Appendix 3 - Customer’s decision to open a term deposit account

## Part 1 - KNN Model

### Step 1 : Collecting Data

The data set is collected in CSV format from UCI.edu and below is the reference for the same.

S. Moro, P. Cortez, and P. Rita. (2014) A data-driven approach to predictthe success of bank telemarketing. Decision Support Systems,. [Online].Available: <https://archive.ics.uci.edu/ml/datasets/bank> marketing

### Step 2 : Exploring, preprocessing and cleaning the data

Primary setup

knitr::opts\_knit$set(root.dir = '/Users/sobil/Documents/MSC/Sem 1/Data Mining & Machine Learning/Project/Bank/')  
remove(list = ls())  
set.seed(1)

loading all the libraries required

library(gmodels)  
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

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(fastDummies)

#### 1) reading the raw csv file

bank <- read.csv("bank-additional-full.csv", sep = ";")

#### 2) exploratory analysis

Structure of the bank data frame

str(bank)

## 'data.frame': 41188 obs. of 21 variables:  
## $ age : int 56 57 37 40 56 45 59 41 24 25 ...  
## $ job : Factor w/ 12 levels "admin.","blue-collar",..: 4 8 8 1 8 8 1 2 10 8 ...  
## $ marital : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2 2 2 3 3 ...  
## $ education : Factor w/ 8 levels "basic.4y","basic.6y",..: 1 4 4 2 4 3 6 8 6 4 ...  
## $ default : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1 1 ...  
## $ housing : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3 3 ...  
## $ loan : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 3 1 1 1 1 1 ...  
## $ contact : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...  
## $ month : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7 7 7 7 ...  
## $ day\_of\_week : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ duration : int 261 149 226 151 307 198 139 217 380 50 ...  
## $ campaign : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ pdays : int 999 999 999 999 999 999 999 999 999 999 ...  
## $ previous : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ poutcome : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ emp.var.rate : num 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...  
## $ cons.price.idx: num 94 94 94 94 94 ...  
## $ cons.conf.idx : num -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...  
## $ euribor3m : num 4.86 4.86 4.86 4.86 4.86 ...  
## $ nr.employed : num 5191 5191 5191 5191 5191 ...  
## $ y : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...

Summary of the bank data frame

summary(bank)

## age job marital   
## Min. :17.00 admin. :10422 divorced: 4612   
## 1st Qu.:32.00 blue-collar: 9254 married :24928   
## Median :38.00 technician : 6743 single :11568   
## Mean :40.02 services : 3969 unknown : 80   
## 3rd Qu.:47.00 management : 2924   
## Max. :98.00 retired : 1720   
## (Other) : 6156   
## education default housing loan   
## university.degree :12168 no :32588 no :18622 no :33950   
## high.school : 9515 unknown: 8597 unknown: 990 unknown: 990   
## basic.9y : 6045 yes : 3 yes :21576 yes : 6248   
## professional.course: 5243   
## basic.4y : 4176   
## basic.6y : 2292   
## (Other) : 1749   
## contact month day\_of\_week duration   
## cellular :26144 may :13769 fri:7827 Min. : 0.0   
## telephone:15044 jul : 7174 mon:8514 1st Qu.: 102.0   
## aug : 6178 thu:8623 Median : 180.0   
## jun : 5318 tue:8090 Mean : 258.3   
## nov : 4101 wed:8134 3rd Qu.: 319.0   
## apr : 2632 Max. :4918.0   
## (Other): 2016   
## campaign pdays previous poutcome   
## Min. : 1.000 Min. : 0.0 Min. :0.000 failure : 4252   
## 1st Qu.: 1.000 1st Qu.:999.0 1st Qu.:0.000 nonexistent:35563   
## Median : 2.000 Median :999.0 Median :0.000 success : 1373   
## Mean : 2.568 Mean :962.5 Mean :0.173   
## 3rd Qu.: 3.000 3rd Qu.:999.0 3rd Qu.:0.000   
## Max. :56.000 Max. :999.0 Max. :7.000   
##   
## emp.var.rate cons.price.idx cons.conf.idx euribor3m   
## Min. :-3.40000 Min. :92.20 Min. :-50.8 Min. :0.634   
## 1st Qu.:-1.80000 1st Qu.:93.08 1st Qu.:-42.7 1st Qu.:1.344   
## Median : 1.10000 Median :93.75 Median :-41.8 Median :4.857   
## Mean : 0.08189 Mean :93.58 Mean :-40.5 Mean :3.621   
## 3rd Qu.: 1.40000 3rd Qu.:93.99 3rd Qu.:-36.4 3rd Qu.:4.961   
## Max. : 1.40000 Max. :94.77 Max. :-26.9 Max. :5.045   
##   
## nr.employed y   
## Min. :4964 no :36548   
## 1st Qu.:5099 yes: 4640   
## Median :5191   
## Mean :5167   
## 3rd Qu.:5228   
## Max. :5228   
##

checking the overall distribution of prediction/ target charcterstics/feature

gmodels::CrossTable(bank$y)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 41188   
##   
##   
## | no | yes |   
## |-----------|-----------|  
## | 36548 | 4640 |   
## | 0.887 | 0.113 |   
## |-----------|-----------|  
##   
##   
##   
##

converting the target feature into factor with adding labels for more information

bank$y <- factor(bank$y, levels = c("yes", "no"), labels = c("Yes", "No"))

cheking NAs

apply(X = bank,MARGIN = 2, FUN = function(col) any(is.na(col))) # no NAs

## age job marital education default   
## FALSE FALSE FALSE FALSE FALSE   
## housing loan contact month day\_of\_week   
## FALSE FALSE FALSE FALSE FALSE   
## duration campaign pdays previous poutcome   
## FALSE FALSE FALSE FALSE FALSE   
## emp.var.rate cons.price.idx cons.conf.idx euribor3m nr.employed   
## FALSE FALSE FALSE FALSE FALSE   
## y   
## FALSE

### Step 3.1 - Data transformation & preparation - Normalizing the data with custom function

normalizing the data to standardize range of values of various characterstics #### 1) creating normalized method

nor <- function(val) {  
 return((val - min(val))/(max(val) - min(val)))  
}

getting normalised variables in a data frame

bank\_n\_cont <- as.data.frame(lapply(bank[c(1,11,12,13,14,16,17,18,19,20)], nor))  
summary(bank\_n\_cont)

## age duration campaign pdays   
## Min. :0.0000 Min. :0.00000 Min. :0.00000 Min. :0.0000   
## 1st Qu.:0.1852 1st Qu.:0.02074 1st Qu.:0.00000 1st Qu.:1.0000   
## Median :0.2593 Median :0.03660 Median :0.01818 Median :1.0000   
## Mean :0.2842 Mean :0.05252 Mean :0.02850 Mean :0.9634   
## 3rd Qu.:0.3704 3rd Qu.:0.06486 3rd Qu.:0.03636 3rd Qu.:1.0000   
## Max. :1.0000 Max. :1.00000 Max. :1.00000 Max. :1.0000   
## previous emp.var.rate cons.price.idx cons.conf.idx   
## Min. :0.00000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.00000 1st Qu.:0.3333 1st Qu.:0.3406 1st Qu.:0.3389   
## Median :0.00000 Median :0.9375 Median :0.6033 Median :0.3766   
## Mean :0.02471 Mean :0.7254 Mean :0.5357 Mean :0.4309   
## 3rd Qu.:0.00000 3rd Qu.:1.0000 3rd Qu.:0.6988 3rd Qu.:0.6025   
## Max. :1.00000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## euribor3m nr.employed   
## Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.1610 1st Qu.:0.5123   
## Median :0.9574 Median :0.8597   
## Mean :0.6772 Mean :0.7691   
## 3rd Qu.:0.9810 3rd Qu.:1.0000   
## Max. :1.0000 Max. :1.0000

#### 2) creating a new df of only independent variables with normalised columns

bank\_n <- bank[,-21]  
bank\_n$age <- bank\_n\_cont$age  
bank\_n$duration <- bank\_n\_cont$duration  
bank\_n$campaign <- bank\_n\_cont$campaign  
bank\_n$pdays <- bank\_n\_cont$pdays  
bank\_n$previous <- bank\_n\_cont$previous  
bank\_n$emp.var.rate <- bank\_n\_cont$emp.var.rate  
bank\_n$cons.price.idx <- bank\_n\_cont$cons.price.idx  
bank\_n$cons.conf.idx <- bank\_n\_cont$cons.conf.idx  
bank\_n$euribor3m <- bank\_n\_cont$euribor3m  
bank\_n$nr.employed <- bank\_n\_cont$nr.employed  
summary(bank\_n)

## age job marital   
## Min. :0.0000 admin. :10422 divorced: 4612   
## 1st Qu.:0.1852 blue-collar: 9254 married :24928   
## Median :0.2593 technician : 6743 single :11568   
## Mean :0.2842 services : 3969 unknown : 80   
## 3rd Qu.:0.3704 management : 2924   
## Max. :1.0000 retired : 1720   
## (Other) : 6156   
## education default housing loan   
## university.degree :12168 no :32588 no :18622 no :33950   
## high.school : 9515 unknown: 8597 unknown: 990 unknown: 990   
## basic.9y : 6045 yes : 3 yes :21576 yes : 6248   
## professional.course: 5243   
## basic.4y : 4176   
## basic.6y : 2292   
## (Other) : 1749   
## contact month day\_of\_week duration   
## cellular :26144 may :13769 fri:7827 Min. :0.00000   
## telephone:15044 jul : 7174 mon:8514 1st Qu.:0.02074   
## aug : 6178 thu:8623 Median :0.03660   
## jun : 5318 tue:8090 Mean :0.05252   
## nov : 4101 wed:8134 3rd Qu.:0.06486   
## apr : 2632 Max. :1.00000   
## (Other): 2016   
## campaign pdays previous poutcome   
## Min. :0.00000 Min. :0.0000 Min. :0.00000 failure : 4252   
## 1st Qu.:0.00000 1st Qu.:1.0000 1st Qu.:0.00000 nonexistent:35563   
## Median :0.01818 Median :1.0000 Median :0.00000 success : 1373   
## Mean :0.02850 Mean :0.9634 Mean :0.02471   
## 3rd Qu.:0.03636 3rd Qu.:1.0000 3rd Qu.:0.00000   
## Max. :1.00000 Max. :1.0000 Max. :1.00000   
##   
## emp.var.rate cons.price.idx cons.conf.idx euribor3m   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.3333 1st Qu.:0.3406 1st Qu.:0.3389 1st Qu.:0.1610   
## Median :0.9375 Median :0.6033 Median :0.3766 Median :0.9574   
## Mean :0.7254 Mean :0.5357 Mean :0.4309 Mean :0.6772   
## 3rd Qu.:1.0000 3rd Qu.:0.6988 3rd Qu.:0.6025 3rd Qu.:0.9810   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
##   
## nr.employed   
## Min. :0.0000   
## 1st Qu.:0.5123   
## Median :0.8597   
## Mean :0.7691   
## 3rd Qu.:1.0000   
## Max. :1.0000   
##

#### 3) creating dummy columns for categorical columns

bank\_n <- fastDummies::dummy\_cols(bank\_n, remove\_first\_dummy = TRUE) # remove\_first\_dummy = TRUE to avoid multi-collinearity

removing categorical columns from df (dummy are included)

bank\_n <- bank\_n[,-c(2:10,15)]

#### 4) creating training and testing dataset from exisitng sample

indx <- createDataPartition(bank$y, p = 0.8, list = FALSE)  
bank\_train <- bank\_n[indx,]  
bank\_test <- bank\_n[- indx,]

creating lables for test and training data sets

bank\_train\_labels <- bank[indx,21]  
bank\_test\_labels <- bank[- indx,21]

### Step 4.1 : Training a model on the data - Normalizing the data with custom function

using the knn method of class package with K value equalent to square root of total train observation & odd number to eliminate tie vote issue for 2 factor classification i.e.

bank$y %>% length %>% sqrt %>% round # 203

## [1] 203

bank\_test\_predict <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 203)

### Step 5.1 : Evaluating the model - Normalizing the data with custom function

“Model 1 : k = 203”

confusionMatrix(data = bank\_test\_predict,reference = bank\_test\_labels)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 143 35  
## No 785 7274  
##   
## Accuracy : 0.9004   
## 95% CI : (0.8938, 0.9068)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 7.001e-05   
##   
## Kappa : 0.2307   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.15409   
## Specificity : 0.99521   
## Pos Pred Value : 0.80337   
## Neg Pred Value : 0.90259   
## Prevalence : 0.11266   
## Detection Rate : 0.01736   
## Detection Prevalence : 0.02161   
## Balanced Accuracy : 0.57465   
##   
## 'Positive' Class : Yes   
##

### Step 6.1 : Improving the model

#### 1) Changing k values

bank\_test\_predict2 <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 151)  
print("Model 2 : k = 151")

## [1] "Model 2 : k = 151"

print(confusionMatrix(data = bank\_test\_predict2,reference = bank\_test\_labels))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 144 43  
## No 784 7266  
##   
## Accuracy : 0.8996   
## 95% CI : (0.8929, 0.906)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.0001886   
##   
## Kappa : 0.2292   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.15517   
## Specificity : 0.99412   
## Pos Pred Value : 0.77005   
## Neg Pred Value : 0.90261   
## Prevalence : 0.11266   
## Detection Rate : 0.01748   
## Detection Prevalence : 0.02270   
## Balanced Accuracy : 0.57464   
##   
## 'Positive' Class : Yes   
##

predict = 3 : k = 101

bank\_test\_predict3 <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 101)  
print("Model 3 : k = 101")

## [1] "Model 3 : k = 101"

print(confusionMatrix(data = bank\_test\_predict3,reference = bank\_test\_labels))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 164 45  
## No 764 7264  
##   
## Accuracy : 0.9018   
## 95% CI : (0.8952, 0.9081)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.298e-05   
##   
## Kappa : 0.2577   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.17672   
## Specificity : 0.99384   
## Pos Pred Value : 0.78469   
## Neg Pred Value : 0.90483   
## Prevalence : 0.11266   
## Detection Rate : 0.01991   
## Detection Prevalence : 0.02537   
## Balanced Accuracy : 0.58528   
##   
## 'Positive' Class : Yes   
##

bank\_test\_predict4 <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 75)  
print("Model 4 : k = 75")

## [1] "Model 4 : k = 75"

print(confusionMatrix(data = bank\_test\_predict4,reference = bank\_test\_labels))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 164 47  
## No 764 7262  
##   
## Accuracy : 0.9015   
## 95% CI : (0.8949, 0.9079)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.785e-05   
##   
## Kappa : 0.257   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.17672   
## Specificity : 0.99357   
## Pos Pred Value : 0.77725   
## Neg Pred Value : 0.90481   
## Prevalence : 0.11266   
## Detection Rate : 0.01991   
## Detection Prevalence : 0.02562   
## Balanced Accuracy : 0.58515   
##   
## 'Positive' Class : Yes   
##

bank\_test\_predict5 <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 51)  
print("Model 5 : k = 51")

## [1] "Model 5 : k = 51"

print(confusionMatrix(data = bank\_test\_predict5,reference = bank\_test\_labels))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 173 52  
## No 755 7257  
##   
## Accuracy : 0.902   
## 95% CI : (0.8954, 0.9084)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 9.396e-06   
##   
## Kappa : 0.2679   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.18642   
## Specificity : 0.99289   
## Pos Pred Value : 0.76889   
## Neg Pred Value : 0.90577   
## Prevalence : 0.11266   
## Detection Rate : 0.02100   
## Detection Prevalence : 0.02732   
## Balanced Accuracy : 0.58965   
##   
## 'Positive' Class : Yes   
##

bank\_test\_predict6 <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 31)  
print("Model 6 : k = 31")

## [1] "Model 6 : k = 31"

print(confusionMatrix(data = bank\_test\_predict6,reference = bank\_test\_labels))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 179 63  
## No 749 7246  
##   
## Accuracy : 0.9014   
## 95% CI : (0.8948, 0.9078)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 2.088e-05   
##   
## Kappa : 0.2721   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.19289   
## Specificity : 0.99138   
## Pos Pred Value : 0.73967   
## Neg Pred Value : 0.90632   
## Prevalence : 0.11266   
## Detection Rate : 0.02173   
## Detection Prevalence : 0.02938   
## Balanced Accuracy : 0.59213   
##   
## 'Positive' Class : Yes   
##

bank\_test\_predict7 <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 21)  
print("Model 7 : k = 21")

## [1] "Model 7 : k = 21"

print(confusionMatrix(data = bank\_test\_predict7,reference = bank\_test\_labels))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 193 89  
## No 735 7220  
##   
## Accuracy : 0.9   
## 95% CI : (0.8933, 0.9064)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.0001243   
##   
## Kappa : 0.2813   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.20797   
## Specificity : 0.98782   
## Pos Pred Value : 0.68440   
## Neg Pred Value : 0.90761   
## Prevalence : 0.11266   
## Detection Rate : 0.02343   
## Detection Prevalence : 0.03424   
## Balanced Accuracy : 0.59790   
##   
## 'Positive' Class : Yes   
##

bank\_test\_predict8 <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 11)  
print("Model 8 : k = 11")

## [1] "Model 8 : k = 11"

print(confusionMatrix(data = bank\_test\_predict8,reference = bank\_test\_labels))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 219 130  
## No 709 7179  
##   
## Accuracy : 0.8981   
## 95% CI : (0.8914, 0.9046)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.0008916   
##   
## Kappa : 0.2999   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.23599   
## Specificity : 0.98221   
## Pos Pred Value : 0.62751   
## Neg Pred Value : 0.91012   
## Prevalence : 0.11266   
## Detection Rate : 0.02659   
## Detection Prevalence : 0.04237   
## Balanced Accuracy : 0.60910   
##   
## 'Positive' Class : Yes   
##

Model n : k = 9 to 1 (only odd numbers)

i <- 9  
while (i > 0) {  
 bank\_test\_predict\_n <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = i)  
 print(i)  
 print(confusionMatrix(data = bank\_test\_predict\_n,reference = bank\_test\_labels))  
 i <- i - 2  
}

## [1] 9  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 226 148  
## No 702 7161  
##   
## Accuracy : 0.8968   
## 95% CI : (0.89, 0.9033)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.003164   
##   
## Kappa : 0.302   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.24353   
## Specificity : 0.97975   
## Pos Pred Value : 0.60428   
## Neg Pred Value : 0.91072   
## Prevalence : 0.11266   
## Detection Rate : 0.02744   
## Detection Prevalence : 0.04540   
## Balanced Accuracy : 0.61164   
##   
## 'Positive' Class : Yes   
##   
## [1] 7  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 237 152  
## No 691 7157  
##   
## Accuracy : 0.8977   
## 95% CI : (0.8909, 0.9041)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.001438   
##   
## Kappa : 0.3143   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.25539   
## Specificity : 0.97920   
## Pos Pred Value : 0.60925   
## Neg Pred Value : 0.91195   
## Prevalence : 0.11266   
## Detection Rate : 0.02877   
## Detection Prevalence : 0.04723   
## Balanced Accuracy : 0.61730   
##   
## 'Positive' Class : Yes   
##   
## [1] 5  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 255 195  
## No 673 7114  
##   
## Accuracy : 0.8946   
## 95% CI : (0.8878, 0.9012)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.01836   
##   
## Kappa : 0.3201   
##   
## Mcnemar's Test P-Value : < 2e-16   
##   
## Sensitivity : 0.27478   
## Specificity : 0.97332   
## Pos Pred Value : 0.56667   
## Neg Pred Value : 0.91357   
## Prevalence : 0.11266   
## Detection Rate : 0.03096   
## Detection Prevalence : 0.05463   
## Balanced Accuracy : 0.62405   
##   
## 'Positive' Class : Yes   
##   
## [1] 3  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 283 263  
## No 645 7046  
##   
## Accuracy : 0.8898   
## 95% CI : (0.8828, 0.8965)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.2492   
##   
## Kappa : 0.3279   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.30496   
## Specificity : 0.96402   
## Pos Pred Value : 0.51832   
## Neg Pred Value : 0.91614   
## Prevalence : 0.11266   
## Detection Rate : 0.03436   
## Detection Prevalence : 0.06629   
## Balanced Accuracy : 0.63449   
##   
## 'Positive' Class : Yes   
##   
## [1] 1  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 325 464  
## No 603 6845  
##   
## Accuracy : 0.8705   
## 95% CI : (0.863, 0.8776)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.3068   
##   
## Mcnemar's Test P-Value : 2.392e-05   
##   
## Sensitivity : 0.35022   
## Specificity : 0.93652   
## Pos Pred Value : 0.41191   
## Neg Pred Value : 0.91904   
## Prevalence : 0.11266   
## Detection Rate : 0.03946   
## Detection Prevalence : 0.09579   
## Balanced Accuracy : 0.64337   
##   
## 'Positive' Class : Yes   
##

Model n : k = 19 to 13 (only odd numbers)

i <- 19  
while (i > 11) {  
 bank\_test\_predict\_n <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = i)  
 print(i)  
 print(confusionMatrix(data = bank\_test\_predict\_n,reference = bank\_test\_labels))  
 i <- i - 2  
}

## [1] 19  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 202 91  
## No 726 7218  
##   
## Accuracy : 0.9008   
## 95% CI : (0.8942, 0.9072)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 4.491e-05   
##   
## Kappa : 0.2926   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.21767   
## Specificity : 0.98755   
## Pos Pred Value : 0.68942   
## Neg Pred Value : 0.90861   
## Prevalence : 0.11266   
## Detection Rate : 0.02452   
## Detection Prevalence : 0.03557   
## Balanced Accuracy : 0.60261   
##   
## 'Positive' Class : Yes   
##   
## [1] 17  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 202 102  
## No 726 7207  
##   
## Accuracy : 0.8995   
## 95% CI : (0.8928, 0.9059)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.0002161   
##   
## Kappa : 0.2884   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.21767   
## Specificity : 0.98604   
## Pos Pred Value : 0.66447   
## Neg Pred Value : 0.90848   
## Prevalence : 0.11266   
## Detection Rate : 0.02452   
## Detection Prevalence : 0.03691   
## Balanced Accuracy : 0.60186   
##   
## 'Positive' Class : Yes   
##   
## [1] 15  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 211 110  
## No 717 7199  
##   
## Accuracy : 0.8996   
## 95% CI : (0.8929, 0.906)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.0001886   
##   
## Kappa : 0.2972   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.22737   
## Specificity : 0.98495   
## Pos Pred Value : 0.65732   
## Neg Pred Value : 0.90942   
## Prevalence : 0.11266   
## Detection Rate : 0.02562   
## Detection Prevalence : 0.03897   
## Balanced Accuracy : 0.60616   
##   
## 'Positive' Class : Yes   
##   
## [1] 13  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 213 118  
## No 715 7191  
##   
## Accuracy : 0.8989   
## 95% CI : (0.8922, 0.9053)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.0004194   
##   
## Kappa : 0.2967   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.22953   
## Specificity : 0.98386   
## Pos Pred Value : 0.64350   
## Neg Pred Value : 0.90956   
## Prevalence : 0.11266   
## Detection Rate : 0.02586   
## Detection Prevalence : 0.04018   
## Balanced Accuracy : 0.60669   
##   
## 'Positive' Class : Yes   
##

Model n : k = 29 to 23 (only odd numbers)

i <- 29  
while (i > 21) {  
 bank\_test\_predict\_n <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = i)  
 print(i)  
 print(confusionMatrix(data = bank\_test\_predict\_n,reference = bank\_test\_labels))  
 i <- i - 2  
}

## [1] 29  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 184 65  
## No 744 7244  
##   
## Accuracy : 0.9018   
## 95% CI : (0.8952, 0.9081)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.298e-05   
##   
## Kappa : 0.2783   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.19828   
## Specificity : 0.99111   
## Pos Pred Value : 0.73896   
## Neg Pred Value : 0.90686   
## Prevalence : 0.11266   
## Detection Rate : 0.02234   
## Detection Prevalence : 0.03023   
## Balanced Accuracy : 0.59469   
##   
## 'Positive' Class : Yes   
##   
## [1] 27  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 188 71  
## No 740 7238  
##   
## Accuracy : 0.9015   
## 95% CI : (0.8949, 0.9079)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.785e-05   
##   
## Kappa : 0.2814   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.20259   
## Specificity : 0.99029   
## Pos Pred Value : 0.72587   
## Neg Pred Value : 0.90724   
## Prevalence : 0.11266   
## Detection Rate : 0.02282   
## Detection Prevalence : 0.03144   
## Balanced Accuracy : 0.59644   
##   
## 'Positive' Class : Yes   
##   
## [1] 25  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 189 76  
## No 739 7233  
##   
## Accuracy : 0.9011   
## 95% CI : (0.8944, 0.9074)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 3.319e-05   
##   
## Kappa : 0.2809   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.20366   
## Specificity : 0.98960   
## Pos Pred Value : 0.71321   
## Neg Pred Value : 0.90730   
## Prevalence : 0.11266   
## Detection Rate : 0.02295   
## Detection Prevalence : 0.03217   
## Balanced Accuracy : 0.59663   
##   
## 'Positive' Class : Yes   
##   
## [1] 23  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 191 79  
## No 737 7230  
##   
## Accuracy : 0.9009   
## 95% CI : (0.8943, 0.9073)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 3.864e-05   
##   
## Kappa : 0.2824   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.20582   
## Specificity : 0.98919   
## Pos Pred Value : 0.70741   
## Neg Pred Value : 0.90749   
## Prevalence : 0.11266   
## Detection Rate : 0.02319   
## Detection Prevalence : 0.03278   
## Balanced Accuracy : 0.59751   
##   
## 'Positive' Class : Yes   
##

#### 2) Best among all is for K = 3

bank\_test\_predict.best <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 3)  
print(confusionMatrix(data = bank\_test\_predict.best,reference = bank\_test\_labels))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 283 262  
## No 645 7047  
##   
## Accuracy : 0.8899   
## 95% CI : (0.8829, 0.8966)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.2382   
##   
## Kappa : 0.3282   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.30496   
## Specificity : 0.96415   
## Pos Pred Value : 0.51927   
## Neg Pred Value : 0.91615   
## Prevalence : 0.11266   
## Detection Rate : 0.03436   
## Detection Prevalence : 0.06616   
## Balanced Accuracy : 0.63456   
##   
## 'Positive' Class : Yes   
##

### Step 3.2 - Data transformation & preparation - Normalizing the data with inbuilt R function

Redoing the first 2 steps before transformation

source("/Users/sobil/Documents/MSC/Sem 1/Data Mining & Machine Learning/Project/Bank/bank\_import\_primaryExplore.R")

## 'data.frame': 41188 obs. of 21 variables:  
## $ age : int 56 57 37 40 56 45 59 41 24 25 ...  
## $ job : Factor w/ 12 levels "admin.","blue-collar",..: 4 8 8 1 8 8 1 2 10 8 ...  
## $ marital : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2 2 2 3 3 ...  
## $ education : Factor w/ 8 levels "basic.4y","basic.6y",..: 1 4 4 2 4 3 6 8 6 4 ...  
## $ default : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1 1 ...  
## $ housing : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3 3 ...  
## $ loan : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 3 1 1 1 1 1 ...  
## $ contact : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...  
## $ month : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7 7 7 7 ...  
## $ day\_of\_week : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ duration : int 261 149 226 151 307 198 139 217 380 50 ...  
## $ campaign : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ pdays : int 999 999 999 999 999 999 999 999 999 999 ...  
## $ previous : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ poutcome : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ emp.var.rate : num 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...  
## $ cons.price.idx: num 94 94 94 94 94 ...  
## $ cons.conf.idx : num -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...  
## $ euribor3m : num 4.86 4.86 4.86 4.86 4.86 ...  
## $ nr.employed : num 5191 5191 5191 5191 5191 ...  
## $ y : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 41188   
##   
##   
## | no | yes |   
## |-----------|-----------|  
## | 36548 | 4640 |   
## | 0.887 | 0.113 |   
## |-----------|-----------|  
##   
##   
##   
##

normalizing the data to standardize range of values of various characterstics

getting normalised variables in a data frame

#### 1) Normalization using scale inbulit R method based upon using z score

bank\_n\_cont <- as.data.frame(scale(bank[c(1,11,12,13,14,16,17,18,19,20)]))  
summary(bank\_n\_cont)

## age duration campaign pdays   
## Min. :-2.2093 Min. :-0.9962 Min. :-0.5659 Min. :-5.1494   
## 1st Qu.:-0.7700 1st Qu.:-0.6028 1st Qu.:-0.5659 1st Qu.: 0.1954   
## Median :-0.1942 Median :-0.3019 Median :-0.2049 Median : 0.1954   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.6694 3rd Qu.: 0.2342 3rd Qu.: 0.1561 3rd Qu.: 0.1954   
## Max. : 5.5632 Max. :17.9718 Max. :19.2896 Max. : 0.1954   
## previous emp.var.rate cons.price.idx cons.conf.idx   
## Min. :-0.3495 Min. :-2.2164 Min. :-2.3749 Min. :-2.2249   
## 1st Qu.:-0.3495 1st Qu.:-1.1979 1st Qu.:-0.8649 1st Qu.:-0.4748   
## Median :-0.3495 Median : 0.6481 Median : 0.2995 Median :-0.2803   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.:-0.3495 3rd Qu.: 0.8391 3rd Qu.: 0.7227 3rd Qu.: 0.8864   
## Max. :13.7948 Max. : 0.8391 Max. : 2.0581 Max. : 2.9391   
## euribor3m nr.employed   
## Min. :-1.7223 Min. :-2.8157   
## 1st Qu.:-1.3130 1st Qu.:-0.9403   
## Median : 0.7125 Median : 0.3317   
## Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.7724 3rd Qu.: 0.8452   
## Max. : 0.8208 Max. : 0.8452

#### 2) creating a new df of only independent variables

bank\_n <- bank[,-21]  
bank\_n$age <- bank\_n\_cont$age  
bank\_n$duration <- bank\_n\_cont$duration  
bank\_n$campaign <- bank\_n\_cont$campaign  
bank\_n$pdays <- bank\_n\_cont$pdays  
bank\_n$previous <- bank\_n\_cont$previous  
bank\_n$emp.var.rate <- bank\_n\_cont$emp.var.rate  
bank\_n$cons.price.idx <- bank\_n\_cont$cons.price.idx  
bank\_n$cons.conf.idx <- bank\_n\_cont$cons.conf.idx  
bank\_n$euribor3m <- bank\_n\_cont$euribor3m  
bank\_n$nr.employed <- bank\_n\_cont$nr.employed  
summary(bank\_n)

## age job marital   
## Min. :-2.2093 admin. :10422 divorced: 4612   
## 1st Qu.:-0.7700 blue-collar: 9254 married :24928   
## Median :-0.1942 technician : 6743 single :11568   
## Mean : 0.0000 services : 3969 unknown : 80   
## 3rd Qu.: 0.6694 management : 2924   
## Max. : 5.5632 retired : 1720   
## (Other) : 6156   
## education default housing loan   
## university.degree :12168 no :32588 no :18622 no :33950   
## high.school : 9515 unknown: 8597 unknown: 990 unknown: 990   
## basic.9y : 6045 yes : 3 yes :21576 yes : 6248   
## professional.course: 5243   
## basic.4y : 4176   
## basic.6y : 2292   
## (Other) : 1749   
## contact month day\_of\_week duration   
## cellular :26144 may :13769 fri:7827 Min. :-0.9962   
## telephone:15044 jul : 7174 mon:8514 1st Qu.:-0.6028   
## aug : 6178 thu:8623 Median :-0.3019   
## jun : 5318 tue:8090 Mean : 0.0000   
## nov : 4101 wed:8134 3rd Qu.: 0.2342   
## apr : 2632 Max. :17.9718   
## (Other): 2016   
## campaign pdays previous poutcome   
## Min. :-0.5659 Min. :-5.1494 Min. :-0.3495 failure : 4252   
## 1st Qu.:-0.5659 1st Qu.: 0.1954 1st Qu.:-0.3495 nonexistent:35563   
## Median :-0.2049 Median : 0.1954 Median :-0.3495 success : 1373   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.1561 3rd Qu.: 0.1954 3rd Qu.:-0.3495   
## Max. :19.2896 Max. : 0.1954 Max. :13.7948   
##   
## emp.var.rate cons.price.idx cons.conf.idx euribor3m   
## Min. :-2.2164 Min. :-2.3749 Min. :-2.2249 Min. :-1.7223   
## 1st Qu.:-1.1979 1st Qu.:-0.8649 1st Qu.:-0.4748 1st Qu.:-1.3130   
## Median : 0.6481 Median : 0.2995 Median :-0.2803 Median : 0.7125   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.8391 3rd Qu.: 0.7227 3rd Qu.: 0.8864 3rd Qu.: 0.7724   
## Max. : 0.8391 Max. : 2.0581 Max. : 2.9391 Max. : 0.8208   
##   
## nr.employed   
## Min. :-2.8157   
## 1st Qu.:-0.9403   
## Median : 0.3317   
## Mean : 0.0000   
## 3rd Qu.: 0.8452   
## Max. : 0.8452   
##

#### 3) creating dummy columns for categorical columns

bank\_n <- fastDummies::dummy\_cols(bank\_n, remove\_first\_dummy = TRUE) # remove\_first\_dummy = TRUE to avoid multi-collinearity

removing categorical columns from df (dummy are included)

bank\_n <- bank\_n[,-c(2:10,15)]

#### 4) creating training and testing dataset from exisitng sample

indx <- createDataPartition(bank$y, p = 0.8, list = FALSE)  
bank\_train <- bank\_n[indx,]  
bank\_test <- bank\_n[- indx,]

creating lables for test and training data sets

bank\_train\_labels <- bank[indx,21]  
bank\_test\_labels <- bank[- indx,21]

### Step 4.2 : Training a model on the data - Normalizing the data with inbuilt R function

using the knn method of class package with K odd

bank\_test\_predict <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 51)

### Step 5.2 : Evaluating the model - Normalizing the data with inbuilt R function

“Model 1 : k = 51”

confusionMatrix(data = bank\_test\_predict,reference = bank\_test\_labels)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 365 150  
## No 563 7159  
##   
## Accuracy : 0.9134   
## 95% CI : (0.9072, 0.9194)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 4.566e-15   
##   
## Kappa : 0.4627   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.39332   
## Specificity : 0.97948   
## Pos Pred Value : 0.70874   
## Neg Pred Value : 0.92709   
## Prevalence : 0.11266   
## Detection Rate : 0.04431   
## Detection Prevalence : 0.06252   
## Balanced Accuracy : 0.68640   
##   
## 'Positive' Class : Yes   
##

### Step 6.2 : Improving the model - Normalizing the data with inbuilt R function

#### 1) Changing k values

Skipped model for K values which were higher than 51 (please refer the R script for model performance) - overall the performance was poor than below

Model n : k = 31 to 1 (only odd numbers)"

i <- 31  
while (i > 0) {  
 bank\_test\_predict\_n <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = i)  
 print(i)  
 print(confusionMatrix(data = bank\_test\_predict\_n,reference = bank\_test\_labels))  
 i <- i - 2  
}

## [1] 31  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 375 160  
## No 553 7149  
##   
## Accuracy : 0.9134   
## 95% CI : (0.9072, 0.9194)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 4.566e-15   
##   
## Kappa : 0.4689   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.40409   
## Specificity : 0.97811   
## Pos Pred Value : 0.70093   
## Neg Pred Value : 0.92820   
## Prevalence : 0.11266   
## Detection Rate : 0.04553   
## Detection Prevalence : 0.06495   
## Balanced Accuracy : 0.69110   
##   
## 'Positive' Class : Yes   
##   
## [1] 29  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 382 169  
## No 546 7140  
##   
## Accuracy : 0.9132   
## 95% CI : (0.9069, 0.9192)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 8.234e-15   
##   
## Kappa : 0.4723   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.41164   
## Specificity : 0.97688   
## Pos Pred Value : 0.69328   
## Neg Pred Value : 0.92896   
## Prevalence : 0.11266   
## Detection Rate : 0.04638   
## Detection Prevalence : 0.06689   
## Balanced Accuracy : 0.69426   
##   
## 'Positive' Class : Yes   
##   
## [1] 27  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 376 170  
## No 552 7139  
##   
## Accuracy : 0.9123   
## 95% CI : (0.906, 0.9184)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 6.182e-14   
##   
## Kappa : 0.4656   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.40517   
## Specificity : 0.97674   
## Pos Pred Value : 0.68864   
## Neg Pred Value : 0.92823   
## Prevalence : 0.11266   
## Detection Rate : 0.04565   
## Detection Prevalence : 0.06629   
## Balanced Accuracy : 0.69096   
##   
## 'Positive' Class : Yes   
##   
## [1] 25  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 381 175  
## No 547 7134  
##   
## Accuracy : 0.9123   
## 95% CI : (0.906, 0.9184)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 6.182e-14   
##   
## Kappa : 0.4686   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.41056   
## Specificity : 0.97606   
## Pos Pred Value : 0.68525   
## Neg Pred Value : 0.92879   
## Prevalence : 0.11266   
## Detection Rate : 0.04625   
## Detection Prevalence : 0.06750   
## Balanced Accuracy : 0.69331   
##   
## 'Positive' Class : Yes   
##   
## [1] 23  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 372 183  
## No 556 7126  
##   
## Accuracy : 0.9103   
## 95% CI : (0.9039, 0.9164)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 6.102e-12   
##   
## Kappa : 0.4558   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.40086   
## Specificity : 0.97496   
## Pos Pred Value : 0.67027   
## Neg Pred Value : 0.92762   
## Prevalence : 0.11266   
## Detection Rate : 0.04516   
## Detection Prevalence : 0.06738   
## Balanced Accuracy : 0.68791   
##   
## 'Positive' Class : Yes   
##   
## [1] 21  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 381 189  
## No 547 7120  
##   
## Accuracy : 0.9106   
## 95% CI : (0.9043, 0.9167)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 2.799e-12   
##   
## Kappa : 0.4626   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.41056   
## Specificity : 0.97414   
## Pos Pred Value : 0.66842   
## Neg Pred Value : 0.92866   
## Prevalence : 0.11266   
## Detection Rate : 0.04625   
## Detection Prevalence : 0.06920   
## Balanced Accuracy : 0.69235   
##   
## 'Positive' Class : Yes   
##   
## [1] 19  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 383 197  
## No 545 7112  
##   
## Accuracy : 0.9099   
## 95% CI : (0.9035, 0.916)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.313e-11   
##   
## Kappa : 0.4613   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.41272   
## Specificity : 0.97305   
## Pos Pred Value : 0.66034   
## Neg Pred Value : 0.92882   
## Prevalence : 0.11266   
## Detection Rate : 0.04650   
## Detection Prevalence : 0.07041   
## Balanced Accuracy : 0.69288   
##   
## 'Positive' Class : Yes   
##   
## [1] 17  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 382 190  
## No 546 7119  
##   
## Accuracy : 0.9106   
## 95% CI : (0.9043, 0.9167)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 2.799e-12   
##   
## Kappa : 0.4632   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.41164   
## Specificity : 0.97400   
## Pos Pred Value : 0.66783   
## Neg Pred Value : 0.92877   
## Prevalence : 0.11266   
## Detection Rate : 0.04638   
## Detection Prevalence : 0.06944   
## Balanced Accuracy : 0.69282   
##   
## 'Positive' Class : Yes   
##   
## [1] 15  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 393 212  
## No 535 7097  
##   
## Accuracy : 0.9093   
## 95% CI : (0.9029, 0.9154)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 4.574e-11   
##   
## Kappa : 0.4652   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.42349   
## Specificity : 0.97099   
## Pos Pred Value : 0.64959   
## Neg Pred Value : 0.92990   
## Prevalence : 0.11266   
## Detection Rate : 0.04771   
## Detection Prevalence : 0.07345   
## Balanced Accuracy : 0.69724   
##   
## 'Positive' Class : Yes   
##   
## [1] 13  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 395 215  
## No 533 7094  
##   
## Accuracy : 0.9092   
## 95% CI : (0.9028, 0.9153)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 5.845e-11   
##   
## Kappa : 0.4659   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.42565   
## Specificity : 0.97058   
## Pos Pred Value : 0.64754   
## Neg Pred Value : 0.93012   
## Prevalence : 0.11266   
## Detection Rate : 0.04795   
## Detection Prevalence : 0.07406   
## Balanced Accuracy : 0.69812   
##   
## 'Positive' Class : Yes   
##   
## [1] 11  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 390 224  
## No 538 7085  
##   
## Accuracy : 0.9075   
## 95% CI : (0.901, 0.9137)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.559e-09   
##   
## Kappa : 0.4571   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.42026   
## Specificity : 0.96935   
## Pos Pred Value : 0.63518   
## Neg Pred Value : 0.92942   
## Prevalence : 0.11266   
## Detection Rate : 0.04735   
## Detection Prevalence : 0.07454   
## Balanced Accuracy : 0.69481   
##   
## 'Positive' Class : Yes   
##   
## [1] 9  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 411 234  
## No 517 7075  
##   
## Accuracy : 0.9088   
## 95% CI : (0.9024, 0.915)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.21e-10   
##   
## Kappa : 0.474   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.44289   
## Specificity : 0.96798   
## Pos Pred Value : 0.63721   
## Neg Pred Value : 0.93190   
## Prevalence : 0.11266   
## Detection Rate : 0.04990   
## Detection Prevalence : 0.07831   
## Balanced Accuracy : 0.70544   
##   
## 'Positive' Class : Yes   
##   
## [1] 7  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 410 257  
## No 518 7052  
##   
## Accuracy : 0.9059   
## 95% CI : (0.8994, 0.9121)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 2.564e-08   
##   
## Kappa : 0.4636   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.44181   
## Specificity : 0.96484   
## Pos Pred Value : 0.61469   
## Neg Pred Value : 0.93157   
## Prevalence : 0.11266   
## Detection Rate : 0.04978   
## Detection Prevalence : 0.08098   
## Balanced Accuracy : 0.70332   
##   
## 'Positive' Class : Yes   
##   
## [1] 5  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 409 287  
## No 519 7022  
##   
## Accuracy : 0.9021   
## 95% CI : (0.8955, 0.9085)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 7.978e-06   
##   
## Kappa : 0.4506   
##   
## Mcnemar's Test P-Value : 4.064e-16   
##   
## Sensitivity : 0.44073   
## Specificity : 0.96073   
## Pos Pred Value : 0.58764   
## Neg Pred Value : 0.93118   
## Prevalence : 0.11266   
## Detection Rate : 0.04965   
## Detection Prevalence : 0.08450   
## Balanced Accuracy : 0.70073   
##   
## 'Positive' Class : Yes   
##   
## [1] 3  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 421 336  
## No 507 6973  
##   
## Accuracy : 0.8977   
## 95% CI : (0.8909, 0.9041)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.001438   
##   
## Kappa : 0.4434   
##   
## Mcnemar's Test P-Value : 4.767e-09   
##   
## Sensitivity : 0.45366   
## Specificity : 0.95403   
## Pos Pred Value : 0.55614   
## Neg Pred Value : 0.93222   
## Prevalence : 0.11266   
## Detection Rate : 0.05111   
## Detection Prevalence : 0.09190   
## Balanced Accuracy : 0.70385   
##   
## 'Positive' Class : Yes   
##   
## [1] 1  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 426 479  
## No 502 6830  
##   
## Accuracy : 0.8809   
## 95% CI : (0.8737, 0.8878)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.9681   
##   
## Kappa : 0.3978   
##   
## Mcnemar's Test P-Value : 0.4824   
##   
## Sensitivity : 0.45905   
## Specificity : 0.93446   
## Pos Pred Value : 0.47072   
## Neg Pred Value : 0.93153   
## Prevalence : 0.11266   
## Detection Rate : 0.05172   
## Detection Prevalence : 0.10987   
## Balanced Accuracy : 0.69676   
##   
## 'Positive' Class : Yes   
##

#### 2) Best among all is for K = 9

bank\_test\_predict.best <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 9)  
print(confusionMatrix(data = bank\_test\_predict.best,reference = bank\_test\_labels))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 411 235  
## No 517 7074  
##   
## Accuracy : 0.9087   
## 95% CI : (0.9023, 0.9148)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.537e-10   
##   
## Kappa : 0.4736   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.44289   
## Specificity : 0.96785   
## Pos Pred Value : 0.63622   
## Neg Pred Value : 0.93189   
## Prevalence : 0.11266   
## Detection Rate : 0.04990   
## Detection Prevalence : 0.07843   
## Balanced Accuracy : 0.70537   
##   
## 'Positive' Class : Yes   
##

### Step 3.3 - Data transformation & preparation - Only numeric predictors

Redoing the first 2 steps before transformation

source("/Users/sobil/Documents/MSC/Sem 1/Data Mining & Machine Learning/Project/Bank/bank\_import\_primaryExplore.R")

## 'data.frame': 41188 obs. of 21 variables:  
## $ age : int 56 57 37 40 56 45 59 41 24 25 ...  
## $ job : Factor w/ 12 levels "admin.","blue-collar",..: 4 8 8 1 8 8 1 2 10 8 ...  
## $ marital : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2 2 2 3 3 ...  
## $ education : Factor w/ 8 levels "basic.4y","basic.6y",..: 1 4 4 2 4 3 6 8 6 4 ...  
## $ default : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1 1 ...  
## $ housing : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3 3 ...  
## $ loan : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 3 1 1 1 1 1 ...  
## $ contact : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...  
## $ month : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7 7 7 7 ...  
## $ day\_of\_week : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ duration : int 261 149 226 151 307 198 139 217 380 50 ...  
## $ campaign : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ pdays : int 999 999 999 999 999 999 999 999 999 999 ...  
## $ previous : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ poutcome : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ emp.var.rate : num 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...  
## $ cons.price.idx: num 94 94 94 94 94 ...  
## $ cons.conf.idx : num -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...  
## $ euribor3m : num 4.86 4.86 4.86 4.86 4.86 ...  
## $ nr.employed : num 5191 5191 5191 5191 5191 ...  
## $ y : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 41188   
##   
##   
## | no | yes |   
## |-----------|-----------|  
## | 36548 | 4640 |   
## | 0.887 | 0.113 |   
## |-----------|-----------|  
##   
##   
##   
##

#### 1) Creating new data frame with only numeric predicors and normalizaing them

From previous section, it is clear that inbuilt normalization is performing better, Thus Normalization using scale inbulit R method based upon using z score

bank\_n <- as.data.frame(scale(bank[c(1,11,12,13,14,16,17,18,19,20)]))  
summary(bank\_n)

## age duration campaign pdays   
## Min. :-2.2093 Min. :-0.9962 Min. :-0.5659 Min. :-5.1494   
## 1st Qu.:-0.7700 1st Qu.:-0.6028 1st Qu.:-0.5659 1st Qu.: 0.1954   
## Median :-0.1942 Median :-0.3019 Median :-0.2049 Median : 0.1954   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.6694 3rd Qu.: 0.2342 3rd Qu.: 0.1561 3rd Qu.: 0.1954   
## Max. : 5.5632 Max. :17.9718 Max. :19.2896 Max. : 0.1954   
## previous emp.var.rate cons.price.idx cons.conf.idx   
## Min. :-0.3495 Min. :-2.2164 Min. :-2.3749 Min. :-2.2249   
## 1st Qu.:-0.3495 1st Qu.:-1.1979 1st Qu.:-0.8649 1st Qu.:-0.4748   
## Median :-0.3495 Median : 0.6481 Median : 0.2995 Median :-0.2803   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.:-0.3495 3rd Qu.: 0.8391 3rd Qu.: 0.7227 3rd Qu.: 0.8864   
## Max. :13.7948 Max. : 0.8391 Max. : 2.0581 Max. : 2.9391   
## euribor3m nr.employed   
## Min. :-1.7223 Min. :-2.8157   
## 1st Qu.:-1.3130 1st Qu.:-0.9403   
## Median : 0.7125 Median : 0.3317   
## Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.7724 3rd Qu.: 0.8452   
## Max. : 0.8208 Max. : 0.8452

#### 2) creating training and testing dataset from exisitng sample

indx <- createDataPartition(bank$y, p = 0.8, list = FALSE)  
bank\_train <- bank\_n[indx,]  
bank\_test <- bank\_n[- indx,]

creating lables for test and training data sets

bank\_train\_labels <- bank[indx,21]  
bank\_test\_labels <- bank[- indx,21]

### Step 4.3 : Training a model on the data - Only numeric predictors

using the knn method of class package with K odd

bank\_test\_predict <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 51)

### Step 5.3 : Evaluating the model - Only numeric predictors

“Model 1 : k = 51”

confusionMatrix(data = bank\_test\_predict,reference = bank\_test\_labels)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 417 211  
## No 511 7098  
##   
## Accuracy : 0.9123   
## 95% CI : (0.906, 0.9184)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 6.182e-14   
##   
## Kappa : 0.4896   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.44935   
## Specificity : 0.97113   
## Pos Pred Value : 0.66401   
## Neg Pred Value : 0.93284   
## Prevalence : 0.11266   
## Detection Rate : 0.05063   
## Detection Prevalence : 0.07624   
## Balanced Accuracy : 0.71024   
##   
## 'Positive' Class : Yes   
##

### Step 6.3 : Improving the model - Only numeric predictors

#### 1) Changing k values

Skipped model for K values which were higher than 51 (please refer the R script for model performance) - overall the performance was poor than below

Model n : k = 31 to 1 (only odd numbers)"

i <- 31  
while (i > 0) {  
 bank\_test\_predict\_n <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = i)  
 print(i)  
 print(confusionMatrix(data = bank\_test\_predict\_n,reference = bank\_test\_labels))  
 i <- i - 2  
}

## [1] 31  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 448 211  
## No 480 7098  
##   
## Accuracy : 0.9161   
## 95% CI : (0.9099, 0.922)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5196   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.48276   
## Specificity : 0.97113   
## Pos Pred Value : 0.67982   
## Neg Pred Value : 0.93666   
## Prevalence : 0.11266   
## Detection Rate : 0.05439   
## Detection Prevalence : 0.08000   
## Balanced Accuracy : 0.72695   
##   
## 'Positive' Class : Yes   
##   
## [1] 29  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 447 216  
## No 481 7093  
##   
## Accuracy : 0.9154   
## 95% CI : (0.9092, 0.9213)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5165   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.48168   
## Specificity : 0.97045   
## Pos Pred Value : 0.67421   
## Neg Pred Value : 0.93649   
## Prevalence : 0.11266   
## Detection Rate : 0.05427   
## Detection Prevalence : 0.08049   
## Balanced Accuracy : 0.72606   
##   
## 'Positive' Class : Yes   
##   
## [1] 27  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 449 216  
## No 479 7093  
##   
## Accuracy : 0.9156   
## 95% CI : (0.9094, 0.9215)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5184   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.48384   
## Specificity : 0.97045   
## Pos Pred Value : 0.67519   
## Neg Pred Value : 0.93674   
## Prevalence : 0.11266   
## Detection Rate : 0.05451   
## Detection Prevalence : 0.08073   
## Balanced Accuracy : 0.72714   
##   
## 'Positive' Class : Yes   
##   
## [1] 25  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 453 217  
## No 475 7092  
##   
## Accuracy : 0.916   
## 95% CI : (0.9098, 0.9219)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5218   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.48815   
## Specificity : 0.97031   
## Pos Pred Value : 0.67612   
## Neg Pred Value : 0.93723   
## Prevalence : 0.11266   
## Detection Rate : 0.05500   
## Detection Prevalence : 0.08134   
## Balanced Accuracy : 0.72923   
##   
## 'Positive' Class : Yes   
##   
## [1] 23  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 463 230  
## No 465 7079  
##   
## Accuracy : 0.9156   
## 95% CI : (0.9094, 0.9215)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5255   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.49892   
## Specificity : 0.96853   
## Pos Pred Value : 0.66811   
## Neg Pred Value : 0.93836   
## Prevalence : 0.11266   
## Detection Rate : 0.05621   
## Detection Prevalence : 0.08413   
## Balanced Accuracy : 0.73373   
##   
## 'Positive' Class : Yes   
##   
## [1] 21  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 464 221  
## No 464 7088  
##   
## Accuracy : 0.9168   
## 95% CI : (0.9107, 0.9227)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5304   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.50000   
## Specificity : 0.96976   
## Pos Pred Value : 0.67737   
## Neg Pred Value : 0.93856   
## Prevalence : 0.11266   
## Detection Rate : 0.05633   
## Detection Prevalence : 0.08316   
## Balanced Accuracy : 0.73488   
##   
## 'Positive' Class : Yes   
##   
## [1] 19  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 469 228  
## No 459 7081  
##   
## Accuracy : 0.9166   
## 95% CI : (0.9104, 0.9225)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.532   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.50539   
## Specificity : 0.96881   
## Pos Pred Value : 0.67288   
## Neg Pred Value : 0.93912   
## Prevalence : 0.11266   
## Detection Rate : 0.05694   
## Detection Prevalence : 0.08462   
## Balanced Accuracy : 0.73710   
##   
## 'Positive' Class : Yes   
##   
## [1] 17  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 469 241  
## No 459 7068  
##   
## Accuracy : 0.915   
## 95% CI : (0.9088, 0.921)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5264   
##   
## Mcnemar's Test P-Value : 2.368e-16   
##   
## Sensitivity : 0.50539   
## Specificity : 0.96703   
## Pos Pred Value : 0.66056   
## Neg Pred Value : 0.93902   
## Prevalence : 0.11266   
## Detection Rate : 0.05694   
## Detection Prevalence : 0.08620   
## Balanced Accuracy : 0.73621   
##   
## 'Positive' Class : Yes   
##   
## [1] 15  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 472 239  
## No 456 7070  
##   
## Accuracy : 0.9156   
## 95% CI : (0.9094, 0.9215)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.53   
##   
## Mcnemar's Test P-Value : 2.541e-16   
##   
## Sensitivity : 0.50862   
## Specificity : 0.96730   
## Pos Pred Value : 0.66385   
## Neg Pred Value : 0.93941   
## Prevalence : 0.11266   
## Detection Rate : 0.05730   
## Detection Prevalence : 0.08632   
## Balanced Accuracy : 0.73796   
##   
## 'Positive' Class : Yes   
##   
## [1] 13  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 466 256  
## No 462 7053  
##   
## Accuracy : 0.9128   
## 95% CI : (0.9065, 0.9188)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.971e-14   
##   
## Kappa : 0.5173   
##   
## Mcnemar's Test P-Value : 2.001e-14   
##   
## Sensitivity : 0.50216   
## Specificity : 0.96497   
## Pos Pred Value : 0.64543   
## Neg Pred Value : 0.93852   
## Prevalence : 0.11266   
## Detection Rate : 0.05657   
## Detection Prevalence : 0.08765   
## Balanced Accuracy : 0.73356   
##   
## 'Positive' Class : Yes   
##   
## [1] 11  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 458 270  
## No 470 7039  
##   
## Accuracy : 0.9102   
## 95% CI : (0.9038, 0.9163)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 7.889e-12   
##   
## Kappa : 0.504   
##   
## Mcnemar's Test P-Value : 2.566e-13   
##   
## Sensitivity : 0.49353   
## Specificity : 0.96306   
## Pos Pred Value : 0.62912   
## Neg Pred Value : 0.93741   
## Prevalence : 0.11266   
## Detection Rate : 0.05560   
## Detection Prevalence : 0.08838   
## Balanced Accuracy : 0.72830   
##   
## 'Positive' Class : Yes   
##   
## [1] 9  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 457 288  
## No 471 7021  
##   
## Accuracy : 0.9079   
## 95% CI : (0.9014, 0.914)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 7.895e-10   
##   
## Kappa : 0.4957   
##   
## Mcnemar's Test P-Value : 3.944e-11   
##   
## Sensitivity : 0.49246   
## Specificity : 0.96060   
## Pos Pred Value : 0.61342   
## Neg Pred Value : 0.93713   
## Prevalence : 0.11266   
## Detection Rate : 0.05548   
## Detection Prevalence : 0.09045   
## Balanced Accuracy : 0.72653   
##   
## 'Positive' Class : Yes   
##   
## [1] 7  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 461 286  
## No 467 7023  
##   
## Accuracy : 0.9086   
## 95% CI : (0.9022, 0.9147)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.950e-10   
##   
## Kappa : 0.5002   
##   
## Mcnemar's Test P-Value : 5.397e-11   
##   
## Sensitivity : 0.49677   
## Specificity : 0.96087   
## Pos Pred Value : 0.61714   
## Neg Pred Value : 0.93765   
## Prevalence : 0.11266   
## Detection Rate : 0.05597   
## Detection Prevalence : 0.09069   
## Balanced Accuracy : 0.72882   
##   
## 'Positive' Class : Yes   
##   
## [1] 5  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 478 308  
## No 450 7001  
##   
## Accuracy : 0.908   
## 95% CI : (0.9015, 0.9141)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 6.276e-10   
##   
## Kappa : 0.5068   
##   
## Mcnemar's Test P-Value : 3.034e-07   
##   
## Sensitivity : 0.51509   
## Specificity : 0.95786   
## Pos Pred Value : 0.60814   
## Neg Pred Value : 0.93961   
## Prevalence : 0.11266   
## Detection Rate : 0.05803   
## Detection Prevalence : 0.09542   
## Balanced Accuracy : 0.73647   
##   
## 'Positive' Class : Yes   
##   
## [1] 3  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 475 365  
## No 453 6944  
##   
## Accuracy : 0.9007   
## 95% CI : (0.894, 0.9071)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 5.214e-05   
##   
## Kappa : 0.4819   
##   
## Mcnemar's Test P-Value : 0.002351   
##   
## Sensitivity : 0.51185   
## Specificity : 0.95006   
## Pos Pred Value : 0.56548   
## Neg Pred Value : 0.93876   
## Prevalence : 0.11266   
## Detection Rate : 0.05767   
## Detection Prevalence : 0.10198   
## Balanced Accuracy : 0.73096   
##   
## 'Positive' Class : Yes   
##   
## [1] 1  
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 485 479  
## No 443 6830  
##   
## Accuracy : 0.8881   
## 95% CI : (0.8811, 0.8948)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 0.4257   
##   
## Kappa : 0.4495   
##   
## Mcnemar's Test P-Value : 0.2490   
##   
## Sensitivity : 0.52263   
## Specificity : 0.93446   
## Pos Pred Value : 0.50311   
## Neg Pred Value : 0.93909   
## Prevalence : 0.11266   
## Detection Rate : 0.05888   
## Detection Prevalence : 0.11703   
## Balanced Accuracy : 0.72855   
##   
## 'Positive' Class : Yes   
##

#### 2) Best among all is for K = 19

bank\_test\_predict.best <- class::knn(train = bank\_train, test = bank\_test, cl = bank\_train\_labels, k = 19)  
print(confusionMatrix(data = bank\_test\_predict.best,reference = bank\_test\_labels))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 469 227  
## No 459 7082  
##   
## Accuracy : 0.9167   
## 95% CI : (0.9105, 0.9226)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5324   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.50539   
## Specificity : 0.96894   
## Pos Pred Value : 0.67385   
## Neg Pred Value : 0.93913   
## Prevalence : 0.11266   
## Detection Rate : 0.05694   
## Detection Prevalence : 0.08450   
## Balanced Accuracy : 0.73717   
##   
## 'Positive' Class : Yes   
##

confusionMatrix (data = bank\_test\_predict.best,reference = bank\_test\_labels, mode="prec\_recall")

##Confusion Matrix and Statistics

##

## Reference

##Prediction Yes No

## Yes 469 226

## No 459 7083

##

## Accuracy : 0.9168

## 95% CI : (0.9107, 0.9227)

## No Information Rate : 0.8873

## P-Value [Acc > NIR] : < 2.2e-16

##

## Kappa : 0.5329

##

## Mcnemar's Test P-Value : < 2.2e-16

##

## Precision : 0.67482

## Recall : 0.50539

## F1 : 0.57794

## Prevalence : 0.11266

## Detection Rate : 0.05694

## Detection Prevalence : 0.08438

## Balanced Accuracy : 0.73723

##

## 'Positive' Class : Yes

##