Prajakta Deokule

3330

A1

ASSIGNMENT 3

Part 3- K Means for Glass Dataset

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import weka.clusterers.SimpleKMeans;
import weka.core.Instances;
import java.util.*;
public class KMeansDataset
{
    public static BufferedReader readDataFile(String filename) {
    BufferedReader inputReader = null;

      try
      {
       inputReader = new BufferedReader(new FileReader(filename));
      }
      catch (FileNotFoundException ex) {
         System.err.println("File not found: " + filename);
       }
       return inputReader;
    }
    public static void main(String[] args) throws Exception
    {
           double start = ((System.currentTimeMillis()/1000)%60);
           Scanner scan=new Scanner(System.in);
            SimpleKMeans kmeans = new SimpleKMeans();
            kmeans.setSeed(10);
            kmeans.setPreserveInstancesOrder(true);
              //accepting number of clusters
            System.out.println("Enter number of clusters:");
            kmeans.setNumClusters(scan.nextInt());

            BufferedReader datafile = 
readDataFile("C:\\Users\\HP\\OneDrive\\Desktop\\TYSEM2\\imp  Assignments\\DMDW 
Assignments\\glass.arff");
            Instances data = new Instances(datafile);

            kmeans.buildClusterer(data);
            int[] assignments = kmeans.getAssignments();
            int i=0;
            for(int clusterNum : assignments)
            {
              System.out.printf("Instance %d -> Cluster %d \n", i, clusterNum);
               i++;
            }
         double end = ((System.currentTimeMillis()/1000)%60);
         double elapsedTime = end - start;

System.out.println("Time elapsed: "+elapsedTime+"secs");
}
}
```

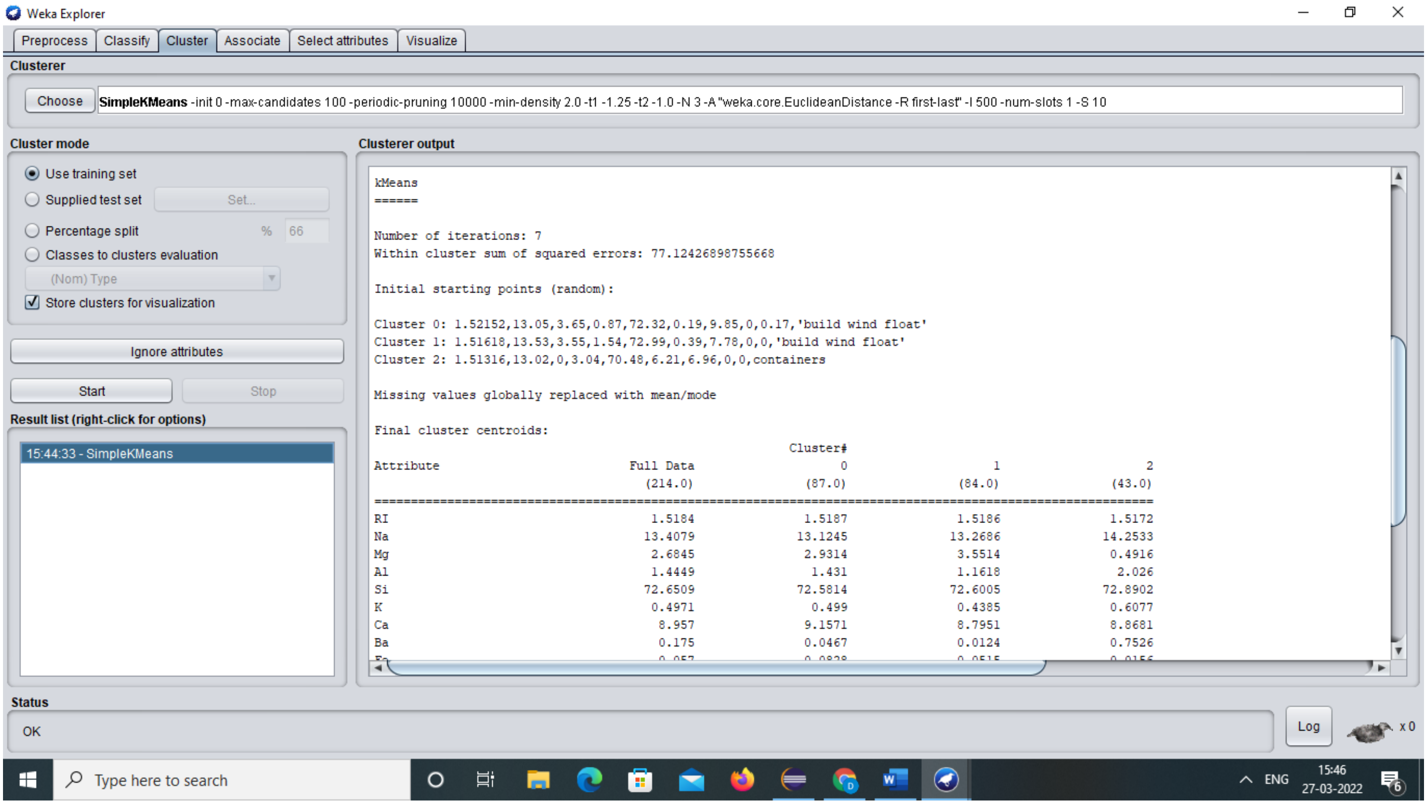
Add weka jar files from https://www.programcreek.com/2014/02/k-means-clustering-in-java/

Output of java code

```
Enter number of clusters:
3
Instance 0 -> Cluster 1
Instance 1 -> Cluster 1
Instance 2 -> Cluster 1
Instance 3 -> Cluster 2
Instance 4 -> Cluster 0
Instance 5 -> Cluster 0
Instance 6 -> Cluster 1
Instance 7 -> Cluster 1
Instance 8 -> Cluster 2
Instance 9 -> Cluster 0
Instance 10 -> Cluster 0
Instance 11 -> Cluster 0
Instance 12 -> Cluster 1
Instance 13 -> Cluster 1
Instance 14 -> Cluster 0
Instance 15 -> Cluster 0
Instance 16 -> Cluster 2
Instance 17 -> Cluster 0
Instance 18 -> Cluster 2
Instance 19 -> Cluster 0
Instance 20 -> Cluster 1
Instance 21 -> Cluster 0
Instance 22 -> Cluster 0
Instance 23 -> Cluster 1
Instance 24 -> Cluster 0
Instance 25 -> Cluster 0
Instance 26 -> Cluster 0
Instance 27 -> Cluster 2
Instance 28 -> Cluster 0
Instance 29 -> Cluster 1
Instance 30 -> Cluster 0
Instance 31 -> Cluster 1
Instance 32 -> Cluster 2
Instance 33 -> Cluster 0
Instance 34 -> Cluster 1
Instance 35 -> Cluster 1
Instance 36 -> Cluster 1
Instance 37 -> Cluster 0
Instance 38 -> Cluster 0
Instance 39 -> Cluster 0
Instance 40 -> Cluster 1
Instance 41 -> Cluster 0
Instance 42 -> Cluster 0
Instance 43 -> Cluster 0
Instance 44 -> Cluster 1
Instance 45 -> Cluster 1
Instance 46 -> Cluster 2
Instance 47 -> Cluster 1
Instance 48 -> Cluster 1
Instance 49 -> Cluster 0
Instance 50 -> Cluster 1
Instance 51 -> Cluster 0
Instance 52 -> Cluster 1
Instance 53 -> Cluster 0
Instance 54 -> Cluster 2
Instance 55 -> Cluster 1
Instance 56 -> Cluster 2
Instance 57 -> Cluster 1
Instance 58 -> Cluster 1
Instance 59 -> Cluster 0
Instance 60 -> Cluster 0
Instance 61 -> Cluster 1
Instance 62 -> Cluster 1
Instance 63 -> Cluster 1
Instance 64 -> Cluster 1
Instance 65 -> Cluster 2
Instance 66 -> Cluster 1
Instance 67 -> Cluster 0
Instance 68 -> Cluster 0
Instance 69 -> Cluster 1
Instance 70 -> Cluster 0
Instance 71 -> Cluster 0
Instance 72 -> Cluster 2
Instance 73 -> Cluster 0
Instance 74 -> Cluster 1
Instance 75 -> Cluster 0
Instance 76 -> Cluster 1
Instance 77 -> Cluster 1
Instance 78 -> Cluster 1
Instance 79 -> Cluster 1
Instance 80 -> Cluster 1
Instance 81 -> Cluster 1
Instance 82 -> Cluster 1
Instance 83 -> Cluster 1
Instance 84 -> Cluster 0
Instance 85 -> Cluster 0
Instance 86 -> Cluster 0
Instance 87 -> Cluster 1
Instance 88 -> Cluster 1
Instance 89 -> Cluster 0
Instance 90 -> Cluster 2
Instance 91 -> Cluster 1
Instance 92 -> Cluster 1
Instance 93 -> Cluster 0
Instance 94 -> Cluster 2
Instance 95 -> Cluster 1
Instance 96 -> Cluster 2
Instance 97 -> Cluster 0
Instance 98 -> Cluster 0
Instance 99 -> Cluster 0
Instance 100 -> Cluster 1
Instance 101 -> Cluster 0
Instance 102 -> Cluster 2
Instance 103 -> Cluster 2
Instance 104 -> Cluster 2
Instance 105 -> Cluster 2
Instance 106 -> Cluster 0
Instance 107 -> Cluster 2
Instance 108 -> Cluster 1
Instance 109 -> Cluster 2
Instance 110 -> Cluster 0
Instance 111 -> Cluster 2
Instance 112 -> Cluster 1
Instance 113 -> Cluster 2
Instance 114 -> Cluster 1
Instance 115 -> Cluster 2
Instance 116 -> Cluster 0
Instance 117 -> Cluster 2
Instance 118 -> Cluster 0
Instance 119 -> Cluster 1
Instance 120 -> Cluster 0
Instance 121 -> Cluster 1
Instance 122 -> Cluster 0
Instance 123 -> Cluster 0
Instance 124 -> Cluster 2
Instance 125 -> Cluster 1
Instance 126 -> Cluster 2
Instance 127 -> Cluster 2
Instance 128 -> Cluster 1
Instance 129 -> Cluster 1
Instance 130 -> Cluster 0
Instance 131 -> Cluster 0
Instance 132 -> Cluster 2
Instance 133 -> Cluster 1
Instance 134 -> Cluster 2
Instance 135 -> Cluster 1
Instance 136 -> Cluster 1
Instance 137 -> Cluster 0
Instance 138 -> Cluster 1
Instance 139 -> Cluster 0
Instance 140 -> Cluster 1
Instance 141 -> Cluster 1
Instance 142 -> Cluster 1
Instance 143 -> Cluster 2
Instance 144 -> Cluster 0
Instance 145 -> Cluster 0
Instance 146 -> Cluster 0
Instance 147 -> Cluster 0
Instance 148 -> Cluster 0
Instance 149 -> Cluster 0
Instance 150 -> Cluster 2
Instance 151 -> Cluster 1
Instance 152 -> Cluster 0
Instance 153 -> Cluster 0
Instance 154 -> Cluster 1
Instance 155 -> Cluster 1
Instance 156 -> Cluster 2
Instance 157 -> Cluster 0
Instance 158 -> Cluster 1
Instance 159 -> Cluster 0
Instance 160 -> Cluster 0
Instance 161 -> Cluster 1
Instance 162 -> Cluster 2
Instance 163 -> Cluster 2
Instance 164 -> Cluster 0
Instance 165 -> Cluster 1
Instance 166 -> Cluster 1
Instance 167 -> Cluster 0
Instance 168 -> Cluster 0
Instance 169 -> Cluster 2
Instance 170 -> Cluster 0
Instance 171 -> Cluster 0
Instance 172 -> Cluster 1
Instance 173 -> Cluster 2
Instance 174 -> Cluster 1
Instance 175 -> Cluster 2
Instance 176 -> Cluster 1
Instance 177 -> Cluster 1
Instance 178 -> Cluster 0
Instance 179 -> Cluster 2
Instance 180 -> Cluster 0
Instance 181 -> Cluster 0
Instance 182 -> Cluster 0
Instance 183 -> Cluster 0
Instance 184 -> Cluster 1
Instance 185 -> Cluster 2
Instance 186 -> Cluster 1
Instance 187 -> Cluster 1
Instance 188 -> Cluster 1
Instance 189 -> Cluster 2
Instance 190 -> Cluster 1
Instance 191 -> Cluster 1
Instance 192 -> Cluster 0
Instance 193 -> Cluster 2
Instance 194 -> Cluster 2
Instance 195 -> Cluster 0
Instance 196 -> Cluster 0
Instance 197 -> Cluster 0
Instance 198 -> Cluster 0
Instance 199 -> Cluster 0
Instance 200 -> Cluster 0
Instance 201 -> Cluster 1
Instance 202 -> Cluster 1
Instance 203 -> Cluster 1
Instance 204 -> Cluster 0
Instance 205 -> Cluster 1
Instance 206 -> Cluster 2
Instance 207 -> Cluster 1
Instance 208 -> Cluster 1
Instance 209 -> Cluster 1
Instance 210 -> Cluster 0
Instance 211 -> Cluster 0
Instance 212 -> Cluster 0
Instance 213 -> Cluster 0
Time elapsed: 1.0secs
```

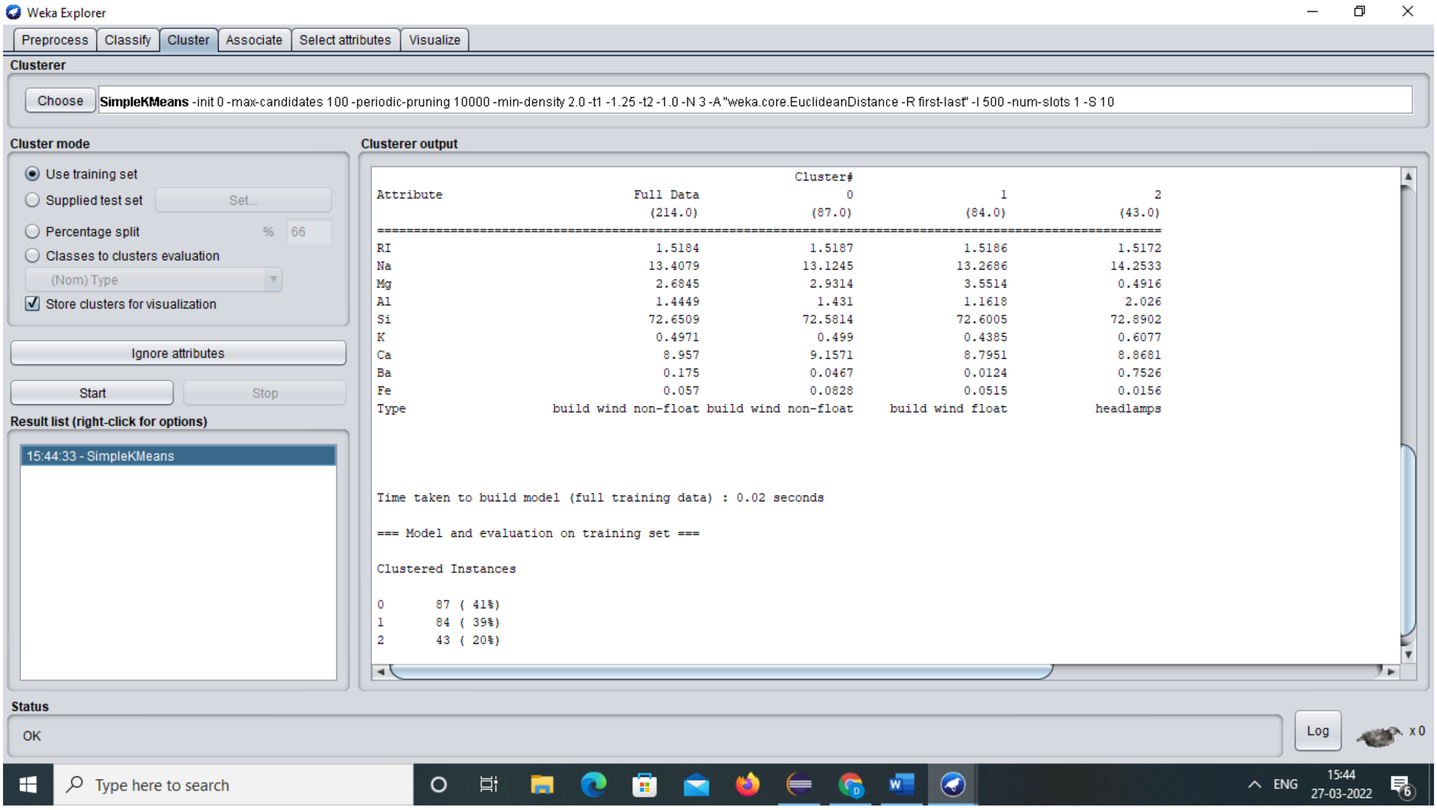
Comparison between WEKA and Java code

K=3



SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 3 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

```
kMeans
======

Number of iterations: 7
Within cluster sum of squared errors: 77.12426898755668

Initial starting points (random):

Cluster 0: 1.52152,13.05,3.65,0.87,72.32,0.19,9.85,0,0.17,'build wind float'
Cluster 1: 1.51618,13.53,3.55,1.54,72.99,0.39,7.78,0,0,'build wind float'
Cluster 2: 1.51316,13.02,0,3.04,70.48,6.21,6.96,0,0,containers

Missing values globally replaced with mean/mode

Final cluster centroids:
                                        Cluster#
Attribute               Full Data              0              1              2
                          (214.0)         (87.0)         (84.0)         (43.0)
==============================================================================
RI                         1.5184         1.5187         1.5186         1.5172
Na                        13.4079        13.1245        13.2686        14.2533
Mg                         2.6845         2.9314         3.5514         0.4916
Al                         1.4449          1.431         1.1618          2.026
Si                        72.6509        72.5814        72.6005        72.8902
K                          0.4971          0.499         0.4385         0.6077
Ca                          8.957         9.1571         8.7951         8.8681
Ba                          0.175         0.0467         0.0124         0.7526
```



Clusterer

SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 3 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Cluster mode

- ● Use training set
- ○ Supplied test set
- ○ Percentage split % 66
- ○ Classes to clusters evaluation (Nom) Type
- ☑ Store clusters for visualization

Result list (right-click for options)

15:44:33 - SimpleKMeans

Clusterer output

```
                                        Cluster#
Attribute               Full Data              0                  1                  2
                          (214.0)           (87.0)             (84.0)             (43.0)
==============================================================================================
RI                         1.5184           1.5187             1.5186             1.5172
Na                        13.4079          13.1245            13.2686            14.2533
Mg                         2.6845           2.9314             3.5514             0.4916
Al                         1.4449            1.431             1.1618              2.026
Si                        72.6509          72.5814            72.6005            72.8902
K                          0.4971            0.499             0.4385             0.6077
Ca                          8.957           9.1571             8.7951             8.8681
Ba                          0.175           0.0467             0.0124             0.7526
Fe                          0.057           0.0828             0.0515             0.0156
Type         build wind non-float build wind non-float  build wind float          headlamps




Time taken to build model (full training data) : 0.02 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       87 ( 41%)
1       84 ( 39%)
2       43 ( 20%)
```

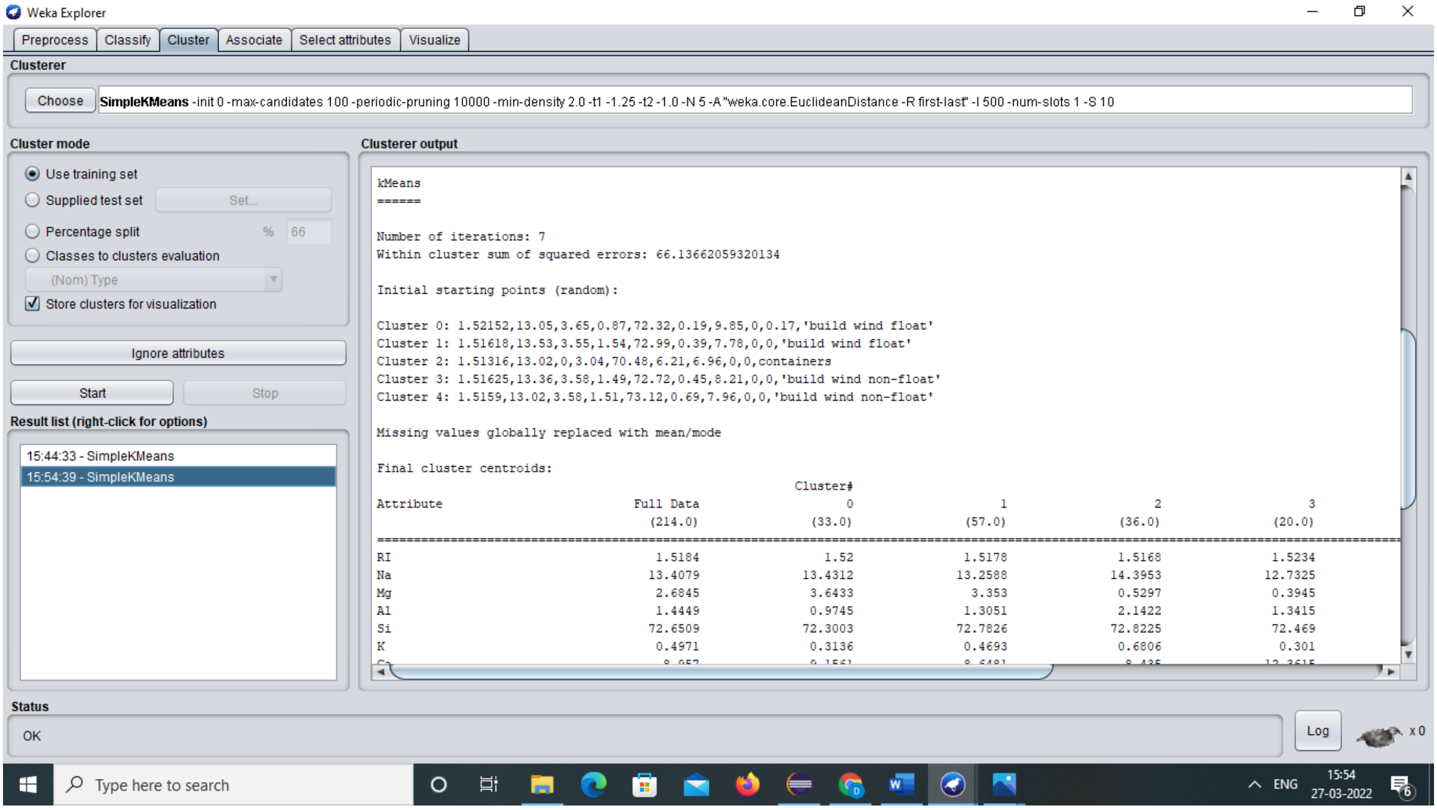
Status

OK

Java code:

`Time elapsed: 1.0secs`

K=5



Clusterer

SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 5 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Cluster mode

- ● Use training set
- ○ Supplied test set
- ○ Percentage split % 66
- ○ Classes to clusters evaluation (Nom) Type
- ☑ Store clusters for visualization

Result list (right-click for options)

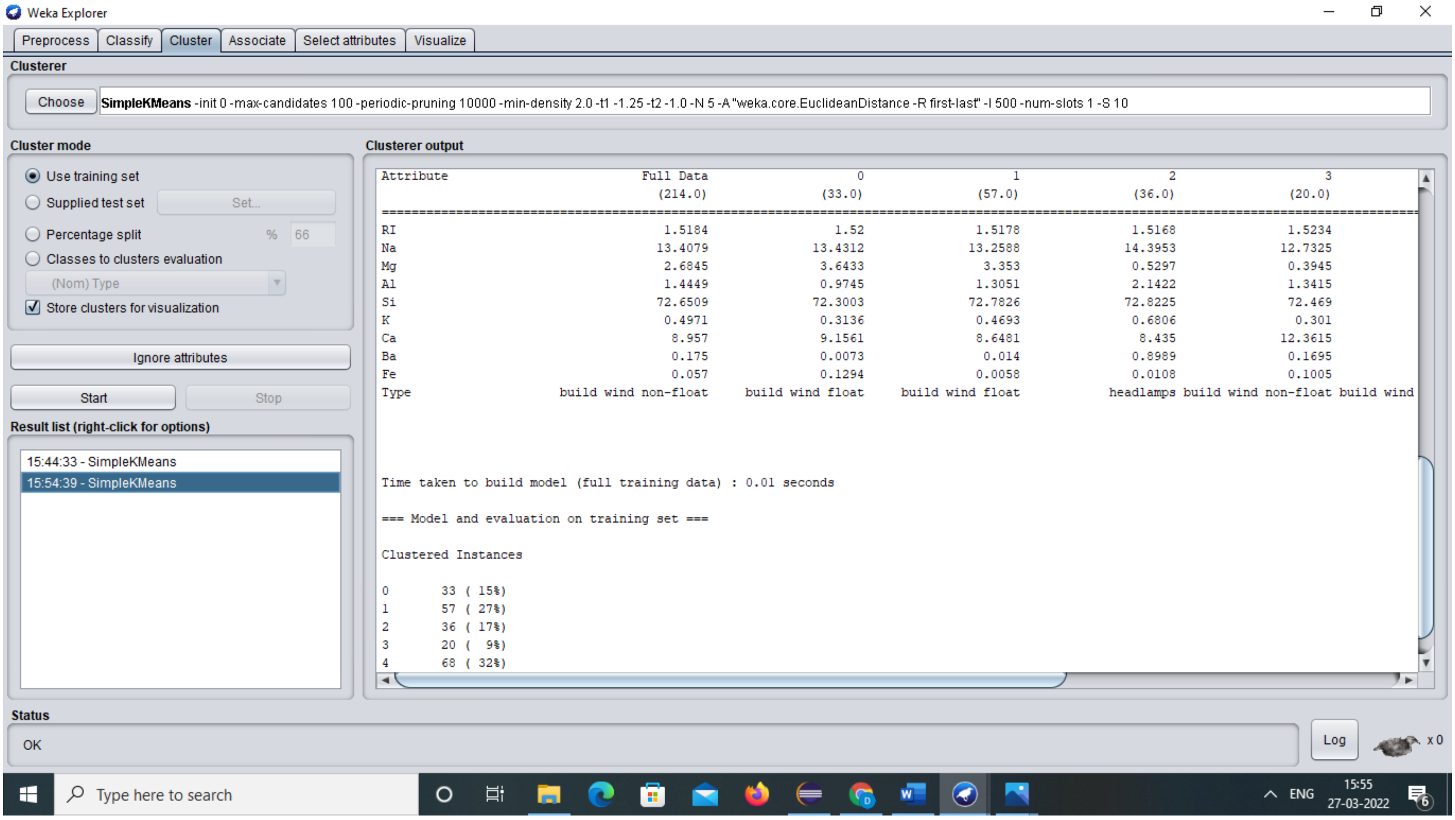
15:44:33 - SimpleKMeans
15:54:39 - SimpleKMeans

Clusterer output

```
kMeans
======

Number of iterations: 7
Within cluster sum of squared errors: 66.13662059320134

Initial starting points (random):

Cluster 0: 1.52152,13.05,3.65,0.87,72.32,0.19,9.85,0,0.17,'build wind float'
Cluster 1: 1.51618,13.53,3.55,1.54,72.99,0.39,7.78,0,0,'build wind float'
Cluster 2: 1.51316,13.02,0,3.04,70.48,6.21,6.96,0,0,containers
Cluster 3: 1.51625,13.36,3.58,1.49,72.72,0.45,8.21,0,0,'build wind non-float'
Cluster 4: 1.5159,13.02,3.58,1.51,73.12,0.69,7.96,0,0,'build wind non-float'

Missing values globally replaced with mean/mode

Final cluster centroids:
                                        Cluster#
Attribute               Full Data              0                  1                  2                  3
                          (214.0)           (33.0)             (57.0)             (36.0)             (20.0)
=============================================================================================================
RI                         1.5184             1.52             1.5178             1.5168             1.5234
Na                        13.4079          13.4312            13.2588            14.3953            12.7325
Mg                         2.6845           3.6433              3.353             0.5297             0.3945
Al                         1.4449           0.9745             1.3051             2.1422             1.3415
Si                        72.6509          72.3003            72.7826            72.8225             72.469
K                          0.4971           0.3136             0.4693             0.6806              0.301
```

Status

OK



SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 5 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Cluster mode

● Use training set
○ Supplied test set
○ Percentage split % 66
○ Classes to clusters evaluation
(Nom) Type
☑ Store clusters for visualization

15:44:33 - SimpleKMeans
15:54:39 - SimpleKMeans

Clusterer output

```
Attribute                  Full Data             0             1             2             3
                             (214.0)        (33.0)        (57.0)        (36.0)        (20.0)
=============================================================================================
RI                            1.5184          1.52        1.5178        1.5168        1.5234
Na                           13.4079       13.4312       13.2588       14.3953       12.7325
Mg                            2.6845        3.6433         3.353        0.5297        0.3945
Al                            1.4449        0.9745        1.3051        2.1422        1.3415
Si                           72.6509       72.3003       72.7826       72.8225        72.469
K                             0.4971        0.3136        0.4693        0.6806         0.301
Ca                             8.957        9.1561        8.6481         8.435       12.3615
Ba                             0.175        0.0073         0.014        0.8989        0.1695
Fe                             0.057        0.1294        0.0058        0.0108        0.1005
Type            build wind non-float build wind float build wind float     headlamps build wind non-float build wind




Time taken to build model (full training data) : 0.01 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       33 ( 15%)
1       57 ( 27%)
2       36 ( 17%)
3       20 (  9%)
4       68 ( 32%)
```

Status

OK

Java code:

```
Time elapsed: 3.0secs
```

Comparison

1.Java code takes initial k points (for k clusters) as initial means while WEKA uses random points as initial means.

2.Java code takes more time as compared to WEKA.

3.Java code is less efficient as compared to WEKA as the sum of squared error in Java code is quite greater than WEKA.

4.WEKA provides detailed analysis and statistics of data.

5.When the number of clusters are increased, there is drastic reduction in sum of squared errors in Java code. In case of WEKA, even though there is reduction in sum of squared errors, there is no drastic change.