Clustering lab

Yi Yang (yiyan338) Helena Llorens Lluís (hllor282)

February 2025

1 Introduction

In this lab, we use the data mining toolkit Weka to solve clustering problems.

2 Dataset

The dataset used in this lab is derived from HARTIGAN. It gives nutrient levels of 27 kinds of food. The attributes include: Name (ignored during clustering), Energy, Protein, Fat, Calcium, and Iron. The **Name** attribute is ignored in clustering since it is categorical and does not contribute to numerical clustering. The **rest of the attributes** are used to cluster.

3 Methodology

3.1 SimpleKMeans

SimpleKMeans is a clustering algorithm that partitions a dataset into a predefined number of clusters (k) based on feature similarity. It starts by randomly selecting k initial centroids. Each data point is then assigned to the nearest centroid using Euclidean distance, which calculates the straight-line distance between points in a multi-dimensional space. After assignment, the centroids are updated as the mean of all points in their respective clusters. This process of assignment and update is repeated iteratively until the centroids stabilize or a maximum number of iterations is reached. The objective is to minimize the total within-cluster variance, ensuring that data points within each cluster are as close as possible to their centroid.

3.2 MakeDensityBasedClusterer

With MakeDensityBasedClusters, SimpleKMeans is adapted into a density-based clusterer by transforming the cluster assignments into density estimates. Instead of merely assigning points to the nearest centroid, this approach considers the distribution of points around each centroid, effectively measuring cluster density. This allows the algorithm to identify clusters of varying shapes and

densities, improving its ability to separate overlapping or non-spherical clusters. By combining the simplicity of K-means with density-based estimation, this method enhances clustering performance, particularly in complex datasets.

4 Experimental Results

4.1 SimpleKMeans

In Weka, changing the seed value affects the initial starting points (centroids) for the SimpleKMeans algorithm. Since K-means clustering relies on these initial centroids to start the clustering process, altering the seed can significantly impact the following:

- The initial centroids determine the initial cluster boundaries. If they are placed near natural data groupings, the algorithm quickly converges to well-separated clusters. Conversely, poor initial placement can lead to suboptimal clusters. This is because each point is assigned to the nearest centroid, and the initial placement influences which points are grouped together in the first iteration.

- Poorly placed initial centroids may result in more iterations or even convergence to a local minimum, leading to a less optimal clustering solution.

4.1.1 Clusters with k = 2

Table 1: K=2 Comparison

Metric	**Seed Value = 10**	**Seed Value = 20**
Number of iterations	2	2
Sum of squared errors	5.0693	6.4508

Both seed values converged to an optimal solution within just two iterations, suggesting that the model is relatively simple and reaches convergence quickly. The lower SSE with seed 10 indicates a better clustering fit, implying that the initial centroids chosen with seed 10 were closer to the true cluster centers.

Table 2: Comparison of initial starting points

Cluster	**Seed**	**Energy**	**Protein**	**Fat**	**Calcium**	**Iron**
0	10	340	20	28	9	2.6
0	20	45	7	1	74	5.4
1	10	170	25	7	12	1.5
1	20	90	14	2	38	0.8

As shown in the table, setting different seed values results in different starting

points. The seed value controls the selection of initial centroids, which in turn influences how the clusters are formed.

Table 3: Comparison of final clusters centroids

Cluster	Seed	Energy	Protein	Fat	Calcium	Iron
0	10	331.1111	19	27.5556	8.7778	2.4667
0	20	91.6667	14.6667	2.3333	56.6667	5.7667
1	10	145.5556	19	6.4444	61.5556	2.3389
1	20	221.875	19.5417	14.875	42.375	1.9583

Different seed values result in different starting points, which, after applying the K-means algorithm, lead to different final cluster centroids. This demonstrates how the choice of initial centroids influences the clustering process and ultimately affects the final cluster assignments.

4.1.2 Clusters with k = 5

Table 4: K=5 Comparison

Metric	Seed Value = 10	Seed Value = 20
Number of iterations	4	3
Sum of squared errors	2.7504	2.1761

A lower SSE indicates a better fit of the clusters to the data, suggesting that the initial centroids chosen with seed 20 were closer to optimal cluster centers. This also explains the quicker convergence.

Table 5: Comparison of initial starting points

Cluster	Seed	Energy	Protein	Fat	Calcium	Iron
0	10	340	20	28	9	2.6
0	20	45	7	1	74	5.4
1	10	170	25	7	12	1.55
1	20	90	14	2	38	0.8
2	10	90	14	2	38	0.8
2	20	245	21	17	9	2.7
3	10	180	22	9	367	2.5
3	20	300	18	25	9	2.3
4	10	300	18	25	9	2.3
4	20	170	25	7	12	1.5

The initial centroids vary significantly between the two seeds, influencing

how the clusters are formed. This large discrepancy in starting points illustrates how sensitive K-means is to the initial centroid selection, which ultimately affects the clustering outcome.

Table 6: Comparison of final clusters centroids

Cluster	Seed	Energy	Protein	Fat	Calcium	Iron
0	10	352.8571	18.5714	30.1429	8.7143	2.4143
0	20	57.5	9	1	78	5.7
1	10	153.125	23.25	5.75	23.75	2.45
1	20	149.1667	16.3333	7.5	64.6667	1.0167
2	10	102.5	13.5	3.8333	87.5	2.5333
2	20	206.6667	21.6667	12.5	10.8333	3.35
3	10	180	22	9	367	2.5
3	20	352.8571	18.5714	30.1429	8.7143	2.4143
4	10	222	18.8	15	8.8	2.02
4	20	146.6667	22.8333	5.1667	86.1667	1.6333

The final centroids also differ notably. The clusters are not simply permutations but represent different groupings, highlighting that K-means does not guarantee consistent labeling across different runs.

4.1.3 Cluster Labels

When comparing K = 2 and K = 5, the SSE is significantly lower for K = 5, indicating better clustering quality. Besides, with more clusters, members within each cluster are more similar to each other(lower SSE), while the separation between different clusters becomes clearer(the difference in the centroids of different clusters is quite significant), improving the overall clustering performance. Therefore, we chose K = 5 for labeling.

Table 7: Cluster Centroids for k = 5 and seed = 10

Attribute	Full Data (27)	Cluster 0 (7)	Cluster 1 (8)	Cluster 2 (6)	Cluster 3 (1)	Cluster 4 (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

- Cluster 0: High-Energy, High-Fat Foods

This cluster is characterized by the highest average energy (352.86) and fat content (30.14) among all clusters. The foods in this group are likely to be

calorie-dense and rich in fats. Despite the high energy and fat content, the protein (18.57) and iron (2.41) values are moderate, suggesting that these foods may not be particularly protein-rich.

- Cluster 1: High-Protein, Low-Fat Foods

Foods in this cluster have the highest protein content (23.25) combined with low fat (5.75) and moderate energy (153.13). This suggests lean protein sources such as poultry, fish, or legumes.

- Cluster 2: Low-Energy, Low-Fat Foods

This group is defined by the lowest values for both energy (102.5) and fat (3.83). These foods are likely low-calorie and low-fat options such as fruits or vegetables. They provide moderate calcium (87.5).

- Cluster 3: Calcium-Rich Foods

This cluster consists of only one item, it stands out due to its extremely high calcium content (367). This suggests calcium-rich foods such as dairy products. The energy (180), protein (22), and fat (9) values are moderate.

- Cluster 4: Moderate Energy, Balanced Macronutrients

This cluster shows a balanced nutrient profile with moderate energy (222), protein (18.8), and fat (15).

4.2 MakeDensityBasedClusters

Table 8: Prior probabilities

Cluster 0	**Cluster 1**	**Cluster 2**	**Cluster 3**	**Cluster 4**
0.25	0.2813	0.2188	0.0625	0.1875

4.2.1 $\sigma = 0.01$:

This value is small enough to prevent over-smoothing, preserving the natural variance in the data. It ensures that clusters with low dispersion do not get artificially expanded, maintaining tight and well-defined clusters.

Table 9: Clusters distributions with $\sigma = 0.01$

Cluster	**Energy**	**Protein**	**Fat**	**Calcium**	**Iron**
0	352.86 ± 34.42	18.57 ± 1.59	30.14 ± 4.12	8.71 ± 0.70	2.41 ± 0.20
1	153.13 ± 27.38	23.25 ± 1.85	5.75 ± 2.86	23.75 ± 28.59	2.45 ± 1.60
2	102.50 ± 37.94	13.50 ± 3.50	3.83 ± 2.85	87.50 ± 54.56	2.53 ± 2.28
3	180.00 ± 101.21	22.00 ± 4.25	9.00 ± 11.26	367.00 ± 78.03	2.50 ± 1.46
4	222.00 ± 27.86	18.80 ± 1.72	15.00 ± 3.16	8.80 ± 2.99	2.02 ± 0.72

4.2.2 $\sigma = 0.1$:

A slightly larger value allows for moderate smoothing, which can be beneficial if some clusters are sparse or have slight overlaps. This prevents the density estimate from becoming too spiky or sensitive to minor fluctuations.

Table 10: Clusters distributions with $\sigma = 0.1$

Cluster	Energy	Protein	Fat	Calcium	Iron
0	352.86 ± 34.42	18.57 ± 1.59	30.14 ± 4.12	8.71 ± 0.70	2.41 ± 0.20
1	153.13 ± 27.38	23.25 ± 1.85	5.75 ± 2.86	23.75 ± 28.59	2.45 ± 1.60
2	102.50 ± 37.94	13.50 ± 3.50	3.83 ± 2.85	87.50 ± 54.56	2.53 ± 2.28
3	180.00 ± 101.21	22.00 ± 4.25	9.00 ± 11.26	367.00 ± 78.03	2.50 ± 1.46
4	222.00 ± 27.86	18.80 ± 1.72	15.00 ± 3.16	8.80 ± 2.99	2.02 ± 0.72

4.2.3 $\sigma = 10$:

The value is significantly larger, which would further increase the smoothing effect, causing the differences between clusters to become more blurred. The clustering boundaries may become more relaxed, leading to the merging of clusters that should have been separated. This could result in a clustering outcome that is less accurate than expected.

Table 11: Clusters distributions with $\sigma = 10$

Cluster	Energy	Protein	Fat	Calcium	Iron
0	352.86 ± 34.42	18.57 ± 10.00	30.14 ± 11.26	8.71 ± 78.03	2.41 ± 10.00
1	153.13 ± 27.38	23.25 ± 10.00	5.75 ± 11.26	23.75 ± 28.59	2.45 ± 10.00
2	102.50 ± 37.94	13.50 ± 10.00	3.83 ± 11.26	87.50 ± 54.56	2.53 ± 10.00
3	180.00 ± 101.21	22.00 ± 10.00	9.00 ± 11.26	367.00 ± 78.03	2.50 ± 10.00
4	222.00 ± 27.86	18.80 ± 10.00	15.00 ± 11.26	8.80 ± 78.03	2.02 ± 10.00

Table 12: Clustered Instances

Cluster	Number of Instances	Percentage
0	7	26%
1	9	33%
2	6	22%
3	1	4%
4	4	15%

From the results, we observe that cluster 4 has one instance classified into cluster 1. This suggests that a large standard deviation can cause the boundaries between clusters to become less distinct, potentially leading to incorrect

classifications. When the standard deviation is too large, the boundaries between clusters become more blurred, resulting in the misclassification of data points. In this case, an instance that should belong to cluster 4 ended up in cluster 1, which highlights the potential negative impact of using a significantly large standard deviation on clustering performance.