Homework 2

Gurpreet Singh

March 12, 2018

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

   The classification dataset consists of 181 rows and 11 variables. For our analysis, we will be focusing three key columns

   - `Class`
   - `Scored Class`
   - `Scored Probability`

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. 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?

   The confusion matrix for the data is shown below. In the table rows represents actual class and columns represents predicted class. 57 observations represents actual true values (57 patients were actually suffering from disease).

   - True Positive – 27
   - True Negative – 119

- False Positive – 5
- False Negative - 30

	Predicted	
Actual	FALSE	TRUE
FALSE	119	5
TRUE	30	27

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

   Accuracy for the data is 0.866298

```
acc <- function(df_class){
        mat<- as.data.frame(table(df_class$class, df_class$scored.class))
        tn <- mat$Freq[1]
        tp <- mat$Freq[4]
        return((tn+tp)/sum(mat$Freq))
}
```

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.

   Error – 0.1933702

```
err <- function(df_class){
        mat<- as.data.frame(table(df_class$class, df_class$scored.class))
        fp <- mat$Freq[2]
        fn <- mat$Freq[3]
        return((fp+fn)/sum(mat$Freq))
}
```

   Verify that you get an accuracy and an error rate that sums to one.

```
acc(df_class)+err(df_class)

  1
```

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 - 0.84375

```
prec <- function(df_class){
        mat<- as.data.frame(table(df_class$class, df_class$scored.class))
```

```
        tp <- mat$Freq[4]
        fp <- mat$Freq[2]
   return((tp)/(tp+fp))
}
```

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 - 0.9596774

```
sens <- function(df_class){
        mat<- as.data.frame(table(df_class$class, df_class$scored.class))
        tp <- mat$Freq[4]
        fn <- mat$Freq[3]
   return((tp)/(tp+fn))
}
```

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 - 0.4736842

```
spec <- function(df_class){
        mat<- as.data.frame(table(df_class$class, df_class$scored.class))
        tn <- mat$Freq[1]
        fp <- mat$Freq[2]
        return((tn)/(tn+fp))
}
```

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.

F1 score - 0.6067416

```
f1_sc <- function(df_class){
   mat<- as.data.frame(table(df_class$class, df_class$scored.class))
        tn <- mat$Freq[1]
        tp <- mat$Freq[4]
        fn <- mat$Freq[3]
        fp <- mat$Freq[2]
        pr <- tp/(tp+fp)
        se <- tp/(tp+fn)
        return(2*pr*se/(pr+se))
}
```

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.

F1 score is calculated based on precision and sensitivity. Bounds for Precision and Sensitivity are between 0 and 1. For any values of precision and sensitivity between their bounds, F1 score will fall in the range of 0 and 1.

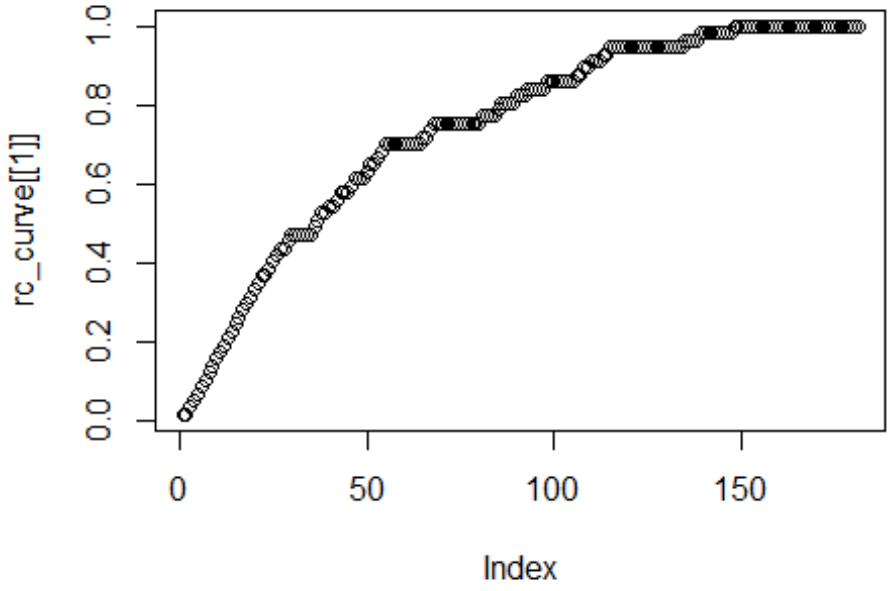
10. Write a function that generates an ROC curve from a data set with a true classification column (class in our example) and a probability column (scored probability in our example). Your function should return a list that includes the plot of the ROC curve and a vector that contains the calculated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

```
http://blog.revolutionanalytics.com/2016/08/roc-curves-in-two-lines-of-code.h
tml

simple_roc <- function(labels, scores){
  labels <- labels[order(scores, decreasing=TRUE)]
  TPR=cumsum(labels)/sum(labels)
  FPR=cumsum(!labels)/sum(!labels)
  df<- data.frame(TPR,FPR)
  dFPR <- c(diff(FPR), 0)
  dTPR <- c(diff(TPR), 0)
  auc <-sum(TPR * dFPR) + sum(dTPR * dFPR)/2
  return(c(df,auc))
}

rc_curve <- simple_roc(df_class$class,df_class$scored.probability)

plot(rc_curve[[1]])
```




```
auc <- rc_curve[[2]]
```

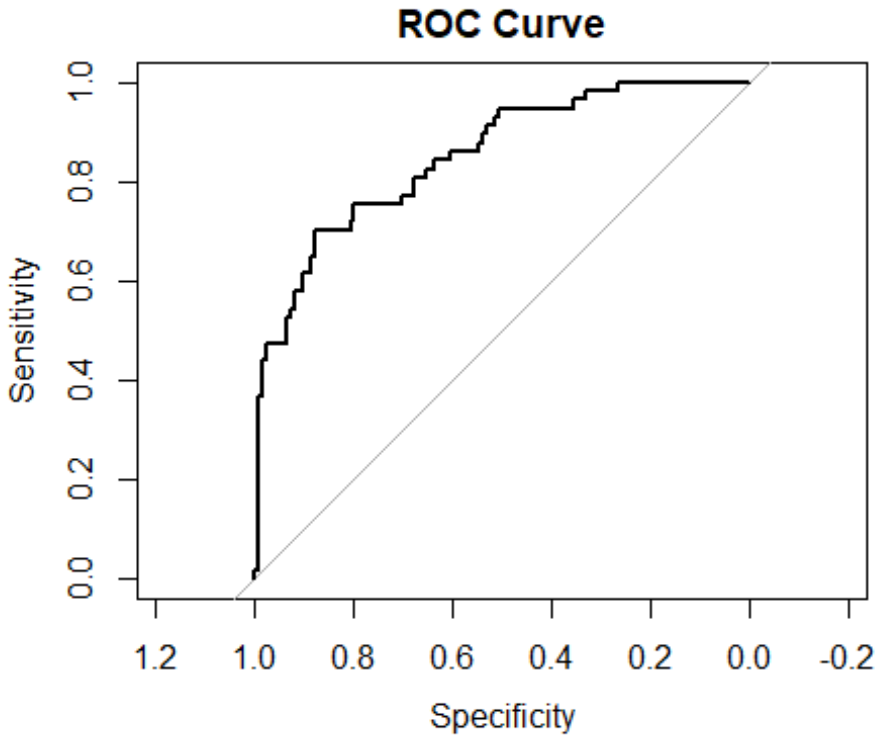
11. Use your created R functions and the provided classification output data set to produce all of the classification metrics discussed above.

Comparison table is created for the classification output data between the functions created and in built functions in the caret package. The results from both were similar.

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

	Manual Functions	Caret Package
Accuracy	0.866298	0.866298
Error	0.1933702	0.1933702
Precision	0.84375	0.7986577
Specificity	0.4736842	0.4736842
Sensitivity	0.9596774	0.9596774
F1 score	0.6067416	0.6067416

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?



Results were similar from the function created for roc curve and from pROC package.

R Code:

https://github.com/gpsingh12/Data-621/blob/master/Hw2/Singh_hw2.Rmd

References:

http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

https://stackoverflow.com/questions/24348973/how-to-retrive-overall-accuracy-value-from-confusionmatrix-in-r

http://blog.revolutionanalytics.com/2016/08/roc-curves-in-two-lines-of-code.html