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 [3]:
```
dt=pd.read_csv(r"C:\Users\DELL E5490\Downloads\fiat500_VehicleSelection_Datas
dt
```

Out[3]:

	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 [4]:
```
dt=dt[['engine_power','price']]
dt.columns=['Engine','Price']
```

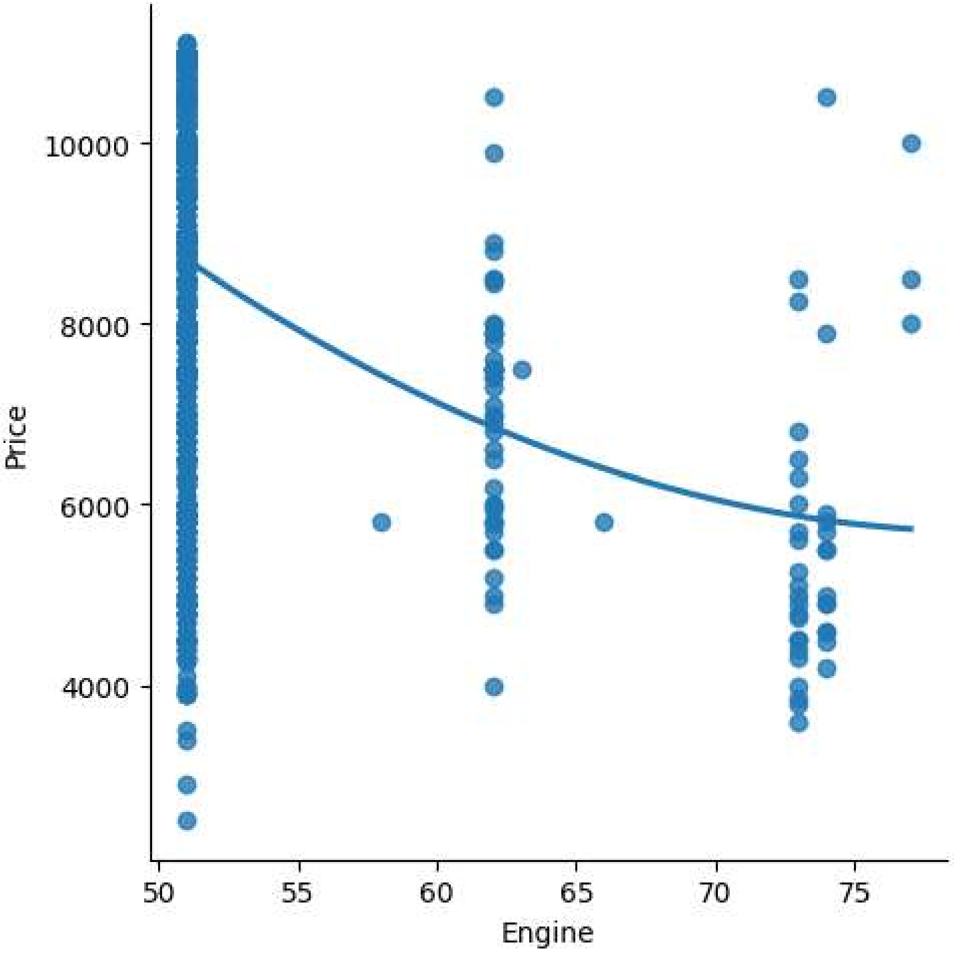
In [5]:
```
dt.head(10)
```

Out[5]:

	Engine	Price
0	51	8900
1	51	8800
2	74	4200
3	51	6000
4	73	5700
5	74	7900
6	51	10750
7	51	9190
8	73	5600
9	51	6000

In [7]:
```
sns.lmplot(x='Engine',y='Price',data=dt,order=2,ci=None)
```

Out[7]:
```
<seaborn.axisgrid.FacetGrid at 0x2a35d8a81c0>
```



In [9]:
```
dt.info()
```

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

In [10]:
```
dt.describe()
```

Out[10]:

	Engine	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 [11]:
```
dt.fillna(method='ffill')
```

Out[11]:

	Engine	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 [13]:

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

In [31]:

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

```
C:\Users\DELL E5490\AppData\Local\Temp\ipykernel_372\3702952713.py:1: Settin
gWithCopyWarning:
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://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  dt.dropna(inplace=True)
```

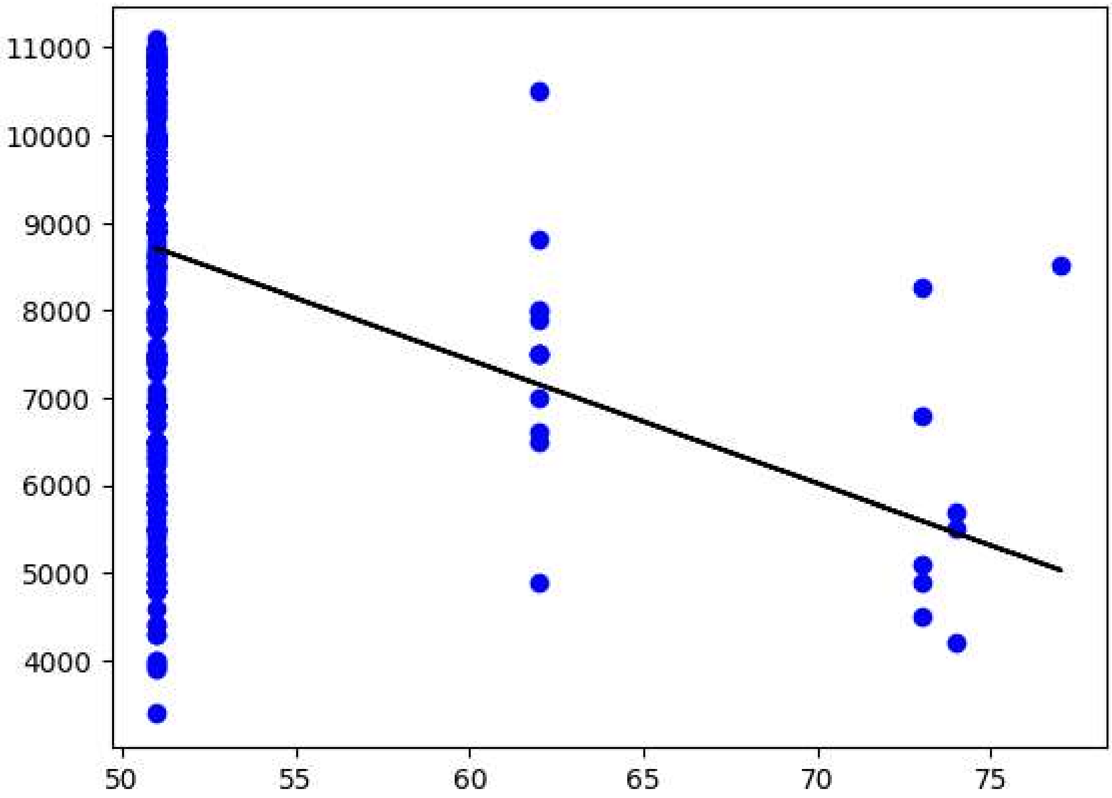
In [15]:

```
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))
```

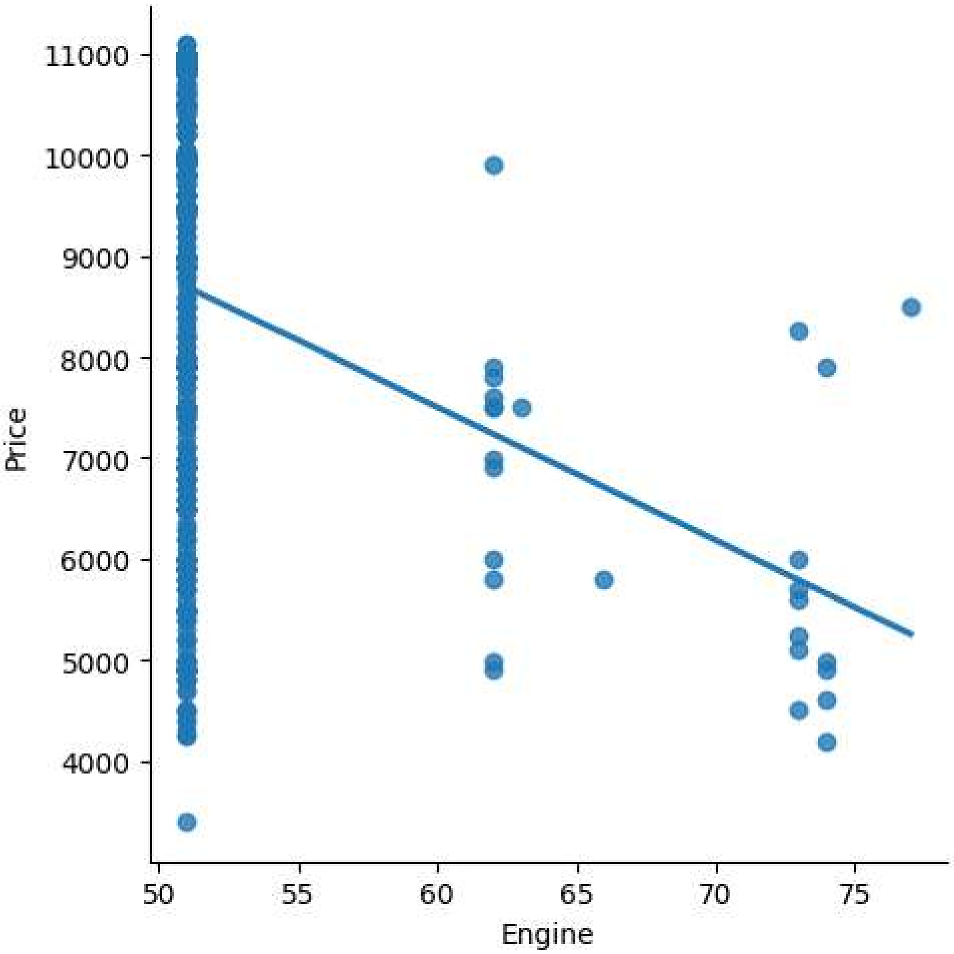
```
0.04821162291901515
```

In [16]:

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



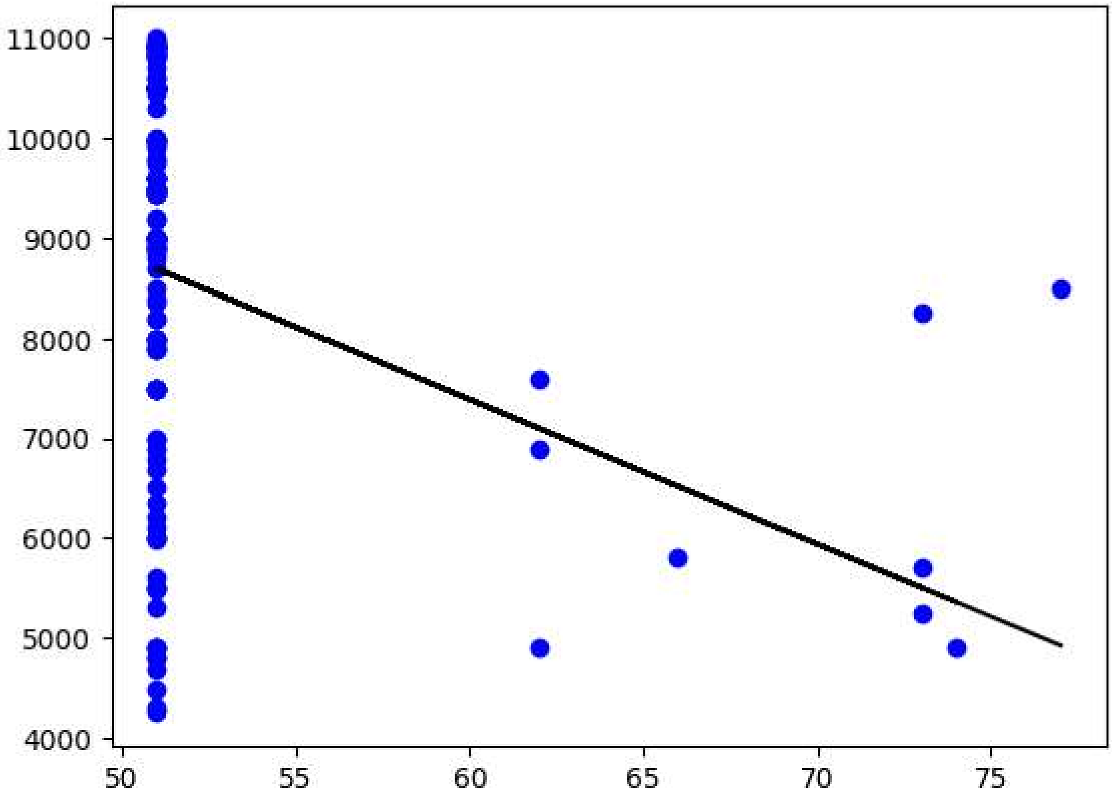
```
In [18]: dt500=dt[:][:500]
         sns.lmplot(x="Engine",y="Price",data=dt500,order=1,ci=None)

Out[18]: <seaborn.axisgrid.FacetGrid at 0x2a35fb95ea0>
```



In [20]:

```
dt500.fillna(method='ffill',inplace=True)
x=np.array(dt500['Engine']).reshape(-1,1)
y=np.array(dt500['Price']).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: 0.0679559435816165
```



In [27]:

```
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)
```

```
82 score: 0.0679559435816165
```