CUNY DATA 621 - Business Analytics and Data Mining

Homework 2 - Classification Metrics

Walt Wells, 2018

1 - Load Data / Environment Prep

```
if (!require('ggplot2')) (install.packages('ggplot2'))
if (!require('caret')) (install.packages('caret'))
if (!require('pROC')) (install.packages('pROC'))

theme_update(plot.title = element_text(hjust = 0.5),
             axis.text.x = element_text(angle = 90, hjust = 1))

data <- read.csv('Data/classification-output-data.csv', header=T)
data <- data[,c(9:11)]
```

2

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?

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

knitr::kable(t)
```

	0	1
0	119	30
1	5	27

In this instance the rows represent the predicted class, while the columns represent the actual class.

3 - 8, 11

Write a function that takes the data set as a dataframe, with actual and predicted classifications identified, and returns the accuracy, error_rate, precision, sensitivity, specificity, and f1 score of the predictions. Verify that you get an accuracy and an error rate that sums to one. Use your created R functions and the provided classification output data set to produce all of the classification metrics discussed above.

```
BinaryClassHelper <- function(t) {
    #given table, get TP, FP, FN, TN, then summary stats
    tp <- t[1,1]; fp <- t[1,2]; fn <- t[2,1]; tn <- t[2,2]

    accuracy <- (tp + tn) / (tp + fp + tn + fn)
    error_rate <- (fp + fn) / (tp + fp + tn + fn)
    precision <- tp / (tp + fp)
    sensitivity <- tp / (tp + fn)
```

```
    specificity <- tn / (tn + fp)
    f1 <- (2 * precision * sensitivity) / (precision     + sensitivity)
    df <- data.frame(accuracy = accuracy,
                     error_rate = error_rate,
                     precision = precision,
                     sensitivity = sensitivity,
                     specificity = specificity,
                     f1 = f1)
    return(df)
}

results <- BinaryClassHelper(t)
print(results$accuracy + results$error_rate)
```

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

```
knitr::kable(results)
```

accuracy	error_rate	precision	sensitivity	specificity	f1
0.8066298	0.1933702	0.7986577	0.9596774	0.4736842	0.8717949

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 $ab < a$).

Since F1 is calculated using the precision and sensitivity scores, and each of those are bounded by 0 and 1, we can be confident that 2 * precision * sensitivity / (precision + sensitivity) will also be between $0 > 1$.

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.

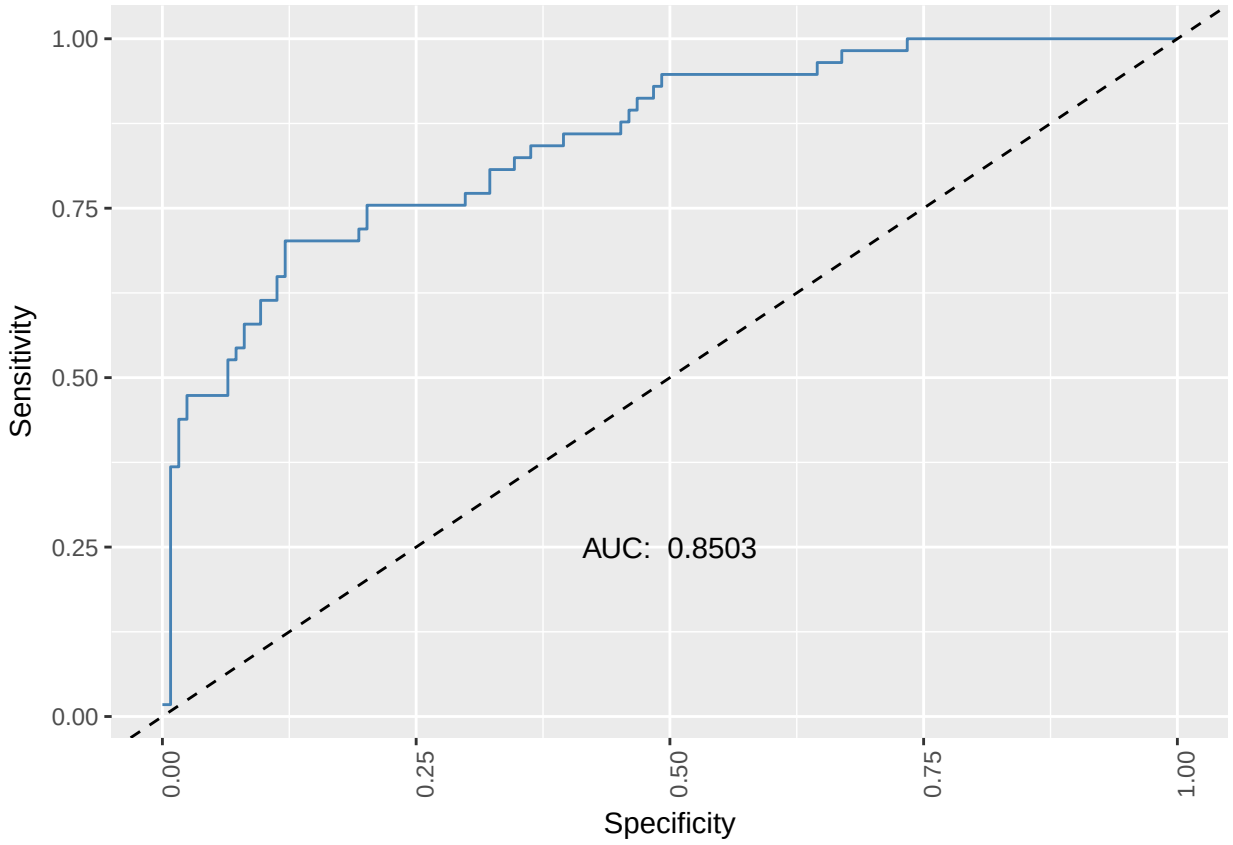
```
myROCandAUC <- function(class, scores){
    # function to calculate ROC
    # reference for ROC: http://blog.revolutionanalytics.com/2016/08/roc-curves-in-two-lines-of-code.ht
    # reference for AUC: http://blog.revolutionanalytics.com/2016/11/calculating-auc.html
    class <- class[order(scores, decreasing=TRUE)]
    Sensitivity <- cumsum(class)/sum(class)
    Specificity <- cumsum(!class)/sum(!class)
    df <- data.frame(Sensitivity,
                     Specificity,
                     class)

    dSpecificity <- c(diff(Specificity), 0)
    dSensitivity <- c(diff(Sensitivity), 0)
    AUC <- round(sum(Sensitivity * dSpecificity) + sum(dSensitivity * dSpecificity) / 2, 4)
```

```
    results <- list(df, AUC)
    return(results)
}

ROCandAUCResults <- myROCandAUC(data$class, data$scored.probability)
rocResults <- ROCandAUCResults[[1]]
AUC <- ROCandAUCResults[[2]]

ggplot(rocResults, aes(Specificity, Sensitivity)) +
    geom_line(color='steelblue') +
    geom_abline(linetype=2) +
    annotate("text", x=.5, y = .25, label=paste("AUC: ", AUC))
```



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?

```
cM <- confusionMatrix(data$scored.class, data$class)
caretR <- data.frame(t(cM$byClass))
caretResults <- data.frame(accuracy = cM$overall[['Accuracy']],
                           error_rate = 1 - cM$overall[['Accuracy']],
                           precision = caretR$Precision,
                           sensitivity = caretR$Sensitivity,
                           specificity = caretR$Specificity,
```

```
                          f1 = caretR$F1)

compareResults <- rbind(results, caretResults)
row.names(compareResults) <- c("HandCalc", "CaretCalc")
knitr::kable(compareResults)
```

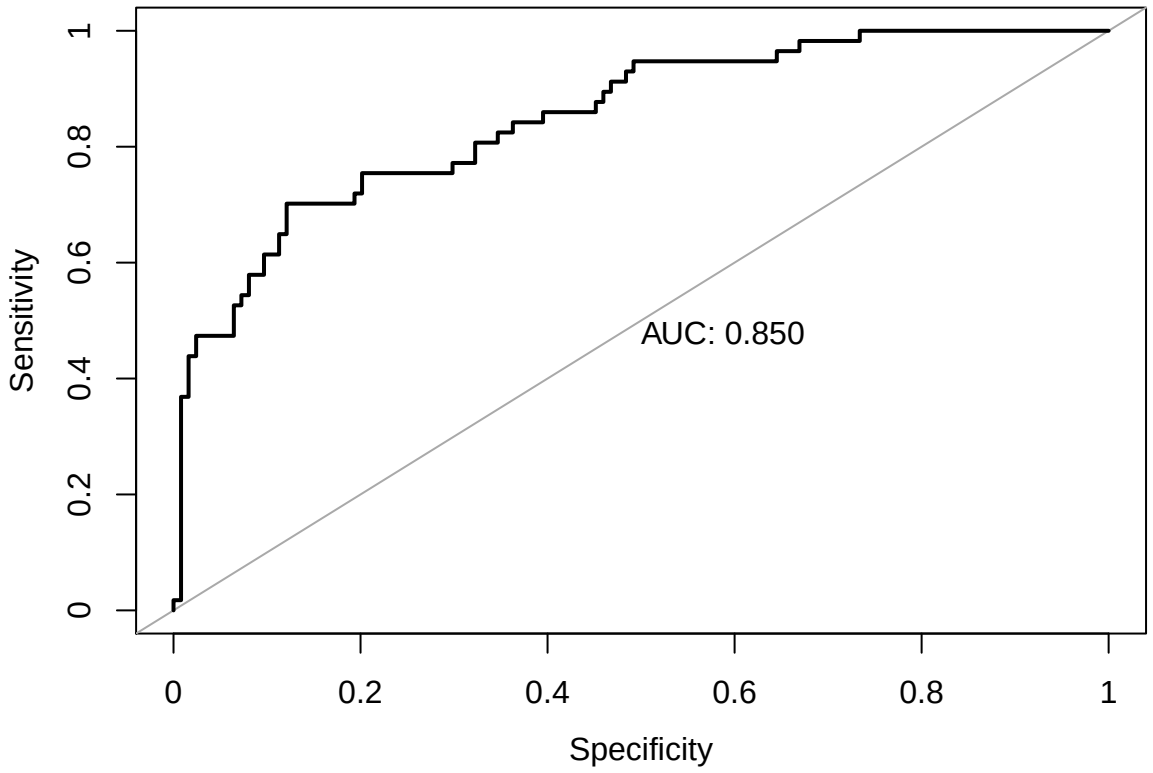
	accuracy	error_rate	precision	sensitivity	specificity	f1
HandCalc	0.8066298	0.1933702	0.7986577	0.9596774	0.4736842	0.8717949
CaretCalc	0.8066298	0.1933702	0.7986577	0.9596774	0.4736842	0.8717949

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?

```
rocobj <- roc(data$class, data$scored.probability)

plot(rocobj, asp=NA, legacy.axes = TRUE, print.auc=TRUE, xlab="Specificity")
```



Appendix

- For full output code visit: https://github.com/wwells/CUNY_DATA_621/blob/master/HW/HW2/HW2_WWells.Rmd