Predicting Hazardous Near-Earth-Objects

Project phase - 3

50461788 - Sonalira
50471077 - Khyathik

Introduction :

Certain objects can enter the Earth's atmosphere due to the gravitational pull of planets. Comets and asteroids are the terms used to refer to these near-Earth objects (NEO).

There have been multiple examples in history where asteroids' devastating impacts have resulted in devastation, destruction, and extinction. Near-Earth objects may incur huge losses if they enter our atmosphere. They might also severely damage our infrastructure and put our lives in jeopardy. We can address these problems thanks to this research.

Problem statement :

The objective of this project is to examine the dataset of near-Earth objects using data-driven analysis. This analysis aids in determining the likelihood of the incoming object, whether it is harmful and enters the earth's atmosphere.

In order to determine whether an asteroid has the potential to cause harm, we plan to use data on its relative velocity, absolute magnitude, and estimated average diameter in this research. We can track, find, and evaluate if an item constitutes a hazard to the earth using the dataset's many fields.

Data Source :

Our dataset, which has up to 40,000 entries, was compiled from a number of sources.

https://api.nasa.gov/

https://www.kaggle.com/datasets/sameepvani/nasa-nearest-earth-objects?resource=download&select=neo.csv

https://www.kaggle.com/code/elnahas/nasa-nearest-earth-objects/notebook

Data Preprocessing :

Data preprocessing has involved a number of processes. Missing values have been examined, and the data has been examined for duplicate values and outliers that could lower prediction accuracy.

One hot encoding, which improves predictions and classification accuracy of a model, is the crucial process of changing the categorical data variables to be fed to machine learning algorithms. Categorical features are frequently preprocessed using One Hot Encoding. The feature of each sample that corresponds to its original category is given a value of 1 by this type of encoding, which generates a new binary feature for each potential category.

Other techniques, such as undesirable outlier detection, scaling, and date formatting, have been used.

EDA and Feature selection :

The dataset was subjected to univariate and bivariate analysis, and numerous plots were created to aid in the interpretation of the data.
In order to understand the data, we have undertaken feature engineering, which entails splitting our data into train and test data sets.

Machine learning models :

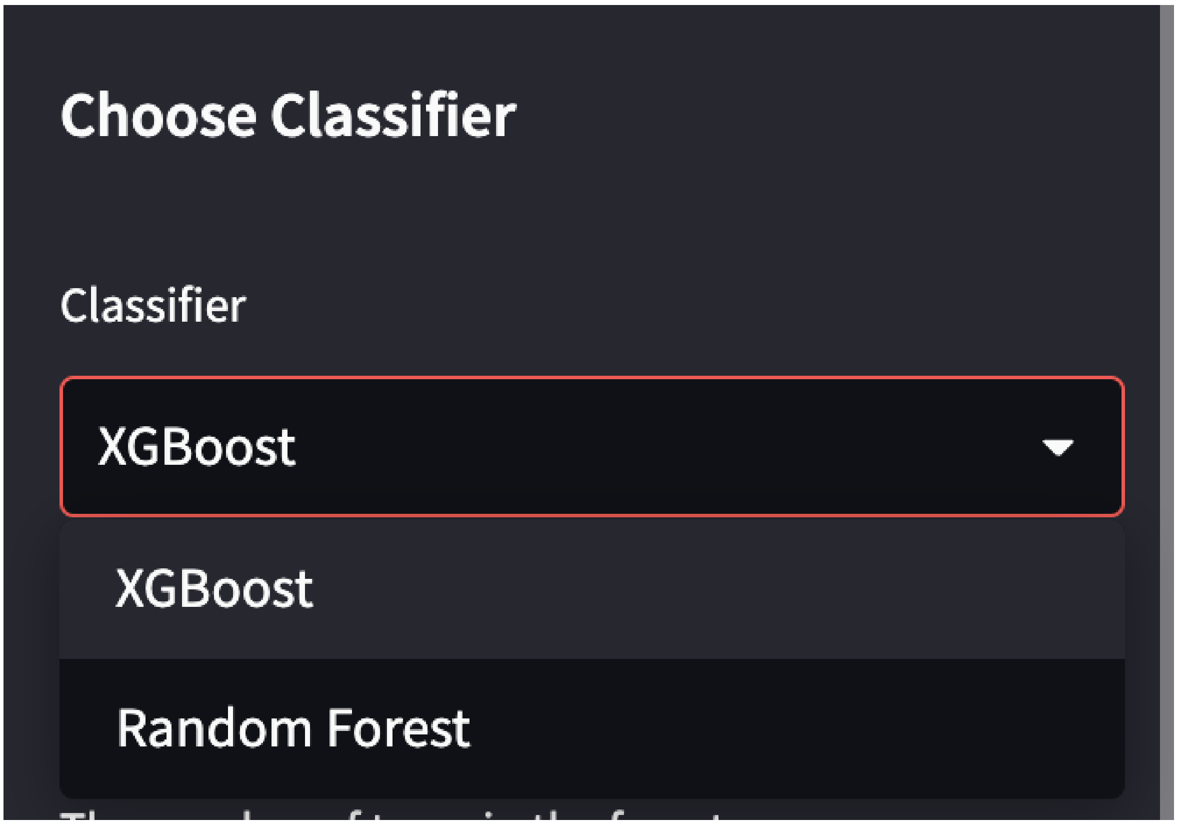
To determine if the asteroid dataset is dangerous or not, various algorithms were used, such as logistic regression, decision trees, random forest classifiers, K-Nearest Neighbours, and XGBoost model. To make precise forecasts, numerical features like est diameter min, est diameter max, absolute magnitude, and relative velocity were used.

Near Earth Object Prediction App :

Here we used streamlit to create app and create models that helps us to predict the required constraint.

Efficient models :

After thoroughly processing the metrics like accuracy, precision, recall, confusion matrix and roc curve from Phase-2 , we observe good results using xgboost and random forest even if we use large datasets. Both XGBoost and random forest prediction based on number of trees but it’s the processing of identifying trees thats makes the difference.
The user of our app has the option to select a classifier from the list of models above. The feature's GUI is shown below.



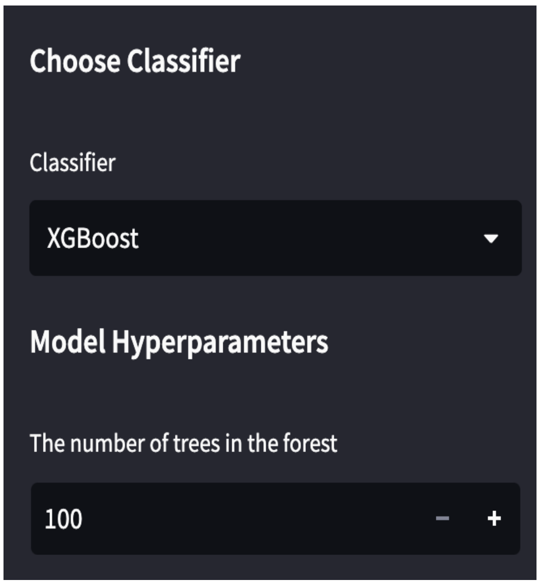
XGBoost :

In XGBoost Decision trees are built sequentially correcting the errors over previous trees. XGBoost implements parallel processing at the node level which makes it efficient in many ways.

Why XGBoost ?

XGBoost can be used to boost performance and speed up the model's overall execution because it can tolerate missing values and unscaled input. In general, it's this algorithm's effectiveness, accuracy, and viability and its ability to perform parallel processing on a single machine is what makes it quick. Additionally, it offers tools for detecting significant variables and doing cross-validation.
Our aim is to process numeric values better, XGBoost handles numeric data very well.

Hyperparameters used to maximize the models predictive accuracy. The user can alter values of hyper parameters with the help of slider given in UI and view results for chosen hyper parameters. And click on the "Classify" button to view the results.

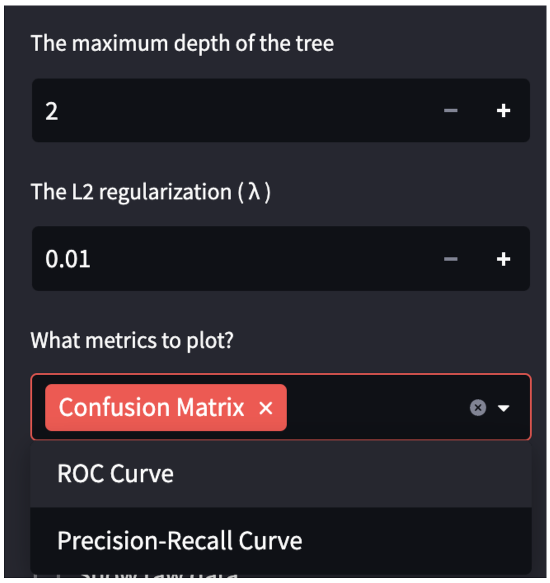


Choose Classifier

Classifier: XGBoost

Model Hyperparameters

The number of trees in the forest: 100



The maximum depth of the tree: 2

The L2 regularization (λ): 0.01

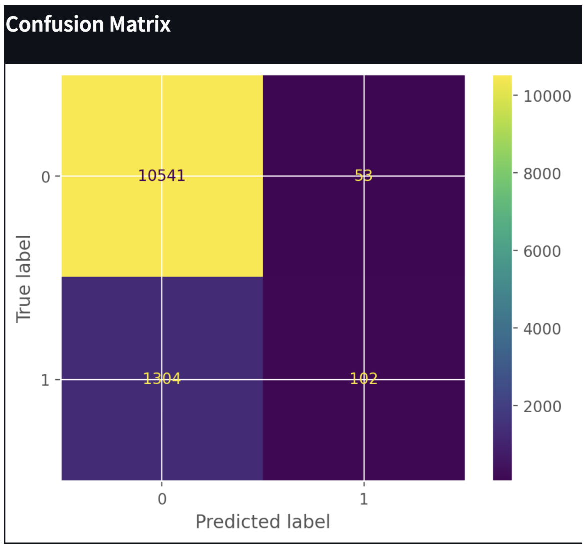
What metrics to plot?: Confusion Matrix

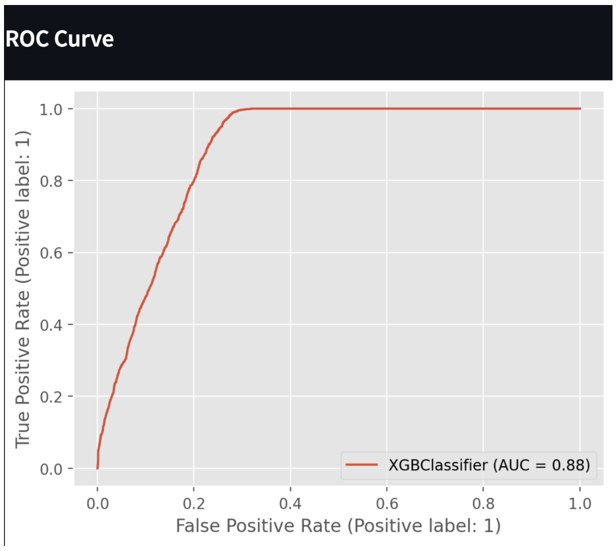
ROC Curve

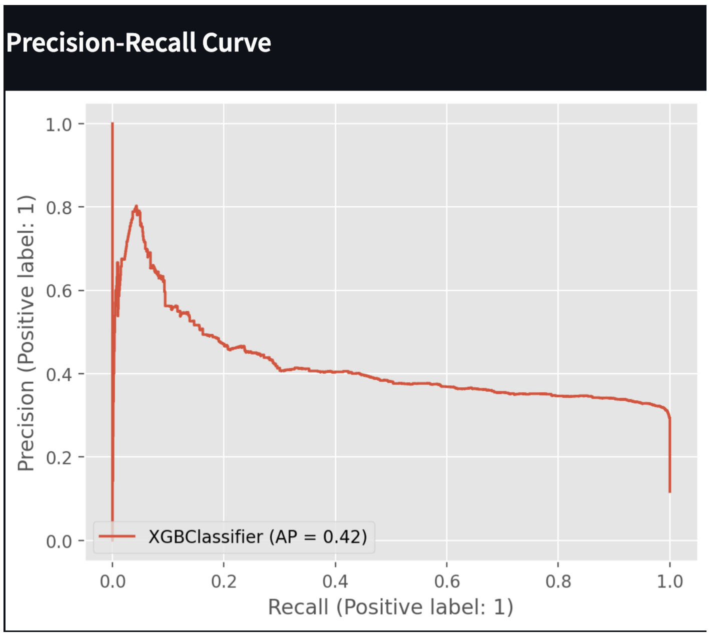
Precision-Recall Curve

Hyperparameters like number of trees, max depth of tree and L2 regularization were used to get optimized results.

Additionally, we used metrics like the confusion matrix, ROC curve, and precision-recall curve to depict the predictions. Here are some illustrations of metrics.





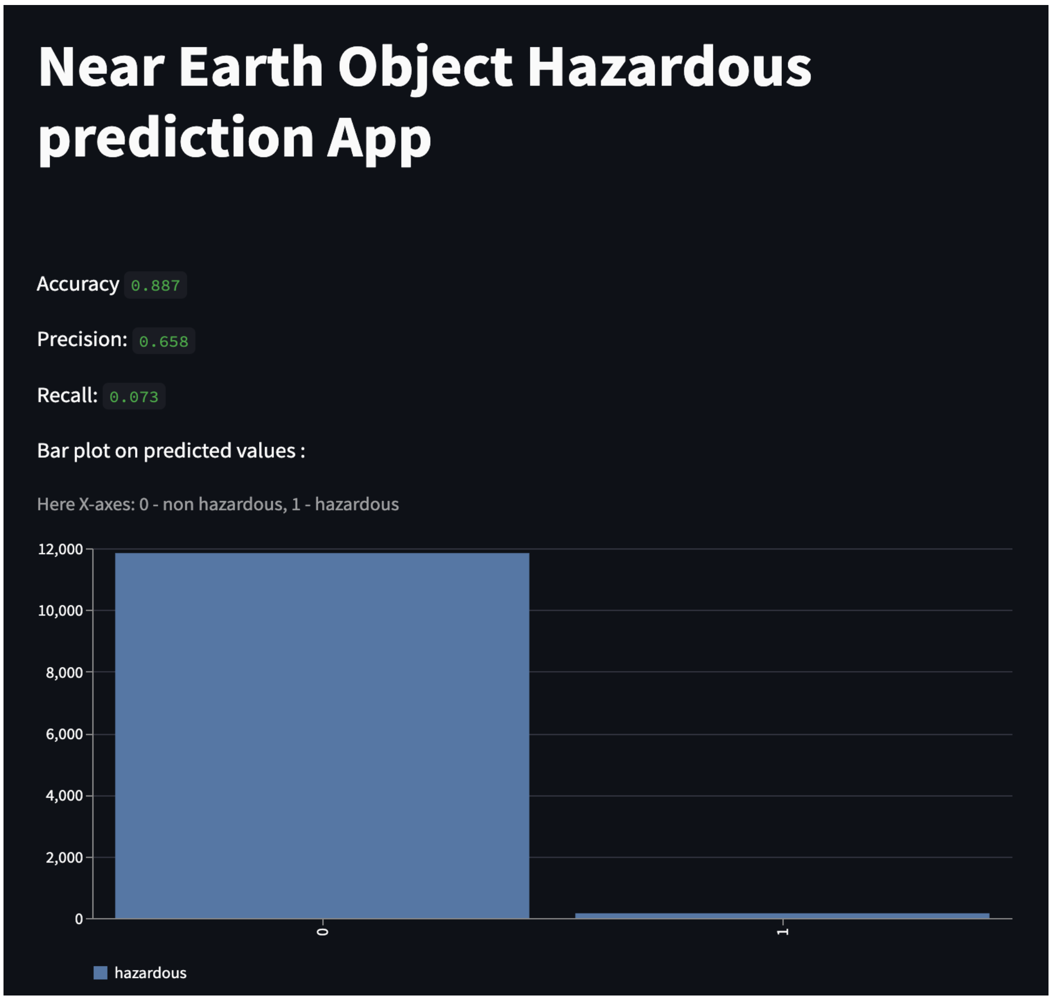


The count of anticipated hazardous and non-hazardous objects according to our model is also displayed in a plot.

Observations:

As the number of trees in decision tree of XGBoost increases accuracy increases from 88% to 89%.

We used L2 Regularisation to prevent overfitting in the data. Having a large lambda with respect to the number of samples will also reduce the gain.



Near Earth Object Hazardous prediction App

Accuracy `0.887`

Precision: `0.658`

Recall: `0.073`

Bar plot on predicted values :

Here X-axes: 0 - non hazardous, 1 - hazardous

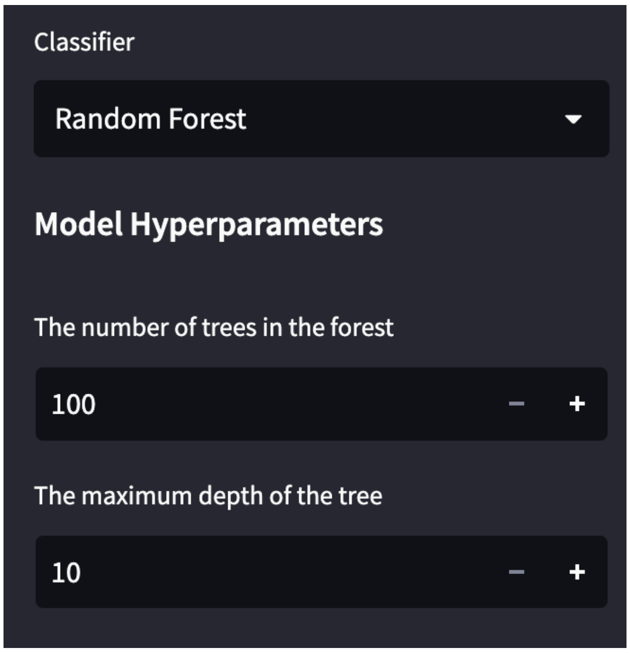
Random forest :

A random forest **mixes multiple decision trees**, whereas a decision tree only combines some decisions. As a result, it is a drawn-out procedure that moves slowly. A random forest is **quick and effective with large data sets,** especially a linear one and this requires extensive training. We used the random forest classifier for this reason.

Why random forest ?

Random forest algorithm **avoids and prevents overfitting** by using multiple trees. The results of numerous decision trees are combined in the random forest algorithm. Some may not provide the precise output that is needed, but when **all the trees are combined**, a final result that is accurate can be used for additional processing. Also, random forest reduces the problem of overfitting with the help of multiple trees.

Hyperparameters include the size of the decision tree forest and the amount of attributes that each tree takes into account when dividing a node. Scikit-Learn implements a set of reasonable default hyperparameters for all models, but there is no guarantee that these are the best ones for a given problem. The optimum hyperparameters are typically impossible to predict in advance, and tweaking a model is when machine learning transforms from a science into trial-and-error based engineering. **Users were given the freedom to conduct this trial-and-error process**.



Classifier

Random Forest

Model Hyperparameters

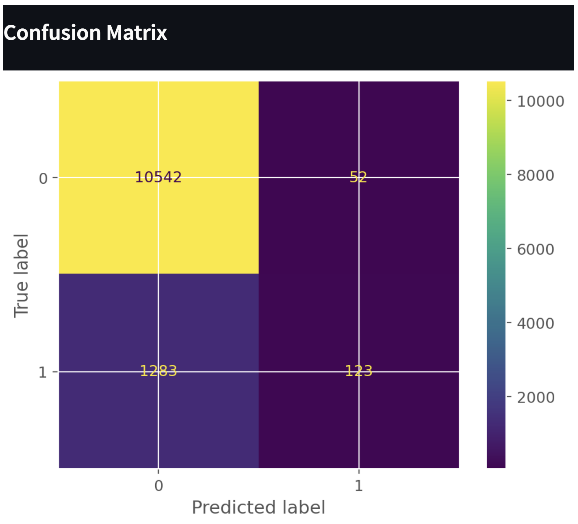
The number of trees in the forest

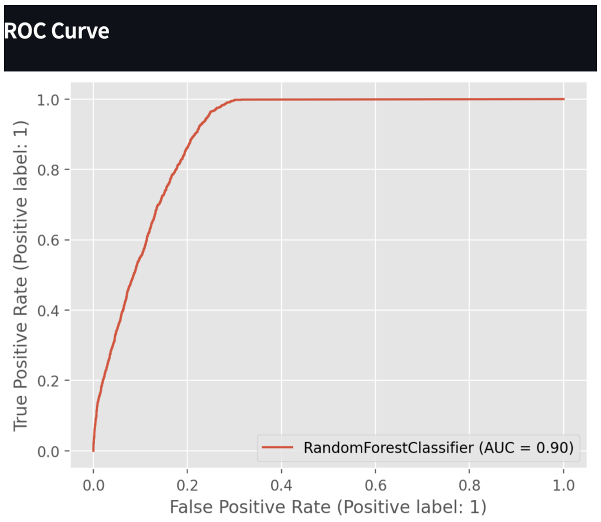
100

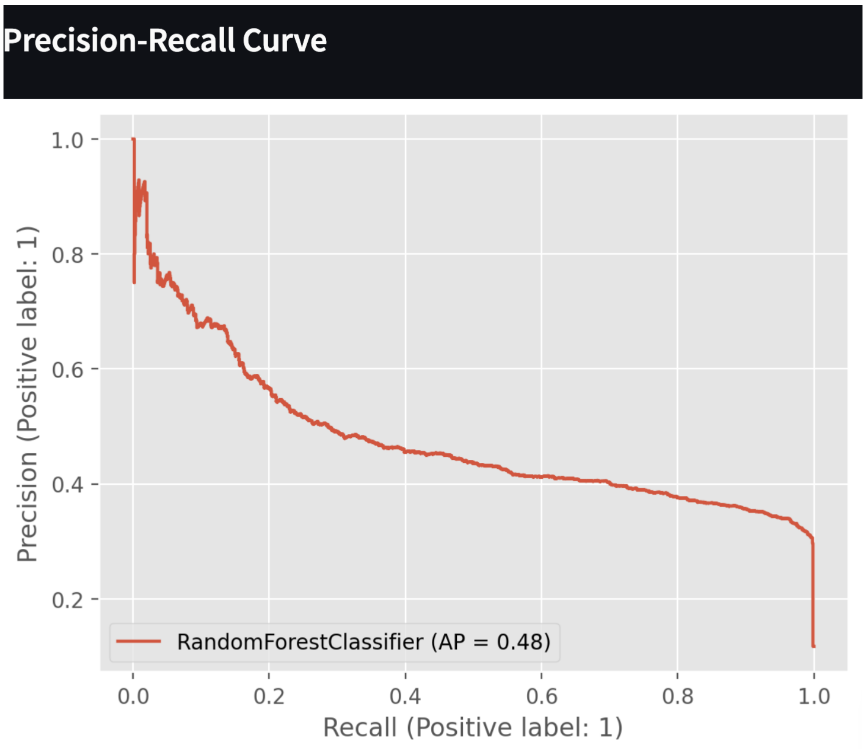
The maximum depth of the tree

10

Below are the visualizations for the random forest model.







Observations from tuning hyperparameters:

- We observe that as the number of trees in a random forest increases the AUC in an ROC curve does not significantly change but AP(Precision) in a precision recall curve increases. Therefore for a large dataset an increased number of trees will improve precision of prediction.

Learnings from product & How does it help :

- A feature that allows users to input their own dataset was added. Using this, users can predict if that dataset contains any hazardous NEOs that create a threat to earth's atmosphere.
- With the help of given model hyper parameters, the user has the privilege to optimise the model and get more accurate results.
- The metrics such as confusion matrix, ROC curve and precision-recall curve helps users to determine if the model is performing well on a given dataset.
- Highest Accuracy of 89% was obtained by using XGBoost. XGBoost outperformed other machine learning frameworks. This is mainly because it combines several models into a single model correcting the errors in those models.

Extensions of our project :

- With the dataset, we can identify potential threats and save lives on earth.
- This also helps in securing NASA missions to the universe, gives best insights about the earth’s surroundings and makes the process smooth.
- In our project we used absolute magnitude, relative velocity to predict hazardous nature of Near earth objects. We can extend this further by predicting the close approach date based on relative velocity and miss distance.

Exploration of other avenues related to the problem:

- We can get to know more about the universe and its formation.
- How life may have developed on Earth, scientists are interested in learning how planets formed. The existence of specific minerals was discovered in NEO’s dust samples from NASA's Stardust mission.