**Prepare a model for glass classification using KNN**

Data Description:

RI : refractive index

Na: Sodium (unit measurement: weight percent in corresponding oxide, as are attributes 4-10)

Mg: Magnesium

AI: Aluminum

Si: Silicon

K:Potassium

Ca: Calcium

Ba: Barium

Fe: Iron

Type: Type of glass: (class attribute)

1 -- building\_windows\_float\_processed

2 --building\_windows\_non\_float\_processed

3 --vehicle\_windows\_float\_processed

4 --vehicle\_windows\_non\_float\_processed (none in this database)

5 --containers

6 --tableware

7 --headlamps

**Ans:**

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

> head(glass)

RI Na Mg Al Si K Ca Ba Fe Type

1 1.52101 13.64 4.49 1.10 71.78 0.06 8.75 0 0.00 1

2 1.51761 13.89 3.60 1.36 72.73 0.48 7.83 0 0.00 1

3 1.51618 13.53 3.55 1.54 72.99 0.39 7.78 0 0.00 1

4 1.51766 13.21 3.69 1.29 72.61 0.57 8.22 0 0.00 1

5 1.51742 13.27 3.62 1.24 73.08 0.55 8.07 0 0.00 1

6 1.51596 12.79 3.61 1.62 72.97 0.64 8.07 0 0.26 1

> View(glass)

> table(glass$Type)

1 2 3 5 6 7

70 76 17 13 9 29

**Structure of variable ‘type’ is int, need to convert it to factor**

> str(glass)

'data.frame': 214 obs. of 10 variables:

$ RI : num 1.52 1.52 1.52 1.52 1.52 ...

$ Na : num 13.6 13.9 13.5 13.2 13.3 ...

$ Mg : num 4.49 3.6 3.55 3.69 3.62 3.61 3.6 3.61 3.58 3.6 ...

$ Al : num 1.1 1.36 1.54 1.29 1.24 1.62 1.14 1.05 1.37 1.36 ...

$ Si : num 71.8 72.7 73 72.6 73.1 ...

$ K : num 0.06 0.48 0.39 0.57 0.55 0.64 0.58 0.57 0.56 0.57 ...

$ Ca : num 8.75 7.83 7.78 8.22 8.07 8.07 8.17 8.24 8.3 8.4 ...

$ Ba : num 0 0 0 0 0 0 0 0 0 0 ...

$ Fe : num 0 0 0 0 0 0.26 0 0 0 0.11 ...

$ Type: int 1 1 1 1 1 1 1 1 1 1 ...

> glass$Type <- factor(glass$Type)

> str(glass)

'data.frame': 214 obs. of 10 variables:

$ RI : num 1.52 1.52 1.52 1.52 1.52 ...

$ Na : num 13.6 13.9 13.5 13.2 13.3 ...

$ Mg : num 4.49 3.6 3.55 3.69 3.62 3.61 3.6 3.61 3.58 3.6 ...

$ Al : num 1.1 1.36 1.54 1.29 1.24 1.62 1.14 1.05 1.37 1.36 ...

$ Si : num 71.8 72.7 73 72.6 73.1 ...

$ K : num 0.06 0.48 0.39 0.57 0.55 0.64 0.58 0.57 0.56 0.57 ...

$ Ca : num 8.75 7.83 7.78 8.22 8.07 8.07 8.17 8.24 8.3 8.4 ...

$ Ba : num 0 0 0 0 0 0 0 0 0 0 ...

$ Fe : num 0 0 0 0 0 0.26 0 0 0 0.11 ...

$ Type: Factor w/ 6 levels "1","2","3","5",..: 1 1 1 1 1 1 1 1 1 1 ...

**Table or proportation of enteries in the datasets to check Biasness.**

> round(prop.table(table(glass$Type))\*100,1)

1 2 3 5 6 7

32.7 35.5 7.9 6.1 4.2 13.6

**‘Type’ Variable is not highly biased towards specific glass type as observed above.**

> summary(glass)

RI Na Mg Al Si

Min. :1.511 Min. :10.73 Min. :0.000 Min. :0.290 Min. :69.81

1st Qu.:1.517 1st Qu.:12.91 1st Qu.:2.115 1st Qu.:1.190 1st Qu.:72.28

Median :1.518 Median :13.30 Median :3.480 Median :1.360 Median :72.79

Mean :1.518 Mean :13.41 Mean :2.685 Mean :1.445 Mean :72.65

3rd Qu.:1.519 3rd Qu.:13.82 3rd Qu.:3.600 3rd Qu.:1.630 3rd Qu.:73.09

Max. :1.534 Max. :17.38 Max. :4.490 Max. :3.500 Max. :75.41

K Ca Ba Fe Type

Min. :0.0000 Min. : 5.430 Min. :0.000 Min. :0.00000 1:70

1st Qu.:0.1225 1st Qu.: 8.240 1st Qu.:0.000 1st Qu.:0.00000 2:76

Median :0.5550 Median : 8.600 Median :0.000 Median :0.00000 3:17

Mean :0.4971 Mean : 8.957 Mean :0.175 Mean :0.05701 5:13

3rd Qu.:0.6100 3rd Qu.: 9.172 3rd Qu.:0.000 3rd Qu.:0.10000 6: 9

Max. :6.2100 Max. :16.190 Max. :3.150 Max. :0.51000 7:29

**Creating a function to normalize the data**

>norm <- function(x){

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

}

**Applying normalization function to the dataset.**

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

> head(glass\_norm)

RI Na Mg Al Si K Ca Ba Fe

1 0.4328358 0.4375940 1.0000000 0.2523364 0.3517857 0.009661836 0.3085502 0 0.0000000

2 0.2835821 0.4751880 0.8017817 0.3333333 0.5214286 0.077294686 0.2230483 0 0.0000000

3 0.2208077 0.4210526 0.7906459 0.3894081 0.5678571 0.062801932 0.2184015 0 0.0000000

4 0.2857770 0.3729323 0.8218263 0.3115265 0.5000000 0.091787440 0.2592937 0 0.0000000

5 0.2752414 0.3819549 0.8062361 0.2959502 0.5839286 0.088566828 0.2453532 0 0.0000000

6 0.2111501 0.3097744 0.8040089 0.4143302 0.5642857 0.103059581 0.2453532 0 0.5098039

**Creating 70:30 train and test datasets**

> 214\*0.7

[1] 149.8

So 150 train observations out of total 214 observations

> train\_glass <- glass\_norm[1:150,]

> test\_glass <- glass\_norm[150:214 ,]

**Get labels for training and test datasets**

> train\_labels <- glass[1:150,1]

> test\_labels <- glass[150:214,1]

**Build a KNN model on training dataset**

>library("class")

**Building the KNN model on training dataset and then test on test dataset.**

test\_acc <- NULL

train\_acc <- NULL

for (i in seq(1,214,2))

{

train\_glass\_pred <- knn(train=train\_glass,test=test\_glass,cl=train\_labels,k=i)

train\_acc <- c(train\_acc,mean(train\_glass\_pred==train\_labels))

test\_glass\_pred <- knn(train = train\_glass, test = test\_glass, cl = train\_labels, k=i)

test\_acc <- c(test\_acc,mean(test\_glass\_pred==test\_labels))

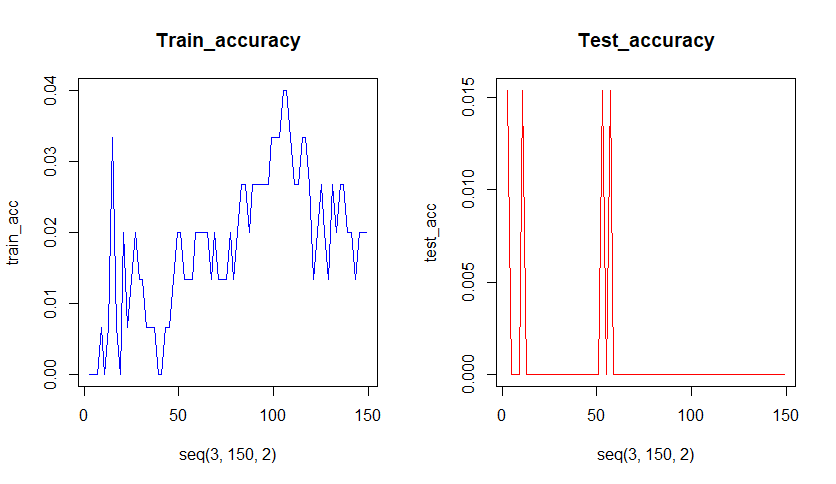
}

**Testing Accuracy**

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

>plot(seq(3,150,2),train\_acc,type="l",main="Train\_accuracy",col="blue")

>plot(seq(3,150,2),test\_acc,type="l",main="Test\_accuracy",col="red")



**For K= 10, Accuracy is good.**

> glass\_pred <- knn(train = train\_glass, test = test\_glass, cl = train\_labels, k=10)

**So we will Deploy the model with k=10 on train and test data.**