ysneenac8

July 31, 2023

[11]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[12]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[13]:
```
df=pd.read_csv("/content/drive/MyDrive/mydatasets/20_states.csv")
df
```

[13]:

	id	name	country_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]
```

[14]:
```
df.head()
```

[14]:
```
     id        name  country_id country_code country_name state_code type  \
0  3901  Badakhshan           1           AF  Afghanistan        BDS  NaN
1  3871     Badghis           1           AF  Afghanistan        BDG  NaN
2  3875     Baghlan           1           AF  Afghanistan        BGL  NaN
3  3884       Balkh           1           AF  Afghanistan        BAL  NaN
4  3872      Bamyan           1           AF  Afghanistan        BAM  NaN

    latitude  longitude
0  36.734772  70.811995
1  35.167134  63.769538
2  36.178903  68.745306
3  36.755060  66.897537
4  34.810007  67.821210
```

1 Data Cleaning and Data Preprocessing

[15]:
```
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
```

[16]:
```
df.describe()
```

[16]:
```
                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
```

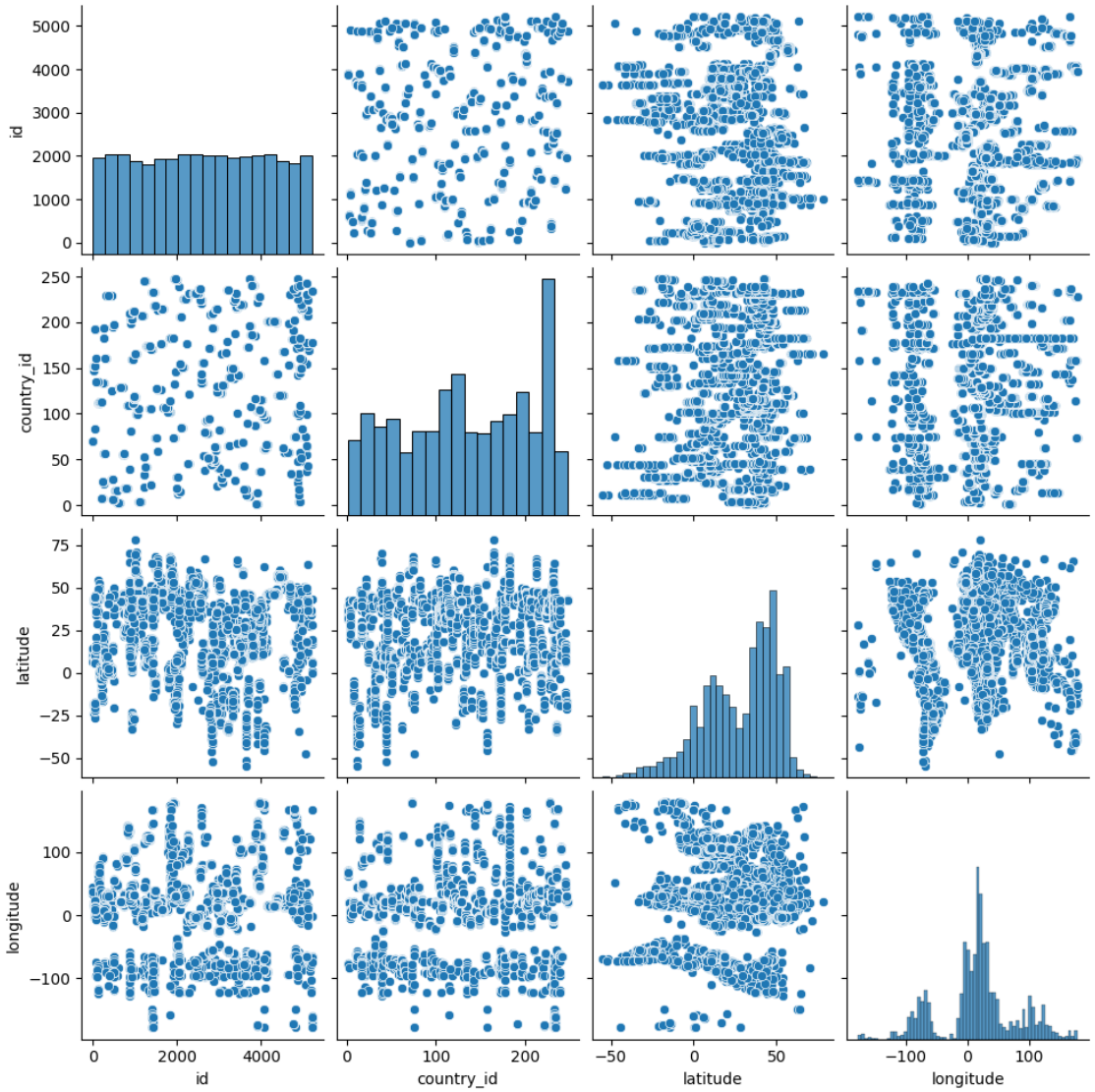
[17]:
```
df.columns
```

[17]:
```
Index(['id', 'name', 'country_id', 'country_code', 'country_name',
       'state_code', 'type', 'latitude', 'longitude'],
      dtype='object')
```

2 EDA and Visualization

[18]:
```
sns.pairplot(df)
```

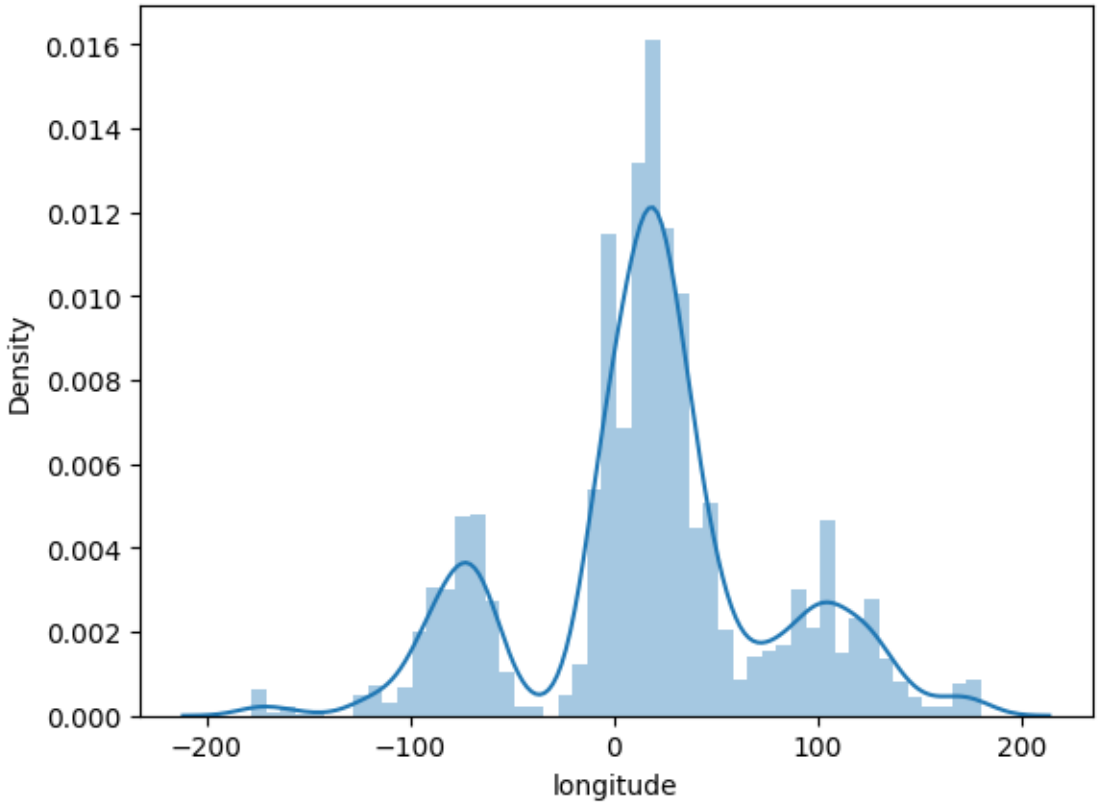
[18]:
```
<seaborn.axisgrid.PairGrid at 0x7c1e28bc98d0>
```



[19]:
```
sns.distplot(df['longitude'])
```

```
<ipython-input-19-4c5c6f107715>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['longitude'])
```

[19]: <Axes: xlabel='longitude', ylabel='Density'>



[20]:
```
df1=df[['id', 'name', 'country_id', 'country_code', 'country_name',
       'state_code', 'type', 'latitude', 'longitude']].dropna()
df1
```

[20]:

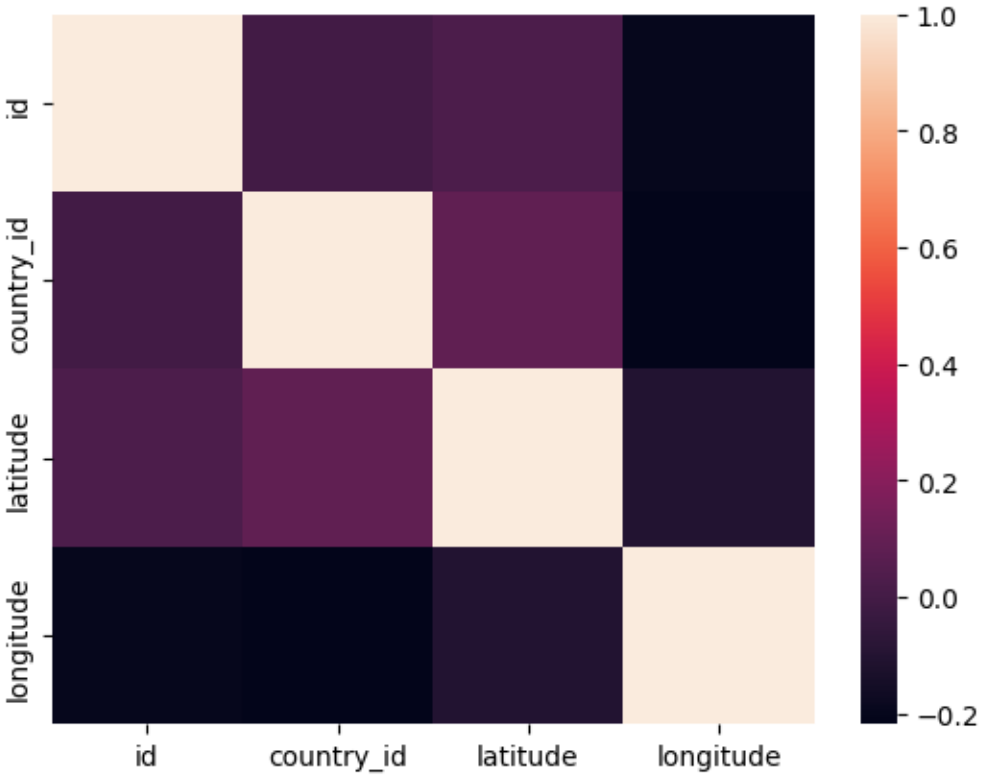
	id	name	country_id	country_code	country_name	state_code	type	latitude	longitude
170	3656	Buenos Aires	11	AR	Argentina	B	province	-37.201729	-59.841070
171	3647	Catamarca	11	AR					
172	3640	Chaco	11	AR					
173	3651	Chubut	11	AR					
174	4880	Ciudad Autónoma de Buenos Aires	11	AR					
…	…	…	…	…					
4968	2041	Yaracuy	239	VE					
4969	2042	Zulia	239	VE					
5033	5074	Saint Croix	242	VI					
5034	5073	Saint John	242	VI					
5035	5072	Saint Thomas	242	VI					

```
171              Argentina          K  province -28.471588 -65.787721
172              Argentina          H  province -27.425718 -59.024378
173              Argentina          U  province -43.293425 -65.111482
174              Argentina          C      city -34.603684 -58.381559
…                    …        …        …         …         …
4968             Venezuela          U     state  10.339389 -68.810885
4969             Venezuela          V     state  10.291024 -72.141613
5033  Virgin Islands (US)         SC  district  17.729352 -64.734370
5034  Virgin Islands (US)         SJ  district  18.335617 -64.800280
5035  Virgin Islands (US)         ST  district  18.342846 -65.077018

[1580 rows x 9 columns]
```

[21]:
```
sns.heatmap(df1.corr())
```

```
<ipython-input-21-3ed1a1a51dc0>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  sns.heatmap(df1.corr())
```

[21]: <Axes: >



[22]:
```
x=df1[['id', 'country_id', 'latitude']]
y=df1['longitude']
```

[23]:
```
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)
```

[24]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

[24]:
```
LinearRegression()
```

[25]:
```
print(lr.intercept_)
```

```
72.72321403916355
```

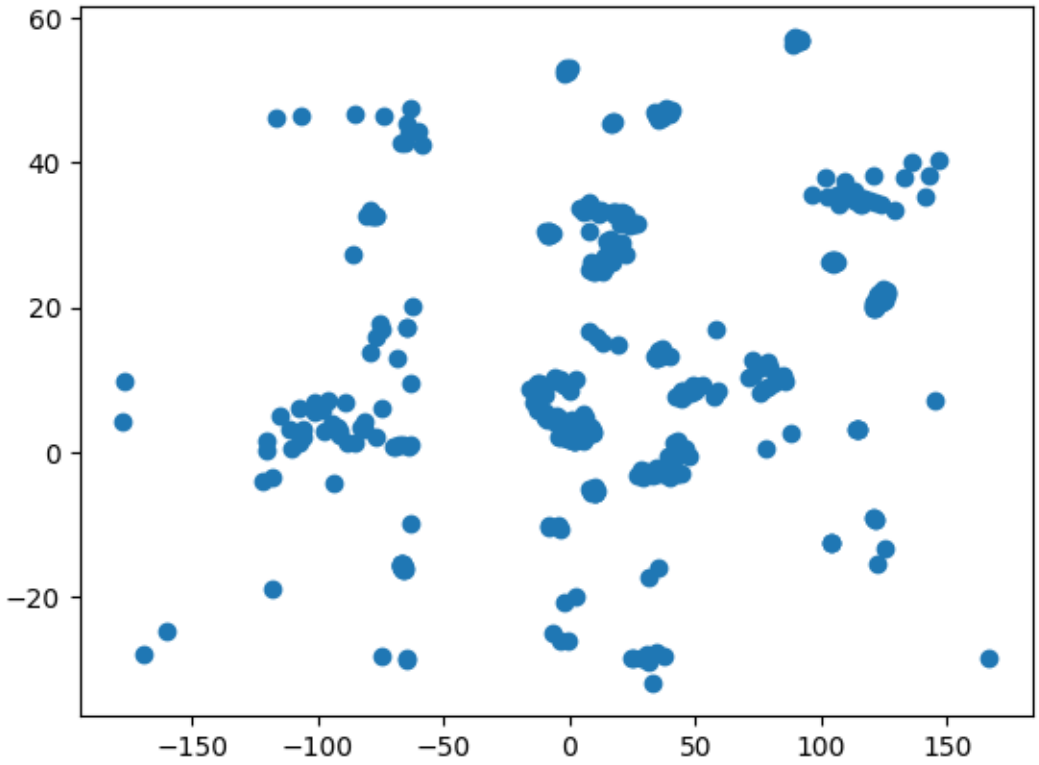
[26]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

[26]:

	Co-efficient
id	-0.009056
country_id	-0.215235
latitude	-0.190062

[27]:
```
prediction =lr.predict(x_test)
plt.scatter(y_test,prediction)
```

[27]:
```
<matplotlib.collections.PathCollection at 0x7c1e25593a30>
```



[28]:
```
lr.score(x_test,y_test)
```

[28]: 0.0739373053113842

[29]:
```
lr.score(x_train,y_train)
```

[29]: 0.10074696331611066

[30]:
```
from sklearn.linear_model import Ridge,Lasso
```

[31]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

[31]: Ridge(alpha=10)

[32]:
```
rr.score(x_test,y_test)
```

[32]: 0.0739370746544511

[33]:
```
rr.score(x_train,y_train)
```

[33]: 0.10074696331426991

[34]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

[34]:
```
Lasso(alpha=10)
```

[35]:
```
la.score(x_test,y_test)
```

[35]:
```
0.07254140312306234
```

[36]:
```
la.score(x_train,y_train)
```

[36]:
```
0.10068580938627081
```

[37]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

[37]:
```
ElasticNet()
```

[38]:
```
en.coef_
```

[38]:
```
array([-0.00905636, -0.21512737, -0.18855583])
```

[39]:
```
en.intercept_
```

[39]:
```
72.66933901255787
```

[40]:
```
prediction = en.predict(x_test)
prediction
```

[40]:
```
array([ 13.14555944,   1.95555306,  -3.07899675,  40.14923419,
        26.86763505,   2.6591175 ,  13.04485257,  -9.50913675,
         4.32337016,   1.22005097,  34.99904987,   2.75117512,
         2.85000091,  46.50339272,  35.20346326,  57.05334995,
       -16.21349247,  14.11084797,   2.97755597,   6.82695261,
        -5.40873848,  42.76726256,   4.04011649,  33.32283664,
       -27.7511951 ,   3.01637027,  32.97199889,   3.22513075,
         3.85587573, -16.01044373,  19.99375858,   4.56632189,
         2.16357252,  32.71882327,   2.82007391, -28.34699875,
       -10.22232751,  46.39104274,  33.81502981, -15.58828091,
        47.00746213,  -2.94140104,  35.69078799,   1.8031218 ,
         4.20298677,  30.08530811,  33.59089856,   6.04123974,
        56.5507371 ,   0.85291202,  21.37304295,   8.39559127,
        34.01501759,  25.66475639,   7.13470698,  -2.79280158,
        -9.16111643,  21.83497615,  -0.58395535,  30.07610785,
        52.90412458,  45.53548242,   1.66442978, -16.17961851,
         2.44528836,  26.66098282,   2.90063087,   3.30264832,
```

```
 45.40140737, -28.48297054,   3.01553243,  42.53539271,
 25.94988959,  46.67035977,  -2.51876367,  -5.19190006,
 42.64288907,  56.39026778,  32.60565891,  46.21279628,
  9.45470522,  11.74976408,  32.48798974,  31.73686023,
 -2.5064696 ,   9.3225697 ,  32.99702889,  -2.92919018,
 46.41821181, -28.68640982,  19.8872298 ,   2.65668927,
 33.06320863,  31.59618295,  33.41473343,  26.28450367,
 14.82390213,  10.64775714,  32.87236995,  32.96629325,
 26.36126773,   4.64449369,   1.45651496, -26.06368977,
  2.79589555,  11.56162041,   9.21656781,  -3.0667955 ,
  3.04587913,  -5.38362753,  24.97518153,   4.67136622,
  4.35410645,  33.29043473, -28.43646454,  28.65467147,
 26.38545984,   4.16615225,   4.53960236,  35.53552939,
  4.49091126,  34.35569534,   4.00895138,  -2.7726044 ,
 -3.47246006,   2.82517127,   0.71595734,  32.96726964,
  2.99769361,  30.50548155, -25.06802738,   7.88419639,
-24.81972226,  -3.31311074, -19.99357816,   3.0883118 ,
 32.9093801 ,  47.62141931,  15.97952959,  17.1140729 ,
  4.28763189, -15.89232125,  -2.92453405,  13.24451523,
  3.27982659,   2.20481058,  -5.6138973 , -18.90560493,
  4.36551998,  46.60923942,  22.00726089,  -2.18442616,
 35.75515101,   9.11002753,  30.22298436,  33.77995905,
  2.98229094,   9.57540718,   3.10829316,   2.82267845,
 -5.57265798,  17.16937768,   1.27447602, -28.57993084,
 -5.59505013,  -2.54572806,  57.0675684 ,  37.92242049,
  3.5072812 ,  56.8534366 ,  -9.47454965,  47.27810909,
 56.93906962,   4.00231414,  30.29570592,  32.68400061,
-28.79166825,  32.63122672,  -9.18010405,  44.20750079,
 46.77236893,  56.81153972,   6.78496065,  20.78970582,
 20.28993851, -16.03097298,   7.6717739 , -16.11305167,
 26.01781822,  25.6338616 ,  30.15442222,   4.45925012,
  8.5800875 ,  56.9969091 ,  25.1848114 , -15.62936835,
 32.72525204,  56.92433711,  57.01640341,  26.24491568,
  7.22208528,  34.5768127 ,  45.6416459 ,  10.08448499,
  8.66992448,  33.70329203,  19.87526111, -28.6701392 ,
 28.27860588,  34.26051815, -26.02085749,   7.46742255,
 46.68710439,   8.35613579,  22.02507987,  25.36055012,
 46.65513891,  10.24735922,  45.86979358,  -4.16322785,
 20.05696106,  21.92430464,   1.49601154, -32.09520151,
-15.65572829,  46.72519554,   2.65857926,  25.71182825,
-13.34876073,  47.39129196,  -3.01196965,  38.23661362,
 -5.55444112,   8.42712138, -12.50505692,  32.93906391,
 30.49324198,  52.56061308,   2.80108384,   6.74149416,
 -4.30444279,  44.1058803 , -16.2579095 ,  33.22021069,
 16.08141209,  52.36429699,  33.69165805,   5.58456808,
  4.19736006,  -5.62886851,  26.25321878,   1.36317169,
 25.29406347,   1.08398728,   2.19450039,  27.02368393,
```

```
 32.65651895,   5.33928982,  53.10686812,   8.17868365,
  2.29434574,  20.48622174,  35.54219631,  17.75154692,
  9.49778374,   1.84903664,   7.57445426,   3.00158868,
  3.01404638, -28.05380992,  16.8958126 ,   8.90188526,
  9.35043079,  33.31844331,  21.88321976,  -4.75353832,
  2.98796477,  21.06131716, -12.54760647,  30.17936135,
  2.47717851,  46.81991733,  22.19097464,   9.69697885,
 31.58445647,  27.02948288,  13.75650004,  21.68994849,
  2.35938456,   5.67348108,  47.08270968,   1.23864155,
 16.96949051,  56.99256595, -10.32453041, -15.61138209,
  9.55128299,  25.57830632,   8.52017588,   5.45054747,
-16.04937393,  25.24398247,  20.67039491,  -0.63030334,
 31.72022482,  57.25886906,   9.91830374,  15.12663749,
  4.07915997,   4.9464428 ,  56.86629573,  11.59242125,
  3.09634984,  39.95825801,  -0.57481607,   6.13953111,
  4.85460304,  -3.61050126,  45.73107354,   2.14192188,
 34.28056092, -15.94873261,  32.77897798,  31.31516379,
-15.62862591,   2.89147728,  -2.94023489,   1.91740185,
 10.57114503,   1.83112554,   1.49475523,  -3.35505608,
 21.46974319,  25.81615354,  33.13456832,  10.2864007 ,
 57.10985692,  32.84971028,  12.81159497,  19.93323027,
  2.42090907,  25.38237688, -15.64550203, -16.21048102,
 25.02742048,  31.54255152,   7.64888828,   2.81928964,
-28.03109414,  25.16664765,   0.60973375,  20.51299884,
 -2.96406388, -20.88431101,   2.24550812,  16.69718307,
 21.86602713,   2.77857341, -28.37726906,  12.30099219,
 53.22840732,   5.7743215 ,  20.64413733,   9.73089666,
 -2.87149869,   4.71037845,   8.96801719,   3.13897314,
 26.6684593 ,   1.47039346,  25.38704478, -10.6364678 ,
  0.54636105,  26.08732218,  34.23466532,  12.5965681 ,
  9.70879176, -10.26797063,  46.97054323, -17.55188112,
 46.51608995,  38.09816313,  25.59326192,  30.50516827,
 26.38748592,  13.04684484,  28.18492411,   7.74568035,
 22.23332094, -16.09844053,  29.22381261, -15.78797009,
 -2.91804539,  35.02424116,  -3.28962638,  46.97974513,
  0.53604039,  45.51782863,  33.0260132 , -10.05180397,
 33.28356313,   3.13772349,  29.32964481,  26.5738538 ,
-29.01266627,   9.77629861,  -2.20095657,   5.03437586,
  9.64895223,   4.64337294,  -5.14957117,  30.56867124,
 13.21619671,  34.18407856,  46.83103146,  35.38810175,
 20.95589196,  -3.40557575,  32.5666482 ,  26.37852095,
  2.03325044,   2.71959639,  25.63427964,   0.59669263,
 14.12060671,  28.93623157,  34.80722628, -15.66637537,
 46.34166859,   8.72388453,  46.25610053,   2.82992062,
 33.02385687,  26.14655523,  33.37723773,   0.87119037,
  4.99824689,  28.82013014,   8.08238989,  27.38254996,
 34.28711307,  22.45039701,  25.08816286,  10.02989923,
```

```
 21.86407624,  25.83886778,   0.8940163 ,  56.99553807,
  3.70717368,   1.82605901,  21.71125491,   4.78699936,
  8.32821634,   4.3021245 ,   1.64214027,   0.30052602,
 35.60135285,  37.90628396,  37.37890161,   4.64727591,
-12.50466845, -15.5934094 ,   2.69423485,  46.9832708 ,
 33.01667063,  21.46449035,   1.56797697, -28.39785939,
  2.71198464,  26.36319208,  27.4385563 ,   3.14853133,
 36.06463744,  34.79629818])
```

[41]:
```
en.score(x_test,y_test)
```

[41]: 0.07385781285585713

[42]:
```
from sklearn import metrics
```

[43]:
```
print("Mean Absolute Error: ", metrics.mean_absolute_error(y_test,prediction))
```

```
Mean Absolute Error:  47.69355568407093
```

[44]:
```
print("Mean Squared Error: ", metrics.mean_squared_error(y_test,prediction))
```

```
Mean Squared Error:  3934.5370042068157
```

[45]:
```
print("Root Mean Squared Error: ", np.sqrt(metrics.
↪mean_squared_error(y_test,prediction)))
```

```
Root Mean Squared Error:  62.725887831156406
```

[46]:
```
import pickle
filename='prediction'
pickle.dump(lr,open(filename,'wb'))
```

[47]:
```
model = pickle.load(open(filename, 'rb'))
real=[[10,20,30],[11,45,10]]
result = model.predict(real)
result
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does
not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

[47]: array([62.62608138, 61.03738127])