Near-Earth Objects

CS 556

November 28, 2022

1 Dataset

This dataset provides information on a set of Near-Earth Objects (NEOs) which have been detected and monitored by NASA. Each of these objects has been watched by telescopes and detecting equipment and various data about its position, velocity, and makeup have been recorded. A few of these data points have been detailed in the following data set, along with NASA's best estimate as to whether any of these objects pose a threat or hazard to Earth at any point in the near future.

Some of the relevant data fields recorded are:

- Object Name ('name'): NASA names each near-Earth object based on how and when it was observed.
- Diameter ('est_diameter_min' and 'est_diameter_max') The estimated diameter or size of the object is recorded, both with minimum and maximum values. If the two are equal, the NEO is roughly spherical; two wildly different values indicate an irregularly-shaped object.
- Velocity ('relative_velocity') The velocity of the object (at time of detection) relative to Earth, in m/s.
- Miss Distance ('miss_distance') The minimum distance (*periapsis* if in Earth's sphere of influence) of the NEO relative to Earth.
- Luminosity (‘absolute_magnitude‘) Measures the radiant energy released or reflected from the body over a fixed amount of time. Higher values indicate brighter objects.
- Hazard (‘hazardous‘) NASA has provided a ‘true‘/‘false‘ value indicating whether they consider this object a threat. This is our target variable.

2 Your Task

Your task is to train machine learning models which can predict, based on the given features, whether an unknown NEO should be considered hazardous or not. First, we'll need to load the provided dataset into a Jupyter Notebook from the provided CSV file. The names of the columns will correspond to those given above.

Take a look at a minimum of three of the features given, and indicate how you think they're distributed and any potential outliers that you'll need to handle before your data can be effectively used. Then, utilizing **only** the tools and processes outlined in this course, you'll train a logistic regression or SVM model on 80% of the dataset (training set) to predict whether a NEO should be considered hazardous or not. Test your model on the remaining 20% of your data and report the accuracy, precision, recall and F1 score of your model.

Next, use PCA to reduce the dimensionality of the dataset to two and then train a new model to predict whether a given NEO is hazardous or not using the new resulting dimensions as features. Generate a scatter plot showing the hazardous objects in red, non-hazardous objects in green and the decision boundary in black. Test your new model and report the accuracy, precision, recall and F1 score. Compare the performance of this new model with the previous one. Which is more effective, and why?

3 Submission

A project submission consists of four files:

1. PDF of exactly one-page reporting:
   (a) CWID / name of student
   (b) Accuracy, precision, recall and F1 score for both models
   (c) salient project features (describe your design choices)
2. CSV files of predictions of Hazard and actual labels for the test set for both models.
3. Working Jupyter Notebook that produces the CSV.
4. Jupyter Notebook downloaded as a python file.

4 Evaluation

Evaluation will be done based on:

- Precision, recall and F1 scores
- Techniques
- Code quality

5 Code of conduct

All scripts/notebooks will be checked (1) against each other (2) against an online database, using plagiarism detection tools. Any case of plagiarism is a serious incident and is susceptible to be reported to the competent authorities of Stevens. Plagiarism is intended as sharing a large fraction of the code that performs critical operations. Plagiarism does not include sharing small isolated pieces of code that perform routine tasks (e.g. input/output, or basic normalization/imputation) if in aggregate they represent a small portion of the code.