

# Customer Segmentation Using K-Means Clustering

## Introduction

This report presents a customer segmentation analysis using K-Means clustering. The goal is to segment customers based on their transaction data, which can help in designing targeted marketing strategies and improving customer relationship management.

## Methodology

### Data Aggregation

The first step involves aggregating transaction data to create a summary for each customer:

- **TotalValue:** Sum of transaction values for each customer.
- **Quantity:** Sum of the quantity of products purchased by each customer.

```
customer_summary = transactions.groupby('CustomerID').agg({'TotalValue': 'sum', 'Quantity': 'sum'}).reset_index()
```

### Adding Customer Profile Data

Customer profile data is merged with the transaction summary to enrich the dataset:

```
customer_summary = customer_summary.merge(customers, on='CustomerID')
```

### Standardizing Data

To ensure consistent scaling across features, the data is standardized using StandardScaler:

```
scaler = StandardScaler()
```

```
features = scaler.fit_transform(customer_summary[['TotalValue', 'Quantity']])
```

### K-Means Clustering

K-Means clustering is applied to segment customers into distinct clusters:

- **Model Initialization:** The model is initialized with 4 clusters and a random state for reproducibility.
- **Fitting Model:** The model is fitted to the standardized features.
- **Cluster Assignment:** Each customer is assigned to a cluster.

```
kmeans = KMeans(n_clusters=4, random_state=42)
```

```
customer_summary['Cluster'] = kmeans.fit_predict(features)
```

### Evaluating Clustering Performance

The Davies-Bouldin Index (DB Index) is calculated to evaluate the clustering performance. A lower DB Index indicates better clustering quality.

```
db_index = davies_bouldin_score(features, customer_summary['Cluster'])
```

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

## Visualization

A scatter plot is used to visualize the customer clusters:

```
plt.figure(figsize=(10, 6))  
  
sns.scatterplot(x=features[:, 0], y=features[:, 1], hue=customer_summary['Cluster'], palette='viridis')  
  
plt.title('Customer Clusters')  
  
plt.show()
```

## Saving Results

The clustered customer data is saved to a CSV file for further analysis:

```
customer_summary.to_csv('Customer_Segmentation.csv', index=False)
```

## Conclusion

The K-Means clustering algorithm successfully segmented customers based on their transaction data. By understanding the characteristics of each cluster, businesses can tailor their marketing efforts, improve customer retention, and drive growth.