NBA Shot Anaylsis

Jessica Gorr

December 5, 2024

Table of Contents

[Introduction 1](#_Toc184314325)

[EDA and Feature Engineering 2](#_Toc184314326)

[Prinicipal Component Analysis 5](#_Toc184314327)

[Clustering Data 7](#_Toc184314328)

[Anomaly Detection 10](#_Toc184314329)

# Introduction

The data is related player shot statistics from NBA.com which attempts to fully describe the kinds of shots players take and the way they end up making their points in games. The dataset includes every player who logged a minute of gameplay for every season from 2013-2014 through 208-2019. The exact columns are below with a few sample rows of data:

1 - year : nba season

2 - player : Player name (string)

3 - team : (categorical): team abbrevaited name

4 - Age (numeric): age in years

5 - GP (numeric): numer of games played

6 - W: (numeric) Number of wins

7 - L:(numeric) number of losses

8 - Min:(numeric) number of minutes played

9 - PCT\_FGA\_2PT: (numeric): Percentage of field goal attempts that were 2 pts

10 - PCT\_FGA\_3PT: (numberic)Percentage of field goal attempts that were 3 pts.

A database of players’ points and field goal attempts from each season between 2013 and 2019.

The goal in this project is to perform PCA and clustering on our data. We would also like to detect outliers via local outlier factor (LOF) by creating a binary variable from a categorical response above.

library(dplyr)  
library(reshape2)  
library(tidyr)  
library(ggplot2)  
library(tibble)  
  
#Load the sample data  
nba = read.csv("https://jessgorr01.github.io/STA551/proj4/nba\_shot\_types.csv")  
  
east\_teams <- c("BOS", "NYK", "MIA", "CHI", "TOR", "PHI", "MIL", "IND", "CLE", "DET", "ORL", "WAS", "ATL", "CHA", "BRO")  
west\_teams <- c("LAL", "GSW", "LAC", "PHX", "SAS", "DEN", "UTA", "POR", "OKC", "MIN", "DAL", "MEM", "NOP", "SAC")  
  
nba <- nba %>%  
 mutate(conference = case\_when(  
 TEAM %in% east\_teams ~ 0, # East Conference  
 TEAM %in% west\_teams ~ 1 # West Conference  
 ))

# EDA and Feature Engineering

In oder to perfom so EDA, we must have a basic understanding of the data. A summary of the data is printed below.

#Summarized descriptive statistics for all variables in the data set  
summary(nba)

## YEAR PLAYER TEAM AGE   
## Length:3006 Length:3006 Length:3006 Min. :19.00   
## Class :character Class :character Class :character 1st Qu.:24.00   
## Mode :character Mode :character Mode :character Median :26.00   
## Mean :26.77   
## 3rd Qu.:29.00   
## Max. :42.00   
##   
## GP W L MIN PCT\_FGA\_2PT   
## Min. : 1.0 Min. : 0 Min. : 0.00 Min. : 0.50 Min. : 0.0   
## 1st Qu.:31.0 1st Qu.:12 1st Qu.:14.00 1st Qu.:12.50 1st Qu.: 53.6   
## Median :60.0 Median :26 Median :26.00 Median :19.40 Median : 68.1   
## Mean :51.9 Mean :26 Mean :25.91 Mean :19.78 Mean : 68.8   
## 3rd Qu.:74.0 3rd Qu.:39 3rd Qu.:36.00 3rd Qu.:27.50 3rd Qu.: 89.0   
## Max. :83.0 Max. :73 Max. :71.00 Max. :42.40 Max. :100.0   
##   
## PCT\_FGA\_3PT PCT\_PTS\_2PT PCT\_PTS\_MR PCT\_PTS\_3PT   
## Min. : 0.00 Min. : 0.00 Min. : 0.0 Min. : 0.000   
## 1st Qu.: 10.60 1st Qu.: 42.60 1st Qu.: 4.9 1st Qu.: 3.325   
## Median : 31.60 Median : 55.40 Median : 11.2 Median : 26.400   
## Mean : 30.77 Mean : 56.38 Mean : 13.1 Mean : 26.579   
## 3rd Qu.: 46.20 3rd Qu.: 73.08 3rd Qu.: 18.8 3rd Qu.: 41.700   
## Max. :100.00 Max. :100.00 Max. :100.0 Max. :100.000   
##   
## PCT\_PTS\_FSTBRK PCT\_PTS\_FT PCT\_PTS\_OFF\_TOS PCT\_PTS\_INTHEPT   
## Min. : 0.00 Min. : 0.00 Min. : 0.00 Min. : 0.00   
## 1st Qu.: 6.40 1st Qu.: 10.70 1st Qu.: 12.50 1st Qu.: 28.30   
## Median : 11.30 Median : 14.90 Median : 15.30 Median : 40.70   
## Mean : 12.14 Mean : 15.65 Mean : 15.86 Mean : 43.28   
## 3rd Qu.: 16.40 3rd Qu.: 19.90 3rd Qu.: 18.50 3rd Qu.: 57.10   
## Max. :100.00 Max. :100.00 Max. :100.00 Max. :100.00   
##   
## PCT\_2PTGFM\_ASSTD PCT\_2PTGFM\_UNASSTD PCT\_3PTGFM\_ASSTD PCT\_3PTGFM\_UNASSTD  
## Min. : 0.00 Min. : 0.00 Min. : 0.00 Min. : 0.00   
## 1st Qu.: 34.80 1st Qu.: 29.40 1st Qu.: 55.60 1st Qu.: 0.00   
## Median : 55.80 Median : 42.35 Median : 87.50 Median : 3.60   
## Mean : 52.22 Mean : 45.35 Mean : 68.96 Mean : 11.08   
## 3rd Qu.: 69.20 3rd Qu.: 62.70 3rd Qu.: 98.10 3rd Qu.: 15.97   
## Max. :100.00 Max. :100.00 Max. :100.00 Max. :100.00   
##   
## PCT\_FGM\_ASSTD PCT\_FGM\_UNASSTD conference   
## Min. : 0.00 Min. : 0.00 Min. :0.0000   
## 1st Qu.: 49.10 1st Qu.: 23.20 1st Qu.:0.0000   
## Median : 65.30 Median : 33.80 Median :1.0000   
## Mean : 61.44 Mean : 36.89 Mean :0.5029   
## 3rd Qu.: 76.10 3rd Qu.: 50.00 3rd Qu.:1.0000   
## Max. :100.00 Max. :100.00 Max. :1.0000   
## NA's :206

We first want to figure out if every season worth of data is usable,because the stly of play has changed over the course of time. Maximizing the amount of data is a priority, but we also want to ensure we’re studying offensive seasons from the same population.

We’ll first look at some histograms of the statistics we have and subset them by season. From left to right, the histograms depict: % of Field Goal Attempts That are 3PT Shots, % of Points That Come From 2PT Shots, % of Points That Come From 3PT Shots, % of Points That Come From Free Throws, % of Points That Occur In The Paint, % of 2PT Field Goals Made That Were Assisted, % of 3PT Field Goals Made That Were Asisted, and % of Total Field Goals Made That were Assisted..

nba %>% select(1:3, 10:11, 13, 15, 17:18, 20, 22) %>%  
 melt(c('YEAR', 'PLAYER', 'TEAM')) %>%  
 ggplot(aes(x = value)) +  
 geom\_histogram(aes(fill = variable), binwidth = 5) +  
 facet\_wrap(YEAR ~ variable, nrow = length(unique(nba$YEAR))) +  
 labs(x = 'Percentage of Field Goals Attempted or Made', y = 'Number of Players (Frequency)') +  
 theme\_bw() +  
 guides(fill = FALSE)

 Here are density plots of the same statistics, once again subsetted by season:

nba %>% select(1:3, 10:11, 13, 15, 17:18, 20, 22) %>%  
 melt(c('YEAR', 'PLAYER', 'TEAM')) %>%  
 ggplot(aes(x = value)) +  
 geom\_density(aes(fill = variable)) +  
 facet\_wrap(YEAR ~ variable, nrow = length(unique(nba$YEAR))) +  
 labs(x = 'Percentage of FGM or FGA', y = 'Relative Likelihood (Density)') +  
 theme\_bw() +  
 guides(fill = FALSE)

 Some of the distributions remain fairly similar year to year, such as % of 2PT FG’s assisted or % of PTS in the paint. However, a few distributions, specifically the 3PT-related distributions, change somewhat noticeably, reflecting the shifting emphases of NBA offenses. THis can be seen in the 2013-2014 season.

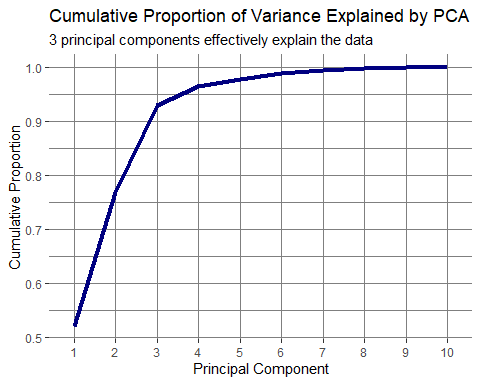
Another interesting distribution to take note of is the distribution of % of made 3PT shots that were assisted. In 2013-2014 data, there’s a gradual increasing slope from 50% of made 3PT field goals assisted to 100% of made 3PT field goals assisted, compared to season 2018-2019, this increase is much steeper. This will play a significant factor in one of the clusters we identify later on in the analysis.

After looking at these distributions, I reasoned that data from 2016-2017 illustrated similar offensive focuses; data from any earlier did not accurately reflect the way NBA offenses play now.

# Prinicipal Component Analysis

In PCA, we want to use the least amount of components possible to explain the most amount of variance. Using too few components might not accurately capture the dataset, but using too many components would overfit the data. Below is a visual of the amount of information captured by each added component:

set.seed(07072019)  
nba.train = nba %>%  
 filter(YEAR %in% c('2016-2017', '2017-2018', '2018-2019'), GP >= 20) %>%  
 select(contains('PCT')) %>%  
 select(2, 4:10, 12, 14)  
pca = summary(prcomp(nba.train))  
t(pca$importance) %>%  
 as.data.frame() %>%  
 tibble::rownames\_to\_column() %>%  
 ggplot() +  
 geom\_line(aes(x = factor(rowname, levels = paste('PC', seq(1, 10, 1), sep = ''), labels = seq(1, 10, 1)), y = `Cumulative Proportion`), group = 1, color = 'navyblue', size = 1.5) +  
 labs(title = 'Cumulative Proportion of Variance Explained by PCA',  
 subtitle = '3 principal components effectively explain the data',  
 x = 'Principal Component') +  
 theme(panel.background = element\_rect(fill = 'white'),  
 panel.grid.major = element\_line(color = 'grey50', size = .1),  
 panel.grid.minor = element\_line(color = 'grey50', size = .1))

 The table below shows the percentage of the data explained by each additional principal component:

pca

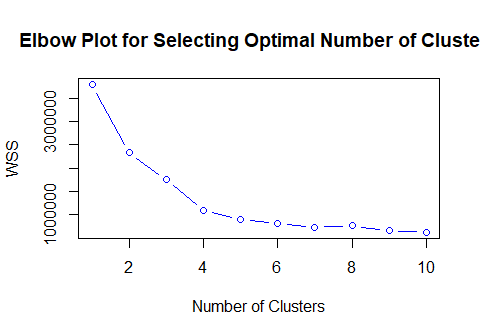
## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6 PC7  
## Standard deviation 39.748 27.4365 22.1141 10.3175 6.1567 6.09903 3.4807  
## Proportion of Variance 0.521 0.2482 0.1613 0.0351 0.0125 0.01227 0.0040  
## Cumulative Proportion 0.521 0.7693 0.9305 0.9656 0.9781 0.99041 0.9944  
## PC8 PC9 PC10  
## Standard deviation 3.27842 2.49386 0.02857  
## Proportion of Variance 0.00354 0.00205 0.00000  
## Cumulative Proportion 0.99795 1.00000 1.00000

Three directions explains 93% of the data, and an additional fourth direction explains an extra 3% of the data. Since the fourth component adds little to our understanding, we’ll use the first three components to transform our previously 10-dimensional dataset into a 3-dimensional dataset.

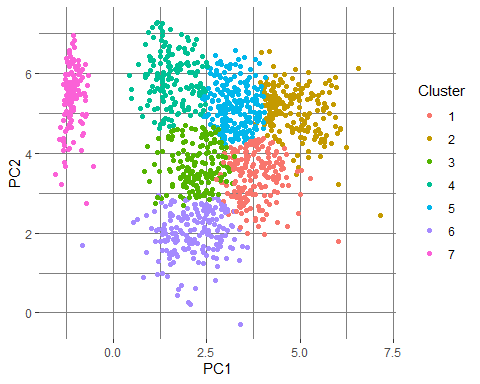
# Clustering Data

We want to figure out how many distinct groups of offensive players truly exist in the NBA. We would like to use this data to perform k-means cluster. First, we need to determine how many clusters are optimal. Intuitively, we know our data should have 10 clusters. To double check, we create an elbow plot which agrees with our initial intuition but the WSS score does not level off as sharply.

nba\_data <- prcomp(nba.train)$rotation[,1:3]  
  
  
wss = NULL  
K = 10  
  
for (i in 1:K){  
 wss[i] = kmeans(nba.train, i, 1 )$tot.withinss  
}  
  
## elbow plot  
plot(1:K, wss, type ="b",  
 col= "blue",  
 xlab="Number of Clusters",  
 ylab = "WSS",  
 main = "Elbow Plot for Selecting Optimal Number of Clusters")

 From the elbow graph, we can see that the optimal amount of clusters is 7. Below the clusters are plotted:

three\_directions <- prcomp(nba.train)$rotation[,1:3]  
sd\_vector <- sapply(nba.train, sd)  
scaled\_best\_directions <- three\_directions / sd\_vector  
PCA\_dimension\_pts <- as.matrix(nba.train) %\*% scaled\_best\_directions  
  
PCA\_kmeans = kmeans(PCA\_dimension\_pts, 7)  
PCA\_kmeans\_df <- cbind(nba %>% filter(YEAR %in% c('2016-2017', '2017-2018', '2018-2019'), GP >= 20), PCA\_dimension\_pts, cluster = PCA\_kmeans$cluster)  
ggplot(PCA\_kmeans\_df) +  
 geom\_jitter(aes(x = PC1, y = PC2, color = factor(cluster, levels = seq(1, 7, 1)))) +  
 guides(color = guide\_legend('Cluster')) +  
 theme(panel.background = element\_rect(fill = 'white'),  
 panel.grid.major = element\_line(color = 'grey50', size = .1),  
 panel.grid.minor = element\_line(color = 'grey50', size = .1),  
 legend.key = element\_rect(fill = 'white'))



As you can see, even without graphing our third dimension, our clustering looks pretty distinct. Below is a table of means of the original variables we inputted for each cluster of players we created:

PCA\_cluster\_means <- PCA\_kmeans\_df %>%   
 select\_if(grepl('PCT', colnames(PCA\_kmeans\_df))) %>%  
 cbind(cluster = PCA\_kmeans\_df$cluster, .) %>%  
 group\_by(cluster) %>%  
 summarise\_all(mean)  
PCA\_cluster\_means

## # A tibble: 7 × 16  
## cluster PCT\_FGA\_2PT PCT\_FGA\_3PT PCT\_PTS\_2PT PCT\_PTS\_MR PCT\_PTS\_3PT  
## <int> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1 50.7 49.3 40.2 12.4 46.3   
## 2 2 37.9 62.1 31.5 8.35 58.1   
## 3 3 73.8 26.2 61.1 13.4 21.8   
## 4 4 88.5 11.5 75.5 9.25 7.68   
## 5 5 61.6 38.4 53.0 10.0 32.9   
## 6 6 66.4 33.6 53.4 16.4 29.1   
## 7 7 99.2 0.751 82.1 6.62 0.0741  
## # ℹ 10 more variables: PCT\_PTS\_FSTBRK <dbl>, PCT\_PTS\_FT <dbl>,  
## # PCT\_PTS\_OFF\_TOS <dbl>, PCT\_PTS\_INTHEPT <dbl>, PCT\_2PTGFM\_ASSTD <dbl>,  
## # PCT\_2PTGFM\_UNASSTD <dbl>, PCT\_3PTGFM\_ASSTD <dbl>, PCT\_3PTGFM\_UNASSTD <dbl>,  
## # PCT\_FGM\_ASSTD <dbl>, PCT\_FGM\_UNASSTD <dbl>

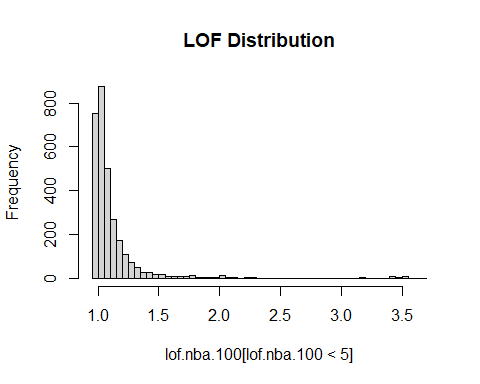
# Anomaly Detection

We begin by creating a binary variable from our response variable, in this case we want to decided if a player is in the eastern or western nba conference. If the TEAM column was labeled as either east team, we label it as 0. If the player is on a western team, we label it as 1. The code for creating this variable is below:

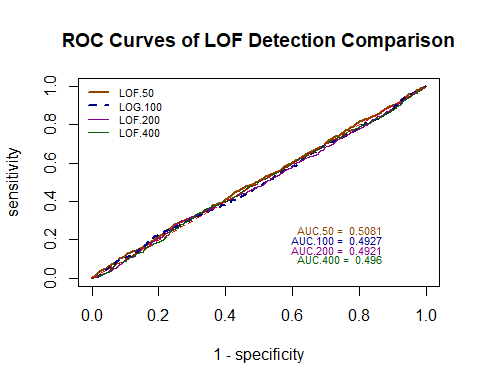
east\_teams <- c("BOS", "NYK", "MIA", "CHI", "TOR", "PHI", "MIL", "IND", "CLE", "DET", "ORL", "WAS", "ATL", "CHA", "BRO")  
west\_teams <- c("LAL", "GSW", "LAC", "PHX", "SAS", "DEN", "UTA", "POR", "OKC", "MIN", "DAL", "MEM", "NOP", "SAC")  
  
nba\_teams <- nba %>%  
 mutate(conference = case\_when(  
 TEAM %in% east\_teams ~ 0, # East Conference  
 TEAM %in% west\_teams ~ 1 # West Conference  
 ))  
  
clean\_nba <- nba\_teams %>%  
 select(-where(is.character))  
  
  
lof.nba.50 <- lof(clean\_nba [, c(4, 6, 7, 9,10)], minPts = 50)  
lof.nba.100 <- lof(clean\_nba [, c(4, 6, 7, 9,10)], minPts = 100)   
lof.nba.200 <- lof(clean\_nba [, c(4, 6, 7, 9,10)], minPts = 200)   
lof.nba.400 <- lof(clean\_nba [, c(4, 6, 7, 9,10)], minPts = 400)   
  
summary(lof.nba.100)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.9578 0.9998 1.0420 1.1158 1.1194 3.6549

hist(lof.nba.100[lof.nba.100 < 5], breaks = 50, main = "LOF Distribution")

 We can also sketch an ROC to assess the global performance in terms of catching rate below:

conference = as.character(clean\_nba$conference)  
  
 ##  
 ROCobj.lof.50 <- roc(conference, lof.nba.50, levels=c("1", "0"), direction = ">")  
 ROCobj.lof.100 <- roc(conference, lof.nba.100, levels=c("1", "0"), direction = ">")  
 ROCobj.lof.200 <- roc(conference, lof.nba.200, levels=c("1", "0"), direction = ">")  
 ROCobj.lof.400 <- roc(conference, lof.nba.400, levels=c("1", "0"), direction = ">")  
 ##  
 sen.LOF.50 = ROCobj.lof.50$sensitivities  
 fnr.LOF.50 = 1 - ROCobj.lof.50$specificities  
 ##  
 sen.LOF.100 = ROCobj.lof.100$sensitivities  
 fnr.LOF.100 = 1 - ROCobj.lof.100$specificities  
 ##  
 sen.LOF.200 = ROCobj.lof.200$sensitivities  
 fnr.LOF.200 = 1 - ROCobj.lof.200$specificities  
 ##  
 sen.LOF.400 = ROCobj.lof.400$sensitivities  
 fnr.LOF.400 = 1 - ROCobj.lof.400$specificities  
par(type="s")  
colors = c("#8B4500", "#00008B", "#8B008B", "#055d03")  
plot(fnr.LOF.50, sen.LOF.50, type = "l", lwd = 2, col = colors[1],  
 xlim = c(0,1),  
 ylim = c(0,1),  
 xlab = "1 - specificity",  
 ylab = "sensitivity",  
 main = "ROC Curves of LOF Detection Comparison")  
lines(fnr.LOF.100, sen.LOF.100, lwd = 2, lty = 2, col = colors[2])  
lines(fnr.LOF.200, sen.LOF.200, lwd = 1, col = colors[3])  
lines(fnr.LOF.400, sen.LOF.400, lwd = 1, col = colors[4])  
  
segments(0,0,1,1, lwd =1, col = "red", lty = 2)  
legend("topleft", c("LOF.50", "LOG.100", "LOF.200", "LOF.400"),   
 col=colors, lwd=c(2,2,1,1,1),  
 lty=c(1,2,1,1,2), bty = "n", cex = 0.7)  
  
##  
AUC.50 = ROCobj.lof.50$auc  
AUC.100 = ROCobj.lof.100$auc  
AUC.200 = ROCobj.lof.200$auc  
AUC.400 = ROCobj.lof.400$auc  
text(0.87, 0.25, paste("AUC.50 = ", round(AUC.50,4)), col=colors[1], cex = 0.7, adj = 1)  
text(0.87, 0.20, paste("AUC.100 = ", round(AUC.100,4)), col=colors[2], cex = 0.7, adj = 1)  
text(0.87, 0.15, paste("AUC.200 = ", round(AUC.200,4)), col=colors[3], cex = 0.7, adj = 1)  
text(0.87, 0.10, paste("AUC.400 = ", round(AUC.400,4)), col=colors[4], cex = 0.7, adj = 1)



From the output, we can see that the smaller values have a better area under the curve but with dimishing returns. Thus, in our case, we would choose as our value.