In [36]: ```python
import pandas as pd
```

In [37]: ```python
data=pd.read_csv("/home/placement/Downloads/fiat500.csv") #reading file
```

In [38]: ```python
import warnings
warnings.filterwarnings('ignore')
```

In [39]: ```python
data.describe()
```

Out[39]:

	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 [40]: ```python
data1=data.drop(['ID','lat','lon'],axis=1) #removing unwanted columns
```

In [41]: `data1`

Out[41]:

|  | 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 [42]: `data2=data.loc[(data.model=='lounge')]`

In [43]: `data2`

Out[43]:

|  | 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 |
| **3** | 4 | lounge | 51 | 2739 | 160000 | 1 | 40.633171 | 17.634609 | 6000 |
| **6** | 7 | lounge | 51 | 731 | 11600 | 1 | 44.907242 | 8.611560 | 10750 |
| **7** | 8 | lounge | 51 | 1521 | 49076 | 1 | 41.903221 | 12.495650 | 9190 |
| **11** | 12 | lounge | 51 | 366 | 17500 | 1 | 45.069679 | 7.704920 | 10990 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **1528** | 1529 | lounge | 51 | 2861 | 126000 | 1 | 43.841980 | 10.515310 | 5500 |
| **1529** | 1530 | lounge | 51 | 731 | 22551 | 1 | 38.122070 | 13.361120 | 9900 |
| **1530** | 1531 | lounge | 51 | 670 | 29000 | 1 | 45.764648 | 8.994500 | 10800 |
| **1534** | 1535 | lounge | 74 | 3835 | 112000 | 1 | 45.845692 | 8.666870 | 4600 |
| **1536** | 1537 | lounge | 51 | 2557 | 80750 | 1 | 45.000702 | 7.682270 | 5990 |

1094 rows × 9 columns

In [44]: `data1=pd.get_dummies(data) #dummies`

In [45]: `data1`

Out[45]:

|  | 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 [46]: `data1.shape`

Out[46]: `(1538, 11)`

In [47]:
```python
y=data1['price']
x=data1.drop('price',axis=1) #removing price
```

In [48]: `y`

Out[48]:
```
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 [49]: `x`

Out[49]:

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

1538 rows × 10 columns

In [50]: `#!pip install scikit-learn #is to install sklearn`

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

Out[52]:

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	model_lounge	model_pop	model_sport
481	482	51	3197	120000	2	40.174702	18.167629	0	1	0
76	77	62	2101	103000	1	45.797859	8.644440	0	1	0
1502	1503	51	670	32473	1	41.107880	14.208810	1	0	0
669	670	51	913	29000	1	45.778591	8.946250	1	0	0
1409	1410	51	762	18800	1	45.538689	9.928310	1	0	0

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

Out[53]:
```
481      7900
76       7900
1502     9400
669      8500
1409     9700
Name: price, dtype: int64
```

In [54]:
```python
x_train.head()
```

Out[54]:

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	model_lounge	model_pop	model_sport
527	528	51	425	13111	1	45.022388	7.58602	1	0	0
129	130	51	1127	21400	1	44.332531	7.54592	1	0	0
602	603	51	2039	57039	1	40.748241	14.52835	0	1	0
331	332	51	1155	40700	1	42.143860	12.54016	1	0	0
323	324	51	425	16783	1	41.903221	12.49565	1	0	0

In [55]:
```python
y_train.head()
```

Out[55]:
```
527    9990
129    9500
602    7590
331    8750
323    9100
Name: price, dtype: int64
```

In [56]:
```python
from sklearn.linear_model import LinearRegression
reg=LinearRegression()
```

In [57]:
```python
reg.fit(x_train,y_train)
```

Out[57]:
```
▼ LinearRegression
LinearRegression()
```

In [58]:
```python
ypred=reg.predict(x_test)
```

In [59]: ypred

```
         9951.43993841, 10514.34035875,  9980.65273426,  4951.73872798,
         7295.05742766,  9562.10832139,  9907.19829915,  5552.67362672,
         9997.10882667,  5143.10046698,  8426.58953237,  7496.77298917,
         7788.58793224,  9652.49535927,  8621.5141248 , 10372.95686992,
         7121.85593538,  9807.45515506,  8055.56244116,  7460.56572873,
        10280.14093316, 10476.37888795,  5412.86365493,  9159.69570754,
         9570.24451488, 10631.67197425, 10191.16722391,  9246.40215298,
         6018.78924109,  9615.95984878, 10520.04528994,  8915.29037502,
         8227.95081736,  9888.28916316,  9469.40296538,  9984.09733556,
        10405.22999333, 10510.19490347,  9686.14366923,  8096.65870148,
        10493.03817153, 10386.55525134,  8807.23351267,  8280.41931335,
         6855.39153451, 10228.17167191,  4741.4113269 ,  8847.08211454,
         5777.60377085, 10140.98307313,  8867.00881496, 10115.19879685,
         9581.17073619, 10485.09114495, 10213.14966572,  9750.35350756,
         9670.90795615,  6722.33834114,  9564.21623605,  8616.86940955,
         6677.30400889, 10373.25962634, 10171.86282428, 10201.5589761 ,
         5933.51739776,  8804.77586124,  9636.29957689,  5533.81938983,
         8355.20070696,  9637.44491292,  4357.89955793,  9958.32874977,
         9779.38235089,  7862.66884791, 10129.74800911, 10459.05959668,
        10231.78369129,  6764.79775959,  6588.0328491 , 10326.6417079 ,
```

In [60]:
```python
from sklearn.metrics import r2_score
r2_score (y_test,ypred)
```

Out[60]: 0.8428319728488683

In [61]:
```python
from sklearn.metrics import mean_squared_error
mean_squared_error(ypred,y_test)
```

Out[61]: 577189.6736608233

In [62]:
```python
import math
a=577189.6736608233
print(math.sqrt(a))
```

759.7300005007195

In [63]: ypred

```
       8830.1166215 ,  9985.15913274, 10382.71023744,  9468.0263143 ,
       8968.98195986, 10125.34089439, 10458.2651463 , 10278.08804577,
       9671.6787843 ,  9329.13714009, 10314.76913411,  5264.56339184,
       9702.21408416,  6171.43279386,  8986.33052433, 10216.19272235,
       9147.3967606 ,  9826.31604212,  8298.03251468,  8311.88829156,
       7566.99918427, 10585.88056004, 10365.38883807, 10134.48005849,
      10264.36282573,  6915.44935844,  9653.38748676, 10541.2624204 ,
       9560.92995691,  8036.36881073,  9719.26456362,  7852.08945425,
      10512.80396135,  9252.12747599,  5726.61394851,  6730.65776903,
       8210.66023805, 10515.83562762, 10009.26844663,  9700.98953567,
      10713.27840286,  7459.58763216,  6787.00375841,  8104.3079721 ,
      10354.65990513,  8853.07922772,  8364.21417446,  9715.8036014 ,

       9717.9187229 , 10052.70924095, 10413.05171654,  7110.93559148,
       9677.88306475,  6332.86407216,  7919.52949511,  9426.02785254,
       4922.86060346,  9277.4623152 , 10063.26423893, 10214.94768556,
       6434.36110141,  9768.93083352,  8988.12723422,  5227.79616128,
       5571.30440259,  4530.22663156, 10314.20984456, 10113.71226961,
       5360.80939407,  8630.38080588,  6927.12547116,  9966.57630777,
      10153.96314322,  5873.22039039,  9650.12316742,  9596.30428421,
       9210.62472152,  9107.96642063, 10077.12276256,  9887.85750934,
```

In [64]:
```python
Results=pd.DataFrame(columns=['price','predicted'])
Results['price']=y_test
Results['predicted']=ypred
Results=Results.reset_index()
Results['Id']=Results.index
Results.head(15)
```

Out[64]:

	index	price	predicted	Id
0	481	7900	5819.193088	0
1	76	7900	7248.829142	1
2	1502	9400	9741.893697	2
3	669	8500	9798.980331	3
4	1409	9700	10055.006246	4
5	1414	9900	9551.495568	5
6	1089	9900	9758.017439	6
7	1507	9950	10122.977837	7
8	970	10700	9654.966181	8
9	1198	8999	9251.140326	9
10	1088	9890	10478.095123	10
11	576	7990	7807.300526	11
12	965	7380	7705.158738	12
13	1488	6800	6295.632449	13
14	1432	8900	9545.404863	14

In [ ]: