732A75-Clustering Lab

Rabnawaz(rabsh696) & Saman Zahid(samza595)

2/18/2018

SimpleKMeans

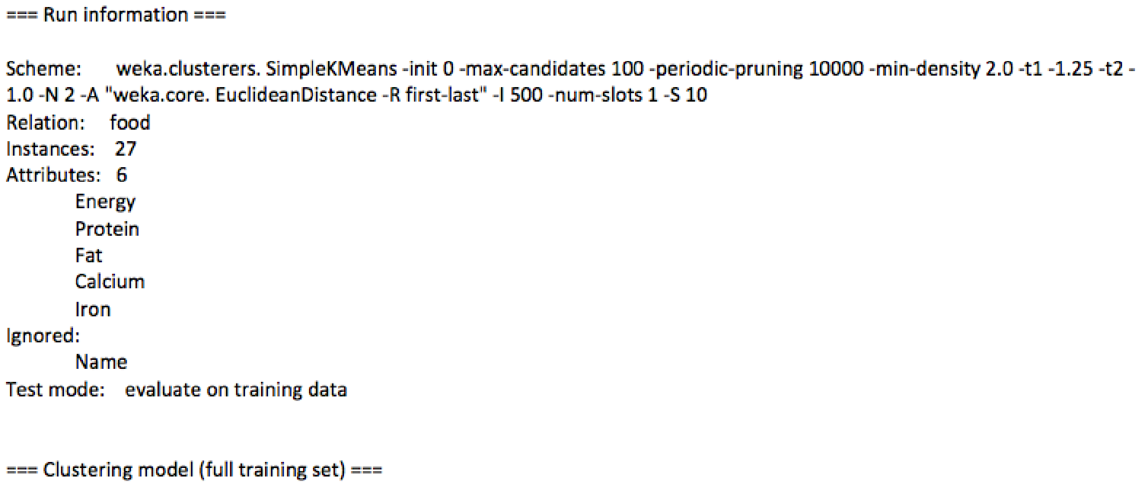
Note: using the feature of ignore attribute in weka before running KMeans

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

Name attribute must be ignored because `name` is a categorical variable while k-means algorithm work on continuous numerical values. All other attributes(Energy, Protein, Fat, Calcium, Iron) are continuous thus all other attributes can be selected for clustering.

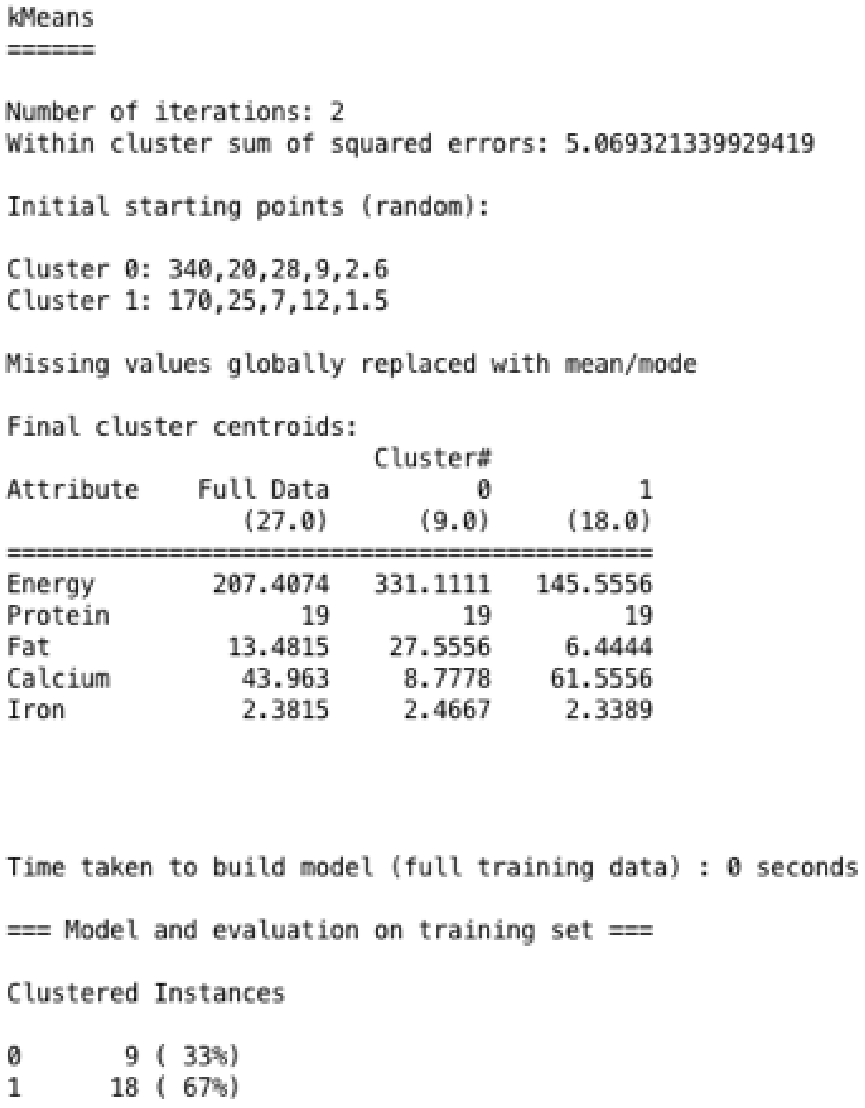
2. Experiment with at least two different numbers of clusters

Basic Information for the procedure



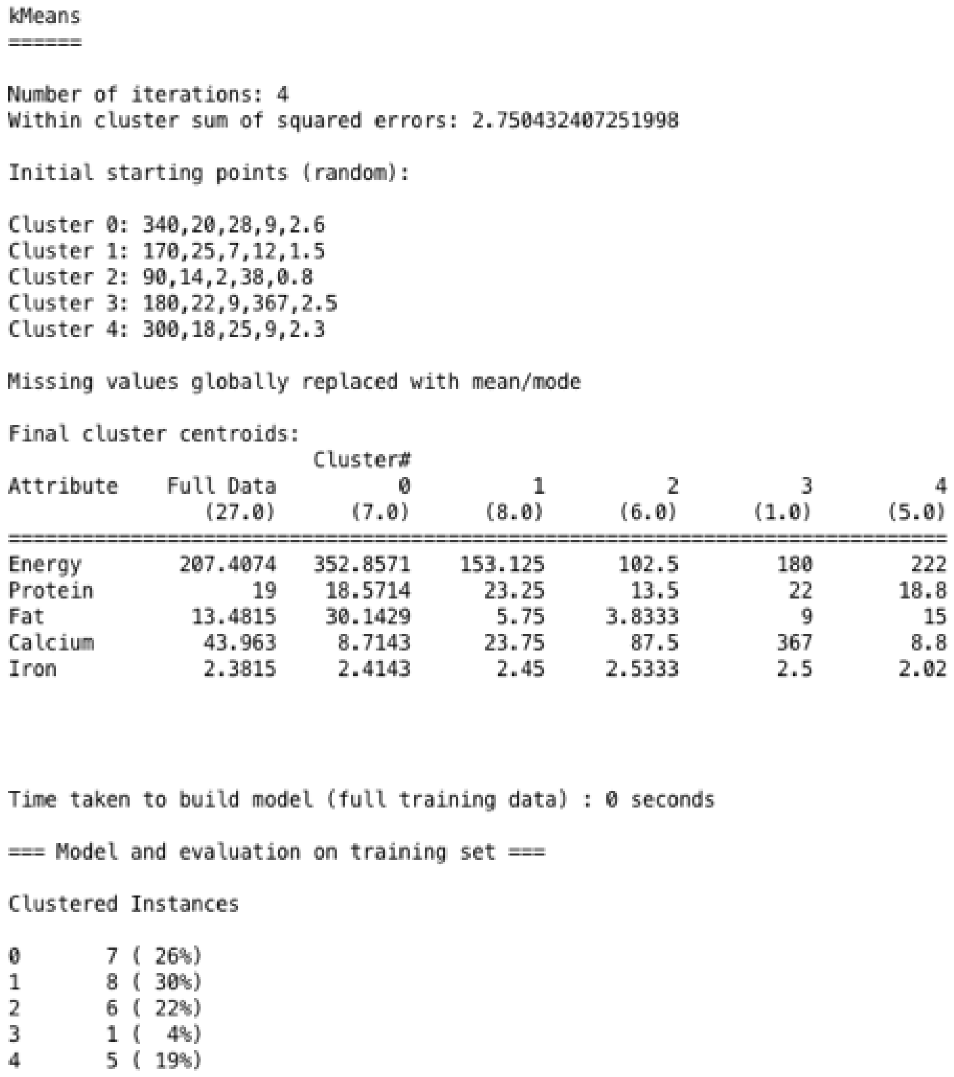
```
=== Run information ===

Scheme:      weka.clusterers. SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -
1.0 -N 2 -A "weka.core. EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10
Relation:    food
Instances:   27
Attributes:  6
             Energy
             Protein
             Fat
             Calcium
             Iron
Ignored:
             Name
Test mode:    evaluate on training data


=== Clustering model (full training set) ===
```

KMeans Method with 2 Clusters with seed value 10(rest of the setting remains default



```
kMeans
======

Number of iterations: 2
Within cluster sum of squared errors: 5.069321339929419

Initial starting points (random):

Cluster 0: 340,20,28,9,2.6
Cluster 1: 170,25,7,12,1.5

Missing values globally replaced with mean/mode

Final cluster centroids:
                          Cluster#
Attribute    Full Data          0          1
                (27.0)      (9.0)     (18.0)
============================================
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




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

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

Clustered Instances

0       9 ( 33%)
1      18 ( 67%)
```

KMeans Method with 5 Clusters with seed value 10(rest of the setting remains default)

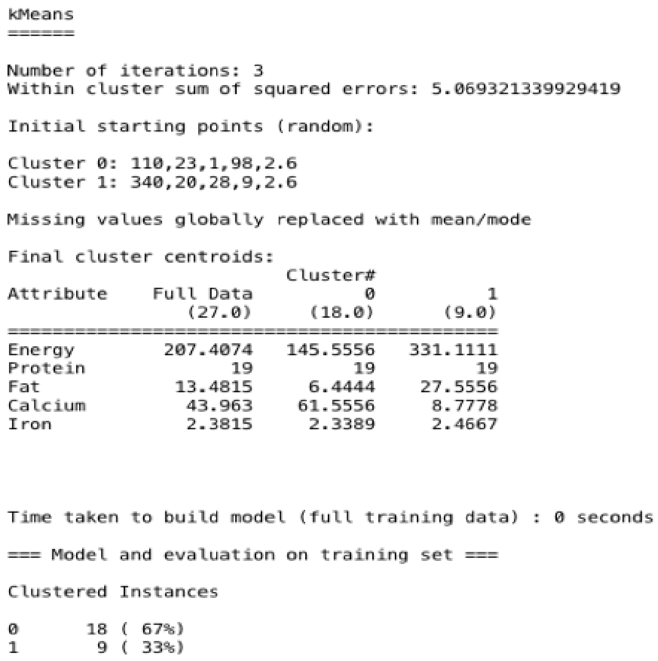


```
kMeans
======

Number of iterations: 4
Within cluster sum of squared errors: 2.750432407251998

Initial starting points (random):

Cluster 0: 340,20,28,9,2.6
Cluster 1: 170,25,7,12,1.5
Cluster 2: 90,14,2,38,0.8
Cluster 3: 180,22,9,367,2.5
Cluster 4: 300,18,25,9,2.3

Missing values globally replaced with mean/mode

Final cluster centroids:
                         Cluster#
Attribute    Full Data          0          1          2          3          4
                (27.0)      (7.0)      (8.0)      (6.0)      (1.0)      (5.0)
=============================================================================
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




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

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

Clustered Instances

0       7 ( 26%)
1       8 ( 30%)
2       6 ( 22%)
3       1 (  4%)
4       5 ( 19%)
```

3. Compare result with previous results. i.e. with different initial cluster centers.

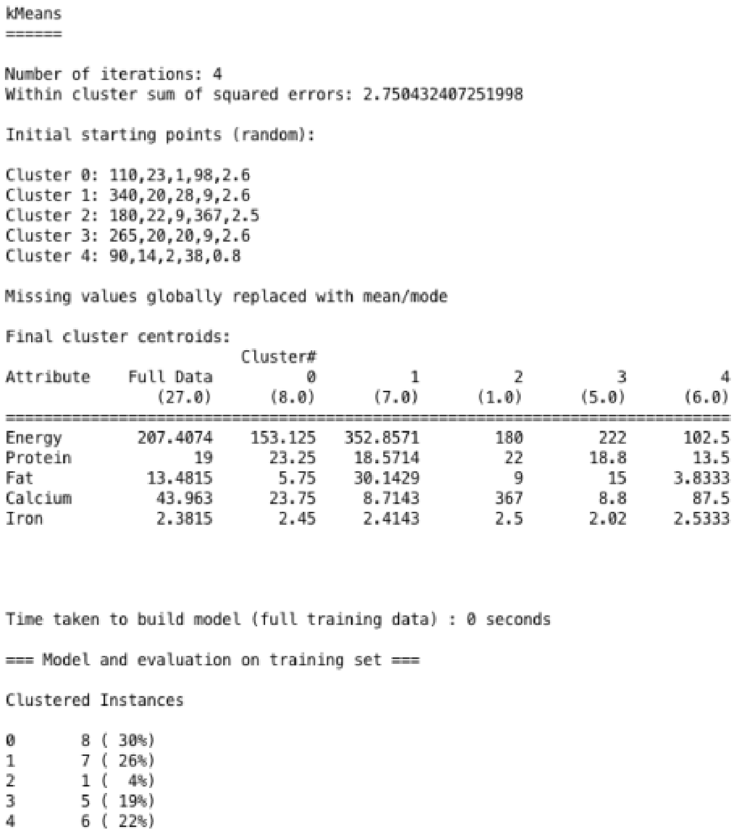
In part 3, by changing seed, the initial randomly chosen centroid value changes, due to which the entire clustering is changed. By changing the seed value from 10 to 5, the same clusters are formed but in different order in both cases (number of clusters = 2 or 5) with the same number of iterations and error rate. But by increasing the seed (for seed =15) the number of objects in each cluster, the formation of cluster (that is cluster centroid) instances changes a lot.

KMeans Method with 2 Clusters with seed value 5(rest of the setting remains default)



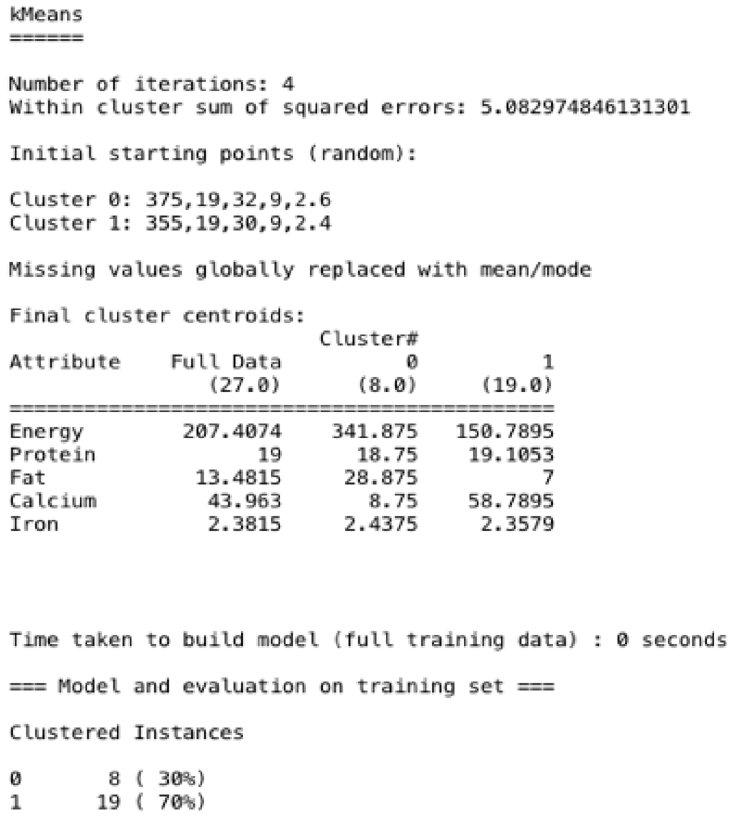
```
kMeans
======

Number of iterations: 3
Within cluster sum of squared errors: 5.069321339929419

Initial starting points (random):

Cluster 0: 110,23,1,98,2.6
Cluster 1: 340,20,28,9,2.6

Missing values globally replaced with mean/mode

Final cluster centroids:
                         Cluster#
Attribute    Full Data          0          1
                (27.0)     (18.0)      (9.0)
============================================
Energy        207.4074   145.5556   331.1111
Protein             19         19         19
Fat            13.4815     6.4444    27.5556
Calcium         43.963    61.5556     8.7778
Iron            2.3815     2.3389     2.4667



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

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

Clustered Instances

0      18 ( 67%)
1       9 ( 33%)
```

KMeans Method with 5 Clusters with seed value 5(rest of the setting remains default)



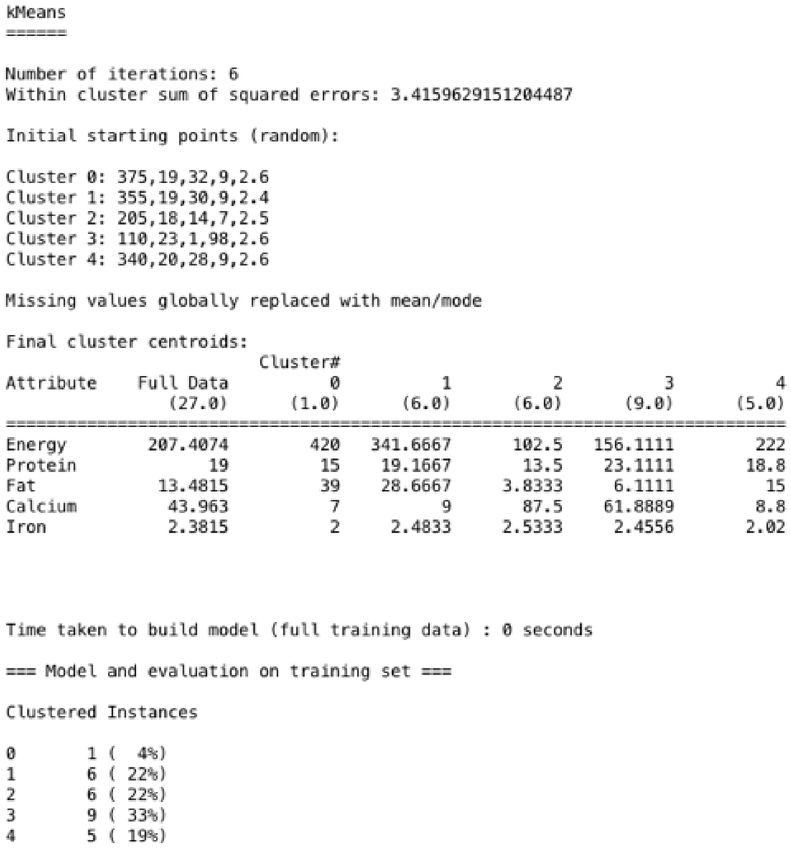
```
kMeans
======

Number of iterations: 4
Within cluster sum of squared errors: 2.750432407251998

Initial starting points (random):

Cluster 0: 110,23,1,98,2.6
Cluster 1: 340,20,28,9,2.6
Cluster 2: 180,22,9,367,2.5
Cluster 3: 265,20,20,9,2.6
Cluster 4: 90,14,2,38,0.8

Missing values globally replaced with mean/mode

Final cluster centroids:
                         Cluster#
Attribute    Full Data          0          1          2          3          4
                (27.0)      (8.0)      (7.0)      (1.0)      (5.0)      (6.0)
==============================================================================
Energy        207.4074    153.125   352.8571        180        222      102.5
Protein             19      23.25    18.5714         22       18.8       13.5
Fat            13.4815       5.75    30.1429          9         15     3.8333
Calcium         43.963      23.75     8.7143        367        8.8       87.5
Iron            2.3815       2.45     2.4143        2.5       2.02     2.5333




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

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

Clustered Instances

0       8 ( 30%)
1       7 ( 26%)
2       1 (  4%)
3       5 ( 19%)
4       6 ( 22%)
```

KMeans Method with 2 Clusters with seed value 15(rest of the setting remains default)



```
kMeans
======

Number of iterations: 4
Within cluster sum of squared errors: 5.082974846131301

Initial starting points (random):

Cluster 0: 375,19,32,9,2.6
Cluster 1: 355,19,30,9,2.4

Missing values globally replaced with mean/mode

Final cluster centroids:
                         Cluster#
Attribute    Full Data          0          1
                (27.0)      (8.0)     (19.0)
============================================
Energy        207.4074    341.875   150.7895
Protein             19      18.75    19.1053
Fat            13.4815     28.875          7
Calcium         43.963       8.75    58.7895
Iron            2.3815     2.4375     2.3579



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

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

Clustered Instances

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

KMeans Method with 5 Clusters with seed value 15(rest of the setting remains default)

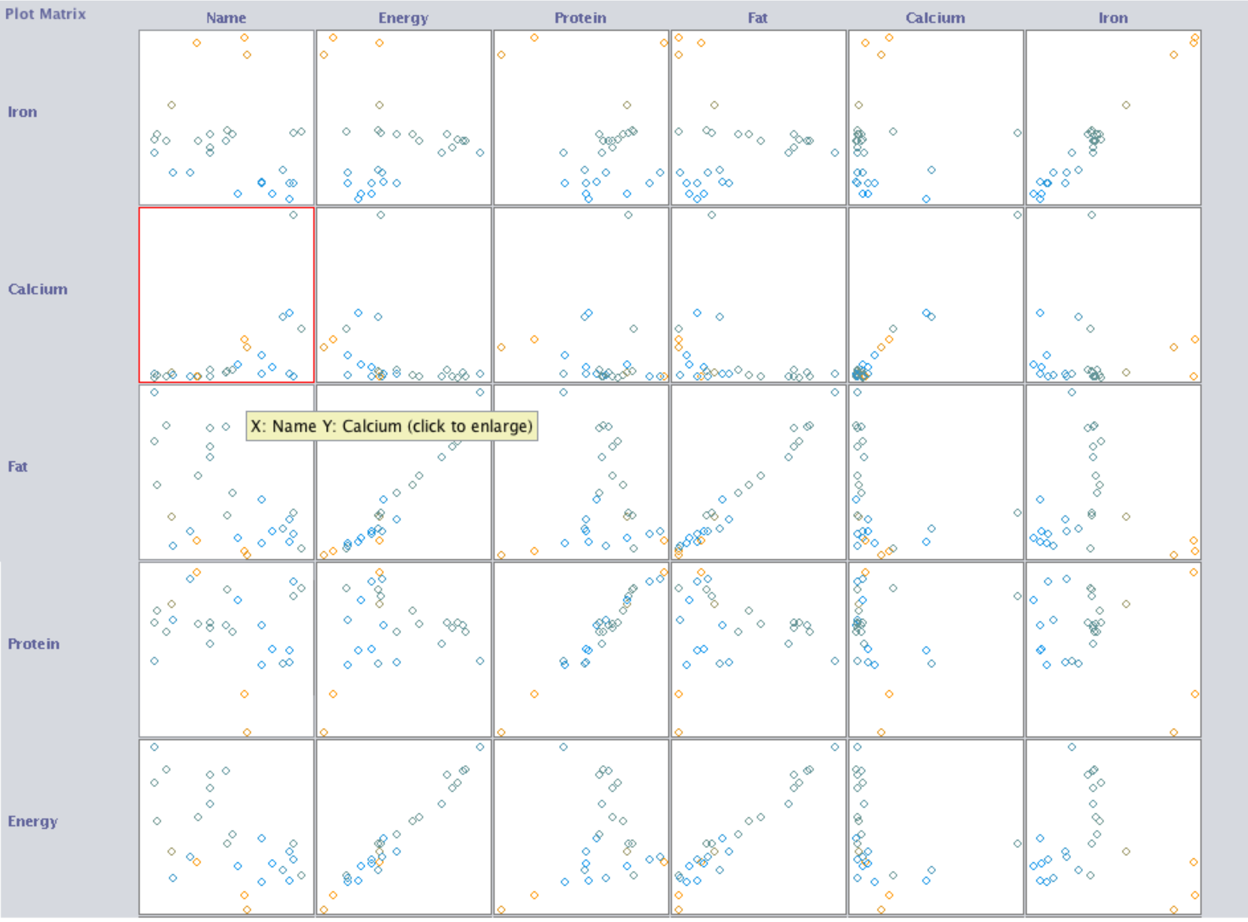


```
kMeans
======

Number of iterations: 6
Within cluster sum of squared errors: 3.4159629151204487

Initial starting points (random):

Cluster 0: 375,19,32,9,2.6
Cluster 1: 355,19,30,9,2.4
Cluster 2: 205,18,14,7,2.5
Cluster 3: 110,23,1,98,2.6
Cluster 4: 340,20,28,9,2.6

Missing values globally replaced with mean/mode

Final cluster centroids:
                         Cluster#
Attribute    Full Data          0          1          2          3          4
                (27.0)      (1.0)      (6.0)      (6.0)      (9.0)      (5.0)
==============================================================================
Energy        207.4074        420   341.6667      102.5   156.1111        222
Protein             19         15    19.1667       13.5    23.1111       18.8
Fat            13.4815         39    28.6667     3.8333     6.1111         15
Calcium         43.963          7          9       87.5    61.8889        8.8
Iron            2.3815          2     2.4833     2.5333     2.4556       2.02



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

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

Clustered Instances

0       1 (  4%)
1       6 ( 22%)
2       6 ( 22%)
3       9 ( 33%)
4       5 ( 19%)
```

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

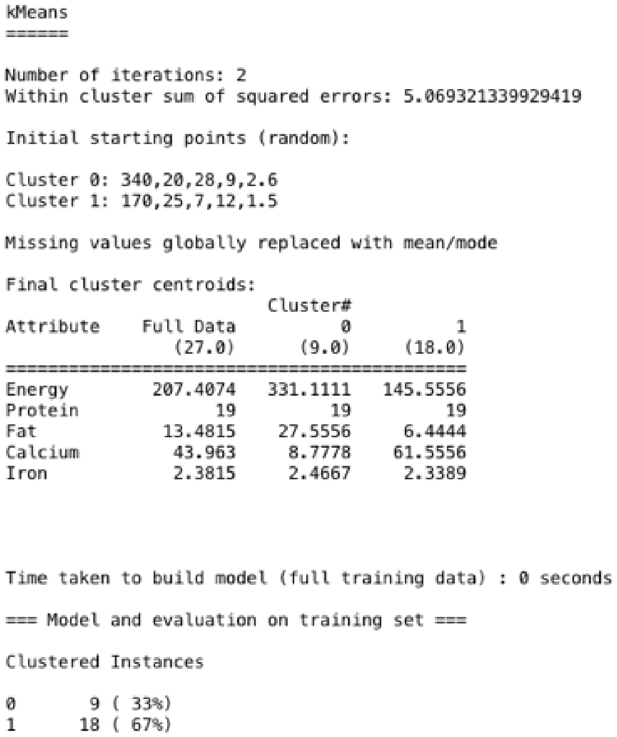
In my opinion the clusters formed are not good, it is because the objects are dispersed and by taking a random initial centroid, it might be possible that centroid is near the outlier which can result in misclassifying the dissimilar to be similar and put together in same cluster.

As from visualization, it can also be observed that the position of objects with very high and very low value differs. Due to very high variance, the possibility of #having outliers increases. and k-mean does not work well with outliers.



5. What does each cluster represent? Choose one of the results. Make up labels which characterize each cluster.

I choose the result with 2 clusters and seed = 10. It can be observed that for cluster 0, energy is very high, fat is high while calcium is low, while for cluster 1, energy is comparatively low, fat is low, and calcium is high. Protein and iron value for both clusters are almost same. So, on the basis of this observation i would name cluster 0 as “High fat low calcium” and cluster 1 as “low fat high calcium” cluster.



```
kMeans
======

Number of iterations: 2
Within cluster sum of squared errors: 5.069321339929419

Initial starting points (random):

Cluster 0: 340,20,28,9,2.6
Cluster 1: 170,25,7,12,1.5

Missing values globally replaced with mean/mode

Final cluster centroids:
                         Cluster#
Attribute    Full Data          0          1
                (27.0)      (9.0)     (18.0)
============================================
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




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

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

Clustered Instances

0       9 ( 33%)
1      18 ( 67%)
```

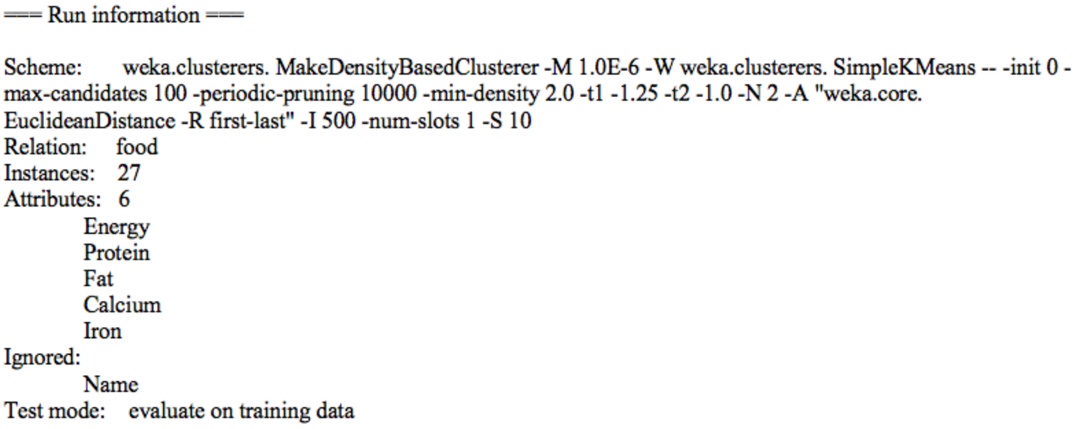
MakeDensityBasedClusters

1. Use the SimpleKMeans clusterer which gave the result you have chosen in 5.

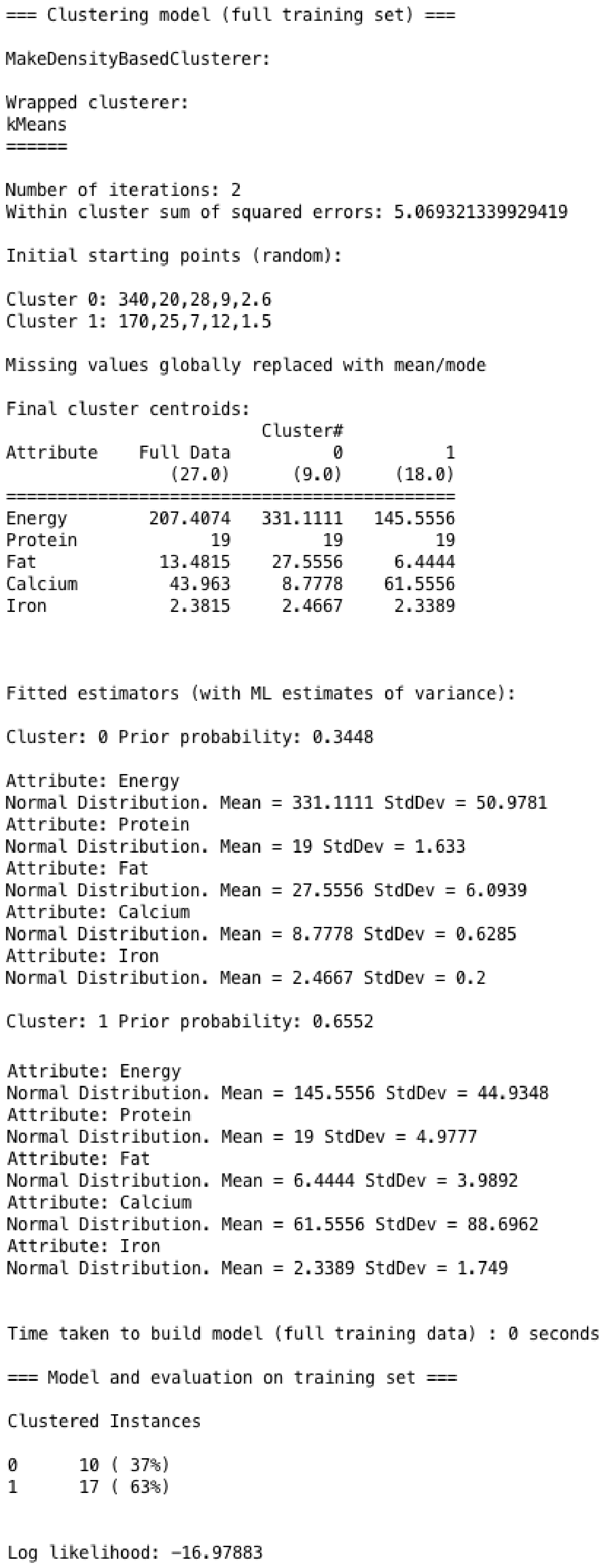
we run the density-based clustering algorithm with number of cluster that we have selected in question 1 part 5 which is 2 and seed value 10 with default standard deviation that is the minimum standard deviation $1 \;\; X \;\; 10^{-6}$

2. Experiment with at least two different standard deviations. Compare the results..

Part 2: By taking standard deviation first as min standard deviation $1 \;\; X \;\; 10^{-6}$, the cluster formed has a difference of 1 object as compared to what we got from k-means algorithm.

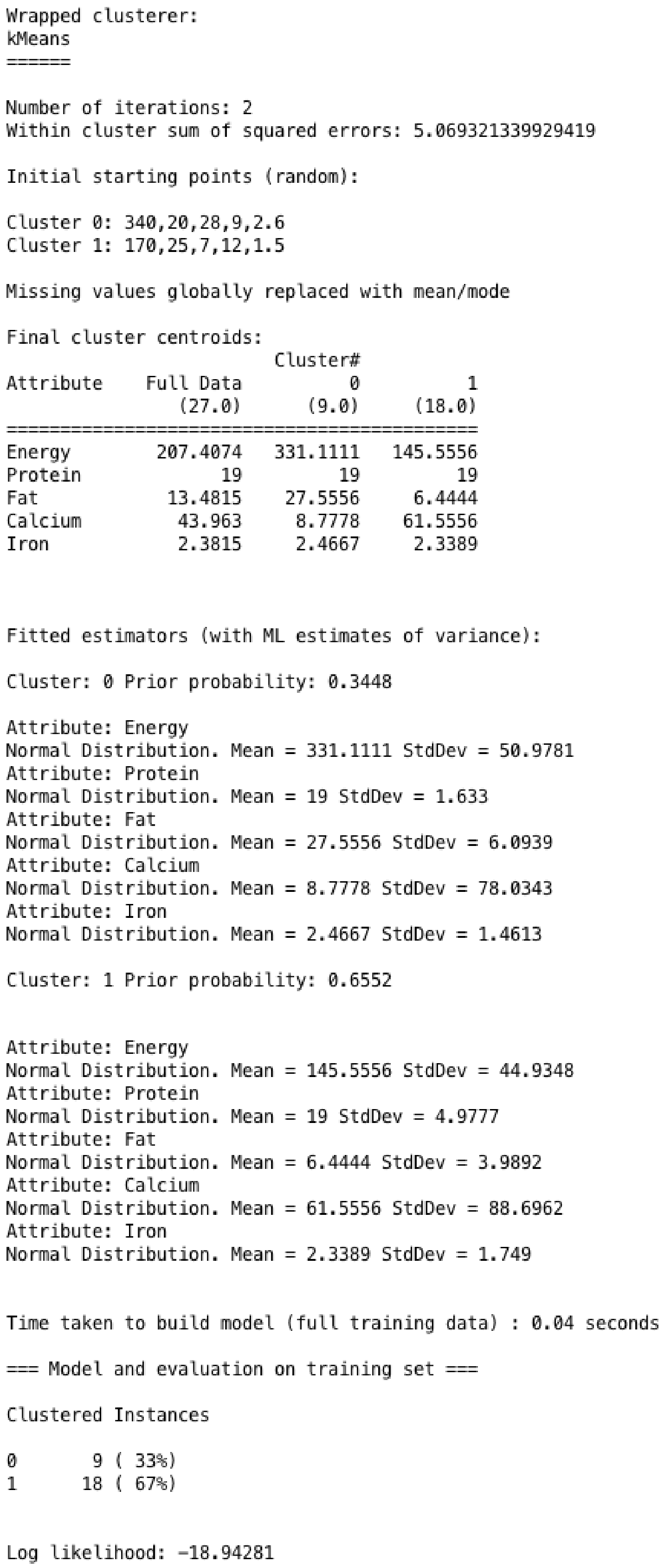


```
=== Run information ===

Scheme:       weka.clusterers. MakeDensityBasedClusterer -M 1.0E-6 -W weka.clusterers. SimpleKMeans -- -init 0 -
max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.
EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10
Relation:     food
Instances:    27
Attributes:   6
              Energy
              Protein
              Fat
              Calcium
              Iron
Ignored:
              Name
Test mode:    evaluate on training data
```



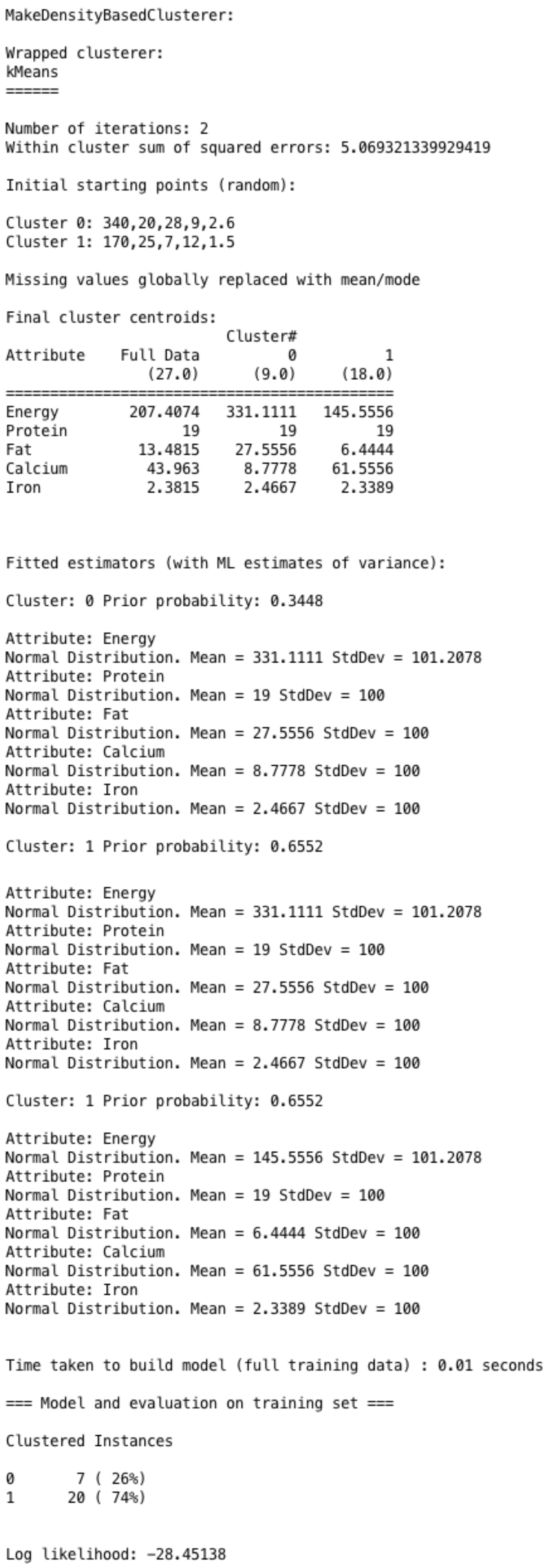
```
=== Clustering model (full training set) ===

MakeDensityBasedClusterer:

Wrapped clusterer:
kMeans
======

Number of iterations: 2
Within cluster sum of squared errors: 5.069321339929419

Initial starting points (random):

Cluster 0: 340,20,28,9,2.6
Cluster 1: 170,25,7,12,1.5

Missing values globally replaced with mean/mode

Final cluster centroids:
                         Cluster#
Attribute    Full Data          0          1
                (27.0)      (9.0)     (18.0)
============================================
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



Fitted estimators (with ML estimates of variance):

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
```

By keeping standard deviation = 1, the result exactly the same as we got from k-means algorithm in question 1 part 5.



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

Number of iterations: 2
Within cluster sum of squared errors: 5.069321339929419

Initial starting points (random):

Cluster 0: 340,20,28,9,2.6
Cluster 1: 170,25,7,12,1.5

Missing values globally replaced with mean/mode

Final cluster centroids:
                         Cluster#
Attribute    Full Data          0          1
                (27.0)      (9.0)     (18.0)
============================================
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



Fitted estimators (with ML estimates of variance):

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 = 78.0343
Attribute: Iron
Normal Distribution. Mean = 2.4667 StdDev = 1.4613

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) : 0.04 seconds

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

Clustered Instances

0       9 ( 33%)
1      18 ( 67%)


Log likelihood: -18.94281
```

Then keeping the standard deviation equals to 100 gives very different result, classifying more objects in “low fat high calcium” cluster“. But it can also be observed that increasing the standard deviation further beyond this point gives the same cluster and the standard deviation of all attributes almost become same.



```
MakeDensityBasedClusterer:

Wrapped clusterer:
kMeans
======

Number of iterations: 2
Within cluster sum of squared errors: 5.069321339929419

Initial starting points (random):

Cluster 0: 340,20,28,9,2.6
Cluster 1: 170,25,7,12,1.5

Missing values globally replaced with mean/mode

Final cluster centroids:
                         Cluster#
Attribute    Full Data          0          1
                (27.0)      (9.0)     (18.0)
============================================
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



Fitted estimators (with ML estimates of variance):

Cluster: 0 Prior probability: 0.3448

Attribute: Energy
Normal Distribution. Mean = 331.1111 StdDev = 101.2078
Attribute: Protein
Normal Distribution. Mean = 19 StdDev = 100
Attribute: Fat
Normal Distribution. Mean = 27.5556 StdDev = 100
Attribute: Calcium
Normal Distribution. Mean = 8.7778 StdDev = 100
Attribute: Iron
Normal Distribution. Mean = 2.4667 StdDev = 100

Cluster: 1 Prior probability: 0.6552

Attribute: Energy
Normal Distribution. Mean = 331.1111 StdDev = 101.2078
Attribute: Protein
Normal Distribution. Mean = 19 StdDev = 100
Attribute: Fat
Normal Distribution. Mean = 27.5556 StdDev = 100
Attribute: Calcium
Normal Distribution. Mean = 8.7778 StdDev = 100
Attribute: Iron
Normal Distribution. Mean = 2.4667 StdDev = 100

Cluster: 1 Prior probability: 0.6552

Attribute: Energy
Normal Distribution. Mean = 145.5556 StdDev = 101.2078
Attribute: Protein
Normal Distribution. Mean = 19 StdDev = 100
Attribute: Fat
Normal Distribution. Mean = 6.4444 StdDev = 100
Attribute: Calcium
Normal Distribution. Mean = 61.5556 StdDev = 100
Attribute: Iron
Normal Distribution. Mean = 2.3389 StdDev = 100


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

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

Clustered Instances

0       7 ( 26%)
1      20 ( 74%)


Log likelihood: -28.45138
```

We know that density-based clustering searches for the density reachable point (objects) from the randomly chosen point. By increasing the standard deviation, the variance increases, due to which the object which was initially density reachable from first cluster “High fat low calcium” cluster then became density reachable from “low fat high calcium” cluster. Notice that only the standard deviation of attributes changes in result and mean remains same throughout the process.