FML Assignment 3

Pavan Kumar Mekala

2024-03-10

## Importing the dataset UniversalBank(1).csv file into Rstudio and checked it by using head and tail funcitons and converting Online, Personal.Loan, CreditCard into factor value by using as.factor() function

UniversalBank<-read.csv("UniversalBank (1).csv")  
head(UniversalBank)

## 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

tail(UniversalBank)

## ID Age Experience Income ZIP.Code Family CCAvg Education Mortgage  
## 4995 4995 64 40 75 94588 3 2.0 3 0  
## 4996 4996 29 3 40 92697 1 1.9 3 0  
## 4997 4997 30 4 15 92037 4 0.4 1 85  
## 4998 4998 63 39 24 93023 2 0.3 3 0  
## 4999 4999 65 40 49 90034 3 0.5 2 0  
## 5000 5000 28 4 83 92612 3 0.8 1 0  
## Personal.Loan Securities.Account CD.Account Online CreditCard  
## 4995 0 0 0 1 0  
## 4996 0 0 0 1 0  
## 4997 0 0 0 1 0  
## 4998 0 0 0 0 0  
## 4999 0 0 0 1 0  
## 5000 0 0 0 1 1

UniversalBank$Personal.Loan <- as.factor(UniversalBank$Personal.Loan)  
UniversalBank$Online <- as.factor(UniversalBank$Online)  
UniversalBank$CreditCard <- as.factor(UniversalBank$CreditCard)  
UniversalBank <-UniversalBank[,c("Personal.Loan","Online","CreditCard")]  
head(UniversalBank)

## Personal.Loan Online CreditCard  
## 1 0 0 0  
## 2 0 0 0  
## 3 0 0 0  
## 4 0 0 0  
## 5 0 0 1  
## 6 0 1 0

## Loading Packages

library(ISLR)  
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(knitr)

## Partitioning the data into training (60%) and validation (40%) sets and adding set.seed(123) for reproducibility and checked them by using dim function

set.seed(123)  
Training\_Index <- createDataPartition(UniversalBank$Personal.Loan, p=0.6 , list = FALSE)  
Training <- UniversalBank[Training\_Index,]  
Validation <- UniversalBank[-Training\_Index,]  
dim(Training)

## [1] 3000 3

dim(Validation)

## [1] 2000 3

##A. Creating a Pivot table for the training data and checked them by using print() and dim() function

P\_Table<-ftable(Training$Online,Training$CreditCard,Training$Personal.Loan, row.vars = c(3,2),dnn=c('Online','CreditCard','Personal.loan'))  
P\_Table

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

##B. Probability of loan acceptance (Loan = 1) conditional on having a bank credit card (CC = 1) and checked it by using print() function

P\_Loan\_Acceptance <- P\_Table[4,2] / (P\_Table[2,2]+P\_Table[4,2])  
print(P\_Loan\_Acceptance)

## [1] 0.0984556

##C. Creating two seperate pivot tables for the training data by using ftable() function and checked them by using print() function

P\_table\_online <- ftable(Training$Online, Training$Personal.Loan, row.vars = c(2))  
P\_table\_cc <- ftable(Training$CreditCard, Training$Personal.Loan, row.vars = c(2))  
print(P\_table\_online)

## 0 1  
##   
## 0 1101 1611  
## 1 112 176

print(P\_table\_cc)

## 0 1  
##   
## 0 1935 777  
## 1 204 84

##D. Computing the following quantities [P(A | B) means “the probability of A given B”] under different conditions; ## 1. P(CC = 1 | Loan = 1) ## 2. P(Online = 1 | Loan = 1) ## 3. P(Loan = 1)(the proportion on loan acceptors) ## 4. P(CC = 1 | Loan = 0) ## 5. P(Online = 1 | Loan = 0) ## 6. P(Loan = 0) and checked it by using print() function

P\_CC\_Loan <- P\_table\_cc[2,2] / (P\_table\_cc[2,1]+P\_table\_cc[2,2])  
print(P\_CC\_Loan)

## [1] 0.2916667

P\_Online\_Loan <- P\_table\_online[2,2] / (P\_table\_online[2,1] + P\_table\_online[2,2])  
print(P\_Online\_Loan)

## [1] 0.6111111

P\_Loan <- sum(Training$Personal.Loan == 1) / nrow(Training)  
print(P\_Loan)

## [1] 0.096

P\_CC\_Not\_Loan <- P\_table\_cc[1,2]/(P\_table\_cc[1,1]+P\_table\_cc[1,2])  
print(P\_CC\_Not\_Loan)

## [1] 0.2865044

P\_Online\_Not\_Loan <- P\_table\_online[1,2] / (P\_table\_online[1,1]+P\_table\_online[1,2])  
print(P\_Online\_Not\_Loan)

## [1] 0.5940265

P\_Not\_Loan <- sum(Training$Personal.Loan == 0) / nrow(Training)  
print(P\_Not\_Loan)

## [1] 0.904

##E. Naive Bayes Probability P(Loan = 1 | CC = 1, Online = 1)

P\_Loan\_CC\_Online <- (P\_CC\_Loan \* P\_Online\_Loan \* P\_Loan) / ((P\_CC\_Loan \* P\_Online\_Loan \* P\_Loan) + (P\_CC\_Not\_Loan \* P\_Online\_Not\_Loan \* P\_Not\_Loan))  
print(P\_Loan\_CC\_Online)

## [1] 0.1000861

##F. Comparing the values obtained from Pivot tables in B

P\_Loan\_CC\_Online - P\_Loan\_Acceptance

## [1] 0.001630457

print("We persume conditional independence in Naive Bayes")

## [1] "We persume conditional independence in Naive Bayes"

print("We can see there is an increase of 0.001630457")

## [1] "We can see there is an increase of 0.001630457"

##G. Comparing Numbers with D

library(e1071)  
Probability\_Loan <- naiveBayes(Personal.Loan ~ ., data = Training)  
loan\_probabilities <- c(Probability\_Loan$apriori[1] / sum(Probability\_Loan$apriori), Probability\_Loan$apriori[2] / sum(Probability\_Loan$apriori))  
Probability\_Loan$tables

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

print("As we can see the individual probabilities are matching with D, so the Naive Bayes probability is 0.09975504")

## [1] "As we can see the individual probabilities are matching with D, so the Naive Bayes probability is 0.09975504"

## ROC and plotting

library(pROC)

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

##   
## Attaching package: 'pROC'

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

Predict\_Probability<-predict(Probability\_Loan, newdata = Validation, type = "raw")  
roc(Validation$Personal.Loan, Predict\_Probability[, 2])

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

## Setting direction: controls < cases

##   
## Call:  
## roc.default(response = Validation$Personal.Loan, predictor = Predict\_Probability[, 2])  
##   
## Data: Predict\_Probability[, 2] in 1808 controls (Validation$Personal.Loan 0) < 192 cases (Validation$Personal.Loan 1).  
## Area under the curve: 0.4986

plot.roc(Validation$Personal.Loan, Predict\_Probability[,2])

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

