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\monim\Downloads\fiat500_VehicleSelection_Dataset.csv")
df
```

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 [3]:

```
df=df[['km','lat']]
df.columns=['KM','LAT']
```

In [4]:

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

Out[4]:

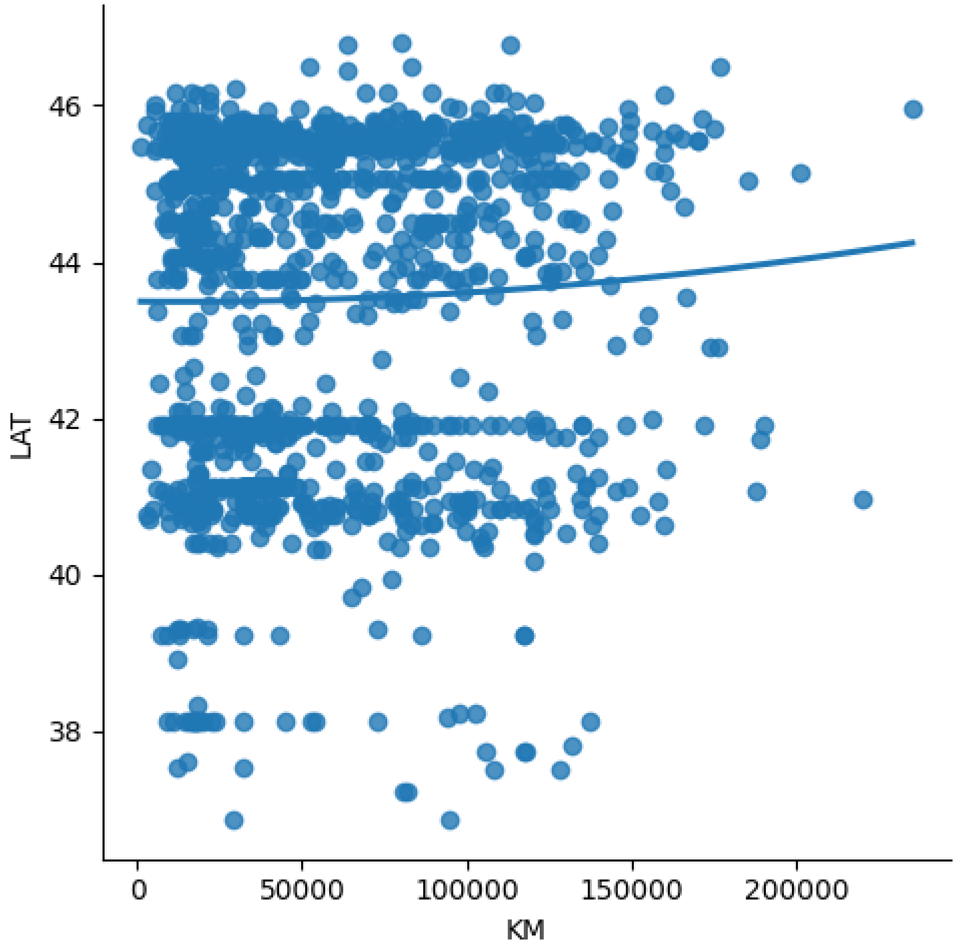
	KM	LAT
0	25000	44.907242
1	32500	45.666359
2	142228	45.503300
3	160000	40.633171
4	106880	41.903221
5	70225	45.000702
6	11600	44.907242
7	49076	41.903221
8	76000	45.548000
9	89000	45.438301

In [5]:

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

Out[5]:

```
<seaborn.axisgrid.FacetGrid at 0x2672a20c940>
```



In [6]:

```
df.describe()
```

Out[6]:

	KM	LAT
count	1538.000000	1538.000000
mean	53396.011704	43.541361
std	40046.830723	2.133518
min	1232.000000	36.855839
25%	20006.250000	41.802990
50%	39031.000000	44.394096
75%	79667.750000	45.467960
max	235000.000000	46.795612

In [7]:

```
df.info()
```

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

In [8]:

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

```
C:\Users\monim\AppData\Local\Temp\ipykernel_14856\4116506308.py:1: Setting
WithCopyWarning:
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 [9]:

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

In [10]:

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

```
C:\Users\monim\AppData\Local\Temp\ipykernel_14856\1379821321.py:1: Setting
WithCopyWarning:
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)
```

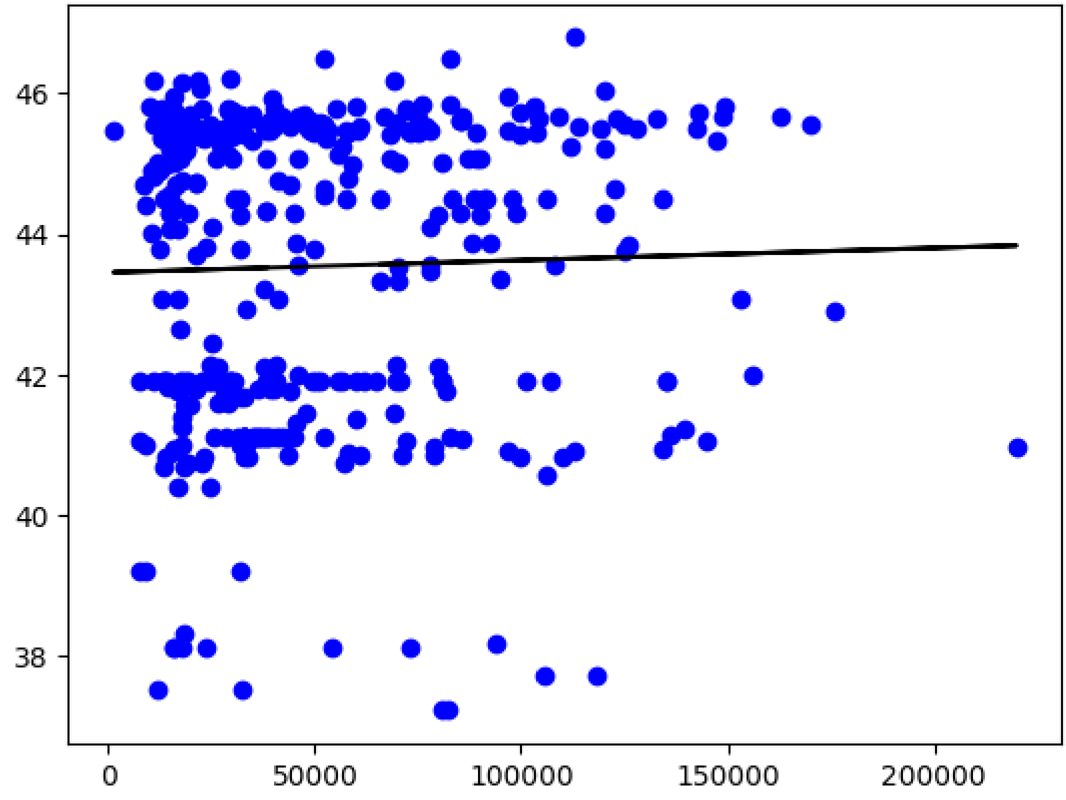
In [11]:

```
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))
```

```
0.0013876695524132066
```

In [12]:

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

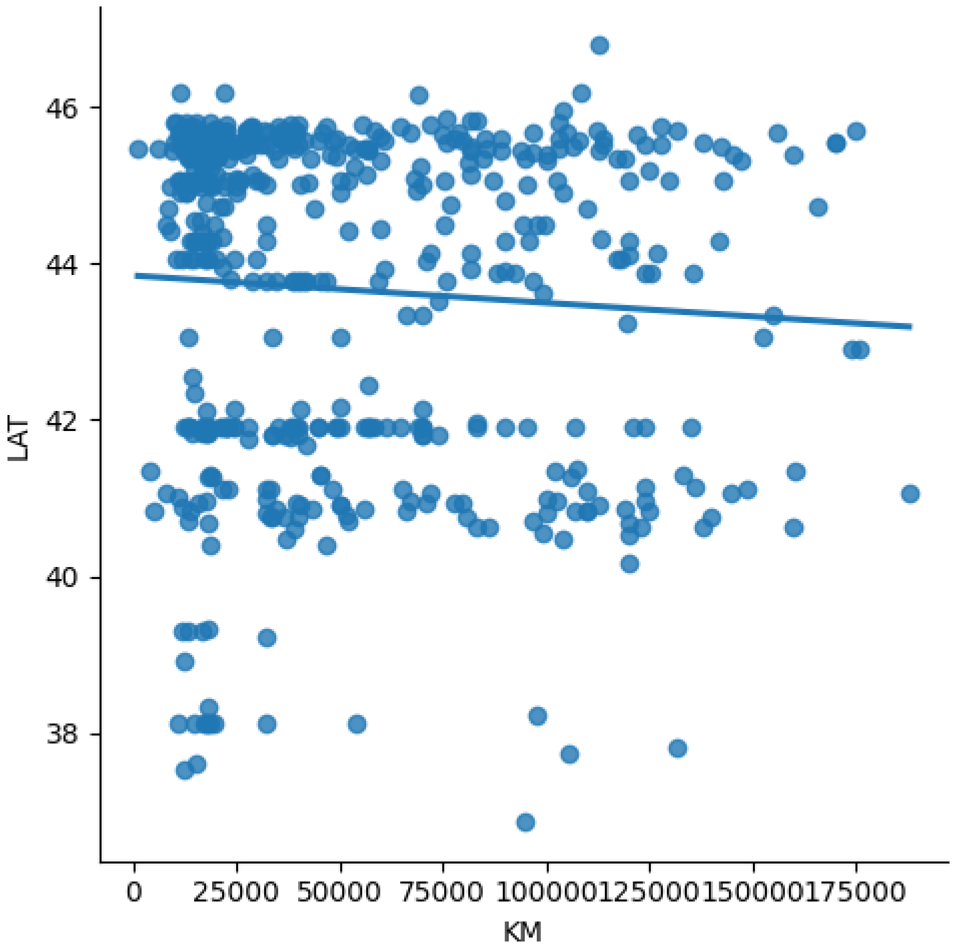


In [13]:

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

Out[13]:

```
<seaborn.axisgrid.FacetGrid at 0x267300e3af0>
```



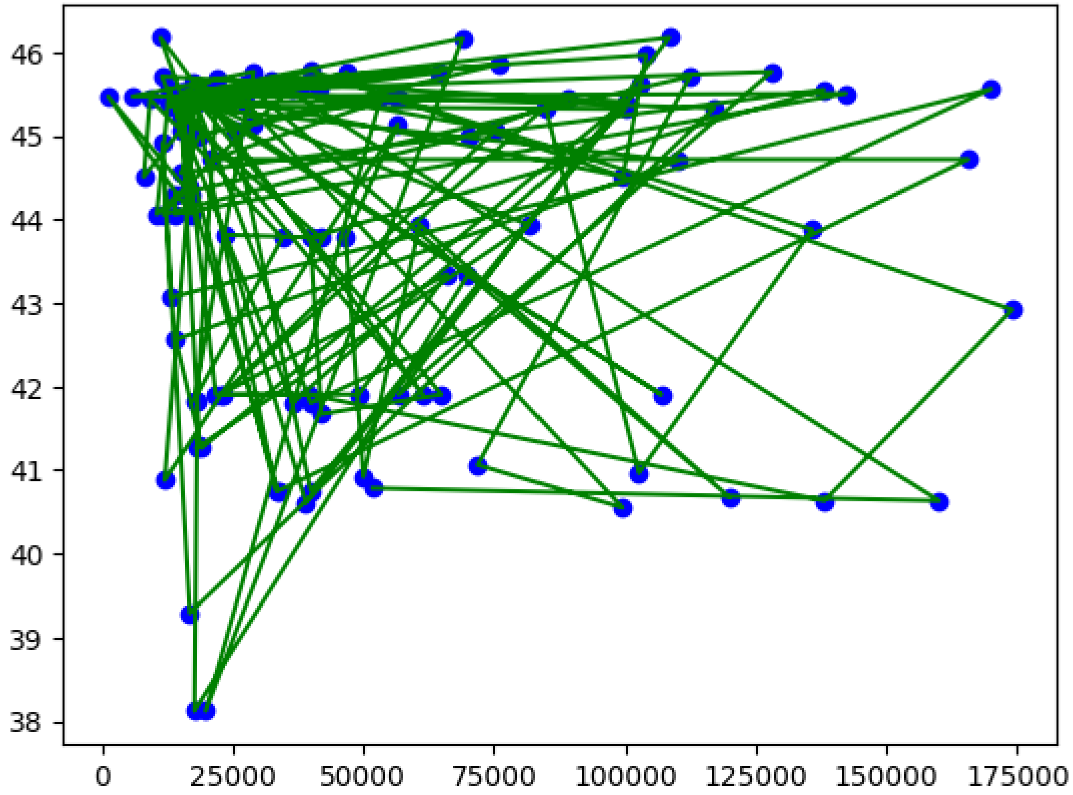
In [16]:

```
df500.fillna(method='ffill',inplace=True)
x=np.array(df500['KM']).reshape(-1,1)
y=np.array(df500['LAT']).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))
```

```
Regression: -0.04923670755888754
```

In [27]:

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



In [30]:

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

Out[30]:

```
LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [31]:

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

```
R2 score: -0.04923670755888754
```

In [ ]: