TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT AND TECHNOLOGY

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

Clustering Food Products based on Nutritional Attributes

BMCS2114 MACHINE LEARNING

2023/2024

Student's name/ ID Number	:	Tay Wei Rong 22WMR03594
Student's name/ ID Number	:	Lim Jo Sun 22WMR01705
Programme	:	Bachelor of Computer Science (Honours) in Data Science
Tutorial Group	:	Group 2
Tutor's name	:	Dr. Lim Siew Mooi

Table of Contents

Abstract

This research investigates applying and optimising clustering algorithms to identify distinct nutritional profiles within a diverse food dataset. Three clustering methods—K-Means, Fuzzy C-Means (FCM), and DBSCAN—were employed, each subjected to hyperparameter tuning to enhance clustering performance. K-Means identified four distinct clusters, with minimal changes observed through tuning, indicating the robustness of initial settings. FCM demonstrated significant improvements in cluster coherence and separation, reflected in enhanced evaluation metrics post-tuning. DBSCAN, initially producing a high number of poorly separated clusters, exhibited substantial improvement after tuning, reducing the number of clusters and achieving better-defined groupings.

The findings reveal that K-Means is effective for datasets with well-separated clusters, while FCM offers a nuanced understanding of overlapping clusters. Although sensitive to parameter selection, DBSCAN's flexibility in identifying clusters of varying shapes and sizes proved advantageous for complex datasets. The study underscores the importance of hyperparameter tuning and method selection based on dataset characteristics.

The insights from this analysis can inform targeted dietary recommendations and nutritional planning, with potential applications in public health and personalised nutrition. Future research should explore additional clustering algorithms, integrate domain knowledge, and leverage advanced dimensionality reduction and feature engineering techniques to refine clustering outcomes further. Scalability considerations and semi-supervised learning approaches are also recommended to enhance the applicability and efficiency of clustering methods in more extensive and more complex datasets.

Keywords: *Open Food Facts, Clustering algorithms, K-Means, Fuzzy C-Means, DBSCAN, hyperparameter tuning, nutritional profiles, dietary recommendations, feature engineering, dimensionality reduction*

1.0 Introduction

Food nutrition clustering is a powerful machine-learning technique used to group foods based on similarities in their nutrient compositions. By analysing various nutrients such as vitamin B-6, calcium, iron, magnesium, folacin, and zinc and considering factors like added sugar, fat, cholesterol, and sodium content, foods can be clustered into distinct groups. This method aims to maximise similarity within clusters while enhancing dissimilarity between them. A seminal study by Windham et al. (1985) exemplified this approach by clustering foods into dairy, grain, and fat commodity groups based on their similar nutrient content, including vitamin B-6, calcium, iron, and magnesium. This study also introduced the concept of assigning a degree of association to each food, indicating its compositional similarity to a prototype food within the cluster group.

Building upon this foundation, a more recent study by O'Hara et al. (2022) applied clustering methods to data from the 2008–2010 Irish National Adult Nutrition Survey (NANS). Here, foods were clustered based on their similarity to 12 nutrients included in the Nutrient Rich Food Index (NRF9.3), such as protein, fibre, vitamin A, and calcium. Employing techniques like k-means clustering and partitioning around medoids, the researchers identified food groups and grouped similar meals based on their NRF9.3 scores and food groupings. Moreover, cluster analysis extends beyond food categorisation; it can also be applied to categorise individuals into dietary patterns based on their similar food consumption frequencies (Alosaimi et al., 2023). Additionally, clustering unhealthy behaviours, such as poor diet, lack of physical activity, and sedentary behaviour, has been linked to adverse mental and physical health outcomes.

Based on these foundational studies, the present research utilises the Open Food Facts database to apply clustering techniques to a broader range of food items based on their nutrient profiles. Open Food Facts is an extensive, crowdsourced database that provides detailed nutritional information for thousands of food products worldwide. By leveraging this rich dataset, this study aims to identify meaningful clusters of foods based on nutrient composition, expanding upon the methodologies and findings of previous research. However, due to the nature of user-generated data, the dataset introduces challenges such as inaccuracies. Therefore, our objectives include several vital considerations. First, we aim to ensure data quality through rigorous validation and verification. Second, we will experiment with various clustering models to evaluate their compatibility and characteristics. By comparing different models, such as k-means, fuzzy c-means, and DBSCAN, we can determine which method best identifies meaningful patterns and structures within the nutrient data. This comparative analysis will help us understand the strengths and limitations of each model in the context of food nutrition clustering. Lastly, by analysing foods based on their nutritional properties, we can understand the potential health effects associated with different food categories, thus helping to evaluate dietary choices and their impact on health outcomes.

1.1 Problem Statement

The increasing complexity of dietary patterns and the growing availability of diverse food products pose significant challenges in understanding the nutritional landscape and its impact on public health. Traditional methods of analysing food nutrient data must improve their ability to identify meaningful patterns and relationships within large datasets. This limitation hampers efforts to provide clear, actionable insights for improving dietary choices and health outcomes.

Open Food Facts, a comprehensive, crowdsourced database, offers a rich source of nutritional information for thousands of food products worldwide. However, the user-generated nature of this data introduces challenges, such as inconsistencies and inaccuracies, which complicate its effective use in nutritional studies.

Given these challenges, robust methods for clustering foods based on their nutrient profiles are critical to ensuring data quality and reliability. Additionally, exploring the compatibility and characteristics of various clustering models is essential to identifying the most effective approaches for uncovering patterns in dietary preferences and evaluating the potential health effects associated with different food categories.

Key Considerations:

1) *Data Quality:* Given the user-generated nature of the dataset, ensuring data quality through validation and verification processes is paramount to maintain the integrity of analyses.

2) *Model Comparison:* Experiment with various clustering models, such as distance-based and density-based models, to evaluate their compatibility and effectiveness in identifying meaningful patterns within the nutrient data.

2) *Nutritional Patterns:* Clustering food products based on nutritional content can reveal patterns in dietary preferences, helping identify trends in consumption and nutritional habits.

3) *Health Impacts:* Analysing food products based on nutritional attributes can provide insights into the potential health impacts associated with different food categories, aiding in assessing dietary choices and their effects on health outcomes.

1.2 Objectives

1. To clean and pre-process the Open Food Facts dataset for reliable nutritional information.

2. To apply 3 clustering algorithms to group food products based on nutritional content, considering dietary preferences and health implications.

3. To analyse clusters to identify nutritional patterns, dietary trends, and health associations.

4. To visualise clusters using dimensionality reduction and data visualisation techniques for more straightforward interpretation.

5. To evaluate clustering algorithms' effectiveness using metrics like Silhouette Score, Davies–Bouldin index and Calinski-Harabasz Index.

2.0 Literature Review

In the field of food nutrition analysis, clustering techniques play a crucial role in uncovering patterns and relationships within large datasets. Researchers can gain insights into dietary habits and health outcomes by grouping similar items based on nutrient profiles. This literature review explores the principles of unsupervised machine learning and various clustering methods, highlighting their applications and challenges in the context of nutritional studies.

2.1 Unsupervised Machine Learning

Unsupervised machine learning is a branch of artificial intelligence focused on training algorithms to identify patterns and structures in data without explicit instructions or labelled examples (Kumar & Singh, 2021). Unlike supervised learning, where algorithms rely on labelled data to learn, unsupervised learning algorithms analyse unlabeled data, enabling them to uncover hidden patterns and relationships autonomously. In the context of clustering, these algorithms divide a dataset into categories or regions, ensuring that data points within the same group or cluster are more similar than those in other clusters (Zhou et al., 2019).

Clustering differs from classification because it involves grouping a large set of entities into smaller clusters based on their similarities without predefined classes. In contrast, classification sorts entities into predefined categories (Ponder-Sutton, 2015). Clustering is particularly useful when the categories are unknown, making it an effective tool for identifying problem areas in business contexts. Clustering algorithms use selected attributes to determine similarity, and individuals with domain knowledge can interpret the results to find meaningful patterns within the clusters.

Practical clustering algorithms should meet several criteria: scalability, the ability to handle various attribute types, the discovery of clusters with arbitrary shapes, minimal requirements for domain knowledge to determine input parameters, and the ability to manage noise, outliers, and missing data (Crase & Thennadil, 2022; Wegmann et al., 2021). Additionally, they should be insensitive to the order of input records, capable of handling high-dimensional data, and provide interpretability and usability. Despite these advantages, clustering methods face challenges such as time complexity, dependence on the definition of similarity measures, and different interpretations of the results produced by the clustering algorithm (Kononenko & Kukar, 2006).

Several clustering techniques are available, including K-Means Clustering, Hierarchical Clustering, Prototype-based Clustering, Density-based Clustering, and Model-based Clustering (Talabis et al., 2014). These techniques can be categorised based on the algorithmic approach used to identify clusters within a dataset. Types of clusters include exclusive or strict

partitioning, overlapping, hierarchical, and fuzzy or probabilistic clusters (Kotu & Deshpande, 2018). The selection of an appropriate clustering technique depends on the data's nature and the analysis's specific objectives (Talabis et al., 2014). For instance, K-Means Clustering is suitable for large datasets, while Hierarchical Clustering is more effective for smaller datasets. Model-based Clustering is advantageous when data points follow the same probability distribution (Kotu & Deshpande, 2018). Researchers must choose relevant attributes for clustering carefully, and employing automated feature selection methods can help reduce the dimensionality of the dataset for more efficient clustering.

2.2 Evaluation Metrics for Clustering

Clustering evaluation metrics are essential for assessing the quality and effectiveness of clustering algorithms. These metrics can be broadly classified into intrinsic and extrinsic measures (Han et al., 2011). Intrinsic metrics evaluate the goodness of a clustering structure without reference to external information, while extrinsic metrics compare the clustering results to a predefined benchmark or ground truth. The ground truth can be considered as supervision in the form of "cluster labels." Hence, extrinsic methods are also known as supervised methods, while intrinsic methods are unsupervised methods. Since the present study is an unsupervised machine learning project, we will focus on intrinsic metrics.

According to Rajendiran et al. (2022), clustering evaluation metrics could be of three types: silhouette score, Davies-Bouldin index (DB), and Calinski-Harabasz index (CH).

Silhouette Score

The Silhouette Score, also known as the Silhouette Coefficient, measures the similarity between an object and its own cluster compared to others. It ranges from -1 to 1, where a higher value indicates better-defined clusters (Rajendiran et al., 2022). The coefficient combines cohesion (intra-cluster distance) and separation (inter-cluster distance), providing a comprehensive view of clustering quality (Rousseeuw, 1987). Cohesion, denoted as $a(i)$ in Eq. (1), is the mean distance between the i object and all the other data within the same cluster, C_I (Han et al., 2011). On the other hand, separation, $b(i)$, given in Eq. (2), is defined as the smallest mean distance of the i to all points in any other cluster, of which i is not a member. In other words, $b(i)$ is the mean distance between i to all the points of its nearest neighbour cluster. Lastly, the Silhouette score of i is then defined as Eq. (3). However, when the silhouette score is negative, for instance, when $b(i) < a(i)$, given in Eq. (3), it means that i is closer to the objects in another cluster than to the objects in the same cluster, C_I. In simple terms, this means the object is likely assigned to the wrong cluster.

(1) $$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} d(i,j)$$

(2) $$b(i) = min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i,j)$$

(3) $$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

Davies-Bouldin Index

The Davies-Bouldin index measures the average similarity ratio of each cluster with its most similar cluster, with lower values indicating better clustering performance and a lower ratio reflecting well-separated and distinct clusters (Davies & Bouldin, 1979). The similarity score, R_{ab}, given in Eq. (4), is calculated between clusters a and b. The d_a is the within-cluster scatter for cluster a that calculates the average distance between each point in the cluster a and its centroid. The D_{ab} is the between-cluster separation that measures the distance between the centroids of the two clusters, a and b (Rajendiran et al., 2022). For each cluster i, the maximum R_{ab} value over all other clusters j is computed, representing the worst-case similarity with any other cluster.

$$R_{ab} = \frac{d_a + d_b}{D_{ab}} \tag{4}$$

$$DB = \frac{1}{k} \sum_{i=1}^{k} max_{i \neq j} R_{ab} \tag{5}$$

Calinski-Harabasz Index

The Calinski-Harabasz Index, also known as the Variance Ratio Criterion, evaluates the ratio of between-cluster dispersion and within-cluster dispersion. A higher Calinski-Harabasz index indicates better-defined clusters, suggesting a model with well-separated clusters and compact within-cluster distances (Calinski & Harabasz, 1974). The between-cluster dispersion, B_k, given in Eq. (6), is calculated as the sum of squared distances between the clusters' centroids and the overall centroid of the dataset, weighted by the number of points in each cluster (Wang & Xu, 2019). The within-cluster dispersion is the sum of squared distances between each point and its cluster centroid, across all clusters, given by Eq. (7). The Calinski-Harabasz index is then calculated using Eq. (8), where k is the number of clusters and n is the number of points in the dataset. The metric is particularly useful when comparing the results of different clustering algorithms or parameter settings for the same algorithm (Halkidi et al., 2001). It balances the trade-off between cluster compactness by minimising within-cluster dispersion and cluster separation by maximising between-cluster dispersion.

$$B_k = \sum_{i=1}^{k} n_i ||c_i - c||^2 \tag{6}$$

(7) $$W_k = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - c_i||^2$$

(9) $$CH = \frac{B_k / (k-1)}{W_k / (n-k)}$$

2.3 K-Means

K-means clustering is an unsupervised learning algorithm that identifies clusters in unlabeled data by estimating the number of clusters (k) and creating k pseudo-centres. As Nettleton (2013) described, it is a widely used technique for partitioning data into distinct groups based on similarity.

2.3.1 Concept and Algorithm

The method commences by creating k pseudo-centers, representing initial cluster centroids (Cohen, 2020). Subsequently, each data point is assigned to the nearest pseudo-center, thereby forming clusters comprised of data points associated with each pseudo-center. To refine cluster assignments, the centroid of each cluster is recalculated based on the mean of its member data points, a step iteratively carried out until the centroids stabilise, as noted by Davies et al. (2018). This iterative process involves updating the pseudo-centers to the centroids' locations, thereby improving cluster cohesion. As convergence is reached, the data is partitioned into k subsets, with each subset containing data points closest to its respective centroid.

The termination criterion for the clustering process typically involves a predefined number of iterations, ensuring convergence and stability. At this point, the current clusters are finalised, and the resulting partitioning of the data is established. Despite its widespread use and simplicity, a fundamental limitation of k-means clustering is the requirement for the user to specify the number of clusters k in advance. This necessitates an informed guess based on the given input variables to ensure effective clustering outcomes.

2.3.2 K-Means Hyperparameters

In machine learning, hyperparameters are model arguments set before the learning process begins and control the learning process (Wistuba et al., 2015). These are distinct from parameters, which are internal values derived automatically during the learning process. The primary hyperparameters in K-means clustering include the number of clusters (k), the initialisation method for cluster centres, the number of iterations (max_iter), and the number of initialisations (n_init) (Gikera et al., 2023). According to the Scikit-Learn documentation, n_init refers to the number of times the algorithm is initialised, while n_clusters (the k-hyperparameter) indicates the number of clusters generated by the K-means algorithm (Pedregosa et al., 2011). Among these, the k-hyperparameter is especially crucial in high-dimensional K-means clustering due to its significant impact on the clustering outcome (Gikera et al., 2023).

2.3.3 Optimal number of *k*

Standard methods for identifying the optimal number of clusters (k) in K-means clustering include the Silhouette Score and the Elbow Method using the sum of squared error. Gul and Rehman (2023) emphasise these approaches for their effectiveness in cluster analysis. The Elbow Method, as explained by Gikera et al. (2023), involves plotting the Within-Cluster Sum of Squares (WCSS) against the number of clusters and identifying the "elbow" point where the curve bends, indicating the optimal k.

Lloyd (1982) describes WCSS as the sum of squared distances between each point and the centroid of its assigned cluster, providing a measure of cluster compactness. Lower WCSS values indicate more compact clusters. When WCSS is plotted against the number of clusters, the point where the slope changes abruptly suggests the optimal number of clusters (Gul & Rehman, 2023). This method allows for a visual and analytical determination of the most suitable k for a given dataset, balancing compactness and cluster separation. Some of the other studies use Silhouette analysis to find the optimal k value, or some simply test their dataset with different k values and pick the best results (Mirzaei & Adeli, 2022).

2.3.4 Applications of K-Means Clustering

The K-Means algorithm is widely used due to its simplicity and effectiveness. It has diverse applications across various fields, including brain region segmentation, feature extraction prior to classification tasks, production data analysis, and real-time detection of liquid loading (Davies, 2014; Mirzaei & Adeli, 2022). Additionally, K-Means has been employed in predicting stock prices, analysing chemotherapeutic responses in cancer, and assessing fabric friction properties, demonstrating its versatility (Li & Huang, 2021; Nguyen et al., 2014; Sun et al., 2017).

In food and beverage/nutrition research, K-Means clustering is particularly valuable. One notable application involves analysing consumer trends in functional foods. The algorithm effectively clusters data points with similar characteristics, offering insights into consumer behaviour and preferences (Sgroi et al., 2024). This capability is crucial for understanding market trends and tailoring products to meet consumer demands.

Another significant application of K-Means is in meal-based dietary intake analysis. Researchers can derive distinct dietary patterns by categorising food groups and meals based on nutrient content and consumption patterns. This enables a detailed analysis of population and individual dietary habits, providing valuable information for nutritional studies (O'Hara et al., 2022).

K-Means clustering is also utilised to identify food consumption patterns among individuals, helping researchers categorise individuals into clusters based on their dietary habits. This

application allows for grouping individuals with similar dietary intake patterns, providing insights into different dietary behaviours and preferences (Qasrawi et al., 2021).

Furthermore, K-Means clustering has predictive capabilities in understanding consumer interest according to seasonal variations. The algorithm analyses food menu sales data and helps restaurants optimise their menus based on consumer demand and seasonal preferences (Pasaribu, 2020). This application highlights the practical benefits of K-Means in the food service industry.

Another application in food and nutrition research is deriving dietary patterns. Researchers can identify distinct dietary patterns among individuals or populations by applying K-Means clustering to nutritional data. This method helps categorise individuals into specific dietary clusters based on their nutrient intake, aiding in the analysis of dietary habits and their impact on health outcomes (Sauvageot, 2017).

Lastly, K-Means clustering has been utilised in dietary pattern analysis associated with colorectal cancer. By identifying dietary patterns related to health outcomes, researchers can better understand the impact of diet on disease risk. This application underscores K-Means' potential to contribute to public health research by elucidating the connections between diet and disease (Qarmich et al., 2022).

2.4 Fuzzy C-Means

Fuzzy C-Means (FCM) clustering is a type of partitive clustering technique that allows data points to belong to multiple clusters with varying degrees of membership. Unlike K-Means, which assigns each data point to a single cluster, FCM uses fuzzy logic to assign membership levels to each point, providing a more nuanced understanding of the data structure (Bezdek, 1981).

2.4.1 Concept and Algorithm

The FCM algorithm is based on the minimisation of an objective function, similar to K-Means, but incorporates a fuzziness parameter m (where $m > 1$) that controls the degree of fuzziness. The objective function is given by Eq. (10), where u_{ij} is the degree of membership of the data point x_i in the cluster c_j (Kononenko & Kukar, 2006; Amma Palanisamy et al., 2018). The membership degrees are updated using Eq. (11), and the cluster centres are then updated as Eq. (12).

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} ||x_i - c_j||^2 \quad (10)$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{||x_i - c_j||}{||x_i - c_k||}\right)^{2/(m-1)}} \quad (11)$$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \quad (12)$$

2.4.2 Fuzzy C-Means Hyperparameter

In Fuzzy C-Means (FCM), hyperparameters are crucial for controlling the algorithm's performance and outcomes. These include the fuzziness parameter *m*, the number of clusters *c*, the maximum number of iterations (max_iter), and the stopping criterion (epsilon) (Pedregosa et al., 2011). The fuzziness parameter *m* determines the degree of cluster fuzziness, influencing how data points can belong to multiple clusters (Wu, 2011; Bezdek, 1981). Higher *m* values result in softer cluster boundaries, whereas lower values produce crisper clusters, resembling hard clustering approaches. The number of clusters *c* is another critical hyperparameter, dictating how many clusters the data will be partitioned into. Methods for selecting *c* include the Elbow

Method, Silhouette Analysis, and the Gap Statistic, each providing a different approach to determine the optimal number of clusters based on data characteristics (Charrad et al., 2014; Arbelaitz et al., 2013). The maximum number of iterations (max_iter) sets the limit on the number of times the algorithm will update cluster centres and memberships (Pedregosa et al., 2011). This ensures the algorithm terminates even if convergence is not achieved. The stopping criterion (epsilon) defines the threshold for changes in cluster centres or membership values, indicating when the algorithm should stop iterating. A smaller epsilon value leads to more precise convergence but requires more iterations.

2.4.3 Optimal Number of *m*

One of the most critical hyperparameters is the fuzziness parameter *m*, as highlighted by Wu (2011). The *m* value is typically set to a value greater than 1, *m* controls the degree of fuzziness in cluster assignments. Bezdek (1981) notes that *m* normally falls within the range of 1.1 to 5. Higher values of *m* result in softer, more gradual membership assignments, allowing data points to belong to multiple clusters simultaneously with varying degrees of membership. Conversely, lower values of *m* tend to produce sharper cluster boundaries, resembling traditional hard clustering approaches like K-Means. Pal and Bezdek (1995) introduced a heuristic rule for selecting an optimal value for *m*, restricting its range to [1.5, 2.5]. They proposed that the median value $m = 2$ could be chosen without specific constraints. This heuristic provides a practical guideline for selecting *m*, ensuring a balance between cluster fuzziness and discriminability. Conversely, Choe and Jordan (1992) proposed an alternative methodology rooted in maximising membership values to ascertain the optimal *m* based on a "good" cluster criterion. However, empirical investigations reveal FCM's robustness across a spectrum of *m* values, evincing a degree of insensitivity to variations.

A seminal contribution by Wu (2011) advances a novel guideline for *m* selection, proposing a range between 1.5 to 4 or alternatively advocating for a suitable larger *m* value. Notably, empirical findings from Wu assert that an *m* value of 4 emerges as the most conducive for FCM, particularly in datasets beset with noise and outliers. Furthermore, corroborative insights from Torra (2015) accentuate the discernible impact of *m* variations on clustering outcomes, accentuating the propensity for larger *m* values to obfuscate class distinctions. While enhancing FCM's resilience to noise, such augmented *m* values concurrently mitigate the distinctiveness of clustering results.

2.4.4 Comparison to Hard Clustering

Fuzzy C-Means (FCM) and K-Means represent two distinct approaches to clustering. K-Means, a hard clustering method, assigns each data point to a single cluster, resulting in distinct and

non-overlapping groups. In contrast, FCM, a soft clustering technique, allows data points to belong to multiple clusters with varying degrees of membership, as defined by the fuzziness parameter *m* (Bezdek, 1981).

Due to this feature, FCM has a significant advantage over K-Means in handling overlapping clusters and irregular pattern datasets more efficiently (Nayak et al., 2023). For instance, Sivarathri and Govardhan (2014) demonstrated that FCM outperforms K-Means in clustering accuracy on the UCI diabetes dataset. Similarly, EtehadTavakol et al. (2010) found that FCM provided more accurate colour segmentation in infrared breast images, avoiding the empty cluster issue encountered with K-Means. In image segmentation, FCM captures gradual transitions between regions more effectively, as highlighted by Kishor Duggirala (2020).

Furthermore, Mingoti and Lima (2006) compared various clustering algorithms, including FCM and the SOM neural network, using 2530 simulated datasets with varying degrees of overlap and outliers. Their results showed that FCM performed well across all scenarios, maintaining stability despite outliers and overlapping clusters. In contrast, these factors significantly affected other algorithms, particularly the SOM neural network.

However, FCM's flexibility comes at the cost of increased computational complexity. FCM typically requires more iterations to converge compared to K-Means, as it continually updates membership degrees for all clusters (Wu, 2011). Ghosh and Dubey (2013) found that the time complexity of FCM increases more rapidly than K-Means as the number of clusters grows. Panda et al. also tested FCM and K-Means with Manhattan and Euclidean distance measures, concluding that while both algorithms performed well, K-Means were more computationally efficient. FCM with Manhattan distance produced the most compact clusters, whereas K-Means with Euclidean distance yielded the most distinct clusters.

On the other hand, some studies have noted limitations in FCM's accuracy. Madhukumar and Santhiyakumari (2015) found that K-Means outperformed FCM in classifying tissue types in T1 contrast axial plane MR images. Similarly, Simhachalam and Ganesan (2016) reported that K-Means provided better classification results than FCM and Gustafson–Kessel (GK) clustering algorithms on datasets such as liver disorder and wine from the UCI repository.

2.4.5 Applications of Fuzzy C-Means Clustering

Fuzzy C-Means (FCM) clustering has demonstrated its versatility across various fields due to its ability to handle overlapping data and provide nuanced cluster memberships. In medical imaging, FCM enhances the segmentation of MRI and CT scans by accurately distinguishing between overlapping tissues and organs (Pham et al., 2000). It is also widely used in pattern

recognition tasks, such as handwriting recognition and image analysis, improving accuracy by accommodating multiple category memberships (Pal & Bezdek, 1995). Furthermore, FCM aids market segmentation by identifying customer segments based on purchasing behaviour, leading to more precise targeting (Pradipta et al., 2018). In bioinformatics, FCM effectively clusters gene expression data, revealing intricate gene functions and interactions (Dembélé & Kastner, 2003). Recent studies have further underscored the practical applications of FCM. Hashemi et al. (2023) highlighted the method's effectiveness in optimising data clustering tasks, emphasising its simplicity and popularity. Another study by Krasnov et al. (2023) reviewed the use of FCM in breast cancer detection, showcasing its potential in medical imaging for early cancer detection through enhanced image segmentation techniques.

FCM's applications in the food and beverage industry are equally notable. Adriyendi (2016) applied K-Means and FCM clustering techniques to analyse food productivity, demonstrating FCM's suitability for categorising food productivity data where data points may belong to multiple clusters simultaneously. Additionally, Li et al. (2009) explored the use of FCM in metabolomics data analysis in the beverage industry, highlighting its effectiveness in identifying patterns and grouping data points with soft assignments, which is beneficial when data points may belong to multiple clusters at once.

2.5 DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm known for identifying clusters of varying shapes and sizes and detecting noise (Ester et al., 1996). Unlike partition-based methods like K-Means, DBSCAN relies on the density of data points to form clusters, making it particularly effective in identifying irregularly shaped clusters. The algorithm works by defining clusters as areas of high point density, separated by areas of low point density.

2.5.1 Concept and Algorithm

The algorithm defines clusters based on the density of data points, requiring two key parameters: epsilon (ϵ), which specifies the radius of the neighbourhood around a data point, and minPoints, which indicates the minimum number of points required to form a dense region (Ester et al., 1996).

DBSCAN classifies points into three categories: core, border, and noise points. Core points are those with at least minPoints within the ϵ radius. Border points lie within the ϵ radius of a core point but do not meet the minPoints criterion themselves. Noise points are those that do not fall into the first two categories and are considered outliers. The algorithm begins with an arbitrary point and retrieves its ϵ-neighbourhood. If the point is a core point, a cluster is formed. The neighbourhood of each core point in the cluster is iteratively scanned for more core points, expanding the cluster until it is fully expanded. This process continues with the next unvisited point.

The objective function of DBSCAN can be represented by Eq. (13), which ensures that clusters are formed based on density. $N_\epsilon(p)$, given by Eq. (13), represents the ϵ-neighbourhood of the point p, and D is the dataset (Ester et al., 1996; Schubert et al., 2017).

$$N_\epsilon(p) = \{q \in D \mid dist(p, q) \leq \epsilon\} \tag{13}$$

2.5.2 DBSCAN Hyperparameter

The performance of DBSCAN is significantly influenced by its hyperparameters, epsilon (ϵ) and minPoints. Epsilon (ϵ) defines the radius for the neighbourhood search around each data point. A smaller $\epsilon\epsilon$ value can result in many small clusters and an increased number of noise points, as only points very close to each other will be considered part of the same cluster. Conversely, a larger $\epsilon\epsilon$ value can lead to fewer, larger clusters, potentially merging distinct clusters and reducing the algorithm's ability to identify meaningful structures (Ester et al., 1996). The

minPoints parameter specifies the minimum number of points required to form a dense region. This value is typically set to the dimensionality of the dataset plus one, ensuring that clusters have sufficient density to be meaningful (Ester et al., 1996).

Selecting appropriate values for these parameters is crucial for DBSCAN's effectiveness. The k-distance graph is a common technique used to determine the optimal ϵ. By plotting the distance to the k-th nearest neighbour (where k is often set to minPoints) for each data point, one can identify the "elbow" point where the distance increases rapidly. This elbow represents a balance point, suggesting a suitable $\epsilon\epsilon$ value (Schubert et al., 2017).

2.5.3 Optimal Number of ϵ

Selecting the optimal ϵ value is critical for the successful application of DBSCAN. The k-distance graph method is one of the most widely used techniques. By plotting the distance of each point to its k-th nearest neighbour and identifying the "elbow" point in the graph where the distance starts to increase sharply, one can estimate a suitable ϵ value. This heuristic approach provides a visual means to balance cluster formation and noise reduction (Ester et al., 1996; Schubert et al., 2017).

In addition to the k-distance graph, other methods include heuristic and adaptive approaches. For instance, a common practice is starting with a small ϵ and incrementally increasing it until a desired clustering structure is achieved, guided by domain knowledge and specific application requirements (Birant & Kut, 2007). Recent advancements have also introduced optimisation algorithms and machine learning models to automate the ϵ selection process, aiming to maximise clustering quality metrics like the silhouette coefficient or the Davies-Bouldin index (Rodriguez & Laio, 2014).

2.5.4 Comparison to Other Clustering Algorithms

DBSCAN offers several advantages over traditional clustering algorithms like K-Means and Fuzzy C-Means (FCM). One of its key strengths is its ability to handle noise effectively. DBSCAN identifies and categorises points that do not belong to any cluster as outliers, making it robust in noisy datasets (Sander et al., 1998). This feature is particularly useful in real-world applications where data often contain noise and outliers. Another significant advantage of DBSCAN is its ability to find clusters of arbitrary shapes. Unlike K-Means, which assume spherical clusters, DBSCAN can detect clusters of various shapes and densities, making it versatile for complex datasets (Schubert et al., 2017). This flexibility is beneficial in geospatial analysis and image processing, where natural clusters do not conform to simple geometric shapes.

However, DBSCAN has its disadvantages. The algorithm's performance heavily depends on the choice of the epsilon (ϵ) and minPoints parameters. Selecting appropriate values for these parameters can be challenging, particularly in high-dimensional data where the distance metric may not be intuitive (Tan et al., 2005). Misestimating ϵ or minPoints can lead to poor clustering results, either by merging distinct clusters or by splitting meaningful clusters into smaller ones. Additionally, DBSCAN can be computationally expensive for large datasets due to the neighbourhood search process. The algorithm's complexity is typically $O(n \log n)$, but it can become inefficient when the dataset size increases significantly, limiting its scalability (Schubert et al., 2017). This computational overhead can be a significant drawback in big data applications where efficiency is crucial.

2.5.5 Applications of DBSCAN Clustering

DBSCAN has demonstrated its robustness and flexibility across various fields due to its ability to discover clusters of arbitrary shapes and effectively handle noise. In geospatial data analysis, DBSCAN is used to detect spatial clusters and anomalies, such as crime hotspots or disease outbreaks (Sander et al., 1998). In image analysis, it identifies clusters of pixels that form distinct objects or regions within images, enhancing the accuracy of image segmentation (Schubert et al., 2017). DBSCAN is also effective in detecting anomalies in financial transactions or network intrusions, making it a valuable tool in cybersecurity (Tan et al., 2005). In market research, DBSCAN helps segment customers based on purchasing behaviour, identifying niche markets that traditional clustering algorithms might miss (Karypis et al., 1999). Additionally, in social network analysis, DBSCAN detects communities and influential nodes, providing insights into social structures and interactions (Kumar et al., 2011).

A study by Bushra and Yi (2021) conducted a comparative analysis of DBSCAN and successive density-based clustering algorithms, emphasising DBSCAN's pioneering role in density-based clustering and its effectiveness in handling spatial data and identifying clusters of arbitrary shapes. This study underscores the importance of DBSCAN in providing accurate clustering results in the presence of noise, making it a valuable tool for analysing complex datasets in food and beverage nutrition research. Another study by Sander et al. (2008) focused on the application of DBSCAN in spatial databases, highlighting its ability to discover clusters of arbitrary shape and distinguish noise effectively. The research also introduced GDBSCAN, an extension of DBSCAN that clusters both point objects and spatially extended objects based on their spatial and non-spatial attributes, demonstrating its applicability to diverse problems, including those related to food, beverages, and nutrition research.

In the food and beverage industry, DBSCAN's applications are equally notable. Alqorni et al. (2021) utilised DBSCAN to determine the quality of Keprok Orange and Siam Orange hybrids at the Research Center of Orange and Subtropic Plants, effectively identifying clusters and handling outliers, showcasing its application in quality assessment within the food and beverage industry. Additionally, another study explored the use of DBSCAN for analysing nutritional data and identifying dietary patterns among individuals. By leveraging DBSCAN's ability to identify clusters of arbitrary shapes and handle outliers, researchers uncovered complex patterns in nutritional datasets, aiding in the analysis of dietary habits and preferences.

3.0 Methodology

This section outlines the methodology used in this research, detailing the processes involved in data collection, preprocessing, model implementation, and evaluation. Each step is crucial for ensuring the integrity and accuracy of the analysis, ultimately leading to meaningful and reliable results.

3.1 Data Collection

The data used in this research is sourced from the Open Food Facts dataset available on Kaggle (https://www.kaggle.com/datasets/openfoodfacts/world-food-facts). The dataset is provided in TSV format and is approximately 114 MB in size. It contains 356,027 rows and 163 columns and covers a wide range of food products from around the world. This dataset includes various attributes such as product names, categories, nutritional information, ingredients, labels, and packaging details.

3.1.1 Data Description

The Open Food Facts dataset includes the following key attributes:

- **Product Name:** The name of the food product.
- **Categories:** Categories to which the product belongs, such as "Beverages," "Snacks," etc.
- **Nutritional Information:** Details on nutrients like energy (kJ), fat, carbohydrates, sugars, fibre, proteins, and salt.
- **Ingredients:** List of ingredients used in the product.
- **Labels:** Information on labels such as "Organic," "Gluten-Free," etc.
- **Packaging:** Details about the product packaging.

3.1.2 Data Overview

In conducting a preliminary analysis of the dataset, several key observations were made to understand its structure and content. Firstly, it was noted that certain columns within the dataset contain missing values, necessitating careful consideration and addressing during the preprocessing stage to ensure the integrity and completeness of the data. Secondly, the dataset encompasses diverse data types, encompassing categorical, numerical, and textual data, reflecting the multifaceted nature of the information it encapsulates. This heterogeneity underscores the need for tailored preprocessing techniques to appropriately handle and transform the various data types for subsequent analysis. Potential duplicate entries within the dataset were flagged for further scrutiny and resolution. Identifying and rectifying duplicate entries is crucial

to prevent skewing results and ensure subsequent analyses' accuracy and reliability. These initial observations provide valuable insights into the dataset's characteristics and serve as a foundation for data preprocessing and analysis.

3.2 Preprocessing

Preprocessing is a crucial step in preparing the dataset for analysis and clustering with the algorithms. This process involves several steps to clean, standardise, and structure the data, ensuring it is suitable for machine learning algorithms. Effective preprocessing enhances the quality of the data, reduces noise, and addresses inconsistencies, thereby improving the accuracy and performance of the subsequent analysis.

3.2.1 Handling Missing Values

Missing data can significantly impact the performance and accuracy of machine-learning models (Emmanuel et al., 2021). Machine learning algorithms rely on complete and accurate data to make predictions and learn patterns. When datasets contain missing values, many machine learning algorithms may fail, leading to a lack of precision in statistical analysis and biased models that produce incorrect results. In this study, we will address missing values across both columns and rows to ensure the integrity of the dataset.

First, we identified the columns with missing values to provide a comprehensive view of the extent of missing data across the dataset. Based on this analysis, we performed the following steps:

1. ***Removal of Columns with High Missing Values***: Columns with more than 20% missing values were removed from the dataset. This threshold was chosen to balance retaining valuable information with minimising noise from incomplete data.
2. ***Row Removal in Nutrition Data***: Since this study focuses on clustering nutrition data, rows with missing values in key nutrition columns were removed to ensure the accuracy of the analysis.
3. ***Imputation of Missing Values***: For columns with fewer missing values, appropriate imputation methods were applied:

- ***Nutrition and Additive Number Columns***: Missing values were imputed with zero, as missing values in nutrition data typically indicate non-detection.
- ***Nutrition Score***: Using available nutrition data, the k-Nearest Neighbors (KNN) imputation was used to predict the nutrition score for products in France and the UK.
- ***Serving Size***: Missing values for serving size were imputed using the median serving size within each category of food products.

3.2.2 Data Validation and Outlier Removal

Invalid data is incomplete, incorrect, outdated, or irrelevant data. If appropriately addressed, valid data can significantly impact the performance of a machine-learning algorithm and lead to accurate predictions (Redman, 2018). For example, complete data can lead to biased statistical analysis and accurate parameter estimation. To overcome this, we will perform data validation and outlier detection to remove the irrelevant data.

3.2.2.1 Identifying and Removing Invalid Data

Invalid data refers to needs to be completed, corrected, outdated, or irrelevant data. The presence of such data can skew results and reduce the accuracy of machine learning models. We addressed invalid data by performing the following steps:

1. ***Identifying Invalid Data***: We examined summary statistics and visualisations to identify anomalies in the data. For example, some nutritional values fell outside the logical range of 0g to 100g, indicating potential data entry errors.

2. ***Removing Extreme and Negative Nutrition Values***: Nutritional values such as fat, carbohydrates, fibre, proteins, and salt showed unusually high maximum values, while sugars and fibre had negative minimum values. To ensure data integrity, we capped all individual nutrient values at 100g and a minimum of 0g. For energy, we set a maximum limit of 3766 kilojoules (kJ) and a minimum of 0kJ. This ensured that extreme values were effectively removed.

3. ***Sum Check for Macronutrients and Micronutrients***: We conducted a sum check of macronutrients (fat, carbohydrates, and proteins) and micronutrients (fibre, sodium, vitamins, and minerals) to ensure they did not collectively exceed 100g per 100g serving. Any sums exceeding 100g indicated potential errors in data reporting. We set a threshold of 105g to account for possible rounding errors. Data points exceeding this threshold were either removed or adjusted proportionally to maintain macronutrient ratios.

3.2.2.2 Addressing Nutrient Discrepancies

We scrutinised the relationships between various macronutrients and micronutrients to identify discrepancies. Specific checks included:

- The combined total of saturated fats, trans fats, and cholesterol should not exceed the total fat content.
- The amount of sugars should not exceed the total carbohydrates.

- Considering the conversion factor from sodium to salt (Na to NaCl), sodium should weigh less than salt.

Our analysis revealed significant discrepancies in the dataset. For instance, in 16,902 rows, the sum of saturated fat, trans fat, and cholesterol exceeded the total fat content. Similarly, sugars surpassed total carbohydrates in 16,088 rows, and sodium content exceeded its salt equivalent in 6,150 rows. These discrepancies were corrected by removing or adjusting the erroneous data.

3.2.2.3 Outlier Detection and Elimination

Outliers are data points that significantly differ from the majority of the data. While some outliers represent natural variations, others may result from measurement errors, data entry mistakes, or poor sampling. We used a combination of summary statistics and visualisations, such as violin plots, to identify outliers.

Approach for Handling Outliers:

1. ***Z-Score Transformation***: We used Z-scores to detect outliers for normally distributed data. Data points with Z-scores more significant than 3 or less than -3 were considered outliers and removed.

2. ***Median Absolute Deviation (MAD)***: We used MAD to detect outliers for skewed distributions. This method is robust to skewness and less sensitive to extreme values.

Outliers identified through these methods were removed to prevent them from distorting the analysis. This ensured that the remaining data accurately represented the underlying patterns and relationships.

3.2.3 Data Type Handling

Inconsistent data types can be caused by human error, missing values, typos, different sources, standards, or misalignment between the operation and data architecture teams (Data Headhunters, 2024). We involve encoding categorical variables and standardising numerical features to ensure uniformity in scale and representation across the dataset. This facilitates practical model training and prevents biases from disparate scales or inconsistent formats.

3.2.3.1 DateTime Type Conversion

The dataset includes a 'created_datetime' column, which is crucial for understanding the temporal aspects of the data. However, this column is initially in an object format, which is unsuitable for analysis. We converted the 'created_datetime' column from an object to a standard DateTime data type to facilitate time-based analyses and ensure consistency in date handling.

3.2.3.2 Label Encoding of Categories Column

The 'categories' column contains numerous categories, many of which have very few entries. To address this, we established a percentage threshold for category frequency, focusing on the most common categories:

1. ***Identify Top Categories***: We identified the top 15 categories by frequency, ensuring these represent the majority of the data.

2. ***Handling Null Values***: Entries without a designated main category were assigned as "Unknown" to differentiate them from less frequent but known categories.

3. ***Grouping Less Frequent Categories***: Categories with fewer than 2 entries were grouped into "Other," and smaller categories were manually grouped into larger, higher-level categories.

While initially considering label encoding, it was determined that one-hot encoding would be more appropriate for clustering tasks. Label encoding could impose an artificial order among categories, potentially misleading the clustering algorithm. One-hot encoding avoids this issue by representing each category as a separate binary feature.

We ultimately selected 17 category tags, including "Other" and "Unknown." This approach maintained model simplicity and interpretability while preserving essential information.

3.2.3.3 Handling Multiple Countries Column

The 'countries_en' column indicates the countries where each product is available. Directly converting these into separate columns would result in a sparse, high-dimensional dataset. To address this, we:

1. ***Explode the List***: The .explode() method was used to transform the list of countries in each row into separate rows, facilitating frequency analysis.

2. ***Group Low-Frequency Countries***: Prioritized the top 10 countries based on frequency and grouped the remaining countries into an "Other" category. The top 10 countries included the United States, France, Switzerland, Germany, Spain, the United Kingdom, Belgium, Russia, Italy, and Australia.

3. ***One-Hot Encoding***: Applied one-hot encoding to the resulting categories, reducing dimensionality while retaining relevant information.

3.2.3.4 One-Hot Encoding for Additives Tags Column

Similarly, for the 'additives_tags' column, which includes diverse additives, we performed one-hot encoding:

1. ***Identify Prevalent Additives***: Focused on the most common additives, ensuring essential information was retained.

2. ***Group Rare Additives***: Less common additives were grouped into an "Other" category, simplifying the model while maintaining key details.

3.2.3.5 Standardising Serving Sizes

The 'serving_size' column contained various formats, making direct comparisons challenging. To standardise serving sizes, we:

1. ***Inspect Unique Formats***: Identified the unique formats in the serving_size column, noting units mixed with remarks (e.g., "1 ONZ," "0.25 cup").

2. ***Extract Serving Sizes***: Regular expressions were used to extract the serving size strings, converting inconsistent entries to NaN.

3. ***Handle Missing Serving Sizes***: For standard categories (not labelled as "Other" or "Unknown"), imputed missing serving sizes using the median serving size of their respective categories. Entries in the "Other" and "Unknown" categories with missing serving sizes were removed to maintain data integrity.

3.2.4 Handling Duplicate Data

Removing duplicate entries from the dataset is crucial for ensuring the integrity and accuracy of the analysis (Kuan, 2022). Duplicate rows can skew results, lead to incorrect conclusions, and affect the performance of machine learning models. In this part, we will remove the duplicated rows and columns.

3.2.4.1 Identifying Duplicate Rows

Duplicate rows, where all feature values are identical, can occur due to various reasons, such as repeated data entry or merging datasets. To identify and remove these duplicates, we performed a thorough analysis. The dataset was examined using the Pandas library, which revealed 294 duplicate rows. These duplicates were then removed to ensure that each entry in the dataset represents a unique data point, preserving the integrity of the subsequent analysis.

3.2.4.2 Handling Redundant Information Columns

The dataset includes both nutrition_grade_fr and nutrition_score_fr_100g columns, which provide insights into the nutritional value of food products in France. These columns may contain redundant information. We examined their relationship through data visualisation techniques to decide whether one can be dropped.

A box plot was used to illustrate the relationship between the categorical nutrition_grade_fr and the numerical nutrition_score_fr_100g. The visualisation showed a clear progression in nutrition scores across different nutrition grades (from 'a' to 'e'), indicating that each grade category had a distinct range of nutrition scores with minimal overlap. This strong correlation suggested that the nutrition_score_fr_100g provided a more detailed and finely analysable numerical range.

Given the redundancy, we decided to drop the nutrition_grade_fr column, as the nutrition_score_fr_100g column encompassed the necessary nutritional information in a more precise numerical format. This decision helped streamline the dataset and reduced redundancy without losing critical information.

3.3 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a critical step in understanding the underlying structure and characteristics of the dataset. EDA involves summarising the main characteristics of the data, often using visual methods and is essential for uncovering patterns and checking assumptions. This section outlines the key steps and findings from the EDA of the Open Food Facts dataset.

3.3.1 Trend Over Time

In this section, we analyse the trend over time of various aspects related to product types and nutritional content to understand how the dataset evolves over the observed period.

3.3.1.1 Product Types Over Time

We start by examining how the number of products in different categories has evolved throughout the observed period.

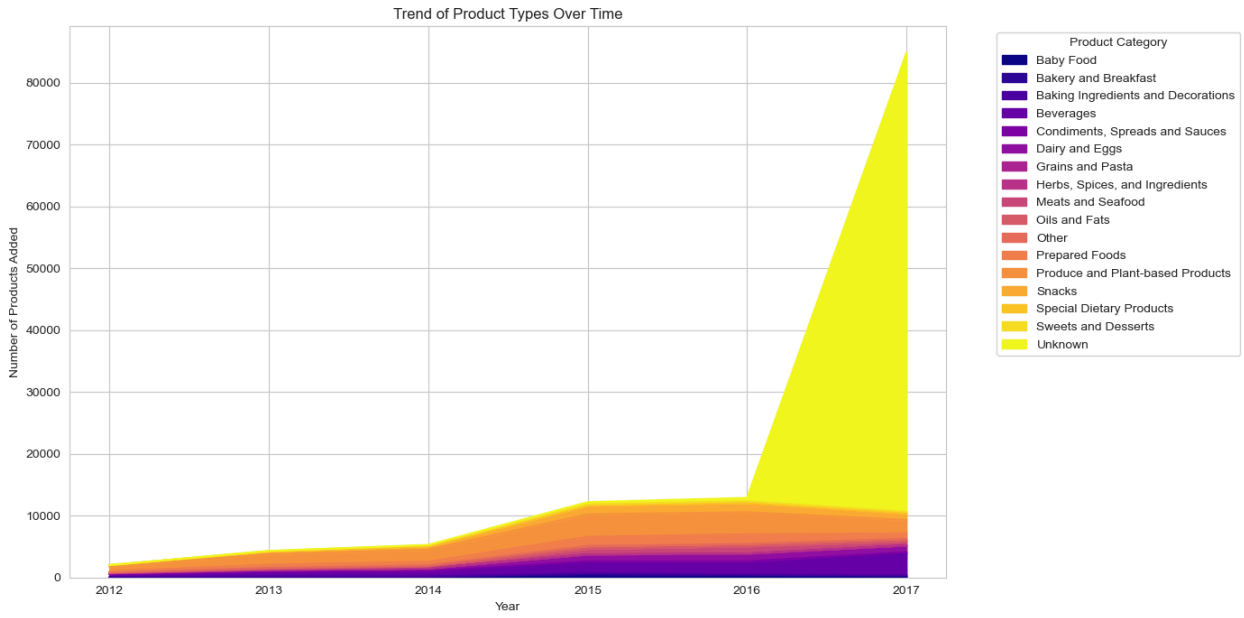


Figure 1. Trend of Product Types Over Time

The first chart visualises the trend of product types over time, highlighting how different categories have been added to the dataset year-over-year. Notably, the "Unknown" category showed a significant rise in 2016, suggesting a potential influx of new products that were not

properly categorised. To gain a clearer understanding, the second chart excludes the "Unknown" category to focus on other categories.

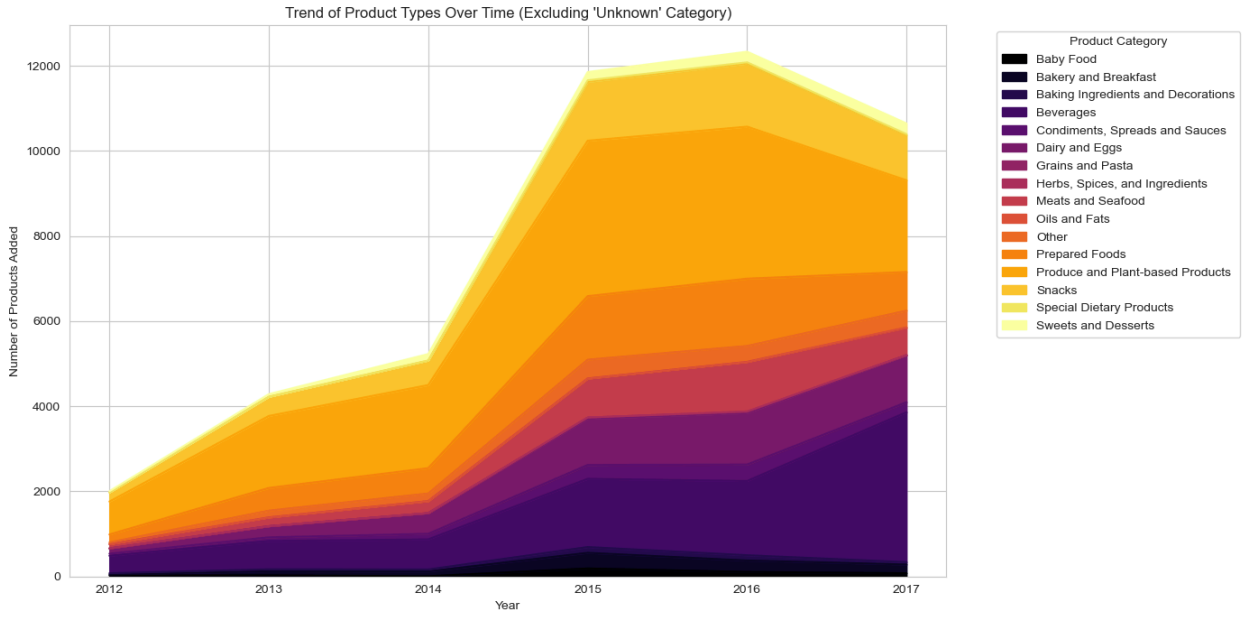


Figure 2. Trend of Product Types Over Time (Excluding 'Unknown' Category')

From the charts, it is evident that all categories show additions in 2014 and 2015. However, some categories like sweets and desserts, special dietary products, snacks, produce and plant-based products, bakery and breakfast items, and baby food show a slight dip in additional trends. Overall, the charts suggest a general increase in products offered between 2012 and 2017, with the rate of growth varying depending on the product category.

3.3.1.2 Nutritional Content Over Time

Next, we explore how the nutritional content of products has evolved over time, particularly focusing on macronutrients and micronutrients.

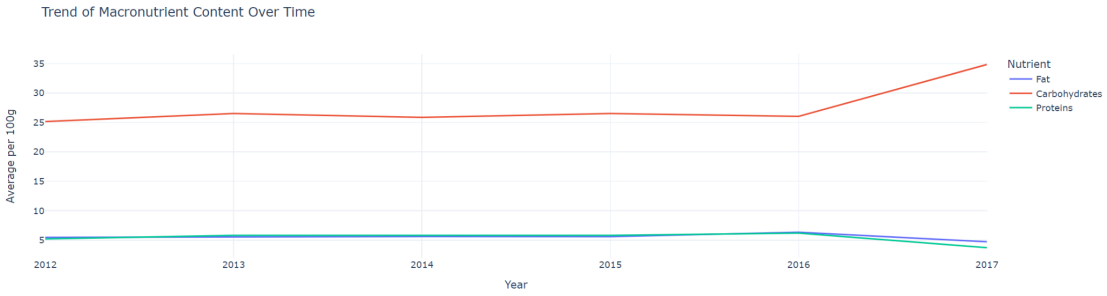


Figure 3. Trend of Macronutrient Over Time

The chart shows the average amount of fat, carbohydrates, and protein in a 100-gram product serving from 2012 to 2017. Carbohydrate content appears to be the most variable among these nutrients. It started high in 2012, dipped slightly in 2014, and then increased again in 2015 and 2016. Fat and proteins show similar patterns but fluctuate less. The overall range of macronutrients appears stable from 2012 to 2016, with slight increases in carbohydrates and decreases in fat and protein in 2017, indicating some changes in product composition.

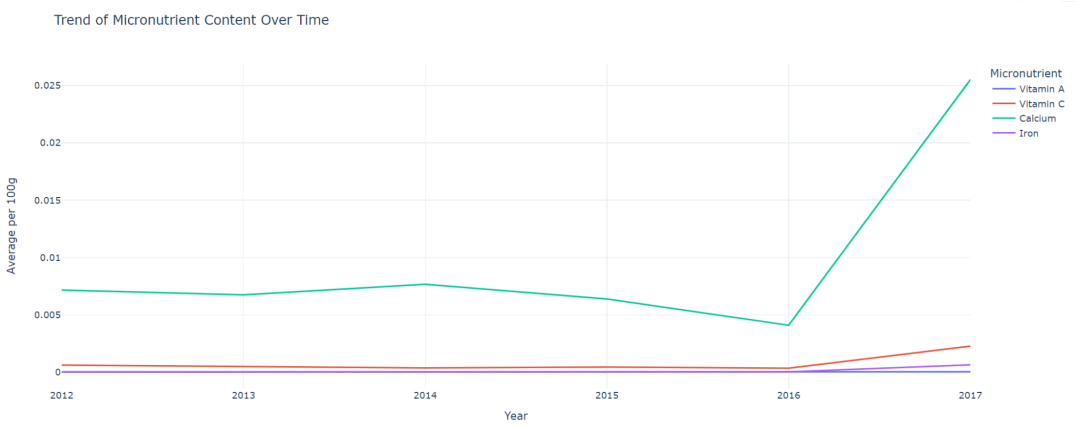


Figure 4. Trend of Micronutrient Over Time

The chart depicts the trend of micronutrient content in food items over the same period. Similar to macronutrients, the average micronutrient content appears consistent except for 2017, where all nutrients show an upward trend, especially calcium, which increases significantly.

3.3.2 Correlation Analysis of Numeric Columns

In this section, we conduct a correlation analysis using a heatmap to examine the relationships between various nutrients and other numeric columns present in the products under study.

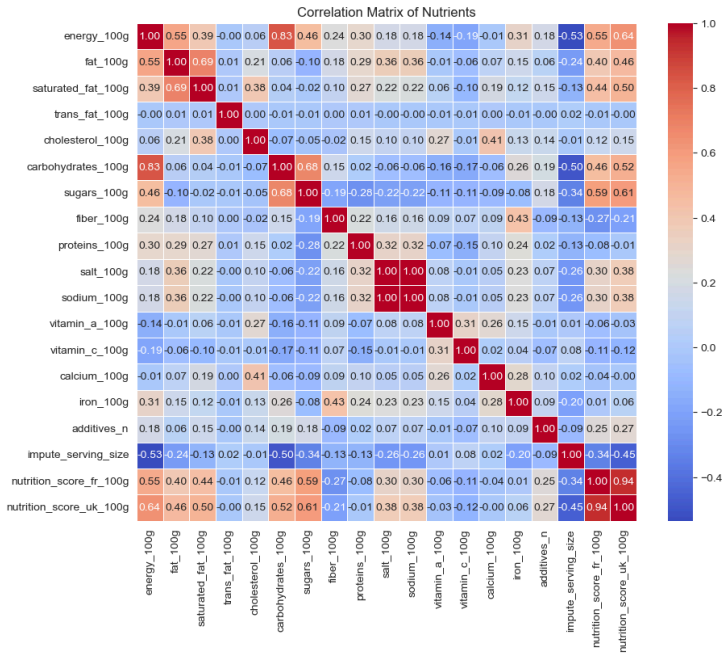


Figure 5. Correlation Heatmap of Numeric Columns in Dataset

A strong positive correlation was observed between energy content and both fat and carbohydrates, indicating that foods high in fat or carbohydrates also tend to have higher energy content. Sugars, a component of carbohydrates, likely contribute to this relationship. Fibre showed a moderate positive correlation with iron and carbohydrates, suggesting that high-fibre foods are also good sources of iron and carbohydrates, often pointing to plant-based foods.

A moderate positive correlation between proteins and fat indicates that foods such as meat or dairy products are generally higher in both nutrients. Nutrition scores in France and the UK were moderately positively correlated with energy, fat, saturated fat, carbohydrates, sugars, salt, and sodium, reflecting the nutritional composition's impact on overall healthiness.

Serving size showed a strong negative correlation with energy, carbohydrates, and nutrition scores, suggesting that foods with higher nutritional content have smaller serving sizes. The number of additives had little correlation with other numeric columns, indicating that additives are not strongly associated with the nutritional composition of food products.

3.3.3 Assessing Nutritional Profiles

This section encompasses discussions on various aspects of food and nutrition, including the health implications of snacks, the health benefits of high-fibre foods, and the impact of additives on nutritional metrics.

3.3.3.1 Are Foods High in Fiber Healthy?

To investigate whether foods high in fibre are healthy, we define "high fibre" foods based on a quantile threshold, compare their nutritional profiles with low-fibre foods, and utilise the nutrition scores (UK and France) as indicators of healthiness.

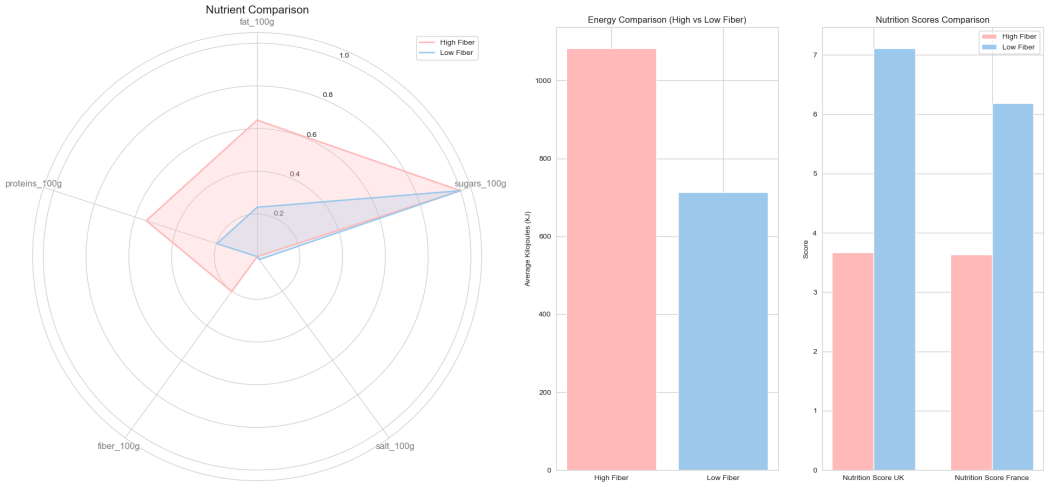


Figure 6. Nutritional Comparison Between High-Fiber and Low-Fiber Products

The radar chart shows that high-fibre foods generally have higher fat and protein content compared to low-fibre foods, which may be beneficial as certain fats and proteins are essential for health. High-fibre foods also have higher energy content but display better (lower) average nutrition scores in France and the UK, indicating healthier profiles overall. Low-fiber foods show slightly elevated salt levels, while both groups have similar sugar levels. This suggests that high-fibre foods may contribute more positively to a balanced diet due to their nutrient-rich composition and favourable nutrition scores.

3.3.3.2 Are Snacks Unhealthy Always?

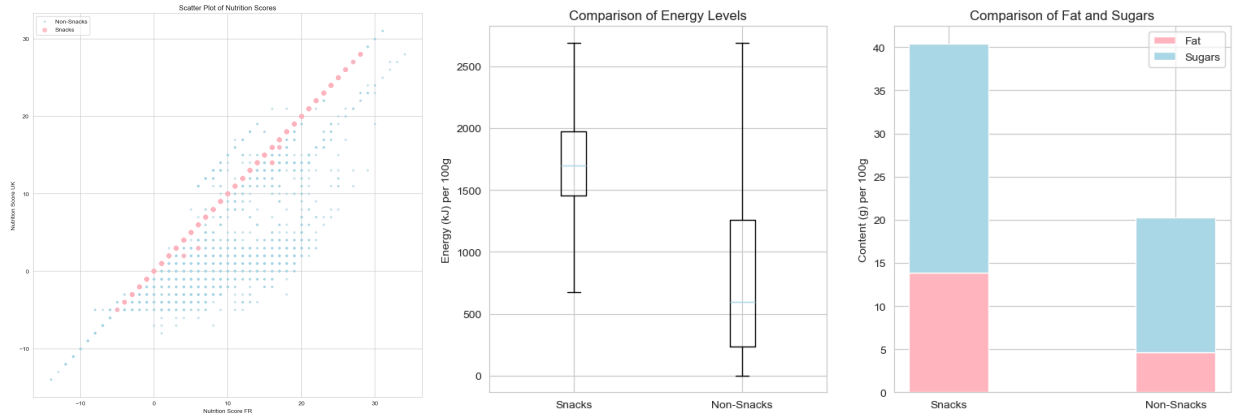


Figure 7. Nutritional Comparison Between Snacks and Non-Snacks

The boxplot indicates that snacks have a higher median energy level than non-snacks, reflecting their higher calorie density. The scatter plot shows that snacks typically receive poorer nutritional scores in both France and the UK, suggesting they are less healthy. The stacked bar chart reveals that snacks generally contain higher fat and sugar levels than non-snacks, contributing to their higher caloric content and potential negative health impact. This underscores the need for moderation and careful selection of snack foods to maintain a healthy diet.

3.3.3.3 Impact of Additives on Nutritional Metrics

This section explores how additives influence various nutritional metrics, shedding light on the potential implications for overall dietary health.

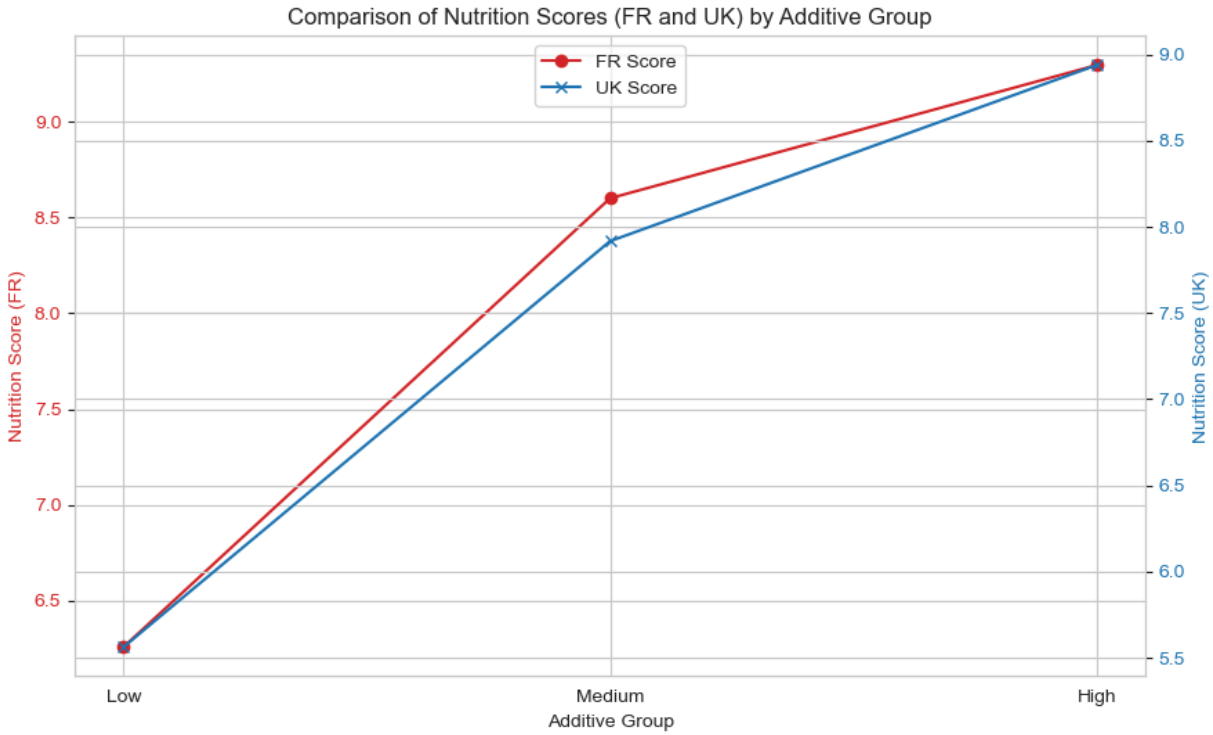


Figure 8. Nutritional Scores and Additive Levels Comparison

The dual-axis plot shows that products with fewer additives tend to have lower (better) nutrition scores, indicating better nutritional quality. As the number of additives increases, both French and UK nutrition scores rise, suggesting a decline in nutritional quality with higher additive counts. This highlights the importance of considering additive levels when evaluating the healthfulness of food products.

3.3.4 Geographical Distribution of Products and Nutritional Attributes

This section delves into the geographical distribution of food products within the dataset and examines how energy attributes vary across different regions.

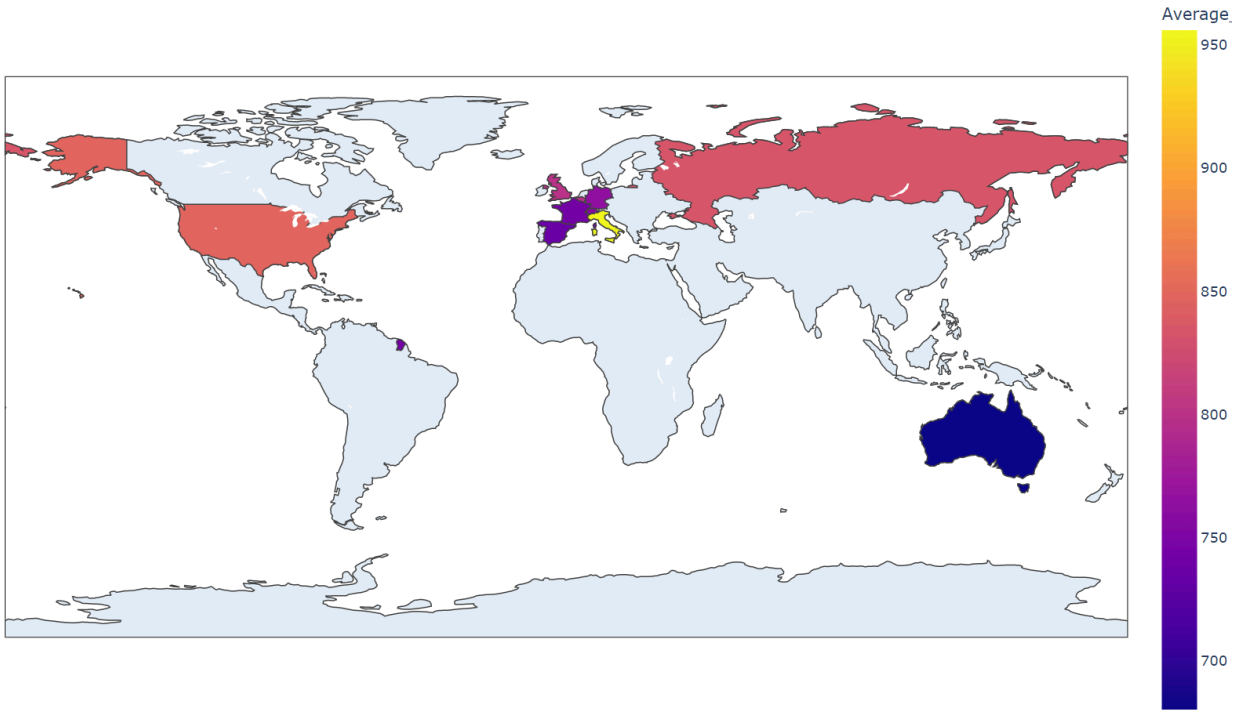


Figure 9. Geographical Distribution of Food Products and Average Energy Content

The map shows that the majority of food products originate from the United States and France. Italy and Germany have the highest average energy content per 100g, indicating a prevalence of high-calorie foods. The United Kingdom and the United States also show relatively high energy values. Conversely, France and Spain exhibit lower average energy contents, suggesting a focus on fresher and less processed foods. Russia and Australia have the lowest average energy contents, possibly reflecting stricter food regulations or cultural preferences for lower-calorie diets.

3.4 Model Implementation

In this section, we detail the methodology used for implementing the clustering algorithms, including data preparation, scaling, and the application of K-Means, Fuzzy C-Means, and DBSCAN clustering techniques.

3.4.1 Data Preparation

Prior to modelling, we prepared the dataset by selecting relevant features and scaling the data. The feature sets considered include basic nutritional information, nutritional scores, encoded additive columns, country columns, category columns, and imputed serving sizes. Each feature set was defined as follows:

- Feature Set 1: Basic nutritional information.
- Feature Set 2: All nutrition columns + Nutritional score.
- Feature Set 3: All nutrition columns + Encoded additive columns.
- Feature Set 4: All nutrition columns + Encoded country columns.
- Feature Set 5: All nutrition columns + Encoded category columns.
- Feature Set 6: All nutrition columns + Imputed serving size.

Given the different nature of these features, the "Unknown" category was separated for distinct clustering analysis.

3.4.2 Scaling the Data

Scaling is crucial to ensure that features contribute equally to the distance calculations used in clustering algorithms.

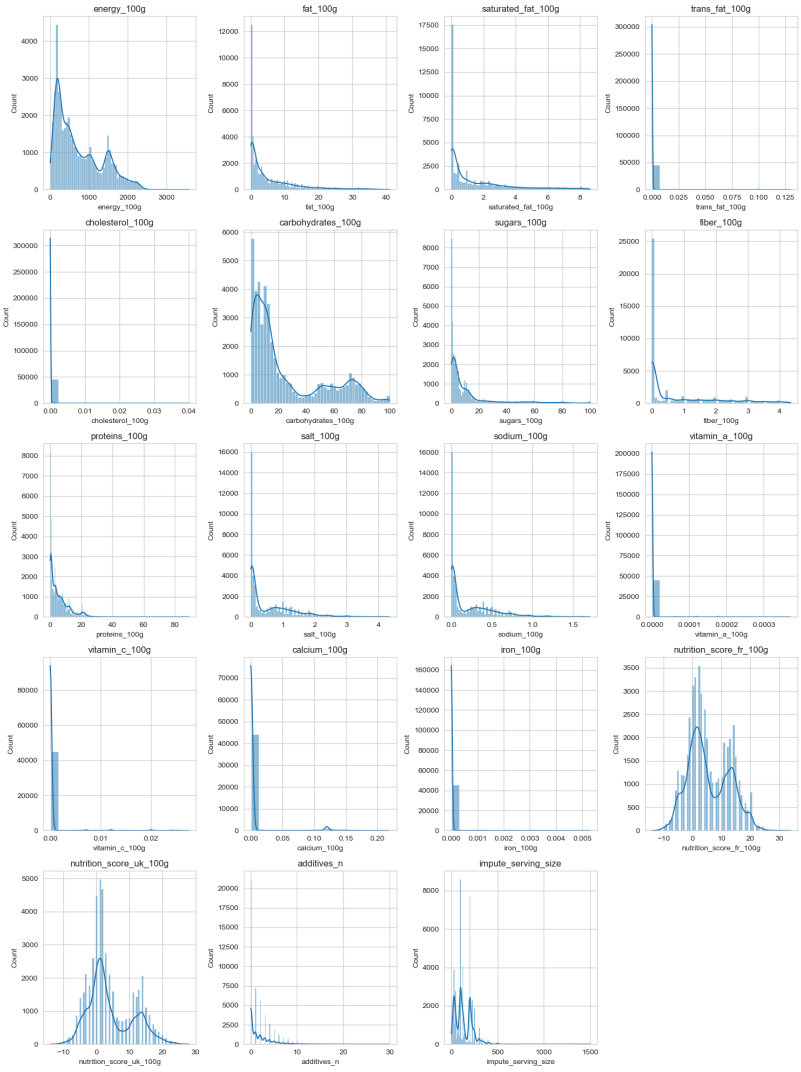


Figure 10. Histogram of Dataset Features Distribution

To ensure the features are comparable, we analysed the data distribution to choose appropriate scaling methods. The histogram analysis revealed that many variables were right-skewed with extreme outliers, particularly in nutritional columns such as fat_100g, saturated_fat_100g, and carbohydrates_100g. Therefore, the Robust Scaler was selected for the nutritional columns to mitigate the effect of outliers by using the median and interquartile range. For the nutrition score columns, the MinMax Scaler was chosen due to its effectiveness on data with moderate skewness.

3.5 Clustering Algorithms

This section details the clustering algorithms utilised in our research, encompassing the methodologies applied to each algorithm to ensure robust and comprehensive analysis.

3.5.1 K-Means Clustering

K-Means clustering is a partitioning method that aims to divide a set of n observations into k clusters, where each observation belongs to the cluster with the nearest mean. The methodology for applying K-Means clustering in this study is as follows:

1. **Initial Comparison**: We initially apply K-Means clustering with default settings (random initialisation with k clusters) to each feature set. The primary objective is identifying the feature set demonstrating the best initial clustering performance. Performance is assessed based on the Silhouette Score, which evaluates the consistency within and separation between clusters.
2. **Visualisation**: We employ Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE) to gain insights into the clustering results for dimensionality reduction and visualisation. PCA reduces the dataset to two principal components, providing a linear projection, while t-SNE offers a non-linear projection to better capture complex relationships within the data.
3. **Evaluation**: The initial clustering performance is further evaluated using the Davies-Bouldin Index and the Calinski-Harabasz Index. The Davies-Bouldin Index quantifies the average similarity ratio between each cluster and its most similar cluster. In contrast, the Calinski-Harabasz Index measures the ratio between-cluster dispersion to within-cluster dispersion.
4. **Optimal Clusters**: The Elbow Method determines the optimal number of clusters (k). This method involves plotting the sum of squared distances (inertia) against the number of clusters and identifying the "elbow point," where the rate of decrease sharply slows down, indicating the optimal k.

3.5.2 Fuzzy C-Means Clustering

Fuzzy C-Means (FCM) clustering is a soft clustering algorithm that allows each data point to belong to multiple clusters with varying degrees of membership. The methodology for FCM clustering in this study includes:

1. **Initial Comparison**: We apply FCM clustering with default settings to each feature set to identify the feature set with the best initial clustering performance. We compute the degree of membership of each data point in the clusters, and performance is assessed using the Silhouette Score.

2. **Visualisation**: Similar to K-Means, PCA and t-SNE are used to visualise the clustering results. This dual approach helps understand the structure of the clusters and the distribution of data points across them.
3. **Evaluation**: The clustering performance is evaluated using the Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index to assess the clustering quality comprehensively.
4. **Optimal Clusters**: The Fuzzy Coefficient determines the optimal number of clusters (*c*). This coefficient measures the fuzziness of the clustering process, guiding the selection of the optimal *c* that balances membership distribution and cluster separability.

3.5.3 DBSCAN Clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm that identifies clusters based on the density of data points. The methodology for applying DBSCAN in this study involves:

1. **Initial Comparison**: We initially apply DBSCAN with default settings (*eps*=0.5, *min_samples*=5) to each feature set. If the default parameters do not yield valid clustering results, adjustments are made (e.g., *eps*=0.8, *min_samples*=10) to improve the clustering performance.
2. **Visualisation**: PCA and t-SNE are utilised to visualise the clustering results, aiding in identifying clusters and noise points.
3. **Evaluation**: The clustering performance is evaluated using the Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index. These metrics comprehensively evaluate the clustering quality, including identifying outliers and noise points.

3.6 Hyperparameter Tuning

After determining the optimal number of clusters for each algorithm, hyperparameter tuning is conducted to enhance clustering performance. The specific tuning processes for each algorithm are as follows:

3.6.1 K-Means Clustering

For K-Means clustering, we tune the following hyperparameters:

- **Number of Initializations (n_init)**: The number of times the K-Means algorithm will be run with different centroid seeds. We explore a range of values [10, 15, 20, 25] to determine the most stable solution.
- **Maximum Iterations (max_iter)**: The maximum number of iterations for a single run of the K-Means algorithm. We test values [300, 400, 500] to ensure convergence.
- **Tolerance (tol)**: The relative tolerance regarding inertia to declare convergence. We experiment with values [1e-4, 1e-3, 1e-2] to balance precision and computational efficiency.

3.6.2 Fuzzy C-Means Clustering

For Fuzzy C-Means clustering, we tune the following hyperparameters:

- **Fuzziness (m)**: The fuzziness parameter determines the cluster overlap degree. We explore values from 1.5 to 2.5 in increments of 0.25.
- **Error Tolerance (epsilon)**: The convergence criterion for the iterative process. We test values [0.001, 0.005, 0.01] to ensure stable convergence.
- **Maximum Iterations (max_iter)**: The maximum number of iterations allowed. We explore values [100, 200, 500, 1000, 1500, 2000] to ensure adequate iteration for convergence.

3.6.3 DBSCAN Clustering

For DBSCAN clustering, we tune the following hyperparameters:

- **Epsilon (eps)** is the maximum distance between two samples for one to be considered in the neighbourhood of the other. We explore values from 0.6 to 0.95 in increments of 0.05.
- **Minimum Samples (min_samples)**: The number of samples in a neighbourhood for a point to be considered a core point. We test values from 10 to 20 to optimise cluster formation and noise identification.

3.7 Evaluation

The clustering performance evaluation is conducted before and after hyperparameter tuning to assess the improvements achieved. The following metrics are used for comprehensive evaluation:

- **Silhouette Score**: This metric measures the quality of clustering by calculating the mean silhouette coefficient for all samples, reflecting clusters' compactness and separation.
- **Davies-Bouldin Index**: This index evaluates the average similarity ratio between each cluster and its most similar cluster, where lower values indicate better clustering.
- **Calinski-Harabasz Index**: This index measures the ratio between cluster dispersion and within-cluster dispersion, with higher values indicating better-defined clusters.

4.0 Results

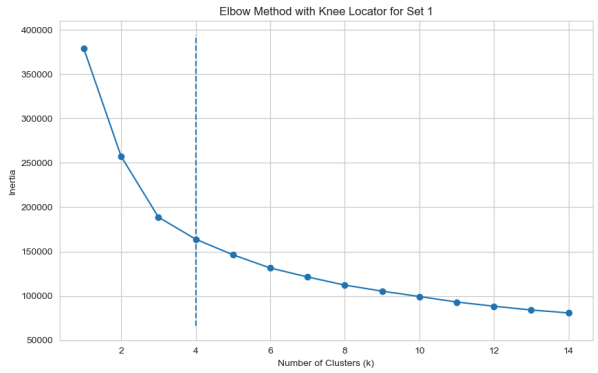
This section presents the results of our clustering analysis, feature set selection, cluster identification, visualisation and hyperparameter tuning. The results include an in-depth performance comparison across different feature sets, visualisation of the clusters using PCA and t-SNE, and evaluation using clustering metrics.

4.1 K-Means Clustering

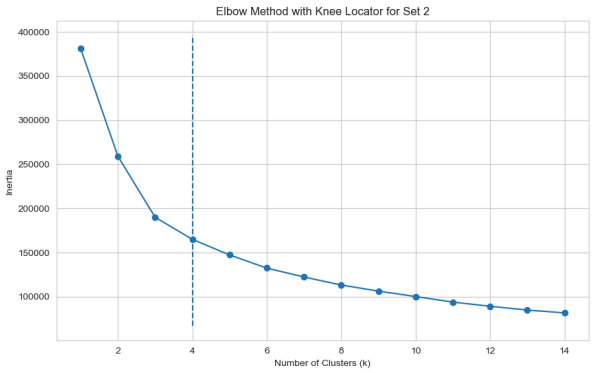
4.1.1 Feature Set Selection

In this study, we applied K-Means clustering to multiple feature sets to identify the most suitable set for clustering. Each feature set was evaluated to determine the optimal number of clusters (K) and the corresponding inertia scores. Inertia, which measures the sum of squared distances between data points and their nearest cluster centroid, was used to compare the compactness of the clusters across different feature sets.

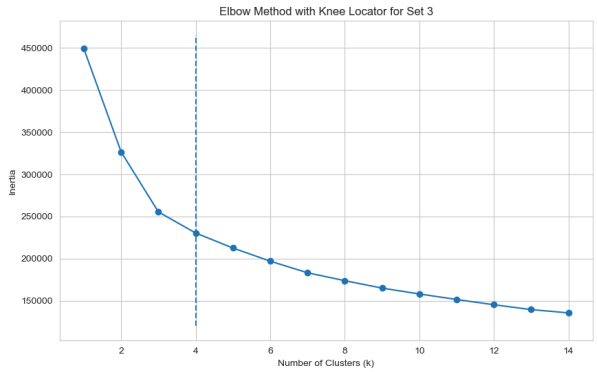
To illustrate our findings, Figures 11 to 16 display the clusters for Feature Sets 1 to 6, respectively. Each figure plots the inertia against the number of clusters (K), highlighting the optimal cluster count based on the Elbow Method. This method identifies the "elbow point" where the inertia decreases significantly before levelling off, indicating the optimal number of clusters.



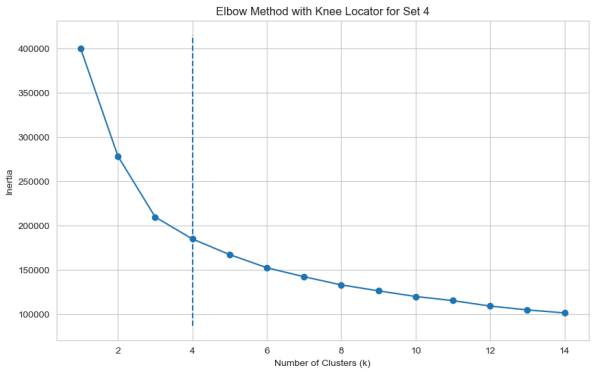
[Optimal k for Set 1 found: 4]



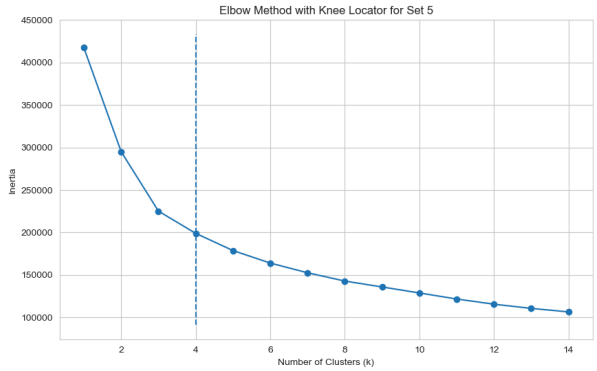
[Optimal k for Set 2 found: 4]



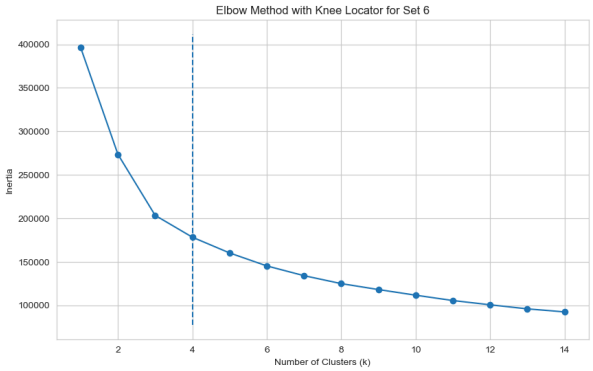
[Optimal k for Set 3 found: 4]



[Optimal k for Set 4 found: 4]



[Optimal k for Set 5 found: 4]



[Optimal k for Set 6 found: 5]

Figure 11-16. K-Means Elbow Method for Optimal Clusters in Feature Sets 1-6

Upon observation, all feature sets demonstrated an optimal cluster count of 4. However, the inertia scores varied significantly, indicating differences in the dataset's dispersion. Feature Set 1 exhibited the lowest inertia score, suggesting superior clustering performance in terms of compactness compared to other sets. The inertia scores for all feature sets are presented in Table 1.

Features Set	Number of Clusters	Inertia Score
1	4	163,922.11
2	4	165,012.20
3	4	230,345.94
4	4	184,971.93
5	4	198,976.69
6	4	178,412.57

Table 1. Inertia Scores for Different Feature Sets

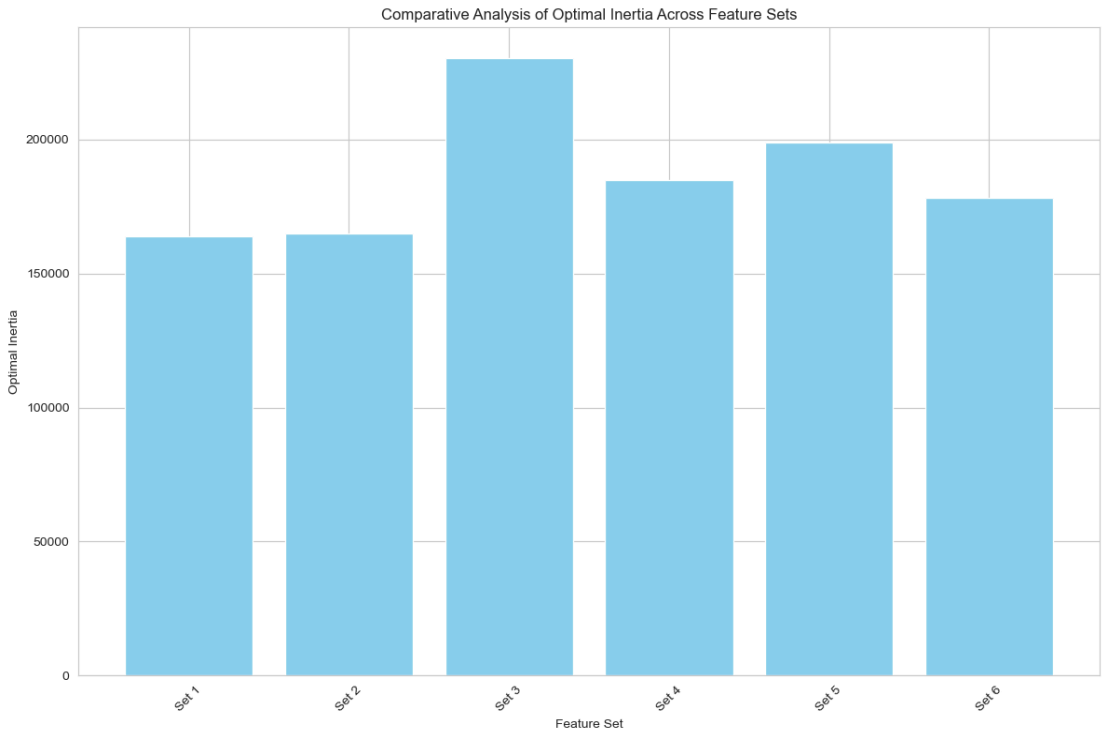


Figure 17. Comparison of Inertia for Optimal Number of k across Feature Sets 1-6

Figure 17 provides a comparative visualisation of the inertia scores for the optimal number of clusters (K) across Feature Sets 1 to 6. This comparison highlights the performance disparities among the feature sets, emphasising the superior compactness of Feature Set 1. Thus, Feature Set 1 will be selected for further analysis.

4.1.2 Cluster Identification and Visualisation

To visualise the clustering results, we employed PCA and t-SNE. These dimensionality reduction techniques provided insights into the data structure and cluster distribution in two-dimensional space.

PCA Visualisation

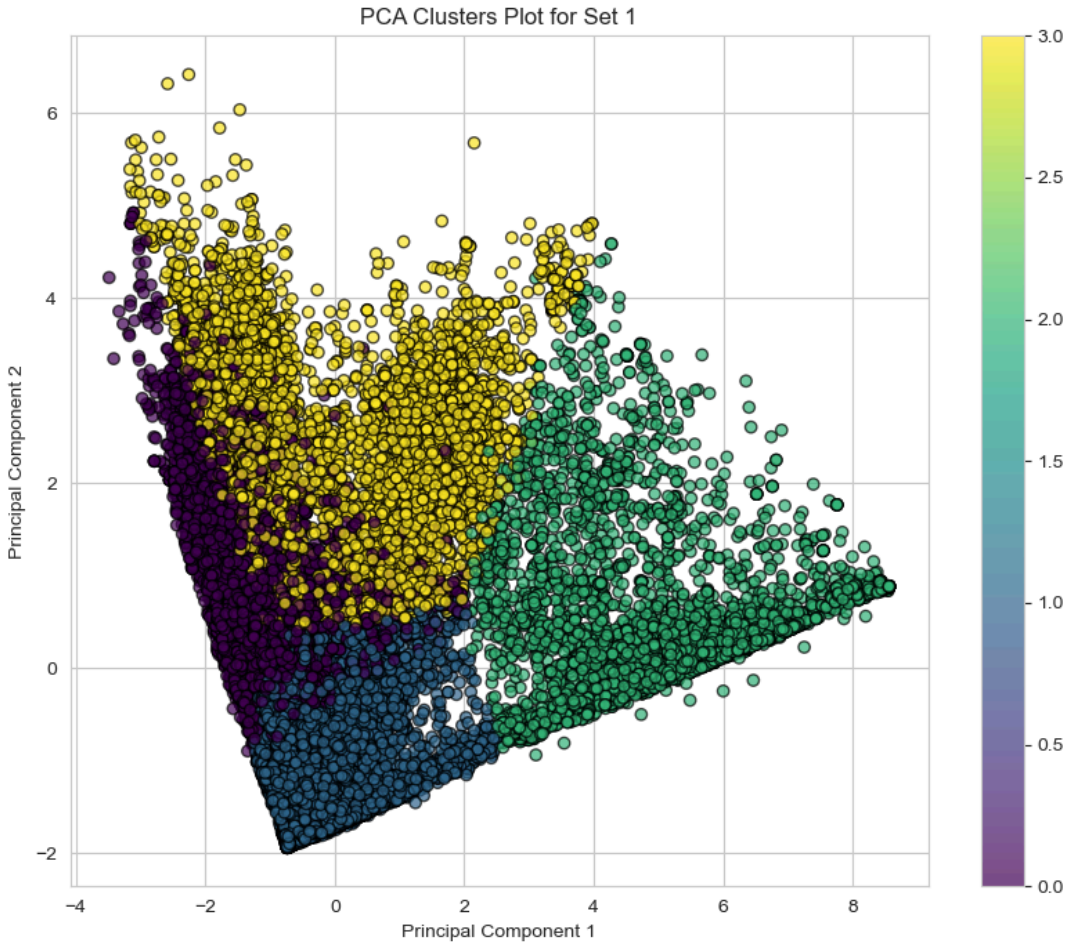


Figure 18. K-Means PCA Visualisation of Clusters

The PCA visualisation revealed several key observations. Firstly, a dense vertical accumulation of points around lower values of Principal Components 1 and 2 suggested a concentration of data points with similar values, likely capturing significant variance within the dataset. This dense line indicates that a substantial portion of the data exhibits similar characteristics, making it a prominent feature in the dataset's variance structure.

Secondly, the clusters displayed a spread-out formation. Cluster 1 (blue), located at the bottom left and extending to the centre, alongside the transition from purple (Cluster 0) to yellow (Cluster 3), indicated distinct nutritional profiles. The separate green cluster on the right represented a unique category of products with markedly different characteristics. This spread suggests that the dataset contains diverse groups with distinct attributes well captured by the clustering algorithm.

Finally, the colour gradient represented different clusters assigned by K-Means. The visual overlap in this two-dimensional reduction suggested that the clustering might not be well-defined in the reduced space, potentially requiring higher-dimensional analysis or alternative techniques like t-SNE for better distinction. The overlapping clusters indicate that while they are reasonably well-separated, some share similarities that make them less distinct in the reduced dimensional space.

t-SNE Visualisation

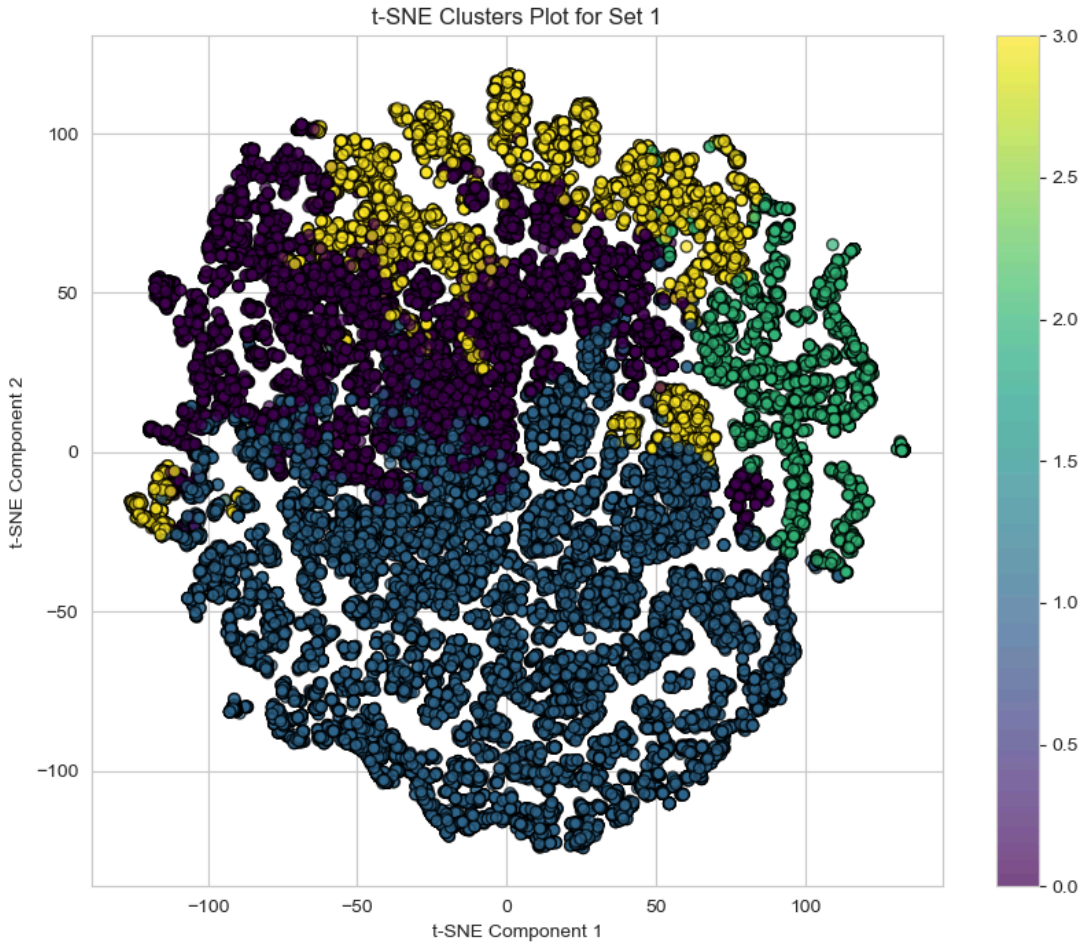


Figure 19. K-Means t-SNE Visualisation of Clusters

The t-SNE visualisation provided a more precise representation of the data structure than PCA, owing to its effectiveness in preserving local neighbourhood structures. The t-SNE plot highlighted that clusters were dispersed with some overlap, suggesting weak group cohesion and variability within the groups. Overlaps between clusters, particularly between purple and yellow and where green meets yellow, indicated shared overlapping features among specific product categories. These overlaps suggest that some product categories have inherent similarities that make their boundaries less distinct. Additionally, a few isolated points or small groups, particularly in cluster 3 (yellow), represented outliers with unique properties not shared by other data points. These outliers highlight unique items in the dataset that do not fit well into the main clusters.

Cluster Centroids Overview

The clusters' centroids provided insights into their characteristics. Table 2 summarises the key attributes of each cluster centroid.

Nutrient	**Cluster 0**	**Cluster 1**	**Cluster 2**	**Cluster 3**
Energy (kJ)	841.03	354.2	1359.12	1468.67
Fat (g)	6.04	1.6	2.68	19.64
Saturated Fat (g)	1.53	0.55	0.94	4.9
Trans Fat (g)	0.000277	0.000077	0.000188	0.000225
Cholesterol (g)	0.000093	0.000048	0.000017	0.000125
Carbohydrates (g)	24.17	14.96	72.2	34.98
Sugars (g)	2.99	6.48	62.44	11.72
Fibre (g)	1.24	0.54	0.61	1
Proteins (g)	11.47	2.31	2.03	8.06
Salt (g)	1.25	0.17	0.16	1.06
Sodium (g)	0.49	0.07	0.06	0.42
Vitamin A (g)	0.00000226	0.00000214	1.34E-07	8.14E-07
Vitamin C (g)	0.000129	0.00076	0.000022	0.000052
Calcium (g)	0.001827	0.009672	0.000586	0.001321

Iron (g)	0.000061	0.000013	0.000018	0.000043

Table 2. K-Means Summary of Cluster Centroids

The centroids of the clusters provided insights into the characteristics of each cluster. Cluster 0 was characterised as "Moderately Balanced Foods," with moderate energy and fat content, low sugars, higher proteins, moderate fibre, and higher salt. Cluster 1, described as "Light and Low-Fat Foods," had low energy and fat, moderate sugars, low proteins and fibre, and very low salt. Cluster 2, termed "High Energy and High Sugar Foods," exhibited high energy, low fat, high sugars, low proteins and fibre, and very low salt. Cluster 3, labelled "High Fat and Protein-Rich Foods," featured high energy, fat, moderate sugars, proteins, fibre, and salt.

Clustering Evaluation Metrics

The clustering performance was evaluated using the Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index. These metrics provided a comprehensive assessment of the clustering quality. The evaluation results are presented in Table 3.

Metric	Score	Interpretation
Silhouette Score	0.336	Moderate separation between clusters. While clusters are not perfectly distinct, they are reasonably well-defined, indicating some overlapping characteristics.
Davies-Bouldin Index	1.370	Moderate clustering quality. Lower values are better, suggesting that clusters are not tightly compact nor highly distinct.
Calinski-Harabasz Index	20,239.057	This index measures the ratio of between-cluster dispersion to within-cluster dispersion. A higher value indicates better-defined clusters, but it can be affected by dataset size.

Table 3. K-Means Clustering Evaluation Metrics

The Silhouette Score of 0.336 indicated moderate separation between clusters, suggesting that while clusters are not perfectly distinct, they are reasonably well-defined. The Davies-Bouldin Index of 1.370 reflected moderate clustering quality, with lower values indicating better-defined clusters. The Calinski-Harabasz Index of 20,239.057 measured the ratio between cluster dispersion and within-cluster dispersion, with higher values indicating better-defined clusters. However, this metric can be affected by the size of the dataset. Larger datasets tend to have higher index values because more data points can artificially inflate the between-cluster dispersion relative to the within-cluster dispersion.

4.1.3 Hyperparamter Tuning

This section focuses on the hyperparameter tuning process for K-Means clustering to refine the clustering performance. Based on previous analysis, the initial optimal number of clusters (K) was determined to be 4. The hyperparameters tuned included the number of initialisations (n_init), maximum iterations (max_iter), and tolerance (tol). The ranges for these hyperparameters were as follows:

- n_init: [10, 15, 20, 25]
- max_iter: [300, 400, 500]
- tol: [1e-4, 1e-3, 1e-2]

The tuning process involved iterating over all combinations of these hyperparameters to identify the best set that minimised the inertia. The results of the hyperparameter tuning are summarised in Table 4.

Hyperparameter	Optimal Value
Number of Clusters (n_clusters, k)	4
Number of Initialisations (n_init)	25
Maximum Iterations (max_iter)	300
Error Tolerance (tol)	1e-4

Table 4. Best Hyperparameters for K-Means Clustering

The optimal hyperparameters resulted in an inertia of 163,921.93, which is a marginal improvement over the default settings. This suggests that the clustering results are relatively stable and not highly sensitive to changes in these hyperparameters. The stability is further confirmed by the minimal changes observed in the clustering evaluation metrics and the cluster centroids.

4.1.4 Performance Comparison

To evaluate the effectiveness of the hyperparameter tuning, we compared the performance metrics and cluster centroids before and after tuning. The clustering evaluation metrics used were the Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index. The results are presented in Figure 20 and Table 5.

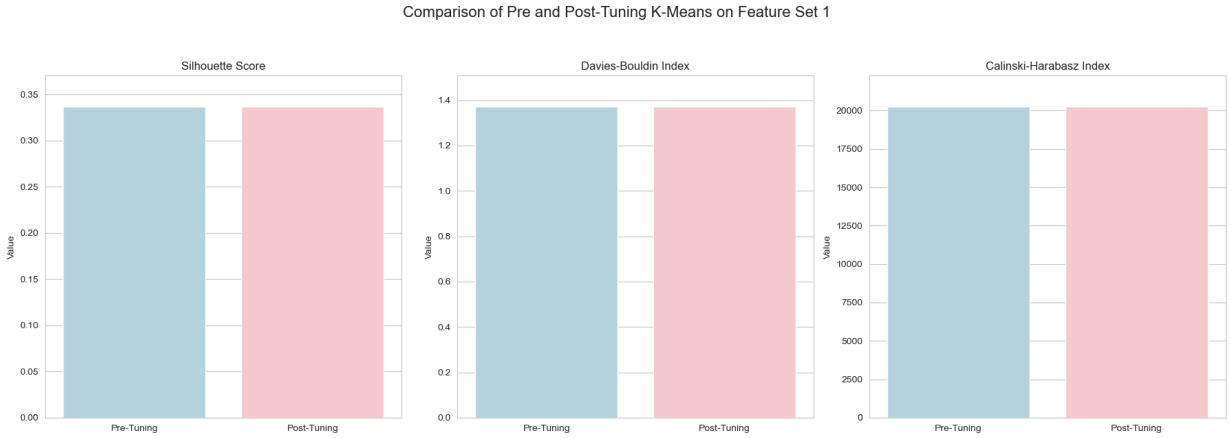


Figure 20. K-Means Clustering Evaluation Metrics Before and After Tuning

Metric	Pre-Tuning	Post-Tuning	Difference
Silhouette Score	0.336	0.3365	+0.0005
Davies-Bouldin Index	1.370	1.371	+0.001
Calinski-Harabasz Index	20239.057	20239.074	+0.017

Table 5. K-Means Clustering Evaluation Metrics Before and After Tuning

The changes in the evaluation metrics were negligible, indicating that hyperparameter tuning did not significantly improve the clustering quality. This suggests that the default or previously used parameters were already near optimal for this dataset, and further tuning of n_init and max_iter provided marginal benefits.

We also visualised the post-tuning clustering results using PCA and t-SNE, shown in Figures 21 and 22, respectively.

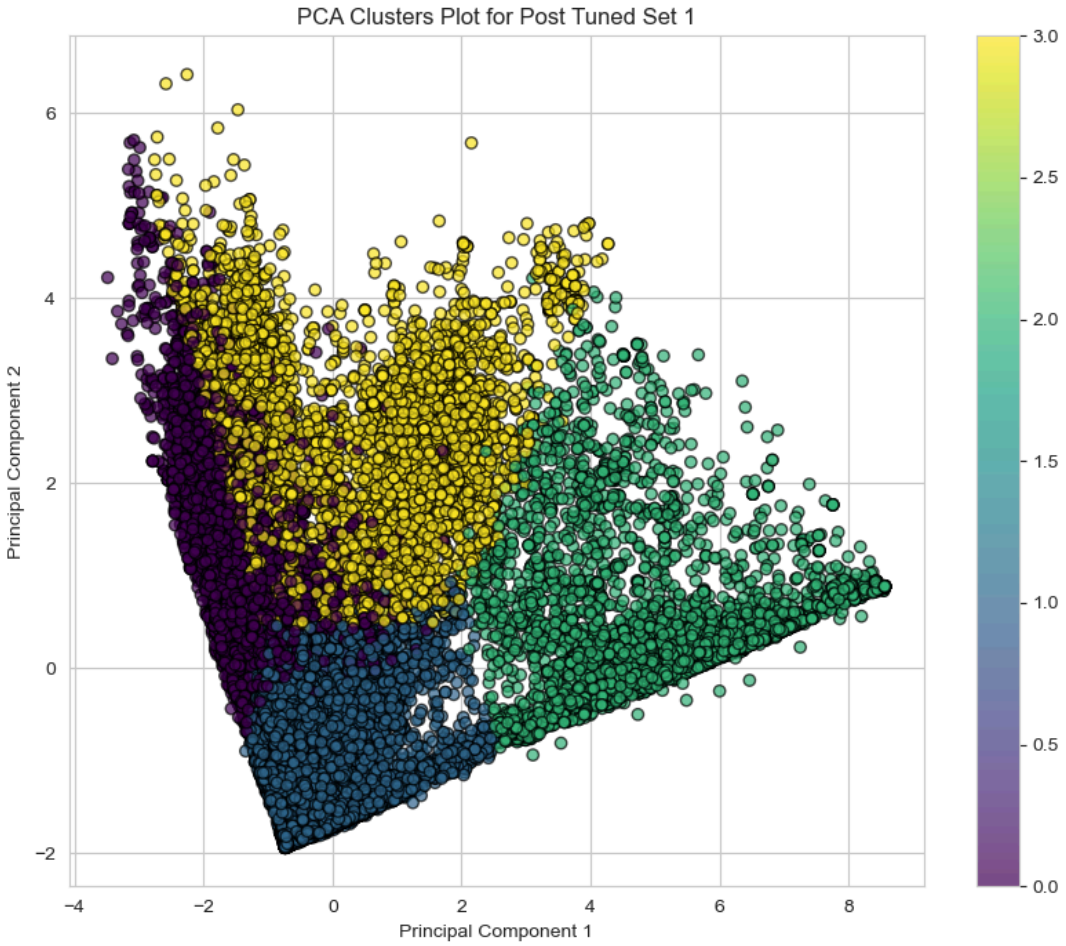


Figure 21. K-Means PCA Visualization of Clusters After Hyperparameter Tuning

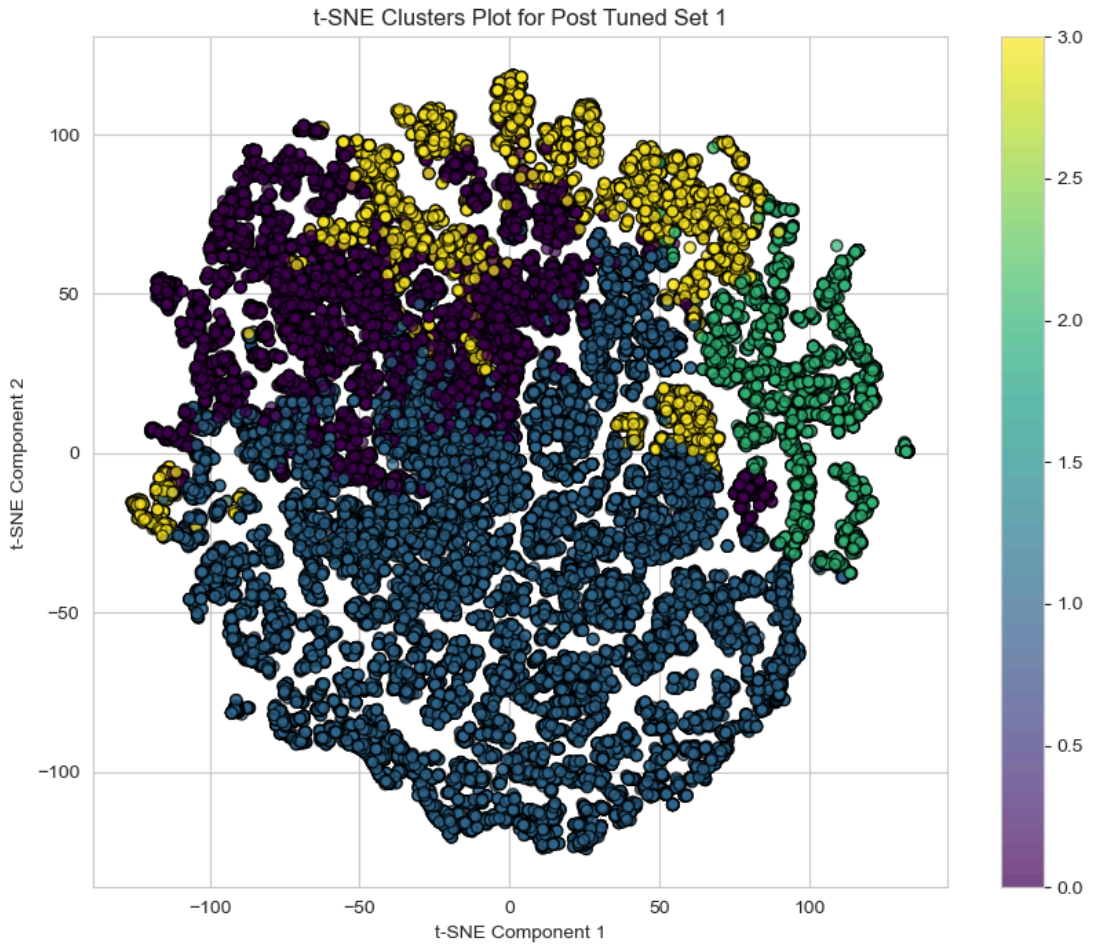


Figure 22. K-Means t-SNE Visualization of Clusters After Hyperparameter Tuning

From the visualisations, the post-tuning clustering results are almost identical to the pre-tuning clusters. This suggests that the K-Means algorithm converged to a stable solution with the optimal number of clusters (K = 4).

Comparison of Cluster Centroids Before and After Tuning

To further analyse the impact of hyperparameter tuning, we compared the clusters' centroids before and after tuning. The comparison is presented in Tables 6-9.

Nutrient	Pre-Tuning	Post-Tuning	Difference
Energy (kJ)	841.03	796	-45.03
Fat (g)	6.04	7.24	1.2
Saturated Fat (g)	1.53	1.92	0.39
Trans Fat (g)	0.000277	0.000221	-0.000056
Cholesterol (g)	0.000093	0.000107	0.000014
Carbohydrates (g)	24.17	23.8	-0.37
Sugars (g)	2.99	2.93	-0.06
Fibre (g)	1.24	0.92	-0.32
Proteins (g)	11.47	12.11	0.64
Salt (g)	1.25	1.53	0.28
Sodium (g)	0.49	0.54	0.05
Vitamin A (g)	0.00000226	0.0000022	-6E-08
Vitamin C (g)	0.000129	0.000076	-0.000053
Calcium (g)	0.001827	0.001621	-0.000206
Iron (g)	0.000061	0.000047	-0.000014

Table 6. Cluster 0 Centroid Comparison

Nutrient	Pre-Tuning	Post-Tuning	Difference
Energy (kJ)	354.2	430	75.8
Fat (g)	1.6	1.83	0.23

Saturated Fat (g)	0.55	0.58	0.03
Trans Fat (g)	0.000077	0.00012	0.000043
Cholesterol (g)	0.000048	0.000049	0.000001
Carbohydrates (g)	14.96	18.87	3.91
Sugars (g)	6.48	6.06	-0.42
Fibre (g)	0.54	0.73	0.19
Proteins (g)	2.31	3.08	0.77
Salt (g)	0.17	0.2	0.03
Sodium (g)	0.07	0.08	0.01
Vitamin A (g)	0.00000214	0.00000225	1.1E-07
Vitamin C (g)	0.00076	0.000659	-0.000101
Calcium (g)	0.009672	0.008241	-0.001431
Iron (g)	0.000013	0.000015	0.000002

Table 7. Cluster 1 Centroid Comparison

Nutrient	Pre-Tuning	Post-Tuning	Difference
Energy (kJ)	1359.12	1348	-11.12
Fat (g)	2.68	2.3	-0.38
Saturated Fat (g)	0.94	0.87	-0.07
Trans Fat (g)	0.000188	0.000192	0.000004
Cholesterol (g)	0.000017	0.000018	0.000001
Carbohydrates (g)	72.2	72.45	0.25
Sugars (g)	62.44	62.85	0.41
Fibre (g)	0.61	0.59	-0.02

Proteins (g)	2.03	1.96	-0.07
Salt (g)	0.16	0.16	0
Sodium (g)	0.06	0.06	0
Vitamin A (g)	0.000000134	0.000000137	3E-09
Vitamin C (g)	0.000022	0.000023	0.000001
Calcium (g)	0.000586	0.000582	-0.000004
Iron (g)	0.000018	0.000018	0

Table 8. Cluster 2 Centroid Comparison

Nutrient	Pre-Tuning	Post-Tuning	Difference
Energy (kJ)	1468.67	1518	49.33
Fat (g)	19.64	19.93	0.29
Saturated Fat (g)	4.9	4.86	-0.04
Trans Fat (g)	0.000225	0.000241	0.000016
Cholesterol (g)	0.000125	0.000115	-0.00001
Carbohydrates (g)	34.98	37.72	2.74
Sugars (g)	11.72	13.22	1.5
Fibre (g)	1	1.07	0.07
Proteins (g)	8.06	7.63	-0.43
Salt (g)	1.06	0.95	-0.11
Sodium (g)	0.42	0.37	-0.05
Vitamin A (g)	0.000000814	0.000000787	-2.7E-08
Vitamin C (g)	0.000052	0.000052	0
Calcium (g)	0.001321	0.001346	0.000025

Iron (g)	0.000043	0.000045	0.000002

Table 9. Cluster 3 Centroid Comparison

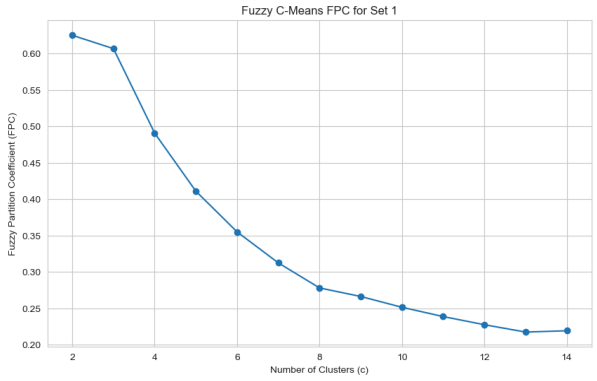
Overall, the changes in centroids before and after tuning are generally minimal. The differences in nutrient values are slight, indicating that the adjustments to hyperparameters did not lead to substantial shifts in the centroids of the clusters. This further supports the conclusion that the model had already converged to a stable solution with the initial settings.

In summary, the hyperparameter tuning of the K-Means algorithm resulted in slight improvements in clustering quality, as indicated by the evaluation metrics. The visualisations suggest subtle refinements in cluster memberships and structures, while the centroid comparisons show only minor changes in nutrient values. These findings imply that the initial hyperparameter settings were already close to optimal, and the tuning process provided marginal enhancements to the clustering results.

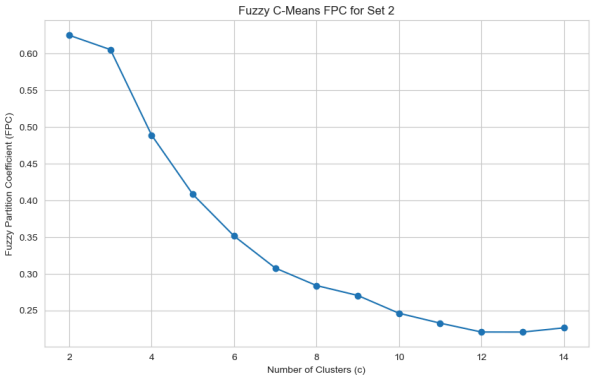
4.2 Fuzzy C- Means Clustering

4.2.1 Feature Set Selection

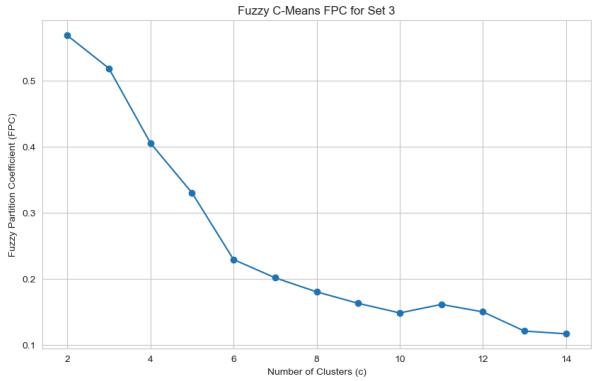
In our analysis, we applied Fuzzy C-Means (FCM) clustering to multiple feature sets to determine the most suitable set for clustering. The optimal number of clusters (c) for each feature set was identified using the Elbow Method, and the Fuzzy Partition Coefficient (FPC) was calculated to evaluate the degree of fuzziness in the cluster assignments. The FPC values range between 0 and 1, with higher values indicating clearer, better-separated clusters.



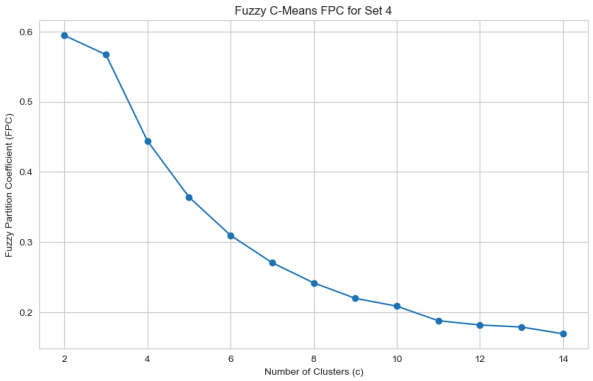
[Optimal c for Set 1 found: 2]



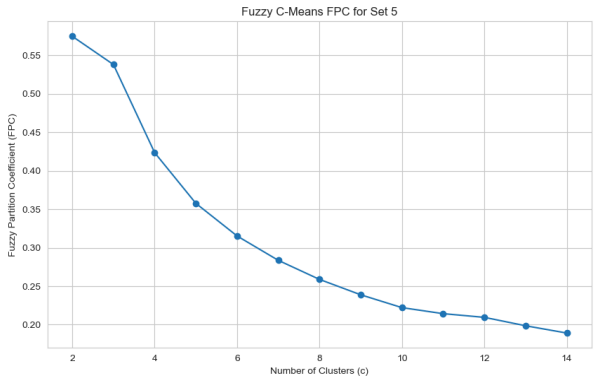
[Optimal c for Set 2 found: 2]

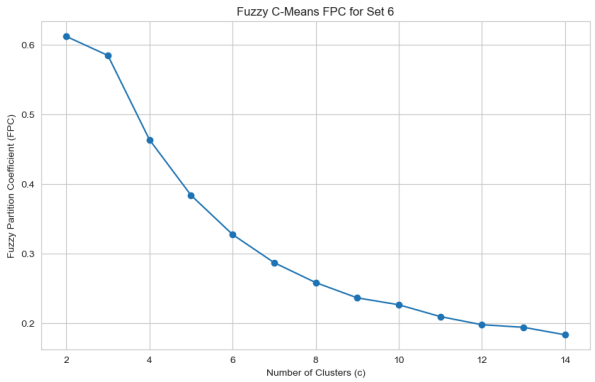


[Optimal c for Set 3 found: 2]



[Optimal c for Set 4 found: 2]





[Optimal c for Set 5 found: 2]

[Optimal c for Set 6 found: 2]

Figure 23-28. Fuzzy C-Menas Elbow Method for Optimal Clusters in Feature Sets 1-6

All feature sets showed an optimal cluster count of 4. However, the FPC values are quite similar, with little variation. Feature Set 1 had the highest FPC value of 0.6253 among all feature sets, indicating the clearest separation between clusters. The FPC values for each feature set are presented in Table 10.

Features Set	Number of Clusters	Inertia Score
1	2	0.6253
2	2	0.6248
3	2	0.5685
4	2	0.5946
5	2	0.5749
6	2	0.6123

Table 10. Fuzzy Partition Coefficients (FPC) for Different Feature Sets

The observation of the FPC values indicates that while there is some distinct separation between the two clusters in each feature set, the separation isn't highly defined. The selected Feature Set 1, with the highest FPC value, was used to visualise the clusters and validate the clustering results.

4.2.2 Cluster Identification and Visualisation

Similar to K-Means clustering, we visualised the clusters obtained from Fuzzy C-Means clustering using PCA and t-SNE. However, a key difference in FCM is that it provides a membership matrix rather than hard clustering labels, representing the degree of membership each data point has to each cluster.

PCA Visualisation

The PCA visualisation for Feature Set 1, shown in Figure 29, utilises a gradient from green (indicating low membership) through white to pink (representing high membership). The deeper colours suggest a strong association with respective clusters, while lighter colours indicate mixed memberships. The PCA clusters plot, shown in Figure 30, converts the fuzzy membership matrix into hard labels by assigning each data point to the cluster for which it has the highest membership. This simplifies the visualisation but loses some nuances of the fuzzy memberships.

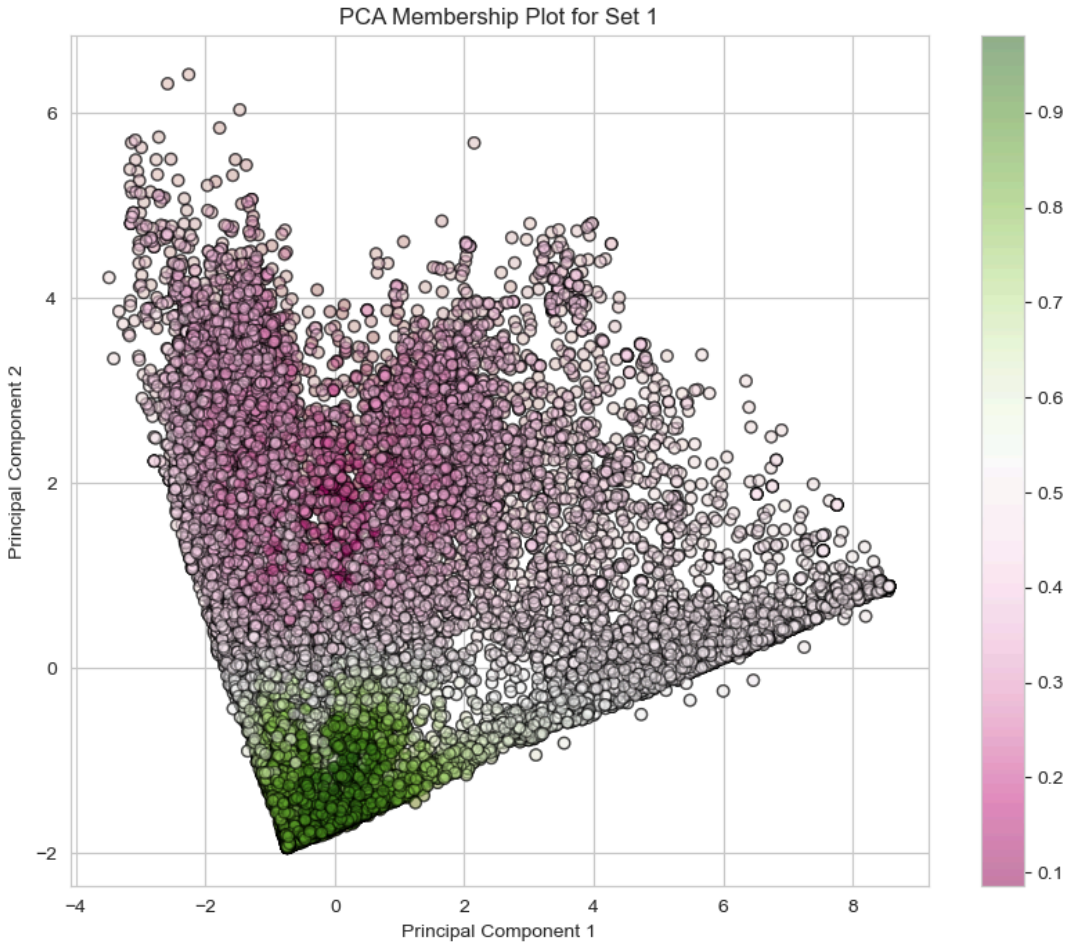


Figure 29. Fuzzy C-Means PCA Membership Plot for Feature Set 1

The PCA membership plot reveals dense and distinct clusters at the top and bottom with deep green and pink colours, indicating a strong association with respective clusters. The middle section of the plot shows areas with mixed colours, indicating regions where data points have significant but not exclusive association with clusters. This gradient reflects the fuzzy nature of FCM, capturing subtleties in data that hard clustering might overlook.

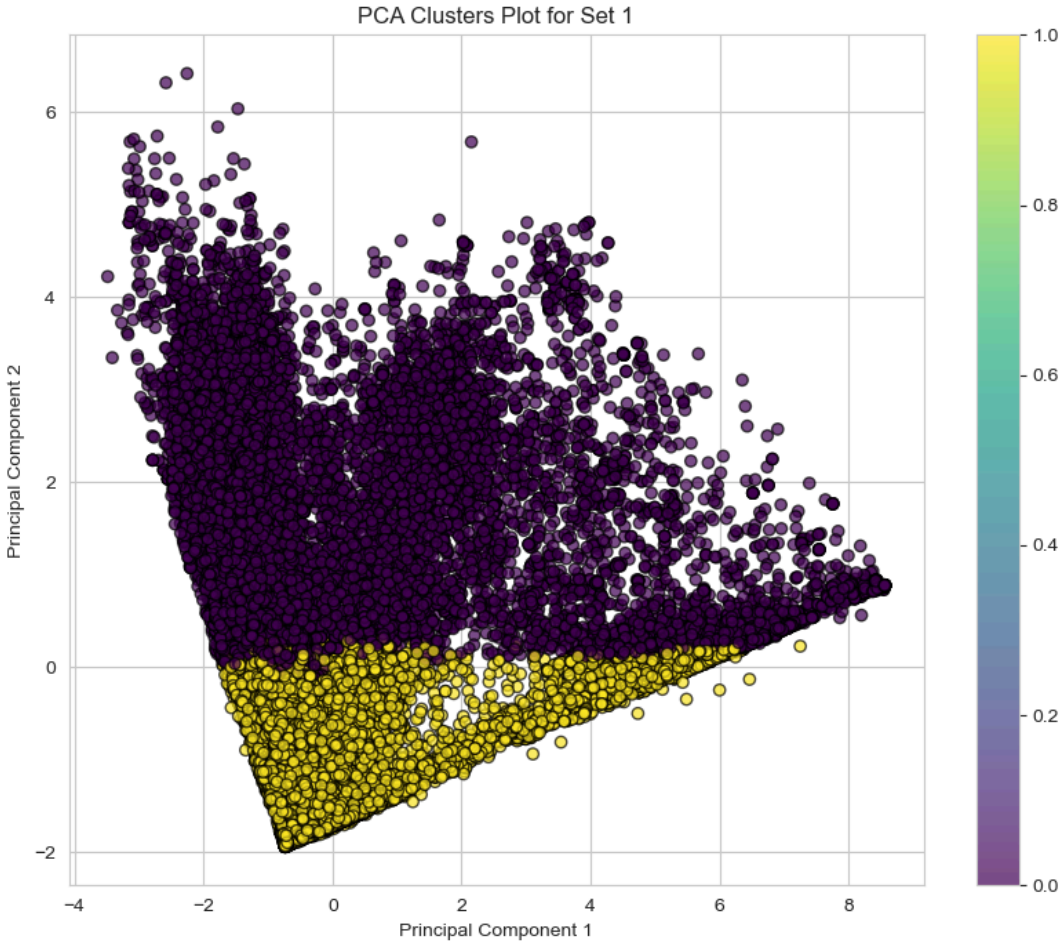


Figure 30. Fuzzy C-Means PCA Clusters Plot for Feature Set 1

The PCA clusters plot demonstrates a distinct purple cluster at the top left and a yellow cluster spreading from the bottom left to the top right. The separation line is relatively sharp, though there is some overlap, particularly around the centre. The yellow cluster appears compact, suggesting tighter cohesion among its members, while the purple cluster is more spread out, indicating a wider variety of data points grouped together.

t-SNE Visualisation

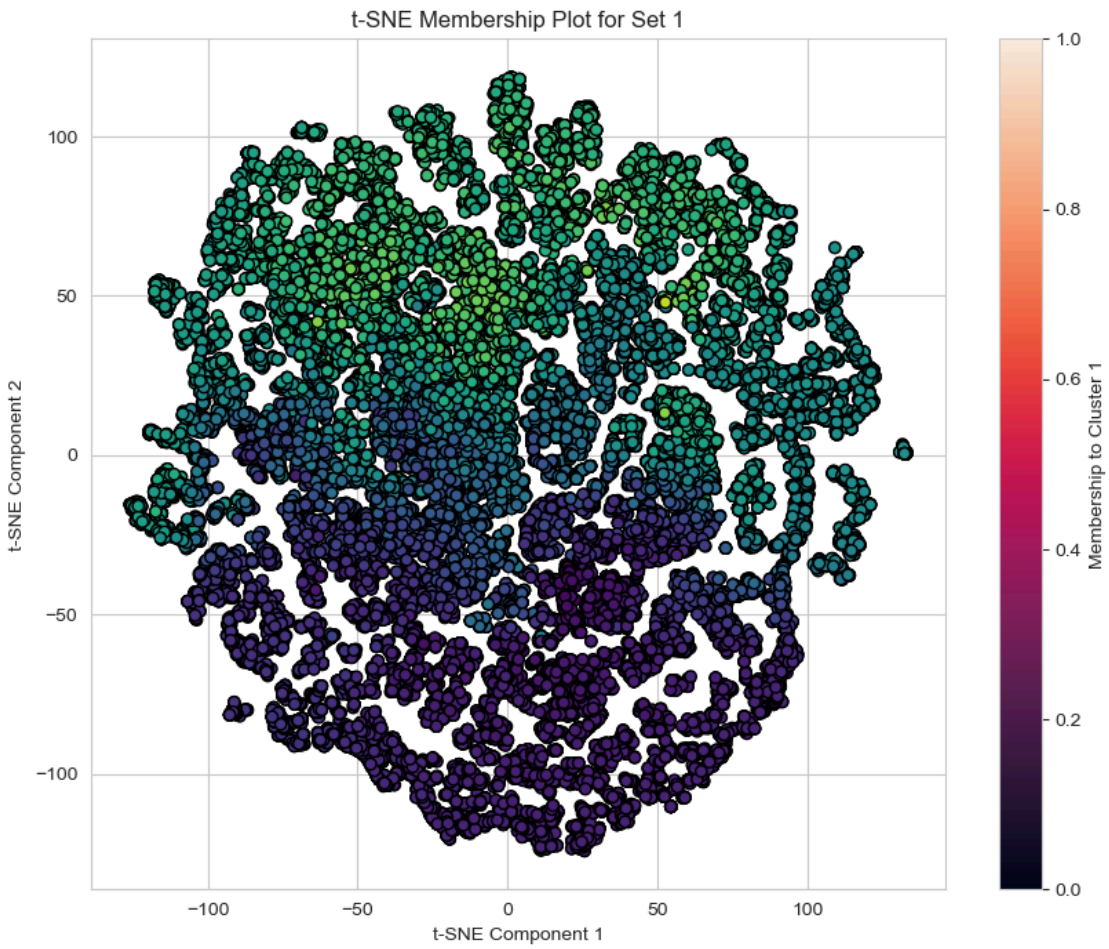


Figure 31. Fuzzy C-Means t-SNE Membership Plot for Feature Set 1

The t-SNE membership plot indicates strong affiliation to Cluster 1 with bright green points, while purple points indicate weaker affiliation. Points with various shades of teal suggest moderate membership levels, likely transitional areas between clusters. The overlap and intermingling of colours highlight the fuzzy nature of the clustering, where data points exhibit characteristics of multiple clusters.

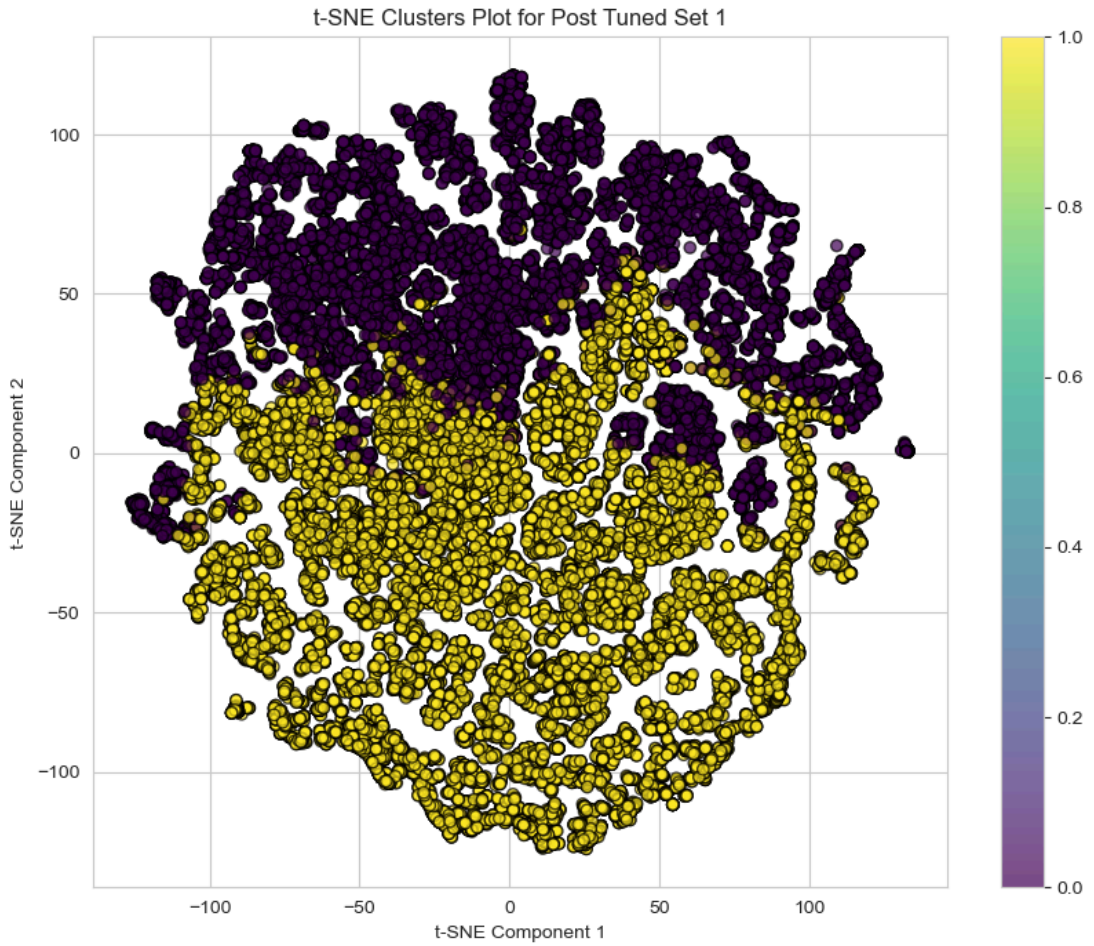


Figure 31. Fuzzy C-Means t-SNE Clusters Plot for Feature Set 1

The t-SNE clusters plot reveals a wide distribution of the clusters, suggesting high variation within each cluster. There is a visible separation between the two clusters, though some overlap exists, indicating transitional data points that share characteristics of both clusters. Both clusters have outliers, with the purple cluster showing sparse outliers far from the main group.

Cluster Centroids Overview

The clusters' centroids provided insights into their characteristics. Table 2 summarises the key attributes of each cluster centroid.

Nutrient	Cluster 0	Cluster 1
Energy (kJ)	1104.65	460.32
Fat (g)	10.05	2.59
Saturated Fat (g)	2.63	0.76
Trans Fat (g)	0.000224	0.00011
Cholesterol (g)	0.000099	0.000052
Carbohydrates (g)	33.95	17.91
Sugars (g)	13.13	9.29
Fibre (g)	1.04	0.59
Proteins (g)	8.65	3.34
Salt (g)	1.01	0.31
Sodium (g)	0.396	0.122
Vitamin A (g)	0.000001	0.000002
Vitamin C (g)	0.000098	0.000713
Calcium (g)	0.001914	0.008867
Iron (g)	0.000046	0.00002

Table 11. Fuzzy C-Means Summary of Cluster Centroids for Feature Set 1

Cluster 0 represents high energy-dense foods with moderate fat and protein content, while Cluster 1 includes lighter foods with lower fat and energy but moderate sugars.

Clustering Evaluation Metrics

The performance of the Fuzzy C-Means clustering was evaluated using the Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index. The results for Feature Set 1 are presented in Table 12.

Metric	Score	Interpretation
Silhouette Score	0.283	Moderate overlap between clusters, indicating potential overlap or shared characteristics among clusters.
Davies-Bouldin Index	1.848	Moderate clustering quality, suggesting clusters are not tightly compact nor distinctly separated.
Calinski-Harabasz Index	10,415.20	A high value indicates well-separated clusters relative to their internal dispersion, influenced by dataset size or variance.

Table 12. Fuzzy C-Means Clustering Evaluation Metrics for Feature Set 1

The Silhouette Score of 0.283 indicates moderate overlap between clusters, suggesting that the boundaries between clusters are not sharply defined. This implies that some data points share characteristics with multiple clusters, which is typical in fuzzy clustering where data points have varying degrees of membership in each cluster. The Davies-Bouldin Index of 1.848 reflects moderate clustering quality, with clusters that are not particularly compact nor distinctly separated. This value indicates that while the clusters are somewhat distinct, there is still a degree of similarity or overlap between them, which aligns with the moderate Silhouette Score.

The Calinski-Harabasz Index of 10,415.20, on the other hand, suggests that the clusters are well-separated relative to their internal dispersion. A higher value of this index generally indicates better-defined clusters. However, considering the moderate values of the other two metrics, the high Calinski-Harabasz Index may be influenced by the dataset's size and variance rather than the clusters' distinctiveness.

4.2.3 Hyperparameter Tuning

This section focuses on the hyperparameter tuning process for Fuzzy C-Means (FCM) clustering to refine the clustering performance. Based on previous analysis, the initial optimal number of clusters (c) was determined to be 2. The hyperparameters tuned included the fuzziness coefficient (m), error tolerance (error), and maximum iterations (maxiter). The ranges for these hyperparameters were as follows:

- Fuzziness Coefficient (m): [1.5, 2.0, 2.5]
- Error Tolerance (error): [0.001, 0.005, 0.01]
- Maximum Iterations (maxiter): [100, 500, 1000]

The tuning process involved iterating over all combinations of these hyperparameters to identify the best set that maximised the Fuzzy Partition Coefficient (FPC). The results of the hyperparameter tuning are summarised in Table 13.

Hyperparameter	Optimal Value
Number of Clusters (n_clusters, c)	2
Fuzziness Coefficient (m)	1.5
Maximum Iterations (max_iter)	100
Error Tolerance (tol)	0.001

Table 13. Best Hyperparameters for Fuzzy C-Means Clustering

The optimal hyperparameters resulted in an FPC of 0.7541, which is a substantial improvement over the initial value of 0.6253. This indicates a more defined separation between clusters, achieved by reducing the fuzziness coefficient and allowing for a more precise convergence with a lower error tolerance and fewer iterations. The changes observed in the clustering evaluation metrics and the cluster centroids further confirm the stability and improvements.

4.2.4 Performance Comparison

To evaluate the effectiveness of the hyperparameter tuning, we compared the performance metrics and cluster centroids before and after tuning. The clustering evaluation metrics used were the Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index. The results are presented in Figure 32 and Table 14.

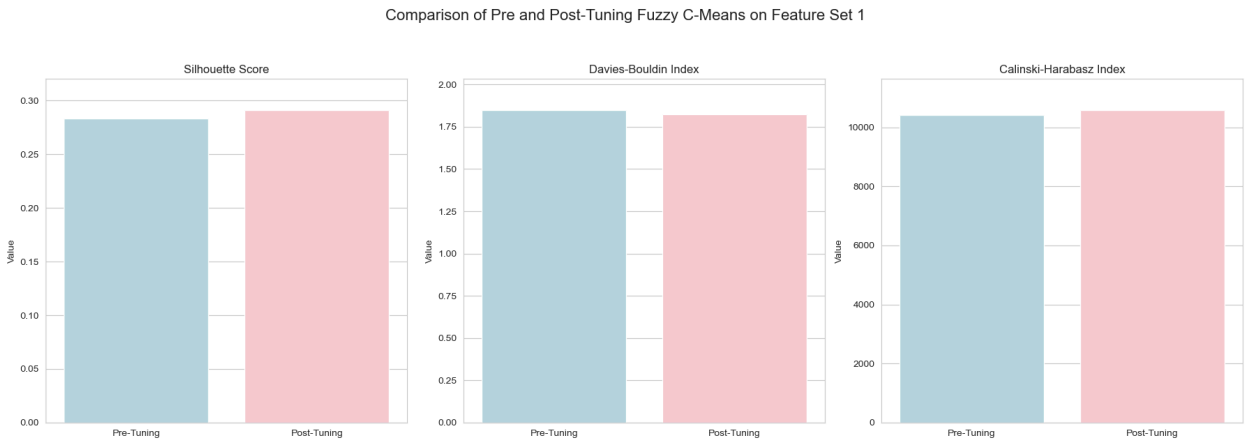


Figure 32. Fuzzy C-Means Clustering Evaluation Metrics Before and After Tuning

Metric	Pre-Tuning	Post-Tuning	Difference
Silhouette Score	0.283	0.291	+0.008
Davies-Bouldin Index	1.848	1.824	-0.024
Calinski-Harabasz Index	10,415	10,579	+164

Table 14. Fuzzy C-Means Clustering Evaluation Metrics Before and After Tuning

The changes in the evaluation metrics indicate that hyperparameter tuning improved clustering quality. The Silhouette Score increased slightly, suggesting a modest cluster separation and coherence improvement. The Davies-Bouldin Index decreased, indicating better compactness and separation of clusters. The Calinski-Harabasz Index increased, reflecting the clusters' enhanced distinctiveness and internal density.

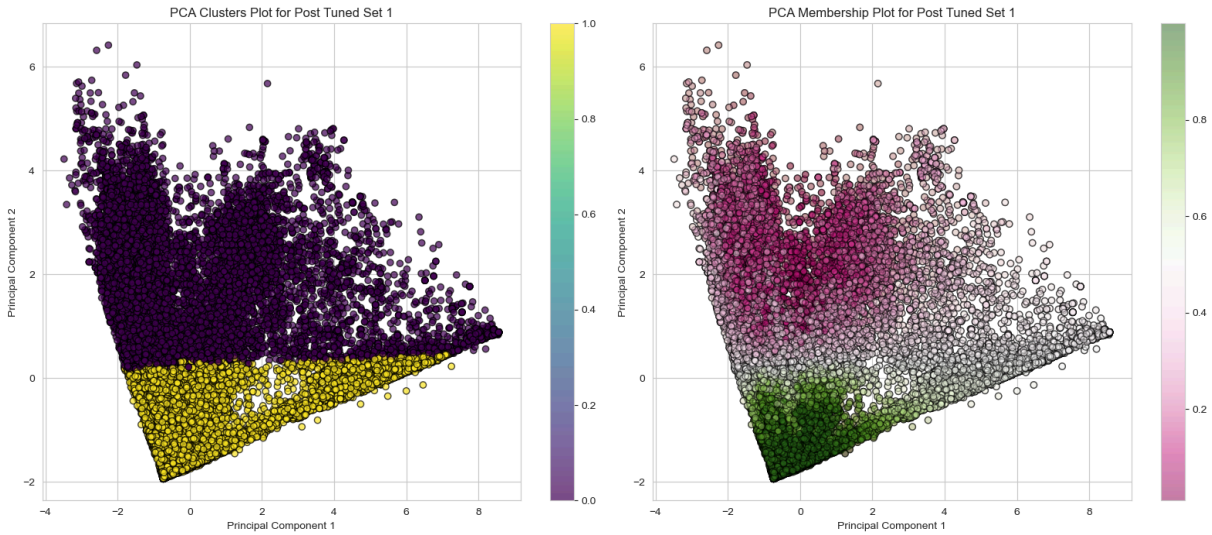


Figure 33. Fuzzy C-Means PCA Visualisation of Clusters After Hyperparameter Tuning

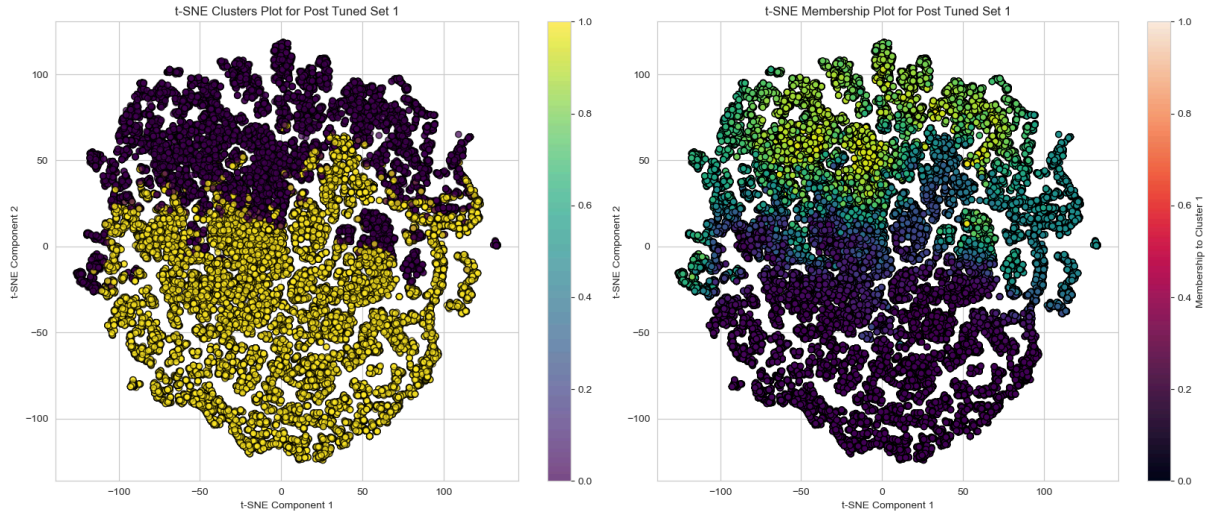


Figure 34. Fuzzy C-Means t-SNE Visualisation of Clusters After Hyperparameter Tuning

From the visualisations, the post-tuning clustering results exhibit subtle improvements in cluster shapes and the compactness and spread of data points. Initially, it was challenging to discern distinct improvements in cluster shapes or the compactness and spread of data points based solely on cluster visualisations. However, some differences were observed in the membership degree plot.

In the PCA visualisation, colours representing membership levels (pink and green for higher and lower memberships) have deepened, suggesting slightly more compact clusters post-tuning. This intensity of colours indicates that the clusters became more defined and compact after the tuning process. Similarly, the t-SNE visualisation reveals a noticeable colour shift, with some data points transitioning from green to a yellowish-green hue. This colour shift could potentially indicate improved separation between clusters, albeit subtly. While the visual changes may not be overtly dramatic, they imply evolving cluster memberships and subtle refinements in cluster structures.

Comparison of Cluster Centroids Before and After Tuning

To further analyse the impact of hyperparameter tuning, we compared the clusters' centroids before and after tuning. The comparison is presented in Tables 15 and 16.

Nutrient	Pre-Tuning	Post-Tuning	Difference
Energy (kJ)	1104.65	1171.07	66.42
Fat (g)	10.05	11.45	1.4
Saturated Fat (g)	2.63	2.99	0.36
Trans Fat (g)	0.000224	0.000221	-0.000003
Cholesterol (g)	0.000099	0.000107	0.000008
Carbohydrates (g)	33.95	34.19	0.24
Sugars (g)	13.13	13.46	0.33
Fibre (g)	1.04	1.02	-0.02
Proteins (g)	8.65	9.22	0.57
Salt (g)	1.01	1.11	0.1
Sodium (g)	0.396	0.435	0.039
Vitamin A (g)	0.000001	0.000001	0
Vitamin C (g)	0.000098	0.000076	-0.000022
Calcium (g)	0.001914	0.001621	-0.000293
Iron (g)	0.000046	0.000047	0.000001

Table 15. Cluster 0 Centroid Comparison for Fuzzy C-Means

Nutrient	Pre-Tuning	Post-Tuning	Difference
Energy (kJ)	460.32	464.86	4.54
Fat (g)	2.59	2.28	-0.31

Saturated Fat (g)	0.76	0.68	-0.08
Trans Fat (g)	0.000110	0.000120	0.00001
Cholesterol (g)	0.000052	0.000049	-0.000003
Carbohydrates (g)	17.91	18.87	0.96
Sugars (g)	9.29	9.21	-0.08
Fibre (g)	0.59	0.65	0.06
Proteins (g)	3.34	3.35	0.01
Salt (g)	0.310	0.299	-0.011
Sodium (g)	0.122	0.118	-0.004
Vitamin A (g)	0.000002	0.000002	0
Vitamin C (g)	0.000713	0.000659	-0.000054
Calcium (g)	0.008867	0.008241	-0.000626
Iron (g)	0.000020	0.000020	0

Table 16. Cluster 1 Centroid Comparison for Fuzzy C-Means

Post-tuning, Cluster 1 displays minor changes in its nutrient profile. The energy content increases slightly (+4.54 kJ), while the fat content decreases marginally (-0.31 g). The changes in other nutrients, such as carbohydrates (+0.96 g), fibre (+0.06 g), and proteins (+0.01 g), are minimal. The slight decrease in vitamin C (-0.000054 g) and calcium (-0.000626 g) does not significantly impact the cluster's overall nutritional characteristics. Cluster 1 remains representative of lighter, sugar-conscious foods suitable for health-conscious diets.

The changes in centroids before and after tuning are generally minimal for both clusters. The differences in nutrient values are slight, indicating that the adjustments to hyperparameters did not lead to substantial shifts in the clusters' centroids. This further supports the conclusion that the model had already converged to a stable solution with the initial settings. The post-tuning centroids reflect subtle refinements in cluster memberships and structures, confirming the effectiveness of the tuning process in providing marginal enhancements to the clustering results.

4.3 DBSCAN Clustering

4.3.1 Feature Set Selection

In contrast to previous clustering approaches like K-Means and Fuzzy C-Means, the DBSCAN algorithm does not require a predefined number of clusters. Instead, it assigns data points to clusters based on density, allowing for automatically identifying a suitable number of clusters. Initially, we performed modelling using the default DBSCAN settings with epsilon (eps) set to 0.5 and a minimum number of samples (min_samples) set to 5. The results of these initial settings are summarised in Table 17.

Features Set	Silhouette Score	Number of Clusters
1	-0.4566	168
2	-0.4396	165
3	-0.3330	398
4	-0.4136	297
5	-0.1276	265
6	-0.4733	188

Table 17. Initial DBSCAN Clustering Results

The initial clustering results indicated poor clustering quality, as evidenced by the high number of clusters and negative silhouette scores across all feature sets. Negative silhouette scores suggest a significant overlap between clusters, with data points being closer to neighbouring clusters than to their own. Additionally, the large number of clusters suggests that the default epsilon parameter may be too small, resulting in the algorithm forming too many clusters.

To improve the clustering results, we adjusted the eps and min_samples parameters. We increased the epsilon value from 0.5 to 0.8 and the minimum points from 5 to 10. The adjusted results are summarised in Table 18.

Features Set	Silhouette Score	Number of Clusters
1	0.1993	10
2	0.1978	10
3	-0.2980	140
4	-0.3808	69
5	-0.0582	56
6	-0.1341	11

Table 18. Adjusted DBSCAN Clustering Results

After adjusting the parameters, Sets 1, 2, and 6 showed positive silhouette scores, indicating better clustering quality with fewer clusters. Set 1 exhibited the highest silhouette score of 0.1993 while forming 10 clusters, making it the best feature set for further analysis.

4.3.2 Cluster Identification and Visualisation

After adjusting the parameters (eps and min_samples) for the DBSCAN algorithm, the results indicated improved clustering quality for certain feature sets. In this section, we will visualise the clustering results for the best-performing feature set (Set 1) using PCA and t-SNE plots, both with and without outliers.

PCA Visualisation

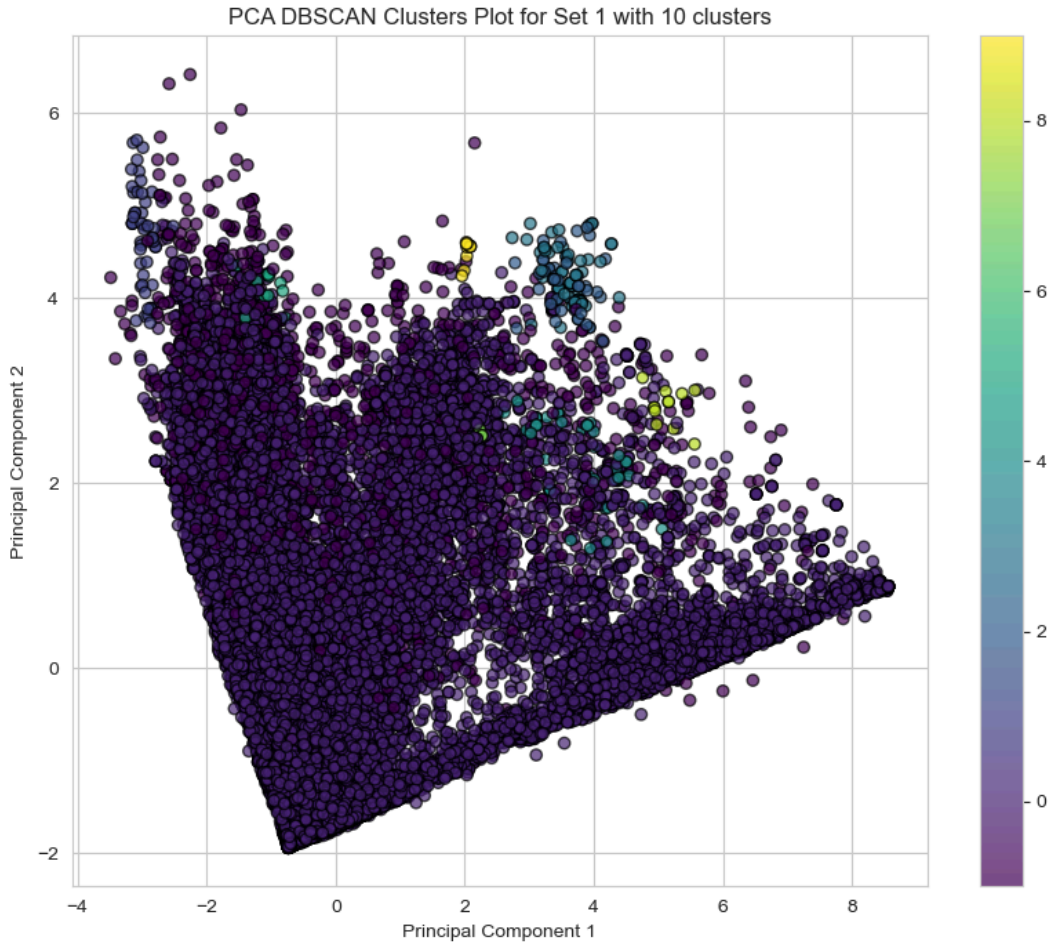


Figure 35. PCA Visualization of DBSCAN Clusters with Outliers for Set 1

The PCA visualisation of DBSCAN clusters for Set 1 with outliers shows a tight central mass of data points, indicating a high-density area where most data points are similar with respect to the first two principal components. Several smaller clusters are visible around the periphery of this

central mass, and these clusters are relatively well-separated from each other and from the main group, although some overlap is present.

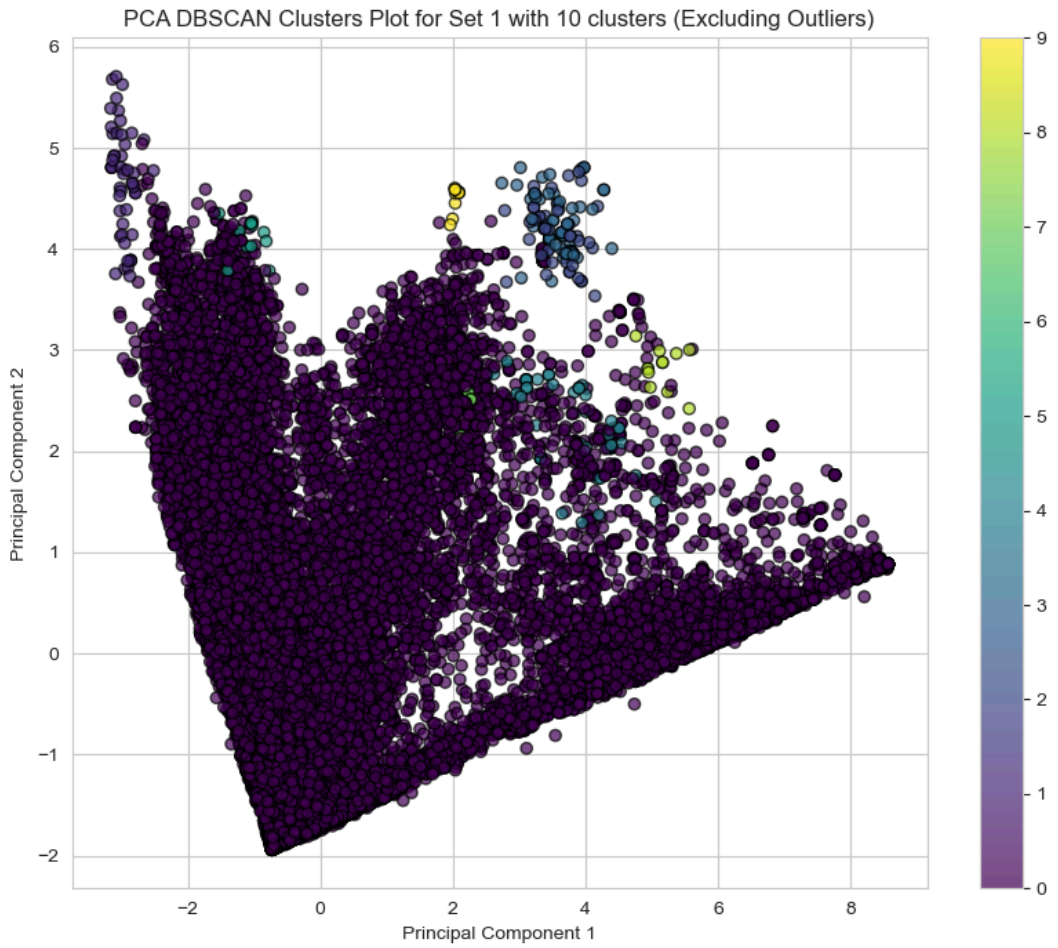


Figure 36. PCA Visualization of DBSCAN Clusters without Outliers for Set 1

The PCA visualisation without outliers enhances the visibility of distinct clusters. The removal of outliers reduces noise and highlights the core structure of the clusters. The clusters appear more compact and well-separated, with less overlap, providing a clearer representation of the clustering results.

t-SNE Visualisation

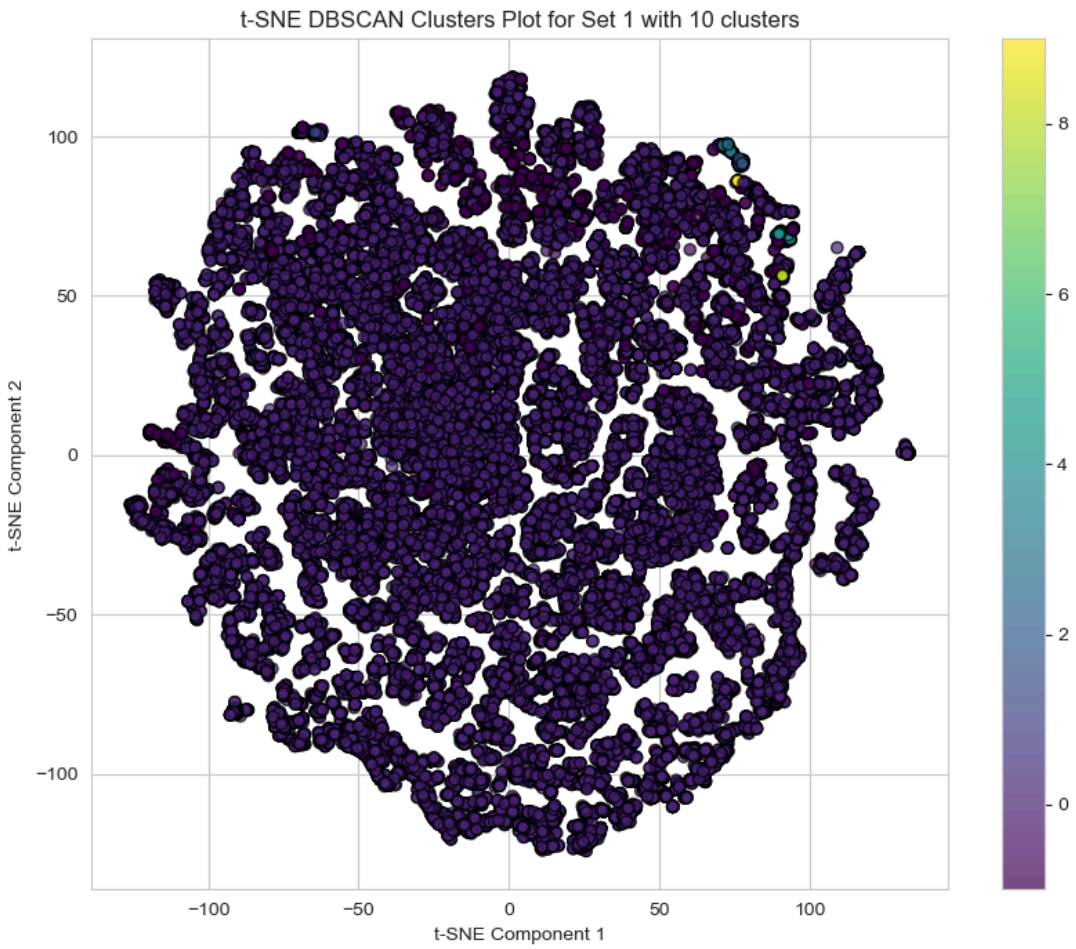


Figure 37. t-SNE Visualization of DBSCAN Clusters with Outliers for Set 1

The t-SNE visualisation with outliers supports the PCA findings, showing a dense central cluster with several smaller clusters around it. The t-SNE plot captures the local structure of the data, highlighting well-separated clusters around the high-density central mass.

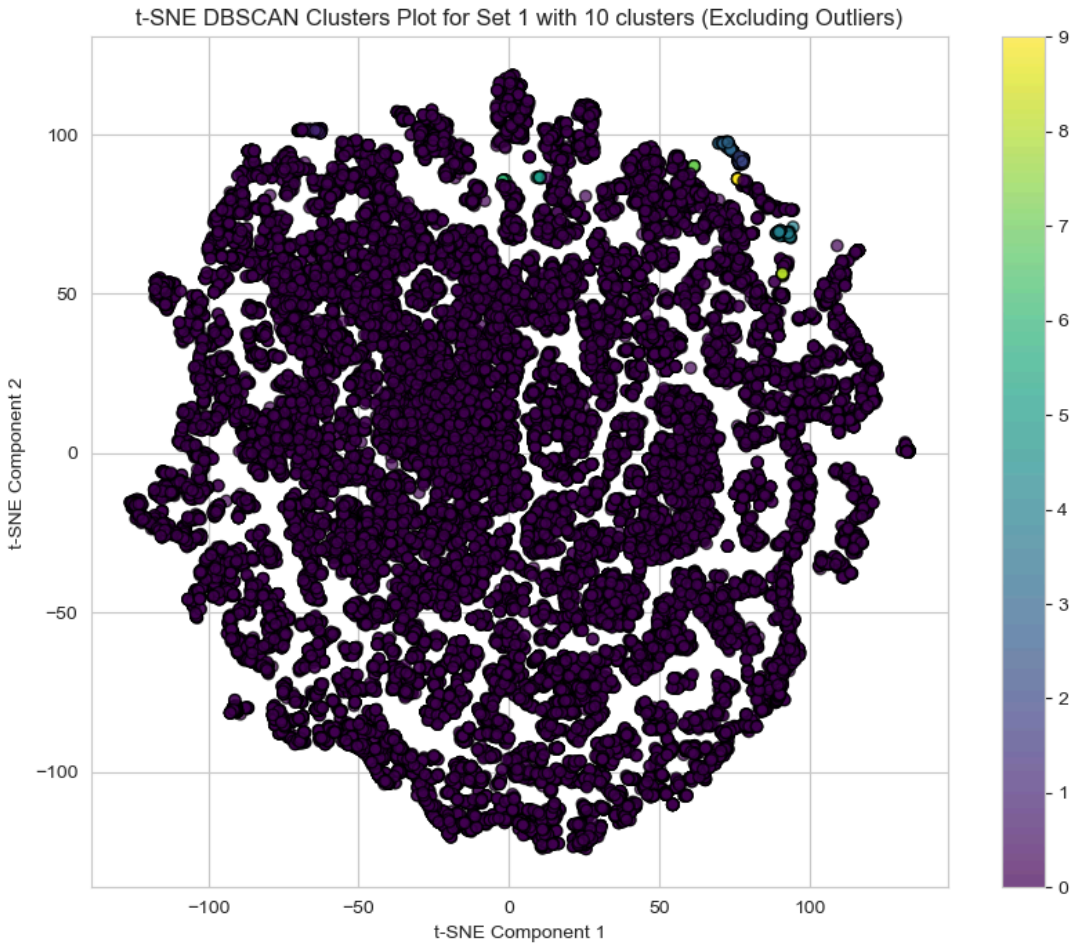


Figure 38. t-SNE Visualization of DBSCAN Clusters without Outliers for Set 1

The t-SNE visualisation without outliers provides a clearer view of the cluster boundaries. The removal of outliers enhances the separation between clusters, resulting in distinct and well-defined clusters. This visualisation highlights the effectiveness of DBSCAN in identifying meaningful clusters when outliers are excluded.

Cluster Centroid Overview

In DBSCAN, the concept of centroids is not inherent, as it classifies points based on density rather than defining a central point for each cluster. To approximate centroids, we calculate the mean of the points within each cluster identified by DBSCAN for Set 1. The summary of the cluster centroids is presented in Table 19.

Nutrient	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8	Cluster 9	Cluster 10
Energy (kJ)	709.01	1080.19	2288.69	2298.00	1909.90	1922.50	1369.90	1926.14	1790.92	2427.40
Fat (g)	5.13	16.13	33.49	33.52	18.97	18.12	2.90	24.86	12.35	37.56
Saturated Fat (g)	1.40	6.06	7.00	6.58	1.79	7.99	0.44	2.34	7.33	7.48
Trans Fat (g)	0.00016	0.00000	0.00213	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
Cholesterol (g)	0.00007	0.00000	0.00000	0.00002	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
Carbohydrates (g)	24.85	1.09	55.14	55.37	61.86	65.11	64.13	48.79	74.96	57.19
Sugars (g)	10.98	0.68	52.21	53.36	52.70	7.15	2.31	37.24	66.78	36.19
Fibre (g)	0.78	0.14	0.01	3.13	3.11	2.78	0.00	3.73	2.71	0.93
Proteins (g)	5.39	27.53	5.40	5.24	8.10	8.50	9.17	8.73	2.41	3.12
Salt (g)	0.57	4.09	0.13	0.13	0.09	2.44	3.92	0.06	0.06	0.54
Sodium (g)	0.22	1.61	0.05	0.05	0.04	0.96	1.54	0.02	0.02	0.21
Vitamin A (g)	0.000002	0.000000	0.000000	0.000000	0.000000	0.000011	0.000000	0.000000	0.000000	0.000000
Vitamin C (g)	0.000431	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
Calcium (g)	0.005571	0.000000	0.001489	0.000000	0.000000	0.000000	0.000000	0.000000	0.004108	0.000000
Iron (g)	0.000028	0.000000	0.000036	0.000000	0.000000	0.000257	0.000000	0.000000	0.000386	0.000000

Table 19. Summary of DBSCAN Cluster Centroids for Feature Set 1

The results of the DBSCAN clustering reveal a wide range of nutritional profiles across different clusters, providing insight into the characteristics within each group. Cluster 1 encompasses foods with a balanced dietary composition, featuring moderate energy levels at 709.01 kJ and a harmonious blend of macronutrients, including moderate fat content at 5.13 g and significant

carbohydrates at 24.85 g. These foods also have a noticeable sugar presence at 10.98 g, indicating a preference for sweeter flavours.

In contrast, Cluster 2 is notable for its emphasis on protein-rich offerings, with substantial protein content at 27.53 g, elevated fat levels at 16.13 g, and minimal carbohydrate intake at 1.09 g. Conversely, Cluster 3 showcases high-energy foods with high levels of fats and sugars, representing the profile of high-calorie, energy-dense snacks. Cluster 4 closely resembles Cluster 3 in energy and fat content but differs with a slightly higher fibre intake, potentially balancing its nutritional profile. The subsequent clusters demonstrate similarly distinct nutritional compositions, providing valuable insights into dietary preferences and consumption patterns across diverse food categories.

Clustering Evaluation Metrics

The performance of the DBSCAN clustering was evaluated using the Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index. The results for Feature Set 1 are presented in Table 20.

Metric	Score	Interpretation
Silhouette Score	0.1993	Indicates reasonably distinct and cohesive clusters but with some degree of overlap.
Davies-Bouldin Index	1.2727	Reflects moderate clustering quality, suggesting that clusters are not particularly compact nor highly distinct.
Calinski-Harabasz Index	238.7583	Suggests well-separated clusters relative to their internal dispersion, with the value influenced by dataset size or variance.

Table 20. DBSCAN Clustering Evaluation Metrics for Feature Set 1

The Silhouette Score of 0.1993 indicates reasonably distinct and cohesive clusters, although there is some degree of overlap between clusters. This suggests that while the clusters are well-formed, certain data points may share characteristics with multiple clusters, leading to less sharply defined boundaries.

The Davies-Bouldin Index of 1.2727 reflects moderate clustering quality, with clusters that are not particularly compact nor highly distinct. This value indicates that while the clusters are somewhat distinct, there is still a degree of similarity or overlap between them, aligning with the moderate Silhouette Score.

The Calinski-Harabasz Index of 238.7583 suggests that the clusters are well-separated relative to their internal dispersion. A higher value of this index generally indicates better-defined clusters. However, considering the moderate values of the other two metrics, the high Calinski-Harabasz Index may be influenced by the dataset's size and variance rather than the distinctiveness of the clusters.

4.3.3 Hyperparameter Tuning

This section focuses on the hyperparameter tuning process for DBSCAN clustering to refine the clustering performance. Based on the previous analysis, the initial parameters used were epsilon (eps) set to 0.8 and minimum samples (min_samples) set to 10. The hyperparameters tuned included the epsilon (eps) and minimum samples (min_samples). The ranges for these hyperparameters were as follows:

- Epsilon (eps): [0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95]
- Minimum Samples (min_samples): [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]

The tuning process involved iterating over all combinations of these hyperparameters to identify the best set that maximised the Silhouette Score. The results of the hyperparameter tuning are summarised in Table 21.

Hyperparameter	Optimal Value
Epsilon (eps)	0.9
Minimum Samples	14

Table 21. Best Hyperparameters for DBSCAN Clustering

The optimal hyperparameters resulted in a Silhouette Score of 0.2769, which is a substantial improvement over the initial value of 0.1993. This indicates more distinct and cohesive clusters, achieved by adjusting the epsilon value to a higher range and increasing the minimum samples to ensure denser clusters. The changes observed in the clustering evaluation metrics further confirm the stability and improvements.

4.3.4 Performance Comparison

To evaluate the effectiveness of the hyperparameter tuning, we compared the performance metrics before and after tuning. The clustering evaluation metrics used were the Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index. The results are presented in Figure 39 and Table 22.

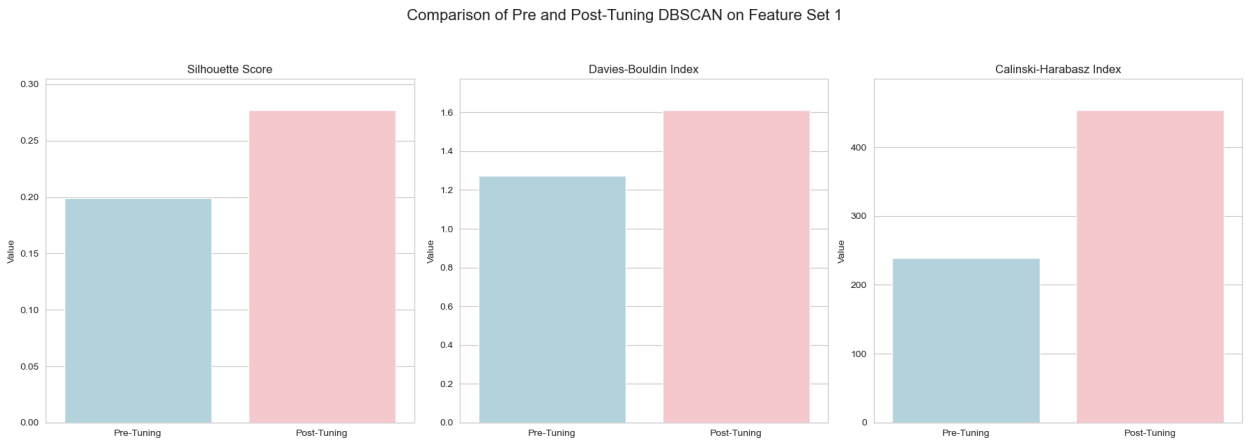


Figure 32. Fuzzy C-Means Clustering Evaluation Metrics Before and After Tuning

Metric	Pre-Tuning	Post-Tuning	Difference
Silhouette Score	0.1993	0.2769	+0.0776
Davies-Bouldin Index	1.2727	1.6112	+0.3385
Calinski-Harabasz Index	238.7583	453.5385	+214.7802

Table 14. Fuzzy C-Means Clustering Evaluation Metrics Before and After Tuning

Following the tuning process, the number of clusters reduced substantially from 10 to 4, indicative of a consolidation of data points into more extensive and potentially more significant groups. This adjustment has effectively eliminated less significant clusters while refining the boundaries between the remaining ones. Concurrently, the Silhouette Score experienced an uptick from 0.1993 to 0.2769, signalling enhanced separation and coherence within the

post-tuning clusters. While the score still indicates some overlap, this improvement underscores a more transparent and more distinct cluster structure.

However, the Davies-Bouldin Index increased from 1.2727 to 1.6112, suggesting a potential decline in clustering quality. This index gauges the average similarity between clusters and typically favours lower values. The uptick in the index may reflect a trade-off between cluster separation and compactness, particularly in light of the reduced number of clusters. Conversely, the Calinski-Harabasz Index witnessed a notable surge from 238.758 to 453.538, aligning with the reduced cluster count and the improved Silhouette Score. This increase implies denser and more separated clusters, indicating improved clustering performance despite the elevated Davies-Bouldin Index.

In summary, the post-tuning results suggest that the DBSCAN algorithm, following adjustments, has established a more coherent internal structure within the dataset, yielding fewer yet more discernible and potentially meaningful clusters. Despite the rise in the Davies-Bouldin Index, the increase in other indices points towards overall improved clustering performance.

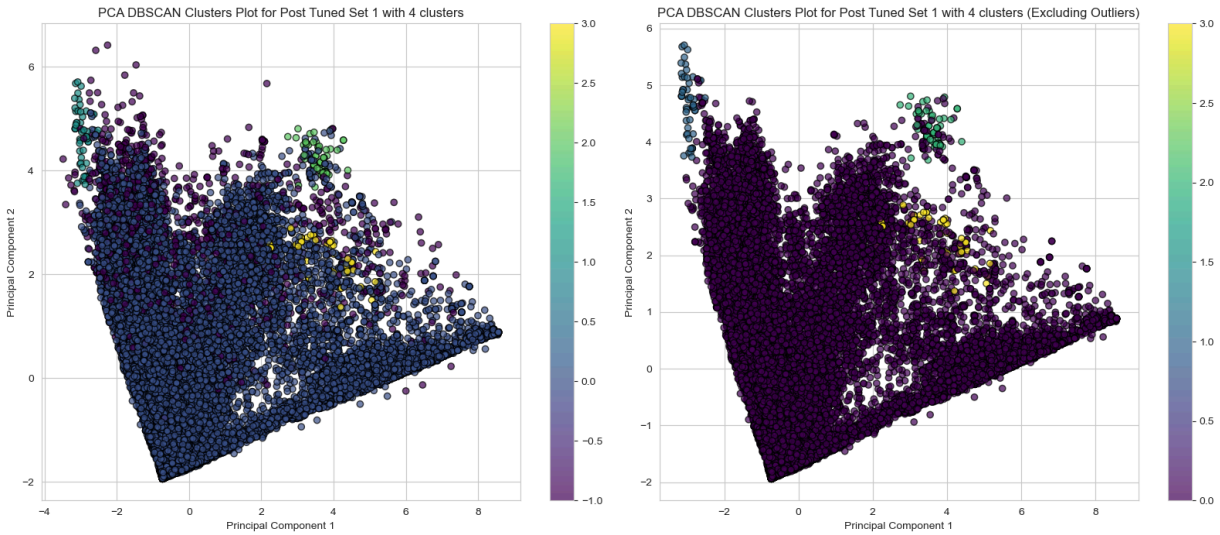


Figure 33. PCA Visualisation of DBSCAN Clusters with and without Outliers for Set 1

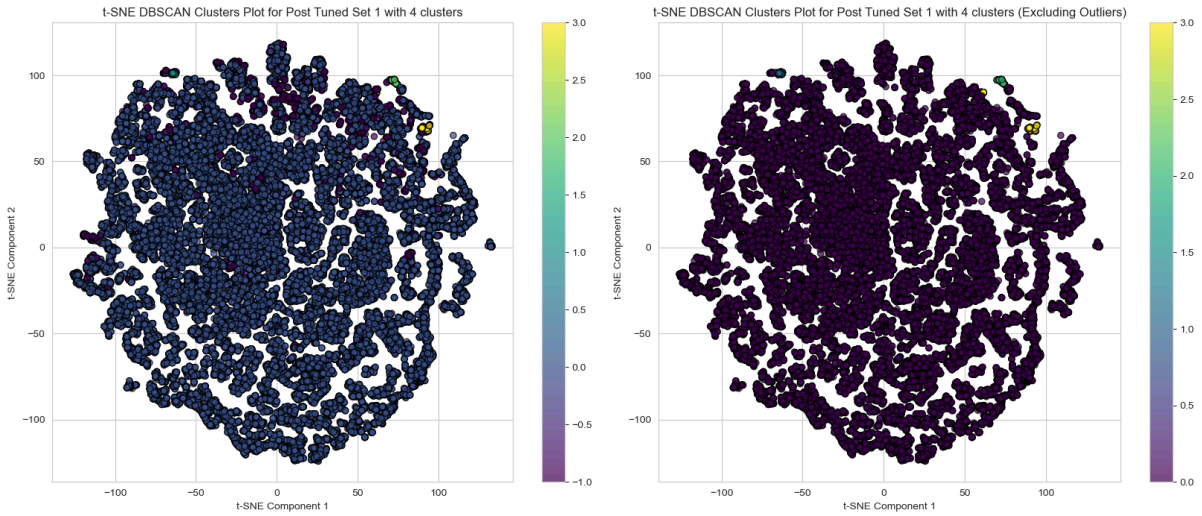


Figure 34. t-SNE Visualisation of DBSCAN Clusters with and without Outliers for Set 1

The visualisation results from PCA and t-SNE plots reveal several critical observations regarding the clustered data post-tuning. Across both visualisation techniques, four main clusters are discernible, each distinguished by different colours and showing relative distinctiveness, albeit with some observable overlap. A predominant cluster emerges, housing most of the data points, similar to the pre-tuning phase. In the t-SNE plot, clusters 1 and 2 appear notably compact, while cluster 3 exhibits some dispersion.

Moreover, outliers are evident in the first charts of both PCA and t-SNE plots, contrasting with the second charts that focus solely on clustered data by excluding outliers. This prevalence of outliers in the initial plots suggests that a significant portion of the dataset may not align well with the dense regions defined by the algorithm's parameters, possibly indicating inherent variance or distinct subgroups within the data.

The provided figures, showcasing PCA and t-SNE visualisations both with and without outliers, underscore the efficacy of parameter tuning in achieving a clearer and more discernible cluster structure. Combined with the reduced cluster count and the enhanced Silhouette Score, this refinement signifies a more coherent and meaningful clustering outcome.

Comparison of Cluster Centroids Before and After Tuning

The clusters' centroids after tuning provide insights into each group's refined nutritional characteristics. Table 15 summarises the key attributes of each cluster centroid after tuning the DBSCAN algorithm for Feature Set 1.

Nutrient	Cluster 0	Cluster 1	Cluster 2	Cluster 3
Energy (kJ)	717.2	1089.13	2293.74	1908.45
Fat (g)	5.25	16.28	33.38	19.41
Saturated Fat (g)	1.43	6.11	6.52	1.91
Trans Fat (g)	0.00017	0	0	0
Cholesterol (g)	0.00007	0	0	0
Carbohydrates (g)	25.05	1.25	55.29	60.64
Sugars (g)	11.08	0.7	53.25	51.81
Fibre (g)	0.78	0.13	3.19	3.21
Proteins (g)	5.41	27.57	5.36	8.25
Salt (g)	0.57	4.05	0.13	0.09
Sodium (g)	0.23	1.6	0.05	0.03
Vitamin A (g)	0.000002	0	0	0
Vitamin C (g)	0.000427	0	0	0
Calcium (g)	0.00555	0	0	0
Iron (g)	0.00003	0	0	0

Table 15. DBSCAN Post Tuning Cluster Centroids for Set 1

Since the number of clusters before and after tuning differs, we will compare the cluster centroids in a generalised manner:

Cluster	Before Tuning	Cluster	After Tuning
0	Energy: Low Fat: Low Sugars: Moderate	0	Energy: Low Fat: Low Sugars: Moderate
1	Energy: Moderate Fat: Moderate Protein: High	1	Energy: Slightly increased Fat: Slightly increased Protein: High
2 and 3	Energy: Very High Fat: Very High Carbohydrates: Very High	2	Similar to pre-tuning Clusters 2 and 3 but with refined separation
4 to 9	Varying from moderate to very high in calorie content with different levels of sugars and fats	3	Reflects similar attributes to pre-tuning Cluster 4 but with better differentiation in nutritional content, particularly protein and carbohydrates

Table 16. Comparison of DBSCAN Clusters Before and After Tuning

The analysis of the clustered data shows significant changes after tuning, especially in how nutritional profiles are categorised within each cluster. Cluster 0 remains consistent, representing low-energy foods with moderate fat content and moderate sugar levels, indicating a clear grouping before and after tuning. In contrast, Cluster 1 has been refined, with foods showing moderate energy and fat and high protein content, providing a more precise capture of their nutritional attributes after tuning. Cluster 2, identified as very high-energy foods abundant in fat and carbohydrates, maintains its characteristics across pre- and post-tuning stages. However, the tuning process enhances the separation within this cluster, resulting in clearer boundaries between its constituent foods.

Similarly, Cluster 3 reflects attributes similar to pre-tuning Clusters 4 to 9 but with improved differentiation in nutritional content, particularly in protein and carbohydrates. The tuning consolidates these diverse clusters into one group, facilitating a more cohesive and understandable grouping of foods with high-calorie content and varying sugar and fat levels. Overall, the tuning process improves the coherence and separation of the clusters, leading to more distinct and meaningful groupings. The refined post-tuning clusters exhibit clearer nutritional profiles, aligning with the observed improvements in clustering evaluation metrics.

5.0 Discussion

This section discusses the key findings from the clustering analysis using K-Means, Fuzzy C-Means (FCM), and DBSCAN algorithms. It also highlights the implications of these findings, the strengths and limitations of each method, and recommendations for future research.

Metric	K-Means	Fuzzy C-Means	DBSCAN
Silhouette Score	0.3365	0.291	0.2769
Davies-Bouldin Index	1.371	1.824	1.6112
Calinski-Harabasz Index	20239.074	10579	453.5385
Number of Clusters	4	2	4

Table 17. Clustering Evaluation Metrics for K-Means, FCM, and DBSCAN After Tuning

The K-Means clustering analysis identified four distinct clusters with varying nutritional profiles. The hyperparameter tuning process involved adjusting the number of initialisations, maximum iterations, and tolerance. After tuning, the Silhouette Score increased slightly from 0.336 to 0.3365, the Davies-Bouldin Index saw a slight increase from 1.370 to 1.371, and the Calinski-Harabasz Index improved marginally from 20239.057 to 20239.074. These changes indicate that the initial settings were already close to optimal, with the post-tuning clusters exhibiting slightly better separation and coherence. The stability of the clustering results suggests that K-Means is a robust method for identifying distinct groups within the dataset.

The FCM analysis also identified clusters with distinct nutritional profiles. The tuning process significantly improved clustering performance by focusing on the fuzziness coefficient, error tolerance, and maximum iterations. The optimised parameters resulted in a Silhouette Score increase from 0.283 to 0.291, a Davies-Bouldin Index decrease from 1.848 to 1.824, and a Calinski-Harabasz Index increase from 10,415 to 10,579. These improvements indicate better-defined clusters with reduced overlap and enhanced separation. The FCM algorithm's ability to assign membership degrees to each data point provides a nuanced understanding of the clustering structure, making it a valuable tool for datasets with overlapping cluster characteristics.

The DBSCAN algorithm, which does not require the pre-specification of the number of clusters, initially produced many clusters with poor separation. Significant improvements were observed after hyperparameter tuning, which involved adjustments to the epsilon and minimum points parameters. The optimised parameters resulted in a Silhouette Score increase from 0.1993 to

0.2769, a Davies-Bouldin Index increase from 1.2727 to 1.6112, and a Calinski-Harabasz Index increase from 238.7583 to 453.5385. The number of clusters was reduced from ten to four, indicating a more coherent and meaningful clustering structure. The post-tuning clusters exhibited better separation and coherence, as the evaluation metrics and visualisations confirmed.

Implications of the Findings

The clustering analysis provides valuable insights into the nutritional profiles of the foods in the dataset. The distinct clusters identified by each algorithm can inform targeted dietary recommendations and nutritional planning. For instance, high-energy, high-fat clusters identified by K-Means and DBSCAN can be targeted for interventions to reduce calorie intake. Meanwhile, the nuanced clustering provided by FCM can be used to identify foods that fit into multiple dietary categories, offering flexibility in dietary planning. The improvements achieved through hyperparameter tuning highlight the importance of optimising clustering algorithms to achieve the best possible results. The differences in clustering performance across the algorithms underscore the need to choose the appropriate method based on the dataset's characteristics and the research objectives.

Strengths and Limitations

K-Means demonstrates simplicity and efficiency in identifying well-separated clusters. Still, its performance highly depends on the initial choice of cluster centroids and the assumption of spherical cluster shapes. The minimal improvements observed through hyperparameter tuning suggest that K-Means may need to capture complex cluster structures better. In contrast, FCM's strength lies in its ability to assign membership degrees to data points, providing a nuanced understanding of cluster structures. The substantial improvements observed through hyperparameter tuning indicate the algorithm's flexibility and robustness. However, the algorithm's sensitivity to the fuzziness coefficient and the potential for overlapping clusters can complicate the interpretation of results.

DBSCAN's primary strength is its ability to identify clusters of varying shapes and sizes and to classify noise points. The significant improvements achieved through hyperparameter tuning demonstrate its flexibility. However, DBSCAN's performance is susceptible to the choice of epsilon and minimum points, and it may struggle with datasets containing clusters of varying densities. Despite these challenges, DBSCAN's capability to identify clusters without pre-defining their number is advantageous in exploratory data analysis.

6.0 Conclusion

The clustering analysis using K-Means, Fuzzy C-Means (FCM), and DBSCAN algorithms provided valuable insights into the nutritional profiles of the foods in the dataset. The hyperparameter tuning process highlighted the importance of optimising algorithm parameters to achieve the best clustering performance. Each algorithm exhibited distinct strengths and limitations, underscoring the need to choose the appropriate method based on the dataset's characteristics and the research objectives.

The K-Means algorithm demonstrated robustness in identifying well-separated clusters, with minimal improvements observed through hyperparameter tuning. After tuning, the FCM algorithm showed significant flexibility and robustness, substantially enhancing cluster coherence and separation. DBSCAN's ability to identify clusters of varying shapes and sizes and to classify noise points was particularly advantageous. However, its performance was susceptible to the choice of epsilon and minimum points.

The findings from this analysis can inform targeted dietary recommendations and nutritional planning, with potential applications in public health and personalised nutrition. The distinct clusters identified by each algorithm can help tailor interventions to reduce calorie intake or offer flexibility in dietary planning.

6.1 Recommendations for Future Research

Future research should focus on exploring the application of different clustering algorithms, such as hierarchical clustering and Gaussian mixture models, to validate the findings and discover alternative clustering structures. It would also be beneficial to include domain knowledge in the clustering process, such as using expert-defined nutritional categories, to improve the interpretability and relevance of the results.

Combining clustering analysis with other data analysis techniques, such as classification and regression, offers a more comprehensive understanding of the dataset. For example, identifying the factors that influence the clustering structure can help in creating targeted interventions and nutritional planning. Further feature engineering to extract more relevant and informative features could enhance the discriminative power of the clustering models. Additionally, incorporating advanced dimensionality reduction techniques beyond PCA and t-SNE, such as UMAP or autoencoders, may reveal additional insights into the dataset's structure.

Furthermore, leveraging semi-supervised or active learning approaches could help refine cluster assignments by integrating domain knowledge or user feedback iteratively. Lastly, future research should consider the scalability of clustering algorithms to larger datasets and the potential for parallel processing to improve computational efficiency. This would enable the application of clustering techniques to big data scenarios, further enhancing their utility in various research and practical applications.

7.0 References

Data Source

Open Food Facts, Kaggle Team, Bournhonesque, R., & Slamich, P. (n.d.). *Open Food Facts database. Kaggle*. Retrieved March 25, 2024, from https://www.kaggle.com/datasets/openfoodfacts/world-food-facts/data

Paper Reference

Adriyendi. (2016). Clustering using K-Means and Fuzzy C-Means on Food Productivity. International Journal of U- and E-service, Science and Technology, 9(12), 291–308. https://doi.org/10.14257/ijunesst.2016.9.12.26

Akbay, A., Elhan, A. H., Özcan, Ç., & Demırtaş, S. (2000). Hierarchical cluster analysis as an approach for systematic grouping of diet constituents on basis of fatty acid, energy and cholesterol content: application on consumable lamb products. Medical Hypotheses, 55(2), 147–154. https://doi.org/10.1054/mehy.1999.1038

Alosaimi, N., Sherar, L. B., Griffiths, P., et al. (2023). Clustering of diet, physical activity and sedentary behaviour and related physical and mental health outcomes: A systematic review. BMC Public Health, 23, 1572. https://doi.org/10.1186/s12889-023-16372-6

Alqorni, F., Mahmudy, W. F., & Widodo, A. (2021). Application of Density based Spatial Clustering Application with Noise (DBSCAN) in determining the quality of Keprok orange and Siam orange hybrid in the Research Center of Orange and Subtropic Plants Batu City. JITeCS (Journal of Information Technology and Computer Science), 6(1), 1–8. https://doi.org/10.25126/jitecs.202161244

Amma Palanisamy, T. S. C., Jayaraman, M., Vellingiri, K., & Guo, Y. (2018). Optimization-based neutrosophic set for medical image processing. Neutrosophic Set in Medical Image Analysis, 189-206. https://doi.org/10.1016/B978-0-12-818148-5.00009-6

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., & Perona, I. (2012). An extensive comparative study of cluster validity indices. Pattern Recognition, 46(1), 243-256. https://doi.org/10.1016/j.patcog.2012.07.021

Ares, G. (2013). Cluster analysis: Application in food science and technology. ResearchGate, 103–120. https://doi.org/10.1002/9781118434635.ch7

Babichev, S., Durnyak, B., Zhydetskyy, V., Pikh, I., & Senkivskyy, V. (2019, September 1). Application of Optics Density-Based clustering algorithm using inductive methods of

complex system analysis. IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/8929869

Balakrishna, Y., Manda, S., Mwambi, H., & Van Graan, A. (2023). Determining classes of food items for health requirements and nutrition guidelines using Gaussian mixture models. Frontiers in Nutrition, 10. https://doi.org/10.3389/fnut.2023.1186221

Bezdek, J. C., Ehrlich, R., & Full, W. (1983). FCM: The fuzzy c-means clustering algorithm. Computers & Geosciences, 10(2-3), 191-203. https://doi.org/10.1016/0098-3004(84)90020-7

Bezdek, J.C. (1981) Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York. http://dx.doi.org/10.1007/978-1-4757-0450-1

Bhattacharjee, P., & Mitra, P. (2020). A survey of density based clustering algorithms. Frontiers of Computer Science, 15(1). https://doi.org/10.1007/s11704-019-9059-3

Birant, D., & Kut, A. (2006). ST-DBSCAN: An algorithm for clustering spatial–temporal data. Data & Knowledge Engineering, 60(1), 208-221. https://doi.org/10.1016/j.datak.2006.01.013

Bushra, A. A., & Yi, G. (2021). Comparative analysis Review of pioneering DBSCAN and successive Density-Based clustering algorithms. IEEE Access, 9, 87918–87935. https://doi.org/10.1109/access.2021.3089036

Caliński, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. Communications in Statistics, 3(1), 1–27. https://doi.org/10.1080/03610927408827101

Cambeses-Franco, C., González‐García, S., Calvo-Malvar, M., Benítez-Estévez, A. J., Leis, R., Sánchez-Castro, J., Gude‐Sampedro, F., Feijoo, G., & Moreira, M. T. (2023). A clustering approach to analyse the environmental and energetic impacts of Atlantic recipes - A Galician gastronomy case study. Journal of Cleaner Production, 383, 135360. https://doi.org/10.1016/j.jclepro.2022.135360

Chanchlani, V., Parsnani, J., Mulani, J., Shetty, P. (n.d.). Diet Recommendation System for Diabetic Patients. Ijera. https://www.ijera.com/special_issue/ICAITR-2101/3,%2010-15.pdf

Charrad, M., Ghazzali, N., Boiteau, V., & Niknafs, A. (2014). NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set. Journal of Statistical Software, 61(6), 1–36. https://doi.org/10.18637/jss.v061.i06

Choe, H., & Jordan, J. B. (1992). On the optimal choice of parameters in a fuzzy c-means algorithm. In 1992 Proceedings IEEE International Conference on Fuzzy Systems (pp. 349-354). San Diego, CA, USA. https://doi.org/10.1109/FUZZY.1992.258640

Cohen, S. (2020). The basics of machine learning: Strategies and techniques. Artificial Intelligence and Deep Learning in Pathology, 13-40. https://doi.org/10.1016/B978-0-323-67538-3.00002-6

Crase, S., & Thennadil, S. N. (2022). An analysis framework for clustering algorithm selection with applications to spectroscopy. PLOS ONE, 17(3), e0266369. https://doi.org/10.1371/journal.pone.0266369

Dalimunthe, S., & Hanafiah, A. (2021). Implementation of agglomerative hierarchical clustering based on the classification of food ingredients content of nutritional substances. IT Journal Research and Development, 6(1), 60–69. https://doi.org/10.25299/itjrd.2021.6872

Data Headhunters (2024, January 7). Handling inconsistent data: Strategies for Standardization. https://dataheadhunters.com/academy/handling-inconsistent-data-strategies-for-standardization/

Davies, D. L., & Bouldin, D. W. (1979). A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1(2), 224-227. https://doi.org/10.1109/TPAMI.1979.4766909

Davies, R., Ghosh-Dastidar, U., Knisley, J., & Samyono, W. (2018). Toward Revealing Protein Function: Identifying Biologically Relevant Clusters With Graph Spectral Methods. Algebraic and Combinatorial Computational Biology, 375-409. https://doi.org/10.1016/B978-0-12-814066-6.00012-X

Dembélé, D., & Kastner, P. (2003). Fuzzy C-means method for clustering microarray data. Bioinformatics, 19(8), 973–980. https://doi.org/10.1093/bioinformatics/btg119

Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Mphago, B., & Tabona, O. (2021). A survey on missing data in machine learning. Journal of Big Data, 8(1). https://doi.org/10.1186/s40537-021-00516-9

Ester, M., Kriegel, H.P., Sander, J., & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. https://www2.cs.sfu.ca/~ester/papers/kdd_96.pdf

EtehadTavakol, M., Sadri, S., & Ng, E. Y. K. (2010). Application of K- and Fuzzy c-Means for Color Segmentation of Thermal Infrared Breast Images. Journal of Medical Systems, 34, 35–42. https://doi.org/10.1007/s10916-008-9213-1

Fahey, M., Ferrari, P., Slimani, N., Vermunt, J. K., White, I. R., Hoffmann, K., Wirfält, E., Bamia, C., Touvier, M., Linseisen, J., Rodríguez‐Barranco, M., Tumino, R., Lund, E., Overvad, K., De Mesquita, B. B., Bingham, S., & Ríboli, E. (2011). Identifying dietary patterns using a normal mixture model: application to the EPIC study. Journal of Epidemiology and Community Health, 66(1), 89–94. https://doi.org/10.1136/jech.2009.103408

Fahey, M., Thane, C. W., Bramwell, G., & Coward, W. A. (2006). Conditional gaussian mixture modelling for dietary pattern analysis. Journal of the Royal Statistical Society. Series a. Statistics in Society/Journal of the Royal Statistical Society. Series a, Statistics in Society, 170(1), 149–166. https://doi.org/10.1111/j.1467-985x.2006.00452.x

Gaio, R., Da Costa, J. P., Santos, A. C., Ramos, E., & Lopes, C. (2012). A restricted mixture model for dietary pattern analysis in small samples. Statistics in Medicine, 31(19), 2137–2150. https://doi.org/10.1002/sim.5336

GeeksforGeeks. (2023, October 17). Handling inconsistent data. GeeksforGeeks. https://www.geeksforgeeks.org/handling-inconsistent-data/

Ghosh, S., & Dubey, S. K. (2013). Comparative Analysis of K-Means and Fuzzy C-Means Algorithms. International Journal of Advanced Computer Science and Applications, 4(4). https://doi.org/10.14569/IJACSA.2013.040406

Gikera, R., Mwaura, J., Muuro, E., & Mambo, S. (2023). K-Hyperparameter Tuning in High-Dimensional Space Clustering: Solving Smooth Elbow Challenges Using an Ensemble Based Technique of a Self-Adapting Autoencoder and Internal Validation Indexes. Journal on Artificial Intelligence, 5, 75-112. https://doi.org/10.32604/jai.2023.043229

Gul, M., & Rehman, M. A. (2023). Big data: An optimized approach for cluster initialization. Journal of Big Data, 10(1), 1-19. https://doi.org/10.1186/s40537-023-00798-1

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001). On Clustering Validation Techniques. Journal of Intelligent Information Systems, 17(2-3), 107-145.

Han, J., Kamber, M., & Pei, J. (2011). Cluster Analysis: Basic Concepts and Methods. Data Mining (Third Edition), 443-495. https://doi.org/10.1016/B978-0-12-381479-1.00010-1

Hashemi, S. E., Gholian-Jouybari, F., & Hajiaghaei–Keshteli, M. (2023). A fuzzy C-means algorithm for optimizing data clustering. Expert Systems With Applications, 227, 120377. https://doi.org/10.1016/j.eswa.2023.120377

JMP (n.d.). Exploratory data analysis. JMP. https://www.jmp.com/en_my/statistics-knowledge-portal/exploratory-data-analysis.html

Kannan, P. K. (2023, August 8). BIRCH Clustering Method: A comprehensive guide for data scientists in DNASEQ and Variants analysis. https://www.linkedin.com/pulse/birch-clustering-method-comprehensive-guide-data-kandavel-phd/

Karypis, G., Han, E. H., & Kumar, V. (1999). CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. Computer. Retrieved from https://www-users.cse.umn.edu/~hanxx023/dmclass/chameleon.pdf

Kishor Duggirala, R. (2020). Segmenting Images Using Hybridization of K-Means and Fuzzy C-Means Algorithms. IntechOpen. doi: 10.5772/intechopen.86374

Kononenko, I., & Kukar, M. (2006). Cluster Analysis. Machine Learning and Data Mining, 321-358. https://doi.org/10.1533/9780857099440.321

Kononenko, I., & Kukar, M. (2006). Cluster Analysis. Machine Learning and Data Mining, 321-358. https://doi.org/10.1533/9780857099440.321

Kotu, V., & Deshpande, B. (2018). Clustering. Data Science (Second Edition), 221-261. https://doi.org/10.1016/B978-0-12-814761-0.00007-1

Krasnov, D., Davis, D. A., Malott, K., Chen, Y., Shi, X., & Wong, A. (2023). Fuzzy C-Means Clustering: A review of applications in breast cancer detection. Entropy, 25(7), 1021. https://doi.org/10.3390/e25071021

Kuan, F. (2022, February 9). Data Cleaning - Remove duplicates. Mage. https://www.mage.ai/blog/data-cleaning-remove-duplicates

Kumar, I., & Singh, S. P. (2021). Machine learning in bioinformatics. Bioinformatics, 443-456. https://doi.org/10.1016/B978-0-323-89775-4.00020-1

Kumar, R., Novak, J., & Tomkins, A. (2006). Structure and evolution of online social networks. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 611–617. Association for Computing Machinery. https://doi.org/10.1145/1150402.1150476

Li, W., & Huang, S. (2021). Research on the prediction method of stock price based on rbf neural network optimization algorithm. E3s Web of Conferences, 235, 03088. https://doi.org/10.1051/e3sconf/202123503088

Li, X., Lü, X., Tian, J., Gao, P., Kong, H., & Xu, G. (2009). Application of fuzzy C-Means clustering in data analysis of metabolomics. Analytical Chemistry, 81(11), 4468–4475. https://doi.org/10.1021/ac900353t

Lloyd, S. (1982). Least squares quantization in PCM. IEEE Transactions on Information Theory, 28(2), 129-137. https://doi.org/10.1109/TIT.1982.1056489

Madhukumar, S., & Santhiyakumari, N. (2015). Evaluation of k-Means and fuzzy C-means segmentation on MR images of brain. The Egyptian Journal of Radiology and Nuclear Medicine, 46(2), 475-479. https://doi.org/10.1016/j.ejrnm.2015.02.008

Mas'ud, A. A., Sundaram, A., Ardila-Rey, J. A., Schurch, R., Muhammad‐Sukki, F., & Bani, N. A. (2021). Application of the Gaussian mixture model to classify stages of electrical tree growth in epoxy resin. Sensors, 21(7), 2562. https://doi.org/10.3390/s21072562

Mingoti, S. A., & Lima, J. O. (2006). Comparing SOM neural network with Fuzzy c-means, K-means and traditional hierarchical clustering algorithms. European Journal of Operational Research, 174(3), 1742-1759. https://doi.org/10.1016/j.ejor.2005.03.039

Mirzaei, G., & Adeli, H. (2022). Machine learning techniques for diagnosis of alzheimer disease, mild cognitive disorder, and other types of dementia. Biomedical Signal Processing and Control, 72, 103293. https://doi.org/10.1016/j.bspc.2021.103293

Mirzaei, G., & Adeli, H. (2022). Machine learning techniques for diagnosis of alzheimer disease, mild cognitive disorder, and other types of dementia. Biomedical Signal Processing and Control, 72, 103293. https://doi.org/10.1016/j.bspc.2021.103293

Mladenov, M. I., Penchev, S., & Deyanov, M. (2014, October). Optical Methods for Food Quality and Safety Assessment – a review. ResearchGate. https://www.researchgate.net/publication/291831376_Optical_methods_for_food_quality_and_safety_assessment_-_a_review

Naghizadeh, A., & Metaxas, D. N. (2019). Condensed Silhouette: An Optimized Filtering Process for Cluster Selection in K-Means. Procedia Computer Science, 176, 205-214. https://doi.org/10.1016/j.procs.2020.08.022

Nayak, J., Rekha, H. S., & Naik, B. (2023). Fuzzy C-Means Clustering: Advances and Challenges (Part II). In L. Rokach, O. Maimon, & E. Shmueli (Eds.), Machine Learning for Data Science Handbook. Springer. https://doi.org/10.1007/978-3-031-24628-9_12

Nettleton, D. (2013). Data Modeling. Commercial Data Mining, 137-157. https://doi.org/10.1016/B978-0-12-416602-8.00009-1

Nettleton, D. (2013). Data Modeling. Commercial Data Mining, 137-157. https://doi.org/10.1016/B978-0-12-416602-8.00009-1

Nguyen, H. T., Jia, G., Shah, Z. K., Pohar, K., Mortazavi, A., Zynger, D. L., Wei, L., Yang, X., Clark, D., & Knopp, M. V. (2015). Prediction of chemotherapeutic response in bladder cancer using K-means clustering of dynamic contrast-enhanced (DCE)-MRI pharmacokinetic parameters. Journal of magnetic resonance imaging : JMRI, 41(5), 1374–1382. https://doi.org/10.1002/jmri.24663

O'Hara, C., O'Sullivan, A., & Gibney, E. R. (2022). A Clustering Approach to Meal-Based Analysis of Dietary Intakes Applied to Population and Individual Data. The Journal of Nutrition, 152(10), 2297-2308. https://doi.org/10.1093/jn/nxac151

O'Hara, C., O'sullivan, A., & Gibney, E. R. (2022). A clustering approach to Meal-Based analysis of dietary intakes applied to population and individual data. National Library of Medicine, 152(10), 2297–2308. https://doi.org/10.1093/jn/nxac151

Pal, N. R., & Bezdek, J. C. (1995). On cluster validity for the fuzzy c-means model. IEEE Transactions on Fuzzy Systems, 3(3), 370-379. https://doi.org/10.1109/91.413225

Panda, S., Sahu, S., Jena, P., & Chattopadhyay, S. (2012). Comparing Fuzzy-C Means and K-Means Clustering Techniques: A Comprehensive Study. In D. Wyld, J. Zizka, & D. Nagamalai (Eds.), Advances in Computer Science, Engineering & Applications, 166, 405–413. Springer. https://doi.org/10.1007/978-3-642-30157-5_45

Pasaribu, J. (2020). Appilication of K-Means algorithm to predict consumer interest according to the season on place reservation and food online software. Journal of Physics. Conference Series, 1477(3), 032004. https://doi.org/10.1088/1742-6596/1477/3/032004

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

Pham, D. L., Xu, C., & Prince, J. L. (2000). Current methods in medical image segmentation. Annual review of biomedical engineering, 2, 315–337. https://doi.org/10.1146/annurev.bioeng.2.1.315

Ponder-Sutton, A. M. (2015). The Automating of Open Source Intelligence. Automating Open Source Intelligence, 1-20. https://doi.org/10.1016/B978-0-12-802916-9.00001-4

Pradipta, I. M. D., Eka, A., Wahyudi, A., & Aryani, S. (2018). Fuzzy c-means clustering for customer segmentation. Int. J. Eng. Emerg. Technol, 3(1), 18-22. Retrieved fromhttps://ojs.unud.ac.id/index.php/ijeet/article/download/41251/25103.

Qarmiche, N., Kinany, K. E., Otmani, N., Rhazi, K. E., & Chaoui, N. E. H. (2023). Cluster analysis of dietary patterns associated with colorectal cancer derived from a Moroccan case–control study. BMJ Health & Care Informatics, 30(1), e100710. https://doi.org/10.1136/bmjhci-2022-100710

Qasrawi, R., Halawa, D. a. A., Ayyad, R., Sabbah, H. A., Taweel, H., & Abdeen, Z. (2021). Cluster analysis for food group consumption patterns in a national sample of Palestinian schoolchildren: evidence from HBSC Survey 2013-2014. Research Gate, 01(1), 33–52. https://doi.org/10.47874/2021p7

Rajendiran, S., Hashemi, M., Frinklea, F., Young, N., Lipke, E., & Cremaschi, S. (2022). Investigating tunable experiment variable effects on hiPSC-CMs maturation via unsupervised learning. Computer Aided Chemical Engineering, 52, 2723-2728. https://doi.org/10.1016/B978-0-443-15274-0.50433-9

Redman, T. C. (2018, April 3). If your data is bad, your machine learning tools are useless. Harvard Business Review. https://hbr.org/2018/04/if-your-data-is-bad-your-machine-learning-tools-are-useless

Reynolds, D. A. (2009). Gaussian Mixture models. In Springer eBooks (pp. 659–663). https://doi.org/10.1007/978-0-387-73003-5_196

Rodriguez, A., & Laio, A. (2014). Machine learning. Clustering by fast search and find of density peaks. Science (New York, N.Y.), 344(6191), 1492–1496. https://doi.org/10.1126/science.1242072

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20, 53-65. https://doi.org/10.1016/0377-0427(87)90125-7

Sander, J., Ester, M., Kriegel, H. P., et al. (1998). Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications. Data Mining and Knowledge Discovery, 2, 169–194. https://doi.org/10.1023/A:1009745219419

Sander, J., Ester, M., Kriegel, H., & Xu, X. (1998). Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. Data Mining and Knowledge Discovery, 2(2), 169–194. https://doi.org/10.1023/a:1009745219419

Sauvageot, N., Schritz, A., Leite, S., Alkerwi, A., Stranges, S., Zannad, F., Streel, S., Hoge, A., Donneau, A. F., Albert, A., & Guillaume, M. (2017). Stability-based validation of dietary patterns obtained by cluster analysis. Nutrition Journal, 16(1). https://doi.org/10.1186/s12937-017-0226-9

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN revisited, revisited. ACM Transactions on Database Systems, 42(3), 1–21. https://doi.org/10.1145/3068335

Sgroi, F., Sciortino, C., Baviera‐Puig, A., & Modica, F. (2024). Analyzing consumer trends in functional foods: A cluster analysis approach. Journal of Agriculture and Food Research, 101041. https://doi.org/10.1016/j.jafr.2024.101041

Simhachalam, B., & Ganesan, G. (2016). Performance comparison of fuzzy and non-fuzzy classification methods. Egyptian Informatics Journal, 17(2), 183-188. https://doi.org/10.1016/j.eij.2015.10.004

Sivarathri, S., & Govardhan, A. (2014). Experiments on Hypothesis: Fuzzy K-Means is Better Than K-Means for Clustering. International Journal of Data Mining & Knowledge Management Process, 4(5), 21–34. https://doi.org/10.5121/ijdkp.2014.4502

Sun, F., Gao, R., Hu, X., Du, Z., & Yu, W. (2017). Experimental study on an effective method for the friction property of fabrics by the comprehensive handle evaluation system for fabrics and yarns system. Textile Research Journal. https://doi.org/10.1177/0040517517690625

Talabis, M. R. M., McPherson, R., Miyamoto, I., Martin, J. L., & Kaye, D. (2014). Analytics Defined. Information Security Analytics, 1-12. https://doi.org/10.1016/B978-0-12-800207-0.00001-0

Tan, P.N., Steinbach, M., & Kumar, V. (2005). Introduction to Data Mining. Pearson.

Torra, V. (2015). On the selection of m for Fuzzy c-Means. In Proceedings of the 2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (pp. 1571-1577). Atlantis Press. https://doi.org/10.2991/ifsa-eusflat-15.2015.224

Wang, X., & Xu, Y. (2019, July). An improved index for clustering validation based on Silhouette index and Calinski-Harabasz index. IOP Conference Series: Materials Science and Engineering, 569(5), 052024. https://doi.org/10.1088/1757-899X/569/5/052024

Wegmann, M., Zipperling, D., Hillenbrand, J., & Fleischer, J. (2021). A review of systematic selection of clustering algorithms and their evaluation. ArXiv. /abs/2106.12792

Windham, C. T., Windham, M. P., Wyse, B. W., & Hansen, R. G. (1985). Cluster analysis to improve food classification within commodity groups. Journal of the American Dietetic Association, 85(10), 1306-1314. https://doi.org/10.1016/S0002-8223(21)03795-0

Wistuba, M., Schilling, N., & Schmidt-Thieme, L. (2015). Learning hyperparameter optimization initializations. In 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA) (pp. 1-10). Paris, France. https://doi.org/10.1109/DSAA.2015.7344817

Wu, K. (2011). Analysis of parameter selections for fuzzy c-means. Pattern Recognition, 45(1), 407-415. https://doi.org/10.1016/j.patcog.2011.07.012

Yera, R., Alzahrani, A. A., Martínez, L., & Rodríguez, R. M. (2023). A Systematic Review on food Recommender Systems for Diabetic patients. International Journal of Environmental Research and Public Health/International Journal of Environmental Research and Public Health, 20(5), 4248. https://doi.org/10.3390/ijerph20054248

Zhang, M., Ma, Y., Junli, L., & Jifu, Z. (2023). A density connection weight-based clustering approach for dataset with density-sparse region. Expert Systems With Applications, 230, 120633. https://doi.org/10.1016/j.eswa.2023.120633

Zhou, T., Song, Z., & Sundmacher, K. (2019). Big Data Creates New Opportunities for Materials Research: A Review on Methods and Applications of Machine Learning for Materials Design. Engineering, 5(6), 1017-1026. https://doi.org/10.1016/j.eng.2019.02.011