Homework 2 - Classification Metrics

Daniel Brooks (daniel.brooks@spsmail.cuny.edu), Daniel Fanelli (daniel.fanelli@spsmail.cuny.edu), Christopher Fenton (christopher.fenton@spsmail.cuny.edu), James Hamski (james.hamski@spsmail.cuny.edu), Youqing Xiang (youqing.xiang@spsmail.cuny.edu)

June 21, 2016

Question 1

Download/read the classification output data set (attached in Blackboard to the assignment).

```
library(knitr)
library(ggplot2)
library(caret)
library(pROC)

data <- read.csv('classification-output-data.csv')
```

Question 2

The data set has three key columns we will use:

- **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

Use the table() function to get the raw confusion matrix for this scored dataset. Make sure you understand the output. In particular, do the rows represent the actual or predicted class? The columns?

```
t <- as.data.frame(table(Actual=data$class, Predicted=data$scored.class))
kable(t)
```

Actual	Predicted	Freq
0	0	119
1	0	30
0	1	5
1	1	27

From the above table, we can see that:

- Column (Actual): the actual class
- Column (Predicted): the predicted class
- Column (Freq): the number of observations
- Row 1: there are 119 observations which are class 0 and correctly predicted with class 0.
- Row 2: there are 30 observations which are class 1 but are predicted with class 0.

- Row 3: there are 5 obervations which are class 0 but are predicted with class 1.
- Row 4: there ae 27 obervations which are class 1 and correctly predicted with class 1.

Question 3

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

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

```
accuracy <- function(t)
{
  a <- as.data.frame(table(Actual=t$class, Precidted=t$scored.class))
  return((a$Freq[1] + a$Freq[4])/sum(a$Freq))
}

accuracy.data <- accuracy(data)
round(accuracy.data, 3)
```

```
## [1] 0.807
```

Question 4

Write 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.

$$ClassificationErrorRate = \frac{FP + FN}{TP + FP + TN + FN}$$

```
ER <- function(t)
{
 a <- as.data.frame(table(Actual=t$class, Precidted=t$scored.class))
 return((a$Freq[2] + a$Freq[3])/sum(a$Freq))
}

ER.data <- ER(data)
round(ER.data, 3)
```

```
## [1] 0.193
```

We now verify that the accuracy and error rate sum to 1.

```
ER.data + accuracy.data
```

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

Question 5

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

$$Precision = \frac{TP}{TP + FP}$$

```
precision <- function(t)
{
 a <- as.data.frame(table(Actual=t$class, Precidted=t$scored.class))
 return(a$Freq[4]/(a$Freq[4]+a$Freq[3]))
}

precision.data <- precision(data)
round(precision.data, 3)
```

```
## [1] 0.844
```

Question 6

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.

$$Sensitivity = \frac{TP}{TP + FN}$$

```
sens <- function(t)
{
 a <- as.data.frame(table(Actual=t$class, Precidted=t$scored.class))
 return(a$Freq[4]/(a$Freq[4] + a$Freq[2]))
}

sens.data <- sens(data)
round(sens.data, 3)
```

```
## [1] 0.474
```

Question 7

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

$$Specificity = \frac{TN}{TN + FP}$$

```
spec <- function(t)
{
 a <- as.data.frame(table(Actual=t$class, Precidted=t$scored.class))
 return(a$Freq[1]/(a$Freq[1]+a$Freq[3]))
}

spec.data <- spec(data)
round(spec.data, 3)
```

```
## [1] 0.96
```

96% of the negative cases were correctly identified.

Question 8

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

$$F1Score = \frac{2 \times Precision \times Sensitivity}{Precision + Sensitivity}$$

```
Fscore <- function(t)
{
  a <- as.data.frame(table(Actual=t$class, Precidted=t$scored.class))
  f1s <- 2*a$Freq[4]/(2*a$Freq[4] + a$Freq[2] + a$Freq[3])
  return(f1s)
}

Fscore.data <- Fscore(data)
round(Fscore.data, 3)
```

```
## [1] 0.607
```

Question 9

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 $0 < a < 1$ and $0 < b < 1$ then $ab < a$)

Imagine a study which results in only true positives. This means that false positives, which are added to the denominator of the precision metric, and false negatives, which are added to the denominator of the sensitivity metric, are equal to zero. In this scenario the F1 Score is equal to 1:

$$F1Score = \frac{2 \cdot \frac{1}{1+0} \cdot \frac{1}{1+0}}{\frac{1}{1+0} + \frac{1}{1+0}} = \frac{2}{2} = 1$$

Now imagine a study which results in either all false positives or all false negatives. In the scenario the F1 Score is equal to 0:

$$F1score = \frac{2 \cdot \frac{0}{0+1} \cdot \frac{0}{0+0}}{\frac{0}{0+1} + \frac{0}{0+0}} = 0$$

Therefore, any study which has at least one true result and one false result will have a F1 score bound by: $0 \leq F1Score \leq 1$.

Question 10

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.

```
ROC <- function(data)
{
  data1 = data
  thresholds <- seq(0,1,0.01)
  Y <- c()
  X <- c()
  for (threshod in thresholds) {
    data1$scored.class <- ifelse(data1$scored.probability > threshod,1,0)
    X <- append(X,1-spec(data1))
    Y <- append(Y,sens(data1))
    }
  df <- data.frame(X=X,Y=Y)
  df <- na.omit(df)
  g <- ggplot(df,aes(X,Y)) + geom_line() + ggtitle('Custom ROC Curve') +
    xlab('Specificity') + ylab('Sensitivity')
  height = (df$Y[-1]+df$Y[-length(df$Y)])/2
  width = -diff(df$X)
  area = round(sum(height*width),4)
  return(list(Plot =g,AUC = area))
}
```

Question 11

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

```
Name <- c('Accuracy','Classification Error Rate', 'Precision', 'Sensitivity','Specificity', 'F1 Score')
Value <- round(c(accuracy(data), ER(data), precision(data), sens(data), spec(data), Fscore(data)),4)
df <- as.data.frame(cbind(Name, Value))
kable(df)
```

Name	Value
Accuracy	0.8066
Classification Error Rate	0.1934
Precision	0.8438
Sensitivity	0.4737
Specificity	0.9597
F1 Score	0.6067

Question 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?

```
confusionMatrix(data$scored.class, data$class, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
```

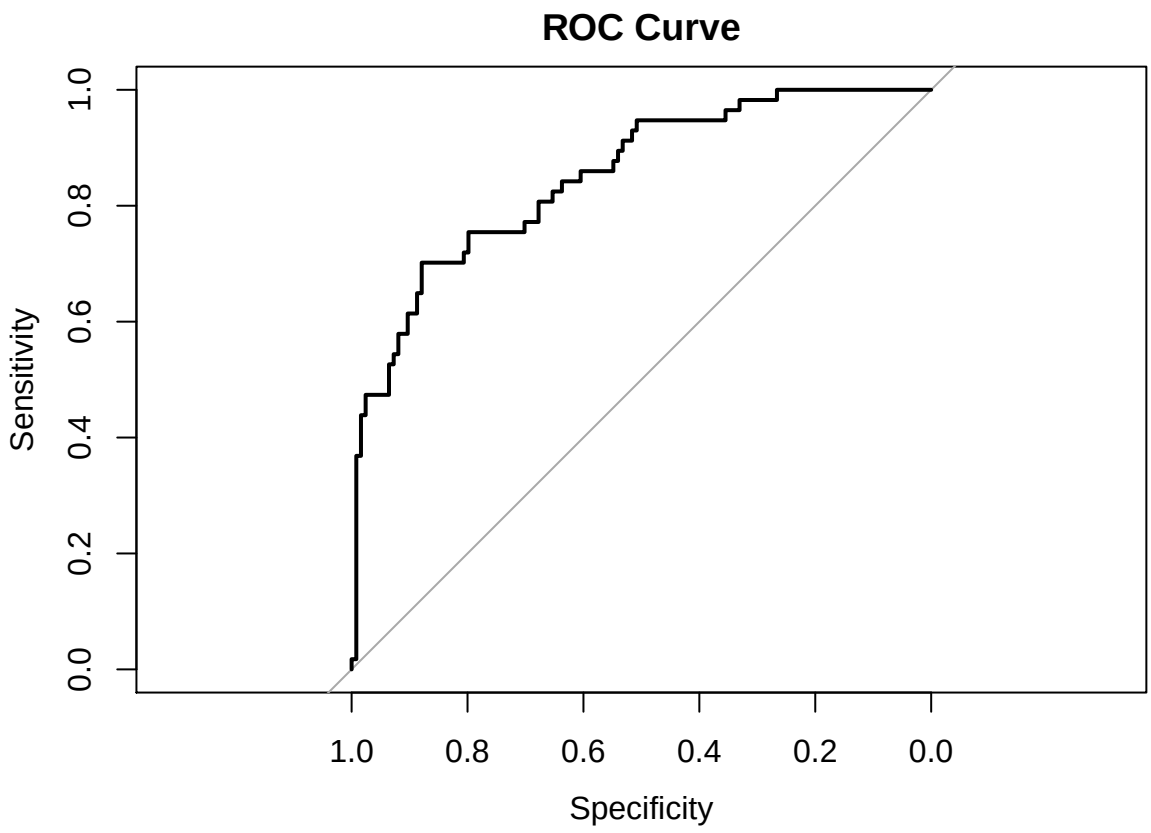
```
## Prediction   0   1
##          0 119  30
##          1   5  27
##
##                Accuracy : 0.8066
##                  95% CI : (0.7415, 0.8615)
##     No Information Rate : 0.6851
##     P-Value [Acc > NIR] : 0.0001712
##
##                   Kappa : 0.4916
##  Mcnemar's Test P-Value : 4.976e-05
##
##             Sensitivity : 0.4737
##             Specificity : 0.9597
##          Pos Pred Value : 0.8438
##          Neg Pred Value : 0.7987
##              Prevalence : 0.3149
##          Detection Rate : 0.1492
##    Detection Prevalence : 0.1768
##       Balanced Accuracy : 0.7167
##
##        'Positive' Class : 1
##
```

We got the same Accuracy, Sensitivity and Specificity.

Question 13

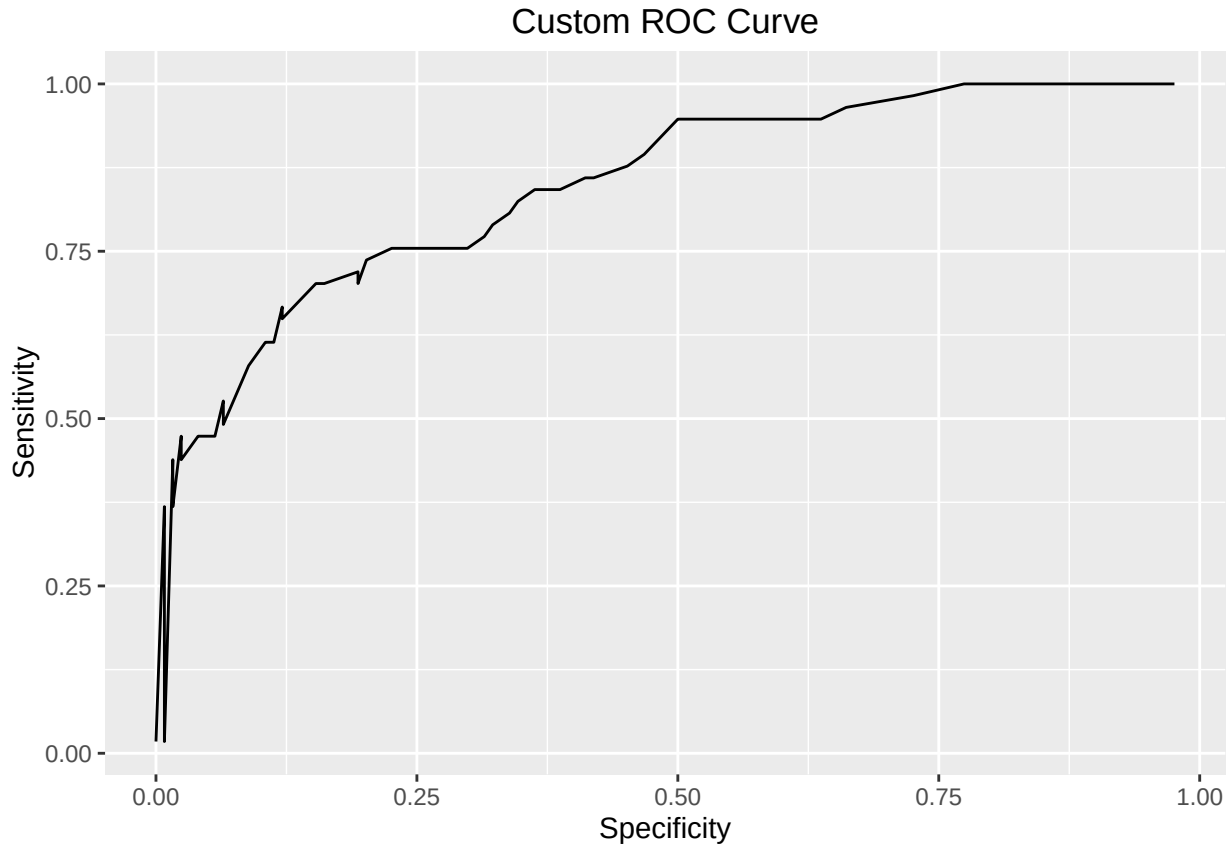
Investigate the pROC package. Use it to generate an ROC curve for the data set. How do the results compare with your own functions?

```
rc <- roc(as.factor(data$class) ~ data$scored.probability)
plot(rc,main='ROC Curve')
```




```
##
## Call:
## roc.formula(formula = as.factor(data$class) ~ data$scored.probability)
##
## Data: data$scored.probability in 124 controls (as.factor(data$class) 0) < 57 cases (as.factor(data$cl
## Area under the curve: 0.8503
```

```
ROCcustom <- ROC(data)
ROCcustom$Plot
```




```
ROCcustom$AUC
```

```
## [1] 0.8247
```

At the end, we got the similar curve and the area under the curve.