Market Value Estimation of Football Players

Submitted in Partial Fulfilment of requirements for the Award of certificate of

Post Graduate Program in Data Science and Business Analytics

Capstone Project Report

Submitted to

GREAT LAKES
INSTITUTE OF MANAGEMENT
Global Mindset - Indian Roots

Submitted by

1. Manvir Singh Kohli – Roll No. 725QXVGQ2O

2. Kunal Malik – Roll No. F4MZM1ML6W

3. Abhishek Sharma – Roll No.EU6UKFRTGH

4. Akansha Gupta – Roll No. FMSWB7IHPO

Under the guidance of

Mr. Pranov Mishra

Batch – (PGPBABI.G.Sep'19)

Year of Completion (March' 2021)

CERTIFICATE

This is to certify that the participants **Manvir Singh Kohli**, **Kunal Malik**, **Abhishek Sharma**, **Akansha Gupta** who are the students of Great Lakes Institute of Management , have successfully completed their project on "**Market Value Estimation of Football Players**"

This project is the record of authentic work carried out by them during the academic year 2019- 2020.

PMishra

Mentor	Program Director
Name: Mr. Pranov Mishra	Name : Dr. Surabhi Basu

Date : 08/03/2021
Place: Gurugram

Contents

List of Figures

List of Tables

1. Introduction

Football is without a doubt the most popular sport in the world with approximately 265 million people (4% of the entire world population) actually participating in it either as players or as referees). In terms of viewership as well, the sport has its rivals beat by a considerable margin, with a fan following of approximately 3.5 billion worldwide.[1]

While majority of us look at football as a source of entertainment, for a select few this sport is also a business. This business aspect of football comes into play at the club level. Just like the Indian Premier League in cricket has teams like the Royal Challengers Bangalore and Chennai Superkings, similarly almost every country in the world has its own football leagues, with each league having individual teams, also called football clubs. These football clubs participate in various competitions all year round. And with the vast viewership that they attract, these clubs represent a huge business opportunity for the owners.

There are various ways in which a football club makes money. First, there are broadcasting rights. For example, "In the United Kingdom, Sky Sports and BT Sports own rights to cover the Premier League, a right they acquired for 5 Billion Pounds in a deal that will run for 4 years (2015–2019). This money will be shared among clubs, equally. That will amount to roughly 81 million pounds per club and that's only considering deals from the U.K alone." [2]

Then there's matchday revenue, that is, money made from ticket sales. These extremely vital for football clubs. The money coming in from the sale of match and season tickets is needed for the day to day functioning of the club. "The Deloitte report for the 2014-15 season showed that Arsenal (a team of UK's Premier League) raked in £101.84 million in matchday."[3]

Next there's sponsorships and merchandise sales. "Brands pay huge amount of money for clubs to advertise them. For example, Chevrolet pays Manchester United roughly 50 million pounds per year to have their logo and name on the United Jersey, Adidas pays United another 75 million pounds per year to sponsor the kits. And that's just two sponsors of many a club can have globally." Further football clubs and the sponsors together "make money from selling kits and other club merchandise to fans around the world. It is common to find shops that sell Jerseys around stadiums, or even club owned stores that sell club merchandise."[2]

Finally, there's the prize money that clubs get by winning competitions. Although this may not be the main source of revenue , winning competitions may be seen as the pathway to maximizing earnings through all the other methods listed above. This is because by winning competitions , clubs earn fame and prestige which in turn leads to bigger prospects for sponsorships , broadcasting rights , merchandise sales etc.

The key to maximizing the income generated from all these sources are the clubs' most important assets – their players. In order to be able to win competitions, clubs aim to attract the best players. These players are the ones for whom fans pay money to watch, whose shirts they wear and whose performances decide which club will succeed. Therefore, it is quite obvious that to attract the best talents, the clubs must shell out large sums of money. Football clubs spend a huge amount of money every year to buy professional football players, during the transfer window. Predicting how much players value in the transfer market is one of the difficult tasks for managers as well as the owners of the club .In the 2018-19 season , the top 5 spending clubs spent a total of €1074.50 million on the purchase of football players[4]. The most expensive football player cost till date cost €222 million.[5] Thus, it seems an interesting prospect to understand what aspects of a player determine his market value and that is what this project attempts to do - on the basis of a combination of the players' real life as well as game-related statistics.

2. Literature Review

More recently, researchers have started to consider the players' market value. Predictive system modeling for football matches is hugely significant in terms of economic values but not merely an interest in academia. The players' market value has taken much attention from the researchers, More recently. Players contracts can be sold from one team to another, in which the market value of a player can be estimated. The estimation of market value by information-driven methodologies has not yet gotten up to speed in expert football. Because of the impediment in the site information researchers have made datasets and information-driven assessing strategies, for individual sports like tennis or snooker, whereby it is more straight forward to assess the players' value, independent of the team. However, to assess a team of players is not as straightforward and needs more complex data-driven approach.[6]

The primary sports club to utilize data analytics to enlist players was Oakland Athletics of the Major League Baseball (MLB) in the USA. The senior supervisor Billy Beane began utilizing measurable information before the finish of the 1990s for settling on choices about group program, a story well known through the bestseller, "Moneyball". Coming back to football , the proprietor of TSG Hoffenheim – a team of Germany's top football league called 'Bundesliga' – and fellow benefactor of SAP, Dietmar Hopp, also utilized the factual investigation of player attributes at Hoffenheim. As a player , Roberto Firmino, cost Hoffenheim just €4 million but was sold for a much higher exchange fee of around €41 million to the English team FC Liverpool. The exchange went through in the year 2015 and became the record-breaking most elevated exchange of FC Liverpool. Hopp recognized two achievement factors for progress: distinguishing youthful gifts and creating them, so that they can contribute both on the monetary record and on the pitch ,thus becoming an early adopter of inventive innovations.[6]

In a research Herm et al. aimed at estimating the transfer fee of football players based on players' talent and judgment variables. Talents used are age, success, assertion, flexibility, and precision. This model shows age is inversely proportional to players' market value. In this research, he used fans to estimate the value. So, the main drawback is the biased result and lacking sufficient knowledge. Frank and Nuesh [4] predicted players' market value by investigating the effect of talent and popularity. They have used nearly 20 parameters in correlation with their teams' success. Singh and Lamba used FIFA 18 dataset to predict market values. They have taken a different approach by predicting the player's value using in-game player prices. They were able to produce an accurate result as gamers can predict the values of in-game players easily. Yigit et al. predicted the market value using a dataset from a football manager simulation game. The data contains 49 attributes, each attribute is a number between 0 and 20 determined by professional football scouts. Felipe et al. has proposed a model that predicts the market value by analyzing the effect of team variable and player position.[6]

The analysis in this project involves a data driven approach whereby the data is a mixture of a player's real life performance statistics been - obtained from the website www.whoscored.com - as well as his gaming attributes in the game FIFA 20 by EA which are posted on the website www.sofifa.com. EA Sports employs a team of 25 EA Producers and 400 outside data contributors, who are led by Head of Data Collection & Licensing Michael Mueller-Moehring. This team is responsible for ensuring all player data is up to date, while a community of over 6,000 FIFA Data Reviewers or Talent Scouts from all over the world are constantly providing suggestions and alterations to the database. It is a complicated process, but its vast scope ensures that the information in the game itself is as accurate as possible, down to the weak-foot strength of lower league players and teenage reserves from minor leagues. It would be impossible for EA Sports' staff to watch every single player in every single game, so a team of over 6,000 volunteers help maintain and update the player database all year round. These FIFA Data Reviewers are known as FIFA Talent Scouts and are coaches, scouts, and football fans whose local knowledge helps maintain accuracy and ensure there are no major inconsistencies in each version of FIFA.
All the data is overseen by Mueller-Moehring's team of EA Producers and has to be backed up and verified before even a minor change is made. Each player in the game has over 300 fields as well as over 35 specific attributes which ultimately determine the rating seen in the game.[7] For the purpose of making the predictions, we will use several algorithms , which are discussed at length later in the report.

3. Problem Statement, Scope and Objective

As highlighted above, the trade of football players between clubs involve huge sums of money. Therefore, the problem at hand is a misevaluation of a player's worth, that can result in huge financial losses for the clubs.

The main objective of the project is therefore to use a suitable machine learning algorithm and come up with a model that can predict the market value of football players with maximum accuracy. Apart from predicting the market value, the project will also attempt to analyse and find out which features significantly determine the value of a footballer.

The model is going to be built using various game related as well as real life statistics of the 4705 players across the top 10 football leagues. The attributes cover the attacking and defending skillsets of a player as per the game EA FIFA 20 along with certain real-life performance metrics in the 2018-2019 season. (Please note that the stats provided to the player in the FIFA 20 game are based on the performance in the 2018-2019 season. Further the model will focus only on outfield players, meaning players playing in all positions except goalkeepers. This is because goalkeepers have a completely different set of attributes and cannot be directly compared with outfield players).

Figure 1: Process Flow of the Project

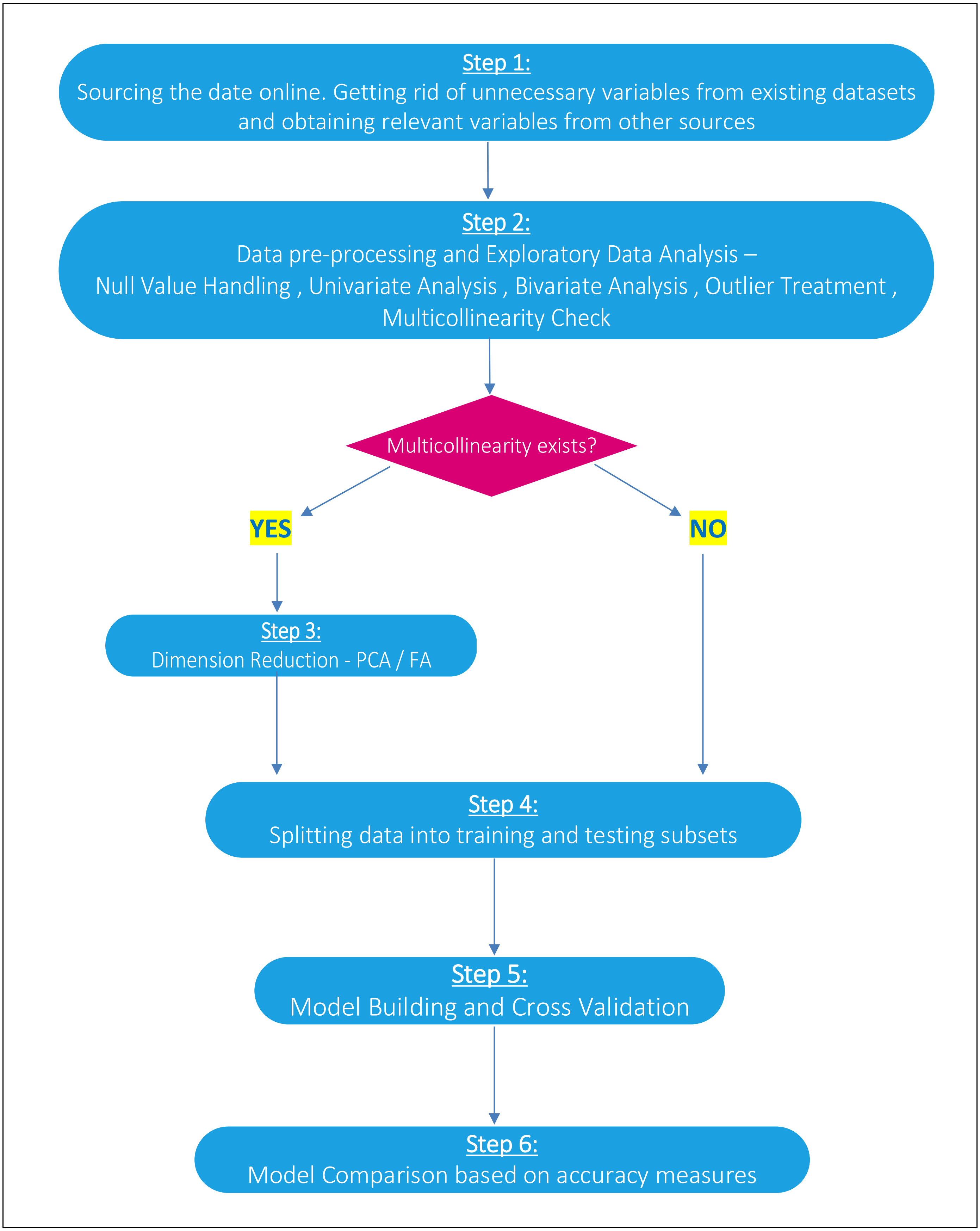

Step 1:
Sourcing the date online. Getting rid of unnecessary variables from existing datasets and obtaining relevant variables from other sources
Step 2:
Data pre-processing and Exploratory Data Analysis –
Null Value Handling , Univariate Analysis , Bivariate Analysis , Outlier Treatment , Multicollinearity Check
Multicollinearity exists?
YES
NO
Step 3:
Dimension Reduction - PCA / FA
Step 4:
Splitting data into training and testing subsets
Step 5:
Model Building and Cross Validation
Step 6:
Model Comparison based on accuracy measures

Table 1:Tools and techniques intended to be used in the analysis

Technique	Tool	Purpose
Data Collection	Chrome and Python (Selenium) and Alteryx	Downloading (partial) data from Kaggle and scraping the rest from www.whoscored.com and then combining the data using Alteryx
Data Pre-Processing and Exploratory Data Analysis	R, R Studio, Python	Performing descriptive statistics, data cleaning, null value handling outlier treatment and creation of visualizations for initial insights on data
Train and Test data split	R, R Studio, Python	Using random sampling to split the data into training and testing sets
Model Building	R, R Studio, Python	Building various predictive machine learning models and interpreting their results

4. Data Source and Description

- Part of the data has been sourced from the Kaggle (https://www.kaggle.com/stefanoleone992/fifa-20-complete-player-dataset#players_20.csv). The data on Kaggle was in turn obtained from the website www.sofifa.com. This part of the data contains each player's gaming attributes as in the game of EA FIFA 20.
- Apart from the variables from sofifa already available in the Kaggle data set , some additional variables were sourced from www.sofifa.com as well as www.whoscored.com. The data was scraped using a python script with the help of Selenium for python.
- The data obtained from the two sources was then combined using an Alteryx workflow to get the final dataset which contains the details 4705 unique football players. The below table captures the different variables in the dataset and a description of each variable.

The following variables were obtained from the Kaggle dataset

Table 2:Variables in the data set sourced from Kaggle

Variable Name	Variable Description
S.No.	Serial number of the observation
sofifa_id	Unique identifier of the player on www.sofifa.com
short_name	Player's truncated name
long_name	Player's full name
age	Age of the player
nationality	Nationality of the player

Player_Continent*	The geographical continent to which the player belongs
Club_Continent**	The geographical continent to which the player's club belongs
Value/Market_Value	Market value of the player in Euros as of 1st Oct 2019
Wages	Per week wage of the player in Euros for the season 2018-19 (ending in May'19)
Position_Final	The position in which the player plays
preferred_foot	Preferred foot of the player for kicking the ball
weak_foot	Weak foot rating on a scale of 1 to 5 as per the FIFA 20 Game
pace	The pace of the player on a scale of 100 as per the FIFA 20 game
shooting	The shooting accuracy of the player on a scale of 100 as per the FIFA 20 game
passing	The passing accuracy of the player on a scale of 100 as per the FIFA 20 game
dribbling	The dribbling skills of the player on a scale of 100 as per the FIFA 20 game
defending	The defending skills of the player on a scale of 100 as per the FIFA 20 game
physic	The physical strength of the player on a scale of 100 as per the FIFA 20 game

[*] Player_Contintent was obtained based on player nationalities so as to reduce the number of factor levels.

[**]Club_Contintent was obtained based on the country to which the clubs belong so as to reduce the number of factor levels

The additional variables below were scraped from the www.sofifa.com and www.whoscored.com

Table 3:Variables in the data set sourced from sofifa.com and whoscored.com

Variable Name	Variable Description
Club_Int_Prestige	The Clubs prestige rated from 1 to 10 amongst all the international clubs in world football, with 10 being the best rating possible
Club_Domestic_Prestige	The Clubs prestige rated from 1 to 10 amongst all the clubs within the league to which the club belongs, with 10 being the best rating possible
Player_international_reputation	Each player's reputation on an international scale from 1 to 5 with 5 being the highest rating
Mins_Total	Total minutes of football played during the 2018-19 season
Goals_Total	No. of goals scored by the player during the 2018-19 season
Assists_Total	No. of assists provided by the player during the 2018-19 season. As assist is a pass that directly results in a goal
Shots_per_game_total	No. of shots attempted by a player per game during the 2018-19 season
Yellow_Cards_Total	No. of yellow cards obtained by the player for committing fouls during the 2018-19 season. A yellow card in a game represents a warning for foul play.
Aerial_Battles_Won_Per_Game_Total	No. of headers won by a player per game throughout the 2019-19 season
Red_Cards_Total	No. of red cards obtained by the player for committing fouls during the 2018-19 season. A red card may be obtained directly or by obtaining 2

	yellow cards in the same game. While both kind of red cards involve the player being sent off in the current game , the former involves a suspension for 3 subsequent games whereas the latter involves a suspension for 1 subsequent game
Man_of_The_Match_Total	No. of man of the match accolades obtained by the player during the 2018-19 season

- The variable "Value" (initially read as "value_eur") is the response / dependent variable and all other variables are will be considered as independent variables in the beginning.

5. Data Preprocessing and Exploratory Data Analysis

The data was read in R as a table named fifa_data from an excel file and was then converted into a data frame. There is a total of 4705 observations across 30 variables. The data type of all variables was checked using the str () command to see if they have all been imported in R in the desired format. The data type of the following variables had to be converted into the correct format –

- S.no. and so_fifa_id were read as numeric variables but were converted into character variables as these numbers have no meaning.
- Nationality, Player_Continent, Club_Continent, preferred_foot, Position_Final were read by R as characters and were converted into factors
- The variables Club_Int_Prestige, Club_Domestic_Prestige, Player_international_reputation, weak_foot, were read by R as numeric but it must be noted that these are ordinal variables.
- Every other variable was read by R in the desired format
- The below table shows each variable and its data type.

Table 4: Variable data types

Variable Name	Data Type	Variable Name	Data Type
S.No.	Character	pace	Numeric
sofifa_id	Character	shooting	Numeric
short_name	Character	passing	Numeric
long_name	Character	dribbling	Numeric
Age	Numeric	defending	Numeric
Nationality	Factor w/ 122 levels	physic	Numeric
Player_Continent	Factor w/ 6 levels	Position_Final	Factor w/ 7 levels
Club_Continent	Factor w/ 7 levels	Mins_Total	Numeric
Club_Int_Prestige	Numeric (Ordinal)	Goals_total	Numeric

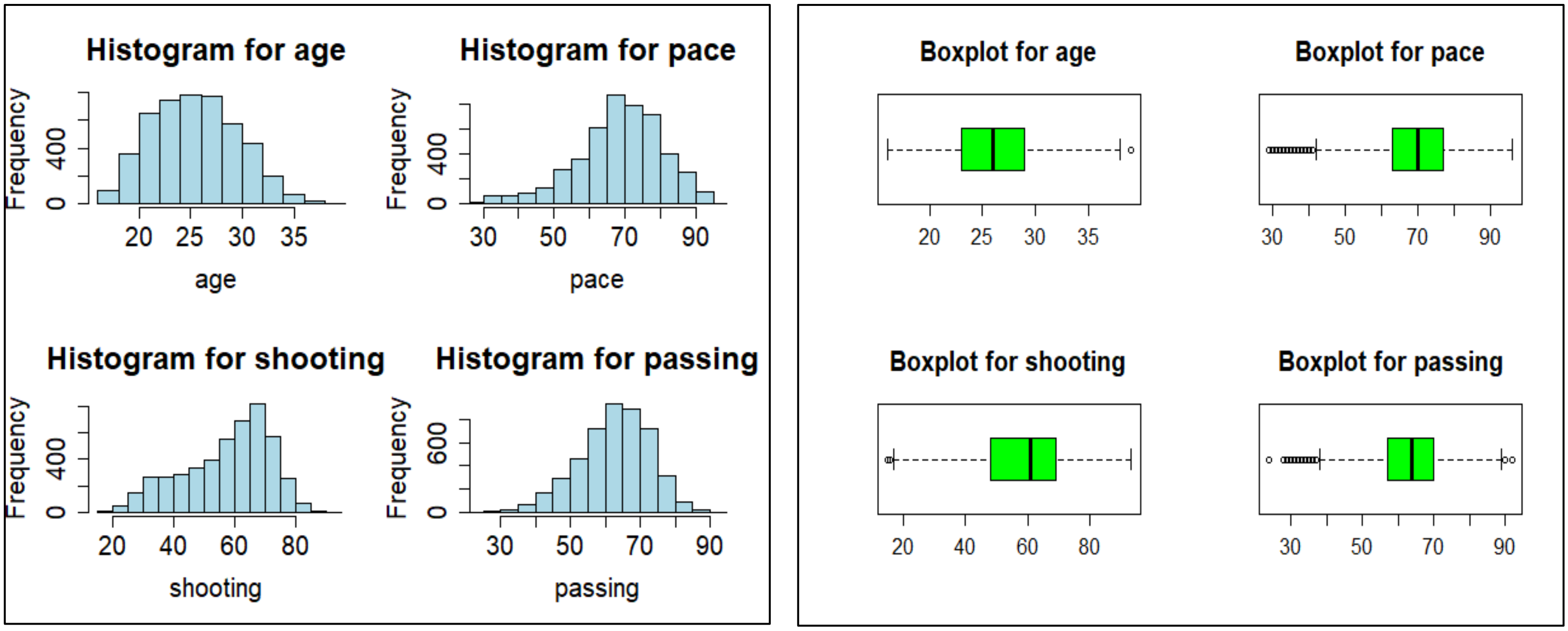
Club_Domestic_Prestige	Numeric (Ordinal)	Assists_Total	Numeric
Value/Market_Value/Mkt_Value	Numeric	Shots_per_game_total	Numeric
Wages	Numeric	Yellow_Cards_Total	Numeric
preferred_foot	Factor w/ 2 levels	Red_Cards_Total	Numeric
Player_international_reputation	Numeric (Ordinal)	Aerial_Battles_Won_Per_Game_Total	Numeric
weak_foot	Numeric (Ordinal)	Man_of_The_Match_Total	Numeric

- After all the variables were converted into the desired data types, the data was checked for null values using the **is.na ()** command in R. The result showed that there were no nulls in the dataset.
- The next step is to identify outliers in the data by performing univariate analysis on all numeric variables in the dataset.

Univariate Analysis – Numerical Variables

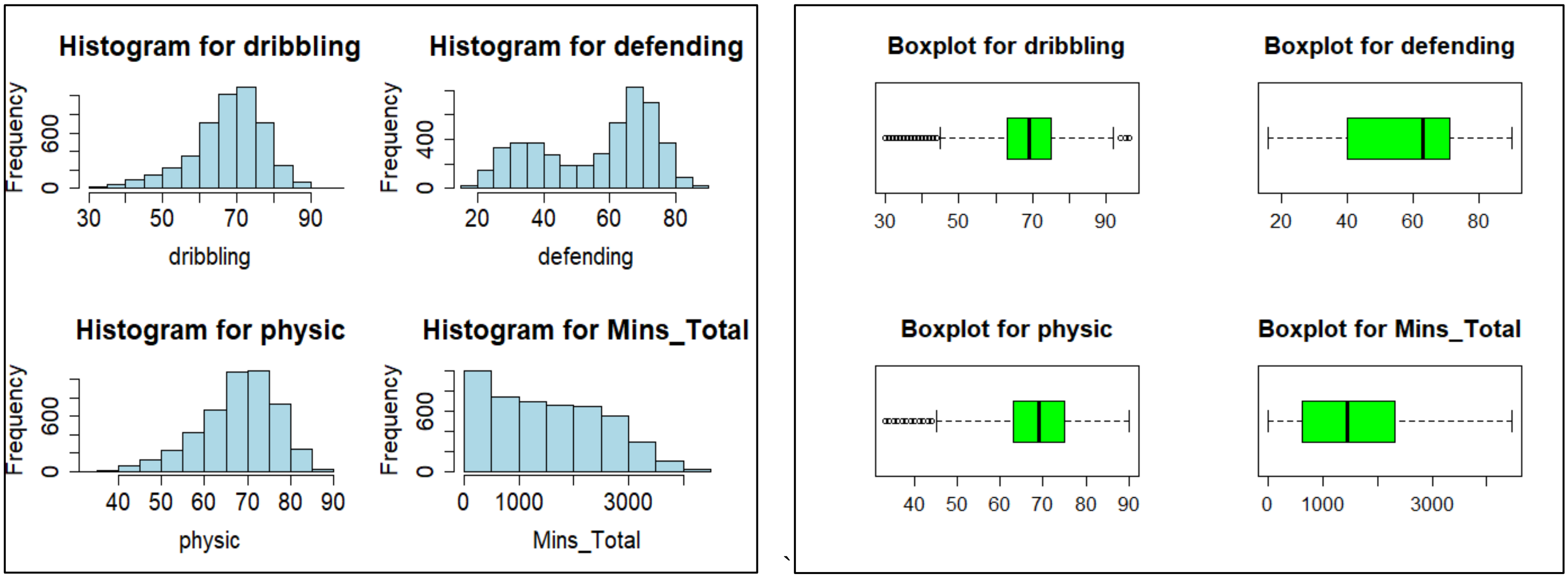
For univariate analysis of non-ordinal numerical variables , histograms and boxplots are plotted to identify outliers. A histogram shows the frequency distribution of the variable. A boxplot shows a 5-point summary of the variable. The left most and right most whiskers of the boxplot show the minimum and the maximum values of the variable respectively. The left outline of the box in the middle shows the first quartile and the right outline shows the 3rd quartile. The thick line in the middle shows the median. Anything that lies outside of the leftmost and rightmost whiskers is considered an outlier by the boxplot. For univariate analysis ordinal numerical variables , bar plots are made which show the number of observations in each rank.

Figure 2: Histograms and Boxplots for age, pace, shooting and passing



1. Age has a slight right-skew as can be seen in the histogram. There aren't too many outliers in age but the same may be treated for better accuracy in the model.
2. Pace, passing, and shooting are all have slight left skewness. All of these have outliers that may need to be treated, particularly pace and passing which have a considerable number of outliers

Figure 3: Histograms and Boxplots for dribbling, defending, physic and Mins_Total



3. In the histograms above (Figure 3), dribbling and physic are the two variables that are slightly right skewed. Both have significant number of outliers. From the boxplot we can see that, compared to all the variables seen so far, the variables defending and Mins_Total seem to have higher variation. They are not distributed normally but also do not seem to have any outliers in them.
4. The plots below (Figure 4) show highly skewed data with large number of outliers in each case. However, barring yellow cards, the attributes are extremely important in considering a player's worth and their extreme values can contain important explanatory powers. Hence it may be worth exploring the behaviour of these variables in the models both with and without outlier treatment.

Figure 4: Histograms and Boxplots for Goals_Total, Assists_Total, Shots_per_game_total and Yellow_Cards_Total

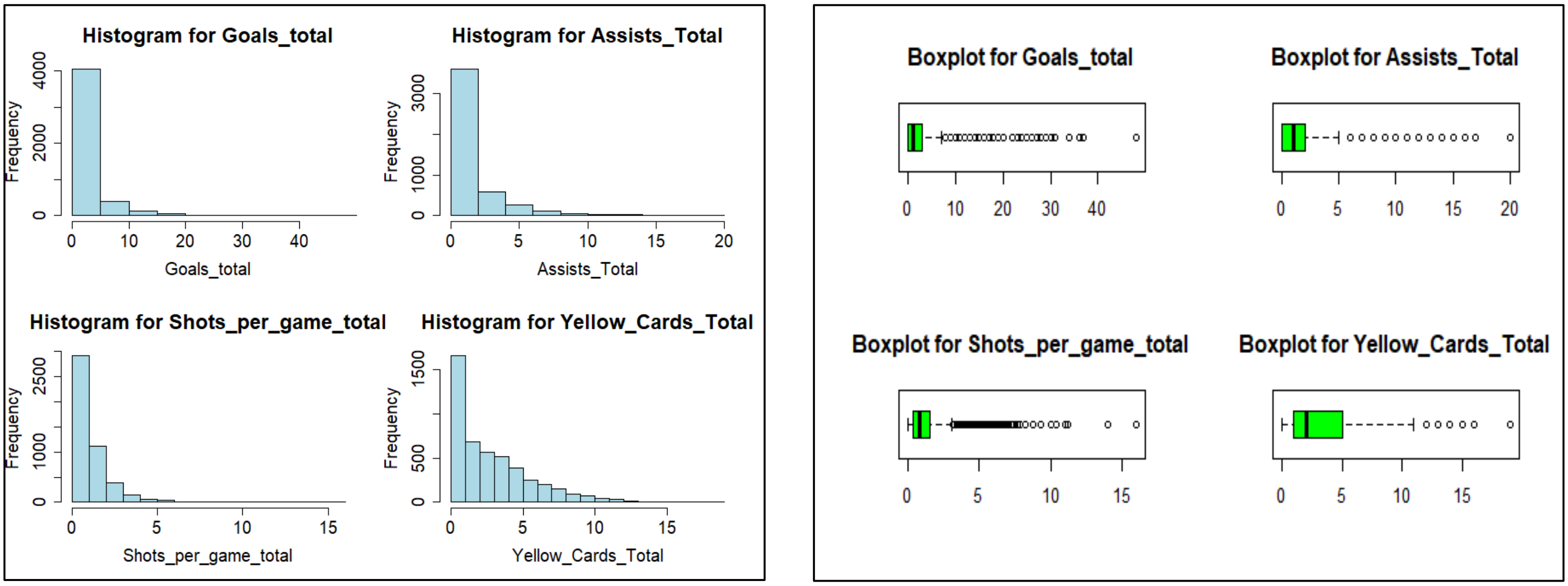
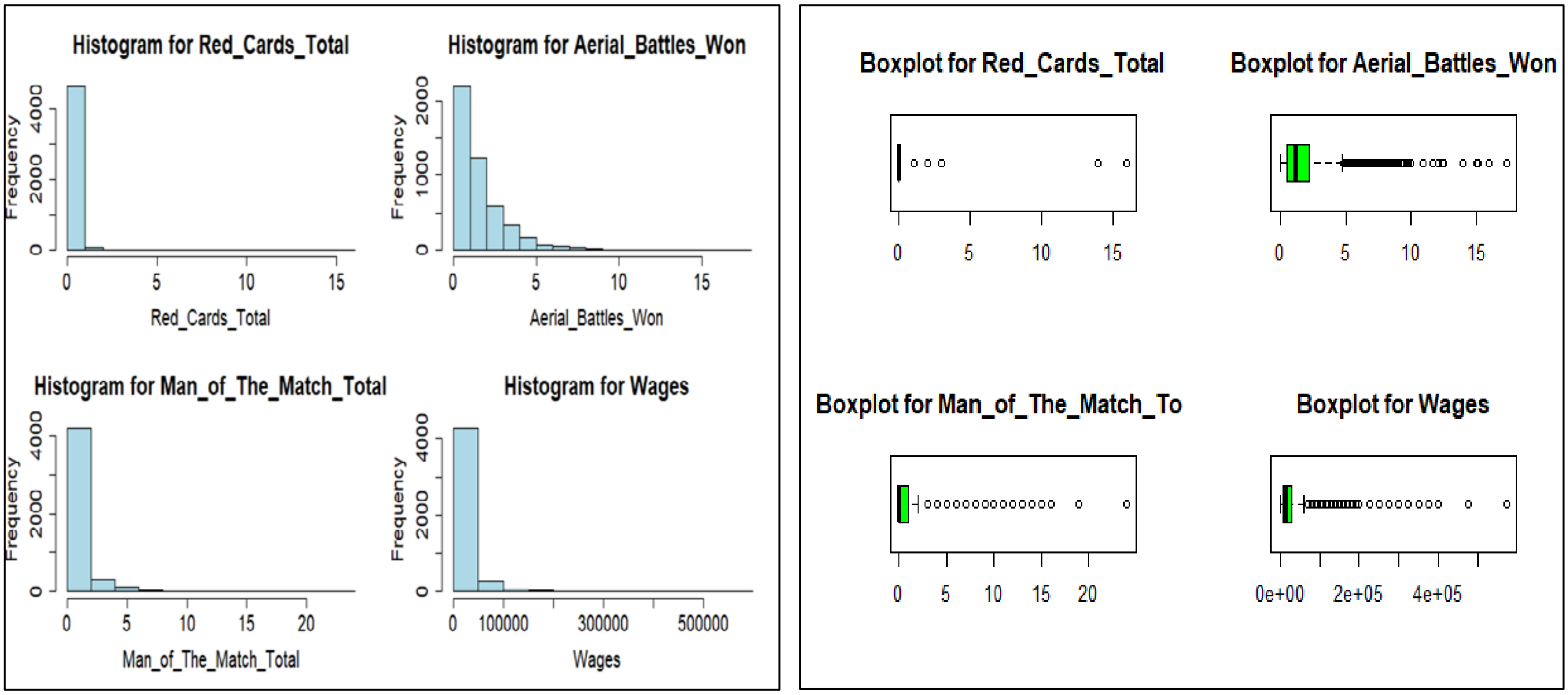


Figure 5: Histograms and Boxplots for Red_Cards_Total, Aerial_Battles_Won_Per_Game_Total and Man_of_The_Match_total



5. Again, the plots above (Figure 5) are extremely right skewed. There are a lot of outliers in each variable. We see can that most of the players did not receive red cards throughout the season. Receiving red cards seems as a highly unlikely event and therefore, the variable may be dropped altogether in the analysis. For all other variables the outliers will need to be treated

Based on the observations above, the outliers in the numerical variables were treated using the following method

1. The Interquartile range (IQR) i.e. the difference between quartile 3 (Q3) and quartile 1(Q1) was calculated for each variable
2. Then the lower limits (LL) and upper limits (UL) for each variable were set equal to Q1 – 1.5*IQR and Q3 + 1.5*IQR respectively.
3. Any value below the lower limit or above the upper limit was considered an outlier and was replaced with the lower limit or the upper limit respectively

However, while treating outliers, two scenarios were considered. In the first scenario, outliers were treated for ALL numeric variables whereas in the other, outlier treatment was done for all numeric variables except Wages, Goals_Total and Assists_Total. This was done considering the relative importance of these variables in explaining the dependent variables.

The figures below show the Univariate Analysis of ordinal numerical variables.

Figure 6:Count of players by Player International Reputation

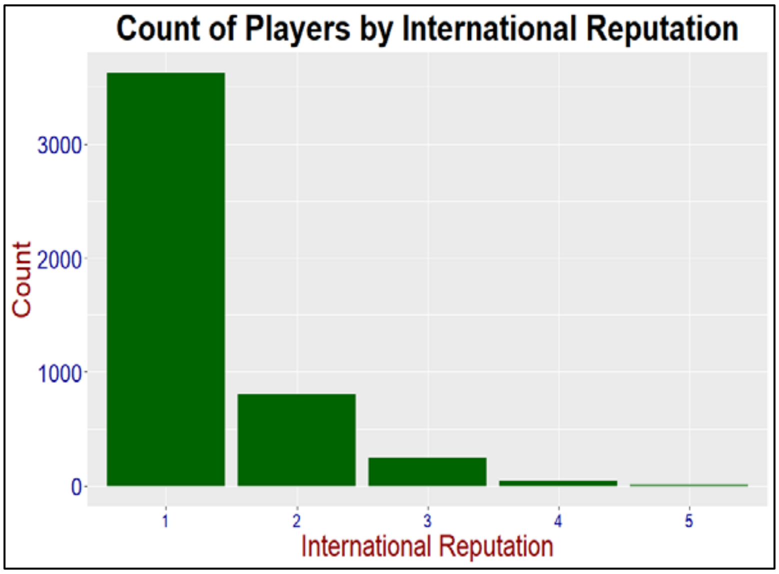


Figure 7: Count of players by Club International Prestige

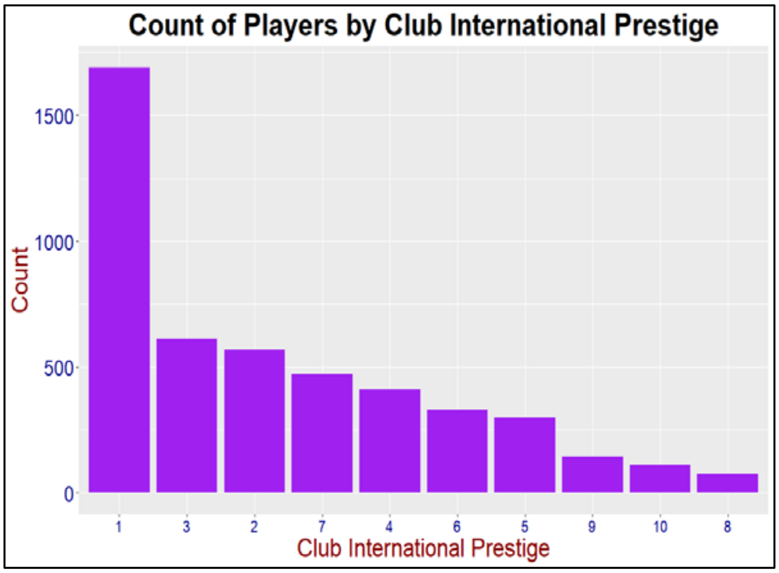


Figure 8: Count of players by Club Domestic Prestige

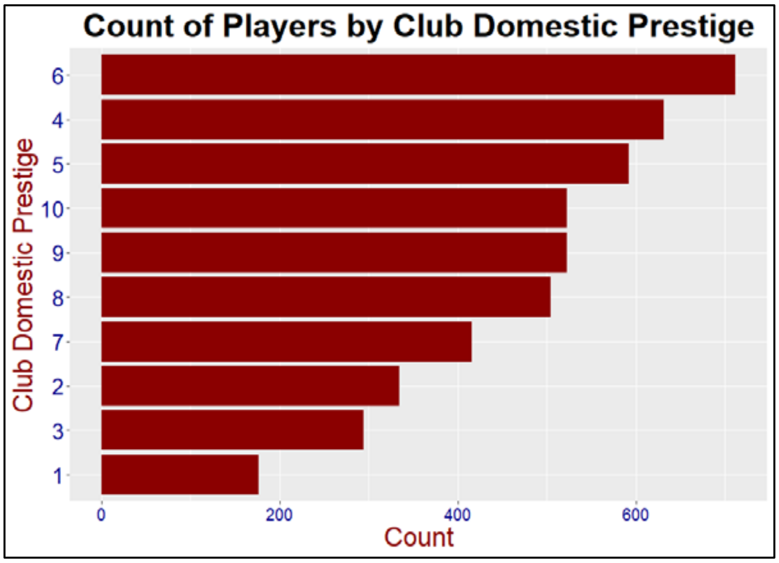
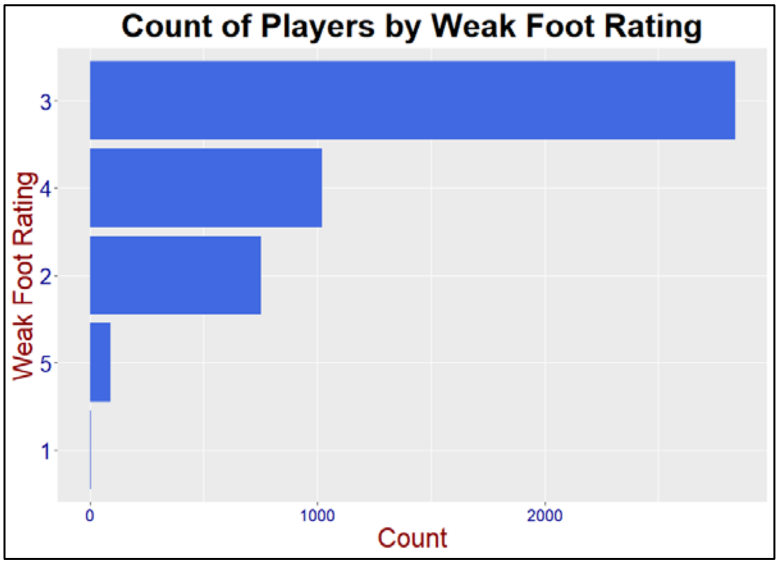


Figure 9: Count of players by weak foot rating



1. International reputation is the reputation of the player internationally, ranked on a scale of 1 to 5, with 5 being the highest reputation level. Majority of the players have an international reputation of 1. Again, this implies that most of the players in the analysis have relatively low repute across the globe. There are only a few players which are reputed highly, and who tend to earn higher wages and sell for higher values (Figure 6)
2. International prestige is the prestige of the football club internationally with 10 being the highest prestige. Multiple clubs may have the same level of international prestige. Most of the players belong to the clubs with an International Prestige of 1 which means majority of the players belong to clubs with a low prestige internationally. This is the case because there are only a few clubs which are consistently able to perform at a high level internationally. (Figure 7)

3. Domestic Prestige is the prestige of the football club within the league that it belongs to. Most players belong to the clubs which have a local prestige that is neither too good nor too bad. (Figure 8)
4. Majority of the players have a skill level 3 when it comes to playing with their weak foot. This means that most of the players are relatively comfortable when made to kick the football with their weaker foot. (Figure 9)

Univariate Analysis – Categorical Variables

Figure 10: Count of players by continent

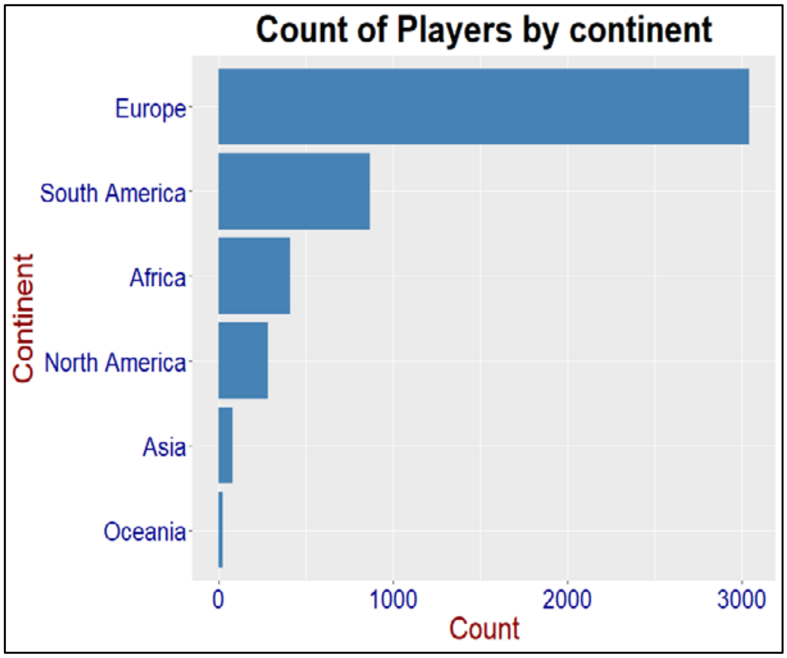


Figure 11: Count of players in Top 10 Nationalities

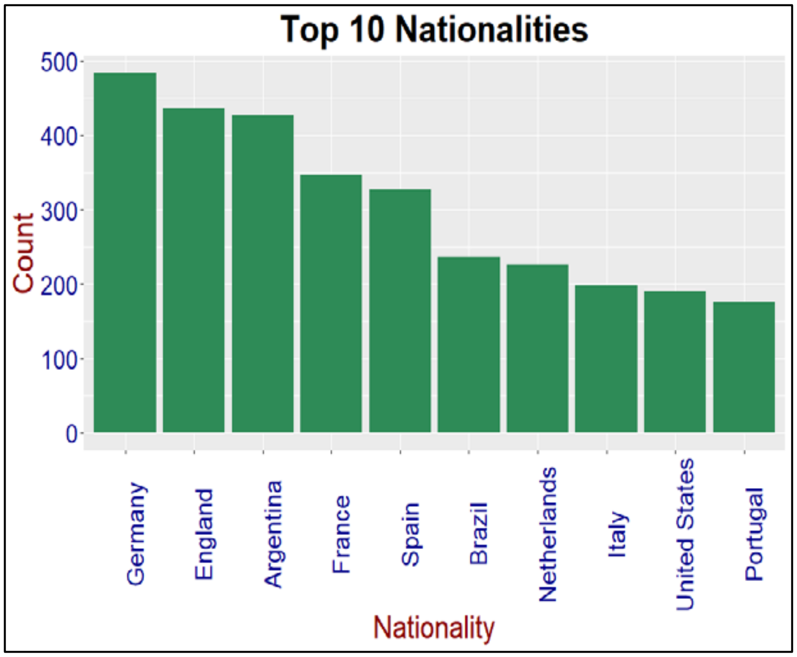


Figure 12: Count of clubs by continent

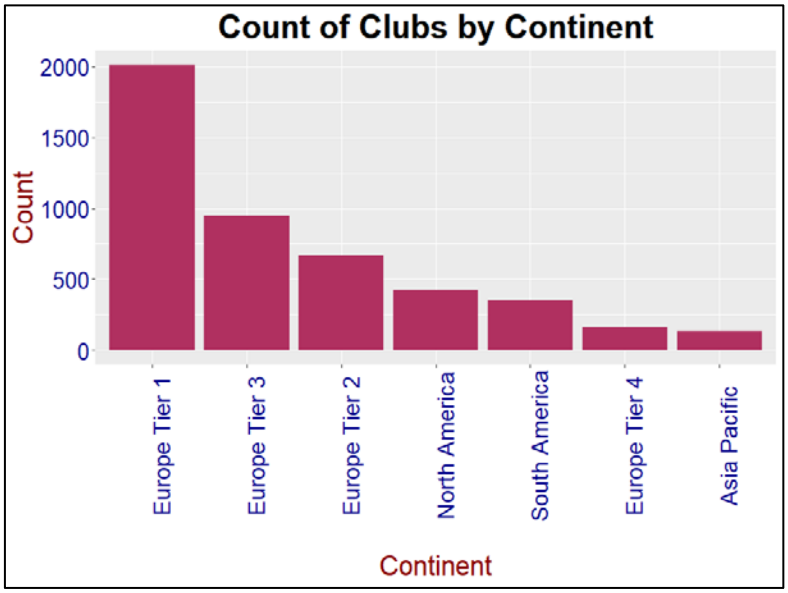


Figure 13: Count of players by preferred foot

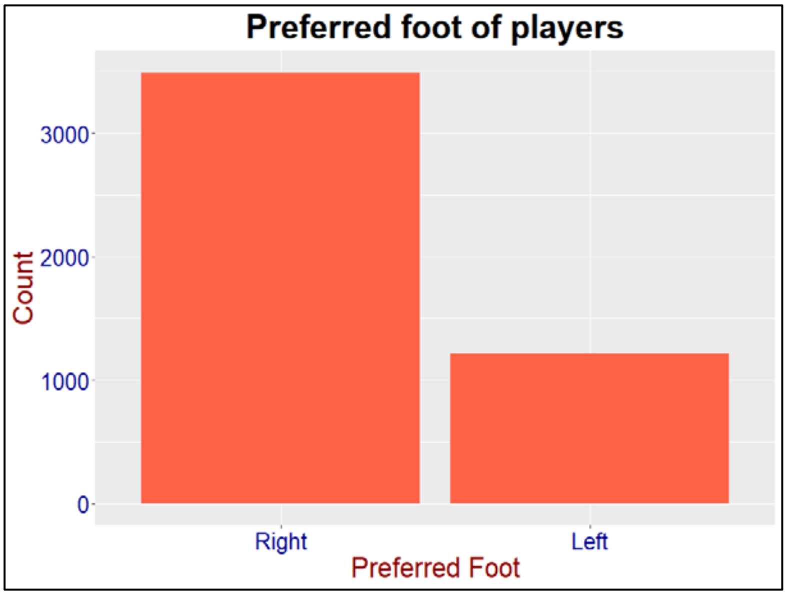
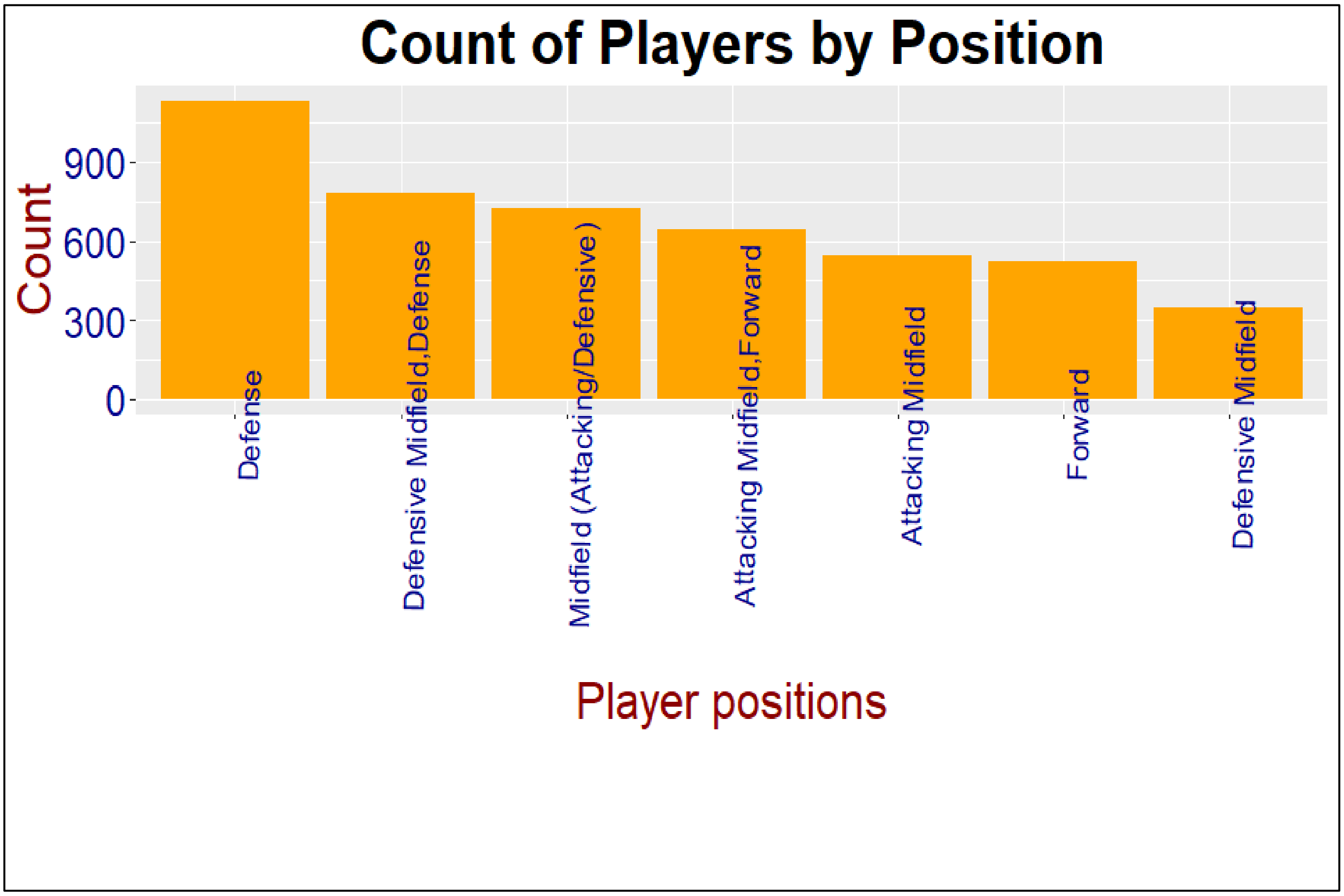


Figure 14: Count of football players by position



The bar plots above give the following information –

1. Most of the football players in the analysis (approx. 64%) are European (Figure 10). This is expected since football is the most popular sport in Europe.
2. From within Europe, most players are German, making up close to 11% of the total players in the analysis (Figure 11). We can see that 3 non-European countries feature in the top 10 nationalities in terms of number of players. These are Argentina, Brazil, and the USA in decreasing order
3. Most of the football clubs in the analysis are also European, as expected. (Figure 12)
4. Coming to player attributes, majority of the players (approximately 75%) are right footed. (Figure 13)
5. Majority of the players (approximately 25%) in the analysis play in a defensive role i.e. either defense or defensive midfield or both. (Figure 14)

Bivariate Analysis – Numerical Variables

Bivariate analysis of numerical variables involves checking the relationship between two numeric variables. A good measure of this relationship is correlation. Correlation is a bivariate analysis that measures the strength of association between two variables and the direction of the relationship without implying any **cause and effect.** In terms of the strength of relationship, the value of the correlation coefficient varies between +1 and -1. Positive Correlation is when the value of one variable increases with respect to another. Negative Correlation is when the value of one variable decreases with respect to another. A value of ± 1 indicates a perfect degree of association between the two variables. **Correlation can be a good initial check for multicollinearity in the data i.e. when two features are highly correlated to each other.**

Depending on the nature of the numeric variables, different measures of correlation can be used as follows:

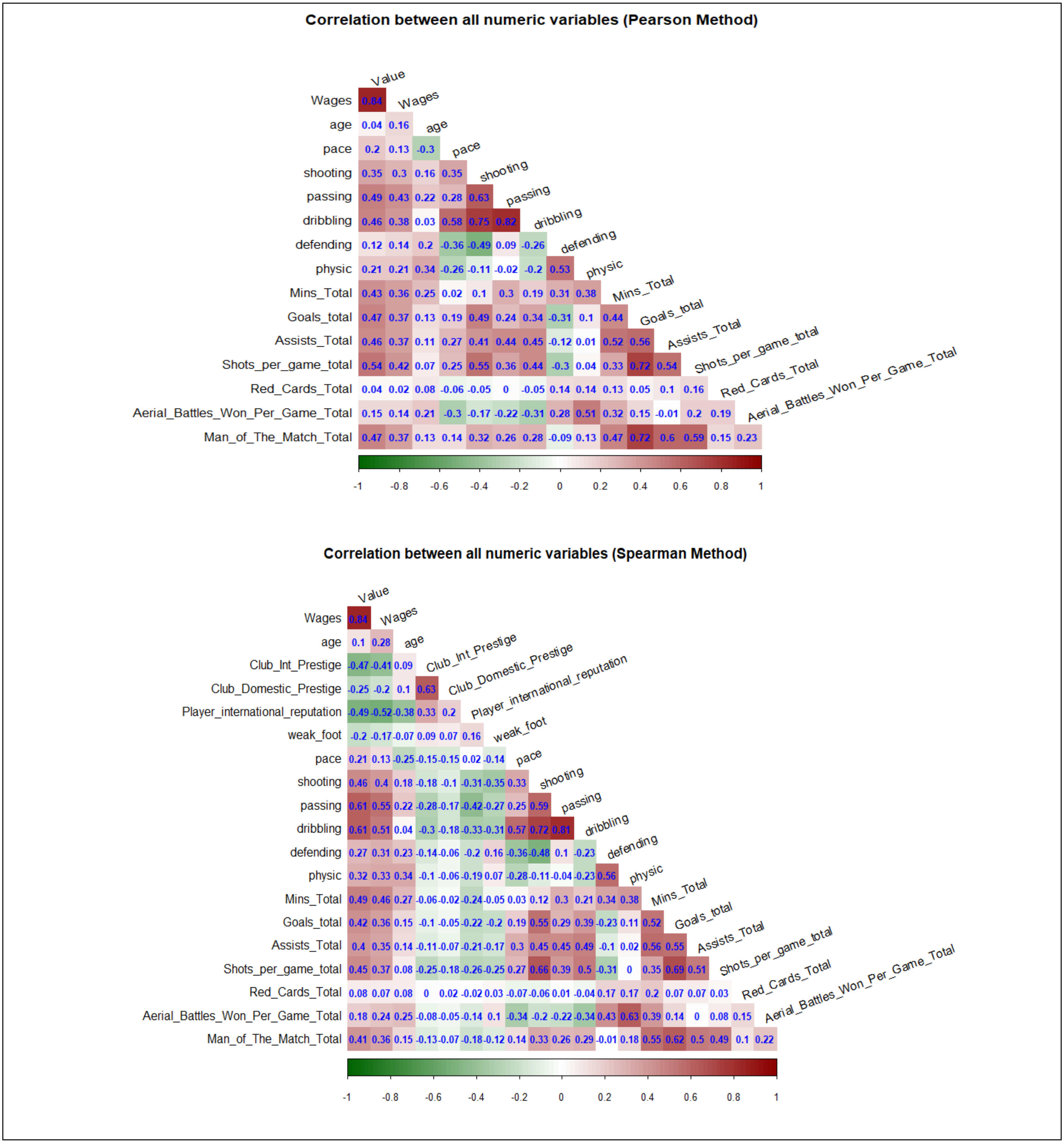
- Continuous vs Continuous: For checking the relationship between 2 continuous variables once can use the **Pearson correlation** which is a measure of the strength of a linear association between two variables

- Continuous vs Ordinal: If one variable is continuous and the other is ordinal, then an appropriate measure of association **is Kendall's coefficient of rank correlation**. If the two variables are denoted by X (continuous) and Y (ordinal), then consider the levels of Y to be numerically coded according to the order of the levels (e.g., assign 1, 2, 3, . . . to the levels). Then Kendall method uses the numerical values of X and the coded numerical values of Y to render a number (coefficient) between –1 and +1 that measures the strength of relationship between X and Y. **If the ordinal variable, Y, has a large number of levels (say, five or six or more), then one may use the Spearman rank correlation coefficient** to measure the strength of association between X and Y. The performance of the Spearman rank correlation coefficient is comparable to that of Kendall's, with the former being somewhat better for large sample sizes (n>30). Since our sample size is large, we can use the Spearman method.

- Ordinal vs Ordinal: **If both variables are ordinal, then an appropriate measure of association is Kendall. If both ordinal variables have a large number of levels, the Spearman rank correlation coefficient can be calculated.** Again, the performance of the Spearman rank correlation coefficient is comparable to that of Kendall's, with the former being somewhat better for large sample sizes (n>30). Since our sample size is large, we can use the Spearman method.

  The figures on the next page show he correlation plots.

Figure 15: Correlation plots



- The figures above shows the correlation matrices between the numeric variables using the Pearson method (top) and between numeric and ordinal as well as between ordinal variables using the Spearman method (bottom)
- Generally, a correlation value greater than 7 can be considered as an indicator of strong correlation between two variables
- Based on the Pearson method, there is a very high correlation between Value and Wages. However, this is a good sign as it suggest that the feature Wages has a strong linear relationship with Value which is our target variable.
- The Pearson method also shows strong correlation between the following variables –

i. Dribbling vs Passing
ii. Dribbling vs Shooting
iii. Goals_total vs Shots_per_game_total
iv. Goals_total vs Man_of_The_Match_total

- When looking at the correlation figures of the Spearman method, one should ignore rows of the matrix containing two continuous variables and focus only on the rows and columns that have atleast one ordinal variable. As such we can see there is no strong correlation of an ordinal variable with any of the other ordinal or non-ordinal or numeric variable.
- Thus, the high correlation between the variables mentioned above could be an indicator of multicollinearity. However, another check for the same is calculation of the Variance Inflation Factor for all the variables
- The VIF for a regression model variable is equal to the ratio of the overall model variance to the variance of a model that includes only that single independent variable.
- Generally, a VIF value of more than 5 indicates high multicollinearity. Below are the VIF results for a basic linear model for Market Value. (VIF values for categorical variables can be ignored)

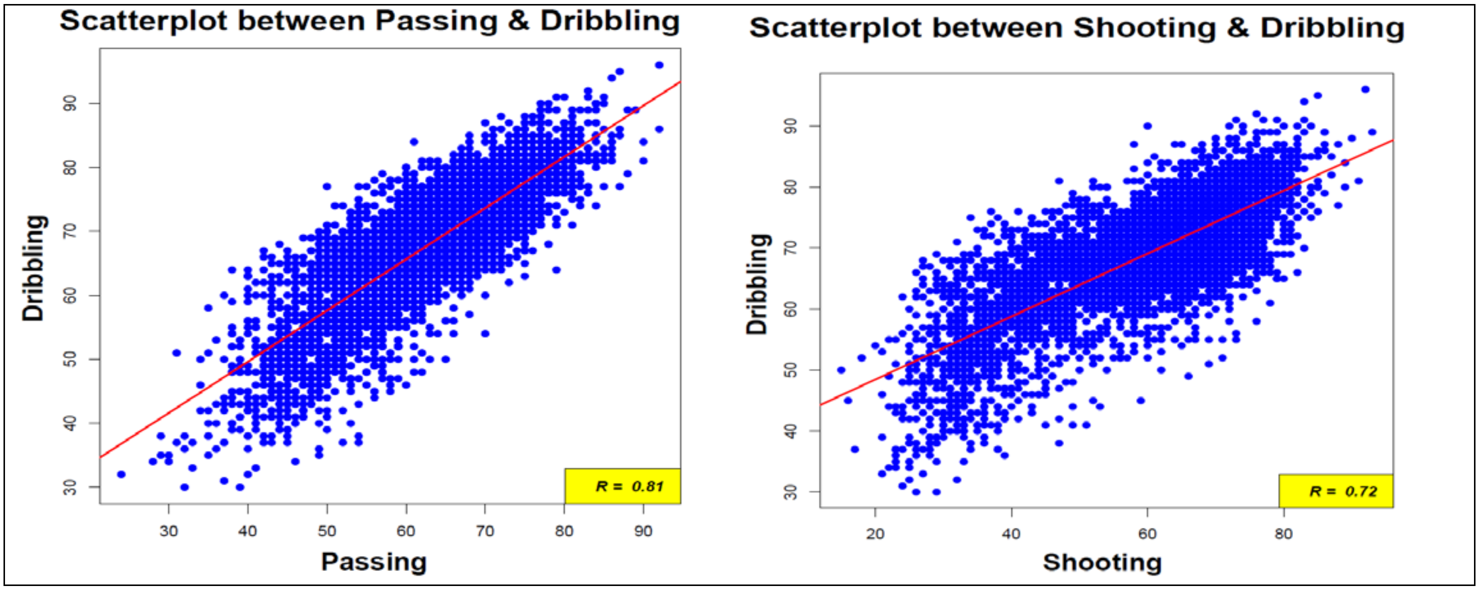
Table 5: VIF values

Variable	VIF	Variable	VIF
age	1.88	Man_of_The_Match_Total	2.64
Club_Int_Prestige	3.69	Player_Continent_Asia	1.42
Club_Domestic_Prestige	2.47	Player_Continent_Europe	3.19
Wages	2.56	Player_Continent_North.America	2.19
Player_international_reputation	2.19	Player_Continent_Oceania	1.06
weak_foot	1.22	Player_Continent_South.America	3.23
pace	2.43	Club_Continent_Europe.Tier.1	12.64
shooting	6.20	Club_Continent_Europe.Tier.2	6.57
passing	7.15	Club_Continent_Europe.Tier.3	8.57
dribbling	7.84	Club_Continent_Europe.Tier.4	2.64
defending	6.96	Club_Continent_North.America	5.25
physic	2.44	Club_Continent_South.America	4.64
Mins_Total	3.18	preferred_foot_Right	1.12
Goals_total	3.86	Position_Final_Attacking.Midfield.Forward	2.22
Assists_Total	2.32	Position_Final_Defense	6.62
Shots_per_game_total	3.18	Position_Final_Defensive.Midfield	2.30
Yellow_Cards_Total	2.02	Position_Final_Defensive.Midfield.Defense	3.82
Red_Cards_Total	1.15	Position_Final_Forward	2.75
Aerial_Battles_Won_Per_Game_Total	2.05	Position_Final_Midfield..Attacking.Defensive	2.47

- Based on the VIF values dribbling, passing, shooting, and defending are variables causing multicollinearity.

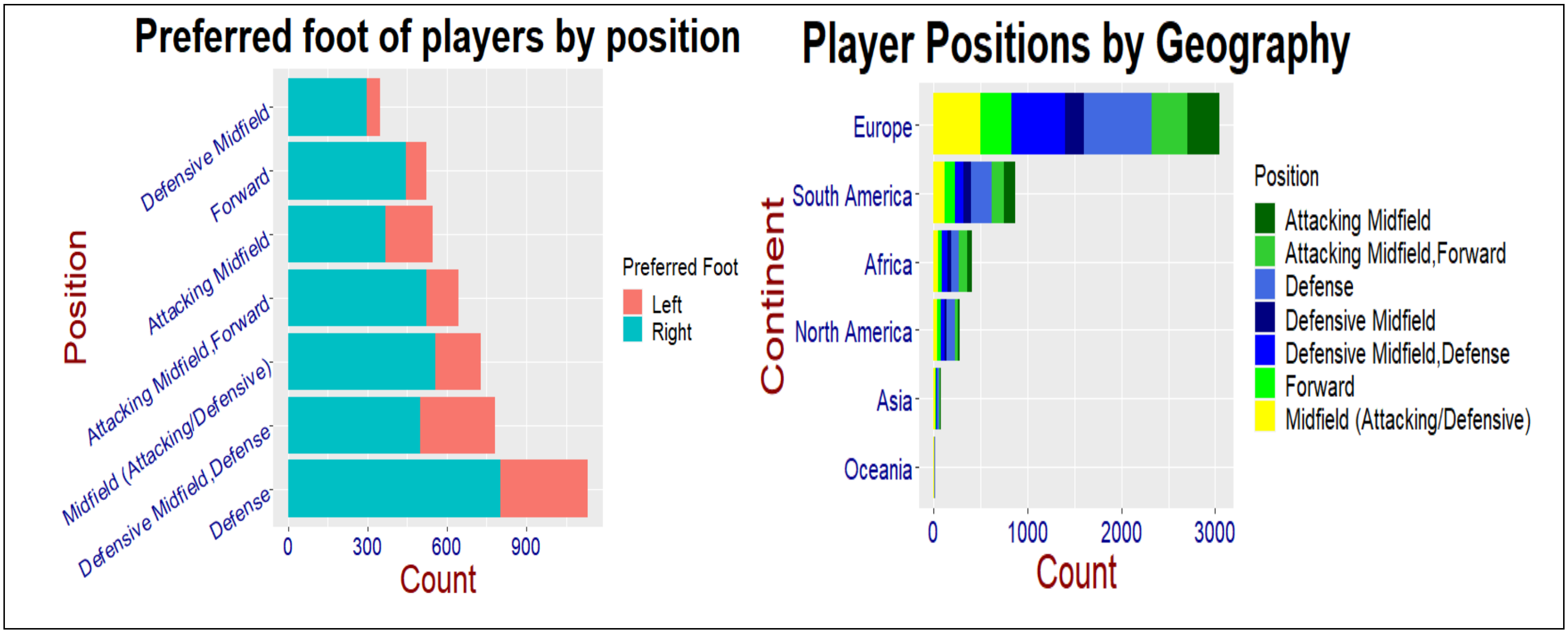
- Thus, dribbling, passing, and shooting are the variables common in the correlation plots and the VIF tables, of which dribbling is the one with high correlation with both the other variables.
- As such, dropping dribbling or another variable between shooting and passing may help to deal with the problem of multicollinearity and dimension reduction techniques are not required.

Figure 16: Scatterplots of dribbling with passing and shooting



Bivariate Analysis – Categorical and Ordinal Variables

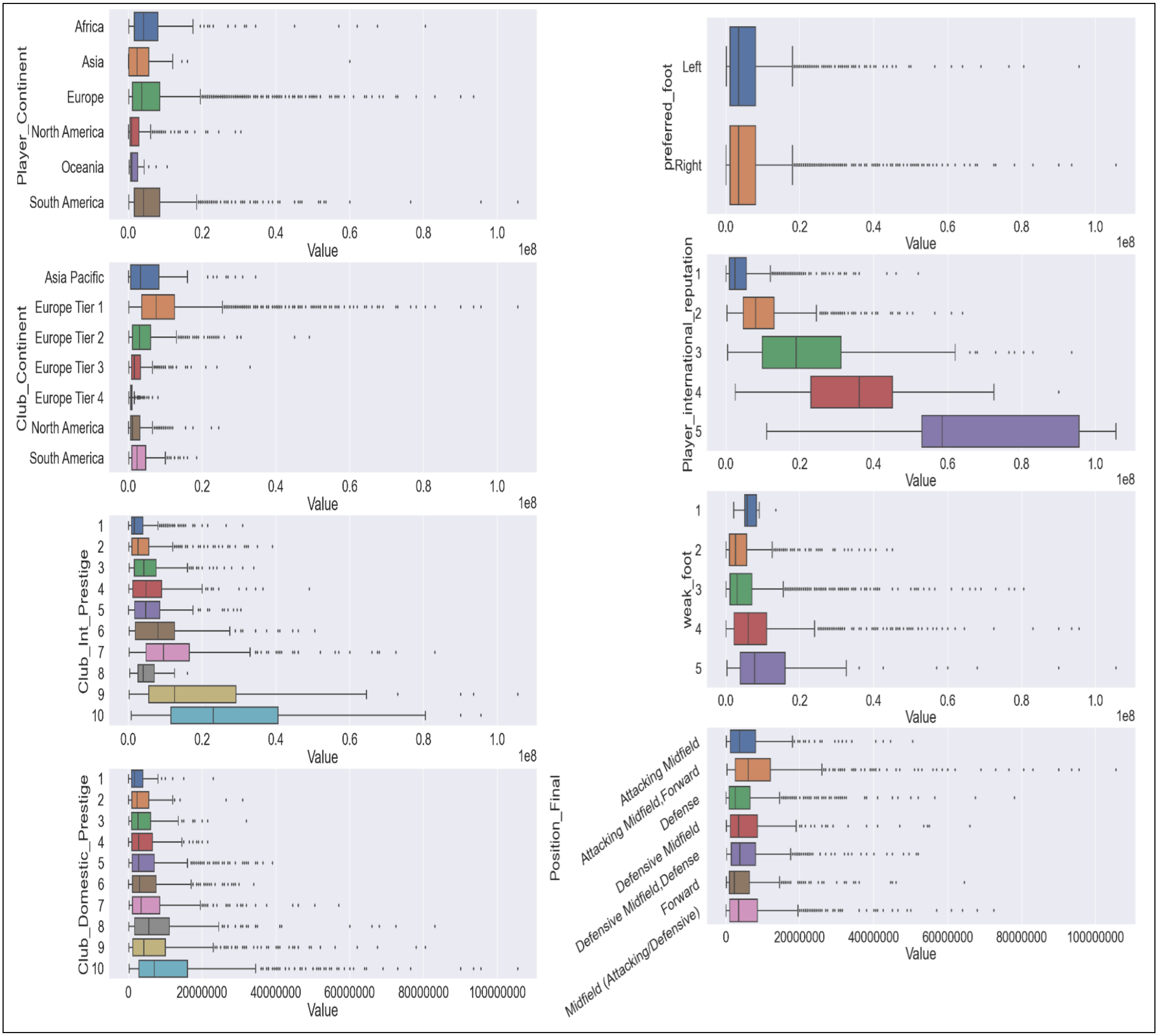
Figure 17: Stacked bar chart for categorical variables



- The above stacked bar chart on the left plots the preference of foot amongst the different positions at which the players in the analysis play. While overall the right foot is preferred, the aim was to check if at any position, there is a preference for the left foot. However, that does not seem to the case, with the right foot being the predominant foot in every position.

- The plot on the right above aims to show the distribution of player positions across different geographies. The defensive positions are coloured blue, attacking positions are coloured green and midfield (neutral) are coloured yellow. While in most continents, defensive players predominate the attacking players, Africa is the only geography where that trend is reversed with more players taking an attacking role rather than a defensive one

Figure 18: Categorical and Ordinal variables vs Market Value



Below insights can be derived from these graphs

- The median market value of players from North America and Oceania seems to be lower than that of players from other continents
- A player playing in Europe’s Tier 1 tends to have a higher market value than players from other geographies, as may have been expected.
- There is an increasing trend in the median market value with respect to an increase in both the Club_Int_Prestige and Club_Domestic_Prestige. Surprisingly though, median market value for players in clubs with an international prestige rating of 8 is less than that of those with a rating of 6 and 7. Also these median values jump drastically for the players belonging to clubs with international prestige of 9 and 10. Also there is very high variation in the market value for players belonging to clubs with an international prestige of 9 and 10.The variation cause by the clubs domestic prestige on the other hand is lower.
- Preferred_foot and Position_Final do not seem to cause a lot of variation in the market value although it seems like full-fledged attacking players tend to be of greater value and earn marginally more than defensive players as they are directly involved in scoring goals. This may imply that scoring a goal is perceived to have greater importance than preventing a goal from being scored.
- Amongst all factor variables, the biggest variation in the market value is caused by the player’s international reputation. The median market value for players rated 5 internationally is much higher than those of player with ratings of 4 and below, as may have been expected. But the fluctuation in the market value for players with an international reputation of 5 is also the highest amongst all reputation levels. This may be because there are only a few players (<10) with an international reputation of 5
- Having looked at all the graphs above, Player_international_reputation seems to be a good differentiating factor for Market Value with a gradual increase in the Value with reputation.
- Looking at Club_Int_Prestige, market value of players in clubs with a rating of 9 and 10 stand out from the others.
- For all other features, the market values are more or less uniform across the categories which does not give much insight into the target variable.

EDA Summary

Below are the key points to be noted based on the exploratory data analysis performed :

- Most of the continuous numerical variables are not distributed normally and have outliers in them. These outliers may need to be treated before making the model. However, some of the variables containing outliers, like Goals_Total are important, and the outliers may actually be helpful in predicting the market value of the corresponding players. Thus, it may be worthwhile to make a model without treating outliers of all variables. It is important to note that if outlier treatment is performed, it will only be done on continuous numeric variables and not the ordinal numeric variables

- Looking at the barplots for the categorical variables, most of the players as well as clubs belong to Europe. As such it may help to reduce the levels of categories by clubbing the categories with lower counts into a single category
- The correlation plots gives us important insights, the most noteworthy being the high correlation between Wages and Market Value. This is beneficial for the model as it suggests that the feature Wages has strong explanatory power about the Market Value and is likely to be an important variable in the model.
- The high correlation and VIF of dribbling, passing, and shooting were suggestive of multicollinearity which may have to be dealt with when making the model. However, with dribbling being common variable, dropping it may be sufficient to deal with multicollinearity
- Finally, looking at the behaviour of the ordinal variables with respect to the target variable, we can see that the Player's International reputation is a good distinguisher of market value.

6. Modelling Approach

Step 1 : Identifying the models to be made

When deciding the machine learning algorithms to be used, two things need to be taken into consideration – first is whether the target variable is numeric or categorical and second whether the purpose of the model is just to make a prediction or to also see the importance of different features used while making the prediction . Considering that the response variable in consideration is of numerical nature and we also want to know which variables contribute the most in making the prediction, the models suitable for use are:

i. Multiple Linear Regression (MLR)
ii. Lasso Regression
iii. Ridge Regression
iv. Classification and Regression Tree (CART)
v. Random Forest (RF)
vi. Gradient Boosting Machine (GBM)
vii. Extreme Gradient Boost (XGB)

Based on our EDA results , it may be worthwhile to build the model on 3 different versions of the data –

i. The original data without any outliers treated
ii. The data with outliers treated for all variables except for Goals_Total, Assists_Total and Wages
iii. The data with outliers treated for all variables.

While outliers are generally not desired in the data, based on the knowledge of the sport , for our models , certain players with extremely high values will be outliers. However, giving them the outlier treatment by fixing the upper (or lower) limit , as defined previously , may actually hamper the ability of the model to predict market values for expensive players. An argument could be made to make separate models for the higher valued players but due to lack of data of points for such cases , that approach is not feasible.

Step 2 : Splitting the data into train and test sets.

The training set is used for building a predictive model. This is the data using which the model will learn how to make the predictions. The testing set acts as a previously unseen data that is used to check the performance of the model. When using the test data set , the model makes a prediction for the target value using all the features and then compares its predictions against the actual value of the target variable. For our model , data has been divided in a ratio of 70:30 where 70% of the data is used as the training dataset and the remaining 30% is reserved for the test data set. The method of sampling used it random sampling. While creating the split, it is important to set a seed value (in this case 1234) so that every time we create a split for a different model, the training and testing datasets are identical.

Step 3 : Making the model and cross validating.

For the purpose of making the model , the PyCaret library in Python was used which allows to build a pipeline for machine learning model. This pipeline allows to configure several options mentioned below:

1. Train Size: This option is used to specify the ratio in which to split the data into train and test sets.
2. Target: This option is used to specify the target variable.
3. Session id: This is similar to setting a seed value as mentioned above.
4. Categorical/numeric/ordinal features: This is used to specify the type of the different features in the dataset so that they are used in the correct format when making the model.
5. Normalize: This option is used to normalize the non-ordinal numeric data. Since the numeric variables in the analysis have different measurement scales, therefore the variables with a larger magnitude may have a dominating effect on the response variable. For example, Total_mins_played is measure in minutes and goes beyond 1000 mins whereas passing is measured on a scale of 100. Normalization is done to bring all the variables to the same scale.
6. Normalize method: This option is used to specify the method of normalization. **The normalization used for our model is z-score normalization**. The technique involves transforming the values of each feature so that all of them are on a common scale where the average value for each independent variable equals zero and standard deviation equals one.

$$X_i = \frac{\mathrm{Xi} - \mu}{\sigma}$$

where x_i is each individual observation for every feature x

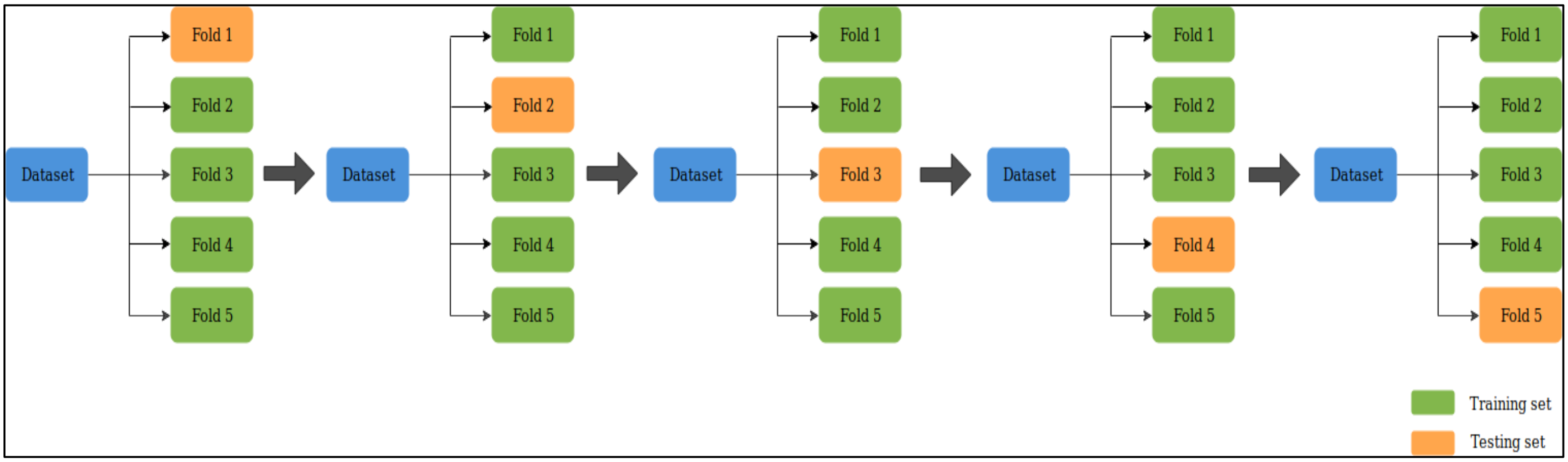
7. Combine Rare Levels: As noted during EDA, there were some categorical variables whereby majority of the observations fell under one level with the other levels having only a few observations. This option is used to combine the less frequent levels together.
8. Rare Level Threshold: This option is used to specify the threshold for considering a level as rare. For example, if the threshold is 0.1, then all levels which make up 10% or less of the total observations in that variable will be clubbed together.
9. Fold Strategy: This option is used to perform cross validation and to specify the type of cross validation to be used. Cross-validation is a statistical validation technique used in machine learning to assess the performance of a machine learning model. It uses the subset of the dataset, trains on it then assess the model performance using the complementary subset of the data-set which is not

used for training. It serves as an assurance that the model is correctly capturing patterns from the data, without considering noise from the data[8].

There are many different methods of cross validation. By default , PyCaret always performs K-fold cross validation (which will also be used for our model). K-Fold CV is where a given data set is split into *K* number of sections/folds where each fold is used as a testing set at some point. Let's take the scenario of 5-Fold cross validation (K=5). Here, the data set is split into 5 folds. In the first iteration, the first fold is used to test the model and the rest are used to train the model. In the second iteration, 2nd fold is used as the testing set while the rest serve as the training set. This process is repeated until each fold of the 5 folds have been used as the testing set[9]. It is important to note here that each fold used for testing in cross validation is a subset of the train data itself and is different from the test data created earlier during the train and test split. After testing on each fold, accuracy measures are calculated, and the final accuracy of the train data is the average of the results of all the folds tested.

10. Fold : Used to specify the number of folds to be used in cross validation

Figure 19: K-Fold Cross Validation (K=5)



After having set up the PyCaret environment, we will make all the models listed above with the same settings

Step 4 : Checking model performance and comparing all models.

After making all the models the final step is to check the accuracy of the model. This is done by running the same model on the test data set which was created earlier and then comparing the predicted market values generated by the model for the test data with the actual market values in the test data. In order the check the accuracy of the models the following metrics were used:

- Root Mean Square Error (RMSE) = $\frac{\sqrt{\Sigma(\mathrm{Yi} - \hat{\mathrm{Y}})^2}}{n}$
- Mean Absolute Percentage Error (MAPE) = $((|\,\mathrm{Yi} - \hat{\mathrm{Y}}\,|\ \mathrm{x}\ 100)/Yi)/n$

Where Yi is the observed (actual) value , Ŷ is the predicted value and n is the sample size

Finally, after all the models have been made and the accuracy measures have been calculated , the model with the least error rate given by RMSE and MAPE will be chosen as the best model.

Model 1 - Multiple Linear Regression

Multiple linear regression is a statistical method that aims to predict a response variable by creating a linear relationship with the predictor variables. An MLR model has an equation of the following type –

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \varepsilon$$

where,

Y = Response Variable
X_i = Predictor Variables
β_0 = Intercept Coefficient
β_i = Slope Coefficients
ε = Error term

The value of the slope coefficient shows the percentage change in the response variable for a unit change in the corresponding predictor variable. The intercept shows the value of the response variable when all the predictor variables are 0. In other words, it shows the minimum possible value of the response variable as predicted by the mode. The error term is the sum of all errors, i.e. the sum of the variance between the predicted value of Y and the actual value of Y. The aim of the MLR model is to choose all the βs such that the error is minimized. This technique of regression is called Ordinary Least Square (OLS) regression and the coefficients so estimated are called Best Linear Unbiased Estimators (BLUE)

Let $\hat{Y}$ be the predicted value, Y be the actual value and $\bar{Y}$ be the mean value of the response variable. The total variation in the data set is given by the difference between Y and $\bar{Y}$ for each value of Y. The squared sum of these differences is called the **Total Sum of Squares** or **TSS.** The squared sum of difference between $\hat{Y}$ and $\bar{Y}$ is called the **Regression Sum of Squares** or **RSS.** It is the proportion of the total variation explained by the regression model. Finally, the remaining variation, not explained by the regression is equal to the squared sum of difference between $\hat{Y}$ and Y and is called the **Residual/Error Sum of Square** or **ESS**.

Thus, TSS = RSS + ESS

or

$$\sum (Y_i - \bar{Y})^2 = \sum(\hat{Y} - \bar{Y})^2 + \sum(\hat{Y} - Y)^2$$

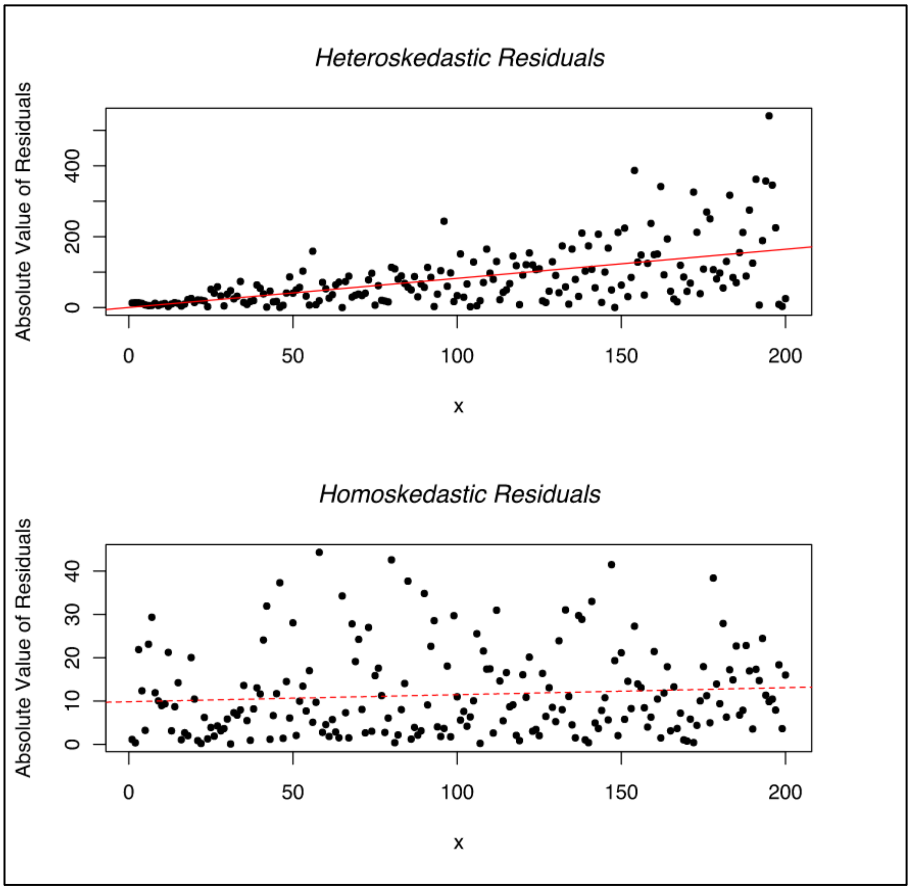
Thus, the aim of the regression is to minimize the ESS and maximize the RSS so that the predicted value is as close as possible to the actual value of the dependent variable

The reason we take the squares of the differences instead of the actual values is that , if we take the actual values , the errors will be cancelled out and the SSE will be 0.

The most significant feature of an MLR model is that the coefficients can be used to determine the magnitude of change in the dependent variable as a result of a unit change in the independent variable. That is, if the coefficient of a feature , say X , is β and the value of X increases by 1 unit then the value of the target variable will increase by β units. However, to rely on the slope coefficients, the MLR model must satisfy the following assumptions:

1. Linearity – There should a linear relationship between the dependent variable and each feature.
2. No Endogeneity – ε_t = 0 i.e. the covariance (or correlation) between the error and each independent variable should be 0. If there is such a correlation it means the error is being explained by one of the independent variables.
3. No autocorrelation – There should be no correlation between errors. This can be checked using the Durbin-Watson test which checks for the null hypothesis that there is no autocorrelation between the errors.
4. No multicollinearity – There should be no correlation between the predictor variables.
5. Normality and homoscedasticity –The error terms should be normally distributed .Homoscedasticity means the error term should have equal variance from the mean i.e. the variance of error terms should be homogenous. The normality of residuals can be checked using the Shapiro-Wilkinson test on error terms which checks for null hypothesis that the errors are normally distributed. The homoscedasticity of errors can be checked by drawing a residual (error) vs fitted (predicted) value graph. The below figure shows the difference between homoscedastic and heteroscedastic errors

Figure 20: Heteroscedastic vs Homoscedastic errors[10]



It is important to note that these assumptions are required to hold true ONLY IF the interpretation of the slope coefficients is important. This is because if the assumptions are violated, then the z/t value of each variable , which

tests the statistical significance of each independent variable (i.e. if the independent variable is different from 0) , is not reliable and hence the value of the coefficient generated is not reliable. Thus, if any of these assumptions is violated then the forecasts, confidence intervals, and scientific insights yielded by a regression model may be (at best) inefficient or (at worst) seriously biased or misleading. A multiple linear regression model can still be used to solely for prediction purposes even if the assumptions do not hold true, but one should not rely on heavily on those predictions and should attempt using other machine learning models for prediction purposes

Another thing to be considered here is that since we are aiming to use the coefficients for their interpretations, therefore **normalization of the numeric variables should NOT be performed.** This is because, if the variables are normalized , then the coefficients will also be normalized and hence will not give a true magnitude of change in the dependent variable.

Also, MLR takes inputs in numeric format only. Therefore, all factor variables in the model will have to be converted into numeric variables. This can be done by one – hot encoding. In this method, n-1 dummy variables are created for each factor variable containing n levels. For example, the variable Player_Continent has 6 levels. Under one-hot encoding, 5 new dummy variables will be created. The reason n-1 variables are created is to avoid multicollinearity in the model (discussed later under MLR). Each dummy variable has a value of 0 or 1 where 1 indicates that the particular dummy variable was the actual observation amongst all other dummies, for that particular row of observation. Thus, if the Player_Continent in a particular row is “Europe” then a dummy variable Player_Contintent_Europe will be created with a value 1 in it and all other dummy variables like Player_Continent_Oceania etc. will have 0s

Below is the list of all non-normalized MLR models made (these models were built in R and not PyCaret to check for assumptions) :-

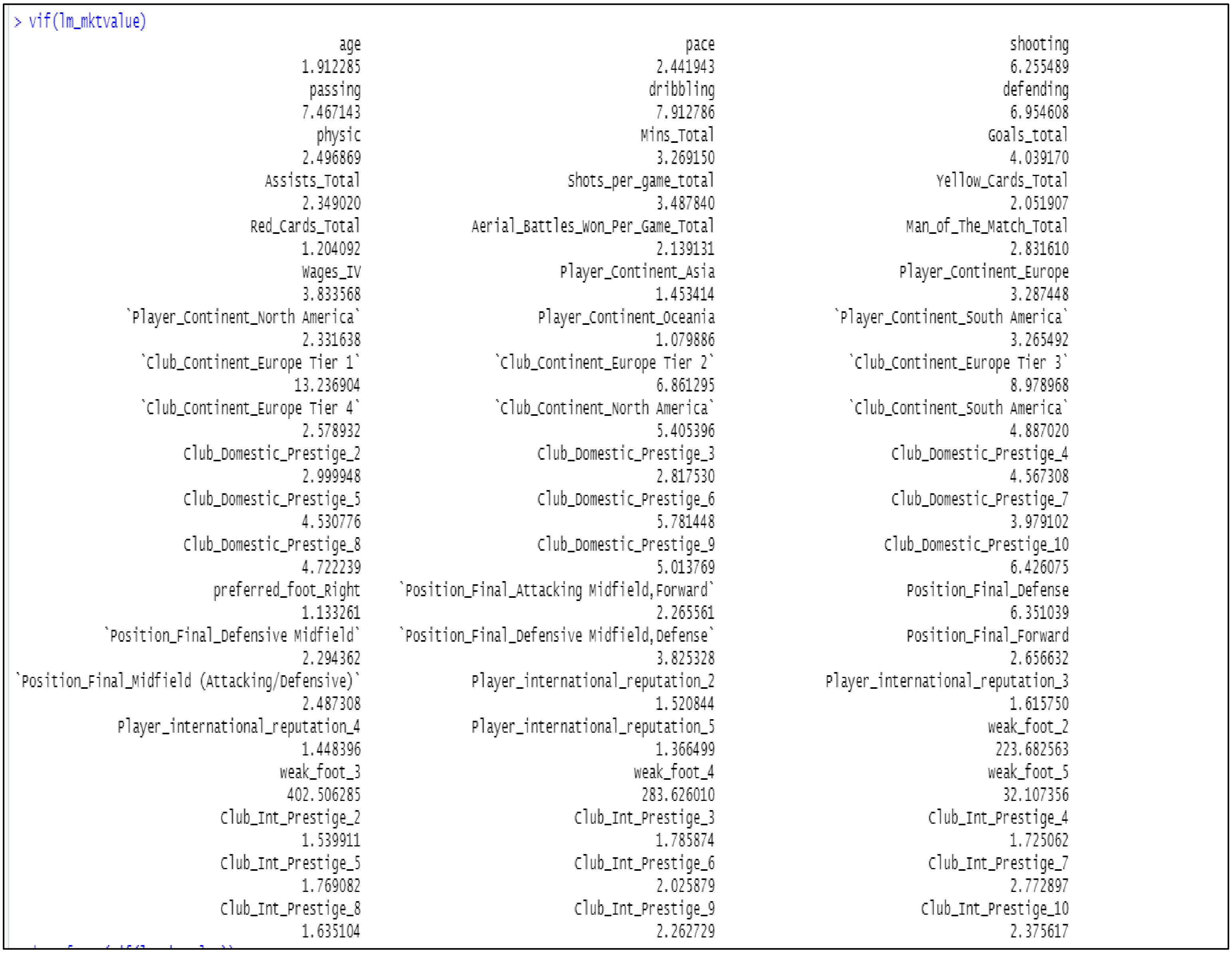
I – MLR for Market Value without outlier treatment
II – MLR for Market Value with outlier treatment for all variables except for Goals_Total, Assists_Total, and Wages
III – MLR for Market Value with outlier treatment for all variables

<u>Model I - Multiple Linear Regression for Market Value without outlier treatment</u>

The model was first made using all the variables, without any outlier treatment. Then, to check for multicollinearity, the variance inflation factor was calculated for each variable. The variable having the highest VIF greater than 5 was dropped. This process was repeated until VIF for all variables was less than 5. While removing variables , care should be taken not to remove the dummy variable representing a single level of a factor variable (example weak_foot3). This is because if we remove a level from the model, then when we use the model in the future on unseen data that includes the removed level, the model will be unable to make a prediction as it did not learn that level of the variable when the model was being made. The following variables were removed based on VIF :

- Dribbling
- Shooting

Figure 21: Calculation of VIF for model I



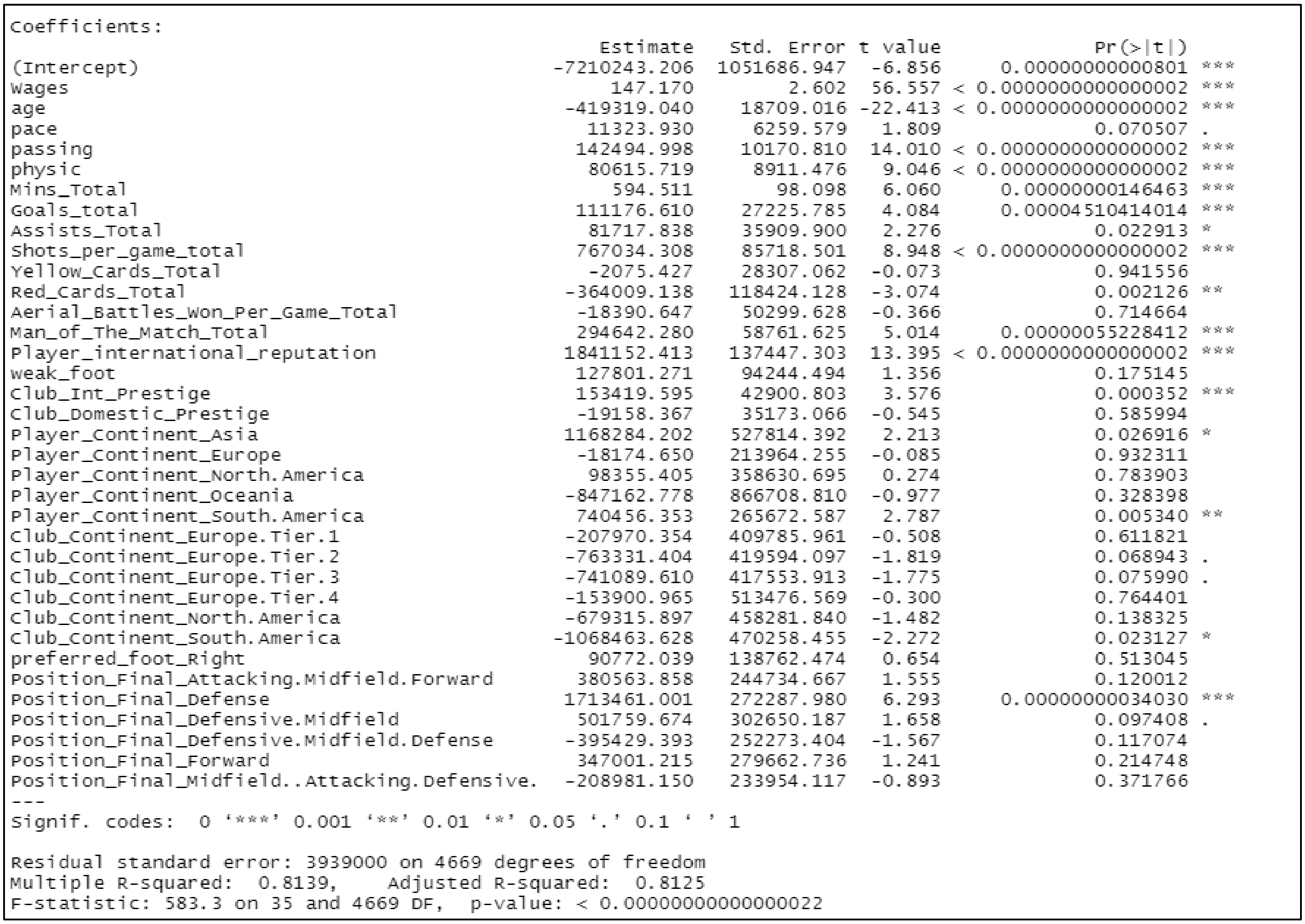
```
> vif(lm_mktvalue)
                                      age                                      pace                                  shooting
                                 1.912285                                  2.441943                                  6.255489
                                  passing                                 dribbling                                 defending
                                 7.467143                                  7.912786                                  6.954608
                                   physic                                 Mins_Total                                Goals_total
                                 2.496869                                  3.269150                                  4.039170
                            Assists_Total                      Shots_per_game_total                         Yellow_Cards_Total
                                 2.349020                                  3.487840                                  2.051907
                           Red_Cards_Total          Aerial_Battles_Won_Per_Game_Total                      Man_of_The_Match_Total
                                 1.204092                                  2.139131                                  2.831610
                                 Wages_IV                      Player_Continent_Asia                     Player_Continent_Europe
                                 3.833568                                  1.453414                                  3.287448
         `Player_Continent_North America`                   Player_Continent_Oceania           `Player_Continent_South America`
                                 2.331638                                  1.079886                                  3.265492
           `Club_Continent_Europe Tier 1`             `Club_Continent_Europe Tier 2`              `Club_Continent_Europe Tier 3`
                                13.236904                                  6.861295                                  8.978968
           `Club_Continent_Europe Tier 4`            `Club_Continent_North America`              `Club_Continent_South America`
                                 2.578932                                  5.405396                                  4.887020
                    Club_Domestic_Prestige_2                   Club_Domestic_Prestige_3                  Club_Domestic_Prestige_4
                                 2.999948                                  2.817530                                  4.567308
                    Club_Domestic_Prestige_5                   Club_Domestic_Prestige_6                  Club_Domestic_Prestige_7
                                 4.530776                                  5.781448                                  3.979102
                    Club_Domestic_Prestige_8                   Club_Domestic_Prestige_9                 Club_Domestic_Prestige_10
                                 4.722239                                  5.013769                                  6.426075
                      preferred_foot_Right `Position_Final_Attacking Midfield,Forward`                     Position_Final_Defense
                                 1.133261                                  2.265561                                  6.351039
      `Position_Final_Defensive Midfield`  `Position_Final_Defensive Midfield,Defense`                     Position_Final_Forward
                                 2.294362                                  3.825328                                  2.656632
`Position_Final_Midfield (Attacking/Defensive)`         Player_international_reputation_2          Player_international_reputation_3
                                 2.487308                                  1.520844                                  1.615750
        Player_international_reputation_4          Player_international_reputation_5                                 weak_foot_2
                                 1.448396                                  1.366499                                223.682563
                               weak_foot_3                                weak_foot_4                                 weak_foot_5
                               402.506285                                283.626010                                 32.107356
                        Club_Int_Prestige_2                        Club_Int_Prestige_3                        Club_Int_Prestige_4
                                 1.539911                                  1.785874                                  1.725062
                        Club_Int_Prestige_5                        Club_Int_Prestige_6                        Club_Int_Prestige_7
                                 1.769082                                  2.025879                                  2.772897
                        Club_Int_Prestige_8                        Club_Int_Prestige_9                       Club_Int_Prestige_10
                                 1.635104                                  2.262729                                  2.375617
```

To get rid of multicollinearity, the variable with the highest VIF was dropped and the model was run again. The process was repeated , removing variables one-by-one till none of the variables had a VIF greater than 5. Variables removed based on VIF :

1. Dribbling
2. Shooting
3. Defending

These results are in line with our EDA whereby dribbling , shooting, and passing were correlated highly After having removed these variables , the model was run again. Below is the summary of the model.

Figure 22: Summary of MLR model I (after removing variables with VIF >5)



```
Coefficients:
                                                  Estimate  Std. Error t value             Pr(>|t|)    
(Intercept)                                   -7210243.206 1051686.947  -6.856     0.00000000000801 ***
Wages                                              147.170       2.602  56.557 < 0.0000000000000002 ***
age                                            -419319.040   18709.016 -22.413 < 0.0000000000000002 ***
pace                                             11323.930    6259.579   1.809             0.070507 .  
passing                                         142494.998   10170.810  14.010 < 0.0000000000000002 ***
physic                                           80615.719    8911.476   9.046 < 0.0000000000000002 ***
Mins_Total                                         594.511      98.098   6.060     0.00000000146463 ***
Goals_total                                     111176.610   27225.785   4.084     0.00004510414014 ***
Assists_Total                                    81717.838   35909.900   2.276             0.022913 *  
Shots_per_game_total                            767034.308   85718.501   8.948 < 0.0000000000000002 ***
Yellow_Cards_Total                               -2075.427   28307.062  -0.073             0.941556    
Red_Cards_Total                                -364009.138  118424.128  -3.074             0.002126 ** 
Aerial_Battles_Won_Per_Game_Total               -18390.647   50299.628  -0.366             0.714664    
Man_of_The_Match_Total                          294642.280   58761.625   5.014     0.00000055228412 ***
Player_international_reputation                1841152.413  137447.303  13.395 < 0.0000000000000002 ***
weak_foot                                       127801.271   94244.494   1.356             0.175145    
Club_Int_Prestige                               153419.595   42900.803   3.576             0.000352 ***
Club_Domestic_Prestige                          -19158.367   35173.066  -0.545             0.585994    
Player_Continent_Asia                          1168284.202  527814.392   2.213             0.026916 *  
Player_Continent_Europe                         -18174.650  213964.255  -0.085             0.932311    
Player_Continent_North.America                   98355.405  358630.695   0.274             0.783903    
Player_Continent_Oceania                       -847162.778  866708.810  -0.977             0.328398    
Player_Continent_South.America                  740456.353  265672.587   2.787             0.005340 ** 
Club_Continent_Europe.Tier.1                   -207970.354  409785.961  -0.508             0.611821    
Club_Continent_Europe.Tier.2                   -763331.404  419594.097  -1.819             0.068943 .  
Club_Continent_Europe.Tier.3                   -741089.610  417553.913  -1.775             0.075990 .  
Club_Continent_Europe.Tier.4                   -153900.965  513476.569  -0.300             0.764401    
Club_Continent_North.America                   -679315.897  458281.840  -1.482             0.138325    
Club_Continent_South.America                  -1068463.628  470258.455  -2.272             0.023127 *  
preferred_foot_Right                             90772.039  138762.474   0.654             0.513045    
Position_Final_Attacking.Midfield.Forward       380563.858  244734.667   1.555             0.120012    
Position_Final_Defense                         1713461.001  272287.980   6.293     0.00000000034030 ***
Position_Final_Defensive.Midfield               501759.674  302650.187   1.658             0.097408 .  
Position_Final_Defensive.Midfield.Defense      -395429.393  252273.404  -1.567             0.117074    
Position_Final_Forward                          347001.215  279662.736   1.241             0.214748    
Position_Final_Midfield..Attacking.Defensive.  -208981.150  233954.117  -0.893             0.371766    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3939000 on 4669 degrees of freedom
Multiple R-squared:  0.8139,	Adjusted R-squared:  0.8125 
F-statistic: 583.3 on 35 and 4669 DF,  p-value: < 0.00000000000000022
```

In the summary above , the following things are worth considering –

- Estimate, which shows the value of each slope coefficient (β)
- t-value, which shows the value of the test statistic for the following hypothesis for each predictor:

$$H_0: \beta = 0$$

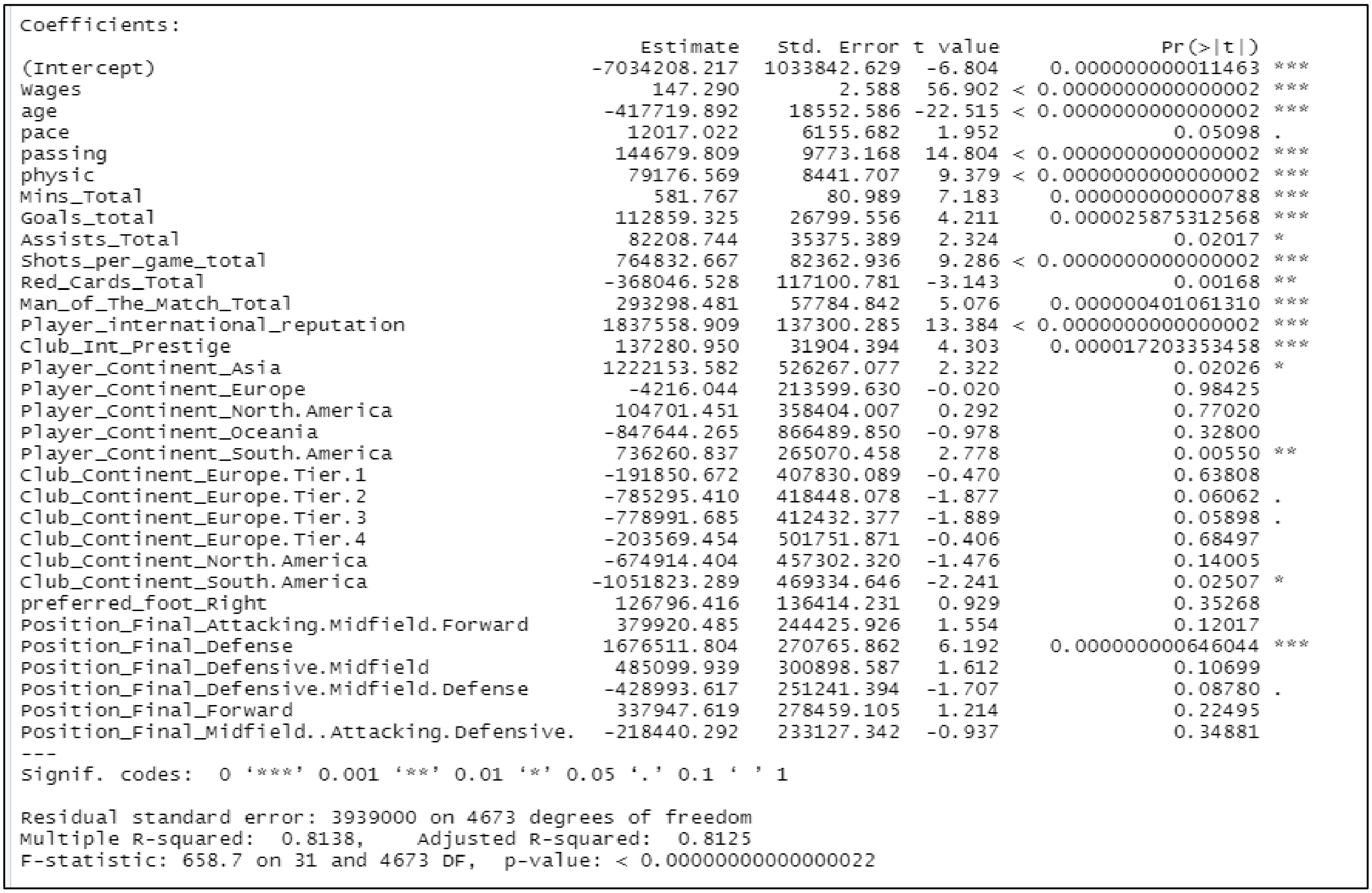
$$H_a: \beta \neq 0$$

- Corresponding to each t-value, a p-value is calculated (given in the column Pr(>|t|). If the p-value is less than the significance level (denoted at the bottom by significance codes) then the null hypothesis is rejected, and the variable and its coefficient is deemed to be statistically significant. For example, the variable age has an extremely small p-value (close to 0) and hence is statistically significant at .001 level of significance (or any significance level higher) than that. In simple words, the p-value shows the probability of the null hypothesis being true. The lower the p-value, the lesser the probability of the null hypothesis being true. Consequently, the lower the p-value, the more significant the variable.
- Multiple R-Squared shows the percentage of variation in the data set that is explained by the variables in the model. Higher the value of R-Squared, higher the variability explained by the model

- Adjusted R squared = 1- [{(1-R squared) *(n-1)} /{n-k-1}] where n is the number of observations and k is the number of IVs. In MLR , the concept of Adjusted R square is more important. The use of Adj R square in a model is to see whether adding a variable is useful or not. Since adding a variable will always increase the value of R square , the value of Adj R square will decrease if the added variable is insignificant
- From the summary above, we can see that the features in the model are able to explain approximately 81% of the variation int a player's market value.
- Further , a lot of variables have a high p-value even at a significance level of 0.1. This means that these variables are statistically very close to 0 and hence are not providing much information about the variability in the model.
- Therefore , these variables can be removed without having a great impact on the overall performance of the model (as measured by Adjusted R-Squared).
- Once again, dummy variables representing individual levels of a factor variable cannot be dropped for the same reason mentioned previously

  Below is the final summary of the MLR model using only the statistically significant variables

Figure 23: Summary of MLR I (only significant variables)



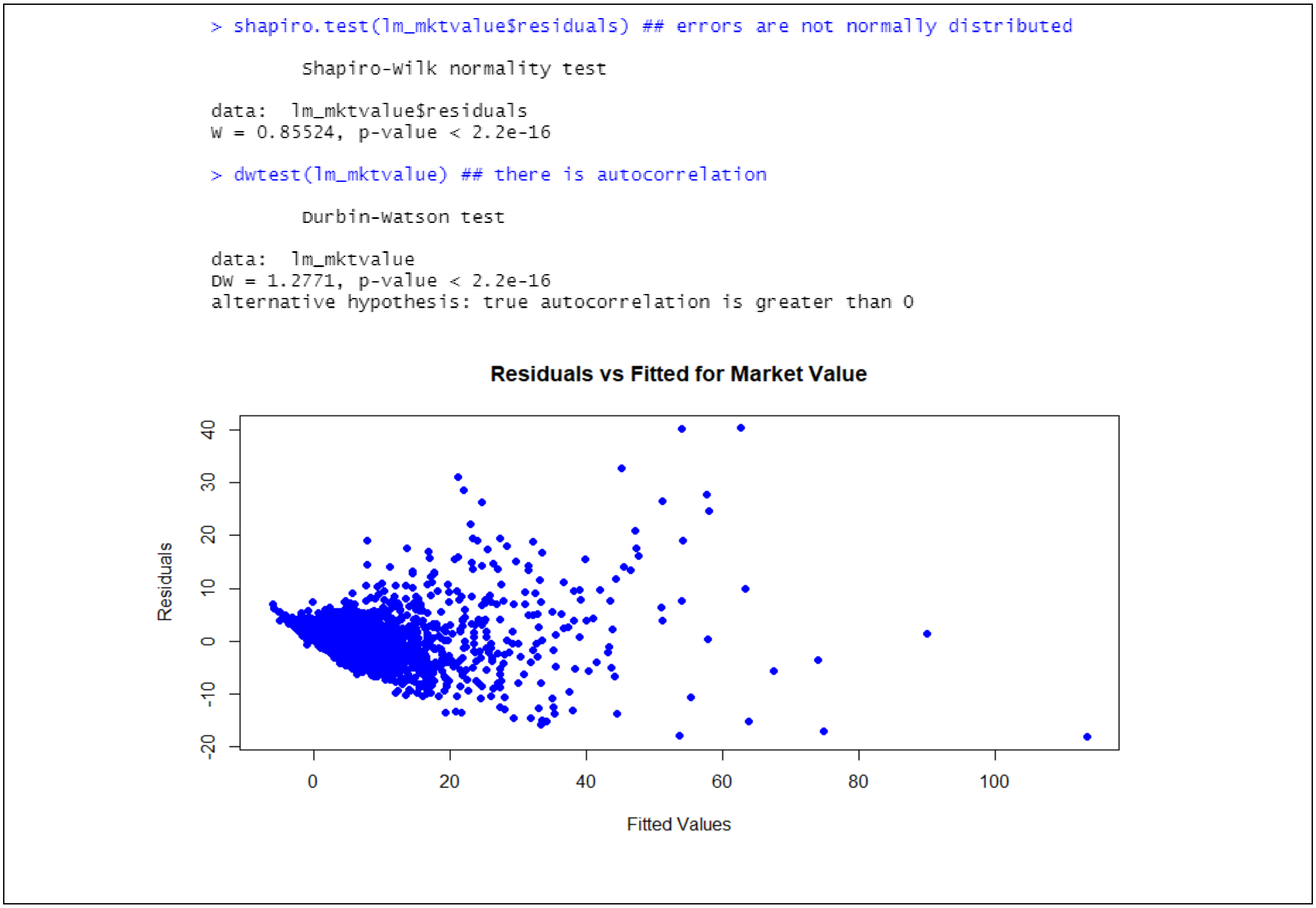
```
Coefficients:
                                          Estimate   Std. Error t value              Pr(>|t|)    
(Intercept)                           -7034208.217  1033842.629  -6.804    0.000000000011463 ***
Wages                                      147.290        2.588  56.902 < 0.0000000000000002 ***
age                                    -417719.892    18552.586 -22.515 < 0.0000000000000002 ***
pace                                     12017.022     6155.682   1.952              0.05098 .  
passing                                 144679.809     9773.168  14.804 < 0.0000000000000002 ***
physic                                   79176.569     8441.707   9.379 < 0.0000000000000002 ***
Mins_Total                                 581.767       80.989   7.183    0.000000000000788 ***
Goals_total                             112859.325    26799.556   4.211    0.000025875312568 ***
Assists_Total                            82208.744    35375.389   2.324              0.02017 *  
Shots_per_game_total                    764832.667    82362.936   9.286 < 0.0000000000000002 ***
Red_Cards_Total                        -368046.528   117100.781  -3.143              0.00168 ** 
Man_of_The_Match_Total                  293298.481    57784.842   5.076    0.000000401061310 ***
Player_international_reputation        1837558.909   137300.285  13.384 < 0.0000000000000002 ***
Club_Int_Prestige                       137280.950    31904.394   4.303    0.000017203353458 ***
Player_Continent_Asia                  1222153.582   526267.077   2.322              0.02026 *  
Player_Continent_Europe                  -4216.044   213599.630  -0.020              0.98425    
Player_Continent_North.America          104701.451   358404.007   0.292              0.77020    
Player_Continent_Oceania               -847644.265   866489.850  -0.978              0.32800    
Player_Continent_South.America          736260.837   265070.458   2.778              0.00550 ** 
Club_Continent_Europe.Tier.1           -191850.672   407830.089  -0.470              0.63808    
Club_Continent_Europe.Tier.2           -785295.410   418448.078  -1.877              0.06062 .  
Club_Continent_Europe.Tier.3           -778991.685   412432.377  -1.889              0.05898 .  
Club_Continent_Europe.Tier.4           -203569.454   501751.871  -0.406              0.68497    
Club_Continent_North.America           -674914.404   457302.320  -1.476              0.14005    
Club_Continent_South.America          -1051823.289   469334.646  -2.241              0.02507 *  
preferred_foot_Right                    126796.416   136414.231   0.929              0.35268    
Position_Final_Attacking.Midfield.Forward  379920.485   244425.926   1.554              0.12017    
Position_Final_Defense                 1676511.804   270765.862   6.192    0.000000000646044 ***
Position_Final_Defensive.Midfield       485099.939   300898.587   1.612              0.10699    
Position_Final_Defensive.Midfield.Defense -428993.617   251241.394  -1.707              0.08780 .  
Position_Final_Forward                  337947.619   278459.105   1.214              0.22495    
Position_Final_Midfield..Attacking.Defensive. -218440.292   233127.342  -0.937              0.34881    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3939000 on 4673 degrees of freedom
Multiple R-squared:  0.8138,	Adjusted R-squared:  0.8125 
F-statistic: 658.7 on 31 and 4673 DF,  p-value: < 0.00000000000000022
```

- As expected, by removing variables, the value of r-squared has fallen, albeit only marginally. However, since only insignificant variables were removed, the value of adjusted r-squared has gone up.

- Since the drop-in r-squared is only marginal, considering the number of variables dropped, this model can be considered much better.
- Next, we test the assumptions of the model created above. For the model's coefficients to be interpretable all the assumptions must be satisfied. By removing variables with high VIF, we have already taken care of multicollinearity in the model.

Figure 24: Checking the assumptions of MLR model I



```
> shapiro.test(lm_mktvalue$residuals) ## errors are not normally distributed

        Shapiro-Wilk normality test

data:  lm_mktvalue$residuals
W = 0.85524, p-value < 2.2e-16

> dwtest(lm_mktvalue) ## there is autocorrelation

        Durbin-Watson test

data:  lm_mktvalue
DW = 1.2771, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0
```

In the figure above, we can see the following:

1. The p-value for the Shapiro-Wilkinson test is extremely low at 0.05 level of significance. Thus, the null hypothesis that the error are normally distributed must be rejected. **This means that the errors in the model are not normally distributed**
2. The p-value for the Durbin-Watson test is extremely low at 0.05 level of significance. Thus, the null hypothesis that the error are not autocorrelated must be rejected. **This means that there is autocorrelation in the model**
3. The residuals vs fitted (predicted) value graph shows a conical shape which is an indicator of the presence of heteroscedastic. Thus, the error terms do not have equal variance from the mean.

Since 3 out of 5 assumptions are violated , therefore the slope coefficients of the model cannot be used to determine the effect of each individual feature on the response variable. However, the model can still be used for predicting the market value. However as stated previously , one should not rely heavily on these predictions as due to the violation of the assumptions they are not completely dependable.

Using the same logic , models were made for wages on data both with and without outlier treatment. After making the model, the assumptions were checked. Finally, the models made were used on the test data.

Below is a summary of the MLR models built so far:

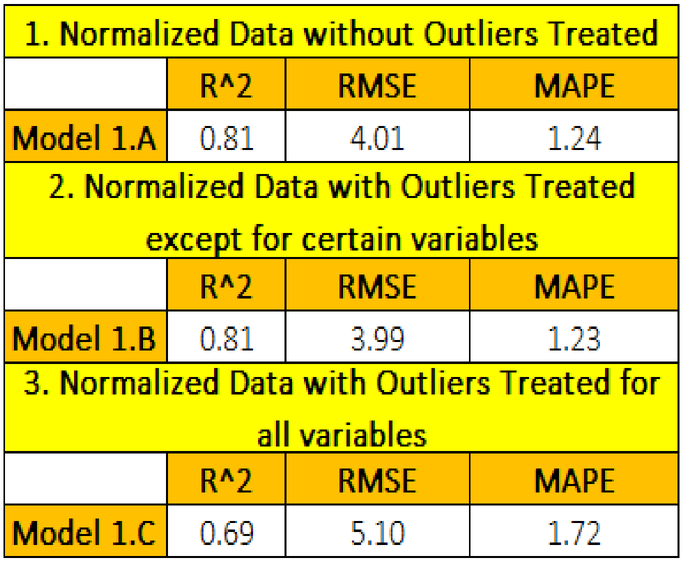
Table 6: Results for MLR models

1. Original Data without Outliers Treated							
Variables removed based on VIF and significance	R^2	RMSE	MAPE	Errors normally distributed ?	Errors Homoscedastic?	Multicollinearity exists ?	Is there Autocorrelation?
Model I	0.81	7.29	3.98	No	No	No	Yes
2. Original Data with Outliers Treated (excluding certain variables)							
Variables removed based on VIF and significance	R^2	RMSE	MAPE	Errors normally distributed?	Errors Homoscedastic	Multicollinearity exists ?	Is there Autocorrelation?
Model II	0.81	7.37	4.03	No	No	No	Yes
3. Original Data with Outliers Treated for all variables							
Variables removed based on VIF and significance	R^2	RMSE	MAPE	Errors normally distributed?	Errors Homoscedastic?	Multicollinearity exists ?	Is there Autocorrelation?
Model III	0.69	9.13	5.04	No	No	No	Yes

- As can be seen from the results in the tables above, none of the models satisfy the assumptions of MLR and therefore they should be used only for prediction purposes and not for interpretation of the slope coefficients. Hence, we now move to making non-normalized versions of the model using PyCaret. Below is a list of the MLR models made:
  - **1.A –** MLR for Market Value without outlier treatment and with non-ordinal numeric variables normalized
  - **1.B –** MLR for Market Value with outlier treatment for all variables except for Goals_Total Assists_Total, and Wages; and with non-ordinal numeric variables normalized
  - **1.C –** MLR for Market Value with outlier treatment for all variables and with non-ordinal numeric variables normalized

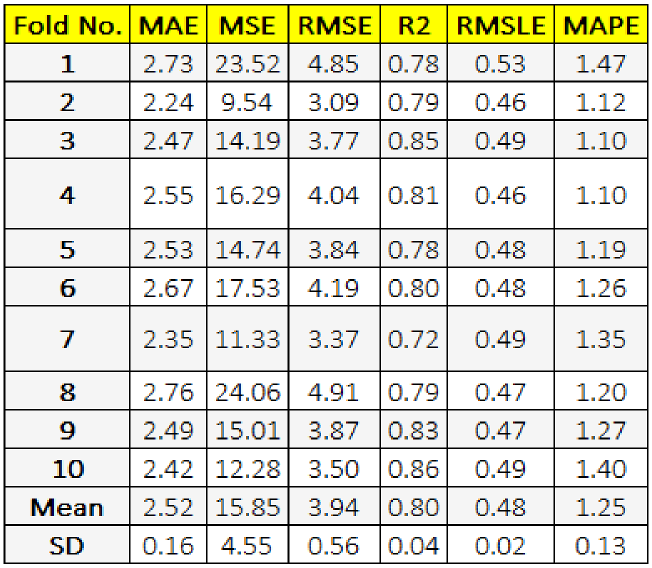
Below is a summary of the normalized MLR models along with the Cross validation (CV) results and Feature Importance plot for the best model:

Table 7: Results for Normalized MLR models



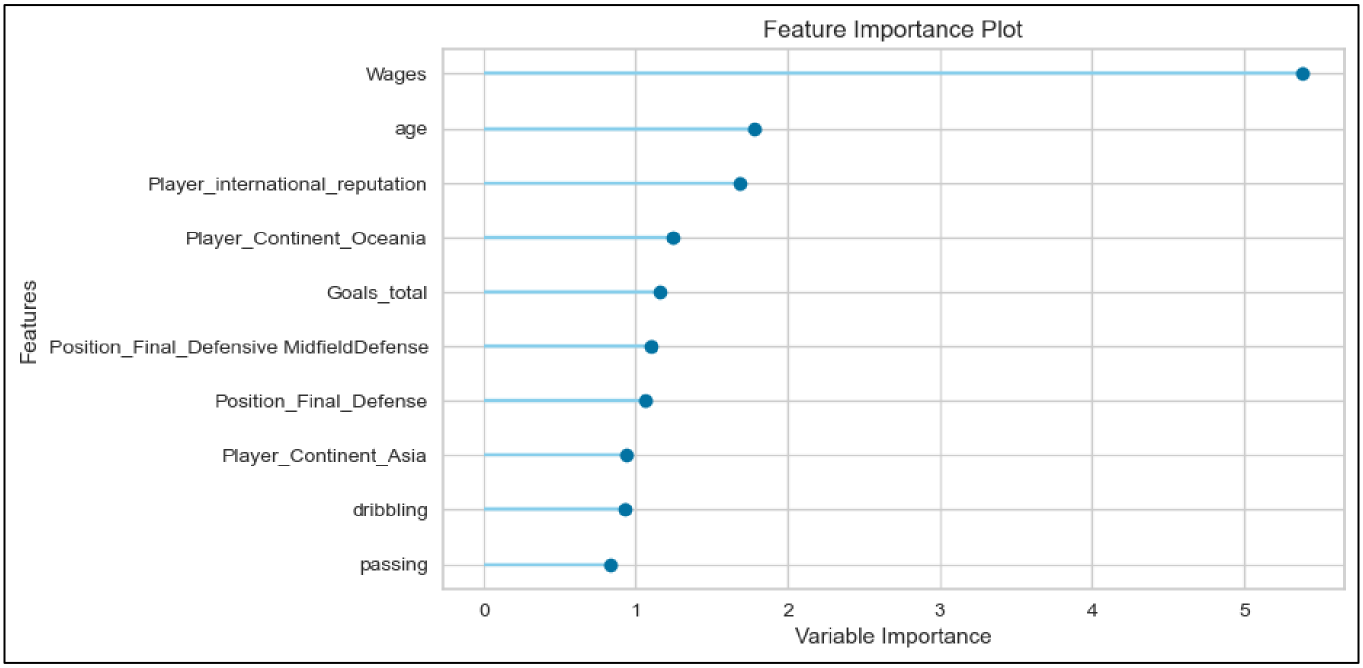
1. Normalized Data without Outliers Treated			
	R^2	RMSE	MAPE
Model 1.A	0.81	4.01	1.24
2. Normalized Data with Outliers Treated except for certain variables			
	R^2	RMSE	MAPE
Model 1.B	0.81	3.99	1.23
3. Normalized Data with Outliers Treated for all variables			
	R^2	RMSE	MAPE
Model 1.C	0.69	5.10	1.72

Table 8: CV Results for Best MLR model (1.B)



Fold No.	MAE	MSE	RMSE	R2	RMSLE	MAPE
1	2.73	23.52	4.85	0.78	0.53	1.47
2	2.24	9.54	3.09	0.79	0.46	1.12
3	2.47	14.19	3.77	0.85	0.49	1.10
4	2.55	16.29	4.04	0.81	0.46	1.10
5	2.53	14.74	3.84	0.78	0.48	1.19
6	2.67	17.53	4.19	0.80	0.48	1.26
7	2.35	11.33	3.37	0.72	0.49	1.35
8	2.76	24.06	4.91	0.79	0.47	1.20
9	2.49	15.01	3.87	0.83	0.47	1.27
10	2.42	12.28	3.50	0.86	0.49	1.40
Mean	2.52	15.85	3.94	0.80	0.48	1.25
SD	0.16	4.55	0.56	0.04	0.02	0.13

Figure 25: Feature Importance Plot for Best MLR model (1.B)



From the above tables we can see that the normalized MLR models have the same R Squared values as the original MLR models. Based on the RMSE results , Model 1.B is the best MLR model. Table 8 shows the cross validation results for this MLR model. We can see that the mean RMSE and MAPE values for all the folds are very close to each other. Further the RMSE and MAPE results from the train data (i.e. the mean RMSE and MAPE of all the folds) are very close to those of the test data suggesting the model is stable.

Finally, looking at the feature importance plot , we can see that Wages is the most important variable followed by age and then Player_international_reputation. Two of these variables were expected to be good predictors during our initial exploratory data analysis.

Model 2 – Lasso Regression

LASSO stands for "Least Absolute Shrinkage and Selection Operator". Lasso Regression is a regularization technique. These techniques are used to deal with overfitting, which happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. In other words, it means that the noise or random fluctuations in the training data are picked up and learned as concepts by the model. To avoid overfitting, the coefficients should be regulated by penalizing potential inflation of coefficients obtained in regression. Lasso Regression shrinks the regression coefficients toward zero by penalizing the regression model with a penalty term called **L1-norm** which is the sum of the absolute value of the magnitude of regression coefficients. LASSO is used when you have more variables and when you want to remove unwanted variables to the model, as it can bring the value of the coefficients to 0

The goal of the algorithm is to minimize:

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

which is the same as minimizing the sum of squares plus the penalty term at the end . Some of the βs are shrunk to exactly zero, resulting in a regression model that's easier to interpret. A tuning parameter, λ controls the strength of the L1 penalty. λ is basically the amount of shrinkage:

- When λ = 0, no parameters are eliminated. The estimate is equal to the one found with linear regression.
- As λ increases, more and more coefficients are set to zero and eliminated (theoretically, when λ = ∞, *all* coefficients are eliminated)[11]

Thus, Lasso can be also seen as an alternative to the subset selection methods for performing variable selection in order to reduce the complexity of the model. Below is a list of the Lasso Models made

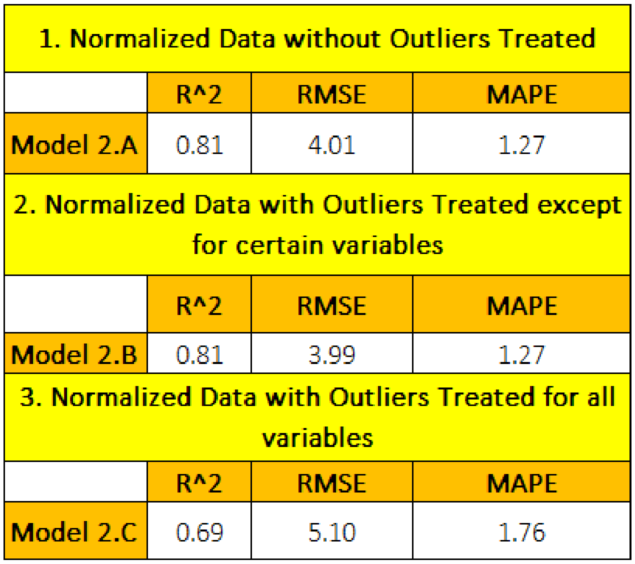
2.A – Lasso Regression for Market Value without outlier treatment and with non-ordinal numeric variables normalized

2.B – Lasso Regression for Market Value with outlier treatment for all variables except for Goals_Total, Assists_Total, and Wages; and with non-ordinal numeric variables normalized

2.C – Lasso Regression for Market Value with outlier treatment for all variables and with non-ordinal numeric variables normalized

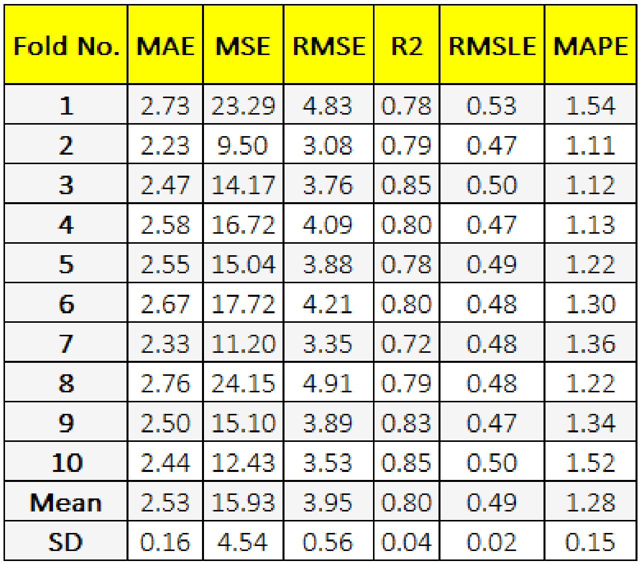
Below is a summary of the Lasso models along with the Cross validation (CV) results and Feature Importance plot for the best model:

Table 9: Results for Lasso models



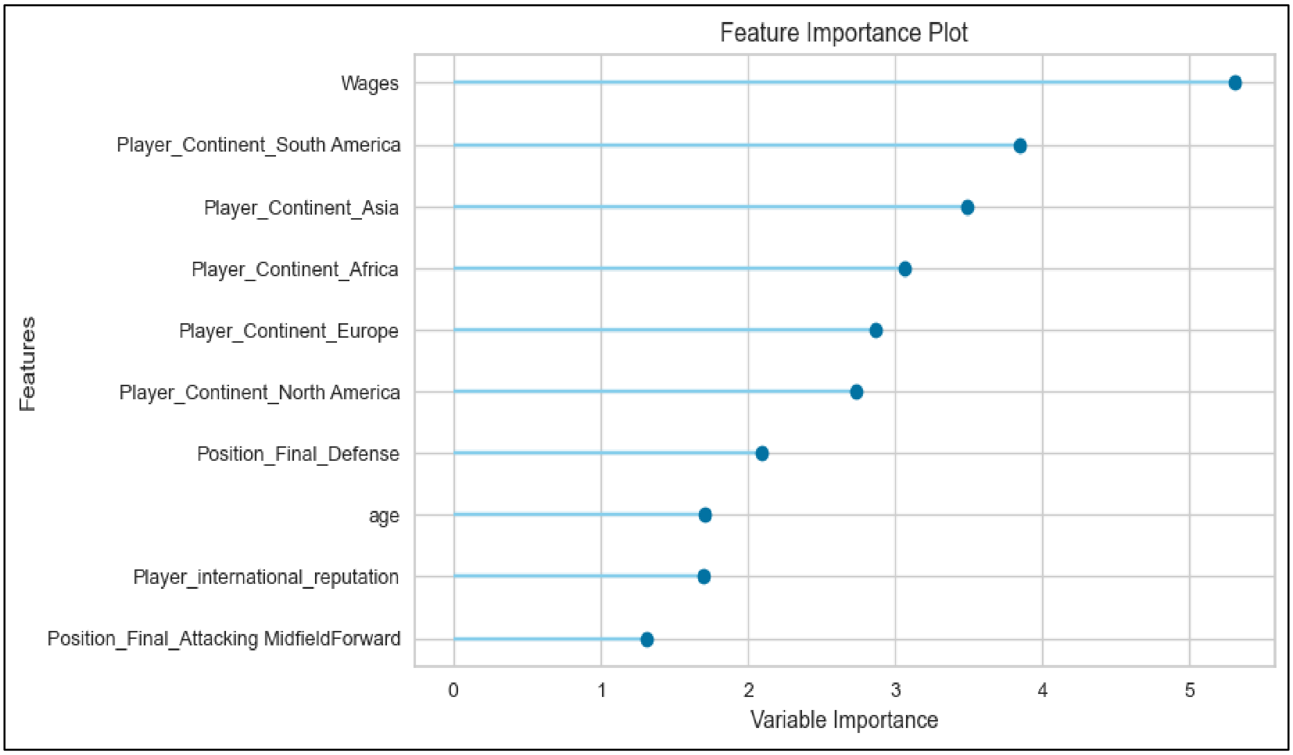
1. Normalized Data without Outliers Treated			
	R^2	RMSE	MAPE
Model 2.A	0.81	4.01	1.27
2. Normalized Data with Outliers Treated except for certain variables			
	R^2	RMSE	MAPE
Model 2.B	0.81	3.99	1.27
3. Normalized Data with Outliers Treated for all variables			
	R^2	RMSE	MAPE
Model 2.C	0.69	5.10	1.76

Table 10: CV Results for Best Lasso model (2.B)



Fold No.	MAE	MSE	RMSE	R2	RMSLE	MAPE
1	2.73	23.29	4.83	0.78	0.53	1.54
2	2.23	9.50	3.08	0.79	0.47	1.11
3	2.47	14.17	3.76	0.85	0.50	1.12
4	2.58	16.72	4.09	0.80	0.47	1.13
5	2.55	15.04	3.88	0.78	0.49	1.22
6	2.67	17.72	4.21	0.80	0.48	1.30
7	2.33	11.20	3.35	0.72	0.48	1.36
8	2.76	24.15	4.91	0.79	0.48	1.22
9	2.50	15.10	3.89	0.83	0.47	1.34
10	2.44	12.43	3.53	0.85	0.50	1.52
Mean	2.53	15.93	3.95	0.80	0.49	1.28
SD	0.16	4.54	0.56	0.04	0.02	0.15

Figure 26: Feature Importance Plot for Best Lasso model (2.B)



The results for the Lasso Models in table 10 are identical to those of the MLR models with model 2.B performing the best. The cross validation results for this model are also extremely close to those of model 1.B of MLR and the train data results are also very close the test data results, suggesting that the model is stable. However , the major difference is in the feature importance plot. While Wages is still the most important variable, Player_Continent is given more importance than age and then Player_international_reputation.

Model 3 – Ridge Regression

Ridge Regression is also a regularization technique. Ridge Regression shrinks the regression coefficients toward zero by penalizing the regression model with a penalty term called **L2-norm** which is the sum of the squared value of the magnitude of regression coefficients. Ridge helps to reduce or shrink the variance and making prediction less sensitive to the unwanted (insignificant) variable by minimizing its coefficient but does not remove the variable like Lasso does. All coefficients are shrunk by the same factor (so none are eliminated). The goal is to minimize

$$\sum_{i=1}^{M}(y_i-\hat{y}_i)^2=\sum_{i=1}^{M}\left(y_i-\sum_{j=0}^{p}w_j\times x_{ij}\right)^2+\lambda\sum_{j=0}^{p}w_j^2$$

which is the same as minimizing the sum of squares , where w represents the coefficients. When λ = 0, ridge regression equals least squares regression. If λ = ∞, all coefficients are shrunk to zero. The ideal penalty is therefore somewhere in between 0 and ∞ [11]. Below is a list of the Ridge Models made

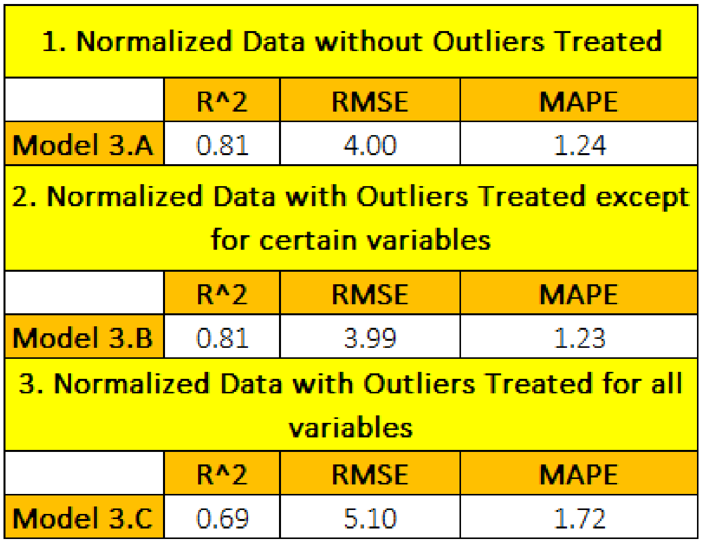
3.A – Ridge Regression for Market Value without outlier treatment and with non-ordinal numeric variables normalized

3.B – Ridge Regression for Market Value with outlier treatment for all variables except for Goals_Total, Assists_Total, and Wages; and with non-ordinal numeric variables normalized

3.C – Ridge Regression for Market Value with outlier treatment for all variables and with non-ordinal numeric variables normalized

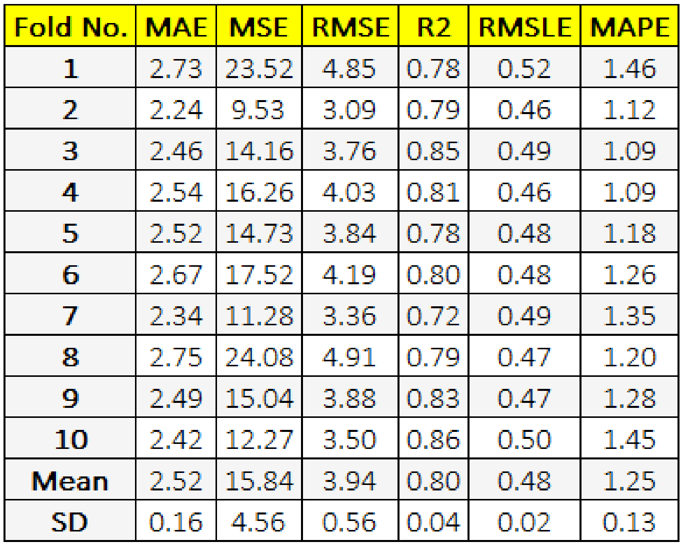
Below is a summary of the Ridge models along with the Cross validation (CV) results and Feature Importance plot for the best model:

Table 11: Results for Ridge models



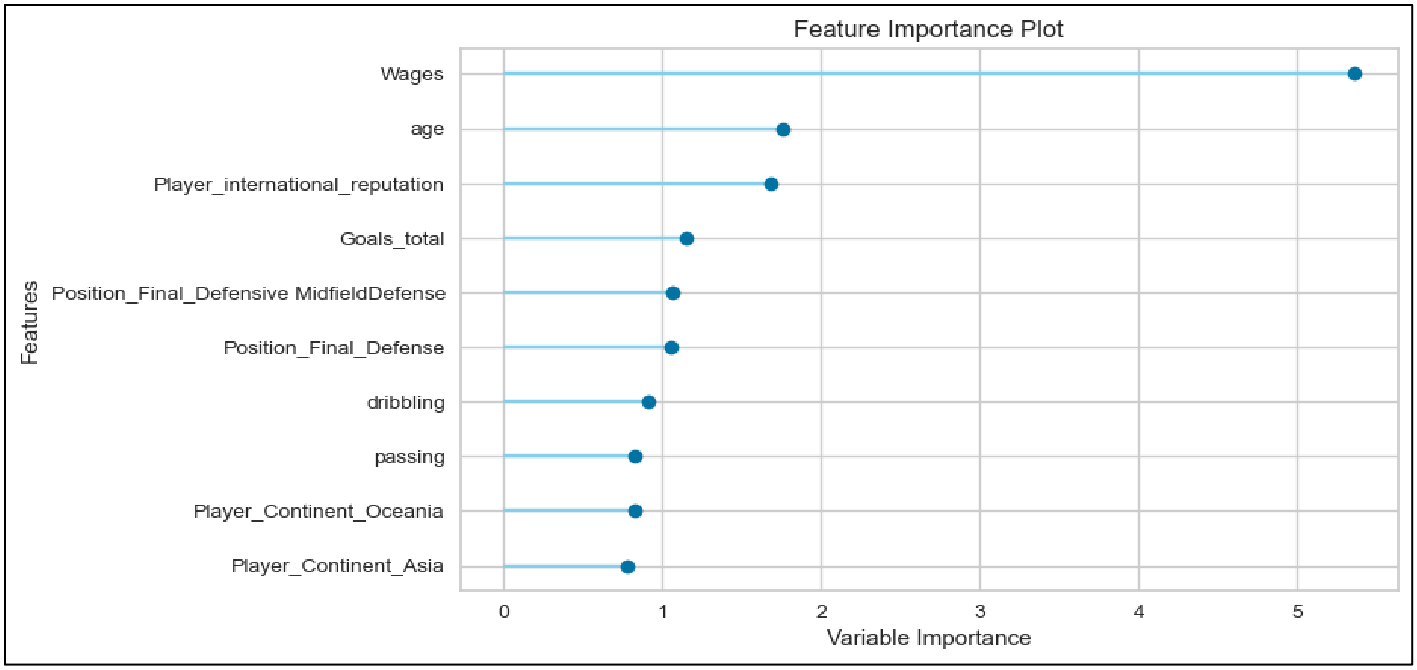
1. Normalized Data without Outliers Treated			
	R^2	RMSE	MAPE
Model 3.A	0.81	4.00	1.24
2. Normalized Data with Outliers Treated except for certain variables			
	R^2	RMSE	MAPE
Model 3.B	0.81	3.99	1.23
3. Normalized Data with Outliers Treated for all variables			
	R^2	RMSE	MAPE
Model 3.C	0.69	5.10	1.72

Table 12: CV Results for Best Ridge model (3.B)



Fold No.	MAE	MSE	RMSE	R2	RMSLE	MAPE
1	2.73	23.52	4.85	0.78	0.52	1.46
2	2.24	9.53	3.09	0.79	0.46	1.12
3	2.46	14.16	3.76	0.85	0.49	1.09
4	2.54	16.26	4.03	0.81	0.46	1.09
5	2.52	14.73	3.84	0.78	0.48	1.18
6	2.67	17.52	4.19	0.80	0.48	1.26
7	2.34	11.28	3.36	0.72	0.49	1.35
8	2.75	24.08	4.91	0.79	0.47	1.20
9	2.49	15.04	3.88	0.83	0.47	1.28
10	2.42	12.27	3.50	0.86	0.50	1.45
Mean	2.52	15.84	3.94	0.80	0.48	1.25
SD	0.16	4.56	0.56	0.04	0.02	0.13

Figure 27: Feature Importance Plot for Best Ridge model (3.B)



The results for the Ridge Models in table 12 are identical to those of the MLR and Lasso models with model 3.B performing the best. The cross validation results for this model are identical to those of the Lasso Model 2.B and the train data results are also very close the test data results, suggesting that the model is stable. The feature importance plot in this case is similar to that of the MLR model with Wages, age & Player_international_reputation being the three most important variables , followed by Goals_Total which also makes sense.

So far, we have drawn only Regression models and all of them have given almost the same results. These models are parametric models. Herein, parametricness is related to pair of model complexity and the number of rows in the train set. In a parametric model, we have a finite number of parameters, and in nonparametric models, the number of parameters is (potentially) infinite. Or in other words, in nonparametric models, the complexity of the model grows with the number of training data; in parametric models, we have a fixed number of parameters (or a fixed structure if you will). Linear models such as linear regression, logistic regression, are examples of a parametric learners; here, we have a fixed size of parameters (the slope coefficient.) In contrast, decision tree models, which we will make now, like CART, Random Forest etc. are considered as non-parametric learning algorithms since the number of parameters grows with the size of the training set. If we increase the number of instances, then the decision tree that is going to be built becomes more complex. The more decision rules could be created based on those new instances inherently.

In the field of statistics, the term parametric is also associated with a specified probability distribution that you “assume” your data follows, and this distribution comes with the finite number of parameters (for example, the mean and standard deviation of a normal distribution); you don’t make/have these assumptions in non-parametric models. So, in intuitive terms, we can think of a non-parametric model as a “distribution” or (quasi) assumptionfree model.[12] Thus, for tree-based models such as *CART, Random Forest*, *GBM and XGB*, there are no model assumptions to validate.

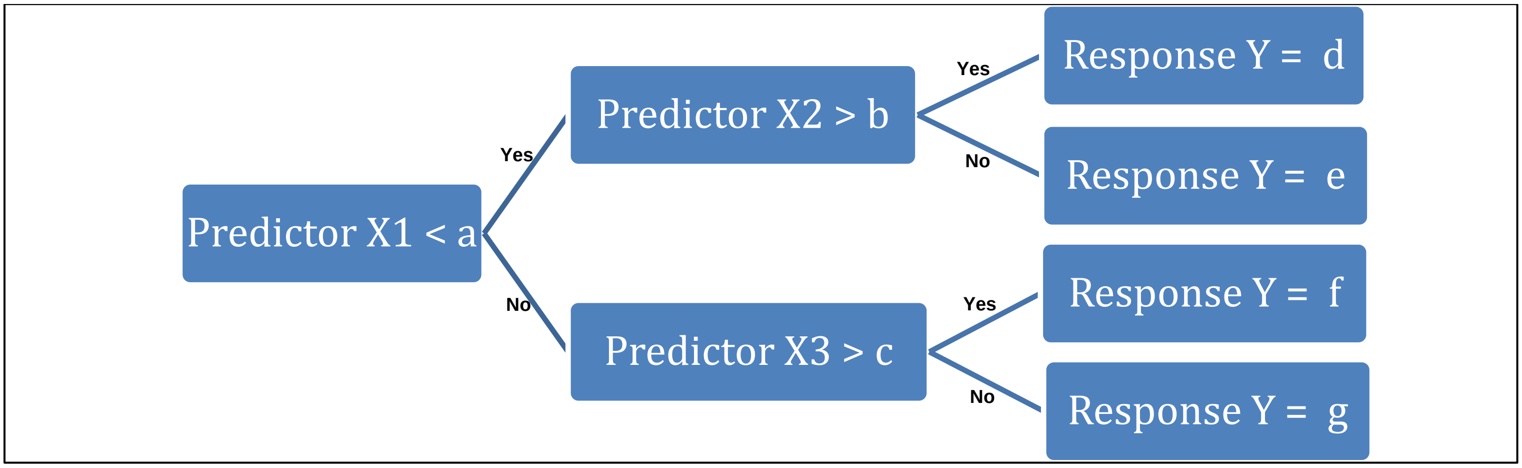
Further, unlike multiple linear regression or logistic regression, tree-based models **are robust to outliers.** This is because tree-based models pick those data points as nodes that best split the data to minimize cost (RSS). This recursive binary (if-then) always choses points as cut-offs in a way that most reduces RSS. Since outliers can't reduce RSS by much, tree-based models are generally more robust to them than say a distribution-based approach like logistic regression. For example: Suppose the goal is to determine the buying behaviour of customers depending upon their house size. House size is numeric continuous variable ranging from 1-1000 sq. ft. Suppose majority customers have house sizes in range of 100-500. If there are some customers with house size of 1000 then what it does is simply split the data on the basis of some value where the error at next level is less than that of the current level. Since this will never be the case with outliers, a split will never be made at an outlier.

Finally, tree-based models can take input variables in any form, factor or numeric. Thus, there is no need for one-hot encoding when applying CART, nor do we need to normalize the variable and bring them to the same scale.

Model 4 - Classification and Regression Tree (CART)

Classification and Regression Tree or CART is a machine learning algorithm that makes predictions for the response variable based on a decision tree. The decision tree consists of a number of root (or internal) nodes where each node corresponds to binary response in a predictor variable. Based on the decision at each root node of the tree, a final leaf node is generated which contains a value for the dependent variable.

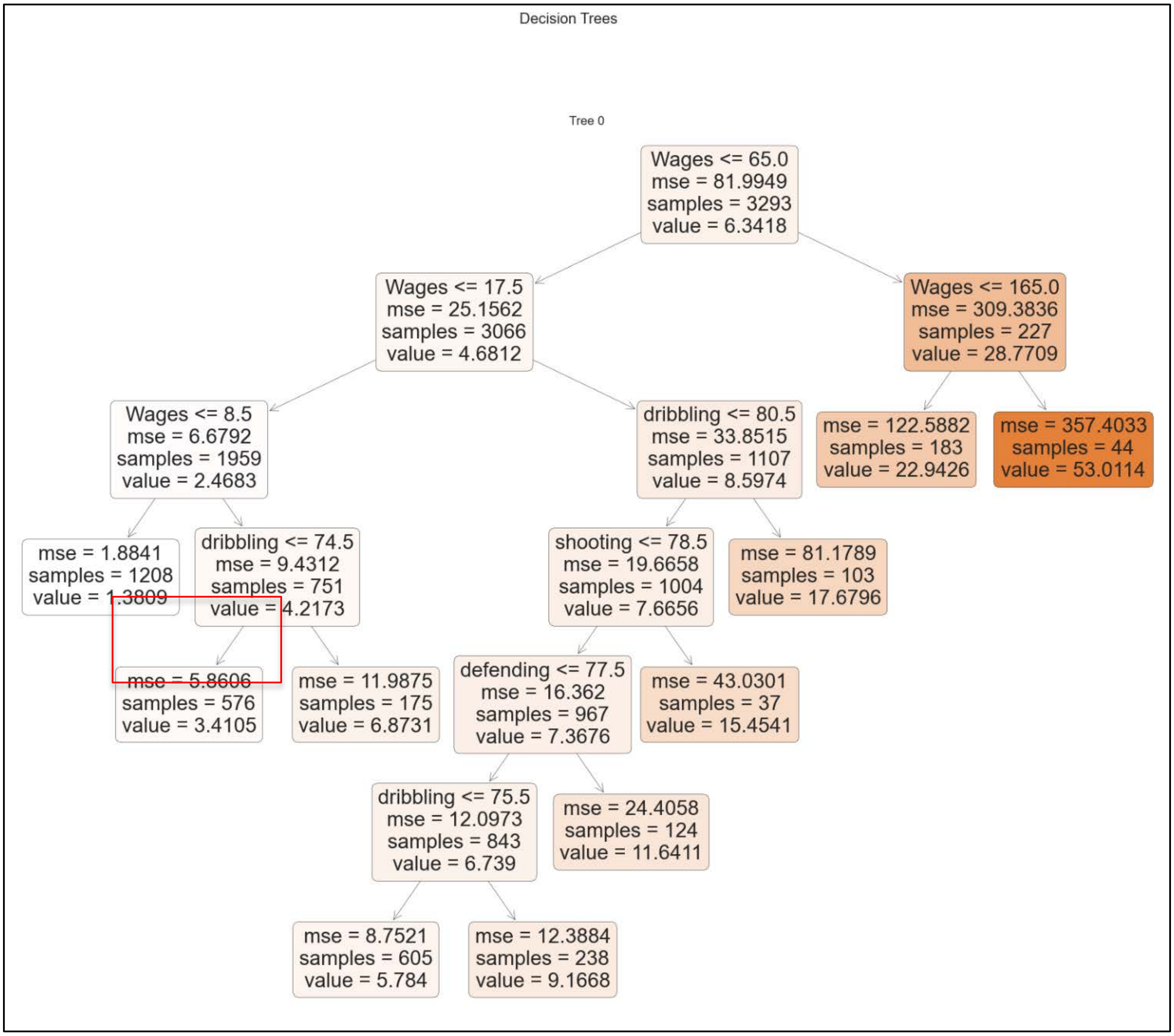
Figure 28: Example of a Decision Tree



The nodes containing binary conditions for each predictor variable X are called root nodes. Based on these conditions, different branches (or paths) are created giving different values for the response variable in the leaf nodes. Generally, the first variable at which a split is made is the most important feature for predicting the response variable. When the response variable is a factor , then the decision tree is called a Classification tree and when the response variable is numeric , then the tree is called a Regression Tree

The figure below is an example of a decision tree based on the FIFA data used in the analysis which aims to predict the market value of a football player

Figure 29: Sample Decision Tree for predicting Market Value



In the above figure since the first split is made at the variable Wages, therefore Wages is the most important variable when predicting market value for a player. If we take the left most terminal node, where the predicated value is 3.4105 (highlighted in red) , then the decision tress can be interpreted as follows – if Wages <=17.5 thousand and dribbling <=74.5 then the market value for the player will be 3.41 million euros. samples=576 shows the number of observations that fall into this category. Similarly, we can interpret all other terminal nodes.

The above decision tree is an example of a weak CART model. A stronger CART model will have more nodes and branches. Generally , when building a CART model , the following parameters need to be specified:

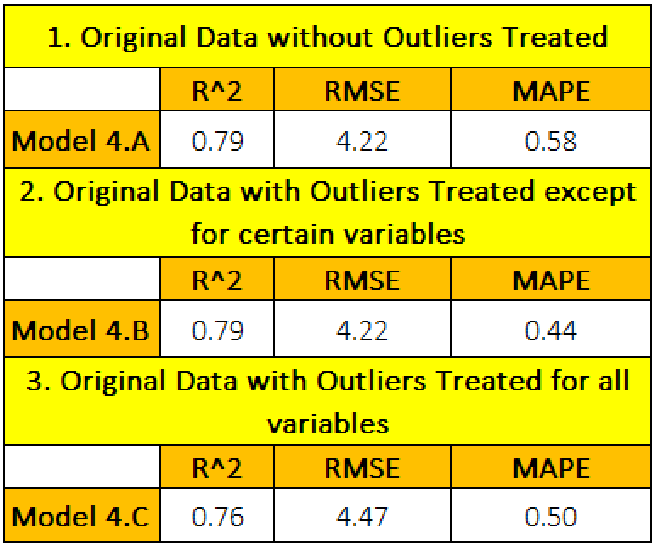
1. The minimum number of samples / observations which must be taken into consideration while creating a split at every node (given by **min_samples_split** in PyCaret)
2. The minimum number of samples / observations that must be in a leaf (or terminal) node. A split point at any depth will only be considered if it leaves at least min_samples_leaf training samples in each of the left and right branches. (given by **min_samples_leaf** in PyCaret)
3. The maximum depth of the tree. If it is not specified, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples. (given by **max_depth** in PyCaret)
4. The number of features to consider when looking for the best split. (given by **max_features** in PyCaret)
5. The complexity parameter which indicates the threshold level of improvement at every node or i.e. the minimum improvement in the model needed at each node. For example, if cp=0.01, then the algorithm will continue to improve the model, till the improvement at every node is more than 0.01. The moment it becomes less than 0.01, the regression tree stops developing further (given by **ccp_alpha** in PyCaret)

Below is a list of the CART Models made:

4.A – CART for Market Value without outlier treatment
4.B – CART for Market Value with outlier treatment for all variables except for Goals_Total, Assists_Total, and Wages
4.C – CART for Market Value with outlier treatment for all variables

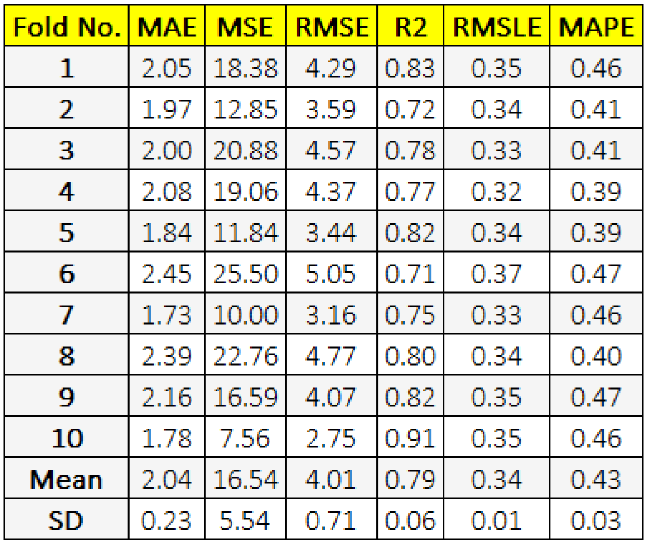
Below is a summary of the CART models along with the Cross validation (CV) results and Feature Importance plot for the best model:

Table 13: Results for CART models



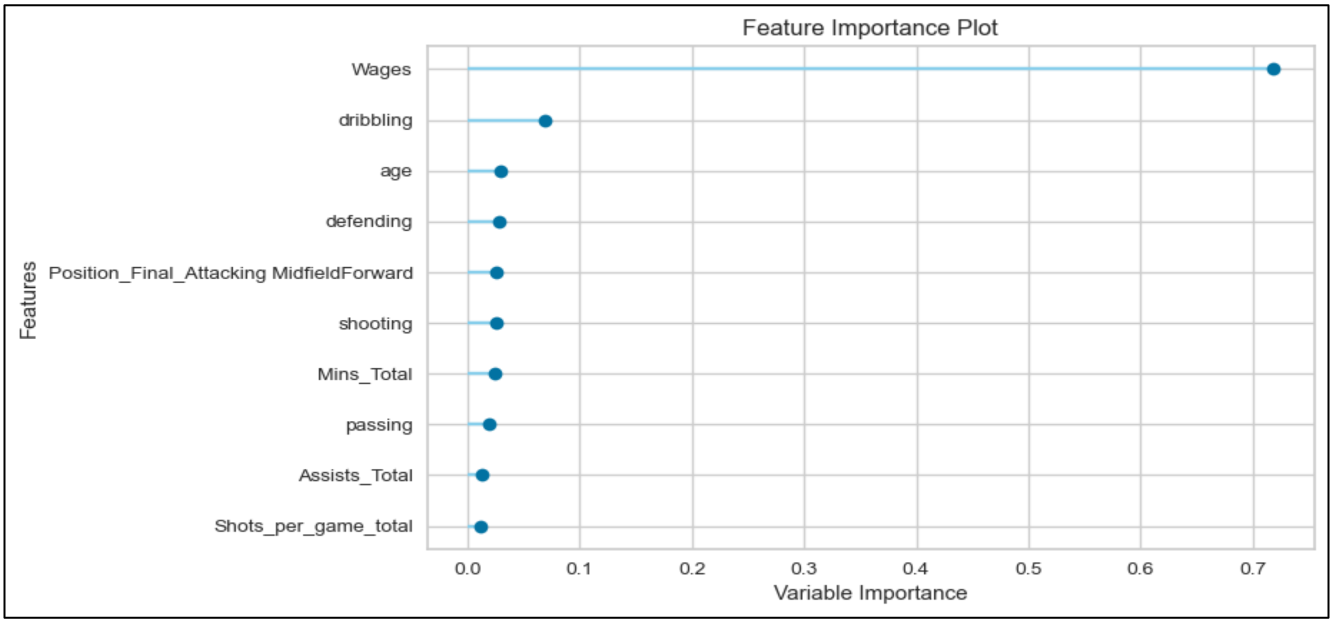
1. Original Data without Outliers Treated			
	R^2	RMSE	MAPE
Model 4.A	0.79	4.22	0.58
2. Original Data with Outliers Treated except for certain variables			
	R^2	RMSE	MAPE
Model 4.B	0.79	4.22	0.44
3. Original Data with Outliers Treated for all variables			
	R^2	RMSE	MAPE
Model 4.C	0.76	4.47	0.50

Table 14: CV Results for Best CART model (4.B)



Fold No.	MAE	MSE	RMSE	R2	RMSLE	MAPE
1	2.05	18.38	4.29	0.83	0.35	0.46
2	1.97	12.85	3.59	0.72	0.34	0.41
3	2.00	20.88	4.57	0.78	0.33	0.41
4	2.08	19.06	4.37	0.77	0.32	0.39
5	1.84	11.84	3.44	0.82	0.34	0.39
6	2.45	25.50	5.05	0.71	0.37	0.47
7	1.73	10.00	3.16	0.75	0.33	0.46
8	2.39	22.76	4.77	0.80	0.34	0.40
9	2.16	16.59	4.07	0.82	0.35	0.47
10	1.78	7.56	2.75	0.91	0.35	0.46
Mean	2.04	16.54	4.01	0.79	0.34	0.43
SD	0.23	5.54	0.71	0.06	0.01	0.03

Figure 30: Feature Importance Plot for Best CART model (4.B)



From the above tables we can see that R Squared values for the CART models are lower than those of the MLR models. Based on the MAPE results , Model 4.B is the best CART model, only marginally better than 4.A. Table 14 shows the cross validation results for this CART model. We can see that the mean RMSE and MAPE values for all the folds are very close to each other. Further the RMSE and MAPE results from the train data (i.e. the mean RMSE and MAPE of all the folds) are very close to those of the test data suggesting the model is stable. Finally, looking at the feature importance plot , we can see that Wages is the again most important variable.

The CART models give us low R-Squared values as compared to the Regression models. This is because the CART model makes prediction based on a single decision tree. An improvement on CART can be to use models that utilize multiple decision trees to make a prediction such as Random Forest , Gradient Boosting Machine and Extreme Gradient Boosting. Since these models involves using multiple trees to make a prediction they are called **Ensemble Methods**.

Ensemble learning is a machine learning paradigm where multiple models (often called “weak learners”) are trained to solve the same problem and combined to get better results. The main hypothesis is that when weak models are correctly combined we can obtain more accurate and/or robust models. This brings us to the question of how to combine these models. There are two major kinds of meta-algorithms that aim at combining weak learners: [13]

1. Bagging

- This method considers homogeneous weak learners, learns them independently from each other in parallel and combines them following some kind of deterministic averaging process.
- In parallel methods we fit the different considered learners independently from each other and, so, it is possible to train them concurrently.
- The most famous such approach is “bagging” (standing for “bootstrap aggregating”) that aims at producing an ensemble model that is more robust than the individual models composing it.

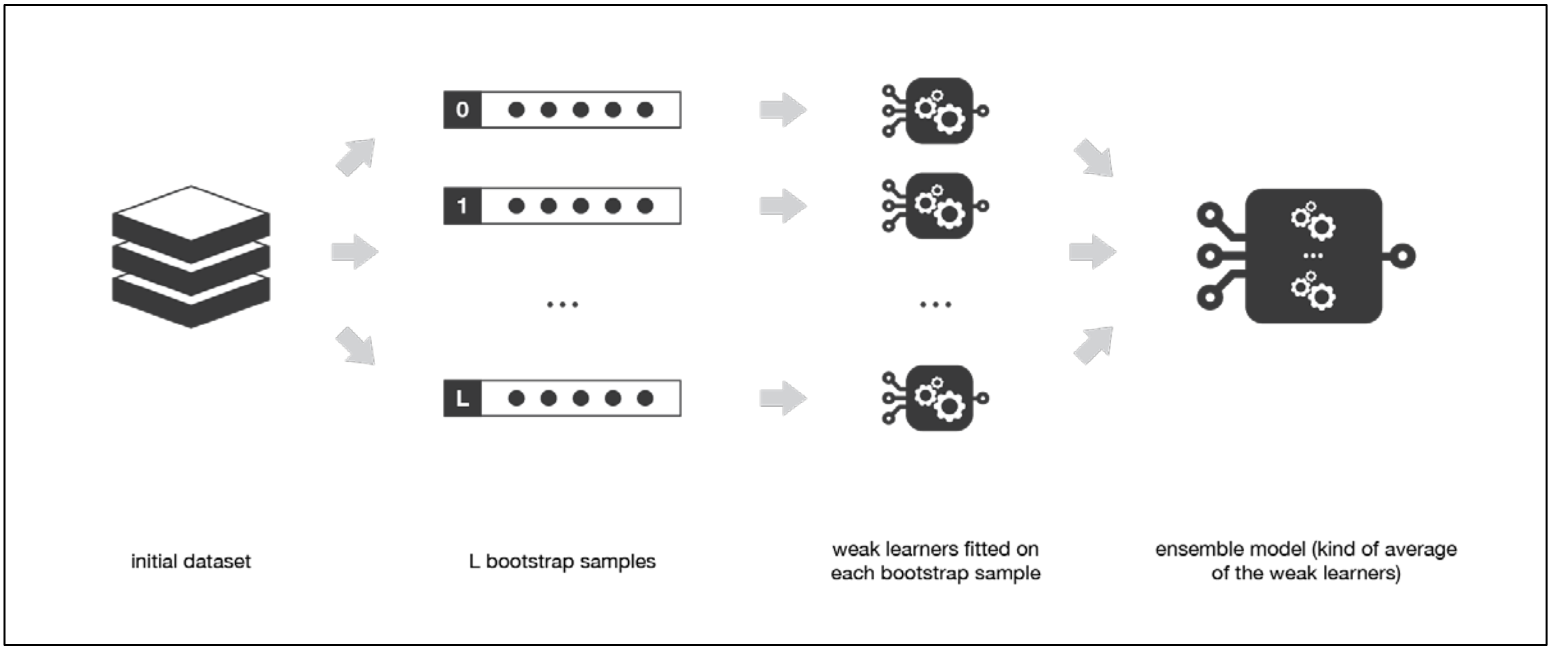
- Bootstrapping is a statistical technique which consists of generating samples ,with replacement, of size B (called bootstrap samples) from an initial dataset of size N.
- The idea of bagging is then simple: we want to fit several independent models and “average” their predictions in order to obtain a model with a lower variance. However, we can’t, in practice, fit fully independent models because it would require too much data. So, we rely on the good “approximate properties” of bootstrap samples to fit models that are almost independent.
- First, we create multiple bootstrap samples so that each new bootstrap sample will act as another (almost) independent dataset drawn from true distribution. Then, we can fit a weak learner for each of these samples and finally aggregate them such that we kind of “average” their outputs and, so, obtain an ensemble model with less variance that its components.

2. <u>Boosting</u>

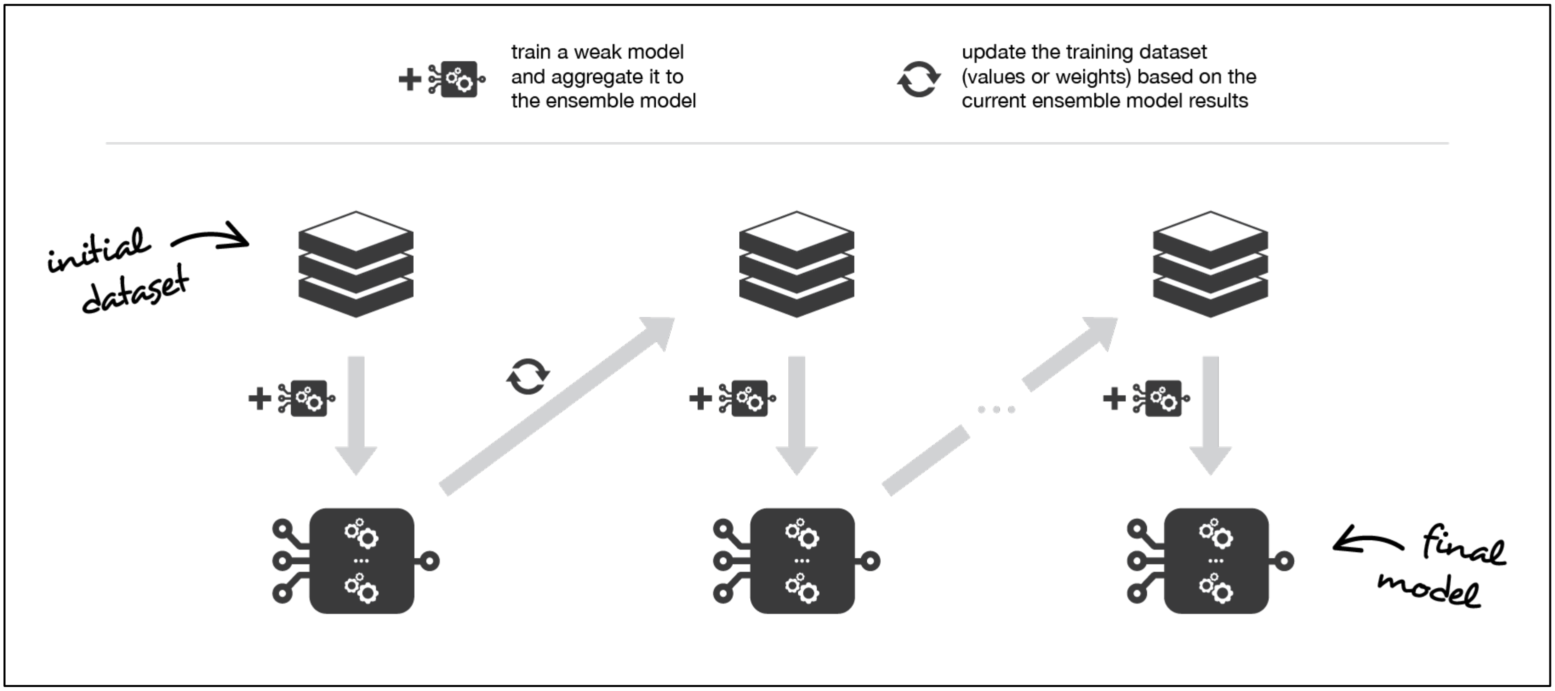
- This method considers homogeneous weak learners, learns them sequentially in a very adaptative way (a base model depends on the previous ones) and combines them following a deterministic strategy.
- In sequential methods the different combined weak models are no longer fitted independently from each other. The idea is to fit models iteratively such that the training of model at a given step depends on the models fitted at the previous steps.
- Boosting methods work in the same spirit as bagging methods: we build a family of models that are aggregated to obtain a strong learner that performs better.
- However, unlike bagging that mainly aims at reducing variance, boosting is a technique that consists in fitting sequentially multiple weak learners in a very adaptative way: each model in the sequence is fitted giving more importance to observations in the dataset that were badly handled by the previous models in the sequence.
- Intuitively, each subsequent model focuses its efforts on the most difficult observations to fit up to now, so that we obtain, at the end of the process, a strong learner.

Figure 31: Bagging vs Boosting[13]

<u>BAGGING</u>


0
1
...
L
...
initial dataset
L bootstrap samples
weak learners fitted on each bootstrap sample
ensemble model (kind of average of the weak learners)

BOOSTING


train a weak model and aggregate it to the ensemble model
update the training dataset (values or weights) based on the current ensemble model results
initial dataset
final model

Model 5 – Random Forest

Random Forest is a machine learning algorithm that works in the same way as CART. However instead of using just one tree , the prediction is based on multiple trees. The random forest approach is a bagging method where deep trees, fitted on bootstrap samples, are combined to produce an output which is an average of the output given by all trees. However, random forests also use another trick to make the multiple fitted trees when growing each tree : instead of only sampling over the observations in the dataset to generate a bootstrap sample, it also samples over features and keep only a random subset of them to build the tree. In other words , a Random Forest models uses a subset of the observations as well as features when making individual trees[13].

Sampling over features has indeed the effect that all trees do not look at the exact same information to make their decisions and, so, it reduces the correlation between the different returned outputs. Another advantage of sampling over the features is that it makes the decision making process more robust to missing data: observations (from the training dataset or not) with missing data can still be regressed or classified based on the trees that take into account only features where data are not missing. Thus, random forest algorithm combines the concepts of bagging and random feature subspace selection to create more robust models[13].

Since Random Forest is also a decision tree based model , it takes in all the parameters as CART. However, there are two additional important parameters to consider when making a random forest mode:

1. The number of trees to be built in the random forest model (given by n_estimators in PyCaret).
2. The number of variables out of the total variables, that should be used while building each tree (given by max_features in PyCaret

Below is a list of the RF Models made

5.A – RF for Market Value without outlier treatment

5.B – RF for Market Value with outlier treatment for all variables except for Goals_Total, Assists_Total, and Wages

5.C – RF for Market Value with outlier treatment for all variables

Below is a summary of the RF models along with the Cross validation (CV) results and Feature Importance plot for the best model:

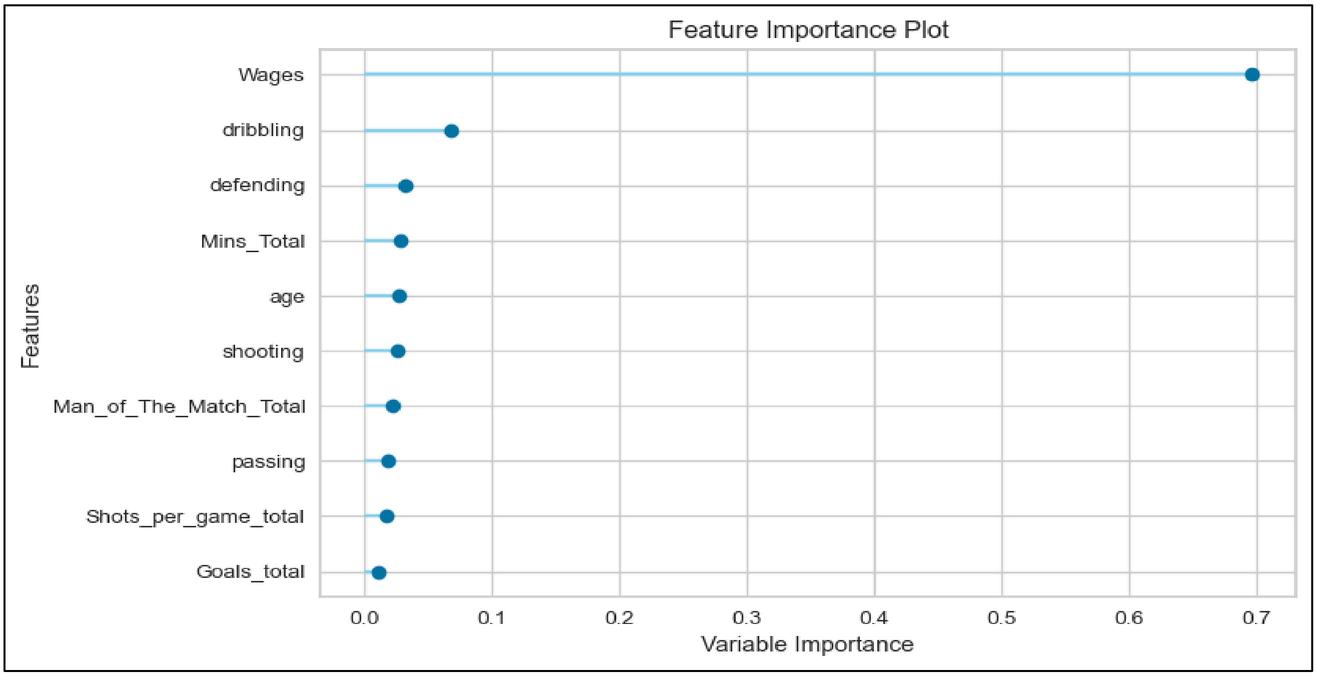
Table 15: Results for RF models

1. Original Data without Outliers Treated			
	R^2	RMSE	MAPE
Model 5.A	0.89	3.10	0.34
2. Original Data with Outliers Treated except for certain variables			
	R^2	RMSE	MAPE
Model 5.B	0.89	3.11	0.34
3. Original Data with Outliers Treated for all variables			
	R^2	RMSE	MAPE
Model 5.C	0.88	3.17	0.34

Table 16: CV Results for Best RF model (5.A)

Fold No.	MAE	MSE	RMSE	R2	RMSLE	MAPE
1	1.62	14.44	3.80	0.86	0.27	0.36
2	1.38	6.21	2.49	0.86	0.25	0.32
3	1.50	11.92	3.45	0.88	0.25	0.32
4	1.58	8.53	2.92	0.90	0.25	0.35
5	1.35	7.67	2.77	0.89	0.22	0.27
6	1.74	11.94	3.46	0.86	0.27	0.38
7	1.28	6.68	2.58	0.83	0.25	0.35
8	1.84	14.82	3.85	0.87	0.24	0.31
9	1.53	8.49	2.91	0.91	0.26	0.38
10	1.42	6.60	2.57	0.92	0.26	0.35
Mean	1.52	9.73	3.08	0.88	0.25	0.34
SD	0.17	3.11	0.49	0.02	0.01	0.03

Figure 32: Feature Importance Plot for Best RF model (5.A)



From the above tables we can see that in comparison with the CART models , the R Squared values for the RF models , have increased significantly and the RMSE and MAPE values have fallen , resulting in better predictions. Model 5.A is the best RF model, only marginally better than both model 5.B and 5.C. Table 16 shows the cross validation results for this CART model. We can see that the mean RMSE and MAPE values for all the folds are very close to each other. Further the RMSE and MAPE results from the train data (i.e. the mean RMSE and MAPE of all the folds) are very close to those of the test data suggesting the model is stable. Finally, once again Wages are given the highest importance when making the prediction.

Model 6 – Gradient Boosting Machine

Gradient Boosting Machine (GBM) is a boosting machine learning algorithm. It re-defines boosting as a numerical optimization problem where the objective is to minimize the loss function of the model (which gives the error) by adding weak learners using gradient descent. Gradient descent is a first-order iterative optimization algorithm for finding a local minimum of a differentiable function. Thus ,it is a sequential ensemble learning technique where the performance of the model improves over iterations. This method creates the model in a stage-wise fashion. It infers the model by enabling the optimization of an absolute differentiable loss function. As we add each weak learner, a new model is created that gives a more precise estimation of the response variable.

Intuitively, gradient boosting is a stage-wise additive model that generates learners during the learning process (i.e., trees are added one at a time, and existing trees in the model are not changed). The contribution of the weak learner to the ensemble is based on the gradient descent optimization process.

In gradient boosting, we try to reduce the loss by adding decision trees. Also, we can minimize the error rate by cutting down the parameters. So, in this case, we design the model in such a way that the addition of a tree does not change the existing tree.

Gradient boosting does not modify the sample distribution as weak learners train on the remaining residual errors of a strong learner (i.e., pseudo-residuals). By training on the residuals of the model, this is an alternative means to give more importance to misclassified observations. Intuitively, new weak learners are being added to concentrate on the areas where the existing learners are performing poorly. The contribution of each weak learner to the final prediction is based on a gradient optimization process to minimize the overall error of the strong learner. [14]

Just like Random Forest , Gradient Boosting also takes all parameters of CART since it also makes decision trees. The only difference in the parameters to be specified is that instead of specifying the number of trees to be made as in Random Forest , we need to specify the number of boosting stages to perform (given by **n_estimators** in PyCaret). Along with that GBM also takes an additional parameter of learning rate(given by **learning_rate** in PyCaret).The learning rate corresponds to how quickly the error is corrected from each tree to the next and is a simple multiplier 0<LR≤1.For example, if the current prediction for a particular example is 0.2 and the next tree predicts that it should actually be 0.8, the correction would be +0.6. At a learning rate of 1, the updated prediction would be the full 0.2+1(0.6)=0.8, while a learning rate of 0.1 would update the prediction to be 0.2+0.1(0.6)=0.26. [15]

Below is a list of the GBM Models made

6.A – GBM for Market Value without outlier treatment

6.B – GBM for Market Value with outlier treatment for all variables except for Goals_Total, Assists_Total, and Wages

6.C – GBM for Market Value with outlier treatment for all variables

Below is a summary of the GBM models along with the Cross validation (CV) results and Feature Importance plot for the best model:

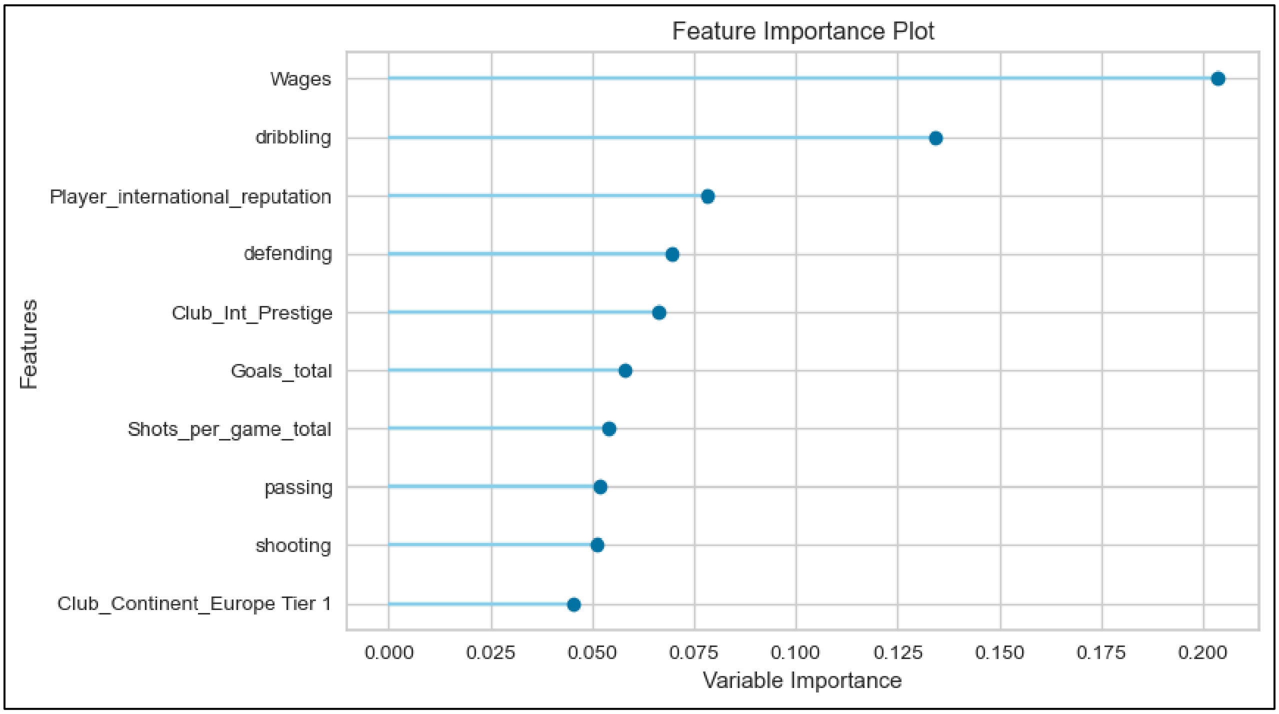
Table 17: Results for GBM models

1. Original Data without Outliers Treated			
	R^2	RMSE	MAPE
Model 6.A	0.93	2.44	0.37
2. Original Data with Outliers Treated except for certain variables			
	R^2	RMSE	MAPE
Model 6.B	0.92	2.58	0.35
3. Original Data with Outliers Treated for all variables			
	R^2	RMSE	MAPE
Model 6.C	0.92	2.66	0.34

Table 18: CV Resus for Best GBM model (6.A)

Fold No.	MAE	MSE	RMSE	R2	RMSLE	MAPE
1	1.42	9.52	3.09	0.91	0.26	0.38
2	1.31	5.05	2.25	0.89	0.26	0.37
3	1.41	8.41	2.90	0.91	0.26	0.36
4	1.44	6.67	2.58	0.92	0.25	0.37
5	1.21	4.83	2.20	0.93	0.22	0.30
6	1.42	7.13	2.67	0.92	0.24	0.34
7	1.27	5.40	2.32	0.86	0.25	0.38
8	1.66	9.12	3.62	0.89	0.25	0.34
9	1.38	7.54	2.75	0.92	0.24	0.36
10	1.27	4.81	2.19	0.94	0.27	0.43
Mean	1.38	7.25	2.66	0.91	0.25	0.36
SD	0.12	2.48	0.44	0.02	0.01	0.03

Figure 33: Feature Importance Plot for Best GBM model (6.A)

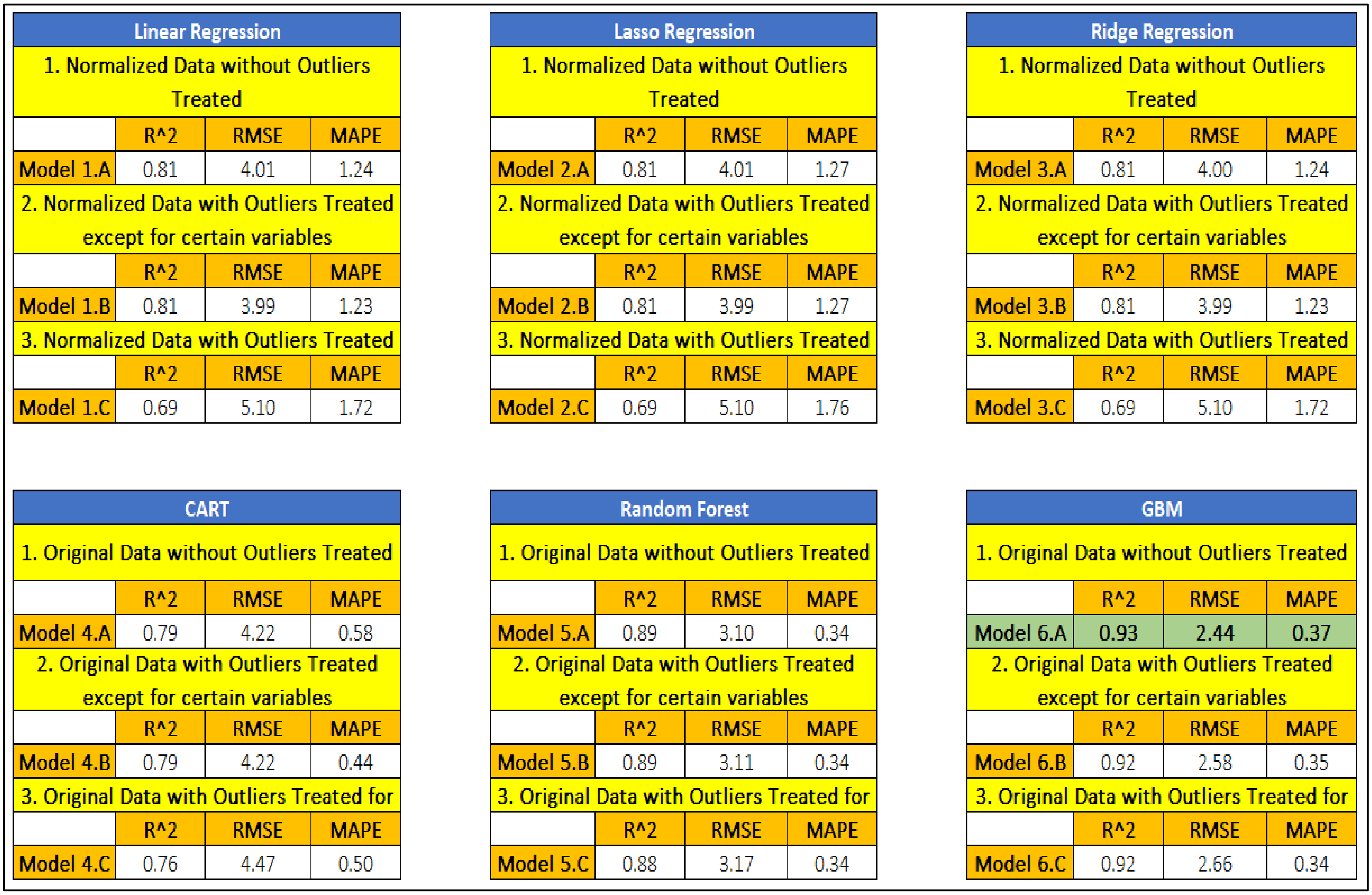


Once again , the R Squared values for the GBM models , is significantly higher than the CART models as multiple decision trees are made instead of just the one .The RMSE and MAPE values have also fallen , resulting in better predictions. Model 6.A is the best GBM model based on the RMSE , only marginally better than both model 6.B and 6.C. Table 18 shows the cross validation results for this CART model. We can see that the mean RMSE and MAPE values for all the folds are very close to each other. Further the RMSE and MAPE results from the train data (i.e. the mean RMSE and MAPE of all the folds) are very close to those of the test data suggesting the model is stable. Finally, once again Wages are given the highest importance when making the prediction , followed by dribbling and then Player_international_reputation.

Model Comparison

Having made all the machine learning models, we can now compare their performance based on the value of R Square, RMSE and MAPE measure. The figure below plots these values for each model for the purpose of comparison.

Figure 34: Model Comparison



Linear Regression			
1. Normalized Data without Outliers Treated			
	R^2	RMSE	MAPE
Model 1.A	0.81	4.01	1.24
2. Normalized Data with Outliers Treated except for certain variables			
	R^2	RMSE	MAPE
Model 1.B	0.81	3.99	1.23
3. Normalized Data with Outliers Treated			
	R^2	RMSE	MAPE
Model 1.C	0.69	5.10	1.72

Lasso Regression			
1. Normalized Data without Outliers Treated			
	R^2	RMSE	MAPE
Model 2.A	0.81	4.01	1.27
2. Normalized Data with Outliers Treated except for certain variables			
	R^2	RMSE	MAPE
Model 2.B	0.81	3.99	1.27
3. Normalized Data with Outliers Treated			
	R^2	RMSE	MAPE
Model 2.C	0.69	5.10	1.76

Ridge Regression			
1. Normalized Data without Outliers Treated			
	R^2	RMSE	MAPE
Model 3.A	0.81	4.00	1.24
2. Normalized Data with Outliers Treated except for certain variables			
	R^2	RMSE	MAPE
Model 3.B	0.81	3.99	1.23
3. Normalized Data with Outliers Treated			
	R^2	RMSE	MAPE
Model 3.C	0.69	5.10	1.72

CART			
1. Original Data without Outliers Treated			
	R^2	RMSE	MAPE
Model 4.A	0.79	4.22	0.58
2. Original Data with Outliers Treated except for certain variables			
	R^2	RMSE	MAPE
Model 4.B	0.79	4.22	0.44
3. Original Data with Outliers Treated for			
	R^2	RMSE	MAPE
Model 4.C	0.76	4.47	0.50

Random Forest			
1. Original Data without Outliers Treated			
	R^2	RMSE	MAPE
Model 5.A	0.89	3.10	0.34
2. Original Data with Outliers Treated except for certain variables			
	R^2	RMSE	MAPE
Model 5.B	0.89	3.11	0.34
3. Original Data with Outliers Treated for			
	R^2	RMSE	MAPE
Model 5.C	0.88	3.17	0.34

GBM			
1. Original Data without Outliers Treated			
	R^2	RMSE	MAPE
Model 6.A	0.93	2.44	0.37
2. Original Data with Outliers Treated except for certain variables			
	R^2	RMSE	MAPE
Model 6.B	0.92	2.58	0.35
3. Original Data with Outliers Treated for			
	R^2	RMSE	MAPE
Model 6.C	0.92	2.66	0.34

In the tables above we can notice the following points :

- The Linear, Lasso and Ridge Regression models all give very similar results. For each of these algorithms, the model made with outlier treatment for all variables is much weaker as compared to the models where no or limited outlier treatment was done.
- Amongst the models based on decision trees, CART performs worse than the regression models. However, the ensemble models, namely RF and GBM, outperform the regression models in each instance.
- In all of the 6 algorithms used, the model with outlier treatment for all variables gives the least accurate predictions
- One major difference between the Regression and Decision Tree models is that, in case of the latter, all three methods of outlier treatment give very close results, which is not the case with the regression models. This was expected as we discussed earlier than tree-based models are robust to outliers. Thus, if outliers are important to be incorporated in the model, as is the case in our problem, then a tree-based model is more reliable.
- The best model of all the models made is model 6.A which is the GBM model without any outlier treatment. It has the highest R squared value which means, using the given features, it is able to explain the variation in the market value of players to the maximum extent. This model also gives the least values of RMSE and MAPE and hence the best accuracy when making the predictions. The results for the training and testing datasets for this model are also very close to each other, suggesting that the model is not suffering from overfitting or underfitting
- The feature importance plot of this GBM model 6.A gives a lot of importance to the player's Wages and International Reputation, which is in line with our expectations based on the EDA performed in the beginning.

7. Conclusion and Recommendations

Having made various machine learning models , the comparison chart above (Fig 34) shows the performance of all model. We can see that the gradient boosting machine model without any outlier treatment (Model 6.A) gave us the best results. While the tables in this Fig 34 represent results on the test dataset, Table 18 shows the results for model 6.A on the train data. The table shows that the train data was divided into 10 folds for cross validation , which each fold giving very close values for R^2 , RMSE and MAPE. The final train data results are an average of the results given by all the folds. Thus, the training dataset gives us an R^2 = 0.91 , RMSE = 2.66 and MAPE = 0.36. These values are very close to those obtained from the test data with R^2 = 0.93 , RMSE = 2.44 and MAPE = 0.37. Thus, the results on the unseen dataset are good and inline with the results on the training dataset which suggest that the model is stable and is ready for use. It managed to explain 93% of the variation in the football players' market value based on the predictors. Further , the predictions from the model did not swerve considerably from the actual market value. The mean deviation was around €2.5 million which is marginal compared to the huge market value of players. And while making the prediction , it is the existing Wages (or salary) or a football player that aided the most in predicting his market value.

As such ,the model that has been built can be used by a football club to determine how much money to bid when considering buying a player from another club. Added to that , the model also indicates what information about the player is most likely to be important when determining the value of such a bid. Since the feature importance plot shows Wages to be the most important variables in predicting the market value, this is the first piece of information that the club should aim to gather when considering buying a new player. Although the model has incorporated 93% of the market prediction, commercial giants have the resources to capture many more metrics per player per game which may not necessarily be available on the internet and thus may be able improve the accuracy of the model even further.

The application of this model is not limited to just football players. The theory can be extended to any sort of competition or sport which involves the buying and selling of sportspersons. Naturally the predictors will have to changes to as to reflect the performance of the players in that particular sport. An example is the Indian Premier League which is the world's most popular cricketing competition involving privately owned clubs. Similar to the transfer window , the IPL has a bidding process before the beginning of every season. By collecting attributes of the participating cricket players, the owners of the club can make a bid based on the player's actual performance and in this way they may be able to prevent themselves from overspending on certain players.

8. References and Bibliography

1. https://www.pledgesports.org/2017/06/top-10-most-popular-sports-in-the-world-by-participation/
2. https://medium.com/@Thorpeskii/many-lovers-of-the-beautiful-game-of-football-still-cant-wrap-their-heads-around-how-clubs-afford-d6a2f5c93d36
3. https://www.sportskeeda.com/football/how-do-football-clubs-make-money#:~:text=The%20Deloitte%20report%20for%20the,Bernabeu%20or%20at%20Old%20Trafford
.
4. https://www.transfermarkt.com/transfers/einnahmenausgaben/statistik/
5. https://en.wikipedia.org/wiki/List_of_most_expensive_association_football_transfers
6. https://journals.sagepub.com/doi/pdf/10.1177/875647930831700
7. https://www.goal.com/en-ae/news/fifa-player-ratings-explained-how-are-the-card-number-stats/1hszd2fgr7wgf1n2b2yjdpgynu
8. https://medium.com/swlh/cross-validation-techniques-to-assess-your-models-stability-3a4d55d90409#:~:text=Cross%2Dvalidation%20is%20a%20statistical,is%20not%20used%20for%20training.
9. https://medium.com/datadriveninvestor/k-fold-cross-validation-6b8518070833
10. https://en.wikipedia.org/wiki/Heteroscedasticity
11. https://www.statisticshowto.com/lasso-regression/

12. https://sebastianraschka.com/faq/docs/parametric_vs_nonparametric.html#:~:text=So%2C%20in%20a%20parametric%20model,parameters%20is%20(potentially)%20infinite.&text=So%2C%20in%20intuitive%20terms%2C%20we,quasi)%20assumption%2Dfree%20model.
13. https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205
14. https://www.mygreatlearning.com/blog/gradient-boosting/
15. https://rcarneva.github.io/understanding-gradient-boosting-part-1.html

9. Appendix

1. Appendix A: Link to all codes – https://drive.google.com/drive/folders/1mkWAiYqV1skwfOH3vRhr4tX_dImSb902?usp=sharing