TDDD41/732A75: Clustering Lab

Omkar Bhutra (omkbh878), Tejshri Mastamardi (tejma768)

4 February 2019

Goals

-Gain familiarity with the data mining toolkit, Weka -Learn to apply clustering algorithms using Weka -Understand outputs produced by clustering tools in Weka

Procedure

Dataset

In this lab we will work with a dataset from HARTIGAN (file.06). The file has been translated into ARFF, the default data file format in Weka. Download the dataset here. The dataset gives nutrient levels of 27 kinds of food. The mounts of energy, protein, fat, calcium and iron have been measured in a 3 ounce portion of the various foods.

Press the Preprocesstab. Now Press the Open button and load food.arff. A description of each attribute can be seen by selecting the attribute from the list in the left hand side of the screen. The description appears in the right hand side of the screen. Press the Edit button, you can read and edit each instances.

More info on Explorer-Preprocessing is available in the Explorer User Guide.

Cluster Data

Several clustering algorithms are implemented in Weka. In this lab we experiment with an implementation of K-means, SimpleKmeans, and an implementation of a density-based method, MakeDensityBasedClusterer in Weka.

To cluster the data, click on the Cluster tab. Press the Choose button to select the clustering algorithm. Click on the line that has appeared to the right of the Choose button to edit the properties of the algorithm. You can find a detailed description of the algorithm by pressing the More button. Set the desired properties and press OK. In the Cluster mode, select "Use training set". Press the Ignore attributes button to specify which attributes should be used in the clustering. Click Start.

Check the output on the right hand side of the screen. You can right click the result set in the "Result list" panel and view the results of clustering in a separate window. The result window shows the centroid of each cluster as well as statistics on the number and percentage of instances assigned to different clusters. Another way of understanding the characteristics of each cluster is through visualization. We can do this by right-clicking the result set on the left "Result list" panel and selecting "Visualize cluster assignments". You also can click the Save button in the visualization window and save the result as an arff file.

More info on Explorer-Clustering is available in the Explorer User Guide.

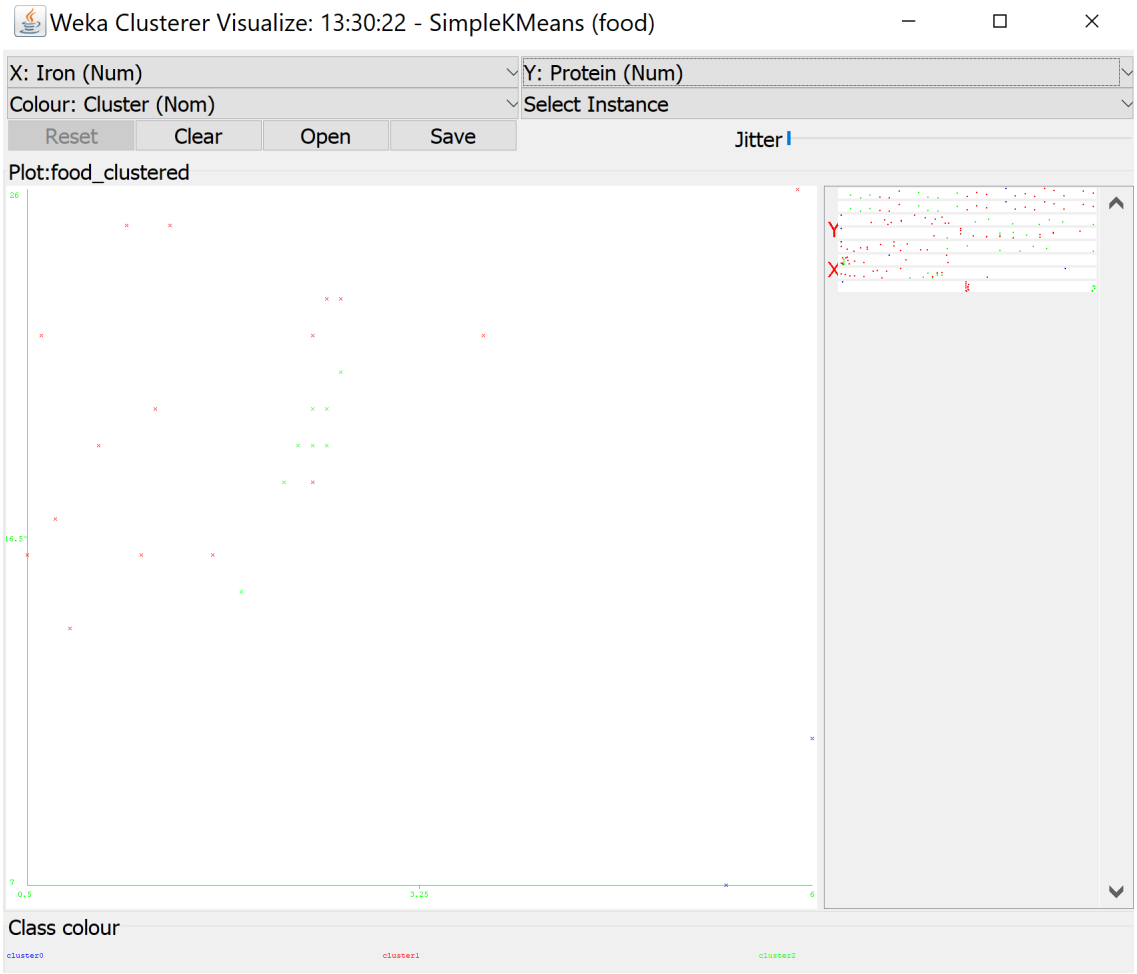
SimpleKmeans Apply "SimpleKMeans" to your data. In Weka euclidian distance is implemented in SimpleKmeans. You can set the number of clusters and seed of a random algorithm for generating initial cluster centers. Experiment with the algorithm as follows:

1. Choose a set of attributes for clustering and give a motivation. (Hint: always ignore attribute "name". Why does the name attribute need to be ignored?)

Answer

The attribute's 'Energy','Protein','Fat' and 'Calcium' are chosen because 'Iron' has relatively similar values for all food items which if included would be present at the samilar value in more than one cluster and also has few items that are outliers based on their Iron content and therefore ignored as the K-Means method is sensitive to outliers and may distort the distribution of the data. Name is only a class label and no mean value computation on the class label is possible, hence removed.

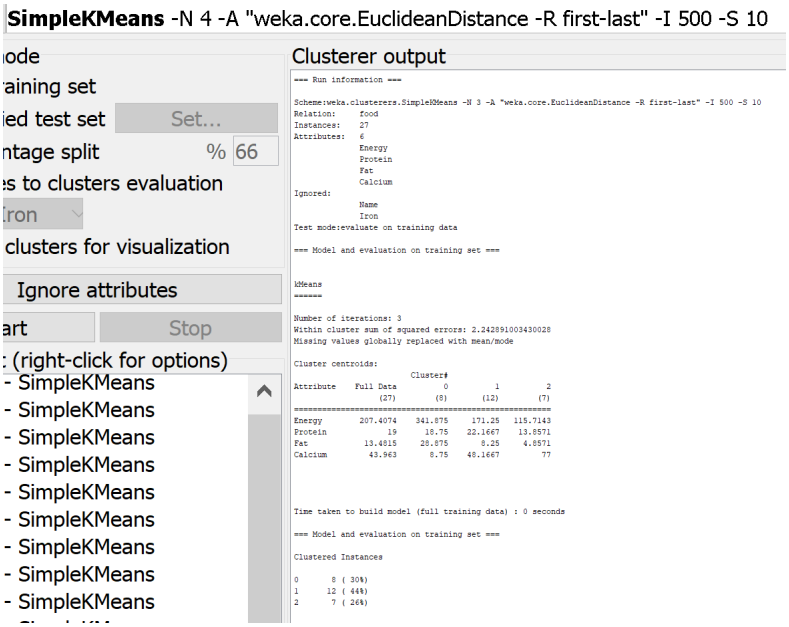
In this clustering model with all variables but 'Name', It can also been seen that a plot of Iron vs. Protein shows that most items in the 1st cluster has too few items and 3rd cluster has most items inside the region of the items of the 2nd cluster. It is suspected that this is due to the prescence of 'Iron'.



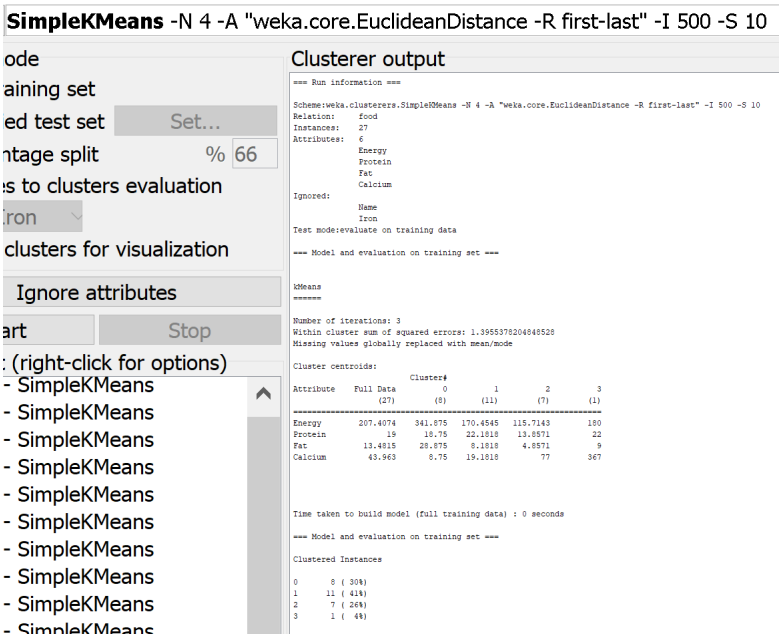
2. Experiment with at least two different numbers of clusters, e.g. 2 and 5, but with the same seed value 10.

Answer

The number of clusters, k=3



and k=4 is explored with the same seed value of 10.

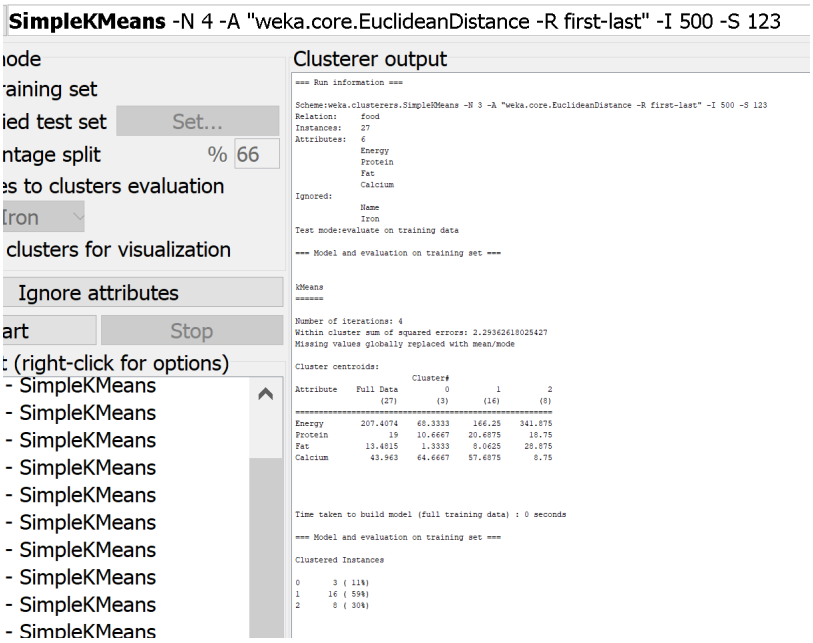


3. Then try with a different seed value, i.e. different initial cluster centers. Compare the results with the previous results. Explain what the seed value controls.

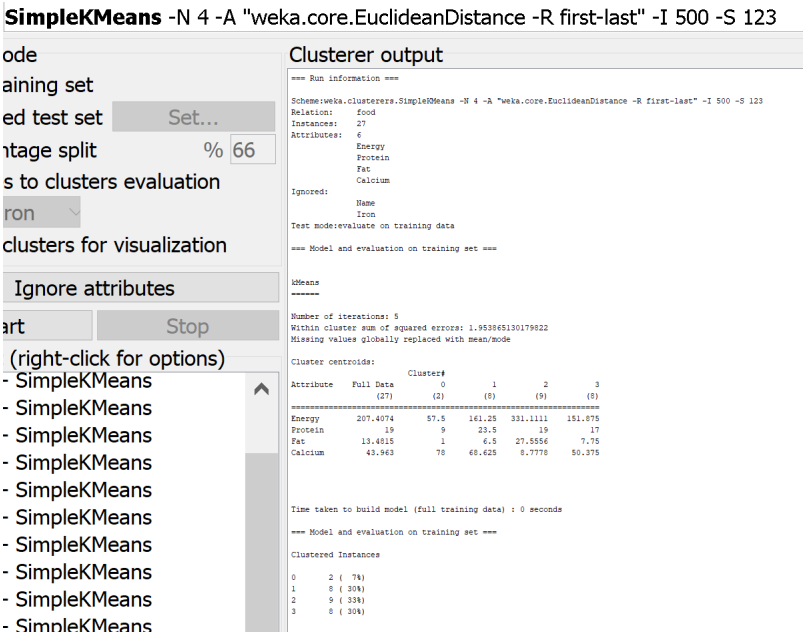
Answer

The seed value is set before any code that contain's randomness so as to run the algorithm in the same manner as done before. In the k-means clustering method, 'k' number of objects are arbitrarily chosen as initial cluster centers i.e initial mean before the algorithm updates cluster means of objects in each cluster.

The number of clusters, k=2



and k=3 is explored with the same seed value of 12.



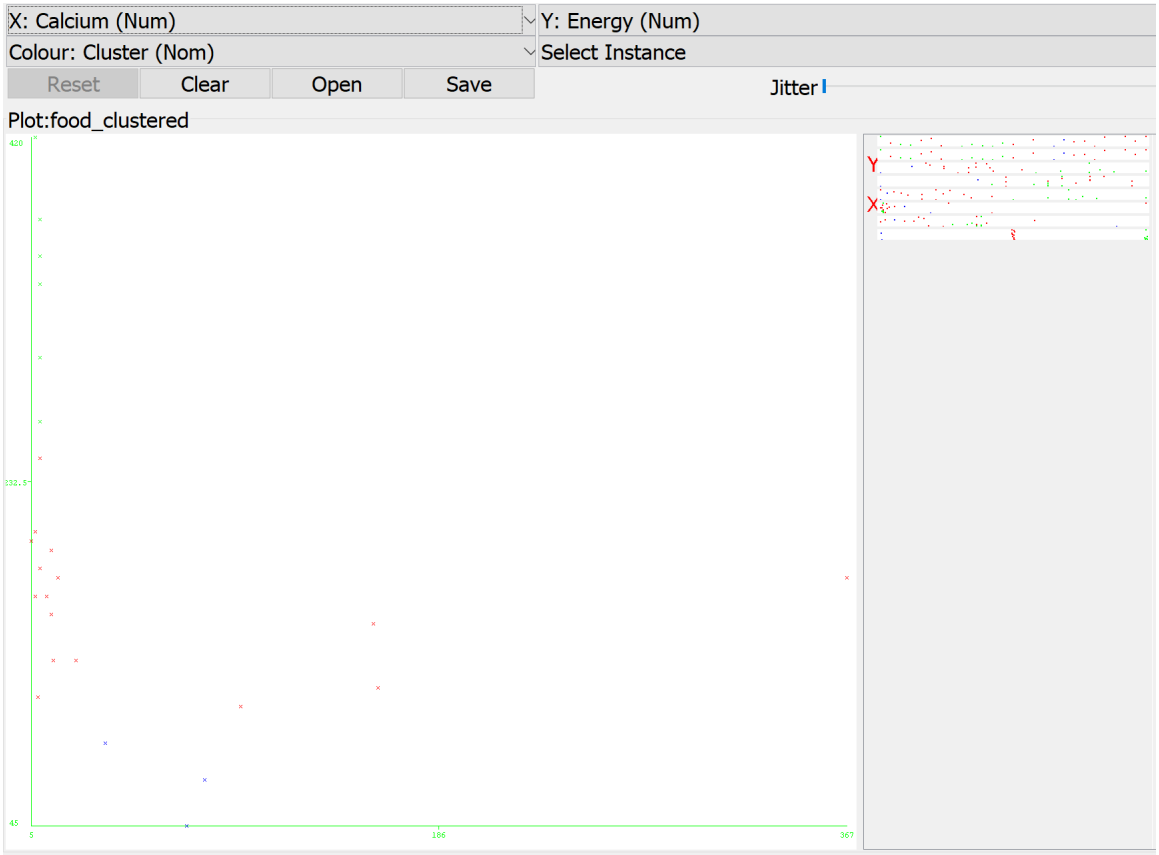
It is observed that although similar outputs are produced, the cluster means that are reported are different. This is due to the different seed value set for the algorithm and this controls the initial cluster centers and hence finally produce slightly different outputs. The cluster number 4 is missing in the first attempt while it can be observed in the 2nd attempt when k=4.

4. Do you think the clusters are “good” clusters? (Are all of its members “similar” to each other? Are members from different clusters dissimilar?)

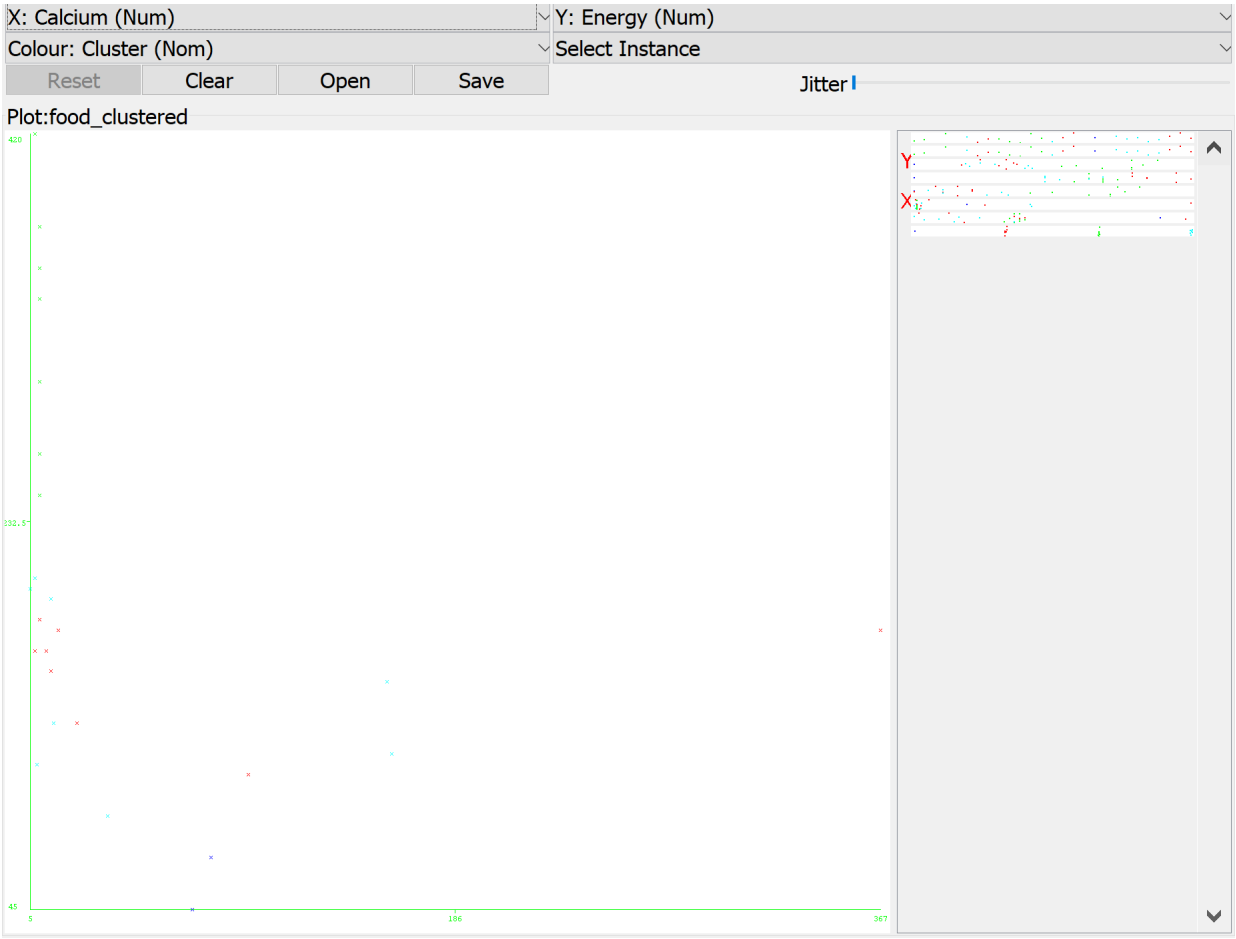
Answer

The k=3 model with the variables chosen forms good clusters as most members of a cluster are similar to each other as their attributes have distinct mean values only true to their cluster and the plot of the attributes chosen show that 2 straight lines can demarcate the almost all items in the clusters with good accuracy. All the items are not completely similar to each other as some outliers are present which distort the distribution of the data.

In the k=3 model, it is observed that one of the clusters might have 2 sub-clusters, this may be due to the prescence of a high and low value in one of the attributes for items within that cluster.



It is confirmed from the results of k=4 with seed 123, that the values of Calcium had 2 different sub clusters and hence are treated as different clusters in the k=4 model.



5. What does each cluster represent? Choose one of the results. Make up labels (words or phrases in English) which characterize each cluster.

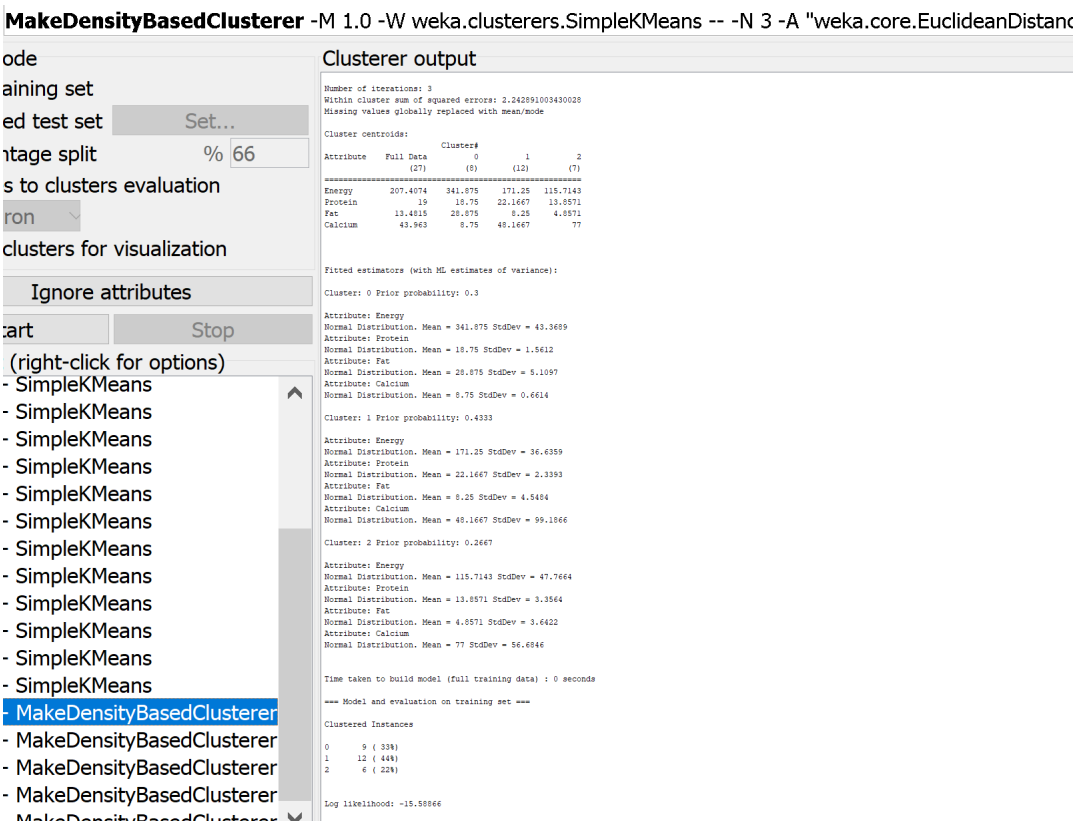
Answer

In k=3 with seed 10, 1. Low calcium and high calorific foods. 2. Protein rich food 3. Calcium rich foods with low calories.

MakeDensityBasedClusters Now with MakeDensityBasedClusters, SimpleKMeans is turned into a denstiy-based clusterer. You can set the minimum standard deviation for normal density calculation. Experiment with the algorithm as the follows:

1. Use the SimpleKMeans clusterer which gave the result you haven chosen in 5).

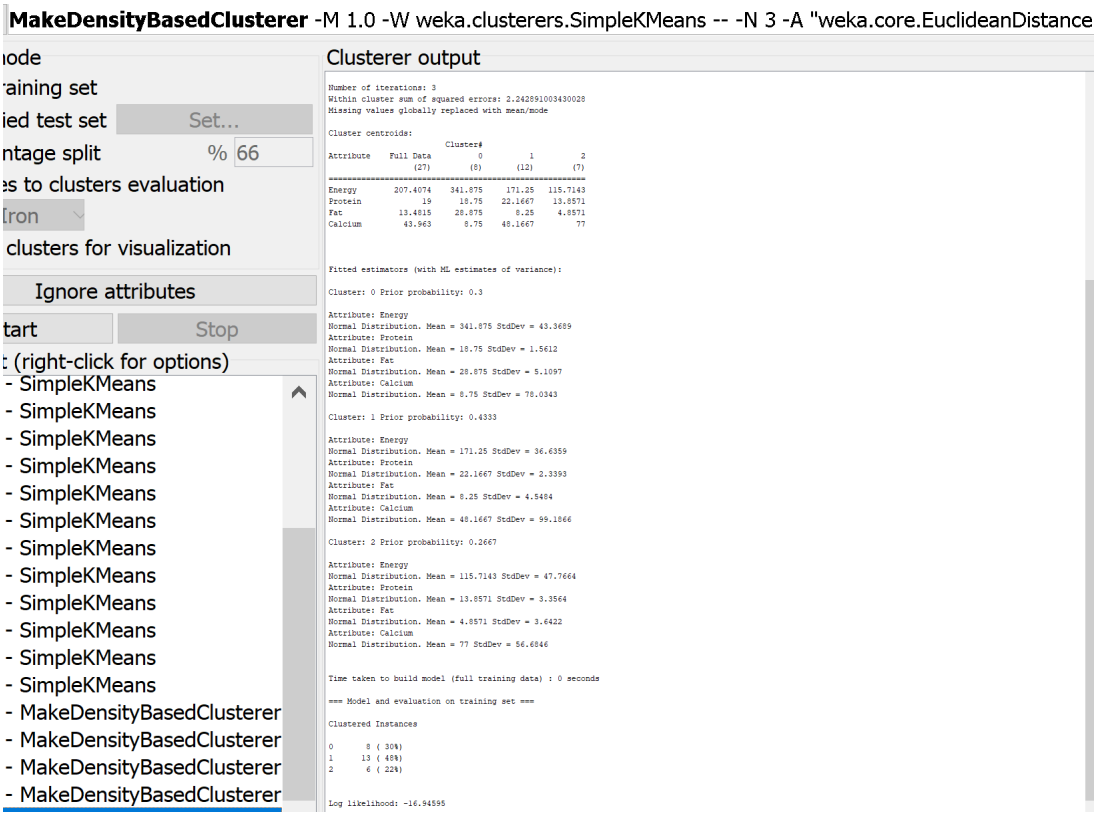
Answer



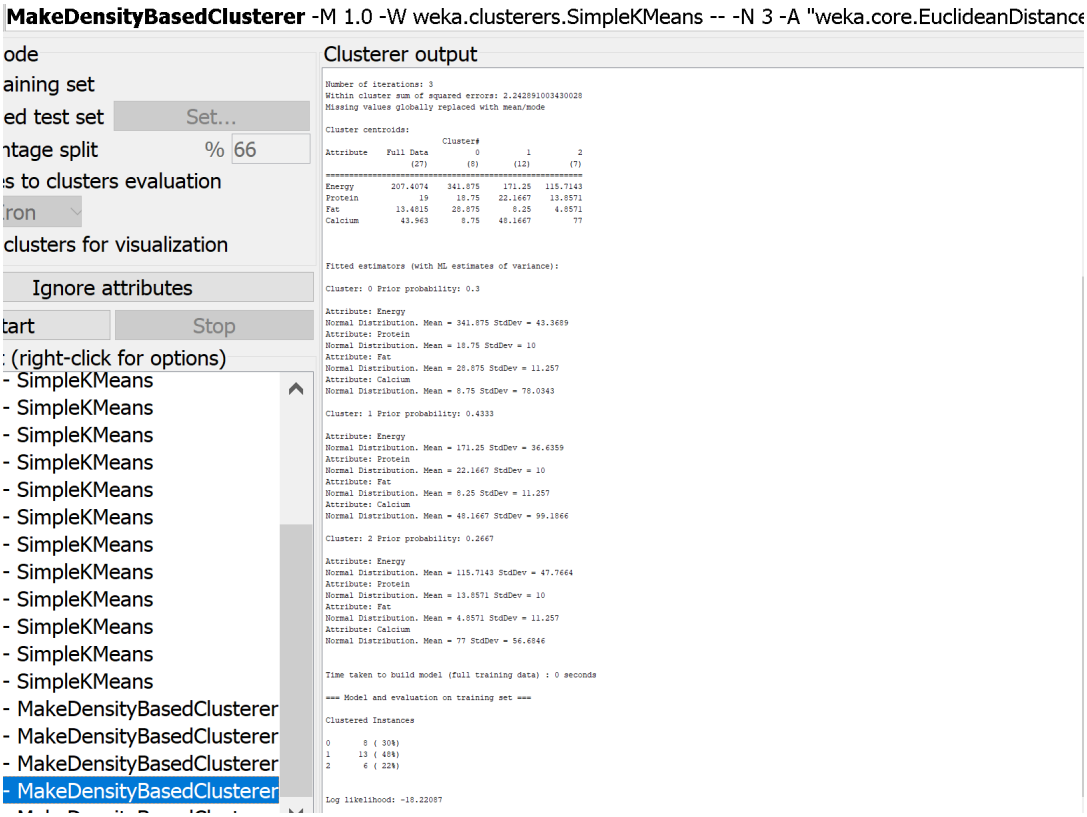
2. Experiment with at least two different standard deviations. Compare the results. (Hint: Increasing the standard deviation to higher values will make the differences in different runs more obvious and thus it will be easier to conclude what the parameter does)

Answer

The same model i.e k=3, with seed 10 is chosen but the standard deviation is set to 1 instead of 1eˆ-6. This has an effect on the log liklihood value which moves



It is seen that with the increase in the value of the standard deviation to the model, the value of the log liklihood decreases from -15.588 to -16.945. On further increasing the value of the standard deviation to 10, a negative spike in the log liklihood to -18.220 is observed.


MakeDensityBasedClusterer -M 1.0 -W weka.clusterers.SimpleKMeans -- -N 3 -A "weka.core.EuclideanDistance
ode
aining set
ed test set
Set...
ntage split
% 66
s to clusters evaluation
ron
clusters for visualization
Ignore attributes
tart
Stop
(right-click for options)
- SimpleKMeans
- SimpleKMeans
- SimpleKMeans
- SimpleKMeans
- SimpleKMeans
- SimpleKMeans
- SimpleKMeans
- SimpleKMeans
- SimpleKMeans
- SimpleKMeans
- SimpleKMeans
- SimpleKMeans
- SimpleKMeans
- MakeDensityBasedClusterer
- MakeDensityBasedClusterer
- MakeDensityBasedClusterer
- MakeDensityBasedClusterer
Clusterer output
Number of iterations: 3
Within cluster sum of squared errors: 2.2428910034300028
Missing values globally replaced with mean/mode
Cluster centroids:
Cluster#
Attribute Full Data 0 1 2
(27) (8) (12) (7)
Energy 207.4074 341.875 171.25 115.7143
Protein 19 18.75 22.1667 13.8571
Fat 13.4815 28.875 8.25 4.8571
Calcium 43.963 8.75 48.1667 77
Fitted estimators (with ML estimates of variance):
Cluster: 0 Prior probability: 0.3
Attribute: Energy
Normal Distribution. Mean = 341.875 StdDev = 43.3689
Attribute: Protein
Normal Distribution. Mean = 18.75 StdDev = 10
Attribute: Fat
Normal Distribution. Mean = 28.875 StdDev = 11.257
Attribute: Calcium
Normal Distribution. Mean = 8.75 StdDev = 78.0343
Cluster: 1 Prior probability: 0.4333
Attribute: Energy
Normal Distribution. Mean = 171.25 StdDev = 36.6359
Attribute: Protein
Normal Distribution. Mean = 22.1667 StdDev = 10
Attribute: Fat
Normal Distribution. Mean = 8.25 StdDev = 11.257
Attribute: Calcium
Normal Distribution. Mean = 48.1667 StdDev = 99.1866
Cluster: 2 Prior probability: 0.2667
Attribute: Energy
Normal Distribution. Mean = 115.7143 StdDev = 47.7664
Attribute: Protein
Normal Distribution. Mean = 13.8571 StdDev = 10
Attribute: Fat
Normal Distribution. Mean = 4.8571 StdDev = 11.257
Attribute: Calcium
Normal Distribution. Mean = 77 StdDev = 56.6846
Time taken to build model (full training data) : 0 seconds
=== Model and evaluation on training set ===
Clustered Instances
0 8 (30%)
1 13 (48%)
2 6 (22%)
Log likelihood: -18.22087