Naive_Bayes

sai prasad desineni

3/6/2022

```
#loading the packages
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(class)
library(ISLR)
library(reshape2)
library(e1071)


#importing the dataset(csv file)
bank = read.csv("C:/Users/desineni/Downloads/UniversalBank.csv")
View(bank)

#converting into varibles
bank$Personal.Loan = factor(bank$Personal.Loan)
bank$Online = factor(bank$Online)
bank$CreditCard = factor(bank$CreditCard)
set.seed(64060)

#splitting of data
set.seed(64060)
Train_index = createDataPartition(bank$Personal.Loan, p = 0.6, list = FALSE)
Train.df = bank[Train_index,]
validation.df = bank[Train_index,]

#task A
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 - the probability that the person will accept loan (with online= 1 and cc = 1)
59 / (59+479)
```

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

```r
#task C
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 - finding the probabilities

#1 p(cc/L)
93/(93+195)
```

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

```r
#2 p(online/L)
179/(179+109)
```

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

```r
#3 p(L)
288/(288+2712)
```

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

```r
#4 p(cc = 1/l = 0)
788/(788+1924)
```

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

```r
#5 p(online = 1/ l = 0)
1631/(1631+1081)
```

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

```r
#6 p(l=0)
2712/(2712+288)
```

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

```r
#Task E
#(FA*FB*FC)/[(FA*FB*FC)+(F4*F5*F6)]
0.3229167*0.6215278*0.096/((0.3229167*0.6215278*0.096)+(0.29056052*0.6014012*0.904))
```

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

```r
#task f
#the value thats obtained from Naive baye's probability is 0.1087106 and the value tahst obtained from


#task G - comparing naive bayes on the training data

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
```