Lab 1: Clustering

Tore Andersson, Zahra Jalil Pour

First the data was explored by examining the correlation matrix.

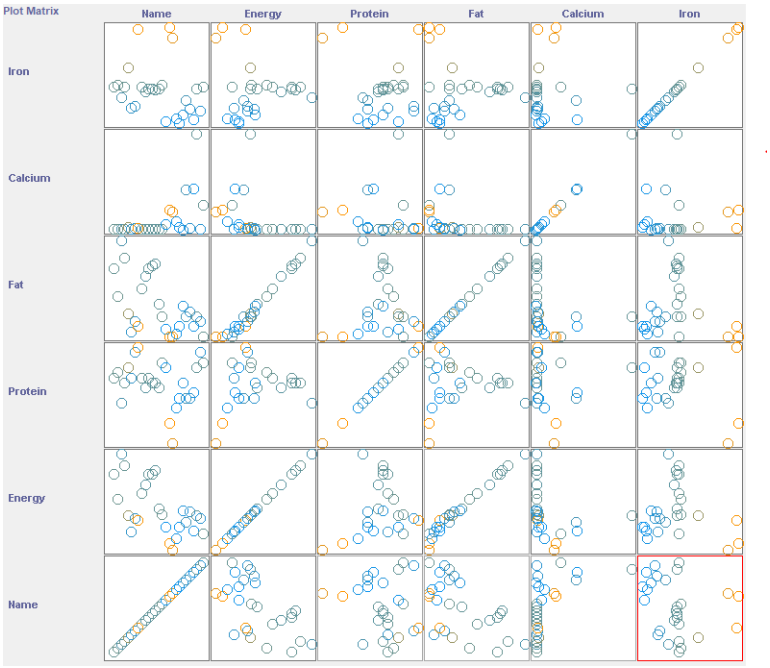

Figure 1: Correlation matrix of the data

From the correlation matrix in figure 1 it shows that there is a high positive correlation between the attributes fat and energy. For the other attributes there are some correlation, with protein and energy having positive correlation.

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?)
Weka uses Euclidian distance for K Means clustering, and the concept of K Means is based on distances between points, and based on these distances, the algorithm will find different clusters. Hence it is important to weigh the value of attributes. We can exclude the attributes that will not have any effect on the clusters. The attribute "name" needs to be ignored as the K-Nearest Neighbour algorithm only works for numerical data. It requires the data points to have a coordinate in the feature space.

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

For the algorithm simpleKMeans all models were both trained and evaluated on the training set. For the tested models all attributes with the exception of "names" were included. Four different klusters(k=2,3,4,5) with different seeds were used seed 1,2,10,100,500,1000,2000 in WEKA. The seed determines the "randomization" of the selected K which decides how the clusters are initialized which can lead to entirely different outcomes.

SimpleKMeans algorithm:

- 1. Initizalize a random k in the data
- 2. Use k as centroid and assign those values with the closes distance to a specific centroid as a cluster.
- 3. In the cluster calculate new centroid and assign those values to the new cluster.
- 4. repeat step 2 and 3 until there are no new changes in the clusters.

kSimpleMeans k = 2

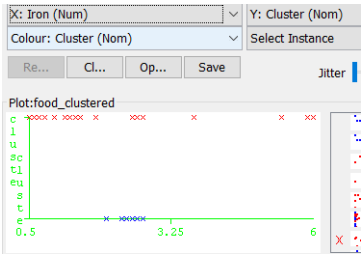

Figure 2: kSimpleMeans k = 2, iron, seed= 10

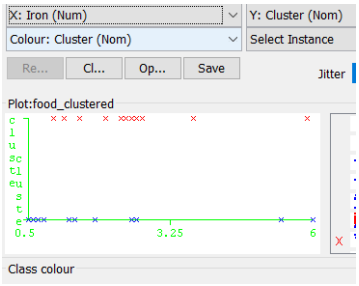

Figure 3: kSimpleMeans, k = 2, iron, seed = 1

For the attributes "Iron" there is some overlap and there is a clear difference that using different seeds as initialization had an impact. The clustering from seed 1 in figure 3 is more spread out than the clustering from seed 10 in figure 2. In figure 3 there is more overlap between the clusters which clustered data points with similar values to different clusters. Which indicates a worse clustering outcome.

In figure 4 and 5 the there is some overlapping clustering for the attribute iron. Where in figure 4 there is more similar clustering for seed 10 than there is in figure 5.

Figure 6 and 7 shows the clustering for the attribute energy which is more separated into two distinct clusters clearly shown in figure 6. This is the best case

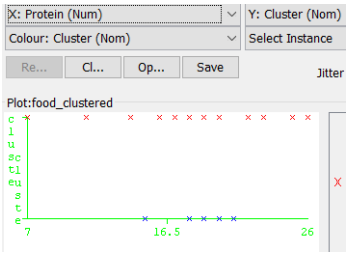

Figure 4: kSimpleMeans k = 2, protein, seed= 10

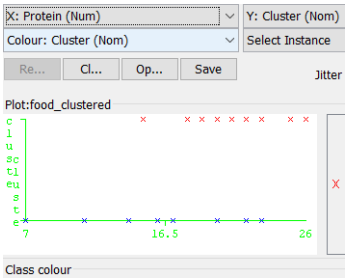

Figure 5: kSimpleMeans, k = 2, protein, seed = 1

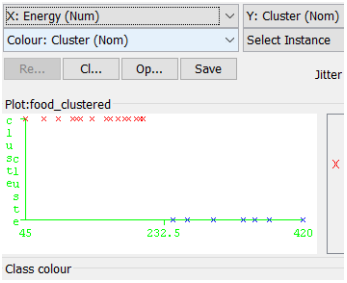

Figure 6: kSimpleMeans k = 2, energy, seed= 10

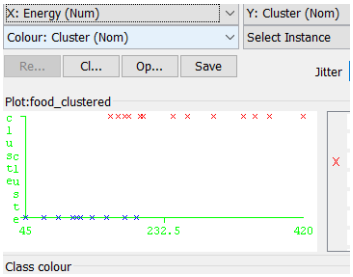

Figure 7: kSimpleMeans, k = 2, energy, seed = 1

scenario where similar values are linked to one cluster and are clearly separated. The cluster must not necessarily contain only similar values for it to be a good cluster but it's preferred for there to be a low amount of overlap between clusters.

In general the seed 10 which is shown in the left hand plots performed better compared to seed 1. This is supported in the output where seed 10 had a lower within cluster SSE than that of seed 1.

Output:

- seed 10 Within cluster sum of squared errors: 5.07
- Clustering distribution seed 10: 0(blue): 9 (33%),1(red): 18 (67%)
- Iterations until convergence: 2
- seed 1 Within cluster sum of squared errors: 5.91
- Clustering distribution seed 1: 0(blue): 11 (41%), 1(red): 16 (59%)
- Iterations until convergence: 6

For kSimpleMeans k = 2 cluster 1 which is the red one could be given the name "low energy foods" and the blue cluster 0 could be given the name "high energy foods".

kSimpleMeans k = 5

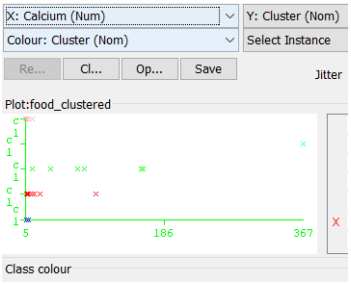

Figure 8: kSimpleMeans k = 5, calcium, seed= 10

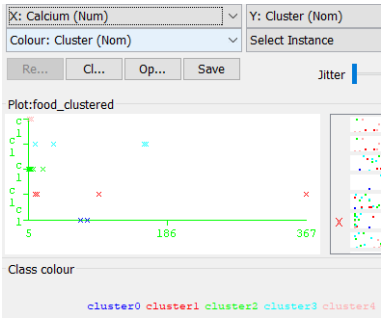

Figure 9: kSimpleMeans, k = 5, calcium, seed = 2

For k = 5, seeds 10 and 2 were used as seed 1 gave the exact same results as 10. k = 5 there is a greater amount of overlapping in between the clusters which is shown in figure 8 and 9 for the attribute calcium.In figure 8 there is a clear outlier for the attribute calcium which has a value of 367 compare to the other data points which has values between approximately (5-180). This outlier is given it's own cluster. This clear overlapping for similar values for different clusters shows that 5 clusters might be to much for this data set.

Figure 10: kSimpleMeans, k = 5, protein, seed= 10

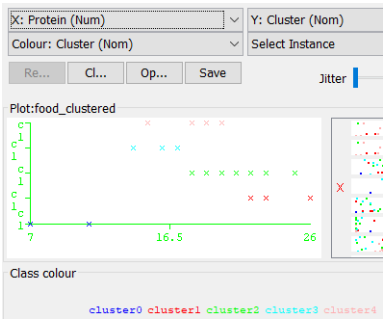

Figure 11: kSimpleMeans, k = 5, protein, seed = 2

For the attribute protein there is less overlapping for similar values but still a large amount of overlapping.

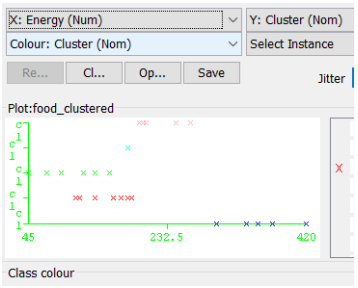

Figure 12: kSimpleMeans k = 5, energy, seed= 10

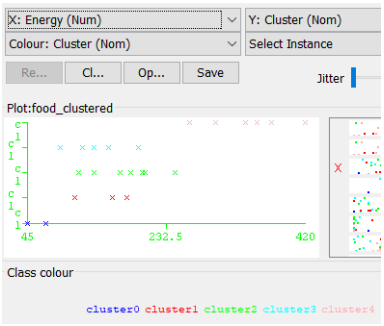

Figure 13: kSimpleMeans, k = 5, energy, seed = 2

Compared to k = 2 models the clusters are more overlapping and there is a less distiction to be made, following the a similar naming scheme the clusters could be named from lower to higher energy contentaints from figure 12 and seed 2. Ex: Cluster 0: Lowest energy, cluster 1: lower energy... cluster 4: highest energy.

Output:

- seed 10 cluster distribution:
  - 0: 7 (26%),1: 8 (30%), 2: 6 (22%),3: 1 (4%),4: 5 (19%)
- seed10: Number of iterations: 4
- seed 10: Within cluster sum of squared errors: 2.75
- seed 2 cluster distribution:
  - 0: 2 (7%), 1: 4 (15%), 2: 8 (30%), 3: 5 (19%), 4: 8 (30%)
- seed 2: Number of iterations: 6
- seed 2: Within cluster sum of squared errors: 2.11

The outputs shows that seed 2 provided a better clustering with lower within SSE.

In general it seems that the data set is better suited to be clustered into 2 rather than 5 clusters, there might be better clustering with either 3 or 4 clusters but this was not explored in this lab. The best explored clustering comes from k = 2 for seed 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.

Different seed values mean different initial cluster centers. In this task we use seed=1,2,10,100,500,1000 and 2000 for different k=2,3,4,5. In k=2 and k=3, we can not see obvious differences, but in k=4 and k=5,by seed=500, minimum instances in each cluster is two. It differs from seed=10. In seed=500, sum of squared error is lower compare to other values of seed. Here we can make several tries because we want to explore the best cluster solution based on the distance metric. The output of clustering by different values of seed and k is shown in the table.

seed//k	min instance	number of iteration	Sum of Square error
seed=1, k=2	11	6	5.9
seed=2, k=2	9	5	5.06
seed=10,k=2	9	2	5.06
seed=100,k=2	9	5	5.06
seed=500,k=2	11	4	5.09
seed=1000,k=2	3	2	4.45
seed=2000,k=2	9	2	5.06
seed=1,k=3	1	2	4.27
seed=2,k=3	5	6	3.60
seed=10,k=3	7	3	4.07
seed=100,k=3	1	5	4.27
seed=500,k=3	2	7	3.43
seed=1000,k=3	2	6	3.43
seed=2000,k=3	1	2	4.27
seed=1,k=4	1	5	3.83
seed=2,k=4	5	6	3.63
seed=10,k=4	1	3	3.22
seed=100,k=4	1	3	3.22
seed=500,k=4	2	6	2.61
seed=1000,k=4	2	5	2.57
seed=2000,k=4	1	6	3.22
seed=1,k=5	1	4	2.75
seed=2,k=5	2	6	2.11
seed=10,k=5	1	4	2.75
seed=100,k=5	1	4	2.03
seed=500,k=5	2	4	1.86
seed=1000,k=5	2	5	1.87
seed=2000,k=5	1	4	2.53

By selecting the random seed=500, number of clusters=5, we will have acceptable clustering.

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

By selecting Seed=500, k=5 we will have good clusters. Although two clusters are mixed with each other but separated by other clusters. Number of iterations and sum of squared error is the lowest among the explored models.

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

When k=5, seed=500 we have two clusters which are mixed to each other (cluster2 and 4) but are separated by other clusters.

- Cluster1, consist of food by highest calcium which have higher Energy, low Iron
- Cluster2, consist of food by highest protein, which have high Energy and high Fat
- Cluster3, consist of food by highest iron, which have minimum Energy, minimum Fat and minimum Protein.
- Cluster4, is a mix of food by high Protein, high fat with minimum Iron
- Cluster5, consist of food by highest fat which have highest Energy

```
kMeans
======

Number of iterations: 4
Within cluster sum of squared errors: 1.8668266858930311
Missing values globally replaced with mean/mode

Cluster centroids:
                        Cluster#
Attribute    Full Data          0          1          2          3          4
                  (27)        (3)        (4)        (2)        (9)        (9)
================================================================================
Energy         207.4074   151.6667     158.75       57.5   157.2222   331.1111
Protein              19    18.3333       23.5          9    19.4444         19
Fat             13.4815     7.6667       6.25          1     7.3333    27.5556
Calcium          43.963   227.6667       34.5         78    14.5556     8.7778
Iron             2.3815     1.6667      3.725        5.7        1.2     2.4667



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

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

Clustered Instances

0       3 ( 11%)
1       4 ( 15%)
2       2 (  7%)
3       9 ( 33%)
4       9 ( 33%)
```

Figure 14: kSimpleMeans k = 5, seed= 500

MakeDensityBasedClusters

Now with Make Density Based Clusters, Simple KMeans 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 Simple K Means clusterer which gave the result you haven chosen in 5).
Here we use k=5, seed=500 for clustering.

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)

In this algorithm, at first the k-means clustering by defined parameters will be formed. Then different standard deviation will be adjusted. Changing the parameter minStdDev in WEKA changes the lowest value that the standard deviation (sd) that can be taken for a single attribute. If the sd for a attribute is lower than the minimum input then the calculations for the density based clusters will use the specified input value instead, for all other std will remain the same.Here we set the sd with different values. As we see by increasing the value of sd, the number of clustering and Log Likelihood is decreasing. The standard deviation effects the size of final clustering. At first the value of sd is small and it doesn't change the number of clusters that we adjusted at first. But by increasing the value of sd, the number of clustering decreases. Standard deviation defines the width of the normal distribution. It represents the distance between the observations and the mean. Hence high value of SD indicates that the data is spread out over a large range of values.