INDEX

List of Tables

List of Figures

1. Problem Setting:

An asteroid is a small, rocky object that orbits the sun. Asteroids are considered to be remnants of the early solar system and can range in size from a few meters to several hundred kilometers. Scientists study asteroids to learn and identify the potential hazards to earth. If an asteroid is headed toward Earth or has a high chance of hitting the globe, it may be deemed hazardous to the planet and rest deemed as non-hazardous. The likelihood that an asteroid may collide with Earth and the potential kinetic energy of such an impact are additional considerations that are considered when classifying asteroids as hazardous or non-hazardous.

2. Problem Definition:

This investigation set out to find the most accurate classification model and predictors for classifying hazardous and non-hazardous asteroids. The size, orbit, and velocity of an asteroid are factors that are used to assess its risk. To estimate the potential dangers posed by new discoveries, the physical features and orbital parameters of known asteroids are provided by NASA's asteroid dataset. The goal is to learn more about the characteristics that lead to the claim that an asteroid is dangerous.

3. Data Sources:

The data for Asteroid information has been taken from NeoWs (Near Earth Object Web Services) which gives information based on their closest approach towards Earth. The link for the dataset is Asteroid dataset. The data was captured and a prescription for a dataset from NASA.

4. Data Description:

The dataset contains around 5000 instances each of different asteroids. There are 39 attributes and 1 target attribute. There are a total 40 attributes which contain the various information about the physical dimension.

Main attributes are as listed below -

Table 4.1: Description of Variables

Column	Description
Name	Asteroid Identifying Name
Absolute Magnitude	Visual magnitude an observer would record if the asteroid were placed 1 Astronomical Unit
Est Dia	Estimated Diameter of the asteroid in various units
Relative Velocity km per sec	Relative velocity of the asteroid in km per second.
Orbiting Body	Planet around which the asteroid is revolving.
Jupiter Tisserand Invariant	Tisserand's parameter (or Tisserand's invariant) is a value calculated from several orbital elements(semi-major axis, orbital eccentricity, and inclination) of a relatively small object and a more substantial' perturbing body'.
Eccentricity	Eccentricity of the asteroid's orbit. Just like many other bodies in the solar system, the realms made by asteroids are not perfect circles, but ellipses. The axis marked eccentricity is a measure of how far from circular each orbit is the smaller the eccentricity number, the more circular the realm.
Semi Major Axis	Value of the Semi Major Axis of the asteroid's orbit. As discussed above, the realm of an asteroid is elliptical rather than circular. Hence, the Semi Major Axis exists.
Orbital Period	Orbital period of the asteroid. Orbital period refers to the time taken by the asteroid to make one full revolution around its orbiting body.
Perihelion Distance	Value of the Perihelion distance of the asteroid. For a body orbiting the Sun, the point of least distance is the perihelion.
Aphelion Dist	Value of Aphelion distance of the asteroid. For a body orbiting the Sun, the point of greatest distance is the aphelion.
Hazardous (Target variable)	This feature denotes whether the asteroid is hazardous or not.

5. Data Mining Tasks:

5.1 Data Understanding

The dataset contains around 4688 instances each of different asteroids. There are 39 attributes that describe the attributes about the asteroid and 1 target attribute that classifies if the asteroid is hazardous or not. Target attribute contains 'Hazardous' or 'Non hazardous' in which the class "Hazardous" is our class of interest. There are both - numeric and categorical attributes. Some of the numeric variables are given in different units which are redundant and can be eliminated.

5.2 Data Processing

Columns such as the "Name" column are unique identifier columns that can be eliminated while modeling as they will not serve any purpose. Estimated diameter of asteroid attributes are represented in multiple units of measure. These columns are redundant and can be eliminated. And few other features such as "Relative Velocity km per hr", "Relative Velocity km per sec", "Miss Dist.(Astronomical)", "Miss Dist.(kilometers)", etc. have high correlation with each other and can be eliminated. Other features such as "Orbiting Body", "Close Approach Date", "Orbit Determination Date", "Equinox" are eliminated either they just had one single value for all the instances or they were of the type Date time objects and does not provide significant information. There are no missing values, which is ideal and are perfect for the application of classification algorithms to the dataset.

5.3 Data Exploration:

The primary goal of Exploratory Data Analysis (EDA) is to assist in the analysis of data prior to making any assumptions. It can aid in the detection of evident errors, as well as a better understanding of various data patterns, the detection of outliers or anomalous events, and the discovery of intriguing relationships between variables.

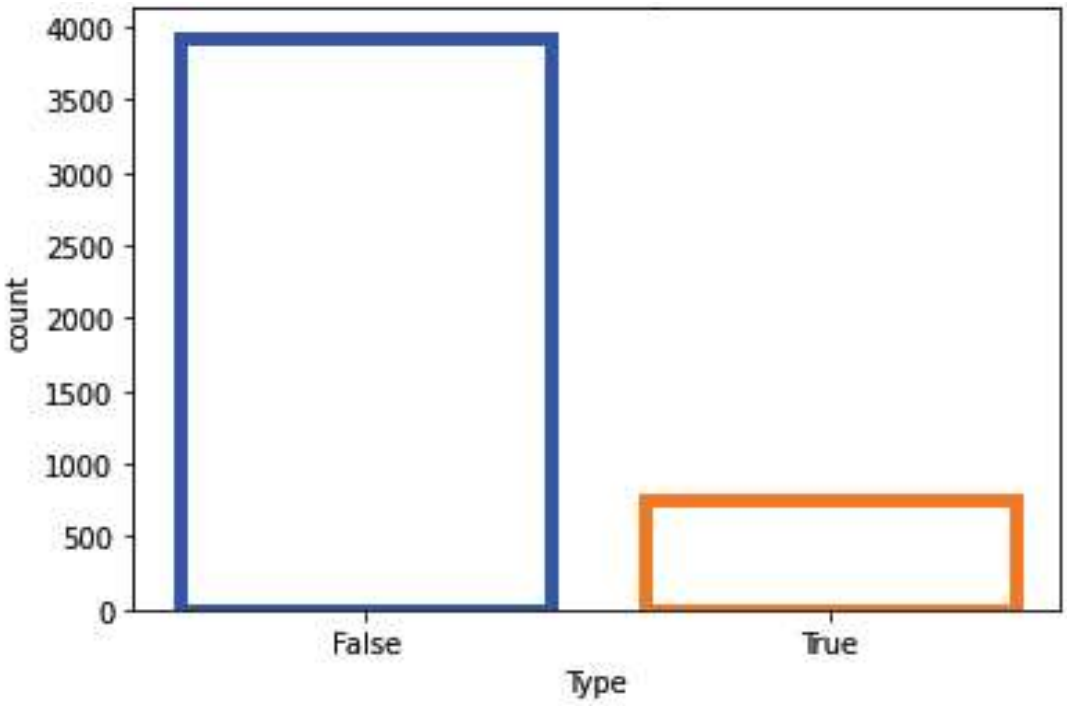
As target variable hazardous contains categorical data can be defined binary. By using the countplot function from seaborn library, we plotted the target variable over the count on the y axis.

In the plot, We defined two classes which are described as:

'**False**': An asteroid which is non hazardous.

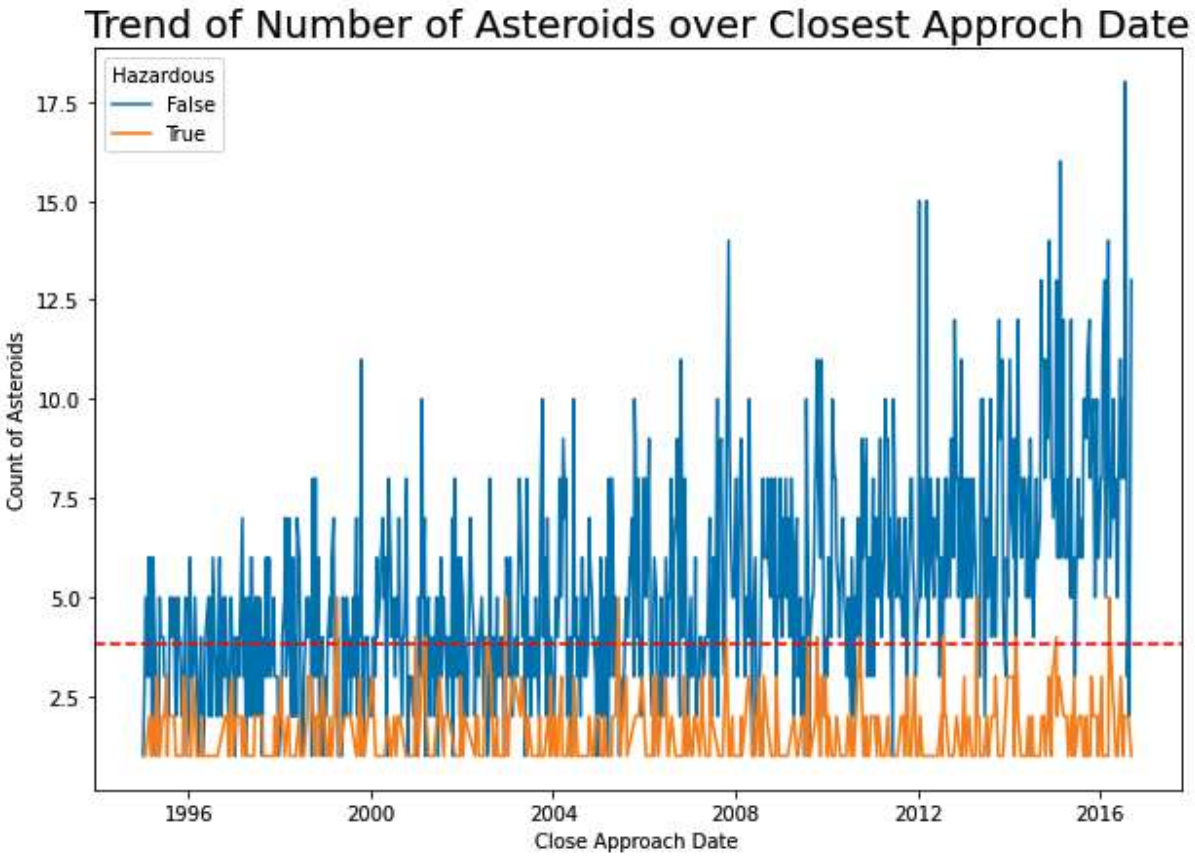
'**True**': An asteroid which is hazardous.

Figure 5.3.1: Distribution of Target Variable



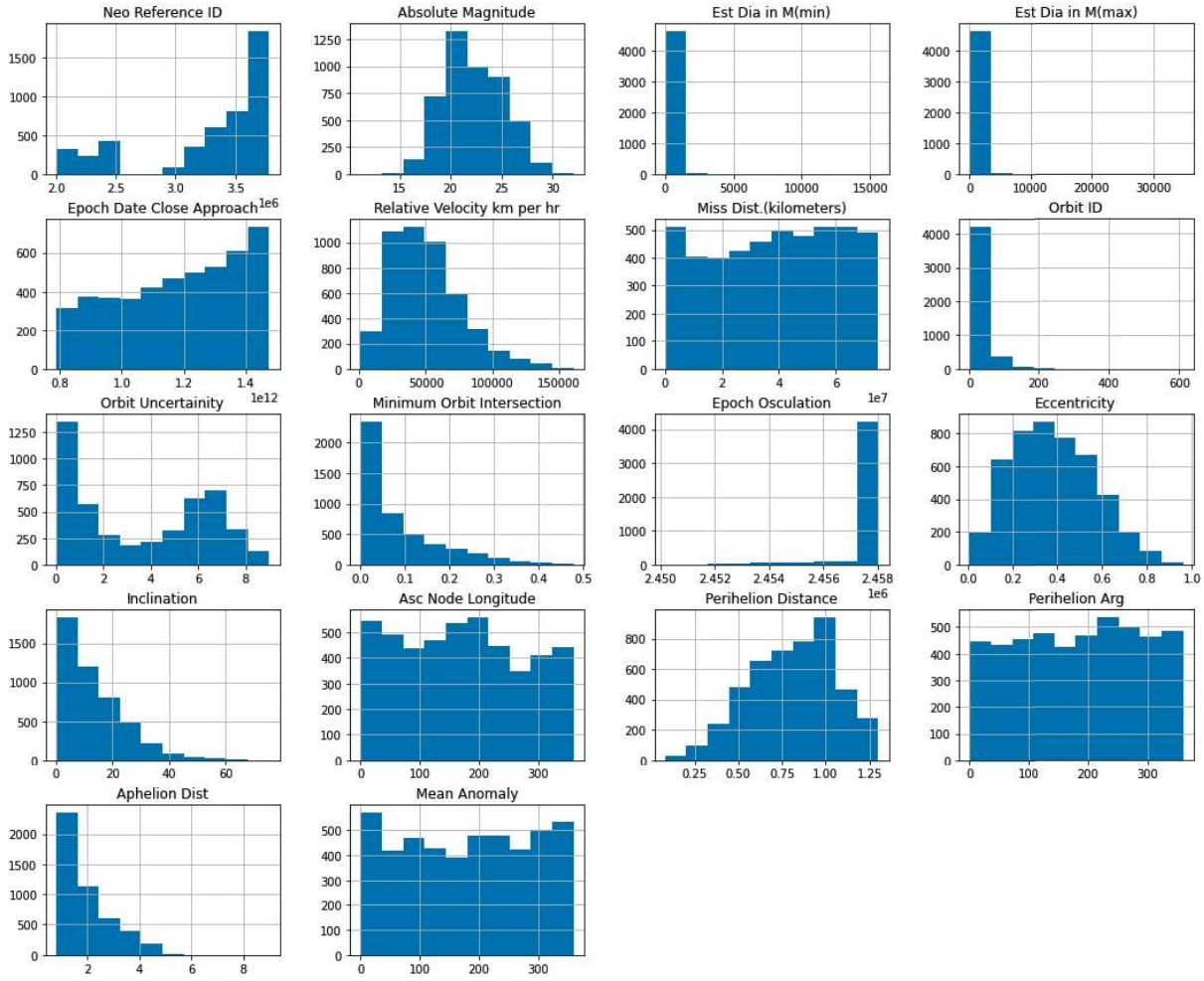
There are 3932 asteroids that are non-hazardous whereas 755 are hazardous. It can be observed that the given dataset is an imbalanced dataset which requires oversampling of data. This oversampling is to be done in a trial and error method which will be done during the model selection process.

Figure 5.3.2: Number of asteroids over the closest approach date



The above time series plot is plotted to see the trend of the approaching asteroids by considering the 'closest approach date'. The plot shows the average number of asteroids that are approaching is indicated by a dotted red line in the plot which counts around 6 asteroids over the time span. When divided between hazardous and non-hazardous, it can be seen that there is a gap between the both. But this is mostly due to the high number of occurrences of non hazardous asteroids compared to hazardous ones.

Figure 5.3.3: Distribution of all the numerical variables



This plot shows the different variables and its distributions. This is done in order to analyze if any transformations are needed on the variables. Few variables such as “Inclination”, “Aphelion Dist” are in log-normal distribution which could be transformed to normal distributions which are ideal for model fitting.

We performed box-cox transformation given by on such variables -

$$y(\lambda) = \begin{cases} \frac{y^{\lambda}-1}{\lambda}, & \text{if } \lambda \neq 0; \\ \log y, & \text{if } \lambda = 0. \end{cases}$$

The results are shown in figure 6 of Transformed variables. This is an experimental step and would be part of the trial and error during the model selection process.

Figure 5.3.4: Transformed variables

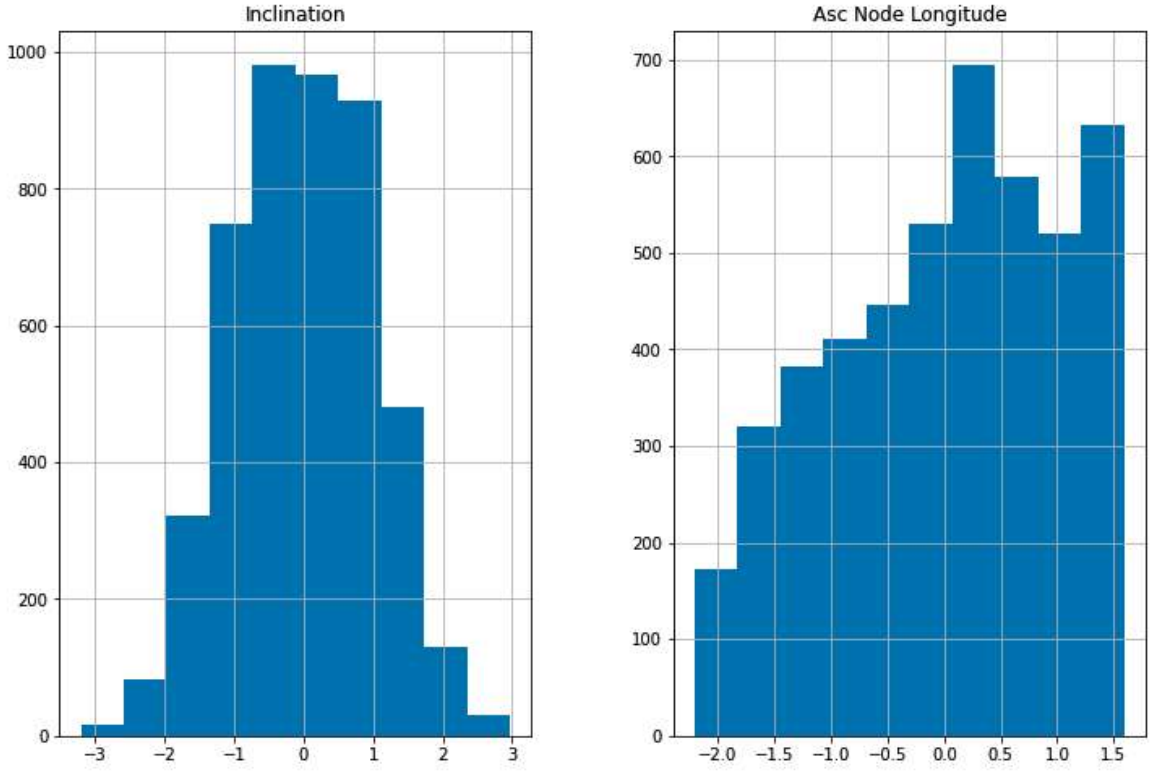
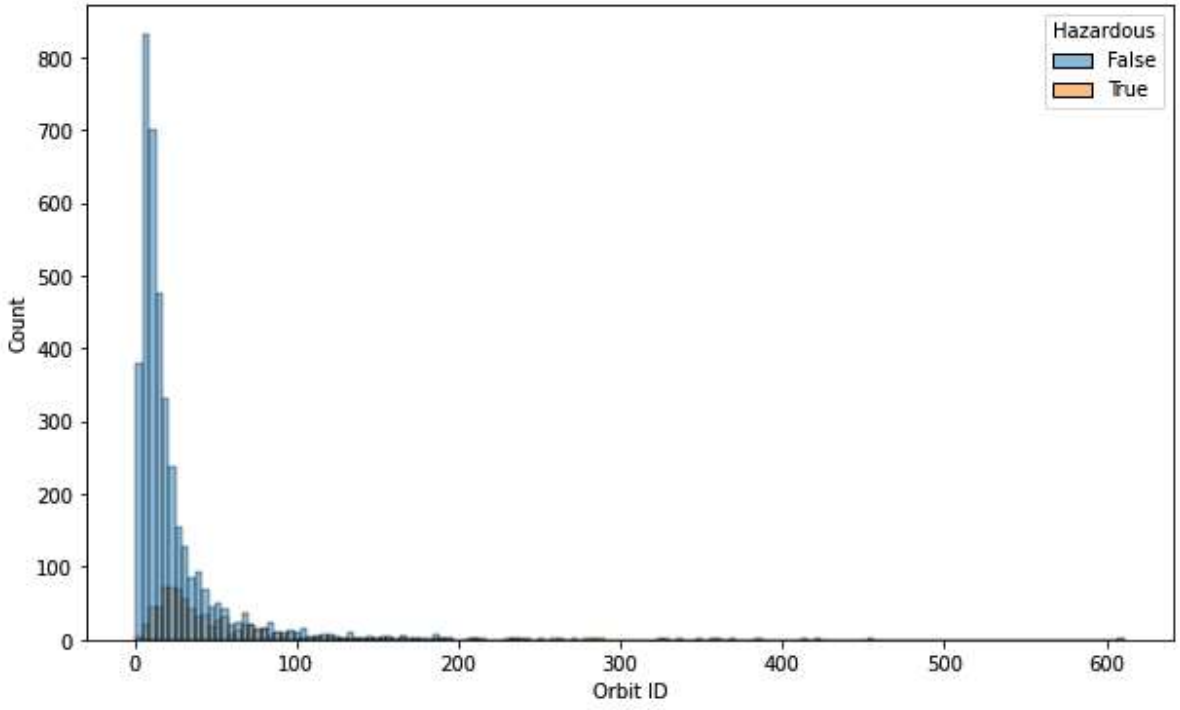


Figure 5.3.5: Histogram of Orbit ID Variable



As shown in figure 5.3.5 plotted histogram of counting of hazardous and non hazardous asteroids over the orbit ID. It is clearly seen that the nature for both hazardous and non hazardous asteroids are right skewed.

The above plot describes the distribution of Orbit ID which is a categorical variable. The distribution of this variable with respect to the target variable does not provide significant difference between each of the classes. Since the categorical variable is already in the Nominal encoding form, we are not required to perform any transformation on this variable.

As a final preprocessing step, we dropped the identifier columns that cannot be used in data modeling. Then we prepared the train and test data by splitting it at 85:15 ratio. We also built a pipeline with the initial step of Standard Scaling to standardize all the numerical variables.

5.4 Dimension Reduction:

To improve model performance, reduce computational load and avoid overfitting, it is necessary to perform correlation analysis on the data. Highly correlated variables do not add much value to the models while still increasing computational loads. To check correlations among variables. We need to plot a heatmap using correlation matrix.

Figure 2 is a constructed heatmap for all attributes present in the dataset, which we can see where more variables are correlated to each other. In order to be smoothening the data mining process it is required to reduce the variables which are highly correlated.

In the above plot there are about 8 attributes containing similar dimensions named "Estimated diameter" having different values considering units need to be reduced to 2 attributes. Also variables like "Relative Velocity" and "Miss distance" need to be reduced.

Figure 5.4.1: Heatmap of correlation analysis

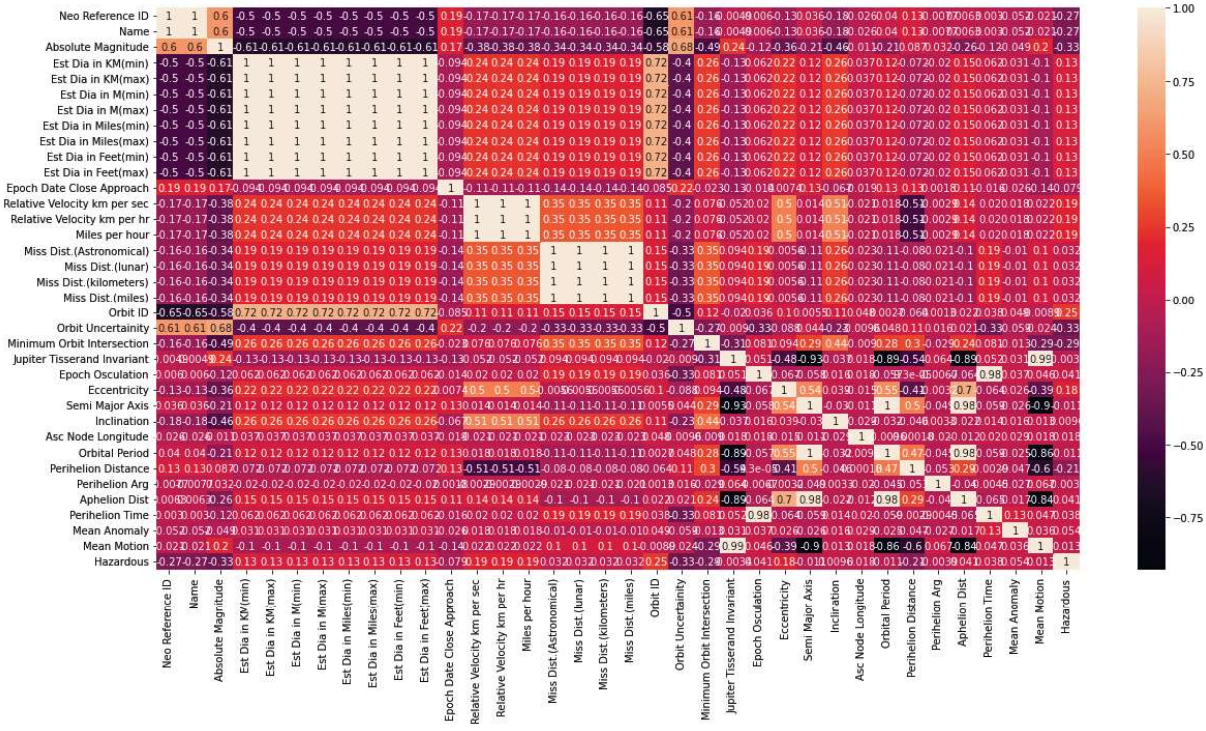
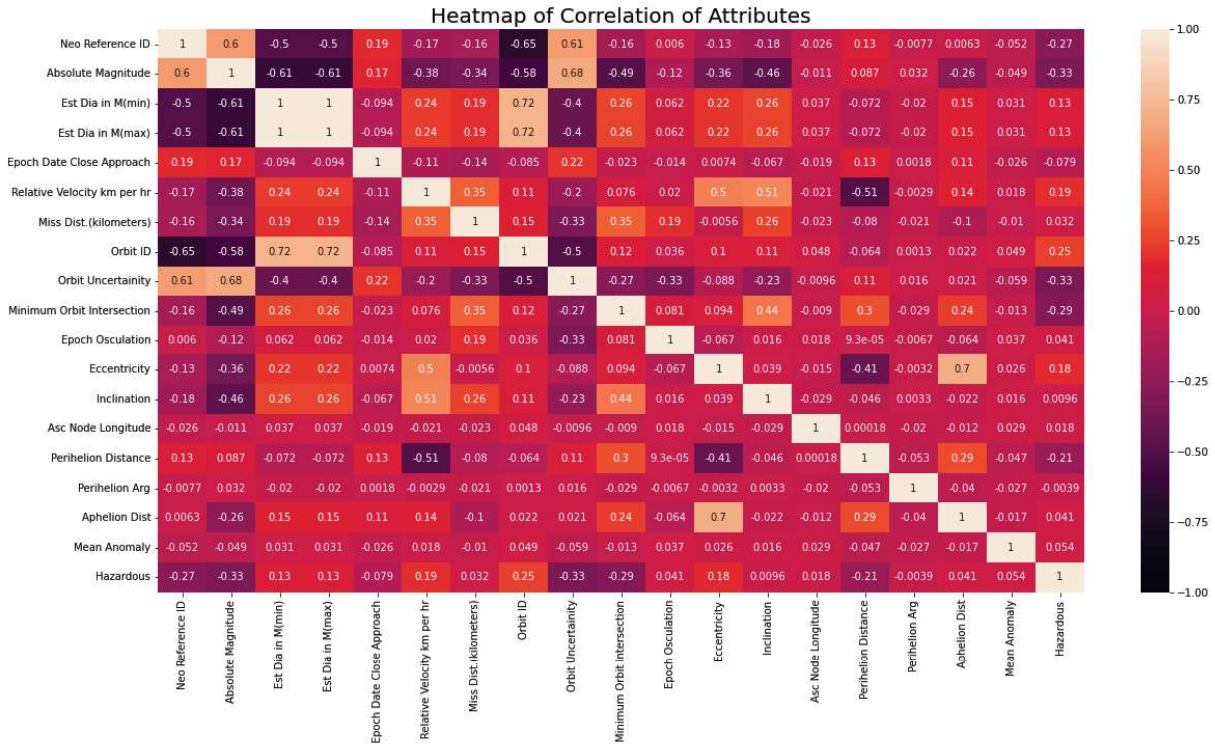


Figure 5.4.2: Heatmap after removing the redundant features



From the heatmap as shown in figure 3 of Heatmap plotted after removing duplicates and highly correlated variables. We have considered an absolute correlation threshold of 0.8 and removed the redundant features based on it but retained one feature out of the sets. A positive correlation indicates the extent to which those variables increase or decrease in parallel with each other while a negative correlation indicates the extent to which one variable increases as the other decreases. From the correlation matrix and plot above, we can clearly see that there are many features correlated to other features either positively or negatively. From the summary of the processing and exploring the dataset of asteroids, we can see the correlation of various attributes.

5.5 Data Partitioning:

To feed the dataset to the machine learning models, it needs to be separated as predictors and response variables. The independent predictor variables with indexes 0 to 8 were grouped and represented by the alias 'X' and the target variable 'Hazardous' with an alias 'y'. Now, to train the models and to evaluate them, the data needed to be partitioned. For this partitioning, we decided to go with an 80:20 ratio of split. The implication of it is that 80% of the dataset would go under the label of training data and remaining 20% under test data. After this partition, we have train_X, test_X, train_y and test_y with dimensions (3983, 17) and (704, 17) respectively.

5.6 Oversampling:

As discussed in the data exploration section, There are 3932 asteroids that are non-hazardous whereas 755 are hazardous. It can be observed that the given dataset is an imbalanced dataset which requires oversampling of data. This oversampling is to be done in a trial and error method which will be done during the model selection process. There is an imbalance in the data which needs to be handled. To achieve this, the success class in training datasets needs to be overpopulated to match the other class to improve predictive performance. To oversample the datasets, we took aid of Synthetic Minority Oversampling Technique, or SMOTE. SMOTE is an oversampling method in which artificial samples for the minority class are created. This approach aids in overcoming the problem of overfitting caused by random oversampling. It concentrates on the feature space in order to produce new examples by interpolating between positive instances which are close together.

The changes after employing SMOTE can be seen as below:

- Original dataset shape ({0: 3350, 1: 633})
- Oversampled dataset shape ({1: 3350, 0: 3350})

6. Data Mining Models/Methods:

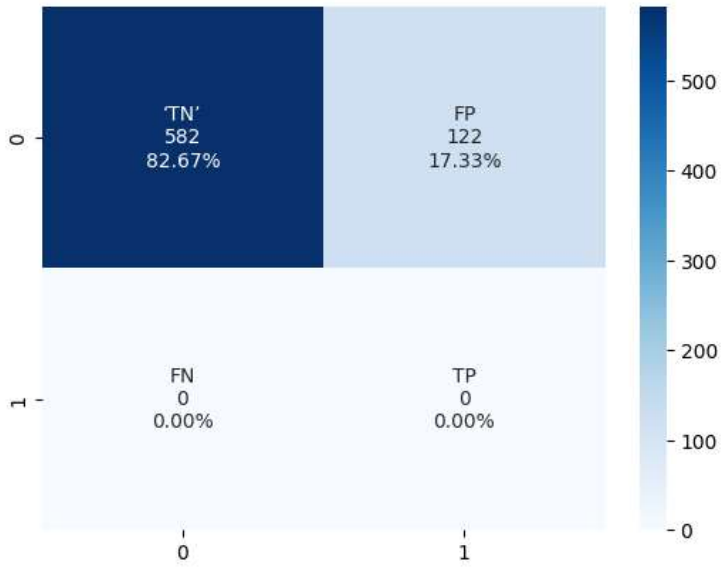
6.1 Logistic Regression:

The logistic model (also known as the logit model) is used to predict the likelihood of a particular class or event occurring. It extends the notions of linear regression, where the response variable is categorical. In its most basic form, the logistic model utilizes a logistic function to model a dependent variable

- Strengths:
  i) It's easy to implement, comprehend, and train.
  ii) It doesn't make any assumptions about class distributions in feature space.
  iii) It works well when the dataset is linearly separable and has good accuracy for many basic data sets.
- Limitations:
  i) The assumption of linearity between the dependent and independent variables is a key constraint of Logistic Regression.
  ii) Average or no multicollinearity between independent variables is required for logistic regression.

Implementation: For the base model, we used ridge regression with primal formulation and intercept scaling as 1.

Figure 6.1: Classification Matrix of Logistic Regression



6.2 Gaussian Naive Bayes:

A Naive Bayes classifier is a probabilistic machine learning model that's used for classification tasks. The crux of the classifier is based on the Bayes theorem.

- Strengths:
  i) Simple to implement
  ii) Very fast - As no iterations are performed and direct probabilities are computed
- Limitations:
  i) Requires conditional independence in the data
  ii) Not great for imbalanced data

Figure 6.2: Classification Matrix of Gaussian Naive Bayes

Figure 6.2 shows classification matrix for Gaussian Naive Bayes clearly seen that there are more misclassifications as True positive value is '0'. True Negative value is around 82%.

6.3 Stochastic Gradient Descent Classifier:

Stochastic Gradient Descent (SGD) is a simple yet efficient optimization algorithm used to find the values of parameters/coefficients of functions that minimize a cost function. Stochastic Gradient Descent (SGD) classifier basically implements a plain SGD learning routine supporting various loss functions and penalties for classification.

- Strengths:
  i) Memory efficient and can handle large dataset
  ii) Determined with hypter parameter loss
- Limitations:
  i) It requires more iterations to converge minimum
  ii) Choice of learning rate can be critical

Implementation: For the base model, we have used maximum iterations 10000 with random state 42.

Figure 6.3: Classification Matrix of SGD Classifier

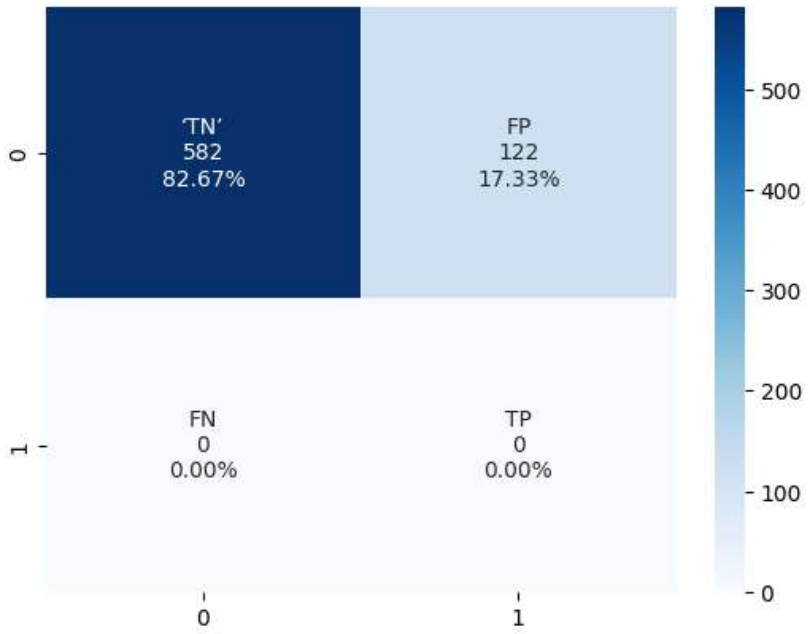


Figure 6.3 shows the classification matrix for Stochastic Gradient Descent clearly seen that there are more misclassifications as True positive value is '0', which has more or less performance as Gaussian Naive Bayes.

6.4 Support Vector Machine:

A support vector machine is a selective classifier formally defined by dividing the hyperplane. Given labeled training data the algorithm outputs the best hyperplane which classified new examples.

In two-dimensional space, this hyperplane is a line splitting a plane into two parts where each class lies on either side.

- Strengths:
  i) It is more productive in high-dimensional spaces
  ii) It is effective in instances, number of dimensions is larger than specimens
- Limitations:
  i) It requires more iterations to converge minimum
  ii) Choice of learning rate can be critical

Figure 6.4: Classification Matrix of SVM

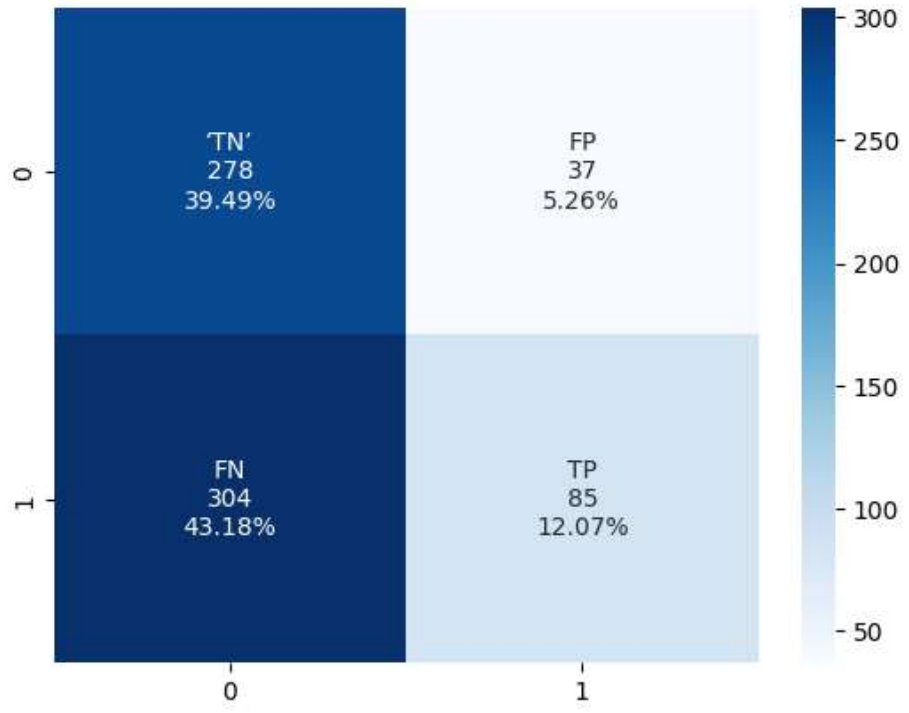


Figure 6.4 shows the classification matrix for Support Vector Machine classifies as True positive value is around 12% and False Negative as around 43%.

6.5 Decision Tree Classifier:

Decision Tree solves the problem of machine learning by transforming the data into tree representation. Each internal node of tree representation denotes an attribute and each leaf node denotes a class label.

- Strengths:
  i) Decision trees require less effort for data preparation during pre-processing.
  ii) Easy to explain to technical teams as well as stakeholders
- Limitations:
  i) Calculation can go far more complex
  ii) Decision tree training is relatively expensive

Figure 6.5: Classification Matrix of Decision Tree Classifier

Figure 6.5 shows the classification matrix of Decision Tree Classifier classifies as True positive value is around 17% and False Negative as around 0.5%. True Negative values trace around 82% which gives correctly classified non-hazardous asteroids.

6.6 Random Forest Classifier:

From a randomly selected portion of the training data, the random forest classifier generates a collection of the decision trees. It then combines the votes from several decision trees to determine the test object's final class.

- Strengths:
  iii) It helps to enhance the decision tree accuracy by reducing overfitting.
  iv) Good at handling missing values in model
  v) Handles nonlinear parameters efficiently
- Limitations:
  iii) It creates a lot of decision trees, therefore requires more computational power
  iv) Requires longer training periods

Implementation: For the base model, we used n_estimators or number of trees as 10, split quality measure as gini index, maximum depth as none.

Figure 6.6: Classification Matrix of Random Forest Classifier

Figure 6.6 shows the classification matrix for Random Forest Classifier classifies as True positive value is around 17% and False Negative as around 0.43% which contains only 3 records. It classifies 82.24% non-hazardous asteroids correctly.

7. **Performance Evaluation:**

We fitted the six models in their base form to get an idea about their performance.

a. Logistic Regression
b. Gaussian Naive Bayes
c. Stochastic Gradient Descent Classifier
d. Support Vector Machine
e. Decision Tree Classifier
f. Random Forest Classifier

Now we need to check the various classification models and calculate the metrics such as Accuracy, F1 score, precision, Recall, and ROC score(AUC).
The following table 7.1 shows the various evaluation scores predicted for the models we applied to our data.

Table 7.1: Comparison of the results of the models

	Model name	Accuracy	F1 Score	Precision	Recall	Roc_score
0	Logistic	0.826705	0.000000	0.000000	0.000000	0.500000
1	GNB	0.602273	0.333333	0.234899	0.573770	0.591009
2	SGDClass	0.826705	0.000000	0.000000	0.000000	0.500000
3	SVC	0.512784	0.333981	0.218830	0.704918	0.588713
4	DTClass	0.994318	0.983740	0.975806	0.991803	0.993324
5	RandomForest	0.995739	0.987854	0.976000	1.000000	0.997423

From the above table, we get the actual performance evaluation of all models. We clearly notice the values for various parameters including Accuracy, F1 score, Precision, recall, and ROC curve.

From the Classification matrix, it can be cleared the key performance characteristics of all applied models.

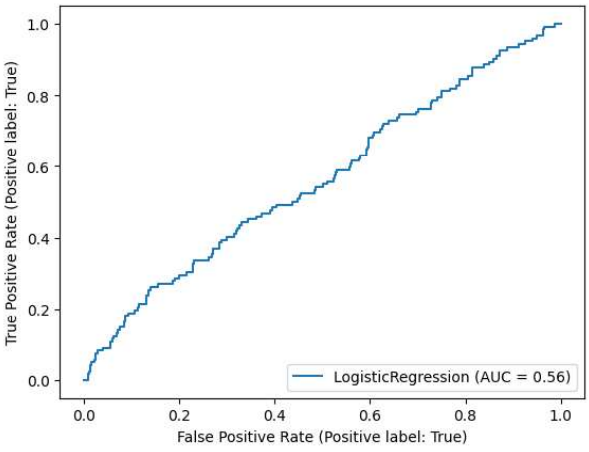
The confusion matrix helps us in analyzing and picking the best model with respect to our criteria. By viewing the amount and percentages in each cell of this matrix, you can quickly see how often the model predicted accurately.

Here, we wouldn't want to misclassify any hazardous asteroid to be non-hazardous because of the impact that it could cause. Therefore, we would prefer zero false negatives, which means that we would want to prefer the model with highest Recall. From the table and classification matrix it is clear that Decision Tree Classifier and Random Forest Classifier models have greater Recall value amongst the six applied models.

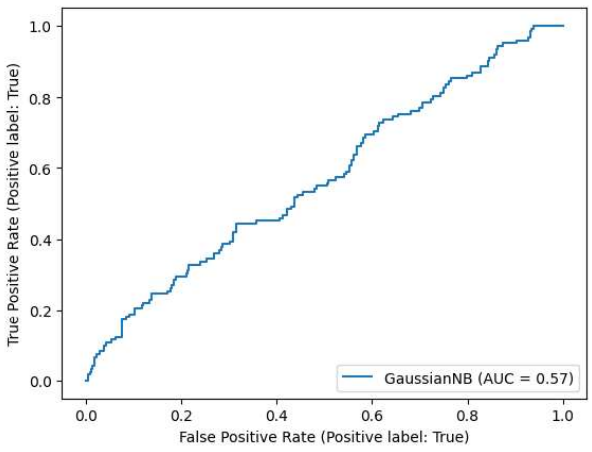
We plotted the ROC curve as displayed in figure 7.2 in order to calculate and visualize the characteristics using the library RocCurveDisplay.

Figure 7.1: ROC-AUC Matrix of models

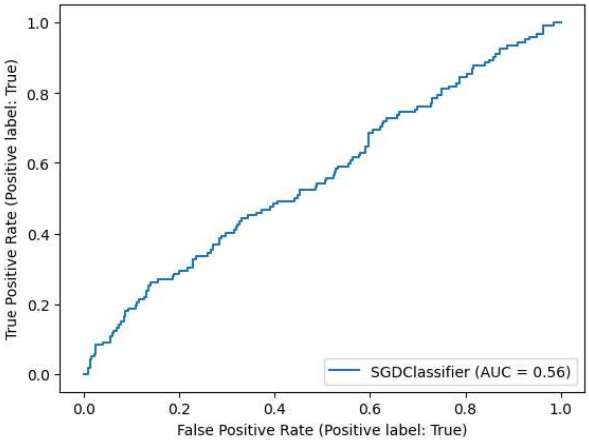
Logistic Regression



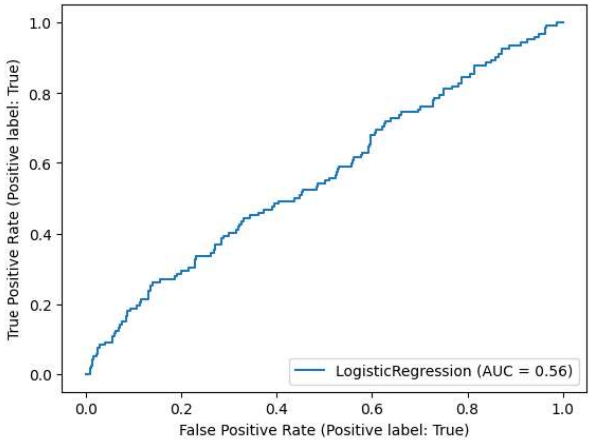
Gaussian Naive Bayes



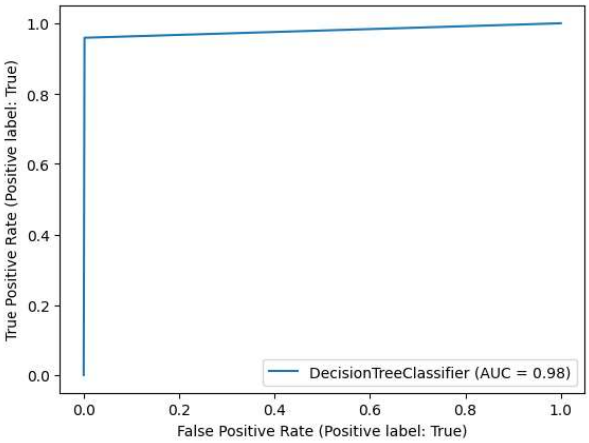
Stochastic Gradient Descent Classifier



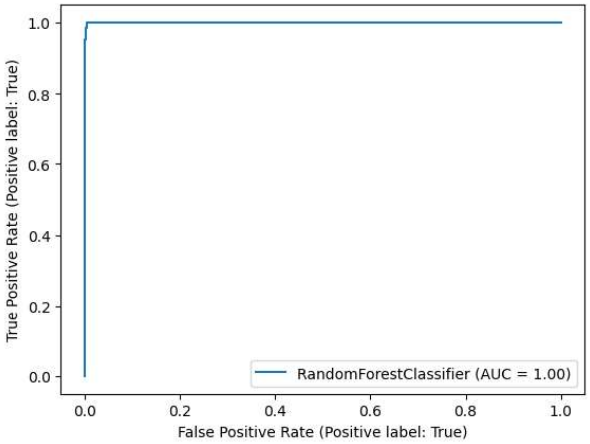
Support Vector Machine



Decision Tree Classifier



Random Forest Classifier



By observing Receiver Operating Characteristics curves, we can see that Decision Tree Classifier and Random Forest Classifier models are performing well. In between these models, Random Forest Classifier is attaining the ideal curve which we wanted with 1 Area Under Curve value.

In order to compare our targeted performance evaluation parameters, we need to plot the comparison chart for the applied models which will allow us to visualize more correctly and concisely the parameters.

Figure 7.2: Comparison of Performance of models

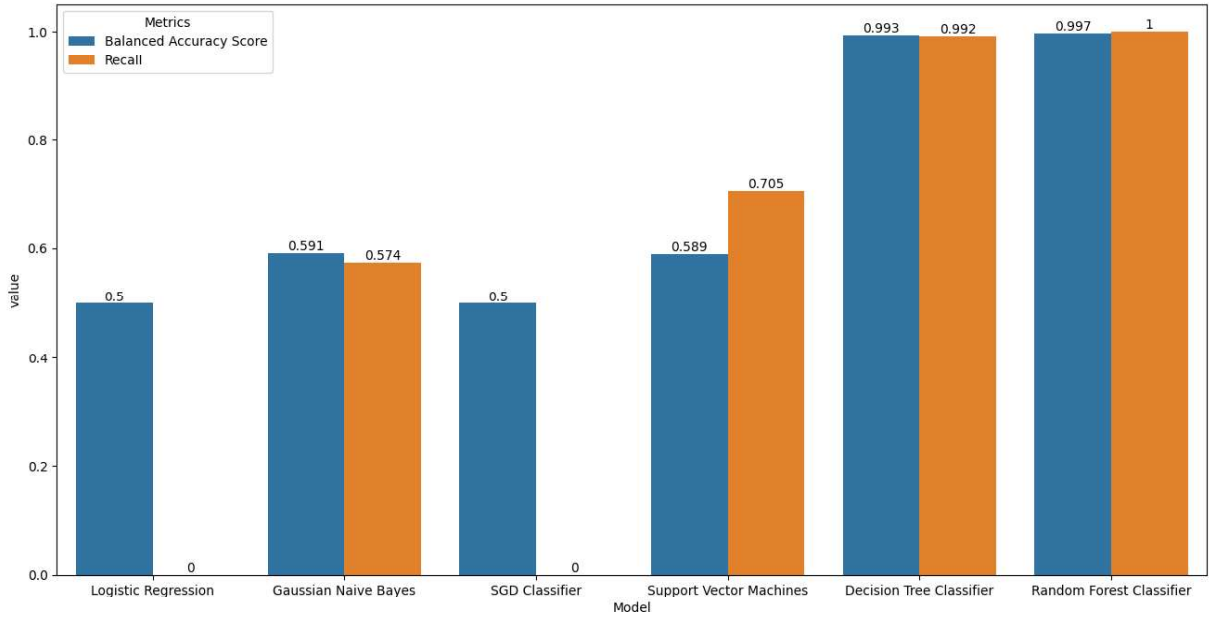


Figure 7.2 shows comparison of performance of all models we performed on the test data, We have reflected our focused matrices in the plot which are Balanced Accuracy Score and Recall of the models. As from the plot it is clearly seen that Decision Tree Classifier and Random Forest Classifier are performing well in comparison with all. In between two of these, Random Forest Classifier model is the best model with 99.7% accuracy and recall is 100%.

The confusion matrix and ROC curve helps us in analyzing and picking the best model with respect to our criteria. Here, we wouldn't want to misclassify any hazardous asteroid to be non-hazardous because of the impact that it could cause. Therefore, we would prefer zero false negatives, which means that we would want to prefer the model with highest Recall. Based on that, we can choose Random Forest Classifier to be the best model as it is the model with the highest recall.

8. **Project Results:**

All the models were trained on NASA's real open-ware dataset on asteroids that was split into train and test datasets.

a. Several classifier models were tried with different optimization (hyper) parameters to get the best Recall score and Accuracy on the test data.
b. Utilizing the model built, we can predict the type of asteroid which has a higher propensity to be hazardous and classify into categories hazardous or non-hazardous.
c. The models Decision Tree classifier that works on single Decision Trees and Random Forest Classifier that works on several Decision Trees produce very similar results in terms of metrics, the latter was chosen based on the ROC-AUC score.

9. **Impact of Project Outcomes:**

The overall objective of the project was to build a data driven model that could accurately classify if an asteroid is hazardous or non-hazardous. With the heuristics of errors that could occur due to unexplained variability, we decided to build a model that not only gets the highest accuracy but also scores the highest recall score. As true data miners and responsible creatures of mother Earth, we are determined to protect our planet from the hazardous asteroids. Thus, we went with maximizing the recall because we prioritized the question - “Of all hazardous asteroids, how many were correctly classified to be hazardous?”, in order not to miss out on even a single asteroid that could potentially cause severe damages to our planet. After trying out several models aforementioned, the answer to this question is observed to be the Random Forest Classifier. This model with optimization provides the best performance of 100% Recall and 99.7% Accuracy on the test data set.

10. References:

1. Center for NEO Studies. (n.d.). Center for NEO Studies. http://neo.jpl.nasa.gov/

2. NASA: Asteroids Classification. (n.d.). NASA: Asteroids Classification | Kaggle. https:///datasets/shrutimehta/nasa-asteroids-classification

3. Asteroids Glossary. (n.d.). Glossary. https://cneos.jpl.nasa.gov/glossary/PHA.html

4. L. (2023, March 24). What are potentially hazardous asteroids? livescience.com. https://www.livescience.com/what-are-potentially-hazardous-asteroids