Assignment 3

Tongxiang Lu

2022-10-14

```
# Install and load all packages.
library(readr)
library(caret)
```

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

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

```
library(ISLR)
library(class)
```

```
# Read mydata.
mydata <- read.csv("UniversalBank.csv")
View(mydata)
```

```
# We use as.factor command to convert Online, CreditCard and Personal.Loan
## variable into categorical types.
DF= mydata
DF$Online_category='Not-Active'
DF$Online_category[DF$Online>0]= 'Active'
DF$Online_category=as.factor(DF$Online_category)
DF$CreditCard=as.factor(DF$CreditCard)
DF$Personal.Loan=as.factor(DF$Personal.Loan)
summary(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     Personal.Loan
##  Min.   :1.000   Min.   : 0.000   Min.   :1.000   Min.   :  0.0   0:4520
##  1st Qu.:1.000   1st Qu.: 0.700   1st Qu.:1.000   1st Qu.:  0.0   1: 480
##  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
##  Securities.Account   CD.Account         Online       CreditCard
```

```
##  Min.   :0.0000     Min.   :0.0000   Min.   :0.0000   0:3530
##  1st Qu.:0.0000     1st Qu.:0.0000   1st Qu.:0.0000   1:1470
##  Median :0.0000     Median :0.0000   Median :1.0000
##  Mean   :0.1044     Mean   :0.0604   Mean   :0.5968
##  3rd Qu.:0.0000     3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :1.0000     Max.   :1.0000   Max.   :1.0000
##    Online_category
##  Active    :2984
##  Not-Active:2016
##
##
##
##
```

Question A

We use the set seed command to set the random seed, and use 60% training and 40% validating Data for Partition.

```
set.seed(1)
Train_Index = createDataPartition(DF$Personal.Loan, p=0.6, list=FALSE)
Train.df=DF[Train_Index,]
Validation.df=DF[-Train_Index,]
```

We use pivot table online as column variable, creditcard as a row variable, and personal.loan as a secondary row variable. The values inside the table convey the count.

```
mytable <- xtabs(~ CreditCard+Personal.Loan+Online_category, data=Train.df)
ftable(mytable)
```

```
##                          Online_category Active Not-Active
## CreditCard Personal.Loan
## 0          0                                1126        780
##            1                                 120         77
## 1          0                                 503        303
##            1                                  52         39
```

```
# Question B
## The probability that this customer will accept the loan offer is 0.09369369.
prob <- (52/(503+52))
prob
```

```
## [1] 0.09369369
```

Question C

The pivot table for the training data: rows(Creditcard) and columns(Personal.Loan) and rows (Online_category)and columns(Personal.Loan).

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

```
##            Personal.Loan
## Creditcard    0    1
##          0 1906  197
##          1  806   91
```

```
table(Online_category =Train.df$Online_category, Personal.Loan =Train.df$Personal.Loan)
```

```
##                 Personal.Loan
## Online_category    0    1
##      Active     1629  172
##      Not-Active 1083  116
```

```
# Question D
##i. P(CC = 1 | Loan = 1) is 0.316.
Prob1 <- 91/(91+197)
Prob1
```

```
## [1] 0.3159722
```

```
##ii. P(Online = 1 | Loan = 1) is 0.597.
Prob2 <- 172/(172+116)
Prob2
```

```
## [1] 0.5972222
```

```
##iii. P(Loan = 1) is 0.096.
Prob3 <- (197+91)/(197+91+1906+806)
Prob3
```

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

```
##iv. P(CC = 1|Loan = 0) is 0.297.
Prob4 <- 806/(1906+806)
Prob4
```

```
## [1] 0.2971976
```

```
##v. P(Online = 1 | Loan = 0) is 0.601.
Prob5 <- 1629/(1629+1083)
Prob5
```

```
## [1] 0.6006637
```

```
##vi. P(Loan = 0) is 0.904.
Prob6 <- (1906+806)/(1906+806+197+91)
Prob6
```

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

Question E

P(L1|C1, O1) = P(L1)[P(C1|L1)P(O1|L1)]/P(L1)[P(C1|L1)P(O1|L1)

```
                                        +P(Lo)[p(C1|Lo)P(O1|Lo)]
          = 0.096[0.316*0.597]/0.096[0.316*0.597]+ 0.904[0.297*0.601]
          = 0.018/(0.018+0.161)
          = 0.101
```

Question F

The value we obtained from the pivot table B is 0.09369369, while the value we get from naive method is 0.101. I think the former one is more accurate.

```
# Question G
library(e1071)
nb.model<-naiveBayes (Personal.Loan~Online_category+CreditCard, data=Train.df)
To_Predict=data.frame(CreditCard ='1',Online_category ='1')
predict(nb.model,To_Predict,type='raw')
```

```
##              0         1
## [1,] 0.8985507 0.1014493
```

We get the same output in the previous method which is 0.101,thus the same answer provided in the above question.