

|  |
| --- |
| NBA Historical Data Mining |
|  |
| April 19  Group 8  Authored by: Your Name |

# Table of Contents

[Table of Contents 2](#_Toc38222409)

[Executive Summary 3](#_Toc38222410)

[Background & Motivation 4](#_Toc38222411)

[Data Description 7](#_Toc38222412)

[NBA Player Season dataset 7](#_Toc38222413)

[MVP Dataset 9](#_Toc38222414)

[Business Intelligence Model: 10](#_Toc38222415)

[VIEW DATA 10](#_Toc38222416)

[DATA CLEANING 10](#_Toc38222417)

[DATA MERGING 10](#_Toc38222418)

[CHECK CORRELATION BETWEEN DATA 10](#_Toc38222419)

[THE MODEL USED FOR PREDICTIONS OF MVP 10](#_Toc38222420)

[Major R Code 16](#_Toc38222421)

[Discussions and Conclusions 24](#_Toc38222422)

[References 25](#_Toc38222423)

# Executive Summary

This report aims at taking advantage of second-hand National Basketball Association (NBA) historical datasets and using different data mining techniques to measure the performance of a player in a single year and trying to create a useful model to predict whether that player could win the most valuable player (MVP) award.

The MVP awards are a very prestigious award in the NBA community and due to its reach in the world between fans, it becomes an important decision making factor in a lot of commercial and professional deals.

In this report, the following efforts that engage in data mining, including:

* Retrieve the related historical player stats and MVP datasets online
* Preprocessing individual dataset
* Merge different datasets
* Check correction between data
* Generate, train and validate the MVP prediction model

**Keywords**: NBA, MVP, data mining, datasets, prediction, decision

# Background & Motivation

The MVP Award is an annual NBA award given since the 1955–56 season to the best performing player of the regular season. As one of the most prestigious awards, the debate regarding who will be the next MVP player is always part of the hot topic among NBA fans and media.

The motivation behind the project was to introduce Machine learning power to the NBA world and help make a very crucial decision like below which based on the NBA MVP awards:

* The business value of this means that this model can be used to predict MVP even before the real results based on the seasonal performance basis. Thus, the managers of the teams can retain a player or sign a new contract with him before getting the MVP award which will save a lot of money as suddenly the fees and rates of contracts increase with the market value of the Player.
* The Betting industry earns a lot, thereby using this prediction model adds some statistics to the Bets.
* The way model generates a value for Team owners and NBA broadcasters make it a very beneficial product for the NBA context as a whole. Because other than Championship, MVP is the biggest award of the season and a lot of things like player salary, player contract, retaining a player decision becomes very important and depends on the MVP result. If this can be predicted earlier before the end of the season, then this can be said that it will save a lot of money for a lot of people.
* Commercial side opportunities for the prediction of awards are limitless.
* Other than for a team owner in NBA, lots of big business companies have NBA famous stars as there Brand Ambassador. A decision can be made using this model before the launch of new products if the player is good, but maybe not winning the MVP this year than launching a product after the award announcement may lead to affect the sales of the product. This situation can be anticipated earlier.
* A similar new MVP can be predicted and signed before the award declaration, which can lead to low-cost deals and the early decision of launching products with correct players before competitors.

The benefit and challenge of predicting the MVP player are apparent, especially in the following aspects:

* NBA fans are likely to make a comparison between players whether they have the best performance in the past regular seasons. And people may curious about who has the best chance to win the MVP award in the current season. For example, “Lebron James” vs. “Giannis Antetokounmpo” in the 2019–2020 season.
* Predicting the MVP player is a complicated thing since there are no obvious criteria to decide whether a player could be considered as an MVP potential winner. Besides, the voting system was not even consistent in the past. For example, in the 1979–80 season, the MVP was selected by a vote of NBA players. However, the award is decided by a panel of sportswriters and broadcasters since the 1980–81 season [1].

Thanks to the increasing comprehensive data in NBA historical statistics, different data mining techniques can be applied to find hidden but valuable knowledge from the past and to impact the sports industry in the future. This project mainly focuses on how to utilize the Random Forest algorithm to create a robust model for predicting the NBA MVP award winner.

# Data Description

## **NBA Player Season dataset**

(filename: Seasons\_Stats.csv )

The data-set contains aggregate individual statistics for 67 NBA seasons. From basic box-score attributes such as points, assists, rebounds etc., to more advanced money-ball like features such as Value Over Replacement. [2] The “seasons\_ststs: dataset contains 53 variables and over 24,000 observations. The variables in the data set are shown below with the appropriate roles and levels:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Model Role | Data Type | Description |
| Index | ID | Categoric | index |
| Year | Input | Categoric | Season |
| Player | Input | Categoric | Player Name |
| Pos | Input | Categoric | Position |
| Age | Input | Numeric | Age |
| Tm | Input | Categoric | Team |
| G | Input | Numeric | Games |
| GS | Input | Numeric | Games Started |
| MP | Input | Numeric | Minutes Played |
| PER | Input | Numeric | Player Efficiency Rating |
| TS% | Input | Numeric | True Shooting % |
| 3PAr | Input | Numeric | 3-Point Attempt Rate |
| FTr | Input | Numeric | Free Throw Rate |
| ORB% | Input | Numeric | Offensive Rebound Percentage |
| DRB% | Input | Numeric | Defensive Rebound Percentage |
| TRB% | Input | Numeric | Total Rebound Percentage |
| AST% | Input | Numeric | Assist Percentage |
| STL% | Input | Numeric | Steal Percentage |
| BLK% | Input | Numeric | Block Percentage |
| TOV% | Input | Numeric | Turnover Percentage |
| USG% | Input | Numeric | Usage Percentage |
| blanl | Reject | N/A | empty |
| OWS | Input | Numeric | Offensive Win Shares |
| DWS | Input | Numeric | Defensive Win Shares |
| WS | Input | Numeric | Win Shares |
| WS/48 | Input | Numeric | Win Shares Per 48 Minutes |
| blank2 | Reject | N/A | empty |
| OBPM | Input | Numeric | Offensive Box Plus/Minus |
| DBPM | Input | Numeric | Defensive Box Plus/Minus |
| BPM | Input | Numeric | Box Plus/Minus |
| VORP | Input | Numeric | Value Over Replacement |
| FG | Input | Numeric | Field Goals |
| FGA | Input | Numeric | Field Goal Attempts |
| FG% | Input | Numeric | Field Goal Percentage |
| 3P | Input | Numeric | 3-Point Field Goals |
| 3PA | Input | Numeric | 3-Point Field Goal Attempts |
| 3P% | Input | Numeric | 3-Point Field Goal Percentage |
| 2P | Input | Numeric | 2-Point Field Goals |
| 2PA | Input | Numeric | 2-Point Field Goal Attempts |
| 2P% | Input | Numeric | 2-Point Field Goal Percentage |
| eFG% | Input | Numeric | Effective Field Goal Percentage |
| FT | Input | Numeric | Free Throws |
| FTA | Input | Numeric | Free Throw Attempts |
| FT% | Input | Numeric | Free Throw Percentage |
| ORB | Input | Numeric | Offensive Rebounds |
| DRB | Input | Numeric | Defensive Rebounds |
| TRB | Input | Numeric | Total Rebounds |
| AST | Input | Numeric | Assists |
| STL | Input | Numeric | Steals |
| BLK | Input | Numeric | Blocks |
| TOV | Input | Numeric | Turnovers |
| PF | Input | Numeric | Personal Fouls |
| PTS | Input | Numeric | Total Points |

## **MVP Dataset**

(filename: MVP\_ESPN\_Data.csv)

The data-set [3] contains the summary of necessary player information of an MVP award winner from 1956 to 2019 seasons. It contains 9 variables and over 64 observations. The variables in the data set are shown below with the appropriate roles and levels:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Model Role | Data Type | Description |
| YEAR | Input | Categoric | Season |
| PLAYER | Input | Categoric | Player Name |
| POS | Input | Categoric | Position |
| TEAM | Input | Categoric | Team |
| FG% | Input | Numeric | Field Goal Percentage |
| PPG | Input | Numeric | Point Per Game |
| RPG | Input | Numeric | Rebound Per Game |
| APG | Input | Numeric | Assist Per Game |
| BLKPGG | Input | Numeric | Block Per Game |

From the above description, it is clear that the data is in Per year per game format and all other data analysis was done on this data set check the frequency and stretch of each data attribute to solve selecting final variables which can lead to appropriate training of the data.

# Business Intelligence Model:

## **VIEW DATA**

## **DATA CLEANING**

## **DATA MERGING**

## **CHECK CORRELATION BETWEEN DATA**

## **THE MODEL USED FOR PREDICTIONS OF MVP**

VIEW DATA**:**

Exploratory Data Analysis:

In this, we view the data using different graphs and exploratory techniques like histograms and correlation plots of the attributes of the data frame.

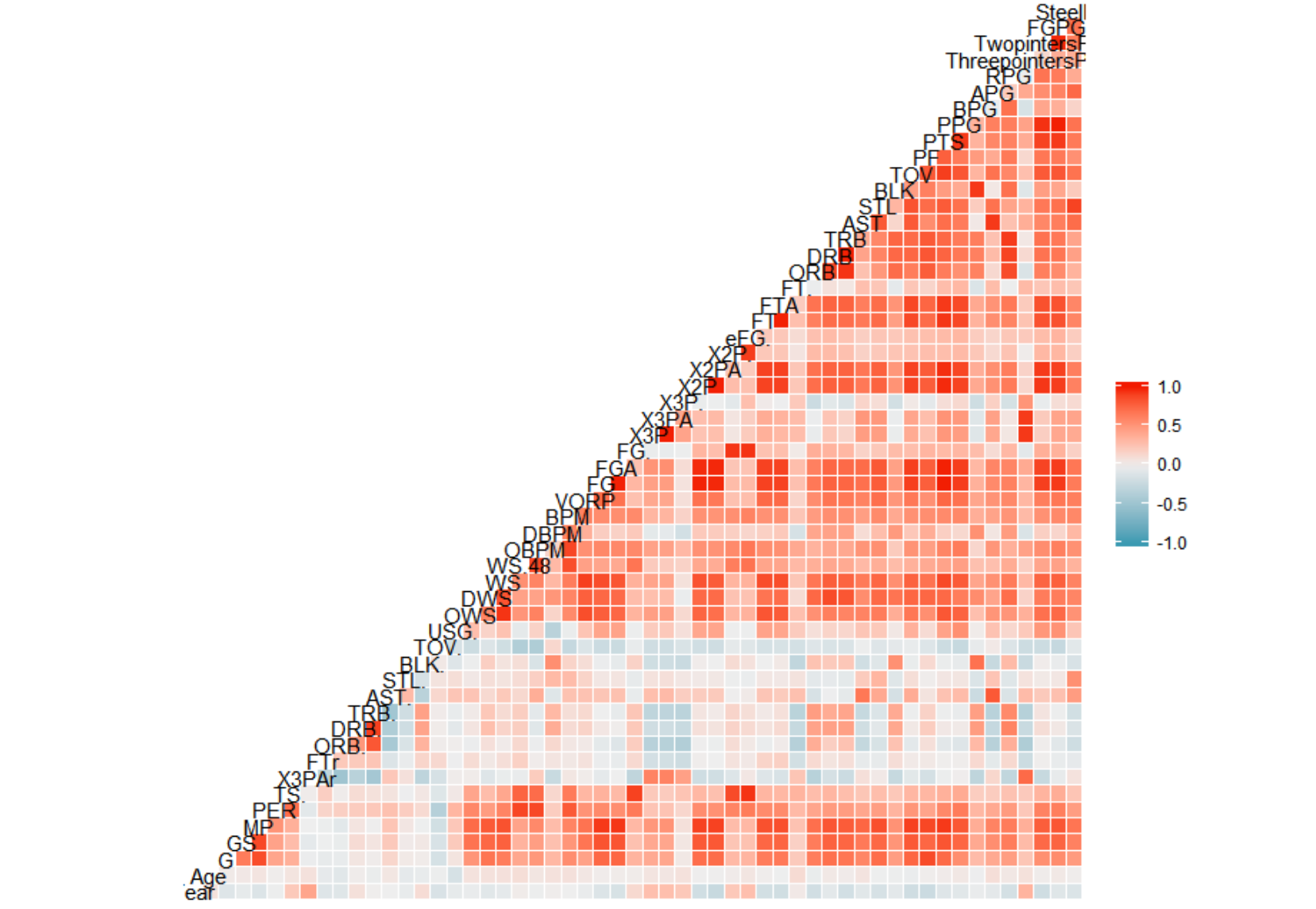
It is important to do Data exploration to more about the data. We can check for Missing Values or sparse data in the Data frame. E.g. **NA**. This helps us in understanding those attributes and decide if the attribute is important than we replace the sparse value from the mean or median of the attributes.

For the Factor, data attributes missing values are generally replaced with the highest frequency factor. Factors in the data set are generally are the Target variable for the prediction models.

When we merge the data frames and create a single data frame with all the numeric variables but one factor which is in the binary output form of **1** or **0**. In out data frame we have taken **MVP** as the column having factor value and thus acting as the **Target variable** for the current model to predict.

Correlation Plot the Stats table:

This plot covers the correlation between data frame attributes which are numeric and all the factors and characters are automatically ignored.

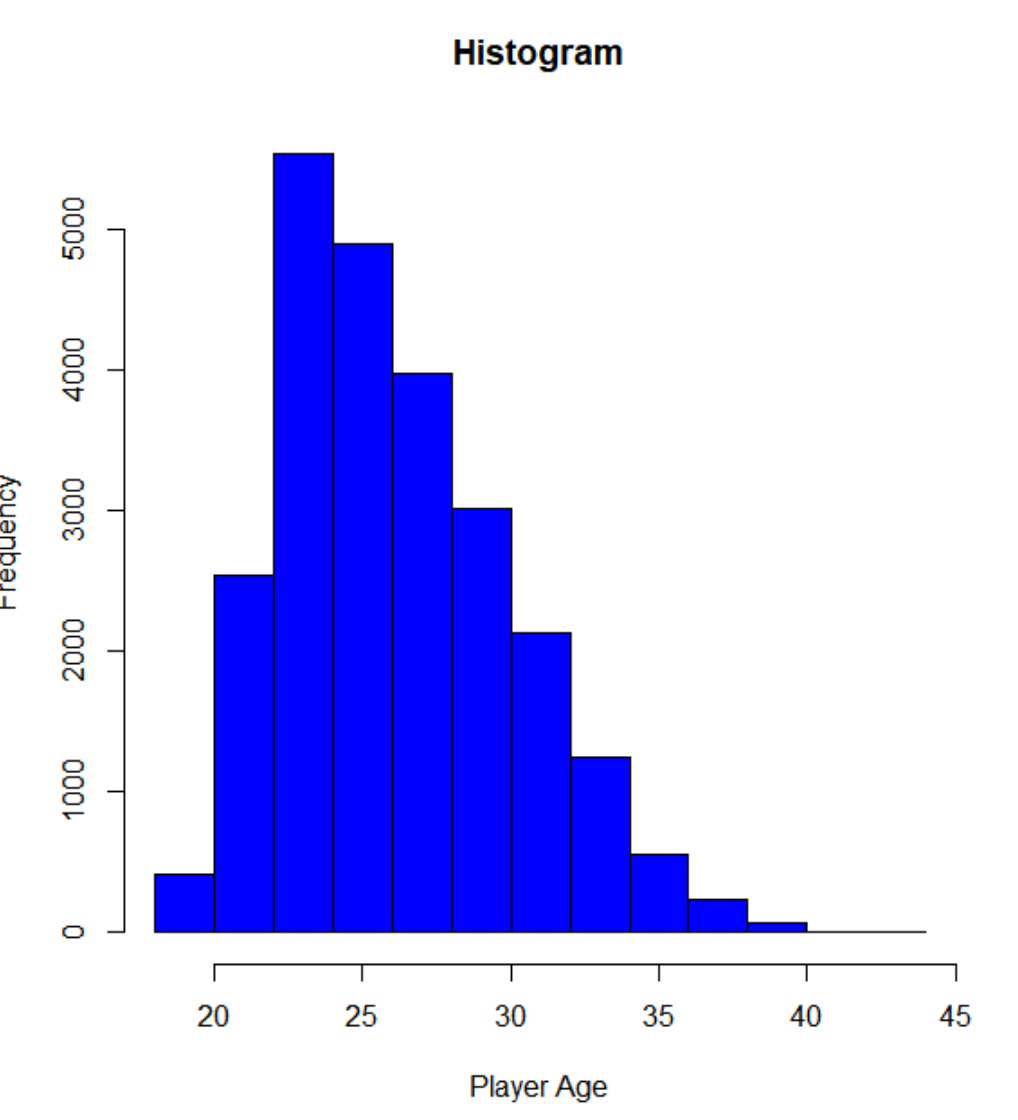
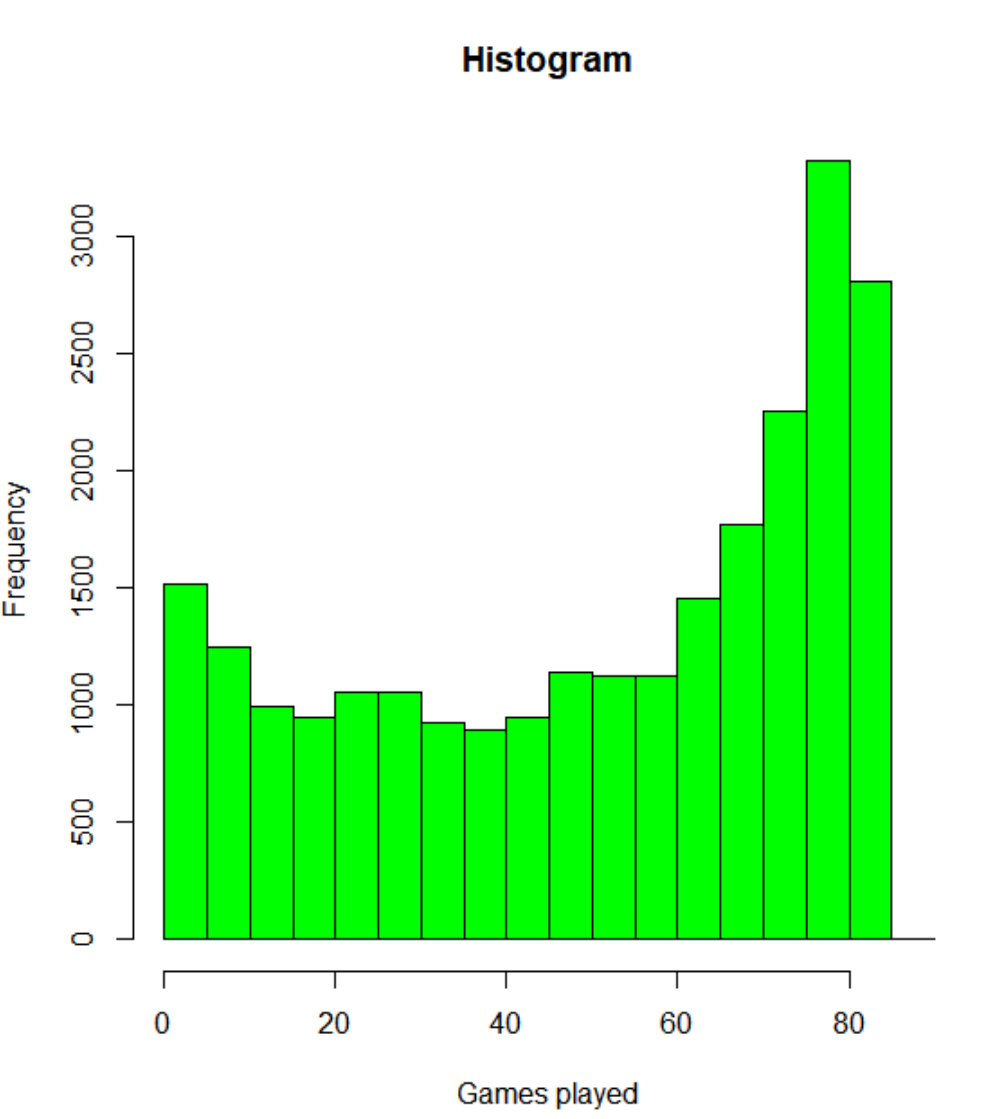


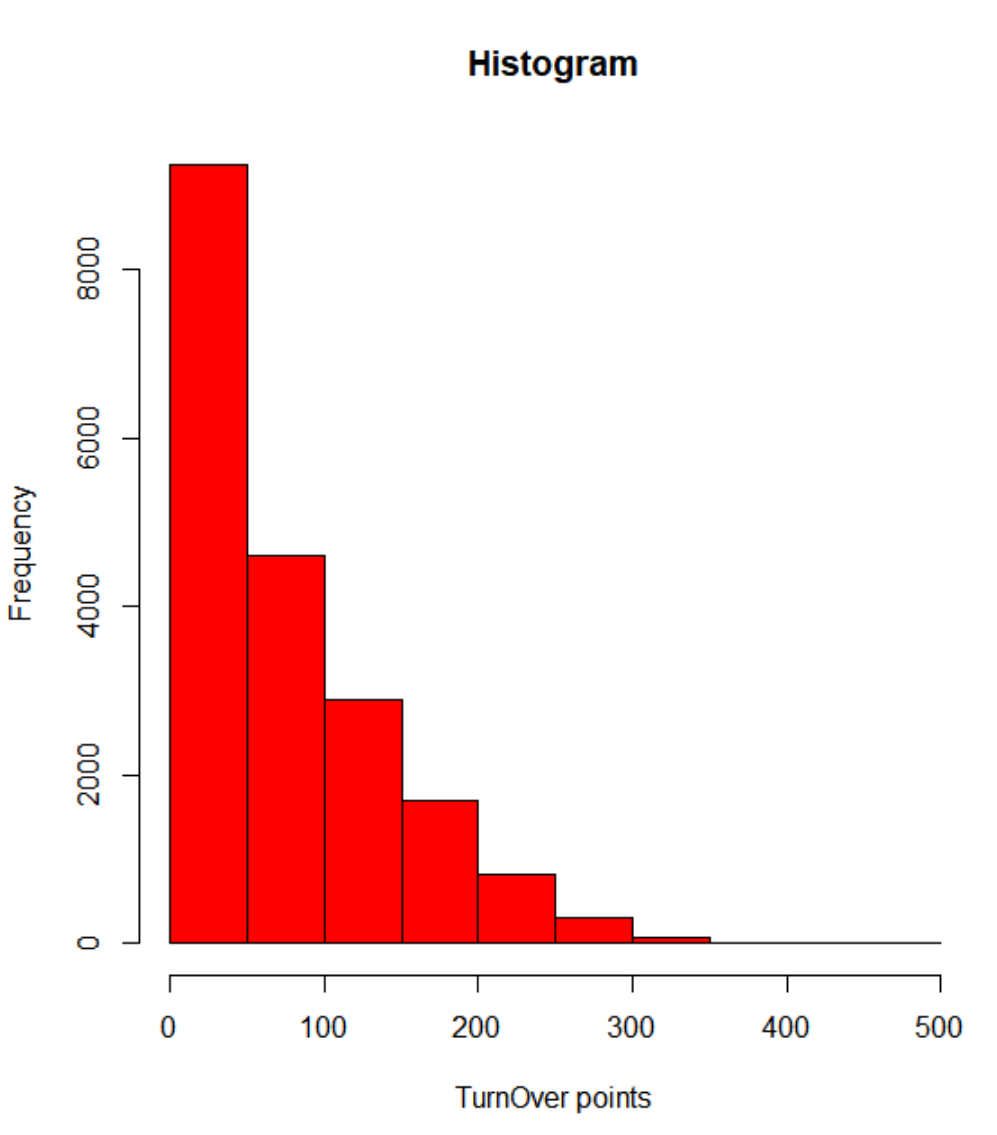
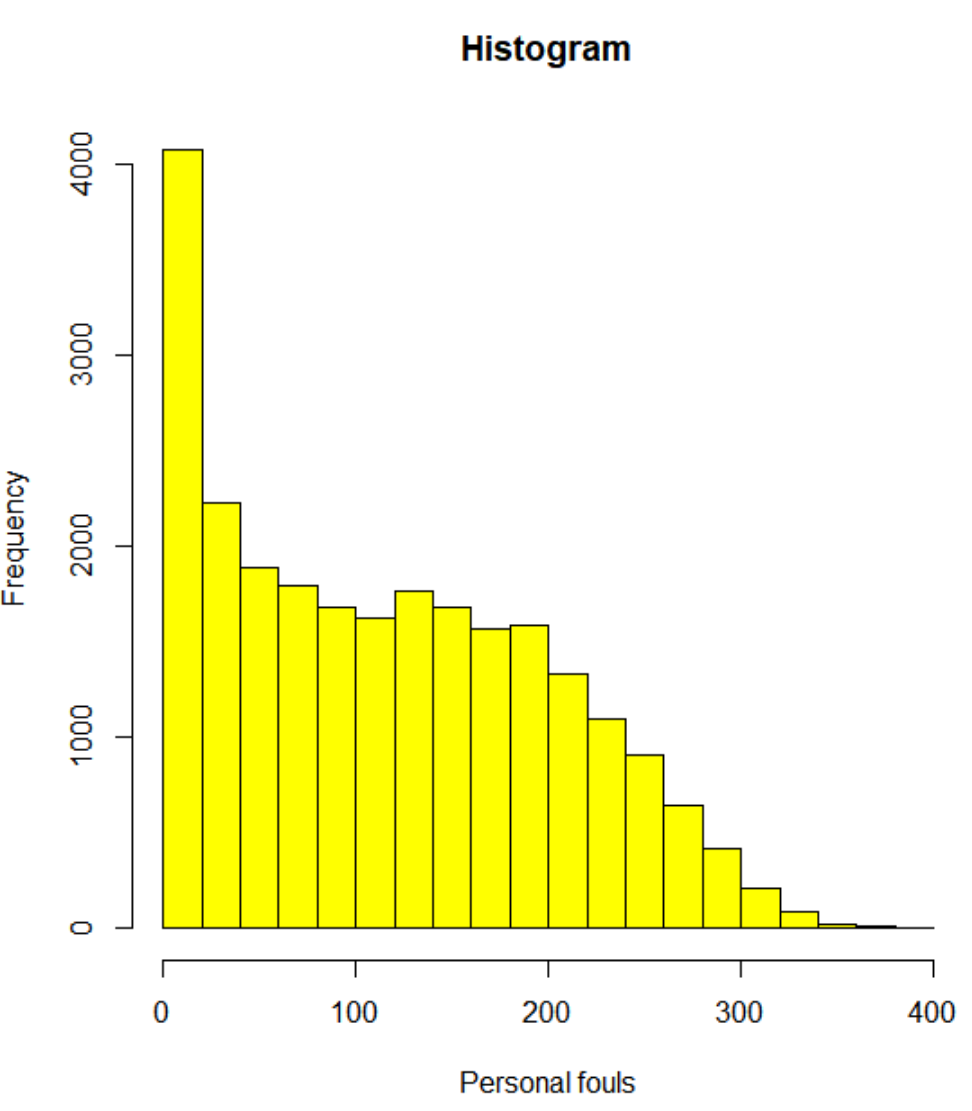
We Can see from this that not all the 50 columns are useful for the data analysis.

This plot of correlation depicts a very simple correlation between each attribute and we can see in the plot that plots with dark red are highly positively related whereas the attributes having light blue are inversely related to other attributes.

We will further analyze some single attributes by there count.

Histogram plots of some attributes:

**DATA CLEANING:**

In this project, we have taken the Mean of the attributes as the numeric values that will replace the missing numeric values in the data set.

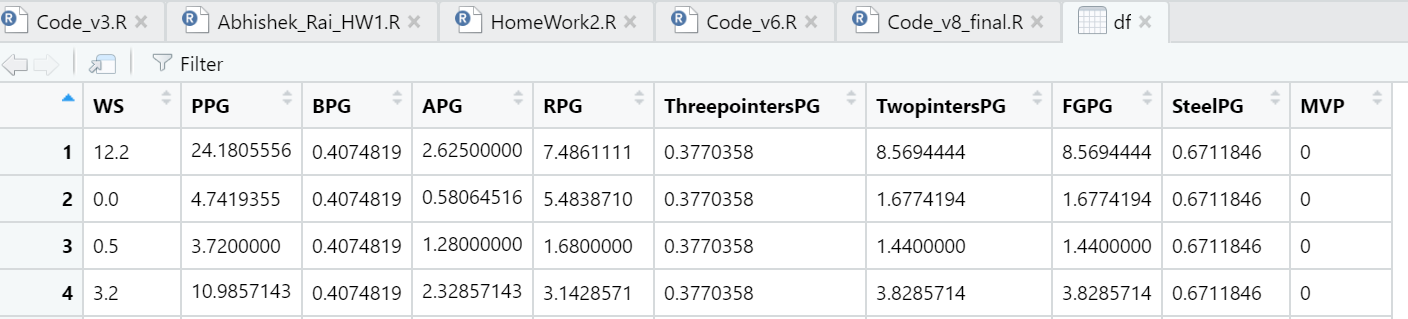
Similarly, for the factor, the missing values are replaced with the level of highest frequency in that data set attribute.

For fields like missing Player names and Years, it is considered to be incorrect data and the rows will be removed from the data frame.

For all the attributes other than win Share that is WS, normal rules of data mutation is applied and for WS we have replaced the missing values with zero cause replacing this with a mean will highly impact the data authenticity as this is a highly correlated value which impacts the delivery of the model.

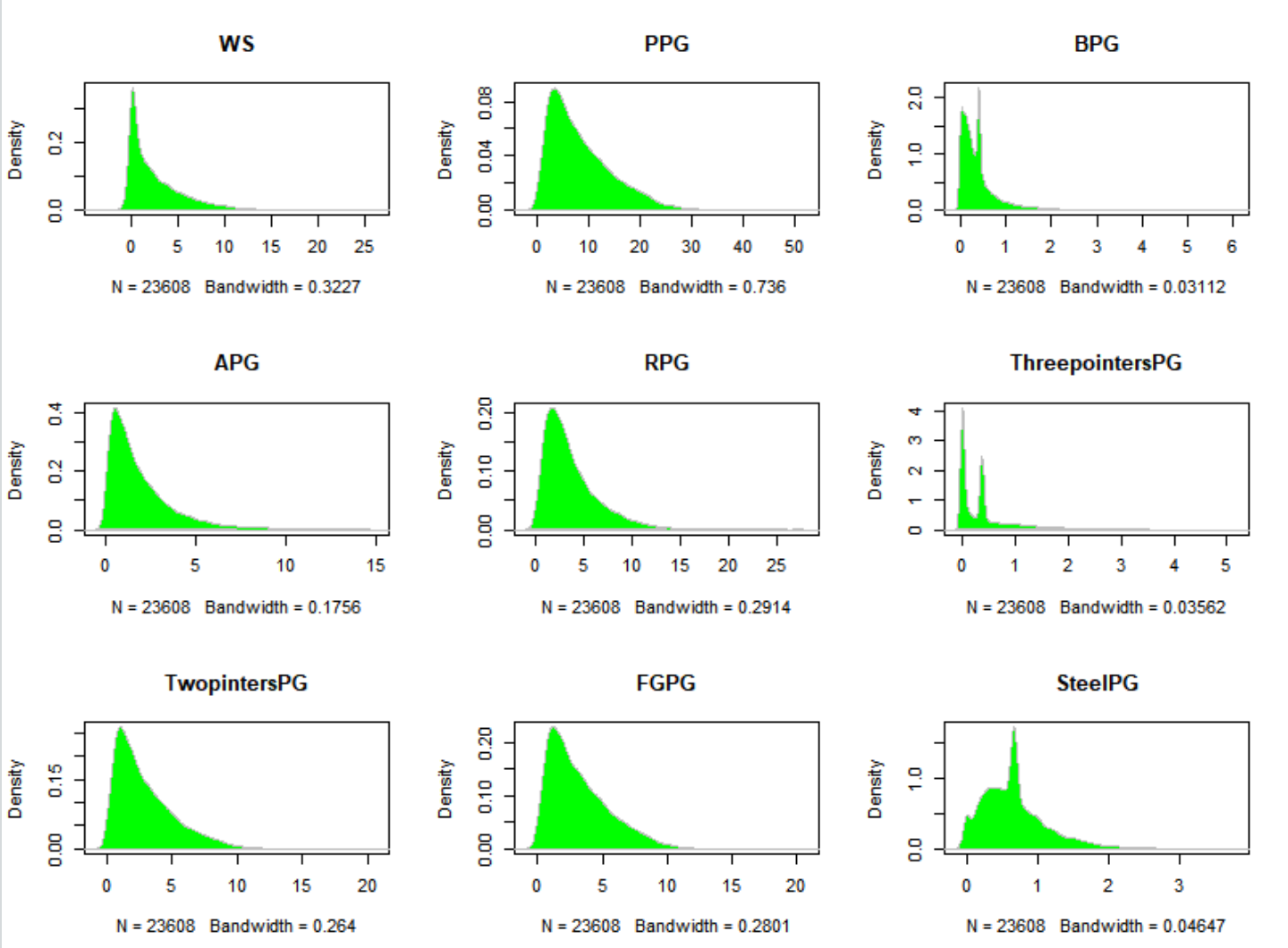
**DATA MERGING (SEASONS STATS AND MVP STATS):**

Df is the merged data set.

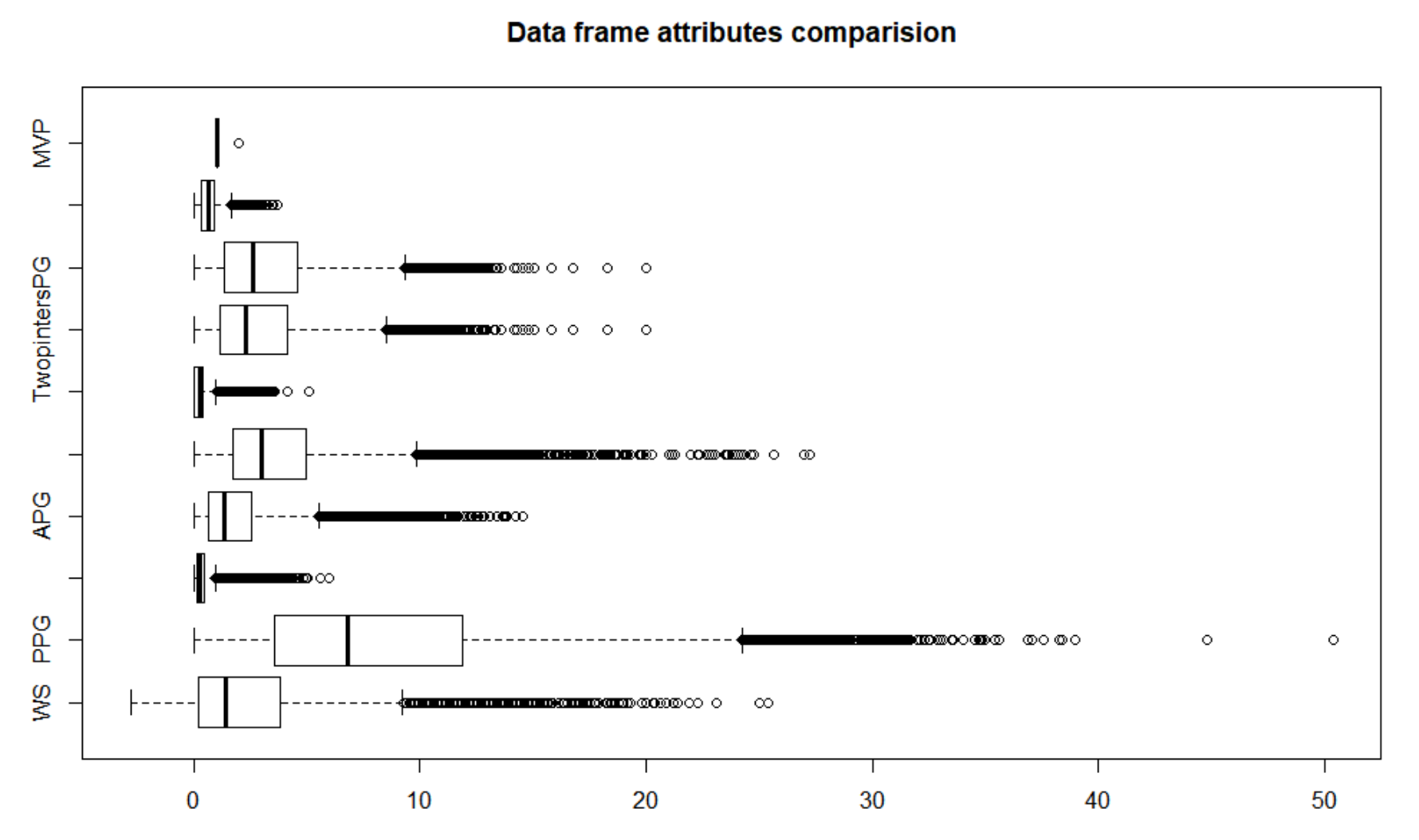


**CHECK A CORRELATION BETWEEN DATA:**

Density plot of the final data set attributes:

****

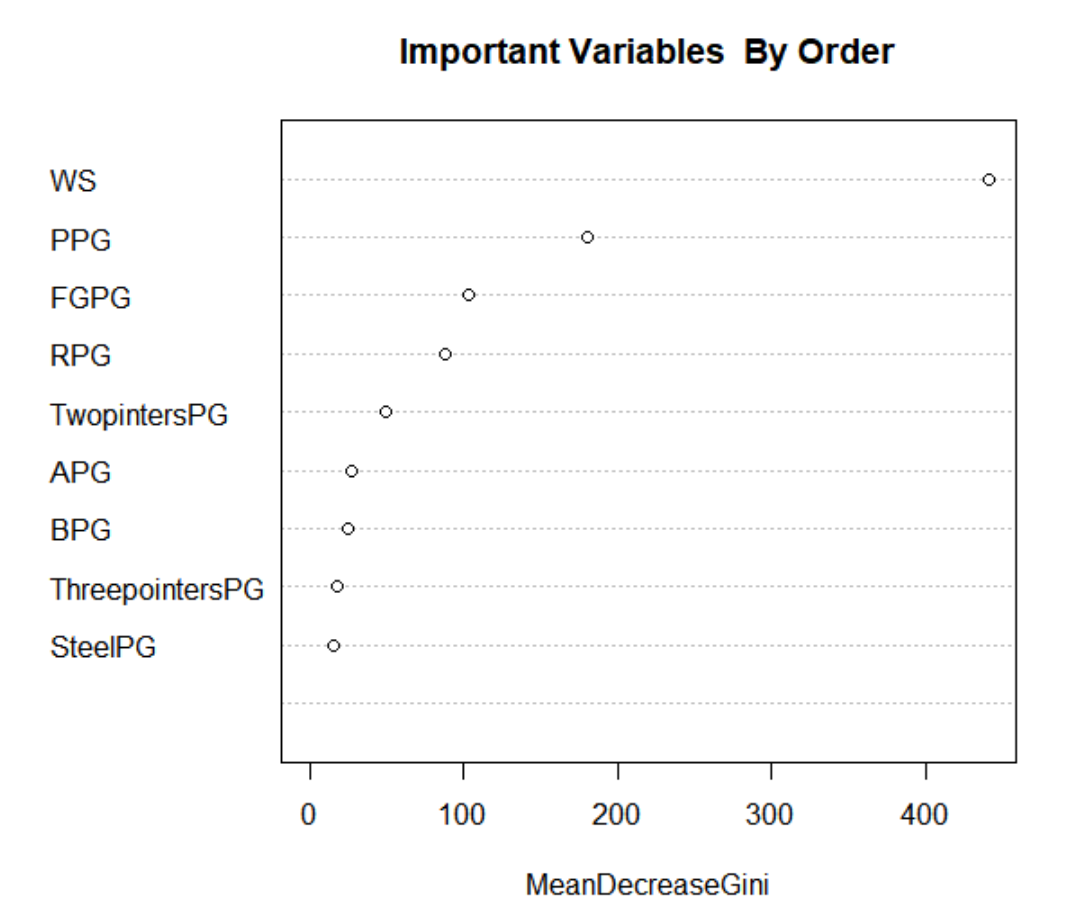
We will do the box plot for all the elements side by as we are passing all of them in the prediction function.



**THE MODEL USED FOR PREDICTIONS OF MVP :**

The model that we are using is **Random forest** [4], and we have reached this decision after comparing the models based on the confusion matrix and other comparable scenarios like the imbalanced data, etc.

In random forest scheme, features are very much crucial for the prediction as we can see the importance plot for all the variables below.



Another essential aspect of the Random forest scheme is not assuming the linear relationship between attributes. Still, it counts on the randomness and therefore proves to be better for a non-linear type of predictions.

In the random forest, we take random samples and use many decision trees to average out the output to better fit the data and predict as accurately as possible. We are taking 100 tree samples for this data set as one of our input for the random forest algorithm.

Before using the algorithm, we need to balance the training data set using the **Synthetic Minority Over-Sampling Technique (SMOTE) [5], [6].** We use this method to balance the data because to avoid overfitting, which is caused by other methods like oversampling and undersampling.

In this method, we are creating the replica of the minority instances in the main dataset using the feature space and therefore leading to no loss of information while creating a balanced data set.

# Major R Code

#Code for the Business Proposal regarding the Prediction of the MVP in NBA based on the various parameters

##Libraries required

**library("stringr")**

**library("caret")**

**library("caTools")**

**library("DMwR")**

**library("doParallel")**

**library("dplyr")**

**library("e1071")**

**library("foreach")**

**library("GGally")**

**library("ggplot2")**

**library("gplots")**

**library("iterators")**

**library("lattice")**

**library("mlr")**

**library("ParamHelpers")**

**library("plyr")**

**library("pROC")**

**library("randomForest")**

**library("ROCR")**

**library("rpart.plot")**

**library("scales")**

**library("stringi")**

**library("tidyr")**

**library("tidyselect")**

**library("tibble")**

**library("unbalanced")**

##################################################

#Importing the data based on 2 different sources

#1.This contains the historical NBA data for each player

**season\_stat<- read.csv("Seasons\_Stats.csv")**

#2.This contains the data of MVP players each year since the award originated.

**mvpdata<- read.csv("MVP\_ESPN\_Data.csv")**

################Cleaning and updating the data as per our requirements,

#so that it can be used to train the model.

#Removing Serial number

**season\_stat<-season\_stat[,-c(1)]**

#changing the data to per game basis

**season\_stat$PPG<-season\_stat$PTS/season\_stat$G**

**season\_stat$BPG<- season\_stat$BLK/season\_stat$G**

**season\_stat$APG<- season\_stat$AST/season\_stat$G**

**season\_stat$RPG<- season\_stat$TRB/season\_stat$G**

**season\_stat$ThreepointersPG<- season\_stat$X3P/season\_stat$G**

**season\_stat$TwopintersPG<-season\_stat$X2P/season\_stat$G**

**season\_stat$FGPG<- season\_stat$FG/season\_stat$G**

**season\_stat$SteelPG<-season\_stat$STL/season\_stat$G**

#analyzing data to know the right features for our model

**hist(season\_stat$Age,col = "BLUE",xlab = "Player Age",main = "Histogram")**

**hist(season\_stat$G,col = "green",xlab = "Games played",main = "Histogram")**

**hist(season\_stat$TOV,col = "Red",xlab = "TurnOver points",main = "Histogram")**

**hist(season\_stat$PF,col = "yellow",xlab = "Personal fouls",main = "Histogram")**

#correlation table

**ggcorr(season\_stat)**

#keeping the important attributes only

**Season\_MVP<- season\_stat[,-c(3:23)]**

**Season\_MVP<- Season\_MVP[,-c(4:31)]**

#changing Column names in the MVP tables

**names(Season\_MVP)[1]<-paste("YEAR")**

**names(Season\_MVP)[2]<- paste("PLAYER")**

#Checking data type of Variables

**Season\_MVP<- Season\_MVP %>% mutate(WS = replace(WS,is.na(WS),0),**

**PPG = replace(PPG,is.na(PPG),mean(PPG,na.rm=T)),**

**BPG = replace(BPG,is.na(BPG),mean(BPG,na.rm=T)),**

**APG = replace(APG,is.na(APG),mean(APG,na.rm=T)),**

**RPG= replace(RPG,is.na(RPG),mean(RPG,na.rm=T)),**

**ThreepointersPG = replace(ThreepointersPG,is.na(ThreepointersPG),mean(ThreepointersPG,na.rm=T)),**

**TwopintersPG = replace(TwopintersPG,is.na(TwopintersPG),mean(TwopintersPG, na.rm = T)),**

**FGPG = replace(FGPG,is.na(FGPG),mean(FGPG, na.rm = T)),**

**SteelPG = replace(SteelPG,is.na(SteelPG),mean(SteelPG,na.rm=T)))**

**Season\_MVP<- Season\_MVP[!(is.na(Season\_MVP$YEAR)),]**

**Season\_MVP$PLAYER<-str\_remove\_all(Season\_MVP$PLAYER,"[\*]")**

**mvpdata<-mvpdata[,-c(3:9)]**

**names(mvpdata)[1]<-paste("YEAR")**

**names(mvpdata)[2]<-paste("PLAYER")**

##########################################

#merging data frames to create one merged data set.

**stats\_merged<-merge(Season\_MVP,mvpdata,by.x = "YEAR", by.y="YEAR", incomparables = NA)**

#changing df name so dont have to load the data again and again

**stats\_merged\_1<-stats\_merged**

#to add a MVP (1/0) column in the merged data set

**stats\_merged\_1 <- stats\_merged\_1 %>%**

**mutate(MVP = ifelse(as.character(stats\_merged\_1$PLAYER.x)==**

**as.character(stats\_merged\_1$PLAYER.y),**

**1, 0))**

**table(stats\_merged\_1$MVP)**

**df<-stats\_merged\_1[,-c(1,2,12)]**

**df$MVP<-as.factor(df$MVP)**

**summary(df)**

###few charts for df comparisions of attributes

#to give the idea about the data set

#Correlation using ggplots

**ggcorr(df)**

# Box plots for all Attributes

**boxplot(df, horizontal=TRUE, main="Data frame attributes comparision")**

# Density plot

**par(mfrow=c(3, 3))**

**colnames <- dimnames(df)[[2]]**

**for (i in 1:10) {**

**d <- density(df[,i])**

**plot(d, type="n", main=colnames[i])**

**polygon(d, col="green", border="gray")**

**}**

#df is the final data set on which we will work data set, which is highly imbalanced.

#############################################

#dividing data in test and train for the the model and prediction.

#setting seed

**set.seed(1123)**

#splitting data in 75% into training set and 25% into Test Data set

**split <- sample.split(df$MVP, SplitRatio = 0.75)**

#train data

**TrainMVP\_Data <- subset(df, split == TRUE)**

#test data

**TestMVP\_Data <- subset(df, split == FALSE)**

## check the count of unique value in the MVP column

**as.data.frame(table(TrainMVP\_Data$MVP))**

##########################################################

#BALANCING DATA USING Synthetic Minority Oversampling Technique

**set.seed(75252)**

**balanced.data <- SMOTE(MVP ~., TrainMVP\_Data, perc.over = 1000, k = 5, perc.under = 2600)**

#This gives us the frequency of MVP data in the balanced data set.

**as.data.frame(table(balanced.data$MVP))**

##########################################################

#model Generation

#now using random forest to implement model (we use comparision techniques to find the best model)

**rf = randomForest(MVP~.,**

**ntree = 100,**

**data = balanced.data)**

#####

#to plot the importance of each variable use

**plot(rf)**

#to check the importance of each attribute

**varImp(rf)**

#plot the importance

**varImpPlot(rf,**

**sort = T,**

**n.var=10,**

**main="Important Variables By Order")**

##############################################

#now using the rf model to predict

**predicted.response <- predict(rf, TestMVP\_Data)**

#use this to compare models.

**confusionMatrix(data=predicted.response,**

**reference=TestMVP\_Data$MVP)**

######prediction on Training data to check confusion matrix.

**pred.resp<-predict(rf,TrainMVP\_Data)**

**confusionMatrix(data = pred.resp,reference = TrainMVP\_Data$MVP)**

#################################

# The last step is to create a Function to use the above model to predict value based on parameters.

**Predict\_MVP\_func<-function(m,WS,PPG,BPG,APG,RPG,ThreepointersPG,**

**TwopintersPG,FGPG,SteelPG)**

**{**

**call\_param\_2<-predict(m,data.frame(WS=WS,PPG=PPG,BPG=BPG,APG=APG,RPG=RPG,**

**ThreepointersPG=ThreepointersPG,**

**TwopintersPG=TwopintersPG,**

**FGPG=FGPG,SteelPG=SteelPG))**

**#print the message**

**message<-paste(ifelse(call\_param\_2=='1',"Player has the very high Chances of becoming an MVP",**

**"Player has High chances of NOT becoming an MVP"))**

**print(message)**

**}**

#Passing the values in fucntion to check the player's chances.

#Prediction 1

**Predict\_MVP\_func(rf,21.2, 0, 1.59, 0,**

**0, 0.08 ,12.9, 13.03, 3.1)**

#Prediction 2

**Predict\_MVP\_func(rf,11.3,16.5, 0.4, 2.9,**

**22.67, 0.37, 6.6, 6.6, 0.67)**

#prediction 3

**Predict\_MVP\_func(rf,10,12, 0.40, 1.8,**

**23, 0.3, 6.608, 8.3, 0.78)**

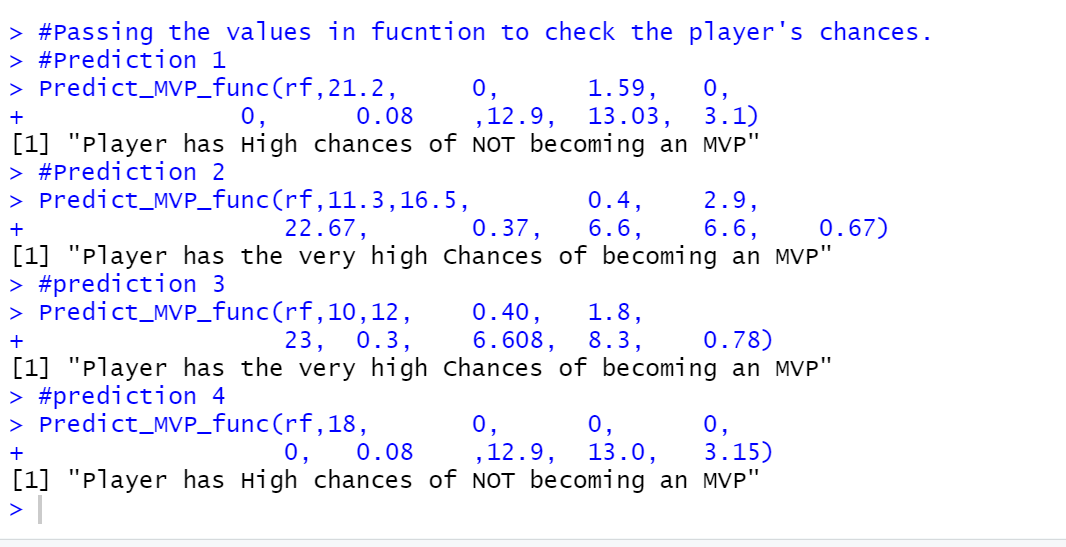
#prediction 4

**Predict\_MVP\_func(rf,18, 0, 0, 0,**

**0, 0.08 ,12.9, 13.0, 3.15)**

###############End of Project

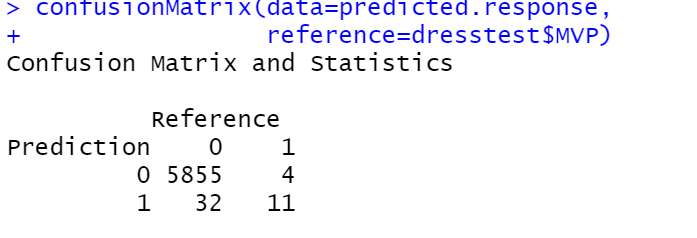
**OUTPUT:**



# Discussions and Conclusions

For the above model, we can predict the output of the chances that a Player can be MVP for this season or not.

We can use the confusion matrix to check the performance.



In our model, we are predicting a great number of true positives and true negatives. But the main selling point of our model is its false-positive rate which is very low. This result saves the client from making wrong investments, which are a major problem for the sports Industry. This is because the players are bought and signed for long term contracts and if the player was bought with intentions of being an MVP and it turns out the player is not MVP in that case the losses are major.

Similarly with the Sports industry if they are signing someone to be their brand ambassador than they should be sure that the player will be MVP if predicted yes because these deals are huge in numbers and are signed for long terms.

Thus this model can help the Sports industry as a whole and signing players becomes easy and therefore can lead to saving a lot more money in the deal.

# References

[1], NBA Most Valuable Player Award, Wikipedia <https://en.wikipedia.org/wiki/NBA_Most_Valuable_Player_Award>

[2], NBA Players stats since 1950, Kaggle

<https://www.kaggle.com/drgilermo/nba-players-stats>

[3], NBA History – MVP, ESPN

<http://www.espn.com/nba/history/awards/_/id/33>

[4], Random forest, Wikipedia

<https://en.wikipedia.org/wiki/Random_forest>

[5], Dealing with unbalanced data in machine learning

<https://www.r-bloggers.com/dealing-with-unbalanced-data-in-machine-learning/>

[6], Dealing with Imbalanced Data

<https://towardsdatascience.com/methods-for-dealing-with-imbalanced-data-5b761be45a18>