FML Assignment 3

shiva gadila

2023-03-05

#Importing dataset

nb <- read.csv("~/Desktop/kent /BUSINESS ANALYTICS /ASSIGNMENT 1/UniversalBank.csv")  
summary("nb.csv")

## Length Class Mode   
## 1 character character

#CONVERTING THE ATTRIBUTE OF THE PREDICTOR TO FACTORS

head(nb)

## ID Age Experience Income ZIP.Code Family CCAvg Education Mortgage  
## 1 1 25 1 49 91107 4 1.6 1 0  
## 2 2 45 19 34 90089 3 1.5 1 0  
## 3 3 39 15 11 94720 1 1.0 1 0  
## 4 4 35 9 100 94112 1 2.7 2 0  
## 5 5 35 8 45 91330 4 1.0 2 0  
## 6 6 37 13 29 92121 4 0.4 2 155  
## Personal.Loan Securities.Account CD.Account Online CreditCard  
## 1 0 1 0 0 0  
## 2 0 1 0 0 0  
## 3 0 0 0 0 0  
## 4 0 0 0 0 0  
## 5 0 0 0 0 1  
## 6 0 0 0 1 0

nb$Personal.Loan <- as.factor(nb$Personal.Loan)  
nb$Online <- as.factor(nb$Online)  
nb$CreditCard <- as.factor(nb$CreditCard)

#CHECKING VALUES FOR NULL

sum(is.na(nb))

## [1] 0

#LOADING THE 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

#DATA COMPONENT TO 60:40

set.seed(123)  
 datapart <- createDataPartition(nb$Personal.Loan,p=.6, list=F)  
 Train <- nb[datapart,]  
 Validate <- nb[-datapart,]

#DATA NORMALIZATION

normalization.model <- preProcess(Train[,-c(10,13:14)],   
 method=c("center","scale"))  
 Train.normalization <- predict(normalization.model,Train)  
 Validate.normalization<- predict(normalization.model,Validate)

#A: In a pivot table, make Online a column variable, CC a row variable, and Loan a secondary row variable for the training data.

pivot<- ftable(Train.normalization[,c(14,10,13)])  
pivot

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

#B. This is the probability that you will be approved for a loan (Loan = 1) if you have a bank credit card (CC = 1) and use online banking services frequently (Online = 1): 51/(51+467) = 0.0984.

#C. Creating two separate pivot tables for the training data. One having Loan (rows) as a function of Online (columns) and the other having Loan (rows) as a function of CC

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.Compute the following quantities [P(A | B) i.e. the probability of A given B]

ftable(Train.normalization[,c(10,13)])

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

ftable(Train.normalization[,c(10,14)])

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

ftable(Train.normalization[,10])

## 0 1  
##   
## 2712 288

#1. P(CC = 1 | Loan = 1) = (84/84+204) = 0.291 #2. P(Online = 1 | Loan = 1) = (176/176+112) = 0.611 #3. P(Loan = 1) = (288/288+2712) = 0.096 #4. P(CC = 1 | Loan = 0) = (777/777+1935) = 0.286 #5. P(Online = 1 | Loan = 0) = (1611/ 1611+1101) = 0.595 #6. P(Loan = 0) = (2712/ 2712+288) = 0.904

#E.Use the quantities computed above to compute the naive Bayes probability 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. Although the probability using Naive Bayes is slightly higher, we can see that the values obtained in steps b, which are 0.0984 and a, which are 0.1000, are practically the same.

#G. Run the Naive Bayes Model on the data

Naiv <- naive\_bayes(Personal.Loan~Online+CreditCard,data=Train.normalization)  
Naiv

##   
## ================================== Naive Bayes ==================================   
##   
## Call:   
## naive\_bayes.formula(formula = Personal.Loan ~ Online + CreditCard,   
## data = Train.normalization)  
##   
## ---------------------------------------------------------------------------------   
##   
## 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  
##   
## ---------------------------------------------------------------------------------

#The results of the Naive Bayes Model for the borrower's use of their credit card, online banking, and loan are 0.1000, which is the same as the result in E.

#Looking at the ROC curve and AUC value  
Naiv <- naiveBayes(Personal.Loan~Online+CreditCard,data=Train.normalization)  
 Naiv

##   
## 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(Naiv,Validate.normalization,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.normalization$Online,predlab[,2])

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

## Setting direction: controls < cases

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

plot.roc(Validate.normalization$Online,predlab[,2])

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

