

```
In [3]: import pandas as pd
import pickle
```

```
In [4]: data=pd.read_csv("/home/palacement/Downloads/fiat500.csv")
```

```
In [5]: data.describe()
```

```
Out[5]:
```

	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 [6]: data.head()
```

```
Out[6]:
```

	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 [7]: `data.tail()`

Out[7]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price
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 [8]: data.info

```
Out[8]: <bound method DataFrame.info of
0      1 lounge      51      882  25000      1
1      2  pop      51     1186  32500      1
2      3  sport     74     4658 142228      1
3      4 lounge     51     2739 160000      1
4      5  pop      73     3074 106880      1
...    ...    ...    ...    ...    ...
1533 1534  sport     51     3712 115280      1
1534 1535 lounge     74     3835 112000      1
1535 1536  pop      51     2223  60457      1
1536 1537 lounge     51     2557  80750      1
1537 1538  pop      51     1766  54276      1

      lat      lon  price
0  44.907242  8.611560  8900
1  45.666359 12.241890  8800
2  45.503300 11.417840  4200
3  40.633171 17.634609  6000
4  41.903221 12.495650  5700
...    ...    ...    ...
1533 45.069679  7.704920  5200
1534 45.845692  8.666870  4600
1535 45.481541  9.413480  7500
1536 45.000702  7.682270  5990
1537 40.323410 17.568270  7900

[1538 rows x 9 columns]>
```

In [9]: data1=data.drop(['model'],axis=1)

```
In [10]: cor=data1.corr()  
cor
```

```
Out[10]:
```

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

```
Out[11]: ['ID',  
          'model',  
          'engine_power',  
          'age_in_days',  
          'km',  
          'previous_owners',  
          'lat',  
          'lon',  
          'price']
```

```
In [12]: data.groupby(['model']).count()
```

```
Out[12]:
```

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price
model								
lounge	1094	1094	1094	1094	1094	1094	1094	1094
pop	358	358	358	358	358	358	358	358
sport	86	86	86	86	86	86	86	86

```
In [13]: data.groupby(['previous_owners']).count()
```

```
Out[13]:
```

	ID	model	engine_power	age_in_days	km	lat	lon	price
previous_owners								
1	1389	1389	1389	1389	1389	1389	1389	1389
2	117	117	117	117	117	117	117	117
3	23	23	23	23	23	23	23	23
4	9	9	9	9	9	9	9	9

```
In [14]: data.groupby(['model']).count()
```

```
Out[14]:
```

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price
model								
lounge	1094	1094	1094	1094	1094	1094	1094	1094
pop	358	358	358	358	358	358	358	358
sport	86	86	86	86	86	86	86	86

```
In [15]: data1=data.drop(['lat','ID'],axis=1)
```

```
In [16]: data1.head()
```

```
Out[16]:
```

	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

```
In [17]: data['price'].sum()
```

```
Out[17]: 13189894
```

```
In [18]: data2=data.loc[(data.model=='lounge')]  
data2
```

```
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
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 [19]: data3=data.loc[(data.km<=20000)]  
data3
```

```
Out[19]:
```

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price	
	6	7	lounge	51	731	11600	1	44.907242	8.61156	10750
	11	12	lounge	51	366	17500	1	45.069679	7.70492	10990
	12	13	lounge	51	456	18450	1	45.426571	11.78813	9700
	20	21	lounge	51	397	19037	1	45.707249	11.47760	10500
	22	23	lounge	51	1035	8000	1	44.506088	12.04417	10600

	1517	1518	pop	51	366	16100	1	44.692520	10.10396	10900
	1518	1519	lounge	51	397	16053	1	38.122070	13.36112	10500
	1520	1521	lounge	51	1035	15000	1	41.903221	12.49565	10990
	1522	1523	lounge	51	366	14618	1	45.707249	11.47760	10500
	1527	1528	pop	51	517	3000	1	40.748241	14.52835	9999

385 rows × 9 columns


```
In [20]: data3=data.loc[(data.model=='pop')]  
data3
```

```
Out[20]:
```

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price	
	1	2	pop	51	1186	32500	1	45.666359	12.241890	8800
	4	5	pop	73	3074	106880	1	41.903221	12.495650	5700
	5	6	pop	74	3623	70225	1	45.000702	7.682270	7900
	10	11	pop	51	790	43286	1	40.871429	14.438960	8950
	13	14	pop	51	3835	120000	1	40.531590	17.436159	4800

	1524	1525	pop	51	2192	53300	1	40.609531	14.980930	7900
	1527	1528	pop	51	517	3000	1	40.748241	14.528350	9999
	1532	1533	pop	51	1917	52008	1	45.548000	11.549470	9900
	1535	1536	pop	51	2223	60457	1	45.481541	9.413480	7500
	1537	1538	pop	51	1766	54276	1	40.323410	17.568270	7900

358 rows × 9 columns

```
In [21]: list(data)
```

```
Out[21]: ['ID',  
          'model',  
          'engine_power',  
          'age_in_days',  
          'km',  
          'previous_owners',  
          'lat',  
          'lon',  
          'price']
```

```
In [22]: data['model']=data['model'].map({'lounge':1,'pop':2,'sport':3})
data
```

```
Out[22]:
```

	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 [23]: data

Out[23]:

	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 [24]: cor=data.corr()  
cor
```

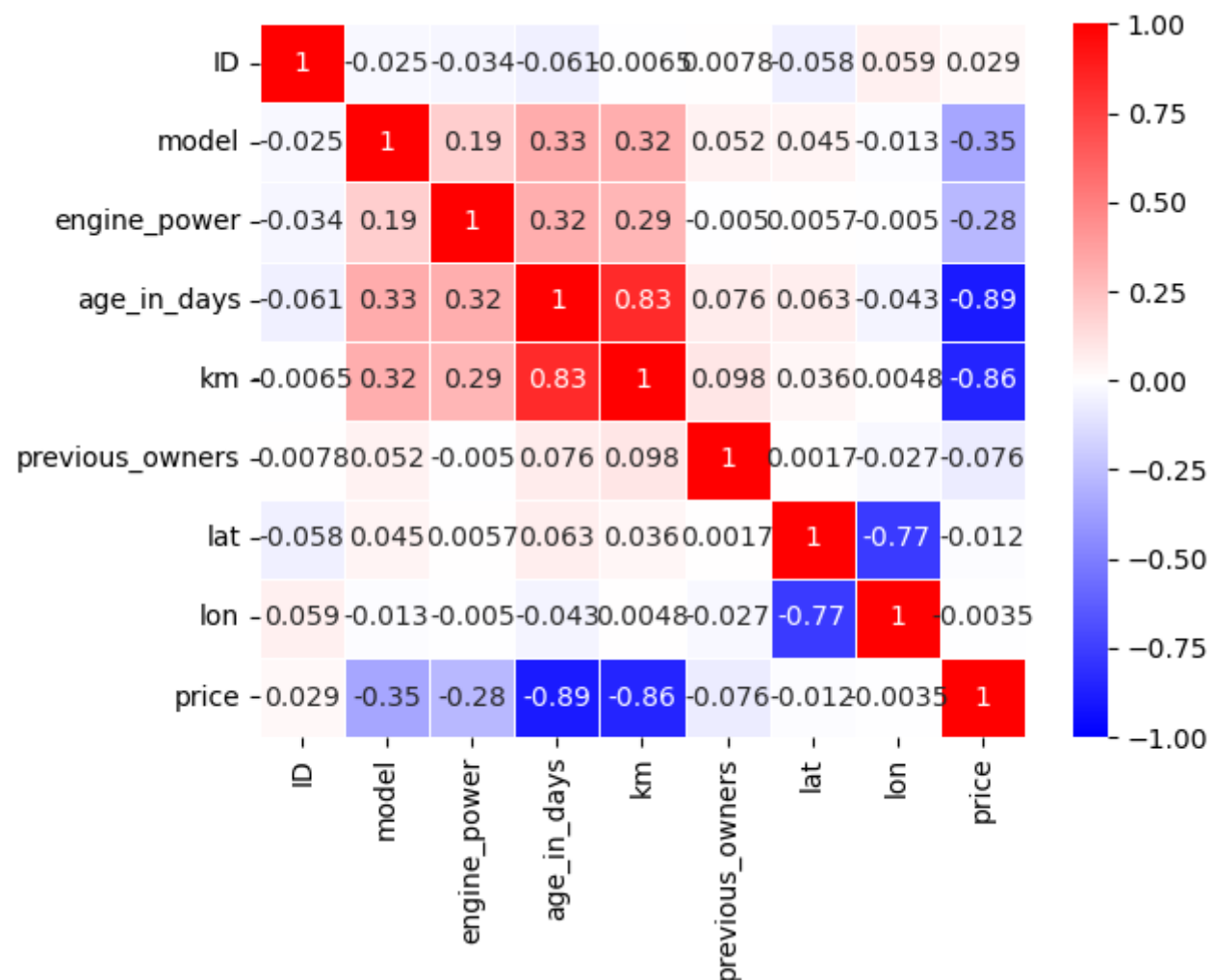
```
Out[24]:
```

	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 [25]: import seaborn as sns
```

```
In [26]: sns.heatmap(cor, vmax=1, vmin=-1, annot=True, linewidth=.5, cmap='bwr')
```

```
Out[26]: <Axes: >
```



In []: