Assignment_3

Krishna Kumar Tavva - 811283461

2023-03-01

Loading required packages & calling libraries:

```
#install.packages("reshape2")
library(class) #classification
library(caret) #creating predictive models
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
library(readr) #Load Data
library(e1071) #naive Bayes classifier
library(reshape2) #restructure and aggregate data
library(dplyr) #data manipulation
```

```
##
## 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(ISLR) #collection of data-sets
library(gmodels) #fits a model
library(pROC) #to plot a graph showing the performance of a classification model
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following object is masked from 'package:gmodels':
##
##     ci
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

Load data

```
ub <- data.frame(read.csv("E:/Fundamentals of Machine Learning/Module 5/UniversalBank.csv"))
head(ub)
```

```
##   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
```

checking for na values

```
any(is.na.data.frame(ub))
```

```
## [1] FALSE
```

Data factoring

```
is.factor(ub$Personal.Loan)
```

```
## [1] FALSE
```

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

```
## [1] FALSE
```

```
ub$Online <- as.factor(ub$Online)
ub$CreditCard <- as.factor(ub$CreditCard)
is.factor(ub$CreditCard)
```

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

Data Partition

```
set.seed(1)
Index_Train <- createDataPartition(ub$Personal.Loan,p=.6, list=F)
Train <- ub[Index_Train,]
Validate <- ub[-Index_Train,]
```

#Data Normalization

```
norm_model <- preProcess(Train[,-c(10,13:14)],
method=c("center","scale"))
Train_norm <- predict(norm_model,Train)
Validate_norm <- predict(norm_model,Validate)
```

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

```
table <- ftable(CreditCard= Train_norm$CreditCard, Loan=Train_norm$Personal.Loan, Online= Train_norm$Onl
table
```

```
##                  Online    0    1
## CreditCard Loan
## 0          0             780 1126
##            1              77  120
## 1          0             303  503
##            1              39   52
```

B.The probability of customer accepting loan and using credit card plus being an online banking user = 52/(52+503) = 0.09369

C.Create two separate pivot tables for the training data

```
#Loan (rows) as a function of Online (columns)
table_1 <- table(Loan=Train_norm$Personal.Loan, Online= Train_norm$Online)
table_1
```

```
##     Online
## Loan    0    1
##    0 1083 1629
##    1  116  172
```

```
#Loan (rows) as a function of CC(columns)
table_2 <- table(Loan=Train_norm$Personal.Loan, CreditCard= Train_norm$CreditCard)
table_2
```

```
##     CreditCard
## Loan    0    1
##    0 1906  806
##    1  197   91
```

D.Compute the following quantities [P(A | B) i.e. the probability of A given B]

```r
ftable(Train_norm[,c(10,13)])
```

```
##               Online    0    1
## Personal.Loan
## 0                    1083 1629
## 1                     116  172
```

```r
ftable(Train_norm[,c(10,14)])
```

```
##               CreditCard    0    1
## Personal.Loan
## 0                        1906  806
## 1                         197   91
```

```r
ftable(Train_norm[,10])
```

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

1. P(CC = 1 | Loan = 1) = 91/(91+197) = 0.31597
2. P(Online= 1 | Loan= 1) = 172/(172+116) = 0.5972
3. P(Loan = 1) = 288/(288+2712) = 0.096
4. P(CC= 1 | Loan= 0) = 806/(806+1906) = 0.29719
5. P(Online=1 |Loan=0) = 1629/(1629+1083) = 0.5940
6. P(Loan = 0) = 2712/(2712+288) = 0.904

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

(0.31597 x 0.5972 x 0.096) / (0.31597 x 0.5972 x 0.096) + (0.29719 x 0.5940 x 0.904) = 0.1068

F. By comparing the value obtained above by using the Naive Bayes probability i.e. 0.1068 to the value obtained in step B i.e. 0.09369 we get to see that both the values are near values, but Naive Bayes has a bit higher probability when compared to that with the direct calculation.

G. Run the Naive Bayes Model

```
naive <- naiveBayes(Personal.Loan~Online+CreditCard,data=Train_norm)
naive
```

```
##
## 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 0.3993363 0.6006637
##   1 0.4027778 0.5972222
##
##    CreditCard
## Y           0         1
##   0 0.7028024 0.2971976
##   1 0.6840278 0.3159722
```

The value obtained by running the Naive Bayes Model for the customer who is accepting the loan and using credit card plus being an online banking user is 0.096 which is near to the value obtained in E

Predicting the Naive Bayes model over the validation data and also looking at the AUC Value and ROC Curve

```
pred_labels <- predict(naive,Validate_norm,type = "raw")
head(pred_labels)
```

```
##              0          1
## [1,] 0.9055932 0.09440679
## [2,] 0.8977658 0.10223419
## [3,] 0.9055932 0.09440679
## [4,] 0.9055932 0.09440679
## [5,] 0.9055932 0.09440679
## [6,] 0.9055932 0.09440679
```

```
roc(Validate_norm$Online,pred_labels[,2])
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls > cases

##
## Call:
## roc.default(response = Validate_norm$Online, predictor = pred_labels[,     2])
##
## Data: pred_labels[, 2] in 817 controls (Validate_norm$Online 0) > 1183 cases (Validate_norm$Online 1
## Area under the curve: 0.8068
```

In general, an AUC of 0.5 suggests no discrimination (i.e., ability to diagnose patients with and without the disease or condition based on the test), 0.7 to 0.8 is considered acceptable, 0.8 to 0.9 is considered excellent, and more than 0.9 is considered outstanding.

Area under the curve: 0.8068

#clearing the all loaded work from Environment

```
rm(list = ls(all.names = TRUE)) #will clear all objects includes hidden objects.
gc() #free up memory and report the memory usage.
```

```
##           used  (Mb) gc trigger  (Mb) max used  (Mb)
## Ncells 2323207 124.1    4487365 239.7  4145562 221.4
## Vcells 3844121  29.4    8388608  64.0  8381803  64.0
```