Data 621 - Homework 2

Group 4 Layla Quinones, Ian Costello, Dmitriy Burtsev & Esteban Aramayo

OCtober 10, 2021

```
library(kableExtra)
library(tidyverse)
library(tidymodels)
library(caret)
```

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

```
HW2_url = "https://raw.githubusercontent.com/MsQCompSci/Data621Group4/main/HW2/classification-output-da
df <- read.csv(HW2_url)
kable(head(df)) %>%
  kable_styling()
```

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 rows represents the *actual* class of for the observation. Given the medical statistics in the data, let's say this relates to the presence (or absence) of a medical condition. In this example, zero represents the absence of the condition, one represents the presence. The columns are the scored class, meaning based on the 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

	0	1
0	119	5
1	30	27

provided, the prediction of whether (to continue with our example) a condition *may be* present. Zero again representing absence, and one representing presence.

Now, where the actual and scored classes differ are the errors. Errors can be false positives (type I) or false negatives (type II). In table 1 below, the 30 value are the false negatives, where the predictor scored zero, but the actual class is one. And the 5 value is the false positive, where the predictor scored one, but the actual class is zero.

```
kable(table(df$class, df$scored.class)) %>%
  kable_styling()
```

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.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

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.

$$Classification\ Error\ Rate = \frac{TP + TN}{TP + FP + TN + FN}$$

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.

$$Precision = \frac{TP}{TP + FP}$$

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.

$$Sensitivity = \frac{TP}{TP + FN}$$

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.

$$Specificity = \frac{TP}{TP + FN}$$

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.

$$F1\ Score = \frac{2 * Precision * Sensitivity}{Precision + Sensitivity}$$

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* $ab < a$.*)*

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.

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?

[IC Added: I really liked this overview of the `caret` package and its capabilities. https://www.machinelearningplus.com/machine-learning/caret-package/]

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?