ML.Assignement.3(Naive Bayes)

2022-10-09

```
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(e1071)
library(class)
library(tidyverse)

## ── Attaching packages
## ───────────────────────────────────────
## tidyverse 1.3.2 ──

## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.1      ✔ stringr 1.4.1
## ✔ readr   2.1.3      ✔ forcats 0.5.2
## ✔ purrr   0.3.5
## ── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ✖ purrr::lift()   masks caret::lift()

#Importing data set for the current environment.
getwd()

## [1] "/Users/thupiliabhinav/Desktop/ML/Assignment 3"

setwd("/Users/thupiliabhinav/Desktop/ML/Assignment 3")
bankdata <- read.csv("UniversalBank.csv")
str(bankdata)

## '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 ...
```

```
#Converting Personal.Loan, Online, CreditCard to factors.

bankdata$Online <- as.factor(bankdata$Online)
is.factor(bankdata$Online)
```

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

```
bankdata$CreditCard <- as.factor(bankdata$CreditCard)
is.factor(bankdata$CreditCard)
```

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

```
bankdata$Personal.Loan <- as.factor(bankdata$Personal.Loan)
is.factor(bankdata$Personal.Loan)
```

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

```
str(bankdata)
```

```
## '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     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...
##  $ 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            : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 2 1 2 1 ...
##  $ CreditCard        : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 2 1 1 ...
```

```
#Data partition into Training and Validation.
set.seed(123)

bankdata1 <- createDataPartition(bankdata$Personal.Loan, p=0.60, list =
FALSE)
train_bank <- bankdata[bankdata1,]
validate_bank <- bankdata[-bankdata1,]

# Data Normalization.
```

```
norm_data <- preProcess(train_bank[,-c(10,13,14)], method = c("center",
"scale"))
predict_tdata <- predict(norm_data, train_bank)
predict_vdata <- predict(norm_data, validate_bank)

#A. Creating Pivot Table with Online as column variable and CC,
Personal.Loan as row variables.

pivot_bank<- ftable(predict_tdata$Personal.Loan, predict_tdata$Online,
predict_tdata$CreditCard, dnn=c('Personal.loan','CreditCard', 'Online'))
pivot_bank
```

```
##                            Online    0    1
## Personal.loan CreditCard                   
## 0             0                    791  310
##               1                   1144  467
## 1             0                     79   33
##               1                    125   51
```

```
#B.Probability of Loan Acceptance (Loan=1) conditional on CC=1 and
Online=1.

prob_bank<-pivot_bank[4,2]/(pivot_bank[2,2]+pivot_bank[4,2])
prob_bank
```

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

```
#C.probability for personal loan and Online.
pivot_bank1<
ftable(predict_tdata$Personal.Loan,predict_tdata$Online,dnn=c('Personal.loan'
,'Online'))
pivot_bank1
```

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

```
#C.probability for personal loan and CreditCard.
pivot_bank2<- ftable(predict_tdata$Personal.Loan,predict_tdata$CreditCard,
dnn=c('Personal.loan','CreditCard'))
pivot_bank2
```

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

```
#D.(i).P(CC=1 | Loan= 1)
prob_bank1<- pivot_bank2[2,2]/(pivot_bank2[2,2]+pivot_bank2[2,1])
prob_bank1
```

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

```r
#D.(ii).P(Online=1 | Loan=1)
prob_bank2 <- pivot_bank1[2,2]/(pivot_bank1[2,2]+pivot_bank1[2,1])
prob_bank2
```

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

```r
#D.(iii).P(Loan=1)
prob_bank3 <- ftable(predict_tdata[,10])
prob_bank3
```

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

```r
prob_bank_3 <- prob_bank3[1,2]/(prob_bank3[1,2]+prob_bank3[1,1])
prob_bank_3
```

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

```r
#D.(iv).P(CC=1 | Loan=0)
prob_bank4 <- pivot_bank2[1,2]/(pivot_bank2[1,2]+pivot_bank2[1,1])
prob_bank4
```

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

```r
#D.(v).P(Online=1 | Loan=0)
prob_bank5 <- pivot_bank1[1,2]/(pivot_bank1[1,2]+pivot_bank1[1,1])
prob_bank5
```

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

```r
#D.(vi).P(Loan=0)
prob_bank6 <- ftable(predict_tdata[,10])
prob_bank6
```

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

```r
prob_bank_6 <- prob_bank6[1,1]/(prob_bank6[1,1]+prob_bank6[1,2])
prob_bank_6
```

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

```r
#E.Computing Naive Bayes using conditional probabilities derived from
D.
nb <-
(prob_bank1*prob_bank2*prob_bank_3)/(prob_bank1*prob_bank2*prob_bank_3+prob_b
ank4*prob_bank5*prob_bank_6)
nb
```

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

#F. Comparison of values between answers derived from B. and E. The probability derived from Bayes probability i.e., B. is 0.0984556 and the probability derived from Naive's Bayes i.e., is 0.1000. The comparison between Bayes and Naive bayes shows that Naive Bayes has a higher probability.

```
#G.Using Naive Bayes directly applied to the data.
nb_model <-naiveBayes(Personal.Loan~Online+CreditCard, data = predict_tdata)
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 0.4059735 0.5940265
##   1 0.3888889 0.6111111
##
##    CreditCard
## Y           0         1
##   0 0.7134956 0.2865044
##   1 0.7083333 0.2916667
```

#From the above table we can observe that for P(Loan=1| CC=1, Online=1), following values are to be considered - 0.096(Loan = 1), 0.6111(Online=1), 0.2916(CC=1). The values derived from the table to calculate Naive Bayes will be same as the value derived in E for calculating Naive Bayes but was calculated from Bayes probability.