# Classification using Logistic Regression
# Asteroids

Bobby Gabriel, bgabriel@bellarmine.edu

**ABSTRACT**
Asteroid impacts can have devastating effects on Earth and Earth's ecosystem, so it is important to gain insight on potential factors that could make an asteroid catastrophic. Logistic Regression classification is a method we will use to identify hazardous asteroids by their physical characteristics. By analyzing these features, we can hope to estimate the probability of an asteroid being hazardous.

## I. INTRODUCTION
This dataset is called the "NASA: Asteroid Classification" and it assesses the speed, magnitude, location, approach date, and other factors of various asteroids. It contains metrics for asteroids with 40 different variables and can be found at https://www.kaggle.com/datasets/shrutimehta/nasa-asteroids-classification. The target of this classification is the 'Hazardous' variable which will predict whether an asteroid is hazardous or not based on all the other variables. I chose this dataset because it looked like a good dataset to compare variables and it contained a decent amount of data points for a proper classification.

## II. BACKGROUND
### *A. Data Set Description*
This data set contains 4687 samples with 40 columns with either data type float or data type bool. Additionally, each column is filled and contains no missing data. It can be found on Kaggle.com with the link above and I chose it because I am interested in space. This dataset was collected by NASA and was accessed using NeoWs (Near Earth Object Web Service). A complete listing of all the variables is shown in **Table 1.**

### *B. Machine Learning Model*
Logistic regression is a method used to analyze and model the relationship between a binary dependent variable and either one or more independent variables. It attempts to estimate the probability of the target variable's value using the values of the independent variables. Logistic Regression is mostly used for classification but can also be used for regression. It is used in fields such as psychology, data science, economics, finance, and many more.

## III. EXPLORATORY ANALYSIS
Contains 40 columns with 4687 samples. Due to the number of variables, I will only be listing the variables I used in my final model which is 24.

**Table 1: Data Types**

| *Variable Name* | *Data Type* |
|---|---|
| V1 Absolute Magnitude | Float64 |
| V2 Est Dia in Feet(min) | Float64 |
| V3 Est Dia in Feet(max) | Float64 |
| V4 Epoch Date Close Approach | Float64 |
| V5 Relative Velocity km per sec | Float64 |
| V6 Miles per hour | Float64 |
| V7 Miss Dist. (Astronomical) | Float64 |
| V8 Orbit ID | Float64 |
| V9 Orbit Uncertainty | Float64 |
| V10 Minimum Orbit Intersection | Float64 |
| V11 Jupiter Tisserand Invariant | Float64 |
| V12 Epoch Osculation | Float64 |
| V13 Eccentricity | Float64 |
| V14 Semi Major Axis | Float64 |
| V15 Inclination | Float64 |
| V16 Asc Node Longitude | Float64 |

| | |
|---|---|
| V17 Orbital Period | Float64 |
| V18 Perihelion Distance | Float64 |
| V19 Perihelion Arg | Float64 |
| V20 Aphelion Dist | Float64 |
| V21 Perihelion Time | Float64 |
| V22 Mean Anomaly | Float64 |
| V23 Mean Motion | Float64 |
| V24 Hazardous | bool |

## IV. METHODS

### *A. Data Preparation*

The first thing I did for data preprocessing was drop unneeded columns. I dropped 13 columns that contained extraneous information or information that was contained in other variables. I then dropped 2 columns that only contained 1 unique value, so they weren't needed. The last thing I had to do was change the target variable 'Hazardous' to an int data type. Currently, it is either true for hazardous or false for non-hazardous, so we change it to 1 or 0. After this, I could then split it up into dependent (Hazardous variable) and independent variables (all other variables).

### *B. Experimental Design*

***All experiments done with Logistic Regression***

**Table X: Experiment Parameters**

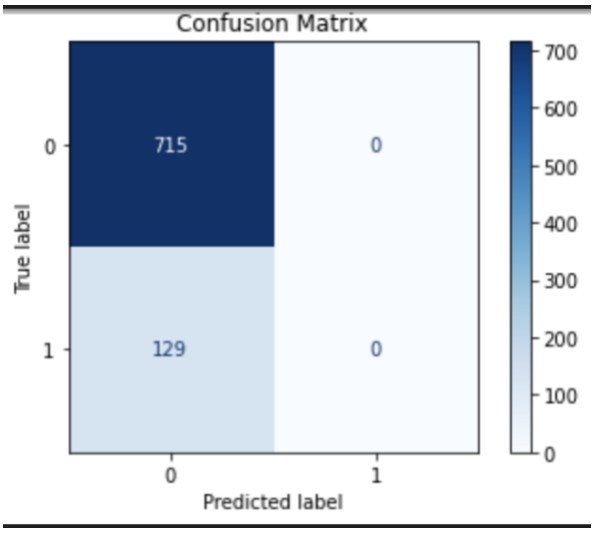
| Experiment Number | Parameters |
|---|---|
| 1 | 80/20 train test split with 50/50 validation split<br><br>Results:<br>**Logistic Regression Accuracy: 82.7%** |
| 2 | 80/20 train test split with 60/40 validation split<br><br>Results:<br>**Logistic Regression Accuracy: 83.5%** |
| 3 | 70/30 train test split with 50/50 validation split<br><br>Results:<br>**Logistic Regression Accuracy: 85.9%** |
| 4 | 70/30 train test split with 60/40 validation split<br><br>Results:<br>**Logistic Regression Accuracy: 84.7%** |

### *C. Tools Used*

The following tools were used for this analysis: Python v3.9.12 running the Anaconda 4.3.22 environment for Apple Macintosh computer was used for all analysis and implementation. In addition to base Python, the following libraries were also used: Pandas 1.4.2, Numpy 1.21.5, SKLearn 0.18.1. I chose these python packages because this is what we use in class.

## V. RESULTS

### *A. Classification Measures*




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.85 | 1.00 | 0.92 | 715 |
| 1 | 0.00 | 0.00 | 0.00 | 129 |
| accuracy | | | 0.85 | 844 |
| macro avg | 0.42 | 0.50 | 0.46 | 844 |
| weighted avg | 0.72 | 0.85 | 0.78 | 844 |

*B. Discussion of Results*

Overall, the model did very well in detecting when an asteroid was not hazardous (0) but could not detect an asteroid that was hazardous (1). The f1-score of the not hazardous side was 92% with a precision of 85%. The scores for the hazardous side were all 0. I believe the model did not do well with a hazardous asteroid due to the low support values. For hazardous, the support was 129 samples while the non-hazardous samples were 715.

*C. Problems Encountered*

The first problem I encountered was finding a suitable dataset to perform the logistic regression on. When looking through Kaggle, it was easy to find datasets suited for simple regression analysis, but harder for classification problems. Another problem I had was finding good train/test split numbers. I tinkered with the numbers a lot before finding a good score with the model.

*D. Limitations of Implementation*

The model performed well for True Negatives but badly for True positives. In other words, it was good at guessing a 0 for hazardous, but bad at guessing a 1. I don't believe this is a problem with the logistic regression, rather than a problem with the dataset and samples provided. The support for the model was only 129 data points for 1 and 715 data points for 0. I believe that is why the guesses for 1 were so low.

*E. Improvements/Future Work*

In the future, I think I could try a different classification model. Logistic Regression is a rather simple model, so I could try exploring different possibilities in that area. I think there were enough variables with 25, but there could have been more data points to help the logistic regression model. Also, if I tinkered with the train/test split and validation split a little bit more I probably could have gotten the model accuracy around 87-90%.

## VI. CONCLUSION

For this project, I utilized a logistic regression model to attempt to classify whether an asteroid was hazardous or not based on 25 different variables. Overall, the model achieved an accuracy score of 85% with all of that coming from correctly identifying the non-hazardous asteroids. With these asteroids, the model obtained a recall score of 100%, and an F1 score of 92% on 715 samples. For the hazardous asteroids, the model performed very badly with a 0% in all score indicators. I believe this was because of a low sample size of 129, but it could be due to other facts. In the end, it was a decent model at 85%, but it could use some more work to detect hazardous asteroids.

## REFERENCES

*http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html*
*https://www.kaggle.com/datasets/shrutimehta/nasa-asteroids-classification*