ASSIGNMENT

a) Which method— heuristic or machine learning— would you choose for developing your maneuver detection method, and why?

→ As the simplicity, ease of implementation, and lack of reliance on labeled data, the heuristic approach is the most practical choice in this case. It offers a fast, interpretable solution for detecting maneuvers based on semi-major axis (SMA) variation.

Maneuver Detection Methodology in Orbital Data

The goal of this project is to develop a heuristic approach for detecting orbital maneuvers using variations in the semi-major axis (SMA) over time. Maneuvers, such as engine burns or orientation changes, are expected to cause significant deviations in the orbital parameters. By identifying these deviations, we can detect when a maneuver occurred.

1.1 Problem Statement

We aim to detect potential maneuvers in orbital data without explicit labels for when these maneuvers occurred. The data includes `Datetime` and `SMA` values, and we use the SMA variation to detect maneuvers.

2. Data Preprocessing

2.1 Loading the Data

The dataset provided contains two key columns:

- **Datetime**: Timestamps of the observations.
- **SMA**: Semi-major axis values representing the orbital parameter of interest.

The data is loaded using pandas, and the `Datetime` column is converted into a pandas datetime format for time-based operations. The data is sorted by `Datetime` to ensure that subsequent operations on consecutive time points are accurate.

#code

```
data['Datetime'] = pd.to_datetime(data['Datetime'])  # Convert
Datetime column to pandas datetime
```

```
data = data.sort_values(by='Datetime')  # Ensure the data is sorted by time
```

2.2 Missing Data Handling

We assumed the dataset did not contain any missing values. However, in real-world scenarios, it's important to check for missing values and handle them accordingly.

3. Feature Extraction

To identify maneuvers, we calculate the **difference in the SMA** between consecutive data points. The intuition is that significant changes in the SMA indicate a maneuver. This is implemented as follows:

#code

data['SMA_diff'] = data['SMA'].diff()

- **SMA_diff**: Represents the change in SMA between consecutive observations.

  The first row of the `SMA_diff` column will be `NaN` (or 0) because there is no previous data point to calculate the difference.

4. Maneuver Detection (Heuristic Approach)

4.1 Thresholding

We assume that a significant change in SMA indicates a maneuver. To detect such significant changes, we use a threshold based on the **standard deviation of SMA differences**. Specifically, maneuvers are detected when the absolute SMA difference exceeds **3 standard deviations** from the mean difference.

#code

```
threshold = 3 * data['SMA_diff'].std()  # 3 standard deviations from the mean change

data['Maneuver'] = np.where(data['SMA_diff'].abs() > threshold, 1, 0)
```

- **Threshold**: We use 3 standard deviations to ensure that only significant outliers (extreme changes) in SMA are flagged as maneuvers.
- **Maneuver**: A binary column where `1` indicates a detected maneuver and `0` indicates no maneuver.

4.2 Maneuver Detection

Once the threshold is applied, we extract the rows where a maneuver was detected:

#code

```
maneuvers = data[data['Maneuver'] == 1]
```

This gives us a subset of the data where maneuvers were detected based on significant SMA changes.

5. Results Visualization

To visualize the results, we plot the SMA values over time and mark the detected maneuvers with red markers. This allows for easy inspection of when and where the maneuvers occurred.

#code

```
plt.figure(figsize=(12, 6))

plt.plot(data['Datetime'], data['SMA'], label='SMA',
color='blue')

plt.scatter(maneuvers['Datetime'], maneuvers['SMA'],
color='red', label='Detected Maneuver', marker='o')

plt.title('SMA with Detected Maneuvers')

plt.xlabel('DateTime')

plt.ylabel('SMA')

plt.legend()

plt.grid(True)
```

```
plt.show()
```

This plot helps in understanding how the SMA changes over time and where the detected maneuvers are located.

6. Output and Storage

The detected maneuvers, along with their corresponding `Datetime` and `SMA` values, are saved to a new CSV file for further analysis or reporting.

#code

```
output_file_path = '/mnt/data/detected_maneuvers.csv'

maneuvers.to_csv(output_file_path, index=False)
```

This allows the results to be easily accessed and used for further analysis.

7. Assumptions

Several assumptions were made during the development of this methodology:

1. **SMA Difference Threshold**: A change in SMA greater than 3 standard deviations is assumed to be indicative of a maneuver.
2. **No Noise Handling**: The data is assumed to be relatively clean and free from noise.
3. **SMA as the Primary Feature**: The methodology assumes that SMA differences alone are sufficient for detecting maneuvers.
4. **Constant Threshold**: The same threshold is used across all data points.

---

8. Summary

This methodology uses a simple heuristic approach to detect maneuvers in orbital data based on variations in the semi-major axis (SMA). By calculating the difference between consecutive SMA values and applying a threshold based on the standard deviation of these differences, we can effectively detect significant changes that are likely to correspond to orbital maneuvers. The methodology is computationally efficient and easy to implement, making it a good first step in analyzing orbital data.