In [23]: `import pandas as pd`

In [24]: `data=pd.read_csv("fiat500 (1).csv")`

In [25]: `data`

Out[25]:

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

In [27]: `data`

Out[27]:

	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

1538 rows × 6 columns

In [46]: `# ridge regression`

In [47]:
```python
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import Ridge

alpha = [1e-15, 1e-10, 1e-8, 1e-4, 1e-3,1e-2, 1, 5, 10, 20,30]

ridge = Ridge()

parameters = {'alpha': alpha}

ridge_regressor = GridSearchCV(ridge, parameters)

ridge_regressor.fit(X_train, y_train)
```

Out[47]:
```
GridSearchCV(estimator=Ridge(),
             param_grid={'alpha': [1e-15, 1e-10, 1e-08, 0.0001, 0.001, 0.01, 1,
                                   5, 10, 20, 30]})
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [48]:
```python
ridge_regressor.best_params_
```

Out[48]:
```
{'alpha': 30}
```

In [49]:
```python
ridge=Ridge(alpha=30)
ridge.fit(X_train,y_train)
y_pred_ridge=ridge.predict(X_test)
```

In [50]:
```python
Ridge_Error=mean_squared_error(y_pred_ridge,y_test)
Ridge_Error
```

Out[50]:
```
579521.7970897449
```

In [51]:
```python
from sklearn.metrics import r2_score
r2_score(y_test,ypred)
```

Out[51]:
```
0.8415526986865394
```

In [52]:
```python
Results= pd.DataFrame(columns=['Price','Predicted'])
Results['Price']=y_test
Results['Predicted']=y_pred_ridge
Results=Results.reset_index()
Results['Id']=Results.index
Results.head(25)
```
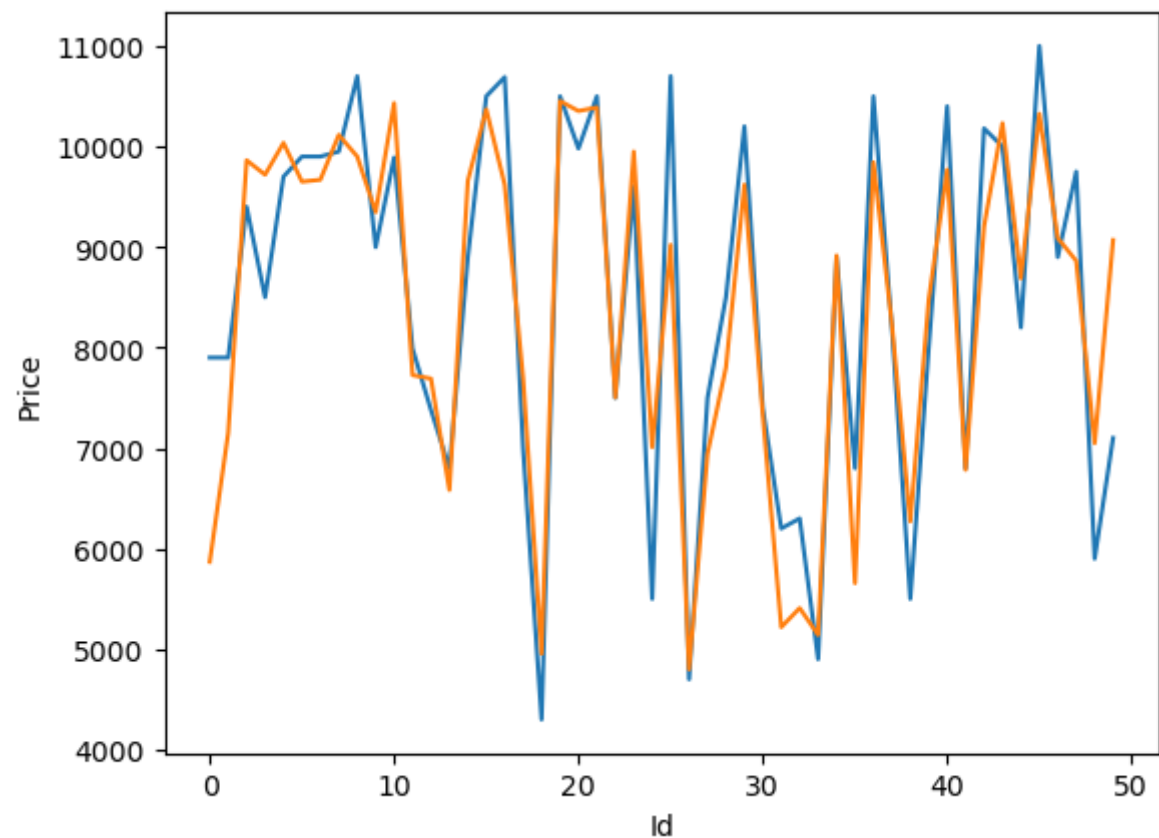
Out[52]:

	index	Price	Predicted	Id
0	481	7900	5869.741155	0
1	76	7900	7149.563327	1
2	1502	9400	9862.785355	2
3	669	8500	9719.283532	3
4	1409	9700	10035.895686	4
5	1414	9900	9650.311090	5
6	1089	9900	9669.183317	6
7	1507	9950	10115.128380	7
8	970	10700	9900.241944	8
9	1198	8999	9347.080772	9
10	1088	9890	10431.237961	10
11	576	7990	7725.756431	11
12	965	7380	7691.089846	12
13	1488	6800	6583.674680	13
14	1432	8900	9659.240069	14
15	380	10500	10370.231518	15
16	754	10690	9620.427488	16
17	30	6990	7689.189244	17
18	49	4300	4954.595074	18
19	240	10500	10452.262871	19
20	344	9980	10353.107796	20

	index	Price	Predicted	Id
21	354	10500	10388.635632	21
22	124	7500	7503.302407	22
23	383	9600	9948.970588	23
24	1389	5500	7009.047336	24

In [53]:
```python
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[53]: []

In [54]:
```python
# Elastic
```