Assignment 3_FML

VaishnaviD

2024-02-27

Problem Statement -

The file UniversalBank.csv contains data on 5000 customers of Universal Bank. The data include customer demographic information (age, income, etc.), the customer's relationship with the bank (mortgage, securities account, etc.), and the customer response to the last personal loan campaign (Personal Loan). Among these 5000 customers, only 480 (= 9.6%) accepted the personal loan that was offered to them in the earlier campaign. In this exercise, we focus on two predictors: Online (whether or not the customer is an active user of online banking services) and Credit Card (abbreviated CC below) (does the customer hold a credit card issued by the bank), and the outcome Personal Loan (abbreviated Loan below).

To load required libraries

```
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)
```

To specify the file location and showing dimensions

```
UniversalBank <- read.csv("C:\\Users\\Vaishnavi\\OneDrive - Kent State University\\FML\\Assignment 3\\UniversalBa
nk.csv")
summary(UniversalBank)
```

```
##        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
```

To confirm that all the data has been properly imported

```
head(UniversalBank)
```

```
##   ID Age Experience Income ZIP.Code Family CCAvg Education Mortgage
## 1  1  25          1     49    91107      4   1.6         1        0
## 2  2  45         19     34    90089      3   1.5         1        0
## 3  3  39         15     11    94720      1   1.0         1        0
## 4  4  35          9    100    94112      1   2.7         2        0
## 5  5  35          8     45    91330      4   1.0         2        0
## 6  6  37         13     29    92121      4   0.4         2      155
##   Personal.Loan Securities.Account CD.Account Online CreditCard
## 1             0                  1          0      0          0
## 2             0                  1          0      0          0
## 3             0                  0          0      0          0
## 4             0                  0          0      0          0
## 5             0                  0          0      0          1
## 6             0                  0          0      1          0
```

```
tail(UniversalBank)
```

```
##        ID Age Experience Income ZIP.Code Family CCAvg Education Mortgage
## 4995 4995  64         40     75    94588      3   2.0         3        0
## 4996 4996  29          3     40    92697      1   1.9         3        0
## 4997 4997  30          4     15    92037      4   0.4         1       85
## 4998 4998  63         39     24    93023      2   0.3         3        0
## 4999 4999  65         40     49    90034      3   0.5         2        0
## 5000 5000  28          4     83    92612      3   0.8         1        0
##      Personal.Loan Securities.Account CD.Account Online CreditCard
## 4995             0                  0          0      1          0
## 4996             0                  0          0      1          0
## 4997             0                  0          0      1          0
## 4998             0                  0          0      0          0
## 4999             0                  0          0      1          0
## 5000             0                  0          0      1          1
```

```
dim(UniversalBank)
```

```
## [1] 5000   14
```

To convert variables to factors

After converting the variables to factors we will then assign the modified dataframe to a new variable

```
UniversalBank$Personal.Loan <- factor(UniversalBank$Personal.Loan)
UniversalBank$Online <- factor(UniversalBank$Online)
UniversalBank$CreditCard <- factor(UniversalBank$CreditCard)
newdf= UniversalBank
```

Question 1 - 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.The values inside the table should convey the count. In R use functions melt() and cast(), or function table().

```
set.seed(64060)
Train_index <- createDataPartition(newdf$Personal.Loan, p = 0.6, list = FALSE)
train.df = newdf[Train_index,]
validation.df = newdf[-Train_index,]
mytable <- xtabs(~ CreditCard + Online + Personal.Loan , data = train.df)
ftable(mytable)
```

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

Question 2 - 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 = 59/(59+479)
probability
```

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

Question 3 - 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 1081 1631
##             1  109  179
```

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

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

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

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

Question 4 - 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)
Probablity1 <- 93/(93+195)
Probablity1
```

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

```
#ii. P(Online = 1 | Loan = 1)
Probablity2 <- 179/(179+109)
Probablity2
```

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

```
#iii. P(Loan = 1) (the proportion of loan acceptors)
Probablity3 <- 288/(288+2712)
Probablity3
```

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

```
#iv. P(CC = 1 | Loan = 0)
Probablity4 <- 788/(788+1924)
Probablity4
```

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

```
#v. P(Online = 1 | Loan = 0)
Probablity5 <- 1631/(1631+1081)
Probablity5
```

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

```
#vi. P(Loan = 0)
Probablity6 <- 2712/(2712+288)
Probablity6
```

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

Question 5 - Use the quantities computed above to compute the naive Bayes probability P(Loan = 1 | CC= 1, Online = 1).

```
TaskProbablity <- (Probablity1*Probablity2*Probablity3)/
  ((Probablity1*Probablity2*Probablity3) +(Probablity4*Probablity5*Probablity6))

TaskProbablity
```

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

Question 6 - Compare this value with the one obtained from the pivot table in (B) ie Question 2. Which is a more accurate estimate? -

The result from question 2 was 0.1096654 and from question 5 it was 0.1087106, which are very similar. The main difference between the exact method and the naive Bayes method is that the exact method requires the same categories of independent variables for prediction, but the naive Bayes method does not. We can say the result from question 2 is more precise because it was based on specific values from the pivot table.

Question 7 - 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).

```
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')
```

```
## Warning in predict.naiveBayes(nb.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(nb.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,] 0.9153656 0.08463445
```

The number we found in Question 7 is 0.08463445, and the number from Question 5 is 0.1087106. The result is almost same that we got from Question 5. The slight difference between them is because of rounding, but it's small.