Preciction of Hazardous and non-Hazardous Asteroids

Anwesa Roy
Machine Learning Engineer Intern
AI Technology and Systems
Kolkata, India
royanwesa15official@gmail.com

***Abstract*— In this project, we used machine-learning techniques to predict whether an asteroid is potentially hazardous or non-hazardous. With a dataset of 4688 asteroids containing features like estimated diameter, relative velocity, orbit uncertainty, etc we attempt to predict the hazardous and non-hazardous asteroids. In particular, we compare different machine learning techniques such as KNN, Logistic Regression, and decision tree analysis. Keywords—hazardous and non-hazardous asteroids, machine learning, KNN, Logistic Regression, Decision Tree.**

I. Introduction

Asteroids are minor planets, especially of the inner Solar System. Larger asteroids have also been called planetoids. These terms have historically been applied to any astronomical object orbiting the Sun that did not resemble a planet-like disc and was not observed to have characteristics of an active comet such as a tail. There are millions of asteroids, ranging in size from hundreds of miles to several feet across. In total, the mass of all the asteroids is less than that of Earth's moon. Despite their size, asteroids can be dangerous. Many have hit Earth in the past, and more will crash into our planet in the future. That's one reason scientists study asteroids and are eager to learn more about their numbers, orbits and physical characteristics. Cosmic impact has the potential to eliminate humankind as we know it. Therefore, it is critical for us to systematically assess the magnitude of this threat. The atmospheric, geological, and biological effects of cosmic impact have become apparent only since the early 1980s, when the Cretaceous-Tertiary extinction was first linked to the impact of a 10-km asteroid. Even much smaller impactors still possess enormous energies and may cause local to regional devastation. It is with this intention that we intended to identify potentially hazardous asteroids using machine learning Techniques.

II. Data Preprocessing

A. Dataset

The data we used for our project was provided on the Kaggle website. We were given 4688 asteroid samples and their associated labels of whether the asteroid is hazardous or non-hazardous. For each asteroid, we were given its Neo Reference ID, Name, Absolute Magnitude, estimated diameters in various units, Close Approach Date, Epoch Date Close Approach, relative velocity in various units, miss distance in various units, orbiting Body, Orbit ID, Orbit Determination Date, Orbit Uncertainty, Minimum Orbit Intersection, Jupiter Tisserand Invariant, Epoch Osculation, Eccentricity, Semi Major Axis, Inclination, Asc Node Longitude, Orbital Period, Perihelion Distance, Perihelion Arg, Aphelion Dist, Perihelion Time, Mean Anomaly, Mean Motion, and Equinox. The dataset is complete, meaning that none of the entries were missing or null.

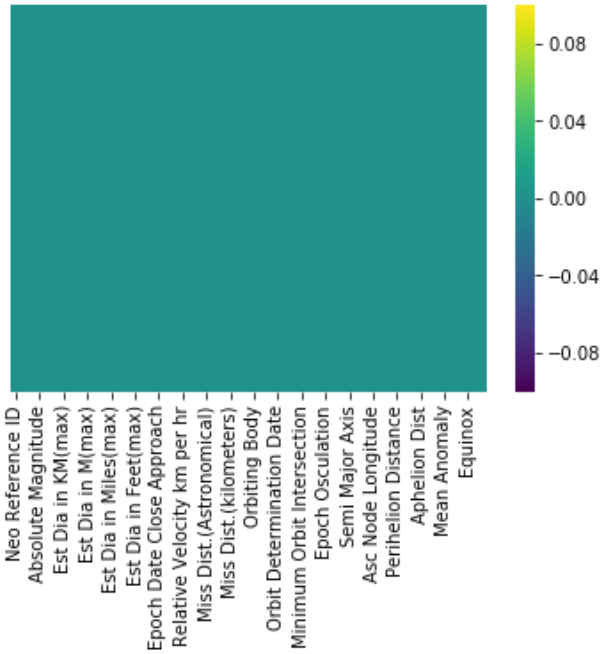

Figure 1. *Finding Null or Missing Values*

B. Feature Selection

There are about 40 features in the dataset including the target variable. Our attempt is to reduce the number of features in the dataset so as to reduce over fitting, improve accuracy, and improve training time.

1) Claculate the number of feature which have zero standard deviation: There are certain features in the dataset which have zero or very less standard deviation. Since these features don't vary, they will have no impact on model performance. There are a few categorical features which are constant throughout the dataset. We have removed the features having zero standard deviation.

2) Remove the features having high correlation: We drew the heatmap of all the features and observed that there were certain features in the dataset, which had high correlation amongst themselves. For instance, features such as estimated diameters, relative velocity were provided in different units. They are basically measures of same parameters and convey the same information and thus, can be removed.

3) Determining important features via data visualization: We plotted the distplots for various features

based on the hue whether an asteroid is hazardous or non-hazardous. There were certain features for which the difference between the mean and the overall distribution were remarkably visible. We considered those data as important parametrs. There were some features for which the difference was not discernible. We considered those as unsignificant features and dropped the columns.

a) Important Features: Following features were considered as important in anlayisis of this particular problem: Orbit Uncertainity and Minimum Orbit Intersection.

b) Moderately Important Features: Following features were considered as moderately-important in anlayisis of this particular problem: Absolute Magnitude, Estimated Diameter, Epoch Date Close Approach, Relative Velocity, Eccentricity, Perihelion Distance, and Mean Anomaly.

c) Unimportant Features: Following features were considered a unimportant and were dropped: Inclination, Asc Node Longitude, Perihelion Arg, and Mean Motion.

Thus, out of forty features, we have a reduced dataset containing thirteen features including the target variable. We observed that there is considerable reduction in training time and model accuracy when the dataset containing thirteen features is used for training than when the entire dataset is used for training.

III. ALGORITHMS

We have used three machine learning algorithms namely: KNN, Logistic Regression and Decision Trees. For each of the methods used, we performed the analysis first using the entire dataset and then by the reduced dataset. For each of the cases we have divided the data into training and test samples in the ratio 7:3 and performed the analysis. After analysis we have evaluated the performance using metrics such as precision, recall, f1 score and confusion matrix. We have also calculated the training time for each of the models and compared one against the other.

A. Logistic Regression

In statistics, logistic regression, or logit regression, or logit model is a regression model where the dependent variable (DV) is categorical. The binary logistic model is used to estimate the probability of a binary response based on one or more predictor (or independent) variables (features). Like other forms of regression analysis, logistic regression makes use of one or more predictor variables that may be either continuous or categorical. Unlike ordinary linear regression, however, logistic regression is used for predicting binary dependent variables (treating the dependent variable as the outcome of a Bernoulli trial) rather than a continuous outcome.

For the research purpose the dependent variable is whether the asteroid is hazardous or non-hazardous. The independent variables chosen are scaled down using the standard scaler.

1) Logistic Regression when All the Features are Used: Displayed below is the confusion matrix for Logistic Regression when all the parameters are used.

TABLE I. CONFUSION MATRIX LOGISTIC REGRESSION ALL ATTRIBUTES

Actual Value of Target Variable	**Predicted Value of Target Variable**	
	Class 0	***Class 1***
Class 0	1139	30
Class 1	48	190

2) Logistic Regression when Principal Features are Used: Displayed below is the confusion matrix for Logistic Regression when the principal parameters are used.

TABLE II. CONFUSION MATRIX LOGISTIC REGRESSION PRINCIPAL ATTRIBUTES

Actual Value of Target Variable	**Predicted Value of Target Variable**	
	Class 0	***Class 1***
Class 0	1140	29
Class 1	48	190

Thus, we observe that the accuracy for the problem improved to a certain extent when principal feature attributes were taken into consideration than when all the attributes were taken into consideration.

B. K Nearest Neighbours

In pattern recognition, the *k*-nearest neighbors algorithm (*k*-NN) is a non-parametric method used for classification and regression. Algorithm of K-Nearest Neighbor (K-NN) is defined as a supervised learning algorithm used for classifying objects based on closest training examples in the feature space. KNN is the most basic type of instance-based learning or lazy learning. It assumes all instances are points in n-dimensional space. A distance measure is needed to determine the "closeness" of instances. KNN algorithm is a simple technique that stores all available cases and classifies new cases based on a similarity measure. It is a type of lethargic knowledge where the function is only approximated nearby and the entire working out is deferred until classification. An entity is classified by the best part of its neighbors. K is always a positive integer. The correct classification is known because the neighbors are selected from a set of objects.

Some of the main advantages of KNN are: (i) it is very too simple to implement and easy to justify the outcome of KNN (ii) Robust to noisy training data (especially if we use Inverse Square of weighted distance as the "distance"), and (iii) Effective if the training data is large. Although KNN has those advantages, it has some disadvantages such as: (i) There is no thumb rule to determine value of parameter K, (ii) A high computation cost since it depends on computing the distance of each test instance to all training samples, and finally (iii) Low accuracy rate in multidimensional data sets with irrelevant features.

1) KNN when All the Features are Used: Displayed below is the confusion matrix for KNN Algorithm when all the features are used.

TABLE III. CONFUSION MATRIX KNN ALL ATTRIBUTES

Actual Value of Target Variable	Predicted Value of Target Variable	
	Class 0	***Class 1***
Class 0	1141	28
Class 1	139	99

2) KNN when Principal Features are Used: Displayed below is the confusion matrix for KNN Algorithm when the principal features are used.

TABLE IV. CONFUSION MATRIX KNN PRINCIPAL ATTRIBUTES

Actual Value of Target Variable	Predicted Value of Target Variable	
	Class 0	***Class 1***
Class 0	1136	33
Class 1	98	140

Thus, we observe that the accuracy for the problem improved to a considerable extent when principal feature attributes were taken into consideration than when all the attributes were taken into consideration.

C. Decision Tree

A decision tree is a classifier that performs recursive partition of the instance space. The decision tree consists of nodes that form a rooted tree, meaning it is a directed tree with a node called "root" that has no incoming edges. All other nodes have exactly one incoming edge. A node with outgoing edges is called an internal or test node. All other nodes are called leaves (also known as terminal or decision nodes). In a decision tree, each internal node splits the instance space into two or more sub-spaces according to a certain discrete function of the input attributes values. In the simplest and most frequent case, each test considers a single attribute, such that the instance space is partitioned according to the attribute's value. In the case of numeric attributes, the condition refers to a range. Each leaf is assigned to one class representing the most appropriate target value. Alternatively, the leaf may hold a probability vector indicating the probability of the target attribute having a certain value. Instances are classified by navigating them from the root of the tree down to a leaf, according to the outcome of the tests along the path. Given this classifier, the analyst can predict hazardous and non-hazardous asteroids (by sorting it down the tree), and understand the behavioral characteristics of the entire population regarding their potential danger.

1) Decision Tree when All the Features are Used: Displayed below is the confusion matrix for Decision Tree Algorithm when all the features are used.

TABLE V. CONFUSION MATRIX DECISION TREE ALL ATTRIBUTES

Actual Value of Target Variable	Predicted Value of Target Variable	
	Class 0	***Class 1***
Class 0	1165	4
Class 1	6	232

2) Decision Tree when Principal Features are Used: Displayed below is the confusion matrix for Decision Tree Algorithm when the principal features are used.

TABLE VI. CONFUSION MATRIX DECISION TREE PRINCIPAL ATTRIBUTES

Actual Value of Target Variable	Predicted Value of Target Variable	
	Class 0	***Class 1***
Class 0	1165	4
Class 1	3	235

Thus, we observe that the accuracy for the problem improved to a certain extent when principal feature attributes were taken into consideration than when all the attributes were taken into consideration.

D. Accuracy score comparison for different models

We have calculated the accuracy of computation first by considering all the attributes and next by considering the principal attributes for each of the training models. We have found that accuracy for each of the models increase to a certain extent when principal feature attributes are taken into consideration than when all the attributes are taken into consideration.

TABLE VII. ACCURACY SCORE COMPARISON

Serial No.	Accuracy Score			
	Algorithm	***All Features***	***Principal Features***	***Percentage Improvement***
1	Logistic Regression	0.94456289	0.94527363	0.075244%
2	KNN	0.88130774	0.90689410	2.903225%
3	Decision Tree	0.99289267	0.99502487	0.214745%

E. Training Time Comparison for different Models

We have calculated the time for computation first by considering all the attributes and next by considering the principal attributes for each of the training models. We have found that the training time for each of the models reduce considerably when principal feature attributes are taken into consideration than when all the attributes are taken into consideration.

TABLE VIII. TRAINING TIME COMPARISON

Serial No.	Training Time in seconds			
	Algorithm	***All Features***	***Principal Features***	***Percentage Improvement***
1	Logistic Regression	2.63017535	1.23752593	52.9489%
2	KNN	1.92393827	1.15565967	39.9326%
3	Decision Tree	1.00369882	0.92249917	8.0900%

IV. Result

We observe that the training time for the machine learning algorithms reduced by a considerable extent when principal features were used for analysis than when all the features were used for the analysis. Also, the accuracy score for the machine learning models increased by a marginal extent when principal features were used for analysis than when all the features were used for analysis. Out of all the algorithms, Decision Tree performed the best taking into consideration both the principal feature attributes as well as the entire set of feature attributes followed by Logistic Regression and KNN. The percentage improvement in training time was the most for Logistic Regression followed by KNN and Decision Tree.

V. Conclusion/Future Work

Out of forty features present in the dataset, we reduced it to a dataset containing thirteen features. There were not significant differences in accuracy when the analysis was carried out using principal features than when carried out using all the features. It appears that the other features were only weakly indicative of potential danger of the asteroids, as Orbit Uncertainty and Minimum Orbit Intersection seemed to dominate the others in terms of being able to accurately predict the potential danger of the asteroids. This shows that the features that were excluded for analysis, were indeed not the major determining parameters for the exercise. The training time however, reduced drastically when principal features were taken into consideration than when all the feature attributes were taken into consideration. This shows the importance of choosing important features and obtaining good data.

We intend to carry this analysis further using even less number of features. We intend to perform the feature engineering process even more intricately and see if there is any remarkable difference in accuracy score of the algorithms. It would be interesting to continue this analysis using user defined algorithms for Naïve Bayes and KNN instead of using built in functions. We also intend to carry out this analysis using other machine learning algorithms like Random Forests or K-Means Clustering.

References

[1] Lior Rokach, Oded Maimon , “DECISION TREES”.

[2] Amal H. Khaleel , Ghaida A. Al-Suhail , Bushra M. Hussan, “A Weighted Voting of K-Nearest Neighbor Algorithm for Diabetes Mellitus”.

[3] Eric Lam, Chongxuan Tang, “Titanic – Machine Learning From Disaster”.

[4] Vaishnav Kshirsagar, Nahush Phalke, “Titanic Survival Analysis using Logistic Regression”

[5] Towards Data science