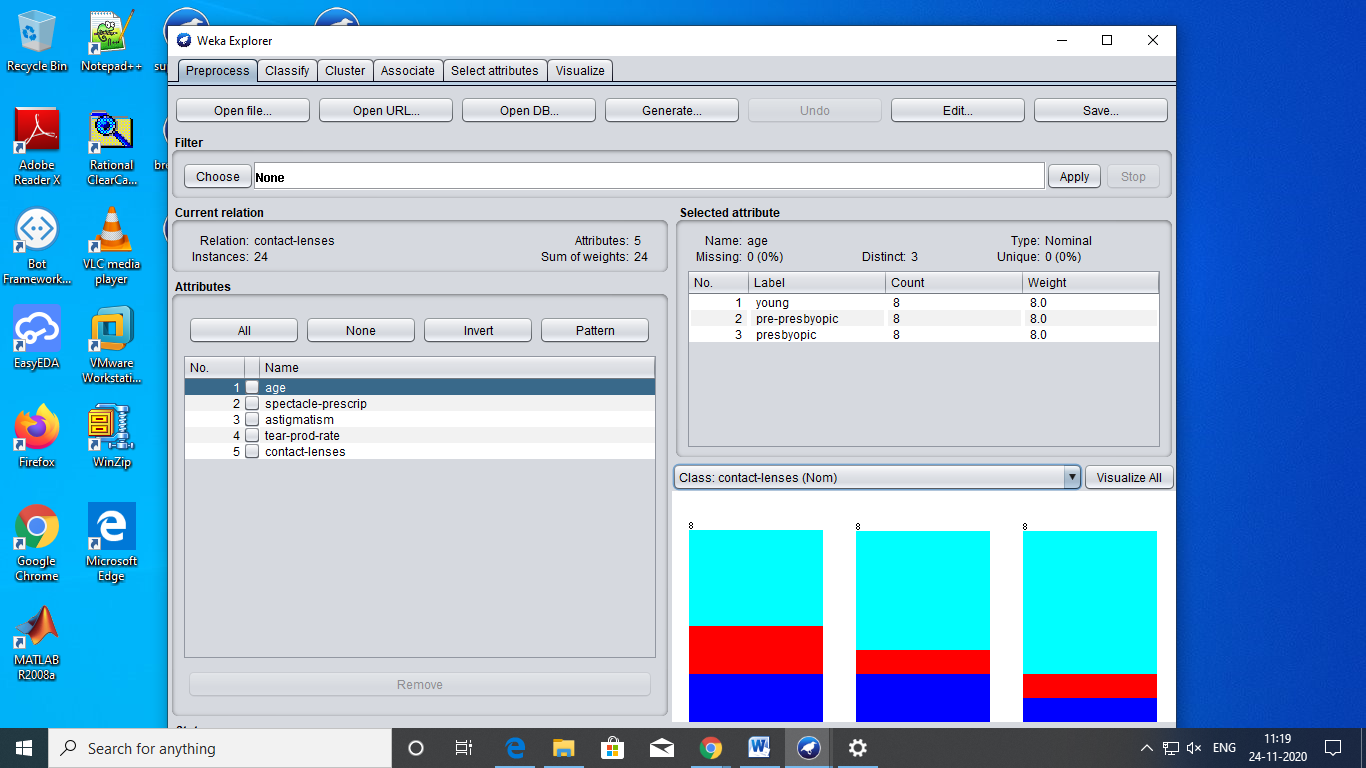
1. **Write a program to implement k-means clustering algorithm**



=== Run information ===

Scheme: weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Relation: contact-lenses

Instances: 24

Attributes: 5

age

spectacle-prescrip

astigmatism

tear-prod-rate

contact-lenses

Test mode: evaluate on training data

=== Clustering model (full training set) ===

kMeans

======

Number of iterations: 2

Within cluster sum of squared errors: 47.0

Initial starting points (random):

Cluster 0: pre-presbyopic,myope,no,normal,soft

Cluster 1: pre-presbyopic,myope,no,reduced,none

Missing values globally replaced with mean/mode

Final cluster centroids:

Cluster#

Attribute Full Data 0 1

(24.0) (12.0) (12.0)

=====================================================

age young young young

spectacle-prescrip myope myope myope

astigmatism no no no

tear-prod-rate reduced normal reduced

contact-lenses none soft none

Time taken to build model (full training data) : 0 seconds

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

Clustered Instances

0 12 ( 50%)

1 12 ( 50%)

public void clusterData(){

kmeans = new SimpleKMeans();

kmeans.setSeed(10);

try {

kmeans.setPreserveInstancesOrder(true);

kmeans.setNumClusters(10);

kmeans.buildClusterer(cpu);

int[] assignments = kmeans.getAssignments();

int i = 0;

for(int clusterNum : assignments) {

System.out.printf("Instance %d -> Cluster %d\n", i, clusterNum);

i++;

}

} catch (Exception e1) {

}

}