**USE CASE STUDY REPORT**

**Group No**.: 9

**Student Names**: Anantha Nithin Raj, Mohammed Rizwanullah

Executive Summary:

**Objective**: To create a multiple regression model to perform the salary prediction problem.

NBA is a very popular sport here in North America and the most recognized players get paid a good amount of salary if their stats are good. The player stats include his turnover rate, points scored, fouls committed, assists etc. during the game. If we are working as an analyst in the management department and we want to predict their salaries to estimate the salary cap we need to understand the dynamics of their performance and conclude the results. The managers are always on the hot spot when it comes to determine the player to be retained and the salary cap to be drafted as that changes the fate of the team and increases its chances of winning the championship. In this case study we were focused on creating a regression model to predict the salary cap for the players. Correlation is also a major metric we considered in this case to identify the relation between dependent variables and independent variables. The predicted salary is not always accurate as there are certain outliers every time which alter the other way. So, we can check the accuracy by taking these into consideration.

# Background and Introduction

The National Basketball Association (NBA) is a men's professional basketball league in North America, composed of 30 teams (29 in the United States and 1 in Canada). It is one of the four major professional sports leagues in the United States and Canada and is widely considered to be the premier men's professional basketball league in the world. NBA is recognized by FIBA which the governing body of basketball in united states. Each team has 5 members each and the team may have as many substitutions as they wish. Each team has 24 seconds to have a shot if they fail to do so the clock is restarted again.

The NBA salary cap is the limit to the amount of money that NBA teams can pay their players. Like many professional sport leagues, the NBA has a salary cap to control cost and benefit parity, defined by the league's collective bargaining agreement (CBA). This limit is subject to a complex system of rules and exceptions and is calculated as a percentage of the league's revenue from the previous season. The CBA ratified in July 2017 that the cap will continue to vary in future seasons based on league revenues. For the 2019–20 season, the cap is set at $109.14 million.

Most leagues (NFL, NHL, MLS) have hard caps while the NBA has a soft salary cap. Hard salary caps forbid teams from going above the salary cap. Soft salary caps allow teams to go above the salary cap but will subject such teams to reduced privileges in free agency. Teams that go above the luxury tax cap are subject to the luxury tax (a tax on every dollar spent over the luxury tax cap).

**Data collection:**

We are scraping the data from the website BASKETBALL Reference (<https://www.basketball-reference.com/contracts/players.html>)

We have data within 4 datasets to build our model.

**Problem:**

The major problem constitutes to predict the salaries of the players by building a model by using the data we have. We also have few problems in our datasets we need be to be cleaned thoroughly.

**Possible solution:**

We are planning to use regression model as our solution for obtaining the result as we think that a supervised machine learning algorithm would be apt for this situation.

We used the R and EXCEL as our major drivers for Data Mining process.

# Data Exploration and Visualization

Provide brief description of techniques used to explore the data including basic charts, distribution plots, correlations, missing values, rescaling, aggregation, hierarchies, zooming, filtering, etc.

**Correlation checks:**

The dependent variables we have considered include:

MPG, PPG, APG, RPG, TOPG, BPG, SPG, PER, AGE

A close up of a keyboard

Description automatically generated

A picture containing text

Description automatically generated

The order for correlation is:

PPG > MPG > TOPG > RPG > PER > SPG > APG

**Correlation values:**

A screenshot of a cell phone

Description automatically generated

We can see that the value of correlation decreases in the order specified.

It is really interesting to see that turnovers are having a good amount of positive correlation so we thought that maybe it is related to the player not giving the opposition a chance to have the ball for a longer time or the way it shows how aggressive he is and having more moments with the ball than others.

**Some variables of interest:**

**Salary** – Monetary figure made per individual in dataset.

**PPG** – Points per game of individuals in dataset.

**Age** – Age, in years, of individuals in dataset.

|  |
| --- |
|  |
|  |
|  |

**G** – Total games played in regular season. Maximum of 82

|  |
| --- |
|  |
|  |
|  |

**MPG**– Minutes played per game, per individual, in regular season.

**PER** – PER strives to measure a player's per-minute performance, while adjusting for pace. A league-average PER is always 15.00, which permits comparisons of player performance across seasons. Primarily an offensive measure of skills.

|  |  |
| --- | --- |
| All-time great season | 35.0+ |
| Runaway MVP candidate | 30.0-35.0 |
| Strong MVP candidate | 27.5-30.0 |
| Weak MVP candidate | 25.0-27.5 |
| Definite All-Star | 22.5-25.0 |
| Borderline All-Star | 20.0-22.5 |
| Second offensive option | 18.0-20.0 |
| Third offensive option | 16.5-18.0 |
| Slightly above-average player | 15.0-16.5 |
| Rotation player | 13.0-15.0 |
| Non-rotation player | 11.0-13.0 |
| Fringe roster player | 9.0-11.0 |
| Player who won't stick in the league | 0-9.0 |

Summaries of above-mentioned variables are:

A close up of text on a black surface

Description automatically generated

**Data visualization:**

We have created interactive visualizations for the players such that if you hover the pointer over the data you can know information about the player.

A picture containing wall

Description automatically generated

A close up of a map

Description automatically generated

# Data Preparation and Preprocessing

Provide information on data summary, dimension reduction, correlation analysis, PCA analysis, variable converting, variable selection, etc.

For deciding on the variables to proceed for we have considered the variables having good Pearson coefficient with the salary variable which is the output variable.

**Data cleaning:**

* We have filtered out only the data for the year 2017 for predicting the values for 2018.
* Firstly we have used feature engineering to manipulate our variables to have more definite sense of the data as we observed that we lacked the data regarding the stats for the variable which account for each game individually.
* We have data in two datasets which had to be merged to have a better look at the overview of the data so we have used the **merge** function in R which helped us to not only merge them but also do it on the basis of the player as the base characteristic.
* Then we have performed couple of correlation checks to monitor on which variables are more related with the outcome variable to eliminate the rest so that we have less noise and reduce multi-collinearity in the process of **data reduction**.
* We then performed few data reduction operation to ensure our data is free of **multicollinearity** through plotting scatter matrices for correlation.

# Data Mining Techniques and Implementation

You are expected to explore multiple data mining techniques as appropriate to your problem. Clearly state the problem in data mining context (e.g., classification, prediction, supervised/unsupervised learning, etc.). It is desirable to have a flowchart for the entire process from data cleaning/manipulation/variable selection and transformation to specific techniques/algorithms implemented in R.

The problem for our dataset is to **predict** salaries of various basketball players for the upcoming NBA season based on their performances.

We have used multiple data mining techniques for the process of model creation which include :

* Regression
* KNN
* Random forest
* Neural network

**Linear regression model:**

A picture containing photo, showing

Description automatically generated

We created coefficients for the model which represent the amount of money the particular player loses or gains for a game. Here we can see that a rebound per game can increase the player’s salary by $988,599.

A close up of text on a white surface

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A close up of a map

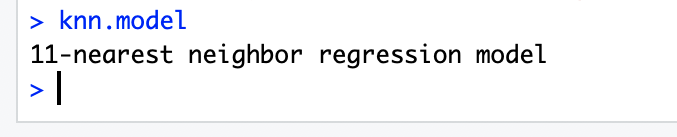
Description automatically generatedA close up of a map

Description automatically generatedA close up of a map

Description automatically generatedA close up of a map

Description automatically generated

**KNN**



**Random forest**

A screenshot of a cell phone

Description automatically generated

A screenshot of a social media post

Description automatically generated

**Neural network**

A close up of a logo

Description automatically generated

# Performance Evaluation

Present performance evaluation for all data mining techniques explored in your study, and select the best approach and explain why it is the best. Please ALWAYS divide your data into training set and validation set, and use validation set to evaluate the performance. If using pruned decision trees, please separate your data set into training, validation, and test. You should use performance measures such as lift chart, ROC curve, confusion matrix, prediction accuracy measures (MAE, Mean Error, MPE, MAPE, RMSE), classification accuracy measures (error rate, accuracy, sensitivity, specificity, false discovery rate, false omission rate, etc.).

For the model creation we have divided the datasets based on the proportion of 60:40 for training and validation. Then we implemented respective techniques for each model and predicted values based on that.

For evaluation processes we used accuracy and correlation function to check the model’s performance with respect to the output variable which is **salary.**

Then we used lift chart as a visual representation to describe the performance metric as a whole.

**Accuracy:**

A screenshot of a cell phone

Description automatically generated

**Correlation with Salary(output):**

A screenshot of a cell phone

Description automatically generated

Here we can see that the model of neural network is not at all suitable with our data so we can eliminate the Neural network as an option by itself.

**Lift charts:**

**Regression:**

A close up of a map

Description automatically generated

**Random Forest:**

A screenshot of a cell phone

Description automatically generated

**KNN:**

A close up of a map

Description automatically generated

**Lift decile for regression:**

A screenshot of a cell phone

Description automatically generated

**Observation:**

From the above information we can observe that the multi-variate regression and Random Forest are best suited for the datasets which we are working on but we preferred Random Forest as it is better in all aspects.

**Regression:**

A screenshot of a cell phone

Description automatically generated

Here we created a function to calculate the salary of the players based on the model.

We chose regression model which took into account the PPG, TOPG etc. and we predicted salaries of few players with the model to see the variance. We found out that with a few outliers our model is good for predicting the salaries of the players.

# Discussion and Recommendation:

Outliers caused the major trouble for us as there are few cases like David Booker who plays for Phoenix Suns is 20 years old and scoring at a high-rate so they retained him for $20 million as it is a bad team. As the Suns overpaid him it clearly constitutes for an outlier.

Managers are focused on the fact to create a perfect team with least amount spent. So they are very careful in the case of selecting players which may be through the draft picking, contract extensions, free agency acquisitions etc. to win the championship. As we have seen from the above outlier that there is a variance of $13million which indicates that always predicted values can’t be exact as the original due to outliers. These outliers are due to statistics of the players who are considered draft busts, who experience injury, aging, traumatic stress, etc.

Our model is based solely on stats and the intangibles are not included.

Overall the model we implemented is fit enough to predict the salary of players.

# Summary

We were interested in predicting the salaries of the NBA players as the salary cap of the NBA involves with lot of features. When we obtained the dataset from Kaggle it was raw we had to clean the data and do some feature engineering to make the best out of it. We then took the variables which were in high correlation to the output variable and performed the regression model to obtain the predicted salaries. To check the fit of our model we had used various techniques for error classification. Our model’s main predictors are PPG,MPG which have strongest relationship over all , the model predicts the salary based on these predictors. We have created a linear model in which the coefficients determine how the players salary changes which may be in the positive or negative way. Our analysis make the role of predictors simplified where we can see how much effect they show individually, for example: the coefficient for PPG is $840,485 for that would increase m amount of salary every time an individual scores a basket.

# R Code for use case study

**Glossary:**

‘Age’-age of player

‘MPG’-minutes per game. An NBA game has 48 minutes of regulation play

'RPG'-total rebounds per game

'APG'-assists per game

'SPG'-steals per game

'BPG'-blocks per game

'PPG'-points per game

'3PPG'-three pointers made per game

'3PAPG'-three pointers attempted per game

'2PPG'-two pointers made per game

'2PAPG'-two pointers attempted per game

'FTPG'-free throws made per game

'FTAPG'-free throws attempted per game

'PER'-player efficiency rating returns a per-minute rating of a player’s performances based on their accomplishments over the course of a game

**Code:**

library(data.table)

library(corrplot)

library(GGally)

library(tidyverse)

library(PerformanceAnalytics)

library(plotly)

library(gains)

#data cleaning

salary <- read.csv("/Users/nithin/Documents/neu/summer 2/case study/NBA\_season1718\_salary.csv")

stats <- read.csv("/Users/nithin/Documents/neu/summer 2/case study/Seasons\_stats.csv")

str(stats)

str(salary)

stats <-

stats %>% filter(Year >= 2017) %>%

select(Year:G, MP, PER, FG:PTS) %>%

distinct(Player, .keep\_all = TRUE) %>%

mutate(MPG = MP/G, PPG = PTS/G, APG = AST/G,

RPG = TRB/G, TOPG = TOV/G, BPG = BLK/G,

SPG = STL/G)

stats\_salary <- merge(stats, salary, by.x = "Player", by.y = "Player")

names(stats\_salary)[40] <- "salary"

stats\_salary <- stats\_salary[-39]

#correlation check

corrplot(cor(stats\_salary %>%

select(salary, MPG:SPG,

Age, PER, contains("%")),

use = "complete.obs"),

method = "number",type = "upper")

stats\_salary\_cor <-

stats\_salary %>%

select(salary, PPG, MPG, TOPG, RPG, PER, SPG, APG, BPG, Age, PF)

ggpairs(stats\_salary\_cor)

names(stats\_salary)[5] <- "Team"

#data visualization

library(ggplot2)

plot\_ly(data = stats\_salary, x = ~salary, y = ~PPG, color = ~Team,

hoverinfo = "text",

text = ~paste("Player: ", Player,

"<br>Salary: ", format(salary, big.mark = ","),"$",

"<br>PPG: ", round(PPG, digits = 3),

"<br>Team: ", Team)) %>%

layout(

title = "Salary vs Point Per Game",

xaxis = list(title = "Salary "),

yaxis = list(title = "Points Per Game")

)

plot\_ly(data = stats\_salary, x = ~salary, y = ~PER, color = ~Team,

hoverinfo = "text",

text = ~paste("Player: ", Player,

"<br>Salary: ", format(salary, big.mark = ","),"$",

"<br>PPG: ", round(PER, digits = 3),

"<br>Team: ", Team)) %>%

layout(

title = "Salary vs Player efficiency ratio",

xaxis = list(title = "Salary "),

yaxis = list(title = "Points Per Game")

)

stats\_salary %>%

ggplot(aes(x = salary, y = PPG)) +

geom\_point() +

geom\_smooth(method = "lm")

#stats\_salary %>%

# ggplot(aes(x = salary, y = PER)) +

# geom\_point() +

# geom\_smooth(method = "lm")

#models creation

# regression

stats\_salary\_model <-

stats\_salary %>% select(salary, MPG:SPG, Age, PF, PER)

#simple linear regression model

lm(salary~., data=stats\_salary\_model)

summary(stats\_salary\_model)

train.index<-sample(row.names(stats\_salary\_model), 0.6\*dim(stats\_salary\_model)[1])

valid.index <- setdiff(row.names(stats\_salary\_model), train.index)

train.df <- stats\_salary\_model[train.index,]

valid.df <- stats\_salary\_model[valid.index,]

regression\_model <- lm(formula = salary~ PPG + MPG + TOPG, data = stats\_salary\_model)

regression\_model.pred<- predict(regression\_model,valid.df[,2:11])

#knn

library(forecast)

library(caret)

#train model with training set

knn.model <- knnreg(train.df[,c(3,2,6)], train.df[,1], k = 11)

#make prediction for validation set

knn.model.pred <- predict(knn.model,valid.df[,c(3,2,6)])

#random forest

library(randomForest)

## random forest

rf.model <- randomForest(salary ~ PPG + MPG + TOPG, data = train.df, ntree = 500,

mtry = 4, nodesize = 5, importance = TRUE)

## variable importance plot

varImpPlot(rf.model, type = 1)

#neural network

library(neuralnet)

neural.model<-neuralnet(salary~PPG + MPG + TOPG,data=train.df,linear.output = F, hidden = c(12,9))

plot(neural.model,rep="best")

neural.model.pred<-compute(neural.model,valid.df[,c(2:11)])#for prediction

neural.model.pred$net.result[1:10]

#correlation and accuracy

#regression

#check accuracy

accuracy.reg<-accuracy(valid.df[,1], regression\_model.pred)

#check correlation

cor.reg<-cor(regression\_model.pred, valid.df[,1])

rss.reg <- sum((regression\_model.pred - valid.df[,1]) ^ 2) ## residual sum of squares

tss.reg <- sum((train.df[,1] - mean(train.df[,1])) ^ 2) ## total sum of squares

rsq.reg <- 1 - rss.reg/tss.reg

#knn

#check accuracy

accuracy.knn<-accuracy(valid.df[,1], knn.model.pred)

#check correlation

cor.knn<-cor(knn.model.pred, valid.df[,1])

rss.knn <- sum((knn.model.pred - valid.df[,1]) ^ 2) ## residual sum of squares

tss.knn <- sum((train.df[,1] - mean(train.df[,1])) ^ 2) ## total sum of squares

rsq.knn <- 1 - rss.knn/tss.knn

#random forest

## confusion matrix

rf.model.pred <- predict(rf.model, valid.df)

accuracy.rf<-accuracy(valid.df[,1], rf.model.pred)

#check correlation

cor.rf<-cor(rf.model.pred, valid.df[,1])

rss.rf <- sum((rf.model.pred - valid.df[,1]) ^ 2) ## residual sum of squares

tss.rf <- sum((train.df[,1] - mean(train.df[,1])) ^ 2) ## total sum of squares

rsq.rf <- 1 - rss.rf/tss.rf

#neural networks

cor.nn<-cor(valid.df$salary,neural.model.pred$net.result)

accuracy.nn<-accuracy(valid.df$salary,neural.model.pred$net.result)

#ERROR analysis

#lift charts

#regression

gain.reg <- gains(valid.df$salary[!is.na(regression\_model.pred)], regression\_model.pred[!is.na(regression\_model.pred)])

options(scipen=999)

salary.fut <- valid.df$salary[!is.na(valid.df$salary)]

par(pty="s")

plot(c(0,gain.reg$cume.pct.of.total\*sum(salary.fut)/1000000)~c(0,gain.reg$cume.obs),

xlab = "# cases", ylab = "Cumulative Expenses (Million)", main = "Lift Chart", type = "l", col = "blue")

#baseline

lines(c(0,sum(salary.fut)/1000000)~c(0,dim(valid.df)[1]), col = "gray", lty = 2)

#do it yourself

plot(cumsum(valid.df$salary[order(regression\_model.pred, decreasing=TRUE)]/1000000),

xlab = "# cases", ylab = "Cumulative Expenses (Million)", main = "Lift Chart", type = "l", col = "blue")

lines(c(0,sum(salary.fut)/1000000)~c(0,dim(valid.df)[1]), col = "gray", lty = 2)

#knn

gain.knn <- gains(valid.df$salary[!is.na(knn.model.pred)], knn.model.pred[!is.na(knn.model.pred)])

options(scipen=999)

salary.fut <- valid.df$salary[!is.na(valid.df$salary)]

par(pty="s")

plot(c(0,gain.knn$cume.pct.of.total\*sum(salary.fut)/1000000)~c(0,gain.knn$cume.obs),

xlab = "# cases", ylab = "Cumulative Expenses (Million)", main = "Lift Chart", type = "l", col = "blue")

#baseline

lines(c(0,sum(salary.fut)/1000000)~c(0,dim(valid.df)[1]), col = "gray", lty = 2)

#do it yourself

plot(cumsum(valid.df$salary[order(knn.model.pred, decreasing=TRUE)]/1000000),

xlab = "# cases", ylab = "Cumulative Expenses (Million)", main = "Lift Chart", type = "l", col = "blue")

lines(c(0,sum(salary.fut)/1000000)~c(0,dim(valid.df)[1]), col = "gray", lty = 2)

#random forest

gain.rf <- gains(valid.df$salary[!is.na(rf.model.pred)],rf.model.pred[!is.na(rf.model.pred)])

options(scipen=999)

salary.fut <- valid.df$salary[!is.na(valid.df$salary)]

par(pty="s")

plot(c(0,gain.rf$cume.pct.of.total\*sum(salary.fut)/1000000)~c(0,gain.rf$cume.obs),

xlab = "# cases", ylab = "Cumulative Expenses (Million)", main = "Lift Chart", type = "l", col = "blue")

#baseline

lines(c(0,sum(salary.fut)/1000000)~c(0,dim(valid.df)[1]), col = "gray", lty = 2)

#do it yourself

plot(cumsum(valid.df$salary[order(rf.model.pred, decreasing=TRUE)]/1000000),

xlab = "# cases", ylab = "Cumulative Expenses (Million)", main = "Lift Chart", type = "l", col = "blue")

lines(c(0,sum(salary.fut)/1000000)~c(0,dim(valid.df)[1]), col = "gray", lty = 2)

#So comparing"

#the correlations between the dependent and independent variables we find most correlation in Random Forest which is "0.8265"

#the accuracy: we get least RMSE in Random Forest "4779607"

#in lift charts a better model for random forest.

#So, we choose Random Forset as our Prediction Model.

#Now we create a fuction for calculating the salary for desired player

salary\_prediction <- function(m, point, minutes, turn\_over){

pre\_new <- predict(m, data.frame(PPG = point, MPG = minutes, TOPG = turn\_over))

msg <- paste("PPG:", point, ",MPG:", minutes, ",TOPG:", turn\_over, " ==> Expected Salary: $", format(round(pre\_new), big.mark = ","), sep = "")

print(msg)}

#Below i have calculated for the player James Harden

salary\_prediction(rf.model, 30.4, 35.4, 4.4)