Nov 8th–10th 2024 NATIONAL STUDENTS' SPACE CHALLENGE

DATA ANALYTICS NSSC'24

Cosmic Collision- Analysing Asteroid Risks with Data

OVERVIEW

Asteroids are small, rocky objects that orbit the Sun, primarily found in the region between Mars and Jupiter known as the asteroid belt. They are remnants from the early solar system, formed over 4.6 billion years ago, and are considered planetesimals, or building blocks of planets that never coalesced into a larger body due to gravitational disturbances, particularly from Jupiter.

Asteroid impacts have been a constant force shaping Earth's history. From minor events that leave barely a trace to catastrophic collisions that have caused mass extinctions, these cosmic encounters have played a significant role in our planet's evolution. Given the potential for devastating consequences, scientists worldwide are dedicated to detecting and tracking the asteroid impact threat. To mitigate this risk, it is crucial to identify and analyse asteroids that could pose a hazard.

Objective of the Analysis: Data-Driven Classification of Hazardous Asteroids

The primary objective of this analysis is to use data analytics and machine learning techniques to determine the likelihood of an asteroid being hazardous to Earth based on various features provided in the dataset. This includes examining characteristics such as the asteroid's size, orbital parameters, velocity, and proximity to Earth's orbit.

By analysing these features, the goal is to develop predictive models that classify asteroids into **hazardous** and **non-hazardous** categories. This classification is crucial for prioritising which asteroids require further monitoring, potential deflection efforts, or other mitigation strategies.

We will also identify asteroids exhibiting unusual or anomalous behaviour that may warrant closer attention.

GENERAL INSTRUCTIONS

1. The dataset for the problem can be found here: Dataset
2. The description of the features used is given here: Description of the Features
3. The problem statement consists of 5 parts, and each part has a few subproblems.
4. These problems are based on the Solar System Dynamics data captured by NASA.
5. Students can participate in teams of 3-4 members.
6. The weightage of each question is mentioned next to the question.
7. Participants are free to use any programming language, environment, and library. However, it is preferable to use Google Colab or Jupyter Notebook as the programming environments.
8. For submission, participants should submit the .ipynb file of the solution. The code should be well-commented to clearly convey the work done. Each team should also submit a PDF report containing a detailed solution and approach. All plots and outputs must be shown in the report with proper explanations and descriptions. Final answers should be highlighted.
9. All code snippets should be attached to the report.
10. Only 15 minutes will be allocated for each team to present their solutions. Therefore, all relevant conclusions must be presented promptly.
11. Participants will be evaluated on the clarity and applicability of the solution, their innovation, and the feasibility of their conclusions.
12. **Important: Ensure that each `.ipynb` code block is clearly associated with the corresponding question and sub-question. Use markdown to label each block, so it is easy to identify which problem and subproblem the block is solving. Failure to do so may lead to disqualification.**

Note: *The problem statement below is worth a total of* ***100 MARKS****. Your score will be scaled to* ***60 MARKS****, with the remaining* ***40 MARKS*** *allocated for the presentation.*

PROBLEM STATEMENT

1. Exploratory Data Analysis (EDA) (20 points)

1.1 Data Inspection (7 points):

- Inspect the dataset and determine the data types of all features (numerical, categorical). **(1 MARK)**
- Calculate and analyse basic statistics for each numerical feature, including range, mean, median, standard deviation, and quartiles. **(2 MARKS)**
- Identify features that have missing values. **(1 MARK)**
- Identify the numerical and categorical features of the dataset to use for further analysis. **(1 MARK)**
- Use imputation to fill the null values in the dataset. How is this process different for numerical and categorical columns? **(2 MARKS)**

1.2 Statistical Inference (6 points):

- Plot the distribution of numerical features to assess the skewness of the data. Does this dataset require normalisation? If yes, normalise/scale the dataset. (Hint: Use histograms) **(2 MARKS)**
- Identify potential outliers in the numerical columns using any statistical technique (e.g., box plots, z-score, etc.). **(2 MARKS)**
- Explore the relationship between different features using scatter plots or correlation matrices. (Hint: Use Seaborn or similar libraries) **(2 MARKS)**

1.3 Visualisation (4 points):

- Create a pairplot using Seaborn to visualise relationships between multiple numerical features simultaneously. **(2 MARKS)**
- What do you infer from these plots? How do the diagonal plots and off-diagonal plots in a pairplot differ in the information they provide? **(2 MARKS)**

1.4 Tackling Class Imbalance (3 points):

- Is there a classification bias (class imbalance) in this dataset? If yes, how would you tackle it? **(2 MARKS)**
- Discuss the implication of class imbalance on model performance. **(1 MARK)**

2. Numerical Interpretation and Mathematical Analysis (20 points)

2.1 Feature Engineering (15 points):

- Combine the approach_date, month, and year features into a single feature representing the day of the year. Convert it into a ‘datetime’ format. **(1 MARK)**
- Calculate the ratio of Miss Distance vs. Semi-major axis. Create a 'Time Until Approach' feature based on the difference between the 'Epoch Date Close Approach' and the current date. **(3 MARKS)**
- Calculate the eccentricity of the orbit, average orbital velocity, and orbital period using Kepler’s Law. **(3 MARKS)**
- Calculate the heliocentric distance, escape velocity, and specific orbital energy. **(3 MARKS)**
- Calculate the Specific Angular Momentum using the formula: $h=\mathrm{sqrt}(GMa(1–e^2))$. **(1 MARK)**
- Calculate the velocity at Perihelion and Aphelion. **(1 MARK)**
- Average the Miss distance of various categories and find the closest approach distance. **(1 MARK)**
- Calculate Synodic Period and Mean Motion using the orbital period. **(2 MARKS)**

2.2 Additional Features (5 points):

- Create additional features as per your understanding of the problem for improving accuracy. More marks are awarded for innovative and effective features. **(5 MARKS)**

3. Handling Binned Values (5 points):

Note: *Binned features are categorical variables where values are grouped into discrete categories such as: [very slow, slow, fast, very fast, etc.].*

- Modify the binned features that have an ordinal relationship in this manner:

  (very slow = 0, slow = 1, fast = 2, very fast = 3, etc). **(2 MARKS)**

- One-hot encode the binned features whose relationship is not strictly ordinal. **(3 MARKS)**

4. Hazardous Classification (35 points):

- Build a robust and efficient classifier to classify asteroids as Hazardous (1) or Not Hazardous (0). **(7 MARKS)**
- Implement K-Fold Cross Validation for training. Train the dataset for all values of K from 2 to 10. Plot the loss and accuracy versus epochs for these K values. **(12 MARKS)**
- Optimise all the hyperparameters used in the classifier by selecting an appropriate optimisation method. **(8 MARKS)**
- Plot the ROC curve and Confusion Matrix to quantify the performance of your classifier. **(4 MARKS)**
- Use SHAP Values, Permutation Importance, or Partial Dependence Plots to list the most and least useful features. **(4 MARKS)**

5. Anomaly Detection (20 points):

- Perform anomaly detection using:
  - (i) Any inbuilt library of your choice. **(4 MARKS)**
  - (ii) Writing your own anomaly detection algorithm. **(12 MARKS)**
- Store the results as a new column in the dataset. Print the number of anomalies detected by each method. **(2 MARKS)**
- Compare the results from both methods by plotting a Confusion Matrix. Print the number of examples flagged by both algorithms. **(2 MARKS)**