MIS-64060-001(A3)

Kiran Kour

2022-10-10

#Importing required packages

```
#install.packages("reshape")
#install.packages("reshape2")
#install.packages("melt")
#install.packages("naivebayes")
#install.packages("pROC")
```

```
library(readr)
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(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
library(class)
library(melt)
library(reshape)
```

```
##
## Attaching package: 'reshape'

## The following object is masked from 'package:class':
##
##     condense
## The following object is masked from 'package:dplyr':
##
##     rename
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following objects are masked from 'package:reshape':
##
##     colsplit, melt, recast
```

```
library(ggplot2)
library(ISLR)
library(naivebayes)
```

```
## naivebayes 0.9.7 loaded
```

```
library(e1071)
library(pROC)
```

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

##
## Attaching package: 'pROC'

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

#Importing the dataset

```
universalbank<- read.csv("UniversalBank.csv")
head(universalbank,n=5)
```

```
##   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
##   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
```

#Checking for missing values using is.na()

```
bank <- is.na.data.frame("universalbank")
```

#Converting the data type of categorical variables to factor

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

#Data Partition and Normalization

```
set.seed(333)
Train_Index<- createDataPartition(universalbank$Personal.Loan, p=0.6, list=FALSE)
Train <-universalbank[Train_Index,]
Valid <-universalbank[-Train_Index,]
```

```
Model_norm <- preProcess(Train[,-c(10,13:14)],method = c("center", "scale"))
Train_norm <- predict(Model_norm,Train)
Valid_norm<- predict(Model_norm,Valid)
```

Part A: Creating pivot table for Training data

```
Table.OCP <- table(Train_norm$Personal.Loan, Train_norm$Online, Train_norm$CreditCard, dnn=c("Personal L
Table.OCP
```

```
## , , Credit Card = 0
##
##               Online
## Personal Loan    0    1
##             0  786 1120
##             1   73  135
##
## , , Credit Card = 1
##
##               Online
## Personal Loan    0    1
##             0  305  501
##             1   31   49
```

Part B: Computing P(Loan | Online & CC)

As we look that the pivot table created in part A out of the total 550 records where of active online banking users with credit cards, 49 had accepted a personal loan, so

$$\mathbf{P}(\text{Loan} = 1 \mid \text{CC} = 1\text{and Online} = 1) = \frac{49}{550} = 0.089$$

.

```
# Computing P(loan | Online & CC)
Table.OCP[2,2,2] / (Table.OCP[2,2,2] + Table.OCP[1,2,2])
```

```
## [1] 0.08909091
```

Part C: Creating two separate pivot tables for 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.

```
Table_Online <- table(Train_norm$Personal.Loan, Train_norm$Online, dnn=c("Personal Loan", "Online"))
Table_Online
```

```
##              Online
## Personal Loan    0    1
##             0 1091 1621
##             1  104  184
```

```
Table_CreditCard <- table(Train_norm$Personal.Loan, Train$CreditCard, dnn=c("Personal Loan", "Credit Ca
Table_CreditCard
```

```
##              Credit Card
## Personal Loan    0    1
##             0 1906  806
##             1  208   80
```

Part D : Computing the following quantities:

i.)

$$\mathbf{P}(\mathrm{CC} = 1 \mid \mathrm{Loan} = 1) = 80/80+208$$

```
prob_CCL <- Table_CreditCard[2,2] / (Table_CreditCard[2,2] + Table_CreditCard[2,1])
prob_CCL
```

```
## [1] 0.2777778
```

ii.)

$$\mathbf{P}(\mathrm{Online} = 1 \mid \mathrm{Loan} = 1) = 184/184+104$$

```
prob_OL <- Table_Online[2,2] / (Table_Online[2,2] + Table_Online[2,1])
prob_OL
```

```
## [1] 0.6388889
```

iii.)

$$\mathbf{P}(\mathrm{Loan} = 1) = 288/288+2712$$

```
prob_Loan <- sum(Train_norm$Personal.Loan==1) / length(Train_norm$Personal.Loan)
prob_Loan
```

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

iv.)

$$\mathbf{P}(\mathrm{CC} = 1 \mid \mathrm{Loan} = 0) = 806/806+1906$$

```
prob_CCNL <-Table_CreditCard[1,2] / (Table_CreditCard[1,2] + Table_CreditCard[1,1])
prob_CCNL
```

```
## [1] 0.2971976
```

v.)

$$\mathbf{P}(\mathrm{Online} = 1 \mid \mathrm{Loan} = 0) = 1621/1621+1091$$

```
prob_ONL <- Table_Online[1,2] / (Table_Online[1,2] + Table_Online[1,1])
prob_ONL
```

```
## [1] 0.5977139
```

vi.)

$$\mathbf{P}(\mathrm{Loan} = 0) = 2712/2712+288$$

```
prob_NL <- sum(Train_norm$Personal.Loan==0) / length(Train_norm$Personal.Loan)
prob_NL
```

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

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

$\mathbf{P}(\mathrm{Loan} = 1 \,|\, \mathrm{CC} = 1,\ \mathrm{Online} = 1) = (0.6388 \text{ x } 0.2777 \text{ x } 0.096)\ /\ (0.6388 \text{ x } 0.2777 \text{ x } 0.096 + 0.5977 \text{ x } 0.2972 \text{ x } 0.904) = 0.095$

```
(prob_OL * prob_CCL * prob_Loan) / (prob_OL * prob_CCL * prob_Loan + prob_ONL * prob_CCNL * prob_NL)
```

```
## [1] 0.09591693
```

Part F : Comparing the value obtained from part Naive bayes probability with the one obtained from the pivot table in (B).

Using the Naive Bayes classifier, we get a higher value for P(Loan = 1 | CC = 1, Online = 1) than with the direct computation obtained in part B. Interestingly, in part D we got the value of $\mathbf{P}(\mathrm{Loan} = 1)$ as 0.096 and also in Naive bayes classifer we got the value as 0.096. So the Naive Bayes approach suggests that the probability a person will accept the loan is independent of whether that person is an online user with a bank-issued credit card.

Part G : Running Naive Bayes on the data

```
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 0.4022861 0.5977139
##   1 0.3611111 0.6388889
##
##    CreditCard
## Y           0         1
##   0 0.7028024 0.2971976
##   1 0.7222222 0.2777778
```

The value that is obtained for 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) from running Naive bayes is 0.09591693 , which is equal to the value derived from part E.

AUC Value and ROC Curve

```
Predicted_labels <-predict(naive,Valid_norm, type = "raw")
head(Predicted_labels)
```

```
##              0          1
## [1,] 0.9107805 0.08921953
## [2,] 0.9107805 0.08921953
## [3,] 0.8955384 0.10446160
## [4,] 0.8955384 0.10446160
## [5,] 0.9181923 0.08180773
## [6,] 0.9107805 0.08921953
```

```
roc(Valid_norm$Online, Predicted_labels[,2])
```

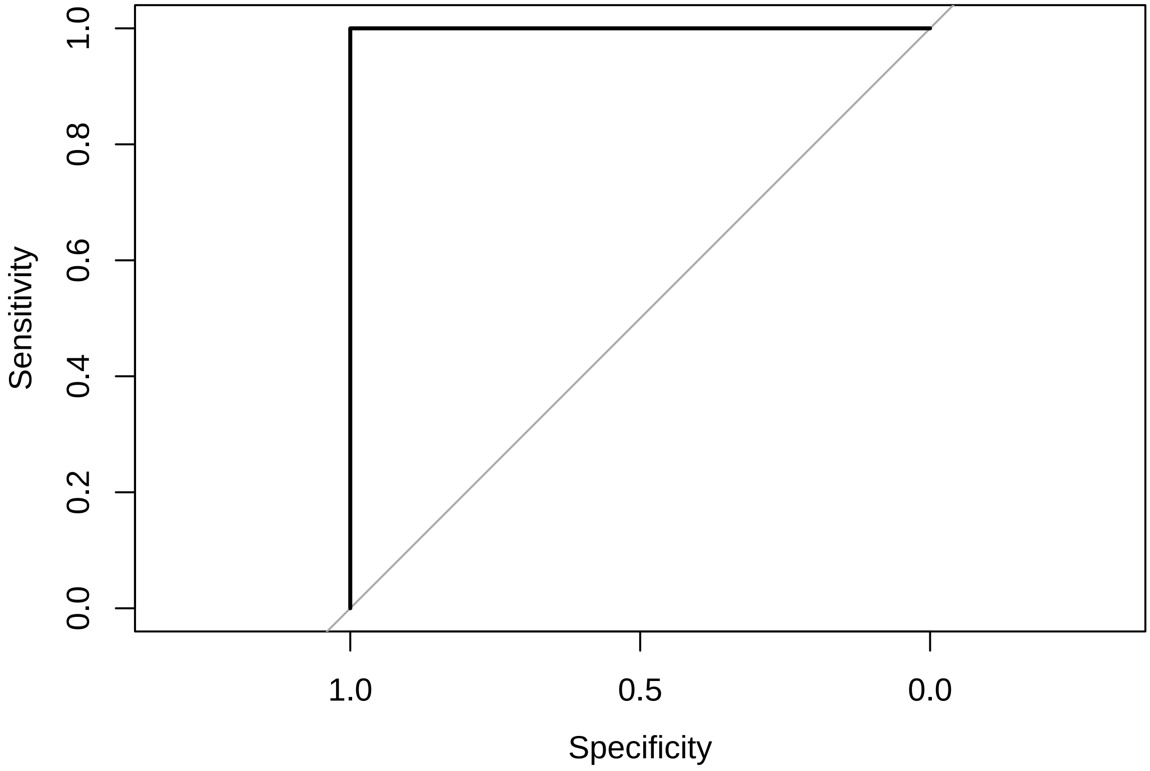
```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
##
## Call:
## roc.default(response = Valid_norm$Online, predictor = Predicted_labels[,     2])
##
## Data: Predicted_labels[, 2] in 821 controls (Valid_norm$Online 0) < 1179 cases (Valid_norm$Online 1)
## Area under the curve: 1
```

```
plot.roc(Valid_norm$Online,Predicted_labels[,2])
```

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

```
## Setting direction: controls < cases
```


Sensitivity
Specificity