Assignment_2

2/18/2022

assignment:: k-NN for classification

This assignment describes the steps for K-NN classification in R.

We used **Universal bank** customers data includes demographic information, the customer's relationship with the bank (mortgage, securities account, etc.), and the customer response to the last personal loan campaign (Personal Loan):

Load the dataset and packages into R.

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.1.3
```

```
uniBank <- read.csv("C:\\Users\\amuna\\Downloads\\UniversalBank.csv")


# categorical predictors are represented using 0 and 1, dummy variables for ordinal or categorical vari

uniBank$education_1 <- ifelse(uniBank$Education == 1, 1, 0)
uniBank$education_2 <- ifelse(uniBank$Education == 2, 1, 0)
uniBank$education_3 <- ifelse(uniBank$Education == 3, 1, 0)

# loan accept dummy variable for loan acceptance, 1 == accept, 0 == not accept.
uniBank$acceptance <- ifelse(uniBank$Personal.Loan == 1, 1, 0)

library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
##
## 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
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.3
```

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

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

```r
library(FNN)
```

```
## Warning: package 'FNN' was built under R version 4.1.3
```

```r
library(gmodels) #sorting the data as a contingency table
```

```
## Warning: package 'gmodels' was built under R version 4.1.3
```

```r
#select the
uniBank_1 <- uniBank %>% select(Age,Experience,Income,Family,CCAvg,education_1,education_2, education_3

# create a K-NN Partition from the bank data into training (60%) and validation (40%) sets and use the

set.seed(1234) #Random number generator
train_index_1 = createDataPartition(uniBank_1$acceptance, p = .6, list = FALSE)
train_data_1 = uniBank_1[train_index_1,] # train
validation_data_1 = uniBank_1[-train_index_1,] # test

summary(train_data_1$acceptance)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.00000 0.00000 0.09467 0.00000 1.00000
```

```r
summary(validation_data_1$acceptance)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   0.098   0.000   1.000
```

```r
# preProcess for normalization :: change the values of numeric columns in the dataset to a common scale

normalize_values <- preProcess(train_data_1[, 1:6], method=c("center", "scale"))
train_data_1[, 1:6] <- predict(normalize_values, train_data_1[, 1:6]) # Replace first two columns with

validation_data_1[, 1:6] <- predict(normalize_values, validation_data_1[, 1:6])
uniBank_1[, 1:6] <- predict(normalize_values, uniBank_1[, 1:6])

summary(train_data_1)
```

```
##       Age              Experience           Income             Family
##  Min.   :-1.93856   Min.   :-2.00902   Min.   :-1.4229   Min.   :-1.2028
##  1st Qu.:-0.89828   1st Qu.:-0.88070   1st Qu.:-0.7723   1st Qu.:-1.2028
```

```
##  Median :-0.03138   Median :-0.01276   Median :-0.2302   Median :-0.3385
##  Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.92221   3rd Qu.: 0.85518   3rd Qu.: 0.5505   3rd Qu.: 0.5258
##  Max.   : 1.87579   Max.   : 1.98350   Max.   : 3.2612   Max.   : 1.3901
##      CCAvg          education_1       education_2     education_3
##  Min.   :-1.0986   Min.   :-0.8479   Min.   :0.000   Min.   :0.0000
##  1st Qu.:-0.6993   1st Qu.:-0.8479   1st Qu.:0.000   1st Qu.:0.0000
##  Median :-0.2429   Median :-0.8479   Median :0.000   Median :0.0000
##  Mean   : 0.0000   Mean   : 0.0000   Mean   :0.286   Mean   :0.2957
##  3rd Qu.: 0.3277   3rd Qu.: 1.1790   3rd Qu.:1.000   3rd Qu.:1.0000
##  Max.   : 4.6066   Max.   : 1.1790   Max.   :1.000   Max.   :1.0000
##     Mortgage      Securities.Account   CD.Account        Online
##  Min.   :  0.00   Min.   :0.0000     Min.   :0.000   Min.   :0.0000
##  1st Qu.:  0.00   1st Qu.:0.0000     1st Qu.:0.000   1st Qu.:0.0000
##  Median :  0.00   Median :0.0000     Median :0.000   Median :1.0000
##  Mean   : 56.97   Mean   :0.1087     Mean   :0.062   Mean   :0.5867
##  3rd Qu.:102.00   3rd Qu.:0.0000     3rd Qu.:0.000   3rd Qu.:1.0000
##  Max.   :617.00   Max.   :1.0000     Max.   :1.000   Max.   :1.0000
##    CreditCard       acceptance
##  Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.00000
##  Median :0.0000   Median :0.00000
##  Mean   :0.2883   Mean   :0.09467
##  3rd Qu.:1.0000   3rd Qu.:0.00000
##  Max.   :1.0000   Max.   :1.00000
```

```
var(train_data_1[, 1:6])
```

```
##                      Age   Experience      Income      Family       CCAvg
## Age          1.00000000  0.994450659 -0.07890015 -0.04865101 -0.06484148
## Experience   0.99445066  1.000000000 -0.06994198 -0.05384665 -0.06253098
## Income      -0.07890015 -0.069941975  1.00000000 -0.17128308  0.64420088
## Family      -0.04865101 -0.053846649 -0.17128308  1.00000000 -0.11983245
## CCAvg       -0.06484148 -0.062530978  0.64420088 -0.11983245  1.00000000
## education_1 -0.01823952  0.006658799  0.21916181 -0.12064719  0.18080587
##              education_1
## Age         -0.018239522
## Experience   0.006658799
## Income       0.219161812
## Family      -0.120647192
## CCAvg        0.180805866
## education_1  1.000000000
```

```
summary(validation_data_1)
```

```
##       Age              Experience          Income              Family
##  Min.   :-1.938555   Min.   :-2.00902   Min.   :-1.422924   Min.   :-1.20280
##  1st Qu.:-0.833261   1st Qu.:-0.88070   1st Qu.:-0.750660   1st Qu.:-1.20280
##  Median :-0.031382   Median :-0.01276   Median :-0.208510   Median :-0.33851
##  Mean   :-0.005115   Mean   :-0.00920   Mean   : 0.008631   Mean   : 0.01023
##  3rd Qu.: 0.835515   3rd Qu.: 0.76839   3rd Qu.: 0.528812   3rd Qu.: 0.52577
##  Max.   : 1.875792   Max.   : 1.98350   Max.   : 3.131127   Max.   : 1.39006
##      CCAvg           education_1         education_2      education_3
```

```
##  Min.   :-1.09863   Min.   :-0.847914   Min.   :0.0000   Min.   :0.000
##  1st Qu.:-0.69927   1st Qu.:-0.847914   1st Qu.:0.0000   1st Qu.:0.000
##  Median :-0.18580   Median :-0.847914   Median :0.0000   Median :0.000
##  Mean   : 0.01749   Mean   : 0.004392   Mean   :0.2725   Mean   :0.307
##  3rd Qu.: 0.38472   3rd Qu.: 1.178972   3rd Qu.:1.0000   3rd Qu.:1.000
##  Max.   : 4.60656   Max.   : 1.178972   Max.   :1.0000   Max.   :1.000
##     Mortgage     Securities.Account   CD.Account        Online
##  Min.   :  0.0   Min.   :0.000      Min.   :0.000   Min.   :0.000
##  1st Qu.:  0.0   1st Qu.:0.000      1st Qu.:0.000   1st Qu.:0.000
##  Median :  0.0   Median :0.000      Median :0.000   Median :1.000
##  Mean   : 55.8   Mean   :0.098      Mean   :0.058   Mean   :0.612
##  3rd Qu.: 98.0   3rd Qu.:0.000      3rd Qu.:0.000   3rd Qu.:1.000
##  Max.   :635.0   Max.   :1.000      Max.   :1.000   Max.   :1.000
##    CreditCard       acceptance
##  Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:0.000
##  Median :0.0000   Median :0.000
##  Mean   :0.3025   Mean   :0.098
##  3rd Qu.:1.0000   3rd Qu.:0.000
##  Max.   :1.0000   Max.   :1.000
```

```
var(validation_data_1[, 1:6])
```

```
##                     Age  Experience      Income      Family       CCAvg
## Age          0.96926598  0.96727353 -0.01871568 -0.04139858 -0.03154457
## Experience   0.96727353  0.97724627 -0.01075193 -0.04893734 -0.03037296
## Income      -0.01871568 -0.01075193  0.99188836 -0.13309845  0.64140092
## Family      -0.04139858 -0.04893734 -0.13309845  0.96012601 -0.09059215
## CCAvg       -0.03154457 -0.03037296  0.64140092 -0.09059215  0.98569828
## education_1 -0.04164858 -0.01720327  0.21557650 -0.11334903  0.12020322
##             education_1
## Age         -0.04164858
## Experience  -0.01720327
## Income       0.21557650
## Family      -0.11334903
## CCAvg        0.12020322
## education_1  1.00160205
```

```
model.task1 <- train(as.factor(acceptance) ~ Age + Experience + Income + Family + CCAvg + education_1 +
                + education_3 + Mortgage + Securities.Account + CD.Account + Online + CreditCard,
                data = train_data_1, method = "knn")
model.task1
```

```
## k-Nearest Neighbors
##
## 3000 samples
##   13 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 3000, 3000, 3000, 3000, 3000, 3000, ...
## Resampling results across tuning parameters:
```

```
##
##   k  Accuracy   Kappa
##   5  0.9393603  0.5748286
##   7  0.9410654  0.5688549
##   9  0.9402266  0.5545432
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 7.
```

Test the model on to a new dataframe

```
predict_customer_df = data.frame(Age = as.integer(40), Experience = as.integer(10), Income = as.integer
                         Family = as.integer(2), CCAvg = as.integer(2), Mortgage = as.integer(0),
                         `Securities.Account` = as.integer(0), `CD.Account` = as.integer(0),
                         Online = as.integer(1), CreditCard = as.integer(1), education_1 = as.integer(0
                         education_2 = as.integer(1), education_3 = as.integer(0))

predict_customer_df = predict(normalize_values, predict_customer_df)

model.task1.1 <- knn(train = train_data_1[, 1:13], test = predict_customer_df[, 1:13],
                cl = train_data_1$acceptance,
                k = 7)
```

balance of overfitting and underfitting

```
cust.acc.df <- data.frame(k = seq(1, 14, 1), accuracy = rep(0, 14))
for(i in 1:14) {
  best_knn_1 <- knn(train = train_data_1[, 1:13, drop = FALSE], test = validation_data_1[, 1:13, drop =
               cl = as.factor(train_data_1$acceptance), k = i)
  cust.acc.df[i, 2] <- confusionMatrix(best_knn_1, as.factor(validation_data_1$acceptance))$overall[1]
}

cust.acc.df
```

```
##     k accuracy
## 1   1   0.9455
## 2   2   0.9375
## 3   3   0.9395
## 4   4   0.9365
## 5   5   0.9390
## 6   6   0.9365
## 7   7   0.9385
## 8   8   0.9365
## 9   9   0.9355
## 10 10   0.9345
## 11 11   0.9365
## 12 12   0.9340
## 13 13   0.9350
## 14 14   0.9370
```

choose k = 1, to test set because that has the highest accuracy

```
# confusion matrix for when k = 3
conf_knn_3 <- knn(train = train_data_1[, 1:13, drop = FALSE], test = validation_data_1[, 1:13, drop = FA
            cl = as.factor(train_data_1$acceptance), k = 1, prob=TRUE)
confusionMatrix(conf_knn_3, as.factor(validation_data_1$acceptance))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1770   75
##          1   34  121
##
##                Accuracy : 0.9455
##                  95% CI : (0.9346, 0.955)
##     No Information Rate : 0.902
##     P-Value [Acc > NIR] : 1.031e-12
##
##                   Kappa : 0.66
##
##  Mcnemar's Test P-Value : 0.0001275
##
##             Sensitivity : 0.9812
##             Specificity : 0.6173
##          Pos Pred Value : 0.9593
##          Neg Pred Value : 0.7806
##              Prevalence : 0.9020
##          Detection Rate : 0.8850
##    Detection Prevalence : 0.9225
##       Balanced Accuracy : 0.7992
##
##        'Positive' Class : 0
##
```

Test the model on a new data set

```
test_customer_df = data.frame(Age = as.integer(40), Experience = as.integer(10), Income = as.integer(84)
                        Family = as.integer(2), CCAvg = as.integer(2), Mortgage = as.integer(0),
                        `Securities.Account` = as.integer(0), `CD.Account` = as.integer(0),
                        Online = as.integer(1), CreditCard = as.integer(1), education_1 = as.integer(0)
                        education_2 = as.integer(1), education_3 = as.integer(0))

test_customer_df = predict(normalize_values, test_customer_df)

model.task1.2 <- knn(train = train_data_1[, 1:13], test = test_customer_df[, 1:13],
                cl = train_data_1$acceptance,
                k =  1, prob = TRUE)

model.task1.2
```

```
## [1] 0
## attr(,"prob")
## [1] 1
## attr(,"nn.index")
##      [,1]
## [1,] 2812
## attr(,"nn.dist")
##          [,1]
## [1,] 2.013674
## Levels: 0
```

customer is still in level 0: the loan will not be accepted

Repartition the data, this time into training, validation, and test sets (50% : 30% : 20%)

```
# 50% for training, 30% for validation and 20% for testing, split the data
uniBank_2 <- uniBank %>% select(Age, Experience, Income, Family, CCAvg, Mortgage, Securities.Account, CD

set.seed(1234)

Test_Index_2 = createDataPartition(uniBank_2$acceptance, p=.2, list = FALSE)
Test_Data_2 = uniBank_2[Test_Index_2,]
Test_Validation_Data_2 = uniBank_2[-Test_Index_2,] # Validation and training data

train_Index_2 = createDataPartition(uniBank_2$acceptance, p=.5, list = FALSE)
train_Data_2 = uniBank_2[train_Index_2,]
train_Validation_Data_2 = Test_Validation_Data_2[-train_Index_2,] #validation data

summary(train_Data_2)
```

```
##       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.: 39.00   1st Qu.:1.000
##  Median :45.00   Median :20.00   Median : 64.00   Median :2.000
##  Mean   :44.99   Mean   :19.75   Mean   : 74.73   Mean   :2.392
##  3rd Qu.:55.00   3rd Qu.:29.25   3rd Qu.:101.25   3rd Qu.:3.000
##  Max.   :67.00   Max.   :42.00   Max.   :205.00   Max.   :4.000
##      CCAvg          Mortgage      Securities.Account   CD.Account
##  Min.   : 0.00   Min.   :  0.00   Min.   :0.000      Min.   :0.000
##  1st Qu.: 0.70   1st Qu.:  0.00   1st Qu.:0.000      1st Qu.:0.000
##  Median : 1.50   Median :  0.00   Median :0.000      Median :0.000
##  Mean   : 1.96   Mean   : 60.98   Mean   :0.106      Mean   :0.066
##  3rd Qu.: 2.60   3rd Qu.:109.00   3rd Qu.:0.000      3rd Qu.:0.000
##  Max.   :10.00   Max.   :617.00   Max.   :1.000      Max.   :1.000
##      Online         CreditCard    education_1      education_2
##  Min.   :0.0000   Min.   :0.00   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.00   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :1.0000   Median :0.00   Median :0.0000   Median :0.0000
##  Mean   :0.5936   Mean   :0.29   Mean   :0.4248   Mean   :0.2804
```

```
##  3rd Qu.:1.0000   3rd Qu.:1.00   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.00   Max.   :1.0000   Max.   :1.0000
##   education_3       acceptance
##  Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:0.000
##  Median :0.0000   Median :0.000
##  Mean   :0.2948   Mean   :0.102
##  3rd Qu.:1.0000   3rd Qu.:0.000
##  Max.   :1.0000   Max.   :1.000
```

```
summary(Test_Data_2)
```

```
##       Age          Experience        Income           Family
##  Min.   :23.00   Min.   :-3.00   Min.   :  8.00   Min.   :1.000
##  1st Qu.:36.00   1st Qu.:10.00   1st Qu.: 37.25   1st Qu.:1.000
##  Median :45.50   Median :20.00   Median : 62.00   Median :2.000
##  Mean   :45.56   Mean   :20.35   Mean   : 72.85   Mean   :2.364
##  3rd Qu.:55.00   3rd Qu.:30.00   3rd Qu.: 99.00   3rd Qu.:3.000
##  Max.   :67.00   Max.   :42.00   Max.   :203.00   Max.   :4.000
##      CCAvg           Mortgage      Securities.Account   CD.Account
##  Min.   : 0.000   Min.   :  0.00   Min.   :0.000      Min.   :0.000
##  1st Qu.: 0.700   1st Qu.:  0.00   1st Qu.:0.000      1st Qu.:0.000
##  Median : 1.500   Median :  0.00   Median :0.000      Median :0.000
##  Mean   : 1.967   Mean   : 59.12   Mean   :0.105      Mean   :0.055
##  3rd Qu.: 2.500   3rd Qu.:106.00   3rd Qu.:0.000      3rd Qu.:0.000
##  Max.   :10.000   Max.   :617.00   Max.   :1.000      Max.   :1.000
##      Online        CreditCard     education_1     education_2     education_3
##  Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.00
##  1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.00
##  Median :1.000   Median :0.000   Median :0.000   Median :0.000   Median :0.00
##  Mean   :0.591   Mean   :0.294   Mean   :0.431   Mean   :0.289   Mean   :0.28
##  3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.00
##  Max.   :1.000   Max.   :1.000   Max.   :1.000   Max.   :1.000   Max.   :1.00
##    acceptance
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.086
##  3rd Qu.:0.000
##  Max.   :1.000
```

```
summary(train_Validation_Data_2)
```

```
##       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.: 39.00   1st Qu.:1.000
##  Median :45.00   Median :20.00   Median : 63.50   Median :2.000
##  Mean   :45.12   Mean   :19.87   Mean   : 73.03   Mean   :2.403
##  3rd Qu.:55.00   3rd Qu.:30.00   3rd Qu.: 95.00   3rd Qu.:3.000
##  Max.   :67.00   Max.   :43.00   Max.   :218.00   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   : 1.884   Mean   : 56.67   Mean   :0.1156     Mean   :0.06156
##  3rd Qu.: 2.500   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.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :1.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.5981   Mean   :0.2903   Mean   :0.4169   Mean   :0.2853
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##   education_3       acceptance
##  Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.00000
##  Median :0.0000   Median :0.00000
##  Mean   :0.2978   Mean   :0.09409
##  3rd Qu.:1.0000   3rd Qu.:0.00000
##  Max.   :1.0000   Max.   :1.00000
```

```
# Use preProcess() from the caret package to normalize Sales and Age.

Norm_Values = preProcess(train_Data_2[,1:6], method = c("center", "scale"))
train_Norm_DF2 <- predict(Norm_Values, train_Data_2)
Validation_Norm_DF2 <- predict(Norm_Values, train_Validation_Data_2)
train_Validation_Norm_DF2 <- predict(Norm_Values, Test_Validation_Data_2)
Test_Norm_DF2 <- predict(Norm_Values, Test_Data_2)

summary(train_Norm_DF2)
```

```
##       Age               Experience          Income             Family
##  Min.   :-1.9252099   Min.   :-1.98902   Min.   :-1.4307   Min.   :-1.2092
##  1st Qu.:-0.8746752   1st Qu.:-0.85254   1st Qu.:-0.7660   1st Qu.:-1.2092
##  Median : 0.0007704   Median : 0.02168   Median :-0.2300   Median :-0.3405
##  Mean   : 0.0000000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.8762160   3rd Qu.: 0.83033   3rd Qu.: 0.5687   3rd Qu.: 0.5281
##  Max.   : 1.9267507   Max.   : 1.94496   Max.   : 2.7932   Max.   : 1.3968
##      CCAvg            Mortgage       Securities.Account   CD.Account
##  Min.   :-1.0993   Min.   :-0.5828   Min.   :0.000      Min.   :0.000
##  1st Qu.:-0.7068   1st Qu.:-0.5828   1st Qu.:0.000      1st Qu.:0.000
##  Median :-0.2582   Median :-0.5828   Median :0.000      Median :0.000
##  Mean   : 0.0000   Mean   : 0.0000   Mean   :0.106      Mean   :0.066
##  3rd Qu.: 0.3586   3rd Qu.: 0.4589   3rd Qu.:0.000      3rd Qu.:0.000
##  Max.   : 4.5081   Max.   : 5.3138   Max.   :1.000      Max.   :1.000
##      Online         CreditCard    education_1      education_2
##  Min.   :0.0000   Min.   :0.00   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.00   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :1.0000   Median :0.00   Median :0.0000   Median :0.0000
##  Mean   :0.5936   Mean   :0.29   Mean   :0.4248   Mean   :0.2804
##  3rd Qu.:1.0000   3rd Qu.:1.00   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.00   Max.   :1.0000   Max.   :1.0000
##   education_3       acceptance
##  Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:0.000
##  Median :0.0000   Median :0.000
```

```
##  Mean   :0.2948   Mean   :0.102
##  3rd Qu.:1.0000   3rd Qu.:0.000
##  Max.   :1.0000   Max.   :1.000
```

```
var(train_Norm_DF2[,1:6])
```

```
##                     Age   Experience      Income      Family       CCAvg
## Age         1.000000000  0.994202117 -0.05860490 -0.04778758 -0.05009312
## Experience  0.994202117  1.000000000 -0.04875426 -0.05384630 -0.04695374
## Income     -0.058604897 -0.048754264  1.00000000 -0.16794469  0.64881237
## Family     -0.047787577 -0.053846295 -0.16794469  1.00000000 -0.12010147
## CCAvg      -0.050093121 -0.046953741  0.64881237 -0.12010147  1.00000000
## Mortgage    0.006437659  0.007674841  0.21456012 -0.03188852  0.12154842
##                Mortgage
## Age         0.006437659
## Experience  0.007674841
## Income      0.214560119
## Family     -0.031888525
## CCAvg       0.121548424
## Mortgage    1.000000000
```

```
summary(Validation_Norm_DF2)
```

```
##       Age               Experience           Income             Family
##  Min.   :-1.9252099   Min.   :-1.98902   Min.   :-1.43069   Min.   :-1.209171
##  1st Qu.:-0.8746752   1st Qu.:-0.85254   1st Qu.:-0.76602   1st Qu.:-1.209171
##  Median : 0.0007704   Median : 0.02168   Median :-0.24071   Median :-0.340514
##  Mean   : 0.0114177   Mean   : 0.01057   Mean   :-0.03634   Mean   : 0.009471
##  3rd Qu.: 0.8762160   3rd Qu.: 0.89590   3rd Qu.: 0.43469   3rd Qu.: 0.528144
##  Max.   : 1.9267507   Max.   : 2.03238   Max.   : 3.07195   Max.   : 1.396801
##      CCAvg              Mortgage        Securities.Account   CD.Account
##  Min.   :-1.09935   Min.   :-0.58279   Min.   :0.0000     Min.   :0.00000
##  1st Qu.:-0.70682   1st Qu.:-0.58279   1st Qu.:0.0000     1st Qu.:0.00000
##  Median :-0.25822   Median :-0.58279   Median :0.0000     Median :0.00000
##  Mean   :-0.04307   Mean   :-0.04118   Mean   :0.1156     Mean   :0.06156
##  3rd Qu.: 0.30253   3rd Qu.: 0.38246   3rd Qu.:0.0000     3rd Qu.:0.00000
##  Max.   : 4.50814   Max.   : 5.48585   Max.   :1.0000     Max.   :1.00000
##      Online          CreditCard      education_1      education_2
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :1.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.5981   Mean   :0.2903   Mean   :0.4169   Mean   :0.2853
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##   education_3       acceptance
##  Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.00000
##  Median :0.0000   Median :0.00000
##  Mean   :0.2978   Mean   :0.09409
##  3rd Qu.:1.0000   3rd Qu.:0.00000
##  Max.   :1.0000   Max.   :1.00000
```

```
var(Validation_Norm_DF2[,1:6])
```

```
##                    Age  Experience      Income      Family       CCAvg
## Age         0.99587920  0.99108503 -0.02181377 -0.03282156 -0.03784044
## Experience  0.99108503  0.99863826 -0.01183130 -0.03721544 -0.03481156
## Income     -0.02181377 -0.01183130  0.92122995 -0.13030408  0.55268772
## Family     -0.03282156 -0.03721544 -0.13030408  0.99097195 -0.09755957
## CCAvg      -0.03784044 -0.03481156  0.55268772 -0.09755957  0.86915733
## Mortgage   -0.01723937 -0.01347411  0.20933078 -0.02394103  0.13159617
##               Mortgage
## Age        -0.01723937
## Experience -0.01347411
## Income      0.20933078
## Family     -0.02394103
## CCAvg       0.13159617
## Mortgage    0.94018419
```

```
## The training and validation should be combine and predict for test data
Norm_Values <- preProcess(train_Validation_Data_2, method=c("center", "scale"))

train_Validation_Norm_DF2 <- predict(Norm_Values, train_Validation_Data_2)
Test_Norm_DF2 <- predict(Norm_Values, Test_Data_2)

summary(train_Validation_Norm_DF2[,1:6])
```

```
##       Age              Experience          Income            Family
##  Min.   :-1.94063   Min.   :-2.00095   Min.   :-1.4527   Min.   :-1.2242
##  1st Qu.:-0.88792   1st Qu.:-0.86369   1st Qu.:-0.7602   1st Qu.:-1.2242
##  Median :-0.01067   Median : 0.01112   Median :-0.2129   Median :-0.3516
##  Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.86659   3rd Qu.: 0.88594   3rd Qu.: 0.4908   3rd Qu.: 0.5210
##  Max.   : 1.91929   Max.   : 2.02319   Max.   : 3.2385   Max.   : 1.3936
##      CCAvg            Mortgage
##  Min.   :-1.1330   Min.   :-0.5586
##  1st Qu.:-0.7120   1st Qu.:-0.5586
##  Median :-0.2308   Median :-0.5586
##  Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.3707   3rd Qu.: 0.4369
##  Max.   : 4.8818   Max.   : 5.7001
```

```
summary(Test_Norm_DF2[,1:6])
```

```
##       Age              Experience          Income            Family
##  Min.   :-1.94063   Min.   :-2.00095   Min.   :-1.4527   Min.   :-1.22418
##  1st Qu.:-0.80020   1st Qu.:-0.86369   1st Qu.:-0.7993   1st Qu.:-1.22418
##  Median : 0.03319   Median : 0.01112   Median :-0.2464   Median :-0.35158
##  Mean   : 0.03811   Mean   : 0.04209   Mean   :-0.0041   Mean   :-0.03395
##  3rd Qu.: 0.86659   3rd Qu.: 0.88594   3rd Qu.: 0.5801   3rd Qu.: 0.52103
##  Max.   : 1.91929   Max.   : 1.93571   Max.   : 2.9034   Max.   : 1.39364
##      CCAvg            Mortgage
##  Min.   :-1.1330   Min.   :-0.55858
##  1st Qu.:-0.7120   1st Qu.:-0.55858
```

```
##  Median :-0.2308   Median :-0.55858
##  Mean   : 0.0501   Mean   : 0.02409
##  3rd Qu.: 0.3707   3rd Qu.: 0.48619
##  Max.   : 4.8818   Max.   : 5.52273
```

```
# confusion Matrix and get predict
KNN_train2 <- knn(train=train_Norm_DF2, test = train_Norm_DF2,
                  cl = train_Norm_DF2$acceptance, k=3 ) # we found k=3 best

KNN_Validation2 <- knn(train=train_Norm_DF2[,1:13], test = Validation_Norm_DF2[,1:13],
                       cl = train_Norm_DF2$acceptance, k=3)

KNN_Test2 <- knn(train=train_Norm_DF2, test = Test_Norm_DF2,
                 cl = train_Norm_DF2$acceptance, k=3)

confusionMatrix(KNN_train2, as.factor(train_Norm_DF2$acceptance))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2244   20
##          1    1  235
##
##                Accuracy : 0.9916
##                  95% CI : (0.9872, 0.9948)
##     No Information Rate : 0.898
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9526
##
##  Mcnemar's Test P-Value : 8.568e-05
##
##             Sensitivity : 0.9996
##             Specificity : 0.9216
##          Pos Pred Value : 0.9912
##          Neg Pred Value : 0.9958
##              Prevalence : 0.8980
##          Detection Rate : 0.8976
##    Detection Prevalence : 0.9056
##       Balanced Accuracy : 0.9606
##
##        'Positive' Class : 0
##
```

```
confusionMatrix(KNN_Validation2, as.factor(Validation_Norm_DF2$acceptance))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1804   59
##          1    6  129
```

```
##
##                Accuracy : 0.9675
##                  95% CI : (0.9587, 0.9748)
##     No Information Rate : 0.9059
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7816
##
##  Mcnemar's Test P-Value : 1.12e-10
##
##             Sensitivity : 0.9967
##             Specificity : 0.6862
##          Pos Pred Value : 0.9683
##          Neg Pred Value : 0.9556
##              Prevalence : 0.9059
##          Detection Rate : 0.9029
##    Detection Prevalence : 0.9324
##       Balanced Accuracy : 0.8414
##
##        'Positive' Class : 0
##
```

```
confusionMatrix(KNN_Test2,as.factor(Test_Data_2$acceptance))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 912   0
##          1   2  86
##
##                Accuracy : 0.998
##                  95% CI : (0.9928, 0.9998)
##     No Information Rate : 0.914
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9874
##
##  Mcnemar's Test P-Value : 0.4795
##
##             Sensitivity : 0.9978
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9773
##              Prevalence : 0.9140
##          Detection Rate : 0.9120
##    Detection Prevalence : 0.9120
##       Balanced Accuracy : 0.9989
##
##        'Positive' Class : 0
##
```

After building the model we got 99.8% accurate with a sensitivity of 99.78%. The acceptance of the loan 99.78%.

The model on which we trained data got highest accuracy, sensitivity and specificity compared to the validation and testing data.