```
library(dplyr)
library(ISLR)
library(tibble)
library(caret)
library(class)
library(FNN)
library(fastDummies)
Bank <- UniversalBank
Bank$Education=factor(Bank$Education, levels = c(1, 2, 3),labels =
c("Education_1", "Education_2", "Education_3"))
Bank$`Personal Loan`=factor(Bank$`Personal Loan`, levels = c(0,1))
colnames(UniversalBank)
colnames(Bank)
Bank <-Bank [ , -c(1, 5)] # removing ID and Zip code
#converting categorical predictors to dummy variables
Bank <- dummy_columns(Bank, select_columns = "Education")
dummy_Education <- as.data.frame(dummy.code(Bank$Education)
Bank_without_education <-  Bank[, -c(6)]
dummy_Education <-Bank [ , -c(1, 5)]
UBank_data <-Bank [ , -c(1, 5)]
colnames(UBank_data)
colnames(Bank)
colnames(Bank_without_education)
UBank_data <- Bank_without_education
colnames(UBank_data)
colnames(UBank_data)
#Partitioning the data into Traning(60%) and Validation(40%)

set.seed(2019)
Train_Index   =createDataPartition(UBank_data$Age, p= 0.6, list =FALSE)
Train_Data    =UBank_data[Train_Index,]
Validation_Data  =UBank_data[-Train_Index,]
Generating Test Data
Test_Data <- data.frame(Age=40 , Experience=10, Income = 84, Family = 2, CCAvg
= 2, Education_Education_1 = 0, Education_Education_2 = 1,
Education_Education_3 = 0, Mortgage = 0, Securities.Account = 0, CD.Account =
0, Online = 1,  CreditCard = 1, stringsAsFactors = FALSE)
####Data Normalization
train.norm.df    <- Train_Data
valid.norm.df   <-Validation_Data
test.norm.df    <- Test_Data
maindata.norm.df <-UBank_data
head(maindata.norm.df)
#Use preProcess()from the caret package to normalize
norm.values <- preProcess(Train_Data[,-7], method=c("center", "scale"))
train.norm.df[,-7] <- predict(norm.values, Train_Data[,-7])
valid.norm.df [,-7]<- predict(norm.values, Validation_Data[,-7])

test.norm.df <- predict(norm.values, Test_Data)

maindata.norm.df[,-7] <- predict(norm.values,UBank_data[,-7])
head(maindata.norm.df)

####Perfoming k-NN classification , using k = 1
```

```
library(FNN)
set.seed(2019)
prediction <- knn(train = train.norm.df[,-7], test = valid.norm.df[,-7],
                  cl = train.norm.df[,7], k = 1, prob=TRUE)
prediction <- knn(train = train.norm.df[,-7], test = valid.norm.df[,-7], cl =
train.norm.df[,7], k = 1, prob = TRUE)
actual= valid.norm.df$`Personal Loan`
prediction_prob = attr(prediction,"prob")

table(prediction,actual)
mean(prediction==actual)
cutoff = 0.5
dim(train.norm.df[,-7]
    [1] 3001   13
    dim(valid.norm.df[,-7])
    [1] 1999   13

    length(train.norm.df[,7])
    > length(cl)

library(class)
NROW(train.norm.df)
sqrt(3001)

####Generating loop to find best k
set.seed(2019)

fitControl <- trainControl(method = "repeatedcv", number = 3, repeats = 2)
searchGrid=expand.grid(k = 1:10)
knn.model=train(`Personal Loan`~.,data = Train_Data,
method='knn',tuneGrid=searchGrid, trControl = fitControl,)
knn.model

confusionMatrix(predictions,valid.norm.df$`Personal Loan`)
# Validation data results using best k value [i.e: k = 3]

library(FNN)
set.seed(2019)
prediction <- knn(train = train.norm.df[,-7], test = valid.norm.df[,-7],
                  cl = train.norm.df[,7], k = 3, prob=TRUE)
prediction <- knn(train = train.norm.df[,-7], test = valid.norm.df[,-7], cl =
train.norm.df[,7], k = 1, prob = TRUE)


actual= valid.norm.df$`Personal Loan`
prediction_prob = attr(prediction,"prob")

# confusion matrix for the best k value =3
table(prediction,actual)
#accuracy of the best k=3
mean(prediction==actual)

library(FNN)
```

```
#Classifying the customer using the best k  [performing k-NN classification on
test data]

Test_Data <- data.frame(Age=40 , Experience=10, Income = 84, Family = 2, CCAvg
= 2, Education_Education_1 = 0, Education_Education_2 = 1,
Education_Education_3 = 0, Mortgage = 0, Securities.Account = 0, CD.Account =
0, Online = 1,  CreditCard = 1, stringsAsFactors = FALSE)
maindata.norm.df <- as.data.frame(maindata.norm.df)
prediction_test <- knn(train = maindata.norm.df[,-7], test = Test_Data,  cl =
maindata.norm.df[,7], k = 1, prob=TRUE)
head(prediction_test)
```