



UNIVERSITY OF
TORONTO

**Department of Mechanical and
Industrial Engineering**

MIE1628: Big Data Science
Final Project: Identity Potentially Hazardous Asteroids

Date: December 17th, 2020

Submitted by Team 10:

Yangzhen Hu
Pei Yao Li
Jinhan Lin
Mengqiao Zhang
Chenjie Zhao

Table of Contents

1.0 Introduction	2
1.1 Problem and Objective	2
1.2 Feature and Target Introduction	2
1.3 Data Preprocessing	3
2.0 Exploratory Data Analysis	5
2.1 Raw Feature Exploration	5
2.2 Correlation Exploration via Heatmaps	6
2.3 Feature Engineering	8
3.0 Feature Selection	13
3.1 Manual Feature Selection	13
3.2 Algorithm Feature Selection	15
4.0 Model Implementation	16
4.1 Model Selection	16
4.2 Relevant Metrics	17
4.3 Undersampling and Oversampling	18
4.4 Model Evaluations	19
5.0 Final Proposed Model and Conclusion	20
5.1 Hyperparameter Tuning and Test Performance	20
5.2 Conclusion and Future Improvement	21
6.0 Reference	22
7.0 Appendix	25
Appendix A- Definition of Features	25

1.0 Introduction

This section below introduces the problem, objective, dataset as well as data preprocessing steps. The code for this section is mainly written by Mengqiao, with contributions of a few functions from Peiyao and Chenjie. The analysis of this section is mainly written by Mengqiao with some parts written by Yangzhen and Peiyao.

1.1 Problem and Objective

There are many asteroids and meteoroids in our galaxy, some are pretty to look at but some are also dangerous. For example, in 2013, a 60-foot meteor exploded above Russia, releasing the equivalent energy of 30 atomic bombs and injured more than 1000 people [1]. Cosmic impacts from asteroids could have devastating consequences, so it's crucial for us to systematically assess the likelihood of this threat by identifying potentially hazardous asteroids. Near Earth asteroids (NEAs) whose Minimum Orbit Intersection Distance (MOID) with the Earth is 0.05 au or less and whose absolute magnitude (H) is 22.0 or brighter are called potentially hazardous asteroids (PHA)[2]. Identifying PHAs can be helpful to institutions like NASA where they can monitor these PHAs and come up with emergency plans to deflect and destroy them if necessary [3]. If we fail to detect a PHA, there could be a detrimental impact to lives on earth [4]. Since there are various types of data available on asteroids, it might be possible to train machine learning models to help detect PHAs.

The overall objective of this project is to correctly identify all potentially hazardous asteroids (PHA). In order to do so, the team explored various machine learning models like gradient boosting, decision tree (GBT), support vector classifier (SVC), random forest (RF) and logistic regression (LG). After implementing these models, hyperparameter tuning was done on the model with the best performance. The final tuned model was used for testing to assess its performance on a test dataset.

1.2 Feature and Target Introduction

The training and testing datasets provided are maintained by the Jet Propulsion Laboratory of California Institute of Technology. The training data has 44 both numerical and categorical features, and one target column for over 900,000 asteroids. There are 9 categorical features and the remaining are numerical features. The target in this project is called PHA which stands for Potentially Hazardous Asteroid. PHA is a binary class target, "Y" stands for the object is a PHA, and "N" means the object is not a PHA. There are 750,882 rows in the training set and 187,721 rows in the testing set. In the training set, there are only 0.2% of objects are flagged as YES and over 99.8% of them are NO. It can be clearly stated that the training dataset is heavily imbalanced. Top numerical features and categorical features are necessary to be defined and understood for later feature engineering and feature selection. The following 2 tables show the full name and definitions of top numerical features and categorical features. For other features in the dataset that are less crucial, definitions are shown in Appendix A. However, there are some missing and categorical text classes in the dataset, so the next part will present how data was cleaned and imputed.

Table 1-1: Definitions of Top Numerical Features

Feature Name	Full Name	Definition
H	Absolute Magnitude Parameter	Visual magnitude an observer would record if the asteroid were placed 1 Astronomical Unit (au) away[5].
moid	Minimum orbit intersection distance	a measure used in astronomy to assess potential close approaches and collision risks between astronomical objects.[6].
e	orbit eccentricity	The orbital eccentricity of an astronomical object is a dimensionless parameter that determines the amount by which its orbit around another body deviates from a perfect circle. [7].
a	semi-major axis	The semi-major axis is one half of the major axis[8].
ad	aphelion distance	The perihelion (q) and aphelion (Q) are the nearest and farthest points respectively of a body's direct orbit around the Sun.[9].

Table 1-2: Definitions of Top Categorical Features

Feature Name	Full Name	Definition
NEO	Near-Earth Object	NEOs are asteroids and comets with perihelion distance q less than 1.3au[10].
Class	Object Classification	Object Classification[11].
orbit_id	The ID of JPL NEA orbit	The ID of JPL NEA orbit[12].

1.3 Data Preprocessing

The first stage of data cleaning is to fill missing values. Two approaches were taken to fill missing values of the data, one approach is to fill according to the definition of the feature and the second approach is to fill with mean values. In the first approach, the definition of feature “NEO” (near-earth object) and definition of feature “q” (perihelion distance) are utilized to fill the missing values of “NEO”. According to the definition of NEO, if an object’s perihelion distance (q) is less than 1.3 au, it’s then identified as NEO[13]. Hence, missing values in the “NEO” column are imputed by applying this concept since the values of perihelion distance in feature “q” can be found in the dataset. The rest of the missing columns like H, per_y, ad, rms, and some of the sigmas with missing values were all filled with their

respective mean values. Note that the columns decided to drop already (explained in section 3.0) were not filled. The code for this part is in notebook cmd 11-14, and 17.

After filling in missing values, the next stage of data preprocessing is to convert categorical features into numerical formats so they can be processed by machine learning models. The binary classes categorical features like “NEO” and the target “pha” are all converted to 0 and 1 where 0 means no and 1 means yes. For the rest of the categorical features, one-hot encoding was utilized to convert them to the numerical format that can be utilized by machine learning models. The code for this section is in notebook cmd 30.

The step after that is normalization, for this project the min-max scaling method is utilized for normalization. Since these are physical data, not necessarily following a normal distribution, 0 mean and 1 standard deviation method is not utilized. By using the min-max scaling method as shown by the formula below, all values can be reduced into range 0-1 by dividing by a straight ratio (Xmax-Xmin). This allows all features to be in the same order of magnitude with each other. The code for this section is in notebook cmd 52.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Lastly, the training dataset is split into training and validation dataset by a ratio of 70% training and 30% validation. This allows the team to use the validation data for performance evaluations and leave the testing dataset untouched until the final testing stage. The code for this section is in notebook cmd 54.

2.0 Exploratory Data Analysis

Data exploration is an important part of every data analysis project. To know the data and understand its trends is important, thus; this section below discusses the exploratory data analysis as well as feature engineering after completing the data exploration. The code and analysis of this section are mainly written by Yangzhen and Jinhan with some parts written by Mengqiao as well. The code for this section can be found in the notebook from cmd 21 to 41.

2.1 Raw Feature Exploration

Despite the lack of useful information in sigma features, there are informative features in the original datasets. Features like moid and q are helpful features as their distribution plot can be found in Figure 2-1 below. The first distribution plot on the left belongs to the earth and the object's minimum orbit intersection distance (moid). Log transformation was taken to show a clear and useful plot while preserving important information. A borderline can be found between potential hazardous asteroids (PHA) and non-PHA. This means this feature can be helpful to distinguish the target. It is one of the conditions that define a PHA, so it makes sense that it would be an important feature. A borderline can also be found in the third distribution plot which belongs to perihelion distance (q). The perihelion distance is another distance aspect of asteroids. The graph shows the smaller the perihelion distance is, the more likely the asteroid to be a PHA. Although the distribution of the eccentricity (e), which is the second plot, does have an indicative boundary, the overlapping area is not large. The feature, e, can still be an acceptable feature to use in predicting. Lastly, the histogram of the feature neo (near-earth object flag) indicates that all PHA are near-object. Therefore, from the above analysis it can be concluded that these features can be very helpful to the target predictions.

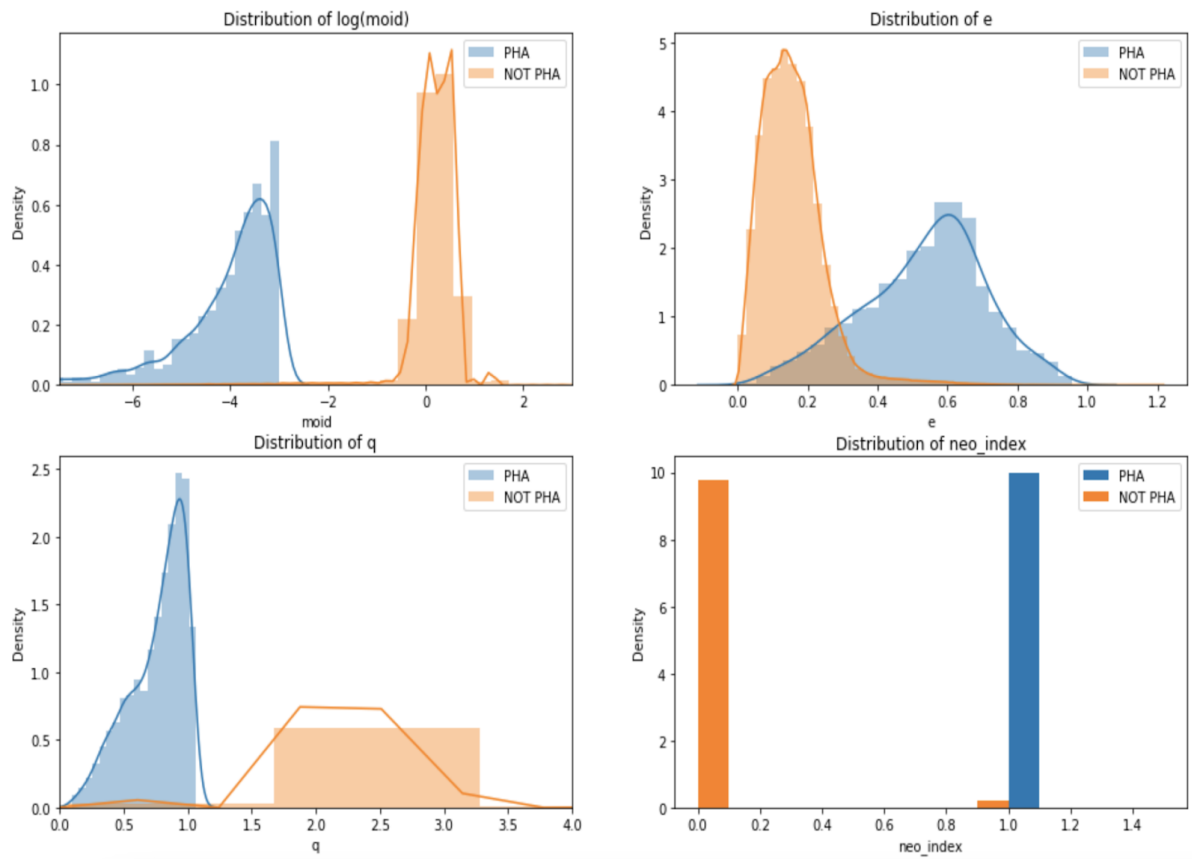


Figure 2-1: Distribution Plots and Histogram Plot For Key Features

2.2 Correlation Exploration via Heatmaps

In the dataset given, most numerical features also came with a sigma value of one standard deviation. In order to see whether the sigma values for those features would have an impact on the feature's correlation with the target, sigma features are added or subtracted from their respective feature, and the correlation between each feature and pha is calculated. For e, a, q, i, om, w, ma, ad, n, tp and per columns, Figure 2-2 was plotted as the correlation between original features. In the following step, the group added one sigma for each column and plotted the heat map according to Figure 2-3. Then the same methodology was followed and plotted the minus one sigma heat map according to Figure 2-4.

As shown in the following figures, it's clear to see that the features that do not plus or minus have the highest correlation. Hence, the current features' values are decided to be kept and did not add or minus sigma values from the original feature values.

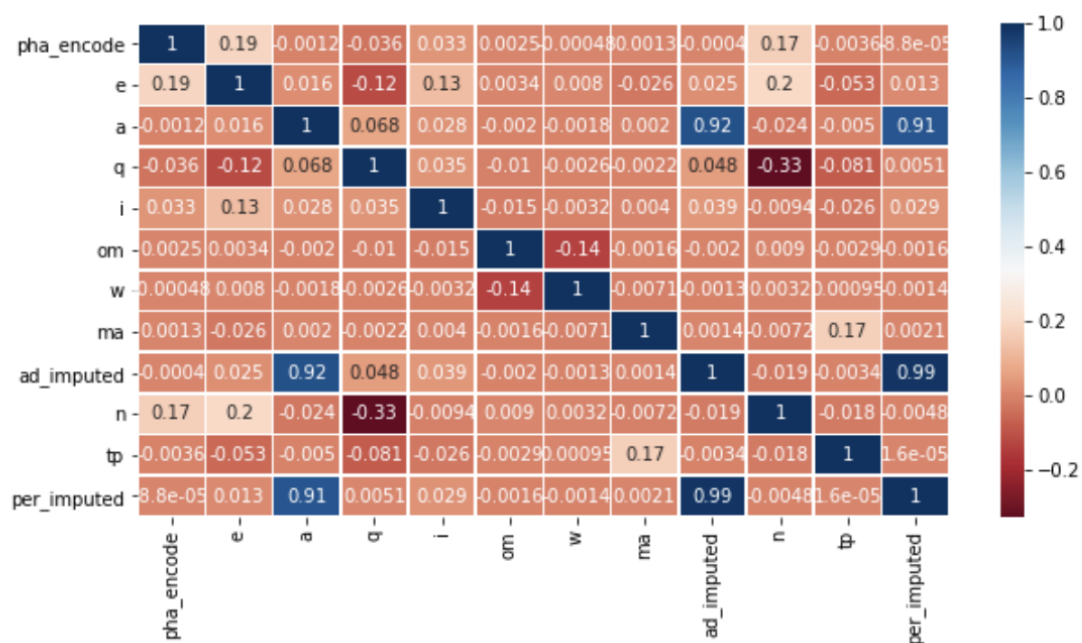


Figure 2-2: Original Feature Heatmap

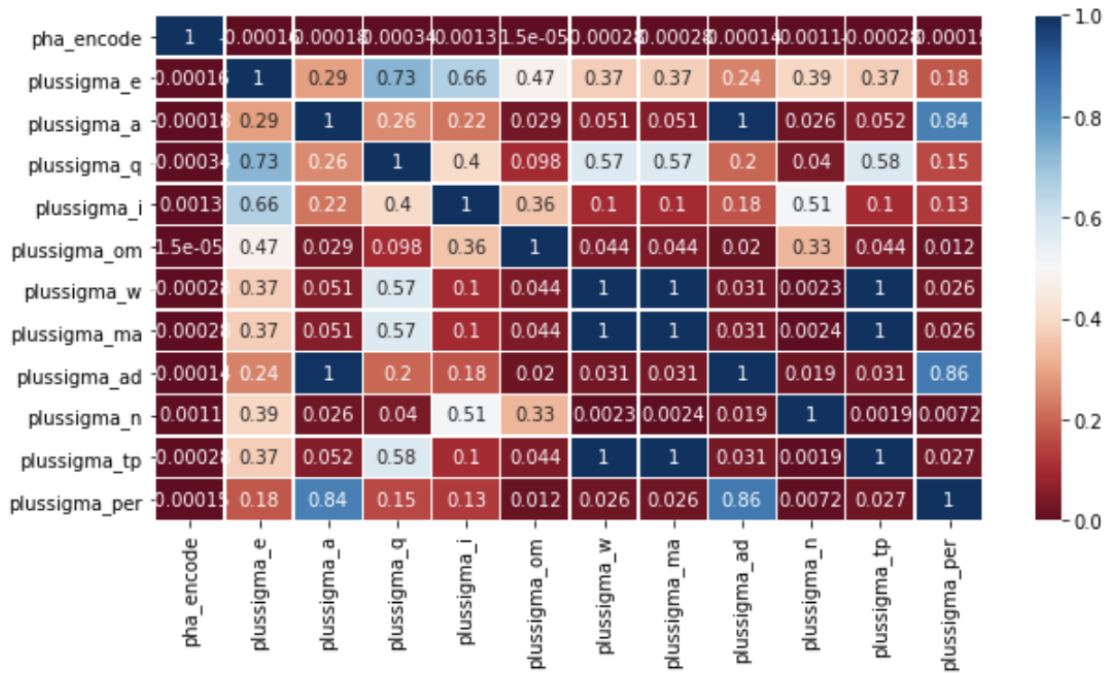


Figure 2-3: Feature Heatmap With Sigma Added

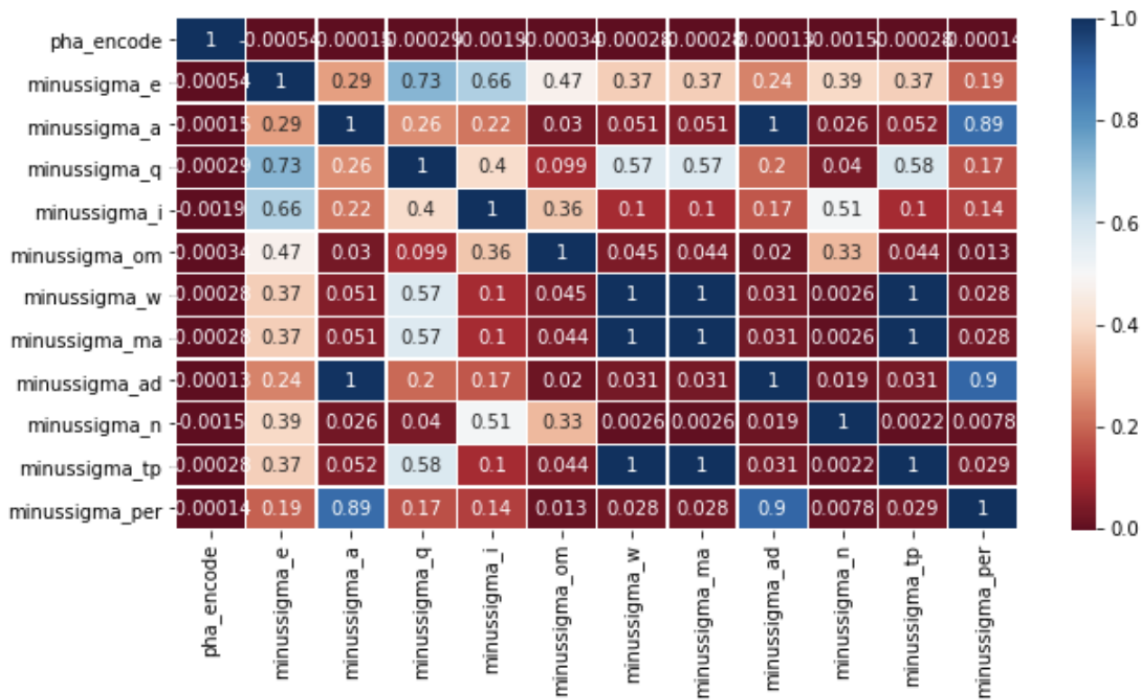


Figure 2-4: Feature Heatmap With Sigma Subtracted

2.3 Feature Engineering

Feature engineering could improve the models' performances by creating new features based on existing features and domain knowledge. In this project, 5 engineered features were added according to target definition and physics concepts. All new features were justified by their mutual information (MI) score compared to the original feature and their distribution plot vs the target to show how well the feature can separate the two target classes. Mutual Information is a measurement of the mutual dependence between two variables, it quantifies the "amount of information" obtained about one variable through observing the other random variable [14]. "Sklearn" was used to implement MI after confirming with the TA as this calculation is not available in pyspark. If the MI score is high, it generally means this feature is important to predict the target.

By considering the definition of PHA, 3 categorical features are created and the reason and meaning behind those features are explained below.

a) H_Engineered

The feature "H_Engineered" has been created to determine if H is larger than 22 or less. H is a parameter that represents the absolute visual magnitude of an asteroid. One essential requirement of PHA is that this asteroid has H less than 22. By creating this feature, data could be categorized from numerical to binary categorical data which could save computational efficiency since the precision of H was not crucial in the prediction of the target.

b) moid_Engineered

Similar to H mentioned above, minimum orbit intersection distance (moid) is also a binary requirement to determine PHA. The threshold was set as 0.05 au since a PHA needs to have moid less than 0.05au.

c) moid_H_Engineered

"moid_H_Engineered" was created to see if this object has both moid less than 0.05au. Theoretically, this feature could directly determine if the object is PHA or not, however, exceptions do exist in the dataset. Therefore, this feature was created in order to be trained for a better prediction.

Refer to Figure 2-5 below, engineered features can clearly separate PHA and non PHA. This means that those features could be beneficial to the prediction.

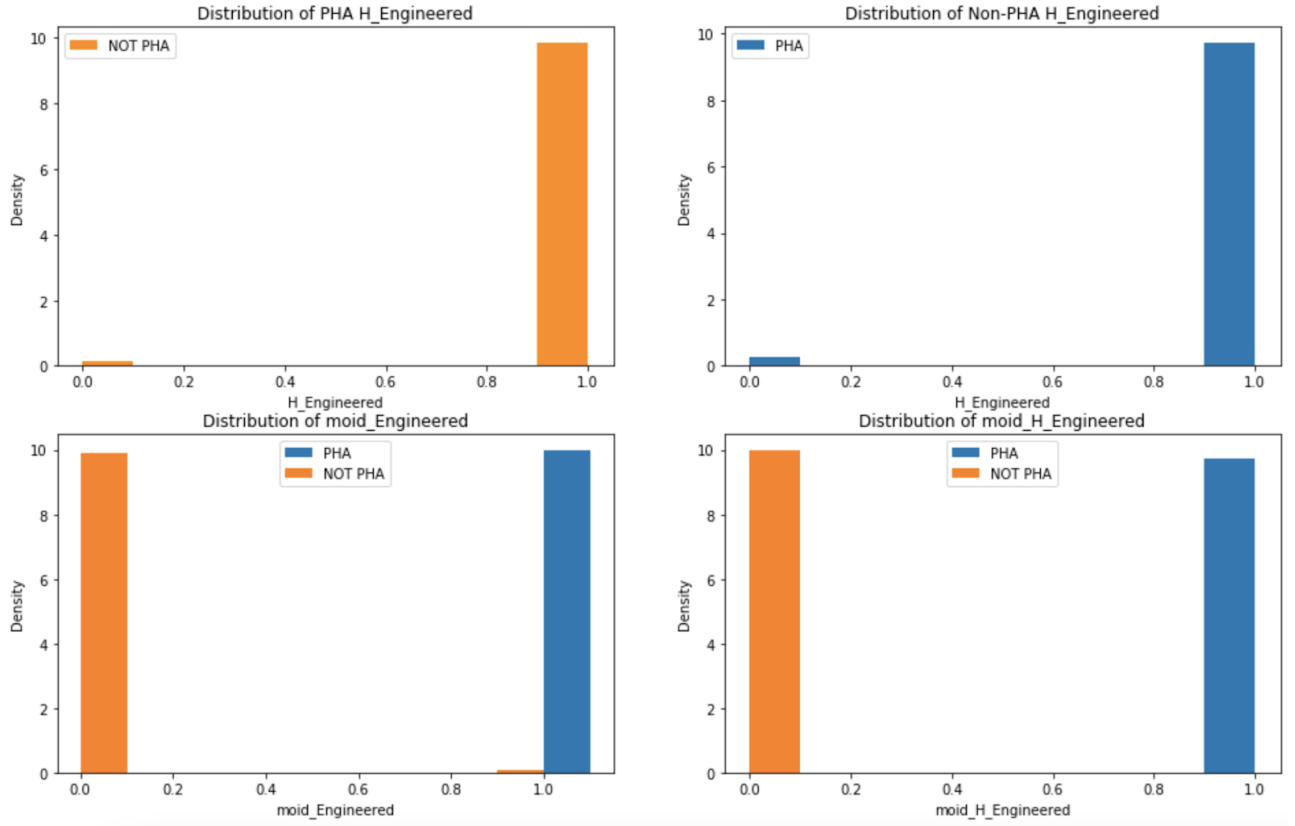


Figure 2-5: Histogram Plots For Engineered Categorical Features

Besides these categorical new features, two numerical features are added which are derived from the real-world physical perspective. The explanations are listed below.

a) a_e_Engineered

it is the ratio between the semi-major axis and orbit eccentricity. Since the semi-major axis is an important property to define an ellipse, it could be crucial to define the orbit of an asteroid. Meanwhile, eccentricity is a parameter that determines the orbit is a circle, an ellipse or a trajectory. When eccentricity is 0, a circle will be obtained. In this project, an asteroid could only have an ellipse orbit, therefore, only between 0 and 1 will be considered. Physically, when a larger eccentricity is obtained (in 0 and 1 scale), it generally means the semi-major axis could be larger if perimeters are the same. As the relation elaborated above, a ratio is the best method to represent the correlation.

b) ada_Engineered

“ad” is the aphelion distance of the orbit and “a” is the semi-major axis of the orbit. Theoretically, aphelion distance and semi-major axis are on the same axis in a plane. The ratio between them could represent where the sun object(Earth in this project) is located on the major axis of the asteroid's orbit. Ratio and difference between these two features are analyzed and ratio of a and ad was kept due to its high mutual information result.

Figure 2-6 below shows a comparison among original features and engineered features. The two distributions at the top are from the original features semi-major axis (a) and aphelion distance (ad). The two classes are heavily overlapping which means these features do not perform well in separating the two classes. In contrast, the two distributions from engineered features at the bottom show great improvement that classes are easier to be identified with the new features.

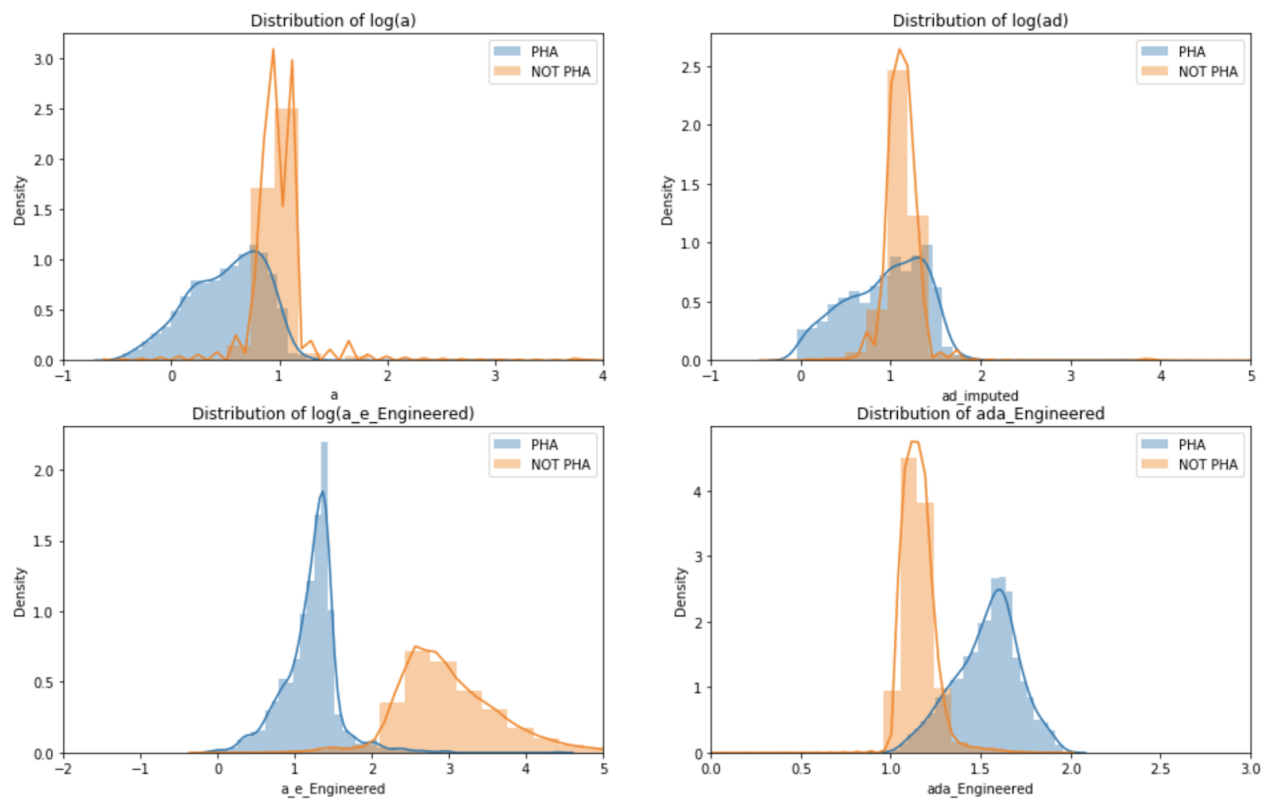


Figure 2-6: Distribution Plots For Engineered Numerical Features

The following tables indicate features that are implemented after feature engineering and those not implemented. For the plots, the orange color represents objects that are not PHA and the blue color means objects are flagged as PHA. For numerical features, the plot shows the distribution of PHA objects and non-PHA objects. If two distributions separate apart, that means the feature could identify the target well. For the categorical features, the plot shows the percentage of PHA and non-PHA in two classes. If the plot can show clearly about the classes, that means the feature is good to tell the difference of the target.

Table 2-1: Justification For Implemented Engineered Features

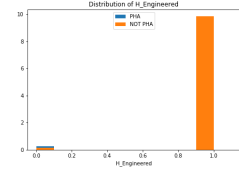
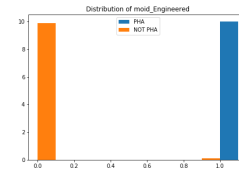
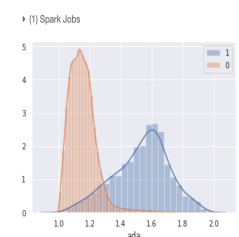
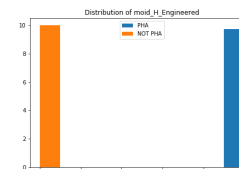
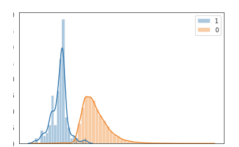
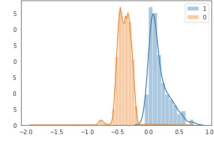
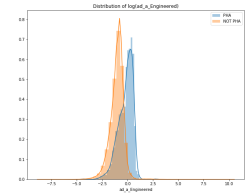
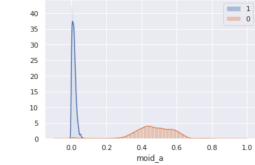
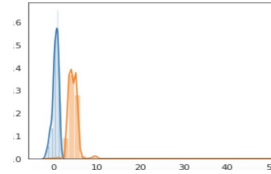
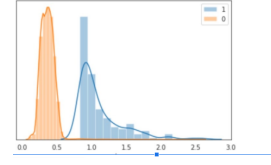
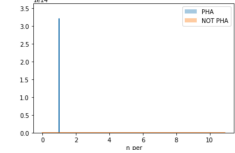
Feature Name	Formula	MI Score	Distribution Plot (If applicable)	Justification of Including or Excluding the Feature
H_Engineered	$H < 22, H > 22$	0.16919767		MI score Improved from the original feature
moid_Engineered	$moid < 0.05, moid > 0.05$	0.00991162		MI score Improved from the original feature
ada_Engineered	ad/a	0.00665146		MI score Improved from the original feature
moid_H_Engineered	$moid < 0.05 \& H < 22$	0.01495196		MI score Improved from the original feature
a_e_Engineered	a/e	0.00847964		MI score Improved from the original feature

Table 2-2: Justification For Not Implemented Engineered Features

Feature Name	Formula	MI Score	Distribution Plot (If applicable)	Justification of Including or Excluding the Feature
n_ad_Engineered	$n \cdot ad$	0.0086736		MI Score Not Improved
ad_a_Engineered	$ad - a$	0.003268		MI Score Not Improved
moid_a_Engineered	$moid / a$	0.00969687		MI Score Not Improved
q_e_Engineered	$q - e$	0.00900825		MI Score Not Improved
n_e_Engineered	$n + e$	0.00908471		MI Score Not Improved
n_per	$n \cdot per$	0.17250248		Discarded Due to Constant Value

3.0 Feature Selection

Both manual feature selection and two feature selection algorithms are performed in this project and they will be discussed below. The code and analysis of this section are written by Yangzhen, Jinhan, and Mengqiao. The code for this section can be found in the notebook from cmd 15 and cmd 43 to 50.

3.1 Manual Feature Selection

After uploading both train and test datasets, The team first checked the dimensions of datasets, for the training set, there are 750882 rows and the test dataset has 187721 rows. There are 44 features except for the target value. By studying the physical parameters, the team dropped irrelevant features and cleaned the rest of the data. After checking the missing values, the team decided to drop the name, prefix, diameter, albedo, diameter_sigma columns because more than 85.5% of the data are missing. Since the information in the object ID, spkid, full_name, pdes are unique values in each row and are irrelevant to the PHA prediction, it could not facilitate the following analysis, so they were removed as well. Equinox was also removed because it's a constant value. There are also some features that are the same information but given in different formats and units such as epoch_mjd and epoch_cal in Modified Julian Date format and in "yyyymmdd" format. Since features like tp_cal, moid_ld, epoch_cal, epoch are the same as tp, moid and epoch_mjd, just in a different format. For instance, the features per and per_y have exactly the same correlation with other features (Figure 3-1), so only one of them was kept. For the analogous reason, the team decided to drop those columns to avoid redundancy. A summary of all features manually dropped and the reason can be found in Table 3-1 below.

Table 3-1: Features Dropped during Manual Feature Selection

Feature Dropped	Definition	Reason to Drop
name	Name of objects	97.7% missing value
prefix	Prefix of the names	99.9% missing value
diameter	Diameter of the asteroids	85.5% missing value
albedo	Parameter of luminosity	85.6% missing value
diameter_sigma	1-sigma uncertainty of albedo	85.5% missing value
id	An ID combined with letters and digits	Unique values, and irrelevant to prediction
spkid	A numerical ID code which indicates different Asteroids	Unique values, and irrelevant to prediction
full_name	An ID contains time information	Unique values, and irrelevant to prediction
pdes	Object primary designation	Unique values, and irrelevant to prediction

equinox	Equinox of the reference frame	Constant values
epoch, epoch_cal	epoch: Epoch of osculation epoch_cal: epoch in the form of yyyyymmdd epoch_mjd: epoch in modified Julian day form	All three features give the same information, just in 3 different formats/ units. Epoch_mjd has a better format for processing so kept epoch_mjd and dropped epoch and epoch_cal
tp_cal	tp: The time at which an object is at perihelion tp_cal: in the form of yyyyymmdd	Both features are tp just in different formats, kept tp due to having a better format for processing, and dropped tp_cal to avoid redundancy.
moid_ld	Earth minimum orbit intersection distance au unit	Can be calculated from moid
per	Sidereal orbit Period in day	Dropped due to high correlation with per_y, see explanation above

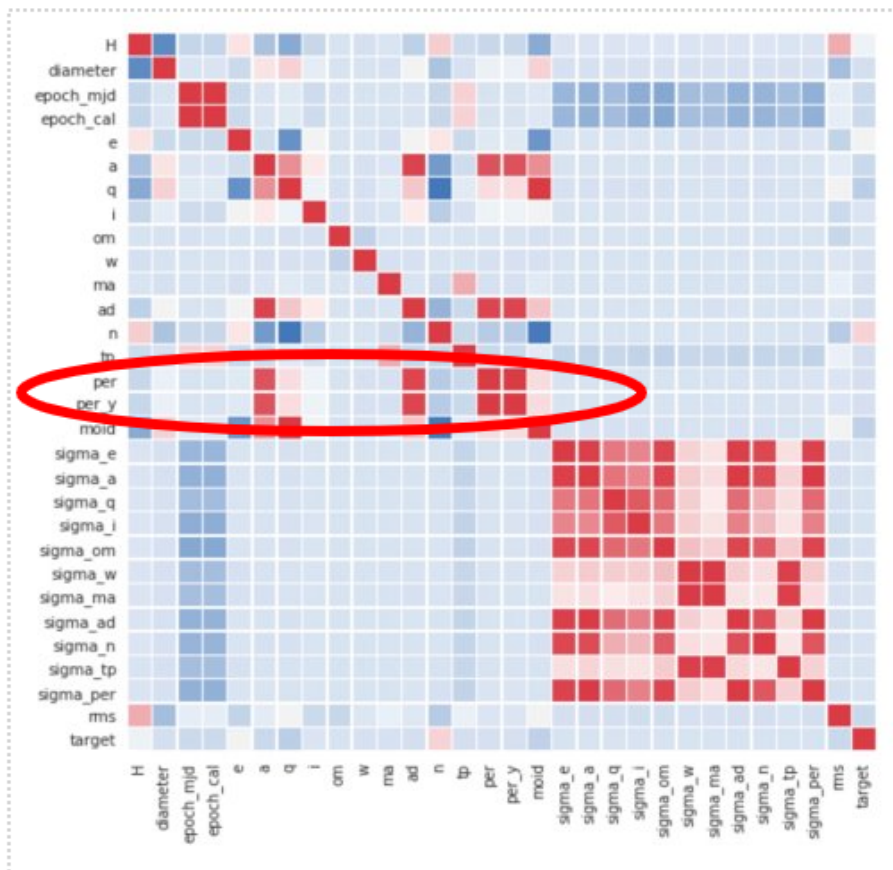


Figure 3-1: Feature Selection According To The Heatmap

3.2 Algorithm Feature Selection

In order to save computational efficiency, the random forest method was used to select important features, and then mutual information was used to validate the result.

Random forest consists of hundreds of decision trees, and each of them built over a random extraction of the observations from the dataset, and features are randomly extracted. Because not every tree sees all features, this could make sure that the results are less prone to overfitting[15]. After selection, top 15 categories are moid_H_Engineered, a_e_Engineered, class_onehot_MBA, class_onehot_APO, sigma_q_imputed, neo_index, per_y_imputed, ada_Engineered, sigma_a_imputed, sigma_ma_imputed, orbit_onehot_62, orbit_onehot_57, H_imputed, orbit_onehot_29 and sigma_e_imputed. Mutual information was used to validate results from the random forest feature selection algorithm. According to Figure 3-2 below, the top features from both mutual information and random forest feature selection are similar. Those same features are circled out in blue and they are still in a high rank in MI ranking. Since the random forest considers all features and classes in the feature, there should be more similar features compared to MI ranking.

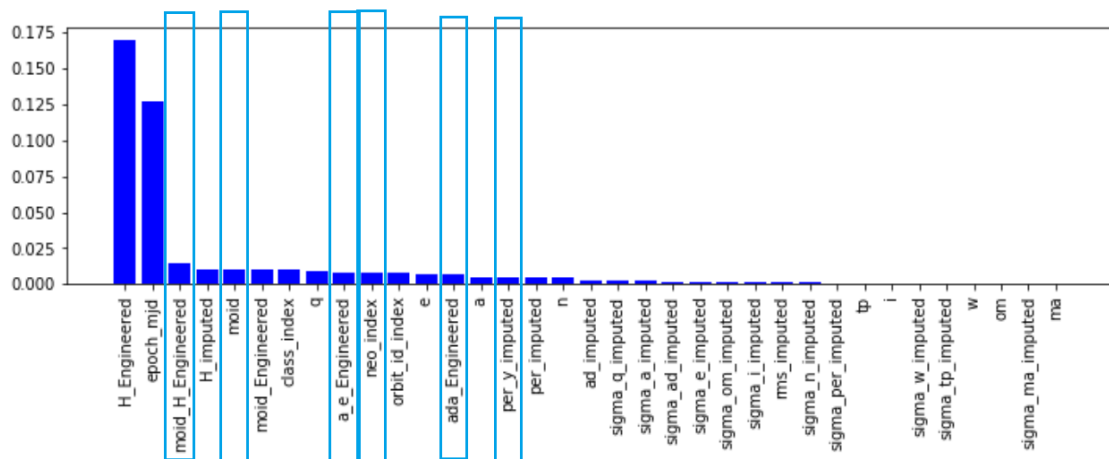


Figure 3-2: Mutual Information Rank With Random Forest Top Features Circled

With the validation with MI and Random Forest, it can also prove that features engineered in the previous section have a good performance. According to Figure 3-3 below, all engineered features are circled out and it's clear to see all of them have a relatively high MI score, thus these new features are indeed helpful for target prediction. Hence, all features with high feature importance from the random forest were kept and features that have close to 0 feature importance are dropped.

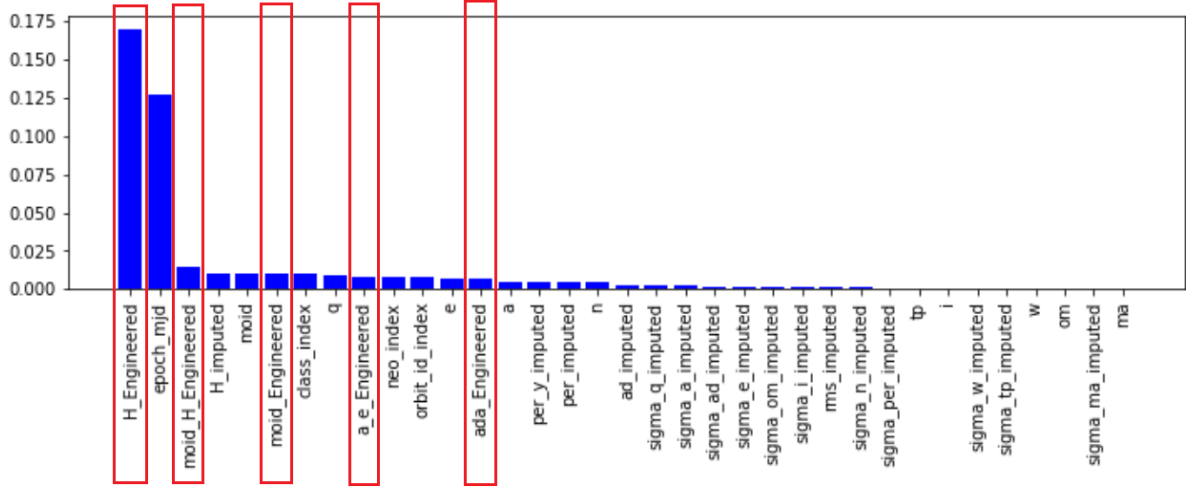


Figure 3-3: MI Rank With Engineered Feature Circled

By the end of this point, raw data are cleaned, imputed by definition, and 5 features are created according to PHA definition and physical aspects. With the assist of mutual information and random forest, features are selected according to the feature importance. The following part will show how models are implemented and how hyperparameters are tuned.

4.0 Model Implementation

The above sections explained data preprocessing, feature engineering, and feature selection, and so the next step is model implementation. This section below will discuss model implementation and their performances. The code and analysis of this section are written by Chenjie and Peiyao. The code for this section is in notebook cmd 60 to 144.

4.1 Model Selection

Four models below are selected for this classification task. These models are selected with considerations of the input features and their abilities to handle class imbalanced data.

(a) Gradient Boosted Decision Trees (GBDT)

Boosting techniques have shown their superior performance in handling data imbalance compared to many other machine learning models [15]. Boosting techniques propose a series of weak classifiers. Each classifier is aiming to reduce the residual error from the previous classifier. As a result, data imbalance poses a less significant impact on boosting-based algorithms. The errors caused by wrong predictions in the minority class cannot be ignored as long as a sufficient number of weak classifiers are being trained to handle these errors. GBDT is a concrete example taking advantage of the boosting technique [16]. Trees naturally handle numerical features well at each node. GBDT employs small multi-level decision trees as weak classifiers and uses these weak classifiers to propose a low-variance solution that handles the minority class [15]. The low-variance nature of the output of the trained model also mitigates overfitting, which is important since data from the minority class is already scarce. As expected, it performed best for the original data input.

(b) Random Forest (RF)

RF is very similar to GBDT as it also uses multiple multi-level decision trees as the weak classifiers and combines their results [17]. However, unlike GBDT, RF does not use boosting, which means it would not increase its focus on misclassified samples as training progresses. RF is selected to be compared with the GBDT. It performed worse than GBDT for the original data input, but with the assistance of random oversampling, it achieved a higher recall than GBDT.

(c) Support Vector Classifier (SVC)

SVC learns to classify data samples by creating a hyperplane that separates two classes with the maximum margin [18]. With a soft margin, hinge loss can be minimized for data classes that are not linearly separable. SVC does not handle imbalance data natively. However, one could associate higher weights for data points in the minority class [19]. This technique balances the influence of minority and majority class on the proposed decision boundary. When the original data is used, the weight associated with each minority sample is around 500 times of the majority sample's weight. Classifying each minority sample incorrectly or even allowing the sample to reside between the margin and the decision boundary hyperplane significantly increases the hinge loss. As a result, the decision boundary should remain far away from the minority class training samples. The impact is seen in the result for original data, the SVM had a recall of 1.0 but a low precision of 0.46.

(d) Logistic Regression

LR is a common classification model that also gives the probability of the classification [20]. It is employed here to be compared with the more sophisticated methods above.

4.2 Relevant Metrics

Below are the four relevant metrics for model evaluation. Given the nature of this dataset and the potential impact of PHAs, the primary evaluation metric for the models is recall, and the secondary metric is F1 Score. The reason for this is explained below.

(a) Recall

Recall in this project is the proportion of true positives over real PHA, given by the formula below. This is the most important metric because missing to identify a real PHA may have a detrimental impact on our planet.

$$\text{Recall} = \frac{\text{True Positive}}{\text{Total Real PHA}} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

(b) Precision

Precision in this project is the proportion of true positives over total model predicted PHA. This is a less important metric since it is less dangerous to propose some regular asteroids as PHA. In order to achieve high recall, we are willing to sacrifice some precision to have some false positives, which are essentially false alarms. The cost of a false alarm is not very high since no immediate action is required. False

positives merely add additional non-hazardous asteroids to be continuously monitored by institutions like NASA as if they are potentially hazardous. This requires additional resources to monitor false-positive asteroids but still has significantly less impact than failing to identify a hazardous one. Thus precision is not the priority.

$$\text{Precision} = \frac{\text{True Positive}}{\text{Total Model Predicted PHA}} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

(c) F1 Score

F1 score is the harmonic mean of precision and recall. F1 Score aims to balance the effect of recall and precision; thus, given similar recalls, we would prefer a high F1 score.

$$\text{F1 Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

(d) Accuracy

Accuracy in this project is the proportion of the sum of true positives and true negatives over the number of total samples. Given the extremely imbalanced data for this task, models can easily achieve high accuracy by always predicting the majority class. In fact, all models implemented were able to achieve a minimum of 99% accuracy, but that does not mean their recall and precision were high. Therefore accuracy is not as deterministic for the model evaluation as in other normal classification problems with balanced data.

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total Samples}}$$

4.3 Undersampling and Oversampling

Undersampling and oversampling were performed in order to improve the model performance on the heavily imbalanced data. Both methods aim to balance the class distribution. Undersampling removes the instances of the majority classes and oversampling creates new instances of the minority classes. Synthetic Minority Oversampling Technique (SMOTE) is one of the popular oversampling methods that is considered to use. Due to the high performance of the model without using SMOTE and lack of mature implementation of SMOTE in pyspark packages, random oversampling was used instead in this project. Random oversampling is to randomly select the instances from the minority class and add extra duplicates of these samples in the training data. The implementation of these two methods are the functions *oversample* and *undersample* in the submitted code in cmd. They were both written in pyspark code only to avoid using “Imlearn” in python to satisfy the project requirement. The inputs of these two function are the same: *minor_ratio* is the minority class ratio in the returned dataframe, *df* is the dataframe that the operation is performed on and *target_column* is the column name for the target. The code for this section can be found in cmd56.

4.4 Model Evaluations

Four models were built: Logistic Regression, Gradient Boosting, SVM, and Random Forest, and six different situations were tried in this project: 10%, 20%, and 30% of the minority class ratio for the aforementioned undersampling and oversampling methods. As discussed in the section above, the most important metric for model evaluation is recall and give the similar recall, it's preferable to have a higher F1 score; therefore, the recall and F1 score of each model is shown in Table 4-1 below.

In general, the oversampled and undersampled data performed better than using the original data, and the 30% vs 70% class ratio performed the best. Within the 30% vs 70% oversampled category, the random forest model performed the best with a very high recall of 99% and a relatively good F1 score of 78%. The second-best model would be the gradient boosting classifier in the undersampled 30% vs 70% ratio category. This is also well aligned with the expectation that the decision tree based ensemble models perform the best given the nature of our predictions. Although the SVC achieved a recall of 100% on the original data, the F1 score is significantly lower than the top two models, hence it is not the most ideal. Thus, the best performing model that meets the metrics requirement is the random forest model with the oversampled dataset of 30% vs 70% class ratio. The performance of the final selected model is bolded below and its result is highlighted in green. This final selected model is untuned, the next section will discuss its hyperparameter tuning and its final performance on the test dataset.

Table 4-1: Model Performances Comparison

Dataset	Target Class Ratio Yes vs No	Gradient Boosting		SVC		RandomForest		Logistic Regression	
		Recall	F1	Recall	F1	Recall	F1	Recall	F1
Undersampled	10% vs 90%	97%	86%	96%	96%	98%	57%	97%	89%
	20% vs 80%	98%	75%	94%	95%	99%	61%	97%	68%
	30% vs 70%	99%	68%	99%	48%	100%	41%	98%	59%
Original Data	0.2% vs 98.8%	96%	98%	100%	46%	88%	94%	96%	98%
Oversampled	10% vs 90%	97%	93%	100%	50%	97%	94%	97%	96%
	20% vs 80%	98%	89%	99%	34%	99%	64%	97%	78%
	30% vs 70%	98%	88%	100%	0%	99%	78%	98%	63%

5.0 Final Proposed Model and Conclusion

The final model underwent hyperparameter tuning using cross-validation and grid search, then its true performance was tested using the test dataset, and the result and conclusion are discussed in this section. The code of this section is written by Peiyao, and the analysis is written by Chenjie and Peiyao. The code for this section is in the notebook from cmd 147 to 150.

5.1 Hyperparameter Tuning and Test Performance

The best model, random forest, was tuned for further improvement on the performance. Grid search was performed through the number of trees, max depth, and max bin to find the best hyperparameters. The tuned hyperparameters are shown in Table 5-1, and the performance of the tuned model on both the validation data and test data is shown in Table 5-2. To tune the model, both training and validation data were combined together to perform cross-validation. After the model's best parameter was selected, it was then fitted again only on the training data to verify its performance on the validation data. Note that the team did not use the testing data at all until the very last testing stage to avoid information leakage.

Table 5-1: Hyperparameter Tuning

Hyperparameters and Definition	Gird Search on Hyperparameters	Chosen Hyperparameter
maxDepth: The maximum depth of the tree	5, 20 30	maxDepth=30
numTree: The number of trees in the forest	10 ,20	numTrees=20
numBins: The maximum number of leaf nodes	32, 64	maxBins=32

Table 5-2: Hyperparameter tuning performance

Dataset	Recall	F1 score
Default hyperparameter	99%	78%
CV Tuned hyperparameter on validation data	96%	98%
CVTuned hyperparameter on test data	99%	99%

The test data performance is very good as the recall stays the same and F1 score increased from 78% to 99% which means almost all the potential PHA were able to be correctly identified and there is almost no false alarm. Since the test performance did not vary much from the train and validation performance, it can be concluded that the model is not overfitted or underfitted, and the model performance is stunning. At first the team did not expect non-neural models to have the capacity to successfully fit such a dataset, but the result proves traditional non-neural models can already be used to identify PHAs, which is very exciting.

5.2 Conclusion and Future Improvement

It is demonstrated through the results that undersampling and oversampling can be effective methods to improve model performance given an imbalanced dataset. Non-neural methods such as RF and GBT have the capability to perform well on this dataset. These models can readily be utilized to effectively predict potential hazardous asteroids, which can assist institutions such as NASA to monitor PHA in order to formulate response plans in advance.

Combining 0.3 vs 0.7 oversampling technique and the Random Forest algorithm resulted in the best model when both Recall and F1 score are taken into account. The second-best model was Gradient Boosted Decision Tree which achieved the highest F1 Score when the original data is used for training, demonstrating its ability to handle class imbalance data. The third best model was Support Vector Classifier with class balance weight, which achieved a Recall of 1.0. This can be crucial for not missing PHAs in real-world settings. However, the drawback of SVC is the large amount of false-positive predictions which resulted in a low F1 Score. Thus, overall the best model was Random Forest with an oversampled dataset of 0.3 vs 0.7.

There are also two recommendations for future improvement. First, Naive Bayes classifiers can be explored to predict targets with probabilities so confidence of the prediction can be retrieved. Second, given the extremely imbalanced dataset, outlier detection algorithms can be employed to see if detecting PHAs as outliers would result in a better performance.

6.0 Reference

- [1] H. William, “NASA asteroid-watchers team up on emergency plan with White House, FEMA”, CBS NEWS, 2018, [Online]. Available: [NASA asteroid-watchers team up on emergency plan with White House, FEMA - CBS News](#)
- [2] “PHA (Potentially Hazardous Asteroid)” NASA. [Online]. Available: <https://cneos.jpl.nasa.gov/glossary/PHA.html>.
- [3] W. Hanneke, “NASA Offers New Plan to Detect and Destroy Dangerous Asteroids”, 2018, [Online]. Available: [NASA Offers New Plan to Detect and Destroy Dangerous Asteroids - Scientific American](#)
- [4] L. S. S. Telescope, “Potentially Hazardous Asteroids (PHAs),” *Potentially Hazardous Asteroids (PHAs) | Rubin Observatory*. [Online]. Available: <https://www.lsst.org/science/solar-system/potentially-hazardous-asteroids>.
- [5] “Absolute magnitude,” *Space Wiki*. [Online]. Available: https://space.fandom.com/wiki/Absolute_magnitude#:~:text=In astronomy, absolute magnitude is,compared without regard to distance.
- [6] “Minimum orbit intersection distance,” *Wikipedia*, 12-Oct-2020. [Online]. Available: [https://en.wikipedia.org/wiki/Minimum_orbit_intersection_distance#:~:text=Minimum orbit intersection distance \(MOID\),of a collision with Earth](https://en.wikipedia.org/wiki/Minimum_orbit_intersection_distance#:~:text=Minimum orbit intersection distance (MOID),of a collision with Earth).
- [7] “Orbital eccentricity,” *Wikipedia*, 14-Oct-2020. [Online]. Available: https://en.wikipedia.org/wiki/Orbital_eccentricity.
- [8] “Semi-major and semi-minor axes,” *Wikipedia*, 05-Oct-2020. [Online]. Available: https://en.wikipedia.org/wiki/Semi-major_and_semi-minor_axes.
- [9] “Apsis,” *Wikipedia*, 11-Dec-2020. [Online]. Available: <https://en.wikipedia.org/wiki/Apsis>.
- [10] “NEO Basics,” NASA. [Online]. Available: https://cneos.jpl.nasa.gov/about/neo_groups.html.
- [11] “NASA PDS: Small Bodies Node,” PDS. [Online]. Available: https://pdssbn.astro.umd.edu/data_other/objclass.shtml.
- [12] “Accessible NEAs,” NASA. [Online]. Available: <https://cneos.jpl.nasa.gov/nhats/>.

- [13] Lucy A. McFadden, Richard P. Binzel, “Encyclopedia of the Solar System” (Second Edition), 2007. [Online]. Available: [https://www.sciencedirect.com/topics/earth-and-planetary-sciences/near-earth-objects#:~:text=A%20near%2DEarth%20object%20\(NEO,System%20\(Third%20Edition\)%2C%202014](https://www.sciencedirect.com/topics/earth-and-planetary-sciences/near-earth-objects#:~:text=A%20near%2DEarth%20object%20(NEO,System%20(Third%20Edition)%2C%202014)
- [14] “Mutual information,” *Wikipedia*, 29-Nov-2020. [Online]. Available: https://en.wikipedia.org/wiki/Mutual_information.
- [15] A. Dubey, “Feature Selection Using Random forest,” *Medium*, 15-Dec-2018. [Online]. Available: <https://towardsdatascience.com/feature-selection-using-random-forest-26d7b747597f>.
- [16] J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi, and M. Asadpour, “Boosting methods for multi-class imbalanced data classification: an experimental review,” *Journal of Big Data*, vol. 7, no. 1, p. 70, Sep. 2020
- [17] J. H. Friedman, “Greedy Function Approximation: A Gradient Boosting Machine,” *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.
- [18] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001
- [19] N. Cristianini and E. Ricci, “Support Vector Machines,” in *Encyclopedia of Algorithms*, M.-Y. Kao, Ed. Boston, MA: Springer US, 2008, pp. 928–932.
- [20] X. Wang and Q. He, “Enhancing Generalization Capability of SVM Classifiers with Feature Weight Adjustment,” in *Knowledge-Based Intelligent Information and Engineering Systems*, Berlin, Heidelberg, 2004, pp. 1037–1043
- [21] J. C. Stoltzfus, “Logistic Regression: A Brief Primer,” *Academic Emergency Medicine*, vol. 18, no. 10, pp. 1099–1104, 2011
- [22] “SPK: The SPICE Ephemeris Subsystem.” Navigation Ancillary Information Facility, Apr-1998.
- [23] “Provisional designation in astronomy,” *Wikipedia*, 27-Nov-2020. [Online]. Available: https://en.wikipedia.org/wiki/Provisional_designation_in_astronomy.
- [24] “Geometric albedo,” *Wikipedia*, 15-Apr-2020. [Online]. Available: https://en.wikipedia.org/wiki/Geometric_albedo.
- [25] J. Tatum, “9.9: Osculating Elements,” *Physics LibreTexts*, 13-Jul-2020. [Online]. Available: https://phys.libretexts.org/Bookshelves/Astronomy__Cosmology/Book:_Celestial_Mechanics

(Tatum)/09: The Two Body Problem in Two Dimensions/9.09: Osculating Elements.
[Accessed: 16-Dec-2020].

[26] “Equinox (celestial coordinates),” *Wikipedia*, 10-Nov-2020. [Online]. Available: [https://en.wikipedia.org/wiki/Equinox_\(celestial_coordinates\)](https://en.wikipedia.org/wiki/Equinox_(celestial_coordinates)).

[27] “Equinox (celestial coordinates),” *Wikipedia*, 10-Nov-2020. [Online]. Available: [https://en.wikipedia.org/wiki/Equinox_\(celestial_coordinates\)](https://en.wikipedia.org/wiki/Equinox_(celestial_coordinates)).

[28] “Orbital inclination,” *Wikipedia*, 18-Aug-2020. [Online]. Available: https://en.wikipedia.org/wiki/Orbital_inclination.

[29] Longitude of the ascending node. (2020, August 18). Retrieved December 16, 2020, from https://en.wikipedia.org/wiki/Longitude_of_the_ascending_node

[30] “Argument of periapsis,” *Wikipedia*, 31-Oct-2020. [Online]. Available: https://en.wikipedia.org/wiki/Argument_of_periapsis.

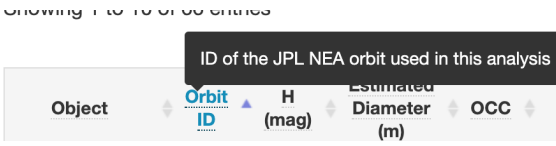
[31] Mean anomaly. (2020, October 27). Retrieved December 16, 2020, from https://en.wikipedia.org/wiki/Mean_anomaly

[32] Mean motion. (2020, June 20). Retrieved December 16, 2020, from https://en.wikipedia.org/wiki/Mean_motion

7.0 Appendix

Appendix A- Definition of Features

Feature Name	Definition With Related Links
Object ID(Column 1 in data sheet)	An ID combined with letters and digits.
SPK-ID(column 2 in data sheet)	A numerical ID code which indicates different Asteroids [22].
Object fullname	consisting of the year of discovery and an alphanumeric code indicating the half-month of discovery and the sequence within that half-month [23].
Pdes	Object primary designation
Name	Name of the Asteroid
Prefix	Prefix of the Asteroid
NEO(Near-Earth Object flag)	perihelion distance q less than 1.3 au [10].
pha(Potentially Hazardous Asteroids)	Defined based on parameters that measure the asteroid's potential to make threatening close approaches to the Earth. Specifically, all asteroids with an Earth Minimum Orbit Intersection Distance (MOID) of 0.05 au or less and an absolute magnitude (H) of 22.0 or less are considered PHAs. Asteroids that can't get any closer to the Earth (i.e., MOID) than 0.05 au (roughly 7,480,000 km or 4,650,000 mi) or are smaller than about 140 m in diameter (i.e., H = 22.0 with assumed albedo of 14%) are not considered PHAs [2].
H(absolute magnitude parameter)	<div> <p>Formula for H: (Absolute Magnitude)</p> $H = m_{Sun} - 5 \log_{10} \frac{\sqrt{a}r}{d_0}$ </div> <p>visual magnitude an observer would record if the asteroid were placed 1 Astronomical Unit (au) away, and 1 au from the Sun and at a zero phase angle [5].</p>
diameter(object diameter in KM)	Diameter of the asteroids

Albedo (geometric albedo)	the ratio of its actual brightness as seen from the light source to that of an idealized flat, fully reflecting, diffusively scattering disk with the same cross-section [24].
Diameter_sigma (1-sigma uncertainty in object diameter in KM)	1-sigma uncertainty
Orbit_id	 <p>Object ID as shown above [12]</p>
Epoch (epoch of osculation)	The "epoch" serves as a reference point from which time is measured [25].
epoch_mjd	It should be modified Julian day
epoch_cal	Calculated date, in yyyyymmdd form
Equinox (Equinox of reference frame)	an epoch is a moment in time used as a reference point for some time-varying astronomical quantity [26].
e (eccentricity)	The orbital eccentricity of an astronomical object is a dimensionless parameter that determines the amount by which its orbit around another body deviates from a perfect circle [7].
a (semi-major axis au Unit)	The semi-major axis is one half of the major axis, and thus runs from the centre, through a focus, and to the perimeter [8].
q (perihelion distance au Unit)	The perihelion is the point in the orbit of a planet, asteroid or comet that is nearest to the sun [27].
i	The inclination angle (i) is the angle the asteroid's orbit is inclined with respect to a reference plane. The reference plane for asteroids and other solar system celestial bodies is the orbital plane of earth around the sun [28].

om(omega)- Longitude of the ascending node	one of the orbital elements used to specify the orbit of an object in space [29].
w(argument of perihelion)	the angle from a specified reference direction, called the origin of longitude, to the direction of the ascending node, as measured in a specified reference plane [30].
ma	Mean anomaly the fraction of an elliptical orbit's period that has elapsed since the orbiting body passed periapsis, expressed as an angle which can be used in calculating the position of that body in the classical two-body problem. It is the angular distance from the pericenter which a fictitious body would have if it moved in a circular orbit, with constant speed, in the same orbital period as the actual body in its elliptical orbit [31].
ad(aphelion distance)	The perihelion (q) and aphelion (Q) are the nearest and farthest points respectively of a body's direct orbit around the Sun [9].
n (Mean motion)	Mean motion is the angular speed required for a body to complete one orbit, assuming constant speed in a circular orbit which completes in the same time as the variable speed, elliptical orbit of the actual body [32]
tp (Time of perihelion passage TDB Unit)	The time at which an object is at perihelion (its closest distance to the sun).
tp_cal	Calculated time yyymmdd
per	Sidereal orbit Period in day
per_y	Sidereal orbit Period in year
moid	Minimum orbit intersection distance [6].
moid_Id	Minimum orbit intersection distance in Lunar distance
sigma_e	1-sigma uncertainty

sigma_a	1-sigma uncertainty
sigma_q	1-sigma uncertainty
sigma_i	1-sigma uncertainty
sigma_om	1-sigma uncertainty
sigma_w	1-sigma uncertainty
sigma_ma	1-sigma uncertainty
sigma_ad	1-sigma uncertainty
sigma_n	1-sigma uncertainty

sigma_tp	1-sigma uncertainty
sigma_per	1-sigma uncertainty
Class	Abbreviation and description of each class [11].
rms	Normalized RMS of orbit fit.