BIOS 611 HW 4

Andrew Walther

10/12/2020

```
library(tidyverse)
library(caret)
library(MLmetrics)
library(ROCR)
library(ModelMetrics)
BMI_Gender <- read.csv("~/OneDrive - University of North Carolina at Chapel Hill/UNC - Fall 2020/BIOS 6
#gender indicator, male = 1 & female = 0
BMI_Gender <- BMI_Gender %>% mutate(gender = ifelse(Gender == "Male",1,0))
```

Problem 1

Build a glm in R to classify individuals as either Male or Female based on their weight and height. What is the accuracy of the model?

```
#creates training and testing data sets from the full data set (20% test, 80% train)
set.seed(611)
trainIndex <- createDataPartition(BMI_Gender$gender, p = .8, list = FALSE, times = 1)
TRAIN <- BMI_Gender[trainIndex,]
TEST  <- BMI_Gender[-trainIndex,]
```

```
#glm model to predict gender (0,1)
model_glm <- glm(gender ~ Weight + Height, data = TRAIN, family=binomial)
summary(model_glm)
```

```
##
## Call:
## glm(formula = gender ~ Weight + Height, family = binomial, data = TRAIN)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.258  -1.180   1.100   1.168   1.246
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.420233   1.103329   0.381    0.703
## Weight       0.001592   0.003069   0.519    0.604
## Height      -0.003328   0.006178  -0.539    0.590
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 554.48  on 399  degrees of freedom
## Residual deviance: 553.92  on 397  degrees of freedom
## AIC: 559.92
##
## Number of Fisher Scoring iterations: 3
```

The Akaike information criterion (AIC) value for this glm model on the training data is 559.9.

```
#make predictions for gender
TEST$model_prob <- predict(model_glm, TEST, type="response")
TEST <- TEST  %>% mutate(model_pred = 1*(model_prob > .50) + 0, gender_binary = 1*(gender == 0) + 0)

#calculate prediction accuracy
TEST <- TEST %>% mutate(accurate = 1*(model_pred == gender_binary))
sum(TEST$accurate)/nrow(TEST)
```

```
## [1] 0.54
```

The model predicts the correct gender with about 54% accuracy.

Problem 2

Use the gbm package to train a similar model. Don't worry about hyper-parameter tuning for now. What is the accuracy of the model?

```
library(gbm)
```

```
## Loaded gbm 2.1.8
```

```
model_gbm <- gbm(gender ~ Height + Weight, distribution="bernoulli", data = TRAIN, n.trees = 100, intera
summary(model_gbm, plot=FALSE)
```

```
##           var  rel.inf
## Weight Weight 54.03246
## Height Height 45.96754
```

```
#make predictions for gender
TEST$model_prob <- predict(model_gbm, TEST, type="response")
```

```
## Using 100 trees...
```

```
TEST <- TEST  %>% mutate(model_pred = 1*(model_prob > .50) + 0, gender_binary = 1*(gender == 0) + 0)

#calculate prediction accuracy
TEST <- TEST %>% mutate(accurate = 1*(model_pred == gender_binary))
sum(TEST$accurate)/nrow(TEST)
```

```
## [1] 0.53
```

The gbm model appears to only achieve 53% accuracy when making predictions on gender.

Problem 3

Filter the data set so that it contains only 50 Male examples. Create a new model for this data set. What is the F1 Score of the model?

```
#load data and select 50 males along with all females
BMI_Gender <- read.csv("~/OneDrive - University of North Carolina at Chapel Hill/UNC - Fall 2020/BIOS 6
BMI_Gender <- BMI_Gender %>% mutate(gender = ifelse(Gender == "Male",1,0))
BMI_Gender <- BMI_Gender %>% arrange(Gender)
```

```
BMI_Gender_50males <- BMI_Gender[1:305,]
BMI_Gender <- BMI_Gender[,2:5]
```

```
#recreate train/test sets
trainIndex <- createDataPartition(BMI_Gender_50males$gender, p = .8, list = FALSE, times = 1)
TRAIN <- BMI_Gender_50males[trainIndex,]
TEST  <- BMI_Gender_50males[-trainIndex,]
```

```
#updated gbm model
model <- gbm(gender ~ Height + Weight, distribution = "bernoulli", data = TRAIN, n.trees = 100, interac
pred <- predict(model, TEST, type = "response")
```

```
## Using 100 trees...
```

```
sum((pred>0.5)==TEST$gender)/nrow(TEST)
```

```
## [1] 0.8032787
```

```
#f1 score calculation
F1_Score(TEST$gender, as.numeric(pred > 0.5))
```

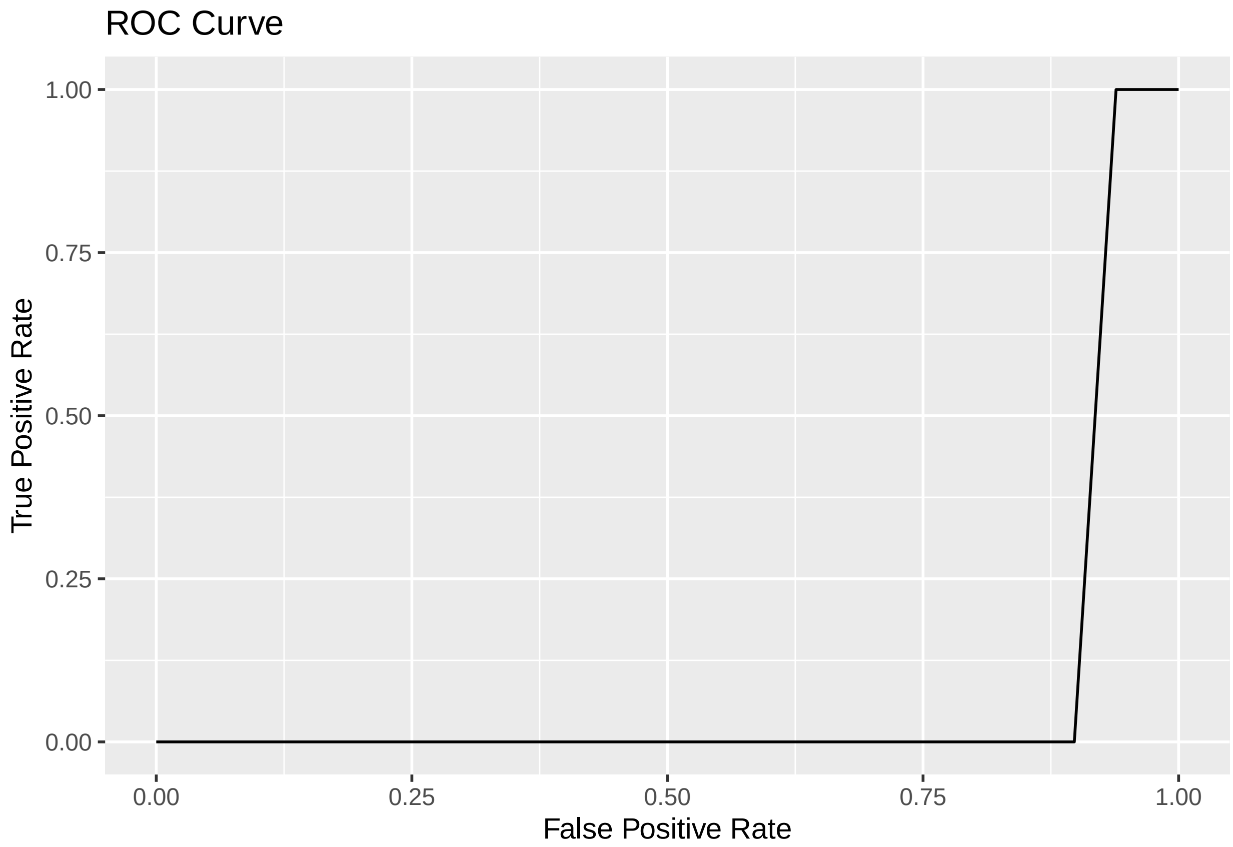
> When we include only 50 males (and all of the females), we get a prediction accuracy of about 801% and an F1 score of 0.92. The function returns 0.92 when executed inside R, but there is always an issue where the error "Error in FUN(X[[i]], . . .) : only defined on a data frame with all numeric variables" occurs when this function is left active and the markdown document knits. To avoid this, I set this function to not evaluate when the document knits.

Problem 4

For the model in the previous example, plot an ROC curve. What does this ROC curve mean?

```
roc <- do.call(rbind, Map(function(threshold){
    p <- pred > threshold;
    tp <- sum(p[TEST$gender])/sum(TEST$gender);
    fp <- sum(p[!TEST$gender])/sum(!TEST$gender);
    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");
```



An optimal ROC curve "hugs" the top left corner of the graph indicating that the true positive rate nearly reaches 1 before the false positive rate increases. However, in our example, the false positive rate shot up to over 0.9 before the true positive rate increased toward 1. Since we have little "area under the curve" compared to what could be possible, classifier is not a very high performer as judged by the ROC curve.

Problem 5

Using K-Means, cluster the same data set. Can you identify the clusters with the known labels? Provide an interpretation of this result.

```
BMI_Gender <- read.csv("~/OneDrive - University of North Carolina at Chapel Hill/UNC - Fall 2020/BIOS 6
BMI_Gender <- BMI_Gender %>% mutate(gender = ifelse(Gender == "Male",1,0))
BMI_Gender <- BMI_Gender[,2:5]

library(Rtsne)
library(factoextra)
```

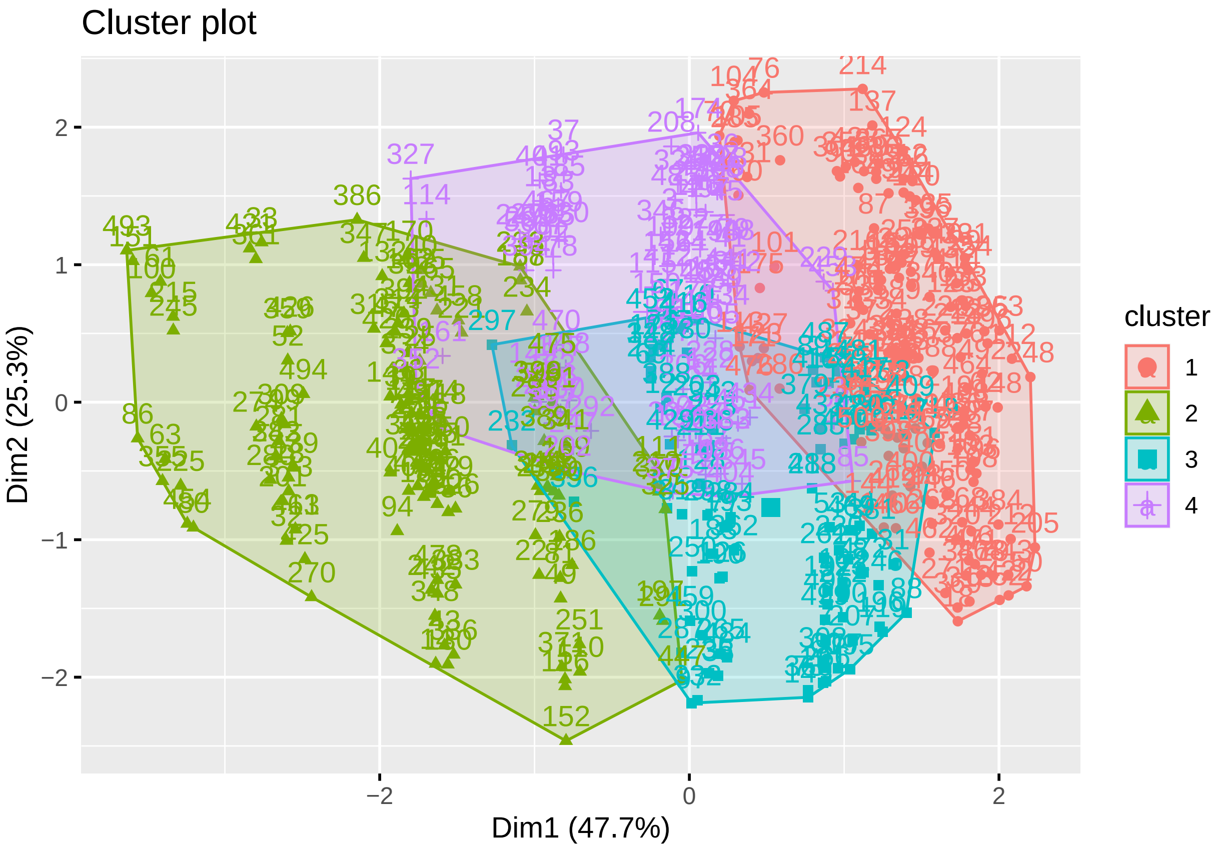
```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(zip)
```

```
##
## Attaching package: 'zip'

## The following objects are masked from 'package:utils':
##
##     unzip, zip
```

```
kmeans_gender <- kmeans(x = BMI_Gender, centers = 4)
fviz_cluster(list(data=BMI_Gender, cluster=kmeans_gender$cluster))
```



For all n number of clusters, there is a significant amount of overlap among observations in different clusters. Additionally, the clusters aren't really identifiable as "males" or "females" or by any specific height/weight so this clustering isn't particularly informative. If each individual had a given name or identifier that would match them to their characteristics, clustering like this might be a bit more informative!