**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**

**#importing data set**

glass<-read.csv(file.choose(),header = T)

View(glass)

**#display top 10 records**

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

7 1.51743 13.30 3.60 1.14 73.09 0.58 8.17 0 0.00 1

8 1.51756 13.15 3.61 1.05 73.24 0.57 8.24 0 0.00 1

9 1.51918 14.04 3.58 1.37 72.08 0.56 8.30 0 0.00 1

10 1.51755 13.00 3.60 1.36 72.99 0.57 8.40 0 0.11 1

**#display the column name**

colnames(glass)

[1] "RI" "Na" "Mg" "Al" "Si" "K" "Ca" "Ba" "Fe" "Type"

**#shows the structure of the data set**

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 ...

**#check whether any null values present in the data se**t

sum(is.na(glass))

0

**#here we have to Prepare a model for glass classification using KNN**

**# so 'Type' is our target or label or dependent variable**

**#using available information we have to classify whether new glass lies which type**

#**table of our target variable 'Type**

table(glass$Type)

1 2 3 5 6 7

70 76 17 13 9 29

**#here we can see that all variable are in numerical format**

**#convert all those variable into text format for better understanding**

**# recode 'Type' as a factor**

**#following are the type of glasses**

**#1 -- building\_windows\_float\_processed <- T1**

**#2 --building\_windows\_non\_float\_processed <- T2**

**#3 --vehicle\_windows\_float\_processed <- T3**

**#4 --vehicle\_windows\_non\_float\_processed <-T4**

**#5 --containers <-T5**

**#6 --tableware <- T6**

**#7 --headlamps <- T7**

glass$Type <- factor(glass$Type, levels = c(1,2,3,4,5,6,7),

labels = c("T1", "T2","T3","T4","T5","T6","T7"))

table(glass$Type)

T1 T2 T3 T4 T5 T6 T7

70 76 17 0 13 9 29

View(glass)

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/ 7 levels "T1","T2","T3",..: 1 1 1 1 1 1 1 1 1 1 ...

**# use the scale() function to z-score standardize a data frame**

glass\_z <- as.data.frame(scale(glass[-10]))

View(glass\_z)

head(glass\_z)

RI Na Mg Al Si K Ca Ba

1 0.8708258 0.2842867 1.2517037 -0.6908222 -1.12444556 -0.67013422 -0.1454254 -0.3520514

2 -0.2487502 0.5904328 0.6346799 -0.1700615 0.10207972 -0.02615193 -0.7918771 -0.3520514

3 -0.7196308 0.1495824 0.6000157 0.1904651 0.43776033 -0.16414813 -0.8270103 -0.3520514

4 -0.2322859 -0.2422846 0.6970756 -0.3102663 -0.05284979 0.11184428 -0.5178378 -0.3520514

5 -0.3113148 -0.1688095 0.6485456 -0.4104126 0.55395746 0.08117845 -0.6232375 -0.3520514

6 -0.7920739 -0.7566101 0.6416128 0.3506992 0.41193874 0.21917466 -0.6232375 -0.3520514

Fe

1 -0.5850791

2 -0.5850791

3 -0.5850791

4 -0.5850791

5 -0.5850791

6 2.0832652

#**SPLITTING THE DATA SET**

set.seed(123)

datas<-sample(1:nrow(glass\_z),size=nrow(glass\_z)\*0.7,replace = FALSE)

**#Creating training and testing data set**

train\_glass<-glass\_z[datas,]

test\_glass<-glass\_z[-datas,]

head(train\_glass)

RI Na Mg Al Si K Ca Ba

159 -0.19935717 0.1495824 0.5029557 0.1504066 -0.7887650 0.12717719 -0.1173188 -0.3520514

207 -0.63072325 1.8762465 -1.8611468 0.8514306 0.5926898 -0.76213169 -0.2016386 2.4233841

179 -0.02483502 1.2884459 -0.3081879 0.3506992 -0.3497980 -0.76213169 0.2129337 -0.3520514

14 -0.29155755 -0.6708892 0.6069485 -0.3503248 0.7217978 0.06584554 -0.4054114 -0.3520514

195 -0.50559416 1.4109044 -1.8611468 1.0717524 0.8250841 -0.76213169 -0.3070383 2.8055093

170 0.51848867 -0.1688095 -1.8611468 0.6311088 0.4894035 -0.04148484 1.6604234 -0.3520514

Fe

159 -0.5850791

207 -0.5850791

179 -0.5850791

14 1.1596076

195 0.1333213

170 -0.5850791

head(test\_glass)

RI Na Mg Al Si K Ca Ba

2 -0.2487502 0.5904328 0.6346799 -0.1700615 0.1020797 -0.02615193 -0.7918771 -0.3520514

3 -0.7196308 0.1495824 0.6000157 0.1904651 0.4377603 -0.16414813 -0.8270103 -0.3520514

10 -0.2685075 -0.4994473 0.6346799 -0.1700615 0.4377603 0.11184428 -0.3913581 -0.3520514

11 -0.8743957 -0.8423310 0.5376200 0.2305236 0.7088870 0.26517339 -0.6091842 -0.3520514

12 -0.2421645 -0.7443642 0.6762770 -0.3503248 0.4635819 0.15784301 -0.2789317 -0.3520514

15 -0.2421645 -0.9770353 0.6277471 -0.2702078 0.8250841 0.12717719 -0.3210916 -0.3520514

Fe

2 -0.5850791

3 -0.5850791

10 0.5438358

11 1.8780079

12 -0.5850791

15 -0.5850791

**#creating training and testing labels**

train\_glass\_label<-glass[datas,10]

test\_glass\_label<-glass[-datas,10]

head(train\_glass\_label)

[1] T3 T7 T6 T1 T7 T5

Levels: T1 T2 T3 T4 T5 T6 T7

head(test\_glass\_label)

[1] T1 T1 T1 T1 T1 T1

Levels: T1 T2 T3 T4 T5 T6 T7

**#---- Training a model on the data ----**

**# load the "class" library**

install.packages("class")

library(class)

knn\_glass\_pred1 <- knn(train = train\_glass, test = test\_glass,

cl = train\_glass\_label, k=1)

knn\_glass\_pred1

[1] T2 T2 T3 T1 T1 T1 T3 T1 T1 T1 T1 T1 T3 T2 T2 T1 T1 T1 T1 T2 T1 T1 T2 T2 T3 T2 T2 T1 T2 T2

[31] T2 T2 T2 T2 T2 T2 T2 T2 T5 T2 T2 T2 T2 T2 T1 T1 T2 T3 T2 T3 T2 T5 T5 T6 T1 T6 T6 T2 T5 T1

[61] T2 T7 T7 T7 T7

Levels: T1 T2 T3 T4 T5 T6 T7

**#calculate the proportion of correct classification for k=1**

accur<-100\*sum(test\_glass\_label==knn\_glass\_pred1)/NROW(test\_glass\_label)

accur

67.69231

table(knn\_glass\_pred2,test\_glass\_label)

test\_glass\_label

knn\_glass\_pred2 T1 T2 T3 T4 T5 T6 T7

T1 15 5 5 0 0 3 1

T2 7 16 2 0 1 0 3

T3 0 0 0 0 0 0 0

T4 0 0 0 0 0 0 0

T5 0 1 0 0 1 0 0

T6 0 0 0 0 0 1 0

T7 0 0 0 0 0 0 4

**##--------Evaluating model performance ----**

**# load the "gmodels" library**

library(gmodels)

CrossTable(x = test\_glass\_label, y = knn\_glass\_pred2)

Cell Contents

|-------------------------|

| N |

| Chi-square contribution |

| N / Row Total |

| N / Col Total |

| N / Table Total |

|-------------------------|

Total Observations in Table: 65

| knn\_glass\_pred2

test\_glass\_label | T1 | T2 | T5 | T6 | T7 | Row Total |

-----------------|-----------|-----------|-----------|-----------|-----------|-----------|

T1 | 15 | 7 | 0 | 0 | 0 | 22 |

| 2.739 | 0.808 | 0.677 | 0.338 | 1.354 | |

| 0.682 | 0.318 | 0.000 | 0.000 | 0.000 | 0.338 |

| 0.517 | 0.241 | 0.000 | 0.000 | 0.000 | |

| 0.231 | 0.108 | 0.000 | 0.000 | 0.000 | |

-----------------|-----------|-----------|-----------|-----------|-----------|-----------|

T2 | 5 | 16 | 1 | 0 | 0 | 22 |

| 2.362 | 3.897 | 0.154 | 0.338 | 1.354 | |

| 0.227 | 0.727 | 0.045 | 0.000 | 0.000 | 0.338 |

| 0.172 | 0.552 | 0.500 | 0.000 | 0.000 | |

| 0.077 | 0.246 | 0.015 | 0.000 | 0.000 | |

-----------------|-----------|-----------|-----------|-----------|-----------|-----------|

T3 | 5 | 2 | 0 | 0 | 0 | 7 |

| 1.128 | 0.404 | 0.215 | 0.108 | 0.431 | |

| 0.714 | 0.286 | 0.000 | 0.000 | 0.000 | 0.108 |

| 0.172 | 0.069 | 0.000 | 0.000 | 0.000 | |

| 0.077 | 0.031 | 0.000 | 0.000 | 0.000 | |

-----------------|-----------|-----------|-----------|-----------|-----------|-----------|

T5 | 0 | 1 | 1 | 0 | 0 | 2 |

| 0.892 | 0.013 | 14.312 | 0.031 | 0.123 | |

| 0.000 | 0.500 | 0.500 | 0.000 | 0.000 | 0.031 |

| 0.000 | 0.034 | 0.500 | 0.000 | 0.000 | |

| 0.000 | 0.015 | 0.015 | 0.000 | 0.000 | |

-----------------|-----------|-----------|-----------|-----------|-----------|-----------|

T6 | 3 | 0 | 0 | 1 | 0 | 4 |

| 0.828 | 1.785 | 0.123 | 14.312 | 0.246 | |

| 0.750 | 0.000 | 0.000 | 0.250 | 0.000 | 0.062 |

| 0.103 | 0.000 | 0.000 | 1.000 | 0.000 | |

| 0.046 | 0.000 | 0.000 | 0.015 | 0.000 | |

-----------------|-----------|-----------|-----------|-----------|-----------|-----------|

T7 | 1 | 3 | 0 | 0 | 4 | 8 |

| 1.849 | 0.091 | 0.246 | 0.123 | 24.992 | |

| 0.125 | 0.375 | 0.000 | 0.000 | 0.500 | 0.123 |

| 0.034 | 0.103 | 0.000 | 0.000 | 1.000 | |

| 0.015 | 0.046 | 0.000 | 0.000 | 0.062 | |

-----------------|-----------|-----------|-----------|-----------|-----------|-----------|

Column Total | 29 | 29 | 2 | 1 | 4 | 65 |

| 0.446 | 0.446 | 0.031 | 0.015 | 0.062 | |

-----------------|-----------|-----------|-----------|-----------|-----------|-----------|

**#confusion matrix**

install.packages("caret")

library(caret)

confusionMatrix(table(knn\_glass\_pred3,test\_glass\_label))

Confusion Matrix and Statistics

test\_glass\_label

knn\_glass\_pred3 T1 T2 T3 T4 T5 T6 T7

T1 16 6 6 0 0 3 2

T2 6 15 1 0 1 0 2

T3 0 0 0 0 0 0 0

T4 0 0 0 0 0 0 0

T5 0 1 0 0 1 0 0

T6 0 0 0 0 0 0 0

T7 0 0 0 0 0 1 4

Overall Statistics

Accuracy : 0.5538

95% CI : (0.4253, 0.6773)

No Information Rate : 0.3385

P-Value [Acc > NIR] : 0.0002953

Kappa : 0.3511

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: T1 Class: T2 Class: T3 Class: T4 Class: T5 Class: T6 Class: T7

Sensitivity 0.7273 0.6818 0.0000 NA 0.50000 0.00000 0.50000

Specificity 0.6047 0.7674 1.0000 1 0.98413 1.00000 0.98246

Pos Pred Value 0.4848 0.6000 NaN NA 0.50000 NaN 0.80000

Neg Pred Value 0.8125 0.8250 0.8923 NA 0.98413 0.93846 0.93333

Prevalence 0.3385 0.3385 0.1077 0 0.03077 0.06154 0.12308

Detection Rate 0.2462 0.2308 0.0000 0 0.01538 0.00000 0.06154

Detection Prevalence 0.5077 0.3846 0.0000 0 0.03077 0.00000 0.07692

Balanced Accuracy 0.6660 0.7246 0.5000 NA 0.74206 0.50000 0.74123

**#using a simple loop we can find best k value**

i=1

k\_value=1

for(i in 1:25){

knn.models<-knn(train = train\_glass, test = test\_glass,cl = train\_glass\_label, k=i)

k\_value[i]<-100\*sum(test\_glass\_label == knn.models)/NROW(test\_glass\_label)

k=i

cat(k,"=",k\_value[i],'\n')

}

1 = 67.69231

2 = 60

3 = 58.46154

4 = 56.92308

5 = 55.38462

6 = 58.46154

7 = 56.92308

8 = 55.38462

9 = 56.92308

10 = 56.92308

11 = 56.92308

12 = 55.38462

13 = 56.92308

14 = 56.92308

15 = 55.38462

16 = 55.38462

17 = 55.38462

18 = 55.38462

19 = 52.30769

20 = 53.84615

21 = 55.38462

22 = 53.84615

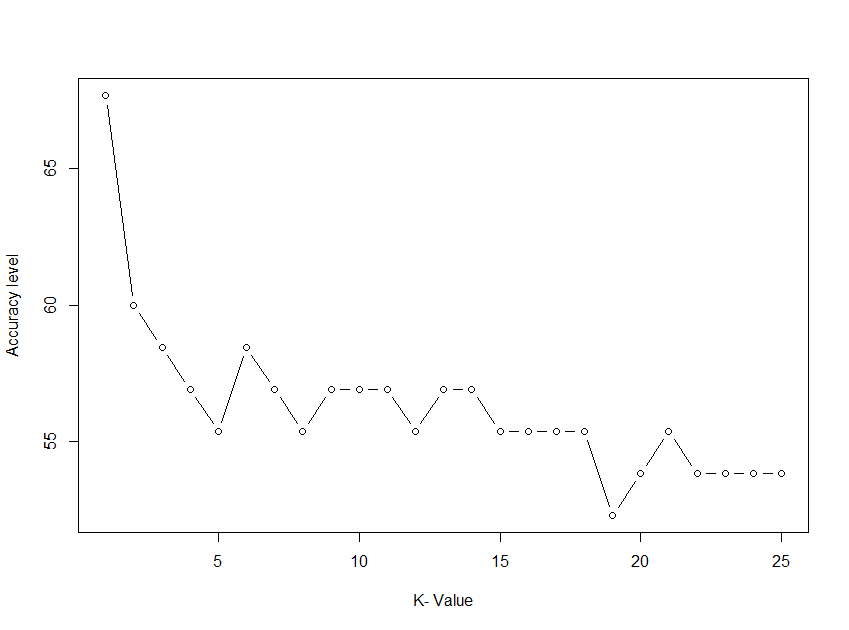
23 = 53.84615

24 = 53.84615

25 = 53.84615

**# plot the accuracy of k-value**

plot(k\_value, type="b", xlab="K- Value",ylab="Accuracy level")



**#final dataframe of k=1**

View(data.frame(test\_glass\_label,knn\_glass\_pred1))

|  | **test\_glass\_label** | **knn\_glass\_pred1** |
| --- | --- | --- |
|  |  |  |
| **1** | T1 | T2 |
| **2** | T1 | T2 |
| **3** | T1 | T3 |
| **4** | T1 | T1 |
| **5** | T1 | T1 |
| **6** | T1 | T1 |
| **7** | T1 | T3 |
| **8** | T1 | T1 |
| **9** | T1 | T1 |
| **10** | T1 | T1 |
| **11** | T1 | T1 |
| **12** | T1 | T1 |
| **13** | T1 | T3 |
| **14** | T1 | T2 |
| **15** | T1 | T2 |
| **16** | T1 | T1 |
| **17** | T1 | T1 |
| **18** | T1 | T1 |
| **19** | T1 | T1 |
| **20** | T1 | T2 |
| **21** | T1 | T1 |
| **22** | T1 | T1 |
| **23** | T2 | T2 |
| **24** | T2 | T2 |
| **25** | T2 | T3 |
| **26** | T2 | T2 |
| **27** | T2 | T2 |
| **28** | T2 | T1 |
| **29** | T2 | T2 |
| **30** | T2 | T2 |
| **31** | T2 | T2 |
| **32** | T2 | T2 |
| **33** | T2 | T2 |
| **34** | T2 | T2 |
| **35** | T2 | T2 |
| **36** | T2 | T2 |
| **37** | T2 | T2 |
| **38** | T2 | T2 |
| **39** | T2 | T5 |
| **40** | T2 | T2 |
| **41** | T2 | T2 |
| **42** | T2 | T2 |
| **43** | T2 | T2 |
| **44** | T2 | T2 |
| **45** | T3 | T1 |
| **46** | T3 | T1 |
| **47** | T3 | T2 |
| **48** | T3 | T3 |
| **49** | T3 | T2 |
| **50** | T3 | T3 |
| **51** | T3 | T2 |
| **52** | T5 | T5 |
| **53** | T5 | T5 |
| **54** | T6 | T6 |
| **55** | T6 | T1 |
| **56** | T6 | T6 |
| **57** | T6 | T6 |
| **58** | T7 | T2 |
| **59** | T7 | T5 |
| **60** | T7 | T1 |
| **61** | T7 | T2 |
| **62** | T7 | T7 |
| **63** | T7 | T7 |
| **64** | T7 | T7 |
| **65** | T7 | T7 |

Showing 1 to 11 of 65 entries, 2 total columns

**# take k=5**

knn\_glass\_pred5 <- knn(train = train\_glass, test = test\_glass,

cl = train\_glass\_label, k=5)

knn\_glass\_pred5

[1] T2 T2 T2 T1 T1 T1 T1 T1 T1 T1 T1 T1 T1 T2 T2 T1 T2 T1 T1 T2 T1 T1 T2 T2 T2 T2 T1 T1 T1 T1

[31] T2 T5 T2 T2 T2 T2 T1 T2 T5 T2 T2 T2 T2 T1 T1 T2 T1 T1 T2 T1 T1 T2 T2 T1 T1 T1 T6 T2 T2 T1

[61] T2 T7 T7 T7 T7

Levels: T1 T2 T3 T4 T5 T6 T7

**#calculate the proportion of correct classification for k=5**

accur<-100\*sum(test\_glass\_label==knn\_glass\_pred5)/NROW(test\_glass\_label)

accur

|  |
| --- |
| 52.30769 |
|  |
| |  | | --- | | > | |

table(knn\_glass\_pred5,test\_glass\_label)

test\_glass\_label

knn\_glass\_pred5 T1 T2 T3 T4 T5 T6 T7

T1 15 6 5 0 0 3 1

T2 7 14 2 0 2 0 3

T3 0 0 0 0 0 0 0

T4 0 0 0 0 0 0 0

T5 0 2 0 0 0 0 0

T6 0 0 0 0 0 1 0

T7 0 0 0 0 0 0 4

**#create data frame using knn\_glass\_pred5**

View(data.frame(test\_glass\_label,knn\_glass\_pred5))

head(data.frame(test\_glass\_label,knn\_glass\_pred5))

test\_glass\_label knn\_glass\_pred5

1 T1 T2

2 T1 T2

3 T1 T2

4 T1 T1

5 T1 T1

6 T1 T1

**#conclusion:**

**#here we are plotting the the accuracy of different k value**

**# k=1 will give the better accuracy**