JOURNAL
COMPUTER SCIENCES INSTITUTE

JCSI 38 (2026) 43-50
Received: 19 September 2025
Accepted: 18 October 2025

Clustering methods in machine learning

Bartłomiej Głuszczak*, Paweł Powroźnik

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

Clustering is a fundamental task in machine learning for discovering hidden structures in unlabelled data. The article reviews key clustering methods, including centroid, density, hierarchy and model-based approaches. Their advantages, limitations and applications are analysed to provide a comprehensive overview of the state of clustering in machine learning. Their effectiveness is compared on the basis of selected metrics to evaluate the outcome of a given clustering. Recent developments and challenges, including scalability and interpretability problems, are also discussed.

Keywords: clustering; machine learning; unsupervised learning; algorithms; data analysis; clustering method

*Corresponding author

Email address: s95406@pollub.edu.pl (B. Głuszczak)

1. Introduction

Machine learning is a rapidly developing field in which clustering methods play a key role in unsupervised data analysis [1]. Clustering partitions data into subsets (clusters) in such a way that objects within the same cluster are more similar to each other than to objects from other clusters.

The development of information technology and the growing amount of data generated in various fields, such as medicine, economics, and natural sciences, require increasingly advanced analysis tools. In this context, clustering becomes not only a tool that supports the discovery of patterns in data, but also a key element of recommendation systems image analysis, and anomaly detection systems.

There are many different approaches to clustering that are tailored to the specifics of the data and the objectives of the analysis. Each of these approaches offers unique capabilities and has specific limitations.

2. Literature review

With the rise in popularity of machine learning around the world over the past decade, the topic of data clustering has been rapidly developed in scientific literature

Based on one of the chapters of the book [2] by Ch. K. Reddy and Ch. C. Aggarwal, we learn that despite the many existing methods, there are some that are used in most cases, making them the most popular. One of these is the clustering method with divisions.

In one of his articles [3], J. McQueen presented the K-means algorithm – the most popular algorithm belonging to this group of methods. It enables effective data clustering by optimizing the objective function measured by Euclidean distances. This research sparked the development of numerous modifications of the method, which were adapted to various applications, such as image analysis and market segmentation.

They are effective for large data sets, but depend on the random initial setting of cluster means.

When expressed using mathematical formulas, the algorithm looks as follows:

- Assigning each data point (x_i) to the nearest cluster centre (μ_j) (formula 1).

$$\min_j \|x_i - \mu_j\|^2 \quad (1)$$

- Update cluster measures as the average of data points assigned to the cluster (formula 2).

$$\mu_j = \frac{1}{|C_j|} \sum_{x_i \epsilon C_j} x_i, \quad (2)$$

where $|C_j|$ is the number of clusters. These two steps are repeated until the cluster centres do not change their positions significantly.

Another work by L. Kaufman and P. J. Rousseeuw [4] discusses hierarchical clustering methods, which involve the creation of tree-like structures resembling binary trees, called dendrograms (Fig. 1), which help to visualize the data clustering process. Dendrograms show the direct relationship between clusters. These methods are intuitive, but also involve high computational costs for large data sets.

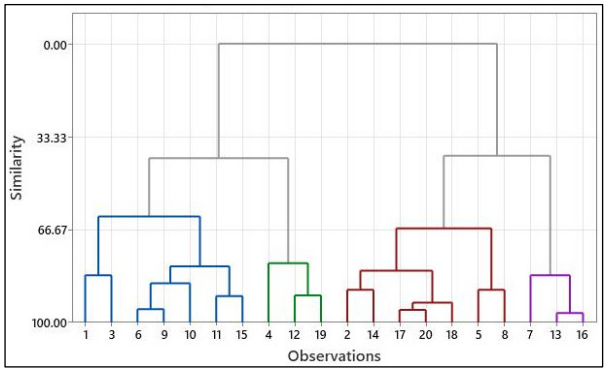

Figure 1: Dendrogram showing data clustering.

Other very popular clustering methods are density-based methods. Algorithms such as DBSCAN identify clusters as areas with high point density separated by areas with low density. In M. Ester's work, we learn that these methods work well with irregularly shaped clusters

(Fig. 2), but require the selection of appropriate parameters [5].

Density criteria can be determined using the following formula:

$$MinPts \leq |N_\varepsilon(p)| \quad (3)$$

where $N_\varepsilon(p)$ denotes the set of points within distance ε of point p (the ε neighborhood), and $MinPts$ is the minimum number of points required for the region to be considered high-density (typically with distance measured in the chosen metric e.g., Euclidean).

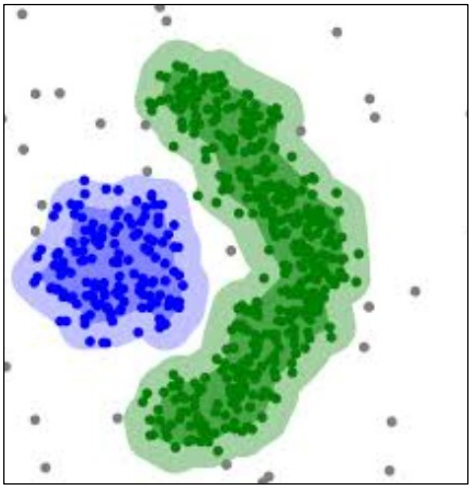

Figure 2: Example result of grouping using the DBSCAN algorithm.

3. Research methodology

The purpose of this chapter is to present in detail the experimental procedure that enabled a reliable comparison of selected clustering algorithms on a selected data set.

It outlines the data preprocessing steps, parameter settings, evaluation metrics, and stability checks used to ensure that differences in performance can be attributed to the algorithms rather than artifacts of the pipeline.

3.1. Clustering methods

The choice of clustering algorithms determines both the quality of the segments obtained and the possibility of interpreting the results in a business context. This chapter presents five complementary clustering methods: K-means, Gaussian Mixture Model (GMM), DBSCAN, Mean Shift, and BIRCH. Each of the described algorithms is characterized by different sensitivity to data scale, parameterization, and cluster shape.

A discussion of their theoretical assumptions, computational complexity, and practical advantages and limitations allows not only for a comparison of the effectiveness of the tools, but also for the identification of scenarios in which a given algorithm can bring the greatest analytical benefits.

3.1.1.K-Means Method

The K-means method is one of the most commonly used and popular clustering methods [6]. It is a widely used algorithm, largely due to its simplicity and efficiency [7]. This method involves dividing a data set into a predetermined number of clusters [8], in which random points are initially designated as the central points, and then all elements of the set are assigned to centroids using a distance measure, usually Euclidean distance. Finally, based on the arithmetic mean of all cluster values, the correct centroid for each cluster is indicated [9]. Thanks to its speed in handling large data sets, the K-means method has found its place in many fields, such as commerce and transaction data [10].

The operation of the K-means algorithm can be based on five steps:

1. Select the number of clusters *k.*
2. Freely select central points.
3. Assign each point to one of the clusters.
4. Calculate the new central point based on the arithmetic mean of all points in the cluster.

The grouping result is largely influenced by the distribution of the original central points (step 2), which means that the final result may not always be sufficiently optimal.

3.1.2. GMM Method

The Gaussian Mixture Model (GMM) is another popular [11] clustering technique in unsupervised learning. GMM uses probabilistic [12] assignment of points to clusters, which means that it does not assign a given point to a specific cluster directly, but is based on measures of the probability that an observation may belong to a selected group. Gaussian mixture models are an extremely versatile technique which, unlike the K-means method, can determine cluster membership for data with overlapping distributions. In practice, model parameters are estimated via the Expectation–Maximization algorithm, and the chosen covariance parameterization (full, diagonal, tied, or spherical) controls the admissible cluster shapes and degree of overlap.

3.1.3. DBSCAN Method

DBSCAN is a density-based clustering method that requires two parameters instead of the number of clusters: the proximity radius ε (eps) and the minimum number of points minPts. The key idea is to treat clusters as areas of sufficient density, separated by zones of sparser observations.

The algorithm goes through all points one by one and classifies them as core, edge, or noise. A core point has at least minPts neighbours within radius ε; all objects reachable by a chain of cores form a single cluster. Objects lying within ε of a core but too sparse to become a core are marked as edge and attached to the nearest cluster [13]. Points not assigned to any core are marked as noise, which ensures natural detection of outliers.

DBSCAN detects clusters of any shape, handles different group sizes, and eliminates the need to guess *k*. The ε parameter is most often selected from the k-distance graph using the elbow method , which signals the transition between dense and sparse regions. A limitation is its sensitivity to parameter selection and decreasing

effectiveness in data with varying density or very high dimension, where the concept of density becomes blurred.

3.1.4. Mean-Shift Method

Mean Shift works by shifting points toward the mean until they stop at the maximum data density. This is where it gets its name. By recognizing the density of points, the Mean Shift algorithm is able to generate a reasonable number of necessary clusters on its own [14]. Mean Shift is called a parameterless clustering algorithm [15], which does not require prior assumptions about the number of clusters or their shapes.

The method works well with irregular data structures. It has found particular application in image processing. Tasks such as image segmentation, object tracking, or real-time applications.

Unfortunately, Mean Shift also has disadvantages, one of which is cost [16]. The algorithm requires significant computing resources, which significantly affects the operation time, especially with large data sets.

3.1.5. BIRCH Method

BIRCH is a clustering algorithm designed for very large data sets that creates a compact representation of a CF (Clustering Feature) tree.

BIRCH processes complex data sets with minimal effort, while remaining efficient and highly scalable. It begins by dividing large data sets into more compact ones, eliminating the need to devote enormous resources to working on the entire data set.

BIRCH has a time and memory complexity close to O(n) and can process streaming data, which is why it is popular in event log mining and is one of the most effective algorithms for quickly analysing hundreds of millions of records. Despite the strengths of the algorithm, an overly large CF tree may be created, leading to increased memory consumption costs [17]. However, the appropriate selection of threshold parameters (threshold, branching factor) and a possible condensation phase can significantly reduce this problem without compromising the quality of segmentation. Furthermore, the hierarchical structure of the CF-tree can later serve as input for more accurate secondary clustering methods, making BIRCH a useful module in complex big data analytics pipelines.

3.2. Data collection

The dataset used for the research was downloaded from the publicly available platform Kaggle.com.

The set contains approximately 9,000 records compiling data on consumers and their credit card transactions. It was developed to segment customers based on purchasing patterns. This data can be put to practical use in recommendation and marketing systems, for example.

The set (Fig. 3) includes attributes such as balance, total purchase amount, information about installments purchases or interest.

BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE
0.875	163.31	0	163.31	0
0.75	0	0	0	1153.925205
1	441	0	441	0
0.75	81.2	81.2	0	419.341394
1	1105.74	0	1105.74	2220.336732
0.125	0	0	0	921.973821
1	1142.12	0	1142.12	1113.924335
0.5	0	0	0	19.712686
0.571429	171	66.5	104.5	0
1	0	0	0	460.284824
1	384.81	169	215.81	0
1	85.89	0	85.89	0
1	1208.28	796.3	411.98	76.915478
1	492.03	0	492.03	0
1	180.25	0	180.25	0
0.285714	0	0	0	157.253266
1	280	0	280	254.298735
1	161.46	38.4	123.06	0
0.857143	958.8	958.8	0	566.910978
1	291.62	0	291.62	1596.771919
1	2806.78	2588.53	218.25	0
0.714286	550	50	500	388.81823

Figure 3: Fragment of a data set.

Before starting the research, it was necessary to perform preliminary data cleaning to remove missing information. Cells containing missing data were filled in using the median. The column with the customer ID, which did not provide key information for the clustering algorithms, was removed. Finally, the data was normalized using standardization.

This collection contains diverse attributes, making it a good environment for testing different clustering methods and verifying how algorithms will separate customer groups.

3.3. Implementation of algorithms

This chapter presents details of the implementation of five clustering algorithms—K-means, Gaussian Mixture Model (GMM), DBSCAN, Mean Shift, and BIRCH-along with the data preparation process, parameter selection, and a common execution template.

The whole thing was implemented in Python 3.11 using the pandas 2.2, NumPy 1.26, and scikit-learn libraries. All experiments were conducted in the Google Colab environment.

Each clustering method was called in the cluster_data function (listing 1).

Listing 1: Function invoking the clustering algorithm

```
def cluster_data(
    X: np.ndarray, method: str, **kwargs
) -> Tuple[np.ndarray, Any, float, int]:
    method = method.lower()
    start = time.perf_counter()
    if "kmeans" in method:
        model = KMeans(**kwargs, n_init="auto", random_state=42)
        labels = model.fit_predict(X)
        n_clusters = model.n_clusters
    elif "gmm" in method:
        model = GaussianMixture(**kwargs, random_state=42)
        labels = model.fit_predict(X)
        n_clusters = model.n_components
    elif method == "dbscan":
        model = DBSCAN(**kwargs)
        labels = model.fit_predict(X)
        n_clusters = len(set(labels)) - (1 if -1 in labels else 0)
    elif method == "meanshift":
        bandwidth = 11
        model = MeanShift(**kwargs)
        labels = model.fit_predict(X)
        n_clusters = len(np.unique(labels))
    elif "birch" in method:
        model = Birch(**kwargs)
        labels = model.fit_predict(X)
        n_clusters = len(np.unique(labels))
    else:
        raise ValueError(f"Nieznana metoda: {method}")
    runtime = time.perf_counter() - start
    return labels, model, runtime, n_clusters
```

This function takes a data matrix, the name of the method, and its input parameters as input. The clustering

algorithms used were implemented using the Scikit-learn library.

The implementation process consisted of five repeatable steps:

1. Configuration of parameters specific to a given method.
2. Training the model on the feature matrix X.
3. Reducing the dimensions of the dataset for better visualization.

Visualizing the data on a two-dimensional graph.

4. Calculating the clustering quality metrics.

3.4. Selection of parameters

Each of the implemented methods accepts different input parameters. Determining the appropriate parameter values results in more accurate clustering results, which is why this stage is the most crucial.

The K-Means method requires the specified number of clusters that we want to obtain as a result of clustering. In order to find the most optimal number, an elbow curve graph is used. It involves calculating the sum of the squares of the distances of all points from the centroids of their clusters, otherwise known as WCSS (Within-Cluster Sum of Squares) or often referred to as inertia calculation (formula 3).

$$WCSS = \sum_{i=1}^{k} \sum_{j=1}^{n_i} distance(x_j^{(i)}, c_i)^2, \quad (3)$$

where

- k – number of clusters;
- n –maximum number of clusters;
- $distance(x_j^{(i)}, c_i)^2$ – Euclidean distance square.

The WCSS summary on a line graph (Fig. 4) for different k values helps to find the optimal value (the point where the graph breaks).

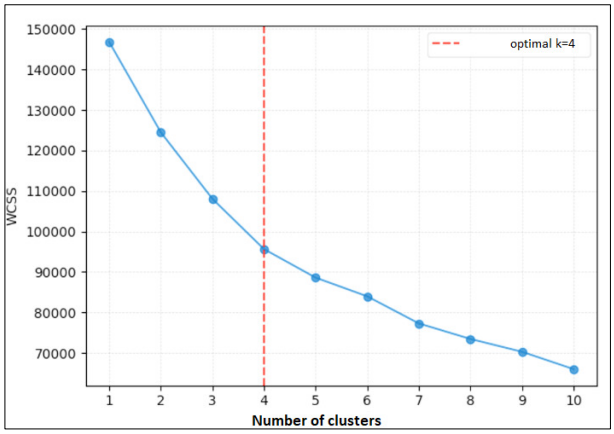

Figure 4: Chart illustrating the Elbow Method.

The effectiveness of GMM depends primarily on two decisions: the number of components in the mixture n_components and the structure of the covariance matrix covariance_type. The experiment used a two-stage selection procedure, combining information criteria with observation of cluster stability.

For this purpose, the Calinski-Harabasz Index metric was calculated for each pair of parameters (Fig. 5), and then the highest value was selected.

Of all pairs, the highest score was obtained with a luster count of 2 and a full covariance matrix type.

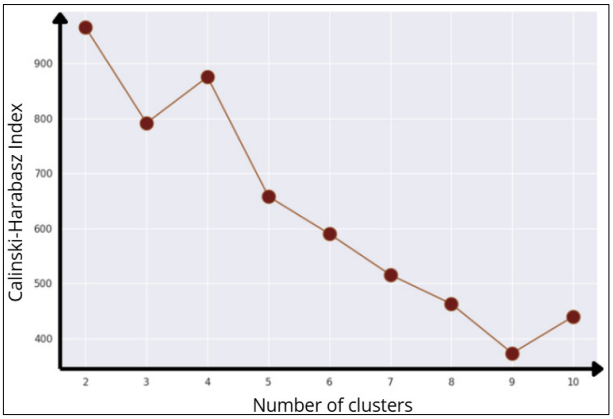

Figure 5: Calinski-Harabasz Index chart.

DBSCAN requires two hyperparameters to be specified: the proximity radius *eps* and the minimum number of points min_samples needed for an observation to be considered a core point. Since there is no need to specify the number of clusters, these parameters determine both the number of clusters and the amount of data classified as noise.

The min_samples selection is calculated using the natural logarithm of the number of records in the set (formula 4).

$$\min\_samples = \ln x, \quad (4)$$

where x is equal records number.

The value for the *eps* parameter is usually selected using the kNN (k-Nearest Neighbours) algorithm and its visualization on a graph (Fig. 6). As in the previous elbow method, look for a point of sudden change in value. In this case, a sudden increase in value could be observed around *eps* = 4.

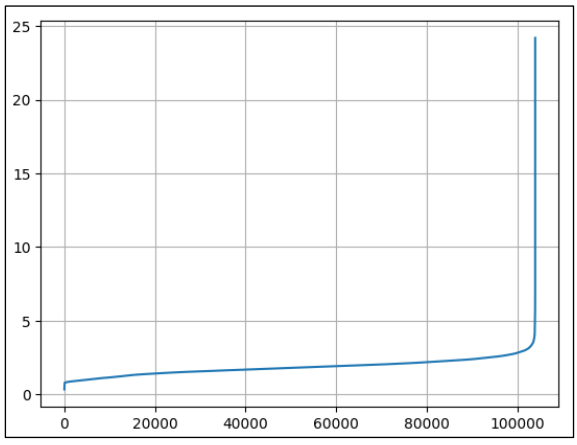

Figure 6: k-Nearest Neighbours chart.

Unlike methods requiring a predetermined number of clusters, Mean Shift primarily requires an accurate estimate of the kernel bandwidth, which determines the resolution of the detected structures. To calculate this value, a ready-made tool available in the used libraries– estimate_bandwidth – is used. It returned a value of 4. Unfortunately, for the bandwidth parameter = 4, the algorithm separated 80 data groups, so the Silhouette and Calinski-Harabasz metrics were analysed to determine the best parameter value.

Tabel 1: Summary of metric results for different bandwidth values

Bandwidth	Clusters count	Silhouette Score	CHI
8,32	13	0,63	139,27
9,44	9	0,67	139,27
10,55	8	0,73	189,89
11,66	5	0,75	264,29
12,77	2	0,78	85,29
13,89	2	0,78	85,29

The most advantageous range is between 10 and 12. At values above 11, the algorithm creates 8 clusters, achieving high results for both metrics, which indicates a compact and well-distributed structure, which is why this value was used.

BIRCH is an algorithm designed for very large data sets; its main hyperparameters are threshold and n_clusters. They determine both the shape of the CF tree and the final number of clusters. The n_clusters parameter was calculated using an elbow chart, as in the case of the K-Means algorithm, indicating a value of 2.

3.5. Discussion

In this chapter, we move on to analysing the clustering results obtained.

The Customer Segmentation set was clustered using five techniques and then evaluated using three commonly used internal measures: Silhouette Score, Calinski-Harabasz Index, and Davies-Bouldin Index. Each of these measures emphasizes a different aspect of the quality of the division: the compactness and separability of clusters, their homogeneity, and the clarity of the boundaries between clusters. By comparing the values of the metrics with the size and shape of the groups obtained, we can identify which algorithms best reflect the structure of the data.

The results of the groupings were first collected and presented using graphs visualizing the distribution of observations in the first two principal components of PCA, which together explain 72% of the variance in the data set.

Figure 7 shows a graph of the K-means clustering method. The set has been divided into four clearly visible groups, with marked central points.

Figure 8 shows the division using the GMM algorithm. Two asymmetrical clusters with overlapping data points are visible.

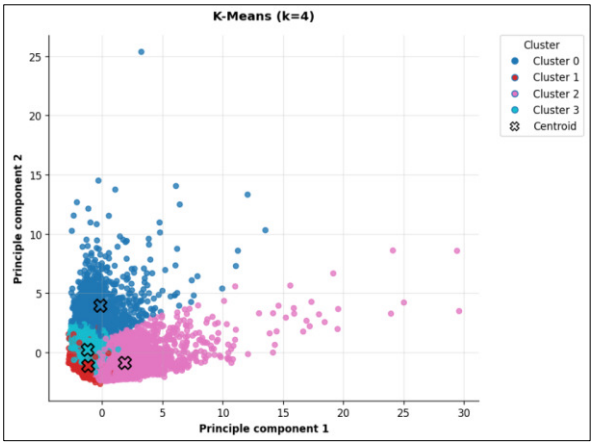

Figure 7: Visualization of clustering using the K-means method.

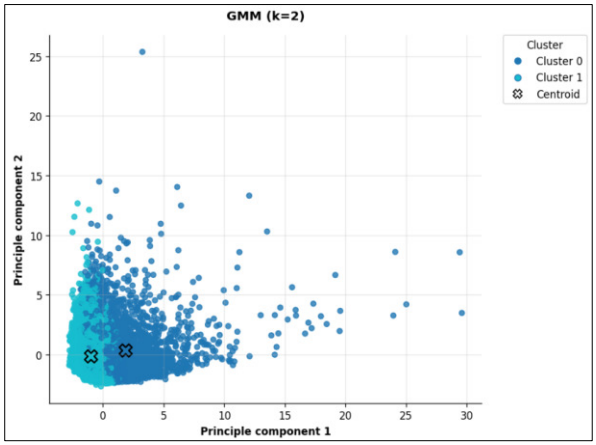

Figure 8: Visualization of clustering using the GMM method.

The DBSCAN method grouped the set into two clusters (Fig. 9), with one of them having a significant advantage in the number of assigned data points. The other was assigned less frequently distributed points and outliers.

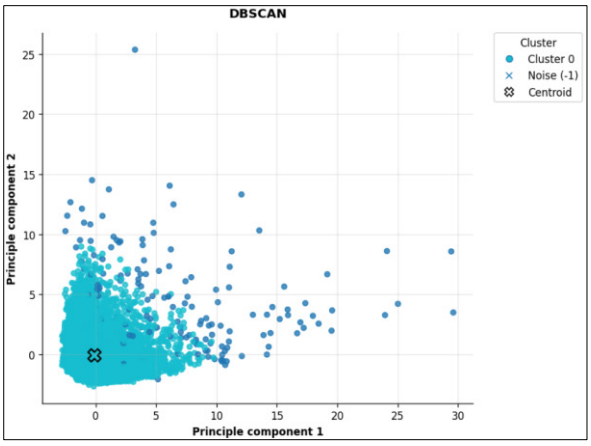

Figure 9: Visualization of clustering using the DBSCAN method.

The Mean-Shift method (Fig. 10) assigned virtually all data points to a single group, creating a very compact data cluster. This separated the outliers into the remaining seven clusters. The remaining clusters can be considered atypical situations, which are very rare.

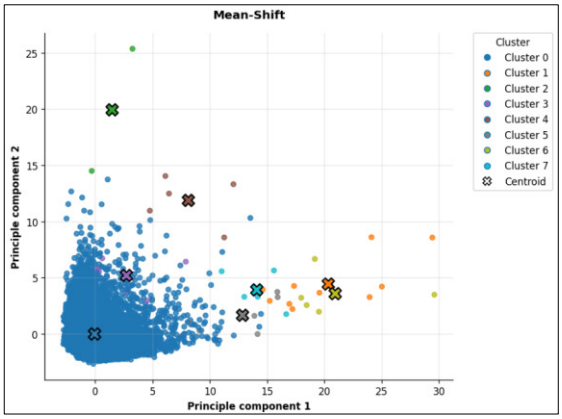

Figure 10: Visualization of grouping using the Mean-Shift method.

The last BIRCH method clearly divided the set of points into two groups (Fig. 11). The first one extends vertically upwards along the vertical axis, and the second one extends widely along the horizontal axis, collecting points with lower density.

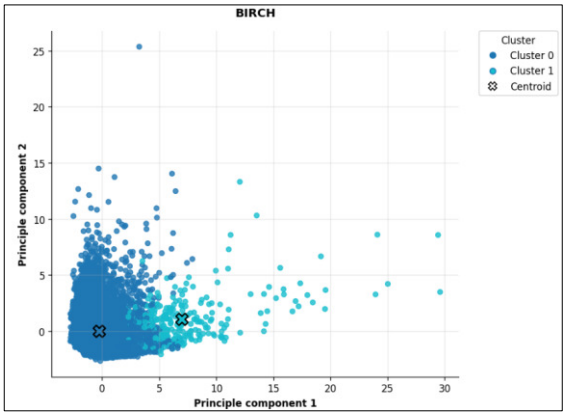

Figure 11: Visualization of clustering using the BIRCH method.

For the final analysis of the clustering results, the metrics were compared. The previously defined measures were collected during the research phase.

The first one was the Silhouette Score measure. It assesses the quality of data division into clusters by comparing the consistency of data points within their own cluster with the distances to the nearest neighbouring group.

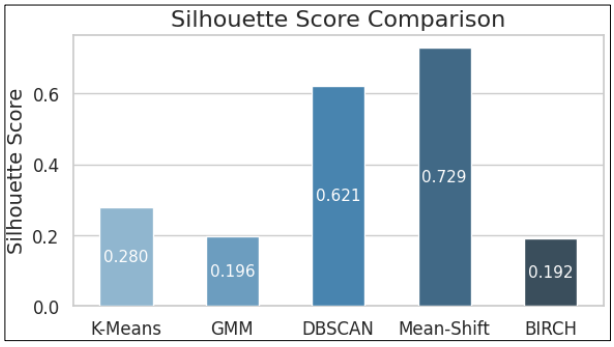

Figure 12: Silhouette Score results.

The second measure used was the Davies-Boulding Index (Fig. 13). It measures the average "similarity" of each cluster to the most similar neighbouring cluster, combining its compact structure and separation from other groups into a single value. Lower results indicate better division.

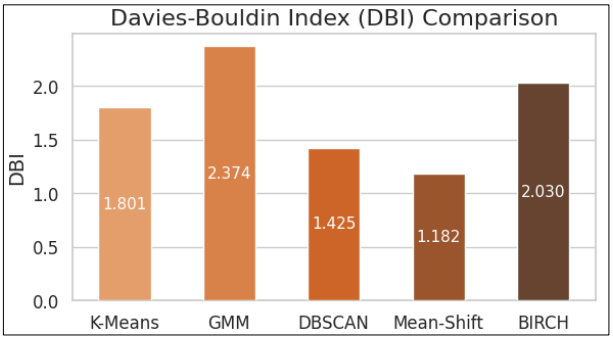

Figure 13: Davies-Bouldin Index results.

The Calinski–Harabasz Index (Fig. 14) compares inter-cluster variance with intra-cluster variance, thereby assessing how well clusters are separated from each other and at the same time how compact they are. The higher the value of the index, the better the clustering structure, because high dispersion between groups and low dispersion within them indicates a clear division.

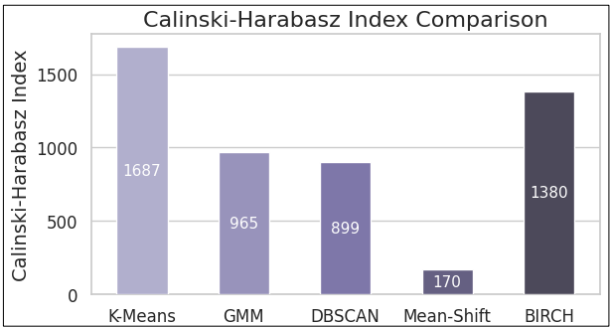

Figure 14: Calinski-Harabasz Index results.

A comparison of three classic quality measures shows that none of the five algorithms dominates in all comparisons – each method reveals strengths and weaknesses depending on the criterion adopted. Mean Shift and DBSCAN perform best in terms of Silhouette and the lowest Davies-Bouldin, which means compact and well-separated clusters with irregular shapes. K-means and, secondarily, BIRCH achieve the highest Calinski-Harabasz index, so they better maximize the distances between centroids with low intra-group variance, but their clusters overlap more. So there is no universal winner: DBSCAN/Mean Shift are better when natural separation of segments is important, while K-means or BIRCH are better when simple, centroid representation and computational scalability are the priority.

Another method for comparing the results of the groupings was a box plot (Fig. 15) showing the stability of the methods using the external measure Normalized Mutual Information (NMI).

The NMI measure is mainly used to check the similarity of two divisions. This requires labels (actual values), which we do not have when grouping. However, there is a method that involves running the clustering on the entire set, and we define its results as the base

partition. Then, through a specified number of iterations, we randomly select a certain part of the data set and perform the clustering again. For each iteration, the results are compared and the NMI is calculated. In this case, 30 iterations were selected for each clustering method on 80% of the data set.

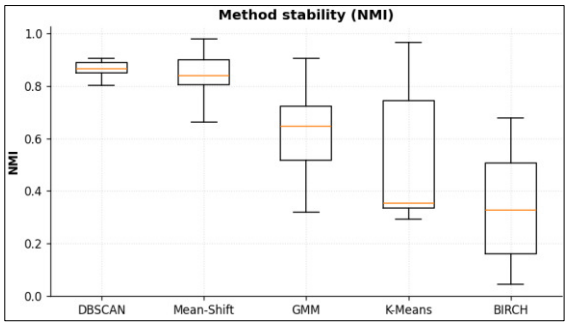

Figure 15: Box plot of method stability (NMI).

The box plot shows the distribution of NMI values for the clustering methods. The lower the "box" the more stable the method. The DBSCAN method proved to be the most repeatable. A high median (orange line) and a small spread of values mean that the clustering results for random data groups overlap significantly. However, it should be noted that the DBSCAN method was characterized by high data noise, which may affect the result in the chart. The Mean-Shift method is also highly stable. The result for GMM can be considered moderate, but it can be seen that there is a much greater difference between the highest and lowest values compared to the previous methods. The K-Means algorithm proved to be highly sensitive to changes in the data set. The large spread of values and low median value show how smoothly the cluster boundaries changed with reduced, randomly selected sets. The BIRCH method is characterized by the lowest and most dispersed stability. Due to the way it works, a small change in the data set changes the tree structure on which the algorithm is based.

4. Conclusions

The study compared five clustering algorithms on the "Customer Segmentation" dataset after uniform data preparation. The results and visualizations obtained were consistent. Density-based methods – Mean Shift and DBSCAN – produced the most compact and well-separated clusters (highest Silhouette, lowest DBI), while K-means and BIRCH achieved the best Calinski-Harabasz variance score and the shortest running time. The final choice of algorithm therefore depends on the priority. Research shows that when selecting the appropriate algorithm, an optimal compromise between cost and quality should be sought. Simpler and less expensive methods can sometimes be more effective than more advanced methods when the parameters are selected appropriately. PCA visualizations confirmed the existence of one large segment of typical customers and several niche groups with different risk profiles and business values. The research conducted shows that the final grouping result depends mainly on the algorithm used and the appropriately selected input parameters.

References

[1] M.W. Berry, A. Mohamed, B.W. Yap, Supervised and Unsupervised Learning for Data Science, Springer, Cham, 2019.

[2] C.K. Reddy, C.C. Aggarwal, Data Clustering: Algorithms and Applications, CRC Press, Boca Raton, 2014.

[3] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1: Statistics, University of California Press, Berkeley (1967) 281-297.

[4] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, Wiley, New York, 1990.

[5] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, Proceedings of the 2nd International Conference on Knowledge Discovery & Data Mining (1996) 226-231, https://doi.org/10.5555/3001460.3001507.

[6] M. Suyal, S. Sharma, A review on analysis of K-means clustering machine learning algorithm based on unsupervised learning, Journal of Artificial Intelligence and Systems 6 (2024) 85-95, https://doi.org/10.33969/AIS.2024060106.

[7] R.R. Muhima, M. Kurniawan, O.T. Pambudi, A LOF K-means clustering on hotspot data, International Journal of Artificial Intelligence & Robotics 2(1) (2020) 29-33, https://doi.org/10.25139/ijair.v2i1.2634.

[8] M. Ahmed, R. Seraj, S.M.S. Islam, The K-means algorithm: A comprehensive survey and performance evaluation, Electronics 9(8) (2020) 1295, https://doi.org/10.3390/electronics9081295.

[9] C. Larasvaty, S. Khomsah, R. Nisa, Cluster analysis of COVID-19 in Indonesia using K-means method, Journal of DINDA: Data Science, Information Technology, and Data Analytics 3(1) (2023) 36-39, https://doi.org/10.20895/dinda.v3i1.822.

[10] X. Huang, Z. Song, Clustering analysis on e-commerce transaction based on K-means clustering, Journal of Networks 9(2) (2014) 443-450, https://doi.org/10.4304/jnw.9.2.443-450.

[11] W.L. Zhang, G.X. Li, W. Gao, The research of speech emotion recognition based on Gaussian mixture model, Applied Mechanics and Materials 668-669 (2014) 1126-1129, https://doi.org/10.4028/www.scientific.net/AMM.668-669.1126.

[12] A. Khoshkbarforoushha, R. Ranjan, P. Strazdins, Resource distribution estimation for data-intensive workloads: Give me my share & no one gets hurt!, In Advances in Service-Oriented and Cloud Computing (2016) 228-237.

[13] M. Hahsler, M. Piekenbrock, D. Doran, dbscan: Fast density-based clustering with R, Journal of Statistical Software 91(1) (2019) 1-30, https://doi.org/10.18637/jss.v091.i01.

[14] D. Comaniciu, P. Meer, Mean shift: A robust approach toward feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5) (2002) 603-619, https://doi.org/10.1109/34.1000236.

[15] K.G. Derpanis, Mean Shift Clustering York University, 2005, https://www.cse.yorku.ca/~kosta/CompVis_Notes/mean_shift.pdf [16.09.2025].

[16] Y. Li, H. Jiang, J. Lu, X. Li, Z. Sun, M. Li, MR-BIRCH: A scalable MapReduce-based BIRCH clustering algorithm, Journal of Intelligent & Fuzzy Systems 40(3) (2021) 5295-5305, https://doi.org/10.3233/JIFS-202079.

[17] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, 2nd ed., Springer, New York, 2009.