Mai Ngo

3/10/2023

FUNDAMENTALS OF DATA SCIENCE – DSC 441

HOMEWORK 5 – FINAL

**Single Problem (100 pts – see rubric)**

***a. Data gathering and integration.***

I’m using the NFL draft data set: 1985 – 2015.

Link: <https://www.kaggle.com/datasets/ulrikthygepedersen/nfl-draft-1985-2015>

Per our conversation, I have decided instead of working with 32 NFL teams. I will split the teams into two 2 conferences, AFC and NFC. There are 3 extra teams during (1985 - 2015) due to relocation and change of ownership.

* American Football Conference (AFC)
* National Football Conference (NFC)

I will do this step in the main CVS files. Main file: 8435 observations, 34 variables. Therefore, there will be 2 files: NFC\_data.csv: 4326 observations, 34 variables; and AFC\_data.csv.: 4109 observations, 34 variables.

National Football Conference (NFC):

* PHI - Philadelphia Eagles
* SFO - San Francisco 49ers
* GNB - Green Bay Packers
* MIN - Minnesota Vikings
* NYG - New York Giants
* TAM - Tampa Bay Buccaneers
* SEA - Seattle Seahawks
* DET - Detroit Lions
* NOR - New Orleans Saints
* RAM - Los Angeles Ram
* CHI - Chicago Bears
* WAS - Washington Commanders
* CAR - Carolina Panthers
* ATL - Atlanta Falcons
* ARI - Arizona Cardinals
* PHO – Phoenix Cardinals
* DAL - Dallas Cowboys
* STL – Saint Louis Rams

American Football Conference (AFC):

* KAN - Kansas City Chiefs
* CIN - Cincinnati Bengals
* BUF - Buffalo Bills
* NWE - New England Patriots
* NYJ - New York Jets
* MIA - Miami Dolphins
* BAL - Baltimore Ravens
* OAK - Oakland Raiders
* RAI - Las Vegas Raiders
* JAX - Jacksonville Jaguars
* DEN - Denver Broncos
* IND - Indianapolis Colts
* SDG - San Diego Chargers (now Los Angles Chargers)
* TEN - Tennessee Titans
* CLE - Cleveland Browns
* HOU - Houston Texans
* PIT - Pittsburgh Steelers

library(tidyverse)

AFCdata <- AFC\_data %>% mutate(Conference = 'AFC')

#Adding ‘conference’ column indicates type of conference.

#4109 observation, 35 variables.

NFCdata <- NFC\_data %>% mutate(Conference = 'NFC')

#Adding ‘conference’ column indicates type of conference.

#4326 observation, 35 variables.

NFLdata <- full\_join(AFCdata, NFCdata)

#Join 2 data frames.

#8435 observations, 35 variables.

summary(NFLdata)

Chart, scatter chart

Description automatically generated

* We can see that there are 15 variables have significant number of missing values Therefore, I will remove those features.

> nflData <- NFLdata %>% select(-c("column\_a", "player\_id", "cmp", "pass\_att" , "pass\_yds", "pass\_td", "pass\_int", "rush\_att", "rush\_yds", "rush\_tds",

"rec","rec\_yds", "rec\_tds", "tkl", "def\_int"))

* Then I notice all players during period of 1985-1993 didn't have University record. Therefore, I will remove observations (rows) from this year frame. Our data set now will be NFL draft during period 1994-2015.
* The reason I did this as well because of college sport division, where the football player went to college also has high impact on their draft opportunity.

nflData = nflData[nflData$year >= "1994" & nflData$year <= "2015", ]

#5538 observations, 20 variables.

Variables removal with reasoning:

* I notice variable ‘sk’ has a lot of missing values, almost half of the data set.
* Remove the ‘tm’ (team) column since we already have conference type.
* ‘hof’ variable Hall of Fame has all ‘no’ values.
* ‘to’ variable says how long they stay with their 1st team which won’t be needed.
* ‘position\_standard’ is the same as ‘position’ variable.
* ‘player’ variable is name of each player.
* ‘college\_univ’ variable is unique as well.
* 3 variables: ‘first4av’, ‘carav’ and ‘drav’ all represent a player’s approximate value.

Since we’re not doing deep-dive analysis. I will keep one variable ‘carav’

* Thus, remove these variables.

nflData <- nflData %>% select(-c("sk", "tm", "hof", "to", "position\_standard", "player", "college\_univ", "first4av", "drav"))

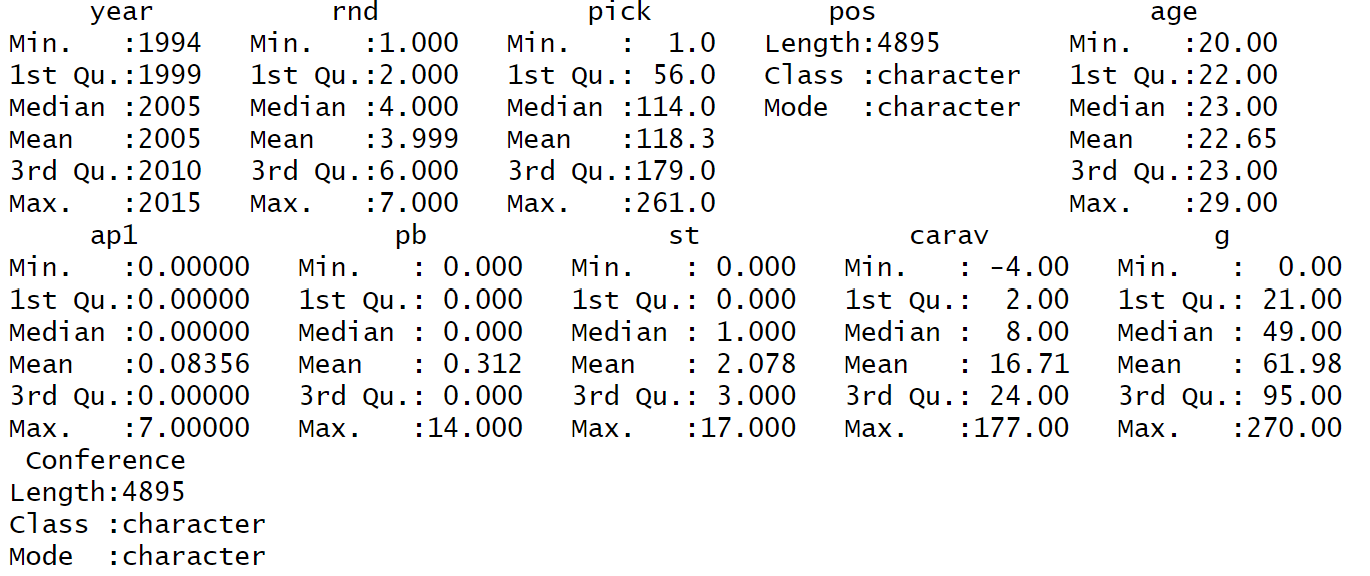
nflData <- na.omit(nflData)

#Remove missing values here and there.

**FINAL DATASET**

summary(nflData)

#4895 observation, 11 variables.



***b. Data Exploration***

Using data exploration to understand what is happening is important throughout the pipeline and is not limited to this step. However, it is important to use some exploration early on to make sure you understand your data. You must at least consider the distributions of each variable and at least some of the relationships between pairs of variables.

**Numerical variable: year, ‘rnd’, pick, age, ‘ap1’, ‘pb’, ‘st’, ‘carav’, ‘g’**

* ‘rnd’ - Round: 7 rounds of NFL draft.
* Pick: Picking order among all the players during the draft season.
* ‘ap1’ – All-Pro 1st: Number of times a player got 1st picked to any teams (top choice/All-Pro). Which means the best player at given position at that given season.
* ‘pb’ – Pro-bowler: Number of times the player was a Pro-bowler. Kind of like All-pro but the player is being chosen based more on popularity and audience preference,

rather than stat.

* ‘st’ – Starter: Number of seasons the player was his team's primary starter at his position, rather than bench players.
* ‘carav’ – Career Approximate Value: The seasonal value of a player at given position at that given year.
* ‘g’ – Games played: Number of games played.
* **Using Density plot to visualize the correlation between All-Pro 1st team and Pro-Bowler.**
* From the graph, we can see that player who got high vote for All-Pro 1st Team.

Also get high vote for Pro-Bowler, and vice versa.

stat\_value <- nflData %>% pivot\_longer(cols = c("ap1", "pb"),

names\_to="stat\_value", values\_to="value")

ggplot(stat\_value, aes(x=value, colour=stat\_value))

+ geom\_density()

+ ggtitle('NFL Draft: All-Pro 1st Team vs. Pro-Bowler.')

Graphical user interface, application

Description automatically generated

* **Using Scatterplot to visualize the correlation between Pick order and Career AV.**
* From the graph, we can see that higher Career Approximate Value leads to higher chance to be in the 1st draft. Which also means, players got drafted from later rounds have a smaller number of played games and smaller/ sparsely distributed

Career AV.

* The players are equally distributed among 7 rounds. Higher number of games

played corresponds to higher Career AV.

ggplot(nflData, aes(x = pick, y = carav, color = rnd))

+ geom\_point(aes(size = g), alpha = 0.4) + xlab("Pick")

+ ylab("Career Approximate Value")

+ ggtitle("NFL Draft: Pick vs. Career Approximate Value.")

+ labs(color = "rnd", size = "g")

+ scale\_color\_distiller(palette = "Paired")

Chart, scatter chart

Description automatically generated

* **Using Histogram to visualize ‘age’ variable.**
* Majority of players got drafted at 22-23 years old – graduating college.

ggplot(nflData, aes(age))

+ geom\_histogram(binwidth = 1, color="red")

+ ggtitle('NFL Draft: Distribution of Ages variable')

Chart, histogram

Description automatically generated

* **Using Histogram to visualize ‘st’ Starter variable.**
* Majority of players were primary starter at their position 1st year they got draft,

and then significantly decrease.

ggplot(nflData, aes(st))

+ geom\_histogram(binwidth = 2, color="pink")

+ ggtitle('NFL Draft: Distribution of Starter variable')

Chart, histogram

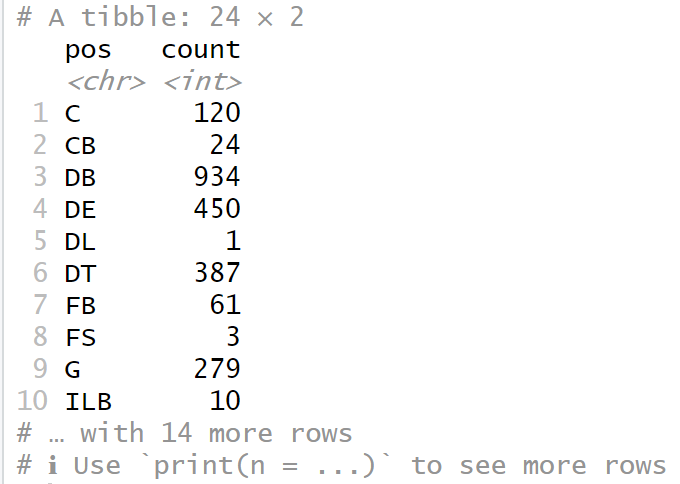
Description automatically generated

**Categorical variable: ‘pos’ position and conference type.**

* I want to see if each conference draft positions differently. Is there preference for certain positions?
* **Using Bar graph distribution for each conference with their draft positions.**
* Based on the graph, we can see that both conferences have almost the same

draft ratio.

nflData %>% group\_by(pos) %>% summarise("count" = n())



ggplot(nflData, aes(x=Conference, fill= pos))

+ geom\_bar(position="stack")

+ ggtitle('NFL Draft: Position by Conference')

A picture containing bar chart

Description automatically generated

***c. Data Cleaning***

* Since there were so many missing values from the original dataset. I have cleaned it at the beginning. Just double check.

sum(is.na(nflData))

[1] 0

* I used boxplot to check for outliers. There are couple variables have outliers. This is very common because sport performance usually has good indicator at the very first few years then gradually decrease.
* Overall, I decided to do data normalization on these variables:
* ‘age’: bin, and smooth by median.
* ‘ap1’, ‘pb’, ‘st’: Min-max normalization: [1,10]
* ‘carav’: z-score normalization

**‘age’ variable**

Chosen v numerical variable: ages 🡪 ages\_bins

Using equal width: N (bins count) = 2

B (max value of 'ages') = 29 | A (min value of 'ages') = 20

New value: ☞ Less\_23\_Age range: [20-23]

☞ Greater\_23\_Age range [23-29]

Reason: Majority of player get drafted fresh out of college between 22-23 years old. After 23 years-old, potentially they were agent-free, and should be accumulated into the same group due to the likelihood of getting drafted low. In fact, ‘age’ variable median is 23; mean is 22.64.

nflData\_1 <- nflData

nflData\_1 <- nflData\_1 %>% mutate(age\_bins = cut(age,

breaks=c(-Inf, 23, Inf), labels=c("<23 Age",">23 Age")))

Less\_23\_Age <- nflData\_1 %>% filter(age\_bins == "<23 Age") %>%

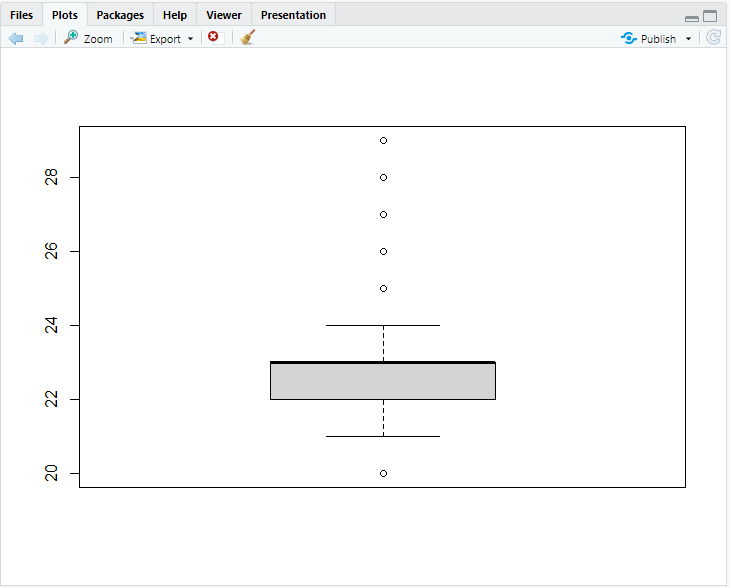
mutate(ages = median(age))

Greater\_23\_Age <- nflData\_1 %>% filter(age\_bins == ">23 Age") %>%

mutate(ages = median(age))

nflData\_1 <- bind\_rows(list(Less\_23\_Age,Greater\_23\_Age))

boxplot(nflData$age)



summary(nflData\_1$ages)

Min. 1st Qu. Median Mean 3rd Qu. Max.

23.00 23.00 23.00 23.12 23.00 24.00

nflData\_1 <- nflData\_1%>% select(-c("age", "age\_bins"))

#Remove ‘age\_bins’ and original ‘age’ values.

#Using nflData\_1 data frame moving forward.

**‘ap1’ variable**

norm\_minmax <- function(x,new\_max=1,new\_min=0){(((x-min(x))

\*(new\_max-new\_min))/(max(x)-min(x)))+new\_min}

normalise\_ap1 <- as.data.frame(lapply(nflData["ap1"], norm\_minmax))

summary(normalise\_ap1$ap1)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.00000 0.00000 0.00000 0.01194 0.00000 1.00000

nflData\_1$ap1 <- normalise\_ap1$ap1

boxplot(nflData$ap1)

**Table

Description automatically generated**

**‘pb’ variable**

normalise\_pb <- as.data.frame(lapply(nflData["pb"], norm\_minmax))

summary(normalise\_pb$pb)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.00000 0.00000 0.00000 0.02228 0.00000 1.00000

nflData\_1$pb <- normalise\_pb$pb

boxplot(nflData$pb)

Table

Description automatically generated with medium confidence

**‘st’ variable**

normalise\_st <- as.data.frame(lapply(nflData["st"], norm\_minmax))

summary(normalise\_st$st)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.00000 0.00000 0.05882 0.12226 0.17647 1.00000

nflData\_1$st <- normalise\_st$st

boxplot(nflData$st)

**Chart, box and whisker chart

Description automatically generated**

**‘carav’ variable**

norm\_zscore<-function(x){((x-mean(x))/sd(x))}

normalise\_carav <- as.data.frame(lapply(nflData["carav"], norm\_zscore))

summary(normalise\_carav$carav)

Min. 1st Qu. Median Mean 3rd Qu. Max.

-0.9610 -0.6827 -0.4043 0.0000 0.3380 7.4364

nflData\_1$carav <- normalise\_carav$carav

boxplot(nflData$carav)

**Chart, box and whisker chart

Description automatically generated**

head(nflData\_1)

year rnd pick pos ap1 pb st carav g Conference ages

1 2015 1 2 QB 0 0.00000000 0.11764706 -0.3990474 23 AFC 23

2 2015 1 3 OLB 0 0.00000000 0.00000000 -0.8099976 10 AFC 23

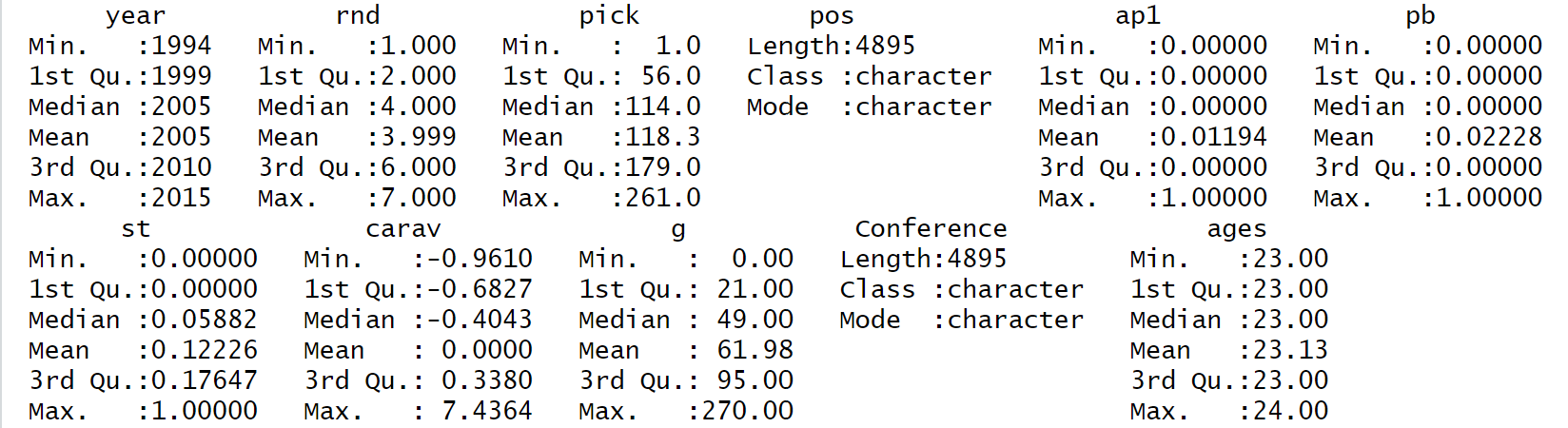
3 2015 1 4 WR 0 0.07142857 0.05882353 -0.3990474 26 AFC 23

4 2015 1 6 DE 0 0.00000000 0.11764706 -0.3990474 26 AFC 23

5 2015 1 12 NT 0 0.00000000 0.11764706 -0.5360308 27 AFC 23

6 2015 1 14 WR 0 0.00000000 0.00000000 -0.6730142 24 AFC 23

summary(nflData\_1)

****

***d. Data Preprocessing***

Making dummy variable for ‘pos’ variable.

nflData\_df <- nflData\_1

library(lattice)

library(caret)

library(ggplot2)

nflData\_df$Conference <- as.factor(nflData\_df$Conference)

dummy <- dummyVars(Conference ~ ., data = nflData\_df)

nflData\_dummies <- as.data.frame(predict(dummy, newdata = nflData\_df))

#4895 observations,33 variables.

summary(nflData\_dummies)

Diagram

Description automatically generated with low confidence

***e. Clustering***

library(stats)

library(factoextra)

library(ggplot2)

* I did both HAC and k-means methods.
* For HAC method, I used daisy() function and metric = “gower” (can be used with both categorical and numerical data), since I want to keep categorical variable ‘pos’. HAC method suggests k =2
* However, I decided to move forward with k-means since the dendrogram from HAC methods is too ‘clustered’ at the bottom.

A picture containing graphical user interface

Description automatically generated

**K-means method**

* Moving forward using nflData\_dummies data set.

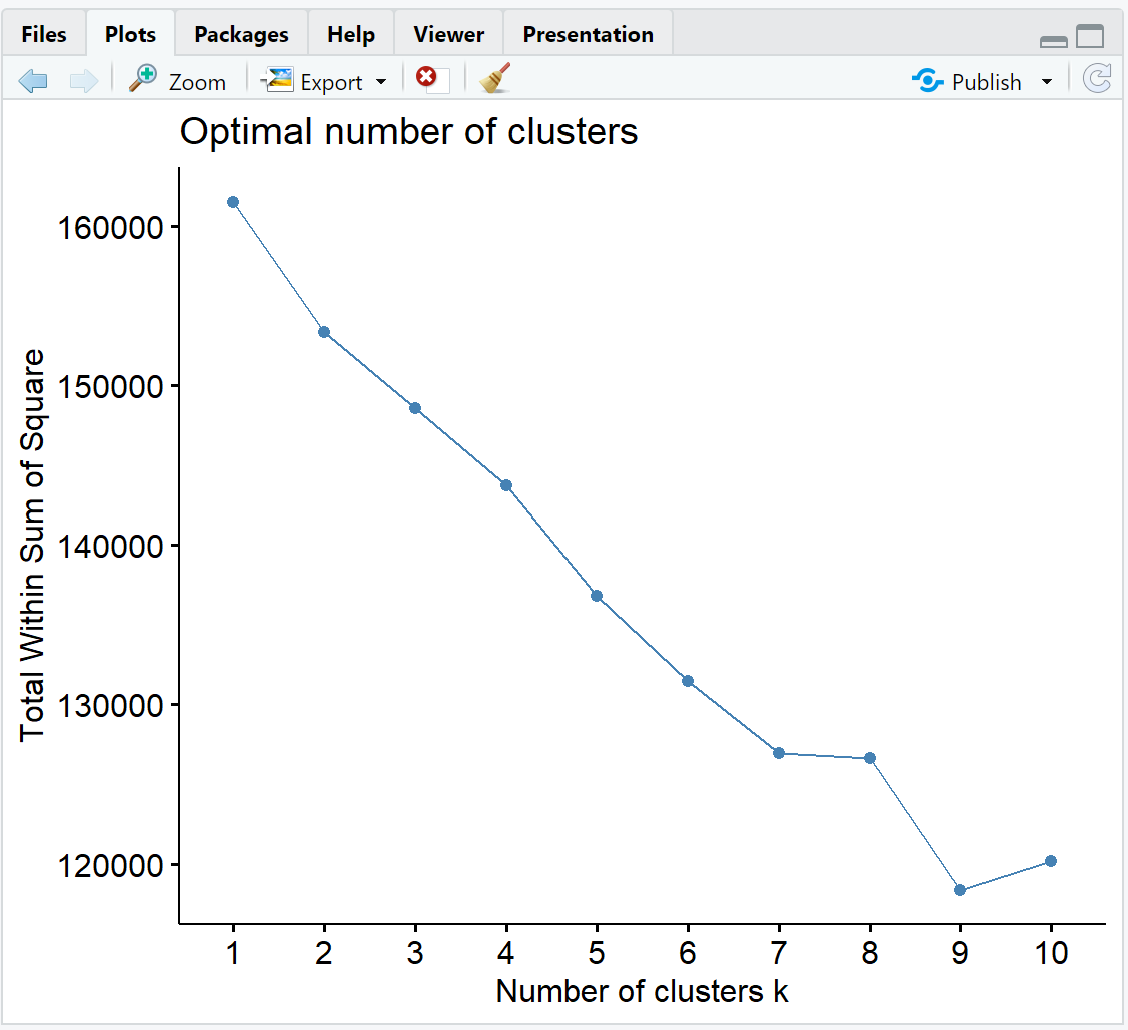
set.seed(1997)

preproc <- preProcess(nflData\_dummies, method=c("center", "scale"))

predictors <- predict(preproc, nflData\_dummies)

fviz\_nbclust(predictors, kmeans, method = "wss")

#Find k



fviz\_nbclust(predictors, kmeans, method = "silhouette")

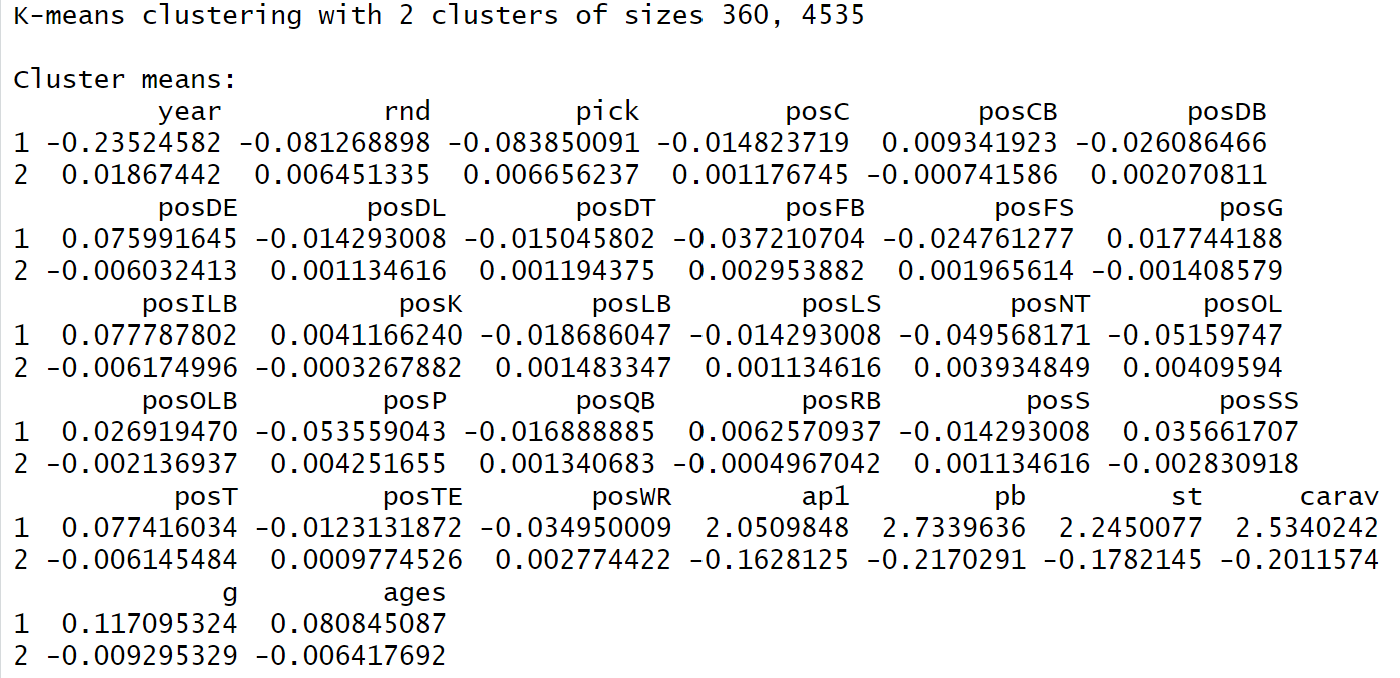
Chart, line chart

Description automatically generated

* Since ‘wss’ method suggests k=2 as well. Even though ‘silhouette’ method suggests k=9, the fact that we have 2 class labels and HAC method also gives k=2. I will move forward with k=2

fit <- kmeans(predictors, centers = 2, nstart = 25)

fit



fviz\_cluster(fit, data = predictors)

Scatter chart

Description automatically generated

***f. Classification***

* I moved forward with the nflData\_dummies dataset, but the accuracy result is very low (around 50%), and SVM was not generate result. My assumption is the dummy ‘pos’ variable creates this issue, so I removed it.
* Then I still receive 50% accuracy result, testing both normalized and original dataset (remove “pos” categorical variable). **Therefore, I will show the step using the original dataset.**
* I will use kNN (tune k) and SVM (tune C) for classification.
* First, I will run PCA.

**Run PCA**

nflData\_2 <- nflData %>% select(-c("pos"))

#4895 observations, 10 variables.

* Now the data set has all numerical variables, thus no need to convert to dummies.
* ‘pos’ – position variable was resulting 25 numerical dummy variables.

nflData\_2 <- nflData\_2 %>% select(-c("Conference"))

#Remove class label ‘Conference’

predictors <- nflData\_2

set.seed(1234)

preproc <- preProcess(predictors, method=c("center", "scale"))

predictors <- predict(preproc, predictors)

# Normalizing, scaling data. And fit ‘predictors’ data frame.

head(predictors)

# A tibble: 6 × 9

year rnd pick age ap1 pb st carav g

*<dbl>* *<dbl>* *<dbl>* *<dbl>* *<dbl>* *<dbl>* *<dbl>* *<dbl>* *<dbl>*

1 1.64 -1.50 -1.62 -1.88 -0.174 -0.273 -0.0263 -0.358 -0.767

2 1.64 -1.50 -1.61 -1.88 -0.174 -0.273 -0.698 -0.775 -1.02

3 1.64 -1.50 -1.59 -1.88 -0.174 0.602 -0.362 -0.358 -0.708

4 1.64 -1.50 -1.57 -1.88 -0.174 -0.273 -0.0263 -0.358 -0.708

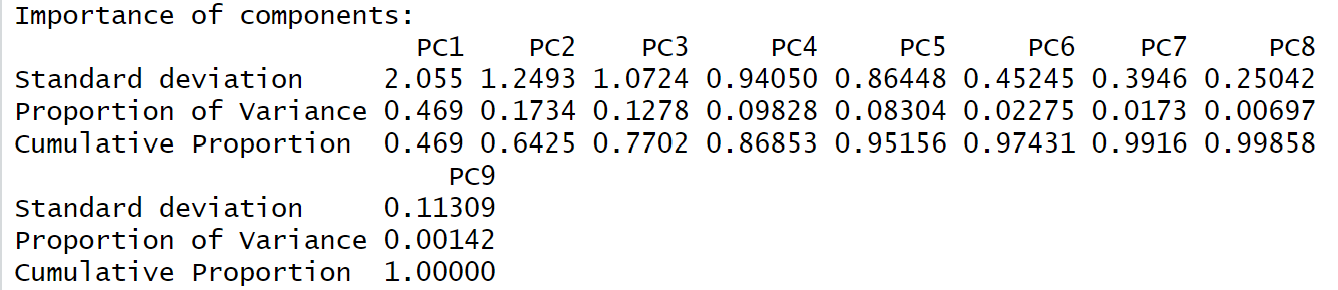
5 1.64 -1.50 -1.48 -0.743 -0.174 -0.273 -0.0263 -0.497 -0.689

6 1.64 -1.50 -1.45 -0.743 -0.174 -0.273 -0.698 -0.636 -0.748

pca = prcomp(predictors)

summary(pca)

* From the results shown below, the variance is captured by almost all principal components at +99% variance (8 PCs). Thus, I will use the original dataset itself for classification.



nfl.pca = as.data.frame(pca$x)

nfl.pca$Conference <- nflData$Conference

ggplot(data = nfl.pca, aes(x = PC1, y = PC2, color = Conference))

+ geom\_point(alpha= 0.4)

+ ggtitle("NFL Draft Conference classification using PCA")

+ scale\_color\_manual(values=c('red','black'))

**Chart, scatter chart

Description automatically generated**

**SVM (tune C) classification method**

grid <- expand.grid(C = seq(1,2,0.1))

#Set grid search

ctrl <- trainControl(method="cv", number = 10)

nflData\_2$Conference <- nflData$Conference

#Put back ‘Conference’ class label

svm\_grid <- train(Conference ~., data = nflData\_2, method = "svmLinear",

trControl = train\_control, tuneGrid = grid)

svm\_grid

Text

Description automatically generated

* C = 1.9 give the best accuracy result of approximately 50%

**kNN (tune k) classification method**

set.seed(9876)

ctrl <- trainControl(method="cv", number = 10)

knnFit <- train(Conference ~ ., data = nflData\_2, method = "knn",

trControl = ctrl, preProcess = c("center","scale"), tuneLength = 15)

knnFit

Text, table

Description automatically generated

* k = 27 gives the best accuracy result approximately 50%

**Use PCA again to visualize the labels for kNN and SVM.**

**kNN**

fviz\_nbclust(predictors, kmeans, method = "silhouette")

**Graphical user interface, chart, line chart

Description automatically generated**

fit <- kmeans(predictors, centers = 2, nstart = 25)

nfl.pca$Conference = as.factor(fit$cluster)

ggplot(data = nfl.pca, aes(x = PC1, y = PC2, color = Conference))

+ geom\_point(alpha= 0.4)

+ ggtitle("NFL Draft Conference KNN classification using PCA")

+ scale\_color\_manual(values=c('red','black'))

**Chart, scatter chart

Description automatically generated**

**SVM**

svm\_predictors <- predict(svm\_grid, predictors)

svm\_predictor <- as.data.frame(svm\_predictors)

nfl.pca$Conference <- svm\_predictor$svm\_predictors

ggplot(data = nfl.pca, aes(x = PC1, y = PC2, color = Conference))

+ geom\_point(alpha= 0.4)

+ ggtitle("NFL Draft Conference SVM classification using PCA")

+ scale\_color\_manual(values=c('red','black'))

**Chart, scatter chart

Description automatically generated**

***g. Evaluation***

library(tibble)

library(bitops)

library(rattle)

library(pROC)

**Confusion matrix (60/40)**

set.seed(4000)

nflData$Conference <- as.factor(nflData$Conference)

nflData$pos <- as.factor(nflData$pos)

index = createDataPartition(y=nflData$Conference, p=0.6, list=FALSE)

train\_nfl = nflData[index,]

test\_nfl = nflData[-index,]

train\_control = trainControl(method = "cv", number = 10)

tree <- train(Conference ~., data = train\_nfl, method = "rpart",

trControl = train\_control)

pred\_nfl <- predict(tree, test\_nfl)

cm <- confusionMatrix(test\_nfl$Conference, pred\_nfl)

cm

**A screenshot of a computer

Description automatically generated with low confidence**

**Precision and Recall**

metrics <- as.data.frame(cm$byClass)

metrics

**Text, letter

Description automatically generated**

metrics[c("Precision"),]

[1] 0.8232891

> metrics[c("Recall"),]

[1] 0.495695

metrics[c("Specificity"),]

[1] 0.4773414

> metrics[c("F1"),]

[1] 0.61881

> metrics[c("Balanced Accuracy"),]

[1] 0.4865182

**ROC plot**

library(mlbench)

train\_control = trainControl(method = "cv", number = 10)

dtree <- train(Conference ~., data = train\_nfl, method = "rpart", trControl = train\_control)

dtree

Text, letter

Description automatically generated

pred\_nfl2 <- predict(dtree, test\_nfl)

confusionMatrix(test\_nfl$Conference, pred\_nfl2)

A screenshot of a computer

Description automatically generated with medium confidence

pred\_prob <- predict(dtree, test\_nfl, type = "prob")

head(pred\_prob)

AFC NFC

1 0.4715026 0.5284974

2 0.5161290 0.4838710

3 0.5161290 0.4838710

4 0.5161290 0.4838710

5 0.5161290 0.4838710

6 0.4715026 0.5284974

roc\_obj <- roc((test\_nfl$Conference), pred\_prob[,1])

plot(roc\_obj, print.auc=TRUE)

Chart

Description automatically generated

**Explain how these performance measures makes your classifier look compared to accuracy.**

* After doing classification, it is clearly show that my model is not doing great with 'classification'. The ROC curve has AUC value exactly at 0.5 which indicates I could have made errors in building training algorithm.
* However, looking overall, my objective in analyzing this dataset is to find if there is a specific pattern, indicator, or performance stat; that can classify a football player drafted for a particular NFL Conference. This doesn't seem so.

***h. Report***

For part a. I merge two datasets; each dataset represents NFL teams for each Conference: National Football Conference (NFC) and American Football Conference (AFC). Then I did some research on football statistics to understand more about the data I am working on. Then I remove some data due to missing values and use it as final dataset. Move on to looking at data distribution, part b. I have plotted several graphs and decided to separate by categorical and numerical values. Then I transform 'age' value by binning (smooth by median), number of games played by z-scores, and 3 stats that represent votes by min-max normalization.

Move on clustering and classification, I did a lot of trials and errors and decided to show the best result. Overall, I got 50% accuracy result. Initially, I thought it was causing by the ‘pos’ position variable, this categorical variable produces 25 dummy variables. By this time, I start to think that maybe it should be 50% since my class label are the NFL conferences (AFC and NFC). The possibility of a player goes to either conference should be 50-50. These players pretty much have good/similar performance stats and both conferences recruit the same rate of players/positions per round.

Overall, I have learned most from doing this assignment. By using my own dataset, I have learned to read and understand these concepts thoroughly. Maybe I should have classified manually player's position using Approximate Value (AV). For example, quarter back position has higher AV than tight-ends position. By doing this, I could reduce number of dummy variables potentially. Another take away I learned from analyzing this dataset and doing this homework too, is that I cannot expect to have good result like the 'homework'. Raw data has so many characteristics which requires trials and errors, knowledge, experience, and intuition to reduce the expected result to make better decision.

Finally, I have learned how to use R properly. Maybe this homework result is not what I expected, but I completely understand the code and the algorithm. I can confidently explain every step and line of code.

***i. Reflection***

Data science in general is the studies of data by using a variety of methods to make better decision. There are two main branches; first, utilizing and analyzing data to make and predict better decisions (business management related purposes). Second branch is more on machine learning (algorithm improvement for data processing). Even though data can be spoken/analyzed by algorithm, it is still necessary to have human intuition involved to get the data 'makes sense'. Throughout the course, I have learned about the data mining process, what needed to be done when encountering a large dataset; by looking at variable's distribution, correlation, and missing values. I also learned about 3 data classification methods: Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Decision Tree. All are supervised learning machine algorithms. In addition, I also learned about clustering (unsupervised learning machine algorithm) with 2 methods: HAC and K-means. I also learned how to use R, and frankly say, it is very difficult. There are many ways to make inputs for the same purpose (with different packages and tools). By this time, I have known some basic R libraries (tidyverse, caret, dyplr, etc.). To verify the accuracy of a model, I can use test and train datasets in many different combinations. And specifically, from this Homework 5, my model was not resulting high accuracy value on neither classification methods. I realized that the datasets that I have been learning/working on throughout the class makes 'data science' seems very achievable, but it does not seem so. Lots of practicing, trials and errors to gain more knowledge and experience.