Untitled

Ashvitha M

06/03/2022

```r
bank <- read.csv("C:/Users/mashv/Downloads/UniversalBank(1).csv")

summary(bank)
```

```
##        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
library(caret)
```

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

```
## Warning in register(): Can't find generic 'scale_type' in package ggplot2 to
## register S3 method.
```

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

```r
library(ISLR)
library(e1071)
library(class)


#converting variables

bank$Personal.Loan <- factor(bank$Personal.Loan)
bank$Online <- factor(bank$Online)
bank$CreditCard <- factor(bank$CreditCard)


#TASK A

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

```r
#Task B

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

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

```r
#Task c

#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
```

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

```
##              CreditCard
```

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

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

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

```r
#Task D

#i P(CC = 1 | Loan = 1) (the proportion of credit card holders among the loan
Probability = (93/(93+195))
Probability
```

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

```r
#ii P(Online = 1 | Loan = 1)
Probability2 =(179/(179+109))
Probability2
```

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

```r
#iii P(Loan = 1) (the proportion of loan acceptors)
Probability3=(288/288+2712)
Probability3
```

```
## [1] 2713
```

```r
#iv P(CC = 1 | Loan = 0)
Probability4=(788/(788+1924))
Probability4
```

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

```r
#v P(Online = 1 | Loan = 0)
Probability5=(1631/(1631+1081))
Probability5
```

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

```r
#vi P(Loan = 0)
Probability6=(2712/2712+288)
Probability6
```

```
## [1] 289
```

```
#Task E

#Use the quantities computed above to compute the naive Ba1 probability
#P(Loan = 1 | CC = 1, Online = 1)

naiveBayes_Probability <- (Probability*Probability2*Probability3)/((Probability*Probability2*Probability
(Probability4*Probability5*Probability6))

naiveBayes_Probability
```

```
## [1] 0.9151253
```

```
#Task F

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

#0.9151253 in task E is very similar to task B 0.1096654. The different between the exact and naive bay

#Task G

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

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
```