

FML Assignment 3

Shivani Pitla

2022-10-16

```
#installation of all the required packages
```

```
library("class")  
library("caret")
```

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

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

```
library("e1071")  
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("ggplot2")  
library("gmodels")  
library("melt")  
library("reshape")
```

```
##
```

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

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

```
##
```

```
##      rename
```

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

```
##
```

```
##      condense
```

```

library("reshape2")

##
## Attaching package: 'reshape2'

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

library("readr")
library("ISLR")
library("pROC")

## 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

#importing the dataset

universalbank <- read.csv("C:/Users/shiva/Downloads/UniversalBank (1).csv")

#normalizing and cleaning the data

#Converting the predictor attributes to factors
universalbank$Personal.Loan <- as.factor(universalbank$Personal.Loan)
universalbank$Online <- as.factor(universalbank$Online)
universalbank$CreditCard <- as.factor(universalbank$CreditCard)
#checking for na values
test.na <- is.na.data.frame(universalbank)
#Data Partition
set.seed(310)
data_part <- createDataPartition(universalbank$Personal.Loan,p=.6, list=F)
Train <- universalbank[data_part,]
Validate <- universalbank[-data_part,]
#Data Normalization
normmodel_ub <- preProcess(Train[, -c(10,13:14)],
                           method=c("center", "scale"))
Trainnorm_ub <- predict(normmodel_ub, Train)
Validatenorm_ub <- predict(normmodel_ub, 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

```
table1_ub <- ftable(Trainnorm_ub[,c(14,10,13)])
table1_ub
```

```
##               Online    0    1
## CreditCard Personal.Loan
## 0           0           791 1115
##           1           75  134
## 1           0          323  483
##           1           35   44
```

B. The probability of the customers accepting loan who owns a bank credit card and is actively using online banking services = $51/(51+467) = 0.0984$

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_ub <- melt(Trainnorm_ub,id=c("Personal.Loan"),variable="Online")
```

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

```
melt2_ub <- melt(Trainnorm_ub,id=c("Personal.Loan"), variable="CreditCard")
```

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

```
castt1_ub <- dcast(melt1_ub, Personal.Loan~Online)
```

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

```
castt2_ub <- dcast(melt2_ub, Personal.Loan~CreditCard)
```

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

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

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

```
##               Online    0    1
## Personal.Loan
## 0           1114 1598
## 1           110  178
```

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

```
##          CreditCard    0    1
## Personal.Loan
## 0                1906  806
## 1                209   79
```

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

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

1. $P(CC = 1 \mid Loan = 1) = 79/(79+209) = \mathbf{0.2743}$
2. $P(Online = 1 \mid Loan = 1) = 178/(178+110) = \mathbf{0.6180}$
3. $P(Loan = 1) = 288/(288+2712) = \mathbf{0.096}$
4. $P(CC = 1 \mid Loan = 0) = 806/(806+1906) = \mathbf{0.2972}$
5. $P(Online = 1 \mid Loan = 0) = 1598/(1598+1114) = \mathbf{0.5892}$
6. $P(Loan = 0) = 2712/(2712+288) = \mathbf{0.904}$

E. Use the quantities computed above to compute the naive Bayes probability $P(Loan = 1 \mid CC = 1, Online = 1)$

$$(0.2743 \times 0.6180 \times 0.096) / (0.2743 \times 0.6180 \times 0.096) + (0.2972 \times 0.5892 \times 0.904) = \mathbf{0.1000}$$

F. When we compare the value acquired in step b, which is 0.0984, to the value achieved in step a, which is 0.1000, we can see that both numbers are practically identical, although Naive Bayes has a slightly greater probability than that with the direct calculation.

G. Run the Naive Bayes Model on the data

```
naive <- naiveBayes(Personal.Loan~Online+CreditCard,data=Trainnorm_ub)
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 0.4107670 0.5892330
## 1 0.3819444 0.6180556
##
## CreditCard
## Y 0 1
## 0 0.7028024 0.2971976
## 1 0.7256944 0.2743056
```

When the Naive Bayes Model is run for the consumer taking the loan, using a credit card, and using online banking, the result is 0.1000, which is equivalent to the result in E.

Using the validation data to predict the Naive Bayes model while also examining the AUC value and ROC curve

```
predlabels_ub <- predict(naive,Validatenorm_ub,type = "raw")
head(predlabels_ub)
```

```
## 0 1
## [1,] 0.9074743 0.09252571
## [2,] 0.8968469 0.10315307
## [3,] 0.8968469 0.10315307
## [4,] 0.9074743 0.09252571
## [5,] 0.8968469 0.10315307
## [6,] 0.9074743 0.09252571
```

```
roc(Validatenorm_ub$Online,predlabels_ub[,2])
```

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

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

```
##
```

```
## Call:
```

```
## roc.default(response = Validatenorm_ub$Online, predictor = predlabels_ub[, 2])
```

```
##
```

```
## Data: predlabels_ub[, 2] in 792 controls (Validatenorm_ub$Online 0) < 1208 cases (Validatenorm_ub$Online 1)
```

```
## Area under the curve: 1
```

```
plot.roc(Validatenorm_ub$Online,predlabels_ub[,2])
```

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

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

