Implement multiple Linear Regression using 'cost-of-living-2018' dataset and find the performance of the model using error finding methods.

In [59]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
```

In [60]:
```
df=pd.read_csv('C:\python\cost-of-living-2018.csv')
```

In [61]:
```
df.head()
```

Out[61]:

	Rank	City	Cost of Living Index	Rent Index	Cost of Living Plus Rent Index	Groceries Index	Restaurant Price Index	Local Purchasing Power Index
0	NaN	Hamilton, Bermuda	145.43	110.87	128.76	143.47	158.75	112.26
1	NaN	Zurich, Switzerland	141.25	66.14	105.03	149.86	135.76	142.70
2	NaN	Geneva, Switzerland	134.83	71.70	104.38	138.98	129.74	130.96
3	NaN	Basel, Switzerland	130.68	49.68	91.61	127.54	127.22	139.01
4	NaN	Bern, Switzerland	128.03	43.57	87.30	132.70	119.48	112.71

In [62]:
```
x=df.drop(['Rank','City','Rent Index'],axis=1)
x
```

Out[62]:

	Cost of Living Index	Cost of Living Plus Rent Index	Groceries Index	Restaurant Price Index	Local Purchasing Power Index
0	145.43	128.76	143.47	158.75	112.26
1	141.25	105.03	149.86	135.76	142.70
2	134.83	104.38	138.98	129.74	130.96
3	130.68	91.61	127.54	127.22	139.01
4	128.03	87.30	132.70	119.48	112.71
...	...	...	...	...	...
535	24.65	15.80	26.93	13.94	77.70
536	24.61	15.32	25.23	15.21	53.23
537	23.78	14.40	23.19	17.66	23.75
538	23.44	15.15	24.02	14.14	111.99
539	20.86	13.26	21.98	12.06	66.25

540 rows × 5 columns

In [63]:
```
y=df.iloc[:,3].values
y
```

Out[63]:
```
array([110.87,  66.14,  71.7 ,  49.68,  43.57,  52.32,  57.25,  39.83,
        52.91,  49.28,  42.39,  40.3 ,  24.58, 100.  ,  40.45, 115.36,
        50.66,  61.59,  40.12,  62.82,  37.07,  76.24,  50.3 ,  27.71,
        64.  ,  33.64,  73.3 ,  34.82,  71.89,  26.81,  45.27,  68.43,
        24.95,  51.55,  58.13,  62.13,  38.43,  29.43,  61.85,  58.22,
        66.67,  76.67,  28.06,  26.98,  64.41,  78.69,  40.52,  25.07,
        37.17,  28.89,  78.72,  46.95,  31.81,  31.64,  37.34,  40.5 ,
        45.99,  36.56,  63.2 ,  37.62,  40.91,  29.48,  26.92,  44.81,
        79.57,  43.65,  21.49,  46.52,  30.59,  35.55,  46.46,  28.04,
        30.46,  47.53,  66.26,  35.76,  48.09,  40.23,  32.13,  47.33,
        39.97,  31.46,  81.81,  24.97,  36.93,  35.28,  36.82,  36.34,
        68.75,  53.8 ,  22.48,  25.55,  43.31,  29.33,  26.2 ,  20.68,
        23.89,  37.59,  40.73,  45.65,  43.59,  29.7 ,  45.03,  40.42,
        42.15,  64.73,  33.61,  25.04,  33.35,  24.94,  30.13,  34.58,
        30.82,  56.71,  32.23,  66.  ,  28.57,  28.42,  27.07,  53.29,
        36.06,  34.6 ,  28.03,  71.45,  26.08,  31.38,  36.17,  41.89,
        40.24,  43.34,  35.2 ,  79.54,  39.02,  35.04,  23.88,  51.72,
        40.38,  24.7 ,  54.98,  29.69,  21.54,  29.01,  40.01,  22.05,
        51.63,  42.48,  37.85,  54.  ,  37.21,  30.87,  34.7 ,  43.45,
        29.38,  61.91,  27.25,  30.91,  40.21,  37.35,  45.03,  42.09,
        43.11,  31.37,  44.89,  25.51,  31.61,  24.8 ,  23.89,  26.45,
        53.5 ,  40.  ,  31.21,  25.34,  40.62,  40.02,  46.94,  45.83,
        23.25,  30.31,  24.85,  38.76,  34.84,  35.67,  31.36,  34.86,
        33.54,  34.2 ,  25.05,  28.12,  22.45,  26.05,  36.03,  24.4 ,
        29.75,  29.24,  32.27,  42.07,  28.35,  22.22,  35.5 ,  30.28,
        41.01,  31.86,  27.52,  28.86,  30.77,  29.64,  28.93,  40.31,
        21.57,  66.88,  25.08,  36.65,  49.8 ,  20.19,  21.61,  30.94,
        28.94,  25.24,  21.38,  25.11,  47.73,  24.48,  18.63,  29.57,
        28.32,  33.09,  33.61,  37.42,  24.83,  22.95,  24.91,  17.97,
        19.85,  36.18,  29.98,  30.11,  32.95,  30.31,  44.1 ,  35.48,
        30.69,  45.26,  31.76,  27.78,  25.69,  32.  ,  21.96,  33.71,
        29.48,  23.92,  24.96,  22.04,  21.43,  24.58,  31.23,  37.42,
        26.97,  29.09,  26.68,  19.72,  35.51,  31.11,  29.33,  23.7 ,
        26.09,  21.48,  28.38,  26.6 ,  31.9 ,  25.51,  35.29,  30.43,
        34.12,  33.41,  33.36,  21.14,  24.63,  24.63,  27.77,  33.32,
        25.7 ,  34.91,  28.61,  35.54,  18.79,  24.96,  17.22,  66.84,
        13.  ,  42.64,  25.5 ,  30.04,  36.95,  20.62,  15.21,  24.63,
        27.6 ,  28.33,  34.98,  64.25,  11.57,  28.72,  13.97,  26.54,
        27.36,  24.22,  27.45,  23.8 ,  35.99,  12.01,  28.91,  29.77,
        32.3 ,  32.7 ,  20.8 ,  34.5 ,  26.67,  17.66,  19.28,  20.61,
        10.41,  21.15,  11.36,  29.86,  13.74,  30.65,  12.29,  14.58,
        15.39,  29.33,   9.41,  32.46,  21.19,  14.1 ,  21.61,  17.91,
        17.85,  21.02,  41.95,  15.58,  26.21,  26.91,  14.44,  20.87,
        15.23,  14.58,  14.9 ,  17.9 ,  14.79,  16.73,  25.06,  22.87,
        18.05,  15.59,  14.65,  20.09,  12.38,  12.23,  11.61,   9.81,
        23.11,  34.1 ,  12.28,  16.16,  25.41,  18.08,  14.41,  11.96,
        14.24,  12.56,  15.08,  24.39,  19.26,  12.78,  15.67,  22.45,
        12.34,  24.56,  25.37,  10.67,  13.57,  12.63,  17.42,  12.49,
        13.91,   8.14,  17.8 ,  12.68,  14.05,  22.5 ,  16.46,  12.1 ,
        14.6 ,  10.48,  28.98,  18.59,  13.95,  15.52,   6.86,  19.34,
        20.17,  15.74,  17.96,  19.64,  33.51,  14.76,  11.34,  13.01,
        18.18,  18.58,  15.45,  10.34,  15.1 ,  10.73,  14.35,  11.1 ,
        12.73,  14.  ,  12.28,  17.91,   8.12,  12.79,  20.93,  12.96,
```

```
10.13,  19.94,  18.04,  12.37,   8.71,  12.27,  12.4 ,  11.84,
10.78,  13.47,  18.81,   7.95,   8.86,   7.29,  13.6 ,   7.65,
 6.73,   6.09,  14.56,  12.16,   7.77,  11.38,  10.55,  17.11,
12.81,   8.46,   9.96,  12.73,   8.6 ,  12.21,  13.86,   7.42,
 7.47,  16.45,   7.18,   9.26,  11.52,  10.38,   9.96,  11.34,
 4.71,  19.93,   9.88,   6.05,  17.84,  14.88,   5.73,  11.85,
 8.61,   9.22,  10.71,  14.26,   5.83,  11.72,  12.33,   9.8 ,
11.25,  10.96,   6.84,  10.12,  10.81,   9.37,   7.59,   9.35,
 7.79,   7.29,   6.84,   6.83,   9.63,   9.52,   7.58,   7.44,
11.52,   7.22,  22.72,   6.56,   9.74,   9.65,   3.82,   8.79,
 9.76,   7.39,   9.3 ,  10.23,  10.6 ,   7.95,   6.67,   9.38,
 6.47,   6.24,   4.69,   7.12,   8.27,   4.66,   7.77,   4.9 ,
12.43,   5.19,   7.46,   6.89,   5.43,   6.63,   4.96,   4.13,
 4.01,   5.7 ,   8.1 ,   4.01,   4.68,   8.29,   4.85,   6.31,
 5.35,   4.34,   6.25,   5.1 ])
```

In [64]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

In [65]:
```
print('train',len(x_train))
print('test',len(x_test))
```

```
train 432
test 108
```

In [66]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[66]:
```
LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [67]:
```
pre=lr.predict(x_test)
pre
```

Out[67]:
```
array([22.44273391, 28.93255449, 21.02092615, 17.41165782, 25.03628039,
       16.43740387, 50.30042626, 46.46163641, 21.48668067, 25.36940724,
        9.51976101, 21.96485847, 37.19864048, 41.89796391, 33.61921415,
       23.10139549, 19.24996566, 29.69983918, 21.3853473 ,  9.64216387,
       27.77570791, 53.80339637, 78.71471218, 12.4818598 , 11.84223134,
       36.9323983 , 24.69804269, 14.64406423, 46.93742952,  7.57240939,
       26.45330073, 28.71780417,  8.61941042, 12.96153422, 26.05600162,
       26.90497208, 35.19698229, 12.66693315, 37.33634448, 29.32491286,
       33.59703958, 27.76250381, 40.7301611 , 42.08840744, 32.69608216,
        6.82500769, 28.57361556, 10.23753372, 15.09344694, 11.56763657,
       12.30073494, 29.46957713, 42.07293157, 28.38342547,  3.8185151 ,
       12.39636345, 32.47236417, 36.1838549 ,  7.20842733, 20.09390548,
       29.00582783, 24.93640528, 71.45692178, 18.19214669, 14.76094181,
       20.60163908, 14.04651196, 33.41725457,  9.62882999, 40.02875988,
       21.60890051, 29.47320785, 24.97034386, 28.8931641 , 29.32024909,
       36.56622061, 22.21499457,  7.29867764, 40.61818068,  9.29263086,
       27.24731733, 11.34054072, 14.43467857, 15.59755271, 29.97638531,
       21.60057527, 52.31268056, 29.42540381, 43.57969709, 28.93761697,
       81.81294876,  8.27020731, 24.94412786, 30.86906652,  5.82854307,
       49.27961943, 76.6647087 , 68.43872875, 28.9719621 , 12.00203198,
        8.30415786,  6.73169125, 58.22442488,  6.45984831, 64.0043829 ,
       30.31401156, 23.2471538 , 34.91077286])
```

In [68]:
```
import sklearn.metrics as mc
mc.r2_score(y_test,pre)
```

Out[68]:
```
0.9999998023256144
```

In [69]:
```
mc.mean_absolute_error(y_test,pre)
```

Out[69]:
```
0.006008972279509756
```

In [70]:
```
mc.mean_squared_error(y_test,pre)
```

Out[70]:
```
5.351183372947178e-05
```