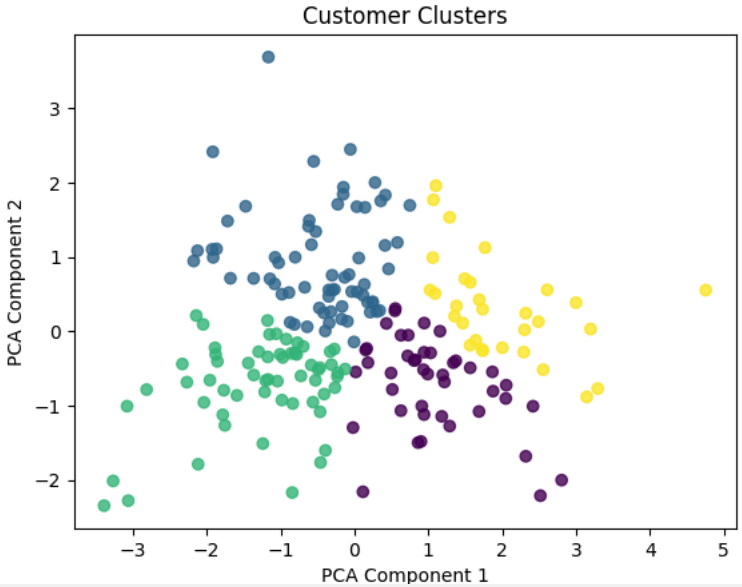
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns


# Load Data
customers = pd.read_csv("Customers.csv")
products = pd.read_csv("Products.csv")
transactions = pd.read_csv("Transactions.csv")


# Convert date columns to datetime
customers['SignupDate'] = pd.to_datetime(customers['SignupDate'])
transactions['TransactionDate'] = pd.to_datetime(transactions['TransactionDate'])



# Merge datasets for analysis
df = transactions.merge(customers, on="CustomerID").merge(products, on="ProductID")
df.head()
```

	TransactionID	CustomerID	ProductID	TransactionDate	Quantity	TotalValue	Price_x	CustomerName	Region	SignupDate	ProductNa
0	T00001	C0199	P067	2024-08-25 12:38:23	1	300.68	300.68	Andrea Jenkins	Europe	2022-12-03	ComfortLiv Blueto Spea
1	T00112	C0146	P067	2024-05-27 22:23:54	1	300.68	300.68	Brittany Harvey	Asia	2024-09-04	ComfortLiv Blueto Spea

```
from sklearn.cluster import KMeans
from sklearn.metrics import davies_bouldin_score
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler


# Prepare Features for Clustering
clustering_features = df.groupby('CustomerID').agg({
    'TotalValue': 'sum',
    'Quantity': 'sum',
    'Price_x': 'mean'
}).reset_index()


# Scaling Features
scaler = StandardScaler()
scaled_clustering_features = scaler.fit_transform(clustering_features[['TotalValue', 'Quantity', 'Price_x']])


# Apply K-Means Clustering
kmeans = KMeans(n_clusters=4, random_state=42)
clusters = kmeans.fit_predict(scaled_clustering_features)


# Assign Clusters
clustering_features['Cluster'] = clusters


# Calculate DB Index
db_index = davies_bouldin_score(scaled_clustering_features, clusters)
print("Davies-Bouldin Index:", db_index)
```

```
Davies-Bouldin Index: 1.12180191226693
```

```
# Visualize Clusters with PCA
pca = PCA(n_components=2)
reduced_features = pca.fit_transform(scaled_clustering_features)
plt.scatter(reduced_features[:, 0], reduced_features[:, 1], c=clusters, cmap='viridis', alpha=0.8)
plt.title("Customer Clusters")
plt.xlabel("PCA Component 1")
plt.ylabel("PCA Component 2")
plt.show()
```



```
# Save Clustering Results
clustering_features.to_csv("Customer_Clusters.csv", index=False)
```

Start coding or generate with AI.