

importing pandas&Numpy

```
In [2]: import pandas as pd
import numpy as np
```

Reading fiat500.csv

```
In [3]: data=pd.read_csv("/home/placement/Downloads/fiat500.csv")
data.describe()
```

Out[3]:

	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

List columns

```
In [4]: list(data)
```

```
Out[4]: ['ID',  
         'model',  
         'engine_power',  
         'age_in_days',  
         'km',  
         'previous_owners',  
         'lat',  
         'lon',  
         'price']
```

removing model column from table

```
In [5]: data=data.drop('model',axis=1)
data
```

Out[5]:

	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

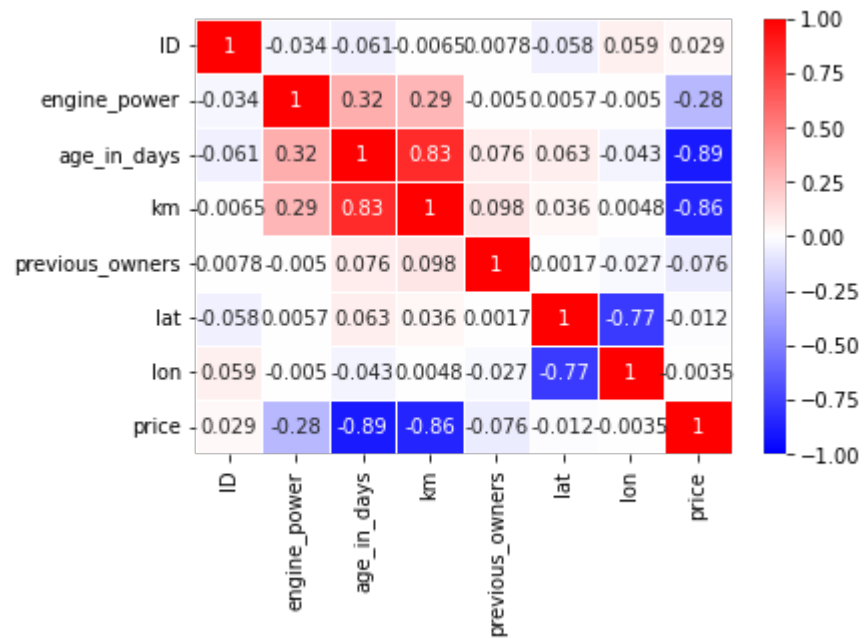
```
In [6]: cor=data.corr()  
cor
```

Out[6]:

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

Out[7]: <Axes: >



```
In [8]: list(data)
```

```
Out[8]: ['ID',
         'engine_power',
         'age_in_days',
         'km',
         'previous_owners',
         'lat',
         'lon',
         'price']
```

```
In [9]: data.tail(10)
```

```
Out[9]:
```

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price
1528	1529	51	2861	126000	1	43.841980	10.51531	5500
1529	1530	51	731	22551	1	38.122070	13.36112	9900
1530	1531	51	670	29000	1	45.764648	8.99450	10800
1531	1532	73	4505	127000	1	45.528511	9.59323	4750
1532	1533	51	1917	52008	1	45.548000	11.54947	9900
1533	1534	51	3712	115280	1	45.069679	7.70492	5200
1534	1535	74	3835	112000	1	45.845692	8.66687	4600
1535	1536	51	2223	60457	1	45.481541	9.41348	7500
1536	1537	51	2557	80750	1	45.000702	7.68227	5990
1537	1538	51	1766	54276	1	40.323410	17.56827	7900

```
In [10]: data.head(10)
```

```
Out[10]:
```

	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
5	6	74	3623	70225	1	45.000702	7.682270	7900
6	7	51	731	11600	1	44.907242	8.611560	10750
7	8	51	1521	49076	1	41.903221	12.495650	9190
8	9	73	4049	76000	1	45.548000	11.549470	5600
9	10	51	3653	89000	1	45.438301	10.991700	6000

checking the null values

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

```
Out[11]:
```

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...
1533	False	False	False	False	False	False	False	False
1534	False	False	False	False	False	False	False	False
1535	False	False	False	False	False	False	False	False
1536	False	False	False	False	False	False	False	False
1537	False	False	False	False	False	False	False	False

1538 rows × 8 columns


```
In [12]: data.notnull()
```

```
Out[12]:
```

	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price
0	True	True	True	True	True	True	True	True
1	True	True	True	True	True	True	True	True
2	True	True	True	True	True	True	True	True
3	True	True	True	True	True	True	True	True
4	True	True	True	True	True	True	True	True
...
1533	True	True	True	True	True	True	True	True
1534	True	True	True	True	True	True	True	True
1535	True	True	True	True	True	True	True	True
1536	True	True	True	True	True	True	True	True
1537	True	True	True	True	True	True	True	True

1538 rows × 8 columns

count

```
In [13]: data.count()
```

```
Out[13]: ID                1538
engine_power              1538
age_in_days               1538
km                        1538
previous_owners           1538
lat                       1538
lon                       1538
price                     1538
dtype: int64
```

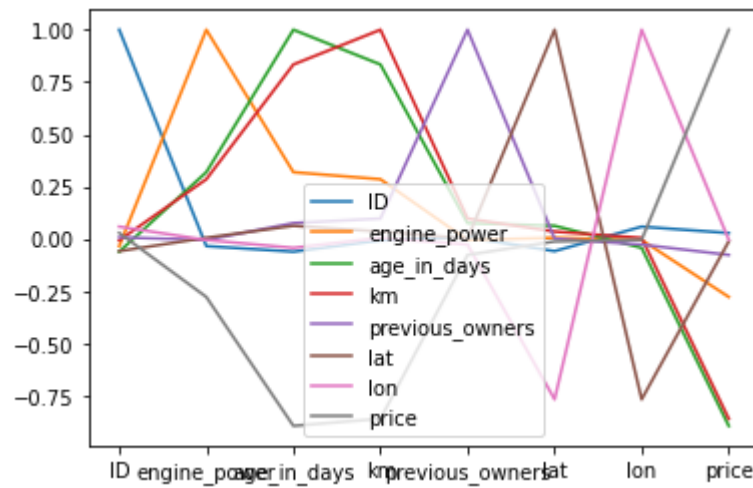
Correlation

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

plotting cor

```
In [14]: cor.plot()
```

```
Out[14]: <Axes: >
```



```
In [15]: data.groupby(["previous_owners"]).count()
```

Out[15]:

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

```
In [16]: data['price'].sum()
```

Out[16]: 13189894

```
In [17]: det=data.drop(["ID","engine_power","lat","lon","previous_owners","price"],axis=1)
det
```

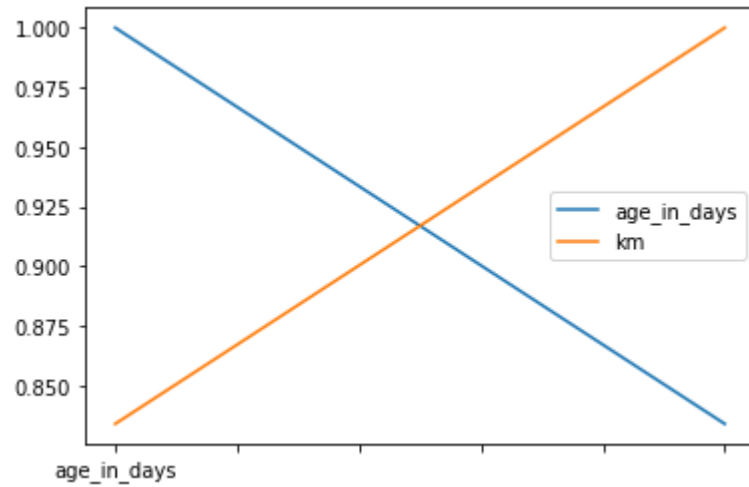
Out[17]:

	age_in_days	km
0	882	25000
1	1186	32500
2	4658	142228
3	2739	160000
4	3074	106880
...
1533	3712	115280
1534	3835	112000
1535	2223	60457
1536	2557	80750
1537	1766	54276

1538 rows × 2 columns

```
In [18]: cor1=det.corr()  
cor1.plot()
```

Out[18]: <Axes: >



```
In [19]: data.loc[(data.price<5000)].count()
```

```
Out[19]: ID          99  
engine_power      99  
age_in_days       99  
km                99  
previous_owners   99  
lat               99  
lon               99  
price             99  
dtype: int64
```

In []:

In []:

In []:

