REPORT

Overview of the dataset:

The dataset is based on observation data of various asteroids present in our solar system with more than 4500 rows and 24 columns out of which 22 columns are features(3 categorical and the rest numerical). This dataset covers all the important and necessary features like Mean Anomaly, Ascension Node Longitude, Perihelion Argument, etc, which are needed to define an asteroid's orbit. All of the 22 said features contain some NaN values ranging from 500 -1800, on average, the whole dataset has around 25 percent values missing.

Combining the above-mentioned statistics of the NaN data and the importance of given features, it becomes highly important to deal with them effectively with logically cohesive ideas and hypotheses.

EDA, DATA-CLEANING and FEATURE ENGINEERING:

This section contains various sub-sections which will extensively talk about various statistical inferences from the data, usage of astronomical concepts and logic behind the hypothesis of dealing with various issues in the dataset.

1.)Basic Plots and inferences from them

- We have used graphs to analyze the data and find relations between columns throughout our notebook. It is used in filling up null values, creating new features, Normalizing data, verifying Model Outputs, and various other aspects.

2.) Dealing with DATES

- It had been identified that the Epoch Close Date Approach, when converted to Datetime, gave the year, month, and day present in those respective columns. So, for all missing values of year, month, and date, corresponding values from the Epoch Close Date Approach column were added, which handled over 70% of the missing values.
- Furthermore, we noticed that the dataset was ordered according to the date (oldest to newest), so using pandas, we filled in the missing values of years, months, and dates with the corresponding units it was falling between
- For example, rows 1 to 180 were for asteroids in the year 1995, so all missing values of the year column between these 2 rows will be filled with the year 1995, this was reciprocated for month and date.

- As all values were filled using this method, the year, month, and date columns were used to make a DateTime column, which was once again converted to Unix and replaced the existing Epoch Date Close Approach, therefore creating a new completely filled column.

3. Orbital Geometry Unveiled:

- The focus of this section is creating important geometrical features like the *Eccentricity* & *Inclination Angle* of the orbits and also dealing with the NaN values of the following features:-
  Semi-Major Axis, Aphelion Distance, Mean Motion, and Tisserand.
- A few geometrical and astronomical formulas to take note of are:

$$\text{Mean Motion} = \frac{360}{365.25 \cdot \text{Semi Major Axis}^{3/2}}$$

$$\text{Eccentricity} = \frac{\text{Aphelion Dist}}{\text{Semi Major Axis}} - 1$$

$$T = \frac{a_J}{a_A} \cdot 2\cos(5.2^\circ)\cos(i^\circ)\sqrt{(1-e^2)\cdot\left(\frac{a_A}{a_J}\right)}$$

- The first equation related Mean Motion and Semi-Major Axis with each other, we used the equation to fill both Mean motion and Semi-Major axis(by inversing the equation), which reduced NaN values by 73 percent and and 69.5 percent, respectively. Left-out NaN values indicate that, in some places, both these values are missing at the same time.

- We use the second equation to find a new feature called Eccentricity. Since both the Semi-Major Axis and the Aphelion Distance have missing values, new feature eccentricity will also have NaN values.
  This new Eccentricity column will then be used to calculate the Inclination Angle of the orbit via the Tisserand formula, which, yet again, will also have null values.

- Then MICE imputer algorithm was used on Tisserand, Aphelion, Mean Motion, Inclination, and Semi Major axes to fill the remaining missing values, which couldn't be filled with the formulas mentioned above. The missing value of Eccentricity was filled with the new imputed values of Aphelion Distance and Semi-Major Axis via equation 2 mentioned above.

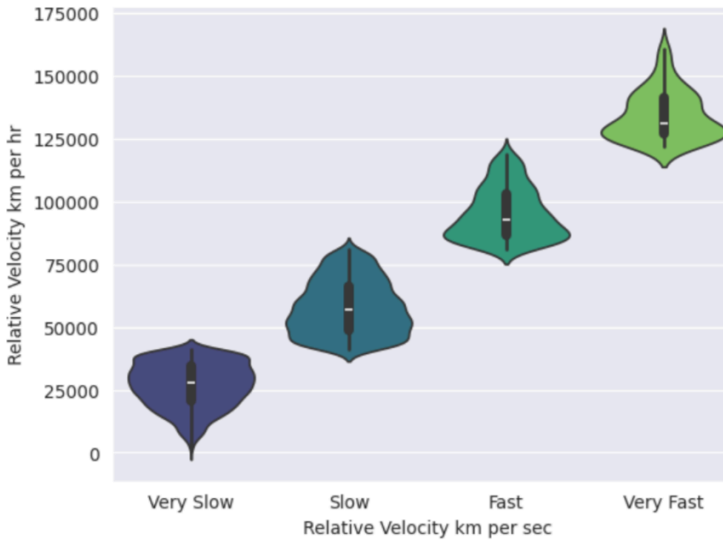
- One might think, why not create an Inclination and Eccentricity column after filling every NaN value of the columns needed to calculate them? The hypothesis behind our methodology of doing the contrary is 'CONTEXT'. MICE imputes values by fitting a very simplified regressor on the columns fed to it, Hence, to give more context on how they are related should give better-imputed values.

- To support our above hypothesis regarding ‘ CONTEXT’ , we calculated the error between the observed Tisserand Invariant values (final values after imputation in the dataset). Notably, the Tisserand formula was never used to directly replace missing values of the Tisserand Invariant itself but instead, it was used to generate a new feature. The 1,738 missing values (representing 40% of the dataset) were filled via the imputation algorithm. Hence, applying the formula to assess the error in these imputed values is justifiable, which returned an error of 0.94 percent.

4. Crafting Insights from Speed and Space:

- In this section, we address the handling of Velocity and Distance columns. Among the three velocity columns, one was categorical, while the other two were numerical but expressed in different units. For the distance, there were four numerical columns, each representing distance in different units

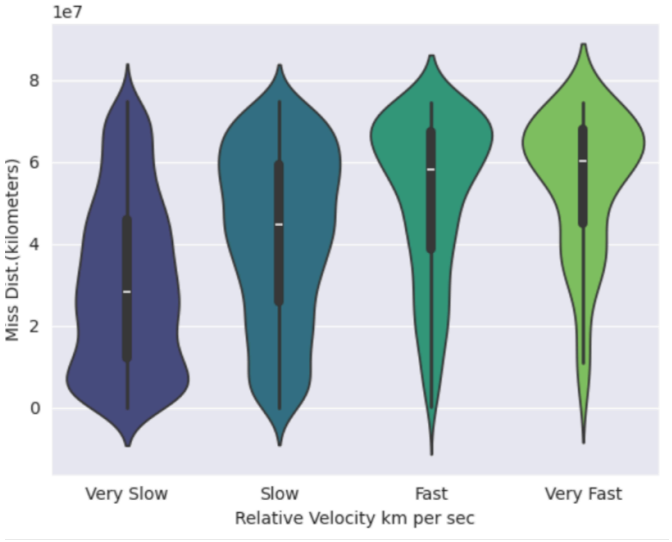
- To begin, we applied basic unit conversions, which significantly reduced the number of missing values in the numerical columns. After this, we tried drawing statistical inferences from the below given plot.



The plot clearly showed the variation in numerical velocities corresponding to different velocity categories, Hence we can use these variations to our advantage in the following manner: any row with missing categorical value had, let's say, numerical velocity < max numerical velocity of very slow category, the categorical null value will be filled by very slow and so on.
The reverse of this method can be used to fill the NaN values of numerical columns where categorical columns are not null.

- After the initial round of filling NaN values, we observed that no row had both numerical distances and categorical velocity missing simultaneously. This led us to form a hypothesis: if there is a decent statistical variation in distances corresponding

to different velocity categories, we can apply a similar statistical methodology as before to further reduce the remaining NaN values



There is a noticeable variation between lower and higher speeds, but the difference between fast and very fast speeds is much more subtle. Fortunately, this is not a significant issue since the Null values being addressed here account for less than 5.5 percent of the overall data on average. By applying the statistical method discussed earlier combined with the statistical inferences of the current plot, we effectively resolve all missing values related to distances and narrow down the categorical velocity null values further.

- For the remaining missing values, we use MICE imputer. We also average the different numerical distance and velocity columns (each of which is represented in various units)condensing them into two numerical columns for distance and velocity, respectively.

5. Time is a valuable thing!

- As the heading says focus within this section is dealing with Features that are Time-related, i.e., Epoch Osculation, Perhilion time, Orbital period, and also Mean Anomaly *(Mean Anomaly is a spatial feature but has a direct relation with the time features, hence handling it here).*

- The main relation that will be the guiding light for most of the analysis here is:

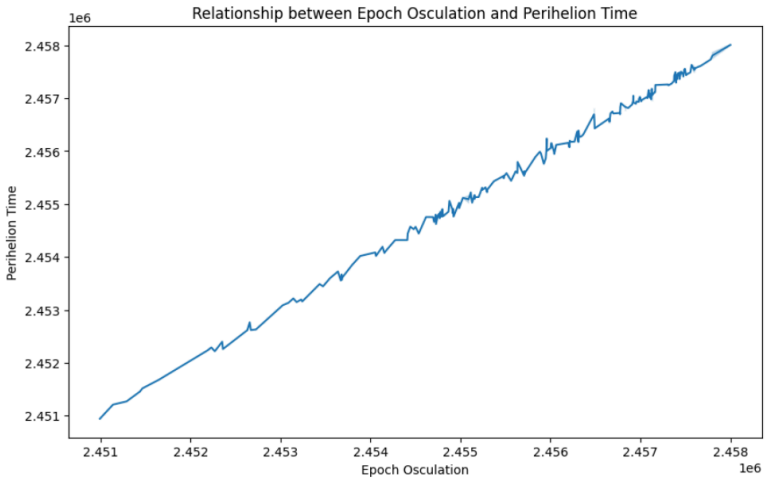
$$(\text{Perihelion Time} - \text{Epoch Osculation}) = \frac{\text{Mean Anomaly}}{\text{Mean Motion}}$$

(NOTE: Mean Motion missing values were already dealt with before. Hence main focus is the other three features).

We used the above relation to fill the null values of Perhillion Time, Epoch oscillation and Mean Anomaly.

- We used something known as Linear Interpolation to fill the rest of the values of epoch osculation, which is nothing but missing values filled with the average of neighboring valid values. This is done because of a lot of similar ranges of occurring values.

- Now, to deal with perihelion time, we need to focus on the equation mentioned above, which shows a somewhat linear relation between Perhillion Time and Epoch Osculation, but it should show variation as there is a factor of both Mean Anomaly and Mean Motion(which is not important there ratio will be constant), but to our findings, the relation was approximately linear.

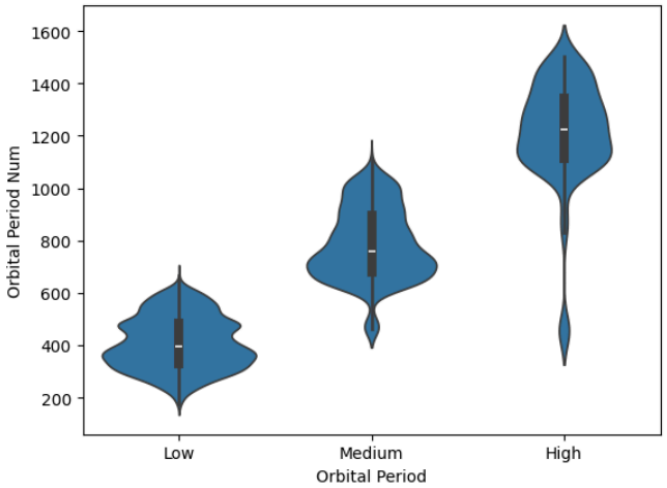


So, based on this statistical inference and combining the fact that all the missing values of the Epoch Osculation column have been filled, fitting a linear regressor to impute Perhillion time was our logical conclusion.
After filling the values of perihelion time with the regressor, we used the previously mentioned formula to fill the rest of the values of Mean Anomaly.

- Now only time column that needs to be dealt with is the Orbital Period, which is categorical in nature. We will handle it by first creating a new feature named Orbital Period Num, which is nothing but the numerical value of the orbital period.
It will be addressed with the following equation:

$$\text{Orbital Period} = \frac{360^\circ}{\text{Mean Motion}}$$

- After finding the numerical values, we again plot a violin plot to analyze the variation of numerical values with corresponding categorical values.



The above variation follows the logical and theoretical expectations, and Hence, we use the same methodology of Statistical Inference based imputation applied in Distance-Velocity analysis to find the missing values on current statistical inferences.

6. A Big Dilemma

- Since both the Perhelion Argument and Ascending Node Longitude are spatial features and part of 6 basic parameters of Kepler orbits, which are used to define every orbit of the celestial body, Hence it's impossible to fill them with some physics or geometrical equation.

- The only good way was to fit a linear regressor using all the important geometrical and spatial features to fill these values, which might or might not make this important column irrelevant.

- Irrelevancy itself can be checked, and it is discussed in the Data Preparation section, where we did feature selection to remove bad features.

7. Uncertainty: The Constant Headache

- The data of this problem statement in its rawest form was observational data with missing values(roughly 25-30 percent of the raw data was missing), but as we know, Observed Value tends to deviate from Theoretical Values by some margins in every scientific reading, and combining this with the fact that we tried to impute those 25-30 percent data with various theoretical and statistical methods clearly shows that raw data's uncertainty column (which is also completely observational and has around 40 percent data missing) becomes irrelevant.

- We cannot completely ignore uncertainty as it's grading how much the orbit we defined is uncertain, So we decided to get Uncertainty of Orbit by scratch with the relevance of the final imputed dataset.

- Before calculating Uncertainty, there is one major feature that we need to find, which is the Heliocentric Distance of the asteroid (WILL CONSIDER IT AS OBSERVED VALUE),
  Which we can approximate via using cosine law:-

$$d_{\text{sa}}^2 = 1 + d_{\text{ea}}^2 - 2 \cdot 1 \cdot d_{\text{ea}} \cdot \cos(i^\circ)$$

Where:

- $d_{\text{sa}} = \text{heliocentric distance}$
- $d_{\text{ea}} = \text{closest distance of Earth and Sun}$
- $i = \text{angle of inclination}$

- After this, we find uncertainty in velocity(v) and heliocentric distance(r), which are denoted by Δr and Δv, respectively, which is calculated using the vis viva equation and the velocity heliocentric distance we imputed in the data frame.

- Now comes the fun part: How will the uncertainty be calculated???
  We will use the uncertainty parameter defined by the Minor Planet Centre(MPC):

$$lr = \left(\Delta\tau \cdot e + 10 \cdot \frac{\Delta P}{P}\right) \cdot 3600 \cdot 3 \cdot \frac{k_o}{P}$$

Where:

- $\Delta\tau = \text{Uncertainty in Perihelion Time in days}$
- $\Delta P = \text{Uncertainty in Orbital Period in days}$
- $P = \text{Perihelion Time in Years}$
- $e = \text{Eccentricity of orbit}$
- $k_o = 0.98560766860142$

$$U = \min\left\{9, \max\left\{0, \left\lfloor 9 \cdot \frac{\log(lr)}{\log(648000)} \right\rfloor + 1\right\}\right\}$$

This gives us a categorical value ranging from 0-9.

- Now, to get Uncertainty, we need ΔP and $\Delta\tau$. The actual Parameter defined by MPC used these ΔP and $\Delta\tau$ based on multiple observations for a specific asteroid, But since we don't have multiple observations for each asteroid, the uncertainty in P and τ has to be derived from Δr and Δv, which we did in the following way:

$V^2 = \mu\left(\frac{2}{r} - \frac{1}{a}\right)$

where,

- $V \rightarrow$ Velocity
- $r \rightarrow$ Distance
- $a \rightarrow$ Semimajor axis
- $\mu \rightarrow$ Gravitational parameter

We also get,

$$a = \left(\frac{2}{r} - \frac{V^2}{\mu}\right)^{-1}$$

Now we know,

$$da = \frac{\partial a}{\partial r}\,dr + \frac{\partial a}{\partial v}\,dv$$

Thus,

$$\boxed{\frac{da}{a} = \frac{\frac{2}{r^2}\Delta r + \frac{2V}{\mu}\Delta V}{\left(\frac{2}{r} - \frac{V^2}{\mu}\right)}} \tag{1}$$

Now,

$$P = Ka^{3/2}$$

Taking the natural logarithm,

$$\ln P = \ln K + \ln a^{3/2}$$

Differentiating,

$$\boxed{\frac{dP}{P} = \frac{3}{2}\frac{da}{a}} \tag{2}$$

We know that:

$$(\text{Perihelion Time} - \text{Epoch Osculation}) = \frac{\text{Mean Anomaly}}{\text{Mean Motion}}$$

And we can approximate:

$$\boxed{d\tau \approx \frac{dP}{P} \cdot (d\tau - \text{epoch oscillation}} \tag{3}$$

- The above three equations highlighted can be used to find the value ofΔP and $\Delta \tau$. After calculating the required values, we can plug them in the Uncertainty calculation equation, which will return us categorical values ranging from 0-9 (higher means more uncertainty). Based on our research and observation of the result, we narrowed down these 10 categories to the original 3 categories (0,1,2&3 =Low, 4&5=Medium, 6+=High), which results in:-

Uncertainity Parameter	count
3.0	1283
4.0	1160
2.0	782
5.0	701
6.0	286
1.0	213
7.0	81
8.0	16
9.0	6
0.0	6

---->

Uncertainity Parameter	count
low	2284
medium	1861
high	389

8. MORE FEATURE ENGINEERING!!!!

- **Features Engineered till now**:-Eccentricity, Inclination angle, Closest Distance, Orbital Period Numerical, Heliocentric Distance, Uncertainty (created from scratch with the relevance of the current data)
- **Created Suggested Features under this section**:- Specific Angular Momentum, specific Energy, Escape velocity, Velocity at Perihelion, velocity at Aphelion, Synodic Period
- **Extra features created**:-
  True anomaly
  Mean anomaly

Normalising Data and Talk about Balancing

- Here, we used Histplots to see if the data was Gaussian in nature or not, if it was Gaussian, we preferred to use Standard Scaler, while for other values, we used Robust Scaler.

- For categorical values, we used MinMAx Scaler as these values neither are Gaussian and do not contain too many outliers either, so scaling them from 0 to 1 was the best method.

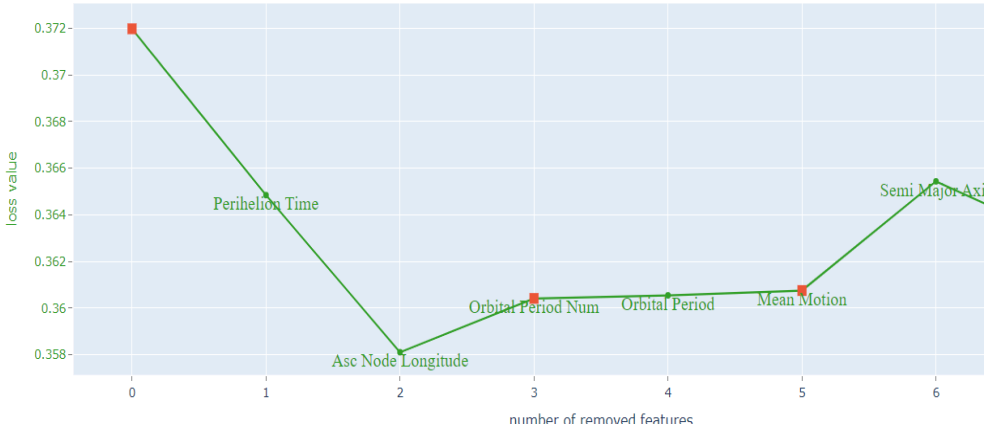
- Around 19 percent of data was hazardous, indicating highly unbalanced data. So, to tackle this, one of the most popular techniques that comes to mind is SMOTE, which we eventually avoided because of the following reasons:-
  First of all, state-of-the-art models (like XGboost, Catboost) aren't affected by SMOTE a lot, and lastly, Smote has uncertainty on having a good performance on DNN(i.e., depending on the algorithm one is using).
  (BASED ON Smote or Not to Smote research paper)

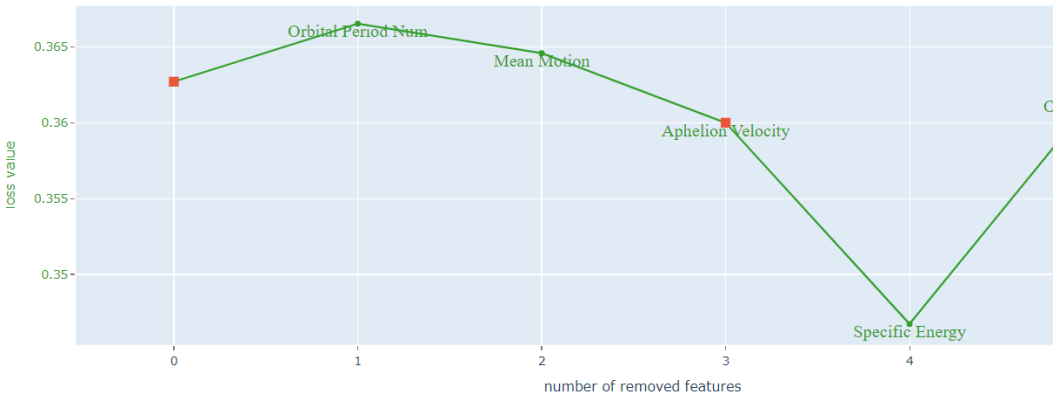
- Since there is a need to deal with Imbalance data via some mechanism as per the requirements of the problem statement. Hence, we avoided using SOTA models like xgboost and instead Created a DNN model whose mechanism of dealing with imbalanced data is discussed later.

Data Preparation for Model

- The first thing to do after dealing with EDA, normalization, and setting up binned values is feature selection, i.e., whether the features we created are relevant to the dataset or not

- For Feature selection, we used the Recursive Shap Values algorithm of Catboost, which gave us the following plots:-





- It is clear from the above plot that removing Asc Node Longitude, Perihelion Time, and Specific Energy reduces the loss significantly. Hence, these features are to be removed.

- Also additionally dropped the Name column as it logically contributed nothing.

- Finally, heading to model building, we did a train test split with 0.3 percent.

Model Training and Optimization

- As discussed before, the model we built was DNN. The overview of the model is that we used 3 dense Relu layers, followed by 1 dropout layer and 1 output layer. All the Relu layers were initialized with ***He Normal*** weights. As discussed before, imbalance was not dealt with oversampling techniques but with one very useful feature of TensorFlow `tf.nn.weighted_cross_entropy_with_logits` , this is a cross-entropy loss function which is different from your conventional cross-entropy loss function in one way, it applies weights to the under-represented class to increase its importance in loss, so that DNN doesn't become biassed towards over-represented class.
- We then used the Keras-Tuner Library(Highly compatible with tensorflow) to tune various hyperparameters such as the number of neurons in each layer, learning rate, dropout rate of the dropout layer, and also the weights for the weighted cross entropy loss function. This returns us the following model with the following parameters:-

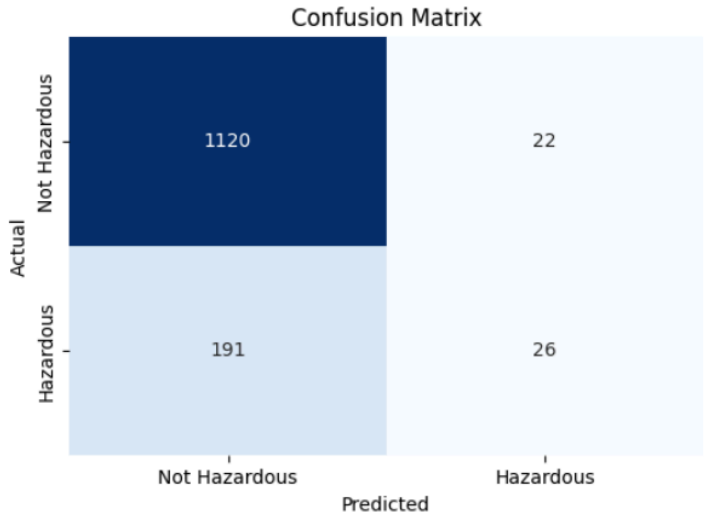
```
|Best Value So Far |Hyperparameter
|10                |units1
|6                 |units2
|18                |units3
|0.29              |drop_rate
|0.0025            |learning_rate
|2.75              |pos_weight
|10                |tuner/epochs
|0                 |tuner/initial_epoch
|1                 |tuner/bracket
|0                 |tuner/round
```

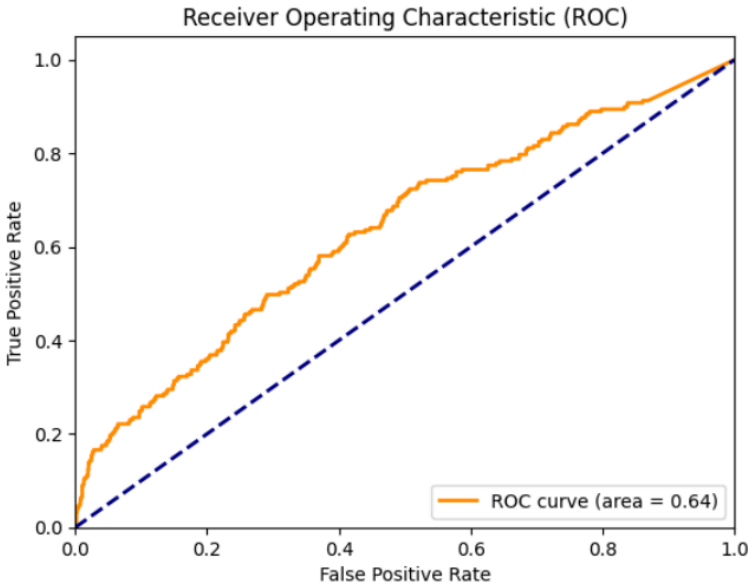
- After this, training of the model and K fold evaluation of the model was done, which showed good results, suggesting no overfitting.

Performance

- Since the data is imbalanced, the better metric to score is F1, but regardless we checked both accuracy & F1 score which are 0.843267 and 0.913167 respectively. Suggesting good predictability

- Confusion matrix and AUC ROC curve both suggest that model is working fine with good predictability too





Anomaly Detection

1. Library Based:-

- Here we used one of the most popular techniques called Isolation Forest to get the anomalies with contamination rate 0.05.

- The result we got was 227 Anomalies.

2. Self Made Algo:-

- The reasoning and concept behind the self created Algo is simple
    1. Most of the celestial objects having Jupiter Tisserant Tj<3 are comets , therefore flagging those with Tj<3 as anomaly.

    2. If Tj>3 but uncertainty parameter is greater than or equal to 7(refer to uncertainty section) will be flagged too.This is a bit weird to digest but logically makes sense, as a celestial object whose observed data is highly uncertain ,it becomes a highly unpredictable object. Hence a need to be flagged.

    3. This algo returned 140 Anomalies in total.

After this we plotted the confusion plot to check how many anomalies does Self made algo and Library based anomaly detection have in common.