In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import preprocessing,svm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [2]:
```
df=pd.read_csv(r"C:\Users\91720\Downloads\fiat500_VehicleSelection_Dataset (1).csv")
df
```

Out[2]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon	price
0	1	lounge	51	882	25000	1	44.907242	8.611560	8900
1	2	pop	51	1186	32500	1	45.666359	12.241890	8800
2	3	sport	74	4658	142228	1	45.503300	11.417840	4200
3	4	lounge	51	2739	160000	1	40.633171	17.634609	6000
4	5	pop	73	3074	106880	1	41.903221	12.495650	5700
...	...	...	...	...	...	...	...	...	...
1533	1534	sport	51	3712	115280	1	45.069679	7.704920	5200
1534	1535	lounge	74	3835	112000	1	45.845692	8.666870	4600
1535	1536	pop	51	2223	60457	1	45.481541	9.413480	7500
1536	1537	lounge	51	2557	80750	1	45.000702	7.682270	5990
1537	1538	pop	51	1766	54276	1	40.323410	17.568270	7900

1538 rows × 9 columns

In [3]:
```
df=df[['lon','price']]
df.columns=['l','Pri']
```

In [4]:
```
df.head(10)
```

Out[4]:

	l	Pri
0	8.611560	8900
1	12.241890	8800
2	11.417840	4200
3	17.634609	6000
4	12.495650	5700
5	7.682270	7900
6	8.611560	10750
7	12.495650	9190
8	11.549470	5600
9	10.991700	6000

In [5]:
```
df.describe()
```

Out[5]:

	l	Pri
count	1538.000000	1538.000000
mean	11.563428	8576.003901
std	2.328190	1939.958641
min	7.245400	2500.000000
25%	9.505090	7122.500000
50%	11.869260	9000.000000
75%	12.769040	10000.000000
max	18.365520	11100.000000

In [6]:

```
df.info()
```

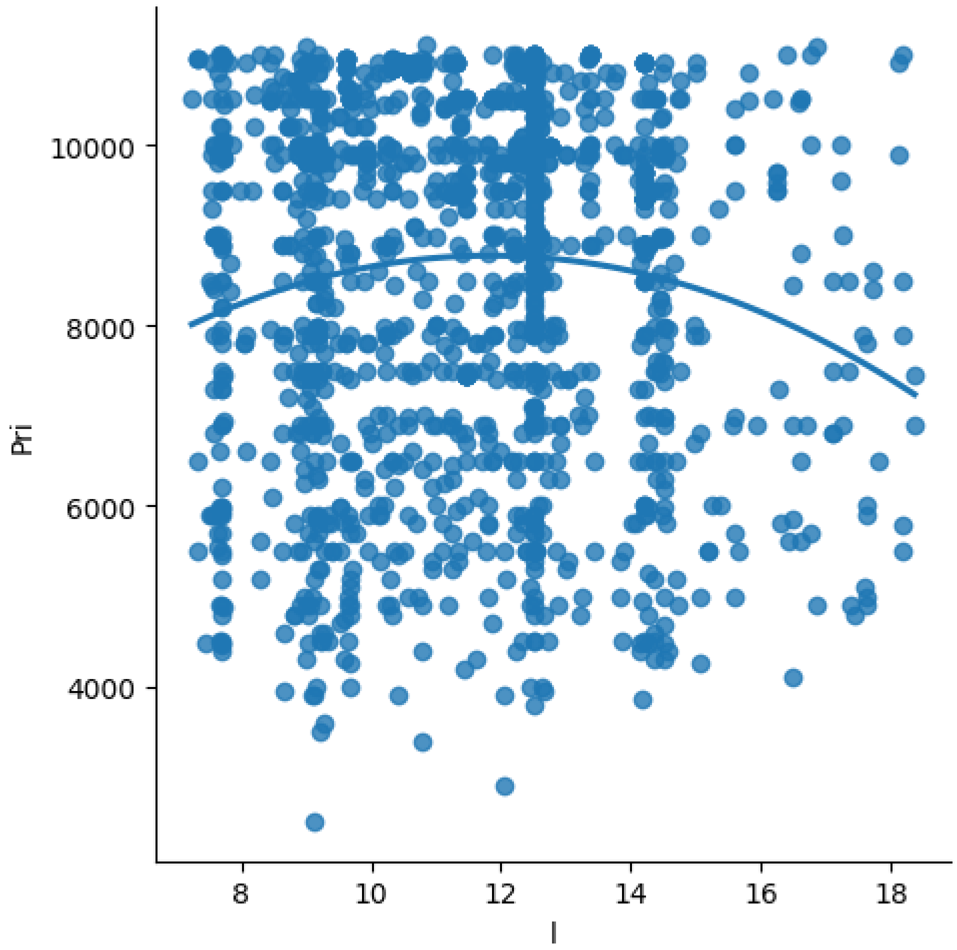
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1538 entries, 0 to 1537
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   l       1538 non-null   float64
 1   Pri     1538 non-null   int64  
dtypes: float64(1), int64(1)
memory usage: 24.2 KB
```

In [7]:

```
sns.lmplot(x="l",y="Pri",data=df,order=2,ci=None)
```

Out[7]:

```
<seaborn.axisgrid.FacetGrid at 0x205dc53fe50>
```



In [15]:

```
x=np.array(df['l']).reshape(-1,1)
y=np.array(df['Pri']).reshape(-1,1)
df.dropna()
```

Out[15]:

	l	Pri
0	8.611560	8900
1	12.241890	8800
2	11.417840	4200
3	17.634609	6000
4	12.495650	5700
...	...	...
1533	7.704920	5200
1534	8.666870	4600
1535	9.413480	7500
1536	7.682270	5990
1537	17.568270	7900

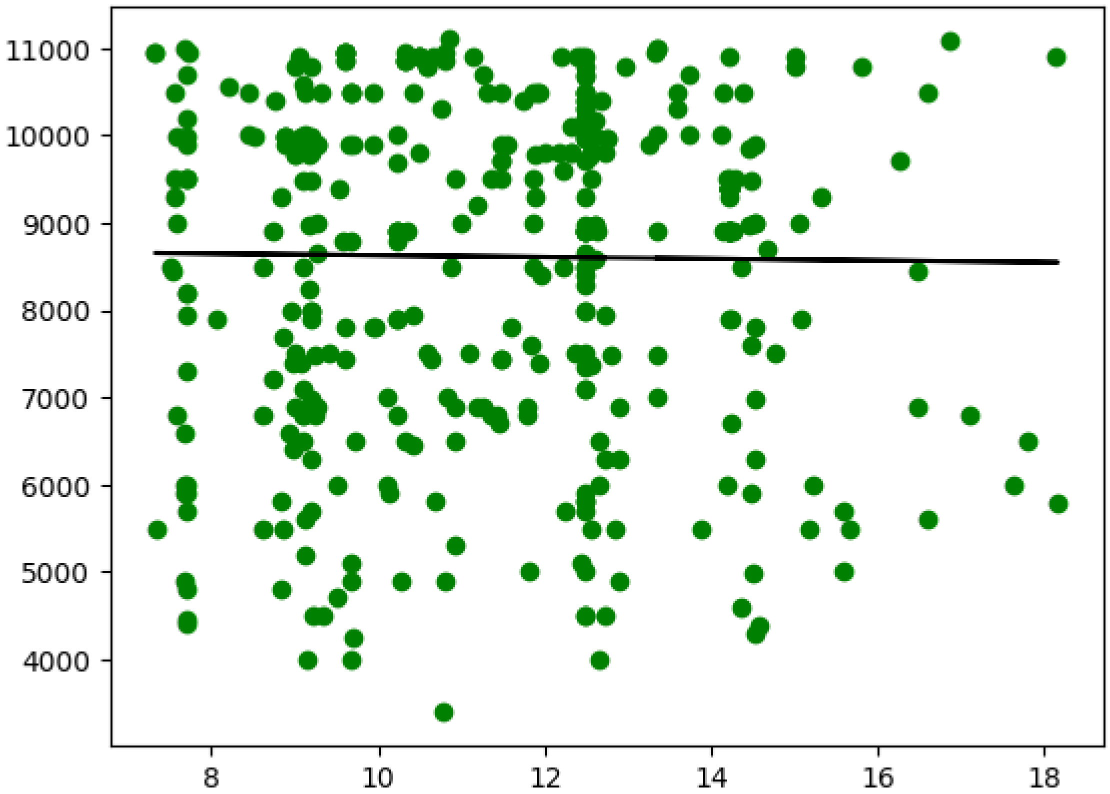
1538 rows × 2 columns

In [9]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
regr=LinearRegression()
regr.fit(x_train,y_train)
print(regr.score(x_test,y_test))
```

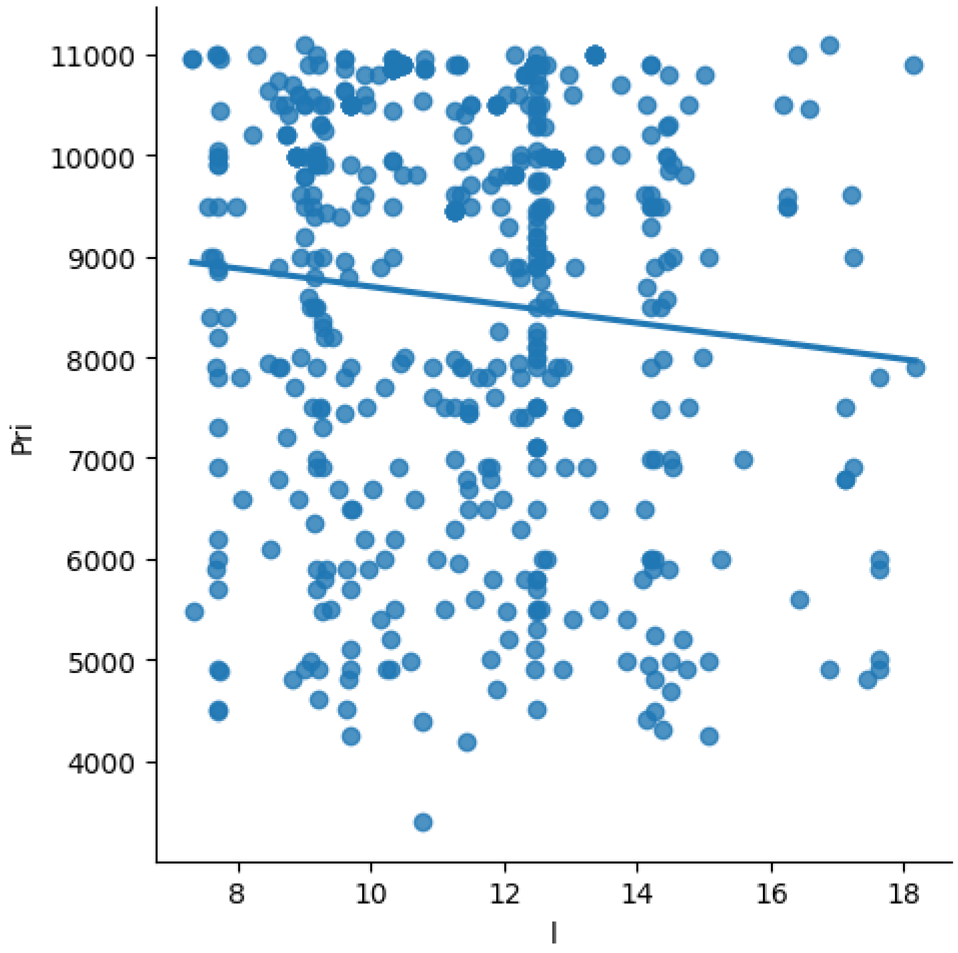
```
-0.006644220228226416
```

In [10]:
```
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='g')
plt.plot(x_test,y_pred,color='k')
plt.show()
```



In [11]:
```
df500=df[:][:500]
sns.lmplot(x="l",y="Pri",data=df500,order=1,ci=None)
```

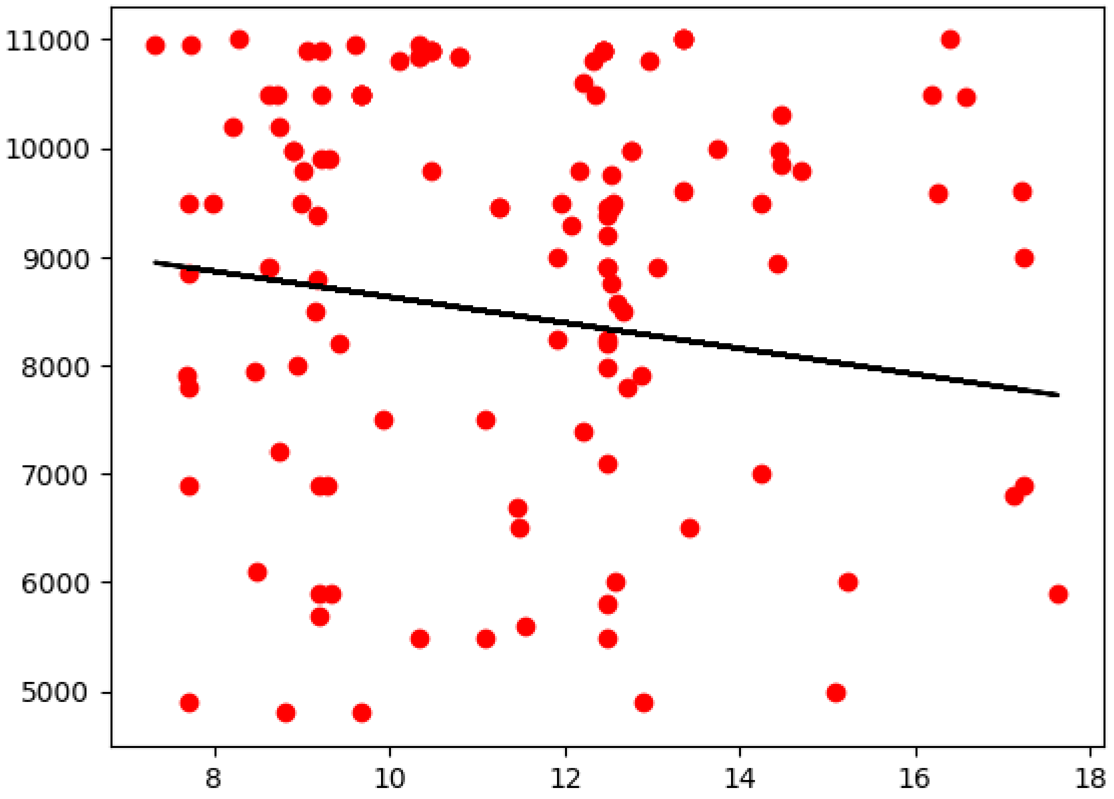
Out[11]:
```
<seaborn.axisgrid.FacetGrid at 0x205dc559d50>
```



In [12]:

```
df500.fillna(method='ffill',inplace=True)
x=np.array(df500['l']).reshape(-1,1)
y=np.array(df500['Pri']).reshape(-1,1)
df500.dropna(inplace=True)
X_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
regr=LinearRegression()
regr.fit(X_train,y_train)
print("Regression:",regr.score(x_test,y_test))
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='r')
plt.plot(x_test,y_pred,color='k')
plt.show()
```

```
Regression: -0.07668022718598899
```



In [14]:

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
model=LinearRegression()
model.fit(X_train,y_train)
y_pred=model.predict(x_test)
r2=r2_score(y_test,y_pred)
print("R2_score:",r2)
```

```
R2_score: -0.07668022718598899
```

concluse

```
          Dataset we have taken is poor for linear model but with the smaller data works well with linear mode
l.
```