732A61 Data Mining

Lab 1 by Emil Klasson Svensson

Simple K Means

1. Choose a set of attributes for clustering and give a motivation

I choose to remove the Names and Calcium variables. The Names shouldn't be included since it has no values and only serves as an index for the data. The Calcium I removed since the range it varies on is so small except for some outliers and it doesn't seem like any good separation would be achievable. The same for protein, there are three observations that are outside the point-spread but since we can't choose where the initialization is supposed to be.

2. Experiment with two different number of K and compare

K	Seed	Within Cluster SSE	Itterations
2	10	2.68675	2
3	10	1.53962	4
5	10	0.72018	8

The within cluster SSE becomes lower as we increase the number of clusters, this is because the area that the cluster covers are smaller and therefor becomes smaller. Also with more cluster, more and more iterations are needed for the algorithm to find stable means.

3. Experiment with different seed-values, what does the seed value do?

K	Seed	Within Cluster SSE	Itterations
2	15	2.68241	4
2	5	2.68675	2
2	1	2.68241	2

The seed controls the random initialization of the centroids so that the same result is given when you run it with the same seed value at different times.

4. *Clustering*

Here are some results from the clustering. A general note during the different clusters is that none of the variables form spherical shapes and more or less seems like non-convex shapes. These shapes are hard for the K-means algorithm to handle. Here is a result from the K-means with K = 2.

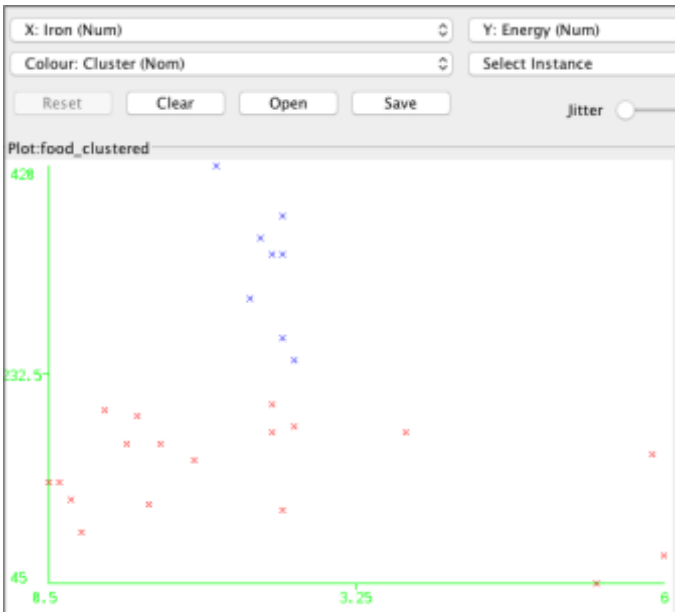
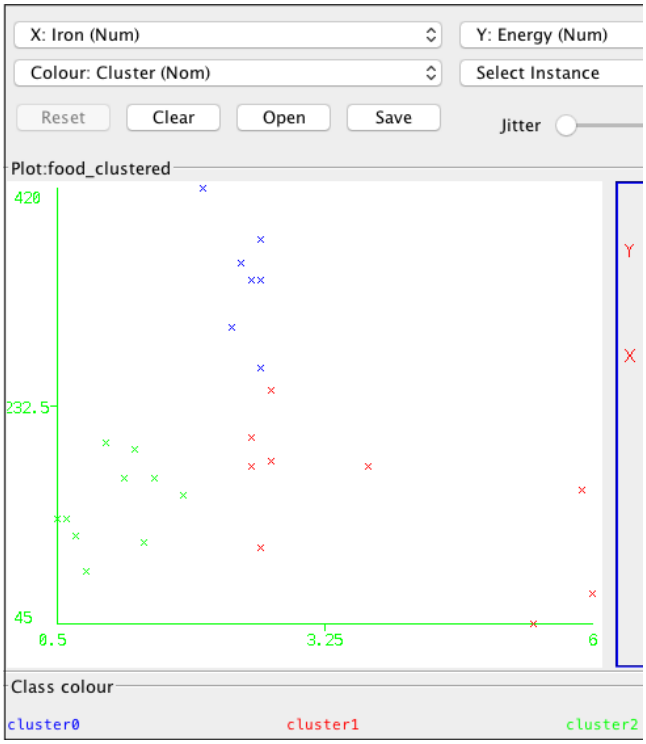
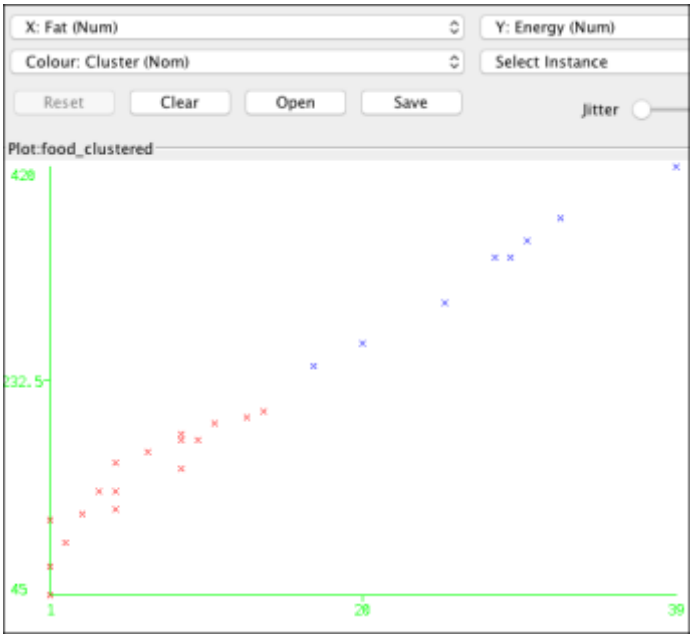


Figure 1 K-means with K = 2

The two clusters have a boundary that separates them at the Y-intercept of 232.5. The red cluster has been left with the three outliers in the Iron. Below is a result from K = 3.



The K = 3 didn't manage to make a cluster of the three outliers to the right and instead split the before (K = 2) red cluster in to two instead. With the right initialization the desired clustering it might be achievable.



Between the Energy and Fat there seems like it is a strong correlation and the cluster have divided them in two in a clear way around 232.5 units on energy and 20 units at fat.

5. *Name the clusters*

The K = 2 clustering seems more desirable since an increase of cluster don't improve or identify anything interesting. Returned clusters should be called something like a high energy/fat cluster and a low energy/fat cluster.

Density based clustering

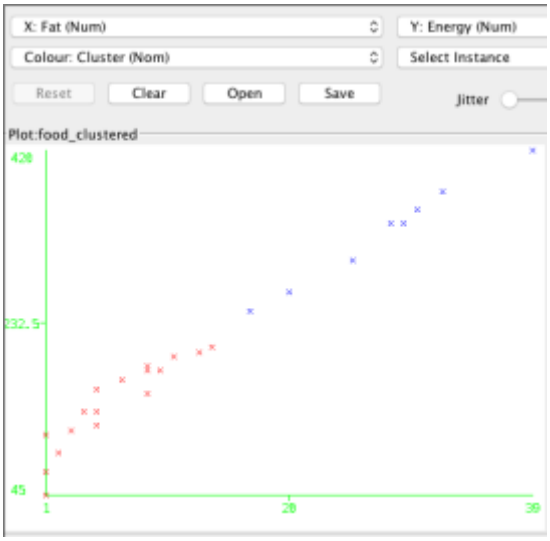
1. *Use the SimpleKMeans clusterer which gave the result you haven chosen in 5)*

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

MakeDensityBasedClusterer:

Wrapped clusterer:
kMeans
======
```

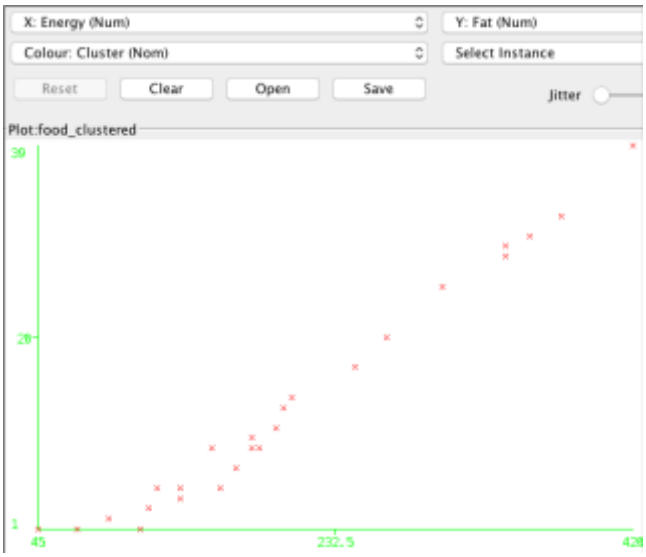
The MakeDensityBasedCluster uses a clustering technique of our choice and on top of this clustering calculates densities for the different clusters. In our case we wanted to use over K-means the print out above is just a comfirmation that we used k-means as the base that the MakeDensityBasedCluster function wraps around.

2. *With different standard-deviations*

Std = 0.001



With std = 1000



In the default-value the clustering is unchanged from the basic K-means clustering.
The more we increase the standard deviation the more the cluster with most observations (and higher density) takes over more and more until every observation is in one of the groups. The function tries to increase the minimum standard deviation within the clusters for values and the only way of doing that is to expand the cluster – this is why in the second graph with std = 1000 everything belongs to one cluster.

I tried with different values as well but I wanted to prove a point with this so I decided to go with a large value relative to the mean and a small value (the default value).