In [120]:

```
import pandas as pd
```

In [121]:

```
data=pd.read_csv("/home/placement/Desktop/divyasri/fiat500.csv")
```

In [122]:

```
data.describe()
```

Out[122]:

	ID	engine_power	age_in_days	km	previous_owners	lat
count	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
std	444.126671	3.988023	1289.522278	40046.830723	0.416423	2.133518
min	1.000000	51.000000	366.000000	1232.000000	1.000000	36.855839
25%	385.250000	51.000000	670.000000	20006.250000	1.000000	41.802990
50%	769.500000	51.000000	1035.000000	39031.000000	1.000000	44.394096
75%	1153.750000	51.000000	2616.000000	79667.750000	1.000000	45.467960
max	1538.000000	77.000000	4658.000000	235000.000000	4.000000	46.795612

In [123]:

```
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 [124]:

```
data.head(10)
```

Out[124]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	pr
0	1	lounge	51	882	25000	1	44.907242	8.611560	8
1	2	pop	51	1186	32500	1	45.666359	12.241890	8
2	3	sport	74	4658	142228	1	45.503300	11.417840	4
3	4	lounge	51	2739	160000	1	40.633171	17.634609	6
4	5	pop	73	3074	106880	1	41.903221	12.495650	5
5	6	pop	74	3623	70225	1	45.000702	7.682270	7
6	7	lounge	51	731	11600	1	44.907242	8.611560	10
7	8	lounge	51	1521	49076	1	41.903221	12.495650	9
8	9	sport	73	4049	76000	1	45.548000	11.549470	5
9	10	sport	51	3653	89000	1	45.438301	10.991700	6

In [125]:

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

In [126]:

```
data1
```

Out[126]:

	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 [127]:

```
data2=data1.loc[(data1.model=='lounge')]
```

In [128]:

```
data2
```

Out[128]:

	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 [129]:

```
data1=pd.get_dummies(data1)
```

In [130]:

```
data2
```

Out[130]:

	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 [131]:

```
data1.shape
```

Out[131]:

```
(1538, 8)
```

In [132]:

```
y=data1['price']
```

In [133]:

```
y
```

Out[133]:

```
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 [134]:

```
x=data1.drop(['price'],axis=1)
```

In [135]:

```
x
```

Out[135]:

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

1538 rows × 7 columns

In [136]:

```
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 [137]:

```
x_test.head(5)
```

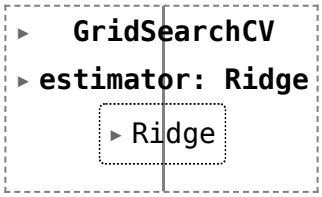
Out[137]:

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

In [138]:

```
from sklearn.model_selection import GridSearchCV #for ridge
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[138]:



In [139]:

```
import warnings
warnings.filterwarnings("ignore")
```

In [140]:

```
ridge_regressor.best_params_ #alpha value or constant
```

Out[140]:

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

In [141]:

```
ridge=Ridge(alpha=30)
ridge.fit(x_train,y_train)
y_pred_ridge=ridge.predict(x_test) #predicted value
```

In [142]:

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

Out[142]:

```
579521.7970897449
```

In [143]:

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

Out[143]:

```
0.8421969385523054
```

In [144]:

```
data2=data.loc[(data.model=='lounge')]
```

In [145]:

```
data2
```

Out[145]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lo
0	1	lounge	51	882	25000	1	44.907242	8.61156
3	4	lounge	51	2739	160000	1	40.633171	17.63460
6	7	lounge	51	731	11600	1	44.907242	8.61156
7	8	lounge	51	1521	49076	1	41.903221	12.49565
11	12	lounge	51	366	17500	1	45.069679	7.70492
...	...	...	...	...	...	...	...	.
1528	1529	lounge	51	2861	126000	1	43.841980	10.51531
1529	1530	lounge	51	731	22551	1	38.122070	13.36112
1530	1531	lounge	51	670	29000	1	45.764648	8.99450
1534	1535	lounge	74	3835	112000	1	45.845692	8.66687
1536	1537	lounge	51	2557	80750	1	45.000702	7.68227

1094 rows × 9 columns

In [146]:

```
data3=data2.drop(['ID','lat','lon'],axis=1)
```

In [147]:

```
data3
```

Out[147]:

	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 [148]:

```
data3=pd.get_dummies(data3)
```

In [149]:

```
data3
```

Out[149]:

	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 [150]:

```
y=data3['price']
```

In [151]:

```
y
```

Out[151]:

```
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 [152]:

```
x=data3.drop(['price'],axis=1)
```

In [153]:

```
x
```

Out[153]:

	engine_power	age_in_days	km	previous_owners	model_lounge
0	51	882	25000	1	1
3	51	2739	160000	1	1
6	51	731	11600	1	1
7	51	1521	49076	1	1
11	51	366	17500	1	1
...	...	...	...	...	...
1528	51	2861	126000	1	1
1529	51	731	22551	1	1
1530	51	670	29000	1	1
1534	74	3835	112000	1	1
1536	51	2557	80750	1	1

1094 rows × 5 columns

In [154]:

```
data3.shape
```

Out[154]:

```
(1094, 6)
```

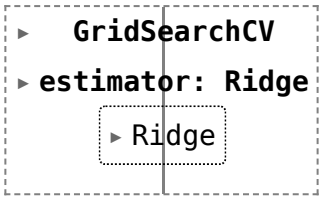
In [155]:

```
from sklearn.model_selection import train_test_split #spliting into training  and t
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=42)
```

In [156]:

```
from sklearn.model_selection import GridSearchCV #for ridge
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[156]:



In [157]:

```
ridge_regressor.best_params_ #alpha value or constant
```

Out[157]:

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

In [158]:

```
ridge=Ridge(alpha=30)
ridge.fit(x_train,y_train)
y_pred_ridge=ridge.predict(x_test) #predicted value
```

In [159]:

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

Out[159]:

```
519771.8129989745
```

In [160]:

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

Out[160]:

```
0.8373030813683994
```

In [162]:

```
Results=pd.DataFrame(columns=['Actual','Predicted'])
Results['Actual']=y_test
Results['Predicted']=y_pred_ridge
Results=Results.reset_index()
Results['ID']=Results.index #replaces id with index number
Results.head(10)
```

Out[162]:

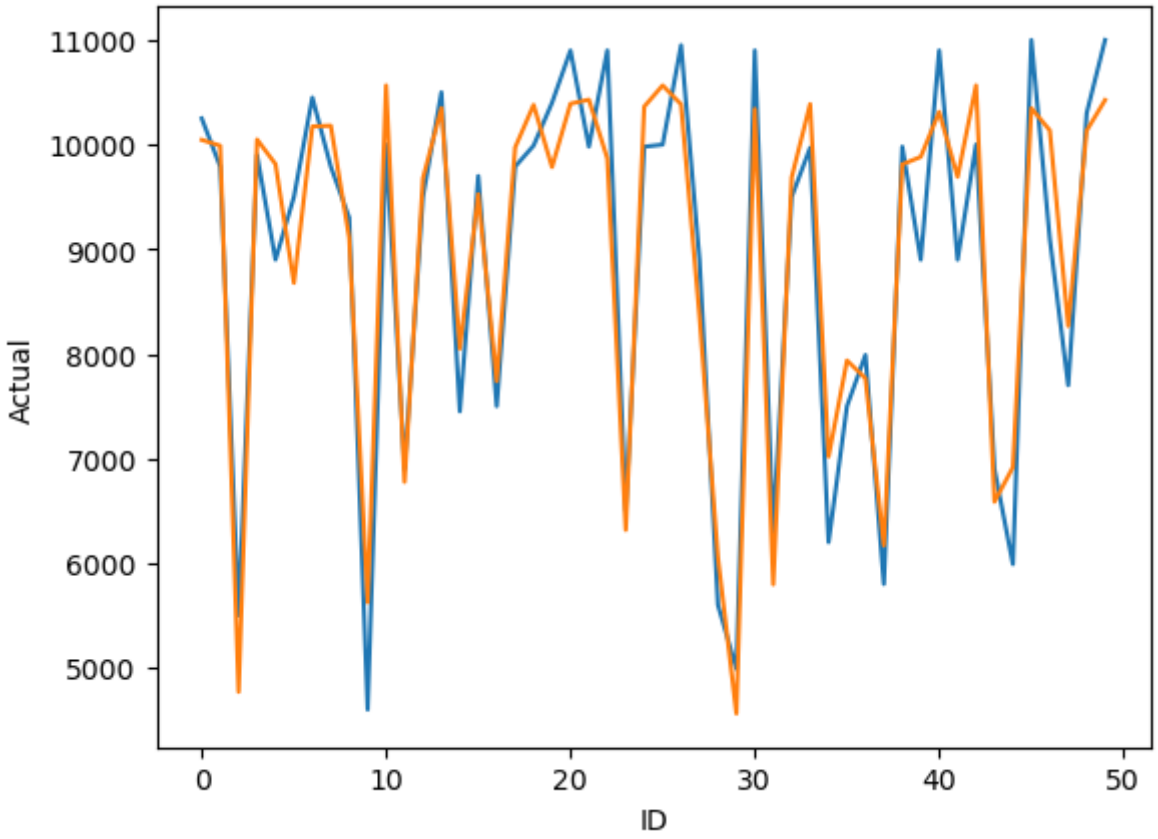
	index	Actual	Predicted	ID
0	676	10250	10045.347779	0
1	215	9790	9989.171535	1
2	146	5500	4769.099603	2
3	1319	9900	10048.683238	3
4	1041	8900	9813.944798	4
5	1425	9500	8678.143561	5
6	409	10450	10173.797921	6
7	617	9790	10180.627008	7
8	1526	9300	9107.315259	8
9	1010	4600	5625.007407	9

In [171]:

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.lineplot(x='ID',y='Actual',data=Results.head(50)) #red is actual
sns.lineplot(x='ID',y='Predicted',data=Results.head(50)) #blue is predicted
```

Out[171]:

```
<Axes: xlabel='ID', ylabel='Actual'>
```



In [ ]:

```
```