1_I"    "AF1_C"    "AF1_T"    "AF2_I"    "AF2_C"
[81] "AF2_T"    "AF3_I"    "AF3_C"    "AF3_T"    "HomeResult"
> |
```

Fig 4.3 List of all columns' names for I, C, T

Apart from that, the stratified 3 folds cross-validation without replacement is also applied in feature selection experiments to tune and select the suitable parameters for the models of Influence, Creativity and Threat. There are 9 class D, 14 class L and 18 class W in fold 1. 9 class D, 14 class L and 17 class W. 9 class D, 13 class L and 17 class W. Besides, the modelling, parameters adjustment and predicting processes of separate I, C and T indexes are similar, as discussed in **Section 4.1.1**.

The parameters fixing method was also applied in the ***randomForest*** function which operated 254 models: running from number 1 to 84 (the number of overall variables) in mtry and 6 values, 500, 1000, 1500, 2000, 2500 and 3000 in ntree for each 3-fold (which are generated from the training data by using stratified 3 folds cross-validation) in the training data. Pick the lowest model error rate among them, use these parameters to predict the three test sets, and then summarize their prediction results as the overall prediction accuracy. The following **Table 4.14** demonstrates that the most suitable parameters tuned in these three training models.

Test set	mtry	ntree	Prediction accuracy
Fold 1	9	500	70.73%
Fold 2	5	500	67.50%
Fold 3	6	1000	74.36%

Table 4.14 Prediction accuracy and parameters of each test set in I, C, T

In summary, combining the results of the three sets of prediction data, **Table 4.15** demonstrates the following conclusion:

RF Influence, Creativity and Threat =

(2+34+49)/ (2+7+18+2+34+5+0+3+49) =85/120≈70.8%

	Reference		
Prediction	D	L	W
D	2	2	0
L	7	34	3
W	18	5	49

Table 4.15 Overall confusion matrix for RF with I, C, T

Comparing the overall prediction accuracy while using the ICT index in **Section 4.1.1** in **Table 4.16**, the separate indexes are approximately 70.8% and the combined index is approximately 63.3%. Although ICT is a composite index for evaluating players, the data website does not give the weighting assigned by the three internal features: Influence, Creativity and Threat, so it does not represent the scientific dimension and accuracy of this comprehensive index. As a result, the following experiments would use the Influence, Creativity, and Threat as the new baseline features and develop the feature selection based on it.

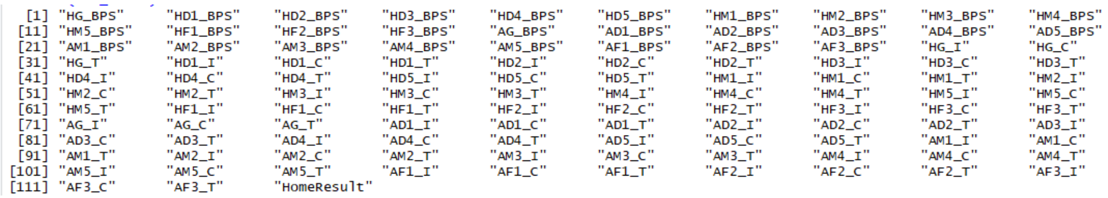
Features	Prediction accuracy
I, C, T	70.8%
ICT index	63.3%

Table 4.16 Prediction accuracy comparison between I, C, T and ICT index

From **Section 4.3.2**, the following experiments would add BPS index, Selection and Cost into the I, C and T's dataset respectively to develop the first layer of feature selection and summarize the findings and results of this series of models to begin the second layer of feature selection in the next stage.

4.3.2 I, C and T with BPS index

In this section, the experiments add BPS index with I, C and T to compare the prediction result with the other two features. As the first model of feature selection, **Section 4.3.2** will be described in more detail from the perspective of the three training models than others next. As **Fig 4.4** shows there are 120 rows and 113 columns including 84 columns for I, C and T, 28 columns for BPS index and the last one is the ***HomeResult*** which is the target variable.



```
  [1] "HG_BPS"     "HD1_BPS"    "HD2_BPS"    "HD3_BPS"    "HD4_BPS"    "HD5_BPS"    "HM1_BPS"    "HM2_BPS"    "HM3_BPS"    "HM4_BPS"
 [11] "HM5_BPS"    "HF1_BPS"    "HF2_BPS"    "HF3_BPS"    "AG_BPS"     "AD1_BPS"    "AD2_BPS"    "AD3_BPS"    "AD4_BPS"    "AD5_BPS"
 [21] "AM1_BPS"    "AM2_BPS"    "AM3_BPS"    "AM4_BPS"    "AM5_BPS"    "AF1_BPS"    "AF2_BPS"    "AF3_BPS"    "HG_I"       "HG_C"
 [31] "HG_T"       "HD1_I"      "HD1_C"      "HD1_T"      "HD2_I"      "HD2_C"      "HD2_T"      "HD3_I"      "HD3_C"      "HD3_T"
 [41] "HD4_I"      "HD4_C"      "HD4_T"      "HD5_I"      "HD5_C"      "HD5_T"      "HM1_I"      "HM1_C"      "HM1_T"      "HM2_I"
 [51] "HM2_C"      "HM2_T"      "HM3_I"      "HM3_C"      "HM3_T"      "HM4_I"      "HM4_C"      "HM4_T"      "HM5_I"      "HM5_C"
 [61] "HM5_T"      "HF1_I"      "HF1_C"      "HF1_T"      "HF2_I"      "HF2_C"      "HF2_T"      "HF3_I"      "HF3_C"      "HF3_T"
 [71] "AG_I"       "AG_C"       "AG_T"       "AD1_I"      "AD1_C"      "AD1_T"      "AD2_I"      "AD2_C"      "AD2_T"      "AD3_I"
 [81] "AD3_C"      "AD3_T"      "AD4_I"      "AD4_C"      "AD4_T"      "AD5_I"      "AD5_C"      "AD5_T"      "AM1_I"      "AM1_C"
 [91] "AM1_T"      "AM2_I"      "AM2_C"      "AM2_T"      "AM3_I"      "AM3_C"      "AM3_T"      "AM4_I"      "AM4_C"      "AM4_T"
[101] "AM5_I"      "AM5_C"      "AM5_T"      "AF1_I"      "AF1_C"      "AF1_T"      "AF2_I"      "AF2_C"      "AF2_T"      "AF3_I"
[111] "AF3_C"      "AF3_T"      "HomeResult"
```

Fig 4.4 List of all columns' name for I, C, T with BPS index

Training with Fold 1and 2, test with fold 3 in I, C and T with BPS index

The best prediction accuracy result is 82.5% which comes from both the default models and the adjusted model. The default model will be selected because of a slightly higher kappa value. The kappa coefficient is a method used to evaluate consistency in statistics. It can be used to assess the accuracy of a multi-class model. In the random forest function in the ***randomForest*** package the default setting is 10 for mtry and 500 for ntree.

Training with Fold 1and 3, test with fold 2 in I, C and T with BPS index

The lowest error rate with the training model is 22.16% in test fold 2, which acquired from the model with 4 for mtry and 1000 for ntree. After predicting the test data based on these parameters, the best prediction accuracy is 74.36%.

Training with Fold 1and 3, test with fold 2 in I, C and T with BPS index

As a result of the tests with the training models it is found shown that the adjusted model has with 6 for mtry and 2000 for ntree. The prediction results from it illustrate that all of them have the same value which is 80.49%. Comparing with the error rate in the training model, the final parameters selected the above decision: 6 for mtry and 2000 for ntree which model has the lowest error rate in its training model.

In summary, the overall prediction accuracy by using I, C and T with BPS Index in Random Forest is shown in the following **Table 4.17**:

RF I, C and T with BPS Index = (7+37+51) / (7+8+12+37+4+1+51) =95/120=79.2%

	Reference		
Prediction	D	L	W
D	7	0	1
L	8	37	0
W	12	4	51

Table 4.17 Overall confusion matrix for RF with I, C, T and BPS index

4.3.3 I, C and T with Cost

Using I, C and T with Cost is one series of models to do the feature selection, which has the same data size as **Section 4.3.2** that includes 120 rows and 113 columns in total. Apart from that, it is necessary to repeat the steps to figure out the suitable parameters and compare with the default settings. Consequently, the first model selected 11 for mtry and 3000 for ntree that has the lowest modelling error rate at 18.25% and the highest prediction accuracy at 75% in the first data subset which tested fold 3. The second model selected 10 for mtry and 500 for ntree which had a 27.5% error rate in modelling and the highest prediction accuracy rate of 70% in the second data subset which tested fold 2. The last one chose 31 for mtry and 500 for ntree, and it had a higher prediction accuracy than that with the default parameters which was 67.5% in the third data subset which tested fold 1. In summary, the following **Table 4.18** demonstrates the details for each class and the overall prediction accuracy:

RF I, C and T with Cost = (6+33+46)/ (6+8+13+5+33+3+2+4+46) = 85/120≈70.8%

	Reference		
Prediction	D	L	W
D	6	5	2
L	8	33	4
W	13	3	46

Table 4.18 Overall confusion matrix for RF with I, C, T and Cost

4.3.4 I, C and T with Selection

Using I, C and T with Selection is another one series of models to do the feature selection. As a consequence, 20 for mtry and 500 for ntree are the selection of the first model that had the highest prediction accuracy of 70.73% in the first data subset which tested fold 3. The second model decided to use 22 for mtry and 3000 for ntree that had a 30.75% error rate and the highest prediction accuracy 67.5% in the second data subset which tested fold 2. The third model selected 23 for mtry and 500 for ntree, with a prediction accuracy of 71.79% in the third data subset which tested fold 1. In summary, the following **Table 4.19** demonstrates the details for each class and the overall prediction accuracy:

RF I, C and T with Selection = (3+33+48)/ (3+4+20+5+33+6+1+3+48) =84/120=70%

	Reference		
Prediction	D	L	W
D	3	2	1
L	4	33	3
W	20	6	48

Table 4.19 Overall confusion matrix for RF with I, C, T and Selection

4.3.5 Conclusion in the first layer

In summary, BPS index, Cost and Selection were added, respectively into the dataset of I, C and T. As a result, as can be seen in Table 4.20, using I, C and T, and BPS index as the features produced the highest prediction accuracy of 79.2%, which increased by 9% when only using I, C and T. This means that the BPS index will be retained after the first layer.

Analysing the other two results, Cost and Selection cannot directly and clearly reflect the changes in the performance of players in each match. Cost fluctuations will not fluctuate greatly in a short period of time due to one match as, it is a long-term trend, so when this variable is added there is basically no change in accuracy. Furthermore, the statistics of Selection are mainly collected from soccer fans, and are more based on the degree of love for players, it not only from the direct performance on the court, but also the impact of their personal life, therefore, even if a player does not perform well in the last match, his selection value may be very high, which will affect the prediction of the match result. Even in the following **Table 4.20**, after adding Selection, the final result has been slightly reduced.

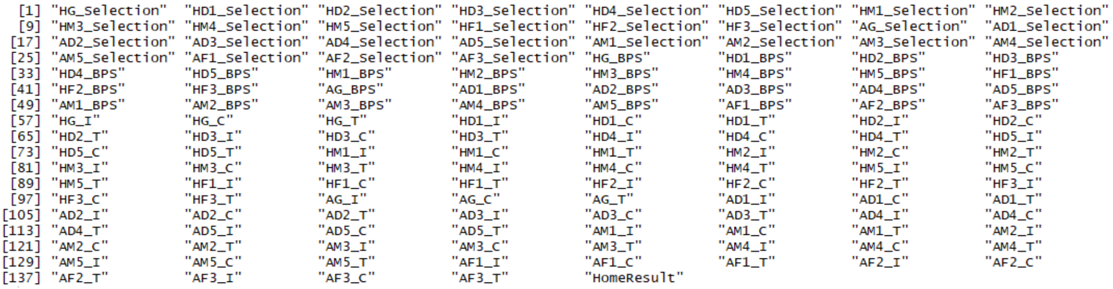
Features	Prediction accuracy
I, C and T with BPS index	79.2%
I, C and T with Cost	70.8%
I, C and T with Selection	70%

Table 4.20 Comparison result of the first layer in feature selection

In the following layer, the model adds Cost and Selection, respectively again based on I, C and T and the BPS index to carry out the second layer of feature selection.

4.3.6 I, C, T, BPS index with Selection

From **Section 4.3.6**, the models continue working on the feature selection, while in the second layer, the data size has extended to 141 columns which include 28 variables for Influence, 28 variables for Creativity, 28 variables for Threat, 28 variables for the BPS index, 28 variables for Selection and ***HomeResult*** as the target variable as **Fig 4.5**, below, shows.



```
  [1] "HG_Selection"  "HD1_Selection" "HD2_Selection" "HD3_Selection" "HD4_Selection" "HD5_Selection" "HM1_Selection" "HM2_Selection"
  [9] "HM3_Selection" "HM4_Selection" "HM5_Selection" "HF1_Selection" "HF2_Selection" "HF3_Selection" "AG_Selection"  "AD1_Selection"
 [17] "AD2_Selection" "AD3_Selection" "AD4_Selection" "AD5_Selection" "AM1_Selection" "AM2_Selection" "AM3_Selection" "AM4_Selection"
 [25] "AM5_Selection" "AF1_Selection" "AF2_Selection" "AF3_Selection" "HG_BPS"        "HD1_BPS"       "HD2_BPS"       "HD3_BPS"
 [33] "HD4_BPS"       "HD5_BPS"       "HM1_BPS"       "HM2_BPS"       "HM3_BPS"       "HM4_BPS"       "HM5_BPS"       "HF1_BPS"
 [41] "HF2_BPS"       "HF3_BPS"       "AG_BPS"        "AD1_BPS"       "AD2_BPS"       "AD3_BPS"       "AD4_BPS"       "AD5_BPS"
 [49] "AM1_BPS"       "AM2_BPS"       "AM3_BPS"       "AM4_BPS"       "AM5_BPS"       "AF1_BPS"       "AF2_BPS"       "AF3_BPS"
 [57] "HG_I"          "HG_C"          "HG_T"          "HD1_I"         "HD1_C"         "HD1_T"         "HD2_I"         "HD2_C"
 [65] "HD2_T"         "HD3_I"         "HD3_C"         "HD3_T"         "HD4_I"         "HD4_C"         "HD4_T"         "HD5_I"
 [73] "HD5_C"         "HD5_T"         "HM1_I"         "HM1_C"         "HM1_T"         "HM2_I"         "HM2_C"         "HM2_T"
 [81] "HM3_I"         "HM3_C"         "HM3_T"         "HM4_I"         "HM4_C"         "HM4_T"         "HM5_I"         "HM5_C"
 [89] "HM5_T"         "HF1_I"         "HF1_C"         "HF1_T"         "HF2_I"         "HF2_C"         "HF2_T"         "HF3_I"
 [97] "HF3_C"         "HF3_T"         "AG_I"          "AG_C"          "AG_T"          "AD1_I"         "AD1_C"         "AD1_T"
[105] "AD2_I"         "AD2_C"         "AD2_T"         "AD3_I"         "AD3_C"         "AD3_T"         "AD4_I"         "AD4_C"
[113] "AD4_T"         "AD5_I"         "AD5_C"         "AD5_T"         "AM1_I"         "AM1_C"         "AM1_T"         "AM2_I"
[121] "AM2_C"         "AM2_T"         "AM3_I"         "AM3_C"         "AM3_T"         "AM4_I"         "AM4_C"         "AM4_T"
[129] "AM5_I"         "AM5_C"         "AM5_T"         "AF1_I"         "AF1_C"         "AF1_T"         "AF2_I"         "AF2_C"
[137] "AF2_T"         "AF3_I"         "AF3_C"         "AF3_T"         "HomeResult"
```

Fig 4.5 List of all columns' name for I, C, T, BPS index with Selection

Except for the difference between the data size between the first layer and the second layer, the processes are the same as previous experiments, building models and tuning parameters based on stratified 3 folds cross-validation, calculating their overall prediction accuracy. The first model is built based on fold 1 and fold 2 which has the error rate of 17.75% and adjusts the parameters with 20 for mtry and 2500 for ntree, with the result that the prediction accuracy of this subset model is 79.49%. The second model is built based on fold 1 and fold 3 which has an error rate of 14.52% and selects the default parameters with 11 for mtry and 500 for ntree, with the result that the predictive accuracy of this subset model is 80.49%. The third model is built based on fold 2 and fold 3 which has an error rate of 12.5% and selects the default parameters with 15 for mtry and 3000 for ntree. **Table 4.21** shows the confusion matrix of this subset model. The following equation shows the overall prediction result which is 80.8%:

RF I, C, T, BPS index with Selection= (6+40+51)/ (6+5+16+40+1+1+51) =97/120≈80.8%

	Reference		
Prediction	D	L	W
D	6	0	1
L	5	40	0
W	16	1	51

Table 4.21 Overall confusion matrix for RF with I, C, T, BPS index and Selection

4.3.7 I, C, T, BPS index with Cost

This model uses Influence, Creativity, Threat, and the BPS index with Cost in these experiments which aims to do the second layer feature selection. The format and size of the dataset are the same as in **Section 4.3.6** except that the Cost feature is used instead of Selection. The first model is built based on fold 1 and fold 2 where the adjusted parameters are 35 for mtry and 3000 for ntree. The result was that the prediction accuracy of this subset model was 82.5%. The second model was built based on fold 1 and fold 3 which has the error rate 16.05% with the parameters with 9 for mtry and 3000 for ntree. The prediction accuracy of this subset model was 74.36%. The third model was built based on fold 2 and fold 3 which had the error rate of 20.25% and selected the default parameters with 9 for mtry and 500 for ntree, with the outcome that the prediction accuracy of this subset model was 82.93%. **Table 4.22** demonstrates the overall predictive results:

RF I, C, T, BPS index with Cost= (7+37+51)/ (7+3+17+3+37+2+1+51) =95/120≈79.2%

	Reference		
Prediction	D	L	W
D	7	3	1
L	3	37	0
W	17	2	51

Table 4.22 Overall confusion matrix for RF with I, C, T, BPS index and Cost

4.3.8 Conclusion on the second layer

In summary, as **Table 4.23** below illustrates, adding Selection based on the first layer would produce slightly better progress than before where 80.8% was reached. Although mentioned in the first layer, Selection brings a very small reduction, but after adding the BPS index, it has a positive impact on the overall predictive compared with Cost, because it is more fluid as the choice of players will be different for each game. Comparing with Cost, if the player does not have a better performance than his previous game, he might keep the same Cost as the last one.

Features	Prediction accuracy
I, C, T, BPS index with Selection	80.8%
I, C, T, BPS index with Cost	79.2%

Table 4.23 Comparison result of the second layer in feature selection

4.3.9 I, C, T, BPS index, Selection with Cost

This series of models is used to finish the final layer of feature selection. If the result is better than the second layer, it is going to add Cost based on I, C, T, the BPS index and Selection, it also means that all of the features will be used in the final models, otherwise, keep the features from the second layer in the end. As is illustrated below, the results of this experiment are consistent with **Section 4.2.7**, therefore the final feature will continue to use the results in **Section 4.2.6** calculated from **Table 4.24**.

Final= (7+37+51)/ (7+5+15+3+37+1+1+51) =95/120≈79.2%

	Reference		
Prediction	D	L	W
D	7	3	0
L	5	37	1
W	15	1	51

Table 4.24 Overall confusion matrix for RF with I, C, T, BPS index, Selection with Cost

4.3.10 Overall Feature Conclusion

From **Section 4.3.1** to **Section 4.3.9**, the complete feature selection process has been fully presented. As the final result shows in **Table 4.25**, the most useful and practical feature based on the random forest classifier prediction is to use **Influence, Creativity, Threat, BPS index and Selection**, and its prediction result is **80.8%**. Combined with the results given in **Table 4.25**, 70.8% of the first red mark is the result of the baseline feature selection, and the 80.8% in red is the final selection result and prediction accuracy.

ICT index (63.30%)	I, C, T (70.8%)	BPS index	Cost	Selection	Prediction accuracy
	√	√			79.2%
	√		√		70.8%
	√			√	70.0%
	√	√		√	80.8%
	√	√	√		79.2%
	√	√	√	√	79.2%

Table 4.25 Final results comparison of Feature selection

From the predictive results the change in the predicted value is small, because compared with other features the BPS index is an objective feature for the player evaluation. The BPS index specification table in **Section 3.3** explains that it is obtained by adding or subtracting the score according to the detailed on-field data, therefore, after adding the BPS index, the prediction accuracy rate has increased a lot even exceeded 80%.

4.4 Discussion

4.4.1 Matches Analysis

Following the series of experiments with feature selections detailed in **Section 4.3** in Section **4.3.6** the features that are finally determined to be used as the final features selection were chosen and these are: Influence, Creativity, Threat, BPS index and Selection in Random Forest. Further, the ultimate prediction accuracy of this entire project is 80.8% as 4.3.10 has shown. The evaluation results of each class which are derived from the assessments of their predictive sensitivity by using Recall index are demonstrated in **Table 4.26.** As is shown, the predictive sensitivity of class D, which represents the Draw matches, is about 22.22%. Class L which represents the Loss is about 97.56%, and the last class W, which represents the matches won, is about 98.08%. Both class W and class L have only one misclassification match and their average predictive accuracy is greater than 97%. This result is probably generated due to the limited data size in this model. There are 5 actual draw games which were misclassified as lost games and 16 actual draw games were misclassified as games won. The first conclusion is that one side of teams had excellent performance but did not score a goal or had the same goals as the opponent at the end of a game, and the second conclusion is that the performance of the substitute players had a significant impact on the outcome.

	Reference		
Prediction	D	L	W
D	6	0	1
L	5	40	0
W	16	1	51
Recall	22.22%	97.56%	98.08%

Table 4.26 Final confusion matrix after feature selection

4.4.2 Players Analysis

Fig 4.6 to **Fig 4.8** illustrated the importance of the top ten variables affecting the result of the soccer game when using fold 1 to fold 3 as test sets. Online soccer players or coaches can not only determine which position and players have the hugest influence on the result of the match but also find out which features of a specific player can impact the match result.

The left index in the figure is MeanDecreaseAccuracy which is a value of a feature is changed to a random number, and the accuracy of random forest prediction is reduced. A larger value indicates the greater importance of the feature. Combining the results of the three sub-models indicates that the BPS index of the first place’s defender from the Away team is the primary feature which influences the final result in test fold 1 and 3, while the BPS index of the first place’s defender from the Home team is also the most significant element in test fold 2. Furthermore, the BPS indexes of other places’ defenders from both the home and away teams are the main features affecting the match. As introduced in Chapter 3, the BPS index value is a direct coefficient of performance on the player's field, and the defender is the last line of defence except for the goalkeeper in the match, according to the rules of the soccer match, where games won acquire 3 points, drawn games acquire 1 point, and lost games do not acquire any points. So, this is why these features in relation to all the places of defenders are so important in the starting line-up. Therefore, these features in the three figures are nearly half of the total.

From the perspective of the offensive position, the impact of either home or away team's No. 1 forwards on the result of the match is also relatively high, and both the BPS index and Influence are essential features for them. Moreover, the first place’s midfielders from both sides also play an important role in the game because they are the connection between the attack and defence in the team and, as **Fig 4.8** demonstrates, even the second place’s midfielder has a significant influence on the results.

The right index MeanDecreaseGini is the effect of each feature on the heterogeneity of the observed values where each node of the classification tree is calculated to compare the importance of the variables. The larger the value, the more important the variable is. Here the roles and results of MeanDecreaseGini and MeanDecreaseAccuracy are relatively similar.

Combining the two coefficients of MeanDecreaseAccuracy and MeanDecreaseGini, the away player's data is a more important factor in determining the result of the match. Although the home players have home advantage and more fans’ support, they also have more pressure than the away players. As a result, from the previous two *importance* figures, the first place’s forward of the away team is also the key element which decides the score. From a tactical point of view, the away team pays more attention to the defence to ensure that no points are lost. On this basis, the away team can catch the home team's mistakes to create scoring opportunities, which can also provide some help for coaches and online soccer players when arranging tactics and choosing the starting line-up.

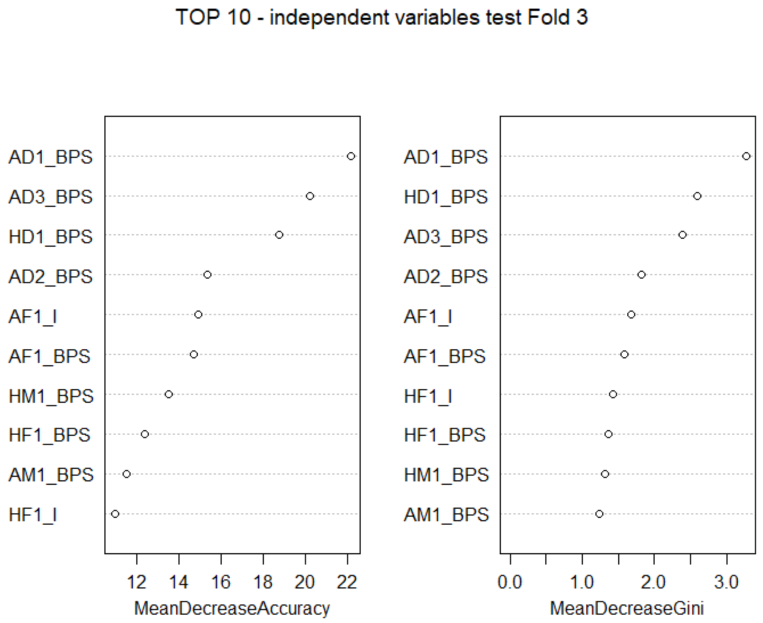


Fig 4.6 Feature importance of testing fold 3

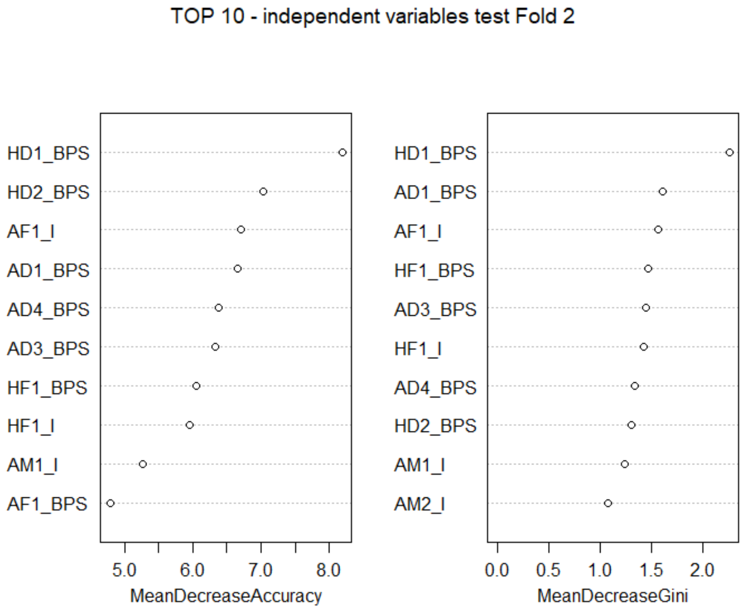


Fig 4.7 Feature importance of testing fold 2

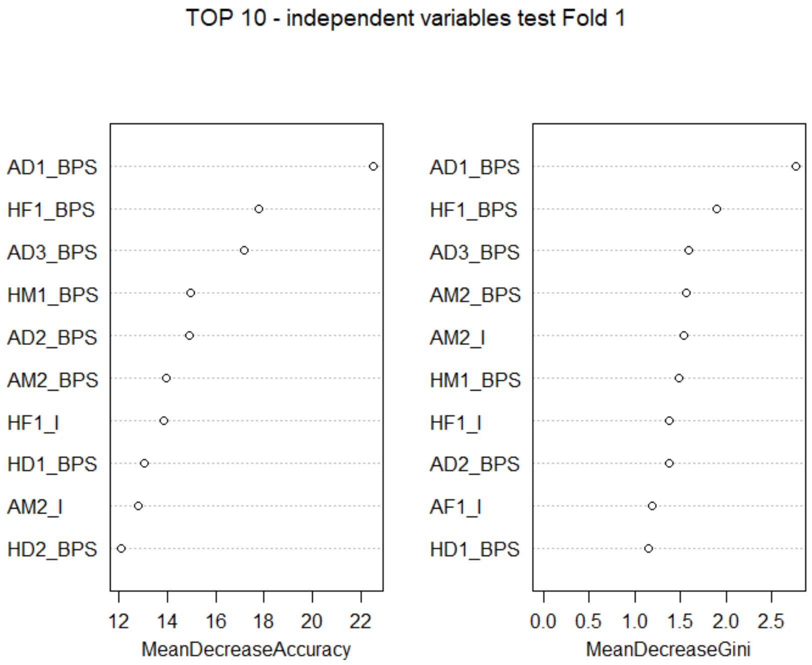


Fig 4.8 Feature importance of testing fold 1

4.5 The Latest Prediction

This section aims to predict the result on the latest matches in English Premier League which happened from the 13th to 15th match weeks based on the 1st – 12th weeks' model so as to estimate its stability.

The size of the data set used is one of the significant limitations of this project. In the modelling process, the Premier League has played a total of 12 weeks (120 matches) of competition, and this is the reason why the project selected 12 weeks of data. But when the project was coming to an end, more matches were taking place, so in **Section 4.4**, the model will use the data from the previous 12 weeks as the training set, and the data from the 13th to the 15th week (which just finished) is used as the test set. The project observes and evaluates the situation based on player information in the starting line-up to predict the result of the match.

Combining the experimental results from **Section 4.1** to **Section 4.3**, the latest model will use Influence, Creativity, Threat, BPS index and Selection as features, using the Random Forest as a classifier. The dataset includes the results shown in **Table 4.27** below, which demonstrates the data distribution of the training and test sets.

	Data size	Number of W	Number of D	Number of L
Training data	120	52	41	27
Test data	30	15	9	6

Table 4.27 Data distribution of training data and test data

Like all the models above, it also requires stratified 3 folds cross-validation in the training set. Unlike before, the arithmetic square root of the independent variable number (140) is used as the constraint value of the mtry to reduce the training time in this model. The range of 500, 1000, 1500, 2000, 2500, and 3000 is still ntree, and the test is performed with three folds. The parameter with the lowest error rate of the model is selected as the final parameter. As **Fig 4.9** shows, according to the description in the above sentence, a total of 216 models($\sqrt{140} \times 6 \times 3$) will be established in the training set. The point represented by the red circle in the figure is the 142th model, which has the lowest model error rate with 26.76%.

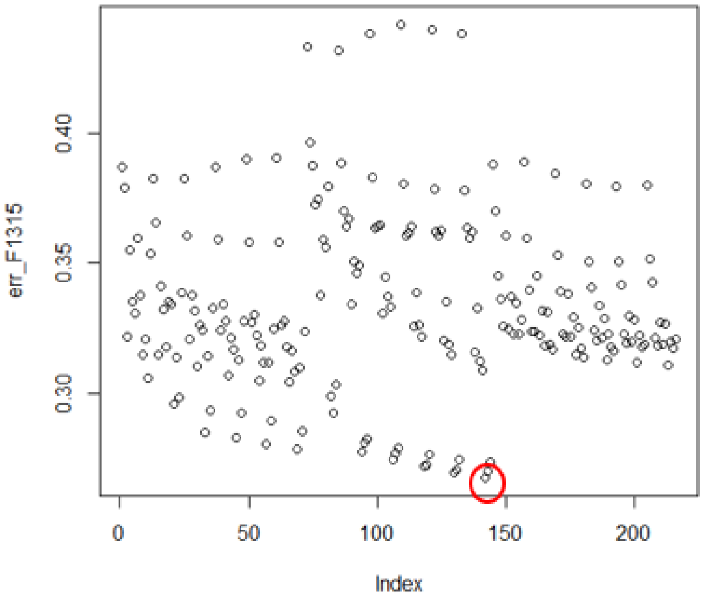
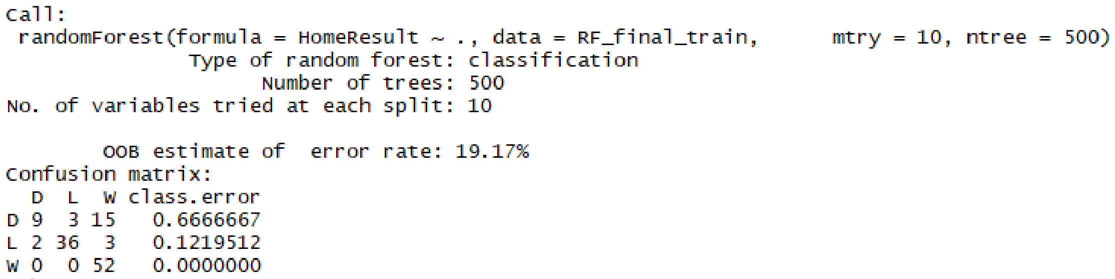


Fig 4.9 Scatter Plot of the error rate of 216 sub models

The parameter combination of the 142th model is mtry=10 and ntree=500. Select this set of parameters to establish the final training model. **Fig 4.10** demonstrates that the lowest error rate based on the OOB method (out-of-bag error) of the training model is 19.17%. OOB is a method of evaluation within a random forest. It builds an unbiased estimate of the error during the generation process that calculates the misclassification rate inside of the training model.

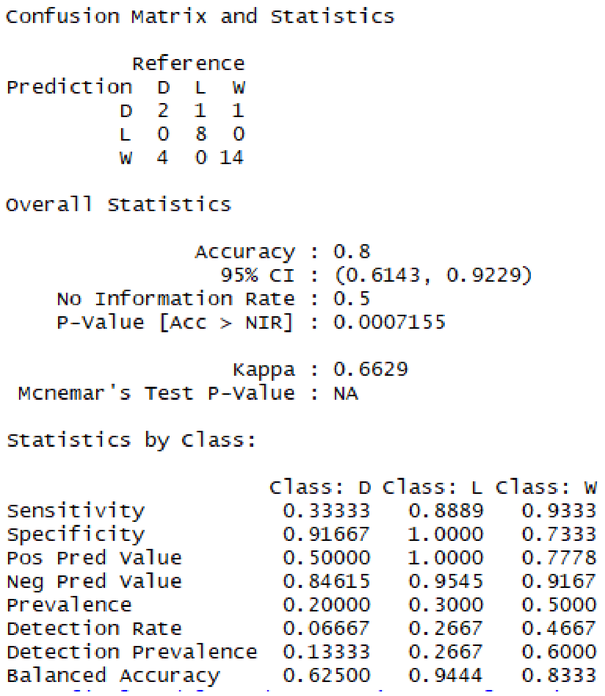


```
Call:
 randomForest(formula = HomeResult ~ ., data = RF_final_train,      mtry = 10, ntree = 500)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 10

        OOB estimate of  error rate: 19.17%
Confusion matrix:
  D  L  W class.error
D 9  3 15   0.6666667
L 2 36  3   0.1219512
W 0  0 52   0.0000000
```

Fig 4.10 Assessment of the training model

The test set was predicted with the adjusted training model, and finally, 80% of the prediction accuracy was obtained, as shown in the following **Fig 4.11**. Besides, it can be seen that the prediction rates (Sensitivity value also called Recall, which index has been used above to measure the prediction accuracy of each class all the time) of class W and class L are quite good, reaching 93.33% and 88.89% respectively. Class D also reached 33.33%, although it is much worse than the other two, it is still not easy to identify the draw in the football match, but is better than the result from the model outlined in **Section 4.4**. The Kappa value is 0.6629, according to the division of kappa in **Section 3.7**, it belongs to substantial consistency with training model.



```
Confusion Matrix and Statistics

          Reference
Prediction  D  L  W
         D  2  1  1
         L  0  8  0
         W  4  0 14

Overall Statistics

               Accuracy : 0.8
                 95% CI : (0.6143, 0.9229)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : 0.0007155

                  Kappa : 0.6629
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: D Class: L Class: W
Sensitivity           0.33333   0.8889   0.9333
Specificity           0.91667   1.0000   0.7333
Pos Pred Value        0.50000   1.0000   0.7778
Neg Pred Value        0.84615   0.9545   0.9167
Prevalence            0.20000   0.3000   0.5000
Detection Rate        0.06667   0.2667   0.4667
Detection Prevalence  0.13333   0.2667   0.6000
Balanced Accuracy     0.62500   0.9444   0.8333
```

Fig 4.11 Confusion Matrix and Statistics of 13th to 15th result

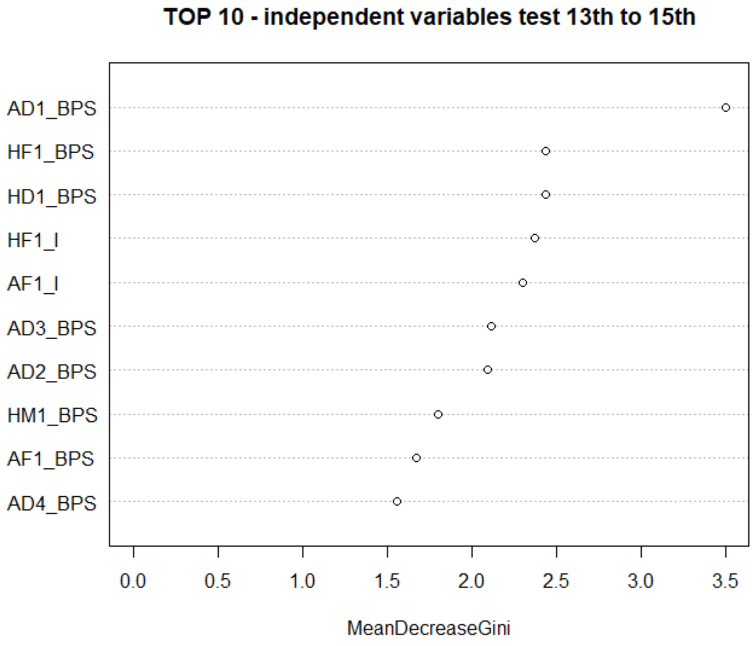


Fig 4.12 Top 10 importance independent variables from the final result

As **Fig 4.12**, above, demonstrates, the findings are similar to those discussed in the analysis in **Section 4.4. The** BPS index is still one of the most important features of the player, and the performance of the away players is more influential in relation to the final result. The difference is that the player variables from the offensive side are more important than the defensive side.

5. CONCLUSION

This final chapter presents a general description of the entire project, summarising the final model results, in addition to detailing the shortcomings that need improvement in the future.

5.1 Research Overview

This research finally answered the research question indicated in Chapter 1, verifying that the personal data of the players in each line-up can be used to predict the result of a soccer game. In addition, it provided information relevant to decisions and tactics for online soccer players and coaches, using the model. The model achieved 81.8% prediction accuracy based on the first 12 weeks of soccer games in the English Premier League, as well as 80% accuracy for predicting the 13th to 15th weeks based on the same training model.

5.2 Problem Definition

In this project, the changes in the player trading market in different seasons would have a particular impact on the statistics. Data from the first 12 weeks of the Premier League 2018-2019 season was taken, which means that only 120 games were used to build the dataset. In the feature selection step, when four variables were added, the number of rows and columns was almost the same, and when five variables were added, the number of columns even exceeded the number of rows. Since this project used a non-linear model in supervisory learning, it took more time to train than a linear model. In addition to adding more seasons' data, the further project must consider the choice of features in the future. Although the ICT index and the BPS index are quite distinctive features in the Fantasy Premier League, these features are relatively subjective. If the research can collect more direct player data, one may further improve the prediction accuracy, via such methods as using the BPS index sub-items as features.

5.3 Design/Experimentation, Evaluation & Results

This project used the data of the two teams' starting players in the first 12 weeks of the 2018-2019 season to predict the final results of soccer games in the English Premier League. The features used for modelling and predicting were from the Fantasy Premier League, provided via the official statistics of the English Premier League official website and EA Sports. By comparing four popular supervised Machine Learning algorithms, including Random Forest, Support Vector Machine, Naïve Bayes and K-Nearest

Neighbour, the classifier that was most suitable for this project was selected. In the comparison of model selections, the dataset selected the ICT index as the feature which was the most characteristic and representative feature in the Fantasy Premier League. As a result of the comparison, Random Forest was chosen, having a prediction accuracy of 63.30%. After that, the model determined that the data of the two teams' players were more accurate than the data of one team. On this basis, influence, creativity, and threat are established as final baseline features by comparing the ICT index, which is a compound feature, with separate versions. Afterward, the features BPS index, cost and selection are added in turn by the wrapper method to perform multi-level feature selection. Finally, the model achieved 80.8% prediction accuracy by using influence, creativity, threat, BPS index and selection. Moreover, according to the expansion experiment detailed in **Section 4.5**, predicting the matches in the 13th to the 15th weeks, the model also obtained 80% prediction accuracy.

Compared with other research literature that predict the result of football matches that Ulmer & Fernandez (2014) got the best prediction accuracy 52% with linear classifier from stochastic gradient descent using gameday data and team performance which quite lower than 80.8% in this project. Hijmans & Bhulai (2017) applied Generalized boosted models to predict the match result of Dutch national football team which generated the highest prediction accuracy 60.22% with squad attributes and players attributes. It is also less accurate than the prediction accuracy of this project. Hucaljuk and Rakipović (2011) achieved 68% in ANN classifier for the Champions League. The best prediction accuracy which mentioned in the literature review has around 96% in their own polynomial algorithm (Martins et al., 2017).

5.4 Contributions and impact

The primary purpose of this research was to help predict the results of soccer games using player data relating to the starting line-up. The average overall accuracy can exceed 80% based on the models applied to the first 12 weeks and the 13th to the 15th weeks. This provides an effective predictive reference for fans and online soccer players. Further analysis of this Random Forest model can also provide information for strategic tactical arrangements. As indicated in the conclusion of **Section 4.4.2**, it is necessary to pay attention to the choice of defenders in their team, but also to observe the performance of the opponent's defensive line and the performance of the attacking line, as the highest-performing forward will also have a significant impact on the result of the match.

5.5 Future Work & recommendations

Except for Random Forest, Support Vector Machine, Naïve Bayes and K-Nearest Neighbour applied in this research. Artificial neural network (Hucaljuk and Rakipović, 2011) and Bayesian Network (Owramipur, Eskandarian & Mozneb, 2013) might be other ideal models by which to expand algorithm selection. These are frequently used methods in other literature, and they can further explore the relationship between players in different positions and the direct or indirect effects on the outcome of a soccer game. Both Delen, Cogdell and Kasap (2012) and Goddard (2005) have suggested that algorithms can be compared from the perspective of classification models and regression models, which are the two branches of supervisory learning, and this research focused on the comparison and selection of classification models, thereby indicating that more Machine Learning algorithms in the future can further determine the stability of the model and improve the accuracy of the prediction.

In addition, as mentioned in **Section 5.2**, collecting and adding more objective and specific features instead of general indexes in future models might provide general recommendations or strategy for fans, online soccer players or coaches. As **Section 2.4** suggested, adding the statistics of referees or substitute players to expand the model can predict the final result based on the data of all the participants on the field.

BIBLIOGRAPHY

Ancona, N., Cicirelli, G., Branca, A., & Distante, A. (2001). Goal detection in football by using Support Vector Machines for classification. *IJCNN'01. International Joint Conference On Neural Networks. Proceedings (Cat. No.01CH37222)*, 611-616. doi: 10.1109/IJCNN.2001.939092

Baboota, R., & Kaur, H. (2018). Predictive analysis and modelling football results using Machine Learning approach for English Premier League. *International Journal Of Forecasting*. doi: 10.1016/j.ijforecast.2018.01.003

Bush, M., Barnes, C., Archer, D., Hogg, B., & Bradley, P. (2015). Evolution of match performance parameters for various playing positions in the English Premier League. *Human Movement Science*, 39, 1-11. doi: 10.1016/j.humov.2014.10.003

Cui, T., Li, J., Woodward, J., & Parkes, A. (2013). An ensemble based Genetic Programming system to predict English football premier league games. *2013 IEEE Conference On Evolving And Adaptive Intelligent Systems (EAIS)*, (978-1-4673-5855-2), 1-6. doi: 10.1109/EAIS.2013.6604116

Delen, D., Cogdell, D., & Kasap, N. (2012). A comparative analysis of data mining methods in predicting NCAA bowl outcomes. *International Journal Of Forecasting*, 28(2), 543-552. doi: 10.1016/j.ijforecast.2011.05.002

Goddard, J. (2005). Regression models for forecasting goals and match results in association football. *International Journal Of Forecasting*, 21(2), 331-340. doi: 10.1016/j.ijforecast.2004.08.002

Hai, M., Zhang, Y., & Zhang, Y. (2017). A Performance Evaluation of Classification Algorithms for Big Data. *Procedia Computer Science*, 122, 1100-1107. doi: 10.1016/j.procs.2017.11.479

Hijmans, A., & Bhulai, S. (2017). Dutch football prediction using Machine Learning classifiers, pp.1-24.

Hucaljuk, J., & Rakipović, A. (2011). Predicting football scores using Machine Learning techniques. *2011 Proceedings Of The 34Th International Convention MIPRO*, (978-953-233-059-5), pp.1-5.

Jelinek, H., Kelarev, A., Robinson, D., Stranieri, A., & Cornforth, D. (2014). Using meta-regression data mining to improve predictions of performance based on heart rate dynamics for Australian football. *Applied Soft Computing*, 14, 81-87. doi: 10.1016/j.asoc.2013.08.010

Joseph, A., Fenton, N., & Neil, M. (2006). Predicting football results using Bayesian nets and other Machine Learning techniques. *Knowledge-Based Systems*, 19(7), 544-553. doi: 10.1016/j.knosys.2006.04.011

Leung, C., & Joseph, K. (2014). Sports Data Mining: Predicting Results for the College Football Games. *Procedia Computer Science*, 35, 710-719. doi: 10.1016/j.procs.2014.08.153

Lu, K., Chen, J., Little, J., & He, H. (2018). Lightweight convolutional neural networks for player detection and classification. *Computer Vision And Image Understanding*, 172, 77-87. doi: 10.1016/j.cviu.2018.02.008

Martins, R., Martins, A., Neves, L., Lima, L., Flores, E., & do Nascimento, M. (2017). Exploring polynomial classifier to predict match results in football championships. *Expert Systems With Applications*, 83, 79-93. doi: 10.1016/j.eswa.2017.04.040

McHale, I., & Relton, S. (2018). Identifying key players in soccer teams using network analysis and pass difficulty. *European Journal Of Operational Research*, 268(1), 339-347. doi: 10.1016/j.ejor.2018.01.018

Min, B., Kim, J., Choe, C., Eom, H., & (Bob) McKay, R. (2008). A compound framework for sports results prediction: A football case study. *Knowledge-Based Systems*, 21(7), 551-562. doi: 10.1016/j.knosys.2008.03.016

Owramipur, F., Eskandarian, P., & Mozneb, F. (2013). Football Result Prediction with Bayesian Network in Spanish League-Barcelona Team. *International Journal Of Computer Theory And Engineering*, 812-815. doi: 10.7763/ijcte. 2013.v5.802

Pariath, R., Shah, S., Surve, A., & Mittal, J. (2018). Player Performance Prediction in Football Game. *2018 Second International Conference On Electronics, Communication And Aerospace Technology (ICECA)*, 1-6. doi: 10.1109/ICECA.2018.8474750

Razali, N., Mustapha, A., Yatim, F., & Ab Aziz, R. (2017). Predicting Football Matches Results using Bayesian Networks for English Premier League (EPL). *IOP Conference Series: Materials Science And Engineering*, 226, 012099. doi: 10.1088/1757-899x/226/1/012099

Sarangi, S., & Unlu, E. (2010). Key Players and Key Groups in Teams: A Network Approach Using Soccer Data. *SSRN Electronic Journal*. doi: 10.2139/ssrn.1679776

Ulmer, B., & Fernandez, M. (2014). Predicting Soccer Match Results in the English Premier League. *School Of Computer Science Stanford University*, 1-5.

Weston, M., Bird, S., Helsen, W., Nevill, A., & Castagna, C. (2006). The effect of match standard and referee experience on the objective and subjective match workload of English Premier League referees. *Journal Of Science And Medicine In Sport*, 9(3), 256-262. doi: 10.1016/j.jsams.2006.03.022

Weston, M., Castagna, C., Impellizzeri, F., Rampinini, E., & Abt, G. (2007). Analysis of physical match performance in English Premier League soccer referees with particular reference to first half and player work rates. *Journal Of Science And Medicine In Sport*, 10(6), 390-397. doi: 10.1016/j.jsams.2006.09.001

Yijie, Z., & Jun, X. (2015). Competition Results Prediction Model Based on Athlete Ability Data Simulation and Analysis. *2015 Sixth International Conference On Intelligent Systems Design And Engineering Applications (ISDEA)*. doi:10.1109/isdea.2015.64