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

Name : Srish Jana

Roll Number : 202401100400189,

Section: C

Under the supervision of

“ABHISHEK SHUKLA”

KIET Group of Institutions, Ghaziabad

Affiliated to Dr. A.P.J. Abdul Kalam Technical University,Lucknow(Formerly UPTU)

May, 2025

Introduction

In today's highly competitive e-commerce landscape, understanding customer behaviour is crucial for businesses to enhance customer satisfaction, improve retention, and drive sales. One effective approach to gaining such insights is **customer segmentation** — the practice of dividing a customer base into distinct groups based on shared characteristics or behaviours.

This project aims to implement a customer segmentation strategy using the **RFM (Recency, Frequency, Monetary) model** combined with **K-Means clustering**. RFM analysis is a proven method in marketing analytics that evaluates customers based on:

- **Recency**: How recently a customer made a purchase.
- **Frequency**: How often they make purchases.
- **Monetary Value**: How much money they spend.

By analyzing transactional data from an e-commerce platform, we group customers into clusters that represent different levels of engagement and value to the business. These insights help organizations design personalized marketing strategies such as loyalty rewards, re-engagement campaigns, and cross-selling initiatives. Ultimately, customer segmentation enables smarter decision-making and more efficient allocation of marketing resources.

Objective

The primary objective of this project is to perform customer segmentation using RFM (Recency, Frequency, Monetary) analysis and K-Means clustering. This allows businesses to understand customer behavior, target marketing campaigns more effectively, and increase overall customer satisfaction and retention.

---

1. Data Preprocessing

To ensure clean and meaningful data, the following preprocessing steps were performed:

- **Missing Customer IDs Removed**: Transactions with missing Customer ID were discarded, as they could not be attributed to any individual customer.
- **Returns Filtered**: Transactions associated with product returns were removed. These were identified using Invoice No. values starting with the letter 'C'.
- **Invoice Date Conversion**: The Invoice Date column was converted to datetime format for accurate recency calculation.
- **Total Price Computed**: A new column Total Price was created by multiplying Quantity with Unit Price, representing the revenue per transaction.

---

2. RFM Feature Engineering

Customers were segmented using RFM metrics calculated as follows:

- **Recency (R)**: Number of days since the customer's last purchase (calculated as the difference between a snapshot date and the most recent purchase date).
- **Frequency (F)**: Number of unique purchase transactions per customer (based on distinct InvoiceNo entries).
- **Monetary Value (M)**: Total amount spent by the customer (sum of TotalPrice).

Customers with non-positive monetary values were excluded from further analysis.

---

3. Data Transformation

To handle scale differences between RFM variables and to normalize data:

- **Log Transformation**: Applied to the RFM values to reduce skewness.
- **Standardization**: StandardScaler was used to bring all RFM values onto a comparable scale (mean = 0, standard deviation = 1).

---

4. Clustering with K-Means

- The **Elbow Method** was used to determine the optimal number of clusters. Based on the plot of Within-Cluster Sum of Squares (SSE), **4 clusters** were selected for segmentation.
- K-Means clustering algorithm was applied to the scaled RFM values to assign customers to one of the four distinct segments.

---

5. Cluster Profiling

Each cluster was profiled based on the average RFM scores and customer count:

- **Cluster 0**: High-value frequent buyers (Low Recency, High Frequency, High Monetary)
- **Cluster 1**: At-risk customers (High Recency, Low Frequency, Low Monetary)
- **Cluster 2**: Moderate-value seasonal shoppers
- **Cluster 3**: One-time or low-value customers

These profiles were further visualized using:

- **Scatter Plots**: Showed relationships like Recency vs Monetary, aiding interpretation of customer value.
- **Box Plots**: Illustrated the distribution of RFM scores across clusters.

---

6. Actionable Insights

Based on the clustering and RFM profiles, the following strategic insights are proposed:

- **Reward Loyalty**: Customers in high-value clusters can be offered loyalty benefits or exclusive discounts.
- **Re-engage At-Risk Customers**: Personalized emails or limited-time offers can help bring back inactive or low-engagement users.
- **Upsell to Seasonal Shoppers**: Target seasonal buyers during relevant periods with timely campaigns.
- **Cross-Selling Opportunities**: Analyze popular product pairings within frequent buyers' purchases to recommend bundles.

Conclusion

The implementation of RFM-based customer segmentation with K-Means clustering offers deep insights into customer behaviour, enabling better marketing decisions and improved customer experience. This analytical framework can be extended with additional demographic or behavioural features for even more refined targeting.

CODE :

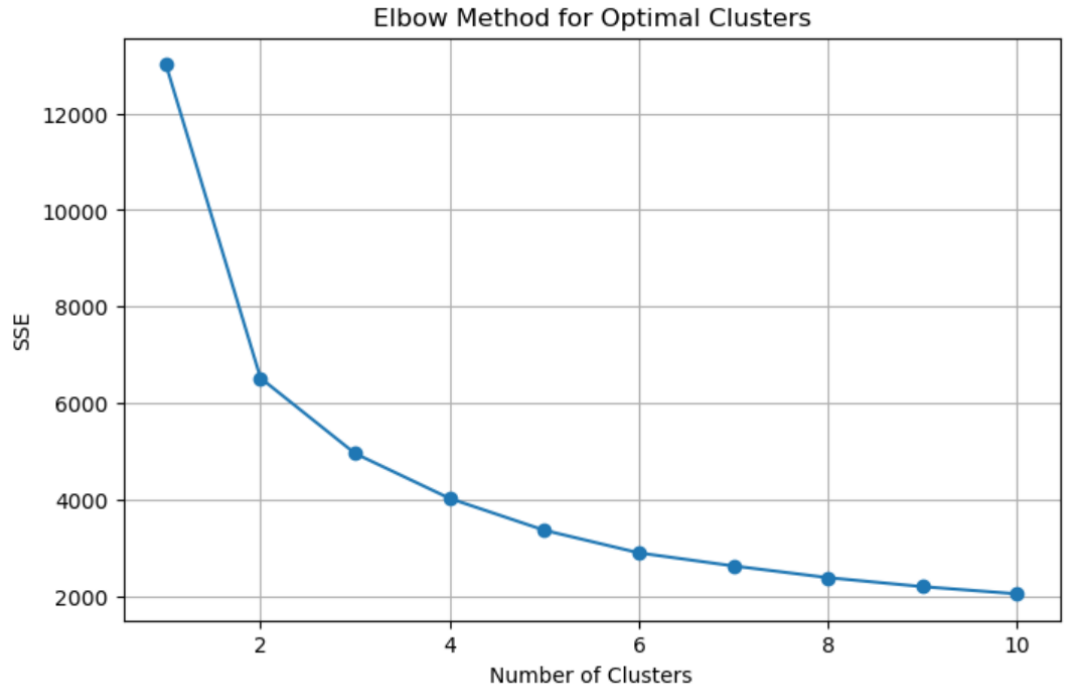
```
import pandas as pd

import numpy as np

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt

import seaborn as sns

# ---------------------------------

# 1. Load & Preprocess the Dataset

# ---------------------------------

df = pd.read_csv("C:/Users/srish/Downloads/9. Customer Segmentation in E-commerce.csv")

# Drop rows with missing CustomerID

df = df[df["CustomerID"].notna()]

# Remove returns (InvoiceNo starting with 'C')

df = df[~df["InvoiceNo"].astype(str).str.startswith('C')]

# Convert InvoiceDate to datetime

df["InvoiceDate"] = pd.to_datetime(df["InvoiceDate"], dayfirst=True,errors = "coerce")

# Add TotalPrice column

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

# ---------------------------------

# 2. RFM Feature Calculation

# ---------------------------------

snapshot_date = df["InvoiceDate"].max() + pd.DateOffset(days=1)

rfm = df.groupby("CustomerID").agg({

    "InvoiceDate": lambda x: (snapshot_date - x.max()).days,  # Recency

    "InvoiceNo": "nunique",  # Frequency

    "TotalPrice": "sum"  # Monetary

}).reset_index()
```

```
rfm.columns = ["CustomerID", "Recency", "Frequency", "Monetary"]

# Remove customers with zero or negative Monetary value

rfm = rfm[rfm["Monetary"] > 0]

# ---------------------------------

# 3. Data Transformation (Log + Scaling)

# ---------------------------------

rfm_log = rfm.copy()

rfm_log[["Recency", "Frequency", "Monetary"]] = np.log1p(rfm_log[["Recency", "Frequency", "Monetary"]])

scaler = StandardScaler()

rfm_scaled = scaler.fit_transform(rfm_log[["Recency", "Frequency", "Monetary"]])

# ---------------------------------

# 4. Elbow Method to Find Optimal K

# ---------------------------------

sse = []

for k in range(1, 11):

    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)

    kmeans.fit(rfm_scaled)

    sse.append(kmeans.inertia_)

plt.figure(figsize=(8, 5))

plt.plot(range(1, 11), sse, marker="o")

plt.title("Elbow Method for Optimal Clusters")

plt.xlabel("Number of Clusters")

plt.ylabel("SSE")

plt.grid(True)

plt.show()

# ---------------------------------

# 5. K-Means Clustering (k = 4)

# ---------------------------------

kmeans = KMeans(n_clusters=4, random_state=42, n_init=10)
```

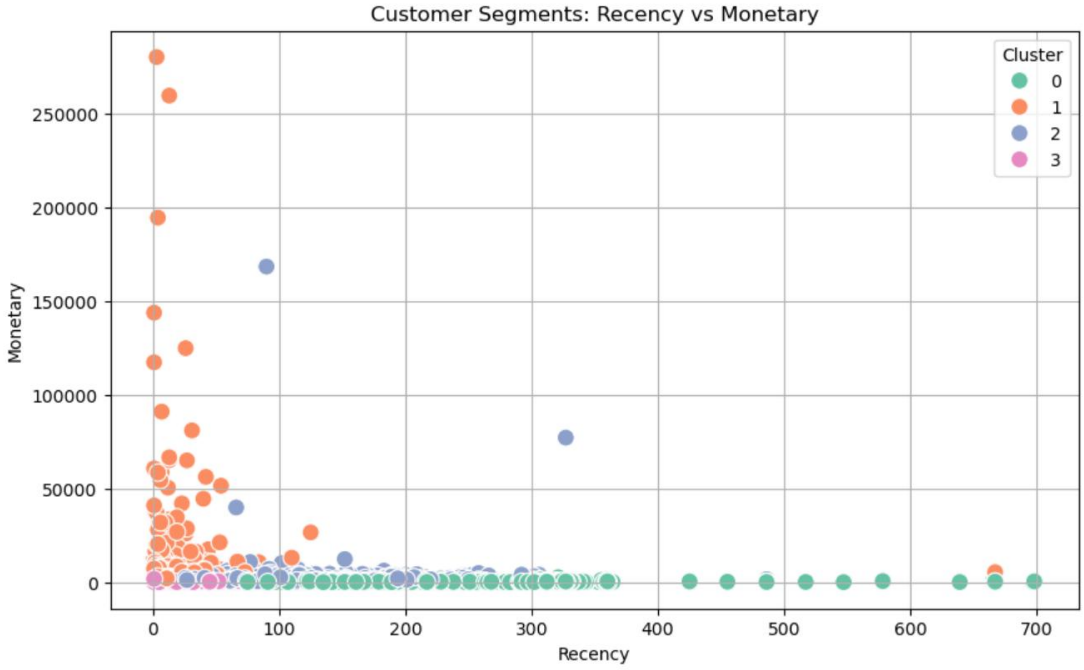
```
rfm["Cluster"] = kmeans.fit_predict(rfm_scaled)

# ---------------------------------

# 6. Cluster Profiling

# ---------------------------------

cluster_summary = rfm.groupby("Cluster").agg({

   "Recency": "mean",

   "Frequency": "mean",

   "Monetary": "mean",

   "CustomerID": "count"

}).rename(columns={"CustomerID": "Count"})

print("\nCluster Summary:")

print(cluster_summary)

# ---------------------------------

# 7. Cluster Visualizations

# ---------------------------------

# Recency vs Monetary Scatter Plot

plt.figure(figsize=(12, 8))

sns.scatterplot(data=rfm, x="Recency", y="Monetary", hue="Cluster", palette="Set2", s=100)

plt.title("Customer Segments: Recency vs Monetary")

plt.grid(True)

plt.show()

# Box Plots for Each RFM Dimension by Cluster

plt.figure(figsize=(12, 8))

sns.boxplot(data=rfm, x="Cluster", y="Recency", hue="Cluster", palette="Set3", legend=False)

plt.title("Recency Distribution by Cluster")

plt.grid(True)

plt.show()


plt.figure(figsize=(12, 8))
```

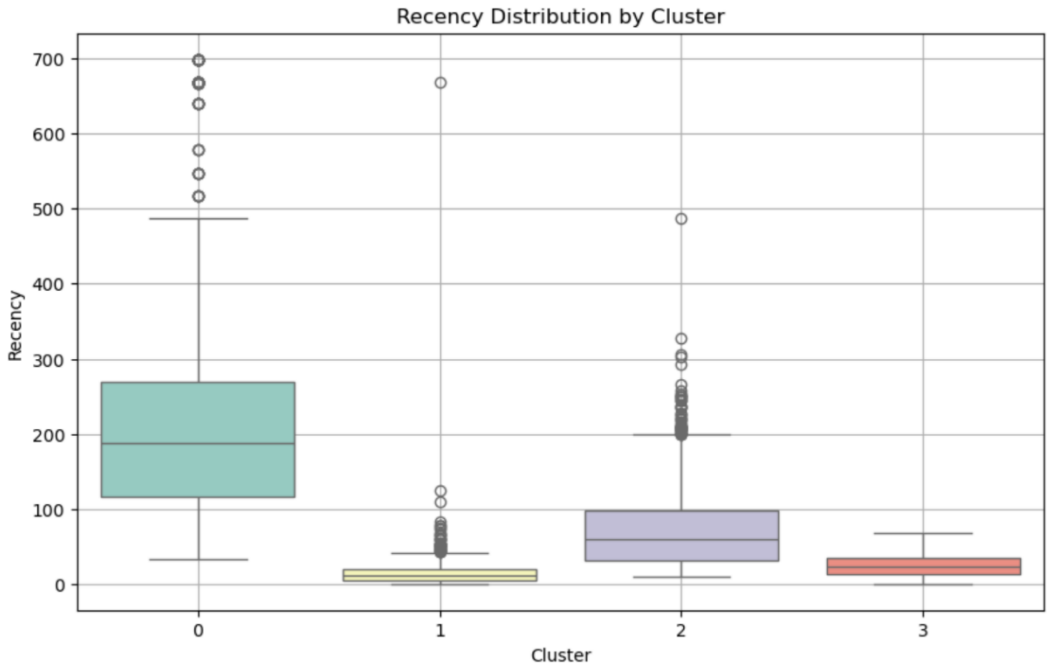
```
sns.boxplot(data=rfm, x="Cluster", y="Frequency", hue="Cluster", palette="Set3", legend=False)
plt.title("Frequency Distribution by Cluster")
plt.grid(True)
plt.show()
plt.figure(figsize=(12, 8))
sns.boxplot(data=rfm, x="Cluster", y="Monetary", hue="Cluster", palette="Set3", legend=False)
plt.title("Monetary Distribution by Cluster")
plt.grid(True)
plt.show()
```

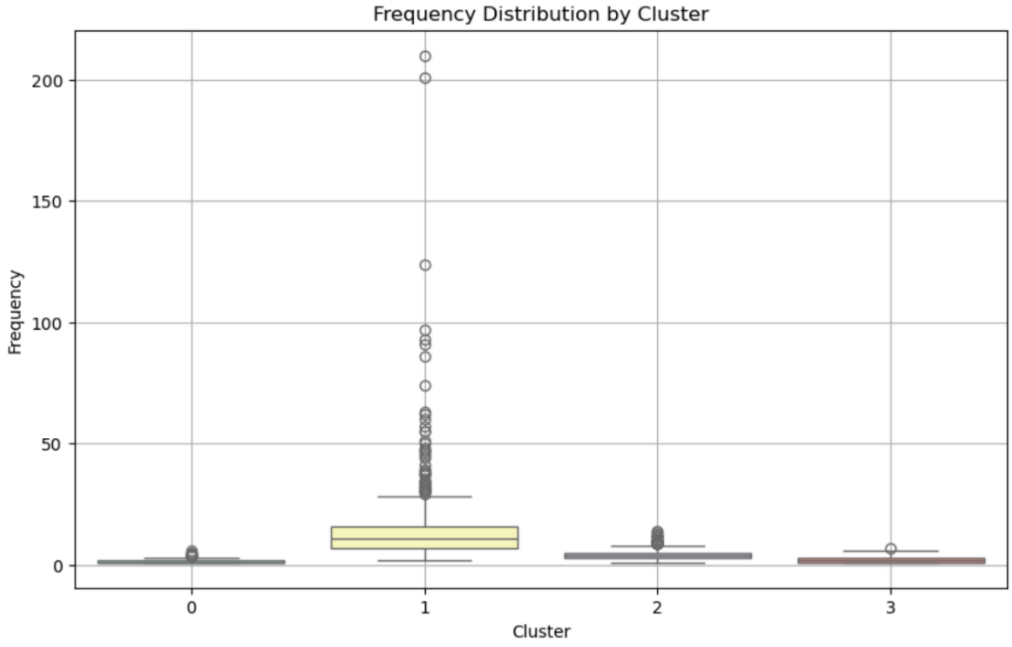
OUTPUTs

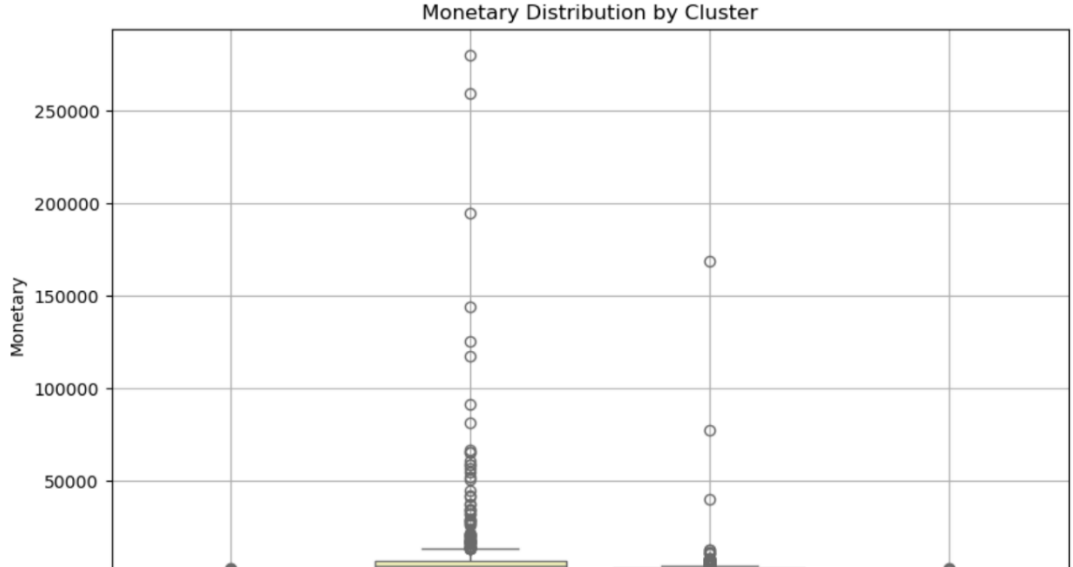



```
Cluster Summary:
             Recency  Frequency     Monetary  Count
Cluster
0         207.729763   1.365186   360.851528   1643
1          17.018576  14.447368  8445.865402    646
2          74.681197   4.388889  2041.588883   1170
3          25.156997   2.075085   539.068032    879
```




Recency Distribution by Cluster
Recency
Cluster
0
1
2
3
0
100
200
300
400
500
600
700


Frequency Distribution by Cluster
Frequency
Cluster
0
1
2
3
0
50
100
150
200


Monetary Distribution by Cluster
Monetary
50000
100000
150000
200000
250000