Smooth Operator

Indian Institute of Information Technology, Nagpur

Overview

INTRODUCTION

Cosmic Collision-
Analysing Asteroid Risks with Data

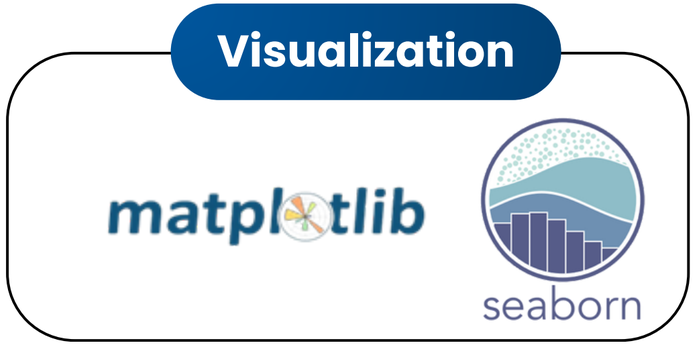
INTRODUCTION

Objective of the Analysis:

- Develop machine learning models to classify potentially hazardous asteroids

- Key Features for Analysis:
  - Orbital parameters
  - Velocity
  - Size
  - Earth proximity

- Goal: Enable automated identification and monitoring of space threats through anomaly detection



use SMOTE as it generates new synthetic data

scikit learn

Imbalanced learn

1. Exploratory Data Analysis (EDA)

Data Profiling

- Understanding data
- Identifying issues

pandas

NumPy

1.Cleaning

2.Catogorization

Numerical

Categorical

Statistical Inferences

Does this data require normalization ?

scikit learn

YES

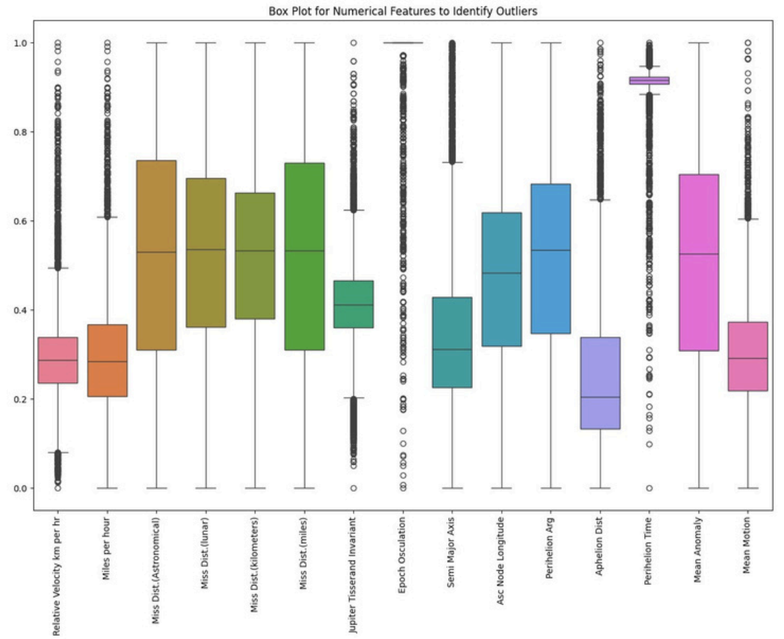
Outlier Detection

Using Z-score method

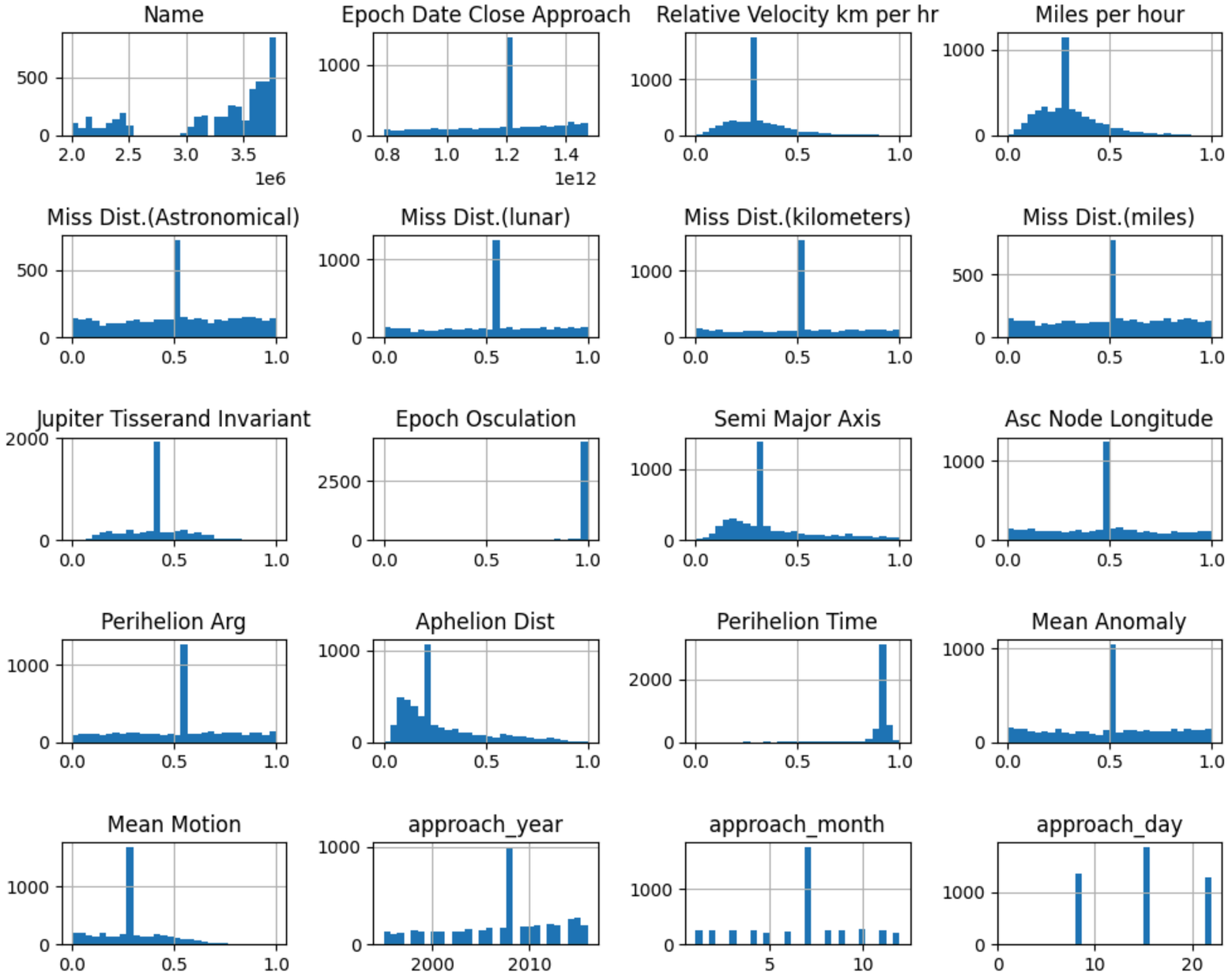
$$Z = \frac{x - \mu}{\sigma}$$

Using Box Plot

matplotlib



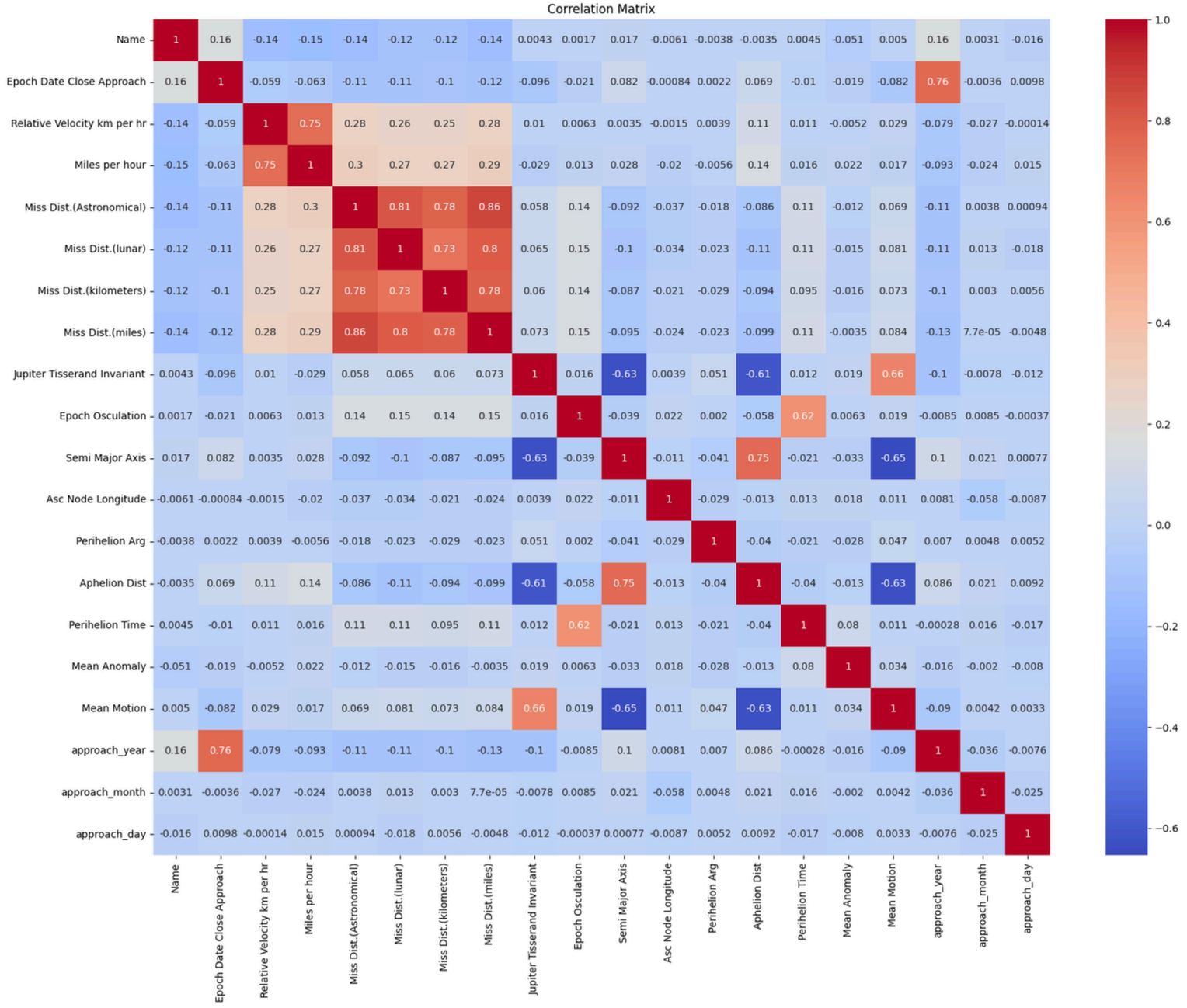
Histogram of distribution of numerical features



1. Exploratory Data Analysis (EDA)

Correlation Analysis

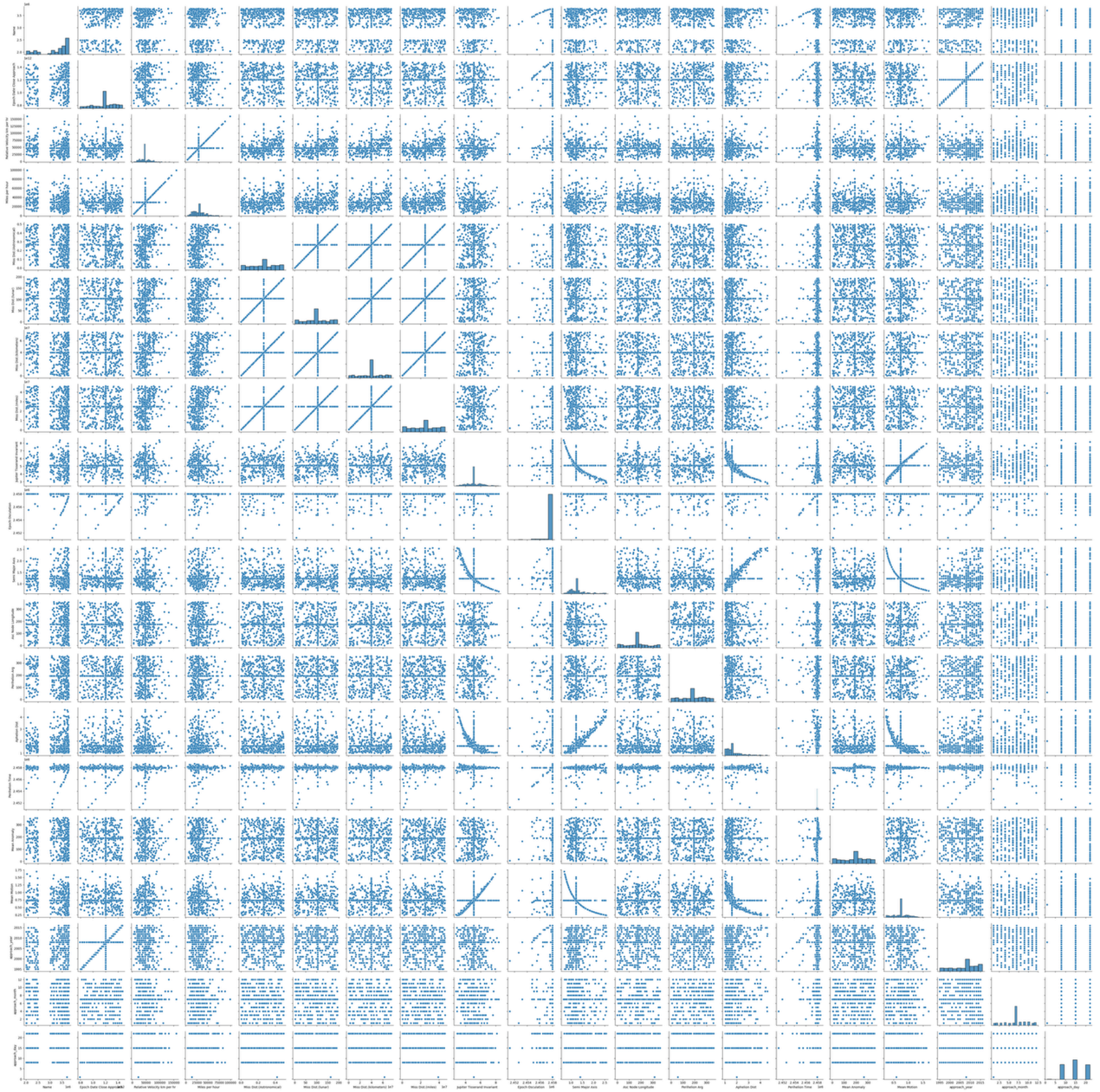
matplotlib



1. Exploratory Data Analysis (EDA)

Visualization

seaborn



Diagonal Plots

- Show the distribution of individual variables (e.g., histograms).
- Reveal patterns like skewness, concentration ranges, and outliers.

Off-Diagonal Plots

- Display scatter plots between pairs of variables.
- Help identify relationships, such as correlations or clustering patterns.

Key Inferences

- Distribution Patterns (Diagonals): Detect skewness or the need for normalization.
- Relationships (Off-Diagonals): Spot correlations or redundant features worth further exploration.

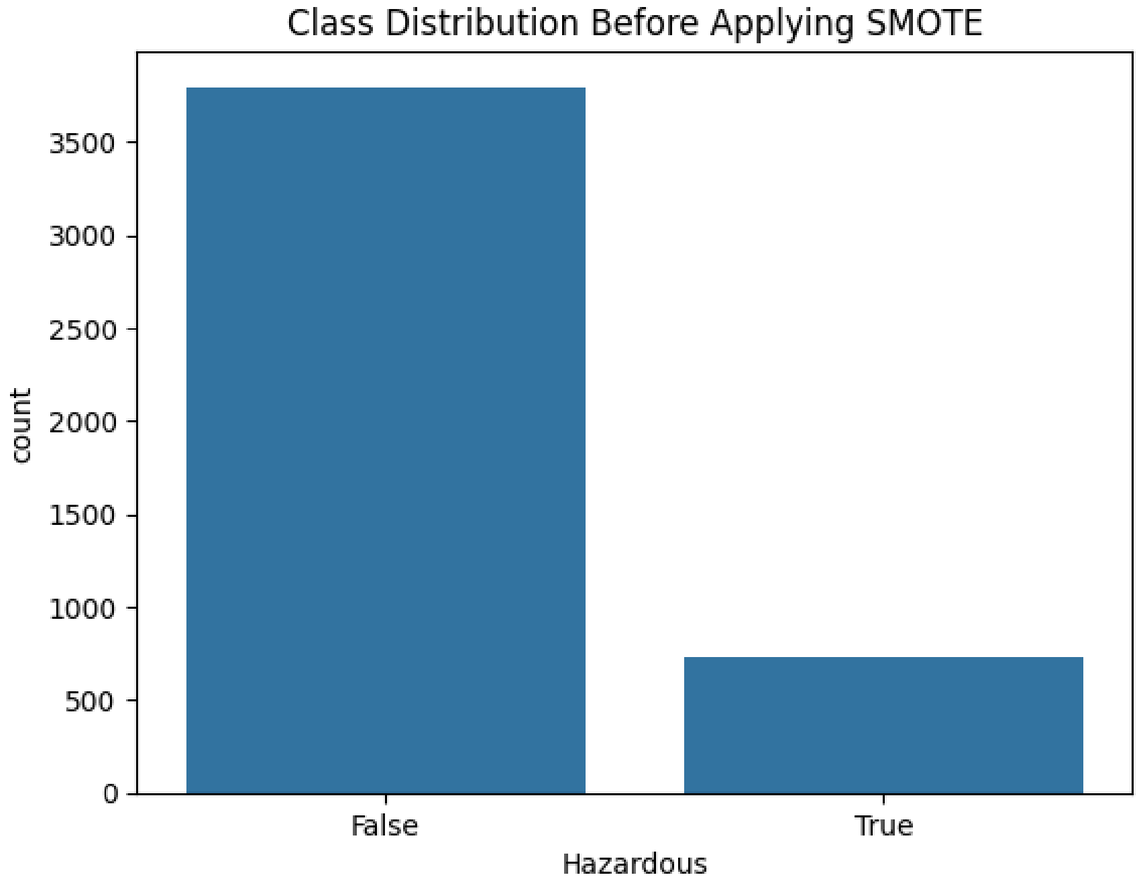
1. Exploratory Data Analysis (EDA)

Class Imbalances

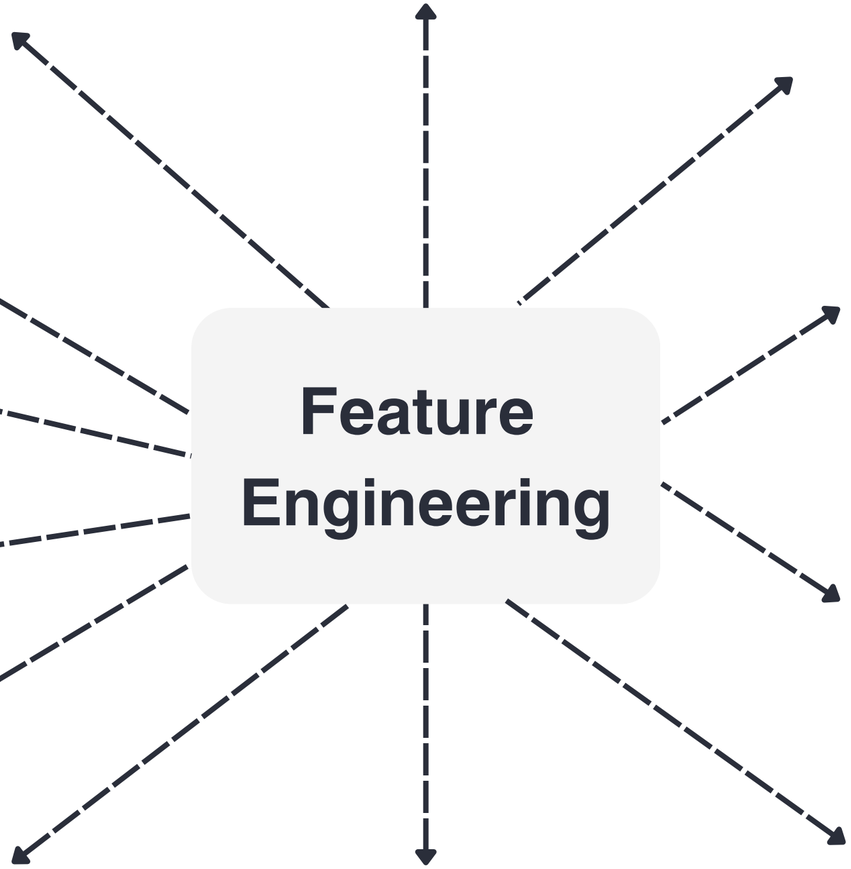
Imbalanced learn

seaborn

SMOTE: Synthetic Minority Oversampling Technique



2. Numerical Interpretation & Mathematical Analysis



1. Ratio of Miss Distance to Semi-Major Axis

$$\text{Ratio} = \frac{\text{Miss Distance}}{\text{Semi-major Axis}}$$

2. Time Until Approach

$$\text{Time Until Approach} = \text{Epoch Date Close Approach} - \text{Current Date}$$

3. Orbital Eccentricity (e)

The eccentricity e can be calculated if you have the perihelion (r_p) and aphelion (r_a) distances:

$$e = \frac{r_a - r_p}{r_a + r_p}$$

4. Average Orbital Velocity (v)

Using the semi-major axis a and the gravitational parameter $\mu = G \times M$ (where G is the gravitational constant and M is the mass of the Sun):

$$v = \sqrt{\frac{GM}{a}}$$

5. Orbital Period (T) Using Kepler's Third Law

Given the semi-major axis a in astronomical units (AU):

$$T = 2\pi\sqrt{\frac{a^3}{GM}}$$

6. Heliocentric Distance (r)

$$r = \frac{a(1-e^2)}{1 + e \cdot \cos(\theta)}$$

where θ is the true anomaly, a is the semi-major axis, and e is the eccentricity.

7. Escape Velocity (v_e)

$$v_e = \sqrt{\frac{2GM}{r}}$$

where r is the distance from the Sun.

8. Specific Orbital Energy (ε)

$$\varepsilon = -\frac{GM}{2a}$$

where a is the semi-major axis.

9. Specific Angular Momentum (h)

$$h = \sqrt{GMa(1-e^2)}$$

10. Velocity at Perihelion (v_p) and Aphelion (v_a)

$$v_p = \sqrt{\frac{GM(1+e)}{a(1-e)}}$$

$$v_a = \sqrt{\frac{GM(1-e)}{a(1+e)}}$$

12. Synodic Period (S)

For two objects orbiting the Sun with orbital periods T_1 and T_2:

$$S = \frac{1}{\left|\frac{1}{T_1} - \frac{1}{T_2}\right|}$$

13. Mean Motion (n)

$$n = \frac{2\pi}{T}$$

where T is the orbital period.

2. Numerical Interpretation & Mathematical Analysis

Additional Features

1. Time Until Next Close Approach (Cumulative Feature):

Instead of just calculating "Time Until Approach", create a feature that calculates the time difference between successive close approaches, which could provide insight into periodicity and clustering behavior of the objects.

	Name	next_approach_date	time_until_approach_days
0	3703080	2008-01-01	-6137
1	3723955	1995-01-01	-10885
2	2446862	1995-07-08	-10697
3	3092506	1995-07-15	-10690
4	3514799	2008-07-15	-5941

2. Energy Loss in Atmosphere of earth of asteroid:

For objects passing through Earth's atmosphere, a feature estimating potential energy loss or disintegration could provide insights into the survivability of impact (assuming typical asteroid density) because we don’t have it in dataset.

Final data preview with energy loss (GJ):

	Name	assumed_mass_kg	velocity_m_s	energy_loss_gj
0	3703080	1.308997e+11	3000	498.335135
1	3723955	1.308997e+11	1000	55.370571
2	2446862	1.308997e+11	1000	55.370571
3	3092506	1.308997e+11	3000	498.335135
4	3514799	1.308997e+11	3000	498.335135

3. Handling Binned Values

Ordinal Relationship

using Ordinal Encoder

Relative Velocity km per sec

Very Slow	0
Slow	1
Fast	2
Very Fast	3

Orbit Uncertainity & Orbit Period

Low	0
Medium	1
High	2

Nominal Relationship

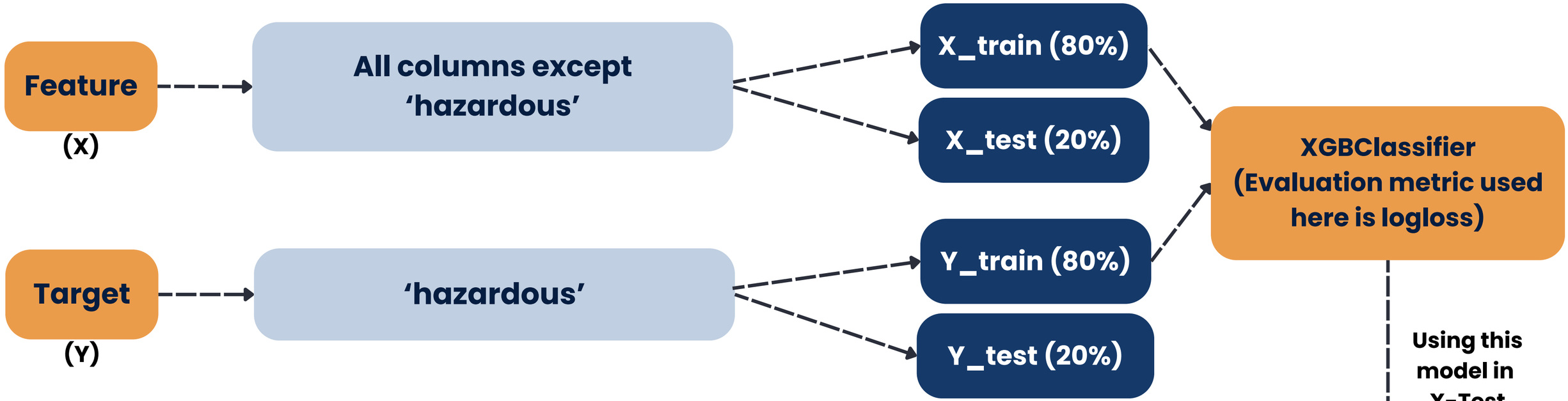
using Label Encoder

Hazardous

True	0
False	1

4. Hazardous Classification

- **Build a robust and efficient classifier to classify asteroids as Hazardous or Not Hazardous**




```
Confusion Matrix:
 [[743  25]
 [ 99  40]]
Accuracy Score: 0.8632855
```

```
Classification Report:
              precision    recall  f1-score   support

         0.0       0.88      0.97      0.92       768
         1.0       0.62      0.29      0.39       139

    accuracy                           0.86       907
   macro avg       0.75      0.63      0.66       907
weighted avg       0.84      0.86      0.84       907
```

4. Hazardous Classification

- **Implement K-Fold Cross Validation for training. Train the dataset for all values of K from 2 to 10. Plot the loss and accuracy versus epochs for these K values.**

Feature (X) → **All columns except 'hazardous'**

Target (Y) → **'hazardous'**

X_train (rest of the dataset)

X_test (kth split of the datset)

Y_train (rest of the dataset)

Y_test (kth split of the dataset)

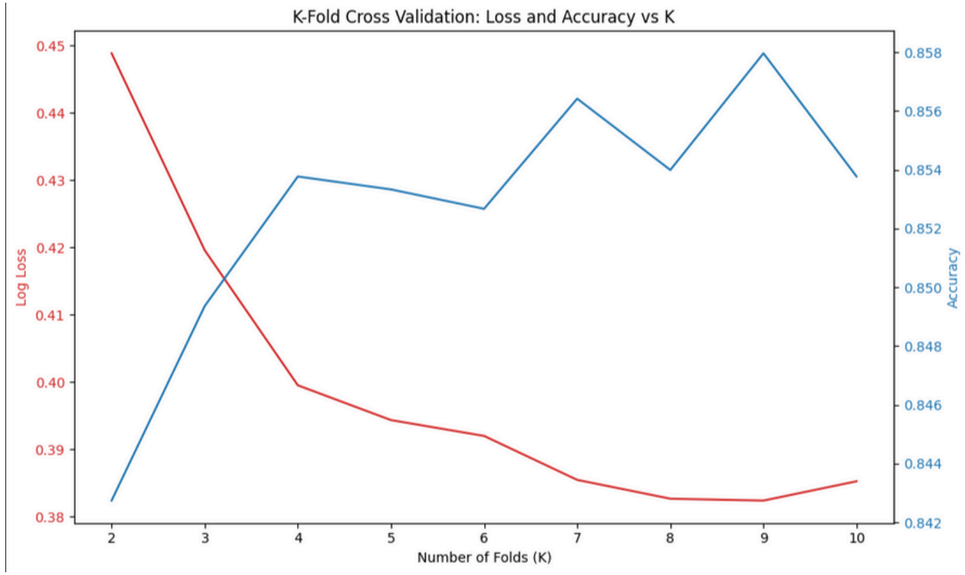
XGBClassifier (Evaluation metric used here is logloss)

Using this model in X-Test dataset we get

y_pred

After k iteration we take average of all y_pred

Checking our prediction model with actual y_test dataset we get



4. Hazardous Classification

- **Optimise all the hyperparameters used in the classifier by selecting an appropriate optimisation method**

User Defined Parameters

```
'n_estimators': [100, 200, 300, 400, 500],
'learning_rate': [0.001, 0.01, 0.1, 0.3],
'max_depth': [3, 5, 7, 9],
'subsample': [0.6, 0.8, 1.0],
'colsample_bytree': [0.6, 0.8, 1.0],
'gamma': [0, 0.1, 0.2, 0.3],
'min_child_weight': [1, 3, 5],
'reg_alpha': [0, 0.01, 0.1, 1],
'reg_lambda': [0.01, 0.1, 1],
```

applying randomized search to get 100 different combination of parameters to k fold

kfold of 5 splits

after all the iterations we can get the best hyperparameter combination

Best Hyperparameters:
{'subsample': 0.8,
'reg_lambda': 0.01,
'reg_alpha': 0.1,
'n_estimators': 400,
'min_child_weight': 1,
'max_depth': 5,
'learning_rate': 0.1,
'gamma': 0.2,
'colsample_bytree': 1.0}

Best Accuracy: 0.859063063589301 7
Model Log Loss: 0.03463977986898241

4. Hazardous Classification

- **Plot the ROC curve and Confusion Matrix to quantify the performance of your classifier.**

Feature → **All columns except 'hazardous'** → **X_train (70%)** / **X_test (30%)**

(X)

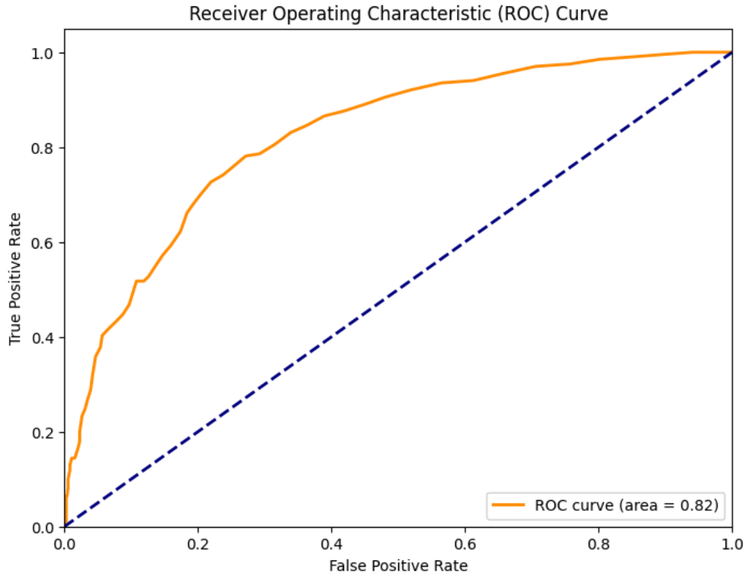
Target → **'hazardous'** → **Y_train (70%)** / **Y_test (30%)**

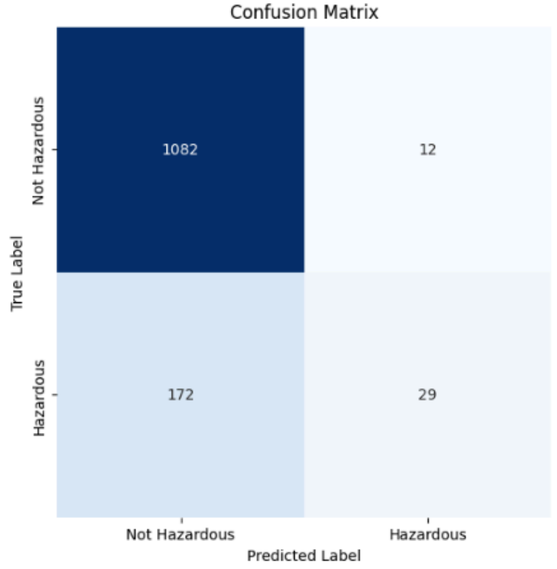
(Y)

Random Forest Classifier

tpr **fpr**

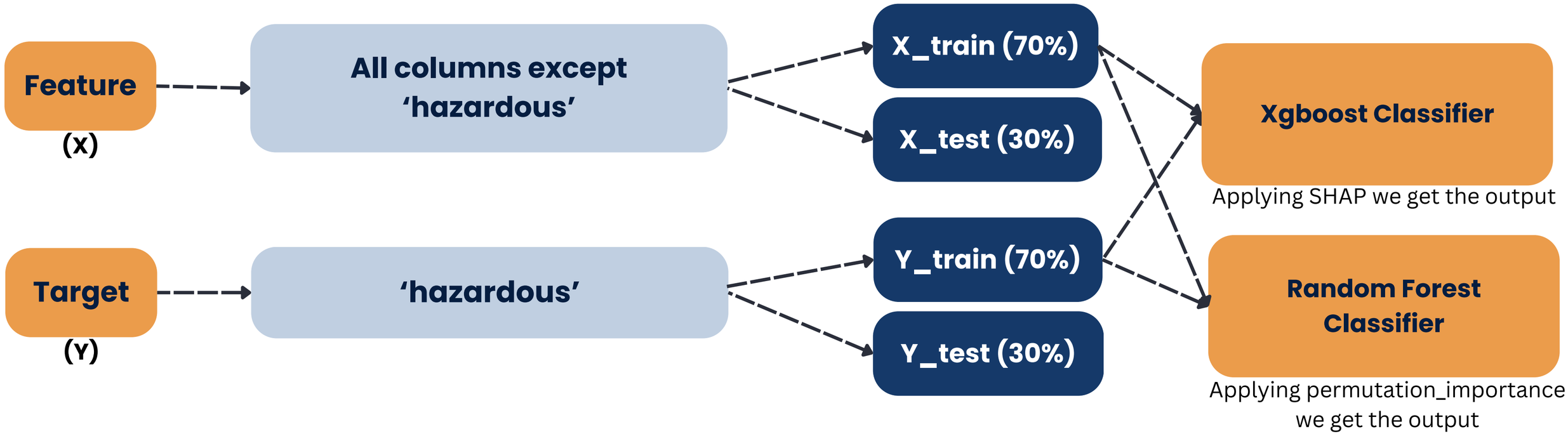
plotting graph of tpr and fpr we get ROC curve and confusion matrix



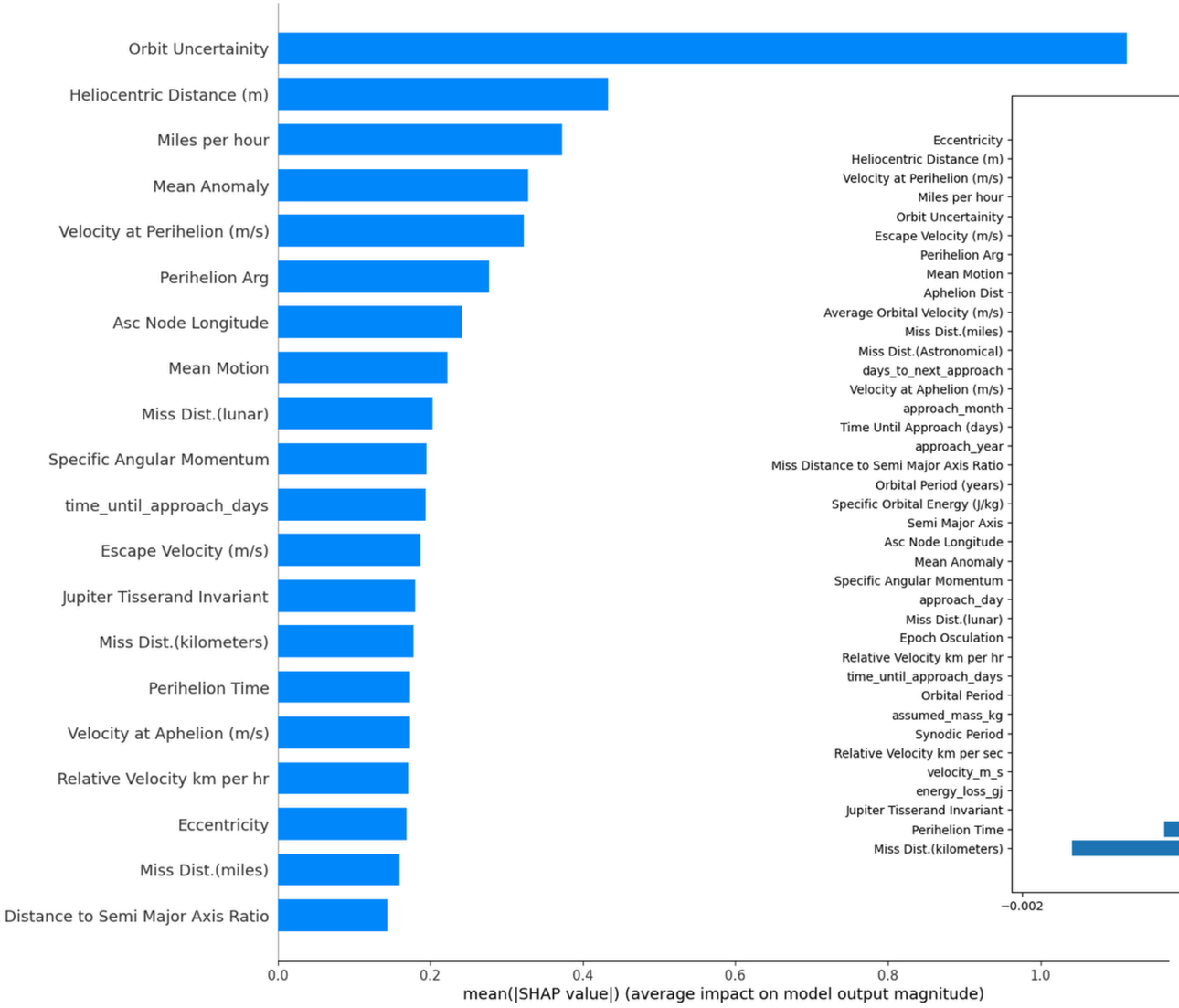


4. Hazardous Classification

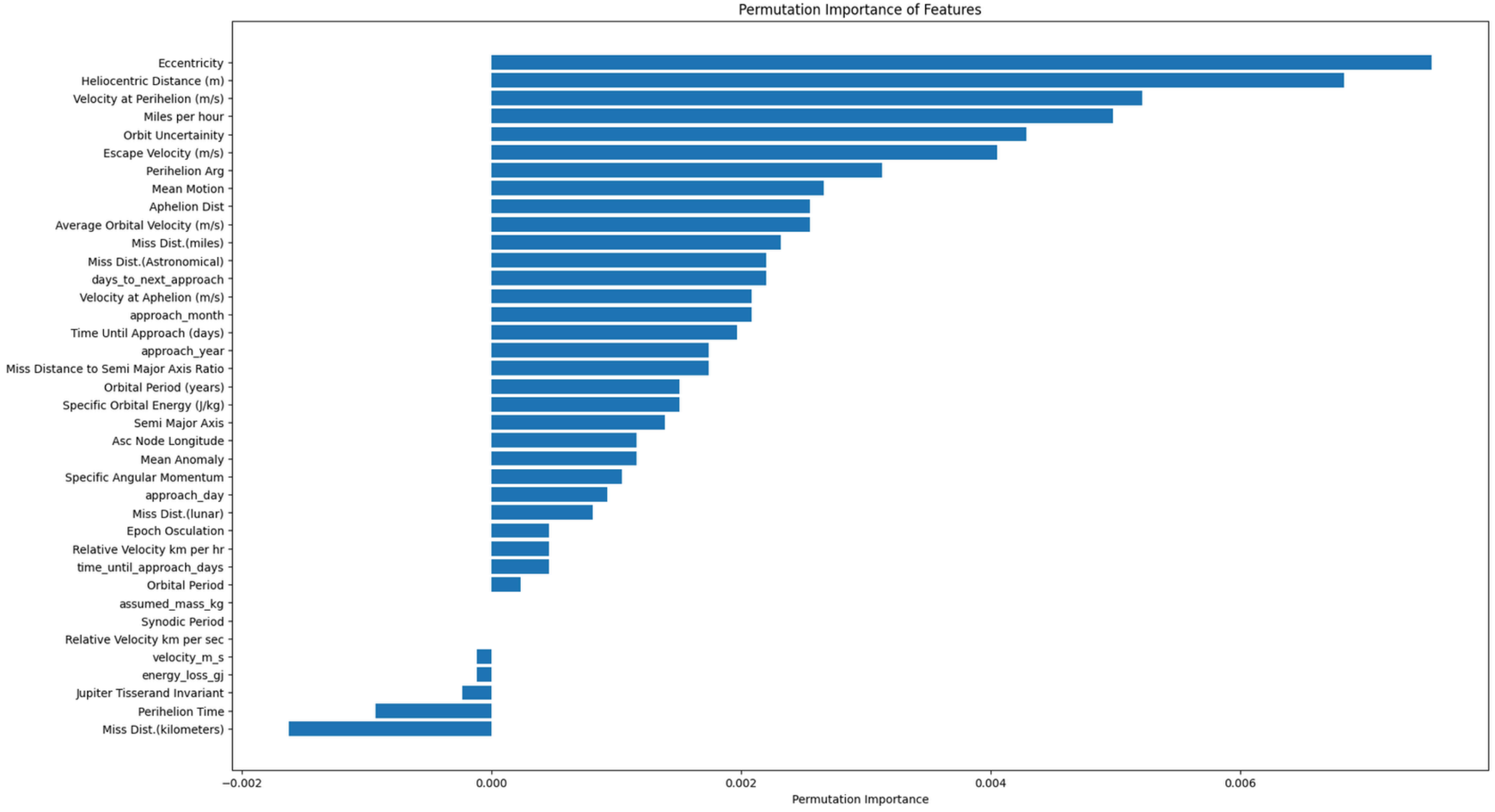
- **Use SHAP Values, Permutation Importance, or Partial Dependence Plots to list the most and least useful features**



SHAP



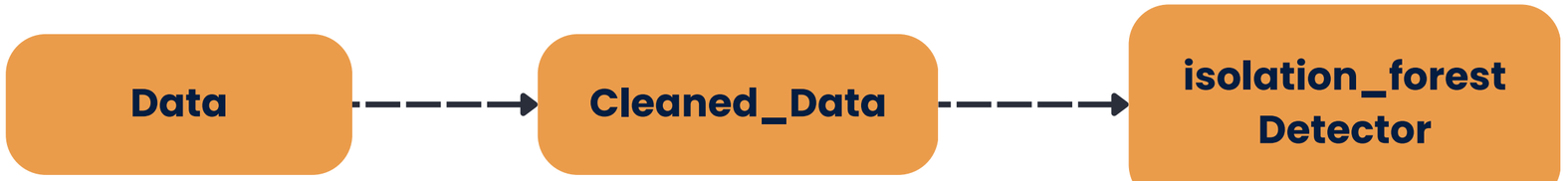
Permutation Importance



5. Anomaly Detection

unsupervised models

- **Perform anomaly detection using: (i) Any inbuilt library of your choice**



Removing unecessary columns

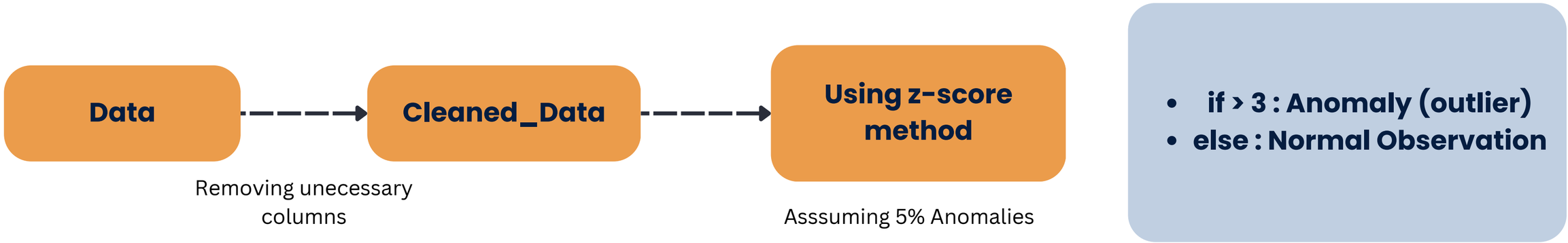
Asssuming 5% Anomalies

- **-1: Anomaly (Outlier)**
- **1: Normal Observation**

```
Number of anomalies detected by Isolation Forest: 216
   isolation_forest_anomaly
0                         1
1                         1
2                         1
3                         1
4                         1
```

5. Anomaly Detection

- **Perform anomaly detection using: (ii) Writing your own anomaly detection algorithm. Along with storing the results as a new column in the dataset and printing the number of anomalies detected**



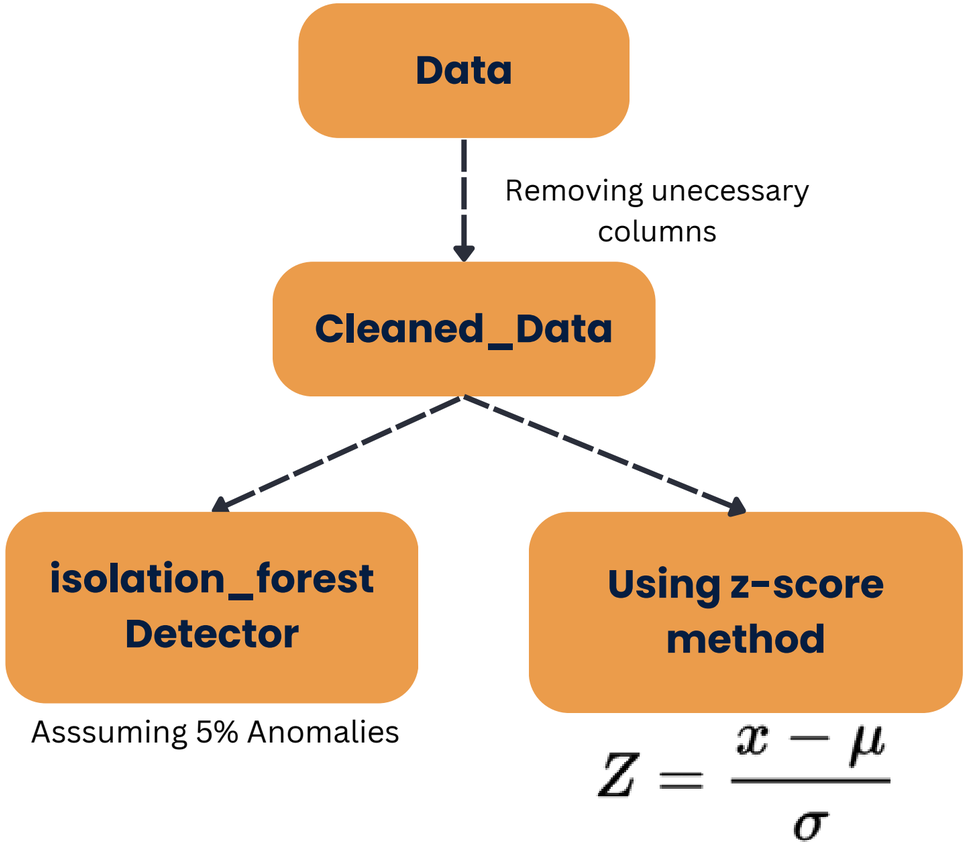
$$Z = \frac{x - \mu}{\sigma}$$

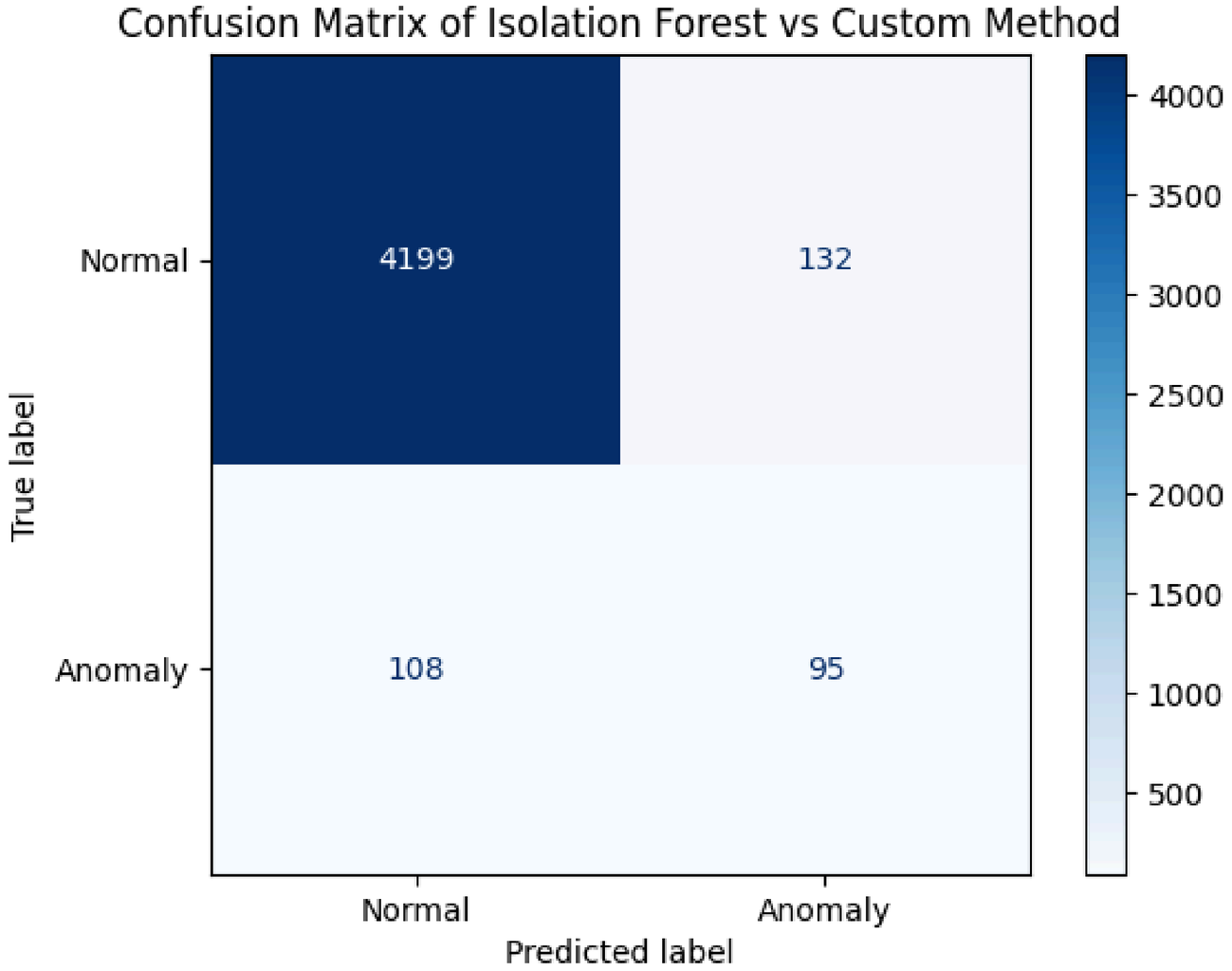
Anomalies detected by custom method: 203

5. Anomaly Detection

unsupervised models

- **Compare the results from both methods by plotting a Confusion Matrix. Print the number of examples flagged by both algorithms.**





Number of anomalies detected by Isolation Forest: 227
Anomalies detected by custom method: 203
Number of examples flagged as anomalies by both methods: 95