

```
In [68]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

In [69]: data=pd.read_csv(r"C:\Users\user\Downloads\fiat500_VehicleSelection_Dataset (1
data
```

Out[69]:

	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
...	
1544	NaN	NaN	NaN	NaN	NaN	NaN	NaN	len
1545	NaN	NaN	NaN	NaN	NaN	NaN	NaN	cor
1546	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Null val
1547	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1548	NaN	NaN	NaN	NaN	NaN	NaN	NaN	sea

1549 rows × 11 columns

```
In [70]: df=data.head(100)
df
```

Out[70]:

	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	8.61155986
1	2.0	pop	51.0	1186.0	32500.0	1.0	45.666359	12.2418896
2	3.0	sport	74.0	4658.0	142228.0	1.0	45.503300	11.4178
3	4.0	lounge	51.0	2739.0	160000.0	1.0	40.633171	17.6346096
4	5.0	pop	73.0	3074.0	106880.0	1.0	41.903221	12.4956506
...
95	96.0	sport	51.0	4292.0	165600.0	1.0	44.715408	11.3083006
96	97.0	pop	51.0	1066.0	28000.0	1.0	41.769051	12.6628106
97	98.0	sport	51.0	2009.0	86000.0	2.0	40.633171	17.6346096
98	99.0	lounge	51.0	456.0	18592.0	2.0	45.393600	10.4822396
99	100.0	pop	51.0	731.0	41558.0	2.0	45.571220	9.15913966

100 rows × 11 columns



```
In [71]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 11 columns):
#   Column          Non-Null Count  Dtype  
---  -
0   ID              100 non-null   float64
1   model           100 non-null   object  
2   engine_power    100 non-null   float64
3   age_in_days     100 non-null   float64
4   km              100 non-null   float64
5   previous_owners 100 non-null   float64
6   lat             100 non-null   float64
7   lon             100 non-null   object  
8   price           100 non-null   object  
9   Unnamed: 9      0 non-null     float64
10  Unnamed: 10     0 non-null     object  
dtypes: float64(7), object(4)
memory usage: 8.7+ KB
```

In [72]: `df.describe()`

Out[72]:

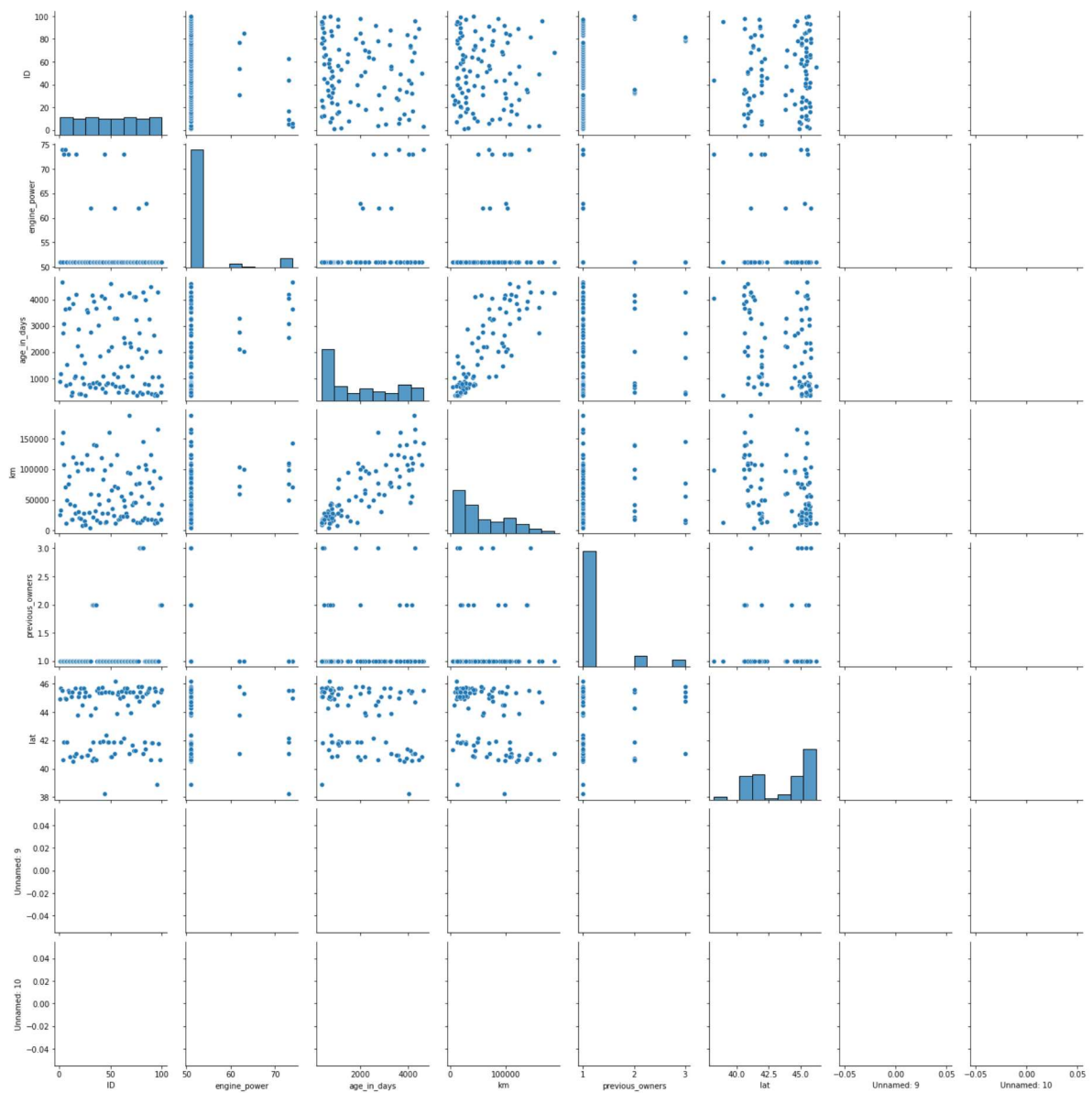
	ID	engine_power	age_in_days	km	previous_owners	lat	Un
count	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	
mean	50.500000	53.010000	1935.300000	58812.180000	1.180000	43.612648	
std	29.011492	6.014284	1414.251278	44728.034639	0.500101	2.083451	
min	1.000000	51.000000	366.000000	4000.000000	1.000000	38.218128	
25%	25.750000	51.000000	723.500000	19781.750000	1.000000	41.744165	
50%	50.500000	51.000000	1446.000000	44032.000000	1.000000	44.831066	
75%	75.250000	51.000000	3265.500000	95075.750000	1.000000	45.396568	
max	100.000000	74.000000	4658.000000	188000.000000	3.000000	46.176498	

In [73]: `df.columns`

Out[73]: Index(['ID', 'model', 'engine_power', 'age_in_days', 'km', 'previous_owners', 'lat', 'lon', 'price', 'Unnamed: 9', 'Unnamed: 10'], dtype='object')

```
In [74]: sns.pairplot(df)
```

```
Out[74]: <seaborn.axisgrid.PairGrid at 0x1ed6b9dbd60>
```



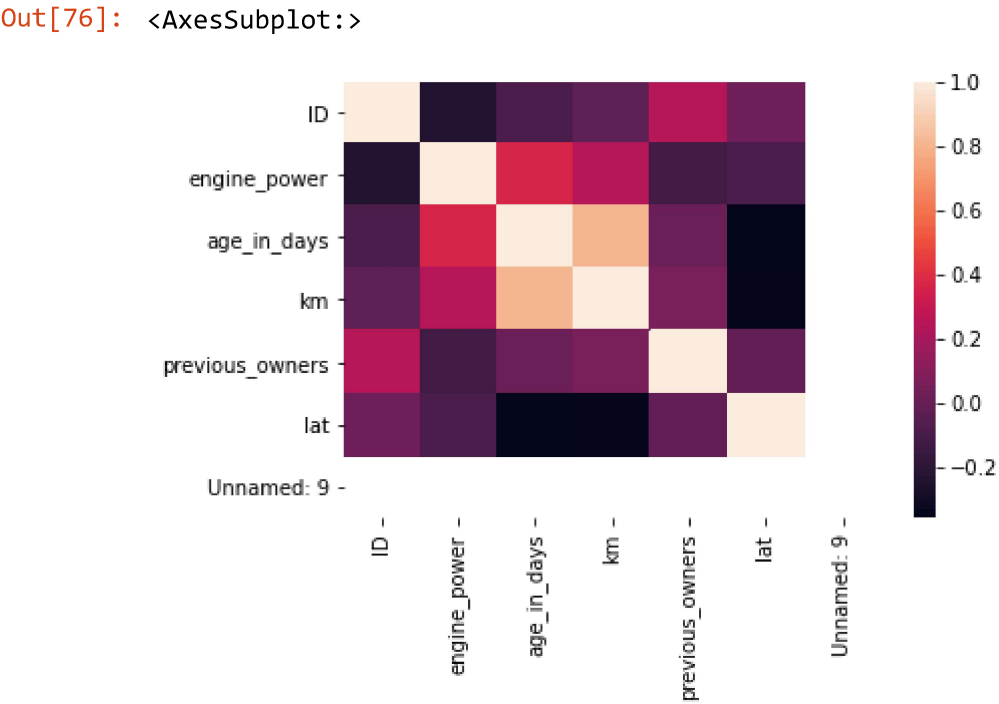
```
In [75]: da=df[['ID', 'model', 'engine_power', 'age_in_days', 'km', 'previous_owners',
               'lat', 'lon', 'price', 'Unnamed: 9', 'Unnamed: 10']]
da
```

Out[75]:

	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	8.61155986
1	2.0	pop	51.0	1186.0	32500.0	1.0	45.666359	12.2418896
2	3.0	sport	74.0	4658.0	142228.0	1.0	45.503300	11.4178
3	4.0	lounge	51.0	2739.0	160000.0	1.0	40.633171	17.6346092
4	5.0	pop	73.0	3074.0	106880.0	1.0	41.903221	12.4956502
...
95	96.0	sport	51.0	4292.0	165600.0	1.0	44.715408	11.3083006
96	97.0	pop	51.0	1066.0	28000.0	1.0	41.769051	12.6628102
97	98.0	sport	51.0	2009.0	86000.0	2.0	40.633171	17.6346092
98	99.0	lounge	51.0	456.0	18592.0	2.0	45.393600	10.4822392
99	100.0	pop	51.0	731.0	41558.0	2.0	45.571220	9.15913962

100 rows × 11 columns

```
In [76]: sns.heatmap(da.corr())
```



```
In [78]: x=df[['engine_power', 'age_in_days', 'km', 'lat']]
y=df['price']
```

```
In [79]: 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 [80]: from sklearn.linear_model import LinearRegression

lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[80]: LinearRegression()

```
In [81]: print(lr.intercept_)

10078.100390681766
```

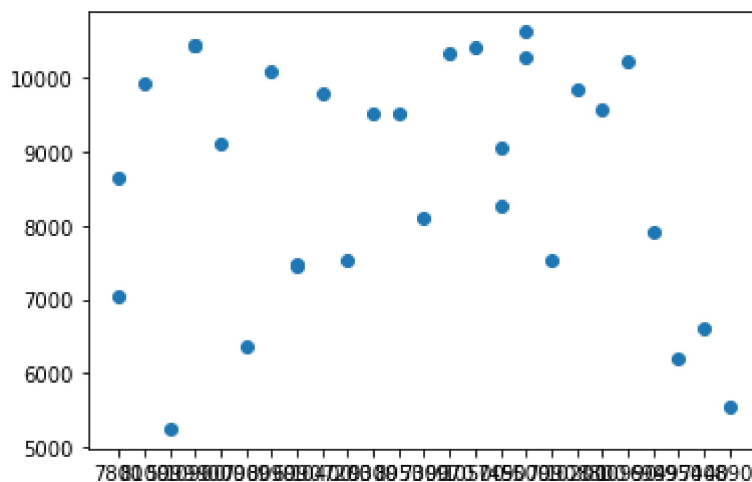
```
In [82]: coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[82]:

	Co-efficient
engine_power	-18.563262
age_in_days	-0.877070
km	-0.016371
lat	43.956990

```
In [83]: prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[83]: <matplotlib.collections.PathCollection at 0x1ed7260f670>



```
In [84]: print(lr.score(x_test,y_test))
```

0.7320821613627828

```
In [85]: print(lr.score(x_train,y_train))
```

0.93328163999647

```
In [86]: from sklearn.linear_model import Ridge,Lasso
```

```
In [87]: rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[87]: Ridge(alpha=10)

```
In [88]: rr.score(x_test,y_test)
```

Out[88]: 0.7320400698949787

```
In [89]: la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[89]: Lasso(alpha=10)

```
In [90]: la.score(x_test,y_test)
```

Out[90]: 0.7322147171785393

```
In [ ]:
```