# DATA MODELLING ----  
  
set.seed(128)  
  
# libraries  
library(dplyr)

library(caret)

library(gmodels)  
library(cluster)  
library(pROC)

library(ROCR)  
  
# load dataset  
data <- read.csv("Data/data\_preprocessed.csv")  
  
# Factoring target variable  
data$success <- factor(data$success, levels = c("No", "Yes"))  
table(data$success)

# splitting data  
training\_indices <- createDataPartition(data$success, p = 0.8,   
 list = FALSE)  
train\_data <- data[training\_indices,]  
test\_data <- data[-training\_indices,]  
  
train\_data <- train\_data[,1:11]  
test\_data <- test\_data[,1:11]  
summary(train\_data[,1:11])

# Measures ----  
accuracy <- function(actual, predicted) {  
 correct <- sum(actual == predicted)  
 total <- length(actual)  
 accuracy <- correct / total  
 return(accuracy)  
}  
  
precision <- function(actual, predicted, positive\_label) {  
 true\_positive <- sum(actual == positive\_label & predicted == positive\_label)  
 false\_positive <- sum(actual != positive\_label & predicted == positive\_label)  
 precision <- true\_positive / (true\_positive + false\_positive)  
 return(precision)  
}  
  
# Controls  
ctrl\_1 <- trainControl(method = "cv", number = 10)  
ctrl\_2 <- trainControl(method = "CV", number = 30)  
ctrl\_3 <- trainControl(method = "CV", number = 50)  
  
# KNN ----  
knn\_model <- train(success ~ .,  
 data = train\_data,  
 method = "knn",  
 trControl = ctrl\_3,  
 preProcess = c("center", "scale"))  
  
knn\_predict <- predict(knn\_model, newdata = test\_data)  
  
CrossTable(test\_data$success, knn\_predict, dnn=c('actual', 'predict'),  
 prop.chisq = FALSE, prop.t = FALSE, prop.r = FALSE)

knn\_accuracy <- accuracy(test\_data$success, knn\_predict)  
knn\_precision <- precision(test\_data$success, knn\_predict, "Yes")  
knn\_sensitivity <- sensitivity(knn\_predict, test\_data$success, positive = "Yes")  
knn\_specificity <- specificity(knn\_predict, test\_data$success, negative = "No")  
  
  
# Naive Bayes ----  
nb\_model <- train(success ~., data = train\_data, method = "naive\_bayes",  
 trControl = ctrl\_3)  
  
nb\_predict <- predict(nb\_model, newdata = test\_data)  
  
CrossTable(nb\_predict, test\_data$success, dnn=c('predict', 'actual'),  
 prop.chisq = FALSE, prop.t = FALSE, prop.r = FALSE)

nb\_accuracy <- accuracy(test\_data$success, nb\_predict)  
nb\_precision <- precision(test\_data$success, nb\_predict, "Yes")  
nb\_sensitivity <- sensitivity(nb\_predict, test\_data$success, positive = "Yes")  
nb\_specificity <- specificity(nb\_predict, test\_data$success, negative = "No")  
  
# Decision Trees ----  
dt\_model <- train(success ~., data = train\_data, method = "rpart",  
 trControl = ctrl\_3)  
  
dt\_predict <- predict(dt\_model, newdata = test\_data)  
  
CrossTable(dt\_predict, test\_data$success, dnn=c('predict', 'actual'),  
 prop.chisq = FALSE, prop.t = FALSE, prop.r = FALSE)

dt\_accuracy <- accuracy(test\_data$success, dt\_predict)  
dt\_precision <- precision(test\_data$success, dt\_predict, "Yes")  
dt\_sensitivity <- sensitivity(dt\_predict, test\_data$success, positive = "Yes")  
dt\_specificity <- specificity(dt\_predict, test\_data$success, negative = "No")  
  
# SVM ----  
svm\_model <- train(success ~., data = train\_data, method = "svmRadial",  
 trainControl = ctrl\_3)  
  
svm\_predict <- predict(svm\_model, newdata = test\_data)  
  
CrossTable(svm\_predict, test\_data$success, dnn=c('predict', 'actual'),  
 prop.chisq = FALSE, prop.t = FALSE, prop.r = FALSE)

svm\_accuracy <- accuracy(test\_data$success, svm\_predict)  
svm\_precision <- precision(test\_data$success, svm\_predict, "Yes")  
svm\_sensitivity <- sensitivity(svm\_predict, test\_data$success, positive = "Yes")  
svm\_specificity <- specificity(svm\_predict, test\_data$success, negative = "No")  
  
# Logistic Regression ----  
# training  
lr\_model <- train(success ~ ., data = train\_data, method = "glm",   
 family = "binomial", trControl = ctrl\_3)  
  
# obtain predicted values  
lr\_predict <- predict(lr\_model, test\_data)  
  
  
CrossTable(lr\_predict, test\_data$success, dnn=c('predict', 'actual'),  
 prop.chisq = FALSE, prop.t = FALSE, prop.r = FALSE)

lr\_accuracy <- accuracy(test\_data$success, lr\_predict)  
lr\_precision <- precision(test\_data$success, lr\_predict, "Yes")  
lr\_sensitivity <- sensitivity(lr\_predict, test\_data$success, positive = "Yes")  
lr\_specificity <- specificity(lr\_predict, test\_data$success, negative = "No")