In [23]:

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

In [24]:

```
df=pd.read_csv(r"C:\Users\DELL\Downloads\fiat500_VehicleSelection_Dataset.csv")
df
```

Out[24]:

	ID	model	engine_power	age_in_days	km	previous_owners	lat	l
0	1	lounge	51	882	25000	1	44.907242	8.6115
1	2	pop	51	1186	32500	1	45.666359	12.2418
2	3	sport	74	4658	142228	1	45.503300	11.4178
3	4	lounge	51	2739	160000	1	40.633171	17.6346
4	5	pop	73	3074	106880	1	41.903221	12.4956
...	...	...	...	...	...	...	...	
1533	1534	sport	51	3712	115280	1	45.069679	7.7049
1534	1535	lounge	74	3835	112000	1	45.845692	8.6668
1535	1536	pop	51	2223	60457	1	45.481541	9.4134
1536	1537	lounge	51	2557	80750	1	45.000702	7.6822
1537	1538	pop	51	1766	54276	1	40.323410	17.5682

1538 rows × 9 columns

In [25]:

```
df=df[['engine_power','km']]
df.columns=['Engine','Kilo']
```

In [26]:

```
df.head(10)
```

Out[26]:

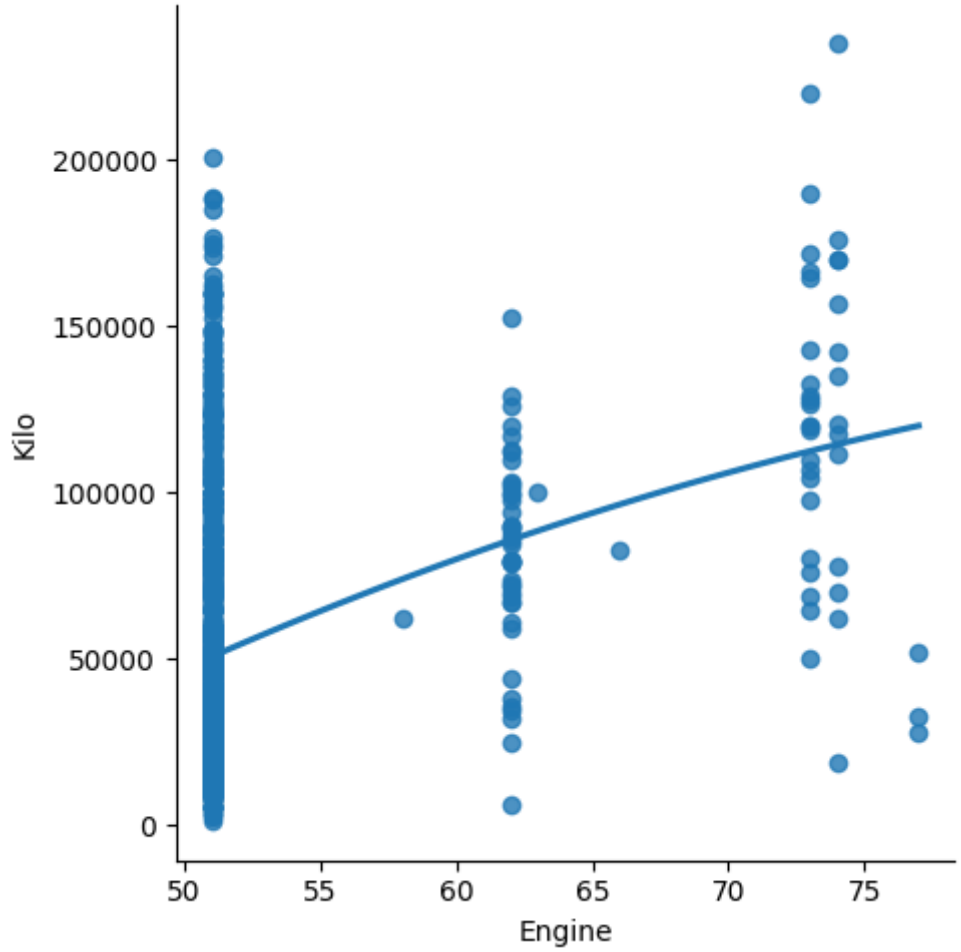
	Engine	Kilo
0	51	25000
1	51	32500
2	74	142228
3	51	160000
4	73	106880
5	74	70225
6	51	11600
7	51	49076
8	73	76000
9	51	89000

In [27]:

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

Out[27]:

```
<seaborn.axisgrid.FacetGrid at 0x1e945bb9e70>
```



In [28]:

```
df.describe()
```

Out[28]:

	Engine	Kilo
count	1538.000000	1538.000000
mean	51.904421	53396.011704
std	3.988023	40046.830723
min	51.000000	1232.000000
25%	51.000000	20006.250000
50%	51.000000	39031.000000
75%	51.000000	79667.750000
max	77.000000	235000.000000

In [29]:

```
df.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   Kilo    1538 non-null   int64
dtypes: int64(2)
memory usage: 24.2 KB
```

In [51]:

```
df.fillna(method='ffill',inplace=True)
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_5708\4116506308.py:1: SettingWi
thCopyWarning:
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-doc
s/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)
  df.fillna(method='ffill',inplace=True)
```

In [34]:

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

In [45]:

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

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_5708\1379821321.py:1: SettingWi
thCopyWarning:
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-doc
s/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)
  df.dropna(inplace=True)
```

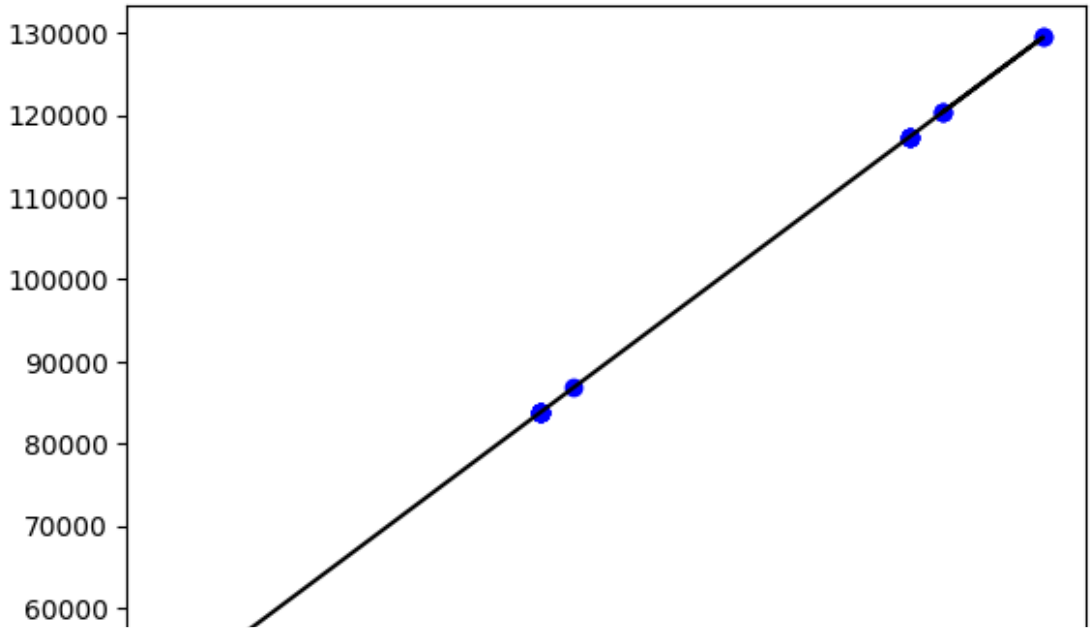
In [46]:

```
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))
```

```
0.054152294106464716
```

In [39]:

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

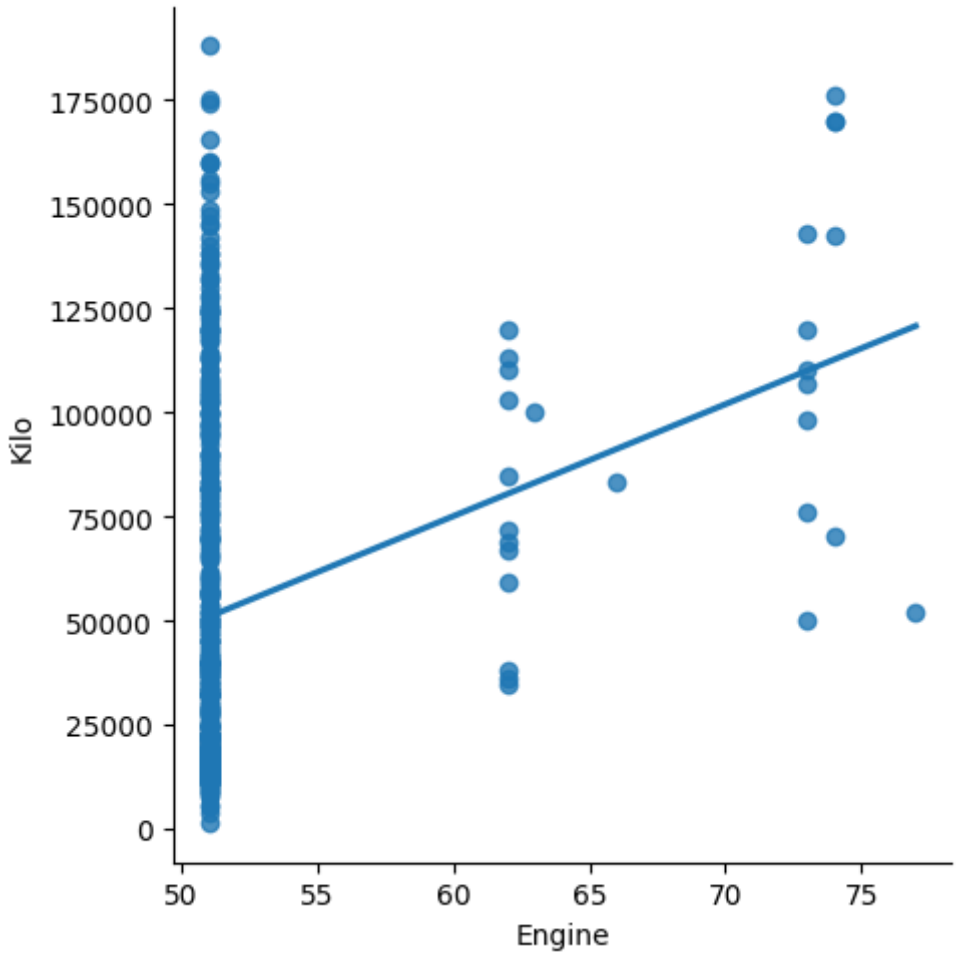


In [41]:

```
df500=df[:][: 500]
sns.lmplot(x="Engine",y="Kilo",data=df500,order=1,ci=None)
```

Out[41]:

```
<seaborn.axisgrid.FacetGrid at 0x1e947631c00>
```



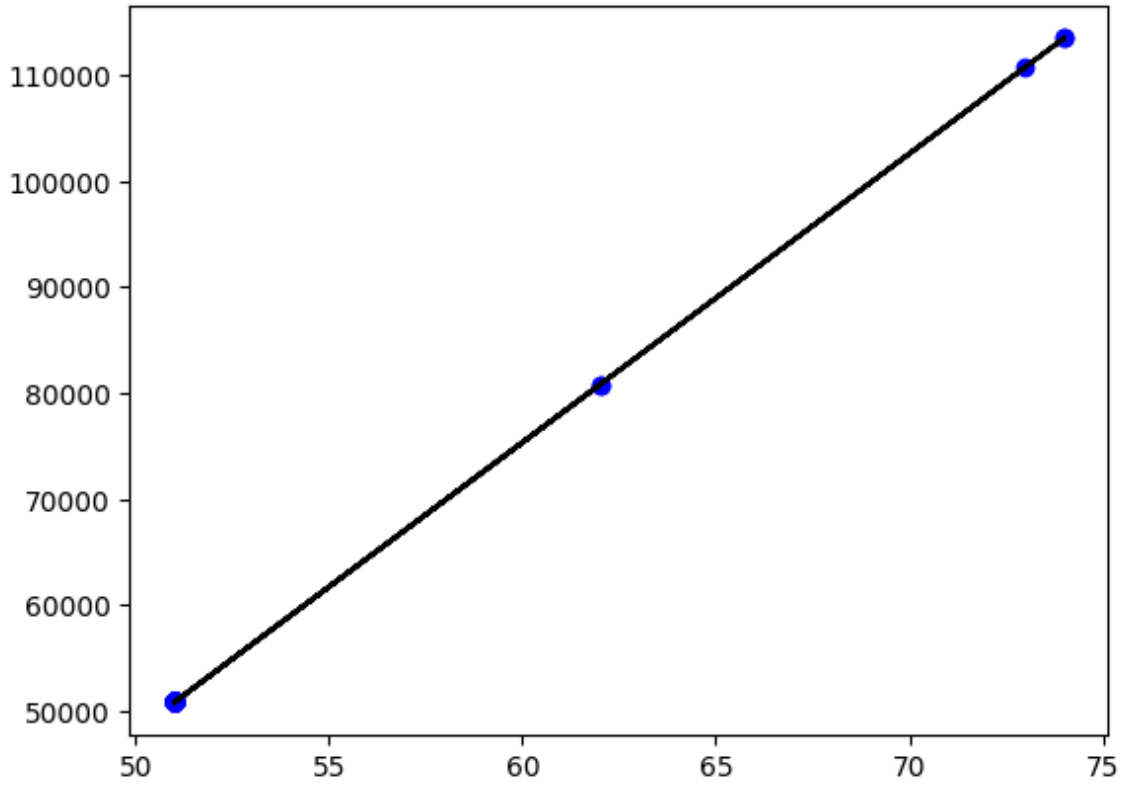
In [44]:

```
df500.fillna(method='ffill',inplace=True)
x=np.array(df500['Engine']).reshape(-1,1)
y=np.array(df500['Kilo']).reshape(-1,1)
df500.dropna(inplace=True)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
```

In [52]:

```
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_pred,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```

```
Regression: 0.054152294106464716
```



In [53]:

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
model=LinearRegression()
model.fit(x_train,y_train)
```

Out[53]:

```
▾ LinearRegression
LinearRegression()
```

In [57]:

```
y_pred=model.predict(x_test)
r2=r2_score(y_test,y_pred)
print("R2 score:",r2)
```

```
R2 score: 0.054152294106464716
```

In [ ]: