kNN_Assignment_ML_MD

Abhishek_Sau

20/02/2021

```
#Setting up Working Directory and importing dataset

setwd("D:\\Study\\Assignments\\MachLearn\\MachLearnAssignment2")
unibank_main<-read.csv("UniversalBank.csv")
head(unibank_main)
```

```
##   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
```

```
str(unibank_main)
```

```
## 'data.frame':    5000 obs. of  14 variables:
##  $ ID                : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Age               : int  25 45 39 35 35 37 53 50 35 34 ...
##  $ Experience        : int  1 19 15 9 8 13 27 24 10 9 ...
##  $ Income            : int  49 34 11 100 45 29 72 22 81 180 ...
##  $ ZIP.Code          : int  91107 90089 94720 94112 91330 92121 91711 93943 90089 93023 ...
##  $ Family            : int  4 3 1 1 4 4 2 1 3 1 ...
##  $ CCAvg             : num  1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...
##  $ Education         : int  1 1 1 2 2 2 2 3 2 3 ...
##  $ Mortgage          : int  0 0 0 0 0 155 0 0 104 0 ...
##  $ Personal.Loan     : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ Securities.Account: int  1 1 0 0 0 0 0 0 0 0 ...
##  $ CD.Account        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Online            : int  0 0 0 0 0 1 1 0 1 0 ...
##  $ CreditCard        : int  0 0 0 0 1 0 0 1 0 0 ...
```

```
#Calling Libraries
library(class)
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
```

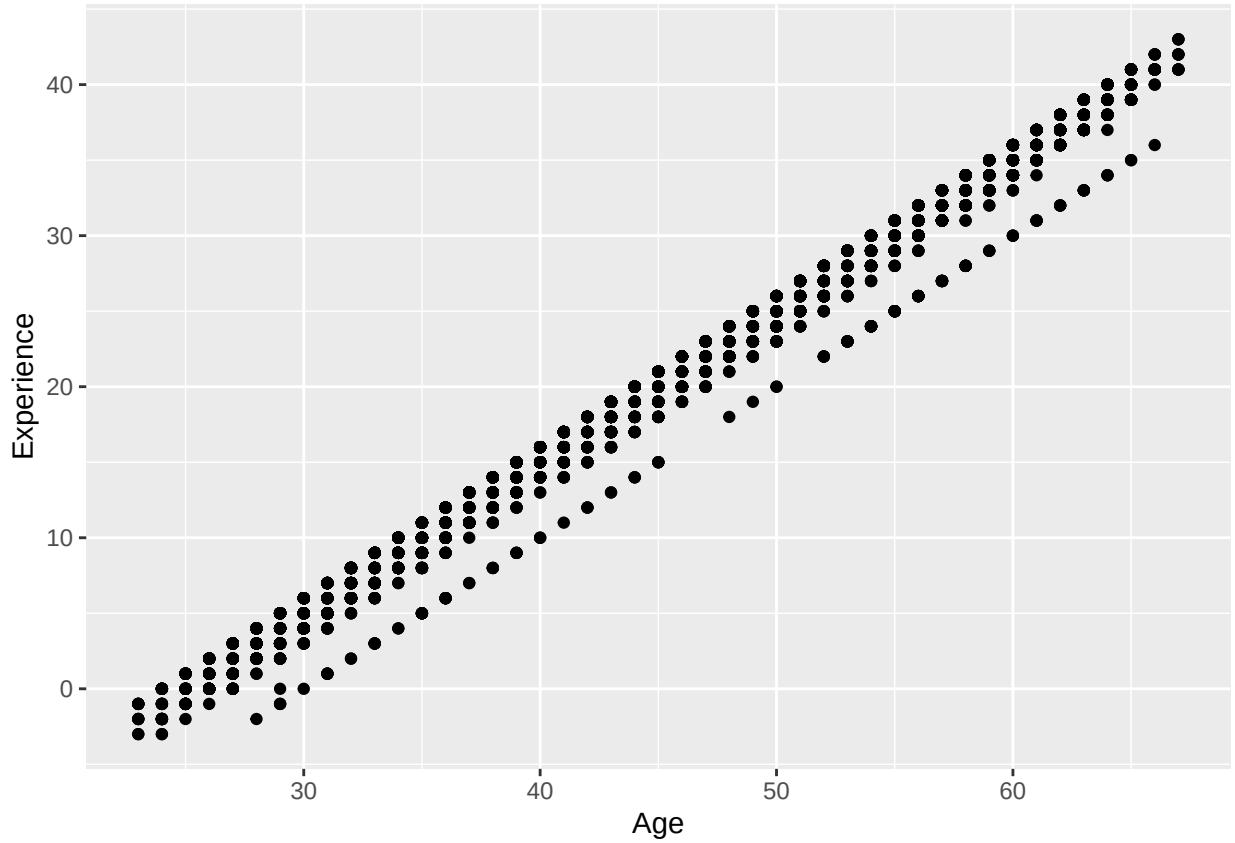
```
library(caret)
```

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

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

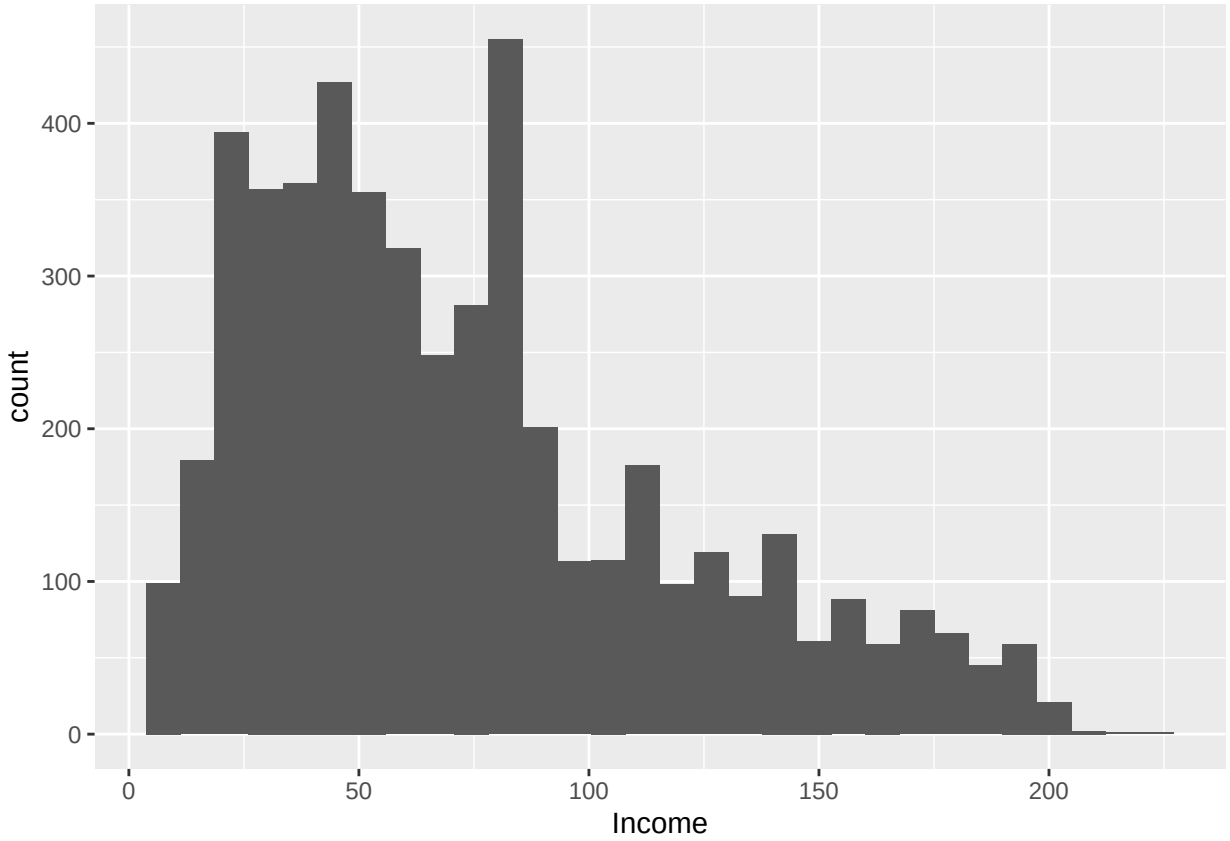
```
library(ISLR)
library(ggplot2)

#Plotting the data
ggplot(unibank_main, aes(x = Age, y= Experience)) +geom_point()
```

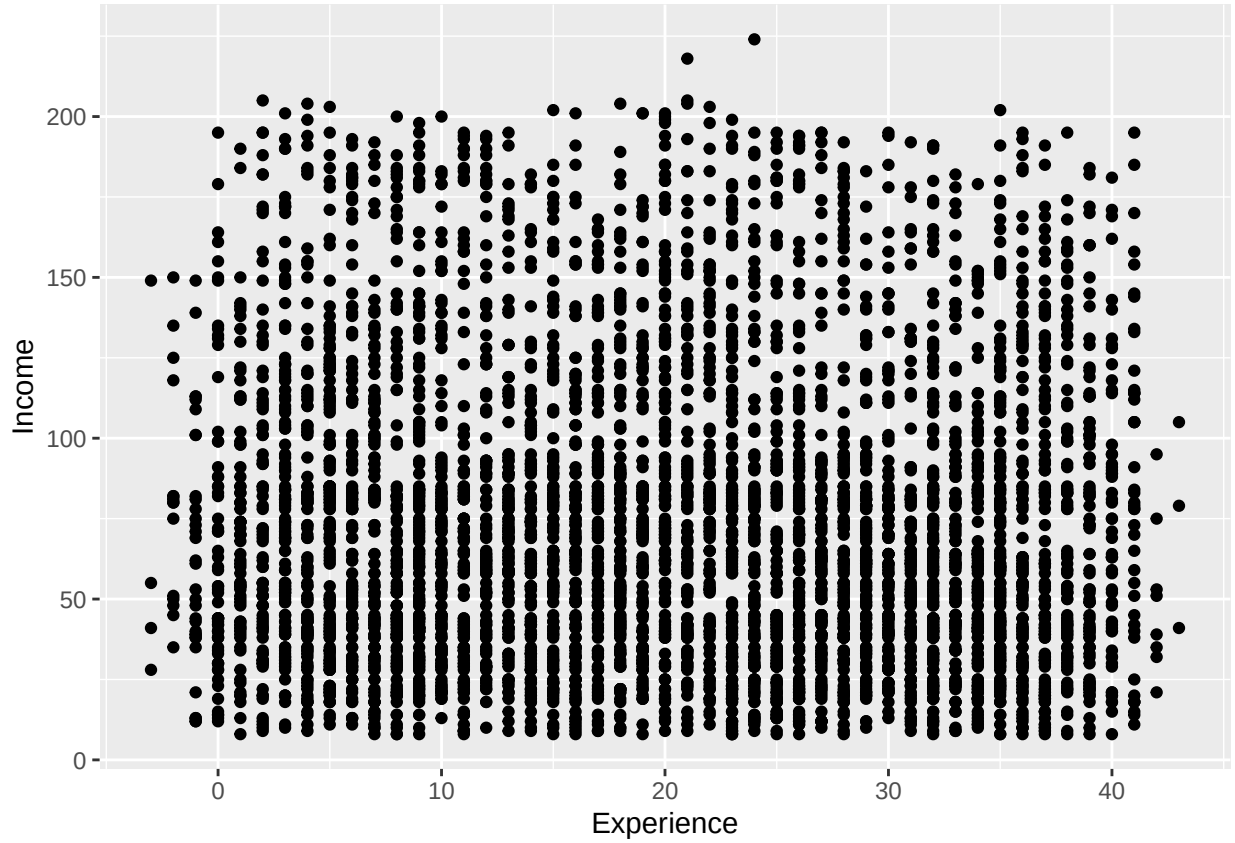



```
ggplot(unibank_main, aes(x= Income)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(unibank_main, aes(x= Experience, y=Income)) +geom_point()
```




```
#Separating categorical variables and numerical variables

uniban_cat <- c("Education", "Personal.Loan") #categorical Variables

uniban_num <- c("Age","Experience","Income","Family",
                "CCAvg","Mortgage","Securities.Account","CD.Account",
                "Online","CreditCard")        #numeric variables

cat_UB<-unibank_main[which(colnames(unibank_main) %in% uniban_cat)]
cat_UB<-data.frame(apply(cat_UB,2,as.character))

#Converting datatype for catagorical columns to factor
unibank_main$Personal.Loan <- as.factor(unibank_main$Personal.Loan)
unibank_main$Education <- as.factor(unibank_main$Education)
str(unibank_main)
```

```
## 'data.frame':    5000 obs. of  14 variables:
##  $ ID                : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Age               : int  25 45 39 35 35 37 53 50 35 34 ...
##  $ Experience        : int  1 19 15 9 8 13 27 24 10 9 ...
##  $ Income            : int  49 34 11 100 45 29 72 22 81 180 ...
##  $ ZIP.Code          : int  91107 90089 94720 94112 91330 92121 91711 93943 90089 93023 ...
##  $ Family            : int  4 3 1 1 4 4 2 1 3 1 ...
##  $ CCAvg             : num  1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...
##  $ Education         : Factor w/ 3 levels "1","2","3": 1 1 1 2 2 2 2 3 2 3 ...
##  $ Mortgage          : int  0 0 0 0 0 155 0 0 104 0 ...
```

```
##  $ Personal.Loan     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...
##  $ Securities.Account: int  1 1 0 0 0 0 0 0 0 0 ...
##  $ CD.Account        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Online            : int  0 0 0 0 0 1 1 0 1 0 ...
##  $ CreditCard        : int  0 0 0 0 1 0 0 1 0 0 ...
```

```
#Categorical variables to dummy variables conversion

library(fastDummies)
dummy_UB <- fastDummies::dummy_columns(cat_UB %>% select(-Personal.Loan)) #Dummy variable for "Educatio
dummy_UB <- dummy_UB %>% select(-Education) %>% mutate(Personal.Loan=unibank_main$Personal.Loan)
head(dummy_UB)
```

```
##   Education_1 Education_2 Education_3 Personal.Loan
## 1           1           0           0             0
## 2           1           0           0             0
## 3           1           0           0             0
## 4           0           1           0             0
## 5           0           1           0             0
## 6           0           1           0             0
```

```
str(dummy_UB)
```

```
## 'data.frame':    5000 obs. of  4 variables:
##  $ Education_1  : int  1 1 1 0 0 0 0 0 0 0 ...
##  $ Education_2  : int  0 0 0 1 1 1 1 0 1 0 ...
##  $ Education_3  : int  0 0 0 0 0 0 0 1 0 1 ...
##  $ Personal.Loan: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...
```

```
#Numerical Variables
numeric_UB <- unibank_main[(which(colnames(unibank_main) %in% uniban_num))]
head(numeric_UB)
```

```
##   Age Experience Income Family CCAvg Mortgage Securities.Account CD.Account
## 1  25          1     49      4   1.6        0                  1          0
## 2  45         19     34      3   1.5        0                  1          0
## 3  39         15     11      1   1.0        0                  0          0
## 4  35          9    100      1   2.7        0                  0          0
## 5  35          8     45      4   1.0        0                  0          0
## 6  37         13     29      4   0.4      155                  0          0
##   Online CreditCard
## 1      0          0
## 2      0          0
## 3      0          0
## 4      0          0
## 5      0          1
## 6      1          0
```

```
str(numeric_UB)
```

```
## 'data.frame':    5000 obs. of  10 variables:
##  $ Age              : int  25 45 39 35 35 37 53 50 35 34 ...
##  $ Experience       : int  1 19 15 9 8 13 27 24 10 9 ...
##  $ Income           : int  49 34 11 100 45 29 72 22 81 180 ...
##  $ Family           : int  4 3 1 1 4 4 2 1 3 1 ...
##  $ CCAvg            : num  1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...
##  $ Mortgage         : int  0 0 0 0 0 155 0 0 104 0 ...
```

```
##  $ Securities.Account: int  1 1 0 0 0 0 0 0 0 0 ...
##  $ CD.Account        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Online            : int  0 0 0 0 0 1 1 0 1 0 ...
##  $ CreditCard        : int  0 0 0 0 1 0 0 1 0 0 ...
```

```
new_UB_subset <- cbind(numeric_UB,dummy_UB)
head(new_UB_subset)
```

```
##   Age Experience Income Family CCAvg Mortgage Securities.Account CD.Account
## 1  25          1     49      4   1.6        0                  1          0
## 2  45         19     34      3   1.5        0                  1          0
## 3  39         15     11      1   1.0        0                  0          0
## 4  35          9    100      1   2.7        0                  0          0
## 5  35          8     45      4   1.0        0                  0          0
## 6  37         13     29      4   0.4      155                  0          0
##   Online CreditCard Education_1 Education_2 Education_3 Personal.Loan
## 1      0          0           1           0           0             0
## 2      0          0           1           0           0             0
## 3      0          0           1           0           0             0
## 4      0          0           0           1           0             0
## 5      0          1           0           1           0             0
## 6      1          0           0           1           0             0
```

```
str(new_UB_subset)
```

```
## 'data.frame':    5000 obs. of  14 variables:
##  $ Age               : int  25 45 39 35 35 37 53 50 35 34 ...
##  $ Experience        : int  1 19 15 9 8 13 27 24 10 9 ...
##  $ Income            : int  49 34 11 100 45 29 72 22 81 180 ...
##  $ Family            : int  4 3 1 1 4 4 2 1 3 1 ...
##  $ CCAvg             : num  1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...
##  $ Mortgage          : int  0 0 0 0 0 155 0 0 104 0 ...
##  $ Securities.Account: int  1 1 0 0 0 0 0 0 0 0 ...
##  $ CD.Account        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Online            : int  0 0 0 0 0 1 1 0 1 0 ...
##  $ CreditCard        : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Education_1       : int  1 1 1 0 0 0 0 0 0 0 ...
##  $ Education_2       : int  0 0 0 1 1 1 1 0 1 0 ...
##  $ Education_3       : int  0 0 0 0 0 0 0 1 0 1 ...
##  $ Personal.Loan     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...
```

```
dim(new_UB_subset)
```

```
## [1] 5000   14
```

```
#Splitting this combined data set with numeric and dummy variables into Training and test set

set.seed(12)

split_index <- createDataPartition(new_UB_subset$Personal.Loan, p=0.6, times = 1, list = FALSE)

train_UB <- new_UB_subset[split_index, ] #60% data into training set
test_UB <- new_UB_subset[-split_index, ] #40% data into test set

#Normalization

train_normal <- train_UB
```

```
test_normal <- test_UB

summary(train_UB)
```

```
##       Age           Experience        Income           Family
##  Min.   :23.00   Min.   :-3.00   Min.   :  8.00   Min.   :1.000
##  1st Qu.:35.00   1st Qu.:10.00   1st Qu.: 38.00   1st Qu.:1.000
##  Median :45.00   Median :20.00   Median : 64.00   Median :2.000
##  Mean   :45.39   Mean   :20.17   Mean   : 73.65   Mean   :2.386
##  3rd Qu.:55.00   3rd Qu.:30.00   3rd Qu.: 98.00   3rd Qu.:3.000
##  Max.   :67.00   Max.   :43.00   Max.   :224.00   Max.   :4.000
##      CCAvg            Mortgage      Securities.Account   CD.Account
##  Min.   : 0.000   Min.   :  0.00   Min.   :0.0000     Min.   :0.00000
##  1st Qu.: 0.700   1st Qu.:  0.00   1st Qu.:0.0000     1st Qu.:0.00000
##  Median : 1.500   Median :  0.00   Median :0.0000     Median :0.00000
##  Mean   : 1.954   Mean   : 56.82   Mean   :0.1037     Mean   :0.06033
##  3rd Qu.: 2.600   3rd Qu.:101.00   3rd Qu.:0.0000     3rd Qu.:0.00000
##  Max.   :10.000   Max.   :635.00   Max.   :1.0000     Max.   :1.00000
##      Online        CreditCard      Education_1      Education_2
##  Min.   :0.000   Min.   :0.0000   Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000
##  Median :1.000   Median :0.0000   Median :0.0000   Median :0.000
##  Mean   :0.591   Mean   :0.2993   Mean   :0.4267   Mean   :0.274
##  3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.000
##  Max.   :1.000   Max.   :1.0000   Max.   :1.0000   Max.   :1.000
##   Education_3     Personal.Loan
##  Min.   :0.0000   0:2712
##  1st Qu.:0.0000   1: 288
##  Median :0.0000
##  Mean   :0.2993
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

```
str(train_UB)
```

```
## 'data.frame':    3000 obs. of  14 variables:
##  $ Age               : int  25 39 35 53 50 65 29 67 60 38 ...
##  $ Experience        : int  1 15 9 27 24 39 5 41 30 14 ...
##  $ Income            : int  49 11 100 72 22 105 45 112 22 130 ...
##  $ Family            : int  4 1 1 2 1 4 3 1 1 4 ...
##  $ CCAvg             : num  1.6 1 2.7 1.5 0.3 2.4 0.1 2 1.5 4.7 ...
##  $ Mortgage          : int  0 0 0 0 0 0 0 0 0 134 ...
##  $ Securities.Account: int  1 0 0 0 0 0 0 1 0 0 ...
##  $ CD.Account        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Online            : int  0 0 0 1 0 0 1 0 1 0 ...
##  $ CreditCard        : int  0 0 0 0 1 0 0 0 1 0 ...
##  $ Education_1       : int  1 1 0 0 0 0 0 1 0 0 ...
##  $ Education_2       : int  0 0 1 1 0 0 1 0 0 0 ...
##  $ Education_3       : int  0 0 0 0 1 1 0 0 1 1 ...
##  $ Personal.Loan     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...
```

```
dim(train_UB)
```

```
## [1] 3000   14
```

```r
normalization_model <- preProcess(train_UB[,-14],method = c("center", "scale"))

train_normal <- predict(normalization_model,train_UB[,-14])
test_normal <- predict(normalization_model,test_UB [,-14])

head(train_normal)
```

```
##           Age Experience      Income     Family      CCAvg   Mortgage
## 1  -1.7729794 -1.6675909 -0.53167517  1.4129805 -0.1985582 -0.5626351
## 3  -0.5557714 -0.4499480 -1.35132939 -1.2133773 -0.5354243 -0.5626351
## 4  -0.9035451 -0.9717950  0.56838706 -1.2133773  0.4190296 -0.5626351
## 7   0.6614366  0.5937459 -0.03556868 -0.3379247 -0.2547026 -0.5626351
## 8   0.4006063  0.3328224 -1.11406106 -1.2133773 -0.9284348 -0.5626351
## 11  1.7047577  1.6374398  0.67623629  1.4129805  0.2505966 -0.5626351
##    Securities.Account CD.Account     Online CreditCard Education_1 Education_2
## 1           2.9399694 -0.2533491 -1.2018761 -0.6535058   1.1590091   -0.614235
## 3          -0.3400262 -0.2533491 -1.2018761 -0.6535058   1.1590091   -0.614235
## 4          -0.3400262 -0.2533491 -1.2018761 -0.6535058  -0.8625184    1.627499
## 7          -0.3400262 -0.2533491  0.8317552 -0.6535058  -0.8625184    1.627499
## 8          -0.3400262 -0.2533491 -1.2018761  1.5296983  -0.8625184   -0.614235
## 11         -0.3400262 -0.2533491 -1.2018761 -0.6535058  -0.8625184   -0.614235
##    Education_3
## 1   -0.6535058
## 3   -0.6535058
## 4   -0.6535058
## 7   -0.6535058
## 8    1.5296983
## 11   1.5296983
```

```r
head(test_normal)
```

```
##           Age Experience     Income     Family      CCAvg   Mortgage
## 2  -0.0341108 -0.1020501 -0.8552229  0.5375279 -0.2547026 -0.5626351
## 5  -0.9035451 -1.0587695 -0.6179546  1.4129805 -0.5354243 -0.5626351
## 6  -0.7296582 -0.6238970 -0.9630721  1.4129805 -0.8722904  0.9720673
## 9  -0.9035451 -0.8848205  0.1585600  0.5375279 -0.7600017  0.4671007
## 10 -0.9904885 -0.9717950  2.2939749 -1.2133773  3.8999792 -0.5626351
## 13  0.2267195  0.2458479  0.8703649 -0.3379247  1.0366174 -0.5626351
##    Securities.Account CD.Account     Online CreditCard Education_1 Education_2
## 2           2.9399694 -0.2533491 -1.2018761 -0.6535058   1.1590091   -0.614235
## 5          -0.3400262 -0.2533491 -1.2018761  1.5296983  -0.8625184    1.627499
## 6          -0.3400262 -0.2533491  0.8317552 -0.6535058  -0.8625184    1.627499
## 9          -0.3400262 -0.2533491  0.8317552 -0.6535058  -0.8625184    1.627499
## 10         -0.3400262 -0.2533491 -1.2018761 -0.6535058  -0.8625184   -0.614235
## 13          2.9399694 -0.2533491 -1.2018761 -0.6535058  -0.8625184   -0.614235
##    Education_3
## 2   -0.6535058
## 5   -0.6535058
## 6   -0.6535058
## 9   -0.6535058
## 10   1.5296983
## 13   1.5296983
```

```r
dim(train_normal)
```

```
## [1] 3000   13
```

```r
dim(test_normal)
```

```
## [1] 2000   13
```

```r
#-----------------------------------------------------------------------------------------------------------
# Question - 1 -> Modelling k-NN with K=1 and sample data

library(class)
library(gmodels)
ques_1_sample <- data.frame(Age = 40, Experience = 10, Income = 84, Family = 2, CCAvg = 2,
                            Mortgage = 0, Securities.Account = 0, CD.Account = 0, Online = 1, CreditCar
                            Education_1 = 0, Education_2 = 1,Education_3 = 0)

knn_test1 <- knn(train_normal,ques_1_sample,cl= train_UB$Personal.Loan, k=1, prob = TRUE)
knn_test1
```

```
## [1] 1
## attr(,"prob")
## [1] 1
## Levels: 0 1
```

```r
#-----------------------------------------------------------------------------------------------------------
# Question 2- Finding the best value of K to avoid overfitting

chooseK_data <- data.frame(k = seq(1,20,1), accuracy = rep(0,20))

for(i in 1:20){
  k_choose <- knn(train_normal, test_normal, cl = train_UB$Personal.Loan, k=i)
  chooseK_data[i,2] <- confusionMatrix(k_choose, test_UB$Personal.Loan)$overall[1]
}
chooseK_data
```

```
##     k accuracy
## 1   1   0.9620
## 2   2   0.9535
## 3   3   0.9655
## 4   4   0.9600
## 5   5   0.9600
## 6   6   0.9610
## 7   7   0.9555
## 8   8   0.9560
## 9   9   0.9550
## 10 10   0.9510
## 11 11   0.9515
## 12 12   0.9510
## 13 13   0.9510
## 14 14   0.9490
## 15 15   0.9500
## 16 16   0.9495
## 17 17   0.9490
## 18 18   0.9475
## 19 19   0.9480
```

```
## 20 20   0.9485
```

```
max(chooseK_data[c("accuracy")])
```

```
## [1] 0.9655
```

```
#K with highest accuracy , k=3 (best K)

knn_best <- knn(train_normal,test_normal,cl= train_UB$Personal.Loan, k=3, prob = TRUE)

# Question 3 - Confusion Matrix using the best K

test_CM <- test_UB$Personal.Loan
pred_CM <- knn_best
CrossTable(x=test_CM, y=pred_CM,prop.chisq = TRUE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  2000
##
##
##              | pred_CM
##      test_CM |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |      1802 |         6 |      1808 |
##              |     7.987 |   110.335 |           |
##              |     0.997 |     0.003 |     0.904 |
##              |     0.966 |     0.044 |           |
##              |     0.901 |     0.003 |           |
## -------------|-----------|-----------|-----------|
##            1 |        63 |       129 |       192 |
##              |    75.208 |  1038.988 |           |
##              |     0.328 |     0.672 |     0.096 |
##              |     0.034 |     0.956 |           |
##              |     0.032 |     0.064 |           |
## -------------|-----------|-----------|-----------|
## Column Total |      1865 |       135 |      2000 |
##              |     0.932 |     0.068 |           |
## -------------|-----------|-----------|-----------|
##
##
```

```
#Question 4 - Classify the customer with best K

knn_2 <- knn(train_normal,ques_1_sample,cl= train_UB$Personal.Loan, k=3, prob = TRUE)
```

```r
#Question 5 - New split - 50-30-20

# New Split
set.seed(1204)
split_index2 <- createDataPartition(new_UB_subset$Personal.Loan, p=0.5, times = 1, list = FALSE)

new_train_UB <- new_UB_subset [split_index2, ]
testVal_UB <- new_UB_subset [-split_index2, ]
str(testVal_UB)
```

```
## 'data.frame':    2500 obs. of  14 variables:
##  $ Age               : int  45 39 35 37 29 59 60 38 46 29 ...
##  $ Experience        : int  19 15 9 13 5 32 30 14 21 5 ...
##  $ Income            : int  34 11 100 29 45 40 22 130 193 62 ...
##  $ Family            : int  3 1 1 4 3 4 1 4 2 1 ...
##  $ CCAvg             : num  1.5 1 2.7 0.4 0.1 2.5 1.5 4.7 8.1 1.2 ...
##  $ Mortgage          : int  0 0 0 155 0 0 0 134 0 260 ...
##  $ Securities.Account: int  1 0 0 0 0 0 0 0 0 0 ...
##  $ CD.Account        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Online            : int  0 0 0 1 1 1 1 0 0 1 ...
##  $ CreditCard        : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Education_1       : int  1 1 0 0 0 0 0 0 0 1 ...
##  $ Education_2       : int  0 0 1 1 1 1 0 0 0 0 ...
##  $ Education_3       : int  0 0 0 0 0 0 1 1 1 0 ...
##  $ Personal.Loan     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 2 1 ...
```

```r
split_index_TV <- createDataPartition(testVal_UB$Personal.Loan, p=0.6, times = 1, list= FALSE)
new_test_UB <- testVal_UB [-split_index_TV, ]
new_val_UB <- testVal_UB [split_index_TV, ]

train_normal2 <- new_train_UB
test_normal2 <- new_test_UB
val_normal2 <- new_val_UB

#Normalization of new split data
normalization_model2 <- preProcess(new_train_UB[,-14],method = c("center", "scale"))

train_normal2 <- predict(normalization_model2,new_train_UB[,-14])
test_normal2 <- predict(normalization_model2,new_test_UB[,-14])
val_normal2 <- predict(normalization_model2,new_val_UB[,-14])

#Using best k value -  k=3

knn_3 <- knn(train_normal2,val_normal2,cl= new_train_UB$Personal.Loan, k=3, prob = TRUE)

#confusion matrix

test_CM2 <- new_val_UB$Personal.Loan
pred_CM2 <- knn_3
CrossTable(x=test_CM2, y=pred_CM2, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
```

```
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  1500
##
##
##              | pred_CM2
##     test_CM2 |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |      1347 |         9 |      1356 |
##              |     0.993 |     0.007 |     0.904 |
##              |     0.960 |     0.093 |           |
##              |     0.898 |     0.006 |           |
## -------------|-----------|-----------|-----------|
##            1 |        56 |        88 |       144 |
##              |     0.389 |     0.611 |     0.096 |
##              |     0.040 |     0.907 |           |
##              |     0.037 |     0.059 |           |
## -------------|-----------|-----------|-----------|
## Column Total |      1403 |        97 |      1500 |
##              |     0.935 |     0.065 |           |
## -------------|-----------|-----------|-----------|
##
##
```