KIET Group of Institutions, Ghaziabad

May, 2025

BY SNEHA YADAV(202401100300249)

Under the of Supervision of

"MR.ABHISHEK SHUKLA"

Introduction

🛍️ Customer Segmentation in E-Commerce using RFM Analysis and K-Means Clustering

🔍 Introduction

In the highly competitive world of e-commerce, understanding customer behavior is key to driving growth and improving customer satisfaction. **Customer segmentation** is a powerful technique that helps businesses group customers based on common characteristics and

behaviors. This allows companies to tailor marketing strategies, improve targeting, and increase customer retention.

This project uses **RFM (Recency, Frequency, Monetary) analysis** combined with **K-Means clustering** to segment customers of an online retail store.

🧠 What is RFM Analysis?

- **Recency**: How recently a customer made a purchase (the fewer days, the better).
- **Frequency**: How often a customer makes a purchase (more frequent = more loyal).
- **Monetary**: How much money a customer spends (higher spenders may be more valuable).

By calculating these metrics for each customer, we create a profile that reflects their shopping behavior.

⚙️ What This Code Does

1. **Loads and cleans** the e-commerce transactional data.
2. Calculates **RFM metrics** for each customer.
3. **Normalizes** the RFM values to prepare for clustering.
4. Uses the **Elbow Method** to determine the optimal number of customer segments.
5. Applies **K-Means Clustering** to group customers into distinct clusters.

6. **Visualizes** the results and provides a summary of each cluster.

Methodology

The methodology for customer segmentation in this project is based on **RFM analysis** and **K-Means clustering**, structured into the following steps:

---

1. Data Collection and Loading

- The dataset used consists of transactional data from an e-commerce platform.
- It includes customer purchase history, invoice details, purchase quantities, and prices.
- The dataset is loaded into a Pandas DataFrame for processing.

---

2. Data Preprocessing

- **Missing values** in the CustomerID column are removed, as these entries are essential for customer-level analysis.
- The InvoiceDate column is converted to datetime format for calculating recency.
- A new feature TotalPrice is created by multiplying Quantity and UnitPrice to compute the revenue per transaction.

---

3. RFM Feature Engineering

- The **Recency** value is calculated as the number of days between the customer's most recent purchase and the latest date in the dataset.
- **Frequency** is measured as the number of unique invoices per customer, representing how often they purchase.
- **Monetary** value is the total amount spent by each customer.

---

4. Data Normalization

- Since RFM values vary in scale, the features are normalized using StandardScaler from Scikit-learn.
- This ensures all features contribute equally to the clustering algorithm.

---

5. Optimal Cluster Selection (Elbow Method)

- The **Elbow Method** is used to identify the optimal number of clusters (K) by plotting Within-Cluster Sum of Squares (WCSS) against a range of cluster numbers.
- The "elbow point" in the plot suggests a balance between compactness and number of clusters.

---

6. Customer Segmentation (K-Means Clustering)

- The K-Means clustering algorithm is applied to the scaled RFM data using the chosen value of K.
- Each customer is assigned to a cluster based on similarities in their RFM profile.

---

7. Cluster Profiling and Visualization

- A summary table is created to show the average **Recency**, **Frequency**, **Monetary**, and **number of customers** in each cluster.
- A scatter plot is used to visualize how clusters differ in terms of recency and monetary values.

Code

```
 # STEP 1: Upload the file manually

from google.colab import files

uploaded = files.upload()


# STEP 2: Import libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt
```

```
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

# STEP 3: Load the dataset (change the filename if needed)
df = pd.read_csv("9. Customer Segmentation in E-commerce.csv", encoding='ISO-8859-1')

# STEP 4: Clean and prepare the data
df.dropna(subset=['CustomerID'], inplace=True)
df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'])
df['TotalPrice'] = df['Quantity'] * df['UnitPrice']

# STEP 5: RFM calculation
latest_date = df['InvoiceDate'].max()
rfm = df.groupby('CustomerID').agg({
    'InvoiceDate': lambda x: (latest_date - x.max()).days,
    'InvoiceNo': 'nunique',
    'TotalPrice': 'sum'
}).reset_index()
rfm.columns = ['CustomerID', 'Recency', 'Frequency', 'Monetary']
rfm = rfm[rfm['Monetary'] > 0]

# STEP 6: Normalize RFM data
scaler = StandardScaler()
```

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


# STEP 7: Elbow method to find optimal clusters
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, random_state=42, n_init=10)
    kmeans.fit(rfm_scaled)
    wcss.append(kmeans.inertia_)


# Plot the elbow curve
plt.figure(figsize=(8, 4))
plt.plot(range(1, 11), wcss, marker='o')
plt.title("Elbow Method for Optimal K")
plt.xlabel("Number of Clusters")
plt.ylabel("WCSS")
plt.grid(True)
plt.tight_layout()
plt.show()


# STEP 8: Apply KMeans with chosen number of clusters (e.g., 4)
kmeans = KMeans(n_clusters=4, random_state=42, n_init=10)
rfm['Cluster'] = kmeans.fit_predict(rfm_scaled)


# STEP 9: View cluster summary
```

```
summary = rfm.groupby('Cluster').agg({
    'Recency': 'mean',
    'Frequency': 'mean',
    'Monetary': 'mean',
    'CustomerID': 'count'
}).rename(columns={'CustomerID': 'Count'}).reset_index()
print(summary)

# STEP 10: Visualize clusters
sns.scatterplot(data=rfm, x='Recency', y='Monetary', hue='Cluster', palette='Set2')
plt.title("Customer Segmentation Based on RFM")
plt.show()
```

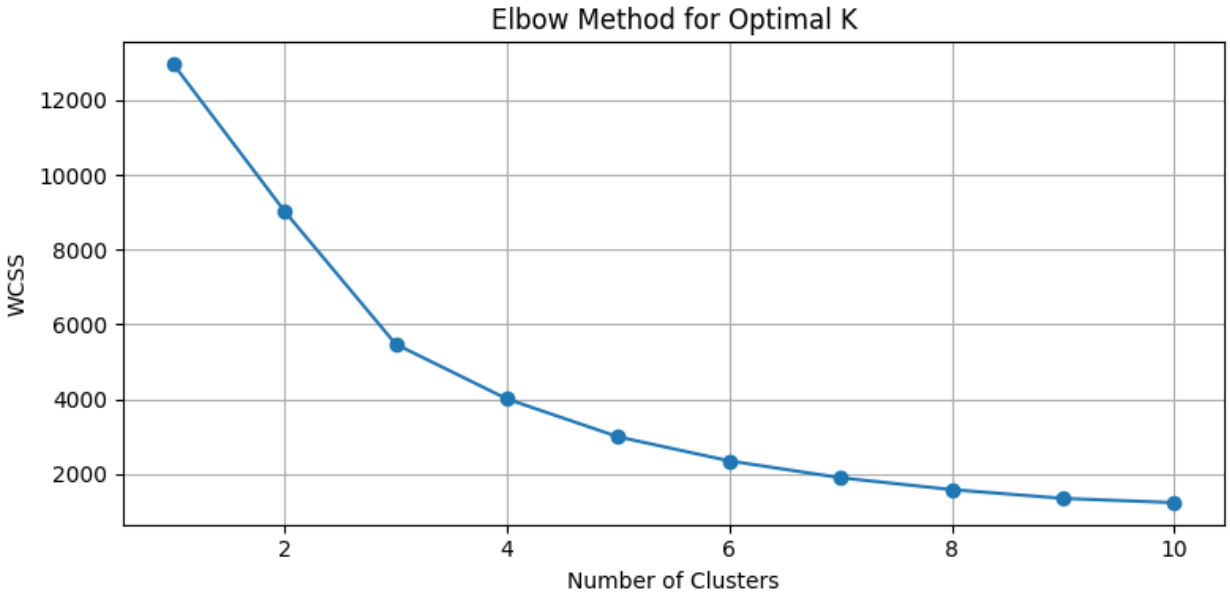
Output/Result

□ 9. Customer Segmentation in E-commerce.csv(text/csv) - 44496850 bytes, last modified: 4/18/2025 - 100% done

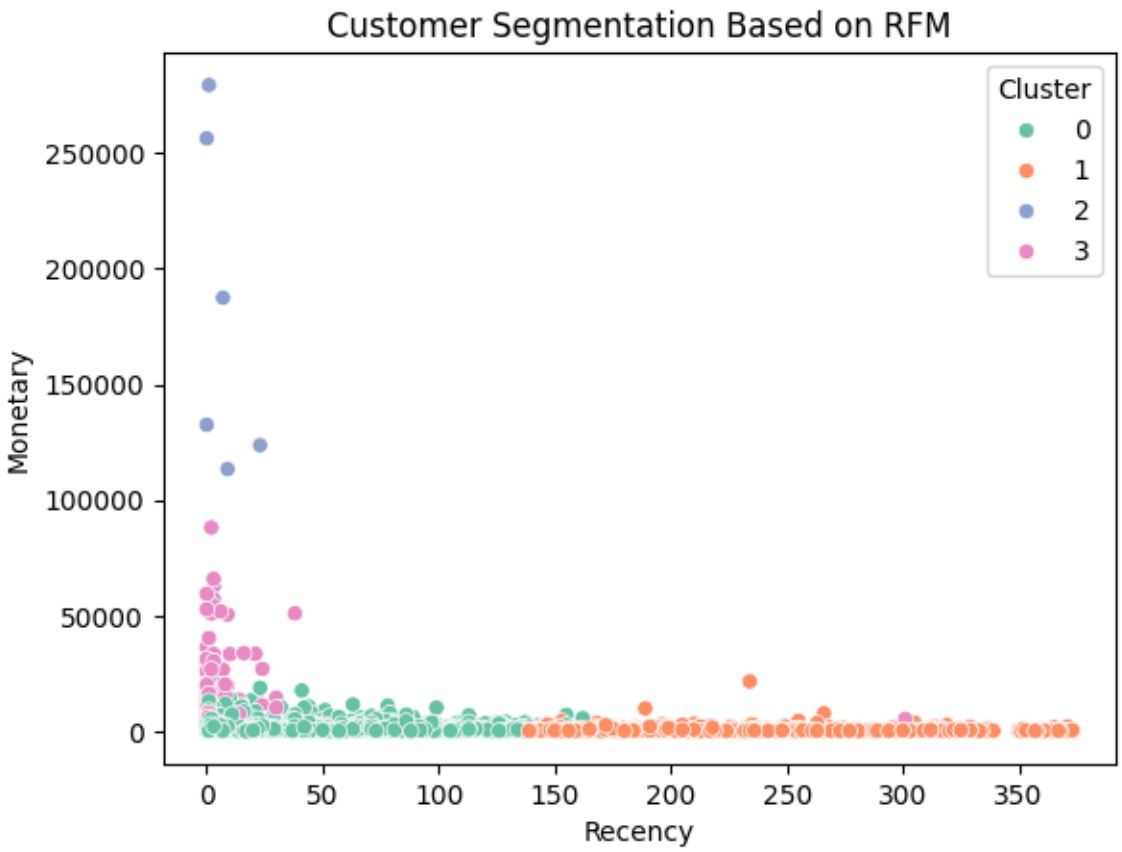
Saving 9. Customer Segmentation in E-commerce.csv to 9. Customer Segmentation in E-commerce.csv

<ipython-input-4-89b941be2f6a>:18: UserWarning: Could not infer format, so each element will be parsed individually, falling back to `dateutil`. To ensure parsing is consistent and as-expected, please specify a format.

df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'])



	Cluster	Recency	Frequency	Monetary	Count
0	0	40.137405	4.818702	1487.377625	3144
1	1	243.889831	1.838041	485.190255	1062
2	2	6.666667	89.000000	182181.981667	6
3	3	8.181818	40.672727	18441.961455	110


Customer Segmentation Based on RFM
Cluster
0
1
2
3
Monetary
250000
200000
150000
100000
50000
0
0
50
100
150
200
250
300
350
Recency