```
In [52]:
           import pandas as a
           import warnings
           warnings.filterwarnings("ignore")
In [53]: data=pd.read_csv("fiat500.csv")
In [54]: data
Out[54]:
                   ID model engine_power age_in_days
                                                          km previous_owners
                                                                                    lat
                                                                                              Ion price
               0
                    1 lounge
                                       51
                                                   882
                                                        25000
                                                                            1 44.907242
                                                                                         8.611560
                                                                                                  8900
                                       51
                                                  1186
                                                        32500
                                                                            1 45.666359 12.241890
                                                                                                  8800
              1
                         pop
               2
                                                  4658
                                                       142228
                                                                            1 45.503300 11.417840
                                                                                                  4200
                        sport
                                       74
                    4 lounge
                                       51
                                                  2739
                                                       160000
                                                                            1 40.633171 17.634609
                                                                                                  6000
                                       73
                                                  3074 106880
                                                                            1 41.903221 12.495650
                                                                                                  5700
                         pop
                                                  3712 115280
            1533 1534
                        sport
                                       51
                                                                            1 45.069679
                                                                                         7.704920
                                                                                                  5200
                 1535
                                                  3835
                                                       112000
            1534
                       lounge
                                       74
                                                                            1 45.845692
                                                                                         8.666870
                                                                                                  4600
```

1 45.481541

1 45.000702

1 40.323410 17.568270

9.413480

7.682270

7500

5990

7900

1538 rows × 9 columns

1535 1536

1537 1538

1537

1536

In [55]: data=data.loc[(data.previous_owners==1)]

pop

pop

lounge

51

51

51

2223

2557

1766

60457

80750

54276

In [56]: data

Out[56]:

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

1389 rows × 9 columns

In [57]: data=data.drop(['ID','lat','lon'],axis=1)

In [58]: data

Out[58]:

	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	рор	51	1766	54276	1	7900

1389 rows × 6 columns

```
In [59]: data=pd.get_dummies(data)
In [60]: data.shape
Out[60]: (1389, 8)
In [61]: y=data['price']
x=data.drop('price',axis=1)
```

```
In [62]: y
Out[62]: 0
                  8900
                  8800
          2
                  4200
          3
                  6000
                  5700
          4
                   . . .
          1533
                  5200
          1534
                  4600
          1535
                  7500
          1536
                  5990
          1537
                  7900
          Name: price, Length: 1389, dtype: int64
In [63]: from sklearn.model_selection import train_test_split
          x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.1,random state=42)
In [64]: x_test.head(5)
Out[64]:
               engine_power age_in_days
                                         km previous_owners model_lounge model_pop model_sport
           625
                        51
                                 3347 148000
                                                        1
                                                                     1
                                                                               0
                                                                                          0
           187
                        51
                                 4322
                                     117000
                                                        1
                                                                     1
                                                                               0
                                                                                          0
           279
                                      120000
                        51
                                 4322
                                                        1
                                                                     0
                                                                               1
           734
                        51
                                  974
                                       12500
                                                        1
                                                                     0
                                                                               1
                                                                                          0
```

```
In [65]: | from sklearn.model selection import GridSearchCV
         from sklearn.linear model import Ridge
         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[65]:
                GridSearchCV
          ▶ estimator: ElasticNet
                ▶ ElasticNet
In [66]: elastic regressor.best params
Out[66]: {'alpha': 0.01}
In [67]: elastic=ElasticNet(alpha=.01)
         elastic.fit(x train,y train)
         y pred elastic=elastic.predict(x test)
In [68]: from sklearn.metrics import r2 score
         r2 score(y test,y pred elastic)
Out[68]: 0.8488682857174344
In [69]: from sklearn.metrics import mean squared error
         elastic Error=mean squared error(y pred elastic,y test)
         elastic Error
Out[69]: 603966.023413073
```

localhost:8888/notebooks/elastic25.ipynb

```
In [70]: Results= pd.DataFrame(columns=['Price','Predicted'])
    Results['Price']=y_test
    Results['Predicted']=y_pred_elastic
    Results=Results.reset_index()
    Results['Id']=Results.index
    Results.head(10)
```

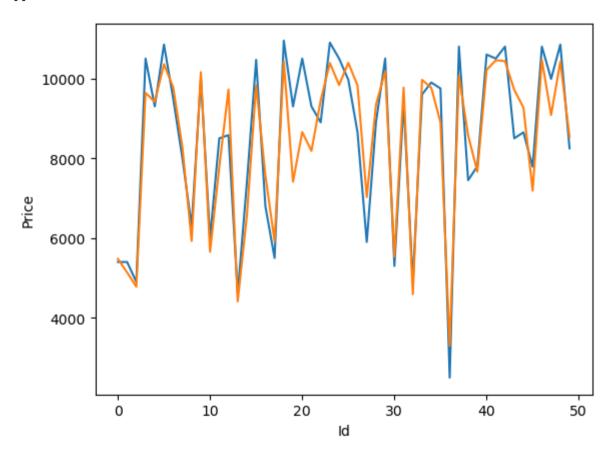
Out[70]:

	index	Price	Predicted	ld
0	625	5400	5477.052458	0
1	187	5399	5137.435504	1
2	279	4900	4778.564980	2
3	734	10500	9640.895436	3
4	315	9300	9415.174300	4
5	652	10850	10356.323449	5
6	1472	9500	9781.272728	6
7	619	7999	8276.238400	7
8	992	6300	5925.267808	8
9	1154	10000	10158.433547	9

```
In [71]: 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[71]: []



In	[1:	
In	[1:	
In	[]:	
In	[]:	
In	[1:	