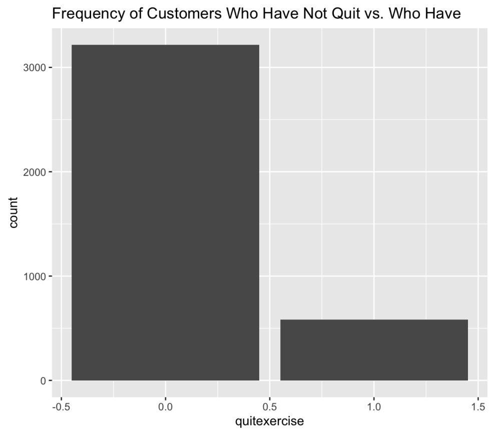
Dana DiVincenzo

In this assignment, I was given information about different users of a popular fitness app which tracks stress, provides an exercise program, and aims to improve overall health. I developed a logistic model to predict whether or not the app user would quit. The goal of the app company is to gain as many users as possible who will not quit, so they can maximize revenue. One risk that could harm the company would be spending money advertising to people who are unlikely to buy the app, or who purchase the app and then quit soon after.

An initial graphical examination of the data showed that there are far more users in our dataset who have not quit the app than those who have quit. The graph is below.

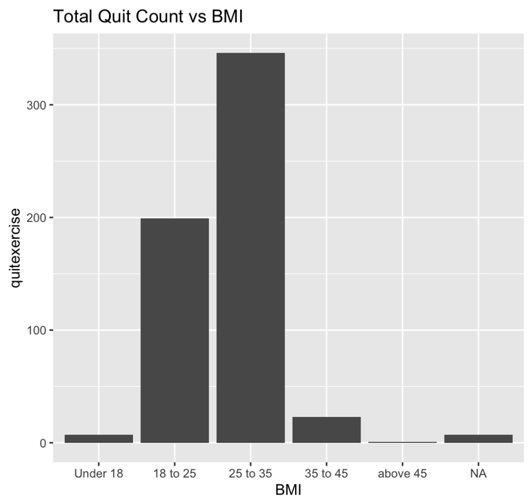


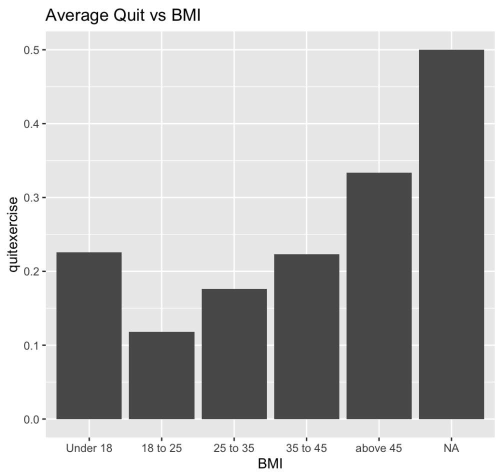
I then created a logistic model using the data given in the dataset about the customers. Aftersome tuning, I created a model which predicted whether or not a user would quit the app, based on the users gender, age, and if they do aerobic exercise daily. The accuracy of this model was 84.6%. This is extremely close to the accuracy of the pre-tuning model which contained all independent variables supplied. By decreasing the amount of independent variables used, we decrease the risk of overfitting, as well as decrease the amount of processing time for a computer to run this model on a very large version of the dataset.

All variables in this model were significant. By examining the odds ratio of the coefficients, we can see that gender = female increases the odds of quitting the app compared to males. We can also see that aerobicdaily = yes also increases the odds of quitting the app compared to aerobicdaily = no. An increase in age causes an increase in likelihood of quitting the app as well.

This information says to me that people who already exercise daily find that they don’t need an app to work out, and end up quitting because the cost is not worth it. I would recommend marketing the app more towards people who are beginner exercisers, or do not work out but wish to get fitter. If the app is not for advanced athletes there is no point in marketing towards them. I would also recommend marketing more towards males, as they are less likely to quit, and will cause larger financial gain over time. I would also recommend targeting the app towards younger adults. Further breakdown could be done within the age group to see which ages are the least likely to quit.

The app company was particularly curious about if users with a BMI from 18 to 25 were different from the others in the dataset in terms of their ability to maintain the exercise program without quitting. Visualizations were made to show the frequency of quits within this group, and the average number of quits within this group. These graphs are below.





We can see based on the graphs that although the total number of quits in this group is in the middle to higher end compared to others, on average, they actually have the lowest quit rate. It makes sense that most customers in this dataset would have a BMI between 18 and 25, as that is the healthy range for adults. The higher BMI groups have a higher average quit rate, so I would advise against marketing towards those with extremely high BMIs.

The company also gave a payoff formula of a false negative costing $20 and a true positive gaining $39.99. Given these numbers, the threshold was calculated to be .01, or 10%. However, this is an extremely low threshold. This can mean a few things: the calculation may have been done incorrectly, the data supplied by the company may be wrong, or the costs of a false negative is too high compared to the gains of a true positive. If it is an issue of a mistake/miscommunication, some simple fixing could be done to the calculation to either verify the threshold or find the correct one.

```
library(Hmisc)
library(survival)
library(rms)
library(ggplot2)
library(dplyr)
library(readr)
library(psych)

# read in the file
appdata<-read.csv("/Users/dana/Downloads/WorkoutAppBurnout.csv")
```

```
#displays string representation of data in each col
str(appdata, give.attr = FALSE)
summary(appdata)

# fixing the values for MultiVitamin
unique(appdata[,"MultiVitamin"])
appdata <- appdata[appdata$MultiVitamin != 5,]
unique(appdata[,"MultiVitamin"])

# quitexercise is my dependant variable and does not need to be factored
# Needs to be factored:
# gender, smoker, multivitamin, aerobicdaily, marital status
# lets assume here that education is heirarchial (ie, bachelors is better than high school) and
does not need to be factored

appdata$Gender <- as.factor(appdata$Gender)
appdata$Smoker <- as.factor(appdata$Smoker)
appdata$MultiVitamin <- as.factor(appdata$MultiVitamin)
appdata$aerobicdaily <- as.factor(appdata$aerobicdaily)
# divorced, married, single
appdata$maritalstatus <- as.factor(appdata$maritalstatus)


str(appdata, give.attr = FALSE)

# making a frequency plot of the dependant variable
ggplot(appdata, aes(x= quitexercise)) + geom_histogram(stat = "count") + labs(title =
"Frequency of Customers Who Have Not Quit vs. Who Have ")

################ making the models

# making a logistic regression model with all independent variables
# set family = binomail
fullappmodel <- glm(quitexercise ~ Gender + Age + Education + Smoker + CoffeeOzPerDay +
                  MultiVitamin + aerobicdaily + timeatdesk + meetingtime + CommuteTime
                + BMI + RestingHeart +maritalstatus + incomeindollars, family = binomial,
                  data = appdata)

# this summary tells me that Gender1 (female), Age, CoffeeOzPerDay, meetingtime, and
RestingHeart are significant
summary(fullappmodel)

# using stepAIC to make tuned model
appmodelAIC <- stepAIC(fullappmodel, trace=0)
```

```
# only variables here are Gender1 (female), Age, CoffeeOzPerDay, aerobicdaily1, timeatdesk,
meetingtime, and RestingHeart
# Gender1 (female), Age, CoffeeOzPerDay, meetingtime, and RestingHeart are significant
summary(appmodelAIC)

# finding the coefficients of the model in terms of odds
# as each variable increases, the odds of the dependent variable increase by the associated
coeff
# this tells me that timeatdesk is equal to 1, and meetingtime is equal to 1.01, so lets remove
them and do it again
coefsexp <- coef(appmodelAIC) %>% exp() %>% round(2)
coefsexp

# making a new model with variables removed
newappmodel <- glm(quitexercise ~ Gender + Age + CoffeeOzPerDay + aerobicdaily +
RestingHeart, family = binomial,
                   data = appdata)
# everything in this model is significant
summary(newappmodel)

# finding the coefficients of the model in terms of odds
# as each variable increases, the odds of the dependent variable increase by the associated
coeff
# this tells me that coffeeozperdat is equal to 1.02, and restingheart is equal to 1.01, so lets
remove them and do it again
coefsexp <- coef(newappmodel) %>% exp() %>% round(2)
coefsexp

# making a new model with variables removed
newappmodel2 <- glm(quitexercise ~ Gender + Age + aerobicdaily, family = binomial,
                  data = appdata)
# everything in this model is significant
summary(newappmodel2)

# everything here is above 1
coefsexp <- coef(newappmodel2) %>% exp() %>% round(2)
coefsexp

############# now lets test the accuracy of each of our models, if they get significantly less
accurate I can use the models with more variables

### first on fullappmodel
# creating a column called predFill
```

```
appdata$predFull <- predict(fullappmodel, type = "response", na.action = na.exclude)

# building the confusion matrix
# lets just have our threshold as .5 for now
confMatrixModelFull <- confusion.matrix(appdata$quitexercise, appdata$predFull, threshold =
.5)
confMatrixModelFull

#calculating the accuracy of the full model
accuracyFull <- sum(diag(confMatrixModelFull)) / sum(confMatrixModelFull)
accuracyFull #0.853 this means, 85% of the time, my model predicted the category correctly


### now on appmodelAIC
appdata$predFull2 <- predict(appmodelAIC, type = "response", na.action = na.exclude)

# building the confusion matrix
# lets just have our threshold as .5 for now
confMatrixModelFull2 <- confusion.matrix(appdata$quitexercise, appdata$predFull2, threshold =
.5)
confMatrixModelFull2

#calculating the accuracy of the full model
accuracyFull2 <- sum(diag(confMatrixModelFull2)) / sum(confMatrixModelFull2)
accuracyFull2 #0.853 this means. This did not change from the full model


### now on newappmodel
appdata$predFull3 <- predict(newappmodel, type = "response", na.action = na.exclude)

# building the confusion matrix
# lets just have our threshold as .5 for now
confMatrixModelFull3 <- confusion.matrix(appdata$quitexercise, appdata$predFull3, threshold =
.5)
confMatrixModelFull3

#calculating the accuracy of the full model
accuracyFull3 <- sum(diag(confMatrixModelFull3)) / sum(confMatrixModelFull3)
accuracyFull3 #0.852. This is an insignificant change


### now on newappmodel2
appdata$predFull4 <- predict(newappmodel2, type = "response", na.action = na.exclude)
```

```
# building the confusion matrix
# lets just have our threshold as .5 for now
confMatrixModelFull4 <- confusion.matrix(appdata$quitexercise, appdata$predFull4, threshold =
.5)
confMatrixModelFull4

#calculating the accuracy of the full model
accuracyFull4 <- sum(diag(confMatrixModelFull4)) / sum(confMatrixModelFull4)
accuracyFull4 #0.846. This is an insignificant change

# the accuracy stayed basically the same, so lets use newappmodel2

# lets crossvalidate the model if we have time at the end


####### Finding the best threshold

# payoff = $39.99 * true positives – $20 * false negative

#this creates a dataframe and an empty column called "payoff"
#threshold payoff
# the threshold is the best cutoff.We are going to do threshold tuning
#1      0.1     NA
#2      0.2     NA
#3      0.3     NA
#4      0.4     NA
#5      0.5     NA
payoffMatrix <- data.frame(threshold = seq(from = 0.1, to = 0.5, by = 0.1), payoff = NA)
payoffMatrix

# builds confusion matrices in an itertive manner with varying thresholds
# calculates the payoff and saves it to the corresponding row that is created
# we can look at the values in this matrix and identify the threshold that provides the maximum
payoff
#threshold  payoff
#threshold payoff
#1      0.1  17735 # this row has the highest value
#2      0.2   7477
#3      0.3  -3081
#4      0.4  -8780
#5      0.5 -10940
for(i in 1:length(payoffMatrix$threshold)) {
  confMatrix <- confusion.matrix(appdata$quitexercise, appdata$predFull4, threshold =
payoffMatrix$threshold[i])
```

```
  payoffMatrix$payoff[i] <- confMatrix[2,2]*39.99 + confMatrix[1,2]*(-20)
}

payoffMatrix


########## Are people who have a BMI between 18-25 different from the others in the dataset
in terms of
# their ability to maintain the exercise program without quitting?


unique(appdata[,"BMI"])
range(appdata[,"BMI"])
appdata$BMI <- cut(appdata$BMI, c(12,18,25,35,45,57), labels = c("Under 18", "18 to 25", "25
to 35", "35 to 45", "above 45"))

str(appdata, give.attr = FALSE)

ggplot(data=appdata) +
  aes(x=BMI, y=quitexercise) +
  geom_bar(stat="identity") + labs(title = "Total Quit Count vs BMI")


ggplot(data=appdata) +
  aes(x=BMI, y=quitexercise) +
  geom_bar(stat="summary") + labs(title = "Average Quit vs BMI")

######## threshold calculation
```