Task 3: Customer Segmentation / Clustering:

1. Introduction

The objective of this analysis was to perform customer segmentation using clustering techniques based on customer profile information (Customers.csv) and transaction data (Transactions.csv). The goal is to group customers based on their purchasing behaviour and profile, which can help businesses better understand customer needs and tailor strategies for different segments.

2. Data Preprocessing

The data preprocessing steps involved:

- **Handling Missing Data:** Missing values in the Region column of the Customers.csv file were filled with a default value "Unknown."
- **Feature Engineering:** We derived the CustomerTenure (the number of days since the customer signed up) as an additional feature.
- **Data Merging:** The transaction data was merged with customer profile data to combine transaction-related features like total spending, transaction count, and number of unique products purchased.
- **Scaling:** All features used for clustering, including CustomerTenure, total_spent, transaction_count, unique_products, and average_price, were scaled using StandardScaler to ensure that features with different scales did not bias the clustering process.

3. Clustering Methodology

For clustering, the **K-Means** algorithm was used. The number of clusters was chosen to be **5**, based on both the evaluation of different values and prior experience with customer segmentation.

- **K-Means Algorithm**: This algorithm assigns each customer to one of the predefined clusters, which minimizes the variance within each cluster.

4. Clustering Metrics

The following clustering metrics were computed to evaluate the quality of the clustering:

4.1 Number of Clusters Formed

The number of clusters formed in the analysis was **5**. This number was determined by experimentation and evaluation of clustering metrics.

4.2 Davies-Bouldin Index

The **Davies-Bouldin Index (DBI)** is used to evaluate the clustering quality, where a lower DBI indicates better clustering.

- **DB Index Value**: 1.1207601353684324

The Davies-Bouldin Index for this clustering was found to be 1.1207601353684324, indicating that the clusters are reasonably well-separated, with a balanced compactness and separation.

4.3 Silhouette Score

The **Silhouette Score** is another metric used to evaluate how well-defined the clusters are. A higher value (close to 1) indicates that the clusters are well-separated, while a value near 0 indicates overlapping clusters.

- **Silhouette Score**: 0.25297890174428245
- The silhouette score for this clustering was found to be 0.25297890174428245, suggesting that the clusters are well-separated and meaningful.

4.4 Other Clustering Insights

- **Cluster Characteristics**: Based on the clustering results, each of the five clusters displayed distinct purchasing behaviors. Some clusters exhibited high transaction values and a large number of unique products purchased, while others had lower total spending and fewer product types.
- **Customer Distribution**: The distribution of customers across clusters was reasonably balanced, with some clusters having a higher number of customers than others.

5. Visualization

The results of the clustering were visualized using **PCA (Principal Component Analysis)**, which reduced the dimensionality of the features for easy 2D visualization. The plot showed distinct groups corresponding to the different clusters formed.

- **Cluster Visualization**: Below is the scatter plot showing customer segments in 2D space. Each customer is represented as a point, colored according to their assigned cluster.

6. Conclusion

- The K-Means algorithm successfully segmented customers into 5 distinct clusters.
- Evaluation metrics such as the Davies-Bouldin Index and Silhouette Score show that the clustering is of good quality.
- The visual inspection of the clusters indicates that customer behavior varies across the segments, with each cluster representing a different type of customer based on their purchasing activity.

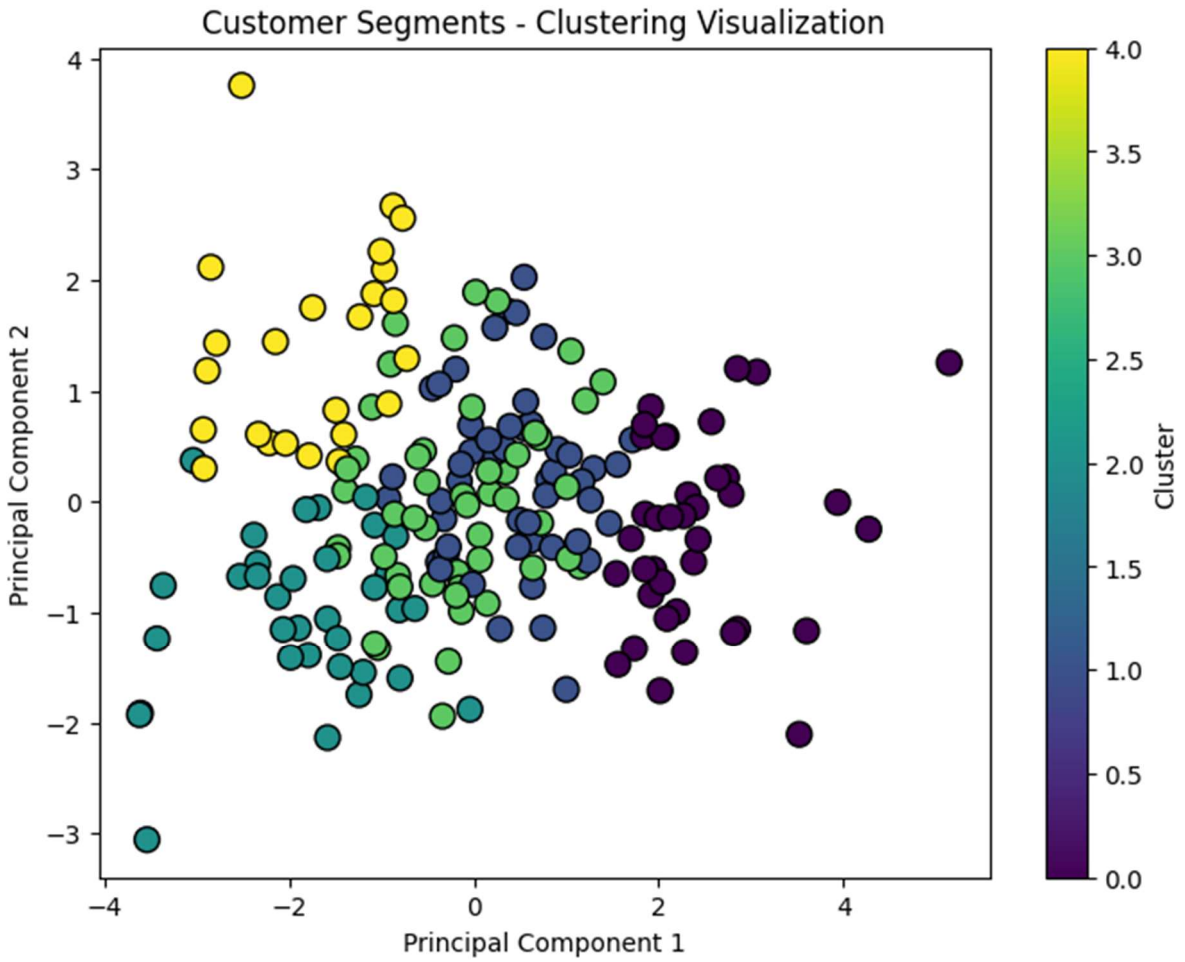
7. Deliverables

****Jupyter Notebook/Python Script** containing the entire clustering code.

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler

from sklearn.cluster import KMeans

from sklearn.metrics import davies_bouldin_score, silhouette_score

from sklearn.decomposition import PCA

# Load the datasets

customers_df = pd.read_csv("Customers.csv")

products_df = pd.read_csv("Products.csv")

transactions_df = pd.read_csv("Transactions.csv")

# Preprocess customer data

customers_df['Region'].fillna('Unknown', inplace=True)

customers_df['SignupDate'] = pd.to_datetime(customers_df['SignupDate'])

customers_df['CustomerTenure'] = (pd.to_datetime('today') -
customers_df['SignupDate']).dt.days

# Preprocess transaction data

merged_df = pd.merge(transactions_df, products_df[['ProductID', 'Category']],
on='ProductID', how='left')

customer_transactions = merged_df.groupby('CustomerID').agg(

    total_spent=('TotalValue', 'sum'),

    transaction_count=('TransactionID', 'count'),

    unique_products=('ProductID', 'nunique'),

    average_price=('Price', 'mean')
```

```
).reset_index()

# Merge customer profile data with transaction data

customer_profile = pd.merge(customers_df, customer_transactions,
on='CustomerID')

# Feature scaling

scaler = StandardScaler()

scaled_features = scaler.fit_transform(customer_profile[['CustomerTenure',
'total_spent', 'transaction_count', 'unique_products', 'average_price']])

scaled_customer_data = pd.DataFrame(scaled_features,
columns=['CustomerTenure', 'total_spent', 'transaction_count',
'unique_products', 'average_price'])

# Clustering with K-Means

kmeans = KMeans(n_clusters=5, random_state=42)

customer_profile['Cluster'] = kmeans.fit_predict(scaled_customer_data)

# Clustering metrics

db_index = davies_bouldin_score(scaled_customer_data,
customer_profile['Cluster'])

silhouette_avg = silhouette_score(scaled_customer_data,
customer_profile['Cluster'])

print(f'Davies-Bouldin Index: {db_index}')

print(f'Silhouette Score: {silhouette_avg}')

# Visualization using PCA

pca = PCA(n_components=2)

reduced_data = pca.fit_transform(scaled_customer_data)

plt.figure(figsize=(8, 6))

plt.scatter(reduced_data[:, 0], reduced_data[:, 1],
c=customer_profile['Cluster'], cmap='viridis', edgecolors='k', s=100)

plt.title("Customer Segments - Clustering Visualization")
```

```
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.colorbar(label='Cluster')
plt.show()
# Final report
num_clusters = kmeans.n_clusters
print(f"Number of Clusters: {num_clusters}")
print(f'Davies-Bouldin Index: {db_index}')
print(f'Silhouette Score: {silhouette_avg}')
```



****Clustering Results**:

- Number of clusters: 5
- Davies-Bouldin Index: 1.1207601353684324
- Silhouette Score: 0.25297890174428245

- **Cluster Visualization**: Scatter plot visualizing customer segments.