# Assignment 3

Praneeth Simha

2023-03-05

```
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("tidyr")
library("e1071")
library("reshape2")
```

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

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library("pROC")
```

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

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

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

```
library("naivebayes")
```

```
## naivebayes 0.9.7 loaded
```

```
Unibank <- read.csv("C:/Users/ADMIN/Downloads/UniversalBank.csv")
summary(Unibank)
```

```
##        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 predictors to factors
head(Unibank)
```

```
##   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
```

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

```
#Partition the data into training (60%) and validation (40%) sets
set.seed(2022)
training_data_index = createDataPartition(Unibank$Personal.Loan, p=.6, list = F) #60% trainin
g data
training_data_df = Unibank[training_data_index,]
validation_data_df = Unibank[-training_data_index,] #Validation Data
```

```
#Pivot tables
melt_Unibank = melt(training_data_df, id=c("CreditCard","Personal.Loan"), variable = "Onlin
e") #function melt()
```

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

```
dcast_Unibank = dcast(melt_Unibank, CreditCard+Personal.Loan~Online) #function cast()
```

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

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

```
##   CreditCard Personal.Loan Online
## 1          0             0   1904
## 2          0             1    204
## 3          1             0    808
## 4          1             1     84
```

```
# CreditCard, Personal.Loan, Online DF
```

```
#Pivot tables having loan rows as function of online coloumns and remaining having loan rows as function of credit card
loan_melt_Unibank = melt(training_data_df, id=c("Personal.Loan"), variable = "Online")
```

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

```
cc_melt_Unibank = melt(training_data_df, id=c("CreditCard"), variable = "Online")
```

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

```
dcast_loan_Unibank = dcast(loan_melt_Unibank, Personal.Loan~Online)
```

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

```
dcast_cc_Unibank = dcast(cc_melt_Unibank, CreditCard~Online)
```

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

```
dcast_loan_Unibank[,c(1,13)]
```

```
##   Personal.Loan Online
## 1             0   2712
## 2             1    288
```

```
dcast_cc_Unibank[,c(1,14)]
```

```
##   CreditCard Online
## 1          0   2108
## 2          1    892
```

```
#Calculate the following quantities: P(A | B), or the likelihood that A will occur given B.
table(training_data_df[,c(14,10)])
```

```
##           Personal.Loan
## CreditCard    0    1
##          0 1904  204
##          1  808   84
```

```
table(training_data_df[,c(13,10)])
```

```
##       Personal.Loan
## Online    0    1
##      0 1123  120
##      1 1589  168
```

```
table(training_data_df[c(10)])
```

```
## Personal.Loan
##    0    1
## 2712  288
```

```
#Running the naivebayes model on the data
train.naive.bayes = training_data_df[,c(10,13:14)]
naive.bayes = naiveBayes(Personal.Loan~., data=train.naive.bayes)
naive.bayes
```

```
##
## 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 0.4140855 0.5859145
##   1 0.4166667 0.5833333
##
##    CreditCard
## Y           0         1
##   0 0.7020649 0.2979351
##   1 0.7083333 0.2916667
```

```
#Looking at the ROC curve and AUC value
Naive <- naiveBayes(Personal.Loan~Online+CreditCard,data=training_data_df)
 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 0.4140855 0.5859145
##   1 0.4166667 0.5833333
##
##    CreditCard
## Y           0         1
##   0 0.7020649 0.2979351
##   1 0.7083333 0.2916667
```

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

```
##              0          1
## [1,] 0.9026813 0.09731871
## [2,] 0.9026813 0.09731871
## [3,] 0.9036110 0.09638897
## [4,] 0.9052982 0.09470181
## [5,] 0.9036110 0.09638897
## [6,] 0.9026813 0.09731871
```

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

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

```
## Setting direction: controls > cases
```

```
##
## Call:
## roc.default(response = training_data_df$Online, predictor = predlab[,     2])
##
## Data: predlab[, 2] in 1243 controls (training_data_df$Online 0) > 1757 cases (training_dat
a_df$Online 1).
## Area under the curve: 0.7886
```

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

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