DEENA 20104016

importing libraries

LINEAR REGRESSION

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

In [2]:
```
data = pd.read_csv("20_states.csv")
```

Out[2]:

	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 [3]:

Out[3]:

	id	name	country_id	country_code	country_name	state_code	type	latitude	lon
0	3901	Badakhshan	1	AF	Afghanistan	BDS	NaN	36.734772	70.8
1	3871	Badghis	1	AF	Afghanistan	BDG	NaN	35.167134	63.7
2	3875	Baghlan	1	AF	Afghanistan	BGL	NaN	36.178903	68.7
3	3884	Balkh	1	AF	Afghanistan	BAL	NaN	36.755060	66.8
4	3872	Bamyan	1	AF	Afghanistan	BAM	NaN	34.810007	67.8

In [4]:

Out[4]:

```
<bound method DataFrame.info of         id                          name  coun
try_id country_code country_name  \
0     3901                    Badakhshan          1           AF  Afghanistan
1     3871                       Badghis          1           AF  Afghanistan
2     3875                       Baghlan          1           AF  Afghanistan
3     3884                         Balkh          1           AF  Afghanistan
4     3872                        Bamyan          1           AF  Afghanistan
...    ...                           ...        ...          ...          ...
5072  1953    Mashonaland West Province        247           ZW     Zimbabwe
5073  1960            Masvingo Province        247           ZW     Zimbabwe
5074  1954  Matabeleland North Province        247           ZW     Zimbabwe
5075  1952  Matabeleland South Province        247           ZW     Zimbabwe
5076  1957            Midlands Province        247           ZW     Zimbabwe

     state_code type   latitude  longitude
0           BDS  NaN  36.734772  70.811995
1           BDG  NaN  35.167134  63.769538
2           BGL  NaN  36.178903  68.745306
3           BAL  NaN  36.755060  66.897537
4           BAM  NaN  34.810007  67.821210
...         ...  ...        ...        ...
5072         MW  NaN -17.485103  29.788925
5073         MV  NaN -20.624151  31.262637
5074         MN  NaN -18.533157  27.549585
5075         MS  NaN -21.052337  29.045993
5076         MI  NaN -19.055201  29.603549

[5077 rows x 9 columns]>
```

In [5]:

Out[5]:

	id	country_id	latitude	longitude
count	5077.000000	5077.000000	5008.000000	5008.000000
mean	2609.765413	133.467599	27.576415	17.178713
std	1503.376799	72.341160	22.208161	61.269334
min	1.000000	1.000000	-54.805400	-178.116500
25%	1324.000000	74.000000	11.399747	-3.943859
50%	2617.000000	132.000000	34.226432	17.501792
75%	3905.000000	201.000000	45.802822	41.919647
max	5220.000000	248.000000	77.874972	179.852222

Train the model

In [6]:
```
x = data[['country_id']]
```

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()

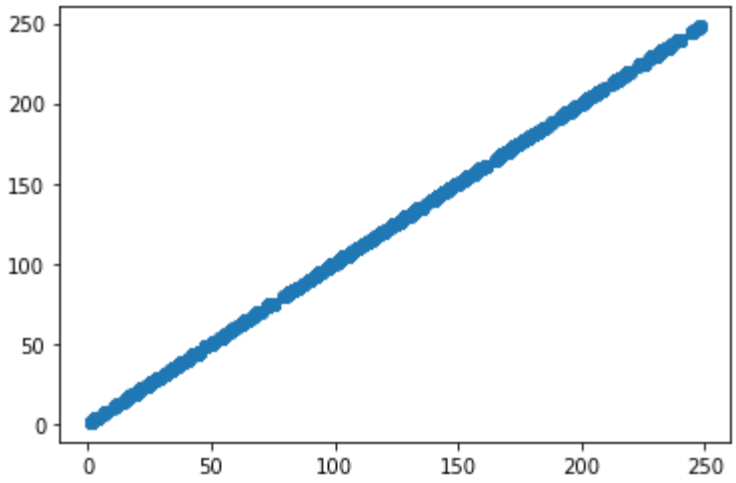
In [9]:
```
coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
```

Out[9]:

	Co-efficient
country_id	1.0

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

Out[10]: <matplotlib.collections.PathCollection at 0x1c9763678e0>



In [11]:

Out[11]: 1.0

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]: 0.9999999999997199

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

Out[14]: Lasso(alpha=10)

In [15]:

Out[15]: 0.9999964636950108

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))
```

```
[0.99981207]
0.02495084714635709
0.9999999646435956
[231.98135089  90.00803707 107.00484225 ... 131.00033191 218.98379399
 104.00540604]
```

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 : 5.301970586103602e-14
 Mean Squared Error : 3.912476266130561e-27
 Root Mean Absolute Error : 2.302600830822312e-07
```

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([10., 20.])
```