ASSIGNMENT3

rajiv

3/6/2022

```r
ULbank = read.csv("C:\\Users\\rdevi\\Downloads\\UniversalBank.csv")
View(ULbank)

#loading required libraries
```

```r
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```r
library(ISLR)
library(e1071)

#converting variables
```

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

```r
set.seed(64060)

#spliting data into training 60% and validatio 40%

Train_Index = createDataPartition(ULbank$Personal.Loan, p = 0.6, list = FALSE)
Train.df = ULbank[Train_Index,]
Validation.df = ULbank[-Train_Index]

#Task-A. Create a pivot table for the training data with Online as a column variable, CC as a row varia
```

```r
mytable = xtabs(~ CreditCard+Personal.Loan+Online,data = Train.df)
ftable(mytable)
```

```
##                          Online    0    1
## CreditCard Personal.Loan
## 0          0                     772 1152
##            1                      75  120
## 1          0                     309  479
##            1                      34   59
```

```r
#Task-B.Consider the task of classifying a customer who owns a bank credit card and is actively using o
```

```r
59/(479+59)
```

```
## [1] 0.1096654
```

```r
#Task-C.Create two separate pivot tables for the training data. One will have Loan (rows) as a unction
```

```r
table(Personal.Loan = Train.df$Personal.Loan, CreditCard = Train.df$CreditCard)
```

```
##              CreditCard
## Personal.Loan    0    1
##             0 1924  788
##             1  195   93
```

```r
table(Personal.Loan = Train.df$Personal.Loan, Online = Train.df$Online)
```

```
##              Online
## Personal.Loan    0    1
##             0 1081 1631
##             1  109  179
```

```r
table(Personal.Loan = Train.df$Personal.Loan)
```

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

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

#1. P(CC = 1 | Loan = 1)
```

```r
93/(93+195)
```

```
## [1] 0.3229167
```

```r
#2. P(Online = 1 | Loan = 1)
```

```r
179/(179+109)
```

```
## [1] 0.6215278
```

```r
#3. P(Loan = 1)
```

```r
288/(288+2712)
```

```
## [1] 0.096
```

```r
#4. P(CC = 1 | Loan = 0)
```

```r
788/(788+1924)
```

```
## [1] 0.2905605
```

```r
#5 P(Online = 1 | Loan = 0)
```

```r
1631/(1631+1081)
```

```
## [1] 0.6014012
```

```r
#6 P(Loan = 0)
```

```r
2712/(2712+288)
```

```
## [1] 0.904
```

```r
#Task-E.Use the quantities computed above to compute the naive Bayes probability P(Loan = 1 | CC = 1, O
```

```r
(0.3229167*0.6215278*0.096)/((0.3229167*0.6215278*0.096)+(0.29056052*0.6014012*0.904))
```

```
## [1] 0.1087106
```

```r
# The value obtained from naivebayes probability is 0.1087106

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

##The value from the pivot table is 0.1096654 and the naive bayes probability is 0.1087106 and these va


#Task-G.Which of the entries in this table are needed for computing P(Loan = 1 | CC = 1, Online = 1)? R
```

```r
nb.model = naiveBayes(Personal.Loan~CreditCard+Online, data = Train.df)
To_Predict = data.frame(CreditCard = 1 , Online = 1)
predict(nb.model,To_Predict,type = 'raw')
```

```
## Warning in predict.naiveBayes(nb.model, To_Predict, type = "raw"): Type mismatch
## between training and new data for variable 'CreditCard'. Did you use factors
## with numeric labels for training, and numeric values for new data?
```

```
## Warning in predict.naiveBayes(nb.model, To_Predict, type = "raw"): Type mismatch
## between training and new data for variable 'Online'. Did you use factors with
## numeric labels for training, and numeric values for new data?
```

```
##              0          1
## [1,] 0.9153656 0.08463445
```