RFM Analysis on Retail Data

OUTLINE

- Abstract
- Introduction
- Problem Statement
- Motivation
- Objective and Scope
- Literature Survey
- Resource Requirements
- Existing & Proposed System
- Architecture
- Modules
- UML Diagrams
- Data Preprocessing
- Model Implementation
  - ✓ Algorithm Definitions
  - ✓ Code Description
  - ✓ Results from Algorithm
  - ✓ Performance Comparision
- Output Screens
- Future Scope
- Conclusion

ABSTRACT

This project applies machine learning techniques to perform customer segmentation using retail transaction data. The dataset consists of 541,909 records with 8 features, including invoice number, stock code, quantity, invoice date, unit price, customer ID, and country. The primary aim is to group customers based on their purchasing behavior using various clustering algorithms. K-Means clustering is implemented with both the silhouette score and elbow method to determine the optimal number of clusters. Additionally, DBSCAN and hierarchical clustering are used to explore other segmentation techniques. Customer segmentation is performed different feature sets: Recency, Frequency, and Monetary (RFM), Recency and Monetary (RM), and Frequency and Monetary (FM). This analysis provides insights into customer purchasing patterns and behaviors, offering valuable guidance for targeted marketing and customer relationship management. The project aims to help improve retention, increase sales by leveraging effective customer segmentation strategies.

Keywords: Customer segmentation, K-Means, DBSCAN, hierarchical clustering, silhouette score, elbow method, RFM, RM, FM, machine learning, retail data.

INTRODUCTION

Customer segmentation is a crucial task for businesses aiming to understand their customer base and improve their marketing strategies. By grouping customers based on similar purchasing behavior, businesses can tailor their offerings, increase customer satisfaction, and enhance customer retention. This project applies machine learning techniques to perform customer segmentation using retail transaction data. The data consists of 541,909 records with 8 key features: invoice number, stock code, quantity, invoice date, unit price, customer ID, and country. These features provide valuable insights into customers' purchasing patterns, allowing for the identification of distinct customer segments. The primary objective of this project is to segment customers based on their purchasing behavior, using various clustering algorithms to identify meaningful patterns. To achieve this, the project applies K-Means clustering, a widely used unsupervised learning algorithm, to group customers based on their Recency, Frequency, and Monetary (RFM) metrics, which are key indicators of customer behavior. The silhouette score and elbow method are utilized to determine the optimal number of clusters for the K-Means algorithm, ensuring the segmentation is meaningful and actionable.

PROBLEM STATEMENT

- Customer segmentation is a critical task in retail businesses to tailor marketing strategies, improve customer satisfaction, and increase revenue. Retail businesses often struggle to efficiently identify distinct customer groups based on purchasing behavior due to the complexity and volume of transaction data. Without proper segmentation, businesses may fail to target the right customers with personalized offers, resulting in lower engagement and reduced profitability.
- The challenge is to effectively analyze large-scale retail transaction data and apply appropriate clustering techniques to uncover meaningful customer segments. The dataset contains 541,909 records with various features such as invoice details, product information, and customer data, which need to be processed and clustered to identify groups based on behavior. This project aims to address this challenge by applying clustering algorithms like K-Means, DBSCAN, and hierarchical clustering to segment customers, enabling businesses to make data-driven decisions for targeted marketing, promotions, and customer retention strategies.

MOTIVATION

The motivation behind this project stems from the growing need for businesses, especially in the retail sector, to better understand and engage their customers. Traditional marketing approaches often fail to target customers effectively, leading to missed opportunities and suboptimal customer experiences. By leveraging clustering algorithms to segment customers based on purchasing behavior, businesses can identify distinct customer groups and tailor their strategies accordingly. This data-driven approach enables personalized marketing, targeted promotions, and enhanced customer retention, which can significantly improve customer satisfaction and boost sales. The ability to process and analyze large-scale transaction data provides a competitive advantage, allowing companies to make informed decisions and optimize their marketing efforts. This project aims to contribute to the ongoing effort to enhance customer engagement through intelligent and efficient segmentation techniques.

OBJECTIVE OF PROJECT

The objective of this project is to segment retail customers based on their purchasing behavior using clustering algorithms. The dataset, consisting of 541,909 records with 8 attributes, will be analyzed to identify distinct customer groups. The project aims to apply K-Means clustering with both the silhouette score and elbow method to determine the optimal number of clusters. It will also explore DBSCAN and hierarchical clustering to evaluate their effectiveness in segmentation. The segmentation will be performed using various feature sets like Recency, Frequency, and Monetary (RFM), Recency and Monetary (RM), and Frequency and Monetary (FM) to understand different aspects of customer behavior. The main goal is to provide actionable insights that can help businesses optimize marketing strategies, enhance customer retention, and deliver personalized promotions, ultimately boosting customer engagement and business performance.

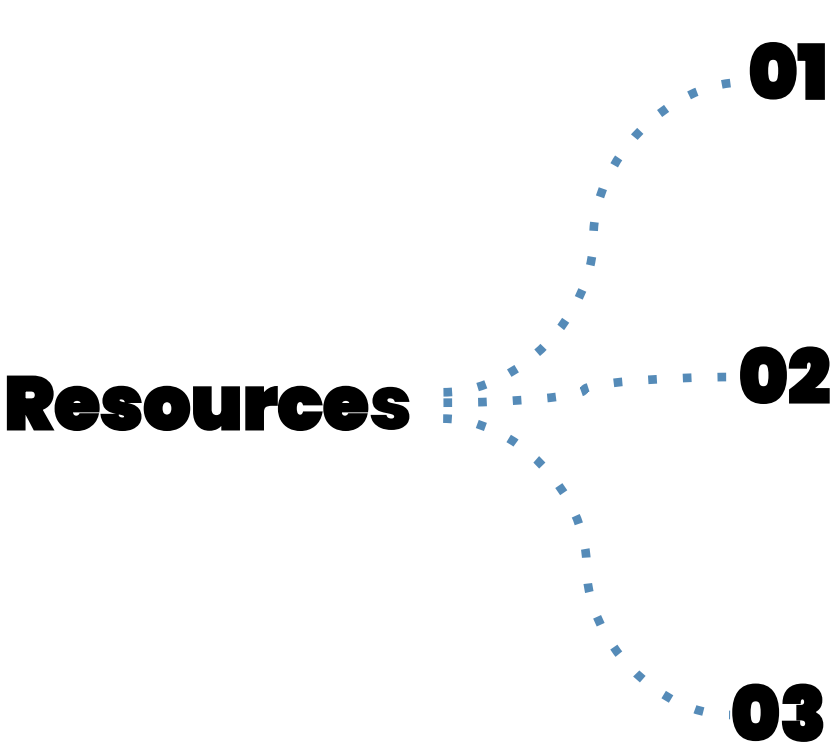
SCOPE

The scope of this project is to apply clustering algorithms on retail transaction data to identify meaningful customer segments based on purchasing behavior. The project will explore clustering techniques such as K-Means, DBSCAN, and hierarchical clustering to segment customers based on Recency, Frequency, and Monetary (RFM) metrics, as well as variations of these metrics. The project will implement the silhouette score and elbow method for determining the optimal number of clusters, followed by evaluating the effectiveness of different algorithms. The final goal is to provide businesses with actionable insights for personalized marketing, customer retention strategies, and optimized promotions, enabling data-driven decision-making for improving customer engagement and business performance.

LITERATURE SURVEY

Author(s)	Year	Title of Study	Objective	Methodology	Findings/Conclusion
Kumar et al.	2020	Customer Segmentation for Targeted Marketing Using K-Means Clustering	To segment retail customers based on purchasing behavior for targeted marketing strategies.	Applied K-Means clustering on transaction data to classify customers based on purchasing frequency and monetary value.	Found that K-Means effectively identified high-value customers, leading to increased targeted promotions.
Gupta and Sharma	2019	Exploring DBSCAN for Customer Segmentation in E-commerce	To explore DBSCAN's effectiveness in segmenting customers based on behavior in an online retail setup.	Used DBSCAN to find density-based clusters in customer data, focusing on purchase behavior and transaction volume.	DBSCAN outperformed K-Means in detecting customer segments with varied purchasing behavior.
Lee et al.	2021	Hierarchical Clustering for Market Basket Analysis	To explore hierarchical clustering for analyzing customer buying patterns and market basket analysis.	Implemented hierarchical clustering on product-level data to identify product affinity and customer segments.	Hierarchical clustering was found useful in discovering hidden product affinities and improving cross-selling.
Zhang and Li	2018	Segmentation of Retail Customers Using RFM Model	To perform customer segmentation using the RFM model for a retail business.	Combined K-Means clustering with Recency, Frequency, and Monetary (RFM) metrics to create customer segments.	RFM-based segmentation enabled the company to target customers with high engagement and loyalty.
Patel et al.	2017	Application of Clustering Algorithms in Retail Business	To apply various clustering algorithms (K-Means, DBSCAN, and hierarchical) to retail transaction data.	Analyzed customer purchasing behavior using different clustering techniques and evaluated their performance.	K-Means was most effective for large datasets, while DBSCAN was better at identifying outliers in smaller segments.

Resource Requirements



Hardware

Processor: I5/Intel Processor, **RAM:** 8GB (min), **Hard Disk:** 128 GB**, Key Board:** Standard Windows Keyboard, **Mouse:** Two or Three Button Mouse)

Software

(**Operating System:** Windows 10, **Programming Language:** Python 3.10.8, **IDE:** VS Code)

Packages/Libraries

Sklearn, Pandas, Seaborn, numpy

FUNCTIONAL & NON-FUNCTIONAL REQUIREMENTS

- Requirement's analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and non-functional requirements.
- **Functional Requirements**: These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

FUNCTIONAL & NON-FUNCTIONAL REQUIREMENTS

- Examples of functional requirements:
- Authentication of user whenever he/she logs into the system
- System shutdown in case of a cyber-attack
- A verification email is sent to user whenever he/she register for the first time on some software system.
- **Non-functional requirements**: These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.

EXISTING SYSTEM

Existing systems for customer segmentation in retail primarily rely on basic clustering algorithms like K-Means and DBSCAN to segment customers based on purchasing behavior. While K-Means is widely used, it struggles with handling outliers and non-globular clusters. DBSCAN, on the other hand, is better for detecting density-based clusters and outliers but is less effective when dealing with large datasets or high-dimensional data. These systems typically do not use advanced metrics like Recency, Frequency, and Monetary (RFM), and lack real-time processing capabilities. Furthermore, they fail to adapt to evolving customer behaviors, which limits their effectiveness in dynamic retail environments.

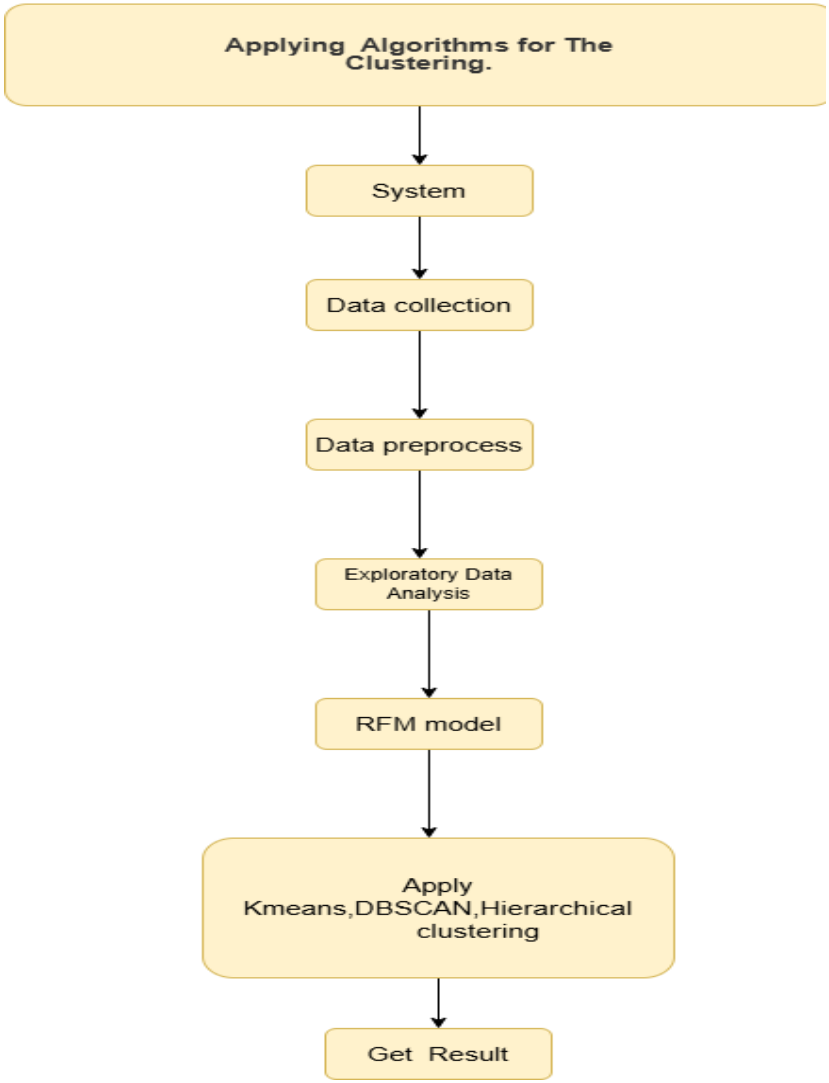
DISADVANTAGES

- **Limited Clustering Techniques**: Existing systems primarily rely on basic clustering algorithms like K-Means and DBSCAN, which may not capture all complex customer patterns effectively.
- **Handling Outliers**: K-Means struggles to identify and handle outliers, leading to less accurate segmentation in some cases.
- **Lack of Real-Time Processing**: Existing systems often fail to handle real-time data, making them less adaptable to dynamic customer behavior and trends.
- **Scalability Issues**: Traditional methods may struggle with large datasets, leading to inefficiencies in processing and segmentation.
- **No Use of Advanced Metrics**: Existing systems often do not incorporate advanced metrics like Recency, Frequency, and Monetary (RFM), limiting the depth of customer analysis.

PROPOSED SYSTEM

- The proposed system improves upon existing methods by incorporating the RFM (Recency, Frequency, and Monetary) model for customer segmentation.
- It applies advanced clustering techniques such as K-Means with silhouette score, DBSCAN, and hierarchical clustering to segment customers based on RFM metrics.
- The system handles large datasets, processes data in real-time, and performs feature engineering to enhance segmentation accuracy.
- This approach provides businesses with more precise, actionable insights for targeted marketing, personalized promotions, and customer retention strategies, improving overall business performance.

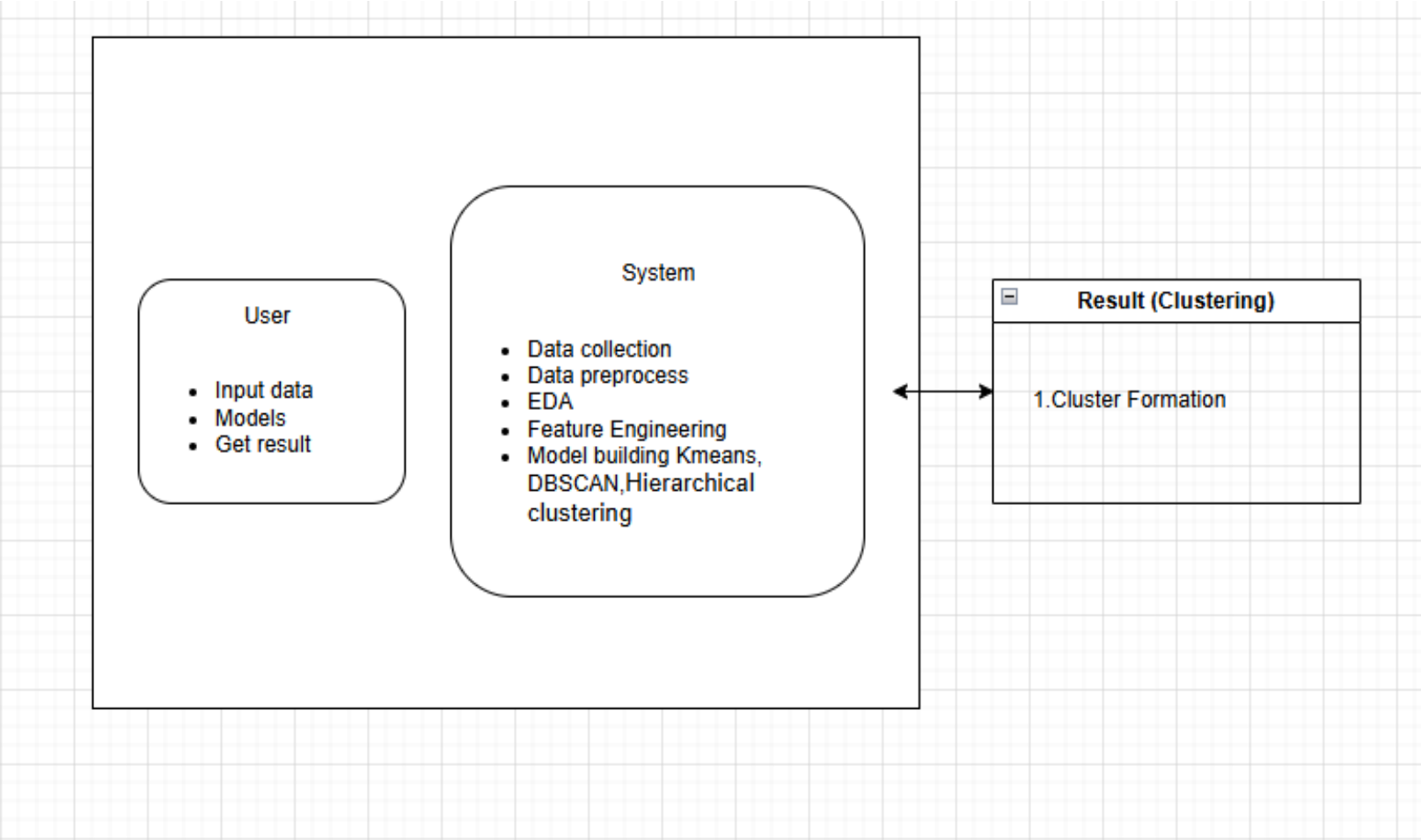
BLOCK DIAGRAM



ADVANTAGES

- **Incorporation of RFM Metrics**: The system leverages the Recency, Frequency, and Monetary (RFM) model, providing a deeper and more meaningful segmentation of customers.
- **Advanced Clustering Techniques**: Uses K-Means with silhouette score, DBSCAN, and hierarchical clustering to handle complex customer behaviors more effectively.
- **Real-Time Processing**: The system can process data in real-time, making it adaptable to dynamic customer behaviors and providing up-to-date insights.
- **Scalability**: The proposed system can efficiently handle large datasets, ensuring fast and accurate segmentation across big data.
- **Accurate Customer Segmentation**: By integrating feature engineering and advanced clustering, the system provides more precise and actionable customer segments for targeted marketing and retention.
- **Flexibility**: The system is designed to evolve and adapt to changing customer behaviors over time, offering more relevant and timely insights for businesses.

SYSTEM ARCHITECTURE



MODULES

- **System Overview**
- 1.1 **Data Collection**

  Retail transaction data includes Invoice Number, Stock Code, Quantity, Invoice Date, Unit Price, Customer ID, and Country, with 541,909 records. Data preparation involves addressing missing values, ensuring data consistency, and standardizing formats.
- 1.2 **Data Preprocessing**

  Missing values are dropped, and outliers are handled. Numerical features are standardized using MinMax or Standard Scaling. Categorical variables are encoded using One-Hot or Label Encoding.
- 1.3 **Exploratory Data Analysis (EDA)**

  EDA includes statistical summaries, distribution plots, and visualizations to detect patterns, relationships, and anomalies in the dataset. Scatter plots and heatmaps help understand customer behavior.

MODULES

- **Total Amount**: Quantity × Unit Price
- **Date Splitting**: Year, Month, Day from Invoice Date
- **RFM Metrics**: Recency (R), Frequency (F), Monetary (M)
- These features enhance customer segmentation.
- 1.4 **Feature Engineering**

  New features are created:
- **Total Amount**: Quantity × Unit Price
- **Date Splitting**: Year, Month, Day from Invoice Date
- **RFM Metrics**: Recency (R), Frequency (F), Monetary (M)
- These features enhance customer segmentation.

MODULES

- 1.5 **RFM Model Creation**
  RFM metrics segment customers based on purchasing behavior, with Recency, Frequency, and Monetary values computed for each customer.
- 1.6 **Model Training and Segmentation**
  Clustering models are trained on:
- Recency and Monetary (RM)
- Frequency and Monetary (FM)
- Recency, Frequency, and Monetary (RFM)
- Clustering methods include:
- **K-Means** with Elbow and Silhouette Score
- **DBSCAN** and **Hierarchical Clustering**

System Architecture using UML Diagrams of RFM

- Unified Modelling Language (UML) is a general-purpose modelling language. The main sim of UML is to define a standard way to visualize the way a system has been designed.
- UML is quite similar to blueprints used in other fields of engineering. UML is not a programming language, it is rather a visual language.
- We use UML diagrams to show the behaviour and structure of a system.

- TYPES OF UML DIAGRAMS
  - USE CASE DIAGRAM
  - CLASS DIAGRAM
  - SEQUENCE DIAGRAM
  - COLLABORATION DIAGRAM
  - ACTIVITY DIAGRAM
  - COMPONENT DIAGRAM
  - STATE CHART DIAGRAM
  - DEPLOYMENT DIAGRAM

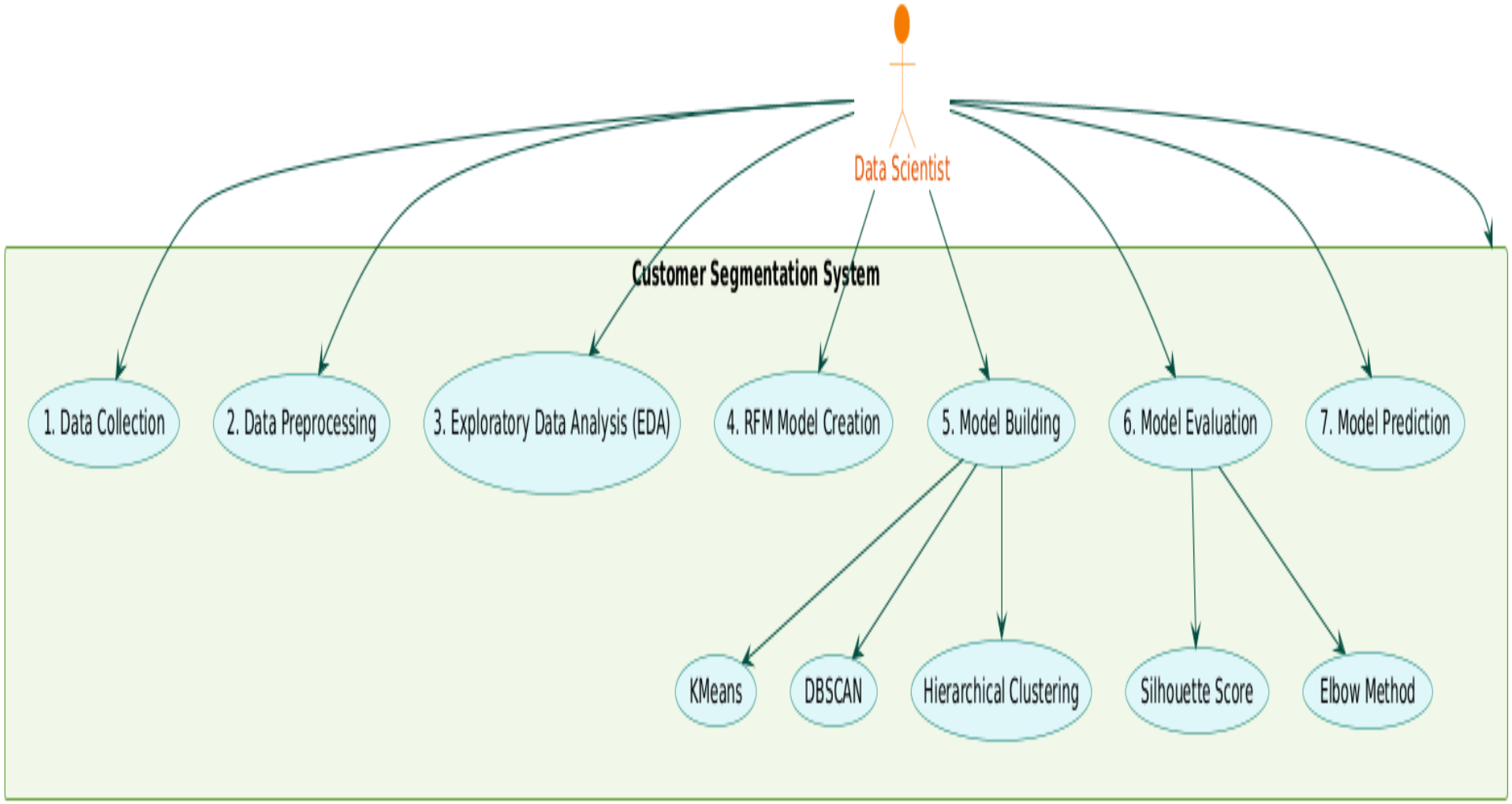
USE CASE DIAGRAM OF RFM

- A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis.

- Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

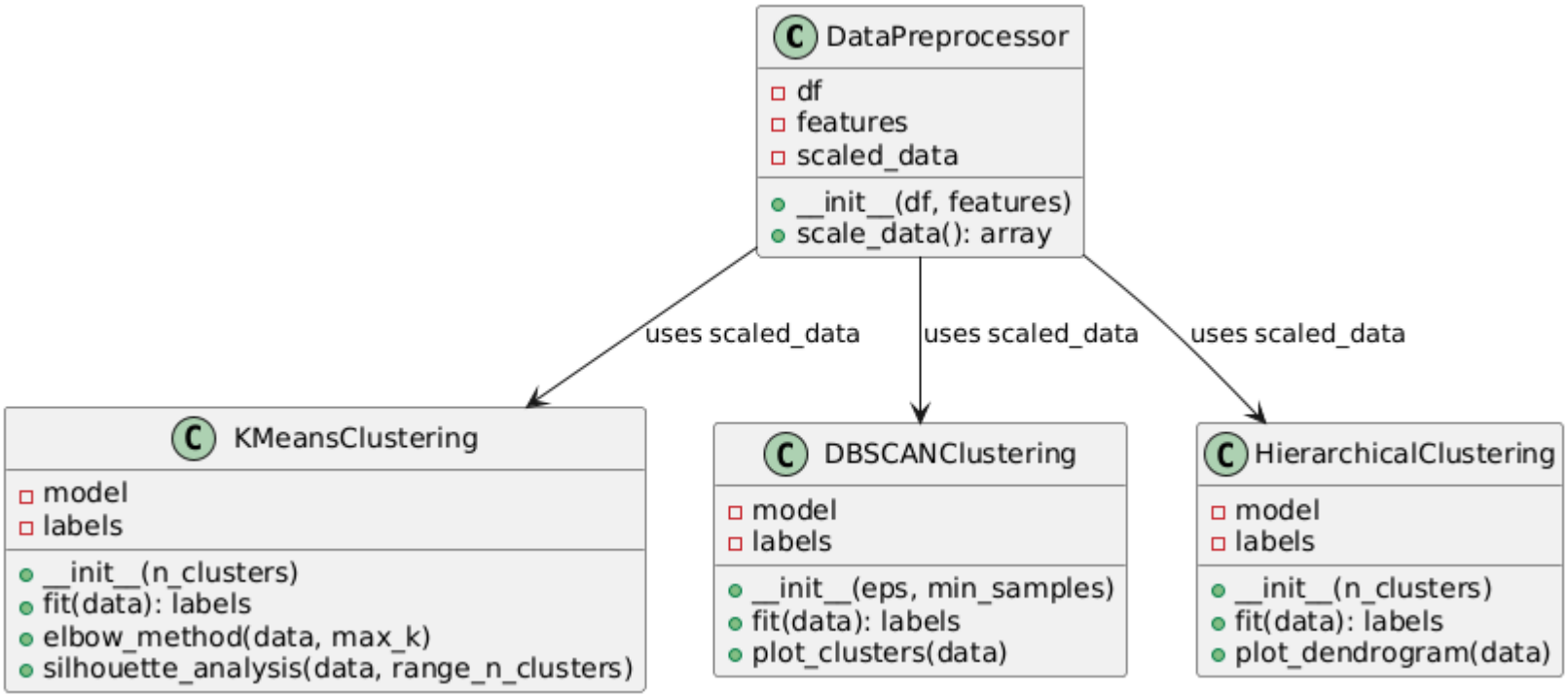
- The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

USE CASE DIAGRAM OF RFM



CLASS DIAGRAM OF RFM

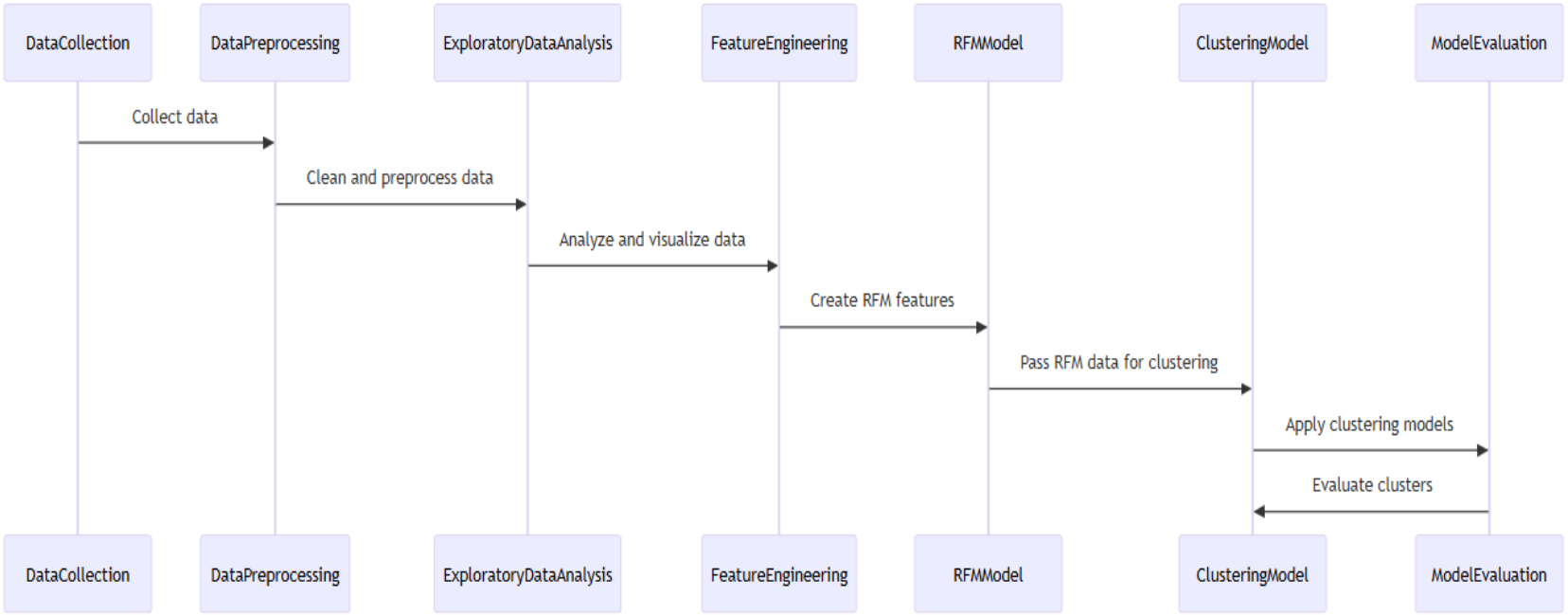
- In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



SEQUENCE DIAGRAM OF RFM

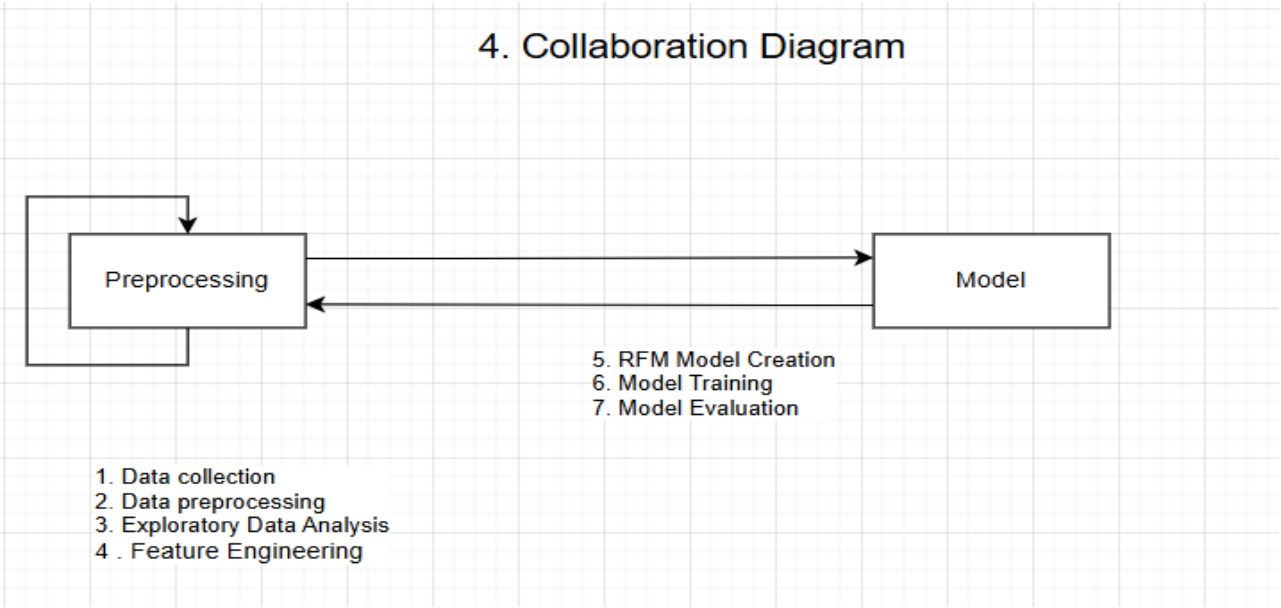
- A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order.
- It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams

SEQUENCE DIAGRAM OF RFM



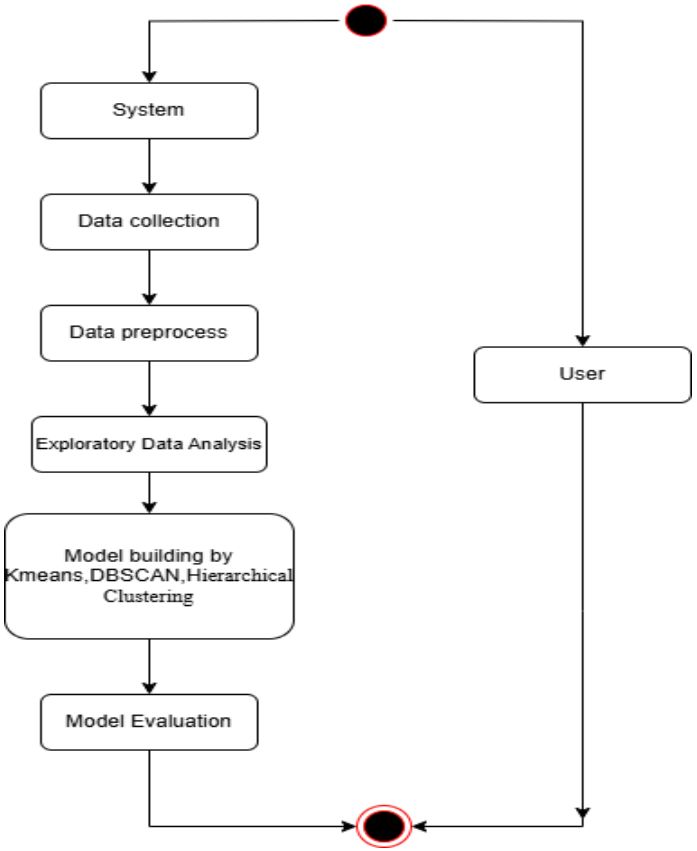
COLLABORATION DIAGRAM OF RFM

- In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization



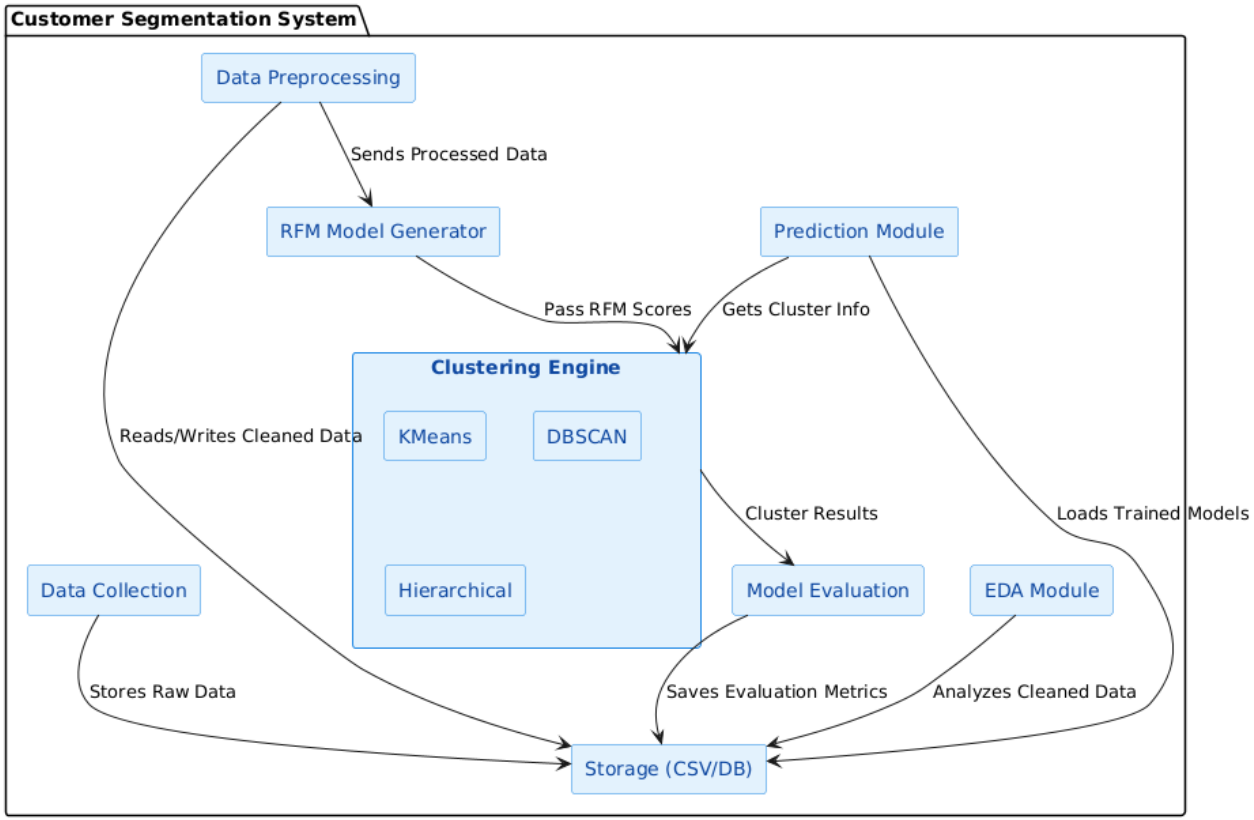
ACTIVITY DIAGRAM OF RFM

- Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

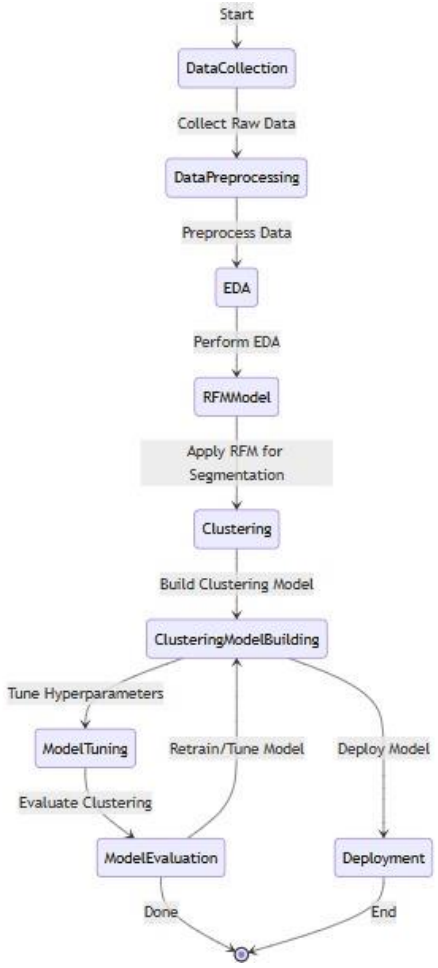


COMPONENT DIAGRAM OF RFM

- A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical **c**omponents in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development.



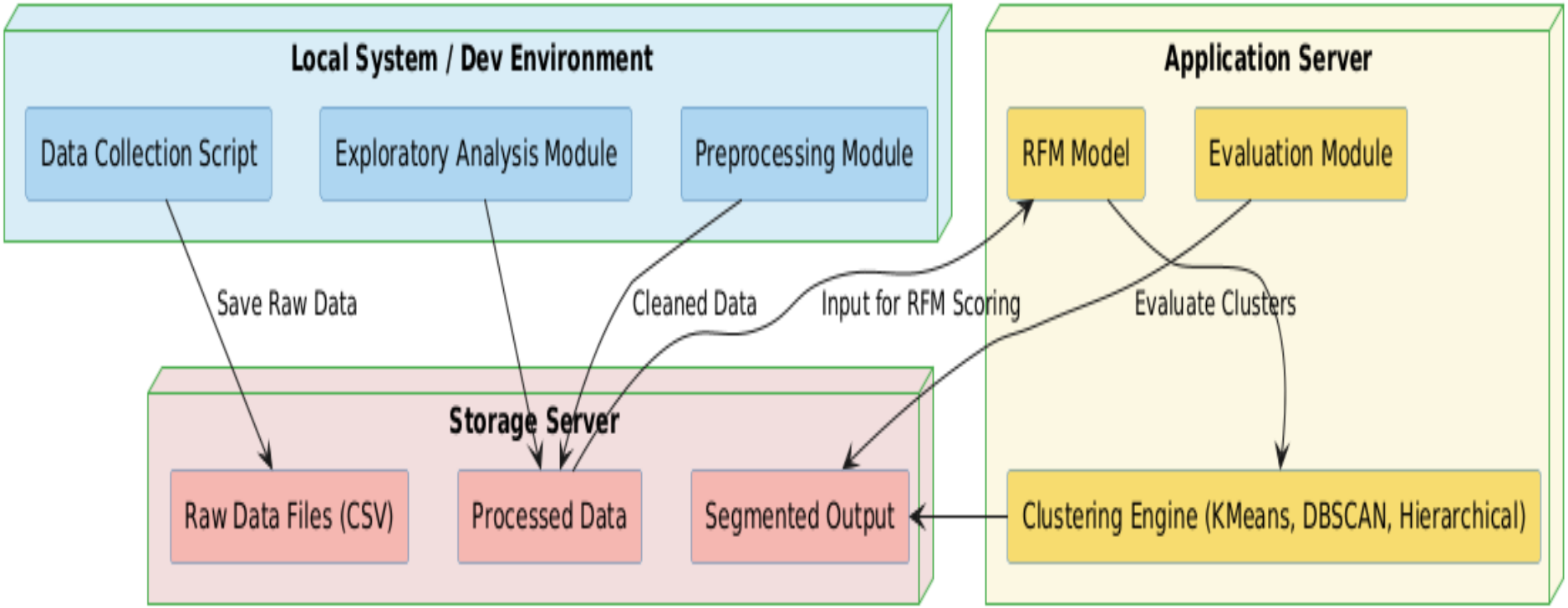
State Chart Diagram of RFM



- **State Chart Diagram** (also called a **State Machine Diagram**) shows the **dynamic behavior** of a system by modeling its **states**, **transitions**, and **events**. It is especially useful to model **reactive systems**, like objects that change states based on events.
- Key Components
  - **State**: Represents a specific condition or situation of an object
  - **Initial State**: The starting point of the object's lifecycle, shown as a filled black circle
  - **Final State**: Indicates the end of the object's lifecycle, shown as a circle with a dot inside.
  - **Transition**: A directed arrow showing movement from one state to another.

DEPLOYMENT DIAGRAM OF RFM

- Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.



Data Pre-processing

- **Data Cleaning:** Remove or impute missing values.
- **Feature Selection:** Choose relevant features impacting hospital stay.
- **Data Transformation:** Normalize or standardize numerical data.
- **Encoding Categorical Data:** Use one-hot encoding or label encoding for categorical variables or map.
- **Data Splitting:** Divide the dataset into training and testing sets.
- **Feature Engineering:** Create new features that might improve model performance.

Code Snippets for Pre-processing

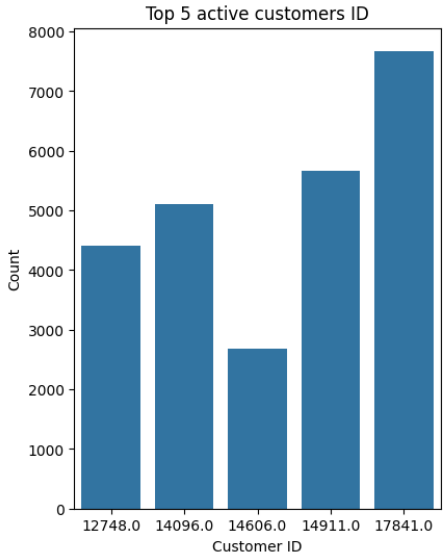
Preprocessing the dataset

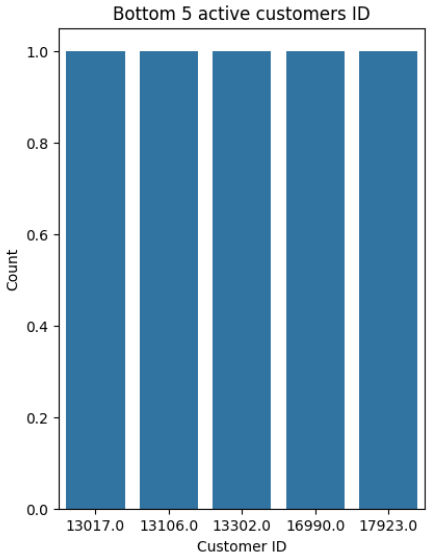
Preprocessing of a dataset refers to the steps taken to clean and prepare the data for analysis or modeling. It is a critical part of the data science workflow because raw data is often incomplete, inconsistent, or not in the right format for machine learning algorithms to perform well. The goal of preprocessing is to improve the quality of the data and make it more suitable for analysis. Here are common steps involved in data preprocessing:

```python
# check null values
data.isna().sum()
```

Python

```
InvoiceNo          0
StockCode          0
Description     1454
Quantity           0
InvoiceDate        0
UnitPrice          0
CustomerID     135080
Country            0
dtype: int64
```






```python
# droping false transactions
df = data[~data['InvoiceNo'].str.contains('C', na=False)]
```

```python
df.shape
```

```
(392732, 8)
```

Try to convert invoice date into year,month,day,hour like way

```python
df['InvoiceDate_year'] = df['InvoiceDate'].dt.year
df['InvoiceDate_month'] = df['InvoiceDate'].dt.month
df['InvoiceDate_day'] = df['InvoiceDate'].dt.day
df['InvoiceDate_hour'] = df['InvoiceDate'].dt.hour
df['InvoiceDate_minute'] = df['InvoiceDate'].dt.minute
df['InvoiceDate_second'] = df['InvoiceDate'].dt.second
```

Algorithm Definition & Methods

K-Means Clustering

K-Means is a centroid-based algorithm that partitions data into a predefined number of clusters. The algorithm assigns each data point to the nearest cluster centroid and iteratively updates the centroids by calculating the mean of the points assigned to each cluster. This process continues until the centroids no longer change significantly. K-Means is efficient and works well with spherical clusters and well-separated data. However, it requires specifying the number of clusters beforehand and may struggle with non-spherical or unevenly distributed data..

DBSCAN

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

DBSCAN is a density-based clustering algorithm that groups points in regions of high density while marking points in low-density areas as noise. It doesn’t require the number of clusters to be predefined. Instead, it relies on two parameters: the distance threshold to define a neighborhood and the minimum number of points required to form a cluster. DBSCAN is effective at finding irregularly shaped clusters and is robust to noise, making it suitable for datasets with outliers. However, it can have difficulty when clusters vary in density.

Hierarchical Clustering

Hierarchical Clustering

Hierarchical clustering creates a tree-like structure called a dendrogram, where each data point starts as its own cluster and is merged (agglomerative) or split (divisive) based on a similarity measure. The algorithm doesn’t require the number of clusters to be defined upfront, making it flexible in capturing clusters of various sizes and shapes. However, it can be computationally expensive for large datasets. The results can be visualized using a dendrogram, which shows how clusters are merged or split. Hierarchical clustering is useful when the relationships between clusters are important to explore.

Determing Optimal Number of Clusters using

Elbow Method

The Elbow Method is a graphical technique used to find the optimal number of clusters in a dataset. It involves plotting the within-cluster sum of squares (WSS) against the number of clusters (k). As the number of clusters increases, the WSS decreases because the data points within each cluster become closer to their respective centroids. However, after a certain point, the rate of decrease slows down and forms an "elbow" shape on the graph. The "elbow" point is considered the optimal number of clusters, as it represents the point where adding more clusters no longer significantly improves the clustering quality.

Silhouette Score

Silhouette Score

The Silhouette Score is a metric used to assess how well-separated the clusters are. It measures both the cohesion (how similar the points within a cluster are to each other) and separation (how distinct the clusters are from each other). The score ranges from -1 to 1:

•A score closer to **+1** indicates that the clusters are well-separated and the points are well-clustered.

•A score close to **0** suggests that the clusters are overlapping.

•A score close to **-1** indicates poor clustering, where points might be assigned to the wrong clusters.

In summary, the Elbow Method helps determine the ideal number of clusters, while the Silhouette Score evaluates the quality of the clustering by measuring how well-separated and cohesive the clusters are. Both techniques are useful for assessing the effectiveness of clustering algorithms.

RFM MODEL

CustomerID	Recency	Frequency	Monetary	R	F	M	RFMGroup	RFMScore	Recency_log	Frequency_log	Monetary_log
12346.0	325	1	77183.60	4	4	1	441	9	5.783825	0.000000	11.253942
12347.0	2	182	4310.00	1	1	1	111	3	0.693147	5.204007	8.368693
12348.0	75	31	1797.24	3	3	1	331	7	4.317488	3.433987	7.494007
12349.0	18	73	1757.55	2	2	1	221	5	2.890372	4.290459	7.471676
12350.0	310	17	334.40	4	4	3	443	11	5.736572	2.833213	5.812338
...	...	...	...	...	...	...	...	...	...	...	...
18280.0	277	10	180.60	4	4	4	444	12	5.624018	2.302585	5.196285
18281.0	180	7	80.82	4	4	4	444	12	5.192957	1.945910	4.392224
18282.0	7	12	178.05	1	4	4	144	9	1.945910	2.484907	5.182064
18283.0	3	721	2045.53	1	1	1	111	3	1.098612	6.580639	7.623412
18287.0	42	70	1837.28	2	2	1	221	5	3.737670	4.248495	7.516041

4339 rows × 11 columns

- The RFM model is a customer segmentation technique that uses three key metrics: Recency, Frequency, and Monetary value. It helps businesses categorize customers based on how recently they made a purchase, how often they purchase, and how much they spend, allowing for targeted marketing and improved customer relationship management.

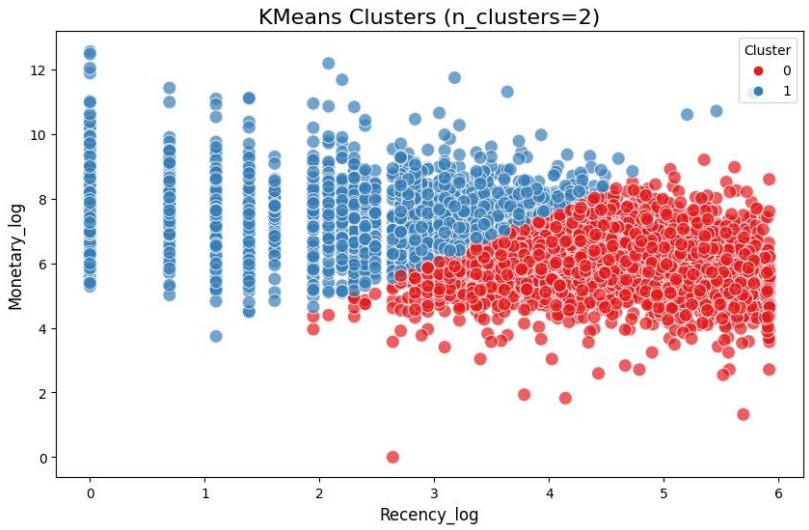
K-Means with silhouette_score | RM |

Code Snippet:

Apply K-Means with silhouette_score | RM |

```
from sklearn.metrics import silhouette_score
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_samples, silhouette_score
from sklearn.cluster import KMeans
features_rec_mon=['Recency_log','Monetary_log']
X_features_rec_mon=rfm_df[features_rec_mon].values
scaler_rec_mon=preprocessing.StandardScaler()
X_rec_mon=scaler_rec_mon.fit_transform(X_features_rec_mon)
X=X_rec_mon
range_n_clusters = [2,3,4,5,6,7,8,9,10,11,12,13,14,15]
for n_clusters in range_n_clusters:
    clusterer = KMeans(n_clusters=n_clusters)
    preds = clusterer.fit_predict(X)
    centers = clusterer.cluster_centers_

    score = silhouette_score(X, preds)
    print("For n_clusters = {}, silhouette score is {}".format(n_clusters, score))
```

Output of Code Snippet:



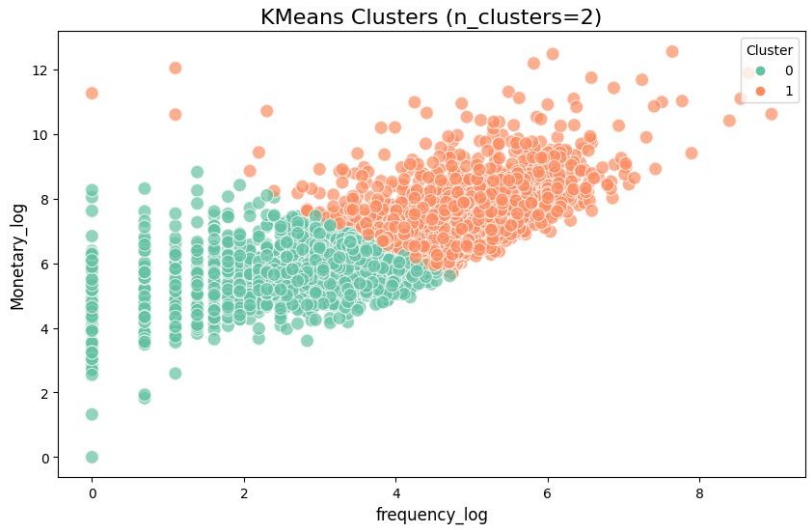
K-Means with silhouette_score | FM |

Code Snippet:

Apply K-Means with silhouette_score | FM |

```
from sklearn.metrics import silhouette_score
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_samples, silhouette_score
from sklearn.cluster import KMeans
features_rec_mon=['Frequency_log','Monetary_log']
X_features_rec_mon=rfm_df[features_rec_mon].values
scaler_rec_mon=preprocessing.StandardScaler()
X_rec_mon=scaler_rec_mon.fit_transform(X_features_rec_mon)
X=X_rec_mon
range_n_clusters = [2,3,4,5,6,7,8,9,10,11,12,13,14,15]
for n_clusters in range_n_clusters:
    clusterer = KMeans(n_clusters=n_clusters)
    preds = clusterer.fit_predict(X)
    centers = clusterer.cluster_centers_

    score = silhouette_score(X, preds)
    print("For n_clusters = {}, silhouette score is {}".format(n_clusters, score))
```

Output of Code Snippet:



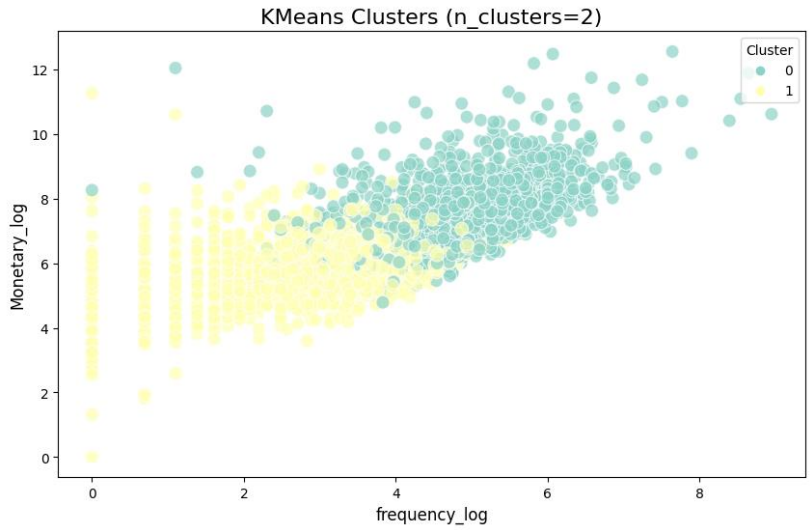
K-Means with silhouette_score | RFM |

Code Snippet:

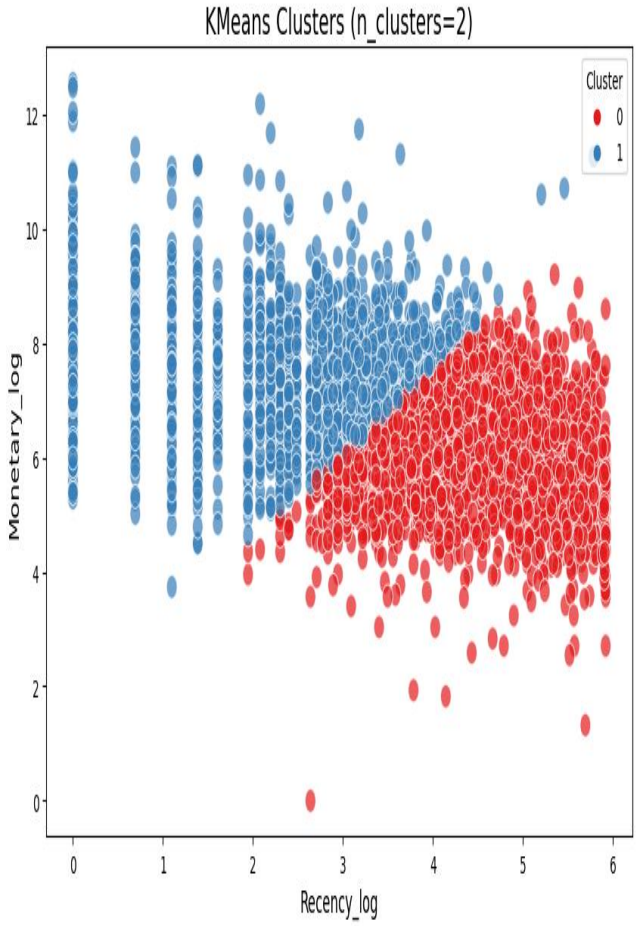
Apply K-Means with silhouette_score | RFM |

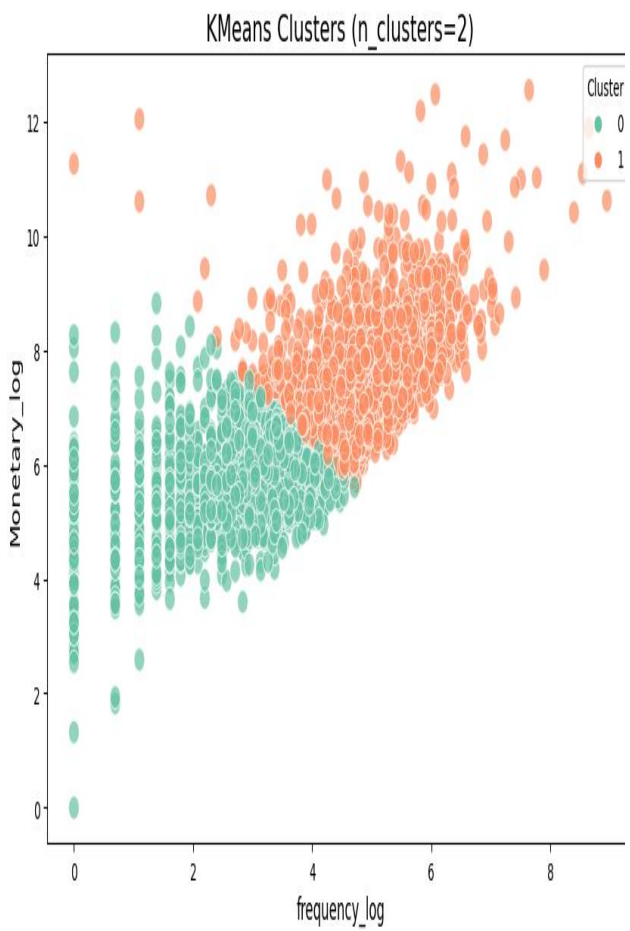
```
from sklearn.metrics import silhouette_score
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_samples, silhouette_score
from sklearn.cluster import KMeans
features_rec_mon=['Recency_log','Frequency_log','Monetary_log']
X_features_rec_mon=rfm_df[features_rec_mon].values
scaler_rec_mon=preprocessing.StandardScaler()
X_rec_mon=scaler_rec_mon.fit_transform(X_features_rec_mon)
X=X_rec_mon
range_n_clusters = [3,2,4,5,6,7,8,9,10,11,12,13,14,15]
for n_clusters in range_n_clusters:
    clusterer = KMeans(n_clusters=n_clusters)
    preds = clusterer.fit_predict(X)
    centers = clusterer.cluster_centers_

    score = silhouette_score(X, preds)
    print("For n_clusters = {}, silhouette score is {}".format(n_clusters, score))
```

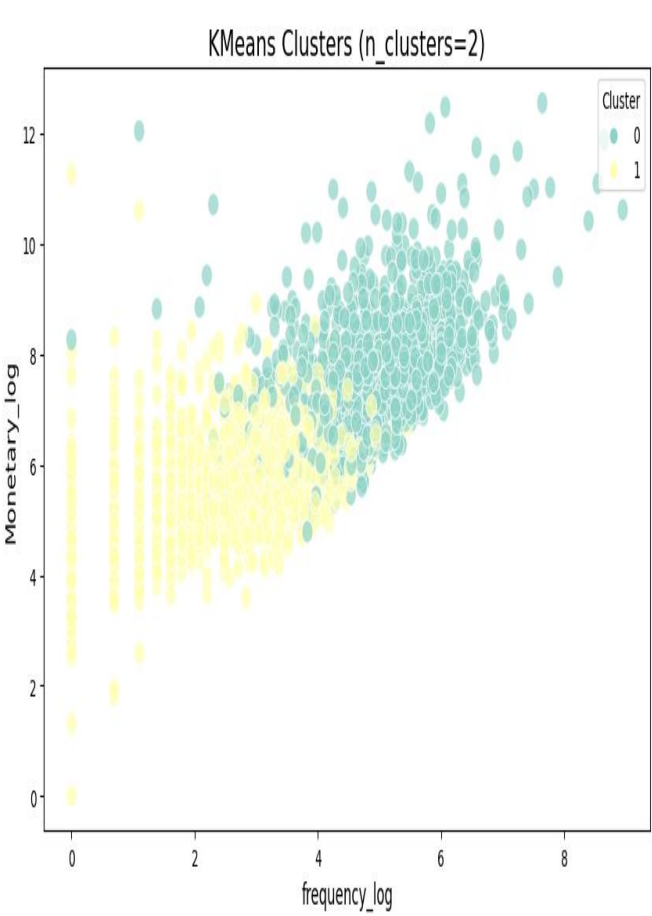
Output with Code Snippet:



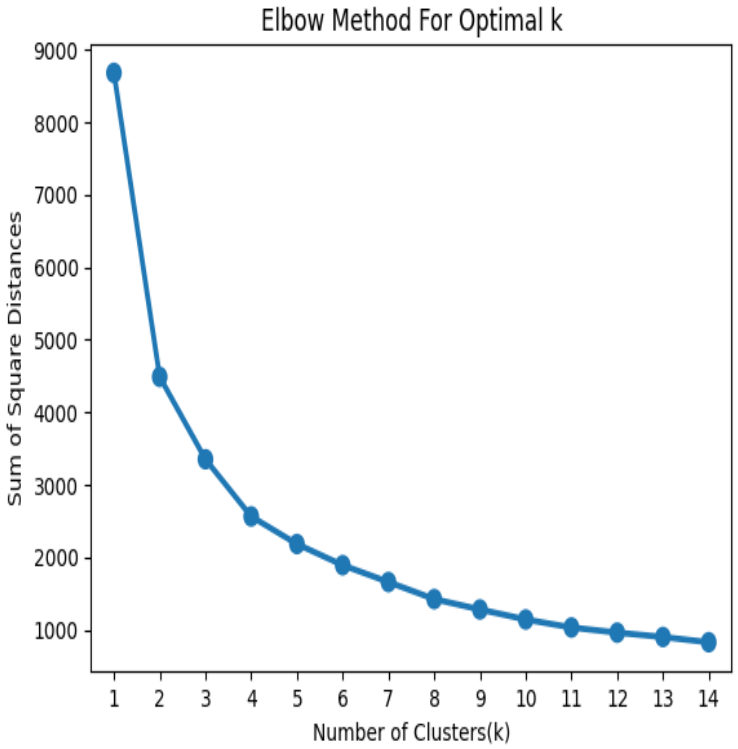
Comparison Between RM, FM and, RFM

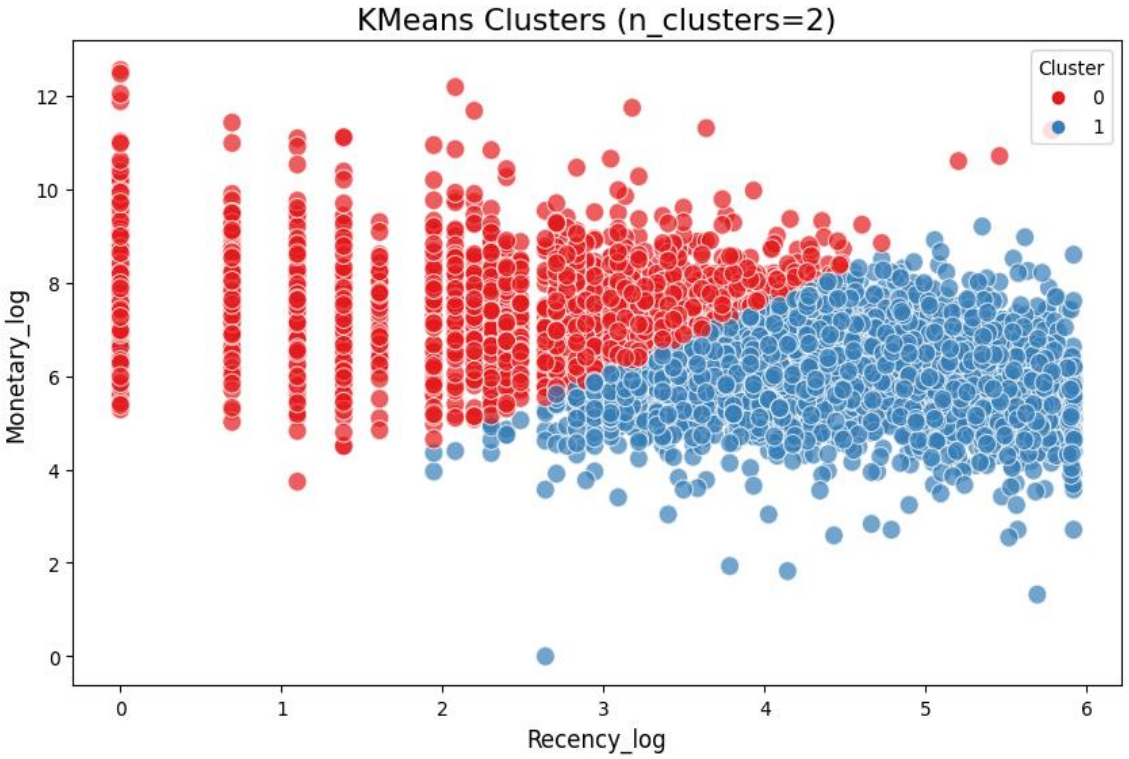




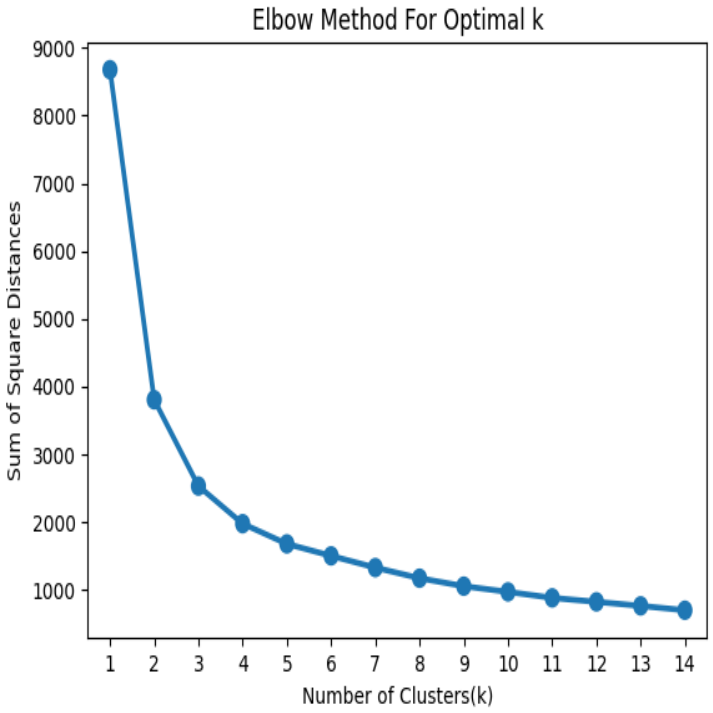


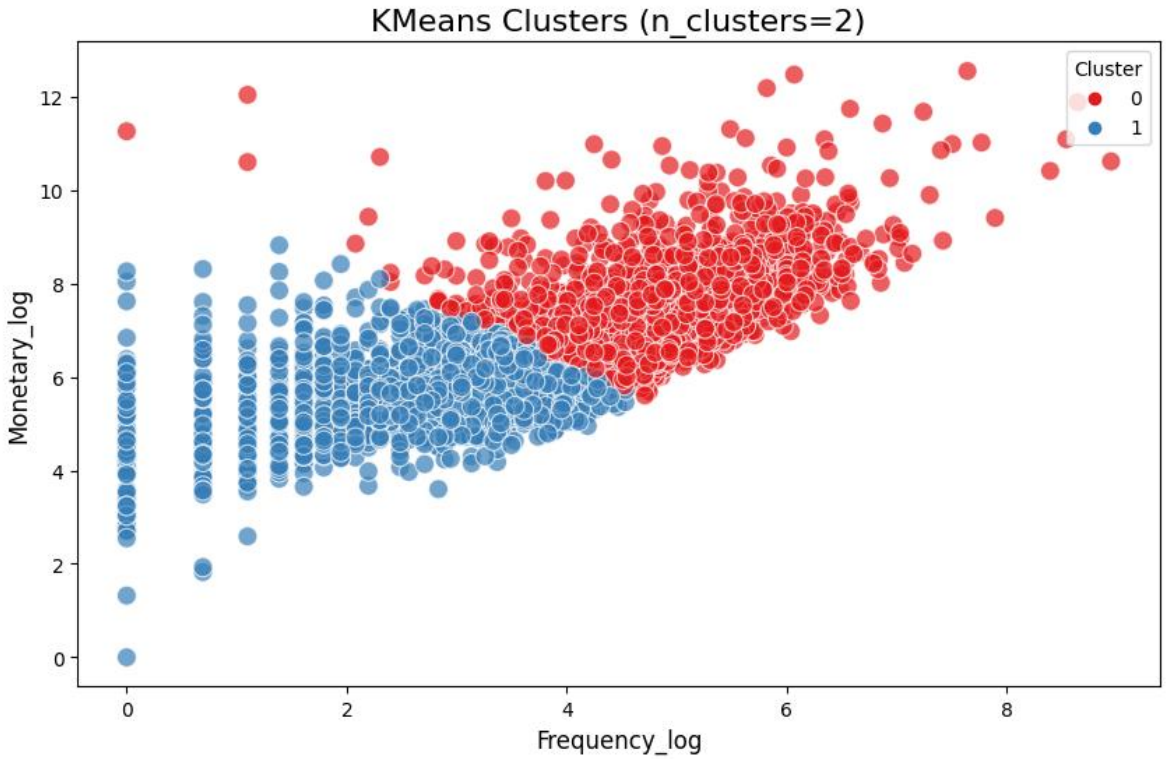
K-Means with Elbow method | RM |



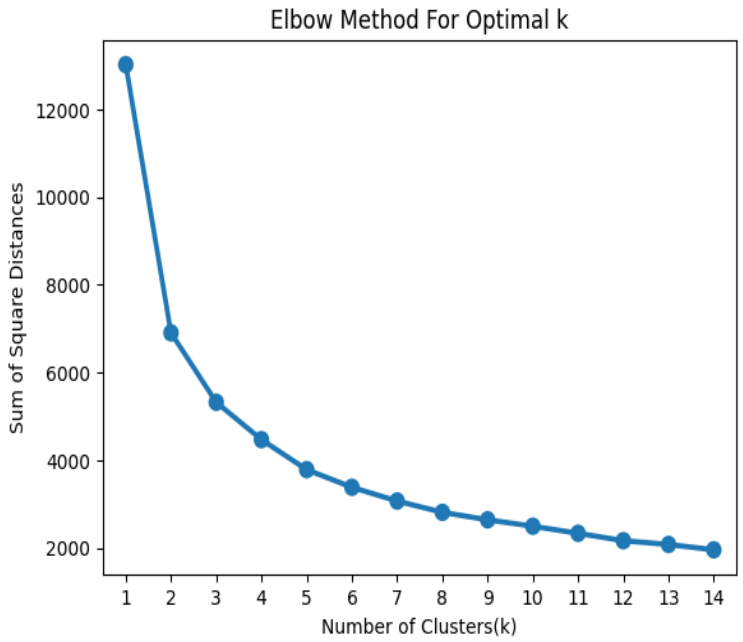


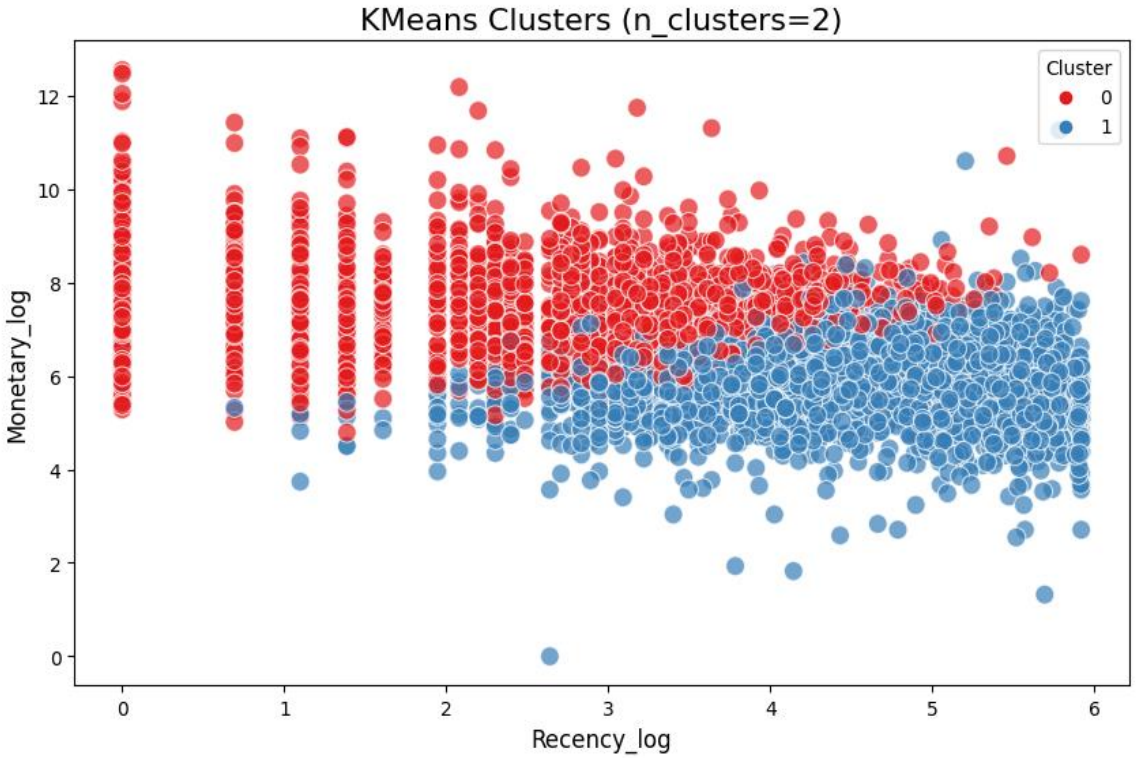
K-Means with Elbow method | FM |





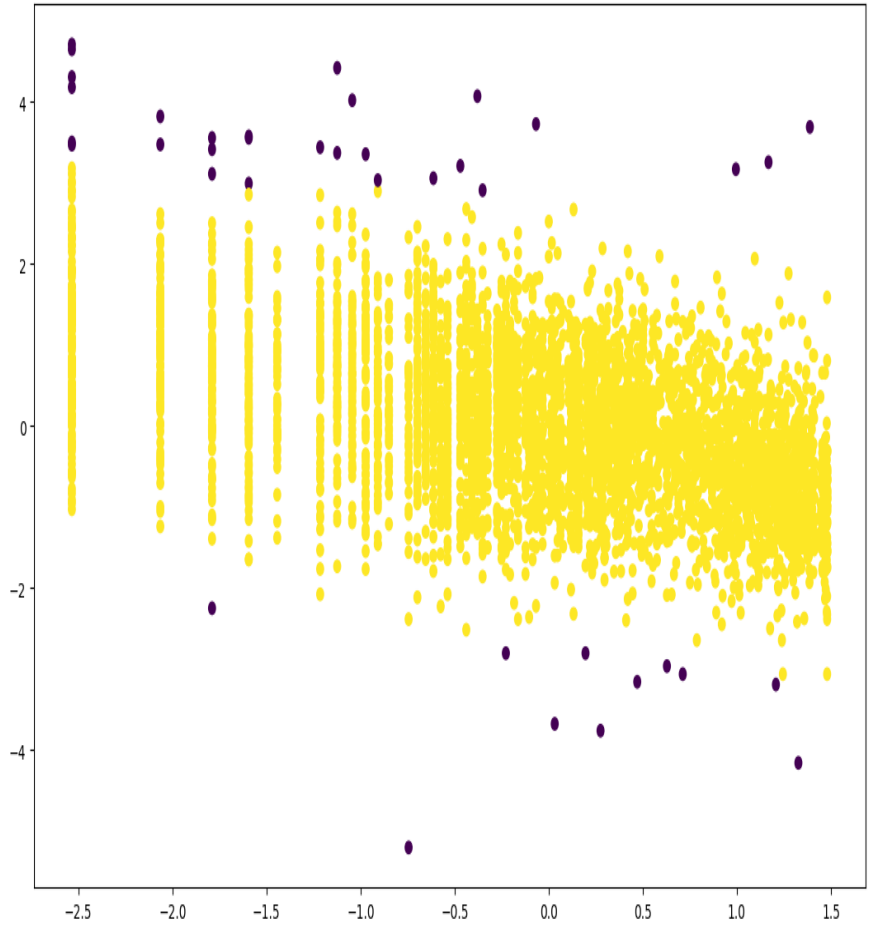
K-Means with Elbow methods | RFM |



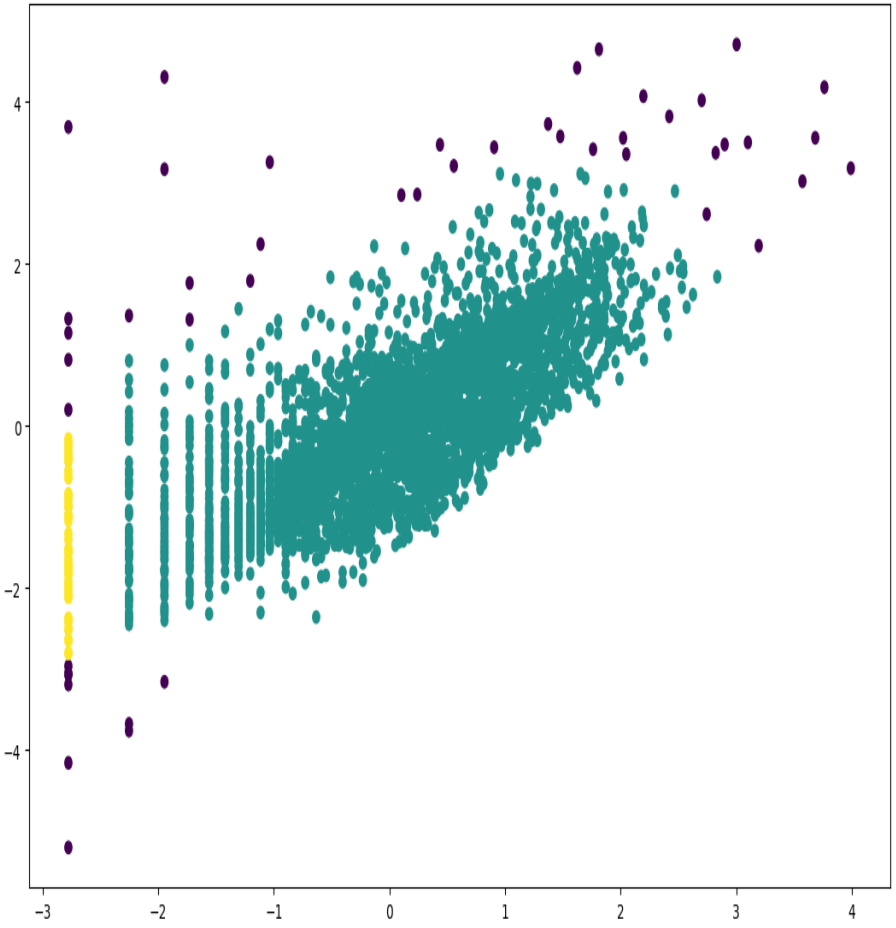


DBSCAN | RM | And DBSCAN | FM |

Dbscan with RM (2 clusters)

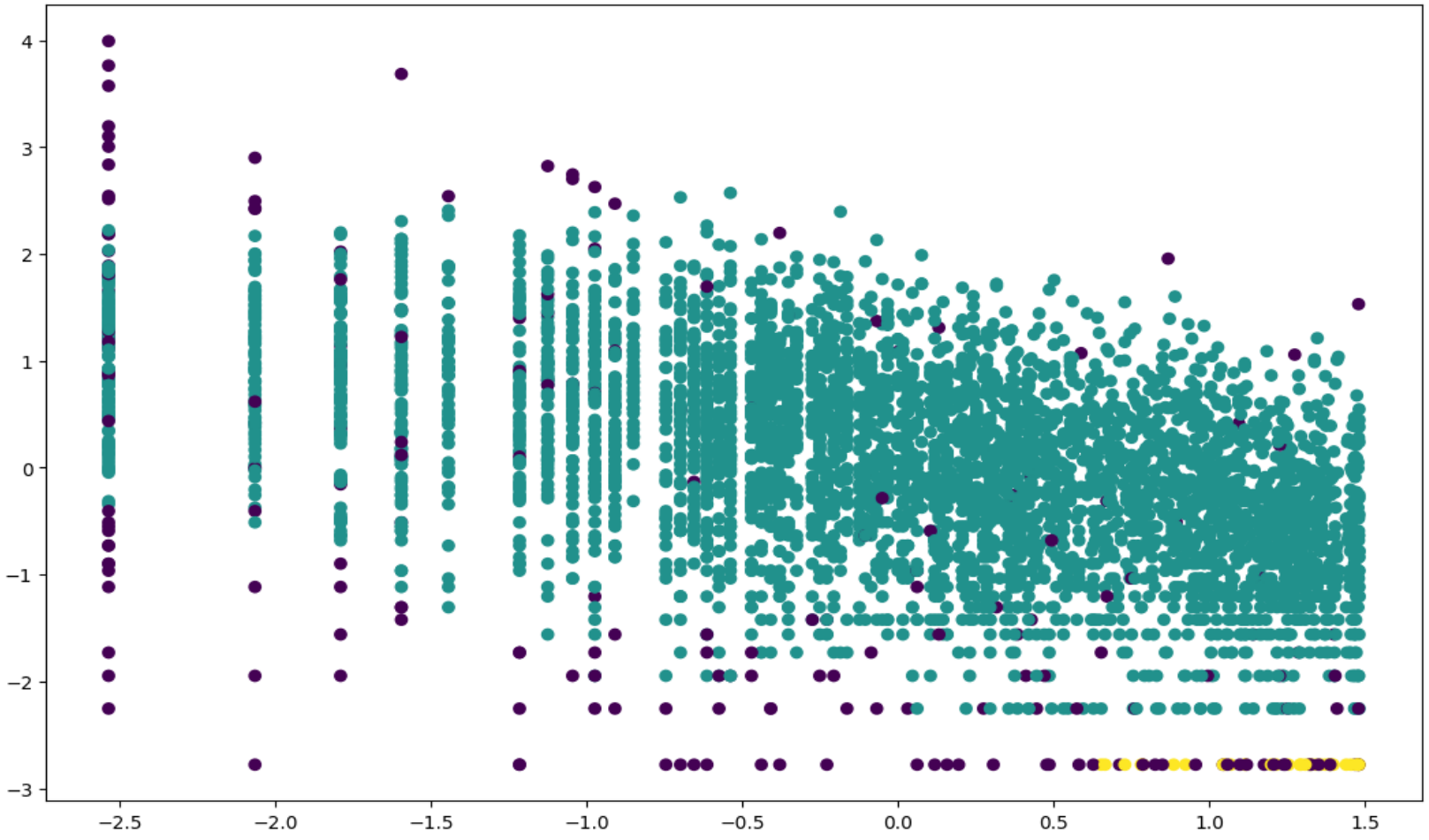


Dbscan with FM(3 Clusters)



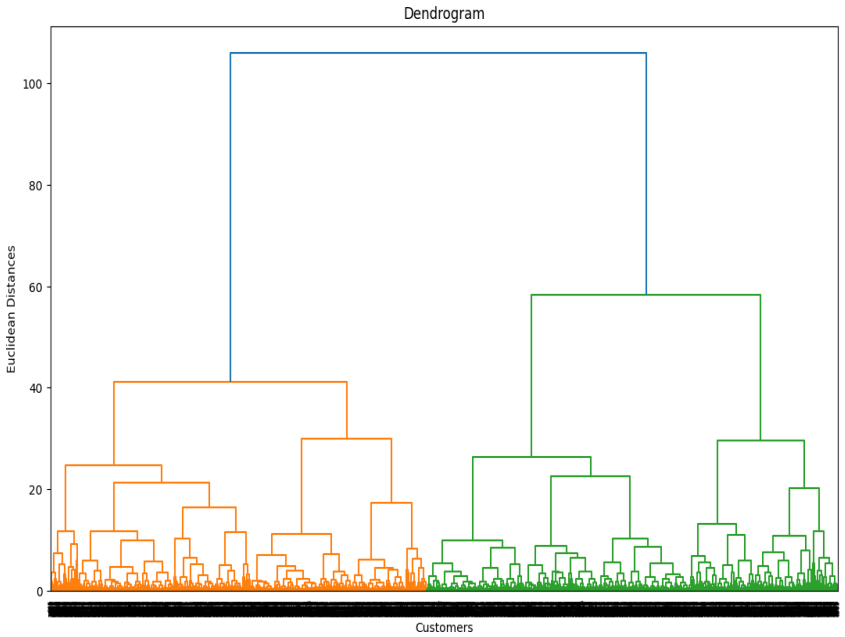
DBSCAN | RFM |

Dbscan with RFM(3 Clusters)

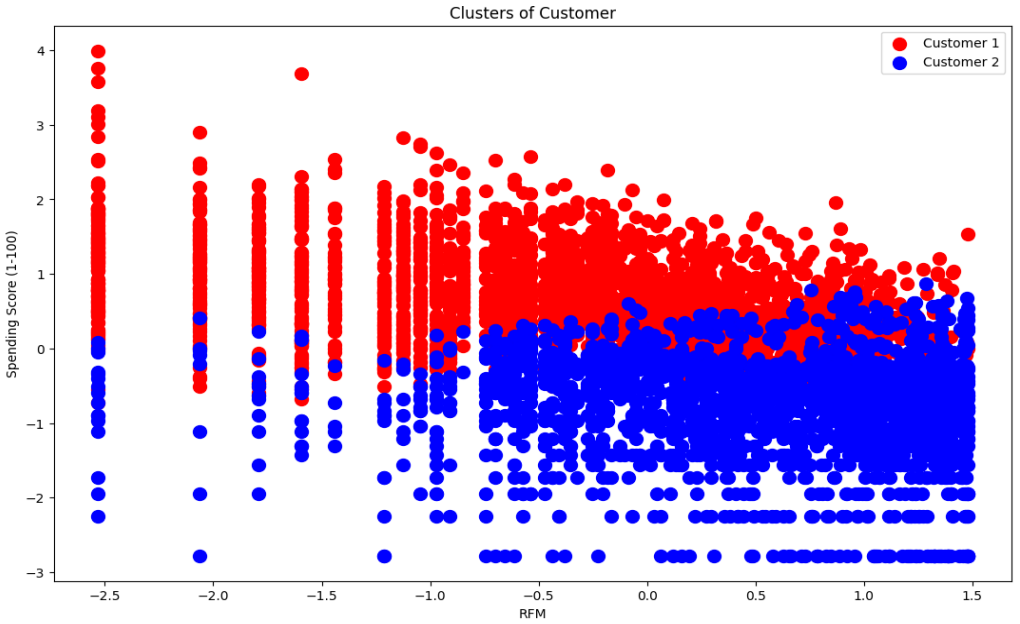


Hierarchical clustering | RFM |

Dendrogram



Cluster formation of RFM



FUTURE SCOPE

- The feature scope of this project includes several key customer behavior metrics that are critical for effective segmentation: Recency, Frequency, and Monetary value (RFM), as well as combinations of these metrics (RM, FM). Recency measures how recently a customer made a purchase, helping to identify active versus inactive customers. Frequency tracks how often a customer makes purchases, revealing customer loyalty and engagement levels. Monetary value evaluates how much a customer spends, distinguishing high-value customers from those with lower spending habits. By combining these features into different datasets, the project can explore various segmentation models and understand customer behavior from multiple perspectives. These features serve as the foundation for applying clustering techniques like K-Means, DBSCAN, and Hierarchical Clustering. Ultimately, the features allow businesses to categorize customers based on their purchasing patterns, enabling the creation of targeted marketing strategies, personalized promotions, and improved customer retention efforts.

CONCLUSION

- In conclusion, this customer segmentation project highlights the effectiveness of clustering techniques, including K-Means, DBSCAN, and Hierarchical Clustering, in identifying distinct customer segments based on different feature sets (RM, FM, and RFM). The K-Means algorithm consistently identifies two clusters as optimal across all datasets, indicating a clear division in customer behavior. While DBSCAN detects density-based clusters and identifies 2 clusters for RM and 3 for FM and RFM, it reveals more complex patterns in some datasets. Hierarchical Clustering also aligns with K-Means, identifying two clusters for the RFM dataset. The segmentation results suggest that two distinct customer groups are most apparent, allowing businesses to tailor strategies more effectively. These insights can help businesses design targeted marketing campaigns and personalized promotions, enhancing customer engagement and improving retention. Overall, the project emphasizes the value of clustering for understanding customer behavior and optimizing business strategies.

REFERENCES

- **J. Han, M. Kamber, and J. Pei**, *Data Mining: Concepts and Techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann, 2011.
- **P. Kotler and K. L. Keller**, *Marketing Management*, 15th ed. Pearson Education, 2016.
- **S. W. Wu, H. Y. Lin, and S. H. Wu**, "Applying RFM model and K-Means clustering for customer segmentation in e-commerce," in *Proc. IEEE Int. Conf. Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, Shenzhen, China, 2017, pp. 192-197. doi: 10.1109/CIVEMSA.2017.7995319.
- **A. K. Jain**, "Data clustering: 50 years beyond K-Means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651-666, 2010. doi: 10.1016/j.patrec.2009.09.011.
- **J. MacQueen**, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Stat. Probab.*, Berkeley, CA, USA, 1967, pp. 281–297.

REFERENCES

- **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)*, Portland, OR, USA, 1996, pp. 226-231.
- **S. Guha, R. Rastogi, and K. Shim**, "CURE: An efficient clustering algorithm for large databases," in *Proc. ACM SIGMOD Int. Conf. Management of Data*, 1998, pp. 73-84. doi: 10.1145/276304.276312.
- **G. Kaur and S. Kang**, "Customer segmentation using clustering and data mining techniques," in *Proc. 2016 2nd Int. Conf. Next Generation Computing Technologies (NGCT)*, Dehradun, India, 2016, pp. 398-402. doi: 10.1109/NGCT.2016.7877452.
- **C. Chen, M. Zhang, Y. Liu, and S. Ma**, "Understanding user intent in online shopping: A latent variable model," in *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 10, pp. 1936-1949, 2018. doi: 10.1109/TKDE.2018.2797932.


Thank You!