Assignment 3

Zachariah Alex

2022-10-17

```
#LOADING LIBRARY FUNCTIONS

library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
library(class)
library(e1071)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#READING CSV FILE

data1<-read.csv("UniversalBank.csv",header=TRUE)
head(data1)
```

```
##   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
```

```
#CONVERTING TO FACTORS

data1$Personal.Loan <- as.factor(data1$Personal.Loan)
data1$Online <- as.factor(data1$Online)
data1$CreditCard <- as.factor(data1$CreditCard)

is.factor(data1$Personal.Loan)
```

```
## [1] TRUE
```

```
is.factor(data1$Online)
```

```
## [1] TRUE
```

```
is.factor(data1$CreditCard)
```

```
## [1] TRUE
```

```
#PARTITIONING DATA TO 60:40 RATIO

set.seed(350)
data_partition<-createDataPartition(data1$Personal.Loan,p=.6,list=FALSE,times=1)

train<-data1[data_partition,]
valid<-data1[-data_partition,]
```

```
#NORMALIZING THE DATA

norm <- preProcess(train[,-c(10,13,14)], method=c("center","scale"))

train_norm <-predict(norm,train)

valid_norm<- predict(norm,valid)

head(valid_norm)
```

```
##          ID         Age  Experience      Income   ZIP.Code     Family
## 1 -1.729541 -1.77296006 -1.66106228 -0.53986834 -1.1733370  1.3986272
## 2 -1.728849 -0.02743385 -0.09125246 -0.86623492 -1.7502323  0.5302742
## 3 -1.728158 -0.55109172 -0.44009909 -1.36666367  0.8741312 -1.2064318
## 5 -1.726774 -0.90019696 -1.05058069 -0.62689943 -1.0469641  1.3986272
## 6 -1.726082 -0.72564434 -0.61452240 -0.97502378 -0.5987085  1.3986272
## 7 -1.725390  0.67077663  0.60644079 -0.03943959 -0.8310533 -0.3380788
##        CCAvg  Education   Mortgage Personal.Loan Securities.Account CD.Account
## 1 -0.1803172 -1.0559855 -0.5524148             0          2.9347083 -0.2450523
## 2 -0.2376761 -1.0559855 -0.5524148             0          2.9347083 -0.2450523
## 3 -0.5244705 -1.0559855 -0.5524148             0         -0.3406358 -0.2450523
## 5 -0.5244705  0.1336339 -0.5524148             0         -0.3406358 -0.2450523
## 6 -0.8686237  0.1336339  0.9871660             0         -0.3406358 -0.2450523
## 7 -0.2376761  0.1336339 -0.5524148             0         -0.3406358 -0.2450523
##   Online CreditCard
## 1      0          0
## 2      0          0
## 3      0          0
## 5      0          1
## 6      1          0
## 7      1          0
```

```
head(train_norm)
```

```
##            ID        Age Experience     Income    ZIP.Code     Family      CCAvg
## 4  -1.727466 -0.9001970 -0.9633690  0.5697780  0.52958080 -1.2064318  0.4506305
## 8  -1.724698  0.4089477  0.3448058 -1.1273282  0.43380938 -1.2064318 -0.9259826
## 10 -1.723314 -0.9874733 -0.9633690  2.3103998 -0.08754981 -1.2064318  4.0068808
## 11 -1.722622  1.7180924  1.6529807  0.6785669  0.86846427  1.3986272  0.2785538
## 12 -1.721930 -1.4238548 -1.3122157 -0.6268994 -1.64369365  0.5302742 -1.0407003
## 13 -1.721238  0.2343951  0.2575942  0.8743868 -0.04051414 -0.3380788  1.0815781
##    Education   Mortgage Personal.Loan Securities.Account CD.Account Online
## 4  0.1336339 -0.5524148             0         -0.3406358 -0.2450523      0
## 8  1.3232533 -0.5524148             0         -0.3406358 -0.2450523      0
## 10 1.3232533 -0.5524148             1         -0.3406358 -0.2450523      0
## 11 1.3232533 -0.5524148             0         -0.3406358 -0.2450523      0
## 12 0.1336339 -0.5524148             0         -0.3406358 -0.2450523      1
## 13 1.3232533 -0.5524148             0          2.9347083 -0.2450523      0
##    CreditCard
## 4           0
## 8           1
## 10          0
## 11          0
## 12          0
## 13          0
```

A: Creating a pivot table for the training data with Online as a column variable, CC as a row variable, and Loan as a secondary row variable

```
table_loan<-table(train_norm$CreditCard,train_norm$Personal.Loan,train_norm$Online)
View(table_loan)
```

B: The probability of loan acceptance conditional on having a bank credit card and being an active user of online banking services:

$$(Loan = 1, CC = 1, Online = 1) = (53/(468 + 53)) = 0.1017$$

C:Creating separate pivot tables for the training data with Loan (rows) as a function of Online (columns)

```
table_1<-table(train_norm$Personal.Loan,train_norm$Online)
View(table_1)
```

C: Creating separate pivot tables for the training data with Loan (rows) as a function of CC.

```
table_2<-table(train_norm$Personal.Loan,train_norm$CreditCard)
View(table_2)
```

Creating Personal Loan table

```
table_3<-table(train_norm$Personal.Loan)
proptable_3<-prop.table(table_3)
View(proptable_3)
```

D:Computing the following probabilities from the table

$$i.P(CC = 1|Loan = 1) = 84/288 = 0.2916$$

$$ii.P(Online = 1|Loan = 1) = 180/288 = 0.625$$

$$iii.P(Loan = 1) = 288/3000 = 0.096$$

$$iv.P(CC = 1|Loan = 0) = 804/(1908 + 804) = 804/2712 = 0.2964$$

$$v.P(Online = 1|Loan = 0) = 1593/(1593 + 1119) = 1593/2712 = 0.5873$$

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

E: Using the quantities computed above and computing the naive Bayes probability

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

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

$$\frac{P(CC = 1|Loan = 1) * P(Online = 1|Loan = 1).P(Loan = 1))}{P(CC = 1|Loan = 1) * P(Online = 1|Loan = 1).P(Loan = 1)) + P(CC = 1|Loan = 0)* P(Online = 1|Loan = 0) * P(Loan = 0))}$$

$$(0.2916 * 0.625 * 0.096)/((0.2916 * 0.625 * 0.096) + (0.2964 * 0.5873 * 0.904)) = 0.01014$$

$$P(Loan = 1|CC = 1, Online = 1) = 0.01014$$

F: Comparing this value with the one obtained from the pivot table in B

From pivot table (B)

$$(Loan = 1, CC = 1, Online = 1) = (53/(468 + 53)) = 0.1017$$

From computing the naive Bayes probability

$$P(Loan = 1|CC = 1, Online = 1) = 0.01014$$

We can say that the probability from the pivot table is more accurate as we are directly taking values from the frequency table while naive bayes probability calculation is based upon assumptions.

G: Running Naive bayes theorem-Using the quantities computed above to compute the naive Bayes probability

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

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

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 0.7035398 0.2964602
##   1 0.7083333 0.2916667
##
##    Online
## Y           0         1
##   0 0.4126106 0.5873894
##   1 0.3750000 0.6250000
```

naiveBayes Probability =0.096 ,which is more accurate than step E.

In step E there is a possibility of manual calculation errors and roundoff errors and assumptions.