In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as py
import seaborn as sns
```

In [2]:
```
d=pd.read_csv(r"C:\Users\user\Downloads\fiat500_VehicleSelection_Dataset (2).csv")
d
```

Out[2]:

	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 [3]:
```
d.head()
```

Out[3]:

	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 [4]:
```
d.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 [5]:
```
d.describe()
```

Out[5]:

	ID	engine_power	age_in_days	km	previous_owners	lat	lo
count	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000	1538.0000
mean	769.500000	51.904421	1650.980494	53396.011704	1.123537	43.541361	11.5634
std	444.126671	3.988023	1289.522278	40046.830723	0.416423	2.133518	2.3281
min	1.000000	51.000000	366.000000	1232.000000	1.000000	36.855839	7.2454
25%	385.250000	51.000000	670.000000	20006.250000	1.000000	41.802990	9.5050
50%	769.500000	51.000000	1035.000000	39031.000000	1.000000	44.394096	11.8692
75%	1153.750000	51.000000	2616.000000	79667.750000	1.000000	45.467960	12.7690
max	1538.000000	77.000000	4658.000000	235000.000000	4.000000	46.795612	18.3655

In [6]:
```
d.columns
```

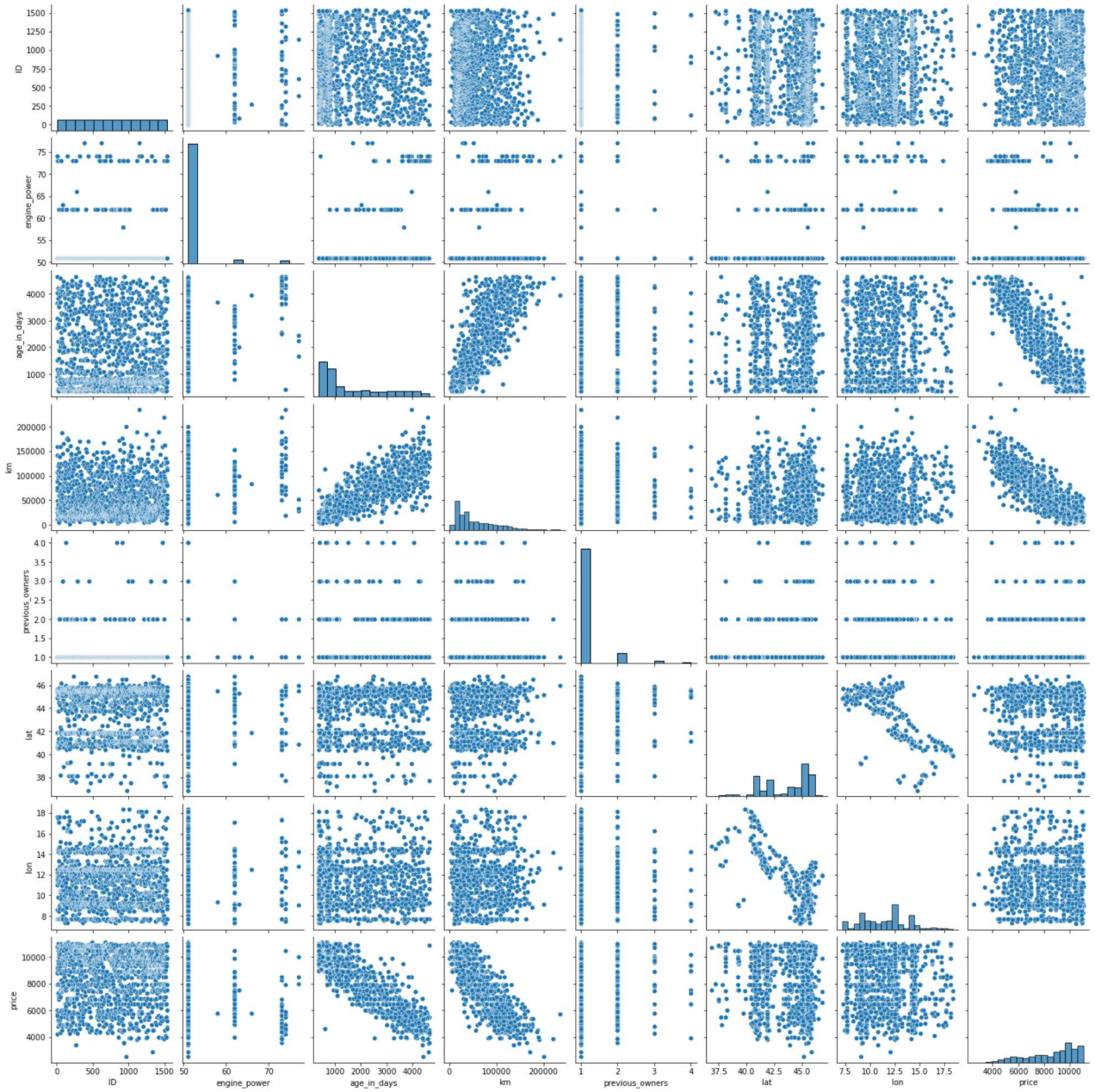
Out[6]:
```
Index(['ID', 'model', 'engine_power', 'age_in_days', 'km', 'previous_owners',
       'lat', 'lon', 'price'],
      dtype='object')
```

In [7]:
```
d.index
```

Out[7]:
```
RangeIndex(start=0, stop=1538, step=1)
```

In [8]:
```
sns.pairplot(d)
```

Out[8]:
```
<seaborn.axisgrid.PairGrid at 0x1f3cbdaf4c0>
```

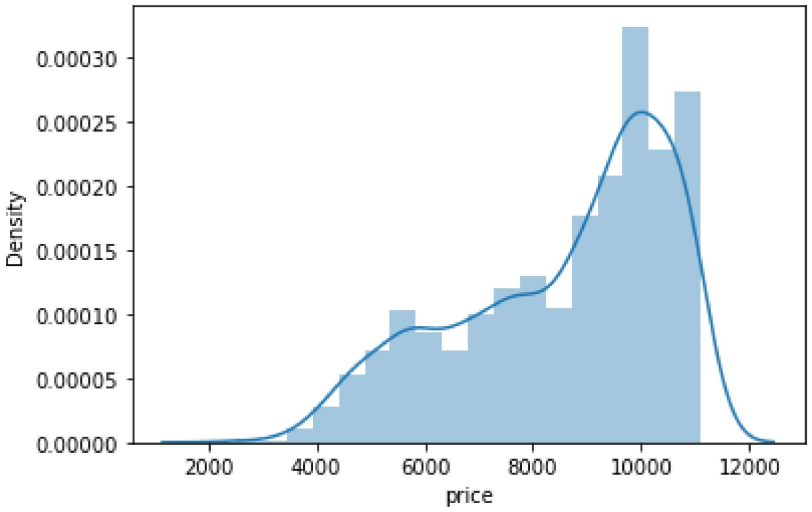


In [9]:

```
sns.distplot(d['price'])
```

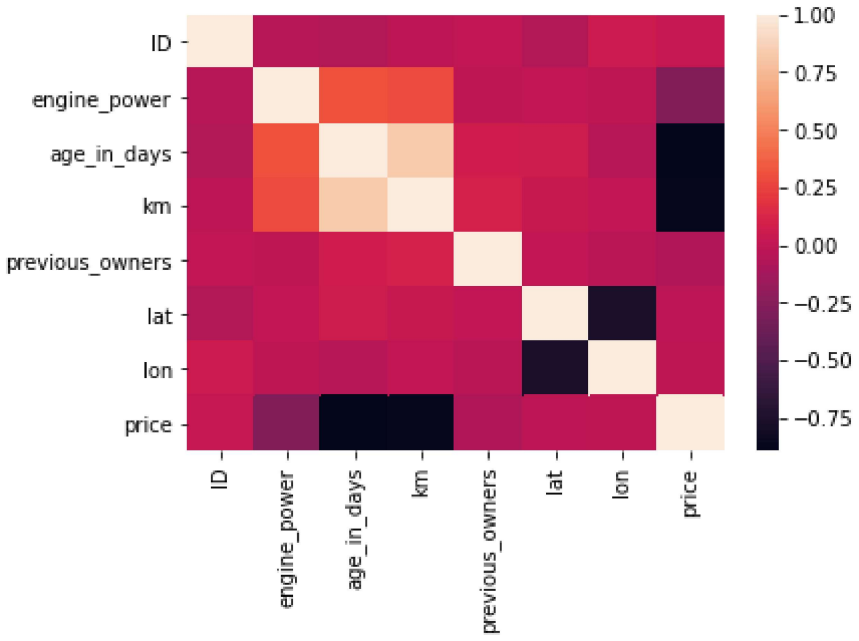
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[9]: <AxesSubplot:xlabel='price', ylabel='Density'>



In [10]:
```
d1=d[['ID','engine_power', 'age_in_days', 'km', 'previous_owners',
       'lat', 'lon', 'price']]
sns.heatmap(d1.corr())
```

Out[10]: <AxesSubplot:>



In [11]:
```
x=d1[['ID','engine_power', 'age_in_days', 'km', 'previous_owners','lat', 'lon']]
y =d1['price']
```

In [12]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
```

In [13]:
```
from sklearn.linear_model import LinearRegression
```

In [14]:
```
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[14]:
```
LinearRegression()
```

In [15]:
```
print(lr.intercept_)
```

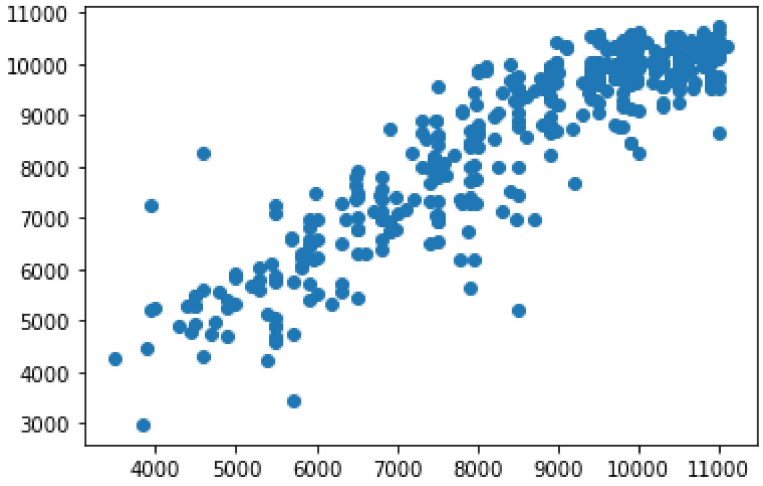
```
8817.24210723314
```

In [ ]:
```

```

In [16]:
```
prediction =lr.predict(x_test)
py.scatter(y_test,prediction)
```

Out[16]:
```
<matplotlib.collections.PathCollection at 0x1f3d0ec6250>
```



In [17]:
```
print(lr.score(x_test,y_test))
```

```
0.8204307134941244
```

In [18]:
```
print(lr.score(x_train,y_train))
```

```
0.8510592128642411
```

In [19]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [21]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[21]:
```
Ridge(alpha=10)
```

In [22]:
```
rr.score(x_test,y_test)
```

Out[22]:
```
0.8204305148861909
```

In [23]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[23]:
```
Lasso(alpha=10)
```

In [24]:
```
la.score(x_test,y_test)
```

Out[24]:
```
0.8203904923559537
```

In [ ]: