Classifying Hazardous and Non-Hazardous Asteroids Using Machine Learning

Trotter McLemore

Received October 17th, 2022
Accepted December 9th, 2022
Electronic access December 31st, 2022

Asteroids present a real danger to humanity, and though a serious collision has not occurred in the recent past, that does not mean one will not happen in the future. The recent DART mission by NASA is one example of how a collision could be avoided, but the first step in a similar future mission would be identifying a potentially hazardous asteroid. This currently has to be done by hand, but could be done more efficiently by a machine learning algorithm. This project examines how to use machine learning models to predict if an asteroid is hazardous to Earth based on the NASA Near Earth Object Web Service (NeoWs) dataset. Before building the models, the data was preprocessed by dropping outliers and unnecessary columns. Four models were tested: a logistic regression, a support vector machine, a random forest classifier, and XGBoost. XGBoost had the best results across all three metrics measured, with an accuracy of 94.46%, 86.86% precision, and 77.78% recall. A model with such high accuracy could identify an asteroid early, before it became a true threat, giving NASA or other organizations time to prepare a defense mechanism.

Introduction

Throughout the history of the Earth, asteroids have been impacting its climate and inhabitants. Examples from the past, like the Cretaceous–Paleogene extinction event, hypothesized to have been caused by an asteroid, prove that though extinction-size asteroids are not common, a large asteroid on a collision course with Earth would have disastrous consequences[1,2]. Additionally, smaller asteroids can have an effect on us[3]. The 2013 Chelyabinsk meteor airburst, though it was small compared to many other asteroids in our solar system, caused over 30 million dollars in damage and injured over 1,000 people[4]. These smaller sized asteroids are far more common, and can damage satellites, space stations, and even buildings on Earth.

Fortunately, NASA has been developing methods to deflect asteroids from Earth, lowering the risk of these asteroids damaging human structures or colliding with Earth[5]. One example of this is the recent DART mission, where a spacecraft hit an asteroid to alter its course. This proved that an asteroid could be redirected from Earth if needed[6]. Though other asteroid protection techniques exist, such as the strategy detailed by Fernandez 2021 that calls for large metal rods to fragment the asteroid, NASA claims that a mission similar to DART is the safest and most realistic way to protect Earth from an asteroid[7,8]. However, before NASA can use their techniques, they first have to identify hazardous asteroids early, before they start becoming a true threat.

Currently, much of the necessary calculations and data-crunching have to be done by hand, but an accurate machine learning model could help greatly reduce the amount of work that has to be done. This project aimed to create such a model on the NeoWS asteroid dataset that performed well enough that it could take the place of human calculations. Discussed in this paper are four different machine learning models. These models included a logistic regression, a support vector machine, a random forest classifier, and an XGBoost model. To determine which model performed the best, GridSearch Cross Validation was used to test many hyperparameters, and then the accuracy, precision, and recall of each model was found based on these parameters. After testing many combinations of parameters on all four models, the XGBoost model proved to perform the best, with an accuracy of 94.46%, 86.86% precision, and 77.78% recall on the test set.

A previous model by Rabeendran & Denneau 2021 has explored this area with similar results, creating a neural net that identifies hazardous asteroids based on the ATLAS dataset[9]. Though this model is fairly similar to the model presented later in this paper, the fundamental difference between the two lies in the dataset. The ATLAS dataset is a collection of images, compared to the NeoWs dataset used for this model, which is completely numerical. Neither dataset or model is valued above the other; they work in different situations.

Another model, presented by Bahel et al. 2021, also uses ML to identify hazardous asteroids, but again, the difference between the past model and the model discussed in this paper is the dataset it was trained on[10]. Bahel et al. use a dataset that collects different features than NeoWs, leading to a model that

uses specific features only found in the dataset it was trained on.

Altogether, the reasons for the model presented in this paper are twofold: first, to create a model trained on a new dataset, and second, to create an XGBoost model. Creating a model trained on the NeoWs dataset has not been done and published previously, and doing so helps to fill a void in the scientific communities' understanding of asteroids. An XGBoost model has the possibility of becoming more accurate than the random forest model presented by Bahel et al. due to its power to iteratively become more accurate (this is discussed in the methods section), while also using less computing power than the neural net used by Rabeendran and Denneau[11]. In the future, all three models will be able to work together to provide a more complete picture of asteroid hazardousness based on the specific situation and data available.

Data

Data Source

The data used for this project is a dataset from the NASA service called "NeoWs", short for Near Earth Object Web Service[12]. NeoWs was created to give the general public access to astronomical data about near Earth objects. This specific dataset only contains information about asteroids, of which there are 3,749 included in the table. Outside of hazardousness, every variable collected either related to the asteroid's orbit (e.g. orbital period, relative velocity) or its size (e.g. estimated diameter, absolute magnitude). Together, the table shows a complete picture of every asteroid, including its size, movement, and future movement. However, in the context of a machine learning model, each feature had a different level of importance, discussed in the next subsection.

Exploratory Data Analysis

In exploratory data analysis, it's important to find out which variables are useful and which are not[13]. This can improve the model's performance by forcing it to focus on truly important variables, rather than wasting valuable computing power on variables that have no correlation with asteroid hazardousness. Shown in Figure 1 is a computed correlation matrix, allowing one to visually see which variables should be discarded and which should be retained. Additionally, Figure 1 allows one to see which variables have such a high correlation that only one needs to be kept. This is explained further in the Pre-Processing section.

As seen in the bottom row of the correlation plot in Figure 1, no feature had a high (> 0.5) correlation with hazardousness, so there weren't any variables that initially showed themselves to be very important to the model. However, outside of correlations, there were multiple pieces of information that were useful when building the model. There are a total of 3,398 asteroids in the training set, with 2,889 being non-hazardous and the remaining 509 being hazardous. In the test set, there are 938 total asteroids, with 153 hazards and 785 non-hazards. These values show a heavy imbalance towards the non-hazardous side, a problem addressed in later sections. Additionally, about 10% of features had correlations over 0.4 with other features, something that is relevant to preprocessing.

Data Split

The data was split into training and testing data using an 80/20 split. Additionally, the random state of the split was set to the arbitrary number 31 across all models, ensuring that there was no difference in the data the different models trained and tested on[14]. This was necessary because the random state of the split determines what number the computer bases its "randomness" on. Using a new number each trial would mean that each model got a slightly different training and testing dataset, so the results could not be compared. Using the same random state number across trials and models means that the results between them can be fairly compared, because each model gets the exact same training and testing split.

Methods

Preprocessing

The original dataset had exactly 40 columns, but many of them were redundant or unnecessary. Many of these columns, such as Est Dia in KM(min), Miss Dist.(lunar), and Relative Velocity km per hr, were the exact same as other columns, but with different units. The rest of these columns, however, had such high correlations (over .70, visualized in Fig. 1 provided in the exploratory data analysis section) with other variables that it was unnecessary to keep both columns. The .70 correlation cutoff was chosen because after testing the logistic regression model on multiple cutoffs, the results were best on the .70 cutoff. 24 columns were dropped, meaning the standard and final models used a dataset with only 16 columns. Here is a list of the final 16 columns: Neo Reference ID, Est Dia in M (min), Jupiter Tisserand Invariant, Asc Node Longitude, Perihelion Time, Mean Anomaly, Perihelion Dist, Eccentricity, Epoch Date Close Approach, Relative Velocity KM per sec, Miss Dist (Kilometers), Orbit Uncertainty, Perihelion Arg, Minimum Orbit Intersection, Inclination, Hazardous.

In addition to dropping unnecessary columns, outliers in the Est Dia in M(min) column were removed. This column originally had quite a few outliers, which affected the model's performance. After testing a logistic regression model on data

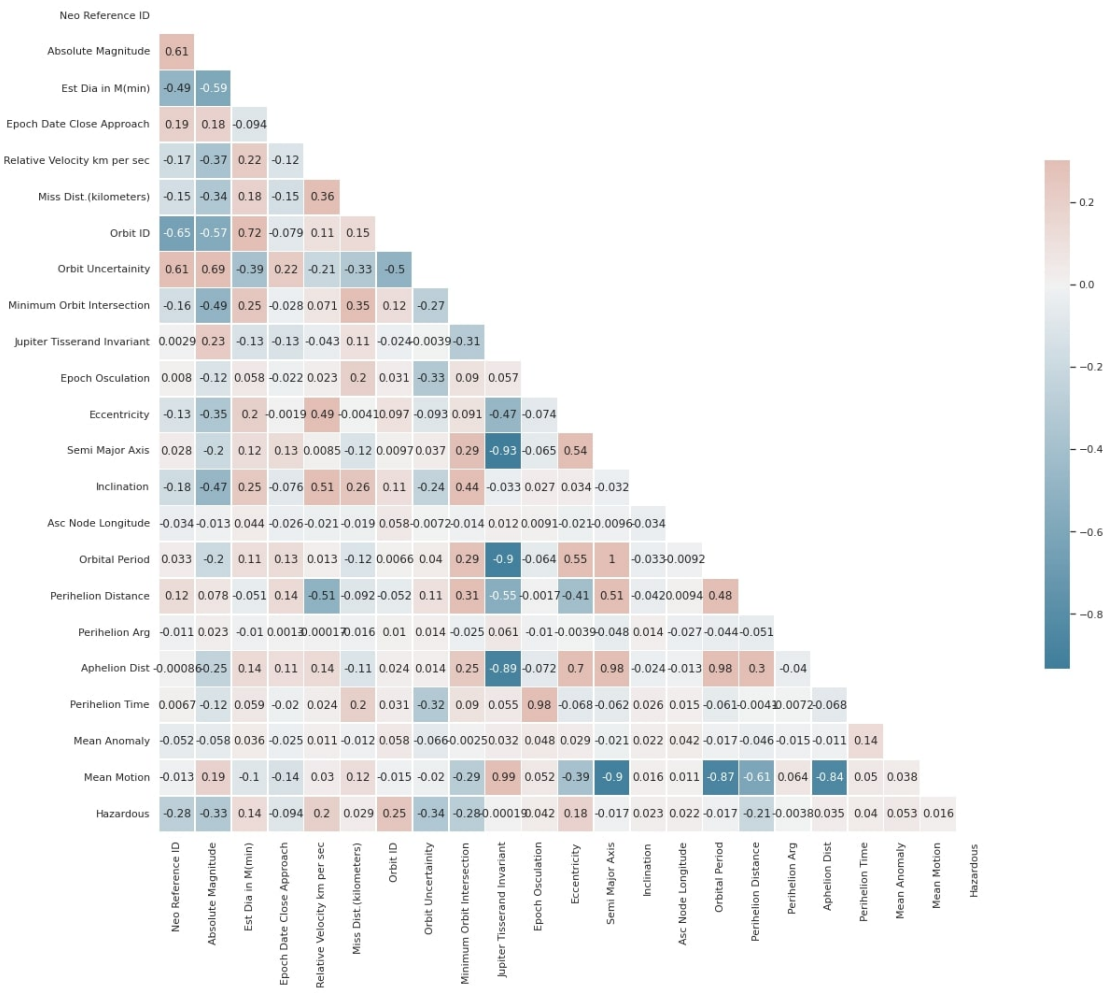

Fig. 1 Correlation Plot for Asteroid Data

with and without diameter outliers, the model proved to perform better without any outliers in the training data. However, outliers in the test set were not removed, because it was important to see how the model performed on all data, including outlier data.

The final pre-processing step was to normalize the data across all variables. Normalizing data, though not necessary for all models, helped the models perform better, leading to the decision to normalize the data on every model before training.

Standard Models

Three standard models were created, with each model generally performing better than the last. GridSearch Cross Validation (CV) was used to find the ideal hyperparameters of each model [15]. This tool requires an input of different hyperparameters to test, which vary depending on the model. After the inputs were set, while running the cross validation, the program separated the training set of data into a second, smaller training set and a validation set. It trained every combination of given parameters, and then tested every combination on the validation set it created. After iterating through this process between two and ten times, depending on the CPU usage of the model, the program outputted what the best-performing parameters were. This practice of testing different parameters helps solve the imbalanced data problem in the NeoWs dataset. Some parameters help the model weigh the smaller, hazardous part of the data extra, lessening the impact of the imbalance. These ideal hyperparameters were used to train the real model, which was then tested on the test data using these hyperparameters to get the true metrics of the model.

The first standard model created was a logistic regression model. Logistic regression is a simple method to classify data by estimating the probability of a classification based on the

input data and set parameters[16]. For this model, the penalty and cost parameters were tuned. Since logistic regression is a simpler model, 10-fold validation was run.

The second standard model tested was a Support Vector Machine (SVM). An SVM model works by mapping the input data in an n-dimensional space, allowing non-linear data to be more easily categorized[17]. The cost, kernel, degree, and gamma parameters of this model were tuned. This model was very CPU-intensive, so only 2-fold validation was possible within a reasonable amount of time.

The third and final standard model created was a random forest classifier (RFS) model. A random forest classifier works by taking small subsets of the data and creating many decision trees in parallel, with the final classification being a combination or average of the individual trees[18]. This process is also called "bagging". Though this model performed very well without any hyperparameter tuning, tuning the *nestimators* and *maxfeatures* parameters provided a slightly better accuracy. 3-fold validation was run for this model.

XGBoost Model

After creating three standard models, a final model was created to improve accuracy, precision, and recall: XGBoost. XGBoost models work by performing a process called "boosting", where a single weak model is continuously improved by going through multiple iterations[11]. This technique is very powerful because it allows a model to build upon past mistakes and become more accurate. Though this model's methods are different, the process of using GridSearchCV and tuning hyperparameters was very similar to that of the standard models. The *nestimators*, *learningrate*, *maxdepth*, and *colsamplebytree* parameters were tuned for this model. Since so many different parameters were being tuned, only run 3-fold validation was able to be run.

Table 1 shows the accuracy, precision, and recall of each model on the test set, using the ideal hyperparameters found with GridSearch Cross Validation. For the most part, each model improved from the last, leading to the XGBoost model. This final model had an accuracy of 94.46%, a precision of 86.86%, and a recall of 77.78%. All of these metrics are a large improvement from the first logistic regression model, but recall had the largest improvement. In the context of the data, a low recall means that the model is misidentifying many hazardous asteroids as non-hazardous, which could be very dangerous. Additionally, a low recall would mean that the model was not getting enough data from one of the classification groups, whereas a higher recall shows that the data imbalance between non-hazardous and hazardous is not affecting the model. So, the large gain in recall across the models is very beneficial to the final model.

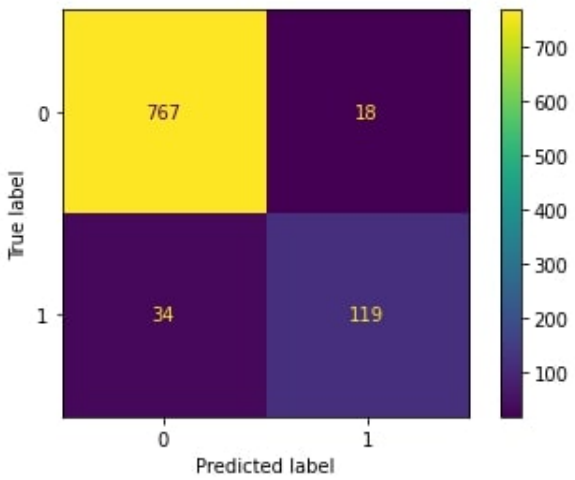

Fig. 2 XGBoost Confusion Matrix

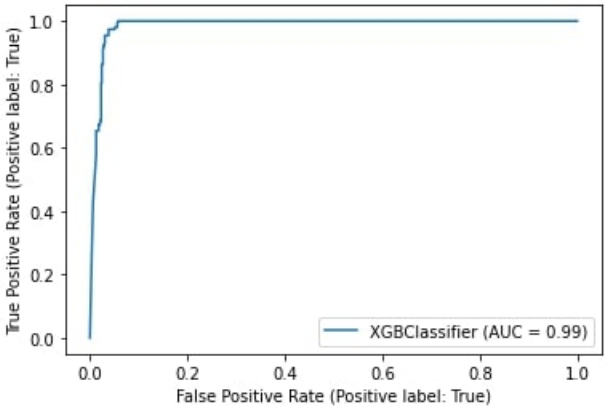

Fig. 3 XGBoost ROC Curve

The relationship between accuracy, precision, and recall can be visualized in Figure 2, a confusion matrix. The bottom left corner shows recall, asteroids that were hazardous but the model predicted would not be hazardous. The top right corner shows precision, asteroids that weren't hazardous but the model predicted were. Although it is beneficial to have a high precision, in the case of the NeoWs data, recall is much more important, because it could lead to hazardous asteroids not being identified. The other two boxes show every asteroid that the model predicted correctly. Figure 3 is an ROC curve, showing the true positive rate (hazards that the model said were hazards) and false positive rate (non-hazards that the model said were hazards) for different classification thresholds. If it was necessary to improve the true positive rate (similar to recall), one could use an equation to find the best threshold that maximizes true positives while minimizing false posi-

Table 1 Results on the Test Set.

	Accuracy	Precision	Recall	Hyperparameters
Logistic Regression	0.90618	0.78761	0.58169	Penalty = None
Support Vector Machine	0.91364	0.76086	0.68627	C = 100, Gamma = 0.1, Kernel = rbf
Random Forest Classifier	0.93496	0.85937	0.71895	Max features = None, N estimators = 100
XGBoost	0.94456	0.86861	0.77777	Learning rate = 0.05, Colsample bytree = 1, Max depth = 6, N estimators = 100

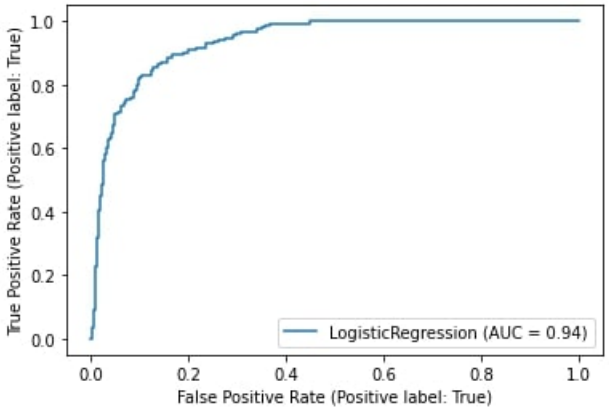

Fig. 4 Logistic Regression ROC Curve

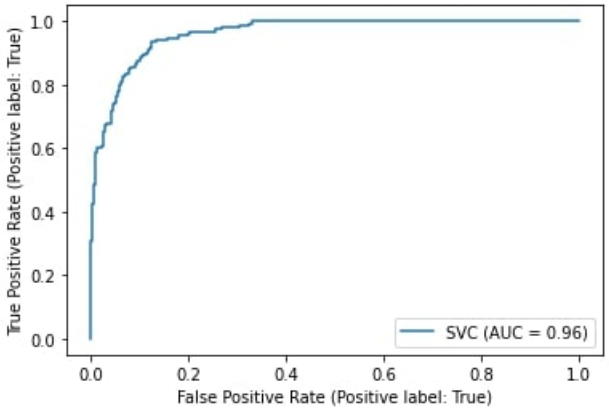

Fig. 5 SVM ROC Curve

tives, and apply that threshold to the model. Figures 4 through 6 show the ROC curves for the three standard models.

1 Discussion

1.1 Limitations

The final model had a few limitations based on the computational power it had access to. The first limitation was the number of iterations that cross validation could be run for. Though the goal was at least five iterations for every model, most of the models required so much processing power that only two or three iterations could be run in a reasonable amount of time. More iterations would have improved the model, though probably not significantly [19]. The second limitation was the number of hyperparameters that could be tuned. Each model had many more parameters than what were included in the tuning, and including them would have greatly increased the model's CPU usage and run time. Again, though, tuning more hyperparameters would most likely not have led to a marked improvement in the standard or final models [19]. Additionally, the model itself is limited in its predictions by the pre-processing steps taken. The decision to cut out outliers means that the model will likely not perform well when given real data points similar to the outliers that were cut.

2 Conclusion

The goal of this project was to build multiple models on the NeoWs asteroid dataset and create a model that performed well across all three measured metrics: accuracy, precision, and recall. Three standard models were built, in addition to a final model, with each successive model performing better than the last. To do this, the data was first split into 80% train data and 20% test data. Then, each model was trained on the training set using GridSearch Cross Validation to find the ideal hyperparameters. Finally, each model was tested on the test set using those hyperparameters to find the results. The first model made was a logistic regression model. This model had an accuracy of 90.62%, a precision of 78.76%, and a recall of 58.17%. The second model made was a support vector machine, which had 91.36% accuracy, 76.09% precision, and 68.63% recall. The third model was a random forest classifier, which had an accuracy of 93.49%, a precision of 85.94%, and a recall of 71.89%. The final model, an XGBoost model, had an accuracy of 94.46%, a precision of 86.86%, and a recall of 77.78%. Though the three standard models didn't perform poorly, the final, XGBoost model, had the best results.

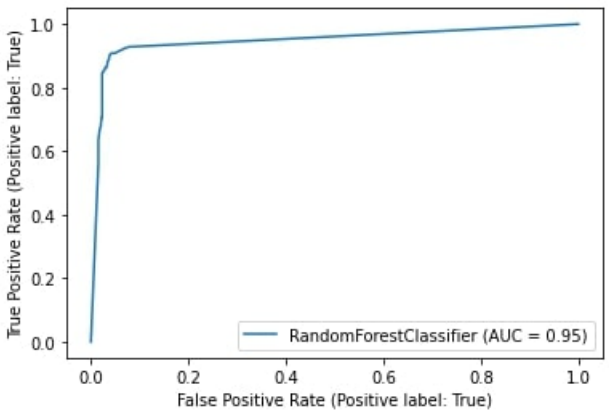

Fig. 6 RFS ROC Curve

The model could work together with other models, helping give access to the most amount of information about asteroids as possible. Already discussed in the introduction were the works by Rabeendran Denneau and Bahel et al., which detail a similar model but trained on different data. Other works by Carruba, et al. 2021 and Djorgovski, et al. 2014 use machine learning in an asteroid focused setting, but their models have different goals[20,21]. However, the model works in tandem with these projects, allowing us to have a complete picture of an asteroid by knowing its hazardousness, its movement, and its relationship to planets other than Earth.

Though there were a few limitations because of processing power, the final model still performed well enough to take the place of human calculations if needed. Additionally, the model is trained on data that does not already have a published ML paper about it, helping to fill a niche in the scientific community and be able to work in situations in which other models may not.

3 Acknowledgments

I'm extremely grateful to my mentor, Prathm Juneja, for his patience, knowledge, and guidance throughout this project. It would not have been possible without him. Additionally, I would like to thank Tyler Moulton, who helped shape this paper into what it is today.

References

1 M. Aftosmis, W. Spurlock, L. Wheeler and J. Dotson, *High-Fidelity Blast Modeling of Impact from Hypothetical Asteroid 2021 Pdc. ads*, https://ui.adsabs.harvard.edu/abs/2021plde.confE.261A.

2 L. Alvarez, W. Alvarez, F. Asaro and H. Michel, *Science*, **208**, 1095–1108.

3 E. Ryan and W. Ryan, *Ground-Based Near-Earth Object Studies in the post-Russian (Chelyabinsk) Meteor Airburst World. ads*, https://ui.adsabs.harvard.edu/abs/2013amos.confE.101R.

4 A. Kuzmin, *Reuters.https://www.reuters.com/article/cnews-us-russia-meteorite-idCABRE91E05Z20130215*.

5 T. Statler, S. Raducan, O. Barnouin, M. DeCoster, S. Chesley, B. Barbee, H. Agrusa, S. Cambioni, A. Cheng, E. Dotto, S. Eggl, E. Fahnestock, F. Ferrari, D. Graninger, A. Herique, I. Herreros, M. Hirabayashi, S. Ivanovski, M. Jutzi and K. Wünnemann, *After DART: Using the first full-scale test of a kinetic impactor to inform a future planetary defense mission*, https://doi.org/10.48550/arXiv.2209.11873, arXiv.

6 N.A.S.A., *DART: Double Asteroid Redirection Test. jhuapl*, https://dart.jhuapl.edu/Mission/index.php.

7 S. Fernandez, *How to protect Earth from incoming asteroids, according to experts*.

8 N.A.S.A., *Saving Earth from Asteroids. jhuapl*, https://www.nasa.gov/feature/saving-earth-from-asteroids.

9 A. Rabeendran and L. Denneau, *A Two-Stage Deep Learning Detection Classifier for the ATLAS Asteroid Survey*, https://doi.org/10.1088/1538-3873/abc900, arXiv.

10 V. Bahel, P. Bhongade, J. Sharma, S. Shukla and M. Gaikwad, International Conference on Computational Intelligence and Computing Applications (ICCICA, pp. 1–4,.

11 J. Friedman, *Annals of Statistics*, 1189–1232.

12 NeoWs, *NASA: Asteroids Classification. Kaggle*, https://www.kaggle.com/datasets/shrutimehta/nasa-asteroids-classification.

13 S. Morgenthaler, *WIREs Comp Stat*, **1**, 33–44.

14 R. Pramoditha, *Why do we set a random state in machine learning models?. tds*, https://towardsdatascience.com/why-do-we-set-a-random-state-in-machine-learning-models-bb2dc

15 D. Berrar, *Cross-Validation*, https://www.researchgate.net/profile/Daniel-Berrar/publication/324701535_Cross-Validation/links/5cb4209c92851c8d22ec4349/Cross-Validation.pdf.

16 M. LaValley, *Circulation*, **117**, year.

17 S. Suthaharan, *Support vector machine. Machine learning models and algorithms for big data classification*, https://link.springer.com/chapter/10.1007/978-1-4899-7641-3_9.

18 S. Learn, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

19 J. Wong, T. Manderson, M. Abrahamowicz, D. Buckeridge and R. Tamblyn, *Can Hyperparameter Tuning Improve the Performance of a Super Learner? Epidemiology*, https://journals.lww.com/epidem/Fulltext/2019/07000/Can_Hyperparameter_Tuning_Improve_the_Performance.9.aspx.

20 V. Carruba, S. Aljbaae, R. Domingos and W. Barletta, *Artificial Neural Network classification of asteroids in the M1:2 mean-motion resonance with Mars*, https://doi.org/10.1093/mnras/stab914, arXiv.

21 S. Djorgovski, A. Mahabal, C. Donalek, M. Graham, A. Drake, M. Turmon and T. Fuchs, https://doi.org/10.1109/eScience.2014.7, Automated Real-Time Classification and Decision Making in Massive Data Streams from Synoptic Sky Surveys. arXiv.