In [230]:
```
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
```

In [231]:
```
data=pd.read_csv("/home/placenent/Downloads/fiat500.csv")
```

In [232]:
```
data.describe()
```

Out[232]:

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price
count	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000
mean	769.500000	51.904421	1650.980494	53396.011704	1.123537	43.541361	11.563428	8576.003901
std	444.126671	3.988023	1289.522278	40046.830723	0.416423	2.133518	2.328190	1939.958641
min	1.000000	51.000000	366.000000	1232.000000	1.000000	36.855839	7.245400	2500.000000
25%	385.250000	51.000000	670.000000	20006.250000	1.000000	41.802990	9.505090	7122.500000
50%	769.500000	51.000000	1035.000000	39031.000000	1.000000	44.394096	11.869260	9000.000000
75%	1153.750000	51.000000	2616.000000	79667.750000	1.000000	45.467960	12.769040	10000.000000
max	1538.000000	77.000000	4658.000000	235000.000000	4.000000	46.795612	18.365520	11100.000000

In [233]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1538 entries, 0 to 1537
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   ID               1538 non-null   int64  
 1   model            1538 non-null   object 
 2   engine_power     1538 non-null   int64  
 3   age_in_days      1538 non-null   int64  
 4   km               1538 non-null   int64  
 5   previous_owners  1538 non-null   int64  
 6   lat              1538 non-null   float64
 7   lon              1538 non-null   float64
 8   price            1538 non-null   int64  
dtypes: float64(2), int64(6), object(1)
memory usage: 108.3+ KB
```

In [234]:

```
#to remove a columns
data1=data.drop(['lat','lon','ID'],axis=1)
data1
```

Out[234]:

	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 [235]:

```
data3=data1.loc[(data.model=='lounge')]
data3
```

Out[235]:

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

1094 rows × 6 columns

In [236]:
```
data=pd.get_dummies(data)
data
```

Out[236]:

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price	model_lounge	model_pop	model_sport
0	1	51	882	25000	1	44.907242	8.611560	8900	1	0	0
1	2	51	1186	32500	1	45.666359	12.241890	8800	0	1	0
2	3	74	4658	142228	1	45.503300	11.417840	4200	0	0	1
3	4	51	2739	160000	1	40.633171	17.634609	6000	1	0	0
4	5	73	3074	106880	1	41.903221	12.495650	5700	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...
1533	1534	51	3712	115280	1	45.069679	7.704920	5200	0	0	1
1534	1535	74	3835	112000	1	45.845692	8.666870	4600	1	0	0
1535	1536	51	2223	60457	1	45.481541	9.413480	7500	0	1	0
1536	1537	51	2557	80750	1	45.000702	7.682270	5990	1	0	0
1537	1538	51	1766	54276	1	40.323410	17.568270	7900	0	1	0

1538 rows × 11 columns

In [237]:
```
data.shape
```

Out[237]:
```
(1538, 11)
```

In [238]:
```
data2=pd.get_dummies(data3)
```

In [239]:
```
data2
```

Out[239]:

	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 [240]:
```
#for checking rows & columns
data2.shape
```

Out[240]:
```
(1094, 6)
```

In [241]:
```
#pridected value we removed from data frame
y=data2['price']
x=data2.drop('price',axis=1)
```

In [242]:
```
y
```

Out[242]:
```
0        8900
3        6000
6       10750
7        9190
11      10990
        ...  
1528     5500
1529     9900
1530    10800
1534     4600
1536     5990
Name: price, Length: 1094, dtype: int64
```

In [243]:
```
#divide the data into testing & training
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 [244]:
```
#to show starting rows
x_test.head(5)
```

Out[244]:

	engine_power	age_in_days	km	previous_owners	model_lounge
676	51	762	18609	1	1
215	51	701	25000	1	1
146	51	4018	152900	1	1
1319	51	731	20025	1	1
1041	51	640	38231	1	1

In [245]:

```
x_train.head(5)
```

Out[245]:

	engine_power	age_in_days	km	previous_owners	model_lounge
441	51	762	36448	1	1
701	51	701	27100	1	1
695	51	3197	51083	1	1
1415	51	670	33000	1	1
404	51	456	14000	1	1

In [246]:

```
y_test.head(5)
```

Out[246]:

```
676     10250
215      9790
146      5500
1319     9900
1041     8900
Name: price, dtype: int64
```

In [247]:

```
y_train.head(5)
```

Out[247]:

```
441      8980
701     10300
695      5880
1415    10490
404      9499
Name: price, dtype: int64
```

In [248]:
```
# Ridge Regression
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[248]:
```
▸   GridSearchCV
▸ estimator: Ridge
      ▸ Ridge
```

In [249]:
```
ridge_regressor.best_params_
```

Out[249]:
```
{'alpha': 30}
```

In [250]:
```
ridge=Ridge(alpha=30)
ridge.fit(x_train,y_train)
y_pred_ridge=ridge.predict(x_test)
```

In [251]:
```
from sklearn.metrics import mean_squared_error
Ridge_Error=mean_squared_error(y_pred_ridge,y_test)
Ridge_Error
```

Out[251]:
```
519771.8129989745
```

In [252]:
```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred_ridge)
```

Out[252]:
```
0.8373030813683994
```

In [253]:

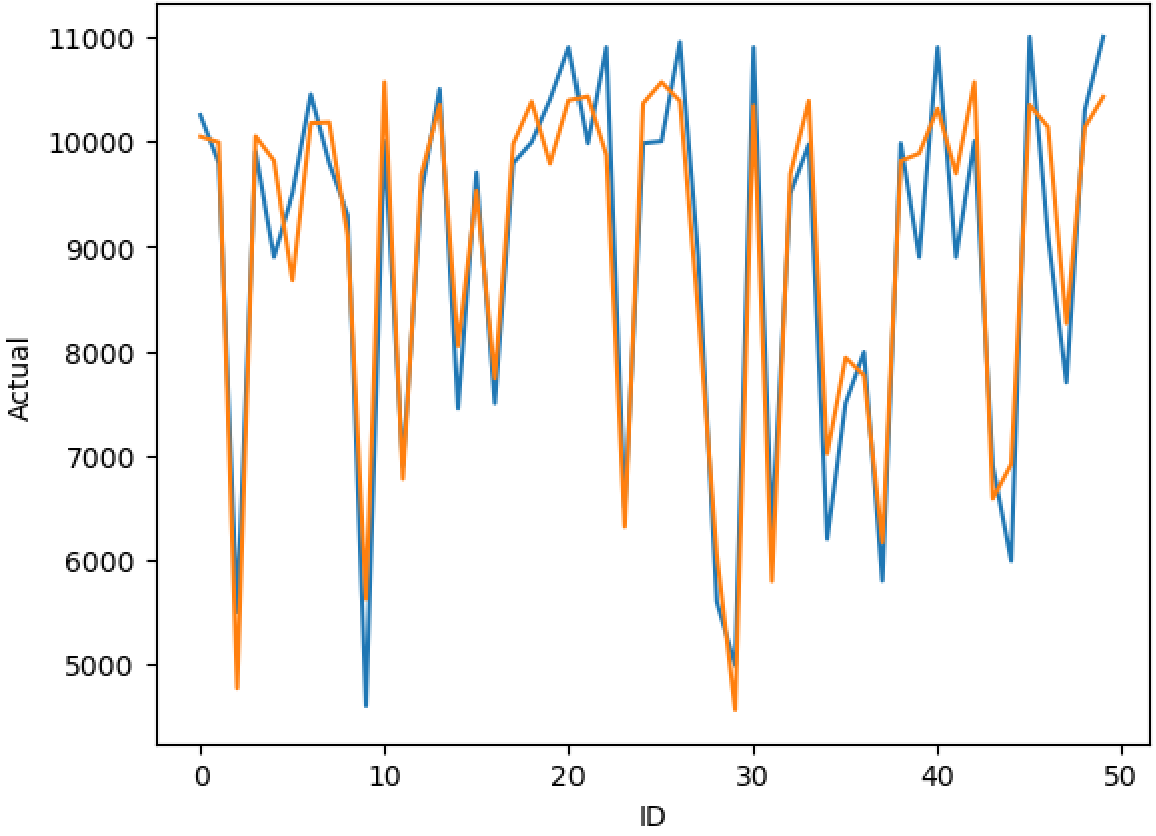
```
Results=pd.DataFrame(columns=['Actual','predicted'])
Results['Actual']=y_test
#Results=pd.DataFrame(columns=['price','predicted'])
#Results['price']=y_test
Results['predicted']=y_pred_ridge
#Results['km']=x_test['km']
Results=Results.reset_index()
Results['ID']=Results.index
Results.head(10)
```

Out[253]:

```
   index  Actual     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
5   1425    9500   8678.143561   5
6    409   10450  10173.797921   6
7    617    9790  10180.627008   7
8   1526    9300   9107.315259   8
9   1010    4600   5625.007407   9
```

```
In [254]: import seaborn as sns
          import matplotlib.pyplot as plt
          sns.lineplot(x='ID',y='Actual',data=Results.head(50))
          sns.lineplot(x='ID',y='predicted',data=Results.head(50))
          plt.plot()
```

```
Out[254]: []
```



In [ ]: