ASSIGNMENT 3

LAYA SREE GANGULA

2022-10-26

```
unibank <- read.csv("C://Users//heere//Downloads//UniversalBank (2) (1).csv")

#Converting to factor variable

unibank$Personal.Loan <- as.factor(unibank$Personal.Loan)
unibank$Online <- as.factor(unibank$Online)
unibank$CreditCard <- as.factor(unibank$CreditCard)

#Checking if there is any null variables
head(is.na(unibank))
```

```
##         ID   Age Experience Income ZIP.Code Family CCAvg Education Mortgage
## [1,] FALSE FALSE      FALSE  FALSE    FALSE  FALSE FALSE     FALSE    FALSE
## [2,] FALSE FALSE      FALSE  FALSE    FALSE  FALSE FALSE     FALSE    FALSE
## [3,] FALSE FALSE      FALSE  FALSE    FALSE  FALSE FALSE     FALSE    FALSE
## [4,] FALSE FALSE      FALSE  FALSE    FALSE  FALSE FALSE     FALSE    FALSE
## [5,] FALSE FALSE      FALSE  FALSE    FALSE  FALSE FALSE     FALSE    FALSE
## [6,] FALSE FALSE      FALSE  FALSE    FALSE  FALSE FALSE     FALSE    FALSE
##      Personal.Loan Securities.Account CD.Account Online CreditCard
## [1,]         FALSE              FALSE      FALSE  FALSE      FALSE
## [2,]         FALSE              FALSE      FALSE  FALSE      FALSE
## [3,]         FALSE              FALSE      FALSE  FALSE      FALSE
## [4,]         FALSE              FALSE      FALSE  FALSE      FALSE
## [5,]         FALSE              FALSE      FALSE  FALSE      FALSE
## [6,]         FALSE              FALSE      FALSE  FALSE      FALSE
```

```
library(ggplot2)
library(caret)
```

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

```
library(reshape2)
library(melt)
library(ISLR)
library(class)


#data partition 60:40
set.seed(300)
unibank.part <-createDataPartition (unibank$Personal.Loan, p = 0.6, list = F)
```

```r
train <-unibank[unibank.part,]
validate <- unibank[-unibank.part,]


#Normalization
norm1<-preProcess(train[,-c(10,13:14)],method=c("center","scale"))
train_n <-predict(norm1,train)
validate_n <-predict(norm1,validate)

#A. Using the training data, create a pivot table using Online as a column
#variable, CC as a row variable, and Loan as a secondary row variable.
table1 <- ftable(train_n[,c(14,10,13)])
table1
```

```
##                          Online    0    1
## CreditCard Personal.Loan
## 0          0                      780 1127
##            1                       85  114
## 1          0                      316  489
##            1                       39   50
```

```r
#B. Based on the pivot table, it is possible to determine the probability that
#this customer will accept the loan offer, which equals 50/50+489 = 0.092.

#C. Two separate pivot tables were created using the training data. In one,
#the internet (columns) are a function of the personal loan (rows), whereas the
#credit card is a function of the other.
melt1 <- melt(train_n,id=c("Personal.Loan"),variable="Online")
```

```
## Warning: attributes are not identical across measure variables; they will be
## dropped
```

```r
melt2 <- melt(train_n,id=c("Personal.Loan"), variable="CreditCard")
```

```
## Warning: attributes are not identical across measure variables; they will be
## dropped
```

```r
cast1 = dcast(melt1, Personal.Loan~Online)
```

```
## Aggregation function missing: defaulting to length
```

```r
cast2 <- dcast(melt2, Personal.Loan~CreditCard)
```

```
## Aggregation function missing: defaulting to length
```

```r
#D. Calculating the specified amounts [P(A | B) signifies "the probability that
#A will occur given B.
ftable(train_n[,c(10,13)])
```

```
##                Online    0    1
## Personal.Loan
## 0                    1096 1616
## 1                     124  164
```

```
ftable(train_n[,c(10,14)])
```

```
##                CreditCard    0    1
## Personal.Loan
## 0                        1907  805
## 1                         199   89
```

```
#ftable(train_n[,10])
#1. P(CC = 1 | Loan = 1) = (89/89+199) = 0.309
#2. P(Online = 1 | Loan = 1)  = (164/164+124) = 0.569
#3. P(Loan =# 1) = (288/288+2712) = 0.096
#4. P(CC = 1 | Loan = 0) = (805/805+1907) = 0.296
#5. P(Online = 1 | Loan = 0) = (1616/ 1616+1096) = 0.595
#6. P(Loan = 0) = (2712/ 2712+288) = 0.904


#E. Making use of the quantities computed above to compute the naive Bayes
#probability P(Loan = 1 | CC = 1, Online = 1).
#0.309 x 0.569 x 0.096) / (0.309 x 0.569 x 0.096) + (0.296 x 0.595 x 0.904) =
#0.091

#F. On comparing the naive bayes probability value i.e. 0.091 with the one
#obtained from the pivot table in (B) i.e.0.092 it can be noticed that both the
#values lie near to each other and also the naive bayes has a higher accurate
#estimate when compared to "B".

#G. implementing Naive Bayes on the data
library(naivebayes)
```

```
## naivebayes 0.9.7 loaded
```

```
naive_b<- naive_bayes(Personal.Loan~Online+CreditCard,data=train_n)
naive_b
```

```
##
## ================================== Naive Bayes ==================================
##
##  Call:
## naive_bayes.formula(formula = Personal.Loan ~ Online + CreditCard,
##     data = train_n)
##
## ---------------------------------------------------------------------------------
##
## Laplace smoothing: 0
##
## ---------------------------------------------------------------------------------
##
```

```
##  A priori probabilities:
##
##     0     1
## 0.904 0.096
##
## --------------------------------------------------------------------------------
##
##  Tables:
##
## --------------------------------------------------------------------------------
##  ::: Online (Bernoulli)
## --------------------------------------------------------------------------------
##
## Online         0         1
##      0 0.4041298 0.4305556
##      1 0.5958702 0.5694444
##
## --------------------------------------------------------------------------------
##  ::: CreditCard (Bernoulli)
## --------------------------------------------------------------------------------
##
## CreditCard         0         1
##          0 0.7031711 0.6909722
##          1 0.2968289 0.3090278
##
## --------------------------------------------------------------------------------
```

```
# For the consumer who is accepting the loan, using a credit card, and using
#online banking, the value generated by running the Naive Bayes Model is 0.096,
#which is similar to the value obtained in E.
```