In [4]:
```
import pandas as pd
```

In [5]:
```
data=pd.read_csv("/home/placement/Downloads/fiat500.csv")
```

In [6]:
```
data1=data.loc[(data.km<=50000)]
```

In [7]:
```
data1
```

Out[7]:

	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 [8]:
```
data2=data.groupby(['model']).count()
```

In [9]:
```
data2
```

Out[9]:

model	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price
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 [10]:
```
data1=data1.rename(columns={'model_name':'model'})
list(data1)
```

Out[10]:
```
['ID',
 'model',
 'engine_power',
 'age_in_days',
 'km',
 'previous_owners',
 'lat',
 'lon',
 'price']
```

In [11]:
```
data1=data1.rename(columns={'engine_power':'engine'})
list(data1)
```

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

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

Out[12]:

model	ID	engine_power	age_in_days	km	previous_owners	lat	lon	price
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=data.drop(['model'],axis=1)
```

In [14]:
```
cor=data.corr()
```

In [15]:
```
cor
```

Out[15]:

	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 [16]:
```
data1['model']=data1['model'].map({'lounge':1,'pop':2,'sport':3})
```

In [17]:
```
data1
```

Out[17]:

	ID	model	engine	age_in_days	km	previous_owners	lat	lon	price
0	1	1	51	882	25000	1	44.907242	8.61156	8900
1	2	2	51	1186	32500	1	45.666359	12.24189	8800
6	7	1	51	731	11600	1	44.907242	8.61156	10750
7	8	1	51	1521	49076	1	41.903221	12.49565	9190
10	11	2	51	790	43286	1	40.871429	14.43896	8950
...	...	...	...	...	...	...	...	...	...
1525	1526	1	51	790	41870	1	45.707249	11.47760	9500
1526	1527	1	51	1705	23600	1	38.122070	13.36112	9300
1527	1528	2	51	517	3000	1	40.748241	14.52835	9999
1529	1530	1	51	731	22551	1	38.122070	13.36112	9900
1530	1531	1	51	670	29000	1	45.764648	8.99450	10800

907 rows × 9 columns

In [18]:
```
import seaborn as sns
sns.heatmap(cor,vmax=1,vmin=-1,annot=True,linewidth=.5,cmap='bwr')
```

Out[18]: <Axes: >