In [1]:

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

In [2]:

```
data=pd.read_csv("/home/palcement/Downloads/fiat500.csv")
data
```

Out[2]:

	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

1538 rows × 9 columns

In [3]:
```
data1=data.loc[(data.previous_owners==1)]
data1
```

Out[3]:

	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 [4]:

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

Out[4]:

	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 [5]:

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

Out[5]:

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

1389 rows × 8 columns

In [6]:

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

In [7]:
```
y
```

Out[7]:
```
0       8900
1       8800
2       4200
3       6000
4       5700
        ...
1533    5200
1534    4600
1535    7500
1536    5990
1537    7900
Name: price, Length: 1389, dtype: int64
```

In [8]:
```
X
```

Out[8]:

	engine_power	age_in_days	km	previous_owners	model_lounge	model_pop	model_sport
0	51	882	25000	1	1	0	0
1	51	1186	32500	1	0	1	0
2	74	4658	142228	1	0	0	1
3	51	2739	160000	1	1	0	0
4	73	3074	106880	1	0	1	0
...	...	...	...	...	...	...	...
1533	51	3712	115280	1	0	0	1
1534	74	3835	112000	1	1	0	0
1535	51	2223	60457	1	0	1	0
1536	51	2557	80750	1	1	0	0
1537	51	1766	54276	1	0	1	0

1389 rows × 7 columns

In [46]:
```
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 [47]:
```
X_train
```

Out[47]:

	engine_power	age_in_days	km	previous_owners	model_lounge	model_pop	model_sport
915	51	397	17081	1	1	0	0
12	51	456	18450	1	1	0	0
638	51	397	21276	1	1	0	0
190	51	821	19000	1	1	0	0
701	51	701	27100	1	1	0	0
...	...	...	...	...	...	...	...
1201	51	790	50740	1	0	1	0
1239	51	4383	107600	1	0	1	0
1432	51	701	42095	1	1	0	0
951	51	3684	78000	1	1	0	0
1235	51	1613	45000	1	1	0	0

930 rows × 7 columns

```
In [48]: y_train
```

```
Out[48]: 915     10900
         12       9700
         638     10850
         190      9990
         701     10300
                 ...  
         1201     8300
         1239     3950
         1432     8900
         951      6500
         1235     8800
         Name: price, Length: 930, dtype: int64
```

```
In [49]: y_test
```

```
Out[49]: 625      5400
         187      5399
         279      4900
         734     10500
         315      9300
                 ...  
         115     10650
         370      9900
         1179     5900
         93      10050
         147      9900
         Name: price, Length: 459, dtype: int64
```

In [50]:
```
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[50]:
```
▸ GridSearchCV
▸ estimator: ElasticNet
  ▸ ElasticNet
```

In [51]:
```
elastic_regressor.best_params_
```

Out[51]:
```
{'alpha': 0.01}
```

In [52]:
```
elastic=ElasticNet(alpha=0.1)
elastic.fit(X_train,y_train)
y_pred_elastic=elastic.predict(X_test)
```

In [53]:
```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred_elastic)
```

Out[53]:
```
0.8601270407940889
```

In [54]:
```
from sklearn.metrics import mean_squared_error
elastic_Error=mean_squared_error(y_pred_elastic,y_test)
elastic_Error
```

Out[54]:
```
515678.8171884504
```

In [55]:

```
Results=pd.DataFrame(columns=['Price','Predicted']) #price and predicted names are our wish
Results['Price']=y_test
Results['Predicted']=y_pred_elastic
Results=Results.reset_index()
Results['Id']=Results.index
```

In [56]:

```
Results
```

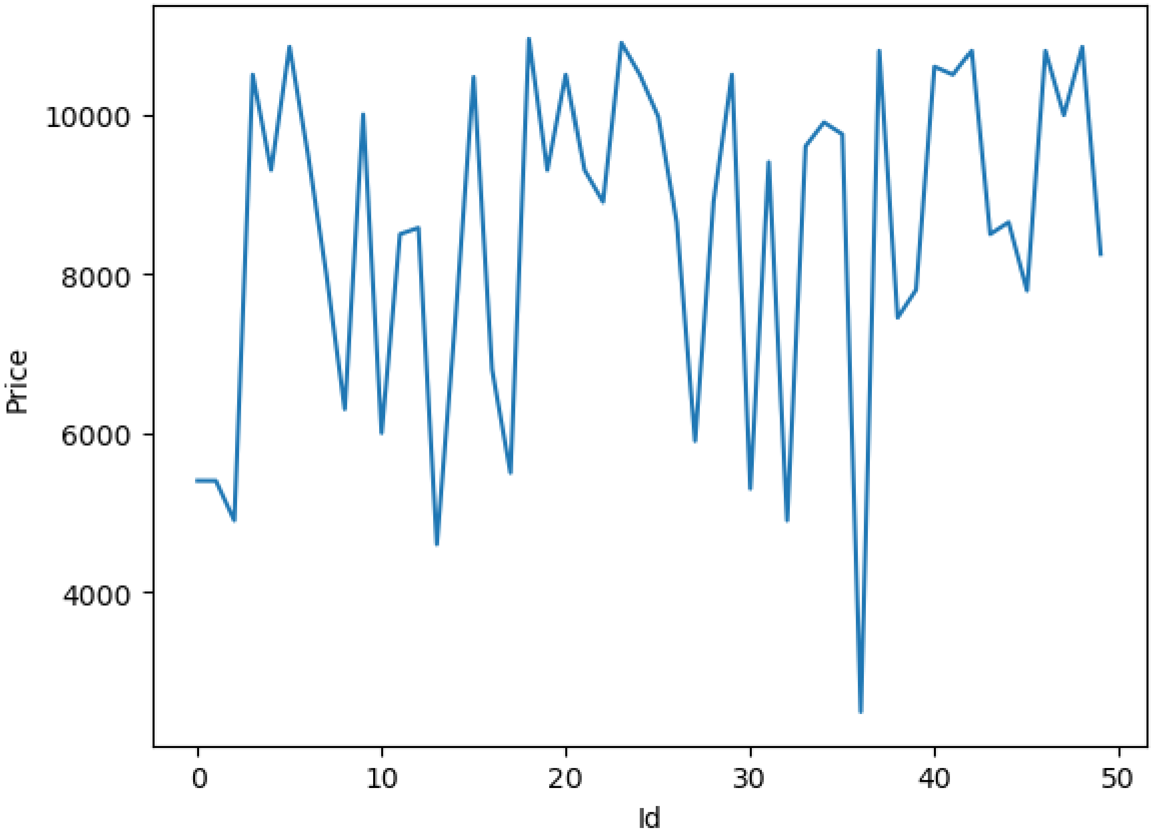
Out[56]:

```
     index  Price     Predicted   Id
  0    625   5400   5478.361166    0
  1    187   5399   5124.950418    1
  2    279   4900   4833.208393    2
  3    734  10500   9688.909121    3
  4    315   9300   9402.252771    4
...    ...    ...           ...  ...
454    115  10650  10389.152700  454
455    370   9900  10260.455864  455
456   1179   5900   6773.307749  456
457     93  10050  10370.171613  457
458    147   9900  10063.002616  458

459 rows × 4 columns
```

```
In [58]: import seaborn as sns
         import matplotlib.pyplot as plt
         sns.lineplot(x='Id',y='Price',data=Results.head(50)) #Orange color
         sns.lineplot(x='Id',y='Predicted',data=Results.head(50))  #Blue color
         plt.plot()

Out[58]: []
```



In [ ]: