KNN MODEL zoo data set

##Data set zoo constructing knn classifier##

zoo <- read.csv(file.choose())

View(zoo)

str(zoo)##to known the structure of the variable##

##First column is not required for prediction in the model so we will remove the first column##

zoo <- zoo[-1]

View(zoo)

## Now we see the diagonis by table##

table(zoo$type)

str(zoo)

## Here we should convert the y dependent variable to categorical variable##

zoo$type <- as.factor(zoo$type)

View(zoo)

str(zoo)

summary(zoo)

#Create a function to normalize the data

norm <- function(x){

return((x-min(x))/(max(x)-min(x)))

}

zoo\_n<- as.data.frame(lapply(zoo[1:16], norm))

View(zoo\_n)

##splitting the data into the test case and train##

set.seed(123)

ind <- sample(2,nrow(zoo\_n), replace = T, prob = c(0.7,0.3))

train <- zoo\_n[ind==1,]

test <- zoo\_n[ind==2,]

zoo\_train <- zoo\_n[1:72,]

zoo\_test <- zoo\_n[72:101,]

#Get labels for training and test datasets

zoo\_train\_label <- zoo[1:72,17]

zoo\_test\_label <- zoo[72:101,17]

# Building the KNN model on training dataset and also need labels which we are including c1. Once we build the preduction model

# we have to test on test dataset

test\_acc <- NULL

train\_acc <- NULL

library(class)

## writing a function for checking different k values

for (i in seq(2,50,1))

{

zoo\_train\_pred <- knn(train=zoo\_train,test=zoo\_train,cl=zoo\_train\_label,k=i)

train\_acc <- c(train\_acc,mean(zoo\_train\_pred==zoo\_train\_label))

zoo\_test\_pred <- knn(train = zoo\_test, test = zoo\_test, cl = zoo\_test\_label, k=i)

test\_acc <- c(test\_acc,mean(zoo\_test\_pred==zoo\_test\_label))

}

train\_acc

# Testing Accuracy

# Plotting 2 different graphs on same window

par(mfrow=c(1,2)) # c(1,2) => indicates 1 row and 2 columns

plot(seq(2,50,1),train\_acc,type="l",main="Train\_accuracy",col="blue")

plot(seq(2,50,1),test\_acc,type="l",main="Test\_accuracy",col="red")

zoo\_pred <- knn(train = zoo\_train, test = zoo\_test, cl = zoo\_train\_label, k=3)

zoo\_pred

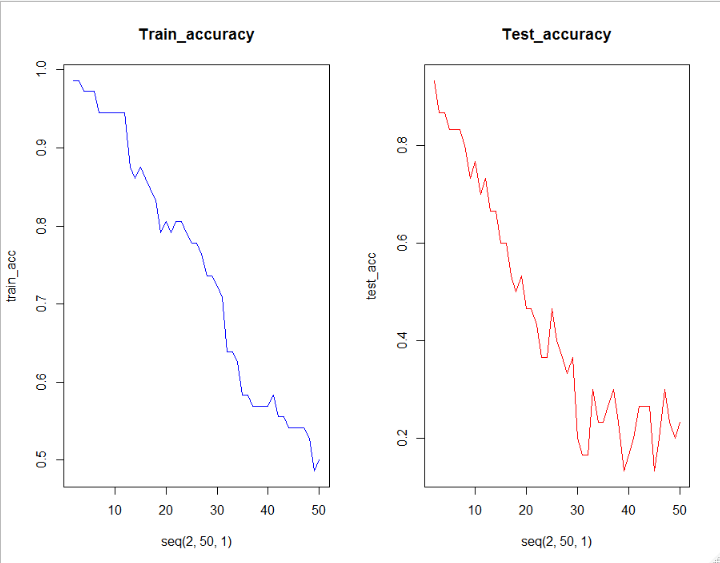
## Now evualuation the model performance

install.packages("gmodels")

library(gmodels)

# Create cross table of predicted and actual

CrossTable( x = zoo\_test\_label, y = zoo\_pred)



##accuracy=23/30 0.76%

Glass Data set

##Prepare a model for glass classification using KNN

ibrary(caTools)

library(dplyr)

library(ggplot2)

library(caret)

library(class)

library(corrplot)

glass <- read.csv(file.choose())

##EDA##

View(glass)

str(glass)

summary(glass)

## we should covert the type data in factors##

glass$Type=as.factor(glass$Type)

View(glass)

str(glass)

## sandardize the data##

standard.features <- scale(glass[,1:9])

#Join the standardized data with the target column

data <- cbind(standard.features,glass[10])

#Check if there are any missing values to impute.

anyNA(data)

## There are no null values##

# Data partition

set.seed(123)

ind <- sample(2,nrow(data), replace = T, prob = c(0.7,0.3))

train <- data[ind==1,]

test <- data[ind==2,]

train\_label <- train[,10]

test\_label <- test[,10]

# KNN Model

trcontrol <- trainControl(method = "repeatedcv", number = 10,repeats = 2

# classprobs are needed when u want to select ROC for optimal K Value

)

set.seed(222)

fit <- train(Type ~., data = train, method = 'knn', tuneLength = 20,

trControl = trcontrol, preProc = c("center","scale"))

plot(fit)

fit

pred <- predict(fit, newdata = test )

confusionMatrix(pred, test$Type)

##knn model

test\_acc <- NULL

train\_acc <- NULL

for (i in seq(2,50,1))

{

glass\_train\_pred <- knn(train=train,test=train,cl=train\_label,k=i)

train\_acc <- c(train\_acc,mean(glass\_train\_pred==train\_label))

glass\_test\_pred <- knn(train = test, test = test, cl = test\_label, k=i)

test\_acc <- c(test\_acc,mean(glass\_test\_pred==test\_label))

}

train\_acc

# Testing Accuracy

# Plotting 2 different graphs on same window

par(mfrow=c(1,2)) # c(1,2) => indicates 1 row and 2 columns

plot(seq(2,50,1),train\_acc,type="l",main="Train\_accuracy",col="blue")

plot(seq(2,50,1),test\_acc,type="l",main="Test\_accuracy",col="red")

glass\_pred <- knn(train = train, test = test, cl = train\_label, k=3)

glass\_pred

## Now evualuation the model performance

install.packages("gmodels")

library(gmodels)

# Create cross table of predicted and actual

CrossTable( x = test\_label, y = glass\_pred)

##Accurary 52/57=91.22

