#Downlaoding and Loading Librarys

#install.packages("caret")  
#install.packages("psych")  
#install.packages("FNN")  
#install.packages("ggplot2")  
#install.packages("lattice")  
#install.packages("ISLR")  
   
library("caret")

## Loading required package: ggplot2

## Loading required package: lattice

library("psych")

##   
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

library("psych")  
library("FNN")  
library("ggplot2")  
library("lattice")  
library("ISLR")  
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("gmodels")

#Loading Data from csv

UniversalBank <- read.csv("UniversalBank.csv")  
Bank = UniversalBank[,c(-1,-5)]

#Dummy Variables

Educationbreakout = Bank$Education  
Testfactor = factor(Educationbreakout, labels=c("highschool","some college","college grad"))  
Bank[["Education"]] = Testfactor  
levels(Bank$Education)

## [1] "highschool" "some college" "college grad"

Dummy<- dummyVars("~Education", data=Bank)  
y = predict(Dummy,newdata=Bank)  
Bank = cbind(Bank, y)  
Bank = Bank[, c(-6)]

* Partitioning the data into training as 60% and validation as 40% sets; Normalization of the data and k-NN classification using k=1 :-

set.seed(10)  
Part= createDataPartition(Bank$Personal.Loan, p=0.6, list=FALSE)  
Train\_Data = Bank[Part,]  
dim(Train\_Data)

## [1] 3000 14

Validation\_data = Bank[-Part,]  
dim(Validation\_data)

## [1] 2000 14

#Normalization

Norm = preProcess(Train\_Data, method = c('center', 'scale'))  
Train = predict(Norm,Train\_Data)  
Validation\_Norm = predict(Norm,Validation\_data)  
Train\_Predict = Train[,-7]  
validation\_Predict = Validation\_Norm[,-7]  
Train\_labels = factor(Train\_Data[,7], levels=c(0,1), labels = c("N","Y"))  
Val\_labels = factor(Validation\_data[,7], levels=c(0,1), labels = c("N","Y"))

#Summary Data

#summary(Train\_Norm)  
#summary(Validation\_data)

Predict = knn(Train\_Predict,validation\_Predict,  
 cl=Train\_labels,k=1,prob = TRUE)  
str(Predict)

## Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 1 2 1 1 ...  
## - attr(\*, "prob")= num [1:2000] 1 1 1 1 1 1 1 1 1 1 ...  
## - attr(\*, "nn.index")= int [1:2000, 1] 2183 418 1003 462 1258 2488 389 473 1977 2341 ...  
## - attr(\*, "nn.dist")= num [1:2000, 1] 0.926 0.987 0.546 0.372 0.876 ...

#Setting parameters for loan acceptance criteria

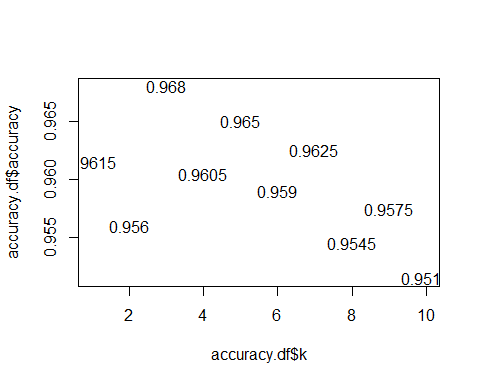
Ideal\_loan\_parameters = c(40, 10, 84, 2, 2, 0, 0, 0, 1, 1, 0, 1, 0)  
Predict2 = knn(Train\_Predict, Ideal\_loan\_parameters, cl=Train\_labels, k=1, prob = TRUE)  
summary(Predict2)

## Y   
## 1

#Question 1: Customer would be approved with the above credentials

#Hypertuning w/ Data Val

accuracy.df = data.frame(k = seq(1, 10, 1),accuracy = rep(0,10))  
 Val\_labels = as.factor(Val\_labels)  
 for(i in 1:10)   
 {  
 PredictedTest3 = knn(Train\_Predict, validation\_Predict,   
 cl=Train\_labels, k=i,prob = TRUE)  
   
 accuracy.df[i, 2] <- confusionMatrix(PredictedTest3, Val\_labels)$overall[1]   
 }   
plot(x = accuracy.df$k,y = accuracy.df$accuracy, type = "n")  
 text(x = accuracy.df$k,y = accuracy.df$accuracy, labels = accuracy.df$accuracy)



head(accuracy.df)

## k accuracy  
## 1 1 0.9615  
## 2 2 0.9560  
## 3 3 0.9680  
## 4 4 0.9605  
## 5 5 0.9650  
## 6 6 0.9590

# Question 2: k = 3 is the highest accuracy from the graph shown at 96.8%

#Using K =3 @ 98.6% for the following model   
PredictedTest\_label4 <- knn(Train\_Predict, validation\_Predict, cl=Train\_labels, k=3)  
CrossTable(x=Val\_labels,y=PredictedTest\_label4 ,prop.chisq = FALSE)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Row Total |  
## | N / Col Total |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 2000   
##   
##   
## | PredictedTest\_label4   
## Val\_labels | N | Y | Row Total |   
## -------------|-----------|-----------|-----------|  
## N | 1804 | 11 | 1815 |   
## | 0.994 | 0.006 | 0.907 |   
## | 0.971 | 0.077 | |   
## | 0.902 | 0.005 | |   
## -------------|-----------|-----------|-----------|  
## Y | 53 | 132 | 185 |   
## | 0.286 | 0.714 | 0.092 |   
## | 0.029 | 0.923 | |   
## | 0.026 | 0.066 | |   
## -------------|-----------|-----------|-----------|  
## Column Total | 1857 | 143 | 2000 |   
## | 0.928 | 0.071 | |   
## -------------|-----------|-----------|-----------|  
##   
##

Ideal\_loan\_parameters = c(40, 10, 84, 2, 2, 0, 0, 0, 1, 1, 0, 1, 0)  
Predict3 = knn(Train\_Predict, Ideal\_loan\_parameters, cl=Train\_labels, k=3, prob = TRUE)  
summary(Predict3)

## Y   
## 1

```{r}

PredictedTest\_label4 <- knn(Train\_Predict, validation\_Predict, cl=Train\_labels, k=1)

CrossTable(x=Val\_labels,y=PredictedTest\_label4 ,prop.chisq = FALSE)

```

Cell Contents

|-------------------------|

| N |

| N / Row Total |

| N / Col Total |

| N / Table Total |

|-------------------------|

Total Observations in Table: 2000

| PredictedTest\_label4

Val\_labels | N | Y | Row Total |

-------------|-----------|-----------|-----------|

N | 1789 | 26 | 1815 |

| 0.986 | 0.014 | 0.907 |

| 0.972 | 0.163 | |

| 0.894 | 0.013 | |

-------------|-----------|-----------|-----------|

Y | 51 | 134 | 185 |

| 0.276 | 0.724 | 0.092 |

| 0.028 | 0.838 | |

| 0.025 | 0.067 | |

-------------|-----------|-----------|-----------|

Column Total | 1840 | 160 | 2000 |

| 0.920 | 0.080 | |

-------------|-----------|-----------|-----------|

#Question 4: with best k customer would still be approved  
  
#Question 5: data to be re-run with 50%, 30%, 20%  
  
# Approved @ K =1 K =3 best K   
# 50% : Y, Y, k = 3 @ 96.28  
# 30% : Y, Y, k = 1/3 @ 95.68   
# 20% : Y, Y, k = 3 @ 94.97  
# To run data with new parameters simply insert desired percentage from 0-1 in line 47 of code   
# Part= createDataPartition(Bank$Personal.Loan, \*p=0.6\*, list=FALSE)

# Question 5 cnt…: As we adjust the p ratio for the test we can see that we get fewer and few Y for acceptance for the personal loan. As we look at the accuracy of the model we do not see substantial differences in the model accuracy based on the p ratio. I believe that with the split used that this model remains accurate throughout the entire listed range of training and validation test sets.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cell Contents | | |  | |  | |  | |
| 60% |  |  | |  | |  | |
| Total Observations in Table: 2000 | | | | | | |  | |
|  | PredictedTest\_label4 | | | | |  | |
| Val\_labels | N | Y | | Row Total | |  | |
| ------------- | ----------- | ----------- | | ----------- | |  | |
| N | 1804 | 11 | | 1815 | |  | |
|  | 0.994 | 0.006 | | 0.907 | |  | |
|  | 0.971 | 0.077 | |  | |  | |
|  | 0.902 | 0.005 | |  | |  | |
| ------------- | ----------- | ----------- | | ----------- | |  | |
| Y | 53 | 132 | | 185 | |  | |
|  | 0.286 | 0.714 | | 0.092 | |  | |
|  | 0.029 | 0.923 | |  | |  | |
|  | 0.026 | 0.066 | |  | |  | |
| ------------- | ----------- | ----------- | | ----------- | |  | |
| Column Total | 1857 | 143 | | 2000 | |  | |
|  | 0.928 | 0.071 | |  | |  | |
| ------------- | ----------- | ----------- | | ----------- | |  | |
|  |  |  | |  | |  | |
| 50% |  |  | |  | |  | |
| Total Observations in Table: 2500 | | | | | | |  | |
|  | PredictedTest\_label4 | | | | |  | |
| Val\_labels | N | Y | | Row Total | |  | |
| ------------- | ----------- | ----------- | | ----------- | |  | |
| N | 2246 | 12 | | 2258 | |  | |
|  | 0.995 | 0.005 | | 0.903 | |  | |
|  | 0.965 | 0.069 | |  | |  | |
|  | 0.898 | 0.005 | |  | |  | |
| ------------- | ----------- | ----------- | | ----------- | |  | |
| Y | 81 | 161 | | 242 | |  | |
|  | 0.335 | 0.665 | | 0.097 | |  | |
|  | 0.035 | 0.931 | |  | |  | |
|  | 0.032 | 0.064 | |  | |  | |
| ------------- | ----------- | ----------- | | ----------- | |  | |
| Column Total | 2327 | 173 | | 2500 | |  | |
|  | 0.931 | 0.069 | |  | |  | |
| ------------- | ----------- | ----------- | | ----------- | |  | |
|  |  |  | |  | |  | |
| 30% |  |  | |  | |  | |
| Total Observations in Table: 3500 | | | | | | |  | |
|  | PredictedTest\_label4 | | | | |  | |
| Val\_labels | N | Y | | Row Total | |  | |
| ------------- | ----------- | ----------- | | ----------- | |  | |
| N | 3157 | 22 | | 3179 | |  | |
|  | 0.993 | 0.007 | | 0.908 | |  | |
|  | 0.961 | 0.103 | |  | |  | |
|  | 0.902 | 0.006 | |  | |  | |
| ------------- | ----------- | ----------- | | ----------- | |  | |
| Y | 129 | 192 | | 321 | |  | |
|  | 0.402 | 0.598 | | 0.092 | |  | |
|  | 0.039 | 0.897 | |  | |  | |
|  | 0.037 | 0.055 | |  | |  | |
| ------------- | ----------- | ----------- | | ----------- | |  | |
| Column Total | 3286 | 214 | | 3500 | |  | |
|  | 0.939 | 0.061 | |  | |  | |
|  |  |  | |  | |  | |
| 20% |  |  | |  | |  | |
| Total Observations in Table: 4000 | | | | | | |  | |
|  | PredictedTest\_label4 | | | | |  | |
| Val\_labels | N | Y | | Row Total | |  | |
| ------------- | ----------- | ----------- | | ----------- | |  | |
| N | 3600 | 21 | | 3621 | |  | |
|  | 0.994 | 0.006 | | 0.905 | |  | |
|  | 0.952 | 0.095 | |  | |  | |
|  | 0.9 | 0.005 | |  | |  | |
| ------------- | ----------- | ----------- | | ----------- | |  | |
| Y | 180 | 199 | | 379 | |  | |
|  | 0.475 | 0.525 | | 0.095 | |  | |
|  | 0.048 | 0.905 | |  | |  | |
|  | 0.045 | 0.05 | |  | |  | |
| ------------- | ----------- | ----------- | | ----------- | |  | |
| Column Total | 3780 | 220 | | 4000 | |  | |
|  | 0.945 | 0.055 | |  | |  | |