

```
In [45]: import pandas as pd  
data=pd.read_csv("/home/placement/Desktop/EEE(222)/fiat500.csv")
```

```
In [46]: data.describe()
```

Out[46]:

	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 [47]: data.loc[(data.km<=50000)]
```

```
Out[47]:
```

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price
0	1	lounge	51	882	25000	1	44.907242	8.61156	8900
1	2	pop	51	1186	32500	1	45.666359	12.24189	8800
6	7	lounge	51	731	11600	1	44.907242	8.61156	10750
7	8	lounge	51	1521	49076	1	41.903221	12.49565	9190
10	11	pop	51	790	43286	1	40.871429	14.43896	8950
...	...	...	...	...	...	...	...	...	...
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
1529	1530	lounge	51	731	22551	1	38.122070	13.36112	9900
1530	1531	lounge	51	670	29000	1	45.764648	8.99450	10800

907 rows × 9 columns

In [48]: data

Out[48]:

	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
...	...	...	...	...	...	...	...	...	...
1533	1534	sport	51	3712	115280	1	45.069679	7.704920	5200
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

1538 rows × 9 columns

In [49]: data.groupby(['model']).count()

Out[49]:

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price
<b>model</b>								
<b>lounge</b>	1094	1094	1094	1094	1094	1094	1094	1094
<b>pop</b>	358	358	358	358	358	358	358	358
<b>sport</b>	86	86	86	86	86	86	86	86

In [50]: data=data.rename(columns={'model':'model\_name'})

```
In [51]: list(data)
```

```
Out[51]: ['ID',  
          'model_name',  
          'engine_power',  
          'age_in_days',  
          'km',  
          'previous_owners',  
          'lat',  
          'lon',  
          'price']
```

```
In [52]: data1=data.drop('model_name',axis=1)
```

```
In [53]: data1
```

```
Out[53]:
```

	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 [54]: data1.corr()

Out[54]:

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

In [55]: cor=data1.corr()

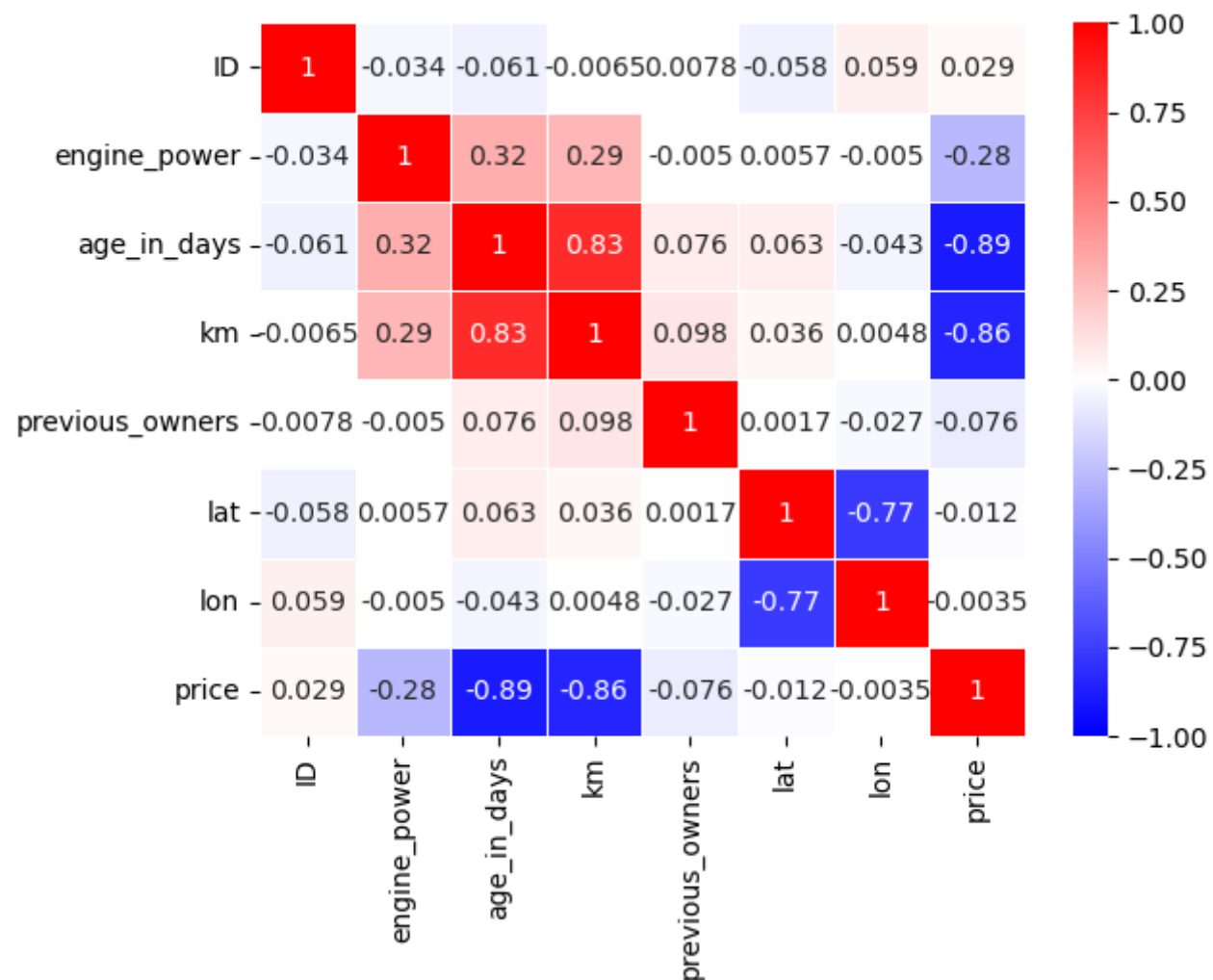
In [56]: cor

Out[56]:

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

```
In [57]: import seaborn as sns
sns.heatmap(cor, vmax=1, vmin=-1, annot=True, linewidths=.5, cmap='bwr')
```

Out[57]: <Axes: >



```
In [58]: list(data1)
```

```
Out[58]: ['ID',  
          'engine_power',  
          'age_in_days',  
          'km',  
          'previous_owners',  
          'lat',  
          'lon',  
          'price']
```

```
In [59]: y=data1['price']  
x=data1.drop('price',axis=1)
```

```
In [60]: y
```

```
Out[60]: 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 [61]: data.head()

Out[61]:

	ID	model_name	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 [62]: data.tail()

Out[62]:

	ID	model_name	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 [63]: data.head(100)
```

```
Out[63]:
```

	ID	model_name	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
...	...	...	...	...	...	...	...	...	...
95	96	sport	51	4292	165600	1	44.715408	11.308300	5950
96	97	pop	51	1066	28000	1	41.769051	12.662810	8500
97	98	sport	51	2009	86000	2	40.633171	17.634609	7800
98	99	lounge	51	456	18592	2	45.393600	10.482240	10900
99	100	pop	51	731	41558	2	45.571220	9.159140	8790

100 rows × 9 columns

```
In [64]: data.tail(1000)
```

```
Out[64]:
```

	ID	model_name	engine_power	age_in_days	km	previous_owners	lat	lon	price
<b>538</b>	539	lounge	51	4535	124000	1	45.069679	7.70492	4450
<b>539</b>	540	lounge	51	1096	63500	1	46.792019	11.98032	9900
<b>540</b>	541	lounge	51	2131	44659	1	38.122070	13.36112	8900
<b>541</b>	542	pop	51	4353	104000	1	45.633308	9.58734	4300
<b>542</b>	543	lounge	51	1431	68000	1	45.420921	8.94065	8800
...	...	...	...	...	...	...	...	...	...
<b>1533</b>	1534	sport	51	3712	115280	1	45.069679	7.70492	5200
<b>1534</b>	1535	lounge	74	3835	112000	1	45.845692	8.66687	4600
<b>1535</b>	1536	pop	51	2223	60457	1	45.481541	9.41348	7500
<b>1536</b>	1537	lounge	51	2557	80750	1	45.000702	7.68227	5990
<b>1537</b>	1538	pop	51	1766	54276	1	40.323410	17.56827	7900

1000 rows × 9 columns

```
In [65]: cor=data.corr()  
cor
```

/tmp/ipykernel\_4290/4173678507.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.

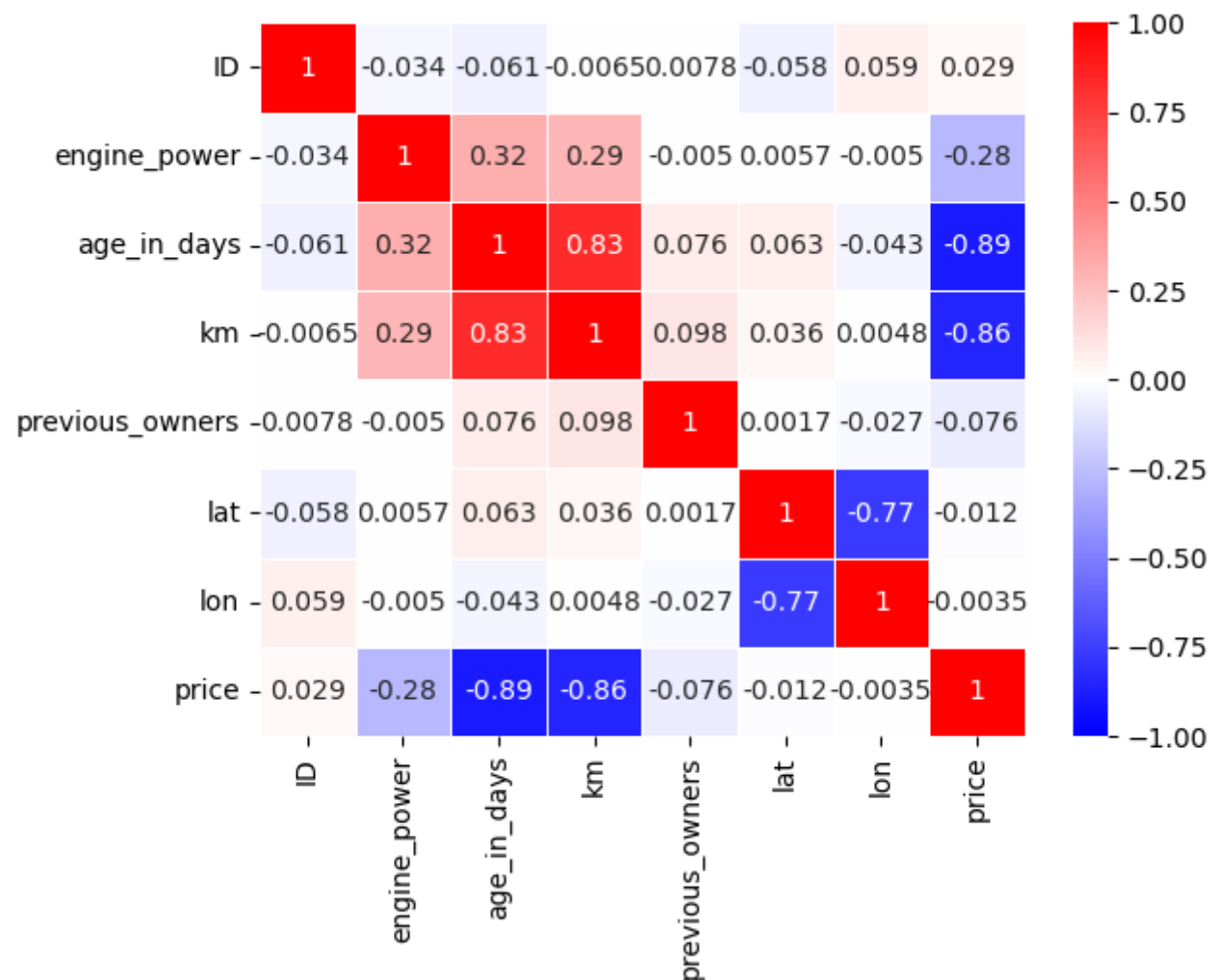
```
cor=data.corr()
```

Out[65]:

	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 [66]: import seaborn as sns
sns.heatmap(cor, vmax=1, vmin=-1, annot=True, linewidths=.5, cmap='bwr')
```

Out[66]: <Axes: >



In [ ]: