Group 65 Progress Report: Asteroid Classification using Machine Learning

Claire Nielsen, Jake Read, Rebecca Di Filippo
{nielsc2, readj9, diflilr}@mcmaster.ca

1 Introduction

There are tens of thousands of Near Earth Asteroids (NEAs) on orbit paths close to Earth, with more being discovered daily (CNEOS, 2025a). Some of these become classified as Potentially Hazardous Asteroids (PHAs), as they become large enough to potentially cause a problem. As more and more NEAs are discovered every day, it would be helpful to develop a machine learning classification system that would be able to classify NEAs to determine which could be hazardous. Classification of asteroids requires time-consuming analysis of a variety of features. An autonomous model would allow space agencies to dedicate more resources to observation of the detected PHAs and developing mitigation strategies.

Accurate and expedient classification of asteroids is crucial to determining potential risks to planetary health. This project develops a model to classify NEAs based on physical and orbital characteristics to determine which could be hazardous to Earth. This project cleans and pre-processes the dataset, selects relevant features, and trains a model using machine learning to classify asteroids as hazardous or non-hazardous. We handle class imbalance using techniques like oversampling(e.g., SMOTE)), or weighted loss, and then evaluate the model.

The dataset we use contains an existing classification for, as well as features of thousands of discovered asteroids. The dataset is derived from NASA and JPL's Small-body and Asteroid database. The dataset contains 45 features of any known object, and we engineer a feature set that yields the best results, clean the data and handle imbalance.

2 Related Work

CNEOS (2025b) defines Potentially Hazardous Asteroids (PHAs) based on parameters that affect the asteroid's potential to be threatening, including its closest distance to Earth, size and albedo. There have been several systems developed using various machine learning models.

Results of a comparison of machine learning models to classify NEAs were published in 2022, and included comparison of logistic regression, naive Bayes, support vector machines (SVMs), gradient boosting, and MultiLayer Perceptrons (MLPs). The article states that *multilayer perception* and *gradient boosting* yielded the most accurate results (H. Klimczak, 2022).

A further indicator of the performance of gradient boosting came from a classification project using the NGBoost classifier, a gradient boosting framework that uses natural gradient descent for optimization. Using NGBoost produced an overall accuracy of 99.22% (Al Mahmud Al Mamun, 2024).

Another article published in 2019 compares the Hierarchal Clustering Method (HCM) to a newer, *supervised learning HCM* method. The supervised HCM method proved superior to the classical method, correctly identifying all asteroids within the target family and and yielding an accuracy of about 85% (V. Carruba, 2019).

A different approach was taken by Pasko (2018), who observed subgroups of NEAs with high concentrations of PHAs, to better determine which characteristics can be used as flags of hazardous asteroids. Using a *Support Vector Machine (SVM)* model, the extracted subgroups of NEAs contained about 90% of the real and virtual PHAs.

After the progress report for this project was written, more research was done to compare our model's performance to other published works. A study published in 2024 used boosting algorithms including XGBoost, LightGBM, and AdaBoost to classify asteroid families (A. Sai Vignesh, 2024). Asteroid families are determined by asteroids that have similar orbit and characteristics, that are believed to each have originated from the same parent body. The study trained a model of each algorithm. While the classification task was different, we can note that our model's F1 score is within reasonable range of the scores using similar algorithms published in this study, which can be an indicator of good performance.

3 Dataset

We are using a public dataset available on Kaggle, licensed under Open Data Commons Open Database License, by user sakhawak18. The database is called *Asteroid Dataset*, found here (Hossain and Zabed, 2023). As stated in the summary, it is officially maintained and updated weekly, and has a Kaggle-calculated usability score of 10.00. The original source of the data is NASA's Jet Propulsion Laboratory Small-Body and Asteroid databases, containing both orbital and physical properties for hundreds of thousands of known asteroids (NASA, 2025).

The dataset contains orbital and physical properties of thousands of discovered and analyzed asteroids. There is a target label pha present, representing the classification to be predicted by the model. The features our model uses are as follows:

- **Flags:** neo (Near-Earth Object flag), pha (Potentially Hazardous Asteroid flag, 1 = hazardous, 0 = non-hazardous)
- **Physical Properties:** H (absolute magnitude), diameter (km), albedo (surface reflectivity), diameter_sigma (uncertainty in diameter)
- **Orbital Elements:**
  - Core orbital parameters: e (eccentricity), a (semi-major axis), q (perihelion distance), i (inclination), om (Ω, longitude of ascending node), w (ω, argument of perihelion), ma (mean anomaly)
  - Derived orbital distances: ad (aphelion distance), moid (minimum orbit intersection distance), moid_ld (MOID in lunar distances)
  - Motion and timing: n (mean motion), tp (time of perihelion passage), per (orbital period in days), per_y (orbital period in years)
- **Uncertainties (Standard Deviations):** sigma_e, sigma_a, sigma_q, sigma_i, sigma_om, sigma_w, sigma_ma, sigma_ad, sigma_n, sigma_tp, sigma_per
- **Model Fit Quality:** rms (root-mean-square residual, indicating fit accuracy of the orbital solution)
- **Target Label:** pha (Potentially Hazardous Asteroid flag — the classification label to be predicted)

The dataset also contains identifier, metadata, epoch and reference data features that will not be used in our model. The excluded features are:

- **Identifiers and Metadata:** id, spkid, full_name, pdes, name, prefix, orbit_id, class
- **Epoch and Reference Data:** epoch, epoch_mjd (Modified Julian Date), epoch_cal (calendar format), equinox, tp_cal (perihelion passage in calendar format)

We import the dataset using the Kaggle API, complying with its terms of service. Once the dataset is imported it is preprocessed, which includes a number of steps. The dataset contains some non-numeric columns in the identifiers and metadata (including but not limited to id, name, and class) that are dropped as they cannot contribute to prediction. Some categorical categories, including the pha flag, are converted to numerical binary values and processed as such. Any string values remaining in the dataset are forced to NaN to ensure that no unpredicted errors occur in processing. The resulting missing values are then handled by converting them to the column mean.

4 Features

As mentioned, the model handles most but not all features provided by the dataset. This is because the dataset provides data that is not required for predictive purposes, which in this case includes identifiers, metadata, epoch and reference data. The included features contain the data that is most relevant to the categorization of an NEA as hazardous.

NASA categorizes PHAs based on the value of moid (minimum orbit intersection distance), H (absolute magnitude), and albedo (surface reflectivity). The minimum orbit intersection distance is the closest the asteroid will come to Earth on its natural orbit path, with hazardous asteroids having a moid ≤ 0.05au, or astronomical units (149,597,870,700 m, which approximates the distance between the Earth and the sun). The absolute magnitude of an asteroid represents the visual magnitude an observer would record if the asteroid were placed exactly 1au away and 1au from the Sun at a zero phase angle (effectively measuring size). PHAs are characterized by an absolute magnitude H ≤ 22. This magnitude translates to asteroids that are no more than 140m in diameter, and corresponds to an assumed albedo = 0.14. The albedo of an asteroid represents its ratio of the light received to light reflected by that body, ranging from 0 (pitch black) to 1 (perfect reflector). Our model takes in these features as well as the others to produce an accurate prediction of the pha flag.

We are using feature selection in our model. Feature engineering involves creating new features based on overlapping/redundant features from the original dataset. We may have some very similar features, but have chosen not to engineer them for use with this model. Our *feature_engineering* class contains code to select which feature sets to use. We are not using embedding, as any relevant data to the prediction is represented numerically, and we do not have to process string data or images.

5 Implementation

When we started this project, our plan was to use a Gradient Boosted Trees model. We chose this model for a number of reasons. From research it seems like they handle structured tabular data well, manage non-linear relationships, and allow weighting for class imbalance, all of which apply to our dataset. Furthermore, research into existing solutions and related work confirmed that Gradient Boosted Trees have potential for and already have highly accurate results when classifying PHAs.

The model we have implemented is a Histogram-based Gradient Boosting Classifier (HGBC). This model outperforms traditional Gradient Boosting with large sample sizes, which applies to this project. This will be referred to as a Main Model. Originally, the Main Model used a logistic loss function for classification. Since the Progress Report, we have optimized the model to prioritize Recall instead of the F1 score. In order to accomplish this, we have switched to an F2 loss function.

One way we have noted our model's performance is by comparing it to similar models, one of which was cited in the Related Work Section. The models created by A. Sai Vignesh (2024) also use boosting algorithms to classify asteroids, this time classifying by family instead of hazardous-non-hazardous. Our model yields highers scores than several of their models, with the only exception being that the Precision and F1 Scores of the lightGBM model outscoring ours. Overall, this is a fantastic marker of performance and the team is content with our results.

Histogram-based Gradient Boosting Classifiers are much faster than traditional Gradient Boosting Classifiers (GBCs). GBCs require sorting of all samples present for each feature, resulting in high node splitting complexity of $O(n_{features} * n_{samples} \log n_{samples})$ at every node. HGBCs do not use the costly operation of sorting samples and instead reference an implicitly sorted histogram. Building a histogram has complexity $O(n_{samples})$, so the node splitting complexity of HGBCs is $O(n_{features} * n_{samples})$, which is much lower (scikit learn).

The program is run by running main.py, with potential arguments:

- -ht, to run hyperparameter tuning before training the model.
- -fs, to change the feature set used for training. Any combination may be used, default is all.

- `-ms`, to specify a save path for the trained model.
- `-ml`, to specify a save math to load the trained model from.

Assuming the program is run with default arguments, the data is then downloaded (or loaded from existing raw save) and preprocessed (or loaded from existing preprocessed save). The data is then split into training and testing data, with 80% allocated to training and 20% allocated to testing. Since the progress report, we have implemented a stratified split, meaning when the data is split we ensure that there is a similar ratio of positive and negative classifications (hazardous and non-hazardous) in each split. After the test-train split, it is normalized.

If desired, the hyperparameter tuning is then run on the Main Model, which finds the best parameters of the model by random search. This yields tuned hyperparameters for learning rate, maximum tree depth, and maximum number of boosting iterations (trees).

Each time the program runs a Baseline Model is set by majority vote. The Baseline Model predicts the majority class in the dataset by counting which classifier occurs most frequently. Since the vast majority of samples are classified as non-hazardous, the Baseline Model predicts 0 for all samples. This Baseline Model exists only to establish a very weak baseline so that we can see that our Main Model performs better. If a Main Model has been previously trained, it is then loaded. If there is no existing model file, one is trained and saved for future use. The Main Model and Baseline Model are then both used to predict the testing data that was set aside during the earlier 80-20 train-test split. The predicted data is then evaluated, and the models' results are compared.

Implementation of the Main Model has changed slightly since the progress report. Below is a list of the changes that have been made and why.

- Swapped to balanced class weights from sklearn, which automatically upweights the rarer positive (hazardous) class. This change helps the Model prioritize classification of hazardous asteroids since they are weighted heavier.
- Switched hyperparameter tuning to optimize for the F-beta (F2) metric, which has a higher focus on recall.
- Implemented Random Over Sampling (ROS), which duplicates instances of the positive class so they appear more often in the dataset. This makes the rarer case more common so that there can be a stronger bias towards classifying an asteroid as hazardous.
- Switched to a stratified train-test split to ensure a similar ratio of positive to negative instances in the train and test splits.

6 Results and Evaluation

At the Progress Report stage, our model performed quite well. As mentioned, we use and are continuing to use a train-test split of 80% training data and 20% testing data, with cross-validation used for hyperparameter tuning. Since the progress report, we have switched to a stratified train-test split, which ensures there is a similar ratio of positive to negative classes in each split. We are evaluating our model's performance based on accuracy, precision, recall, and resulting F1 score, as well as through a confusion matrix. For the Progress Report, we evaluated the model's performance on F1 score, since it represents a harmonic mean of Accuracy, Precision and Recall scores. Since then, we have switched to evaluate the model's performance on Recall to prioritize correct classification of the rarer hazardous asteroids.

One experiment the team conducted was testing different feature sets to be used to train the Main Model. Pictured below is a comparison of F1 scores found through training the Model with different feature sets.

This graph demonstrates that training with all features produced the best resultant F1 score. The same can be said for Accuracy and Precision. However, Recall was ever so slightly lower on this feature set than the model trained with the "Orbital Derived, Physical, and Orbital Core" feature set.

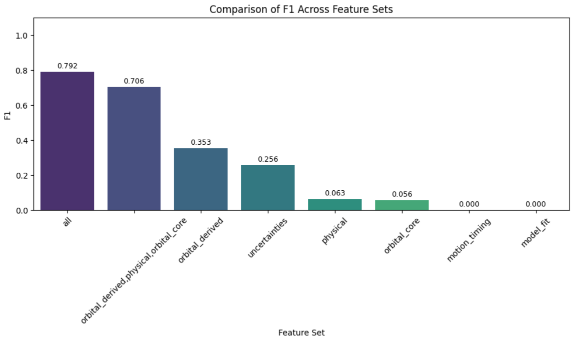

Figure 1: Bar graph comparing F1 Score per Feature Set.

After this test was conducted, the team implemented the changes to make the Model prioritize Recall optimization over F1 score. This was done to prioritize correctly classifying all hazardous asteroids, as they represent a safety risk and should be identified. To do this, several changes were implemented, and are described in the Implementation Section. As demonstrated in the progress report, we can represent the performance of the Majority-Vote Baseline Model compared to the new Main Model in a bar graph:

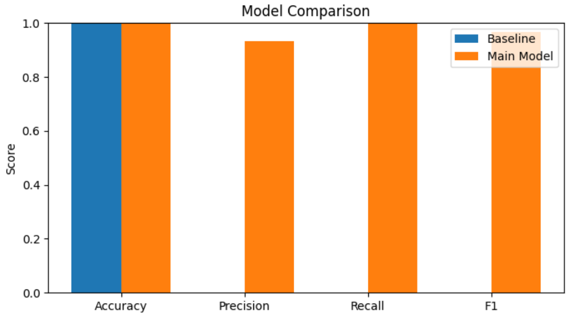

Figure 2: Bar graph comparing the Baseline Model's performance metrics to the Main Model after Recall Optimization.

In this graph, we can see that the Baseline Model has incredibly high accuracy, and almost zero precision, recall, and F1 score. Since the baseline classifies all samples as non-hazardous, it results in a high accuracy score since most samples are correctly classified. However, since it incorrectly classifies all hazardous asteroids, every other score is zero. However, we can also see that our Main Model performs quite well. Compared to the graph generated in the Progress Report, we can see that every score has improved.

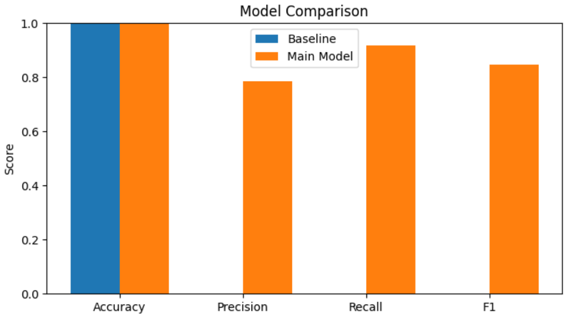

Figure 3: Bar graph comparing the Baseline Model's performance metrics to the Main Model in the Progress Report.

The switch to prioritizing Recall led to an improvement across all scores. The confusion matrices produced by the model provide a visual depiction of predicted and actual true classification labels. Below is the confusion matrix generated with the Progress Report model, followed by the new Model:

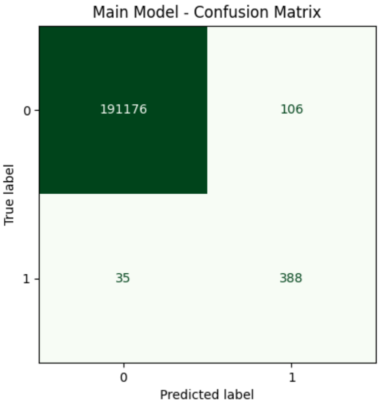

Figure 4: Confusion Matrix produced by Main Model at Progress Report Stage.

The bottom left quadrant represents asteroids that were hazardous but predicted to be non-hazardous. Since this quadrant is empty the updated model correctly classified **all** hazardous asteroids. We can also see a decrease in non-hazardous asteroids being classified as hazardous, and an increase in correctly identified hazardous asteroids.

The final trial the team ran was attempting more rigorous hyperparameter tuning, to see if additional tuning would yield more optimized hyperparameters and eventually results. Trials were completed with 10, 100, and eventually 500

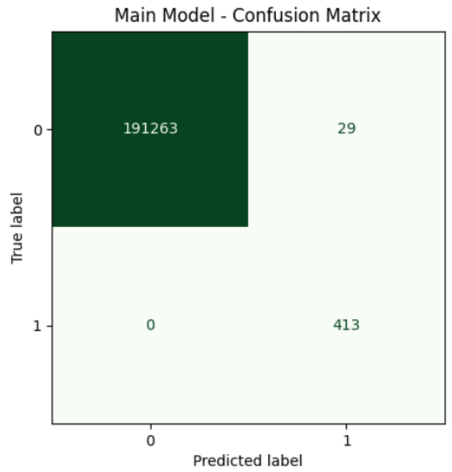

Figure 5: Confusion Matrix produced by Updated Main Model.

runs. The hyperparameters fluctuated slightly, but the results of the Main Model stayed the same up to 1000ths of a percent. Recalling the progress report, tuning the model with 10 runs did improve results, but given the results did not change in between longer trials we can infer that 10 runs of hyperparameter tuning is sufficient.

7 Reflection and Progress

Since the progress report, the team has addressed the plans for future improvements and changes. Listed below are the planned changes, and how they have been implemented.

- We will work on refining our model by running with different feature sets to see which have the greatest impact on its performance.
- We will experiment with more rigorous hyperparameter tuning.
- We will consider a focus on recall instead of F1 score.
- We will work with different techniques to handle the class imbalance due to the much smaller number of hazardous asteroids in our dataset.

The results and analysis of the model's F1 score using different feature sets is above in Figure 1, and resultant Precision and Recall across feature sets is below in Figures 6 and 7. The conclusion of these trials was that running with all features produces vastly superior results, and decline in scores seems to follow as the feature set gets smaller. The improved model was run with much more extensive hyperparameter tuning trials. While it did find new best hyperparameters, the model's metrics did not change substantially. Due to the nature of our classification task, we've considered harsher penalties for missing a positive (hazard) classification, since it is hazardous. By prioritizing recall we saw **zero** missed hazardous asteroids. Class imbalance was addressed in part while prioritizing classification of hazardous asteroids. The team switched to balanced class weights from sklearn, which automatically upweights the rarer positive class. We also implemented random over sampling (ROS), which duplicates instances of the positive class so they appear more often in the dataset. The model incorrectly predicted more asteroids were hazardous than before this change, however the lack of missed hazardous asteroids means that the outcome of this change was more positive than negative.

The results demonstrated in the progress report were satisfactory, but could be improved upon. Our hypothesis was that it was possible that omitting some features could reduce false positives, increase correctly classified negatives (non-hazardous), or have other effects on the results. By training with several additional feature subsets, the team can now be confident that using the full set of features yields the highest Accuracy, Precision, and F1 score compared to those that were trained using smaller feature sets. By training to prioritize recall, the model now correctly classifies every hazardous asteroid. While some scenarios may not prioritize this, given the hazardous nature of the asteroids we believe that these results are the most important; furthermore they can be achieved while improving all metrics (Accuracy, Precision, Recall and F1 score).

8 Error Analysis

At the conclusion of the project, we are prepared to present final results produced by the Model, as well as analysis of patters in error discovered through the results. First and foremost, our model performed very well. These are the results from a trial run with the finalized model prioritizing recall, compared to the results displayed in the Progress Report:

Metric	PR Score (%)	Final Score (%)
Accuracy	99.96	99.98
Precision	91.42	93.44
Recall	88.18	100.00
F1 Score	89.77	96.61

Table 1: Comparison of Main Model with and without Hyperparameter Tuning.

Our model now excels at correctly classifying hazardous asteroids, which is accomplished by prioritizing recall in training. We believe that this is an important consideration, given the hazardous nature of the positive class, it is important that they are identified so that further action may be taken. Referencing the confusion matrix in the Figure 5, the only mistake the model made is misclassifying 29 non-hazardous asteroids as hazardous. It is worth noting that before prioritizing recall, the model misclassified 106 asteroids as hazardous. This demonstrates that even though we wanted the model to err on the side of "better safe than sorry", the changes we made to implement this have increased overall accuracy and confidence.

There were patterns that arose when testing the model with different feature sets. When using any subset other than the full complete feature set, there was a noticeable drop in all scores produced. The comparison of F1 Scores across feature sets is pictured above in Figure 1, below are two additional graphs representing comparison of Precision and Recall across all feature sets. Please note these trials were done before the prioritization of Recall, which is why the scores for the full feature set do not match the quoted final scores.

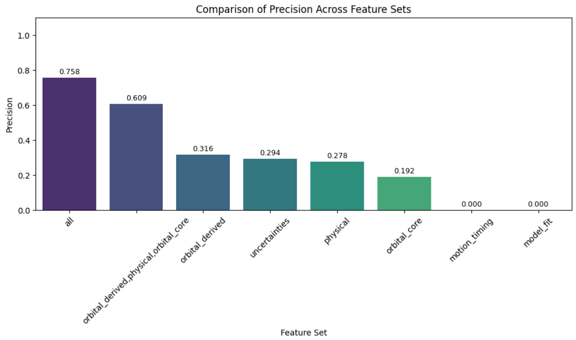

Figure 6: Bar graph comparing the Precision across all tested Feature Sets.

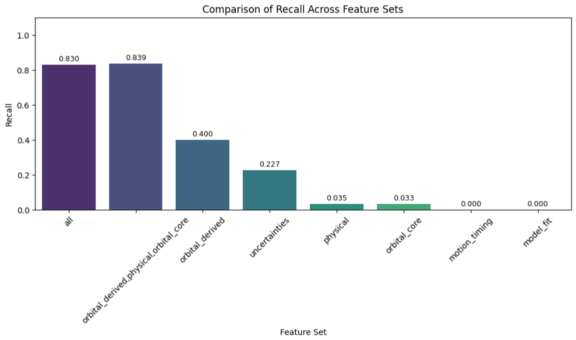

Figure 7: Bar graph comparing the Recall across all tested Feature Sets.

These graphs demonstrate that using the full feature set is vastly superior to any subset. Using the full feature set or the Orbital Derived, Orbital Core and Physical features (ODOCP) yielded decent results, there is a steep decline when using only one feature, and scores of 0 when using only the Motion and Timing feature or the Model Fit feature. Therefore, we can conclude that performance is significantly increased by increasing the number of feature sets considered, with Orbital Derived, Uncertainties, Physical Properties, and Orbital Core providing the most impact. When tested, the model trained on the ODOCP Feature set incorrectly classified non-hazardous asteroids as hazardous (false positive) more frequently than the model trained on all features. This is why we see comparable Recall results but much poorer Precision results on the above bar graphs.

While we have come a long way and have an excellent model to show for it, there is always more that could be done. If we were to continue working on this model, we could attempt to address these errors by improving the performance of the model. Further work could be done to divide feature sets, to see if removing a smaller group helps or hinders performance. We could also run further hyperparameter searches, or train multiple models in parallel from different starting points to select one with the best predictions. All of these options are good for future development, as they require much more time and computational power. This could be good optimal extension work, as it could provide very marginal gain but comes at a very high cost. All in all, the results are already excellent so we are confident in the final model we have produced.

Team Contributions

Claire Nielsen handled the research into related works and strategies, and wrote the Progress Report.

After the Progress Report, Claire conducted research into additional related works and compared the team's model's performance to other models. Claire summarized results from trials and changes after the progress report, performed the error analysis of the project, and wrote the Final Project Report.

Rebecca Di Filippo and **Jake Read** handled the coding of the model up to this point in the project. Rebecca handled the preprocessing of the dataset, feature engineering and selection, and Baseline Model for comparison. Jake wrote the starting gradient boosting algorithm, loss function, and optimization technique. Rebecca and Jake collaborated on the evaluation strategies for the model (including the training/testing split, cross-validation, and metrics).

After the Progress Report, Rebecca ran several trials to train the model with different feature sets to see if any could improve the results. Rebecca also ran more intensive hyperparameter tuning to demonstrate how more tuning runs can improve overall results. After the Progress Report, Jake optimized the model to prioritize recall over F1 score, ensuring that fewer hazardous asteroids would be misclassified. Jake then ran trials with the new model to ensure the prioritized recall improved results.

Repository Link

Our GitHub repository can be found here.

List of Figures

List of Tables

References

M. Sai Pravardhitha M. Sam Kennanya A. Sai Vignesh, T. Meena. 2024. Asteroid family classification using boosting algorithms. In *2024 Asia Pacific Conference on Innovation in Technology (APCIT*, Mysuru, India.

Md Rasel Hossain Mst Mahfuza Sharmin Md Ziaul Haque Al Mahmud Al Mamun, Md Ashik Iqbal. 2024. Near-earth asteroids classification using ngboost classifier. *Material Science and Engineering International Journal*, 8.

CNEOS. 2025a. Discovery statistics. Technical report, NASA Jet Propulstion Laboratory, Centre for Near Earth Observation Studies.

CNEOS. 2025b. Neo basics. Technical report, NASA Jet Propulstion Laboratory, Centre for Near Earth Observation Studies.

B. Carry A. Penttilä W. Kotlowski A. Kryszczyńska E. Wilawer H. Klimczak, D. Oszkiewicz1. 2022. Comparison of machine learning algorithms used to classify the asteroids observed by all-sky surveys. *Astronomy and Astrophysics*, 667.

Mir Sakhawat Hossain and Md. Akib Zabed. 2023. Machine learning approaches for classification and diameter prediction of asteroids. In *Proceedings of International Conference on Information and Communication Technology for Development*, pages 43–55, Singapore. Springer Nature Singapore.

NASA. 2025. Small-body database query.

Vadym Pasko. 2018. Prediction of orbital parameters for undiscovered potentially hazardous asteroids using machine learning. In *Stardust Final Conference, Advances in Asteroids and Space Debris Engineering and Science*, pages 33–40.

scikit learn. *Ensembles: Gradient boosting, random forests, bagging, voting, stacking*.

A. Lucchini V. Carruba, S. Aljbaae. 2019. Comparison of machine learning algorithms used to classify the asteroids observed by all-sky surveys. *Monthly Notices of the Royal Astronomical Society*, 488:1377–1386.