Machine Learning Assignment 3

AKANKSHA NADUKULA

2022-10-15

```
UniversalBank = read.csv("C:/Users/gauth/Downloads/UniversalBank (2).csv")
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   :93153
##  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
```

```
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(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(class)
library(e1071)
library(reshape)
```

```
##
## Attaching package: 'reshape'
```

```
## The following object is masked from 'package:class':
##
##     condense
```

```
## The following object is masked from 'package:dplyr':
##
##     rename
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following objects are masked from 'package:reshape':
##
##     colsplit, melt, recast
```

```
df= UniversalBank

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

```
#TASK1
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                     787   76
##            1                    1144  124
## 1          0                     307   35
##            1                     477   50
```

```
#TASK2
#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)
#Probability of Loan acceptance given having a bank credit card and user
Probability = (50/(50+477))
Probability
```

```
## [1] 0.09487666
```

```
#Task3
table(Personal.Loan = train.df$Personal.Loan, Online = train.df$Online)
```

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

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

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

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

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

```
#TASK4
#i. P(CC = 1 | Loan = 1) (the proportion of credit card holders among the loan
#acceptors)
Probability1 <- 85/(85+200)
Probability1
```

```
## [1] 0.2982456
```

```r
#ii. P(Online = 1 | Loan = 1)
Probability2 <- 174/(174+111)
Probability2
```

```
## [1] 0.6105263
```

```r
#iii. P(Loan = 1) (the proportion of loan acceptors)
Probability3 <- 285/(285+2715)
Probability3
```

```
## [1] 0.095
```

```r
#iv. P(CC = 1 | Loan = 0)
Probability4 <- 784/(784+1931)
Probability4
```

```
## [1] 0.2887661
```

```r
#v. P(Online = 1 | Loan = 0)
Probability5 <- 1621/(1621+1094)
Probability5
```

```
## [1] 0.5970534
```

```r
#vi. P(Loan = 0)
Probability6 <- 2715/(2715+285)
Probability6
```

```
## [1] 0.905
```

```r
#Task 5
#Use the quantities computed above to compute the naive Ba1 probability
#P(Loan = 1 | CC = 1, Online = 1).
Task5Probability <- (Probability1*Probability2*Probability3)/((Probability1*
                                    Probability2*Probability3)+(Probability4*Probability5*Probability6))
Task5Probability
```

```
## [1] 0.09980052
```

Task6 Compare this value with the one obtained from the pivot table in (B). Which is a more accurate estimate? The value derived in 2 was 0.09487666 and in the Task 5 is 0.09980052. There is not much difference in the derived values as there is not much difference in the methods. The only 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, whereas the #naive bayes method does not. We can say that the value derived from the Task 2 is more #accurate as we have taken the exact values from the pivot table.

```r
#Task7
#Run naive Bayes on the data. Examine the model output on training data, and
#find the entry that corresponds to P(Loan = 1 | CC = 1, Online = 1).
#Compare this to the number you obtained in (E).

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')
```

```
##              0         1
## [1,] 0.8986774 0.1013226
```

The value derived from the Task 7 is 0.1013226 and the value derived from the Task 5 is 0.09980052. The values are almost alike. There is only a minute difference because of the rounding. The difference will not affect the rank order of the ouput.