FML Assignment

Gayathri Yenigalla

2023-03-05

```r
#IMPORTING THE DATASET
mybank <- read.csv("C:/Users/gaya3/Downloads/UniversalBank.csv")
summary("mybank.csv")

##    Length     Class      Mode
##         1 character character

#TRANSFORMING THE PREDICTOR ATTRIBUTE INTO CATEGORICAL VARIABLES.
mybank$Personal.Loan <- as.factor(mybank$Personal.Loan)
mybank$Online <- as.factor(mybank$Online)
mybank$CreditCard <- as.factor(mybank$CreditCard)

#CHECKING FOR NULL VALUES
sum(is.na(mybank))

## [1] 0

#IMPORTING THE REQUIRED LIBRARIES
library(class)
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(e1071)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(ggplot2)
library(reshape)

##
## Attaching package: 'reshape'
```

```
## The following object is masked from 'package:dplyr':
##
##     rename

## The following object is masked from 'package:class':
##
##     condense

library(melt)
library(ISLR)
library(reshape2)

##
## Attaching package: 'reshape2'

## The following objects are masked from 'package:reshape':
##
##     colsplit, melt, recast

library(readr)
library(naivebayes)

## naivebayes 0.9.7 loaded

library(pROC)

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

##
## Attaching package: 'pROC'

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

#DIVIDING THE DATASET INTO A 60:40 RATIO.
set.seed(123)
datapart <- createDataPartition(mybank$Personal.Loan,p=.6, list=F)
Train <- mybank[datapart,]
Validate <- mybank[-datapart,]

#DATA NORMALIZATION
norms_model <- preProcess(Train[,-c(10,13:14)],
                method=c("center","scale"))
Train_norms <- predict(norms_model,Train)
Validate_norms <- predict(norms_model,Validate)

#A. GENERATE A PIVOT TABLE BASED ON THE TRAINING DATASET THAT HAS ONLINE AS
ONE OF THE COLUMN VARIABLES, CC AS A ROW VARIABLE, AND LOAN AS A SUB-ROW
VARIABLE.
tab1<- ftable(Train_norms[,c(14,10,13)])
tab1
```

```
##                           Online    0    1
## CreditCard Personal.Loan
## 0          0                      791 1144
##            1                       79  125
## 1          0                      310  467
##            1                       33   51
```

#B.”The likelihood of a loan being approved (Loan = 1) given that the applicant has a bank credit card (CC = 1) and is an active user of online banking services (Online = 1) is calculated as 51 divided by the sum of 51 and 467, resulting in a probability of 0.0984.”

#C. Construct two distinct pivot tables based on the training dataset. One table should show Loan as the row variable and Online as the column variable, while the other table should show Loan as the row variable and CC as the column variable.

```
melt1 = melt(Train, id=c("CreditCard","Personal.Loan"), variable = "Online")

## Warning: attributes are not identical across measure variables; they will
be
## dropped

castbank = dcast(melt1, CreditCard+Personal.Loan~Online)

## Aggregation function missing: defaulting to length

castbank[,c(1:2,14)]

##   CreditCard Personal.Loan Online
## 1          0             0   1935
## 2          0             1    204
## 3          1             0    777
## 4          1             1     84
```

#D.Calculate the following values [P(A | B), which represents the probability of A given B].

```
ftable(Train_norms[,c(10,13)])

##               Online    0    1
## Personal.Loan
## 0                    1101 1611
## 1                     112  176

ftable(Train_norms[,c(10,14)])

##               CreditCard    0    1
## Personal.Loan
## 0                        1935  777
## 1                         204   84

ftable(Train_norms[,10])
```

```
##     0    1
##
##  2712  288
```

#1. The probability of having a bank credit card (CC = 1) given that the loan is approved (Loan = 1) is calculated as 84 divided by the sum of 84 and 204, resulting in a probability of 0.291. #2.The probability of being an active user of online banking services (Online = 1) given that the loan is approved (Loan = 1) is calculated as 176 divided by the sum of 176 and 112, resulting in a probability of 0.611. #3. The probability of loan approval (Loan = 1) is calculated as 288 divided by the sum of 288 and 2712, resulting in a probability of 0.096. #4. The probability of having a bank credit card (CC = 1) given that the loan is not approved (Loan = 0) is calculated as 777 divided by the sum of 777 and 1935, resulting in a probability of 0.286. #5. The probability of being an active user of online banking services (Online = 1) given that the loan is not approved (Loan = 0) is calculated as 1611 divided by the sum of 1611 and 1101, resulting in a probability of 0.595. #6. The probability of loan rejection (Loan = 0) is calculated as 2712 divided by the sum of 2712 and 288, resulting in a probability of 0.904.

#E. Utilize the values calculated previously to compute the naive Bayes probability of P(Loan = 1 | CC = 1, Online = 1)

```
#(0.291 x 0.611 x 0.096) / (0.271 x 0.611 x 0.096) + (0.286 x 0.595 x 0.904)
= 0.1000
```

#F. Observing the results obtained in steps b and a, we can conclude that the values are nearly identical, with Naive Bayes yielding a slightly higher probability of acceptance, which is 0.1000 compared to 0.0984.

#G. Execute the Naive Bayes model on the given dataset

```
Naive <- naive_bayes(Personal.Loan~Online+CreditCard,data=Train_norms)
Naive

##
## ================================== Naive Bayes
==================================
##
##  Call:
## naive_bayes.formula(formula = Personal.Loan ~ Online + CreditCard,
##     data = Train_norms)
##
## ---------------------------------------------------------------------------
-------
##
## Laplace smoothing: 0
##
## ---------------------------------------------------------------------------
-------
##
##  A priori probabilities:
```

```
##
##     0     1
## 0.904 0.096
##
## ---------------------------------------------------------------------------
-------
##
##  Tables:
##
## ---------------------------------------------------------------------------
-------
##  ::: Online (Bernoulli)
## ---------------------------------------------------------------------------
-------
##
## Online         0         1
##      0 0.4059735 0.3888889
##      1 0.5940265 0.6111111
##
## ---------------------------------------------------------------------------
-------
##  ::: CreditCard (Bernoulli)
## ---------------------------------------------------------------------------
-------
##
## CreditCard         0         1
##          0 0.7134956 0.7083333
##          1 0.2865044 0.2916667
##
## ---------------------------------------------------------------------------
-------

#Naive Bayes Model results for the consumer taking the loan, using their
credit card, and using online banking are 0.1000, which is equivalent to the
result in E.
```

#Analyzing the area under the curve (AUC) value and receiver operating characteristic (ROC) curve

```
Naive <- naiveBayes(Personal.Loan~Online+CreditCard,data=Train_norms)
Naive

##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##     0     1
```

```
## 0.904 0.096
##
## Conditional probabilities:
##    Online
## Y           0         1
##   0 0.4059735 0.5940265
##   1 0.3888889 0.6111111
##
##    CreditCard
## Y           0         1
##   0 0.7134956 0.2865044
##   1 0.7083333 0.2916667
```

```
predlab <- predict(Naive,Validate_norms,type = "raw")
head(predlab)
```

```
##              0          1
## [1,] 0.9082737 0.09172629
## [2,] 0.9021538 0.09784623
## [3,] 0.9061594 0.09384060
## [4,] 0.9082737 0.09172629
## [5,] 0.9082737 0.09172629
## [6,] 0.8999139 0.10008606
```

```
roc(Validate_norms$Online,predlab[,2])
```

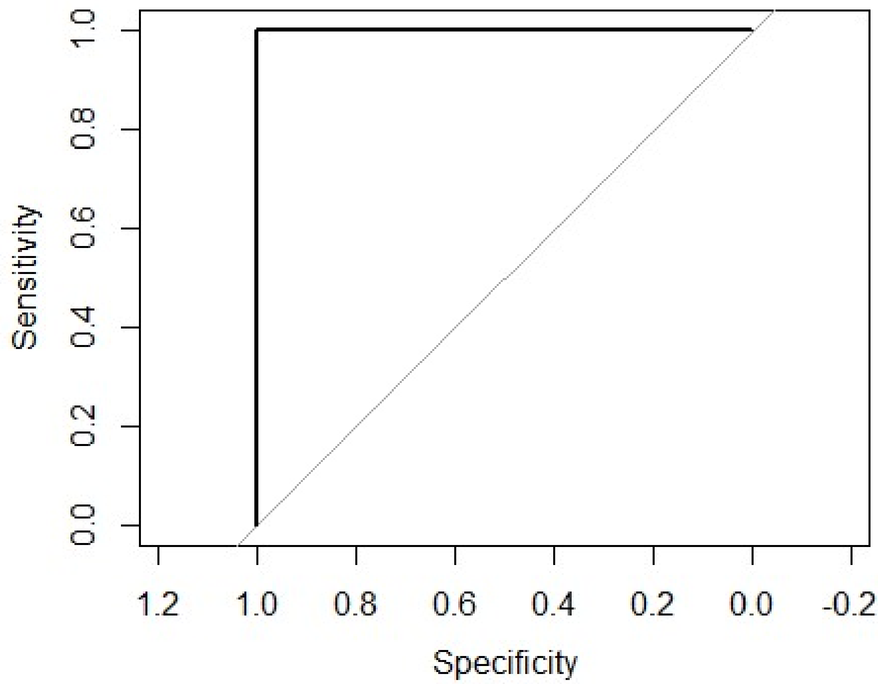
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
##
## Call:
## roc.default(response = Validate_norms$Online, predictor = predlab[,
2])
##
## Data: predlab[, 2] in 803 controls (Validate_norms$Online 0) < 1197 cases
(Validate_norms$Online 1).
## Area under the curve: 1
```

```
plot.roc(Validate_norms$Online,predlab[,2])
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```



Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
Specificity