# NEO Guardian: Planetary Defense UI


Created by Yasassri Ekanayake

(Undergraduate at SLIIT  (Bsc. In information Technology spec. in Data science))

**Email:-**

Yasas0721@gmail.com

**Phone:-**

m. 0772016432
L. 0812492072

**Address:-**

315/1.Ketapidalla Road,

Gohagoda,

Katugasthota,

Kandy,

Sri Lanka

# Before Introduction about the project and work

### *Disclaimer : A Note from the Developer*

Before I proceed to discuss the technical aspects of Neo Guardian, I would like to provide a context for this application. **This application was developed primarily as a means of obtaining experience and pursuing my interest in Machine Learning (EDA) and Astronomy**.

This application represents my first major project that combines these two areas of interest. As this application represents a learning experience, I also acknowledge that there may be areas that could potentially be improved.

Therefore, if you notice any inaccuracies within the application from a technical perspective, misinterpretations of astronomical data, or areas where the application could potentially improve from a Machine Learning perspective:

Please don't hesitate to provide feedback. I would love to learn how I could improve the accuracy of the application or its functionality.

Thankyou for taking the time to review my work!

Yasassri Ekanayake,

Undergraduate at SLIIT (Bsc. In information Technology spec. in Data science)

Sri Lanka

# Introduction – The Frontier of Planetary Defense

In the modern astronomical era, the detection of Near-Earth Objects (NEOs) has accelerated significantly. While thousands of asteroids are tracked daily by agencies like NASA and ESA, the raw data—consisting of orbital elements, absolute magnitudes, and complex velocities—remains largely unintelligible to the general public and non-specialist stakeholders.

NEO Guardian was developed to bridge this critical information gap. It is a sophisticated, AI-driven interface designed to translate complex astronomical datasets into intuitive, visual, and actionable intelligence. By leveraging Machine Learning (ML), specifically a Random Forest classification architecture, the system evaluates the risk profile of an object in real-time, providing a "probability of hazard" that goes beyond simple binary labels.

This project serves two primary purposes:

Decision Support: Providing a rapid-scan tool for immediate threat assessment.

Public Education: Using comparative visualization to ground abstract space data into real-world physical dimensions, such as comparing asteroid diameters to iconic global landmarks.

Through the integration of data science and web-based dashboarding, NEO Guardian represents a step forward in making planetary defense data accessible, transparent, and engaging for the next generation of space observers.

### Technical Methodology

The process of creating Neo Guardian was an intensive Data Science process, from raw astronomical data to a functional prediction interface. The technical process can be broken down into four main phases:

**Phase I: Data Acquisition & Pre-processing**

The foundation of this model is built from the **NASA NeoWS Near Earth Object Web Service dataset**, which holds verified historical observations of asteroids.

Data Cleaning: Handling of missing values, removal of non-relevant orbital parameters for threat prediction.

Feature Selection: Determining the "Big Four" variables that define the threat profile of an asteroid.

- Absolute Magnitude (H): Used for estimating mass and size of the asteroid.
- Relative Velocity: Speed of asteroid relative to Earth.
- Miss Distance: Distance of asteroid path from the center of Earth.
- MOID (Minimum Orbit Intersection Distance): Most important parameter for calculating the threat of collision.

**Phase II: Model Selection & Training**

For "intelligence" of Neo Guardian, we used a Random Forest Classifier.

Why Random Forest? The Random Forest classifier works as an "Ensemble." This means that it builds a large number of decision trees during training and outputs a classification that corresponds to the mode of all those decision trees. This minimizes the possibility of "overfitting"; hence, it will work correctly on new, unseen asteroids.

Weighted Importance: The model was optimized to assign the maximum weight to Absolute Magnitude and MOID, as these are critical considerations for planetary defense.

**Phase III: Mathematical Feature Engineering**

To make the data relatable, the system performs real-time mathematical transformations.

- Diameter Calculation : We use the NASA-standard formula to convert Magnitude ($H$) into Meters ($D$).

$$D = \frac{1329}{\sqrt{0.15}} \cdot 10^{-0.2H} \cdot 1000$$

- Orbit Pathing: the system uses Eccentricity ($e$) to plot the visual geometry of the asteroid’s orbit, allowing users to see how”oval” or circular” the path is compared to Earth’s orbit.

Phase IV: UI/UX Development

In the fourth and final phase, the UI/UX development was carried out using the Streamlit interface. The UI was designed with the following features:

- **Visual Hierarchy:** Large Hazard Indicators (Red/Green) for immediate recognition.
- **Comparison Logic:** A dynamic bar chart was implemented to compare the asteroid's size with real-world landmarks like the Eiffel Tower or the Empire State Building.
- **State Management:** Using "Session State" to maintain the availability of mission reports and scan history throughout the user session.

# 3. Functions & Features

Neo Guardian is built with a set of interactive tools to give a 360-degree perspective of the threats to the planet.

- **AI-Powered Threat Scanner:** At the heart of the app is a real-time prediction engine. With the input of astronomical variables, the user gets an instant "Hazardous" or "Safe" designation with a probability percentage.

- **Real-World Scale Comparison:** This feature connects science to reality. It dynamically compares the computed diameter of the asteroid with famous landmarks such as the Eiffel Tower, Burj Khalifa, giving an instant physical size perspective.

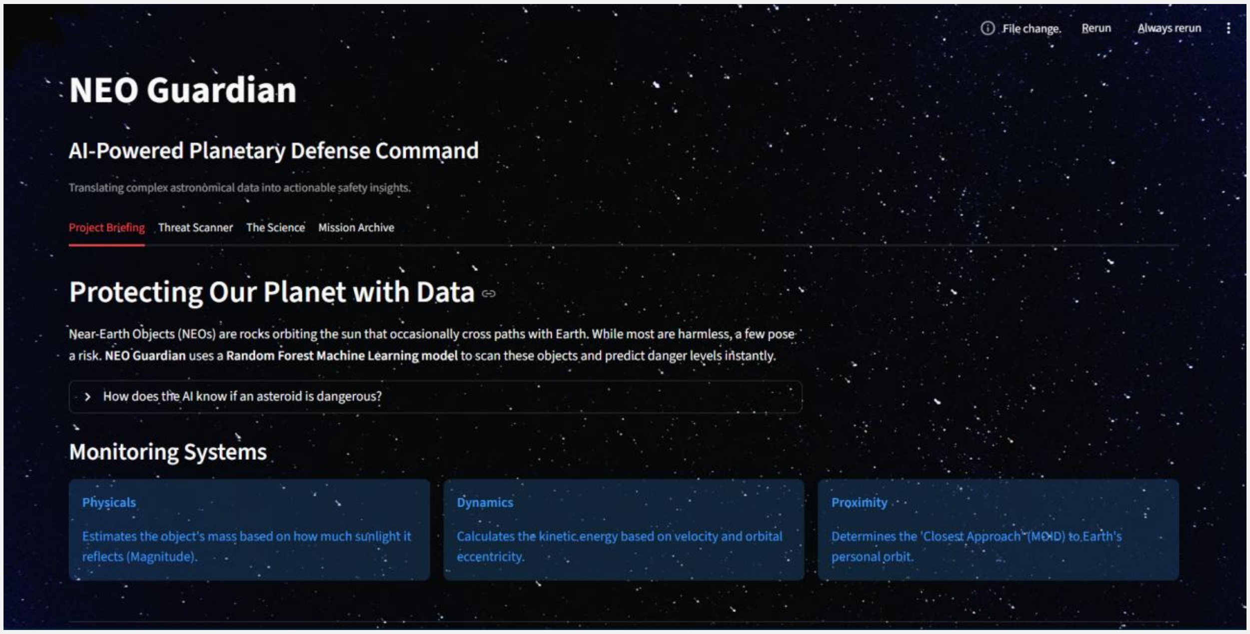
- **Visual Orbital Geometry:** With the object's eccentricity, the app creates a geometric representation of the asteroid's orbit. This enables the user to see how close the orbit path comes to Earth's orbital plane.

- **Impact Consequence Estimator:** With the object's mass and speed, the system computes a "Kinetic Class" (City-level threat vs. Small Airburst) and estimates a possible blast radius in kilometers.

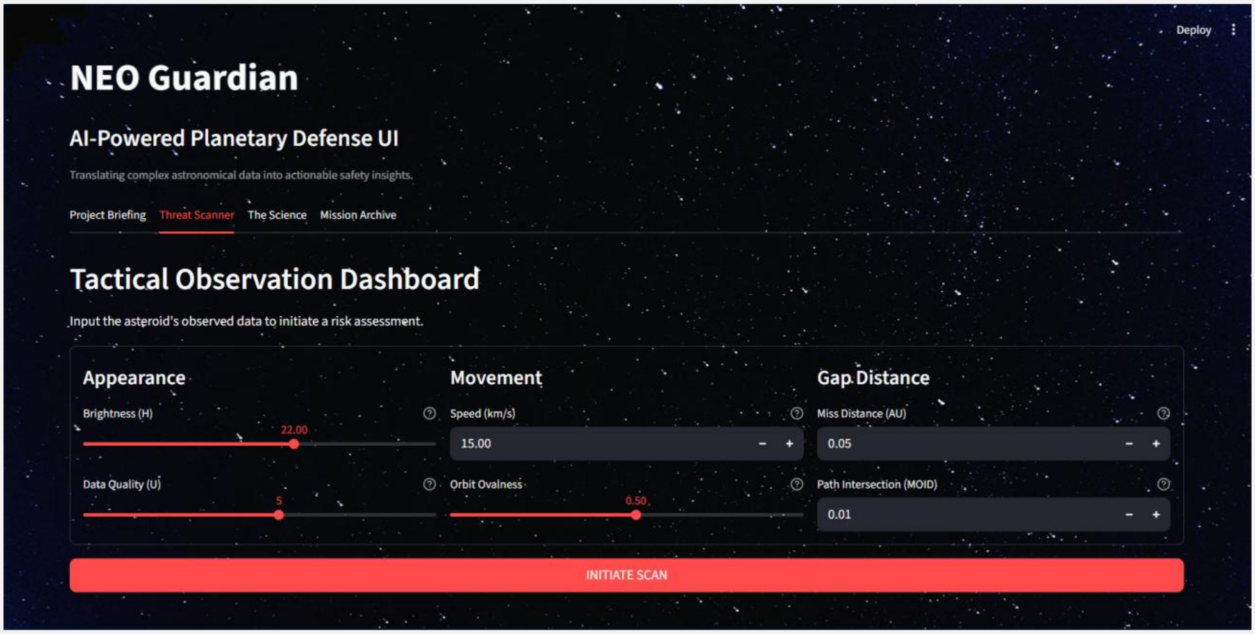
- **Mission Log & History:** A session-based tracking system that records every scanned object. This enables researchers to compare multiple threats simultaneously within a single session.

Screenshots

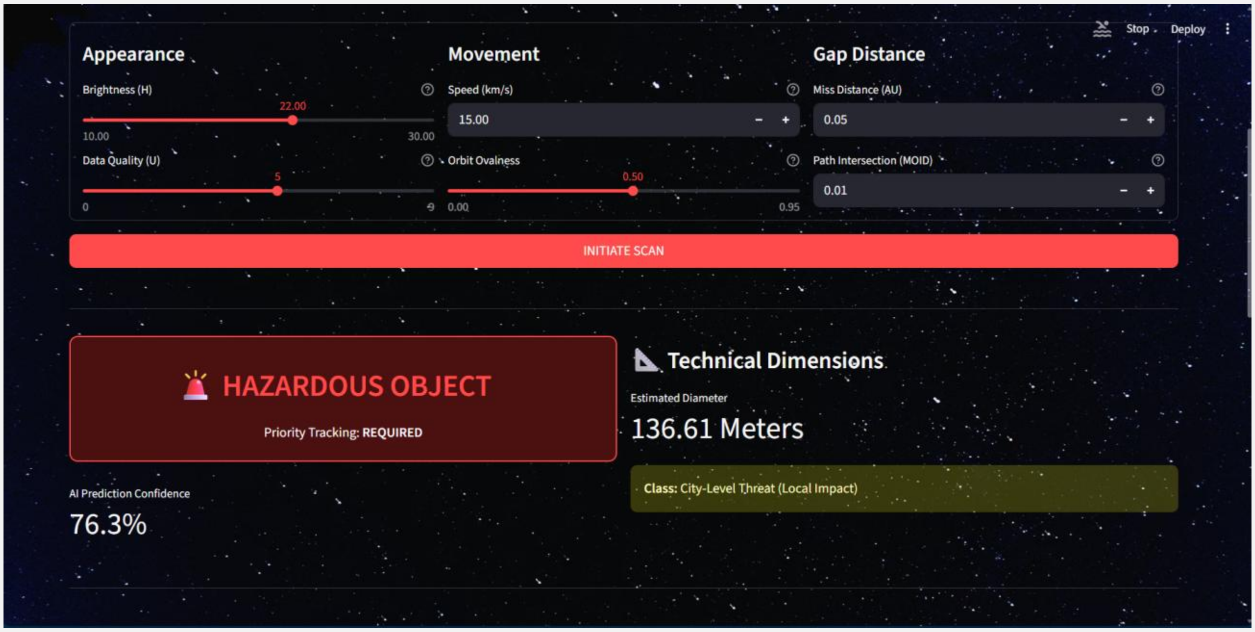
1. **Planetary defense interface(Project Briefing)**

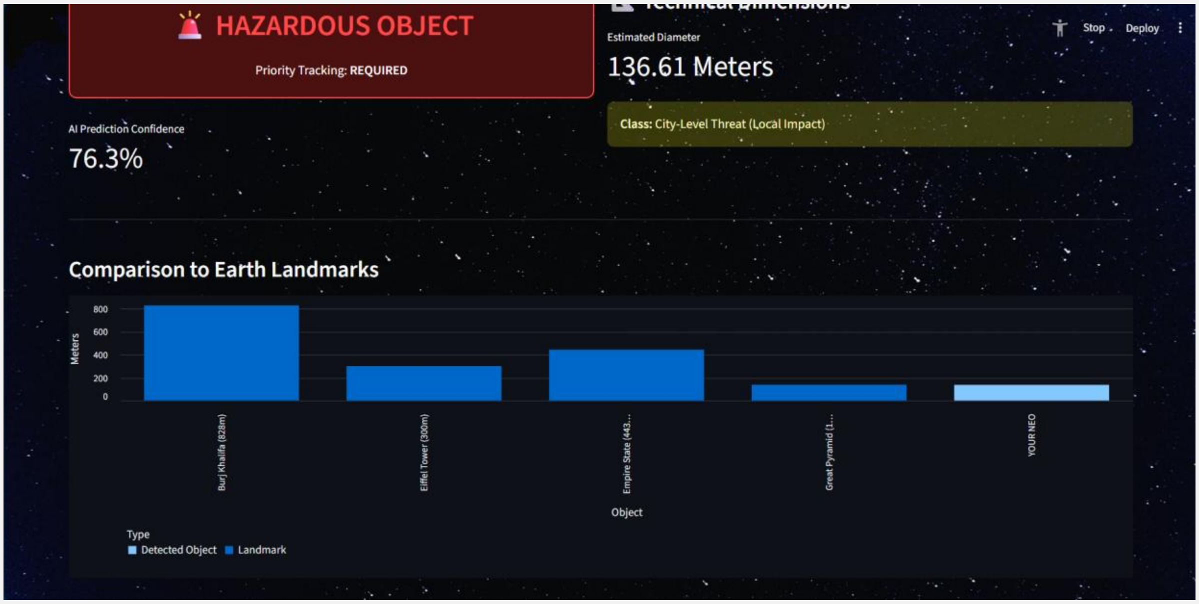


2. Threat Scanner (Tactical Observation Dashboard)

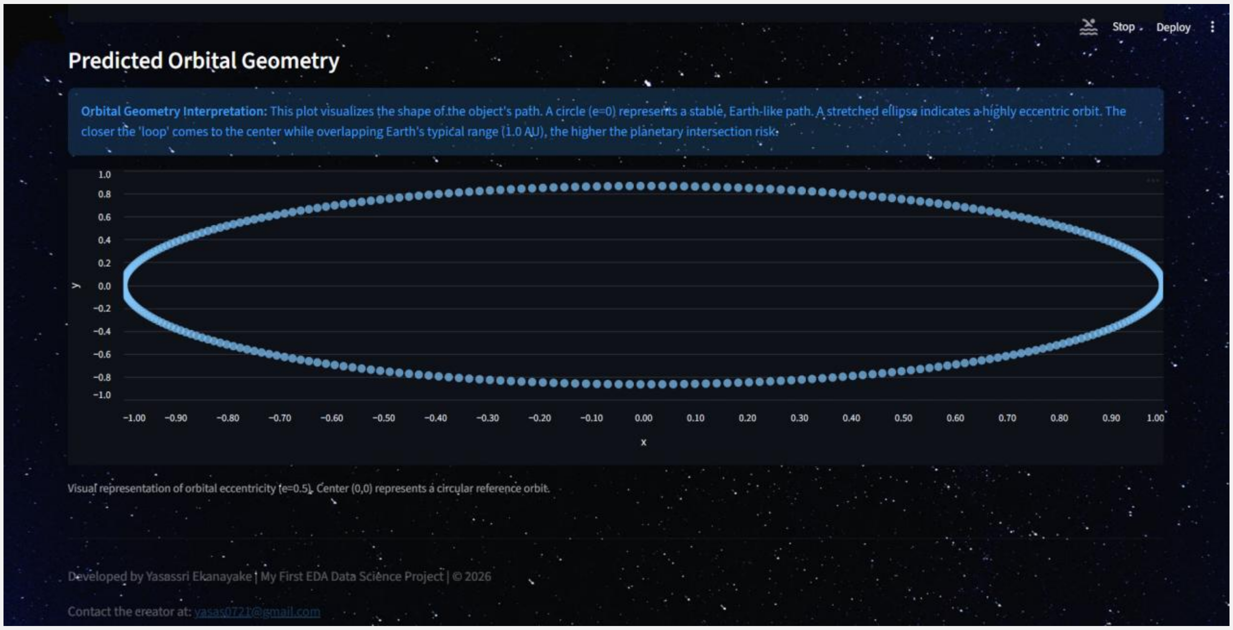



3. Threat Scanner (Tactical Observation Dashboard)-(After running)

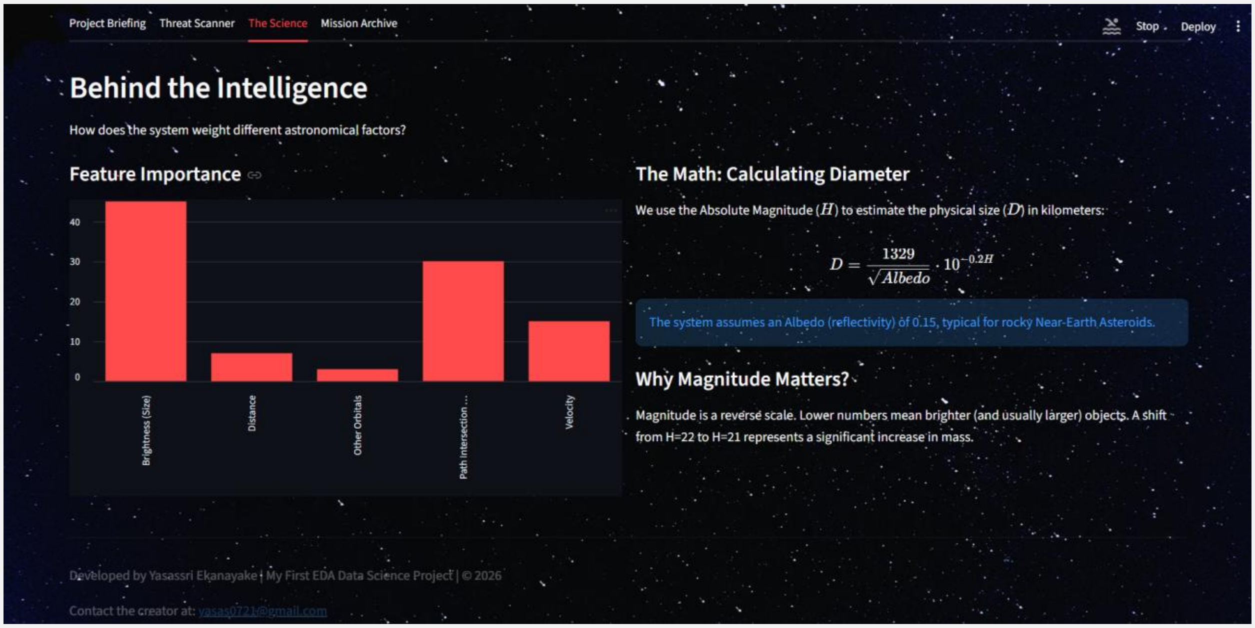




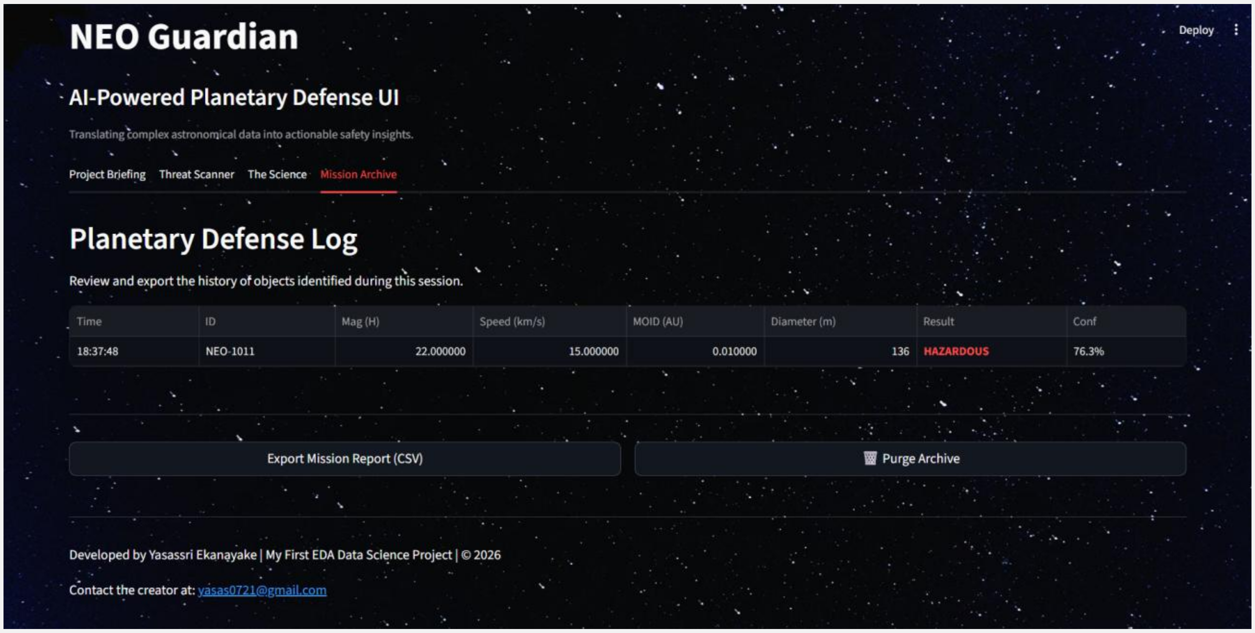

4. Model Analysis



5. The Science section (Explain how the model logic works)




6. Misson Archive Log (History)



# 4. Test cases to check the model

## 1. Test Case 1 :-

**The "Dino-Killer" (Extreme Hazard)**

This tests if the model recognizes a large, fast object with a path that crosses ours perfectly.

- **Brightness /Absolute Magnitude (H):** 15.0 (This will result in a massive diameter, ~3-5 km)
- **Speed:** 35.0 km/s (Very high kinetic energy)
- **Miss Distance:** 0.01 AU (Extremely close)
- **Path Intersection (MOID):** 0.002 (The orbits almost touch)
- **Eccentricity:** 0.8 (A dangerous, elongated orbit)

**Expected Result :-** **HAZARDOUS**. The UI should show a "Planet-Killer" class and the bar chart should dwarf the Burj Khalifa.

## 2. Test Case 2:-

**The "Shooting Star" (Small & Safe)**

**This tests the diameter math. A very faint object should be small enough to burn up in the atmosphere.**

- **Brightness /Absolute Magnitude(H) :** 28.0 (This should result in a tiny diameter, ~10-20 meters)
- **Speed:** 12.0 km/s (Relatively slow)
- **Miss Distance:** 1.5 AU (Very far away)
- **Path Intersection (MOID):** 0.5 (Orbits are far apart)

**Expected Result :-** **SAFE.** The UI should classify it as a "Meteoroid" and state it would likely burn up in the atmosphere.

## 3. Test Case 3 :-

**The "Near Miss" (Large but Distant)**

This tests if your AI understands that **Size** isn't everything—if it's not hitting us, it's not hazardous.

- **Brightness /Absolute Magnitude(H) :** 18.0 (Large stadium-sized rock)
- **Speed:** 20.0 km/s
- **Miss Distance:** 3.5 AU (Deep space, nowhere near Earth)
- **Path Intersection (MOID):** 2.1 (The paths never cross)

**Expected Result :-** **SAFE**. Even though the object is big, the distance makes it a "Nominal" threat.

## 4. Test Case 4 :-

**The "Uncertain Threat" (The Edge Case)**

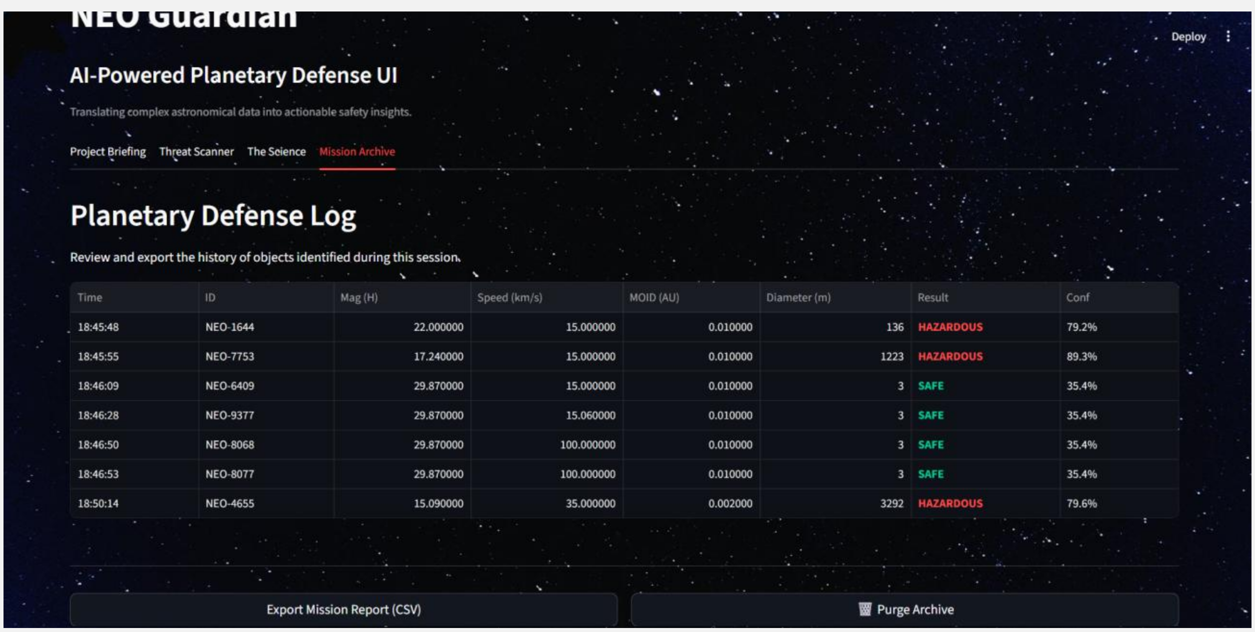
This tests how the **Uncertainty (U)** parameter and **Confidence** score behave.

- **Brightness /Absolute Magnitude(H) :** 21.5 (Right on the 140m "Hazardous" threshold)
- **Speed:** 18.0 km/s
- **Miss Distance:** 0.045 AU (Right on the 0.05 AU "Close Call" threshold)
- **Data Quality (U):** 9 (Maximum uncertainty)

**Expected Result: Varies.** We can see a lower **AI Confidence Score** (closer to 50-60%) because the data is "blurry" and the parameters are right on the edge of danger.

# 4. Tech Stack

| Component | Technology | Role in Project |
|---|---|---|
| Language | Python 3.9+ | The backbone of all logic, math and AI integration |
| UI Framework | Streamlit | Transform python scripts into an interactive,web- based UI |
| Machine learning | Scikit-learn | Used to build,train, and execute the Random Forest model. |
| Data Analysis | Pandas & numpy | Handles data structures and array math |
| Serialization | Joblib | Allows to pre trained .pk1 model to be loaded instantly |
| Version control | GitHub | Manages code updates and serves as the source for deployment. |
| Hosting | Streamlit Cloud | Hosts the live application for global access via a URL.<br>(still working on this) |



# 5. Why Neo Guardian is Important

Planetary defense is no longer a science fiction concept; rather, it has become a vital area of modern-day astronomical research.

Neo Guardian seeks to fill three major gaps that exist today in aerospace research:

**a. Data Accessibility and Translation :-**
NASA and other government agencies have been releasing large volumes of open-source data. However, this data comes in a form that only an astronomer would understand *(H, MOID, AU).* Neo Guardian will act as a translator that will interpret this data into a clear visual form that a government official, a student, or a citizen would understand.

b. **Advanced Predictive Risk Assessment** :-
The traditional method of tracking objects only involves "Distance" as a potential threat. However, with Neo Guardian's use of Machine Learning (Random Forest), it will analyze relationships between many factors instead of just one. By analyzing how "Velocity," "Magnitude," and "Orbital Intersection" relate to each other, it will give a better probability assessment than traditional methods.

**5.3 Human-Scale Context (The "Relatability" Factor) :-**

The human mind cannot readily comprehend the danger potential of a number like "500 meters" or "0.05 AU" without additional context. The project has solved this problem through the implementation of the Scale Comparison Engine. The ability to understand that the discovered asteroid is "The size of two Eiffel Towers" provides a human mind with instant understanding of the physical potential of the object, which is critical to understanding the potential damage and the ability to communicate the information in a clear and effective manner.

# 6. Conclusion and Future changes

Neo Guardian is a significant milestone in my journey as a Data Science student, as I have successfully married the complexity of Orbital Mechanics with the predictive capability of Machine Learning, not only detecting potential threats but also making the enormity of space accessible and understandable to everyone.

As I move forward, I am excited about the prospect of taking this project further in the following ways:

- **Integration of Live APIs:** Moving from static data sets to live NASA NeoWS API Calls.

- **Advanced Visualizations**: Using 3D Orbital Simulations to give users a more immersive experience.

- **Refined Model:** Using alternative algorithms like Gradient Boosting (XGBoost) and comparing their accuracy against my current Random Forest Model.

This is merely the beginning of my journey into the fascinating space of Data Science and the cosmos, and I am excited about taking the feedback from this project and creating even more powerful data-driven solutions in the future.

***Thankyou And have a nice day!***