**PROJECT REPORT**

**ON**

**POTENTIAL ASTEROID HAZARD**



Submitted by-

**Team 20**

Sneha Ravichandran (001096251)

Pavan Sai Pati (001521280)

**Abstract**

This report consists of the final findings of the Supervised Machine Learning algorithms used for the prediction of the Potential Hazard Asteroids.

The dataset used for the project has been taken from Kaggle. The data is then cleaned as per the requirement and then various models were built in Jupyter Notebook (Anaconda 3) to examine the models’ performance on certain parameters.

After a preliminary study of the available algorithms and data review, it became apparent that the problem fell under non-linear Classification category. The study focuses on various algorithms by using classifiers like- Logistic Regression Classification, Decision Tree Classification, Gaussian Naïve Bayes Classification, and Random Forest Classification.

**Table of Contents**

* Introduction
  + Background
  + Motivation
  + Goal
* Methodologies and Algorithms
  + Data Review
  + Software and Libraries used
  + Data Cleaning
  + Feature Selection
  + Exploratory Data Analysis
  + Methods used
    - Logistic Regression
    - Logistic Regression using SMOTE
    - Gaussian Naïve Bayes
    - Gaussian Naïve Bayes with Hyperparameter tuning
    - Decision Tree
    - Random Forest
* Description of Dataset
* Results and Analysis
* Conclusion
* Reference

**Introduction**

**Background**

Potential Asteroid Hazard prediction system aims to predict asteroids hazardous to earth based on the data collected till date by the Jet Propulsion Laboratory of California Institute of Technology(an organization under NASA). This system has ‘PHA’ as the target variable and its predictions are based on a variety of intrinsic and contextual attributes such as orbit\_id, near earth object value, diameter, moid\_id, epoch etc.

**Motivation**

Outer space is an intriguing topic. Every year, the Earth is hit by about 6100 meteors large enough to reach the ground, or about 17 every day, research has revealed. The vast majority fall unnoticed, in uninhabited areas. Over 1000 people were injured by the Chelyabinsk meteor airburst event over Russia in 2013. Such events could be avoided if an asteroid was monitored right from when it is determined to be a hazard to Earth. Thus, we decided to work on this project, so to predict and prevent a probable asteroid hit, and eventually save lives.

**Goal**

Our goal is to determine whether a particular asteroid poses a hazard to Earth or not. We will be achieving this by building a machine learning model to learn the features and tune it to predict the target (PHA) variable.

* Using supervised machine learning methods to predict the PHA value of an asteroid as Y/N
* To determine the most efficient algorithms with the highest accuracy score for the given Dataset
* Visualizing the performance of the models using confusion matrix

**Methodology & Algorithm**

**Data Review**

The dataset contains records of all the asteroids recorded till date, it has 958524 records (435 MB size) and contains 45 columns.

**Software and Libraries used**

The dataset is downloaded from Kaggle. Jupyter notebook is used for the coding aling with several Libraries

That are-

* Numpy
* Pandas
* Matplotlib
* Seaborn
* Metrics
* Scikitlearn Tree
* DecisionTreeClassifier
* train\_test\_split
* accuracy\_score
* sklearn.ensemble
* RandomForestClassifier
* plotly.graph\_objects
* sklearn.preprocessing
* Get\_dummies
* GridSearchCV
* Pydotplus
* Graphviz
* R2\_score
* Sqrt
* Mean\_squared\_error

And several others for classification models.

**Data Cleaning**

Data provided was heterogenous with a couple columns containing missing values. Many rows had Null values which can degrade the model’s performance, hence we need to take care of all the rows with Null values. We used the mean values of the features to fill in the missing fields instead of leaving them Null and removed the rows that had null values after filling the mean values.

**Feature Selection**

#### The data originally had 45 columns, this can lead to too much noise and degrade the performance of the model. To avoid this noise we dropped and retained selected features based on the correlation matrix and comparing how it affects the target variable. Also, removed duplicate columns which were highly correlated to each other.

**Exploratory Data Analysis**

After cleaning the data and sorting it we have done feature selection on it using various pandas and matplotlib commands. We have shown a histogram of all relevant columns using hist() and visualized the percentage of near earth objects and also printed a pie chart of PHA count in the dataset. We have visualized the distribution of asteroids among multiple classes

**Models Used**

The Dependent Target Variable is PHA which contains a boolean value of Y or N

As the Target Variable is a discrete class value, the prediction model used is **Classification**.

The dataset with its matrix of features (independent variables) is trained on various Classification models to predict the class value of the dependent target variable.

Different types of classifiers used in the project are-

* Logistic Regression Classification
* Logistic Regression Classification using SMOTE
* Gaussian Naïve Bayes Classification
* Gaussian Naïve Bayes Classification with Hyperparameter tuning
* Decision Tree Classification
* Random Forest Classification

We have also tried using the above models after tuning the hyperparamaeters and after oversampling the minority class.

1. **Logistic Regression Classification**

This classifier is used to transform its output using the logistic sigmoid function to return a probability value. Logistic Regression was pretty accurate with the prediction of the target variable for our data set. But the false negatives were high.

1. **Logistic Regression Classification using SMOTE**

SMOTE is an oversampling technique where the synthetic samples are generated for the minority class(True positives (PHA=Y). This algorithm helps to overcome the overfitting problem posed by random oversampling.

This method helped reduced the false negatives for our dataset, but eventually improved the recall and decreased the overall accuracy of the logistic regression.

1. **Gaussian Naïve Bayes Classification**

This classifier is based upon Naïve Bayes Theorem which works on strong independence assumptions between the features. It is easy and faster in comparison to other methods. This model did not give us good results when compared to the other algorithms hence hyper parameter tuning was done to select the best parameters.

1. **Gaussian Naïve Bayes Classification with Hyperparameter tuning**

Hyperparameter optimization or tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. Cross validation is often used to generalize the performance. It picked up the best value for var\_smoothing attribute and gave us a better accuracy rate then the Gaussian Naïve Bayes Classification without hyperparameter tuning.

1. **Decision Tree Classification**

Decision tree is highly useful in classification problems where the total number of features and rows is high. A decision tree is represented as upside down where its root is at the top of the tree then it splits into branches and when it cannot further split then the end branch is called as decision. Growing a decision tree requires to choose features and conditions to select optimal tree which has maximum prediction. The tree is grown arbitrary.

This model gave the best result for the asteroid dataset. The accuracy was 99.9%

1. **Random Forest Classification**

Random forest is at par with decision tree in terms of getting result both have given satisfactory results. With accuracy score reaching as high as 99%.

The random forest is a flexible algorithm which is easy to use and takes very less time when compared to logistic regression. In decision tree only one tree is made but in random forest, our algorithm randomly creates a specified number of decision trees. And chooses the tree which is best for our model.

This model gave us the second best result with respect to the F1 score the a tie along with the decision tree model for the accuracy score.

**Visualization**

Visualization is done at classification models as well as in EDA. Where bar graph, line graphs, pie charts, confusion matrices etc. are used to represent data as well as data frames have been used. The following libraries were used as well: Matplotlib, Seaborn and Plotly.graph.

**Description of Dataset**

The Potential Asteroids Hazard Dataset contains all the information recorded regarding asteroids to date.

The PHA column (Potentially hazardous asteroid column) tells us if an asteroid is a potential hazard to Earth or not and uses Boolean value (Y or N)

The dataset consists of one .csv file which is 435 MB and has 958524 records and contains 45 columns.

Of which:

* 7 attributes are String,
* 2 attributes are Boolean,
* 36 attributes are Decimal

List of Attributes:

|  |  |  |
| --- | --- | --- |
| S.no | Attributes | Description |
| 1 | SPK-ID | * Object primary SPK-ID |
| 2 | Object ID | Object internal database ID |
| 3 | Object fullname | Object full name/designation |
| 4 | pdes | * Object primary designation |
| 5 | name | Object IAU name |
| 6 | NEO | Near-Earth Object (NEO) flag |
| 7 | **PHA** | **Potentially Hazardous Asteroid (PHA) flag** |
| 8 | H | * Absolute magnitude parameter( the visual magnitude an observer would record if the asteroid were placed 1 Astronomical Unit(au) away) |
| 9 | Diameter | object diameter (from equivalent sphere) in km Unit |
| 10 | Albedo | * Geometric albedo( propotion of the incident light or radiation that is reflected by a surface) |
| 11 | Diameter\_sigma | * 1-sigma uncertainty in object diameter km Unit |
| 12 | Orbit\_id | * It contains the Orbit solution ID |
| 13 | Epoch | Epoch of osculation in modified Julian day form. |
| 14 | Equinox | Points in the sky where the ecliptic (the Sun's annual pathway) and the celestial equator intersect. |
| 15 | e | It determines the amount by which its orbit around another body deviates from a perfect circle |
| 16 | a | * Semi-major axis au Unit. |
| 17 | q | perihelion distance au Unit ( the point in the orbit of an asteroid which is closest to the sun) |
| 18 | I | Inclination :angle with respect to x-y ecliptic plane |
| 19 | tp | Time of perihelion passage TDB Unit |
| 20 | moid\_ld | Earth Minimum Orbit Intersection Distance au Unit (distance between the closest points of the osculating orbits of two bodies) |
| 21 | ma | Mean anomaly |
| 22 | Sigma\_e | Error in measuring the e value |
| 23 | Sigma\_a | Error in measuring the a value |
| 24 | Sigma\_q | Error in measuring the q value |
| 25 | Sigma\_i | Error in measuring the i value |
| 26 | Sigma\_o | Error in measuring the o value |
| 27 | Sigma\_m | Error in measuring the m value |
| 28 | Sigma\_w | Error in measuring the w value |
| 29 | Sigma\_ma | Error in measuring the ma value |
| 30 | Sigma\_ad | Error in measuring the ad value |
| 31 | Sigma\_n | Error in measuring the ad value |
| 32 | Sigma\_tp | Error in measuring the ad value |
| 33 | Sigma\_per | Error in measuring the ad value |
| 34 | class | Orbit classification |
| 35 | rms | radio millimeter submillimeter wavelength |
| 36 | Prefix | Prefix given to the asteroid name |
| 37 | Epoch\_mjd | It uses 63-bit date/time, which allows times to be stored up to July 31, 31086, 02:48:05.47 |
| 38 | Epoch\_cal | Epoch of osculation calculated in calendar date/time format |
| 39 | om | Sidereal orbital period |
| 40 | w | Normalized rms of orbit fit |
| 41 | ad | Aphelion distance |
| 42 | n | Mean motion degree/d |
| 43 | Tp\_cal | Time of perihelion passage TDB Unit |
| 44 | Per | Argument of Perihelion |
| 45 | Per\_y | Longitude of the ascending node |

**Data Source**

The dataset has been taken from Kaggle:

Asteroid Dataset

NASA JPL Asteroid Dataset

<https://www.kaggle.com/sakhawat18/asteroid-dataset>

**Results And Analysis**

**Data Exploration**

The given dataset has large number of rows that is 958,524 and 45 columns. Our aim is to predict if a particular asteroid is hazardous to earth or not based on the different features provided. On Initial inspection, couple of the features contained null values so to sort this we used data cleaning and feature selection along with correlation matrix to prepare the data for the machine learning models. We have dropped several columns and rows and our final data has 35 columns and 938,597 rows.

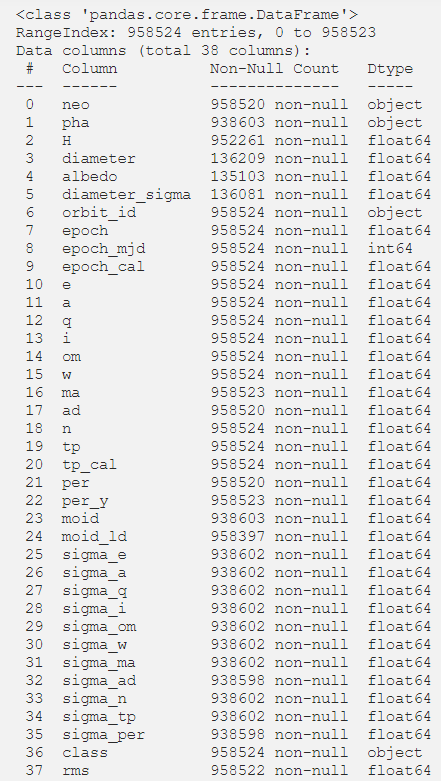
**Exploratory Data Analysis :**

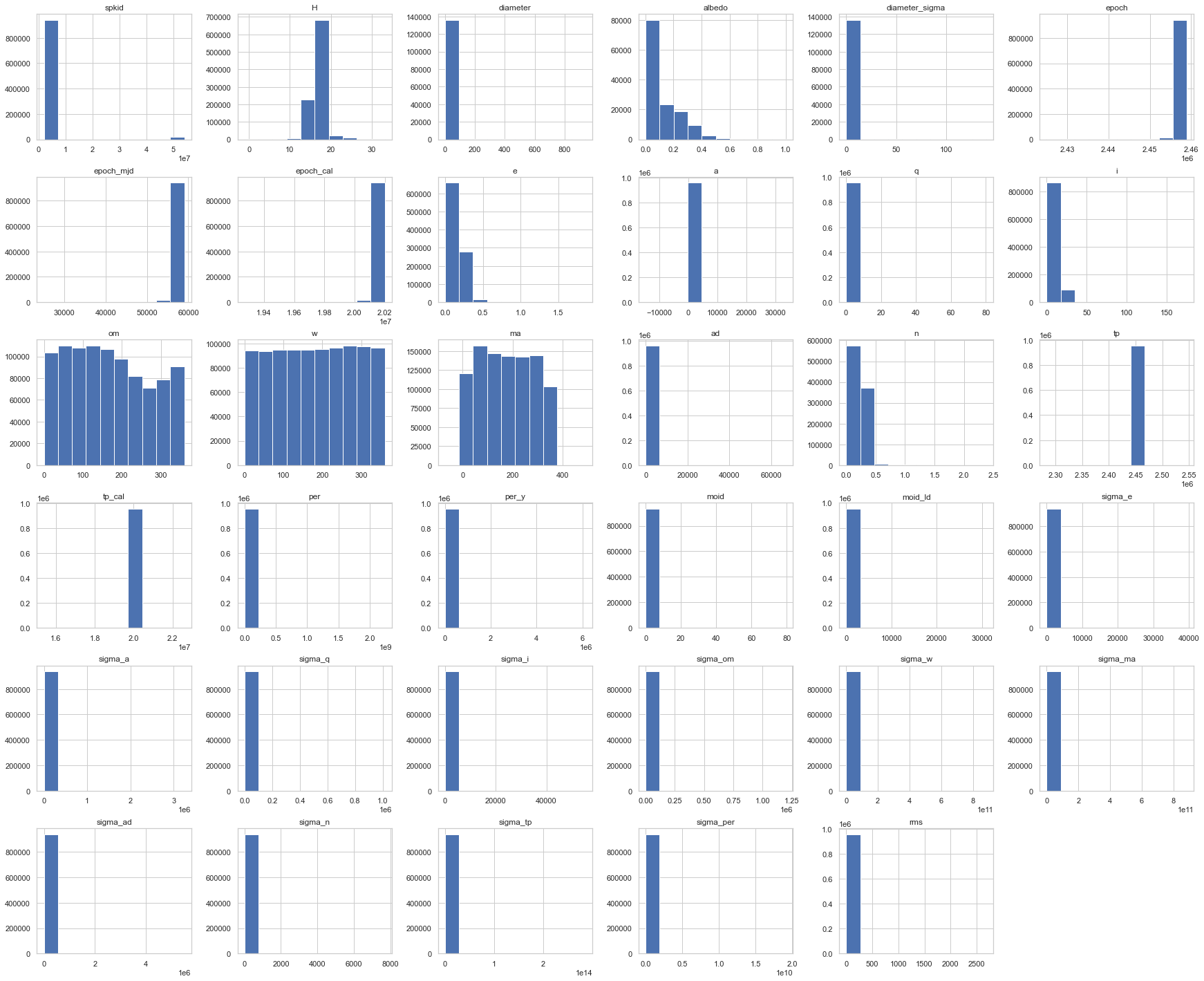
Diagram

Description automatically generated

In the above PieChart, we are checking the count of potential hazard and non potential hazard asteroids

Printing the type of data in the dataset and their respective records count.





The above graph shows the histogram for all the features.

Chart, bar chart

Description automatically generated

In the above chart we are categorizing the number of asteroids per class.

Diagram

Description automatically generated

In the above diagram, we are counting the near earth and non near earth object.

Chart, treemap chart

Description automatically generated

We are visualizing the correlation matrix to get an idea of the data and the correlation between the features.

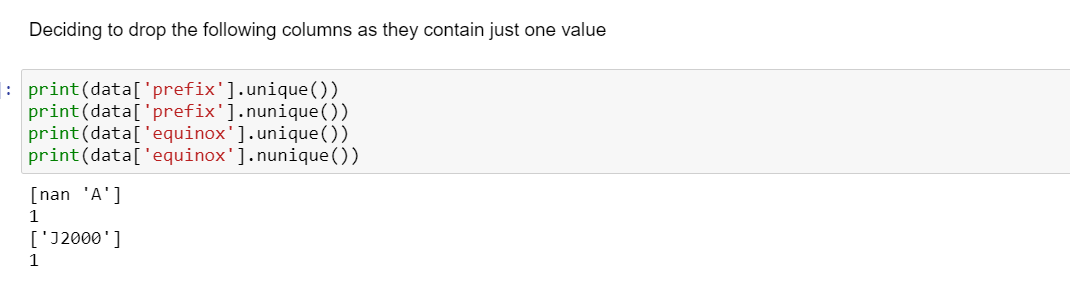
**Data Preprocessing**

* **Dropping useless Features**

The Features that are not governing a direct relationship with the prediction of the Target and the ones that are redundant are removed from the dataset to provide noise free data to the model.

* **Dropping redundant Features**

On checking number of unique values in all the Features,



The outcome shows one unique value in all the records for the features ’Prefix‘ and for ’equinox’ it is always 1. Thus such Features have same value aren’t of our use and thus decided to drop them from the dataset. Id, pdes and name are redundant features as well.



* **Replacing the values for Features with Maximum Missing Data**

It is found that the Feature ‘diameter’, ‘ albedo’, ‘diameter\_sigma’ and ‘H’ has the highest missing data. As the variables are required for the prediction of target variable ‘pha’ with their high correlation, these Features have been replaced with the mean value of the respective feature in the dataset.

Graphical user interface, application

Description automatically generated

* **Removing features based on correlation matrix**

Graphical user interface, chart, treemap chart

Description automatically generated



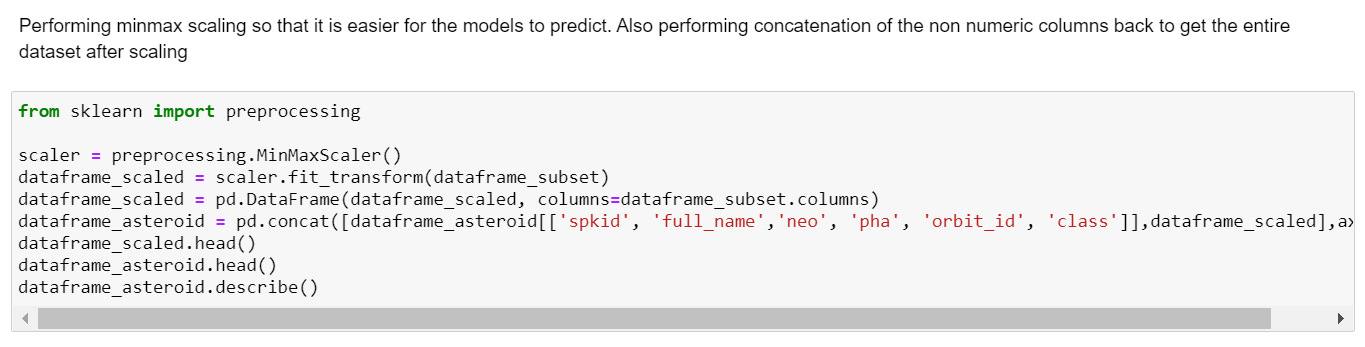
In the above code, we are dropping the redundant correlated features.

* **Dropping records with Null value**

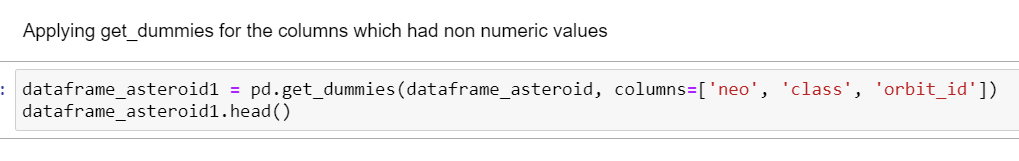
Dropping ‘sigma\_ad’ and ‘ma’ rows that contain null values.



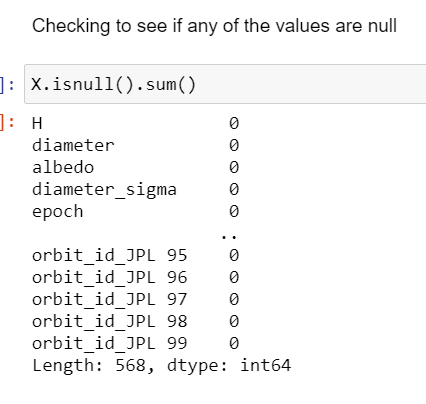
* **Performing MinMax Scaling**



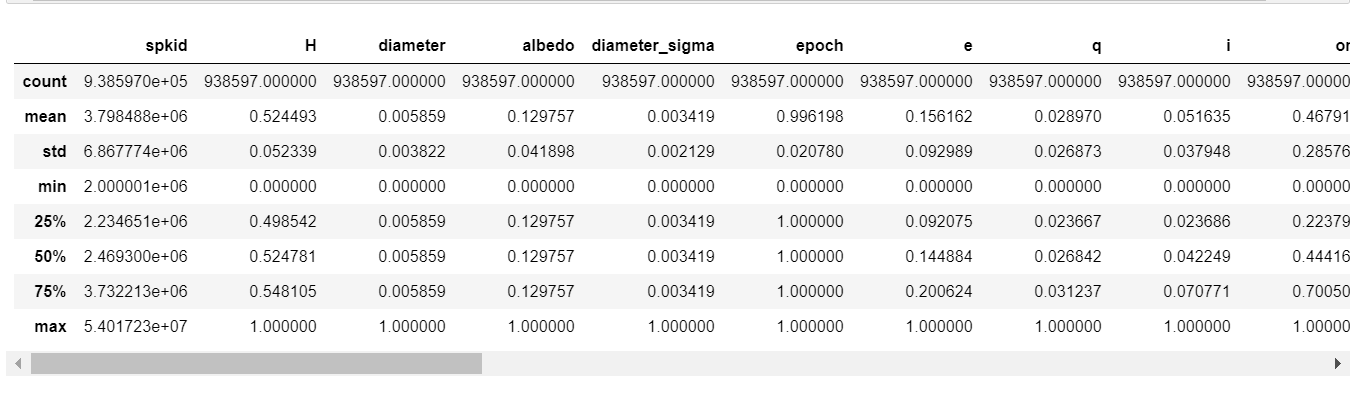
* **Applying get\_dummies for the non-numeric columns**

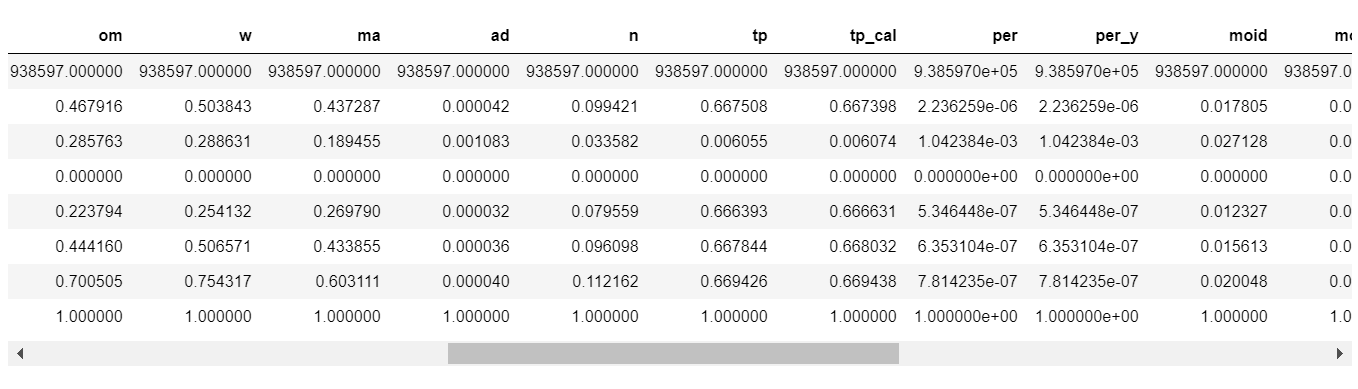


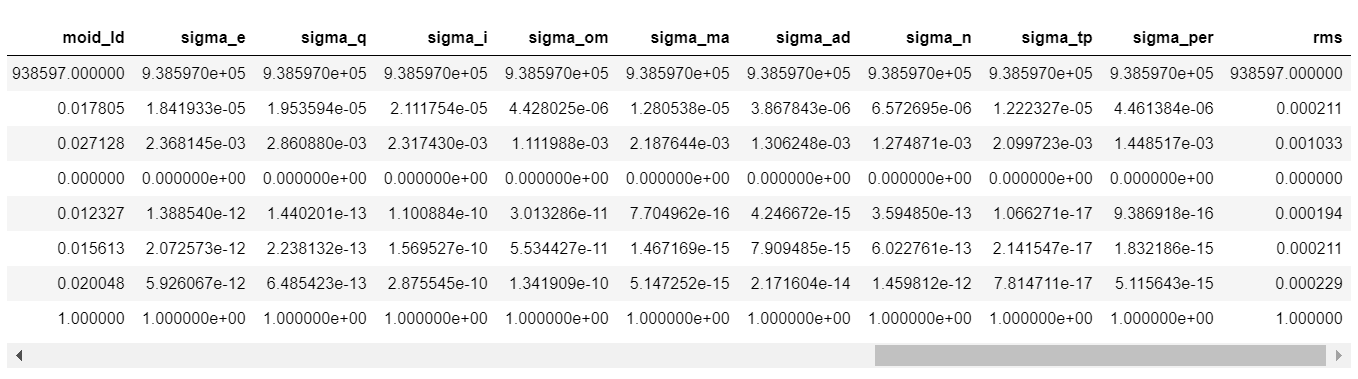
* **Checking to see if the dataset contains null values after preprocessing**



**Description of the Dataset after Data cleaning**









**Correlation after Data preprocessing and data cleaning**

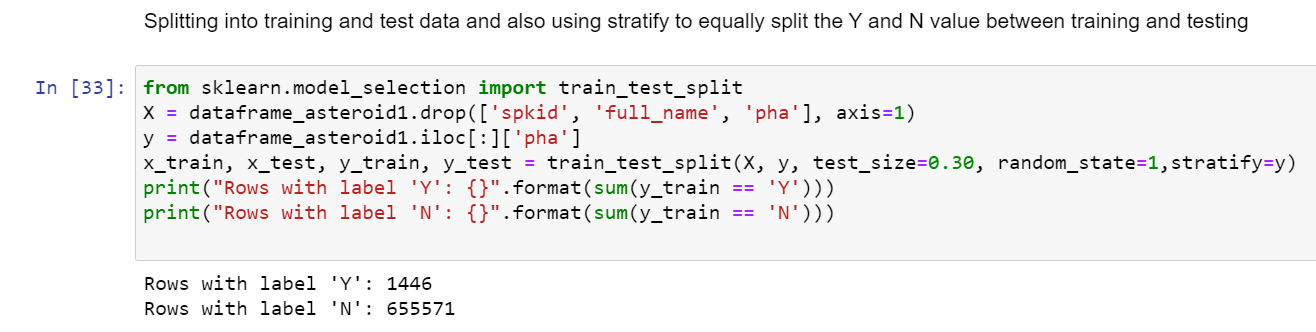
Graphical user interface, chart, treemap chart

Description automatically generatedThe above is the correlation matrix of the features after data preprocessing and data cleaning.

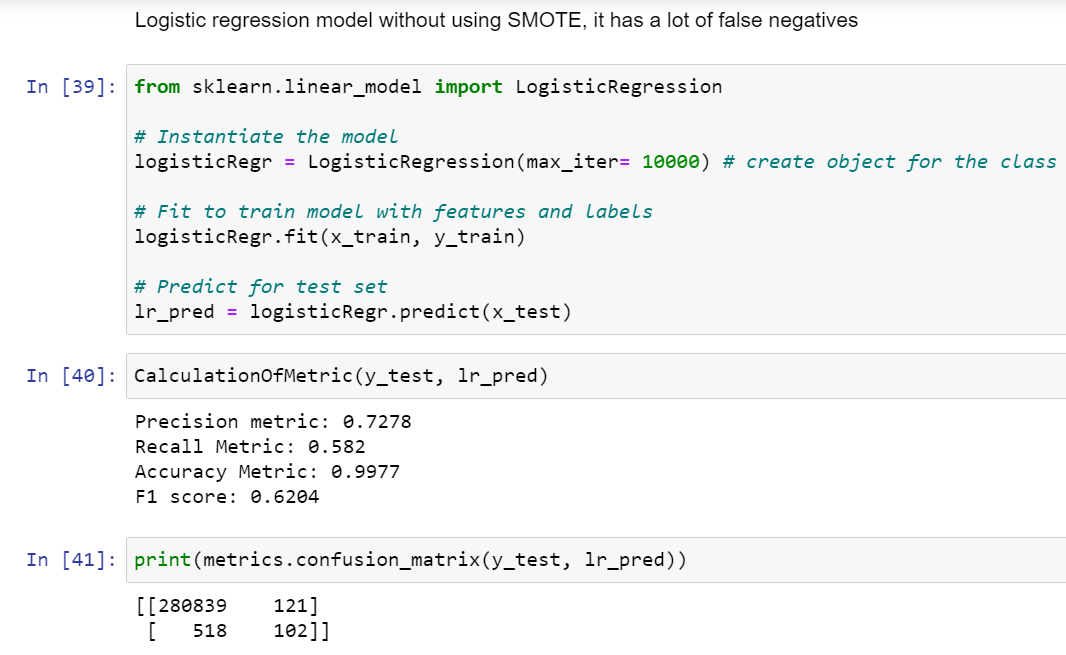
**Models for Predicting the Potential Hazard Asteroids**

**Splitting Training and Testing Data**

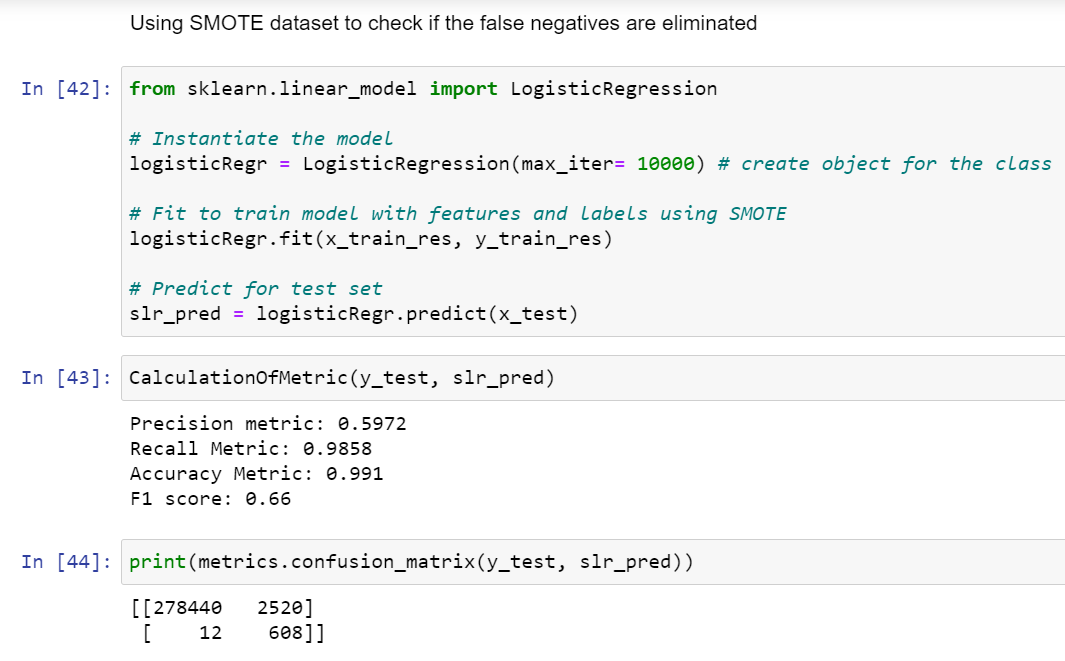
Data is separated into Training set and Test set into 70% and 30% and using stratify option to balance the labels in both the training and test set.



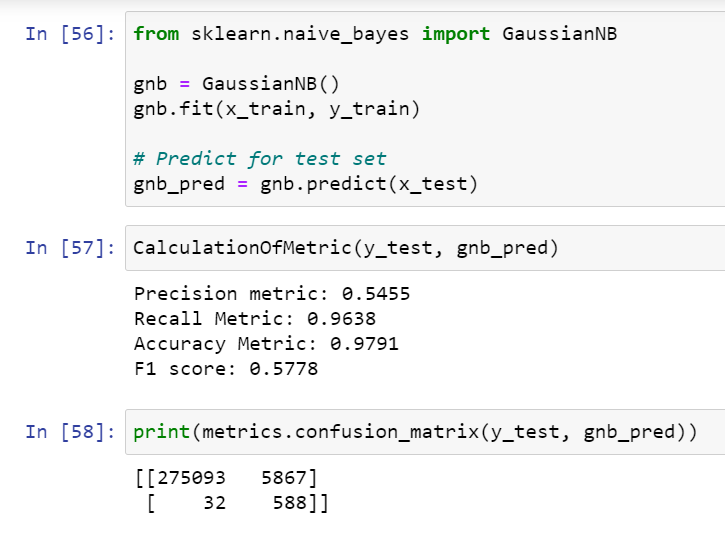
1. **Logistic Regression Classification**:



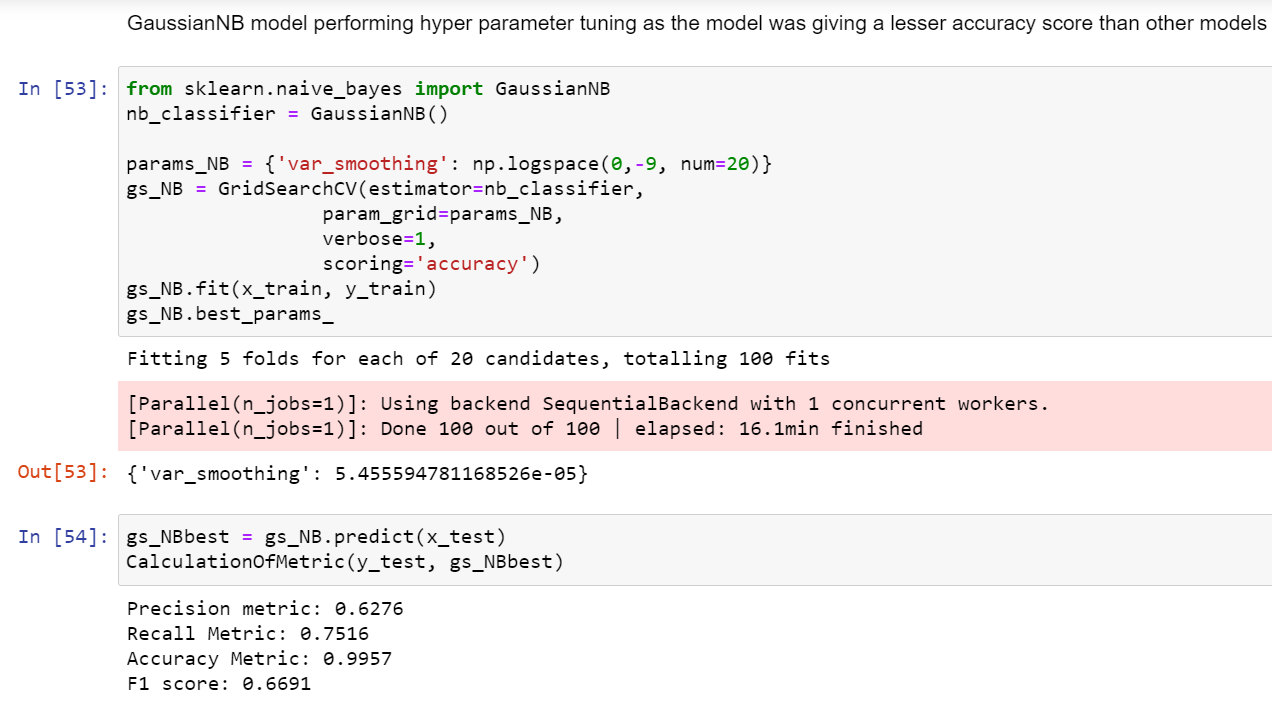
1. **Logistic Regression Classification after oversampling using SMOTE:**

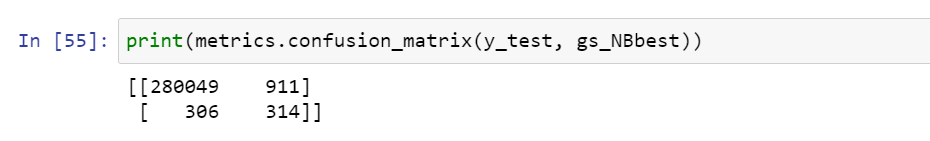


1. **Gaussian Naïve Bayes Classification:**

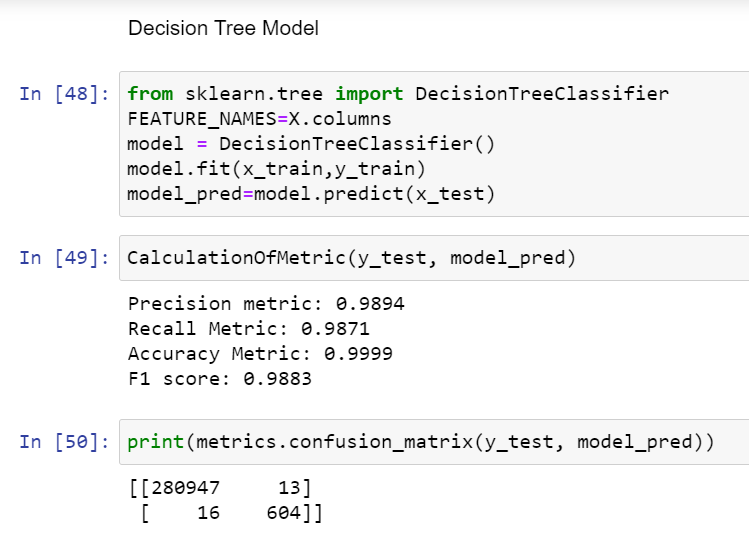


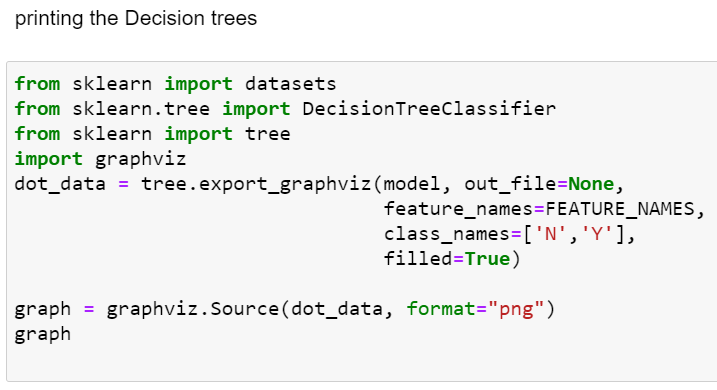
1. **Gaussian Naïve Bayes Classification after Hyperparameter Tuning:**





1. **Decision Tree Classification:**

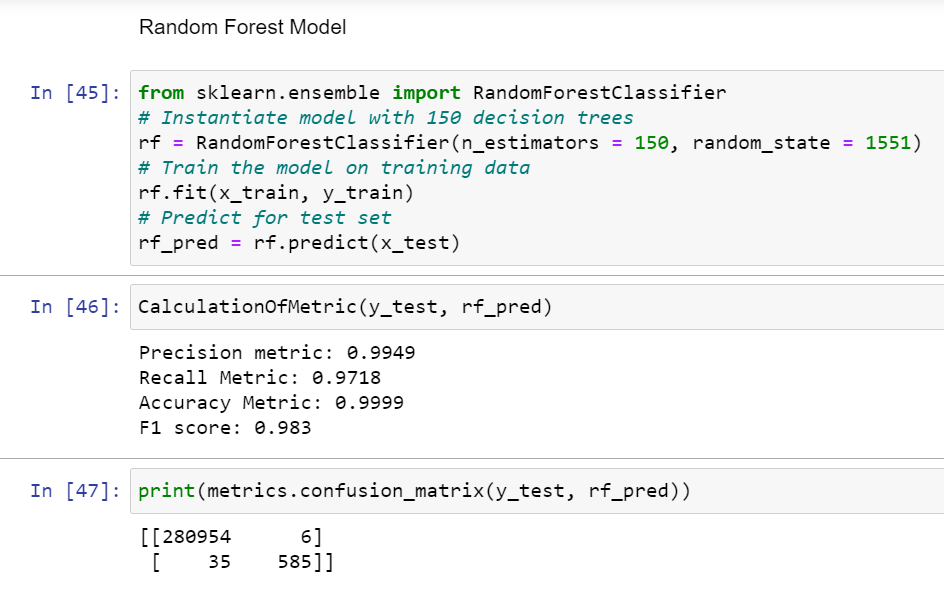




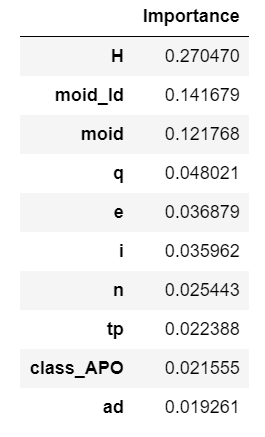
Timeline

Description automatically generated

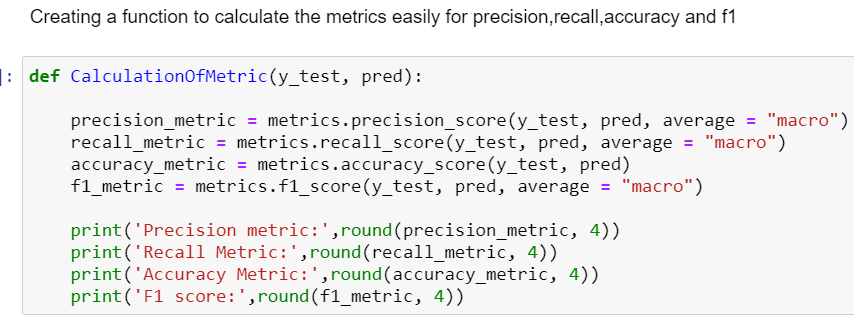
1. **Random Forest Classification:**



Feature importance for Random forest classifier:



**Conclusion**



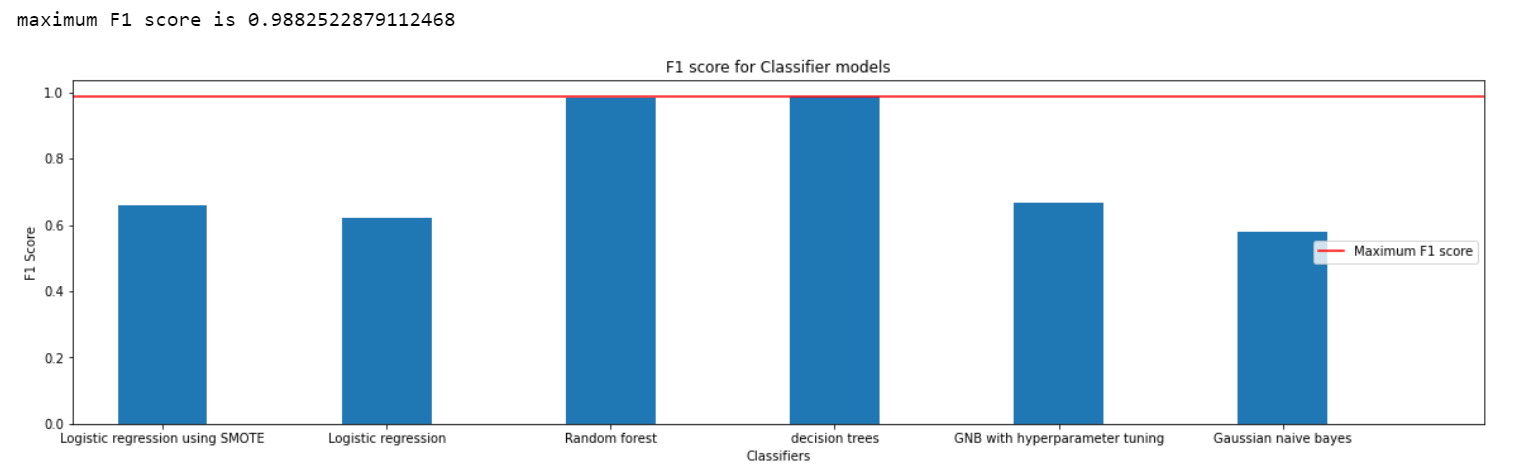
**Table of Outputs**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Precision | Recall | Accuracy | F1 score |
| Logistic Regression | 0.7278 | 0.582 | 0.9977 | 0.6204 |
| Logistic Regression (SMOTE) | 0.5972 | 0.9858 | 0.991 | 0.66 |
| Random Forest | 0.9949 | 0.9718 | 0.9999 | 0.983 |
| Gaussian Naive Bayes (Hyperparameter Tuning) | 0.6274 | 0.7516 | 0.9957 | 0.6691 |
| Gaussian Naive Bayes | 0.5455 | 0.9638 | 0.9791 | 0.5778 |
| Decision Tree | 0.991 | 0.9871 | 0.9999 | 0.989 |

**Accuracy Scores** :Chart, bar chart

Description automatically generated

**F1 Scores :**



After analysing the above algorithms, the **Decision Tree Classifier and the Random Forest Algorithm Classifier** produced the best accuracy (99.99%) for the prediction of Target Variable (Potential Asteroid Hazard).



**References**

* <https://cneos.jpl.nasa.gov/tools/ast_size_est.html> (JPL website for Exploratory data analysis)
* <https://www.kaggle.com/basu369victor/prediction-of-asteroid-diameter/code> (Regarding the diameter and albedo calculation)
* https://www.sciencedirect.com/topics/computer-science/logistic-regression
* https://www.newgenapps.com/blog/random-forest-analysis-in-ml-and-when-to-use-it/
* <https://www.datacamp.com/community/tutorials/naive-bayes-scikit-learn>
* <https://cosmosmagazine.com/space/earth-hit-by-17-meteors-a-day/>
* <https://medium.com/analytics-vidhya/detecting-potentially-hazardous-asteroids-using-deep-learning-part-1-9873a0d97ac8>
* <https://ssd.jpl.nasa.gov/sbdb_query.cgi>
* <https://en.wikipedia.org/wiki/Decision_tree_learning>
* https://www.geeksforgeeks.org/ml-feature-scaling-part-2/
* https://stackoverflow.com/questions/47606873/jupyter-no-module-named-imblearn-after-installation
* https://www.kaggle.com/marissafernandes/asteroid-prediction
* https://stackoverflow.com/questions/39828535/how-to-tune-gaussiannb
* https://www.youtube.com/watch?v=pooXM9mM7FU
* https://www.youtube.com/watch?v=VqKq78PVO9g
* https://www.youtube.com/watch?v=FheTDyCwRdE
* https://en.wikipedia.org/wiki/Hyperparameter\_optimization#Bayesian\_optimization
* <https://ssd.jpl.nasa.gov/sbdb_query.cgi>
* <https://towardsdatascience.com/all-about-missing-data-handling-b94b8b5d2184>
* <https://medium.com/@michaeldelsole/what-is-one-hot-encoding-and-how-to-do-it-f0ae272f1179>
* <https://likegeeks.com/python-correlation-matrix/>