Tanvi Dalal
Mark Moreno
Boran Sheu
Parthiv Borgohain

Problem Statement:
Can we predict through certain variables whether a certified asteroid will be hazardous or not?

Description:
This dataset consists of NASA-certified asteroids that are classified as the nearest earth object. This has information about several asteroids such as the diameter, relative velocity, and more. The question we are trying to answer is can we predict through certain variables whether a certified asteroid will be hazardous or not?

Importance:
Space is often a topic that we are unfamiliar with. Why? The reasons we came up with are "The distance", "The unknown", and "The dangerousness" which causes the universe to be relatively less mentioned daily.
Although outer space may seem like a blank space to us, it is actually filled with an infinite number of objects. By orbiting or existing near Earth, there is a certain probability that they will disrupt Earth's natural phenomena. And thus, we wanted to find out what are the relevant factors or combination of factors for these asteroids to be hazardous to Earth. By figuring out such features, scientists might be able to come up with crisis management plans beforehand and would definitely increase human welfare.

Exploratory analysis:
First, our data consisted of 90836 rows and 10 columns and had 6 variables that were usable in our analysis. The variables that we used were the estimated diameter minimum, estimated diameter maximum, absolute magnitude (a variable that finds the luminosity of a certain asteroid with lower numbers meaning a closer asteroid), relative velocity, the distance it missed the earth, and our target variable, whether it was hazardous to earth or not. Some data cleaning had to be done to remove the useless variables such as sentry object and orbiting body. Along with this, we were able to remove the estimated diameter maximum because est_diameter_minimum was perfectly correlated with it.
Using the NASA Near earth objects dataset we were able to find some interesting patterns in the data that we will continue to use in the next section of the report. To get a better understanding of the data we plotted the correlations (Fig. 5). This correlation chart showed the variable with the strongest relationship with our target variable was the absolute magnitude variable with a -.37 correlation. This falls into a moderate relationship category and we will continue to analyze this relationship in the future parts of the research. Another interesting data point we found in the analysis was

in the ***estimated diameter max*** variable. While not a strong pattern, using the group by functions we were able to find that only 5% of asteroids under .25 kilometers were hazardous compared to all the other sizes greater than that which were hazardous at about a 40% rate(Fig. 6,7). Another insight from the data that was found involved the ***absolute magnitude*** variable. We found that the only asteroids that were hazardous were the asteroids that had an absolute magnitude of 24 or less (Fig. 8). This shows us that the objects closest to the earth are the most dangerous and past a certain point they are simply too far from earth to be a danger to us. Finally, we will use machine learning to try and use these variables to help us classify these asteroids as hazardous.

Solution and Insights:

-Naive Bayes:

The NASA data that we are running is classification data and our focus is to find out the probability of **hazard** happening given each different factors that occured. We believe that Naive Bayes can help us take a look at parameter fitting in a very simple and intuitive way.

What we did with the data:

1. <u>Initial problems</u>: When trying to bin "estimate diameter min & max", if we just give it a qcut of 5, the range would be quite widespread (see Table 4). The last interval ranges dramastically and we wonder, what if the higher value of estimated diameter actually has more influence than the others that are under 1? (This is what we got in the EDA process, the higher parts of the bins get a more even result.) We tried to make the qcut into 10, and it lowered the accuracy.
2. <u>How we solved it</u>: We decided to make it into a binary situation, split it into values that are higher than "2" and those that are under "2". The accuracy improved from 76% to 89%.

What were the results and How can we interpret them:

1. <u>Accuracy</u>: For the training data set and testing data set, we came up with the accuracy of 89.69% and 89.61%. If we look at the prior probability for the negative class, it is 90.13%, which means that if we all guess negative, we have 90.13% to get it right! We can conclude that the Naive Bayes model is not performing well.
2. <u>Importance:</u> By running the ratio Positive class/ Negative class, if Ratio is far away from1⇒ a feature is important! From the feature stats (see Table 5), absolute magnitude is the most important feature and miss distance is the least important feature.
3. <u>F1 score:</u> The train F1 score is 0.05 and test F1 score is 0.06 due to recall value being close to zero (0.036).

-Logistic Regression:

In the dataset, we are trying to predict the change in hazardous given the different variables. Since **hazardous** in the dataset is listed as either

“True” or **“False”**, this becomes a classification problem. For this, we need to understand each variable’s importance, and to do that we tried logistic regression as one of our approaches.

Process:

Initially we made the dependent variable **“hazardous”** into binary numbers to run this regression, and then fit a logistic regression with all the applicable variables: est_diameter_min, est_diameter_max, relative_velocity, miss_distance, and absolute_magnitude.

Results:

Our baseline accuracy was **0.90242981** and our prediction accuracy was **0.90326960** which is an improvement of around 0.08%. Since this was not a very large increase, we can conclude that logistic regression is not the best model to use for this dataset.

After looking at the weights of each variable, we found that the most significant variables were **est_diameter_min** and **est_diameter_max** and the least significant was **relative_velocity**. This is interesting because we can see that it is not the speed of the asteroid that is important, but rather the size of the asteroid.

-KNN:

Process:

We ran a Knn on the target variable, hazardous, after changing it into a numerical variable. We ran all of the variables individually and together to see which model produced the best results.

Results:

There was no significant change in the baseline accuracy of **.90242981** and the final accuracy of my combined model of all the variables ran in a single KNN model. This model predicted only 5 true positives leading to an F score that was incredibly small of .**0006** on the training data. This was nearly the same in all of the models we had created. True positives were hard to predict in this dataset due to the unbalanced nature of the data along with small variations between the data.

-Trees and other Ensemble Methods:

We tried to fit decision tree and other ensemble learning models on our training and test datasets. As shown in Table 1, the decision tree was clearly overfitting the training data with **100%** Training Accuracy and only **89.3%** Test accuracy which was even lesser than the baseline accuracy. Gradient Boosting was giving us the best test accuracy of **91.52%** which was marginally better than the baseline accuracy.

After this, we tried to tune our parameters for Random Forests and Gradient Boosting to try and improve our accuracy by varying the no. of trees and max depth for the gradient boosting model (illustrated in Figures 1 and 2). We also tried to find out the best value no. of predictors to take for random forests (Figure 3).

After tuning our parameters and running the new models again on training and test data, our accuracy and F1 score figures improved slightly. The figures obtained are shown in Table 2.

Results:

The precision and recall values obtained on the Test dataset for our Gradient Boosted Decision Tree Model are **0.65** and **0.38** (Confusion Matrix is given in Table 3). The accuracy on the test set is **92.015%**. The F1 score obtained is **0.48**. So, we have obtained a reasonable improvement over a hypothetical baseline model which always predicts 'No'.

As for feature importance in the Gradient Boosted Decision Tree, nearly all the four variables have similar importance as shown in Fig. 4. As per the plot, miss_distance is the most important feature.

It seems like no feature is dominant. Perhaps this explains the not-so-high F1 Score obtained by the gradient boosted tree model for the test data.

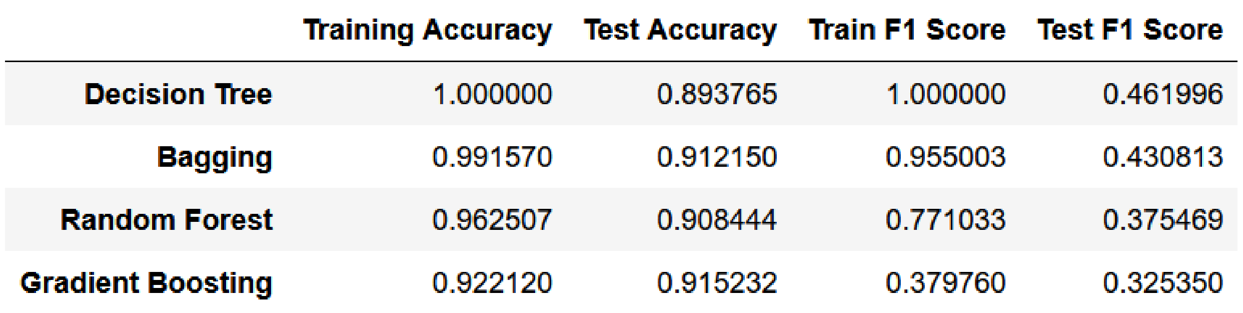
Conclusion and Next Steps:

So, the following are some of the key takeaways of our analysis:

- Only **4 variables** are being used to predict **whether a NFO is hazardous or not** (**miss_distance, absolute magnitude, est_diameter_max and relative_velocity**). The remaining variables are not useful for predicting the target variable and were dropped before fitting the models.
- None of the 4 selected variables seem to be heavily impacting the target variable. All **4 variables seem to have similar values for feature importance** in the Gradient Boosted Tree Model. Other classification models did not give a wide variation in relative importance of features.
- Given the aforementioned point, it seems reasonable to expect that the predictive model built will not have significant improvement over the baseline model as the given 4 variables do not seem to be great predictors. Our best model obtained a **test accuracy of 92.015%** compared to the **baseline accuracy of around 90%** and an **F1 score of around 0.48**
- Among the 4 selected variables, **miss_distance** seems to be the **most important feature** as per the best model (Gradient Boosted Decision Tree).

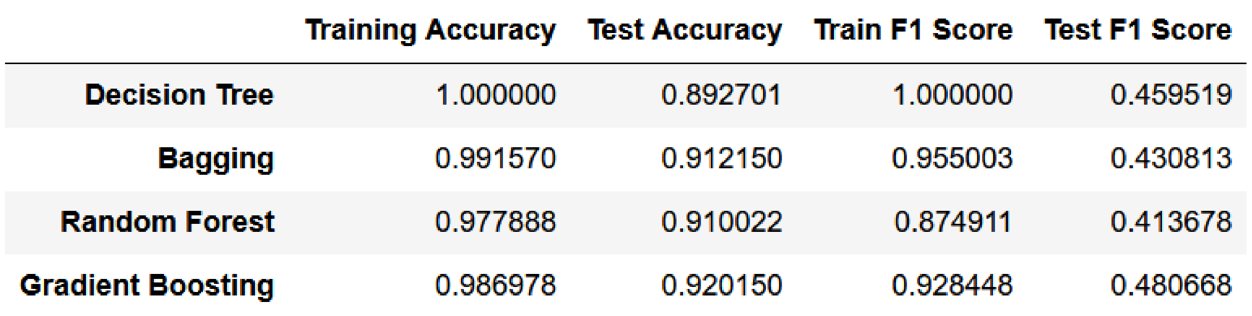
As a next step, we can try working with a bigger dataset with a less uneven distribution of True and False values for the target variable. This might help us build a better predictive model. Including more features in the model will also probably help us arrive at a better model as the current variables did not turn out to be highly impactful in predicting the target class.

Appendix



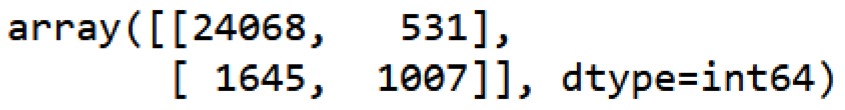
	Training Accuracy	Test Accuracy	Train F1 Score	Test F1 Score
Decision Tree	1.000000	0.893765	1.000000	0.461996
Bagging	0.991570	0.912150	0.955003	0.430813
Random Forest	0.962507	0.908444	0.771033	0.375469
Gradient Boosting	0.922120	0.915232	0.379760	0.325350

Table 1



	Training Accuracy	Test Accuracy	Train F1 Score	Test F1 Score
Decision Tree	1.000000	0.892701	1.000000	0.459519
Bagging	0.991570	0.912150	0.955003	0.430813
Random Forest	0.977888	0.910022	0.874911	0.413678
Gradient Boosting	0.986978	0.920150	0.928448	0.480668

Table 2

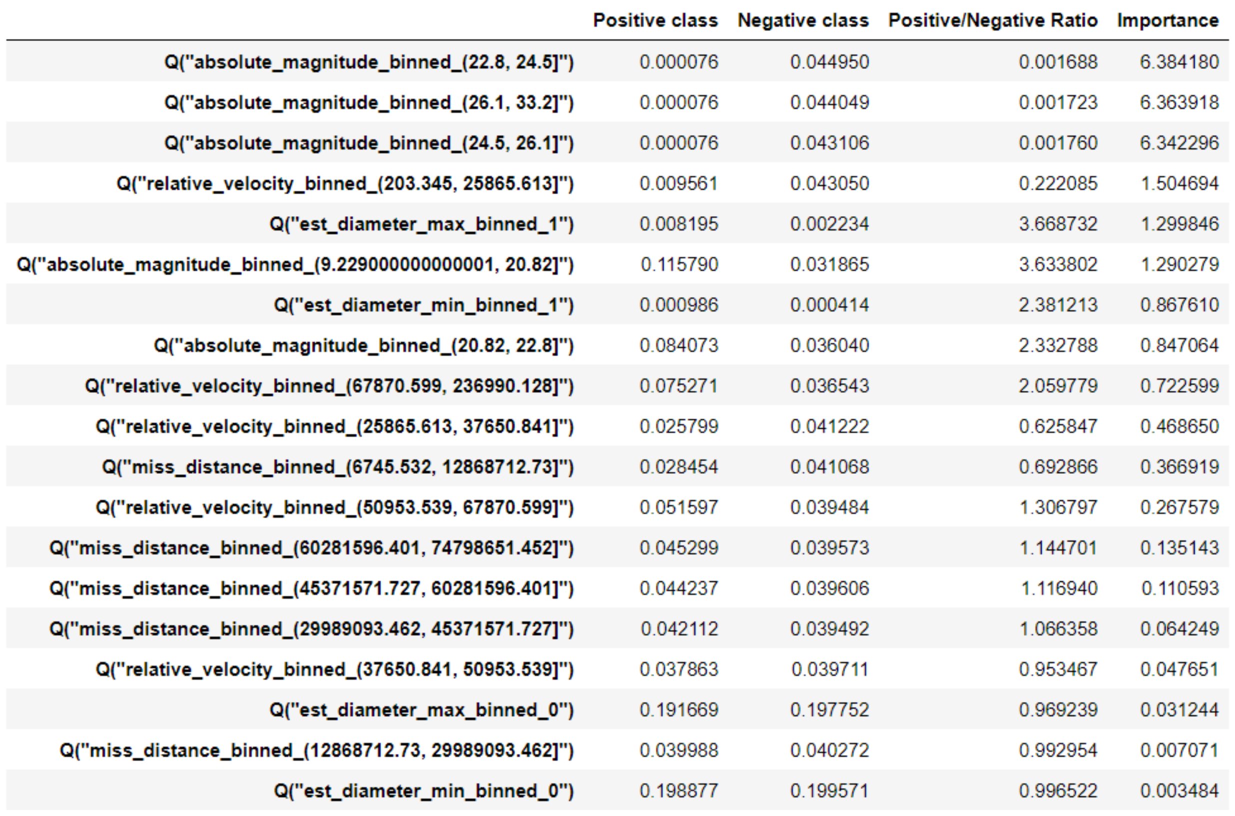


```
array([[24068,   531],
       [ 1645,  1007]], dtype=int64)
```

Table 3

[(-0.0003910, 0.016] < (0.016, 0.0335] < (0.0335, 0.0732] < (0.0732, 0.182] < (0.182, 37.893]

Table 4



	Positive class	Negative class	Positive/Negative Ratio	Importance
Q("absolute_magnitude_binned_(22.8, 24.5]")	0.000076	0.044950	0.001688	6.384180
Q("absolute_magnitude_binned_(26.1, 33.2]")	0.000076	0.044049	0.001723	6.363918
Q("absolute_magnitude_binned_(24.5, 26.1]")	0.000076	0.043106	0.001760	6.342296
Q("relative_velocity_binned_(203.345, 25865.613]")	0.009561	0.043050	0.222085	1.504694
Q("est_diameter_max_binned_1")	0.008195	0.002234	3.668732	1.299846
Q("absolute_magnitude_binned_(9.229000000000001, 20.82]")	0.115790	0.031865	3.633802	1.290279
Q("est_diameter_min_binned_1")	0.000986	0.000414	2.381213	0.867610
Q("absolute_magnitude_binned_(20.82, 22.8]")	0.084073	0.036040	2.332788	0.847064
Q("relative_velocity_binned_(67870.599, 236990.128]")	0.075271	0.036543	2.059779	0.722599
Q("relative_velocity_binned_(25865.613, 37650.841]")	0.025799	0.041222	0.625847	0.468650
Q("miss_distance_binned_(6745.532, 12868712.73]")	0.028454	0.041068	0.692866	0.366919
Q("relative_velocity_binned_(50953.539, 67870.599]")	0.051597	0.039484	1.306797	0.267579
Q("miss_distance_binned_(60281596.401, 74798651.452]")	0.045299	0.039573	1.144701	0.135143
Q("miss_distance_binned_(45371571.727, 60281596.401]")	0.044237	0.039606	1.116940	0.110593
Q("miss_distance_binned_(29989093.462, 45371571.727]")	0.042112	0.039492	1.066358	0.064249
Q("relative_velocity_binned_(37650.841, 50953.539]")	0.037863	0.039711	0.953467	0.047651
Q("est_diameter_max_binned_0")	0.191669	0.197752	0.969239	0.031244
Q("miss_distance_binned_(12868712.73, 29989093.462]")	0.039988	0.040272	0.992954	0.007071
Q("est_diameter_min_binned_0")	0.198877	0.199571	0.996522	0.003484

Table 5

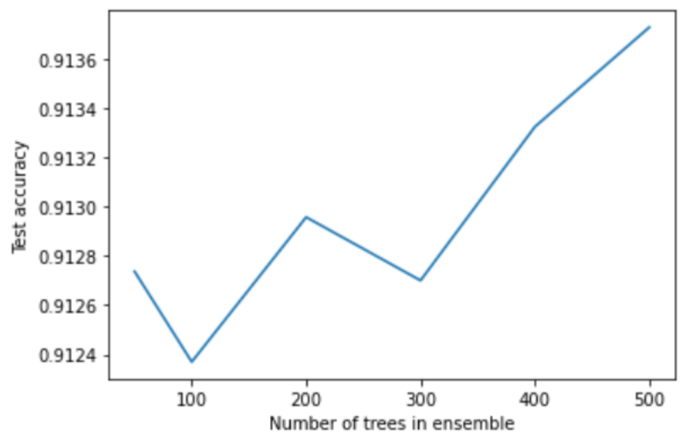


Fig. 1

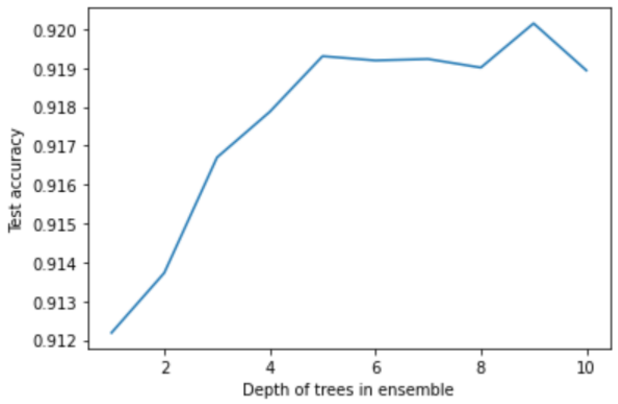


Fig. 2

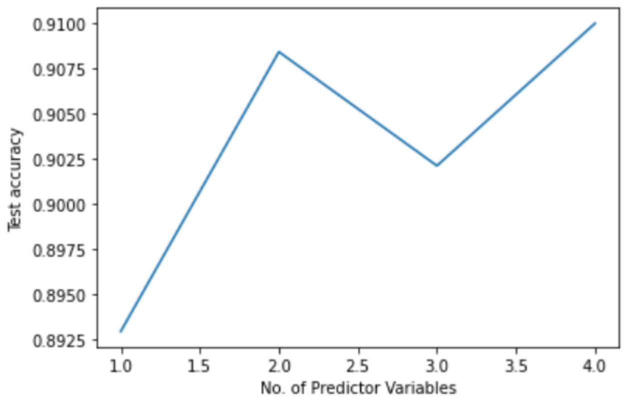


Fig. 3

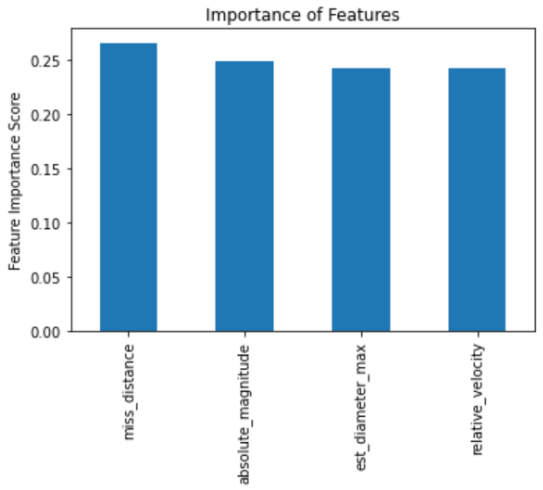


Fig. 4

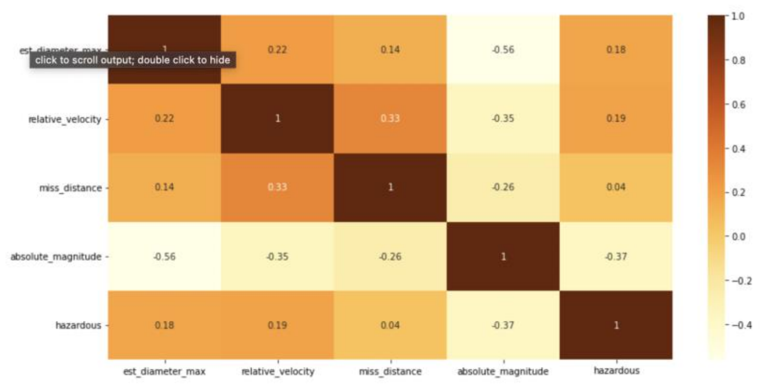

est_diameter_max
click to scroll output; double click to hide
relative_velocity
miss_distance
absolute_magnitude
hazardous
est_diameter_max
relative_velocity
miss_distance
absolute_magnitude
hazardous

Fig 5.

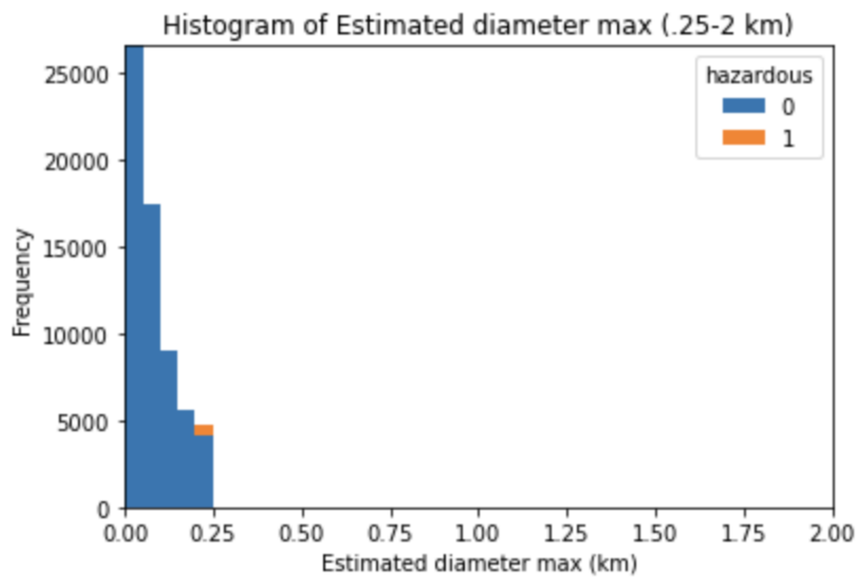

Histogram of Estimated diameter max (.25-2 km)
hazardous
0
1
Frequency
Estimated diameter max (km)

Fig 6.

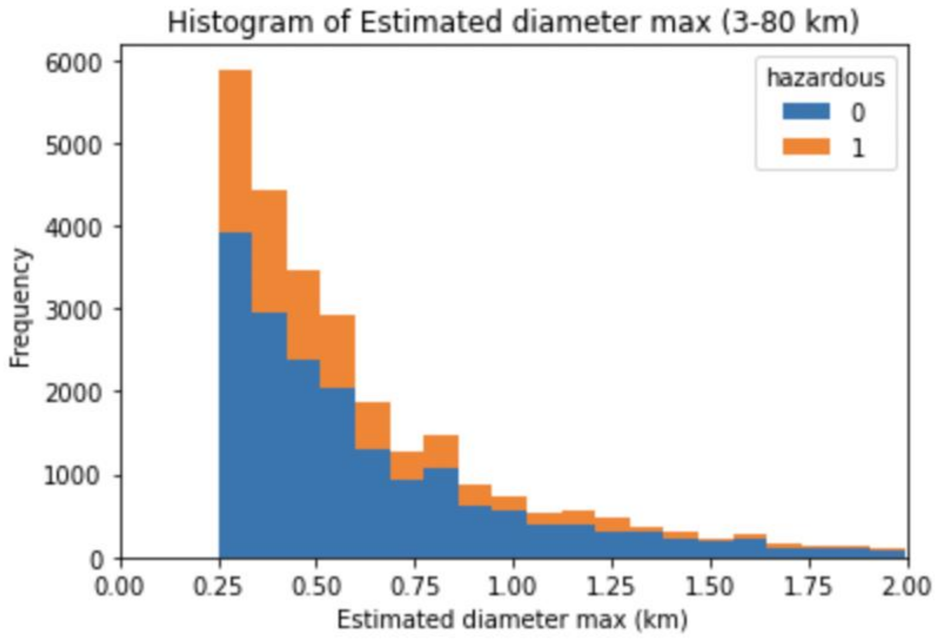


Fig 7.

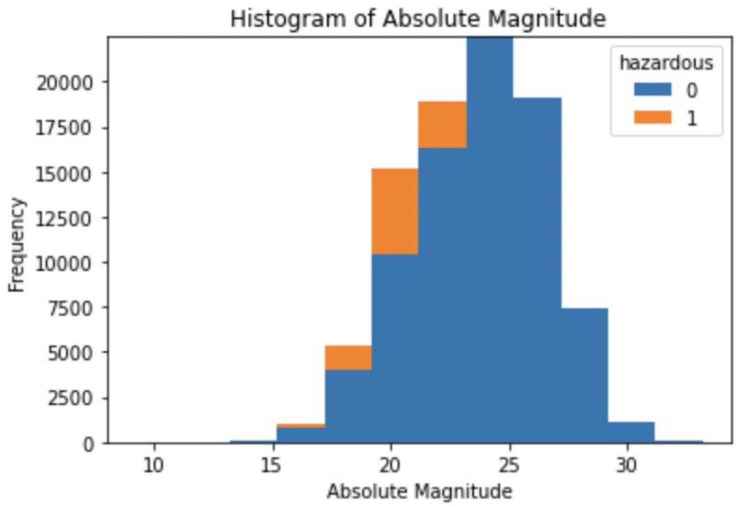


Fig 8.

https://www.kaggle.com/datasets/sameepvani/nasa-nearest-earth-objects

Dataset.