In [6]:
```
import pandas as pd
import numpy as np
```

In [8]:
```
data=pd.read_csv("/home/placement/Desktop/nio/fiat500.csv")# reading the file to analyze the data
```

In [ ]:
```
```

In [9]:
```
data.head()
```

Out[9]:

	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 [10]:
```
data.groupby(['price']).count()#counting the price
```

Out[10]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon
price								
2500	1	1	1	1	1	1	1	1
2900	1	1	1	1	1	1	1	1
3390	1	1	1	1	1	1	1	1
3500	1	1	1	1	1	1	1	1
3600	1	1	1	1	1	1	1	1
...	...	...	...	...	...	...	...	...
10990	9	9	9	9	9	9	9	9
10999	5	5	5	5	5	5	5	5
11000	13	13	13	13	13	13	13	13
11090	2	2	2	2	2	2	2	2
11100	1	1	1	1	1	1	1	1

222 rows × 8 columns

In [11]:
```
data.describe()
```

Out[11]:

	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 [12]:
```
list(data.columns)
```

Out[12]:
```
['ID',
 'model',
 'engine_power',
 'age_in_days',
 'km',
 'previous_owners',
 'lat',
 'lon',
 'price']
```

In [13]:
```
data['price'].sum()
```

Out[13]:
```
13189894
```

In [14]:
```
data['price'].unique()
```

Out[14]:
```
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 [15]:

```
data2=data.loc[(data.model=='lounge')] # replacing the data with lounge
data2
```

Out[15]:

	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 [16]:

```
data3=data.loc[(data.km<5000)]#analysing the data and modifying it as we need
data3
```

Out[16]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price
29	30	lounge	51	670	4000	1	41.349751	13.35332	9500
178	179	lounge	51	1066	1232	1	45.467960	9.18178	9900
523	524	lounge	51	425	3600	1	40.695560	14.48085	9490
847	848	lounge	51	487	2790	1	45.766979	11.73840	10400
1527	1528	pop	51	517	3000	1	40.748241	14.52835	9999

In [17]:

```
data3=data.loc[(data.km<5000)]#analysing the data and modifying it as we need
data3
```

Out[17]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price
29	30	lounge	51	670	4000	1	41.349751	13.35332	9500
178	179	lounge	51	1066	1232	1	45.467960	9.18178	9900
523	524	lounge	51	425	3600	1	40.695560	14.48085	9490
847	848	lounge	51	487	2790	1	45.766979	11.73840	10400
1527	1528	pop	51	517	3000	1	40.748241	14.52835	9999

In [18]:

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

Out[18]:

	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
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
...	...	...	...	...	...	...	...	...	...
1532	1533	pop	51	1917	52008	1	45.548000	11.549470	9900
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

1452 rows × 9 columns

In [19]:

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

In [ ]:

```
data=pd.read_csv("/home/placement/Desktop/nio/fiat500.csv")#readin the file to analyze the data
```

In [ ]:

```

```

In [139]:

```
data.describe()
```

Out[139]:

	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 [140]:

```
list(data)
```

Out[140]:

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

In [141]:

```
data.head()
```

Out[141]:

	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 [142]:
```
data.tail(18)
```

Out[142]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price
1520	1521	lounge	51	1035	15000	1	41.903221	12.49565	10990
1521	1522	lounge	51	3774	85000	1	44.294300	9.67444	4000
1522	1523	lounge	51	366	14618	1	45.707249	11.47760	10500
1523	1524	pop	51	2251	79800	1	45.512051	10.42701	6450
1524	1525	pop	51	2192	53300	1	40.609531	14.98093	7900
1525	1526	lounge	51	790	41870	1	45.707249	11.47760	9500
1526	1527	lounge	51	1705	23600	1	38.122070	13.36112	9300
1527	1528	pop	51	517	3000	1	40.748241	14.52835	9999
1528	1529	lounge	51	2861	126000	1	43.841980	10.51531	5500
1529	1530	lounge	51	731	22551	1	38.122070	13.36112	9900
1530	1531	lounge	51	670	29000	1	45.764648	8.99450	10800
1531	1532	sport	73	4505	127000	1	45.528511	9.59323	4750
1532	1533	pop	51	1917	52008	1	45.548000	11.54947	9900
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
1537	1538	pop	51	1766	54276	1	40.323410	17.56827	7900

In [143]:
```
data.shape#describe the rows and coums
```

Out[143]:
```
(1538, 9)
```

In [144]:

```
data["model"]=data['model'].map({'lounge':1,'pop':2,'sport':3})# replace the strings with integers in the de
```

In [145]:

```
data
```

Out[145]:

```
        ID  model  engine_power  age_in_days      km  previous_owners        lat        lon  price
   0     1      1            51          882   25000                1  44.907242   8.611560   8900
   1     2      2            51         1186   32500                1  45.666359  12.241890   8800
   2     3      3            74         4658  142228                1  45.503300  11.417840   4200
   3     4      1            51         2739  160000                1  40.633171  17.634609   6000
   4     5      2            73         3074  106880                1  41.903221  12.495650   5700
 ...   ...    ...           ...          ...     ...              ...        ...        ...    ...
1533  1534      3            51         3712  115280                1  45.069679   7.704920   5200
1534  1535      1            74         3835  112000                1  45.845692   8.666870   4600
1535  1536      2            51         2223   60457                1  45.481541   9.413480   7500
1536  1537      1            51         2557   80750                1  45.000702   7.682270   5990
1537  1538      2            51         1766   54276                1  40.323410  17.568270   7900

1538 rows × 9 columns
```

In [146]:

```
data
```

Out[146]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price
0	1	1	51	882	25000	1	44.907242	8.611560	8900
1	2	2	51	1186	32500	1	45.666359	12.241890	8800
2	3	3	74	4658	142228	1	45.503300	11.417840	4200
3	4	1	51	2739	160000	1	40.633171	17.634609	6000
4	5	2	73	3074	106880	1	41.903221	12.495650	5700
...	...	...	...	...	...	...	...	...	...
1533	1534	3	51	3712	115280	1	45.069679	7.704920	5200
1534	1535	1	74	3835	112000	1	45.845692	8.666870	4600
1535	1536	2	51	2223	60457	1	45.481541	9.413480	7500
1536	1537	1	51	2557	80750	1	45.000702	7.682270	5990
1537	1538	2	51	1766	54276	1	40.323410	17.568270	7900

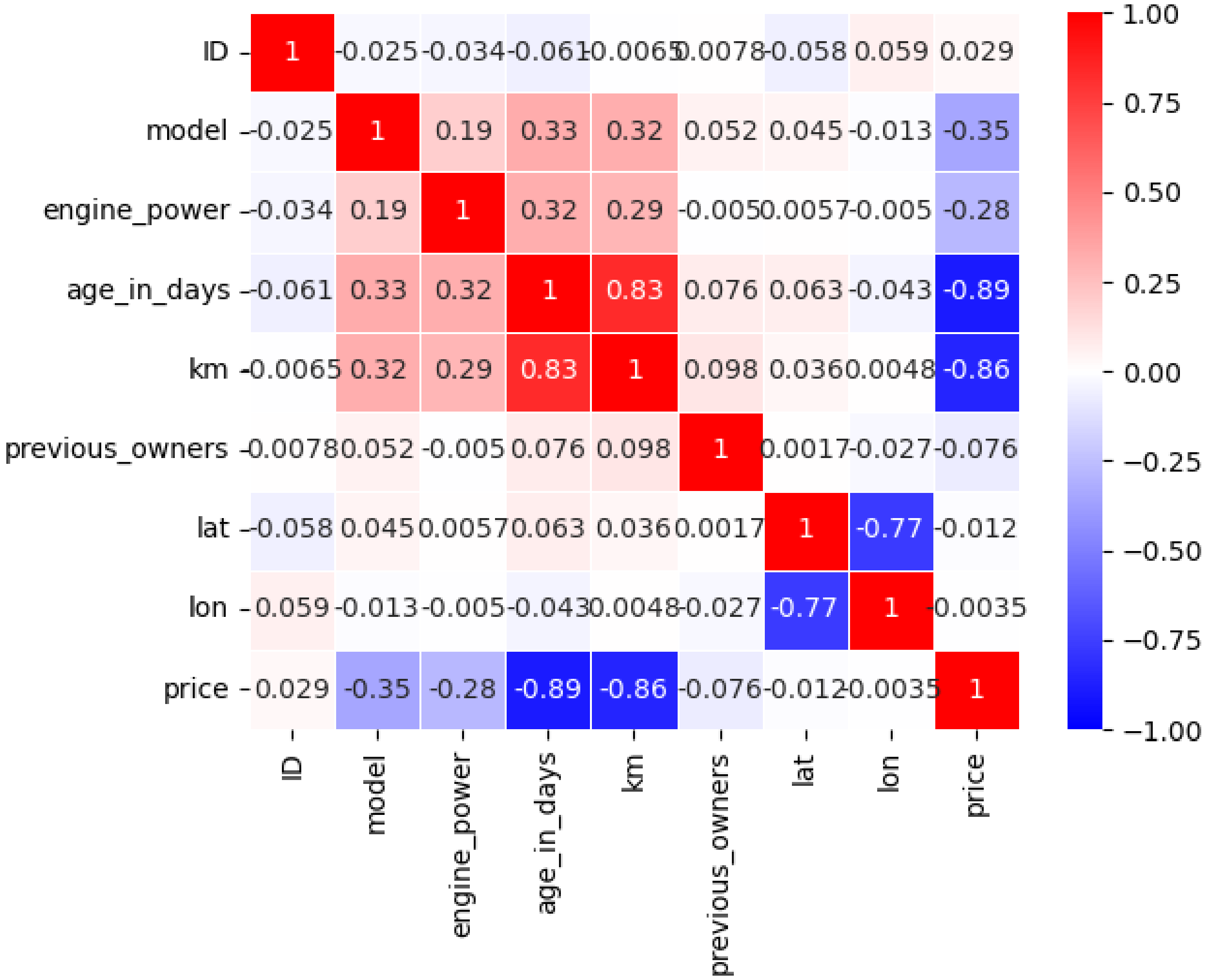
1538 rows × 9 columns

```
In [147]: cor=data.corr()
          cor  #used for correlation of the data
```

```
Out[147]:
                       ID     model  engine_power  age_in_days        km  previous_owners       lat       lon     price
              ID  1.000000 -0.024740     -0.034059    -0.060753 -0.006537         0.007803 -0.058207  0.058941  0.028516
           model -0.024740  1.000000      0.189906     0.326508  0.319580         0.052480  0.044901 -0.013200 -0.349885
    engine_power -0.034059  0.189906      1.000000     0.319190  0.285495        -0.005030  0.005721 -0.005032 -0.277235
     age_in_days -0.060753  0.326508      0.319190     1.000000  0.833890         0.075775  0.062982 -0.042667 -0.893328
              km -0.006537  0.319580      0.285495     0.833890  1.000000         0.097539  0.035519  0.004839 -0.859373
 previous_owners  0.007803  0.052480     -0.005030     0.075775  0.097539         1.000000  0.001697 -0.026836 -0.076274
             lat -0.058207  0.044901      0.005721     0.062982  0.035519         0.001697  1.000000 -0.766646 -0.011733
             lon  0.058941 -0.013200     -0.005032    -0.042667  0.004839        -0.026836 -0.766646  1.000000 -0.003541
           price  0.028516 -0.349885     -0.277235    -0.893328 -0.859373        -0.076274 -0.011733 -0.003541  1.000000
```

In [160]:
```
import seaborn as sns
sns.heatmap(cor,vmax=1,vmin=-1,annot=True,linewidths=.5,cmap='bwr') #create a heatmap to show the correlate
```

Out[160]:
```
<Axes: >
```



In [20]:

```

import pandas as pd
import numpy as np
```

In [21]:

```
data=pd.read_csv("/home/placement/Desktop/nio/fiat500.csv")#readin the file to analyze the data
```

In [22]:

```
data.head()
```

Out[22]:

```
   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 [23]:
```
cor_mat=data.corr()
cor_mat
```

```
/tmp/ipykernel_6617/2947089049.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 o
f numeric_only to silence this warning.
  cor_mat=data.corr()
```

Out[23]:

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price
ID	1.000000	-0.034059	-0.060753	-0.006537	0.007803	-0.058207	0.058941	0.028516
engine_power	-0.034059	1.000000	0.319190	0.285495	-0.005030	0.005721	-0.005032	-0.277235
age_in_days	-0.060753	0.319190	1.000000	0.833890	0.075775	0.062982	-0.042667	-0.893328
km	-0.006537	0.285495	0.833890	1.000000	0.097539	0.035519	0.004839	-0.859373
previous_owners	0.007803	-0.005030	0.075775	0.097539	1.000000	0.001697	-0.026836	-0.076274
lat	-0.058207	0.005721	0.062982	0.035519	0.001697	1.000000	-0.766646	-0.011733
lon	0.058941	-0.005032	-0.042667	0.004839	-0.026836	-0.766646	1.000000	-0.003541
price	0.028516	-0.277235	-0.893328	-0.859373	-0.076274	-0.011733	-0.003541	1.000000

In [24]:
```
data1=data.drop(["model"],axis=1)
```

```
In [26]: data1
```

Out[26]:

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price
0	1	51	882	25000	1	44.907242	8.611560	8900
1	2	51	1186	32500	1	45.666359	12.241890	8800
2	3	74	4658	142228	1	45.503300	11.417840	4200
3	4	51	2739	160000	1	40.633171	17.634609	6000
4	5	73	3074	106880	1	41.903221	12.495650	5700
...	...	...	...	...	...	...	...	...
1533	1534	51	3712	115280	1	45.069679	7.704920	5200
1534	1535	74	3835	112000	1	45.845692	8.666870	4600
1535	1536	51	2223	60457	1	45.481541	9.413480	7500
1536	1537	51	2557	80750	1	45.000702	7.682270	5990
1537	1538	51	1766	54276	1	40.323410	17.568270	7900

1538 rows × 8 columns

```
In [ ]:
```

In [25]:
```
cor_mat=data1.corr()
cor_mat
```

Out[25]:

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price
ID	1.000000	-0.034059	-0.060753	-0.006537	0.007803	-0.058207	0.058941	0.028516
engine_power	-0.034059	1.000000	0.319190	0.285495	-0.005030	0.005721	-0.005032	-0.277235
age_in_days	-0.060753	0.319190	1.000000	0.833890	0.075775	0.062982	-0.042667	-0.893328
km	-0.006537	0.285495	0.833890	1.000000	0.097539	0.035519	0.004839	-0.859373
previous_owners	0.007803	-0.005030	0.075775	0.097539	1.000000	0.001697	-0.026836	-0.076274
lat	-0.058207	0.005721	0.062982	0.035519	0.001697	1.000000	-0.766646	-0.011733
lon	0.058941	-0.005032	-0.042667	0.004839	-0.026836	-0.766646	1.000000	-0.003541
price	0.028516	-0.277235	-0.893328	-0.859373	-0.076274	-0.011733	-0.003541	1.000000

In [ ]: