REPORT : CLUSTERING

Here is the complete code for customer segmentation using clustering:

```
# Import necessary librariesfrom sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import davies_bouldin_score
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Aggregate transaction data by CustomerID
customer_transactions = merged_df.groupby('CustomerID').agg({
    'TotalValue': 'sum',  # Total transaction value per customer'Quantity': 'sum'     # Total quantity
purchased per customer
}).reset_index()

# Standardize the data (scaling TotalValue and Quantity)
scaler = StandardScaler()
scaled_data = scaler.fit_transform(customer_transactions[['TotalValue', 'Quantity']])

# Apply KMeans clustering with 4 clusters
kmeans = KMeans(n_clusters=4, random_state=42)
customer_transactions['Cluster'] = kmeans.fit_predict(scaled_data)

# Calculate Davies-Bouldin Index (DB Index) for clustering evaluation
db_index = davies_bouldin_score(scaled_data, customer_transactions['Cluster'])

# Visualize the clusters using a scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x=customer_transactions['TotalValue'], y=customer_transactions['Quantity'],
hue=customer_transactions['Cluster'], palette='Set2')
plt.title(f'Customer Segments (DB Index: {db_index:.2f})')
plt.xlabel('Total Value')
plt.ylabel('Quantity')
plt.legend(title='Cluster')
plt.show()

# Save the clustering results (CustomerID and their corresponding Cluster)
customer_transactions[['CustomerID', 'Cluster']].to_csv('/content/Clustering.csv', index=False)
```

Key Points:

1. *Data Aggregation: The transaction data is aggregated by CustomerID to calculate the total value and total quantity for each customer.*
2. *Scaling: The data is scaled using StandardScaler to ensure that both TotalValue and Quantity are on the same scale, which is important for clustering algorithms.*
3. *Clustering: The KMeans clustering algorithm is applied with 4 clusters (you can adjust the number of clusters between 2 and 10).*
4. *Evaluation: The Davies-Bouldin Index (DB Index) is calculated to evaluate the clustering quality. A lower DB Index indicates better clustering.*
5. *Visualization: A scatter plot is created to visualize the clusters, with the x-axis representing TotalValue and the y-axis representing Quantity. Each point is colored according to its cluster.*
6. *Output: The clustering results (CustomerID and their respective cluster) are saved to a CSV file (Clustering.csv).*

Report:

- *Number of Clusters: 4 (can be adjusted as per the requirement).*
- *DB Index: The DB Index value will be printed in the plot title, indicating the quality of the clustering.*
- *Other Metrics: You can explore other metrics like silhouette score or inertia for further evaluation of the clustering.*

This script will perform the customer segmentation, evaluate the clustering, and visualize the results.