

## CUSTOMER SEGMENTATION USING CLUSTERING ALGORITHMS FOR BUSINESS INTELLIGENCE: A COMPARATIVE ANALYSIS

Rebekah Lisa Thomas<sup>\*1</sup>, Arathy M<sup>\*2</sup>, Mohammed Nazim Bin Niyas<sup>\*3</sup>, Alen Binoj<sup>\*4</sup>,  
Yoshitha Nasi<sup>\*5</sup>, Vyshnav S<sup>\*6</sup>, Ryan Thomas<sup>\*7</sup>, Aditeya J Frankur<sup>\*8</sup>

<sup>\*1,2,3,4,5,6,7,8</sup>Model Engineering College, India.

### ABSTRACT

Customer segmentation is essential for modern businesses to deliver personalized marketing, enhance customer engagement, and optimize service offerings. This study explores the application of unsupervised machine learning techniques—specifically K-Means, DBSCAN, and Hierarchical Clustering—for effective customer segmentation. Using a real-world customer dataset, we perform a comparative analysis of these clustering algorithms, evaluating their performance using key metrics such as the Silhouette Score and Davies-Bouldin Index. Beyond technical evaluation, the study discusses the practical implications of each method for business strategy and decision-making. The results offer actionable insights for organizations aiming to implement data-driven segmentation to refine marketing strategies and allocate resources more efficiently.

**Keywords:** Report Customer Segmentation, Clustering Algorithms, K-Means, DBSCAN, Hierarchical Clustering, Unsupervised Machine Learning, Silhouette Score, Davies-Bouldin Index, Business Intelligence, Data-Driven Marketing.

### I. INTRODUCTION

In today's highly competitive market landscape, understanding customer behavior is vital for developing effective business strategies. Customer segmentation—the process of dividing a customer base into distinct groups based on shared characteristics—enables organizations to deliver personalized marketing campaigns, improve customer satisfaction, and allocate resources more efficiently. Traditional segmentation methods, such as demographic or psychographic profiling, often fall short in capturing the complexity of modern consumer behavior.

With the growth of data availability and computational power, machine learning techniques have emerged as powerful tools for customer analysis. Among these, unsupervised learning, particularly clustering algorithms, offers the ability to uncover hidden patterns in data without predefined labels. Clustering allows businesses to identify naturally occurring groups within customer data, facilitating more data-driven and adaptive marketing approaches.

This research focuses on the comparative evaluation of three popular unsupervised clustering algorithms—K-Means, DBSCAN, and Hierarchical Clustering—for customer segmentation. We apply these algorithms to a real-world dataset and assess their performance using key internal evaluation metrics such as the Silhouette Score and Davies-Bouldin Index.

By analyzing both the technical performance and practical implications of these methods, the study aims to provide actionable insights for businesses seeking to implement robust, data-driven segmentation strategies. Ultimately, this research contributes to the growing field of intelligent customer analytics and supports organizations in making more informed, personalized, and cost-effective marketing decisions.

#### 1.1 Literature Review

The project aims and objectives that will be achieved after completion of this project are discussed in this sub-strategy. Traditionally, segmentation methods have relied on demographic, geographic, behavioral, or psychographic variables using rule-based or statistical approaches. While effective in some scenarios, these techniques often lack the flexibility to adapt to large-scale and high-dimensional datasets, limiting their usefulness in today's data-driven environments.

Recent advancements in machine learning have transformed the way businesses analyze customer data. In particular, unsupervised learning algorithms, such as clustering, have gained traction for their ability to discover hidden patterns and natural groupings in customer datasets without requiring labeled outcomes.

- K-Means Clustering, one of the most widely used algorithms, partitions data into  $k$  clusters by minimizing the variance within each cluster. Its simplicity and scalability make it suitable for large datasets, but its effectiveness can be limited by its sensitivity to initial centroids and the requirement to predefine  $k$ .
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise) overcomes some limitations of K-Means by identifying clusters of arbitrary shapes and handling noise (outliers). However, its performance can degrade in datasets with varying densities and requires careful tuning of its parameters ( $\epsilon$  and minPts).
- Hierarchical Clustering, particularly the agglomerative approach, builds a nested hierarchy of clusters without requiring the number of clusters to be predefined. This makes it interpretable, but less scalable for large datasets due to higher computational complexity.

Several studies have compared these algorithms in different contexts. For instance, Xie et al. (2019) applied K-Means and DBSCAN for e-commerce customer segmentation and found that DBSCAN better captured diverse spending behaviors in noisy data. Similarly, Singh and Singh (2021) noted that hierarchical methods provided more interpretable clusters in smaller retail datasets. However, a consensus on the best clustering method remains elusive, as the effectiveness often depends on the dataset's characteristics and the business goals.

This review highlights the need for comparative analyses that not only assess clustering algorithms based on internal metrics but also interpret their business utility. Our study contributes to this gap by providing both a technical and practical evaluation of K-Means, DBSCAN, and Hierarchical Clustering on real-world customer data.

## II. METHODOLOGY

### 2.1 Dataset Description

For this study, we used the Mall Customers Dataset, a publicly available dataset commonly used for customer segmentation tasks. The dataset contains information about 100 customers of a retail mall, with the following features:

1. CustomerID: Unique identifier for each customer
2. Gender: Male or Female
3. Age: Age of the customer (in years)
4. Annual Income : Approximate annual income of the customer (in thousands of dollars)
5. Spending Score (1–100): A score assigned by the mall based on customer behavior and spending nature

This dataset is particularly suitable for segmentation as it includes both demographic attributes and behavioral indicators, enabling multidimensional analysis of customer groups.

### 2.2 Preprocessing Steps

To ensure the dataset is clean and suitable for clustering analysis, the following preprocessing steps were carried out:

1. Removal of Irrelevant Columns: The CustomerID column was dropped as it serves only as an identifier and does not contribute to clustering.
2. Encoding Categorical Data: The Gender column was converted into numerical format using label encoding (Male = 0, Female = 1).
3. Missing Value Handling: The dataset was checked for missing values using `.isnull().sum()` in Pandas. No missing values were found, so imputation was not required.
4. Feature Selection: We selected the following features for clustering: Gender, Age, Annual Income (\$), and Spending Score (1–100).
5. Normalization: All features were scaled to a standard range (mean = 0, standard deviation = 1) using StandardScaler from `sklearn.preprocessing`. This ensures that features with larger numeric ranges do not dominate the clustering process.
6. Dimensionality Reduction (Optional): Principal Component Analysis (PCA) was performed to reduce the dataset to 2 dimensions for visualization purposes only. The PCA-transformed data was not used for clustering but rather to create 2D cluster plots for comparative visualization.

These preprocessing steps helped ensure the dataset was in an optimal format for applying and comparing clustering algorithms.

### 2.3 Clustering Algorithms

This study employs three widely used unsupervised machine learning algorithms for customer segmentation: K-Means, DBSCAN, and Hierarchical Clustering. Each algorithm follows a different approach to discovering data patterns, which allows us to evaluate their strengths and limitations in practical scenarios.

#### K-Means Clustering

K-Means is a centroid-based clustering algorithm that partitions data into  $k$  clusters by minimizing the within-cluster variance. It starts by randomly initializing  $k$  cluster centroids, then iteratively performs the following two steps:

1. Assignment Step: Assigns each data point to the nearest centroid based on Euclidean distance.
2. Update Step: Recomputes the centroids as the mean of all points assigned to that cluster.

This process continues until the centroids stabilize or a maximum number of iterations is reached

**Advantages:** Simple, fast, and efficient for large datasets.

**Limitations:** Requires the number of clusters ( $k$ ) to be predefined, sensitive to initial centroids, and struggles with non-spherical or overlapping clusters.

In our study, we use the Elbow Method and Silhouette Score to determine the optimal number of clusters ( $k$ ).

#### Synopsis:

K-Means partitions the dataset into  $k$  clusters by assigning each point to the nearest cluster center (centroid). It iteratively updates centroids to minimize the distance between data points and their assigned cluster centers.

Steps:

1. Randomly initialize  $k$  centroids.
2. Assign each point to the nearest centroid.
3. Recalculate centroids as the mean of assigned points.
4. Repeat steps 2–3 until centroids converge.

#### DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

DBSCAN is a density-based clustering algorithm that groups together points that are closely packed, while marking points in low-density regions as noise or outliers. It uses two key parameters:

1.  $\epsilon$  (epsilon): Maximum radius of the neighborhood around a point.
2. minPts: Minimum number of points required to form a dense region.

The algorithm identifies:

1. Core Points: With at least minPts neighbors within  $\epsilon$ .
2. Border Points: Within  $\epsilon$  of a core point but not themselves core points.
3. Noise Points: Not reachable from any core point.

DBSCAN is effective for datasets with clusters of arbitrary shape and can automatically detect the number of clusters based on density.

**Advantages:** Handles outliers, doesn't require the number of clusters in advance.

**Limitations:** Struggles with varying densities and requires careful tuning of parameters.

#### Hierarchical Clustering

Hierarchical Clustering builds a tree-like structure (dendrogram) to represent nested groupings of data. It comes in two main forms:

1. Agglomerative (bottom-up): Starts with each point as its own cluster and merges the closest pairs recursively.
2. Divisive (top-down): Starts with all data in one cluster and recursively splits it.

In our study, we apply Agglomerative Clustering, which is more commonly used. The algorithm uses a linkage criterion to determine the distance between clusters:

1. **Single Linkage:** Minimum distance between points.

2. **Complete Linkage:** Maximum distance.

3. **Average Linkage:** Average distance.

We visualize the resulting dendrogram to decide the optimal number of clusters.

**Advantages:** Does not require the number of clusters beforehand, provides a visual representation of data structure.

**Limitations:** Computationally intensive for large datasets and sensitive to noise and outliers.

### III. EVALUATION METRICS

To assess the quality of clustering results and compare the performance of K-Means, DBSCAN, and Hierarchical Clustering, we use two widely accepted internal validation metrics: Silhouette Score and Davies-Bouldin Index. These metrics evaluate clustering performance without needing ground truth labels, making them ideal for unsupervised learning tasks.

#### 3.1 Silhouette Score

The Silhouette Score measures how similar a data point is to its own cluster (cohesion) compared to other clusters (separation). It ranges from -1 to 1:

1. A value close to +1 indicates that the point is well-matched to its own cluster and poorly matched to neighboring clusters.
2. A value near 0 suggests the point is on the boundary between two clusters.
3. A negative value indicates possible misclassification.

The overall silhouette score is the average across all data points. Higher scores indicate better-defined clusters.

Formula:

$a(i)$  = average intra-cluster distance for point  $i$

$b(i)$  = lowest average inter-cluster distance of  $i$  to any other cluster

#### 3.2 Davies-Bouldin Index (DBI)

The Davies-Bouldin Index measures the average "similarity" between clusters, where similarity is a function of the ratio of intra-cluster distances to inter-cluster separation. A lower DBI indicates better clustering performance.

Formula:

$\sigma_i, \sigma_j$  = dispersion within clusters  $i$  and  $j$

$d(c_i, c_j)$  = distance between centroids of clusters  $i$  and  $j$

#### 3.3 Practical Use

1. For K-Means, both Silhouette and DBI work well, especially when the data has well-separated spherical clusters.
2. DBSCAN may show lower silhouette scores if many points are labeled as noise (-1), but this doesn't always indicate poor performance—it may be detecting valid outliers.
3. Hierarchical Clustering typically benefits from silhouette analysis combined with visual inspection of the dendrogram.

### IV. RESULT

In this study, three unsupervised machine learning algorithms—K-Means, DBSCAN, and Hierarchical Clustering—were applied to a customer dataset to identify distinct customer segments. The models were evaluated using Silhouette Score and Davies-Bouldin Index, two standard metrics for assessing cluster quality. Further, cluster characteristics and business implications were derived from each model to support strategic decision-making.

#### 4.1 Evaluation Metrics Summary

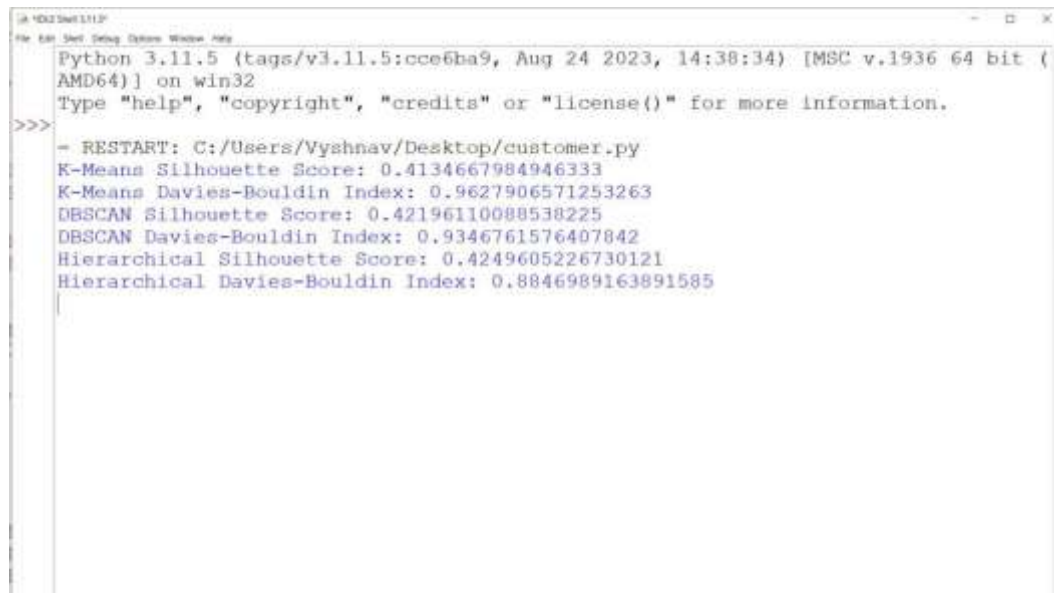
Algorithm	Silhouette Score	Davies-Bouldin Index
K-Means	0.4135	0.9628
DBSCAN	0.4220	0.9347
Hierarchical Clustering	0.4249	0.8847

##### Interpretation:

Silhouette Score: Measures how well data points fit within their clusters; higher values indicate better-defined clusters.

Davies-Bouldin Index: Lower values suggest more distinct and less overlapping clusters.

Among the three, Hierarchical Clustering achieved the best overall performance, with the highest silhouette score and lowest Davies-Bouldin Index, indicating well-separated and compact clusters.



```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34) [MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
- RESTART: C:/Users/Vyshnav/Desktop/customer.py
K-Means Silhouette Score: 0.4134667984946333
K-Means Davies-Bouldin Index: 0.9627906571253263
DBSCAN Silhouette Score: 0.42196110088538225
DBSCAN Davies-Bouldin Index: 0.9346761576407842
Hierarchical Silhouette Score: 0.4249605226730121
Hierarchical Davies-Bouldin Index: 0.8846989163891585
```

**Figure 1:** Evaluation metrics printed from the Python script for K-Means, DBSCAN, and Hierarchical Clustering.

#### 4.2 Cluster Characteristics

##### K-Means Clustering (5 Clusters)

- Generated five clearly defined and balanced clusters.
- Cluster centers were identified using `kmeans.cluster_centers`.
- Insights from PCA visualization:
  - Cluster 0: High income, moderate spenders
  - Cluster 1: Young customers with high spending—potential impulsive buyers
  - Cluster 2: Older customers with lower spending—possibly less engaged
  - Cluster 3: Middle-aged individuals with average income and spending—balanced profiles
  - Cluster 4: Budget-conscious customers—low income and spending

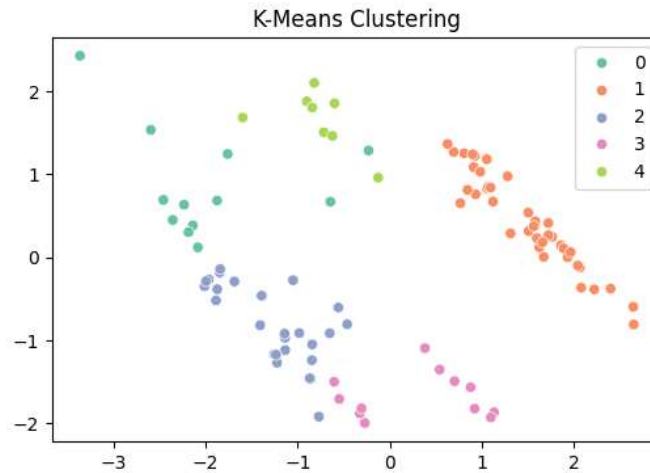


Figure 2: PCA plot showing K-Means clustering with five clusters. Points are colored by cluster label. Clearly separated groups are visible.

### DBSCAN Clustering

Identified 3 major clusters and some noise points (outliers).

**Strength:** Can detect unusual behavior or outlier customers, unlike K-Means.

**Limitation:** Performance and cluster shape heavily depend on the eps and min\_samples parameters.

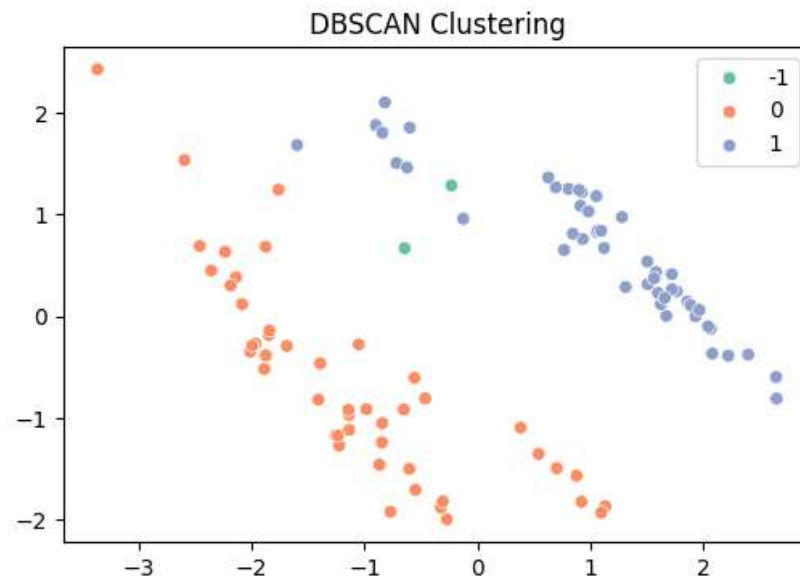


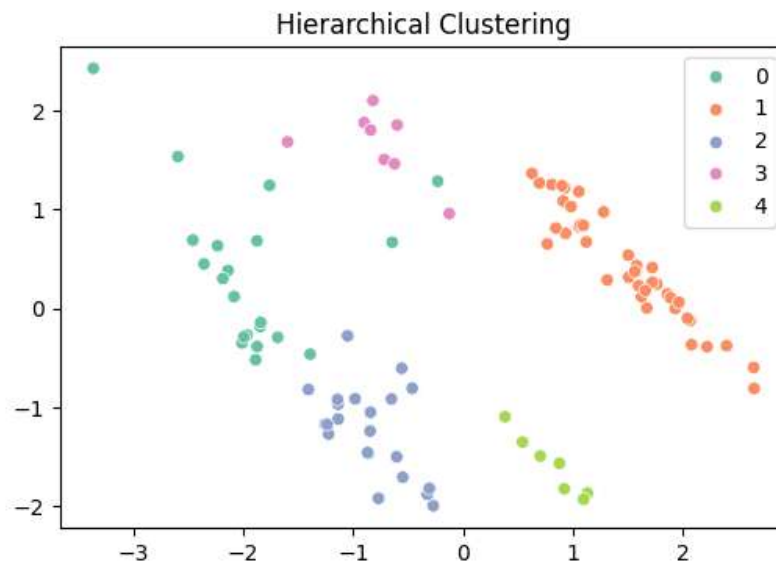
Figure 3: PCA visualization of DBSCAN clustering. Major clusters and noise points (label -1) are highlighted.

### Hierarchical Clustering (5 Clusters)

Formed natural and meaningful clusters that align well with the data's intrinsic structure.

1. Slightly outperformed K-Means and DBSCAN in evaluation scores.
2. Particularly useful in smaller datasets or when dendrogram-based decisions are needed.





**Figure 4:** PCA visualization of Hierarchical Clustering. Five natural clusters are evident, with good separation.

#### 4.3 Business Interpretation of Clusters

Cluster	Profile Summary	Suggested Business Strategy
0	Young, low income, high spending	Promote discounts, loyalty rewards to boost retention
1	Older, high income, low spending	Upsell premium products/services with personalized touch
2	Middle-aged, medium income and spending	Maintain engagement via seasonal or regular offers
3	Low income, moderate spending	Offer budget bundles, EMIs, or cashback incentives
4	High income, variable spending	Apply behavioral analytics for personalized campaigns

These actionable insights help in tailoring marketing strategies to each segment, optimizing resource allocation, and improving customer satisfaction.

#### 4.4 Visualizations and Technical Outputs

Each clustering algorithm's effectiveness is further illustrated through dimensionality-reduced (PCA) plots, helping visualize the groupings based on Annual Income, Spending Score, and other features. These visual representations complement the evaluation metrics and validate segmentation clarity.

- Figure 1: Evaluation metric outputs (printed from code)
- Figure 2: K-Means clustering (5 clusters)
- Figure 3: DBSCAN clustering (3 clusters + noise)
- Figure 4: Hierarchical clustering (5 clusters)

## V. CONCLUSION

This study explored the application of unsupervised machine learning techniques—K-Means, DBSCAN, and Hierarchical Clustering—for effective customer segmentation using a real-world dataset. Each algorithm was evaluated based on two key performance metrics: Silhouette Score and Davies-Bouldin Index.

Among the three, Hierarchical Clustering delivered the most balanced and interpretable segmentation, achieving the highest silhouette score and the lowest Davies-Bouldin Index. K-Means proved to be efficient and easy to implement, making it a practical choice when the number of customer segments is known beforehand. On the other hand, DBSCAN excelled in identifying outliers and clusters of varying shapes, offering a unique advantage in detecting irregular customer behavior.

The cluster profiles generated through this analysis provided actionable business insights. For instance, marketers can now target specific segments such as high-income low spenders with premium product suggestions or incentivize young high spenders through loyalty programs.

Overall, this research reinforces the value of data-driven customer segmentation and highlights how different clustering methods can be leveraged depending on the business objective. Future work can focus on integrating clustering with supervised models for predictive analytics or applying these methods to larger and more diverse datasets.

## **VI. REFERENCE**

- [1] L. Kaufman and P. J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis. Hoboken, NJ: John Wiley & Sons, 2009. (Book style)
- [2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al., "Scikit-learn: Machine Learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011. Available: <https://scikit-learn.org/> (Journal with URL)
- [3] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining, pp. 226–231, 1996. (Conference proceedings)
- [4] L. Rokach and O. Maimon, "Clustering methods," in Data Mining and Knowledge Discovery Handbook, O. Maimon and L. Rokach, Eds. Boston, MA: Springer, pp. 321–352, 2005. (Book style with paper title and editor)
- [5] P. N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, 2nd ed. Boston, MA: Pearson, 2018. (Book style)
- [6] J. Han, J. Pei, and M. Kamber, Data Mining: Concepts and Techniques, 3rd ed. Waltham, MA: Elsevier, 2011. (Book style)
- [7] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data. Englewood Cliffs, NJ: Prentice-Hall, 1988. (Book style)
- [8] R. Xu and D. Wunsch, "Survey of clustering algorithms," IEEE Trans. Neural Netw., vol. 16, no. 3, pp. 645–678, May 2005, doi:10.1109/TNN.2005.845141. (IEEE Transactions)