Logo, company name

Description automatically generated

***Faculty of Science and Technology***

**Assignment Coversheet**

|  |  |
| --- | --- |
| **Student ID number &**  **Student Name** | U3258398 William Gunn |
| **Unit name** | Software technology |
| **Unit number** | 4483 |
| **Unit Tutor** | Pranav Gupta |
| **Assignment name** | ST1 Capstone Project – Semester 1 2023 |
| **Due date** | 12/05/2023 |
| **Date submitted** | 12/05/2023 |

**You must keep a photocopy or electronic copy of your assignment.**

**Student declaration**

I certify that the attached assignment is my own work. Material drawn from other sources has been appropriately and fully acknowledged as to author/creator, source and other bibliographic details.

**Signature of student: WG Date: 13/4/23**

Table of Contents

[Introduction 3](#_Toc133346613)

[Methodology 4](#_Toc133346614)

[1. Design and Development 4](#_Toc133346615)

[2. Implementation 4](#_Toc133346616)

[3. Deployment 4](#_Toc133346617)

[Design and development 5](#_Toc133346618)

[Algorithm for dataset: 5](#_Toc133346619)

[Dataset description 5](#_Toc133346620)

[Exploratory Data Analysis 6](#_Toc133346621)

[Questions to answer when analysing the dataset: 6](#_Toc133346622)

[Predictive Data Analytics: 7](#_Toc133346623)

[Steps: 7](#_Toc133346624)

[Model Preparation and Development 8](#_Toc133346625)

[Implementation and Deployment 15](#_Toc133346626)

[Conclusion 16](#_Toc133346627)

[References 17](#_Toc133346628)

[Appendix 18](#_Toc133346629)

[Journal 18](#_Toc133346630)

[Week 9: 18](#_Toc133346631)

[Week 10: 18](#_Toc133346632)

[Week 11: 18](#_Toc133346633)

# Introduction

This report describes the characteristics of the asteroids that orbit around the earth and assesses if they are lethal by trajectory and mass. The report identifies which asteroids are hazardous and what characteristics make the asteroid hazardous. I have decided to use a dataset from Kaggle published by NASA that records the relevant data.

Space is a three dimensional perfect vacuum(Space, 2022) that contains different kinds of objects, with some of the smallest objects being asteroids. Asteroids are small, rocky objects(SpacePlace, 2023), sometimes referred to as minor planets, that consist of clay and silicate. There are approximately 1 million(NASA, 2023) asteroids in space which orbit the Sun, and roughly 30 thousand(NASA, 2023) that orbit around the earth. These asteroids come in many different shapes and sizes and orbit at an extreme velocity, some being able to deal serious damage if they were to collide with Earth.

Why a tool would be useful.

* Sending a mission into space to do research would be very expensive
* Could help a planetary defence coordination office work faster and more efficiently

Although space can be unpredictable, a software tool could evaluate which asteroids are likely to be the most dangerous, so that an assessment can be made on how best to deal with them.

This report presents the details of a prototype software in the form of a python module and a google collab repository. Based on a data driven scientific approach, involving exploratory data analysis, predictive analysis and implementation as a GUI and desktop application with Tkinter and Streamlit respectively.

# Methodology

## Design and Development

Perform Exploratory data analysis (EDA) and predictive data analysis (PDA) to identify the best artificial intelligence learning model to perform the calculations/predictions needed to solve a real world problem

## Implementation

The best performing AI learning model gets implemented into a GUI program through the use of Tkinter

## Deployment

After successful implementation deploy the tool as a web application

# Design and development

## Algorithm for dataset:

LOAD DATASET

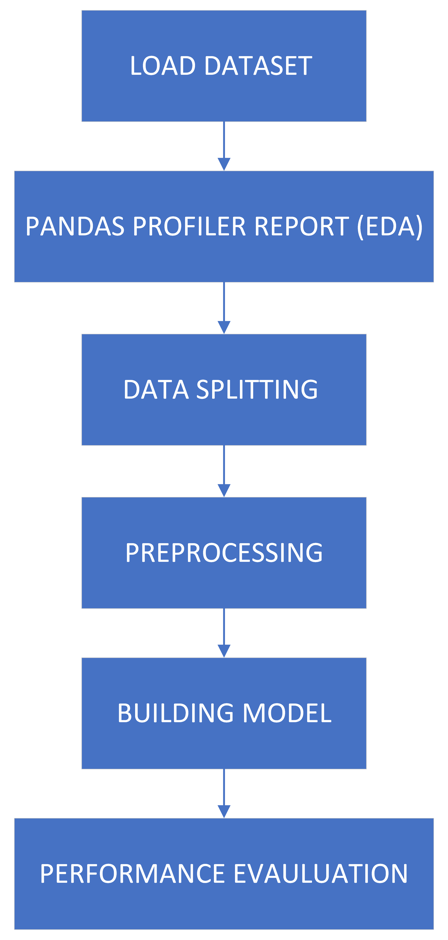
PANDAS PROFILER REPORT (EDA)

DATA SPLITTING

PREPROCESSING

BUILDING MODEL

PERFORMANCE EVAULUATION



## Dataset description

Only one dataset is being used in this project and it is publicly available on Kaggle. The dataset consists of almost 100000 asteroids and has 9 features and 1 target attribute. The 9 features are id, name, estimated minimum diameter, estimated maximum diameter, relative velocity, miss distance, orbiting body, sentry object, absolute magnitude and the target variable is the specification of the asteroid and determining if it is hazardous or not hazardous. The task at hand is to develop a software tool to predict if an asteroid is going to be hazardous or not depending on its specs. The data was collected by NASA.

## Exploratory Data Analysis

The EDA process was done on the experimental environment google collab, as it is easy and fast to run and can be done from a browser. The programming language python was used to create the scripts which were run on google collab Jupyter notebook.

### Questions to answer when analysing the dataset:

1. What is the closest object to earth
2. What is the farthest object to earth
3. What is the biggest object
4. What has the most velocity
5. How many asteroids are hazardous

The first step to EDA is to understand the basic description of the data:

from google.colab import drive

drive.mount("/content/drive")

import pandas as pd

from pandas\_profiling import ProfileReport

#Read the dataset

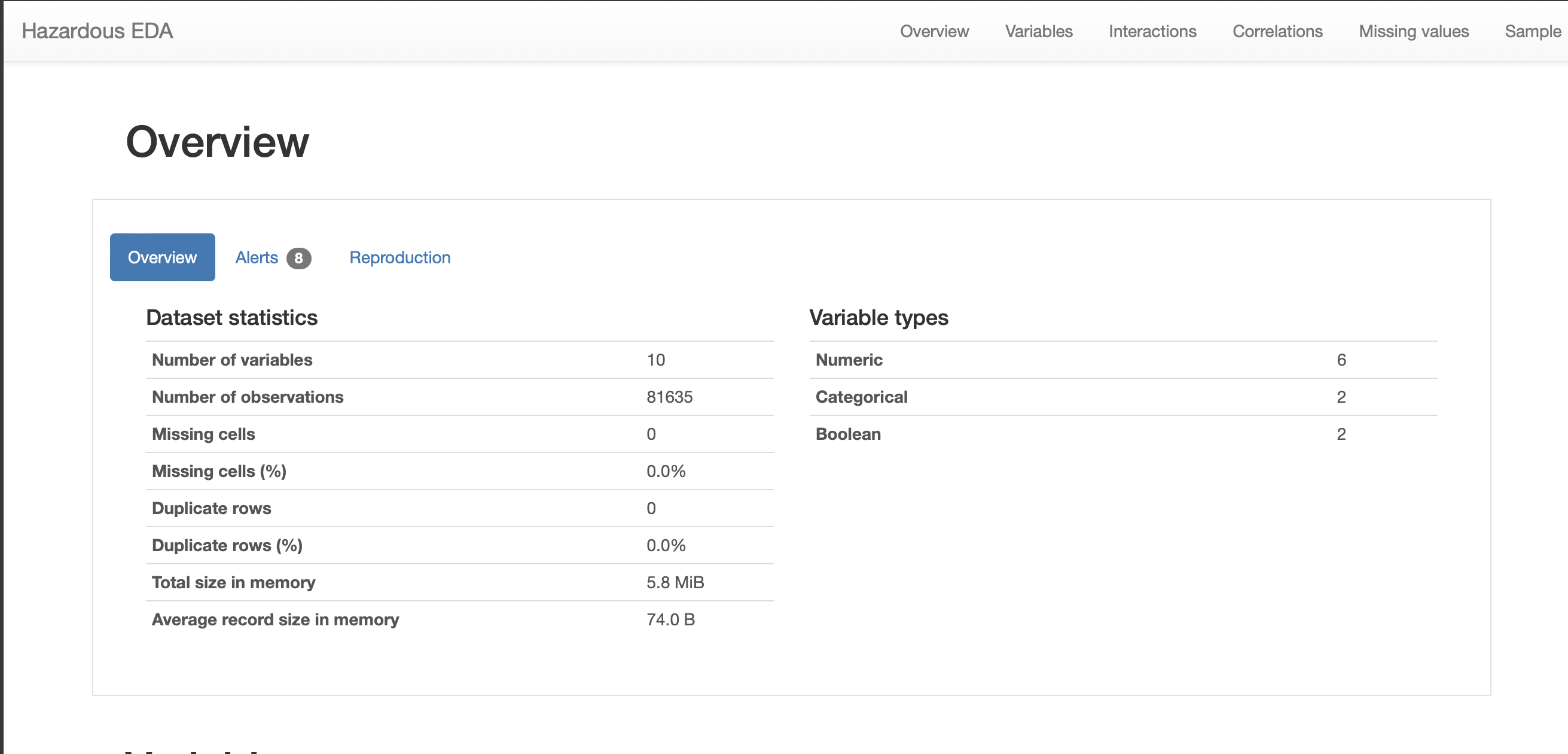
df = pd.read\_csv('/content/drive/MyDrive/MyCapstone/neo\_v2.csv')

#obtain full profiler report

profile = ProfileReport(df,title="Hazardous EDA",

html={'style':{'full\_width':True}})

profile.to\_notebook\_iframe()



## Predictive Data Analytics:

To preform PDA several steps are required. This includes pre-processing, classifier comparison to identify the best machine learning classifier and performance evaluation with different objective metrics such as accuracy, classification report, confusion matrix, ROC\_AUC curve and prediction report was obtained using python scikit-learn

### Steps:

**Pre processing**

* The data has both continuous and categorical attributes/values
* Attribute transformation, standardisation, normalisation (scikit-learn OrdinalEncoder() to perform attribute transformation

**Normalisation**

* Drop the target from the data frame, normalise it then reattach the target to the data frame

#===Pre-processing====

from sklearn.exceptions import DataDimensionalityWarning

#encode object columns to integers

from sklearn import preprocessing

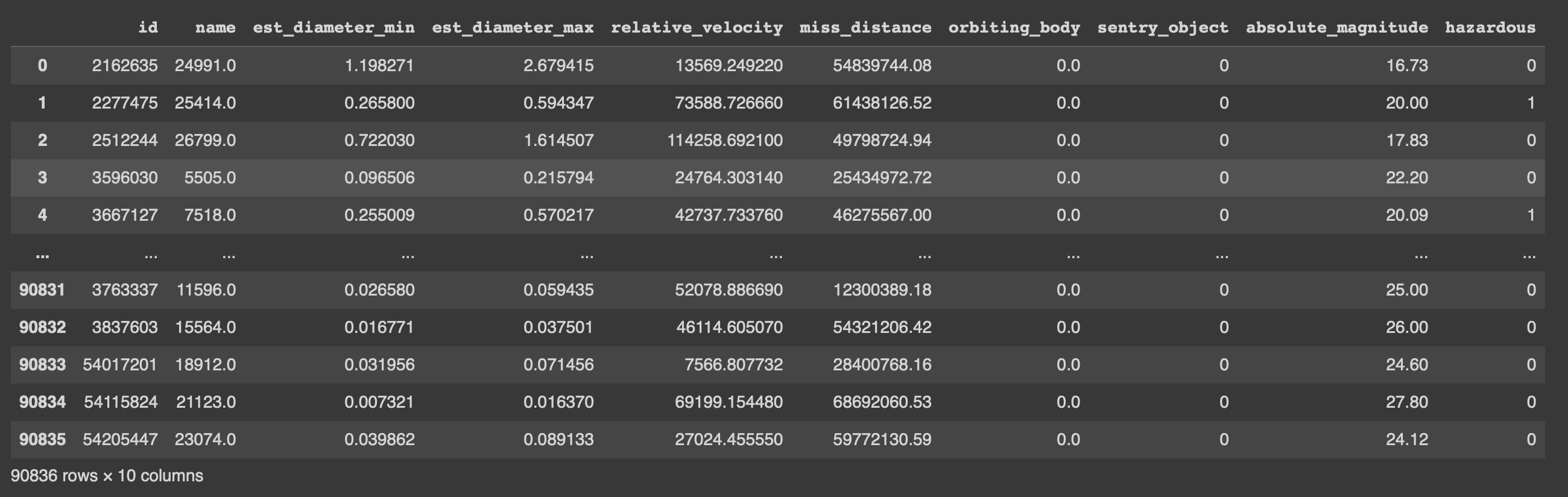
from sklearn.preprocessing import OrdinalEncoder

for col in df:

if df[col].dtype =='object':

df[col]=OrdinalEncoder().fit\_transform(df[col].values.reshape(-1,1))

df



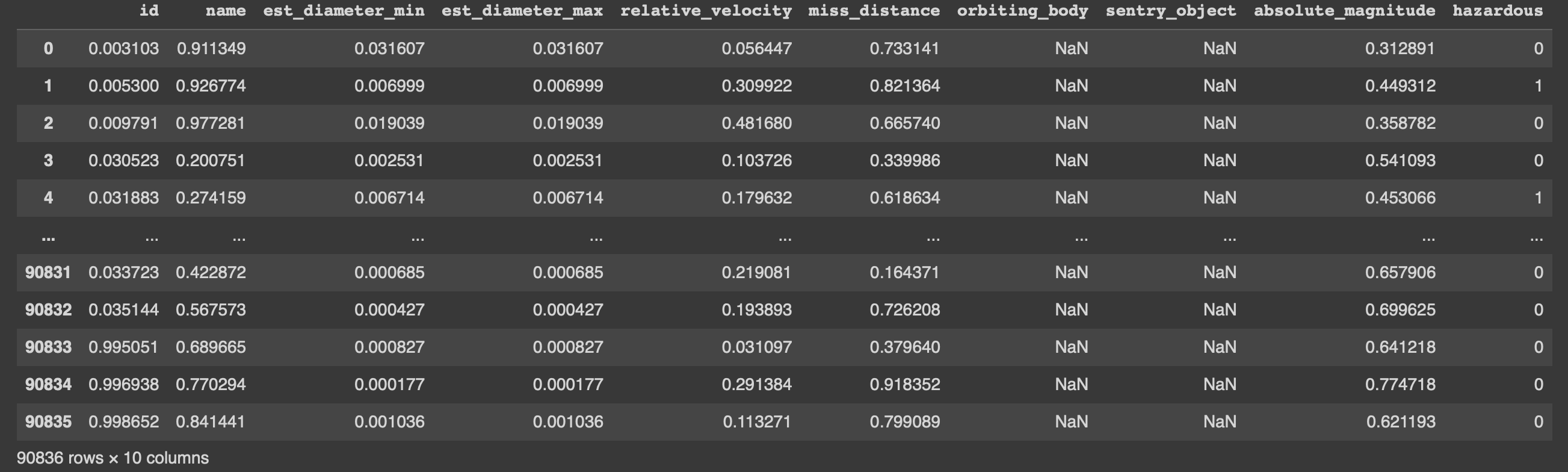
class\_label =df['hazardous']

df = df.drop(['hazardous'], axis =1)

df = (df-df.min())/(df.max()-df.min())

df['hazardous']=class\_label

df



asteroid\_data = df.copy()

le = preprocessing.LabelEncoder()

id = le.fit\_transform(list(asteroid\_data["id"]))

name = le.fit\_transform(list(asteroid\_data["name"]))

est\_diameter\_min = le.fit\_transform(list(asteroid\_data["est\_diameter\_min"]))

est\_diameter\_max = le.fit\_transform(list(asteroid\_data["est\_diameter\_max"]))

relative\_velocity = le.fit\_transform(list(asteroid\_data["relative\_velocity"]))

miss\_distance = le.fit\_transform(list(asteroid\_data["miss\_distance"]))

orbiting\_body = le.fit\_transform(list(asteroid\_data["orbiting\_body"])) # true = 0 false = 1

sentry\_object = le.fit\_transform(list(asteroid\_data["sentry\_object"])) # true = 0 false = 1

absolute\_magnitude = le.fit\_transform(list(asteroid\_data["absolute\_magnitude"]))

hazardous = le.fit\_transform(list(asteroid\_data["hazardous"])) # not hazardous = 0, hazardous = 1

## Model Preparation and Development

To prepare this model and develop it we need to convert the data frame into validation subsets by taking a random sample of 80% of the data and training the AI model on it then leaving the remaining 20% for testing.

The development of this process is done by creating a test set by dropping all of the rows from the data frame then creating an x and y axis with x being everything but the last column and y being the target class (last column).

x = list(zip(est\_diameter\_min, est\_diameter\_max, relative\_velocity, miss\_distance, orbiting\_body, absolute\_magnitude))

y = list(hazardous)

# Test options and evaluation metric

num\_folds = 5

seed = 7

scoring = 'accuracy'

# Model Test/Train

# Splitting what we are trying to predict into 4 different arrays -

# X train is a section of the x array(attributes) and vise versa for Y(features)

#0.2 means 80% training 20% testing, If we train the model with higher data it already has seen that information and knows we will have better accuracy

import sklearn.model\_selection

x\_train, x\_test, y\_train, y\_test = sklearn.model\_selection.train\_test\_split(x, y, test\_size = 0.20, random\_state=seed)

#size of train and test subsets after splitting

import numpy as np

np.shape(x\_train), np.shape(x\_test)



# Predictive analytics model development by comparing different Scikit-learn classification algorithms

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import train\_test\_split

from sklearn.model\_selection import KFold

from sklearn.model\_selection import cross\_val\_score

from sklearn.model\_selection import GridSearchCV

from sklearn.metrics import classification\_report

from sklearn.metrics import confusion\_matrix, ConfusionMatrixDisplay

from sklearn.metrics import accuracy\_score

from sklearn.pipeline import Pipeline

from sklearn.linear\_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis

from sklearn.naive\_bayes import GaussianNB

from sklearn.svm import SVC

from sklearn.ensemble import AdaBoostClassifier

from sklearn.ensemble import GradientBoostingClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.ensemble import ExtraTreesClassifier

models = []

models.append(('NB', GaussianNB()))

models.append(('SVM', SVC()))

models.append(('GBM', GradientBoostingClassifier()))

models.append(('RF', RandomForestClassifier()))

# evaluate each model in turn

results = []

names = []

print("Performance on Training set")

for name, model in models:

kfold = KFold(n\_splits=num\_folds,shuffle=True,random\_state=seed)

cv\_results = cross\_val\_score(model, x\_train, y\_train, cv=kfold, scoring='accuracy')

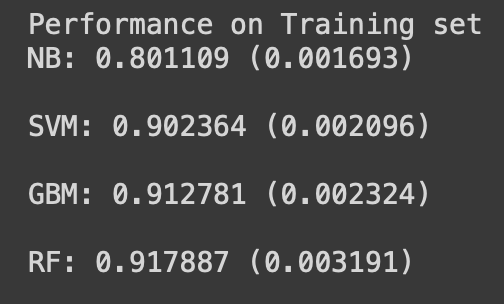
results.append(cv\_results)

names.append(name)

msg = "%s: %f (%f)" % (name, cv\_results.mean(), cv\_results.std())

msg += '\n'

print(msg)



# Compare Algorithms' Performance

import matplotlib.pyplot as plt

fig = plt.figure()

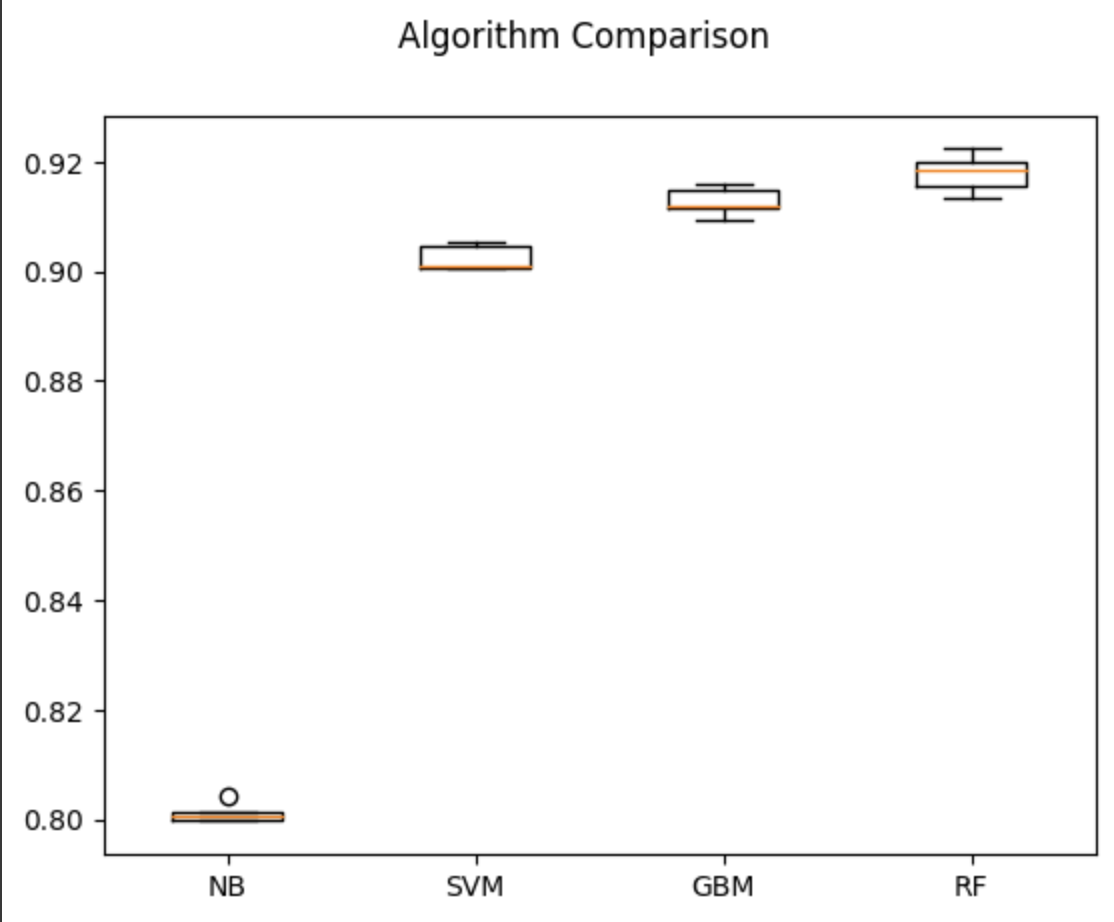
fig.suptitle('Algorithm Comparison')

ax = fig.add\_subplot(111)

plt.boxplot(results)

ax.set\_xticklabels(names)

plt.show()



#Model Evaluation by testing with independent/external test data set.

# Make predictions on validation/test dataset

models.append(('DT', DecisionTreeClassifier()))

models.append(('NB', GaussianNB()))

models.append(('SVM', SVC()))

models.append(('GBM', GradientBoostingClassifier()))

models.append(('RF', RandomForestClassifier()))

dt = DecisionTreeClassifier()

nb = GaussianNB()

gb = GradientBoostingClassifier()

rf = RandomForestClassifier()

best\_model = rf

best\_model.fit(x\_train, y\_train)

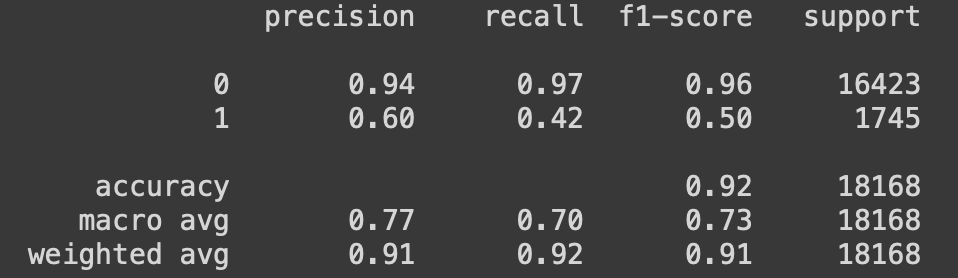
y\_pred = best\_model.predict(x\_test)

print("Best Model Accuracy Score on Test Set:", accuracy\_score(y\_test, y\_pred))



#Model Performance Evaluation Metric 1 - Classification Report

print(classification\_report(y\_test, y\_pred))



#Model Performance Evaluation Metric 2

#Confusion matrix

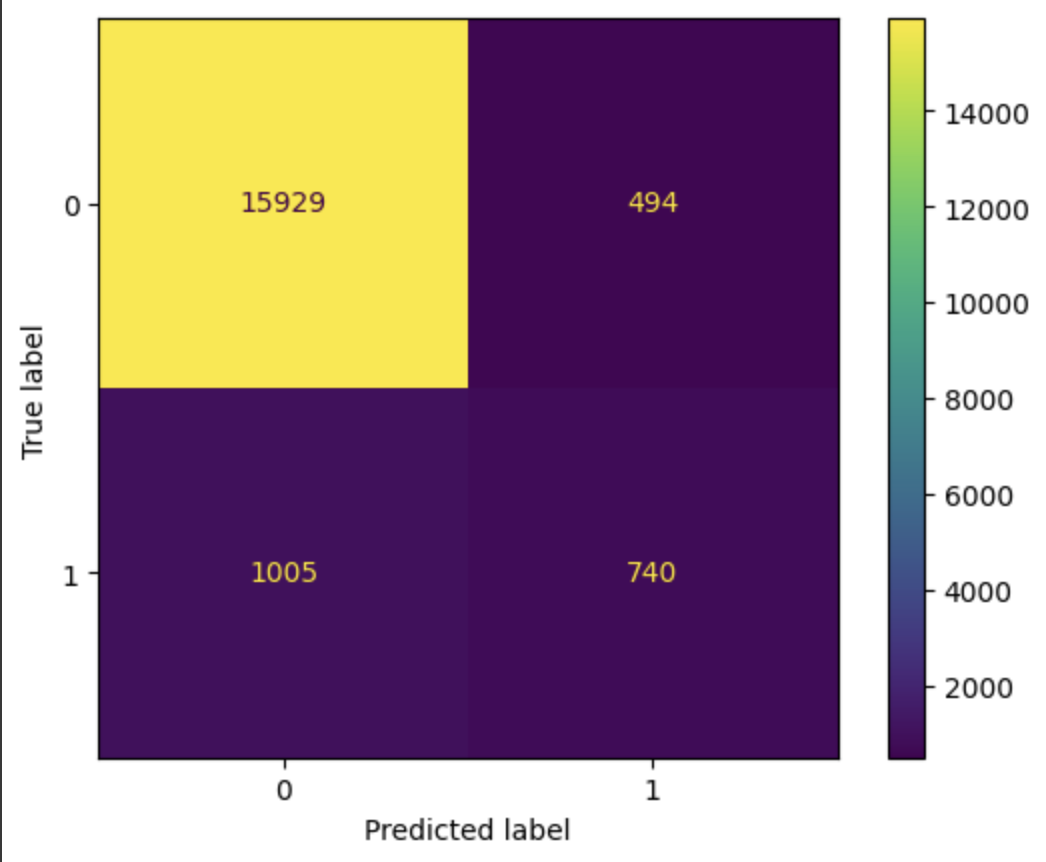
from sklearn.metrics import confusion\_matrix, ConfusionMatrixDisplay

cm = confusion\_matrix(y\_test, y\_pred)

disp = ConfusionMatrixDisplay(confusion\_matrix=cm)

disp.plot()

plt.show()



#Model Evaluation Metric 3- ROC-AUC curve

from sklearn.metrics import roc\_auc\_score

from sklearn.metrics import roc\_curve

best\_model = rf

best\_model.fit(x\_train, y\_train)

rf\_roc\_auc = roc\_auc\_score(y\_test,best\_model.predict(x\_test))

fpr,tpr,thresholds = roc\_curve(y\_test, best\_model.predict\_proba(x\_test)[:,1])

plt.figure()

plt.plot(fpr,tpr,label = 'Random Forest(area = %0.2f)'% rf\_roc\_auc)

plt.plot([0,1],[0,1],'r--')

plt.xlim([0.0,1.0])

plt.ylim([0.0,1.05])

plt.xlabel('False positive rate')

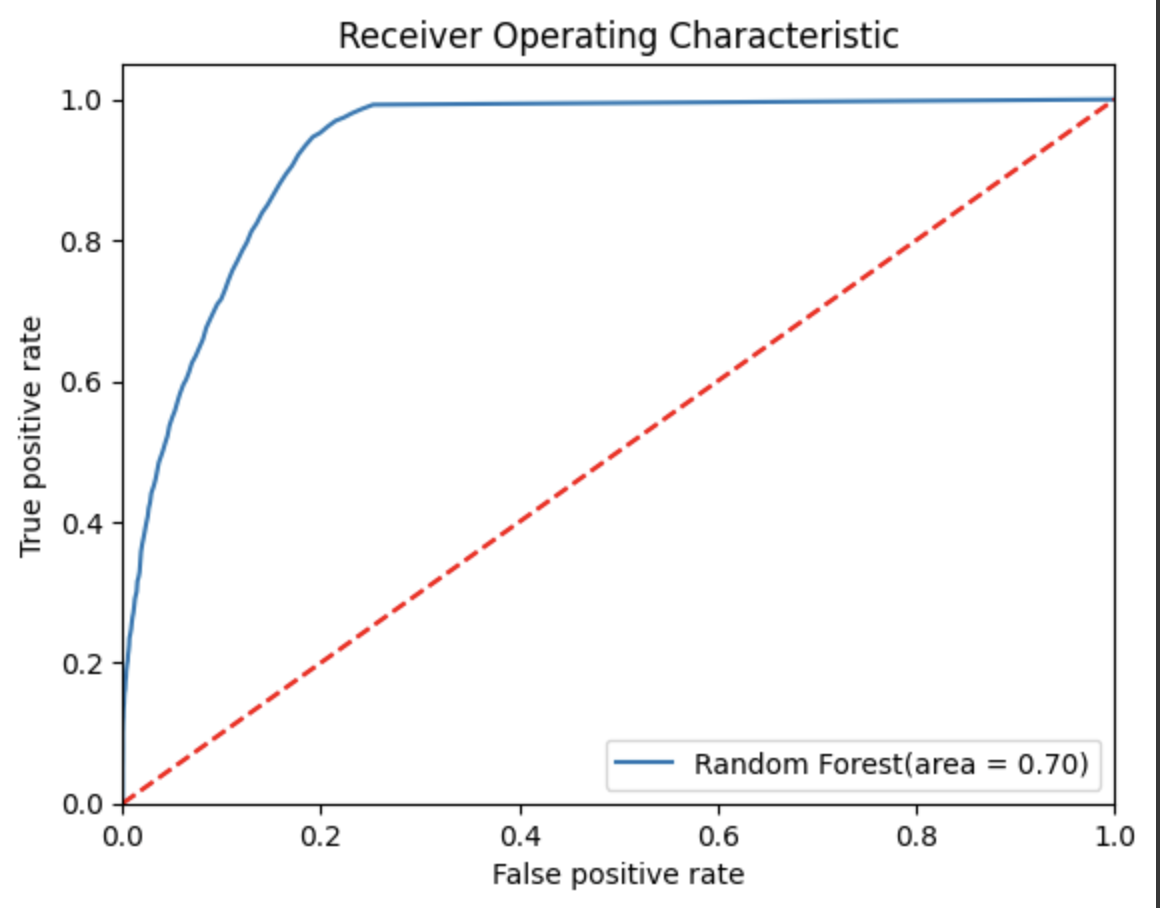
plt.ylabel('True positive rate')

plt.title('Receiver Operating Characteristic')

plt.legend(loc='lower right')

plt.savefig('LOC\_ROC')

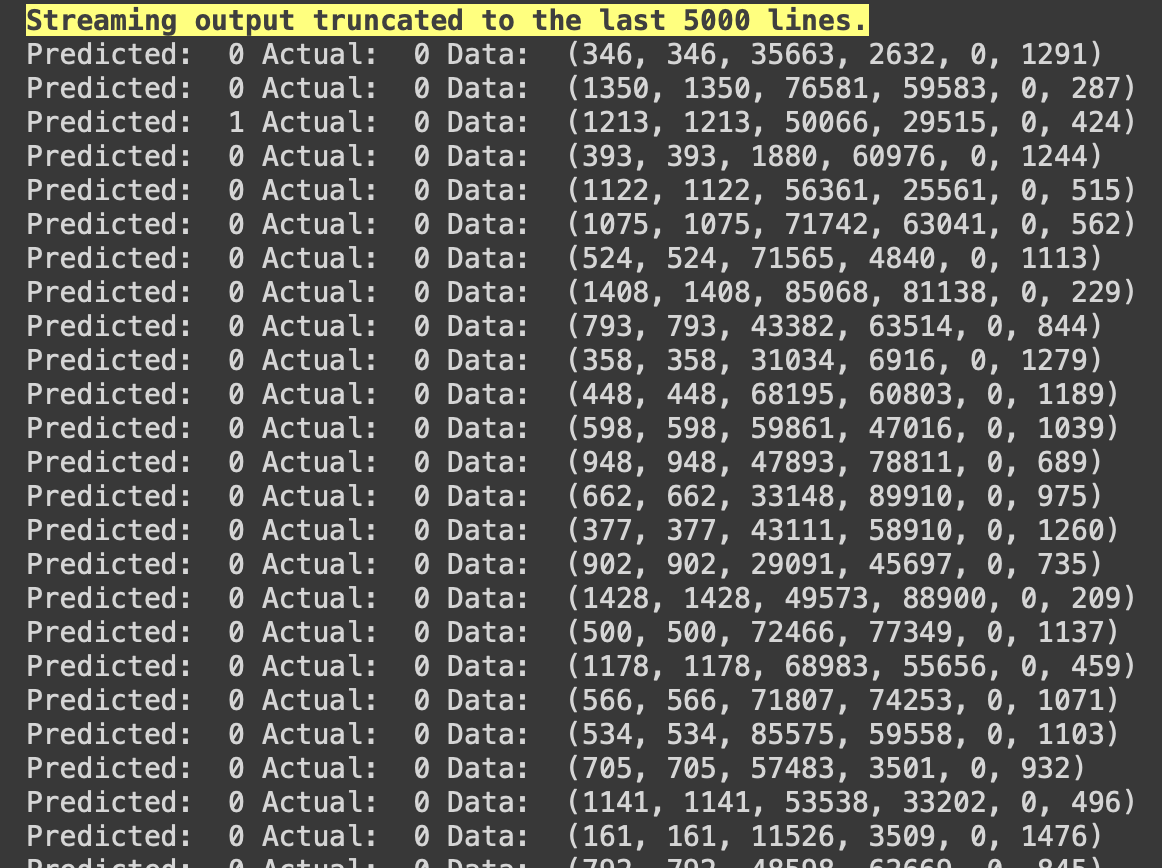
plt.show()



#Model Evaluation Metric 4-prediction report

for x in range(len(y\_pred)):

print("Predicted: ", y\_pred[x], "Actual: ", y\_test[x], "Data: ", x\_test[x],)



# Implementation and Deployment

Once the best performing AI is chosen for predicting hazardous asteroids the program is then implemented as a GUI and then deployed onto the web.

1. The program is first implemented into a GUI with the Tkinter software tool
2. Then after the program is then deployed onto the web using streamlit

As seen here: <https://drive.google.com/drive/folders/1XbsncL4CKOadM6Wta9ckDoDNSaVrhLLg?usp=share_link>

# Conclusion

The development of this program allows for safe research of this important issue, by utilising a data driven prediction platform. It bases new data off previous data samples and helps solve a modern-day problem in a risk-free environment. As seen from the outcome of this project AI can be trained to predict what asteroids are hazardous to Earth with high accuracy. With this program implemented and deployed, people tasked with protecting our planet from asteroids will have a tool for them to develop a plan for handling dangerous asteroids.

# References

published, E.H. from C.G. (2022). *What is Space?* [online] Space.com. Available at: <https://www.space.com/24870-what-is-space.html#:~:text=Space%20is%20an%20almost%20perfect>.

www.kaggle.com. (n.d.). *NASA - Nearest Earth Objects*. [online] Available at: <https://www.kaggle.com/datasets/sameepvani/nasa-nearest-earth-objects>.

NASA (2011). *What Is an Asteroid? | NASA Space Place – NASA Science for Kids*. [online] Nasa.gov. Available at: <https://spaceplace.nasa.gov/asteroid/en/>.

NASA (2019). *Asteroids*. [online] Solar System Exploration: NASA Science. Available at: <https://solarsystem.nasa.gov/asteroids-comets-and-meteors/asteroids/overview/?page=0&per_page=40&order=name+asc&search=&condition_1=101%3Aparent_id&condition_2=asteroid%3Abody_type%3Ailike>.

Nasa.gov. (2019). *Discovery Statistics*. [online] Available at: <https://cneos.jpl.nasa.gov/stats/totals.html>.

# Appendix

# Journal

### Week 9:

I have found the data set I want to use for my capstone project, and I have also came up with 5 questions I want to answer when conducting my Exploratory Data Analysis.

### Week 10:

I have completed my EDA, Predictive data analysis (PDA) and model preparation development. I have found that Random Forest (RF) was the best AI learning model for my predictions and plan to use it for my implementation and deployment.

### Week 11:

I have completed my implementation of my GUI application using Tkinter and have started deployment using streamlit and the creation of my PowerPoint presentation slides.

### Week 12:

I have competed my deployment on streamlit and have also completed my PowerPoint presentation.