DATA-621 Project 2

Harpreet Shoker

23-Jun-2018

Cover page

Abstract

In this assignment we will be discussing various classification metrics. We will be implementing these metrics using R functions and then verify these functions using various libraries like caret and pROC.

The data set classification-output contains 181 rows and 11 columns and we are using three key columns - class,scored.classand scored.probability.

First we have created a raw confusion matrix using table() function where rows represent actual class values of 0 or 1. Columns represent predicted class values of 0 or 1. Then with the help of confusion matrix function we have calculated accuracy,classification error,precision,sensitivity and specificity etc. We have used "caret" package to compare and verify our results. Package caret has several functions that attempt to streamline the model building and evaluation process, as well as feature selection and other techniques.

We tried to plot ROC curve for the dataset by creating own Roc_curve R function . At last we are exploring pROC package.

pROC package provides Tools for visualizing, smoothing and comparing receiver operating characteristic(ROC curves). (Partial) area under the curve (AUC) can be compared with statistical tests. By using functions from pROC package we have compared our RoC plots . Overall we got the same results by using own R functions and using caret and proc functions.

Complete each of the following steps as instructed:

Loading libraries

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```
library(pracma)
```

```
## Warning: package 'pracma' was built under R version 3.4.3
```

Problem 1 - Loading DATA

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

```
# read in csv file provided for assignment
myurl <- "https://raw.githubusercontent.com/Harpreet1984/DATA621/master/classification-output-data.csv"
data <- read.csv(myurl)
head(data)
```

```
##   pregnant glucose diastolic skinfold insulin  bmi pedigree age class
## 1        7     124        70       33     215 25.5    0.161  37     0
## 2        2     122        76       27     200 35.9    0.483  26     0
## 3        3     107        62       13      48 22.9    0.678  23     1
## 4        1      91        64       24       0 29.2    0.192  21     0
## 5        4      83        86       19       0 29.3    0.317  34     0
```

```
## 6        1     100        74       12      46 19.5    0.149  28     0
##   scored.class scored.probability
## 1            0         0.32845226
## 2            0         0.27319044
## 3            0         0.10966039
## 4            0         0.05599835
## 5            0         0.10049072
## 6            0         0.05515460
```

Loading the csv file from my github account and creating a data frame mydata The data set contains 181 rows and 11 columns

Problem 2 - Raw confusion matrix

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?

```
tab<-table(data$class,data$scored.class)
tab
```

```
##
##       0   1
##   0 119   5
##   1  30  27
```

Here rows represent actual class values of 0 or 1. Columns represent predicted class values of 0 or 1. So in the top left corner 119 is the number of observations where the class was correctly predicted to be 0. The top right corner shows 5 observations where the class of 0 was incorrectly predicted as 1. Similarly, we have 30 observations of class 1 incorrectedly predicted as class 0 and 27 observations of class 1 correctly predicted.

Problem 3 - Calculating ACcuracy

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

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

First creating a helper function that calculates the various elements of the confusion matrix. This helper function will be used in further problems of the assignment.

```
confusion_mat <- function(data){
  data.frame(tp=nrow(data[data$class==1 & data$scored.class==1,]),
             tn=nrow(data[data$class==0 & data$scored.class==0,]),
             fp=nrow(data[data$class==0 & data$scored.class==1,]),
             fn=nrow(data[data$class==1 & data$scored.class==0,])
  )
}
```

Calulating Accuracy

```
accuracy<-function(data){
  f <- confusion_mat(data)
  (f$tp+f$tn)/(f$tp+f$fp+f$tn+f$fn)
```

```
}
accuracy(data)
```

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

Problem 4 - Calculating Classification error

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

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

```
classification_error<-function(data){
  f <- confusion_mat(data)
  (f$fp+f$fn)/(f$tp+f$fp+f$tn+f$fn)
}
classification_error(data)
```

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

verifying sum of accuracy and classification error is 1

```
sum <- classification_error(data)+accuracy(data)
sum
```

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

Problem 5 - Calculating Precision

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(data){
  f <- confusion_mat(data)
  (f$tp)/(f$tp+f$fp)
}
precision(data)
```

```
## [1] 0.84375
```

Problem 6 - Calculating sensivity

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

```
sensitivity<-function(data){
  f <- confusion_mat(data)
  (f$tp)/(f$tp+f$fn)
}
sensitivity(data)
```

```
## [1] 0.4736842
```

Problem 7 - Calculating Specificity

Write a function that takes the data set as a dataframe, with actual and predicted classifications identified, and returns the specificity of the predictions. $Specificity = \frac{TN}{TN+FP}$

```
specificity<-function(data){
  f <- confusion_mat(data)
  (f$tn)/(f$tn+f$fp)
}
specificity(data)
```

```
## [1] 0.9596774
```

Problem 8 - F1 Score

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 $F1score = \frac{2*precision*sensitivity}{precision+sennsitivity}$

```
f1_score<-function(data){
  p<- precision(data)
  s<- sensitivity(data)
  2*p*s/(p+s)
}
f1_score(data)
```

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

Problem 9

efore 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.

```
# assume p is prescision and s is sensitivity.
p <- runif(100, min = 0, max = 1)
s <- runif(100, min = 0, max = 1)
f <- (2*p*s)/(p+s)
summary(f)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01929 0.20482 0.33662 0.39738 0.59282 0.93077
```

Ran simulation of 100 to prove that F1 is always between 0 and 1

Another way

We see from above results Both *Precision* and *Sensitivity* have a range from 0 to 1. Consider that if $a > 0$ and $0 < b < 1$, then $ab < a$ (a fraction of any positive number will be smaller than the original number).

Then $P \times S < P$ and $P \times S < S$.

Then $P \times S + P \times S < P + S$, or

$2 \times P \times S < P + S$.

The fraction of these two values will be lower than 1. Also, since both values are positive, $F1$ *score* will be positive. If P is zero, then S is zero and $F1$ *Score* is not defined. If P is one and S is one, then $F1$ *Score* is one.

So we have $0 < F$ *Score* ≤ 1.

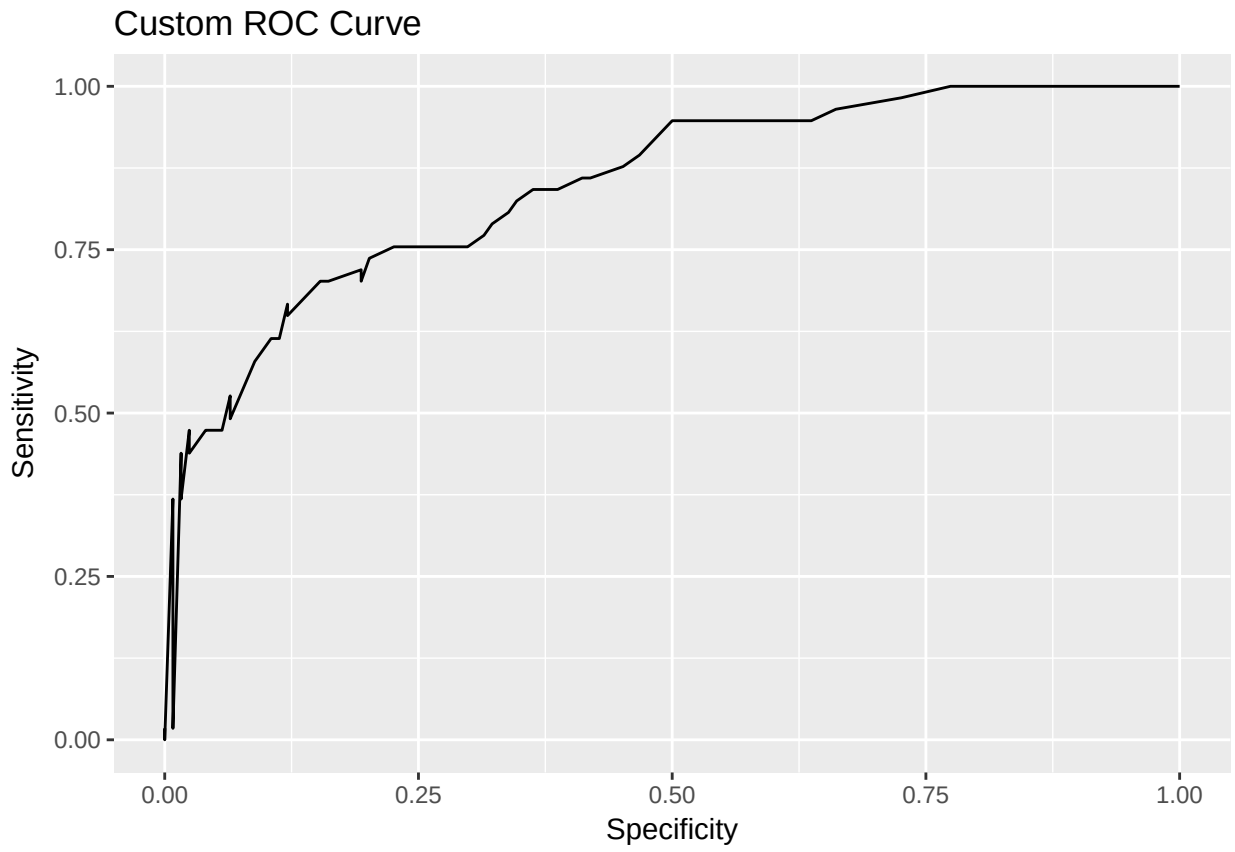
Problem 10 - ROC curve

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.

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

```
Roc_curve(data)
```

```
## $Plot
```




```
##
## $AUC
## [1] 0.8489
```

Problem 11.

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

```
Names <- c('Accuracy','Classification Error Rate', 'Precision', 'Sensitivity','Specificity', 'F1 Score'
Value <- round(c(accuracy(data), classification_error(data), precision(data), sensitivity(data), specif
new <- as.data.frame(cbind(Names, Value))
head(new)
```

```
##                        Names  Value
## 1                   Accuracy 0.8066
## 2 Classification Error Rate 0.1934
## 3                  Precision 0.8438
## 4                Sensitivity 0.4737
## 5                Specificity 0.9597
## 6                   F1 Score 0.6067
```

Problem 12 - Package caret

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?

```
#install.packages("caret")
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.4
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following objects are masked _by_ '.GlobalEnv':
##
##     precision, sensitivity, specificity
```

```
caret_data <- table(data$class,data$scored.class)
confusionMatrix(caret_data,reference = data$class)
```

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

Problem 13 - Package pROC

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

```
#install.packages("pROC")
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 3.4.4
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
```

```
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
pROC_data <- roc(data$class,data$scored.probability)
plot(pROC_data, main = "pROC")
```