ljin8_FML_Assignment3

Lei Jin

2024-02-26

Load required libraries

```r
rm(list = ls()) #cleaning the environment

library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.2

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.3.2

## Loading required package: lattice
```

```r
library(class)
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.3.2
```

```r
library(class)
library(ggplot2)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.2

##
## 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
```

```r
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.3.2
```

```r
library(reshape2)
library(pander)
```

```
## Warning: package 'pander' was built under R version 4.3.2
```

Read the data

```r
data <- read.csv("C:\\Users\\leile\\OneDrive\\School-Kent\\Fundamental of machine learning\\FML ASSIGNM
```

#Understand the data

```r
str(data)
```

```
## 'data.frame':    5000 obs. of  14 variables:
##  $ ID                : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Age               : int  25 45 39 35 35 37 53 50 35 34 ...
##  $ Experience        : int  1 19 15 9 8 13 27 24 10 9 ...
##  $ Income            : int  49 34 11 100 45 29 72 22 81 180 ...
##  $ ZIP.Code          : int  91107 90089 94720 94112 91330 92121 91711 93943 90089 93023 ...
##  $ Family            : int  4 3 1 1 4 4 2 1 3 1 ...
##  $ CCAvg             : num  1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...
##  $ Education         : int  1 1 1 2 2 2 2 3 2 3 ...
##  $ Mortgage          : int  0 0 0 0 0 155 0 0 104 0 ...
##  $ Personal.Loan     : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ Securities.Account: int  1 1 0 0 0 0 0 0 0 0 ...
##  $ CD.Account        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Online            : int  0 0 0 0 0 1 1 0 1 0 ...
##  $ CreditCard        : int  0 0 0 0 1 0 0 1 0 0 ...
```

```r
summary(data)
```

```
##        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
```

#Converting the Personal loan, Online and CreditCard in to factor

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

#Partition the data into training (60%) and validation (40%) sets

```r
set.seed(123)
train_index <- createDataPartition(data$Personal.Loan, p = 0.6, list = FALSE)
train_data <- data[train_index, ]
valid_data <- data[-train_index, ]
nrow(train_data)
```

```
## [1] 3000
```

```r
nrow(valid_data)
```

```
## [1] 2000
```

#Question(A):Create 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. The values inside the table should convey the count. In R use functions melt() and cast(), or function table().

```r
attach(train_data)
melt_data <- melt(train_data, id.vars = c("CreditCard", "Personal.Loan"), measure.vars = "Online")
View(melt_data)

povit_table <- dcast(melt_data, CreditCard+Personal.Loan~variable, fun.aggregate = length)
povit_table
```

```
##   CreditCard Personal.Loan Online
## 1          0             0   1935
## 2          0             1    204
## 3          1             0    777
## 4          1             1     84
```

```r
X <- ftable(CreditCard,Personal.Loan,Online)
pandoc.table(X,style="grid", split.tables = Inf)
```

```
##
##
## +------------+---------------+--------+-----+------+
## |            |               | Online |  0  |  1   |
## +------------+---------------+--------+-----+------+
## | CreditCard | Personal.Loan |        |     |      |
## +------------+---------------+--------+-----+------+
## |     0      |       0       |        | 791 | 1144 |
## +------------+---------------+--------+-----+------+
## |            |       1       |        | 79  | 125  |
## +------------+---------------+--------+-----+------+
## |     1      |       0       |        | 310 | 467  |
## +------------+---------------+--------+-----+------+
## |            |       1       |        | 33  |  51  |
## +------------+---------------+--------+-----+------+
```

#Question(B):Consider the task of classifying a customer who owns a bank credit card and is actively using online banking services. Looking at the pivot table, what is the probability that this customer will accept the loan offer? [This is the probability of loan acceptance (Loan = 1) conditional on having a bank credit card (CC = 1) and being an active user of online banking services (Online= 1)].

```r
Loancc1 <- 51/518
Loancc1
```

```
## [1] 0.0984556
```

```r
paste("Probability of Loan acceptance given having a bank credit card and user of online services in pe
```

```
## [1] "Probability of Loan acceptance given having a bank credit card and user of online services in pe
```

#Question(C):Create two separate pivot tables for the training data. One will have Loan (rows) as a function of Online (columns) and the other will have Loan (rows) as a function of CC.

```r
Loan_online <- melt(train_data, id.vars = c("Personal.Loan"), measure.vars = "Online")
View(Loan_online)
povit_table1 <- dcast(Loan_online, Personal.Loan~variable, fun.aggregate = length)
povit_table1
```

```
##   Personal.Loan Online
## 1             0   2712
## 2             1    288
```

```r
X1 <- ftable(Personal.Loan,Online )
pandoc.table(X1,style="grid", split.tables = Inf)
```

```
##
##
```

```
## +---------------+--------+------+------+
## |               | Online |  0   |  1   |
## +---------------+--------+------+------+
## | Personal.Loan |        |      |      |
## +---------------+--------+------+------+
## |       0       |        | 1101 | 1611 |
## +---------------+--------+------+------+
## |       1       |        | 112  | 176  |
## +---------------+--------+------+------+
```

```r
CreditCard_online<- melt(train_data, id.vars = c("CreditCard"), measure.vars = "Online")
View(CreditCard_online)
povit_table2 <- dcast(CreditCard_online, CreditCard~variable, fun.aggregate = length)
povit_table2
```

```
##   CreditCard Online
## 1          0   2139
## 2          1    861
```

```r
X2 <- ftable(CreditCard,Online )
pandoc.table(X2,style="grid", split.tables = Inf)
```

```
##
##
## +------------+--------+-----+------+
## |            | Online |  0  |  1   |
## +------------+--------+-----+------+
## | CreditCard |        |     |      |
## +------------+--------+-----+------+
## |     0      |        | 870 | 1269 |
## +------------+--------+-----+------+
## |     1      |        | 343 | 518  |
## +------------+--------+-----+------+
```

#Question(D):Compute the following quantities [P(A | B) means "the probability ofA given B"]: i. P(CC = 1 | Loan = 1) (the proportion of credit card holders among the loan acceptors) ii. P(Online = 1 | Loan = 1) iii. P(Loan = 1) (the proportion of loan acceptors) iv. P(CC = 1 | Loan = 0) v. P(Online = 1 | Loan = 0) vi. P(Loan = 0)

```r
table(train_data[,c(14,10)])  # Creating a pivot table for column 14 and 10 which is credit card and pe
```

```
##           Personal.Loan
## CreditCard    0    1
##          0 1935  204
##          1  777   84
```

```r
table(train_data[,c(13,10)])  #  Creating a pivot table for column 13 and 10 which is online and person
```

```
##       Personal.Loan
## Online    0    1
##      0 1101  112
##      1 1611  176
```

```r
table(train_data[,c(10)]) # Pivot table for Personal loan.  There are 2712 and 288 from training
```

```
##
##    0    1
## 2712  288
```

P (CC = 1 | Loan = 1)

```r
CCLoan1 = 84/(84+204) # by referring the above pivot table we can get the CC= 1 and lLoan = 1 values, w
CCLoan1
```

```
## [1] 0.2916667
```

P(Online = 1 | Loan = 1)

```r
ONLoan1 =176/(176+112) # by referring the above pivot table we can get the online = 1 and Loan = 1 valu
ONLoan1
```

```
## [1] 0.6111111
```

P(Loan = 1) (the proportion of loan acceptors)

```r
Loan1 =288/(288+2712) # by referring the above pivot table we can get the Loan = 1
Loan1
```

```
## [1] 0.096
```

P(CC = 1 | Loan = 0)

```r
CCLoan0= 777/(777+1935) # by referring the above pivot table we can get the CC = 1 and Loan = 0 values ,
CCLoan0
```

```
## [1] 0.2865044
```

P(Online = 1 | Loan = 0)

```r
O1L0= 1611/(1611+1101)  # by referring the above pivot table we can get the online = 1 and Loan = 0 val
O1L0
```

```
## [1] 0.5940265
```

P(Loan=0)

```r
Loan0= 2712/(2712+288)  # by referring the above pivot table we can get the Loan = 0 values
Loan0
```

```
## [1] 0.904
```

#Question(E):Use the quantities computed above to compute the naive Bayes probability P(Loan = 1 | CC= 1, Online = 1).

```
Naive_Bay_Prob <- ((Loan1*CCLoan1*ONLoan1)/((Loan1*CCLoan1*ONLoan1)+(O1L0*CCLoan0*Loan0)))
Naive_Bay_Prob
```

```
## [1] 0.1000861
```

#Question(F):Compare this value with the one obtained from the pivot table in (B). Which is a more accurate estimate?

##9.85% is very similar to 10% from Naive Bayes method. The exact method requires the exact same independent variable classifications to make predictions, while the Naive Bayes method does not. If we want to choose one as more accurate, we might consider the value obtained directly from the data (9.85% from the pivot table) to be slightly more accurate, as it directly reflects the observed frequency in the dataset. However, both values are very close and provide reasonable estimates of the probability.

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

```
naive.train = train_data[,c(10,13,14)] # training data is from Personal loan, Creditcard and online. co
naive.test =valid_data[,c(10,13,14)] # testing set data from the same columns of data
naivebayes = naiveBayes(Personal.Loan~.,data=naive.train) # applying naivebayes algorithm to personal l
naivebayes
```

```
##
## 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 0.4059735 0.5940265
##   1 0.3888889 0.6111111
##
##    CreditCard
## Y           0         1
##   0 0.7134956 0.2865044
##   1 0.7083333 0.2916667
```

Answer: the naivebayes is the same output we got in the manual calculation method. (0.291)*(0.611)*(0.096)/((0.291)*(0.611)*(0.0
= 0.1000861 which is the same as the manual calculation.

```
#Check the probability
Aprior_Prob_N = naivebayes$apriori
Loan_Online_N = naivebayes$tables$Online
Loan_CC_N = naivebayes$tables$CreditCard
```

```r
#probability Calculation from Naive Bayes Model.
L_CC1 = Loan_CC_N[2,2] #0.2916666
L_ON1 = Loan_Online_N[2,2] #0.611111
L1 = Aprior_Prob_N[1]
L2 = Aprior_Prob_N[2]
L = L2/(L1+L2) #0.096
L_CC2 = Loan_CC_N[1,2] #0.2865044
L_ON2 = Loan_Online_N[1,2]  #0.5940265
L_not = 1-L #0.904

naive_bayes_Final <- ((L_CC1*L_ON1*L)/((L_CC1*L_ON1*L)+(L_CC2*L_ON2*L_not)))
naive_bayes_Final
```

```
##         1
## 0.1000861
```

```r
paste("naive Bayes probability by using Naive bayes function is", round(naive_bayes_Final,4)*100,"%")
```

```
## [1] "naive Bayes probability by using Naive bayes function is 10.01 %"
```

#Again, the naivebayes is the same output we got in the manual calculation method.