

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder, StandardScaler, OneHotEncoder
from sklearn.cluster import KMeans, AgglomerativeClustering
from sklearn.metrics import silhouette_score
from sklearn.decomposition import PCA
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.model_selection import GridSearchCV

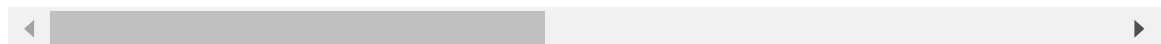
import warnings
warnings.filterwarnings('ignore')
```

```
In [2]: # Load the dataset
file_path = '/Users/khai/Library/Mobile Documents/com~apple~CloudDocs/Documents/
data = pd.read_csv(file_path)
data.head()
data
```

```
Out[2]:
```

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size
0	1	55	Male	Blouse	Clothing	53	Kentucky	L
1	2	19	Male	Sweater	Clothing	64	Maine	L
2	3	50	Male	Jeans	Clothing	73	Massachusetts	S
3	4	21	Male	Sandals	Footwear	90	Rhode Island	M
4	5	45	Male	Blouse	Clothing	49	Oregon	M
...
3895	3896	40	Female	Hoodie	Clothing	28	Virginia	L
3896	3897	52	Female	Backpack	Accessories	49	Iowa	L
3897	3898	46	Female	Belt	Accessories	33	New Jersey	L
3898	3899	44	Female	Shoes	Footwear	77	Minnesota	S
3899	3900	52	Female	Handbag	Accessories	81	California	M

3900 rows × 18 columns



```
In [3]: # Data Preprocessing
data.drop('Customer ID', axis=1, inplace=True)

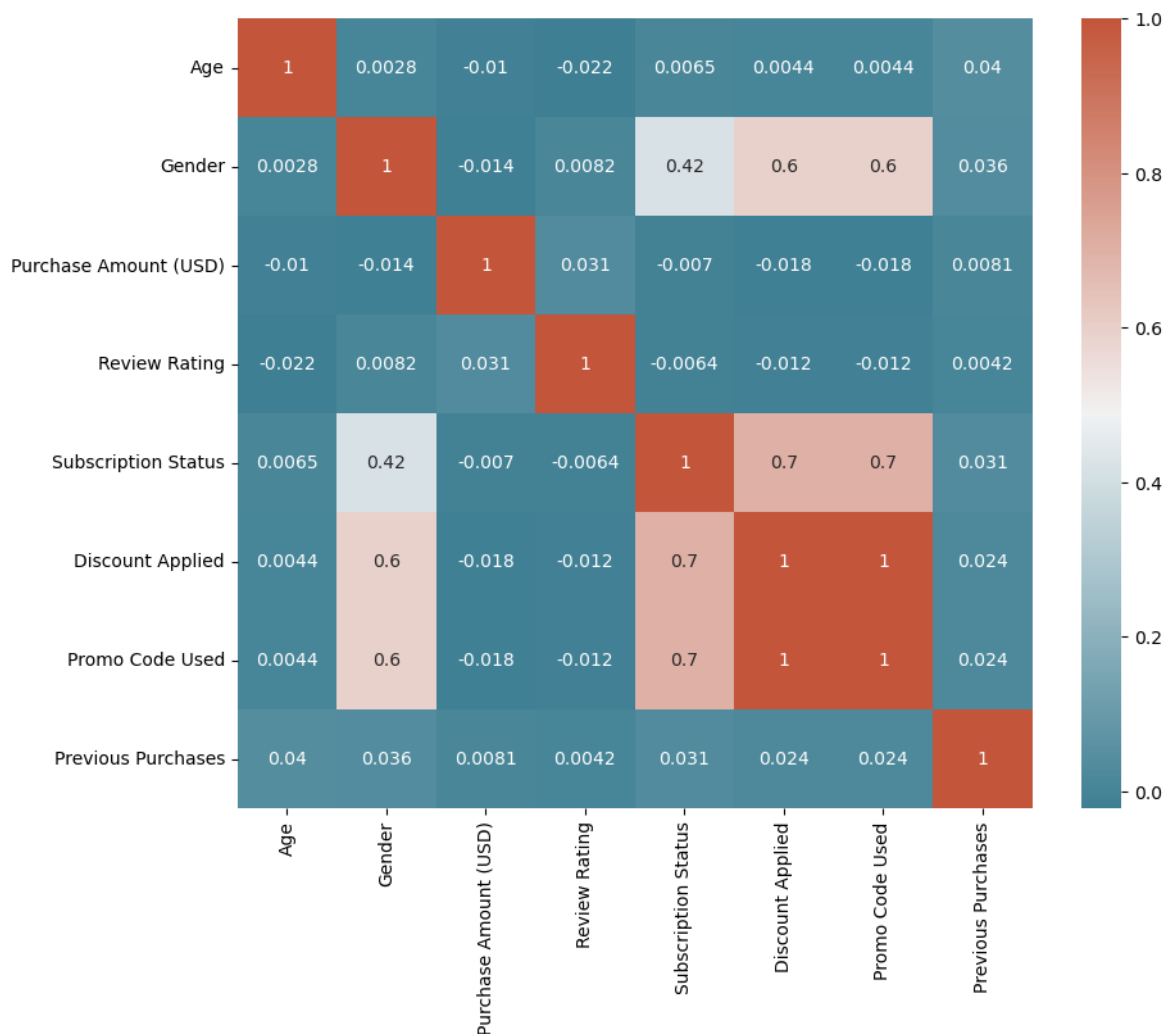
# There is no missing data
```

```
# Label Encoding for binary columns
le = LabelEncoder()
binary_columns = ['Gender', 'Subscription Status', 'Discount Applied', 'Promo Co
for col in binary_columns:
    data[col] = le.fit_transform(data[col])

numerical_columns = ['Age', 'Purchase Amount (USD)', 'Review Rating', 'Previous
categorical_columns = ['Item Purchased', 'Category', 'Location', 'Size', 'Color',
    'Shipping Type', 'Payment Method', 'Frequency of Purchase
```

```
In [4]: corr = data.loc[:, ~data.columns.isin(categorical_columns)].corr() #exclude cate

plt.subplots(figsize=(10,8))
sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns, annot=True)
plt.show()
```



```
In [5]: # One-Hot Encoding for non-binary categorical columns
data = pd.get_dummies(data, columns=categorical_columns)
```

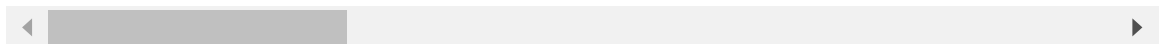
```
In [6]: # Standard Scaler for numerical columns
scaler = StandardScaler()
data[numerical_columns] = scaler.fit_transform(data[numerical_columns])
```

```
In [7]: # Display the first few rows of the preprocessed dataset
data
```

Out[7]:

	Age	Gender	Purchase Amount (USD)	Review Rating	Subscription Status	Discount Applied	Promo Code Used	Previous Purchases
0	0.718913	1	-0.285629	-0.907584	1	1	1	-0.78583
1	-1.648629	1	0.178852	-0.907584	1	1	1	-1.61659
2	0.390088	1	0.558882	-0.907584	1	1	1	-0.16278
3	-1.517099	1	1.276716	-0.349027	1	1	1	1.63710
4	0.061263	1	-0.454531	-1.466141	1	1	1	0.39102
...
3895	-0.267563	0	-1.341267	0.628448	0	0	0	0.46029
3896	0.521618	0	-0.454531	1.047366	0	0	0	1.08329
3897	0.127028	0	-1.130139	-1.186862	0	0	0	-0.09350
3898	-0.004502	0	0.727784	0.069891	0	0	0	-0.09350
3899	0.521618	0	0.896686	-0.907584	0	0	0	0.52947

3900 rows × 139 columns



```
In [8]: # Export the DataFrame to a CSV file
data.to_csv("shopping_behavior_updated1.csv", index=False)
```

```
In [9]: # Load the dataset
file_path = 'shopping_behavior_updated1.csv'
data = pd.read_csv(file_path)
```

Using the silhouette score to find the optimal number of clusters

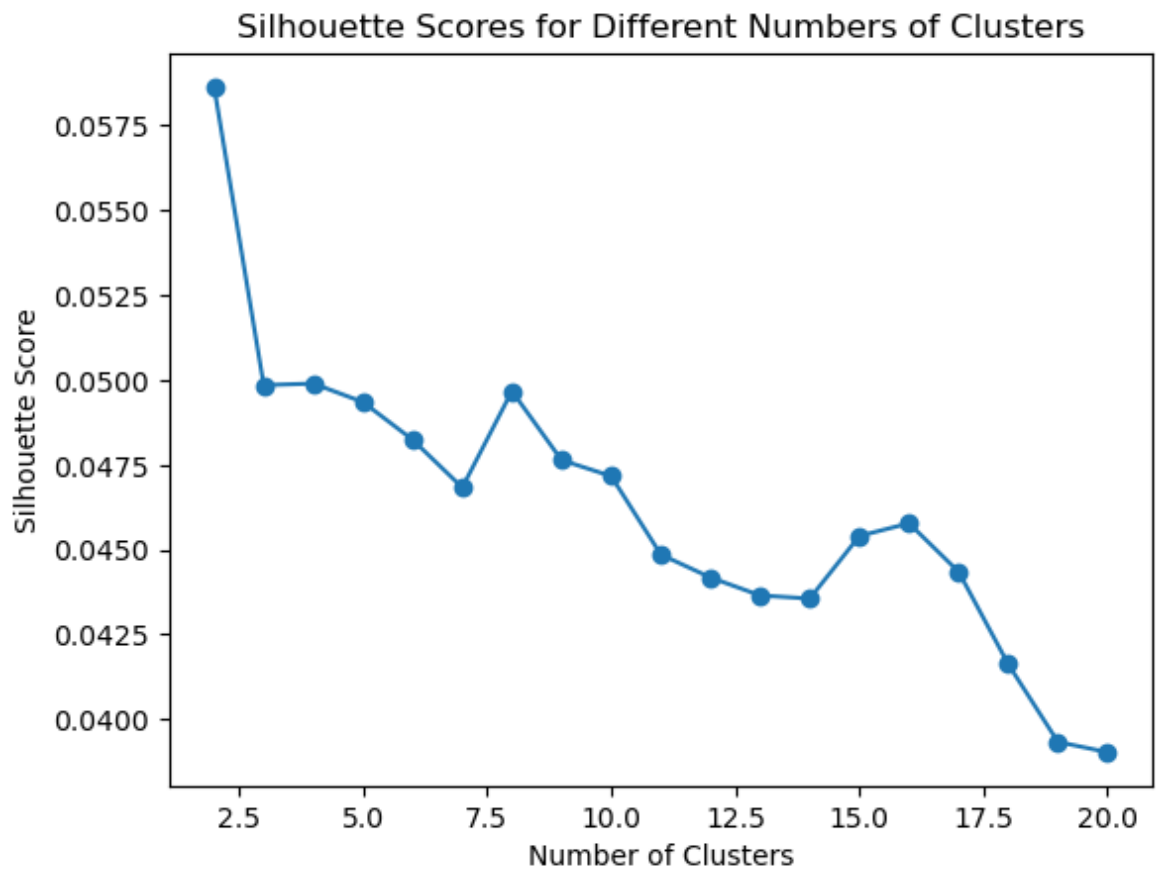
```
In [10]: from sklearn.metrics import silhouette_score

def calculate_silhouette_scores(data, range_clusters):
    silhouette_scores = [] # List to store silhouette scores for different clusters
    for cluster in range_clusters:
        kmeans = KMeans(n_clusters=cluster, init='k-means++', n_init=10)
        kmeans.fit(data)
        labels = kmeans.labels_
        silhouette_avg = silhouette_score(data, labels)
        silhouette_scores.append(silhouette_avg)
    return silhouette_scores

range_clusters = range(2, 21) # Silhouette score requires at least 2 clusters
silhouette_scores = calculate_silhouette_scores(data, range_clusters)

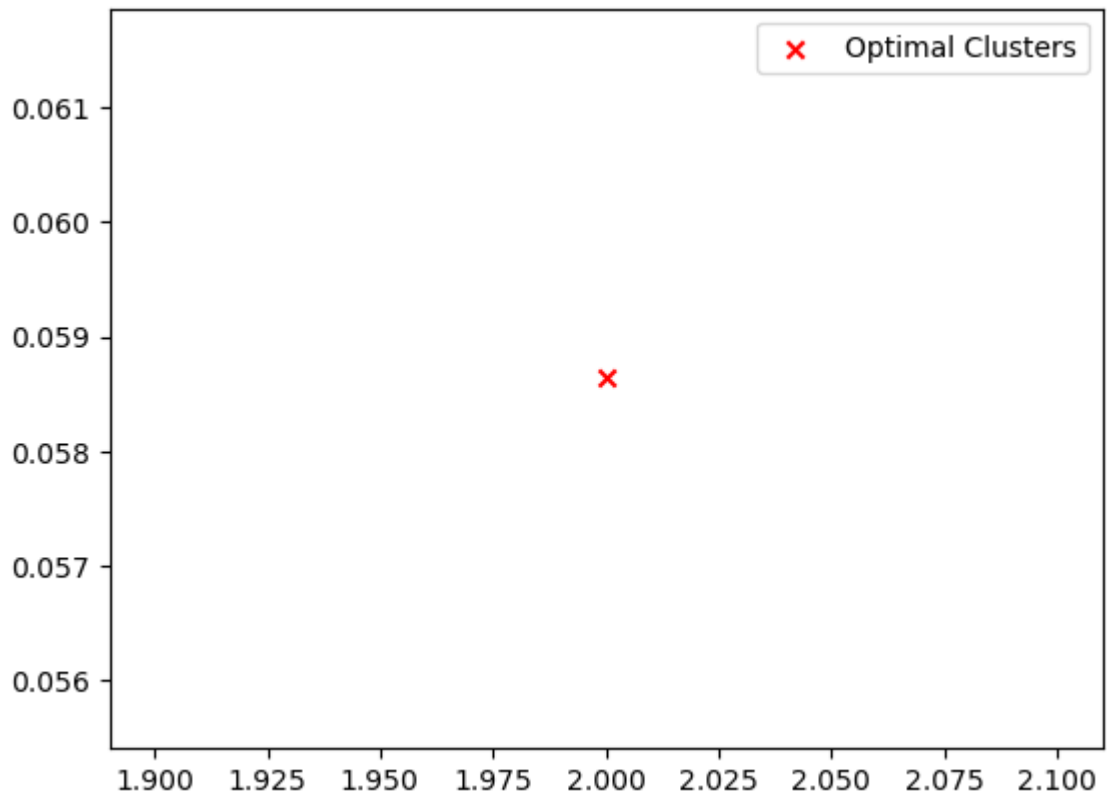
# Plotting the silhouette scores
plt.plot(range_clusters, silhouette_scores, marker='o')
plt.title('Silhouette Scores for Different Numbers of Clusters')
plt.xlabel('Number of Clusters')
```

```
plt.ylabel('Silhouette Score')  
plt.show()
```



```
In [11]: # Find the optimal number of clusters  
optimal_clusters = range_clusters[np.argmax(silhouette_scores)]  
print("Optimal Number of Clusters:", optimal_clusters)  
  
# Highlight the optimal point on the plot  
plt.scatter(optimal_clusters, max(silhouette_scores), color='red', marker='x', 1  
plt.legend()  
plt.show()
```

Optimal Number of Clusters: 2



K-Means

```
In [12]: # Apply PCA to reduce dimensions to 2
pca = PCA(n_components=2)
data_pca = pca.fit_transform(data)

# Train K-Means on the PCA-transformed data
kmeans = KMeans(n_clusters=2, init='k-means++', n_init=10, random_state=42)
y_kmeans = kmeans.fit_predict(data_pca)
kmeans
```

```
Out[12]: ▼ KMeans
KMeans(n_clusters=2, n_init=10, random_state=42)
```

```
In [13]: kmeans.inertia_
```

```
Out[13]: 5364.563495712744
```

```
In [14]: kmeans.cluster_centers_
```

```
Out[14]: array([[ -0.63066199,  0.56041181],
                [ 0.62872447, -0.55869011]])
```

```
In [15]: kmeans.n_iter_
```

```
Out[15]: 8
```

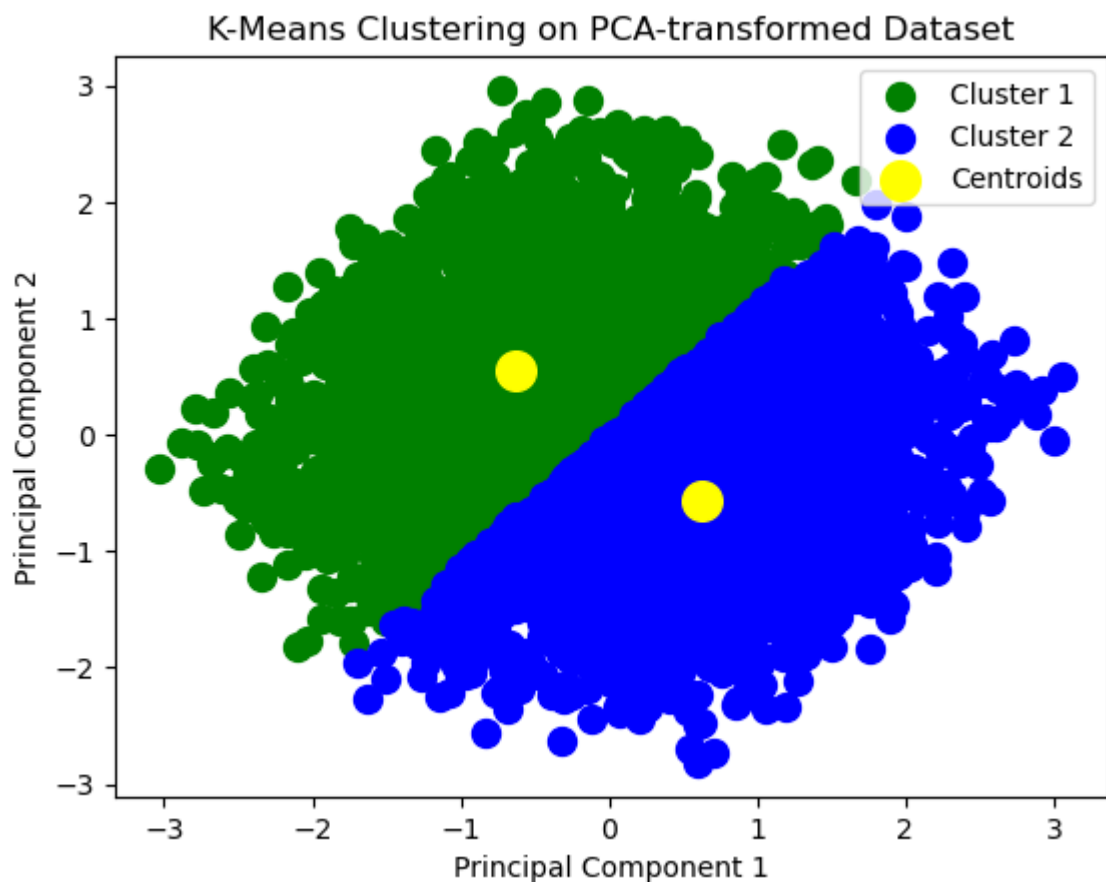
```
In [16]: kmeans_silhouette = silhouette_score(data, kmeans.labels_).round(2)
print(f"Silhouette Score: {kmeans_silhouette}")
```

Silhouette Score: 0.05

```
In [17]: # Visualize the clusters
plt.scatter(data_pca[y_kmeans == 0, 0], data_pca[y_kmeans == 0, 1], s=100, c='green')
plt.scatter(data_pca[y_kmeans == 1, 0], data_pca[y_kmeans == 1, 1], s=100, c='blue')

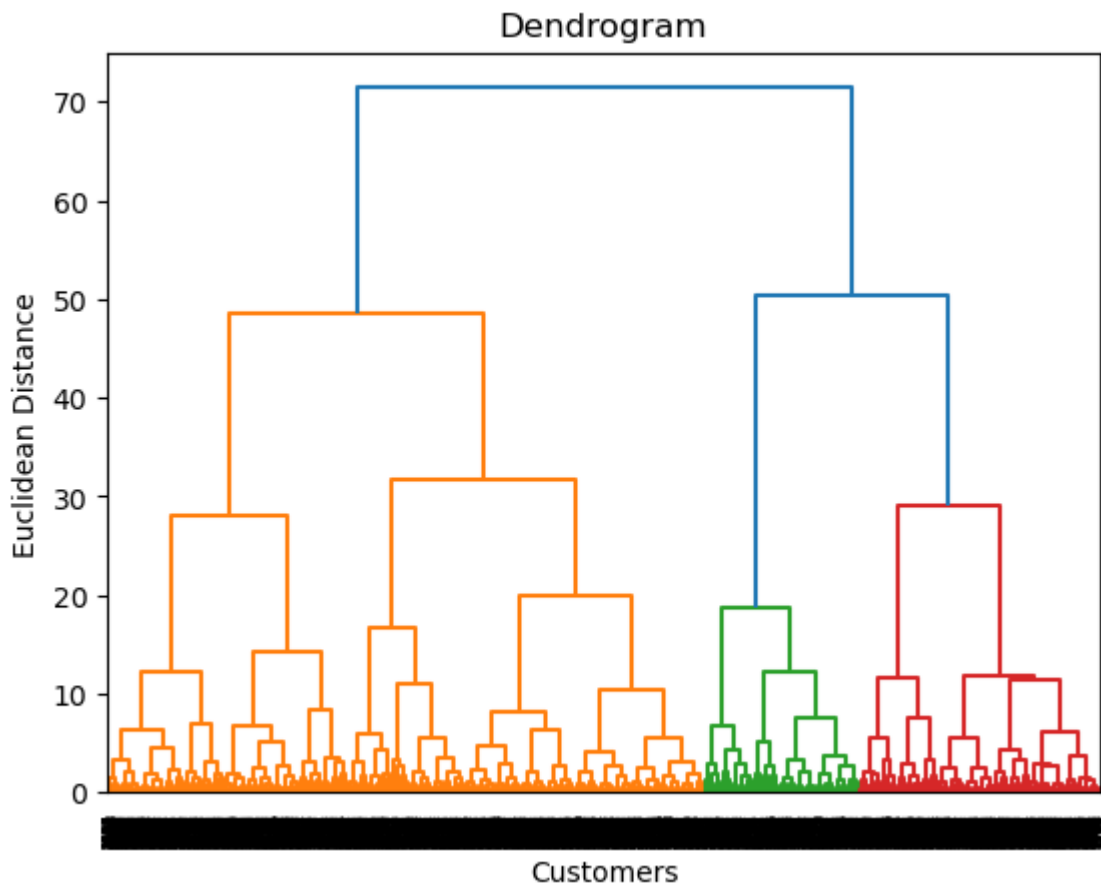
# Plot the centroids in the PCA space
centers_pca = kmeans.cluster_centers_
plt.scatter(centers_pca[:, 0], centers_pca[:, 1], s=200, c='yellow', label='Centroids')

# Finalize the plot
plt.title('K-Means Clustering on PCA-transformed Dataset')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend()
plt.show()
```



Hierarchical clustering

```
In [18]: import scipy.cluster.hierarchy as sch
# Create the dendrogram using the Ward's method on the PCA-transformed data
dendrogram = sch.dendrogram(sch.linkage(data_pca, method="ward"))
plt.title("Dendrogram")
plt.xlabel("Customers")
plt.ylabel("Euclidean Distance")
plt.show()
```



To determine the number of clusters from a dendrogram, we would typically look for the longest vertical lines that are not crossed by any horizontal line (extended across the entire width of the plot).

There are 2 very prominent gaps:

- The topmost blue line suggests the data could be split into 2 clusters.
- Another significant gap can be seen a bit lower, where the dendrogram branches into three main arms.

Based on these observations, it appears there could be 2 or 3 clusters, depending on where we choose to "cut" the dendrogram based on the specific context and domain knowledge.

**For the sake of simplicity we assume there are 2 clusters.
Now, let's plot the Agglomerative Clustering**

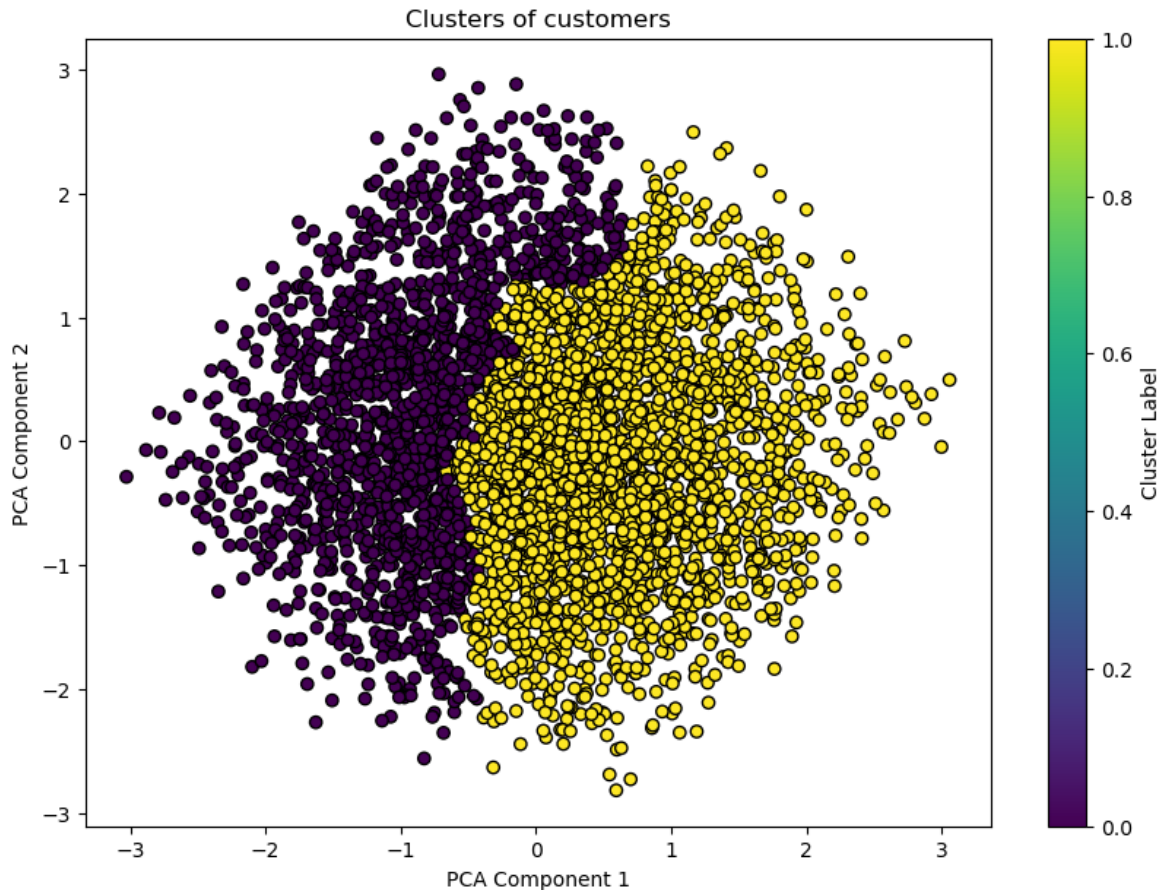
```
In [19]: from sklearn.cluster import AgglomerativeClustering
cluster=AgglomerativeClustering(n_clusters=2,metric="euclidean",linkage="ward")
```

```
In [20]: y_hc = cluster.fit_predict(data_pca)
```

```
In [21]: silhouette_avg = silhouette_score(data_pca, y_hc)
print(f"Silhouette Score: {silhouette_avg}")
```

Silhouette Score: 0.30703113304411606

```
In [22]: # Visualize the clusters
plt.figure(figsize=(10, 7))
plt.scatter(data_pca[:, 0], data_pca[:, 1], c=y_hc, cmap='viridis', edgecolors='k')
plt.title('Clusters of customers')
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.colorbar(label='Cluster Label')
plt.show()
```



Comparing the clustering algorithms

In this comparative study of clustering algorithms applied to a dataset on shopping habits, the distinct characteristics of each method yielded varied insights. The KMeans algorithm, which is optimal for spherical cluster shapes and when the cluster count is pre-set, produced a silhouette score of 0.05. This score, on the lower end, suggests a lack of well-defined, spherical clusters in the dataset. Furthermore, Agglomerative Clustering, offering a hierarchical perspective of data clustering and excelling in situations where inter-data point relationships are pivotal, demonstrated a markedly better silhouette score of 0.3. This score suggests a much clearer and meaningful clustering structure, pointing to the potential significance of hierarchical relationships in the dataset.

The insights derived from these clustering methods regarding the dataset highlight different dimensions of customer segmentation and purchasing patterns. KMeans likely pinpointed broad customer groups, albeit with limited clarity. In contrast, Agglomerative Clustering, with its emphasis on layered relationships, likely unveiled more intricate patterns in consumer preferences and behaviors. Therefore, these methods offer a

holistic view of the patterns in the shopping behavior data, each contributing distinct perspectives that, collectively, facilitate a better comprehension of customer segmentation strategies.

For simplicity and easier interpretation for non-technical employees (i.e. Marketing, Sales, Business Development, etc) we choose the KMeans clustering algorithm for the ecommerce platform customer segmentation.

Interpretation of Cluster 0 and Cluster 1 for KMeans

```
In [23]: # Undo Standard Scaling for numerical columns
data_undo = data.copy()
data_undo[numerical_columns] = scaler.inverse_transform(data_undo[numerical_colu

# 1. Add cluster labels to the original data_undo
data_undo['Cluster'] = y_kmeans
```

```
In [24]: # 2. Explore cluster statistics
cluster_statistics = data_undo.groupby('Cluster').mean()
print("Cluster Statistics:")
for i in cluster_statistics.columns:
    print(cluster_statistics[i])
```

```

Cluster Statistics:
Cluster
0    52.102051
1    36.034872
Name: Age, dtype: float64
Cluster
0    0.702564
1    0.657436
Name: Gender, dtype: float64
Cluster
0    61.421026
1    58.107692
Name: Purchase Amount (USD), dtype: float64
Cluster
0    3.731026
1    3.768872
Name: Review Rating, dtype: float64
Cluster
0    0.292821
1    0.247179
Name: Subscription Status, dtype: float64
Cluster
0    0.451795
1    0.408205
Name: Discount Applied, dtype: float64
Cluster
0    0.451795
1    0.408205
Name: Promo Code Used, dtype: float64
Cluster
0    34.745641
1    15.957436
Name: Previous Purchases, dtype: float64
Cluster
0    0.034359
1    0.038974
Name: Item Purchased_Backpack, dtype: float64
Cluster
0    0.038462
1    0.044103
Name: Item Purchased_Belt, dtype: float64
Cluster
0    0.047179
1    0.040513
Name: Item Purchased_Blouse, dtype: float64
Cluster
0    0.039487
1    0.034359
Name: Item Purchased_Boots, dtype: float64
Cluster
0    0.042051
1    0.040513
Name: Item Purchased_Coat, dtype: float64
Cluster
0    0.047179
1    0.037949
Name: Item Purchased_Dress, dtype: float64
Cluster
0    0.035385
1    0.036410

```

Name: Item Purchased_Gloves, dtype: float64
Cluster
0 0.042564
1 0.035897
Name: Item Purchased_Handbag, dtype: float64
Cluster
0 0.038462
1 0.040513
Name: Item Purchased_Hat, dtype: float64
Cluster
0 0.036410
1 0.041026
Name: Item Purchased_Hoodie, dtype: float64
Cluster
0 0.040513
1 0.043077
Name: Item Purchased_Jacket, dtype: float64
Cluster
0 0.026667
1 0.036923
Name: Item Purchased_Jeans, dtype: float64
Cluster
0 0.052821
1 0.034872
Name: Item Purchased_Jewelry, dtype: float64
Cluster
0 0.043590
1 0.044103
Name: Item Purchased_Pants, dtype: float64
Cluster
0 0.040513
1 0.041538
Name: Item Purchased_Sandals, dtype: float64
Cluster
0 0.043590
1 0.036923
Name: Item Purchased_Scarf, dtype: float64
Cluster
0 0.039487
1 0.047179
Name: Item Purchased_Shirt, dtype: float64
Cluster
0 0.040513
1 0.036410
Name: Item Purchased_Shoes, dtype: float64
Cluster
0 0.040513
1 0.040000
Name: Item Purchased_Short, dtype: float64
Cluster
0 0.037949
1 0.043077
Name: Item Purchased_Skirt, dtype: float64
Cluster
0 0.034872
1 0.039487
Name: Item Purchased_Sneakers, dtype: float64
Cluster
0 0.039487
1 0.042051

Name: Item Purchased_Socks, dtype: float64
Cluster
0 0.043077
1 0.039487
Name: Item Purchased_Sunglasses, dtype: float64
Cluster
0 0.040000
1 0.044103
Name: Item Purchased_Sweater, dtype: float64
Cluster
0 0.034872
1 0.040513
Name: Item Purchased_T-shirt, dtype: float64
Cluster
0 0.328718
1 0.307179
Name: Category_Accessories, dtype: float64
Cluster
0 0.433333
1 0.457436
Name: Category_Clothing, dtype: float64
Cluster
0 0.155385
1 0.151795
Name: Category_Footwear, dtype: float64
Cluster
0 0.082564
1 0.083590
Name: Category_Outerwear, dtype: float64
Cluster
0 0.025128
1 0.020513
Name: Location_Alabama, dtype: float64
Cluster
0 0.021026
1 0.015897
Name: Location_Alaska, dtype: float64
Cluster
0 0.018974
1 0.014359
Name: Location_Arizona, dtype: float64
Cluster
0 0.024103
1 0.016410
Name: Location_Arkansas, dtype: float64
Cluster
0 0.023590
1 0.025128
Name: Location_California, dtype: float64
Cluster
0 0.017436
1 0.021026
Name: Location_Colorado, dtype: float64
Cluster
0 0.018974
1 0.021026
Name: Location_Connecticut, dtype: float64
Cluster
0 0.021026
1 0.023077

```
Name: Location_Delaware, dtype: float64
Cluster
0    0.016410
1    0.018462
Name: Location_Florida, dtype: float64
Cluster
0    0.020513
1    0.020000
Name: Location_Georgia, dtype: float64
Cluster
0    0.017949
1    0.015385
Name: Location_Hawaii, dtype: float64
Cluster
0    0.021026
1    0.026667
Name: Location_Idaho, dtype: float64
Cluster
0    0.021538
1    0.025641
Name: Location_Illinois, dtype: float64
Cluster
0    0.022564
1    0.017949
Name: Location_Indiana, dtype: float64
Cluster
0    0.017436
1    0.017949
Name: Location_Iowa, dtype: float64
Cluster
0    0.015385
1    0.016923
Name: Location_Kansas, dtype: float64
Cluster
0    0.019487
1    0.021026
Name: Location_Kentucky, dtype: float64
Cluster
0    0.021538
1    0.021538
Name: Location_Louisiana, dtype: float64
Cluster
0    0.016923
1    0.022564
Name: Location_Maine, dtype: float64
Cluster
0    0.025641
1    0.018462
Name: Location_Maryland, dtype: float64
Cluster
0    0.016923
1    0.020000
Name: Location_Massachusetts, dtype: float64
Cluster
0    0.020000
1    0.017436
Name: Location_Michigan, dtype: float64
Cluster
0    0.024103
1    0.021026
```

```
Name: Location_Minnesota, dtype: float64
Cluster
0    0.020000
1    0.021026
Name: Location_Mississippi, dtype: float64
Cluster
0    0.023590
1    0.017949
Name: Location_Missouri, dtype: float64
Cluster
0    0.025128
1    0.024103
Name: Location_Montana, dtype: float64
Cluster
0    0.022051
1    0.022564
Name: Location_Nebraska, dtype: float64
Cluster
0    0.022564
1    0.022051
Name: Location_Nevada, dtype: float64
Cluster
0    0.019487
1    0.016923
Name: Location_New Hampshire, dtype: float64
Cluster
0    0.018462
1    0.015897
Name: Location_New Jersey, dtype: float64
Cluster
0    0.022051
1    0.019487
Name: Location_New Mexico, dtype: float64
Cluster
0    0.023590
1    0.021026
Name: Location_New York, dtype: float64
Cluster
0    0.021026
1    0.018974
Name: Location_North Carolina, dtype: float64
Cluster
0    0.018974
1    0.023590
Name: Location_North Dakota, dtype: float64
Cluster
0    0.021026
1    0.018462
Name: Location_Ohio, dtype: float64
Cluster
0    0.014872
1    0.023590
Name: Location_Oklahoma, dtype: float64
Cluster
0    0.020000
1    0.017949
Name: Location_Oregon, dtype: float64
Cluster
0    0.018974
1    0.018974
```

```
Name: Location_Pennsylvania, dtype: float64
Cluster
0    0.016923
1    0.015385
Name: Location_Rhode Island, dtype: float64
Cluster
0    0.022564
1    0.016410
Name: Location_South Carolina, dtype: float64
Cluster
0    0.016923
1    0.018974
Name: Location_South Dakota, dtype: float64
Cluster
0    0.020000
1    0.019487
Name: Location_Tennessee, dtype: float64
Cluster
0    0.014359
1    0.025128
Name: Location_Texas, dtype: float64
Cluster
0    0.020000
1    0.01641
Name: Location_Utah, dtype: float64
Cluster
0    0.020000
1    0.02359
Name: Location_Vermont, dtype: float64
Cluster
0    0.015385
1    0.024103
Name: Location_Virginia, dtype: float64
Cluster
0    0.018462
1    0.018974
Name: Location_Washington, dtype: float64
Cluster
0    0.016923
1    0.024615
Name: Location_West Virginia, dtype: float64
Cluster
0    0.018462
1    0.020000
Name: Location_Wisconsin, dtype: float64
Cluster
0    0.020513
1    0.015897
Name: Location_Wyoming, dtype: float64
Cluster
0    0.274359
1    0.265641
Name: Size_L, dtype: float64
Cluster
0    0.453846
1    0.446154
Name: Size_M, dtype: float64
Cluster
0    0.16359
1    0.17641
```

Name: Size_S, dtype: float64
Cluster
0 0.108205
1 0.111795
Name: Size_XL, dtype: float64
Cluster
0 0.042051
1 0.033333
Name: Color_Beige, dtype: float64
Cluster
0 0.045128
1 0.040513
Name: Color_Black, dtype: float64
Cluster
0 0.035385
1 0.042564
Name: Color_Blue, dtype: float64
Cluster
0 0.034872
1 0.037436
Name: Color_Brown, dtype: float64
Cluster
0 0.040000
1 0.038462
Name: Color_Charcoal, dtype: float64
Cluster
0 0.040000
1 0.045128
Name: Color_Cyan, dtype: float64
Cluster
0 0.032308
1 0.038462
Name: Color_Gold, dtype: float64
Cluster
0 0.047692
1 0.033846
Name: Color_Gray, dtype: float64
Cluster
0 0.037949
1 0.048718
Name: Color_Green, dtype: float64
Cluster
0 0.037949
1 0.037436
Name: Color_Indigo, dtype: float64
Cluster
0 0.040513
1 0.034872
Name: Color_Lavender, dtype: float64
Cluster
0 0.037436
1 0.040513
Name: Color_Magenta, dtype: float64
Cluster
0 0.038974
1 0.042051
Name: Color_Maroon, dtype: float64
Cluster
0 0.041538
1 0.049231

Name: Color_Olive, dtype: float64
Cluster
0 0.041026
1 0.037949
Name: Color_Orange, dtype: float64
Cluster
0 0.039487
1 0.036923
Name: Color_Peach, dtype: float64
Cluster
0 0.037436
1 0.041026
Name: Color_Pink, dtype: float64
Cluster
0 0.036923
1 0.040513
Name: Color_Purple, dtype: float64
Cluster
0 0.037949
1 0.037949
Name: Color_Red, dtype: float64
Cluster
0 0.046667
1 0.042051
Name: Color_Silver, dtype: float64
Cluster
0 0.043077
1 0.045128
Name: Color_Teal, dtype: float64
Cluster
0 0.041538
1 0.032821
Name: Color_Turquoise, dtype: float64
Cluster
0 0.042564
1 0.042564
Name: Color_Violet, dtype: float64
Cluster
0 0.043590
1 0.029231
Name: Color_White, dtype: float64
Cluster
0 0.037949
1 0.051282
Name: Color_Yellow, dtype: float64
Cluster
0 0.248205
1 0.251795
Name: Season_Fall, dtype: float64
Cluster
0 0.261538
1 0.250769
Name: Season_Spring, dtype: float64
Cluster
0 0.244615
1 0.245128
Name: Season_Summer, dtype: float64
Cluster
0 0.245641
1 0.252308

Name: Season_Winter, dtype: float64
Cluster
0 0.170256
1 0.151282

Name: Shipping Type_2-Day Shipping, dtype: float64
Cluster
0 0.171282
1 0.160000

Name: Shipping Type_Express, dtype: float64
Cluster
0 0.164103
1 0.182051

Name: Shipping Type_Free Shipping, dtype: float64
Cluster
0 0.156923
1 0.175385

Name: Shipping Type_Next Day Air, dtype: float64
Cluster
0 0.180513
1 0.154872

Name: Shipping Type_Standard, dtype: float64
Cluster
0 0.156923
1 0.176410

Name: Shipping Type_Store Pickup, dtype: float64
Cluster
0 0.146154
1 0.167692

Name: Payment Method_Bank Transfer, dtype: float64
Cluster
0 0.168718
1 0.174872

Name: Payment Method_Cash, dtype: float64
Cluster
0 0.180000
1 0.164103

Name: Payment Method_Credit Card, dtype: float64
Cluster
0 0.165641
1 0.160513

Name: Payment Method_Debit Card, dtype: float64
Cluster
0 0.172308
1 0.174872

Name: Payment Method_PayPal, dtype: float64
Cluster
0 0.167179
1 0.157949

Name: Payment Method_Venmo, dtype: float64
Cluster
0 0.151282
1 0.142051

Name: Frequency of Purchases_Annually, dtype: float64
Cluster
0 0.133846
1 0.146667

Name: Frequency of Purchases_Bi-Weekly, dtype: float64
Cluster
0 0.143590
1 0.155897

Name: Frequency of Purchases_Every 3 Months, dtype: float64

Cluster

0 0.134359

1 0.143590

Name: Frequency of Purchases_Fortnightly, dtype: float64

Cluster

0 0.140513

1 0.143077

Name: Frequency of Purchases_Monthly, dtype: float64

Cluster

0 0.155385

1 0.133333

Name: Frequency of Purchases_Quarterly, dtype: float64

Cluster

0 0.141026

1 0.135385

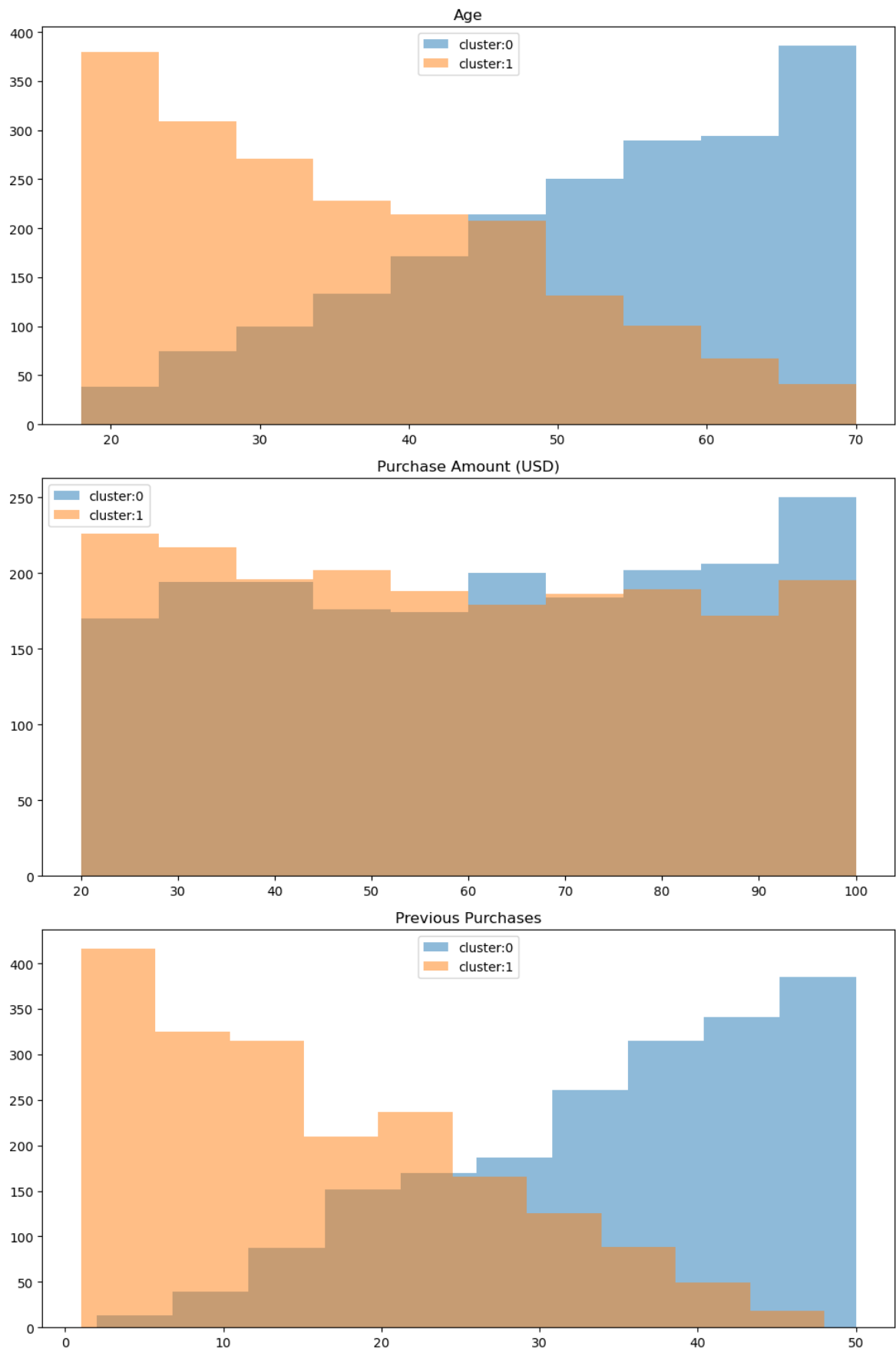
Name: Frequency of Purchases_Weekly, dtype: float64

```
In [25]: # 3. Visualize original features by cluster
plt.figure(figsize=(10, 15))

columns_to_visualize = ['Age', 'Purchase Amount (USD)', 'Previous Purchases']

for i, col_name in enumerate(columns_to_visualize):
    plt.subplot(len(columns_to_visualize), 1, i+1)
    for cluster in range(2):
        feature_data = data_undo.loc[data_undo['Cluster'] == cluster, col_name]
        plt.hist(feature_data, alpha=0.5, label=f'cluster:{cluster}')
    plt.title(col_name, fontsize=12)
    plt.legend()

plt.tight_layout()
plt.show()
```



```
In [26]: data_undo['Review Rating'] = data_undo['Review Rating'].astype(int) # Ensure th

# Create a pivot table to count the number of occurrences of each rating value i
pivot_table = pd.crosstab(data_undo['Cluster'], data_undo['Review Rating'], marg

# Create a stacked bar chart
plt.figure(figsize=(10, 5))
```

```

cluster_0 = pivot_table.iloc[0]
cluster_1 = pivot_table.iloc[1]
cluster_0.plot(kind='bar', alpha=0.5, color='deepskyblue', label='Cluster 0')
cluster_1.plot(kind='bar', alpha=0.5, color='orange', label='Cluster 1', bottom=

# Add a Legend and a title
plt.legend()
plt.title('Distribution of Review Rating by cluster')
plt.xlabel('Review Rating')
plt.ylabel('Count')
plt.xticks(np.arange(1, 6))

plt.tight_layout()
plt.show()

```



```

In [27]: # Get a list of binary columns (except column 'Cluster')
binary_cols = [col for col in data_undo.columns if data_undo[col].nunique() == 2]

# Calculate statistics for each binary column for cluster 0 and cluster 1
for col in binary_cols:
    clust0 = data_undo.loc[data_undo['Cluster'] == 0, col]
    clust1 = data_undo.loc[data_undo['Cluster'] == 1, col]
    print()
    print(f"# '{col}' Statistics:")
    print(f"Cluster 0:")
    print(f"  Mean: {clust0.mean():.4f}")
    print(f"  Median: {clust0.median():.4f}")
    print(f"  Standard Deviation: {clust0.std():.4f}")
    print(f"  Percentage of True values: {clust0.mean()*100:.1f}%")
    print(f"Cluster 1:")
    print(f"  Mean: {clust1.mean():.4f}")
    print(f"  Median: {clust1.median():.4f}")
    print(f"  Standard Deviation: {clust1.std():.4f}")
    print(f"  Percentage of True values: {clust1.mean()*100:.1f}%")

```

```
# 'Gender' Statistics:
Cluster 0:
  Mean: 0.7026
  Median: 1.0000
  Standard Deviation: 0.4572
  Percentage of True values: 70.3%
Cluster 1:
  Mean: 0.6574
  Median: 1.0000
  Standard Deviation: 0.4747
  Percentage of True values: 65.7%

# 'Subscription Status' Statistics:
Cluster 0:
  Mean: 0.2928
  Median: 0.0000
  Standard Deviation: 0.4552
  Percentage of True values: 29.3%
Cluster 1:
  Mean: 0.2472
  Median: 0.0000
  Standard Deviation: 0.4315
  Percentage of True values: 24.7%

# 'Discount Applied' Statistics:
Cluster 0:
  Mean: 0.4518
  Median: 0.0000
  Standard Deviation: 0.4978
  Percentage of True values: 45.2%
Cluster 1:
  Mean: 0.4082
  Median: 0.0000
  Standard Deviation: 0.4916
  Percentage of True values: 40.8%

# 'Promo Code Used' Statistics:
Cluster 0:
  Mean: 0.4518
  Median: 0.0000
  Standard Deviation: 0.4978
  Percentage of True values: 45.2%
Cluster 1:
  Mean: 0.4082
  Median: 0.0000
  Standard Deviation: 0.4916
  Percentage of True values: 40.8%

# 'Item Purchased_Backpack' Statistics:
Cluster 0:
  Mean: 0.0344
  Median: 0.0000
  Standard Deviation: 0.1822
  Percentage of True values: 3.4%
Cluster 1:
  Mean: 0.0390
  Median: 0.0000
  Standard Deviation: 0.1936
  Percentage of True values: 3.9%
```

```
# 'Item Purchased_Belt' Statistics:
Cluster 0:
  Mean: 0.0385
  Median: 0.0000
  Standard Deviation: 0.1924
  Percentage of True values: 3.8%
Cluster 1:
  Mean: 0.0441
  Median: 0.0000
  Standard Deviation: 0.2054
  Percentage of True values: 4.4%

# 'Item Purchased_Blouse' Statistics:
Cluster 0:
  Mean: 0.0472
  Median: 0.0000
  Standard Deviation: 0.2121
  Percentage of True values: 4.7%
Cluster 1:
  Mean: 0.0405
  Median: 0.0000
  Standard Deviation: 0.1972
  Percentage of True values: 4.1%

# 'Item Purchased_Boots' Statistics:
Cluster 0:
  Mean: 0.0395
  Median: 0.0000
  Standard Deviation: 0.1948
  Percentage of True values: 3.9%
Cluster 1:
  Mean: 0.0344
  Median: 0.0000
  Standard Deviation: 0.1822
  Percentage of True values: 3.4%

# 'Item Purchased_Coat' Statistics:
Cluster 0:
  Mean: 0.0421
  Median: 0.0000
  Standard Deviation: 0.2008
  Percentage of True values: 4.2%
Cluster 1:
  Mean: 0.0405
  Median: 0.0000
  Standard Deviation: 0.1972
  Percentage of True values: 4.1%

# 'Item Purchased_Dress' Statistics:
Cluster 0:
  Mean: 0.0472
  Median: 0.0000
  Standard Deviation: 0.2121
  Percentage of True values: 4.7%
Cluster 1:
  Mean: 0.0379
  Median: 0.0000
  Standard Deviation: 0.1911
  Percentage of True values: 3.8%
```

'Item Purchased_Gloves' Statistics:

Cluster 0:

Mean: 0.0354
Median: 0.0000
Standard Deviation: 0.1848
Percentage of True values: 3.5%

Cluster 1:

Mean: 0.0364
Median: 0.0000
Standard Deviation: 0.1874
Percentage of True values: 3.6%

'Item Purchased_Handbag' Statistics:

Cluster 0:

Mean: 0.0426
Median: 0.0000
Standard Deviation: 0.2019
Percentage of True values: 4.3%

Cluster 1:

Mean: 0.0359
Median: 0.0000
Standard Deviation: 0.1861
Percentage of True values: 3.6%

'Item Purchased_Hat' Statistics:

Cluster 0:

Mean: 0.0385
Median: 0.0000
Standard Deviation: 0.1924
Percentage of True values: 3.8%

Cluster 1:

Mean: 0.0405
Median: 0.0000
Standard Deviation: 0.1972
Percentage of True values: 4.1%

'Item Purchased_Hoodie' Statistics:

Cluster 0:

Mean: 0.0364
Median: 0.0000
Standard Deviation: 0.1874
Percentage of True values: 3.6%

Cluster 1:

Mean: 0.0410
Median: 0.0000
Standard Deviation: 0.1984
Percentage of True values: 4.1%

'Item Purchased_Jacket' Statistics:

Cluster 0:

Mean: 0.0405
Median: 0.0000
Standard Deviation: 0.1972
Percentage of True values: 4.1%

Cluster 1:

Mean: 0.0431
Median: 0.0000
Standard Deviation: 0.2031
Percentage of True values: 4.3%


```
# 'Item Purchased_Jeans' Statistics:
Cluster 0:
  Mean: 0.0267
  Median: 0.0000
  Standard Deviation: 0.1611
  Percentage of True values: 2.7%
Cluster 1:
  Mean: 0.0369
  Median: 0.0000
  Standard Deviation: 0.1886
  Percentage of True values: 3.7%

# 'Item Purchased_Jewelry' Statistics:
Cluster 0:
  Mean: 0.0528
  Median: 0.0000
  Standard Deviation: 0.2237
  Percentage of True values: 5.3%
Cluster 1:
  Mean: 0.0349
  Median: 0.0000
  Standard Deviation: 0.1835
  Percentage of True values: 3.5%

# 'Item Purchased_Pants' Statistics:
Cluster 0:
  Mean: 0.0436
  Median: 0.0000
  Standard Deviation: 0.2042
  Percentage of True values: 4.4%
Cluster 1:
  Mean: 0.0441
  Median: 0.0000
  Standard Deviation: 0.2054
  Percentage of True values: 4.4%

# 'Item Purchased_Sandals' Statistics:
Cluster 0:
  Mean: 0.0405
  Median: 0.0000
  Standard Deviation: 0.1972
  Percentage of True values: 4.1%
Cluster 1:
  Mean: 0.0415
  Median: 0.0000
  Standard Deviation: 0.1996
  Percentage of True values: 4.2%

# 'Item Purchased_Scarf' Statistics:
Cluster 0:
  Mean: 0.0436
  Median: 0.0000
  Standard Deviation: 0.2042
  Percentage of True values: 4.4%
Cluster 1:
  Mean: 0.0369
  Median: 0.0000
  Standard Deviation: 0.1886
  Percentage of True values: 3.7%
```

```
# 'Item Purchased_Shirt' Statistics:
Cluster 0:
  Mean: 0.0395
  Median: 0.0000
  Standard Deviation: 0.1948
  Percentage of True values: 3.9%
Cluster 1:
  Mean: 0.0472
  Median: 0.0000
  Standard Deviation: 0.2121
  Percentage of True values: 4.7%

# 'Item Purchased_Shoes' Statistics:
Cluster 0:
  Mean: 0.0405
  Median: 0.0000
  Standard Deviation: 0.1972
  Percentage of True values: 4.1%
Cluster 1:
  Mean: 0.0364
  Median: 0.0000
  Standard Deviation: 0.1874
  Percentage of True values: 3.6%

# 'Item Purchased_Shorts' Statistics:
Cluster 0:
  Mean: 0.0405
  Median: 0.0000
  Standard Deviation: 0.1972
  Percentage of True values: 4.1%
Cluster 1:
  Mean: 0.0400
  Median: 0.0000
  Standard Deviation: 0.1960
  Percentage of True values: 4.0%

# 'Item Purchased_Skirt' Statistics:
Cluster 0:
  Mean: 0.0379
  Median: 0.0000
  Standard Deviation: 0.1911
  Percentage of True values: 3.8%
Cluster 1:
  Mean: 0.0431
  Median: 0.0000
  Standard Deviation: 0.2031
  Percentage of True values: 4.3%

# 'Item Purchased_Sneakers' Statistics:
Cluster 0:
  Mean: 0.0349
  Median: 0.0000
  Standard Deviation: 0.1835
  Percentage of True values: 3.5%
Cluster 1:
  Mean: 0.0395
  Median: 0.0000
  Standard Deviation: 0.1948
  Percentage of True values: 3.9%
```

```
# 'Item Purchased_Socks' Statistics:
Cluster 0:
  Mean: 0.0395
  Median: 0.0000
  Standard Deviation: 0.1948
  Percentage of True values: 3.9%
Cluster 1:
  Mean: 0.0421
  Median: 0.0000
  Standard Deviation: 0.2008
  Percentage of True values: 4.2%

# 'Item Purchased_Sunglasses' Statistics:
Cluster 0:
  Mean: 0.0431
  Median: 0.0000
  Standard Deviation: 0.2031
  Percentage of True values: 4.3%
Cluster 1:
  Mean: 0.0395
  Median: 0.0000
  Standard Deviation: 0.1948
  Percentage of True values: 3.9%

# 'Item Purchased_Sweater' Statistics:
Cluster 0:
  Mean: 0.0400
  Median: 0.0000
  Standard Deviation: 0.1960
  Percentage of True values: 4.0%
Cluster 1:
  Mean: 0.0441
  Median: 0.0000
  Standard Deviation: 0.2054
  Percentage of True values: 4.4%

# 'Item Purchased_T-shirt' Statistics:
Cluster 0:
  Mean: 0.0349
  Median: 0.0000
  Standard Deviation: 0.1835
  Percentage of True values: 3.5%
Cluster 1:
  Mean: 0.0405
  Median: 0.0000
  Standard Deviation: 0.1972
  Percentage of True values: 4.1%

# 'Category_Accessories' Statistics:
Cluster 0:
  Mean: 0.3287
  Median: 0.0000
  Standard Deviation: 0.4699
  Percentage of True values: 32.9%
Cluster 1:
  Mean: 0.3072
  Median: 0.0000
  Standard Deviation: 0.4614
  Percentage of True values: 30.7%
```

```
# 'Category_Clothing' Statistics:
Cluster 0:
  Mean: 0.4333
  Median: 0.0000
  Standard Deviation: 0.4957
  Percentage of True values: 43.3%
Cluster 1:
  Mean: 0.4574
  Median: 0.0000
  Standard Deviation: 0.4983
  Percentage of True values: 45.7%

# 'Category_Footwear' Statistics:
Cluster 0:
  Mean: 0.1554
  Median: 0.0000
  Standard Deviation: 0.3624
  Percentage of True values: 15.5%
Cluster 1:
  Mean: 0.1518
  Median: 0.0000
  Standard Deviation: 0.3589
  Percentage of True values: 15.2%

# 'Category_Outerwear' Statistics:
Cluster 0:
  Mean: 0.0826
  Median: 0.0000
  Standard Deviation: 0.2753
  Percentage of True values: 8.3%
Cluster 1:
  Mean: 0.0836
  Median: 0.0000
  Standard Deviation: 0.2768
  Percentage of True values: 8.4%

# 'Location_Alabama' Statistics:
Cluster 0:
  Mean: 0.0251
  Median: 0.0000
  Standard Deviation: 0.1566
  Percentage of True values: 2.5%
Cluster 1:
  Mean: 0.0205
  Median: 0.0000
  Standard Deviation: 0.1418
  Percentage of True values: 2.1%

# 'Location_Alaska' Statistics:
Cluster 0:
  Mean: 0.0210
  Median: 0.0000
  Standard Deviation: 0.1435
  Percentage of True values: 2.1%
Cluster 1:
  Mean: 0.0159
  Median: 0.0000
  Standard Deviation: 0.1251
  Percentage of True values: 1.6%
```

```
# 'Location_Arizona' Statistics:
Cluster 0:
  Mean: 0.0190
  Median: 0.0000
  Standard Deviation: 0.1365
  Percentage of True values: 1.9%
Cluster 1:
  Mean: 0.0144
  Median: 0.0000
  Standard Deviation: 0.1190
  Percentage of True values: 1.4%

# 'Location_Arkansas' Statistics:
Cluster 0:
  Mean: 0.0241
  Median: 0.0000
  Standard Deviation: 0.1534
  Percentage of True values: 2.4%
Cluster 1:
  Mean: 0.0164
  Median: 0.0000
  Standard Deviation: 0.1271
  Percentage of True values: 1.6%

# 'Location_California' Statistics:
Cluster 0:
  Mean: 0.0236
  Median: 0.0000
  Standard Deviation: 0.1518
  Percentage of True values: 2.4%
Cluster 1:
  Mean: 0.0251
  Median: 0.0000
  Standard Deviation: 0.1566
  Percentage of True values: 2.5%

# 'Location_Colorado' Statistics:
Cluster 0:
  Mean: 0.0174
  Median: 0.0000
  Standard Deviation: 0.1309
  Percentage of True values: 1.7%
Cluster 1:
  Mean: 0.0210
  Median: 0.0000
  Standard Deviation: 0.1435
  Percentage of True values: 2.1%

# 'Location_Connecticut' Statistics:
Cluster 0:
  Mean: 0.0190
  Median: 0.0000
  Standard Deviation: 0.1365
  Percentage of True values: 1.9%
Cluster 1:
  Mean: 0.0210
  Median: 0.0000
  Standard Deviation: 0.1435
  Percentage of True values: 2.1%
```

```
# 'Location_Delaware' Statistics:
Cluster 0:
  Mean: 0.0210
  Median: 0.0000
  Standard Deviation: 0.1435
  Percentage of True values: 2.1%
Cluster 1:
  Mean: 0.0231
  Median: 0.0000
  Standard Deviation: 0.1502
  Percentage of True values: 2.3%

# 'Location_Florida' Statistics:
Cluster 0:
  Mean: 0.0164
  Median: 0.0000
  Standard Deviation: 0.1271
  Percentage of True values: 1.6%
Cluster 1:
  Mean: 0.0185
  Median: 0.0000
  Standard Deviation: 0.1346
  Percentage of True values: 1.8%

# 'Location_Georgia' Statistics:
Cluster 0:
  Mean: 0.0205
  Median: 0.0000
  Standard Deviation: 0.1418
  Percentage of True values: 2.1%
Cluster 1:
  Mean: 0.0200
  Median: 0.0000
  Standard Deviation: 0.1400
  Percentage of True values: 2.0%

# 'Location_Hawaii' Statistics:
Cluster 0:
  Mean: 0.0179
  Median: 0.0000
  Standard Deviation: 0.1328
  Percentage of True values: 1.8%
Cluster 1:
  Mean: 0.0154
  Median: 0.0000
  Standard Deviation: 0.1231
  Percentage of True values: 1.5%

# 'Location_Idaho' Statistics:
Cluster 0:
  Mean: 0.0210
  Median: 0.0000
  Standard Deviation: 0.1435
  Percentage of True values: 2.1%
Cluster 1:
  Mean: 0.0267
  Median: 0.0000
  Standard Deviation: 0.1611
  Percentage of True values: 2.7%
```

```
# 'Location_Illinois' Statistics:
Cluster 0:
  Mean: 0.0215
  Median: 0.0000
  Standard Deviation: 0.1452
  Percentage of True values: 2.2%
Cluster 1:
  Mean: 0.0256
  Median: 0.0000
  Standard Deviation: 0.1581
  Percentage of True values: 2.6%

# 'Location_Indiana' Statistics:
Cluster 0:
  Mean: 0.0226
  Median: 0.0000
  Standard Deviation: 0.1485
  Percentage of True values: 2.3%
Cluster 1:
  Mean: 0.0179
  Median: 0.0000
  Standard Deviation: 0.1328
  Percentage of True values: 1.8%

# 'Location_Iowa' Statistics:
Cluster 0:
  Mean: 0.0174
  Median: 0.0000
  Standard Deviation: 0.1309
  Percentage of True values: 1.7%
Cluster 1:
  Mean: 0.0179
  Median: 0.0000
  Standard Deviation: 0.1328
  Percentage of True values: 1.8%

# 'Location_Kansas' Statistics:
Cluster 0:
  Mean: 0.0154
  Median: 0.0000
  Standard Deviation: 0.1231
  Percentage of True values: 1.5%
Cluster 1:
  Mean: 0.0169
  Median: 0.0000
  Standard Deviation: 0.1290
  Percentage of True values: 1.7%

# 'Location_Kentucky' Statistics:
Cluster 0:
  Mean: 0.0195
  Median: 0.0000
  Standard Deviation: 0.1383
  Percentage of True values: 1.9%
Cluster 1:
  Mean: 0.0210
  Median: 0.0000
  Standard Deviation: 0.1435
  Percentage of True values: 2.1%
```

```
# 'Location_Louisiana' Statistics:
Cluster 0:
  Mean: 0.0215
  Median: 0.0000
  Standard Deviation: 0.1452
  Percentage of True values: 2.2%
Cluster 1:
  Mean: 0.0215
  Median: 0.0000
  Standard Deviation: 0.1452
  Percentage of True values: 2.2%

# 'Location_Maine' Statistics:
Cluster 0:
  Mean: 0.0169
  Median: 0.0000
  Standard Deviation: 0.1290
  Percentage of True values: 1.7%
Cluster 1:
  Mean: 0.0226
  Median: 0.0000
  Standard Deviation: 0.1485
  Percentage of True values: 2.3%

# 'Location_Maryland' Statistics:
Cluster 0:
  Mean: 0.0256
  Median: 0.0000
  Standard Deviation: 0.1581
  Percentage of True values: 2.6%
Cluster 1:
  Mean: 0.0185
  Median: 0.0000
  Standard Deviation: 0.1346
  Percentage of True values: 1.8%

# 'Location_Massachusetts' Statistics:
Cluster 0:
  Mean: 0.0169
  Median: 0.0000
  Standard Deviation: 0.1290
  Percentage of True values: 1.7%
Cluster 1:
  Mean: 0.0200
  Median: 0.0000
  Standard Deviation: 0.1400
  Percentage of True values: 2.0%

# 'Location_Michigan' Statistics:
Cluster 0:
  Mean: 0.0200
  Median: 0.0000
  Standard Deviation: 0.1400
  Percentage of True values: 2.0%
Cluster 1:
  Mean: 0.0174
  Median: 0.0000
  Standard Deviation: 0.1309
  Percentage of True values: 1.7%
```



```
# 'Location_Minnesota' Statistics:
Cluster 0:
  Mean: 0.0241
  Median: 0.0000
  Standard Deviation: 0.1534
  Percentage of True values: 2.4%
Cluster 1:
  Mean: 0.0210
  Median: 0.0000
  Standard Deviation: 0.1435
  Percentage of True values: 2.1%

# 'Location_Mississippi' Statistics:
Cluster 0:
  Mean: 0.0200
  Median: 0.0000
  Standard Deviation: 0.1400
  Percentage of True values: 2.0%
Cluster 1:
  Mean: 0.0210
  Median: 0.0000
  Standard Deviation: 0.1435
  Percentage of True values: 2.1%

# 'Location_Missouri' Statistics:
Cluster 0:
  Mean: 0.0236
  Median: 0.0000
  Standard Deviation: 0.1518
  Percentage of True values: 2.4%
Cluster 1:
  Mean: 0.0179
  Median: 0.0000
  Standard Deviation: 0.1328
  Percentage of True values: 1.8%

# 'Location_Montana' Statistics:
Cluster 0:
  Mean: 0.0251
  Median: 0.0000
  Standard Deviation: 0.1566
  Percentage of True values: 2.5%
Cluster 1:
  Mean: 0.0241
  Median: 0.0000
  Standard Deviation: 0.1534
  Percentage of True values: 2.4%

# 'Location_Nebraska' Statistics:
Cluster 0:
  Mean: 0.0221
  Median: 0.0000
  Standard Deviation: 0.1469
  Percentage of True values: 2.2%
Cluster 1:
  Mean: 0.0226
  Median: 0.0000
  Standard Deviation: 0.1485
  Percentage of True values: 2.3%
```

```
# 'Location_Nevada' Statistics:
Cluster 0:
  Mean: 0.0226
  Median: 0.0000
  Standard Deviation: 0.1485
  Percentage of True values: 2.3%
Cluster 1:
  Mean: 0.0221
  Median: 0.0000
  Standard Deviation: 0.1469
  Percentage of True values: 2.2%

# 'Location_New Hampshire' Statistics:
Cluster 0:
  Mean: 0.0195
  Median: 0.0000
  Standard Deviation: 0.1383
  Percentage of True values: 1.9%
Cluster 1:
  Mean: 0.0169
  Median: 0.0000
  Standard Deviation: 0.1290
  Percentage of True values: 1.7%

# 'Location_New Jersey' Statistics:
Cluster 0:
  Mean: 0.0185
  Median: 0.0000
  Standard Deviation: 0.1346
  Percentage of True values: 1.8%
Cluster 1:
  Mean: 0.0159
  Median: 0.0000
  Standard Deviation: 0.1251
  Percentage of True values: 1.6%

# 'Location_New Mexico' Statistics:
Cluster 0:
  Mean: 0.0221
  Median: 0.0000
  Standard Deviation: 0.1469
  Percentage of True values: 2.2%
Cluster 1:
  Mean: 0.0195
  Median: 0.0000
  Standard Deviation: 0.1383
  Percentage of True values: 1.9%

# 'Location_New York' Statistics:
Cluster 0:
  Mean: 0.0236
  Median: 0.0000
  Standard Deviation: 0.1518
  Percentage of True values: 2.4%
Cluster 1:
  Mean: 0.0210
  Median: 0.0000
  Standard Deviation: 0.1435
  Percentage of True values: 2.1%
```

'Location_North Carolina' Statistics:

Cluster 0:

Mean: 0.0210

Median: 0.0000

Standard Deviation: 0.1435

Percentage of True values: 2.1%

Cluster 1:

Mean: 0.0190

Median: 0.0000

Standard Deviation: 0.1365

Percentage of True values: 1.9%

'Location_North Dakota' Statistics:

Cluster 0:

Mean: 0.0190

Median: 0.0000

Standard Deviation: 0.1365

Percentage of True values: 1.9%

Cluster 1:

Mean: 0.0236

Median: 0.0000

Standard Deviation: 0.1518

Percentage of True values: 2.4%

'Location_Ohio' Statistics:

Cluster 0:

Mean: 0.0210

Median: 0.0000

Standard Deviation: 0.1435

Percentage of True values: 2.1%

Cluster 1:

Mean: 0.0185

Median: 0.0000

Standard Deviation: 0.1346

Percentage of True values: 1.8%

'Location_Oklahoma' Statistics:

Cluster 0:

Mean: 0.0149

Median: 0.0000

Standard Deviation: 0.1211

Percentage of True values: 1.5%

Cluster 1:

Mean: 0.0236

Median: 0.0000

Standard Deviation: 0.1518

Percentage of True values: 2.4%

'Location_Oregon' Statistics:

Cluster 0:

Mean: 0.0200

Median: 0.0000

Standard Deviation: 0.1400

Percentage of True values: 2.0%

Cluster 1:

Mean: 0.0179

Median: 0.0000

Standard Deviation: 0.1328

Percentage of True values: 1.8%

'Location_Pennsylvania' Statistics:

Cluster 0:

Mean: 0.0190
Median: 0.0000
Standard Deviation: 0.1365
Percentage of True values: 1.9%

Cluster 1:

Mean: 0.0190
Median: 0.0000
Standard Deviation: 0.1365
Percentage of True values: 1.9%

'Location_Rhode Island' Statistics:

Cluster 0:

Mean: 0.0169
Median: 0.0000
Standard Deviation: 0.1290
Percentage of True values: 1.7%

Cluster 1:

Mean: 0.0154
Median: 0.0000
Standard Deviation: 0.1231
Percentage of True values: 1.5%

'Location_South Carolina' Statistics:

Cluster 0:

Mean: 0.0226
Median: 0.0000
Standard Deviation: 0.1485
Percentage of True values: 2.3%

Cluster 1:

Mean: 0.0164
Median: 0.0000
Standard Deviation: 0.1271
Percentage of True values: 1.6%

'Location_South Dakota' Statistics:

Cluster 0:

Mean: 0.0169
Median: 0.0000
Standard Deviation: 0.1290
Percentage of True values: 1.7%

Cluster 1:

Mean: 0.0190
Median: 0.0000
Standard Deviation: 0.1365
Percentage of True values: 1.9%

'Location_Tennessee' Statistics:

Cluster 0:

Mean: 0.0200
Median: 0.0000
Standard Deviation: 0.1400
Percentage of True values: 2.0%

Cluster 1:

Mean: 0.0195
Median: 0.0000
Standard Deviation: 0.1383
Percentage of True values: 1.9%

```
# 'Location_Texas' Statistics:
Cluster 0:
  Mean: 0.0144
  Median: 0.0000
  Standard Deviation: 0.1190
  Percentage of True values: 1.4%
Cluster 1:
  Mean: 0.0251
  Median: 0.0000
  Standard Deviation: 0.1566
  Percentage of True values: 2.5%

# 'Location_Utah' Statistics:
Cluster 0:
  Mean: 0.0200
  Median: 0.0000
  Standard Deviation: 0.1400
  Percentage of True values: 2.0%
Cluster 1:
  Mean: 0.0164
  Median: 0.0000
  Standard Deviation: 0.1271
  Percentage of True values: 1.6%

# 'Location_Vermont' Statistics:
Cluster 0:
  Mean: 0.0200
  Median: 0.0000
  Standard Deviation: 0.1400
  Percentage of True values: 2.0%
Cluster 1:
  Mean: 0.0236
  Median: 0.0000
  Standard Deviation: 0.1518
  Percentage of True values: 2.4%

# 'Location_Virginia' Statistics:
Cluster 0:
  Mean: 0.0154
  Median: 0.0000
  Standard Deviation: 0.1231
  Percentage of True values: 1.5%
Cluster 1:
  Mean: 0.0241
  Median: 0.0000
  Standard Deviation: 0.1534
  Percentage of True values: 2.4%

# 'Location_Washington' Statistics:
Cluster 0:
  Mean: 0.0185
  Median: 0.0000
  Standard Deviation: 0.1346
  Percentage of True values: 1.8%
Cluster 1:
  Mean: 0.0190
  Median: 0.0000
  Standard Deviation: 0.1365
  Percentage of True values: 1.9%
```

'Location_West Virginia' Statistics:

Cluster 0:

Mean: 0.0169
Median: 0.0000
Standard Deviation: 0.1290
Percentage of True values: 1.7%

Cluster 1:

Mean: 0.0246
Median: 0.0000
Standard Deviation: 0.1550
Percentage of True values: 2.5%

'Location_Wisconsin' Statistics:

Cluster 0:

Mean: 0.0185
Median: 0.0000
Standard Deviation: 0.1346
Percentage of True values: 1.8%

Cluster 1:

Mean: 0.0200
Median: 0.0000
Standard Deviation: 0.1400
Percentage of True values: 2.0%

'Location_Wyoming' Statistics:

Cluster 0:

Mean: 0.0205
Median: 0.0000
Standard Deviation: 0.1418
Percentage of True values: 2.1%

Cluster 1:

Mean: 0.0159
Median: 0.0000
Standard Deviation: 0.1251
Percentage of True values: 1.6%

'Size_L' Statistics:

Cluster 0:

Mean: 0.2744
Median: 0.0000
Standard Deviation: 0.4463
Percentage of True values: 27.4%

Cluster 1:

Mean: 0.2656
Median: 0.0000
Standard Deviation: 0.4418
Percentage of True values: 26.6%

'Size_M' Statistics:

Cluster 0:

Mean: 0.4538
Median: 0.0000
Standard Deviation: 0.4980
Percentage of True values: 45.4%

Cluster 1:

Mean: 0.4462
Median: 0.0000
Standard Deviation: 0.4972
Percentage of True values: 44.6%

```
# 'Size_S' Statistics:
Cluster 0:
  Mean: 0.1636
  Median: 0.0000
  Standard Deviation: 0.3700
  Percentage of True values: 16.4%
Cluster 1:
  Mean: 0.1764
  Median: 0.0000
  Standard Deviation: 0.3813
  Percentage of True values: 17.6%

# 'Size_XL' Statistics:
Cluster 0:
  Mean: 0.1082
  Median: 0.0000
  Standard Deviation: 0.3107
  Percentage of True values: 10.8%
Cluster 1:
  Mean: 0.1118
  Median: 0.0000
  Standard Deviation: 0.3152
  Percentage of True values: 11.2%

# 'Color_Beige' Statistics:
Cluster 0:
  Mean: 0.0421
  Median: 0.0000
  Standard Deviation: 0.2008
  Percentage of True values: 4.2%
Cluster 1:
  Mean: 0.0333
  Median: 0.0000
  Standard Deviation: 0.1796
  Percentage of True values: 3.3%

# 'Color_Black' Statistics:
Cluster 0:
  Mean: 0.0451
  Median: 0.0000
  Standard Deviation: 0.2076
  Percentage of True values: 4.5%
Cluster 1:
  Mean: 0.0405
  Median: 0.0000
  Standard Deviation: 0.1972
  Percentage of True values: 4.1%

# 'Color_Blue' Statistics:
Cluster 0:
  Mean: 0.0354
  Median: 0.0000
  Standard Deviation: 0.1848
  Percentage of True values: 3.5%
Cluster 1:
  Mean: 0.0426
  Median: 0.0000
  Standard Deviation: 0.2019
  Percentage of True values: 4.3%
```

```
# 'Color_Brown' Statistics:
Cluster 0:
  Mean: 0.0349
  Median: 0.0000
  Standard Deviation: 0.1835
  Percentage of True values: 3.5%
Cluster 1:
  Mean: 0.0374
  Median: 0.0000
  Standard Deviation: 0.1899
  Percentage of True values: 3.7%

# 'Color_Charcoal' Statistics:
Cluster 0:
  Mean: 0.0400
  Median: 0.0000
  Standard Deviation: 0.1960
  Percentage of True values: 4.0%
Cluster 1:
  Mean: 0.0385
  Median: 0.0000
  Standard Deviation: 0.1924
  Percentage of True values: 3.8%

# 'Color_Cyan' Statistics:
Cluster 0:
  Mean: 0.0400
  Median: 0.0000
  Standard Deviation: 0.1960
  Percentage of True values: 4.0%
Cluster 1:
  Mean: 0.0451
  Median: 0.0000
  Standard Deviation: 0.2076
  Percentage of True values: 4.5%

# 'Color_Gold' Statistics:
Cluster 0:
  Mean: 0.0323
  Median: 0.0000
  Standard Deviation: 0.1769
  Percentage of True values: 3.2%
Cluster 1:
  Mean: 0.0385
  Median: 0.0000
  Standard Deviation: 0.1924
  Percentage of True values: 3.8%

# 'Color_Gray' Statistics:
Cluster 0:
  Mean: 0.0477
  Median: 0.0000
  Standard Deviation: 0.2132
  Percentage of True values: 4.8%
Cluster 1:
  Mean: 0.0338
  Median: 0.0000
  Standard Deviation: 0.1809
  Percentage of True values: 3.4%
```



```
# 'Color_Green' Statistics:
Cluster 0:
  Mean: 0.0379
  Median: 0.0000
  Standard Deviation: 0.1911
  Percentage of True values: 3.8%
Cluster 1:
  Mean: 0.0487
  Median: 0.0000
  Standard Deviation: 0.2153
  Percentage of True values: 4.9%

# 'Color_Indigo' Statistics:
Cluster 0:
  Mean: 0.0379
  Median: 0.0000
  Standard Deviation: 0.1911
  Percentage of True values: 3.8%
Cluster 1:
  Mean: 0.0374
  Median: 0.0000
  Standard Deviation: 0.1899
  Percentage of True values: 3.7%

# 'Color_Lavender' Statistics:
Cluster 0:
  Mean: 0.0405
  Median: 0.0000
  Standard Deviation: 0.1972
  Percentage of True values: 4.1%
Cluster 1:
  Mean: 0.0349
  Median: 0.0000
  Standard Deviation: 0.1835
  Percentage of True values: 3.5%

# 'Color_Magenta' Statistics:
Cluster 0:
  Mean: 0.0374
  Median: 0.0000
  Standard Deviation: 0.1899
  Percentage of True values: 3.7%
Cluster 1:
  Mean: 0.0405
  Median: 0.0000
  Standard Deviation: 0.1972
  Percentage of True values: 4.1%

# 'Color_Maroon' Statistics:
Cluster 0:
  Mean: 0.0390
  Median: 0.0000
  Standard Deviation: 0.1936
  Percentage of True values: 3.9%
Cluster 1:
  Mean: 0.0421
  Median: 0.0000
  Standard Deviation: 0.2008
  Percentage of True values: 4.2%
```

```
# 'Color_Olive' Statistics:
Cluster 0:
  Mean: 0.0415
  Median: 0.0000
  Standard Deviation: 0.1996
  Percentage of True values: 4.2%
Cluster 1:
  Mean: 0.0492
  Median: 0.0000
  Standard Deviation: 0.2164
  Percentage of True values: 4.9%

# 'Color_Orange' Statistics:
Cluster 0:
  Mean: 0.0410
  Median: 0.0000
  Standard Deviation: 0.1984
  Percentage of True values: 4.1%
Cluster 1:
  Mean: 0.0379
  Median: 0.0000
  Standard Deviation: 0.1911
  Percentage of True values: 3.8%

# 'Color_Peach' Statistics:
Cluster 0:
  Mean: 0.0395
  Median: 0.0000
  Standard Deviation: 0.1948
  Percentage of True values: 3.9%
Cluster 1:
  Mean: 0.0369
  Median: 0.0000
  Standard Deviation: 0.1886
  Percentage of True values: 3.7%

# 'Color_Pink' Statistics:
Cluster 0:
  Mean: 0.0374
  Median: 0.0000
  Standard Deviation: 0.1899
  Percentage of True values: 3.7%
Cluster 1:
  Mean: 0.0410
  Median: 0.0000
  Standard Deviation: 0.1984
  Percentage of True values: 4.1%

# 'Color_Purple' Statistics:
Cluster 0:
  Mean: 0.0369
  Median: 0.0000
  Standard Deviation: 0.1886
  Percentage of True values: 3.7%
Cluster 1:
  Mean: 0.0405
  Median: 0.0000
  Standard Deviation: 0.1972
  Percentage of True values: 4.1%
```

```
# 'Color_Red' Statistics:
Cluster 0:
  Mean: 0.0379
  Median: 0.0000
  Standard Deviation: 0.1911
  Percentage of True values: 3.8%
Cluster 1:
  Mean: 0.0379
  Median: 0.0000
  Standard Deviation: 0.1911
  Percentage of True values: 3.8%

# 'Color_Silver' Statistics:
Cluster 0:
  Mean: 0.0467
  Median: 0.0000
  Standard Deviation: 0.2110
  Percentage of True values: 4.7%
Cluster 1:
  Mean: 0.0421
  Median: 0.0000
  Standard Deviation: 0.2008
  Percentage of True values: 4.2%

# 'Color_Teal' Statistics:
Cluster 0:
  Mean: 0.0431
  Median: 0.0000
  Standard Deviation: 0.2031
  Percentage of True values: 4.3%
Cluster 1:
  Mean: 0.0451
  Median: 0.0000
  Standard Deviation: 0.2076
  Percentage of True values: 4.5%

# 'Color_Turquoise' Statistics:
Cluster 0:
  Mean: 0.0415
  Median: 0.0000
  Standard Deviation: 0.1996
  Percentage of True values: 4.2%
Cluster 1:
  Mean: 0.0328
  Median: 0.0000
  Standard Deviation: 0.1782
  Percentage of True values: 3.3%

# 'Color_Violet' Statistics:
Cluster 0:
  Mean: 0.0426
  Median: 0.0000
  Standard Deviation: 0.2019
  Percentage of True values: 4.3%
Cluster 1:
  Mean: 0.0426
  Median: 0.0000
  Standard Deviation: 0.2019
  Percentage of True values: 4.3%
```

```
# 'Color_White' Statistics:
Cluster 0:
  Mean: 0.0436
  Median: 0.0000
  Standard Deviation: 0.2042
  Percentage of True values: 4.4%
Cluster 1:
  Mean: 0.0292
  Median: 0.0000
  Standard Deviation: 0.1685
  Percentage of True values: 2.9%

# 'Color_Yellow' Statistics:
Cluster 0:
  Mean: 0.0379
  Median: 0.0000
  Standard Deviation: 0.1911
  Percentage of True values: 3.8%
Cluster 1:
  Mean: 0.0513
  Median: 0.0000
  Standard Deviation: 0.2206
  Percentage of True values: 5.1%

# 'Season_Fall' Statistics:
Cluster 0:
  Mean: 0.2482
  Median: 0.0000
  Standard Deviation: 0.4321
  Percentage of True values: 24.8%
Cluster 1:
  Mean: 0.2518
  Median: 0.0000
  Standard Deviation: 0.4342
  Percentage of True values: 25.2%

# 'Season_Spring' Statistics:
Cluster 0:
  Mean: 0.2615
  Median: 0.0000
  Standard Deviation: 0.4396
  Percentage of True values: 26.2%
Cluster 1:
  Mean: 0.2508
  Median: 0.0000
  Standard Deviation: 0.4336
  Percentage of True values: 25.1%

# 'Season_Summer' Statistics:
Cluster 0:
  Mean: 0.2446
  Median: 0.0000
  Standard Deviation: 0.4300
  Percentage of True values: 24.5%
Cluster 1:
  Mean: 0.2451
  Median: 0.0000
  Standard Deviation: 0.4303
  Percentage of True values: 24.5%
```

```
# 'Season_Winter' Statistics:
Cluster 0:
  Mean: 0.2456
  Median: 0.0000
  Standard Deviation: 0.4306
  Percentage of True values: 24.6%
Cluster 1:
  Mean: 0.2523
  Median: 0.0000
  Standard Deviation: 0.4344
  Percentage of True values: 25.2%

# 'Shipping Type_2-Day Shipping' Statistics:
Cluster 0:
  Mean: 0.1703
  Median: 0.0000
  Standard Deviation: 0.3760
  Percentage of True values: 17.0%
Cluster 1:
  Mean: 0.1513
  Median: 0.0000
  Standard Deviation: 0.3584
  Percentage of True values: 15.1%

# 'Shipping Type_Express' Statistics:
Cluster 0:
  Mean: 0.1713
  Median: 0.0000
  Standard Deviation: 0.3769
  Percentage of True values: 17.1%
Cluster 1:
  Mean: 0.1600
  Median: 0.0000
  Standard Deviation: 0.3667
  Percentage of True values: 16.0%

# 'Shipping Type_Free Shipping' Statistics:
Cluster 0:
  Mean: 0.1641
  Median: 0.0000
  Standard Deviation: 0.3705
  Percentage of True values: 16.4%
Cluster 1:
  Mean: 0.1821
  Median: 0.0000
  Standard Deviation: 0.3860
  Percentage of True values: 18.2%

# 'Shipping Type_Next Day Air' Statistics:
Cluster 0:
  Mean: 0.1569
  Median: 0.0000
  Standard Deviation: 0.3638
  Percentage of True values: 15.7%
Cluster 1:
  Mean: 0.1754
  Median: 0.0000
  Standard Deviation: 0.3804
  Percentage of True values: 17.5%
```

```
# 'Shipping Type_Standard' Statistics:
Cluster 0:
  Mean: 0.1805
  Median: 0.0000
  Standard Deviation: 0.3847
  Percentage of True values: 18.1%
Cluster 1:
  Mean: 0.1549
  Median: 0.0000
  Standard Deviation: 0.3619
  Percentage of True values: 15.5%

# 'Shipping Type_Store Pickup' Statistics:
Cluster 0:
  Mean: 0.1569
  Median: 0.0000
  Standard Deviation: 0.3638
  Percentage of True values: 15.7%
Cluster 1:
  Mean: 0.1764
  Median: 0.0000
  Standard Deviation: 0.3813
  Percentage of True values: 17.6%

# 'Payment Method_Bank Transfer' Statistics:
Cluster 0:
  Mean: 0.1462
  Median: 0.0000
  Standard Deviation: 0.3534
  Percentage of True values: 14.6%
Cluster 1:
  Mean: 0.1677
  Median: 0.0000
  Standard Deviation: 0.3737
  Percentage of True values: 16.8%

# 'Payment Method_Cash' Statistics:
Cluster 0:
  Mean: 0.1687
  Median: 0.0000
  Standard Deviation: 0.3746
  Percentage of True values: 16.9%
Cluster 1:
  Mean: 0.1749
  Median: 0.0000
  Standard Deviation: 0.3800
  Percentage of True values: 17.5%

# 'Payment Method_Credit Card' Statistics:
Cluster 0:
  Mean: 0.1800
  Median: 0.0000
  Standard Deviation: 0.3843
  Percentage of True values: 18.0%
Cluster 1:
  Mean: 0.1641
  Median: 0.0000
  Standard Deviation: 0.3705
  Percentage of True values: 16.4%
```

'Payment Method_Debit Card' Statistics:

Cluster 0:

Mean: 0.1656
Median: 0.0000
Standard Deviation: 0.3719
Percentage of True values: 16.6%

Cluster 1:

Mean: 0.1605
Median: 0.0000
Standard Deviation: 0.3672
Percentage of True values: 16.1%

'Payment Method_PayPal' Statistics:

Cluster 0:

Mean: 0.1723
Median: 0.0000
Standard Deviation: 0.3777
Percentage of True values: 17.2%

Cluster 1:

Mean: 0.1749
Median: 0.0000
Standard Deviation: 0.3800
Percentage of True values: 17.5%

'Payment Method_Venmo' Statistics:

Cluster 0:

Mean: 0.1672
Median: 0.0000
Standard Deviation: 0.3732
Percentage of True values: 16.7%

Cluster 1:

Mean: 0.1579
Median: 0.0000
Standard Deviation: 0.3648
Percentage of True values: 15.8%

'Frequency of Purchases_Annually' Statistics:

Cluster 0:

Mean: 0.1513
Median: 0.0000
Standard Deviation: 0.3584
Percentage of True values: 15.1%

Cluster 1:

Mean: 0.1421
Median: 0.0000
Standard Deviation: 0.3492
Percentage of True values: 14.2%

'Frequency of Purchases_Bi-Weekly' Statistics:

Cluster 0:

Mean: 0.1338
Median: 0.0000
Standard Deviation: 0.3406
Percentage of True values: 13.4%

Cluster 1:

Mean: 0.1467
Median: 0.0000
Standard Deviation: 0.3539
Percentage of True values: 14.7%

'Frequency of Purchases_Every 3 Months' Statistics:

Cluster 0:

Mean: 0.1436
Median: 0.0000
Standard Deviation: 0.3508
Percentage of True values: 14.4%

Cluster 1:

Mean: 0.1559
Median: 0.0000
Standard Deviation: 0.3629
Percentage of True values: 15.6%

'Frequency of Purchases_Fortnightly' Statistics:

Cluster 0:

Mean: 0.1344
Median: 0.0000
Standard Deviation: 0.3411
Percentage of True values: 13.4%

Cluster 1:

Mean: 0.1436
Median: 0.0000
Standard Deviation: 0.3508
Percentage of True values: 14.4%

'Frequency of Purchases_Monthly' Statistics:

Cluster 0:

Mean: 0.1405
Median: 0.0000
Standard Deviation: 0.3476
Percentage of True values: 14.1%

Cluster 1:

Mean: 0.1431
Median: 0.0000
Standard Deviation: 0.3502
Percentage of True values: 14.3%

'Frequency of Purchases_Quarterly' Statistics:

Cluster 0:

Mean: 0.1554
Median: 0.0000
Standard Deviation: 0.3624
Percentage of True values: 15.5%

Cluster 1:

Mean: 0.1333
Median: 0.0000
Standard Deviation: 0.3400
Percentage of True values: 13.3%

'Frequency of Purchases_Weekly' Statistics:

Cluster 0:

Mean: 0.1410
Median: 0.0000
Standard Deviation: 0.3481
Percentage of True values: 14.1%

Cluster 1:

Mean: 0.1354
Median: 0.0000
Standard Deviation: 0.3422
Percentage of True values: 13.5%

Based on the comprehensive statistical analysis, we draw the following key insights:

- **Age:** Cluster 0 exhibits an inclination towards an older demographic, while Cluster 1 tends to be younger.
- **Purchase Amount (USD):** Notably, Cluster 0 showcases a tendency for larger purchase amounts compared to Cluster 1. This aligns logically with the observation that Cluster 0 comprises an older population, often characterized by stable careers and higher income.
- **Past Purchases:** Cluster 0 demonstrates a higher frequency of past purchases in contrast to Cluster 1. This observation is consistent with the notion that the older demographic in Cluster 0 has a broader range of needs, including items for their children (toys, stationary, clothes, foods, etc.) and household essentials for the family.
- **Review Rating:** Interestingly, no significant distinctions emerge between Cluster 0 and Cluster 1 concerning the Review Rating column. Both clusters tend to assign similar star ratings, indicating that age may not correlate with increased critical evaluation or heightened expectations regarding the purchased products.
- **Binary Columns:** Analysis of binary columns reveals no substantial variations between Cluster 0 and Cluster 1.

In summary, the two identified customer segments are as follows:

1. **Cluster 0:** Comprising older, wealthier customers with a higher frequency of past purchases.
2. **Cluster 1:** Encompassing younger, more budget-conscious customers with fewer past purchases.

It's essential to note that while the Review Rating column and binary columns may not exhibit significant differences between clusters based on basic statistical measures, further investigation may be warranted. Additional analyses, such as advanced statistical modeling or detailed data visualization, could uncover more nuanced patterns or interactions that contribute to a deeper understanding of customer behavior and preferences.

These insights can provide valuable guidance for businesses to tailor their marketing strategies and product offerings to better meet the distinct needs and preferences of each customer segment.

1. **Cluster 0: Older, Affluent Customers with Diverse Needs:**

- **Tailoring Products and Services:**

- Given the older and likely more financially stable nature of Cluster 0, businesses can develop and highlight premium products or services that align with their affluent status.
- Consider offering a diverse range of products that cater to the varied needs of this demographic, including family-oriented items, luxury goods, and household essentials.
- **Targeted Marketing Campaigns:**
 - Craft marketing messages that resonate with the life experiences and values of an older demographic. This may include emphasizing product durability, family values, and the convenience of premium offerings.
- **Personalized Customer Experience:**
 - Implement personalized marketing strategies, leveraging data on past purchases and preferences to enhance the overall customer experience.
 - Loyalty programs or exclusive offers can be designed to reward the loyalty of this customer segment.
- **Channel Selection:**
 - Choose advertising and communication channels that are more likely to reach and engage an older audience. This might include traditional media, such as television or print, in addition to digital channels.

2. **Cluster 1: Younger, Budget-Conscious Customers with Potential for Future Growth:**

- **Affordable Product Lines:**
 - Develop and promote affordable product lines to cater to the budget-conscious nature of Cluster 1. This could involve creating entry-level or basic versions of products to appeal to this segment.
- **Educational Marketing:**
 - Craft marketing campaigns that highlight the value proposition of products, emphasizing quality, functionality, and cost-effectiveness.
 - Use educational content to inform younger customers about the benefits of products and how they align with their needs.
- **Digital and Social Media Engagement:**
 - Given the likely tech-savvy nature of younger consumers, focus marketing efforts on digital and social media platforms. Leverage influencers or online communities to amplify brand visibility.
- **Customer Engagement Strategies:**
 - Implement strategies to foster customer loyalty and long-term relationships. This could involve loyalty programs, interactive social media engagement, and responsive customer support.
- **Anticipation of Future Needs:**

- Recognize the potential for future growth within this segment. As younger customers advance in their careers and increase their spending capacity, our ecommerce platform should suggest more relevant products to them to meet their changing needs.

Classification

Predict if a customer's purchase will use a promo code

This indicates that the customer is a 'bargain hunter', a person who looks for a place or ecommerce platform to buy something at a price that is cheaper than usual market price. For the ecommerce to entice bargain hunter, they must have pricing and marketing strategies that cater to them. For example, offering first-come-first-serve promo code for certain items at a limited time such as Payday Campaign (the last week of each month).

```
In [28]: # Set the x attributes and y attribute
y = data['Promo Code Used'].values

X = data.drop(["Promo Code Used", 'Discount Applied'], axis=1)
```

Why 'Discount Applied' is dropped from X ?

Data leakage occurs when information from the target variable is inadvertently incorporated into the features during model training.

In this case, the presence of a perfect correlation coefficient of 1.0 between 'Promo Code Used' and 'Discount Applied' indicates a perfect linear relationship. This implies that if 'Discount Applied' contains information about the use of a promo code, keeping it as a feature in the model would lead to data leakage. Including 'Discount Applied' during training could artificially boost the model's performance, as it may unintentionally learn from the target variable.

To mitigate data leakage, we exclude the 'Discount Applied' column from the features (X). This ensures that the model learns solely from independent features, preventing any influence from information that would not be available during the prediction phase, and promotes a more accurate assessment of the model's generalization to new, unseen data.

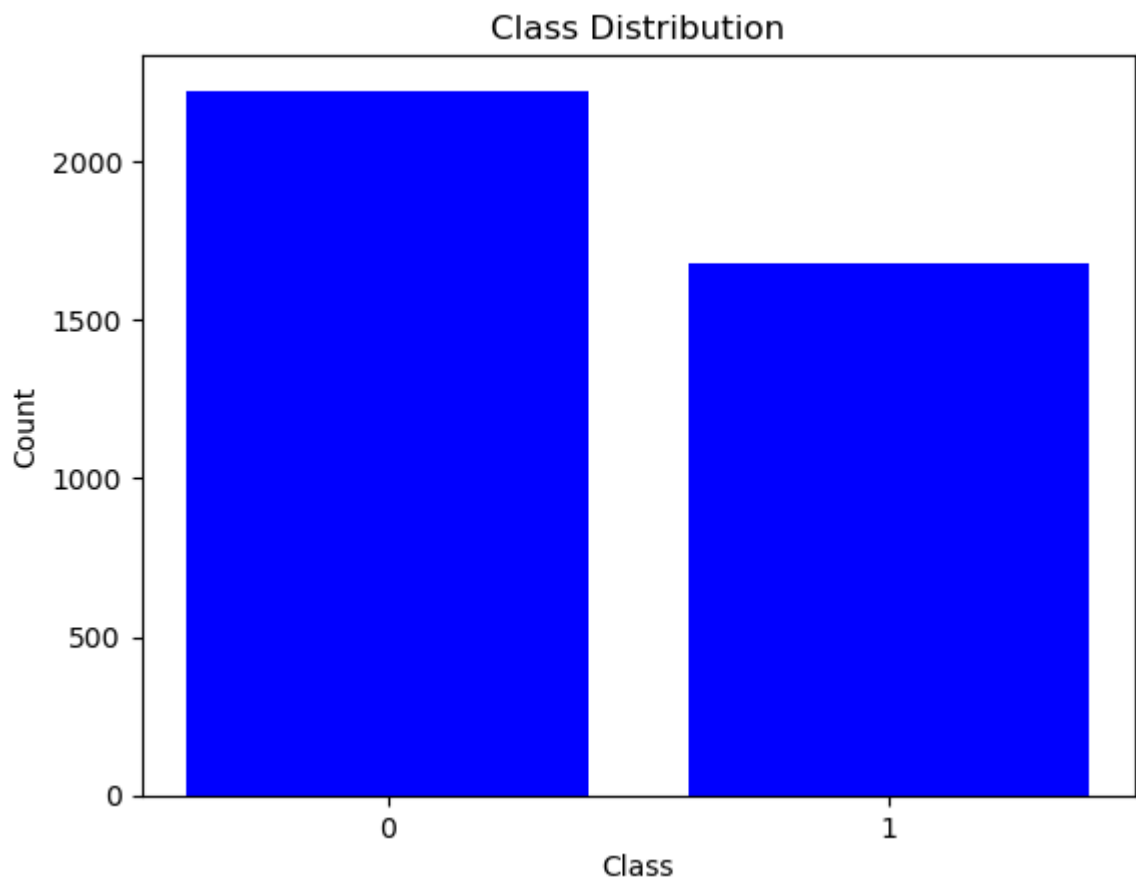
```
In [29]: class_counts = np.bincount(y)
count_0 = class_counts[0]
count_1 = class_counts[1]

print("Count of 0s:", count_0)
print("Count of 1s:", count_1)
```

```
Count of 0s: 2223
Count of 1s: 1677
```

```
In [30]: # Plotting the histogram with a single color
plt.bar([0, 1], class_counts, color='blue') # You can replace 'blue' with your
plt.xlabel('Class')
```

```
plt.ylabel('Count')
plt.title('Class Distribution')
plt.xticks([0, 1])
plt.show()
```



```
In [31]: import pandas as pd
from sklearn.utils import resample

# Separate majority and minority classes
majority_class = data[data['Promo Code Used'] == 0]
minority_class = data[data['Promo Code Used'] == 1]

# Downsample the majority class
majority_downsampled = resample(majority_class, replace=False, n_samples=count_1)

# Combine the minority class with the downsampled majority class
balanced_data = pd.concat([majority_downsampled, minority_class])

# Update X and y with balanced data
y_balanced = balanced_data['Promo Code Used'].values
X_balanced = balanced_data.drop(["Promo Code Used", 'Discount Applied'], axis=1)

# Display the counts after balancing
class_counts_balanced = np.bincount(y_balanced)
count_0_balanced = class_counts_balanced[0]
count_1_balanced = class_counts_balanced[1]

print("Count of 0s (balanced):", count_0_balanced)
print("Count of 1s (balanced):", count_1_balanced)
```

```
Count of 0s (balanced): 1677
Count of 1s (balanced): 1677
```

```
In [32]: # Split dataset
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state=42)

# Set seed for reproducibility
np.random.seed(42)
```

```
In [33]: # feature scaling

#Standardise feature variables

from sklearn.preprocessing import StandardScaler
X_features = X
X = StandardScaler().fit_transform(X)
```

```
In [34]: from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report

# Initialize the Decision Tree classifier
dt_classifier = DecisionTreeClassifier(random_state=42)

# Train the Decision Tree model on the training data
dt_classifier.fit(X_train, y_train)

# Make predictions on the testing data
y_pred = dt_classifier.predict(X_test)

# Evaluate the performance of the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")

# Display classification report
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

Accuracy: 0.81

Classification Report:

	precision	recall	f1-score	support
0	0.83	0.83	0.83	527
1	0.79	0.79	0.79	448
accuracy			0.81	975
macro avg	0.81	0.81	0.81	975
weighted avg	0.81	0.81	0.81	975

Training the Decision Tree Classification model

```
In [35]: from sklearn.tree import DecisionTreeClassifier, plot_tree
decision_tree_classifier = DecisionTreeClassifier()
```

```
In [36]: # Scaling

scaler = StandardScaler()
```

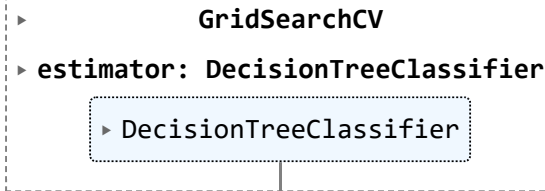
```
X_train_scale = scaler.fit_transform(X_train)
X_test_scale = scaler.fit_transform(X_test)
```

```
In [37]: param_grid = {
        'criterion': ['gini', 'entropy'],
        'max_depth': [None, 5, 10, 15],
        'min_samples_split': [2, 5, 10],
        'min_samples_leaf': [1, 2, 4]
    }
```

```
In [38]: grid_search = GridSearchCV(estimator = decision_tree_classifier, param_grid = pa
```

```
In [39]: grid_search.fit(X_train_scale, y_train)
```

```
Out[39]:
```



```
In [40]: # Get the best hyperparameters from the grid search
        best_params = grid_search.best_params_
```

```
In [41]: # Use the best model to make predictions
        best_model = grid_search.best_estimator_
        y_pred = best_model.predict(X_test)
```

```
In [42]: # Evaluate the performance of the best model
        from sklearn.metrics import confusion_matrix, accuracy_score
        accuracy = accuracy_score(y_test, y_pred)
```

```
In [43]: # Print the results
        print("Best Hyperparameters: ", best_params)
        print(f"Best Model Accuracy: {accuracy:.2f}")
```

```
Best Hyperparameters: {'criterion': 'gini', 'max_depth': 5, 'min_samples_leaf':
1, 'min_samples_split': 5}
Best Model Accuracy: 0.81
```

```
In [44]: DT = DecisionTreeClassifier(criterion = 'gini', max_depth = 5, min_samples_leaf
```

```
In [45]: from sklearn.metrics import classification_report
        DT.fit(X_train, y_train)
        DT_pred = DT.predict(X_test)
        acc_DT = accuracy_score(y_test, DT_pred)
        report = classification_report(y_test, DT_pred)
```

```
In [46]: print(f"Best Model Accuracy: {accuracy:.2f}")
        print("Classification Report:\n", report)
```

Best Model Accuracy: 0.81

Classification Report:

	precision	recall	f1-score	support
0	0.77	0.97	0.86	527
1	0.95	0.66	0.78	448
accuracy			0.83	975
macro avg	0.86	0.81	0.82	975
weighted avg	0.85	0.83	0.82	975

The model is tailored to predict whether customers will use a promo code, with two distinct classes: Class 0 (Do not use promo code) and Class 1 (Use promo code). Key performance metrics reveal an accuracy of 83%, indicating the correct classification of 83% of samples in the test dataset.

- **Class 0 Metrics:**

- *Precision (Positive Predictive Value): 77%*
 - 77% of predictions for customers who do not use promo codes were accurate.
- *Recall (Sensitivity or True Positive Rate): 97%*
 - The model correctly identified 97% of customers who do not use promo codes out of all actual cases.
- *F1-score: 86%*
 - A balanced F1-score of 86% for Class 0.

- **Class 1 Metrics:**

- *Precision (Positive Predictive Value): 95%*
 - 95% accuracy in predicting customers who use promo codes.
- *Recall (Sensitivity or True Positive Rate): 66%*
 - The model captured 66% of customers who use promo codes out of all actual cases.
- *F1-score: 78%*
 - An F1-score of 78% for Class 1.

Interpretation:

- For predicting customers who do not use promo codes (Class 0), the model maintains a high level of accuracy and recall, ensuring correct identification of the majority in this category.
- For predicting customers who use promo codes (Class 1), the model emphasizes high precision, aligning with our objective of offering promo codes more generously. This results in a willingness to potentially miss some customers who would use promo codes, as indicated by the lower recall.

Considerations:

- Our preference for maximizing the number of customers using promo codes suggests a need to further optimize the model to increase recall for Class 1.

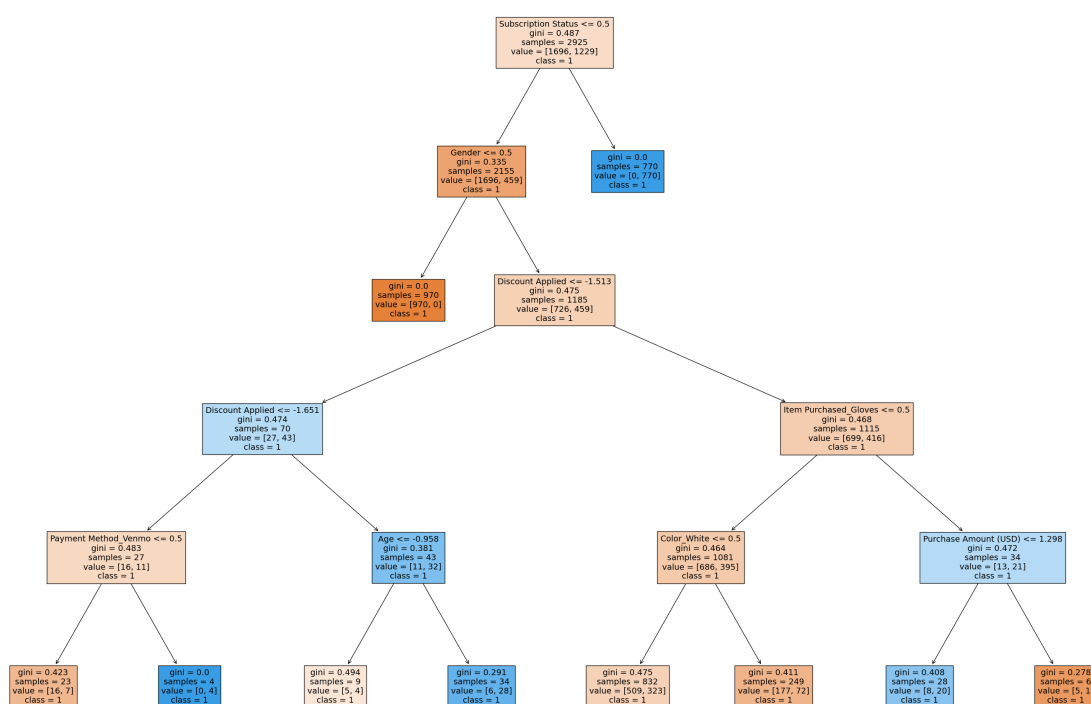
- The willingness to sacrifice precision for higher recall acknowledges the possibility of sending more promotions to customers who may not use them. However, this aligns with the strategy of casting a broader net to capture a larger audience.
- As the ecommerce platform owner, the implementation of a transaction limit for promo code redemption (e.g., first 10,000 transactions) mitigates the impact of potentially increased promotions, ensuring a controlled and manageable distribution and costs. Therefore, we will not lose too much profits in case too many customers use the promo codes.

Recommendation:

- Considering the goal of maximizing promo code usage, further research and model refinement should focus on strategies to increase recall for Class 1. This iterative process should balance the trade-off between precision and recall to align with our objective of extending promo code offers generously.

In conclusion, the model provides a strong foundation for predicting promo code usage, and with a deliberate focus on increasing recall for Class 1, the ecommerce should offer promotion codes more generously while controlling the costs through first-come-first-serve mechanism of limiting the promo code to be used only for the first 10,000 transactions.

```
In [47]: # Plot the Decision Tree
plt.figure(figsize = (36, 24))
plot_tree(DT, filled = True, feature_names = data.columns, class_names = data['P
plt.show()
```



Interpret the Decision Tree

First, it checks for Subscription Status ..

In []: