HW2_data621

Dominika Markowska-Desvallons

10/9/2021

DATA 621

Homework 2 Shana Green and Dominika Markowska-Desvallons 10/10/2021

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

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

```
df <- read.csv("https://raw.githubusercontent.com/hrensimin05/Data621/main/classification-output-data%2

summary(df)
```

```
##     pregnant         glucose        diastolic        skinfold
##  Min.   : 0.000   Min.   : 57.0   Min.   : 38.0   Min.   : 0.0
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 64.0   1st Qu.: 0.0
##  Median : 3.000   Median :112.0   Median : 70.0   Median :22.0
##  Mean   : 3.862   Mean   :118.3   Mean   : 71.7   Mean   :19.8
##  3rd Qu.: 6.000   3rd Qu.:136.0   3rd Qu.: 78.0   3rd Qu.:32.0
##  Max.   :15.000   Max.   :197.0   Max.   :104.0   Max.   :54.0
##     insulin            bmi           pedigree           age
##  Min.   :  0.00   Min.   :19.40   Min.   :0.0850   Min.   :21.00
##  1st Qu.:  0.00   1st Qu.:26.30   1st Qu.:0.2570   1st Qu.:24.00
##  Median :  0.00   Median :31.60   Median :0.3910   Median :30.00
##  Mean   : 63.77   Mean   :31.58   Mean   :0.4496   Mean   :33.31
##  3rd Qu.:105.00   3rd Qu.:36.00   3rd Qu.:0.5800   3rd Qu.:41.00
##  Max.   :543.00   Max.   :50.00   Max.   :2.2880   Max.   :67.00
##      class         scored.class    scored.probability
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.02323
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.11702
##  Median :0.0000   Median :0.0000   Median :0.23999  
##  Mean   :0.3149   Mean   :0.1768   Mean   :0.30373  
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.43093  
##  Max.   :1.0000   Max.   :1.0000   Max.   :0.94633  
```

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?

```
df%>%select(scored.class, class) %>%
  table()
```

```
##             class
## scored.class   0   1
##            0 119  30
##            1   5  27
```

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.

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

```
fun<- function(x){
  c<- table(x$class,  x$scored.class)
  acc <- (c[1,1]+c[2,2])/sum(c)
  return(acc)
}

(acc<-fun(df))
```

```
## [1] 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.

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

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

```
fun2<- function(x) {
    c <- table(x$class, x$scored.class)
    err <- (c[1, 2] + c[2, 1]) / sum(c)
    return(err)
}
(err <- fun2(df))
```

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

```
fun(df) + fun2(df)
```

```
## [1] 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.

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

```
p <- function(x) {
    c<- table(x$class, x$scored.class)
    precision <- c[2, 2] / (c[2, 2] +c[1, 2])
    return(precision)
}
(precision <- p(df))
```

```
## [1] 0.84375
```

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.

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

```
sens <-  function(x) {
    c <- table(factor(x$class, levels = c(0, 1)),
                factor(x$scored.class, levels = c(0, 1)))
    ss <- c[2, 2] / (c[2, 2] + c[2, 1])
    return(ss)
}
(ss <- sens(df))
```

```
## [1] 0.4736842
```

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.

$$\text{Specificity} = \frac{TN}{TN + FP}$$

```
fun_spec<- function(x) {

    c <- table(factor(x$class, levels = c(0, 1)),
                factor(x$scored.class, levels = c(0, 1)))
    s <- c[1, 1] / (c[1, 1] + c[1, 2])
    return(s)
}
(s <- fun_spec(df))
```

```
## [1] 0.9596774
```

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.

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

```r
score <- function(x){
  (2*p(x)*sens(x))/(p(x)+sens(x))
}

score(df)
```

```
## [1] 0.6067416
```