# DEENA 20104016

## importing libraries

## LINEAR REGRESSION

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

In [2]:
```
data = pd.read_csv("21_cities.csv")
```

Out[2]:

| | id | name | state_id | state_code | state_name | country_id | country_code | country |
|---|---|---|---|---|---|---|---|---|
| **0** | 52 | Ashkāsham | 3901 | BDS | Badakhshan | 1 | AF | Afgh |
| **1** | 68 | Fayzabad | 3901 | BDS | Badakhshan | 1 | AF | Afgh |
| **2** | 78 | Jurm | 3901 | BDS | Badakhshan | 1 | AF | Afgh |
| **3** | 84 | Khandūd | 3901 | BDS | Badakhshan | 1 | AF | Afgh |
| **4** | 115 | Rāghistān | 3901 | BDS | Badakhshan | 1 | AF | Afgh |
| **...** | ... | ... | ... | ... | ... | ... | ... | |
| **150449** | 131496 | Redcliff | 1957 | MI | Midlands Province | 247 | ZW | Zir |
| **150450** | 131502 | Shangani | 1957 | MI | Midlands Province | 247 | ZW | Zir |
| **150451** | 131503 | Shurugwi | 1957 | MI | Midlands Province | 247 | ZW | Zir |
| **150452** | 131504 | Shurugwi District | 1957 | MI | Midlands Province | 247 | ZW | Zir |
| **150453** | 131508 | Zvishavane District | 1957 | MI | Midlands Province | 247 | ZW | Zir |

150454 rows × 11 columns

In [3]: [illegible]

Out[3]:

| | id | name | state_id | state_code | state_name | country_id | country_code | country_name |
|---|---|---|---|---|---|---|---|---|
| **0** | 52 | Ashkāsham | 3901 | BDS | Badakhshan | 1 | AF | Afghanistan |
| **1** | 68 | Fayzabad | 3901 | BDS | Badakhshan | 1 | AF | Afghanistan |
| **2** | 78 | Jurm | 3901 | BDS | Badakhshan | 1 | AF | Afghanistan |
| **3** | 84 | Khandūd | 3901 | BDS | Badakhshan | 1 | AF | Afghanistan |
| **4** | 115 | Rāghistān | 3901 | BDS | Badakhshan | 1 | AF | Afghanistan |

In [4]: [illegible]

Out[4]:

```
<bound method DataFrame.info of             id                 name  state_id
state_code         state_name  \
0           52            Ashkāsham      3901        BDS          Badakhshan
1           68             Fayzabad      3901        BDS          Badakhshan
2           78                 Jurm      3901        BDS          Badakhshan
3           84              Khandūd      3901        BDS          Badakhshan
4          115            Rāghistān      3901        BDS          Badakhshan
...        ...                  ...       ...        ...                 ...
150449  131496             Redcliff      1957         MI   Midlands Province
150450  131502             Shangani      1957         MI   Midlands Province
150451  131503             Shurugwi      1957         MI   Midlands Province
150452  131504    Shurugwi District      1957         MI   Midlands Province
150453  131508  Zvishavane District      1957         MI   Midlands Province

        country_id country_code country_name  latitude  longitude wikiDataId
0                1           AF  Afghanistan  36.68333   71.53333   Q4805192
1                1           AF  Afghanistan  37.11664   70.58002    Q156558
2                1           AF  Afghanistan  36.86477   70.83421  Q10308323
3                1           AF  Afghanistan  36.95127   72.31800   Q3290334
4                1           AF  Afghanistan  37.66079   70.67346   Q2670909
...            ...          ...          ...       ...        ...        ...
150449         247           ZW     Zimbabwe -19.03333   29.78333    Q584001
150450         247           ZW     Zimbabwe -19.78333   29.36667  Q32017959
150451         247           ZW     Zimbabwe -19.67016   30.00589  Q32019023
150452         247           ZW     Zimbabwe -19.75000   30.16667   Q7505444
150453         247           ZW     Zimbabwe -20.30345   30.07514  Q24235929

[150454 rows x 11 columns]>
```

In [5]:

Out[5]:

| | id | state_id | country_id | latitude | longitude |
|---|---|---|---|---|---|
| **count** | 150454.000000 | 150454.000000 | 150454.000000 | 150454.000000 | 150454.000000 |
| **mean** | 76407.091689 | 2678.377677 | 140.658460 | 31.556175 | 2.369557 |
| **std** | 44357.755335 | 1363.513591 | 70.666123 | 22.813220 | 68.012770 |
| **min** | 1.000000 | 1.000000 | 1.000000 | -75.000000 | -179.121980 |
| **25%** | 38160.250000 | 1451.000000 | 82.000000 | 19.000000 | -58.468150 |
| **50%** | 75975.500000 | 2174.000000 | 142.000000 | 40.684720 | 8.669980 |
| **75%** | 115204.750000 | 3905.000000 | 207.000000 | 47.239220 | 27.750000 |
| **max** | 153528.000000 | 5116.000000 | 247.000000 | 73.508190 | 179.466000 |

# Train the model

In [6]:

```
x = data[['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]:

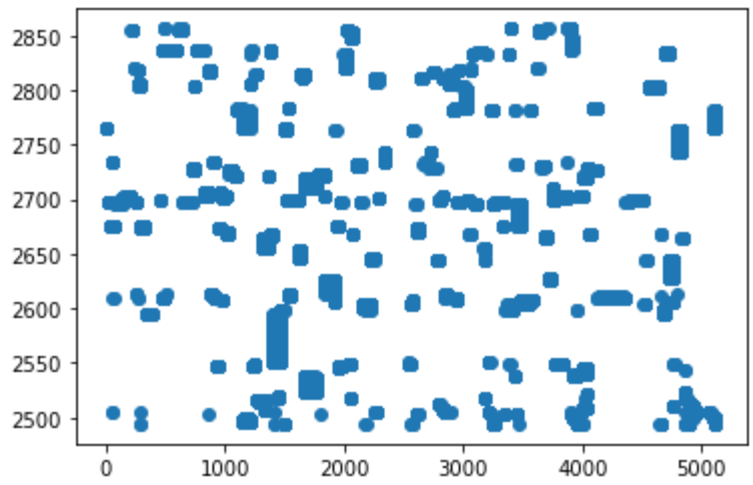
```
coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
```

Out[9]:

| | Co-efficient |
|---|---|
| **id** | -0.00236 |

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

Out[10]: <matplotlib.collections.PathCollection at 0x21aa9cbc8e0>
```



```
In [11]:

Out[11]: 0.006921032780112135
```

## 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]: 0.006921032780112135
```

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

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

```
In [15]:

Out[15]: 0.006921030424558938
```

## 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))
```

```
[-0.00235992]
2856.3994709831677
0.00692103266205768
[2758.29988029 2832.27635    2596.18504995 ... 2543.42663745 2565.88129369
 2628.66229422]
```

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 : 1206.9559267949646
 Mean Squared Error : 1844738.5077261804
 Root Mean Absolute Error : 34.74127123170602
```

# 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([2856.37589117, 2856.35229194])
```