Assignment 3

Niharika Dobanaboina

2022-10-17

Reading the CSV file:

```
UniversalBank<- read.csv("UniversalBank.csv")
head(UniversalBank)
```

```
##   ID Age Experience Income ZIPCode Family CCAvg Education Mortgage PersonalLoan
## 1  1  25          1     49   91107      4   1.6         1        0            0
## 2  2  45         19     34   90089      3   1.5         1        0            0
## 3  3  39         15     11   94720      1   1.0         1        0            0
## 4  4  35          9    100   94112      1   2.7         2        0            0
## 5  5  35          8     45   91330      4   1.0         2        0            0
## 6  6  37         13     29   92121      4   0.4         2      155            0
##   SecuritiesAccount CDAccount Online CreditCard
## 1                 1         0      0          0
## 2                 1         0      0          0
## 3                 0         0      0          0
## 4                 0         0      0          0
## 5                 0         0      0          1
## 6                 0         0      1          0
```

Converting PersonalLoan,Online,CreditCard variables to factors:

```
UniversalBank$PersonalLoan<-as.factor(UniversalBank$PersonalLoan)
UniversalBank$Online<-as.factor(UniversalBank$Online)
UniversalBank$CreditCard<-as.factor(UniversalBank$CreditCard)
```

Splitting data into Training and Validation sets:

```
set.seed(345)
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
Split_data<-createDataPartition(UniversalBank$PersonalLoan,p=.6,list=FALSE,times=1)
Training_set<-UniversalBank[Split_data,]
Validation_set<-UniversalBank[-Split_data,]
```

Normalizing training and Validation set:

```
Normalization<-preProcess(Training_set[,-c(10,13,14)],method=c("center","scale"))

Training_norm<-predict(Normalization,Training_set)
Validation_norm<-predict(Normalization,Validation_set)
```

A. Creating a pivot table for the training data with CreditCard as a row variable, PersonalLoan as a secondary row variable and Online as a column variable:

```
table_1<-table(Training_norm$CreditCard,Training_norm$PersonalLoan,Training_norm$Online)
View(table_1)

prop_table1<-prop.table(table_1)

View(prop_table1)
```

B. Probabality of a customer accepting the loan offer who owns a credit card and is also actively using online banking services:

From the Pivot table in A,

$$P(Loan = 1|CreditCard = 1, Online = 1) = 52/(52 + 495) = 0.095$$

C. Creating a Pivot table having Loan as a row variable and CreditCard as column Variable:

```
table_loan_cc <- table(Training_norm$PersonalLoan,Training_norm$CreditCard)
View(table_loan_cc)
```

C. Creating a Pivot table having Loan as a row variable and Online as column Variable:

```
table_loan_online <- with(Training_norm,table(PersonalLoan,Online))
View(table_loan_online)
```

D. (i)P(CC=1|Loan=1):

$$P(CC = 1 \cap Loan = 1)/P(Loan = 1) = 86/288 = 0.2986$$

D. (ii.)P(Online=1| Loan=1):

$$P(Online = 1 \cap Loan = 1)/P(Loan = 1) = 176/288 = 0.6111$$

D. (iii.) P(Loan=1):

```
table_loan <- table(Training_norm$PersonalLoan)
View(table_loan)

prop_loan<-prop.table(table_loan)
View(prop_loan)
```

$$P(Loan = 1) = 0.096$$

D. (iv.) P(CC=1|Loan=0):

$$P(CC = 1 \cap Loan = 0)/P(Loan = 0) = 820/271 = 0.3023$$

D. (v.) P(Online=1|Loan=0):

$$P(Online = 1 \cap Loan = 0)/P(Loan = 0) = 0.5896$$

D. (vi.) P(Loan=0):

$$P(Loan = 0) = 0.904$$

E. Computing Naive Bayes probablitity:

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

$$= P(CC = 1|Loan = 1)*P(Online = 1|Loan = 1)*P(Loan = 1)/P(CC = 1|Loan = 1*)P(Online = 1|Loan = 1)*$$
$$P(Loan = 1) + P(CC = 1|Loan = 0) * P(Online = 1|Loan = 0) * P(Loan = 0)$$
$$= 0.2986 * 0.6111 * 0.096/(0.2986 * 0.6111 * 0.096 + 0.3023 * 0.5896 * 0.904)$$
$$= 0.09802$$
$$Therefore, P(Loan = 1|CC = 1, Online = 1) = 0.09802$$

F. Comparison of P(Loan=1|CC=1,Online=1

Upon comparing the above computed Naive Bayes Probablity with the value obtained in B, it can be observed that both the values are closely same.However,the probablity obtained in B is more precise as it is calculated directly from the frequency tables based on the count.

G. Naive Bayes Probablity:

```
library(e1071)

nb_model<-naiveBayes(PersonalLoan~Online+CreditCard,data=Training_norm)
nb_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:
##    Online
## Y           0         1
##   0 0.4103982 0.5896018
##   1 0.3888889 0.6111111
##
##    CreditCard
## Y           0         1
##   0 0.6976401 0.3023599
##   1 0.7013889 0.2986111
```

Comparing this value with the probablity obtained in (E), Naive Bayes calculates probablities considering conditional independence. Therefore, the probablity calculated in (G) is more accurate.