

```

from os import P_PIDFD
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import davies_bouldin_score
import matplotlib.pyplot as plt
import seaborn as sns

def perform_clustering(transactions_path, customers_path, products_path, output_path, n_clusters=5):

    transactions = pd.read_csv( transactions_path)
    customers = pd.read_csv( customers_path )
    products = pd.read_csv(products_path )

    data = transactions.merge(customers, on='CustomerID').merge(products, on='ProductID')

    customer_data = data.groupby('CustomerID').agg({
        'TotalValue': 'sum',
        'Quantity': 'sum',
        'Price_y': 'mean'
    }).reset_index()

    scaler = StandardScaler()
    normalized_data = scaler.fit_transform(customer_data[['TotalValue', 'Quantity', 'Price_y']])

    kmeans = KMeans(n_clusters=n_clusters, random_state=42)
    clusters = kmeans.fit_predict(normalized_data)

    customer_data['Cluster'] = clusters

    db_index = davies_bouldin_score(normalized_data, clusters)
    print(f"Davies-Bouldin Index: {db_index}")

    customer_data.to_csv(output_path, index=False)
    print(f"Clustering results saved to {output_path}")

    plt.figure(figsize=(10, 6))
    sns.scatterplot(
        x=normalized_data[:, 0], y=normalized_data[:, 1],
        hue=clusters, palette='viridis', s=100
    )
    plt.title('Customer Clusters')
    plt.xlabel('Normalized TotalValue')
    plt.ylabel('Normalized Quantity')
    plt.legend(title='Cluster')
    plt.grid()
    plt.show()

    perform_clustering(
        transactions_path='Transactions.csv',
        customers_path='Customers.csv',
        products_path='Products.csv',
        output_path='Clustering_Results.csv',
        n_clusters=5
    )

```

