**Unsupervised Machine Learning**

**SEMESTER : II**

**Topic: Anomaly Detection**

**Name : Yash Dudeja**

**SAP ID: 86092300013**

**Roll No : A013**

Problem Statement :

To use a unsupervised technique to find anomalies in the Ethereum transactions, so we get to know the irregular patterns.

1. **Introduction**:

Anomaly detection plays a pivotal role in various domains, including cybersecurity, finance, and industrial systems, where the identification of rare and potentially harmful events is paramount. Despite its significance, existing anomaly detection techniques often face challenges in accurately distinguishing anomalies from normal behavior, particularly in complex and dynamic environments. In this project, we have used a unsupervised machine learning technique for anomaly detection that helps us to identify anomalies. By integrating data sources and utilizing sophisticated anomaly scoring mechanisms, our method demonstrates and bifurcates the outliers which are the anomalies in the dataset.

1. **Objective**:

The objective or the purpose of this project is to find the outliers/irregular patterns present in the dataset by using unsupervised technique and help in reducing frauds.

1. **Methodology**:

**Isolation Forest:** Isolation Forest is a popular anomaly detection algorithm that is based on the concept of isolating anomalies rather than explicitly modeling normal data points. Here's a brief overview of its methods and steps:

* + 1. Random Partitioning:
       1. Isolation Forest randomly selects a feature and then randomly selects a split value between the maximum and minimum values of that feature to create a partition.
       2. This process is repeated recursively until each data point is isolated in its own partition, or until a predefined maximum depth is reached.
    2. Anomaly Score Calculation:
       1. The algorithm measures the number of partitions needed to isolate each data point.
       2. Anomalies are expected to be isolated more quickly since they are less numerous and have attribute values that are very different from those of normal instances.
       3. Therefore, the anomaly score is inversely proportional to the number of partitions required: the fewer the partitions, the higher the anomaly score.
    3. Decision Making:
       1. A decision threshold is defined to classify data points as either anomalies or normal.
       2. Data points with anomaly scores above the threshold are labeled as anomalies, while those below are labeled as normal.
    4. Advantages:
       1. Isolation Forest is efficient and scalable, especially for high-dimensional data.
       2. It does not rely on distance or density measures, making it robust to outliers and noise.
       3. It can handle both global and local anomalies effectively.
    5. Steps:
       1. Initialization: Start with the entire dataset.
       2. Feature Selection: Randomly select a feature.
       3. Split Selection: Randomly select a split value for the chosen feature.
       4. Partitioning: Partition the data based on the selected feature and split value.
       5. Recursive Partitioning: Repeat steps 2-4 recursively until each data point is isolated or until a maximum depth is reached.
    6. Anomaly Score Calculation:
       1. Calculate the anomaly score for each data point based on the number of partitions required to isolate it.
    7. Decision Making:
       1. Classify data points as anomalies or normal based on the anomaly score and a predefined threshold.

Isolation Forest offers a simple yet powerful approach to anomaly detection, making it suitable for a wide range of applications.

**DBSCAN** : (Density-Based Spatial Clustering of Applications with Noise) is a popular clustering algorithm used in machine learning and data mining. It's particularly effective for identifying clusters of varying shapes and sizes in a dataset, as well as detecting outliers (noise). Here's a brief overview of the methodology behind DBSCAN:

1. Density-Based: DBSCAN identifies clusters based on the density of data points. It defines clusters as dense regions of points separated by regions of lower density.
2. Core Points: In DBSCAN, a core point is a data point that has at least a specified number of other points (MinPts) within a specified distance (eps) in its neighborhood.
3. Border Points: Border points are not core points themselves but are within the neighborhood of a core point. They can belong to the cluster of that core point.
4. Noise Points: Noise points are data points that do not belong to any cluster. They are neither core points nor border points.

Algorithm Steps:

Step 1:Parameter Selection: The algorithm requires two parameters to be set: eps (the maximum distance between two points to be considered in the same neighborhood) and MinPts (the minimum number of points required to form a dense region or cluster).

Step 2: Neighbor Search: For each data point, calculate its neighborhood within eps distance.

Step 3: Core Point Identification: Identify core points by checking if the number of neighbors within eps distance is greater than or equal to MinPts.

Step 4: Cluster Formation: Form clusters by connecting core points to their directly reachable neighbors (including other core points) through a series of chains of core points and border points.

Step 5: Noise Point Identification: Assign remaining points that are not core points or reachable from any core points as noise points.

Key Advantages:

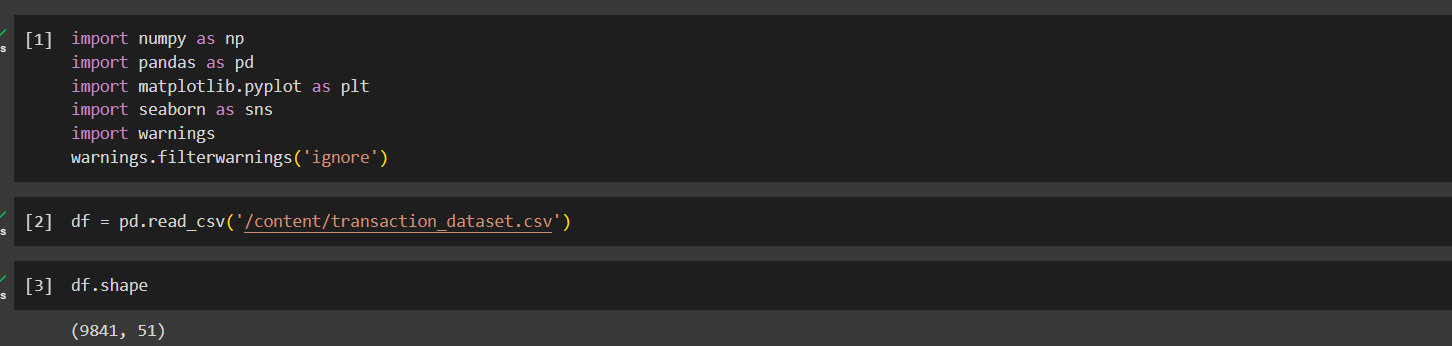
1. Can find clusters of arbitrary shapes.
2. Robust to outliers and noise.
3. Doesn't require specifying the number of clusters beforehand.

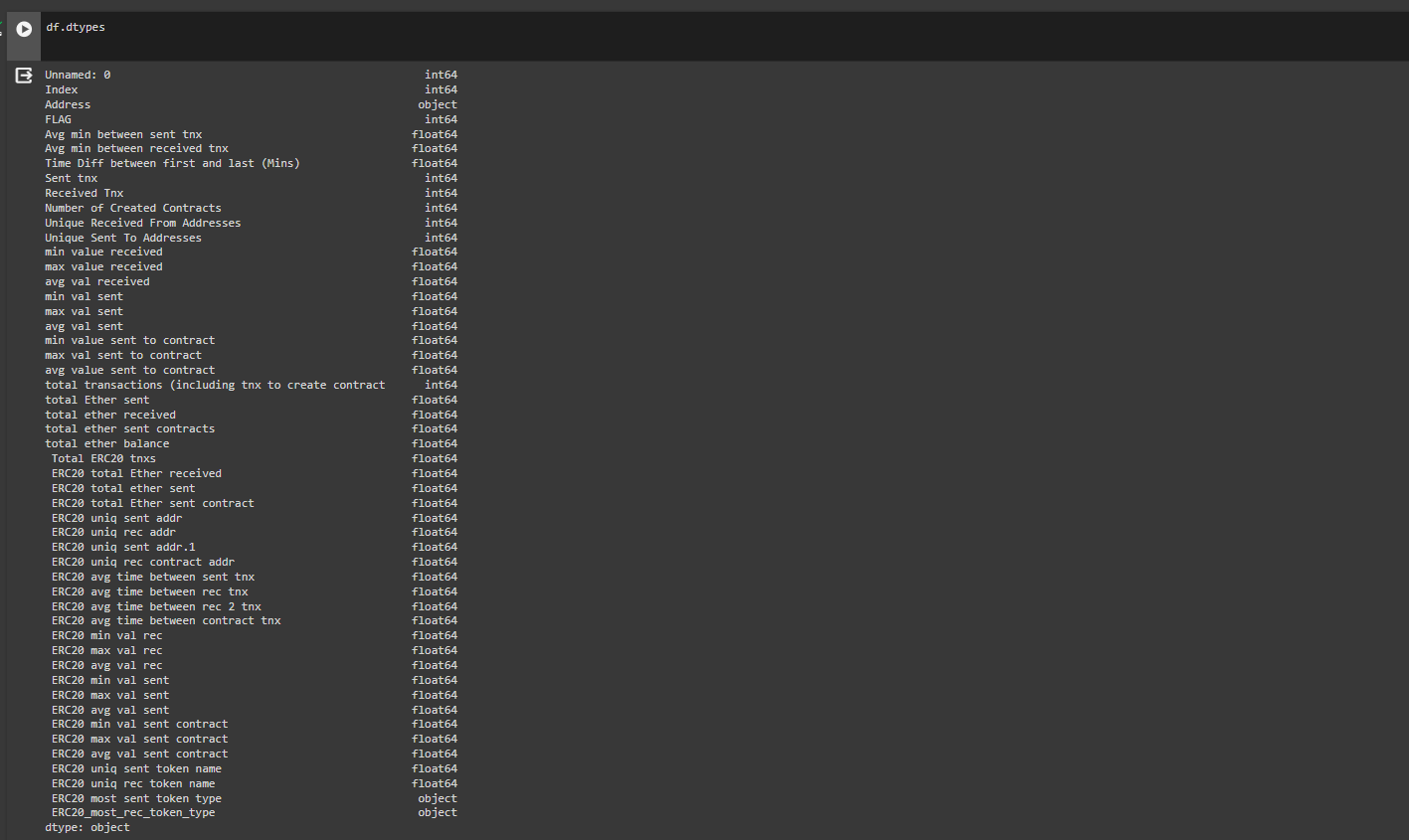
Key Disadvantages:

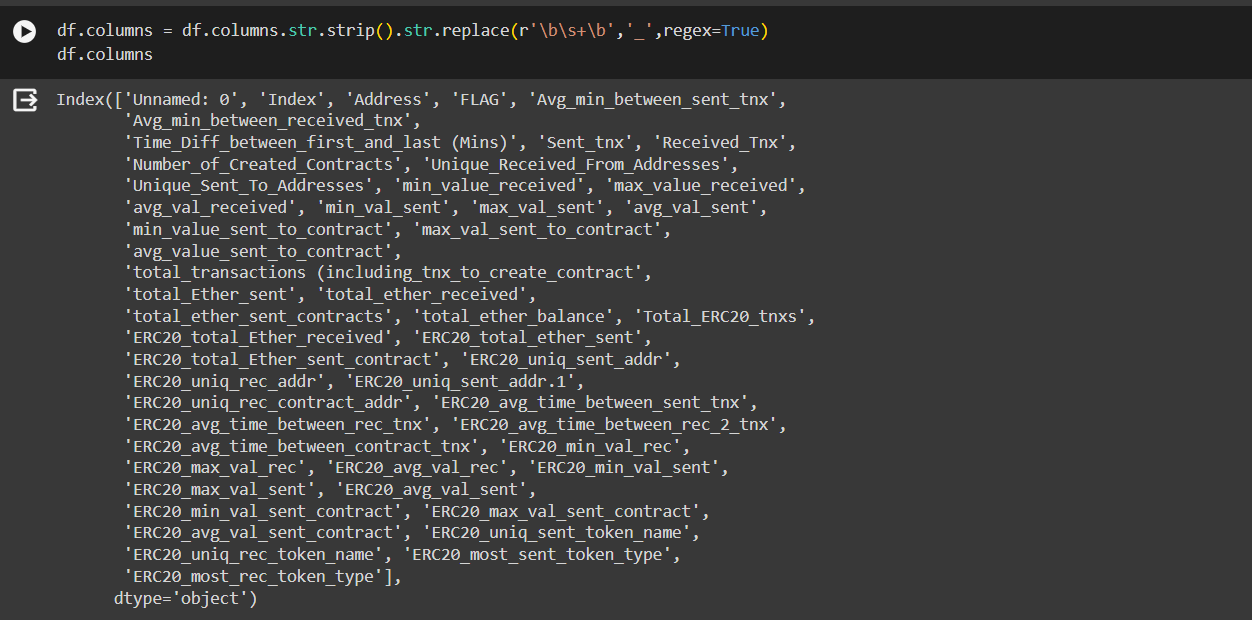
1. Sensitivity to parameter settings, especially eps and MinPts.
2. Can struggle with clusters of varying densities.
3. Computationally intensive for large datasets.

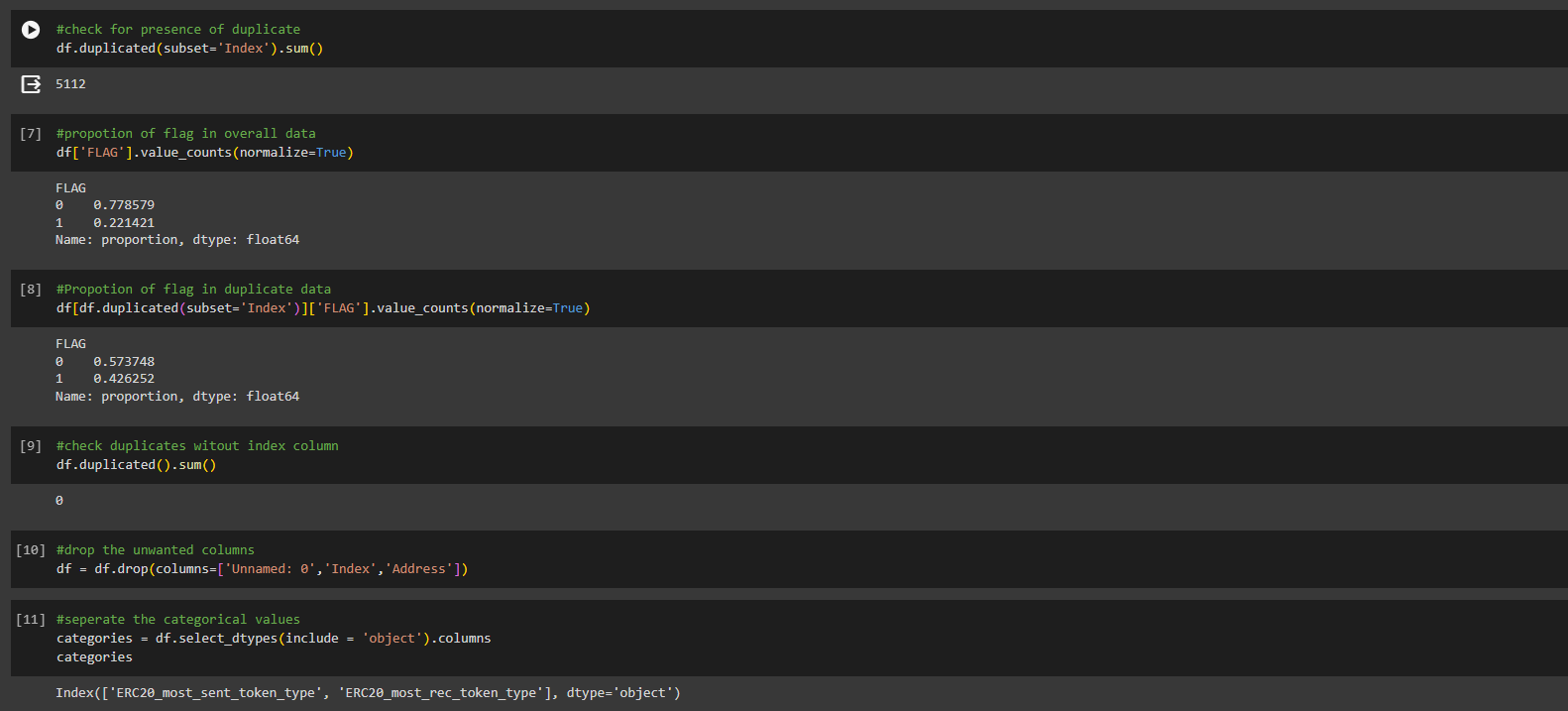
Applications:

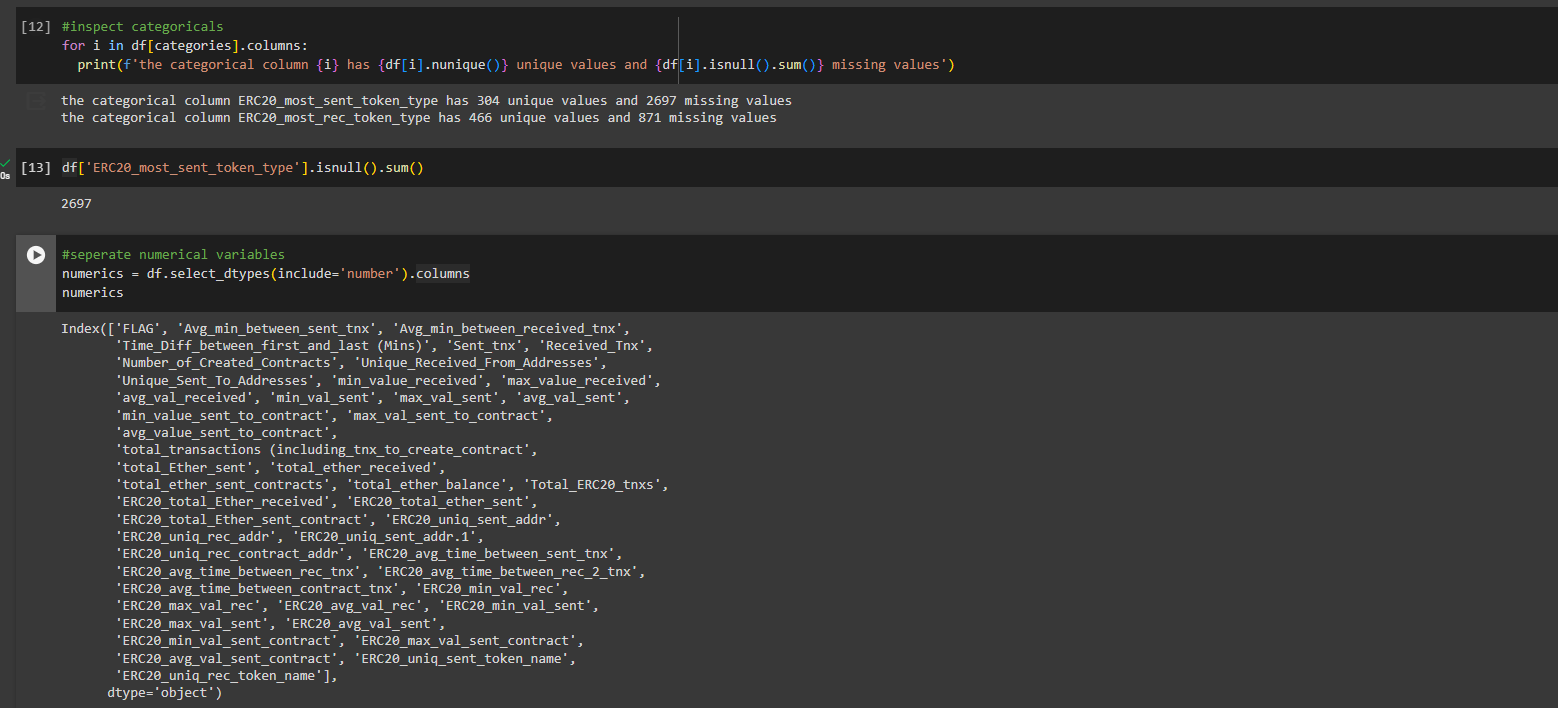
1. Fraud detection
2. Image segmentation
3. Customer segmentation
4. Anomaly detection
5. **Results**:

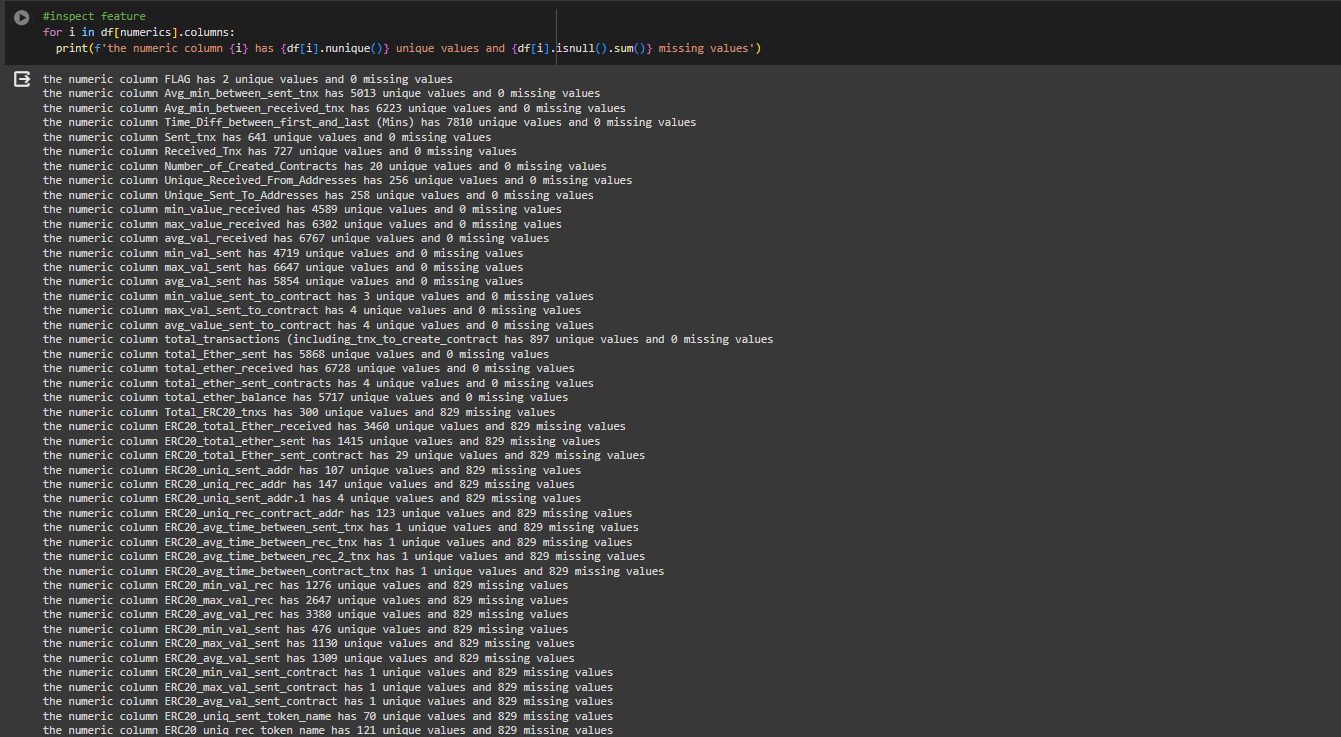


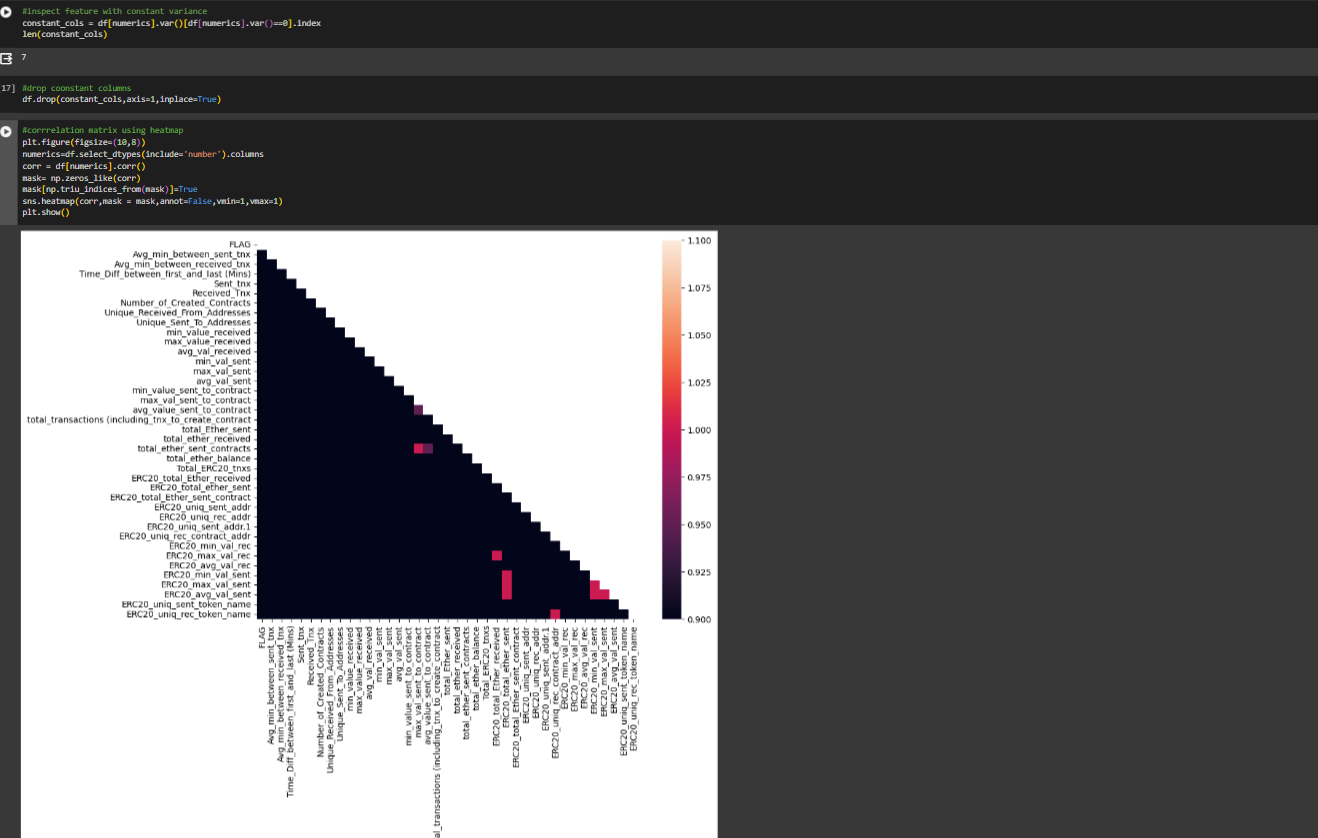


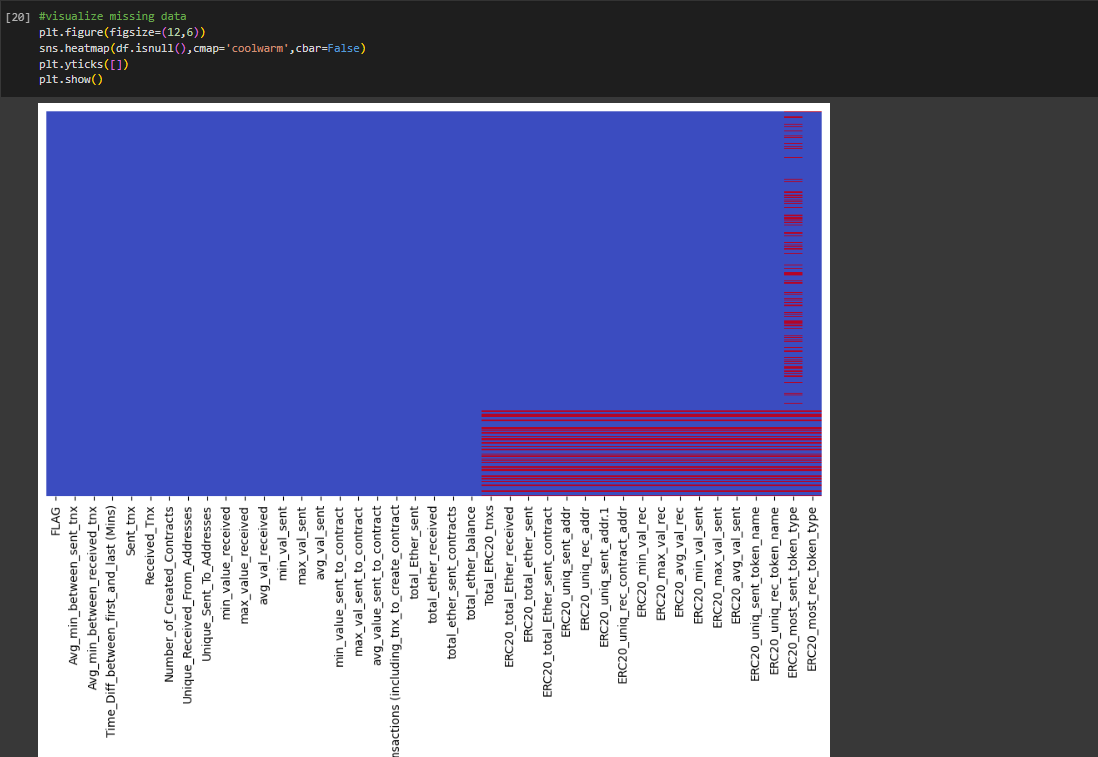


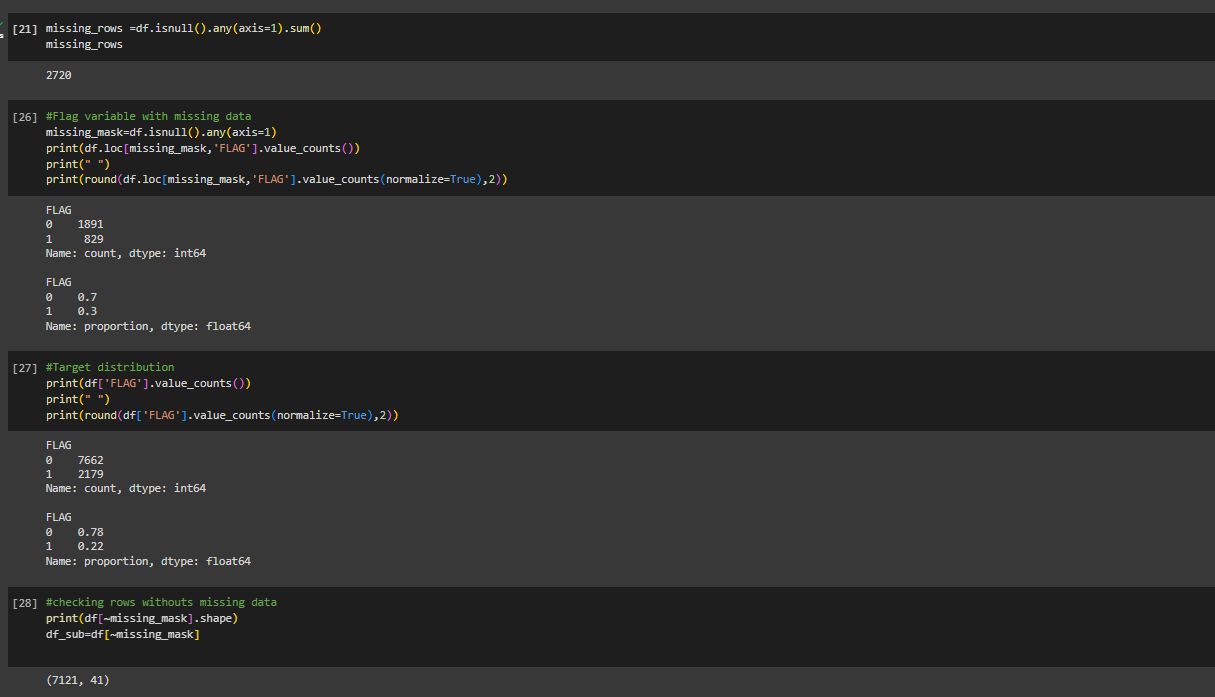


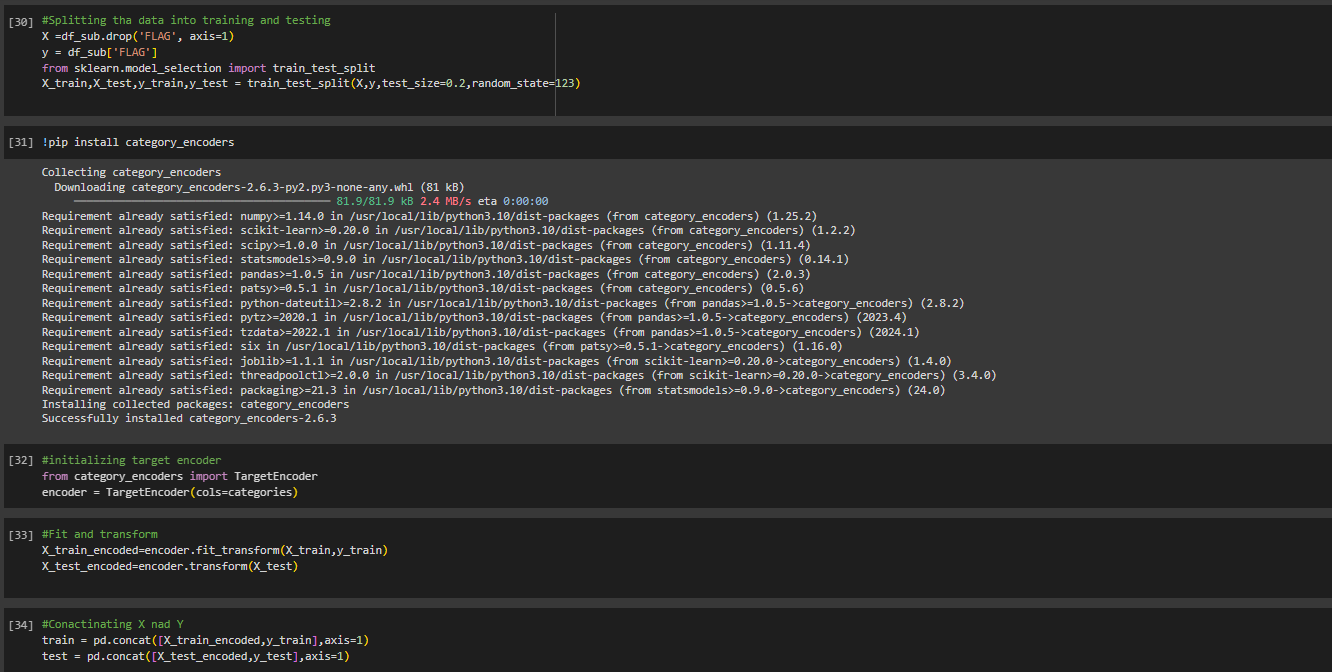




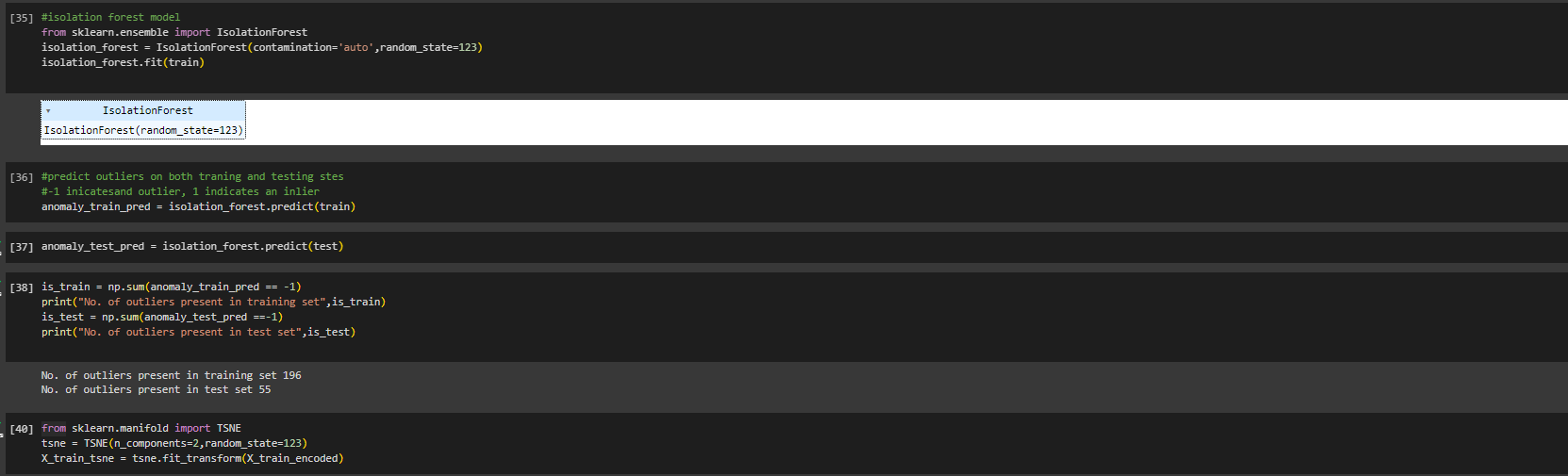


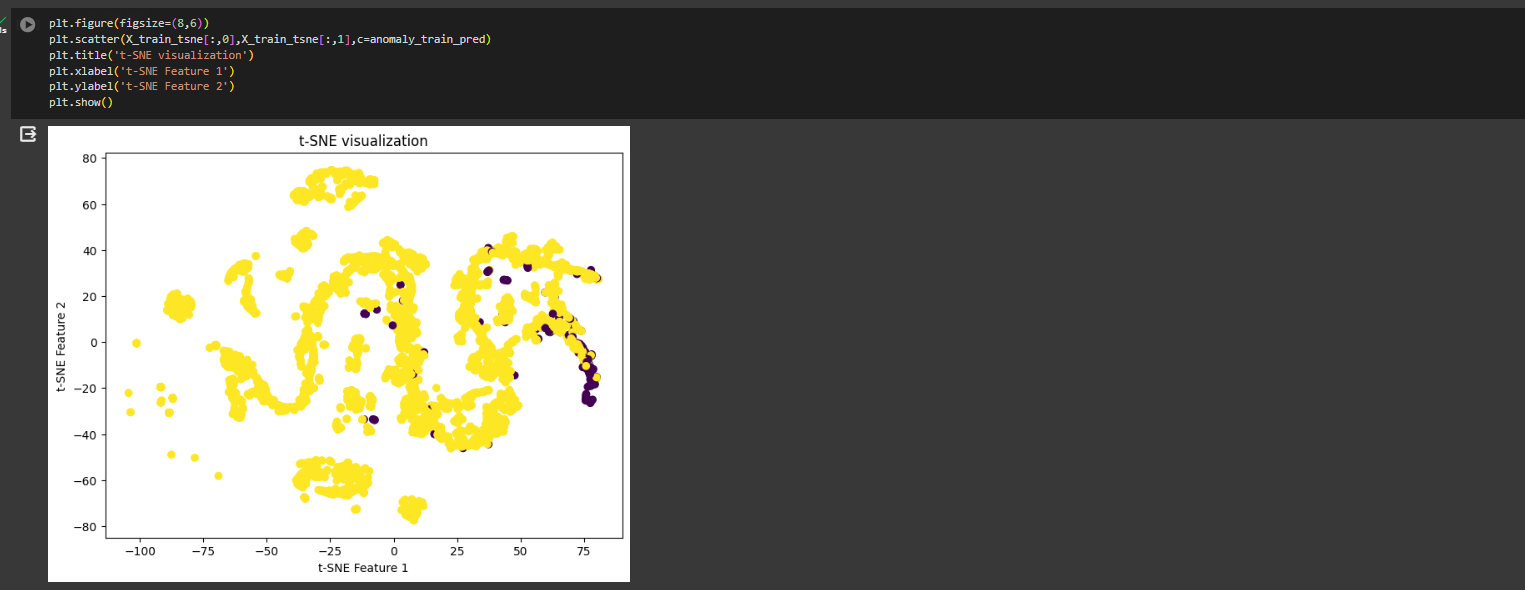




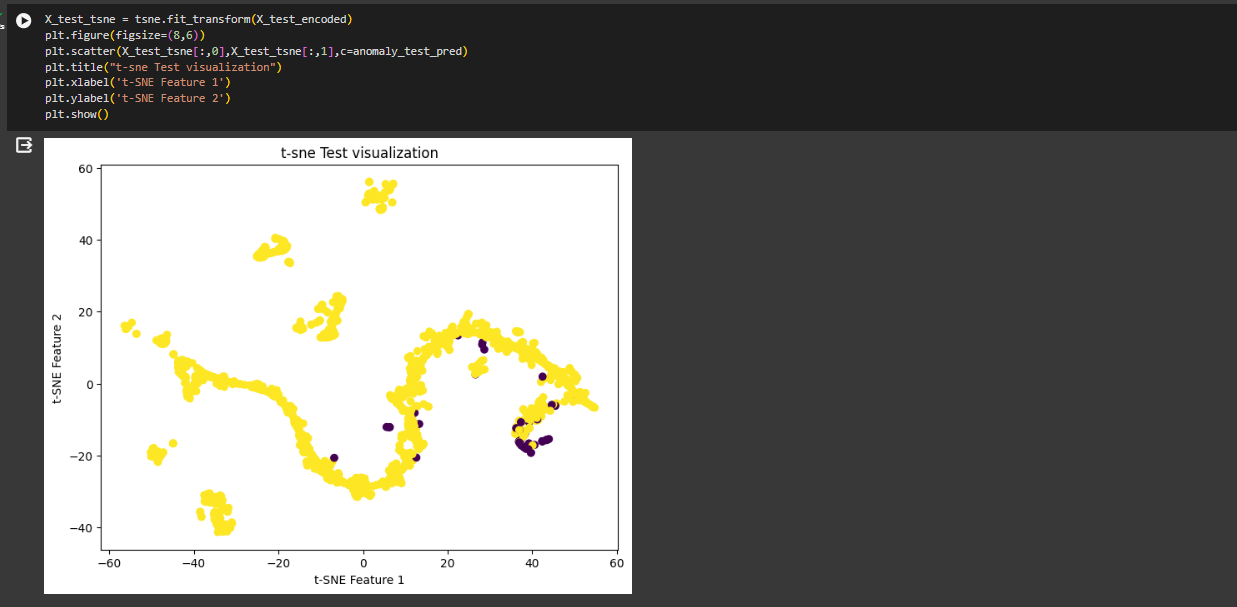


**ISOLATION FOREST**



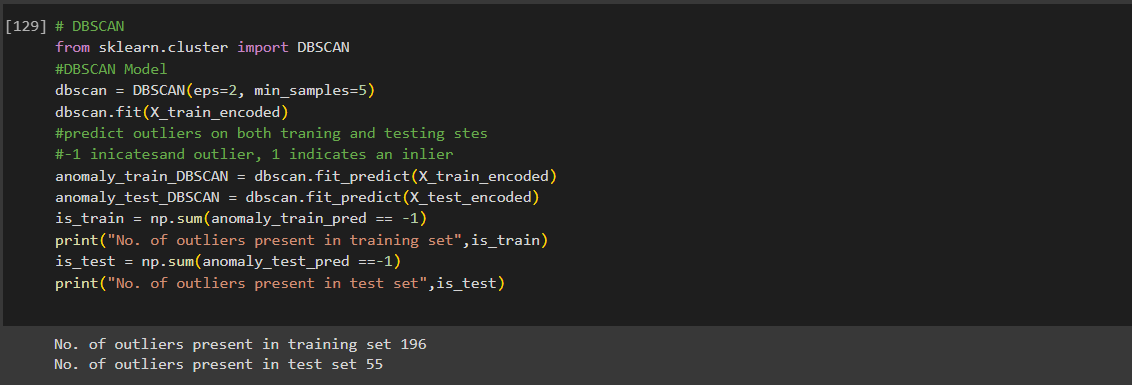


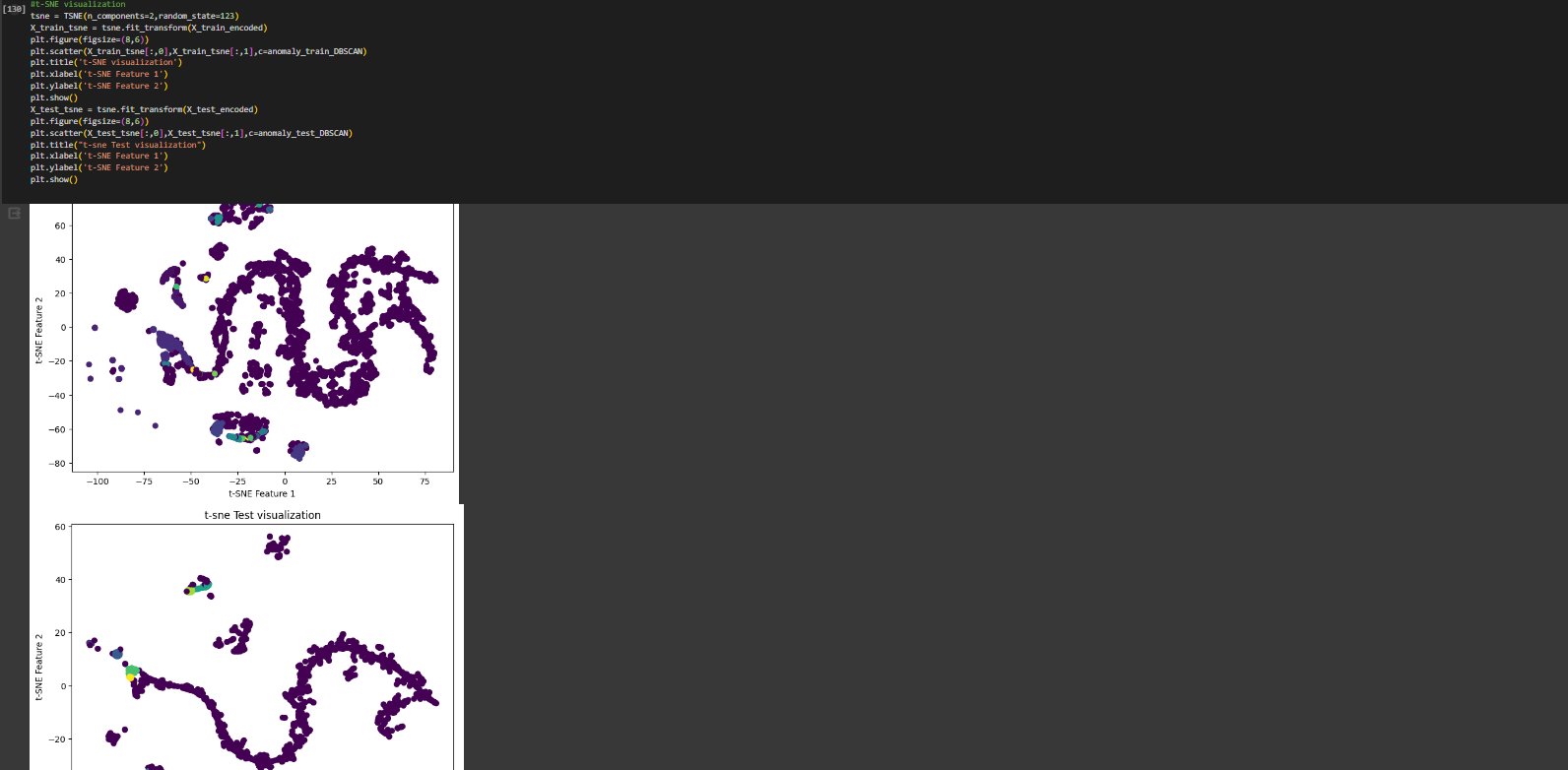
Outliers in Training Set

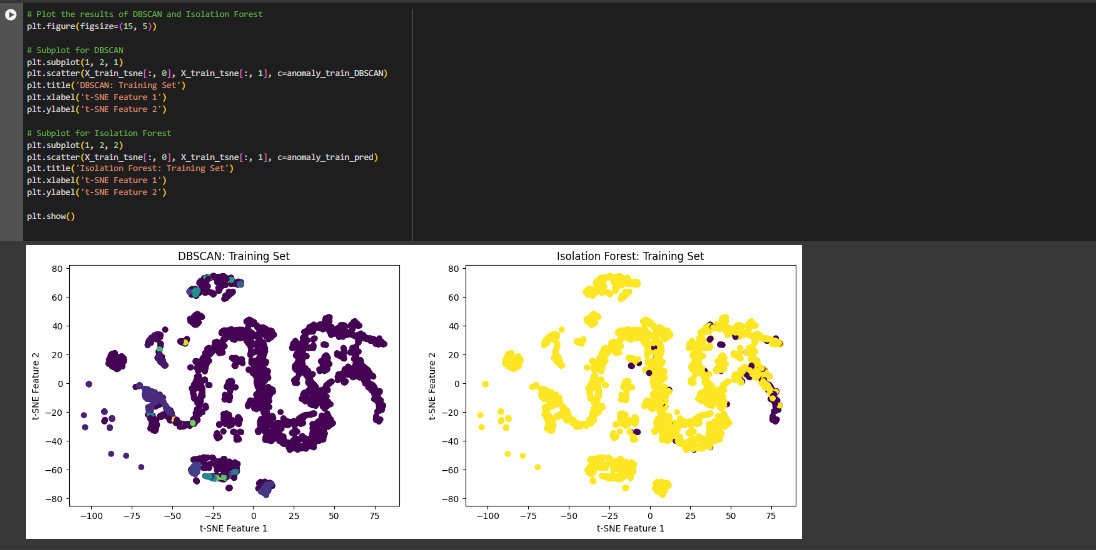


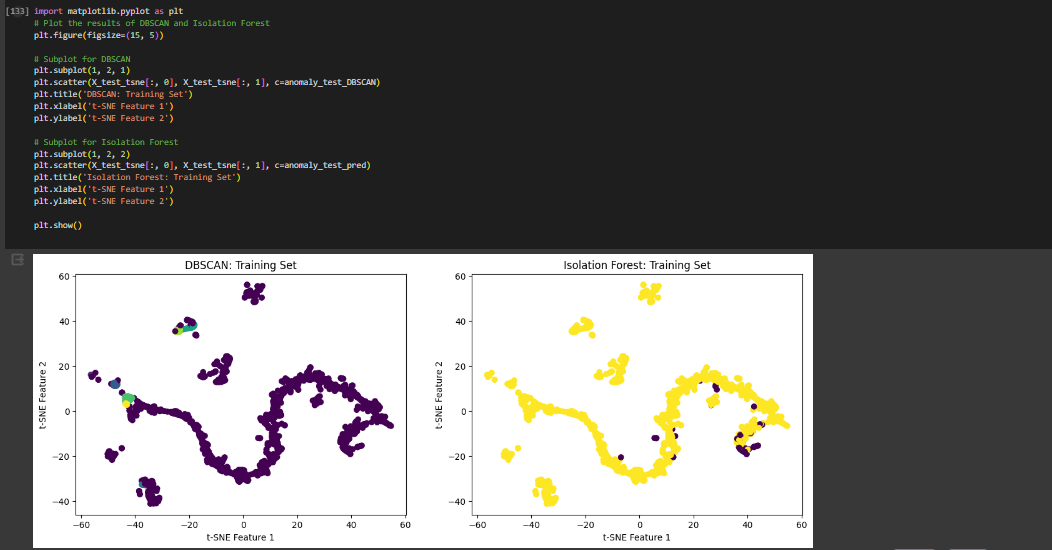
Outliers in Testing Set

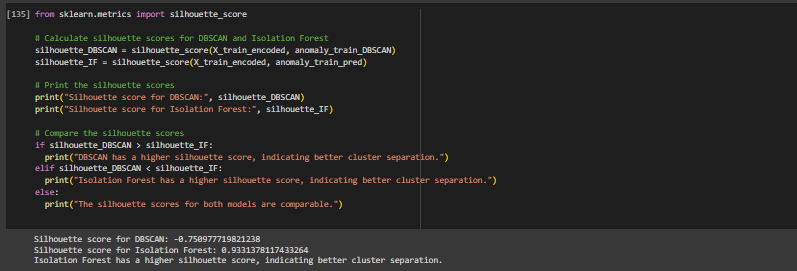
**DBSCAN**

****

****

****

****

****

1. **Conclusion**:

Isolation Forest offers several advantages over DBSCAN and other unsupervised techniques in certain scenarios:

* 1. Efficiency and Scalability:
     + 1. Isolation Forest is particularly efficient for high-dimensional datasets because it does not rely on distance calculations, making it less affected by the curse of dimensionality.
       2. DBSCAN, on the other hand, computes distances between all pairs of data points, which can become computationally expensive as the dataset grows or as the dimensionality increases.
  2. Robustness to Outliers:
     + 1. Isolation Forest is inherently robust to outliers and noise in the data. It isolates anomalies by their rarity and distinctiveness, rather than their proximity to other points.
       2. DBSCAN, while effective at identifying dense regions of data points, can be sensitive to outliers and may struggle to correctly classify them, especially in datasets with varying densities or non-globular shapes.
  3. Parameter Sensitivity:
     + 1. Isolation Forest has fewer hyperparameters to tune compared to DBSCAN, making it easier to use and less sensitive to parameter settings.
       2. DBSCAN requires tuning parameters such as epsilon (ε) and the minimum number of points (MinPts), which can significantly affect its performance and may require domain knowledge or extensive experimentation to optimize.
  4. Handling Varying Density:
     + 1. Isolation Forest is effective at detecting anomalies in datasets with varying densities or complex shapes because it does not rely on density-based clustering.
       2. DBSCAN is well-suited for identifying dense regions but may struggle with datasets containing clusters of varying densities or irregular shapes, as it relies on defining density-connected points.
  5. Interpretability:
     + 1. Isolation Forest provides straightforward anomaly scores for each data point, making it easier to interpret and understand the results.
       2. DBSCAN, while effective, may require additional post-processing or interpretation of the clustering results to identify anomalies or outliers.

As in the result you can see the silhouette score for isolation forest in more than DBSCAN

Overall, the choice between Isolation Forest, DBSCAN, and other unsupervised techniques depends on the specific characteristics of the dataset, the nature of the anomalies, and the computational resources available. Isolation Forest is particularly well-suited for high-dimensional datasets with varying densities and where outliers are a concern.