# FML Assignment 3

Nikitha Chigurupati

2022-10-16

#IMPORTING THE DATASET

```
Df <- read.csv("C:/Users/Nikitha/Downloads/UniversalBank.csv")
```

#CONVERTING THE PREDICTOR ATTRIBUTE TO FACTORS

```
Df$Personal.Loan <- as.factor(Df$Personal.Loan)
Df$Online <- as.factor(Df$Online)
Df$CreditCard <- as.factor(Df$CreditCard)
```

#CHECKING FOR NULL VALUES

```
sum(is.na(Df))
```

```
## [1] 0
```

#LOADING THE LIBRARIES

```
library(class)
```

```
## Warning: package 'class' was built under R version 4.1.3
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.3
```

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

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

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

```
library(e1071)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
##
## 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(ggplot2)
library(reshape)
```

```
## Warning: package 'reshape' was built under R version 4.1.3
```

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

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

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

```r
library(melt)
```

```
## Warning: package 'melt' was built under R version 4.1.3
```

```r
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 4.1.3
```

```r
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.1.3
```

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

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

```r
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.1.3
```

```r
library(naivebayes)
```

```
## Warning: package 'naivebayes' was built under R version 4.1.3
```

```
## naivebayes 0.9.7 loaded
```

```r
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 4.1.3
```

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

```
##
## Attaching package: 'pROC'
```

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

#DATA PARTITION TO 60:40

```r
set.seed(123)
datapart <- createDataPartition(Df$Personal.Loan,p=.6, list=F)
Train <- Df[datapart,]
Validate <- Df[-datapart,]
```

#DATA NORMALIZATION

```r
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

```
tab1<- ftable(Train_norm[,c(14,10,13)])
tab1
```

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

#B. 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)] = 51/(51+467) = 0.0984.

#C. Creating two separate pivot tables for the training data. One having Loan (rows) as a function of Online (columns) and the other having Loan (rows) as a function of CC

```
melt1 = melt(Train, id=c("CreditCard","Personal.Loan"), variable = "Online")
```

```
## Warning: attributes are not identical across measure variables; they will be
## dropped
```

```
castbank = dcast(melt1, CreditCard+Personal.Loan~Online)
```

```
## Aggregation function missing: defaulting to length
```

```
castbank[,c(1:2,14)]
```

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

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

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

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

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

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

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

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

#1. P(CC = 1 | Loan = 1) = (84/84+204) = 0.291 #2. P(Online = 1 | Loan = 1) = (176/176+112) = 0.611 #3. P(Loan = 1) = (288/288+2712) = 0.096 #4. P(CC = 1 | Loan = 0) = (777/777+1935) = 0.286 #5. P(Online = 1 | Loan = 0) = (1611/ 1611+1101) = 0.595 #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)

```
#(0.291 x 0.611 x 0.096) / (0.271 x 0.611 x 0.096) + (0.286 x 0.595 x 0.904) = 0.1000
```

#F. We can see that the values attained in steps b, 0.0984, and a, 0.1000, are practically identical, although the probability with Naive Bayes is slightly higher.

#G. Run the Naive Bayes Model on the data

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

```
##
## ================================== Naive Bayes ==================================
##
##  Call:
## naive_bayes.formula(formula = Personal.Loan ~ Online + CreditCard,
##     data = Train_norm)
##
## ---------------------------------------------------------------------------------
##
## Laplace smoothing: 0
##
## ---------------------------------------------------------------------------------
##
##  A priori probabilities:
##
##     0     1
## 0.904 0.096
##
## ---------------------------------------------------------------------------------
##
##  Tables:
##
## ---------------------------------------------------------------------------------
##  ::: Online (Bernoulli)
## ---------------------------------------------------------------------------------
##
## Online         0         1
##      0 0.4059735 0.3888889
##      1 0.5940265 0.6111111
##
## ---------------------------------------------------------------------------------
##  ::: CreditCard (Bernoulli)
## ---------------------------------------------------------------------------------
##
## CreditCard         0         1
##          0 0.7134956 0.7083333
##          1 0.2865044 0.2916667
##
## ---------------------------------------------------------------------------------
```

```
#Naive Bayes Model results for the consumer taking the loan, using their credit card, and using
 online banking are 0.1000, which is equivalent to the result in E.
```

#Examining the AUC value and ROC curve

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

```
predlab <- predict(Naive,Validate_norm,type = "raw")
head(predlab)
```

```
##              0          1
## [1,] 0.9082737 0.09172629
## [2,] 0.9021538 0.09784623
## [3,] 0.9061594 0.09384060
## [4,] 0.9082737 0.09172629
## [5,] 0.9082737 0.09172629
## [6,] 0.8999139 0.10008606
```

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

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

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

```
##
## Call:
## roc.default(response = Validate_norm$Online, predictor = predlab[,     2])
##
## Data: predlab[, 2] in 803 controls (Validate_norm$Online 0) < 1197 cases (Validate_norm$Onlin
e 1).
## Area under the curve: 1
```

```
plot.roc(Validate_norm$Online,predlab[,2])
```

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