In [1]:

```
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
```

In [2]:

```
data=pd.read_csv("/home/placement/Downloads/sid.csv")
```

In [3]:

```
list(data)
```

Out[3]:

```
['ID',
 'model',
 'engine_power',
 'age_in_days',
 'km',
 'previous_owners',
 'lat',
 'lon',
 'price']
```

In [4]:

```
data1=data.loc[(data.previous_owners==1)]
```

In [5]:

```
data1
```

Out[5]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lo
0	1	lounge	51	882	25000	1	44.907242	8.61156
1	2	pop	51	1186	32500	1	45.666359	12.24189
2	3	sport	74	4658	142228	1	45.503300	11.41784
3	4	lounge	51	2739	160000	1	40.633171	17.63460
4	5	pop	73	3074	106880	1	41.903221	12.49565
...	...	...	...	...	...	...	...	.
1533	1534	sport	51	3712	115280	1	45.069679	7.70492
1534	1535	lounge	74	3835	112000	1	45.845692	8.66687
1535	1536	pop	51	2223	60457	1	45.481541	9.41348
1536	1537	lounge	51	2557	80750	1	45.000702	7.68227
1537	1538	pop	51	1766	54276	1	40.323410	17.56827

1389 rows × 9 columns

In [6]:

```
data1=data1.drop(['lat','lon','ID'],axis=1)
```

In [7]:

```
data1
```

Out[7]:

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

1389 rows × 6 columns

In [8]:

```
data1=pd.get_dummies(data1)
```

In [9]:

```
data1
```

Out[9]:

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

1389 rows × 8 columns

In [10]:

```
y=data1['price']
x=data1.drop('price',axis=1)
```

In [11]:

```
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=42)
```

In [12]:

```
x_test
```

Out[12]:

	engine_power	age_in_days	km	previous_owners	model_lounge	model_pop	model_
625	51	3347	148000	1	1	0	
187	51	4322	117000	1	1	0	
279	51	4322	120000	1	0	1	
734	51	974	12500	1	0	1	
315	51	1096	37000	1	1	0	
...	...	...	...	...	...	...	
115	51	397	16135	1	1	0	
370	51	366	11203	1	0	1	
1179	74	3804	62000	1	1	0	
93	51	397	17250	1	1	0	
147	51	762	15917	1	1	0	

459 rows × 7 columns

In [13]:

```
from sklearn.model_selection import GridSearchCV
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]}
elastic_regressor = GridSearchCV(elastic, parameters)
elastic_regressor.fit(x_train, y_train)
```

Out[13]:

```
▸      GridSearchCV
▸ estimator: ElasticNet
      ▸ ElasticNet
```

In [14]:

```
elastic_regressor.best_params_
```

Out[14]:

```
{'alpha': 0.01}
```

In [15]:

```
#x_test=test data,x_train =train data,y_teat=test data,y_train=train data
```

In [16]:

```
elas=ElasticNet(alpha=0.1)
elas.fit(x_train,y_train)#create elas constant with training data
y_pred_elas=elas.predict(x_test)#create predicted values in y_pred_elas
```

In [17]:

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

Out[17]:

```
0.8601270407940889
```

In [18]:

```
from sklearn.metrics import mean_squared_error
elastic_error=mean_squared_error(y_test,y_pred_elas)
elastic_error
```

Out[18]:

```
515678.8171884504
```

In [19]:

```
import seaborn as sns
import matplotlib.pyplot as plt
```

In [20]:

```
y_pred_elas
```

```
3,
        4945.96227363,  9925.91539944,  8759.58490807,  5412.1319524
4,
        9913.48612468,  7309.72876585, 10080.96312405,  8240.8260428
,
       10335.51197773, 10391.46788208,  9712.73120399,  9571.6903694
3,
       10468.74919903,  9077.31512242, 10165.43239312,  9695.7228776
8,
       10351.88848518,  6693.72998721,  8707.67021922,  8712.4195826
8,
        7062.66926379,  9830.8128588 , 10376.7009449 ,  9678.5349336
4,
        4529.65420198,  6064.46834278, 10093.67764805,  9799.8102707
,
        5952.02574344, 10237.5736284 ,  6846.80416328,  9786.9009726
3,
       10534.40843939,  8869.02252909, 10315.69674418, 10534.4084393
9,
        9869.38995966,  6833.21852828,  9788.88698234,  9406.3100667
```

In [21]:

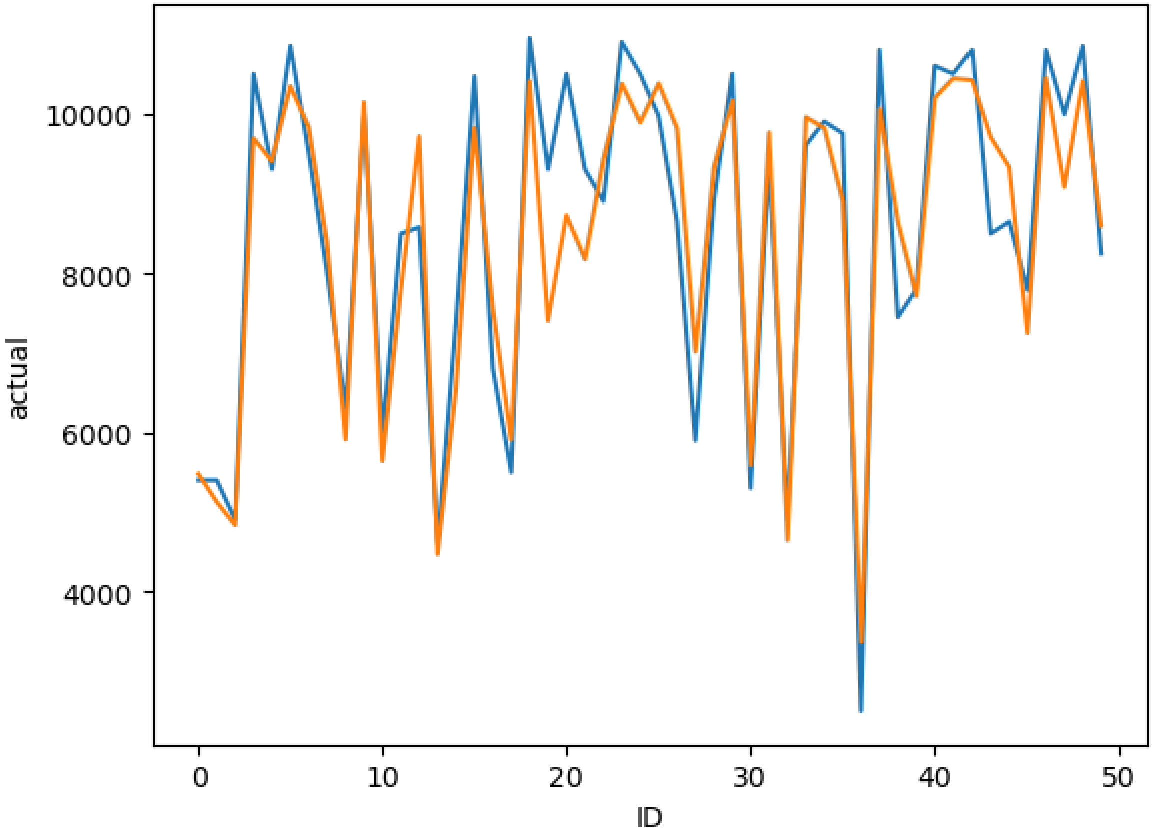
```
res=pd.DataFrame(columns=['actual','predicted'])
res['actual']=y_test
res['predicted']=y_pred_elas
res=res.reset_index()
res['ID']=res.index
```

In [22]:

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

Out[22]:

```
[]
```



In [ ]: