assignment 3

vineeth goud maddi

3/5/2022

```
library(reshape2)
library(gmodels)
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(ISLR)
```

```
library(e1071)
ubank <- read.csv("./UniversalBank.csv")
ubank$Personal.Loan <-factor(ubank$Personal.Loan)
ubank$Online<-factor(ubank$Online)
ubank$CreditCard<-factor(ubank$CreditCard)
df = ubank
#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
set.seed(64060)
train.index <- createDataPartition(df$Personal.Loan, p= 0.6, list = FALSE)
train.df <-df[train.index,]
validation.df <- df[-train.index,]
mytable <- xtabs(~ CreditCard + Online + Personal.Loan, data = train.df)
ftable(mytable)
```

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

```
#B. 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)]
Probability = (59/(59+479))
Probability
```

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

```
#C. Creating two separate pivot tables for the training data
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$Personal.Loan, CreditCard = train.df$CreditCard)
```

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

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

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

```r
#D.[P(A | B) means "the probability ofA given B"]
#Probability_1
Probability1 = (93/(93+195))
Probability1
```

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

```r
#probability_2
Probability2 = (179/(179+109))
Probability2
```

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

```r
#Probability_3
Probability3 = (288/(288+2712))
Probability3
```

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

```r
#Probability_4
Probability4 = (788/(788+1924))
Probability4
```

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

```r
#Probability_5
Probability5 = (1631/(1631+1081))
Probability5
```

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

```r
#Probability_6
Probability6 = (2712/(2712+288))
Probability6
```

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

```r
#E naive Bayes probability P(Loan = 1 | CC = 1,online = 1)

naivebayesprobab <-(0.32*0.62*0.09)/(0.32*0.62*0.09 + 0.29*0.60*0.90)
naivebayesprobab
```

```
## [1] 0.1023525
```

```r
#F. compare the value one obtained from the pivot table in (B)

#Pivot table probability = 0.10
#Naive bayes probability = 0.10
#The key assumption we make while using naive bayes is that all variables are independent and have equa

#G. Run naive Bayes on the data. Examine the model output on training data,
#find the entry that corresponds to P(Loan = 1 | CC = 1, Online = 1)
nb.model<-naiveBayes(Personal.Loan~ Online+CreditCard, data = train.df)
To_Predict=data.frame(Online=1, CreditCard=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 'Online'. 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 'CreditCard'. Did you use factors
## with numeric labels for training, and numeric values for new data?
```

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