Aim 2

Ryan Canfield

2023-11-19

### This is the code for the preliminary results for completing Aim 2.

### Other code will be added later which will include the final model with the highest accuracy.

## By Ryan Canfield

## DSE4900 Data Scienc Capstone

## Dr. Santa Barabara Group (Ryan Canfield, Pat Norcross, and Alex Buterra)

## 11/14/23

# Libraries and Packages  
library(factoextra) # Used for PCA

library(FactoMineR)

# Reading in the data set and previewing the data  
df <- read.csv("combined\_dataset.csv")

## Preprocessing   
# Dropping columns  
df2 = subset(df, select = -c(X, index, GROUP, ELIGIBILITYRESCREENQ1, ELIGIBILITYRESCREENQ2, ELIGIBILITYRESCREENQ3, ELIGIBILITYRESCREENQ4, GENDER2, RACEOTHER, LIVING, NUMBEROFROOMMATES, INCOME, EMPLOYMENT, HEALTHHIX23, DRUGADDICTION, ALCOHOLADDICTION, SU15))  
  
## THIS WAS DONE BUT AFTER DOING WORK THIS IS NOT NECCSARY BECAUSE THESE COLUMNS ARE NOT USED.  
# Binary column race into 1 for white all other for 0   
# living 0 alone 1 other people  
df2$RACE <- ifelse(df2$RACE != 5, 0, 1)  
df2$EDUCATION <- ifelse(df2$EDUCATION <= 5, 0, 1)  
  
# Fixes the Na's found in the age column  
df2$AGE2[is.na(df2$AGE2)] <- round(median(df2$AGE2, na.rm = TRUE))  
# Keep HISPANICLATINO column(if they are part or not )

# Preliminary model to see if we can use total scores to predict intensity   
Starting.Intensity <- lm(RTINTENSITY ~ MDISCORE + RSSCORE + BDSSCORE + SPASSCORE + CESDSCORE + STAISCORE + AUDITSCORE, data = df2)  
summary.lm(Starting.Intensity)

# Plotting the forward stepwise linear regression model based off of the most significant variable  
g4 <- ggplot(data = df2, aes(x = STAISCORE, y = RTINTENSITY)) +   
 geom\_point () +   
 labs(title = "Comparison of Total Anxiety -v- Workout Intesnsity",   
 subtitle ="For 35 paitents with Muscle Dismorphia",   
 x = "Total Anxiety Score",   
 y = "Intensity",   
 caption = "[source: data from our client Dr. SantaBarbara]") +   
 geom\_smooth(method = "lm", se = 0, colour = "#28B463")   
g4

## PCA for MDI  
# Just getting the MDI variables MDI1 - MDIScore  
subsetMDI <- df2[c(22:38)]  
  
set.seed(310) #I use this so that I get the same answer each time  
  
# Preforming PCA  
res.pca.MDI <- prcomp(subsetMDI[c(1:16)], scale = TRUE) # leave off the last variable, which is the response variable "quality"  
  
# Eigenvalues  
eig.val.MDI <- get\_eigenvalue(res.pca.MDI)  
eig.val.MDI

# Scree plot to look at which dimensions to keep  
fviz\_eig(res.pca.MDI)

# Looking at those dimensions to extract variables with high variance  
res.var.MDI <- get\_pca\_ind(res.pca.MDI)  
  
stored <- res.var.MDI$coord[, 1:3]  
# ^^^ Store this into a data frame  
  
MDIvari <- data.frame(stored)  
MDIvari

# MDI6 MDI8 MDI9 MDI10  
# MDI2 MDI11 MDI13  
# MDI4

# This lets you look at what variables are used or questions ask  
which.var.MDI <- get\_pca\_var(res.pca.MDI)  
which.var.MDI$coord

### This is the same code as abov just replaced with different survey data  
## Not the different subsets  
  
## PCA for RS  
subsetRS <- df2[c(39:49)]  
  
set.seed(310)   
res.pca.RS <- prcomp(subsetRS[c(1:10)], scale = TRUE) # leave off the last variable, which is the response variable "quality"  
eig.val.RS <- get\_eigenvalue(res.pca.RS)  
eig.val.RS

fviz\_eig(res.pca.RS)

res.var.RS <- get\_pca\_ind(res.pca.RS)  
RSvari <- data.frame(res.var.RS$coord[, 1:4])  
  
# RS7 RS8   
# RS2 RS4  
# RS1

# This lets you look at what variables are used or questions ask  
which.var.RS <- get\_pca\_var(res.pca.RS)  
which.var.RS$coord

## PCA for BDS  
subsetBDS <- df2[c(50:59)]  
set.seed(310)   
res.pca.BDS <- prcomp(subsetBDS[c(1:9)], scale = TRUE) # leave off the last variable, which is the response variable "quality"  
eig.val.BDS <- get\_eigenvalue(res.pca.BDS)  
eig.val.BDS

fviz\_eig(res.pca.BDS)

res.var.BDS <- get\_pca\_ind(res.pca.BDS)  
BDSvari <- data.frame(res.var.BDS$coord[, 1:4])  
  
# BDS4 BDS5 BDS7 BDS3  
# Results for Variables - Prints component matrix

# This lets you look at what variables are used or questions ask  
which.var.BDS <- get\_pca\_var(res.pca.BDS)  
which.var.BDS$coord

# PCA for SPAS  
subsetSPAS <- df2[c(60:72)]  
set.seed(310)   
res.pca.SPAS <- prcomp(subsetSPAS[c(1:12)], scale = TRUE) # leave off the last variable, which is the response variable "quality"  
eig.val.SPAS <- get\_eigenvalue(res.pca.SPAS)  
eig.val.SPAS

fviz\_eig(res.pca.SPAS)

res.var.SPAS <- get\_pca\_ind(res.pca.SPAS)  
SPASvari <- data.frame(res.var.SPAS$coord[, 1:4])  
  
# SPAS1 SPAS2 SPAS4 SPAS6 SPAS8

# This lets you look at what variables are used or questions ask  
which.var.SPAS <- get\_pca\_var(res.pca.SPAS)  
which.var.SPAS$coord

# PCA for CESD  
subsetCESD <- df2[c(73:83)]  
set.seed(310)   
res.pca.CESD <- prcomp(subsetCESD[c(1:10)], scale = TRUE) # leave off the last variable, which is the response variable "quality"  
eig.val.CESD <- get\_eigenvalue(res.pca.CESD)  
eig.val.CESD

fviz\_eig(res.pca.CESD)

res.var.CESD <- get\_pca\_ind(res.pca.CESD)  
CESDvari <- data.frame(res.var.CESD$coord[, 1:3])  
  
# CESD3 CESD9 CESD6

# This lets you look at what variables are used or questions ask  
which.var.CESD <- get\_pca\_var(res.pca.CESD)  
which.var.CESD$coord

# PCA for STAI  
subsetSTAI <- df2[c(84:104)]  
set.seed(310)   
res.pca.STAI <- prcomp(subsetSTAI[c(1:20)], scale = TRUE) # leave off the last variable, which is the response variable "quality"  
eig.val.STAI <- get\_eigenvalue(res.pca.STAI)  
eig.val.STAI

fviz\_eig(res.pca.STAI)

res.var.STAI <- get\_pca\_ind(res.pca.STAI)  
STAIvari <- data.frame(res.var.STAI$coord[, 1:5])  
  
# STAI1 STAI4 STAI9 STAI10 STAI12 STAI16 STAI19

# This lets you look at what variables are used or questions ask  
which.var.STAI <- get\_pca\_var(res.pca.STAI)  
which.var.STAI$coord

set.seed(310)   
# PCA for AUDIT  
subsetAUDIT <- df2[c(115:118)]  
subsetAUDIT <- replace(subsetAUDIT, is.na(subsetAUDIT), 0)  
  
res.pca.AUDIT <- prcomp(subsetAUDIT[c(1:3)], scale = TRUE) # leave off the last variable, which is the response variable "quality"  
eig.val.AUDIT <- get\_eigenvalue(res.pca.AUDIT)  
eig.val.AUDIT

fviz\_eig(res.pca.AUDIT)

res.var.AUDIT <- get\_pca\_ind(res.pca.AUDIT)  
  
AUDITvari <- data.frame(res.var.AUDIT$coord[, 1:1])  
  
# AUDIT1 AUDIT2 AUDIT3

# This lets you look at what variables are used or questions ask  
which.var.AUDIT <- get\_pca\_var(res.pca.AUDIT)  
which.var.AUDIT$coord

library(dplyr)

# Renaming the columns to tell the dimensions apart.  
MDIvari <- MDIvari %>% rename("MDI.Dim.1" = "Dim.1", "MDI.Dim.2" = "Dim.2", "MDI.Dim.3" = "Dim.3")  
RSvari <- RSvari %>% rename("RS.Dim.1" = "Dim.1", "RS.Dim.2" = "Dim.2", "RS.Dim.3" = "Dim.3", "RS.Dim.4" = "Dim.4")  
BDSvari <- BDSvari %>% rename("BDS.Dim.1" = "Dim.1", "BDS.Dim.2" = "Dim.2", "BDS.Dim.3" = "Dim.3", "BDS.Dim.4" = "Dim.4")  
SPASvari <- SPASvari %>% rename("SPAS.Dim.1" = "Dim.1", "SPAS.Dim.2" = "Dim.2", "SPAS.Dim.3" = "Dim.3", "SPAS.Dim.4" = "Dim.4",)  
CESDvari <- CESDvari %>% rename("CESD.Dim.1" = "Dim.1", "CESD.Dim.2" = "Dim.2", "CESD.Dim.3" = "Dim.3")  
STAIvari <- STAIvari %>% rename("STAI.Dim.1" = "Dim.1", "STAI.Dim.2" = "Dim.2", "STAI.Dim.3" = "Dim.3", "STAI.Dim.4" = "Dim.4", "STAI.Dim.5" = "Dim.5")  
AUDITvari <- AUDITvari %>% rename("AUDIT.Dim.1" = "res.var.AUDIT.coord...1.1.")

# Forming a new dataset with the new PCA dimensions and the workout variables.   
questionaire.Variables <- cbind(MDIvari, RSvari, BDSvari, SPASvari, CESDvari, STAIvari, AUDITvari)  
df2.1 <- df2[ , 19:21]  
  
# Here are the scores if they are needed.  
#questionaire.Scores <- cbind(subsetMDI[c(17)], subsetRS[c(11)], subsetBDS[c(10)], subsetSPAS[c(13)], subsetCESD[c(11)], subsetSTAI[c(21)], subsetAUDIT[c(4)])  
#questionaire.Scores  
  
# This is the dataset I am using for the analysis.  
df3 <- cbind(df2.1, questionaire.Variables)  
head(df3)

## Linear Regression

# Preforming linear regression on all the surveys to see if any dimensions are significant.  
MDI.Intensity <- lm(RTINTENSITY ~ MDI.Dim.1 + MDI.Dim.2 + MDI.Dim.3, data = df3)  
summary.lm(MDI.Intensity)

RS.Intensity <- lm(RTINTENSITY ~ RS.Dim.1 + RS.Dim.2 + RS.Dim.3 + RS.Dim.4, data = df3)  
summary.lm(RS.Intensity)

BDS.Intensity <- lm(RTINTENSITY ~ BDS.Dim.1 + BDS.Dim.2 + BDS.Dim.3 + BDS.Dim.4, data = df3)  
summary.lm(BDS.Intensity)

SPAS.Intensity <- lm(RTINTENSITY ~ SPAS.Dim.1 + SPAS.Dim.2 + SPAS.Dim.3 + SPAS.Dim.4, data = df3)  
summary.lm(SPAS.Intensity)

CESD.Intensity <- lm(RTINTENSITY ~ CESD.Dim.1 + CESD.Dim.2 + CESD.Dim.3, data = df3)  
summary.lm(CESD.Intensity)

STAI.Intensity <- lm(RTINTENSITY ~ STAI.Dim.1 + STAI.Dim.2 + STAI.Dim.3 + STAI.Dim.4 + STAI.Dim.5, data = df3)  
summary.lm(STAI.Intensity)

AUDIT.Intensity <- lm(RTINTENSITY ~ AUDIT.Dim.1, data = df3)  
summary.lm(AUDIT.Intensity)

# Significant variables  
# MDI 2  
# RS 2 4  
# STAI 4

# Final linear regression model taken from the significant variables above  
FINAL.Intensity <- lm(RTINTENSITY ~ MDI.Dim.2 + RS.Dim.4 + STAI.Dim.4, data = df3)  
summary.lm(FINAL.Intensity)

## Do scores first an see if anything is significant   
## Then go into PCA  
## TALK ABOUT WHAT QUESTIONS GO INTO THE THREE  
  
g1 <- ggplot(data = df3, aes(x = STAI.Dim.4, y = RTINTENSITY)) + # create cty-v-hwy graph from data set mpg  
 geom\_point () + # crate scatterplot with points jittered  
 labs(title = "Comparison of Anxiety -v- Workout Intesnsity", # create title above graph   
 subtitle ="For 35 paitents with Muscle Dismorphia", # create subtitle below title  
 x = "Anxiety PCA Dimension 4", # label x-axis  
 y = "Intensity", # label y-axis  
 caption = "[source: data from our client Dr. SantaBarbara]") # insert captionbelow graph # display graph  
  
g2 <- g1 + # create graph g2 starting with graph g1  
 geom\_smooth(method = "lm", se = 0) # add linear regression line with standard error envelope  
g2 # display graph

#To run stepwise regrression:  
#First, let us define the null (intercept-only) model. We need this to build our forward stepwise regression:  
intercept\_only1 <- lm(DAYSPASTWEEK ~ 1, data = df3)  
summary.lm(intercept\_only1)

#Next, let us define the model with all explanatory variables included. We need this both for forward and backward stepwise regression:  
  
all1 <- lm(RTINTENSITY ~ ., data = df3)  
summary.lm(all1)

#If we want to see the output of the forward stepwise regression, we can use this command:  
forward$anova

# Note that forward$coefficients does not give us the full summary like summary.lm does. If we want that, we need to fit the model with these predictors and print out the output:  
bestforward <- lm(RTINTENSITY ~ RS.Dim.4 + MINSPERWORKOUT + SPAS.Dim.3 + STAI.Dim.5 + STAI.Dim.4 + STAI.Dim.2, data = df3)  
summary.lm(bestforward)

# Plotting the forward stepwise linear regression model based off of the most significant variable  
g3 <- ggplot(data = df3, aes(x = RS.Dim.4, y = RTINTENSITY)) +   
 geom\_point () +   
 labs(title = "Comparison of Nutrition Mental Health -v- Workout Intesnsity",   
 subtitle ="For 35 paitents with Muscle Dismorphia",   
 x = "Nutrition PCA Dimension 4",   
 y = "Intensity",   
 caption = "[source: data from our client Dr. SantaBarbara]") +   
 geom\_smooth(method = "lm", se = 0, colour = "red")   
g3

## `geom\_smooth()` using formula 'y ~ x'

## Make regression trees

library(gbm)

library(tree)

library(randomForest)

library(tidyverse)

head(df3)

# Bottom of the node = the level of intensity   
tree\_model <- tree(RTINTENSITY ~ ., df3)  
  
plot(tree\_model)  
text(tree\_model, pretty = 0, cex = 0.7)

summary(tree\_model)

pred <- predict(tree\_model, df3)  
mean((pred - df3$RTINTENSITY)^2)

#calculate residual standard error  
sqrt(deviance(tree\_model)/df.residual(tree\_model))

set.seed(310)  
  
cv\_tree\_model <- cv.tree(tree\_model, K = 10)

data.frame(n\_leaves = cv\_tree\_model$size,  
 CV\_RSS = cv\_tree\_model$dev) %>%  
 mutate(min\_CV\_RSS = as.numeric(min(CV\_RSS) == CV\_RSS)) %>%  
 ggplot(aes(x = n\_leaves, y = CV\_RSS)) +  
 geom\_line(col = "grey55") +  
 geom\_point(size = 2, aes(col = factor(min\_CV\_RSS))) +  
 scale\_x\_continuous(breaks = seq(1, 17, 2)) +  
 scale\_y\_continuous(labels = scales::comma\_format()) +  
 scale\_color\_manual(values = c("deepskyblue3", "green")) +  
 theme(legend.position = "none") +  
 labs(title = "Muscle Dismorphia Dataset - Regression Tree",  
 subtitle = "Selecting the complexity parameter with cross-validation",  
 x = "Terminal Nodes",  
 y = "CV RSS")

# Seeing in pruning the tree helps.  
pruned\_tree\_model <- prune.tree(tree\_model, best = 2)  
cv.pred <- predict(pruned\_tree\_model, df3)  
mean((cv.pred - df3$RTINTENSITY)^2)