In [1]:
```
import pandas as pd
```

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

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

Out[3]:
```
                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 [4]:
```
data.head(10)
```

Out[4]:

	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
5	6	pop	74	3623	70225	1	45.000702	7.682270	7900
6	7	lounge	51	731	11600	1	44.907242	8.611560	10750
7	8	lounge	51	1521	49076	1	41.903221	12.495650	9190
8	9	sport	73	4049	76000	1	45.548000	11.549470	5600
9	10	sport	51	3653	89000	1	45.438301	10.991700	6000

In [5]:
```
data1=data.drop(['lat','ID'],axis=1)
```

In [6]:
```
data1
```

Out[6]:

	model	engine_power	age_in_days	km	previous_owners	lon	price
0	lounge	51	882	25000	1	8.611560	8900
1	pop	51	1186	32500	1	12.241890	8800
2	sport	74	4658	142228	1	11.417840	4200
3	lounge	51	2739	160000	1	17.634609	6000
4	pop	73	3074	106880	1	12.495650	5700
...	...	...	...	...	...	...	...
1533	sport	51	3712	115280	1	7.704920	5200
1534	lounge	74	3835	112000	1	8.666870	4600
1535	pop	51	2223	60457	1	9.413480	7500
1536	lounge	51	2557	80750	1	7.682270	5990
1537	pop	51	1766	54276	1	17.568270	7900

1538 rows × 7 columns

In [7]:
```
data1=pd.get_dummies(data1)
```

In [8]:
```
data1.shape
```

Out[8]:
```
(1538, 9)
```

In [9]:

```
data1
```

Out[9]:

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

1538 rows × 9 columns

In [10]:

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

In [11]:
```
y
```

Out[11]:
```
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 [12]:
```
#!pip3 install scikit-learn
```

In [13]:
```
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 [14]:
```
x_test.head(5)
```

Out[14]:

	engine_power	age_in_days	km	previous_owners	lon	model_lounge	model_pop	model_sport
481	51	3197	120000	2	18.167629	0	1	0
76	62	2101	103000	1	8.644440	0	1	0
1502	51	670	32473	1	14.208810	1	0	0
669	51	913	29000	1	8.946250	1	0	0
1409	51	762	18800	1	9.928310	1	0	0

In [15]:

```
y_test.head(10)
```

Out[15]:

```
481      7900
76       7900
1502     9400
669      8500
1409     9700
1414     9900
1089     9900
1507     9950
970     10700
1198     8999
Name: price, dtype: int64
```

In [16]:

```
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[16]:

```
LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [17]:

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

In [18]:
```
ypred
```

Out[18]:
```
array([ 5744.90385381,  7193.08174957,  9821.47324371,  9773.81447482,
       10071.07869345,  9641.63868511,  9649.51258809, 10157.61850988,
        9847.94985009,  9309.71516629, 10482.47565268,  7779.30441243,
        7673.74056422,  6505.72766141,  9619.50289515, 10411.90973558,
        9670.05131682,  7699.50515859,  4877.54154624, 10452.07538228,
       10421.3266699 , 10430.13589594,  7496.76330979,  9961.06726287,
        7050.37003331,  8984.92007278,  4828.00807556,  6998.7171637 ,
        7878.93692686,  9631.98929627,  7384.85817189,  5253.54744257,
        5442.11536379,  5042.43511877,  8999.34734771,  5716.87620176,
        9811.44986065,  8238.65183735,  6305.39688435,  8391.28310387,
        9759.44418847,  6769.38469788,  9169.54303692, 10154.06276081,
        8635.65508981, 10298.38883653,  9048.46307993,  8857.07716202,
        7059.85863786,  9061.62298677,  9451.88481593, 10320.57884532,
       10145.81082462,  6753.76039054,  9742.26723839,  9421.71535673,
        9510.86366496, 10468.92783011,  9789.73337471,  7192.86742992,
       10015.85984307,  7050.64774752,  9908.93831182,  7147.76311105,
        6439.07590176,  9965.68603975,  9828.44710958,  8554.15295046,
        8504.01172278,  6382.30585686,  7747.58493918,  6835.01812084,
        8314.98295981, 10435.98168268,  7353.94396099,  8551.12772532,
```

In [19]:
```
from sklearn.metrics import r2_score
r2_score(y_test,ypred)
```

Out[19]:
```
0.8426736468631826
```

In [20]:
```
from sklearn.metrics import mean_squared_error
l=mean_squared_error(ypred,y_test)
```

In [21]:
```
l
```

Out[21]:
```
577771.1158642182
```

In [22]:
```
n=577771.1158642182
print(n**(1/2))
```

```
760.1125678899266
```

In [23]:
```
ypred
```

Out[23]:
```
array([ 5744.90385381,  7193.08174957,  9821.47324371,  9773.81447482,
       10071.07869345,  9641.63868511,  9649.51258809, 10157.61850988,
        9847.94985009,  9309.71516629, 10482.47565268,  7779.30441243,
        7673.74056422,  6505.72766141,  9619.50289515, 10411.90973558,
        9670.05131682,  7699.50515859,  4877.54154624, 10452.07538228,
       10421.3266699 , 10430.13589594,  7496.76330979,  9961.06726287,
        7050.37003331,  8984.92007278,  4828.00807556,  6998.7171637 ,
        7878.93692686,  9631.98929627,  7384.85817189,  5253.54744257,
        5442.11536379,  5042.43511877,  8999.34734771,  5716.87620176,
        9811.44986065,  8238.65183735,  6305.39688435,  8391.28310387,
        9759.44418847,  6769.38469788,  9169.54303692, 10154.06276081,
        8635.65508981, 10298.38883653,  9048.46307993,  8857.07716202,
        7059.85863786,  9061.62298677,  9451.88481593, 10320.57884532,
       10145.81082462,  6753.76039054,  9742.26723839,  9421.71535673,
        9510.86366496, 10468.92783011,  9789.73337471,  7192.86742992,
       10015.85984307,  7050.64774752,  9908.93831182,  7147.76311105,
        6439.07590176,  9965.68603975,  9828.44710958,  8554.15295046,
        8504.01172278,  6382.30585686,  7747.58493918,  6835.01812084,
        8314.98295981, 10435.98168268,  7353.94396099,  8551.12772532,
```

In [24]:

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

Out[24]:

```
      price     predicted
 481   7900   5744.903854
  76   7900   7193.081750
1502   9400   9821.473244
 669   8500   9773.814475
1409   9700  10071.078693
1414   9900   9641.638685
1089   9900   9649.512588
1507   9950  10157.618510
 970  10700   9847.949850
1198   8999   9309.715166
1088   9890  10482.475653
 576   7990   7779.304412
 965   7380   7673.740564
1488   6800   6505.727661
1432   8900   9619.502895
```

In [26]:

```
Results['DIFF']=Results.apply(lambda row:row.price-row.predicted,axis=1)
```

```
In [27]: Results
```

```
Out[27]:
             price      predicted          DIFF
       481    7900    5744.903854   2155.096146
        76    7900    7193.081750    706.918250
      1502    9400    9821.473244   -421.473244
       669    8500    9773.814475  -1273.814475
      1409    9700   10071.078693   -371.078693
       ...     ...            ...           ...
       291   10900   10039.516366    860.483634
       596    5699    6282.912490   -583.912490
      1489    9500    9991.170743   -491.170743
      1436    6990    8331.810958  -1341.810958
       575   10900   10357.579392    542.420608

508 rows × 3 columns
```

```
In [ ]:
```