nwyxocjaq

July 31, 2023

[53]:

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

[54]:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[55]:

```
df=pd.read_csv("/content/drive/MyDrive/mydatasets/19_nuclear_explosions.csv")
df
```

[55]:

```
     WEAPON SOURCE COUNTRY WEAPON DEPLOYMENT LOCATION Data.Source  \
0                      USA                 Alamogordo         DOE
1                      USA                  Hiroshima         DOE
2                      USA                   Nagasaki         DOE
3                      USA                     Bikini         DOE
4                      USA                     Bikini         DOE
…                    …                        …         …
2041                 CHINA                    Lop Nor         HFS
2042                 INDIA                    Pokhran         HFS
2043                 INDIA                    Pokhran         NRD
2044                PAKIST                     Chagai         HFS
2045                PAKIST                     Kharan         HFS

      Location.Cordinates.Latitude  Location.Cordinates.Longitude  \
0                            32.54                        -105.57
1                            34.23                         132.27
2                            32.45                         129.52
3                            11.35                         165.20
4                            11.35                         165.20
…                             …                            …
2041                         41.69                          88.35
2042                         27.07                          71.70
2043                         27.07                          71.70
2044                         28.90                          64.89
```

```
2045                         28.49                          63.78

      Data.Magnitude.Body  Data.Magnitude.Surface  Location.Cordinates.Depth  \
0                     0.0                     0.0                      -0.10
1                     0.0                     0.0                      -0.60
2                     0.0                     0.0                      -0.60
3                     0.0                     0.0                      -0.20
4                     0.0                     0.0                       0.03
…                   …                     …                        …
2041                  5.3                     0.0                       0.00
2042                  5.3                     0.0                       0.00
2043                  0.0                     0.0                       0.00
2044                  0.0                     0.0                       0.00
2045                  5.0                     0.0                       0.00

      Data.Yeild.Lower  Data.Yeild.Upper Data.Purpose   Data.Name Data.Type  \
0                 21.0              21.0           Wr     Trinity     Tower
1                 15.0              15.0       Combat   Littleboy   Airdrop
2                 21.0              21.0       Combat      Fatman   Airdrop
3                 21.0              21.0           We        Able   Airdrop
4                 21.0              21.0           We       Baker        Uw
…                …               …          …         …       …
2041               3.0              12.0           Wr         Nan        Ug
2042               0.0              20.0           Wr  Shakti 1-3        Ug
2043               0.0               1.0           Wr         Nan        Ug
2044               0.0              35.0           Wr         Nan        Ug
2045               0.0              18.0           Wr         Nan        Ug

      Date.Day  Date.Month  Date.Year
0           16           7       1945
1            5           8       1945
2            9           8       1945
3           30           6       1946
4           24           7       1946
…        …         …        …
2041        29           7       1996
2042        11           5       1998
2043        13           5       1998
2044        28           5       1998
2045        30           5       1998

[2046 rows x 16 columns]
```

[56]:

```
df.head()
```

[56]:

```
  WEAPON SOURCE COUNTRY WEAPON DEPLOYMENT LOCATION Data.Source  \
0                   USA                 Alamogordo         DOE
```

```
1                   USA                  Hiroshima         DOE
2                   USA                   Nagasaki         DOE
3                   USA                     Bikini         DOE
4                   USA                     Bikini         DOE

   Location.Cordinates.Latitude  Location.Cordinates.Longitude  \
0                         32.54                        -105.57   
1                         34.23                         132.27   
2                         32.45                         129.52   
3                         11.35                         165.20   
4                         11.35                         165.20   

   Data.Magnitude.Body  Data.Magnitude.Surface  Location.Cordinates.Depth  \
0                  0.0                     0.0                      -0.10   
1                  0.0                     0.0                      -0.60   
2                  0.0                     0.0                      -0.60   
3                  0.0                     0.0                      -0.20   
4                  0.0                     0.0                       0.03   

   Data.Yeild.Lower  Data.Yeild.Upper Data.Purpose  Data.Name Data.Type  \
0              21.0              21.0           Wr    Trinity     Tower   
1              15.0              15.0       Combat  Littleboy   Airdrop   
2              21.0              21.0       Combat     Fatman   Airdrop   
3              21.0              21.0           We       Able   Airdrop   
4              21.0              21.0           We      Baker        Uw   

   Date.Day  Date.Month  Date.Year  
0        16           7       1945  
1         5           8       1945  
2         9           8       1945  
3        30           6       1946  
4        24           7       1946  
```

1 Data Cleaning and Data Preprocessing

[57]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2046 entries, 0 to 2045
Data columns (total 16 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   WEAPON SOURCE COUNTRY          2046 non-null   object 
 1   WEAPON DEPLOYMENT LOCATION     2046 non-null   object 
 2   Data.Source                    2046 non-null   object 
 3   Location.Cordinates.Latitude   2046 non-null   float64
 4   Location.Cordinates.Longitude  2046 non-null   float64
 5   Data.Magnitude.Body            2046 non-null   float64
 6   Data.Magnitude.Surface         2046 non-null   float64
 7   Location.Cordinates.Depth      2046 non-null   float64
 8   Data.Yeild.Lower               2046 non-null   float64
 9   Data.Yeild.Upper               2046 non-null   float64
 10  Data.Purpose                   2046 non-null   object
 11  Data.Name                      2046 non-null   object
 12  Data.Type                      2046 non-null   object
 13  Date.Day                       2046 non-null   int64
 14  Date.Month                     2046 non-null   int64
 15  Date.Year                      2046 non-null   int64
dtypes: float64(7), int64(3), object(6)
memory usage: 255.9+ KB
```

```
[58]: df.describe()
```

```
[58]:        Location.Cordinates.Latitude  Location.Cordinates.Longitude  \
      count                   2046.000000                    2046.000000
      mean                      35.462429                     -36.015037
      std                       23.352702                     100.829355
      min                      -49.500000                    -169.320000
      25%                       37.000000                    -116.051500
      50%                       37.100000                    -116.000000
      75%                       49.870000                      78.000000
      max                       75.100000                     179.220000

             Data.Magnitude.Body  Data.Magnitude.Surface  Location.Cordinates.Depth  \
      count          2046.000000             2046.000000                2046.000000
      mean              2.145406                0.356696                  -0.490829
      std               2.625453                1.203569                  10.981072
      min               0.000000                0.000000                -400.000000
      25%               0.000000                0.000000                   0.000000
      50%               0.000000                0.000000                   0.000000
      75%               5.100000                0.000000                   0.000000
      max               7.400000                6.000000                   1.451000

             Data.Yeild.Lower  Data.Yeild.Upper     Date.Day   Date.Month  \
      count       2046.000000       2046.000000  2046.000000  2046.000000
      mean         208.444528        323.431021    16.683773     7.282502
      std         1641.962943       2055.203066     8.799878     3.132347
      min            0.000000          0.000000     1.000000     1.000000
      25%            0.000000         18.250000     9.000000     5.000000
      50%            0.001000         20.000000    17.000000     8.000000
      75%           20.000000        150.000000    25.000000    10.000000
      max        50000.000000      50000.000000    31.000000    12.000000
```

```
         Date.Year
count  2046.000000
mean   1970.896383
std      10.372760
min    1945.000000
25%    1962.000000
50%    1970.000000
75%    1979.000000
max    1998.000000
```

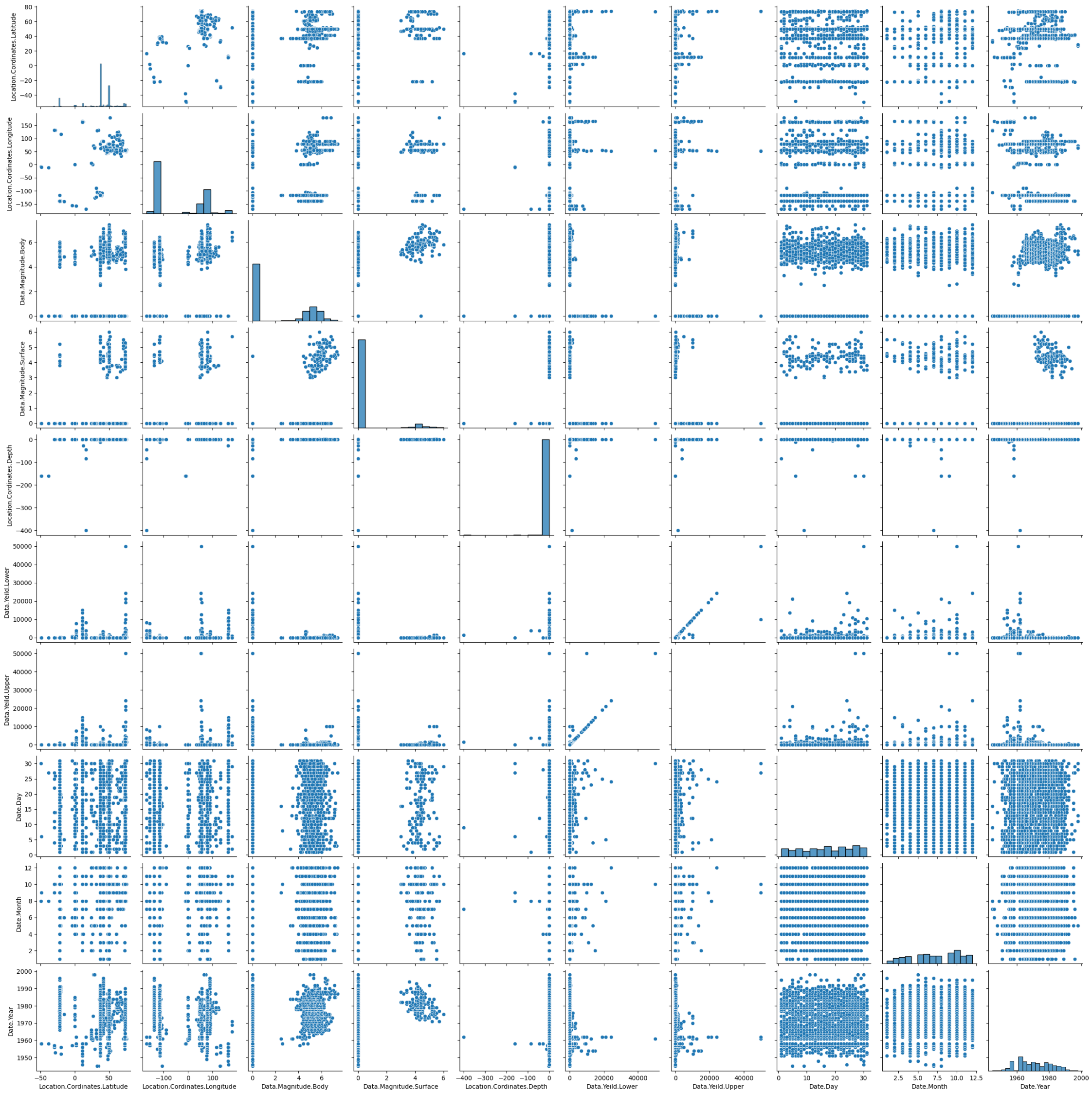
[59]:
```
df.columns
```

[59]:
```
Index(['WEAPON SOURCE COUNTRY', 'WEAPON DEPLOYMENT LOCATION', 'Data.Source',
       'Location.Cordinates.Latitude', 'Location.Cordinates.Longitude',
       'Data.Magnitude.Body', 'Data.Magnitude.Surface',
       'Location.Cordinates.Depth', 'Data.Yeild.Lower', 'Data.Yeild.Upper',
       'Data.Purpose', 'Data.Name', 'Data.Type', 'Date.Day', 'Date.Month',
       'Date.Year'],
      dtype='object')
```

2 EDA and Visualization

[60]:
```
sns.pairplot(df)
```

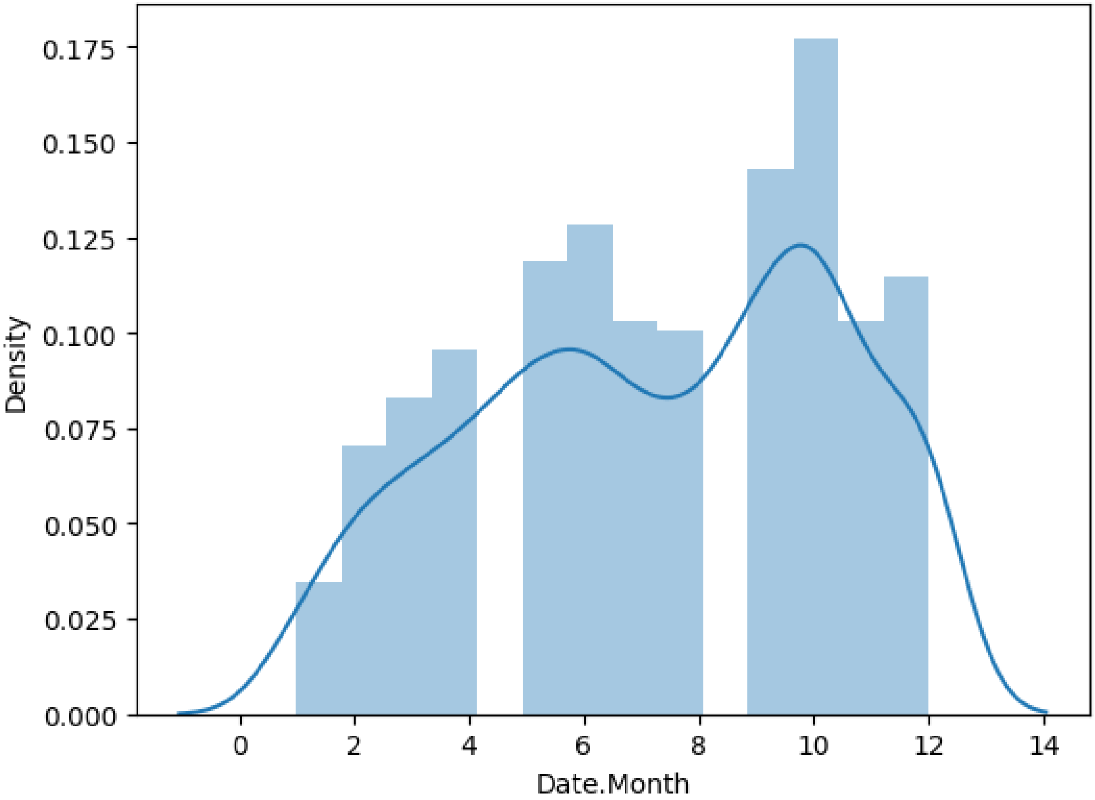
[60]:
```
<seaborn.axisgrid.PairGrid at 0x7da730e49540>
```



[61]:
```
sns.distplot(df['Date.Month'])
```

```
<ipython-input-61-8444b77a8752>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['Date.Month'])
```

```
[61]: <Axes: xlabel='Date.Month', ylabel='Density'>
```

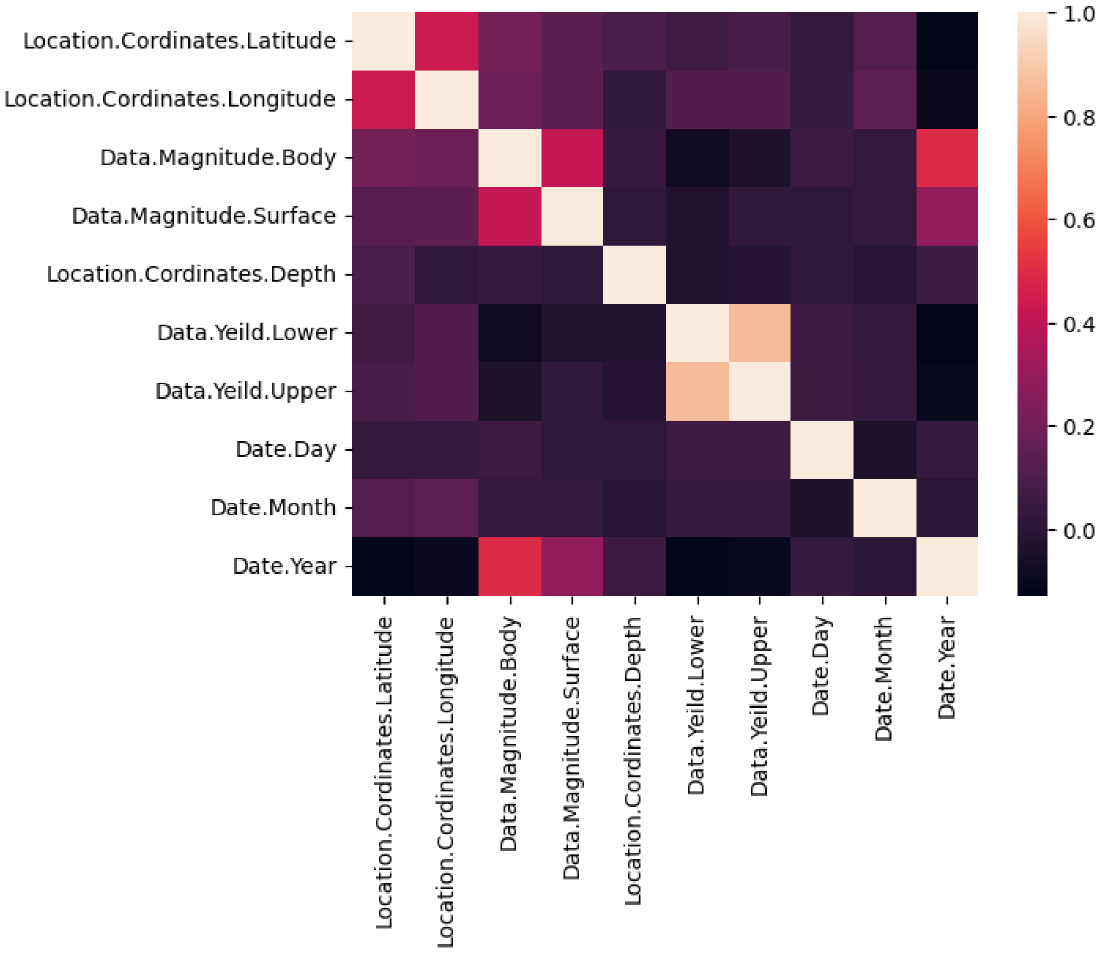


```
[62]: df1=df[[       'Location.Cordinates.Latitude', 'Location.Cordinates.Longitude',
             'Data.Magnitude.Body', 'Data.Magnitude.Surface',
             'Location.Cordinates.Depth', 'Data.Yeild.Lower', 'Data.Yeild.Upper',␣
        ↪'Date.Day', 'Date.Month',
             'Date.Year']]
      df1
```

```
[62]:       Location.Cordinates.Latitude  Location.Cordinates.Longitude  \
      0                            32.54                        -105.57
      1                            34.23                         132.27
      2                            32.45                         129.52
      3                            11.35                         165.20
      4                            11.35                         165.20
      …                              …                            …
      2041                         41.69                          88.35
      2042                         27.07                          71.70
      2043                         27.07                          71.70
      2044                         28.90                          64.89
      2045                         28.49                          63.78
      Data.Magnitude.Body  Data.Magnitude.Surface  Location.Cordinates.Depth  \
0                     0.0                     0.0                      -0.10   
1                     0.0                     0.0                      -0.60   
2                     0.0                     0.0                      -0.60   
3                     0.0                     0.0                      -0.20   
4                     0.0                     0.0                       0.03   
…                   …                      …                         …   
2041                  5.3                     0.0                       0.00   
2042                  5.3                     0.0                       0.00   
2043                  0.0                     0.0                       0.00   
2044                  0.0                     0.0                       0.00   
2045                  5.0                     0.0                       0.00   

      Data.Yeild.Lower  Data.Yeild.Upper  Date.Day  Date.Month  Date.Year  
0                 21.0              21.0        16           7       1945  
1                 15.0              15.0         5           8       1945  
2                 21.0              21.0         9           8       1945  
3                 21.0              21.0        30           6       1946  
4                 21.0              21.0        24           7       1946  
…                  …                 …         …         …         …  
2041               3.0              12.0        29           7       1996  
2042               0.0              20.0        11           5       1998  
2043               0.0               1.0        13           5       1998  
2044               0.0              35.0        28           5       1998  
2045               0.0              18.0        30           5       1998  

[2046 rows x 10 columns]
```

[63]:
```python
sns.heatmap(df1.corr())
```

[63]:
```
<Axes: >
```



```
[64]: x=df1[[       'Location.Cordinates.Latitude', 'Location.Cordinates.Longitude',
             'Data.Magnitude.Body', 'Data.Magnitude.Surface',
             'Location.Cordinates.Depth', 'Data.Yeild.Lower', 'Data.Yeild.Upper',␣
       ↪'Date.Day',
             'Date.Year']]
      y=df1['Date.Month']
```

```
[65]: from sklearn.model_selection import train_test_split
      x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

```
[66]: from sklearn.linear_model import LinearRegression
      lr=LinearRegression()
      lr.fit(x_train,y_train)
```

```
[66]: LinearRegression()
```

```
[67]: print(lr.intercept_)
```

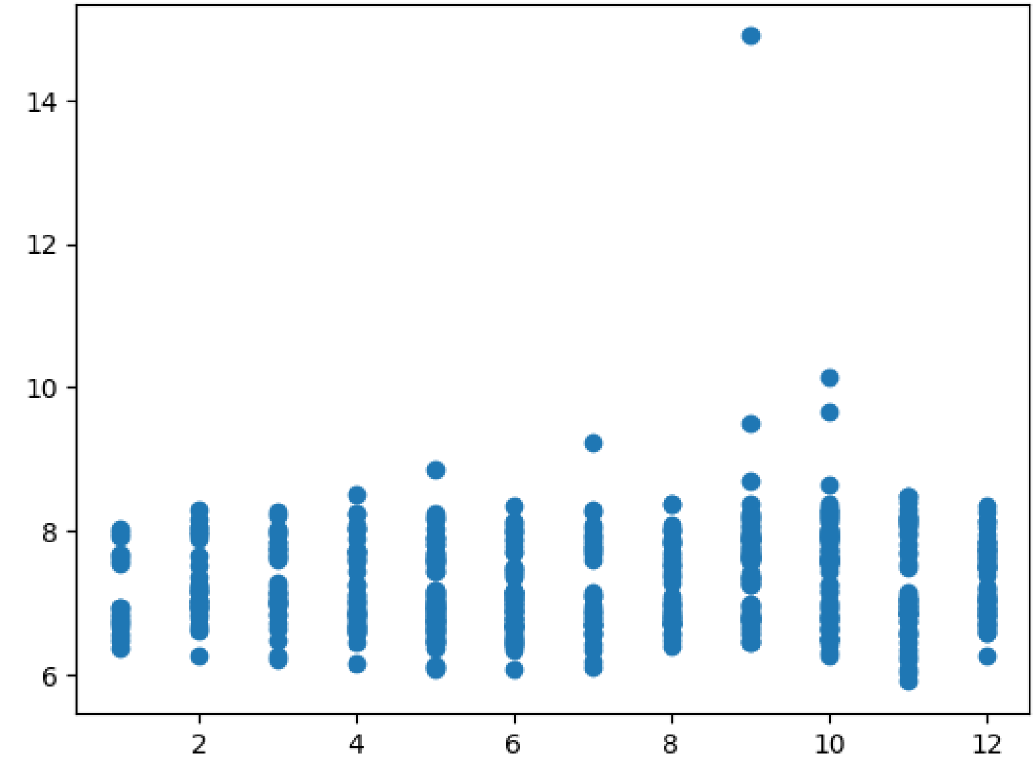
```
-21.295740333211487
```

```
[68]: coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
      coeff
```

```
[68]:                                Co-efficient
      Location.Cordinates.Latitude       0.007195
      Location.Cordinates.Longitude      0.004895
      Data.Magnitude.Body               -0.031879
      Data.Magnitude.Surface            -0.022461
      Location.Cordinates.Depth         -0.006506
      Data.Yeild.Lower                  -0.000116
      Data.Yeild.Upper                   0.000168
      Date.Day                          -0.023432
      Date.Year                          0.014668
```

```
[69]: prediction =lr.predict(x_test)
      plt.scatter(y_test,prediction)
```

```
[69]: <matplotlib.collections.PathCollection at 0x7da721d32f50>
```



```
[70]: lr.score(x_test,y_test)
```

[70]:
```
0.0029374704424917075
```

[71]:
```
lr.score(x_train,y_train)
```

[71]:
```
0.03812256911806777
```

[72]:
```
from sklearn.linear_model import Ridge,Lasso
```

[73]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

[73]:
```
Ridge(alpha=10)
```

[74]:
```
rr.score(x_test,y_test)
```

[74]:
```
0.0029444123882234052
```

[75]:
```
rr.score(x_train,y_train)
```

[75]:
```
0.03812256610260967
```

[76]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

[76]:
```
Lasso(alpha=10)
```

[77]:
```
la.score(x_test,y_test)
```

[77]:
```
0.006521986666867319
```

[78]:
```
la.score(x_train,y_train)
```

[78]:
```
0.029650155894626384
```

[79]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

[79]:
```
ElasticNet()
```

[80]:
```
en.coef_
```

[80]:
```
array([ 0.00481387,  0.00476747, -0.        , -0.        , -0.        ,
       -0.00010227,  0.00015079, -0.01661944,  0.00421704])
```

[81]:
```
en.intercept_
```

```
[81]: -0.804629708509343
```

```
[82]: prediction = en.predict(x_test)
      prediction
```

```
[82]: array([ 6.5905297 ,  7.62938273,  6.45728632,  7.85309353,  7.07646191,
              6.86934062,  7.88587415,  6.32708996,  7.02901698,  6.85222377,
              7.66596404,  8.01545347,  6.8564408 ,  6.61207073,  7.01074436,
              8.065279  ,  7.85619876,  7.04814398,  7.877247  ,  7.85674715,
              6.73809358,  6.65725618,  6.66359606,  8.16833717,  8.14823015,
              7.73617514,  6.91156431,  6.98939636,  8.01397304,  7.70202952,
              7.69000749,  6.72804519,  7.87832697,  6.77962634,  7.9230658 ,
              7.61470722,  7.71902556,  7.89627456,  6.69025101,  8.13784547,
              7.023513  ,  6.86537228,  6.57889429,  7.03312602,  6.8451187 ,
              7.79405258,  6.60843531,  8.22967083,  7.0399654 ,  7.04632964,
              7.89596563,  6.64292962,  6.68216612,  6.99958404,  7.67499291,
              6.82369932,  6.76282   ,  6.7981484 ,  6.77920462,  6.86142969,
              6.89836944,  7.85209628,  6.9477234 ,  6.78799276,  7.09492176,
              6.82761244,  6.31497072,  7.14090818,  7.62374075,  7.66988756,
              7.71749802,  7.77842225,  6.51985529,  6.66141892,  7.6322076 ,
              7.87495427,  7.97406075,  6.8447479 ,  6.93632044,  7.7190724 ,
              6.70786526,  7.64151978,  7.93256605,  7.61647153,  6.45306928,
              7.69096159,  7.01556219,  6.9854808 ,  8.00580147,  6.69800464,
              6.86137155,  6.88099566,  7.17081581,  8.01568268,  7.74286725,
              6.77909481,  6.8787695 ,  7.13973688,  6.97348083,  7.76423842,
              7.05662024,  6.90958075,  6.35611181,  6.88342221,  7.91583914,
              6.80757252,  6.70264877,  6.69688238,  6.56443443,  6.65800229,
              7.49052566,  7.12977207,  6.81683033,  8.14441916,  6.72749954,
              7.6863401 ,  6.68991779,  7.96279586,  6.32634385,  7.61595313,
              7.03656385,  7.65739254,  7.8821612 ,  7.96416737,  6.98308888,
              7.09041997,  7.75401094,  7.19245373,  6.45177498,  7.86593786,
              7.64027202,  6.43938287,  7.70446235,  6.77284618,  7.71704068,
              7.6412094 ,  6.79942559,  6.99885714,  6.81468282,  7.89823498,
              6.85596406,  6.94101422,  8.28929687,  8.12651482,  6.45327349,
              7.04558282,  7.17278115,  8.07261479,  6.7743341 ,  9.36423201,
              6.98216877,  6.94784544,  7.74053799,  6.95662634,  7.84340422,
              7.9835639 ,  6.83684784,  8.05116887,  6.7501197 ,  6.86263248,
              6.81174603,  6.80286917,  6.80732858,  6.67510347,  7.68694783,
              6.86167922,  7.98375414,  7.784069  ,  8.24387731,  6.99938481,
              6.92416209,  8.86623094,  7.66877764,  7.89351437,  8.22489412,
              6.44640875,  7.82041966,  7.18606654,  7.70810092,  6.70933717,
              7.02759839,  6.61955155,  6.86439813,  6.74481914,  7.5718901 ,
              8.05118662,  6.8207988 ,  6.91445087,  6.99758173,  7.66567683,
              6.48046838,  6.4017531 ,  6.91151562,  7.72027051,  6.92768025,
              6.97738244,  7.71962978,  8.05635078,  7.86367686,  8.16615224,
              7.14800897,  7.78550585,  8.07058755,  8.04678397,  6.73659576,
              8.00961283,  7.9030569 ,  8.20837353,  6.81150522,  7.98115442,
              7.03950339,  6.98567673,  6.84754857,  6.81577488,  7.01578717,
              8.03727153,  6.92340461,  8.11830506,  7.98201051,  8.02772118,
              7.79153406,  7.85701326,  7.67020601,  6.96930598,  7.36555343,
              8.13810654,  6.95218913,  6.75459703,  6.85709634,  6.41781094,
              6.59871507,  7.59782248,  7.68769486,  7.86581757,  6.93183404,
              6.91061292,  6.99834166,  6.39402295,  6.91516971,  8.04432128,
              6.67785998,  8.14008738,  7.59590551,  6.99432105,  6.83982136,
              7.8504308 ,  8.03192403,  6.38857893,  7.60727036,  6.92394461,
              7.93534892,  7.74607927,  6.97349539,  6.81483293,  6.96382652,
              6.82950278,  7.13617021,  7.17060419,  7.76574251,  6.31396578,
              7.80720109,  6.87802432,  7.64497603,  6.61111748,  7.12646713,
              8.22245487,  6.84428048,  6.79986555,  6.90679654,  6.32406343,
              7.66576391,  7.97787256,  6.3971981 ,  6.9463429 ,  7.00331229,
              7.00430747,  7.87548266,  8.09352993,  6.77958058,  8.00181805,
              6.74502565,  7.61501341,  6.64554674,  7.83518164,  7.86127646,
              6.96180976,  6.89204561,  7.01494728,  6.7994747 ,  8.05749427,
              7.81557783,  7.8175731 ,  7.67836563,  6.85693821,  7.7955919 ,
              7.95683828,  6.70289747,  6.8003236 ,  6.86539885,  6.81526525,
              6.77575716,  6.8414646 ,  7.84870512,  7.7780396 ,  7.06112427,
              7.96919463,  7.7458571 ,  8.07735795,  6.80658247,  7.42976943,
              7.64139095,  6.85486897,  7.65072036,  6.79073868,  7.90165413,
              8.06104741,  7.04785   ,  7.89894385,  8.10941047,  8.09690336,
              6.92743896,  6.79183941,  6.87354164,  6.80786864,  6.86512358,
              7.75446537,  7.99079754,  6.98307329,  6.30915038,  7.77987186,
              6.82394802,  7.74885815,  7.89474444,  7.88039227,  7.75035897,
              6.71439858,  7.01127627,  7.63713063,  6.7525071 ,  6.70883998,
              7.71477505,  7.08950105,  8.33826522,  7.92983828,  7.11529406,
              6.38654148,  6.90644916,  6.67834136,  8.02834858,  6.80683595,
              6.99758173,  8.08891089,  7.69498746,  6.77849744,  6.70711451,
              7.72559059,  6.70373991,  8.06467812,  6.97349054,  7.92014301,
              7.08678475,  7.05359392,  6.92533216,  7.9476093 ,  7.40075118,
              7.0524032 ,  7.73645689,  6.71579729,  8.17820896,  6.92763302,
              6.77487235,  6.74829007,  7.85673092,  6.8753357 ,  6.83718468,
              6.65180141,  6.48404121,  6.6485734 ,  8.61662829,  7.05657805,
              6.69893285,  7.11565875,  7.10259299,  6.40464119,  7.69904107,
              7.80292935,  6.85650219,  6.81737162,  8.02577803,  7.03641476,
             10.00273699,  6.70304051,  7.11201856,  7.96750535,  7.30397072,
              6.76544565,  6.66966323,  6.94505218,  8.03094183,  6.68778109,
              6.82812293,  6.72373396,  6.73218406,  6.74432173,  6.91870154,
              7.1960238 ,  6.96062321,  7.69260067,  6.6419875 ,  7.89326432,
              8.04933018,  6.99931317,  7.8921    ,  6.47143638,  7.10962312,
              7.7159186 ,  7.75018267,  9.42468788,  6.56482358,  6.62798563,
              6.82066266,  6.83686873,  6.90801645,  7.66207401,  6.98364443,
              6.64530902,  6.91523062,  7.06888798,  6.81885033,  7.03126698,
              8.01878812,  6.87354164,  6.77855543,  6.69242726,  7.04073553,
              6.64541573,  7.48900618,  6.69187217,  6.68093039,  6.67412433,
              7.86449869,  6.69148988,  7.96186605,  6.77453271,  7.92863404,
              7.9934859 ,  7.09529562,  6.87775117,  7.04869782,  6.27323927,
              6.72809028,  7.64075795,  6.63360314,  7.83391073,  7.70899909,
              8.14033608,  6.67656402,  6.68228259,  7.07249358,  7.01866691,
              7.98749692,  7.80555919,  7.6241385 ,  6.71746132,  6.70389229,
              8.59744235,  6.45086533,  7.61482451,  7.60942494,  7.99400859,
              7.77104171,  6.74647619,  7.11514918,  6.90909521,  6.82389091,
              7.75213284,  7.99507326,  6.76148581,  7.66200256,  8.19754683,
              8.04432607,  6.53404396,  8.14481082,  7.03950339,  6.58912176,
              8.3793023 ,  6.7681039 ,  8.00677805,  6.97951908,  6.57883122,
              6.79942559,  6.91468925,  6.89883678,  6.5471161 ,  7.72143664,
              7.79121885,  6.68557704,  6.72741446,  6.91473321,  6.76565562,
              7.87390382,  7.95962644,  7.60658638,  7.91600483,  8.18001085,
              6.96969438,  7.67913631,  6.79907443,  6.64086942,  6.71054632,
              6.55045442,  8.11338995,  6.99014246,  6.78764683,  8.14886167,
              7.61832356,  6.74997896,  6.75970024,  7.79424219,  6.38811634,
              7.07309334, 14.14458449,  6.91989833,  7.64869462,  7.01068553,
              6.96630219,  7.77347625,  7.63183321,  7.95671728,  7.27498932,
              6.75299418,  6.91853772,  6.89055387,  7.94937106,  7.73257866,
              7.70681617,  7.73734277,  7.86178515,  7.74134848,  6.6086328 ,
              6.6367235 ,  7.86634115,  7.75345505,  6.86090654,  6.45032318,
              7.11495424,  8.0274709 ,  7.84488304,  7.61974073,  6.65862718,
              6.69570692,  7.014991  ,  6.96855988,  6.72918313,  7.87295342,
              7.97819045,  6.81898488,  6.67481604,  7.49415606,  7.90751097,
              6.70309387,  6.7313828 ,  6.58293901,  6.841059  ,  6.7775215 ,
              7.61243129,  7.73651093,  7.63630833,  7.94766808,  7.71764808,
              7.07231104,  7.8737436 ,  6.75989956,  6.9477328 ,  7.74301116,
              7.06288666,  8.05535614,  6.62447251,  7.75290787,  7.06031059,
              7.81188424,  6.87330895,  7.78782676,  6.71978015,  7.76815925,
              7.76990687,  8.11018197,  6.65950937,  7.74848186,  6.71861562,
              7.65048094,  6.60843531,  7.92908719,  7.1114257 ,  7.01220669,
              7.70722644,  7.91750265,  7.72852101,  6.90248788,  6.43139988,
              6.71604599,  7.53174787,  8.0384372 ,  6.83528503,  6.94003543,
              7.78570385,  7.32495315,  6.8653615 ,  6.79442877,  7.98234436,
              7.36649081,  7.87984896,  6.65328785,  7.8000957 ])
```

[83]:
```
en.score(x_test,y_test)
```

[83]:
```
0.004671013479204511
```

[84]:
```
from sklearn import metrics
```

[85]:
```
print("Mean Absolute Error: ", metrics.mean_absolute_error(y_test,prediction))
```

```
Mean Absolute Error:  2.701342563894735
```

[86]:
```
print("Mean Squared Error: ", metrics.mean_squared_error(y_test,prediction))
```

```
Mean Squared Error:  9.940471898360673
```

```
[87]: print("Root Mean Squared Error: ", np.sqrt(metrics.
       ↪mean_squared_error(y_test,prediction)))
```

```
Root Mean Squared Error:  3.152851391734262
```

```
[88]: import pickle
      filename='prediction'
      pickle.dump(lr,open(filename,'wb'))
```

```
[91]: model = pickle.load(open(filename, 'rb'))
      real=[[10,20,30,40,50,54,22,1,2002],[11,45,10,33,52,23,66,2,2003]]
      result = model.predict(real)
      result
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does
not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

```
[91]: array([6.03318637, 6.94680641])
```