8/12/2022

Capstone project

Potentially Hazardous Asteroids Prediction

Paola Yescas Martinez
BUSINESS ANALYTICS & INSIGHTS

Table of Contents

Executive Summary

The Potentially Hazardous Asteroids Prediction project consists in a deep analysis of the characteristics of the discovered asteroids in the inner solar system to identify which of those are NEO's and which variables are the most important to determine an asteroid as a Potentially Hazardous to Earth.

This project is conducted with the purpose to aware the scientific community and governments across the globe to take actions before a possible asteroid impact, considering that these collisions could cause important biological and geological changes.

The "Asteroids" dataset is a public dataset provided by Kaggle; its owner is the Jet Propulsion Laboratory (JPL) by NASA. This dataset includes information related to characteristics of asteroids discovered in the universe, such as identification number, name designated by the International Astronomical Union (IAU), comet designation prefix, geometric albedo, diameter, absolute magnitude parameter, near Earth object, among others. This information is used to identify those asteroids that are considered as potentially hazardous to Earth. An exhaustive predictive analysis has been conducted under this project by using multiple tools such as Python and SAS Enterprise Miner to develop a machine learning model for predicting whether an asteroid is potentially hazardous or not, according to the parameters mentioned below. The objective is to know which variables are the most important and the weight of each one to predict a potentially hazardous asteroid (depending on the model).

The outcome for this analysis will be the column feature called 'pha_Potentially_Hazardous_Asteroid,' considered as binary which has two values: 'Y' and 'N' (Yes/No) defining if it is dangerous to Earth or not.

Outcome

Based on this analysis, the Random Forest model has proved to be the best model. This model considered Minimm_Orbit_Intersection_Distance_au, H_Absolute_magnitude_parameter, sigma_i, sigma_n, e_Eccentricity, class_APO, i_Inclination as significant variables that can predict in a successful manner a potentially hazardous asteroid.

Introduction

Background

Asteroids, sometimes called minor planets, are rocky, airless remnants left over from the early formation of our solar system about 4.6 billion years ago. There are lots of asteroids in our solar system. Most of them live in the main asteroid belt a region between the orbits of Mars and Jupiter. ("What Is an Asteroid? | NASA Space Place – NASA Science for Kids") There are three types of asteroids C-type (carbonaceous), M-type (metallic), and S-type (silicaceous).

Asteroid belt: The most of known asteroid's orbit within the asteroid belt between the orbits of Mars and Jupiter, in relatively low-eccentricity orbits. ("Asteroid - Infogalactic: the planetary knowledge core") Is estimated that this belt contains between 1.1 and 1.9 million asteroids larger than 1 km (0.6 mi) in diameter.

Near-Earth asteroids are asteroids that have orbits that pass close to that of Earth. ("How Big of an Object Orbiting Closer to the Sun Than Earth Could Be ...") In April 2022, a total of 28,772 near-Earth asteroids were known and 878 have a diameter of one kilometer or larger. ("Asteroid — Wikipedia Republished // WIKI 2") Many asteroids have natural satellites (minor-planet moons).

The current known asteroids count is: 1,113,527.

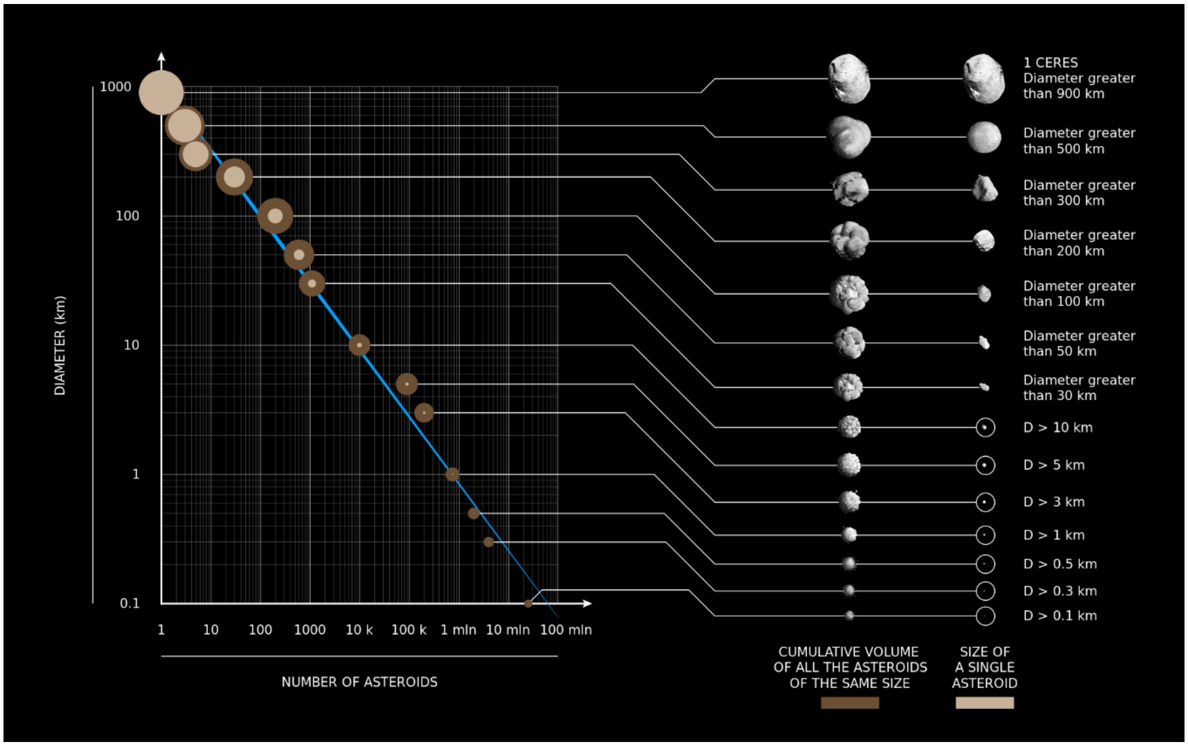


Fig1. The asteroids of the Solar System, categorized by size and number.

The Jet Propulsion Laboratory (JPL) was founded in 1930 in the Arroyo Seco, a dry canyon washes north of the Rose Bowl in Pasadena, California when Caltech professor Theodore von Kármán oversaw pioneering work in rocket propulsion.

1936:

- The Caltech group's first tests of an alcohol-fueled rocket motor. (“History - Robotic Space Exploration”)
- The US Army helped Caltech acquire land in the Arroyo Seco for test pits and temporary workshops. (“History - Robotic Space Exploration”)

1943:

- Was named for the first time as “Jet Propulsion Laboratory.”

1944:

- Technology involved aerodynamics and propellant chemistry and evolve into tools for space flight, secure communications, spacecraft navigation and control, planetary exploration.

1945:

- A supersonic wind tunnel was developed and an array of environmental test technologies and create a new discipline called the systems engineering.

1957:

- In the first space experiment discovered belts of trapped radiation encircling Earth. ("Blog - ET101 Life on other planets")

1958:

- "JPL vaulted the U.S. into space and prompted the formation of NASA." ("History - Robotic Space Exploration") JPL was transferred from Army authority to that of the new civilian space agency. ("INTRODUCTION - NASA")

1960:

- The development of a robotic spacecraft to explore other worlds started. This began with NASA's Apollo astronaut lunar landings.

1966 – 1968:

- More than five spacecrafts were launched to discover the space.

1976:

- Started Mars biological experiments.

NASA has concentrated its dwindling post-Apollo budget on building the Space Shuttle, and funding for planetary exploration has decreased substantially.

JPL is the owner of the Hubble Space Telescope that is one of the most important telescopes in the world that has taken the pictures and images that the entire universe had seen.

JPL has worked with NASA in different space projects and missions, from orbits research to spacecraft explorations planets such as Mars, Venus, Saturn, and other technological development.

A brief history about NASA

The National Aeronautics and Space Administration (NASA) is an independent agency of the United States federal government in charge of the civil space program, aeronautical research and space research and it was established in 1958 succeeding the National Advisory Committee for Aeronautics (NACA).

The following are NASA facilities around the US focused on different aerospace activities:

Inherited from NACA:

- Langley Research Center (LaRC), located in Hampton, Virginia is focused on aeronautical research.

- Ames Research Center (ARC) located in California's Silicon Valley is focused on wind-tunnel research on the aerodynamics of propeller-driven aircraft, research and technology in aeronautics, spaceflight, and information technology, leadership in astrobiology, small satellites, robotic lunar exploration, intelligent/adaptive systems, and thermal protection. (“NASA facilities | National Aeronautics and Space ... - Fandom”)

- George W. Lewis Research Center focused on air-breathing and in-space propulsion and cryogenics, communications, power energy storage and conversion, microgravity sciences, and advanced materials.

- Hugh L. Dryden Flight Research Facility (AFRC), located inside Edwards Air Force Base, is the home of the Shuttle Carrier Aircraft (SCA). (“NASA explained”) (“NASA - Wikipedia”)

Transferred from the Army:

- The Jet Propulsion Laboratory (JPL), located in Los Angeles County, CA. JPL is managed by the nearby California Institute of Technology (Caltech) and is focused on the construction and operation of robotic planetary spacecraft, conducts Earth-orbit and astronomy missions, responsible for operating NASA's Deep Space Network. ("NASA facilities - Wikipedia") This laboratory provides the data of this analytics study.

- George C. Marshall Space Flight Center (MSFC), located on the Redstone Arsenal near Huntsville, Alabama, its the lead center for International Space Station (ISS) design and assembly (Saturn V rocket and Spacelab), payloads and related crew training.

Built by NASA:

- The Goddard Space Flight Center (GSFC), located in Greenbelt, Maryland, it is the largest combined organization of scientists and engineers in the US focused on increasing knowledge of the Earth, the Solar System, and the Universe via observations from space.

- John C. Stennis Space Center, located in Mississippi–Louisiana. It is used for rocket testing by over thirty local, state, national, international, private, and public companies, and agencies.

- Manned Spacecraft Center (MSC), located in Houston, Texas is the NASA center for human spaceflight training, research, and flight control.

- John F. Kennedy Space Center (KSC), located in Florida, was named the "Launch Operations Center" nowadays it continues to manage and operate unmanned rocket launch facilities for America's civilian space program from three pads at Cape Canaveral.

Subordinate facilities:

- Wallops Flight Facility in Wallops Island, Virginia.
- Michoud Assembly Facility in New Orleans, Louisiana.
- White Sands Test Facility in Las Cruces, New Mexico.
- Deep Space Network stations in Barstow, California.
- Madrid, Spain
- Canberra, Australia.

Problem Statement

Identification of the characteristics that could determine an asteroid as potential hazardous to Earth and identify those that are near that could collide with Earth or another near space object or planet that could affect us as a whole, addressing distinctive characteristics of each asteroid such as albedo, diameter, absolute magnitude, distance, inclination, orbit intersection and so on.

Scientists will aware Governments across the globe to take actions before a possible asteroid impact, considering that collisions between these objects and the Earth have important agents of biological and geological change.

Objectives & Measurement

With the analysis of the "Asteroids" dataset that will be conducted in this project will be known the asteroid characteristics that highly define a potentially hazardous to Earth and do some calculations to estimate the date when that asteroid will be nearest the Earth, the Moon, the International Space Station, satellites, and spacecrafts. With this information, scientists, astrophysicists, and employees in charge of different areas of the Space Agencies could take actions and give advice to governments considering the size, velocity and other factors of the asteroid that could damage Earth in some way to take actions with the people and ecosystems before any biological issue, climate change or catastrophe could happen on Earth whether an asteroid gets closer or in the worse scenario collides with Earth or other near space objects. There are substantial bunkers in various locations of the planet where people can hide for some of these kinds of catastrophe.

Assumptions

1. Only 5% of the universe is known, noting that at the moment this answer is unclear.
2. Around sixty asteroids impacted with Earth in the past.
3. What could happen if an asteroid collides with moon?
    - Could be extreme climate variations on the planet.
    - "It could even plunge Earth into a new ice age within a few hundred years." ("What Happens to Earth If an Asteroid Destroys the Moon?")

- Some of the lunar fragments could hit Earth and they could be as big as the asteroid that wiped out the dinosaur sixty-six million years ago, they would release less energy on the planet.

4. Asteroids one kilometer or larger in diameter are likely to impact our planet once every 100,000 years. In addition, comets are less dangerous and have a chance to hit our planet once every half a million years.

Data sources

Data Set Introduction

The “Asteroids” dataset is a public dataset provided by Kaggle; its owner is the Jet Propulsion Laboratory (JPL) by NASA. The dataset contains 958,524 entries with 45 columns including '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', among others.

Exclusions

The following variables are excluded from the model:

EXCLUSION	VARIABLE	DEFINITION
Irrelevant variables	Id	Object internal database ID
	orbit_id	Orbit solution ID
	Equinox	Equinox of reference frame
	full_name	Object full name/designation
	Pdes	Object primary designation
Variables with more than the 50% of missing values	Name	Object name IAU (International Astronomical Union)
	Prefix	Comet designation prefix
	Diameter	Object diameter (from equivalent sphere) (km)
	Albedo	Albedo is ratio of the light received by a body to the light reflected by that body. Albedo values range from 0 (pitch black) to 1 (perfect reflector). (“Albedo - NASA”)

	diameter_sigma	1-sigma uncertainty in object diameter (km), values with probability of 68%
Redundant variables	Epoch	Epoch of osculation in Julian day form. Epoch of osculation changes every 200 days (e, a, q, i, om, w, ma).
	epoch_mjd	Epoch of osculation in modified Julian day form (TDB)
	Per	Sidereal orbital period (days). The sidereal period is the amount of time that it takes an object to make a full orbit. ("Orbital period - Wikipedia")
	per_y	Sidereal orbital period (years)
	Q	Perihelion distance (au). An orbit's closest point to the Sun.
	W	Argument of perihelion (deg). Angle in the orbit plane between the ascending node and the perihelion point.
	Ad	Aphelion distance (au). An orbit's farthest point to the Sun.
	moid_ld	Minimum Orbit Intersection Distance (Lunar Distance).
	Tp	Time of perihelion passage (TDB).
	sigma_q	Perihelion distance (1-sigma uncertainty) (au), values with probability of 68%
	sigma_w	Argument of perihelion (1-sigma uncertainty) (deg), values with probability of 68%
	sigma_tp	Time of perihelion passage (1-sigma uncertainty) (TDB), values with probability of 68%
	sigma_per	Sidereal orbital period (1-sigma uncertainty) (d), values with probability of 68%

	sigma_ad	Aphelion distance (1-sigma uncertainty) (au), values with probability of 68%

Data Dictionary

Variable	Name	Description
id	Object internal database ID	Internal ID
spkid	Object primary SPK-ID	Type of designation (NASA)
full_name	Object full name/designation IAU*	Full name designated
pdes	Object primary designation IAU*	Primary designation
name	Object IAU* name	Name designation
prefix	Comet designation prefix	Comet designation prefix
neo	Near-Earth Object, flag (Y/N)	An asteroid or comet with a perihelion distance less than or equal to 1.3 au*, 99% of NEOs are asteroids
pha	Potentially Hazardous Asteroid, flag (Y/N)	Potentially Hazardous Asteroid to Earth
H	Absolute magnitude parameter	An asteroid's absolute magnitude is the visual magnitude an observer would record if the asteroid were placed 1 Astronomical Unit (au*) away, and 1 au* from the Sun and at a zero-phase angle ("Glossary - NASA")

diameter	Object diameter (km)	Object diameter (from equivalent sphere)
albedo	Geometric albedo	Albedo is ratio of the light received by a body to the light reflected by that body. Albedo values range from 0 (pitch black) to 1 (perfect reflector) ("Glossary - NASA")
diameter_sigma	1-sigma uncertainty in object diameter (km)	1-sigma uncertainty in object diameter (km), 68% prob
orbit_id	Orbit solution ID	Orbit solution ID
epoch	Epoch of osculation in Julian day form	Epoch of osculation changes every 200 days (e, a, q, i, om, w, ma)
epoch_mjd	Epoch of osculation in modified Julian day form (TDB*)	Epoch of osculation changes every 200 days (e, a, q, i, om, w, ma)
epoch_cal	Epoch of osculation in calendar date/time form (TDB*)	Epoch of osculation changes every 200 days (e, a, q, i, om, w, ma)
equinox	Equinox of reference frame	J200 (standard equinox) Julian epoch
e	Eccentricity	An orbital parameter describing the eccentricity of the orbit ellipse. Eccentricity e is the ratio of half the distance between the foci c to the semi-major axis a: e=c/a. For example, an orbit with e=0 is

		circular, e=1 is parabolic, and e between 0 and 1 is elliptic
a	Semi-major axis (au*)	One half of the major axis of the elliptical orbit; also, the mean distance from the Sun ("Glossary - Center for NEO Studies")
q	Perihelion distance (au*)	An orbit's closest point to the Sun
i	Inclination (deg*)	Angle between the orbit plane and the ecliptic plane
om	Longitude of the ascending node (deg*)	"Angle in the ecliptic plane between the inertial-frame x-axis and the line through the ascending node" ("Node - JPL Solar System Dynamics")
w	Argument of perihelion (deg*)	"Angle in the orbit plane between the ascending node and the perihelion point" ("Glossary - NASA")
ma	Mean anomaly (deg*)	"The product of an orbiting body's mean motion and time past perihelion passage" ("Ma - JPL Solar System Dynamics")
ad	Aphelion distance (au*)	An orbit's farthest point to the Sun
n	Mean motion (deg/day)	The angular speed required for a body to make one orbit around an ideal ellipse with a specific semi-

		major axis. It is equal to 2 times pi (π) divided by the orbital period ("Glossary - N")
tp	Time of perihelion passage (TDB*)	"The time at which an object is at perihelion (its closest distance to the sun)." ("Tp - JPL Solar System Dynamics") The barycenter is the center of mass of a system of bodies, e.g., the center of mass of the solar system or the Earth-Moon system
tp_cal	Time of perihelion passage, calendar (TDB*)	"The time at which an object is at perihelion (its closest distance to the sun)." ("Tp - JPL Solar System Dynamics") The barycenter is the center of mass of a system of bodies, e.g., the center of mass of the solar system or the Earth-Moon system
per	Sidereal orbital period (day)	"The sidereal period is the amount of time that it takes an object to make a full orbit" ("Orbital period – Wikipedia)
per_y	Sidereal orbital period (year)	"The sidereal period is the amount of time that it takes an object to make a full orbit" ("Orbital period - Wikipedia")

moid	Minimum Orbit Intersection Distance (au*)	MOID is a measure used in astronomy to assess potential close approaches and collision risks between astronomical objects. "It is defined as the distance between the closest points of the osculating orbits of two bodies" ("terminology - Space Exploration Stack Exchange")
moid_ld	Minimum Orbit Intersection Distance (LD*)	MOID is a measure used in astronomy to assess potential close approaches and collision risks between astronomical objects. "It is defined as the distance between the closest points of the osculating orbits of two bodies" ("terminology - Space Exploration Stack Exchange")
sigma_e	Eccentricity (1-sigma uncertainty)	Eccentricity (1-sigma uncertainty, 68% prob)
sigma_a	Semi-major axis (1-sigma uncertainty) (au*)	Semi-major axis (1-sigma uncertainty, 68% prob)
sigma_q	Perihelion distance (1-sigma uncertainty) (au*)	Perihelion distance (1-sigma uncertainty, 68% prob)
sigma_i	Inclination (1-sigma uncertainty) (deg)	Inclination (1-sigma uncertainty, 68% prob)

sigma_om	Longitude of the ascending node (1-sigma uncertainty) (deg)	Longitude of the ascending node (1-sigma uncertainty, 68% prob)
sigma_w	Argument of perihelion (1-sigma uncertainty) (deg)	Argument of perihelion (1-sigma uncertainty, 68% prob)
sigma_ma	Mean anomaly (1-sigma uncertainty) (deg)	Mean anomaly (1-sigma uncertainty, 68% prob)
sigma_ad	Aphelion distance (1-sigma uncertainty) (au*)	Aphelion distance (1-sigma uncertainty, 68% prob)
sigma_n	Mean motion (1-sigma uncertainty) (deg/day)	Mean motion (1-sigma uncertainty, 68% prob)
sigma_tp	Time of perihelion passage (1-sigma uncertainty) (TDB*)	Time of perihelion passage (1-sigma uncertainty, 68% prob)
sigma_per	Sidereal orbital period (1-sigma uncertainty) (day)	Sidereal orbital period (1-sigma uncertainty, 68% prob)
class	Orbit classification	The path followed by a celestial body in inertial space
rms	Normalized RMS (Root Mean Squared) (arcsec*)	Normalized RMS (Root Mean Squared) of orbit fit

*IAU= International Astronomical Union, is a nongovernmental organisation with the objective of advancing astronomy in all aspects, including promoting astronomical research, outreach, education, and development through global cooperation. ("International Astronomical Union - Wikipedia")

*au= Astronomical unit defined by IAU as exactly 149,597,870,700meters. Its approximately the average distance between earth and the sun (about 150 billion meters). ("Glossary - NASA")

*LD= The term LD (Lunar Distance) refers to the average distance between the Earth and Moon. For data reported on this site, is used a mean semimajor axis for the moon of 384,400 km (~.002570 au) to define one LD.

*TDB= Barycentric Dynamical Time, TCB progresses faster at a differential rate of about 0.5 second/year.

*arcsec= arcsecond, denoted by the symbol ″, is1/60 of an arcminute,1/3,600 of a degree, 1/1,296,000 of a turn, and π/648,000 (about 1/206,264.8) of a radian.

Data Exploration

Data exploration is where a data analyst uses visual exploration to understand a dataset, which variables, values, characteristics contain the data. Some characteristics can include size or amount of data, completeness of the data, correctness of the data, possible relationships amongst data elements or files/tables in the data.

Data Exploration Techniques

Python:

Python is a high-level, interpreted, general-purpose programming language. It supports multiple programming paradigms, including structured, object-oriented, and functional programming. ("Python Definition - Harbourfront Technologies") For Data Analytics, is an easy platform to learn, the programming language is flexible, it has a lot of libraries for numerical computation, data manipulation, graphics, data visualization (build plots).

- Import libraries:

```
import pandas as pd
import matplotlib.pylab as plt
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

%matplotlib inline

from pathlib import Path
```

Fig 2. Python libraries

- Read the data:

```
df= pd.read_csv('Asteroid Dataset.csv')
df.head()
```

```
/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshell.py:2882: DtypeWarning: Columns (3,4,5) have mixed types.Specify dtype option on import or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

	id	spkid	full_name	pdes	name	prefix	neo	pha	H	diameter	...	sigma_i	sigma_om	sigma_w	sigma_ma	sigma_ad	sigma_n	sigma_tp	sigma_per	class	
0	a0000001	2000001	1 Ceres	1	Ceres	NaN	N	N	3.40	939.400	...	4.610000e-09	6.170000e-08	6.620000e-08	7.820000e-09	1.110000e-11	1.200000e-12	3.780000e-08	9.420000e-09	MBA	0.43
1	a0000002	2000002	2 Pallas	2	Pallas	NaN	N	N	4.20	545.000	...	3.470000e-06	6.270000e-06	9.130000e-06	8.860000e-06	4.960000e-09	4.650000e-10	4.080000e-05	3.680000e-06	MBA	0.35
2	a0000003	2000003	3 Juno	3	Juno	NaN	N	N	5.33	246.596	...	3.220000e-06	1.660000e-05	1.770000e-05	8.110000e-06	4.360000e-09	4.410000e-10	3.530000e-05	3.110000e-06	MBA	0.33
3	a0000004	2000004	4 Vesta	4	Vesta	NaN	N	N	3.00	525.400	...	2.170000e-07	3.880000e-07	1.790000e-07	1.210000e-06	1.650000e-09	2.610000e-10	4.100000e-06	1.270000e-06	MBA	0.39
4	a0000005	2000005	5 Astraea	5	Astraea	NaN	N	N	6.90	106.699	...	2.740000e-06	2.890000e-05	2.980000e-05	8.300000e-06	4.730000e-09	5.520000e-10	3.470000e-05	3.490000e-06	MBA	0.52

5 rows × 45 columns

Fig 3. First five rows of the dataset

- Size of the dataset:

```
df.shape
```

```
(958524, 45)
```

```
df.columns
```

```
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')
```

Fig 4. Size of the dataset, number of columns and rows. Variables of each column.

- Description of the variables contained in the dataset:

```
df.describe()
```

	spkid	H	diameter	albedo	diameter_sigma	epoch	epoch_mjd	epoch_cal	e	a	...	sigma_q	sigma_i
count	9.585240e+05	952261.000000	136209.000000	135103.000000	136081.000000	9.585240e+05	958524.000000	9.585240e+05	958524.000000	958524.000000	...	9.386020e+05	9.386020e+05
mean	3.810114e+06	16.906411	5.506429	0.130627	0.479184	2.458869e+06	58868.781950	2.019693e+07	0.156116	2.902143	...	1.983462e+01	1.168449e+00
std	6.831541e+06	1.790405	9.425164	0.110323	0.782895	7.016716e+02	701.671573	1.930354e+04	0.092643	39.719503	...	2.905651e+03	1.282231e+02
min	2.000001e+06	-1.100000	0.002500	0.001000	0.000500	2.425052e+06	25051.000000	1.927062e+07	0.000000	-14702.447870	...	1.960000e-11	4.610000e-09
25%	2.239632e+06	16.100000	2.780000	0.053000	0.180000	2.459000e+06	59000.000000	2.020053e+07	0.092193	2.387835	...	1.460000e-07	6.100000e-06
50%	2.479262e+06	16.900000	3.972000	0.079000	0.332000	2.459000e+06	59000.000000	2.020053e+07	0.145002	2.646969	...	2.270000e-07	8.690000e-06
75%	3.752518e+06	17.714000	5.765000	0.190000	0.620000	2.459000e+06	59000.000000	2.020053e+07	0.200650	3.001932	...	6.580000e-07	1.590000e-05
max	5.401723e+07	33.200000	939.400000	1.000000	140.000000	2.459000e+06	59000.000000	2.020053e+07	1.855356	33488.895950	...	1.020000e+06	5.533000e+04

Fig 5. Description of the variables (count, mean, std, among others)

- Information of the data type, null values, which contain each column:

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 958524 entries, 0 to 958523
Data columns (total 45 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   id              958524 non-null  object 
 1   spkid           958524 non-null  int64  
 2   full_name       958524 non-null  object 
 3   pdes            958524 non-null  object 
 4   name            22064 non-null   object 
 5   prefix          18 non-null      object 
 6   neo             958520 non-null  object 
 7   pha             938603 non-null  object 
 8   H               952261 non-null  float64
 9   diameter        136209 non-null  float64
 10  albedo          135103 non-null  float64
 11  diameter_sigma  136081 non-null  float64
 12  orbit_id        958524 non-null  object 
 13  epoch           958524 non-null  float64
 14  epoch_mjd       958524 non-null  int64  
 15  epoch_cal       958524 non-null  int64  
 16  equinox         958524 non-null  object 
 17  e               958524 non-null  float64
 18  a               958524 non-null  float64
 19  q               958524 non-null  float64
 20  i               958524 non-null  float64
 21  om              958524 non-null  float64
 22  w               958524 non-null  float64
 23  ma              958523 non-null  float64
 24  ad              958520 non-null  float64
 25  n               958524 non-null  float64
 26  tp              958524 non-null  float64
 27  tp_cal          958524 non-null  float64
 28  per             958520 non-null  float64
 29  per_y           958523 non-null  float64
 30  moid            938603 non-null  float64
 31  moid_ld         958397 non-null  float64
 32  sigma_e         938602 non-null  float64
 33  sigma_a         938602 non-null  float64
 34  sigma_q         938602 non-null  float64
 35  sigma_i         938602 non-null  float64
 36  sigma_om        938602 non-null  float64
 37  sigma_w         938602 non-null  float64
 38  sigma_ma        938602 non-null  float64
 39  sigma_ad        938598 non-null  float64
 40  sigma_n         938602 non-null  float64
 41  sigma_tp        938602 non-null  float64
 42  sigma_per       938598 non-null  float64
 43  class           958524 non-null  object 
 44  rms             958522 non-null  float64
dtypes: float64(32), int64(3), object(10)
memory usage: 329.1+ MB
```

Fig 6. Information for each column (data type, presence of null values)

- Exploration of missing values of each variable:

```
df.isnull().sum() / len(df)*100

id                 0.000000
spkid              0.000000
full_name          0.000000
pdes               0.000000
name              97.698128
prefix            99.998122
neo                0.000417
pha                2.078300
H                  0.653400
diameter          85.789714
albedo            85.905100
diameter_sigma    85.803068
orbit_id           0.000000
epoch              0.000000
epoch_mjd          0.000000
epoch_cal          0.000000
equinox            0.000000
e                  0.000000
a                  0.000000
q                  0.000000
i                  0.000000
om                 0.000000
w                  0.000000
ma                 0.000104
ad                 0.000417
n                  0.000000
tp                 0.000000
tp_cal             0.000000
per                0.000417
per_y              0.000104
moid               2.078300
moid_ld            0.013250
sigma_e            2.078404
sigma_a            2.078404
sigma_q            2.078404
sigma_i            2.078404
sigma_om           2.078404
sigma_w            2.078404
sigma_ma           2.078404
sigma_ad           2.078821
sigma_n            2.078404
sigma_tp           2.078404
sigma_per          2.078821
class              0.000000
rms                0.000209
dtype: float64
```

Fig 7. Percentage of missing values per variable.

- Identify different values of a variable:

```
print(df['equinox'].unique())
print(df['class'].unique())

['J2000']
['MBA' 'OMB' 'MCA' 'AMO' 'IMB' 'TJN' 'CEN' 'APO' 'ATE' 'AST' 'TNO' 'IEO'
 'HYA']
```

Fig 8. Identification of different values in an object variable.

- Exploring the target variable:

```
print(len(df[df['pha'] == 'N']))
print(len(df[df['pha'] == 'Y']))
print(len(df[df['pha'] == 'Y'])/ len(df[df['pha'] == 'N']) * 100)

936537
2066
0.22059993358511196
```

Fig 9. Identification the number of each value on the target variable.

- Check the correlation between variables:

```
[ ] correlation= df.corr().round(1)
```

```
[ ] import matplotlib.pyplot as plt
    plt.figure(figsize = (15,10))

    sns.heatmap(correlation, vmin=-1, vmax=1, annot=True, cmap="Spectral", linewidths=.8)
```

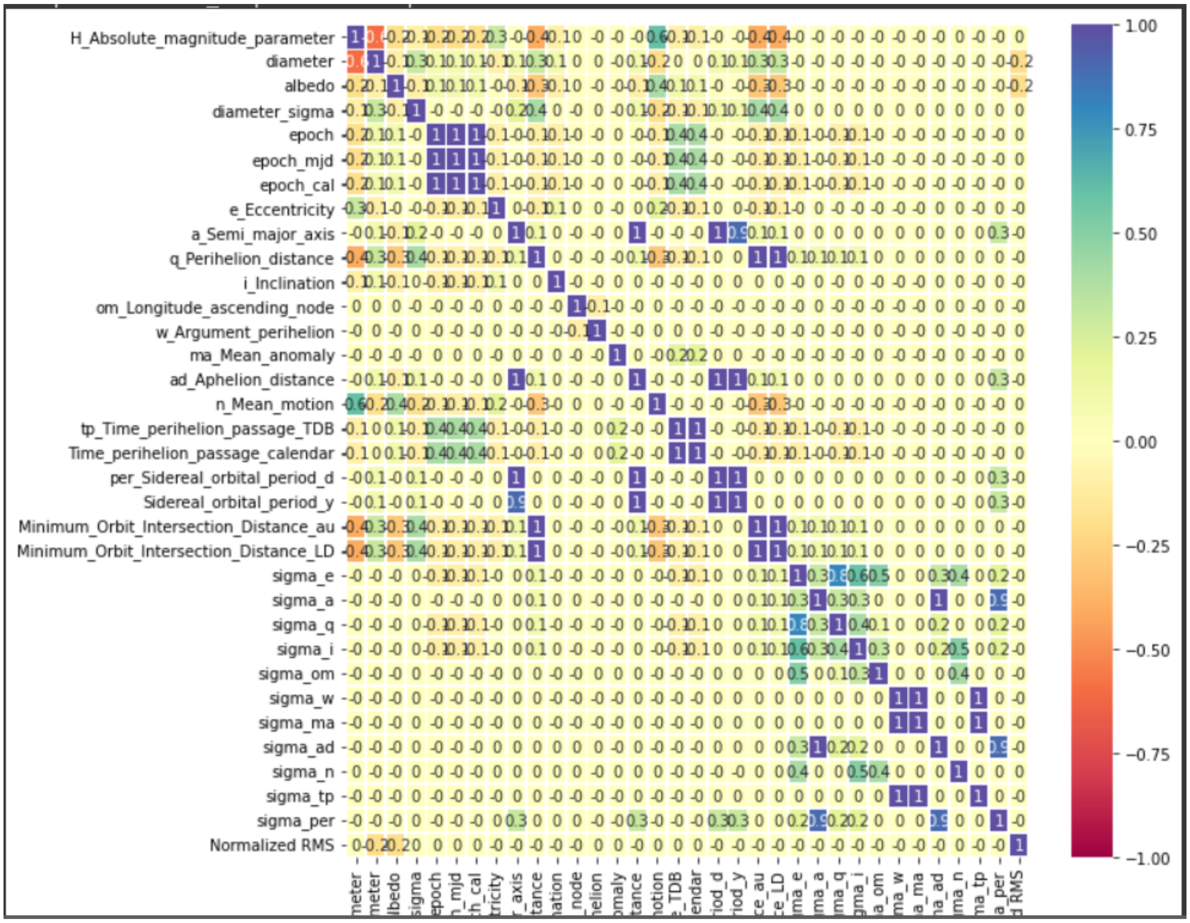


Fig 10. Heatmap of a Correlation table

SAS Enterprise Miner:

It is a powerful tool that streamlines data mining and use analytics to build predictive and descriptive models. SAS Enterprise Miner aids in the analysis of complicated data, the discovery of trends, and the development of models so that fraud may be detected more quickly, resource demands can be forecasted, and customer attrition can be reduced.

- Import the data:

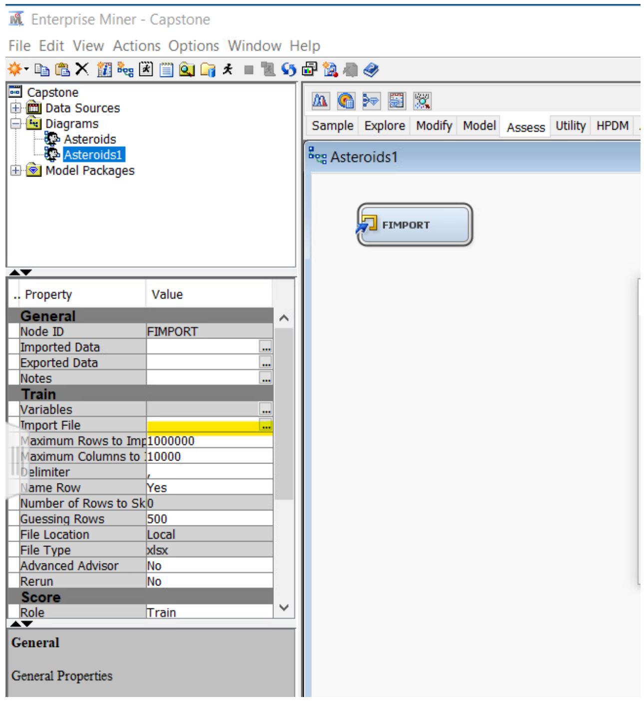


Fig 11. Import the dataset in SAS Enterprise Miner.

- Data exploration:

Sample Properties

Property	Value
Rows	958524
Columns	21
Library	EMWS1
Member	FIMPORT_DATA
Type	DATA
Sample Method	Random
Fetch Size	Max
Fetched Rows	20000
Random Seed	12345

Sample Statistics

Obs #	Variabl...	Label	Type	Percen...	Minimum	Maximum	Mean	Numbe...	Mode ...	Mode
1	Near Ear...	Near Ear...	CLASS	0	.	.	.	2	97.635	N
2	Potentiall...	Potentiall...	CLASS	2.12	.	.	.	3	97.665	N
3	class	class	CLASS	0	.	.	.	12	89.335	MBA
4	H Absolu...	H Absolu...	VAR	0.635	3.6	31	16.90862	.	.	
5	Minimum...	Minimum...	VAR	2.12	.0000254	44.87	1.414503	.	.	
6	Normaliz...		VAR	0	0.012883	7.0376	0.557231	.	.	
7	Time per...	Time per...	VAR	0	18821212	21730131	20195678	.	.	
8	a Semi ...	a Semi ...	VAR	0	0.671604	789.4764	2.89126	.	.	
9	e Eccent...	e Eccent...	VAR	0	.0000524	0.993775	0.1559	.	.	
10	epoch cal	epoch cal	VAR	0	19600927	20200531	20196741	.	.	
11	i Inclinati...	i Inclinati...	VAR	0	0.02587	158.5566	8.981551	.	.	
12	ma Mea...	ma Mea...	VAR	0	0.00398	397.4935	177.2318	.	.	
13	n Mean ...	n Mean ...	VAR	0	.0000444	1.790746	0.236641	.	.	
14	om Long...	om Long...	VAR	0	0.005629	359.9954	168.3643	.	.	
15	sigma a	sigma a	VAR	2.12	2.74E-10	45403	6.231341	.	.	
16	sigma e	sigma e	VAR	2.12	2.78E-9	21099	2.049534	.	.	
17	sigma i	sigma i	VAR	2.12	2.19E-6	2664.3	0.645793	.	.	
18	sigma ma	sigma ma	VAR	2.12	1.59E-6	3.08E10	2318164	.	.	
19	sigma n	sigma n	VAR	2.12	5.67E-11	71.902	0.035145	.	.	
20	sigma om	sigma om	VAR	2.12	1.85E-6	8599.6	3.192094	.	.	
21	spkid	spkid	VAR	0	2000032	54017230	3857641	.	.	

Fig 12. Characteristics of the variables in a random selection.

Variable	Role	Mean	Deviation	Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
H_Absolute_magnitude_parameter	INPUT	14.92473	1.225642	100000	0	-1.1	15.1	22.7	-1.60105	7.16522
Minimum_Orbit_Intersection_Dista	INPUT	1.338382	1.138567	100000	0	0.000027	1.22147	42.2656	25.64743	776.448
Normalized_RMS	INPUT	0.539199	0.047942	99999	1	0.078624	0.54205	0.92905	-0.52666	1.598976
a_Semi_major_axis	INPUT	2.710406	2.471195	100000	0	0.642338	2.598747	564.6993	133.1815	27589.39
e_Eccentricity	INPUT	0.13826	0.068908	100000	0	0.001142	0.133455	0.963287	1.203373	6.281122
i_Inclination	INPUT	7.821722	5.672079	100000	0	0.022056	6.357662	160.4309	1.66635	9.842906
ma_Mean_anomaly	INPUT	180.2442	103.6216	100000	0	0.004635	180.4294	359.9987	-0.00683	-1.19747
n_Mean_motion	INPUT	0.236167	0.051391	100000	0	0.000073	0.235265	1.914515	3.343252	68.13137
om_Longitude_ascending_node	INPUT	167.9403	101.9061	100000	0	0.001562	159.089	359.9967	0.210604	-1.07532
sigma_a	INPUT	0.00005	0.008323	99999	1	1.03E-11	1.38E-8	2.3695	255.1628	68966.2
sigma_e	INPUT	2.871E-7	0.000016	99999	1	4.82E-12	4.31E-8	0.003144	129.5133	20826.55
sigma_i	INPUT	5.539E-6	0.000023	99999	1	4.61E-9	4.86E-6	0.003226	81.66945	8192.579
sigma_ma	INPUT	0.000312	0.057458	99999	1	7.82E-9	0.000021	18.008	308.2277	96495.93
sigma_n	INPUT	3.611E-9	7.548E-8	99999	1	1.2E-12	1.87E-9	9.17E-6	68.66587	5976.502
sigma_om	INPUT	0.000071	0.000801	99999	1	6.17E-8	0.00004	0.20168	214.8994	49827.99

Fig 13. Characteristics of the variables in a random selection, including skewness and kurtosis.

Data Cleansing

Python:

- Drop irrelevant variables:

```
df= df.drop(['name','prefix','id','spkid','full_name', 'equinox','orbit_id','pdes'], axis=1)
```

Fig 14. Drop irrelevant variables in python.

- Rename actual column names to have a better understanding:

```
[ ]  df= df.rename(columns={'neo':'Near_Earth_Object', 'pha': 'Potentially_Hazardous_Asteroid', 'H': 'H_Absolute_magnitude_parameter', 'e':'e_Eccentricity', 'a':'a_Semi_major_axis',
                        'q':'q_Perihelion_distance', 'i':'i_Inclination', 'om':'om_Longitude_ascending_node', 'w':'w_Argument_perihelion', 'ma': 'ma_Mean_anomaly', 'ad':'ad_Aphelion_di
                        'n':'n_Mean_motion','tp':'tp_Time_perihelion_passage_TDB', 'tp_cal':'Time_perihelion_passage_calendar', 'per':'per_Sidereal_orbital_period_d',
                        'per_y':'Sidereal_orbital_period_y', 'moid':'Minimum_Orbit_Intersection_Distance_au', 'moid_ld':'Minimum_Orbit_Intersection_Distance_LD', 'rms':'Normalized RMS'
```

Fig 15. Rename column names of diverse variables.

- Drop redundant (those that have equal or more than 0.8) and irrelevant variables:

```
[ ]  df= df.drop(['epoch','epoch_mjd','epoch_cal','Minimum_Orbit_Intersection_Distance_au','per_Sidereal_orbital_period_d','Sidereal_orbital_period_y',
              'w_Argument_perihelion','ad_Aphelion_distance', 'Minimum_Orbit_Intersection_Distance_LD', 'sigma_w', 'tp_Time_perihelion_passage_TDB',
              'sigma_per', 'sigma_ad', 'sigma_tp'], axis=1)

df_modif= df_modif.drop(['epoch_cal','Time_perihelion_passage_calendar'], axis=1)
```

Fig 16. Drop redundant variables in python

- Verify the correlation of actual variables:

```
correlation2= df.corr().round(1)
```

```
[ ] plt.figure(figsize = (15,10))
    sns.heatmap(correlation2, vmin=-1, vmax=1, annot=True, cmap="Spectral", linewidths=.8)
```

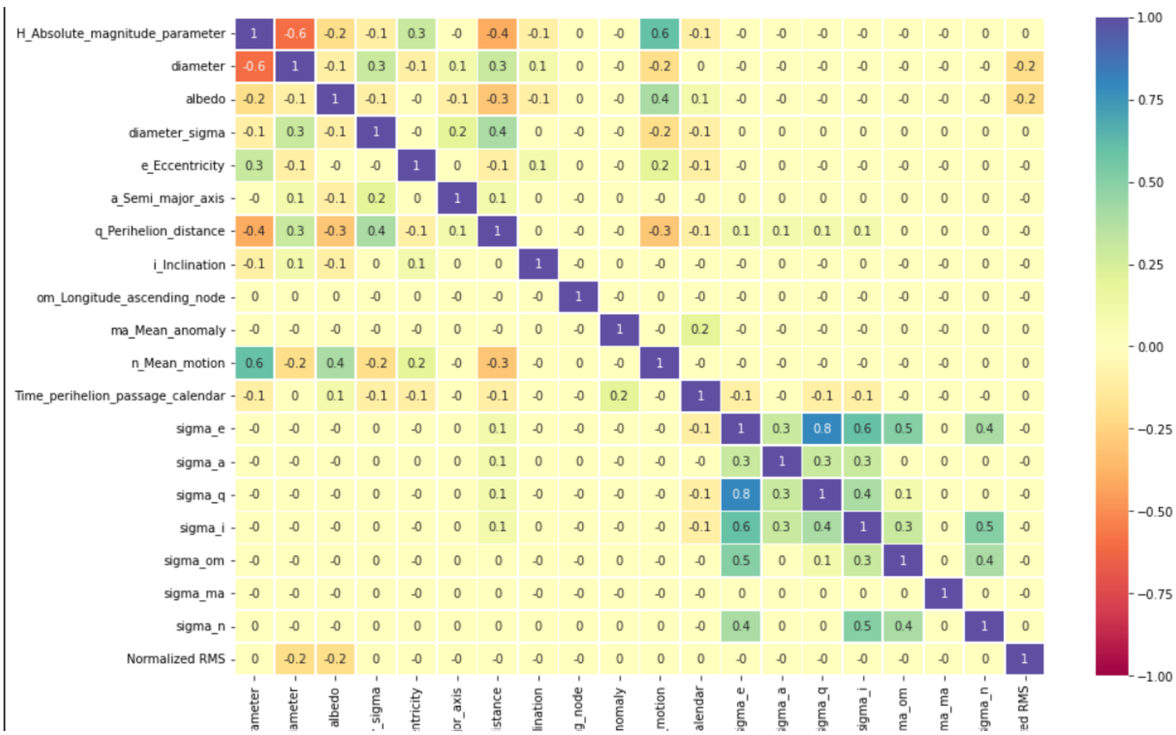


Fig 17. Heatmap of a Correlation table

- Download the data in .xlsx format to work on it in SAS Miner:

```
[ ] df.to_excel('asteroids2.xlsx', sheet_name='sheet1', index=False)
```

Fig 18. Export the dataset to .xlsx format.

SAS Enterprise Miner:

Variables rejected in SAS Miner:

- Irrelevant (ordinal):
    - epoch_cal- date
    - time_perihelion_cal- date

Variable summary:

```
Variable Summary

              Measurement    Frequency
Role            Level          Count

ID            INTERVAL            1
INPUT         INTERVAL           15
INPUT         NOMINAL             2
REJECTED      ORDINAL             2
TARGET        BINARY              1
```

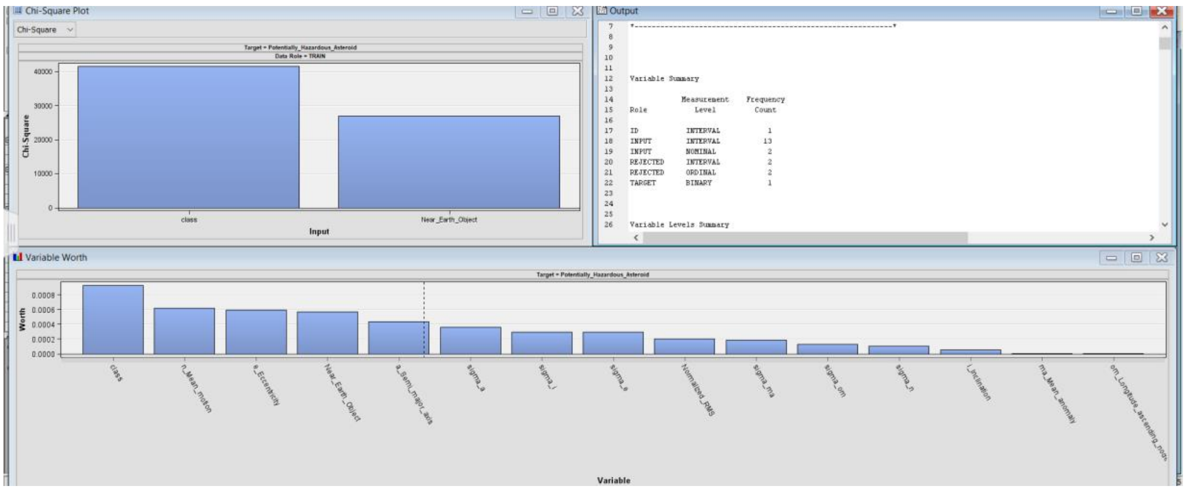


Fig 19. Variable summary and variables worth.

Summary of the dataset

Key points of this dataset:

1. This dataset contains forty-five columns and 958,524 rows. Out of these forty-five columns, thirty-two are float64, three int64 and ten are object.

2. There are twenty-six variables with missing values.

3. A strong correlation has been detected between sixteen variables with more than 0.8x.

4. Some skewness upper than one hundred has been detected in five variables: i) a_Semi_major_axis, ii) sigma_a, iii) sigma_e, iv) sigma_ma, v) sigma_om. We will conduct imputations; stat explore and transform variables to improve skewness and kurtosis because it is important for the model.

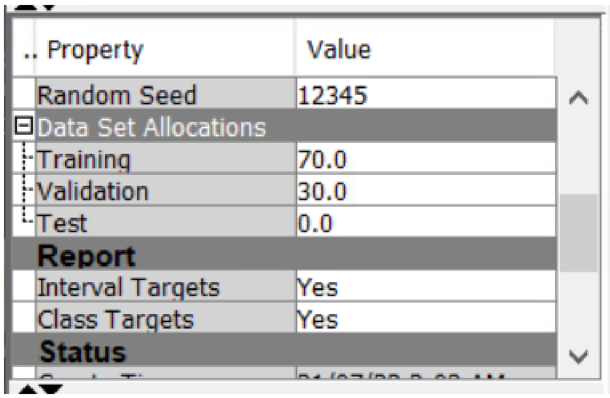
5. Out of the total number of pha_Potentially_hazardous_asteroids in the dataset, 936,537 are 'N' and 2,066 are 'Y'.

Data Preparation and Feature Engineering

Data Preparation Needs

- **SAS Enterprise Miner:**

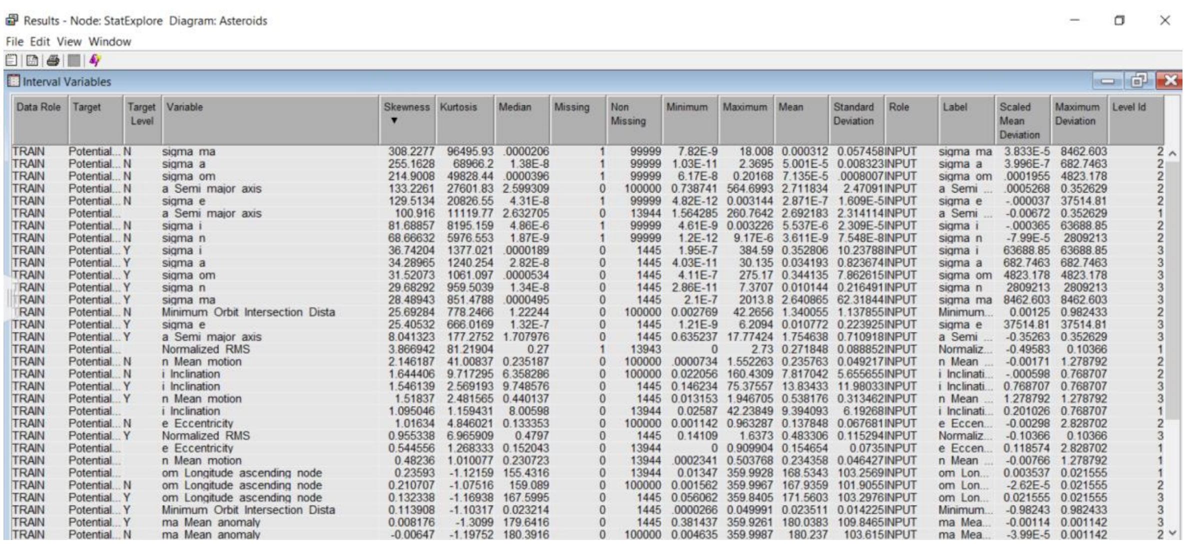
1. Data Partition:



Property	Value
Random Seed	12345
Data Set Allocations	
Training	70.0
Validation	30.0
Test	0.0
Report	
Interval Targets	Yes
Class Targets	Yes
Status	

Fig 20. Partition the data into training and validation.

2. Check the skewness and kurtosis via Stat Explore:



Data Role	Target	Target Level	Variable	Skewness	Kurtosis	Median	Missing	Non Missing	Minimum	Maximum	Mean	Standard Deviation	Role	Label	Scaled Mean Deviation	Maximum Deviation	Level Id
TRAIN	Potential...	N	sigma ma	308.2277	96495.93	.0000206	1	99999	7.82E-9	18.008	0.000312	0.057458	INPUT	sigma ma	3.833E-5	8462.603	2
TRAIN	Potential...	N	sigma a	255.1628	68966.2	1.38E-8	1	99999	1.03E-11	2.3695	5.001E-5	0.008323	INPUT	sigma a	3.996E-7	682.7463	2
TRAIN	Potential...	N	sigma om	214.9008	49828.44	.0000396	1	99999	6.17E-8	0.20168	7.135E-5	.0008007	INPUT	sigma om	.0001955	4823.178	2
TRAIN	Potential...	N	a Semi major axis	133.2261	27601.83	2.599309	0	100000	0.738741	564.6993	2.711834	2.47091	INPUT	a Semi ...	.0005268	0.352629	2
TRAIN	Potential...	N	sigma e	129.5134	20826.55	4.31E-8	1	99999	4.82E-12	0.003144	2.871E-7	1.609E-5	INPUT	sigma e	-.000037	37514.81	2
TRAIN	Potential...		a Semi major axis	100.916	11119.77	2.632705	0	13944	1.564285	260.7642	2.692183	2.314114	INPUT	a Semi ...	-0.00672	0.352629	1
TRAIN	Potential...	N	sigma i	81.68857	8195.159	4.86E-6	1	99999	4.61E-9	0.003226	5.537E-6	2.309E-5	INPUT	sigma i	-.000365	63688.85	2
TRAIN	Potential...	N	sigma n	68.66632	5976.553	1.87E-9	1	99999	1.2E-12	9.17E-6	3.611E-9	7.548E-8	INPUT	sigma n	-7.99E-5	2809213	2
TRAIN	Potential...	Y	sigma i	36.74204	1377.021	.0000189	0	1445	1.95E-7	384.59	0.352806	10.23788	INPUT	sigma i	63688.85	63688.85	3
TRAIN	Potential...	Y	sigma a	34.28965	1240.254	2.82E-8	0	1445	4.03E-11	30.135	0.034193	0.823674	INPUT	sigma a	682.7463	682.7463	3
TRAIN	Potential...	Y	sigma om	31.52073	1061.097	.0000534	0	1445	4.11E-7	275.17	0.344135	7.862615	INPUT	sigma om	4823.178	4823.178	3
TRAIN	Potential...	Y	sigma n	29.68292	959.5039	1.34E-8	0	1445	2.86E-11	7.3707	0.010144	0.216491	INPUT	sigma n	2809213	2809213	3
TRAIN	Potential...	Y	sigma ma	28.48943	851.4788	.0000495	0	1445	2.1E-7	2013.8	2.640865	62.31844	INPUT	sigma ma	8462.603	8462.603	3
TRAIN	Potential...	N	Minimum Orbit Intersection Dista	25.69284	778.2466	1.22244	0	100000	0.002769	42.2656	1.340055	1.137855	INPUT	Minimum...	0.00125	0.982433	2
TRAIN	Potential...	Y	sigma e	25.40532	666.0169	1.32E-7	0	1445	1.21E-9	6.2094	0.010772	0.223925	INPUT	sigma e	37514.81	37514.81	3
TRAIN	Potential...	Y	a Semi major axis	8.041323	177.2752	1.707976	0	1445	0.635237	17.77424	1.754638	0.710918	INPUT	a Semi ...	-0.35263	0.352629	3
TRAIN	Potential...		Normalized RMS	3.866942	81.21904	0.27	1	13943	0	2.73	0.271848	0.088852	INPUT	Normaliz...	-0.49583	0.10366	1
TRAIN	Potential...	N	n Mean motion	2.146187	41.00837	0.235187	0	100000	.0000734	1.552263	0.235763	0.049217	INPUT	n Mean ...	-0.00171	1.278792	2
TRAIN	Potential...	N	i Inclination	1.644406	9.717295	6.358286	0	100000	0.022056	160.4309	7.817042	5.655655	INPUT	i Inclinati...	-.000598	0.768707	2
TRAIN	Potential...	Y	i Inclination	1.546139	2.569193	9.748576	0	1445	0.146234	75.37557	13.83433	11.98033	INPUT	i Inclinati...	0.768707	0.768707	3
TRAIN	Potential...	Y	n Mean motion	1.51837	2.481565	0.440137	0	1445	0.013153	1.946705	0.538176	0.313462	INPUT	n Mean ...	1.278792	1.278792	3
TRAIN	Potential...		i Inclination	1.095046	1.159431	8.00598	0	13944	0.02587	42.23849	9.394093	6.19268	INPUT	i Inclinati...	0.201026	0.768707	1
TRAIN	Potential...	N	e Eccentricity	1.01634	4.846021	0.133353	0	100000	0.001142	0.963287	0.137848	0.067681	INPUT	e Eccen...	-0.00298	2.828702	2
TRAIN	Potential...	Y	Normalized RMS	0.955338	6.965909	0.4797	0	1445	0.14109	1.6373	0.483306	0.115294	INPUT	Normaliz...	-0.10366	0.10366	3
TRAIN	Potential...		e Eccentricity	0.544556	1.268333	0.152043	0	13944	0	0.909904	0.154654	0.0735	INPUT	e Eccen...	0.118574	2.828702	1
TRAIN	Potential...		n Mean motion	0.48236	1.010077	0.230723	0	13944	.0002341	0.503768	0.234358	0.046427	INPUT	n Mean ...	-0.00766	1.278792	1
TRAIN	Potential...		om Longitude ascending node	0.23593	-1.12159	155.4316	0	13944	0.01347	359.9928	168.5343	103.2569	INPUT	om Lon...	0.003537	0.021555	1
TRAIN	Potential...	N	om Longitude ascending node	0.210707	-1.07516	159.089	0	100000	0.001562	359.9967	167.9359	101.9055	INPUT	om Lon...	-2.62E-5	0.021555	2
TRAIN	Potential...	Y	om Longitude ascending node	0.132338	-1.16938	167.5995	0	1445	0.056062	359.8405	171.5603	103.2976	INPUT	om Lon...	0.021555	0.021555	3
TRAIN	Potential...	Y	Minimum Orbit Intersection Dista	0.113908	-1.10317	0.023214	0	1445	.0000266	0.049991	0.023511	0.014225	INPUT	Minimum...	-0.98243	0.982433	3
TRAIN	Potential...	Y	ma Mean anomaly	0.008176	-1.3099	179.6416	0	1445	0.381437	359.9261	180.0383	109.8465	INPUT	ma Mea...	-0.00114	0.001142	3
TRAIN	Potential...	N	ma Mean anomaly	-0.00647	-1.19752	180.3916	0	100000	0.004635	359.9987	180.237	103.615	INPUT	ma Mea...	-3.99E-5	0.001142	2

Fig 21. Stat Explore node.

3. Input missing values:

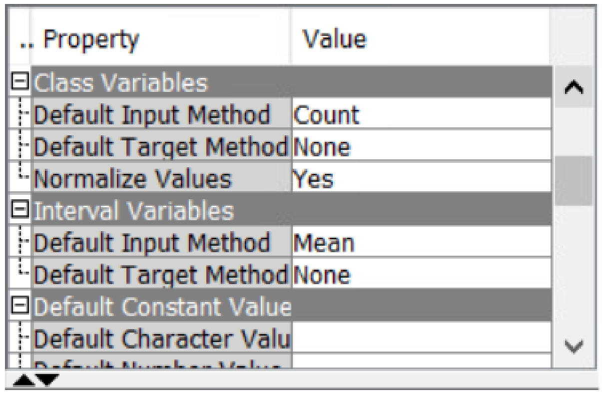


Fig 22. Input with mean method.

4. Cap&Floor because of the high skewness presented:

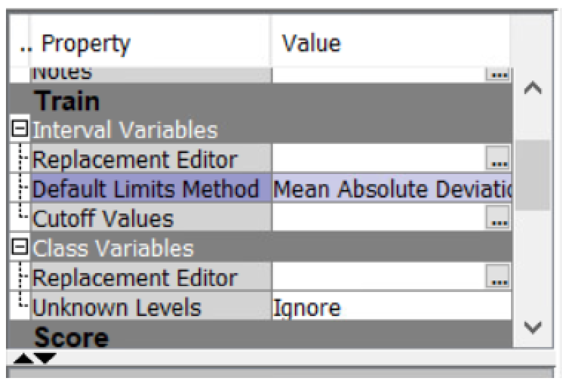
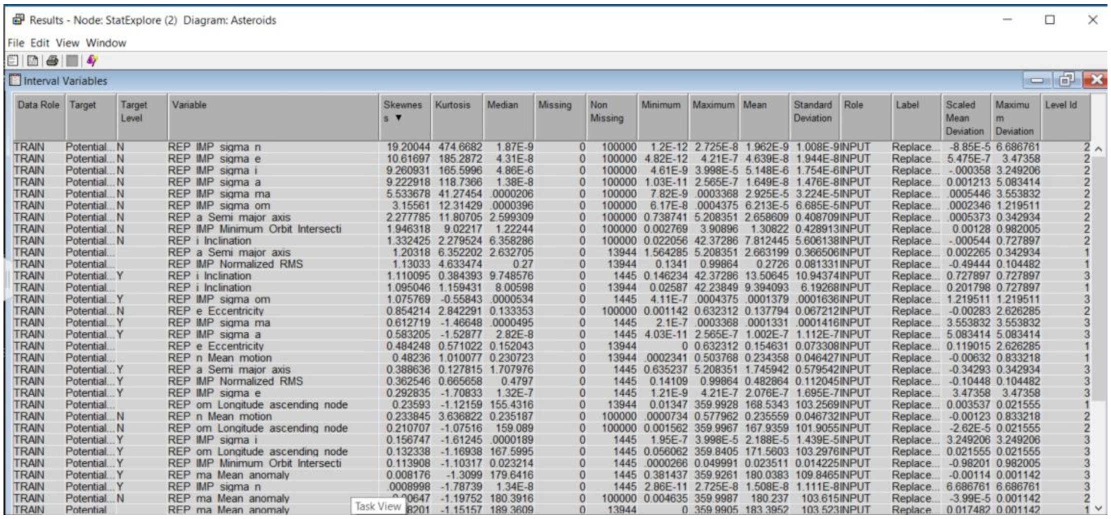


Fig 23. Cap&Floor MAD.

5. Check the skewness and kurtosis again via Stat Explore:



Results - Node: StatExplore (2) Diagram: Asteroids

File Edit View Window

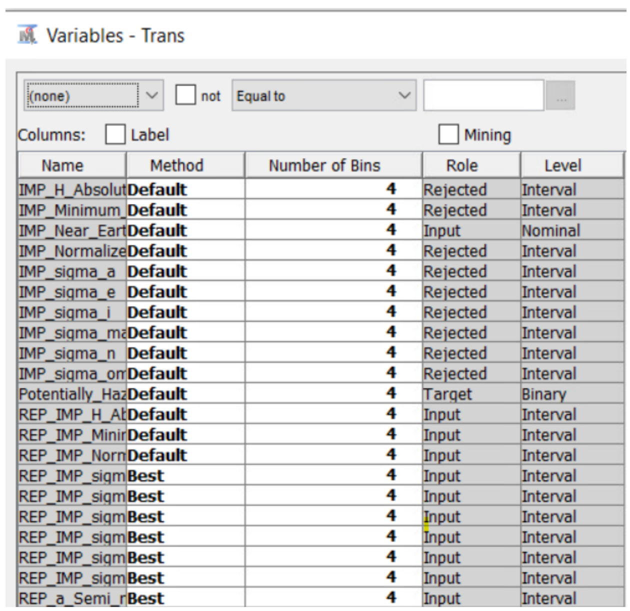
Interval Variables

Data Role	Target	Target Level	Variable	Skewness	Kurtosis	Median	Missing	Non Missing	Minimum	Maximum	Mean	Standard Deviation	Role	Label	Scaled Mean Deviation	Maximum Deviation	Level Id
TRAIN	Potential...	N	REP IMP sigma n	19.20044	474.6682	1.87E-9	0	100000	1.2E-12	2.725E-8	1.962E-9	1.008E-9	INPUT	Replace...	-8.85E-5	6.686761	2
TRAIN	Potential...	N	REP IMP sigma e	10.61697	185.2872	4.31E-8	0	100000	4.82E-12	4.21E-7	4.639E-8	1.944E-8	INPUT	Replace...	5.475E-7	3.47358	2
TRAIN	Potential...	N	REP IMP sigma i	9.260931	165.5996	4.86E-6	0	100000	4.61E-9	3.998E-5	5.148E-6	1.754E-6	INPUT	Replace...	-.000358	3.249206	2
TRAIN	Potential...	N	REP IMP sigma a	9.222918	118.7366	1.38E-8	0	100000	1.03E-11	2.565E-7	1.649E-8	1.476E-8	INPUT	Replace...	0.001213	5.083414	2
TRAIN	Potential...	N	REP IMP sigma ma	5.533678	41.27454	.0000206	0	100000	7.82E-9	.0003368	2.925E-5	3.224E-5	INPUT	Replace...	.0005446	3.553832	2
TRAIN	Potential...	N	REP IMP sigma om	3.15561	12.31429	.0000396	0	100000	6.17E-8	.0004375	6.213E-5	6.685E-5	INPUT	Replace...	.0002346	1.219511	2
TRAIN	Potential...	N	REP a Semi major axis	2.277785	11.80705	2.599309	0	100000	0.738741	5.208351	2.658609	0.408709	INPUT	Replace...	.0005373	0.342934	2
TRAIN	Potential...	N	REP IMP Minimum Orbit Intersecti	1.946318	9.02217	1.22244	0	100000	0.002769	3.90896	1.30822	0.428913	INPUT	Replace...	0.00128	0.982005	2
TRAIN	Potential...	N	REP i Inclination	1.332425	2.279524	6.358286	0	100000	0.022056	42.37286	7.812445	5.606138	INPUT	Replace...	-.000544	0.727897	2
TRAIN	Potential...		REP a Semi major axis	1.20318	6.352202	2.632705	0	13944	1.564285	5.208351	2.663199	0.366506	INPUT	Replace...	0.002265	0.342934	1
TRAIN	Potential...		REP IMP Normalized RMS	1.13033	4.633474	0.27	0	13944	0.1341	0.99864	0.2726	0.081331	INPUT	Replace...	-0.49444	0.104482	1
TRAIN	Potential...	Y	REP i Inclination	1.110095	0.384393	9.748576	0	1445	0.146234	42.37286	13.50645	10.94374	INPUT	Replace...	0.727897	0.727897	3
TRAIN	Potential...		REP i Inclination	1.095046	1.159431	8.00598	0	13944	0.02587	42.23849	9.394093	6.19268	INPUT	Replace...	0.201798	0.727897	1
TRAIN	Potential...	Y	REP IMP sigma om	1.075769	-0.55843	.0000534	0	1445	4.11E-7	.0004375	.0001379	.0001636	INPUT	Replace...	1.219511	1.219511	3
TRAIN	Potential...	N	REP e Eccentricity	0.854214	2.842291	0.133353	0	100000	0.001142	0.632312	0.137794	0.067212	INPUT	Replace...	-0.00283	2.626285	2
TRAIN	Potential...	Y	REP IMP sigma ma	0.612719	-1.46648	.0000495	0	1445	2.1E-7	.0003368	.0001331	.0001416	INPUT	Replace...	3.553832	3.553832	3
TRAIN	Potential...	Y	REP IMP sigma a	0.583205	-1.52877	2.82E-8	0	1445	4.03E-11	2.565E-7	1.002E-7	1.112E-7	INPUT	Replace...	5.083414	5.083414	3
TRAIN	Potential...		REP e Eccentricity	0.484248	0.571022	0.152043	0	13944	0	0.632312	0.154631	0.073308	INPUT	Replace...	0.119015	2.626285	1
TRAIN	Potential...		REP n Mean motion	0.48236	1.010077	0.230723	0	13944	.0002341	0.503768	0.234358	0.046427	INPUT	Replace...	-0.00632	0.833218	1
TRAIN	Potential...	Y	REP a Semi major axis	0.388636	0.127815	1.707976	0	1445	0.635237	5.208351	1.745942	0.579542	INPUT	Replace...	-0.34293	0.342934	3
TRAIN	Potential...	Y	REP IMP Normalized RMS	0.362546	0.665658	0.4797	0	1445	0.14109	0.99864	0.482864	0.112045	INPUT	Replace...	-0.10448	0.104482	3
TRAIN	Potential...	Y	REP IMP sigma e	0.292835	-1.70833	1.32E-7	0	1445	1.21E-9	4.21E-7	2.076E-7	1.695E-7	INPUT	Replace...	3.47358	3.47358	3
TRAIN	Potential...		REP om Longitude ascending node	0.23593	-1.12159	155.4316	0	13944	0.01347	359.9928	168.5343	103.2569	INPUT	Replace...	0.003537	0.021555	1
TRAIN	Potential...	N	REP n Mean motion	0.233845	3.636822	0.235187	0	100000	.0000734	0.577962	0.235559	0.046732	INPUT	Replace...	-0.00123	0.833218	2
TRAIN	Potential...	N	REP om Longitude ascending node	0.210707	-1.07516	159.089	0	100000	0.001562	359.9967	167.9359	101.9055	INPUT	Replace...	-2.62E-5	0.021555	2
TRAIN	Potential...	Y	REP IMP sigma i	0.156747	-1.61245	.0000189	0	1445	1.95E-7	3.998E-5	2.188E-5	1.439E-5	INPUT	Replace...	3.249206	3.249206	3
TRAIN	Potential...	Y	REP om Longitude ascending node	0.132338	-1.16938	167.5995	0	1445	0.056062	359.8405	171.5603	103.2976	INPUT	Replace...	0.021555	0.021555	3
TRAIN	Potential...	Y	REP IMP Minimum Orbit Intersecti	0.113908	-1.10317	0.023214	0	1445	.0000266	0.049991	0.023511	0.014225	INPUT	Replace...	-0.98201	0.982005	3
TRAIN	Potential...	Y	REP ma Mean anomaly	0.008176	-1.3099	179.6416	0	1445	0.381437	359.9261	180.0383	109.8465	INPUT	Replace...	-0.00114	0.001142	3
TRAIN	Potential...	Y	REP IMP sigma n	.0008998	-1.78739	1.34E-8	0	1445	2.86E-11	2.725E-8	1.508E-8	1.111E-8	INPUT	Replace...	6.686761	6.686761	3
TRAIN	Potential...	N	REP ma Mean anomaly	[illegible]00647	-1.19752	180.3916	0	100000	0.004635	359.9987	180.237	103.615	INPUT	Replace...	-3.99E-5	0.001142	2
TRAIN	Potential...		REP ma Mean anomaly	[illegible]8201	-1.15157	189.3609	0	13944	0	359.9905	183.3952	103.523	INPUT	Replace...	0.017482	0.001142	1

Task View

Fig 24. Stat Explore node.

6. Transform variables with the best method because the skewness and kurtosis remain high:



Variables - Trans

(none) not Equal to

Columns: Label Mining

Name	Method	Number of Bins	Role	Level
IMP_H_Absolut	Default	4	Rejected	Interval
IMP_Minimum_	Default	4	Rejected	Interval
IMP_Near_Eart	Default	4	Input	Nominal
IMP_Normalize	Default	4	Rejected	Interval
IMP_sigma_a	Default	4	Rejected	Interval
IMP_sigma_e	Default	4	Rejected	Interval
IMP_sigma_i	Default	4	Rejected	Interval
IMP_sigma_ma	Default	4	Rejected	Interval
IMP_sigma_n	Default	4	Rejected	Interval
IMP_sigma_om	Default	4	Rejected	Interval
Potentially_Haz	Default	4	Target	Binary
REP_IMP_H_Ab	Default	4	Input	Interval
REP_IMP_Minir	Default	4	Input	Interval
REP_IMP_Norn	Default	4	Input	Interval
REP_IMP_sigm	Best	4	Input	Interval
REP_IMP_sigm	Best	4	Input	Interval
REP_IMP_sigm	Best	4	Input	Interval
REP_IMP_sigm	Best	4	Input	Interval
REP_IMP_sigm	Best	4	Input	Interval
REP_IMP_sigm	Best	4	Input	Interval
REP_a_Semi_r	Best	4	Input	Interval

Fig 25. Variable transformation.

7. Verify the skewness and kurtosis via Stat Explore:

Interval Variables

Data Role	Target	Target Level	Variable	Skewness	Kurtosis	Median	Missing	Non Missing	Minimum	Maximum	Mean	Standard Deviation	Role	Label	Scaled Mean Deviation	Maximum Deviation	Level Id
TRAIN	Potentiall...		REP e Eccentricity	0.484248	0.571022	0.152043	0	13944	0	0.632312	0.154631	0.073308	INPUT	Replace...	0.119015	2.626285	1
TRAIN	Potentiall...	N	REP e Eccentricity	0.854214	2.842291	0.133353	0	100000	0.001142	0.632312	0.137794	0.067212	INPUT	Replace...	-0.00283	2.626285	2
TRAIN	Potentiall...	Y	REP e Eccentricity	-1.01587	0.079541	0.55223	0	1445	0.012176	0.632312	0.501098	0.145189	INPUT	Replace...	2.626285	2.626285	3
TRAIN	Potentiall...		REP IMP Minimum Orbit Intersecti	.	.	1.41559	0	13944	1.41559	1.41559	1.41559	0	INPUT	Replace...	0.083459	0.982005	1
TRAIN	Potentiall...	N	REP IMP Minimum Orbit Intersecti	1.946318	9.02217	1.22244	0	100000	0.002769	3.90896	1.30822	0.428913	INPUT	Replace...	0.00128	0.982005	2
TRAIN	Potentiall...	Y	REP IMP Minimum Orbit Intersecti	0.113908	-1.10317	0.023214	0	1445	.0000266	0.049991	0.023511	0.014225	INPUT	Replace...	-0.98201	0.982005	3
TRAIN	Potentiall...		REP n Mean motion	0.48236	1.010077	0.230723	0	13944	.0002341	0.503768	0.234358	0.046427	INPUT	Replace...	-0.00632	0.833218	1
TRAIN	Potentiall...	N	REP n Mean motion	0.233845	3.636822	0.235187	0	100000	.0000734	0.577962	0.235559	0.046732	INPUT	Replace...	-0.00123	0.833218	2
TRAIN	Potentiall...	Y	REP n Mean motion	-0.25671	-1.42105	0.440137	0	1445	0.013153	0.577962	0.432363	0.13516	INPUT	Replace...	0.833218	0.833218	3
TRAIN	Potentiall...		REP i Inclination	1.095046	1.159431	8.00598	0	13944	0.02587	42.23849	9.394093	6.19268	INPUT	Replace...	0.201798	0.727897	1
TRAIN	Potentiall...	N	REP i Inclination	1.332425	2.279524	6.358286	0	100000	0.022056	42.37286	7.812445	5.606138	INPUT	Replace...	-.000544	0.727897	2
TRAIN	Potentiall...	Y	REP i Inclination	1.110095	0.384393	9.748576	0	1445	0.146234	42.37286	13.50645	10.94374	INPUT	Replace...	0.727897	0.727897	3
TRAIN	Potentiall...		REP IMP H Absolute magnitude par	-0.30138	2.941032	17.7	0	13944	9.7	21.8	17.67091	0.860331	INPUT	Replace...	0.183342	0.342637	1
TRAIN	Potentiall...	N	REP IMP H Absolute magnitude par	-1.13055	2.784731	15.1	0	100000	9.7	22.7	14.93068	1.175522	INPUT	Replace...	-.000159	0.342637	2
TRAIN	Potentiall...	Y	REP IMP H Absolute magnitude par	-0.9301	0.669906	20.3	0	1445	14	22.4	20.04967	1.473278	INPUT	Replace...	0.342637	0.342637	3
TRAIN	Potentiall...		REP IMP Normalized RMS	1.13033	4.633474	0.27	0	13944	0.1341	0.99864	0.2726	0.081331	INPUT	Replace...	-0.49444	0.104482	1
TRAIN	Potentiall...	N	REP IMP Normalized RMS	-0.50818	1.483254	0.54213	0	100000	0.1341	0.92905	0.53931	0.047804	INPUT	Replace...	.0002033	0.104482	2
TRAIN	Potentiall...	Y	REP IMP Normalized RMS	0.362546	0.665658	0.4797	0	1445	0.14109	0.99864	0.482864	0.112045	INPUT	Replace...	-0.10448	0.104482	3
TRAIN	Potentiall...		REP om Longitude ascending node	0.23593	-1.12159	155.4316	0	13944	0.01347	359.9928	168.5343	103.2569	INPUT	Replace...	0.003537	0.021555	1
TRAIN	Potentiall...	N	REP om Longitude ascending node	0.210707	-1.07516	159.089	0	100000	0.001562	359.9967	167.9359	101.9055	INPUT	Replace...	-2.62E-5	0.021555	2
TRAIN	Potentiall...	Y	REP om Longitude ascending node	0.132338	-1.16938	167.5995	0	1445	0.056062	359.8405	171.5603	103.2976	INPUT	Replace...	0.021555	0.021555	3
TRAIN	Potentiall...		REP ma Mean anomaly	-0.08201	-1.15157	189.3609	0	13944	0	359.9905	183.3952	103.523	INPUT	Replace...	0.017482	0.001142	1
TRAIN	Potentiall...	N	REP ma Mean anomaly	-0.00647	-1.19752	180.3916	0	100000	0.004635	359.9987	180.237	103.615	INPUT	Replace...	-3.99E-5	0.001142	2
TRAIN	Potentiall...	Y	REP ma Mean anomaly	0.008176	-1.3099	179.6416	0	1445	0.381437	359.9261	180.0383	109.8465	INPUT	Replace...	-0.00114	0.001142	3

Fig 26. Stat Explore node.

Python:

1. Change the variables from object to category

```
df['neo']=df['neo'].astype('category')
df['pha']=df['pha'].astype('category')
df['class']=df['class'].astype('category')
```

Fig 27. Change variables type.

2. Managing of missing values "drop"

```
df_modif = df_modif.dropna()
```

Fig 28. Drop missing values.

3. Convert the category variables into dummies

```
df_modif= pd.get_dummies(df_modif, columns=['class', 'Near_Earth_Object'])
```

Fig 29. Variables transformation to binary.

4. Assign values to “X” and “y”

```
X = df_modif.drop('Potentially_Hazardous_Asteroid', axis=1)
y = df_modif['Potentially_Hazardous_Asteroid']
```

Fig 30. X and y values.

5. Define the test size

```
X_train, X_valid, y_train, y_valid = train_test_split(X, y, test_size=0.3, random_state=1)
```

Fig 31. Train & validation size.

6. Fit the model with Standard Scaler

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_valid = scaler.transform(X_valid)
```

Fig 32. Standard Scaler.

Model Exploration

Modeling:

Python libraries for modeling:

```
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import r2_score
from sklearn.linear_model import LinearRegression

!pip install dmba
from dmba import classificationSummary
from dmba import regressionSummary

from sklearn.metrics import classification_report, plot_confusion_matrix, plot_roc_curve
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier
from xgboost import XGBClassifier, XGBRFClassifier
```

Fig 33. Python libraries for modeling

Model

Regression model

Data analysts use regression models to examine relationships between variables. Regression models are often used by organizations to determine which independent variables hold the most influence over dependent variables—information that can be leveraged to make essential business decisions. (Stobierski, 2021)

Full Regression

Run the Full Regression:

Fit Statistics

Target=Potentially_Hazardous_Asteroid Target Label=Potentially_Hazardous_Asteroid

Fit Statistics	Statistics Label	Train	Validation
AIC	Akaike's Information Criterion	3101.22	.
ASE	Average Squared Error	0.00	0.00
AVERR	Average Error Function	0.00	0.00
DFE	Degrees of Freedom for Error	656985.00	.
DFM	Model Degrees of Freedom	36.00	.
DFT	Total Degrees of Freedom	657021.00	.
DIV	Divisor for ASE	1314042.00	563164.00
ERR	Error Function	3029.22	1328.87
FPE	Final Prediction Error	0.00	.
MAX	Maximum Absolute Error	1.00	1.00
MSE	Mean Square Error	0.00	0.00
NOBS	Sum of Frequencies	657021.00	281582.00
_NW_	Number of Estimate Weights	36.00	.
RASE	Root Average Sum of Squares	0.03	0.03
RFPE	Root Final Prediction Error	0.03	.
RMSE	Root Mean Squared Error	0.03	0.03
SBC	Schwarz's Bayesian Criterion	3511.45	.
SSE	Sum of Squared Errors	859.71	383.09
SUMW	Sum of Case Weights Times Freq	1314042.00	563164.00
MISC	Misclassification Rate	0.00	0.00

```
Event Classification Table

Data Role=TRAIN Target=Potentially_Hazardous_Asteroid Target Label=Potentially_Hazardous_Asteroid

  False      True      False      True
Negative   Negative   Positive   Positive

   383      669321      199       1062


Data Role=VALIDATE Target=Potentially_Hazardous_Asteroid Target Label=Potentially_Hazardous_Asteroid

  False      True      False      True
Negative   Negative   Positive   Positive

   185      286855       83        436
```

Fig 34. Full regression fit statistics.

Stepwise Regression

Stepwise Regression is a method of fitting regression models in which the choice of predictive variables is conducted by an automatic procedure. In each step, a variable is considered for addition to or subtraction from the set of explanatory variables based on some prespecified criterion. (Cvetkov, 2021)

SAS Enterprise Miner:

Run the Stepwise Regression:

Fit Statistics

Target=Potentially_Hazardous_Asteroid Target Label=Potentially_Hazardous_Asteroid

Fit Statistics	Statistics Label	Train	Validation
AIC	Akaike's Information Criterion	3099.72	.
ASE	Average Squared Error	0.00	0.00
AVERR	Average Error Function	0.00	0.00
DFE	Degrees of Freedom for Error	656998.00	.
DFM	Model Degrees of Freedom	23.00	.
DFT	Total Degrees of Freedom	657021.00	.
DIV	Divisor for ASE	1314042.00	563164.00
ERR	Error Function	3053.72	1318.47
FPE	Final Prediction Error	0.00	.
MAX	Maximum Absolute Error	1.00	1.00
MSE	Mean Square Error	0.00	0.00
NOBS	Sum of Frequencies	657021.00	281582.00
_NW_	Number of Estimate Weights	23.00	.
RASE	Root Average Sum of Squares	0.03	0.03
RFPE	Root Final Prediction Error	0.03	.
RMSE	Root Mean Squared Error	0.03	0.03
SBC	Schwarz's Bayesian Criterion	3361.81	.
SSE	Sum of Squared Errors	862.79	380.58
SUMW	Sum of Case Weights Times Freq	1314042.00	563164.00
MISC	Misclassification Rate	0.00	0.00

Event Classification Table

Data Role=TRAIN Target=Potentially_Hazardous_Asteroid Target Label=Potentially_Hazardous_Asteroid

False Negative	True Negative	False Positive	True Positive
393	669321	199	1052

Data Role=VALIDATE Target=Potentially_Hazardous_Asteroid Target Label=Potentially_Hazardous_Asteroid

False Negative	True Negative	False Positive	True Positive
187	286858	80	434

Fig 35. Stepwise regression fit statistics.

Forward Regression

Forward Regression is a stepwise regression approach that begins with an empty model and at each step gradually adds variables to the regression model to find a model that best explains the data. (Cvetkov, 2021)

SAS Enterprise Miner:

Run the Forward Regression:

Fit Statistics

Target=Potentially_Hazardous_Asteroid Target Label=Potentially_Hazardous_Asteroid

Fit Statistics	Statistics Label	Train	Validation
AIC	Akaike's Information Criterion	3095.62	.
ASE	Average Squared Error	0.00	0.00
AVERR	Average Error Function	0.00	0.00
DFE	Degrees of Freedom for Error	656997.00	.
DFM	Model Degrees of Freedom	24.00	.
DFT	Total Degrees of Freedom	657021.00	.
DIV	Divisor for ASE	1314042.00	563164.00
ERR	Error Function	3047.62	1319.44
FPE	Final Prediction Error	0.00	.
MAX	Maximum Absolute Error	1.00	1.00
MSE	Mean Square Error	0.00	0.00
NOBS	Sum of Frequencies	657021.00	281582.00
_NW_	Number of Estimate Weights	24.00	.
RASE	Root Average Sum of Squares	0.03	0.03
RFPE	Root Final Prediction Error	0.03	.
RMSE	Root Mean Squared Error	0.03	0.03
SBC	Schwarz's Bayesian Criterion	3369.11	.
SSE	Sum of Squared Errors	861.97	380.00
SUMW	Sum of Case Weights Times Freq	1314042.00	563164.00
MISC	Misclassification Rate	0.00	0.00

Event Classification Table

Data Role=TRAIN Target=Potentially_Hazardous_Asteroid Target Label=Potentially_Hazardous_Asteroid

False Negative	True Negative	False Positive	True Positive
393	669325	195	1052

Data Role=VALIDATE Target=Potentially_Hazardous_Asteroid Target Label=Potentially_Hazardous_Asteroid

False Negative	True Negative	False Positive	True Positive
182	286859	79	439

Fig 36. Forward regression fit statistics.

Backward Regression

Backward Regression is a stepwise regression approach that begins with a full model and at each step gradually eliminates variables from the regression model to find a reduced model that best explains the data. Also known as Backward Elimination Regression. It reduces the number of predictors; the multicollinearity problem and it is one of the ways to resolve the overfitting. (Cvetkov, 2021)

SAS Enterprise Miner:

Run the Backward Regression:

Fit Statistics

Target=Potentially_Hazardous_Asteroid Target Label=Potentially_Hazardous_Asteroid

Fit Statistics	Statistics Label	Train	Validation
AIC	Akaike's Information Criterion	3095.62	.
ASE	Average Squared Error	0.00	0.00
AVERR	Average Error Function	0.00	0.00
DFE	Degrees of Freedom for Error	656997.00	.
DFM	Model Degrees of Freedom	24.00	.
DFT	Total Degrees of Freedom	657021.00	.
DIV	Divisor for ASE	1314042.00	563164.00
ERR	Error Function	3047.62	1319.44
FPE	Final Prediction Error	0.00	.
MAX	Maximum Absolute Error	1.00	1.00
MSE	Mean Square Error	0.00	0.00
NOBS	Sum of Frequencies	657021.00	281582.00
_NW_	Number of Estimate Weights	24.00	.
RASE	Root Average Sum of Squares	0.03	0.03
RFPE	Root Final Prediction Error	0.03	.
RMSE	Root Mean Squared Error	0.03	0.03
SBC	Schwarz's Bayesian Criterion	3369.11	.
SSE	Sum of Squared Errors	861.97	380.00
SUMW	Sum of Case Weights Times Freq	1314042.00	563164.00
MISC	Misclassification Rate	0.00	0.00

Event Classification Table

Data Role=TRAIN Target=Potentially_Hazardous_Asteroid Target Label=Potentially_Hazardous_Asteroid

False Negative	True Negative	False Positive	True Positive
393	669325	195	1052

Data Role=VALIDATE Target=Potentially_Hazardous_Asteroid Target Label=Potentially_Hazardous_Asteroid

False Negative	True Negative	False Positive	True Positive
182	286859	79	439

Fig 38. Backward regression fit statistics.

Logistic Regression

Regression analysis is a type of predictive modeling technique which is used to find the relationship between a dependent variable (usually known as the "Y" variable) and either one independent variable (the "X" variable) or a series of independent variables. (Thanda, 2022)

Logistic regression is the correct type of analysis to use when the analysis conducted is with binary data. ("What is Logistic Regression? A Beginner's Guide - CareerFoundry") Binary data is the output or dependent variable is dichotomous or categorical in nature; for example, "yes" or "no," "pass" or "fail." ("Regression.docx - What is logistic regression? Logistic...") Even though, the independent variables could be: i) continuous (interval data, each value are equally split; ratio data, each value are equally split and there is a true or meaningful "zero"); ii) discrete ordinal, scale/range data (e.g., 1 to 5); iii) discrete nominal, categorical scale data.

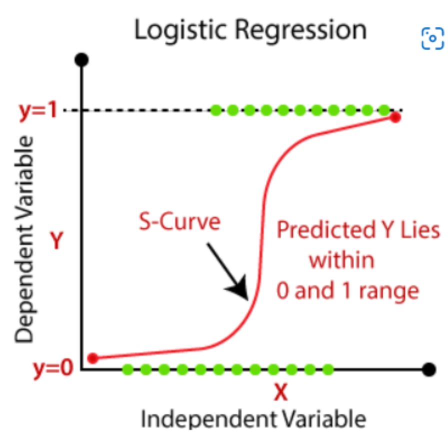


Fig 38: Logistic Regression graph representation (Seth, 2020)

Logistic Regression assumptions:

- The target/dependent variable is binary or dichotomous.
- The predictor variables should not present a multicollinearity or the multicollinearity between them should be small.
- Independent variables linearly related to the log odds.

- This kind of analysis require an enormous size sample of data.

Python code:

```
model = LogisticRegression(max_iter=100000)
model.fit(X_train,y_train)
y_preds = model.predict(X_valid)
y_preds1 = model.predict(X_train)

print(classification_report(y_valid,y_preds))

print(classificationSummary(y_train,y_preds1))
print(classificationSummary(y_valid,y_preds))
```

```
              precision    recall  f1-score   support

           N       1.00      1.00      1.00    279083
           Y       0.66      0.31      0.42       618

    accuracy                           1.00    279701
   macro avg       0.83      0.66      0.71    279701
weighted avg       1.00      1.00      1.00    279701

Confusion Matrix (Accuracy 0.9981)

       Prediction
Actual      0      1
     0 650964    222
     1   1009    439
None
Confusion Matrix (Accuracy 0.9981)

       Prediction
Actual      0      1
     0 278986     97
     1    426    192
```

Fig 39. Logistic Regression classification report

Decision Trees

A Decision Tree is a type of algorithm that includes conditional 'control' statements to classify data and it can deal with complex data. ("the consumer decision process model represents - Kazuyasu") It starts at a node which then branches in two or more directions. Each branch offers different possible outcomes, incorporating a variety of decisions and chance events until a final outcome is achieved. (Hillier, 2021)

Decision trees can be used to deal with complex datasets and can be pruned if necessary to avoid overfitting.

Parts of a decision tree:

- Decision nodes: shown in a square, represents a decision.
- Chance nodes: shown in a circle, represents the probability or uncertainty.
- End nodes: shown in a triangle, represents the outcome.

All of the nodes mentioned above are connected through branches.

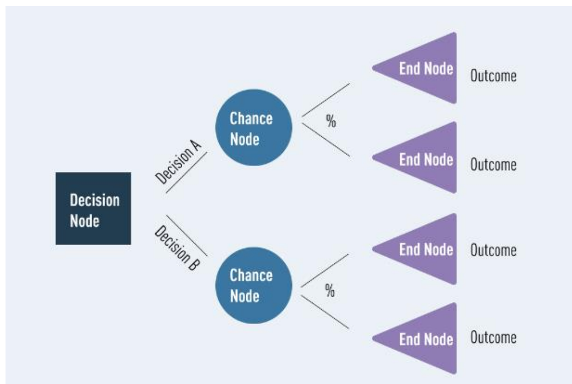


Figure 40: Decision Tree representation

Advantages of decision trees:

- Easy for interpreting data in a visual way.
- Good for managing a combination of numerical and non-numerical data.

Disadvantages of decision trees:

- Overfitting could be a problem if a decision tree's design is too complex.
- It is not a clever idea when the data contains continuous variables.
- "In predictive analysis, calculations can quickly grow cumbersome, especially when a decision path includes many chance variables." ("What Is a Decision Tree and How Is It Used? - CareerFoundry")
- Outcomes could be biased if the dataset is imbalanced.

Maximal Tree

SAS Enterprise Miner

1. Grab into the diagram a decision tree, via interactive model check the maximal tree.

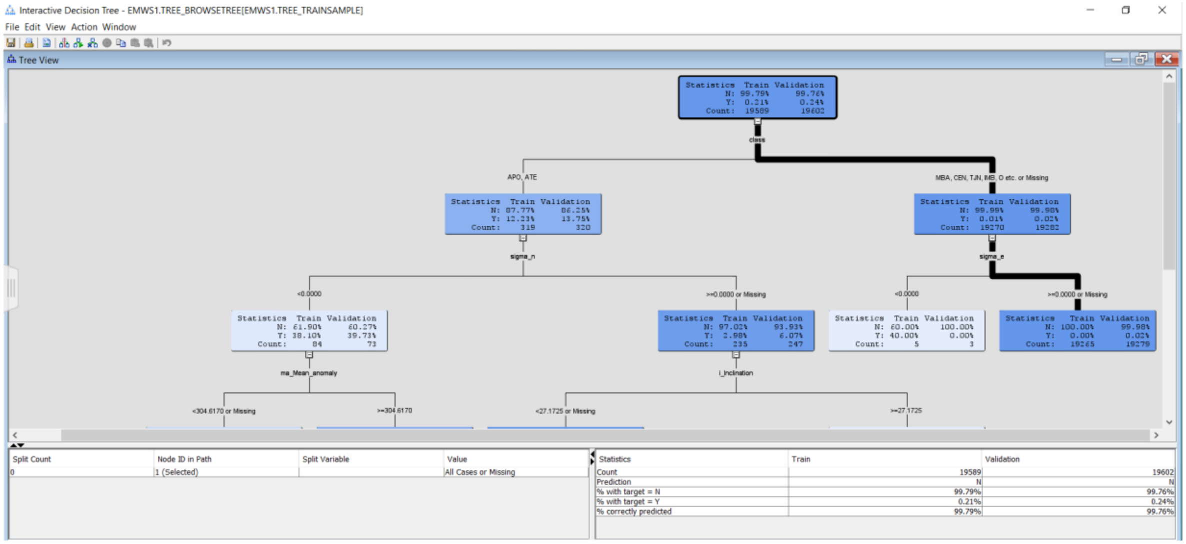


Fig 41. Maximal tree

2. Freeze the model

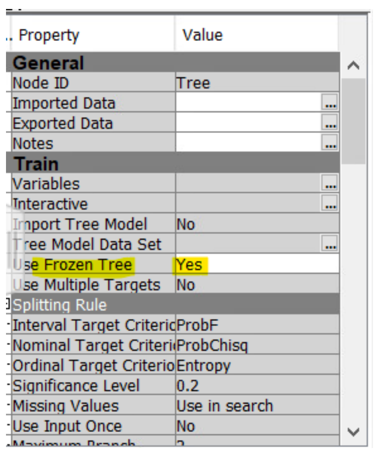


Fig 42. Selection of frozen tree.

3. Run the maximal tree

Fit Statistics

Target=Potentially_Hazardous_Asteroid Target Label=Potentially_Hazardous_Asteroid

Fit Statistics	Statistics Label	Train	Validation
NOBS	Sum of Frequencies	657021.00	281582.00
MISC	Misclassification Rate	0.00	0.00
MAX	Maximum Absolute Error	1.00	1.00
SSE	Sum of Squared Errors	2129.68	917.18
ASE	Average Squared Error	0.00	0.00
RASE	Root Average Squared Error	0.04	0.04
DIV	Divisor for ASE	1314042.00	563164.00
DFT	Total Degrees of Freedom	657021.00	.

Event Classification Table

Data Role=TRAIN Target=Potentially_Hazardous_Asteroid Target Label=Potentially_Hazardous_Asteroid

False Negative	True Negative	False Positive	True Positive
1351	669431	89	94

Data Role=VALIDATE Target=Potentially_Hazardous_Asteroid Target Label=Potentially_Hazardous_Asteroid

False Negative	True Negative	False Positive	True Positive
587	286899	39	34

Fig 43. Maximal tree fit statistics.

Misclassification Tree

SAS Enterprise Miner

1. Change the assessment measure to Misclassification

.. Property	Value
Number of Surrogate R	0
Split Size	.
Split Search	
Use Decisions	No
Use Priors	No
Exhaustive	5000
Node Sample	20000
Subtree	
Method	Assessment
Number of Leaves	1
Assessment Measure	Misclassification
Assessment Fraction	0.25
Cross Validation	
Perform Cross Validati	No
Number of Subsets	10
Number of Repeats	1
Seed	12345
Observation Based Imp	
Observation Based Imp	No
Number Single Var Imp	5

Fig 44. Assessment measure: misclassification.

1. Run the node

```
Fit Statistics

Target=Potentially_Hazardous_Asteroid Target Label=Potentially_Hazardous_Asteroid

   Fit
Statistics     Statistics Label                  Train     Validation

  _NOBS_       Sum of Frequencies            657021.00      281582.00
  _MISC_       Misclassification Rate             0.00           0.00
  _MAX_        Maximum Absolute Error             1.00           1.00
  _SSE_        Sum of Squared Errors             82.86          27.75
  _ASE_        Average Squared Error              0.00           0.00
  _RASE_       Root Average Squared Error         0.01           0.01
  _DIV_        Divisor for ASE              1314042.00      563164.00
  _DFT_        Total Degrees of Freedom      657021.00              .
```

```
Event Classification Table

Data Role=TRAIN Target=Potentially_Hazardous_Asteroid Target Label=Potentially_Hazardous_Asteroid

  False       True       False       True
Negative    Negative    Positive    Positive

   14        669492        28         1431


Data Role=VALIDATE Target=Potentially_Hazardous_Asteroid Target Label=Potentially_Hazardous_Asteroid

  False       True       False       True
Negative    Negative    Positive    Positive

    5        286929         9          616
```

Fig 45. Misclassification tree fit statistics.

Average Squared Error Tree

SAS Enterprise Miner:

1. Change the assessment measure to Average Squared Error

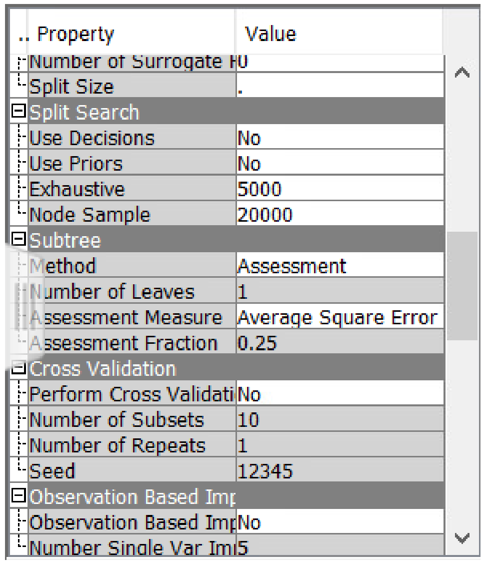


Fig 46. Assessment measure ASE.

2. Run the node

Fit Statistics

Target=Potentially_Hazardous_Asteroid Target Label=Potentially_Hazardous_Asteroid

Fit Statistics	Statistics Label	Train	Validation
NOBS	Sum of Frequencies	657021.00	281582.00
MISC	Misclassification Rate	0.00	0.00
MAX	Maximum Absolute Error	1.00	1.00
SSE	Sum of Squared Errors	57.04	17.36
ASE	Average Squared Error	0.00	0.00
RASE	Root Average Squared Error	0.01	0.01
DIV	Divisor for ASE	1314042.00	563164.00
DFT	Total Degrees of Freedom	657021.00	.

Event Classification Table

Data Role=TRAIN Target=Potentially_Hazardous_Asteroid Target Label=Potentially_Hazardous_Asteroid

False Negative	True Negative	False Positive	True Positive
14	669492	28	1431

Data Role=VALIDATE Target=Potentially_Hazardous_Asteroid Target Label=Potentially_Hazardous_Asteroid

False Negative	True Negative	False Positive	True Positive
5	286929	9	616

Fig 47. Average Squared Error tree fit statistics.

Classification Tree

Python code:

```
model1 = DecisionTreeClassifier()
model1.fit(X_train,y_train)
y_preds = model1.predict(X_valid)
y_preds1 = model.predict(X_train)

print(classification_report(y_valid,y_preds))

print(classificationSummary(y_train,y_preds1))
print(classificationSummary(y_valid,y_preds))
```

```
              precision    recall  f1-score   support

           N       1.00      1.00      1.00    279083
           Y       0.98      0.97      0.98       618

    accuracy                           1.00    279701
   macro avg       0.99      0.99      0.99    279701
weighted avg       1.00      1.00      1.00    279701

Confusion Matrix (Accuracy 0.9981)

       Prediction
Actual      0      1
     0 650964    222
     1   1009    439
None
Confusion Matrix (Accuracy 0.9999)

       Prediction
Actual      0      1
     0 279070     13
     1     17    601
```

Fig 48. Decision tree classifier classification report.

```
feat_importances = pd.DataFrame(model1.feature_importances_, index=X.columns)
print(feat_importances.sort_values(0))

plt.figure(figsize=(15,5))
plt.xticks(rotation=90)
plt.title('Feature Importances')
sns.barplot(data= feat_importances.sort_values(0).T);
```

```
Near_Earth_Object_Y                    0.000000
class_MBA                              0.000000
class_IMB                              0.000000
class_IEO                              0.000000
class_CEN                              0.000000
class_ATE                              0.000000
class_AST                              0.000000
class_APO                              0.000000
class_AMO                              0.000000
Near_Earth_Object_N                    0.000000
sigma_n                                0.000000
class_MCA                              0.000000
sigma_ma                               0.000000
sigma_i                                0.000000
sigma_e                                0.000000
class_TJN                              0.000000
class_TNO                              0.000000
class_OMB                              0.000000
sigma_a                                0.000346
sigma_om                               0.000703
a_Semi_major_axis                      0.001146
om_Longitude_ascending_node            0.001298
ma_Mean_anomaly                        0.001470
i_Inclination                          0.002485
e_Eccentricity                         0.002692
Normalized RMS                         0.003049
n_Mean_motion                          0.004035
Minimum_Orbit_Intersection_Distance_au 0.185518
H_Absolute_magnitude_parameter         0.797258
```

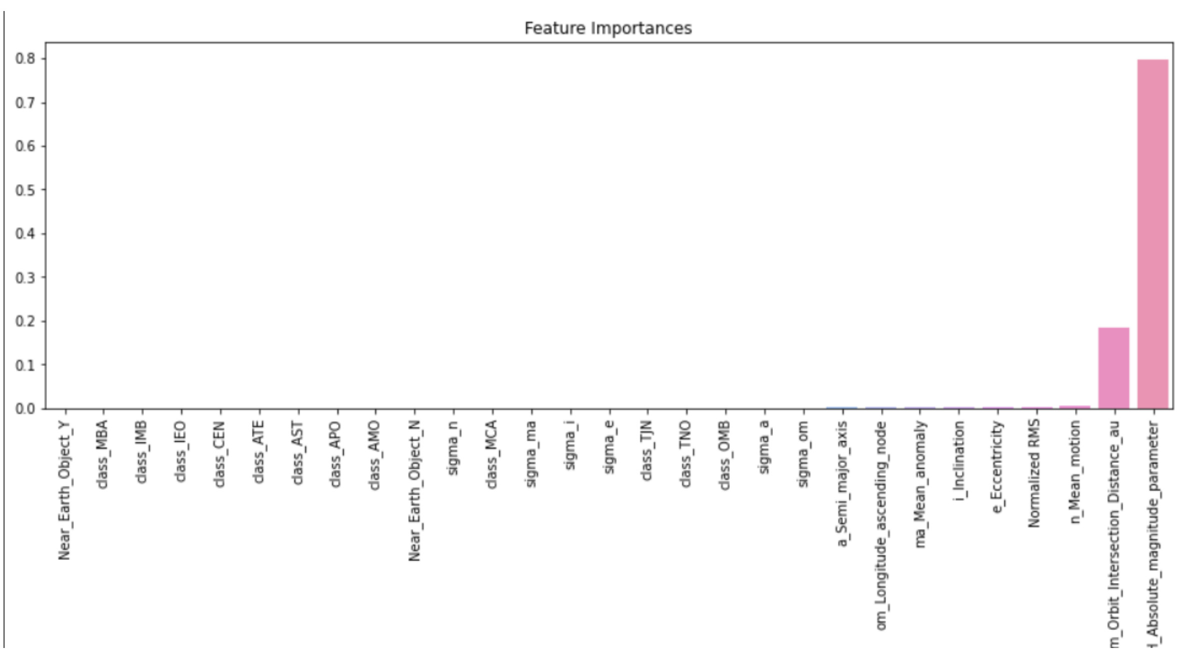


Fig 49. Decision tree classifier feature importance.

Random Forest

"Random forest is a combination of decision trees that can be modeled for prediction and behavior analysis. The decision tree in a forest cannot be pruned for sampling and hence, prediction selection." ("Random Forest - Overview, Modeling Predictions, Advantages")

The random forest technique can manage large data sets due to its capability to work with many variables running to thousands.

Python code:

```
model6 = RandomForestClassifier()
model6.fit(X_train,y_train)
y_preds = model6.predict(X_valid)
y_preds1 = model4.predict(X_train)

print(classification_report(y_valid,y_preds))

print(classificationSummary(y_train,y_preds1))
print(classificationSummary(y_valid,y_preds))
```

```
              precision    recall  f1-score   support

           N       1.00      1.00      1.00    279083
           Y       1.00      0.98      0.99       618

    accuracy                           1.00    279701
   macro avg       1.00      0.99      0.99    279701
weighted avg       1.00      1.00      1.00    279701

Confusion Matrix (Accuracy 0.9999)

       Prediction
Actual      0      1
     0 651161     25
     1     13   1435
None
Confusion Matrix (Accuracy 1.0000)

       Prediction
Actual      0      1
     0 279081      2
     1     11    607
```

Fig 50. Random Forest Classifier classification report

```
feat_importances = pd.DataFrame(model6.feature_importances_, index=X.columns)
print(feat_importances.sort_values(0))

plt.figure(figsize=(15,5))
plt.xticks(rotation=90)
plt.title('Feature Importances')
sns.barplot(data= feat_importances.sort_values(0).T);
```

```
class_IMB                           0.000000e+00
class_CEN                           0.000000e+00
class_AST                           0.000000e+00
class_TJN                           0.000000e+00
class_OMB                           6.949933e-07
class_TNO                           1.225290e-04
class_MCA                           1.278402e-04
class_IEO                           1.677927e-04
sigma_ma                            3.888299e-04
class_MBA                           6.973380e-04
class_ATE                           1.202840e-03
sigma_a                             2.566668e-03
sigma_om                            1.004544e-02
class_AMO                           1.058686e-02
om_Longitude_ascending_node         1.063325e-02
ma_Mean_anomaly                     1.094432e-02
Normalized RMS                      1.271083e-02
Near_Earth_Object_Y                 1.308221e-02
sigma_e                             1.708701e-02
Near_Earth_Object_N                 1.790301e-02
n_Mean_motion                       2.106766e-02
a_Semi_major_axis                   2.264575e-02
class_APO                           3.248903e-02
i_Inclination                       3.699731e-02
e_Eccentricity                      3.774149e-02
sigma_i                             5.101628e-02
sigma_n                             8.890138e-02
H_Absolute_magnitude_parameter      2.936121e-01
Minimum_Orbit_Intersection_Distance_au  3.072616e-01
```

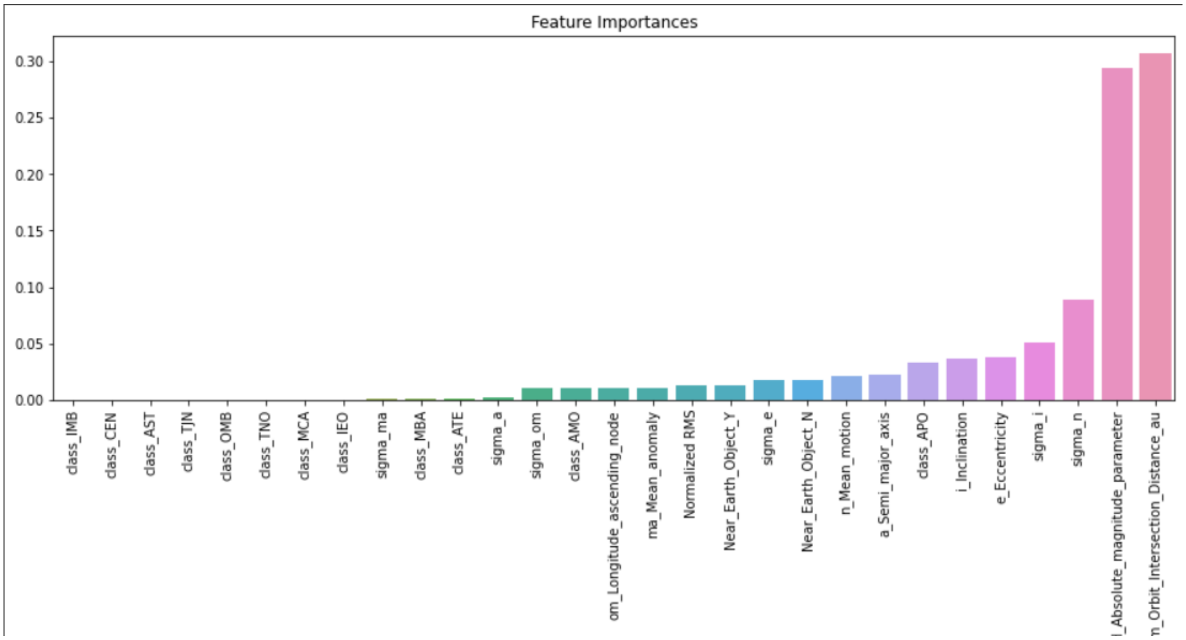


Fig 51. Random Forest Classifier feature importance.

Boosting

Boosting is an ensemble learning method that combines a set of weak learners into a strong learner to minimize training errors. In boosting, a random sample of data is selected, fitted with a model, and then trained sequentially—that is, each model tries to compensate for the weaknesses of its predecessor. With each iteration, the weak rules from each individual classifier are combined to form one, strong prediction rule. (IBM Cloud Education, 2021)

Advantages:

- Ease of Implementation: No data preprocessing is required.
- Reduction of bias.
- Computational Efficiency.

Disadvantages:

- Intense computation: Boosting algorithms can be slower to train when compared to bagging as a large number of parameters can also influence the behavior of the model.

AdaBoost Classifier

Yoav Freund and Robert Schapire are credited with the creation of the AdaBoost algorithm. This method operates iteratively, identifying misclassified data points and adjusting their weights to minimize the training error. The model continues optimize in a sequential fashion until it yields the strongest predictor. (IBM Cloud Education, 2021)

Python code:

```
model2 = AdaBoostClassifier()
model2.fit(X_train,y_train)
y_preds = model2.predict(X_valid)
y_preds1 = model.predict(X_train)

print(classification_report(y_valid,y_preds))

print(classificationSummary(y_train,y_preds1))
print(classificationSummary(y_valid,y_preds))
```

```
              precision    recall  f1-score   support

           N       1.00      1.00      1.00    279083
           Y       0.99      0.98      0.98       618

    accuracy                           1.00    279701
   macro avg       1.00      0.99      0.99    279701
weighted avg       1.00      1.00      1.00    279701

Confusion Matrix (Accuracy 0.9981)

       Prediction
Actual      0      1
     0 650964    222
     1   1009    439
None
Confusion Matrix (Accuracy 0.9999)

       Prediction
Actual      0      1
     0 279077      6
     1     13    605
```

Fig 52. Ada Boost Classifier classification report

```
feat_importances = pd.DataFrame(model2.feature_importances_, index=X.columns)
print(feat_importances.sort_values(0))

plt.figure(figsize=(15,5))
plt.xticks(rotation=90)
plt.title('Feature Importances')
sns.barplot(data= feat_importances.sort_values(0).T);
```

```
Near_Earth_Object_Y                       0.00
class_ATE                                 0.00
class_AST                                 0.00
class_APO                                 0.00
class_AMO                                 0.00
Near_Earth_Object_N                       0.00
sigma_ma                                  0.00
sigma_om                                  0.00
class_CEN                                 0.00
sigma_i                                   0.00
sigma_e                                   0.00
class_IMB                                 0.00
class_MBA                                 0.00
class_MCA                                 0.00
class_OMB                                 0.00
class_TJN                                 0.00
class_TNO                                 0.00
sigma_a                                   0.00
class_IEO                                 0.00
sigma_n                                   0.02
a_Semi_major_axis                         0.04
Normalized RMS                            0.06
om_Longitude_ascending_node               0.06
n_Mean_motion                             0.08
e_Eccentricity                            0.08
ma_Mean_anomaly                           0.10
Minimum_Orbit_Intersection_Distance_au    0.12
i_Inclination                             0.20
H_Absolute_magnitude_parameter            0.24
```

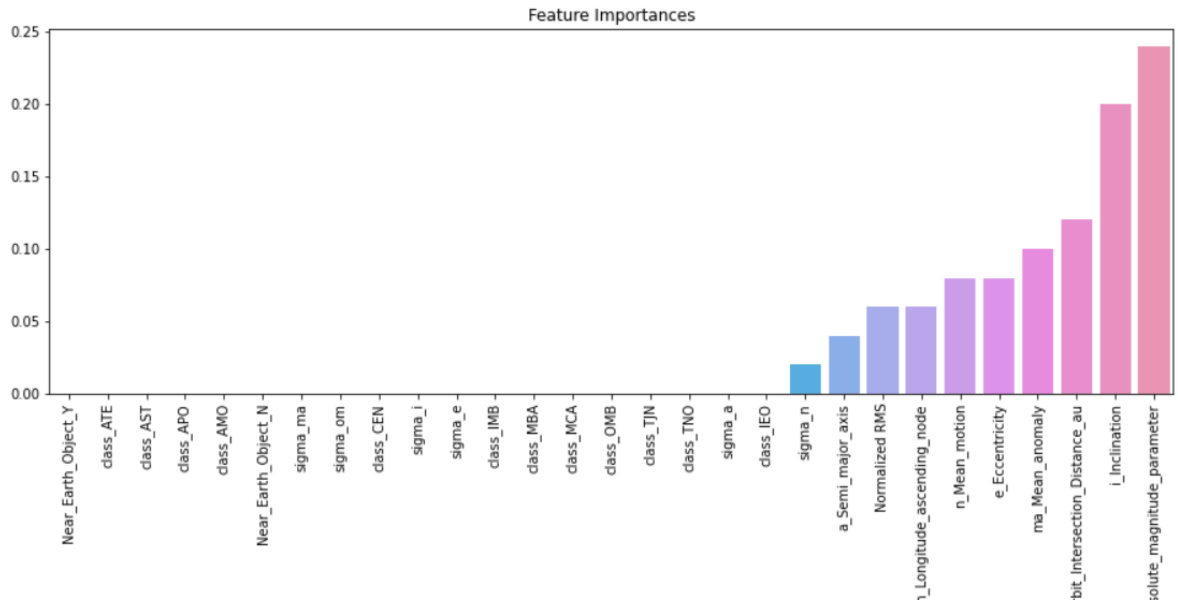


Fig 53. Ada Boost Classifier feature importance.

Gradient Boosting Classifier

Building on the work of Leo Breiman, Jerome H. Friedman developed gradient boosting, which works by sequentially adding predictors to an ensemble with each one correcting for the errors of its predecessor. However, instead of changing weights of data points like AdaBoost, the gradient boosting trains on the residual errors of the previous predictor. The name, gradient boosting, is used since it combines the gradient descent algorithm and boosting method. (IBM Cloud Education, 2021)

Python code:

```
model4 = GradientBoostingClassifier()
model4.fit(X_train,y_train)
y_preds = model4.predict(X_valid)
y_preds1 = model4.predict(X_train)

print(classification_report(y_valid,y_preds))

print(classificationSummary(y_train,y_preds1))
print(classificationSummary(y_valid,y_preds))
```

```
              precision    recall  f1-score   support

           N       1.00      1.00      1.00    279083
           Y       0.99      0.99      0.99       618

    accuracy                           1.00    279701
   macro avg       0.99      1.00      0.99    279701
weighted avg       1.00      1.00      1.00    279701

Confusion Matrix (Accuracy 0.9999)

       Prediction
Actual      0      1
     0 651161     25
     1     13   1435
None
Confusion Matrix (Accuracy 0.9999)

       Prediction
Actual      0      1
     0 279074      9
     1      6    612
```

Fig 54. Gradient Boosting Classifier classification report

```
feat_importances = pd.DataFrame(model4.feature_importances_, index=X.columns)
print(feat_importances.sort_values(0))

plt.figure(figsize=(15,5))
plt.xticks(rotation=90)
plt.title('Feature Importances')
sns.barplot(data= feat_importances.sort_values(0).T);
```

Near_Earth_Object_Y	0.000000
class_CEN	0.000000
class_ATE	0.000000
class_AST	0.000000
class_APO	0.000000
class_AMO	0.000000
Near_Earth_Object_N	0.000000
sigma_ma	0.000000
sigma_om	0.000000
class_IEO	0.000000
sigma_i	0.000000
sigma_e	0.000000
class_MBA	0.000000
class_MCA	0.000000
class_OMB	0.000000
class_TJN	0.000000
class_TNO	0.000000
sigma_a	0.000000
class_IMB	0.000000
om_Longitude_ascending_node	0.000003
sigma_n	0.000009
ma_Mean_anomaly	0.000014
Normalized RMS	0.000033
e_Eccentricity	0.000053
i_Inclination	0.000652
a_Semi_major_axis	0.007465
n_Mean_motion	0.013853
Minimum_Orbit_Intersection_Distance_au	0.141250
H_Absolute_magnitude_parameter	0.836669

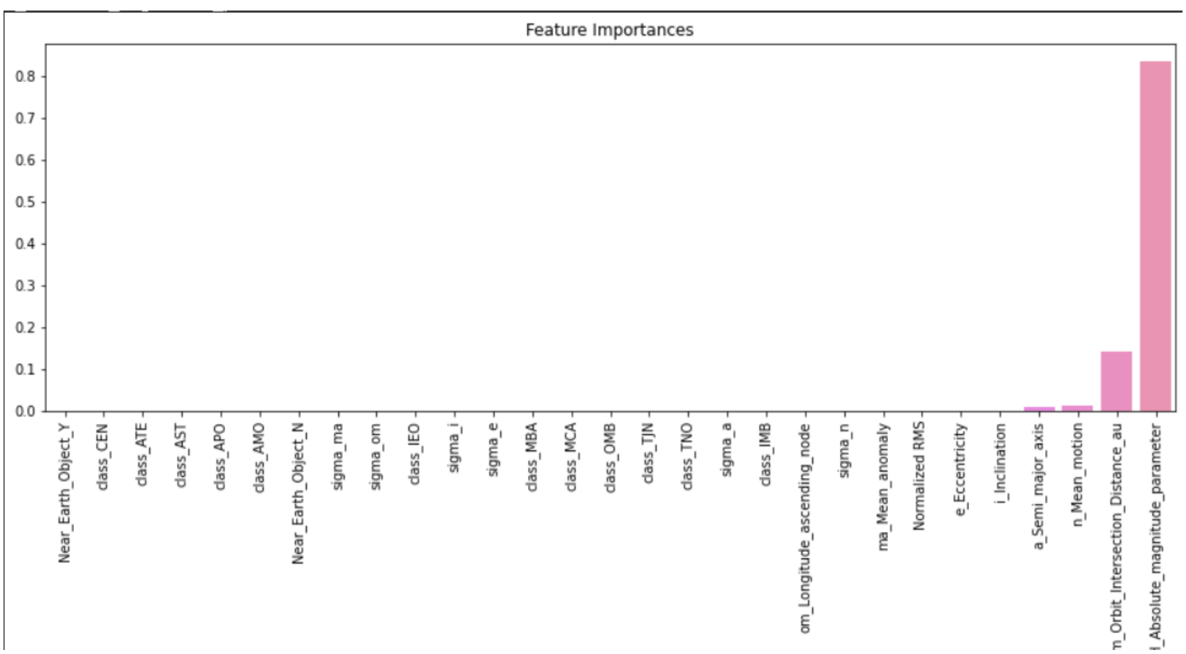


Fig 55. Gradient Boosting Classifier feature importance.

XGB Classifier

XGBoost is an implementation of gradient boosting that is designed for computational speed and scale. XGBoost leverages multiple cores on the CPU, allowing for learning to occur in parallel during training. ("What is Boosting? | IBM")

Python code:

```
model3 = XGBClassifier()
model3.fit(X_train,y_train)
y_preds = model3.predict(X_valid)
y_preds1 = model.predict(X_train)

print(classification_report(y_valid,y_preds))

print(classificationSummary(y_train,y_preds1))
print(classificationSummary(y_valid,y_preds))
```

```
              precision    recall  f1-score   support

           N       1.00      1.00      1.00    279083
           Y       0.99      0.98      0.99       618

    accuracy                           1.00    279701
   macro avg       0.99      0.99      0.99    279701
weighted avg       1.00      1.00      1.00    279701

Confusion Matrix (Accuracy 0.9981)

       Prediction
Actual      0      1
     0 650964    222
     1   1009    439
None
Confusion Matrix (Accuracy 0.9999)

       Prediction
Actual      0      1
     0 279076      7
     1     11    607
```

Fig 56. XGBC Classifier classification report

```
feat_importances = pd.DataFrame(model3.feature_importances_, index=X.columns)
print(feat_importances.sort_values(0))

plt.figure(figsize=(15,5))
plt.xticks(rotation=90)
plt.title('Feature Importances')
sns.barplot(data= feat_importances.sort_values(0).T);
```

```
Near_Earth_Object_Y                     0.000000
class_CEN                               0.000000
class_ATE                               0.000000
class_AST                               0.000000
class_AMO                               0.000000
Near_Earth_Object_N                     0.000000
class_MBA                               0.000000
sigma_ma                                0.000000
class_IEO                               0.000000
class_MCA                               0.000000
sigma_e                                 0.000000
class_OMB                               0.000000
n_Mean_motion                           0.000000
class_TJN                               0.000000
class_TNO                               0.000000
class_IMB                               0.000000
class_APO                               0.002638
sigma_om                                0.004815
e_Eccentricity                          0.006265
sigma_a                                 0.006511
a_Semi_major_axis                       0.006592
i_Inclination                           0.007965
Normalized RMS                          0.008326
sigma_n                                 0.009549
om_Longitude_ascending_node             0.010456
sigma_i                                 0.010711
ma_Mean_anomaly                         0.010836
Minimum_Orbit_Intersection_Distance_au  0.269408
H_Absolute_magnitude_parameter          0.645927
```

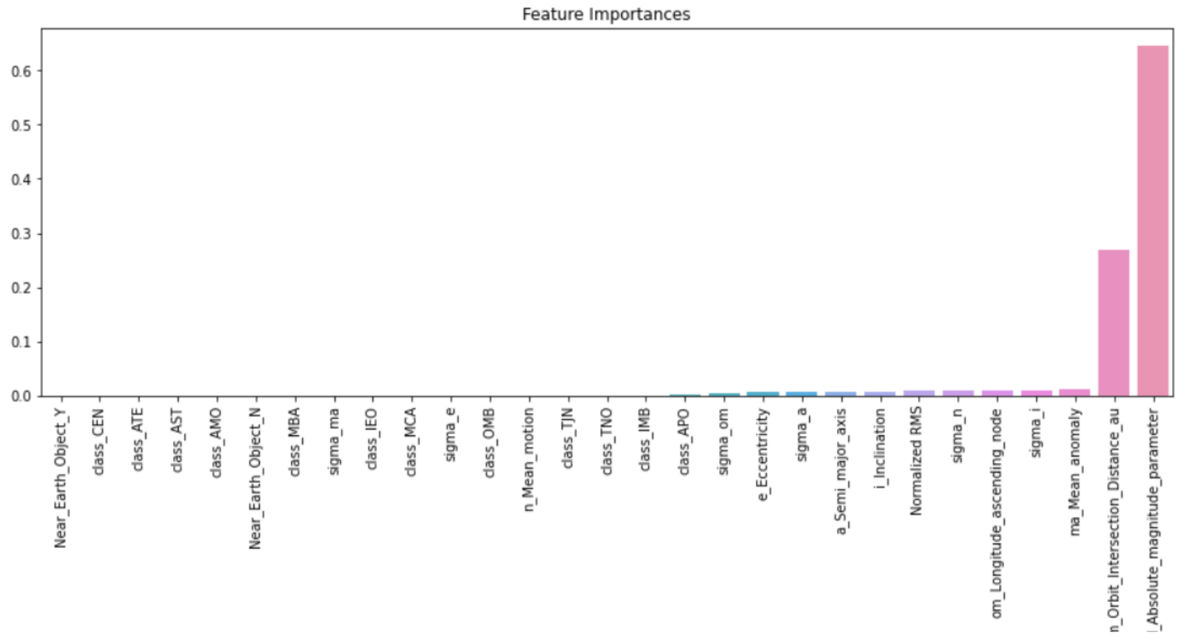


Fig 57. XGBC Classifier feature importance.

XGBRF Classifier

XGBoost is an implementation of gradient boosting that is designed for computational speed and scale. XGBoost leverages multiple cores on the CPU, allowing for learning to occur in parallel during training. (“What is Boosting? | IBM”) Used with decision trees.

Python code:

```
model5 = XGBRFClassifier()
model5.fit(X_train,y_train)
y_preds = model5.predict(X_valid)
y_preds1 = model4.predict(X_train)

print(classification_report(y_valid,y_preds))

print(classificationSummary(y_train,y_preds1))
print(classificationSummary(y_valid,y_preds))
```

```
              precision    recall  f1-score   support

           N       1.00      1.00      1.00    279083
           Y       0.99      0.99      0.99       618

    accuracy                           1.00    279701
   macro avg       0.99      1.00      0.99    279701
weighted avg       1.00      1.00      1.00    279701

Confusion Matrix (Accuracy 0.9999)

       Prediction
Actual      0      1
     0 651161     25
     1     13   1435
None
Confusion Matrix (Accuracy 0.9999)

       Prediction
Actual      0      1
     0 279074      9
     1      6    612
```

Fig 58. XGBRF Classifier classification report.

```
feat_importances = pd.DataFrame(model5.feature_importances_, index=X.columns)
print(feat_importances.sort_values(0))

plt.figure(figsize=(15,5))
plt.xticks(rotation=90)
plt.title('Feature Importances')
sns.barplot(data= feat_importances.sort_values(0).T);
```

```
Normalized RMS                          0.000000
class_TNO                               0.000000
class_TJN                               0.000000
class_OMB                               0.000000
class_MCA                               0.000000
class_MBA                               0.000000
class_IMB                               0.000000
class_IEO                               0.000000
class_CEN                               0.000000
class_ATE                               0.000000
class_AST                               0.000000
sigma_ma                                0.000000
Near_Earth_Object_Y                     0.000000
n_Mean_motion                           0.000000
ma_Mean_anomaly                         0.000000
om_Longitude_ascending_node             0.000000
sigma_om                                0.000000
e_Eccentricity                          0.007530
a_Semi_major_axis                       0.009974
i_Inclination                           0.023462
Near_Earth_Object_N                     0.057155
sigma_i                                 0.071936
class_APO                               0.075642
class_AMO                               0.092245
sigma_n                                 0.118028
sigma_e                                 0.118513
Minimum_Orbit_Intersection_Distance_au  0.124622
sigma_a                                 0.127227
H_Absolute_magnitude_parameter          0.173667
```

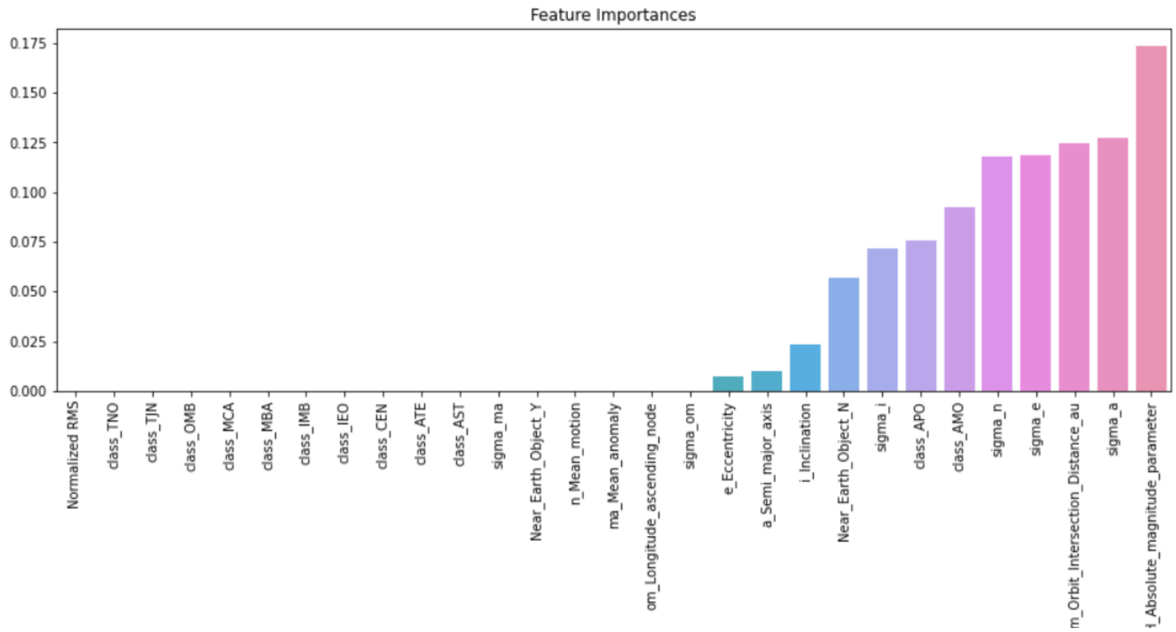


Fig 59. XGBRF Classifier feature importance.

Model Comparison

Python code:

```
def fit_and_score(models, X_train, X_valid, y_train, y_valid):
    np.random.seed(1)

    model_scores = {}

    for name, model in models.items():
        model.fit(X_train,y_train)
        model_scores[name] = model.score(X_valid,y_valid)

    model_scores = pd.DataFrame(model_scores, index=['Accuracy']).transpose()
    model_scores = model_scores.sort_values('Accuracy')

    return model_scores
```

```
models = {'LogisticRegression': LogisticRegression(max_iter=100000),
          'DecisionTreeClassifier': DecisionTreeClassifier(),
          'RandomForestClassifier': RandomForestClassifier(),
          'AdaBoostClassifier': AdaBoostClassifier(),
          'GradientBoostingClassifier': GradientBoostingClassifier(),
          'XGBClassifier': XGBClassifier(),
          'XGBRFClassifier': XGBRFClassifier()}
```

```
baseline_model_scores = fit_and_score(models, X_train, X_valid, y_train, y_valid)
baseline_model_scores
```

	Accuracy
LogisticRegression	0.998130
DecisionTreeClassifier	0.999893
AdaBoostClassifier	0.999932
XGBClassifier	0.999936
GradientBoostingClassifier	0.999946
XGBRFClassifier	0.999946
RandomForestClassifier	0.999950

Fig 60. Model comparison.

SAS Enterprise Miner

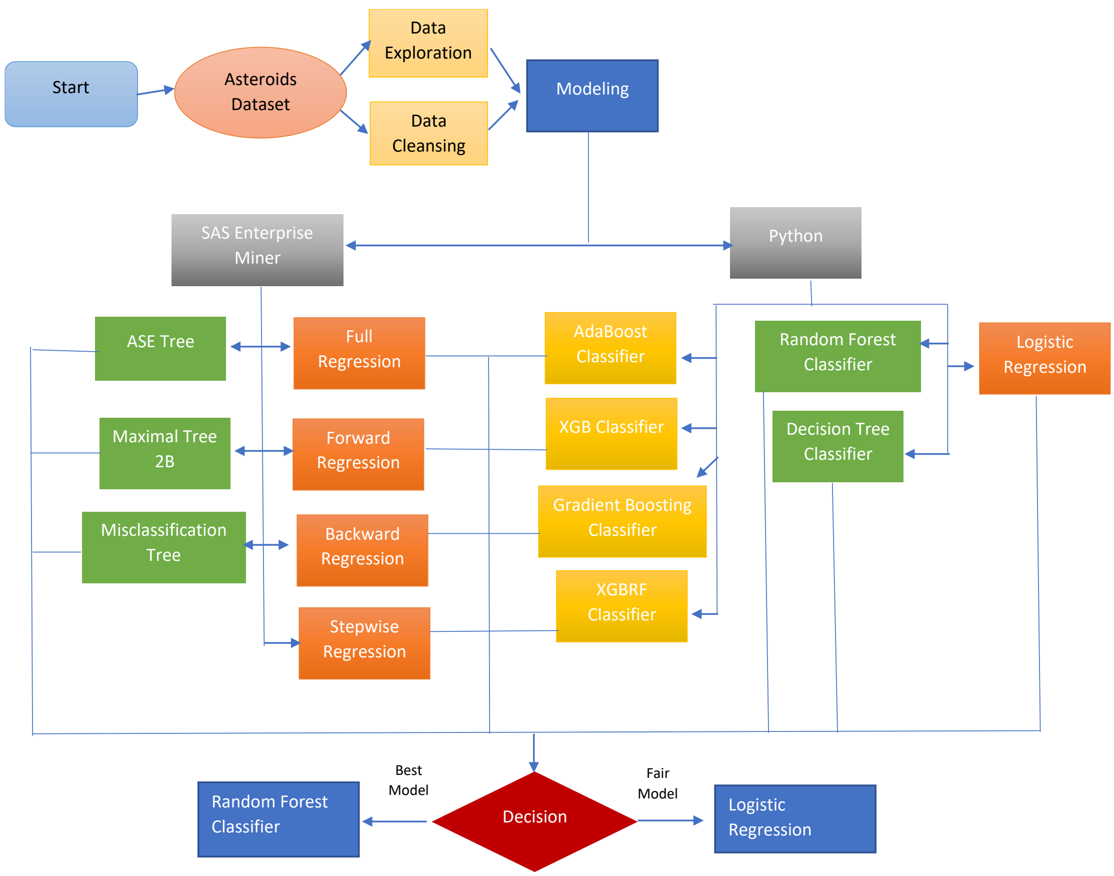
Fit Statistics

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Valid: Maximum Absolute Error	Valid: Root Average Squared Error	Valid: Average Squared Error	Valid: Root Mean Square Error	Valid: Mean Square Error	Valid: Gini Coefficient	Valid: Bin-Based Two-Way Kolmogorov-Smirnov Statistic
Y	Tree3	Tree3	ASE tree	Potentially Hazardous Asteroid	0.999034	0.005553	3.083E-5	.	.	1	0.952
	Tree2	Tree2	Misclassification tree	Potentially Hazardous Asteroid	0.997763	0.00702	4.928E-5	.	.	1	0.952
	Reg3	Reg3	Backward Regression	Potentially Hazardous Asteroid	0.999722	0.025976	.0006748	0.025976	.0006748	0.999	0.952
	Reg4	Reg4	Forward Regression	Potentially Hazardous Asteroid	0.999722	0.025976	.0006748	0.025976	.0006748	0.999	0.952
	Reg	Reg	Stepwise Regression	Potentially Hazardous Asteroid	0.999699	0.025996	.0006758	0.025996	.0006758	0.999	0.952
	Reg2	Reg2	Full Regression	Potentially Hazardous Asteroid	0.99963	0.026082	.0006802	0.026082	.0006802	0.999	0.952
	Tree	Tree	Maximal tree 2B	Potentially Hazardous Asteroid	0.999898	0.040356	0.001629	.	.	0.935	0.896

Fig 63. Model comparison.

Model Recommendation

Model Selection



Model Tables:

SAS Enterprise Miner:

Model Name	Gini Coefficient	Bin-Based Kolmogorov-Smirnov Statistic	ASE	RASE	MSE	RMSE	MAE
ASE Tree	1	0.952	0.00003083	0.005553	0	0	0.999034
Misclassification Tree	1	0.952	0.00004928	0.00702	0	0	0.997763
Backward Regression	0.999	0.952	0.0006748	0.025976	0.000675	0.025976	0.999722
Forward Regression	0.999	0.952	0.0006748	0.025976	0.000675	0.025976	0.999722
Stepwise Regression	0.999	0.952	0.0006748	0.025996	0.000676	0.025996	0.999699
Full Regression	0.999	0.952	0.0006802	0.026082	0.00068	0.026082	0.99963
Maximal Tree 2B	0.935	0.896	0.001629	0.040356	0	0	0.999898

Python:

Model Name	Training Accuracy	Validation Accuracy	F1 Score
Logistic Regression	0.9981	0.9981	0.71
Decision Tree Classifier	0.9981	0.9999	0.99
AdaBoost Classifier	0.9981	0.9999	0.99
XGB Classifier	0.9981	0.999	0.99
XGBRF Classifier	0.999	0.999	0.99
Gradient Boosting Classifier	0.9999	0.9999	0.99
Random Forest	0.9999	1	0.99

As is shown, the best model predictions are the Random Forest model presenting the highest accuracy of 0.999950, a f1-score of 1 for 'N' and 0.99 for 'Y'. The variables that worth with almost 30% of participation in these models are Minimum_Orbit_Intersection_Distance and Absolute_Magnitude_Parameter.

This model was the best because is considering a combination of decision trees without pruning for sampling and hence. The dataset analyzed is a huge size dataset, the random forest can manage large data sets due to its capability to work with many variables running to thousands.

Model Assumptions and Limitations

The model analyzed is considered a deterministic model, which allows the analyst to calculate a future event exactly, without the involvement of randomness. ("Stochastic vs Deterministic Models: Understand the Pros and Cons") The model has all the necessary data to predict the outcome with certainty.

This is why all the models run presents the same accuracy for training and validation and it will not present a variable drift neither a variable drift monitoring will be necessary.

Model Sensitivity to Key Drivers

a. The accuracy of positive and negative predictions is 1.00
b. Fraction of positives that were correctly identified is 0.98.
c. Fraction of negatives that were correctly identified is 1.00.
d. The percent of correct positive predictions is 0.99.
e. The percent of correct positive predictions is 1.00.

Conclusion

In summary, this project is considered as a deterministic model because it is about a science topic, it has a minimum risk but with a high impact. Asteroids have a lot of variables to consider for an analysis, in this case study that the target is only to identify which one could collide to Earth the worth variables are the minimum orbit intersection distance and the absolute magnitude parameter. The best model was the Random Forest Classifier because this model oversees the multicollinearity and the missing values in a better way. Also, at the end of this analysis is known that to consider an asteroid as a potentially hazardous it is not necessary that this asteroid is a near earth object.

If the scientific community wants to know the dimension of an impact or when a pha will collide to Earth, other analysis must be conducted.

This project is considered as a minimum risk but with a high impact because whether those asteroids will not be detected on time, it will be dangerous and in the worst cases could be mortal for the entire humanity, because could exist some asteroid as the one that collided with Earth (Yucatan, Mexico) that produced the dinosaur extinction.

For this reason, is extremely important to check this information periodically, to be informed for any changes in the universe that could change everything, such as the dead of a neutron star that could produce a blackhole, and other factors that could impact and change to every information collected.

Recommendations

The JPL employees must actualize each two hundred days the information provided in the "Asteroids" dataset, because of the epoch of osculation (epoch is a moment in time used as a reference point for some time-varying astronomical quantity), for the next variables change the data: eccentricity, semi-major axis, inclination, long ascending node, argument of periapsis and true anomaly and these elements could change due to universe movements.

The data analyst must know the terminology mentioned in the data to interpret it and know which variables could cause multicollinearity and which are not relevant. Share the final result with Space Agencies and other parties, in case that the analysis result will be urgent because of the risk, then quick actions must be taken. Generally, the management team of Space entities must know the protocols that need to be taken and notice to governments. In some cases, depending on the asteroid size, NASA own spacecrafts to destroy asteroids.

In the data analysis, it will be important to give a special attention to moid and H variables. Considering that the asteroid and Earth could pass through the moid at the same time and the collision risk could increase, in case of the H will determine with the light perceived the closeness with Earth.

The majority of the space data must be analyzed periodically by scientists, astrophysicists, astronomers to continue investigating all that happens in the universe.

References

Ascending node: Cosmos. Ascending Node | COSMOS. (n.d.). Retrieved July 29, 2022, from https://astronomy.swin.edu.au/cosmos/A/Ascending+Node

Hossain, M. S. (2022, July 22). Asteroid dataset. Kaggle. Retrieved July 29, 2022, from https://www.kaggle.com/sakhawat18/asteroid-dataset

Jee, C. (2022, April 15). A huge asteroid flew very close to Earth last week. how did we miss it? MIT Technology Review. Retrieved July 29, 2022, from https://www.technologyreview.com/2019/07/29/134013/a-huge-asteroid-flew-very-close-to-earth-last-week-how-did-we-miss-it/

Monzon, I. (2020, November 8). What happens to Earth if an asteroid destroys the Moon? International Business Times. Retrieved July 29, 2022, from https://www.ibtimes.com/what-happens-earth-if-asteroid-destroys-moon-2839516#:~:text=If%20it%20ends%20up%20getting%20hit%20by%20a,send%20huge%20chunks%20of%20debris%20barreling%20towards%20Earth.

NASA. (n.d.). Extras. NASA. Retrieved July 29, 2022, from https://cneos.jpl.nasa.gov/extras.html

NASA. (n.d.). History. NASA. Retrieved July 29, 2022, from https://www.jpl.nasa.gov/who-we-are/history

NASA. (n.d.). Orbit. NASA. Retrieved July 29, 2022, from https://ssd.jpl.nasa.gov/glossary/orbit.html

NASA. (n.d.). Small-body database query. NASA. Retrieved July 29, 2022, from https://ssd.jpl.nasa.gov/tools/sbdb_query.html#!#results

News/Current Events. NASA unveils plan to Test Asteroid Defense Technique. (n.d.). Retrieved July 29, 2022, from https://freerepublic.com/focus/f-news/3565892/posts

Wikimedia Foundation. (2022, July 23). Osculating orbit. Wikipedia. Retrieved July 29, 2022, from https://en.wikipedia.org/wiki/Osculating_orbit

Wikimedia Foundation. (2022, July 6). Mean anomaly. Wikipedia. Retrieved July 29, 2022, from https://en.wikipedia.org/wiki/Mean_anomaly

Wikimedia Foundation. (2022, June 26). Minimum orbit intersection distance. Wikipedia. Retrieved July 29, 2022, from https://en.wikipedia.org/wiki/Minimum_orbit_intersection_distance

Lance Wills, H. of A. I. G. @D. (2022, July 1). What is concept drift? Model Drift in machine learning. Datatron. Retrieved August 9, 2022, from https://datatron.com/what-is-model-drift/#:~:text=The%20most%20accurate%20way%20to%20detect%20model%20drift,deviate%20farther%20and%20farther%20from%20the%20actual%20values.

Logistic regression: What is logistic regression and why do we need it? Analytics Vidhya. (2021, August 26). Retrieved August 9, 2022, from https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/

NASA. (2022, April 19). Asteroids. NASA. Retrieved August 9, 2022, from https://solarsystem.nasa.gov/asteroids-comets-and-meteors/asteroids/overview/?page=0&per_page=40&order=name%2Basc&search=&condition_1=101%3Aparent_id&condition_2=asteroid%3Abody_type%3Ailike

Random Forest. Corporate Finance Institute. (2021, September 2). Retrieved August 9, 2022, from https://corporatefinanceinstitute.com/resources/knowledge/other/random-forest/

Shendre, S. (2020, May 14). Model Drift in machine learning models. Medium. Retrieved August 9, 2022, from https://towardsdatascience.com/model-drift-in-machine-learning-models-8f7e7413b563

Stobierski, T. (2021, August 11). What is statistical modeling for data analysis? Northeastern University Graduate Programs. Retrieved August 9, 2022, from https://www.northeastern.edu/graduate/blog/statistical-modeling-for-data-analysis/#:~:text=Data%20analysts%20use%20regression%20models%20to%20examine%20relationships,can%20be%20leveraged%20to%20make%20essential%20business%20decisions.

Wikimedia Foundation. (2022, August 6). Asteroid. Wikipedia. Retrieved August 9, 2022, from https://en.wikipedia.org/wiki/Asteroid

Stochastic vs deterministic models: Understand the pros and cons. Blog. (n.d.). Retrieved August 12, 2022, from https://blog.ev.uk/stochastic-vs-deterministic-models-understand-the-pros-and-cons#:~:text=Deterministic%20%28from%20determinism%2C%20which%20means%20lack%20of%20free,necessary%20to%20predict%20%28determine%29%20the%20outcome%20with%20certainty.