ML Assignment3 Naives Bayes

Pavan Chaitanya

2022-10-12

```
#Accessing Libraries
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(ISLR)
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(e1071)


UniBank = read.csv("C:/Users/Pavan Chaitanya/Downloads/UniversalBank.csv")
View(UniBank)
summary(UniBank)

##        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

df= UniBank
#converting variables
UniBank$Personal.Loan <- factor(UniBank$Personal.Loan)
UniBank$Online <- factor(UniBank$Online)
UniBank$CreditCard <- factor(UniBank$CreditCard)

#Question A:

set.seed(64060)
train.index <- createDataPartition(df$Personal.Loan, p =0.6, list = FALSE)
train.df = df[train.index,]
validation.df = df[-train.index,]


# Using the ftable() instead of melt() and cast().
mytable <- xtabs(~ CreditCard + Online + Personal.Loan , data = train.df)
ftable(mytable)

##                    Personal.Loan    0    1
## CreditCard Online
## 0          0                      787   76
##            1                     1144  124
## 1          0                      307   35
##            1                      477   50

#Question B:
```

```r
Probability = (50/(50+477))
Probability

## [1] 0.09487666

#0.09487666 is the probability that this customer will accept the loan offer

#Question C:

#Loan (rows) as a function of Online (columns)
table(Personal.Loan = train.df$Personal.Loan, Online = train.df$Online)

##                Online
## Personal.Loan    0    1
##              0 1094 1621
##              1  111  174

#Loan (rows) as a function of CC(Credit Card)
table(Personal.Loan = train.df$Personal.Loan, CreditCard =
train.df$CreditCard)

##                CreditCard
## Personal.Loan    0    1
##              0 1931  784
##              1  200   85

table(Personal.Loan = train.df$Personal.Loan)

## Personal.Loan
##    0    1
## 2715  285

#Question D:

#i. P(CC = 1 | Loan = 1) (the proportion of credit card holders among the
loan acceptors)
Prob1 <- 85/(85+200)
Prob1

## [1] 0.2982456

#ii. P(Online = 1 | Loan = 1)
Prob2 <- 174/(174+111)
Prob2

## [1] 0.6105263

#iii. P(Loan = 1) (the proportion of loan acceptors)
Prob3 <- 285/(285+2715)
Prob3

## [1] 0.095
```

```r
#iv. P(CC = 1 | Loan = 0)
Prob4 <- 784/(784+1931)
Prob4

## [1] 0.2887661

#v. P(Online = 1 | Loan = 0)
Prob5 <- 1621/(1621+1094)
Prob5

## [1] 0.5970534

#vi. P(Loan = 0)
Prob6 <- 2715/(2715+285)
Prob6

## [1] 0.905

#Question E :

#P(Loan = 1 | CC = 1, Online = 1).

Probability <- (Prob1*Prob2*Prob3)/((Prob1*Prob2*Prob3)+(Prob4*Prob5*Prob6))
Probability

## [1] 0.09980052

#Question F :

# The value derived in  Question B is 0.09487666.The value derived in
Question E is 0.09980052.The only difference between the exact method and the
naive-bayes method is the exact method would need the the exact same
independent variable classifications to predict,whereas the naive bayes
method does not.We can say that the value derived from the Question B is more
accurate as we have taken the exact values from the pivot table.

#Question G :

nb.model<-naiveBayes(Personal.Loan~ Online +CreditCard, data=train.df)
To_Predict=data.frame(Online= 1, CreditCard= 1)
predict(nb.model,To_Predict,type='raw')

##              0         1
## [1,] 0.8986774 0.1013226

#The value derived from the Question G is 0.1013226.
#The value derived from the Question E is 0.09980052.
#There is a slight difference in both the values.
```