Data Set Info :

- The data set(Asteroids Classification)is taken from Kaggle.

- This is the Kaggle link where i got my data set by Shruti mehta-
https://www.kaggle.com/datasets/shrutimehta/nasa-asteroids-classification/data

- The actual data is collected from the (http://neo.jpl.nasa.gov/).

Asteroids are small, rocky bodies that orbit the sun which are primarily found in the asteroid belt between Mars and Jupiter.Some asteroids are classified as near-Earth asteroids(NEO'S) if their orbits bring them close to the Earth's orbit.

As a data scientist at NASA I am responsible for analyzing the dataset of near Earth objects (NEOs) to improve our comprehension of these entities and strengthen our defense strategies for planets. This dataset contains details about NEOs, such as their features like absolute brightness estimated size orbital traits and other relevant parameters.With the growing number of detected asteroids, there's an increasing need to classify these objects efficiently to prioritize monitoring and potential mitigation efforts.

Experiment Strategy:

- Developing a classification model to accurately predict whether an asteroid is potentially hazardous based on the given features. This classification will help NASA focus on the most threatening NEOs, ensuring efficient use of resources and timely action.

- Identifing which asteroids near Earth are dangerous to prioritize monitoring and actions.

Steps:

Data Exploration and Cleaning:

Understanding features like size, speed, and orbit.

Cleaning the data and handle any missing values.

Model Selection:

K-Nearest Neighbors (KNN): Simple and effective for classification.

Decision Tree: Easy to interpret and handles complex relationships.

Training and Evaluation:

Train both models using a training dataset.

Use cross-validation to test the models.

Evaluate using accuracy, precision, recall, F1 score, and ROC-AUC, focusing on recall to avoid missing dangerous asteroids.

Model Optimization:

Fine-tune model parameters using grid search or random search.

Combine models using ensemble methods if needed.

Deployment:

Implement the best-performing model into NASA's system.

Continuously monitor and update the model with new data.

Expected Benefits:

Better detection of dangerous asteroids.

Enhanced preparedness for potential asteroid impacts.

Conclusion:

By using KNN and Decision Tree models, NASA can improve its ability to identify and respond to asteroid threats, ensuring Earth's safety and advancing space monitoring technology.

I have explained all the features of my data set below:

Orbital Parameters—Columns, e.g., Eccentricity, Semi Major Axis, Inclination, Asc Node Longitude, Orbital Period, Perihelion Distance, Aphelion Dist, Perihelion Arg: These provide much detail about the orbital dynamics of each NEO. Clustering on these parameters will help in picking out groups of asteroids that share similar orbits, possibly suggesting a common origin or similar evolutionary paths. Such a clustering structure could reveal asteroid families.

Absolute Magnitude and Estimated Size—Absolute magnitude (related to brightness) and estimated size parameters, such as Est Dia in KM, M, Miles, and Feet: Such parameters allow inferences to be drawn concerning the composition and structure of NEOs. Different materials will reflect sunlight differently, affecting brightness and apparent size estimates. Clustering NEOs on these characteristics can give inferences about material composition, probably linking it with some families or origins of specific asteroids.

Minimum Orbit Intersection and Jupiter Tisserand Invariant: These parameters are necessary for the understanding of how NEOs interact with other solar system bodies—first of all, how they are affected by Jupiter, which has quite strong forces. Grouping NEOs by these attributes can enable guessing the groups more prone to such gravitational interaction and hence drawing insight into dynamic processes forming NEO orbits.

Miss Distance (Astronomical, Lunar, Kilometers, Miles) These are metrics of NEOs showing how close they come to the earth and, therefore, a means of prioritizing NEOs for further study in the face of threat or accessibility for missions. Clustering NEOs using their approach distance to Earth could help in the identification of clusters more relevant for defense or exploration missions.

Uncertainty of Orbit and Date Determination: These two features represent the confidence in orbital data and the most recent update of it. Clustering along these features will identify NEOs for which the observation is more urgent to update and refine the prediction of their trajectory, an aspect of crucial importance to assess a threat and plan possible mitigation.

Epoch Osculation, Mean Anomaly and Mean Motion: These orbital elements give further details about the NEOs' position and movement in their orbits at a given time; they are, therefore, useful in the precise clustering and analysis of temporal patterns in NEO behavior.

The dataset points, as provided in the richness of Task1 , are used to apply the clustering analysis in an effective way to the requirements of Task 1. This will further enhance the understanding of NEO characteristics, prediction, and mitigation of the potential threat and mission planning for improvement in knowledge and defense strategies against NEOs.

EDA(Exploratory Data Analysis) and Preprocessing

- Exploratory Data Analysis is a data exploaration technique to understand the various aspects of the data.

- Aim : Understanding and exploring the data , make sure the data is clean ,do not have any missing values or even null values in the data set, knowing the important variables in the data set and removing null values that may actually hinder the accuracy of conclusions when we work on model building.

- EDA helps us to identify the faulty points in the data

- EDA helps us to understand the relationship between the variables which gives us the wider perspective on the data.

- It includes several techniques in a sequence that we have to follow.

- steps :

1. Data cleaning - to get rid of the redundencies variables which means removing unwanted columns because it may overfit or underfit the model

2. Analysis of relationship between the variables.

- Here , I have used a Heatmap which represents the correlation values between the variables from my data set.

- I have used heatmap because it uses a color scale to represent correlation coefficients which makes us easy to understand the data.

- Each cell represents the correlation coefficient between two variables. The correlation coefficient ranges from -1 to 1.

- The values which are close to 1 or 1indicate a strong positive correlation between the variables.

- For example:

- There's a strong positive correlation between 'Semi Major Axis' and 'Orbital Period'. This suggests that as the semi-major axis of an orbit increases, the orbital period also increases proportionally.

- There's a moderate positive correlation (0.724) between 'Orbit ID' and 'Est Dia in KM(min)'. This suggests that certain orbits may have asteroids of relatively consistent sizes.

- The values which are close to -1 indicate a strong negative correlation between the variables.

- For example:

- There's a strong negative correlation (-0.930) between 'Jupiter Tisserand Invariant' and 'Semi Major Axis'. This suggests that as the Jupiter Tisserand Invariant decreases, the semi-major axis tends to increase.

- The values which are close to 0 indicate no significant correlation between the variables.

For example:

There's no significant correlation between 'Neo Reference ID' and other variables since most correlation coefficients are close to 0.

Conclusions and Future Work

I examined the dataset to identify any missing data. Visualized the distributions and relationships using box plots and pie charts.

To establish a baseline I utilized a Dummy Classifier to predict the class (False) achieving an accuracy of, around 83.87%. This choice was made due to the dominance of one class.

Implementing KNN with K=6 led to an accuracy of 86.78% showing improvements in accuracy and precision compared to the baseline but lower recall for the class (True).

Initially the Decision Tree Classifier showed scores with an accuracy of 99.50% hinting at potential overfitting.

To address overfitting I employed grid search and cross validation to adjust hyperparameters like max_depth and ccp_alpha resulting in an enhanced accuracy of 99.64% with optimized parameters (max_depth=4, ccp_alpha=0.000310).

I used nested cross validation to evaluate how well the models generalize.

The Decision Tree Classifier demonstrated an accuracy of 99.62% with variability indicating its robustness.

In assessing model performance I considered metrics such as accuracy, recall, F1 score and ROC AUC where

I depicted the outcomes of the model through confusion matrices and ROC curves to gain insights, into model predictions and performance.

In evaluating usability the chosen methodology involving Decision Trees and KNN, with validation and hyperparameter tuning proved effective for achieving high accuracy in classification tasks. Decision Trees offered simplicity in interpretation and strong performance making them highly practical, for this task. Similarly KNN demonstrated effectiveness in understanding how various configurations impact performance.

- The generated value within the context

1. Accuracy

Formula; Accuracy = (Positives + True Negatives) / (Total Instances)

This metric gauges the correctness of the models predictions.

For instance; Accuracy = (3926 + 743) / (3926 + 6 + 12 + 743) = 0.9878

2. Precision

Formula; Precision = Positives / (True Positives + False Positives)

It reveals the percentage of identifications that were accurate.

For example; Precision = 743 / (743 + 6) = 0.9878

3. Recall (Sensitivity)

Formula; Recall = Positives / (Positives + False Negatives)

This metric assesses the models capability to recognize all instances.

In the calculations the recall rate was determined to be 98.78%. The F1 Score formula, which balances precision and recall yielded a value of, about 98.78% well. Moving on to ROC AUC with a score of 0.9961 it measures how well the model can distinguish between classes.

When it comes to visualizations and interpretations the confusion matrix offers insights into positives true negatives, false positives and false negatives. For instance in one scenario presented in the matrix table;

Actual \ Predicted

Negative; 3926 predicted correctly and 6 incorrectly

Positive; 743 predicted correctly and 12 incorrectly

Additionally the ROC curve illustrates how well the model performs across threshold levels by plotting positive rates against false positive rates. An AUC of 0.9961 indicates a ability.

It's important to note that this analysis primarily focused on K Nearest Neighbors (KNN) and Decision Trees methodologies. Exploring methods such, as Random Forests (bagging and boosting) could potentially enhance performance and robustness.

While we have already used hyperparameter tuning we could consider incorporating strategies such, as regularization to reduce overfitting and improve the models ability to generalize. Exploring feature engineering methods such as Principal Component Analysis (PCA) or domain techniques may prove beneficial in identifying features and potentially enhancing model performance.

Future Plans:

Implementing and evaluating techniques like Random Forests to gauge their impact on performance and resilience.

Employing regularization methods to further combat overfitting. Improve the models generalizability.

Investigating feature engineering approaches, like PCA and domain methods to pinpoint features and potentially enhance model performance.

Exploring an array of algorithms, including Support Vector Machines (SVM) neural networks or other ensemble techniques to potentially achieve performance and deeper insights.