3.R

ribakhan

2022-03-06

```
UniversalBank=read.csv("~/Desktop/Fundamentals of Machine Learning/UniversalBank.csv")#calling the need

#Calling the required libraries
library(caret)
```

```
## Loading required package: ggplot2

## Warning in register(): Can't find generic `scale_type` in package ggplot2 to
## register S3 method.

## Loading required package: lattice
```

```
library(ggplot2)
library(lattice)
library(class)
library(ISLR)
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(reshape2)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'

## The following object is masked from 'package:reshape2':
##
##     smiths
```

```r
library(e1071)

summary(UniversalBank)
```

```
##        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   :93152
##  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
```

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

#set seed
set.seed(64060)
#divide data set into training(60%) and validation(40%)sets
train.index <- sample(row.names(UniversalBank), 0.6*dim(UniversalBank)[1])
test.index <- setdiff(row.names(UniversalBank), train.index)
train.df <- UniversalBank[train.index,]
test.df <- UniversalBank[test.index,]
train <- UniversalBank[train.index,]
test = UniversalBank[train.index,]


#TASK A
#Create a pivot table
#Here i am using the (melt) function
melted.UniversalBank = melt(train,id=c("CreditCard","Personal.Loan"),variable= "Online")
```

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

```
recast.UniversalBank=dcast(melted.UniversalBank,CreditCard+Personal.Loan~Online)
```

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

```
recast.UniversalBank[,c(1:2,14)]
```

```
##   CreditCard Personal.Loan Online
## 1          0             0   1931
## 2          0             1    200
## 3          1             0    784
## 4          1             1     85
```

```
#TASK B
#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)
#Answer : 85/3000 = 2.8%

#TASK C
#create two separate pivot table for training data
melted.UniversalBankc1 = melt(train,id=c("Personal.Loan"),variable = "Online")
```

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

```
melted.UniversalBankc2 = melt(train,id=c("CreditCard"),variable = "Online")
```

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

```
recast.UniversalBankc1=dcast(melted.UniversalBankc1,Personal.Loan~Online)
```

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

```
recast.UniversalBankc2=dcast(melted.UniversalBankc2,CreditCard~Online)
```

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

```
Loanline=recast.UniversalBankc1[,c(1,13)]
LoanCC = recast.UniversalBankc2[,c(1,14)]

#looking at the table
Loanline
```

```
##   Personal.Loan Online
## 1             0   2715
## 2             1    285
```

```r
summary(Loanline)
```

```
##  Personal.Loan     Online
##  0:1           Min.   : 285.0
##  1:1           1st Qu.: 892.5
##                Median :1500.0
##                Mean   :1500.0
##                3rd Qu.:2107.5
##                Max.   :2715.0
```

```r
LoanCC
```

```
##   CreditCard Online
## 1          0   2131
## 2          1    869
```

```r
summary(LoanCC)
```

```
##  CreditCard     Online
##  0:1        Min.   : 869
##  1:1        1st Qu.:1184
##             Median :1500
##             Mean   :1500
##             3rd Qu.:1816
##             Max.   :2131
```

```r
#Task D
#Compute the following quantities [P(A | B) means "the probability ofA given B"]:

table(train[,c(14,10)])
```

```
##           Personal.Loan
## CreditCard    0    1
##          0 1931  200
##          1  784   85
```

```r
table(train[,c(13,10)])
```

```
##       Personal.Loan
## Online    0    1
##      0 1094  111
##      1 1621  174
```

```r
table(train[,c(10)])
```

```
##
##    0    1
## 2715  285
```

```
# i.) P(CC = 1 | Loan = 1) (the proportion of credit card holders among the loan acceptors)
# Answer : 85/(85+200) = 0.29%

# ii.) P(Online = 1 | Loan = 1)
# Answer : 174/(174+111) = 0.61%

# iii.) P(Loan = 1) (the proportion of loan acceptors)
Loanline
```

```
##   Personal.Loan Online
## 1             0   2715
## 2             1    285
```

```
# Answer: 285/(285+ 2715) = 0.095%

# iv.) P(CC = 1 | Loan = 0)
#Answer : 784/(784+ 1931) = 0.28%

# v.) P(Online = 1 | Loan = 0)
# Answer : 1621/(1621+1094) = 0.59%

#vi. P(Loan = 0)
# Answer : 2715/(2715+285) = 0.90%

#TASK E
#Use the quantities computed above to compute the naive Bayes probability
((85/(85+200))* (174/(174+111))* (285/(285+2715)))/((85/(85+200))* (174/(174+111))* (285/(285+2715))) +
```

```
## [1] 1.15603
```

```
# Answer : 1.15603



#TASK F
#Compare this value with the one obtained from the pivot table in (B). Which is a more accurate estimat

# Answer : The calucalted answer is nearly the same.
# the difference between the exact method and the naive-baise method is the exact
#method would need the the exact same independent variable classifications to predict, where the naive


# TASK G
# Run Navie Bayes for the data

naive.train = train.df[,c(10,13:14)]
naive.test = test.df[,c(10,13:14)]
naivebayes = naiveBayes(Personal.Loan~.,data=naive.train)
naivebayes
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
```

```
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##     0     1
## 0.905 0.095
##
## Conditional probabilities:
##    Online
## Y           0         1
##   0 0.4029466 0.5970534
##   1 0.3894737 0.6105263
##
##    CreditCard
## Y           0         1
##   0 0.7112339 0.2887661
##   1 0.7017544 0.2982456
```

```
summary(naivebayes)
```

```
##           Length Class  Mode
## apriori   2      table  numeric
## tables    2      -none- list
## levels    2      -none- character
## isnumeric 2      -none- logical
## call      4      -none- call
```

```
#the naive bayes is the exact same output we recieved in the previous methods. (.280)(.603)(.09)/(.280.
(0.29 * 0.48 * 0.09)/(0.29 * 0.48 * 0.09) + (0.28 * 0.49 * 0.90)
```

```
## [1] 1.12348
```

```
# Answer = 1.12348
```