Data 621 - Homework 2

Anthony Munoz

3/15/2020

Contents

** 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.

Data.

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

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

pregnant	glucose	diastolic	skinfold	insulin	bmi	pedigree	age	class	scored.class	scored.probability
7	124	70	33	215	25.5	0.161	37	0	0	0.3284523
2	122	76	27	200	35.9	0.483	26	0	0	0.2731904
3	107	62	13	48	22.9	0.678	23	1	0	0.1096604
1	91	64	24	0	29.2	0.192	21	0	0	0.0559984
4	83	86	19	0	29.3	0.317	34	0	0	0.1004907
1	100	74	12	46	19.5	0.149	28	0	0	0.0551546

2. Confusion Matrix

The data set has three key columns we will use: 1. class: the actual class for the observation 2. scored.class: the predicted class for the observation (based on a threshold of 0.5) 3. 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?

```r
  table(data$scored.class,data$class)
```

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

3. 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 <- function(df){


  return((sum(df$class == 1 & df$scored.class  == 1) + sum(df$class == 0 & df$scored.class  == 0)) / nr
   )
}

accuracy(data)
```

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

4. Classification Error rate

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.

```
classification.error.rate <- function(df){


  return((sum(df$class == 1 & df$scored.class  == 0) + sum(df$class == 0 & df$scored.class  == 1)) / nr

}

classification.error.rate(data)
```

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

5. 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 <- function(df){


  return((sum(df$class == 1 & df$scored.class  == 1)/ (sum(df$class == 1 & df$scored.class  == 1) + sum

}

precision(data)
```

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

6. Sensitivity

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 <- function(df){


  return((sum(df$class == 1 & df$scored.class  == 1)/ (sum(df$class == 1 & df$scored.class  == 1) + sum

}


sensitivity(data)
```

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

7. 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 <- function(df){


  return((sum(df$class == 0 & df$scored.class  == 0)/ (sum(df$class == 0 & df$scored.class  == 0) + sum

}



specificity(data)
```

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

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 <- function(df){


  return((2 * precision(df) * sensitivity(df) / (precision(df) + sensitivity(df))))
}



f1score(data)
```

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

9. F! Bounds Score

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.

Answer

we can assume that sensitivity and precision will always been between 0 and 1 by then if we place them on the F1 score formula we can noticed tat the result will be between 0 and 1. by using the hint we can say that a and b are precision and sensitivity and its true that they are between 0 and 1.

10. ROC Curve

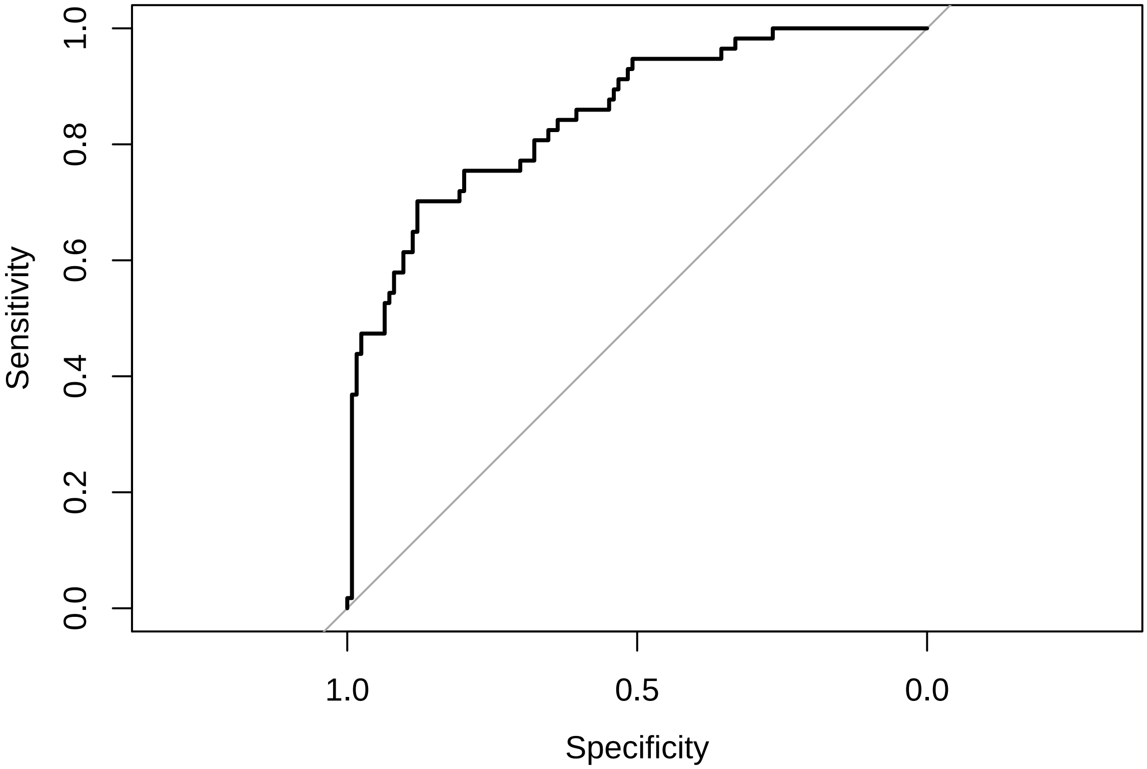
Write a function that generates an ROC curve from a data set with a true classification column (class in our example) and a probability column (scored.probability in our example). Your function should return a list that includes the plot of the ROC curve and a vector that contains the calculated area 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(df){
 plot(roc(df$class,df$scored.probability))
}

ROC.Curve(data)
```

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

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



11. Metrics Functions

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

```
all.results <- function(df){
  df.kable <- c(paste('Accuracy - ',accuracy(df)),paste('Classification Error rate -',classification.er

  kable(df.kable, col.names = "Results Metrics")
}

all.results(data)
```

Results Metrics
Accuracy - 0.806629834254144
Classification Error rate - 0.193370165745856
Precision - 0.84375
Sensitivity - 0.473684210526316
Specificity - 0.959677419354839
F1 Score - 0.606741573033708

12. Caret Confusion Matrix

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(table(data$class,data$scored.class) , 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
##
```

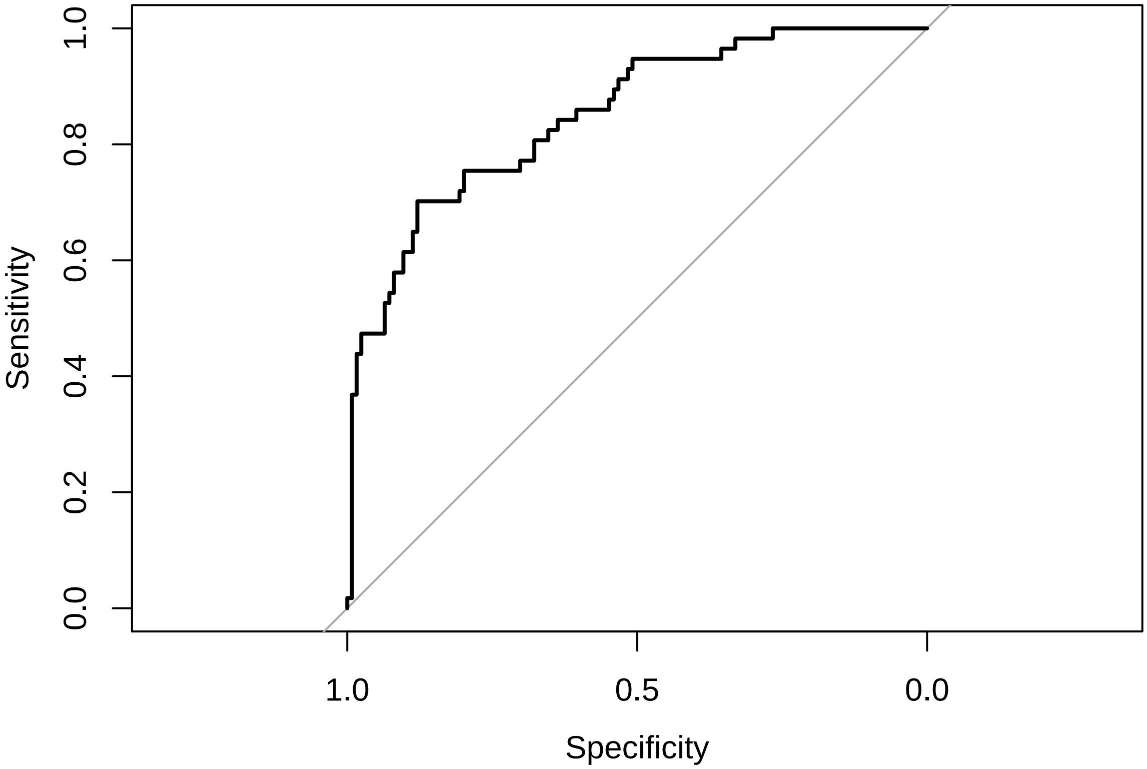
13. pROC Curve

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

```
plot(roc(data$class,data$scored.probability))
```

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

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



### Appendix

Rcode: Github

GitHub PDF: Github

PDF: nbviewer