```
In [38]: import pandas as pd
         import warnings
         warnings.filterwarnings("ignore")
In [39]: data=pd.read_csv("fiat500 (1).csv")
In [40]: data
```

Out[40]:

		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
15	<b>33</b> 1	1534	sport	51	3712	115280	1	45.069679	7.704920	5200
15	<b>34</b> 1	1535	lounge	74	3835	112000	1	45.845692	8.666870	4600
15	<b>35</b> 1	1536	pop	51	2223	60457	1	45.481541	9.413480	7500
15	<b>36</b> 1	1537	lounge	51	2557	80750	1	45.000702	7.682270	5990
15	<b>37</b> 1	1538	pop	51	1766	54276	1	40.323410	17.568270	7900

1538 rows × 9 columns

In [41]: | data=data.loc[(data.previous\_owners==1)]

In [42]: data

Out[42]:

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 1537	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 [43]: data=data.drop(['ID','lat','lon'],axis=1)

In [44]: data

Out[44]:

	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 [45]: data=pd.get_dummies(data)
In [46]: data.shape
Out[46]: (1389, 8)
In [47]: y=data['price']
    x=data.drop('price',axis=1)
```

```
In [48]: y
Out[48]: 0
                    8900
                    8800
           2
                    4200
           3
                     6000
           4
                     5700
           1533
                    5200
           1534
                    4600
           1535
                    7500
           1536
                    5990
           1537
                    7900
           Name: price, Length: 1389, dtype: int64
In [49]: 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 [50]: X_test.head(5)
Out[50]:
```

	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	51	4322	120000	1	0	1	0
734	51	974	12500	1	0	1	0
315	51	1096	37000	1	1	0	0

```
In [51]: 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[51]:
                                             GridSearchCV
          GridSearchCV(estimator=ElasticNet(),
                       param grid={'alpha': [1e-15, 1e-10, 1e-08, 0.0001, 0.001, 0.01, 1,
                                              5, 10, 20]})
                                       ▼ estimator: ElasticNet
                                       ElasticNet()
                                             ▼ ElasticNet
                                            ElasticNet()
```

```
In [52]: elastic_regressor.best_params_
Out[52]: {'alpha': 0.01}
```

```
In [57]: elastic=ElasticNet(alpha=0.01)
    elastic.fit(X_train,y_train)
    y_pred_elastic=elastic.predict(X_test)

In [58]: from sklearn.metrics import r2_score
    r2_score(y_test,y_pred_elastic)

Out[58]: 0.8488682857174344

In [63]: from sklearn.metrics import mean_squared_error
    elastic_Error=mean_squared_error(y_pred_elastic,y_test)
    elastic_Error
Out[63]: 603966.023413073
```

```
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(25)
```

## Out[70]:

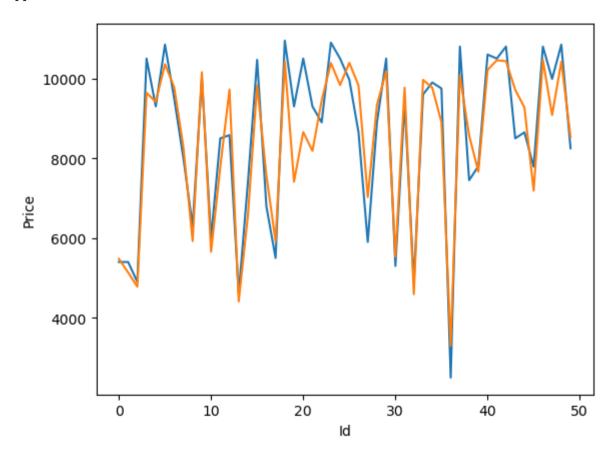
	index	Price	Predicted	ld
0	625	5400	5477.052458	0
1	187	5399	5137.435504	1
2	279	4900	4778.564980	2
3	734	10500	9640.895436	3
4	315	9300	9415.174300	4
5	652	10850	10356.323449	5
6	1472	9500	9781.272728	6
7	619	7999	8276.238400	7
8	992	6300	5925.267808	8
9	1154	10000	10158.433547	9
10	757	6000	5654.915390	10
11	1299	8500	7779.899617	11
12	400	8580	9724.510940	12
13	314	4600	4411.587148	13
14	72	7400	6568.196031	14
15	265	10470	9832.106012	15
16	800	6800	7576.247388	16
17	116	5500	5921.661919	17
18	181	10950	10422.823376	18
19	564	9300	7412.883090	19
20	1008	10500	8656.046516	20

	index	Price	Predicted	ld
21	1035	9300	8184.755615	21
22	1194	8900	9448.594403	22
23	131	10900	10388.473661	23
24	688	10499	9836.026696	24

```
In [69]: 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[69]: []



```
In [ ]:

In [ ]:
```

In	]:	
In	]:	
In	1:	