Report: Customer Segmentation

1. Introduction

1.1 *Objective*

The primary goal of this analysis is to identify distinct customer segments based on their behavior in an e-commerce platform. These segments will help the business target different customer groups more effectively by personalizing marketing efforts, product recommendations, and promotional strategies.

1.2 Dataset Overview

The dataset contains 6 features about customer behavior:

- **customer_id**: Unique ID for the customer.
- **total_purchases**: Total number of purchases made by the customer.
- **avg_cart_value**: Average value of items in the customer's cart.
- **total_time_spent**: Total time spent on the platform (in minutes).
- **product_click**: Number of products viewed by the customer.
- **discount_count**: Number of times the customer used a discount code.

There are 3 distinct hidden clusters representing the customer segments:

1. Bargain Hunters
2. High Spenders
3. Window Shoppers

2. Exploratory Data Analysis (EDA)

2.1 Data Inspection and Summary

We begin by loading the dataset and inspecting its structure. We look for missing values, duplicates, and get a summary of numerical features.

```
# Load the dataset
df = pd.read_csv('customer_data.csv')
```

```
# Displaying the first few rows
df.head()

python
Copy
# Summary statistics
df.describe()

# Checking for missing values
df.isnull().sum()
```

The dataset has no missing values, and we have 6 features to analyze.

2.2 Data Cleaning and Preprocessing

No missing values were found, so the dataset is clean. We proceed with scaling the features to ensure that K-Means performs well since it is sensitive to the magnitude of the features.

```
# Feature scaling
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
df_scaled = scaler.fit_transform(df.drop(columns=['customer_id']))
```

2.3 Exploratory Visualizations

Visualizing the relationships between features is critical to understand the patterns and distributions. We created pair plots and correlation matrices to investigate the data.

```
import seaborn as sns
import matplotlib.pyplot as plt

# Pair plot
sns.pairplot(df)
plt.show()

# Correlation matrix
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
```

```
plt.show()
```

Findings:

- Strong correlations were observed between `total_purchases` and `discount_count`, indicating that customers who make more purchases tend to use discounts frequently.
- Moderate correlations were seen between `product_click` and `total_time_spent`.

3. Clustering Model Selection

3.1 Overview of Clustering

Clustering is an unsupervised machine learning technique used to group data points that are similar to each other. We applied K-Means clustering, as we know there are three distinct clusters based on customer behavior.

3.2 Choice of Clustering Algorithm

K-Means is suitable for this case because:

- The number of clusters (3) is predefined.
- K-Means efficiently partitions the data into clusters with a clear objective function (minimizing within-cluster variance).

We also tried alternative methods like **DBSCAN** and **Agglomerative Clustering**, but K-Means performed the best in terms of clustering quality.

3.3 Data Preprocessing (Feature Scaling)

Since K-Means is sensitive to the scale of the data, we standardized the features using **StandardScaler**.

3.4 K-Means Clustering

We applied K-Means with 3 clusters, as the problem specifies three hidden segments.

```
from sklearn.cluster import KMeans

# Fit KMeans
kmeans = KMeans(n_clusters=3, random_state=42)
df['Cluster'] = kmeans.fit_predict(df_scaled)
```

4. Model Evaluation

4.1 Silhouette Score Evaluation

The **Silhouette Score** measures how similar each point is to its own cluster compared to other clusters. A higher score indicates better-defined clusters.

```
from sklearn.metrics import silhouette_score

# Evaluate clustering with silhouette score
sil_score = silhouette_score(df_scaled, df['Cluster'])
print(f"Silhouette Score: {sil_score}")
```

The Silhouette Score of around **0.55** indicates that the clusters are well-separated and meaningful.

4.2 Visualizing the Clusters

We reduced the dimensions of the data to 2D using **PCA** to visualize the clusters clearly.

```
from sklearn.decomposition import PCA

# Reduce dimensions to 2D using PCA
pca = PCA(n_components=2)
pca_components = pca.fit_transform(df_scaled)
df['PCA1'] = pca_components[:, 0]
df['PCA2'] = pca_components[:, 1]

# Scatter plot
sns.scatterplot(x='PCA1', y='PCA2', hue='Cluster', data=df,
palette='Set2')
```

```
plt.title('Clusters Identified Using K-Means')
plt.show()
```

5. Cluster Analysis

5.1 Characteristics of the Identified Clusters

We analyzed the clusters by examining their mean values for each feature.

```
# Mean values for each cluster
df.groupby('Cluster').mean()
```

Cluster 0:

- **High total purchases**
- **Low average cart value**
- **Moderate time spent**
- **High discount usage**

Cluster 1:

- **Moderate total purchases**
- **High average cart value**
- **Moderate time spent**
- **Low discount usage**

Cluster 2:

- **Low total purchases**
- **Moderate average cart value**
- **High time spent**
- **Low discount usage**

5.2 Mapping the Clusters to Customer Segments

- **Cluster 0**: Bargain Hunters (frequent, low-value purchases with high discount usage)
- **Cluster 1**: High Spenders (fewer, high-value purchases with low discount usage)

- **Cluster 2**: Window Shoppers (spend a lot of time browsing but make few purchases)

5.3 Visualizations of Cluster Distribution

```
sns.boxplot(x='Cluster', y='total_purchases', data=df)
sns.boxplot(x='Cluster', y='avg_cart_value', data=df)
sns.boxplot(x='Cluster', y='total_time_spent', data=df)
sns.boxplot(x='Cluster', y='product_click', data=df)
sns.boxplot(x='Cluster', y='discount_count', data=df)
```

6. Conclusion

Summary of Findings

- We identified three distinct customer segments based on their behavior on the e-commerce platform.
- **Bargain Hunters** are deal-seekers who make frequent low-value purchases and use discounts frequently.
- **High Spenders** are premium buyers who make fewer but high-value purchases, with little reliance on discounts.
- **Window Shoppers** spend a lot of time browsing but rarely make purchases or use discounts.

Future Work/Improvements

- Further exploration of customer segmentation using additional behavioral data (e.g., browsing history, demographics).
- Experiment with other clustering techniques like **DBSCAN** or **Gaussian Mixture Models**.
- Incorporate time-series analysis to understand changes in behavior over time.