Assign\_2

Abi

2022-10-02

library('caret')

## Loading required package: ggplot2

## Loading required package: 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('class')  
library('gmodels')  
library('FNN')

##   
## Attaching package: 'FNN'

## The following objects are masked from 'package:class':  
##   
## knn, knn.cv

# Import dataset UniversalBank.csv   
UniversalBank <- read.csv("C:/Users/abinaya/Downloads/UniversalBank.csv")  
#Displaying column names  
colnames(UniversalBank)

## [1] "ID" "Age" "Experience"   
## [4] "Income" "ZIP.Code" "Family"   
## [7] "CCAvg" "Education" "Mortgage"   
## [10] "Personal.Loan" "Securities.Account" "CD.Account"   
## [13] "Online" "CreditCard"

# Summary of UniversalBank dataset  
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

# Making columns ID and ZIP.Code as NULL   
UniversalBank$ID <- NULL  
UniversalBank$ZIP.Code <- NULL  
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 Min. :0.000   
## 1st Qu.: 0.700 1st Qu.:1.000 1st Qu.: 0.0 1st Qu.:0.000   
## Median : 1.500 Median :2.000 Median : 0.0 Median :0.000   
## Mean : 1.938 Mean :1.881 Mean : 56.5 Mean :0.096   
## 3rd Qu.: 2.500 3rd Qu.:3.000 3rd Qu.:101.0 3rd Qu.:0.000   
## Max. :10.000 Max. :3.000 Max. :635.0 Max. :1.000   
## 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

# Making the Personal Loan column as factor   
UniversalBank$Personal.Loan = as.factor(UniversalBank$Personal.Loan)

# Normalization  
Normal\_Data <- preProcess(UniversalBank,method = "range")  
UniversalBank\_Norm <- predict(Normal\_Data,UniversalBank)  
summary(UniversalBank\_Norm)

## Age Experience Income Family   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.2727 1st Qu.:0.2826 1st Qu.:0.1435 1st Qu.:0.0000   
## Median :0.5000 Median :0.5000 Median :0.2593 Median :0.3333   
## Mean :0.5077 Mean :0.5023 Mean :0.3045 Mean :0.4655   
## 3rd Qu.:0.7273 3rd Qu.:0.7174 3rd Qu.:0.4167 3rd Qu.:0.6667   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## CCAvg Education Mortgage Personal.Loan  
## Min. :0.0000 Min. :0.0000 Min. :0.00000 0:4520   
## 1st Qu.:0.0700 1st Qu.:0.0000 1st Qu.:0.00000 1: 480   
## Median :0.1500 Median :0.5000 Median :0.00000   
## Mean :0.1938 Mean :0.4405 Mean :0.08897   
## 3rd Qu.:0.2500 3rd Qu.:1.0000 3rd Qu.:0.15906   
## Max. :1.0000 Max. :1.0000 Max. :1.00000   
## 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

# Partition the data into training 60% and validation 40% sets  
Train\_index <- createDataPartition(UniversalBank$Personal.Loan, p = 0.6, list = FALSE)  
train.df = UniversalBank\_Norm[Train\_index,]  
validation.df = UniversalBank\_Norm[-Train\_index,]

# Classifying the customer as per the date provided   
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

Prediction <- knn(train = train.df[,1:7],test = To\_Predict[,1:7], cl = train.df$Personal.Loan, k = 1)  
print(Prediction)

## [1] 1  
## attr(,"nn.index")  
## [,1]  
## [1,] 2564  
## attr(,"nn.dist")  
## [,1]  
## [1,] 92.36447  
## Levels: 1

# Customer is classified as 1.

# 2) Finding choice of k that balances between overfitting and ignoring the predictor  
set.seed(123)  
UniversalBank\_control <- 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 = UniversalBank\_control)  
knn.model

## k-Nearest Neighbors   
##   
## 3000 samples  
## 11 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Cross-Validated (3 fold, repeated 2 times)   
## Summary of sample sizes: 2000, 2000, 2000, 2000, 2000, 2000, ...   
## Resampling results across tuning parameters:  
##   
## k Accuracy Kappa   
## 1 0.9551667 0.7018414  
## 2 0.9486667 0.6594404  
## 3 0.9493333 0.6359184  
## 4 0.9485000 0.6277238  
## 5 0.9443333 0.5807057  
## 6 0.9411667 0.5499240  
## 7 0.9398333 0.5309776  
## 8 0.9393333 0.5245481  
## 9 0.9371667 0.5023891  
## 10 0.9368333 0.4971245  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was k = 1.

# The choice of K that balances between overfitting and ignoring predictor is K=3

# 3) Confusion matrix  
predictions <- predict(knn.model,validation.df)  
confusionMatrix(predictions,validation.df$Personal.Loan)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1793 58  
## 1 15 134  
##   
## Accuracy : 0.9635   
## 95% CI : (0.9543, 0.9713)  
## No Information Rate : 0.904   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7663   
##   
## Mcnemar's Test P-Value : 8.845e-07   
##   
## Sensitivity : 0.9917   
## Specificity : 0.6979   
## Pos Pred Value : 0.9687   
## Neg Pred Value : 0.8993   
## Prevalence : 0.9040   
## Detection Rate : 0.8965   
## Detection Prevalence : 0.9255   
## Balanced Accuracy : 0.8448   
##   
## 'Positive' Class : 0   
##

# 4) Classify the customer using the best k  
To\_Predict\_Normaliz = 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\_Normaliz = predict(Normal\_Data, To\_Predict)  
predict(knn.model, To\_Predict\_Normaliz)

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

# 5) Repartition the data into 50% for training ,30% for validation, 20% for test  
train\_size = 0.5  
Train\_index = createDataPartition(UniversalBank$Personal.Loan, p = 0.5, list = FALSE)  
train.df = UniversalBank\_Norm[Train\_index,]  
test\_size = 0.2  
Test\_index = createDataPartition(UniversalBank$Personal.Loan, p = 0.2, list = FALSE)  
Test.df = UniversalBank\_Norm[Test\_index,]  
valid\_size = 0.3  
Validation\_index = createDataPartition(UniversalBank$Personal.Loan, p = 0.3, list = FALSE)  
validation.df = UniversalBank\_Norm[Validation\_index,]  
Testingknn <- knn(train = train.df[,-8], test = Test.df[,-8], cl = train.df[,8], k =3)  
Validationknn <- knn(train = train.df[,-8], test = validation.df[,-8], cl = train.df[,8], k =3)  
Trainingknn <- knn(train = train.df[,-8], test = train.df[,-8], cl = train.df[,8], k =3)  
  
# Comparing the confusion matrix of the test set with the training and validation sets.  
confusionMatrix(Testingknn, Test.df[,8])

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 900 36  
## 1 4 60  
##   
## Accuracy : 0.96   
## 95% CI : (0.9459, 0.9713)  
## No Information Rate : 0.904   
## P-Value [Acc > NIR] : 1.476e-11   
##   
## Kappa : 0.7292   
##   
## Mcnemar's Test P-Value : 9.509e-07   
##   
## Sensitivity : 0.9956   
## Specificity : 0.6250   
## Pos Pred Value : 0.9615   
## Neg Pred Value : 0.9375   
## Prevalence : 0.9040   
## Detection Rate : 0.9000   
## Detection Prevalence : 0.9360   
## Balanced Accuracy : 0.8103   
##   
## 'Positive' Class : 0   
##

confusionMatrix(Trainingknn, train.df[,8])

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 2254 51  
## 1 6 189  
##   
## Accuracy : 0.9772   
## 95% CI : (0.9706, 0.9827)  
## No Information Rate : 0.904   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.8566   
##   
## Mcnemar's Test P-Value : 5.611e-09   
##   
## Sensitivity : 0.9973   
## Specificity : 0.7875   
## Pos Pred Value : 0.9779   
## Neg Pred Value : 0.9692   
## Prevalence : 0.9040   
## Detection Rate : 0.9016   
## Detection Prevalence : 0.9220   
## Balanced Accuracy : 0.8924   
##   
## 'Positive' Class : 0   
##

confusionMatrix(Validationknn, validation.df[,8])

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 1347 45  
## 1 9 99  
##   
## Accuracy : 0.964   
## 95% CI : (0.9533, 0.9728)  
## No Information Rate : 0.904   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7665   
##   
## Mcnemar's Test P-Value : 1.908e-06   
##   
## Sensitivity : 0.9934   
## Specificity : 0.6875   
## Pos Pred Value : 0.9677   
## Neg Pred Value : 0.9167   
## Prevalence : 0.9040   
## Detection Rate : 0.8980   
## Detection Prevalence : 0.9280   
## Balanced Accuracy : 0.8404   
##   
## 'Positive' Class : 0   
##