Assignment 3

Kevin Gardner

2/28/2022

Following is the link to my GitHub account:

https://github.com/Kgardner22/64060_-kgardner

IMPORT AND PREPARE DATA:

Import the UniversalBank.csv file

```
UniversalBank <- read.table('C:/R/MyData/UniversalBank.csv', header = T, sep = ',')

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
##  Min.   :1.000   Min.   : 0.000   Min.   :1.000   Min.   :  0.0
##  1st Qu.:1.000   1st Qu.: 0.700   1st Qu.:1.000   1st Qu.:  0.0
##  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
##  Personal.Loan   Securities.Account   CD.Account         Online
##  Min.   :0.000   Min.   :0.0000     Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.0000     1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.000   Median :0.0000     Median :0.0000   Median :1.0000
##  Mean   :0.096   Mean   :0.1044     Mean   :0.0604   Mean   :0.5968
##  3rd Qu.:0.000   3rd Qu.:0.0000     3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :1.000   Max.   :1.0000     Max.   :1.0000   Max.   :1.0000
##    CreditCard
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.294
##  3rd Qu.:1.000
##  Max.   :1.000
```

Create a copy of the original data file to preserve

```
Original_File <- UniversalBank
```

Load required libraries

```
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(reshape2) #used for melt() and dcast();

## Warning: package 'reshape2' was built under R version 4.1.2

library(e1071) #used for naiveBayes();
```

We need to divide the data into training (60%) and validation (40%) sets

```
set.seed(64060)

Train_Index <- createDataPartition(UniversalBank$Personal.Loan, p=0.6, list =
FALSE) #60% for train data
Train.df <- UniversalBank[Train_Index,]
Validation.df <- UniversalBank[-Train_Index,] #Remaining 40% for validation
data
```

REQUIREMENT A:

Create a pivot table for the training data with Online as a column variable, CreditCard as a row variable, and Personal.Loan as a secondary row variable. The values inside the table should convey the count. Use functions melt() and cast(), or function table().

Pivot table created using ftable

```
Table1 <- xtabs(~ CreditCard + Online + Personal.Loan, data=Train.df)
ftable(Table1)
```

```
##                   Personal.Loan    0    1
## CreditCard Online
## 0          0                    787   76
##            1                   1144  124
## 1          0                    307   35
##            1                    477   50
```

Optional view of this same pivot table using melt();

```
Table1_Long=melt(Table1, measure.vars=c("No", "Yes"),
variable.name="Personal.Loan", value.name = "value")
Table1_Long
```

```
##   CreditCard Online Personal.Loan value
## 1          0      0             0   787
## 2          1      0             0   307
## 3          0      1             0  1144
## 4          1      1             0   477
## 5          0      0             1    76
## 6          1      0             1    35
## 7          0      1             1   124
## 8          1      1             1    50
```

Optional view of this same pivot table using dcast();

```
Table1_Wide = dcast(Table1_Long, CreditCard + Online ~ Personal.Loan,
value.var = "value" )
Table1_Wide
```

```
##   CreditCard Online    0   1
## 1          0      0  787  76
## 2          0      1 1144 124
## 3          1      0  307  35
## 4          1      1  477  50
```

REQUIREMENT B:

Looking at the pivot tables created, what is the probability that this customer will accept the loan offer (Personal.Loan=1)?

```
ftable(Table1)
```

```
##                   Personal.Loan    0    1
## CreditCard Online
## 0          0                    787   76
##            1                   1144  124
## 1          0                    307   35
##            1                    477   50
```

P(Personal.Loan=1 | CreditCard=1, Online=1)

P(50|477+50) = 0.0949 = 9.49%

ANSWER: 0.0949

REQUIREMENT C:

Create two separate pivot tables for the training data. One will have Personal.Loan (rows) as a function of Online (columns) and the other will have Personal.Loan (rows) as a function of CreditCard.

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

##            Personal.Loan
## CreditCard    0    1
##          0 1931  200
##          1  784   85

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

##       Personal.Loan
## Online    0    1
##      0 1094  111
##      1 1621  174
```

REQUIREMENT D:

Compute the following quantities [P(A|B) means "the probability of A given B"]

i. P(CreditCard=1 | Personal.Loan=1) (85/(200+85)) = (85/285) = 0.2982 #Note: I'm using the CreditCard table above

   ANSWER = 0.2982

ii. P(Online=1 | Personal.Loan=1) (174/(111+174)) = (174/285) = 0.6105 #Note: I'm using the Online table above

   ANSWER = 0.6105

iii. P(Personal.Loan=1) ((200+85)/(1931+784+200+85)) = (285/3000) = 0.095 #Note: I'm using the CreditCard table above

```
ANSWER = 0.095
```

iv. P(CreditCard=1 | Personal.Loan=0) (784/(1931+784)) = (784/2715) = 0.2888 #Note: I'm using the CreditCard table above

   ANSWER = 0.2888

v. P(Online=1 | Personal.Loan=0) (1621/(1094+1621)) = (1621/2715) = 0.5971 #Note: I'm using the Online table above

ANSWER = 0.5971

vi. P(Personal.Loan=0) ((1931+784)/(1931+784+200+85)) = (2715/3000) = 0.905
#Note: I'm using the CreditCard table above

ANSWER = 0.905

REQUIREMENT E: Use the quantities computed above to compute the naive Bayes probability P(Personal.Loan=1 | CreditCard=1, Online=1)

Using the quantities from the tables generated in requirement C, we can compute the Naive Bayes Calculations as follows:

P = (((85/285)(174/285)(285/3000)) / (((85/285)(174/285)(285/3000))+((784/2715)(1621/2715)(2715/3000))) P = (((0.2982456)(0.6105263)(0.095)) / (((0.2982456)(0.6105263)(0.095)) / ((0.2887661)(0.5970534)(0.905))) P = 0.0172982 / 0.1733281 P = 0.0998003

ANSWER = 0.0998

REQUIREMENT F: Compare the value calculated in requirement E with the one obtained from the pivot table in requirement B.

In requirement B, we calculated this as: P(Personal.Loan=1 | CreditCard=1, Online=1) (50|477+50) = 0.0949 This is the Complete (Exact) Bayes Calculation

In requirement E, we calculated this as: P = (0.0172982 / 0.1733281) = 0.0998 This is the Naive Bayes Calculation as described on page 194 of our textbook.

Which is a more accurate estimate?

ANSWER = In reading our textbook, pages 193-194, my understanding is that the answer of 0.0949 calculated in requirement B is more accurate since this is referred to as the Complete (Exact) Bayes Calculation. The Naive Bayes Calculation of 0.0998 from requirement E is an extremely close estimate of the Exact Bayes Calculation. Our Naive Bayes Calculation from requirement E is extremely close to the result of the naiveBayes() calculation in requirement G which was 0.1013226.

REQUIREMENT G: Which of the entries in this table are needed for computing P(Personal.Loan=1 | CreditCard=1, Online=1)? Run naiveBayes on the data. Examine the model output on training data and find the entry that corresponds to P(Personal.Loan=1 | CreditCard=1, Online=1). Compare this to the number you obtained in requirement E.

```
nb.model<-naiveBayes(Personal.Loan~CreditCard+Online, data=Train.df)
To_Predict=data.frame(CreditCard=1, Online=1)
predict(nb.model, To_Predict, type='raw') #type set to raw to get
probabilities;
```

```
##              0         1
## [1,] 0.8986774 0.1013226
```

These results show, given CreditCard=1 and Online=1, the probability of the personal loan being accepted (Personal.Loan=1) is 0.1013226.

The number we calculated in requirement E was 0.0998003