MIT | Academy of Engineering

(An Autonomous Institute Affiliated to Savitribai Phule University)

Soft Computing -lab

TYBTECH 2023-24 SEM -6

END SEMESTER EXAMINATION

Project Report

ON

Diameter prediction and asteroid classification based on ensemble learning

Students name:

202101070094 Omkar Awari

202101070110 Ayush Gohatre

202101070120 Sanika Thakare

202101070081 Charu Vaidya

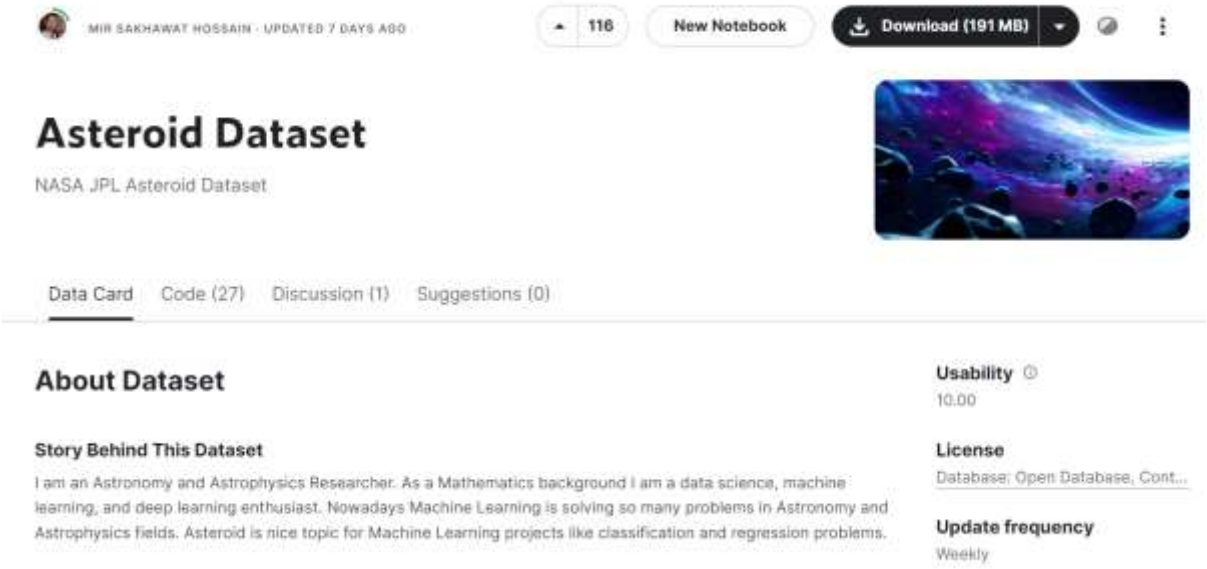
Guided by :

Dr. Smita Kulkarni

Contents

1.Dataset Description :

1. NASA JPL Asteroid Dataset
2. Collected this Dataset from kaggle which is officially maintained by Jet Propulsion Laboratory of California Institute of Technology which is an organization under NASA.
3. In this Dataset all kinds of Data related to Asteroid is included.
4. This Dataset is publicly available in their website.
5. Contains more than 9 lakh unique values.
6. Unbalanced dataset that is best suitable for ensemble learning methods.



2.Problem Statement:

To study and implement Ensemble Learning Algorithms and Comparative analysis for asteroid classification and diameter prediction.

3.Tools and libraries:

For several ensemble learning models, feature importance, gridsearch cv, to find the best parameters, data preprocessing and cleaning, EDA , there are several libraries that are always needed to import first and then used. The detailed description of this libraries and tools can be found as below:

IDE used : GOOGLE Colab notebook

Dataset : Stored at Google Drive

Libraries used :

1.numpy:

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

2.matplotlib:

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is commonly used for creating plots, charts, and other types of data visualizations.

3.seaborn:

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. Seaborn is particularly useful for visualizing complex datasets and for making your plots more visually appealing.

4.pandas:

Pandas is a powerful data manipulation and analysis library for Python. It provides data structures and functions for efficiently manipulating large datasets, including tools for reading and writing data in different file formats, data cleaning, reshaping, aggregating, and more.

5.sklearn (scikit-learn):

Scikit-learn is a versatile machine learning library for Python. It features various algorithms for classification, regression, clustering, dimensionality reduction, and more, along with tools for model evaluation and selection, preprocessing, and data splitting.

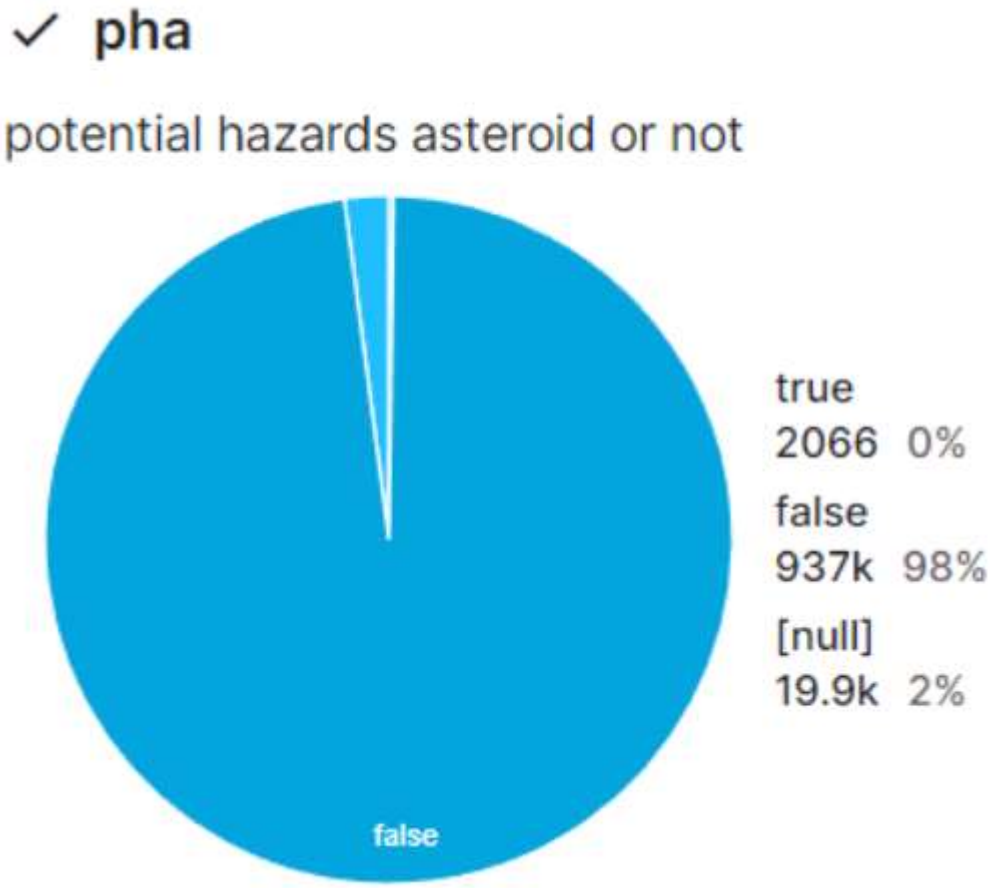
6.XGBoost:

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements machine learning algorithms under the Gradient Boosting framework, and it is known for its performance and scalability in machine learning competitions and real-world applications.

7.CatBoost:

CatBoost is a gradient boosting library specifically designed for categorical variables. It provides state-of-the-art results and handles categorical features automatically, without the need for extensive preprocessing. CatBoost is particularly useful in scenarios where you have datasets with a mix of categorical and numerical features.

4.Data Exploration:



5.Data description

Total no. of unique values :

958524

Dimensions of dataset :

5 rows* 45 columns

Columns in dataset :

Index(['id', 'spkid', 'full_name', 'pdes', 'name', 'prefix', 'neo', 'pha', 'H',
'diameter', 'albedo', 'diameter_sigma', 'orbit_id', 'epoch',
'epoch_mjd', 'epoch_cal', 'equinox', 'e', 'a', 'q', 'i', 'om', 'w',
'ma', 'ad', 'n', 'tp', 'tp_cal', 'per', 'per_y', 'moid', 'moid_ld',
'sigma_e', 'sigma_a', 'sigma_q', 'sigma_i', 'sigma_om', 'sigma_w',
'sigma_ma', 'sigma_ad', 'sigma_n', 'sigma_tp', 'sigma_per', 'class',
'rms'],
dtype='object')

Asteroid classification done on column :

pha

No.of null values:

id	0
spkid	0
full_name	0
pdes	0
name	936460
prefix	936460
neo	4
pha	19921

H	6263
diameter	822315
albedo	823421
diameter_sigma	822443
orbit_id	0
epoch	0
epoch_mjd	0
epoch_cal	0
equinox	0
e	0
a	0
q	0
i	0
om	0
w	0
ma	1
ad	4
n	0
tp	0
tp_cal	0
per	4
per_y	1
moid	19921
moid_ld	127
sigma_e	1922
sigma_a	19922
sigma_q	19922
sigma_i	19922
sigma_om	19922
sigma_w	19922
sigma_ma	19922
sigma_ad	19926
sigma_n	19922
sigma_tp	19922
sigma_per	19926
class	0
rms	2

Explaining dataset values:

SPK-ID: Object primary SPK-ID

Object ID: Object internal database ID

Object fullname: Object full name/designation

pdes: Object primary designation

name: Object IAU name

NEO: Near-Earth Object (NEO) flag

PHA: Potentially Hazardous Asteroid (PHA) flag

H: Absolute magnitude parameter

Diameter: object diameter (from equivalent sphere) km Unit

Albedo: Geometric albedo

Diameter_sigma: 1-sigma uncertainty in object diameter km Unit

Orbit_id: Orbit solution ID

Epoch: Epoch of osculation in modified Julian day form

Equinox: Equinox of reference frame

e: Eccentricity

a: Semi-major axis au Unit

q: perihelion distance au Unit

i: inclination; angle with respect to x-y ecliptic plane

tp: Time of perihelion passage TDB Unit

moid_ld: Earth Minimum Orbit Intersection Distance au Unit

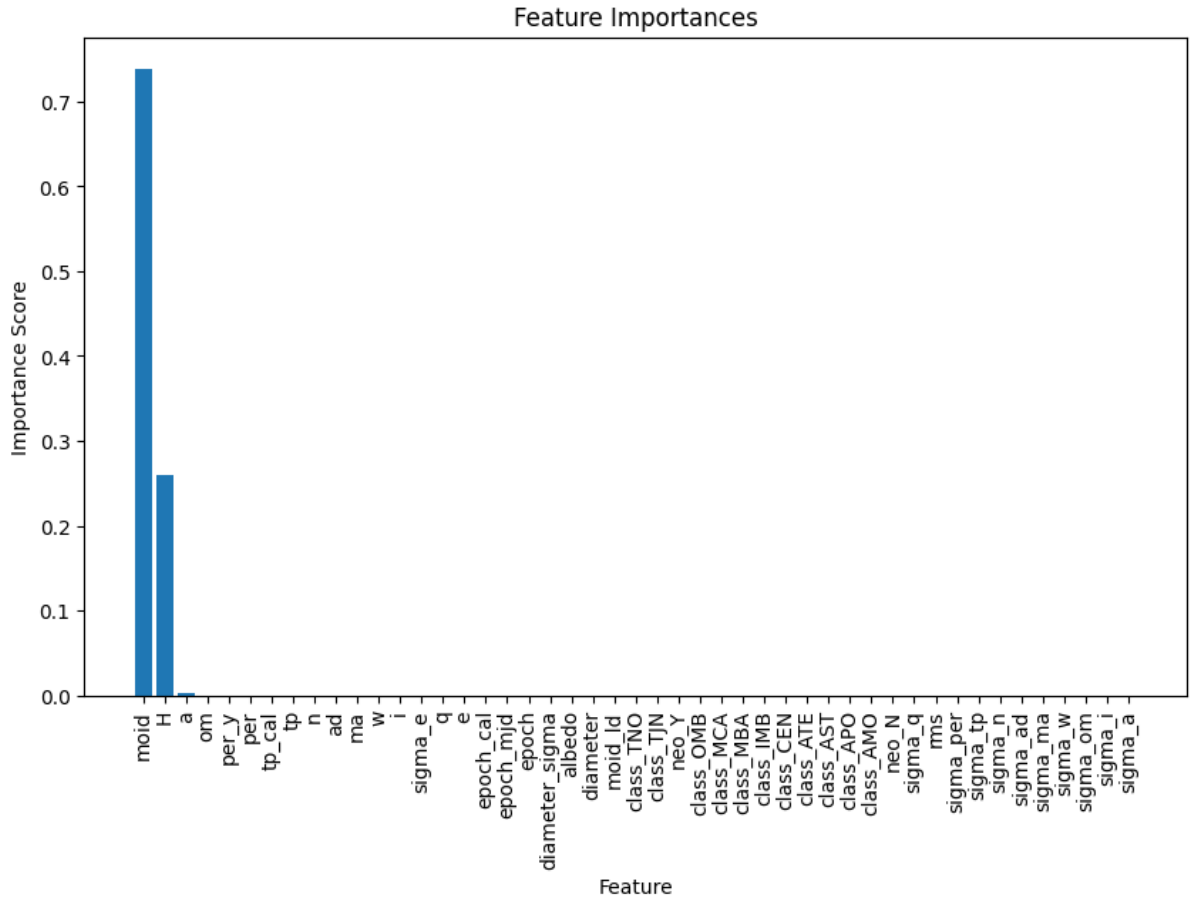
6.Data Cleaning

1.Removed null values.

2.Dropped following columns

- id
- spkid
- orbit_id
- full_name
- equinox

3.Calculate the Feature importance and removing the one with lowest importance :



4.Encoding is done.

```
df= pd.get_dummies(df)
df
```

	pha	H	diameter	albedo	diameter_sigma	epoch	epoch_mjd	epoch_cal	e	a	...	class_APO	class_AST	class_ATE	class_CEN	class_IMB
0	0	3.40	939.400	0.0900	0.200	2458600.5	58600	20190427.0	0.076009	2.769165	...	False	False	False	False	False
1	0	4.20	545.000	0.1010	18.000	2458000.5	58000	20200531.0	0.229972	2.773841	...	False	False	False	False	False
2	0	5.33	246.596	0.2140	10.594	2458000.5	58000	20200531.0	0.256936	2.668285	...	False	False	False	False	False
3	0	3.00	625.400	0.4228	0.200	2458600.5	58600	20190427.0	0.088721	2.361418	...	False	False	False	False	False
4	0	6.90	106.699	0.2740	3.140	2458000.5	58000	20200531.0	0.190913	2.574037	...	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
891599	0	16.20	3.760	0.0210	1.375	2459000.5	59000	20200531.0	0.158579	3.169836	...	False	False	False	False	False
891841	0	17.40	2.696	0.0610	0.701	2459000.5	59000	20200531.0	0.074890	2.550597	...	False	False	False	False	False
894103	0	17.20	3.271	0.0720	1.074	2459000.5	59000	20200531.0	0.287884	3.050244	...	False	False	False	False	False
901055	0	16.00	3.009	0.0760	0.961	2459000.5	59000	20200531.0	0.240246	3.191395	...	False	False	False	False	False
909489	0	18.30	1.600	0.0230	0.283	2459000.5	59000	20200531.0	0.108726	2.418140	...	False	False	False	False	False

131142 rows × 48 columns

5.splitting of dataset.

```
] X_train.shape
```

```
(91799, 47)
```

```
] X_test.shape
```

```
(39343, 47)
```

```
y_train.shape
```

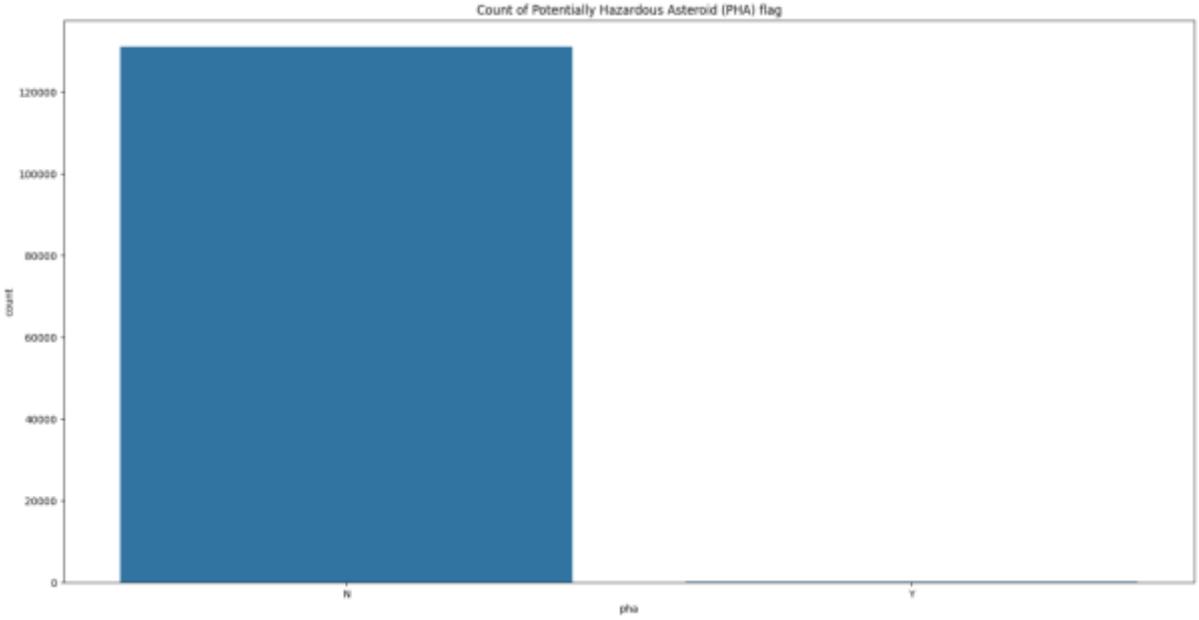
```
(91799,)
```

```
] y_test.shape
```

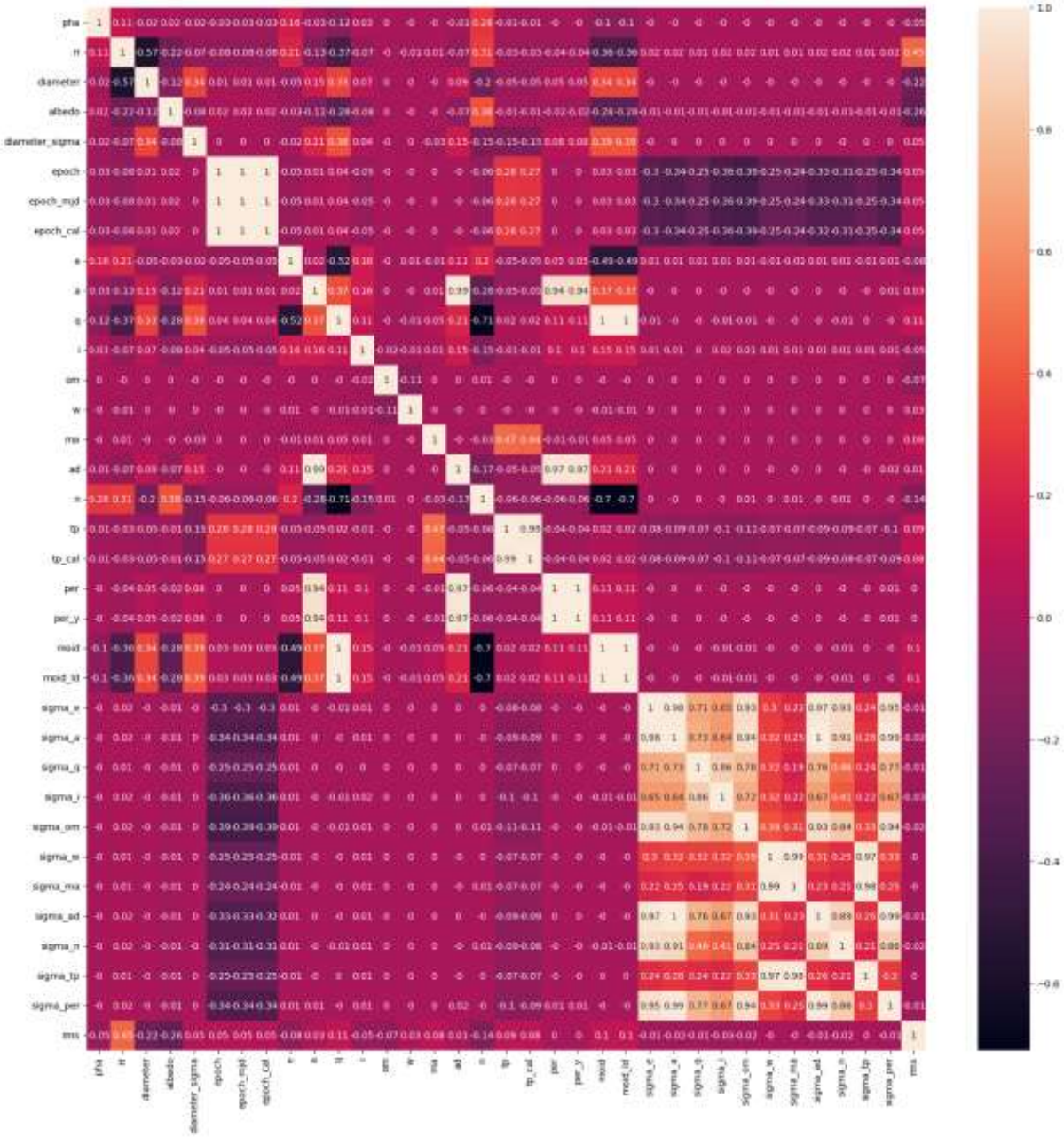
```
(39343,)
```

7.EDA with graphical report

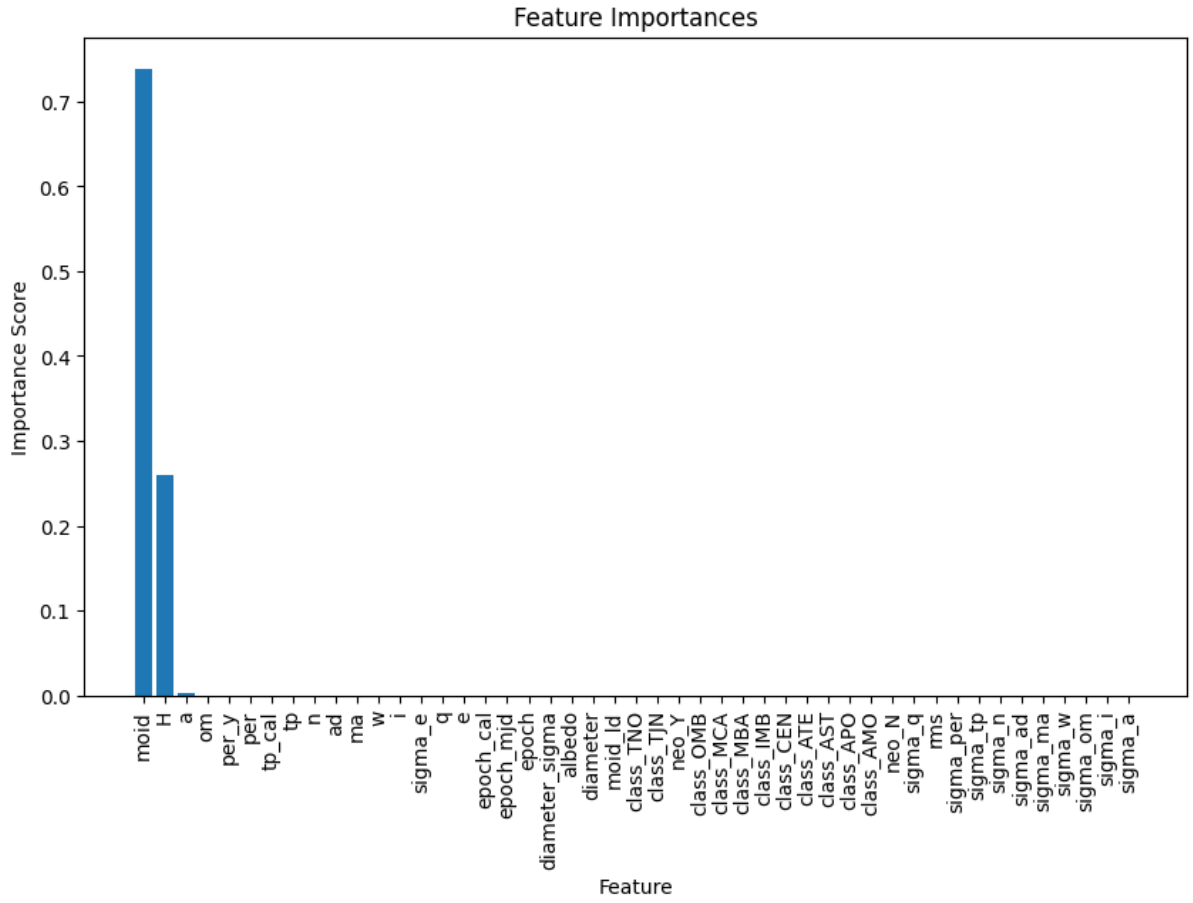
No.of pha flag :



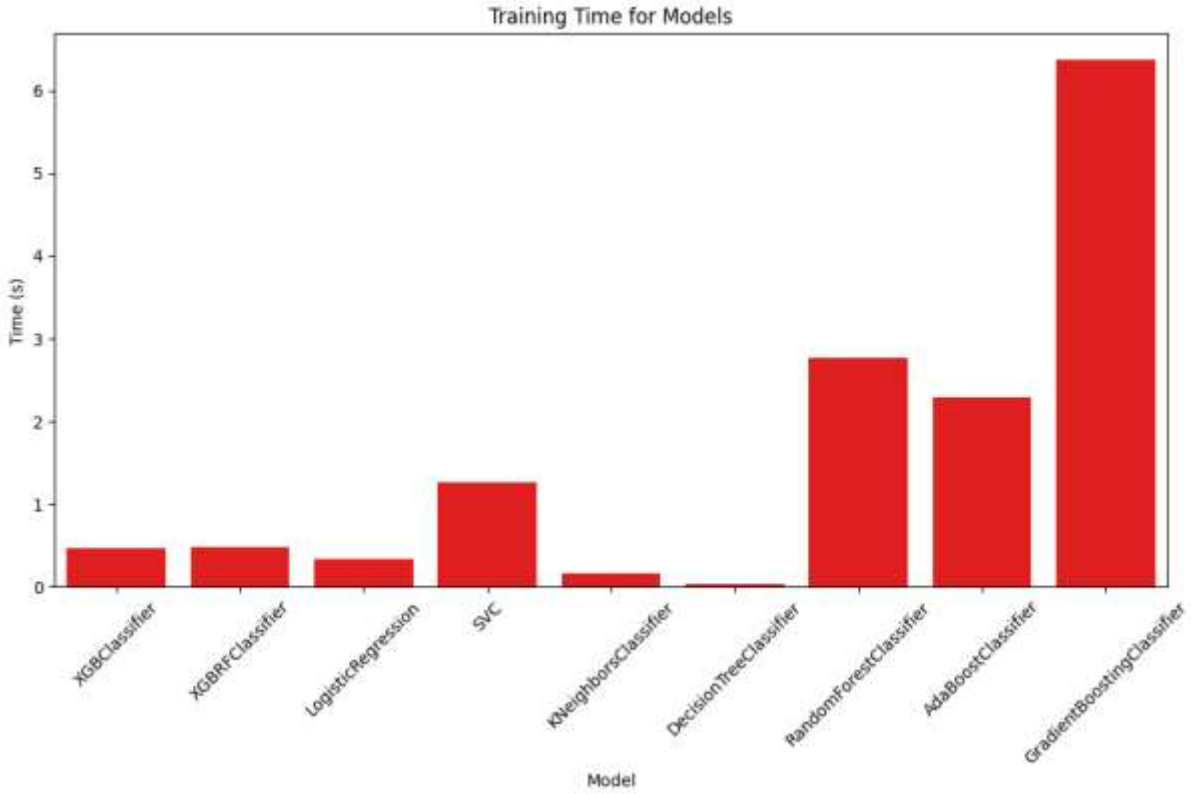
Heatmap:


pha
H
diameter
albedo
diameter_sigma
epoch
epoch_mjd
epoch_cal
e
a
q
i
om
w
ma
ad
n
tp
tp_cal
per
per_y
moid
moid_ld
sigma_e
sigma_a
sigma_q
sigma_i
sigma_om
sigma_w
sigma_ma
sigma_ad
sigma_n
sigma_tp
sigma_per
rms
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

Feature importance :



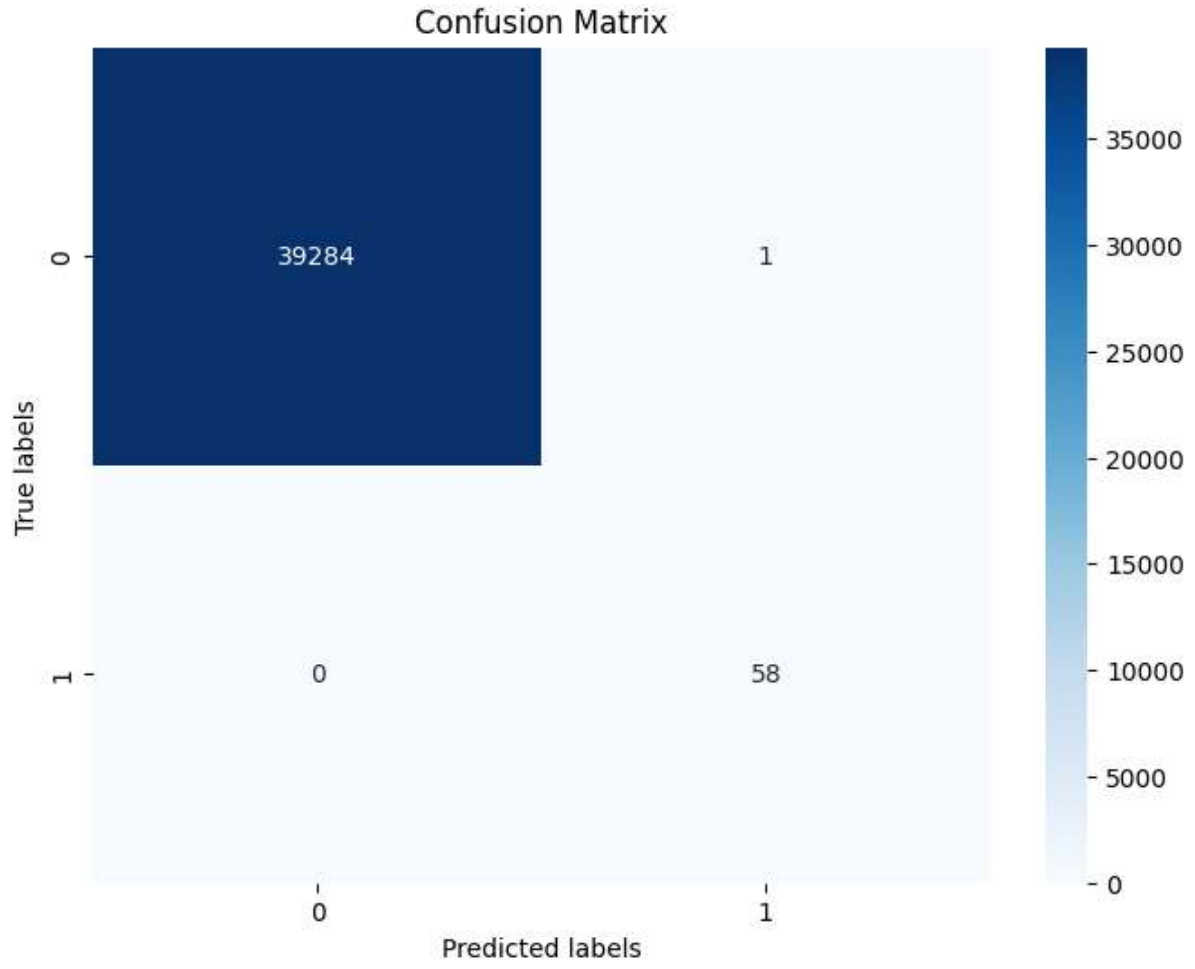
Time taken by models to train :

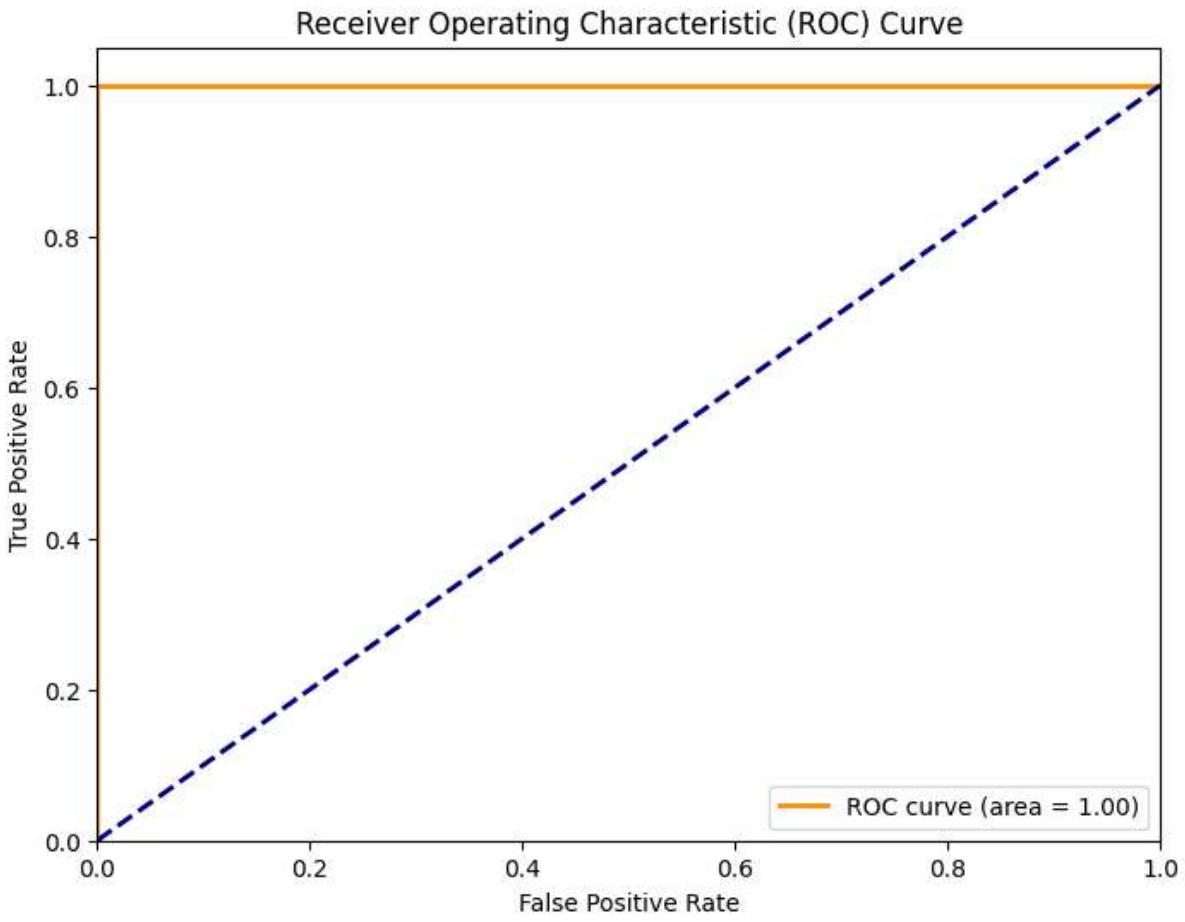


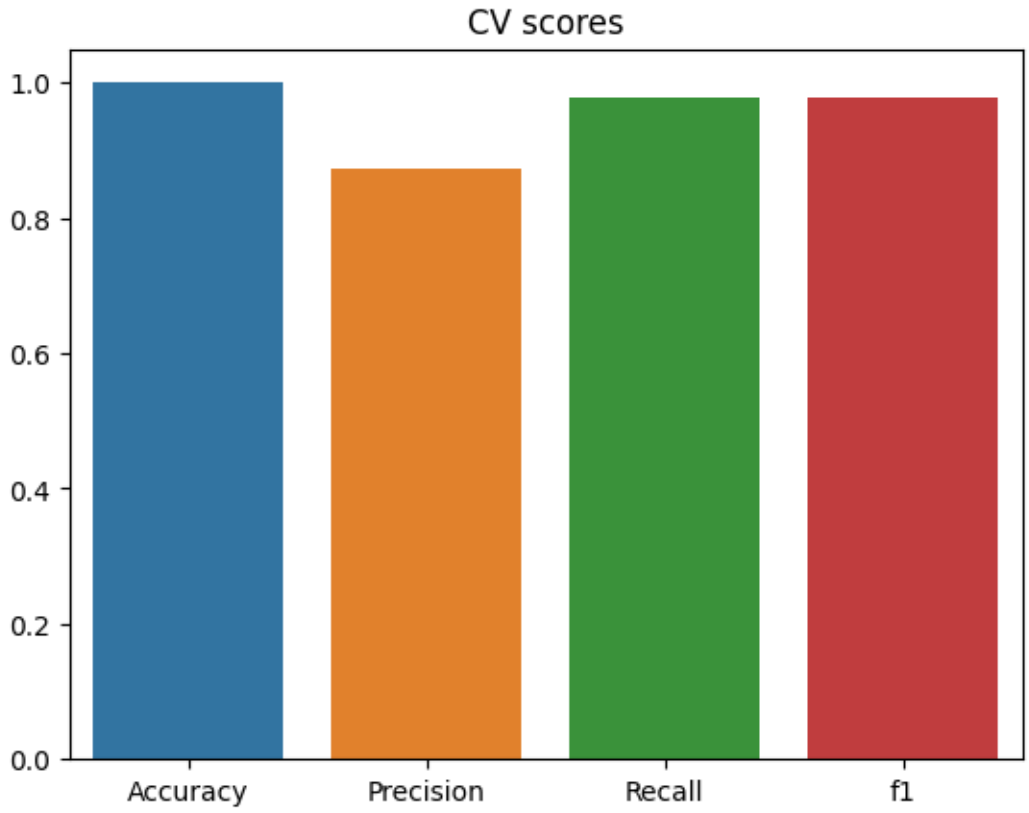
8.ML model implementation with results

1.adaboost:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	39285
1	0.98	1.00	0.99	58
accuracy			1.00	39343
macro avg	0.99	1.00	1.00	39343
weighted avg	1.00	1.00	1.00	39343



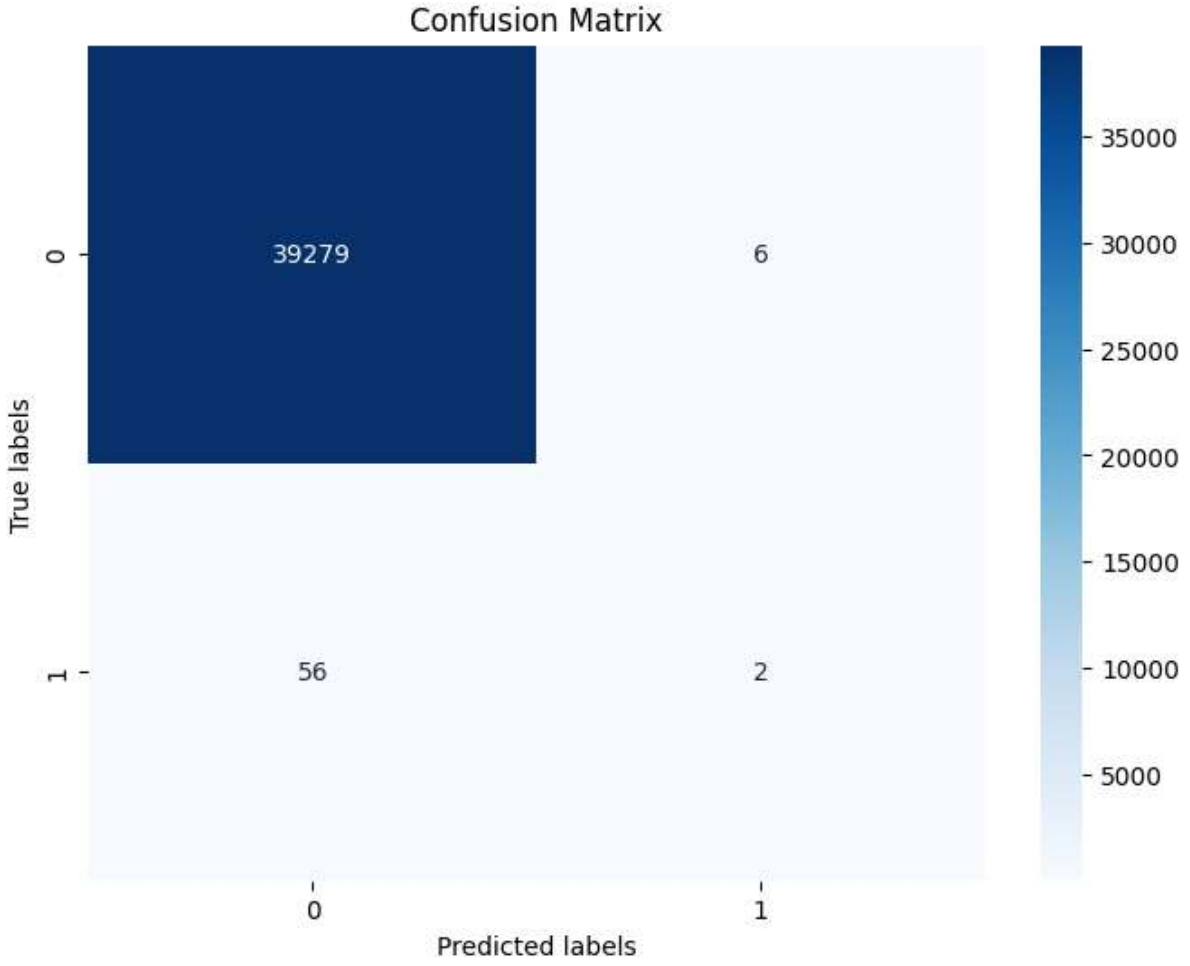

Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 1.00)



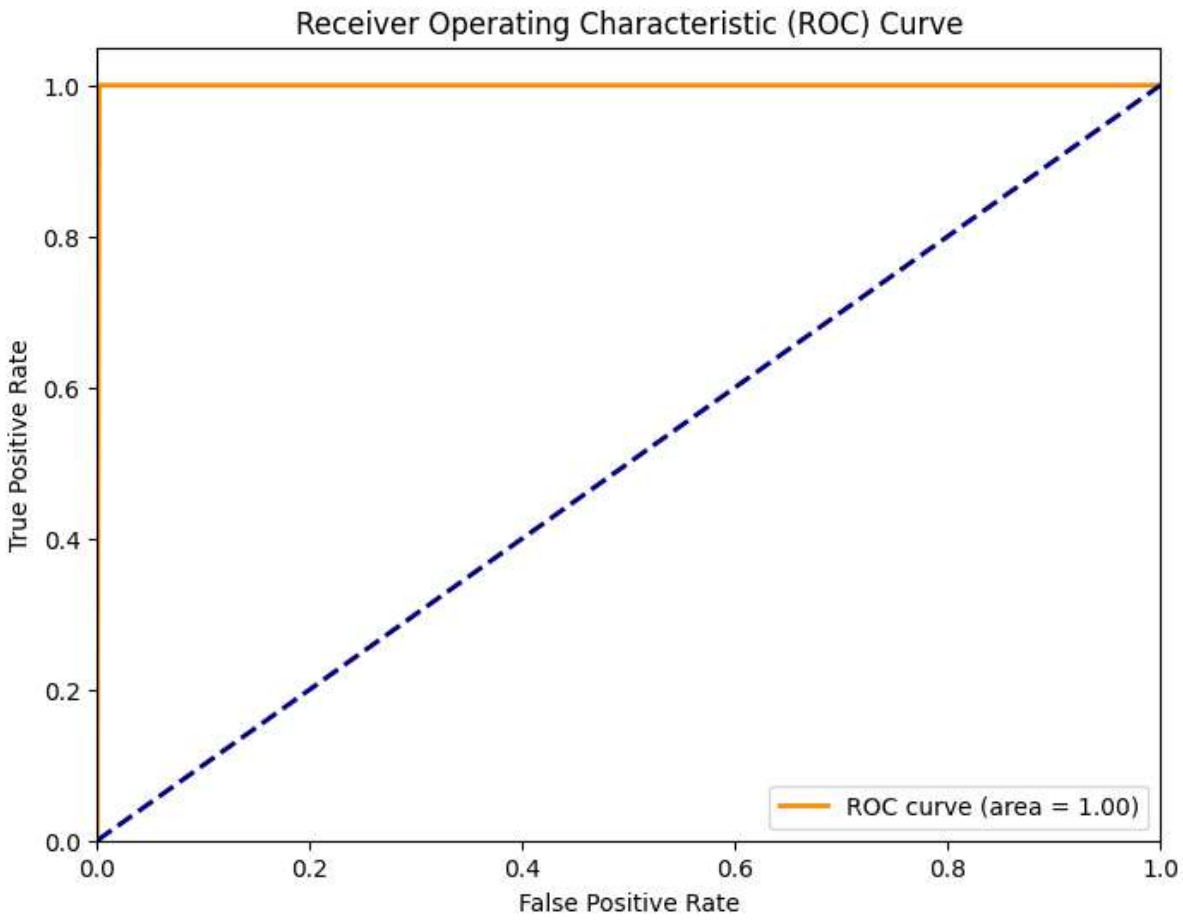
	Accuracy	Precision	Recall	f1
0	0.999497	0.873469	0.977778	0.977778

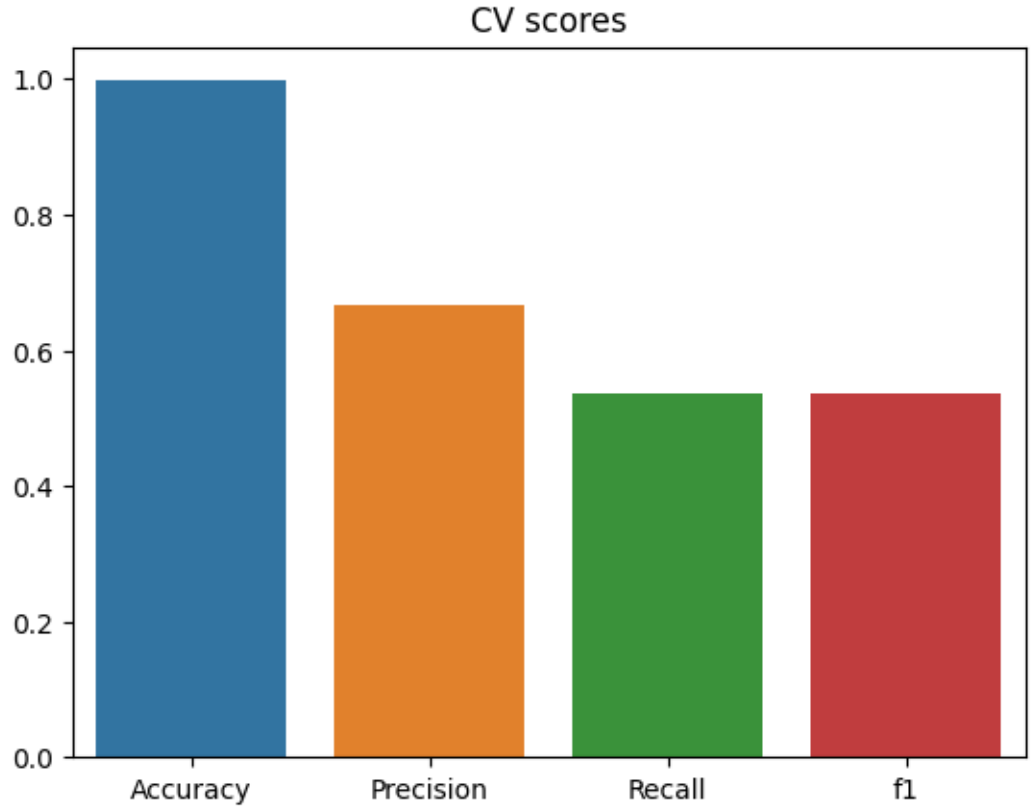
2.xgb classifier:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	39285
1	0.25	0.03	0.06	58
accuracy			1.00	39343
macro avg	0.62	0.52	0.53	39343
weighted avg	1.00	1.00	1.00	39343

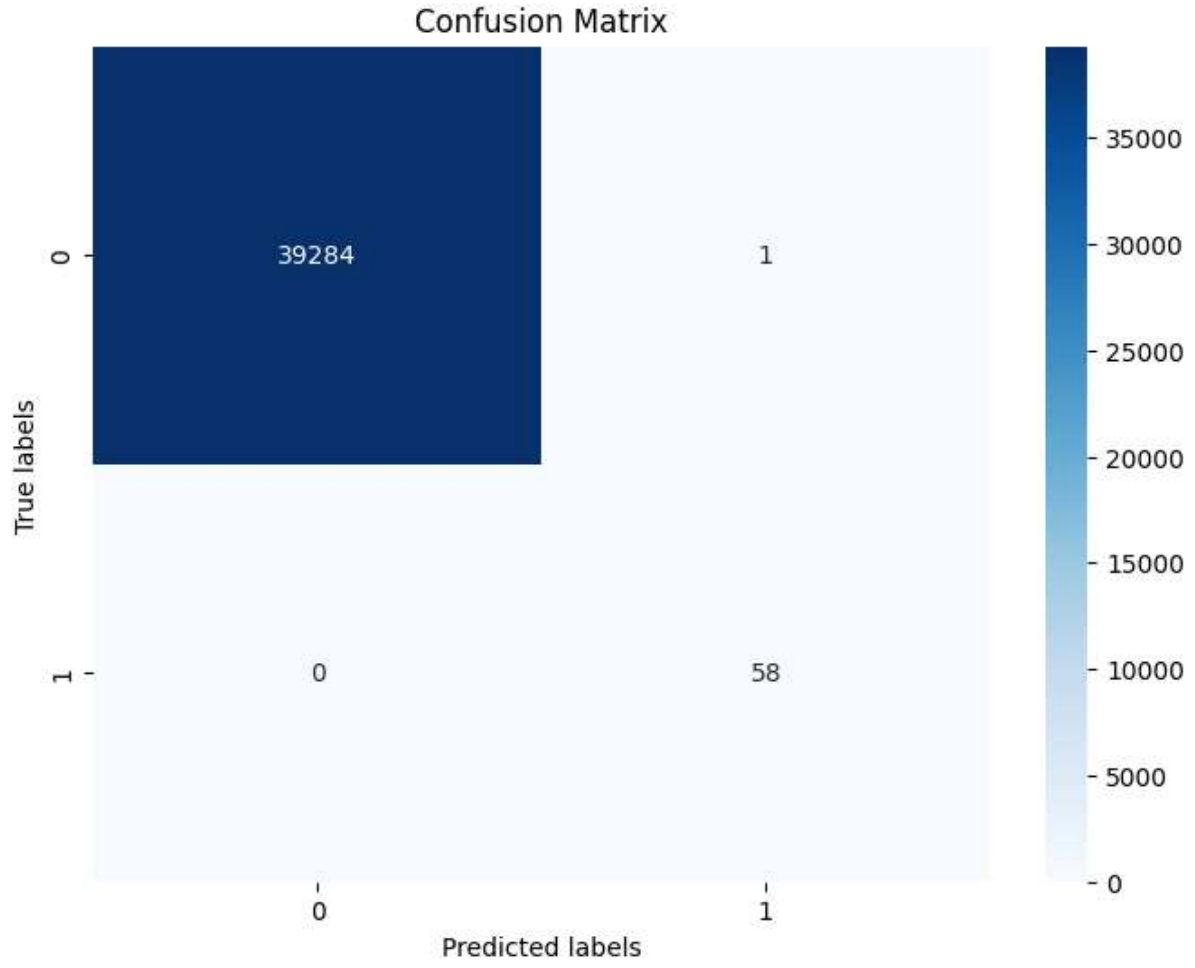


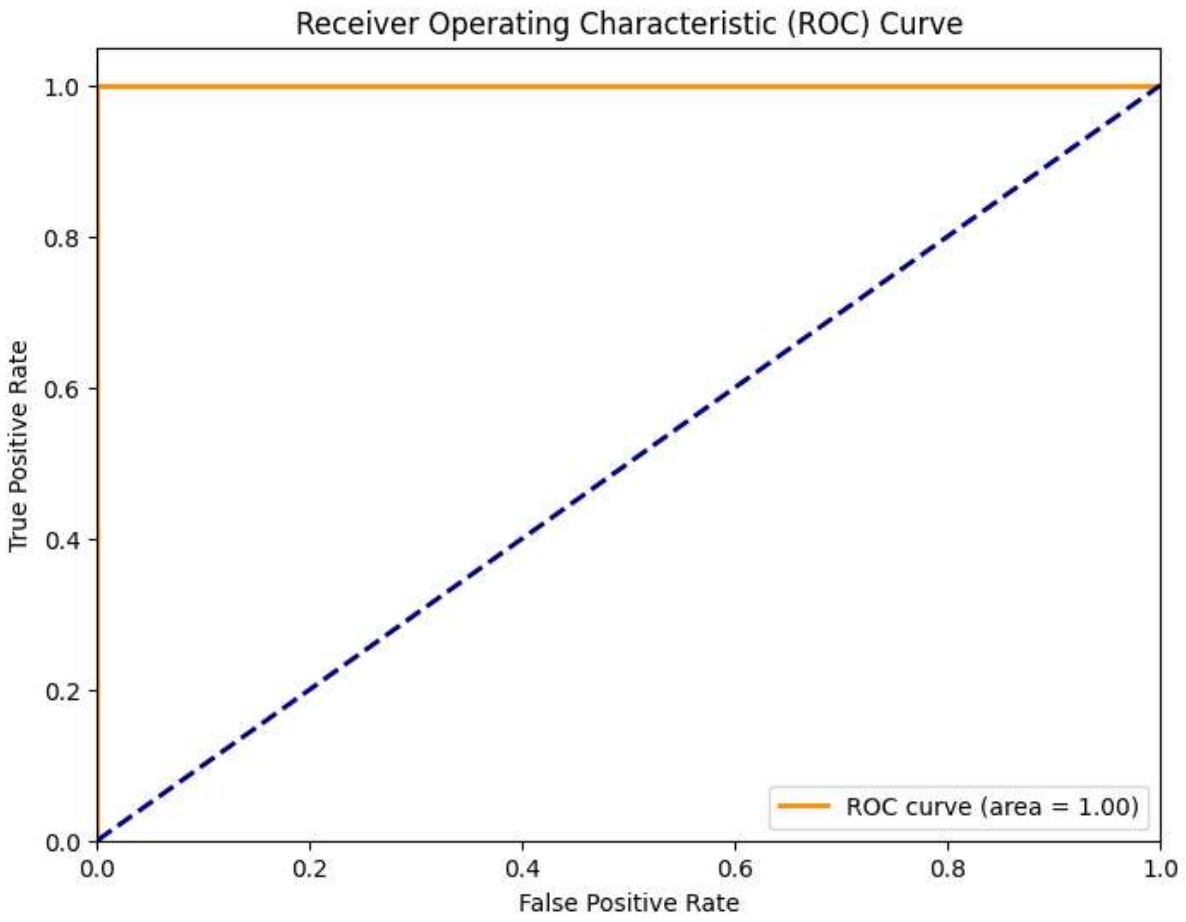

```
Best parameters: {'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 100}
Training time: 85.01 seconds
```


Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 1.00)


CV scores
0.0
0.2
0.4
0.6
0.8
1.0
Accuracy
Precision
Recall
f1

3.gradientboosting classifier:






```
Best parameters: {'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 300}
Training time: 637.90 seconds
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     39285
           1       0.98      1.00      0.99        58

    accuracy                           1.00     39343
   macro avg       0.99      1.00      1.00     39343
weighted avg       1.00      1.00      1.00     39343
```

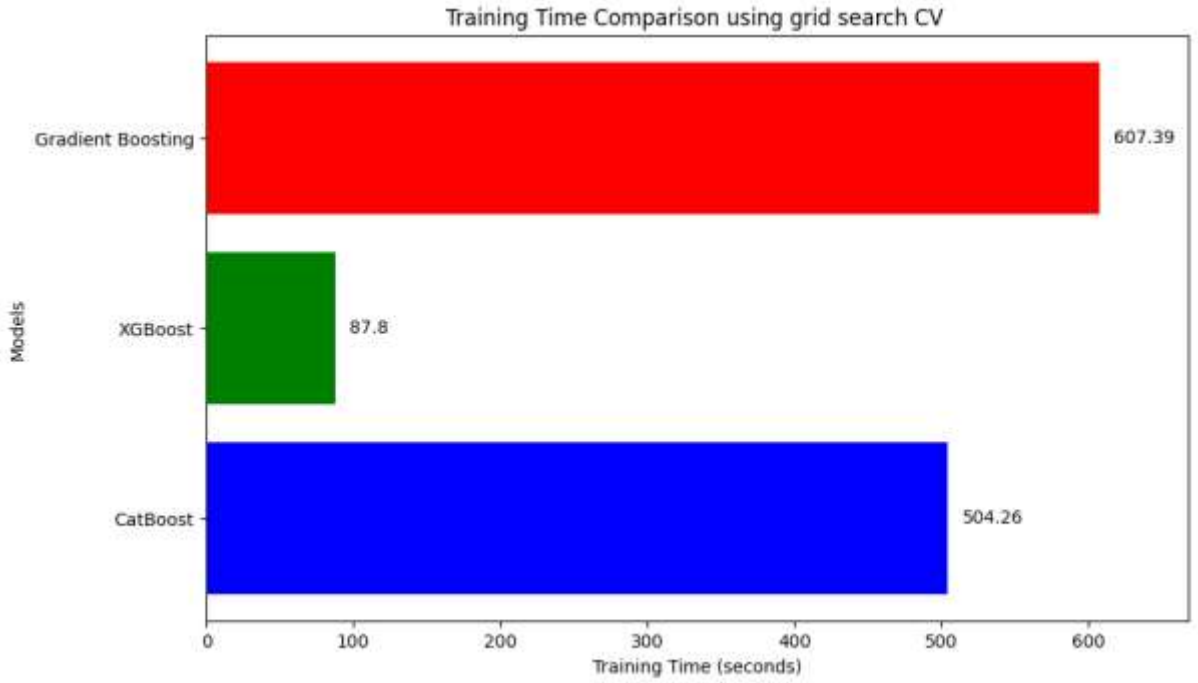
4.cat-boost:

```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     39285
           1       0.00      0.00      0.00        58

    accuracy                           1.00     39343
   macro avg       0.50      0.50      0.50     39343
weighted avg       1.00      1.00      1.00     39343
```

9.Comparative Analysis of ML models



Explanation :

Function	XGBoost	CatBoost	Light GBM
Important parameters which control overfitting	1. **learning_rate or eta** – optimal values lie between 0.01-0.2 2. **max_depth** 3. **min_child_weight:** similar to min_child leaf; default is 1	1. **Learning_rate** 2. **Depth** - value can be any integer up to 16. Recommended - [1 to 10] 3. No such feature like min_child_weight 4. **l2-leaf-reg**: L2 regularization coefficient. Used for leaf value calculation (any positive integer allowed)	1. **learning_rate** 2. **max_depth**: default is 20. Important to note that tree still grows leaf-wise. Hence it is important to tune **num_leaves** (number of leaves in a tree) which should be smaller than 2^(max_depth). It is a very important parameter for LGBM 3. **min_data_in_leaf**: default=20, alias= min_data, min_child_samples
Parameters for categorical values	Not Available	1. **cat_features**: It denotes the index of categorical features 2. **one_hot_max_size**: Use one-hot encoding for all features with number of different values less than or equal to the given parameter value (max – 255)	1. **categorical_feature:** specify the categorical features we want to use for training our model
Parameters for controlling speed	1. **colsample_bytree**: subsample ratio of columns 2. **subsample**: subsample ratio of the training instance 3. **n_estimators**: maximum number of decision trees; high value can lead to overfitting	1. **rsm:** Random subspace method. The percentage of features to use at each split selection 2. No such parameter to subset data 3. **iterations**: maximum number of trees that can be built; high value can lead to overfitting	1. **feature_fraction:** fraction of features to be taken for each iteration 2. **bagging_fraction:** data to be used for each iteration and is generally used to speed up the training and avoid overfitting 3. **num_iterations:** number of boosting iterations to be performed; default=100

10.References:

1.Machine Learning Approaches for Classification and Diameter Prediction of Asteroids.

January 2023

DOI: 10.1007/978-981-19-7528-8_4

In book: Proceedings of International Conference on Information and Communication Technology for Development

Lab: Mir Sakhawat Hossain's Lab

Mir Sakhawat HossainMir Sakhawat HossainMd. Akib Zabed

from Research Gate

https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

https://catboost.ai/en/docs/concepts/pythonreference_catboostclassifier

https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

https://scikitlearn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

11.Conclusion:

Different models from ensemble learning that are implemented above gives us different feature importances in each model. Similarly the Evaluation Metrices for the models are changed once we performed cross validation technique. We also found that training time is highest for the ensemble learning models.