Association_lab02

thiqu264 (Thijs Quast), lensch874 (Lennart Schilling) - Group10

5-3-2019

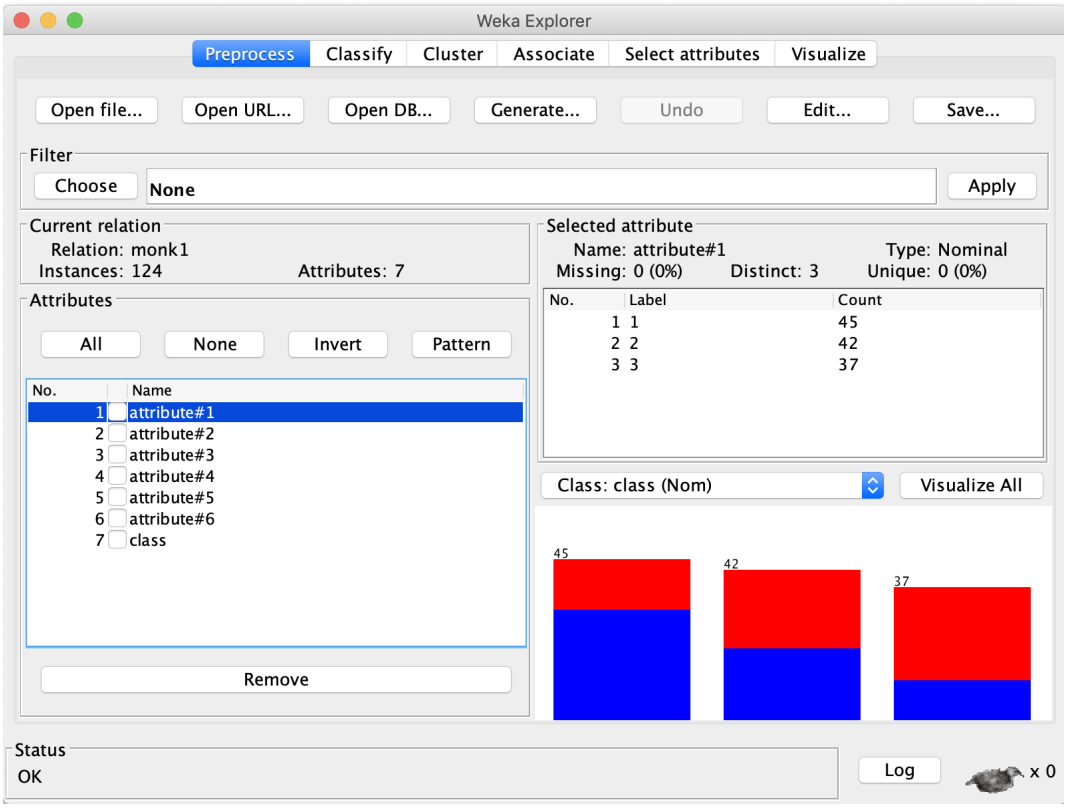
Contents

Dataset

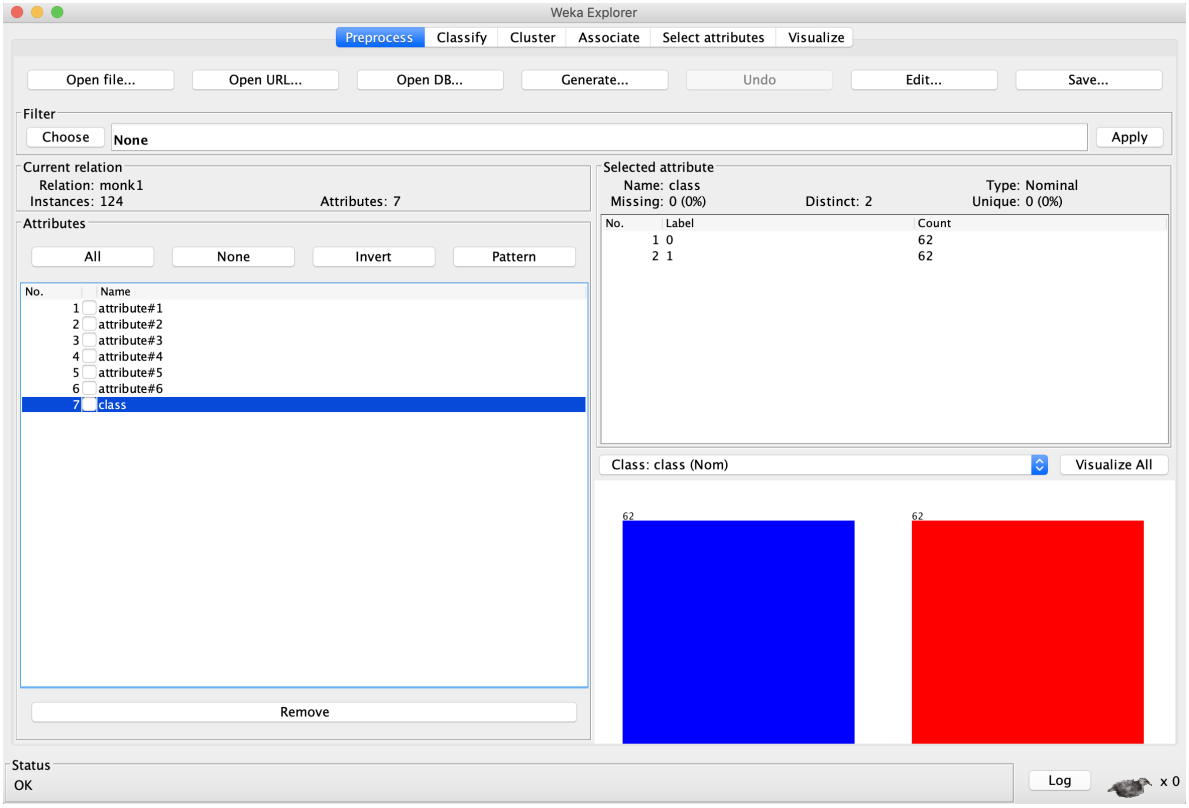
For this problem we analyze the Monk1 dataset. This dataset is particularly interesting because it was used in the first comparison of different learning algorithms. This research was so powerful because the comparisons were made by different researchers, who to some extent were supporters of one of the techniques used in the comparison.

Clusters may not correspond to classes

Import data



Classes

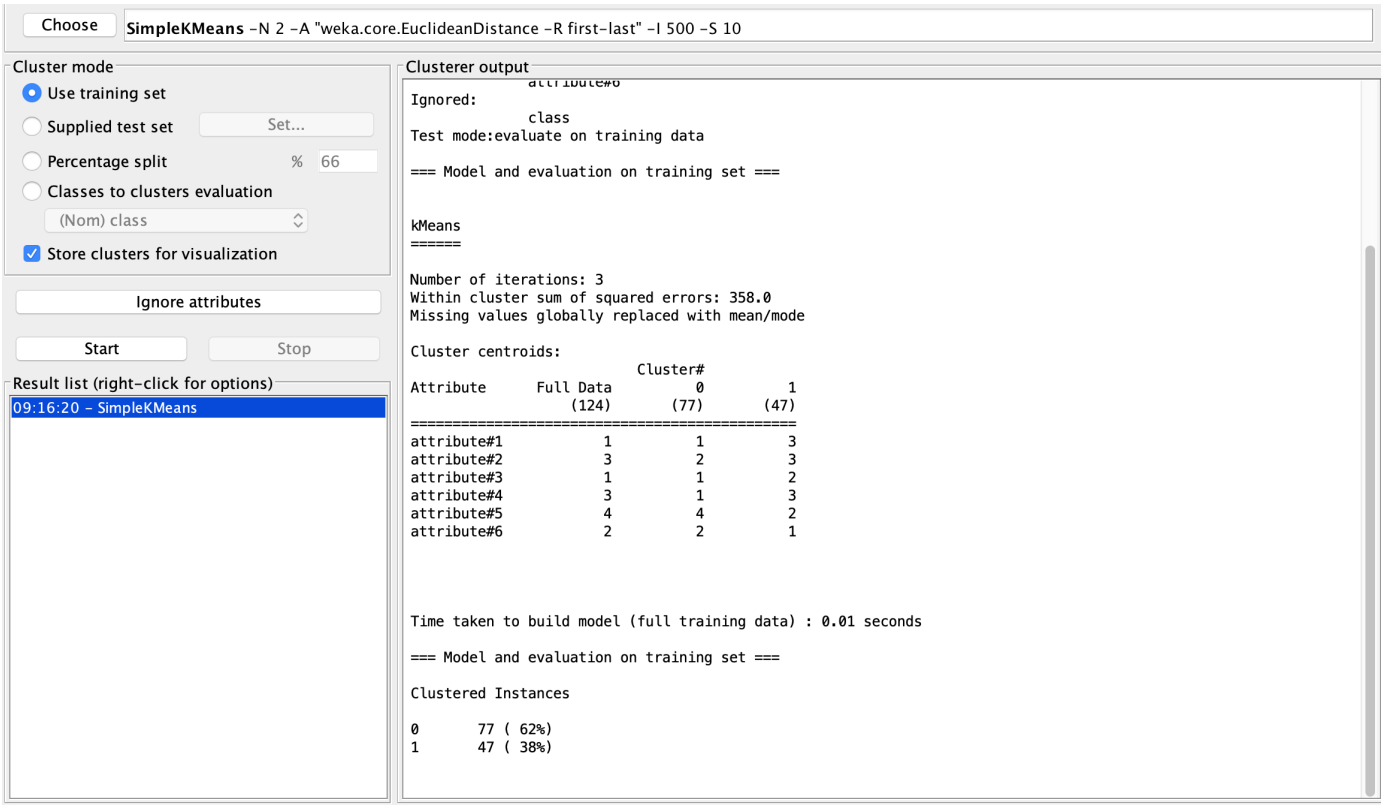


As can be seen, the dataste has two classes of 62 observations each.

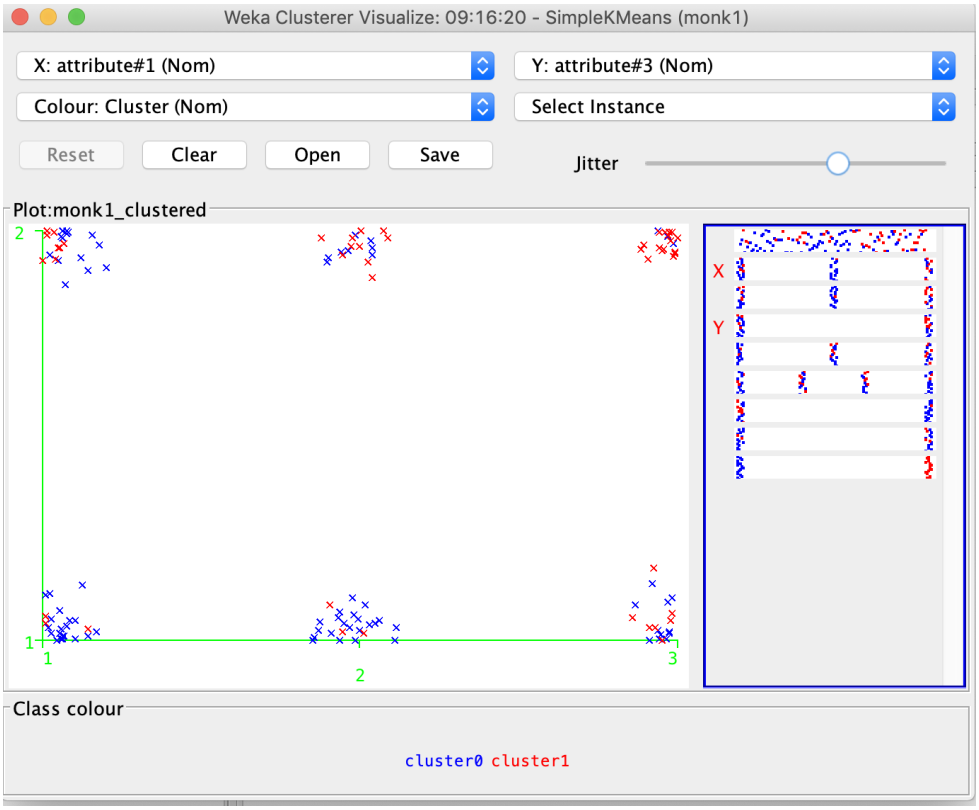
A note on the dataset

As we can see from how the data is distributed according to each attribute, it is very difficult to cluster the observations according to their class. For all the attributes the clusters shown show both blue as well as red observations, which mean it is very difficult for the algorithm to make a distinction between both.

SimpleKMeans, N=2



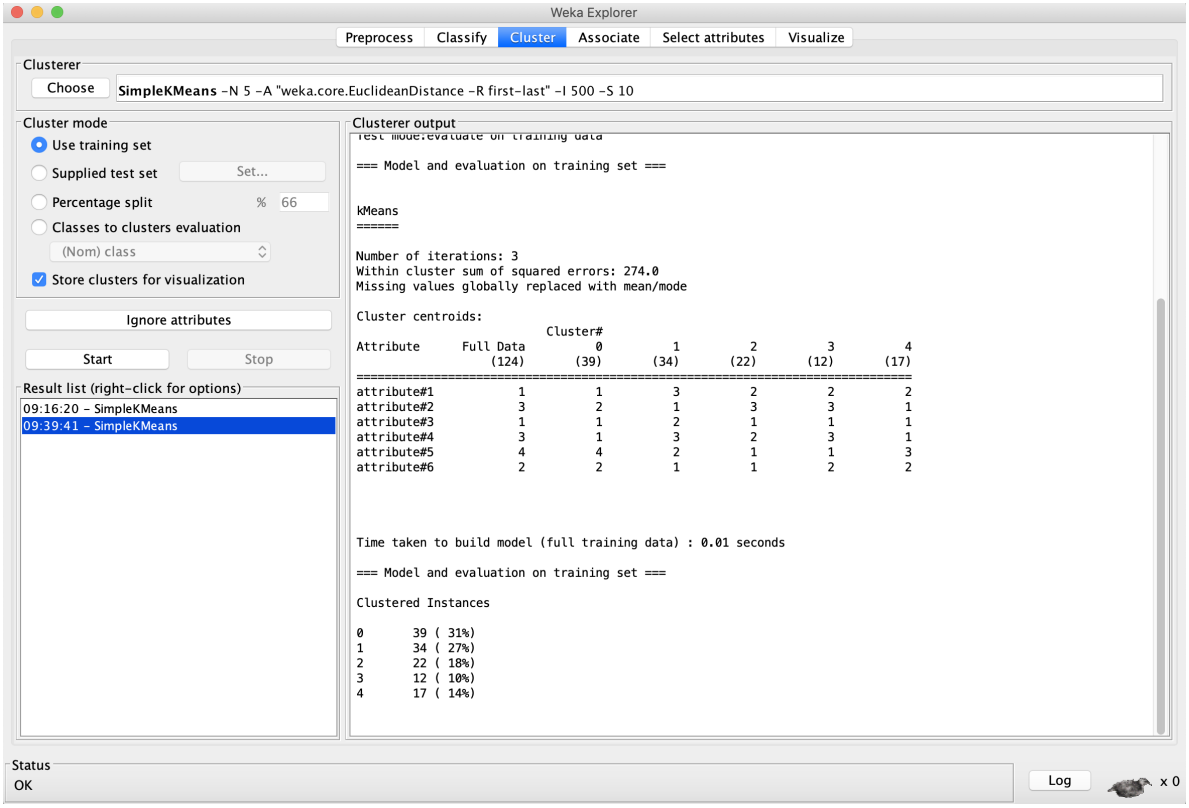
The dataset is now clustered into two clusters, using SimpleKMeans. Cluster0 has 77 observations, whereas cluster1 has 47 observations. From the knowledge we have on the dataset, that each class has 62 observations, the findings above are already an indication that the SimpleKMeans has difficulties clustering.



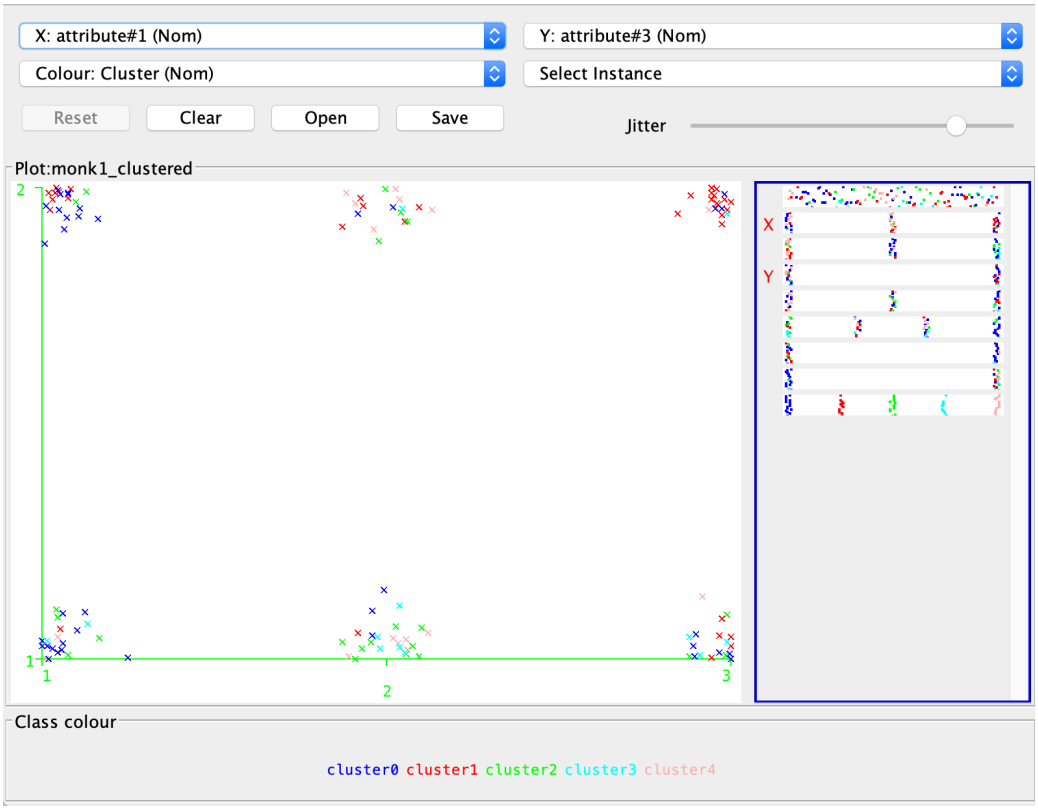
When trying to cluster according to SimpleKMeans with 2 clusters, the algorithm has a hard time clustering correctly based on the attributes. As an example we plot attribute3 against attribute1, as can be seen, in the clusters that are found, there is still overlap between between the blue and the red cluster. Meaning we cannot find specific characteristics of an observation which define to which cluster the observation belongs.

From the output we show that SimpleKMeans with 2 clusters is not able to discover the class division in the data.

SimpleKMeans, N=5



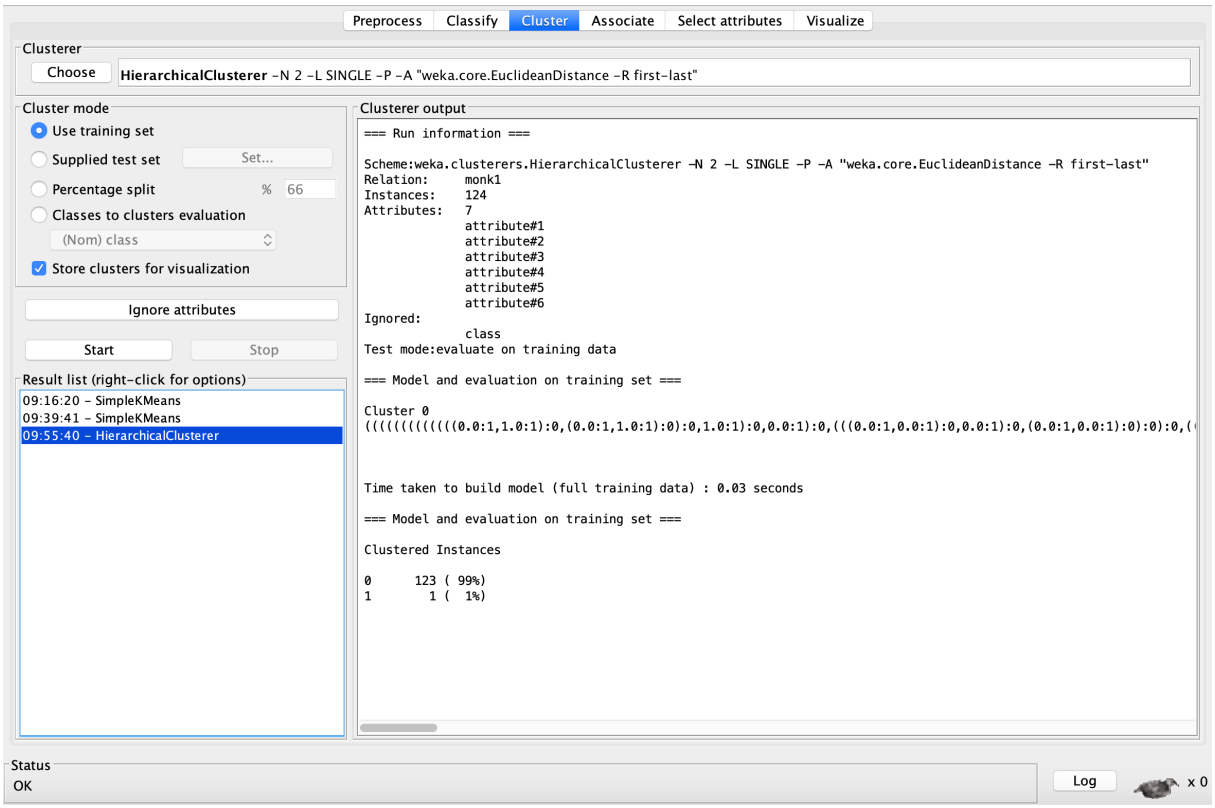
Subsequently we use SimpleKMeans to divide the data into 5 clusters. Intuitively, this does not make much sense as we already know there are only 2 clusters. Still the lab requires us to do so. Cluster sizes range from 12 (10%) to 39 (31%).



If we now plot the same attributes against each other as earlier, we can see there is again creat overlap between the observations per cluster. As expected, clustering with 5 clusters does not improve our results.

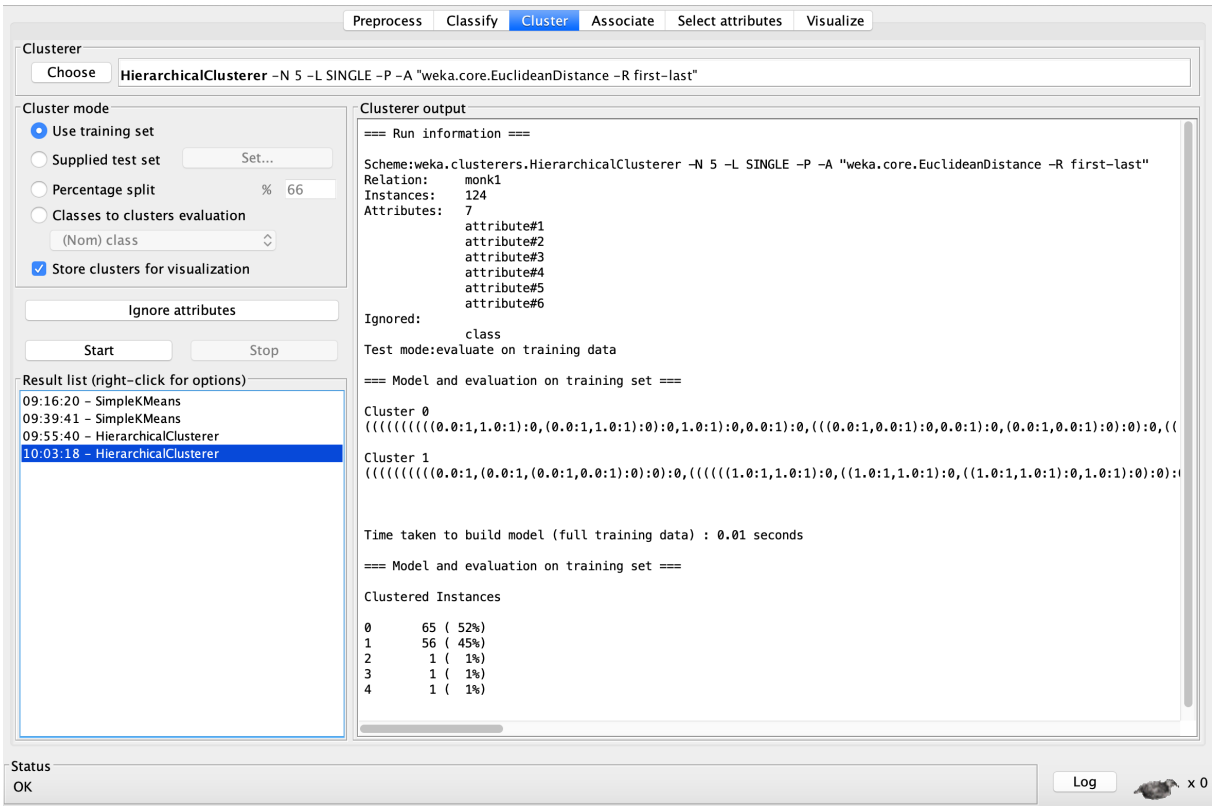
HierarichalCluster, N=2

Since SimpleKMeans did not perform well, we now try a different algorithm, namely HierarichalCluster with N=2.

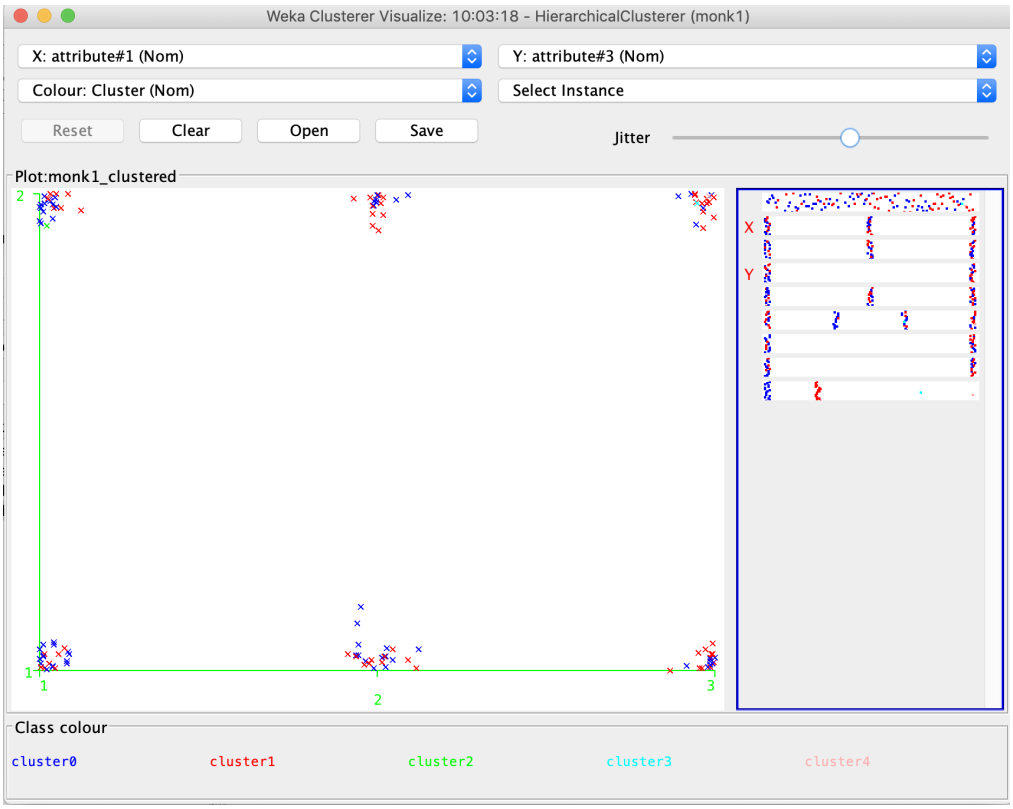


As can be seen, the algorithm now appoints, 123 (99%) of the observations to cluster0! This is clearly not what we want, as we want an equal division of observations amongst the clusters.

HierarichalCluster, N=5



As obvious, the algorithm now divides the data into 5 clusters. Although we know there are only 2 different classes, perhaps the results above still might not be so bad. As namely can be seen, cluster0 and cluster1 are almost equal. Cluster2, Cluster3 and Cluster4 only contain 1 observation each, which is not much at all.



However, when we now visualize the cluster assignment, still there is great overlap between the observations of cluster0 and cluster1.

Results from clustering

From all the results described above, we conclude that both algorithms, with different number of clusters, in all cases fail to cluster according to classes in a proper manner. This was actually already mentioned in the lab, that it is not expected that we would be able to cluster properly. Now, it is up to us to determine why clustering of this dataset is troublesome.

Association analysis

For the Association analysis we apply the Apriori algorithm. As it is a binary problem, the rules that are shown above are actually suitable for this problem. Since rules that do not indicate class=1, will therefore automatically indicate class=0. All the 19 rules which are found have a confidence of 1.

Many rules which are found are redundant, for instance the same rules require the same values for certain attributes. Therefore can narrow down the number of rules to a total of four. The four final rules that we find are:

Table 1: Rules for clusters

Rules	Class	Occurances	Confidence
attribute#5=1	1	29	1
attribute#1=3 attribute#2=3	1	17	1
attribute#1=2 attribute#2=2	1	15	1
attribute#1=1 attribute#2=1	1	9	1

Why can the clustering algorithms not find a clustering that matches the class division in the database?

As can be seen from the scatterplot above, when an observation has a certain value for e.g. attribute 1 and attribute 3, it can still belong to both classes. Therefore when clustering, this is not very useful. However, when performing an Association Analysis, perhaps the algorithm comes up with a new rule for each class based on the values for the attributes.

The problem with clustering this dataset is that we do not know how the classes were defined. When clustering, usually we look at distances among the observations, however classes could be defined in another way. Therefore, for a clustering algorithm is is difficult to cluster if classes are not defined based on distances.

Do clustering algorithms perform poorly?

Based on the monk1 dataset we argue that the clustering algorithms perform poorly. However, this is particulary due to how the dataset is constructed. As mentioned above, it is difficult to determine how classes are defined. If the dataset looks differently, the algorithms could perform bad. However, based on this dataset we argue the clustering algorithms perform poorly.