R Notebook

```
library(caret)
```

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

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

```
library(reshape2)
library(e1071)
```

```
bank = read.csv("UniversalBank.csv")
head(bank)
```

```
##   ID Age Experience Income ZIP.Code Family CCAvg Education Mortgage
## 1  1  25          1     49    91107      4   1.6         1        0
## 2  2  45         19     34    90089      3   1.5         1        0
## 3  3  39         15     11    94720      1   1.0         1        0
## 4  4  35          9    100    94112      1   2.7         2        0
## 5  5  35          8     45    91330      4   1.0         2        0
## 6  6  37         13     29    92121      4   0.4         2      155
##   Personal.Loan Securities.Account CD.Account Online CreditCard
## 1             0                  1          0      0          0
## 2             0                  1          0      0          0
## 3             0                  0          0      0          0
## 4             0                  0          0      0          0
## 5             0                  0          0      0          1
## 6             0                  0          0      1          0
```

```
summary(bank)
```

```
##        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   :93152
##  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
```

```
bank$Personal.Loan <- factor(bank$Personal.Loan)
bank$Online <- factor(bank$Online)
bank$CreditCard <- factor(bank$CreditCard)
```

```
set.seed(2022)
Train_Index <- createDataPartition(bank$Personal.Loan, p = 0.6, list = FALSE)
Train_Data <- bank[Train_Index,]
Validation_Data <- bank[-Train_Index,]
```

```
# A

Pivot_Table <- xtabs(~ CreditCard + Personal.Loan + Online, data = Train_Data)
ftable(Pivot_Table)
```

```
##                          Online    0    1
## CreditCard Personal.Loan
## 0          0                     789 1115
##            1                      81  123
## 1          0                     334  474
##            1                      39   45
```

```
# B

# Calculate the probability of loan acceptance for Loan = 1 CC = 1 and Online = 1

# Probability of CC and Online users
(45/519)
```

```
## [1] 0.0867052
```

```
# Probability of 3,000 Test_Data users
(45/3000)
```

```
## [1] 0.015
```

```
# C

Pivot_Loan_Online = table(Personal.Loan = Train_Data$Personal.Loan, Online = Train_Data$Online)
Pivot_Loan_Online
```

```
##              Online
## Personal.Loan    0    1
##             0 1123 1589
##             1  120  168
```

```
Pivot_Loan_CC = table(Personal.Loan = Train_Data$Personal.Loan, CreditCard = Train_Data$CreditCard)
Pivot_Loan_CC
```

```
##              CreditCard
## Personal.Loan    0    1
```

```
##              0 1904  808
##              1  204   84
```

```
# D

# 1. P(CC=1 | Loan=1)
P1 <- 84/(84+204)
P1
```

```
## [1] 0.2916667
```

```
# 2. P(Online=1 | Loan=1)
P2 <- 168/(168+120)
P2
```

```
## [1] 0.5833333
```

```
# 3. P(Loan=1)
P3 <- 288/(288+(3000-288))
P3
```

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

```
# 4. P(CC=1 | Loan=0)
P4 <- 808/(808+1904)
P4
```

```
## [1] 0.2979351
```

```
# 5. P(Online=1 | Loan=0)
P5 <- 1589/(1589+1123)
P5
```

```
## [1] 0.5859145
```

```
# 6. P(Loan=0)
P6 <- 2712/3000
P6
```

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

```
# E
(P1*P2*P3)/((P1*P2*P3) +(P4*P5*P6))
```

```
## [1] 0.09379447
```

```
# F

# Both values are similiar, but the method in E is a more accurate estimate. The result in B are highly
```

```
# G
nb_model <- naiveBayes(Personal.Loan ~ Online + CreditCard, data = Train_Data)
nb_model
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
```

```
##     0     1
## 0.904 0.096
##
## Conditional probabilities:
##    Online
## Y           0         1
##   0 0.4140855 0.5859145
##   1 0.4166667 0.5833333
##
##    CreditCard
## Y           0         1
##   0 0.7020649 0.2979351
##   1 0.7083333 0.2916667
```

```
predict(nb_model, data.frame(Online=1, CreditCard= 1), type = 'raw')
```

```
## Warning in predict.naiveBayes(nb_model, data.frame(Online = 1, CreditCard =
## 1), : 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, data.frame(Online = 1, CreditCard =
## 1), : 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,] 0.9017482 0.09825177
```

```
# The output is closest to what is found in E, both output .09
```