Close Encounters: A Study of Small Body Trajectories in Our Solar System

By Andre Kleber, Trevor McCalmont, & Julien Weinstein

Introduction

Asteroids are metallic or rocky bodies without atmospheres that are in direct orbit with the sun and whose sizes and shapes vary widely. They originated in the early solar system from violent collisions of proto-planets, generating clumps of small particles that gradually grew in size and accreted more mass from surrounding particles via gravity. So far, more than 1 million asteroids have been identified in our solar system alone, and many more remain undiscovered[1]. Asteroids carry a special significance for humanity as they were instrumental in the formation of our planet and the evolution of life on Earth.

Today, asteroids remain a central focus of scientific study and popular culture. Asteroids are as rich in data about our solar system's origins as they are replete with rare minerals that have numerous applications for humanity. There is even some evidence to suggest that asteroids contain organic compounds like amino acids, which scientists hypothesize could have seeded Earth with life itself[2]. Additionally, given the exhaustibility of natural resources on our planet, asteroid mining may become a profitable and necessary endeavor to extract raw materials that are key for producing goods like crop fertilizer, and sourcing rare metals like nickel, cobalt, gold, and palladium. Such resources could power entire industries and life-sustaining activities that currently depend on Earth's limited materials.

A final relevance to studying asteroids is the one that is probably most commonplace in the popular imagination–that of an extinction-level collision of an asteroid with Earth. While there is currently no known asteroid forecasted to deliver us such a doomsday scenario, it still remains a perennial possibility that presents an existential threat to humanity. Accordingly, asteroids present a unique opportunity for us to both peer into the past and look ahead to the future, and to ultimately better understand how we can leverage their creative and destructive powers. To that end, this project uses modern machine learning methods to develop models that can predict the trajectories of asteroids passing Earth, and will hopefully enable us to be better prepared for our next encounter with one.

Data Source

The Jet Propulsion Laboratory out of the California Institute of Technology maintains a Small Body Database (SBDB) containing the orbit and position for all known small bodies in outer space. A Small Body is a natural astronomical object that is not a planet or satellite. This usually means all asteroids and comets, but can also include dwarf planets. For our purposes, these

[1] Choi, C. Q., & Harvey, A. (2021, November 22)
[2] Reuell, P. (2019, July 9)

bodies are of interest because of a potential collision between a small body and planet Earth. By tracking their orbits and positions, we can study past trajectories and predict future paths of near-Earth objects through the solar system, and assess the risk of our paths crossing. Therefore, we will leverage the SBDB to acquire a large dataset of asteroid trajectories and use its features to generate models that predict the likelihood of a close-approach with Earth.

The SBDB API allows users to submit queries that return asteroid datasets according to the chosen parameters, which include date ranges for the query ranging from as early as January 1, 1900 up through January 1, 2100, and a maximum distance specification which includes only asteroids that have come within that distance of Earth. Based on the query, the API returns important features of each asteroid such as its diameter (km), velocity (m/s), absolute magnitude (which is an astronomical term for the luminosity of the body), distance from the earth at closest approach (km), the year, day, and time of this approach, and the 3-sigma uncertainty in some of these values.

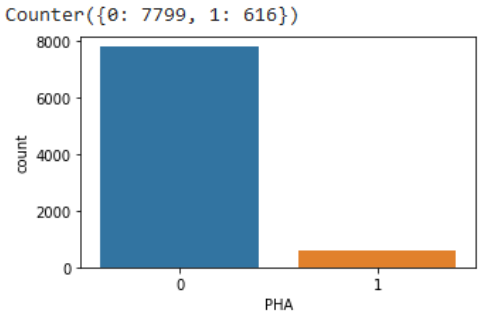
Data Preparation

For this study, we conducted a query for all asteroids that have or will approach within 0.2 astronomical units (au), or about 30 million kilometers, from Earth. For reference, the moon is about 385,000 kilometers from Earth. The query includes all aforementioned parameters, as well as the min/max and uncertainty in each numerical variable. For example, the distance metric includes a min/max estimate of 3 standard deviations in each direction, and the velocity metric includes a relative and infinite velocity, which is the velocity relative to a massless body. There is also a column for the uncertainty in diameter and in time of closest approach.Taken together, there are nine informative features for each asteroid. The query was executed in a Google CoLab notebook using the Python *requests* library. There were 147,577 records returned by the query.

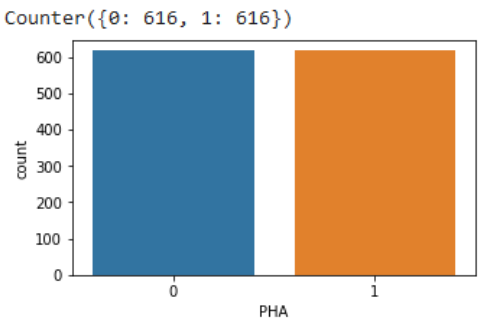
Once converted from json to Pandas DataFrame format, we had to perform some basic operations to clean and manipulate the data into a usable format for our models. This entailed dropping irrelevant columns (e.g. Orbit ID), removing rows with missing values, and reformatting columns into appropriate data types. For example, many of the numerical columns were in Python Object format, so we converted those columns to floats. Finally, the basis for our models is a special designation for asteroids that come within 0.05 au, or about 7 million kilometers, of Earth and have an absolute magnitude of less than or equal to 22. Asteroids that come this close to our planet are called Potentially Hazardous Asteroids (PHAs). We created a new, binary column based on these cutoff values that will serve as class labels for our supervised and unsupervised learning models, with the label 0 signifying the asteroids that did not meet this criteria, and 1 representing the PHAs.

One issue the dataset presents is that of class imbalance. Most asteroids do not qualify as PHAs, so in looking at the class distribution of our data below, the majority class is by far *not* a PHA.

Original Distribution



Under-Sampled Distribution



In order to deal with this class imbalance, we employed the *imbalanced-learn* Python library to undersample the majority class and more evenly distribute the labeled data. To test our baseline, we generated a dummy classifier with the *most-frequent* class specification, and found that resampling brought down the dummy classifier score from 92% to an expected value of 50%. Next, due to the variation in units of our dataset's features, we used scikit-learn's *StandardScaler* preprocessing method to scale the data to unit variance for use in our models. A last step we took was to split the balanced, scaled data to an 80/20 train-test-split. With our data now fully preprocessed, we moved into the model building phase.

Part A. Supervised Learning

Motivation

We want to use features of small bodies to predict whether they will be a Potentially Hazardous Asteroid (PHA), defined as coming within 0.05 AU of Earth, or about 4.5 million miles, and having an absolute magnitude of less than 22 units. Absolute magnitude measures the luminosity of celestial objects. An object's absolute magnitude is equal to the apparent magnitude that the object would have if it were viewed from a distance of exactly 10 parsecs (32.6 light-years). By hypothetically placing all objects at a standard reference distance from the observer, their luminosities can be directly compared among each other on a magnitude scale.

Methods & Evaluation

Methods

Since our problem is a classification task, we tried out several classifier models like RandomForest, KNeighbors (KNN), SupportVectorClassifier (SVC) and Logistic Regression. We also explored two deep learning models generated by PyTorch and TensorFlow.

We used the default parameters for training the standard classifiers (RandomForest, KNN, SVC and Logistic Regression), including the random_state parameter for reproducibility.

For the PyTorch deep learning model, we built a class to render our data in batch format. The PyTorch Dataloader utility enabled us to create mini batches of size 32 which we then used to

train our model. The architecture of our PyTorch model comprised three linear hidden layers, a ReLu activation function applied to the output of those hidden layers, and a final single-node output layer. As a loss function we chose the BCEWithLogitsLoss function which automatically applies a sigmoid activation and circumvents the need for explicitly creating a final sigmoid activation layer. For the optimizer we chose the Adam algorithm, which provides an accelerated gradient descent process by using exponentially weighted averages.

In addition to the PyTorch model, we set up a TensorFlow model for comparison reasons with the same architecture, loss function and optimizer.

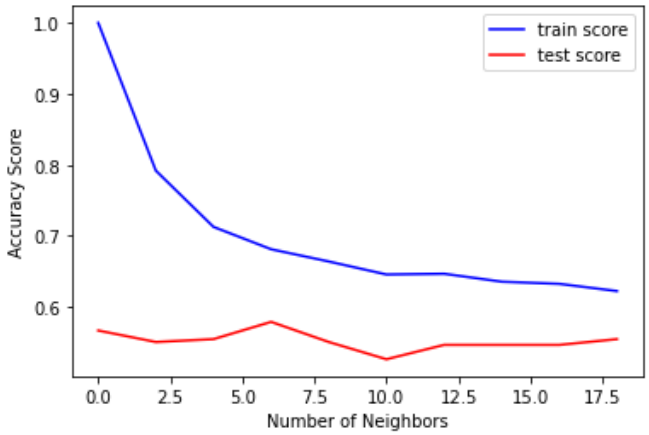
Evaluation

Table 1: Evaluation metric of supervised models (Macro Averages)

Classifier	**Accuracy**	**Recall**	**Precision**
RandomForest	0.68	0.69	0.69
KNeighbors	0.56	0.56	0.56
SupportVectorClassifier	0.58	0.59	0.59
LogisticRegression	0.49	0.49	0.49
PyTorch Model	0.61	0.61	0.61
TensorFlow Model	0.60	0.61	0.58
DummyClassifier	0.48	0.24	0.50

Some of our supervised models returned strong performance metrics. Random Forest led the way achieving an accuracy of 68% with recall and precision at 69%. The PyTorch deep learning methodology scored second best with accuracy, precision, and recall at 61%. The Random Forest Classifier model scoring as high as it did seems strong to us. Our intuition was that the variables in this dataset would not have much predictive power. It is a pleasant surprise that we are able to predict potentially hazardous asteroids with this much success.

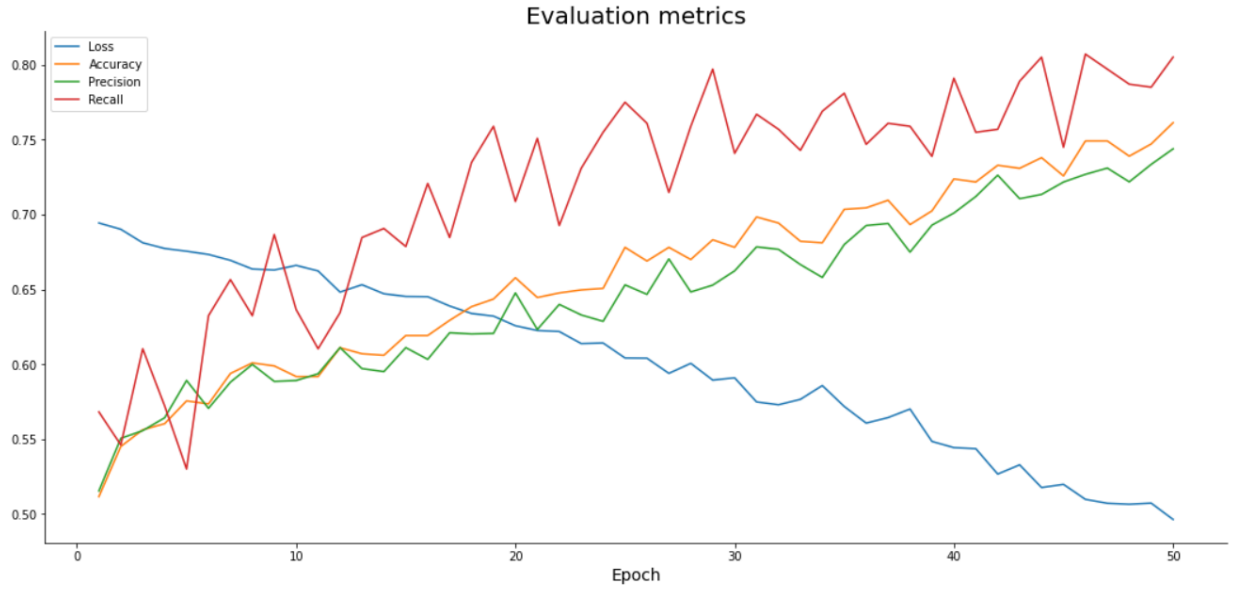
Going into each model more specifically, we can first look at the K Nearest Neighbors model. While little hyperparameter tuning was necessary to achieve these results, we found that train scores for the K Nearest Neighbor classifier resemble an exponential rate of decline as k increases. The test scores were fairly consistent regardless of how many neighbors used.



Our RandomForest classifier is made up of a series of decision trees that predict a target label by majority vote. RandomForest selects a random sample from the training set, creates a decision tree for it and gets a prediction. This operation is repeated for the assigned number of trees, and for classification the prediction is decided by majority vote. For our data, RandomForest's default parameters worked well for generating accurate predictions. These parameters included an *n_estimators* of 100 and a *Gini* split quality function.

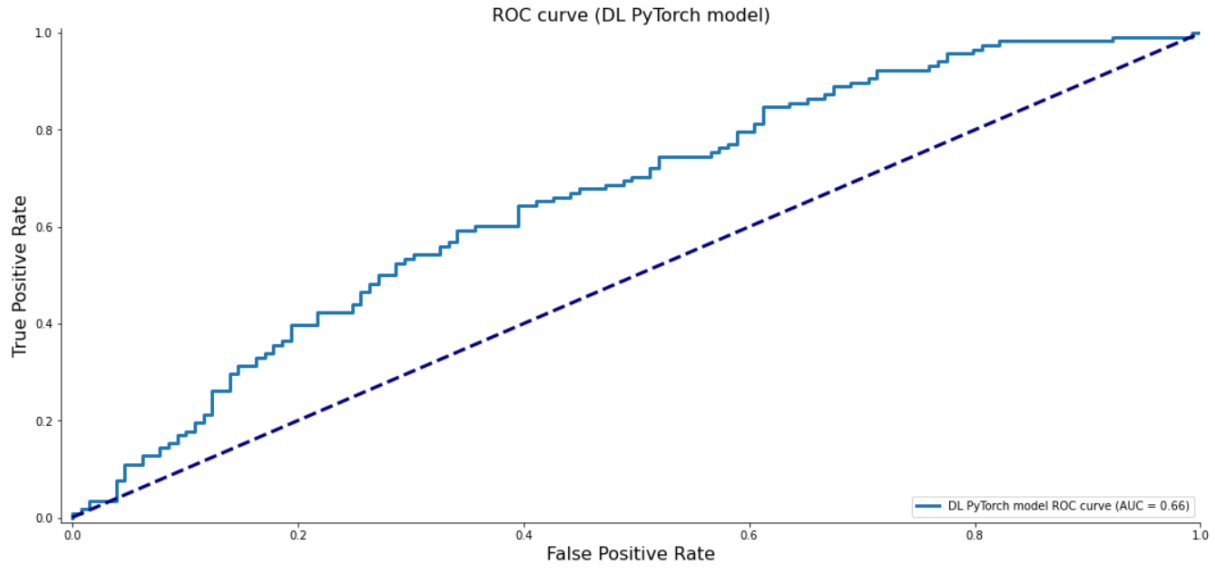
Lastly, our Deep Learning models returned high accuracy scores as well. As shown below, the TensorFlow model's loss decreased significantly over 50 epochs, while the accuracy, precision, and recall improved over time.

TensorFlow Evaluation Metrics (Training Phase)



The PyTorch model also performed well, achieving an area under the ROC curve of 0.66. Since the accuracy score alone does not account for true and false positives (TP and FP), the area

under the curve (AUC) allows us to assess the TP vs. FP rate as an aggregate measure of performance across all possible classification thresholds.



Failure analysis

Initially, we used over-sampling to enlarge our sample of PHAs, and some models scored very strong results with accuracies over 95%. However, upon review, we believe that for certain model types, the over-sampling was actually contaminating our data set and causing some data leakage. To achieve balanced classes of PHAs and Non-PHAs, we needed to oversample our PHAs by roughly 13 times larger than the number of PHAs in the data set. Our hypothesis is that some models would see these data points in the training data and then re-create them exactly in the testing data. Model types that are based on regression would not benefit from this, but particularly the Random Forest, Deep Learning, and K Nearest Neighbors models all demonstrated some concerningly high accuracy scores. After switching to under-sampling, our accuracy scores dropped to a more believable range, and we think this would generalize to unseen data with greater accuracy.

With respect to the failures of specific models, the Support Vector Classifier was not very successful in returning high accuracy scores. The default Radial Basis Function kernel (a non-linear classifier) was the most robust, but still fell far below that of some of our other models. Two reasons for this are that SVCs do not function well on large datasets, or ones with a high amount of noise. Both of these may be true for our dataset, and could explain the weakness of the SVC for our data. Another classifier that performed poorly was the logistic regression model. This can be explained by the fact that the target PHA label may not be linearly correlated with the features of our dataset, so the logistic regression cannot predict targets with good accuracy, even on the training data.

Another model type that did not fit our dataset well was the polynomial regression model. Even in testing various degrees for the polynomial regression, we were not able to achieve high scores on the model. We tested out a polynomial regression model with degree 1 through 5, and found that there was a slight increase in the test accuracy at degree 3, but beyond that, accuracy began to decrease dramatically. This dropoff is attributable to the nature of increasing the degree of a polynomial. The larger the degree, the more the model will approximate the sine curve, which does not correspond to our data.

Part B. Unsupervised Learning

Motivation

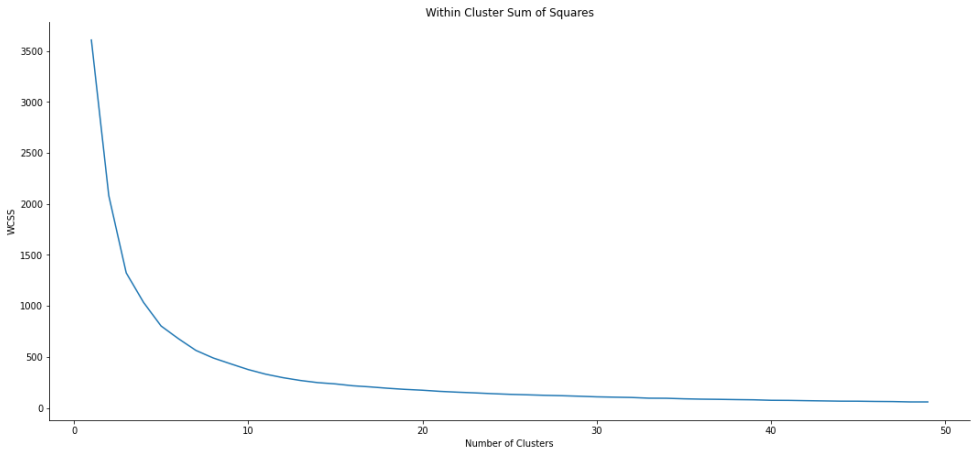
Unsupervised learning methods discover structure in unlabeled data. We wanted to examine if there were any clustering-like groupings within the asteroids based on their attributes or whether there are outliers and anomalies that could lead us to some meaningful insights.

Unsupervised Learning Methods

The first unsupervised learning method we will investigate is K-Means Clustering. K-Means Clustering groups the data into k clusters while minimizing the sum of squared distance between each data point and its respective cluster center. If this method is going to be successful, when we plot our data and clusters, we should notice k distinct clusters with somewhat well-defined edges between them.

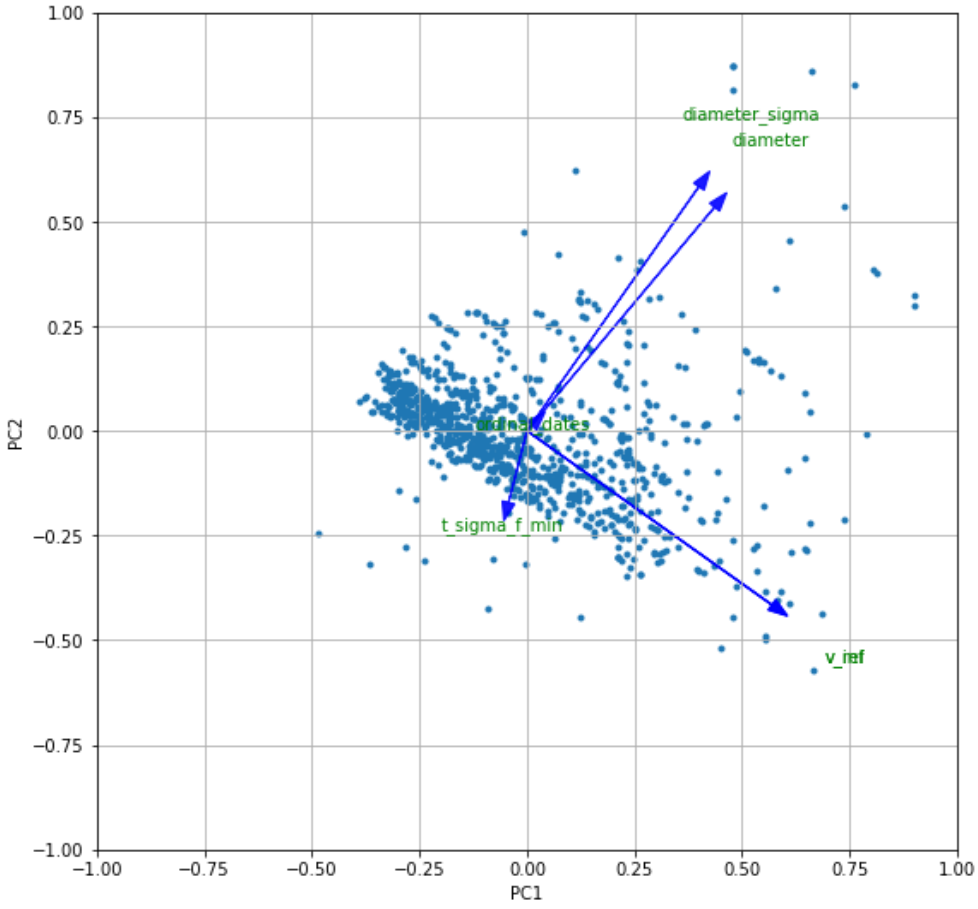
Because distance is a factor in K-Means model performance, it is important that the data is scaled as outlined in the Data Preparation section above. Next, we will analyze various values of k to learn the range of values that will optimize our analysis. We are looking to minimize the sum of squared distance while not overcomplicating the problem and still extracting real world meaning. Imagine looking at clusters of 50 data points and trying to gather meaningful insights.

Looking at the Within Cluster Sum of Squares for k = 1 to 50, we can see that the sum of squares distance decreases rapidly from k = 1 to 8 and then gradually tapers off from there.



We will use a value of k = 8 moving forward, but there are multiple values of k that are justifiable choices depending on the context and domain. We now apply K-Means clustering to our dataset. However, our data is 6-dimensional. There are two issues here. First, how will we visualize our 6-dimensional data to see if our clusters are useful? And second, data points in higher dimensions tend to be closer together which would affect our distance-based clustering metrics. To improve the evaluation of our K-Means Clustering, we can first apply Principal Component Analysis, another unsupervised method, to reduce the dimensionality of our data.

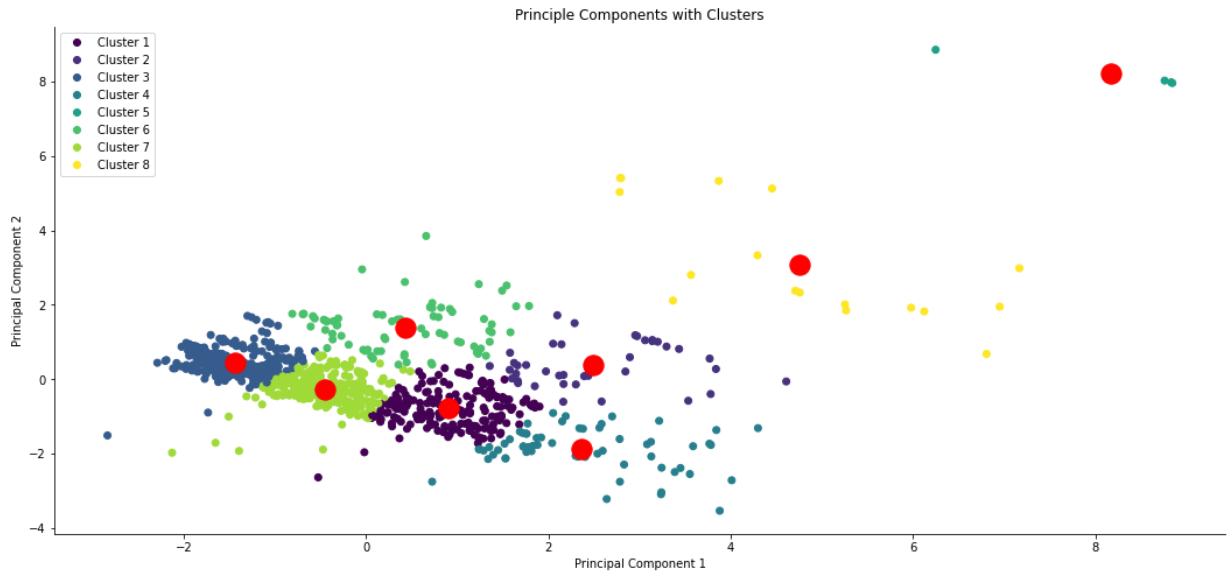
Principal Component Analysis (PCA) is a linear transformation used for dimensionality reduction in data analysis and modeling. PCA takes many variables as inputs and produces new variables, called components, that explain the maximum amount of variance from the original data. In this context, the variance is the variability in the dataset, so by maximizing the variance, PCA extracts as much information as possible from the original variables in order to best describe the data from the original dataset while also reducing the dimensionality. We can create a scatter plot of the first two principal components with the vectors from the original dataset, seen below.



To interpret each principal component we can look at the magnitude and direction of the coefficients above as they relate to the original variables. Relative velocity (v_rel) has the

strongest association with the first principal component and a negative association with the second principal component. Diameter and diameter_sigma have the second and third strongest associations with the first principal component and also have a strongly positive association with the second principal component.

From here, it's relatively straightforward to take our principal components and input those to a K-Means Clustering model using k = 8 clusters.



Unsupervised Evaluation

With respect to the Principal Component Analysis above, we are not sure that our results translate much to the real world. While we can look at the coefficients in the PCA, relative velocity having a strong correlation to the first principal component does not help us cluster the asteroids or determine potentially hazardous asteroids in any meaningful way.

In terms of evaluating the above K-Means Clustering, we do not have distinct boundaries between our clusters. Particularly at lower values of principal component 2 (y-axis), the clusters blend together which suggests that K-Means Clustering may not be the most useful model in this situation. We can also look at the silhouette score of 0.414 which suggests that K-Means is a moderate clustering technique but not particularly strong.

Additionally, combining PCA and K-Means Clustering may not be the best combination of unsupervised methods. PCA transforms the original dataset into principal component space. When we visualize the clusters, we are visualizing them in principal component space which is much more challenging to interpret. There's a trade-off between interpretability and effectiveness of the model, as K-Means is less likely to produce useful clusters in higher dimensional space. However, K-Means is still useful in identifying outliers as seen with Cluster 5

in the visualization above. As stated above, the interpretation of these unsupervised methods remains challenging.

Discussion

Part A

Our first takeaway from this project is that we are pleasantly surprised with how successfully we can predict potentially hazardous asteroids. Our supervised learning methods generated models that can predict PHAs significantly better than the dummy classifier.

Our natural intuition based on our dataset was that an asteroid's characteristics like its diameter and velocity would not have any causal relationship with whether or not it comes close to Earth. In the vastness of the solar system, these flying rocks should be moving at random, stochastically floating through space in whatever trajectory they were first set on. One would not expect that the size or speed of an asteroid could dictate where it would end up.

However, our supervised models produced reasonably accurate predictions of their eventual proximities to Earth based on these features. The success of the models therefore suggests that perhaps our intuition is fundamentally wrong. That is, it could be that the likelihood of an asteroid coming dangerously close to Earth *is* a function of these features. Several of our classifiers - particularly Random Forest and both Deep Learning models - performed well on this dataset, indicating that this conclusion is not merely a fluke of a singular model.

Another key learning from creating the supervised models was the importance of appropriately cleaning, pre-processing, and splitting the data to be trained for the models. Executing each of these steps properly was vital in ensuring the validity of our models. Since these processes involve a high degree of human discretion, and the possibility of generating functioning models *without* even including some of them, there is plenty of room for error. Some of the most salient examples here are balancing the class distribution of our imbalanced original dataset, scaling the variables to unit variance, and preventing data leakage. None of these pre-processing steps are necessary to create models that work, so it is a matter of awareness and being careful to follow best practices. Still, our models and their conclusions are limited by our knowledge and experience, and in a real-world setting would benefit from the outside perspective of colleagues, supervisors, and potentially even stakeholders.

And lastly, this was mentioned previously, but we would be remiss if we did not address the importance of using undersampling instead of oversampling in this particular analysis. When oversampling the minority class, our results were unbelievably good (95-99% accuracy) and when undersampling, the prediction scores revert to a much more reasonable level (61-69%). We are pleased to have been able to identify the source of the data contamination at the eleventh hour. If our minority and majority classes did not have such an extreme imbalance of over 10:1, our hypothesis is that a combination of under and oversampling would have yielded

positive results. However, we believe the excessive oversampling of our minority class contaminated our test data set.

Part B

The primary takeaway from the unsupervised learning methods was that clustering does not provide particularly valuable insights for the asteroid data set. We did not find interesting or distinct clusters and while there are some outliers in this data set, they do not provide as much value in the domain of outer space, as opposed to a domain like cybersecurity. It's possible that with additional explanatory variables, we would be able to ascertain more information about the outliers and the potential value they provide, if we were able to obtain data about the mineral composition of the asteroids for example.

The results are not entirely surprising. With the variables in the original dataset focusing mainly on size and velocity, it would be challenging to find unique insights about different asteroids. With more explanatory variables or a more complex dataset we might expect more unique clusters. It is disappointing that the unsupervised methods were not as successful from the perspective of generating new knowledge about our dataset and about asteroids as the supervised learning methods.

In a very idealized world with more time and resources, we would like to find additional data points or data sets that contain more information about the asteroids. As mentioned in the introduction, asteroids have the potential to be very valuable based on the mineral composition. Obviously the data collection would be a nearly impossible task. In a more reasonable scenario, we might try to append more data based on where the asteroid originated from (i.e. collisions of which objects), or examine the trajectories of the asteroids in a time series format.

Ethical Considerations

To most people, asteroids are a curious phenomenon that occur "out there" in space, and will never directly affect humanity, or at least not for many generations. Asteroids also hold a special place in human consciousness as a source of great power and destruction, as evinced by the dramatic extinction of dinosaurs from this planet due to a massive asteroid impact. They are also a potential source of wealth and resources if we ever develop methods for asteroid mining. Therefore, there are several ethical implications of studying asteroid paths and our work here. The most basic ethical consideration is that of our interpretation of our model results. Importantly, we must adhere to basic data science ethics principles such as creating models that are as reliable and truthful as possible. This includes preventing data leakage and other errors that could invalidate our models, as well as being mindful of the conclusions we draw from their results and how we choose to interpret those findings. We must also account for and be sure to communicate any drawbacks or blind-spots we had in producing the models.

More practically, we need to ensure that any third parties that gain access to or use our data do so in an ethical and responsible manner. For example, companies could use our data to profit off of the materials embedded in asteroids. This could harm the environment with the resources needed to reach and mine those asteroids, and contribute to wealth inequality by enriching

those with already vast amounts of wealth. Another issue that could arise is abuse of our findings. Someone could gain access to our models and modify the code to produce false data that suggests an impending future asteroid impact with Earth, and use that knowledge to control groups of people or profit off of their fear. Therefore, we need to employ effective methods to protect our data from misuse and prevent ethical violations.

Statement of Work

André: Data cleaning and pre-processing, supervised learning methods, explored unsupervised learning models, contributed to the final report

Julien: Found dataset, set up API queries, data cleaning and pre-processing, basic supervised learning models, contributed to the final report

Trevor: Unsupervised learning models, contributed to the final report, organizational lead/team project management

Sources

Choi, C. Q., & Harvey, A. (2021, November 22). Asteroids: Fun facts and information about these space rocks. Space.Com. Retrieved May 12, 2022, from https://www.space.com/51-asteroids-formation-discovery-and-exploration.html

Reuell, P. (2019, July 9). Harvard study suggests asteroids might play key role in spreading life. Harvard Gazette. Retrieved May 12, 2022, from https://news.harvard.edu/gazette/story/2019/07/harvard-study-suggests-asteroids-might-play-key-role-in-spreading-life/

Colab Notebook

https://colab.research.google.com/drive/16SHwrHcYeJEZ5WCwv8Xzm4oM9Jwd6est