Classification for Potentially Hazardous Asteroids

Spartificial Innovations Pvt. Ltd.

Jishnu N

Satyajit Sen

Shubhojit Sen

Dr. C. Kavitha

Table of Contents

ABSTRACT

As the amount of data and computing capacity continues to grow more accessible, the prevalence of incorporating Artificial Intelligence (AI) to tackle present-day issues also continues to expand. AI-based solutions are becoming vital to the field of space science. Asteroids, the minor planets of our solar system, have the propensity to cause significant damage to both humans and other life on Earth. Given the abundance of information on asteroid characteristics, it becomes a fertile territory to employ Machine Learning (ML) to identify the Potentially Hazardous Asteroids (PHAs) so as to diminish the likelihood of danger posed by them. The objective of this project is to contrast six different supervised ML algorithms to determine how efficacious they are at making correct identifications of PHAs. The Kaggle dataset 'NASA: Asteroids Classification' has been used for this endeavour and the algorithms Logistic Regression, K-Nearest Neighbours, Support Vector Machine, Random Forest, XGBoost, and Balanced Bagging have been chosen for the same. An evaluation of the proposed models was conducted using accuracy, precision, F1 score, recall. Overall, an accuracy of 99.86% was achieved with Random Forest.

1. INTRODUCTION

Machine Learning (ML) is an expansive concept that utilizes data to create analytics and inferences, which lends itself to being applied in predictive modelling. Over the past decade, the computing power and capacity to process large amounts of data has grown exponentially, offering researchers the possibility to implement ML in a variety of fields. The application of ML in the area of astronomy is the focus of this project. Astronomy produces large amounts of data through the use of sensors and other tools, which can be used in many different ways.

The authors suggest the use of various ML algorithms to detect Potentially Hazardous Asteroids (PHA). The subject of Near-Earth Objects (NEOs) and Near-Earth Asteroids is one of the most talked about in space science. They are an integral part to the comprehensive understanding of planetary systems and their features, including our solar system. Asteroids are rocky remnants, that currently orbit the sun, left over from the origination of our solar system that took place 4.6 billion years ago.[1] A significant majority of the asteroids in our solar system can be found congregating between the orbits of Mars and Jupiter in the doughnut-shaped ring aptly called the main asteroid belt.[1] Asteroids come in a multitude of sizes varying anywhere from hundreds of thousands of metres to the size of pebbles.[1] Planets have the capability to affect the paths of the asteroids by gravitationally influencing them, causing their trajectories to alter.[2] There's an appreciable degree of expert consensus that NEOs and NEAs, by the virtue of having smashed onto Earth in the past, happened to be a pivotal factor to the development of the biosphere that is found on our planet.[3]

The effects of a hazardous asteroid can be anywhere from thermal radiation and overpressure shock, to tsunamis and seismic shaking, to large scale extinction events altogether.[4] While it is true that only a minuscule proportion of NEOs approach our planet close enough to dispense harm in the first place, a sufficiently massive asteroid, however, would be detrimental enough to obliterate a vast region in its entirety.[4] Since the gravitational tug of the planets of our solar system can potentially result in any object's orbit developing into an Earth-crossing one over time[2], the risk of a collision with our planet in the future only arises with time.[4] A comprehensive purchase on the characteristics of such objects might help in determining the most suitable methods to reroute them if and when they find themselves on a path that's potentially threatening to Earth. These objects in question largely consist of PHAs. Should an asteroid that could cause harm be recognized before it can reach and hit Earth, it may be possible to avoid the crash entirely with an appropriate space mission either by altering the asteroid's course using a gravitational tugboat of some sort or by totally eliminating it using a nuclear weapon.[4]

The aim of this project is to recognize NEAs and NEOs that pose risks and classify them as hazardous or non-hazardous. A variety of machine learning algorithms have been employed to accomplish this objective. The various models were trained on data by NASA, after which the outcomes thereof were analyzed so as to determine the most accurate model that provided the most precise predictions.

2. LITERATURE

The literature study conducted in the course of this project is summarized in 'Table 1'.

Title	Author(s)	Concept used	Reference
An introduction to logistic regression analysis and reporting	Chao-Ying Joanne Peng, Kuk Lida Lee, and Gary M. Ingersoll	Logistic Regression	[5]
Nearest neighbour pattern classification	Thomas M. Cover and Peter E. Hart	K-Nearest Neighbours	[6]
Support-Vector Networks	Corinna Cortes and Vladimir Vapnik	Support Vector Machine	[7]
Random forest classifier for remote sensing classification.	Mahesh Pal	Random Forest	[8]
An effective adaptive customization framework for small manufacturing plants using extreme gradient boosting-XGBoost and random forest ensemble learning algorithms in an Industry 4.0 environment	Sonia K. Kiangala and Zenghui Wang	XGBoost	[9]
Bagging Predictors	Leo Breiman	Balanced Bagging	[10]

Table 1: Literature Study

2.1 *An introduction to logistic regression analysis and reporting* (2002) [5] by *Peng, Chao-Ying Joanne, Kuk Lida Lee, and Gary M. Ingersoll* provided us with a set of guidelines for using logistic regression techniques. It posited that the tables, figures, and charts should be included to comprehensively assess the results and for the assumptions that are verified to be discussed. It also demonstrated the preferred pattern for the application of logistic methods with an illustration of logistic regression applied to a dataset in testing a research hypothesis. Recommendations are also offered for appropriate reporting formats of logistic regression results and the minimum observation-to-predictor ratio. The authors of the paper also evaluated the use and interpretation of logistic regression presented in 8 articles published in The Journal of Educational Research between 1990 and 2000. They found that all 8 studies met or exceeded recommended criteria.[5]

2.2 *Nearest neighbour pattern classification* (1967) [6] by *Cover, Thomas M., and Peter E. Hart* acquainted us with the workings of the K- nearest neighbours algorithm. The nearest neighbour decision rule assigns to an unclassified sample point the classification of the nearest of a set of previously classified points. This rule is independent of the underlying joint distribution on the sample points and their classifications, and hence the probability of error R of such a rule must be at least as great as the Bayes probability of error R*--the minimum probability of error overall decision rules taking underlying probability structure into account. However, in a large sample analysis, it is shown in the M-category case that R* < R < R*(Z - MR*/(M-l)), where these bounds are the tightest possible, for all suitably smooth underlying distributions. Thus, for any number of categories, the probability of error of the nearest neighbour rule is bounded above by twice the Bayes probability of error. In this sense, it may be said that half the classification information in an infinite sample set is contained in the nearest neighbour.[6]

2.3 *Support-Vector Networks* (1995) [7] by *Cortes, Corinna, and Vladimir Vapnik* acquainted us with the workings of the support vector machine algorithm. The support-vector network is for two-group classification problems wherein the machine conceptually implements the following idea: input vectors are non-linearly mapped to a very high dimensional feature space. In this feature space, a linear decision surface is constructed. Special properties of the decision surface ensure high generalization ability of the learning machine. The idea behind the support-vector network was previously implemented for the restricted case where the training data can be separated without errors, this paper, however, extends this result to non-separable training data. The high generalization ability of support-vector networks utilizing polynomial input transformations is demonstrated. The comparison of the performance of the support-vector network to various classical learning algorithms that all took part in a benchmark study of Optical Character Recognition is also performed in this paper wherein the support-network algorithm exhibits a very fine performance in the comparison study.[7]

2.4 *Random forest classifier for remote sensing classification* (2005) [8] by Pal, Mahesh acquainted us with the working of the random forest classifying algorithm. The paper states that growing an ensemble of decision trees and allowing them to vote for the most popular class produced a significant increase in classification accuracy for land cover classification. The paper presents results obtained with the random forest classifier and compares its performance with the support vector machines (SVMs) in terms of classification accuracy, training time, and user-defined parameters. Landsat Enhanced Thematic Mapper Plus (ETM+) data of an area in the UK with seven different land covers were used. Results from this study suggest that the random forest classifier performs equally well as SVMs in terms of classification accuracy and training time. This study also concluded that the number of user-defined parameters required by random forest classifiers is less than that required b SVMs and also happen to be easier to define.[8]

2.5 *An effective adaptive customization framework for small manufacturing plants using extreme gradient boosting-XGBoost and random forest ensemble learning algorithms in an Industry 4.0 environment* (2021) [9] by *Kiangala, S.K. and Wang, Z.* acquainted us with the working of the XGBoost classifying algorithm. The paper states that the prevailing competitive manufacturing industry calls for continuous customer satisfaction for business sustainability. With the emergence of the Industry 4.0 paradigm, product customization, which gives customers the means to personalized products to meet their needs, has become a strategy to increase companies' value. High-tech manufacturing firms are already diving deep into Industry 4.0 standards, adopting innovative strategies to outstand themselves in the market, while small manufacturing plants are slow in embracing the digital transformation. The high cost involved in acquiring indispensable resources and the lack of expertise are some of the obstacles low tech businesses face in endorsing this new paradigm. Inspired by the customization challenges of a small manufacturing plant, this paper develops an effective adaptive customization platform that encodes the customization data history of a small manufacturing plant, from a static database, into a dynamic machine learning model to produce personalized products for their customers accurately. The research presented in the paper improves customers' experience by reducing the customization system's complexity consisting of inputting several parameters to obtain personalized products to a single entry. The backend of the platform uses powerful machine learning (ML) algorithms like extreme gradient boosting (XGBoost) and Random Forest (RF) ensemble learning to match a single customer input to the desired customized product category. The research experiments done by the authors of this paper convey insights, such as the best scenarios to use XGBoost over RF algorithms for regression problems with non-linear data. The excellent experimental results achieved on both machine learning models show the merits of this customization platform.[9]

2.6 *Bagging Predictors* (1996) [10] by *Breiman, Leo* acquainted us with the workings of the bagging classifying algorithms. Bagging predictors is a method for generating multiple versions of a predictor and using these to get an aggregated predictor. The aggregation averages over the versions when predicting a numerical outcome and does a plurality vote when predicting a class. The multiple versions are formed by making bootstrap replicates of the learning set and using these as new learning sets. Tests on real and simulated data sets using classification and regression trees and subset selection in linear regression show that bagging can give substantial gains in accuracy. The vital element is the instability of the prediction method. If perturbing the learning set can cause significant changes in the predictor constructed, then bagging can improve accuracy.[10]

3. MATERIALS & METHODS

The workflow of this project has been illustrated in 'Figure 1'.

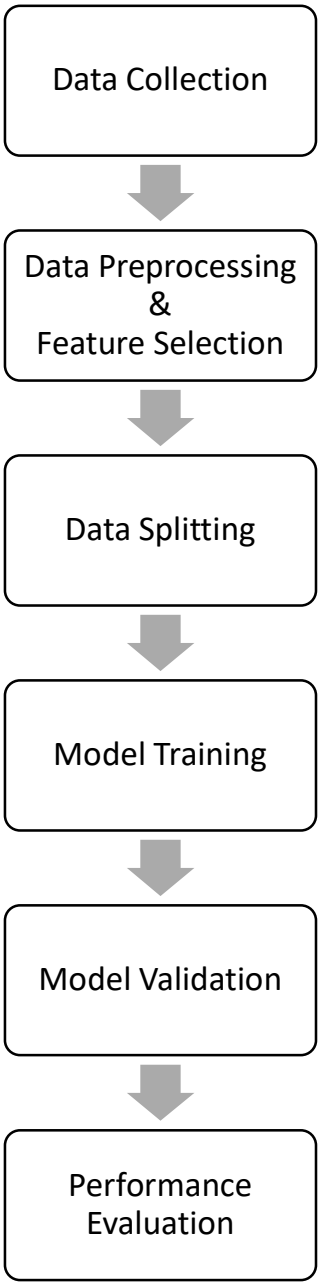

Figure 1: Workflow of the Project

3.1 *Dataset Description*

This project employed the public domain dataset 'NASA: Asteroids Classification' from the public data platform Kaggle. This data was originally sourced from the Center for Near Earth Object Studies, NASA Jet Propulsion Laboratory (JPL), California Institute of Technology, and reposed on Kaggle via the NASA API. The dataset has 4687 rows and 40 columns, as illustrated in 'Table 2'.

	Column Name	**Type**
0	Potentially Hazardous Flag	Output
1	Neo Reference ID	Input
2	Name	Input
3	Absolute Magnitude	Input
4	Est Dia in KM(min)	Input
5	Est Dia in KM(max)	Input
6	Est Dia in M(min)	Input
7	Est Dia in M(max)	Input
8	Est Dia in Miles(min)	Input
9	Est Dia in Miles(max)	Input
10	Est Dia in Feet(min)	Input
11	Est Dia in Feet(max)	Input
12	Close Approach Date	Input
13	Epoch Date Close Approach	Input
14	Relative Velocity km per sec	Input
15	Relative Velocity km per hr	Input
16	Miles per hour	Input
17	Miss Dist.(Astronomical)	Input
18	Miss Dist.(lunar)	Input
19	Miss Dist.(kilometers)	Input
20	Miss Dist.(miles)	Input
21	Orbiting Body	Input
22	Orbit ID	Input
23	Orbit Determination Date	Input
24	Orbit Uncertainity	Input
25	Minimum Orbit Intersection	Input
26	Jupiter Tisserand Invariant	Input
27	Epoch Osculation	Input
28	Eccentricity	Input
29	Semi Major Axis	Input
30	Inclination	Input
31	Asc Node Longitude	Input
32	Orbital Period	Input
33	Perihelion Distance	Input
34	Perihelion Arg	Input
35	Aphelion Dist	Input
36	Perihelion Time	Input
37	Mean Anomaly	Input
38	Mean Motion	Input
39	Equinox	Input

Table 2: Features in the dataset

The dataset was also found to be imbalanced, as illustrated in ‘Figure 2’.

[51]:
```
sns.set(font_scale=2.5)
sns.countplot(x="Hazardous",data=ast_data)
plt.show()
```

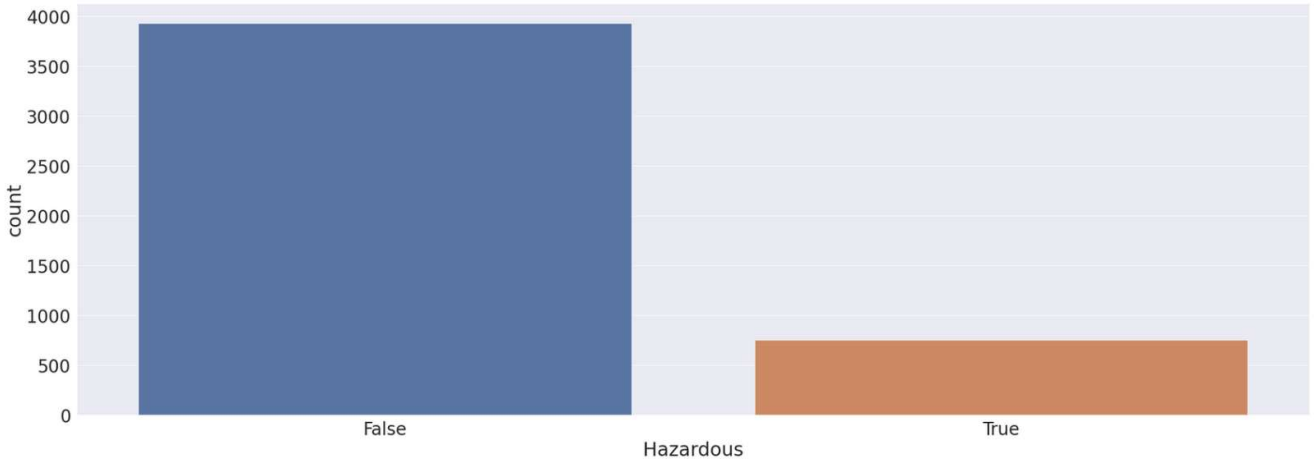

Figure 2: Dataset Imbalance

3.2 *Data Preprocessing & Feature Selection*

We checked if any of the feature values of the datapoints are null-valued. As illustrated in 'Figure 3', it was found that none of the datapoints have any null-valued feature values.

```
ast_data.isna().sum()>0
```

```
Neo Reference ID              False
Name                          False
Absolute Magnitude            False
Est Dia in KM(min)            False
Est Dia in KM(max)            False
Est Dia in M(min)             False
Est Dia in M(max)             False
Est Dia in Miles(min)         False
Est Dia in Miles(max)         False
Est Dia in Feet(min)          False
Est Dia in Feet(max)          False
Close Approach Date           False
Epoch Date Close Approach     False
Relative Velocity km per sec  False
Relative Velocity km per hr   False
Miles per hour                False
Miss Dist.(Astronomical)      False
Miss Dist.(lunar)             False
Miss Dist.(kilometers)        False
Miss Dist.(miles)             False
Orbiting Body                 False
Orbit ID                      False
Orbit Determination Date      False
Orbit Uncertainity            False
Minimum Orbit Intersection    False
Jupiter Tisserand Invariant   False
Epoch Osculation              False
Eccentricity                  False
Semi Major Axis               False
Inclination                   False
Asc Node Longitude            False
Orbital Period                False
Perihelion Distance           False
Perihelion Arg                False
Aphelion Dist                 False
Perihelion Time               False
Mean Anomaly                  False
Mean Motion                   False
Equinox                       False
Hazardous                     False
dtype: bool
```

Figure 3: The dataset has no null values

3.2.1 Features Dropped

a) *Neo Reference ID*
The feature 'Neo Reference ID' denotes an identification number given to the asteroids. This feature would not be useful for the machine learning model since an identification number of the asteroid is of no bearing on if it is hazardous. The column containing this feature was dropped.

b) *Name*
The feature 'Name' denotes a numeric name given to the asteroids. This feature would not be useful for the machine learning model since the name of the asteroid is of no bearing on if it is hazardous. The column containing this feature was dropped.

c) *Close Approach Date*
The feature 'Close Approach Date' corresponds to the UTC date when the asteroid was closest to Earth. The time at which an asteroid was closest to Earth is of no bearing on if it is hazardous. The column containing this feature was dropped.

d) *Epoch Date Close Approach*
The feature 'Close Approach Date' corresponds to the Unix Time date when the asteroid was closest to Earth. The time at which an asteroid was closest to Earth is of no bearing on if it is hazardous. The column containing this feature was dropped.

e) *Orbit ID*
The feature 'Orbit ID' denotes the identification number given to the orbits of the asteroids. This feature would not be useful for the machine learning model since the name of the orbit of the asteroid is of no bearing on if it is hazardous. The column containing this feature was dropped.

f) *Orbit Determination Date*
The feature 'Orbit Determination Date' corresponds to the date and time of when the orbit of the asteroid was determined. The time at which an asteroid's orbit was determined has no bearing on if it is hazardous. The column containing this feature was dropped.

g) *Orbiting Body*

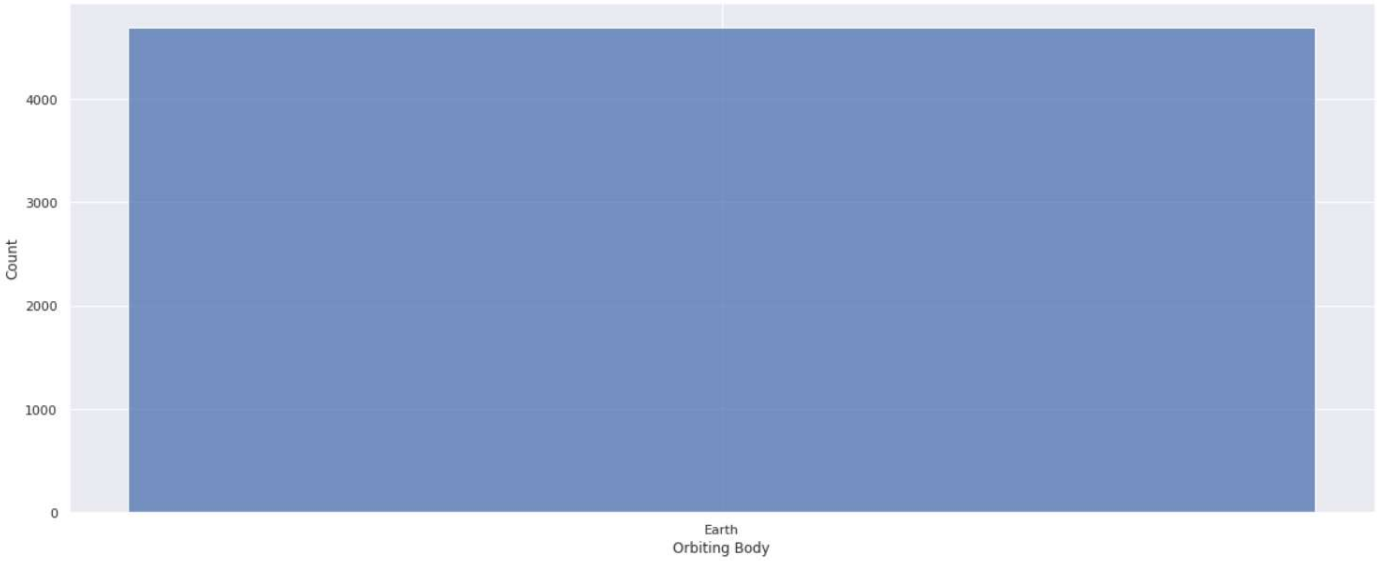

Figure 4: The histogram of the feature 'Orbiting Body'

As illustrated in 'Figure 4', the feature 'Orbiting Body' of every datapoint, i.e. every asteroid, has the value 'Earth'. This feature would not be useful for the machine learning model since all the datapoints in the dataset in question have the same feature value. The column containing this feature was dropped.

h) *Epoch Osculation*

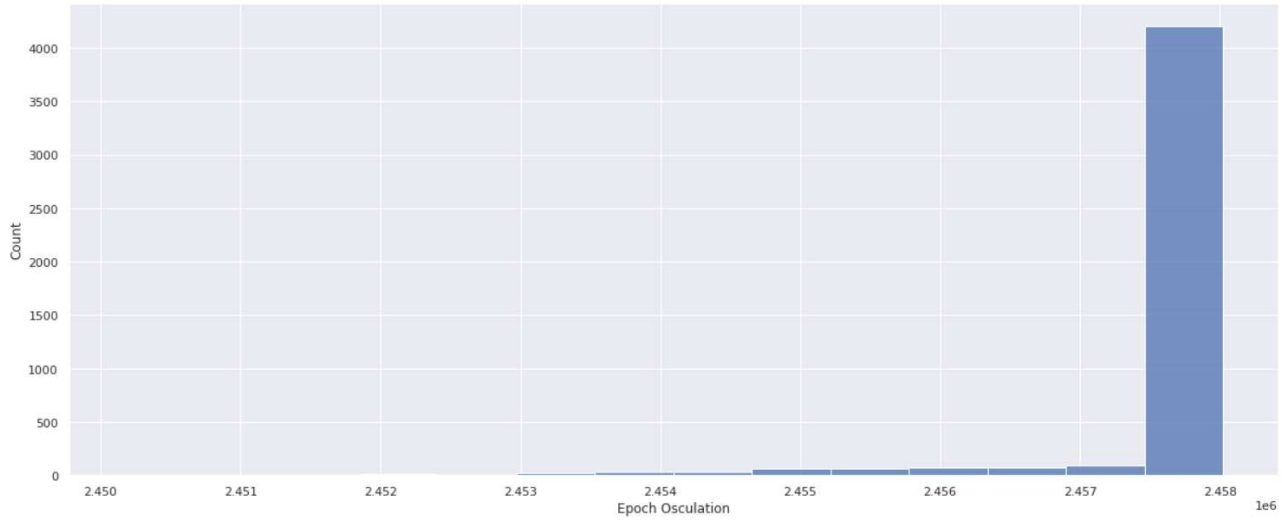

Figure 5: The histogram of the feature 'Epoch Osculation'

As illustrated in 'Figure 5', out of a dataset of 4687 datapoints, around 4500 datapoints have the same value for the feature 'Epoch Osculation'. This feature would not be useful to the machine learning model since it is not sufficiently varied. The column containing this feature was dropped.

i) *Equinox*

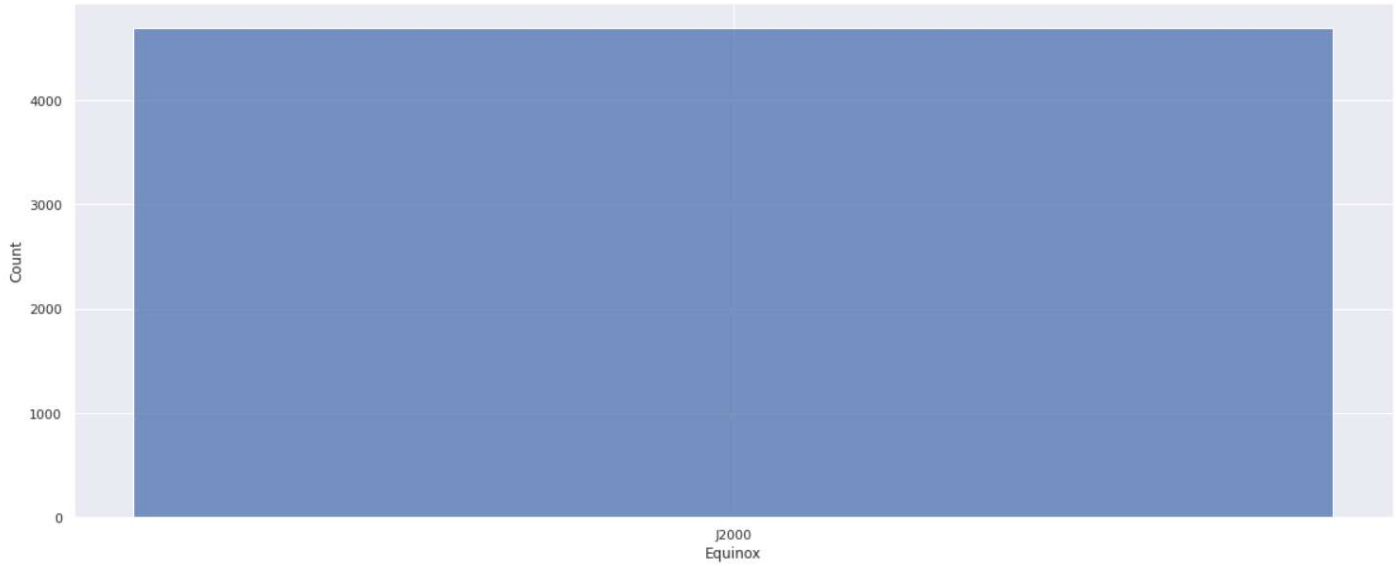

Figure 6: The histogram of the feature 'Equinox'

As illustrated in 'Figure 6', the feature 'Equinox' of every datapoint, i.e. every asteroid, has the value 'J2000'. This feature would not be useful for the machine learning model since all the datapoints in the dataset in question have the same feature value. The column containing this feature was dropped.

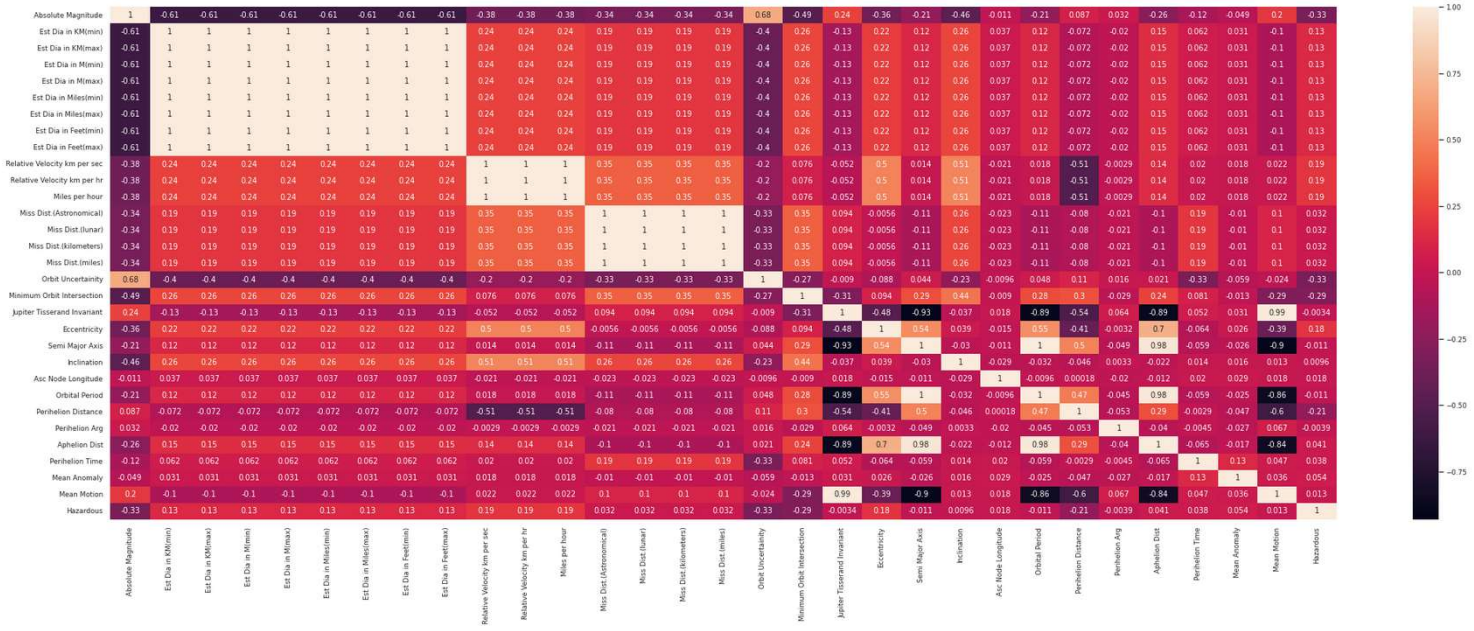

Figure 7: The correlation heatmap of the remaining features

j) *Est Dia in KM(min)*

As illustrated in 'Figure 7', the feature 'Est Dia in KM(max)' is highly positively correlated with the features 'Est Dia in KM(min)', 'Est Dia in M(min)', 'Est Dia in M(max)', 'Est Dia in Miles(min)', 'Est Dia in Miles(max)', 'Est Dia in Feet(min)', and 'Est Dia in Feet(max)'. Thus, the column containing the feature 'Est Dia in KM(min)' was dropped.

k) *Est Dia in M(min)*

As illustrated in 'Figure 7', the feature 'Est Dia in KM(max)' is highly positively correlated with the features 'Est Dia in KM(min)', 'Est Dia in M(min)', 'Est Dia in M(max)', 'Est Dia in Miles(min)', 'Est Dia in Miles(max)', 'Est Dia in Feet(min)', and 'Est Dia in Feet(max)'. Thus, the column containing the feature 'Est Dia in M(min)' was dropped.

l) *Est Dia in M(max)*

As illustrated in 'Figure 7', the feature 'Est Dia in KM(max)' is highly positively correlated with the features 'Est Dia in KM(min)', 'Est Dia in M(min)', 'Est Dia in M(max)', 'Est Dia in Miles(min)', 'Est Dia in Miles(max)', 'Est Dia in Feet(min)', and 'Est Dia in Feet(max)'. Thus, the column containing the feature 'Est Dia in M(max)' was dropped.

m) *Est Dia in Miles(min)*

As illustrated in 'Figure 7', the feature 'Est Dia in KM(max)' is highly positively correlated with the features 'Est Dia in KM(min)', 'Est Dia in M(min)', 'Est Dia in M(max)', 'Est Dia in Miles(min)', 'Est Dia in Miles(max)', 'Est Dia in Feet(min)', and 'Est Dia in Feet(max)'. Thus, the column containing the feature 'Est Dia in Miles(min)' was dropped.

n) *Est Dia in Miles(max)*

As illustrated in 'Figure 7', the feature 'Est Dia in KM(max)' is highly positively correlated with the features 'Est Dia in KM(min)', 'Est Dia in M(min)', 'Est Dia in M(max)', 'Est Dia in Miles(min)', 'Est Dia in Miles(max)', 'Est Dia in Feet(min)', and 'Est Dia in Feet(max)'. Thus, the column containing the feature 'Est Dia in Miles(max)' was dropped.

o) *Est Dia in Feet(min)*

As illustrated in 'Figure 7', the feature 'Est Dia in KM(max)' is highly positively correlated with the features 'Est Dia in KM(min)', 'Est Dia in M(min)', 'Est Dia in M(max)', 'Est Dia in Miles(min)', 'Est Dia in Miles(max)', 'Est Dia in Feet(min)', and 'Est Dia in Feet(max)'. Thus, the column containing the feature 'Est Dia in Feet(min)' was dropped.

p) *Est Dia in Feet(max)*

As illustrated in 'Figure 7', the feature 'Est Dia in KM(max)' is highly positively correlated with the features 'Est Dia in KM(min)', 'Est Dia in M(min)', 'Est Dia in M(max)', 'Est Dia in Miles(min)', 'Est Dia in Miles(max)', 'Est Dia in Feet(min)', and 'Est Dia in Feet(max)'. Thus, the column containing the feature 'Est Dia in Feet(max)' was dropped.

q) *Relative Velocity km per hr*

As illustrated in 'Figure 7', the feature 'Relative Velocity km per sec' is highly positively correlated with the features 'Relative Velocity km per hr' and 'Miles per hour'. Thus, the column containing the feature 'Relative Velocity km per hr' was dropped.

r) *Miles per hour*

As illustrated in 'Figure 7', the feature 'Relative Velocity km per sec' is highly positively correlated with the features 'Relative Velocity km per hr' and 'Miles per hour'. Thus, the column containing the feature 'Miles per hour' was dropped.

s) *Miss Dist.(lunar)*

As illustrated in 'Figure 7', the feature 'Miss Dist.(Astronomical)' is highly positively correlated with the features 'Miss Dist.(lunar)', 'Miss Dist.(kilometers)', and 'Miss Dist.(miles)'. Thus, the column containing the feature 'Miss Dist.(lunar)' was dropped.

t) *Miss Dist.(kilometers)*

As illustrated in 'Figure 7', the feature 'Miss Dist.(Astronomical)' is highly positively correlated with the features 'Miss Dist.(lunar)', 'Miss Dist.(kilometers)', and 'Miss Dist.(miles)'. Thus, the column containing the feature 'Miss Dist.(kilometers)' was dropped.

u) *Miss Dist.(miles)*

As illustrated in 'Figure 7', the feature 'Miss Dist.(Astronomical)' is highly positively correlated with the features 'Miss Dist.(lunar)', 'Miss Dist.(kilometers)', and 'Miss Dist.(miles)'. Thus, the column containing the feature 'Miss Dist.(miles)' was dropped.

v) *Aphelion Dist*

As illustrated in 'Figure 7', the feature 'Semi Major Axis' is highly positively correlated with the features 'Aphelion Dist' and 'Orbital Period'. Thus, the column containing the feature 'Aphelion Dist' was dropped.

w) *Orbital Period*

As illustrated in 'Figure 7', the feature 'Semi Major Axis' is highly positively correlated with the features 'Aphelion Dist' and 'Orbital Period'. Thus, the column containing the feature 'Orbital Period' was dropped.

x) *Mean Motion*

As illustrated in 'Figure 7', the feature 'Jupiter Tisserand Invariant' is highly positively correlated with the feature 'Mean Motion'. Thus, the column containing the feature 'Mean Motion' was dropped.

y) *Semi Major Axis*

As illustrated in 'Figure 7', the feature 'Jupiter Tisserand Invariant' is highly negatively correlated with the feature 'Semi Major Axis'. Hence, the feature 'Semi Major Axis' was dropped.

The original dataset with 40 features was reduced to a dataset with 15 features via feature reduction. The remaining features have been illustrated in 'Table 3'.

	Column Name	**Type**
1	Potentially Hazardous Flag	Output
2	Absolute Magnitude	Input
3	Est Dia in KM(max)	Input
4	Relative Velocity km per sec	Input
5	Miss Dist.(Astronomical)	Input
6	Orbit Uncertainity	Input
7	Minimum Orbit Intersection	Input
8	Jupiter Tisserand Invariant	Input
9	Eccentricity	Input
10	Inclination	Input
11	Asc Node Longitude	Input
12	Perihelion Distance	Input
13	Perihelion Arg	Input
14	Perihelion Time	Input
15	Mean Anomaly	Input

Table 3: Remaining features in the dataset after feature reduction

3.3 *Data Splitting*

The dataset with the remaining features was split into two sets: the train set and the test (validation) set in the ratio 8:2.

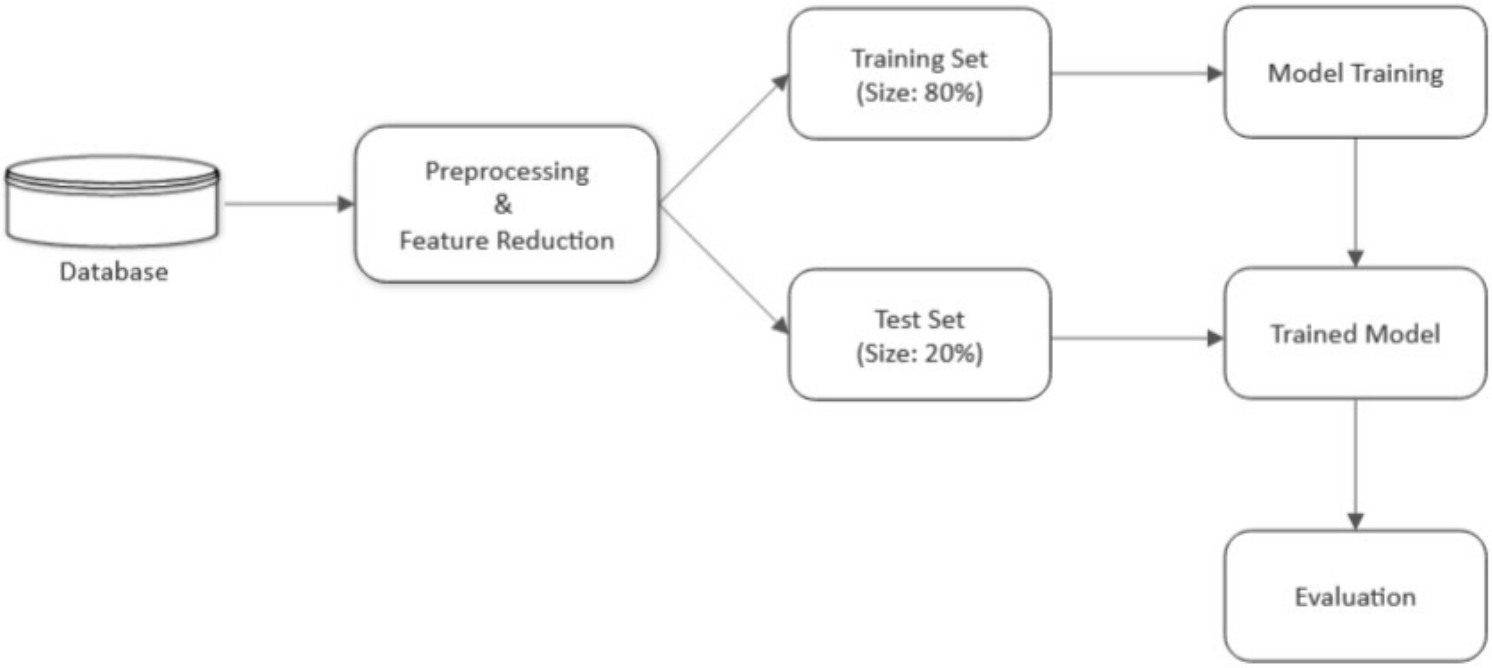

Figure 8: Architecture of the proposed models

3.4 *Machine Learning Classification*

The goal of this project is to compare the performance of various classification algorithms at predicting if an asteroid with some given parameters is potentially hazardous. The classification algorithms used in this study are Logistic Regression, K-Nearest Neighbours Classifier, Support Vector Machine Classifier, Random Forest Classifier, XGBoost Classifier, and Balanced Bagging Classifier.

3.5 *Evaluation Metrics*

In classification problems, a confusion matrix is a mathematical representation of the performance of a model that is used to showcase the obtained results for the model in question. For a binary classification problem, such as the one tackled in this project, there are two classes namely "Positive" (Hazardous Asteroid) and "Negative" (Non-Hazardous Asteroid), which entails a confusion of four elements in a 2×2 matrix. True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are the four elements in question. True Positive (TP) refers to the number of observations that are actually positive and are also predicted as such by the model. Similarly, True Negative (TN) refers to the number of observations that are actually negative and are also predicted as such by the model. False Positive (FP) refers to the number of observations that are actually negative but are predicted as positive by the model. Similarly, False Negative (FP) refers to the number of observations that are actually as positive but are predicted as negative by the model. Accuracy, Precision, Recall, and F1-Score are the metrics that have been utilized in this project to evaluate the performance of the models.

1. Accuracy: It is the overall performance of the model and is defined as the sum of TP and TN divided by the total number of observations.

$$\frac{(\text{True Positives} + \text{True Negatives})}{\begin{pmatrix}\text{True Positives} + \\ \text{True Negatives} + \\ \text{False Positives} + \\ \text{False Negatives}\end{pmatrix}} = \frac{\begin{pmatrix}\text{Number of} \\ \text{correct predictions}\end{pmatrix}}{\begin{pmatrix}\text{Number of} \\ \text{all predictions}\end{pmatrix}} = \frac{\begin{pmatrix}\text{Number of} \\ \text{correct predictions}\end{pmatrix}}{(\text{Size of dataset})}$$

2. Precision: It is the ratio of TP to the sum of TP and FP.

$$\frac{(\text{True Positives})}{\begin{pmatrix}\text{True Positives} + \\ \text{False Positives}\end{pmatrix}} = \frac{\begin{pmatrix}\text{Number of} \\ \text{correctly predicted} \\ \text{positive instances}\end{pmatrix}}{\begin{pmatrix}\text{Total number of} \\ \text{positive predictions} \\ \text{made by the model}\end{pmatrix}} = \frac{\begin{pmatrix}\text{Number of} \\ \text{asteroids correctly} \\ \text{predicted as} \\ \text{potentially hazardous}\end{pmatrix}}{\begin{pmatrix}\text{Total number of} \\ \text{asteroids predicted as} \\ \text{potentially hazardous} \\ \text{by the model}\end{pmatrix}}$$

3. Recall: It is the ratio of TP to the sum of TP and FN.

$$\frac{(\text{True Positives})}{\begin{pmatrix}\text{True Positives} + \\ \text{False Negatives}\end{pmatrix}} = \frac{\begin{pmatrix}\text{Number of} \\ \text{correctly predicted} \\ \text{positive instances}\end{pmatrix}}{\begin{pmatrix}\text{Total number of} \\ \text{positive instances} \\ \text{in the dataset}\end{pmatrix}} = \frac{\begin{pmatrix}\text{Number of} \\ \text{asteroids correctly} \\ \text{predicted as} \\ \text{potentially hazardous}\end{pmatrix}}{\begin{pmatrix}\text{Total number of} \\ \text{potentially hazardous} \\ \text{asteroids in the} \\ \text{dataset}\end{pmatrix}}$$

4. F1-Score: It is the harmonic mean of Precision and Recall.

$$\frac{1}{\left(\frac{(\text{Precision}^{-1}) + (\text{Recall}^{-1})}{2}\right)} = \frac{1}{\left(\frac{(\text{Recall} + \text{Precision})}{2 \times (\text{Precision} \times \text{Recall})}\right)} = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Recall} + \text{Precision})}$$

4. RESULTS

4.01 The performance of each implemented model is illustrated in 'Table 4'.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	91.9%	81.65%	61.38%	70.08%
K- Nearest Neighbours	89.98%	81.93%	46.26%	59.13%
Support Vector Machine	97.01%	92.09%	88.28%	90.14%
Random Forest	99.86%	99.54%	99.54%	99.54%
XGBoost	99.64%	98.2%	99.54%	98.87%
Balanced Bagging	99.25%	96.05%	99.32%	97.66%

Table 4: Performance of each implemented model

4.02 The confusion matrix of the model based on Logistic Regression is illustrated in 'Figure 9'.

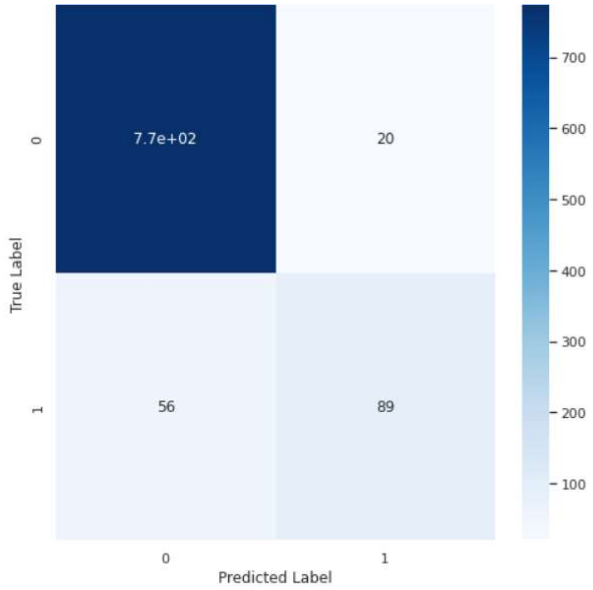

Figure 9: The confusion matrix of the Logistic Regression model

4.03 The confusion matrix of the model based on K-Nearest Neighbours is illustrated in 'Figure 10'.

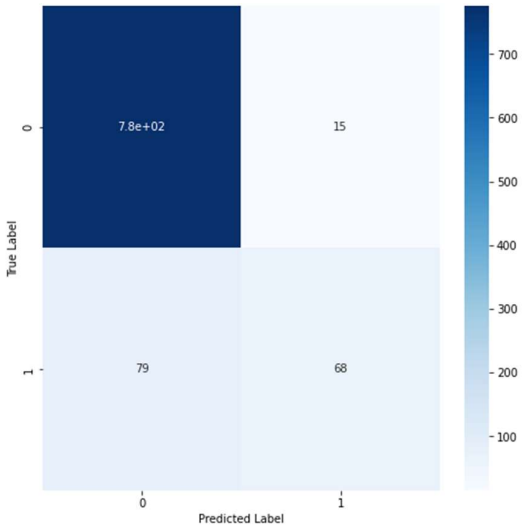

Figure 10: The confusion matrix of the K-Nearest Neighbours model

4.04 The confusion matrix of the model based on Support Vector Machine is illustrated in 'Figure 11'.

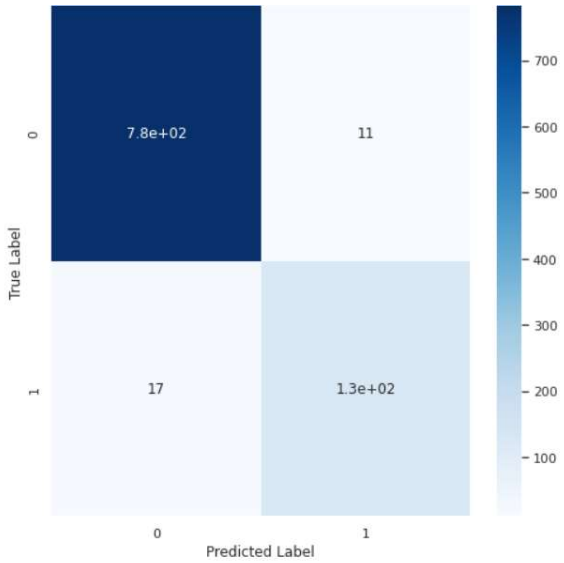

Figure 11: The confusion matrix of the Support Vector Machine model

4.05 The confusion matrix of the model based on Random Forest is illustrated in 'Figure 12'.

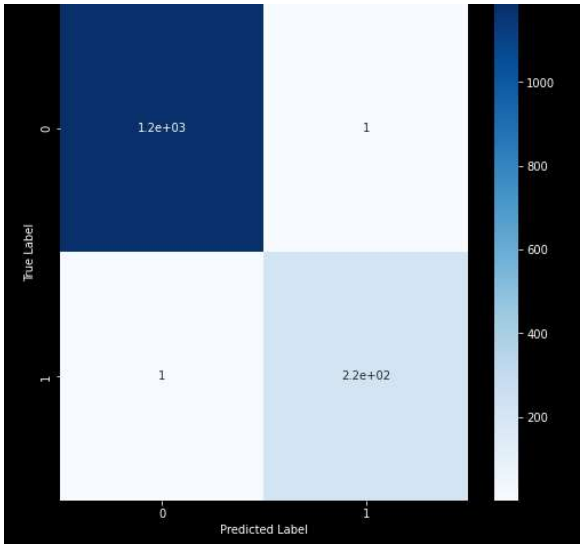

Figure 12: The confusion matrix of the Random Forest model

4.06 The confusion matrix of the model based on XGBoost is illustrated in 'Figure 13'.

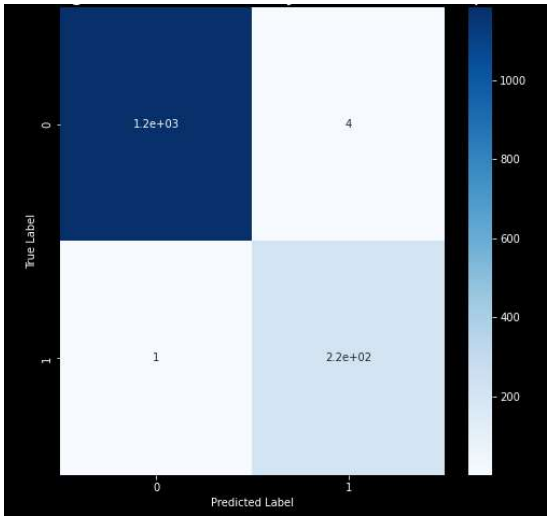

Figure 13: The confusion matrix of the XGBoost model

4.07 The confusion matrix of the model based on Balanced Bagging is illustrated in 'Figure 14'.

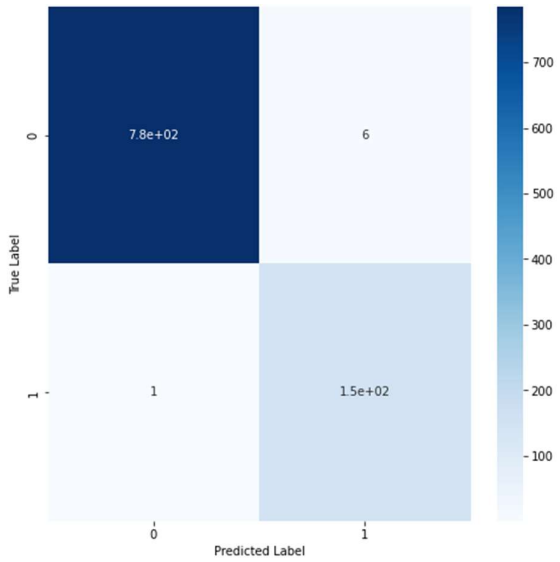

Figure 14: The confusion matrix of the Balanced Bagging model

4.08 The accuracy score of each model is graphed in 'Figure 15'.

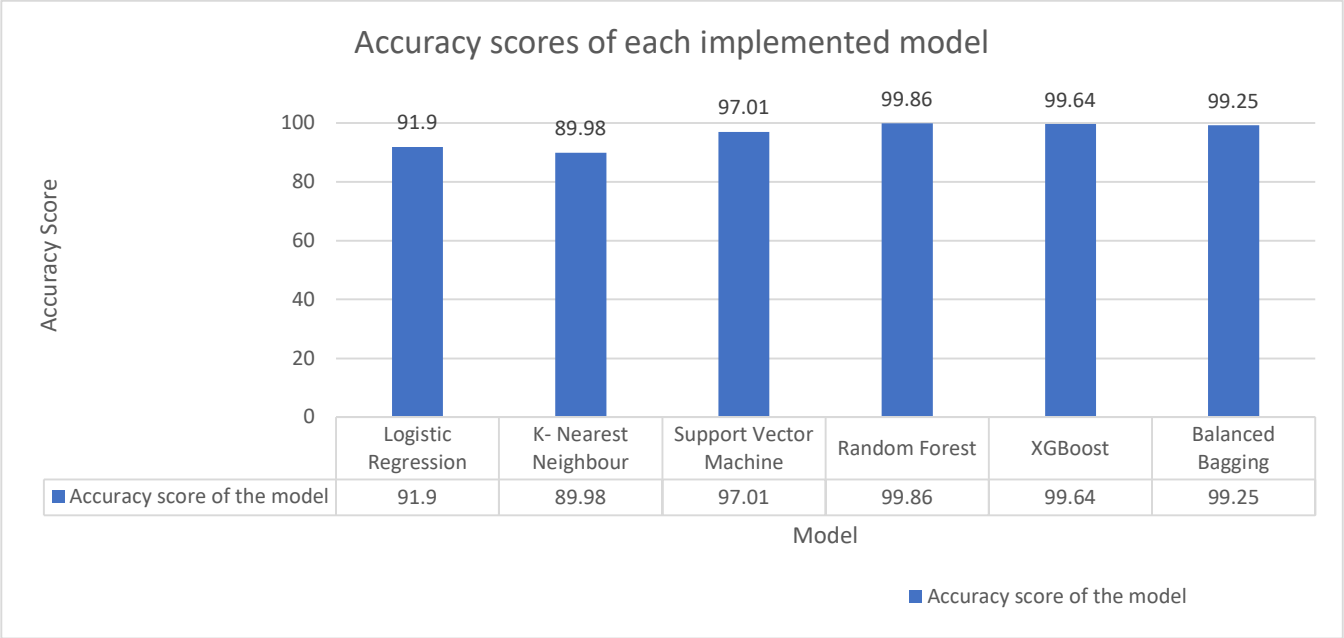

Figure 15: Accuracy scores of each implemented model

4.09 The precision score of each model is graphed in 'Figure 16'.

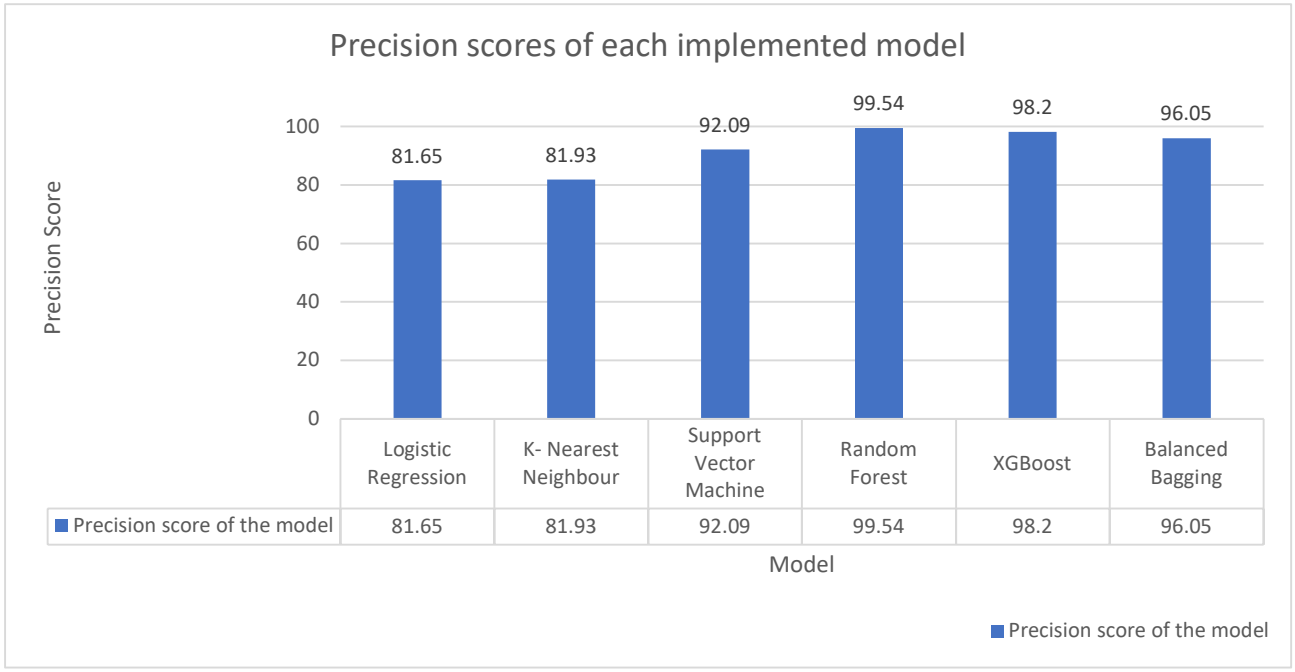

Figure 16: Precision scores of each implemented model

4.10 The recall score of each model is graphed in 'Figure 17'.

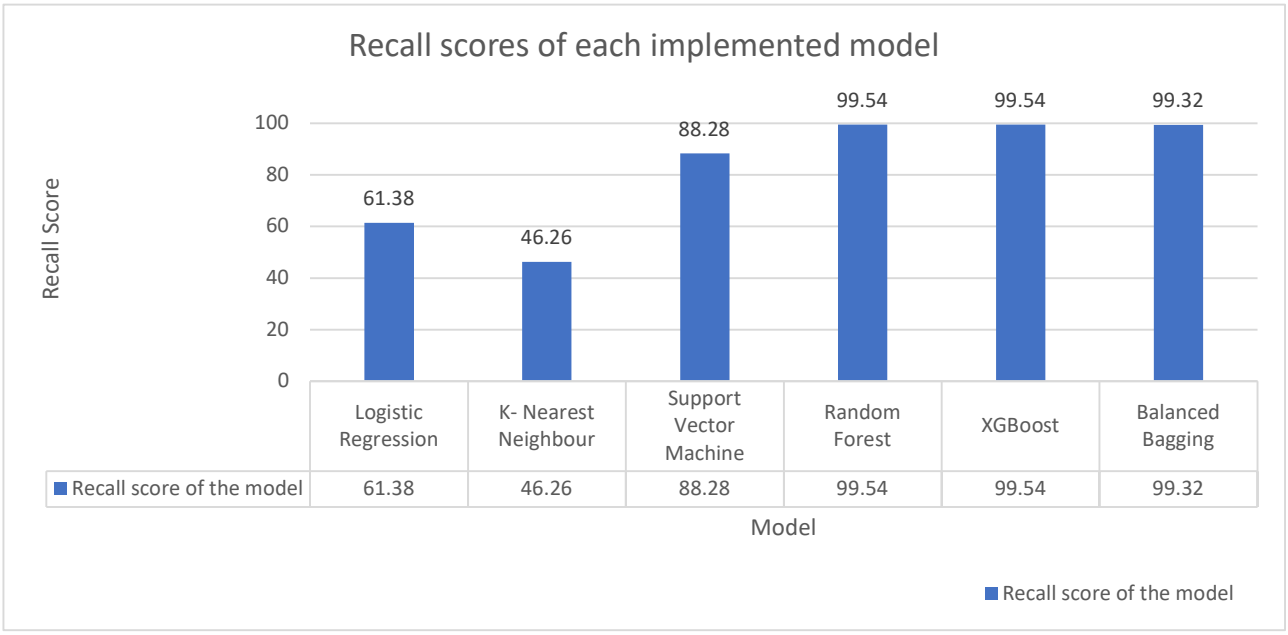

Figure 17: Recall scores of each implemented model

4.11 The F1-score of each model is graphed in 'Figure 18'.

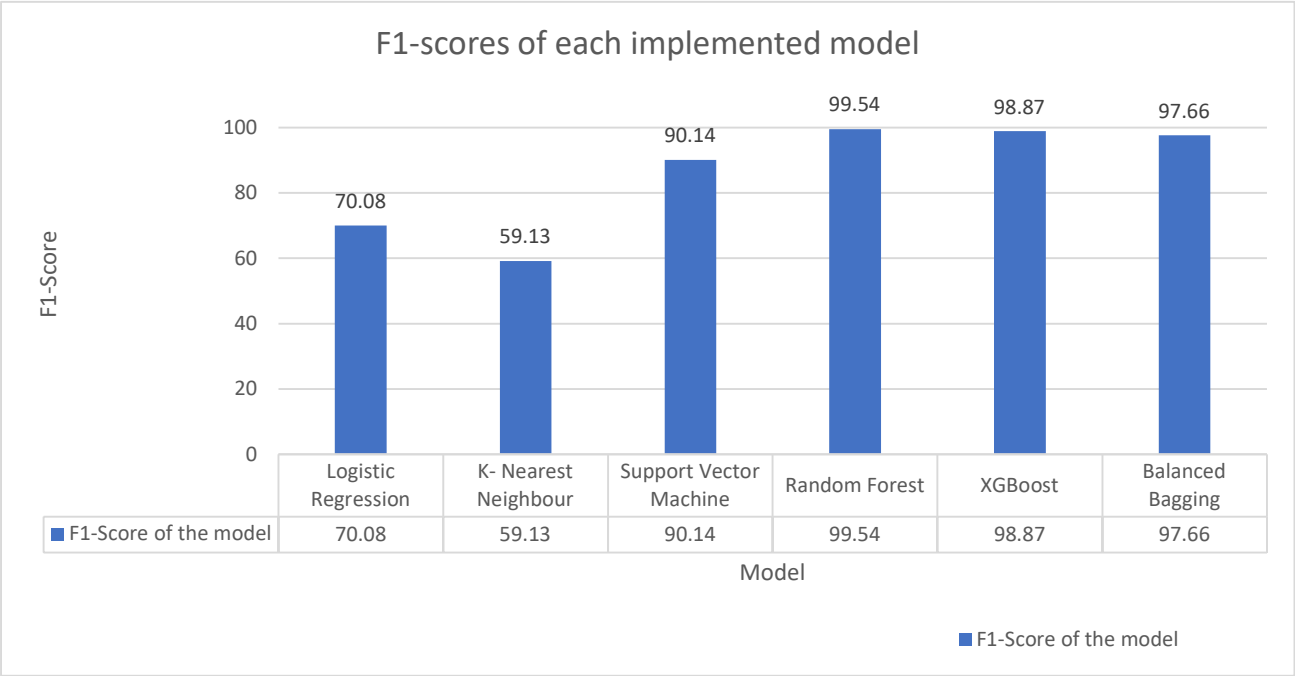

Figure 18: F1-scores of each implemented model

5. DISCUSSION

Most of the models were without any hyperparameters except for the models based on the K-Nearest Neighbours, Support Vector Machine, and Random Forest.

The only hyperparameter in the K-Nearest Neighbours model was *n* which signified the number of neighbours. An increase in the KNN score of the model was observed whilst changing the value of *n* from 1 to 4. The highest KNN score was observed at *n*=4. The model exhibited a significant decrease in the KNN score for *n*=5 that continued to fractionally worsen for 5<*n*<7. The KNN score stabilized at 7<*n*<10 while still being a lot worse than the one at *n*=4. After *n*=10 and subsequent values for *n*, the KNN score continued to significantly worsen further still.

The Support Vector Machine model had two hyperparameters, namely *C* and *Kernel*. *C* represents the regularization parameter which wherein the strength of the regularization is inversely proportional to the argument given. It only accepts positive arguments. *Kernel* represents the choice of kernel for the model. The possible kernel options for our problem were *linear*, *poly*, *rbf*, and *sigmoid*. While the model gave varying results for values of *C* ranging from 1 to 50 with respect to every choice of kernel, the best possible result was obtained using *C*=1 and *kernel='poly'*.

The model based on Random Forest has a hyperparameter *n_estimator* which represents the number of trees.

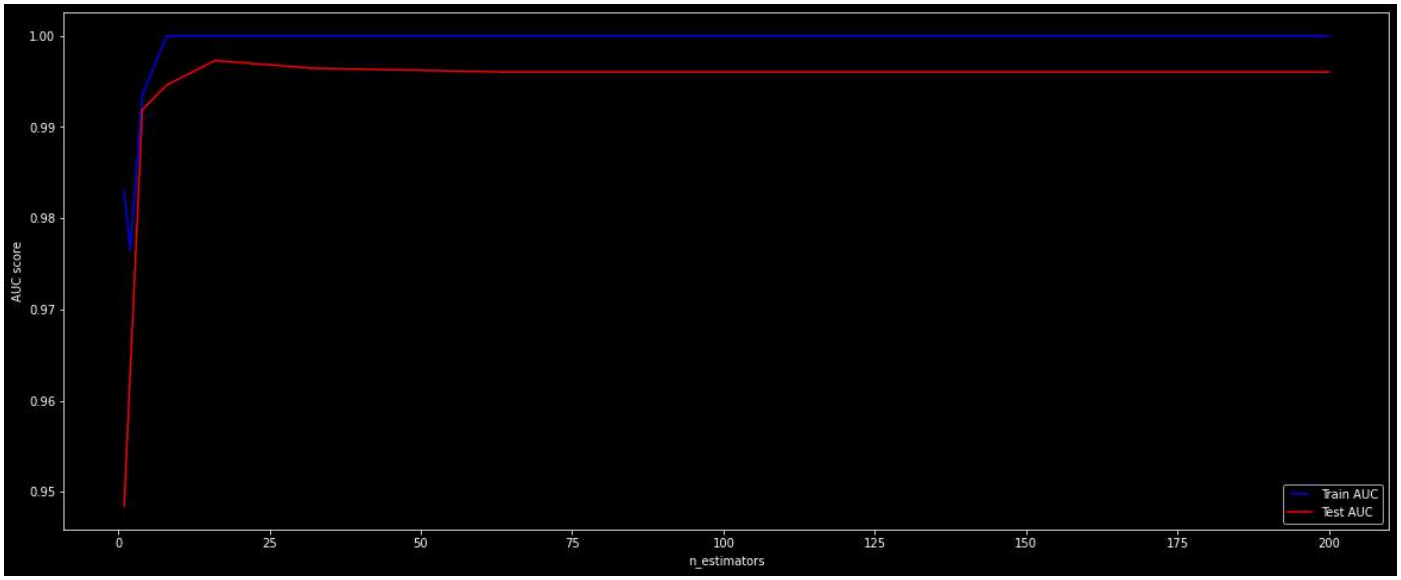

Figure 19: Train AUC score v/s Test AUC score for the Random Forest model

The Area Under Curve score (AUC score) is the metric inherent to the Random Forest algorithm to evaluate the performance of the model against the value of *n_estimator* being used. Typically, the higher the value of *n_estimator*, i.e. the higher the number of trees used, the better the ability of the model to understand the data. However, the training time is directly proportional to the number of trees being used. Thus, it was imperative to find the right value for *n_estimator* wherein said trade-off was a practical one. It was observed that the performance of the model degraded for *n_estimator*>32. The model also got a higher AUC score with *n_estimator*=32 than with *n_estimator*=16. Therefore, *n_estimator*=32 was finalized by the authors. Plots for the other parameters such as *max*_depth, *min_samples_split*, *max_features*, etc of the Random Forest model showed that underfitting and overfitting have been successfully avoided.

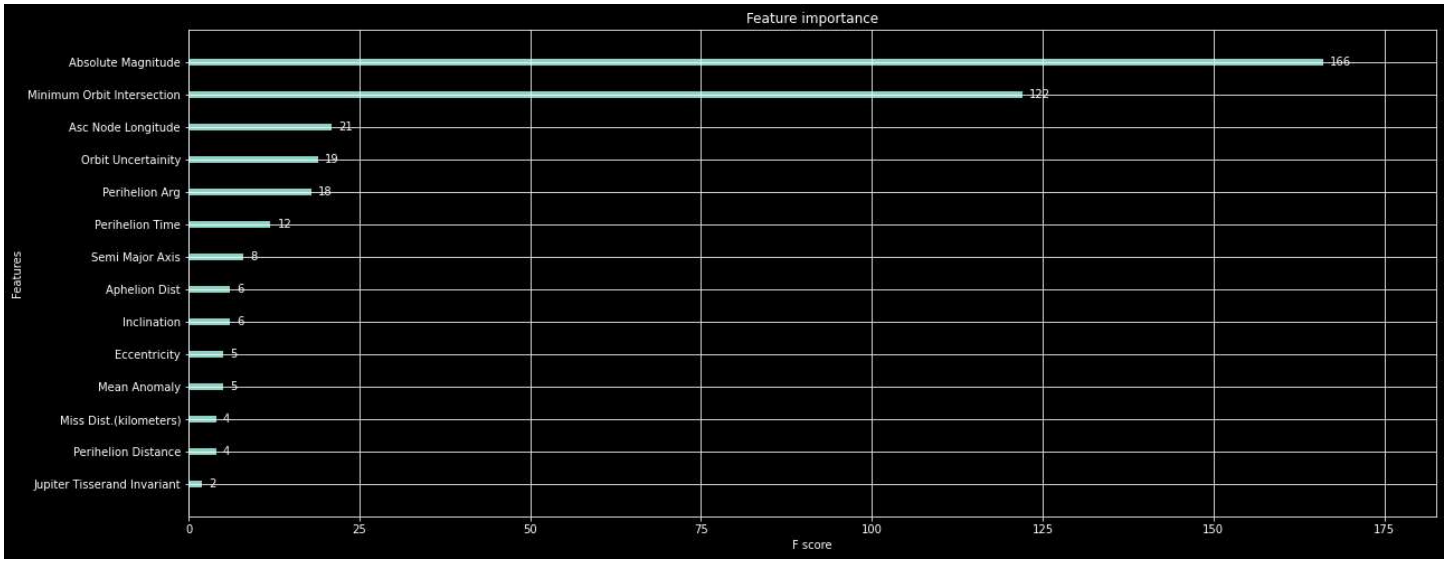

Figure 20: Feature importance plot

As it can be inferred from the results, the model based on the K-Nearest Neighbours algorithm performed the worst of all implemented models. The likeliest reason for this to happen is the significant imbalance in the data. The model based on logistic regression performed considerably well given that it's one of the simpler linear machine learning algorithms. Unsurprisingly, the models based on ensemble learning outperformed the other models in every evaluation metric employed with the model based on Random Forest outperforming every other model implemented by the authors in every evaluation metric used.

6. CONCLUSION

Classification algorithms are one of the most prominent facets in machine learning and data science. The main focus of this project was to use multiple supervised learning classification algorithms for the identification of Potentially Hazardous Asteroids (PHAs) through asteroid characteristics. Logistic Regression, K- Nearest Neighbours, Support Vector Machine, Random Forest, XGBoost, Balanced Bagging were used in this study.

Some key observations:

- Accuracy alone is not a good representation of model performance, especially for classification problems wherein the dataset is considerably imbalanced.

- Random Forest classification algorithm gave the greatest overall performance with respect to every performance metric employed in comparison to the other algorithms implemented by the authors in this study.

In the future, the authors aim to extend this study to include the overall analysis of asteroids with respect to additional characteristics thereof such as location, composition, etc. That shall have a higher scope for the domain of studies focused on asteroids and for space science as a whole.

REFERENCES

[1] “Asteroids”. NASA. 11 January 2023.

[2] “NEO Basics – Introduction”. Center for Near Earth Object Studies, NASA/JPL. 11 January 2023.

[3] “NEO Basics – Life on Earth”. Center for Near Earth Object Studies, NASA/JPL. 11 January 2023.

[4] “NEO Basics – Target Earth”. Center for Near Earth Object Studies, NASA/JPL. 11 January 2023.

[5] Peng, Chao-Ying Joanne, Kuk Lida Lee, and Gary M. Ingersoll. “An Introduction to Logistic Regression Analysis and Reporting.” The Journal of Educational Research, 96, (2002).

[6] Cover, Thomas M., and Peter E. Hart. “Nearest Neighbor Pattern Classification.” IEEE Transactions on Information Theory, 13, (1967).

[7] Cortes, Corinna, and Vladimir Vapnik. “Support-Vector Networks.” Machine Learning, 20, (1995).

[8] Pal, Mahesh. “Random forest classifier for remote sensing classification.” International Journal of Remote Sensing, 26, (2005).

[9] Kiangala, S.K., and Wang, Z. “An effective adaptive customization framework for small manufacturing plants using extreme gradient boosting-XGBoost and random forest ensemble learning algorithms in an Industry 4.0 environment.” Machine Learning with Applications, 4, (2021).

[10] Breiman, Leo. “Bagging Predictors.” Machine Learning, 24, (1996).