

# Customer Segmentation Analysis Report

This report presents the results of a customer segmentation analysis performed on the customer database. The analysis uses both customer profile information and transaction history to identify distinct customer segments using K-means clustering.

## **Methodology**

### **Feature Engineering**

The following features were used for clustering:

1. Transaction-based features:
  - Transaction count
  - Total spends
  - Average transaction value
  - Total quantity purchased
  - Average quantity per transaction
  - Purchase frequency (transactions per day since signup)
2. Profile-based features:
  - Region (one-hot encoded)

### **Clustering Approach**

- Algorithm: K-means clustering
- Number of clusters tested: 2 to 10
- Features were standardized using StandardScaler
- Best model selected based on optimal number of clusters and DB Index

## **Clustering Results**

### **Optimal Clustering**

- Number of clusters: 5
- Davies-Bouldin Index: 1.1861
- Silhouette Score: 0.2967
- Calinski-Harabasz Score: 48.3040

### **Visualization Analysis**

1. Elbow Curve Analysis
  - Shows the relationship between the number of clusters and within-cluster sum of squares
  - Helps identify the optimal number of clusters where adding more clusters provides diminishing returns
2. Davies-Bouldin Index
  - Measures the average similarity between each cluster and its most similar cluster
  - Lower values indicate better clustering
  - Used as the primary metric for selecting the optimal number of clusters
3. Silhouette Score
  - Measures how similar objects are to their own cluster compared to other clusters
  - Ranges from -1 to 1, where higher values indicate better-defined clusters
4. PCA Visualization
  - 2D representation of the clusters using principal component analysis
  - Demonstrates the separation between different customer segments

### **Technical Implementation**

The clustering analysis was implemented using:

- Python with scikit-learn for clustering
- Pandas for data manipulation
- Matplotlib and Seaborn for visualization
- Custom implementation of Davies-Bouldin Index