In [26]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [27]:
```
df=pd.read_csv("/Users/bob/Downloads/1_fiat500_VehicleSelection_Dat
df.fillna(0,inplace=True)
df
```

Out[27]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	
0	1.0	lounge	51.0	882.0	25000.0	1.0	44.907242	8.611
1	2.0	pop	51.0	1186.0	32500.0	1.0	45.666359	12.24
2	3.0	sport	74.0	4658.0	142228.0	1.0	45.503300	1
3	4.0	lounge	51.0	2739.0	160000.0	1.0	40.633171	17.63
4	5.0	pop	73.0	3074.0	106880.0	1.0	41.903221	12.49
...	...	...	...	...	...	...	...	
1544	0.0	0	0.0	0.0	0.0	0.0	0.000000	
1545	0.0	0	0.0	0.0	0.0	0.0	0.000000	
1546	0.0	0	0.0	0.0	0.0	0.0	0.000000	Nul
1547	0.0	0	0.0	0.0	0.0	0.0	0.000000	
1548	0.0	0	0.0	0.0	0.0	0.0	0.000000	

1549 rows × 11 columns

In [28]:
```
df.head()
```

Out[28]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	
0	1.0	lounge	51.0	882.0	25000.0	1.0	44.907242	8.611559
1	2.0	pop	51.0	1186.0	32500.0	1.0	45.666359	12.24188
2	3.0	sport	74.0	4658.0	142228.0	1.0	45.503300	11.41
3	4.0	lounge	51.0	2739.0	160000.0	1.0	40.633171	17.63460
4	5.0	pop	73.0	3074.0	106880.0	1.0	41.903221	12.49565

In [29]:
```
df.info()
```

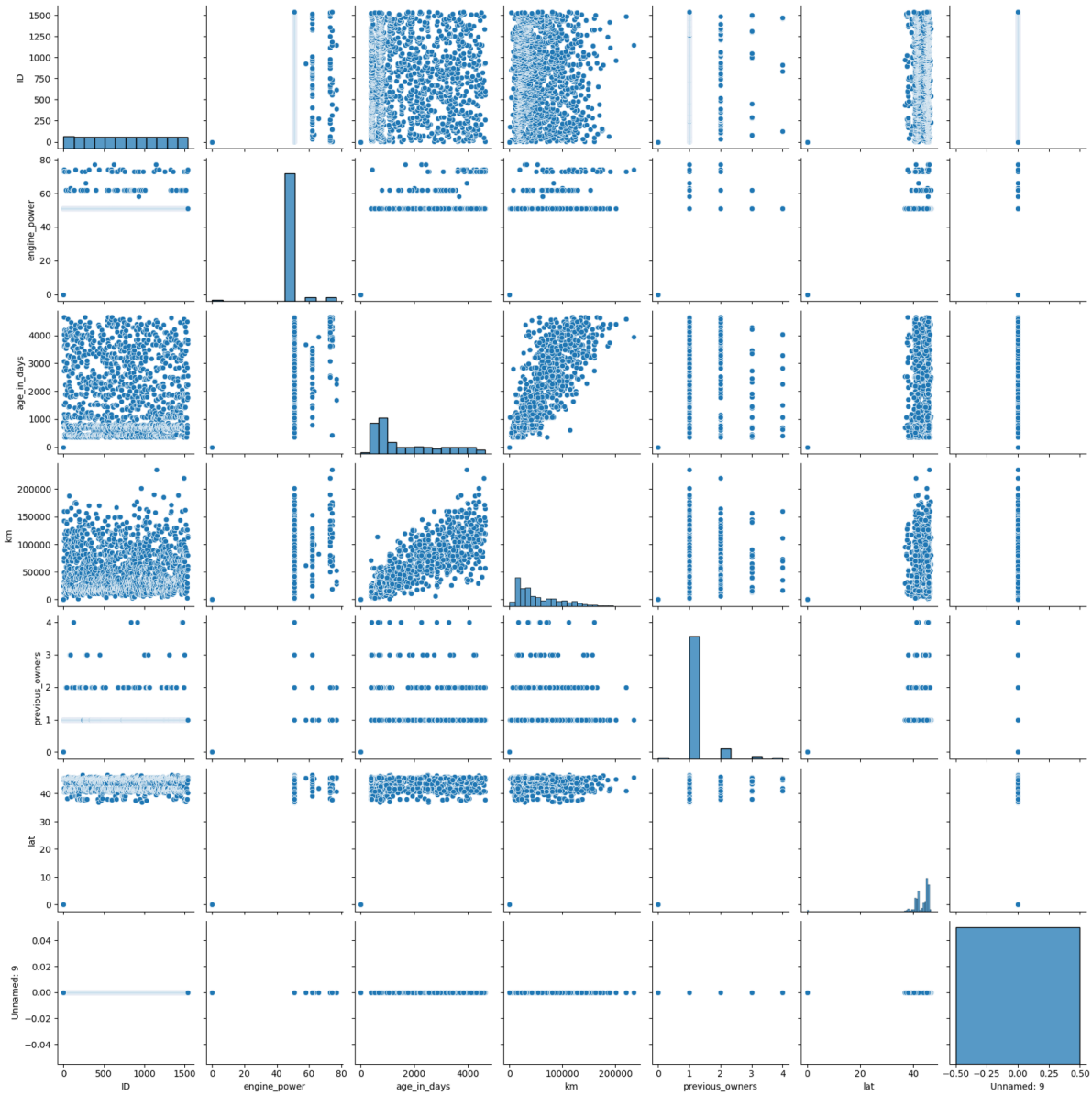
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1549 entries, 0 to 1548
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   ID               1549 non-null   float64
 1   model            1549 non-null   object 
 2   engine_power     1549 non-null   float64
 3   age_in_days      1549 non-null   float64
 4   km               1549 non-null   float64
 5   previous_owners  1549 non-null   float64
 6   lat              1549 non-null   float64
 7   lon              1549 non-null   object 
 8   price            1549 non-null   object 
 9   Unnamed: 9       1549 non-null   float64
 10  Unnamed: 10      1549 non-null   object 
dtypes: float64(7), object(4)
memory usage: 133.2+ KB
```

In [30]:
```
df.columns
```

Out[30]:
```
Index(['ID', 'model', 'engine_power', 'age_in_days', 'km', 'previo
us_owners',
       'lat', 'lon', 'price', 'Unnamed: 9', 'Unnamed: 10'],
      dtype='object')
```

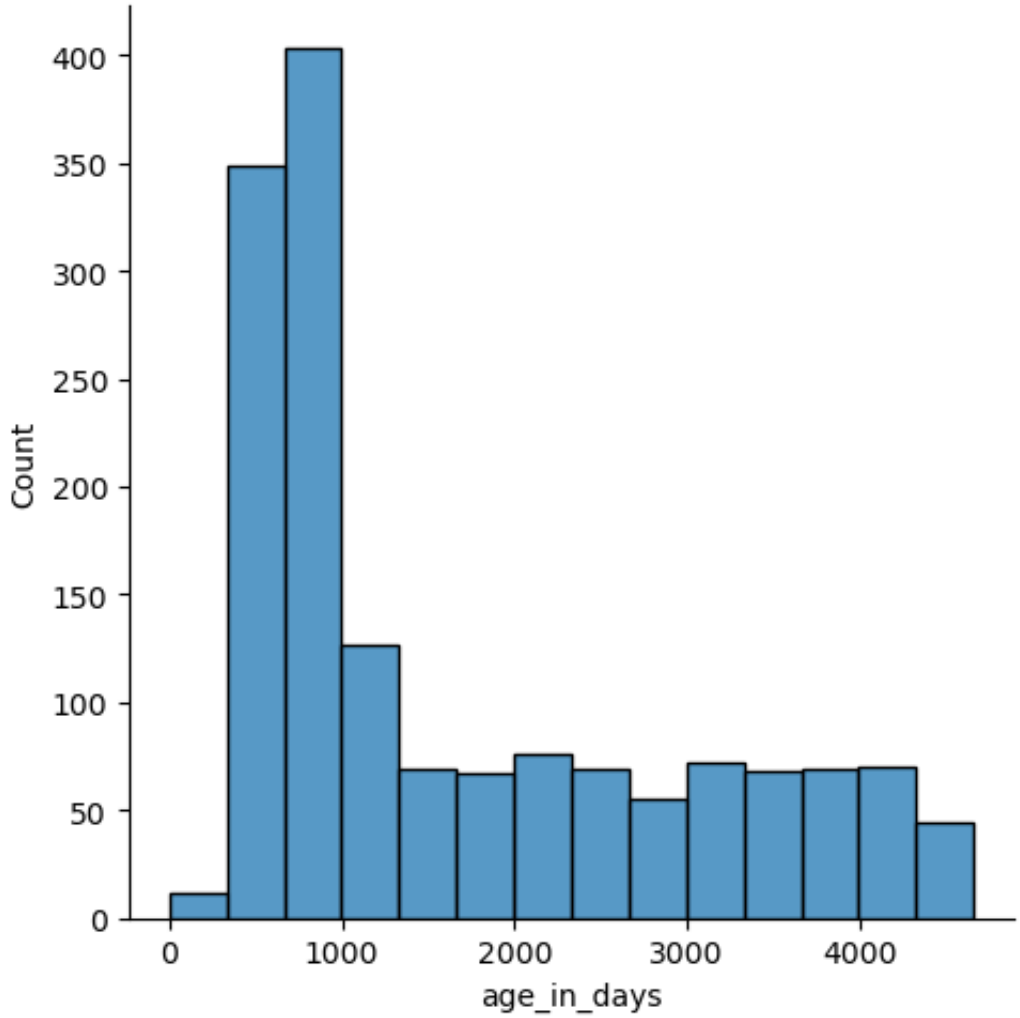
In [31]:
```
sns.pairplot(df)
```

Out[31]: <seaborn.axisgrid.PairGrid at 0x7fda987f92d0>



In [33]:
```
sns.displot(df['age_in_days'])
```

Out[33]:
```
<seaborn.axisgrid.FacetGrid at 0x7fdac53f3880>
```



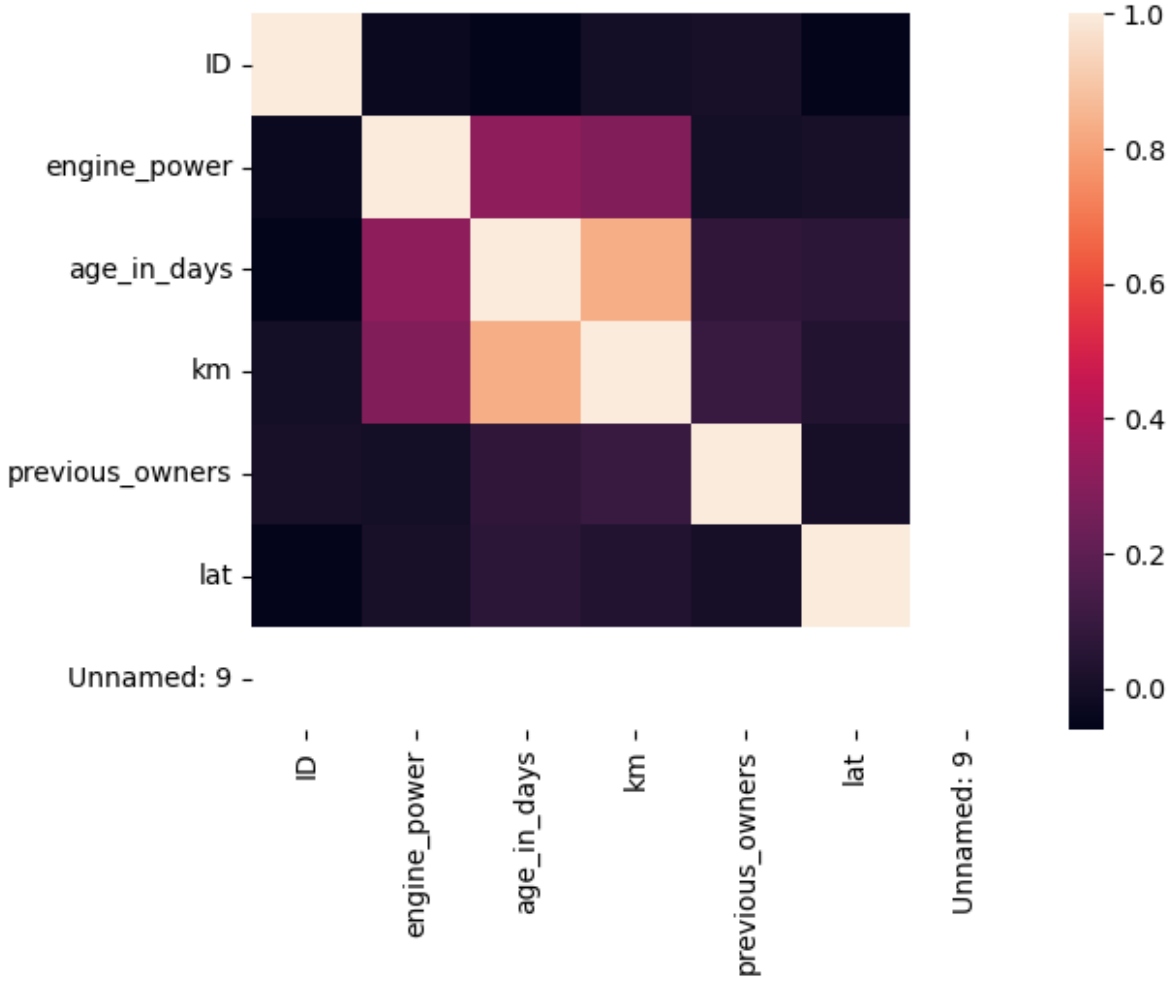
In [34]:
```
df1=df.drop(['Unnamed: 10'],axis=1)
df1
df1=df1.drop(df1.index[1537:])
df1.isna().sum()
```

Out[34]:
```
ID                0
model             0
engine_power      0
age_in_days       0
km                0
previous_owners   0
lat               0
lon               0
price             0
Unnamed: 9        0
dtype: int64
```

In [35]:
```
sns.heatmap(df1.corr())
```

```
/var/folders/2n/rrl24lws3pb1nz8_t911srvm0000gn/T/ipykernel_12149/781785195.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.
  sns.heatmap(df1.corr())
```

Out[35]: <Axes: >



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

In [37]:
```
df1.isna().sum()
```

Out[37]:
```
ID                0
model             0
engine_power      0
age_in_days       0
km                0
previous_owners   0
lat               0
lon               0
price             0
Unnamed: 9        0
dtype: int64
```

In [38]:
```
y=df1['price']
x=df1.drop(['price','Unnamed: 9','model'],axis=1)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
print(x_train)
```

```
          ID  engine_power  age_in_days        km  previous_owners
lat  \
1317  1318.0          51.0        701.0   25692.0              1.0
45.438789
940    941.0          51.0       2162.0   71000.0              1.0
41.903221
120    121.0          51.0       1096.0   50000.0              1.0
40.921669
805    806.0          51.0        456.0   13238.0              1.0
41.903221
832    833.0          51.0       2496.0   89000.0              1.0
40.667141
...      ...           ...          ...       ...              ...
...
1429  1430.0          51.0       2192.0  100000.0              1.0
43.817020
756    757.0          51.0        425.0   17770.0              1.0
41.408138
340    341.0          51.0       1247.0   22686.0              1.0
41.903221
1466  1467.0          51.0       1492.0   58000.0              4.0
44.988739
1169  1170.0          51.0       1431.0   87000.0              1.0
44.508839

              lon
1317  10.76424026
940   12.49565029
120   14.19626999
805   12.49565029
832   16.60445023
...           ...
1429  10.68284988
756   13.76593971
340   12.49565029
1466  9.010499954
1169  11.46907997

[1075 rows x 7 columns]
```

In [39]:
```
model=LinearRegression()
model.fit(x_train,y_train)
model.intercept_
```

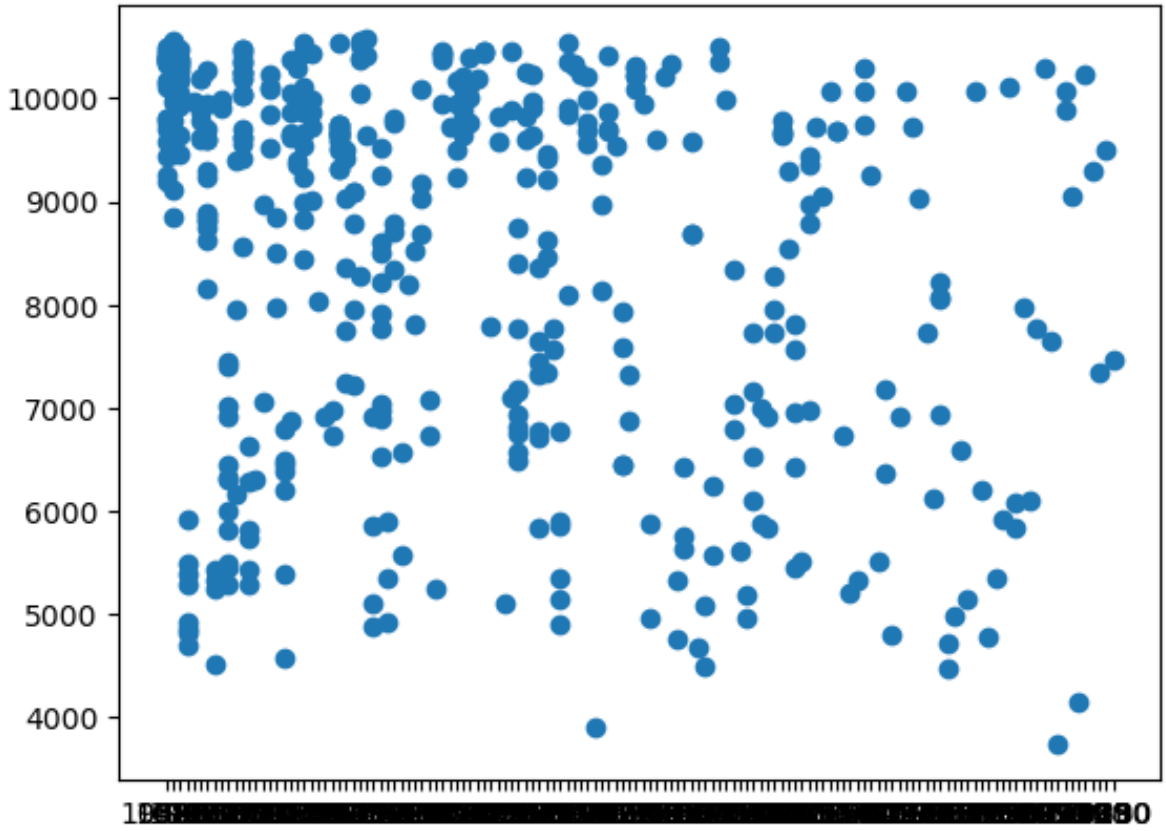
Out[39]:
```
8366.390730791745
```

In [40]:
```
model.coef_
```

Out[40]:
```
array([-2.45733778e-03,  8.04013394e+00, -8.79036602e-01, -1.83470
200e-02,
       -1.47760343e+00,  5.00300356e+01,  3.81983637e+00])
```

In [41]:
```
prediction=model.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[41]:
```
<matplotlib.collections.PathCollection at 0x7fda68de19f0>
```



In [42]:
```
model.score(x_test,y_test)
```

Out[42]:
```
0.855974762274458
```

In [ ]: