Team name: Team X

Participants name: Shashank Yadav, Kesav Patneedi, Sidesh Kumar Patra

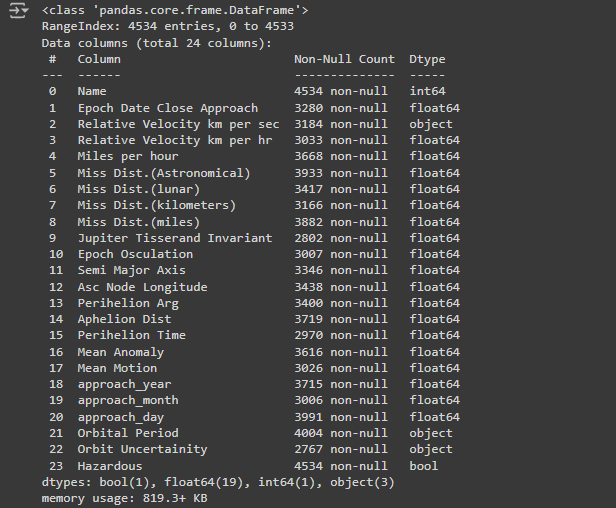
17 October 2024

Link of complete file :[Ipnyb file](https://colab.research.google.com/drive/1ZhRAjwoXtKRZ_RP8qEIDfZKaTl5Eo3TI?usp=sharing)

Data analysis report

# **1. Exploratory Data Analysis (EDA)**

## 1.1 data inspection [link of snippet](https://colab.research.google.com/drive/1ZhRAjwoXtKRZ_RP8qEIDfZKaTl5Eo3TI?authuser=1#scrollTo=nEl6EZuoM7_j&line=35&uniqifier=1)

* We have used panda's attributes info() , describe() , isnull() etc to inspect the data

### Data Overview

* **Total Rows:** 4534 entries.
* **Total Columns:** 24 columns.

### Missing Values

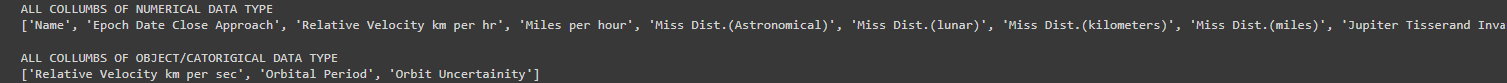
Several features have missing values, with the most significant being:

* **Epoch Date Close Approach:** 1254 missing values
* **Relative Velocity km per sec:** 1350 missing values
* **Miss Distances (Astronomical, Lunar, Kilometers, Miles):** Each has a few hundred missing entries.
* **Jupiter Tisserand Invariant, Epoch Osculation, Semi Major Axis:** Over 1000 missing values.



### Data Types

* **Numerical Columns:** 19 features (e.g., Miss Distances, Velocity, Orbital Parameters).
* **Categorical Columns:** 3 features:
  + **Relative Velocity km per sec (ordinal)** (values such as "Very Slow," "Slow").
  + **Orbital Period** and **Orbit Uncertainty** are stored as objects.

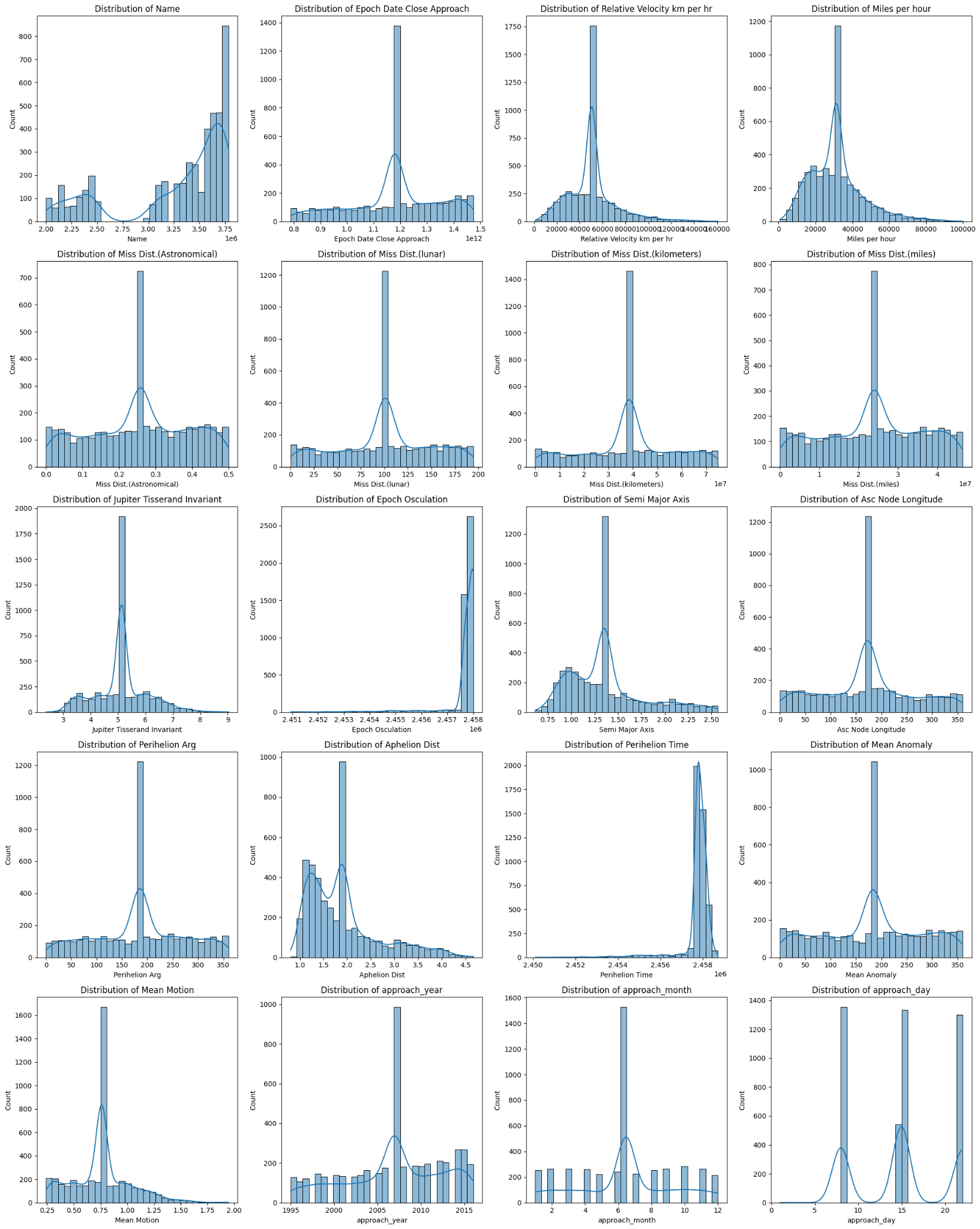


**Imputation:**

* Missing values in numerical columns have been replaced with their respective mean values.
* Categorical columns were imputed with their most frequent category (mode).

## 1.2 Statistical Inference

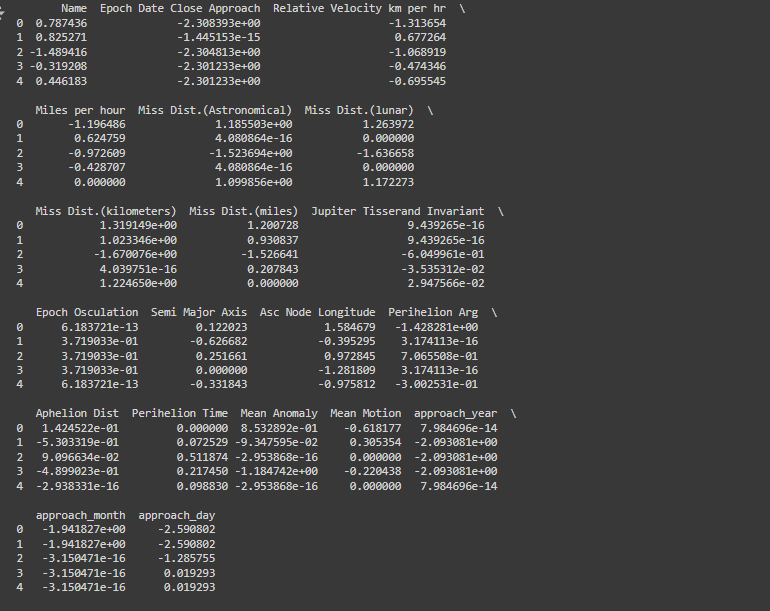
* Histograms show that some features have skewed distributions, indicating that scaling or normalization might be necessary for certain columns and check the imbalance in the data .[link of snippet](https://colab.research.google.com/drive/1ZhRAjwoXtKRZ_RP8qEIDfZKaTl5Eo3TI?authuser=1#scrollTo=dVZAawR5O0KR&line=7&uniqifier=1)

****

**Class Imbalance:**

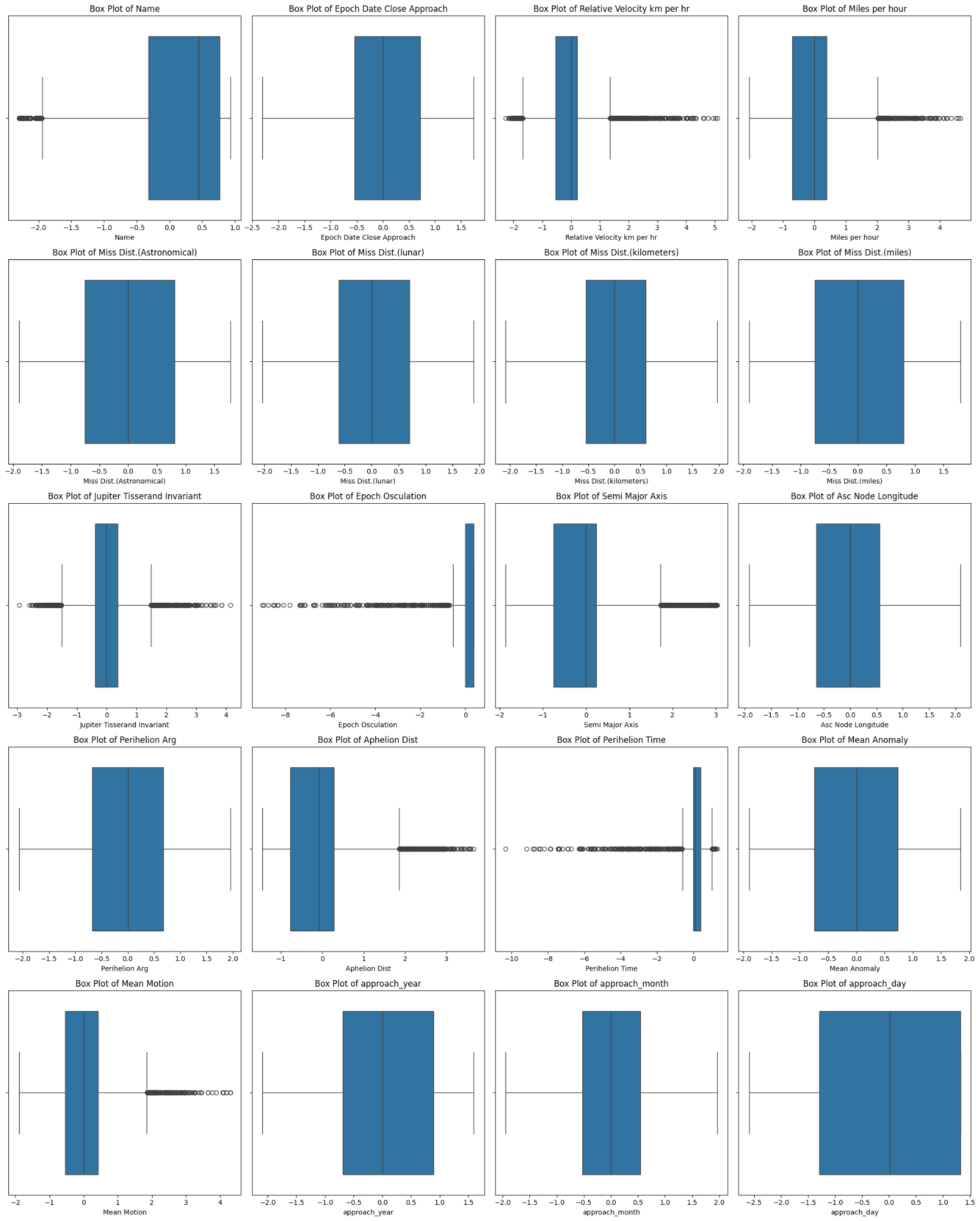
* There is a **class imbalance** in the dataset:
  + **Non-hazardous asteroids (False):** 83.77%
  + **Hazardous asteroids (True):** 16.23%

**normalization:**[**link of snippet**](https://colab.research.google.com/drive/1ZhRAjwoXtKRZ_RP8qEIDfZKaTl5Eo3TI?authuser=1#scrollTo=hdeztz2wZz4p&line=7&uniqifier=1)

* StandardScaler scales the features to have a mean of 0 and a standard deviation of 1. We do the normalization in every numerical feature in the data set

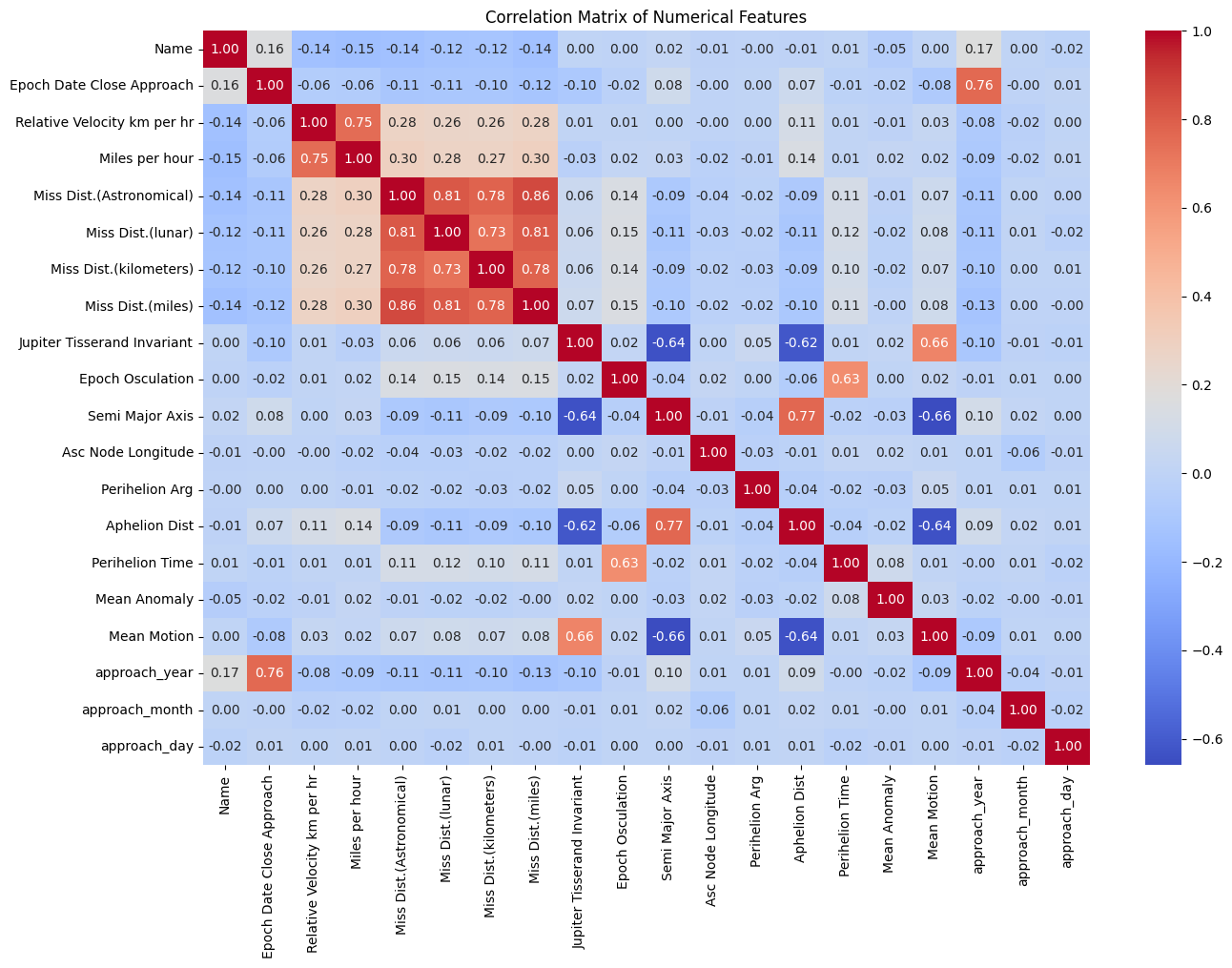
**Outlier Detection:**[**link of snippet**](https://colab.research.google.com/drive/1ZhRAjwoXtKRZ_RP8qEIDfZKaTl5Eo3TI?authuser=1#scrollTo=tcwkIgOEUcIl&line=4&uniqifier=1)

* The box plots reveal several outliers in features such as **Miss Distances (Astronomical, Lunar, Kilometers, Miles)** and **Relative Velocity**. Outliers may need to be treated depending on their impact on the model's performance.



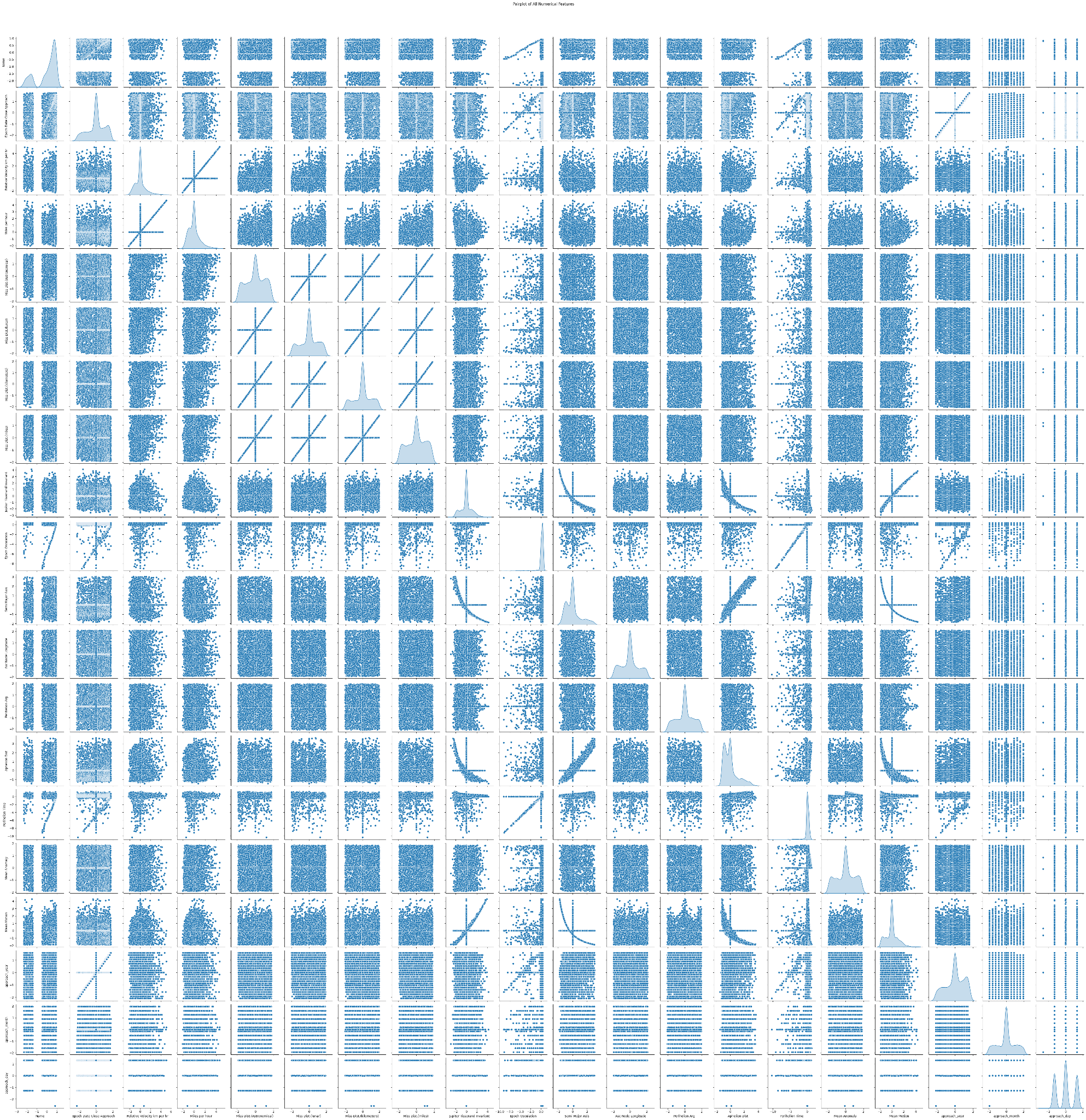
**Feature Relationships (Correlation Matrix):**[**link of snippet**](https://colab.research.google.com/drive/1ZhRAjwoXtKRZ_RP8qEIDfZKaTl5Eo3TI?authuser=1#scrollTo=tcwkIgOEUcIl&line=4&uniqifier=1)

* The correlation matrix shows the relationships between numerical features. For example, the **Miss Distance** features are highly correlated with one another, as expected, and there's a noticeable correlation between some orbital parameters.



## 1.3 Visualisation:[link of snippet](https://colab.research.google.com/drive/1ZhRAjwoXtKRZ_RP8qEIDfZKaTl5Eo3TI?authuser=1#scrollTo=5nPyB9IeafCi&line=4&uniqifier=1)

* **All Numerical Features**: This pairplot will display the relationships among all selected numerical features without the target variable, allowing you to see how they interact with each other.
* **Diagonal**: The diagonal will show the kernel density estimates of each numerical feature, providing insight into their distributions.
* **Off-diagonal plots**: will show **pairwise relationships** between two features. Typically, these are **scatter plots** that display how one feature changes relative to another.



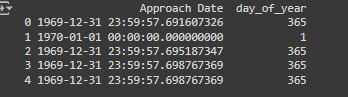
## 1.4 Tackling Class Imbalance:

* Most classifiers (e.g., **Logistic Regression**, **Random Forest**) allow you to assign **weights** to classes based on their frequency.
* This can be done by setting class\_weight='balanced', which automatically adjusts for imbalance by assigning a higher weight to the minority class.

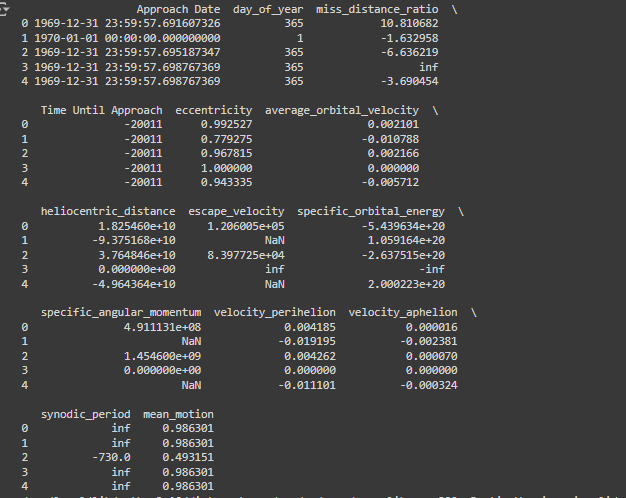
# **2.Numerical Interpretation and Mathematical Analysis :**

## 2.1 feature engineering :

* Converting in datatime formate: using data time library [link of snippet](https://colab.research.google.com/drive/1ZhRAjwoXtKRZ_RP8qEIDfZKaTl5Eo3TI?authuser=1#scrollTo=axGrPsLOhQ0R&line=5&uniqifier=1)

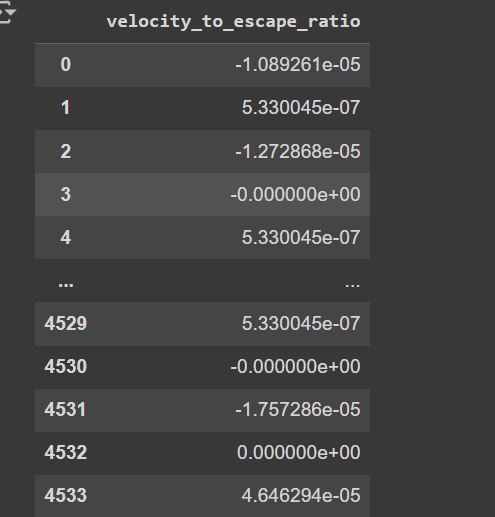


* Calculating required values using the given formulas [link of snippet](https://colab.research.google.com/drive/1ZhRAjwoXtKRZ_RP8qEIDfZKaTl5Eo3TI?authuser=1#scrollTo=WrzCWfc5fCA7&line=7&uniqifier=1)



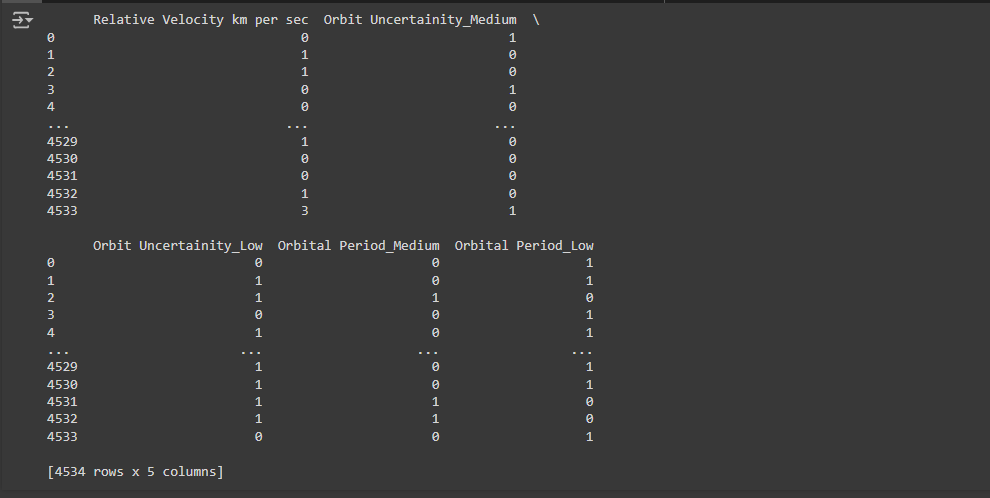
## 2.2 Additional Features:

**Velocity to Escape Velocity Ratio**:[link of snippet](https://colab.research.google.com/drive/1ZhRAjwoXtKRZ_RP8qEIDfZKaTl5Eo3TI?authuser=1#scrollTo=TkNPZmCEsCKu&line=2&uniqifier=1)

* **Feature**: The ratio of the asteroid's relative velocity to its escape velocity.
* **info**: This ratio can indicate how likely an asteroid is to escape Earth's gravity in case of a close approach.
* **formula**= Relative\_Velocity\_km\_per\_hr / escape\_velocity

# **3. Handling Binned Values:** [**link of snippet**](https://colab.research.google.com/drive/1ZhRAjwoXtKRZ_RP8qEIDfZKaTl5Eo3TI?authuser=1#scrollTo=eX4NYpr_w2nO&line=8&uniqifier=1)

* **Ordinal Mapping**: The ordinal feature **"Relative Velocity km per sec"** is mapped to numeric values according to the specified order. Ie. (very slow = 0, slow = 1, fast = 2, very fast = 3).
* **One-Hot Encoding**: The **Orbit Uncertainty** feature is transformed into multiple binary columns. The drop\_first=True argument helps avoid the dummy variable trap



# 

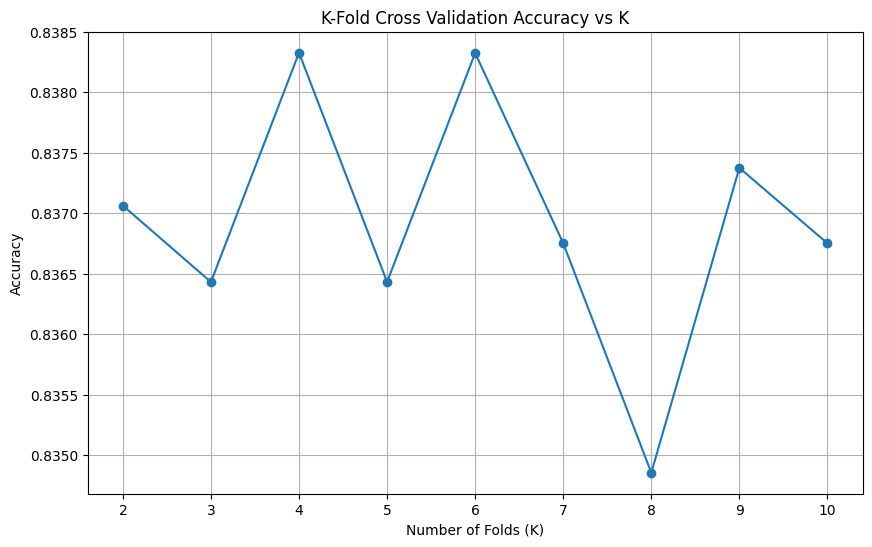
# **4. Hazardous Classification:** [**link of snippet**](https://colab.research.google.com/drive/1ZhRAjwoXtKRZ_RP8qEIDfZKaTl5Eo3TI?authuser=1#scrollTo=fZcRNK6axxA0&line=6&uniqifier=1)

### Building a Classifier:

* We'll start by building a classifier to classify asteroids as hazardous (1) or not hazardous (0). we used **Random Forest** as it’s robust and efficient for classification tasks.

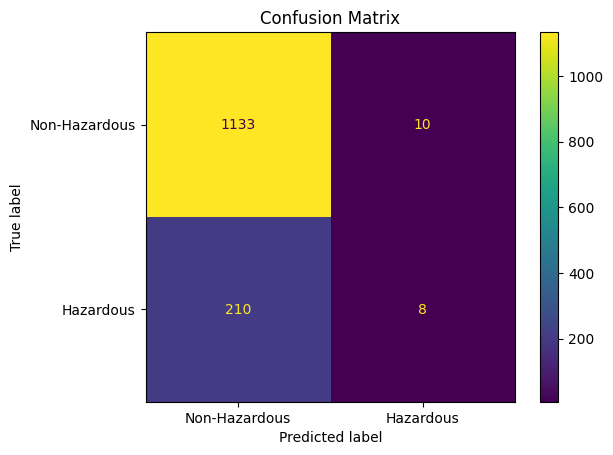
### K-Fold Cross Validation:

* We'll implement K-Fold Cross Validation, iterating over values of K from 2 to 10, and plot the accuracy and loss.



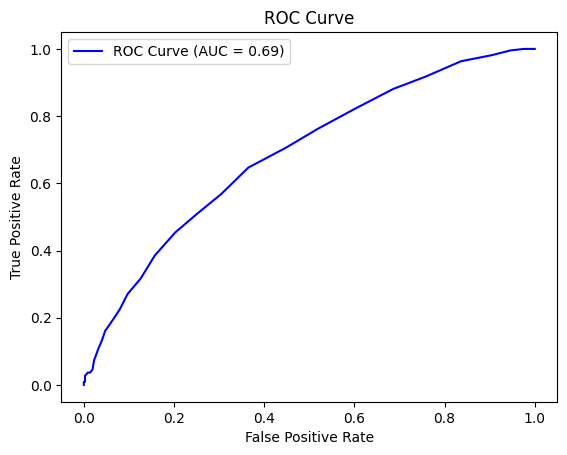
### Hyperparameter Optimization:

* For hyperparameter tuning, we can use techniques like **Grid Search** or **Randomized Search** with cross-validation.



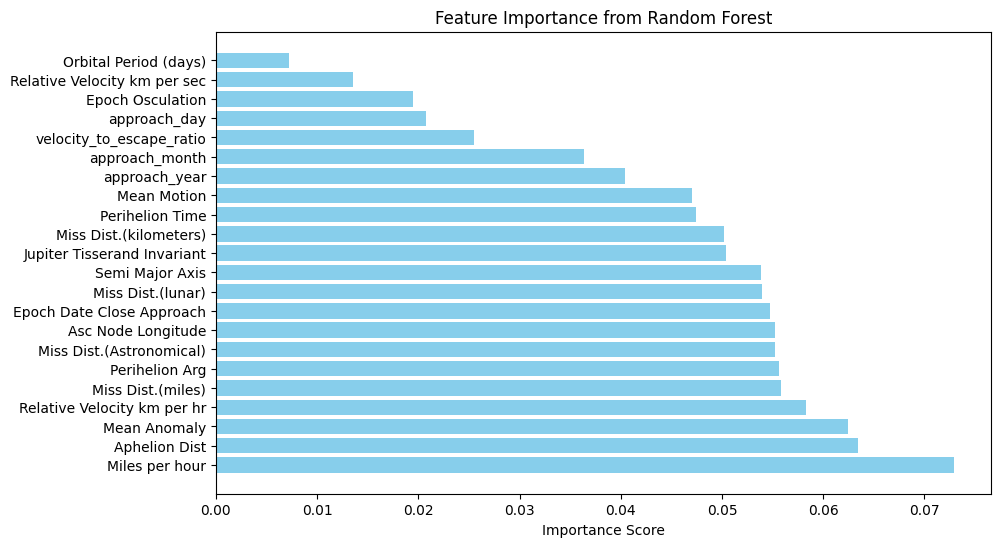
### ROC Curve and Confusion Matrix:

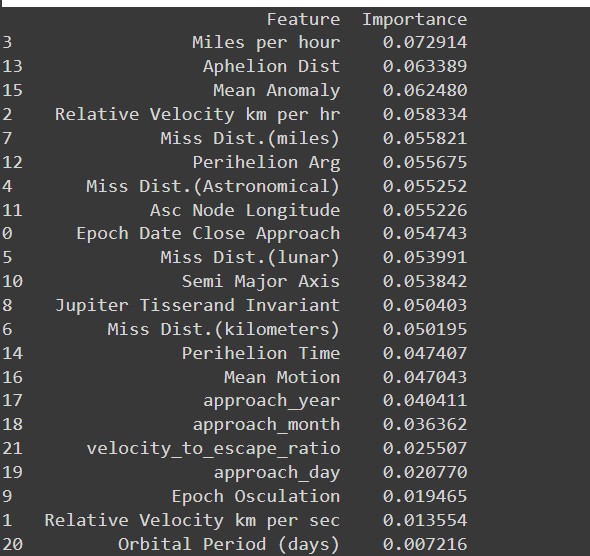
* We'll plot the ROC curve and confusion matrix to evaluate the classifier's performance.



### Feature Importance:

* Permutation importance :Random Forest classifiers have a built-in method for computing feature importance based on how much each feature contributes to reducing impurity
* Most important= miles per hour
* least important=orbital period





# **5. Anomaly Detection:** [**link of snippet**](https://colab.research.google.com/drive/1ZhRAjwoXtKRZ_RP8qEIDfZKaTl5Eo3TI?authuser=1#scrollTo=gaoa0hjI10yd&line=17&uniqifier=1)

### Isolation Forest:

* Anomaly detection is performed using the **Isolation Forest** algorithm.
* Detected anomalies are stored in a new column (anomaly\_iso\_forest).

### **Z-Score Method**:

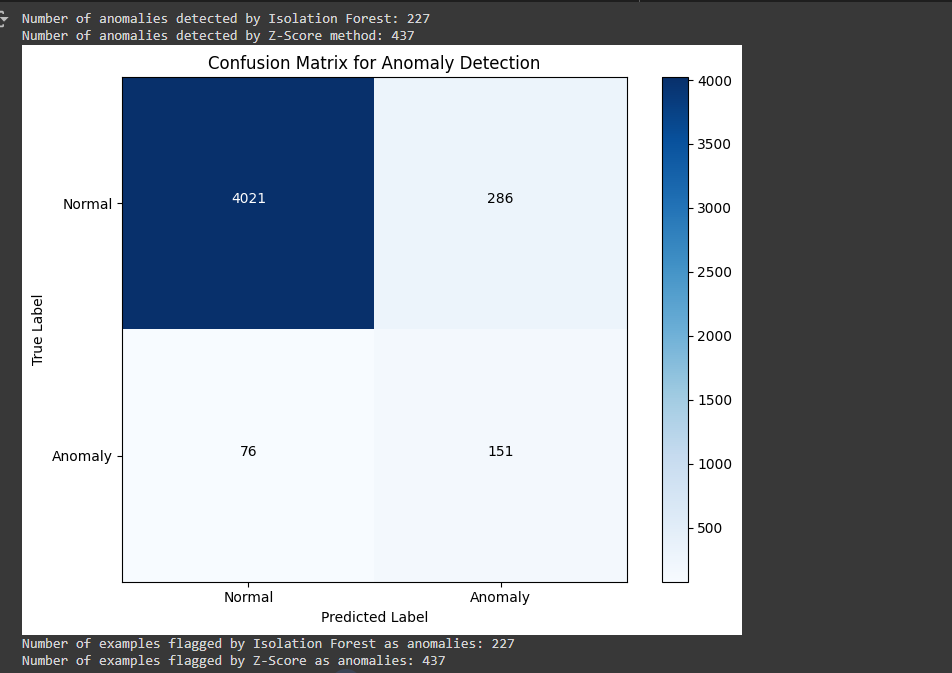
* Z-scores are calculated for each feature, and anomalies are detected based on a specified threshold.
* Detected anomalies are stored in another new column (anomaly\_zscore).

### **Counting Anomalies**:

* The number of anomalies detected by both methods is printed.

### **Confusion Matrix**:

* A confusion matrix is plotted to visualize how the two methods compare in terms of detected anomalies.



### 