Naive Bayes

Nisha Chaurasia

2023-02-27

#The purpose of this assignment is to use Naive Bayes for classification.Will be using 3 different methods to compare #1.Easy method #2.Using the Naive Bayers equation #3.Using the Naive Bayers function in R

##loading required library

```
rm(list = ls()) #cleaning the environment
library(readr)
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(knitr)
library(class)
library(ggplot2)
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
```

```
library(e1071)
library(reshape2)
library(tinytex)
library(pivottabler)
library(gt)
library(glue)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(pander)
```

##Import Data "UniversalBank.csv"

```
library(readr)
Bankdata1 <- read.csv("C:/Users/Chaur/OneDrive/Desktop/FML/Assignment_2_KNN/UniversalBank.csv")
head(Bankdata1)
```

```
##   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
```

##Understand the bank data structure

```
str(Bankdata1)
```

```
## '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 ...
```

```
summary(Bankdata1)
```

```
##        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

```
Bankdata1$Personal_Loan = as.factor(Bankdata1$Personal_Loan)
Bankdata1$Online = as.factor(Bankdata1$Online)
Bankdata1$CreditCard = as.factor(Bankdata1$CreditCard)
```

##Partitioning the data into training (60%) and validation (40%) sets Also showed the summary statistics of both train and Validation data set.

```
set.seed(70)
train_index = createDataPartition(Bankdata1$Personal_Loan, p= .6, list=FALSE)
Validation_index <- setdiff(row.names(Bankdata1), train_index)
train_df <- Bankdata1[train_index, ]
nrow(train_df)
```

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

```
summary(train_df)
```

```
##        ID            Age          Experience        Income
##  Min.   :   1   Min.   :23.00   Min.   :-3.00   Min.   :  8.00
##  1st Qu.:1224   1st Qu.:35.00   1st Qu.:10.00   1st Qu.: 39.00
##  Median :2503   Median :45.00   Median :20.00   Median : 64.00
##  Mean   :2502   Mean   :45.33   Mean   :20.09   Mean   : 74.62
##  3rd Qu.:3768   3rd Qu.:55.00   3rd Qu.:30.00   3rd Qu.: 99.00
```

```
##  Max.   :4999   Max.   :67.00   Max.   :42.00   Max.   :224.00
##     ZIP_Code         Family          CCAvg          Education
##  Min.   :90005   Min.   :1.000   Min.   : 0.000   Min.   :1.000
##  1st Qu.:91910   1st Qu.:1.000   1st Qu.: 0.700   1st Qu.:1.000
##  Median :93555   Median :2.000   Median : 1.600   Median :2.000
##  Mean   :93179   Mean   :2.394   Mean   : 1.965   Mean   :1.875
##  3rd Qu.:94609   3rd Qu.:3.000   3rd Qu.: 2.600   3rd Qu.:3.000
##  Max.   :96651   Max.   :4.000   Max.   :10.000   Max.   :3.000
##     Mortgage      Personal_Loan Securities.Account   CD.Account    Online
##  Min.   :  0.00   0:2712        Min.   :0.0000     Min.   :0.000   0:1228
##  1st Qu.:  0.00   1: 288        1st Qu.:0.0000     1st Qu.:0.000   1:1772
##  Median :  0.00                 Median :0.0000     Median :0.000
##  Mean   : 56.98                 Mean   :0.1027     Mean   :0.058
##  3rd Qu.:100.00                 3rd Qu.:0.0000     3rd Qu.:0.000
##  Max.   :612.00                 Max.   :1.0000     Max.   :1.000
##  CreditCard
##  0:2140
##  1: 860
##
##
##
##
```

```r
Validation_df <- Bankdata1[Validation_index, ]
nrow(Validation_df)
```

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

```r
summary(Validation_df)
```

```
##        ID            Age          Experience        Income
##  Min.   :   3   Min.   :23.00   Min.   :-3.00   Min.   :  8.00
##  1st Qu.:1279   1st Qu.:35.00   1st Qu.:10.00   1st Qu.: 38.00
##  Median :2496   Median :45.00   Median :20.00   Median : 63.00
##  Mean   :2498   Mean   :45.35   Mean   :20.13   Mean   : 72.51
##  3rd Qu.:3717   3rd Qu.:55.00   3rd Qu.:29.25   3rd Qu.: 95.00
##  Max.   :5000   Max.   :67.00   Max.   :43.00   Max.   :218.00
##     ZIP_Code         Family        CCAvg          Education        Mortgage
##  Min.   : 9307   Min.   :1.0   Min.   :0.000   Min.   :1.000   Min.   :  0.00
##  1st Qu.:91950   1st Qu.:1.0   1st Qu.:0.670   1st Qu.:1.000   1st Qu.:  0.00
##  Median :93308   Median :2.0   Median :1.500   Median :2.000   Median :  0.00
##  Mean   :93114   Mean   :2.4   Mean   :1.898   Mean   :1.891   Mean   : 55.78
##  3rd Qu.:94596   3rd Qu.:3.0   3rd Qu.:2.500   3rd Qu.:3.000   3rd Qu.:102.00
##  Max.   :96651   Max.   :4.0   Max.   :9.000   Max.   :3.000   Max.   :635.00
##  Personal_Loan Securities.Account   CD.Account    Online   CreditCard
##  0:1808        Min.   :0.000      Min.   :0.000   0: 788   0:1390
##  1: 192        1st Qu.:0.000      1st Qu.:0.000   1:1212   1: 610
##                Median :0.000      Median :0.000
##                Mean   :0.107      Mean   :0.064
##                3rd Qu.:0.000      3rd Qu.:0.000
##                Max.   :1.000      Max.   :1.000
```

##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().

```
attach(train_df)
melted_bank = melt(train_df,id.vars = c("CreditCard","Personal_Loan"), measure.vars = "Online")
View(melted_bank)
pivot_table <- dcast(melted_bank, CreditCard + Personal_Loan ~ variable, fun.aggregate=length)
pivot_table
```

```
##   CreditCard Personal_Loan Online
## 1          0             0   1937
## 2          0             1    203
## 3          1             0    775
## 4          1             1     85
```

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

```
##
##
## +------------+---------------+--------+-----+------+
## |            |               | Online |  0  |  1   |
## +------------+---------------+--------+-----+------+
## | CreditCard | Personal_Loan |        |     |      |
## +------------+---------------+--------+-----+------+
## |     0      |       0       |        | 799 | 1138 |
## +------------+---------------+--------+-----+------+
## |            |       1       |        | 83  | 120  |
## +------------+---------------+--------+-----+------+
## |     1      |       0       |        | 309 | 466  |
## +------------+---------------+--------+-----+------+
## |            |       1       |        | 37  |  48  |
## +------------+---------------+--------+-----+------+
```

##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)].

```
P_acceptance <- (48/514)
P_acceptance
```

```
## [1] 0.09338521
```

```
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 <- addmargins(table(train_df[,c(13,10)]))
pandoc.table(Loan_online,style="grid", split.tables = Inf)
```

```
##
##
## +---------+------+-----+------+
## |    |  0   |  1  | Sum  |
## +=========+======+=====+======+
## |  **0**  | 1108 | 120 | 1228 |
## +---------+------+-----+------+
## |  **1**  | 1604 | 168 | 1772 |
## +---------+------+-----+------+
## | **Sum** | 2712 | 288 | 3000 |
## +---------+------+-----+------+
```

```r
Loan_CC <- addmargins(table(train_df[,c(14,10)]))
pandoc.table(Loan_CC,style="grid", split.tables = Inf)
```

```
##
##
## +---------+------+-----+------+
## |    |  0   |  1  | Sum  |
## +=========+======+=====+======+
## |  **0**  | 1937 | 203 | 2140 |
## +---------+------+-----+------+
## |  **1**  | 775  | 85  | 860  |
## +---------+------+-----+------+
## | **Sum** | 2712 | 288 | 3000 |
## +---------+------+-----+------+
```

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

```r
##P (CC = 1 | Loan = 1) (the proportion of credit card holders among the loan acceptors)
count_A1 <- Loan_CC[2, 2] #85
count_A2 <- Loan_CC[3, 2] #288
A = (count_A1/count_A2)
paste("The proportion of credit card holders among the loan acceptors is", round(A*100,2),"%")
```

```
## [1] "The proportion of credit card holders among the loan acceptors is 29.51 %"
```

```r
##P(Online=1|Loan=1)
count_B1 <- Loan_online[2, 2] #168
count_B2 <- Loan_online[3, 2] #288
B = (count_B1/count_B2)
paste("The proportion of online active among the loan acceptors is", round(B*100,2),"%")
```

```
## [1] "The proportion of online active among the loan acceptors is 58.33 %"
```

```r
#P (Loan = 1) (the proportion of loan acceptors)
count_C1 <- Loan_online[3, 2] #288
count_C2 <- Loan_online[3, 3] #3000
C = (count_C1/count_C2)
paste("the proportion of loan acceptors is", round(C*100,2),"%")
```

```
## [1] "the proportion of loan acceptors is 9.6 %"
```

```r
#P(CC=1|Loan=0)
count_D1 <- Loan_CC[2, 1] #775
count_D2 <- Loan_CC[3, 1] #2712
D = (count_D1/count_D2)
paste("The proportion of credit card holders among the non-loan acceptors is", round(D*100,2),"%")
```

```
## [1] "The proportion of credit card holders among the non-loan acceptors is 28.58 %"
```

```r
#P(Online=1|Loan=0)
count_E1 <- Loan_online[2, 1] #1604
count_E2 <- Loan_online[3, 1] #2712
E = (count_E1/count_E2)
paste("The proportion of Online active among the non-loan acceptors is", round(E*100,2),"%")
```

```
## [1] "The proportion of Online active among the non-loan acceptors is 59.14 %"
```

```r
#P(Loan=0)
count_F1 <- Loan_online[3,1] #2712
count_F2 <- Loan_online[3,3] #3000
F = (count_F1/count_F2)
paste("The proportion of non- Loan acceptors", round(F*100,2),"%")
```

```
## [1] "The proportion of non- Loan acceptors 90.4 %"
```

##e. Use the quantities computed above to compute the naive Ba1 probability P(Loan = 1 | CC = 1, Online = 1).

$$P(L=1|CC=1,Onl=1) = \frac{(P(CC=1|L=1)*P(Onl=1|L=1))*P(L=1)}{(P(CC=1|L=1)*P(Onl=1|L=1))*P(L=1)+(P(CC=1|L=0)*P(Onl=1|L=0))$$

```r
Naive_Bay_Prob <- ((A*B*C)/((A*B*C)+(D*E*F)))
Naive_Bay_Prob
```

```
## [1] 0.09761391
```

```r
paste("naive Bayer probability is", round(Naive_Bay_Prob,4)*100,"%")
```

```
## [1] "naive Bayer probability is 9.76 %"
```

##f. Compare this value with the one obtained from the pivot table in (b). Which is a more accurate estimate?

##9.34% are very similar to the 9.76%.The exact method requires the exact same independent variable classifications to make predictions, while the Naive Bayes method does not.Which means exact method may be more rigid and precise in its predictions, but may also be limited by the requirement for exact classification of independent variables. In contrast, the Naive Bayes method may be more flexible in its predictions, but may also be less precise due to the simplifying assumption of independence among features

##Question(g). Which of the entries in this table are needed for computing P (Loan = 1 | CC = 1, Online = 1)? In R, 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).

```
#We only need 3 entries i.e Personal_loan, CreditCard and Online to predict P.
naive_train = train_df[,c(10,13:14)]
naive_Validation = Validation_df[,c(10,13:14)]
naivebayes_M = naiveBayes(Personal_Loan~.,data=naive_train)
naivebayes_M
```

```
##
## 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 0.4085546 0.5914454
##   1 0.4166667 0.5833333
##
##    CreditCard
## Y           0         1
##   0 0.7142330 0.2857670
##   1 0.7048611 0.2951389
```

```
Aprior_Prob_N = naivebayes_M$apriori
Loan_Online_N = naivebayes_M$tables$Online
Loan_CC_N = naivebayes_M$tables$CreditCard

#probability Calculation from Naive Bayes Model.

L_CC1 = Loan_CC_N[2,2] #0.2951389
L_ON1 = Loan_Online_N[2,2] #0.5833333
L1 = Aprior_Prob_N[1]
L2 = Aprior_Prob_N[2]
L = L2/(L1+L2) #0.096
L_CC2 = Loan_CC_N[1,2] #0.285767
L_ON2 = Loan_Online_N[1,2]  #0.5914454
```

```
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
## 0.09761391
```

```
paste("naive Ba1 probability by using Naive bayers function is", round(naive_bayes_Final,4)*100,"%")
```

```
## [1] "naive Ba1 probability by using Naive bayers function is 9.76 %"
```

```
detach(train_df)

#We got the same exact output we receive in Previous method.i.e in question (e): because the joint and
```