HW4_Virkud

Arti Virkud

10/7/2020

Datasetup

Problem 1:

Build a glm in R to classifier individuals as either Male or Female based on their weight and height.

```
set.seed(2020)
N <- length(dat$female)
index.train <- sample(N, N*2/3)
dat.train <- dat[index.train,]
dat.test <- dat[-index.train,]
model1 <- glm(female ~ Height + Weight, data=dat.train)
```

What is the accuracy of the model?

```
dat.test$model_prob <- predict(model1, dat.test, type = "response")
dat.test2 <- dat.test  %>% mutate(model_pred = 1*(model_prob > .50) + 0)
dat.test3 <- dat.test2 %>% mutate(accurate = 1*(model_pred == female))
sum(dat.test3$accurate)/nrow(dat.test3)
```

```
## [1] 0.4610778
```

The accuracy is 46.1%. ## Problem 2: Use the 'gbm' package to train a similar model. Don't worry about hyper parameter tuning for now.

```
set.seed(2020)
N <- length(dat$female)
index.train <- sample(N, N*2/3)
dat.train <- dat[index.train,]
dat.test <- dat[-index.train,]
model2 <- gbm(female ~ Height + Weight, data=dat.train)
```

```
## Distribution not specified, assuming bernoulli ...
```

What is the accuracy of the model?

```
dat.test$model_prob2 <- predict(model2, dat.test, type = "response")
```

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

```
dat.test2 <- dat.test  %>% mutate(model_pred2 = 1*(model_prob2 > .50) + 0)
dat.test3 <- dat.test2 %>% mutate(accurate2 = 1*(model_pred2 == female))
sum(dat.test3$accurate2)/nrow(dat.test3)
```

```
## [1] 0.4790419
```

The accuracy is 47.9%.

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?

```
set.seed(2020)
ftr_male <- dat %>% filter(female==0) %>% slice_sample(n=50)
ftr_female <- dat %>% filter(female==1)
ftr <- rbind(ftr_male, ftr_female)
N <- length(ftr$female)
index.train <- sample(N, N*2/3)
ftr.train <- ftr[index.train,]
ftr.test <- ftr[-index.train,]
model3 <- glm(female ~ Height + Weight, data=ftr.train)

ftr.test$model_prob <- predict(model3, ftr.test, type = "response")
ftr.test2 <- ftr.test  %>% mutate(pred = 1*(model_prob > .50) + 0)
confusionMatrix(data=as.factor(ftr.test2$pred), reference=as.factor(ftr.test2$female), mode = "prec_reca
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0  0  0
##          1 15 87
##
##                Accuracy : 0.8529
##                  95% CI : (0.7691, 0.9153)
##     No Information Rate : 0.8529
##     P-Value [Acc > NIR] : 0.5683057
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 0.0003006
##
##               Precision :     NA
##                  Recall : 0.0000
##                      F1 :     NA
##              Prevalence : 0.1471
##          Detection Rate : 0.0000
##    Detection Prevalence : 0.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 0
##
```

```
f1=87/(87+0.5*(15))
```

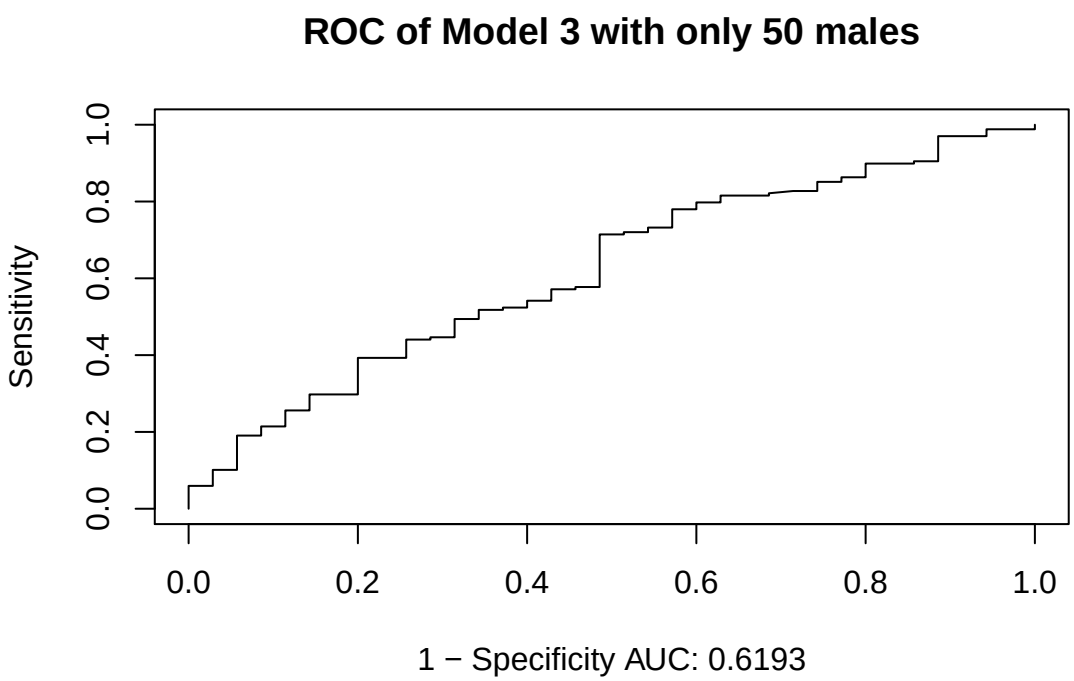
The F1 score is 0.921. ##Problem 4 For the model in the previous example plot an ROC curve. What does this ROC curve mean?

```
set.seed(2020)
fit.roc <- roc(ftr.train$female, model3$fitted)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
plot(1-fit.roc$specificities, fit.roc$sensitivities, col="black", pch=16,type="l",
     xlab=paste("1 - Specificity AUC:",
                round(pROC::auc(fit.roc),4)
                ),
     ylab="Sensitivity")
title("ROC of Model 3 with only 50 males")
```

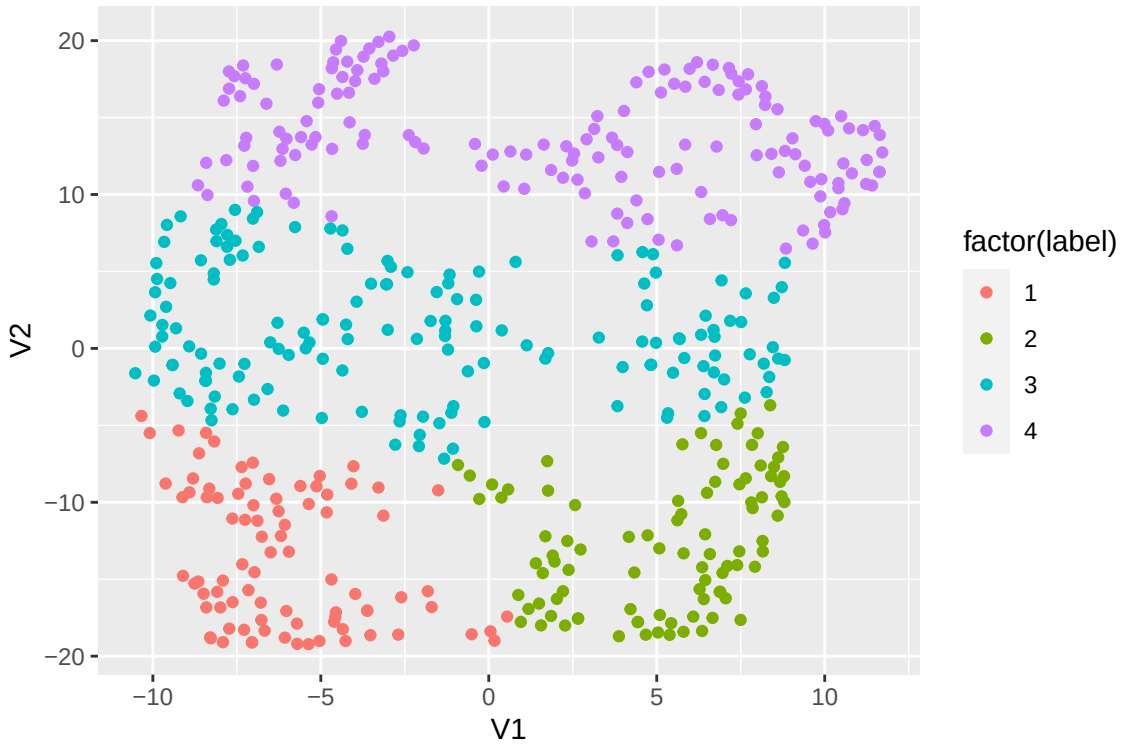


The ROC curve shows the values for the sensitivity (true positive rate, or what proportion of the positive values are correctly identified as positive) and 1 - specificity (taken together, this is also known as the false positive rate, or the proportion of true negatives that are predicted as positive) for different threshold values. We cannot tell which threshold produced each combination of sensitivity and specificity plotted above. An ideal model would be one that correctly identifies all positive value and all negative values (which would be when sensitivity = 1 and 1 - specificity = 0). As such, "better" models will be ones that are closer to the lefthand corner of the graph. Another way of examining is to calcultae the area under the curve. The closer AUC is to 1, the "better" the model.

This indicates that this model performs better than random guessing.

##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.

```
dat2 <- data %>% mutate(female = if_else(Gender=="Female",1,0)) %>% select(-Gender, -Index) %>% distinc
set.seed(0)
cluster <- kmeans(x=dat2,4)
fit <- Rtsne(dat2, dims = 2)
ggplot(fit$Y %>% as.data.frame() %>% as_tibble() %>% mutate(label=cluster$cluster),aes(V1,V2)) +
    geom_point(aes(color=factor(label)))
```




```
cluster$centers
```

```
##     Height    Weight    female
## 1 183.8256  69.13953 0.5000000
## 2 153.6848  72.10870 0.5326087
## 3 173.2941 108.79085 0.5098039
## 4 168.5253 144.25949 0.5063291
```

The six clusters are distinct with no delineation by male and female (all the cluster centers are close to ~0.5). The clusters seem to distinguish by groups of individuals who are above average height and below average weight (cluster 1), below average height and weight (cluster 2), above average height and weight (cluster 3), and average height and above average weight (cluster 4).