

Asteroid Mining Corporation Script

By Oneeb Aamer

The Asteroid Mining Corporation challenge consisted of writing a script to classify asteroids based on their visible wavelength spectra. Although I have no prior knowledge in this field, I decided to approach the problem from a computer scientist's point of view. This means that data was abstracted into graphs and arrays so that I could focus on the challenge at hand.

How to use the script:

1. Download the script and the "trainingData" file. Ensure that they are placed in the same location, and that the script is outside of the data files.
2. Move the input data (csv file) into the same location as the script.
3. Run the script. It will ask you to enter the name of the csv file that you want to test. Make sure to match the cases
4. After entering the name, the program will train the algorithm and begin to output results based on how likely they are to be the asteroid's classification. The longer the application runs, the more precise the results will be, until the algorithm has gone through each test case.

How it works:

Although I could have used a machine learning approach, I decided to design my own algorithm entirely as I felt that was more fitting towards the challenge. My algorithm works by converting the training data and test data into 2d arrays and then finding the difference of each reflectance individually. It also considers uncertainties, where a higher uncertainty means the difference will be weighed less. While this process can be accurate, it is very slow. That is why I decided to take a heuristic approach.

The program will keep running until the computer has a definite answer, which could take forever, or until the user is satisfied with the probability. As the computer trains the probability grows stronger, and the user gets a more accurate answer. This could take several minutes or seconds, it all depends on the input data.

Requirements:

The script requires Python to be installed alongside all the standard libraries. The libraries used were csv, glob, os, math and time. Other than that, the script can be run on any device which supports python.

Future improvements:

I acknowledge that my algorithm has room for improvement, which would be possible with more time. The main improvement would be to make it more efficient, as the algorithm is already quite accurate. Part of the reason why the script is slow is due to the massive use of loops in my algorithm. This was necessary to iterate over every data point and determine the difference with the asteroid's data.