Under Attack - Binary Classification of Hazardous Objects from Space

Katie Knight*
Anna-Maria Nau*
Christoph Metzner*
kknigh22@vols.utk.edu
anau@vols.utk.edu
cmetzner@vols.utk.edu
DSE 511 - Project 3
University of Tennessee, Knoxville
Knoxville, Tennessee, USA

Figure 1: Potential threat of extinction through extraterrestrial objects. (https://eparisextra.com/living/nasa-attempts-to-stop-hypothetical-asteroid-from-hitting-earth-and-fails/)

ABSTRACT

This study utilizes machine learning algorithms to identify potentially hazardous objects, such as asteroids, and therefore serve as an early warning system. To perform this binary classification task, that is, to predict whether an asteroid is hazardous or not, four supervised classifiers, including naïve Bayes, support vector machine, decision tree, and random forest were trained on data provided by the NASA API called NeoWS (Near Earth Object Web Service) and which is readily available on Kaggle (https://www.kaggle.com/shrutimehta/nasa-asteroids-classification). Specifically, each algorithm was trained and tuned on two training sets (i.e., one min-max normalized, and one reduced via principal component analysis), and then evaluated on a testing set (20%). In addition, to identify the most important predictors, feature importance was applied. In total, eight classifiers were built and tested, and the highest performance was achieved by the decision tree with a F1-score of 0.979, a recall of 0.9655, and a precision of 0.9929.

KEYWORDS

machine learning, supervised, classification, pca, feature importance

*All three authors contributed equally to this research.

Knoxville '21, Fall, 2021, Knoxville, TN

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

ACM Reference Format:
Katie Knight, Anna-Maria Nau, and Christoph Metzner. 2021. Under Attack - Binary Classification of Hazardous Objects from Space. In *Knoxville '21: DSE511 - Introduction to Data Science and Computing I, Fall, 2021, Knoxville, TN*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

In "A Synopsis of the Astronomy of Comets", Edmond Halley first voiced his concerns about the devastating impact of extraterrestrial objects, such as asteroids or comets, on the modern world in the 18^{th} century [16]. A group centered around Luis Alvarez hypothesized that such objects caused several mass extinctions and, generally, has a significant influence on the well-being of our planet [1]. In the face of potential extinction, US Congress ordered NASA, in collaboration with The University of Arizona and Hawaii, to develop strategies for observation, tracking, characterization, and damage mitigation [13] for such objects. One specific strategy is the development of statistical models utilizing machine learning that are able to classify potentially hazardous objects based on their properties, such as shape, size, velocity, or shape of orbit. These extraterrestrial objects, asteroids or comets, are referred as near-earth objects (NEOs). NEOs orbit the sun, may potentially cross Earth's orbit [2], and can range in size from a dust particle to kilometers in diameter [11]. NEOs are categorized into near-Earth asteroids (NEAs) and near-Earth comets (NECs) based on their origin in space and geometric appearance. NEAs originate from the Main Belt [3], while NECs come from either the Kuiper belt or the Oort cloud [8].

There is a wide range of publications that propose different strategies for impact risk assessment of NEOs based on common features such as size or velocity. For example, a roughly twenty-year-old approach is to use scales rating the potential impact of NEOs such as the *Torino scale* or *Palermo scale* [14]. Their advantage is that the layman can easily interpret such scales. Another approach focuses on using VNIR spectroscopy to characterize the compositional and physical structure of NEOs to determine their hazardous potential [7].
Nugent et al. [15] were the first to apply machine learning algorithms to the classification of hazardous NEOs. A more recent study successfully applied more complex machine learning algorithms based on neural network frameworks to infer the physical properties and impact risks of asteroids by analyzing energy deposition curves [17]. We hypothesize that using machine learning algorithms can play a vital part in the detection of hazardous objects. Specifically, we ask if standard "off-the-shelf" machine learning algorithms can be trained to help identify whether a NEOs is hazardous or not.

2 MATERIALS AND METHODS

This work applied four supervised classification algorithms to a NEOs dataset to predict whether a NEO is hazardous or not. Specifically, the classification performances of 1) naïve Bayes as the baseline, 2) support vector machine, 3) decision tree, and 4) random forest were compared. Furthermore, the impact of reducing the number of features via principal component analysis on the classification performance was analyzed. The hyperparameters of all algorithms were optimized using k-fold cross validation (k=5) on the training set via the function *GridSearchCV* from the machine learning library *sklearn*.
The performance of the developed models were evaluated on the testing set using the performance metrics *F1-Score*, *Precision*, and *Recall*. Lastly, to identify the top predictors, feature importance was applied and discussed.

2.1 Data source

The source data for this study was provided by the NASA API called NeoWS (Near Earth Object Web Service available at https://api.nasa.gov), and is readily available at https://www.kaggle.com/shrutimehta/nasa-asteroids-classification. The dataset contains information on 4687 asteroids (rows), and 40 features (columns), one being the target feature indicating the ground truth (hazardous or non-hazardous). Out of the 4687 samples, 3932 are labeled as non-hazardous, and 755 as hazardous. In addition, the data contains no missing values and all features but one are numeric with varying scales. The dataset includes typically measured features of an asteroid, such as the *absolute magnitude*, *estimated diameter*, *relative velocity*, distance measures of the object to the sun such as *perihelion distance* (minimum distance) and *aphelion distance* (maximum distance), or the shape of the objects orbit represented by *eccentricity*.

2.2 Data preprocessing

After analyzing the input features, various redundant, inexpressive, or time-related variables were identified and removed. For example, features describing the *estimated diameter* of the asteroids in different scales (e.g., feet, meter, and miles) were dropped due to redundancy. The features *orbiting body* and *equinox* were dropped since they contained only one unique value. Time-related features such as *close approach date* or *epoch date close approach* were also removed since knowing the date does not contribute to the fact whether an asteroid will be hazardous or not. After feature removal, the dataset contained 20 input features and one binary target variable. The data was randomly split into an 80:20 ratio as train and test set, respectively. The train set contains 3749 samples (3139 non-hazardous and 610 hazardous), and the test set 938 samples (793 non-hazardous and 145 hazardous).

2.2.1 Min-Max Normalization. Since the data contained different scales of measurement, the input features were normalized to ensure equal contribution during model fitting. Specifically, min-max normalization was applied which transformed all input features into the range [0, 1], meaning that the minimum and maximum value of each feature is 0 and 1, respectively.

2.2.2 Principal Component Analysis (PCA). Principal Component Analysis (PCA) is an unsupervised dimensionality reduction technique. It is a way to reduce the number of features while maintaining the majority of the important information. It transforms a number of variables that may be correlated into a smaller number of uncorrelated features, known as principal components (PCs). The PCs are linear combinations of the original variables weighted by their variances (or eigenvalues) in a particular orthogonal dimension such that the first PC accounts for the largest variance in the data, the second PC accounts for the second largest variance

in the data, and so on. To evaluate the usefulness of the PCs and to determine the number of components to use for model training, the total explained variance ratio metric was used which is the sum of the percentage of variance that is attributed by each of the selected components. The total explained variance ratio chosen for this study was 0.95, which resulted in nine PCs as shown by 2.

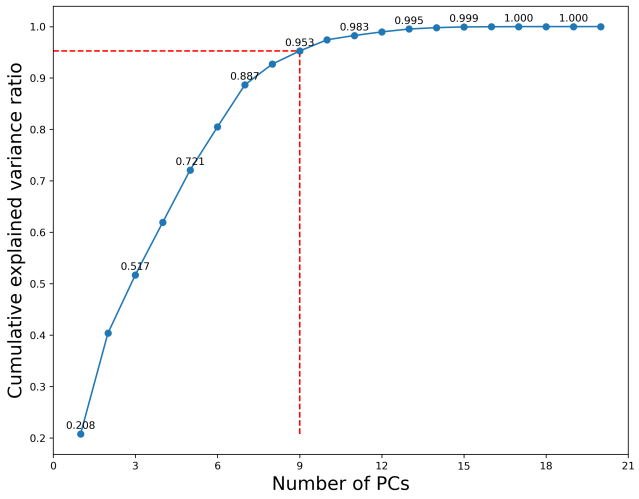

Figure 2: Line plot showing the cumulative explained variance ratio for varying number of PCs. The plot shows that nine PCs are able to achieve a total explained variance of 0.95.

2.3 Supervised Classification Algorithms

2.3.1 Gaussian Naïve Bayes. The baseline classifier for this binary-classification problem was set to be the Gaussian Naïve Bayes algorithm. This algorithm belongs to the class of generative models, since it tries to learn the *class-conditional* density $p(x|y)$ and the *class priors* $p(y)$ for each value of y (i.e., each category). This allows to use the Bayes rule to compute the posterior probability $p(y|x)$.

$$p(y|x) = \frac{p(x|y) \times p(y)}{\sum_{y'=1}^{C} p(x|y')p(y')} \tag{1}$$

In naïve Bayes it is assumed that all features are conditionally independent given the class label and in our case to follow a Gaussian distribution. This assumption is usually false but simplifies the computation tremendously [12].

2.3.2 Support Vector Machine. This algorithm belongs to the class of discriminant models that separate the samples based on a boundary (i.e., hyperplane). Vladimir Vapnik [18] developed support vector machine (SVM) applying the idea of structural risk minimization. Specifically, SVM tries to identify a decision boundary that maximizes the margin, distance from the closest positive to the closest negative data sample [5]. The main advantage of SVM is its ability to generalize and properly classify inseparable categories by projecting the data in lower space into higher dimensions using the *kernel trick* [18].

2.3.3 Decision Tree. The decision tree algorithm [10] is a supervised learning algorithm, most useful for either solving problems related to regression or classification. This algorithm creates a training model that predicts a variable class via learning decision rules inferred from the training data. For each class label, the decision tree begins at the "root" attribute, with each value of the root attribute compared to the target variable. This comparison dictates what "branch" corresponds to the attribute value, and creates a set of "if-then-else" decision rules.

2.3.4 Random Forest. The random forest classifier consists of multiple decision trees. Each tree in the algorithm produces a prediction, with the most common prediction among each tree becoming the overall model's prediction [4]. Since individual decision trees tend to overfit and exhibit high variance, the advantage of randomness is the production of decision trees with decoupled prediction errors; averaging the prediction can cancel out some of these errors.

2.4 Performance Evaluation

Performance evaluation is addressed using the harmonic mean of the precision and recall (also known as the *true positive rate*), the so-called F1-score performance metric. The F1-score is defined as follows:

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \tag{2}$$

$$precision = \frac{tp}{tp + fp} \tag{3}$$

$$recall = \frac{tp}{tp + fn} \tag{4}$$

where tp being the true positive events, fp the false positive events, tn true negative events, and fn the false negative events. Also, to further evaluate the accuracy of specific models, the receiver operating characteristic curve (ROC) plots are used. These plots provide an accuracy measure by plotting the recall (y-axis) vs. 1 - specificity (x-axis). The closer the curve is to the upper left corner, the better the discriminatory ability of the model [6]. The outcomes of the confusion matrix for the test are also reported in the results section.

2.5 Feature Importance

Feature importance analysis lets us look at what features were most useful for each algorithm. Since Gaussian Naive Bayes and SVM Classifier using RBF-Kernel do not have an intrinsic way of determining feature importance, we used SciKit's permutation importance as a means to inspect which features were significant for classifying NEOs as hazardous or not. Permutation feature importance is when a model's score decreases after one feature value is randomly shuffled 1. Because this severs the feature and target relationship the score drop indicates how much the model relies on the feature.

3 RESULTS

Each model was trained on the normalized and normalized+pca train data, and evaluated on the normalized and normalized+pca test data. Given this, a total of eight models (2 of each algorithm)

were trained and evaluated.
The performance results are presented in Table 1. All models only trained on the normalized data significantly outperformed their respective counterpart trained on normalized+pca data. Decision tree, random forest, and support vector machine outperformed our baseline, the naïve bayes classifier. The differences in F1-Score range from 30 for support vector machine algorithm up to 50 points for the decision tree. The highest F1-Score with 0.979, recall of 0.9655, and precision of 0.9929 were achieved by the decision tree algorithm trained on normalized data followed closely by the random forest model. The fastest algorithm among all models is naïve bayes due to its simplicity, closely followed by the decision tree. Table 2) shows the predicted conditions for each sample of all models. As already indicated, the decision tree performs slightly better than the random forest algorithm on the min-max scaled data.

Table 1: Performance evaluation of the trained models for the F1-score, recall, and precision. The algorithms naïve bayes (NB), support vector machine (SVM), decision tree (DT), and random forest (RF) were trained on *min-max-scaled* data and/or on features created by principal component analysis (PCA). The average computing time per model training during hyperparameter tuning is reported.

Algorithm	Data	F1-Score	Recall	Precision	Time [s]
NB	Scaled	0.8464	0.8552	0.8378	0.0160
NB	PCA	0.515	0.4138	0.6818	**0.0104**
SVM	Scaled	0.8865	0.8621	0.9124	2.544
SVM	PCA	0.5959	0.5034	0.73	5.376
DT	Scaled	**0.979**	**0.9655**	**0.9929**	0.02973
DT	PCA	0.4758	0.4069	0.5728	0.03461
RF	Scaled	0.9718	0.9517	0.9928	3.7710
RF	PCA	0.5641	0.4552	0.7416	5.4340

Table 2: Confusion matrix of the trained models. The algorithms naïve bayes (NB), support vector machine (SVM), decision tree (DT), and random forest (RF) were trained on *min-max-scaled* data and/or on features created by principal component analysis (PCA).

Algorithm	Data	TP	FP	FN	TN
NB	Scaled	769	24	21	124
NB	PCA	765	28	85	60
SVM	Scaled	781	12	20	125
SVM	PCA	766	27	72	73
DT	Scaled	792	1	5	140
DT	PCA	749	44	86	59
RF	Scaled	792	1	7	138
RF	PCA	770	23	79	66

The individual model performances are further analyzed using the receiver operating characteristic curve (ROC-curve) that visualizes the diagnostic ability of a binary classifier with varying threshold boundaries (Figure 3). The model performance heavily depends on which method was used for preprocessing the raw data. In general, the min-max scaled data outperformed the models fed with data preprocessed with principal component analysis. The best model performance is arguably the random forest classifier with only min-max scaled data, shown as the yellowish line. Furthermore, our baseline model, the naïve bayes classifier, is slightly better than the decision tree.

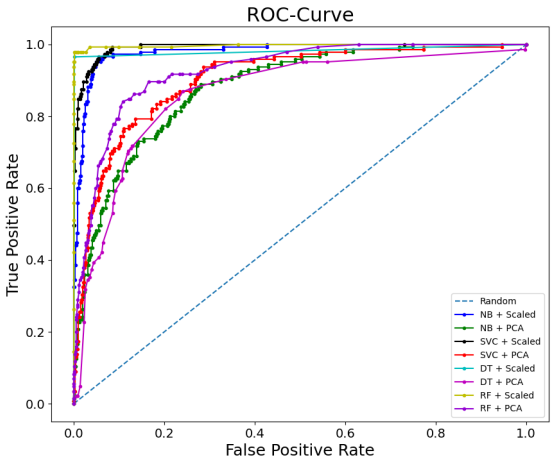

Figure 3: Receiver operating characteristic curve for the eight different models. All algorithms performed better on only min-max scaled data than their respective counterpart using the components explaining at least 95% of the variance contained by the data of the principal component analysis.

3.1 Feature Importance results

Figures 3, 4, 5, and 6 show the resulting significant features for each classifier.

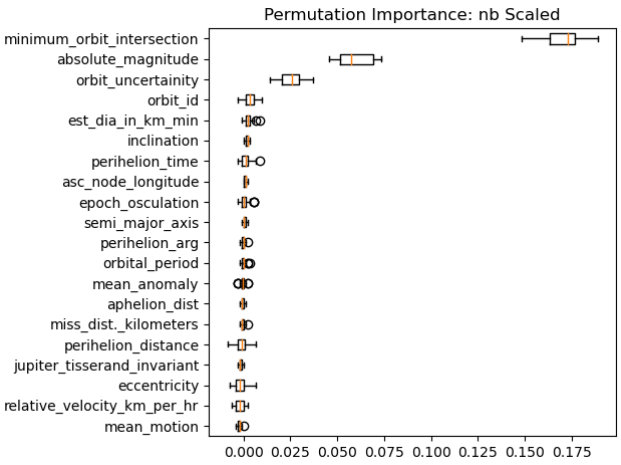

Figure 4: Naive Bayes feature importance.

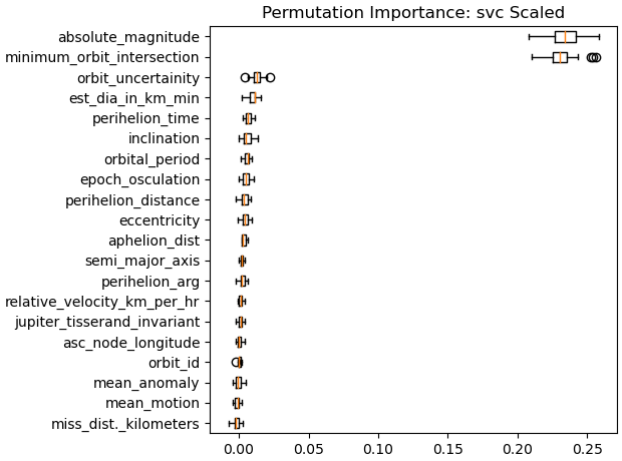

Figure 5: SVC feature importance.

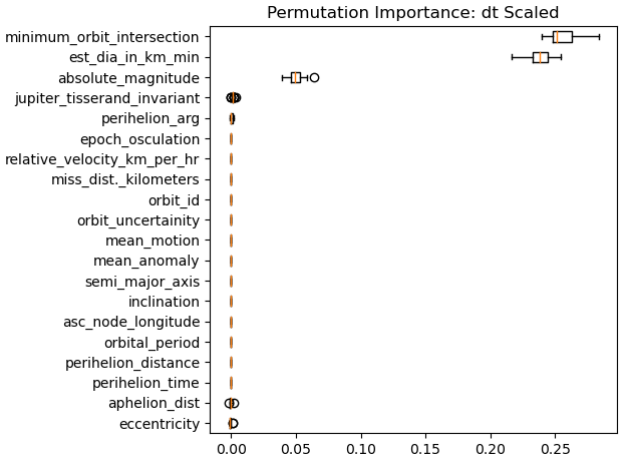

Figure 6: Decision tree feature importance.

4 DISCUSSION

Decision tree and random forest algorithm showed similar performances due to the nature of the random forest algorithm and the already stable, high-performing results of the individual decision tree. Given the already well performing decision tree it is unlikely that an ensemble of decision trees will perform significantly better. It is more likely, that the random forest performs slightly worse due to the inherent variation of the algorithm. Models trained only on min-max scaled data performed significantly better than the models trained on PCA-transformed data. This could stem from the fact that PCA does not consider the relationship between the independent variables with the dependent variable or target. In fact, PCA will treat components with the highest variances as its features but said features may be insignificant for the target variable. As a result, PCA could create many meaningless features and purge actually useful ones from the feature space.

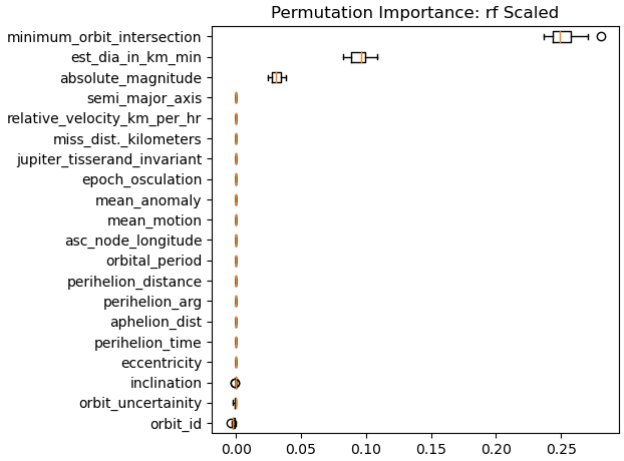

Figure 7: Random forest feature importance.

Feature importance analysis showed that 3 of our 4 algorithms favored minimum orbit intersection as the most important feature. Minimum orbit intersection distance measures any possible near approaches and collision risks between astronomical objects [9] and is considered an important calculation to use when determining if any object is at risk for collision with the Earth. Absolute magnitude also featured prominently with one algorithm (SVC), and appeared among the top three for the others. NASA defines this as "the visual magnitude an observer would record if the asteroid were placed 1 Astronomical Unit (au) away", and assists in measuring the diameter of any NEO. Interestingly, two algorithms (DT and RF) also favored the estimated diameter in kilometers as important; though it may be that this feature is essentially the same as absolute magnitude. Also of note is that only two classifiers (SVC and NB) favored orbit uncertainty as significant.

Permutation scores do indicate the relative predictive power of a feature for a model, but the scores are only useful in this context, and features with high scores should only be considered in that context. Additionally, feature importance is not the same as statistical inference, and give us no information about the nature of the relationship of that feature (e.g., linear, etc.).

When we first began this experiment, we aimed to also include the distributed gradient boosting library, XGBoost[1], among the classifiers to test. However, after running into significant difficulty with getting this library to work properly (namely, memory seemed to be a significant issue), we chose to test the Random Forest classifier instead. Still, future research may explore XGBoost as another possibility for this type of classification.

5 CONCLUSION

In this study, we demonstrated that "off-the-shelf" machine learning algorithms can be used to identify the hazardous nature of extraterrestrial objects. Based on the performance metrics F1-score, precision, and recall, the decision tree slightly outperforms the random forest algorithm. However, in cases that requires to have

[1]https://xgboost.readthedocs.io/en/stable/

a flexible threshold for binary classification one may want to use the random forest. The use of principal component analysis to accomplish feature reduction reduced the performance of all algorithms, Naïve Bayes, Support Vector Machine, Decision Tree, and Random Forest significantly, and is therefore, not recommended as a preprocessing step.

REFERENCES

[1] Luis W Alvarez, Walter Alvarez, Frank Asaro, and Helen V Michel. 1980. Extraterrestrial cause for the Cretaceous-Tertiary extinction. *Science* 208, 4448 (1980), 1095–1108.

[2] Space Studies Board, National Research Council, et al. 2010. *Defending planet earth: Near-Earth-Object surveys and hazard mitigation strategies*. National Academies Press.

[3] William F Bottke Jr, Alessandro Morbidelli, Robert Jedicke, Jean-Marc Petit, Harold F Levison, Patrick Michel, and Travis S Metcalfe. 2002. Debiased orbital and absolute magnitude distribution of the near-Earth objects. *Icarus* 156, 2 (2002), 399–433.

[4] L. Breiman. 2001. Random Forests. *Machine Learning* 45, 1 (2001), 5–32.

[5] Ronan Collobert and Samy Bengio. 2004. Links between perceptrons, MLPs and SVMs. In *Proceedings of the twenty-first international conference on Machine learning*. ACM, 23.

[6] Jerome Fan, Suneel Upadhye, and Andrew Worster. 2006. Understanding receiver operating characteristic (ROC) curves. *Canadian Journal of Emergency Medicine* 8, 1 (2006), 19–20.

[7] Michael J Gaffey, Paul A Abell, and Paul S Hardersen. 2006. COMPOSITIONAL AND PHYSICAL CHARACTERIZATIONS OF NEOs FROM VNIR SPECTROSCOPY. (2006).

[8] Sarah Greenstreet, Henry Ngo, and Brett Gladman. 2012. The orbital distribution of near-Earth objects inside Earth's orbit. *Icarus* 217, 1 (2012), 355–366.

[9] José M Hedo, Manuel Ruíz, and Jesús Peláez. 2018. On the minimum orbital intersection distance computation: a new effective method. *Monthly Notices of the Royal Astronomical Society* 479, 3 (2018), 3288–3299.

[10] R. Olshen L. Breiman, J. Friedman and C. Stone. 1984. *Classification and Regression Trees*. Wadsworth.

[11] Amy Mainzer, T Grav, J Bauer, J Masiero, RS McMillan, RM Cutri, R Walker, E Wright, P Eisenhardt, DJ Tholen, et al. 2011. NEOWISE observations of near-Earth objects: preliminary results. *The Astrophysical Journal* 743, 2 (2011), 156.

[12] Kevin P Murphy et al. 2006. Naive bayes classifiers. *University of British Columbia* 18, 60 (2006), 1–8.

[13] NASA. [n. d.]. Near-Earth Object Observations Program. https://www.nasa.gov/planetarydefense/neoo.

[14] NASA. [n. d.]. Torino Impact Scale. https://web.archive.org/web/20070224184143/http://impact.arc.nasa.gov/torino.cfm.

[15] Carrie R Nugent, John Dailey, Roc M Cutri, Frank J Masci, and Amy K Mainzer. 2017. Machine learning and next-generation asteroid surveys. In *AAS/Division for Planetary Sciences Meeting Abstracts# 49*, Vol. 49.

[16] Alfred Romer. 1984. Halley's comet. *The Physics Teacher* 22, 8 (1984), 488–493.

[17] A. M. Tarano, J. Gee, L. Wheeler, S. Close, and D. Mathias. 2020. Automating the Inference of Asteroid Physical Properties and Motion. In *AGU Fall Meeting Abstracts*, Vol. 2020. Article P008-02, P008-02 pages.

[18] Vladimir Vapnik. 2013. *The nature of statistical learning theory*. Springer science & business media.