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

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

In [3]:
```
data.head()
```

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 |
| **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 |

In [4]:
```
data1=data.loc[(data.previous_owners==1)]
```

In [5]:
```
data1
```

Out[5]:

| | 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 |

1389 rows × 9 columns

In [6]:
```
data.describe()
```

Out[6]:

| | 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 [7]:
```
data=pd.get_dummies(data)
```

In [8]:
```
data
```

Out[8]:

| | 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 [9]:
```
data1=data.drop(['ID','lat','lon'],axis=1)
```

In [10]:
```
data1
```

Out[10]:

| | 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 [11]:
```
data.shape
```

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

In [12]:
```
y=data1['price']
x=data1.drop('price',axis=1)
```

In [13]:
```
y
```

Out[13]:
```
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 [14]:
```
x
```

Out[14]:

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

1538 rows × 7 columns

In [15]:
```
data.info
```

Out[15]:
```
<bound method DataFrame.info of         ID  engine_power  age_in_days      km  previous_owners        lat  \
0        1            51          882   25000                1  44.907242
1        2            51         1186   32500                1  45.666359
2        3            74         4658  142228                1  45.503300
3        4            51         2739  160000                1  40.633171
4        5            73         3074  106880                1  41.903221
...    ...           ...          ...     ...              ...        ...
1533  1534            51         3712  115280                1  45.069679
1534  1535            74         3835  112000                1  45.845692
1535  1536            51         2223   60457                1  45.481541
1536  1537            51         2557   80750                1  45.000702
1537  1538            51         1766   54276                1  40.323410

            lon  price  model_lounge  model_pop  model_sport
0      8.611560   8900             1          0            0
1     12.241890   8800             0          1            0
2     11.417840   4200             0          0            1
3     17.634609   6000             1          0            0
4     12.495650   5700             0          1            0
...         ...    ...           ...        ...          ...
1533   7.704920   5200             0          0            1
1534   8.666870   4600             1          0            0
1535   9.413480   7500             0          1            0
1536   7.682270   5990             1          0            0
1537  17.568270   7900             0          1            0

[1538 rows x 11 columns]>
```

In [16]:
```
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 [17]:
```
x_test.head()
```

Out[17]:

| | 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 [18]:
```
y_test.head()
```

Out[18]:
```
481     7900
76      7900
1502    9400
669     8500
1409    9700
Name: price, dtype: int64
```

In [19]:
```
y_train.head()
```

Out[19]:
```
527    9990
129    9500
602    7590
331    8750
323    9100
Name: price, dtype: int64
```

In [20]:
```
y_test.head()
```

Out[20]:
```
481     7900
76      7900
1502    9400
669     8500
1409    9700
Name: price, dtype: int64
```

In [21]:
```
from sklearn.linear_model import LinearRegression
reg=LinearRegression() #creating object of LinearRegression
reg.fit(x_train,y_train) #training are fitting LR object using training data
```

Out[21]:
```
▾ LinearRegression

LinearRegression()
```

In [22]:

```
ypred=reg.predict(x_test)
ypred
```

```
 8840.08397206,  9916.27565791, 10287.45603992,  9964.3213269 ,
 8403.51255128,  9345.81907605,  8521.46225147,  9743.68712672,
 9791.34520178,  9779.16293972,  6753.27416058,  7354.16762745,
 8760.24542762,  9923.66596418,  9812.92276721, 10466.90125415,
 8163.46726237,  6659.46839415,  9987.65677522,  8866.7826029 ,
 9952.37340054, 10187.72427693, 10231.39378767, 10091.11325493,
 9365.98570732, 10009.10088406,  9141.00566394, 10099.11667176,
 7803.77049829,  6009.84398185,  8800.33824151, 10237.60733785,
 5609.98366311, 10097.61555355,  9684.99946572,  7644.67379732,
 9276.37891542,  7371.5492091 , 10287.98873148, 10067.26428381,
10552.64805598,  9966.72383894, 10068.46126756,  6232.53552963,
10584.55044373,  9965.98687522, 10529.44404458,  9602.67646085,
 9665.77720284,  6186.06948587,  8073.87436253, 10345.58323918,
 6344.74803956,  7361.62678204, 10058.57116223,  6792.219309  ,
 7897.72464823,  5261.45936067,  4540.24137423,  8709.36468047,
 6882.0117409 ,  7406.73353952,  6795.61189392,  7047.27998963,
 9945.33400083,  8856.93910595,  9378.02074127, 10389.561154  ,
10092.46332921, 10381.52000388,  9723.92466625,  5996.3331428 ,
 9786.14866981,  7708.49649098,  5583.48163469,  4932.92788329,
```

In [23]:
```
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[23]:
```
▸ GridSearchCV
▸ estimator: Ridge
   ▸ Ridge
```

In [24]:
```
ridge_regressor.best_params_
```

Out[24]:
```
{'alpha': 30}
```

In [25]:
```
ridge=Ridge(alpha=30)
ridge.fit(x_train,y_train)
y_pred_ridge=ridge.predict(x_test)
```

In [26]:
```
ypred=ridge_regressor.predict(x_test)
```

In [27]:
```
ypred
```

```
         7880.06692968,  5421.58854259,  9908.29872913, 10387.14911484,
         9677.04064086,  8839.64921963,  7758.04496364,  4272.30358351,
         9879.24583245, 10338.32300002,  5727.64824413, 10176.30365568,
         9496.76688181,  7988.08471853,  5547.3732031 ,  9891.06127373,
        10434.86357188,  6372.31423723,  9587.41260499,  9566.83486068,
        10319.17531038,  9522.2341334 ,  9810.96209247,  9614.20063768,
         6793.94641746,  7934.18123669, 10386.9650737 , 10348.55568749,
         7374.24832772,  9962.8313526 , 10427.75958382, 10537.05691733,
        10307.86764216, 10057.84775154,  9522.79576023,  7736.32759234,
         9304.38969384, 10047.72953114, 10001.16290595,  9982.03783683,
         9356.45889265,  9557.79360856,  9751.29744779,  9816.27815842,
         8772.11589722,  6246.5681686 ,  6297.83377181,  8210.43368425,
         8609.44775082,  6555.65279459,  6842.30381123,  5501.4318499 ,
         8113.64452407,  9844.22280324,  7768.73012027,  9871.53450863,
        10142.9208104 ,  5807.87769524,  9831.08379287, 10040.2716485 ,
         8019.70650821,  4516.41557946, 10577.27619954,  3821.65817325,
         9948.97058812, 10508.26237361,  5721.78144051,  5476.739495  ,
        10392.79176695,  6788.56652042,  8948.54811289, 10424.84703341,
         9450.82424662,  9972.71312863,  8547.62184364,  7954.86597122,
        10396.93269263,  5351.3569159 ,  9895.97517221, 10199.69230161,
```

In [28]:
```
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import ElasticNet

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[28]:
```
▸ GridSearchCV
▸ estimator: ElasticNet
    ▸ ElasticNet
```

In [29]:
```
elastic_regressor.best_params_
```

Out[29]: {'alpha': 0.01}

In [30]:
```
elastic=ElasticNet(alpha=30)
elastic.fit(x_train,y_train)
y_pred_elastic=elastic.predict(x_test)
```

In [31]:
```
from sklearn.metrics import mean_squared_error
mean_squared_error(ypred,y_test)
```

Out[31]: 579521.7970897449

In [32]:
```
from sklearn.metrics import mean_squared_error
Ridge_Error=mean_squared_error(y_pred_ridge,y_test)
Ridge_Error
```

Out[32]: 579521.7970897449

In [33]:
```
from sklearn.metrics import mean_squared_error
Elastic_Error=mean_squared_error(y_pred_elastic,y_test)
Elastic_Error
```

Out[33]: 580334.1755711779

In [34]:
```
from sklearn.metrics import r2_score
r2_score(y_test,ypred)  #ytest=actual price,ypred=predicted price
```

Out[34]: 0.8421969385523054

In [35]:
```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred_ridge)  #ytest=actual price,ypred=predicted price
```

Out[35]: 0.8421969385523054

In [36]:
```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred_elastic)
```

Out[36]:
```
0.8419757289065801
```

In [ ]: