In [1]:
```python
import pandas as pd
import pickle
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```python
a=pd.read_csv("C:\\Users\\reshma_koduri\\OneDrive\\Documents\\fiat500 crt.csv")
```

In [3]:
```python
a
```

Out[3]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	pric
0	1	lounge	51	882	25000	1	44.907242	8.611560	890
1	2	pop	51	1186	32500	1	45.666359	12.241890	880
2	3	sport	74	4658	142228	1	45.503300	11.417840	420
3	4	lounge	51	2739	160000	1	40.633171	17.634609	600
4	5	pop	73	3074	106880	1	41.903221	12.495650	570
...	...	...	...	...	...	...	...	...	.
1533	1534	sport	51	3712	115280	1	45.069679	7.704920	520
1534	1535	lounge	74	3835	112000	1	45.845692	8.666870	460
1535	1536	pop	51	2223	60457	1	45.481541	9.413480	750
1536	1537	lounge	51	2557	80750	1	45.000702	7.682270	599
1537	1538	pop	51	1766	54276	1	40.323410	17.568270	790

1538 rows × 9 columns

In [4]:
```python
a.head()
```

Out[4]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price
0	1	lounge	51	882	25000	1	44.907242	8.611560	8900
1	2	pop	51	1186	32500	1	45.666359	12.241890	8800
2	3	sport	74	4658	142228	1	45.503300	11.417840	4200
3	4	lounge	51	2739	160000	1	40.633171	17.634609	6000
4	5	pop	73	3074	106880	1	41.903221	12.495650	5700

In [5]:
```python
a.tail()
```

Out[5]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price
1533	1534	sport	51	3712	115280	1	45.069679	7.70492	5200
1534	1535	lounge	74	3835	112000	1	45.845692	8.66687	4600
1535	1536	pop	51	2223	60457	1	45.481541	9.41348	7500

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price
1536	1537	lounge	51	2557	80750	1	45.000702	7.68227	5990
1537	1538	pop	51	1766	54276	1	40.323410	17.56827	7900

In [6]:

```
a.describe()
```

Out[6]:

	ID	engine_power	age_in_days	km	previous_owners	lat	
count	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000	1538.0
mean	769.500000	51.904421	1650.980494	53396.011704	1.123537	43.541361	11.5
std	444.126671	3.988023	1289.522278	40046.830723	0.416423	2.133518	2.3
min	1.000000	51.000000	366.000000	1232.000000	1.000000	36.855839	7.2
25%	385.250000	51.000000	670.000000	20006.250000	1.000000	41.802990	9.5
50%	769.500000	51.000000	1035.000000	39031.000000	1.000000	44.394096	11.8
75%	1153.750000	51.000000	2616.000000	79667.750000	1.000000	45.467960	12.7
max	1538.000000	77.000000	4658.000000	235000.000000	4.000000	46.795612	18.3

In [7]:

```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1538 entries, 0 to 1537
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   ID               1538 non-null   int64
 1   model            1538 non-null   object
 2   engine_power     1538 non-null   int64
 3   age_in_days      1538 non-null   int64
 4   km               1538 non-null   int64
 5   previous_owners  1538 non-null   int64
 6   lat              1538 non-null   float64
 7   lon              1538 non-null   float64
 8   price            1538 non-null   int64
dtypes: float64(2), int64(6), object(1)
memory usage: 108.3+ KB
```

In [8]:

```
a.shape
```

Out[8]:

```
(1538, 9)
```

In [9]:

```
b=a.drop(["ID","lat","lon"],axis=1)
b
```

Out[9]:

	model	engine_power	age_in_days	km	previous_owners	price
0	lounge	51	882	25000	1	8900
1	pop	51	1186	32500	1	8800

	model	engine_power	age_in_days	km	previous_owners	price
2	sport	74	4658	142228	1	4200
3	lounge	51	2739	160000	1	6000
4	pop	73	3074	106880	1	5700
...	...	...	...	...	...	...
1533	sport	51	3712	115280	1	5200
1534	lounge	74	3835	112000	1	4600
1535	pop	51	2223	60457	1	7500
1536	lounge	51	2557	80750	1	5990
1537	pop	51	1766	54276	1	7900

1538 rows × 6 columns

In [10]:

```
c=pd.get_dummies(b,dtype=int)
c
```

Out[10]:

	engine_power	age_in_days	km	previous_owners	price	model_lounge	model_pop	model
0	51	882	25000	1	8900	1	0	
1	51	1186	32500	1	8800	0	1	
2	74	4658	142228	1	4200	0	0	
3	51	2739	160000	1	6000	1	0	
4	73	3074	106880	1	5700	0	1	
...	...	...	...	...	...	...	...	
1533	51	3712	115280	1	5200	0	0	
1534	74	3835	112000	1	4600	1	0	
1535	51	2223	60457	1	7500	0	1	
1536	51	2557	80750	1	5990	1	0	
1537	51	1766	54276	1	7900	0	1	

1538 rows × 8 columns

In [11]:

```
y=c['price']
y
```

Out[11]:

```
0       8900
1       8800
2       4200
3       6000
4       5700
        ...
1533    5200
1534    4600
1535    7500
1536    5990
```

```
1537    7900
Name: price, Length: 1538, dtype: int64
```

In [12]:
```
x=c.drop(['price'],axis=1)
x
```

Out[12]:

	engine_power	age_in_days	km	previous_owners	model_lounge	model_pop	model_sport
0	51	882	25000	1	1	0	0
1	51	1186	32500	1	0	1	0
2	74	4658	142228	1	0	0	1
3	51	2739	160000	1	1	0	0
4	73	3074	106880	1	0	1	0
...	...	...	...	...	...	...	...
1533	51	3712	115280	1	0	0	1
1534	74	3835	112000	1	1	0	0
1535	51	2223	60457	1	0	1	0
1536	51	2557	80750	1	1	0	0
1537	51	1766	54276	1	0	1	0

1538 rows × 7 columns

linear regression

In [13]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=55)
```

In [14]:
```
from sklearn.linear_model import LinearRegression
reg=LinearRegression()
reg.fit(x_train,y_train)
```

Out[14]:
```
LinearRegression()
```

In [15]:
```
ypred=reg.predict(x_test)
ypred
```

Out[15]:
```
array([ 6465.84412726,  8133.50478681,  7956.46087553,  5535.74773992,
       10378.00092676, 10503.05910467,  7532.57999941, 10375.7831635 ,
       10468.0801048 ,  5647.97023737,  7378.85984909,  9694.683737  ,
       10360.43915959,  6668.7026613 ,  9791.61611077,  6831.58592236,
       10363.78468208, 10451.25770356,  9809.83768325,  7953.48286989,
       10061.76521842, 10267.86475424,  9376.22409925, 10009.90885599,
        6831.25395007, 10394.7367039 , 10381.91615015, 10443.89484079,
       10136.01749567,  9700.83404847,  9838.68301684, 10469.81258691,
        9206.33323998,  9900.54470231,  7541.97878901,  9978.86277667,
        8897.38549375,  9977.68207432,  6517.42310638,  6839.88149322,
        9815.72927481,  7380.91254223,  9895.44077646,  9827.06493056,
       10085.16343208,  9656.13467159,  9738.41457218,  8411.08415076,
```

```
       10080.54513826,  9537.3525905 ,  5796.38467904,  9973.24129729,
        8289.01122572,  3318.65039475, 10137.49201122,  9931.72212476,
       10141.74794349, 10105.73262738,  9760.17156056,  4604.48294988,
        9906.87515835,  8368.01733388,  9634.82648027,  7080.65545083,
        4996.31646768,  7365.00741941, 10511.96489132,  6616.74051791,
        6926.04238014,  5666.87255667, 10411.92274021,  9906.23413997,
        9831.73740946,  9752.83622131,  8118.5329684 ,  9637.64560722,
        9222.59581028,  6559.33943026,  8114.56203276, 10156.1333325 ,
        9198.43166815,  9745.0945197 ,  9963.23597425,  9742.60351482,
        6721.62939328,  5506.64502741,  9972.60654421, 10403.31992372,
        5501.52551503, 10470.98324134, 10375.43666708,  5524.33684273,
        9344.37345859, 10477.00257381, 10332.17658892,  9834.46039322,
        9401.09936925,  6092.12926   ,  7267.98177687, 10485.75142227,
        9334.26905777,  6556.89304732,  8385.33521256, 10027.39086769,
        9405.12362035,  9432.61829752, 10087.64551421,  5116.64181136,
        9487.56471886,  9921.98538916, 10222.60867087,  6732.83162132,
        3708.32189122,  6528.14340126, 10000.4495037 , 10422.84499704,
       10023.3732288 ,  5117.05300207, 10409.24519869,  8481.13490293,
        7008.90869057,  9965.66349896,  7641.97287503,  9656.13467159,
        5615.63383078,  8446.35431339,  7718.58035929, 10443.58280784,
        9926.96968712,  5648.39671324,  9977.49728883,  8879.57218696,
       10383.12888763,  6226.51509377,  9941.94656987,  9253.6480772 ,
        8877.51575604,  9694.6185794 ,  9881.16811077, 10321.6257729 ,
        8877.16164005,  9110.40619686,  9879.74766161,  7030.16018898,
        6409.42035813,  4616.7709324 , 10314.42626691,  9359.55190759,
        9975.38347449,  9927.75010803,  7869.45228741,  8034.58285931,
        5431.21931832, 10458.91527447,  8854.49643924,  7877.44031379,
        8852.51297153,  9138.1755682 ,  4543.7730655 ,  7693.67014104,
        5625.54710873,  9677.85688921,  6811.48178609,  6672.72413247,
        7325.20327026,  9725.03402279,  7831.77484005, 10031.3888334 ,
        5477.19563475, 10485.75142227,  9537.18329781,  7322.01797142,
       10405.08705547,  9859.38782385,  5491.39821623, 10398.48629865,
        8664.31680592,  8407.38594041,  9617.39326375,  8653.48109946,
        5964.31095338,  5929.85905593,  9440.19021298,  9879.74766161,
        8405.59965181,  9860.901057  , 10409.54923973,  9881.77149265,
        9768.42134841,  5758.9582845 ,  8603.23090877,  5445.43858216,
       10325.03876265,  9929.24004264,  9264.89558168,  9820.91932554,
        9835.93300301,  9776.12793848, 10107.43154094, 10461.21966184,
        9821.61971397,  7608.263182  ,  9704.017369  ,  5275.53898388,
       10308.15303584,  8945.40107027, 10444.74375702,  6658.29705694,
        8069.23250141, 10421.36476768,  7511.08194624, 10063.96547069,
       10367.7446327 ,  6473.88601726,  6565.0968456 , 10446.12955653,
        6852.22064045,  9681.19068825, 10172.76217385,  8764.67273805,
        6325.28647528,  5792.03228428,  5338.21787921, 10007.78088986,
        9188.89648435,  4280.67102076,  8561.24352821,  6081.00460391,
        6731.13515647, 10433.88090805, 10486.34691766,  5834.93610149,
       10342.3635837 ,  9865.60517096, 10174.14570972, 10091.61026224,
        6200.85865303,  7669.28228284,  9175.5777271 ,  9726.61860497,
        9845.74629333,  4916.9404613 ,  6303.93740313,  9335.04001276,
        8315.79763499,  9954.62255133, 10111.50477965,  9854.38167378,
        5470.75741199,  7181.05657963,  7485.27841906, 10082.05067811,
        5987.76338317,  9481.22375348,  6617.40397146,  7793.98709568,
        7761.64165478,  9436.6152524 ,  5394.31293749,  7696.78554824,
        8728.47857236,  9728.97254471,  9977.68207432,  9383.7745482 ,
        8578.55218552, 10391.92037764,  5862.54583937, 10160.58021014,
        9830.31811513,  7217.84732793,  8011.58905243, 10045.06110407,
       10466.55552055, 10323.84353616,  7755.32059093,  5786.9144475 ,
        9814.12422634,  8926.59118653, 10521.85653551,  5508.36892129,
       10102.84236912,  5903.58662085, 10414.89290418,  7814.64159022,
        5764.03790711, 10416.88544477, 10484.98369347, 10344.75459517,
        8093.11987802,  9710.53150171,  9998.89344283,  5959.16838456,
       10067.83861103, 10213.17519382,  8496.20081041, 10402.23835141,
        8039.06344948,  5172.04235643,  9738.6670111 ,  8207.98005082,
        8358.31050992,  9796.72394612, 10437.25924815,  5656.69181141,
```

```
        4402.71552775, 10389.97999768,  9164.46926304,  7664.30139634,
        8019.0849124 ,  9733.26148076,  8774.43385328,  9341.96994118,
        8346.34221399,  6182.49477324,  7179.88588739,  9739.30802947,
        3992.44498845,  7860.67439985, 10323.10280151,  8195.23215555,
        6240.38316025,  8888.96223997,  6733.04029047, 10141.74794349,
       10278.46964367,  9381.08693729,  9289.74360882,  5988.37407734,
        6733.36949363,  4717.22521833,  8007.6298925 ,  5947.47973885,
       10047.04882578,  8886.55408984, 10116.55988961,  9680.47738373,
       10438.56831498,  6788.43808518,  9914.39730371,  9182.44586337,
       10329.7166505 ,  9970.81961316,  9326.22041158,  4688.66815726,
        8120.45919656, 10050.86790598,  9652.04290585,  7266.55546644,
        9814.12422634,  7137.46811236, 10428.64417938, 10030.30315722,
       10167.21949838, 10134.42533173,  5453.64302486,  7663.02627129,
        8659.09393133,  9949.04694581, 10095.91244071,  7348.97154593,
        9790.64147424,  5436.23340305,  8818.33690172,  9977.68207432,
        7734.84034279,  9954.46942863,  6097.76695296,  9956.69590702,
       10363.32661717,  6463.09056181, 10187.8869288 ,  9827.06493056,
        6980.64192646,  9869.21417042, 10266.61965224,  9836.72549824,
       10038.24781668,  7265.13576821, 10547.84376708, 10322.49201395,
        9506.91773036,  9794.35343249,  9669.81817225,  8814.36868907,
        7055.82578764,  5667.35171576,  6957.85093775,  4569.99086943,
        7624.3138036 ,  8080.75925714,  6952.81901478,  6322.24167244,
        4948.593497  ,  4819.06952197, 10072.27821177, 10300.64560078,
        8544.02586692, 10007.50197313, 10313.9883275 ,  9326.27152267,
        9768.42134841,  8004.93689085,  9253.30718215,  7328.56122088,
       10382.59200434,  7386.65758355,  6960.15263687, 10083.65598648,
        9890.77893991, 10560.87203251,  9945.9111532 ,  7066.31798543,
        8835.47946785,  6848.48596676, 10462.47656803,  5090.74634196,
        5456.74857413, 10442.47847404,  5797.4469298 ,  8867.35915678,
        9931.28881807,  8704.23617435, 10561.39177714, 10066.46024492,
       10189.46647437,  9631.35669718, 10092.672513  ,  9458.50937966,
        6589.10512111,  9894.83758075,  7602.96977867,  7111.04902753,
        9136.09445993,  3976.76113097,  4543.68569047, 10025.31080807,
       10427.94838585, 10019.81402092,  9905.33725094,  9977.68207432,
        9124.63353185,  6059.03026114,  9879.74766161,  7293.99711961,
        8652.91360293, 10352.87993624,  6957.25360421,  6529.10705277,
       10134.89328807,  9881.3096962 ,  9479.11621146,  8314.73538423,
       10094.80047913, 10259.23927847,  9758.2613837 ,  7393.98143101,
        9912.91796928,  6617.40397146,  6937.95488692,  8403.77863984,
       10447.473273  ,  7219.25388609, 10494.80705318,  7162.61979701,
        9948.65146179,  6540.63315915, 10075.34769195,  9738.41457218,
        7275.65860991, 10061.48783511,  6052.23318399,  9083.99278713,
        9279.752627  ,  6906.63678132,  9521.69095227, 10370.81855943,
        4047.95087042,  9941.5463794 , 10159.18250101,  9781.51574009,
        9880.56491506,  6709.98020272, 10426.57001974,  7370.24941303,
        9269.43072927,  6266.01627583,  7803.56212035,  9316.1878805 ,
        7336.83763272,  9922.62342069, 10096.24489075,  8308.86154669,
        9559.56767536,  9400.26863406, 10136.01749567,  5414.89094311,
        6220.05931313, 10147.83474322,  7053.93170704, 10273.98251501,
        9925.53716365, 10351.77376221,  9762.43852916, 10066.96247856,
        9975.55672271,  9582.03535375, 10485.75142227, 10034.54583262])
```

In [16]:

```
from sklearn.metrics import r2_score
r2_score(y_test,ypred)
```

Out[16]:

```
0.8408716955150235
```

In [17]:

```
from sklearn.metrics import mean_squared_error
mean_squared_error(ypred,y_test)
```

Out[17]:

```
598312.6697577912
```

In [18]:

```
results=pd.DataFrame(columns=['price','predicted'])
results['price']=y_test
results['predicted']=ypred
results=results.reset_index()
results['ID']=results.index
results.head(5)
```

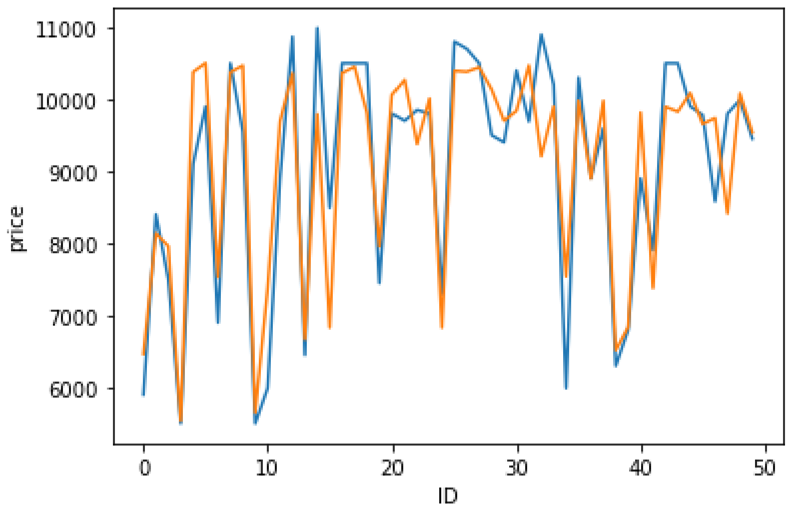
Out[18]:

	index	price	predicted	ID
0	448	5900	6465.844127	0
1	1190	8400	8133.504787	1
2	1428	7500	7956.460876	2
3	600	5500	5535.747740	3
4	323	9100	10378.000927	4

In [19]:

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.lineplot(x='ID',y='price',data=results.head(50))
sns.lineplot(x='ID',y='predicted',data=results.head(50))
plt.plot()
```

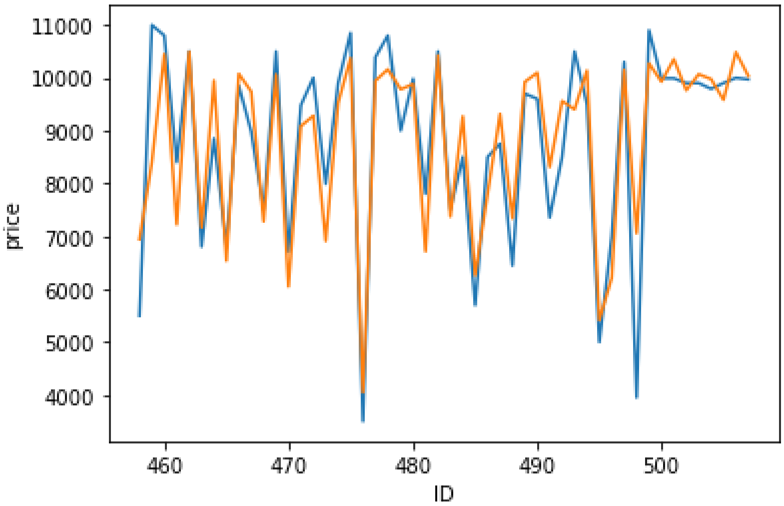
Out[19]: []



In [20]:

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.lineplot(x='ID',y='price',data=results.tail(50))
sns.lineplot(x='ID',y='predicted',data=results.tail(50))
plt.plot()
```

Out[20]: []



ridge regression

In [21]:
```
new=[[51,2197,70000,1,1,0,0]]
real=reg.predict(new)
real
```

Out[21]:
```
array([7884.39470064])
```

In [22]:
```
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import Ridge
alpha = [1e-15, 1e-10, 1e-8, 1e-4, 1e-3,1e-2, 1, 5, 10, 20,30]
ridge=Ridge()
parameters={'alpha':alpha}
regressor=GridSearchCV(ridge,parameters)
regressor.fit(x_train,y_train)
```

Out[22]:
```
GridSearchCV(estimator=Ridge(),
             param_grid={'alpha': [1e-15, 1e-10, 1e-08, 0.0001, 0.001, 0.01, 1,
                                   5, 10, 20, 30]})
```

In [23]:
```
regressor.best_params_
```

Out[23]:
```
{'alpha': 30}
```

In [24]:
```
ridge=Ridge(30)
ridge.fit(x_train,y_train)
y_pred=ridge.predict(x_test)
```

In [25]:
```
y_pred
```

Out[25]:
```
array([ 6460.04576022,  8128.10945264,  7951.38801966,  5529.91706546,
       10373.05027407, 10498.11720582,  7527.10638452, 10370.81904397,
       10463.12931468,  5668.78483233,  7373.38059458,  9689.7058629 ,
       10355.40918901,  6662.99772943,  9786.62469875,  6852.4920757 ,
       10358.84845823, 10446.30926291,  9794.05003032,  7974.82739411,
       10056.72274423, 10255.76562376,  9397.70811094, 10004.88695642,
        6852.20103509, 10389.78371382, 10376.95117407, 10438.96120503,
       10157.55759992,  9695.8553154 ,  9833.63170262, 10464.86155482,
        9227.65255129,  9891.97103462,  7536.61274081,  9973.84521311,
        8918.67995976,  9972.71933754,  6511.5904269 ,  6834.49084653,
```

```
        9810.7616038 ,  7375.51432724,  9890.46197265,  9848.51308494,
       10080.09147009,  9640.32760006,  9733.40348174,  8432.21588888,
       10075.48626464,  9532.35713774,  5787.23409153,  9964.73791394,
        8283.71549567,  3339.34368331, 10132.43896077,  9926.7249216 ,
       10136.74851611, 10100.71112129,  9755.1845402 ,  4590.04707113,
        9901.89475758,  8362.41326225,  9629.85696605,  7071.83878585,
        4981.7204652 ,  7386.06927526, 10496.42550144,  6610.96196146,
        6920.31483945,  5650.28453471, 10406.9535733 ,  9901.25382872,
        9826.76750269,  9737.09741472,  8113.15305741,  9632.60859403,
        9240.38536008,  6553.73109867,  8135.69518773, 10151.06390196,
        9193.15868985,  9740.07005296,  9958.23436976,  9737.40341138,
        6716.07972191,  5500.79130881,  9957.03449251, 10398.37817823,
        5483.43989797, 10466.01960239, 10370.47259594,  5518.46732105,
        9365.83569873, 10472.06431411, 10327.21855964,  9829.46299741,
        9422.59323894,  6075.56228257,  7262.36913212, 10480.79816411,
        9328.9899957 ,  6551.47481857,  8379.91803901, 10048.94614312,
        9399.99687049,  9427.50148424, 10082.62653423,  5133.89468839,
        9482.76464761,  9916.9757693 , 10217.52995614,  6728.39933193,
        3705.15728317,  6522.48520915,  9995.42892525, 10417.90052461,
       10018.32234001,  5111.34879808, 10404.31640223,  8475.92121956,
        6992.89005767,  9987.09229616,  7636.32177392,  9640.32760006,
        5609.684453  ,  8467.49445772,  7739.64660695, 10438.63543909,
        9918.44565749,  5642.52319708,  9972.46658381,  8874.54594624,
       10378.16374216,  6220.91236031,  9936.90705342,  9248.4891543 ,
        8898.9072111 ,  9689.49228295,  9876.1641916 , 10316.66921719,
        8898.52692464,  9131.9406658 ,  9874.7577174 ,  7024.75674385,
        6403.61653927,  4637.25625644, 10309.49693664,  9354.41907909,
        9970.33928812,  9922.70013095,  7866.25039858,  8029.16045561,
        5454.28785756, 10453.96576433,  8842.27028881,  7861.44494319,
        8873.55683819,  9132.88345234,  4537.86450829,  7688.0118197 ,
        5608.82887034,  9672.84181312,  6795.28763321,  6657.49232091,
        7319.44709006,  9720.09279517,  7826.1923333 , 10026.40481917,
        5460.36301636, 10480.79816411,  9532.02566067,  7332.27315862,
       10400.14506318,  9854.41405533,  5485.68172633, 10393.54522824,
        8658.8606688 ,  8402.14167177,  9612.38663222,  8674.6874185 ,
        5954.56807357,  5920.52761887,  9434.93724289,  9874.7577174 ,
        8400.26141868,  9855.87285966, 10404.5804043 ,  9876.79459794,
        9763.41984368,  5749.5418367 ,  8598.05394611,  5425.35399625,
       10320.08173026,  9924.18985747,  9286.24626605,  9815.73450416,
        9830.93540153,  9797.47521635, 10128.9756376 , 10456.28360644,
        9816.65122027,  7602.86032315,  9698.99863924,  5266.12294692,
       10303.1441443 ,  8936.83814544, 10439.8100027 ,  6652.95887972,
        8063.80525842, 10416.39428206,  7532.04207459, 10058.92268921,
       10362.79541243,  6468.07319489,  6560.5395512 , 10441.1818321 ,
        6846.28787327,  9676.18758991, 10167.7175296 ,  8759.4056782 ,
        6319.81926732,  5812.78631811,  5318.09360624, 10002.73217875,
        9183.63772553,  4264.01884547,  8582.65293506,  6103.09409176,
        6725.46194596, 10428.93489431, 10474.3385959 ,  5852.29130789,
       10337.40413166,  9857.08971172, 10165.54627329, 10083.02235301,
        6196.47355216,  7690.43674186,  9197.06354186,  9748.05367931,
        9837.18027898,  4900.46962091,  6325.09323436,  9356.5568851 ,
        8336.94268117,  9949.64925856, 10106.45535866,  9845.84067332,
        5465.23312162,  7175.67294508,  7479.74430525, 10103.59831955,
        5971.33230457,  9476.04637871,  6611.3417916 ,  7788.65251193,
        7756.31158851,  9458.10416181,  5414.90564832,  7717.84106327,
        8723.2712294 ,  9723.96277297,  9972.71933754,  9405.27083753,
        8573.31111993, 10386.99400083,  5856.66952312, 10155.59192915,
        9825.24086045,  7212.05192397,  8006.34584473, 10040.04807158,
       10461.60494335, 10318.90044728,  7749.76253863,  5770.01146394,
        9809.11633858,  8948.04378159, 10516.91201135,  5502.62384127,
       10097.79415796,  5924.31129803, 10409.94954237,  7837.85098846,
        5784.36299973, 10411.95558125, 10476.50305232, 10339.80858577,
        8114.20181019,  9705.51186216,  9993.85974954,  5953.38656421,
       10062.76906869, 10208.19289939,  8517.30688426, 10397.28431363,
```

```
        8060.18004011,  5166.35802556,  9733.64344204,  8202.72274661,
        8353.00510129,  9791.75892946, 10432.31276258,  5652.31471679,
        4388.36508214, 10385.05389187,  9178.92786874,  7658.66050891,
        8013.90865185,  9728.22489009,  8795.69082928,  9363.48584567,
        8369.95316592,  6203.32892855,  7174.71884185,  9734.28437089,
        4011.72702212,  7881.8692336 , 10341.08928653,  8189.9632996 ,
        6261.16878976,  8883.9346878 ,  6727.52946631, 10136.74851611,
       10273.51911529,  9399.04277961,  9312.25021025,  5982.75038603,
        6754.18137369,  4711.45464419,  7998.67131959,  5944.05661945,
       10041.98218701,  8881.52687401, 10138.00849739,  9675.50149375,
       10433.59542665,  6771.58454198,  9909.4025202 ,  9177.25599952,
       10324.77274136,  9965.76362084,  9321.1588991 ,  4709.29231574,
        8115.06568438, 10045.82695375,  9647.03143503,  7287.71195426,
        9809.11633858,  7131.94168967, 10423.64556835, 10025.25130057,
       10188.75524484, 10129.358933  ,  5436.74614627,  7684.4655792 ,
        8680.47589547,  9944.04732301, 10090.8651974 ,  7343.52357446,
        9785.61064625,  5456.90202974,  8802.54742594,  9972.71933754,
        7729.52779626,  9949.45527193,  6091.96577543,  9948.18105788,
       10358.36423736,  6483.85841937, 10182.85394874,  9848.51308494,
        7001.73035545,  9864.22569734, 10261.68455545,  9831.68823398,
       10033.1824071 ,  7259.83460483, 10542.89561345, 10317.53533726,
        9528.31638367,  9789.36163817,  9664.80421887,  8809.17601472,
        7050.06678877,  5650.6138622 ,  6952.13273114,  4587.09078042,
        7645.27144974,  8075.46639238,  6947.43970357,  6316.4362527 ,
        4969.42444321,  4815.43825462, 10067.24760147, 10295.7057518 ,
        8565.22036414, 10029.06002631, 10308.99195315,  9320.9806898 ,
        9763.41984368,  8000.66557169,  9248.06653624,  7351.02805252,
       10377.64071044,  7380.87948825,  6981.18970999, 10078.57045846,
        9885.73412636, 10555.9220593 ,  9940.91196835,  7087.46155383,
        8830.21629545,  6844.07796557, 10457.50034059,  5084.81560997,
        5451.03692353, 10433.99132596,  5791.82368414,  8862.19952249,
        9926.27789885,  8718.610182  , 10556.44173135, 10061.41711501,
       10184.40616497,  9626.32056232, 10087.61194562,  9479.95567917,
        6572.63094227,  9889.84552902,  7597.62143219,  7132.10591122,
        9157.58578907,  3970.8371063 ,  4564.24877151, 10020.30053398,
       10422.99075731, 10014.73740929,  9889.74748557,  9972.71933754,
        9119.28908984,  6053.51891326,  9874.7577174 ,  7314.92046157,
        8674.3497589 , 10347.93279203,  6978.35907627,  6549.98834616,
       10129.84060055,  9876.27864851,  9500.58585287,  8332.35308855,
       10089.76672329, 10254.30521245,  9753.23507803,  7388.5271733 ,
        9934.4492359 ,  6611.3417916 ,  6955.31869418,  8425.0336098 ,
       10442.49914097,  7240.34987498, 10486.33881663,  7157.25206957,
        9943.62478548,  6531.8656957 , 10070.28954422,  9733.40348174,
        7269.77469442, 10056.4316231 ,  6035.64466984,  9105.3416283 ,
        9274.75226627,  6898.70650053,  9516.69768688, 10365.8813532 ,
        4068.44764387,  9963.11364417, 10154.11264461,  9776.52573873,
        9875.54774796,  6730.94422527, 10421.63880363,  7364.77136108,
        9290.94298828,  6286.7721054 ,  7798.30796994,  9311.06110838,
        7357.93097601,  9917.64082045, 10084.15595275,  8330.04800249,
        9580.94519882,  9394.92658117, 10157.55759992,  5408.95582523,
        6240.75356717, 10142.76647169,  7074.94190606, 10269.03261333,
        9920.5408243 , 10346.77211061,  9783.82860426, 10061.90550194,
        9970.51251214,  9603.46440117, 10480.79816411, 10056.08633218])
```

In [26]:

```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
```

Out[26]:

```
0.8410484735796884
```

In [27]:

```
results=pd.DataFrame(columns=['actual','predicted'])
results['actual']=y_test
results['predicted']=y_pred
```

```
results=results.reset_index()
results['ID']=results.index
results.head(10)
```

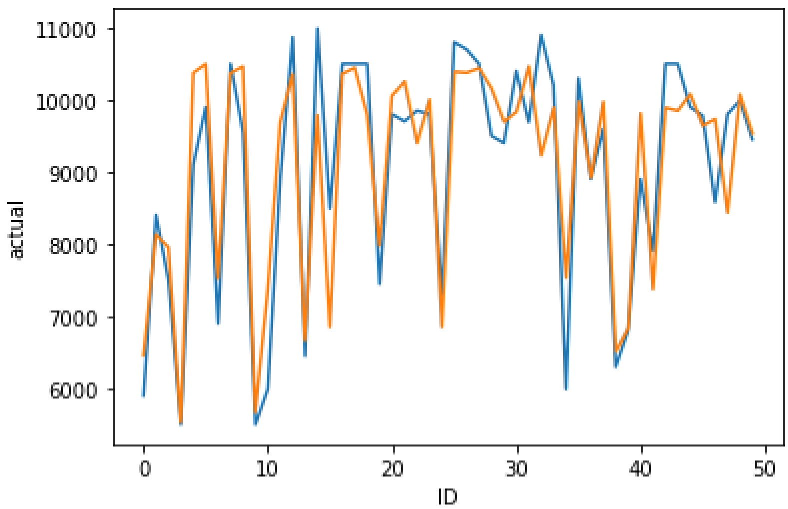
Out[27]:

	index	actual	predicted	ID
0	448	5900	6460.045760	0
1	1190	8400	8128.109453	1
2	1428	7500	7951.388020	2
3	600	5500	5529.917065	3
4	323	9100	10373.050274	4
5	75	9900	10498.117206	5
6	941	6900	7527.106385	6
7	297	10500	10370.819044	7
8	1067	9550	10463.129315	8
9	860	5500	5668.784832	9

In [28]:

```
sns.lineplot(x='ID',y='actual',data=results.head(50))
sns.lineplot(x='ID',y='predicted',data=results.head(50))
plt.plot()
```

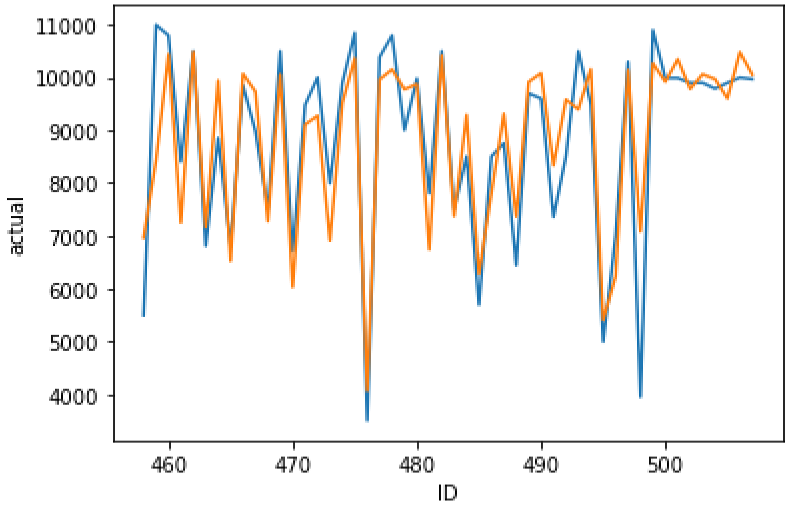
Out[28]: []



In [29]:

```
sns.lineplot(x='ID',y='actual',data=results.tail(50))
sns.lineplot(x='ID',y='predicted',data=results.tail(50))
plt.plot()
```

Out[29]: []



elastic regression

In [30]:
```
from sklearn.linear_model import ElasticNet
elastic=ElasticNet()
parameters = {'alpha': [1e-15, 1e-10, 1e-8, 1e-4, 1e-3,1e-2, 1, 5, 10, 20]}
regg=GridSearchCV(elastic,parameters)
regg.fit(x_train,y_train)
```

Out[30]:
```
GridSearchCV(estimator=ElasticNet(),
             param_grid={'alpha': [1e-15, 1e-10, 1e-08, 0.0001, 0.001, 0.01, 1,
                                   5, 10, 20]})
```

In [31]:
```
regg.best_params_
```

Out[31]:
```
{'alpha': 0.01}
```

In [32]:
```
elastic=ElasticNet(alpha=0.01)
elastic.fit(x_train,y_train)
y_pred_elastic=elastic.predict(x_test)
```

In [33]:
```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred_elastic)
```

Out[33]:
```
0.8409083488550535
```

In [34]:
```
elastic_error=mean_squared_error(y_test,y_pred_elastic)
elastic_error
```

Out[34]:
```
598174.855446253
```

In [35]:
```
results=pd.DataFrame(columns=['actual','predicted'])
results['actual']=y_test
results['predicted']=y_pred_elastic
results=results.reset_index()
results["ID"]=results.index
results.head(5)
```

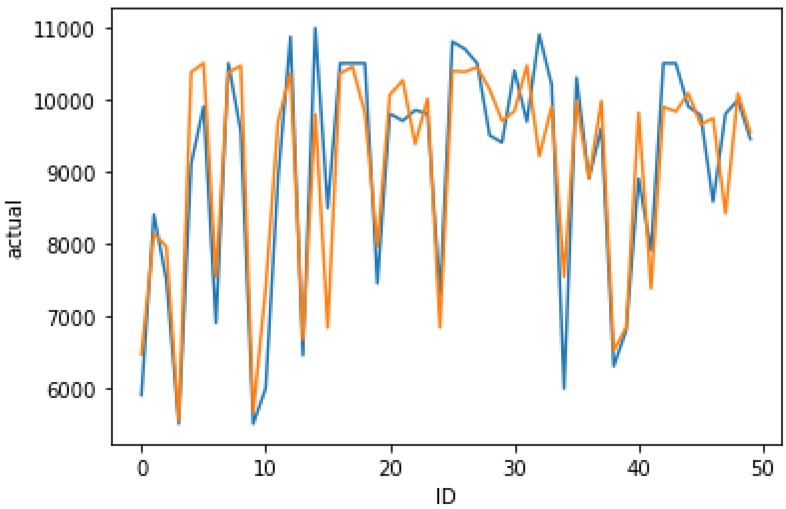
Out[35]:

	index	actual	predicted	ID
0	448	5900	6464.885982	0
1	1190	8400	8132.564609	1
2	1428	7500	7955.612027	2
3	600	5500	5534.828704	3
4	323	9100	10377.059978	4

In [36]:

```
sns.lineplot(x='ID',y='actual',data=results.head(50))
sns.lineplot(x='ID',y='predicted',data=results.head(50))
plt.plot()
```

Out[36]: []



random forest

In [37]:

```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestRegressor
reg=RandomForestRegressor()
n_estimators=[25,50,75,100,125,150,175,200]
criterion=['mse']
max_depth=[3,5,10]
parameters={'n_estimators': n_estimators,'criterion':criterion,'max_depth':max_depth
rfc_reg = GridSearchCV(reg, parameters)
rfc_reg.fit(x_train,y_train)
```

Out[37]:

```
GridSearchCV(estimator=RandomForestRegressor(),
             param_grid={'criterion': ['mse'], 'max_depth': [3, 5, 10],
                         'n_estimators': [25, 50, 75, 100, 125, 150, 175, 200]})
```

In [38]:

```
rfc_reg.best_params_
```

Out[38]:

```
{'criterion': 'mse', 'max_depth': 5, 'n_estimators': 100}
```

In [39]:

```
reg=RandomForestRegressor(n_estimators=125,criterion='mse',max_depth=5)
```

In [40]:

```
reg.fit(x_train,y_train)
```

Out[40]:
```
RandomForestRegressor(max_depth=5, n_estimators=125)
```

In [41]:
```
ypred=reg.predict(x_train)
ypred
```

Out[41]:
```
array([ 9748.64498057,  8889.99983482,  9794.64368513, ...,
        5180.03011865, 10451.7891713 ,  7114.07516633])
```

In [42]:
```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
```

Out[42]:
```
0.8410484735796884
```

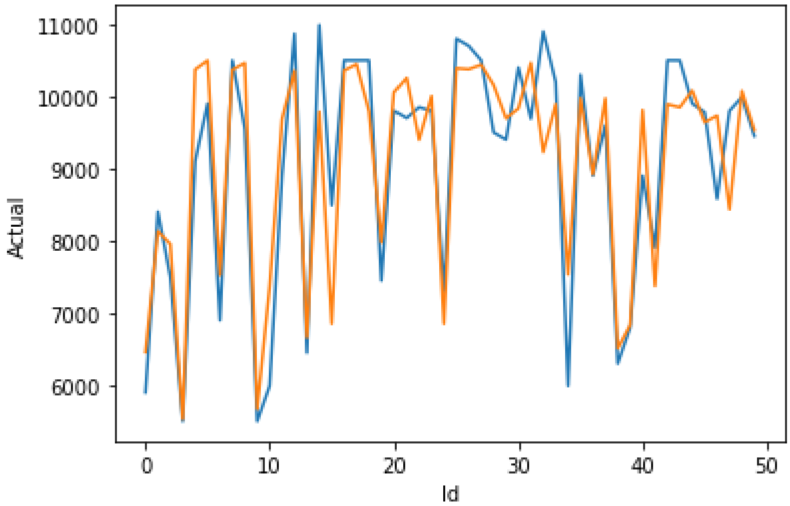
In [43]:
```
Results= pd.DataFrame(columns=['Actual','Predicted'])
Results['Actual']=y_test
Results['Predicted']=y_pred
Results=Results.reset_index()
Results['Id']=Results.index
Results.head(10)
```

Out[43]:

	index	Actual	Predicted	Id
0	448	5900	6460.045760	0
1	1190	8400	8128.109453	1
2	1428	7500	7951.388020	2
3	600	5500	5529.917065	3
4	323	9100	10373.050274	4
5	75	9900	10498.117206	5
6	941	6900	7527.106385	6
7	297	10500	10370.819044	7
8	1067	9550	10463.129315	8
9	860	5500	5668.784832	9

In [44]:
```
sns.lineplot(x='Id',y='Actual',data=Results.head(50))
sns.lineplot(x='Id',y='Predicted',data=Results.head(50))
plt.plot()
```

Out[44]:
```
[]
```



In [ ]: