Untitled

Shariq Mian

3/20/2022

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 1.1.4 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()

df=read.csv('https://github.com/mianshariq/SPS/raw/91ba7daade82c6d57e1c585f10181eda7023b976/Data%20621/HW2/classification-output-data.csv')  
head(df,3)

## pregnant glucose diastolic skinfold insulin bmi pedigree age class  
## 1 7 124 70 33 215 25.5 0.161 37 0  
## 2 2 122 76 27 200 35.9 0.483 26 0  
## 3 3 107 62 13 48 22.9 0.678 23 1  
## scored.class scored.probability  
## 1 0 0.3284523  
## 2 0 0.2731904  
## 3 0 0.1096604

df1 <- df %>%   
 select(class, scored.class, scored.probability)  
tail(df1,10)

## class scored.class scored.probability  
## 172 0 0 0.1707249  
## 173 0 0 0.1608049  
## 174 1 0 0.2457727  
## 175 0 0 0.1099905  
## 176 1 1 0.6764516  
## 177 0 0 0.3114196  
## 178 1 1 0.7072096  
## 179 1 1 0.8882766  
## 180 0 0 0.4224679  
## 181 0 0 0.1199810

df1 %>%   
 select(class, scored.class) %>%  
 table()

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

accuracy\_fun = function(df){  
 sumofall=nrow(df)  
 TN =sum(df1$class == 0 & df1$scored.class ==0)  
 TP =sum(df1$class == 1 & df1$scored.class ==1)  
 return((TN+TP)/sumofall)  
}  
  
accuracy\_fun(df1)

## [1] 0.8066298

classification\_error\_rate = function(df){  
 sumofall=nrow(df)  
 FP =sum(df1$class == 0 & df1$scored.class ==1)  
 FN =sum(df1$class == 1 & df1$scored.class ==0)  
 return((FP+FN)/sumofall)  
}  
  
classification\_error\_rate(df1)

## [1] 0.1933702

total=accuracy\_fun(df1)+classification\_error\_rate(df1)  
total

## [1] 1

precision\_func = function(df){  
 TP =sum(df1$class == 1 & df1$scored.class ==1)  
 FP =sum(df1$class == 0 & df1$scored.class ==1)  
 return((TP)/(TP+FP))  
}  
  
precision\_func(df1)

## [1] 0.84375

sensitivity\_func = function(df){  
 TP =sum(df1$class == 1 & df1$scored.class ==1)  
 FN =sum(df1$class == 1 & df1$scored.class ==0)  
 return((TP)/(TP+FN))  
}  
  
sensitivity\_func(df1)

## [1] 0.4736842

specificity\_func = function(df){  
 FP =sum(df1$class == 0 & df1$scored.class ==1)  
 TN =sum(df1$class == 0 & df1$scored.class ==0)  
 return((TN)/(TN+FP))  
}  
  
specificity\_func(df1)

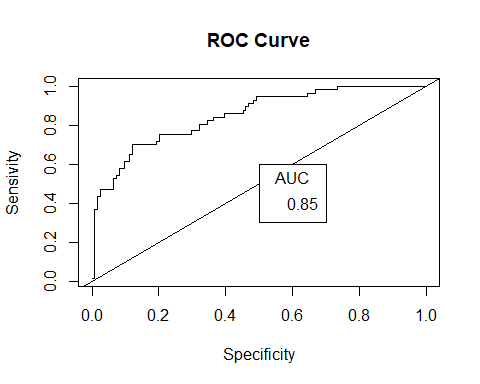
## [1] 0.9596774

F1Score = function(df){  
 precision=precision\_func(df1)  
 sensitivity=sensitivity\_func(df1)  
 return((2\*precision\*sensitivity)/(precision+sensitivity))  
}  
  
F1Score(df1)

## [1] 0.6067416

F1 Score bound be between 0 and 1. If a (Precision) is between 0 and 1 and Sensitivity is between 0 and 1, then the limit approaches 0 but not 0. Then a\*b is less than a. In this case when the limit approaches 0 then F1 score approaches 0.If a approaches 1 then F1 score approaches 1.

ROCcurve = function(x, y){  
 x <- x[order(y, decreasing = TRUE)]  
 TPRate <- cumsum(x) / sum(x)  
 FPRate <- cumsum(!x) / sum(!x)  
 xy <- data.frame(TPRate, FPRate, x)  
   
 FPRate\_df <- c(diff(xy$FPRate), 0)  
 TPRate\_df <- c(diff(xy$TPRate), 0)  
 AUC <- round(sum(xy$TPRate \* FPRate\_df) + sum(TPRate\_df \* FPRate\_df)/2, 3)  
   
 plot(xy$FPRate, xy$TPRate, type = "l",  
 main = "ROC Curve",  
 xlab = "Specificity",  
 ylab = "Sensivity")  
 abline(a = 0, b = 1)  
 legend(.5, .6, AUC, title = "AUC")  
}  
  
ROCcurve(df$class,df$scored.probability)



acc=accuracy\_fun(df)  
clas=classification\_error\_rate(df)  
pres=precision\_func(df)   
sens=sensitivity\_func(df)  
spec=specificity\_func(df)  
f1=F1Score(df)  
output <- c("Accuracy", "Classification Error Rate", "Precision", "Sensitivity",   
 "Specificity", "F1 Score")  
val <- c(acc, clas, pres, sens , spec, f1)  
result <- cbind(Metric = output, "Calculated Result" = round(val, 4))   
knitr::kable(result)

|  |  |
| --- | --- |
| Metric | Calculated Result |
| Accuracy | 0.8066 |
| Classification Error Rate | 0.1934 |
| Precision | 0.8438 |
| Sensitivity | 0.4737 |
| Specificity | 0.9597 |
| F1 Score | 0.6067 |

library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

df\_table = with(df1, table(scored.class, class)[2:1,2:1])  
confusionMatrix(df\_table)

## Confusion Matrix and Statistics  
##   
## class  
## scored.class 1 0  
## 1 27 5  
## 0 30 119  
##   
## Accuracy : 0.8066   
## 95% CI : (0.7415, 0.8615)  
## No Information Rate : 0.6851   
## P-Value [Acc > NIR] : 0.0001712   
##   
## Kappa : 0.4916   
##   
## Mcnemar's Test P-Value : 4.976e-05   
##   
## Sensitivity : 0.4737   
## Specificity : 0.9597   
## Pos Pred Value : 0.8438   
## Neg Pred Value : 0.7987   
## Prevalence : 0.3149   
## Detection Rate : 0.1492   
## Detection Prevalence : 0.1768   
## Balanced Accuracy : 0.7167   
##   
## 'Positive' Class : 1   
##

library(pROC)

## Type 'citation("pROC")' for a citation.

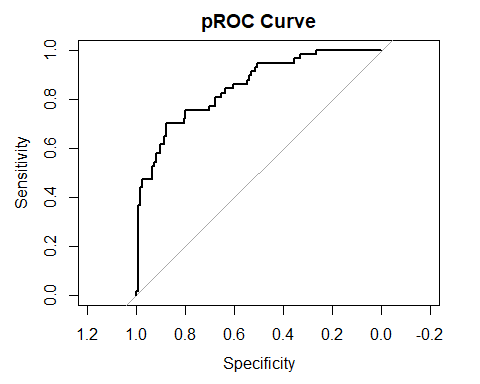
##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

roc=plot(roc(df1$class, df1$scored.probability), main="pROC Curve")

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases



roc$auc

## Area under the curve: 0.8503