In [53]:
```
!pip3 install pandas
```

```
Requirement already satisfied: pandas in ./anaconda3/lib/python3.10/site-packages (1.5.3)
Requirement already satisfied: python-dateutil>=2.8.1 in ./anaconda3/lib/python3.10/site-packages (from pan
das) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in ./anaconda3/lib/python3.10/site-packages (from pandas) (202
2.7)
Requirement already satisfied: numpy>=1.21.0 in ./anaconda3/lib/python3.10/site-packages (from pandas) (1.2
3.5)
Requirement already satisfied: six>=1.5 in ./anaconda3/lib/python3.10/site-packages (from python-dateutil>=
2.8.1->pandas) (1.16.0)
```

In [54]:
```
import pandas as pd
import numpy as np
```

In [55]:
```
data=pd.read_csv("/home/placement/Downloads/fiat500.csv")
```

In [56]:
```
data
```

Out[56]:

	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 [57]:
```
data=data.loc[data.previous_owners==1]
data
```

Out[57]:

	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 [58]:
```
data=data.drop(['ID','lat','lon'],axis=1)
```

In [59]:
```
data
```

Out[59]:

	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 [60]:
```
data=pd.get_dummies(data)
```

In [61]:
```
data.shape
```

Out[61]:
```
(1389, 8)
```

In [62]:
```
y=data['price']
x=data.drop('price',axis=1)
```

In [63]:
```
y
```

Out[63]:
```
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 [64]:
```
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 [65]:
```
x_test.head(5)
```

Out[65]:

	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

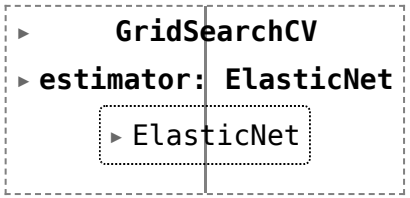
In [ ]:
```

```

In [66]:
```
import pandas as a
import pickle
import warnings
warnings.filterwarnings("ignore")
```

In [ ]:

In [67]:
```
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import Ridge

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[67]:



In [68]:
```
elastic_regressor.best_params_
```

Out[68]:
```
{'alpha': 0.01}
```

In [69]:
```
elastic=ElasticNet(alpha=0.01)
elastic.fit(x_train,y_train)
y_pred_elastic=elastic.predict(x_test)
```

In [70]:
```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred_elastic)
```

Out[70]:
```
0.8488682857174344
```

In [71]:
```
from sklearn.metrics import mean_squared_error
elastic_Error=mean_squared_error(y_pred_elastic,y_test)
elastic_Error
```

Out[71]:
```
603966.023413073
```

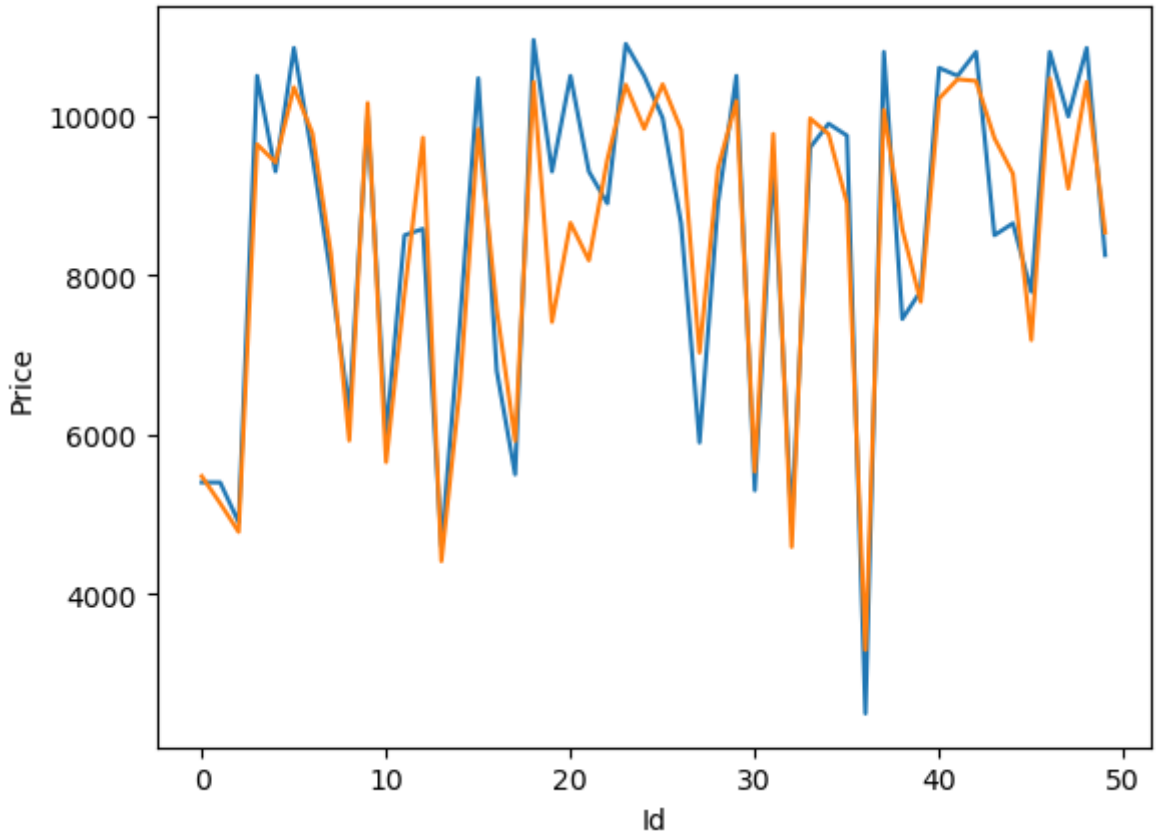
In [73]:
```
Results=pd.DataFrame(columns=['Price','Predicted'])
Results['Price']=y_test
Results['Predicted']=y_pred_elastic
#result['km']=x_test['km']
Results=Results.reset_index()
Results['Id']=Results.index
Results.head(10)
```

Out[73]:

	index	Price	Predicted	Id
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 [74]:
```
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[74]: []



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: