Assignment_3

Jyothsna P - 811251679

2023-03-05

```
###install.packages("e1071")
###loading the required packages:dplyr
###loading the required packages:ISLR
###loading the required packages:tidyverse

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(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
library(readr)
library(ISLR)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2
## --

## v tibble  3.1.8     v stringr 1.5.0
## v tidyr   1.2.1     v forcats 1.0.0
## v purrr   1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
```

Read the UniversalBankData to the R environment

```
UniversalBankData_1 <- read.csv("C:/Users/peddi/OneDrive/Desktop/Spring 2023/FML/Module 4/Assignment 2/U
```

Check the descriptive statistics of universalBank Data

```
summary(UniversalBankData_1)
```

```
##        ID             Age          Experience       Income          ZIP.Code
##  Min.   :   1   Min.   :23.00   Min.   :-3.0   Min.   :  8.00   Min.   : 9307
##  1st Qu.:1251   1st Qu.:35.00   1st Qu.:10.0   1st Qu.: 39.00   1st Qu.:91911
##  Median :2500   Median :45.00   Median :20.0   Median : 64.00   Median :93437
##  Mean   :2500   Mean   :45.34   Mean   :20.1   Mean   : 73.77   Mean   :93153
##  3rd Qu.:3750   3rd Qu.:55.00   3rd Qu.:30.0   3rd Qu.: 98.00   3rd Qu.:94608
##  Max.   :5000   Max.   :67.00   Max.   :43.0   Max.   :224.00   Max.   :96651
##      Family          CCAvg          Education        Mortgage
##  Min.   :1.000   Min.   : 0.000   Min.   :1.000   Min.   :  0.0
##  1st Qu.:1.000   1st Qu.: 0.700   1st Qu.:1.000   1st Qu.:  0.0
##  Median :2.000   Median : 1.500   Median :2.000   Median :  0.0
##  Mean   :2.396   Mean   : 1.938   Mean   :1.881   Mean   : 56.5
##  3rd Qu.:3.000   3rd Qu.: 2.500   3rd Qu.:3.000   3rd Qu.:101.0
##  Max.   :4.000   Max.   :10.000   Max.   :3.000   Max.   :635.0
##  Personal.Loan   Securities.Account   CD.Account          Online
##  Min.   :0.000   Min.   :0.0000     Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.0000     1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.000   Median :0.0000     Median :0.0000   Median :1.0000
##  Mean   :0.096   Mean   :0.1044     Mean   :0.0604   Mean   :0.5968
##  3rd Qu.:0.000   3rd Qu.:0.0000     3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :1.000   Max.   :1.0000     Max.   :1.0000   Max.   :1.0000
##    CreditCard
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.294
##  3rd Qu.:1.000
##  Max.   :1.000
```

Partitioning the data to train data 60% and validation data 40%.

```
set.seed(143)

#CreateDataPartition function helps in creating an index to partition data into desired ratio
Index_Train <- createDataPartition(UniversalBankData_1$Personal.Loan,p=0.6,list = FALSE)

#Index_train holds the row index of 60% of data.
TrainData <- UniversalBankData_1[Index_Train,]
ValidationDAta <- UniversalBankData_1[-Index_Train,]
```

Question: A

```r
## Creating pivot table for Online as a column variable, creditcard as row variable and personal loan a

table(CreditCard= TrainData$CreditCard, Online= TrainData$Online, Loan= TrainData$Personal.Loan)
```

```
## , , Loan = 0
##
##           Online
## CreditCard    0    1
##          0  764 1119
##          1  342  492
##
## , , Loan = 1
##
##           Online
## CreditCard    0    1
##          0   73  120
##          1   35   55
```

Question: B

Probability of Loan acceptance of a customer who is owing a bank credit card and actively using online services by using the above pivot table.

```r
## Calculating the Probability of accepting loan = Number of observations where Loan = 1 / Total number

LoanAcceptProb = 55/(55+492)
LoanAcceptProb
```

```
## [1] 0.1005484
```

Question: C

Creating two separate pivot tables for the training data.

Creating a pivot table for Loan(in rows) and Online(in column) for the training data using the table function.

```r
PTforLoanVSOnline = table(Loan= TrainData$Personal.Loan, Online=TrainData$Online)
print(PTforLoanVSOnline)
```

```
##     Online
## Loan    0    1
##    0 1106 1611
##    1  108  175
```

Creating a pivot table for loan(in rows) as a function of Credit card (as column)for the training data using table function.

```
PTforCCVSLoan = table( Loan= TrainData$Personal.Loan,CC= TrainData$CreditCard)
print(PTforCCVSLoan)
```

```
##     CC
## Loan    0    1
##    0 1883  834
##    1  193   90
```

Question: D

```
##Calculating the probability of credit card given by loan when ( CC=1 | Loan=1 )

DI= PTforCCVSLoan[2,2]/
(PTforCCVSLoan[2,1]+PTforCCVSLoan[2,2])

DI
```

```
## [1] 0.3180212
```

```
##Calculating the probability of Online given by Loan when ( Online=1| Loan=1 )

DII= PTforLoanVSOnline[2,2]/
  (PTforLoanVSOnline[2,1]+PTforLoanVSOnline[2,2])

DII
```

```
## [1] 0.6183746
```

```
##Calculating the probability of Loan=1

# Numerator - Sum of all values where loan=1 - sum(PTforCCVSLoan[2,])
# Denominator - Sum of all values in the matrix - sum(PTforCCVSLoan)

DIII= sum(PTforCCVSLoan[2,])/ sum(PTforCCVSLoan)


DIII
```

```
## [1] 0.09433333
```

```
##Calculating the probability of credit card given by Loan when ( CC=1 | Loan=0 )

DIV = PTforCCVSLoan[1,2]/
  (PTforCCVSLoan[1,2]+PTforCCVSLoan[1,1])

DIV
```

```
## [1] 0.3069562
```

```r
##Calculating the probability of Online given by Loan when (Online=1| Loan= 0 )

DV = PTforLoanVSOnline[1,2]/
  (PTforLoanVSOnline[1,2]+PTforLoanVSOnline[1,1])

DV
```

```
## [1] 0.5929334
```

```r
#Calculating the Probability of Loan= 0

# Numerator - Sum of all values where loan=0 - sum(PTforCCVSLoan[1,])
# Denominator - Sum of all values in the matrix - sum(PTforCCVSLoan)

DVI = sum(PTforCCVSLoan[1,])/
         sum(PTforCCVSLoan)


DVI
```

```
## [1] 0.9056667
```

Question: E

Computing the naive Bayes probability using the quantities computed above for P(Loan = 1 | CC= 1, Online = 1)

```r
E1= (DI*DII*DIII)/((DI*DII*DIII)+(DIV*DV*DVI))

E1
```

```
## [1] 0.1011591
```

Question: F

Comparing the probability values obtained in B and in naive Bayes model in E.

The value obtained in the B is 0.1005484 and value obtained by computing the naive Bayes probability in the E is 0.1011591. By comparing them it is clear that probability obtained from solving naive Bayes model have higher probability than that in B.

Question: G

```r
## Creating the Test data by applying the filters on train data on condition when CC=1 and Online=1

TestData = filter(TrainData, CreditCard== 1 & Online== 1)

head(TestData)
```

```
##   ID Age Experience Income ZIP.Code Family CCAvg Education Mortgage
## 1 45  46         20    104    94065      1   5.7         1        0
## 2 48  37         12    194    91380      4   0.2         3      211
## 3 66  59         35    131    91360      1   3.8         1        0
## 4 69  47         21     60    93407      3   2.1         1        0
## 5 74  41         16     85    94606      1   4.0         3        0
## 6 76  31          7    135    94901      4   3.8         2        0
##   Personal.Loan Securities.Account CD.Account Online CreditCard
## 1             0                  0          0      1          1
## 2             1                  1          1      1          1
## 3             0                  0          0      1          1
## 4             0                  0          0      1          1
## 5             0                  0          0      1          1
## 6             1                  0          1      1          1
```

Performing the Naive Bayes model on the training data.

```r
nb_Model = naiveBayes(Personal.Loan~ Online + CreditCard, data = TrainData)
nb_Model
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##          0          1
## 0.90566667 0.09433333
##
## Conditional probabilities:
##    Online
## Y        [,1]      [,2]
##   0 0.5929334 0.4913779
##   1 0.6183746 0.4866460
##
##    CreditCard
## Y        [,1]      [,2]
##   0 0.3069562 0.4613160
##   1 0.3180212 0.4665328
```

Making the predictions and returning the probability values by using the naive Bayes model.

```
Predicted_Test_Model<- predict(nb_Model,TestData, type = "raw")

head(Predicted_Test_Model,n = 8)##showing the first few values from the predicted_test_model
```

```
##              0         1
## [1,] 0.8973112 0.1026888
## [2,] 0.8973112 0.1026888
## [3,] 0.8973112 0.1026888
## [4,] 0.8973112 0.1026888
## [5,] 0.8973112 0.1026888
## [6,] 0.8973112 0.1026888
## [7,] 0.8973112 0.1026888
## [8,] 0.8973112 0.1026888
```

Comparing the probabilities obtained by the Naive Bayes model on training data in E is 0.1011591 and on test data in G is 0.1026888, which are almost same.