In [14]:

```
!pip3 install pandas
```

Requirement already satisfied: pandas in ./anaconda3/lib/python3.10/si
te-packages (1.5.3)
Requirement already satisfied: python-dateutil>=2.8.1 in ./anaconda3/l
ib/python3.10/site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in ./anaconda3/lib/python
3.10/site-packages (from pandas) (2022.7)
Requirement already satisfied: numpy>=1.21.0 in ./anaconda3/lib/python
3.10/site-packages (from pandas) (1.23.5)
Requirement already satisfied: six>=1.5 in ./anaconda3/lib/python3.10/
site-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)

In [15]:

```
import pandas as pd
import numpy as np
```

In [16]:

```
data=pd.read_csv("/home/placement/Downloads/sid.csv")
```

In [17]:

```
data.describe()
```

Out[17]:

	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 [18]:

```python
data.head()
```

Out[18]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	pri
0	1	lounge	51	882	25000	1	44.907242	8.611560	890
1	2	pop	51	1186	32500	1	45.666359	12.241890	880
2	3	sport	74	4658	142228	1	45.503300	11.417840	420
3	4	lounge	51	2739	160000	1	40.633171	17.634609	600
4	5	pop	73	3074	106880	1	41.903221	12.495650	570

In [19]:

```python
data.tail(3)
```

Out[19]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon
1535	1536	pop	51	2223	60457	1	45.481541	9.41348
1536	1537	lounge	51	2557	80750	1	45.000702	7.68227
1537	1538	pop	51	1766	54276	1	40.323410	17.56827

In [20]:

```python
data.tail(1234)
```

Out[20]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price
304	305	pop	51	701	37950	1	41.770081	12.22711	7950
305	306	lounge	51	456	12600	1	44.904430	8.19994	10200
306	307	lounge	51	397	21886	1	45.512569	10.32901	10850
307	308	pop	51	3896	60000	1	45.621410	12.45399	4900
308	309	lounge	51	2282	50400	1	44.907242	8.61156	7900
...	...	...	...	...	...	...	...	...	...
1533	1534	sport	51	3712	115280	1	45.069679	7.70492	5200
1534	1535	lounge	74	3835	112000	1	45.845692	8.66687	4600
1535	1536	pop	51	2223	60457	1	45.481541	9.41348	7500
1536	1537	lounge	51	2557	80750	1	45.000702	7.68227	5990

In [21]:

```python
list(data.columns)
```

Out[21]:

```
['ID',
 'model',
 'engine_power',
 'age_in_days',
 'km',
 'previous_owners',
 'lat',
 'lon',
 'price']
```

In [22]:

```python
data['price'].sum()
```

Out[22]:

```
13189894
```

In [23]:

```python
data['previous_owners'].unique()
```

Out[23]:

```
array([1, 2, 3, 4])
```

In [24]:

```python
data['price'].unique()
```

Out[24]:

```
array([ 8900,  8800,  4200,  6000,  5700,  7900, 10750,  9190,  5600,
        8950, 10990,  9700,  4800,  9300,  9500,  5250,  7990,  7300,
       10500,  6990, 10600, 10200,  9990, 10800,  6800,  4950, 10640,
        5900,  5200,  9790,  5000,  8990,  7200,  9950,  9000,  4890,
       10900,  5999, 10400,  7500,  4900,  4300,  6999,  5990,  5500,
        7450,  8250,  9800,  9900,  4490,  7400, 10700,  7800, 10050,
        4799,  8100,  5800,  9390,  7490,  9970,  8980, 10465,  5950,
        8500,  8790, 10000,  9400,  6100,  6500, 10650, 10950, 11000,
        7700,  6300, 10250,  4990,  8200, 10550,  6900,  6700,  9490,
       10279, 11090,  8000,  5400,  8700, 10280,  4500,  4250,  9450,
        9590,  9600,  5399, 10670,  5300, 10850,  7600,  5100,  6600,
        9435, 10300,  4390,  8390, 10470,  3390,  9980,  9850,  5490,
        7950,  9750,  4600, 10999,  9100,  6200,  8400,  8750,  8290,
        7100,  9999,  8999,  5699,  8579,  6350,  8600,  9979,  8580,
        9499, 10450, 10590,  4690,  6599,  4400,  9200,  8850,  4700,
        8350,  6490,  7999,  8899,  7000,  6400,  8300,  4450, 10490,
        8499, 10499,  9480,  5850,  7480,  6290,  8450,  4299,  4399,
       10790,  7590,  9899,  9840,  9890,  4790,  9290,  6699,  4999,
       11100,  8650,  5499,  5880,  6499, 10870, 10690,  7495,  5799,
       10100,  5450, 10350,  3990,  8190,  6190, 10390,  7390,  7790,
       10399,  3500,  3600,  8399,  6890,  2500,  7190,  7380,  3900,
        9780,  9879,  7699,  9550,  7885, 10180,  3800,  9699,  7479,
        5790,  6250,  7350,  9299,  8490,  8799, 10890,  7799,  3950,
        6790,  4000,  5550,  6450,  9690,  6799,  2900,  6950,  5199,
        8890,  8979,  3850,  5290,  4100,  4750])
```

In [25]:

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

Out[25]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon
0	1	lounge	51	882	25000	1	44.907242	8.611560
3	4	lounge	51	2739	160000	1	40.633171	17.634609
6	7	lounge	51	731	11600	1	44.907242	8.611560
7	8	lounge	51	1521	49076	1	41.903221	12.495650
11	12	lounge	51	366	17500	1	45.069679	7.704920
...	...	...	...	...	...	...	...	.
1528	1529	lounge	51	2861	126000	1	43.841980	10.515310
1529	1530	lounge	51	731	22551	1	38.122070	13.361120
1530	1531	lounge	51	670	29000	1	45.764648	8.994500
1534	1535	lounge	74	3835	112000	1	45.845692	8.666870
1536	1537	lounge	51	2557	80750	1	45.000702	7.682270

1094 rows × 9 columns

In [26]:

```python
data3=data.loc[(data.km<50000)]
data
```

Out[26]:

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

1538 rows × 9 columns

In [27]:

```python
data
```

Out[27]:

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

1538 rows × 9 columns

In [28]:

```python
data.head()
```

Out[28]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	pri
0	1	lounge	51	882	25000	1	44.907242	8.611560	89(
1	2	pop	51	1186	32500	1	45.666359	12.241890	88(
2	3	sport	74	4658	142228	1	45.503300	11.417840	42(
3	4	lounge	51	2739	160000	1	40.633171	17.634609	60(
4	5	pop	73	3074	106880	1	41.903221	12.495650	57(

In [29]:

```python
data.describe()
```

Out[29]:

	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 [30]:

```python
data4=data.loc[(data.km<50000)&(data.model=='pop')]
data4
```

Out[30]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon
1	2	pop	51	1186	32500	1	45.666359	12.24189
10	11	pop	51	790	43286	1	40.871429	14.43896
36	37	pop	51	852	17000	1	45.505161	8.93910
39	40	pop	51	1858	13373	1	41.903221	12.49565
41	42	pop	51	609	28500	1	45.746021	9.04997
...	...	...	...	...	...	...	...	...
1472	1473	pop	51	731	17000	1	45.069679	7.70492
1489	1490	pop	51	366	22148	1	45.707249	11.47760
1505	1506	pop	51	1886	11000	1	44.813751	7.24540
1517	1518	pop	51	366	16100	1	44.692520	10.10396
1527	1528	pop	51	517	3000	1	40.748241	14.52835

158 rows × 9 columns

In [31]:

```python
data5=data.loc[(data.model=='pop')^(data.model=='lounge')]
data5
```

Out[31]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon
0	1	lounge	51	882	25000	1	44.907242	8.611156
1	2	pop	51	1186	32500	1	45.666359	12.241890
3	4	lounge	51	2739	160000	1	40.633171	17.634609
4	5	pop	73	3074	106880	1	41.903221	12.495650
5	6	pop	74	3623	70225	1	45.000702	7.682270
...	...	...	...	...	...	...	...	...
1532	1533	pop	51	1917	52008	1	45.548000	11.549470
1534	1535	lounge	74	3835	112000	1	45.845692	8.666870
1535	1536	pop	51	2223	60457	1	45.481541	9.413480
1536	1537	lounge	51	2557	80750	1	45.000702	7.682270
1537	1538	pop	51	1766	54276	1	40.323410	17.568270

1452 rows × 9 columns

In [32]:

```python
data.price
```

Out[32]:

```
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
```
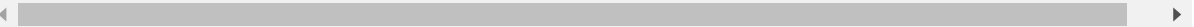
In [33]:

```python
dat=pd.read_csv("/home/placement/Downloads/sid.csv")
```

In [38]:

```python
dat.head()
```

Out[38]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	pri
0	1	lounge	51	882	25000	1	44.907242	8.611560	89(
1	2	pop	51	1186	32500	1	45.666359	12.241890	88(
2	3	sport	74	4658	142228	1	45.503300	11.417840	42(
3	4	lounge	51	2739	160000	1	40.633171	17.634609	60(
4	5	pop	73	3074	106880	1	41.903221	12.495650	57(

In [35]:

```
cor_mat=dat.corr()
cor_mat
```

/tmp/ipykernel_6800/821658463.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  cor_mat=dat.corr()

Out[35]:

	ID	engine_power	age_in_days	km	previous_owners	lat
ID	1.000000	-0.034059	-0.060753	-0.006537	0.007803	-0.058207
engine_power	-0.034059	1.000000	0.319190	0.285495	-0.005030	0.005721
age_in_days	-0.060753	0.319190	1.000000	0.833890	0.075775	0.062982
km	-0.006537	0.285495	0.833890	1.000000	0.097539	0.035519
previous_owners	0.007803	-0.005030	0.075775	0.097539	1.000000	0.001697
lat	-0.058207	0.005721	0.062982	0.035519	0.001697	1.000000
lon	0.058941	-0.005032	-0.042667	0.004839	-0.026836	-0.766646
price	0.028516	-0.277235	-0.893328	-0.859373	-0.076274	-0.011733

In [40]:

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

In [41]:

```
data
```

Out[41]:

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

1538 rows × 9 columns

In [ ]: