**1. Silhouette Scores Analysis (KMeans)**

* With two clusters in KMeans, I get the highest Silhouette Score of 0.1206, suggesting that splitting the data into two clusters is the best option among the numbers I tested. Despite being the best score, 0.1206 is still low, indicating weak clustering strength.
* As I add more clusters (from three to five), the Silhouette Score decreases further, showing that additional clusters don’t improve the separation, and may even be overlapping or forced. The pattern of decreasing scores with more clusters suggests that my data might only have two distinguishable groups, if any.

**2. Gaussian Mixture Model (GMM) Results**

* The GMM Silhouette Score is 0.1006, which is close to KMeans with three clusters or more. This score is lower than KMeans with two clusters, indicating that GMM doesn't capture the data structure as well. This may imply that the data lacks a Gaussian shape and doesn’t align well with GMM's assumptions of Gaussian or ellipsoidal clusters.

**3. DBSCAN Results**

* The negative Silhouette Score of -0.378 in DBSCAN tells me that it’s not a good fit for this data. A negative score suggests significant noise classification or poorly matched clusters, implying that my data doesn’t have the dense, well-separated structure that DBSCAN typically performs well with.

**Summary and Recommendations**

**Based on my analysis:**

* Two Clusters in KMeans appear to be the best approach, but with a low Silhouette Score, even this isn’t particularly strong.
* Low Silhouette Scores Overall suggest that clustering may not be suitable for this dataset, or further dimensionality reduction or transformation (like t-SNE or UMAP) could help reveal clusters.
* Alternative Evaluation Metrics: Exploring metrics like the Davies-Bouldin Index could help me better interpret the data’s cluster structure.
* Further Exploration: Examining other data characteristics, like density and distribution, could confirm if clustering or other techniques, like classification or dimensionality reduction, are more suitable**.**

**Comparison Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Clustering Method** | **Number of Clusters** | **Silhouette Score** | **Observations** |
| **KMeans** | 2 | 0.1206 | Highest score; weak separation |
|  | 3 | 0.1044 | Decreased score; possibly forced clusters |
|  | 4 | 0.0982 | Further decrease; overlapping clusters likely |
|  | 5 | 0.0930 | Lowest score; no improvement |
| **GMM** | N/A | 0.1006 | Performs similarly to KMeans (3 clusters), but not as well as KMeans (2 clusters) |
| **DBSCAN** | N/A | -0.378 | Negative score; high noise classification |

**Second approach we adopted**

**Data Preprocessing**

* You drop irrelevant columns (ID, Satisfaction, Loyalty) that won't be used in clustering.
* Categorical columns are converted to numerical using one-hot encoding with pd.get\_dummies().

**Feature Scaling**

Features are scaled using StandardScaler to standardize the dataset, which is crucial for clustering algorithms that rely on distance calculations.

**Finding Optimal Clusters (K-Means)**

We implemented the Elbow Method to determine the optimal number of clusters. We calculated the sum of squared distances (SSE) for cluster numbers ranging from 1 to 10, plotting the results to identify the "elbow" point that indicates the ideal cluster count.

**Applying K-Means Clustering**

We then fitted the K-Means algorithm using the identified optimal number of clusters, assumed to be 3, and added the resulting cluster labels back to the original data.

**Visualizing Clusters**

To visualize the clusters, we created a scatter plot based on the first two principal components or selected features, using color to represent different clusters.

**Analyzing Cluster Characteristics**

We analyzed the characteristics of each cluster by calculating the mean of each feature for each cluster using groupby(), providing insights into the distinct attributes of the clusters.

**Gaussian Mixture Model Clustering**

We also tested various component numbers for a Gaussian Mixture Model (GMM), computing BIC and silhouette scores for cluster counts ranging from 2 to 9. These scores were plotted to help identify the best number of clusters.

**Applying GMM Clustering**

Next, we fitted the GMM with the chosen optimal number of clusters and added the labels to the original DataFrame. We similarly visualized these clusters using a scatter plot.