In [7]:
```
import seaborn as sns
```

In [9]:
```
df=sns.load_dataset('healthexp')
```

In [11]:
```
df.head(2)
```

Out[11]:

	Year	Country	Spending_USD	Life_Expectancy
0	1970	Germany	252.311	70.6
1	1970	France	192.143	72.2

In [28]:
```
df['Country'].unique()
```

Out[28]:
```
array(['Germany', 'France', 'Great Britain', 'Japan', 'USA', 'Canada'],
      dtype=object)
```

In [44]:
```
x=df.drop(columns=['Life_Expectancy','Country'],axis=1)
y=df['Life_Expectancy']
```

In [45]:
```
x
```

Out[45]:

	Year	Spending_USD
0	1970	252.311
1	1970	192.143
2	1970	123.993
3	1970	150.437
4	1970	326.961
...	...	...
269	2020	6938.983
270	2020	5468.418
271	2020	5018.700
272	2020	4665.641
273	2020	11859.179

274 rows × 2 columns

In [46]:
```
y
```

Out[46]:
```
0      70.6
1      72.2
2      71.9
3      72.0
4      70.9
       ...
269    81.1
270    82.3
271    80.4
272    84.7
273    77.0
Name: Life_Expectancy, Length: 274, dtype: float64
```

In [50]:
```
from sklearn.model_selection import train_test_split
```

In [53]:
```
x_train,x_test,y_train,y_test=train_test_split(x, y, test_size=0.33, random_state=4
```

In [54]:
```
x_train
```

Out[54]:

	Year	Spending_USD
250	2017	5150.470
78	1987	1976.166
185	2006	3567.061
266	2019	4610.794
234	2014	4626.679
...	...	...
188	2006	2561.219
71	1986	578.610
106	1993	1930.889
270	2020	5468.418
102	1992	1651.139

183 rows × 2 columns

In [55]:
```
from sklearn.linear_model import LinearRegression
```

In [56]:
```
regressor=LinearRegression()
```

In [57]:
```
regressor.fit(x_train,y_train)
```

Out[57]:
```
▾ LinearRegression
LinearRegression()
```

In [58]:
```
regressor.intercept_
```

Out[58]:
```
-538.111768851428
```

In [59]:
```
regressor.coef_
```

Out[59]:
```
array([ 0.30962587, -0.00080345])
```

In [65]:
```
y_pred_test=regressor.predict(x_test)
```

In [66]:
```
y_pred_test
```

Out[66]:
```
array([73.74212291, 80.0999976 , 81.1461463 , 77.90874396, 83.4993729 ,
       81.8296441 , 83.347026  , 77.68349543, 78.3317948 , 81.20214948,
       76.25038382, 79.17104879, 81.60683641, 81.87442744, 79.031728  ,
       79.25841831, 81.03963672, 75.76694309, 76.46500759, 78.3032554 ,
       74.9084194 , 75.32062112, 80.04322208, 81.7036527 , 83.05411457,
       80.89748008, 78.43162328, 74.11484225, 71.87319784, 77.82231885,
       72.9487926 , 76.52470292, 73.23403543, 77.99704419, 78.15842828,
       75.92565634, 82.21835435, 78.32003751, 71.92119341, 75.10147715,
       74.77182682, 78.29475172, 76.05400332, 73.46492469, 77.70569516,
       80.80855093, 72.97150527, 81.75737824, 78.41410744, 82.85319342,
       77.13976652, 79.65537287, 79.78873751, 72.42711801, 80.49352817,
       72.19939402, 82.4482505 , 77.90386511, 74.0816274 , 79.65356555,
       79.49025837, 77.16879219, 74.31127565, 78.45907111, 72.72594393,
       78.28636111, 82.06015043, 75.83805739, 81.04268384, 77.04055689,
       76.65830303, 80.87959841, 80.2299662 , 76.37831332, 72.69190184,
       83.30022577, 81.36796977, 77.41492608, 76.64571961, 76.33201329,
       75.4730632 , 78.17257607, 81.47099653, 74.21779126, 79.88644563,
       81.7372828 , 74.70977072, 77.67619643, 79.61330044, 77.90566148,
       74.01278847])
```

In [71]:
```
import numpy as np
```

In [72]:
```
#performance metrix
from sklearn.metrics import mean_squared_error,mean_absolute_error
```

In [73]:
```
mse=mean_squared_error(y_test,y_pred_test)
mae=mean_absolute_error(y_test,y_pred_test)
```

In [74]:
```
rmse=np.sqrt(mse)
```

In [75]:
```
print(mse)
print(mae)
print(rmse)
```

```
1.2844217000106994
0.9537321916395342
1.133323298979907
```

In [76]:
```
#check accuracy of the model

from sklearn.metrics import r2_score
```

In [77]:
```
score=r2_score(y_test,y_pred_test)
```

In [79]:
```
score
```

Out[79]:
```
0.8841615461695458
```

In [ ]:
```

```