FML assignment 3 amettu1

Adithya Reddy Mettu

2024-03-11

#Loading the required Packages

```
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(lattice)
library(ISLR)
library(e1071)
library(caret)
```

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

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
library(class)
library(ggplot2)
library(tidyr)
library(gmodels)
library(lattice)
```

#Loading the Universal bank data and sumary

```
unibank.df <- read.csv("C:/Users/adith/Downloads/UniversalBank.csv")
summary(unibank.df)
```

```
##        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
```

#converting the varaibles by using factors.

```
unibank.df$Personal.Loan <- factor(unibank.df$Personal.Loan)
unibank.df$Online <- factor(unibank.df$Online)
unibank.df$CreditCard <- factor(unibank.df$CreditCard)
```

#lets perform the given tasks from here #set the seed also create a data partation of 60% training and remaining as validation

```
set.seed(123)
Train.index <- createDataPartition(unibank.df$Personal.Loan,p = 0.6,list =
FALSE)
train.df <- unibank.df[Train.index,]
validation.df <- unibank.df[-Train.index,]
```

#Create 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.

```
P.table <- xtabs(~ CreditCard + Online + Personal.Loan,data = train.df)
ftable(P.table)

##                   Personal.Loan    0    1
## CreditCard Online
## 0          0                    791   79
```

```
##             1                    1144  125
## 1           0                     310   33
##             1                     467   51
```

#Consider the task of classifying a customer who owns a bank credit card and is actively using online banking services. Looking at the pivot table, what is the probability that this customer will accept the loan offer? [This is the probability of loan acceptance (Loan = 1) conditional on having a bank credit card (CC = 1) and being an active user of online banking services (Online = 1)].

```
probability = 51/(51+467)
probability
```

```
## [1] 0.0984556
```

#Create 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 = train.df$Personal.Loan, Online = train.df$Online)
```

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

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

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

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

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

#consider p as probablity #Compute the following quantities [P(A | B) means “the probability of A given B”]:

#i. P(CC = 1 | Loan = 1) (the proportion of credit card holders among the loan #acceptors)

```
p1 <- 84/(84+204)
p1
```

```
## [1] 0.2916667
```

#ii. P(Online = 1 | Loan = 1)

```
p2 <- 176/(176+112)
p2
```

```
## [1] 0.6111111
```

#iii. P(Loan = 1) (the proportion of loan acceptors)

```
p3 <- 288/(288+2712)
p3

## [1] 0.096
```

#iv. P(CC = 1 | Loan = 0)

```
p4 <- 777/(777+1935)
p4

## [1] 0.2865044
```

#v. P(Online = 1 | Loan = 0)

```
p5 <- 1611/(1611+1101)
p5

## [1] 0.5940265
```

#vi. P(Loan = 0)

```
p6 <- 2712/(2712+288)
p6

## [1] 0.904
```

#Task5 #Use the quantities computed above to compute the naive Bayes probability P(Loan = 1 | CC #= 1, Online = 1).

```
computed_probablity <- (p1 * p2 * p3)/((p1 * p2 * p3) + (p4 * p5 * p6))
computed_probablity

## [1] 0.1000861
```

#Compare this value with the one obtained from the pivot table in (B). Which is a more #accurate estimate? #Value we got from question 2 was 0.0984556 and in the question 5 is 0.1000861 are almost same. The only difference between by the exact method and naive bayes method is the exact method would need the exact same independent variable classification to predict, whereas the naive bayes method does not. We can confirm that the value get from the question 2 is more accurate. Since we have taken the exact values from the pivot table.

#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 (E).

```
naviebayes.model <- naiveBayes(Personal.Loan ~ Online + CreditCard, data =
train.df)
```

```
to.predict = data.frame(Online=1, CreditCard= 1)
predict(naviebayes.model, to.predict,type = 'raw')

## Warning in predict.naiveBayes(naviebayes.model, to.predict, 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(naviebayes.model, to.predict, 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,] 0.9079363 0.09206369
```

#The value we got from question 7 is 0.09206369 and value derived from the task 5 is 0.1000861. # the result is almost same that we got from Task5. # There is only a minute difference because of the rounding. #The difference will not effect the rank order of the output.