MAchine Learning Assingment 3

```
#Assignment 3 Machine Learning

setwd("C:/Users/nikes/Downloads/Machine Learning Assingment/Assingment 3")

library(readr)

UniversalBank <- read_csv("UniversalBank.csv")
```

```
## Rows: 5000 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (14): ID, Age, Experience, Income, ZIP Code, Family, CCAvg, Education, M...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
colnames(UniversalBank) <- c('ID', 'Age', 'Experience', 'Income', 'ZIP_Code', 'Family', 'CCAvg',
                             'Education', 'Mortgage', 'Personal_Loan',
                             'Securities_Account', 'CD_Account', 'Online', 'CreditCard')

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
```

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

```
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(reshape2)

library(ISLR)

library(e1071)
```

```
# A. Pivot table for Universal Bank

UniversalBank$Personal_Loan= as.factor(UniversalBank$Personal_Loan)

UniversalBank$Online = as.factor(UniversalBank$Online)

UniversalBank$CreditCard= as.factor(UniversalBank$CreditCard)

set.seed(123)

train.index <- sample(row.names(UniversalBank), 0.6*dim(UniversalBank)[1])

test.index <- setdiff(row.names(UniversalBank), train.index)

train.df <- UniversalBank[train.index, ]

test.df <- UniversalBank[test.index, ]
```

```
train <- UniversalBank[train.index, ]

test = UniversalBank[train.index,]

melted.UniversalBank = melt(train, id=c("CreditCard","Personal_Loan"),variable= "Online")
```

```
## Warning: attributes are not identical across measure variables; they will be
## dropped
```

```
recast.UniversalBank= dcast(melted.UniversalBank,Personal_Loan+CreditCard ~ Online)
```

```
## Aggregation function missing: defaulting to length
```

```
recast.UniversalBank[,c(1:2,14)]
```

```
##   Personal_Loan CreditCard Online
## 1             0          0   1930
## 2             0          1    792
## 3             1          0    187
## 4             1          1     91
```

```
mytable <- xtabs(~ CreditCard + Online + Personal_Loan , data = train.df)

ftable(mytable)
```

```
##                    Personal_Loan    0    1
## CreditCard Online
## 0          0                      785   65
##            1                     1145  122
## 1          0                      317   34
##            1                      475   57
```

```
# B. Probability of the customer accepting loan offer

##Probablity = 57/(57+475) = 0.10
```

```
# C. Separate pivot table

table(Personal_Loan=train$Personal_Loan, Online=train$Online)
```

```
##              Online
## Personal_Loan    0    1
##             0 1102 1620
##             1   99  179
```

```
table(Personal_Loan=train$Personal_Loan, CreditCard=train$CreditCard)
```

```
##              CreditCard
## Personal_Loan    0    1
##             0 1930  792
##             1  187   91
```

D. [P(A | B) means "the probability of A given B"]

i. Proportion of credit card holders among the loan acceptors = 91/278 = 0.32

ii. P(Online = 1 | Loan = 1) = 179/ 278 = 0.64

iii. P(Loan = 1) (the proportion of loan acceptors) = 278/ 2722 = 0.10

iv. P(CC = 1 | Loan = 0) = 792/ 2722 = 0.29

v. P(Online = 1 | Loan = 0) = 1620/2722 = 0.59

vi. P(Loan = 0) = 2722/3000 = 0.90 " # E. naive Bayes probability P(Loan = 1 | CC= 1, Online = 1).

= (0.32*0.64*0.1) / (0.32*0.64*0.1) + (0.29*0.59*0.9)

= 0.11

```
# F. Comparing value with the one obtained from the pivot table in (B).


Pivot table Probablity = (278/3000) =0.092

Naive Bayes Probablity = 0.32*0.59*0.1 / (0.32*0.64*0.1 + 0.29*0.59*0.9)

= 0.11

##As using the naive bayes the main assumption we are making is all variable are independent and have e
```

```
# G. Running naive Bayes on the data. P(Loan = 1 | CC = 1, Online = 1)

naive.train = train.df[,c(10,13:14)]

naive.test = test.df[,c(10,13:14)]

naivebayes = naiveBayes(Personal_Loan~.,data=naive.train)

naivebayes
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##          0          1
## 0.90733333 0.09266667
##
## Conditional probabilities:
##    Online
## Y           0         1
##   0 0.4048494 0.5951506
##   1 0.3561151 0.6438849
```

```
##
##    CreditCard
## Y           0         1
##   0 0.7090375 0.2909625
##   1 0.6726619 0.3273381
```

```
## We can see and analyze the result the prior probablity is exactly 0.092 as shown here which is exact
```