In [178]:
```
import pandas as pd
```

In [179]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [180]:
```
data=pd.read_csv("/home/placement/Downloads/fiat500.csv")
```

In [181]:
```
data.describe()
```

Out[181]:

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price
count	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000
mean	769.500000	51.904421	1650.980494	53396.011704	1.123537	43.541361	11.563428	8576.003901
std	444.126671	3.988023	1289.522278	40046.830723	0.416423	2.133518	2.328190	1939.958641
min	1.000000	51.000000	366.000000	1232.000000	1.000000	36.855839	7.245400	2500.000000
25%	385.250000	51.000000	670.000000	20006.250000	1.000000	41.802990	9.505090	7122.500000
50%	769.500000	51.000000	1035.000000	39031.000000	1.000000	44.394096	11.869260	9000.000000
75%	1153.750000	51.000000	2616.000000	79667.750000	1.000000	45.467960	12.769040	10000.000000
max	1538.000000	77.000000	4658.000000	235000.000000	4.000000	46.795612	18.365520	11100.000000

In [182]:
```
data.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 [183]:
```
data.head()
```

Out[183]:

	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 [184]:
```
data1=data.loc[(data.previous_owners==1)]
```

In [185]:
```
data1
```

Out[185]:

	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
...	...	...	...	...	...	...	...	...	...
1533	1534	sport	51	3712	115280	1	45.069679	7.704920	5200
1534	1535	lounge	74	3835	112000	1	45.845692	8.666870	4600
1535	1536	pop	51	2223	60457	1	45.481541	9.413480	7500
1536	1537	lounge	51	2557	80750	1	45.000702	7.682270	5990
1537	1538	pop	51	1766	54276	1	40.323410	17.568270	7900

1389 rows × 9 columns

In [186]:
```
data1=data.drop(['ID','lat','lon'],axis=1)
```

In [187]:
```
data1
```

Out[187]:

	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

1538 rows × 6 columns

In [188]:
```
data1=pd.get_dummies(data)
```

In [189]:
```
data1
```

Out[189]:

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price	model_lounge	model_pop	model_sport
0	1	51	882	25000	1	44.907242	8.611560	8900	1	0	0
1	2	51	1186	32500	1	45.666359	12.241890	8800	0	1	0
2	3	74	4658	142228	1	45.503300	11.417840	4200	0	0	1
3	4	51	2739	160000	1	40.633171	17.634609	6000	1	0	0
4	5	73	3074	106880	1	41.903221	12.495650	5700	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...
1533	1534	51	3712	115280	1	45.069679	7.704920	5200	0	0	1
1534	1535	74	3835	112000	1	45.845692	8.666870	4600	1	0	0
1535	1536	51	2223	60457	1	45.481541	9.413480	7500	0	1	0
1536	1537	51	2557	80750	1	45.000702	7.682270	5990	1	0	0
1537	1538	51	1766	54276	1	40.323410	17.568270	7900	0	1	0

1538 rows × 11 columns

In [190]:
```
data1.shape
```

Out[190]: (1538, 11)

In [191]:
```
y=data1['price']
x=data1.drop('price',axis=1)
```

In [192]:
```
y
```

Out[192]:
```
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 [193]:
```
x
```

Out[193]:

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	model_lounge	model_pop	model_sport
0	1	51	882	25000	1	44.907242	8.611560	1	0	0
1	2	51	1186	32500	1	45.666359	12.241890	0	1	0
2	3	74	4658	142228	1	45.503300	11.417840	0	0	1
3	4	51	2739	160000	1	40.633171	17.634609	1	0	0
4	5	73	3074	106880	1	41.903221	12.495650	0	1	0
...	...	...	...	...	...	...	...	...	...	...
1533	1534	51	3712	115280	1	45.069679	7.704920	0	0	1
1534	1535	74	3835	112000	1	45.845692	8.666870	1	0	0
1535	1536	51	2223	60457	1	45.481541	9.413480	0	1	0
1536	1537	51	2557	80750	1	45.000702	7.682270	1	0	0
1537	1538	51	1766	54276	1	40.323410	17.568270	0	1	0

1538 rows × 10 columns

In [194]:
```
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 [195]:
```
x_test.head()
```

Out[195]:

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	model_lounge	model_pop	model_sport
481	482	51	3197	120000	2	40.174702	18.167629	0	1	0
76	77	62	2101	103000	1	45.797859	8.644440	0	1	0
1502	1503	51	670	32473	1	41.107880	14.208810	1	0	0
669	670	51	913	29000	1	45.778591	8.946250	1	0	0
1409	1410	51	762	18800	1	45.538689	9.928310	1	0	0

In [196]:
```
y_test.head()
```

Out[196]:
```
481     7900
76      7900
1502    9400
669     8500
1409    9700
Name: price, dtype: int64
```

In [197]:
```
x_train.head()
```

Out[197]:

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	model_lounge	model_pop	model_sport
527	528	51	425	13111	1	45.022388	7.58602	1	0	0
129	130	51	1127	21400	1	44.332531	7.54592	1	0	0
602	603	51	2039	57039	1	40.748241	14.52835	0	1	0
331	332	51	1155	40700	1	42.143860	12.54016	1	0	0
323	324	51	425	16783	1	41.903221	12.49565	1	0	0

In [198]:
```
y_train.head()
```

Out[198]:
```
527    9990
129    9500
602    7590
331    8750
323    9100
Name: price, dtype: int64
```

In [199]:
```
from sklearn.linear_model import ElasticNet
from sklearn.model_selection import GridSearchCV

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[199]:
```
▸ GridSearchCV
▸ estimator: ElasticNet
    ▸ ElasticNet
```

In [200]:
```
elastic_regressor.best_params_
```

Out[200]:
```
{'alpha': 0.01}
```

In [201]:
```
elastic=ElasticNet(alpha=.01)
elastic.fit(x_train,y_train)
y_pred_elastic=elastic.predict(x_test)
```

In [202]:
```
y_pred_elastic
```

Out[202]:
```
array([ 5818.88155606,  7251.60740134,  9741.37548309,  9798.17563202,
       10054.18656906,  9551.03709384,  9756.60308305, 10122.24386003,
        9655.23079132,  9250.43675403, 10477.46946422,  7805.98706383,
        7703.41076005,  6299.65982145,  9544.86952947, 10422.52128897,
        9622.08375661,  7754.80253699,  4897.38480628, 10580.47637966,
       10461.71330862, 10442.89200323,  7512.81872938, 10027.42619282,
        6990.9795985 ,  8993.93966292,  4822.9274055 ,  6987.38971162,
        7821.74361059,  9682.14401582,  7342.84571741,  5338.98378321,
        5422.36411921,  5088.69094616,  8964.54155191,  5701.00133273,
        9923.63970799,  8329.65576797,  6219.5421374 ,  8392.88958456,
        9695.35125166,  6862.17433129,  9106.3590491 , 10067.46187307,
        8620.73186465, 10175.36762819,  9067.11601592,  8866.57789002,
        7092.28977184,  9057.74520111,  9474.38711846, 10405.31440756,
       10112.23689137,  6823.98653466,  9699.84010068,  9386.45543089,
        9637.09234337, 10553.05108845,  9846.08937468,  7250.04609281,
        9989.41187642,  7082.51579887,  9976.61490792,  7243.17422389,
        6488.79520193,  9738.09464099,  9852.77779753,  8572.18811531,
        8505.72502159,  6482.56240442,  7881.15444662,  6870.90829793,
        8262.43238047, 10550.05609925,  7433.16232868,  8636.30752919,
```

In [203]:
```python
from sklearn.metrics import r2_score
r2_score(y_test,y_pred_elastic)
```

Out[203]:
```
0.8429739684420192
```

In [204]:
```python
from sklearn.metrics import mean_squared_error
elastic_Error=mean_squared_error(y_pred_elastic,y_test)
elastic_Error
```

Out[204]:
```
576668.2037947337
```

In [205]:
```
Results=pd.DataFrame(columns=['Actual','predicted'])
Results['Actual']=y_test
Results['Predicted']=y_pred_elastic
Results=Results.reset_index()
Results['Id']=Results.index
Results
```

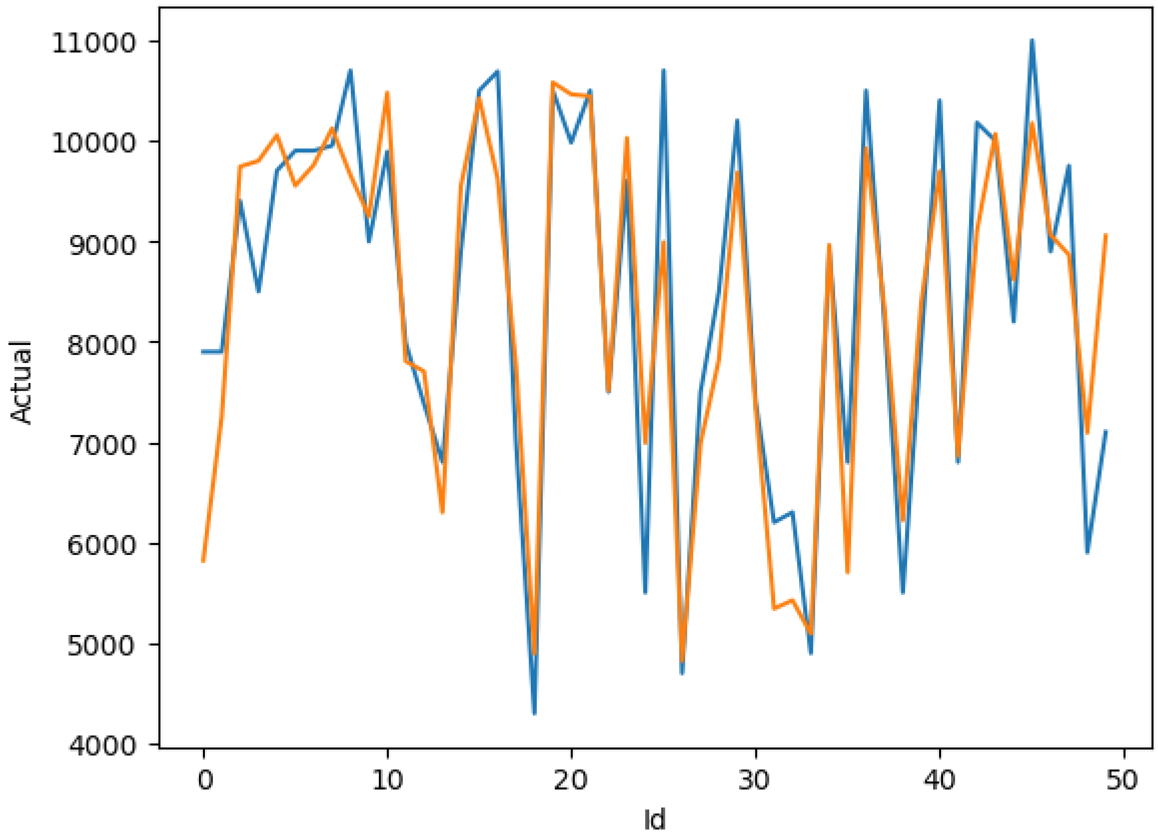
Out[205]:

	index	Actual	predicted	Predicted	Id
0	481	7900	NaN	5818.881556	0
1	76	7900	NaN	7251.607401	1
2	1502	9400	NaN	9741.375483	2
3	669	8500	NaN	9798.175632	3
4	1409	9700	NaN	10054.186569	4
...	...	...	...	...	...
503	291	10900	NaN	10120.713199	503
504	596	5699	NaN	6291.601668	504
505	1489	9500	NaN	10020.222896	505
506	1436	6990	NaN	8247.810365	506
507	575	10900	NaN	10337.015702	507

508 rows × 5 columns

In [206]:
```
import seaborn as sns
import matplotlib.pyplot as plt
sns.lineplot(x='Id',y='Actual',data=Results.head(50))
sns.lineplot(x='Id',y='Predicted',data=Results.head(50))
plt.plot()
```

Out[206]: []



In [ ]:

```
 
```