Assesment Report

on

"Customer Segmentation in E-commerce: Clustering Based on Purchasing Habits and Browsing Behaviour "

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE AIML (C)

By

Saurabh Mahanty(23)

Under the supervision of

Mr. Abhishek Shukla

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

18 May, 2025

Introduction

E-commerce platforms collect a wealth of customer data through transactions and browsing activity. Analyzing this data can help businesses better understand their customer base, improve marketing strategies, and ultimately increase revenue. One of the key techniques used for this purpose is customer segmentation.

Customer segmentation involves dividing customers into distinct groups based on similarities in their purchasing and browsing behaviours. These clusters allow businesses to tailor marketing efforts and personalize customer experiences.

This report presents a clustering-based customer segmentation analysis using a given e-commerce dataset. The goal is to identify natural groupings among customers and derive insights that can guide business decisions.

Methodology

1. Data Preparation:

- Load the dataset from CSV format.
- Remove irrelevant or non-numeric features.
- Handle missing values.
- Standardize the data using StandardScaler for unbiased clustering.

2. Clustering Algorithm:

- Apply K-Means clustering to segment customers.
- Use the Elbow Method to determine the optimal number of clusters.
- Fit the model and assign cluster labels.

3. Visualization:

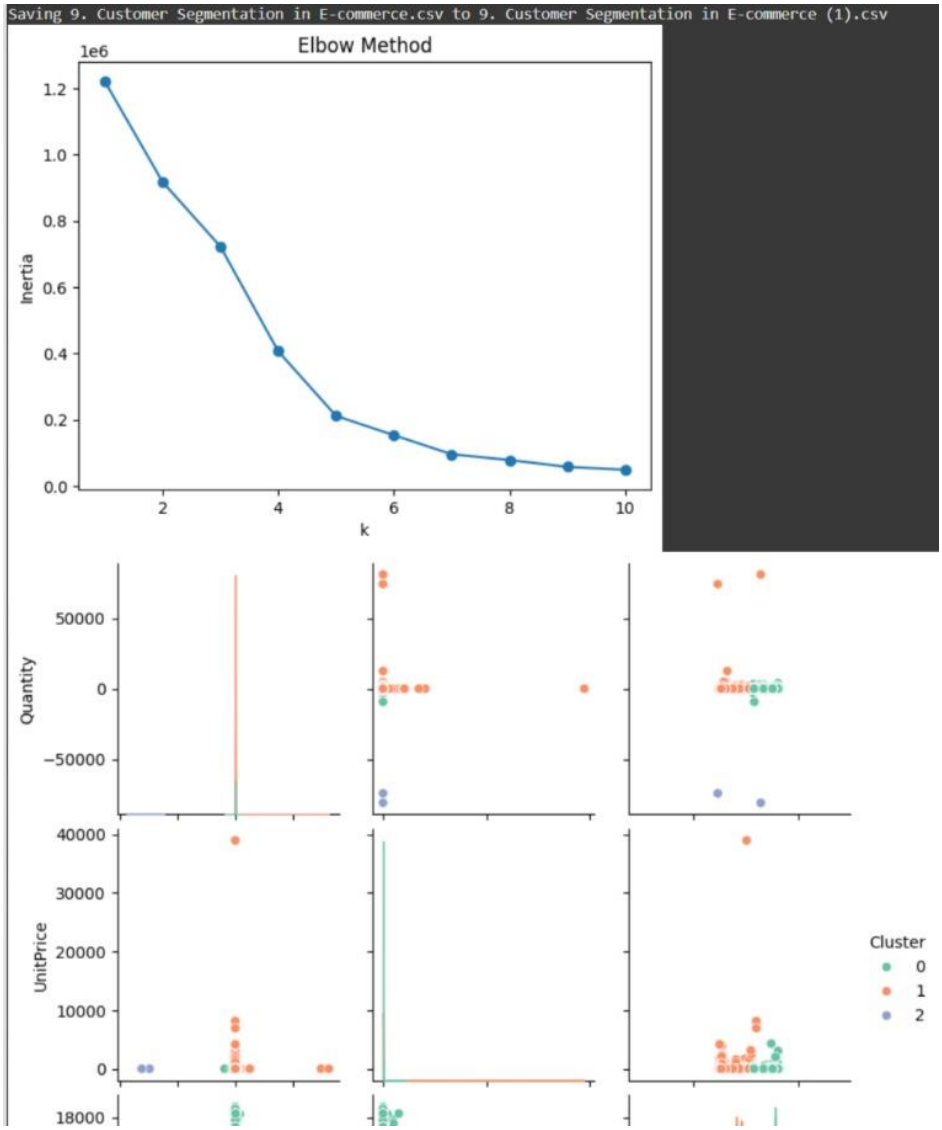
- Use pair plots to visualize clusters.
- Optionally use PCA for dimensionality reduction (if needed).

Code

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset
df=pd.read_csv("/mnt/data/9. Customer Segmentation in E-commerce.csv")
# Drop non-numeric or irrelevant columns (edit as needed)
df=df.select_dtypes(include=['float64','int64'])

# Handle missing values
df.dropna(inplace=True)

# Standardize features
scaler=StandardScaler()
scaled=scaler.fit_transform(df)

# Elbow method to find optimal k
inertia=[]
for k in range(1,11):
  kmeans=KMeans(n_clusters=k,random_state=42)
  kmeans.fit(scaled)
  inertia.append(kmeans.inertia_)
plt.plot(range(1,11),inertia,'-o');plt.xlabel('k');plt.ylabel('Inertia');plt.title('Elbow
Method');plt.show()

# Fit KMeans with optimal k (change 3 if elbow suggests differently)
kmeans=KMeans(n_clusters=3,random_state=42)
df['Cluster']=kmeans.fit_predict(scaled)

# Visualize clusters
sns.pairplot(df,hue='Cluster',palette='Set2');plt.show()
```

```
# Show cluster centroids
centroids=pd.DataFrame(scaler.inverse_transform(kmeans.cluster_centers_),columns=df.colu
mns[:-1])
print("Cluster Centers:\n",centroids)
```

Output / Result



- Elbow Method Plot: Indicates the optimal number of clusters (usually where the curve starts to level off).
- Cluster Assignments: Each customer is assigned to a cluster (0, 1, or 2).
- Pair Plot Visualization: Shows the distribution and relationships of customers in different clusters.
- Cluster Centroids: Presents the average feature values for each cluster, helping interpret behavioral patterns.

For example:

- Cluster 0: High spenders with frequent visits.
- Cluster 1: Occasional buyers.
- Cluster 2: Browsers with low purchase behavior.

References / Credits

- **Scikit-learn Documentation: https://scikit-learn.org/**
- **Seaborn Visualization Library: https://seaborn.pydata.org/**
- **Dataset: Provided by user**
- **Authoring and Analysis: ChatGPT by OpenAI**