DATA 621 Assignment 2

Kai Lukowiak

2018-03-14

Import Data

```r
library(tidyverse)
library(knitr)
df <- read_csv('~/DATA621/Assignments/Assignment2/classification-output-data.csv')
sample_n(df, size = 5) %>% kable()
```

pregnant	glucose	diastolic	skinfold	insulin	bmi	pedigree	age	class	scored.class	scored.probability
1	109	38	18	120	23.1	0.407	26	0	0	0.0837734
0	107	62	30	74	36.6	0.757	25	1	0	0.1688625
8	188	78	0	0	47.9	0.137	43	1	1	0.8882766
5	139	64	35	140	28.6	0.411	26	0	0	0.3581843
2	90	70	17	0	27.3	0.085	22	0	0	0.0532099

Confusion Matrix

R's table function can be used to create a confusion matrix. For an more indepth explenation of this please see this excelent website.

```r
x <- table(df$scored.class, df$class)
colnames(x) <- c('Actual Positive', 'Actual Negative')
rownames(x) <- c("Predicted Positive", 'Predicted Negative')
x %>% kable()
```

	Actual Positive	Actual Negative
Predicted Positive	119	30
Predicted Negative	5	27

The sum of the rows and columns can give insight into model performance. The rows represent the predicted values while the columns represent the actual values.

```r
# Lable each row for easier computation:
a <- x[1, 1]; b <- x[1, 2]; c <- x[2, 1]; d <- x[2, 2]

Sensitivity <- a / (a + c)
Specificity <- d / (b + d)
PosPredVal <- a / (a + b)
NegPredVal <- d / (c + d)
Accuracy <- (a + d) / (a + b + c +d)
```

- The overall accuracy of the model (0.8066298) shows the total correct classification over all scores. This can be misleading because if 90% of the data is `positive`, a clasifier which only predicts `positive` will be 90% accurate.

- Sensitivity (0.9596774) of the model is the the ratio of predicted positives to total positives.
- Specificity (0.4736842) of the model is the accuracy of negative classification (the opposite of sensitivity).
- Positive and Negative predicted values (0.7986577 and 0.84375 respectively) are positive and negative values that were correctly specified.

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.

```r
accFunc <- function(df, actual, predicted, metric){
  confMat <- table(df[[actual]], df[[predicted]])
  if (metric == 'accuracy'){
    accuracy <- (confMat[1, 1] + confMat[2, 2]) / sum(confMat)
  return(accuracy)
  }

}

accFunc(df, 9, 10, 'accuracy')
```

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

Classification Error Rate

```r

accFunc <- function(df, actual, predicted, metric){
  confMat <- table(df[[actual]], df[[predicted]])
  if (metric == 'accuracy'){
    accuracy <- (confMat[1, 1] + confMat[2, 2]) / sum(confMat)
  return(accuracy)
  } else if (metric == 'classError'){
    classError <- (confMat[1, 2] + confMat[2, 1]) / sum(confMat)
    return(classError)
  }
}

accFunc(df, 9, 10, 'classError')
```

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

To verify that these sum to one:

```r
accFunc(df, 9, 10, 'classError') + accFunc(df, 9, 10, 'accuracy')
```

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

This test is passed.