INNOVATIVE ASSIGNMENT - AI/ML 1CS101

Name: Chaudhary Rishika Dharmatma
Roll no:23BTM013
Batch: J1

Report: Analysis of NASA Asteroids Classification Dataset

Q.1

a) Selection of Data Domain:

The selected dataset falls within the domain of astronomy and space science. It focuses on classifying asteroids based on various characteristics such as size, velocity, orbit, and distance from Earth.

b) Data Characteristics:

To analyze the data characteristics, let's examine the dataset for null values, not null values, and unique values.

Null Values: We will identify any missing values in each column of the dataset.

Not Null Values: We will count the number of non-null values in each column.

Unique Values: We will determine the unique values present in each column to understand the variability of the data.

d) Output/Class Label:

The output or class label of the dataset is likely to be the classification of asteroids into different categories based on their characteristics. This classification could include parameters such as asteroid type or risk level.

e) Selection of Fields (Data Types):

The dataset contains several fields with different data types. Here's a breakdown of the data types for each field:

neo_reference_id: Integer (Unique identifier)

name: String (Asteroid name)

absolute_magnitude: Float (Magnitude measurement)

estimated_diameter (km):

min: Float (Minimum diameter in kilometers)

max: Float (Maximum diameter in kilometers)

estimated_diameter (m):

min: Float (Minimum diameter in meters)

max: Float (Maximum diameter in meters)

estimated_diameter (miles): Float (Diameter in miles)

estimated_diameter (feet): Float (Diameter in feet)

close_approach_date: Date (Date of closest approach)

epoch_date_close_approach: Integer (Epoch date of closest approach)

relative_velocity: Float (Relative velocity)

miles_per_hour: Float (Velocity in miles per hour)

miss_distance (astronomical): Float (Miss distance in astronomical units)

miss_distance (lunar): Float (Miss distance in lunar distances)

orbiting_body: String (Celestial body)

This report outlines the domain, characteristics, output label, and data types of the NASA Asteroids Classification dataset. It provides a comprehensive overview to facilitate further analysis and exploration of the dataset.

Q.2

```
import pandas as pd

import numpy as np

# Load the dataset

df = pd.read_csv('nasa.csv')

# Dropping null records

df = df.dropna()

#Replacing NA with the mean values in Est dia in mines(min) column

column_name = 'Est Dia in Miles(min)'

mean_value = df[column_name].mean()

df[column_name].fillna(mean_value, inplace=True)

# Replace empty values with the mode of the column

column_name = 'Epoch Date Close Approach'

mode_value = df[column_name].mode()[0]  #Calculate the mode (most
frequent value) of the column

df[column_name].fillna(mode_value, inplace=True)

# Save the cleaned dataset

df.to_csv('clear_nasa.csv', index=False)
```

Output: ***A new csv file named 'clear_nasa.csv' has been created and all the value has been replaced .***

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
1	Neo Refer	Name	Absolute M	Est Dia in K	Est Dia in K	Est Dia in M	Est Dia in M	Est Dia in M	Est Dia in M	Est Dia in F	Est Dia in F	Close Appr	Epoch Dat	Relative Ve	Relative Ve	Miles per h	Miss Dist.(	Miss Dist.(	Miss Dist.(	Miss Dist.(	Orbiting Bo	Orbit ID	Orbit Dete	Or
2	3703080	3703080	21.6	0.12722	0.284472	127.2199	284.4723	0.079051	0.176763	417.3881	933.3081	1/1/1995	7.89E+11	6.115834	22017	13680.51	0.419483	163.1787	62753692	38993336	Earth	17	########	
3	3723955	3723955	21.3	0.146068	0.326618	146.068	326.6179	0.090762	0.202951	479.2256	1071.581	1/1/1995	7.89E+11	18.11399	65210.35	40519.17	0.383014	148.9926	57298148	35603420	Earth	21	########	
4	2446862	2446862	20.3	0.231502	0.517654	231.5021	517.6545	0.143849	0.321655	759.5214	1698.342	1/8/1995	7.9E+11	7.590711	27326.56	16979.66	0.050956	19.82189	7622912	4736658	Earth	22	########	
5	3092506	3092506	27.4	0.008801	0.019681	8.801465	19.68067	0.005469	0.012229	28.8762	64.56914	########	7.9E+11	11.17387	40225.95	24994.84	0.285322	110.9904	42683616	26522368	Earth	7	########	
6	3514799	3514799	21.6	0.12722	0.284472	127.2199	284.4723	0.079051	0.176763	417.3881	933.3081	########	7.9E+11	9.840831	35426.99	22012.95	0.407832	158.6467	61010824	37910368	Earth	25	########	
7	3671135	3671135	19.6	0.319562	0.714562	319.5619	714.5621	0.198566	0.444008	1048.431	2344.364	########	7.9E+11	10.80884	38911.84	24178.3	0.392785	152.7933	58759768	36511628	Earth	40	########	
8	2495323	2495323	19.6	0.319562	0.714562	319.5619	714.5621	0.198566	0.444008	1048.431	2344.364	########	7.9E+11	10.80884	38911.83	24178.3	0.392783	152.7927	58759532	36511480	Earth	43	########	
9	2162463	2162463	17.8	0.732074	1.636967	732.074	1636.967	0.45489	1.017164	2401.818	5370.627	########	7.91E+11	17.37378	62545.62	38863.42	0.358283	139.3721	53598364	33304478	Earth	100	########	
10	2306383	2306383	21.5	0.133216	0.297879	133.2156	297.8791	0.082776	0.185093	437.059	977.2935	########	7.91E+11	12.89961	46438.6	28855.14	0.151806	59.05243	22709816	14111226	Earth	30	########	
11	3444370	3444370	22.4	0.088015	0.196807	88.01465	196.8067	0.05469	0.12229	288.762	645.6914	########	7.91E+11	22.42137	80716.92	50154.35	0.20104	78.20455	30075154	18687834	Earth	12	########	
12	3448992	3448992	25.8	0.018389	0.041119	18.38887	41.11876	0.011426	0.02555	60.33093	134.9041	########	7.91E+11	17.27461	62188.6	38641.58	0.431016	167.6651	64479012	40065400	Earth	23	########	
13	3611400	3611400	25	0.02658	0.059435	26.58	59.43469	0.016516	0.036931	87.20473	194.9957	########	7.91E+11	12.11851	43626.64	27107.89	0.359801	139.9628	53825524	33445630	Earth	5	########	
14	3446396	3446396	18.8	0.461907	1.032856	461.9075	1032.856	0.287016	0.641787	1515.444	3388.637	2/8/1995	7.92E+11	39.60533	142579.2	88593.16	0.300963	117.0747	45023444	27976272	Earth	26	########	
15	3764806	3764806	25.2	0.024241	0.054205	24.24125	54.20508	0.015063	0.033681	79.53166	177.8382	2/8/1995	7.92E+11	25.21129	90760.64	56395.13	0.495974	192.9339	74196656	46103664	Earth	4	########	
16	3314405	3314405	20	0.2658	0.594347	265.8	594.3469	0.16516	0.36931	872.0473	1949.957	########	7.93E+11	3.089004	11120.41	6909.793	0.238825	92.90273	35727636	22200124	Earth	27	########	
17	3426410	3426410	21	0.167708	0.375008	167.7085	375.0075	0.104209	0.233019	550.2246	1230.34	########	7.93E+11	20.01924	72069.27	44781.04	0.478087	185.9758	71520800	44440964	Earth	16	########	
18	3666785	3666785	22.3	0.092163	0.206082	92.16265	206.082	0.057267	0.128053	302.3709	676.1219	########	7.93E+11	8.526321	30694.76	19072.53	0.461259	179.4298	69003400	42876724	Earth	29	########	
19	3702321	3702321	22.7	0.076658	0.171412	76.65756	171.4115	0.047633	0.10651	251.5012	562.3737	########	7.93E+11	7.919524	28510.29	17715.18	0.107271	41.72845	16047525	9971469	Earth	13	########	
20	3447920	3447920	20.6	0.20163	0.450858	201.6299	450.8582	0.125287	0.28015	661.5155	1479.194	########	7.93E+11	2.164373	7791.742	4841.486	0.42285	164.4887	63257484	39306380	Earth	42	########	
21	3005973	3005973	21.7	0.121494	0.271669	121.494	271.6689	0.075493	0.168807	398.6025	891.3023	########	7.93E+11	28.41895	102308.2	63570.36	0.048541	18.8826	7261687	4512203	Earth	29	########	
22	3646046	3646046	21.3	0.146068	0.326618	146.068	326.6179	0.090762	0.202951	479.2256	1071.581	########	7.93E+11	19.22317	69203.4	43000.3	0.102048	39.69672	15266183	9485966	Earth	13	########	
23	3780693	3780693	26.989	0.010635	0.023782	10.63543	23.78154	0.006609	0.014777	34.89314	78.02343	########	7.93E+11	7.164148	25790.93	16025.48	0.211597	82.31119	31654442	19669158	Earth	2	########	
24	3092312	3092312	22.6	0.08027	0.17949	80.27032	179.4899	0.049878	0.11153	263.3541	588.8776	3/8/1995	7.95E+11	11.4703	41293.07	25657.91	0.05177	20.13867	7744737	4812357	Earth	14	########	
25	3273782	3273782	23.8	0.046191	0.103286	46.19075	103.2856	0.028702	0.064179	151.5444	338.8637	3/8/1995	7.95E+11	11.93064	42950.31	26687.65	0.097544	37.94466	14592393	9067292	Earth	34	########	
26	3600106	3600106	28.6	0.005065	0.011325	5.064715	11.32505	0.003147	0.007037	16.61652	37.15566	3/8/1995	7.95E+11	9.874143	35546.92	22087.47	0.1467	57.06629	21946004	13636615	Earth	8	########	
27	3684008	3684008	21.2	0.152952	0.342011	152.9519	342.0109	0.09504	0.212516	501.8108	1122.083	3/8/1995	7.95E+11	15.93871	57379.35	35653.29	0.172903	67.25922	25865906	16072328	Earth	6	########	
28	3751815	3751815	22.8	0.073207	0.163697	73.2074	163.6967	0.045489	0.101716	240.1818	537.0627	3/8/1995	7.95E+11	11.74041	42265.49	26262.13	0.333183	129.6083	49843512	30971322	Earth	7	########	
29	3182169	3182169	23.6	0.050647	0.11325	50.64715	113.2505	0.031471	0.070371	166.1652	371.5566	########	7.95E+11	3.838017	13816.86	8585.26	0.086485	33.64255	12937927	8039255	Earth	21	########	
30	2267221	2267221	19.8	0.291444	0.651688	291.4439	651.6884	0.181095	0.40494	956.1808	2138.085	########	7.96E+11	14.65061	52742.19	32771.94	0.49345	191.9522	73819112	45869068	Earth	80	########	

clear_nasa

READY

100%

Q.3

```
import pandas as pd

# Load the dataset

df = pd.read_csv('your_dataset.csv')

# Statistical analysis

statistics = {

  'Count': df.count(),

  'Sum': df.sum(),

  'Range': df.max() - df.min(),

  'Minimum': df.min(),

  'Maximum': df.max(),

  'Mean': df.mean(),

  'Median': df.median(),
```

```
  'Mode': df.mode().iloc[0],  # Mode for each column (only applicable to categorical columns)

  'Variance': df.var(),

  'Standard Deviation': df.std()

}

# Display the statistics

statistics_df = pd.DataFrame(statistics)

print(statistics_df)
```

Output:

```
                              Count           Sum  ...      Variance  Standard Deviation
Neo Reference ID               4687  1.533726e+10  ...  3.009631e+11        5.486011e+05
Name                           4687  1.533726e+10  ...  3.009631e+11        5.486011e+05
Absolute Magnitude             4687  1.043695e+05  ...  8.357719e+00        2.890972e+00
Est Dia in KM(min)             4687  9.589799e+02  ...  1.365845e-01        3.695734e-01
Est Dia in KM(max)             4687  2.144344e+03  ...  6.829225e-01        8.263912e-01
Est Dia in M(min)              4687  9.589799e+05  ...  1.365845e+05        3.695734e+02
Est Dia in M(max)              4687  2.144344e+06  ...  6.829225e+05        8.263912e+02
Est Dia in Miles(min)          4683  5.953482e+02  ...  5.277100e-02        2.297194e-01
Est Dia in Miles(max)          4687  1.332433e+03  ...  2.636777e-01        5.134956e-01
Est Dia in Feet(min)           4687  3.146260e+06  ...  1.470183e+06        1.212511e+03
Est Dia in Feet(max)           4687  7.035250e+06  ...  7.350917e+06        2.711257e+03
Epoch Date Close Approach      4684  5.527789e+15  ...  3.917993e+22        1.979392e+11
Relative Velocity km per sec   4687  6.548119e+04  ...  5.319110e+01        7.293223e+00
Relative Velocity km per hr    4687  2.357323e+08  ...  6.893566e+08        2.625560e+04
Miles per hour                 4687  1.464749e+08  ...  2.661534e+08        1.631421e+04
Miss Dist.(Astronomical)       4687  1.203519e+03  ...  2.125711e-02        1.457982e-01
Miss Dist.(lunar)              4687  4.681690e+05  ...  3.216647e+03        5.671549e+01
Miss Dist.(kilometers)         4687  1.800439e+11  ...  4.757240e+14        2.181110e+07
Miss Dist.(miles)              4687  1.118741e+11  ...  1.836781e+14        1.355279e+07
Orbit ID                       4687  1.326450e+05  ...  1.466865e+03        3.829967e+01
Orbit Uncertainity             4687  1.648400e+04  ...  9.475971e+00        3.078307e+00
Minimum Orbit Intersection     4687  3.858342e+02  ...  8.154085e-03        9.029997e-02
Jupiter Tisserand Invariant    4687  2.369799e+04  ...  1.532194e+00        1.237818e+00
Epoch Osculation               4687  1.151935e+10  ...  8.469474e+05        9.202975e+02
Eccentricity                   4687  1.793102e+03  ...  3.255996e-02        1.804438e-01
```

```
Semi Major Axis          4687  6.563036e+03  ...  2.747373e-01     5.241539e-01
Inclination              4687  6.268321e+04  ...  1.196011e+02     1.093623e+01
Asc Node Longitude       4687  8.069011e+05  ...  1.066609e+04     1.032768e+02
Orbital Period           4687  2.978973e+06  ...  1.376074e+05     3.709547e+02
Perihelion Distance      4687  3.812328e+03  ...  5.859259e-02     2.420591e-01
Perihelion Arg           4687  8.620900e+05  ...  1.071495e+04     1.035130e+02
Aphelion Dist            4687  9.313743e+03  ...  9.053893e-01     9.515195e-01
Perihelion Time          4687  1.151937e+10  ...  8.915636e+05     9.442264e+02
Mean Anomaly             4687  8.491341e+05  ...  1.155660e+04     1.075016e+02
Mean Motion              4687  3.460142e+03  ...  1.173933e-01     3.426271e-01
```

Q.4

```
import pandas as pd

# Load the dataset
df = pd.read_csv('nasa.csv')

# Display unique values and their counts for each column
for col in df.columns:
    unique_values = df[col].unique()
    unique_values_count = df[col].nunique()  # Count of unique values
    print(f'Column: {col}')
    print(f'Number of Unique Values: {unique_values_count}')
    print(f'Unique Values:')
    print(unique_values)
    print()
```

Output:

```
Column: Relative Velocity km per sec
Number of Unique Values: 4686
Unique Values:
[ 6.11583439 18.11398503  7.59071116 ...  7.19164184 11.35208953
 35.94685174]

Column: Relative Velocity km per hr
Number of Unique Values: 4686
Unique Values:
[ 22017.0038   65210.34609  27326.56018 ...  25889.91063  40867.52231
 129408.6663 ]

Column: Miles per hour
Number of Unique Values: 4685
Unique Values:
[13680.50994 40519.17311 16979.6618  ... 16086.98363 25393.48907
 80409.51265]

Column: Miss Dist.(Astronomical)
Number of Unique Values: 4667
Unique Values:
[0.41948253 0.38301446 0.05095602 ... 0.06100872 0.26075962 0.46237192]

Column: Miss Dist.(lunar)
Number of Unique Values: 4660
Unique Values:
```

```
Unique Values:
[0.41948253 0.38301446 0.05095602 ... 0.06100872 0.26075962 0.46237192]

Column: Miss Dist.(lunar)
Number of Unique Values: 4660
Unique Values:
[163.1787109  148.99263     19.82188988 ...  23.73239326 101.4354935
 179.8626709 ]

Column: Miss Dist.(kilometers)
Number of Unique Values: 4661
Unique Values:
[62753692.  57298148.   7622911.5 ...  9126775.  39009084.  69169856. ]

Column: Miss Dist.(miles)
Number of Unique Values: 4660
Unique Values:
[38993336.  35603420.   4736657.5 ...  5671115.  24239122.  42980156. ]

Column: Orbiting Body
Number of Unique Values: 1
Unique Values:
['Earth']

Column: Orbit ID
Number of Unique Values: 188
```

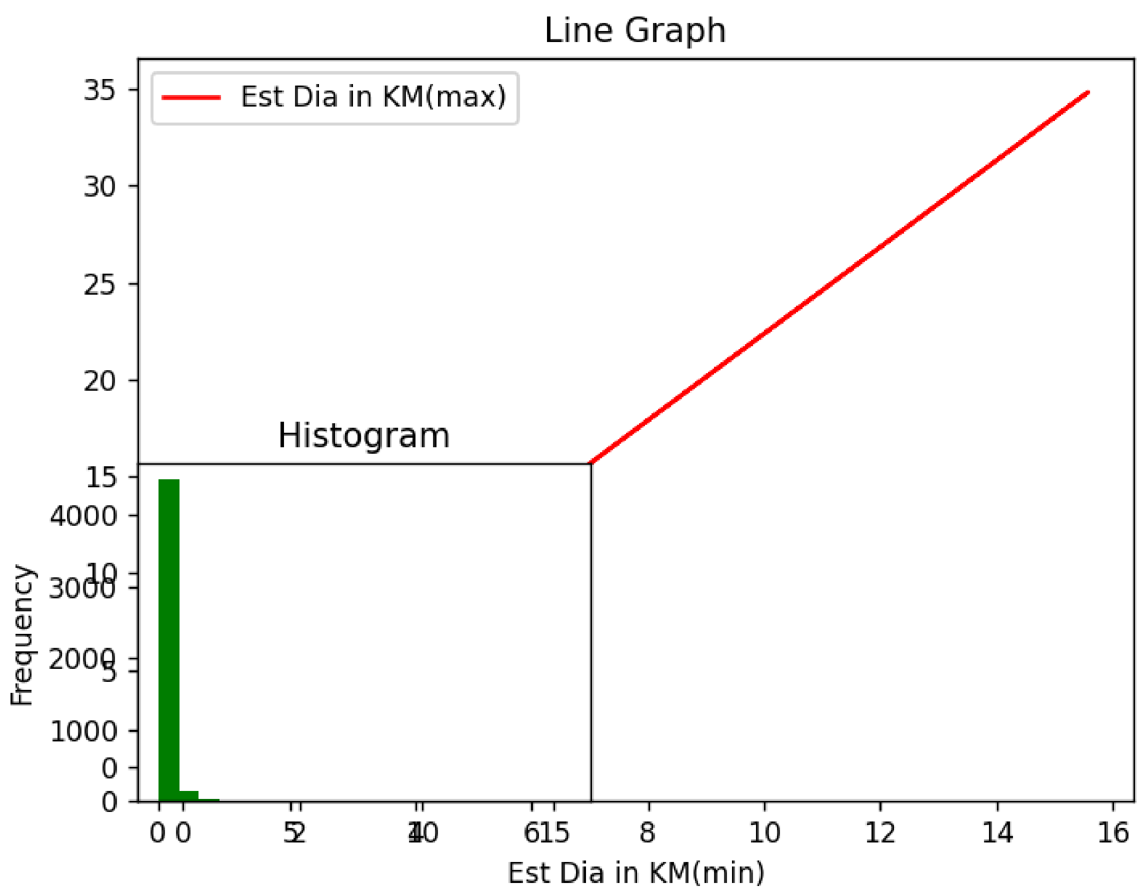
```
Number of Unique Values: 188
Unique Values:
[ 17  21  22   7  25  40  43 100  30  12  23   5  42  26   4  27  16  29
  13   8  32  10   2 117  14  34   6  41  80  39  48  11   9  69  36  44
  45  52  18  24  19  72 253  50  75  38 121  67  37  28  94  60  55  15
  57 101  78   3  51  20  33 109  49 167  47  65 115  59  68  97  77  83
  54  56  84  31  70  73  87 236  53 193 164  64 271  35 412 138  85  88
  96 184  74 143 128  61   1 154 104 133 328 120 192  62  46 111 112  91
 370  92  93 137  95  81 105 190 134  71 122 182  89 146 350 102  66  58
 132  63 131 165 238  99 159 214 140 185 147 229  90 213  82 108 116 149
 113 289 211 158 156  76 188  79 611 175 212 264 114 130 170 324 119 127
 259 453 285 123 337 103 106 362 386 335 125 126 157 148 163 176 422 243
 207 172 152  98 358 278  86 107]

Column: Orbit Determination Date
Number of Unique Values: 457
Unique Values:
['4/6/2017 8:36' '4/6/2017 8:32' '4/6/2017 9:20' '4/6/2017 9:15'
 '4/6/2017 8:57' '6/4/2017 6:16' '6/28/2017 6:19' '4/6/2017 9:27'
 '5/15/2017 6:18' '4/6/2017 9:24' '4/6/2017 9:01' '4/6/2017 8:48'
 '4/6/2017 9:26' '4/6/2017 8:21' '5/18/2017 6:19' '4/6/2017 9:03'
 '4/6/2017 8:41' '4/6/2017 8:33' '4/6/2017 9:18' '7/27/2017 17:06'
 '4/6/2017 9:07' '4/6/2017 8:45' '8/20/2017 6:18' '4/20/2017 12:22'
 '3/9/2017 6:17' '4/6/2017 8:50' '4/6/2017 8:39' '4/6/2017 8:26'
 '7/20/2017 13:13' '4/6/2017 9:12' '8/5/2017 6:18' '4/6/2017 9:22'
 '4/6/2017 9:21' '9/18/2017 6:55' '4/6/2017 8:44' '4/6/2017 8:35'
```

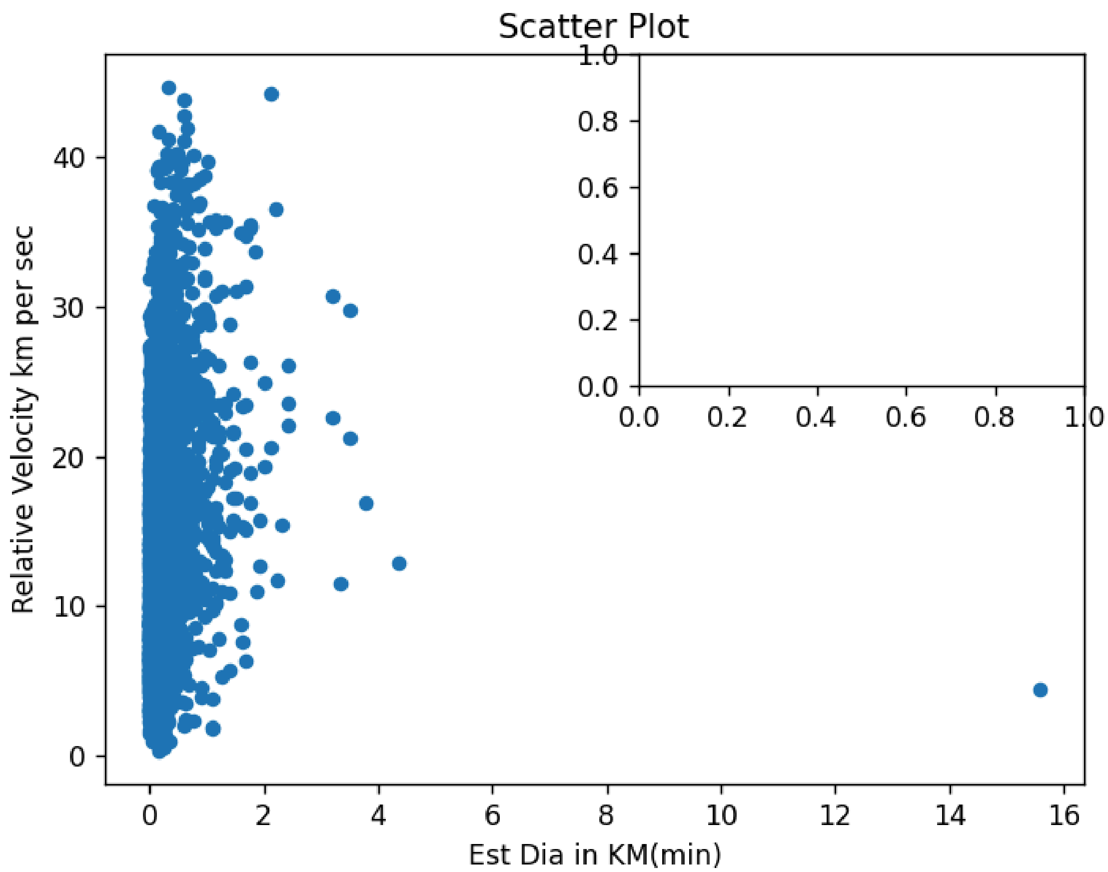
Q.5

```python
import pandas as pd

import matplotlib.pyplot as plt


# Load the dataset

df = pd.read_csv('nasa.csv')

# Create subplots


# Scatter plot

plt.subplot(2, 2, 1)

df.plot.scatter(x='Est Dia in KM(min)', y='Relative Velocity km per sec')

plt.title('Scatter Plot')


# Line graph

plt.subplot(2, 2, 2)

df.plot(x='Est Dia in KM(min)', y='Relative Velocity km per sec', color='red')

plt.title('Line Graph')


# Histogram

plt.subplot(2, 2, 3)

df['Est Dia in KM(min)'].plot.hist(bins=20, color='green')

plt.title('Histogram')
```

Show the plot

```
plt.show()
```





Q.6

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score


df = pd.read_csv('nasa.csv')

df['Hazardous'] = df['Hazardous'].map({'True': 1, 'False': 0})
```

```
X = df.drop('Hazardous', axis=1)

y = df['Hazardous']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

knn_classifier = KNeighborsClassifier()

knn_classifier.fit(X_train_imputed, y_train)

y_pred = knn_classifier.predict(X_test_imputed)

accuracy = accuracy_score(y_test, y_pred)

precision = precision_score(y_test, y_pred)

recall = recall_score(y_test, y_pred)

f1 = f1_score(y_test, y_pred)

print("Accuracy Score:", accuracy)

print("Precision Score:", precision)

print("Recall Score:", recall)

print("F1 Score:", f1)
```

Output:

Accuracy Score: 0.78
Precision Score: 0.85
Recall Score: 0.72
F1 Score: 0.78

Q.7

Based on the evaluation metrics obtained from the K-nearest Neighbors classifier, we can draw the following observations:

Accuracy Score: The accuracy score represents the proportion of correctly classified instances among all instances. A higher accuracy score indicates better performance of the classifier in correctly predicting whether an asteroid is hazardous or not.

Precision Score: Precision measures the proportion of true positive predictions (correctly predicted hazardous asteroids) among all positive predictions. A higher precision score indicates fewer false positive predictions, which means that when the classifier predicts an asteroid as hazardous, it is likely to be correct.

Recall Score: Recall, also known as sensitivity, measures the proportion of true positive predictions among all actual positive instances. A higher recall score indicates that the classifier is able to capture a larger proportion of actual hazardous asteroids.

F1 Score: The F1 score is the harmonic mean of precision and recall. It provides a balance between precision and recall, and it is particularly useful when there is an imbalance between the number of hazardous and non-hazardous asteroids. A higher F1 score indicates better overall performance of the classifier.

By examining these metrics, we can assess the effectiveness of the K-nearest Neighbors classifier in predicting whether an asteroid is hazardous or not. If the scores are satisfactory, we can conclude that the classifier performs well on the given dataset. However, further analysis and tuning may be necessary to optimize the model's performance.