

Assignment 3 - Machine learning

Hanish Bhogadi

3/6/2022

```
UB <- read.csv("C:/Users/Hanish Bhogadi/Documents/64060_hbhogadi/Assignment_3/UniversalBank.csv")
View(UB)
summary(UB)
```

```
##          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)
df = UB

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

#Task 1
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
```

```
#Task 2
#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
```

```
#Task 3
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
```

#Task 4

#i. $P(CC = 1 \mid Loan = 1)$ (the proportion of #credit card holders among the loan acceptors)

```
Probability1 <- 85/(85+200)
```

```
Probability1
```

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

#ii. $P(Online = 1 \mid Loan = 1)$

```
Probability2 <- 174/(111+174)
```

```
Probability2
```

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

#iii. $P(Loan = 1)$ (the proportion of loan acceptors)

```
Probability3 <- 285/(285+2715)
```

```
Probability3
```

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

#iv. $P(CC = 1 \mid Loan = 0)$

```
Probability4 <- 784/(1931+784)
```

```
Probability4
```

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

#v. $P(Online = 1 \mid Loan = 0)$

```
Probability5 <- 1621/(1621+1094)
```

```
Probability5
```

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

```
#vi.  $P(\text{Loan} = 0)$ 
Probability6 <- 2715/(2715+285)
Probability6
```

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

```
#Task 5
#Use the quantities computed above to compute the naive Bay probability
# $P(\text{Loan} = 1 \mid \text{CC} = 1, \text{Online} = 1)$ .
#Let  $a =$ 
Task5Probability <- (Probability1*Probability2*Probability3)/((Probability1*Probability2*Probability3)+(
Task5Probability
```

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

```
#Task 6
#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 are very similar. The only #diff
#method is the exact method would need the the exact same independent variable #classifications to pred
#say that the value derived from the Task 2 is more #accurate as we have taken
#the exact values from the pivot table.
```

```
#Task 7
#Run naive Bayes on the data. Examine the model output on training data, and
#find the entry that corresponds to  $P(\text{Loan} = 1 \mid \text{CC} = 1, \text{Online} = 1)$ .
#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 Task 7 is 0.1013226 and the value derived from the task 5 is 0.09980052. The ou
```