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

In [3]:
```
df=pd.read_csv(r"C:\Users\Admin\Downloads\20_states - 20_states.csv")
df
```

Out[3]:

	id	name	country_id	country_code	country_name	state_code	type	latitude
0	3901	Badakhshan	1	AF	Afghanistan	BDS	NaN	36.734772
1	3871	Badghis	1	AF	Afghanistan	BDG	NaN	35.167134
2	3875	Baghlan	1	AF	Afghanistan	BGL	NaN	36.178903
3	3884	Balkh	1	AF	Afghanistan	BAL	NaN	36.755060
4	3872	Bamyan	1	AF	Afghanistan	BAM	NaN	34.810007
...	...	...	...	...	...	...	...	...
5072	1953	Mashonaland West Province	247	ZW	Zimbabwe	MW	NaN	-17.485103
5073	1960	Masvingo Province	247	ZW	Zimbabwe	MV	NaN	-20.624151
5074	1954	Matabeleland North Province	247	ZW	Zimbabwe	MN	NaN	-18.533157
5075	1952	Matabeleland South Province	247	ZW	Zimbabwe	MS	NaN	-21.052337
5076	1957	Midlands Province	247	ZW	Zimbabwe	MI	NaN	-19.055201

5077 rows × 9 columns

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5077 entries, 0 to 5076
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   id            5077 non-null   int64  
 1   name          5077 non-null   object 
 2   country_id    5077 non-null   int64  
 3   country_code  5063 non-null   object 
 4   country_name  5077 non-null   object 
 5   state_code    5072 non-null   object 
 6   type          1597 non-null   object 
 7   latitude      5008 non-null   float64
 8   longitude     5008 non-null   float64
dtypes: float64(2), int64(2), object(5)
memory usage: 357.1+ KB
```

In [5]:
```
df.columns
```

Out[5]:
```
Index(['id', 'name', 'country_id', 'country_code', 'country_name',
       'state_code', 'type', 'latitude', 'longitude'],
      dtype='object')
```

In [6]:
```
df1=df.head(100)
```

In [7]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   id            100 non-null    int64  
 1   name          100 non-null    object 
 2   country_id    100 non-null    int64  
 3   country_code  100 non-null    object 
 4   country_name  100 non-null    object 
 5   state_code    100 non-null    object 
 6   type          0 non-null      object 
 7   latitude      100 non-null    float64
 8   longitude     100 non-null    float64
dtypes: float64(2), int64(2), object(5)
memory usage: 7.2+ KB
```

In [8]:
```
x=df1[['country_id', 'latitude']]
y=df1[['longitude']]
```

In [9]:
```
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 [10]:
```
from sklearn.linear_model import LinearRegression
lr= LinearRegression()
lr.fit(x_train,y_train)
```

Out[10]:
```
LinearRegression()
```

In [11]:
```
print('Linear Regresion(score):',lr.score(x_test,y_test))
print('Linear Regression(train score)',lr.score(x_train,y_train))
```

```
Linear Regresion(score): 0.6645519944056819
Linear Regression(train score) 0.4086262164353551
```

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

Out[12]:
```
Ridge(alpha=10)
```

In [13]:
```
print('Ridge(test score):',rr.score(x_test,y_test))
```

```
Ridge(test score): 0.6341943446949351
```

In [14]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[14]:
```
Lasso(alpha=10)
```

In [15]:
```
print('Lasso (test score)',la.score(x_test,y_test))
```

```
Lasso (test score) 0.4405986697441683
```

In [16]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[16]:
```
ElasticNet()
```

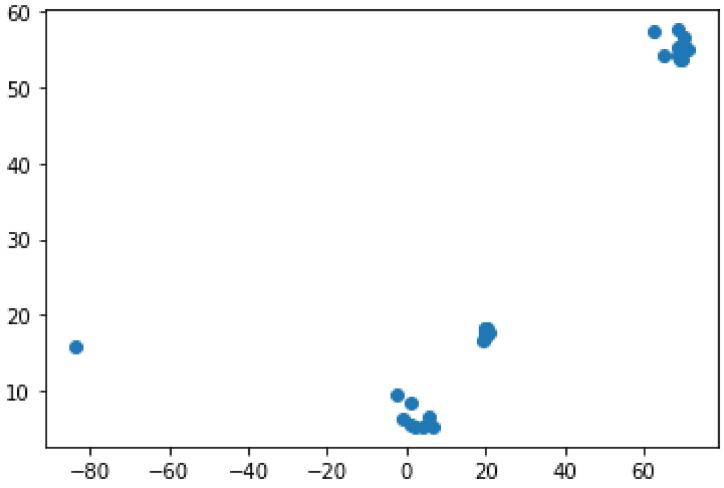
In [17]:
```
print(en.score(x_test,y_test))
```

```
0.5542522470713167
```

In [18]:
```
import pickle
file="predict"
pickle.dump(lr,open(file,'wb'))
```

In [19]:
```
prediction= lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[19]:
```
<matplotlib.collections.PathCollection at 0x2a8658a7820>
```



In [ ]:
```

```