11/16/2024

DBSCAN VS OPTICS Interpretations

Contents

[Dataset 01 1](#_Toc182696580)

[Description of Bank Dataset: 1](#_Toc182696581)

[Correlation Analysis 3](#_Toc182696582)

[Dendrogram 4](#_Toc182696583)

[PCA and TSNE Visualization 5](#_Toc182696584)

[DBSCAN Interpretation with different parameters Values 7](#_Toc182696585)

[Interpretation of the OPTICS Results 9](#_Toc182696586)

[Reachability plot of OPTICS 10](#_Toc182696587)

[Comparison of DBSCAN & OPTICS 11](#_Toc182696588)

[Dataset 02 11](#_Toc182696589)

[Correlation Analysis 11](#_Toc182696590)

[Dendrogram 12](#_Toc182696591)

[DBSCAN Interpretation with different values on TSNE Data 13](#_Toc182696592)

[OPTICS Interpretation 14](#_Toc182696593)

# Dataset 01

## Description of Bank Dataset:

The dataset contains information about bank customers and their banking behaviors. Below is the detailed description of the provided columns in my dataset:

1. **CLIENTNUM**: A unique identifier for each customer. This serves as a primary key but does not hold analytical significance.
2. **Attrition\_Flag**: Indicates whether the customer is an active or inactive customer. This is the target variable for customer churn analysis.
3. **Customer\_Age**: The age of the customer in years. This numerical feature helps understand the age distribution of customers and its influence on banking behavior.
4. **Gender**: The gender of the customer (Male or Female). A categorical feature that can help analyze customer segmentation.
5. **Dependent\_count**: The number of dependents associated with the customer. A numerical feature that reflects family size and may correlate with spending patterns.
6. **Education\_Level**: The educational background of the customer (e.g., High School, Graduate). This categorical feature can be used to understand customer profiles.
7. **Marital\_Status**: The marital status of the customer (e.g., Married, Single). This categorical feature could provide insights into spending and savings habits.
8. **Income\_Category**: The income range of the customer (e.g., Less than $40K, $80K-$120K). This categorical feature is important for analyzing financial behavior.
9. **Card\_Category**: The type of credit card held by the customer (e.g., Blue, Gold). A categorical feature that may be linked to spending behavior.
10. **Months\_on\_book**: The total number of months the customer has been associated with the bank. A numerical feature indicating customer tenure.
11. **Total\_Relationship\_Count**: The total number of products the customer holds with the bank. A numerical feature showing the breadth of engagement with the bank.
12. **Months\_Inactive\_12\_mon**: The number of months the customer has been inactive in the last 12 months. This numerical feature can indicate potential churn.
13. **Contacts\_Count\_12\_mon**: The number of contacts the customer had with the bank in the last 12 months. A numerical feature representing engagement frequency.
14. **Credit\_Limit**: The credit limit assigned to the customer. A numerical feature reflecting financial capacity and credit behavior.
15. **Total\_Revolving\_Bal**: The total revolving balance on the customer’s account. This numerical feature helps understand debt patterns.
16. **Avg\_Open\_To\_Buy**: The average open-to-buy credit line available to the customer. A numerical feature that provides insight into credit utilization.
17. **Total\_Amt\_Chng\_Q4\_Q1**: The ratio of the total transaction amount between the fourth and first quarters. This numerical feature helps analyze changes in spending.
18. **Total\_Trans\_Amt**: The total amount of transactions made by the customer. A numerical feature representing financial activity.
19. **Total\_Trans\_Ct**: The total number of transactions made by the customer. A numerical feature showing the frequency of transactions.
20. **Total\_Ct\_Chng\_Q4\_Q1**: The ratio of the total transaction count between the fourth and first quarters. This feature captures changes in transaction patterns.
21. **Avg\_Utilization\_Ratio**: The average utilization ratio of the customer’s credit line. A numerical feature reflecting financial behavior.

I have run data cleaning functions and then got:

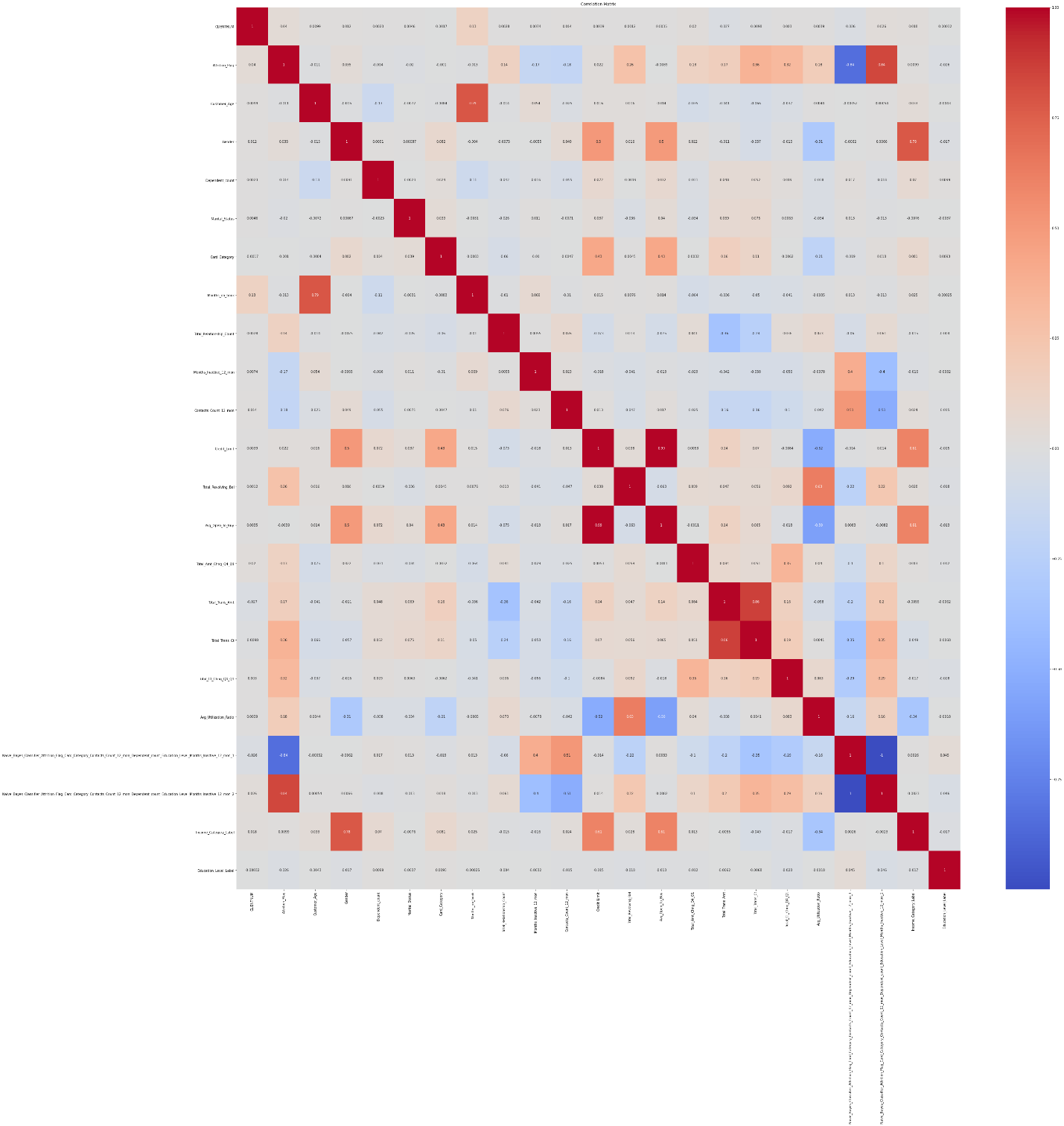
**Data after imputing missing values: (10127, 23)**

## Correlation Analysis

A few notable correlations are observed, such as a strong positive correlation between Total\_Trans\_Ct and Total\_Trans\_Amt indicating that customers who transact more frequently also tend to have higher transaction amounts.

Total\_Relationship\_Count shows a positive correlation with Months\_on\_book suggesting that longer-tenured customers typically maintain more relationships with the bank. On the other hand, Avg\_Utilization\_Ratio has a strong negative correlation with Avg\_Open\_To\_Buy, reflecting that customers who utilize a higher proportion of their credit have less available credit. Moderate correlations are also observed between Credit\_Limit and Avg\_Open\_To\_Buy , implying that customers with higher credit limits generally have more available credit.

Other relationships are weak or negligible, emphasizing that certain features are more independent and may not significantly impact one another. These insights can guide feature selection and preprocessing for clustering or predictive analysis tasks.



## Dendrogram

The dendrogram visually depicting how data points are grouped based on their similarities. Each leaf at the bottom corresponds to an individual data point while branches merge points or clusters at increasing levels of similarity as we move upward.

The height of the vertical lines indicates the dissimilarity between clusters being merged. Shorter vertical lines signify closely related clusters, while longer lines suggest more distinct clusters. In this dendrogram, the dataset can be divided into distinct clusters by cutting the tree at a particular height, which represents a chosen dissimilarity threshold.

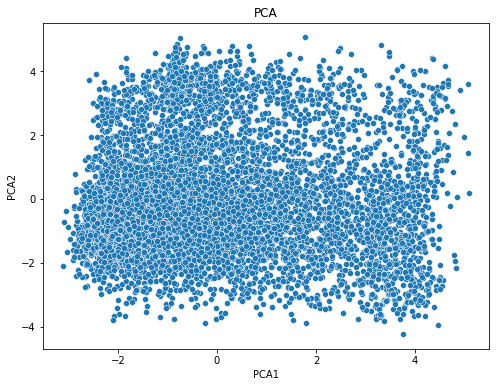
The branching structure shows the hierarchical nature of relationships among data points, with the largest clusters (on the far left, middle, and right) representing the major groupings in the bank data.

A green red and blue lines

Description automatically generated

## PCA and TSNE Visualization

The PCA plot reveals a scattered distribution of points, suggesting a lack of clear clustering or distinct patterns. This indicates that the original dataset, likely consisting of various banking features like customer demographics, transaction history, and credit scores, does not exhibit obvious groupings. The data points are relatively evenly spread across the space, without forming well-defined clusters.



The t-SNE plot reveals a complex and non-linear structure within the data , in given picture it suggesting that the underlying relationships between data points are intricate and cannot be easily captured by linear dimensionality reduction techniques like PCA. The points are dispersed throughout the space, with some local clusters and elongated structures.

The lack of distinct, well-separated clusters indicates that the data points are not easily classifiable into distinct groups based on their features. This might suggest that the dataset contains a high degree of variability or that the features used for analysis are not sufficiently informative for clear A blue and white dot pattern

Description automatically generated with medium confidenceseparation.

## DBSCAN Interpretation with different parameters Values

The DBSCAN algorithm, applied to the bank dataset with varying epsilon and MinPoints values, reveals insights into the data's underlying structure. Lower epsilon values with higher MinPoints identify finer-grained clusters, while larger epsilon values with lower MinPoints capture broader patterns. The data exhibits a moderate level of density, with both dense and sparse regions. A significant portion of the data is classified as noise, indicating that it might not belong to well-defined clusters.

As per my understanding of given results can be following results as show in below figure.

**Epsilon (ε) = 0.5:**

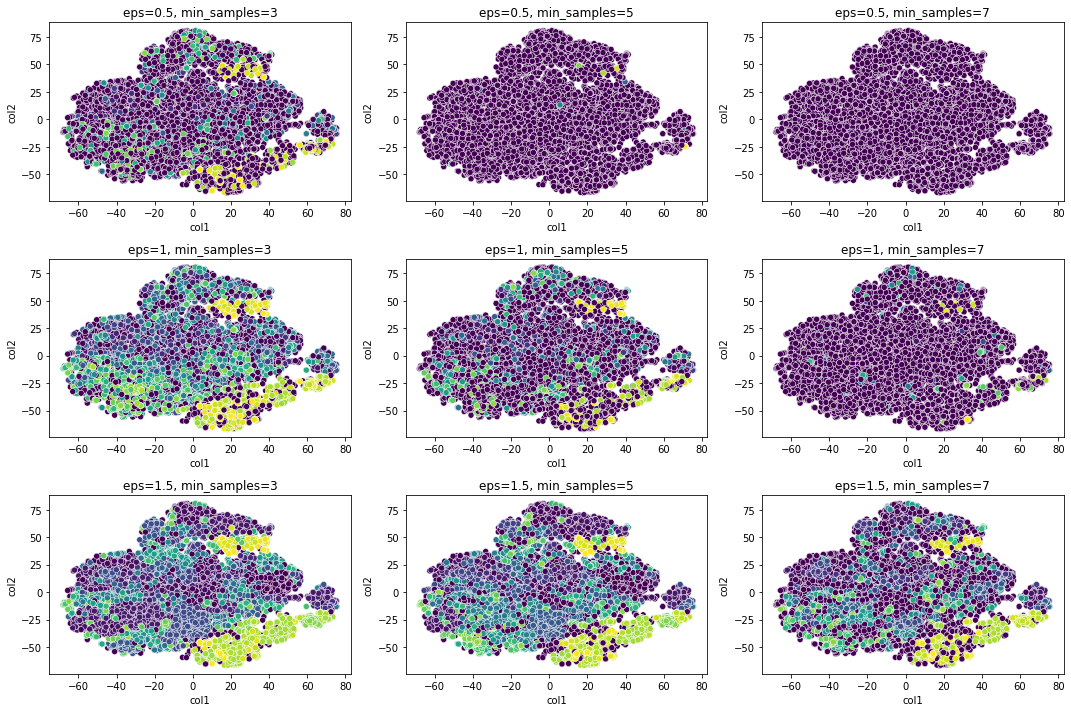
* **MinPoints = 3:** This configuration yields a large number of clusters likely due to the small neighborhood radius. Many isolated points are classified as noise.
* **MinPoints = 5:** With a higher MinPoints, the number of clusters decreases, and more points are classified as noise. This indicates that smaller clusters are merged or eliminated.
* **MinPoints = 7:** Further increasing MinPoints leads to even fewer clusters, suggesting that only the densest regions are identified as clusters.

**Epsilon (ε) = 1:**

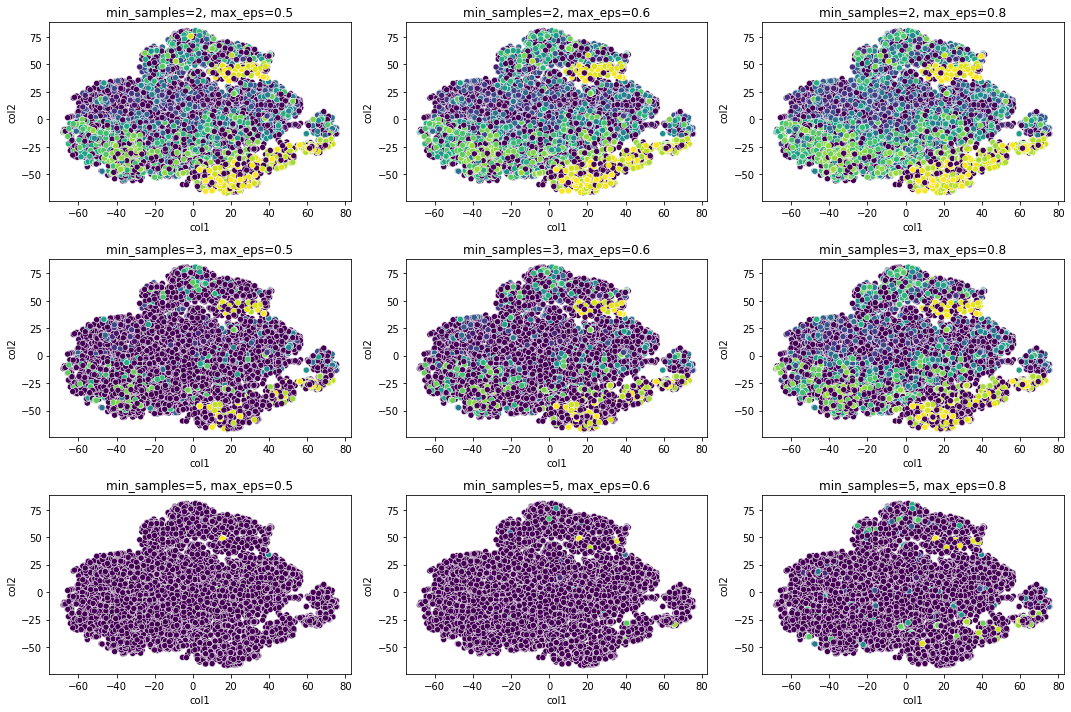
* **MinPoints = 3:** This configuration results in fewer clusters than with ε = 0.5 indicating that larger neighborhoods are considered. The number of noise points also decreases.
* **MinPoints = 5:** Similar to the previous case increasing MinPoints leads to fewer clusters and more noise points.
* **MinPoints = 7:** with a higher MinPoints, the number of clusters is further reduced and the algorithm becomes more stringent in identifying dense regions.

**Epsilon (ε) = 1.5:**

* **MinPoints = 3:** This configuration produces the fewest clusters suggesting that the large neighborhood radius encompasses many points leading to fewer distinct groups.
* **MinPoints = 5:** With a higher MinPoints the number of clusters increases slightly indicating that some previously merged clusters are now separated.
* **MinPoints = 7:** Further increasing MinPoints leads to a similar number of clusters as with MinPoints = 5 suggesting that the majority of the data points are distributed in relatively dense regions.



## Interpretation of the OPTICS Results



***Parameters:***

**max\_eps:** This parameter defines the maximum radius of the neighborhood around a data point. Points within this radius are considered neighbors.

**min\_samples:** This parameter specifies the minimum number of points required to form a dense region.

***Different values interpretions:***

**min\_samples=2:**

* + **max\_eps=0.5:** This configuration results in a relatively large number of clusters, indicating that even small dense regions are identified as separate clusters.
  + **max\_eps=0.6:** Increasing max\_eps leads to fewer clusters as larger neighborhoods are considered, and some previously identified clusters are merged.
  + **max\_eps=0.8:** With a further increase in max\_eps the number of clusters decreases even more suggesting that only the densest regions are identified as clusters.

**min\_samples=3:**

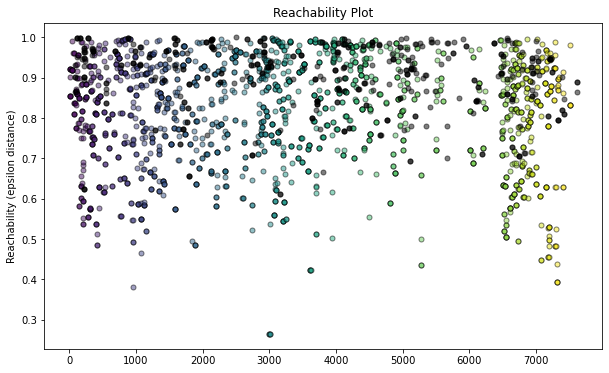
* + **max\_eps=0.5:** Similar to the previous case, a smaller max\_eps leads to a larger number of clusters.
  + **max\_eps=0.6:** Increasing max\_eps results in fewer clusters, as more points are considered neighbors.
  + **max\_eps=0.8:** With the largest max\_eps value, the number of clusters is further reduced, indicating that only the most prominent dense regions are identified.

**min\_samples=5:**

* + **max\_eps=0.5:** This configuration produces the fewest clusters, as a higher min\_samples value requires more points to form a dense region.
  + **max\_eps=0.6:** Increasing max\_eps leads to a slight increase in the number of clusters as larger neighborhoods allow for more points to be considered neighbors.
  + **max\_eps=0.8:** With the largest max\_eps value, the number of clusters remains relatively low indicating that only the most significant dense regions are identified.

So we can say that bank data appears to have a moderate level of density, with some dense regions and some sparse areas.A smaller max\_eps with a higher min\_samples can identify finer-grained clusters, while a larger max\_eps with a lower min\_samples can capture broader, more general patterns.

## Reachability plot of OPTICS

The reachability plot provides valuable insights into the underlying structure of the bank dataset. By analyzing the valleys and noise points in the plot we can identify potential clusters and outliers. The depth of the valleys indicates the density of the clusters, while the distance between valleys suggests the separation between them. 

## Comparison of DBSCAN & OPTICS

DBSCAN or OPTICS provides better results for the bank dataset. Both algorithms have their strengths and weaknesses, and the optimal choice depends on the characteristics of the data and the desired outcomes.

**DBSCAN** is well-suited for identifying clusters of arbitrary shape and handling noise. However, it can be sensitive to the choice of hyperparameters particularly the epsilon value. If epsilon is too small, many points may be classified as noise, while if it is too large clusters may merge. DBSCAN may struggle with datasets that have varying densities.

**OPTICS** is a more robust algorithm that can handle varying densities and identify clusters of different shapes. It also provides a clustering ordering, which can be useful for visualizing the data and identifying potential subclusters. However, OPTICS can be computationally expensive, especially for large datasets.

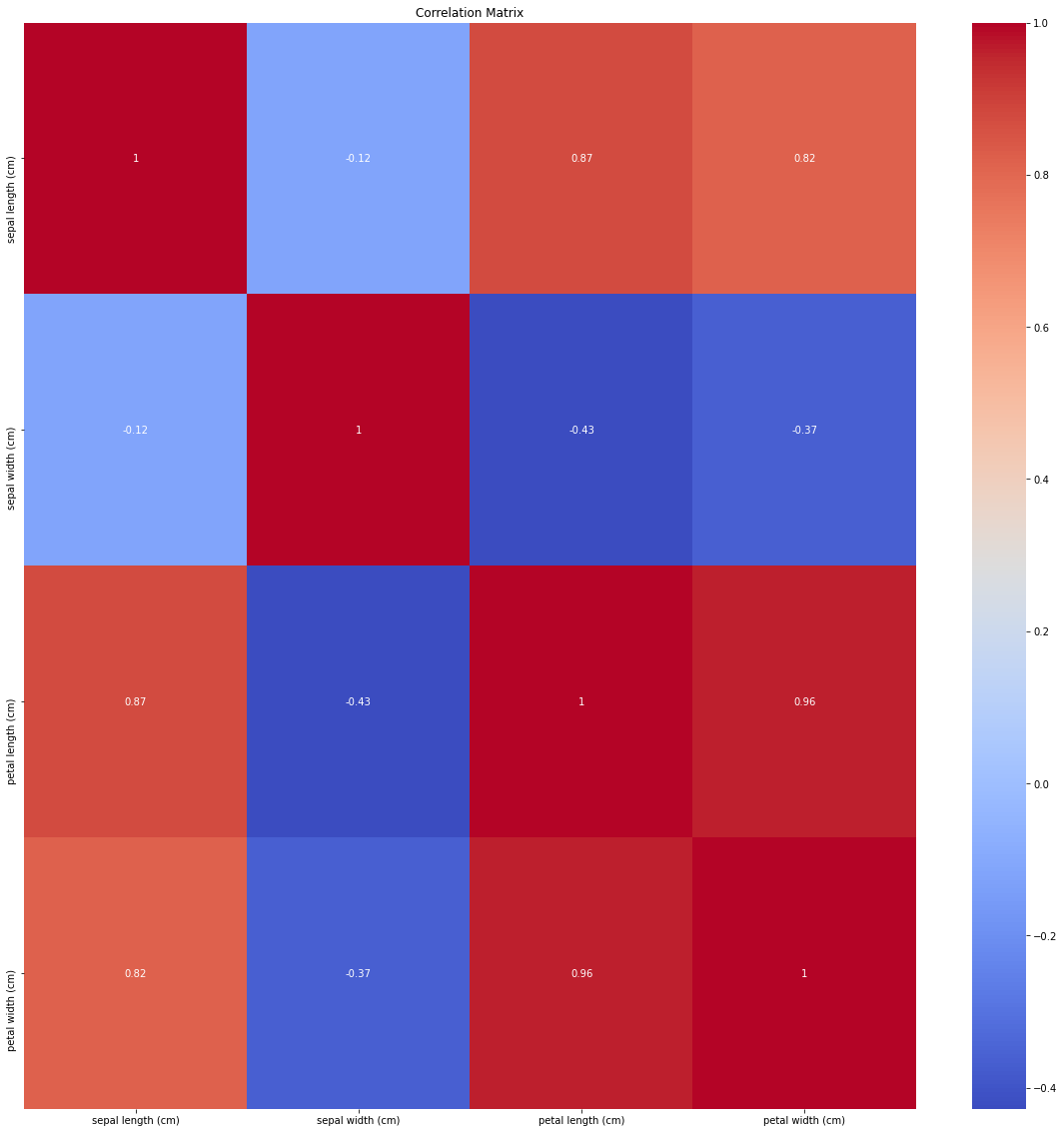
For my bank dataset, the choice between DBSCAN and OPTICS can be:

* **Data Density:** If the data has varying densities, OPTICS might be a better choice, as it can handle these variations more effectively.
* **Cluster Shape:** If the clusters are expected to have arbitrary shapes, DBSCAN can be a good choice as it is more flexible in identifying clusters of different forms.
* **Noise Handling:** Both algorithms can handle noise but OPTICS might be more robust in identifying and handling noisy points.
* **Computational Cost:** If computational resources are limited DBSCAN might be a more efficient option as it has a lower computational complexity than OPTICS.

So we can conclude that OPTICS appears to be a promising choice for this dataset due to its ability to handle varying densities and identify clusters of arbitrary shapes. This is particularly important for bank datasets which often exhibit complex structures and outliers.

# Dataset 02

## Correlation Analysis

The correlation matrix reveals strong positive correlations between petal length and petal width, and between petal length and sepal length. A weak positive correlation exists between sepal length and sepal width, while a weak negative correlation is observed between petal length and sepal width. 

## Dendrogram

The vertical axis represents the distance between clusters, and the horizontal axis shows the individual data points. As we move up the dendrogram with the height of the vertical lines indicating the distance between the merged clusters. The dendrogram reveals distinct clusters that the Iris dataset can be well-separated into groups based on its features. A white background with green and red lines

Description automatically generated

## DBSCAN Interpretation with different values on TSNE Data

As per my understanding, The DBSCAN algorithm's performance, Lower eps values with higher min\_samples identify finer-grained clusters, while higher eps values with lower min\_samples capture broader patterns. In the case of the my dataset, a moderate eps and min\_samples combination seems to effectively balance the trade-off between identifying distinct clusters and avoiding over-segmentation.

For gicendataset, values like eps=0.9 and min\_samples=6 seem to produce a good compromise, identifying three distinct clusters that align well with the known class labels of the dataset.

If we consider low values of both parameter then it tends to over-cluster the data, resulting in numerous small clusters. Many data points are classified as noise.

A group of white rectangles with different colored squares

Description automatically generated

## OPTICS Interpretation

The results highlight that Lower min\_samples and max\_eps values tend to over-cluster the data, leading to numerous small, potentially noisy clusters. Conversely, higher values can result in under-clustering, where the data is grouped into a few large, less informative clusters.

For given dataset, a balanced approach with moderate min\_samples and max\_eps values appears to yield the most accurate clustering results. This configuration effectively captures the three distinct classes inherent in the dataset.