Customer Segmentation Report

Step 1: Data Loading and Exploration

- The dataset was loaded into the jupiter using `pandas`.
- It was then explored the dataset using `head()`

Step 2: Data Processing

- Merged all The data related to customers
- Changed date and time to relevant formats
- Calculated the aggregated data like sum of transactions amount and no of transactions and merged it.

Step 3: Feature Scaling

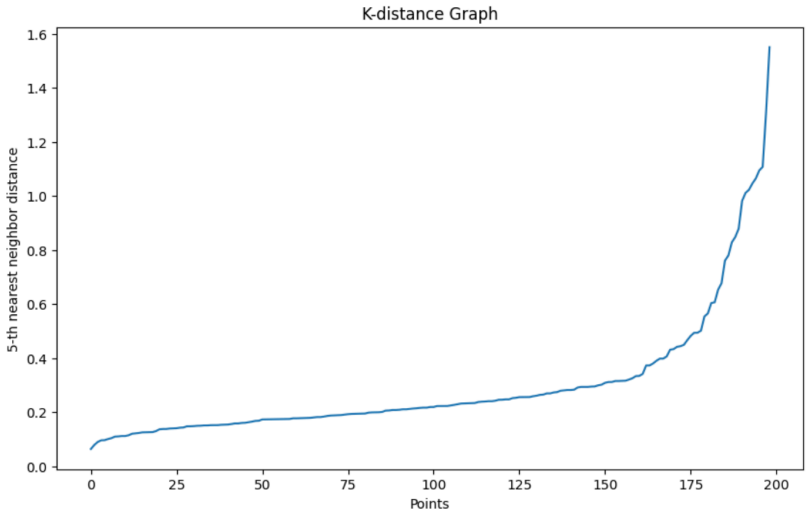
- Applied standardscaler on numeric data like transaction amount and one hot encoder in categorical data like product categories
- This ensures that features with larger scales do not dominate the clustering process.

Step 4: Dimensionality Reduction

- Used Principal Component Analysis (PCA)
- This step helps in visualizing clusters in 2D and selecting relevant data.

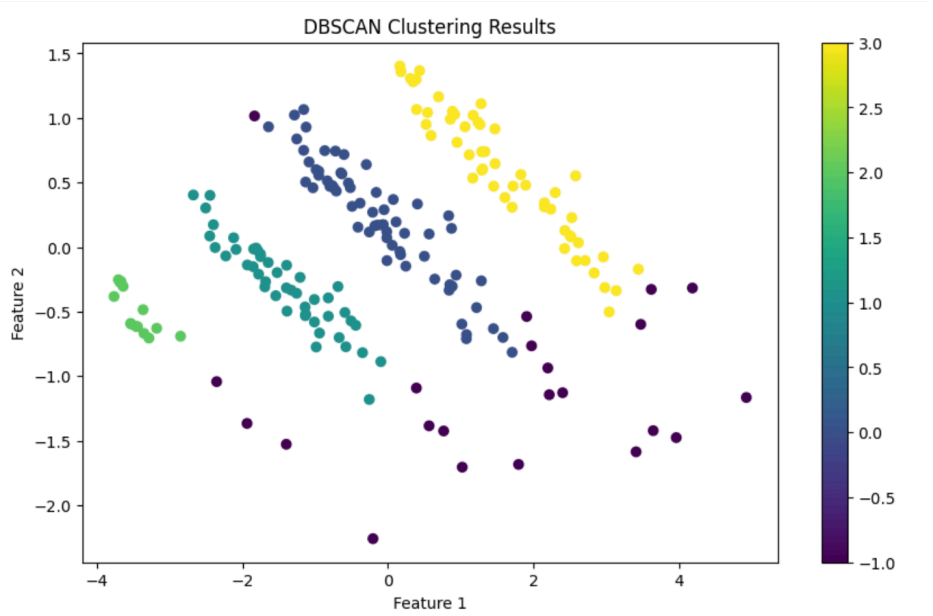
Step 5: Calculating K-nearest neighbors to find epsilon for DBSCAN

- We define a function `plot_k_distance_graph` that calculates the distance to the k-th nearest neighbor for each point.
- The distances are sorted and plotted.
- We look for an "elbow" in the resulting graph to choose epsilon.



Step 5: Calculating K-nearest neighbors to find epsilon for DBSCAN

- We use scikit-learn's DBSCAN implementation:
- We set epsilon=0.38 based on our k-distance graph.
- We set min_samples=6
- We fit the model to our data and predict the clusters.
- No of clusters=4



- DB index Value:

```
#ds score
from sklearn.metrics import davies_bouldin_score
ds=davies_bouldin_score(X,clusters)
ds
```

```
np.float64(1.197866734271172)
```