

## Clustering Report

This analysis applies two clustering techniques:

- **KMeans Clustering:** A partition-based clustering approach.
- **Hierarchical Clustering:** An agglomerative clustering technique visualized with a dendrogram.

Both techniques are evaluated using the Davies-Bouldin Index (DBI) to assess the clustering quality.

### K-MEANS CLUSTERING

1. **Parameter Selection:**
  - a. Number of clusters (k) is evaluated for values between 2 and 9.
  - b. DBI is calculated for each k value to determine the optimal number of clusters.
2. **Optimal Number of Clusters:**
  - a. The optimal k is chosen based on the minimum DBI value in this range.
3. **Final Clustering:**
  - a. KMeans is executed using the optimal k, and the final DBI is calculated to measure the compactness and separation of the clusters.

### Results:

- **Optimal k:** Determined dynamically (exact value is printed in the notebook during execution).
- **DBI for KMeans:**
  - The minimum DBI observed during parameter tuning.
  - Final DBI for the optimal number of clusters
  - For Example, K=9 is taken in the code

### HIERARCHICAL CLUSTERING

1. **Dendrogram:**
  - a. A dendrogram is plotted using the Ward linkage method to visualize the clustering structure.
  - b. This helps identify a reasonable number of clusters visually.
2. **Cluster Formation:**
  - a. Agglomerative clustering is performed with a selected number of clusters (Clusters=8)

### 3. DBI Calculation:

- a. The DBI is computed for the resulting hierarchical clusters to evaluate their quality.

### Results:

- **Clusters Formed:** A fixed number of clusters (Clusters=8) as chosen for hierarchical clustering.

### DB\_INDEX VALUES

K-Means: 0.5286694

Hierarchical: 0.65212084

```
from sklearn.metrics import davies_bouldin_score
db_score = davies_bouldin_score(clustering_data_scaled, cluster_labels)
print(f"Overall Davies-Bouldin Index: {db_score}")
unique_clusters = np.unique(cluster_labels)
for cluster in unique_clusters:
    cluster_data = clustering_data_scaled[cluster_labels == cluster]
```

4]

```
Overall Davies-Bouldin Index: 0.5286943349679493
```

**Overall Davies-Bouldin Index: 0.5286943349679493**

```
from sklearn.metrics import davies_bouldin_score  
  
db_score = davies_bouldin_score(clustering_data_scaled, y_hc)  
print(f"Davies-Bouldin Index: {db_score}")
```

Davies-Bouldin Index: 0.6521208498821581

**Davies-Bouldin Index: 0.6521208498821581**