

# **Comparative Analysis of Clustering Algorithms for Customer Segmentation and Improved Marketing Strategies**



**Student Name- Parag Vijay Vade**

**Student Number- 20002536**

Applied Research Project submitted in partial fulfilment of the requirements of  
Master of Science (MSc) in Business Analytics  
at Dublin Business School

**Supervised by – Dr Obinna Izima**

**Declaration**

'I declare that this Applied Research Project I have submitted to Dublin Business School for the award of MSc. Business Analytics is the result of my own investigations. Except where otherwise stated, where it is clearly acknowledged by references. Furthermore, this work has not been submitted for any other degree.'

Signed: Parag Vijay Vade

Student Number: 20002536

Date: 20th May 2024.

**Acknowledgements**

I want to express my heartfelt gratitude for the invaluable guidance I received from my supervisor, Dr. Obinna Izma. I am profoundly grateful for his constant support, which has been instrumental in making this endeavour possible and his willingness to give his time generously has been very much appreciated.

I would like to thank my Professors Assem Abdelakh, David Williams, Derek Reynolds, Kunwar Madan, Rudi O'Reilly Meehan and Yusuf Aytas for their significant contributions to my academic growth. The combined efforts of all the professors have played a pivotal role in shaping my abilities as I manoeuvred through the challenges of my MSc. in Business Analytics. Also, I feel immense gratitude towards my parents and my sister for their faith in me.

## Table of Contents

Abstract.....	6
Chapter 1. Introduction .....	7
1.1 Background and Significance of the Study.....	7
1.2 Research Problem.....	8
1.3 Research Aim and Objective.....	8
1.4 Dissertation Roadmap .....	9
Chapter 2. Literature Review .....	10
2.1 Significance of Clustering from Marketing Perspective.....	10
2.2 Understanding Customer Value using Recency, Frequency and Monetary parameters (RFM) .	10
2.3 K-Means .....	11
2.3.1 Functioning of K-means Clustering.....	11
2.3.2 K-Means Hyperparameters .....	12
2.3.3 Elbow Plot for optimal number of clusters .....	12
2.3.4 Applications in Customer Segmentation .....	13
2.3.5 Limitations of K-Means.....	14
2.4 DBSCAN.....	15
2.4.1 Functioning of DBSCAN Clustering .....	15
2.4.2 DBSCAN Hyperparameters.....	16
2.4.3 Applications in Customer Segmentation .....	16
2.4.4 Limitations of DBSCAN .....	17
2.5 OPTICS .....	17
2.5.1 Functioning of OPTICS Clustering.....	18
2.5.2 OPTICS Hyperparameters .....	19
2.5.4 Limitations of OPTICS.....	19
2.6 Dimensionality Reduction .....	20
2.6.1 Advantages of dimensionality reduction in clustering.....	20
2.6.2 Using UMAP over PCA.....	20
2.6.3 Functioning of UMAP .....	21
2.7 Comparative Analysis of K-Means, DBSCAN and OPTICS .....	22
Chapter 3. Research Methodology .....	25
3.1 Introduction to Methodology .....	25
3.2 Data Collection .....	27
3.3 Data Characteristics .....	27
3.3 Data Pre-processing .....	28

3.3.1 Data Cleaning.....	28
3.3.2 Feature Engineering.....	29
3.3.3 Removing Outliers .....	30
3.3.4 Data Normalization .....	33
3.4 Dimensionality Reduction .....	34
3.5 Evaluation Metrics .....	35
3.5.1 Silhouette Score .....	35
3.5.2 Davies-Bouldin Index .....	36
3.6 Clustering Algorithms .....	37
3.6.1 K-Means .....	38
3.6.2 DBSCAN.....	39
3.6.3 OPTICS .....	40
3.7 Software and Tools.....	41
Chapter 4. Results and Analysis .....	42
Chapter 5. Conclusion, Limitations and Future Work.....	47
5.1 Conclusion .....	47
5.2 Limitations and Future Work .....	48
References.....	49

## Index of Figures

Figure 1. Elbow Plot Illustration (Zhang et al., 2020).....	13
Figure 2. Reachability Plot (Gialampoukidis et al., 2016).....	18
Figure 3. Methodology Flow .....	26
Figure 4. Frequency Boxplot.....	31
Figure 5. Monetary Value Boxplot.....	31
Figure 6. Recency Boxplot.....	32
Figure 7. Scaled Features .....	33
Figure 8. UMAP Visualization .....	35
Figure 9. Elbow Plot .....	38
Figure 10. Silhouette Scores for Cluster Range.....	39
Figure 11. OPTICS Hyperparameter Tuning using GridSearch .....	41
Figure 12. K-Means Cluster Visualization .....	42
Figure 13. DBSCAN Cluster Visualization .....	44
Figure 14. OPTICS Cluster Visualization.....	45

## Index of Tables

Table 1. Data Description .....	27
Table 2. Clustering Result.....	45

## List of Acronyms

ARI- Adjusted Rand Index

DBI- Davies-Bouldin Index

DBSCAN- Density-Based Spatial Clustering of Applications with Noise

NMI- Normalized Mutual Information

OPTICS- Ordering Points to Identify the Clustering Structure

RFM- Recency, Frequency, Monetary

PCA- Principal Component Analysis

UMAP- Uniform Manifold Approximation and Projection

## Abstract

This study evaluates the effectiveness of K-Means, DBSCAN, and OPTICS clustering algorithms for customer segmentation. Using the Recency, Frequency, and Monetary (RFM) model, customer value was quantified, and customers were segment based on their transactional behavior.

Dataset was obtained from UCI machine learning repository and contained transaction details of an online retail business. The data underwent cleaning, feature engineering, normalization, and dimensionality reduction using UMAP. The clustering algorithms were then applied and evaluated using Silhouette Scores and Davies-Bouldin Indices.

K-Means effectively grouped customers, achieving a Silhouette Score of 0.445 and a Davies-Bouldin Index of 0.736. DBSCAN handled noise and identified arbitrary shapes but produced scattered clusters with a lower Silhouette Score of 0.132 and a higher Davies-Bouldin Index of 1.435. Although OPTICS had similar scores to DBSCAN, it resulted in smoother clusters and handled varying densities more effectively than DBSCAN.

To summarize, K-Means provided the best cluster separation. DBSCAN and OPTICS were better for noise handling and variable densities.

# Chapter 1. Introduction

## 1.1 Background and Significance of the Study

In recent years, the landscape of marketing has undergone a profound transformation propelled by advancements in information technology, communications, and data analytics. Businesses are increasingly embracing data-driven decision-making practices as the standard approach (Shah and Murthi, 2021). According to the Global Marketing Trends report, chief marketing officers (CMOs) highlight the implementation of machine learning algorithms to improve customer personalization and gain insights into customer behavior as one of their top three priorities. (Deloitte, 2023)

As businesses strive to navigate the complexities of a globalized market, the imperative for effective customer segmentation and targeted marketing strategies has become increasingly paramount. This study embarks on a comprehensive exploration of clustering algorithms tailored for customer segmentation, aiming to shed light on their comparative efficacy and potential for enhancing marketing strategies.

When faced with the daunting task of marketing to a vast customer base of millions of individuals, businesses often encounter the challenge of developing personalized marketing campaigns at scale. Recognizing the impracticality of creating distinct marketing initiatives for each customer, companies can turn to clustering as a strategic solution. (Christy *et al.*, 2021) By employing clustering algorithms, such as K-Means or DBSCAN, the expansive customer dataset can be segmented into a more manageable number of clusters. This approach allows businesses to condense the complexity of their customer base into a smaller set of distinct groups, each exhibiting similar characteristics or behaviours.

Through this method, companies can maximize the efficiency and impact of their marketing efforts, ensuring that messages resonate with targeted customer segments while streamlining resources and operational logistics. In this context, aim of this research is to compare three clustering algorithms namely K-Means, DBSCAN and OPTICS for customer segmentation.

## 1.2 Research Problem

The research problem addressed in this dissertation pertains to examining and comparing effectiveness of clustering algorithms in customer segmentation and their subsequent impact on refining marketing strategies. The focal point lies in evaluating the performance of K-Means, DBSCAN, and OPTICS algorithms concerning the segmentation of customers based on their behavioural traits and demographic characteristics as revealed by Recency, Frequency and Monetary (RFM) values. This study seeks to explore how these algorithms can be leveraged to identify distinct customer segments within a heterogeneous customer base.

## 1.3 Research Aim and Objective

This research has following aims and objectives:

- i. Understanding significance of clustering algorithms in customer segmentation.
- ii. Implementing RFM approach for analysing customer value.
- iii. Analysing performance of K-Means, DBSCAN and OPTICS clustering algorithms.
- iv. Interpreting and analyzing clusters generated by K-Means, DBSCAN and OPTICS algorithms.
- v. Explain limitations of research.

## 1.4 Dissertation Roadmap

The subsequent chapters of this dissertation project are structured to ensure logical sequence of information. They are outlined as follows:

- Chapter 2 - Literature Review: This section critically examines previous research on K-Means, DBSCAN, and OPTICS algorithms, as well as the utilization of the RFM approach for segmentation of customers.
- Chapter 3 - Research Methodology: This chapter outlines the methodology approach adopted, detailing the procedures employed and the architectural design implemented in this study.
- Chapter 4 - Analysis and Results: Here, the implementations are explained, along with the performance metrics utilized. Additionally, the chapter delves into a discussion of the obtained results.
- Chapter 5 – Conclusion, Limitations and Future Work: This concluding chapter summarizes the findings of the research and presents recommendations for future work in context of limitations identified in this study.

## Chapter 2. Literature Review

### 2.1 Significance of Clustering from Marketing Perspective

Clustering is a fundamental data analysis method in statistics and machine learning. Clustering algorithms fall in the category of unsupervised machine learning algorithms, making them critically important in business for their ability to classify data into meaningful groups without prior definitions (Hennig Christian *et al.*, 2015). This capability enables companies to uncover natural classifications within their data, facilitating more informed decision-making across various business domains such as marketing, product development, and customer service.

From a marketing perspective, clustering is invaluable for segmenting customers based on similarities in behavior, preferences, or demographics, thus allowing for targeted marketing campaigns. For instance, K-means clustering and DBSCAN clustering are widely used to identify customer segments that exhibit similar purchasing patterns or preferences (Sembiring *et al.*, 2020). This targeted approach not only optimizes resource allocation but also increases the effectiveness of promotional strategies by tailoring offerings to specific customer needs.

### 2.2 Understanding Customer Value using Recency, Frequency and Monetary parameters (RFM)

The RFM (Recency, Frequency, Monetary) approach is a widely acknowledged method for customer segmentation that quantifies customer value based on their purchasing behavior. This technique categorizes customers according to the recency of their last purchase, the frequency of their transactions, and the monetary value of their purchases, providing a robust framework for personalized marketing strategies (Anitha and Patil, 2022). Doğan *et al.* (2018) implemented the RFM technique for customer segmentation by combining it with clustering methods to analyze customer data from a sports retailing

company. The study revealed that the traditional segmentation method, based only on monetary value, failed to capture important behaviours that could inform more effective marketing strategies. By integrating RFM values, the new segmentation models allowed for a more detailed understanding of customer behaviours, leading to potential improvements in customer retention and loyalty strategies.

## 2.3 K-Means

K-means is one of the three algorithms used for the purpose of comparing clustering algorithms in this research. K-means clustering has emerged as a pivotal unsupervised learning technique for its robustness in segmenting a dataset into distinct clusters. This algorithm partitions  $n$  observations into  $k$  clusters, each defined by the proximity to the nearest mean. (Sinaga and Yang, 2020)

### 2.3.1 Functioning of K-means Clustering

The operational mechanism of K-means involves a series of phases, starting with the initialization of ' $k$ ' centroids randomly selected from the dataset as highlighted by Arthur and Vassilvitskii. (2007) who introduced the k-means++ technique to optimize this initialization phase. Following this, each data point is assigned to the nearest cluster, with distances typically calculated using the Euclidean metric. The centroids are then recalculated and updated iteratively, a process that repeats until the centroids' positions stabilize. (Sinaga and Yang, 2020)

In a Euclidean plane, distance between point  $p$  with coordinates  $(p_1, p_2)$  and point  $q$  with coordinated  $(q_1, q_2)$  will be denoted by-

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$$

### 2.3.2 K-Means Hyperparameters

The selection of hyperparameters in K-means clustering, particularly the number of clusters and the initialization method, requires meticulous tuning to reflect the dataset's intrinsic structure accurately. (Kansal *et al.*, 2018). These hyperparameters are crucial, as they significantly influence the clustering outcome'. The initialization method, K-means++, aims to enhance convergence likelihood by selecting well-spaced initial points (Arthur and Vassilvitskii, 2007)

### 2.3.3 Elbow Plot for optimal number of clusters

Number of clusters need to be specified beforehand in K-means algorithm. One of the most common methods used to determine the optimal number of clusters is the elbow method, which involves plotting the within-cluster sum of squares (WCSS) against the number of clusters and identifying the point where the WCSS begins to decrease sharply, known as the 'elbow point', as seen in figure 1. This heuristic is widely accepted for its simplicity in estimating the number of clusters that best captures the variability of the data without overfitting. (Makwana and Kodinariya, 2013) .

WCSS is calculated as -

$$\sum_{i=1}^k \sum_{X_j \in S_i} \|X_j - \mu_i\|^2$$

$X_j$  = data point in  $S_i$  cluster

$\mu_i$  = centroid of cluster

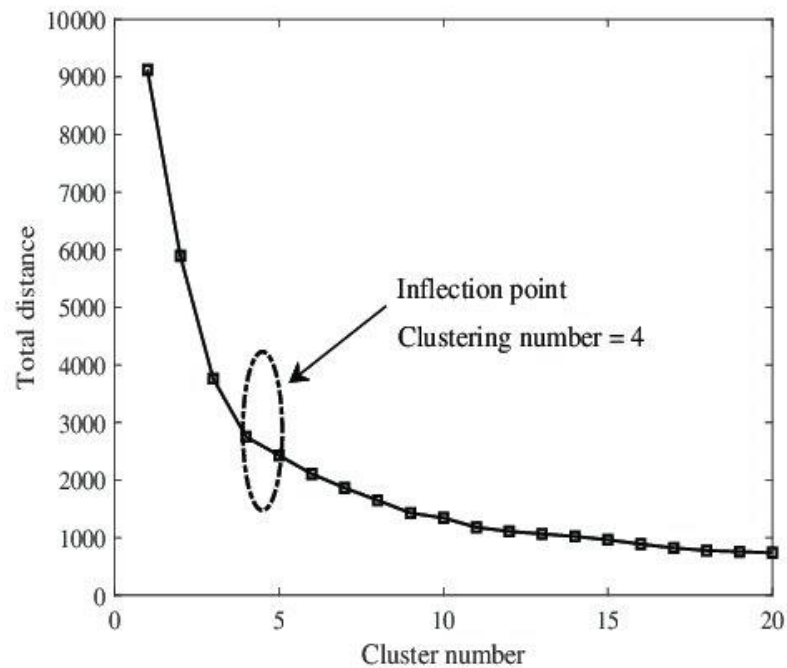


Figure 1. Elbow Plot Illustration(Zhang et al., 2020)

### 2.3.4 Applications in Customer Segmentation

K-means clustering algorithm was utilized by Kansal et al. (2018) to segment customers into five distinct groups based on spending behavior and visit frequency. This method proved effective in grouping customers into meaningful categories that could help tailor marketing strategies more precisely. However, RFM approach was not used along with K-Means which could have captured customer behaviour more comprehensively and improved customer segmentation.

In another study by Anitha and Patil (2022), K-Means was used along with RFM technique. Clusters were validated by analysing Silhouette Coefficient. The study demonstrates how integrating K-Means clustering with RFM analysis can significantly enhance customer segmentation and provide actionable insights for business intelligence in retail.

### 2.3.5 Limitations of K-Means

Although widely used for its simplicity, K-means clustering has several known limitations that can impact its effectiveness across various scenarios. One major issue is its sensitivity to how initial cluster centres are chosen, which can lead to inconsistent outcomes due to random initialization (Celebi, Kingravi and Vela, 2013). However, this drawback is addressed by K-means++ technique as discussed in 'Functioning of K-Means Clustering' section.

Additionally, K-means assumes clusters are spherical and similarly sized, which is may not be the case in real-world data. The algorithm may struggle with clusters of different sizes because it minimizes variance, naturally forming clusters of equal size, which isn't always ideal (Ikotun *et al.*, 2023). In high-dimensional spaces, K-means' effectiveness decreases because distances between points become less meaningful, a challenge known as the curse of dimensionality. To address this concern, dimensionality reduction techniques such as PCA or UMAP are used. Finally, K-means requires specifying optimal number of clusters beforehand, which is not always practical (Sinaga and Yang, 2020).

Multiple improvements over K-means have been proposed by researchers including Mini-batch K-means, X-means, K-medoids among others and discussed comprehensively by Ikotun *et al.* (2023). However, for the purpose of this research, we are comparing K-means with other clustering approaches namely DBSCAN and OPTICS.

In summary, K-means clustering remains a fundamental technique in data analysis, highlighted by its frequent adaptation and discussion across a range of studies. Its efficacy in handling large datasets efficiently ensures its continued relevance in clustering tasks across diverse domains.

## 2.4 DBSCAN

DBSCAN, also known as Density-Based Spatial Clustering of Applications with Noise, is widely used in machine learning and data mining for its ability to identify clusters of various shapes and sizes while effectively detecting outliers. It is a first density-based clustering algorithm and was proposed by Martin Ester and others (Deng, 2020). DBSCAN is also used for customer segmentation among other use cases which is the core interest of this research.

### 2.4.1 Functioning of DBSCAN Clustering

In DBSCAN, points in the dataset are classified into three distinct categories: core points, border points, and noise points, each based on their spatial relationships with neighbouring points. (Bhattacharjee and Mitra, 2021)

Core points are central to the formation of clusters. They are defined as points with a sufficient number of neighbouring points within a specified distance, denoted by  $\epsilon$ . This means that core points have a dense neighborhood, indicating a high likelihood of belonging to a meaningful cluster within the dataset. Border points, on the other hand, are points that are within the  $\epsilon$  radius of a core point but do not have enough neighbors themselves to qualify as core points. Hence, they form the periphery of clusters and are included in the cluster of their nearest core point. Noise points, sometimes referred to as outliers, do not fall within the  $\epsilon$  distance of any core points and lack sufficient neighbouring points to be considered part of a cluster. These points are typically isolated from the main clusters and are often considered as noise or irrelevant data points. (Bushra and Yi, 2021)

This categorization of points into core, border, or noise categories forms the basis of DBSCAN's clustering mechanism, allowing it to effectively identify clusters of varying densities

and shapes within the dataset while also being able to detect and disregard noisy or irrelevant data points.

#### 2.4.2 DBSCAN Hyperparameters

Tuning of hyperparameters is critical for performance of DBSCAN algorithm. Deng (2020) has explained at length meaning and importance of two crucial hyperparameters of DBSCAN algorithm. First hyperparameter epsilon ( $\epsilon$ ) plays a critical role in determining the radius within which the algorithm searches for neighbouring points around each data point. Essentially, it defines the maximum distance that a point can be from another point to be considered a neighbour. A smaller  $\epsilon$  value results in tighter clusters, as points need to be closer to each other to be considered part of the same cluster. On the other hand, a larger  $\epsilon$  value can lead to more expansive clusters, potentially merging multiple smaller clusters into one.

The MinPts hyperparameter sets the minimum number of neighbors that a point must have within the  $\epsilon$  radius to be considered a core point. This parameter helps control the density threshold required for a cluster. A higher MinPts value results in more stringent density requirements, meaning that only points in denser regions will be labelled as core points. Conversely, a lower MinPts value allows for the inclusion of points in less dense areas, potentially leading to the formation of larger clusters encompassing more sparse regions of the dataset. Adjusting MinPts can influence the granularity and size of the resulting clusters, impacting the overall clustering performance of the algorithm. (Deng, 2020)

#### 2.4.3 Applications in Customer Segmentation

Monalisa et al.(2023) employed DBSCAN to segment customers based on RFM models and demographic variables, achieving a silhouette index of 0.4222. The relatively low silhouette index of 0.4222 suggests that the clusters may not be highly distinct, indicating

potential limitations in the clustering effectiveness or the selection of parameters such as epsilon and MinPts.

Similarly, ŞAHİNBAŞ (2022) explored the performance of DBSCAN alongside K-Means for airline customer segmentation. K-Means was found to be more effective than DBSCAN, as indicated by the higher silhouette value, suggesting better cluster homogeneity. This might indicate limitations in DBSCAN's ability to handle specific types of data distributions typical to the airline industry, such as high dimensionality or sparse data.

#### 2.4.4 Limitations of DBSCAN

DBSCAN's performance can be affected by parameter selection, making it sensitive to variations in  $\epsilon$  and MinPts. It may also struggle with clusters of varying densities or irregular shapes and is less effective in high-dimensional spaces where distance becomes less clear. Despite these limitations, DBSCAN remains a valuable tool for clustering tasks involving noisy datasets and diverse cluster characteristics. (Bushra and Yi, 2021)

Furthermore, the application of DBSCAN extends beyond typical clustering tasks. For instance, it has been used in image processing for the segmentation of super pixels and in the detection of anomalies in network traffic (Shen *et al.*, 2016). These applications benefit from DBSCAN's ability to adjust to the intrinsic properties of the dataset without the need for predefined cluster numbers.

## 2.5 OPTICS

The OPTICS (Ordering Points To Identify the Clustering Structure) clustering algorithm was developed as an improvement over the DBSCAN, offering better performance on datasets with varying densities. (Subudhi and Panigrahi, 2022)

### 2.5.1 Functioning of OPTICS Clustering

The recent advancements in the OPTICS clustering algorithm highlight its distinction and improvements over traditional DBSCAN through its unique mechanisms like core-distance and reachability-distance. Core distance is defined as the maximum distance within which a point must have a minimum number of neighbors to be considered a core point, similar to the core points in DBSCAN. Reachability distance, on the other hand, is indicative of how closely points are clustered. Lower values of reachability distance suggest higher density. Reachability distance is pivotal for constructing reachability plot that captures the intrinsic structure of the data set. (Gialampoukidis et al., 2016)

The ordered list generated by OPTICS does not form clusters immediately but rather uses reachability plots to discern the structure. Peaks in this plot typically signify transitions between clusters or noise, while valleys suggest potential clusters (Deng *et al.*, 2015). This technique enhances the detection of clusters, especially in cases where data varies in density. Figure 2 illustrates formation of clusters in OPTICS based on reachability plot.

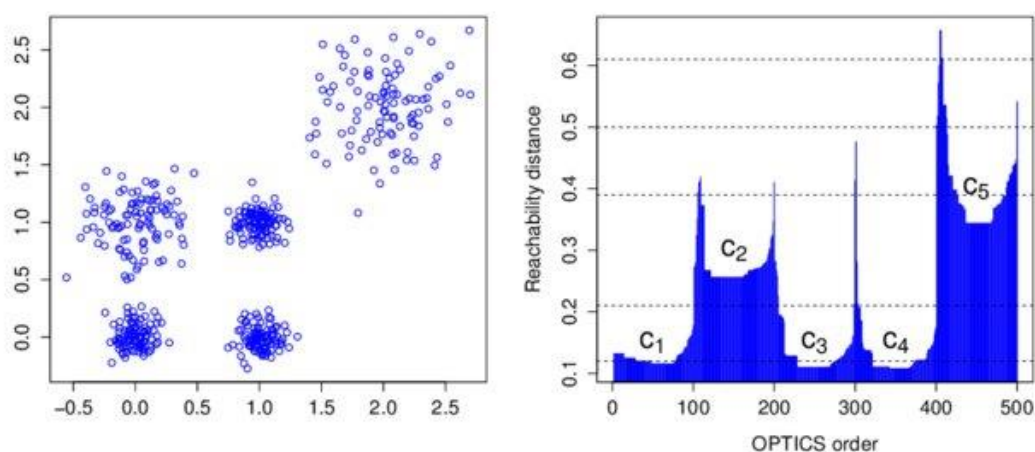


Figure 2. Reachability Plot (Gialampoukidis et al., 2016)

### 2.5.2 OPTICS Hyperparameters

The OPTICS clustering algorithm uses several hyperparameters that significantly influence its performance and effectiveness. Key among these hyperparameters are minPts and  $\epsilon$  (epsilon), both of which are same as DBSCAN hyperparameters.

Adjusting these parameters can drastically change the resulting clustering structure, as they define what is considered a cluster's density threshold. For example, a smaller  $\epsilon$  might lead to many small clusters, while a larger  $\epsilon$  could merge those into fewer, larger clusters. Similarly, a higher minPts value increases the density requirement, potentially leading to fewer, more densely packed clusters (Subudhi and Panigrahi, 2022)

The hyperparameter xi in the OPTICS clustering algorithm plays a crucial role in determining cluster boundaries based on reachability-distance. Xi is a threshold value that defines the minimum steepness or drop in reachability distance necessary to start a new cluster. A smaller xi value results in more sensitive cluster separation, potentially identifying more clusters, while a larger xi value may result in fewer, more significant clusters by only considering substantial changes in density as different clusters. (Deng *et al.*, 2015)

### 2.5.4 Limitations of OPTICS

OPTICS processes large datasets with a complexity of  $O(n^2)$  in the worst-case scenario, which can lead to significant computational overhead for large datasets as highlighted by Kim *et al.* (2019).

Moreover, Kanagala and Krishnaiah (2016) explains how effectiveness of OPTICS heavily depends on the choice of its parameters, namely minPts and  $\epsilon$  (epsilon). Incorrect settings for these parameters can lead to over-segmentation or under-segmentation of the data, which is also the case with DBSCAN.

Lastly, while OPTICS is designed to handle noise better than DBSCAN by adjusting the epsilon parameter dynamically, it can still struggle with high noise levels, especially in data with complex structures or very sparse regions. This can result in misclassification of noise as part of clusters or the creation of multiple small clusters that are actually noise, affecting the overall quality of the clustering. (Kim *et al.*, 2019)

## 2.6 Dimensionality Reduction

### 2.6.1 Advantages of dimensionality reduction in clustering

Dimensionality reduction is crucial for clustering, especially in handling large, high-dimensional datasets such as those involved in customer segmentation. By reducing the number of variables, dimensionality reduction techniques simplify the complexities involved in clustering processes, enhance visualization, and reduce computational costs. Moreover, it also helps in preserving the essential patterns in the data while eliminating noise and redundant features, thereby improving the performance of clustering algorithms (Allaoui *et al.*, 2020)

### 2.6.2 Using UMAP over PCA

Uniform Manifold Approximation and Projection (UMAP) has demonstrated several advantages over Principal Component Analysis (PCA), particularly in clustering applications. UMAP preserves both local and global relationships between data points, which is crucial for revealing intrinsic groupings within customer data. This is unlike PCA, which primarily focuses on global relationships. (Hozumi *et al.*, 2021)

Allaoui *et al.* (2020) observed that UMAP, when used before clustering algorithms like K-means, improved the clustering accuracy by adapting better to the local and global data

structure. They demonstrated up to 60% improvement in clustering performance on image datasets, indicating UMAP's robustness in diverse applications.

Furthermore, UMAP effectively manages non-linear relationships within data, making it superior for datasets where customer behaviours and interactions are complex and multi-dimensional. PCA mainly relies on linear variance and may miss non-linear interactions (Allaoui et al., 2020)

While utilizing PCA for customer segmentation in e-commerce, Bandyopadhyay et al.(2021) noted improvements in clustering by reducing dimensionality of the data. However, their study suggested limitations in capturing complex, non-linear patterns using PCA when compared to more advanced techniques like UMAP.

### 2.6.3 Functioning of UMAP

UMAP was introduced by McInnes et al. (2018) in 2018 while elaborating its functioning in the same research. UMAP operates by projecting a radius from each data point, with neighbouring points falling within this radius considered part of the same cluster. However, for high-dimensional data, directly applying this approach isn't feasible as we lack a clear visualization of the data. Choosing a small radius could isolate data points, while a large radius might include too many points in the same cluster.

To address this, UMAP employs a variable radius strategy: from each point, a radius extending up to the  $n$ th nearest neighbour is projected. The likelihood of a point joining a cluster is determined using fuzzy radii, where the probability diminishes as distance increases. This information is represented as a graph, with neighbouring points connected by coloured edges, darker edges indicating a higher probability of being neighbors. Conflicting edges, where multiple edges point to the same point, are merged into a single edge with a combined

probability. Ultimately, each pair of data points will have at most one edge between them. UMAP tackles this as an optimization problem, employing stochastic gradient descent. (McInnes et al., 2018)

Key hyperparameters include `n_neighbours`, determining the *n*th nearest neighbour used for radius extension, and `min_dist`, controlling the minimum distance between similar data points in the low-dimensional space. Adjusting these parameters allows for fine-tuning UMAP's performance across different datasets and applications.

In context of above discussion, UMAP will be used for dimensionality reduction for the purpose of this research for its multiple advantages over PCA.

## 2.7 Comparative Analysis of K-Means, DBSCAN and OPTICS

The comparative analysis of clustering algorithms such as K-Means, DBSCAN, and OPTICS is instrumental in understanding their respective strengths and limitations, particularly in the context of customer segmentation. This analysis serves as a crucial element of literature review of this study by providing a detailed examination of how each algorithm approaches the challenge of clustering large datasets within various business applications.

A study by Brahmana et al. (2020) to segment customers into meaningful groups based on their transactional behaviours using K-Means, K-Medoids, and DBSCAN involved a dataset of 334,641 transactions which was condensed into 1661 RFM data points. K-Means exhibited the highest level of clustering validity with a Davies-Bouldin Index of 0.33 and a Silhouette Index of 0.91. K-Medoids and DBSCAN showed lower levels of validity compared to K-Means. DBSCAN was noted to form clusters effectively, but it was sensitive to the parameter settings, which required careful adjustment to avoid suboptimal clustering.

In another relevant research, K-means and DBSCAN was employed for enhancing customer segmentation by Hossain (2017). K-means was tested with different numbers of clusters and distance metrics including Euclidean and Manhattan. DBSCAN was applied with various settings for epsilon and MinPts. The comparison between K-means and DBSCAN revealed that while K-means is faster and suitable for well-defined and spherical clusters, DBSCAN provided better results in terms of identifying non-linear cluster shapes and managing noise within the data. However, this study focuses on annual spending data for customer segmentation instead of more comprehensive RFM approach, which may have further improved cluster formations.

Furthermore, for segmenting customers of a shopping mall, Saxena et al. (2024) experimented with K-Means, Hierarchical Clustering, Affinity Propagation, and DBSCAN. Unlike the RFM model-based segmentation that primarily focuses on customer transactional behavior used by Brahmana et al. (2020) , this study incorporates demographic factors like gender and income, thus offering a multi-dimensional view of customer segmentation.

The study concludes that each clustering method has its own strengths and is suitable for different types of data characteristics. K-Means and hierarchical clustering are effective for general customer segmentation tasks, while affinity propagation and DBSCAN offer more flexibility and robustness in handling complex data structures.

In summary, after analyzing multiple clustering approaches and their applications in customer segmentation in this literature review, it can be concluded that effective clustering depends on factors such as type of model, dataset used, hyperparameter tuning among others.

Novelty of this research lies in using RFM in combination with not only K-Means and DBSCAN but also OPTICS algorithm to determine most effective approach for segmentation of customers and using this information for making data driven marketing decisions.

## Chapter 3. Research Methodology

### 3.1 Introduction to Methodology

In this chapter, we delve into the methodology adopted for conducting a comparative analysis of three distinct clustering algorithms—K-Means, DBSCAN, and OPTICS—applied to the task of customer segmentation. This study leverages the Recency, Frequency, and Monetary (RFM) model to quantify customer behavior, providing a structured approach to evaluate customer value and segment customers accordingly. The choice of clustering algorithms is motivated by their differing operational principles and suitability for handling various data characteristics in customer segmentation tasks.

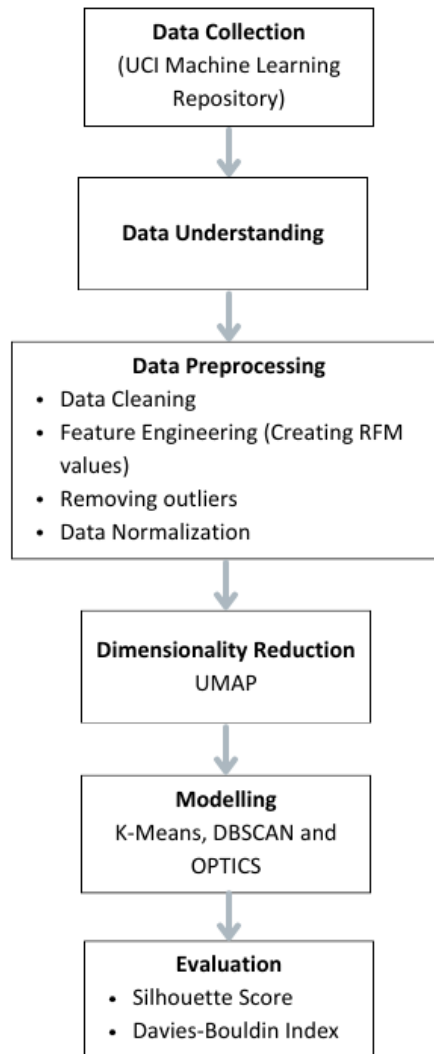
The research methodology focuses on data preprocessing, including cleaning and standardizing RFM values, to secure high-quality data for clustering. Dimensionality reduction is achieved using UMAP to simplify the complexity of high-dimensional data and enhance the effectiveness of clustering processes. Parameter selections for K-Means, DBSCAN, and OPTICS are carefully adjusted to suit varying data densities, with clustering performance evaluated using silhouette scores and Davies-Bouldin indices. These metrics facilitate a comparative analysis of each algorithm's capability in segmenting customers.

Implementation details specify the use of Python and various libraries. Scikit-learn is employed for clustering, UMAP-learn for dimensionality reduction, and Plotly for visualization. These tools ensure reproducibility and address computational demands. Notably, challenges associated with the computationally intensive OPTICS algorithm are managed to maintain accuracy without compromising performance.

This methodology chapter sets the foundation for the subsequent analysis of results, where the insights drawn from the clustered data are interpreted to inform targeted

marketing strategies, ultimately aiming to enhance customer engagement and business performance.

The methodology along with implementation is depicted in figure 3.



*Figure 3. Methodology Flow*

### 3.2 Data Collection

The dataset utilized in this research was sourced from the UCI Machine Learning Repository, specifically the Online Retail dataset (Chen Daqing, 2015) which is openly available in public domain. This transnational dataset encompasses all transactions recorded from December 1, 2010, to December 9, 2011, for a UK-based, non-store online retail firm specializing in unique all-occasion gifts. The dataset's significance is underscored by its prior use in the research paper "Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining" by Daqing Chen, Sai Laing Sain, and Kun Guo, published in 2012.

### 3.3 Data Characteristics

Characterized as multivariate and sequential, the dataset facilitates time-series analysis and is pertinent for tasks such as classification and clustering within the business domain. It contains a total of 541,909 instances, distributed across six features, which are predominantly categorical and integer types. These features include InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID and Country. Details of these features are mentioned in the table below.

*Table 1. Data Description*

Feature	Description	Data Type
<b>InvoiceNo</b>	A unique 6-digit transaction ID, where codes starting with 'c' denote cancellations	Object
<b>StockCode</b>	A 5-digit code for each distinct product	Object
<b>Description</b>	Product name	Object
<b>Quantity</b>	Item quantities per transaction	Integer
<b>InvoiceDate</b>	Date and time of the transaction	Object
<b>UnitPrice</b>	Product price per unit in sterling	Float
<b>CustomerID</b>	A unique 5-digit number for each customer	Float

Country	Country of the customer	Object
---------	-------------------------	--------

Each variable provides essential insights into the business operations, customer behaviors, and transactional dynamics of the company. The comprehensive nature of this dataset, with detailed attributes ranging from product information to customer details and transaction timestamps, makes it a robust foundation for performing sophisticated customer segmentation and analysis as envisaged in this study. Moreover, the inclusion of wholesaler data alongside regular customer transactions enriches the dataset's diversity, offering a nuanced view of the market dynamics and consumer purchasing patterns in a business-to-business (B2B) context alongside conventional retail scenarios.

### 3.3 Data Pre-processing

The data preprocessing stage is critical to ensuring the quality and reliability of the results obtained from clustering analyses. This section outlines the procedures followed in cleaning the dataset, engineering relevant features, and preparing the data for further analysis.

#### 3.3.1 Data Cleaning

The initial step in data preprocessing involved cleaning the dataset to ensure its suitability for analysis. Specifically, the dataset contained a significant number of missing values in the `CustomerID` field, which is essential for customer segmentation. Out of the original 541,909 transactions recorded, rows with missing `CustomerID` values were identified and removed, reducing the dataset to 406,829 transactions. This step was crucial as it directly impacts the reliability of the customer segmentation process by ensuring that only transactions linked to identifiable customers are analyzed.

### 3.3.2 Feature Engineering

In this study, the feature engineering process was crucial for transforming the raw transaction data into actionable insights through the creation of RFM metrics. These metrics—Recency, Frequency, and Monetary—serve as the foundation for understanding customer behavior and facilitating effective segmentation.

The Recency metric captures how recent the last transaction was for each customer. To compute this, the `InvoiceDate` was first converted into a datetime format to handle date calculations accurately. Following this conversion, the dataset was sorted by `CustomerID` and `Date`, and the rank of each transaction was established with respect to its recency, assigning the rank of 1 to the most recent transaction per customer. The most recent date for each customer was then extracted to a new dataframe `df_rec`. From this, the Recency value was calculated as the difference in days between the earliest transaction date in the dataset and each customer's last purchase date.

The Frequency metric measures the total number of transactions per customer. It was computed by counting the number of transactions for each customer from the previously created `df_rec` dataframe, which included only the most recent transactions per customer. This data was grouped by `CustomerID`, and the count of dates (transactions) was calculated, creating a `df_freq` dataframe. Each row in `df_freq` represents a unique customer and their total number of transactions during the observed period.

Monetary value quantifies the total spending per customer. This was derived by multiplying the `Quantity` by the `UnitPrice` for each item in each transaction, summing this product over all transactions for each customer. The resulting monetary values were then

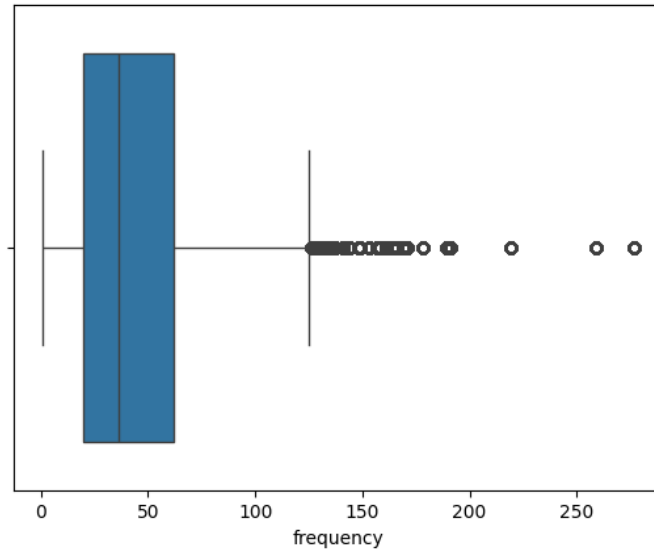
aggregated at the customer level to compute the total spend per customer, stored in the dataframe `m`.

After individually calculating the RFM metrics, these were combined to create a comprehensive view of each customer's transactional behavior. The `df\_freq` dataframe containing the Frequency metric was merged with `df\_rec` on `CustomerID`, integrating the most recent purchase date (Recency) with the transaction count (Frequency). Following this, the Monetary values from dataframe `m` were merged into this combined dataset. The resultant dataframe, `rfm`, thus incorporated all three RFM metrics—each key to segmenting customers based on their transactional patterns. To facilitate clustering analysis, the `rfm` dataframe was further refined to include only relevant RFM metrics: `CustomerID`, `Recency`, `Frequency`, and `Monetary` values. This final dataframe, `finaldf`, encapsulates the essential characteristics of each customer, serving as the input for subsequent clustering processes.

The construction and integration of these RFM metrics not only ensure a robust dataset for clustering but also provide a structured approach to understanding customer dynamics, which is critical for effective segmentation and targeted marketing strategies.

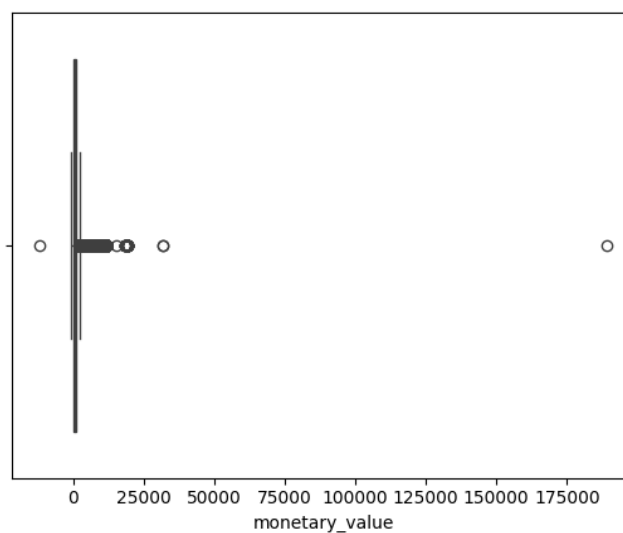
### 3.3.3 Removing Outliers

Outliers can skew the analysis and lead to misleading conclusions. Therefore, an examination of outliers was conducted for the `recency`, `frequency`, and `monetary\_value` variables using boxplots. Subsequent removal of outliers was based on the calculation of Z-scores for these variables, with entries having a Z-score less than 3 being retained.



*Figure 4. Frequency Boxplot*

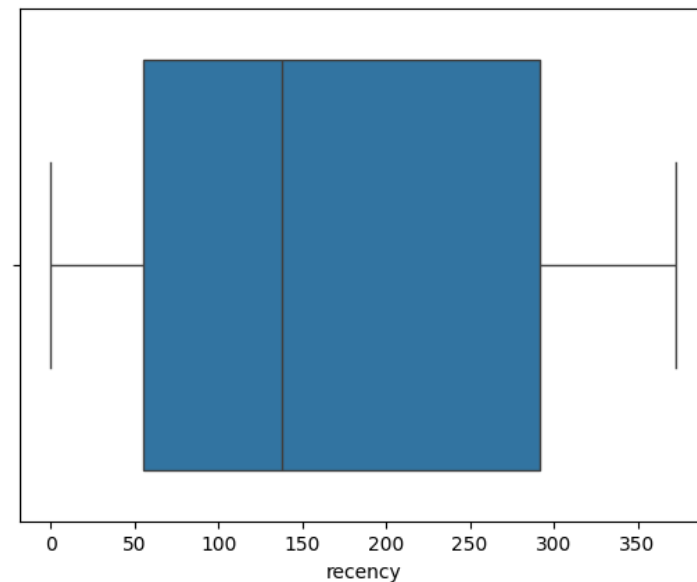
The frequency boxplot in figure 4 shows a concentrated distribution with a majority of the data clustered towards the lower end, indicating that most customers have fewer transactions. The whiskers extend to roughly 50 transactions, beyond which all data points are considered outliers. These outliers represent customers with unusually high transaction frequencies, which might be due to bulk purchasers or wholesalers.



*Figure 5. Monetary Value Boxplot*

Similar to `frequency`, the `monetary\_value` data, as shown in figure 5 is highly skewed with most of the data concentrated at the lower end of the scale. The outliers in this

case represent customers who have spent significantly more than typical customers, possibly due to purchasing high-value items or in bulk. Notably, there are a few extreme outliers, with values reaching up to around 175,000, which could greatly skew any analysis.



*Figure 6. Recency Boxplot*

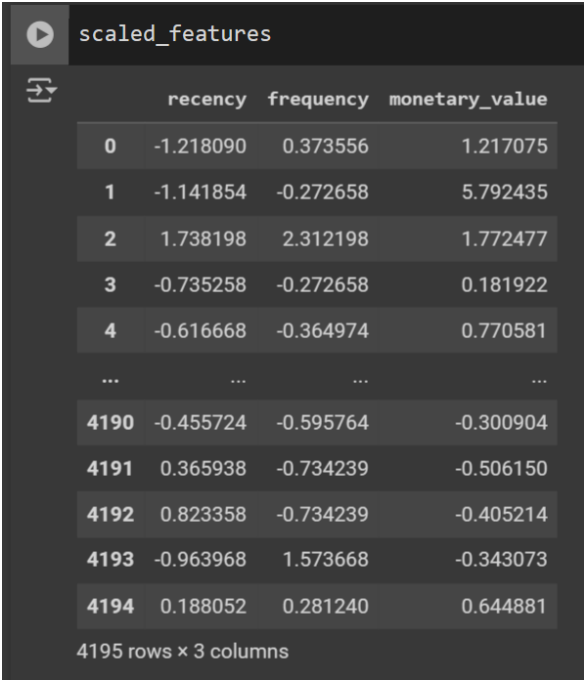
Unlike `frequency` and `monetary_value`, `recency` shows a more uniform distribution across its range with no visible outliers as depicted in figure 6. This suggests that the customer interaction over the period studied was fairly consistent, with most customers making their most recent purchases within a similar time frame.

To remove outliers, Z-score for each data point in the `recency`, `frequency`, and `monetary_value` metrics were calculated. Data points with a Z-score greater than 3 or less than -3 were removed, as these scores indicate that the data points are significantly distant from the mean, lying outside the typical distribution of the data. This approach helps in mitigating the impact of extreme values that could distort the clustering process.

### 3.3.4 Data Normalization

Normalization is a crucial step in the data preprocessing phase, particularly in preparation for clustering analysis, where the scale of variables significantly impacts the results. This transformation is vital as it neutralizes the effect of differing scales among variables, allowing each feature to contribute equally to the analysis without any single feature dominating due to its scale.

In the provided dataset, standardization was applied to the RFM features, leveraging the `StandardScaler` from Scikit-learn's preprocessing module. The output of this transformation is a set of scaled features, where each feature now has a mean of zero and a standard deviation of one. These scaled features are stored in a new DataFrame called `scaled_features`, which can then be used in subsequent analyses.



```
scaled_features
```

	recency	frequency	monetary_value
0	-1.218090	0.373556	1.217075
1	-1.141854	-0.272658	5.792435
2	1.738198	2.312198	1.772477
3	-0.735258	-0.272658	0.181922
4	-0.616668	-0.364974	0.770581
...	...	...	...
4190	-0.455724	-0.595764	-0.300904
4191	0.365938	-0.734239	-0.506150
4192	0.823358	-0.734239	-0.405214
4193	-0.963968	1.573668	-0.343073
4194	0.188052	0.281240	0.644881

4195 rows x 3 columns

*Figure 7. Scaled Features*

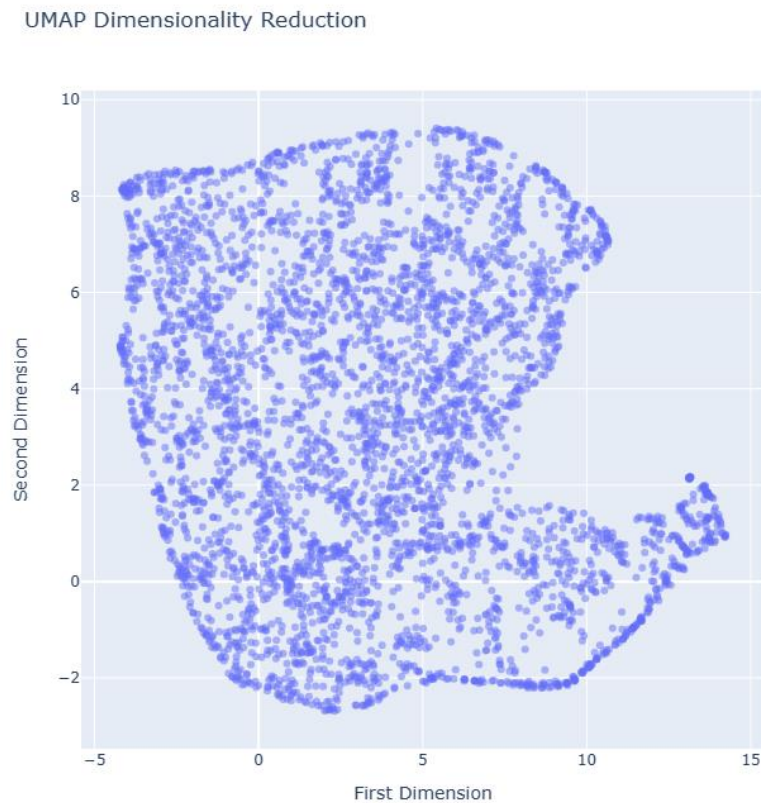
### 3.4 Dimensionality Reduction

In this study, the Uniform Manifold Approximation and Projection (UMAP) technique is employed to reduce the dimensionality of the feature space while preserving the intrinsic structure of the data. This method facilitates a more effective clustering by simplifying the visualization and analysis. (Allaoui et al., 2020)

In the application to this dataset, UMAP is configured to project RFM features into a two-dimensional space. The choice of two dimensions is intentional to facilitate easy visualization and to support the clustering algorithms that will follow. The implementation involves several key parameters:

- `n_components`: Set to 2, indicating the number of dimensions to which the data should be reduced.
- `n_neighbors`: A hyperparameter that controls the local structure of the data. It determines how UMAP balances local versus global structure in the data.
- `min_dist`: This parameter controls how tightly UMAP is allowed to pack points together. Smaller values result in more clustered embeddings, which is beneficial for identifying denser clusters effectively.

The specific settings were chosen based on preliminary experiments that suggested they provide a good balance between maintaining data integrity and achieving a meaningful reduction for clustering purposes.



*Figure 8. UMAP Visualization*

## 3.5 Evaluation Metrics

Cluster evaluation is a critical aspect of clustering analysis in data mining and machine learning. It involves assessing the quality and validity of the clusters produced by different clustering algorithms. Effective cluster evaluation helps in understanding the structure of the data, the appropriateness of the clustering algorithm, and the optimal parameter settings. Key metrics used for cluster evaluation include the Silhouette Score and the Davies-Bouldin Index, both of which provide insights into the intra-cluster cohesion and inter-cluster separation, thereby guiding the selection of the most suitable clustering approach.

### 3.5.1 Silhouette Score

The Silhouette Score is a widely used metric for evaluating the quality of clusters formed by various clustering algorithms. It measures how similar a data point is to its own

cluster compared to other clusters. The score is defined for each sample and ranges from -1 to +1, where score close to +1 indicates a well-clustered point, a score around 0 indicates a point on the boundary between clusters, and a score close to -1 indicates a point likely misclassified. (Shahapure and Nicholas, 2020)

Specifically, the Silhouette Score  $s(i)$  for a data point  $i$  is calculated as follows:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

where:

- $a(i)$  is the mean distance between  $i$  and all other points in the same cluster.
- $b(i)$  is the mean distance between  $i$  and all points in the nearest cluster to which  $i$  does not belong.

### 3.5.2 Davies-Bouldin Index

The Davies-Bouldin Index (DBI) is another prominent metric used for cluster validation. It looks at two things: how close points in a cluster are to each other (compactness) and how far apart different clusters are (separation). Compactness measures how similar the points in a single cluster are. If points are close together, compactness is high. On the other hand, separation measures how distinct or separate clusters are from each other. If clusters are far apart, separation is high. (Ashari et al., 2022)

The DBI is calculated by averaging the similarity ratio for each cluster. This ratio compares the compactness of a cluster with the separation from the most similar other cluster.

$$DBI = \frac{1}{k} \sum_{i=1}^k \max_{i \neq j} \left( \frac{d_i + d_j}{d_{ij}} \right)$$

where:

- $k$  is the number of clusters.
- $d_i$  is the average distance between each point in the  $i^{\text{th}}$  cluster and the centroid of the  $i^{\text{th}}$  cluster.
- $d_{ij}$  is the distance between the centroids of the  $i^{\text{th}}$  and  $j^{\text{th}}$  clusters.

A lower DBI value means better clusters. It indicates that clusters are tight and well-separated. A higher DBI value means poorer clustering, with more overlap and less distinct clusters.

Ashari et al. (2022) demonstrated the use of the Davies-Bouldin Index for evaluating the KMeans clustering of IMDb movie data. They computed the DBI for different numbers of clusters and selected the number that minimized the DBI, thereby achieving optimal clustering results.

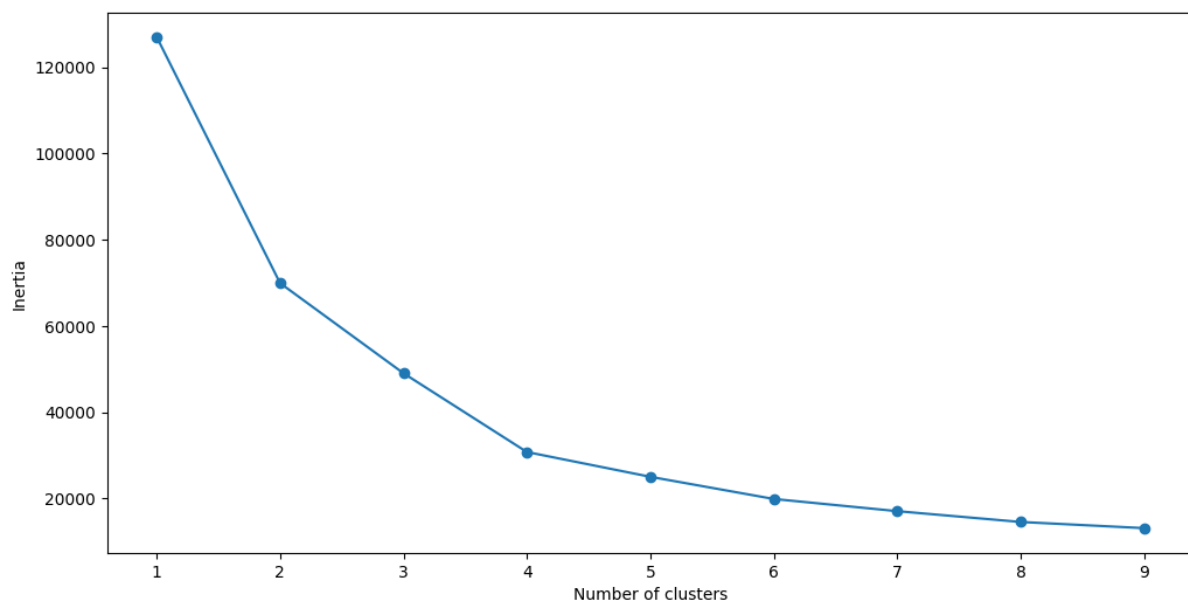
### 3.6 Clustering Algorithms

This section focuses on the application of three distinct clustering algorithms- K-Means, DBSCAN, and OPTICS, each chosen for their unique approaches to clustering and their potential to reveal different aspects of customer behavior. These algorithms are instrumental in segmenting customers into groups with similar characteristics, facilitating targeted marketing strategies and optimizing resource allocation.

The subsequent subsections will delve into the implementation of each clustering method, their parameter settings and rationale behind their selection, culminating in a comparative analysis of their performance in segmenting the customer base.

### 3.6.1 K-Means

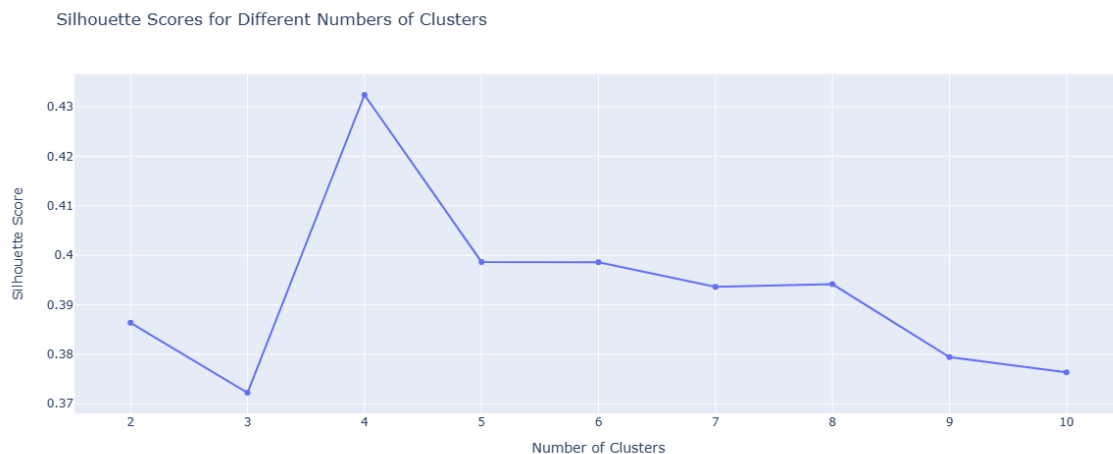
To determine the optimal number of clusters (K), the elbow method was utilized. This approach involves plotting the sum of squared distances from each point to its assigned centre (inertia) against the number of clusters as mentioned in section 2.3.3. In this study, the elbow plot suggested that the reduction in inertia slows significantly around four clusters, suggesting an optimal cluster count.



*Figure 9. Elbow Plot*

Further validation of the cluster count was done using silhouette scores, which measure the quality of the clusters formed. A silhouette score near +1 indicates that the sample is well-clustered and distant from neighboring clusters. A silhouette score near 0 indicates ambiguity regarding which cluster the sample belongs to. (Shahapure and Nicholas, 2020). The silhouette scores for different numbers of clusters were calculated as depicted in

figure 10, showing that four clusters had a relatively high score, further supporting the choice derived from the elbow method.



*Figure 10. Silhouette Scores for Cluster Range*

Using the insights from the elbow method and silhouette scores, K-Means was executed with four clusters. The algorithm was initialized with the 'k-means++' method to ensure that the initial cluster centres are optimally spaced (Arthur and Vassilvitskii, 2007), reducing the likelihood of poor clustering performance due to sub-optimal initial positions.

### 3.6.2 DBSCAN

DBSCAN is a robust clustering algorithm that identifies clusters as areas of high density separated by areas of low density. To effectively apply DBSCAN, two critical hyperparameters must be optimized: ``eps`` (epsilon), which determines the maximum distance between two points for one to be considered as in the neighborhood of the other, and `'min_samples'` which defines the minimum number of points required to form a dense region.

The optimal values for these parameters were determined using a grid search approach. The range of `eps` values tested was from 0.1 to 0.5, increased in increments of 0.05, and `min\_samples` values from 3 to 10. Through this grid search, the best configuration found was an `eps` of 0.15 and `min\_samples` of 3, yielding the highest silhouette score of 0.132.

### 3.6.3 OPTICS

The primary advantage of OPTICS over other clustering algorithms is its ability to identify clusters in datasets with varying densities, making it particularly useful for complex customer data. (Deng *et al.*, 2015)

To optimize the performance of the OPTICS algorithm, a grid search was conducted to fine-tune its hyperparameters, which included the minimum number of samples in a neighborhood for a point to be considered a core point (`min_samples`), the minimum steepness required to start a new cluster (`xi`), and the minimum number of points that a cluster needs to contain (`min_cluster_size`). The grid search explored a range of values for these parameters: 5, 10, 15, and 20 for the minimum number of samples, 0.01, 0.03, 0.05, 0.07, and 0.09 for the minimum steepness, and 10, 20, 30, and 40 for the minimum cluster size.

The silhouette score was used to evaluate the performance of each hyperparameter combination. The optimal configuration identified had a minimum of 15 samples, a steepness of 0.01, and a minimum cluster size of 10, achieving the highest silhouette score.

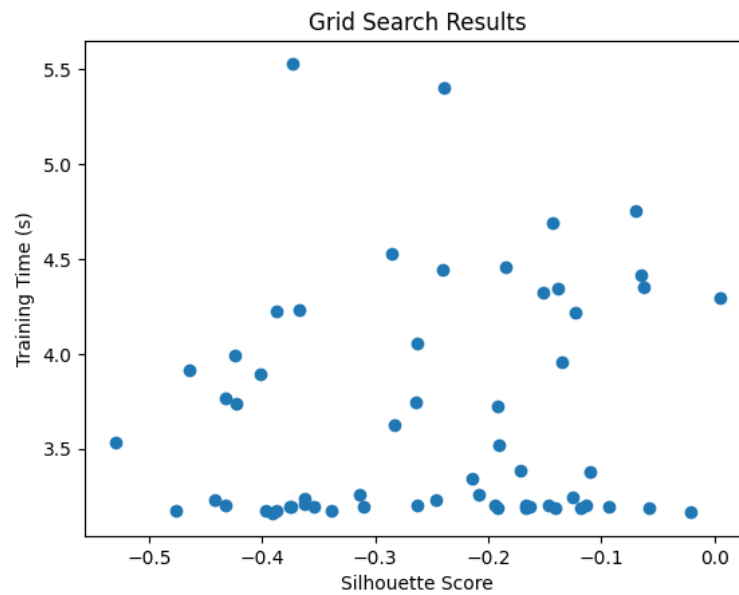


Figure 11. OPTICS Hyperparameter Tuning using GridSearch

### 3.7 Software and Tools

In this work, Google Colab was utilized as the primary computational platform due to its cloud-based environment and Jupyter Notebook based interface which facilitates interactive and flexible coding environment. Python was chosen as the programming language for its extensive libraries and frameworks, which are well-suited for data analysis and machine learning. Key Python libraries such as Scikit-Learn, Pandas, NumPy, Matplotlib, Seaborn, and Plotly were employed to implement clustering algorithms (KMeans, DBSCAN, OPTICS), perform data preprocessing and computing evaluation metrics.

## Chapter 4. Results and Analysis

K-Means visualization shows the clusters identified by the K-Means algorithm with four clusters. Clear boundaries between clusters suggest that the chosen number of clusters (four) is appropriate for this dataset. The clusters appear well-separated, indicating that the algorithm has effectively grouped the data into distinct segments. The training time for K-Means was 1.063 seconds, which is relatively quick, demonstrating the efficiency of the K-Means algorithm.

The silhouette score obtained is 0.445, which is a relatively good score. It suggests that the clusters are reasonably well-separated and distinct, although there might still be some overlap or ambiguity. The Davies-Bouldin Index obtained is 0.736, suggesting that the clusters are compact and well-separated, indicating good clustering performance.

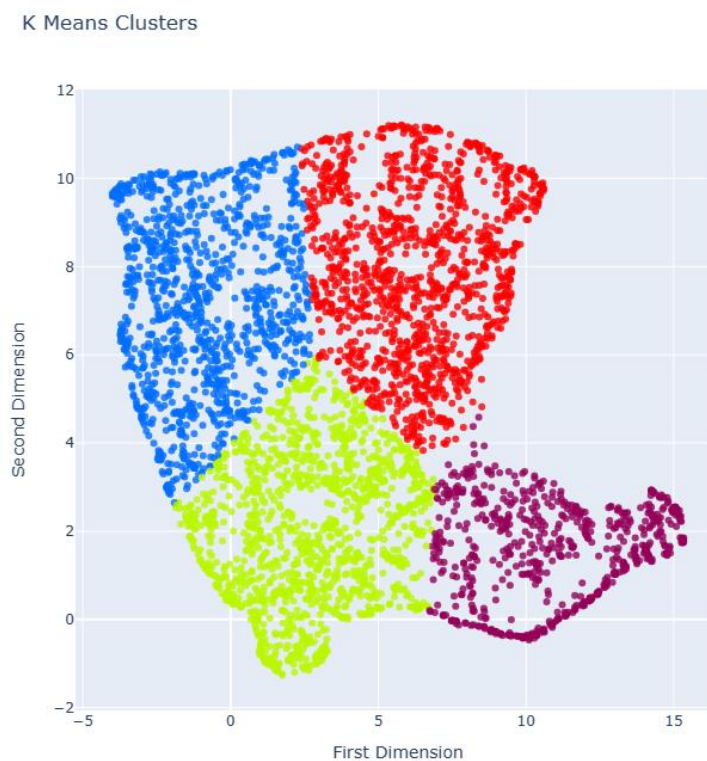
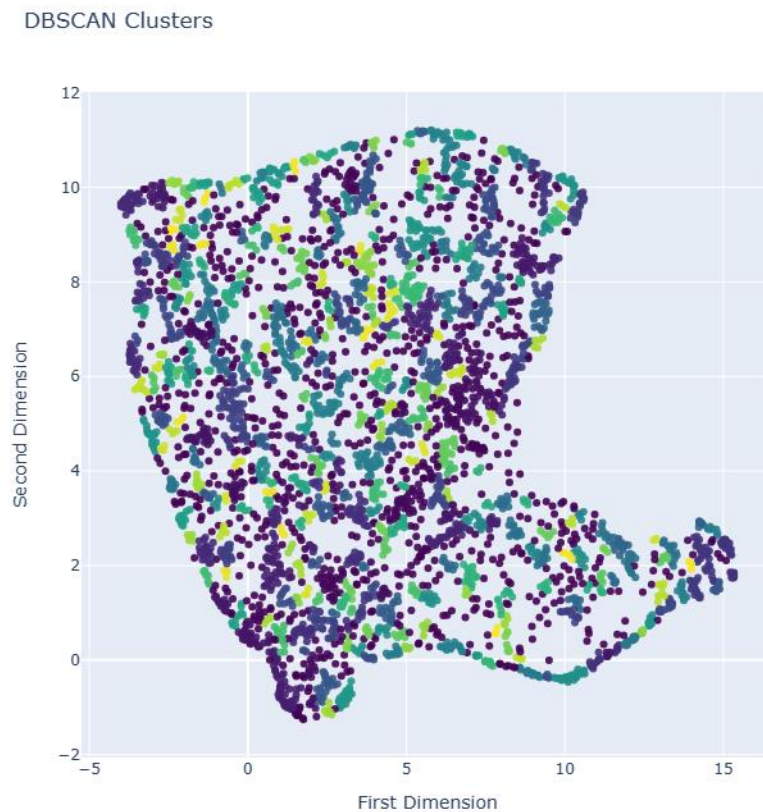


Figure 12. K-Means Cluster Visualization

For DBSCAN, training time was 0.032 seconds, which is extremely fast. This highlights the efficiency of DBSCAN in clustering datasets, especially with appropriate parameter settings. However, DBSCAN has identified many small clusters and a significant number of noise points. The clusters appear to be more scattered and less distinct compared to the clusters formed by K-Means and OPTICS. The presence of many noise points indicates that DBSCAN is sensitive to the density variations in the data.

The silhouette score obtained is 0.132 for DBSCAN, which is relatively low. This indicates that the clusters are not very well-separated and that there is some ambiguity in the cluster assignments. This score is similar to the silhouette score obtained with OPTICS. DBSCAN's Davies-Bouldin Index of 1.435 while modest, indicates some level of cluster separation but also suggests overlapping or closely packed clusters.

While comparing K-Means and DBSCAN on Airline customer database, ŞAHİNBAŞ (2022) noticed that K-Means produced better Silhouette score than DBSCAN owing to variable density of the data and high dimensionality. DBSCAN result of this study resonates with findings of aforementioned research.



*Figure 13. DBSCAN Cluster Visualization*

OPTICS is designed to handle varying densities better than DBSCAN, and this is evident in the way it has identified clusters in denser regions while still detecting sparser clusters. However, silhouette score of 0.132 suggests that the clusters formed are not very distinct overall. This could be due to the complex structure of the data or the presence of noise. The Davies-Bouldin Index value of 1.435 implies moderate clustering quality. In the scatter plot visualization, clusters appear to be relatively compact in some areas but there is some overlap, especially in the central region which is consistent with the relatively low silhouette score and moderate Davies-Bouldin Index.

Notably, time taken by OPTICS is considerably more than DBSCAN. This is because in addition to calculating core distance, OPTICS also calculates reachability distance for each point resulting in more computations per point as observed in section 2.5.4. (Kim *et al.*, 2019)

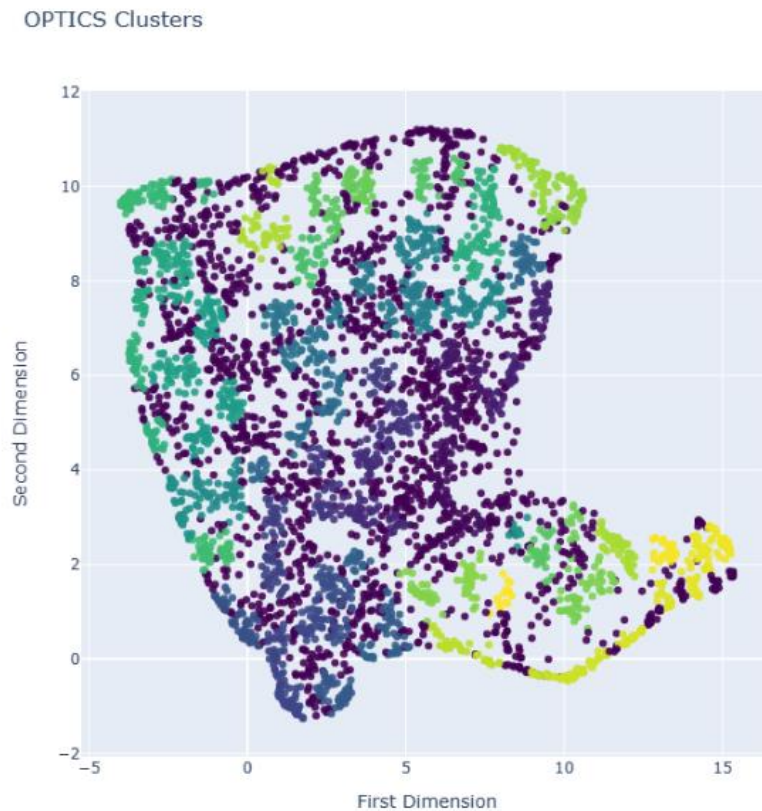


Figure 14. OPTICS Cluster Visualization

Table 2. Clustering Result

Algorithm	Silhouette Score	Davies-Bouldin Index	Execution time in seconds
K-Means	0.445	0.736	1.063
DBSCAN	0.132	1.435	0.032
OPTICS	0.132	1.435	6.272

Although DBSCAN and OPTICS have similar Silhouette scores and Davies-Bouldin Index values, their visualizations reveal different clustering structures due to the inherent differences in their algorithms.

DBSCAN clusters appear to be more scattered and numerous, with many small clusters and a significant number of noise points. While OPTICS also identifies numerous clusters, the clusters appear more contiguous and less fragmented compared to DBSCAN.

This can be explained by a fixed density criterion defined by `eps` and `min_samples` in DBSCAN. Therefore, points are either part of a cluster or classified as noise. OPTICS uses a reachability distance to determine cluster membership, which allows it to identify clusters of varying densities more effectively. This results in smoother transitions and more cohesive clusters, even in regions with varying point densities.

These visual differences highlight the importance of considering not just numerical evaluation metrics but also visual inspection and understanding of the clustering algorithms' mechanisms when choosing the best method.

## Chapter 5. Conclusion, Limitations and Future Work

### 5.1 Conclusion

This study conducted a comparative analysis of three clustering algorithms—K-Means, DBSCAN, and OPTICS—to evaluate their effectiveness in customer segmentation for improved marketing strategies. The findings from the analysis provide valuable insights into the strengths and limitations of each algorithm in handling customer data with varying characteristics.

K-Means effectively grouped customer data into four distinct clusters with clear boundaries. A Silhouette score and Davies-Bouldin Index indicate good clustering performance. However, K-Means requires the number of clusters to be predefined and assumes spherical clusters, limiting its flexibility.

DBSCAN quickly identified clusters of arbitrary shapes with a training time of 0.032 seconds. However, it produced a low silhouette score and a higher Davies-Bouldin Index. The clusters were more scattered, and many noise points were detected, reflecting DBSCAN's sensitivity to density variations.

OPTICS handled varying densities well, identifying both dense and sparse clusters. Performance of OPTICS on both evaluation parameters indicate moderate clustering quality. OPTICS required more computation time but produced smoother, more cohesive clusters compared to DBSCAN.

To conclude, K-Means showed the best cluster separation, while DBSCAN and OPTICS highlighted the importance of density-based approaches in handling noise and varying densities.

Significant business intelligence can be gained by analysing customers of each cluster by personnel with domain knowledge. This will inform businesses about the consumer behaviour and help improving marketing approaches for different categories of customers identified based on their respective clusters.

## 5.2 Limitations and Future Work

This study focuses on RFM matrix for customer segmentation. Incorporating more parameters such as demographic or geographic information of customers can improve cluster formations and provide better insights into customer behaviour. Griva et al. (2024) suggest that integrating behavioural and geographic data provides a richer, more comprehensive view of customer segments, potentially leading to better business outcomes.

For a dataset with true labels available, evaluation metrics like Adjusted Rand Index (ARI) or Normalized Mutual Information (NMI) can be incorporated for further improving analysis of DBSCAN and OPTICS clustering. True labels compare predicted clustering against known clustering thereby improving validity of clustering performance.

Moreover, further research into automated and adaptive methods for optimizing clustering parameters such as  $\epsilon$  and  $\text{min\_samples}$  for DBSCAN,  $\xi$  for OPTICS could enhance clustering performance and reduce the reliance on manual tuning.

Finally, Hybrid clustering approaches can be explored that combine the strengths of multiple clustering algorithms to achieve better performance. For example, using K-Means for initial clustering followed by DBSCAN or OPTICS to refine cluster boundaries and handle noise.

## References

Allaoui, M., Kherfi, M.L. and Cheriet, A. (2020) 'Considerably improving clustering algorithms using umap dimensionality reduction technique: A comparative study', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer, pp. 317–325. Available at: [https://doi.org/10.1007/978-3-030-51935-3\\_34](https://doi.org/10.1007/978-3-030-51935-3_34).

Anitha, P. and Patil, M.M. (2022) 'RFM model for customer purchase behavior using K-Means algorithm', *Journal of King Saud University - Computer and Information Sciences*, 34(5), pp. 1785–1792. Available at: <https://doi.org/10.1016/j.jksuci.2019.12.011>.

Arthur, D. and Vassilvitskii, S. (2007) *k-means++: The Advantages of Careful Seeding*.

Bandyopadhyay, S., Thakur, S.S. and Mandal, J.K. (2021) 'Product recommendation for e-commerce business by applying principal component analysis (PCA) and K-means clustering: benefit for the society', *Innovations in Systems and Software Engineering*, 17(1), pp. 45–52. Available at: <https://doi.org/10.1007/s11334-020-00372-5>.

Bhattacharjee, P. and Mitra, P. (2021) 'A survey of density based clustering algorithms', *Frontiers of Computer Science*. Higher Education Press Limited Company. Available at: <https://doi.org/10.1007/s11704-019-9059-3>.

Bushra, A.A. and Yi, G. (2021) 'Comparative Analysis Review of Pioneering DBSCAN and Successive Density-Based Clustering Algorithms', *IEEE Access*, 9, pp. 87918–87935. Available at: <https://doi.org/10.1109/ACCESS.2021.3089036>.

Celebi, M.E., Kingravi, H.A. and Vela, P.A. (2013) 'A comparative study of efficient initialization methods for the k-means clustering algorithm', *Expert Systems with Applications*, 40(1), pp. 200–210. Available at: <https://doi.org/10.1016/j.eswa.2012.07.021>.

Chen Daqing (2015) 'Online Retail. UCI Machine Learning Repository'.

Christy, A.J. *et al.* (2021) 'RFM ranking – An effective approach to customer segmentation', *Journal of King Saud University - Computer and Information Sciences*, 33(10), pp. 1251–1257.

Available at: <https://doi.org/10.1016/j.jksuci.2018.09.004>.

Deloitte (2023) *2023 Global Marketing Trends About the Deloitte CMO Program*.

Deng, D. (2020) 'DBSCAN Clustering Algorithm Based on Density', in *Proceedings - 2020 7th International Forum on Electrical Engineering and Automation, IFEEA 2020*. Institute of

Electrical and Electronics Engineers Inc., pp. 949–953. Available at:

<https://doi.org/10.1109/IFEEA51475.2020.00199>.

Deng, Z. *et al.* (2015) 'A scalable and fast OPTICS for clustering trajectory big data', *Cluster Computing*, 18(2), pp. 549–562. Available at: <https://doi.org/10.1007/s10586-014-0413-9>.

Doğan, O., Ayçin, E. and Bulut, Z.A. (2018) *Customer segmentation by using RFM model and clustering methods: A case study in retail industry*, *International Journal of Contemporary Economics and Administrative Sciences*. Available at: [www.ijceas.com](http://www.ijceas.com).

Firman Ashari, I. *et al.* (2022) *Application of Data Mining with the K-Means Clustering Method and Davies Bouldin Index for Grouping IMDB Movies*, *Journal of Applied Informatics and Computing (JAIC)*. Available at: <http://jurnal.polibatam.ac.id/index.php/JAIC>.

Gialampoukidis, I., Vrochidis, S. and Kompatsiaris, I. (2016) 'A hybrid framework for news clustering based on the DBSCAN-Martingale and LDA', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Verlag, pp. 170–184. Available at: [https://doi.org/10.1007/978-3-319-41920-6\\_13](https://doi.org/10.1007/978-3-319-41920-6_13).

Griva, A. *et al.* (2024) 'A two-stage business analytics approach to perform behavioural and geographic customer segmentation using e-commerce delivery data', *Journal of Decision Systems*, 33(1), pp. 1–29. Available at: <https://doi.org/10.1080/12460125.2022.2151071>.

Hennig Christian *et al.* (2015) *Handbook of Cluster Analysis*. Edited by C. Hennig *et al.* Chapman and Hall/CRC. Available at: <https://doi.org/10.1201/b19706>.

Hozumi, Y. *et al.* (2021) 'UMAP-assisted K-means clustering of large-scale SARS-CoV-2 mutation datasets', *Computers in Biology and Medicine*, 131. Available at: <https://doi.org/10.1016/j.compbio.2021.104264>.

Ikotun, A.M. *et al.* (2023) 'K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data', *Information Sciences*, 622, pp. 178–210. Available at: <https://doi.org/10.1016/j.ins.2022.11.139>.

Kanagala Hari and Krishnaiah Jaya (2016) *A COMPARATIVE STUDY OF K-MEANS, DBSCAN AND OPTICS*.

Kansal, T. *et al.* (2018) 'Customer Segmentation using K-means Clustering', in *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*. IEEE, pp. 135–139. Available at: <https://doi.org/10.1109/CTEMS.2018.8769171>.

Kim, J.H. *et al.* (2019) 'A fast algorithm for identifying density-based clustering structures using a constraint graph', *Electronics (Switzerland)*, 8(10). Available at: <https://doi.org/10.3390/electronics8101094>.

Makwana, P. and Kodinariya, T.M. (2013) 'Review on Determining of Cluster in K-means Clustering 2000 + citations Review on determining number of Cluster in K-Means Clustering', *International Journal of Advance Research in Computer Science and Management Studies*, 1(6). Available at: <https://www.researchgate.net/publication/313554124>.

McInnes, L., Healy, J. and Melville, J. (2018) 'UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction'. Available at: <http://arxiv.org/abs/1802.03426>.

Monalisa, S. *et al.* (2023) 'Customer segmentation with RFM models and demographic variable using DBSCAN algorithm', *Telkomnika (Telecommunication Computing Electronics and Control)*, 21(4), pp. 742–749. Available at: <https://doi.org/10.12928/TELKOMNIKA.v21i4.22759>.

ŞAHİNBAŞ, K. (2022) 'Performance Comparison of K-Means and DBSCAN Methods for Airline Customer Segmentation', *Black Sea Journal of Engineering and Science*, 5(4), pp. 158–165. Available at: <https://doi.org/10.34248/bsengineering.1170943>.

Saxena, A. *et al.* (2024) 'Examination of the Criticality of Customer Segmentation Using Unsupervised Learning Methods', *Circular Economy and Sustainability* [Preprint]. Available at: <https://doi.org/10.1007/s43615-023-00336-4>.

Sembiring Brahmana, R.W., Mohammed, F.A. and Chairuang, K. (2020a) 'Customer Segmentation Based on RFM Model Using K-Means, K-Medoids, and DBSCAN Methods', *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, 11(1), p. 32. Available at: <https://doi.org/10.24843/lkjiti.2020.v11.i01.p04>.

Sembiring Brahmana, R.W., Mohammed, F.A. and Chairuang, K. (2020b) 'Customer Segmentation Based on RFM Model Using K-Means, K-Medoids, and DBSCAN Methods', *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, 11(1), p. 32. Available at: <https://doi.org/10.24843/lkjiti.2020.v11.i01.p04>.

Shah, D. and Murthi, B.P.S. (2021) 'Marketing in a data-driven digital world: Implications for the role and scope of marketing', *Journal of Business Research*, 125, pp. 772–779. Available at: <https://doi.org/10.1016/j.jbusres.2020.06.062>.

Shahadat Hossain (2017) 'Customer Segmentation using Centroid Based and Density Based Clustering Algorithms', *International Conference on Electrical Information and Communication Technology (EICT)* [Preprint].

Shahapure, K.R. and Nicholas, C. (2020) 'Cluster quality analysis using silhouette score', in *Proceedings - 2020 IEEE 7th International Conference on Data Science and Advanced Analytics, DSAA 2020*. Institute of Electrical and Electronics Engineers Inc., pp. 747–748. Available at: <https://doi.org/10.1109/DSAA49011.2020.00096>.

Shen, J. *et al.* (2016) 'Real-Time Superpixel Segmentation by DBSCAN Clustering Algorithm', *IEEE Transactions on Image Processing*, 25(12), pp. 5933–5942. Available at: <https://doi.org/10.1109/TIP.2016.2616302>.

Sinaga, K.P. and Yang, M.S. (2020) 'Unsupervised K-means clustering algorithm', *IEEE Access*, 8, pp. 80716–80727. Available at: <https://doi.org/10.1109/ACCESS.2020.2988796>.

Subudhi, S. and Panigrahi, S. (2022) 'Application of OPTICS and ensemble learning for Database Intrusion Detection', *Journal of King Saud University - Computer and Information Sciences*, 34(3), pp. 972–981. Available at: <https://doi.org/10.1016/j.jksuci.2019.05.001>.

Zhang, J. *et al.* (2020) 'Wireless Channel Propagation Scenarios Identification: A Perspective of Machine Learning', *IEEE Access*. Institute of Electrical and Electronics Engineers Inc., pp. 47797–47806. Available at: <https://doi.org/10.1109/ACCESS.2020.2979220>.