

```
In [33]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [11]: df=pd.read_csv("1_fiat500_VehicleSelection_Dataset.csv")
df
```

```
Out[11]:
```

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon
0	1.0	lounge	51.0	882.0	25000.0	1.0	44.907242	8.6115598
1	2.0	pop	51.0	1186.0	32500.0	1.0	45.666359	12.241889
2	3.0	sport	74.0	4658.0	142228.0	1.0	45.503300	11.417
3	4.0	lounge	51.0	2739.0	160000.0	1.0	40.633171	17.634609
4	5.0	pop	73.0	3074.0	106880.0	1.0	41.903221	12.495650
...
1544	NaN	NaN	NaN	NaN	NaN	NaN	NaN	lenq
1545	NaN	NaN	NaN	NaN	NaN	NaN	NaN	conu
1546	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Null valu
1547	NaN	NaN	NaN	NaN	NaN	NaN	NaN	fi
1548	NaN	NaN	NaN	NaN	NaN	NaN	NaN	sear

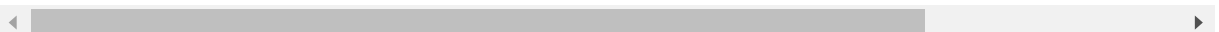
1549 rows × 11 columns



```
In [12]: df.head()
```

```
Out[12]:
```

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



```
In [13]: 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 [14]: df.columns
```

```
Out[14]: Index(['ID', 'model', 'engine_power', 'age_in_days', 'km', 'previous_owners',
               'lat', 'lon', 'price', 'Unnamed: 9', 'Unnamed: 10'],
              dtype='object')
```

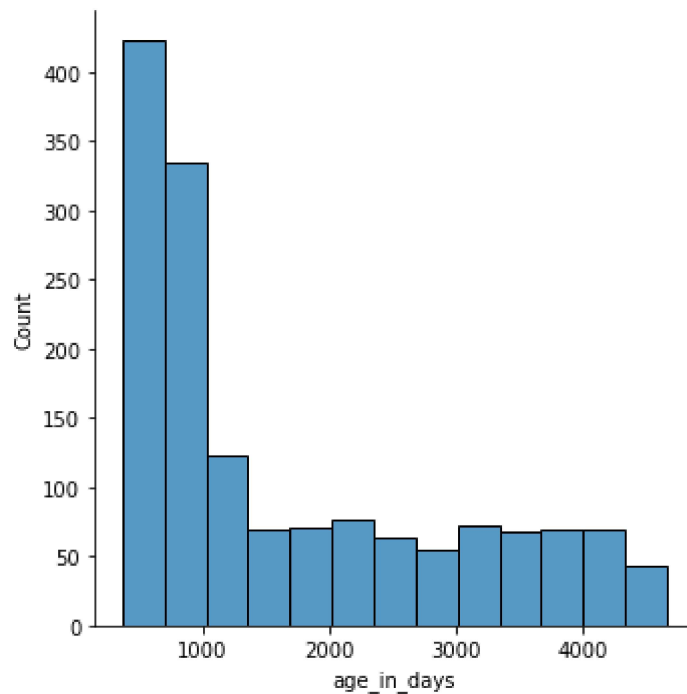
```
In [15]: sns.pairplot(df)
```

```
Out[15]: <seaborn.axisgrid.PairGrid at 0x224c42ff2e0>
```



```
In [16]: sns.displot(df['age_in_days'])
```

```
Out[16]: <seaborn.axisgrid.FacetGrid at 0x224c82ae430>
```

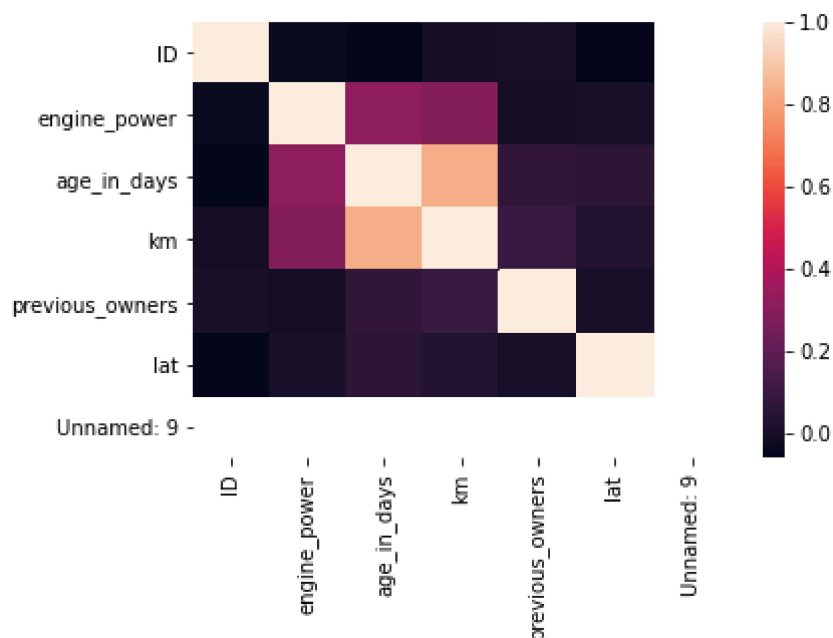


```
In [17]: df1=df.drop(['Unnamed: 10'],axis=1)
df1
df1=df1.drop(df1.index[1537:])
df1.isna().sum()
```

```
Out[17]: ID                0
model                  0
engine_power           0
age_in_days            0
km                     0
previous_owners        0
lat                    0
lon                    0
price                  0
Unnamed: 9             1537
dtype: int64
```

```
In [18]: sns.heatmap(df1.corr())
```

```
Out[18]: <AxesSubplot:>
```



```
In [19]: from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

```
In [20]: df1.isna().sum()
```

```
Out[20]: ID          0
model          0
engine_power   0
age_in_days    0
km             0
previous_owners 0
lat           0
lon           0
price         0
Unnamed: 9     1537
dtype: int64
```

```
In [28]: 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	1
at \						
796	797.0	51.0	701.0	20300.0	1.0	45.6882
59						
932	933.0	62.0	3347.0	94000.0	1.0	41.9032
21						
816	817.0	51.0	852.0	37657.0	1.0	43.2068
79						
1089	1090.0	51.0	882.0	33160.0	1.0	45.7789
99						
572	573.0	51.0	366.0	11236.0	1.0	45.4381
10						
...	
...						
582	583.0	51.0	1066.0	59961.0	1.0	41.8029
90						
518	519.0	51.0	1705.0	65000.0	1.0	45.4635
01						
1244	1245.0	51.0	366.0	5870.0	1.0	41.0833
...						

```
In [29]: model=LinearRegression()
model.fit(x_train,y_train)
model.intercept_
```

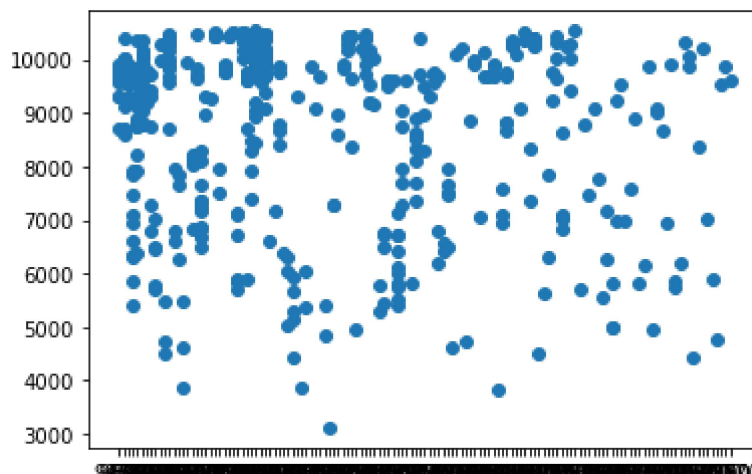
Out[29]: 8953.127402723812

```
In [30]: model.coef_
```

Out[30]: array([-7.30863058e-02, 9.88444121e+00, -8.95842168e-01, -1.76023717e-02,
 9.18435711e+00, 3.43008270e+01, 8.33632411e+00])

```
In [34]: prediction=model.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[34]: <matplotlib.collections.PathCollection at 0x224c942ebe0>



```
In [35]: model.score(x_test,y_test)
```

```
Out[35]: 0.8371453219092423
```

```
In [36]: from sklearn.linear_model import Ridge,Lasso
```

```
In [37]: rr=Ridge(alpha=10)  
rr.fit(x_train,y_train)
```

```
Out[37]: Ridge(alpha=10)
```

```
In [38]: rr.score(x_test,y_test)
```

```
Out[38]: 0.8371411962466114
```

```
In [39]: la=Lasso(alpha=10)  
la.fit(x_train,y_train)
```

```
Out[39]: Lasso(alpha=10)
```

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

```
Out[40]: 0.8370319421895127
```

```
In [ ]:
```