Data Preprocessing.

In [ ]:

```
import pandas as pd

df.columns
df = pd.read_csv("nasa.csv")
drop_column = ["Neo Reference ID", "Name", "Equinox", 'Est Dia in KM(min)
       'Est Dia in Miles(max)', 'Est Dia in Feet(min)', 'Est Dia in Feet(
df = df.drop(drop_column, axis = 1)

df.columns
```

Out[ ]:

```
Index(['Absolute Magnitude', 'Est Dia in M(min)', 'Est Dia in M(max)',
       'Epoch Date Close Approach', 'Relative Velocity km per sec',
       'Relative Velocity km per hr', 'Miles per hour',
       'Miss Dist.(Astronomical)', 'Miss Dist.(lunar)',
       'Miss Dist.(kilometers)', 'Miss Dist.(miles)', 'Orbiting Body',
       'Orbit ID', 'Orbit Uncertainity', 'Minimum Orbit Intersection',
       'Jupiter Tisserand Invariant', 'Epoch Osculation', 'Eccentricit
y',
       'Semi Major Axis', 'Inclination', 'Asc Node Longitude',
       'Orbital Period', 'Perihelion Distance', 'Perihelion Arg',
       'Aphelion Dist', 'Perihelion Time', 'Mean Anomaly', 'Mean Motio
n',
       'Hazardous'],
      dtype='object')
```

In [ ]:

```
hazard = {
    True : 1,
    False : 0
}


df['Hazardous'] = df['Hazardous'].map(hazard)

orb = {
    'Earth' : 1
}

df['Orbiting Body'] = df['Orbiting Body'].map(orb)
df
```

Out[ ]:

	Absolute Magnitude	Est Dia in M(min)	Est Dia in M(max)	Epoch Date Close Approach	Relative Velocity km per sec	Relativ Velocity kr per h
0	21.600	127.219878	284.472297	7.890000e+11	6.115834	22017.0038
1	21.300	146.067964	326.617897	7.890000e+11	18.113985	65210.3460
2	20.300	231.502122	517.654482	7.900000e+11	7.590711	27326.5601
3	27.400	8.801465	19.680675	7.900000e+11	11.173875	40225.9481
4	21.600	127.219878	284.472297	7.900000e+11	9.840831	35426.9917
...	...	...	...	...	...	.
4682	23.900	44.111820	98.637028	1.470000e+12	22.154265	79755.3542
4683	28.200	6.089126	13.615700	1.470000e+12	3.225150	11610.5395
4684	22.700	76.657557	171.411509	1.470000e+12	7.191642	25889.9106
4685	21.800	116.025908	259.441818	1.470000e+12	11.352090	40867.5223
4686	19.109	400.640618	895.859655	1.470000e+12	35.946852	129408.6663

4687 rows × 29 columns

In [ ]:

```
df.columns
```

Out[ ]:

```
Index(['Absolute Magnitude', 'Est Dia in M(min)', 'Est Dia in M(max)',
       'Epoch Date Close Approach', 'Relative Velocity km per sec',
       'Relative Velocity km per hr', 'Miles per hour',
       'Miss Dist.(Astronomical)', 'Miss Dist.(lunar)',
       'Miss Dist.(kilometers)', 'Miss Dist.(miles)', 'Orbiting Body',
       'Orbit ID', 'Orbit Uncertainity', 'Minimum Orbit Intersection',
       'Jupiter Tisserand Invariant', 'Epoch Osculation', 'Eccentricit
y',
       'Semi Major Axis', 'Inclination', 'Asc Node Longitude',
       'Orbital Period', 'Perihelion Distance', 'Perihelion Arg',
       'Aphelion Dist', 'Perihelion Time', 'Mean Anomaly', 'Mean Motio
n',
       'Hazardous'],
      dtype='object')
```

In [ ]:

```
df.info()
df.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4687 entries, 0 to 4686
Data columns (total 29 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Absolute Magnitude            4687 non-null   float64
 1   Est Dia in M(min)             4687 non-null   float64
 2   Est Dia in M(max)             4687 non-null   float64
 3   Epoch Date Close Approach     4687 non-null   float64
 4   Relative Velocity km per sec  4687 non-null   float64
 5   Relative Velocity km per hr   4687 non-null   float64
 6   Miles per hour                4687 non-null   float64
 7   Miss Dist.(Astronomical)      4687 non-null   float64
 8   Miss Dist.(lunar)             4687 non-null   float64
 9   Miss Dist.(kilometers)        4687 non-null   float64
 10  Miss Dist.(miles)             4687 non-null   float64
 11  Orbiting Body                 4687 non-null   int64  
 12  Orbit ID                      4687 non-null   int64  
 13  Orbit Uncertainity            4687 non-null   int64  
 14  Minimum Orbit Intersection    4687 non-null   float64
 15  Jupiter Tisserand Invariant   4687 non-null   float64
 16  Epoch Osculation              4687 non-null   float64
 17  Eccentricity                  4687 non-null   float64
 18  Semi Major Axis               4687 non-null   float64
 19  Inclination                   4687 non-null   float64
 20  Asc Node Longitude            4687 non-null   float64
 21  Orbital Period                4687 non-null   float64
 22  Perihelion Distance           4687 non-null   float64
 23  Perihelion Arg                4687 non-null   float64
 24  Aphelion Dist                 4687 non-null   float64
 25  Perihelion Time               4687 non-null   float64
 26  Mean Anomaly                  4687 non-null   float64
 27  Mean Motion                   4687 non-null   float64
 28  Hazardous                     4687 non-null   int64  
dtypes: float64(25), int64(4)
memory usage: 1.0 MB
```

Out[ ]:

```
Absolute Magnitude              0
Est Dia in M(min)               0
Est Dia in M(max)               0
Epoch Date Close Approach       0
Relative Velocity km per sec    0
Relative Velocity km per hr     0
Miles per hour                  0
Miss Dist.(Astronomical)        0
Miss Dist.(lunar)               0
Miss Dist.(kilometers)          0
Miss Dist.(miles)               0
Orbiting Body                   0
Orbit ID                        0
Orbit Uncertainity              0
Minimum Orbit Intersection      0
Jupiter Tisserand Invariant     0
Epoch Osculation                0
Eccentricity                    0
Semi Major Axis                 0
Inclination                     0
Asc Node Longitude              0
Orbital Period                  0
Perihelion Distance             0
Perihelion Arg                  0
Aphelion Dist                   0
Perihelion Time                 0
Mean Anomaly                    0
Mean Motion                     0
Hazardous                       0
dtype: int64
```

In [ ]:

```python
df.columns
X = ['Absolute Magnitude', 'Est Dia in M(min)', 'Est Dia in M(max)',
        'Epoch Date Close Approach',
       'Relative Velocity km per sec', 'Relative Velocity km per hr',

       'Miles per hour', 'Miss Dist.(Astronomical)', 'Miss Dist.(lunar)',
       'Miss Dist.(kilometers)', 'Miss Dist.(miles)', 'Orbiting Body',
       'Orbit ID', 'Orbit Uncertainity',
       'Minimum Orbit Intersection', 'Jupiter Tisserand Invariant',
       'Epoch Osculation', 'Eccentricity', 'Semi Major Axis', 'Inclinatio
       'Asc Node Longitude', 'Orbital Period', 'Perihelion Distance',
       'Perihelion Arg', 'Aphelion Dist', 'Perihelion Time', 'Mean Anomal
       'Mean Motion']

Y = ['Hazardous']
```

Analysisng dataset

In [ ]:

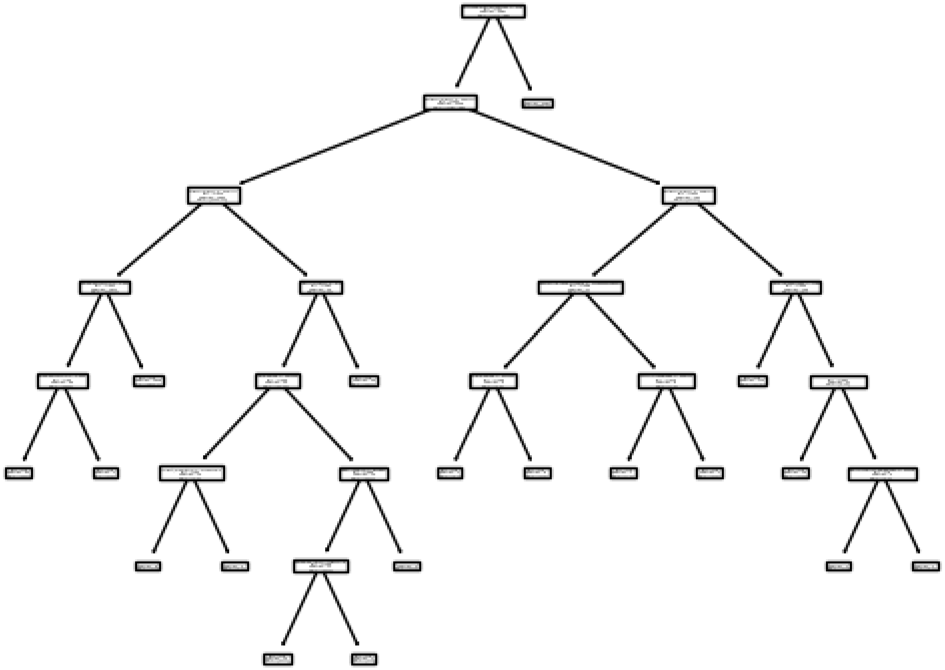
```python
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt

Features = df[X]
Target = df[Y]
```

```


clf = DecisionTreeClassifier()
clf.fit(Features, Target)
tree.plot_tree(clf, feature_names=X)
```

```
Out[ ]: [Text(0.5217391304347826, 0.9375, 'Minimum Orbit Intersection <= 0.05\ng
        ini = 0.27\nsamples = 4687\nvalue = [3932, 755]'),
         Text(0.4782608695652174, 0.8125, 'Est Dia in M(max) <= 232.861\ngini =
        0.431\nsamples = 2406\nvalue = [1651, 755]'),
         Text(0.2391304347826087, 0.6875, 'Est Dia in M(max) <= 210.938\ngini =
        0.014\nsamples = 1654\nvalue = [1642, 12]'),
         Text(0.13043478260869565, 0.5625, 'Perihelion Distance <= 0.437\ngini =
        0.001\nsamples = 1611\nvalue = [1610, 1]'),
         Text(0.08695652173913043, 0.4375, 'Perihelion Distance <= 0.433\ngini =
        0.03\nsamples = 65\nvalue = [64, 1]'),
         Text(0.043478260869565216, 0.3125, 'gini = 0.0\nsamples = 64\nvalue =
        [64, 0]'),
         Text(0.13043478260869565, 0.3125, 'gini = 0.0\nsamples = 1\nvalue = [0,
        1]'),
         Text(0.17391304347826086, 0.4375, 'gini = 0.0\nsamples = 1546\nvalue =
        [1546, 0]'),
         Text(0.34782608695652173, 0.5625, 'Orbit Uncertainity <= 2.5\ngini = 0.
        381\nsamples = 43\nvalue = [32, 11]'),
         Text(0.30434782608695654, 0.4375, 'Orbital Period <= 387.494\ngini = 0.
        499\nsamples = 23\nvalue = [12, 11]'),
         Text(0.21739130434782608, 0.3125, 'Miss Dist.(kilometers) <= 67029004.0
        \ngini = 0.32\nsamples = 10\nvalue = [2, 8]'),
         Text(0.17391304347826086, 0.1875, 'gini = 0.0\nsamples = 8\nvalue = [0,
        8]'),
         Text(0.2608695652173913, 0.1875, 'gini = 0.0\nsamples = 2\nvalue = [2,
        0]'),
         Text(0.391304347826087, 0.3125, 'Mean Anomaly <= 351.851\ngini = 0.355
        \nsamples = 13\nvalue = [10, 3]'),
         Text(0.34782608695652173, 0.1875, 'Perihelion Time <= 2458207.0\ngini =
        0.165\nsamples = 11\nvalue = [10, 1]'),
         Text(0.30434782608695654, 0.0625, 'gini = 0.0\nsamples = 10\nvalue = [1
        0, 0]'),
         Text(0.391304347826087, 0.0625, 'gini = 0.0\nsamples = 1\nvalue = [0,
        1]'),
         Text(0.43478260869565216, 0.1875, 'gini = 0.0\nsamples = 2\nvalue = [0,
        2]'),
         Text(0.391304347826087, 0.4375, 'gini = 0.0\nsamples = 20\nvalue = [20,
        0]'),
         Text(0.717391304347826, 0.6875, 'Est Dia in M(max) <= 238.869\ngini =
        0.024\nsamples = 752\nvalue = [9, 743]'),
         Text(0.6086956521739131, 0.5625, 'Epoch Date Close Approach <= 11150000
        12800.0\ngini = 0.498\nsamples = 17\nvalue = [8, 9]'),
         Text(0.5217391304347826, 0.4375, 'Orbital Period <= 251.338\ngini = 0.1
        98\nsamples = 9\nvalue = [1, 8]'),
         Text(0.4782608695652174, 0.3125, 'gini = 0.0\nsamples = 1\nvalue = [1,
        0]'),
         Text(0.5652173913043478, 0.3125, 'gini = 0.0\nsamples = 8\nvalue = [0,
        8]'),
         Text(0.6956521739130435, 0.4375, 'Asc Node Longitude <= 268.774\ngini =
        0.219\nsamples = 8\nvalue = [7, 1]'),
         Text(0.6521739130434783, 0.3125, 'gini = 0.0\nsamples = 7\nvalue = [7,
        0]'),
         Text(0.7391304347826086, 0.3125, 'gini = 0.0\nsamples = 1\nvalue = [0,
        1]'),
         Text(0.8260869565217391, 0.5625, 'Absolute Magnitude <= 21.85\ngini =
        0.003\nsamples = 735\nvalue = [1, 734]'),
         Text(0.782608695652174, 0.4375, 'gini = 0.0\nsamples = 714\nvalue = [0,
        714]'),
         Text(0.8695652173913043, 0.4375, 'Perihelion Time <= 2458169.375\ngini
        = 0.091\nsamples = 21\nvalue = [1, 20]'),
         Text(0.8260869565217391, 0.3125, 'gini = 0.0\nsamples = 19\nvalue = [0,
        19]'),
         Text(0.9130434782608695, 0.3125, 'Relative Velocity km per sec <= 12.20
        5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
         Text(0.8695652173913043, 0.1875, 'gini = 0.0\nsamples = 1\nvalue = [0,
        1]'),
         Text(0.9565217391304348, 0.1875, 'gini = 0.0\nsamples = 1\nvalue = [1,
        0]'),
         Text(0.5652173913043478, 0.8125, 'gini = 0.0\nsamples = 2281\nvalue =
        [2281, 0]')]
```



Splitting Dataset

In [ ]:
```
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt

X_train,X_test,  Y_train, Y_test = train_test_split(Features, Target,
                                                     test_size=0.25,
                                                     random_state=45,
                                                     shuffle=True)
```

Support Vector Machine

In [ ]:
```
from sklearn.svm import SVC
svc_clf = SVC(kernel='poly')

svc_clf.fit(X_train, Y_train)
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-pac
kages/sklearn/utils/validation.py:1183: DataConversionWarning: A column-ve
ctor y was passed when a 1d array was expected. Please change the shape of
y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

Out[ ]:

```
▾ SVC
SVC(kernel='poly')
```

In [ ]:

```python
from sklearn.metrics import accuracy_score
predicted = svc_clf.predict(X_train)
print("Accuracy for training set : ", (accuracy_score(Y_train, predicted)
```

```
Accuracy for training set :  0.8378378378378378
```

In [ ]:

```python
from sklearn.metrics import classification_report
print( classification_report(Y_train, predicted))
```

```
              precision    recall  f1-score   support

           0       0.84      1.00      0.91      2945
           1       0.00      0.00      0.00       570

    accuracy                           0.84      3515
   macro avg       0.42      0.50      0.46      3515
weighted avg       0.70      0.84      0.76      3515
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-pac
kages/sklearn/metrics/_classification.py:1469: UndefinedMetricWarning: Pre
cision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-pac
kages/sklearn/metrics/_classification.py:1469: UndefinedMetricWarning: Pre
cision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-pac
kages/sklearn/metrics/_classification.py:1469: UndefinedMetricWarning: Pre
cision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [ ]:

```python
from sklearn.metrics import confusion_matrix
confusion_matrix(Y_test, predicted)
```

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
Cell In[160], line 2
      1 from sklearn.metrics import confusion_matrix
----> 2 confusion_matrix(Y_test, predicted)

File /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/sit
e-packages/sklearn/utils/_param_validation.py:211, in validate_params.<loc
als>.decorator.<locals>.wrapper(*args, **kwargs)
    205 try:
    206     with config_context(
    207         skip_parameter_validation=(
    208             prefer_skip_nested_validation or global_skip_validatio
n
    209         )
    210     ):
--> 211         return func(*args, **kwargs)
    212 except InvalidParameterError as e:
    213     # When the function is just a wrapper around an estimator, we
allow
    214     # the function to delegate validation to the estimator, but we
replace
    215     # the name of the estimator by the name of the function in the
error
    216     # message to avoid confusion.
    217     msg = re.sub(
    218         r"parameter of \w+ must be",
    219         f"parameter of {func.__qualname__} must be",
    220         str(e),
    221     )

File /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/sit
e-packages/sklearn/metrics/_classification.py:326, in confusion_matrix(y_t
rue, y_pred, labels, sample_weight, normalize)
    231 @validate_params(
    232     {
    233         "y_true": ["array-like"],
   (...)
    242     y_true, y_pred, *, labels=None, sample_weight=None, normalize=
None
    243 ):
    244     """Compute confusion matrix to evaluate the accuracy of a clas
sification.
    245 
    246     By definition a confusion matrix :math:`C` is such that :math:
`C_{i, j}`
   (...)
    324     (0, 2, 1, 1)
    325     """
--> 326     y_type, y_true, y_pred = _check_targets(y_true, y_pred)
    327     if y_type not in ("binary", "multiclass"):
    328         raise ValueError("%s is not supported" % y_type)

File /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/sit
e-packages/sklearn/metrics/_classification.py:84, in _check_targets(y_tru
e, y_pred)
     57 def _check_targets(y_true, y_pred):
     58     """Check that y_true and y_pred belong to the same classificat
```

```
ion task.
     59
     60     This converts multiclass or binary types to a common shape, an
d raises a
   (...)
     82     y_pred : array or indicator matrix
     83     """
---> 84     check_consistent_length(y_true, y_pred)
     85     type_true = type_of_target(y_true, input_name="y_true")
     86     type_pred = type_of_target(y_pred, input_name="y_pred")

File /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/sit
e-packages/sklearn/utils/validation.py:407, in check_consistent_length(*ar
rays)
    405 uniques = np.unique(lengths)
    406 if len(uniques) > 1:
--> 407     raise ValueError(
    408         "Found input variables with inconsistent numbers of sample
s: %r"
    409         % [int(l) for l in lengths]
    410     )

ValueError: Found input variables with inconsistent numbers of samples: [1
172, 3515]
```

Decision Tree

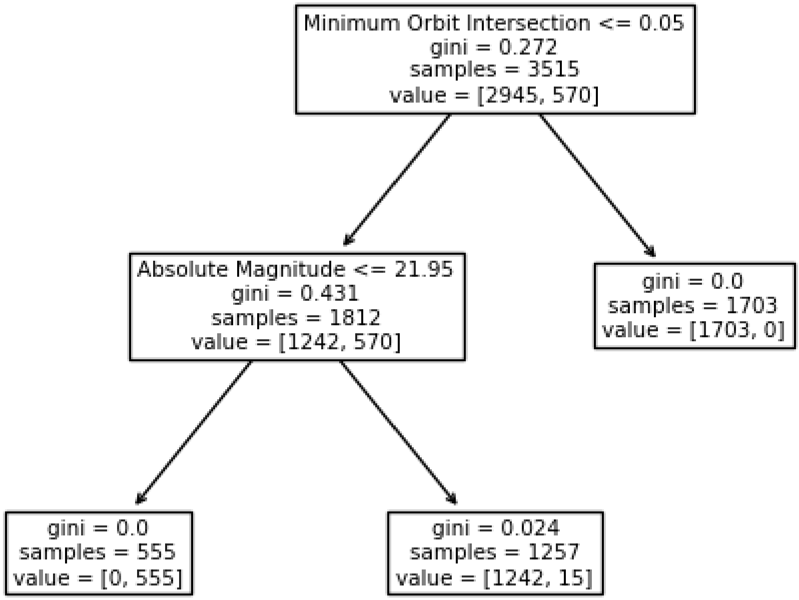
In [ ]:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
import matplotlib.pyplot as plt

clf.fit(X_train, Y_train)
tree.plot_tree(clf, feature_names=X)
```

Out[ ]:

```
[Text(0.6, 0.8333333333333334, 'Minimum Orbit Intersection <= 0.05\ngini
= 0.272\nsamples = 3515\nvalue = [2945, 570]'),
 Text(0.4, 0.5, 'Absolute Magnitude <= 21.95\ngini = 0.431\nsamples = 18
12\nvalue = [1242, 570]'),
 Text(0.2, 0.16666666666666666, 'gini = 0.0\nsamples = 555\nvalue = [0,
555]'),
 Text(0.6, 0.16666666666666666, 'gini = 0.024\nsamples = 1257\nvalue =
[1242, 15]'),
 Text(0.8, 0.5, 'gini = 0.0\nsamples = 1703\nvalue = [1703, 0]')]
```



In [ ]:

```
from sklearn.metrics import accuracy_score
predicted =  clf.predict(X_train)
print(accuracy_score(Y_train, predicted))
```

```
1.0
```

In [ ]:

```
from sklearn.metrics import accuracy_score
predicted =  clf.predict(X_test)
print(accuracy_score(Y_test, predicted))
```

```
1.0
```

Classification report

In [ ]:

```
print("Classification report : ")
from sklearn.metrics import classification_report

print(classification_report(Y_test, predicted))
```

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

           0       1.00      1.00      1.00       987
           1       1.00      1.00      1.00       185

    accuracy                           1.00      1172
   macro avg       1.00      1.00      1.00      1172
weighted avg       1.00      1.00      1.00      1172
```

Confusion matrix

In [ ]:

```
print("Confiusion Matrix : ")
from sklearn.metrics import confusion_matrix
confusion_matrix(Y_test, predicted)
```

```
       Confiusion Matrix :
Out[ ]: array([[987,   0],
               [  0, 185]])
```

```
In [ ]:
```