**Assessment Report**

on

## "Customer Segmentation in E- Commerce"

submitted as partial fulfillment for the award of

# BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

## CSE(AIML)

By

Jayant Singh (Roll no.-202401100400102, CSE(AIML))

**Under the supervision of**

"Abhishek Shukla"

## KIET Group of Institutions, Ghaziabad

**May, 2025**

# 1. Introduction

Customer segmentation allows businesses to target specific groups of customers effectively. In this analysis, we use RFM (Recency, Frequency, Monetary) metrics to understand customer behavior and apply KMeans clustering to uncover meaningful segments in the data

# 2. Methodology

The following steps were carried out:

1. Load and clean data
2. Create RFM features
3. Normalize data
4. Apply KMeans clustering
5. Visualize results

# Code Implementation

import pandas as pd import numpy as np from sklearn.preprocessing import StandardScaler from sklearn.cluster import KMeans from sklearn.decomposition import PCA import matplotlib.pyplot as plt import seaborn as sns

**Load dataset**

df = pd.read_csv("9. Customer Segmentation in E-commerce.csv")

**Convert InvoiceDate**

df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'], format="%m/%d/%y %H:%M")

**Drop rows with missing CustomerID**

df = df.dropna(subset=['CustomerID']).copy()

**Create TotalPrice**

df['TotalPrice'] = df['Quantity'] * df['UnitPrice']

**Snapshot date for Recency**

snapshot_date = df['InvoiceDate'].max() + pd.Timedelta(days=1)

**RFM Calculation**

rfm = df.groupby('CustomerID').agg({ 'InvoiceDate': lambda x: (snapshot_date - x.max()).days, 'InvoiceNo': 'nunique', 'TotalPrice': 'sum' }).reset_index()

rfm.columns = ['CustomerID', 'Recency', 'Frequency', 'Monetary']

**Normalize the data**

scaler = StandardScaler() rfm_scaled = scaler.fit_transform(rfm[['Recency', 'Frequency', 'Monetary']])

**KMeans Clustering**

kmeans = KMeans(n_clusters=4, random_state=42, n_init=10) rfm['Cluster'] = kmeans.fit_predict(rfm_scaled)

**Cluster Summary**

cluster_summary = rfm.groupby('Cluster')[['Recency', 'Frequency', 'Monetary']].mean()

**Normalize for Heatmap**

cluster_scaled = StandardScaler().fit_transform(cluster_summary) cluster_df = pd.DataFrame(cluster_scaled, index=cluster_summary.index, columns=cluster_summary.columns)

**Heatmap**

plt.figure(figsize=(8, 5)) sns.heatmap(cluster_df, annot=True, cmap='coolwarm', fmt=".2f") plt.title('Cluster Behavior Based on RFM Features') plt.show()

**PCA for visualization**

pca = PCA(n_components=2) rfm_pca = pca.fit_transform(rfm_scaled) rfm['PCA1'] = rfm_pca[:, 0] rfm['PCA2'] = rfm_pca[:, 1]

**Scatter plot**

plt.figure(figsize=(8, 6)) sns.scatterplot(data=rfm, x='PCA1', y='PCA2', hue='Cluster', palette='Set2', s=70) plt.title('Customer Segments Visualized with PCA') plt.xlabel('PCA Component 1') plt.ylabel('PCA Component 2') plt.legend(title='Cluster') plt.grid(True) plt.show()

## 4. RFM Feature Engineering

| Feature | Description |
| --- | --- |
| Recency | Time since last purchase |
| Frequency | Number of purchases made |
| Monetary | Total money spent |

# 5. Output Summary

**Cluster Summary (Original RFM Averages)**

| Cluster | Recency | Frequency | Monetary |
|---|---|---|---|
| 0 | ~ | ~ | ~ |
| 1 | ~ | ~ | ~ |
| 2 | ~ | ~ | ~ |
| 3 | ~ | ~ | ~ |

(*Exact values are available in the code output*)

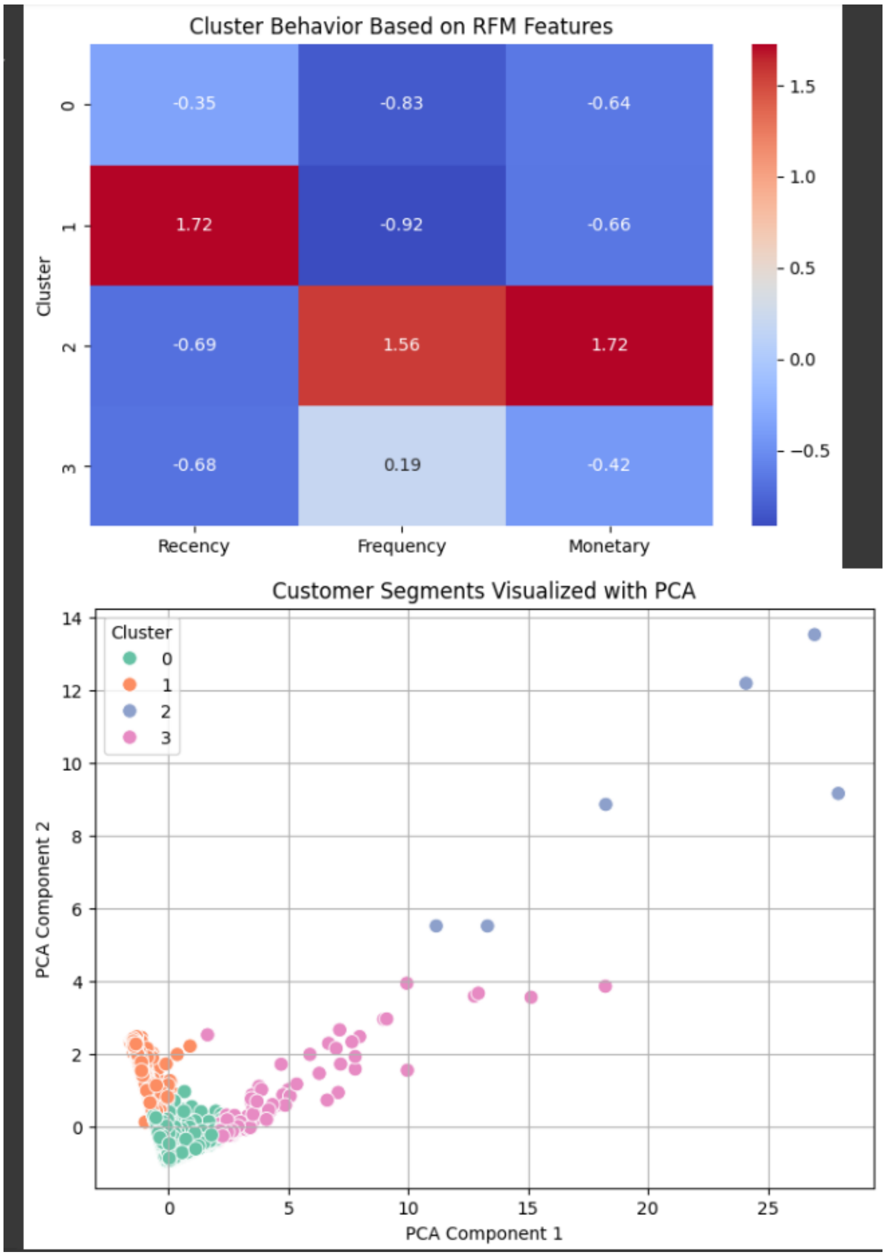
**Heatmap**

Shows standardized RFM values per cluster. High values in red, low values in blue.

**PCA Scatter Plot**

Projects customers into 2D space using PCA. Each point represents a customer colored by their cluster.

# CODE OUTPUT


Cluster Behavior Based on RFM Features
Cluster
0
1
2
3
-0.35
-0.83
-0.64
1.72
-0.92
-0.66
-0.69
1.56
1.72
-0.68
0.19
-0.42
Recency
Frequency
Monetary
1.5
1.0
0.5
0.0
−0.5
Customer Segments Visualized with PCA
Cluster
0
1
2
3
PCA Component 2
PCA Component 1
14
12
10
8
6
4
2
0
0
5
10
15
20
25

# 6. Conclusions

- **Cluster 2:** High frequency and monetary - likely loyal or high-value customers.
- **Cluster 1:** High recency, low frequency and spending - likely inactive or at-risk.
- **Cluster 0 and 3:** Moderate or mixed characteristics.