

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\Teju\Downloads\bottle.csv.zip")  
df
```

C:\Users\Teju\AppData\Local\Temp\ipykernel_19136\2095959205.py:1: DtypeWarning: Columns (47,73) have mixed types. Specify dtype option on import or set low_memory=False.

```
df=pd.read_csv(r"C:\Users\Teju\Downloads\bottle.csv.zip")
```

Out[2]:

Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta
0	1	1	054.0 056.0 19-4903CR-HY-060-0930-05400560-0000A-3	0	10.500	33.4400	NaN	25.64900
1	1	2	054.0 056.0 19-4903CR-HY-060-0930-05400560-0008A-3	8	10.460	33.4400	NaN	25.65600

In [4]:

```
df=df[['Salnty','T_degC']]
df.columns=['Sal','Temp']
```

In [5]:

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

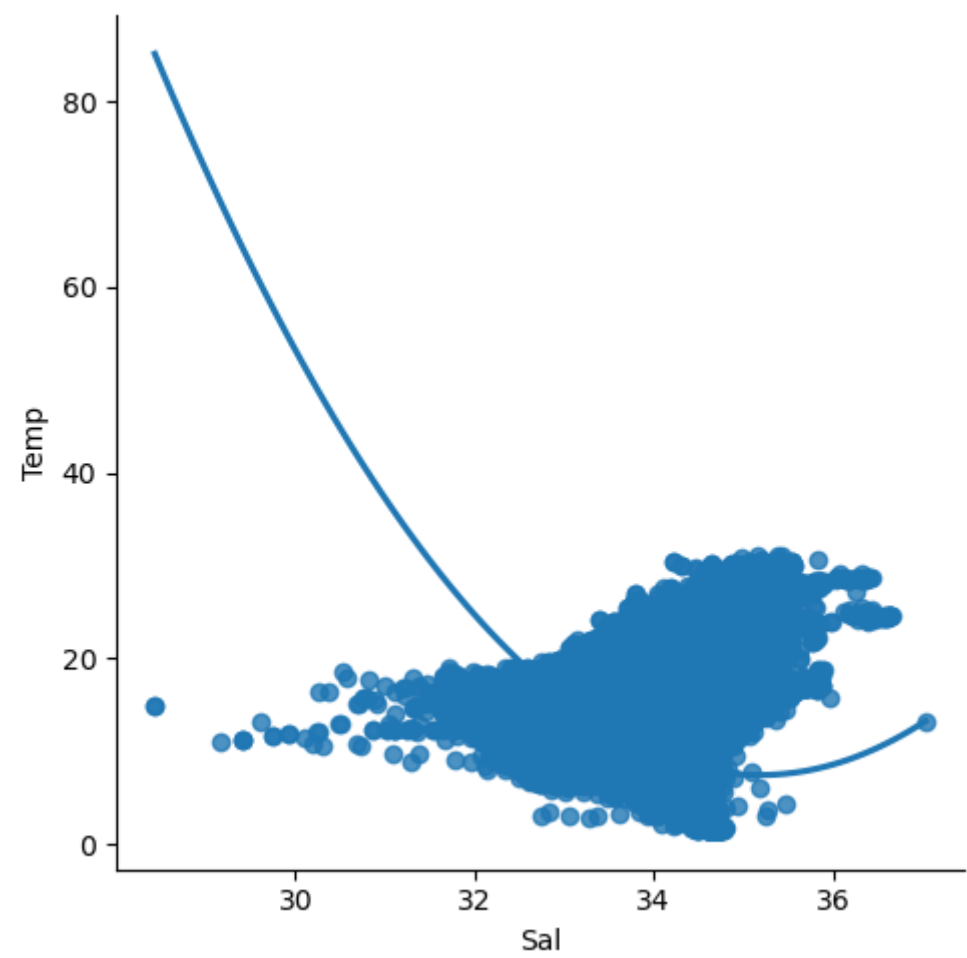
Out[5]:

Sal Temp									
0	33.440	10.50	5	054.0 056.0	19-4903CR-HY-060-0930-05400560-0020A-7	20	10.450	33.4210	NaN 25.64300
1	33.440	10.46							
2	33.437	10.46							
3	33.420	10.45
4	33.421	10.45			20-1611SR-MX-310-2239-09340264-0000A-7	0	18.744	33.4083	5.805 23.87055
5	33.431	10.45	864859	093.4 026.4					
6	33.440	10.45							
7	33.424	10.24			20-1611SR-MX-310-2239-09340264-0002A-3	2	18.744	33.4083	5.805 23.87072
8	33.420	10.06	864859	093.4 026.4					
9	33.494	9.86	864860						
					20-1611SR-MX-310-2239-09340264-0005A-3	5	18.692	33.4150	5.796 23.88911
					20-1611SR-MX-310-2239-09340264-0010A-3	10	18.161	33.4062	5.816 24.01426

In [6]: Cst_Cnt Btl_Cnt Sta_ID Depth_ID Depthm T_degC Salnty O2ml_L STheta

sns.lmplot(x="Sal",y="Temp",data=df,order=2,ci=None)

Out[6]: 34404 864863 093.4 MX-310- 15 17.533 33.3880 5.774 24.15297
<seaborn.axisgrid.FacetGrid at 0x122f3f8f6da0>



In [7]:

df.describe()

Out[7]:

	Sal	Temp
count	817509.000000	853900.000000
mean	33.840350	10.799677
std	0.461843	4.243825
min	28.431000	1.440000
25%	33.488000	7.680000
50%	33.863000	10.060000
75%	34.196900	13.880000
max	37.034000	31.140000

In [8]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 864863 entries, 0 to 864862
Data columns (total 2 columns):
#   Column  Non-Null Count  Dtype  
---  -
0    Sal      817509 non-null  float64
1    Temp      853900 non-null  float64
dtypes: float64(2)
memory usage: 13.2 MB
```

In [9]:

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

C:\Users\Teju\AppData\Local\Temp\ipykernel_19136\4116506308.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 [10]:

```
X=np.array(df['Sal']).reshape(-1,1)
y=np.array(df['Temp']).reshape(-1,1)
```

In [11]:

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

C:\Users\Teju\AppData\Local\Temp\ipykernel_19136\1379821321.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)
```

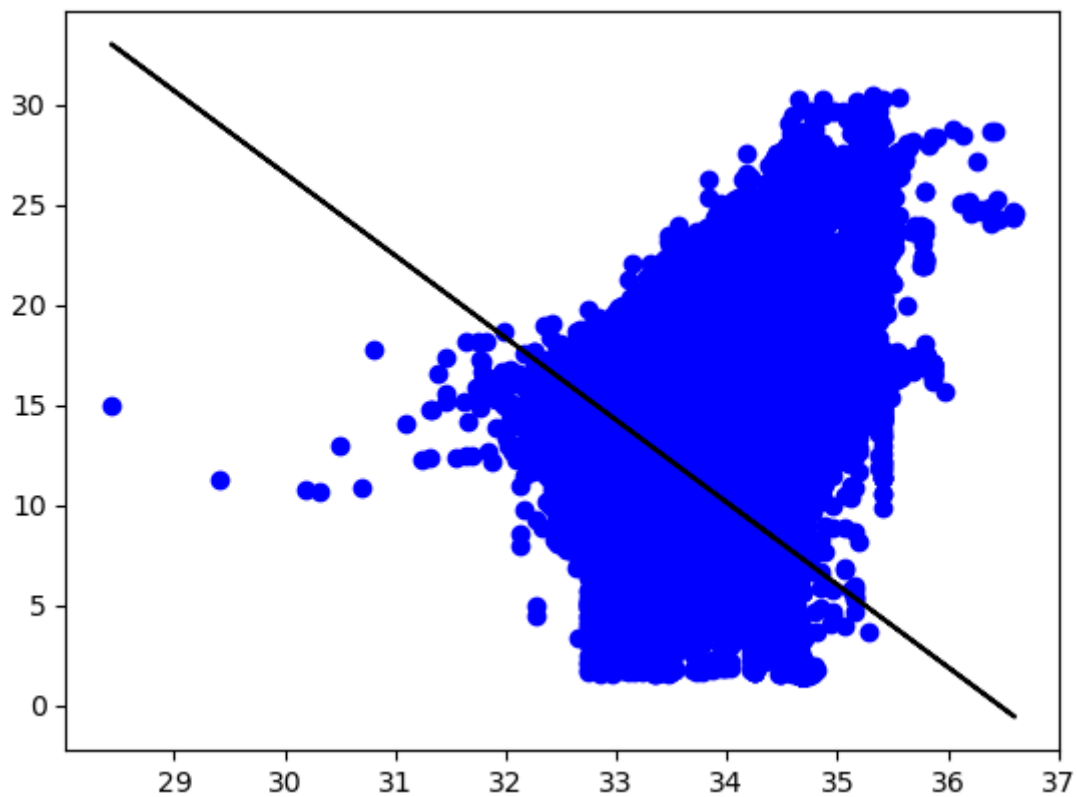
In [12]:

```
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))
```

0.20101936617992366

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()
```

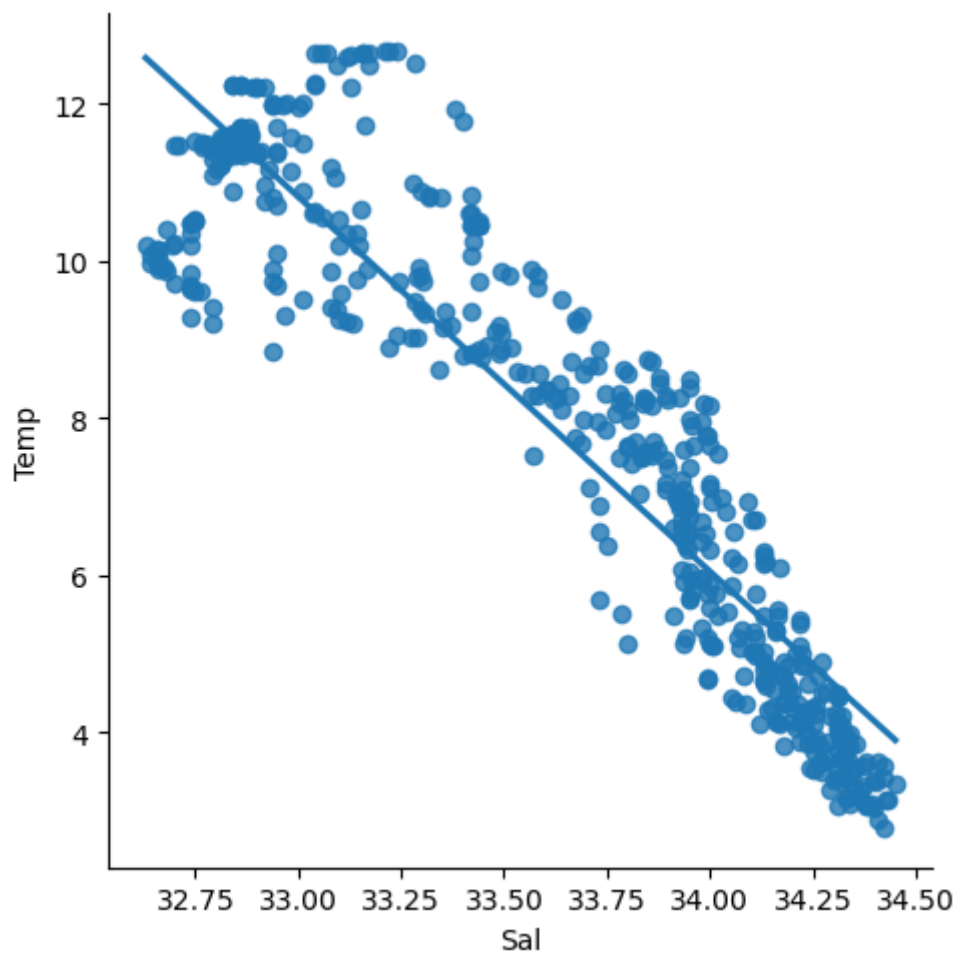


In [14]:

```
df500=df[:][:500]  
sns.lmplot(x='Sal',y='Temp',data=df500,order=1,ci=None)
```

Out[14]:

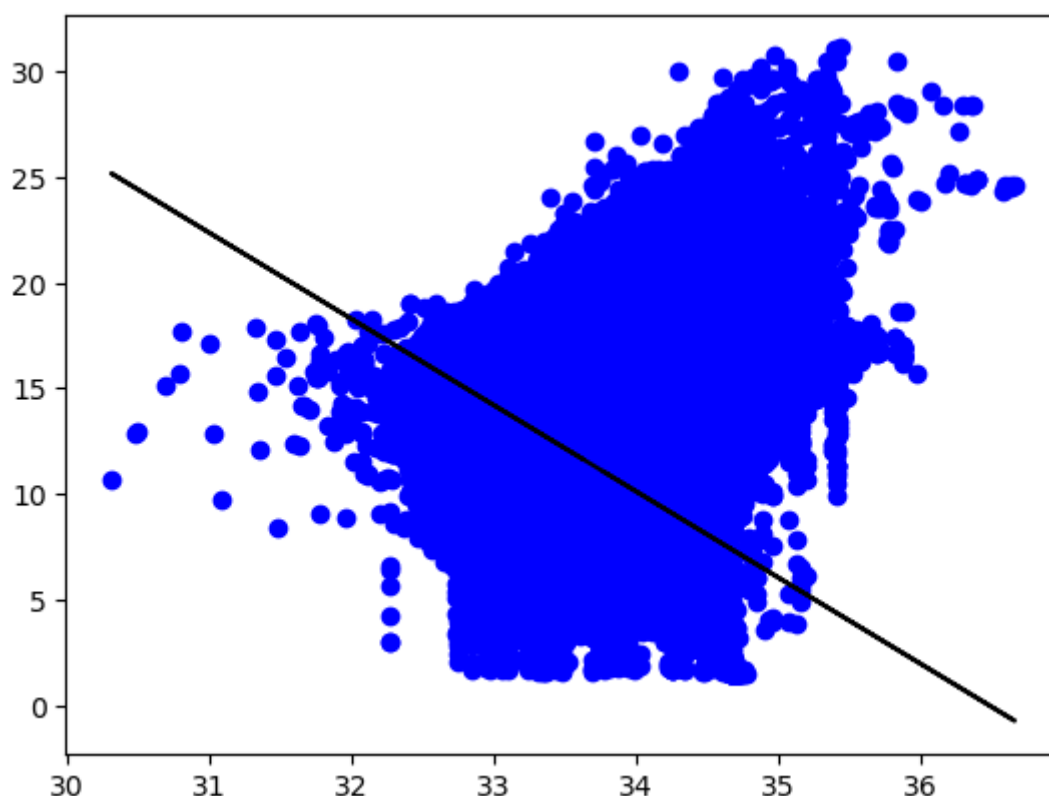
<seaborn.axisgrid.FacetGrid at 0x22f7832c520>



In [15]:

```
df500.fillna(method='ffill',inplace=True)
X=np.array(df['Sal']).reshape(-1,1)
y=np.array(df['Temp']).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))
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: 0.20932403448047887



In [16]:

```
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: 0.20932403448047887

In [17]:

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

Out[17]:

	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 [18]:

```
df=df[['age_in_days','lat']]
df.columns=['age','la']
```

In [19]:

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

Out[19]:

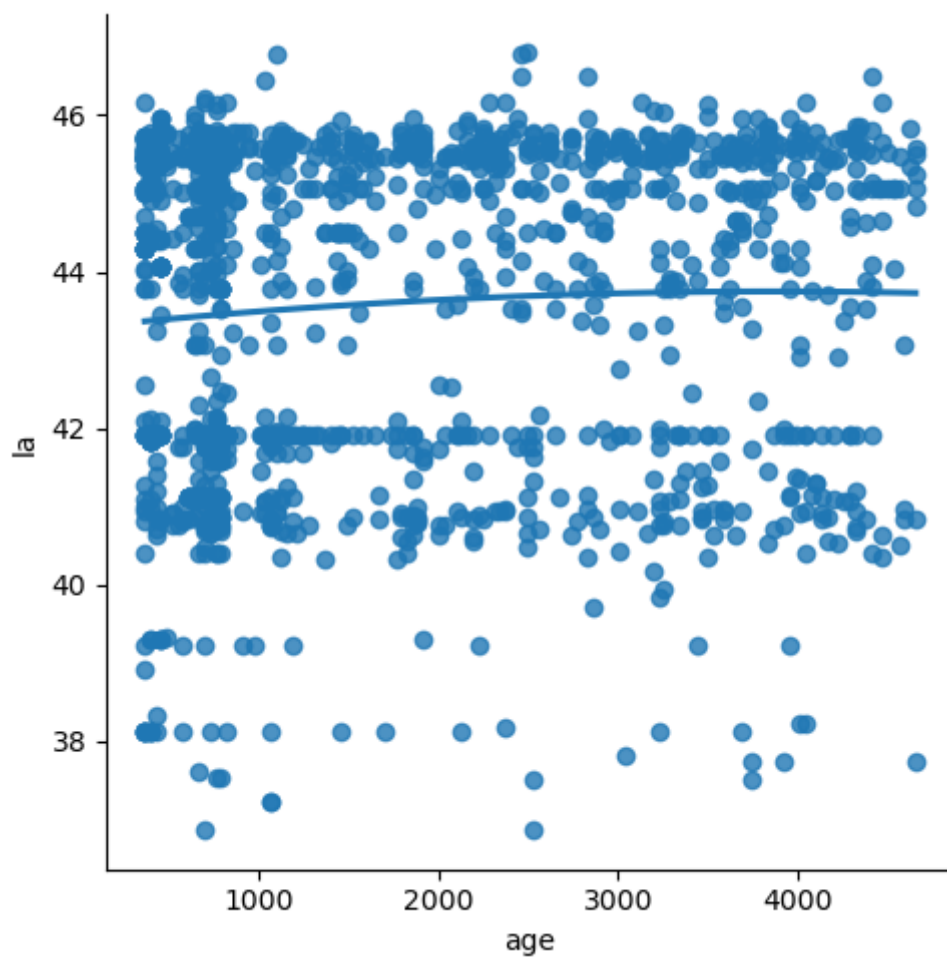
	age	la
0	882	44.907242
1	1186	45.666359
2	4658	45.503300
3	2739	40.633171
4	3074	41.903221
5	3623	45.000702
6	731	44.907242
7	1521	41.903221
8	4049	45.548000
9	3653	45.438301

In [20]:

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

Out[20]:

<seaborn.axisgrid.FacetGrid at 0x22f81f037c0>



In [21]:

```
df.describe()
```

Out[21]:

	age	la
count	1538.000000	1538.000000
mean	1650.980494	43.541361
std	1289.522278	2.133518
min	366.000000	36.855839
25%	670.000000	41.802990
50%	1035.000000	44.394096
75%	2616.000000	45.467960
max	4658.000000	46.795612

In [22]:

```
df.info()
```

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

In [23]:

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

C:\Users\Teju\AppData\Local\Temp\ipykernel_19136\4116506308.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 [24]:

```
X=np.array(df['age']).reshape(-1,1)
y=np.array(df['la']).reshape(-1,1)
```

In [25]:

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

C:\Users\Teju\AppData\Local\Temp\ipykernel_19136\658506679.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)
```

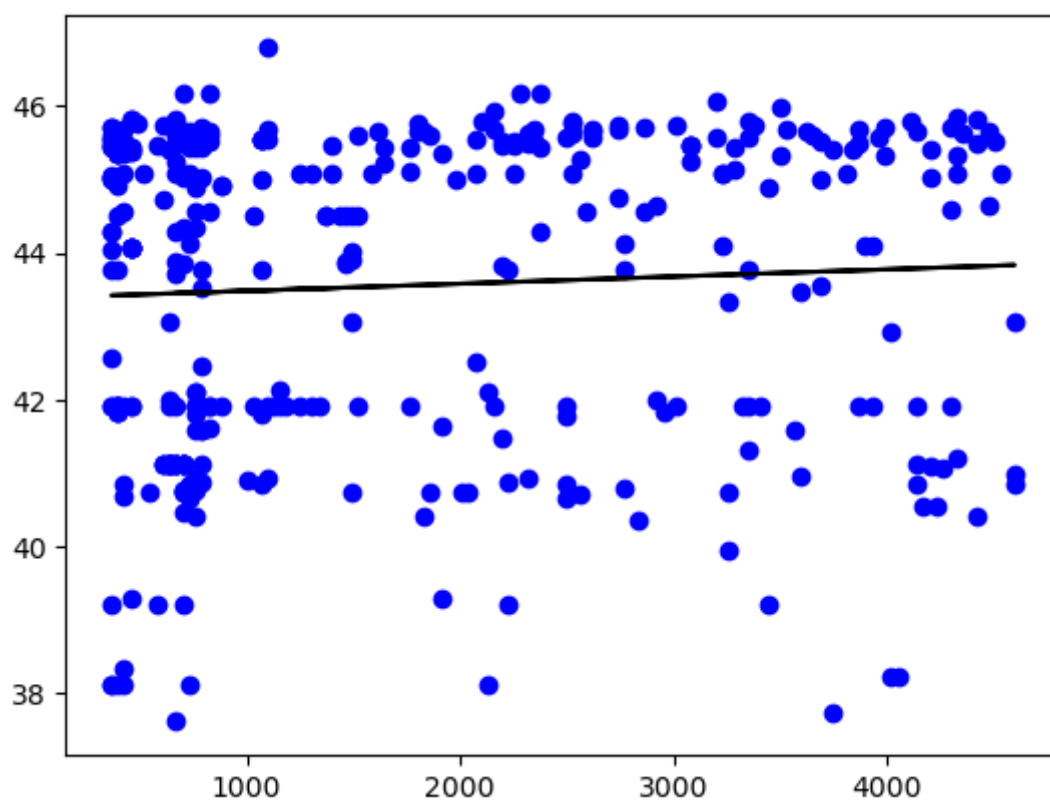
In [26]:

```
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))
```

0.0051381807460785245

In [27]:

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

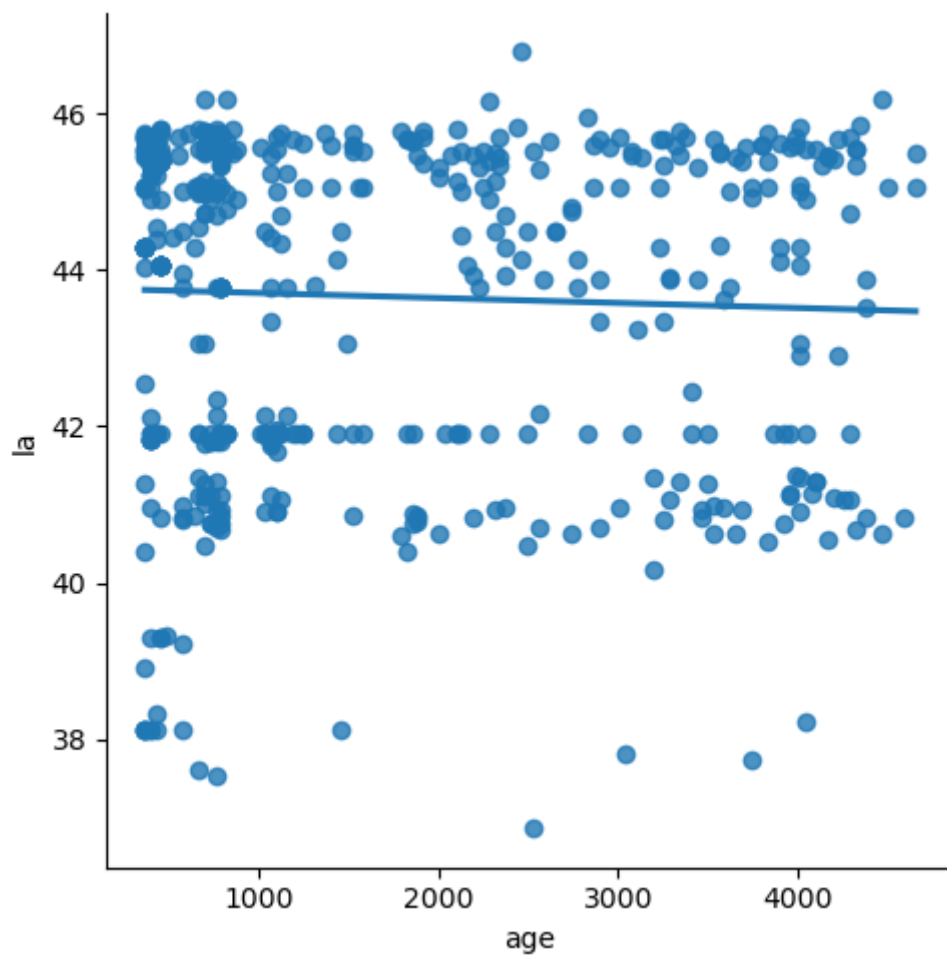


In [28]:

```
df500=df[:][:500]  
sns.lmplot(x='age',y='la',data=df500,order=1,ci=None)
```

Out[28]:

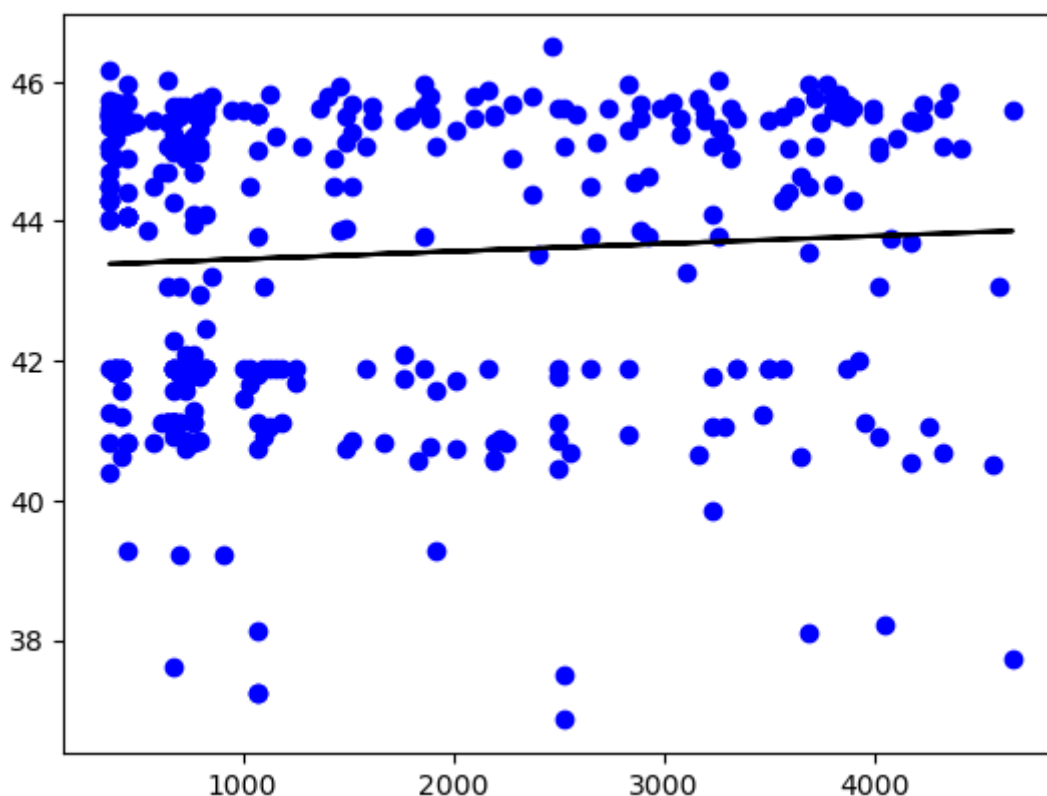
<seaborn.axisgrid.FacetGrid at 0x22f7838b4c0>



In [29]:

```
df500.fillna(method='ffill',inplace=True)
X=np.array(df['age']).reshape(-1,1)
y=np.array(df['la']).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))
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: 0.001914892584655381



In [30]:

```
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: 0.001914892584655381

In [31]:

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

Out[31]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwa:
0	13300000	7420	4	2	3	yes	no	no	
1	12250000	8960	4	4	4	yes	no	no	
2	12250000	9960	3	2	2	yes	no	yes	
3	12215000	7500	4	2	2	yes	no	yes	
4	11410000	7420	4	1	2	yes	yes	yes	
...	
540	1820000	3000	2	1	1	yes	no	yes	
541	1767150	2400	3	1	1	no	no	no	
542	1750000	3620	2	1	1	yes	no	no	
543	1750000	2910	3	1	1	no	no	no	
544	1750000	3850	3	1	2	yes	no	no	

545 rows × 13 columns



In [32]:

```
df=df[['price','area']]
df.columns=['price','area']
```

In [33]:

```
df.head(14)
```

Out[33]:

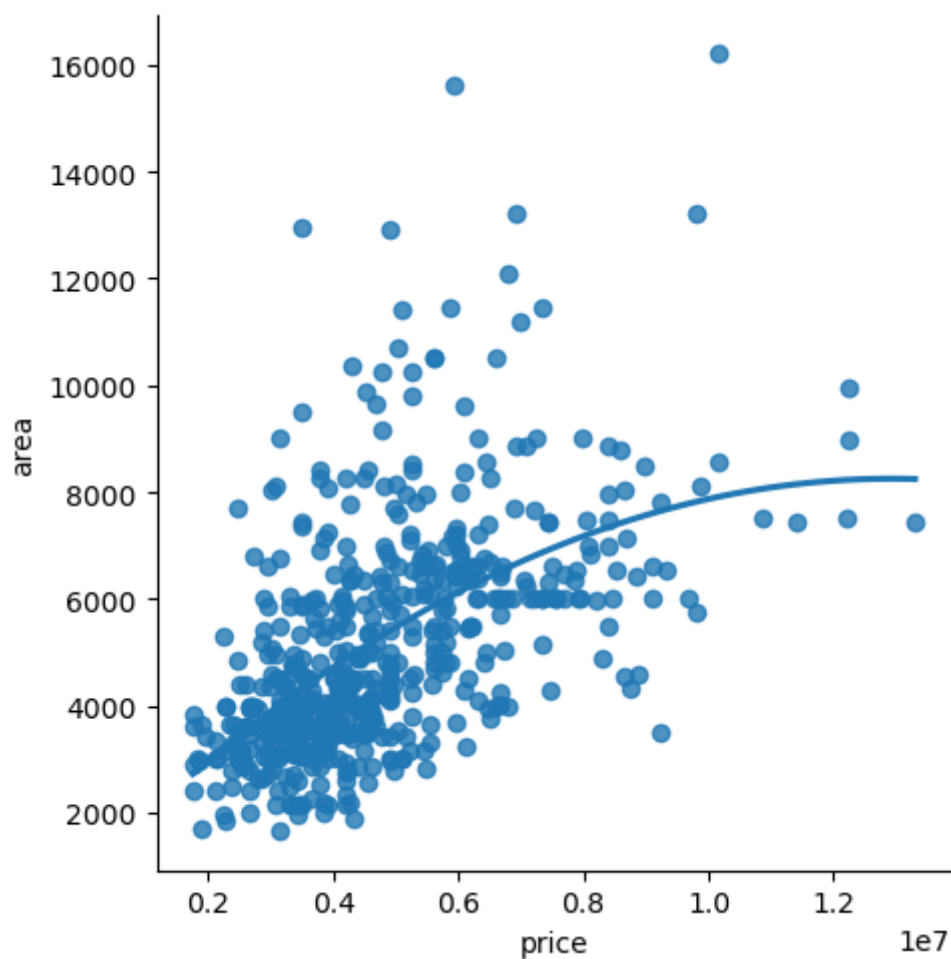
	price	area
0	13300000	7420
1	12250000	8960
2	12250000	9960
3	12215000	7500
4	11410000	7420
5	10850000	7500
6	10150000	8580
7	10150000	16200
8	9870000	8100
9	9800000	5750
10	9800000	13200
11	9681000	6000
12	9310000	6550
13	9240000	3500

In [34]:

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

Out[34]:

<seaborn.axisgrid.FacetGrid at 0x22f811e4d90>



In [35]:

```
df.describe()
```

Out[35]:

	price	area
count	5.450000e+02	545.000000
mean	4.766729e+06	5150.541284
std	1.870440e+06	2170.141023
min	1.750000e+06	1650.000000
25%	3.430000e+06	3600.000000
50%	4.340000e+06	4600.000000
75%	5.740000e+06	6360.000000
max	1.330000e+07	16200.000000

In [36]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 545 entries, 0 to 544
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  -
 0   price   545 non-null     int64
 1   area    545 non-null     int64
dtypes: int64(2)
memory usage: 8.6 KB
```

In [37]:

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

C:\Users\Teju\AppData\Local\Temp\ipykernel_19136\4116506308.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 [38]:

```
X=np.array(df['price']).reshape(-1,1)
y=np.array(df['area']).reshape(-1,1)
```

In [39]:

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

C:\Users\Teju\AppData\Local\Temp\ipykernel_19136\658506679.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)

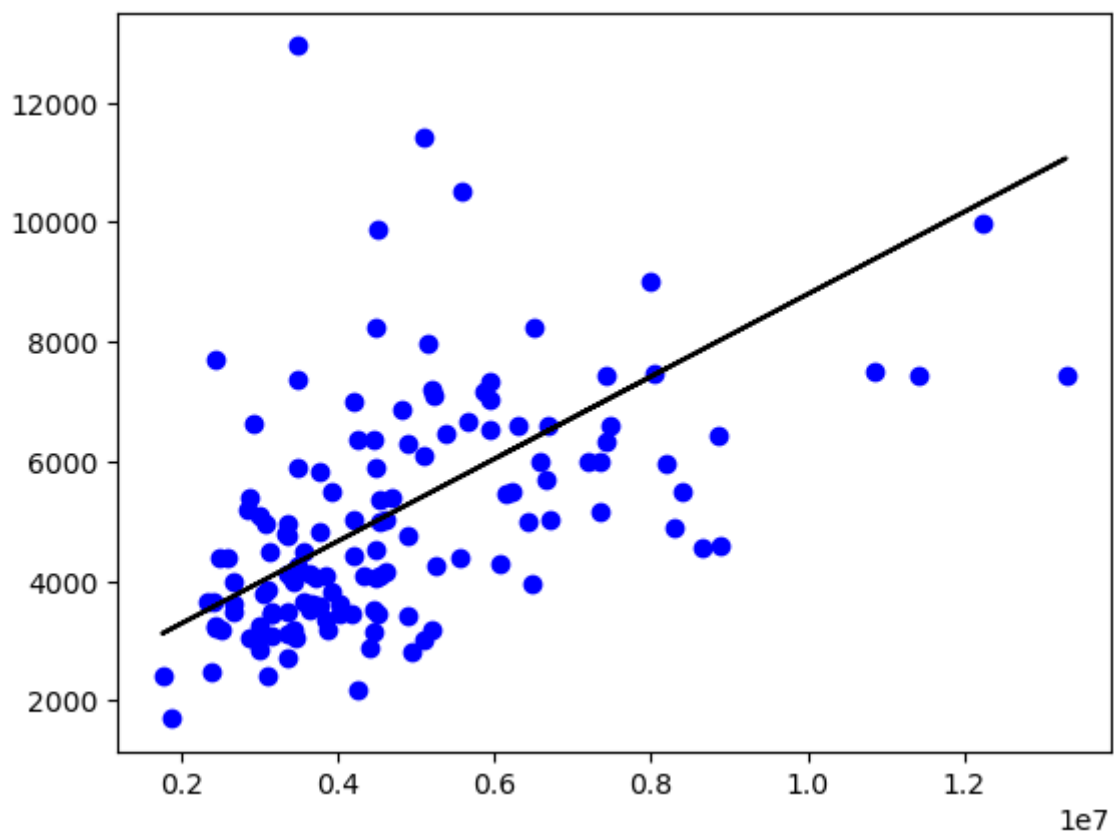
In [40]:

```
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))
```

0.18730964301965647

In [41]:

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

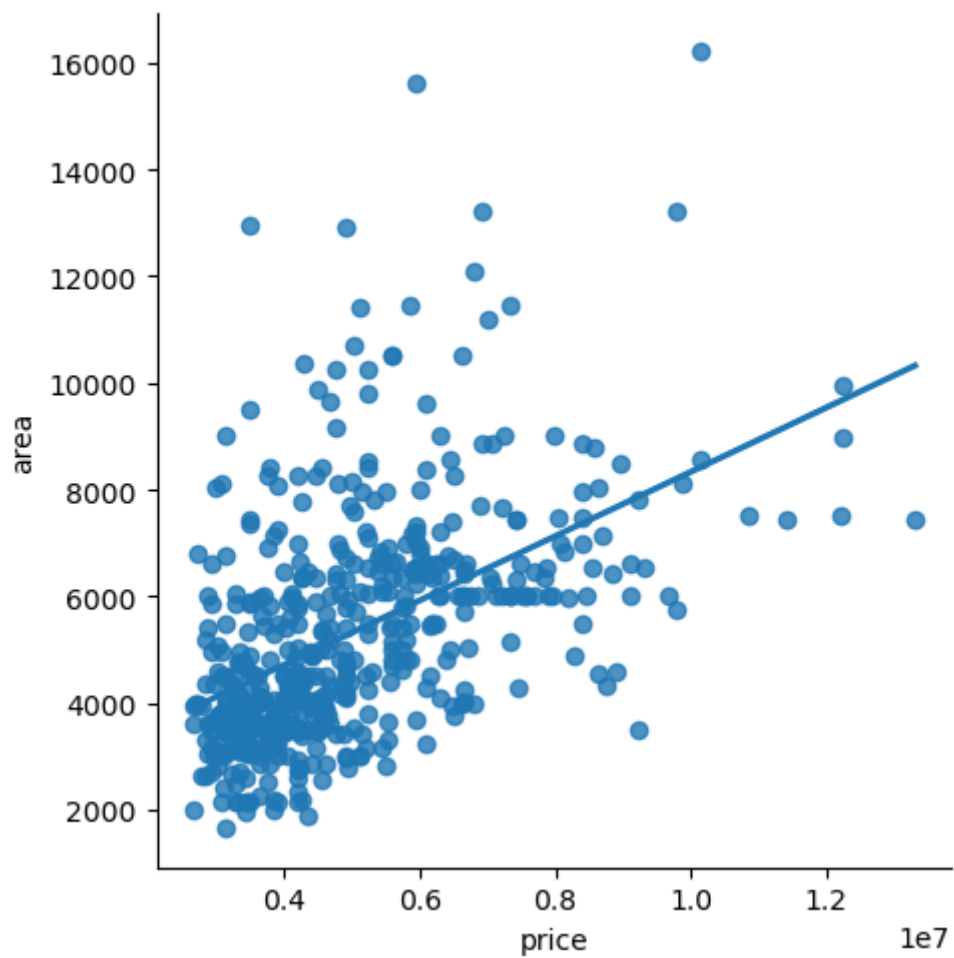


In [42]:

```
df500=df[:][:500]  
sns.lmplot(x='price',y='area',data=df500,order=1,ci=None)
```

Out[42]:

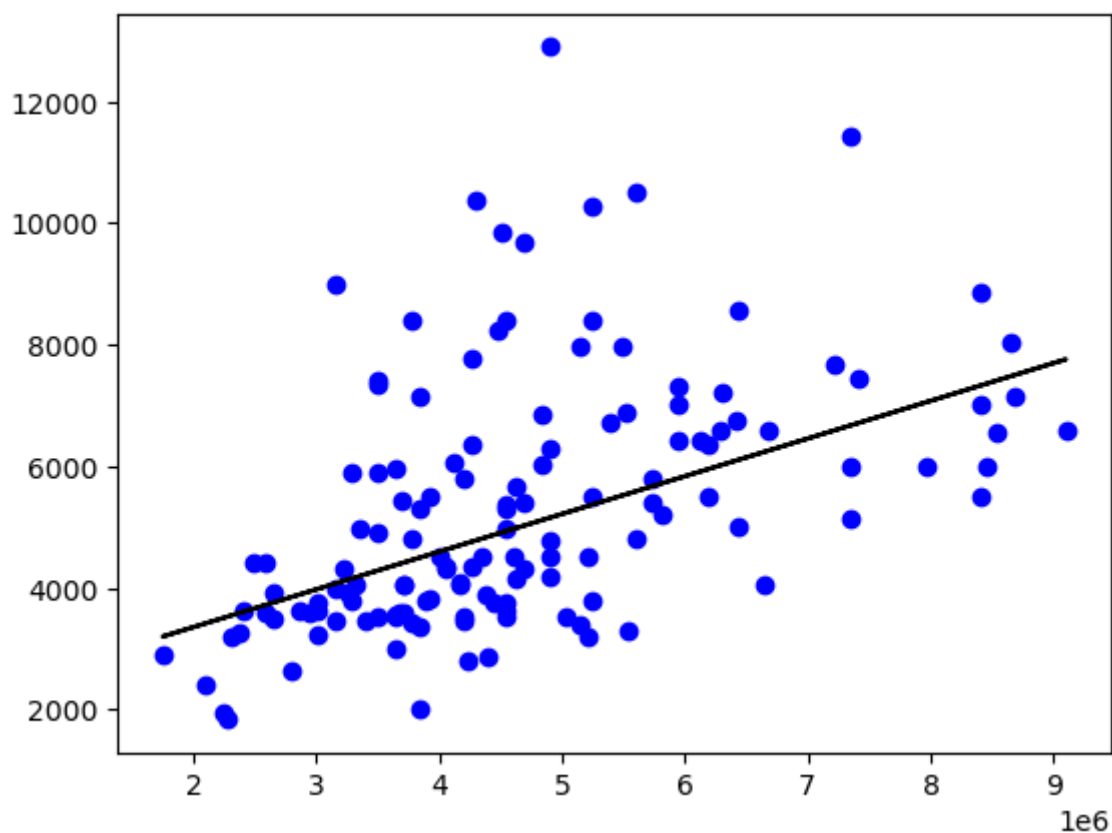
<seaborn.axisgrid.FacetGrid at 0x22f81277a30>



In [43]:

```
df500.fillna(method='ffill',inplace=True)
X=np.array(df['price']).reshape(-1,1)
y=np.array(df['area']).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))
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: 0.21671385576251678



In [44]:

```
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: 0.21671385576251678

