In [5]:
```
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
```

In [6]:
```
data=pd.read_csv("/home/placement/Downloads/fiat500.csv")
```

In [7]:
```
data.describe()
```

Out[7]:

| | 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 [8]:
```
data.info
```

Out[8]:
```
<bound method DataFrame.info of         ID   model  engine_power  age_in_days      km  previous_owners  \
0        1  lounge            51          882   25000                1
1        2     pop            51         1186   32500                1
2        3   sport            74         4658  142228                1
3        4  lounge            51         2739  160000                1
4        5     pop            73         3074  106880                1
...    ...     ...           ...          ...     ...              ...
1533  1534   sport            51         3712  115280                1
1534  1535  lounge            74         3835  112000                1
1535  1536     pop            51         2223   60457                1
1536  1537  lounge            51         2557   80750                1
1537  1538     pop            51         1766   54276                1

            lat        lon  price
0     44.907242   8.611560   8900
1     45.666359  12.241890   8800
2     45.503300  11.417840   4200
3     40.633171  17.634609   6000
4     41.903221  12.495650   5700
```

In [9]:
```
data.head(10)
```

Out[9]:

| | 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 |
| **5** | 6 | pop | 74 | 3623 | 70225 | 1 | 45.000702 | 7.682270 | 7900 |
| **6** | 7 | lounge | 51 | 731 | 11600 | 1 | 44.907242 | 8.611560 | 10750 |
| **7** | 8 | lounge | 51 | 1521 | 49076 | 1 | 41.903221 | 12.495650 | 9190 |
| **8** | 9 | sport | 73 | 4049 | 76000 | 1 | 45.548000 | 11.549470 | 5600 |
| **9** | 10 | sport | 51 | 3653 | 89000 | 1 | 45.438301 | 10.991700 | 6000 |

In [10]:
```
data1=data.loc[(data.previous_owners==1)]
```

In [11]:
```
data1
```

Out[11]:

| | 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 |

In [12]:
```
data1=data.drop(['ID','lat','lon'],axis=1)
```

In [13]:
```
data1
```

Out[13]:

| | 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 |

In [14]:
```
data=pd.get_dummies(data)
```

In [15]:
```
data
```

Out[15]:

| | 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 [16]:
```
data.shape
```

Out[16]: (1538, 11)

In [17]:
```
y=data['price']
x=data.drop('price',axis=1)
```

In [18]:
```
y
```

Out[18]:
```
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 [19]:
```
x
```

Out[19]:

| | 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 |

In [33]:
```
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 [34]:
```
x_test.head(5)
```

Out[34]:

| | 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 [35]:
```
y_test.head(5)
```

Out[35]:
```
481     7900
76      7900
1502    9400
669     8500
1409    9700
Name: price, dtype: int64
```

In [36]:
```
x_train.head()
```

Out[36]:

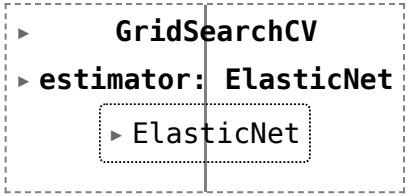
| | ID | engine_power | age_in_days | km | previous_owners | lat | lon | model_lounge | model_pop | model_sport |
|---|---|---|---|---|---|---|---|---|---|---|
| **527** | 528 | 51 | 425 | 13111 | 1 | 45.022388 | 7.58602 | 1 | 0 | 0 |
| **129** | 130 | 51 | 1127 | 21400 | 1 | 44.332531 | 7.54592 | 1 | 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 [37]:
```
y_train.head()
```

Out[37]:
```
527    9990
129    9500
602    7590
331    8750
323    9100
Name: price, dtype: int64
```

In [38]:
```
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[38]:




In [39]:
```
elastic_regressor.best_params_
```

Out[39]:
```
{'alpha': 0.01}
```

In [40]:
```
elastic=ElasticNet(alpha=30)
elastic.fit(x_train,y_train)
y_pred_elastic=elastic.predict(x_test)
```

In [41]:
```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred_elastic)
```

Out[41]: 0.8416206414238153

In [42]:
```
from sklearn.metrics import mean_squared_error
elastic_Error=mean_squared_error(y_pred_elastic,y_test)
elastic_Error
```

Out[42]: 581638.2119710302

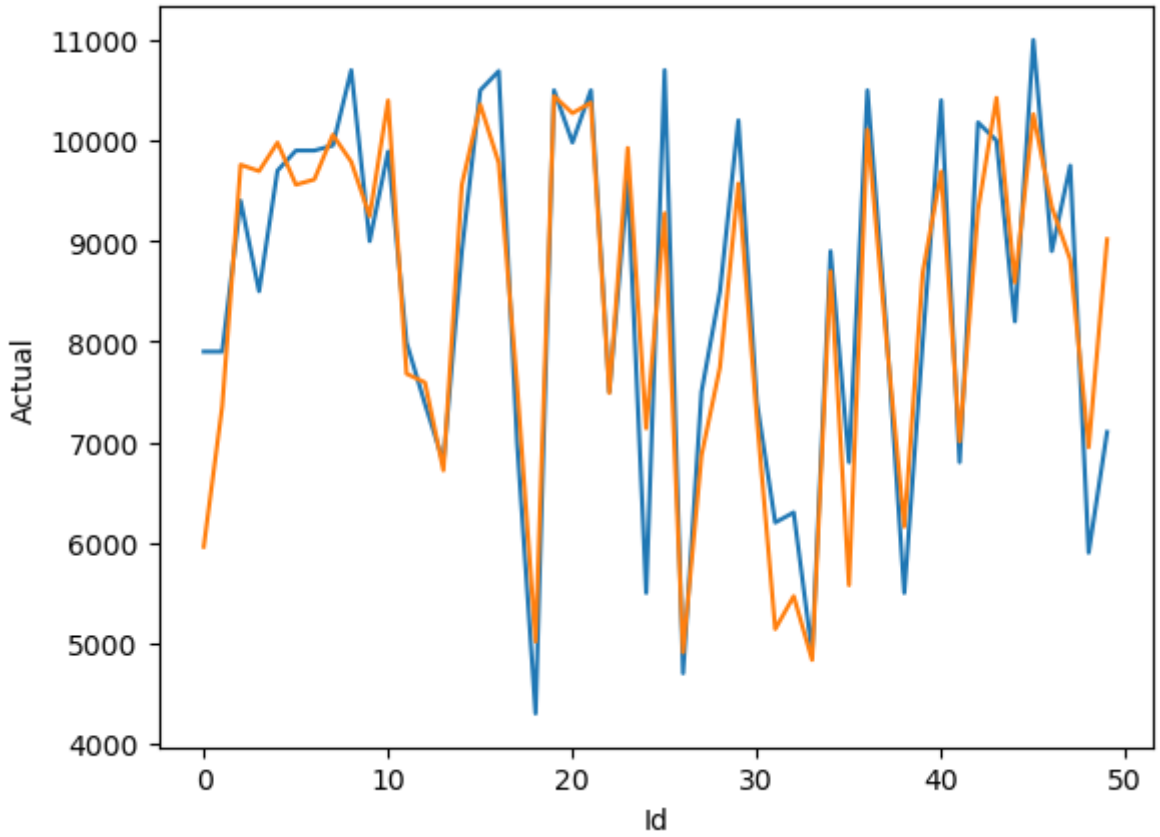
In [43]:
```
Results=pd.DataFrame(columns=['Actual','predicted'])
Results['Actual']=y_test
Results['predicted']=y_pred_elastic
Results=Results.reset_index()
Results['Id']=Results.index
Results
```

| | index | Actual | predicted | Id |
|---|---|---|---|---|
| **0** | 481 | 7900 | 5988.777085 | 0 |
| **1** | 76 | 7900 | 7393.904731 | 1 |
| **2** | 1502 | 9400 | 9726.595326 | 2 |
| **3** | 669 | 8500 | 9693.681751 | 3 |
| **4** | 1409 | 9700 | 9940.773084 | 4 |
| **...** | ... | ... | ... | ... |
| **503** | 291 | 10900 | 10028.732370 | 503 |
| **504** | 596 | 5699 | 6516.798511 | 504 |
| **505** | 1489 | 9500 | 10209.647976 | 505 |
| **506** | 1436 | 6990 | 8224.153844 | 506 |
| **507** | 575 | 10900 | 10329.915814 | 507 |

508 rows × 4 columns

In [32]:
```
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[32]:
```
<function matplotlib.pyplot.plot(*args, scalex=True, scaley=True, data=None, **kwargs)>
```




In [ ]:
```

```