KIET GROUP OF INSTITUTIONS
Connecting Life with Learning
www.kiet.edu
Delhi-NCR, Ghaziabad

“Customer Segmentation in E-commerce: Identify customer clusters based on purchasing habits and browsing behavior ”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

In

Name of discipline : CSE-AI

By

Tanishk Gupta (202401100300259,CSEAI-D)

Under the supervision of

“MR.ABHISHEK SHUKLA”

KIET Group of Institutions, Ghaziabad

May, 2025

Introduction

Customer segmentation is a vital technique in e-commerce used to classify users into distinct groups based on their purchasing patterns and browsing behavior. By doing so, businesses can target each group with more personalized marketing strategies, thus increasing customer satisfaction and boosting revenue. In this project, we implement unsupervised machine learning using K-Means clustering to identify meaningful customer segments based on behavior data.

Methodology

1. Data Collection: The dataset is uploaded manually through Google Colab and read using `pandas`.
2. Data Preprocessing:
    - All missing values are removed to ensure clean data.
    - Only numeric columns are selected for clustering analysis.
3. Feature Scaling:
    - StandardScaler is used to normalize the numerical features so that each feature contributes equally to clustering.
4. Finding Optimal Clusters:
    - The Elbow Method is used by plotting Within-Cluster Sum of Squares (WCSS) to identify the best value of k.
    - From the graph, 4 clusters were chosen as optimal.
5. K-Means Clustering:
    - KMeans is applied with `n_clusters=4`.

   - A new column `Cluster` is added to the original DataFrame.

6. Dimensionality Reduction for Visualization:
   - PCA (Principal Component Analysis) is applied to reduce the dataset to 2D.
   - Data is visualized in a scatter plot where each cluster is color-coded.

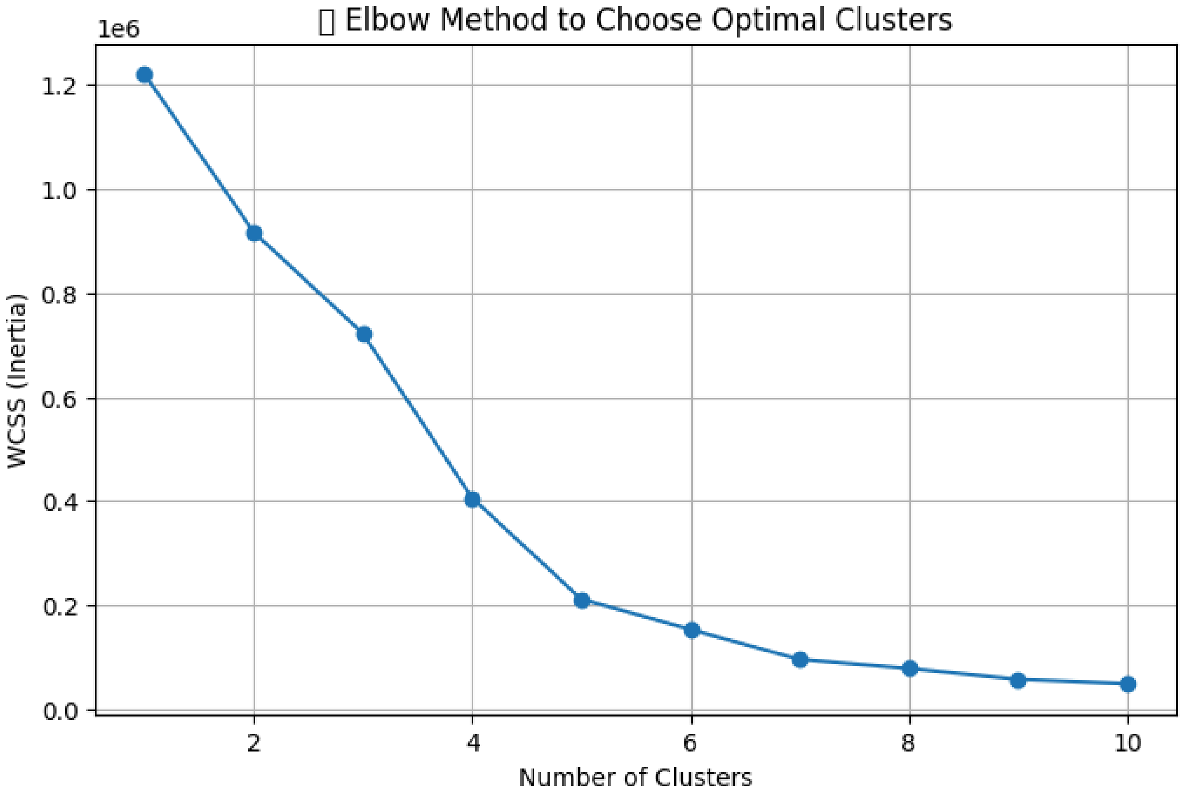
Code

```
# 📦 Required Libraries

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.preprocessing import StandardScaler

from sklearn.cluster import KMeans

from sklearn.decomposition import PCA

from google.colab import files

import io



# 📁 Step 1: Upload CSV File

uploaded = files.upload()



# 📊 Step 2: Read the uploaded CSV file

for filename in uploaded:

   df = pd.read_csv(io.BytesIO(uploaded[filename]))

   print(f"\n✅ File '{filename}' uploaded successfully!")
```

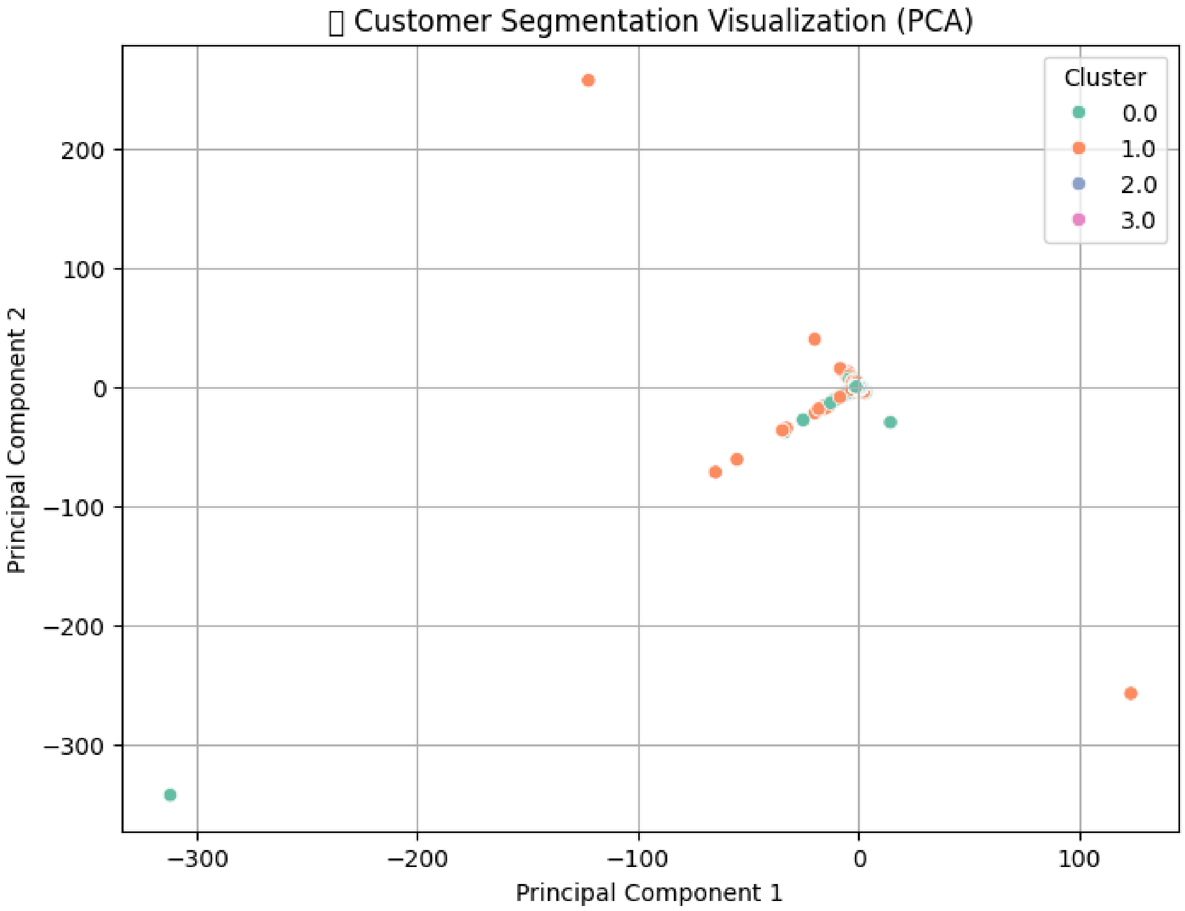
```
# 🔍 Step 3: Data Preprocessing
df_cleaned = df.dropna()  # remove rows with missing values
df_numeric = df_cleaned.select_dtypes(include=['float64', 'int64'])  # keep numeric columns


# 📏 Step 4: Feature Scaling
scaler = StandardScaler()
df_scaled = scaler.fit_transform(df_numeric)


# 📉 Step 5: Elbow Method to determine optimal clusters
wcss = []
for i in range(1, 11):
   kmeans = KMeans(n_clusters=i, random_state=42)
   kmeans.fit(df_scaled)
   wcss.append(kmeans.inertia_)


plt.figure(figsize=(8, 5))
plt.plot(range(1, 11), wcss, marker='o')
plt.title('📌 Elbow Method to Choose Optimal Clusters')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS (Inertia)')
plt.grid(True)
plt.show()
```

```
# 🧠 Step 6: Apply KMeans Clustering
optimal_k = 4  # Based on Elbow Method (change if needed)
kmeans = KMeans(n_clusters=optimal_k, random_state=42)
df_cleaned['Cluster'] = kmeans.fit_predict(df_scaled)

# 🔻 Step 7: Visualize Clusters using PCA
pca = PCA(n_components=2)
pca_components = pca.fit_transform(df_scaled)

# Create a DataFrame for PCA output
pca_df = pd.DataFrame(data=pca_components, columns=['PCA1', 'PCA2'])
pca_df['Cluster'] = df_cleaned['Cluster']

plt.figure(figsize=(8, 6))
sns.scatterplot(data=pca_df, x='PCA1', y='PCA2', hue='Cluster', palette='Set2')
plt.title('👥 Customer Segmentation Visualization (PCA)')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend(title='Cluster')
plt.grid(True)
plt.show()
```

Output



 Elbow Method to Choose Optimal Clusters
1e6
1.2
1.0
0.8
0.6
0.4
0.2
0.0
WCSS (Inertia)
2
4
6
8
10
Number of Clusters



Reference :

- **Dataset: Provided/uploaded by teacher.**
- **Libraries Used: `pandas`, `matplotlib`, `seaborn`, `sklearn`**
- **Tool: Google Colab**
- **Algorithms: KMeans clustering and PCA from `scikit-learn`**