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

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

In [3]:
```
#we are doing ridge
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=data1.drop(['ID','lat','lon'],axis=1)
```

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

Out[5]:

	engine_power	age_in_days	km	previous_owners	price	model_lounge
0	51	882	25000	1	8900	1
3	51	2739	160000	1	6000	1
6	51	731	11600	1	10750	1
7	51	1521	49076	1	9190	1
11	51	366	17500	1	10990	1
...	...	...	...	...	...	...
1528	51	2861	126000	1	5500	1
1529	51	731	22551	1	9900	1
1530	51	670	29000	1	10800	1
1534	74	3835	112000	1	4600	1
1536	51	2557	80750	1	5990	1

1094 rows × 6 columns

In [6]:
```
y=data1['price']
X=data1.drop('price',axis=1)
```

In [7]:
```
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 [8]:

```
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[8]:

```
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 [9]:

```
ridge_regressor.best_params_
```

Out[9]:

```
{'alpha': 30}
```

In [10]:

```
ridge=Ridge(alpha=30)
ridge.fit(X_train,y_train)
y_pred_ridge=ridge.predict(X_test)
```

In [11]:

```
from sklearn.metrics import mean_squared_error
Ridge_Error=mean_squared_error(y_pred_ridge,y_test)
Ridge_Error
```

Out[11]:

```
519771.8129989745
```

In [12]:

```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred_ridge)
```

Out[12]:

```
0.8373030813683994
```

In [13]:

```
Results=pd.DataFrame(columns=['price','predicted'])
Results['price']=y_test
Results['predicted']=y_pred_ridge
Results=Results.reset_index()
Results['ID']=Results.index
Results
```

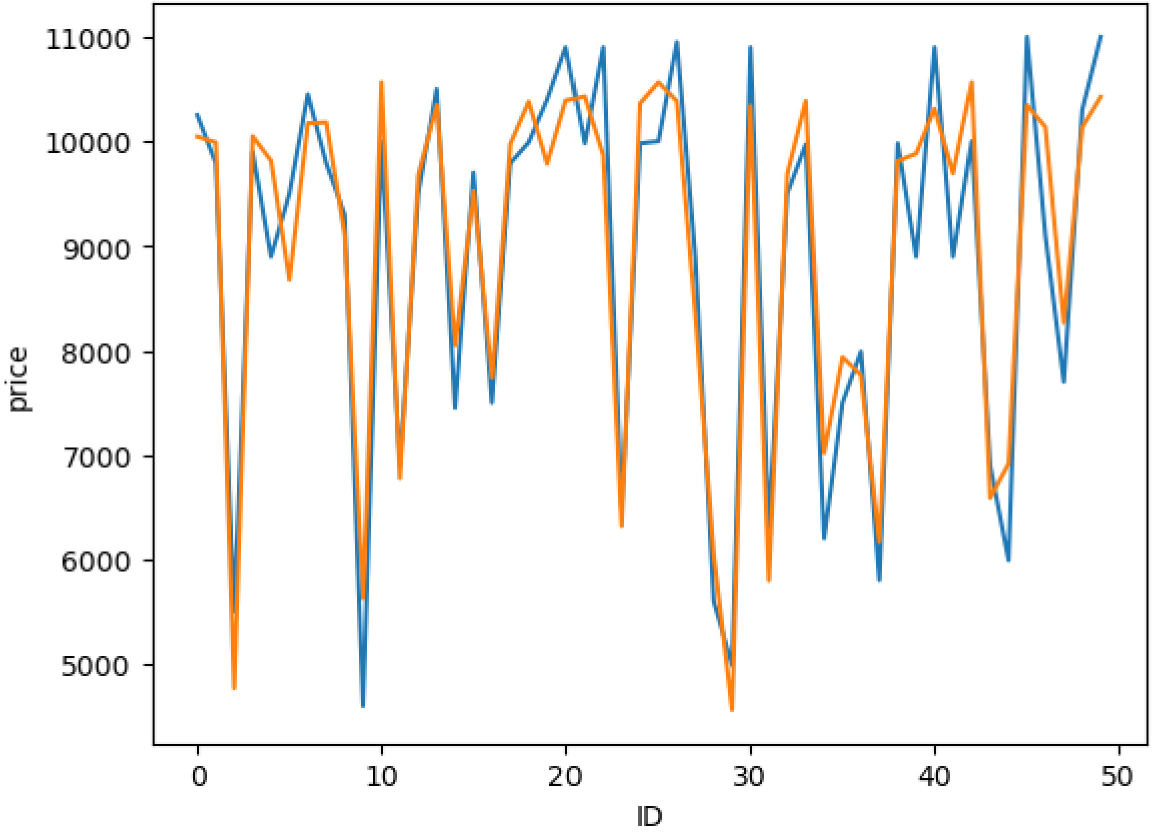
Out[13]:

	index	price	predicted	ID
0	676	10250	10045.347779	0
1	215	9790	9989.171535	1
2	146	5500	4769.099603	2
3	1319	9900	10048.683238	3
4	1041	8900	9813.944798	4
...	...	...	...	...
357	757	6000	5640.378648	357
358	167	10950	10431.681162	358
359	156	8000	8765.506865	359
360	1145	10700	10384.884273	360
361	1393	9400	9929.721685	361

362 rows × 4 columns

In [14]:
```
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[14]:
```
[]
```



In [ ]: