Homework 2

Group 1

Contents

Prepared for:

Dr. Nathan Bastian

City University of New York, School of Professional Studies - Data 621

Prepared by:

Group 1

Senthil Dhanapal

Yadu Chittampalli

Christophe Hunt

1 Data Source

The data is a set of actual classes and predicted classes as provided by Dr. Nathan Bastian through Blackboard. We uploaded the data to our public GitHub repository.

2 Data Explained

We will be using the following columns from the data source:

- class: the actual class for the observation
- scored.class: the predicted class for the observation (based on a threshold of 0.5)
- scored.probability: the predicted probability of success for the observation

Below is the raw confusion matrix for our scored data set

	Predicted Failure	Predicted Success
Actual Failure	119	5
Actual Success	30	27

Here the rows represent the actual classes and the columns represent the predicted classes.

3 Function for Accuracy of Predictions

We developed a function that takes the data set as a dataframe, with actual and predicted classifications identified, and returns the accuracy of the predictions. - Yadu

```
accuracy <- function(df, actual, prediction){

  library(scales)
  library(dplyr)
  library(tidyr)

  if (sum(colnames(df) %in% c(actual, prediction)) != 2){
    return("One or more columns were not found, please verify selections")
  }
  truefalse <- as.data.frame(table(df[actual] == df[prediction])) %>%
               spread('Var1', 'Freq')

  accuracy <- unlist(truefalse[2]/nrow(df))

  return(as.numeric(accuracy))
}

accuracy(scores,"class", "scored.class" )
```

[1] 0.8066298

4 Function Classification Error Rate of Predictions

We developed a function that takes the data set as a dataframe, with actual and predicted classifications identified, and returns the classification error rate of the predictions. It verifies that the accuracy and an error rate sums to one.

```
errorrate <- function(df, actual, prediction){

  truefalse <- data.frame(table(df[actual] == df[prediction])) %>%
                          spread('Var1', 'Freq')

  error <- unlist(truefalse[1]/nrow(df))

  if (error + accuracy(df, actual, prediction) != 1){return("Accuracy and Error Rate does not sum to 1")

  return(as.numeric(error))
}

errorrate(scores, "class", "scored.class")
```

```
## [1] 0.1933702
```

5 Function for Precisions of Predictions

Write a function that takes the data set as a dataframe, with actual and predicted classifications identified, and returns the precision of the predictions. - Senthil

6 Function for Sensitivity of Predictions

Write a function that takes the data set as a dataframe, with actual and predicted classifications identified, and returns the sensitivity of the predictions. Sensitivity is also known as recall. - Senthil

7 Function for Specificity of Predictions

```
Specificity <- function(df, actual, prediction){

  if(sum(colnames(df) %in% c(actual, prediction)) != 2){
    return("One or more columns were not found, please verify selections")
  }
  m <- as.data.frame(table(df[[actual]], df[[prediction]]))
  true_negative  <- m$Freq[m$Var1 == 0 & m$Var2 == 0]
  false_positive <- m$Freq[m$Var1 == 0 & m$Var2 == 1]
  return(true_negative/(true_negative + false_positive))
  }

paste0("The specificity is ", percent(Specificity(df = scores, actual = 'class', prediction = 'scored.cl
```

```
## [1] "The specificity is 96%"
```

8 F1 Score of Predictions

We developed a function that takes the data set as a dataframe, with actual and predicted classifications identified, and returns the F1 score of the predictions.

```
f1_score <- function(df, actual, prediction){
  require(scales)
  if (sum(colnames(df) %in% c(actual, prediction)) != 2) {
    return("One or more columns were not found, please verify selections")
  }

  m <- as.data.frame(table(df[[actual]], df[[prediction]]))

  true_positive  <- as.numeric(m$Freq[m$Var1 == 1 & m$Var2 == 1])
  false_positive <- as.numeric(m$Freq[m$Var1 == 0 & m$Var2 == 1])
  false_negative <- as.numeric(m$Freq[m$Var1 == 1 & m$Var2 == 0])

  precision <- true_positive/(true_positive + false_positive)
  sensitivity <- true_positive/(true_positive + false_negative)
  f1_score <- ((2 * precision * sensitivity) / (precision + sensitivity))
  return(f1_score)
  }

f1_score(df = scores, actual = 'class', prediction = 'scored.class')
```

```
## [1] 0.6067416
```

9 Bounds of F1 Score of Predictions

Before we move on, let's consider a question that was asked: What are the bounds on the F1 score? Show that the F1 score will always be between 0 and 1. (Hint: If 0 < ???? < 1 and 0 < ???? < 1 then ????????- Christophe

10 Function for ROC curve

Write a function that generates an ROC curve from a data set with a true classification column (class in our example) and a probability column (scored.probability in our example). Your function should return a list that includes the plot of the ROC curve and a vector that contains the calculated area under the curve (AUC). Note that I recommend using a sequence of thresholds ranging from 0 to 1 at 0.01 intervals. - Senthil

11. Use your created R functions and the provided classification output data set to produce all of the classification metrics discussed above. - Christophe

```
accuracy(df = scores, actual = 'class', prediction = 'scored.class')
```

```
## [1] 0.8066298
```

12. Investigate the caret package. In particular, consider the functions confusionMatrix, sensitivity, and specificity. Apply the functions to the data set. How do the results compare with your own functions?

```
library(caret)
confusionMatrix(scores[,1], scores[,2])
```

Confusion Matrix and Statistics

```
      Reference
```

Prediction 0 1 0 119 5 1 30 27

```
           Accuracy : 0.8066
             95% CI : (0.7415, 0.8615)
No Information Rate : 0.8232
P-Value [Acc > NIR] : 0.7559

              Kappa : 0.4916
```

Mcnemar's Test P-Value : 4.976e-05

```
        Sensitivity : 0.7987
        Specificity : 0.8438
     Pos Pred Value : 0.9597
     Neg Pred Value : 0.4737
         Prevalence : 0.8232
     Detection Rate : 0.6575
```

Detection Prevalence : 0.6851
Balanced Accuracy : 0.8212

```
   'Positive' Class : 0
```

Here, the accuracy is rounded up to 4 decimal places. The confusion matrix has rows that represent the predicted classes and columns that represent the actual classes.

13. Investigate the pROC package. Use it to generate an ROC curve for the data set. How do the results compare with your own functions? - Senthil