Home Work Assignment - 02

Critical Thinking Group 5
Arindam Barman
Mohamed Elmoudni
Shazia Khan
Kishore Prasad

Contents

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

```
class_data <- read.csv("https://raw.githubusercontent.com/kishkp/data621-ctg5/master/HW2/classification-

#class_data <- read.csv("C:/CUNY/Courses/IS-621/Assignments621/Assignment02/classification-output-data.


#summary(class_data)
```

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?

```
##        Predicted  NO YES
## Actual
## NO               119   5
## YES               30  27
```

```
# here is another function...
#kable(CrossTable(cm, prop.t=TRUE, prop.r=TRUE, prop.c=TRUE))

cm2<- (CrossTable(cm, prop.t=FALSE, prop.r=FALSE, prop.c=FALSE))
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |-------------------------|
##
##
## Total Observations in Table:  181
##
##
##              | Predicted
##       Actual |        NO |       YES | Row Total |
## -------------|-----------|-----------|-----------|
##           NO |       119 |         5 |       124 |
##              |     2.805 |    13.063 |           |
## -------------|-----------|-----------|-----------|
##          YES |        30 |        27 |        57 |
##              |     6.103 |    28.418 |           |
## -------------|-----------|-----------|-----------|
## Column Total |       149 |        32 |       181 |
## -------------|-----------|-----------|-----------|
##
##
```

```
cm2$t
```

```
##       Predicted
## Actual  NO YES
##    NO  119   5
##    YES  30  27
```

Explanation:
There are two possible predicted classes: "yes" and "no". The classifier made a total of 181 predictions (e.g., 181 were being tested for the presence of that disease, in this diabetes). Out of those 181 cases, the classifier predicted "yes" 32 times, and "no" 149 times. In reality, 57 patients in the sample have the disease, and 124 patients do not.

In addition:
• 119 represents true negatives (TN) where we predicted no, and they don't have the disease.
• 27 represents true positives (TP): These are cases in which we predicted yes (they have the disease), and they do have the disease.
• 30 represents false negatives (FN): We predicted no, but they actually do have the disease.
• 5 represents false positives (FP): We predicted yes, but they don't actually have the disease.

Since we have given the table command as table(class_data$class, class_data$scored.class), in the output above: - Rows represent the Actual class - Columns represent predicted class

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 <- function(ds, actual, predicted) {

  # check if object exists
  if (!(exists("ds")))   return("Object not found")

  #check if object is data frame
  if (!(is.data.frame(ds))) return("Invalid datatype: Not a data frame")

  #check if data frameis empty
  if (is.data.frame(ds) && nrow(ds)==0)
    return("dataset is Empty")
  else
    #   table(ds[actual], ds[predicted])
    cm<- with(class_data, table("Actual"=class, "Predicted"= scored.class))
    #cm <- table(select(class_data, get(actual), get(predicted)))

    kable(cm)
    TP <- cm[1,1]
    TN <- cm[2,2]
    FN <- cm[1,2]
    FP <- cm[2,1]
    return ((TP + TN) / (TP + FP + TN + FN))
  }
```

The accuracy is: 0.8066298

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.

```
class_err_rate <- function(ds, actual, predicted) {

#   table(ds[actual], ds[predicted])
    cm <- table(select(class_data, get(actual), get(predicted)))

    kable(cm)
    TP <- cm[1,1]
    TN <- cm[2,2]
    FN <- cm[1,2]
    FP <- cm[2,1]
    return ((FP + FN) / (TP + FP + TN + FN))
}

class_err_rate(class_data, "class", "scored.class")
```

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

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

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. ??= ????????????????+???????? 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. ??= ????????????????+???????? 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. ????1 ????????????????????= 2×????????????????????????????????????×??
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 < ???? < 1 and 0 < ???? < 1 then ???????? < ????.) 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. 11. Use your created R functions and the provided classification output data set to produce all of the classification metrics discussed above.

```
accuracy(class_data, "class", "scored.class")
```

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

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?
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?