Assignment 3

RAJEEV VARMA

2022-10-25

#Importing the dataset  
Univ\_Bank <- read.csv("C:/Users/RAJEEV VARMA/Downloads/UniversalBank.csv")

#Converting the predictor attributes to factors  
Univ\_Bank$Online <- as.factor(Univ\_Bank$Online)  
Univ\_Bank$CreditCard <- as.factor(Univ\_Bank$CreditCard)  
Univ\_Bank$Personal.Loan <- as.factor(Univ\_Bank$Personal.Loan)

#Checking for NULL VALUES and loading packages  
sum(is.na(Univ\_Bank))

## [1] 0

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 Partition to 60:40  
set.seed(123)  
datapart <- createDataPartition(Univ\_Bank$Personal.Loan,p=.6, list=F)  
Train <- Univ\_Bank[datapart,]  
Validate <- Univ\_Bank[-datapart,]

#Data Normalization  
ModelNorm <- preProcess(Train[,-c(10,13:14)],   
 method=c("center","scale"))  
Trainnorm <- predict(ModelNorm,Train)  
Validatenorm <- predict(ModelNorm,Validate)

#A)

tab1<- ftable(Trainnorm[,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) 51/(51+467) = 0.0984.

#C)

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

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

castbank = dcast(melt, 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)

ftable(Trainnorm[,c(10,13)])

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

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

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

ftable(Trainnorm[,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)

#Computing the naive bayes probability:  
#(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 Naive Bayes has a higher probability than that with the direct calculation, the value obtained in step b, which is 0.0984, and the value obtained in step a, which is 0.1000, are almost similar.

#G)

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

##   
## ================================== Naive Bayes ==================================   
##   
## Call:   
## naive\_bayes.formula(formula = Personal.Loan ~ Online + CreditCard,   
## data = Trainnorm)  
##   
## ---------------------------------------------------------------------------------   
##   
## 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 Naive Bayes Model produces a result of 0.1000, which is equivalent to the result in E, when it is run for the consumer applying for the loan, paying with a credit card, and utilizing online banking

#Examining the AUC value and ROC curve  
Naive <- naiveBayes(Personal.Loan~Online+CreditCard,data=Trainnorm)  
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,Validatenorm,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(Validatenorm$Online,predlab[,2])

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

## Setting direction: controls < cases

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

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

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

