In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
from sklearn import preprocessing,svm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [3]:
```
df=pd.read_csv(r"C:\Users\DELL E5490\Downloads\fiat500_VehicleSelection_Dataset.csv")
df
```

Out[3]:

| | ID | model | engine_power | age_in_days | km | previous_owners | lat | lon | price |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | lounge | 51 | 882 | 25000 | 1 | 44.907242 | 8.611560 | 8900 |
| **1** | 2 | pop | 51 | 1186 | 32500 | 1 | 45.666359 | 12.241890 | 8800 |
| **2** | 3 | sport | 74 | 4658 | 142228 | 1 | 45.503300 | 11.417840 | 4200 |
| **3** | 4 | lounge | 51 | 2739 | 160000 | 1 | 40.633171 | 17.634609 | 6000 |
| **4** | 5 | pop | 73 | 3074 | 106880 | 1 | 41.903221 | 12.495650 | 5700 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **1533** | 1534 | sport | 51 | 3712 | 115280 | 1 | 45.069679 | 7.704920 | 5200 |
| **1534** | 1535 | lounge | 74 | 3835 | 112000 | 1 | 45.845692 | 8.666870 | 4600 |
| **1535** | 1536 | pop | 51 | 2223 | 60457 | 1 | 45.481541 | 9.413480 | 7500 |
| **1536** | 1537 | lounge | 51 | 2557 | 80750 | 1 | 45.000702 | 7.682270 | 5990 |
| **1537** | 1538 | pop | 51 | 1766 | 54276 | 1 | 40.323410 | 17.568270 | 7900 |

1538 rows × 9 columns

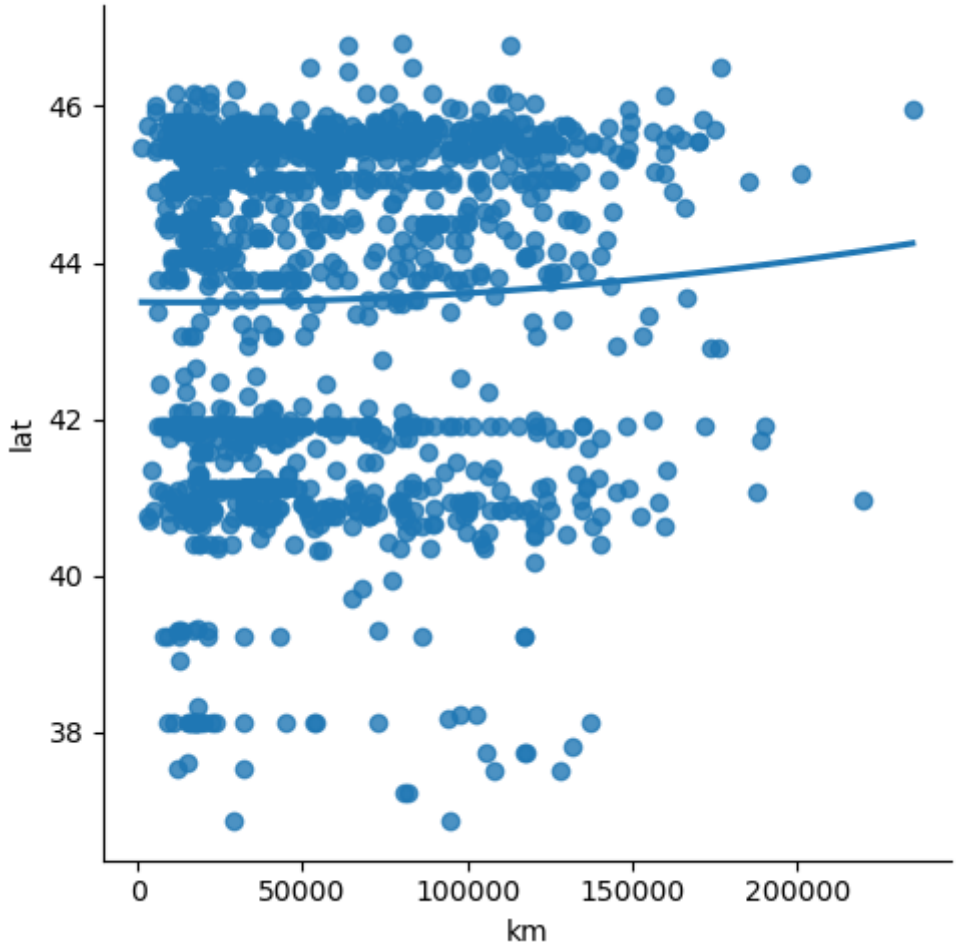
In [4]:
```
df=df[['km','lat']]
df.columns=['km','lat']
df.head(10)
```

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 0x1772a582e30>
```



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\DELL E5490\AppData\Local\Temp\ipykernel_19356\3028625988.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)
  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)
```

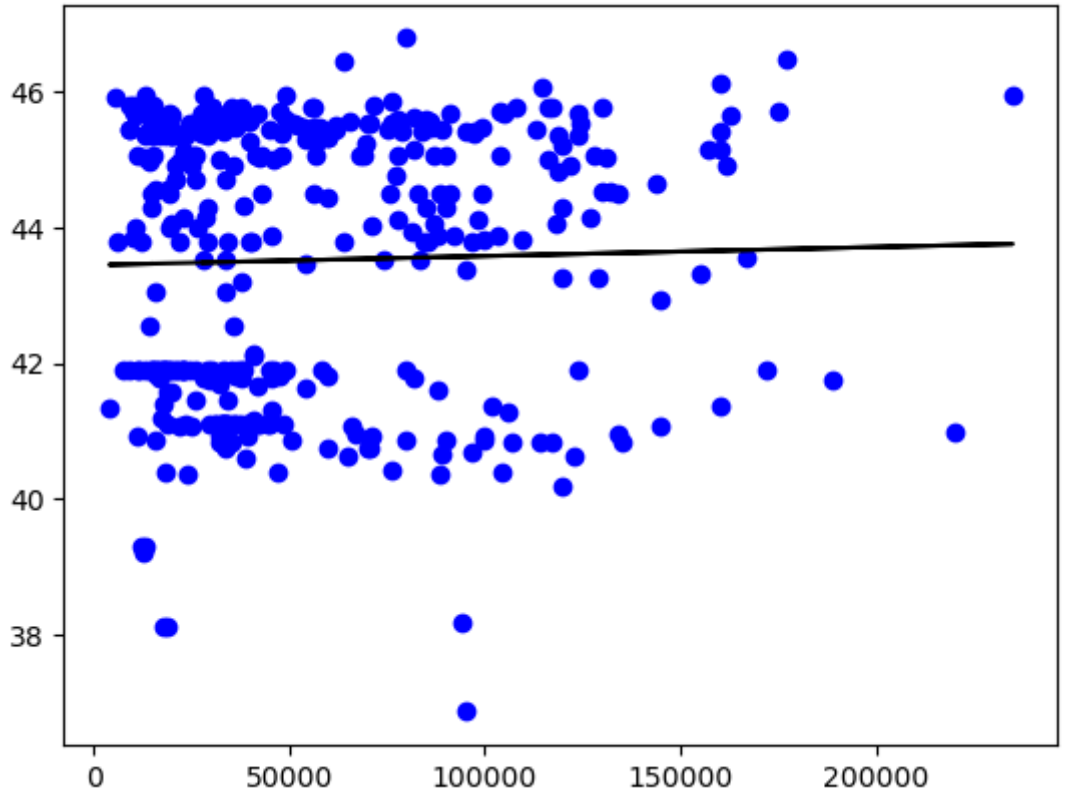
```
C:\Users\DELL E5490\AppData\Local\Temp\ipykernel_19356\1791587065.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

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

In [20]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.5)
```

In [12]:
```
regr=LinearRegression()
regr.fit(x_train,y_train)
print(regr.score(x_test,y_test))
```

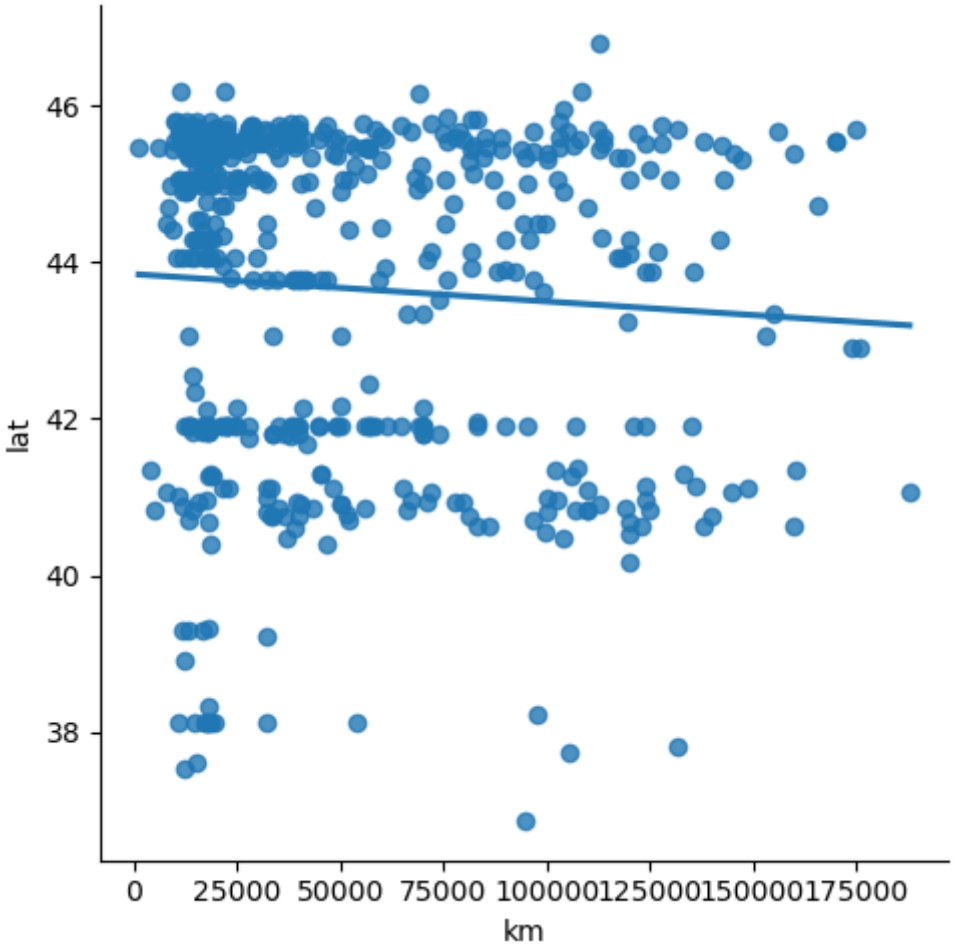
```
0.0008701088012534886
```

In [13]:
```
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 [22]:
```
df500=df[:][:500]
sns.lmplot(x="km",y="lat",data=df500,order=1,ci=None)
```

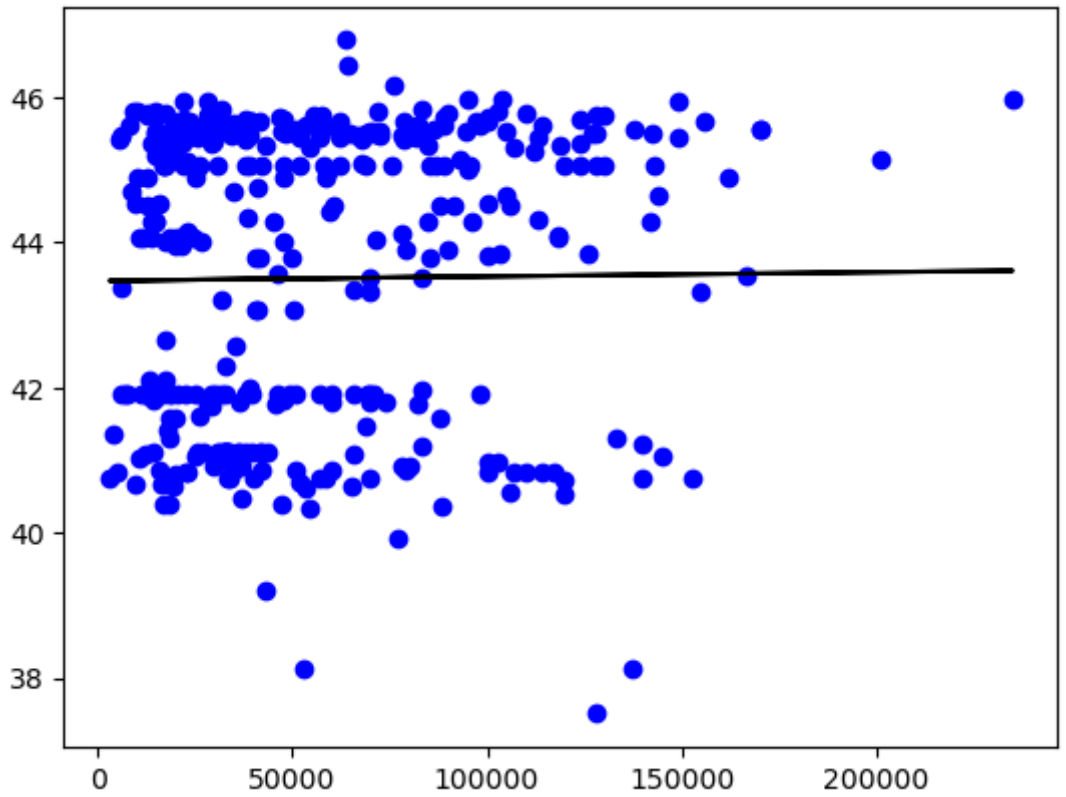
Out[22]:
```
<seaborn.axisgrid.FacetGrid at 0x1773215f3a0>
```



In [23]:

```
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))
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: -0.0026732756969793936
```



In [21]:
```
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)
```

```
R2.score: 0.0014001719274152613
```

In [ ]:
```

```