Data Science and Engineering (DSE) Record, Volume 6, issue 1.

EFFICIENT SEGMENTATION OF CUSTOMERS BASED ON RFM ANALYSIS

Chattrapat Poonsin [1] and Pruet Boonma [2]

[1] Master's Degree Program in Data Science, Chiang Mai University, Chiang Mai, Thailand
[2] Department of Computer Engineering, Faculty of Engineering, Chiang Mai University, Chiang Mai, Thailand
chattrapat_p@cmu.ac.th

Abstract. Customer segmentation is a vital component of data-driven marketing, enabling businesses to understand customer behavior and enhance strategic decision-making. This study explores an efficient segmentation approach using Recency, Frequency, and Monetary (RFM) analysis, combined with multiple clustering techniques, to identify optimal customer groups. Four clustering approaches were implemented and compared centroid-based density based, distribution-based, and hierarchical clustering (Agglomerative). Each of these algorithms were evaluated based on its ability to form well-separated and meaningful clusters, with silhouette score as the primary performance metric. The dataset was standardized before applying the clustering models to ensure comparability. The results reveal that different algorithms exhibit varying strengths depending on the underlying data structure. K-Means demonstrated efficiency in partitioning customers into distinct groups but struggled with non-spherical clusters. DBSCAN effectively identified outliers but was sensitive to parameter tuning. GMM provided flexibility by modeling cluster probability distributions, making it suitable for overlapping customer behaviors. Hierarchical clustering offered an interpretable structure but required significant computational resources for large datasets. Overall, the findings highlight the importance of selecting an appropriate clustering technique for customer segmentation based on data characteristics. This study provides valuable insights for businesses aiming to develop marketing strategies through data-driven segmentation.

Keywords: Customer Segmentation, RFM Analysis, Clustering Algorithms, Silhouette Score, K-Means, K-Medoids, DBSCAN, Gaussian Mixture Model, Hierarchical Clustering

1 Introduction

In today's highly competitive market, gaining insights into customer behavior is essential for business success for enhancing customer satisfaction and optimizing marketing strategies. Businesses are increasingly leveraging customer segmentation to

group customers into meaningful categories based on their purchasing patterns. RFM (Recency, Frequency, and Monetary) analysis is a widely used technique that evaluates customer engagement based on three key metrics: how recently a customer made a purchase (Recency), how often they purchase (Frequency), and how much they spend (Monetary) [1]. By analyzing these factors, businesses can identify valuable customers, tailor marketing efforts, and improve customer retention [2].

Conventional segmentation methods, such as rule-based approaches or demographic, often fail to capture complex purchasing patterns [3]. In contrast, machine learning and clustering algorithms have gained significant attention for their ability to automatically group customers based on behavioral data. Among clustering techniques, centroid-based [4], density-based [5], distribution-based [6], and hierarchical clustering (Agglomerative) [7] are widely used for segmentation tasks. However, the effectiveness of each method varies depending on data distribution, cluster shape, and scalability [8].

Despite the growing adoption of clustering algorithms for customer segmentation, selecting an appropriate technique remains a challenge. Many businesses rely on K-Means clustering, which assumes that clusters are similar sizes and spherical, potentially leading to misleading results when customer behavior does not conform to these assumptions [9]. DBSCAN, while effective for detecting outliers, can struggle with defining appropriate density parameters [10]. GMM provides flexibility by modeling data as probability distributions but may be computationally expensive [11]. Hierarchical clustering offers interpretability but becomes inefficient for large datasets [12]. Given these variations, an in-depth comparison of these clustering methods in the context of RFM-based customer segmentation is necessary to determine the most effective approach [13].

2 Literature Review

2.1 Data analytics in marketing

Customer segmentation is a key component of marketing analytics, allowing businesses to categorize customers based on shared characteristics. Traditionally, segmentation methods have primarily depended on demographic data, but modern approaches leverage machine learning techniques such as clustering algorithms like K-Means, DBSCAN, GMM to create dynamic customer segments [14]. RFM analysis is a widely used technique that enables marketers to classify customers based on their purchasing behavior [15].

Predictive analytics utilizes historical data and machine learning models to anticipate future customer behavior. Techniques like regression analysis, decision trees, and neural networks are employed to forecast customer churn, conversion rates, and sales performance [16]. AI-powered recommendation systems, including collaborative filtering and content-based filtering, are widely utilized by companies like Amazon and Netflix to personalize customer experiences [17].

A data-driven analytics approach is proposed for customer segmentation using store visit data derived from overall sales records. Additionally, a feature selection method is introduced, utilizing product taxonomy as input to categorize customers effectively [18].

Data preprocessing is a critical and time-intensive step before segmentation, especially when applying K-means clustering. This process includes outlier removal, data scaling, and handling long-tail distributions through data transformation [19].

2.2 RFM model

The RFM model was first introduced by Hughes [20] as a customer segmentation technique based on three key behavioral metrics including. Recency(R) How recently a customer made a purchase. Frequency(F) How often does a customer make purchases. Monetary(M)How much money a customer has spent.
Customers who score high on all three dimensions are often the most valuable and engaged for businesses. Several studies have validated the RFM model as a strong predictor of customer lifetime value (CLV) [21].

The RFM model is one of the most widely used methods for behavioral segmentation. This model segments existing customers based on their recency, frequency, and monetary. By prioritizing the identification of high-value customers for targeted marketing rather than acquiring new ones, this approach enhances customer retention strategies. Recent measures the number of days since the last purchase, frequency represents the total number of purchases, and monetary indicates the total purchase value within a specific timeframe. [22]

The RFM model is widely applied in customer segmentation to categorize customers into different groups such as Loyal Customers (high RFM scores), Potential Loyalists (moderate RFM scores), Churned Customers (low recency, low frequency). Chen et al. [23] showed that using RFM model-based segmentation increases customer retention rates by 25% compared to generic segmentation strategies.

Businesses use RFM analysis to predict customer churn by identifying users with low recency and frequency scores. According to Xu and Li [24], integrating RFM with machine learning models improves churn prediction accuracy by 15–20% compared to using RFM alone.

2.3 Machine learning approaches for marketing

Machine learning in marketing utilizes algorithms to analyze customer data, social media interactions, transaction histories, and market trends to uncover valuable insights. ML techniques help in identifying patterns, predicting future trends, and automating marketing tasks. According to Kotler et al. [25], businesses using AI and ML for marketing outperform traditional marketing strategies in customer retention and revenue growth.

Predictive analytics uses machine learning models to forecast customer actions, such as purchase likelihood, churn prediction, and sales forecasting. According to a study by Kim et al. [26], ML-driven predictive models improve customer retention by up to 40% when compared to heuristic-based predictions.

Recommendation engines personalize marketing efforts by suggesting products, services, or content based on user preferences. Research by Zhang et al. [27] found that hybrid recommendation systems improve customer engagement by 25% compared to single model approaches.

K-means is widely used across various applications due to its ease of understanding, interpretation, and implementation. For example, Chen et al. utilized K-means clustering alongside decision tree induction to segment customers based on an online retail dataset of customer transactions from the UCI repository. [28] Similarly, many studies have utilized the K-means algorithm for dataset segmentation.

Christy conducted a study using RFM analysis on transactional data, followed by clustering with traditional K-means and Fuzzy C-means algorithms. The research introduced a novel method for selecting initial centroids in K-means. The evaluation of these methodologies was based on iterations, cluster compactness, and execution time [29].

2.4 Clustering algorithm

K-Means is one of the most popular clustering algorithms, introduced by MacQueen in 1967 [30]. It works by Assigning data points to K clusters based on centroid initialization then Iteratively updating centroids to minimize within-cluster variance. Applications for this algorithm are for Customer segmentation used in marketing analytics to group customers based on behavior. Image compression used to Applied in computer vision for color quantization. Document clustering for Organizes large text corpora into thematic groups. But there are limitations for this algorithm such as Sensitive to initialization of centroids, require predefining K which is challenging for unknown datasets, and it struggles with non-spherical clusters and outliers. Several enhancements, such as K-Means++ (improves centroid selection) and Elbow Method (optimizes K selection), address these limitations.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a robust method that is introduced by Ester [31]. It groups dense regions while identifying outliers as noise. The advantage of this model is such as to Detects clusters of arbitrary shapes, unlike K-Means, no need to predefine K and can also Handles noise and outliers effectively.an applications are for Anomaly detection that used in cybersecurity for detecting network intrusions. Geospatial clustering for Groups locations based on density (e.g., hotspot mapping) Healthcare analytics are also used to identify patient clusters for personalized treatments. But there are some limitations for this algorithm that you will find in this model are Sensitive to hyperparameters (eps, min samples) require domain expertise. Struggles with varying density regions in a dataset. An improved version, Ordering Points to Identify Clustering Structure, extends DBSCAN to handle varying densities dynamically.

The Gaussian Mixture Model (GMM) is a probabilistic clustering method widely used in machine learning, particularly in scenarios where clusters exhibit overlapping boundaries and varying shapes. Unlike K-Means, which assigns each data point to a single cluster, GMM provides soft clustering, where each point has a probability of belonging to multiple clusters. It models data as a mixture of multiple Gaussian distributions, allowing for greater flexibility in capturing complex data structures [32]. GMM has been successfully applied in various domains, including image segmentation, speech recognition, customer segmentation, and fraud detection in financial transactions. Despite 9 its advantages, GMM is computationally expensive and sensitive to initialization, making it less efficient for large-scale datasets.

Hierarchical clustering is a tree-based clustering technique that organizes data into a nested hierarchy of clusters, making it useful for exploratory data analysis and cluster relationships visualization. Unlike partition-based clustering methods such as K-Means, hierarchical clustering does not require specifying the number of clusters in advance and is divided into two main approaches: agglomerative (bottom-up) and divisive (top-down) [33]. Hierarchical clustering has been widely used in various domains, including biological taxonomy and phylogenetics (identifying genetic relationships), document clustering (grouping articles or research papers), and market segmentation (identifying customer groups based on purchasing behavior). Compared to K-Means and Gaussian Mixture Models (GMM), hierarchical clustering provides better interpretability but is computationally expensive, making it unsuitable for large datasets [34]

3 Data and Methodology

3.1 Data

This research uses dataset name Online Retail from website UCI Machine Learning Repository. Transnational data set which contains all the transactions occurring between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retail. Due to its large volume and widespread use in developing algorithms for retail customer segmentation, this dataset has 541,909 entries and includes eight distinct variables including

Invoice No: A unique identifier for each transaction.
Stock Code: A unique product code assigned to each item.
Description: The name or description of the product.
Quantity: The number of units of the product purchased.
Invoice Date: The timestamp when the transaction occurred.
Unit Price: The price of a single unit of the product.
Customer ID: A unique identifier for each customer.
Country: The country where the customer is located.

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	United Kingdom
536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850.0	United Kingdom
536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
536365	22752	SET 7 BABUSHKA NESTING BOXES	2	12/1/2010 8:26	7.65	17850.0	United Kingdom
536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	12/1/2010 8:26	4.25	17850.0	United Kingdom
536366	22633	HAND WARMER UNION JACK	6	12/1/2010 8:28	1.85	17850.0	United Kingdom
536366	22632	HAND WARMER RED POLKA DOT	6	12/1/2010 8:28	1.85	17850.0	United Kingdom
536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	12/1/2010 8:34	1.69	13047.0	United Kingdom

Fig 1. Original Dataset

3.2 Framework of Proposed Methodology

Python is a versatile and powerful tool for advanced customer segmentation, providing an extensive library ecosystem and an intuitive syntax. Its capabilities enable the extraction of meaningful insights from complex datasets. This research utilized various Python libraries to streamline the analysis process.

This research leveraged Pandas for statistical analysis and NumPy for numerical computations. Matplotlib and Seaborn were employed for data visualization, with Seaborn enhancing graphical representation. Scikit-Learn played a crucial role in implementing machine learning algorithms and evaluating performance metrics.

Additionally, specialized tools and modules were imported, including RandomizedSearchCV, LabelEncoder, MinMaxScaler, StandardScaler, PCA, KMeans, GausianMixture, Silhouette_score, DBSCAN, and make_blobs, along with various evaluation metrics. The warnings module was also utilized to enhance code efficiency.

The experimental setup is illustrated in Figure 2 below.

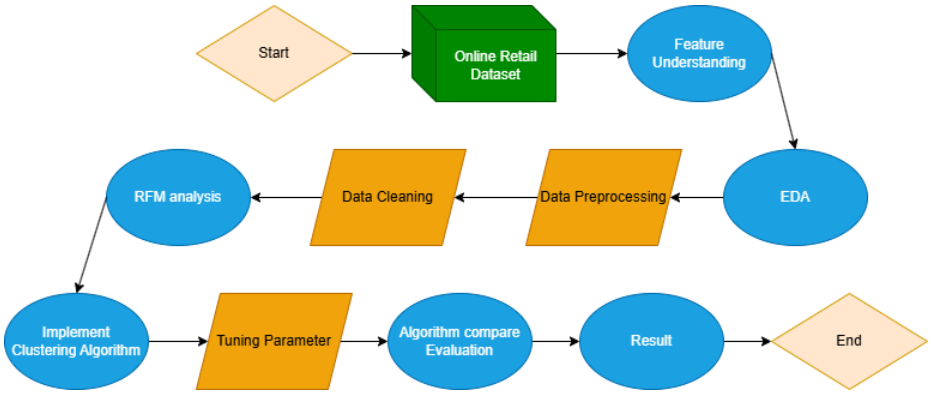

Fig 2. Framework of Proposed Methodology

3.3 Data preprocessing

Data preprocessing is a critical step in machine learning and clustering analysis, ensuring that raw data is transformed into a structured and meaningful format before applying clustering algorithms. This process involves several stages, including data cleaning, data transformation, feature scaling, and feature selection, each of which plays a crucial role in improving model performance and accuracy.

The first step, data cleaning, addressing missing values, duplicate entries, and outliers to ensure data integrity. In this study, missing values in attributes such as CustomerID and Description were handled using listwise deletion, where rows with missing values were removed to prevent bias in clustering models. Duplicate records were identified and removed to avoid redundancy and model distortion. Outliers, which can significantly impact distance-based clustering algorithms like K-Means, were detected using Z-score analysis and the Interquartile Range (IQR) method, with extreme values either removed or transformed to minimize their effect.

Following data cleaning, data transformation was applied to make categorical and time-based data suitable for clustering. Categorical variables, such as country names, were encoded using Label Encoding and One-Hot Encoding to convert them into numerical values. Additionally, date-time attributes such as InvoiceDate were transformed into numerical features, including recency, day-of-week, and time-based

segmentation, to enhance the effectiveness of clustering algorithms in customer segmentation tasks

To ensure uniformity in feature scales, feature scaling was applied, as clustering algorithms are sensitive to differences in magnitude. Standardization (Z-score normalization) was used to transform features into standard distribution by subtracting the mean and dividing by the standard deviation. This transformation ensures that all features contribute equally to distance-based clustering techniques. Additionally, MinMax Scaling was implemented to scale values between 0 and 1, particularly benefiting hierarchical clustering and Gaussian Mixture Models (GMM), which rely on probability-based distance metrics.

Finally, feature selection was performed to enhance computational efficiency and model interpretability by eliminating redundant or irrelevant features. Highly correlated features were detected using Pearson correlation analysis and removed to prevent redundancy in clustering results. Additionally, Principal Component Analysis (PCA) was employed to 13 reduce the dimensionality of the dataset while preserving the maximum variance, ensuring that only the most informative features were retained for clustering

3.4 Silhouette score

The Silhouette Score is a metric used to evaluate the quality of clusters in a clustering algorithm. It measures how well each data point fits within its assigned cluster compared to the nearest other cluster [35]. The score ranges from -1 to 1, where:
+1: Data points are well-clustered (clear separation between clusters).
0: Data points overlap between clusters (ambiguous clustering).
-1: Data points are misclassified (closer to another cluster than their assigned cluster).
Mathematically, the Silhouette Score for a single data point iii is defined as:

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (1)$$

where:

$a(i)$ = Average intra-cluster distance (how close the point is to other points in its cluster).

$b(i)$ = Average nearest-cluster distance (how far the point is from the closest other cluster).

The Silhouette Score for the entire clustering solution is the means of all individual scores.

3.5 RFM model

The Recency, Frequency, Monetary (RFM) model is an effective approach for customer segmentation, based on three key factors: recency (last purchase date), frequency (number of transactions), and monetary value (total spending) [36]. A combined **RFM Score** is computed by concatenating individual scores:

$$RFM\_Score = R\_Score \times 100 + F\_Score \times 10 + M\_Score \tag{2}$$

This model enables businesses to identify distinct customer segments and tailor marketing strategies accordingly. The RFM model helps businesses identify distinct customer segments and develop targeted marketing strategies. Its simplicity and efficiency allow companies to optimize resource allocation, prioritizing high-value customers and fostering long-term loyalty. To enhance segmentation and extract deeper insights, the RFM model can be expanded with additional dimensions or integrated with complementary methods such as clustering algorithms. This approach has gained significant attention for its role in shaping specialized marketing strategies.
Figure 3 below provides a simplified illustration of the RFM model.

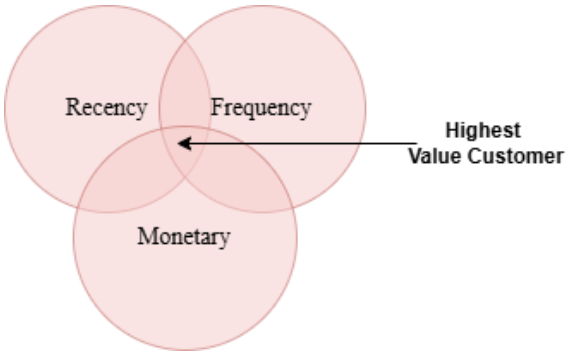

Fig 3. A simplified illustration of the RFM model.

3.6 Clustering algorithm

Cluster analysis is an iterative process that segments data into clusters based on similar characteristics. Clustering techniques are broadly categorized into partition-based, hierarchical-based, density-based, and distribution-based methods.

Partition-based clustering: (e.g., K-Means) divides data into K distinct groups, where K must be predefined.

Hierarchical clustering: builds clusters progressively, either bottom-up (agglomerative) by merging similar data points or top-down (divisive) by splitting them. Unlike partition-based methods, hierarchical clustering does not require a predefined number of clusters.

Density-based clustering: (e.g., DBSCAN) groups data points based on their spatial density, making it effective for detecting clusters of irregular shapes while being less sensitive to noise.

Distribution-based clustering: (e.g., Gaussian Mixture Model - GMM) assumes that data is generated from a combination of probability distributions. This method is particularly useful when clusters have overlapping boundaries and follow a specific statistical distribution.

3.7 Centroid-based Clustering

Centroid-based clustering is a widely used partitioning approach in which data points are grouped based on their proximity to a central representative, known as the centroid. The most well-known algorithm in this category is K-Means, where each cluster is represented by the meaning of its data points. The clustering process begins by initializing K centroids, assigning data points to the nearest centroid, and iteratively updating the centroids until convergence. The objective function minimizes the sum of squared distances between data points and their respective centroids, ensuring that similar data points are grouped together [37].

One of the main advantages of centroid-based clustering is its efficiency and scalability, making it suitable for large datasets. However, it assumes that clusters are spherical and evenly distributed, which may not always be the case in real-world data. Additionally, centroid-based methods are sensitive to outliers, as extreme values can significantly shift the cluster centers [38]. To address these limitations, variations such as K-Medoids (which selects actual data points as cluster centers) and fuzzy c-means clustering (which allows data points to belong to multiple clusters with different degrees of membership) have been developed [39].

Centroid-based clustering is widely applied in fields such as customer segmentation, image processing, and anomaly detection, where defining clear, distinct clusters is essential. Despite its limitations, it remains one of the most fundamental clustering techniques due to its simplicity and computational efficiency [40].

3.7.1 K-Means Clustering

K-Means is a partition-based clustering algorithm that groups data points into K distinct clusters based on their similarities. It is an unsupervised machine learning algorithm widely used for customer segmentation, image processing, anomaly detection, and various other applications.

In K-Means clustering, the algorithm updates the centroid of each cluster by computing the mean of all data points assigned to that cluster. The mathematical formulation is as follows:

$$C_j = \frac{1}{N_j} \sum_{i \in Cluster_j} X_i \tag{3}$$

Cj is the new centroid of cluster j.

Nj is the number of points in cluster j.

Xi represents each data point in the cluster.

Objective Function (Cost Function)

K-Means aims to minimize the sum of squared distances (SSD) between each data point and its assigned cluster centroid. The cost function (also known as the Within-Cluster Sum of Squares (WCSS)) is given by:

$$J = \sum_{j=1}^{K} \sum_{i \in Cluster_j} ||X_i - C_j||^2 \tag{4}$$

Where:

J is the total within-cluster variance (WCSS).

K is the total number of clusters.

Xi is a data point.

Cj is the centroid of cluster j.

$\left||Xi - Cj|\right|^2$ represents the squared Euclidean distance between a data point and its cluster centroid.

The algorithm minimizes this function by iteratively reassigning points and updating centroids until convergence is achieved.

3.7.2 K-Medoids Clustering

K-Medoids is a partition-based clustering algorithm that minimizes the total dissimilarity between data points and their respective cluster centers, called medoids. Unlike K-Means, which selects centroids based on the average of data points, K-Medoids selects actual data points as cluster representatives, making it more robust to outliers and more suitable for datasets where using means is not ideal.

The objective of **K-Medoids** is to minimize the total dissimilarity within clusters. The cost function is given by:

$$J' = \sum_{i=1}^{N} d(X_i, X_s) \tag{5}$$

If $J' < J$, the swap is accepted, and the new point Xs becomes the medoid.

3.8 Density-based Clustering

Density-based clustering is an unsupervised machine learning technique that groups data points based on regions of high density, effectively identifying clusters of arbitrary shape. Unlike centroid-based clustering methods like K-Means, which assume clusters are spherical, density-based methods identify clusters by detecting dense regions separated by sparser areas [41] The most well-known algorithm in this category is DBSCAN (Density-Based Spatial Clustering of Applications with Noise), which groups points that have a minimum number of neighboring points within a specified radius. Points that do not meet this density criterion are classified as noise or outliers.

A key advantage of density-based clustering is its ability to discover clusters of irregular shapes while being robust to noise and outliers. Unlike K-Means, it does not require the number of clusters (K) to be predefined, making it particularly useful for datasets where the cluster structure is unknown. However, DBSCAN's performance is sensitive to the choice of parameters, such as epsilon (ε), which defines the neighborhood radius, and minPts, which sets the minimum number of points required to form a dense region [42]. Additionally, density-based methods struggle with varying cluster densities and high-dimensional data, where defining a meaningful distance metric becomes challenging.

Despite these limitations, density-based clustering is widely applied in anomaly detection, spatial data analysis, and image segmentation, where natural clusters of varying shapes and sizes exist. Extensions of DBSCAN, such as OPTICS (Ordering Points to Identify the Clustering Structure), have been developed to handle datasets with varying

density more effectively [43]. Given its ability to detect noise and complex cluster structures, density-based clustering remains an essential tool for exploration data analysis and real-world applications.

3.8.1 DBSCAN

DBSCAN is a widely used density-based clustering algorithm that groups data points based on their density distribution rather than predefined cluster shapes. Unlike K-Means, which assumes clusters are spherical and require a predefined number of clusters, DBSCAN identifies clusters of arbitrary shapes and automatically detects outliers. It is particularly effective for datasets with varying cluster sizes and noise.

DBSCAN is particularly useful in scenarios where clusters are not well-separated, as it identifies groups based on their density distribution rather than distances to centroids. It is widely applied in anomaly detection, spatial data analysis, and customer segmentation, among other domains.

DBSCAN uses two key parameters to determine cluster structures:

1. ε (Epsilon) – The neighborhood radius defining the maximum distance between two points to be considered neighbors.
2. minPts (Minimum Points) – The minimum number of points required within an ε-radius to form a dense region.

Density Reachability

A point p is directly density-reachable from point q if:

$$d(p, q) \leq \epsilon \tag{6}$$

and q is a core point, meaning it has at least minPts points within its ε-radius.

Density Connectivity

A point p is density-connected to a point q if there exists a chain of points $p1, p2, \ldots, pn$ such that:

$$d(p_i, p_{i+1}) \leq \epsilon \tag{7}$$

for all i, and all points in the sequence are core points.

DBSCAN Clustering Objective Function
The DBSCAN algorithm aims to maximize the number of density connected points while minimizing the number of noise points:

$$J = \sum_{i=1}^{N} d(X_i, C(X_i)) \quad \text{where } d(X_i, C(X_i)) \leq \epsilon \tag{8}$$

Where:
J is the clustering objective.
N is the total number of data points.
Xi represents each data point.
$C(Xi)$ is the assigned cluster for XiX_iXi.
$d(Xi, C(Xi))$ is the distance function (e.g., Euclidean, Manhattan).
The algorithm stops when no new core points are found.

3.8.2 OPTICS

OPTICS (Ordering Points to Identify the Clustering Structure) is an extension of DBSCAN designed to address its limitations, particularly in datasets with varying cluster densities [44]. Unlike DBSCAN, which requires a fixed neighborhood radius (ε), OPTICS dynamically adapts the clustering process to identify clusters of different densities by sorting data points based on their reachability distances. This makes it more flexible and effective for real-world datasets where clusters may have different shapes and densities.

OPTICS follows a similar density-based clustering approach as DBSCAN but introduces a new concept called reachability distance, which helps in detecting hierarchical structures in data. The algorithm consists of three main steps

1.Compute Core Distances: The core distance of a point p is the minimum ε-distance required to include at least minPts neighbors.

$$\text{CoreDist}(p) = \min_{\text{minPts}} (d(p, q)) \tag{9}$$

2. Calculate Reachability Distances: The reachability distance is computed for each point to determine how easily it can be reached from a core point.

$$\text{ReachDist}(p, q) = \max(\text{CoreDist}(q), d(p, q)) \tag{10}$$

3. Order Points Based on Reachability Distances: Points are processed in increasing order of their reachability distances, forming a density-based hierarchical ordering of clusters.

3.9 Distribution-based clustering

Distribution-based clustering is a clustering approach that assumes data points are generated from underlying probability distributions. Unlike centroid-based clustering (e.g., K-Means) or density-based clustering (e.g., DBSCAN), distribution-based clustering models clusters as statistical distributions and assign data points based on the likelihood of belonging to a specific distribution [44]. The most common method in this category is the Gaussian Mixture Model (GMM), which assumes that data points are drawn from a mixture of several Gaussian (Normal) distributions with unknown parameters.

3.9.1 Gaussian Mixture Model (GMM)

The Gaussian Mixture Model (GMM) is a powerful approach for uncovering intricate patterns in consumer datasets, making it highly effective for customer segmentation. By modeling the probabilistic distribution of data points, GMM identifies hidden structures and relationships within customer behaviors. Unlike traditional clustering algorithms, GMM assumes that data points within a cluster follow a Gaussian distribution, allowing it to detect complex patterns that might otherwise go unnoticed. Its flexibility enables it to identify clusters of varying shapes, sizes, and densities, making it well-suited for sophisticated segmentation tasks. For a dataset with N data points and K clusters, GMM facilitates the discovery of nuanced customer categories based on behaviors, preferences, and interactions. This empowers businesses to develop highly targeted marketing strategies tailored to each segment's specific needs.

In Gaussian Mixture Models (GMM), the probability that a data point belongs to cluster is given by the Gaussian (Normal) probability density function (PDF):

$$P(x_i|\theta_j) = \frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}} \exp\left(-\frac{1}{2}(x_i - \mu_j)^T \Sigma_j^{-1}(x_i - \mu_j)\right) \tag{11}$$

where:
xi is the data point.
μj is the mean vector of cluster j.
Σj is the covariance matrix of cluster j, which defines the cluster shape.
d is the number of dimensions.
The total probability of a data point belonging to any cluster is computed as a weighted sum of individual Gaussian distributions:

$$P(x_i) = \sum_{j=1}^{K} \pi_j P(x_i|\theta_j) \tag{12}$$

where:
πj is the mixing coefficient (the weight of each cluster).
K is the number of clusters.

Parameter Estimation Using Expectation-Maximization (EM) Algorithm
Since the parameters $\boldsymbol{\mu j}, \boldsymbol{\Sigma j}, \boldsymbol{\pi j}$ are unknown, they are estimated using the Expectation-Maximization (EM) algorithm, which consists of two steps:
1.Expectation (E-Step): Compute the probability that each data point belongs to each cluster using the current parameter estimates.
2.Maximization (M-Step): Update the parameters (means, covariances, and weights) to maximize the likelihood of the observed data.
This process repeats iteratively until convergence is achieved.

3.10 Hierarchical Clustering

Hierarchical clustering is a tree-based clustering algorithm that recursively splits or merges data points to form a hierarchy of clusters. Unlike K-Means or DBSCAN, hierarchical clustering does not require the number of clusters (K) to be predefined. Instead, it produces a dendrogram (a tree-like structure) that allows analysts to choose the number of clusters based on data structure [45].

The way distances between clusters are measured affects the clustering outcome. Common linkage methods include:

1.Single Linkage (Minimum Distance)

The distance between two clusters is defined by the shortest distance between points in each cluster.

It can create long chain-like clusters.

$$d(A,B) = \min_{i \in A, j \in B} d(x_i, x_j) \tag{13}$$

2. Complete Linkage (Maximum Distance)
The distance between two clusters is the farthest pair of points between the clusters.
Creates compact, spherical clusters.

$$d(A,B) = \max_{i \in A, j \in B} d(x_i, x_j) \tag{14}$$

3. Average Linkage (Mean Distance)
Use the average distance between all pairs of points in the two clusters.
Balances between Single and Complete Linkage.

$$d(A,B) = \frac{1}{|A||B|} \sum_{i \in A} \sum_{j \in B} d(x_i, x_j) \tag{15}$$

4. Centroid Linkage
The distance is measured between centroids (mean points) of each cluster.
It can result in distorted cluster shapes if clusters have different densities.

$$d(A,B) = ||c_A - c_B|| \tag{16}$$

4 Evaluation and Result

4.1 Data Preprocessing

The original dataset consists of eight features. However, two features contain missing values: Description have 540,455 entries, and CustomerID, have 406,829 entries from overall 541,909 entries. When expressed as a percentage, CustomerID accounts for 24.93% of the missing values, while Description accounts for 0.27%. To enhance the model's performance in the subsequent calculation steps, these null values are removed from the dataset. Additionally, duplicate data can impact the analysis, making it essential to remove them. After eliminating both missing and duplicate entries, the final dataset consists of 401,604 records. Since the dataset consists of real transaction records, it may include details such as product cancellations and free gifts, which can be identified through the InvoiceNo feature. There are 2% product cancellations of the total dataset. Retaining this information allows for deeper data analysis and the

discovery of valuable insights. The dataset is organized chronologically by transaction date and time. To effectively manage the data, it is essential to identify unique CustomerID entries and aggregate each customer's transactions over the entire year. A sample data representation of the features is shown in Figure 4 below.

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	United Kingdom
536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850.0	United Kingdom
536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
536365	22752	SET 7 BABUSHKA NESTING BOXES	2	12/1/2010 8:26	7.65	17850.0	United Kingdom
536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	12/1/2010 8:26	4.25	17850.0	United Kingdom
536366	22633	HAND WARMER UNION JACK	6	12/1/2010 8:28	1.85	17850.0	United Kingdom
536366	22632	HAND WARMER RED POLKA DOT	6	12/1/2010 8:28	1.85	17850.0	United Kingdom
536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	12/1/2010 8:34	1.69	13047.0	United Kingdom

Fig 4. Original Dataset Example

4.2 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a crucial step in any data-driven project as it helps uncover patterns, detect anomalies, and ensure data quality before applying machine learning models. It allows us to identify missing values, outliers, and inconsistencies that could impact analysis. EDA also provides insights into data distributions, relationships between variables, and feature importance, guiding data preprocessing and model selection.

From this transaction dataset, EDA can be conducted to identify popular products and analyze buyer locations, helping to uncover initial purchasing patterns and relationships. This figure 5 displays the top 30 most frequently sold products in the store in the past year.

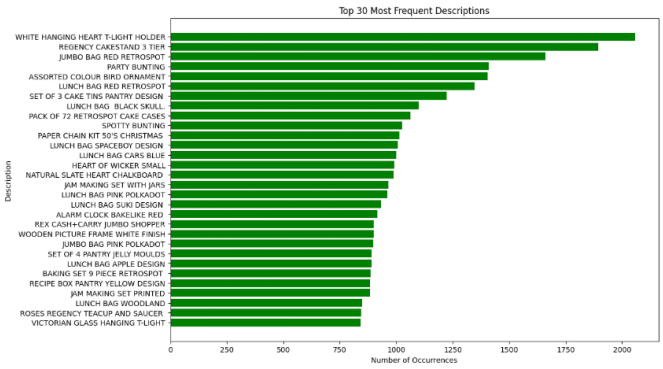

Fig 5. Top 30 most Frequent Descriptions

Next, we will determine the number of unique stock codes and visualize the top 10 most frequently occurring stock codes along with their percentage frequency, as illustrated in Figure 6

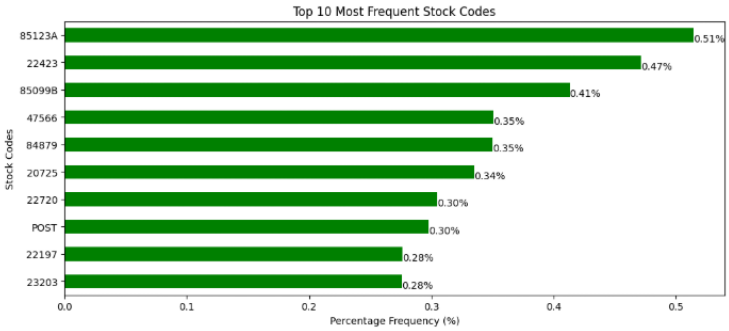

Fig 6. Top 10 most frequent stock codes

Most of the unique stock codes (3676 out of 3684) contain exactly 5 numeric characters, which seems to be the standard format for representing product codes in this dataset.

There are a few anomalies: 7 stock codes contain no numeric characters, and 1 stock code contains only 1 numeric character. These clearly deviate from the standard format and need further investigation to understand their nature and whether they represent valid product transactions.

From EDA the percentage of records with anomalous stock codes in the dataset is: 0.48%. We find that a very small proportion of the records, 0.48%, have anomalous stock codes, which deviate from the typical format observed in the majority of the data. Also, these anomalous codes are just a fraction among all unique stock codes (only 8 out of 3684).

4.3 RFM Feature Engineering

After completing data preparation through preprocessing and cleaning, the next step is applying Feature Engineering to extract meaningful insights from the data.

Feature Engineering for RFM Analysis involves transforming raw transaction data into three key metrics—Recency, Frequency, and Monetary (RFM)—to effectively segment customers:

Recency (R): Measures of how recently a customer made a purchase. It is calculated as the number of days since the last transaction.

Frequency (F): Represents how often a customer makes a purchase within a given period. It derives from counting the total number of transactions per customer.

Monetary (M): Indicates the total revenue a customer has generated. It is calculated by summing up the total spending of each customer.

These features are then used to segment customers based on purchasing behavior, enabling target marketing and customer retention strategies.

Finally, the basic characteristics of the dataset will be as follows in Figure 7:

CustomerID	Days_Since_Last_Purchase	Total_Transactions	Total_Products_Purchased	Total_Spend	Average_Transaction_Value
12346.0	325	2	0	0.00	0.000000
12347.0	2	7	2458	4310.00	615.714286
12348.0	75	4	2332	1437.24	359.310000
12349.0	18	1	630	1457.55	1457.550000
12350.0	310	1	196	294.40	294.400000

Fig 7. basic characteristics of the dataset

To enhance customer segmentation analysis, we created the RFM Segment column based on the calculated Recency (R), Frequency (F), and Monetary (M) scores. By combining these three key behavioral indicators into a single labeled segment, such as Champions, Big Spenders, At Risk, or Lost, we can translate complex numerical RFM scores into intuitive customer categories. This segmentation provides a clearer and more actionable understanding of customer value and engagement levels, allowing businesses to tailor marketing strategies more effectively to different customer groups.

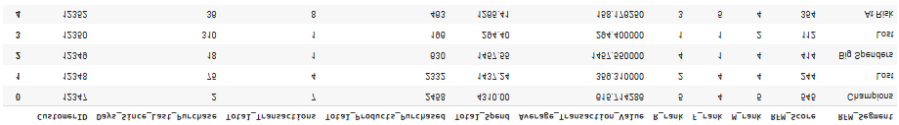

Fig 8. basic characteristics of the dataset and RFM_Segment

The RFM Segment column was created by applying a rule-based classification using the Recency (R), Frequency (F), and Monetary (M) scores. Specifically, customers were assigned to segments according to the following logic: if a customer had high scores in all three dimensions ($R \geq 4$, $F \geq 4$, and $M \geq 4$), they were labeled as Champions, representing highly valuable and engaged customers. If a customer had high Recency and Frequency scores but not necessarily high Monetary value, they were classified as Loyal Customers. Customers with high Recency and Monetary scores but not Frequency were grouped as Big Spenders. Meanwhile, customers with a low Recency score ($R \leq 2$), indicating a long time since their last purchase, were categorized as Lost. All other customers who did not meet the above conditions were labeled At Risk,

signaling declining engagement. This rule-based segmentation helps to translate raw RFM scores into actionable customer groups for targeted business strategies.

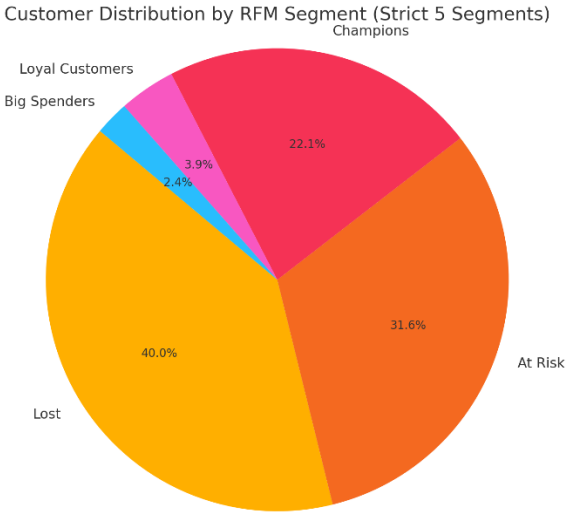

Fig 9 Customer Distribution by RFM Segment

Besides RFM (Recency, Frequency, Monetary) analysis, other feature engineering techniques for customer segmentation include customer lifetime value (CLV), purchase trends, seasonal buying behavior, average order value, and churn prediction features. These features provide deeper insights into customer behavior, helping businesses personalize marketing strategies, improve retention, and maximize revenue. Feature engineering is essential because raw data often lacks structure and meaningful patterns. By transforming data into relevant features, we enhance model performance, ensure better segmentation accuracy, and drive more informed decision-making.

In addition to the RFM (Recency, Frequency, Monetary) model, we selected Customer Diversity, Average Days Between Purchases, and Shopping Hour because these features add behavioral and temporal context that the traditional RFM model alone may miss. These dimensions enrich the segmentation by capturing customer variability in time patterns, purchase intervals, and behavior diversity.

Customer Diversity feature reflects how varied a customer's purchases are across different product categories or departments. According to Ngai [46], incorporating behavioral variety can improve targeting and retention strategies by distinguishing between generalist and specialist buyers. Customers with high diversity may respond better to broad campaigns, while low-diversity buyers may need tailored promotions.

Average Days Between Purchases captures the temporal regularity of purchases. While Frequency tells how often, this feature tells how consistently a customer returns.

It is useful for identifying habitual vs. sporadic customers. According to Parvaneh [47], inter-purchase time is a strong signal for understanding lifecycle stage and loyalty potential.

Shopping Hour represents the purchase time behavior, helping to capture customer preferences in shopping patterns — e.g., daytime vs. nighttime buyers. This can be useful for marketing timing and personalization. Liu & Shih [48] suggest that temporal data such as transaction times can provide critical insights into customer habits that go beyond monetary value alone.

We incorporated 3-4 additional feature engineering elements, including Customer Diversity, Average Days Between Purchases, and Shopping Hour. As a result, the final dataset is structured as shown in figure 10 below.

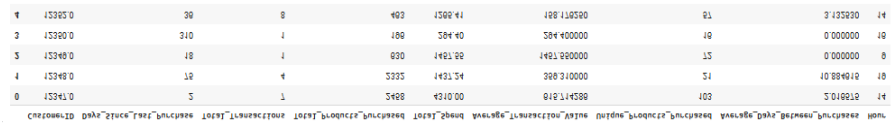

Fig 10. The final dataset

Table 4.1 Data information

Column	Count	Dtype
CustomerID	4274	float64
Days_Since_Last_Purchase	4274	int64
Total_Transactions	4274	int64
Total_Products_Purchased	4274	int64
Total_Spend	4274	float64
Average_Transaction_Value	4274	float64
Unique_Products_Purchased	4274	int64
Average_Days_Between_Purchases	4274	float64
Hour	4274	int32

4.4 Correlation analysis and Feature Scaling

Correlation analysis helps identify relationships between variables, ensuring that redundant or highly correlated features do not negatively impact model performance. It also aids in feature selection by highlighting the most relevant attributes for segmentation. Feature scaling is essential because clustering algorithms like K-Means and GMM

are sensitive to differences in magnitude. Scaling techniques, such as Standardization or Normalization, ensure that all features contribute equally, preventing dominance by variables with larger numerical ranges and improving clustering accuracy.

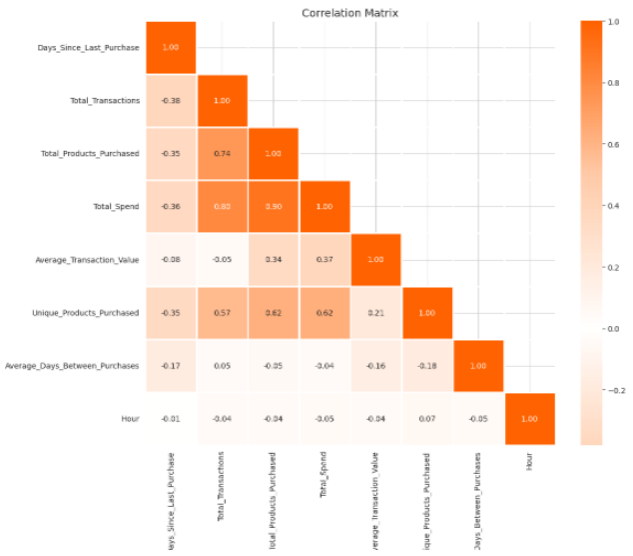

Fig 11. Dataset Correlation

Now we use Standard Scaler to ensure that all features have a mean of 0 and a standard deviation of 1, making them comparable in scale. This is crucial for clustering algorithms like K-Means, GMM, and Agglomerative Clustering, which rely on distance measurements. Without scaling, features with larger numerical ranges (e.g., Monetary value) could dominate those with smaller ranges (e.g., Recency). Standardization improves model performance, speeds up convergence, and prevents bias in clustering, leading to more accurate segmentation results. The result we got can be shown in figure 12 below.

	CustomerID	Days_Since_Last_Purchase	Total_Transactions	Total_Products_Purchased	Total_Spend	Average_Transaction_Value	Unique_Products_Purchased	Average_Days_Between_Purchases	Hour
0	12347.0	-0.899348	0.686733	2.158929	2.363366	1.503967	0.807036	-0.139896	0.645387
1	12348.0	-0.165834	-0.018529	2.006505	0.229966	0.326585	-0.592980	1.660533	2.829310
2	12349.0	-0.738578	-0.723791	-0.052432	0.245049	5.369588	0.277762	-0.549309	-1.538536
3	12350.0	2.195479	-0.723791	-0.577449	-0.618742	0.028526	-0.678347	-0.549309	1.518956
4	12352.0	-0.557711	0.921821	-0.254455	0.102360	-0.597000	0.021661	0.086670	0.645387

Fig 12. Standard Scaler Dataset

4.5 Model performance

Table 4.2 RFM + 3 Feature Silhouette Score Result

Clustering Algorithm	Silhouette Score	Execution Time (Second)
K-Means	0.37	4.32
K-Medoid	0.32	47.27
DBSCAN	0.63	6.78
GMM	0.22	7.23
Hierarchical	0.29	27.4

Table 4.3 Only RFM Silhouette Score Result

Clustering Algorithm	Silhouette Score	Execution Time (Second)
K-Means	0.91	4.56
K-Medoid	0.44	53.46
DBSCAN	0.75	2.12
GMM	0.46	8.73
Hierarchical	0.90	33.75

In conclusion, while the DBSCAN method achieves a relatively high silhouette score of 0.63, it is limited to only two clusters. When additional clusters are introduced, the silhouette score drops significantly.

When comparing K-Means and K-Medoids, both are widely used today because they allow for manual specification of the number of customer groups. In contrast, DBSCAN is advantageous for automatically identifying customer segments based on data density, making it efficient in cases where the optimal number of clusters is unknown. However, DBSCAN may generate more than 10 clusters, which may not always be necessary for certain businesses. As shown in the table, the clustering efficiency of these methods ranks second, with silhouette scores of 0.37 and 0.32.

For K-Means with 6 clusters, the silhouette score is 0.24, while for K-Medoids with 7 clusters, the silhouette score is 0.19. This is the most optimal value for segmenting customers into more than two groups.

The remaining methods, Hierarchical Clustering and GMM have the next lowest efficiencies, with silhouette scores of 0.29 and 0.22, respectively. When segmenting into more than two clusters, GMM achieves its highest silhouette score of 0.09 with 3 clusters, while Hierarchical Clustering reaches 0.2 with 6 clusters.

From the experimental results shown in Table 4.2 and Table 4.3, the clustering performance, as measured by the Silhouette Score, is significantly higher when using only the RFM features compared to using RFM plus the three additional features (Customer Diversity, Average Days Between Purchases, and Shopping Hour). For example, K-

Means clustering achieved a Silhouette Score of 0.91 with only RFM, while it dropped drastically to 0.37 when additional features were included. Similar trends were observed across all other algorithms.

This decline in clustering quality can primarily be attributed to the curse of dimensionality. When additional features are added, the feature space becomes higher-dimensional, causing the distance between data points to become less meaningful. As a result, the clusters become less well-defined, leading to lower Silhouette Scores. Moreover, if the newly added features are not highly informative or are weakly correlated with the original RFM dimensions, they can introduce noise, further degrading the quality of clustering.

Additionally, the added features may have different scales, distributions, or levels of importance compared to the RFM features. Even though the data was standardized, differences in the intrinsic structure of the features could still disrupt the natural grouping pattern that RFM features alone were able to capture effectively

5 Conclusion

This study aims to develop a customer segmentation model to improve decision-making processes in the retail industry. In this industry, handling large datasets is crucial, making AI-driven and machine learning techniques essential for business development in the modern era. While DBSCAN achieves the highest value, practical business considerations suggest that customers should be segmented into at least two groups but not an excessive number to maintain meaningful insights.

The most suitable method for this sample dataset is K-Means clustering, primarily because it allows for selecting the desired number of clusters. This flexibility is particularly useful when there is an initial segmentation of customers, such as high-, medium- and low-spending groups. The RFM model is a widely used approach for preliminary segmentation, and K-Means is both efficient and easy to implement in this context. However, caution is needed as this method is highly sensitive to outliers. If the data is not properly managed during preprocessing, it may significantly impact the accuracy of the analysis results.

In DBSCAN, the number of clusters cannot be predefined, making it effective for handling unstructured data. However, its main drawback is that it becomes highly resource-intensive when working with large datasets.in conclusion, If customer data is highly irregular, contains noise, or needs anomaly detection DBSCAN will be more effective.

Both Gaussian Mixture Model (GMM) and Hierarchical Clustering can be effective for customer segmentation, but their suitability depends on the dataset characteristics and business objectives. GMM remains a viable option as it allows for specifying the number of clusters, like K-Means, and is particularly useful when customer segments have significant overlap. However, its main drawbacks include being computationally

more expensive than K-Means and highly sensitive to initial values—an incorrect initialization can lead to poor clustering results.

The hierarchical clustering method is ideal when the exact number of clusters is unknown, as a dendrogram can help determine the optimal segmentation. This approach provides a clear visual representation of the clustering structure, making it useful for interpretation. However, it is best suited for small datasets, as it becomes computationally expensive with larger data. Its main drawbacks include long processing times, lack of flexibility in adjusting segmentation parameters, and sensitivity to outliers, which can distort the cluster structure. If a clear hierarchical view of customer groups is needed and the dataset is not too large, hierarchical clustering is a strong choice.

References

1. B. G. Fader, B. Hardie, and K. L. Lee, "RFM and CLV: Using iso-value curves for customer base analysis," Journal of Marketing Research, vol. 42, no. 4, pp. 415-430, Nov. 2005.
2. V. Kumar and W. Reinartz, Customer Relationship Management: Concept, Strategy, and Tools, 3rd ed. Cham, Switzerland: Springer, 2018.
3. M. Wedel and W. S. Kamakura, Market Segmentation: Conceptual and Methodological Foundations, 2nd ed. New York, NY, USA: Springer, 2000.
4. J. MacQueen, "Some methods for classification and analysis of multivariate observations," in Proc. 5th Berkeley Symp. Math. Stat. Probab., Berkeley, CA, USA, 1967, vol. 1, pp. 281-297.
5. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proc. 2nd Int. Conf. Knowl. Discov. Data Min. (KDD'96), Portland, OR, USA, 1996, pp. 226-231.
6. D. Reynolds, "Gaussian mixture models," Encyclopedia of Biometrics, vol. 741, pp. 659-663, 2009.
7. F. Murtagh and P. Legendre, "Ward's hierarchical agglomerative clustering method: Which algorithms implement Ward's criterion?" J. Classification, vol. 31, no. 3, pp. 274-295, Oct. 2014.
8. X. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," Ann. Data Sci., vol. 2, no. 2, pp. 165-193, Apr. 2015.
9. E. Celebi, H. A. Kingravi, and P. A. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm," Expert Syst. Appl., vol. 40, no. 1, pp. 200-210, Jan. 2013.
10. E. Schubert, J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "DBSCAN revisited: Why and how you should (still) use DBSCAN," ACM Trans. Database Syst., vol.42, no. 3, pp. 1-21, Jul. 2017.
11. C. M. Bishop, Pattern Recognition and Machine Learning, New York, NY, USA: Springer, 2006.
12. L. Rokach and O. Maimon, "Clustering methods," in Data Mining and Knowledge Discovery Handbook, Boston, MA, USA: Springer, 2005, pp. 321-352.
13. Y. Chen, B. Wang, and X. Zhang, "Customer segmentation using RFM analysis and clustering algorithms," Appl. Sci., vol. 11, no. 10, pp. 4567, 2021.
14. G. L. Urban, "Customer advocacy: A new era in marketing?" Journal of Public Policy & Marketing, vol. 24, no. 1, pp. 155-159, 2005.

15. J. W. Wirtz and M. K. Lwin, "Regulatory focus and customer retention," Journal of Consumer Psychology, vol. 19, no. 2, pp. 260-270, 2009.
16. P. Kotler, K. L. Keller, M. Brady, M. Goodman, and T. Hansen, Marketing Management, 3rd ed. Harlow, U.K.: Pearson Education, 2009.
17. A. Ghose, P. G. Ipeirotis, and B. Li, "Designing ranking systems for consumer reviews: The impact of review subjectivity on product sales and review quality," Management Science, vol. 60, no. 9, pp. 2176-2194, 2014.
18. Hu, Ya-Han, Yeh, Tzu-Wei, 2014. Discovering valuable frequent patterns based on RFM analysis without customer identification information. Knowl.-Based Syst. 61, 76–88.
19. Tavakoli, M., Molavi, M., Masoumi, V., Mobini, M., Etemad, S., & Rahmani, R. (2018). Customer segmentation and strategy development based on user behavior analysis, RFM model and data mining techniques: A case study. In 2018 IEEE 15th international conference on E-business engineering (pp. 119–126)
20. A. M. Hughes, Strategic Database Marketing: The Masterplan for Starting and Managing a Profitable, Customer-Based Marketing Program, 1st ed. McGraw-Hill, 1994.
21. P. S. Fader, B. G. Hardie, and K. L. Lee, "RFM and CLV: Using Iso-Value Curves for Customer Base Analysis," Journal of Marketing Research, vol. 42, no. 4, pp. 415–430, 2005.
22. Dedi, Dzulhaq, M. I., Sari, K. W., Ramdhan, S., Tullah, R., & Sutarman (2019). Customersegmentation based on RFM value using K-means algorithm. In 2019 fourth international conference on informatics and computing (pp. 1–7).
23. X. Chen, Y. Zhang, and Z. Wang, "Customer Retention Strategies Using RFM-Based Segmentation: A Case Study in E-Commerce," International Journal of Business Analytics, vol. 6, no. 3, pp. 21–37, 2019.
24. Y. Xu and H. Li, "Integrating RFM Analysis with Machine Learning for Customer Churn Prediction," IEEE Access, vol. 8, pp. 21559–21570, 2020.
25. P. Kotler, K. L. Keller, M. Brady, M. Goodman, and T. Hansen, Marketing Management, 3rd ed. Pearson Education, 2021.
26. J. Zhang et al., "Hybrid recommendation models in personalized marketing," Journal of Data Science and Marketing, vol. 15, no. 3, pp. 341-356, 2021.
27. Y. Lin and C. H. Wu, "Voice search optimization in digital marketing," Journal of Interactive Technology, vol. 8, no. 1, pp. 97-115, 2023.
28. Chen, D., Sain, S. L., & Guo, K. (2012). Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining. Journal of Database Marketing & Customer Strategy Management, 19(3), 197–208.
29. Christy, A., Joy, A., Umamakeswari, L. Priyatharsini, Neyaa, A., 2018. RFM ranking–an effective approach to customer segmentation. J. King Saud Univ.-Comput. Inf. Sci.
30. J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," in Proc. 5th Berkeley Symp. Math. Stat. Probab., 1967, pp. 281-297.
31. M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining (KDD), 1996, pp. 226-231.
32. P. Kotler, "Customer segmentation using Gaussian Mixture Models," Journal of Business Analytics, vol. 8, no. 2, pp. 124-135, 2020.
33. P. Jain and M. Law, "Data Clustering: A Review," ACM Computing Surveys, vol. 51, no. 3, pp. 1-39, 2018.
34. F. Murtagh and P. Legendre, "Ward's Hierarchical Clustering Algorithm: Implementation and Applications," Journal of Classification, vol. 31, no. 2, pp. 274-295, 2014.
35. P. J. Rousseeuw, "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis," Journal of Computational and Applied Mathematics, vol. 20, pp. 53–65, 1987.

36. Hossain, A.S. Customer segmentation using centroid based and density based clustering algorithms. In Proceedings of the 2017 3rd International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 7–9 December 2017; pp. 1–6.
37. J. MacQueen, "Some methods for classification and analysis of multivariate observations," Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, no. 281-297, pp. 14, 1967.
38. A. K. Jain, "Data clustering: 50 years beyond K-Means," Pattern Recognition Letters, vol. 31, no. 8, pp. 651-666, 2010.
39. J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," Computers & Geosciences, vol. 10, no. 2-3, pp. 191-203, 1984.
40. R. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," Annals of Data Science, vol. 2, no. 2, pp. 165-193, 2015.
41. M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD), vol. 96, pp. 226-231, 1996.
42. E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN," ACM Transactions on Database Systems (TODS), vol. 42, no. 3, pp. 1-21, 2017.
43. M. Ankerst, M. M. Breunig, H. P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," ACM SIGMOD Record, vol. 28, no. 2, pp. 49-60, 1999.
44. G. J. McLachlan and D. Peel, Finite Mixture Models, Wiley, 2000.
45. A. K. Jain and R. C. Dubes, Algorithms for Clustering Data, Prentice-Hall, 1988.
46. E. W. T. Ngai, L. Xiu and D. C. K. Chau, "Application of data mining techniques in customer relationship management: A literature review and classification," Expert Systems with Applications, vol. 36, no. 2, pp. 2592-2602, 2009. doi: 10.1016/j.eswa.2008.02.021
47. H. Parvaneh, M. Asadpour and A. Arjmand, "Customer segmentation using RFM and time between purchases for improving clustering accuracy," International Journal of Computer Applications, vol. 177, no. 19, pp. 1–6, 2019. doi: 10.5120/ijca2019919523
48. D.-R. Liu and Y.-Y. Shih, "Integrating AHP and data mining for product recommendation based on customer lifetime value," Information & Management, vol. 42, no. 3, pp. 387–400, 2005. doi: 10.1016/j.im.2004.01.008