

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]:

```
dt=pd.read_csv(r"C:\Users\Arshad Shaik\Downloads\fiat500_VehicleSelection_Dataset (1).cs
dt
```

Out[2]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	
0	1	lounge	51	882	25000	1	44.907242	8.611
1	2	pop	51	1186	32500	1	45.666359	12.241
2	3	sport	74	4658	142228	1	45.503300	11.417
3	4	lounge	51	2739	160000	1	40.633171	17.634
4	5	pop	73	3074	106880	1	41.903221	12.495
...
1533	1534	sport	51	3712	115280	1	45.069679	7.704
1534	1535	lounge	74	3835	112000	1	45.845692	8.666
1535	1536	pop	51	2223	60457	1	45.481541	9.413
1536	1537	lounge	51	2557	80750	1	45.000702	7.682
1537	1538	pop	51	1766	54276	1	40.323410	17.568

1538 rows × 9 columns



In [4]:

```
dt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1538 entries, 0 to 1537
Data columns (total 9 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   ID              1538 non-null   int64
1   model           1538 non-null   object
2   engine_power    1538 non-null   int64
3   age_in_days     1538 non-null   int64
4   km              1538 non-null   int64
5   previous_owners 1538 non-null   int64
6   lat             1538 non-null   float64
7   lon             1538 non-null   float64
8   price          1538 non-null   int64
dtypes: float64(2), int64(6), object(1)
memory usage: 108.3+ KB
```

In [5]:

```
dt.head(7)
```

Out[5]:

	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
5	6	pop	74	3623	70225	1	45.000702	7.682270
6	7	lounge	51	731	11600	1	44.907242	8.611560

In [6]:

```
dt.tail()
```

Out[6]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	lon
1533	1534	sport	51	3712	115280	1	45.069679	7.704
1534	1535	lounge	74	3835	112000	1	45.845692	8.666
1535	1536	pop	51	2223	60457	1	45.481541	9.413
1536	1537	lounge	51	2557	80750	1	45.000702	7.682
1537	1538	pop	51	1766	54276	1	40.323410	17.568

In [7]:

```
dt.describe()
```

Out[7]:

	ID	engine_power	age_in_days	km	previous_owners	l
count	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000	1538.000000
mean	769.500000	51.904421	1650.980494	53396.011704	1.123537	43.54136
std	444.126671	3.988023	1289.522278	40046.830723	0.416423	2.13357
min	1.000000	51.000000	366.000000	1232.000000	1.000000	36.85580
25%	385.250000	51.000000	670.000000	20006.250000	1.000000	41.80290
50%	769.500000	51.000000	1035.000000	39031.000000	1.000000	44.39400
75%	1153.750000	51.000000	2616.000000	79667.750000	1.000000	45.46790
max	1538.000000	77.000000	4658.000000	235000.000000	4.000000	46.79560

In [8]:

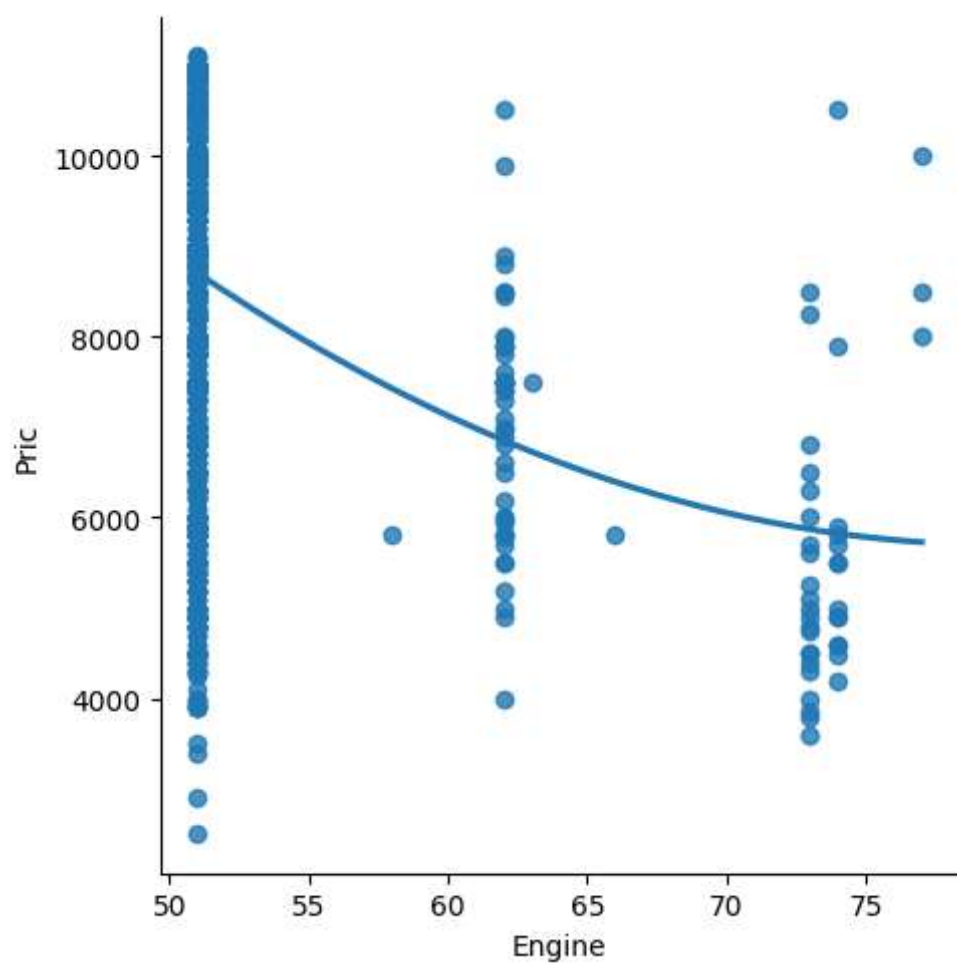
```
dt=dt[['engine_power','price']]
dt.columns=['Engine','Pric']
```

In [9]:

```
sns.lmplot(x='Engine',y='Pric',data=dt,order=2,ci=None)
```

Out[9]:

<seaborn.axisgrid.FacetGrid at 0x2263bed3d90>



In [10]:

```
dt.fillna(method='ffill')
```

Out[10]:

	Engine	Pric
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 [11]:

```
x=np.array(dt['Engine']).reshape(-1,1)
y=np.array(dt['Pric']).reshape(-1,1)
```

In [12]:

```
dt.dropna(inplace=True)
```

C:\Users\Arshad Shaik\AppData\Local\Temp\ipykernel_12228\735218168.py:1:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
dt.dropna(inplace=True)
```

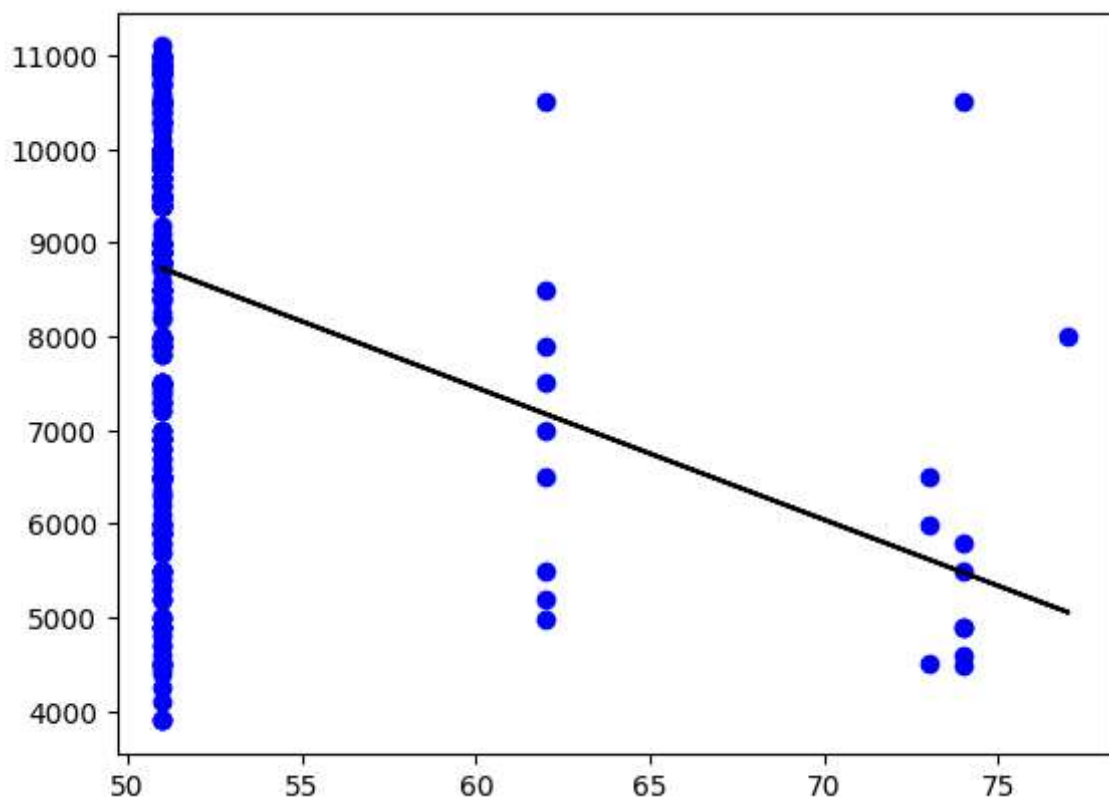
In [13]:

```
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))
```

0.05690227418931182

In [14]:

```
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```

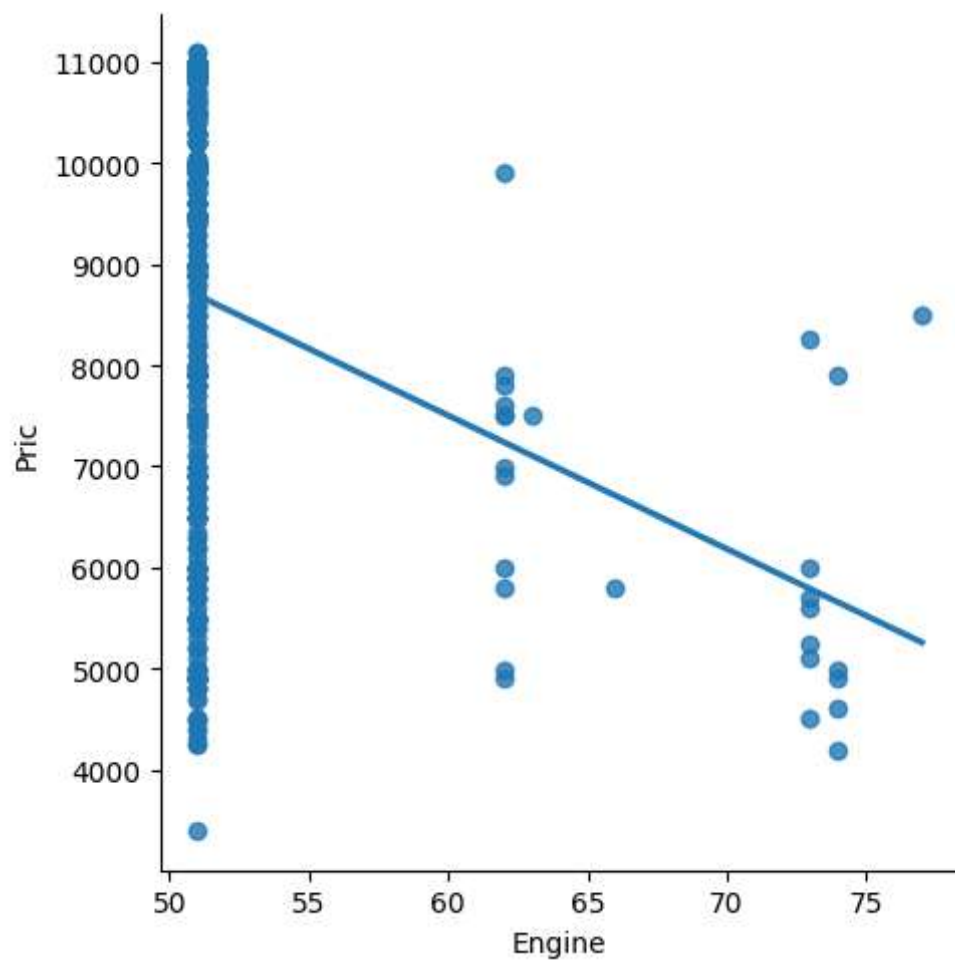


In [15]:

```
dt500=dt[:, :500]  
sns.lmplot(x="Engine", y="Pric", data=dt500, order=1, ci=None)
```

Out[15]:

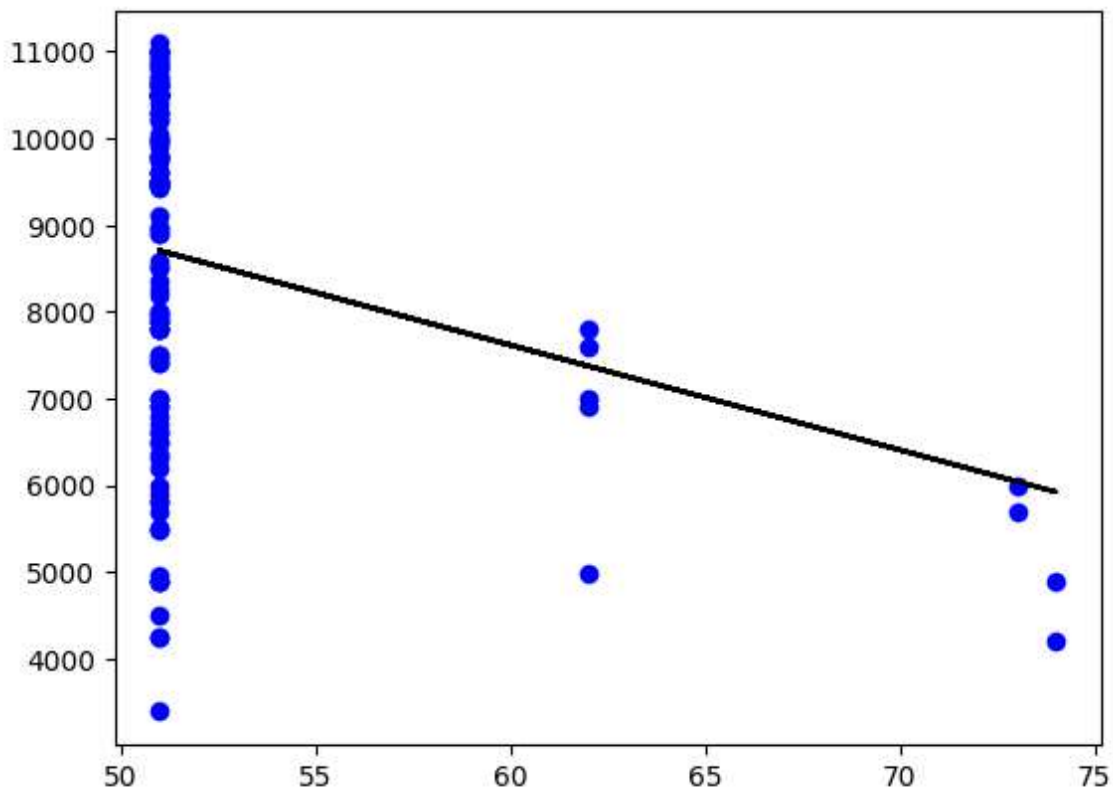
<seaborn.axisgrid.FacetGrid at 0x2263be4a1d0>



In [16]:

```
dt500.fillna(method='ffill',inplace=True)
x=np.array(dt500['Engine']).reshape(-1,1)
y=np.array(dt500['Pric']).reshape(-1,1)
dt500.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='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```

Regression: 0.10787318156159087



In [17]:

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
```

In []: