<u>Project Title:</u> Near-Earth Object (NEO) Detection

Brief Description:

This project simulates the detection of Near-Earth Objects (NEOs) by analyzing telescope configurations, sky conditions, and object properties. With over 30,000 cataloged NEOs, the goal is to optimize observational strategies using data processing, visualization, photometric modeling, and deep learning. The simulation helps understand which conditions improve NEO detectability and supports planetary defense and space research.

Project Link:

https://github.com/ExoSphyre/Neo-ExoSphyre-Detecting-near-Earth-objects

---

Problem Statement and Impact:

NEOs pose potential threats to Earth and are crucial for future space missions. However, detecting them is challenging due to their faintness, motion, and sky coverage limits. Current surveys often miss smaller or distant objects.
This project addresses that by:

- Simulating brightness, visibility, and sky location.
- Estimating how many NEOs are observable with given telescope parameters.
- Applying deep learning to classify NEOs for better tracking and study.

This aids astronomers and researchers in developing more efficient sky surveys and risk assessments.

Objectives:

- Analyze NEO data and create a structured database.
- Simulate NEO brightness and apparent magnitude.
- Identify which NEOs are currently observable based on survey limits.

- Plot spatial distribution of faint and visible NEOs.
- Classify NEO types using a deep learning model.

Contributors: This project was built by a team consisting of:

- **Sandipan Rakshit**
- **Niloy Das (Me)**
- **Mrigangana Sarkar**

Tools/Tech Used:

- Python, SQLite, pandas, NumPy, matplotlib, seaborn
- SPICE (spiceypy) for astronomical geometry and ephemerides
- TensorFlow / Keras for deep learning
- Sky plotting using ecliptic coordinates

Implementation:

Step 1: Data Collection & Preparation

- Downloaded NEO data from NEODyS and stored it in a SQLite database.
- Computed aphelion and perihelion using orbital equations.
- Organized features like eccentricity, magnitude, inclination, etc.

Step 2: Exploratory Data Analysis

- Loaded the dataset and performed statistical analysis (mean, min, max).
- Estimated NEO diameters using magnitude-based formulas.
- Visualized data using histograms, KDE plots, and correlation grids.

Step 3: Apparent Magnitude Simulation

- Computed how bright NEOs appear from Earth.
- Simulated brightness for known objects like (433) Eros at specific times.
- Visualized apparent magnitude vs phase angle and Earth distance.

Step 4: Current Visibility Analysis

- Calculated current brightness of all known NEOs.
- Compared visibility against survey limits (Pan-STARRS mag = 24).
- Found only ~17% are currently observable due to faintness or positioning.

Step 5: Sky Distribution of Faint NEOs

- Computed positions of faint NEOs in ecliptic coordinates.
- Applied filters (e.g., mag > 25, dist < 1 AU) for visualizing undetectable yet nearby objects.
- Used Aitoff projection to show clustering in the sky.

Step 6: Combined Visibility Analysis

- Combined techniques to generate a sky distribution map of both visible and invisible NEOs.
- Showed that ~5,000 are visible under current conditions, while ~24,000 remain undetected.

Step 7: Deep Learning Classification

- Preprocessed the SQLite dataset by scaling features and encoding classes.
- Built a deep neural network with 4 hidden layers to classify NEOs as Apollo, Amor, etc.
- Achieved 99.21% test accuracy and demonstrated excellent classification via a confusion matrix.

Results/Performance:

- **Database Built**: From raw NEODyS data with added orbital calculations.
- **Apparent Magnitude**: Simulated with phase angle and distance; visibility computed.
- **Current Visibility**: Only ~17% of NEOs are bright enough to be detected (mag < 24).
- **Sky Plot**: Faint NEOs cluster in specific regions, useful for targeted surveys.
- **Classification Model**:
  - **Training Accuracy**: >99%
  - **Test Accuracy**: 99.21%
  - **Model**: Deep Neural Network (TensorFlow/Keras)
  - **Performance**: Excellent classification across all NEO types.