Anomaly detection :

Anomaly detection is a vital data analysis technique used to identify patterns or instances within a dataset that significantly deviate from the expected norm. By leveraging various statistical, machine learning, or artificial intelligence methods, anomaly detection aims to uncover data points that are unusual, rare, or suspicious in nature. This process involves establishing a baseline or normal behavior from historical data and then flagging any observations that fall outside this established range. Anomaly detection finds applications in a wide range of fields, including fraud detection, network security, industrial equipment monitoring, and healthcare, where the timely identification of anomalies can lead to enhanced decision-making and proactive intervention.

Flow chart:

Load and visualize the data

Data preprocessing

Create Anomaly detection Model

Train the Model

Detect and plot the Anomaly

1. Read data from CSV file: The data is read from a CSV file and stored in a Pandas DataFrame.
2. Convert Timestamp column to datetime object: The Timestamp column is converted to a datetime object so that it can be used for plotting.
3. Check if data has at least two rows: The data is checked to see if it has at least two rows. If it does not, then an error message is printed and the process ends.
4. Convert data into NumPy array, excluding non-numerical columns: The data is converted into a NumPy array, excluding any non-numerical columns. This is done so that the LOF model can be fit to the data.
5. Create LocalOutlierFactor model: A LocalOutlierFactor (LOF) model is created. The LOF model is a machine learning algorithm that can be used to identify anomalies in data.
6. Fit LOF model to data: The LOF model is fit to the data. This means that the model learns the normal patterns of the data.
7. Find indices of data points outside the range [-3, 3]: The indices of the data points that are outside the range [-3, 3] are found. These data points are considered to be anomalies.
8. Find most common value in second column: The most common value in the second column is found. This value is considered to be the normal value for the data.
9. Create plot: A plot is created, showing the data points with anomalies marked. The most common value is also marked on the plot.
10. Mark anomalies on plot: The data points that are considered to be anomalies are marked on the plot. This is done by drawing a red vertical line at the corresponding timestamp.
11. Mark most common value on plot: The most common value is marked on the plot. This is done by drawing a green horizontal line at the corresponding value.
12. Annotate plot: The plot is annotated with labels and titles. This makes the plot easier to understand.

DATASET:

The dataset comprises trajectory data from satellites, encompassing 186,299 entries, each characterized by 22 distinct parameters. These parameters encompass various aspects of satellite behavior and performance during their missions.

The dataset captures an array of parameters, including analog and digital readings. The analog readings involve parameters such as pitch and roll analog rates for both DTG-1 and DTG-2, fine rates for pitch and yaw, as well as temperature measurements for different components like electronics and thrust systems. These analog parameters offer insights into the satellite's orientation and thermal dynamics during operation.

Additionally, the dataset includes digital parameters that provide information on the status of different systems. This encompasses synchronization status for DTG-2, operational statuses for different components (DTG-1, DTG-2), power supply status, and underwater acoustic positioning system status (DTG-3 USBL\_STS). The presence of digital parameters helps in assessing the health and functionality of various satellite systems.

Incorporating a rich set of analog and digital measurements, this dataset serves as a valuable resource for studying and analyzing the trajectories, behaviors, and operational conditions of satellites. It holds the potential to enable anomaly detection, predictive maintenance, and further insights into the performance of these spaceborne vehicles

Model :

The Local Outlier Factor (LOF) is a popular machine learning algorithm used for anomaly detection in datasets. It quantifies the local density deviation of a data point with respect to its neighbors, allowing it to identify outliers based on their relative isolation from the rest of the data.

LOF operates under the assumption that outliers are typically less densely surrounded by similar data points compared to the normal instances. To achieve this, LOF calculates a score for each data point, representing how isolated it is from its neighbors. Here's a brief model description of how LOF works:

1. Nearest Neighbor Calculation: For each data point in the dataset, LOF determines a specified number of nearest neighbors based on a distance metric, such as Euclidean distance. These neighbors are used to assess the local density around the data point.

2. Local Reachability Density (LRD) Calculation: The LRD of a data point is computed by taking the inverse of the average distance between the data point and its k nearest neighbors. This metric quantifies the average reachability distance of a data point to its neighbors, indicating the local density around it.

3. Local Outlier Factor Calculation: The LOF score for a data point is computed by comparing its LRD to the LRD of its neighbors. If a point has a significantly lower LRD than its neighbors, it suggests that the point is less densely surrounded and is potentially an outlier. The LOF score is essentially a measure of how isolated a data point is in relation to its neighbors.

4. Anomaly Detection: Data points with high LOF scores are considered anomalies, as they have a substantially lower density compared to their neighbors. The LOF algorithm doesn't rely on predefined thresholds but rather assesses the deviation of each point based on its neighborhood context.

5.Scalability and Parameters: LOF's effectiveness can be influenced by parameters such as the number of neighbors considered and the distance metric used. Adjusting these parameters can impact the algorithm's sensitivity to different types of outliers and the noise level in the data.

The Local Outlier Factor algorithm is particularly useful in scenarios where anomalies are not defined by a specific distribution and when global approaches like clustering might not be appropriate. It excels in identifying outliers in datasets with varying local densities and can provide valuable insights into the irregularities within the data.  
  
  
Results:

The anomaly id detected at : 2018-05-10 10:48:05.756000 so we can consider the starting point of the anomaly as 2018-05-10 10:47:05. The model took 6 min 28sec for the detection of anomaly from the 186299 parameters for the manual validation of anomaly the graph is plotted

