Data 621

Cesar Espitia HW #2

6/24/2018

Overview

In this homework assignment, you will work through various classification metrics. You will be asked to create functions in R to carry out the various calculations. You will also investigate some functions in packages that will let you obtain the equivalent results. Finally, you will create graphical output that also can be used to evaluate the output of classification models, such as binary logistic regression.

Question 1

These are setup steps.

```
knitr::opts_chunk$set(echo = TRUE)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(knitr)
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

Question 2

Show the confusion matrix.

```
# read data
origdata = read.csv(file="data/classification-output-data.csv")
data = subset(origdata, select = c(scored.class, class))

CM = table(data$scored.class, data$class)
print(CM)
```

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

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.

```
TP = CM[1,1]
TN = CM[2,2]
FP = CM[2,1]
FN = CM[1,2]

con.accuracy <- function(x) {
  CM = table(x$scored.class, x$class)

  (TP + TN)/sum(CM)
}

con.accuracy(data)
```

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

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.

```
con.CER <- function(x) {
  CM = table(x$class, x$scored.class)

  (FP + FN)/sum(CM)
}

con.CER(data)
```

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

```
con.accuracy(data) + con.CER(data) == 1
```

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

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.

```
con.prec <- function(x) {
  CM = table(x$class, x$scored.class)

  TP / (TP + FP)
}

con.prec(data)
```

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

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.

```
con.sens <- function(x, threshold = 0.5) {
  TP <- sum(x[, 1] > threshold & x[, 2] == 1)
  FN <- sum(x[, 1] <= threshold & x[, 2] == 1)
  TP/(TP + FN)
}

con.sens(data)
```

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

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.

```
con.spec <- function(x, threshold = 0.5) {
  TN <- sum(x[, 1] <= threshold & x[, 2] == 0)
  FP <- sum(x[, 1] > threshold & x[, 2] == 0)
  TN/(TN + FP)
}

con.spec(data)
```

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

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.

```
con.F1 <- function(x, threshold = 0.5) {
  CM = table(x$class, x$scored.class)

  (2*con.prec(x)*con.sens(x)) / (con.prec(x) + con.sens(x))
}

con.F1(data)
```

```
## [1] 0.6342908
```

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 $ab < a$.)

Answer

The bounds can be done with simple logic. If the class and scored class are perfect (equal to each other) then FP and FN would be 0 and Prec and Sens each would be 1. This means that the max would be $(2 * 1 * 1)/2 = 1$

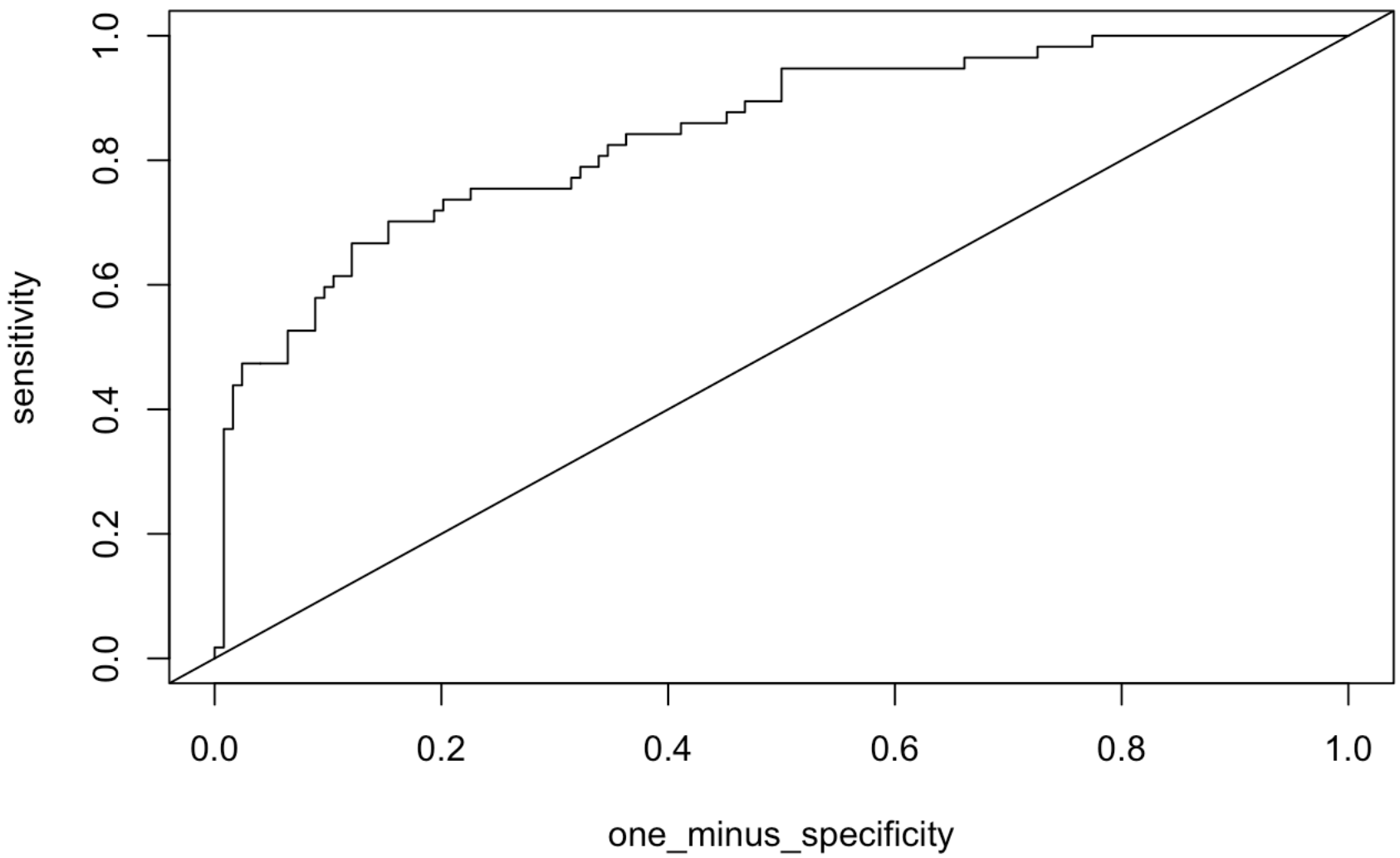
If the model doesn't predict anything meaning that TP reaches 0 then the min is 0.

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.

The data uses the functions for specificity and sensitivity prebuilt in the prior sections. These two functions are the only ones where the thereshold is important to calculate these values, the others are based upon the confusion matrix.

```
data_SPC <- subset(origdata, select = c(scored.probability, class))

myROC <- function(df, intervals = 0.01){
  thresholds <- seq(0, 1, by = intervals)

  sensitivity <- sort(sapply(thresholds, function(x) con.sens(df, threshold = x)))
  one_minus_specificity <- sort(1 - sapply(thresholds, function(x) con.spec(df, thres
hold = x)))

  #create plot
  plot(sensitivity ~ one_minus_specificity, type = "s", xlim=c(0, 1), ylim=c(0, 1), m
ain = "My Function")
  abline(a = 0, b = 1)



  sum(diff(one_minus_specificity) * rollmean(sensitivity, 2))
}

myROC(data_SPC)
```




```
## [1] 0.8488964
```

Question 11

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

```
con.accuracy(data)
```

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

```
con.CER(data)
```

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

```
con.prec(data)
```

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

```
con.sens(data)
```

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

```
con.spec(data)
```

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

```
con.F1(data)
```

```
## [1] 0.6342908
```

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?

```
library(caret)
```

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

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

```
## Loading required package: ggplot2
```

```
confusionMatrix(CM)
```

```
## Confusion Matrix and Statistics
##
##
##       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 : 0.9597
##             Specificity : 0.4737
##          Pos Pred Value : 0.7987
##          Neg Pred Value : 0.8438
##              Prevalence : 0.6851
##          Detection Rate : 0.6575
##    Detection Prevalence : 0.8232
##       Balanced Accuracy : 0.7167
##
##        'Positive' Class : 0
##
```

```
con.sens(data)
```

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

```
con.spec(data)
```

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

The data matches together.

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?

The AUC under pROC is 0.8212 and the calculated one is 0.84 which is close. Obviously the curves are more clean in the pROC package as they aren't built using the stepwise intervals to calculated the data for the curve.

```
library(pROC)
```

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

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

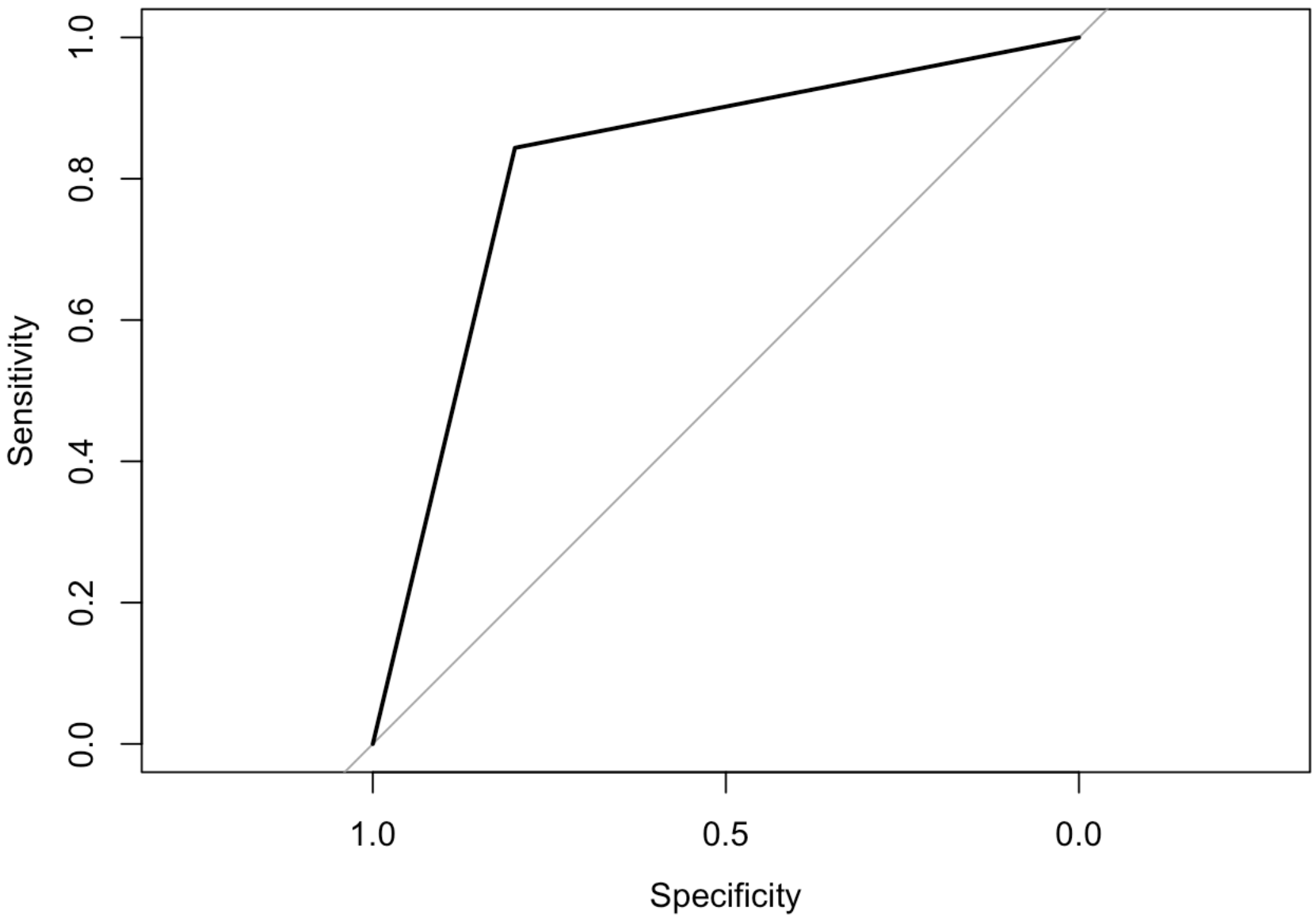
```
##
## Attaching package: 'pROC'
```

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

```
roc(origdata$scored.class, origdata$class)
```

```
##
## Call:
## roc.default(response = origdata$scored.class, predictor = origdata$class)
##
## Data: origdata$class in 149 controls (origdata$scored.class 0) < 32 cases (origdat
a$scored.class 1).
## Area under the curve: 0.8212
```

```
plot.roc(origdata$scored.class, origdata$class)
```


Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.0
0.5
0.0
Specificity