Elastic_net_model

```python
In [35]: import pandas as pd
         import warnings
         warnings.filterwarnings("ignore")
```

```python
In [36]: data=pd.read_csv("/home/placement/Desktop/ramaraju/fiat500.csv")
```

```python
In [37]: data=data.drop(['ID','lat','lon'],axis=1)
```

```python
In [38]: data
```

Out[38]:

	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 [39]:
```python
data['model']=data['model'].map({'lounge':1,'pop':2,'sport':3})
data
```

Out[39]:

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

1538 rows × 6 columns

In [40]:
```python
data1=data.loc[(data.previous_owners==1)]
```

In [41]: `data1`

Out[41]:

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

1389 rows × 6 columns

In [42]:
```python
y=data1['price']
x=data1.drop('price',axis=1)
```

In [43]: `x`

Out[43]:

	model	engine_power	age_in_days	km	previous_owners
0	1	51	882	25000	1
1	2	51	1186	32500	1
2	3	74	4658	142228	1
3	1	51	2739	160000	1
4	2	73	3074	106880	1
...	...	...	...	...	...
1533	3	51	3712	115280	1
1534	1	74	3835	112000	1
1535	2	51	2223	60457	1
1536	1	51	2557	80750	1
1537	2	51	1766	54276	1

1389 rows × 5 columns

In [44]: `y`

Out[44]:
```
0        8900
1        8800
2        4200
3        6000
4        5700
         ...
1533     5200
1534     4600
1535     7500
1536     5990
1537     7900
Name: price, Length: 1389, dtype: int64
```

In [45]:
```python
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 [46]:
```python
x_train.shape
```

Out[46]: (930, 5)

In [47]:
```python
x_test.shape
```

Out[47]: (459, 5)

In [48]:
```python
y_train.shape
```

Out[48]: (930,)

In [49]:
```python
y_test.shape
```

Out[49]: (459,)

In [50]:
```python
x_train.head()
```

Out[50]:

	model	engine_power	age_in_days	km	previous_owners
915	1	51	397	17081	1
12	1	51	456	18450	1
638	1	51	397	21276	1
190	1	51	821	19000	1
701	1	51	701	27100	1

In [51]: `x_test.head()`

Out[51]:

	model	engine_power	age_in_days	km	previous_owners
625	1	51	3347	148000	1
187	1	51	4322	117000	1
279	2	51	4322	120000	1
734	2	51	974	12500	1
315	1	51	1096	37000	1

In [52]: `y_train.head()`

Out[52]:
```
915    10900
12      9700
638    10850
190     9990
701    10300
Name: price, dtype: int64
```

In [53]: `y_test.head()`

Out[53]:
```
625     5400
187     5399
279     4900
734    10500
315     9300
Name: price, dtype: int64
```

In [54]:
```python
from sklearn.linear_model import ElasticNet
from sklearn.model_selection import GridSearchCV

elastic = ElasticNet()

parameters = {'alpha': [1e-15, 1e-10, 1e-8, 1e-4, 1e-3,1e-2, 1, 5, 10, 20]}

elastic_regressor = GridSearchCV(elastic, parameters)

elastic_regressor.fit(x_train, y_train)
```
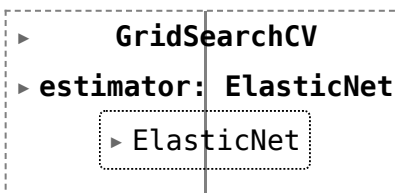
Out[54]:
```
         ▸        GridSearchCV
      ▸ estimator: ElasticNet
            ▸ ElasticNet
```

In [55]:
```python
elastic_regressor.best_params_
```

Out[55]: {'alpha': 1e-15}

In [56]:
```python
elastic=ElasticNet(alpha=1e-15)
elastic.fit(x_train,y_train)
y_pred_elastic=elastic.predict(x_test)
```

In [57]:
```python
from sklearn.metrics import mean_squared_error
Elasticnet_Error=mean_squared_error(y_pred_elastic,y_test)
Elasticnet_Error
```

Out[57]: 522589.1692194676

In [58]:
```python
from sklearn.metrics import r2_score
r2_score(y_test,y_pred_elastic)
```
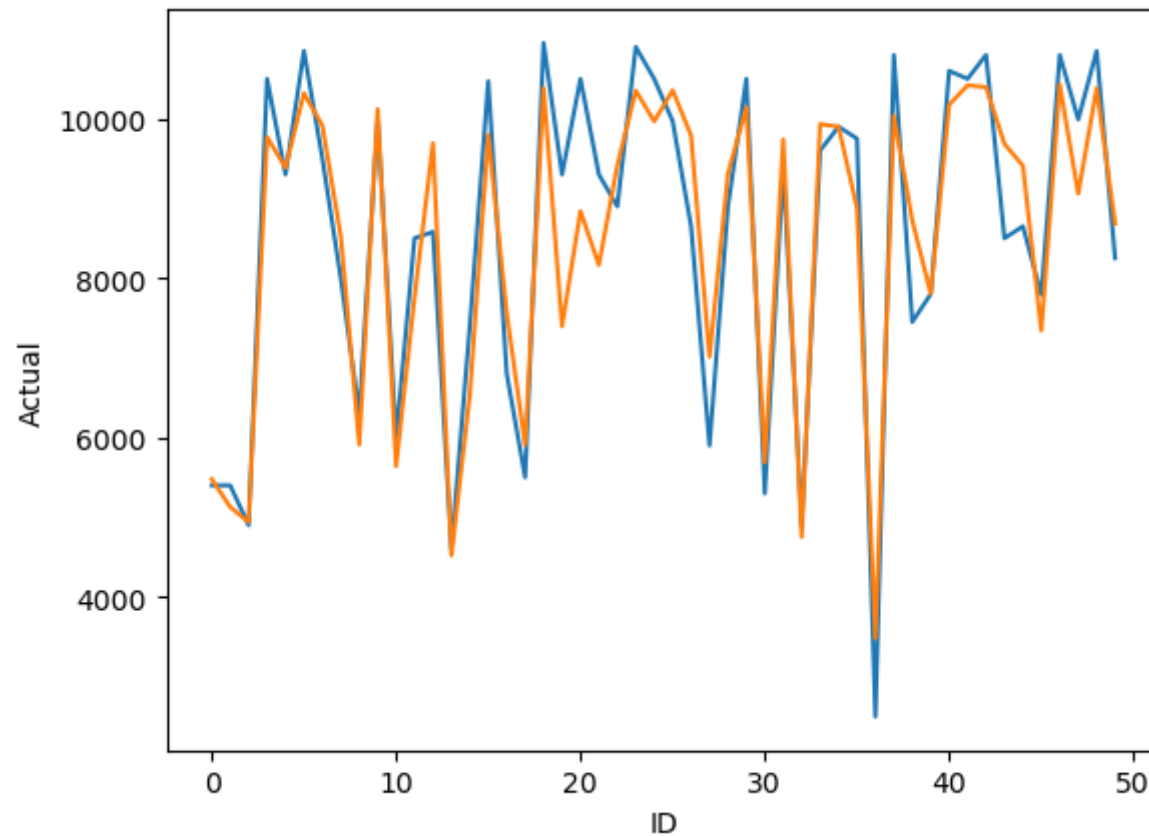
Out[58]: 0.8582526737355334

In [59]:
```python
Results=pd.DataFrame(columns=['Actual','predicted'])
Results['Actual']=y_test
Results['predicted']=y_pred_elastic
Results=Results.reset_index()
Results['ID']=Results.index
Results.head(10)
```

Out[59]:

	index	Actual	predicted	ID
0	625	5400	5478.082470	0
1	187	5399	5128.749813	1
2	279	4900	4939.964669	2
3	734	10500	9770.938056	3
4	315	9300	9383.407921	4
5	652	10850	10319.804281	5
6	1472	9500	9912.760894	6
7	619	7999	8526.411840	7
8	992	6300	5910.610353	8
9	1154	10000	10119.997990	9

In [60]:
```python
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[60]: []

In [ ]: