In [ ]:

In [29]:

```python
# IMPORT LIBRARIES
import numpy  as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [30]:

```python
a=pd.read_csv(r"C:\Users\user\Downloads\fiat500_VehicleSelection_Dataset.csv")
a
```

Out[30]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	l
0	1	lounge	51	882	25000	1	44.907242	8.6115
1	2	pop	51	1186	32500	1	45.666359	12.2418
2	3	sport	74	4658	142228	1	45.503300	11.4178
3	4	lounge	51	2739	160000	1	40.633171	17.6346
4	5	pop	73	3074	106880	1	41.903221	12.4956
...	...	...	...	...	...	...	...	
1533	1534	sport	51	3712	115280	1	45.069679	7.7049
1534	1535	lounge	74	3835	112000	1	45.845692	8.6668
1535	1536	pop	51	2223	60457	1	45.481541	9.4134
1536	1537	lounge	51	2557	80750	1	45.000702	7.6822
1537	1538	pop	51	1766	54276	1	40.323410	17.5682

1538 rows × 9 columns

In [31]:

```python
a=a.head(10)
a
```

Out[31]:

	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	
5	6	pop	74	3623	70225	1	45.000702	7.682270	
6	7	lounge	51	731	11600	1	44.907242	8.611560	1
7	8	lounge	51	1521	49076	1	41.903221	12.495650	
8	9	sport	73	4049	76000	1	45.548000	11.549470	
9	10	sport	51	3653	89000	1	45.438301	10.991700	

In [32]:

```python
# to find
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   ID               10 non-null     int64
 1   model            10 non-null     object
 2   engine_power     10 non-null     int64
 3   age_in_days      10 non-null     int64
 4   km               10 non-null     int64
 5   previous_owners  10 non-null     int64
 6   lat              10 non-null     float64
 7   lon              10 non-null     float64
 8   price            10 non-null     int64
dtypes: float64(2), int64(6), object(1)
memory usage: 848.0+ bytes
```

```
# to display summary of statastic
a.describe()
```

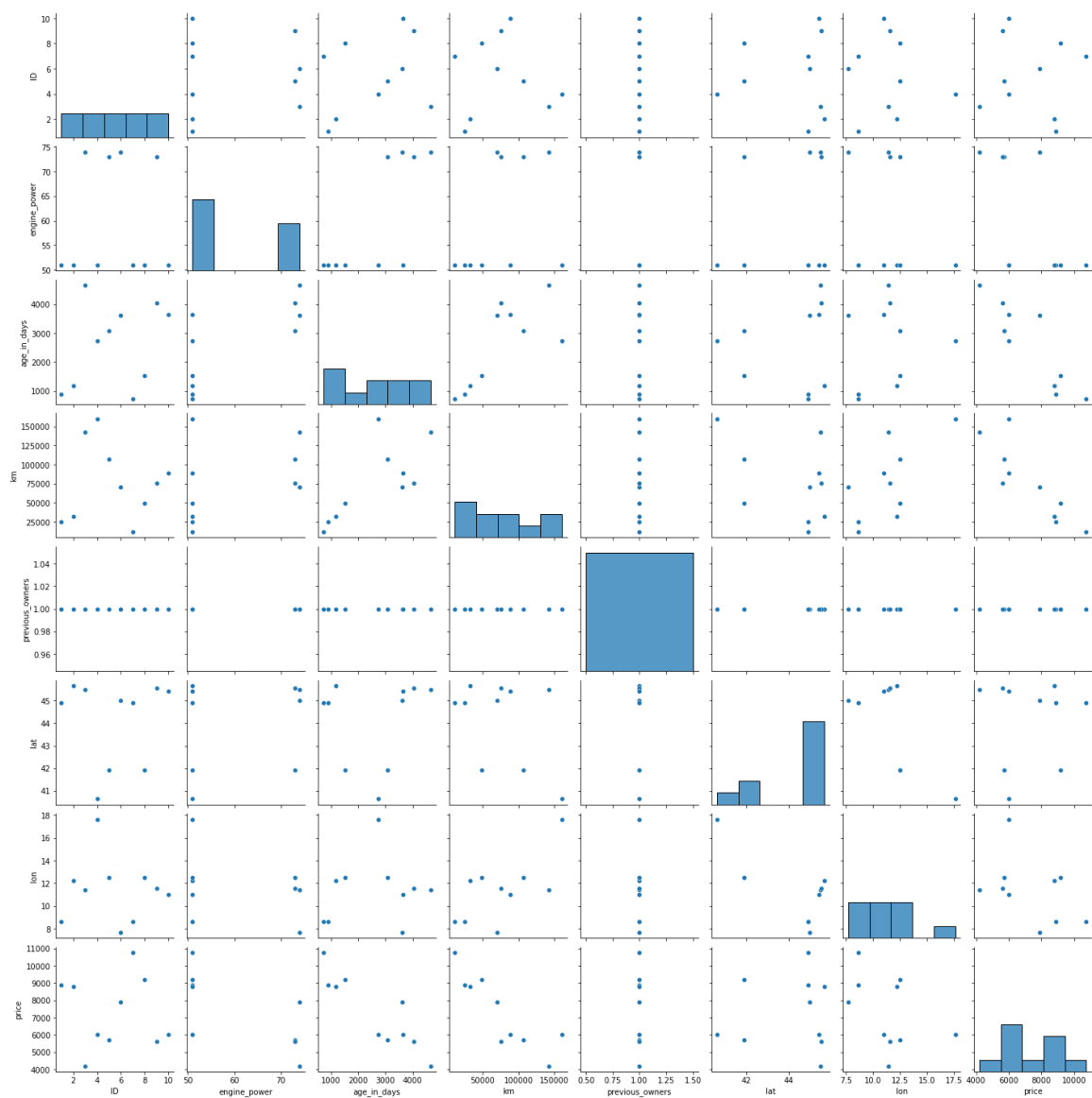|	ID	engine_power	age_in_days	km	previous_owners	lat	
count	10.00000	10.000000	10.000000	10.000000	10.0	10.000000	10.0
mean	5.50000	60.000000	2611.600000	76250.900000	1.0	44.141076	11.3
std	3.02765	11.623731	1427.557214	49399.679798	0.0	1.887936	2.8
min	1.00000	51.000000	731.000000	11600.000000	1.0	40.633171	7.0
25%	3.25000	51.000000	1269.750000	36644.000000	1.0	42.654226	9.2
50%	5.50000	51.000000	2906.500000	73112.500000	1.0	44.953972	11.4
75%	7.75000	73.000000	3645.500000	102410.000000	1.0	45.487050	12.4
max	10.00000	74.000000	4658.000000	160000.000000	1.0	45.666359	17.0

```
# to display colum heading
a.columns
```

```
Index(['ID', 'model', 'engine_power', 'age_in_days', 'km', 'previous_owner
s',
       'lat', 'lon', 'price'],
      dtype='object')
```
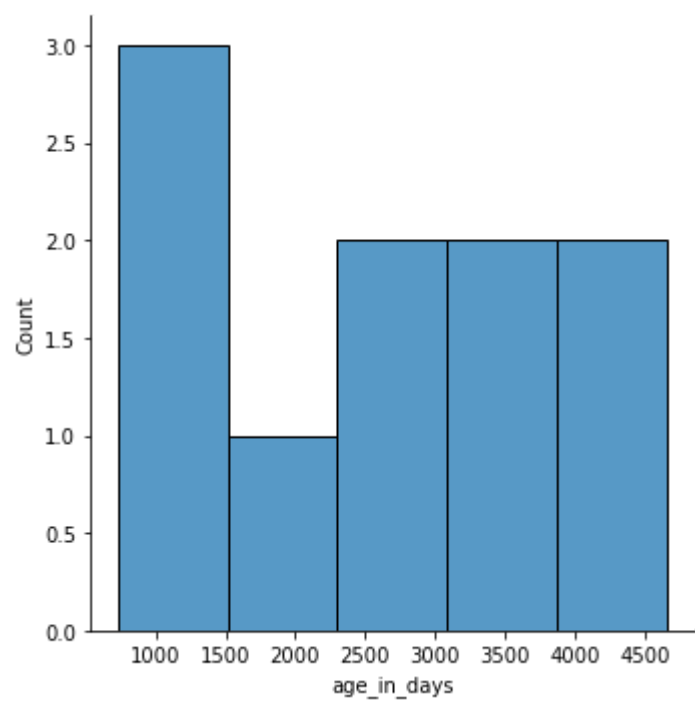
```
sns.pairplot(a)
```

```
<seaborn.axisgrid.PairGrid at 0x20b7d497c10>
```

```
sns.displot(a["age_in_days"])
```

```
<seaborn.axisgrid.FacetGrid at 0x20b0105f040>
```

```
b=a[['model', 'engine_power', 'age_in_days', 'km', 'previous_owners',
     'lat', 'lon', 'price']]
b
```

	model	engine_power	age_in_days	km	previous_owners	lat	lon	price
0	lounge	51	882	25000	1	44.907242	8.611560	8900
1	pop	51	1186	32500	1	45.666359	12.241890	8800
2	sport	74	4658	142228	1	45.503300	11.417840	4200
3	lounge	51	2739	160000	1	40.633171	17.634609	6000
4	pop	73	3074	106880	1	41.903221	12.495650	5700
5	pop	74	3623	70225	1	45.000702	7.682270	7900
6	lounge	51	731	11600	1	44.907242	8.611560	10750
7	lounge	51	1521	49076	1	41.903221	12.495650	9190
8	sport	73	4049	76000	1	45.548000	11.549470	5600
9	sport	51	3653	89000	1	45.438301	10.991700	6000

```
sns.heatmap(b.corr())
```

Out[44]:

```
<AxesSubplot:>
```


In [49]:

```
x=a[['age_in_days', 'km', 'previous_owners',
      'lat', 'lon', 'price']]
y=a['age_in_days']
```

In [50]:

```
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 [51]:

```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[51]:

```
LinearRegression()
```

In [52]:

```
lr.intercept_
```

Out[52]:

```
-9.094947017729282e-13
```

In [53]:

```python
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```
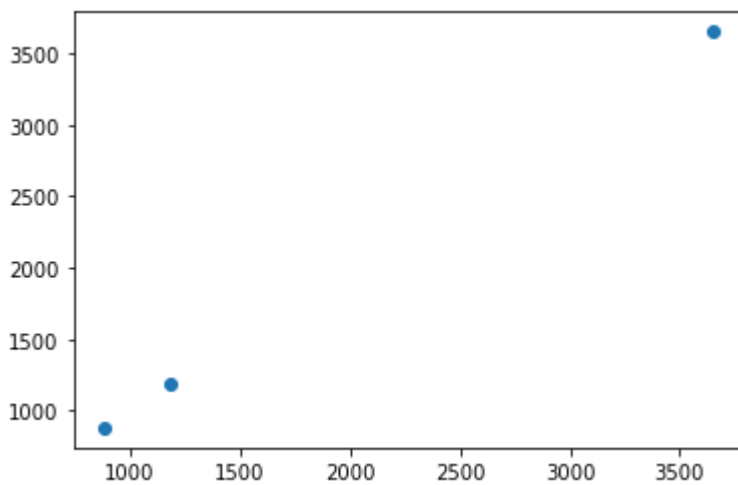
Out[53]:

	Co-efficient
age_in_days	1.000000e+00
km	-1.377314e-17
previous_owners	1.019426e-16
lat	8.115585e-14
lon	-2.991007e-14
price	-4.494383e-17

In [54]:

```python
prediction = lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[54]:

```
<matplotlib.collections.PathCollection at 0x20b005ec5e0>
```


In [55]:

```python
lr.score(x_test,y_test)
```

Out[55]:

```
1.0
```

In [56]:

```python
lr.score(x_train,y_train)
```

Out[56]:

```
1.0
```

In [57]:

```python
from sklearn.linear_model import Ridge,Lasso
```

In [58]:

```python
rr=Ridge(alpha=10)
rr.fit(x_test,y_test)
```

Out[58]:

Ridge(alpha=10)

In [59]:

```python
rr.score(x_test,y_test)
```

Out[59]:

0.9999999999952643

In [60]:

```python
la=Lasso(alpha=10)
la.fit(x_test,y_test)
```

Out[60]:

Lasso(alpha=10)

In [61]:

```python
la.score(x_test,y_test)
```

Out[61]:

0.9999999999999121

In [ ]: