In [1]:
```
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
data=pd.read_csv("/home/placement/Downloads/fiat500.csv")
```

In [3]:
```
data1=data.loc[(data.model=='lounge')]
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
3	4	lounge	51	2739	160000	1	40.633171	17.634609	6000
6	7	lounge	51	731	11600	1	44.907242	8.611560	10750
7	8	lounge	51	1521	49076	1	41.903221	12.495650	9190
11	12	lounge	51	366	17500	1	45.069679	7.704920	10990
...	...	...	...	...	...	...	...	...	...
1528	1529	lounge	51	2861	126000	1	43.841980	10.515310	5500
1529	1530	lounge	51	731	22551	1	38.122070	13.361120	9900
1530	1531	lounge	51	670	29000	1	45.764648	8.994500	10800
1534	1535	lounge	74	3835	112000	1	45.845692	8.666870	4600
1536	1537	lounge	51	2557	80750	1	45.000702	7.682270	5990

1094 rows × 9 columns

In [4]:
```
data1=data.drop(['lat','lon','lon'],axis=1)
```

In [5]:
```
data1=pd.get_dummies(data1)
```

In [6]:
```
data1
```

Out[6]:

	ID	engine_power	age_in_days	km	previous_owners	price	model_lounge	model_pop	model_sport
0	1	51	882	25000	1	8900	1	0	0
1	2	51	1186	32500	1	8800	0	1	0
2	3	74	4658	142228	1	4200	0	0	1
3	4	51	2739	160000	1	6000	1	0	0
4	5	73	3074	106880	1	5700	0	1	0
...	...	...	...	...	...	...	...	...	...
1533	1534	51	3712	115280	1	5200	0	0	1
1534	1535	74	3835	112000	1	4600	1	0	0
1535	1536	51	2223	60457	1	7500	0	1	0
1536	1537	51	2557	80750	1	5990	1	0	0
1537	1538	51	1766	54276	1	7900	0	1	0

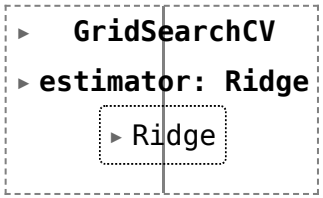
1538 rows × 9 columns

In [7]:
```
y=data1['price']
x=data1.drop('price',axis=1)
```

In [8]:
```
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 [10]:
```
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, 1e-1, 1, 5, 10, 20, 30]

ridge = Ridge()

parameters = {'alpha':alpha}

ridge_regressor = GridSearchCV(ridge, parameters)

ridge_regressor.fit(X_train, Y_train)
```

Out[10]:



In [11]:
```
ridge_regressor.best_params_
```

Out[11]:
```
{'alpha': 30}
```

In [12]:
```
ridge=Ridge(alpha=30)
ridge.fit(X_train,Y_train)
y_pred_ridge=ridge.predict(X_test)
```

In [15]:
```
from sklearn.metrics import mean_squared_error
Ridge_Error=mean_squared_error(y_pred_ridge,Y_test)
Ridge_Error
```

Out[15]:
```
583469.403836385
```

In [14]:
```
from sklearn.metrics import r2_score
r2_score(Y_test,y_pred_ridge)
```

Out[14]:
```
0.8411220101662815
```

In [16]:
```
Results=pd.DataFrame(columns=['price','predicted'])
Results['price']=Y_test
Results['predicted']=y_pred_ridge
Results=Results.reset_index()
Results['ID']=Results.index
Results
```

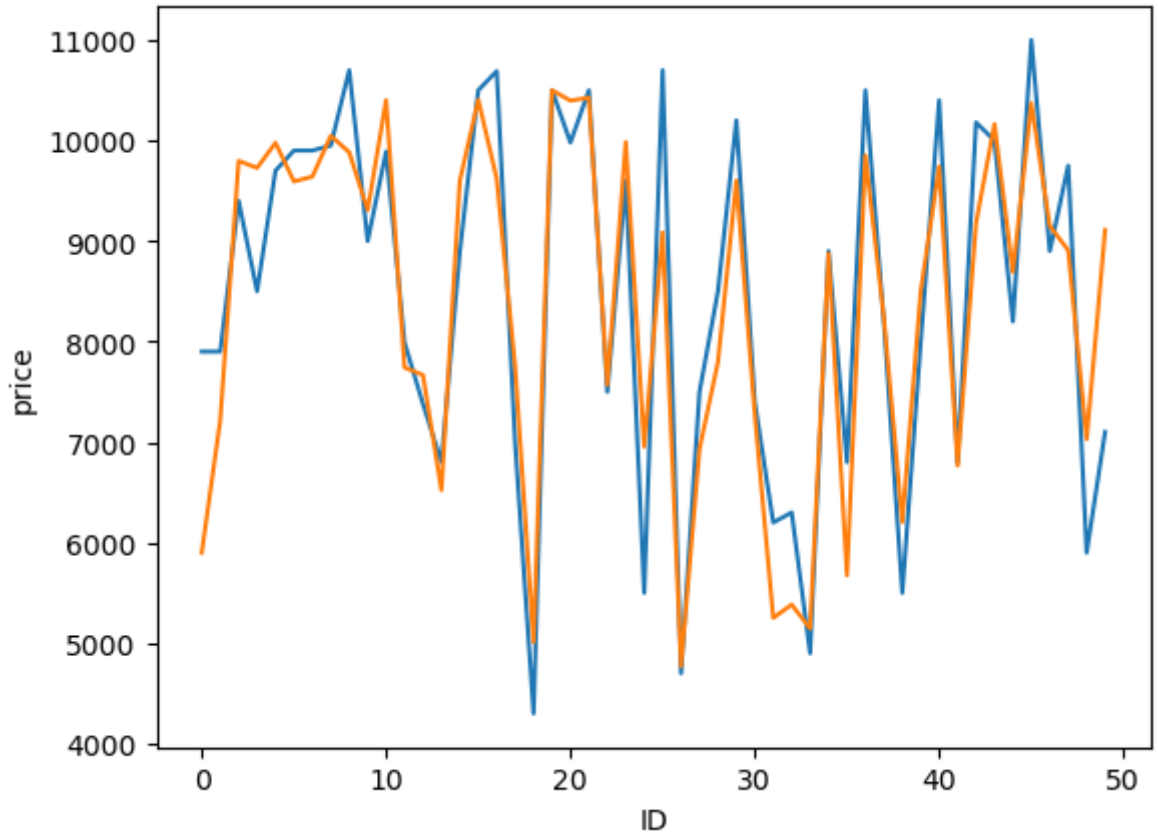
Out[16]:

	index	price	predicted	ID
0	481	7900	5900.920030	0
1	76	7900	7213.135940	1
2	1502	9400	9797.090447	2
3	669	8500	9727.400047	3
4	1409	9700	9976.565475	4
...	...	...	...	...
503	291	10900	10071.495566	503
504	596	5699	6309.976843	504
505	1489	9500	9945.637546	505
506	1436	6990	8314.354246	506
507	575	10900	10386.087480	507

508 rows × 4 columns

In [17]:
```
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[17]: []



In [ ]: