732A75 Advanced Data Mining laboratory 1 report

Yuki Washio and Nicolas Taba

March 4, 2021

1 Introduction

The aim of this laboratory exercise is to gain familiarity with the data mining toolkit Weka by using some of its clustering algorithms and analyzing their output.

In this laboratory, we use the HARTIGAN (file06) dataset. This dataset features 27 kinds of food and informs us of the energy, protein, fat,c alcium and iron content per 3 ounce of portion.

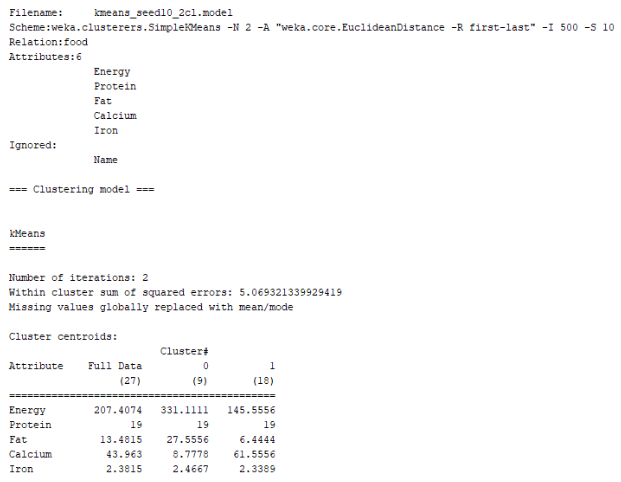
2 Clustering using the Kmeans algorithm

In this section of the laboratory, we use the Kmeans algortithm to cluster the data. We choose to ignore the "name" label because the algortihm does not interpret names as words but as strings. The Kmeans algorithm would try to measure distances between strings which is not a good metric. However, we notice that all of the food are meat and seafood based.

We experiment with the clustering algorithm by trying 2 different number of cluster (2 and 5) starting with the same initial cluster center (seed 10).

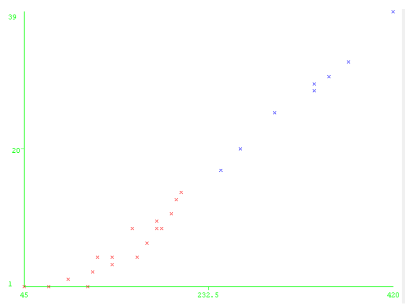
2.1 2 clusters with seed 10

The clustering algorithm separates into two clusters with 9 and 18 elements respectively in each Fig. 1. We observe a linear relationship between fat as a function of energy Fig. 2a. We seem to be able to separate the clusters into High energy, high fat the second cluster having low values for those labels. The clusters seem to be made up of similar elements that discriminate mostly along the energy values as we can see in the graphs that depict Protein, calcium and iron as a function of energy respectively.

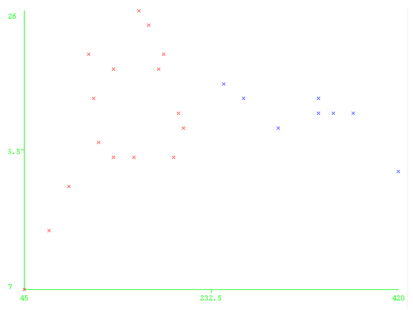


```
Filename:     kmeans_seed10_2cl.model
Scheme:weka.clusterers.SimpleKMeans -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10
Relation:food
Attributes:6
              Energy
              Protein
              Fat
              Calcium
              Iron
Ignored:
              Name

=== Clustering model ===


kMeans
======

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

Cluster centroids:
                         Cluster#
Attribute    Full Data          0          1
                  (27)        (9)       (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
```

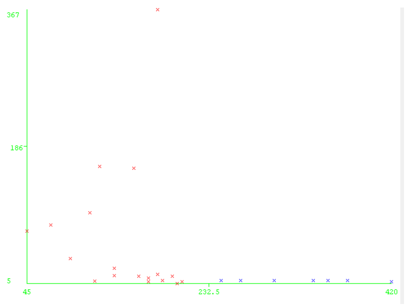
Figure 1: Kmeans algorithm output from Weka for 2 clusters, using seed 10.



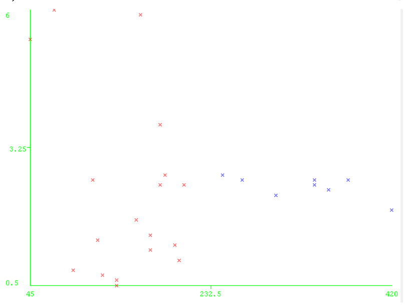
(a) fat as a function of energy



(b) protein as a function of energy



(c) Calcium as a function of energy



(d) Iron as a function of energy

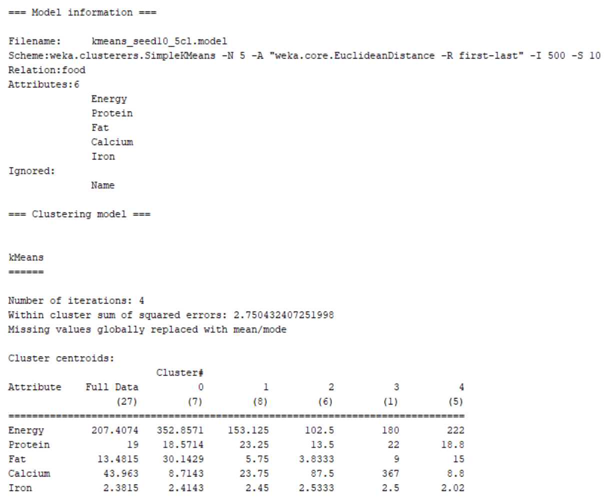
Figure 2: Cluster plot for Kmeans for 2 clusters

It is difficult to assign proper names to these clusters from the data shown in the cluster plots. However, we know that the data itself is of seafood and meats and we observed a linear relationship between the fat and energy parameters. We know that red meats contain more fat than seafood. One would push forwar that we could separate cluster-wise according to this principle. Further investigation of the dataset is needed.

2.2 5 clusters with seed 10

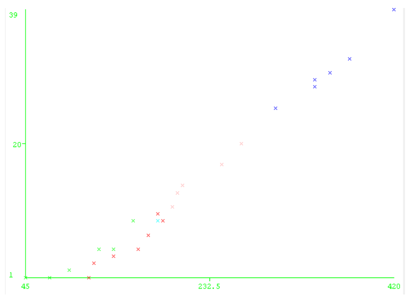
The clusters include all the points, but some of the elements of the same clusters are very dissimilar. In Fig. 4d we can see that the elements of the green cluster are dissimilar to each other when observing iron with respect to energy. This intra-cluster dissimilarity is observed for other variables. These are bad clusters as they have high dissimilarity between elements compared to the previous solution with 2 clusters. We can conclude that one must be careful when choosing the appropriate number of clusters in order for this algorithm to work.

A special attention should be given to the type of data that is being analyzed. As noted previously, we know from the name of the data that it can be separated by red meats and seafood. There might be other ways of separating the data differently, but it is not immediately evident what 5 clusters could be made from the data.

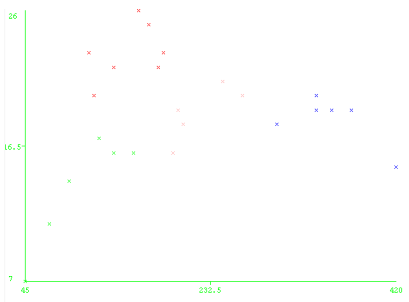


```
=== Model information ===

Filename:     kmeans_seed10_5cl.model
Scheme:weka.clusterers.SimpleKMeans -N 5 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10
Relation:food
Attributes:6
              Energy
              Protein
              Fat
              Calcium
              Iron
Ignored:
              Name

=== Clustering model ===


kMeans
======

Number of iterations: 4
Within cluster sum of squared errors: 2.750432407251998
Missing values globally replaced with mean/mode

Cluster centroids:
                         Cluster#
Attribute    Full Data          0          1          2          3          4
                  (27)        (7)        (8)        (6)        (1)        (5)
=============================================================================
Energy        207.4074   352.8571    153.125      102.5        180        222
Protein             19    18.5714      23.25       13.5         22       18.8
Fat            13.4815    30.1429       5.75     3.8333          9         15
Calcium         43.963     8.7143      23.75       87.5        367        8.8
Iron            2.3815     2.4143       2.45     2.5333        2.5       2.02
```

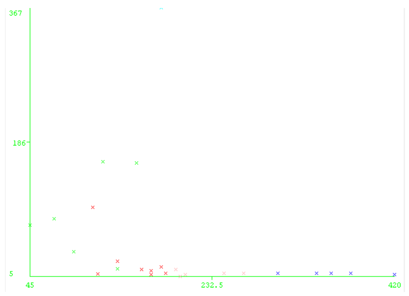
Figure 3: Kmeans algorithm output from Weka for 5 clusters, using seed 10.



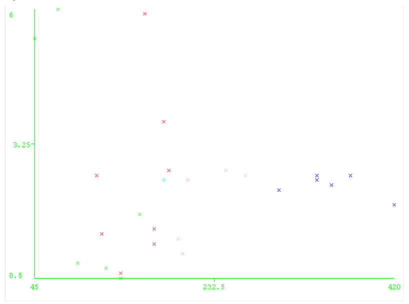
(a) fat as a function of energy



(b) protein as a function of energy



(c) Calcium as a function of energy



(d) Iron as a function of energy

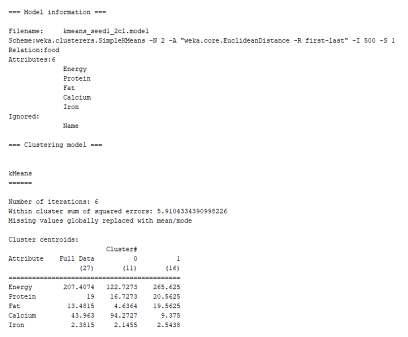
Figure 4: Cluster plot for Kmeans with 5 clusters

2.3 2 clusters with seed 1 and 123

We observe poor clustering when using seed 123, the elements within the clusters are dissimilar to one another as seen in Fig. 5c as attested by the larger sum of square error compared to the two other clustering solutions. We also notice that there are only two element in cluster 0.

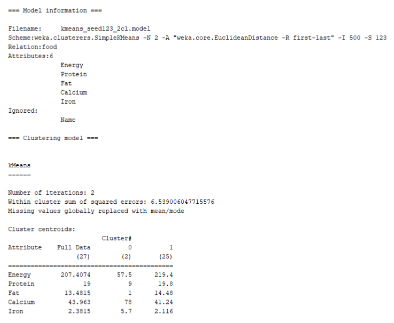
When using different seed, we obtain similar, plausible results. For example, seed 1 yields similar results than those obtained for seed 10. This reinforces our belief that the clustering algorithms yields the correct result given that the other parameters are fixed (number of clusters and data). Our conclusion that the food be separated into 2 categories is reinforced, however it becomes more difficult to decide which one of these clustering solutions is the correct one. The metric we choose to separate the better clustering is the sum of square error in this case. We will choose seed 10.

The seed value controls the generation of random numbers. This in turn decides where the starting centroid is chosen to be in the algorithm. Different seed number will yield different clusters. The seed is intended to be used for reproducibility of results by different users.



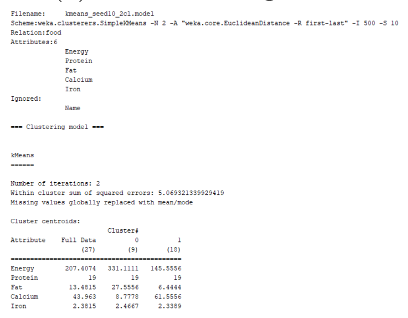
```
=== Model information ===

Filename:     kmeans_seed1_2cl.model
Scheme:weka.clusterers.SimpleKMeans -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 1
Relation:food
Attributes:6
              Energy
              Protein
              Fat
              Calcium
              Iron
Ignored:
              Name

=== Clustering model ===


kMeans
======

Number of iterations: 6
Within cluster sum of squared errors: 5.9104334390998226
Missing values globally replaced with mean/mode

Cluster centroids:
                         Cluster#
Attribute    Full Data          0          1
                  (27)       (11)       (16)
============================================
Energy        207.4074   122.7273    265.625
Protein             19    16.7273    20.5625
Fat            13.4815     4.6364    19.5625
Calcium         43.963    94.2727      9.375
Iron            2.3815     2.1455     2.5438
```

(a) Kmeans using seed 1



```
=== Model information ===

Filename:     kmeans_seed123_2cl.model
Scheme:weka.clusterers.SimpleKMeans -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 123
Relation:food
Attributes:6
              Energy
              Protein
              Fat
              Calcium
              Iron
Ignored:
              Name

=== Clustering model ===


kMeans
======

Number of iterations: 2
Within cluster sum of squared errors: 6.539006047715576
Missing values globally replaced with mean/mode

Cluster centroids:
                         Cluster#
Attribute    Full Data          0          1
                  (27)        (2)       (25)
============================================
Energy        207.4074       57.5      219.4
Protein             19          9       19.8
Fat            13.4815          1      14.48
Calcium         43.963         78      41.24
Iron            2.3815        5.7      2.116
```

(b) Kmeans using seed 123



```
Filename:     kmeans_seed10_2cl.model
Scheme:weka.clusterers.SimpleKMeans -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10
Relation:food
Attributes:6
              Energy
              Protein
              Fat
              Calcium
              Iron
Ignored:
              Name

=== Clustering model ===


kMeans
======

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

Cluster centroids:
                         Cluster#
Attribute    Full Data          0          1
                  (27)        (9)       (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
```

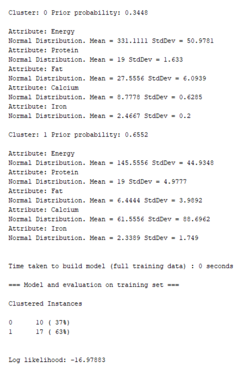
(c) Kmeans using seed 10

Figure 5: Kmeans algorithm for 2 clusters using different seed

3 Using density based clusters

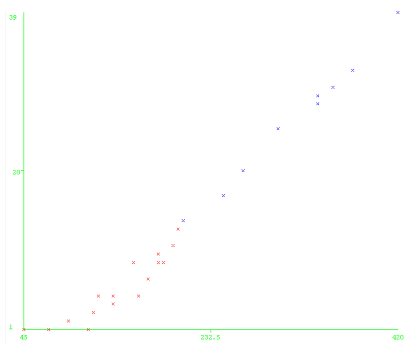
We use here density based clusters to cluster our data. This approach allows us to control how large we allow clusters to vary from their centroid by tuning the standard deviation parameter. We use the SimpleKMeans clusterer with $k = 2$ and seed 10. We will change use values of the standard deviation of $1.e^{-6}$ and 200.

We start with the density based clustering that uses the smallest standard deviation first.



```
Cluster: 0 Prior probability: 0.3448

Attribute: Energy
Normal Distribution. Mean = 331.1111 StdDev = 50.9781
Attribute: Protein
Normal Distribution. Mean = 19 StdDev = 1.633
Attribute: Fat
Normal Distribution. Mean = 27.5556 StdDev = 6.0939
Attribute: Calcium
Normal Distribution. Mean = 8.7778 StdDev = 0.6285
Attribute: Iron
Normal Distribution. Mean = 2.4667 StdDev = 0.2

Cluster: 1 Prior probability: 0.6552

Attribute: Energy
Normal Distribution. Mean = 145.5556 StdDev = 44.9348
Attribute: Protein
Normal Distribution. Mean = 19 StdDev = 4.9777
Attribute: Fat
Normal Distribution. Mean = 6.4444 StdDev = 3.9892
Attribute: Calcium
Normal Distribution. Mean = 61.5556 StdDev = 88.6962
Attribute: Iron
Normal Distribution. Mean = 2.3389 StdDev = 1.749


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

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

Clustered Instances

0      10 ( 37%)
1      17 ( 63%)


Log likelihood: -16.97883
```

(a) Density based clustering output

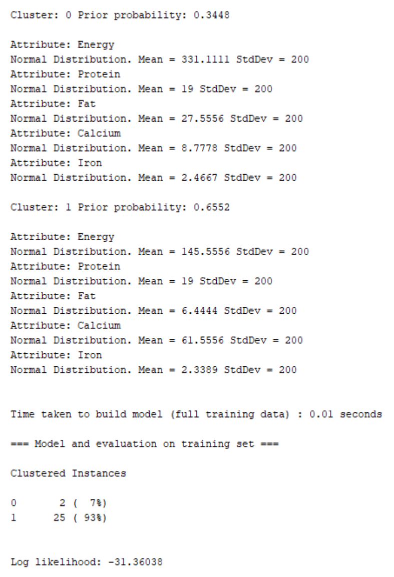


(b) Cluster plot of energy with respect to fat

Figure 6: Density based clustering with standard deviation of $1.e^{-6}$

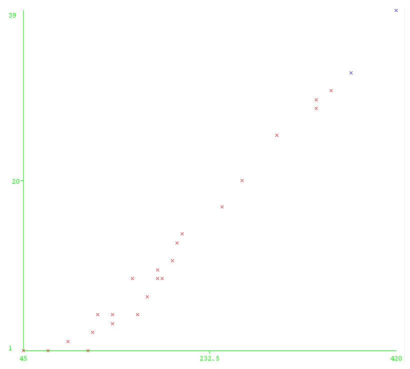
We obtain similar results to the ones we previously had using the kmeans approach. Furthermore, we can observe that the clustering by visualizing fat with respect to energy in Fig. 7b is similar to that obtained previously in Fig. 2a

We now increase the standard deviation to 200 while keeping all other arguments the same.



```
Cluster: 0 Prior probability: 0.3448

Attribute: Energy
Normal Distribution. Mean = 331.1111 StdDev = 200
Attribute: Protein
Normal Distribution. Mean = 19 StdDev = 200
Attribute: Fat
Normal Distribution. Mean = 27.5556 StdDev = 200
Attribute: Calcium
Normal Distribution. Mean = 8.7778 StdDev = 200
Attribute: Iron
Normal Distribution. Mean = 2.4667 StdDev = 200

Cluster: 1 Prior probability: 0.6552

Attribute: Energy
Normal Distribution. Mean = 145.5556 StdDev = 200
Attribute: Protein
Normal Distribution. Mean = 19 StdDev = 200
Attribute: Fat
Normal Distribution. Mean = 6.4444 StdDev = 200
Attribute: Calcium
Normal Distribution. Mean = 61.5556 StdDev = 200
Attribute: Iron
Normal Distribution. Mean = 2.3389 StdDev = 200


Time taken to build model (full training data) : 0.01 seconds

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

Clustered Instances

0       2 (  7%)
1      25 ( 93%)


Log likelihood: -31.36038
```

(a) Density based clustering output



(b) Cluster plot of energy with respect to fat

Figure 7: Density based clustering with standard deviation of 200

We observe from the data in Fig. 7 that the when the standard deviation is large enough, we have clusters with large dissimilarity returned to us by the algorithm. As pointed out earlier, a high standard deviation parameter will result in large amounts of data that are dissimilar to the centroid to be accepted as part of the cluster. This is due to the fact that with a high enough standard deviation, the density for elements that are far from the centroid is high enough to be accepted in the first cluster that is calculated.

In conclusion, we should pay attention to the value we assign to standard deviation as a value that is too high will yield poor quality clusters with dissimilar items within them.