

# Advanced Data Mining

## Lab 1: Clustering

Group 08  
Mohsen Pirmoradiyan, Ahmed Alhasan

February 26, 2020

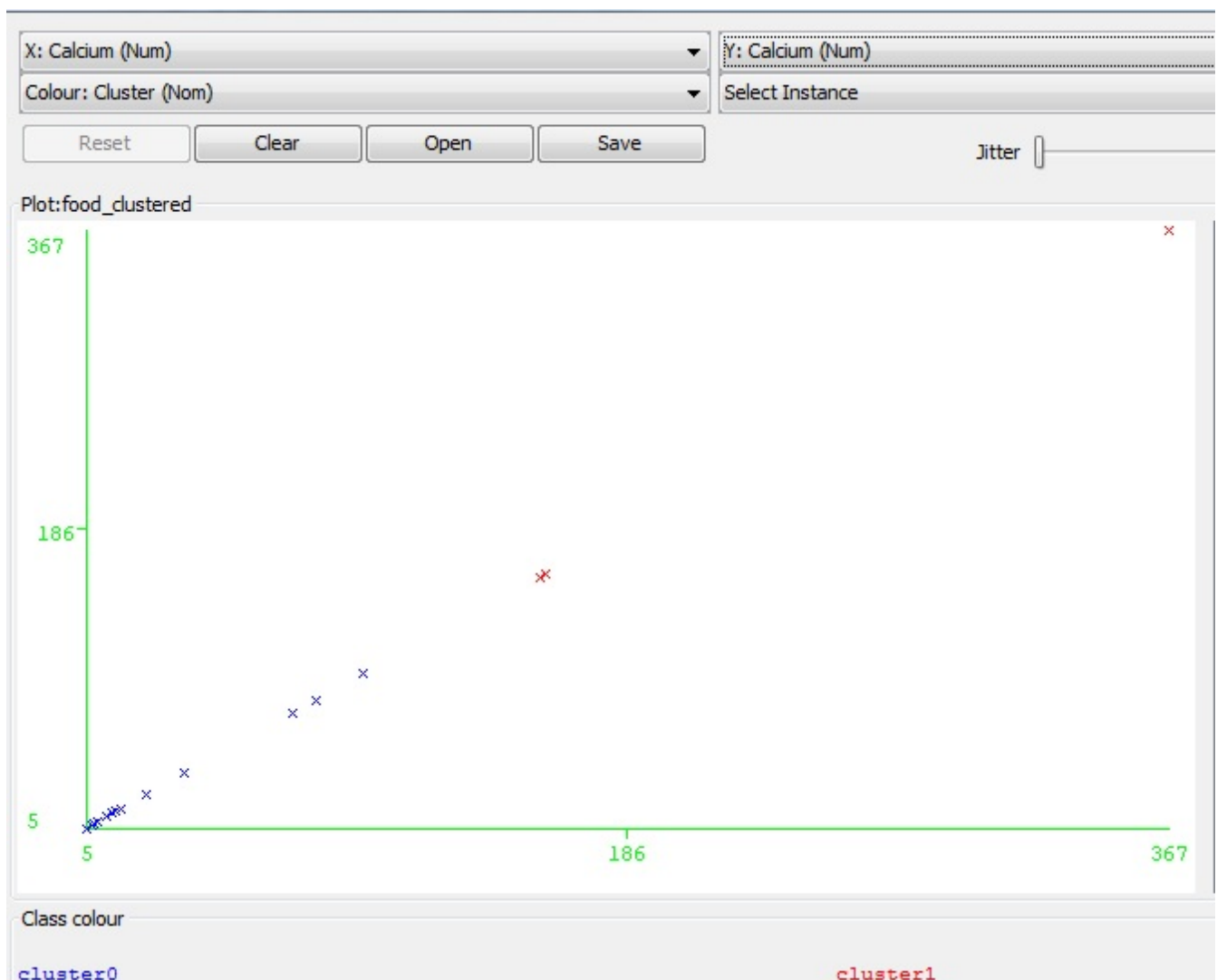
### SimpleKmeans

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:

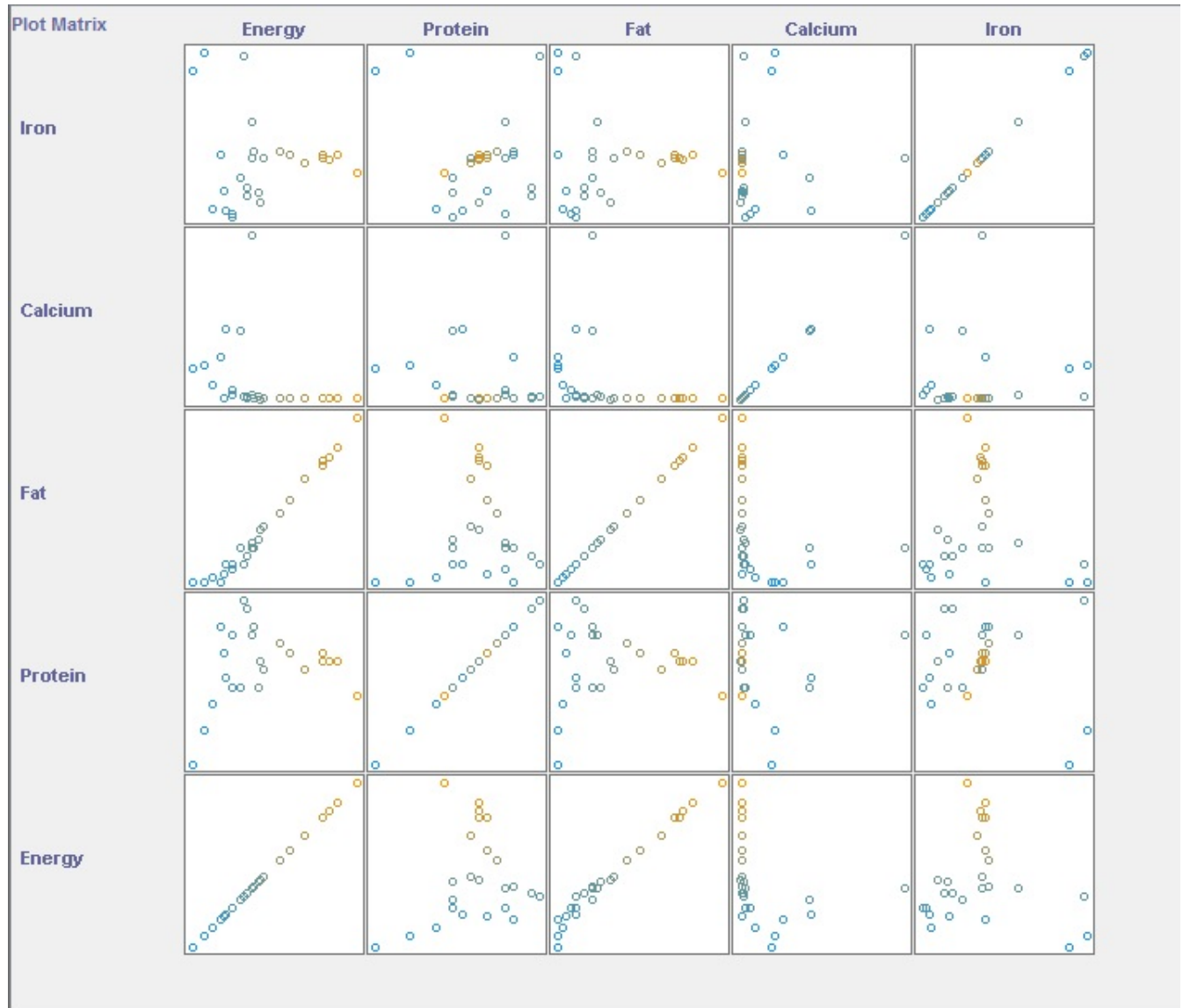
1. Choose a set of attributes for clustering and give a motivation. (**Hint:** always ignore attribute “name”. Why does the name attribute need to be ignored?)
  - Chosing a single attribute will yield the best cluster separation as we get the least within cluster sum of squared errors, however the result clusters only separate the food in regard to low-high values of that attribute and does not show the relation with other attributes.  
For example using only Calcium will result in the lowest within cluster sum of squared errors and that is mainly because of an outlier

```
Number of iterations: 5
Within cluster sum of squared errors: 0.3390209904052583
Missing values globally replaced with mean/mode

Cluster centroids:
Attribute      Full Data      Cluster#
                (27)        (24)        (3)
=====
Calcium        43.963         21    227.6667
```



- Choosing two attributes we get less separation in general, however there are some clustering patterns to be noticed for example choosing Energy and Fat will result in a well separated clusters due to their correlation even better than some single attribute clustering i.e. Iron. The resulted clusters from Energy-Fat describe how high Fat food is also high with Energy



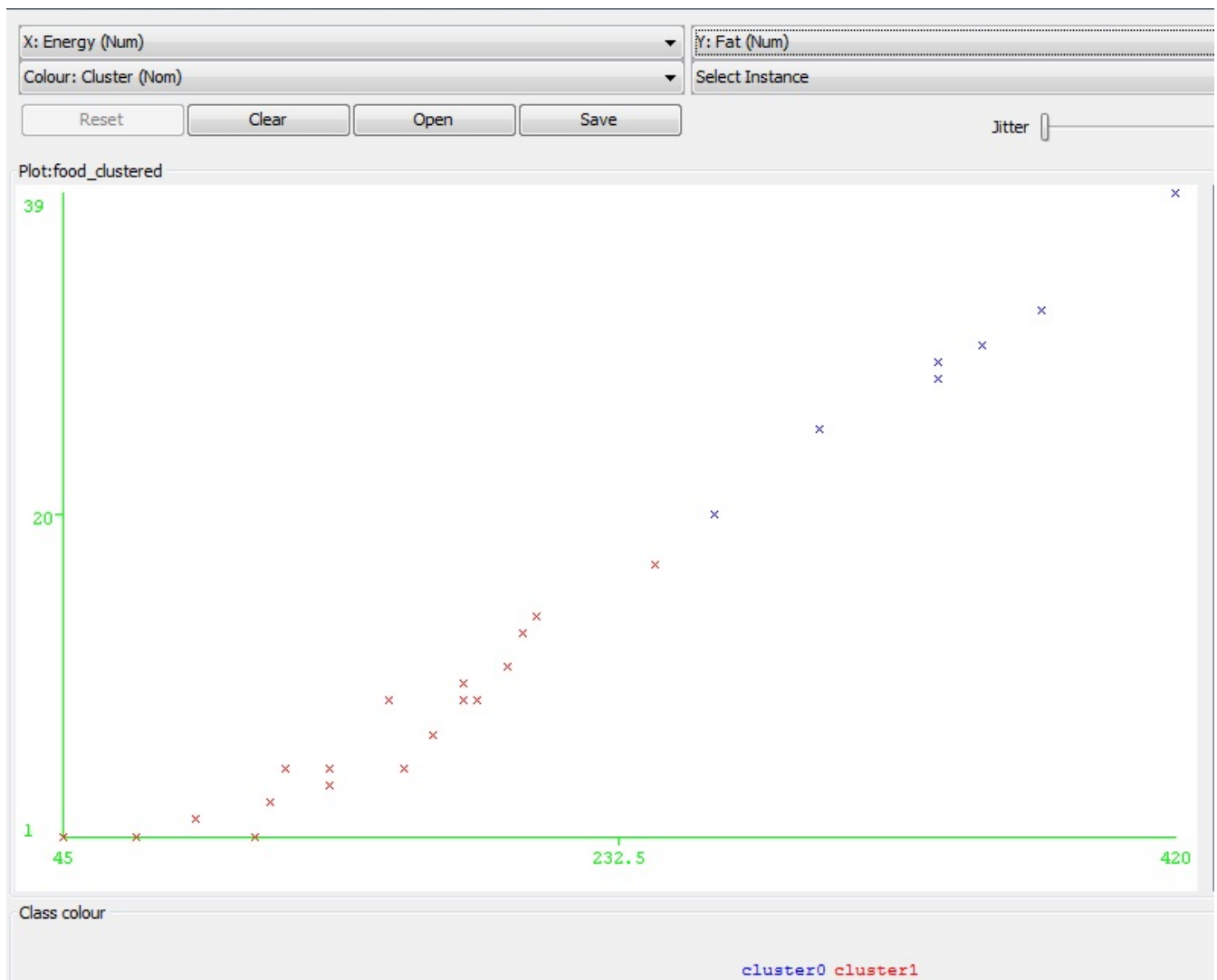
Number of iterations: 2

Within cluster sum of squared errors: 0.8481897660818714

Missing values globally replaced with mean/mode

Cluster centroids:

Attribute	Full Data (27)	Cluster#	
		0 (8)	1 (19)
Energy	207.4074	341.875	150.7895
Fat	13.4815	28.875	7



- Choosing more attributes for clustering will yield a worse separation and meaningless clustering, as we can see with choosing all the 5 attributes where we get the highest within cluster sum of squared errors (worst separation)  
Also we can see that some attributes like Protein and Iron have their centroids are exactly the same or close to each other.



```

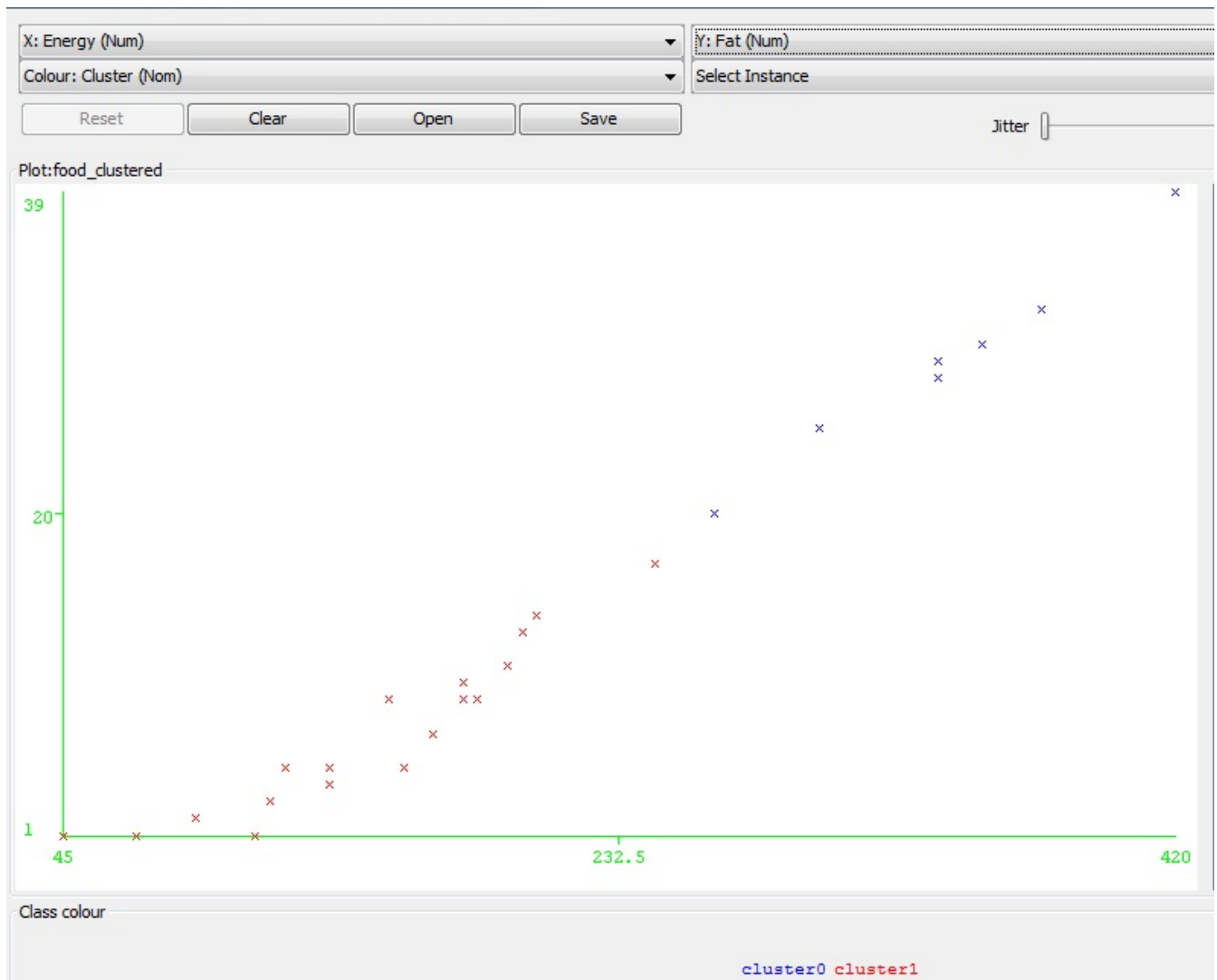
Number of iterations: 2
Within cluster sum of squared errors: 5.069321339929419
Missing values globally replaced with mean/mode

Cluster centroids:

```

Attribute	Full Data (27)	Cluster#	
		0 (9)	1 (18)
Energy	207.4074	331.1111	145.5556
Protein	19	19	19
Fat	13.4815	27.5556	6.4444
Calcium	43.963	8.7778	61.5556
Iron	2.3815	2.4667	2.3389

- The name attribute is ignored because it is categorical and K means clustering is based on numerical euclidean distance
2. Experiment with at least two different numbers of clusters, e.g. 2 and 5, but with the same seed value 10.
- Increasing the number of clusters will result in a better separation until we reach a cluster per single point. 
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.
- Changing the seed value will change the starting guess for the centroid and it mainly affects the number of iterations required to converge and to a lesser extent the final result after convergence 
4. Do you think the clusters are “good” clusters? (Are all of its members “similar” to each other? Are members from different clusters dissimilar?)
- Depending on the size of the clusters, the smaller the sizes the more similar its members but we need to define a sweet spot for dissimilarity.
5. What does each cluster represent? Choose one of the results. Make up labels (words or phrases in English) which characterize each cluster.
- As explained in the first point, the cluster representation depends on which attributes being selected, the more attributes selected the more meaningless the clusters become.
  - For Example clustering the Energy-Fat into two clusters will result one cluster with high-energy high-fat foods and the other cluster with low-energy low-fat foods.



## MakeDensityBasedClusters

Now with MakeDensityBasedClusters, SimpleKMeans is turned into a density-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).
  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)
- Working on the same clustering from point 1.5 and using different standard deviations 1 and 1000 we can see that increasing sd will increase the cluster influence to a wider range of points and the cluster with a higher prior probability will have more instances.
  - As we can see below cluster-1 have all the instances when we increased the standard deviation to 1000 since it has the higher prior.

sd= 1

```
Fitted estimators (with ML estimates of variance):

Cluster: 0 Prior probability: 0.3103

Attribute: Energy
Normal Distribution. Mean = 341.875 StdDev = 43.3689
Attribute: Fat
Normal Distribution. Mean = 28.875 StdDev = 5.1097

Cluster: 1 Prior probability: 0.6897

Attribute: Energy
Normal Distribution. Mean = 150.7895 StdDev = 49.0505
Attribute: Fat
Normal Distribution. Mean = 7 StdDev = 4.5422

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

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

Clustered Instances

0      8 ( 30%)
1     19 ( 70%)

Log likelihood: -8.84267
```

sd= 1000

Fitted estimators (with ML estimates of variance):

Cluster: 0 Prior probability: 0.3103

Attribute: Energy

Normal Distribution. Mean = 341.875 StdDev = 1000

Attribute: Fat

Normal Distribution. Mean = 28.875 StdDev = 1000

Cluster: 1 Prior probability: 0.6897

Attribute: Energy

Normal Distribution. Mean = 150.7895 StdDev = 1000

Attribute: Fat

Normal Distribution. Mean = 7 StdDev = 1000

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

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

Clustered Instances

1      27 (100%)

Log likelihood: -15.6623