Assignment_2

```
UniversalBank <- read.csv("C:/Users/Dell/Desktop/UniversalBank.csv")
View(UniversalBank)

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
```

NULL Variables

```
UniversalBank$ID <- NULL
UniversalBank$ZIP.Code <- NULL
UniversalBank$Personal.Loan = as.factor(UniversalBank$Personal.Loan)
summary(UniversalBank)
```

```
##       Age          Experience       Income           Family
##  Min.   :23.00   Min.   :-3.0   Min.   :  8.00   Min.   :1.000
##  1st Qu.:35.00   1st Qu.:10.0   1st Qu.: 39.00   1st Qu.:1.000
##  Median :45.00   Median :20.0   Median : 64.00   Median :2.000
##  Mean   :45.34   Mean   :20.1   Mean   : 73.77   Mean   :2.396
##  3rd Qu.:55.00   3rd Qu.:30.0   3rd Qu.: 98.00   3rd Qu.:3.000
##  Max.   :67.00   Max.   :43.0   Max.   :224.00   Max.   :4.000
##      CCAvg          Education        Mortgage     Personal.Loan
##  Min.   : 0.000   Min.   :1.000   Min.   :  0.0   0:4520
##  1st Qu.: 0.700   1st Qu.:1.000   1st Qu.:  0.0   1: 480
##  Median : 1.500   Median :2.000   Median :  0.0
##  Mean   : 1.938   Mean   :1.881   Mean   : 56.5
##  3rd Qu.: 2.500   3rd Qu.:3.000   3rd Qu.:101.0
##  Max.   :10.000   Max.   :3.000   Max.   :635.0
##  Securities.Account   CD.Account         Online         CreditCard
##  Min.   :0.0000     Min.   :0.0000   Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.0000     1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000
##  Median :0.0000     Median :0.0000   Median :1.0000   Median :0.000
##  Mean   :0.1044     Mean   :0.0604   Mean   :0.5968   Mean   :0.294
##  3rd Qu.:0.0000     3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.000
##  Max.   :1.0000     Max.   :1.0000   Max.   :1.0000   Max.   :1.000
```

Normalizing the data

```
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(class)


Norm_model <- preProcess(UniversalBank[,-8], method = c("center", "scale"))
UniversalBank_norm=predict(Norm_model,UniversalBank[,-8])
summary(UniversalBank_norm)
```

```
##       Age             Experience            Income             Family
##  Min.   :-1.94871   Min.   :-2.014710   Min.   :-1.4288   Min.   :-1.2167
##  1st Qu.:-0.90188   1st Qu.:-0.881116   1st Qu.:-0.7554   1st Qu.:-1.2167
##  Median :-0.02952   Median :-0.009121   Median :-0.2123   Median :-0.3454
##  Mean   : 0.00000   Mean   : 0.000000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.84284   3rd Qu.: 0.862874   3rd Qu.: 0.5263   3rd Qu.: 0.5259
##  Max.   : 1.88967   Max.   : 1.996468   Max.   : 3.2634   Max.   : 1.3973
##      CCAvg           Education          Mortgage       Securities.Account
##  Min.   :-1.1089   Min.   :-1.0490   Min.   :-0.5555   Min.   :-0.3414
##  1st Qu.:-0.7083   1st Qu.:-1.0490   1st Qu.:-0.5555   1st Qu.:-0.3414
##  Median :-0.2506   Median : 0.1417   Median :-0.5555   Median :-0.3414  
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000  
##  3rd Qu.: 0.3216   3rd Qu.: 1.3324   3rd Qu.: 0.4375   3rd Qu.:-0.3414  
##  Max.   : 4.6131   Max.   : 1.3324   Max.   : 5.6875   Max.   : 2.9286  
##    CD.Account          Online          CreditCard     
##  Min.   :-0.2535   Min.   :-1.2165   Min.   :-0.6452  
##  1st Qu.:-0.2535   1st Qu.:-1.2165   1st Qu.:-0.6452  
##  Median :-0.2535   Median : 0.8219   Median :-0.6452  
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000  
##  3rd Qu.:-0.2535   3rd Qu.: 0.8219   3rd Qu.: 1.5495  
##  Max.   : 3.9438   Max.   : 0.8219   Max.   : 1.5495
```

adding back the target variable

```
UniversalBank_norm$Personal.Loan=UniversalBank$Personal.Loan
```

dividing the data into train and validation

```
Train_Index = createDataPartition(UniversalBank$Personal.Loan,p=0.6, list=FALSE)
Train.df=UniversalBank_norm[Train_Index,]
Validation.df=UniversalBank_norm[-Train_Index,]
```

Task 1

Use the train set and knn method with k=1 to predict if a new customer will accept a loan offer

```
To_Predict=data.frame(Age=40, Experience=10,Income=84,Family=2,
                      CCAvg=2,Education=1, Mortgage=0,
                      Securities.Account=0,CD.Account=0,Online=1,CreditCard=1 )
print(To_Predict)
```

```
##   Age Experience Income Family CCAvg Education Mortgage Securities.Account
## 1  40         10     84      2     2         1        0                  0
##   CD.Account Online CreditCard
## 1          0      1          1
```

#applying the normalization

```
To_Predict_norm <- predict(Norm_model,To_Predict)
print(To_Predict_norm)
```

```
##          Age Experience    Income     Family     CCAvg Education   Mortgage
## 1 -0.4657003 -0.8811162 0.2221371 -0.3453975 0.0355115 -1.048973 -0.5554684
##   Securities.Account CD.Account    Online CreditCard
## 1         -0.3413892 -0.2535149 0.8218687   1.549477
```

#Knn Prediction (first 7 columns are input variables and Personal loan is the target variable)

```
Prediction <-knn(train=Train.df[,1:7],
             test=To_Predict_norm[,1:7],
             cl=Train.df$Personal.Loan,
             k=1)
print(Prediction)
```

```
## [1] 0
## Levels: 0 1
```

- As output is zero new customer will not accept a loan offer

Task 2

crossvalidation for overfitting

```
set.seed(123)
fitControl <- trainControl(method = "repeatedcv",number = 3,repeats = 2)

searchGrid = expand.grid(k = 1:10)

Knn.model=train(Personal.Loan~.,
                data=Train.df,
                method='knn',
                tuneGrid=searchGrid,
                trControl = fitControl)
```

#Task 3

confusion matrix for the validation data that results from using the best k

```
To_Predict=data.frame(Age=40, Experience=10,Income=84,Family=2,
                      CCAvg=2,Education=1, Mortgage=0,
                      Securities.Account=0,CD.Account=0,Online=1,CreditCard=1 )

predictions<-predict(Knn.model,Validation.df)
confusionMatrix(predictions,Validation.df$Personal.Loan)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1797   71
##          1   11  121
##
##                Accuracy : 0.959
##                  95% CI : (0.9494, 0.9673)
##     No Information Rate : 0.904
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7254
##
##  Mcnemar’s Test P-Value : 7.247e-11
##
##             Sensitivity : 0.9939
##             Specificity : 0.6302
##          Pos Pred Value : 0.9620
##          Neg Pred Value : 0.9167
##              Prevalence : 0.9040
##          Detection Rate : 0.8985
##    Detection Prevalence : 0.9340
##       Balanced Accuracy : 0.8121
##
##        ’Positive’ Class : 0
##
```

considering some other attributes

```
To_Predict=data.frame(Age=40, Experience=10,Income=84,Family=2,
                      CCAvg=2,Education=1, Mortgage=0,
                      Securities.Account=0,CD.Account=0,Online=1,CreditCard=1 )

To_Predict_norm=predict(Norm_model,To_Predict)
predict(Knn.model,To_Predict_norm)
```

```
## [1] 0
## Levels: 0 1
```