📌Project Title:

Predicting Hazardous NEOs (Nearest Earth Objects)

📌Project Description:

In this project, you will work with a real-world dataset that tracks Nearest Earth Objects (NEOs) observed by NASA from 1910 to 2024. The dataset contains 338,199 records, each representing an object in space that has been monitored for its proximity to Earth. Some of these objects are classified by NASA as "is_hazardous," indicating that they pose a potential danger to our planet.

Your task is to train a machine learning model that can accurately predict whether a NEO is hazardous or not. This is a critical task, as the ability to accurately identify dangerous objects could be vital for planetary defense.

[Dataset Link]

📌Project Requirements:

1. **Data Importing and Cleaning:**
    - Begin by importing the dataset and addressing any missing values. Ensuring the data is clean and reliable is essential for the success of your model.
2. **Exploratory Data Analysis (EDA):**
    - Explore the data thoroughly using EDA techniques. Use visualization libraries such as Matplotlib, Seaborn, and Plotly to create informative graphs that reveal patterns, trends, and potential correlations within the dataset.
3. **Data Preprocessing:**
    - Prepare the data for modeling by performing necessary preprocessing steps. This includes feature selection, encoding categorical variables, and normalizing or scaling numerical features where needed.
    - Handling Imbalanced Classes: The "is_hazardous" target column is likely imbalanced, with far fewer hazardous objects compared to non-hazardous ones. To address this, consider techniques such as:
        - Oversampling (e.g., SMOTE: Synthetic Minority Over-sampling Technique)
        - Undersampling
        - Class Weight Adjustment
        - Use of Balanced Accuracy as a Metric to ensure the model performs well on both classes.
4. **Model Training and Evaluation:**
    - Train one or more machine learning models using the preprocessed data. Evaluate the performance of your models using appropriate metrics such as Precision, Recall, F1-Score, and AUC-ROC Curve. Make sure to select the best-performing model based on these metrics.
5. **Project Documentation and Submission:**
    - Once the model is trained and evaluated, upload the project to your GitHub profile. Create a well-documented README file that details your process, findings, and insights. Include a link to the repository in your submission.

📌Submission Details:

1) **Submission Format:** You will submit your project using a GitHub repository link. The repository should contain the Jupyter Notebook with all your work, including data cleaning, EDA, preprocessing, model training, and evaluation. Ensure that the notebook is well-documented and presented in a clear, readable format.
2) **README File:** Your repository must include a comprehensive README file. This file should clearly demonstrate your work, summarizing your approach, key findings, and insights. Use the presentation skills we discussed in the previous session to ensure your README is professional, engaging, and informative.
3) **Deadline:** The deadline for submission is *Saturday, 7th September at 11:59 PM*. Please submit your GitHub repository link through the classroom platform by this time.