27ass1

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

In [2]:
```
s=pd.read_csv(r"C:\Users\user\Downloads\fiat500_VehicleSelection_Dataset - fiat
s
```

Out[2]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	l
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	leng
1545	NaN	NaN	NaN	NaN	NaN	NaN	NaN	conc
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 [3]:
```
s=s.head(100)
s
```

Out[3]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lo
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	12.2418899
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	11.3083000
96	97.0	pop	51.0	1066.0	28000.0	1.0	41.769051	12.6628103
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	10.4822397
99	100.0	pop	51.0	731.0	41558.0	2.0	45.571220	9.15913963

100 rows × 11 columns

In [4]:
```
s.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 [5]:
```
s.describe()
```

Out[5]:

	ID	engine_power	age_in_days	km	previous_owners	lat	Unr
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 [6]:
```
s.columns
```

Out[6]:
```
Index(['ID', 'model', 'engine_power', 'age_in_days', 'km', 'previous_owners',
       'lat', 'lon', 'price', 'Unnamed: 9', 'Unnamed: 10'],
      dtype='object')
```

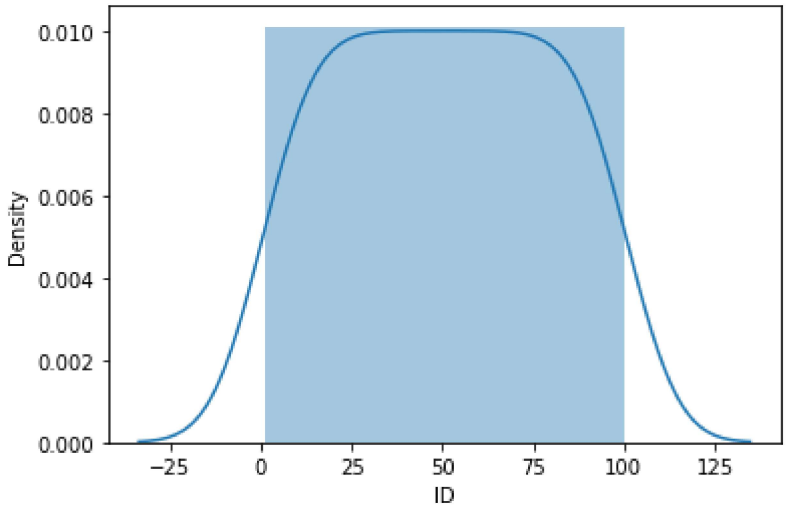
In [9]:
```
sns.pairplot(s)
```

Out[9]:
```
<seaborn.axisgrid.PairGrid at 0x2af16920610>
```

In [8]:
```
sns.distplot(s['ID'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: Fut
ureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function for hi
stograms).
  warnings.warn(msg, FutureWarning)
```

Out[8]: <AxesSubplot:xlabel='ID', ylabel='Density'>



In [11]:
```
s1=s[['ID','engine_power','age_in_days','km', 'previous_owners','lat']]
s
```

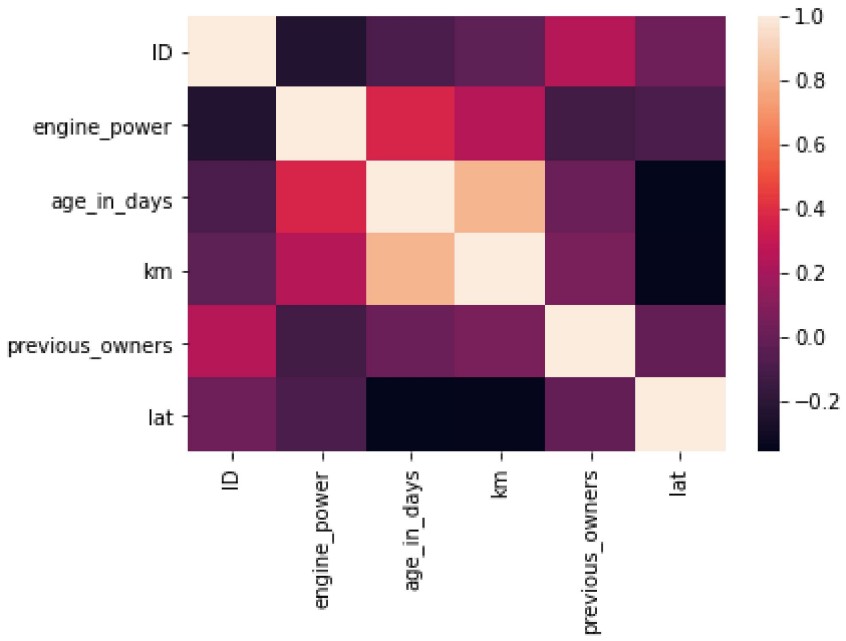
Out[11]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lo
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	12.2418899
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	11.3083000
96	97.0	pop	51.0	1066.0	28000.0	1.0	41.769051	12.6628103
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	10.4822397
99	100.0	pop	51.0	731.0	41558.0	2.0	45.571220	9.15913963

100 rows × 11 columns

In [12]:
```
sns.heatmap(s1.corr())
```

Out[12]: <AxesSubplot:>



In [13]:
```
x=s1[['ID','engine_power','age_in_days','km', 'previous_owners']]
y=s1['lat']
```

In [14]:
```
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 [15]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[15]: LinearRegression()

In [16]:
```
lr.intercept_
```

Out[16]: 45.581529441997134

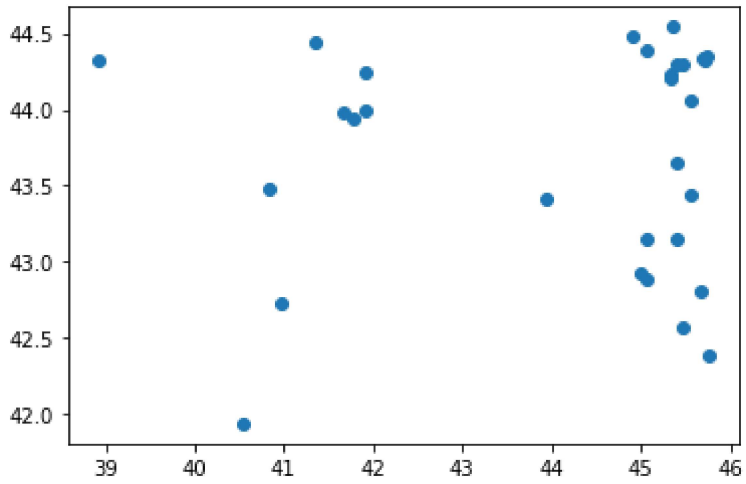
In [17]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[17]:

	Co-efficient
ID	-0.003698
engine_power	-0.002162
age_in_days	-0.000456
km	-0.000003
previous_owners	-0.594245

In [18]:
```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[18]: <matplotlib.collections.PathCollection at 0x2af23055d60>



In [19]:
```
print(lr.score(x_test,y_test))
```

```
-0.12260963124395263
```

In [ ]: