In [114]:
```
import pandas as pd
```

In [115]:
```
data=pd.read_csv("fiat500.csv")
```

In [116]:
```
data
```

Out[116]:

	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 [117]:
```
data=data.drop(['lat','lon','ID',],axis=1)
```

In [118]:
```
data
```

Out[118]:

	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 [119]:
```
data=pd.get_dummies(data)
```

In [120]:
```
data.shape
#data['model'] =data['model'].map({'lounge':1,'pop':2,'sport':3})
```

Out[120]: (1538, 8)

In [121]:
```
data
```

Out[121]:

	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

1538 rows × 8 columns

In [122]:
```
y=data['price']
```

In [123]:
```
x=data.drop('price',axis=1)
```

In [124]:
```
y
```

Out[124]:
```
0       8900
1       8800
2       4200
3       6000
4       5700
        ...
1533    5200
1534    4600
1535    7500
1536    5990
1537    7900
Name: price, Length: 1538, dtype: int64
```

In [125]:
```
!pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in ./anaconda3/lib/python3.10/site-packages (1.2.1)
Requirement already satisfied: joblib>=1.1.1 in ./anaconda3/lib/python3.10/site-packages (from scikit-lear
n) (1.1.1)
Requirement already satisfied: scipy>=1.3.2 in ./anaconda3/lib/python3.10/site-packages (from scikit-learn)
(1.10.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in ./anaconda3/lib/python3.10/site-packages (from sciki
t-learn) (2.2.0)
Requirement already satisfied: numpy>=1.17.3 in ./anaconda3/lib/python3.10/site-packages (from scikit-lear
n) (1.23.5)
```

In [126]:
```
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 [127]:
```
x_test.head(5)
```

Out[127]:

	engine_power	age_in_days	km	previous_owners	model_lounge	model_pop	model_sport
481	51	3197	120000	2	0	1	0
76	62	2101	103000	1	0	1	0
1502	51	670	32473	1	1	0	0
669	51	913	29000	1	1	0	0
1409	51	762	18800	1	1	0	0

In [128]:
```
x_train.shape
```

Out[128]:
```
(1030, 7)
```

In [129]:
```
y_train
```

Out[129]:
```
527      9990
129      9500
602      7590
331      8750
323      9100
        ...
1130    10990
1294     9800
860      5500
1459     9990
1126     8900
Name: price, Length: 1030, dtype: int64
```

In [130]:
```
from sklearn.linear_model import LinearRegression
reg = LinearRegression() #creating object of LinearRegression
reg.fit(x_train,y_train) #training and fitting LR object using training data
```

Out[130]:
```
▾ LinearRegression

LinearRegression()
```

In [131]:
```
ypred=reg.predict(x_test)
```

In [132]:
```
ypred
```

```
array([ 9653.06224923,  7948.63618724,  9704.82523573,  7971.05970955,
       10399.51752022,  9176.43567301,  5803.03205787,  6698.19524313,

        8257.83550573, 10452.95284574,  9948.66454584,  9789.65062843,
       10582.50828537,  7568.91955482,  6804.97705225,  8065.01292384,
       10310.29143419,  8836.34894739,  8390.05091229,  9582.13932508,
        9745.34784981, 10045.45021387, 10294.09872915,  7145.15315349,
        9727.85493167,  6281.78952194,  7901.36245623,  9387.9203723 ,
        5039.55649797,  9351.49777725,  9980.70844784, 10094.79341516,
        6359.24321991,  9856.10227211,  9099.07023804,  5234.05388382,
        5534.45288323,  4495.02309231, 10199.78432943, 10024.87037067,
        5465.58034188,  8520.72057674,  7034.71038647, 10054.65061446,
       10191.12067767,  6008.34860428,  9748.18097947,  9669.4333196 ,
        9145.3756075 ,  9175.66562699, 10087.86753845,  9825.02990067,
        7340.29803785,  5083.8487301 ,  9441.50914802, 10243.05490667,
        5556.42300245, 10676.01945733,  6126.99295838,  9845.16661356,
        9850.77978959,  7840.83596305,  6552.05146566,  9938.82104889,
        8327.79232274,  9119.62204137,  6111.83787367, 10410.00504522,
        6360.97695249,  8601.59209793,  8377.80258216,  9803.81343895,
        8285.09831762, 10091.75635129, 10003.86694939, 10028.60283146,
```

In [133]:
```
from sklearn .metrics import r2_score
r2_score(y_test,ypred)
```

Out[133]:
```
0.8415526986865394
```

In [134]:
```
from sklearn.metrics import mean_squared_error
mean_squared_error(ypred,y_test)
```

Out[134]:
```
581887.727391353
```

In [135]:
```
#from sklearn.metrics import accuracy_score
#accuracy_score(y_test)
```

In [136]:

```
Results= pd.DataFrame(columns=['price','predicted'])
Results['price']=y_test
Results['predicted']=ypred
Results=Results.reset_index()
Results['Id']=Results.index
Results.head(15)
```

Out[136]:

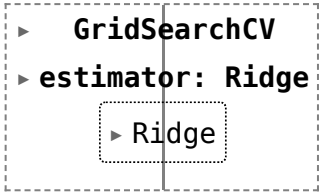
	index	price	predicted	Id
0	481	7900	5867.650338	0
1	76	7900	7133.701423	1
2	1502	9400	9866.357762	2
3	669	8500	9723.288745	3
4	1409	9700	10039.591012	4
5	1414	9900	9654.075826	5
6	1089	9900	9673.145630	6
7	1507	9950	10118.707281	7
8	970	10700	9903.859527	8
9	1198	8999	9351.558284	9
10	1088	9890	10434.349636	10
11	576	7990	7732.262557	11
12	965	7380	7698.672401	12
13	1488	6800	6565.952404	13
14	1432	8900	9662.901035	14

In [137]:
```
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)
```

```
/home/placement/anaconda3/lib/python3.10/site-packages/sklearn/linear_model/_ridge.py:216: LinAlgWarning: I
ll-conditioned matrix (rcond=5.56109e-26): result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
/home/placement/anaconda3/lib/python3.10/site-packages/sklearn/linear_model/_ridge.py:216: LinAlgWarning: I
ll-conditioned matrix (rcond=7.70876e-26): result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
/home/placement/anaconda3/lib/python3.10/site-packages/sklearn/linear_model/_ridge.py:216: LinAlgWarning: I
ll-conditioned matrix (rcond=6.91585e-23): result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
/home/placement/anaconda3/lib/python3.10/site-packages/sklearn/linear_model/_ridge.py:216: LinAlgWarning: I
ll-conditioned matrix (rcond=7.08003e-23): result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
/home/placement/anaconda3/lib/python3.10/site-packages/sklearn/linear_model/_ridge.py:216: LinAlgWarning: I
ll-conditioned matrix (rcond=7.01022e-23): result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
/home/placement/anaconda3/lib/python3.10/site-packages/sklearn/linear_model/_ridge.py:216: LinAlgWarning: I
ll-conditioned matrix (rcond=7.57959e-23): result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
/home/placement/anaconda3/lib/python3.10/site-packages/sklearn/linear_model/_ridge.py:216: LinAlgWarning: I
ll-conditioned matrix (rcond=7.24161e-23): result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
/home/placement/anaconda3/lib/python3.10/site-packages/sklearn/linear_model/_ridge.py:216: LinAlgWarning: I
ll-conditioned matrix (rcond=6.92759e-21): result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
/home/placement/anaconda3/lib/python3.10/site-packages/sklearn/linear_model/_ridge.py:216: LinAlgWarning: I
ll-conditioned matrix (rcond=7.09091e-21): result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
/home/placement/anaconda3/lib/python3.10/site-packages/sklearn/linear_model/_ridge.py:216: LinAlgWarning: I
```

```
ll-conditioned matrix (rcond=7.02112e-21): result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
/home/placement/anaconda3/lib/python3.10/site-packages/sklearn/linear_model/_ridge.py:216: LinAlgWarning: I
ll-conditioned matrix (rcond=7.57414e-21): result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
/home/placement/anaconda3/lib/python3.10/site-packages/sklearn/linear_model/_ridge.py:216: LinAlgWarning: I
ll-conditioned matrix (rcond=7.23284e-21): result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
/home/placement/anaconda3/lib/python3.10/site-packages/sklearn/linear_model/_ridge.py:216: LinAlgWarning: I
ll-conditioned matrix (rcond=6.9277e-17): result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
/home/placement/anaconda3/lib/python3.10/site-packages/sklearn/linear_model/_ridge.py:216: LinAlgWarning: I
ll-conditioned matrix (rcond=7.09099e-17): result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
/home/placement/anaconda3/lib/python3.10/site-packages/sklearn/linear_model/_ridge.py:216: LinAlgWarning: I
ll-conditioned matrix (rcond=7.02123e-17): result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
/home/placement/anaconda3/lib/python3.10/site-packages/sklearn/linear_model/_ridge.py:216: LinAlgWarning: I
ll-conditioned matrix (rcond=7.57407e-17): result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
/home/placement/anaconda3/lib/python3.10/site-packages/sklearn/linear_model/_ridge.py:216: LinAlgWarning: I
ll-conditioned matrix (rcond=7.23274e-17): result may not be accurate.
  return linalg.solve(A, Xy, assume_a="pos", overwrite_a=True).T
```

Out[137]:



In [138]:
```
ridge_regressor.best_params_
```

Out[138]:
```
{'alpha': 30}
```

In [139]:
```
ridge=Ridge(alpha=30)
ridge.fit(x_train,y_train)
y_pred_ridge=ridge.predict(x_test)
```

In [140]:
```
Ridge_Error=mean_squared_error(y_pred_ridge,y_test)
Ridge_Error
```

Out[140]:
```
579521.7970897449
```

In [141]:
```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred_ridge)
```

Out[141]:
```
0.8421969385523054
```

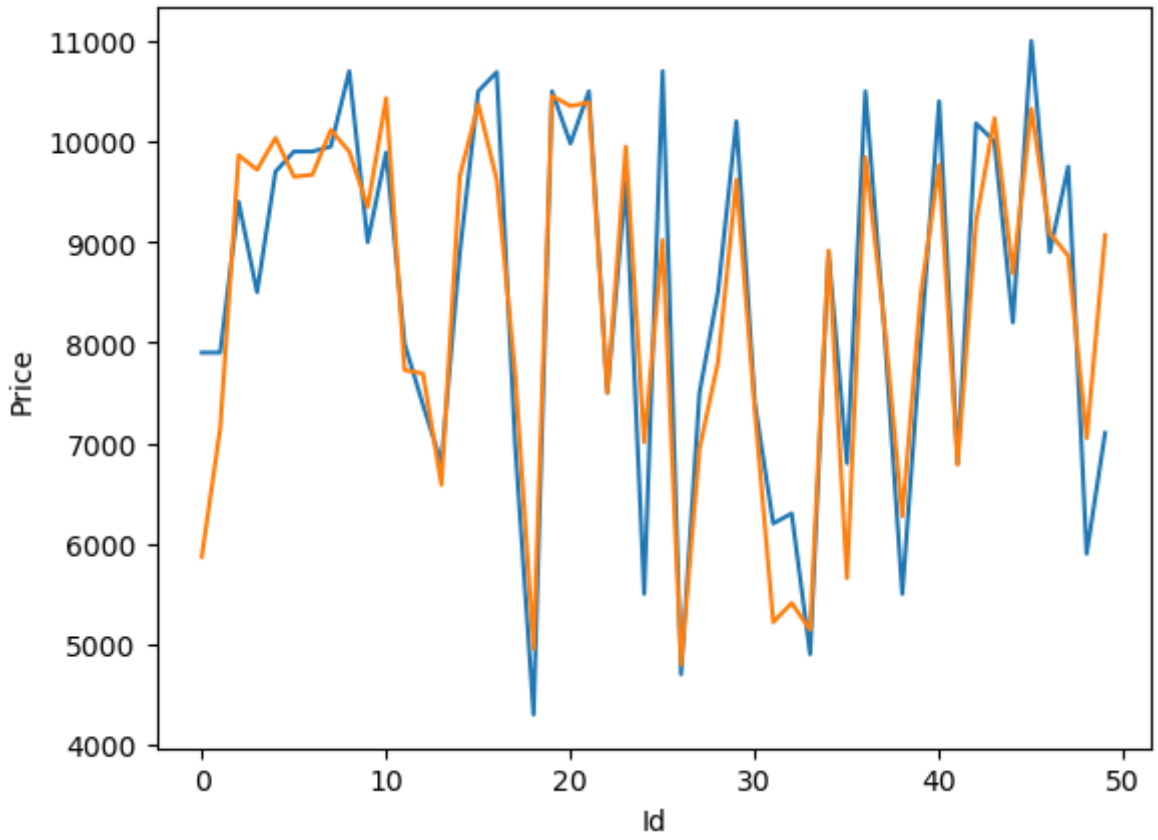
In [145]:
```
Results= pd.DataFrame(columns=['Price','Predicted'])
Results['Price']=y_test
Results['Predicted']=y_pred_ridge
Results=Results.reset_index()
Results['Id']=Results.index
Results.head(10)
```

Out[145]:

	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

In [146]:
```
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[146]: []



In [ ]: