engineering proceedings

MDPI

Proceeding Paper

Quantitative Comparison of Machine Learning Clustering Methods for Tuberculosis Data Analysis †

Marlen Kossakov [1], Assel Mukasheva [2,*], Gani Balbayev [3], Syrym Seidazimov [1], Dinargul Mukammejanova [4] and Madina Sydybayeva [4]

[1] Department of Information Technology, Non-Profit JSC "Almaty University of Power Engineering and Telecommunications Named after Gumarbek Daukeyev", 050013 Almaty, Kazakhstan; m.kossakov@aues.kz (M.K.); syreken.ss@gmail.com (S.S.)
[2] School of Information Technology and Engineering, Kazakh-British Technical University, 050000 Almaty, Kazakhstan
[3] Academy of Logistics and Transport, 050012 Almaty, Kazakhstan; g.balbayev@gmail.com
[4] Faculty of Computer Technologies and Cyber Security, International University of Information Technology, 050000 Almaty, Kazakhstan; m.dinargul.14@gmail.com (D.M.); turlen_a@mail.ru (M.S.)
* Correspondence: mukashevascience@gmail.com

† Presented at the 4th International Conference on Communications, Information, Electronic and Energy Systems (CIEES 2023), Plovdiv, Bulgaria, 23–25 November 2023.

Abstract: In many fields, data-driven decision making has become essential due to machine learning (ML), which provides insights that improve productivity and quality of life. A basic machine learning approach called clustering helps find comparable data points. Clustering plays a critical role in the identification of patient subgroups and the customisation of treatment in the context of tuberculosis (TB) research. While prior studies have recognized its utility, a comprehensive comparative analysis of multiple clustering methods applied to TB data is lacking. Using TB data, this study thoroughly assesses and contrasts four well-known machine learning clustering algorithms: spectral clustering, DBSCAN, hierarchical clustering, and k-means. To evaluate the quality of a cluster, quantitative measures such as the silhouette score, Davies–Bouldin index, and Calinski–Harabasz index are utilised. The results provide quantitative insights that enhance comprehension of clustering and guide future research.

Keywords: machine learning; clustering; tuberculosis; data analysis

check for updates

Citation: Kossakov, M.; Mukasheva, A.; Balbayev, G.; Seidazimov, S.; Mukammejanova, D.; Sydybayeva, M. Quantitative Comparison of Machine Learning Clustering Methods for Tuberculosis Data Analysis. *Eng. Proc.* **2024**, *60*, 20. https://doi.org/10.3390/engproc2024060020

Academic Editor: Renato Filjar

Published: 16 January 2024

1. Introduction

In many fields, machine learning (ML) has become a disruptive force that is reshaping industries and having a significant impact on people's lives [1]. Its capacity to glean valuable insights from data has sparked ground-breaking developments in a variety of industries, including banking, healthcare, and transportation. Machine learning (ML) algorithms have become essential tools for data-driven decision making, ranging from recommendation systems and self-driving cars to personalised medical treatments. They enable us to use data to our advantage and generate well-informed predictions that enhance productivity and quality of life [1,2]. Clustering is a basic machine learning technique that involves assembling related data points according to shared features [3]. In scientific study, clustering is essential, especially when it comes to disease analysis. It helps researchers comprehend complicated illnesses, pinpoint patient subgroups, and customise care for improved results [4]. Researchers have found hidden patterns in patient data by using clustering to explore a variety of medical disorders [3].

Clustering approaches have proven beneficial in the setting of tuberculosis (TB), a severe infectious disease that continues to offer considerable global public health issues [5]. A wide range of factors, such as clinical, demographic, and genetic data, are frequently present in tuberculosis data. Manually analysing such data is laborious and error-prone, but machine learning clustering techniques can effectively reveal insights that may be missed by human analysis [6]. By assisting in the identification of unique patient cohorts, clustering enables doctors to make better-informed judgements on interventions and therapies.

Previous research has acknowledged the value of clustering in tuberculosis (TB) studies [6], with some using machine learning (ML) techniques to examine TB patient data. A thorough comparison review of different clustering algorithms is conspicuously absent in the literature, despite the fact that some studies [7,8] have produced insightful results. Thus, by methodically assessing and contrasting the effectiveness of various ML clustering methods when applied to TB data, this research seeks to close this gap. In order to address this gap, this study compares the effectiveness of four popular clustering techniques: spectral clustering, DBSCAN, hierarchical clustering, and k-means clustering. To objectively evaluate the quality of the clusters generated by each approach, measures for cluster evaluation were used, such as silhouette score, Davies–Bouldin index, and Calinski–Harabasz index.

The selection of tuberculosis data as the primary subject of this study is highly significant. With millions of new cases reported each year, tuberculosis (TB) continues to be a global health concern [6]. In actuality, there were about 10 million new cases of tuberculosis globally in 2019 alone [9]. Using clustering approaches to understand the variability of the disease could greatly improve patient management, yield epidemiological insights, and make it easier to devise focused interventions.

This work initiates a thorough exploration of ML clustering techniques applied to TB data. Important insights are intended to be generated by contrasting different clustering methods and using strict assessment metrics. These findings can then guide future research initiatives and improve our understanding of tuberculosis. This study advances the overarching objective of using machine learning to fight infectious illnesses and enhance public health.

2. Methods

2.1. Data Preprocessing

A thorough preprocessing step was performed on the TB dataset [9] to guarantee the consistency and dependability of our study. This includes managing outliers, resolving missing numbers, and using normalisation to standardise the data [2]. All variables were brought to a common scale using data normalisation, avoiding larger-magnitude variables from unduly influencing the clustering process. To guarantee that every feature contributes equally to the clustering results, standardising the features is an essential first step.

The TB dataset, comprising a diverse set of variables related to patient demographics, clinical information, and outcomes, was categorized into four distinct groups to facilitate a more focused analysis:

- Estimation Data: This category includes variables related to the estimation and prediction of TB cases, such as mortality rates and historical case counts.
- Notification Data: Notification data encompass information related to the formal reporting of TB cases to health authorities, including notification rates, geographical locations, and notification-related features.
- Budget Data: Budget data comprises information on financial allocations, expenditures, and resource utilization for TB control and treatment programs.
- Outcome Data: This category encompasses variables associated with TB treatment outcomes, including treatment success rates, patient recovery status.

Principal Component Analysis (PCA) was applied separately to each set of data in order to minimise the dimensionality of the data while keeping the most useful variables from each category. Principal components, or linear combinations of variables, are identified by PCA, a dimensionality reduction technique, as they capture the largest variation in the data [9]. Our goal was to minimise computational complexity and any noise caused by less relevant variables, while maintaining the essential information needed for clustering, by choosing a subset of the most significant principal components [10,11].

For each category (Estimation, Notification, Budget, and Outcome), the top principal components that collectively explained a predetermined percentage of the total variance (e.g., 95) were retained for subsequent clustering analysis. This feature selection approach allowed the distillation of the essential information from the original data while discarding redundant or less informative variables.

2.2. K-Means

One often utilised clustering method, known as K-means clustering, involves the division of data points into K distinct groups based on their degree of similarity. The objective is to reduce the variance, or sum of squared distances, between each data point and the cluster centroids [8]. Given a dataset X with n data points $x_1, x_2, \ldots, x_n$ and an integer k representing the desired number of clusters, the objective is to find k cluster centroids μ_1 , $\mu_2, \ldots, \mu_k$ that minimize the following objective function:

$$J(c,\ \mu) = \sum_{i=1}^{n} \sum_{j=1}^{k} \left\| x_i - \mu_j \right\|^2, \tag{1}$$

where c is a vector of length n containing the cluster assignments for each data point, indicating which cluster each data point belongs to; μ represents the centroid of cluster j; and $\left\| x_i - \mu_j \right\|^2$ denotes the squared Euclidean distance between data point x_i and centroid μ_j.

The K-means approach iteratively minimises the objective function by assigning data points to the centroid that is closest to them and updating the centroids based on the average of the data points within each cluster [12]. The process is continued until convergence, typically defined as the point at which there are no substantial changes in either the cluster assignments or the centroids.

The process is effective in the subsequent stages:

- The process of initialising k cluster centroids can be performed in two ways: randomly or by selecting data points as initial centroids.
- The process involves assigning each individual data point to the centroid that is closest to it, so creating k clusters.
- The centroids can be recalculated by computing the mean of all data points within each cluster.

Continue to iterate through steps 2 and 3 until convergence is achieved, which is often indicated by the absence of substantial alterations in the cluster assignments or centroids.

2.3. Hierarchical Clustering

Hierarchical clustering, a technique known as either agglomerative or divisive, is utilized to construct a hierarchical structure of clusters by iteratively merging or splitting data points based on their degree of similarity [7]. The algorithm produces a hierarchical data structure that can be visually represented as a dendrogram. It is not necessary to predetermine the number of clusters when using hierarchical clustering.

Given a dataset X with n data points $x_1, x_2, \ldots, x_n$, the agglomerative hierarchical clustering algorithm starts with each data point as its own cluster. The nearest clusters are then repeatedly combined to create larger clusters. The linkage criterion, which specifies the distance between clusters to decide which clusters to merge, is the essential component of this strategy. Typical linking techniques consist of the following:

- Single Linkage: Distance between the closest data points in two clusters.
- Complete Linkage: Distance between the farthest data points in two clusters.
- Average Linkage: Average distance between all data point pairs in two clusters.
- Ward's Linkage: Minimizes the increase in total within-cluster variance when merging clusters.

It produces a cluster hierarchy, which is frequently depicted as a dendrogram. The way the algorithm works is as per the Agglomerative Approach:

1. To initiate the clustering process, each individual data point is initially assigned as a separate cluster;
2. The clusters are merged iteratively by considering the nearest clusters according to the selected linkage criterion;
3. The merging process should be continued until either all data points are assigned to a single cluster or a predetermined ending criterion is satisfied.

A hierarchical structure of clusters is provided by hierarchical clustering, enabling flexible interpretation and investigation of various clustering granularities. Dendrograms can be used to visualize it, and the number of clusters need not be specified beforehand. However, for large datasets, hierarchical clustering can be computationally demanding, and the distance measure and linkage method selected can have a big impact on the outcome [7].

2.4. DBSCAN

A density-based clustering algorithm called DBSCAN, or Density-Based Spatial Clustering of Applications with Noise [13], finds clusters based on the density of data points nearby. Additionally, it finds noise points, or outliers, that are not associated with any cluster. The algorithm has the following mathematical definition:

For a dataset X with n data points $x_1, x_2, \ldots, x_n$ and two parameters:

- ε (epsilon): A radius that defines the neighbourhood around each data point;
- MinPts: The minimum number of data points required to form a dense region (including the data point itself).

DBSCAN categorizes data points into three main types:

- Core Point: A data point with at least $MinPts$ data points within its ε-neighborhood;
- Border Point: A data point within the ε-neighbourhood of a Core Point but with fewer than $MinPts$ data points within its own ε-neighborhood;
- Noise Point: The concept of a "Noise Point" refers to a specific data point inside a dataset that does not meet the criteria of being classified as either a "Core Point" or a "Border Point".

The DBSCAN algorithm proceeds as follows:

1. Randomly select an unvisited data point.
2. If the selected point is a Core Point, create a new cluster and add it to the cluster.
3. Expand the cluster by adding all reachable, unvisited Core Points and Border Points to the cluster.
4. Repeat steps 1–3 until no more data points can be added to the cluster.
5. If there are unvisited data points, return to step 1 and start a new cluster.

Large datasets can benefit from DBSCAN because of its computational complexity, which is linear in the amount of data points. It works especially well with datasets that have different cluster densities and cluster shapes [13].

2.5. Spectral Clustering

Using the spectrum characteristics of the data's similarity matrix, the graph-based clustering algorithm known as "spectral clustering" locates clusters. It works very well for finding complex-shaped and non-convex clusters. One of the most important parameters in spectral clustering is the number of clusters (k), which can be chosen using methods such as eigenvalue analysis or visual evaluation of the eigenvectors [14].

It operates as follows:

- The construction of a similarity graph captures the relationships between data points, where edges represent similarities;
- The graph Laplacian, whether unnormalized or normalized, transforms the graph into a format suitable for spectral analysis;

- Eigenvalue decomposition extracts the eigenvectors and eigenvalues of the Laplacian matrix;
- Spectral embedding reduces the dimensionality of the data while preserving cluster structures in lower-dimensional space;
- Clustering is performed on the embedded data using a chosen algorithm, typically K-means.

The creation of the Laplacian matrix and the eigenvalue decomposition that follows provides the central mathematical ideas of spectral clustering. Here is how these are depicted:

Given the similarity matrix S, the unnormalized Laplacian matrix L is computed as:

$$L = D - S, \tag{2}$$

where D is the degree matrix, defined as a diagonal matrix with the sum of similarities for each data point on the diagonal.

Alternatively, the normalized Laplacian matrix L_{norm} is defined as:

$$L_{\text{norm}} = I - D^{-\frac{1}{2}} S D^{-\frac{1}{2}}, \tag{3}$$

where I is the identity matrix.

Eigenvalue decomposition of the Laplacian matrix yields a set of eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ and corresponding eigenvectors $v_1, v_2, \ldots, v_n$.

To create the matrix X_"embed" for spectral embedding, the top k eigenvectors that match the fewest eigenvalues are chosen [14].

2.6. Silhouette Score

A metric called the silhouette score is employed to assess how well a clustering algorithm produces clusters. In comparison to the closest neighbouring cluster, it gauges how similar each data point inside a cluster is to the other data points within that cluster. Higher values of the silhouette score, which goes from −1 to 1, indicate better-defined and well-separated clusters [15].

Let b(i) be the least average distance from data point i to all data points in a different cluster, defined as the cluster to which i does not belong, and let a(i) be the average distance from data point i to all other data points in the same cluster, given a dataset X with n data points. Next, we compute the silhouette score for data point i as follows:

$$S(i) = \frac{b(i) - a(i)}{\max(a(i),\, b(i))}. \tag{4}$$

The silhouette score for the entire dataset is the mean silhouette score across all data points:

$$\text{silhouett escore} = \frac{1}{n} \sum\nolimits_{i=1}^{n} S(i). \tag{5}$$

A silhouette score approaching 1 signifies that a data point is highly aligned with its own cluster and exhibits low alignment with neighbouring clusters, hence indicating a favourable outcome in terms of clustering.

A score in close proximity to zero suggests that the data point is located precisely on or in very close proximity to the decision boundary that separates two adjacent clusters.

A score that is close to −1 suggests the possibility that the data point might have been erroneously assigned to a different cluster [16].

2.7. Davies–Bouldin Index

A metric called the Davies–Bouldin Index is used to evaluate how well a clustering algorithm produces clusters. It accounts for the size of each cluster and calculates the average similarity between it and its most similar cluster. More distinct and well-defined clusters are indicated by a lower Davies–Bouldin Index [17].

Given a dataset X with n data points and k clusters $\mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_\mathrm{k}$ the Davies–Bouldin Index is calculated as follows:

1. For each cluster C_i, calculate its centroid μ_i representing the centre of the cluster;
2. For each cluster C_i, calculate the average distance between each data point in C_i and the centroid μ_i denoted as R_i:

$$R_i = \frac{1}{|C_i|}\sum\nolimits_{x \in C_i} d(x, \mu_i), \tag{6}$$

where $|C_i|$ represents the number of data points in cluster C_i, and $d(x, \mu_i)$ is the distance between data point x and centroid μ_i (e.g., Euclidean distance);

3. For each cluster C_i, calculate the pairwise dissimilarity between cluster C_i and all other clusters C_j (where $j \neq i$) as:

$$D(C_i,\ C_j) = \frac{R_i + R_j}{d(\mu_i, \mu_j)}; \tag{7}$$

4. For each cluster C_i, find the cluster C_j with which it has the highest similarity, i.e., the minimum $D(C_i,\ C_j)$;
5. The Davies–Bouldin Index is then calculated as the average of these maximum similarities across all clusters:

$$\text{Davies–Bouldin Index} = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i} D(C_i, C_j). \tag{8}$$

Better clustering is shown by a lower Davies–Bouldin Index, which denotes more distinct and well-separated clusters.

Since there is no upper bound for the Davies–Bouldin Index, its meaning must be considered in relation to other clustering outcomes. A lower number is invariably preferable [12,18,19].

2.8. Calinski–Harabasz Index

A statistic used to assess the quality of clusters generated by a clustering algorithm is the Calinski–Harabasz Index, sometimes referred to as the Variance Ratio Criterion. By comparing the within-cluster variance to the between-cluster variance, it evaluates the degree of separation between clusters. Better-defined and well-separated clusters are indicated by a higher Calinski–Harabasz Index [20].

Given a dataset X with n data points and k clusters $C_1, C_2, \ldots, C_k$, the Calinski–Harabasz Index is calculated as follows:

1. Calculate the overall mean of the data points, denoted as μ_{total}:

$$\mu_{\text{total}} = \frac{1}{n}\sum_{x \in X} x; \tag{9}$$

2. Calculate the within-cluster variance (W) as the sum of the variances of each cluster:

$$W = \sum\nolimits_{i=1}^{\mathrm{k}} \sum\nolimits_{x \in C_i} \|x - \mu_i\|^2, \tag{10}$$

where μ_i is the centroid of cluster C_i.

3. Calculate the between-cluster variance (B) as the sum of variances between the cluster centroids and the overall mean:

$$B = \sum\nolimits_{i=1}^{k} |C_i| \cdot \|\mu_i - \mu_{total}\|^2, \tag{11}$$

4. Compute the Calinski–Harabasz Index as the ratio of the between-cluster variance (B) to the within-cluster variance (W):

$$\text{Calinski–Harabasz index} = \frac{B/[K-1]}{W/[n-K]}. \tag{12}$$

Better clustering outcomes are indicated by a higher Calinski–Harabasz Index, which denotes clearly separated clusters and a between-cluster variation that is substantially greater than the within-cluster variance.

The interpretation of the index is based on prior clustering results and has no upper bound. In general, a larger value is preferable [20–22].

3. Results

The clustering results for the estimate, notification, budget, and outcome data types are shown in this section. In terms of silhouette scores, Davies–Bouldin scores, and Calinski–Harabasz scores, it seems that K-means with k = 4 and hierarchical clustering with k = 4 do reasonably well. The graphical depiction of experiment scores on estimates data category for various values of k and ε is presented in Figures 1–4. The output of the estimates data category of the TB dataset, which was processed by ML clustering methods, is shown in Figure 5. From the results acquired through the analysis of cluster evaluation metrics, the best value of k for K-means, hierarchical and spectral clustering, and ε for DBSCAN, was selected. Four clusters are produced by K-means and hierarchical clustering, whereas five clusters are produced by the using the template.

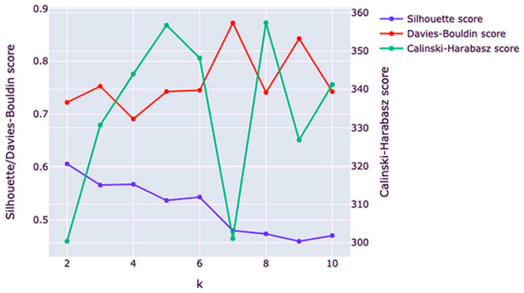

Figure 1. Cluster evaluation metrics result for K-means (Estimations dataset).

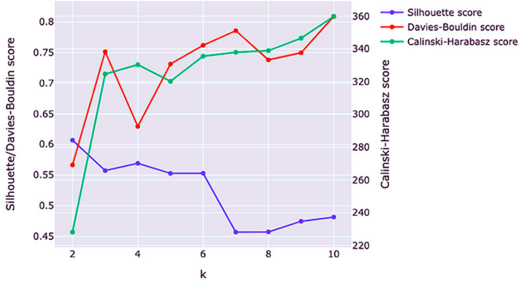

Figure 2. Cluster evaluation metrics result for hierarchical clustering (Estimation dataset).

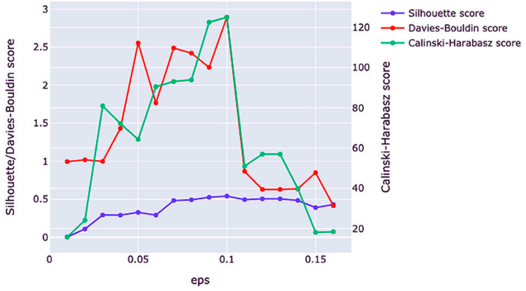

Figure 3. Cluster evaluation metrics result for DBSCAN (Estimation dataset).

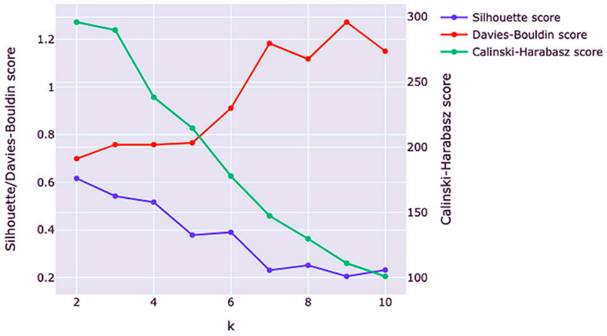

Figure 4. Cluster evaluation metrics result for spectral clustering (Estimation dataset).

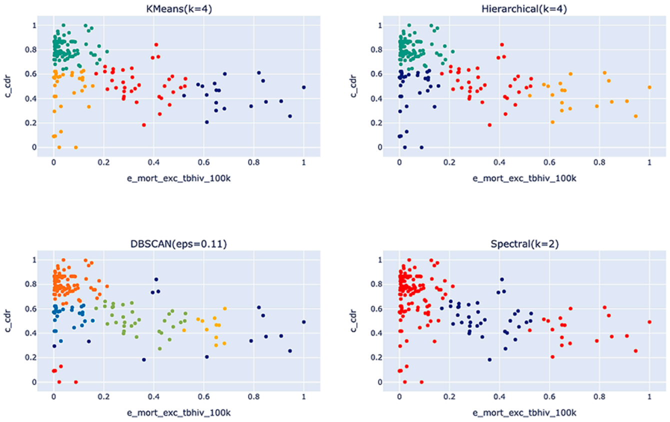

Figure 5. Cluster visualization with best results (Estimation dataset).

Table 1 shows scores for the notification data category of the TB dataset. K-means and hierarchical clustering seem to perform consistently well across different values of k for this data category. Both methods produce clusters with high silhouette scores, indicating well-defined and well-separated clusters. K-means with k = 10 and hierarchical clustering with k = 10 appear to be suitable choices. DBSCAN performs reasonably well, with a stable silhouette score across different epsilon values, indicating relatively well-separated clusters. However, it may not be as effective as K-means or hierarchical clustering in this scenario. Spectral clustering shows good performance when k = 3, but its silhouette score decreases as the number of clusters increases. This suggests that it might be challenging to identify more than three well-separated clusters in the "Notification" data category using spectral clustering. Figure 6 illustrates clustering methods with their best "k" and "ε" values for this data category.

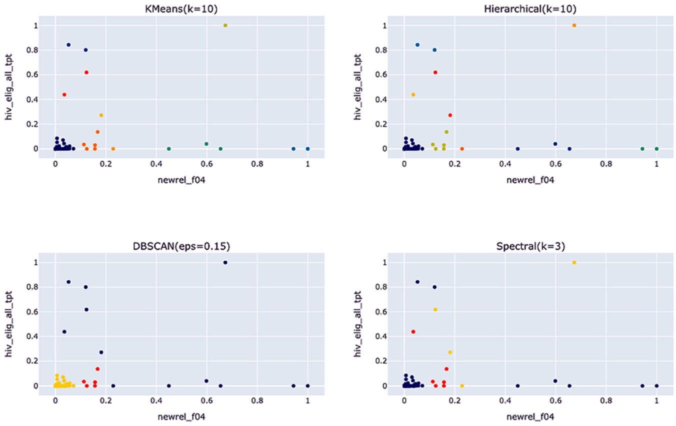

Figure 6. Cluster visualization with best results (Notification dataset).

Table 1. Notifications data category results.

Method	Silhouette Score	Davies–Bouldin Score	Calinski–Harabasz Score
KMeans (k = 2)	0.921888	0.896547	233.083908
KMeans (k = 3)	0.942814	0.460119	564.180086
KMeans (k = 4)	0.936618	0.372054	482.829735
KMeans (k = 5)	0.937319	0.265713	700.255366
KMeans (k = 6)	0.930778	0.360036	825.631145
KMeans (k = 7)	0.885819	0.387040	1084.524869
KMeans (k = 8)	0.883502	0.346847	1548.910942
KMeans (k = 9)	0.883538	0.304381	1781.357804
KMeans (k = 10)	0.882582	0.213537	1962.860713
Hierarchical (k = 2)	0.924705	0.901919	230.272882
Hierarchical (k = 3)	0.942814	0.460119	564.180086
Hierarchical (k = 4)	0.941624	0.318242	627.803845
Hierarchical (k = 5)	0.937319	0.265713	700.255366
Hierarchical (k = 6)	0.895864	0.407496	1021.746228
Hierarchical (k = 7)	0.895045	0.415851	1312.444535
Hierarchical (k = 8)	0.886553	0.390947	1669.164844
Hierarchical (k = 9)	0.809496	0.488679	1827.003481
Hierarchical (k = 10)	0.807232	0.436929	2133.874560
DBSCAN (ε = 0.01)	0.806189	1.244112	86.984049
DBSCAN (ε = 0.02)	0.854295	1.133252	120.141415
DBSCAN (ε = 0.03)	0.878016	1.058013	147.406641
DBSCAN (ε = 0.04)	0.885626	1.026981	159.757449
DBSCAN (ε = 0.05)	0.885626	1.026981	159.757449
DBSCAN (ε = 0.06)	0.911349	0.934416	209.653276
DBSCAN (ε = 0.07)	0.911349	0.934416	209.653276
DBSCAN (ε = 0.08)	0.916373	0.911491	222.581546
DBSCAN (ε = 0.09)	0.916373	0.911491	222.581546
DBSCAN (ε = 0.10)	0.916373	0.911491	222.581546
DBSCAN (ε = 0.11)	0.921888	0.896547	233.083908
DBSCAN (ε = 0.12)	0.921888	0.896547	233.083908
DBSCAN (ε = 0.13)	0.921888	0.896547	233.083908
DBSCAN (ε = 0.14)	0.924705	0.901919	230.272882
DBSCAN (ε = 0.15)	0.924705	0.901919	230.272882
DBSCAN (ε = 0.16)	0.924705	0.901919	230.272882
Spectral (k = 2)	0.927732	0.861168	228.212784
Spectral (k = 3)	0.940250	0.400739	485.133303
Spectral (k = 4)	0.854569	0.614904	436.856959
Spectral (k = 5)	0.809416	0.930598	365.402493
Spectral (k = 6)	0.533718	0.924874	291.017199
Spectral (k = 7)	−0.061049	16.544091	254.267550
Spectral (k = 8)	−0.230689	6.928392	197.336032
Spectral (k = 9)	−0.297636	10.410599	171.848693
Spectral (k = 10)	−0.233370	5.421388	149.588401

From results shown in Table 2 as in the previous analysis, the choice of the best clustering method and parameter configuration should consider the specific goals of your analysis and the context of the data. K-means and hierarchical clustering appear to be strong candidates based on these evaluation scores for "Budget" data category. In Figure 7, the most suitable cluster numbers for each method are illustrated.

Figures 8–12 illustrate that across all clustering methods, it appears that the "Outcome" data category is challenging to cluster effectively. The silhouette scores decrease with increasing cluster numbers for both K-means (Figure 8) and hierarchical clustering (Figure 9) indicating that the data might not have clear natural clusters. DBSCAN also struggles to find well-separated clusters, as indicated by its low silhouette scores and high Davies–Bouldin scores across different epsilon values (Figure 10). The choice of epsilon significantly impacts the results, but none of the configurations seem to produce strong clusters. Spectral clustering follows a similar trend, with decreasing silhouette scores as k increases. This suggests that it is challenging to identify meaningful clusters in this data category using spectral clustering (Figure 11). The "Outcome" data category might inherently lack clear clusters, making it difficult to achieve strong clustering results. Due to this limitation, minimum k and ε values were chosen, Figure 12 illustrates the clustering of the "Outcome" data category.

Table 2. Budget data category results.

Method	Silhouette Score	Davies–Bouldin Score	Calinski–Harabasz Score
KMeans (k = 2)	0.942705	0.289393	625.038025
KMeans (k = 3)	0.905911	0.402616	720.649708
KMeans (k = 4)	0.902634	0.315524	732.222635
KMeans (k = 5)	0.861461	0.391989	904.214898
KMeans (k = 6)	0.885724	0.413518	1143.276872
KMeans (k = 7)	0.881099	0.390891	1746.970170
KMeans (k = 8)	0.882095	0.267941	1786.333354
KMeans (k = 9)	0.823738	0.537225	1981.053036
KMeans (k = 10)	0.849126	0.356672	2011.028563
Hierarchical (k = 2)	0.942705	0.289393	625.038025
Hierarchical (k = 3)	0.905911	0.402616	720.649708
Hierarchical (k = 4)	0.906682	0.359964	769.083541
Hierarchical (k = 5)	0.864937	0.636404	902.603930
Hierarchical (k = 6)	0.883104	0.409142	1117.594503
Hierarchical (k = 7)	0.882580	0.364382	1692.680790
Hierarchical (k = 8)	0.879857	0.456697	1782.923876
Hierarchical (k = 9)	0.813840	0.508241	1974.335965
Hierarchical (k = 10)	0.810734	0.414324	2284.732735
DBSCAN (ε = 0.01)	0.786429	1.070687	109.283498
DBSCAN (ε = 0.02)	0.809581	1.026351	126.473773
DBSCAN (ε = 0.03)	0.830719	0.982316	146.991170
DBSCAN (ε = 0.04)	0.815291	1.142252	104.125035
DBSCAN (ε = 0.05)	0.897377	0.788539	289.932073
DBSCAN (ε = 0.06)	0.906467	0.734014	337.825357
DBSCAN (ε = 0.07)	0.914212	0.679914	392.062907
DBSCAN (ε = 0.08)	0.920791	0.624156	452.104032
DBSCAN (ε = 0.09)	0.920791	0.624156	452.104032
DBSCAN (ε = 0.10)	0.920791	0.624156	452.104032
DBSCAN (ε = 0.11)	0.920791	0.624156	452.104032
DBSCAN (ε = 0.12)	0.920791	0.624156	452.104032
DBSCAN (ε = 0.13)	0.924631	0.581839	493.927175
DBSCAN (ε = 0.14)	0.942705	0.289393	625.038025
DBSCAN (ε = 0.15)	0.942705	0.289393	625.038025
DBSCAN (ε = 0.16)	0.942705	0.289393	625.038025
Spectral (k = 2)	0.942705	0.289393	625.038025
Spectral (k = 3)	0.916820	0.533065	596.430729
Spectral (k = 4)	0.871292	0.573779	575.026763
Spectral (k = 5)	0.833139	0.661008	433.307591
Spectral (k = 6)	0.662441	0.648925	348.889444
Spectral (k = 7)	0.672314	0.584286	381.794204
Spectral (k = 8)	0.375670	0.631871	250.780582
Spectral (k = 9)	0.360341	0.615598	218.433071
Spectral (k = 10)	0.333171	0.678220	247.337298

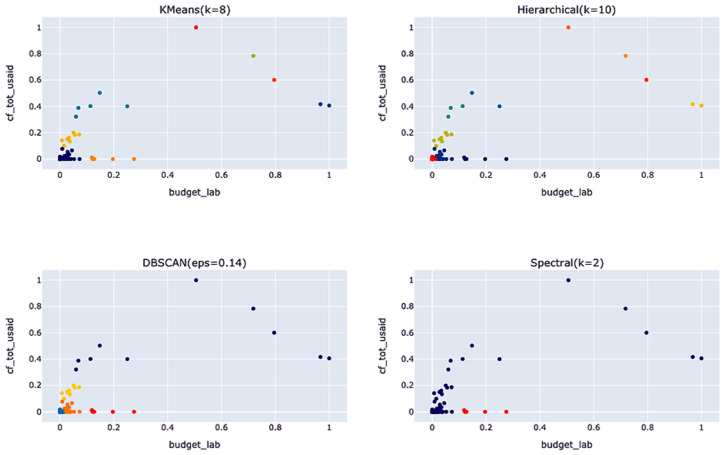

Figure 7. Cluster visualization with best results (Budget dataset).

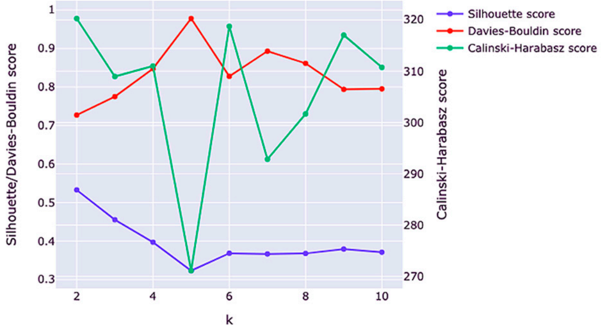

Figure 8. Cluster evaluation metrics result for K-means (Outcome dataset).

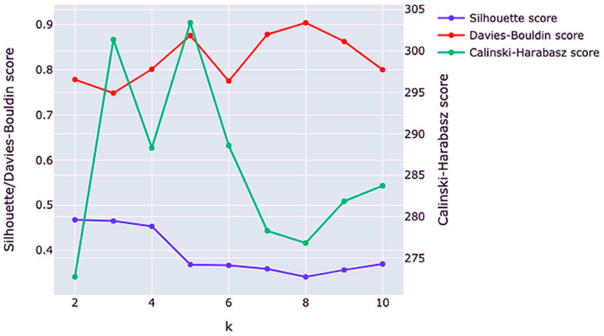

Figure 9. Cluster evaluation metrics result for hierarchical clustering (Outcome dataset).

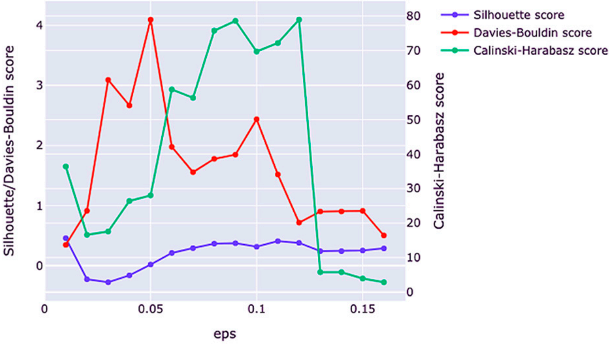

Figure 10. Cluster evaluation metrics result for DBSCAN (Outcome dataset).

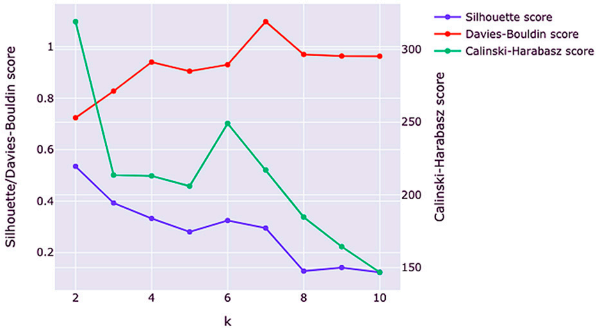

Figure 11. Cluster evaluation metrics result for DBSCAN (Outcome dataset).

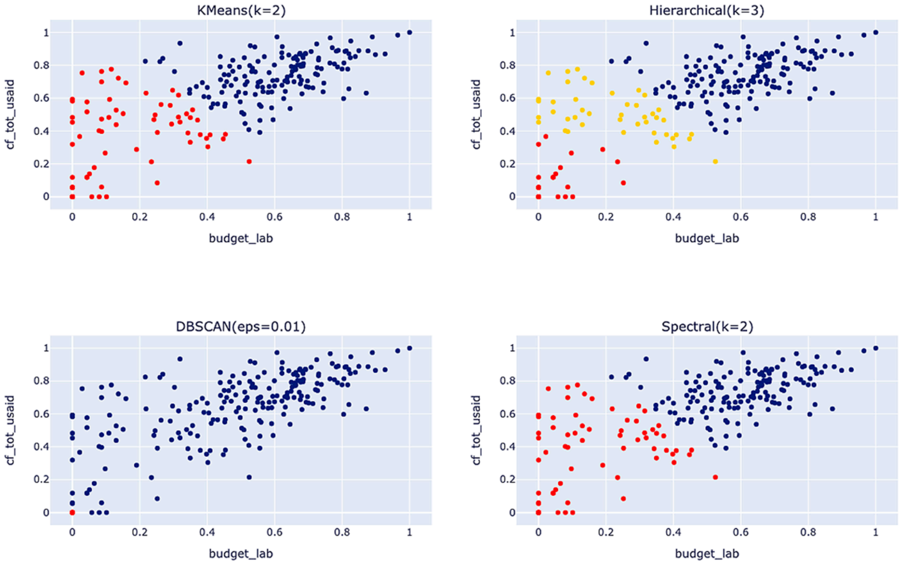

Figure 12. Cluster evaluation metrics result for DBSCAN (Outcome dataset).

4. Conclusions

Four different clustering techniques—K-means, hierarchical clustering, DBSCAN, and spectral clustering—were assessed in this extensive clustering analysis across four different data categories (Estimate, Notification, Budget, and Outcome) from the World Health Organization's TB data collection. Three evaluation criteria were used to evaluate each method's performance: the silhouette score, the Davies–Bouldin score, and the Calinski–Harabasz score.

An increase in the number of clusters (k) was found to frequently lead to a decrease in silhouette scores across all data categories and clustering techniques. This finding revealed that the identification of relevant and well-separated groups grew increasingly difficult when the data was divided into more clusters.

The neighbourhood radius, ε, was discovered to have a significant impact on DBSCAN's performance. In terms of silhouette scores, smaller ε values typically yielded better results, indicating denser, more distinct clusters. Nevertheless, DBSCAN had trouble producing robust clusters for the majority of the epsilon values it looked at.

The "Outcome" data category consistently produced higher Davies–Bouldin scores and lower silhouette scores when compared to other data categories across all clustering algorithms. This recurring pattern implied that there might not be any distinct natural groups in the "Outcome" data by nature.

The features of the dataset and the particular goals of the analysis will determine which clustering approach is used and how the results are evaluated. Subsequent studies ought to concentrate on customising clustering strategies to the distinct difficulties posed by various data kinds and on investigating novel methods to reveal significant patterns in the data.

Author Contributions: Conceptualization, M.K. and A.M., D.M.; methodology, S.S., A.M. and G.B.; software, M.K. and A.M.; validation, A.M., G.B. and M.S.; formal analysis, A.M. and G.B.; investigation, M.K., A.M., G.B. and M.S.; resources, M.K.; data curation, M.K. and A.M.; writing—original draft preparation, M.K. and A.M.; writing—review and editing, M.K., A.M., G.B. and S.S.; visualization, M.K. and A.M.; supervision, G.B.; project administration, A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in reference number [9].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhou, Z.-H. *Machine Learning*; Springer Nature: Singapore, 2021; XIII, 459p.
2. Raschka, S. *Python Machine Learning*; Packt Publishing Ltd.: Birmingham, UK, 2015; 454p.
3. Wardani, R.S.; Sayono, P.; Paramananda, A. Clustering tuberculosis in children using K-Means based on geographic information system. *AIP Conf. Proc.* **2019**, *2114*, 060012. [CrossRef]
4. Momahhed, S.S.; Emamgholipour Sefiddashti, S.; Minaei, B.; Shahali, Z. K-means clustering of outpatient prescription claims for health insureds in Iran. *BMC Public Health* **2023**, *23*, 788. [CrossRef] [PubMed]
5. Dookie, N.; Padayatchi, N.; Naidoo, K. Tuberculosis elimination in the era of coronavirus disease 2019 (COVID-19): A moving target. *Clin. Infect. Dis.* **2022**, *74*, 509–510. [CrossRef] [PubMed]
6. Orjuela-Canon, A.D.; Jutinico, A.L.; Awad, C.; Vergara, E.; Palencia, A. Machine learning in the loop for tuberculosis diagnosis support. *Front. Public Health* **2022**, *10*, 876949. [CrossRef] [PubMed]
7. Jafarzadegan, M.; Safi-Esfahani, F.; Beheshti, Z. Combining hierarchical clustering approaches using the PCA method. *Expert Syst. Appl.* **2019**, *137*, 1–10. [CrossRef]
8. Sinaga, K.P.; Yang, M.-S. Unsupervised K-Means Clustering Algorithm. *IEEE Access* **2020**, *8*, 80716–80727. [CrossRef]
9. WHO. Global Tuberculosis Report, 27 October 2022. Available online: https://www.who.int/ (accessed on 7 July 2023).
10. Reddy, G.T.; Reddy, M.P.; Lakshmanna, K.; Kaluri, R.; Rajput, D.S.; Srivastava, G.; Baker, T. Analysis of Dimensionality Reduction Techniques on Big Data. *IEEE Access* **2020**, *8*, 54776–54788. [CrossRef]
11. Zhao, H.; Zheng, J.; Xu, J.; Deng, W. Fault Diagnosis Method Based on Principal Component Analysis and Broad Learning System. *IEEE Access* **2019**, *7*, 99263–99272. [CrossRef]
12. Ashari, I.; Banjarnahor, R.; Farida, D.; Aisyah, S.; Dewi, A.; Humaya, N. Application of Data Mining with the K-Means Clustering Method and Davies Bouldin Index for Grouping IMDB Movies. *J. Appl. Inform. Comput.* **2022**, *6*, 7–15. [CrossRef]
13. Deng, D. DBSCAN Clustering Algorithm Based on Density. In Proceedings of the 2020 7th International Forum on Electrical Engineering and Automation (IFEEA), Hefei, China, 25–27 September 2020; pp. 949–953. [CrossRef]
14. Sun, G.; Cong, Y.; Dong, J.; Liu, Y.; Ding, Z.; Yu, H. What and How: Generalized Lifelong Spectral Clustering via Dual Memory. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 3895–3908. [CrossRef] [PubMed]
15. Shahapure, K.R.; Nicholas, C. Cluster Quality Analysis Using Silhouette Score. In Proceedings of the 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), Sydney, NSW, Australia, 6–9 October 2020; pp. 747–748.
16. Ogbuabor, G.; Ugwoke, F. Clustering algorithm for a healthcare dataset using silhouette score value. *Int. J. Comput. Sci. Inf. Technol. (IJCSIT)* **2018**, *10*, 27–37. [CrossRef]
17. Mughnyanti, M.; Efendi, S.; Zarlis, M. Analysis of determining centroid clustering x-means algorithm with davies-bouldin index evaluation, In Proceedings of the IOP Conference Series: Materials Science and Engineering. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *725*, 012128. [CrossRef]
18. PyShark. Davies-Bouldin Index for K-Means Clustering Evaluation in Python, 2021, PythonBloggers.com. Available online: https://pythonbloggers.com/2021/06/davies-bouldin-index-fork-means-clustering-evaluation-in-python (accessed on 7 July 2023).
19. Yedilkhan, D.; Mukasheva, A.; Bissengaliyeva, D.; Suynullayev, Y. Performance Analysis of Scaling NoSQL vs SQL: A Comparative Study of MongoDB, Cassandra, and PostgreSQL. In Proceedings of the 2023 IEEE International Conference on Smart Information Systems and Technologies (SIST), Astana, Kazakhstan, 4–6 May 2023; pp. 479–483. [CrossRef]
20. Maulik, U.; Bandyopadhyay, S. Performance evaluation of some clustering algorithms and validity indices. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 1650–1654. [CrossRef]

21. Ashari, I.; Dwi Nugroho, E.; Baraku, R.; Novri Yanda, I.; Liwardana, R. Analysis of Elbow, Silhouette, Davies-Bouldin, Calinski-Harabasz, and Rand-Index Evaluation on K-Means Algorithm for Classifying Flood-Affected Areas in Jakarta. *J. Appl. Inform. Comput.* **2023**, *7*, 95–103. [CrossRef]
22. Mukasheva, A.; Koishiyev, D.; Suimenbayeva, Z.; Rakhmatullaeva, S.; Bolshibayeva, A.; Sadikova, G. Comparison evaluation of Unet-based models with noise augmentation for breast cancer segmentation on ultrasound image. *East.-Eur. J. Enterp. Technol.* **2023**, *5*, 85–97. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.