In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
data=pd.read_csv(r"C:\Users\user\Downloads\19_nuclear_explosions - 19_nuclear_explosions
data
```

Out[2]:

	WEAPON SOURCE COUNTRY	WEAPON DEPLOYMENT LOCATION	Data.Source	Location.Cordinates.Latitude	Location.Cordinates.Longitude	Da
0	USA	Alamogordo	DOE	32.54	-105.57	
1	USA	Hiroshima	DOE	34.23	132.27	
2	USA	Nagasaki	DOE	32.45	129.52	
3	USA	Bikini	DOE	11.35	165.20	
4	USA	Bikini	DOE	11.35	165.20	
...	...	...	...	...	...	
2041	CHINA	Lop Nor	HFS	41.69	88.35	
2042	INDIA	Pokhran	HFS	27.07	71.70	
2043	INDIA	Pokhran	NRD	27.07	71.70	
2044	PAKIST	Chagai	HFS	28.90	64.89	
2045	PAKIST	Kharan	HFS	28.49	63.78	

2046 rows × 16 columns

In [3]:
```
df=data.head(100)
df
```

Out[3]:

e	Data.Magnitude.Body	Data.Magnitude.Surface	Location.Cordinates.Depth	Data.Yeild.Lower	Data.Yeild.Upper
7	0.0	0.0	-0.100	21.0	21.0
7	0.0	0.0	-0.600	15.0	15.0
2	0.0	0.0	-0.600	21.0	21.0
)	0.0	0.0	-0.200	21.0	21.0
)	0.0	0.0	0.030	21.0	21.0
.	...	...	...	...	...
)	0.0	0.0	0.000	1600.0	1600.0
)	0.0	0.0	-0.001	0.0	20.0
)	0.0	0.0	0.000	0.3	0.3
)	0.0	0.0	0.000	14.0	14.0
)	0.0	0.0	0.000	5.5	5.5

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   WEAPON SOURCE COUNTRY           100 non-null    object
 1   WEAPON DEPLOYMENT LOCATION      100 non-null    object
 2   Data.Source                     100 non-null    object
 3   Location.Cordinates.Latitude    100 non-null    float64
 4   Location.Cordinates.Longitude   100 non-null    float64
 5   Data.Magnitude.Body             100 non-null    float64
 6   Data.Magnitude.Surface          100 non-null    float64
 7   Location.Cordinates.Depth       100 non-null    float64
 8   Data.Yeild.Lower                100 non-null    float64
 9   Data.Yeild.Upper                100 non-null    float64
 10  Data.Purpose                    100 non-null    object
 11  Data.Name                       100 non-null    object
 12  Data.Type                       100 non-null    object
 13  Date.Day                        100 non-null    int64
 14  Date.Month                      100 non-null    int64
 15  Date.Year                       100 non-null    int64
dtypes: float64(7), int64(3), object(6)
memory usage: 12.6+ KB
```

In [6]:
```
df.columns
```

Out[6]:
```
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')
```

In [15]:
```
x=df[[ 'Location.Cordinates.Latitude', 'Location.Cordinates.Longitude',
       'Location.Cordinates.Depth', 'Data.Yeild.Lower', 'Data.Yeild.Upper', 'Date.Month
y=df['Date.Year']
```

In [16]:
```
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)
```

In [17]:
```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[17]:
```
LinearRegression()
```

In [18]:
```
print(lr.intercept_)
```

```
1951.7190117003415
```

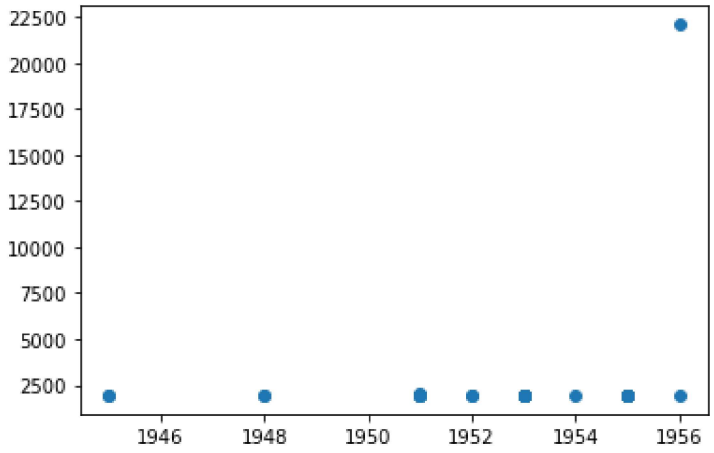
In [19]:
```
coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[19]:

	Co-efficient
Location.Cordinates.Latitude	5.577527e-02
Location.Cordinates.Longitude	9.830217e-07
Location.Cordinates.Depth	-1.062193e-01
Data.Yeild.Lower	-1.007850e+03
Data.Yeild.Upper	1.007850e+03
Date.Month	-1.515702e-01

In [20]:
```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[20]: <matplotlib.collections.PathCollection at 0x2153858f0a0>



In [21]:
```
print(lr.score(x_test,y_test))
```

```
-1668253.0697343305
```

In [22]:
```
print(lr.score(x_train,y_train))
```

```
0.19549753763732247
```

In [23]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [24]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[24]: Ridge(alpha=10)

In [25]:
```
rr.score(x_test,y_test)
```

Out[25]: 0.056974226708429776

In [26]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[26]: Lasso(alpha=10)

In [27]:
```
la.score(x_test,y_test)
```

Out[27]: 0.08076915742845991

In [28]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[28]: ElasticNet()

In [29]:
```
print(en.coef_)
```

```
[ 5.04733978e-02 -1.04656929e-03 -0.00000000e+00  1.85642372e-04
  1.14598462e-10 -7.90011997e-02]
```

In [30]:
```
print(en.intercept_)
```

```
1951.4780452617726
```

In [31]:
```
print(en.predict(x_test))
```

```
[1952.6842048  1953.22999655 1952.43809499 1953.13081332 1953.15894084
 1953.38814746 1951.57178513 1953.07715501 1953.23441484 1952.35224426
 1952.59794982 1954.39468094 1952.67695102 1949.10241844 1953.36731568
 1953.15114386 1953.28829592 1953.20936898 1953.29327114 1953.38796182
 1952.59817259 1953.2329297  1953.07734065 1952.59794982 1951.48702901
 1953.07548422 1953.68410018 1953.38944696 1953.15099535 1953.31044576]
```

In [32]:
```
print(en.score(x_test,y_test))
```

```
0.10625018884314852
```

Evaluation metrics

In [33]:
```
from sklearn import metrics
```

In [34]:
```
print("Mean absolute error",metrics.mean_absolute_error(y_test,prediction))
```

```
Mean absolute error 677.1758937858546
```

In [35]:
```
print("Mean squared error",metrics.mean_squared_error(y_test,prediction))
```

```
Mean squared error 13540662.19934365
```

In [36]:
```
print("Mean squared error",np.sqrt(metrics.mean_squared_error(y_test,prediction)))
```

```
Mean squared error 3679.763878205183
```

Model Saving

In [37]:
```
import pickle
```

In [38]:
```
filename='prediction'
pickle.dump(lr,open(filename,'wb'))
```

In [ ]:
```

```