Assignment 3

2023-10-15

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
UniversalBank<- read.csv("C:\\Users\\13308\\OneDrive\\Documents\\UniversalBank.csv")
#A

library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.2.3
```

```
library(class)
```

```
## Warning: package 'class' was built under R version 4.2.3
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.2.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.2.3

## Loading required package: lattice
```

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

str(UniversalBank)
```

```
## 'data.frame':    5000 obs. of  14 variables:
##  $ ID                : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Age               : int  25 45 39 35 35 37 53 50 35 34 ...
##  $ Experience        : int  1 19 15 9 8 13 27 24 10 9 ...
##  $ Income            : int  49 34 11 100 45 29 72 22 81 180 ...
##  $ ZIP.Code          : int  91107 90089 94720 94112 91330 92121 91711 93943 90089 93023 ...
```

```
##  $ Family            : int  4 3 1 1 4 4 2 1 3 1 ...
##  $ CCAvg             : num  1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...
##  $ Education         : int  1 1 1 2 2 2 2 3 2 3 ...
##  $ Mortgage          : int  0 0 0 0 0 155 0 0 104 0 ...
##  $ Personal.Loan     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...
##  $ Securities.Account: int  1 1 0 0 0 0 0 0 0 0 ...
##  $ CD.Account        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Online            : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 2 1 2 1 ...
##  $ CreditCard        : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 2 1 1 ...
```

```
selected.var <- c(10, 13, 14)
train.index <- sample(c(1:dim(UniversalBank)[1]), dim(UniversalBank)[1]*0.6)
train.df <- UniversalBank[train.index, selected.var]
valid.df <- UniversalBank [-train.index, selected.var]

sum(is.na(train.df$CreditCard))
```

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

```
sum(is.na(train.df$Personal.Loan))
```

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

```
str(UniversalBank$Online)
```

```
##  Factor w/ 2 levels "0","1": 1 1 1 1 1 2 2 1 2 1 ...
```

```
pivot_table <- table(train.df$CreditCard, train.df$Personal.Loan, train.df$Online)
print(pivot_table)
```

```
## , ,  = 0
##
##
##        0    1
##   0  750   70
##   1  336   36
##
## , ,  = 1
##
##
##        0    1
##   0 1146  128
##   1  484   50
```

```
combined_table <- addmargins(pivot_table)
print(combined_table)
```

```
## , ,  = 0
##
##
```

```
##          0    1  Sum
##   0    750   70  820
##   1    336   36  372
##   Sum 1086  106 1192
##
## , ,  = 1
##
##
##          0    1  Sum
##   0   1146  128 1274
##   1    484   50  534
##   Sum 1630  178 1808
##
## , ,  = Sum
##
##
##          0    1  Sum
##   0   1896  198 2094
##   1    820   86  906
##   Sum 2716  284 3000
```

```
#B

#I am calculating the number of Online=1/CC=1/PL=1, divded by the total number of Online=1/CC=1.
48 / 508
```

```
## [1] 0.09448819
```

```
#C

LoanofOnline<- table(train.df$Personal.Loan, train.df$Online)
print(LoanofOnline)
```

```
##
##        0    1
##   0 1086 1630
##   1  106  178
```

```
LoanofCreditCard <- table(train.df$Personal.Loan, train.df$CreditCard)
print(LoanofCreditCard)
```

```
##
##        0    1
##   0 1896  820
##   1  198   86
```

```
#Di

i <- (87 / (87 + 206))
print(i)
```

```
## [1] 0.2969283
```

```r
#Dii

ii<- (174 / (174 + 119))
print(ii)
```

```
## [1] 0.5938567
```

```r
#Diii

293 / 3000
```

```
## [1] 0.09766667
```

```r
#Div

784 / (784 + 1923)
```

```
## [1] 0.2896195
```

```r
#Dv

1606 / (1606 + 1101)
```

```
## [1] 0.5932767
```

```r
#Dvi

(3000 - 293) / 3000
```

```
## [1] 0.9023333
```

```r
#E


print(i*ii)
```

```
## [1] 0.1763329
```

```r
#F

#Overall, the value in (b) is more accurate as we are using the counts of customers that are
#both CreditCard=1 and Online=1 to compute the probability of Loan = 1. The value
#in (e) is using counts of customers that may be one but not the other.

#G

UniversalBank.nb <- naiveBayes(train.df$Personal.Loan ~ ., data = train.df)
UniversalBank.nb
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##          0          1
## 0.90533333 0.09466667
##
## Conditional probabilities:
##    Online
## Y           0         1
##   0 0.3998527 0.6001473
##   1 0.3732394 0.6267606
##
##    CreditCard
## Y           0         1
##   0 0.6980854 0.3019146
##   1 0.6971831 0.3028169
```

```
#It would be the values in the bottom right of each pivot table, and the
#product of both is the same value we received in (E).
```