Customer Segmentation Clustering Report

Objective: The objective of this analysis is to segment customers based on their transaction and profile information using clustering techniques. The aim is to identify different customer groups for personalized marketing and product recommendations.

Clustering Algorithms Evaluated

Three clustering algorithms were applied to the customer segmentation task: **KMeans**, **Agglomerative Clustering**, and **DBSCAN**.

1. KMeans Clustering:

- **Number of clusters formed**: 5 (optimal number chosen using the Elbow Method)
- **DB Index**: 0.979
- **Silhouette Score**: 0.316
- **Interpretation**:
    - KMeans performs well with well-separated and compact clusters. The relatively low DB Index suggests that the clusters are quite compact, but the moderate Silhouette Score indicates that the separation between the clusters could be improved.
    - Overall, KMeans is a good choice for structured, well-defined clusters.

2. Agglomerative Clustering:

- **Number of clusters formed**: 5
- **DB Index**: 1.022
- **Silhouette Score**: 0.299
- **Interpretation**:
    - Agglomerative Clustering also produces reasonable results with 5 clusters. However, its DB Index is slightly higher than KMeans, indicating that the clusters are less compact and have more variation in their shapes.
    - The Silhouette Score is similar to KMeans, which again indicates moderate separation between the clusters. The model's performance is similar to KMeans but with slightly lower compactness.

3. DBSCAN (Density-Based Spatial Clustering of Applications with Noise):

- **DB Index**: *Not applicable* (DBSCAN doesn't require a predefined number of clusters)
- **Silhouette Score**: *Not calculated* (the algorithm's cluster structure depends on the parameters)
- **Number of clusters**: 1 (DBSCAN labeled all points as noise)
- **Interpretation**:
    - DBSCAN failed to form meaningful clusters in this case. It identified all data points as noise, which means it couldn't find any significant clusters with the default settings. This could be due to inappropriate parameter settings such as eps (maximum distance for two points to be neighbors) and min_samples (minimum number of points required to form a dense region).
    - **Conclusion**: While DBSCAN is effective in identifying clusters with arbitrary shapes, it is not suitable for this dataset with the current parameter settings. It

didn't create any significant clusters, and most points were labeled as noise. Tuning DBSCAN's parameters could improve the results, but in this case, it was not the optimal choice.

Why DBSCAN Was Not Chosen

- **Parameter Sensitivity**: DBSCAN requires careful tuning of parameters (eps and min_samples). In this case, it failed to identify clusters due to the inappropriate default settings.
- **Limited Cluster Detection**: DBSCAN detected only one cluster and labeled all other points as noise. This indicated that DBSCAN was not effective in identifying meaningful clusters for this dataset.
- **Conclusion**: Despite its capability to handle arbitrary-shaped clusters, DBSCAN was less effective in this case. The failure to identify clusters led to its exclusion from the final clustering solution.

Conclusion and Final Choice of Clustering Model

- **Best Performing Model**: Based on the **DB Index** and **Silhouette Score**, **KMeans** emerged as the best-performing model. It provided compact and well-separated clusters with moderate quality in terms of the cluster separation (Silhouette Score = 0.316).
- **Agglomerative Clustering** performed similarly to KMeans but with slightly less compact clusters, as evidenced by its slightly higher DB Index (1.022).
- **DBSCAN**, while suitable for finding clusters of arbitrary shapes, did not produce meaningful clusters for this dataset. Thus, **KMeans** was chosen as the optimal clustering algorithm for this task.

Clustering Metrics Summary

Clustering Algorithm	Number of Clusters	DB Index	Silhouette Score
KMeans	3	0.979	0.316
Agglomerative Clustering	3	1.022	0.299
DBSCAN	1 (All Noise)	Not Applicable	Not Calculated

This report provides a thorough comparison of the clustering algorithms applied to the customer segmentation task, highlighting the performance metrics and reasoning behind the final choice of KMeans as the most suitable algorithm for this dataset.