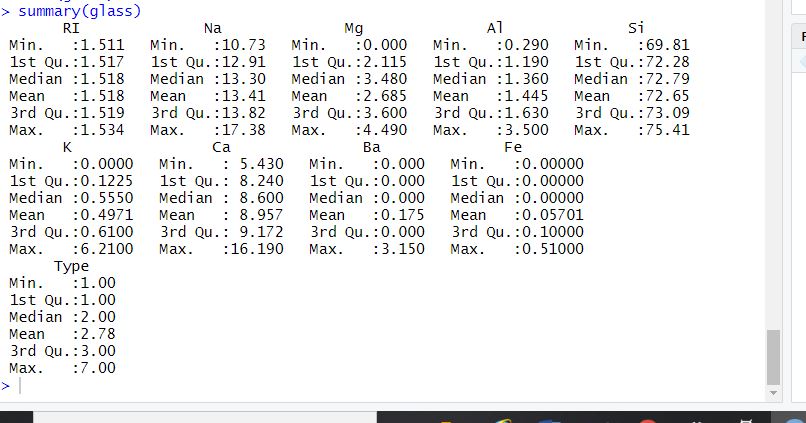
K Nearest Neighbors Documentation

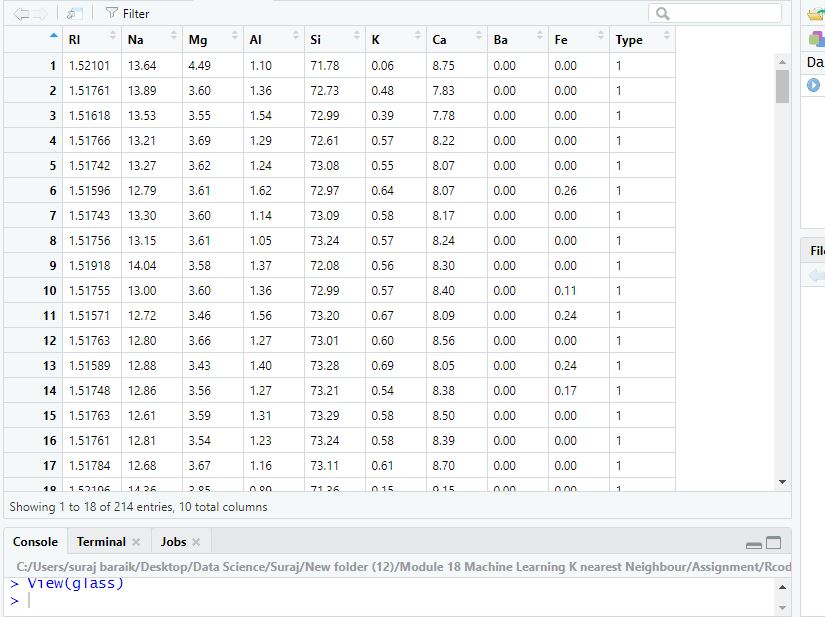
Business Problem:

Prepare a model for glass classification using KNN

About the Data

Glass Identification Database contains 10 attributes including id. The response is glass type which has 7 discrete values.





Install and load the following required libraries:

install.packages("caret") #Confusion Matrix

library(caret)

install.packages("Rtools")

library(Rtools)

install.packages("lattice") # high-level data visualization

library(lattice)

install.packages("ggplot2") #for Data Visualization

library(ggplot2)

install.packages("e1071")

library("e1071")

install.packages("class") #KNN

library("class")

Normalize the Data

#Create a function to normalize the data

norm <- function(x){

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

}

#test normalization

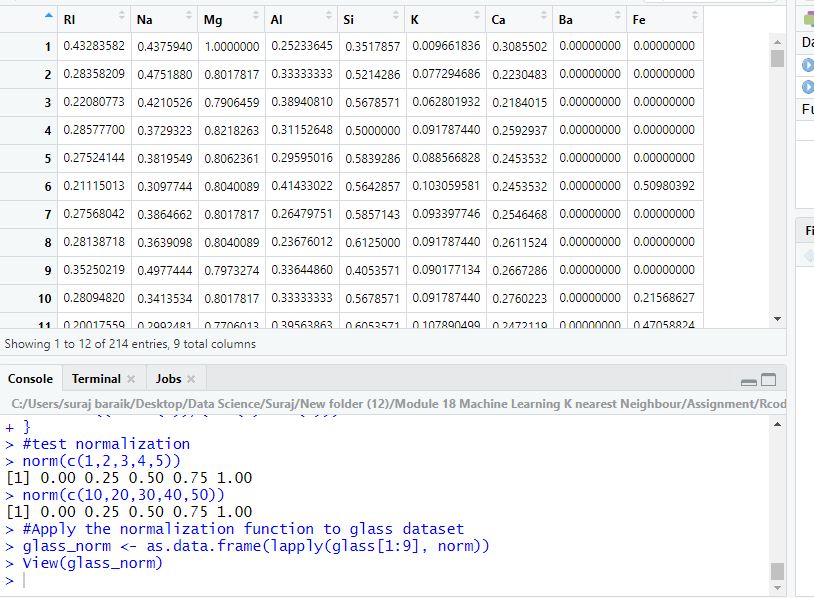
norm(c(1,2,3,4,5))

norm(c(10,20,30,40,50))

#Apply the normalization function to glass dataset

glass\_norm <- as.data.frame(lapply(glass[1:9], norm))

View(glass\_norm)



anyNA(glass\_norm)

#there is no NA values

#create training and test datasets using sequential splitting

glass\_train <- glass\_norm[1:150,]

glass\_test <- glass\_norm[151:214,]

#Get labels for training and test datasets

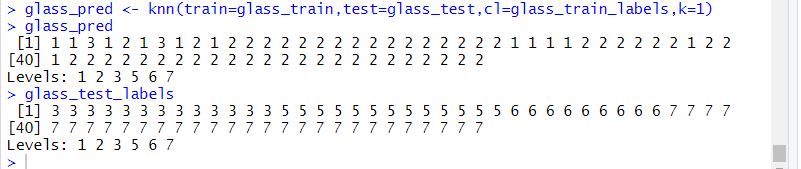
glass\_train\_labels <- glass[1:150,10]

glass\_test\_labels <- glass[151:214,10]

# Build a KNN model on taining dataset

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

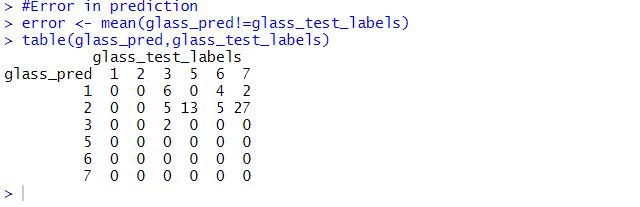
# we have to test on test dataset



#Error in prediction

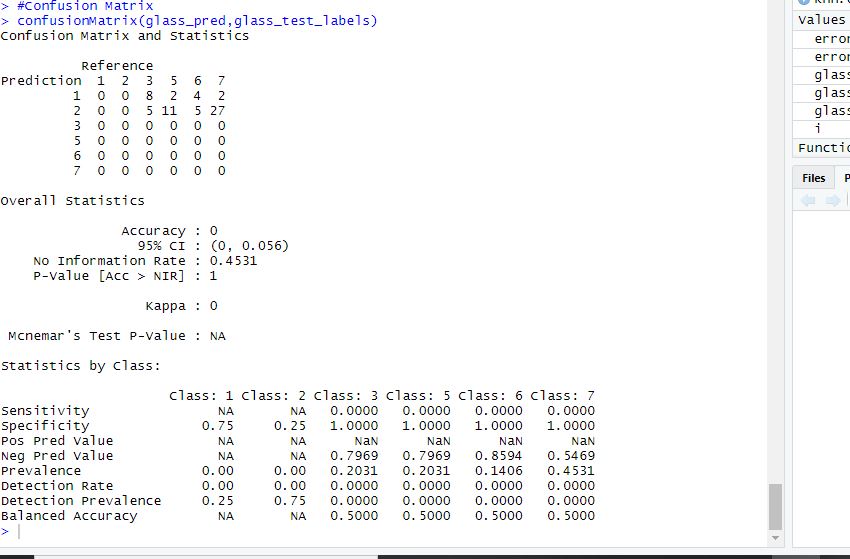
error <- mean(glass\_pred!=glass\_test\_labels)

table(glass\_pred,glass\_test\_labels)



#Confusion Matrix

confusionMatrix(glass\_pred,glass\_test\_labels)

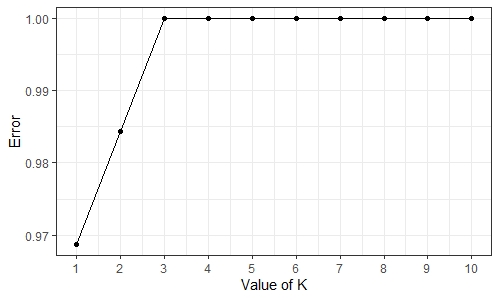


#By summary of confusion matrix for k=1 we are getting very low accuracy that 0.03

#so we need to check for optimum k value

# iterations of for k values

#plotting k value vs error choose k value



#Confusion Matrix

confusionMatrix(glass\_pred,glass\_test\_labels)

#from summary we are getting accuracy as 0 for second optimum value so we can conclude from this that sampling of data is not good

#we did sequention splitting in which there are only 1,2,3 type are there and not others

#so my model is not able to relate further types of test with which in train data

# we need to go for random sampling to get results

confusionMatrix(glass\_pred\_rnd,test$Type)

#from summary accuracy has improved to 0.70 it means this sampling is relevant for prediction

# now we have to check for accuracy with optimum value k

# iterations of for k values

glass\_pred\_rnd <- NULL

error.rate <- NULL

for (i in 1:10) {

glass\_pred\_rnd <- knn(train[1:9],test[1:9],train$Type,k=i)

error.rate[i] <- mean(glass\_pred\_rnd!=test$Type)

}

knn.error1 <- as.data.frame(cbind(k=1:10,error.type =error.rate))

ggplot(knn.error1,aes(k,error.type))+

geom\_point()+

geom\_line() +

scale\_x\_continuous(breaks=1:10)+

theme\_bw() +

xlab("Value of K") +

ylab('Error')

#From plot we can say that k=1 is optimum value but we will check with k=3

#KNN model buildingg with random sampling

glass\_pred\_rnd <- knn(train[1:9],test[1:9],train$Type,k=3)

#Error in prediction

error <- mean(glass\_pred\_rnd!=test$Type)

#Confusion Matrix

confusionMatrix(glass\_pred\_rnd,test$Type)

# From summary accuracy is dipping to 0.6 so it good to go with k=1 with accuracy 0.7