Assignment3

Lava Kumar

3/6/2022

```
#Reading the UniversalBank.csv file.
Universal_Bank <- read.csv("C:/Users/lavak/Documents/R/Assignment3/UniversalBank.csv")
View(Universal_Bank)    #To view the data of dataset

#First required libraries have to be loaded using library fuction.
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(ISLR)
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(class)
library(reshape2)
library(ggplot2)
library(gmodels)
library(lattice)

summary(Universal_Bank)  # To check different values in the dataset
```

```
##        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
```

```r
#variables coversion to factor
Universal_Bank$Personal.Loan <- as.factor(Universal_Bank$Personal.Loan)
Universal_Bank$Online <- as.factor(Universal_Bank$Online)
Universal_Bank$CreditCard <- as.factor(Universal_Bank$CreditCard)
df= Universal_Bank

#Partitioning the data
set.seed(64060)
Train_Index1 <- createDataPartition(df$Personal.Loan, p = 0.6, list = FALSE)
Train1.df = df[Train_Index1,]
validation.df = df[-Train_Index1,]

#TASK1:Created a pivot table for the training data with Online as a #column variable, CC
#as a row variable, and Loan as a secondary row
#variable.The values inside the table conveying the count.

pitable <- xtabs(~ CreditCard + Online + Personal.Loan , data = Train1.df)
ftable(pitable)
```

```
##                   Personal.Loan    0    1
## CreditCard Online
## 0          0                    772   75
##            1                   1152  120
## 1          0                    309   34
##            1                    479   59
```

```r
#TASK2:Calculating the probability that this customer will accept #the loan offer

T2Probability = 59/(59+479)
T2Probability
```

```
## [1] 0.1096654
```

```
#TASK3:Creating two separate pivot tables for the training data. #One will have Loan (rows) as a
#function of Online (columns) and #the other will have Loan (rows) as a function of CC.

table(Personal.Loan = Train1.df$Personal.Loan, Online = Train1.df$Online)
```

```
##               Online
## Personal.Loan    0    1
##             0 1081 1631
##             1  109  179
```

```
table(Personal.Loan = Train1.df$Personal.Loan, CreditCard = Train1.df$CreditCard)
```

```
##               CreditCard
## Personal.Loan    0    1
##             0 1924  788
##             1  195   93
```

```
table(Personal.Loan = Train1.df$Personal.Loan)
```

```
## Personal.Loan
##    0    1
## 2712  288
```

```
#TASK4:Computing the following quantities [P(A | B) means "the probability ofA given B"]:

#i. P(CC = 1 | Loan = 1) (the proportion of credit card holders among the loan
#acceptors)
T4Probability1 <- 93/(93+195)
T4Probability1
```

```
## [1] 0.3229167
```

```
#ii. P(Online = 1 | Loan = 1)
T4Probability2 <- 179/(179+109)
T4Probability2
```

```
## [1] 0.6215278
```

```
#iii. P(Loan = 1) (the proportion of loan acceptors)
T4Probability3 <- 288/(288+2712)
T4Probability3
```

```
## [1] 0.096
```

```
#iv. P(CC = 1 | Loan = 0)
T4Probability4 <- 788/(788+1924)
T4Probability4
```

```
## [1] 0.2905605
```

```r
#v. P(Online = 1 | Loan = 0)
T4Probability5 <- 1631/(1631+1081)
T4Probability5
```

```
## [1] 0.6014012
```

```r
#vi. P(Loan = 0)
T4Probability6 <- 2712/(2712+288)
T4Probability6
```

```
## [1] 0.904
```

```r
#TASK5: Using the quantities computed above
#to compute the naive
#Bayes probability P(Loan = 1 | CC = 1, Online = 1).

T5Probability <- (T4Probability1*T4Probability2*T4Probability3)/
  ((T4Probability1*T4Probability2*T4Probability3) +(T4Probability4*T4Probability5*T4Probability6))

T5Probability
```

```
## [1] 0.1087106
```

```r
#TASK6

#Compare this value with the one obtained from
#the pivot table in
#Task 2. Which is a more accurate estimate?

#As of Task 2, the value we obtained was 0.1096654, and the value we obtained from Task 5 is 0.1087106.
#Unlike the exact technique, the naive Bayes method does not need to categorize independent variables
#before forecasting, as the exact method does.
#As we used the exact data from the pivot table, we
#can verify that the result obtained from Task 2 is more precise.

#Task7

#Which of the entries in this table are needed for computing P(Loan = 1 | CC = 1, Online = 1)?
#Run naive Bayes on the data. Examine the model output on training data, and find the entry
#that corresponds to P(Loan = 1 | CC = 1, Online = 1).
#Compare this to the number you obtained in Task 5.

NB.Model <- naiveBayes(Personal.Loan~ Online + CreditCard, data = Train1.df)
To_Predict1=data.frame(Online=1, CreditCard= 1)
predict(NB.Model, To_Predict1,type = 'raw')
```

```
## Warning in predict.naiveBayes(NB.Model, To_Predict1, type = "raw"): Type
## mismatch between training and new data for variable 'Online'. Did you use
## factors with numeric labels for training, and numeric values for new data?
```

```
## Warning in predict.naiveBayes(NB.Model, To_Predict1, type = "raw"): Type
## mismatch between training and new data for variable 'CreditCard'. Did you use
## factors with numeric labels for training, and numeric values for new data?
```

```
##              0          1
## [1,] 0.9153656 0.08463445
```

```
#We obtained the value 0.08463445 from Task 7, and the value 0.1087106 from Task 5.
#Our results are almost identical to those obtained from Task 5 with only slight difference.
#However,this will not impact the rank order.
```