In [10]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [11]:
```
from sklearn import preprocessing,svm
```

In [12]:
```
df=pd.read_csv(r"C:\Users\my pc\Downloads\fiat500_VehicleSelection_Dataset.csv
df
```

Out[12]:

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

1538 rows × 9 columns

In [13]:
```
df=df[["engine_power","price"]]
```

In [14]:
```
df.columns=["power","price"]
df
```

Out[14]:

	power	price
0	51	8900
1	51	8800
2	74	4200
3	51	6000
4	73	5700
...	...	...
1533	51	5200
1534	74	4600
1535	51	7500
1536	51	5990
1537	51	7900

1538 rows × 2 columns

In [15]:
```
df.head()
```

Out[15]:

	power	price
0	51	8900
1	51	8800
2	74	4200
3	51	6000
4	73	5700

In [16]:
```
df.describe()
```

Out[16]:

	power	price
count	1538.000000	1538.000000
mean	51.904421	8576.003901
std	3.988023	1939.958641
min	51.000000	2500.000000
25%	51.000000	7122.500000
50%	51.000000	9000.000000
75%	51.000000	10000.000000
max	77.000000	11100.000000

In [20]:
```
df.size
```

Out[20]:
```
3076
```

In [21]:
```
df.shape
```

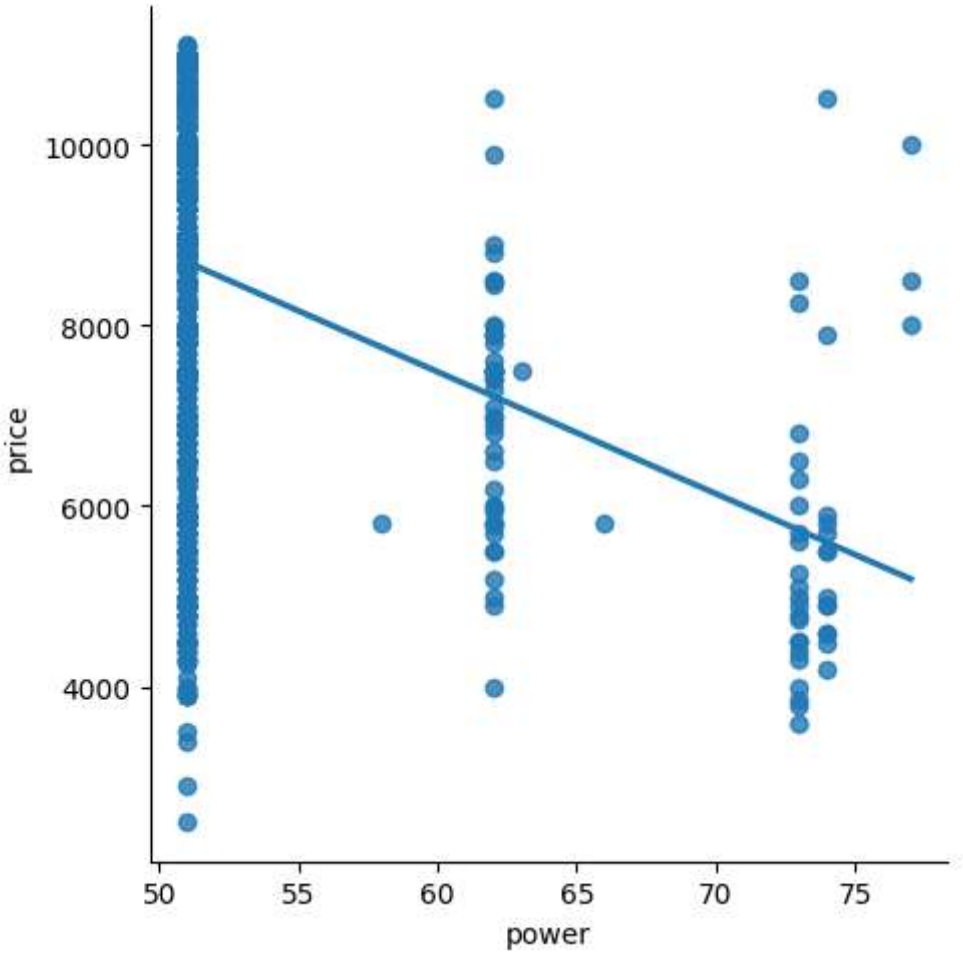
Out[21]:
```
(1538, 2)
```

In [17]:
```
df.isna().any()
```

Out[17]:
```
power    False
price    False
dtype: bool
```

In [18]:
```
sns.lmplot(x="power",y="price",data=df,order=1,ci=None)
```

Out[18]:
```
<seaborn.axisgrid.FacetGrid at 0x2b3ce2e4850>
```



In [22]:
```
df.info()
```

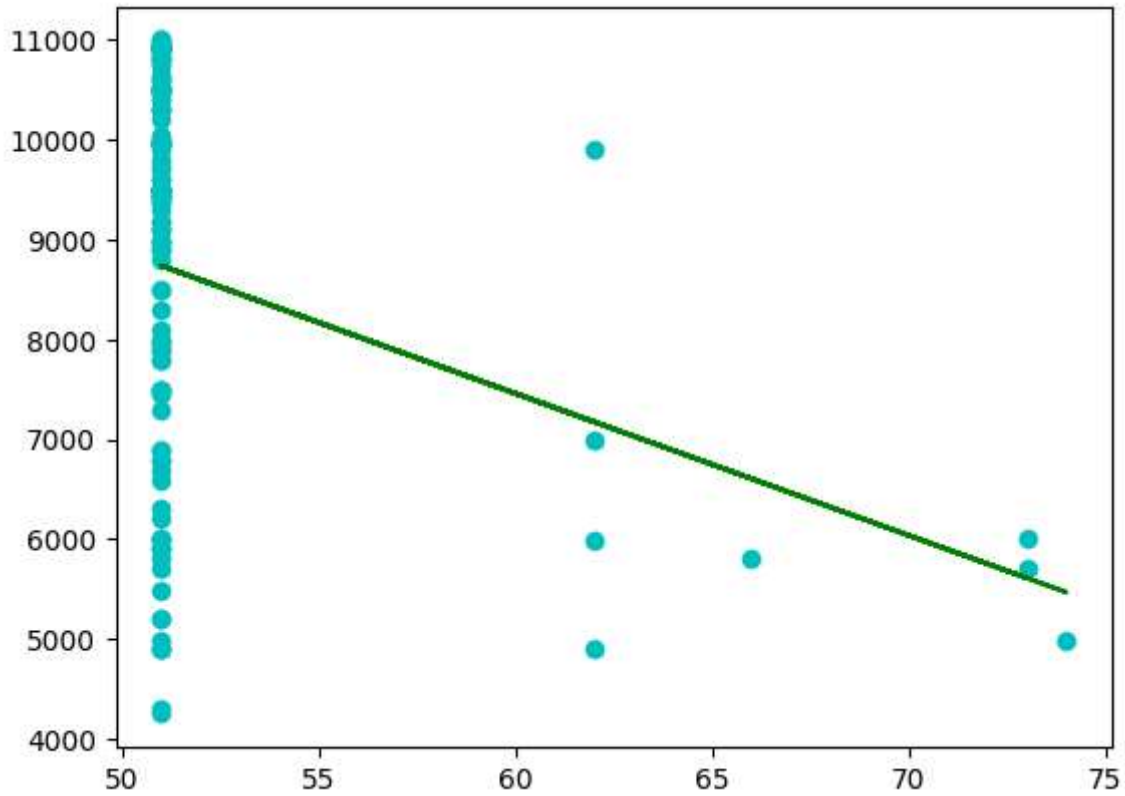
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1538 entries, 0 to 1537
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   power   1538 non-null   int64
 1   price   1538 non-null   int64
dtypes: int64(2)
memory usage: 24.2 KB
```

In [24]:
```
#step 5:Training our model
x=np.array(df["power"]).reshape(-1,1)
y=np.array(df["price"]).reshape(-1,1)
```

In [26]:
```
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 :",regr.score(x_test,y_test))
```

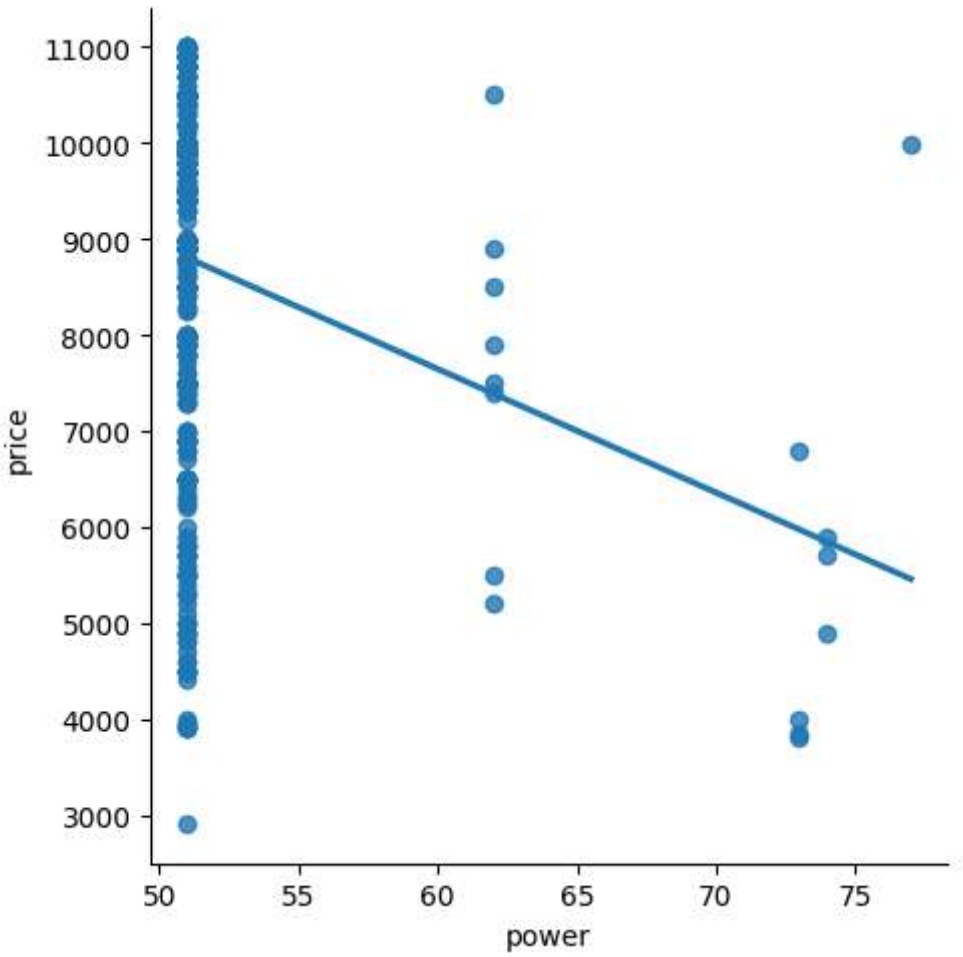
```
regr : 0.06014392334120533
```

In [39]:
```
#step 6:Exploring our results
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color="c")
plt.plot(x_test,y_pred,color="g")
plt.show()
```



In [33]:
```
#step 7:Working with smaller data
df500=df[:][1000:1500]
sns.lmplot(x="power",y="price",data=df500,order=1,ci=None)
```

Out[33]: <seaborn.axisgrid.FacetGrid at 0x2b3de0cfe50>



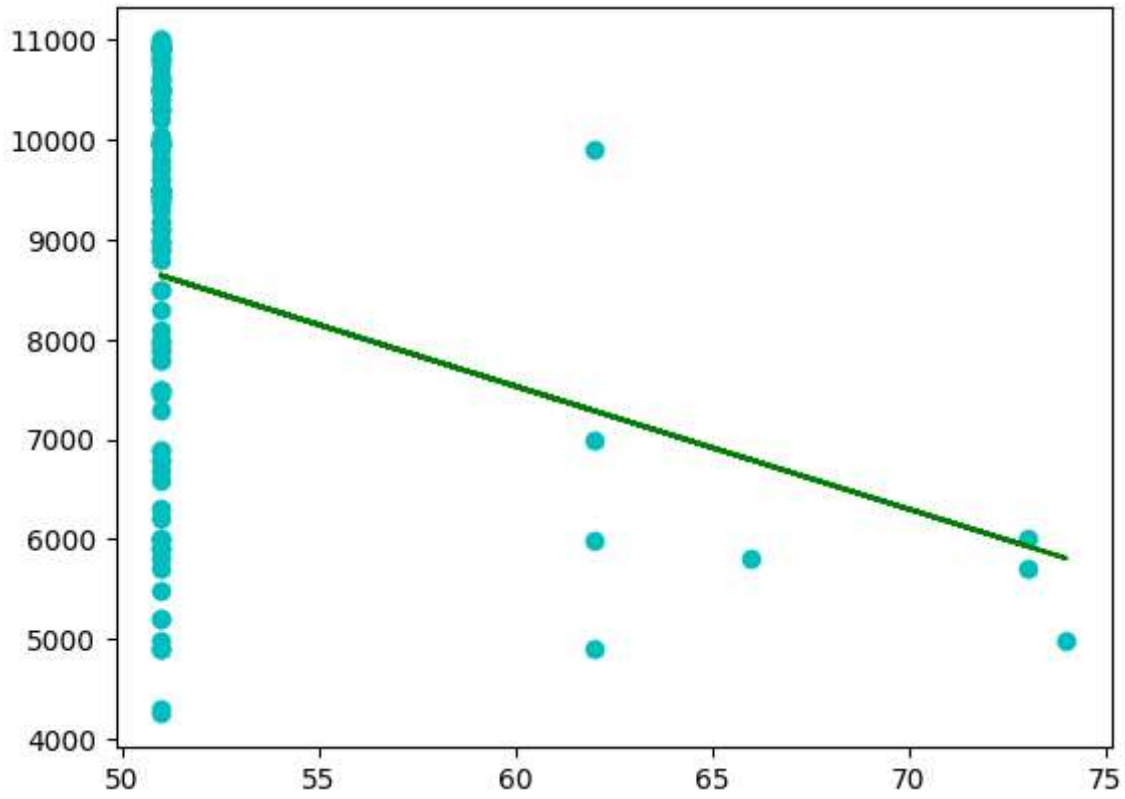
In [31]:
```
x=np.array(df500["power"]).reshape(-1,1)
y=np.array(df500["price"]).reshape(-1,1)
```

In [32]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
```

In [41]:
```
regr=LinearRegression()
regr.fit(x_train,y_train)
print("Regression :",regr.score(x_test,y_test))
```

```
Regression : 0.09561863165183249
```

In [42]:
```
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color="c")
plt.plot(x_test,y_pred,color="g")
plt.show()
```



In [43]:
```
from sklearn.metrics import r2_score
#Train the model
model=LinearRegression()
model.fit(x_train,y_train)
#Evaluate the model on the test data
y_pred=model.predict(x_test)
r2=r2_score(y_test,y_pred)
print("R2 score :",r2)
```

```
R2 score : 0.09561863165183249
```

In [ ]:
```
conclusion:-

The regression score is very less.The model is very poor.Even though we have t
Therefore, Linear Rgression is not accurate for the engine power and the price

Rather than we can conclude that if engine power increases the price decreases
```