In [1]:

```
#import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [18]:

```
data = pd.read_csv(r"C:\Users\user\Downloads\1_fiat500_VehicleSelection_Dataset
data
```

Out[18]:

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

1538 rows × 9 columns

In [21]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1538 entries, 0 to 1537
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   ID               1538 non-null   int64  
 1   model            1538 non-null   object 
 2   engine_power     1538 non-null   int64  
 3   age_in_days      1538 non-null   int64  
 4   km               1538 non-null   int64  
 5   previous_owners  1538 non-null   int64  
 6   lat              1538 non-null   float64
 7   lon              1538 non-null   float64
 8   price            1538 non-null   int64  
dtypes: float64(2), int64(6), object(1)
memory usage: 108.3+ KB
```

In [22]:
```
data.isna().sum()
```

Out[22]:
```
ID                 0
model              0
engine_power       0
age_in_days        0
km                 0
previous_owners    0
lat                0
lon                0
price              0
dtype: int64
```

In [24]:
```
data.describe()
```

Out[24]:

	ID	engine_power	age_in_days	km	previous_owners	lat	
count	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000	1
mean	769.500000	51.904421	1650.980494	53396.011704	1.123537	43.541361	
std	444.126671	3.988023	1289.522278	40046.830723	0.416423	2.133518	
min	1.000000	51.000000	366.000000	1232.000000	1.000000	36.855839	
25%	385.250000	51.000000	670.000000	20006.250000	1.000000	41.802990	
50%	769.500000	51.000000	1035.000000	39031.000000	1.000000	44.394096	
75%	1153.750000	51.000000	2616.000000	79667.750000	1.000000	45.467960	
max	1538.000000	77.000000	4658.000000	235000.000000	4.000000	46.795612	

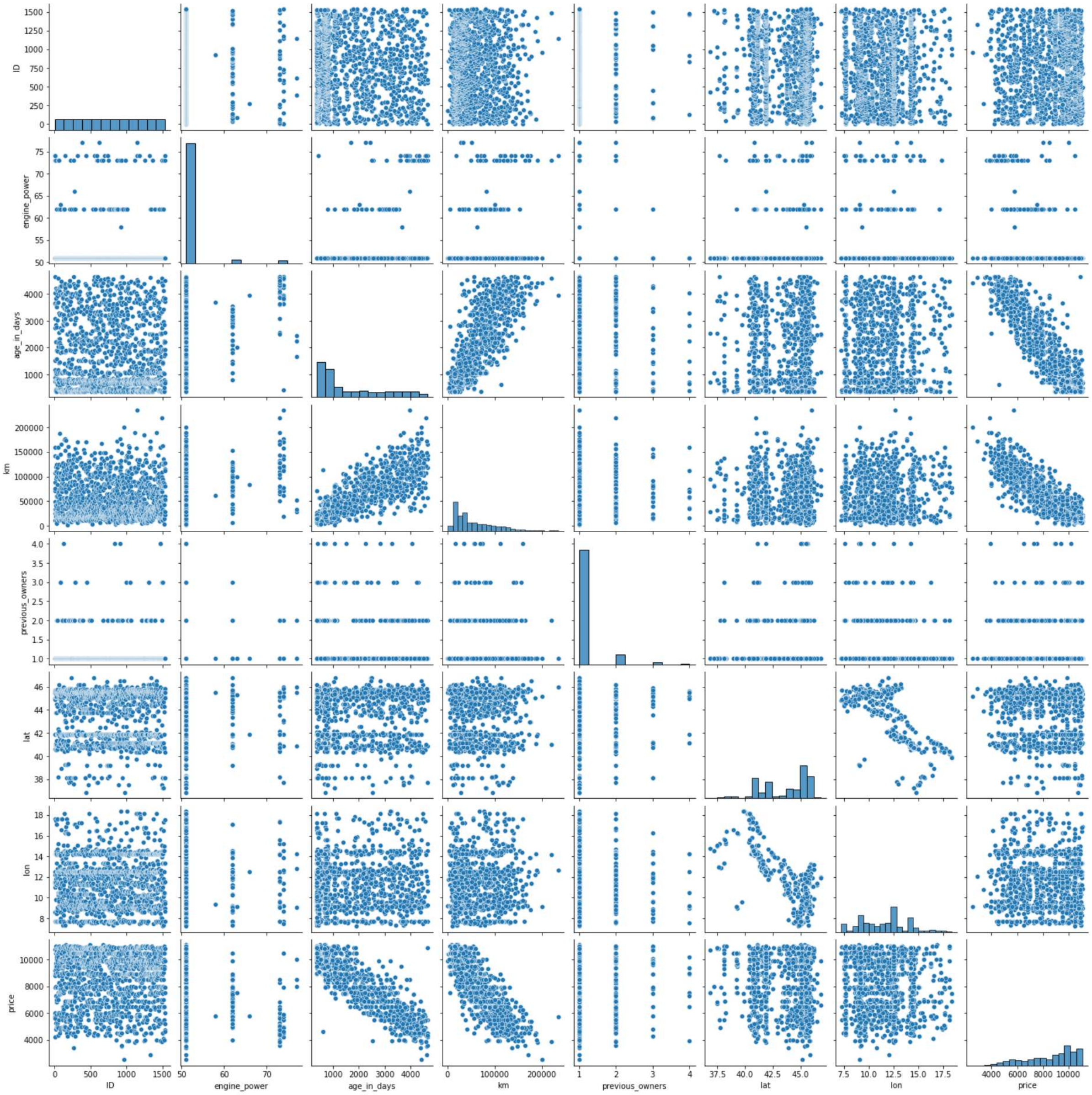
In [25]:
```
data.columns
```

Out[25]:
```
Index(['ID', 'model', 'engine_power', 'age_in_days', 'km', 'previous_owners',
       'lat', 'lon', 'price'],
      dtype='object')
```

EDA and visualization

In [26]:
```
sns.pairplot(data)
```

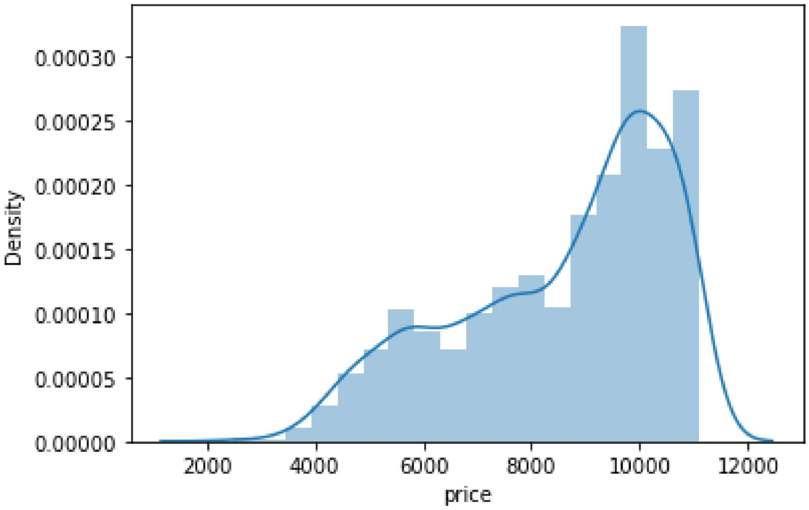
Out[26]:
```
<seaborn.axisgrid.PairGrid at 0x1c5832ced60>
```



```
In [27]: sns.distplot(data['price'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: Fut
ureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function for hi
stograms).
  warnings.warn(msg, FutureWarning)
```

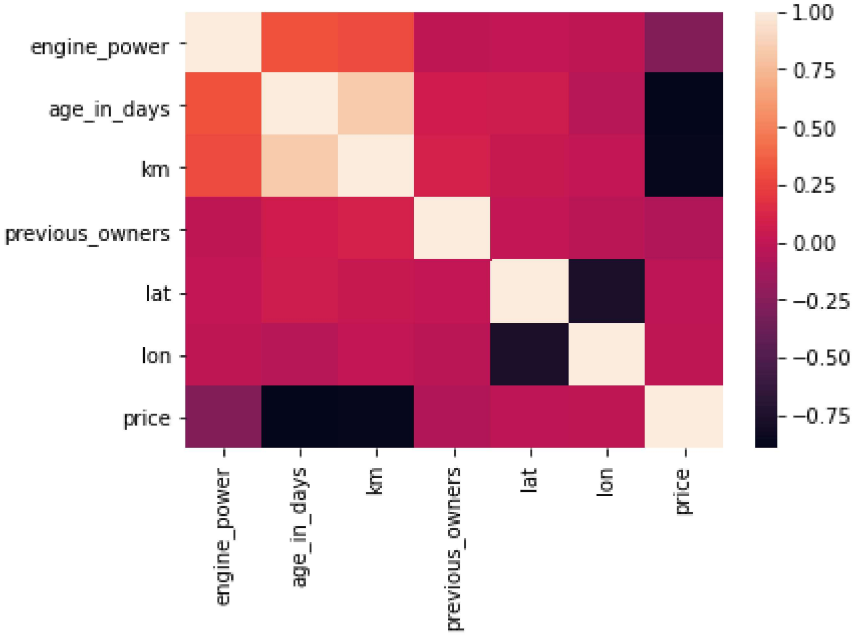
```
Out[27]: <AxesSubplot:xlabel='price', ylabel='Density'>
```



```
In [28]: data1 = data[['engine_power', 'age_in_days', 'km', 'previous_owners',
                'lat', 'lon', 'price']]
```

```
In [31]: sns.heatmap(data1.corr())
```

```
Out[31]: <AxesSubplot:>
```



model building

```
In [33]: x = data1[['engine_power', 'age_in_days', 'km', 'previous_owners',
                'lat', 'lon']]
         y = data1['price']
```

```
In [34]: from sklearn.model_selection import train_test_split

         x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.4)
```

```
In [36]: from sklearn.linear_model import LinearRegression

         lr = LinearRegression()
         lr.fit(x_train,y_train)

Out[36]: LinearRegression()
```

```
In [37]: print(lr.intercept_)

         8465.045963572862
```

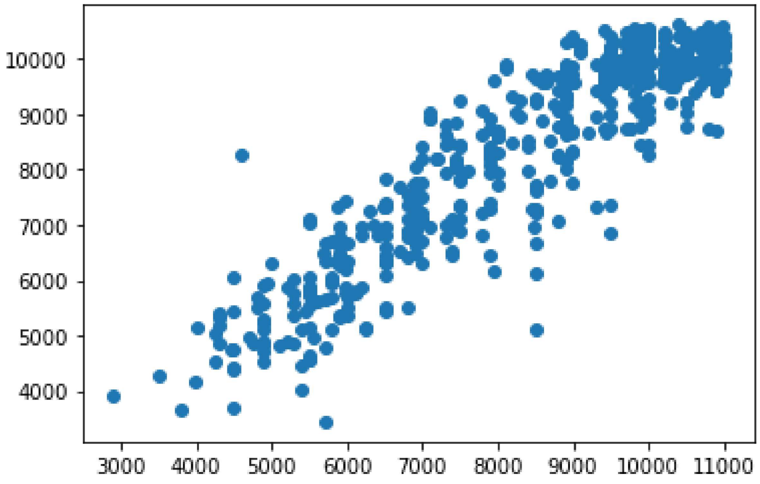
```
In [38]: coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
         coeff
```

Out[38]:

	Co-efficient
engine_power	4.392938
age_in_days	-0.881429
km	-0.018256
previous_owners	12.283227
lat	49.516277
lon	11.348332

```
In [39]: prediction = lr.predict(x_test)
         plt.scatter(y_test,prediction)

Out[39]: <matplotlib.collections.PathCollection at 0x1c5879e7fd0>
```



```
In [40]: print(lr.score(x_test,y_test))

         0.8466888818934549

In [ ]:
```