Assignment 3 FML

Chaitanya

2022-10-26

## ---

#Importing the Dataset

library(readr)  
UniversalBank <- read\_csv("~/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.

View(UniversalBank)

#calling Libraries

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(class)  
  
library(ISLR)

#Converting Personal.loan Variable

UniversalBank$`Personal Loan`=as.factor(UniversalBank$`Personal Loan`)  
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 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 Min. :0.000   
## 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.000   
## Median :0.0000 Median :0.0000 Median :1.0000 Median :0.000   
## Mean :0.1044 Mean :0.0604 Mean :0.5968 Mean :0.294   
## 3rd Qu.:0.0000 3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:1.000   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.000

#Converting Online Variable

UniversalBank$Online = as.factor(UniversalBank$Online)   
  
summary(UniversalBank$Online)

## 0 1   
## 2016 2984

#Converting Creditcard Variable

UniversalBank$CreditCard = as.factor(UniversalBank$CreditCard)   
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 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 0:2016 0:3530   
## 1st Qu.:0.0000 1st Qu.:0.0000 1:2984 1:1470   
## Median :0.0000 Median :0.0000   
## Mean :0.1044 Mean :0.0604   
## 3rd Qu.:0.0000 3rd Qu.:0.0000   
## Max. :1.0000 Max. :1.0000

UniversalBank$Online<-as.factor(UniversalBank$Online)  
is.factor(UniversalBank$Online)

## [1] TRUE

UniversalBank$CreditCard<-as.factor(UniversalBank$CreditCard)  
is.factor(UniversalBank$CreditCard)

## [1] TRUE

str(UniversalBank)

## spec\_tbl\_df [5,000 × 14] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ ID : num [1:5000] 1 2 3 4 5 6 7 8 9 10 ...  
## $ Age : num [1:5000] 25 45 39 35 35 37 53 50 35 34 ...  
## $ Experience : num [1:5000] 1 19 15 9 8 13 27 24 10 9 ...  
## $ Income : num [1:5000] 49 34 11 100 45 29 72 22 81 180 ...  
## $ ZIP Code : num [1:5000] 91107 90089 94720 94112 91330 ...  
## $ Family : num [1:5000] 4 3 1 1 4 4 2 1 3 1 ...  
## $ CCAvg : num [1:5000] 1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...  
## $ Education : num [1:5000] 1 1 1 2 2 2 2 3 2 3 ...  
## $ Mortgage : num [1:5000] 0 0 0 0 0 155 0 0 104 0 ...  
## $ Personal Loan : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...  
## $ Securities Account: num [1:5000] 1 1 0 0 0 0 0 0 0 0 ...  
## $ CD Account : num [1:5000] 0 0 0 0 0 0 0 0 0 0 ...  
## $ Online : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 2 1 2 1 ...  
## $ CreditCard : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 2 1 1 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. ID = col\_double(),  
## .. Age = col\_double(),  
## .. Experience = col\_double(),  
## .. Income = col\_double(),  
## .. `ZIP Code` = col\_double(),  
## .. Family = col\_double(),  
## .. CCAvg = col\_double(),  
## .. Education = col\_double(),  
## .. Mortgage = col\_double(),  
## .. `Personal Loan` = col\_double(),  
## .. `Securities Account` = col\_double(),  
## .. `CD Account` = col\_double(),  
## .. Online = col\_double(),  
## .. CreditCard = col\_double()  
## .. )  
## - attr(\*, "problems")=<externalptr>

#Task\_1

#Data Partition

set.seed(64064)  
library(caret)  
Train\_Index = createDataPartition(UniversalBank$`Personal Loan`,p=0.60, list = FALSE) # 60% reserved for Train  
Train.df=UniversalBank[Train\_Index,]  
Validation.df=UniversalBank[-Train\_Index,]

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

## Personal Loan 0 1  
## CreditCard Online   
## 0 0 789 80  
## 1 1114 119  
## 1 0 317 39  
## 1 492 50

#Task\_B: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/(479+59)  
  
Probability

## [1] 0.1096654

#Task\_C:

#pivot table with Personal loan as row and credit card as column using training data.

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

## Personal Loan  
## CreditCard 0 1  
## 0 1903 199  
## 1 809 89

#pivot table with Personal loan as row and Online as column using training data.

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

## Personal Loan  
## Online 0 1  
## 0 1106 119  
## 1 1606 169

#pivot table for Personal loan

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

## CreditCard  
## Personal Loan 0 1  
## 0 1903 809  
## 1 199 89

#Task\_D:

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

Probability\_6 = 93/(195+93)  
  
Probability\_6

## [1] 0.3229167

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

Probability\_7 = 179/(109+179)  
  
Probability\_7

## [1] 0.6215278

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

Probability\_8 = 288/(2712+288)  
  
Probability\_8

## [1] 0.096

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

Probability\_9 = 788/(1924+788)  
  
Probability\_9

## [1] 0.2905605

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

Probability\_10 = 1631/(1631+1081)  
  
Probability\_10

## [1] 0.6014012

#vi.P(Loan = 0)

Probability\_11 = 2712/(2712+288)  
  
Probability\_11

## [1] 0.904

#Task\_E:

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

naive\_Bayes\_probability <- (Probability\_6\*Probability\_7\*Probability\_8) /   
  
 ((Probability\_6\*Probability\_7\*Probability\_8) +   
  
 (Probability\_9\*Probability\_10\*Probability\_11))  
  
  
naive\_Bayes\_probability

## [1] 0.1087106

#Task\_F:Compare this value with the one obtained from the pivot table in (B).

#Which is a more accurate estimate?

#0.1087106 in task-E is very similar to the 0.1096654 in task-B.  
  
#The difference between the exact and naive bayes methods is that  
  
#the exact approach requires the same independent variable classifications to predict,   
  
#whereas the naive bayes method does not.

#Task\_G:

# 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).

library(e1071)

library(naivebayes)

## naivebayes 0.9.7 loaded

library(mlbench)  
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')

## 0 1  
## [1,] 0.9017024 0.09829763

#The task-G value of 0.1087106 and the task-E value of 0.1087106 are identical.

#As a result, the naive bayes produces the same results as the prior approaches.