Assignment_3

Gloria

2022-10-17

```
library(readr)
UniversalBank <- read_csv("C:/Users/idast/OneDrive/Desktop/UniversalBank.csv")
```

```
## Rows: 5000 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (14): ID, Age, Experience, Income, ZIP Code, Family, CCAvg, Education, M...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#loading Libraries
library(readr)
library(caret)
```

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

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

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

```
View(UniversalBank)
DF= UniversalBank
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$CreditCard )
DF$PersonalLoan=as.factor(DF$PersonalLoan)
```

```
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     PersonalLoan
##  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
##  SecuritiesAccount   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
##
##
##
##
```

#TASK A. #Partition of the data is segregated into 60 and 40 with a pivot table for training data with Online as column variable and CreditCard as row and loan as secondary variable.

#Using the xtabs and ftable functions.

```
set.seed(64060)
Train_Index = createDataPartition(DF$PersonalLoan,p=0.6, list=FALSE)
Train.df=DF[Train_Index,]
```

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

```
##                             PersonalLoan    0    1
## Online_category CreditCard
## Active          0                       1152  120
##                 1                        479   59
## Not-Active      0                        772   75
##                 1                        309   34
```

#Task B #The probability of loan acceptance (Loan = 1) conditionally on having a bank credit card (CC = 1) and being an active user of online banking services (Online = 1)] is 59/(59+479) = 0.10966 OR 11%

#Task C #Creating Pivot tables for training data with one of CreditCard (rows) as a function of Personal-Loan(columns) and another of Online category(rows) with PersonalLoan(columns).

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

```
##           PersonalLoan
## Creditcard    0    1
##          0 1924  195
##          1  788   93
```

```
table(Online_category =Train.df$Online_category, PersonalLoan =Train.df$PersonalLoan)
```

```
##                 PersonalLoan
## Online_category    0    1
##      Active     1631  179
##      Not-Active 1081  109
```

#Task-D #i. P(CC = 1 | Loan = 1) (the proportion of credit card holders among the loan acceptors) Ans. 93/(93+195) = 0.323 in otherwords 32.3%

#ii. P(Online = 1 | Loan = 1) Ans.179/(179+109)= 62.51%

#iii.P(Loan = 1) (the proportion of loan acceptors) Ans. 195+93/(195+93+1924+788)= 0.096 or in otherwords 9.6%

#iv. P(CC = 1 | Loan = 0) Ans.788/(1924+788)= 0.2905 or 29.05%

#v. P(Online = 1 | Loan = 0) Ans.1631/(1631+1081)=0.6014 or 60.14%

#vi. P(Loan = 0) Ans.1924+788/(1924+788+195+93)=0.904 or 90.4%

#Task-E #Using the quantities computed above to compute the naive Bayes probability P(Loan = 1 | CC = 1, Online = 1). #formula P(Y/x1. . . Xn) = (P(X1. . . . . Xn/Y)/P(X1,. . . .Xn)) X1=Online X2=Creditcard Y=PersonalLoan

P(creditcard) =0.60333 P(Online) =0.2936662 = = 0.6251*0.323*0.096/0.60333*0.2936662=0.105598 or 10.55%

#TASK-F #Both normal method (10.96%) and Naive Bayes method values (10.55%) are very similar.Both are similar, with the difference being Naive Bayes assumes attributes are independent of each other making it less accurate.

#TASK-G #Running 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).

```
library(e1071)
nb.model<-naiveBayes (PersonalLoan~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,] 0.8944381 0.1055619
```