Assignment 3

Rachana

2022-10-18

```r
#loading library functions
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```r
library(e1071)
```

```r
# reading csv file
UniversalBank = read.csv("UniversalBank.csv")
data=UniversalBank
head(UniversalBank)
```

```
##   ID Age Experience Income ZIP.Code Family CCAvg Education Mortgage
## 1  1  25          1     49    91107      4   1.6         1        0
## 2  2  45         19     34    90089      3   1.5         1        0
## 3  3  39         15     11    94720      1   1.0         1        0
## 4  4  35          9    100    94112      1   2.7         2        0
## 5  5  35          8     45    91330      4   1.0         2        0
## 6  6  37         13     29    92121      4   0.4         2      155
##   Personal.Loan Securities.Account CD.Account Online CreditCard
## 1             0                  1          0      0          0
## 2             0                  1          0      0          0
## 3             0                  0          0      0          0
## 4             0                  0          0      0          0
## 5             0                  0          0      0          1
## 6             0                  0          0      1          0
```

```r
#Coverting data to factor variables

data$Personal.Loan=as.factor(data$Personal.Loan)
data$Online=as.factor(data$Online)
data$CreditCard=as.factor(data$CreditCard)
```

```r
#splitting data into 60:40 ratio

set.seed(280)
```

```r
split=createDataPartition(data$Personal.Loan,p=0.6,list=FALSE,times=1)
train_data=data[split,]
valid_data=data[-split,]
head(train_data)
```

```
##   ID Age Experience Income ZIP.Code Family CCAvg Education Mortgage
## 1  1  25          1     49    91107      4   1.6         1        0
## 3  3  39         15     11    94720      1   1.0         1        0
## 6  6  37         13     29    92121      4   0.4         2      155
## 7  7  53         27     72    91711      2   1.5         2        0
## 8  8  50         24     22    93943      1   0.3         3        0
## 9  9  35         10     81    90089      3   0.6         2      104
##   Personal.Loan Securities.Account CD.Account Online CreditCard
## 1             0                  1          0      0          0
## 3             0                  0          0      0          0
## 6             0                  0          0      1          0
## 7             0                  0          0      1          0
## 8             0                  0          0      0          1
## 9             0                  0          0      1          0
```

```r
#Normalizing the data

normalized=preProcess(train_data[,-c(10,13,14 )])
train_normalized<-predict(normalized,train_data)

head(train_normalized)
```

```
##          ID        Age Experience      Income   ZIP.Code     Family      CCAvg
## 1 -1.735103 -1.7531219 -1.6441386 -0.52113839 -0.8743157  1.3987705 -0.1874602
## 3 -1.733718 -0.5366014 -0.4295570 -1.34714608  0.6804552 -1.2097945 -0.5318628
## 6 -1.731641 -0.7103900 -0.6030687 -0.95587928 -0.4379643  1.3987705 -0.8762654
## 7 -1.730948  0.6799191  0.6115129 -0.02118637 -0.6143983 -0.3402728 -0.2448607
## 8 -1.730256  0.4192361  0.3512454 -1.10803859  0.3460912 -1.2097945 -0.9336659
## 9 -1.729563 -0.8841787 -0.8633361  0.17444703 -1.3123884  0.5292489 -0.7614646
##    Education   Mortgage Personal.Loan Securities.Account CD.Account Online
## 1 -1.0648339 -0.5665921             0           2.903644 -0.2480939      0
## 3 -1.0648339 -0.5665921             0          -0.344280 -0.2480939      0
## 6  0.1187532  0.9600351             0          -0.344280 -0.2480939      1
## 7  0.1187532 -0.5665921             0          -0.344280 -0.2480939      1
## 8  1.3023403 -0.5665921             0          -0.344280 -0.2480939      0
## 9  0.1187532  0.4577255             0          -0.344280 -0.2480939      1
##   CreditCard
## 1          0
## 3          0
## 6          0
## 7          0
## 8          1
## 9          0
```

#A:Creating a pivot table with online as coloumn and credit card as row and #Personal loan as secondary row

```
table_A<-table(train_normalized$CreditCard,train_normalized$Personal.Loan,train_normalized$Online)
prop1<-prop.table(table_A)
View(prop1)
View(table_A)
```

#the probability that the customer will accept the loan offer given that he is having a bank credit card (CC = 1) and being an active user of online banking services (Online =1)

#B: P(loan=1/(cc=1,online=1))=52/(52+446) = 0.1044

#C1: Pivot table with loan as row funtion and online as coloumn function

```
table_C1<-table(train_normalized$Personal.Loan,train_normalized$Online)
View(table_C1)
```

#C2: Pivot table with loan as row function and creditcard as coloumn function

```
table_C2<-table(train_normalized$Personal.Loan,train_normalized$CreditCard)
View(table_C2)
```

#Creating table for loan=1

```
table_C3<-table(train_normalized$Personal.Loan)
View(table_C3)
prop<-prop.table(table_C3)
View(prop)
```

#D: computing the given probabilities

#i. P(CC = 1 | Loan = 1) (the proportion of credit card holders among the loan acceptors) =93/288=0.3229166

#ii. P(Online = 1 | Loan = 1)= 175/288=0.607

#iii. P(Loan = 1) (the proportion of loan acceptors) =288/3000=0.096

#iv. P(CC = 1 | Loan = 0) =764/2712=0.2817

#v. P(Online = 1 | Loan = 0) =1583/2712=0.58370

#vi. P(Loan = 0) =2712/3000=0.904

#E : Computing NAIVE BAYES PROBABLITY: P(Loan=1|CC=1,Online=1)=0.1209

#F:Values obtained from pivot table(B) is more accurate and precise than E #since they are calculated based on count or frequency directly from #the table

#G :Running Naive Bayes Probability

```
model<-naiveBayes(Personal.Loan~CreditCard+Online,data=train_normalized)
model
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
```

```
##
## A-priori probabilities:
## Y
##     0     1
## 0.904 0.096
##
## Conditional probabilities:
##    CreditCard
## Y           0         1
##   0 0.7182891 0.2817109
##   1 0.6770833 0.3229167
##
##    Online
## Y           0         1
##   0 0.4162979 0.5837021
##   1 0.3923611 0.6076389
```

comparing naviebayes probability of manual calculations error and runoff error,Naive Bayes probablity is more accurate as the manually calculated values might have calculation errors.