Objective :

To cluster customers based on their purchasing behavior to uncover patterns and insights for targeted marketing strategies.

Steps Performed:

1. **Data Loading and Preprocessing :**

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

# Load datasets
customers = pd.read_csv('Customers.csv')
products = pd.read_csv('Products.csv')
transactions =
pd.read_csv('Transactions.csv')
```

```
# Merge datasets to create a unified view
merged_data =
transactions.merge(customers,
on='CustomerID').merge(products,
on='ProductID')

# Feature selection for clustering
clustering_features =
merged_data.groupby('CustomerID').agg(
{
    'TotalValue': 'sum',
    'Quantity': 'sum',
    'Price': 'mean'
}).reset_index()

# Normalize data for clustering
scaler = StandardScaler()
scaled_features =
scaler.fit_transform(clustering_features[['
TotalValue', 'Quantity', 'Price']])
```

2. **Finding Optimal Clusters Using Elbow Method :**

   ```
   # Elbow method to determine the optimal number of clusters
   inertia = []
   k_values = range(1, 11)

   for k in k_values:
       kmeans = KMeans(n_clusters=k, random_state=42)
       kmeans.fit(scaled_features)
       inertia.append(kmeans.inertia_)

   # Plot the Elbow Curve
   plt.figure(figsize=(8, 5))
   plt.plot(k_values, inertia, marker='o')
   plt.xlabel('Number of Clusters')
   plt.ylabel('Inertia')
   plt.title('Elbow Method for Optimal K')
   plt.show()
   ```

3. **Applying K-Means Clustering :**

   ```
   # Choose optimal number of clusters (e.g., k=4 based on the elbow plot)
   optimal_k = 4
   ```

```
kmeans = KMeans(n_clusters=optimal_k,
random_state=42)
clustering_features['Cluster'] =
kmeans.fit_predict(scaled_features)

# Add cluster labels to the dataset
clustered_data =
merged_data.merge(clustering_features[
['CustomerID', 'Cluster']],
on='CustomerID')

# Summary of clusters
cluster_summary =
clustered_data.groupby('Cluster').agg({
    'TotalValue': 'mean',
    'Quantity': 'mean',
    'Price': 'mean'
}).reset_index()
print(cluster_summary)
```

4. **Visualization of Clusters :**

```
sns.scatterplot(
    x=clustering_features['TotalValue'],
    y=clustering_features['Quantity'],
    hue=clustering_features['Cluster'],
    palette='viridis'
)
plt.title('Clusters of Customers')
plt.xlabel('Total Spending')
plt.ylabel('Quantity Purchased')
plt.legend(title='Cluster')
plt.show()
```

5. **Insights :**

Cluster 0: High spenders with low quantity purchases.

Cluster 1: Budget-conscious customers with medium purchases.

Cluster 2: Bulk buyers with average spending per purchase.

Cluster 3: Occasional customers with low overall activity.