BIOS 611 Homework 4

Lauren Kanapka

Problem 1

Usually it would be necessary to split the data into three parts: training, validation, and testing. However, for this application I will not be tuning parameters and so I do not need a separate validation and testing data set. Therefore, I will split the data into two parts. The following code reads in the data and splits it into training and testing. I have chosen to split the data separately by gender.

```
library(tidyverse)
library(MLmetrics)

f1<- MLmetrics::F1_Score

set.seed(884952)

genderData <- read_csv("500_Person_Gender_Height_Weight_Index.csv")

#Split the data into male and female
male<-genderData %>% filter(Gender=="Male")
female<-genderData %>% filter(Gender=="Female")

#Separately divide the male and female data sets into train and test
setMale<-factor(c(rep("Train", nrow(male)*0.6),
                  rep("Test",nrow(male)*0.4))) %>%
         sample(nrow(male), replace=FALSE)
male$set<-setMale

setFemale<-factor(c(rep("Train", nrow(female)*0.6),
                    rep("Test",nrow(female)*0.4))) %>%
           sample(nrow(female), replace=FALSE)
female$set<-setFemale

# Put all the data together
genderData<- rbind(male, female)
genderData$femaleFlg<- genderData$Gender=="Female"

# Put train and test in separate data sets
train<-genderData %>% filter(set=="Train")
test<- genderData %>% filter(set=="Test")
```

Next, fit a logistic regression model to the training data set with gender as the outcome and height and weight as predictors. The accuracy of this model is checked on the test data set. The F1 score is not a very good measure in this context because there is no concept of "true" and "false". However, I have calculated the score by arbitrarily selected female to represent "true". Note that the F1 score will be different if I have chosen male.

```
# Fit a logistic regression model on training data
modelGLM<- glm(femaleFlg ~ Weight + Height, family=binomial(link = "logit"),
            data=train)

# Calculate accuracy on test data
pred <- predict(modelGLM, newdata=test, type="response")
sum((pred>0.5)== test$femaleFlg)/nrow(test)
```

```
## [1] 0.49
```

```
# Calculate F1 score on test data
f1(test$femaleFlg, pred>0.5, positive=TRUE)
```

```
## [1] 0.5603448
```

Since the accuracy is close to 0.5, this model is no better than a model that randomly guesses gender.

Problem 2

Using the same training data set as problem 1, the following code fits a gradient boosting machine model. The accuracy of this model is checked on the testing data set.

```
library(gbm)

# Fit a GBM on training data
modelGBM <- gbm(femaleFlg ~ Weight + Height, distribution ="bernoulli",
                data=train, n.trees = 100, interaction.depth = 2,
                shrinkage=0.1)

# Calculate accuracy on test data
pred <- predict(modelGBM, newdata=test, type="response")
sum((pred>0.5)== test$femaleFlg)/nrow(test)
```

```
## [1] 0.45
```

```
# Calculate F1 score on test data
f1(test$femaleFlg, pred>0.5, positive=TRUE)
```

```
## [1] 0.5
```

It doesn't matter how we tune the parameters, the GBM has poor accuracy that is not any better at prediction than randomly selecting a gender.

Problem 3

The following code creates a new data set that contains only 50 males but all of the original females. This new data set is then split into training and test data sets again.

```
# Randomly select 50 males from the original data
sampleMale<- sample(seq(1,nrow(male)), 50, replace=FALSE)
male<- male[sampleMale,]

# Split the 50 males into train and test
setMale<-factor(c(rep("Train", nrow(male)*0.6),
                  rep("Test",nrow(male)*0.4))) %>%
         sample(nrow(male), replace=FALSE)
male$set<-setMale
```

```r
# Use the same female data as before and put both genders back together
genderData<- rbind(male, female)
genderData$femaleFlg<- genderData$Gender=="Female"

# Put train and test in separate data sets
train<-genderData %>% filter(set=="Train")
test<- genderData %>% filter(set=="Test")
```

The following code fits a logistic regression model on this new training data and checks the accuracy on the test data. Since the new data is imbalanced, a different threshold may be more appropriate than 0.5. This will be checked in the next question using an ROC plot.

```r
modelGLM<- glm(femaleFlg ~ Weight + Height, family=binomial(link = "logit"),
            data=train)

pred <- predict(modelGLM, newdata=test, type="response")
sum((pred>0.5)== test$femaleFlg)/nrow(test)
```

```
## [1] 0.8360656
```

```r
# The f1 function does not work when all values of pred are true
# Manually calculate F1
tp<- sum(test$femaleFlg==1 & (pred>0.5)==1)
fp<- sum(test$femaleFlg==0 & (pred>0.5)==1)
fn<- sum(test$femaleFlg==1 & (pred>0.5)==0)
tp/(tp+0.5*(fp+fn))
```

```
## [1] 0.9107143
```

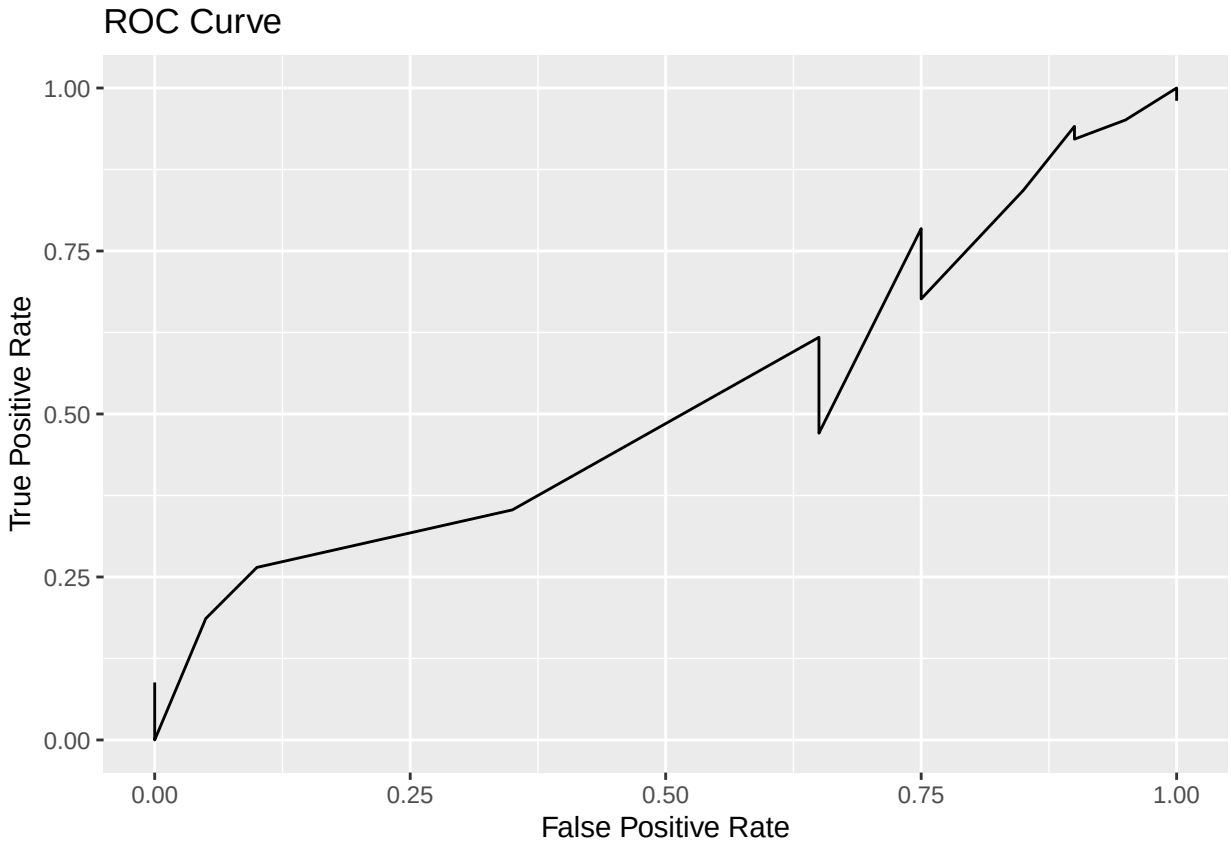
The accuracy and F1 score only appear better than guessing because the data are imbalanced and we chose a threshold of 0.5. We check in the next problem if there is a more appropriate threshold.

Problem 4

The following code creates an ROC curve for the logistic regression model in the previous question.

```r
roc <- do.call(rbind, Map(function(threshold){
    p <- pred > threshold
    tp <- sum(p[test$femaleFlg])/sum(test$femaleFlg)
    fp <- sum(p[!test$femaleFlg])/sum(!test$femaleFlg)
    tibble(threshold=threshold,
           tp=tp,
           fp=fp)
},seq(100)/100))

ggplot(roc, aes(fp,tp)) + geom_line() + xlim(0,1) + ylim(0,1) +
    labs(title="ROC Curve",x="False Positive Rate",y="True Positive Rate")
```



There is no clear threshold that would produce the best balance between true positive and false positives. This suggests that this is not a good model for the data.

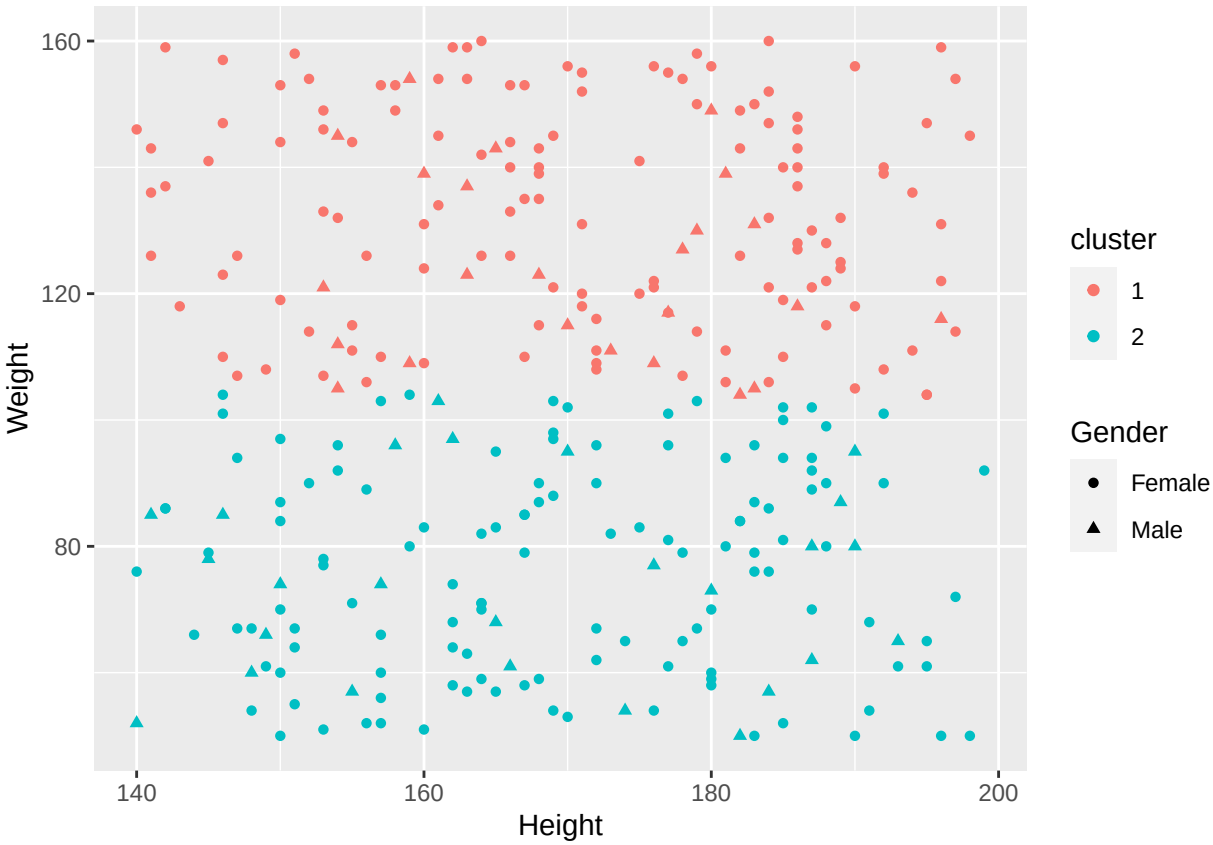
Problem 5

The following code uses k means to put the height and weight data from the data that was imbalanced with respect to gender into two clusters.

```
kmeans<- kmeans(data.frame(genderData$Height, genderData$Weight), 2)

genderData$cluster<- as.factor(kmeans$cluster)

ggplot(genderData, aes(x=Height, y=Weight, color=cluster))+
  geom_point(aes(shape=Gender))
```



Looking at the plot it is clear that the clusters are not meaningful and do not correspond to gender. This is consistent with all the other model results. In this data set height and weight are not predictive of gender.