In [24]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [25]:
```
df=pd.read_csv(r"C:\\Users\\Admin\\Downloads\\fiat500_VehicleSelection_Dataset - fiat500_VehicleSelection_Dataset
df
```

Out[25]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price	Unnamed: 9	Unnamed: 10
0	1.0	lounge	51.0	882.0	25000.0	1.0	44.907242	8.611559868	8900	NaN	NaN
1	2.0	pop	51.0	1186.0	32500.0	1.0	45.666359	12.24188995	8800	NaN	NaN
2	3.0	sport	74.0	4658.0	142228.0	1.0	45.503300	11.41784	4200	NaN	NaN
3	4.0	lounge	51.0	2739.0	160000.0	1.0	40.633171	17.63460922	6000	NaN	NaN
4	5.0	pop	73.0	3074.0	106880.0	1.0	41.903221	12.49565029	5700	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...
1544	NaN	NaN	NaN	NaN	NaN	NaN	NaN	length	5	NaN	NaN
1545	NaN	NaN	NaN	NaN	NaN	NaN	NaN	concat	lonprice	NaN	NaN
1546	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Null values	NO	NaN	NaN
1547	NaN	NaN	NaN	NaN	NaN	NaN	NaN	find	1	NaN	NaN
1548	NaN	NaN	NaN	NaN	NaN	NaN	NaN	search	1	NaN	NaN

1549 rows × 11 columns

In [26]:
```
df.head()
```

Out[26]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price	Unnamed: 9	Unnamed: 10
0	1.0	lounge	51.0	882.0	25000.0	1.0	44.907242	8.611559868	8900	NaN	NaN
1	2.0	pop	51.0	1186.0	32500.0	1.0	45.666359	12.24188995	8800	NaN	NaN
2	3.0	sport	74.0	4658.0	142228.0	1.0	45.503300	11.41784	4200	NaN	NaN
3	4.0	lounge	51.0	2739.0	160000.0	1.0	40.633171	17.63460922	6000	NaN	NaN
4	5.0	pop	73.0	3074.0	106880.0	1.0	41.903221	12.49565029	5700	NaN	NaN

In [27]:
```
df.describe()
```

Out[27]:

	ID	engine_power	age_in_days	km	previous_owners	lat	Unnamed: 9
count	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000	0.0
mean	769.500000	51.904421	1650.980494	53396.011704	1.123537	43.541361	NaN
std	444.126671	3.988023	1289.522278	40046.830723	0.416423	2.133518	NaN
min	1.000000	51.000000	366.000000	1232.000000	1.000000	36.855839	NaN
25%	385.250000	51.000000	670.000000	20006.250000	1.000000	41.802990	NaN
50%	769.500000	51.000000	1035.000000	39031.000000	1.000000	44.394096	NaN
75%	1153.750000	51.000000	2616.000000	79667.750000	1.000000	45.467960	NaN
max	1538.000000	77.000000	4658.000000	235000.000000	4.000000	46.795612	NaN

In [31]:
```
df.info()
```

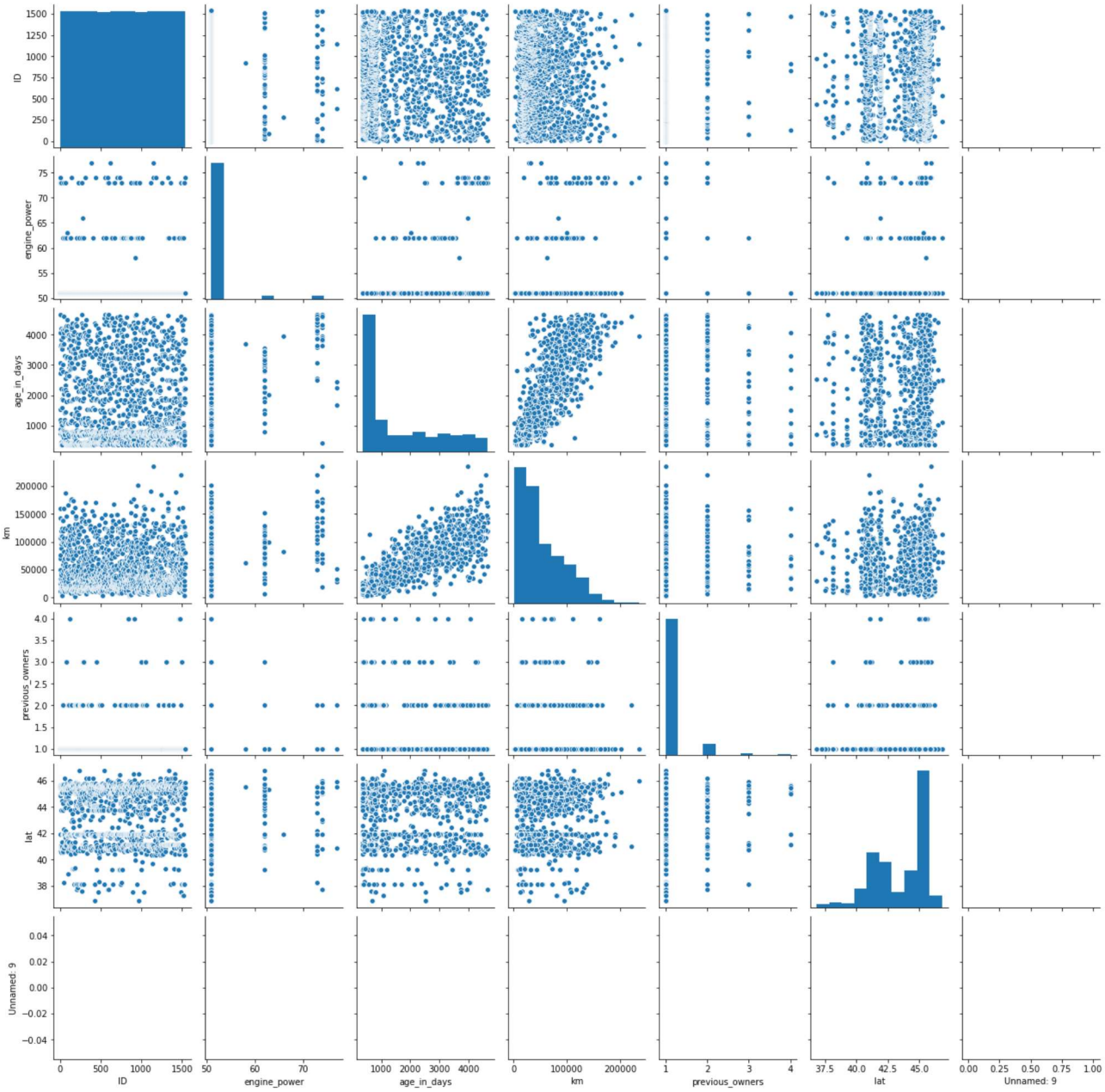
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1549 entries, 0 to 1548
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   ID               1538 non-null   float64
 1   model            1538 non-null   object 
 2   engine_power     1538 non-null   float64
 3   age_in_days      1538 non-null   float64
 4   km               1538 non-null   float64
 5   previous_owners  1538 non-null   float64
 6   lat              1538 non-null   float64
 7   lon              1549 non-null   object 
 8   price            1549 non-null   object 
 9   Unnamed: 9       0 non-null      float64
 10  Unnamed: 10      1 non-null      object 
dtypes: float64(7), object(4)
memory usage: 133.2+ KB
```

In [29]:
```
df.columns
```

Out[29]:
```
Index(['ID', 'model', 'engine_power', 'age_in_days', 'km', 'previous_owners',
       'lat', 'lon', 'price', 'Unnamed: 9', 'Unnamed: 10'],
      dtype='object')
```

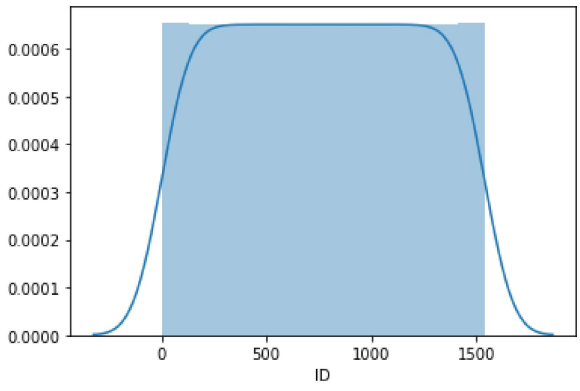
In [40]:
```
sns.pairplot(df)
```

Out[40]:
```
<seaborn.axisgrid.PairGrid at 0x1cfd53f7250>
```



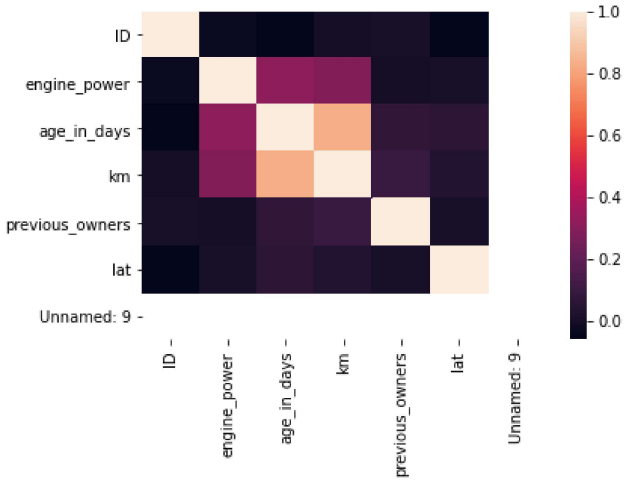
In [42]:
```
sns.distplot(df['ID'])
```

Out[42]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1cfd69ff8b0>
```



In [43]:
```
sns.heatmap(df.corr())
```

Out[43]: <matplotlib.axes._subplots.AxesSubplot at 0x1cfd70b6df0>



In [44]:
```
df1=df.fillna(value=1)
```

In [49]:
```
x=df1[[ 'age_in_days']]
y=df1['km']
```

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=0.4)
```

In [51]:
```
from sklearn.linear_model import LinearRegression
lr= LinearRegression()
lr.fit(x_train,y_train)
```

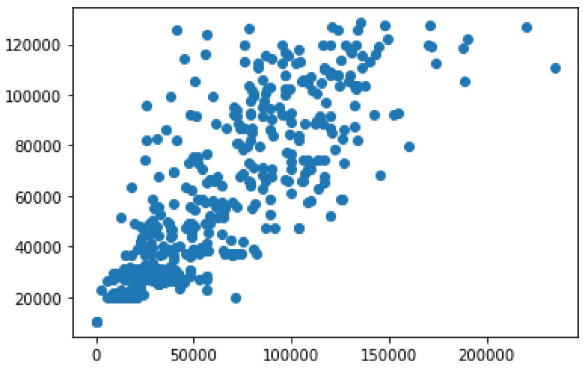
Out[51]: LinearRegression()

In [52]:
```
print(lr.intercept_)
```

```
10372.59539050042
```

In [53]:
```
prediction= lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[53]: <matplotlib.collections.PathCollection at 0x1cfd81b59d0>



In [54]:
```
print(lr.score(x_test,y_test))
```

```
0.6901383300701873
```

In [55]:
```
print(lr.score(x_train,y_train))
```

```
0.702852768612902
```

In [56]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [57]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[57]:
```
Ridge(alpha=10)
```

In [58]:
```
rr.score(x_test,y_test)
```

Out[58]:
```
0.6901383296110324
```

In [59]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[59]:
```
Lasso(alpha=10)
```

In [60]:
```
la.score(x_test,y_test)
```

Out[60]:
```
0.6901383132615997
```

In [ ]: