Customer Segmentation: A Comparative Analysis of Clustering Techniques on Online Retail Data

Jayanath Shanil Praveen Diwakar
Department of Computer Science and Engineering
University of Moratuwa
shanil.22@cse.mrt.ac.lk

***Abstract*—This report details a comprehensive customer segmentation analysis using an online retail dataset. This analysis consists of a complete data preprocessing pipeline to derive key features which are Recency, Frequency, and Monetary. Three distinct clustering algorithms such as K-Means, Hierarchical Clustering, and DBSCAN are applied to the standardized data. A thorough evaluation is performed for each method using metrics such as Silhouette Score and intra-cluster and inter-cluster distances. The results are visualized using Principal Component Analysis (PCA) to provide a clear understanding of the cluster structures. The report concludes with a business-focused interpretation of the resulting customer segments and a discussion of the theoretical underpinnings, strengths, and limitations of each clustering approach.**

***Index Terms*—Customer Segmentation, Clustering, RFM Analysis, K-Means, Hierarchical Clustering, DBSCAN**

I. Introduction

Customer segmentation is a critical component of modern business strategy, enabling companies to tailor marketing efforts, improve customer satisfaction, and optimize resource allocation. This analysis focuses on segmenting customers of a UK-based online retail store using their transactional data. The process involves generating RFM (Recency, Frequency, Monetary) features, which serve as the foundation for understanding customer behavior. The analysis is performed using a comparative approach, applying and evaluating three distinct clustering techniques to identify the most effective method for this specific business problem.

II. Data Preparation and Feature Engineering

The raw transactional data was first subjected to a comprehensive data cleaning process to ensure accuracy and relevance. This involved removing records with missing ‘CustomerID‘ and filtering for transactions solely from the United Kingdom. All cancelled orders and records with invalid quantity or price values were also systematically removed.

Following the cleaning stage, three key customer-level features were engineered:

- **Recency:** The number of days elapsed since a customer's last purchase.
- **Frequency:** The total number of unique purchases made by a customer.
- **Monetary:** The total amount of money a customer has spent.

To ensure that all features contribute equally to the clustering process, the RFM data was standardized using a ‘StandardScaler‘. This step is crucial for distance-based algorithms, as it prevents features with larger scales, such as monetary value, from disproportionately influencing the clustering results. The final standardized data was then used for all subsequent clustering analyses.

III. Clustering Approach and Visualization

Three different clustering algorithms were applied to the standardized RFM data to identify customer segments.

A. K-Means Clustering

K-Means clustering was performed to partition customers into a predefined number of groups. The optimal number of clusters, k, was determined using the Elbow Method. This technique involves plotting the Within-Cluster Sum of Squares (WCSS) for a range of k values and selecting the point where the rate of decrease in WCSS slows down, indicating a good balance between a low WCSS and a reasonable number of clusters. According to this method 3 is considered as the optimal number of clusters. For visualization, Principal Component Analysis (PCA) was applied to reduce the 3-dimensional RFM data to two principal components. This allowed for the creation of a 2D scatter plot where each point represents a customer, colored according to their cluster assignment.

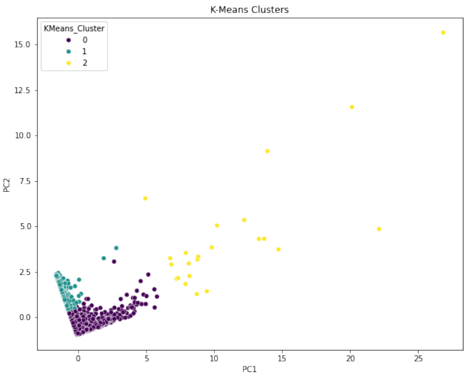

Fig. 1. A 2D visualization of the K-Means clusters.

B. Hierarchical Clustering (AGNES)

Hierarchical clustering was applied to build a hierarchy of clusters. Three different linkage methods were tested: single, complete, and average. A dendrogram was plotted for each linkage method, providing a visual representation of how clusters were formed.

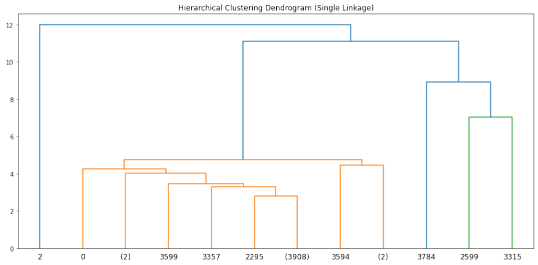

Fig. 2. Dendrogram showing clusters formed using Single Linkage.

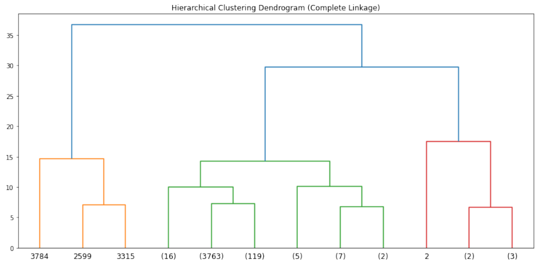

Fig. 3. Dendrogram showing clusters formed using Complete Linkage.

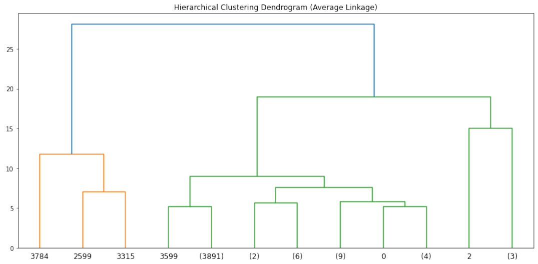

Fig. 4. Dendrogram showing clusters formed using Average Linkage.

The optimal number of clusters was chosen by identifying the longest vertical lines that are not attached to other clusters. These lines represent the greatest dissimilarity between clusters. The next step involves finding a specific point on the y-axis where a significant vertical distance exists between two merges. This large gap indicates that the clusters being joined are quite dissimilar. By drawing an imaginary horizontal line across the dendrogram at this height, the number of vertical lines it intersects represents the optimal number of clusters. Essentially, the longer the vertical line, the more distinct and well-separated the clusters are from each other. The analysis of the dendrograms revealed that the optimal number of clusters for single linkage is 5, while both complete linkage and average linkage are best represented by 4 clusters. The final clusters were visualized using PCA, similar to the K-Means approach.

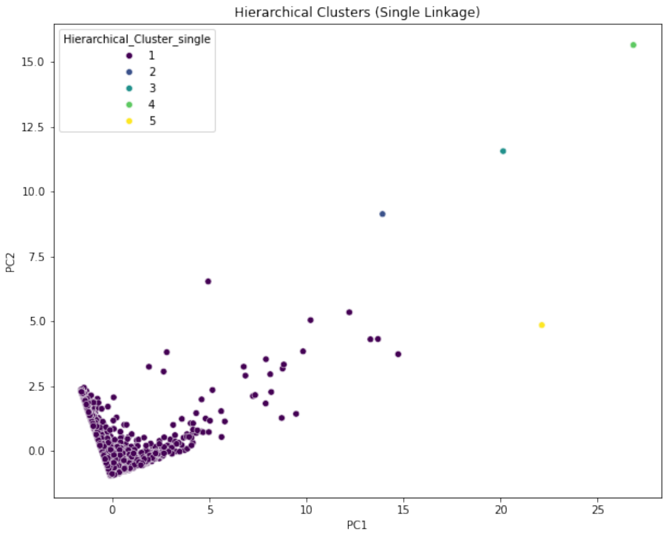

Fig. 5. A 2D visualization of the Hierarchical clusters using single linkage.

C. DBSCAN

DBSCAN, a density-based algorithm, was used to discover arbitrarily shaped clusters and identify noise. The two primary parameters for DBSCAN are min-samples and eps. The min-samples was set to 5, while the optimal eps was chosen by examining a K-distance plot, where the value at the ”elbow” of the curve provides a suitable threshold.The chosen eps value was 0.6575. The resulting clusters were visualized using PCA, which clearly showed the identified clusters as well as the noise points.

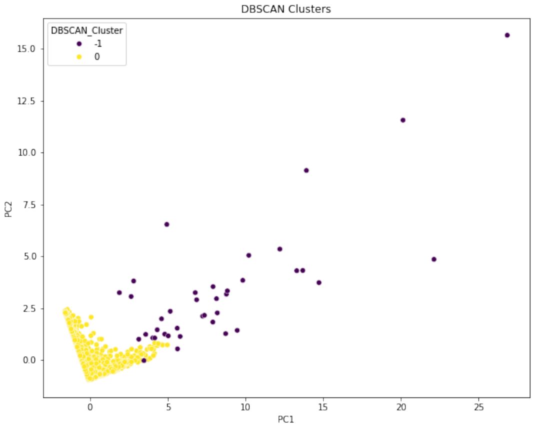

Fig. 6. Visualization of DBSCAN clusters, highlighting noise points.

IV. Clustering Evaluation Metrics

Each clustering result was evaluated using the following metrics.

A. Silhouette Score

The Silhouette Score is a measure of how well-defined clusters are, with a value ranging from -1 to +1. A score close to +1 indicates that a data point is well-matched to its own cluster and poorly matched to neighboring clusters, suggesting a good clustering result. A score near 0 indicates overlapping clusters, while a negative score means points might be assigned to the wrong cluster. For the DBSCAN analysis, noise points were excluded from this calculation, as the score is not defined for them.

B. Inter-cluster and Intra-cluster Distances

Intra-cluster Distance measures the average distance between all data points within the same cluster. The goal is to have a low intra-cluster distance, which indicates that the clusters are compact and dense.

Inter-cluster Distance measures the average distance between the data points of different clusters. The goal is to have a high inter-cluster distance, which indicates that the clusters are well-separated from one another.

V. Clustering Results and Evaluation

This section presents the results and evaluation metrics obtained from applying the three clustering algorithms to the RFM-based customer data.

A. K-Means Clustering

The optimal number of clusters for K-Means was determined to be 3. The following metrics were recorded for this configuration:

- **Silhouette Score:** 0.58
- **Intra-cluster Distance (WCSS):** 5114.68
- **Average Inter-cluster Distance:** 8.95

B. Hierarchical Clustering

The optimal number of clusters identified from the dendrograms was 5 for single linkage and 4 for both complete and average linkages. The evaluation metrics for each linkage method are as follows:

1) Single Linkage:

- **Silhouette Score:** 0.93
- **Average Intra-cluster Distance:** 1.50
- **Average Inter-cluster Distance:** 24.57

2) Complete Linkage:

- **Silhouette Score:** 0.89
- **Average Intra-cluster Distance:** 4.08
- **Average Inter-cluster Distance:** 24.33

3) Average Linkage:

- **Silhouette Score:** 0.90
- **Average Intra-cluster Distance:** 3.54
- **Average Inter-cluster Distance:** 24.15

C. DBSCAN

The DBSCAN algorithm was executed with ‘min-samples = 5‘ and an optimal ‘eps‘ of 0.6575. The algorithm identified only one cluster and several outliers.

- **Silhouette Score:** A Silhouette Score could not be calculated as more than one cluster was not formed.
- **Average Intra-cluster Distance:** 1.37
- **Average Inter-cluster Distance:** Not applicable as there is only one cluster.

When comparing the three clustering methods, Hierarchical Clustering with single linkage performed the best, yielding the highest Silhouette Score (0.93), which signifies a highly effective and well-separated grouping of customers. K-Means had a moderate Silhouette Score, indicating that while it did form distinct clusters, they were not as well-defined as those from hierarchical clustering. DBSCAN proved to be unsuitable for this dataset, as it failed to identify multiple distinct clusters, instead classifying several data points as outliers. This suggests that the customer data has varying densities that DBSCAN’s parameters could not effectively capture in a meaningful way.

VI. Theoretical Discussion

1) **Strengths and Limitations of Each Clustering Method**
   - **K-Means:**
     Strengths: It is fast and efficient, making it great for large datasets. It is also easy to implement and understand.
     Limitations: The number of clusters has to be decided beforehand. It is very sensitive to outliers and can only find clusters that are roughly round or spherical.
   - **Hierarchical Clustering:**
     Strengths: The number of clusters doesn’t need to be decided in advance. The process results in a dendrogram , which shows the relationships and hierarchy between data points, giving more insight into the data’s structure.
     Limitations: This method can be slow and computationally expensive, especially for large datasets. It is also sensitive to noise and the choice of distance metric can significantly change the results.
   - **DBSCAN:**
     Strengths: It is excellent at finding clusters of any shape, not just round ones. It can identify and label outliers as well.
     Limitations: It can be difficult to choose the right density parameters which are Eps and MinPts. It struggles with datasets where the clusters have very different densities.
2) **Assumptions about Data Structure**
   - **K-Means:** This algorithm assumes that clusters are spherical, of a similar size, and are well-separated from each other.

- **Hierarchical Clustering:** This assumes that there is an underlying hierarchical structure in the data, where similar objects are closer together and can be grouped in a nested fashion.
- **DBSCAN:** This algorithm assumes that clusters are areas of high density separated by areas of low density.

3) **Why DBSCAN Detects "Noise" While K-means Cannot** K-Means forces every data point to be assigned to a cluster, even if it is an outlier located far away from any cluster's center. It simply assigns the outlier to the nearest cluster, which can distort the cluster's shape and location.On the other hand DBSCAN, defines clusters based on the concept of density. It finds groups of data points that are closely packed together. Any data point that is not part of a sufficiently dense region is automatically classified as ”noise”.
4) **Necessity of Scaling Features** Scaling features before clustering is a crucial preprocessing step, particularly for algorithms like K-Means and Hierarchical Clustering that rely on distance measurements. The RFM features possessed vastly different scales. For instance, ‘Monetary‘ values can be orders of magnitude larger than ‘Frequency‘ or ‘Recency‘. Without standardization, the distance calculations used by the algorithms would be dominated by the ‘Monetary‘ feature, leading to biased and inaccurate clustering results. Scaling ensures that all features contribute equally to the final segmentation.
5) **Implications of Using Different Distance Metrics** The choice of a distance metric can significantly alter clustering results. Euclidean distance (the straight-line distance) is the most common metric. It works well for finding spherical or compact clusters.

   Manhattan distance (the ”city block” distance) can be more suitable for datasets with high dimensionality or when the data is not well-separated.

   In Hierarchical Clustering (AGNES), some specific distance metrics were used to measure the distance between two clusters.

   Single Link: This is the shortest distance between any two points in the two clusters. This can lead to ”chaining”, where long, thin clusters are formed.

   Complete Link: This is the longest distance between any two points in the two clusters. This tends to find more compact, round clusters.

   Average Link: This is the average of all distances between points in the two clusters.

VII. Interpretation of Clusters and Business Insights

This section provides a business-oriented interpretation of the customer segments identified by the K-Means and Hierarchical Clustering (Single Linkage) methods.

A. K-Means Customer Segments

The K-Means analysis identified three distinct customer segments based on their RFM behavior.

- **Cluster 1: Active, Consistent Buyers** who have recently made purchases with moderate frequency and monetary value. They contribute a solid amount to revenue.
- **Cluster 2: Inactive Customers**, defined by low frequency and monetary value and a very high recency. Purchases were made a long time ago. Marketing strategies must be implemented to re-engage these customers with the business.
- **Cluster 3: Most Valuable Customers** with very high frequency and exceptionally high monetary value, combined with very low recency. They are responsible for a significant portion of revenue.

B. Hierarchical Clustering Customer Segments (Single Linkage)

As the best-performing method, single linkage hierarchical clustering provided a more granular segmentation with five distinct clusters, offering specific business insights.

- **Cluster 1:** The largest segment of **at-risk or inactive** customers, characterized by high recency and low transaction values.Strategies should be designed to be scalable and automated, such as a targeted email campaign with an exclusive discount to encourage a return purchase.
- **Cluster 2:** A single customer identified as a **high-value, new customer** with an exceptionally high monetary value from a recent purchase.
- **Cluster 3:** A small group of **highly engaged and profitable customers** with high frequency, very high monetary value, and low recency.Retention efforts on this segment through high-touch service and loyalty rewards have to be implemented to maximize their lifetime value.
- **Cluster 4:** A single **top-tier contributor** with very high frequency and extremely high monetary value, driven by recent, frequent, and high-value transactions.
- **Cluster 5:** Another single-customer segment identified as a **very frequent but mid-range spender** with very high frequency and moderate monetary value.

References

[1] S. Jaiswal, “Elbow Method for Optimal Value of k in KMeans,” GeeksforGeeks, [Online]. Available: https://www.geeksforgeeks.org/machine-learning/elbow-method-for-optimal-value-of-k-in-kmeans/

[2] F. Pedregosa, G. Varoquaux, A. Gramfort et al., “Scikit-learn: Machine Learning in Python,” J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

[3] S. Sefidian, “How to Determine Epsilon and MinPts Parameters of DBSCAN Clustering,” Sefidian.com, [Online]. Available: https://www.sefidian.com/2022/12/18/how-to-determine-epsilon-and-minpts-parameters-of-dbscan-clustering/