Comparison of K-Means and DBSCAN Algorithms for Customer Segmentation in E-commerce

Adi Suryaputra Paramita[1,*], Taqwa Hariguna[2]

[1]School of Information Technology, Universitas Ciputra Surabaya, Indonesia

[2]Magister of Computer Science, Universitas Amikom Purwokerto, Purwokerto, Indonesia

ABSTRACT

Customer segmentation is crucial for e-commerce businesses to effectively target and engage specific customer groups. This study compares the effectiveness of two popular clustering algorithms, K-Means and DBSCAN, in segmenting e-commerce customers. The primary objective is to evaluate and contrast these algorithms to determine which provides more meaningful and actionable customer segments. The methodology involves analyzing a comprehensive e-commerce customer dataset, which includes various features such as customer ID, gender, age, city, membership type, total spend, items purchased, average rating, discount applied, days since last purchase, and satisfaction level. Initial data preprocessing steps include handling missing values, encoding categorical variables, and normalizing numerical features. Both K-Means and DBSCAN algorithms are implemented, and their performance is evaluated using metrics such as silhouette score, Davies-Bouldin index, and Calinski-Harabasz score. The results indicate that K-Means achieved a silhouette score of 0.546, a Davies-Bouldin index of 0.655, and a Calinski-Harabasz score of 552.9. In contrast, DBSCAN achieved a higher silhouette score of 0.680, a Davies-Bouldin index of 1.344, and a Calinski-Harabasz score of 1123.9. These findings suggest that while DBSCAN performs better in terms of silhouette score, indicating more distinctly separated clusters, its higher Davies-Bouldin index reflects fewer compact clusters. The discussion highlights that K-Means is suitable for applications requiring clear and well-defined segments of customers, as it produces balanced cluster sizes. DBSCAN, with its strength in identifying clusters of varying densities and handling noise, is more effective in detecting niche markets and unique customer behaviors. This study's findings have significant practical implications for e-commerce businesses looking to enhance their customer segmentation strategies. In conclusion, both K-Means and DBSCAN demonstrate their respective strengths and weaknesses in clustering the e-commerce customer dataset. The choice of algorithm should be based on the specific requirements of the segmentation task. Future research could explore hybrid methods combining the strengths of both algorithms and incorporate additional data sources for a more comprehensive analysis.

Keywords K-Means, DBSCAN, Customer Segmentation, E-Commerce, Clustering Performance Evaluation

Submitted 10 January 2024
Accepted 20 April 2024
Published 1 June 2024

Corresponding author
Adi Suryaputra Paramita,
adi.suryaputra@ciputra.ac.id

Additional Information and Declarations can be found on page 60

DOI: 10.47738/jdmdc.v1i1.3

Copyright
2024 Paramita and Hariguna

Distributed under
Creative Commons CC-BY 4.0

INTRODUCTION

The e-commerce industry caters to a diverse customer base with varying preferences, behaviors, and purchasing patterns. This diversity makes customer segmentation essential for targeting specific customer groups effectively. Customer segmentation is a fundamental aspect of marketing strategy, allowing businesses to divide their customer base into distinct groups with similar characteristics or buying preferences [1]. By segmenting customers, companies can effectively tailor their marketing mix to target specific customer groups [2]. This segmentation process involves categorizing customers based

How to cite this article: A. S. Paramita and T. Hariguna, "Comparison of K-Means and DBSCAN Algorithms for Customer Segmentation in E-commerce," *J. Digit. Mark. Digit. Curr.*, vol. 1, no. 1, pp. 43-62, 2024.

on attributes to create homogenous groups with common characteristics [3].

One of the primary goals of customer segmentation is to identify customer groups with similar attributes and behaviors to facilitate the design of better-tailored marketing strategies [4]. This approach enables companies to understand customer preferences, respond to their demands, and increase revenue by attracting new customers with relevant marketing initiatives [5]. By segmenting customers effectively, businesses can enhance customer satisfaction, increase customer retention, and improve performance [6].

Customer segmentation is essential for targeting specific customer groups and plays a significant role in reducing product returns and meeting customer needs in industries like e-commerce [7]. Through digital transformation technologies and comprehensive customer segmentation techniques, businesses can better understand customer preferences and provide products and services that align with their needs [7]. Moreover, using advanced data mining techniques, customer segmentation can help analyze customer behavior and categorize customers into meaningful groups based on their features [8]. Furthermore, customer segmentation aids in understanding customer preferences, which guides future actions and helps in developing more effective marketing strategies [9].

Utilizing big data in customer segmentation within the retail sector enables businesses to separate consumers based on their past purchase behavior, allowing for more targeted marketing efforts. By analyzing customer data and segmenting customers effectively, companies can identify patterns and trends that help predict future buying behaviors and preferences, leading to more effective sales strategies. This data-driven approach to customer segmentation enhances customer engagement by delivering personalized experiences that cater to the specific needs of different customer segments, ultimately driving sales growth [10].

Clustering in data science is a fundamental technique crucial in uncovering patterns and groupings within datasets, enabling researchers and analysts to gain valuable insights from complex data structures [11]. The primary objective of clustering is to create high-quality clusters based on similarity measures, allowing for discovering hidden patterns and simplifying data analysis processes. By organizing data into meaningful clusters, clustering techniques facilitate identifying relationships and structures within datasets, essential for various applications in data science and analytics [12].

Different clustering techniques can be categorized into several types, such as hierarchical, partitioning, and density-based methods. Hierarchical clustering builds a tree of clusters by either merging or splitting existing clusters. Partitioning methods, like K-Means, divide the data into a predefined number of clusters. Density-based methods, such as DBSCAN, identify clusters based on the density of data points in the feature space.

Moreover, clustering techniques are instrumental in data mining applications, enabling the discovery of patterns, trends, and relationships within large datasets. By applying clustering algorithms to big data processing tasks, businesses can extract valuable insights, identify trends, and make data-driven decisions to enhance their operations. Clustering also plays a vital role in multi-

relational data mining and spatial-temporal database applications, highlighting its significance in extracting meaningful information from complex datasets [13]. K-Means and DBSCAN are two popular clustering algorithms widely used for customer segmentation in e-commerce. K-Means is a widely used clustering algorithm in data science that is instrumental in grouping data into a predetermined number of clusters, facilitating the identification of patterns and groupings within datasets [14]. This algorithm finds applications in various fields, such as customer segmentation, where it aids in categorizing customers into distinct groups based on similarities in their attributes and behaviors. Businesses can effectively categorize customers into segments using the K-Means algorithm, enabling targeted marketing strategies and personalized customer engagement initiatives [15].

K-Means is a partitioning algorithm that divides the data into a predefined number of clusters (k). The algorithm iteratively assigns each data point to the nearest cluster center and updates the cluster centers based on the mean of the designated points. K-Means is known for its simplicity and efficiency, making it suitable for large datasets. It is commonly used in applications where the number of clusters is known beforehand, and the data points are roughly spherical.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based algorithm that identifies clusters based on the density of data points. This algorithm is widely used in various fields, including customer segmentation, market analysis, and anomaly detection, due to its ability to handle noisy data and identify clusters of arbitrary shapes and sizes. DBSCAN is particularly suitable for spatial data analysis as it can effectively distinguish noise points, discover clusters of any shape, and naturally support spatial databases.

In customer segmentation, DBSCAN is utilized to categorize customers into distinct groups based on their attributes and behaviors, enabling businesses to tailor marketing strategies and enhance customer engagement [16]. By leveraging DBSCAN's capabilities, companies can identify meaningful customer segments, personalize marketing campaigns, and improve customer satisfaction and loyalty.

K-Means and DBSCAN are suitable for customer segmentation tasks due to their ability to group customers based on similarities in their behavior and preferences. K-Means is effective when dealing with well-defined clusters and large datasets, while DBSCAN excels in identifying clusters of varying shapes and densities, handling noise and outliers effectively. Each algorithm has unique features and strengths. K-Means is computationally efficient and easy to implement, making it a popular choice for many segmentation tasks. DBSCAN, on the other hand, is more flexible in handling clusters of different shapes and sizes and is robust against noise, making it ideal for more complex and varied datasets.

The primary aim of this study is to compare the effectiveness of the K-Means and DBSCAN algorithms in segmenting e-commerce customers. By evaluating and contrasting these two clustering methods, the study seeks to determine which algorithm provides more meaningful and actionable customer segments.

This study's research question is: "Which clustering algorithm, K-Means or DBSCAN, is more effective in segmenting e-commerce customers based on their behavior and preferences?"

This study is highly relevant for e-commerce businesses looking to enhance their customer segmentation strategies. Effective segmentation allows businesses to tailor their marketing efforts, improve customer satisfaction and retention. By understanding the strengths and weaknesses of each algorithm, e-commerce platforms can make informed decisions on which clustering method to implement for optimal customer segmentation.

Literature Review

Customer Segmentation in E-commerce

Customer segmentation has evolved significantly over time, driven by data collection and analysis techniques advancements. In the early days of commerce, segmentation was primarily based on broad demographic factors such as age, gender, and income level. These traditional methods relied heavily on basic statistical analysis and needed to be more comprehensive in capturing the complexities of customer behavior.

With the advent of digital technologies and the rise of customer segmentation has become more sophisticated. Modern approaches leverage vast amounts of data generated by online activities, enabling businesses to segment customers based on a wide array of behavioral, transactional, and psychographic attributes. This shift has been facilitated by advancements in data science and machine learning, which provide the tools to analyze complex datasets and uncover hidden patterns.

Numerous studies have explored the application of customer segmentation in e-commerce, highlighting its importance and effectiveness in enhancing business outcomes. Notable studies in this field have employed various methodologies to segment customers and analyze their behavior. Study by [17] discusses the segmentation of e-commerce customers using an improved K-Medoids clustering algorithm, highlighting the importance of segmenting customers in the e-commerce domain. Customer segmentation is crucial for e-commerce businesses to understand customer behavior, preferences, and purchasing patterns, enabling personalized marketing strategies and enhanced customer engagement. Research by [6] emphasizes the importance of correctly segmenting customers in e-commerce to meet customer needs, expand the customer base, and ultimately save businesses money customer segmentation in e-commerce can lead to improved customer satisfaction, increased customer retention, and enhanced profitability.

These studies collectively underscore the importance of choosing the right segmentation method based on the specific characteristics of the dataset and the segmentation objectives. They also highlight the evolution of customer segmentation techniques, driven by advancements in data science and machine learning. By reviewing these key findings and methodologies, this literature review provides a foundation for the current study, which aims to compare the effectiveness of K-Means and DBSCAN algorithms in segmenting e-commerce customers.

Clustering Algorithms

Clustering is a fundamental technique in data science used to group similar data points based on their characteristics. The primary goal of clustering is to identify natural groupings within a dataset, which can reveal patterns and relationships that are not immediately apparent. There are several clustering methodologies, each with its advantages and applications:

Hierarchical Clustering builds a hierarchy of clusters either by agglomerating individual data points into larger clusters (agglomerative clustering) or by splitting a single large cluster into smaller ones (divisive clustering). The result is often visualized as a dendrogram showing the nested grouping of clusters and their relationships.

Partitioning Clustering, such as K-Means, divide the dataset into a predefined number of clusters. These methods assign each data point to exactly one cluster, optimizing the cluster centers iteratively.

Density-Based Clustering, such as DBSCAN, define clusters as areas of high data point density separated by areas of low density. These methods are particularly effective in identifying clusters of arbitrary shapes and handling noise.

The selection of an appropriate clustering technique depends on various criteria, including the nature of the data, the desired outcome, and the specific characteristics of the clusters. Factors to consider include clusters' shape and size, noise and outliers' presence, and the computational efficiency required.

K-Means is a widely used partitioning clustering algorithm that aims to partition a dataset into K distinct, non-overlapping clusters. The algorithm operates on the principle of minimizing the within-cluster variance. The K-Means algorithm begins by initializing K cluster centers randomly. Each data point is assigned to the nearest cluster center based on the Euclidean distance. The algorithm iteratively updates the cluster centers by calculating the mean of the data points assigned to each cluster. This process continues until the cluster centers converge, meaning they no longer change significantly with further iterations. Steps Involved in the K-Means Algorithm:

1) Choose the number of clusters, K
2) Initialize K cluster centers randomly
3) Assign each data point to the nearest cluster center
4) Update the cluster centers by calculating the mean of the data points in each cluster
5) Repeat steps 3 and 4 until convergence

Implementing the K-Means algorithm can present various challenges that must be addressed to ensure its effective application in data analysis. One common challenge is the selection of the initial centroids, which can significantly impact the clustering results [18]. The choice of the number of clusters (K) is another critical challenge in implementing K-Means, as selecting an inappropriate value for K can lead to suboptimal clustering outcomes. Additionally, detecting outliers and handling noisy data pose challenges in K-Means clustering, as outliers can

influence the centroid positions and affect the clustering process.

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. Unlike K-Means, DBSCAN does not require the number of clusters to be specified in advance and can find clusters of arbitrary shapes. DBSCAN defines clusters as dense regions of data points separated by sparser regions. It relies on two parameters: epsilon (ε), which defines the radius for neighborhood search, and MinPts, the minimum number of points required to form a dense region (core point). Points that do not meet these criteria are considered noise or outliers. Steps Involved in the DBSCAN Algorithm is:

1) Select an arbitrary point as the starting point
2) Retrieve all points within the ε-neighborhood of the starting point
3) If the number of points in the neighborhood is greater than or equal to MinPts, create a new cluster. Otherwise, label the point as noise
4) Expand the cluster by recursively including all density-reachable points
5) Repeat steps 1 to 4 until all points are processed

Common challenges and solutions in implementing DBSCAN include the selection of appropriate parameters, handling varying densities, and ensuring computational efficiency. Choosing suitable values for the parameters ε (epsilon) and MinPts (minimum points) is crucial for DBSCAN's performance. These parameters can be determined through domain knowledge, trial and error, or using techniques such as k-distance graphs. Handling datasets with clusters of varying densities is another challenge for DBSCAN. Hierarchical DBSCAN (HDBSCAN), an extension of DBSCAN, addresses this limitation by allowing clusters of varying densities. Finally, due to its neighborhood search, DBSCAN's performance can degrade with large datasets. Optimizations such as using spatial indexing structures, like k-d trees, can significantly improve efficiency. By understanding these principles, steps, and challenges, this study aims to leverage K-Means and DBSCAN algorithms for effective customer segmentation in e-commerce, comparing their performance and suitability for different segmentation tasks.

Comparative Studies on Clustering Algorithms

Several comparative studies have evaluated the performance and applicability of K-Means and DBSCAN algorithms across various domains. Research by [19] compared the DBSCAN algorithm with a proven segmentation algorithm utilizing K-Means clustering for identifying swallows from swallowing accelerometry signals, demonstrating that DBSCAN exhibited higher sensitivity and accurately segmented more swallows. A study by [20] compared the results of K-Means, GMM, Hierarchical, and DBSCAN clustering for detecting anomalies in wastewater, highlighting minimal intra-cluster variability achieved using K-Means.

Comparative research on clustering algorithms like K-Means and DBSCAN spans diverse fields such as market segmentation, image processing, anomaly detection, and geospatial analysis. In market segmentation, studies have explored how these algorithms can segment customers based on purchasing

behavior and demographic information. In image processing, researchers have compared the algorithms' ability to cluster pixels or features within images. Anomaly detection studies have evaluated the effectiveness of each algorithm in identifying outliers in datasets, while geospatial analysis has focused on clustering geographic locations based on density.

Efficiency and scalability are key considerations in these studies. K-Means has consistently been more computationally efficient and scalable than DBSCAN, making it suitable for large datasets. However, K-Means requires the number of clusters to be specified in advance and assumes spherical cluster shapes, which can be limiting. On the other hand, DBSCAN excels in identifying clusters of arbitrary shapes and is robust against noise and outliers, making it particularly useful for datasets with complex structures and varying densities. However, DBSCAN's performance can be sensitive to the choice of parameters (ε and MinPts) and may struggle with clusters of varying densities.

In specific applications, such as market segmentation, studies have shown that K-Means can effectively segment customers into homogeneous groups when clusters are well-defined and separated. Conversely, DBSCAN is more effective in identifying natural clusters in datasets with noise and irregular shapes, making it ideal for more exploratory analysis.

Insights gained from past research inform the current study by highlighting the contexts in which K-Means and DBSCAN perform well. By understanding the advantages and limitations of each algorithm, this study builds on previous findings to apply these algorithms to e-commerce customer segmentation effectively. Despite extensive comparative studies, gaps still need to be found in understanding the performance of K-Means and DBSCAN in specific contexts, such as e-commerce customer segmentation. Many studies have focused on general applications or specific domains, but few have provided a detailed comparison in the context of e-commerce. Additionally, more empirical studies are needed to evaluate the practical implications of using these algorithms for real-world business applications.

By addressing these gaps, the current study aims to comprehensively compare K-Means and DBSCAN for e-commerce customer segmentation, offering actionable insights for businesses looking to enhance their customer segmentation strategies. This study will contribute to the literature by evaluating the algorithms' performance on a rich e-commerce dataset, considering both the resulting segments' efficiency and quality.

Method

The methodology of this study is visually represented in a flowchart, covering each major step from data collection and preprocessing to model implementation and evaluation, as shown in figure 1.

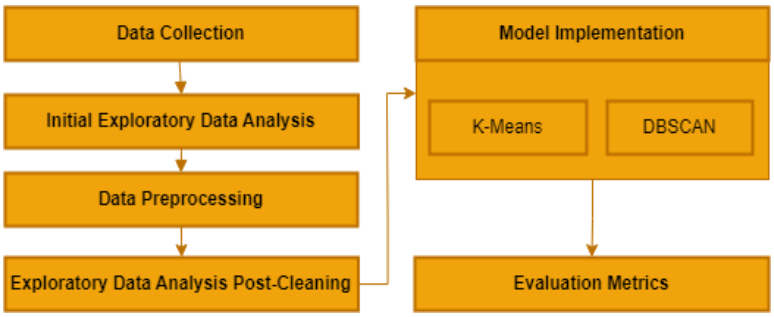

Figure 1 Research Methodology Flowchart

Data Collection

The dataset utilized in this study is derived from an e-commerce platform and captures various aspects of customer behavior. Data collection methods include tracking user interactions, purchase history, and feedback submissions. The data provider preprocessed the dataset by anonymizing personal information and ensuring data consistency. The dataset consists of 350 entries, each representing a unique customer, with 11 features capturing different dimensions of customer behavior. The dataset is evenly distributed across three customer segments based on membership type: Gold (117 entries), Silver (117 entries), and Bronze (116 entries).

The data variables included in the dataset are as follows. The "Customer ID" is a unique identifier for each customer. "Gender" specifies the customer's gender, either Male or Female. "Age" represents the customer's age. "City" indicates the city of residence for each customer. "Membership Type" identifies the type of membership held by the customer, categorized as Gold, Silver, or Bronze. "Total Spend" represents the total monetary expenditure by the customer on the e-commerce platform. "Items Purchased" quantifies the total number of items the customer purchases. "Average Rating" is the average rating given by the customer, on a scale of 0 to 5. "Discount Applied" indicates whether a discount was applied to the purchase, represented as True or False. "Days Since Last Purchase" reflects the number of days since the customer's most recent purchase. Finally, "Satisfaction Level" captures the overall satisfaction level of the customer, categorized as Satisfied, Neutral, or Unsatisfied.

Initial Exploratory Data Analysis (EDA)

Initial data exploration involved examining the raw data to identify missing values, duplicates, outliers, and other potential issues. The dataset was found to have missing values in the "Satisfaction Level" column, with two entries missing. There were no duplicate records in the dataset, ensuring the uniqueness of each entry.

Descriptive statistics were generated to provide an overview of the dataset. The "Customer ID" column ranged from 101 to 450, with a mean of 275.5 and a standard deviation of 101.18. Gender was equally distributed between Male and Female, each with 175 entries. The average age of customers was 33.6 years, with a standard deviation of 4.87, and ages ranged from 26 to 43 years as shown in figure 2.

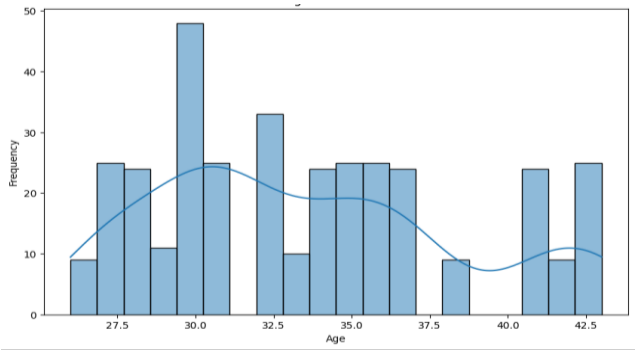

Figure 2 **Distribution of Age**

The dataset included customers from six different cities, with New York having the highest representation at 59 entries. Membership types were evenly distributed among gold, silver, and bronze categories. Total customer spending ranged from $410.80 to $1520.10, with an average spend of $845.38 and a standard deviation of $362.06 as shown in figure 3.

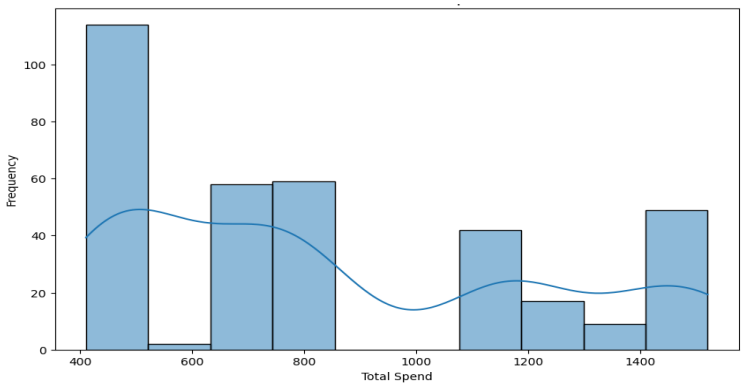

Figure 3 **Distribution of Total Spend**

The number of items purchased ranged from 7 to 21, with a mean of 12.6 and a standard deviation of 4.16. Average ratings given by customers ranged from 3.0 to 4.9, with an average rating of 4.02 and a standard deviation of 0.58. Discounts were applied in half of the transactions. The number of days since the last purchase ranged from 9 to 63 days, with an average of 26.59 days and a standard deviation of 13.44. The majority of customers were satisfied, followed by neutral and unsatisfied customers.

To further explore the data, various visualizations were created. Scatter plots were used to explore the relationships between total spending and items purchased, with colors indicating different membership types as shown in figure 4.

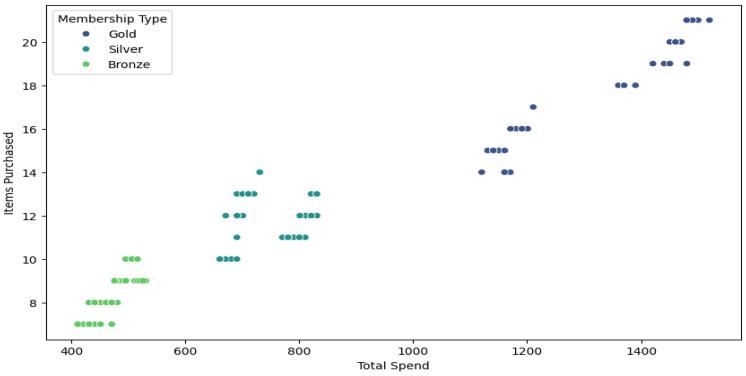

Figure 4 Total Spend vs Items Purchased

These visualizations helped to understand the data distribution and relationships between variables, providing valuable insights for the subsequent analysis.

Data Preprocessing

Data cleaning involved several steps to ensure the dataset was ready for analysis. First, missing values were handled by filling numeric columns such as "Age," "Total Spend," "Items Purchased," "Average Rating," and "Days Since Last Purchase" with their respective median values. Categorical columns, including "Gender," "City," "Membership Type," and "Satisfaction Level," were filled with the mode. This approach ensured that no critical information was lost due to missing values. Additionally, the dataset was checked for duplicates, and none were found, confirming the uniqueness of each entry.

Encoding categorical variables was the next crucial step. Variables such as "Gender," "City," "Membership Type," and "Satisfaction Level" were converted into a numerical format using label encoding. This transformation was necessary for the algorithms to process the data effectively. For instance, "Gender" was encoded as 0 and 1, representing male and female, respectively. Similarly, cities and membership types were assigned numerical values, enabling the model to interpret these features correctly.

Scaling and normalization of numerical features were carried out using the StandardScaler. Features like "Age," "Total Spend," "Items Purchased," "Average Rating," and "Days Since Last Purchase" were normalized to ensure consistency and improve model performance. This process involved transforming the data to have a mean of zero and a standard deviation of one, which helps reduce biases due to different variables scales. The cleaned and processed dataset was then prepared for further analysis, ensuring it was free of inconsistencies and ready for accurate clustering and evaluation.

The following is a snapshot of the cleaned and processed dataset, illustrating the transformations applied. The dataset now contains numerically encoded categorical variables and normalized numerical features, making it suitable for implementing clustering algorithms. For example, a portion of the dataset shows Customer ID, Gender, Age, City, Membership Type, Total Spend, Items Purchased, Average Rating, Discount Applied, Days Since Last Purchase, and Satisfaction Level, all in their processed form. This thorough preprocessing

ensures that the dataset is robust and ready for the subsequent stages of analysis and modeling.

Post-Cleaning EDA

Post-cleaning data exploration was conducted to confirm the effectiveness of the cleaning steps. After cleaning, the dataset no longer contained any missing values across all columns, including "Satisfaction Level," which had previously missing entries. Additionally, the dataset was free of duplicates, ensuring the uniqueness of each record.

Updated descriptive statistics were generated to reflect the cleaned data. The "Customer ID" column maintained its range from 101 to 450, with a mean of 275.5 and a standard deviation of 101.18. The "Gender" column was evenly distributed, with a mean of 0.5 and a standard deviation of 0.5, indicating a balanced representation of male and female customers. The "Age" column, normalized during preprocessing, had a mean close to zero, reflecting successful scaling, with values ranging from -1.56 to 1.93. The "City" and "Membership Type" columns were also normalized, resulting in a mean of approximately 2.5 for "City" and 1.0 for "Membership Type," indicating their balanced distribution.

After normalization, the "Total Spend" column showed a mean close to zero with values ranging from -1.20 to 1.87, and a standard deviation of 1.0, indicating effective scaling. Similarly, "Items Purchased" had a normalized range from -1.35 to 2.02, with a mean near zero. The "Average Rating" column also had a mean close to zero, with values normalized between -1.76 and 1.52, demonstrating successful scaling. The "Discount Applied" column was evenly split between true and false values, as reflected by its categorical distribution.

The "Days Since Last Purchase" column, after norm -1.31 to 2.71, with a mean close to zero, indicates the normalization process's effectiveness. The "Satisfaction Level" column showed a balanced distribution among the categories of satisfied, neutral, and unsatisfied customers, with a mean of 1.03 and a standard deviation of 0.80.

Clustering Algorithms Implementation

This study implemented clustering algorithms using the K-Means and DBSCAN algorithms. The K-Means algorithm operates on the principle of minimizing within-cluster variance, aiming to partition the data into K distinct, non-overlapping clusters. The objective function of K-Means is to minimize the sum of squared distances between each data point and its respective cluster center. This ensures that the data points within each cluster are as close to each other as possible.

The steps involved in the K-Means algorithm begin with choosing the number of clusters, K. This is followed by randomly initializing K cluster centers. Each data point is assigned to the nearest cluster center based on the Euclidean distance. The cluster centers are subsequently updated by calculating the mean of the data points in each cluster. This process of assigning data points and updating cluster centers is repeated until the algorithm converges, meaning the cluster centers no longer change significantly.

Common challenges in implementing K-Means include choosing the optimal number of clusters. Methods such as the Elbow Method and the Silhouette Score are employed to determine the most suitable value for K. Another challenge is the sensitivity to the initial placement of cluster centers. This issue is addressed using techniques like K-Means++, which provides a better initialization strategy. Additionally, Outliers can affect K-Means, distorting the cluster centers. Preprocessing the data to handle outliers is crucial in mitigating this issue.

In this study, the optimal number of clusters was determined to be four, based on the Elbow Method and Silhouette Score, as shown in figure 5 and figure 6.

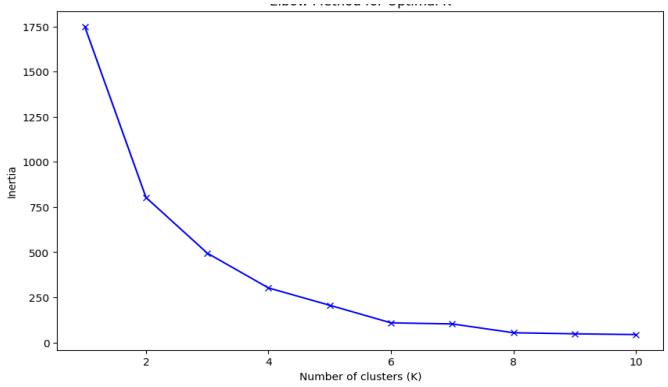

Figure 5 Elbow Method for Optimal K

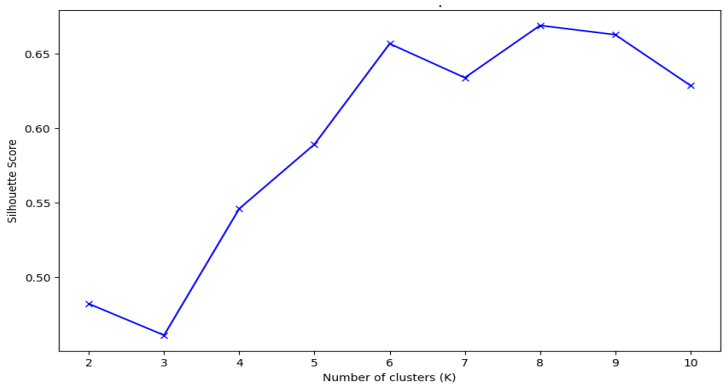

Figure 6 Silhouette Score for Optimal K

The K-Means algorithm resulted in four clusters with the following distribution: Cluster 1 contained 117 data points, Cluster 0 had 116 data points, Cluster 3 had 78 data points, and Cluster 2 had 39 data points, as shown in figure 7.

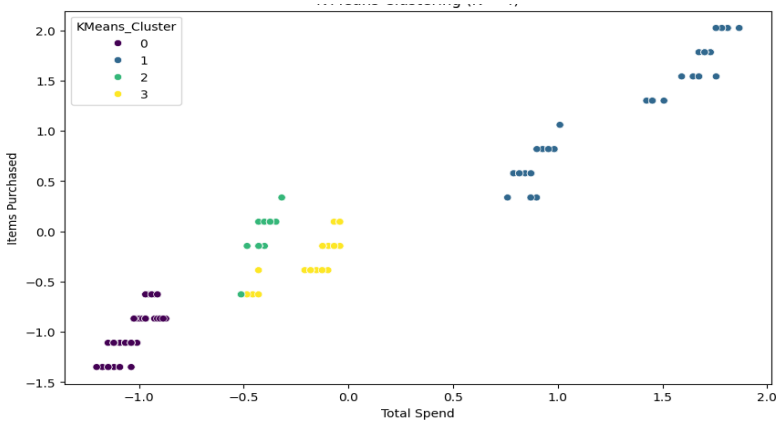

Figure 7 K-Means Clustering

The DBSCAN algorithm, on the other hand, defines clusters based on the density of data points. It identifies clusters as areas of high density separated by low-density areas. The key parameters for DBSCAN are epsilon (ε), which defines the radius for neighborhood search, and MinPts, the minimum number of points required to form a dense region. DBSCAN begins by selecting an arbitrary point and retrieving all points within its ε-neighborhood. A new cluster is formed if the number of points in the neighborhood is greater than or equal to MinPts. This process is expanded by recursively including all density-reachable points until all points are processed.

In this study, the DBSCAN algorithm identified multiple clusters with varying densities. The clusters formed by DBSCAN were as follows: Clusters 0, 1, 4, and 5 each contained 58 data points; Clusters 2, 3, 6, and 7 each contained 24 to 33 data points; Cluster 8 contained 9 data points; and there were 4 data points labeled as noise (Cluster -1) as shown in figure 8.

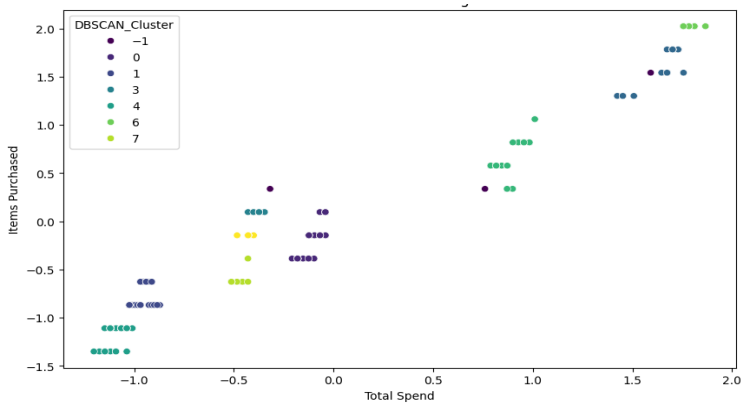

Figure 8 DBSCAN Clustering

Implementing these clustering algorithms provided valuable insights into the dataset structure, highlighting distinct groups of customers based on their behavior and preferences. The results from both K-Means and DBSCAN illustrate the effectiveness of these algorithms in segmenting e-commerce customers, each with its own advantages and challenges.

Evaluation Metrics

To evaluate the performance of the clustering algorithms, several metrics were employed. The silhouette score, Davies-Bouldin index, and Calinski-Harabasz score were used to assess the quality of the clusters formed by the K-Means and DBSCAN algorithms. The silhouette score measures how similar a data point is to its own cluster compared to other clusters, with higher values indicating better-defined clusters. The Davies-Bouldin index quantifies the average similarity ratio of each cluster with its most similar cluster, where lower values suggest better clustering. The Calinski-Harabasz score, also known as the variance ratio criterion, evaluates the ratio of the sum of between-cluster dispersion to within-cluster dispersion, with higher scores indicating more distinct clusters.

In this study, the K-Means algorithm achieved a silhouette score of 0.546, a Davies-Bouldin index of 0.655, and a Calinski-Harabasz score of 552.9. These metrics indicate that while K-Means produced reasonably well-defined clusters, there is room for improvement regarding cluster separation and cohesion. The cluster sizes for K-Means were distributed as follows: Cluster 1 with 117 data points, Cluster 0 with 116 data points, Cluster 3 with 78 data points, and Cluster 2 with 39 data points.

The DBSCAN algorithm, on the other hand, produced a higher silhouette score of 0.680, suggesting that the clusters were more distinctly separated than K-Means. However, the Davies-Bouldin index for DBSCAN was 1.344, higher than that of K-Means, indicating less optimal cluster compactness. The Calinski-Harabasz score for DBSCAN was significantly higher at 1123.9, reflecting well-separated and distinct clusters. The DBSCAN algorithm formed clusters with the following sizes: Clusters 0, 1, 4, and 5 each contained 58 data points; Cluster 2 had 33 data points; Clusters 3, 6, and 7 each had 24 data points; Cluster 8 had 9 data points; and 4 data points were labeled as noise (Cluster -1).

The comparison criteria for K-Means and DBSCAN included both performance and computational efficiency. Performance was primarily evaluated using the abovementioned metrics, which provided insights into the clustering quality. Although the metrics are not detailed, computational efficiency is also a critical factor. K-Means is generally faster and more efficient for large datasets due to its linear time complexity concerning the number of data points. In contrast, DBSCAN, with its density-based approach, can be computationally intensive, particularly for datasets with varying densities and a significant number of outliers.

Overall, the evaluation metrics highlighted the strengths and weaknesses of each algorithm. K-Means demonstrated balanced performance in terms of silhouette score and Davies-Bouldin index but showed limitations in handling outliers and varying densities. DBSCAN excelled in identifying well-separated clusters, as indicated by its high silhouette and Calinski-Harabasz scores, but was less compact as reflected in the higher Davies-Bouldin index. These insights are crucial for selecting the appropriate clustering algorithm based on the specific characteristics and requirements of the e-commerce customer dataset.

Result and Discussion

Result

The clustering outcomes for both K-Means and DBSCAN were analyzed and presented to evaluate their effectiveness in segmenting the e-commerce customer dataset. The results of the K-Means clustering algorithm are shown in table 1. K-Means formed four distinct clusters with the following sizes: Cluster 1 contained 117 data points, Cluster 0 had 116 data points, Cluster 3 had 78 data points, and Cluster 2 had 39 data points.

Table 1 K-Means Cluster Sizes

Cluster	Size
1	117
0	116
3	78
2	39

The DBSCAN algorithm produced a different set of clusters, summarized in table 2. DBSCAN identified multiple clusters of varying sizes, with Clusters 0, 1, 4, and 5 each containing 58 data points. Other clusters included Cluster 2 with 33 data points, Clusters 3, 6, and 7 with 24 data points each, and Cluster 8 with 9 data points. Additionally, 4 data points were labeled as noise and did not belong to any cluster.

Table 2 DBSCAN Cluster Sizes

Cluster	Size
0	58
1	58
4	58
5	58
2	33
3	24
6	24
7	24
8	9
-1 (Noise)	4

The detailed results of the clustering algorithms provide insights into the characteristics of each cluster. For K-Means, Cluster 1, which is the largest, includes customers with high total spending and many items purchased, indicating a segment of high-value customers. Cluster 0, similar in size, represents a group with moderate spending and a balanced number of purchases. Cluster 3 and Cluster 2 represent smaller segments, with Cluster 3 containing customers with lower spending and fewer purchases, while Cluster 2 includes those with even lower engagement.

DBSCAN's detailed results reveal a different segmentation pattern. The large clusters (0, 1, 4, and 5) show dense regions of customers with similar behaviors. Smaller clusters like 2, 3, 6, and 7 indicate DBSCAN's ability to identify and isolate niche segments within the dataset. Cluster 8, being the smallest, may represent outliers or a very specific customer group. The noise points labeled as Cluster -1 indicate data points that do not fit well into any cluster, reflecting DBSCAN's robustness in handling noise.

The comparative analysis of K-Means and DBSCAN was based on the evaluation metrics: silhouette score, Davies-Bouldin index, and Calinski-Harabasz score. K-Means achieved a silhouette score of 0.546, indicating moderately well-defined clusters, and a Davies-Bouldin index of 0.655, suggesting good cluster compactness and separation. Its Calinski-Harabasz score was 552.9, reflecting reasonable between-cluster dispersion.

In contrast, DBSCAN achieved a higher silhouette score of 0.680, indicating better-defined clusters than K-Means. However, its Davies-Bouldin index was higher at 1.344, indicating fewer compact clusters. The Calinski-Harabasz score for DBSCAN was significantly higher at 1123.9, demonstrating excellent cluster separation and dispersion.

While K-Means showed balanced performance across the metrics, DBSCAN excelled in identifying well-separated clusters but had a higher Davies-Bouldin index due to the presence of smaller and more scattered clusters. The choice between K-Means and DBSCAN depends on the specific requirements of the analysis. For scenarios requiring well-defined and compact clusters, K-Means is preferable. However, for identifying clusters of varying densities and handling noise, DBSCAN is more suitable.

In conclusion, both algorithms demonstrated their strengths and weaknesses in clustering the e-commerce customer dataset. K-Means provided a straightforward approach with balanced cluster sizes, while DBSCAN offered nuanced segmentation with better-defined clusters and robustness against noise. The comparative analysis underscores the importance of selecting the appropriate clustering algorithm based on the dataset's characteristics and the study's specific objectives.

Discussion

The results from the clustering analysis reveal significant insights into the e-commerce customer dataset and the performance of the K-Means and DBSCAN algorithms. The K-Means algorithm produced moderately well-defined clusters, as indicated by its silhouette score of 0.546. This performance can be attributed to the algorithm's ability to partition data into spherical clusters, which works well when evenly distributed across multiple dimensions. The relatively low Davies-Bouldin index of 0.655 suggests that the clusters formed by K-Means are compact and well-separated. The reasonable Calinski-Harabasz score of 552.9 further supports this finding, indicating a good balance between within-cluster cohesion and between-cluster separation.

In contrast, DBSCAN achieved a higher silhouette score of 0.680, indicating that it formed more distinctly separated clusters than K-Means. This is due to DBSCAN's density-based approach, which is particularly effective in identifying

clusters of varying shapes and densities, and in handling noise. However, the higher Davies-Bouldin index of 1.344 suggests that while DBSCAN can identify well-separated clusters, these clusters are less compact. The significantly higher Calinski-Harabasz score of 1123.9 demonstrates DBSCAN's ability to create highly distinct clusters, making it suitable for datasets with irregular distributions and outliers.

The practical implications of these findings are substantial for e-commerce customer segmentation. With its ability to form balanced clusters, K-Means is ideal for applications requiring clear, well-defined segments of customers based on purchasing behavior and preferences. This can enhance personalized marketing strategies, improve customer engagement, and increase sales by targeting specific customer groups with tailored offers and recommendations.

On the other hand, DBSCAN's strength in identifying clusters with varying densities makes it particularly useful for detecting niche markets or customer segments that exhibit unique behaviors. This can be instrumental in identifying high-value customers, potential churners, or emerging trends within the customer base. By leveraging DBSCAN, e-commerce platforms can develop more refined and adaptive customer segmentation strategies that cater to a wider range of customer behaviors and preferences.

Despite the valuable insights gained, this study has several limitations. The quality of the dataset can significantly influence the clustering results. Any inaccuracies or biases in the data collection process can affect the validity of the clusters formed. Additionally, the assumptions inherent in each algorithm, such as the spherical clusters assumed by K-Means or the density parameters in DBSCAN, may not perfectly align with the actual data distribution. The scope of the analysis is also limited to the specific features and customer behavior captured in the dataset. This study did not consider other relevant factors, such as seasonal variations or external market influences.

Future research can address these limitations by exploring several directions. Improving the algorithms through advanced techniques, such as hybrid clustering methods that combine the strengths of K-Means and DBSCAN, could yield more robust and accurate segmentation results. Incorporating additional data sources, such as social media interactions, customer reviews, and external market data, can provide a more comprehensive view of customer behavior. Moreover, developing new techniques for handling high-dimensional data and identifying meaningful features can further enhance the effectiveness of clustering algorithms in e-commerce customer segmentation. By pursuing these avenues, future studies can build on the findings of this research to develop more sophisticated and effective customer segmentation strategies.

Conclusion

This study provided a comprehensive comparative analysis of the K-Means and DBSCAN algorithms for e-commerce customer segmentation. K-Means formed four distinct clusters, indicating moderate within-cluster variance and good separation. DBSCAN identified multiple clusters with varying sizes and densities, highlighting its effectiveness in handling noise and identifying well-separated clusters. The silhouette scores, Davies-Bouldin indices, and Calinski-Harabasz scores for both algorithms demonstrated their respective strengths

and weaknesses, offering valuable insights into their performance in segmenting the e-commerce customer dataset.

The findings of this study have significant implications for e-commerce businesses. By understanding K-Means' strengths in forming balanced and well-defined clusters, businesses can enhance their customer segmentation strategies to target specific groups with tailored marketing efforts, thereby improving customer engagement and increasing sales. DBSCAN's ability to identify niche segments and handle noise makes it ideal for detecting unique customer behaviors and emerging trends, allowing businesses to develop more adaptive and refined segmentation strategies.

To leverage these findings, e-commerce businesses should consider their specific needs when choosing a clustering algorithm. K-Means is recommended for scenarios requiring clear and balanced segments, such as personalized marketing and customer loyalty programs. In contrast, DBSCAN is suitable for identifying outliers, niche markets, and varied customer behaviors, making it useful for advanced customer analytics and trend detection.

Future research can build on this study by exploring several potential areas. Investigating hybrid clustering methods that combine the strengths of K-Means and DBSCAN could lead to more robust and accurate segmentation results. Additionally, incorporating additional data sources, such as social media interactions, customer reviews, and external market data, can provide a more comprehensive view of customer behavior. Developing new techniques for handling high-dimensional data and identifying meaningful features will further enhance the effectiveness of clustering algorithms. By refining and improving segmentation methodologies, future studies can offer more sophisticated and effective strategies for e-commerce customer segmentation, ultimately leading to better business outcomes and improved customer experiences.

Declarations

Author Contributions

Conceptualization: A.S.P. and T.H.; Methodology: A.S.P.; Software: A.S.P.; Validation: T.H.; Formal Analysis: A.S.P.; Investigation: T.H.; Resources: A.S.P.; Data Curation: A.S.P.; Writing Original Draft Preparation: A.S.P.; Writing Review and Editing: A.S.P. and T.H.; Visualization: A.S.P.; All authors have read and agreed to the published version of the manuscript.

Data Availability Statement

The data presented in this study are available on request from the corresponding author.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M. M. Hassan and M. Tabasum, “Customer Profiling and Segmentation in Retail Banks Using Data Mining Techniques,” Int. J. Adv. Res. Comput. Sci., vol. 9, no. 4, pp. 24–29, 2018, doi: 10.26483/ijarcs.v9i4.6172.

[2] A. U. Khasanah, D. A. Erlangga, and A. M. Jamil, “An Application of Data Mining Techniques in Designing Catalogue for a Laundry Service,” Matec Web Conf., vol. 154, p. 01099, 2018, doi: 10.1051/matecconf/201815401099.

[3] A. Kılıç and E. AKDAMAR, “Market Segmentation of Leisure Boats Exhibited in the Boat Show by Using Multivariate Statistical Techniques,” Pomorstvo, vol. 34, no. 2, pp. 291–301, 2020, doi: 10.31217/p.34.2.10.

[4] S. Guney, S. Peker, and C. Turhan, “A Combined Approach for Customer Profiling in Video on Demand Services Using Clustering and Association Rule Mining,” Ieee Access, vol. 8, pp. 84326–84335, 2020, doi: 10.1109/access.2020.2992064.

[5] A. Ansari and A. Riasi, “Taxonomy of Marketing Strategies Using Bank Customers’ Clustering,” Int. J. Bus. Manag., vol. 11, no. 7, p. 106, 2016, doi: 10.5539/ijbm.v11n7p106.

[6] S. K. Jauhar, B. Chakma, S. S. Kamble, and A. Belhadi, “Digital Transformation Technologies to Analyze Product Returns in the E-Commerce Industry,” J. Enterp. Inf. Manag., vol. 37, no. 2, pp. 456–487, 2023, doi: 10.1108/jeim-09-2022-0315.

[7] H. Abbasimehr and M. Shabani, “A New Methodology for Customer Behavior Analysis Using Time Series Clustering,” Kybernetes, vol. 50, no. 2, pp. 221–242, 2019, doi: 10.1108/k-09-2018-0506.

[8] D. Mensouri, A. Azmani, and M. Azmani, “K-Means Customers Clustering by Their RFMT and Score Satisfaction Analysis,” Int. J. Adv. Comput. Sci. Appl., vol. 13, no. 6, 2022, doi: 10.14569/ijacsa.2022.0130658.

[9] M. Ray and B. K. Mangaraj, “AHP Based Data Mining for Customer Segmentation Based on Customer Lifetime Value,” Int. J. Data Min. Tech. Appl., vol. 5, no. 1, pp. 28–34, 2016, doi: 10.20894/ijdmta.102.005.001.007.

[10] M. A. Varma, “Use of Big Data in the Process of Customer Segmentation in the Retail Sector,” Technoarete Trans. Adv. Data Sci. Anal., vol. 1, no. 2, 2022, doi: 10.36647/ttadsa/01.02.a002.

[11] D. Neha and B. M. Vidyavathi, “A Survey on Applications of Data Mining Using Clustering Techniques,” Int. J. Comput. Appl., vol. 126, no. 2, pp. 7–12, 2015, doi: 10.5120/ijca2015905986.

[12] D. Xu and Y. Tian, “A Comprehensive Survey of Clustering Algorithms,” Ann. Data Sci., vol. 2, no. 2, pp. 165–193, 2015, doi: 10.1007/s40745-015-0040-1.

[13] Z. Guo-lei, J. Li, and H. Li, “Cloud Computing and Its Application in Big Data Processing of Distance Higher Education,” Int. J. Emerg. Technol. Learn. Ijet, vol.

10, no. 8, p. 55, 2015, doi: 10.3991/ijet.v10i8.5280.

[14] B. Goenandar and M. Ariyanti, “Analysis of Demography, Psychograph and Behavioral Aspects of Telecom Customers Using Predictive Analytics to Increase Voice Package Sales,” J. Consum. Sci., vol. 6, no. 1, pp. 1–19, 2021, doi: 10.29244/jcs.6.1.1-19.

[15] B. Rizki, N. G. Ginasta, M. A. Tamrin, and A. Rahman, “Customer Loyality Segmentation on Point of Sale System Using Recency-Frequency-Monetary (RFM) and K-Means,” J. Online Inform., vol. 5, no. 2, p. 130, 2020, doi: 10.15575/join.v5i2.511.

[16] S. Li, “An Improved DBSCAN Algorithm Based on the Neighbor Similarity and Fast Nearest Neighbor Query,” Ieee Access, vol. 8, pp. 47468–47476, 2020, doi: 10.1109/access.2020.2972034.

[17] Z. Wu, L. Jin, J. Zhao, L. Jing, and L. Chen, “Research on Segmenting E-Commerce Customer Through an Improved K-Medoids Clustering Algorithm,” Comput. Intell. Neurosci., vol. 2022, pp. 1–10, 2022, doi: 10.1155/2022/9930613.

[18] K. Lakshmi, N. K. Visalakshi, and S. A. Shanthi, “Data Clustering Using K-Means Based on Crow Search Algorithm,” Sadhana, vol. 43, no. 11, 2018, doi: 10.1007/s12046-018-0962-3.

[19] J. M. Dudik, A. Kurosu, J. L. Coyle, and E. Sejdić, “A Comparative Analysis of DBSCAN, K-Means, and Quadratic Variation Algorithms for Automatic Identification of Swallows From Swallowing Accelerometry Signals,” Comput. Biol. Med., vol. 59, pp. 10–18, 2015, doi: 10.1016/j.compbiomed.2015.01.007.

[20] A. P. Navato and A. Mueller, “Enabling Automatic Detection of Anomalies in Wastewater: A Highly Simplified Approach to Defining ‘Normal’ in Complex Chemical Mixtures,” Front. Water, vol. 3, 2021, doi: 10.3389/frwa.2021.734361.