DEENA 20104016

importing libraries

LINEAR REGRESSION

In [1]:
```python
import pandas as pd
import numpy as np
```

In [2]:
```python
data = pd.read_csv("19_nuclear_explosions.csv")
```

Out[2]:

	WEAPON SOURCE COUNTRY	WEAPON DEPLOYMENT LOCATION	Data.Source	Location.Cordinates.Latitude	Location.Cordinates.Lo
0	USA	Alamogordo	DOE	32.54	
1	USA	Hiroshima	DOE	34.23	
2	USA	Nagasaki	DOE	32.45	
3	USA	Bikini	DOE	11.35	
4	USA	Bikini	DOE	11.35	
...	...	...	...	...	
2041	CHINA	Lop Nor	HFS	41.69	
2042	INDIA	Pokhran	HFS	27.07	
2043	INDIA	Pokhran	NRD	27.07	
2044	PAKIST	Chagai	HFS	28.90	
2045	PAKIST	Kharan	HFS	28.49	

2046 rows × 16 columns

In [3]:

Out[3]:

	WEAPON SOURCE COUNTRY	WEAPON DEPLOYMENT LOCATION	Data.Source	Location.Cordinates.Latitude	Location.Cordinates.Long
0	USA	Alamogordo	DOE	32.54	-1(
1	USA	Hiroshima	DOE	34.23	1:
2	USA	Nagasaki	DOE	32.45	1:
3	USA	Bikini	DOE	11.35	1(
4	USA	Bikini	DOE	11.35	1(

In [4]:

Out[4]:

```
<bound method DataFrame.info of      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]>
```

In [5]:

Out[5]:

	Location.Cordinates.Latitude	Location.Cordinates.Longitude	Data.Magnitude.Body	Data.Ma
count	2046.000000	2046.000000	2046.000000	
mean	35.462429	-36.015037	2.145406	
std	23.352702	100.829355	2.625453	
min	-49.500000	-169.320000	0.000000	
25%	37.000000	-116.051500	0.000000	
50%	37.100000	-116.000000	0.000000	
75%	49.870000	78.000000	5.100000	
max	75.100000	179.220000	7.400000	

Train the model

In [6]:

```
x = data[['Date.Day']]
```

In [7]:

```
# to split my dataset into training and test data
from sklearn.model_selection import train_test_split
```

In [8]:

```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
```

Out[8]: LinearRegression()

In [9]:

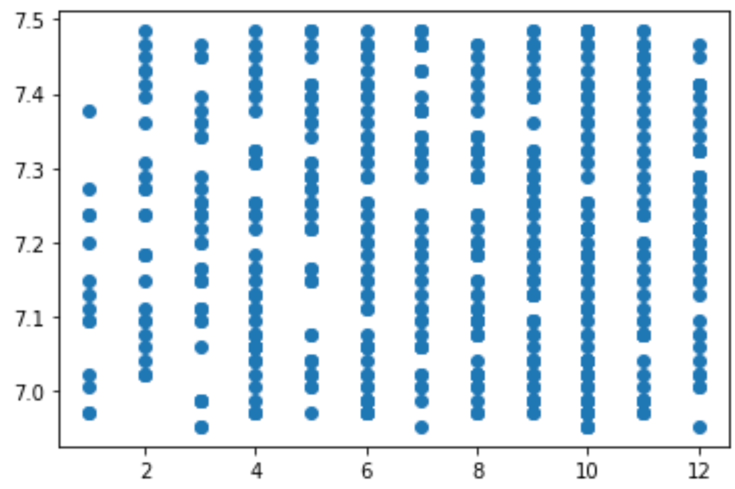
```
coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
```

Out[9]:

	Co-efficient
Date.Day	-0.017747

In [10]:
```
prediction= lr.predict(x_test)
```

Out[10]: <matplotlib.collections.PathCollection at 0x277a610d910>



In [11]:

Out[11]: -0.008014992423696743

LASSO AND RIDGE

In [12]:
```
from sklearn.linear_model import Ridge,Lasso
rr=Ridge(alpha=10)
```

Out[12]: Ridge(alpha=10)

In [13]:

Out[13]: -0.00801464851449385

In [14]:
```
la=Lasso(alpha=10)
```

Out[14]: Lasso(alpha=10)

In [15]:

Out[15]: -0.006835547905959194

ELASTICNET

In [16]:
```
from sklearn.linear_model import ElasticNet
a=ElasticNet()
```

Out[16]: ElasticNet()

In [17]:
```
print(a.coef_)
print(a.intercept_)
print(a.score(x_test,y_test))
```

```
[-0.01120072]
7.393213180123642
-0.006968863432721628
[7.12439579 7.09079362 7.38201246 7.23640304 7.37081173 7.23640304
 7.37081173 7.10199435 7.09079362 7.15799797 7.16919869 7.09079362
 7.32600883 7.13559652 7.23640304 7.09079362 7.20280086 7.04599072
 7.19160014 7.11319507 7.33720956 7.20280086 7.38201246 7.22520231
 7.12439579 7.18039942 7.15799797 7.0795929  7.25880449 7.11319507
 7.14679724 7.05719145 7.10199435 7.09079362 7.14679724 7.23640304
 7.22520231 7.23640304 7.12439579 7.22520231 7.23640304 7.27000521
 7.28120594 7.13559652 7.20280086 7.16919869 7.06839217 7.05719145
 7.22520231 7.34841028 7.19160014 7.31480811 7.21400159 7.28120594
 7.18039942 7.18039942 7.22520231 7.10199435 7.32600883 7.27000521
 7.27000521 7.15799797 7.04599072 7.15799797 7.35961101 7.16919869
 7.12439579 7.13559652 7.10199435 7.15799797 7.09079362 7.23640304
 7.22520231 7.29240666 7.24760376 7.28120594 7.06839217 7.20280086
 7.37081173 7.33720956 7.12439579 7.11319507 7.14679724 7.06839217
 7.35961101 7.21400159 7.30360738 7.12439579 7.19160014 7.0795929
 7.18039942 7.16919869 7.29240666 7.16919869 7.21400159 7.20280086
```

In [18]:
```
from sklearn import metrics
print(" Mean Absolute Error :",metrics.mean_absolute_error(y_test,prediction))
print(" Mean Squared Error :",metrics.mean_squared_error(y_test,prediction))
```

```
 Mean Absolute Error : 2.649414935840175
 Mean Squared Error : 9.407836897692961
 Root Mean Absolute Error : 1.6277023486621178
```

PREDICTION

In [19]:
```
import pickle
fn="prediction"
```

In [20]:
```
import pandas as pd
import pickle
fn="prediction"
```

In [21]:
```
r=[[10],[20]]
result=m.predict(r)
```

Out[21]:
```
array([7.3247476 , 7.14727929])
```