Department of Computer Science

MSc Data science and analytics

Academic Year 2021-2022

Big data on prediction of professional football players' transfer market value using both real game statistics combined with FIFA game attribute data

Manraj Rai
2135198

A report submitted in partial fulfilment of the requirement for the degree of Master of Science

Brunel University
Department of Computer Science
Uxbridge, Middlesex UB8 3PH
United Kingdom
Tel: +44 (0) 1895 203397
Fax: +44 (0) 1895 251686

ABSTRACT

The topic of football player transfer market fees has been at the forefront of discussions over the last decade or so. With Gareth Bale being the first player to be sold for about 100 million euros in 2013 and then subsequently the record signing fee of Neymar which was worth over 200 million euros in 2017, the questions arises of whether or not these enormous transfer fees are justifiable in terms of the footballing skill of the player being purchased or the popularity/marketability of the player in question. However this study will focus solely on the relationship between transfer market value and player skill.

This topic of exploring the transfer market value is highly relevant as the increase of transfer market value has been linked to the similar increase in player weekly wages which have been massively inflated and the need for financial fair play rules have had to be introduced to make the market place more equitable for all clubs involved and to keep professional football to a competitive standard. The question also arises of whether or not the money generated by fans could be better spent elsewhere in the footballing infrastructure instead of these transfer deals.

This study will use a combination of both unsupervised and supervised machine learning methods to explore and predict the transfer market value of football players using the 2022 rendition of the FIFA dataset. The unsupervised methods such as principal component analysis and clustering will be used when exploring the datasets and the supervised methods will be used purely for the prediction. The problem has been set as a classification task rather than regression and the supervised classifiers will be used to predict whether or not a player is worth over a certain price threshold. The novel approach that has been applied to this study is that during the data exploration, clustering has been used on a dataset that contains real match data and this has been compared to the clustering results of the FIFA dataset to make some general comparisons. The ultimate purpose for using classification was to use the true positive, true negative, false positive and false negative counts produced by each predictor to infer whether or not players were fairly valued, undervalued or overvalued. The criteria and aims were set according to these classifiers as the majority of variables used in the prediction were skill based attributes.

According to the results of the classifiers used in this study generally the majority of the players were fairly valued and out of the minority of players that were not fairly valued, the population of players that were overvalued was larger than those that were undervalued.

ACKNOWLEDGEMENTS

I would like to thank Dr Lorraine Ayad for providing continuous support and supervision throughout the entirety of this project.

I certify that the work presented in the dissertation is my own unless referenced

Signature.....Manraj Rai.....

Date.....30/08/2022.....

TOTAL NUMBER OF WORDS:

13252

Table of contents

Chapter 1: Introduction/Justification 4

1.1 Are clubs overpaying or underpaying one another?

1.2 The relationship between transfer market value and player wage

1.3 Aims and criteria

Chapter 2: Literature review 6

2.1 Linear vs non linear supervised methods

2.2 Attributes outside of match performance/skill that affect salary/transfer market value

2.3 Fans holding greater influence in a player's value rather than match performance

2.4 Relationship between player performance and salary in the NBA is non linear

2.5 External factors play an insignificant role in determining MLB players' salaries

Chapter 3: Approach/ methodology 13

Chapter 4: Data cleaning and Exploratory data analysis 15

4.1 Data cleaning for FIFA dataset

4.2 Exploratory data analysis for FIFA dataset

4.3 Principal component analysis for FIFA dataset

4.4 Creation, cleaning and exploratory data analysis of real match dataset

4.5 Cluster analysis and comparison

Chapter 5: Binary classification results: Decision tree and random forest 25

Chapter 6: Binary classification results: Neural networks 28

Chapter 7: Binary classification results: Support vector machines 33

Chapter 8: Binary classification results: K nearest neighbours 37

Chapter 9: Findings and conclusion 40

9.1 Binary classifiers - Looking at counts of true positives and true negatives to find the number of fairly valued footballers

9.2 Binary classifiers - Looking at counts of false positives to find the number of undervalued footballers

9.3 Binary classifiers - Looking at counts of false negatives to find the number of overvalued footballers

Chapter 10: Future progression on current research 43

References 44

Appendix 47

Chapter 1: Introduction/Justification

1.1 Are clubs overpaying or underpaying one another?

The first reason to carry out this study in predicting players' transfer market values was to aid investigations into identifying whether or not clubs were underpaying or overpaying one another to get the transfers deals completed and over the line. So this area of research could be built upon and could possibly assist with the idea of more impartial deals being made, overall saving all clubs involved a significant amount of money being spent from their transfer budget and creating a more equitable transfer market. This would especially benefit clubs who do not command an abundant amount of financial resources. Park and Lee (2017) argue that the reasoning for these excessive transfer fees are down to the fact that the club negotiation teams offer large sums of money due to the pressure and demand of fans on social media as well as the speculation caused from the press media of getting deals completed.

This study indicates that indeed determining factors outside of match performance affect the transfer fees paid by clubs. Moreover, the predictive models created in this study could be used in future research to either reinforce or perhaps diminish the claims made previously. Also if the supervised machine learning predictors are found to have high predictive accuracies, clubs would have the strategic option to create and utilise even more complex models by consulting with machine learning engineers to forecast future transfer fees of both current and upcoming talent to assist in negotiations.

1.2 The relationship between transfer market value and player wage

The second reason to carry out this study is due to the fact that transfer values are correlated to player weekly wages, so this study could be used in future research to infer whether or not specific individual players are being underpaid or overpaid. Maguire (2021) mentions that the premier league player salaries alone have more than doubled over the last decade with the average salary during 2010 being roughly £32,000 and the average salary during 2019 being roughly £72,000 which is around a 125% markup. Although transfer market value is not the greatest indicator of a player's salary, as Auberg (2022) mentions that years left on a player's contracts is a more reliable indicator, these two factors have still both steadily risen alongside one another over recent decades and hence this relationship could be better explored in the future studies to gain previously unseen crucial insights. Andreff (2018) claims that the combination of relaxed financial fair play (FFP) rules with the high demand of top end elite footballers has greatly inflated both their transfer market values and player wages in the French Ligue 1 division, whilst footballers who are not quite on the elite level tend to sign contracts which are worth significantly less money in comparison. An example of a player who is broadly accepted as elite, Kylian Mbappe recently renewed his contract which was worth a reported fee of a million great british pounds a week, (Sky News, 2022).

1.3 Aims and criteria

This study focuses on the prediction of the transfer market value using a variety of machine learning methods. Using the question mentioned above that ponders whether the transfer market values being set are either too high or too low for various players by their respective clubs, a set of aims that correspond to various criterias will be set. These smaller objectives will be much clearer and less vague than the original aim of finding whether clubs are undervaluing or overvaluing the players and hence will intrinsically make achieving this original aim a more feasible task.

Aims/criteria that can be set as objects for aiding investigations into whether clubs are underpaying or overpaying for transfer fees using the binary predictor results

One way is to look at the accuracy of the predictors, as the models created are using attributes that are overwhelmingly skill/match based:

- A high accuracy from a predictor would suggest that the clubs are neither underpaying or overpaying but the players are valued at a justifiable price as the accuracy of the prediction is equalling the actual transfer market value set by the respective clubs

- A low accuracy from a predictor would suggest that the clubs are either underpaying or overpaying for the transfer fees

This is where we can explore the true positive, true negative, false positive and false negative rates for further insight:

- A true positive is an outcome where the model correctly predicts that the player is worth over a particular value, this quantity suggests no overpayment/underpayment

- A true negative is an outcome where the model correctly predicts that the player is worth less than a particular value, this quantity again suggests no overpayment/underpayment

- A false positive is an outcome where the model incorrectly predicts that the player is worth over a particular value, this quantity suggests the clubs are undervaluing/underpaying for the particular transfer

- A false negative is an outcome where the model incorrectly predicts that the player is worth less than a particular value, this quantity suggests the clubs are overvaluing/overpaying for the particular transfer

Note that the reasoning for picking a certain value and the chosen value itself will be mentioned further along in the study in the approach/methodology section.

Chapter 2: Literature review

With the ever increasing average value of football player's transfer market fee constantly on the rise, the question of whether these transfer market fees are justifiable or not always seems to come to the forefront of discussions. In hope of answering this question, the decision was made to explore the idea of predicting the range or ranges that the individual transfer market values fall into using both unsupervised and supervised machine learning methods whilst incorporating real life match statistics for individual players as well as the skill attribute values from the popular video game FIFA 22. The difference between supervised and unsupervised methods is that unsupervised methods do not have labelled inputs or outputs. Hence, we already know what the outcome of the supervised method should be, so supervised methods should be used to predict outcomes from a dataset but we do not know the outcome for unsupervised methods and hence they should be used to explore datasets rather than prediction.

2.1 Linear vs non linear supervised methods

A limited number of past studies have taken place to estimate the value of professional athletes from various sports. From these studies even fewer have used video game data to provide or enhance this prediction. Al-Asadi and Tasdemir (2022) solely used FIFA video game data for their predictions and the models consisted of some linear methods such as singular and multiple regression as well as some non linear methods such as decision trees and random forests. This study also found the non linear methods to perform the best in predicting transfer market value, hinting that the relationship between these attribute skill values and transfer value to be non linear. So to build on this study a plan was made to use other non linear supervised methods such as neural networks, random forests, support vector machines and k nearest neighbours to increase accuracy to find the optimal model.

Al-Asadi and Tasdemir (2022) were predicting a specific transfer market value rather than predicting if the value would be above or below a certain threshold. Hence the studies' research question was set as a regression problem rather than a classification problem. Hence the supervised machine learning methods/classifiers used were linear regression, multiple linear regression, regression tree and random forest regression.

The 4 supervised methods mentioned above all had a global purpose of predicting the transfer market value of the specific footballers in the dataset. The predictions made were all values with a certain amount of error and correlation with respect to the true value in the training dataset.

Linear regression is a supervised method that works by finding the relationship between a dependent target variable and a number of independent variables often called features. The most common formula for linear regression being in the form of y=mx+c. The difference between the simple and multiple regression is that simple linear regression contains a single feature for predicting the target variable whereas for multiple regression a handful of features are used. However, in this study the linear regression refers to a baseline model with a minimally accepted baseline correlation value and the multiple linear regression refers to the

linear regression models that performed better than this baseline model and had the highest possible correlation value.

Decision trees are a type of supervised learning method where each node represents a classification or regression value on a feature/variable and each branched path represents the accumulation of feature classifications/values that lead to the specific class label or prediction value. The basic classification decision tree algorithm works as follows. Firstly, select the attribute that is to be predicted and then pick a cut off point to reduce the classification error. The final step is for each new subregion to repeat the previous step until either almost all data points are classified as the same, no remaining feature is available or the tree size limit has been hit. Basic decision trees tend to be prone to overfitting, become complex with increase in number of features and therefore can be suboptimal. Pruning the tree to obtain a subtree will tend to fix these issues, this subtree is normally selected by minimising the cross validation error, (EDUCBA, 2022).

Random forests are a type of supervised learning method where the decision trees are improved with the use of ensemble/group based methods by combining a group of weaker learners, i.e a random forest consists of a combination of many decision trees working in unison. Hence, a handful of ensemble methods will be given as examples and then will subsequently be explained. The first method is the bagging process which is where the different trees are trained on bootstrapped subsamples. The second method is the random forest process itself, this works by the different decision trees being trained on bootstrapped subsamples using a random subset of attributes at each node split. The final method is the boosting process where an ensemble of complementary decision trees are built. Bagging and random forest are similar due to the fact that they are both algorithms that are chosen to reduce the complexity of models that overfit to the training data. However, the boosting process is an algorithm that increases the complexity of the model that has bias in terms of underfitting the training data, (EDUCBA, 2022).

Returning to the topic of this study, it focused solely on the use of the 2020 rendition of the FIFA dataset, which had roughly 18,000 entries with each entry representing a unique player and each player having over 70 attributes/variables. However, the study only used 9 variables in its actual prediction. The data was sorted into the standard 70/30 training and testing split and then the supervised methods mentioned above were applied.

The linear regression had a root mean square error (RMSE) of 5.46 million and r squared value of 0.47 which shows us that the predictions were not very accurate, the multiple linear regression performed slightly better with a RMSE of 4.66 million and a r squared value of 0.61, the regression tree greatly improved upon the previous method by displaying a RMSE OF 2.71 million and a r squared value of 0.87 and the final method of random forest regression performed the best with a RMSE of 1.64 million and a r squared value of 0.95. All the results mentioned above have been presented in an orderly table and can be found below the ensuing paragraph.

The decision tree and random forest methods perhaps performed better than the standard regression models as Varghese (2018) explains that decision trees support non linearity whereas linear regression models do not and also decision trees tend to identify collinearity more accurately than linear regression models do. For the technical terms mentioned above the root mean square error (RMSE) is the measure of how spread out that the prediction was compared to the true value. The r squared value is the measure that shows the amount of variance in the prediction that is explained by the variables in the dataset. The accuracy is simply the fraction or percentage of the correctly made predictions by the model. The training and testing split is the procedure where the dataset is randomly partitioned, usually one partition is either used to train the model and the other to test the accuracy of the model.

N	Classifier	MAE	RMSE	R^2
1	Linear Regression (Baseline)	5,468,144	5,468,144	0.47
2	Multiple Linear Regression	2,618,108	4,662,630	0.61
3	Regression Tree	835,935	2,713,452	0.87
4	Random Forest Regression	576,874	1,649,921	0.95

Figure 1: performance evaluators for Al-Asadi and Tasdemir (2022)

2.2 Attributes outside of match performance/skill that affect salary/transfer market value

Yaldo and Shamir (2017) is a study that used just FIFA video game data, this study suggested that player value was not merely related to player attributes but the merchandise sales and sponsorship/endorsement deals of the best performing players drastically affected the salary/transfer market value especially for footballers that exceeded the upper quartile of the salary/transfer market value range. To test this hypothesis real match statistics will be incorporated to see if players with the best game performances have a significantly higher transfer market fee.

Yaldo and Shamir (2017) is a similar study to the first one mentioned above. However, this study used the dataset for the 2016 instalment of the FIFA franchise, which has been attained by using a web scraper. This dataset includes roughly 6100 players represented by each entry and 40 attributes/variables were used to predict the weekly wage. The machine learning methods used for the prediction were as follows, additive regression, decision table, k nearest neighbour, a locally weighted learner, random committee, random tree and random subspace. The data was sorted into a 84/16 training and testing split. The performance evaluator for this study was the Pearson correlation coefficient between the predicted and actual values. The random committee performed the best with a correlation of 0.87, followed

jointly by the k nearest neighbour and random subspace with correlations of 0.84, additive regression with a correlation of 0.81, followed jointly by decision table and locally weighted learner with correlations of 0.78 and finally the random tree performing the weakest with a correlation of 0.53.

As all variables used for this prediction were ones that described a players physical attribute even the highest precision models underestimated the highest paid footballers salaries. For example the additive regression model predicted Lionel Messi to make roughly 235,000 euros when in reality he was earning 550,000 euros. Bank of England (2019) suggests that external factors outside of skill such as the globalisation of football and rapid advancements in technology has made football much more accessible and hence more popular around the world and clubs are making much more money from viewership and tv rights which drives the player wages tremendously high. This wage also does not include the individual sponsorship deals that many elite footballers hold with boot brands such as nike or adidas, social media ads and image rights. However these individual sponsorships do play some sort of role in determining a players weekly wage as all these deals make the specific players more marketable and would in turn generate more revenue for the clubs to distribute.

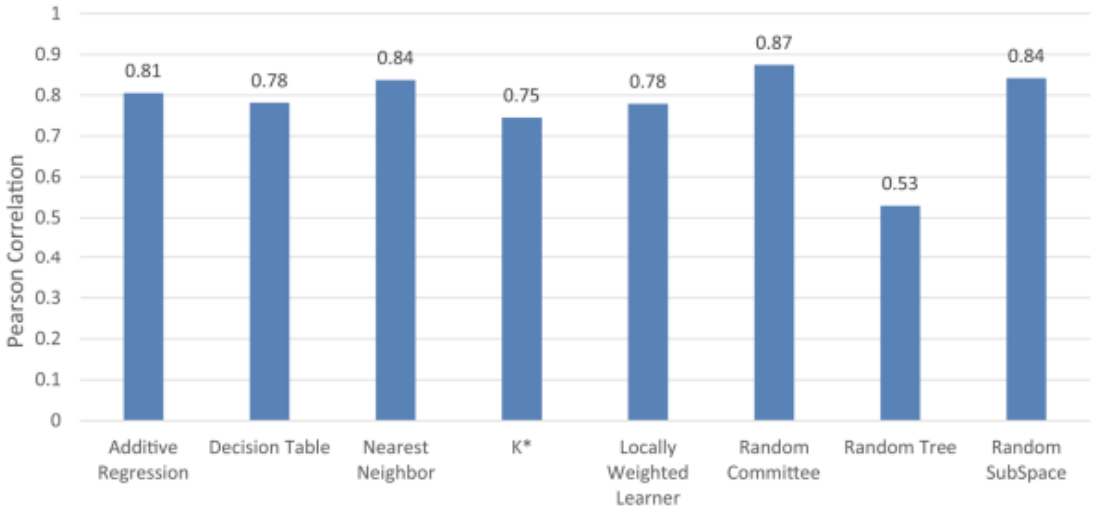

Figure 2: bar chart of performance evaluators for Yaldo and Shamir (2017)

A personal hypothesis was made that a player's popularity and social media presence would play a role in how high the transfer market value is set. Mahadevan (2020) is a study that used a combination of real match statistics with measures of popularity such as the number of web searches and youtube videos created for a player, this study confirms the hypothesis made above that the variables that measured popularity to be better indicators of value rather than skill and game statistics.

2.3 Fans holding greater influence in a player's value rather than match performance

Mahadevan (2020) was a unique study compared to others as it used real match statistics with a blend of popularity measures to predict the players' transfer market values. The dataset used was one that had roughly 4300 entries from 150 unique teams across the European top flight competitions. The dataset was created by the author of the study and this was achieved by collating multiple sources of data from 5 consecutive football seasons. However, this dataset was unfortunately not made readily available to the public. This study also found the best indicators to predict the transfer market value to be player characteristics, goal scored, fatigue which causes players' fitness levels to drop, skill, player and team strategy, position of the player, player match performance and finally the popularity of the player. For the model that was created and used in this study 14 unique attributes/variables were used for the transfer market value prediction. In comparison to the two previously mentioned studies, this study only used a single supervised machine learning method in its prediction rather than creating multiple models and picking the best.

The supervised method that was used was a regression analysis model (ordinary least squares regression). Ordinary least squares regression is a method that is often used for prediction, it differs from the usual regression method by minimising the sum of squared errors from the model. This model had a root mean square error (RMSE) of roughly 6 million which made it a substandard predictor of transfer market value. The study compares this model to another which was based solely upon football fans opinion of what each players' transfer market value should be and this new model was found to have a RMSE of roughly 5.8 million, which suggest fans hold greater power in setting each players' value rather than the mix of real game statistics and the handful of popularity measures that were used in this study. Meneses Flores (2017) was a study that arrived at a similar conclusion stating that models that used performance variables alone were not strong as models that included variables that captured media and social media influence of football fans, with the former model having an adjusted r squared value of 0.23 and the latter having an adjusted r squared value of 0.5. The same study also found that football transfer rumours or mentions from the press and media did not increase the players' values as much when compared to clubs mentioning players from other clubs on social media. An example of this happening is when a club confirms they have interest in signing a new player on social media which could prove to be detrimental as they would have to pay a higher fee to get the deal over the line.

SL. NO.	Model Evaluation		
		RMSE	***MAE***
1.	Trasnfermarkt's prediction	5,800	3100,820
2.	Model's prediction	5,992	329,743
3.	Relative diffrenece	+4.5%	+4.1%

Figure 3: performance evaluators for Mahadevan (2020)

2.4 Relationship between player performance and salary in the NBA is non linear

Due to the limited studies regarding this topic a decision was made to explore studies that predict transfer/salary value in other sports. Papadaki and Tsagris (2020) was one of these studies used for NBA basketball players. The study only constructed a random forest classifier as it found other methods such as KNN to be substandard in performance but the study in hindsight suggested that the performance could be improved by using variable transformation and reduction such as principal component analysis, which is what will be used during the preprocessing and data cleaning/analysis stage of this study to possibly boost the performance of the models that are to be trained.

As mentioned above, Papadaki and Tsagris (2020) was a study which focused on estimating professional basketball players salaries rather than the transfer market value. The study prioritised on fitting a nonlinear predictor to estimate the player wage rather than creating the usual linear model. The study majorly focused on attributes/variables that measured the players' ability on the court. This study used a combination of 4 datasets showing statistics per game, per 36 minutes, per 100 possessions and advanced data. These 4 datasets were acquired for the 3 seasons of 2016-2017, 2017-2018 and 2018-2019, making a grand total of 12 datasets. There were roughly 1600 entries across all datasets and 54 attributes/variables describing performance metrics and other important details. This study also used a single supervised learning method in the form of a nonlinear random forest predictor. The Pearson correlation coefficient between the observed salary and predicted salary was used to evaluate the performance of the various models. The top 3 performing models were the per game dataset from 2017-2018 with a correlation of 0.798, the per 36 minutes dataset from 2017-2018 and the per 100 possessions from 2017-2018 both respectively having correlations of 0.801. This study concludes by confirming that indeed the relationship between the performance variables and player salary is nonlinear.

The study, like the previous ones mentioned above, suggests looking at factors such as player popularity and the spectacle displayed on the court to improve the accuracy in predicting the players' salaries. Furthermore if we decide to strictly look at performance statistics, Lyons et al. (2015) suggests that points per game and field goal percentage are the best indicators for predicting player salaries. Hence suggesting that players that can shoot effectively and have exceptionally efficient conversion rates are more likely to have higher salaries.

Dataset	2016-2017	2017-2018	2018-2019
Per game	0.789	0.798	0.772
Per 36 minutes	0.781	0.801	0.759
Per 100 possessions	0.784	0.801	0.759
Advanced statistics	0.769	0.741	0.716

Figure 4: performance evaluators for Papadaki and Tsagris (2020)

2.5 External factors play an insignificant role in determining MLB players' salaries

Lee et al. (2021) was a case exploring salary/transfer market values of MLB baseball players. This study found a variable that assessed the impact of winning or losing a player to be the most important in positively influencing the transfer market value suggesting that on field performance was the main contributing factor regarding a player's market value increasing. The same study also found a variable describing whether a player will be out of contract or not at end of the season to be the most important in negatively influencing the salary/transfer market value suggesting that if a player is no longer contracted to a team the player's transfer market value will drastically drop which suggests no matter how good a player performed the time remaining on the contract is proportional to the players market value this is also evident in football as many world class players leave on a free transfer due to running down a contract and refusing to sign a new deal.

Unlike the previously mentioned studies, baseball seems to differ from sports like football and basketball as Wasserman (2013) proposes that external factors such as social media and fan influence plays a much smaller role in determining a baseball player's salary. This study comes to the conclusion that by far the most important factor determining how much a player is paid is the quality of on field performances.

Chapter 3: Approach/ methodology

From the studies that have been previously analysed, the best performing supervised learning methods seem to be the random forest predictor, k nearest neighbours and decision trees. We can see this as Al-Asadi and Tasdemir (2022) reported that the 2 best performing models to be the random forest and the decision tree respectively. They had the lowest mean absolute error and root mean square error values as well as the highest r squared values. For Yaldo and Shamir (2017), the k nearest neighbours was the second best performing model out of a total of 8 with a high Pearson correlation coefficient. Furthermore, Papadaki and Tsagris (2020) only trained a random forest which also achieved a high Pearson correlation coefficient. Finally, Lee et al. (2021) again only trained a single gradient boosted decision tree model which was found to have a high accuracy value for the predictions that were made.

The data used for this study was from a closed source website, kaggle and did not involve the use of human participants and security sensitive information. Hence this study was confirmed to not require research ethics approval by the research ethics committee. The ethical approval letter can be found in the appendix section of this study.

The first supervised method that will be used for the prediction is neural networks. The advantages of this method are that it is able to model complex patterns in datasets, it can be used for both purposes of classification and regression, no assumptions about the variables in the dataset are made. The disadvantages are that the process that takes place can not be easily explained or interpreted, it is prone to overfitting towards the training data and as Baeldung (2022) also mentions that neural networks require a significant amount of computational power to train the model on datasets.

The second supervised method that will be used is random forests. The advantages of this model are that it is efficient at processing high dimensional data which is very useful for the dataset that will be used in this study, desired and important features can be directly selected to be used in the model, works for numerical and nominal features and this method is insensitive to noise within the dataset. The disadvantages are that the model is not easy to explain and it can be difficult to fine tune. Another reason to use this method over decision trees as Section (2020) explains that random forests are inherently more precise than decision trees.

The third supervised method that will be used is support vector machines (SVM). Gandhi (2018) explains that SVMs work by using hyperplanes to aid in classifying data points that fall on either side of said hyperplane. A hyperplane, in the context of SVMS, is a boundary that helps the algorithm differentiate and hence correctly identify and class data points during the training process. The advantages are that it is highly accurate in prediction, the model is compact, it has immunity towards overfitting the training data, insensitive to noise and also works for both classification and regression. The disadvantages are that it is tricky to choose the correct kernel function which controls the shape and direction of the hyperplane, it is slow to train on large datasets which could be a potential problem in this study, the model is not easily explainable, this model only works for numerical features although one hot encoding can be used on categorical features to get around this complication.

Chapter 3: Approach/ methodology

The final supervised method that will be used is K nearest neighbours (KNN). Harrison (2018) simplifies the explanation of how KNN works by stating that the algorithm makes the assumption that things that are close in distance are similar to one another. The advantages of using KNN are that it is a fairly simple model, it is easy to explain, very fast to train, no assumptions are made on the variables within the dataset and that it works for both classification and regression. The disadvantages are that the algorithm doesn't try to explain the relationship between inputs and outputs as there is no model created, it is tricky to select a value for k, this method tends to be slower for classification problems and once again this method only traditionally works with numerical variables but one hot encoding can be used to remediate this.

The first limitation to this study is that binary classification will be used over multiclass classification. The reason for using binary classification is that it is less computationally expensive than multiclass classification, also the fact that multiclass classification tends to be harder to train and tweak and produces significantly less accurate predictions than their binary counterparts. Using binary classifications, the details in the method of training must be discussed. For binary classification, the model must be configured to target a categorical variable with 2 levels. For categorical variables, levels are best described as the different groups or classifications that the variable can fall into and take. The individual levels are often labelled/encoded according to which group the variable belongs to. The approach this algorithm takes to make its predictions is by assigning the 2 separate levels with a binary pairing label which is called the one vs one approach.

For binary classification the numerical target variable, release clause, had to be transformed into categorical variables with 2 levels. For this transformation a threshold of 1.6 million euros was chosen as it was found to be the median for the release clause variable, this ensures that for training and testing procedure there would be an equal amount of each binary label throughout the dataset. This balancing of classes removes the possibility of creating bias during the training procedure and aids with creating more meaningful and relevant models. R studio was chosen to carry out binary classification as it had a wider selection of binary classifiers compared to PySpark, an example was that non linear kernel functions for support vector machines were not available on Apache Spark as it is difficult to distribute. The binary classifiers used were decision trees, random forests, neural networks, support vector machines and K nearest neighbours.

To meet the aims and criteria mentioned in the introduction, for each binary classification method a results table will be created with the first column showing the true response value from the training dataset. The response variable indicates whether or not the transfer market for each player was either above or below the 1.6 million euros threshold. Also, the second column will show the predictions created by the various binary classifiers that were described above and are yet to be created and deployed. From these values, a count will be initiated that will essentially tally/accumulate the matching and non matching predictions to the true corresponding response values from the previously created results table.
From these tallied figures, the true positive, false positive, false negative and the true negative rates will be summed up and presented in an accumulation table. Then from these various summed counts we can then infer the number of players that are undervalued,

overvalued and valued fairly according to the criteria and assumptions that were made and mentioned in the introduction section regarding the performance of the binary classifiers.

Chapter 4: Data cleaning and Exploratory data analysis

4.1 Data cleaning for FIFA dataset

All the additional files describing the data analysis and machine learning methods can be found from the following link:
https://github.com/ManrajR1998/Dissertation_files
The FIFA dataset was obtained from the Kaggle website, (Kaggle, 2022). The data cleaning for the FIFA dataset was carried out using the R studio programme. Initially, the dataset contained roughly 19,000 separate entries, each representing a unique football player who plays at the professional level. The dataset originally had 110 variables that each constituted a unique descriptor of the players' footballing background such as attributes, nationality, position, finances and so forth. However, the dataset was pruned leaving a total of 69 variables which initially consisted of 57 numeric variables, 0 categorical variables and 12 character/string variables. The lack of categorical variables was due to R studio not correctly identifying them. With the help of the as.factor() command all misidentified categorical variables were manually transformed, leaving a dataset consisting of 50 numeric variables, 17 categorical variables and 2 character/string variables. More specifically, a distinctive selection and combination of these variables will be used to predict the players' transfer wages according to each model created as well as depending on the supervised method used. As mentioned in the previous section, an extra categorical variable was made from the numeric variable 'release_clause_eur'. It was named 'release_threshold' and was created for a purpose of being a target categorical variable with 2 levels which will be used for binary classification.

4.2 Exploratory data analysis for FIFA dataset

The exploratory data analysis for the FIFA dataset was also carried out using the R studio programme. For each variable in the dataset a plot was produced, more precisely a histogram was produced for the numerical variables and a bar chart for the categorical variables. Due to the fact that there are over 60 variables in this dataset, I will not describe the distributions and patterns of each attribute but rather the variables that produce plots that are distinctive or stand out. The two histograms that appeared prominently were player value and player wage, both having heavy positive skews which suggests a small group of players are paid and valued at much higher rate than the vast majority, as seen in figure 6. Player frequency refers to the number of players that have specific values or fall into specific categories for the variables described in the individual graphs below.

Chapter 4: Data cleaning and Exploratory data analysis

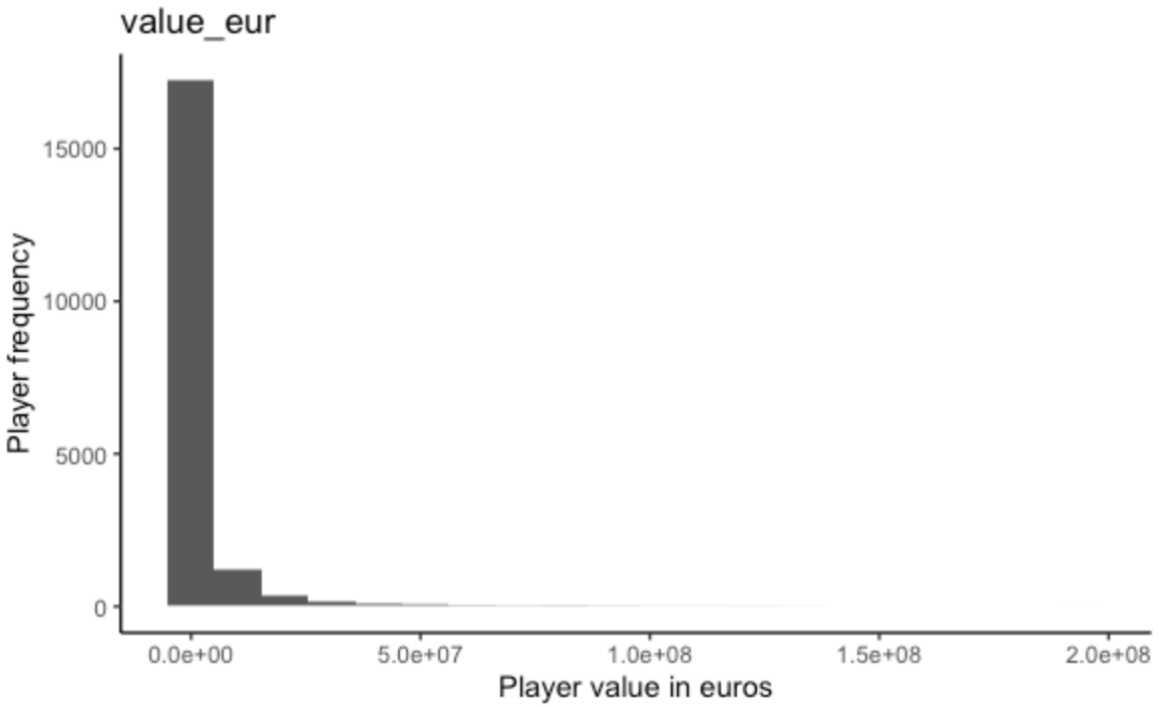

Figure 6: histogram showing the positively skewed distribution for player value

Player value is not to be confused with transfer value/release clause, as they both differ hugely. Player value is what the market thinks the player is roughly valued at. Transfer value/release clause is the valuation that the clubs give to players that are currently employed under them and this value is often stipulated in the individual's contract. So for the vast majority of transfers, the value that the players are sold at is equal to transfer value/release clause that was set by the selling club. The bar chart for player work rate also stood out, with the medium/medium work rate vastly overshadowing all other options, as seen in figure 7. This suggests that the majority of players cover medium distances over the game during both attack and defensive phases, which possibly indicates that the elite players cover more distance over a game in either attack, defence or even both. This also suggests higher work rates could be correlated to larger transfer fee sums. For the work rates shown in the bar chart they are formatted in a way which mentions the attacking work rate first followed by the defensive work rate, i.e Attacking work rate/Defensive work rate. Both the attacking and defensive work rates have 3 levels each. These levels are high, medium and low. For instance a High/High work rate suggests the player puts maximum effort into distance covered for attack and defence.

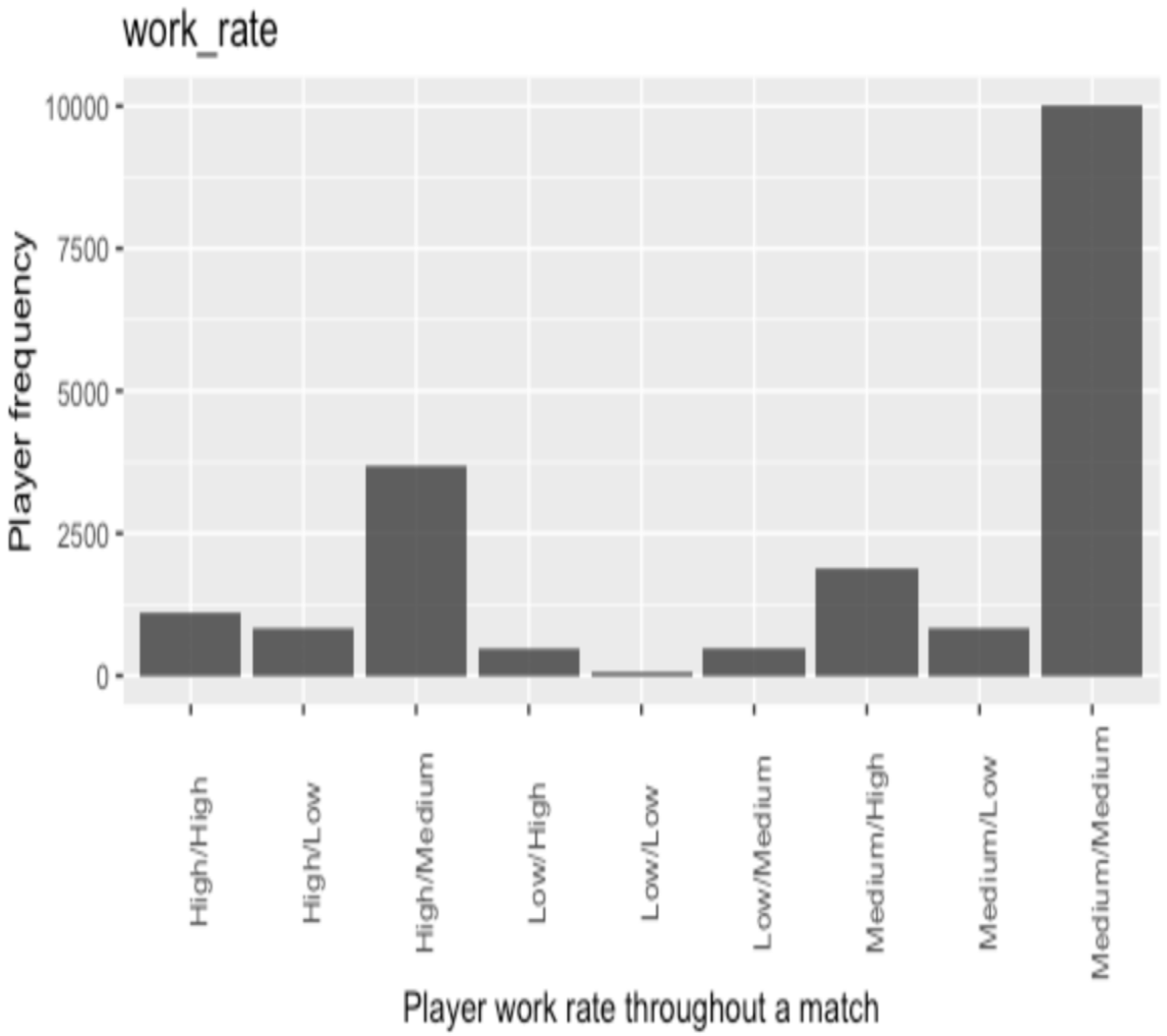

Figure 7: bar chart showing the various distributions of the individual player work rates

As the aim of this study was to predict transfer market value, plots were produced against 'release_clause' as the target variable. For numerical variables a box plot was produced and for categorical variables a mosaic plot was produced. The box plot for potential against transfer value shows players that were worth between 200 million and 300 million euros showed the highest potential, shown in figure 8. This trend continued linearly for all transfer value ranges suggesting that indeed players who have the highest potential are valued accordingly with the highest transfer values/release clauses. Player potential is a variable that can be seen as the overall rating a player is given out of 100 according to the individual's skill seen on the field and the potential rating is an extrapolation of the highest level that the player may reach in terms of performance in the future.

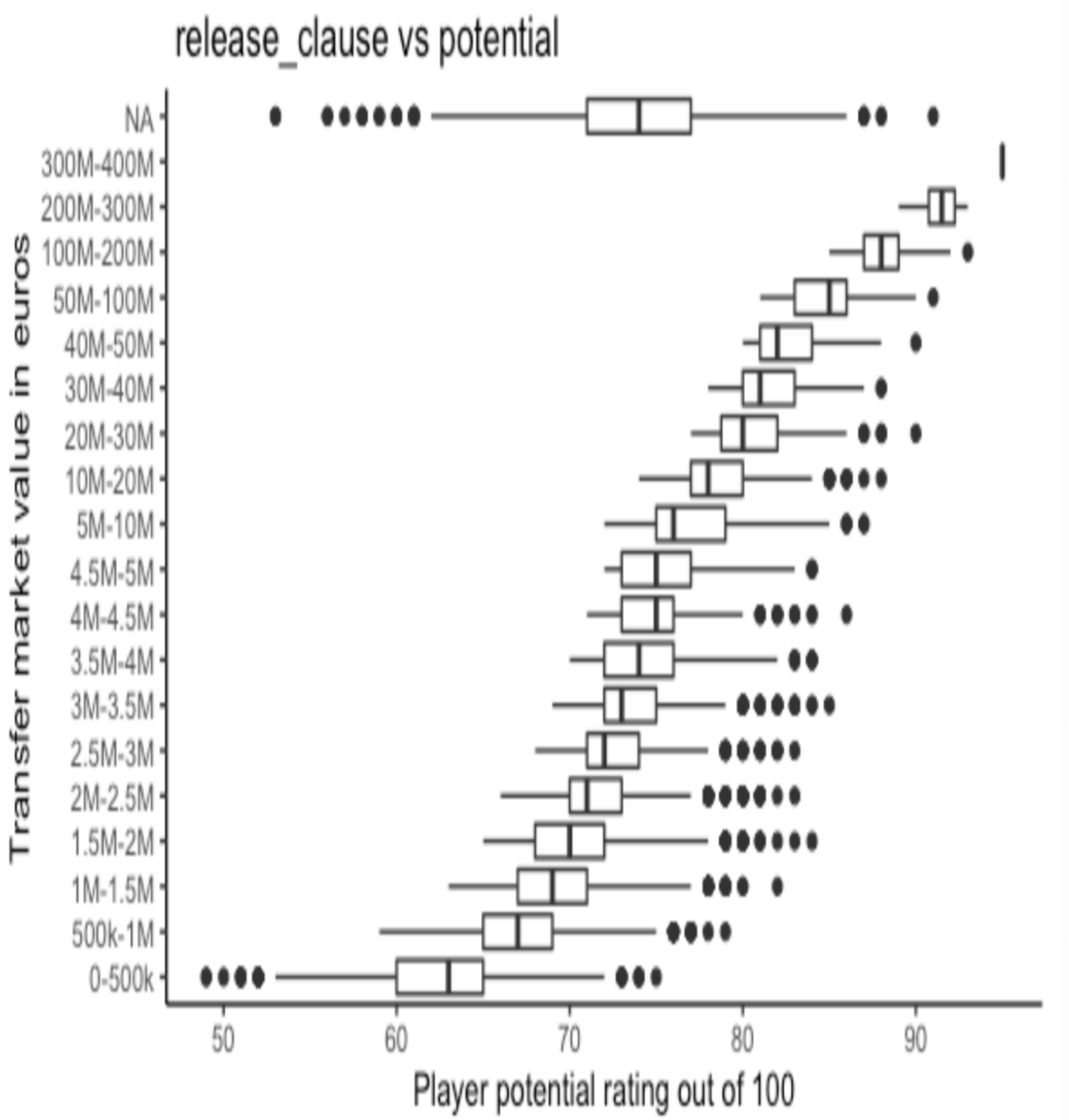

Figure 8: boxplot showing the linear relationship between transfer market value and player potential

The mosaic plot for preferred foot against transfer value shows that right footed players were the overwhelming majority over all transfer value ranges and this ratio increases and is even more prominent in the higher ranges suggesting that players with the highest transfer values are almost exclusively right footed, as seen in figure 9.

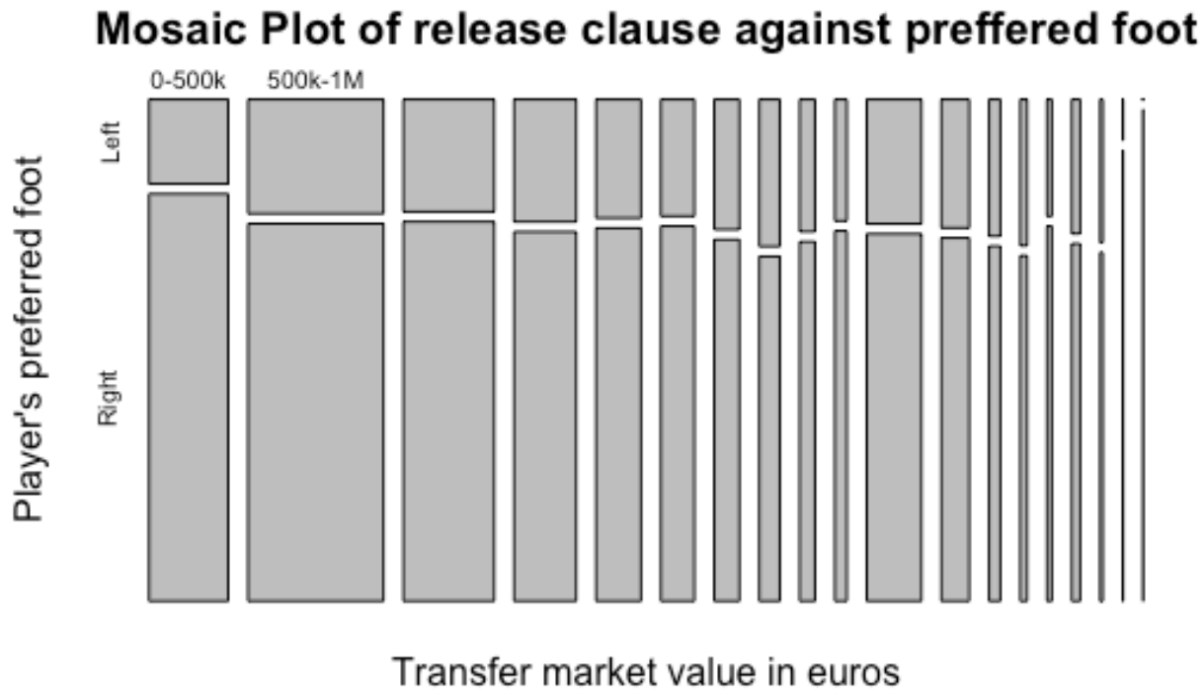

Figure 9: mosaic plot showing the relationship between transfer market value and preferred foot

4.3 Principal component analysis for FIFA dataset

Following this initial cleaning and data analysis of the FIFA dataset a general principal component analysis was conducted, which gives the advantage of reducing the dimensionality and increasing the interoperability of the dataset whilst minimising the data lost to a bare minimum. The principal components from this analysis may be used when data processing takes too long for a particular supervised method. Initially, the target variable and all non numeric attributes such as character/string and categorical were removed. Also all NA's were removed using the na.omit() command. The removal of these non numeric variables is down to the fact that principal component analysis is a mathematical process which works by creating new variables that maximise the variance of the original data, hence only numerical variables can be used for this process. The proportion of explained variance (PEV) was then calculated. The principal components were then created using the prcomp() and then the summary function was run to inspect the cumulative proportion of variance found up to each component variable. After testing a few values and wanting to keep data processing times relatively low, 80% variance was chosen as an acceptable threshold for all principal components to possess.

Using this threshold, the plot of cumulative PEV against the principal components indicated that roughly 8 principal components captured 80% of the variance and hence were suitable enough if required for a prediction of the transfer market value. The number of principal components can be read from figure 10 by counting the number of orange dots below the red dotted line. Following this initial creation of principal components and a second principal component analysis was required for the neural network predictions of transfer market value as data processing time was a problem and the prediction would be impossible without this process taking place.

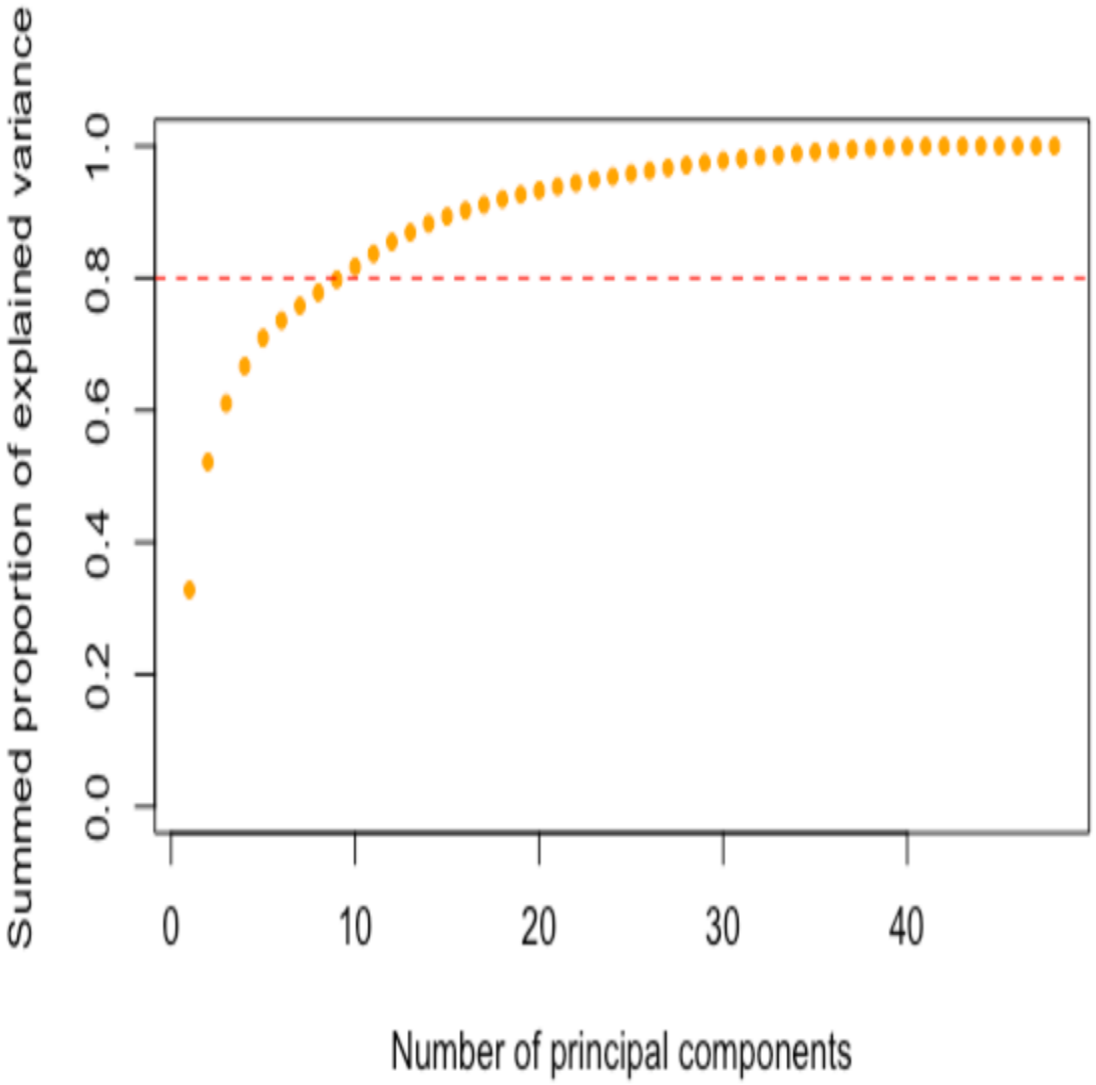

Figure 10: General plot showing the number of principal components required for 80% captured variance

The need for a completely different principal component analysis was down to the fact that neural networks require numeric variables for data processing. Hence, the categorical variables had to be encoded using the one hot encoding method which employs the use of dummy variables from the 'caret' package. Also a minmax normalisation function was applied to the whole dataset after the encoding, which aids the neural network during the training process by streamlining the data which helps it to converge. Finally, the steps of the process explained in the previous principal component analysis were reapplied. The new cumulative PEV plot suggested that 58 principal components captured 80% of variance. Hence, a new dataset was created that would solely be used in the neural network prediction which consisted of the 58 principal components with the inclusion of the binary target variable. For figure 11, the number of orange dots below the red dotted line were not visibility countable. Hence a manual inspection of the principal components had to be conducted to find the threshold that captured 80% of the variance.

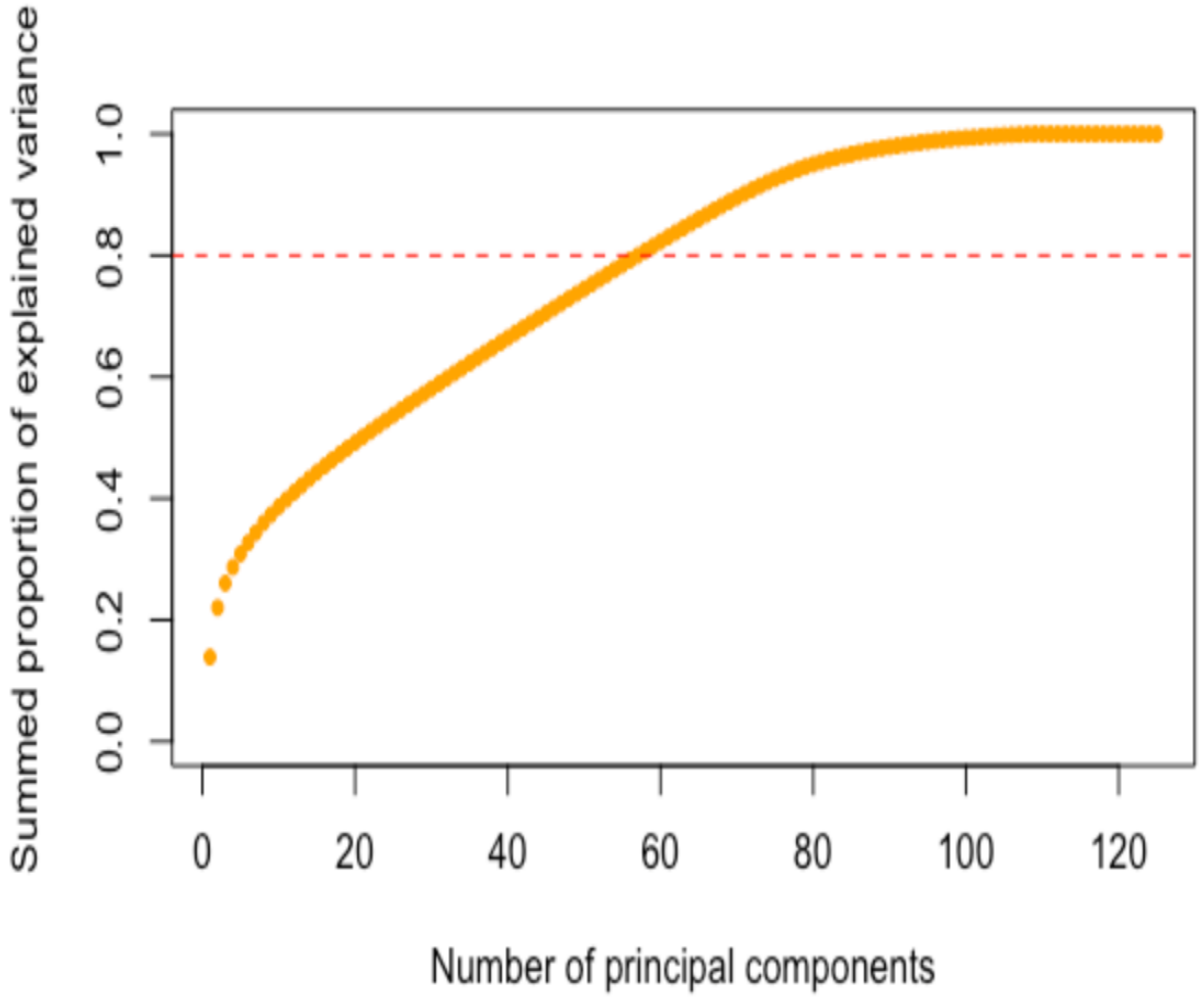

Figure 11: Plot showing the number of principal components required for 80% captured variance in the neural network dataset

4.4 Creation, cleaning and exploratory data analysis of real match dataset

The real life player data was scraped from a combination of various online sources using a freely available web scraper tool plugin for google chrome. The data cleaning and exploratory analysis of this dataset was again done in R studio. The dataset initially had 250 entries, each representing a unique football player with 14 variables that each described a real match statistic. However, for this study only 9 of these variables were required. Out of these 9 variables, 6 were misclassified as string/character variables and hence were rectified using the as.numeric() command. As all variables explored in this dataset were numeric, histograms were produced for each attribute. The most interesting histograms produced were for the variables starting lineup percentage, which can be seen in figure 12, and minutes played percentage. Both variables were negatively skewed which suggests that most of the players in this dataset started matches regularly and played the majority of minutes available throughout the matches.

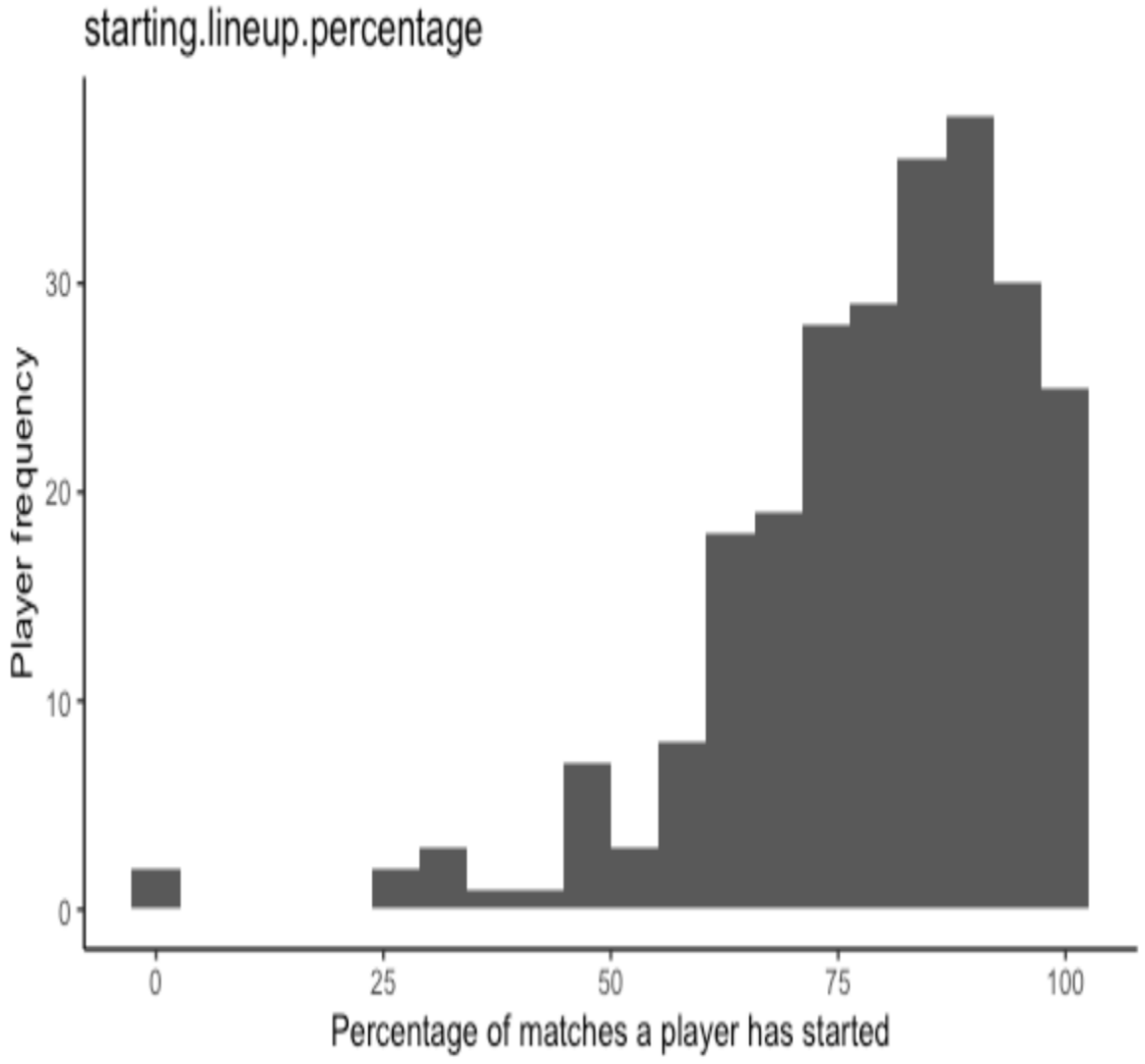

Figure 12: histogram showing the negatively skewed distribution for the starting lineup percentage rates

4.5 Cluster analysis and comparison

Cluster analysis for both the FIFA and real player datasets were also again done in R studio. As we are using clustering in terms of exploring the data and searching for any hidden structures or particular similarities between these 2 datasets, divisive hierarchical clustering was the obvious method of choice. This is due to the fact that we do not know how many clusters that the data can be partitioned into before applying the method. This works really well with divisive hierarchical clustering as this method begins by classifying all data points as one cluster and continuously divides until n distinct clusters are formed. The dendrogram of the real player data indicates 2 distinct groups, the number of optimal clusters/groups is determined by finding the longest vertical distance between any 2 nodes on the dendrogram and then counting the number of divisions that each branch has made up until this point. The smaller cluster from figure 13 had 8% of all data points.

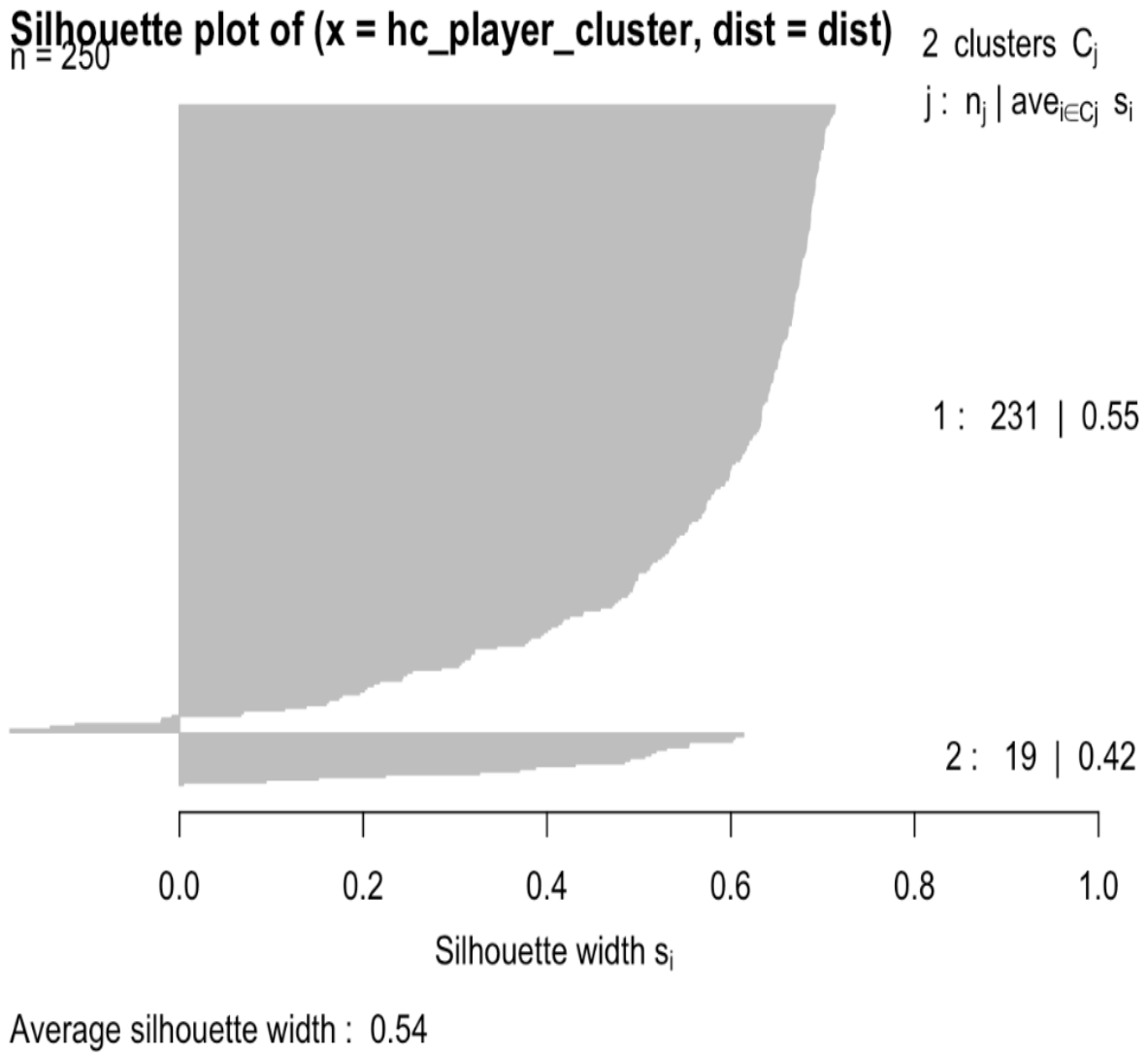

Figure 13: Silhouette plot for real match data showing 2 distinct clusters

The dendrogram of the FIFA data again indicates 2 distinct groups, however the smaller cluster for figure 14 was merely 0.4% of all data points, which is much smaller in comparison to the real life data. As we are comparing the same clustering method across 2 separate datasets which differ completely in terms of the shape and structure, we can only comment on the general aspects of both cluster analyses. So from these results we can perhaps make the claim that the cluster that contains 8% of all data points from the real life data is comparable to the summary results of the FIFA data, which indicates only 10% of players have a transfer market value that exceeds 10 million euros. Hence making the claim that for the real match data that 8% of players in the smaller cluster are much more likely to be worth over 10 million euros compared to 92% of players in the larger cluster.

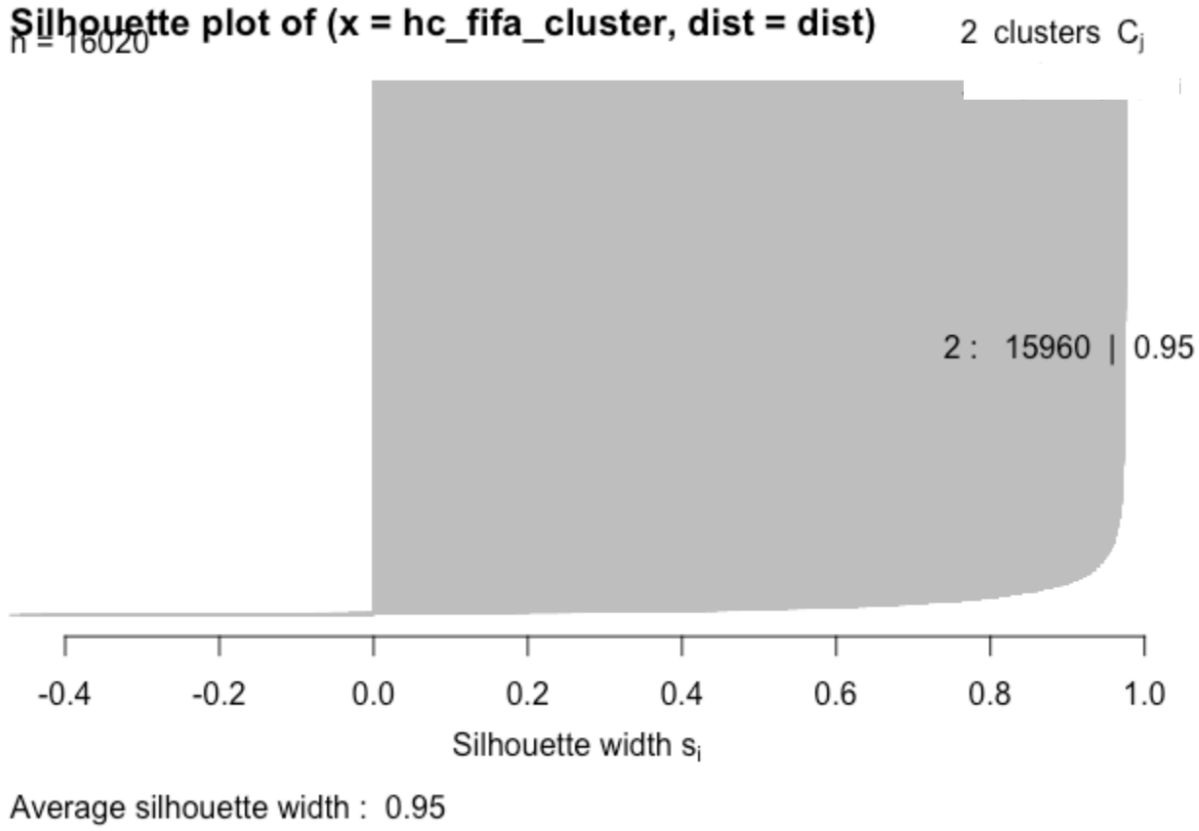

Figure 14: Silhouette plot for FIFA data showing 2 lesser distinct clusters compared to real match data

Note that the Si seen in both x-axis of figure 13 and 14 refer to the silhouette value, this value represents how similar points are to the clusters they belong to and how they differ to other clusters.

Chapter 5: Binary classification results: Decision tree and random forest

R studio was used for the creation, testing and training of the binary decision tree and random forest classifiers. Before any decision trees and random forest classifiers could be tested and applied, a new classification dataset had to be created from the original FIFA dataset. The changes that were made included the removal of redundant variables such as the numerical variable 'release_clause_eur' due to its categorical target variable counterpart for binary classification being available in the dataset. This prevented the possibility of this variable mistakenly being used in the testing stages of any of the supervised learning predictors. The new dataset also included the 8 principal components that were created during the initial principal component analysis which had the general purpose of being readily available for use during the training process if data processing time was a problem. Essentially the principal components for this dataset acted as a fail safe if any problems took place during the training of any of the supervised machine learning methods. As these specific decision trees and random forests were intended to be used for binary classification and data processing time was not yet a problem, the principal components were removed. The dataset then consisted of 57 variables with 52 numeric variables, 0 categorical variables and 5 string/character variables. The lack of categorical variables was again due to R studio misclassifying them incorrectly, once again the as.factor() command was used to mend these errors. The final dataset now consisted of 46 numeric variables, 11 categorical variables and 0 string/character variables.

The next step was to generate the training and testing split by creating a training index using the sample() function. The data split ratio was chosen to be 70% for training and 30% for testing. The set.seed() function was used when creating the training and testing dataset splits, this function is originally used to generate the same starting point when picking a set of random numbers. However, the use of this function when creating the training and testing data was to make sure that each time the code is run the same training and testing split is produced. This is done to make the results of this predictor consistent and hence the same results can be reproduced externally if the same seed is chosen.

The step prior to actually producing and tinkering with the decision tree and random forest predictors was to devise and create the formula that would aid the supervised learners in understanding what quantity or variable to predict. The formula was created using the reformulate() function which is a convenient tool, as the formula includes over 57 variables. Usually the formula for creating a supervised predictor is given as a sum of the independent variables equalling the target variable, however with the use of the reformulate() function only the target variable has to be specified by name and the numerical position of its column in the dataset.

The advantages of decision trees are that the procedure is visible and readable to humans, easily explained, the model is transparent and it processes both numerical and nominal features. The disadvantages of decision trees are that the procedure is inefficient for high dimensional data, tends to overfit to the training set and is sensitive to small noise/perturbations. The binary decision tree classifier was created using the 'tree' package. The tree() function was used in conjunction with the transfer market formula that was mentioned above and was then trained on the previously created partitioned training dataset.

Chapter 5: Binary classification results: Decision tree and random forest

The binary random forest classifier was created using the ‘randomForest’ package. Similarly to the decision tree classifier, the randomForest() function was once again used together with the transfer market formula and was then trained on the same training set. Also 500 was the specified number of decision trees that were found to be optimal in training this random forest predictor to attain the most accurate results.

The decision tree classifier was deployed to produce its predictions using the predict() function. Inside this function the decision tree classifier was specified. Also the test data was allocated to the function, the binary target variable was removed and the mode of prediction was set to classification. A results table was created with one column showing the actual response if whether or not the transfer value was over the threshold of 1.6 million euros from the test data and the second column showed the corresponding prediction of the decision tree classifier. From these results a cumulation table was created using the table() function which indicated the actual values and decision tree predictions. The 4 segments of the table were split between indicating the true positive, false positive, false negative and the true negative rates. For the decision tree classifier the counts were 2348, 74, 130 and 2254 respectively.

```
##        dt
## actual    0    1
##      0 2254   74
##      1  130 2348
```

Figure 15: Cumulative table for decision tree

From taking the sum of the true positive and true negative counts in the above table we can assume that the binary decision tree predicted a total of 4602 players were valued fairly by their respective clubs, which was roughly 95.8% of all players in the training dataset. By looking at the false positive rates we can assume 74 players were undervalued by their respective clubs, which was about 1.5% of all players available. Finally, by inspecting the false negative rates we can assume 130 players were overvalued by their respective clubs, which was about 2.7% of all players evaluated.

The final step was to calculate the accuracy of the classifier using the formula that sums the diagonals of the cumulative table divided by the total of the whole table, i.e the sum of the true positive and true negative divided by sum of the true positive, false positive, false negative and true negative. The accuracy for the decision tree classifier was found to be at a fairly high value of 0.9575. For the 46 numeric features and 10 categorical features that were used in this prediction of the transfer market value, all were highly weighted with each feature having equal importance when predicting the transfer market value.

The random forest classifier was deployed and set up the exact same way as explained for the decision tree classifier. For the random forest classifier the true positive counts were 2360, false positives were numbered at 96, false negatives were numbered at 118 and the true negatives were numbered at 2232.

```
##        rf
## actual    0    1
##      0 2232   96
##      1  118 2360
```

Figure 16: Cumulative table for random forest

From following the same steps from the analysis of the decision tree, we can assume that the binary random forest predicted a total of 4592 players were valued fairly by their respective clubs, which was roughly 95.5% of all players in the training dataset. By again looking at the false positive rates we can assume 96 players were undervalued, which was about 2% of all players available. Similarly, by inspecting the false negative rates we can assume 118 players were overvalued, which was about 2.5% of all players evaluated.

This gave the random forest the accuracy at a significantly high value of 0.9554, which is interestingly slightly less than the decision tree classifier. This is surprising as Selvaraj (2022) explains that random forests are typically said to be better learners than decision trees, however tinkering with the combination and number of decision trees used in the random forest it was not possible to beat the accuracy of the original decision tree classifier. Again similar to the decision tree classifier, the equal weighting across the 46 numeric features and 10 categorical features was critical to the high accuracy value of 0.9554 that was achieved at predicting the transfer market value.

Chapter 6: Binary classification results: Neural networks

R studio was once again used for the creation, testing and training of the binary neural network classifiers. As mentioned in the data cleaning and EDA section, a specially made dataset had to be created for the neural network classifiers as the data processing times were an issue during the training procedures due to the data not converging at the various nodes. The dataset included 58 principal components which had been created after the categorical and numerical variables had been encoded and normalised. This was due to the 58 principal components capturing 80% of the variance of the original dataset. This dataset was made purely of principal components and a single encoded categorical target variable. Hence the dataset consisted of 59 variables with 59 numeric variables, 0 categorical variables and 0 string/character variables. From these 59 numerical variables, 58 were continuous numerical variables and 1 was an integer variable, i.e the target variable. Similarly to the decision tree and random forest predictors, the training and testing split for this new dataset was again chosen to be 70% and 30% respectively using the sample() function to create a new training index. The set.seed() function was also used once again to make the training and testing split reproducible. Similarly, a new formula had to be created using the reformulate() function to instruct the neural networks on how to function to predict the correct target variable using the new PCA training dataset. The neural network formula, similar to the one created for the decision tree and random forest predictors, was the sum of the principal components equalling the binary target variable.

Neural networks are a special kind of machine learning technique that have been derived and inspired by the neural networks and pathways that send and receive signals in the human body, more particularly the brain. They are good at processing and understanding both linear and non linear relationships between different variables, they work by traversing a series of neurons that lead to a certain neuron output. The activation functions in a neural network are the set of functions that a neural network has at its disposal to best find the relationship between specific variables and they are used at nodes to transform the input to the output. Some examples of these activation functions are the linear function, gaussian function and the binary step function. Neural networks are best known for supervised learning, however there are special cases where they can be used for unsupervised learning. Self organising map (SOM) is a type of neural network that can be used for unsupervised learning. SOMs are used to create a 2 dimensional rendition/mapping of a high dimensional dataset whilst preserving the topological structure of the data, so they are a form of dimensionality reduction. Hence, SOMs are mappings of the m dimensions onto a 2 dimensional grid, where m represents the number of attributes/variables in a particular dataset.

For supervised learning neural networks work through a network that consists of different sections, the first section is an input layer i.e the place where the input variables are processed. The middle section of the network is a structure of numerous hidden layers where the data is transformed and streamed, the number of hidden layers and structure can be set by the user. The final section is the output layer, where the predictions of the target variable are sorted and displayed. This type of predictor is called an artificial multilayer neural network and is what will be used to create our 2 binary neural networks to predict the transfer market value. Back-propagation is a method used for training neural networks through the various layers, the process is essentially the data going through the network

layers back and forth until the data converges. Each time the data is sent back to a layer, a process called error propagation takes place where the neural network learns from the errors made in the earlier epochs and hence makes the predictor more resilient to mistakes that could be made in future epochs. Epochs are the number of cycles/times that the training dataset has been processed or read by the predictor. The base neural network with a single hidden layer that consists of 1 node was created using the 'neuralnet' package. The neuralnet() function was used together with the neural network formula and was trained on the newly created PCA training dataset.

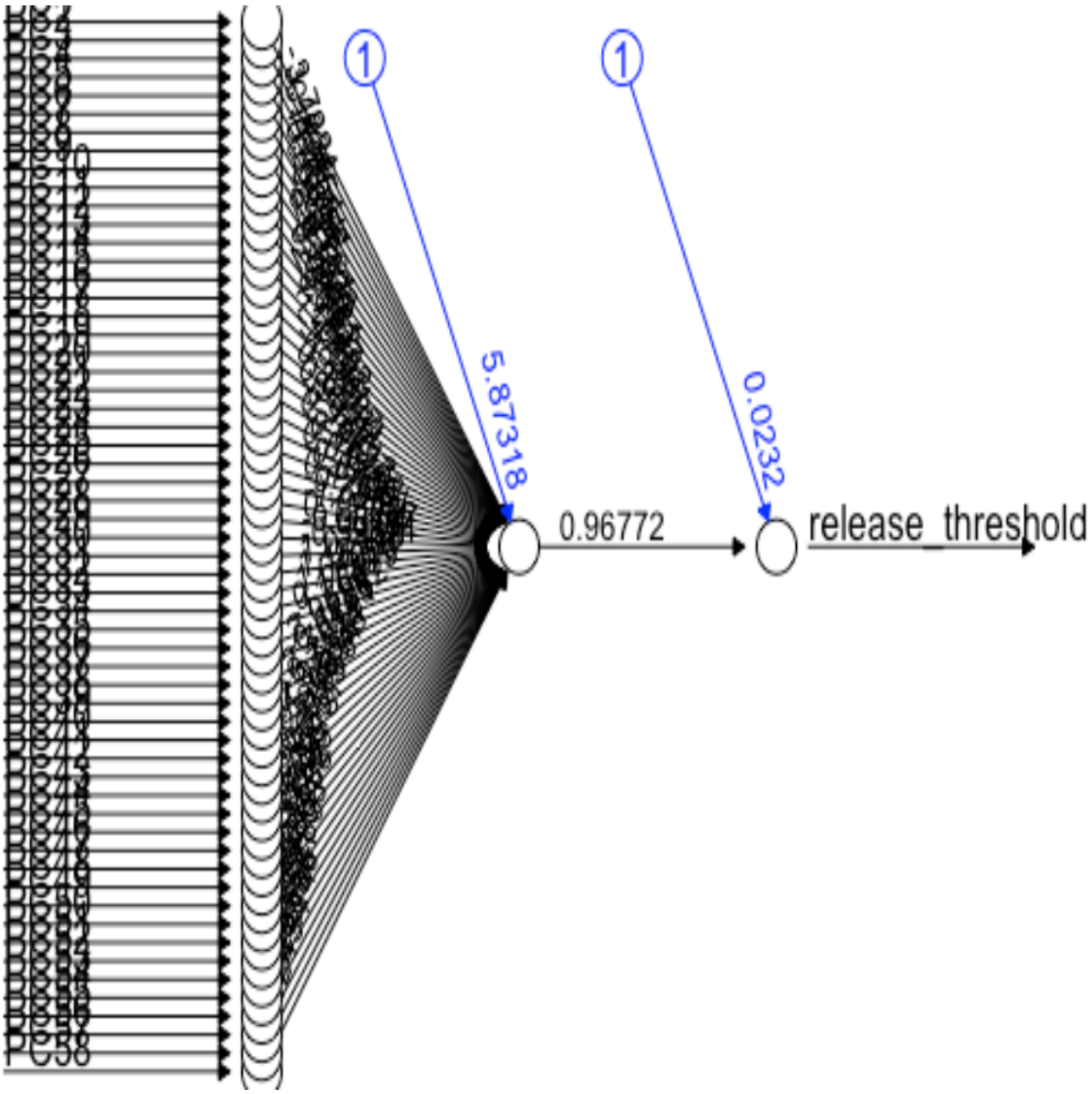

Figure 17: Base neural network plot

Chapter 6: Binary classification results: Neural networks

The second neural network that was trained was one that had a single hidden layer which consisted of 5 nodes. The number of neurons on a hidden layer was experimented and tinkered with to find the optimal training time to accuracy ratio and 5 neurons was found to be the most suitable to produce the highest amount of true positive and true negative counts. Although adding multiple hidden layers gives the advantage of the specific transformations being allocated to a specific layer and hence increases the sophistication of the neural network, a single hidden layer performed the best with the highest accuracy. The neural network with 5 nodes on a single hidden layer was again created using the neuralnet() function on the exact same neural network formula and training dataset.

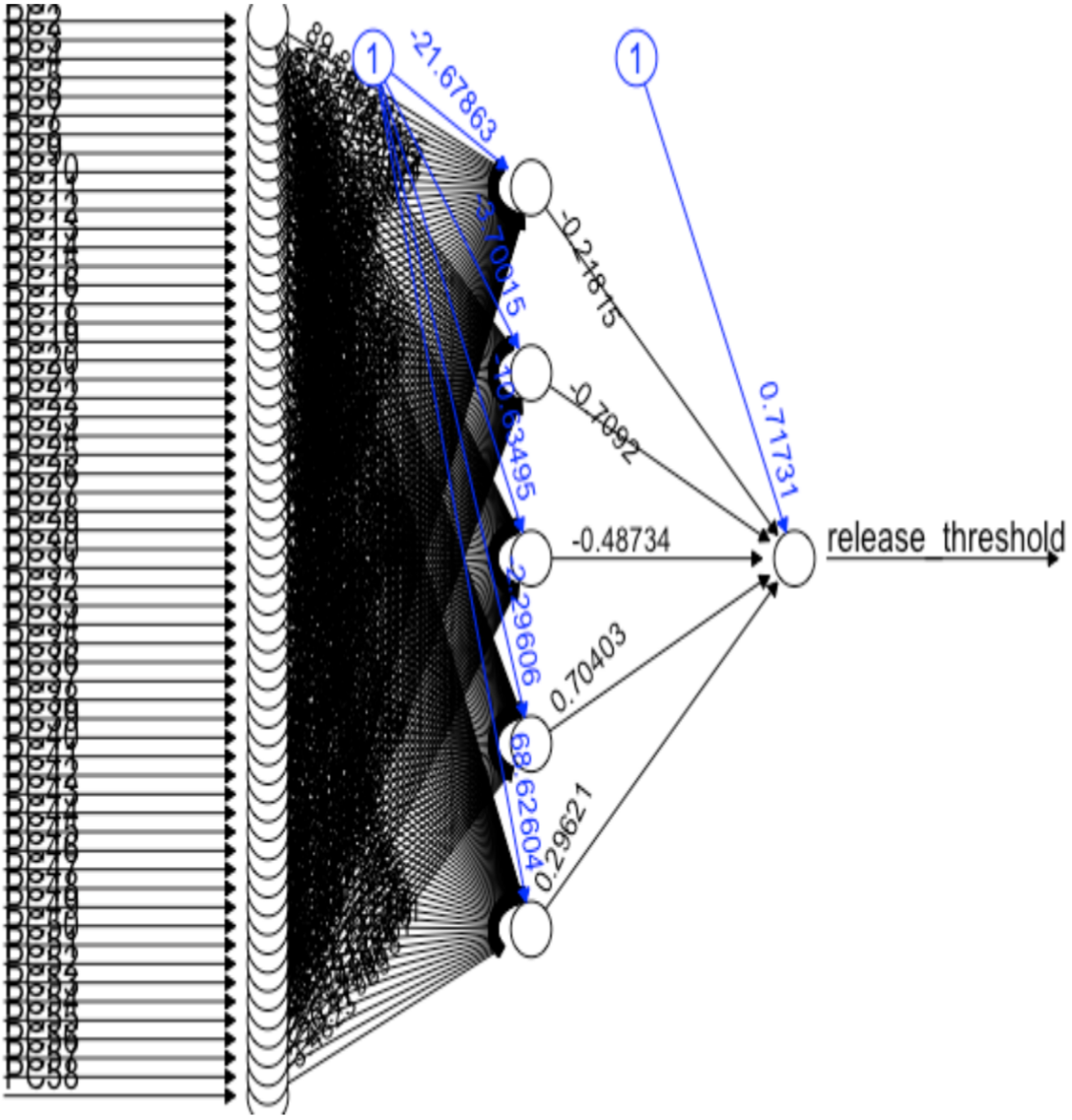

Figure 18: Neural network plot with 5 nodes on hidden layer

Note that for figures 17 and 18, the numbers on the black arrows represent the weights of the previous nodes that contribute to the forward node and for the blue arrows they represent the bias multiplicator constant that affects the forward node.
The base neural network with a single hidden layer and node was distributed to make its predictions using the neuralnet::compute() function. Similarly to the decision tree and random forest predictors, the neural network classifier was specified inside the function. The training dataset with the removal of the binary target variable was also specified inside the function, the numerical position of the target variable was stated so the function could correctly extract it. Once again, a results table was crafted indicating whether or not the transfer market value was over 1.6 million euros using the levels of the binary target variable. The first column displayed the actual values from the test dataset and the second column displayed the predictions made by the base neural network predictor. The cumulation table was created once again using the table() function which showed 2214 true positive counts, 187 false positive counts, 266 false negative counts and 2139 true negative counts.

```
##        nn_1
## actual    0    1
##      0 2139  187
##      1  266 2214
```

Figure 19: Cumulative table for base neural network

From the cumulation table above, we see that the base neural network predicted a total of 4353 players that were valued fairly, which was roughly 90.6% of all players. Looking at the false positive rates we can assume that 187 players were undervalued, this was about 3.9% of all players available. Viewing the false negative rates we can assume that 266 players were overvalued, this was about 5.5% of all players evaluated.

According to these prediction rates, the base neural network had an accuracy of 0.9057. All 58 numeric features used in this prediction were principal components. The first principal component held roughly 13% of the variance, the second component held roughly 8% and the third component held roughly 4%. The point being that the first principal component held more weight in the prediction than the rest of the components and this rule continued for every consecutive component until the 58th was reached.

Chapter 6: Binary classification results: Neural networks

The neural network with 5 nodes across a single hidden layer was set up and utilised the same exact way as the original base neural network classifier. For this new neural network the true positives were numbered at 2224, false positives were numbered at 198, false negatives were numbered at 254 and the true negatives were numbered at 2128.

```
##        nn_5
## actual    0    1
##      0 2128  198
##      1  254 2224
```

Figure 20: Cumulative table for neural network with 5 nodes on a hidden layer

From the cumulation table above, we see that the base neural network predicted a total of 4352 players that were valued fairly, which was again roughly 90.6% of all players. Looking at the false positive rates we can assume that 198 players were undervalued, this was about 4.1% of all players available. Viewing the false negative rates we can assume that 254 players were overvalued, this was about 5.3% of all players evaluated.

This resulted in the accuracy of this neural network to be a value of 0.9055 which is slightly less accurate than the original neural network, however the difference in accuracy is not significant as the difference is less than 0.02%. Again all 58 principal components were weighted unevenly, with the initial components carrying greater weight than the ensuing components to produce the high accuracy of this classifier.

This suggests that the default neural network with a single node was the better model due to the difference in accuracy between the two classifiers being negligent. Another reason was the neural network with 5 nodes at the hidden layer was far less efficient during the training process as data processing was more time consuming for this more complex model. As Brownlee (2020) proposes that a scarcity in the number of nodes distributing the weight across the network leads to underfitting and an abundance in the number of nodes distributing the weight leads to overfitting towards the training data which can cause the predictor to go astray, which suggests that a higher number of neurons on a hidden layer does not necessarily guarantee a higher accuracy during the predictions.

Chapter 7: Binary classification results: Support vector machines

R studio was the platform used for the creation, testing and training of the numerous support vector machine (SVM) classifiers. Again a specific dataset had to be created for these series of predictors as SVMs are known notoriously for not being able to process raw independent categorical variables. This newly created dataset was similar to the dataset that was produced for neural networks however only the categorical variables were one hot encoded and the data normalisation and principal component analysis techniques were not applied as these procedures were not paramount to creating adequate SVM classifiers. Prior to encoding, the original dataset had 57 variables which consisted of 46 numerical variables, 11 categorical variables and 0 string/character variables. After encoding the dataset had 126 variables which consisted of 125 numerical variables, 1 categorical variable and 0 string/character variables. The only categorical variable being the binary target variable for predicting the transfer market value. The same 70/30 training and testing split was again deployed using sample() function to create the new training index and the split was made reproducible with a new set.seed() function. Once again, a new encoded formula was produced for the training process which consisted of the sum of all numerical and encoded categorical independent variables equalling the target binary variable.

Support vector machines (SVM) are a supervised learning method that can be used for both regression and classification. They work by drawing a line between two classes of data points called the hyperplane, however there are many choices for the hyperplane which are dictated by the chosen kernel function. Instances of data points falling on the wrong side of hyperplane leads to misclassification, this is where the idea of a soft margin is introduced and are best described as the more versatile versions of the original hard maximal margins. Soft margins are generally more robust and forgiving than hard margins, which is the main reason that they are preferred during the training process. Also any two classes that are not linearly separable a different kernel function must be used instead of the linear function i.e polynomial function, radial basis function, sigmoid function which is better known as the hyperbolic tangent function.

The first SVM that was trained was one that possessed the radial basis kernel function, the kernel function is also often referred to as Gaussian. The radial basis is a kernel function that is often used to classify relationships between variables that are found to be non linear in a dataset. The SVM with the radial basis kernel function was trained using the ‘kernlab’ package. The ksvm() function alongside the encoded formula was used to train on the encoded training dataset. The kernel was specified as ‘rbfdot’ which represented the polynomial kernel function and was given a cost value of 1, the cost being a parameter that determines how much learning power is given to the classifier to allow it to make a greater number of correct predictions. The number of support vectors created numbered at 1731. This SVM made its predictions using the predict () function, with the classifier and training data specified alongside it. Similar to previous binary supervised methods, results tables were created alongside the cumulative tables. The true positives numbered at 2228, false positives numbered at 55, false negatives numbered at 238 and true negatives numbered at 2285.

```
##        rad_svm
## actual    0    1
##      0 2285   55
##      1  238 2228
```

Figure 21: Cumulative table for radial basis SVM

From the cumulation table, we can see that the radial basis SVM predicted 4513 players that were valued fairly, which was roughly 93.9% of all players. Looking at the false positive rates we can assume that 55 players were undervalued, this was about 1.1% of all players available. Viewing the false negative rates we can assume that 238 players were overvalued, this was about 5% of all players evaluated.

This gave a moderately high accuracy of 0.9390. The original 46 numerical features and 10 categorical features held even weight in the prediction formula for the transfer market value. However after encoding, out of the 125 numerical features the original numerical features held greater weight than the newly encoded categorical features as each level of the original categorical variable was given a unique encoded numerical variable which in turn would hold less weight in the accuracy that was achieved for all SVM predictors produced.

The second SVM that was trained was one that had the polynomial kernel function. The polynomial basis is another kernel function used to classify non linear relationships in a dataset, however it differs from the previously mentioned function due to radial basis kernels using the curvature of the clusters formed by the data points to form the various hyperplanes. The SVM with the polynomial kernel function was also trained using the ‘kernlab’ package, ksvm() function, encoded formula and the same training dataset. However, the kernel that was defined differed and was specified as ‘polydot’ for the polynomial kernel and was again given a cost parameter of 1. The support vectors created for the polynomial SVM were numbered at 935. This SVM made its predictions the exact same way as the radial basis classifier. The true positives numbered at 2345, false positives numbered at 187, false negatives numbered at 121 and true negatives numbered at 2153.

```
##        poly_svm
## actual    0    1
##      0 2153  187
##      1  121 2345
```

Figure 22: Cumulative table for polynomial SVM

From the cumulation table, we can see that the polynomial SVM predicted 4498 players that were valued fairly, which was roughly 93.6% of all players. Looking at the false positive rates we can assume that 187 players were undervalued, this was about 3.9% of all players available. Viewing the false negative rates we can assume that 121 players were overvalued, this was about 2.5% of all players evaluated.

This gave a fairly high accuracy of 0.9359. As mentioned earlier, out of the 125 numerical variables the original numerical values held greater weight than the newly encoded categorical variables in making the predictions above and hence achieving the corresponding accuracy value.

The third SVM that was trained was one that used the linear kernel function and is often the default kernel function for regular SVMs. The linear kernel function is usually deployed to classify data points that could be simply separated using a hyperplane that forms the shape of a simple line. Once again, the 'kernlab' package and the encoded formula and training dataset specified inside the ksvm() function were used to train the linear SVM. The kernel function was specified as 'vanilladot', a reference to the default/vanilla kernel function being linear, and was again given a cost value of 1. The support vectors created for the classifier numbered around 980. This SVM also made its predictions following the exact same procedures of the 2 previous SVM classifiers. The true positives numbered at 2067, false positives numbered at 15, false negatives numbered at 399 and true negatives numbered at 2325.

```
##        lin_svm
## actual    0    1
##      0 2325   15
##      1  399 2067
```

Figure 23: Cumulative table for linear SVM

From the cumulation table, we can see that the linear SVM predicted 4392 players that were valued fairly, which was roughly 91.4% of all players. Looking at the false positive rates we can assume that 15 players were undervalued, this was about 0.3% of all players available. Viewing the false negative rates we can assume that 399 players were overvalued, this was about 8.3% of all players evaluated.

This gave an accuracy of roughly 0.9139. Although this predictor was slightly less accurate than the 2 previous classifiers, the rate of the correctly made predictions was still respectably high. Again the original numerical variables held a greater weight and hence a greater role in accurately predicting the transfer market value compared to the encoded categorical variables.

The final SVM that was created was one that used the hyperbolic tangent kernel function or the sigmoid kernel function. The hyperbolic tangent is yet another kernel function used to classify non linear relationships in a dataset, it works by classifying data points that can be separated by a hyperplane which forms a ‘S’ like shape. Similar to all other SVMs discussed, the ksvm() function from the ‘kernlab’ package was used in conjunction with the encoded training dataset and the instructions of the encoded formula to train the hyperbolic tangent SVM. The kernel was referred to as ‘tanhdot’ which represented the hyperbolic tangent function, as the hyperplane created by the kernel function corresponds in shape to the y=tanh(x) graph/plot. The cost parameter was again assigned to a value of 1.Compared to the most recent previously trained classifiers, the amount of support vectors was much higher at a number of 10,700. This SVM was no different to the ones trained previously, as its predictions were created using the predict() function. From the cumulation table, the true positives numbered at 2466, false positives numbered at 2340, false negatives numbered at 0 and true negatives numbered at 0.

```
##         hyp_svm
## actual     0    1
##      0     0 2340
##      1     0 2466
```

Figure 24: Cumulative table for hyperbolic tangent SVM

From the cumulation table, we can see that the hyperbolic tangent SVM predicted 2466 players that were valued fairly, which was roughly 51.3% of all players. Looking at the false positive rates we can assume that 2340 players were undervalued, this was about 48.7% of all players available. Viewing the false negative rates we can clearly see that 0 players were overvalued.

Chapter 7: Binary classification results: Support vector machines

From these prediction counts, the hyperbolic tangent SVM performed very poorly in comparison to the 3 other classifiers as the accuracy was around 0.5131. From the counts we can also tell that the hyperbolic tangent SVM predicted every footballer to make over 1.6 million euros, this clearly shows that the points in the encoded dataset could not be split and correctly classified with a sigmoid shaped hyperplane. Awasthi (2020) explains that this kernel is most commonly used for the creation of specific types of neural network predictors, which hints as a possible cause for the poor performance of this particular SVM classifier. Similar to all other SVMs created, the performance of this classifier was influenced substantially more by the original numerical variables in comparison to the encoded categorical variables.

Chapter 8: Binary classification results: K nearest neighbours

R studio was the platform used for the creation, testing and training of the single binary k nearest neighbours (KNN) classifier. The dataset procured for this supervised learner was very similar to the dataset created for the support vector machine classifiers, however in accordance with the application of one hot encoding, data normalisation in the form of a minmax function was also applied to the dataset. The procedure of data normalisation was required due to the fact that the scale of the variables in the dataset ranged hugely and the minmax function nullifies any variables that would have a much larger effect on predicting the target variable compared to other attributes and effectively rescales every attribute in the dataset. The procedure of normalisation is imperative as KNN works by calculating distances between the different data points. Before the encoding and data normalisation procedures, the original dataset had 57 variables with 46 numerical variables, 11 categorical variables and 0 string/character variables. After the encoding and data normalisation procedures, the dataset had 126 variables with 125 numerical variables, 1 categorical variable and 0 string/character variables. These are the exact same figures for variables in the SVM encoded dataset, which makes sense as the only difference between the 2 datasets is that data has been normalised which does not change the data type of any attributes. The 70/30 standard training and testing split was created using the sample() to create the new minmax training index, which again used the set.seed() function to make the split replicable. However, compared to every other binary classifier created, the KNN classifier did not require a unique formula to be generated instructing it on how to perform. This is related to the disadvantage of KNN classifiers mentioned in the approach/methodology section, which stated that the KNN algorithm does not try to to explain the relationship between inputs and outputs as this supervised method is unique and does not produce a concrete model.

K nearest neighbours (KNN) is a supervised learning method that can be used for regression and classification, k represents the number of nearest neighbours to the target variable/attribute that is being predicted or classified. The nearness is measured by euclidean distance and the k points selected often require normalisation/standardisation but the procedure is not mandatory. However, this was not the case for the particular dataset used in this study. The k value can be seen as the counterpart to the formulas of the previously mentioned binary classifiers and is regarded as the starting point of the KNN process. The value of k should begin with being the square root of the number of data points

and then a range of values should be tested and fine tuned to optimise the performance. This piece of advice stood out to be notable, as for this study the square root of the number of data points in the training set was found to be optimal and no other tested k value was close to producing a similar rate of correct predictions.

The KNN classifier was trained using the 'class' package. The knn() function was used to create the predictions with the k value that was prescribed inside the function. The other parameters that were assigned to this function included the training and testing datasets with the binary target variables removed respectively in each. Similar to all other binary predictors, the results tables and cumulation tables were produced. For this particular KNN classifier the true positive counts were 1812, false positives were numbered at 353, false negatives were numbered at 668 and the true negatives were numbered at 1973.

```
##        knn
## actual    0    1
##      0 1973  353
##      1  668 1812
```

Figure 25: Cumulative table for KNN

From the cumulation table above, we can see that the KNN classifier predicted 3785 players that were valued fairly, which was roughly 78.7% of all players. Looking at the false positive rates we can assume that 353 players were undervalued, this was about 7.3% of all players available. Viewing the false negative rates we can assume that 668 players were overvalued, this was about 13.9% of all players evaluated.

This gave the KNN classifier the accuracy at an average to fairly high value of 0.7876. Similar to the SVMs, the original 46 numerical features and 10 categorical features held even weight in the predictions made and hence all features shared equal liability in the accuracy value produced. However, after the use of encoding and data normalisation, all 125 of the encoded features still held equal weight as the normalisation process transformed all features to an equal scale.

Although the classifier makes correct predictions of transfer market value at a moderately high rate, it pales in comparison to every other binary classifier with the exception of the SVM with hyperbolic tangent kernel function. This observation is not down to this KNN predictor being a poor classifier rather the standard of the predictions made by the majority of classifiers' being set so high as the accuracies averaged roughly 90% or above, again excluding the hyperbolic tangent SVM.

However, there is always room for improvement and hence reasons to excuse and explain the slightly less impressive results of the KNN were explored. Jain (2020) suggests that datasets with a large number of entries as well as datasets that contain independent categorical variables with a large amount of levels cause KNN classifiers to perform at a subpar level. These two features are both prominent in the dataset which was used for training the KNN classifier and hence are valid reasons why the KNN classifier did not perform at the highest accuracy rates. However, the problem regarding the large amount of levels for categorical variables was nullified to a certain degree using one hot encoding. Also the idea that KNN does not study or learn the intricate details of relationships between variables in datasets but just simply remembers the training data makes KNN classifiers prone to overfitting towards the training data. This training behaviour has given KNN classifiers the reputation of lazy learners.

Chapter 9: Findings and conclusion

9.1 Binary classifiers - Looking at counts of true positives and true negatives to find the number of fairly valued footballers

From the binary performance evaluators mentioned in the various analyses in the results sections above, we can see the sum of the true positive and true negative figures that each classifier produced to infer the number of fairly valued footballers:

Rank of predictor	Supervised ML method	Number of fairly valued footballers	Percentage of fairly valued footballers
1	Decision tree classifier	4602	95.75%
2	Random forest classifier	4592	95.54%
3	Radial basis kernel SVM	4513	93.90%
4	Polynomial kernel SVM	4498	93.59%
5	Linear Kernel SVM	4392	91.39%
6	Base Neural Network with a single node on hidden layer	4353	90.57%
7	Neural Network with 5 nodes on a hidden layer	4352	90.55%
8	KNN classifier	3785	78.76%
9	Hyperbolic tangent kernel SVM	2466	51.31%

From the results above, we can see that generally the decrease of rank of the predictor in terms of its accuracy caused a decrease in the number of fairly valued footballers. Although the majority of binary classifiers performed exceptionally, the evaluation results presented were still surprising. The top 2 performing classifiers were the simplest supervised methods, although the more complex supervised learners were not far behind in terms of the accuracy of predictions. We can see that the aim of finding the true positive and true negative outcomes was a successful task to find the players that were fairly valued. The results show that the majority of models created indicate that most players in the FIFA dataset were fairly valued.

9.2 Binary classifiers - Looking at counts of false positives to find the number of undervalued footballers

Again looking at the various binary evaluators mentioned in the analyses sections, we can see the false positive figures that each classifier produced in terms of predicting the number of undervalued footballers:

Rank of predictor	Supervised ML method	Number of undervalued footballers	Percentage of undervalued players
1	Decision tree classifier	74	1.5%
2	Random forest classifier	96	2%
3	Radial basis kernel SVM	55	1.1%
4	Polynomial kernel SVM	187	3.9%
5	Linear Kernel SVM	15	0.3%
6	Base Neural Network with a single node on hidden layer	187	3.9%
7	Neural Network with 5 nodes on a hidden layer	198	4.1%
8	KNN classifier	353	7.3%
9	Hyperbolic tangent kernel SVM	2340	48.7%

We can see from the results above, that generally the decrease in rank of the predictor in terms of its accuracy caused an increase in the number of undervalued footballers. However, the linear SVM was found to be the only classifier that was an exception to this trend, as it surprisingly only found 15 players that were undervalued which is the smallest amount of the entire group. We can see that the aim of finding the false positive outcomes was again generally a successful task to find the amount of players that were undervalued.

9.3 Binary classifiers - Looking at counts of false negatives to find the number of overvalued footballers

Similarly, by looking at the binary evaluators, we can view the false negative figures produced by each binary classifier to predict the number of overvalued footballers:

Rank of predictor	Supervised ML method	Number of overvalued footballers	Percentage of overvalued players
1	Decision tree classifier	130	2.7%
2	Random forest classifier	118	2.5%
3	Radial basis kernel SVM	238	5%
4	Polynomial kernel SVM	121	2.5%
5	Linear Kernel SVM	399	8.3%
6	Base Neural Network with a single node on hidden layer	266	5.5%
7	Neural Network with 5 nodes on a hidden layer	254	5.3%
8	KNN classifier	668	13.9%
9	Hyperbolic tangent kernel SVM	0	0%

Looking at the results above, it was surprising to find that there seemed to be no obvious trend or relationship with the rank of the predictor in terms of its accuracy and the number of overvalued footballers predicted by the various binary predictors. Looking at all 3 tables of results in a more general sense, we can see that the superiority of the decision tree and random forest classifiers could be down to the fact that they could naturally interpret both numerical and categorical features without the assistance of encoding. The same can not be said for the SVM and KNN classifiers as they use distance based calculations for producing predictions and neural networks simply require all inputs to be in the form of numerical data. So the use of principal component analysis and encoding had clearly affected the results, as 80% of the variance of the original dataset was captured in the principal components and this was seen as an acceptable threshold. This was not the same for the raw dataset used for the decision tree and random forest classifiers. In hindsight to these results, the accuracy

of the SVM, Neural network and KNN classifiers could have possibly been improved by increasing the variance threshold of the principal components used for these respective prediction models.
We can see that the aim of finding the false negative outcomes was again a successful task to find the amount of players that were overvalued. The result tables show that for the majority of models created, a higher population of players were overvalued compared to those that were undervalued. Hence, we can conclude that by and large that the vast majority of players were fairly valued, the second biggest majority were players that were overvalued and the smallest minority were players that were undervalued.

Chapter 10: Future progression on current research

The first way this research could be progressed was if the supervised machine learning models created in this study would be upscaled to a much larger degree by machine learning engineers, this could possibly benefit professional football clubs during actual transfer processes, as the upscaled models would be much more accurate. The upscaled models would be updated and specified to real time data by the expertise of professional machine learning engineers. This would greatly reduce the problem of overpayments for transfer fees and would reduce the time wasted on negotiations. The negotiations would be direct as there would be data backing the offers made for any specific player.

Another way this research could be furthered or expanded upon could be the prediction of players' weekly wages and this does not have to be restricted to both binary or multiclass classification. The use of regression models rather than classification could perhaps uncover further insights that were not discovered from this study. As the only focus of this study was on the results of the binary classification, perhaps multiclass classification models could be created on a more sophisticated platform compared to R studio such as Python.

References

Park, M. and Lee, K., (2017). Liability of High Status: Overpayment to Relieve Status Anxiety in the English Premier League. *Academy of Management Proceedings*, 2017(1), p.13416.

Maguire, K., (2021). *Premier League wages have doubled since 2010 with total wage cost of clubs now approaching £3billion [Study]*. [online] WhatAreTheOdds. Available at: <https://www.whataretheodds.co.uk/premier-league-wages/> [Accessed 27 July 2022].

Auberg, E., (2022). *Terminating a contract between a football club and a football player | EA Sports Law*. [online] EA Sports Law. Available at: <https://www.easportslaw.com/news/terminating-a-contract-between-football-club-and-a-player> [Accessed 27 July 2022].

Andreff, W., (2018). Financial and Sporting Performance in French Football Ligue 1: Influence on the Players' Market. *International Journal of Financial Studies*, 6(4), p.91.

Sky News, (2022). *Kylian Mbappe: LaLiga criticises 'scandalous' deal to make PSG star world's highest paid footballer*. [online] Sky News. Available at: <https://news.sky.com/story/kylian-mbappe-laliga-criticises-scandalous-deal-to-make-psg-star-worlds-highest-paid-footballer-12618564#:~:text=Mbappe%20becomes%20the%20highest%2Dpaid,the%20region%20of%20%C2%A3100m.> [Accessed 27 July 2022].

Al-Asadi, M. and Tasdemir, S., (2022). Predict the Value of Football Players Using FIFA Video Game Data and Machine Learning Techniques. *IEEE Access*, 10, pp.22631-22645.

EDUCBA, (2022). Random Forest vs Decision Tree | Top 10 Differences You Should Know. [online] EDUCBA. Available at: <https://www.educba.com/random-forest-vs-decision-tree/> [Accessed 29 August 2022].

Papadaki, I. and Tsagris, M., (2020). Estimating NBA players salary share according to their performance on court: A machine learning approach. *arXiv preprint arXiv:2007.14694*.

Yaldo, L. and Shamir, L., (2017). Computational Estimation of Football Player Wages. *International Journal of Computer Science in Sport*, 16(1), pp.18-38.

Lee, C.Y., Hsu, P.Y., Cheng, M.S., Leu, J.D., Xu, N. and Kan, B.L. (2021). Using Machine Learning to Predict Salaries of Major League Baseball Players. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (pp. 28-33). Springer, Cham.

Mahadevan, S. (2020). Predicting Market Value of Football Players using Machine Learning Algorithms. 10.13140/RG.2.2.21487.46248.

Varghese, D., (2018). *Comparative study on Classic Machine learning Algorithms*. [online] Towards data science. Available at: <https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222> [Accessed 30 June 2022].

References

Bank of England, (2019). *Why are football players paid so much?*. [online] Bankofengland.co.uk. Available at: <https://www.bankofengland.co.uk/knowledgebank/why-are-football-players-paid-so-much#:~:text=paid%20so%20much.-,Why%20have%20footballers'%20wages%20increased%20so%20much%3F,popular%20and%20so%20more%20profitable.> [Accessed 30 June 2022].

Meneses Flores, R., (2017). What Influences Football Enthusiasts When Setting a Player`s Market Value (pp. 36-39).

Lyons Jr., R., Jackson Jr., E. and Livingston, A., (2015). Determinants of NBA Player Salaries. *The Sport Journal*,.

Wasserman, T., (2013). Determinants of Major League Baseball Player Salaries (pp. 75-79).

Baeldung, (2022). *Advantages and Disadvantages of Neural Networks*. [online] Baeldung.com. Available at: <https://www.baeldung.com/cs/neural-net-advantages-disadvantages> [Accessed 1 July 2022].

Section, (2020). *Introduction to Random Forest in Machine Learning*. [online] Engineering Education (EngEd) Program | Section. Available at: <https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/> [Accessed 1 July 2022].

Gandhi, R., (2018). *Support Vector Machine — Introduction to Machine Learning Algorithms*. [online] towardsdatascience.com. Available at: <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47> [Accessed 1 July 2022].

Harrison, O., (2018). *Machine Learning Basics with the K-Nearest Neighbors Algorithm*. [online] towardsdatascience.com. Available at: <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761> [Accessed 2 July 2022].

Kaggle.com. (2022). FIFA dataset. [online] Available at: <https://www.kaggle.com/datasets/stefanoleone992/fifa-22-complete-player-dataset?select=players_22.csv> [Accessed 22 August 2022].

Selvaraj, N., (2022). *Decision Trees vs Random Forests, Explained - KDnuggets*. [online] KDnuggets. Available at: <https://www.kdnuggets.com/2022/08/decision-trees-random-forests-explained.html> [Accessed 11 August 2022].

Brownlee, J., (2020). *How to Control Neural Network Model Capacity With Nodes and Layers*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/how-to-control-neural-network-model-capacity-with-nodes-and-layers/> [Accessed 11 August 2022].

References

Awasthi, S., (2020). *Seven Most Popular SVM Kernels*. [online] Dataaspirant. Available at: <https://dataaspirant.com/svm-kernels/#t-1608054630732> [Accessed 12 August 2022].

Jain, D., (2020). *KNN: Failure cases, Limitations and Strategy to pick right K*. [online] Medium. Available at: <https://levelup.gitconnected.com/knn-failure-cases-limitations-and-strategy-to-pick-right-k-45de1b986428> [Accessed 13 August 2022].

Appendix

College of Engineering, Design and Physical Sciences Research Ethics Committee
Brunel University London
Kingston Lane
Uxbridge
UB8 3PH
United Kingdom

www.brunel.ac.uk

16 June 2022

LETTER OF CONFIRMATION

Applicant: Mr Manraj Rai

Project Title: Big data on prediction of professional football players' transfer market value using both real game statistics combined with FIFA video game attribute data (V3)

Reference: 37585-NER-Jun/2022- 39848-1

Dear Mr Manraj Rai

The Research Ethics Committee has considered the above application recently submitted by you.

The Chair, acting under delegated authority has confirmed that, according to the information provided in your application, your project does not require ethical review.

Please note that:

- **You are not permitted to conduct research involving human participants, their tissue and/or their data. If you wish to conduct such research, you must contact the Research Ethics Committee to seek approval prior to engaging with any participants or working with data for which you do not have approval.**
- The Research Ethics Committee reserves the right to sample and review documentation relevant to the study.
- If during the course of the study, you would like to carry out research activities that concern a human participant, their tissue and/or their data, you must inform the Committee by submitting an appropriate Research Ethics Application. Research activity includes the recruitment of participants, undertaking consent procedures and collection of data. Breach of this requirement constitutes research misconduct and is a disciplinary offence.

Good luck with your research!

Kind regards,

Professor Simon Taylor

Chair of the College of Engineering, Design and Physical Sciences Research Ethics Committee

Brunel University London