In [1]:

```
pip install --upgrade pip
```

```
Requirement already satisfied: pip in c:\users\sowmika\appdata\local\progr
ams\python\python310\lib\site-packages (23.1.2)Note: you may need to resta
rt the kernel to use updated packages.
```

In [2]:

```
pip install seaborn
```

```
Requirement already satisfied: seaborn in c:\users\sowmika\appdata\local\p
rograms\python\python310\lib\site-packages (0.12.2)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in c:\users\sowmika\ap
pdata\local\programs\python\python310\lib\site-packages (from seaborn) (1.
24.3)
Requirement already satisfied: pandas>=0.25 in c:\users\sowmika\appdata\lo
cal\programs\python\python310\lib\site-packages (from seaborn) (2.0.1)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in c:\users\sowmika
\appdata\local\programs\python\python310\lib\site-packages (from seaborn)
(3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\sowmika\appdat
a\local\programs\python\python310\lib\site-packages (from matplotlib!=3.6.
1,>=3.1->seaborn) (1.0.7)
Requirement already satisfied: cycler>=0.10 in c:\users\sowmika\appdata\lo
cal\programs\python\python310\lib\site-packages (from matplotlib!=3.6.1,>=
3.1->seaborn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\sowmika\appda
ta\local\programs\python\python310\lib\site-packages (from matplotlib!=3.
6.1,>=3.1->seaborn) (4.39.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\sowmika\appda
ta\local\programs\python\python310\lib\site-packages (from matplotlib!=3.
6.1,>=3.1->seaborn) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\sowmika\appdata
\local\programs\python\python310\lib\site-packages (from matplotlib!=3.6.
1,>=3.1->seaborn) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\sowmika\appdata\l
ocal\programs\python\python310\lib\site-packages (from matplotlib!=3.6.1,>
=3.1->seaborn) (9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\sowmika\appdat
a\local\programs\python\python310\lib\site-packages (from matplotlib!=3.6.
1,>=3.1->seaborn) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\sowmika\ap
pdata\local\programs\python\python310\lib\site-packages (from matplotlib!=
3.6.1,>=3.1->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\sowmika\appdata\lo
cal\programs\python\python310\lib\site-packages (from pandas>=0.25->seabor
n) (2023.3)
Requirement already satisfied: tzdata>=2022.1 in c:\users\sowmika\appdata
\local\programs\python\python310\lib\site-packages (from pandas>=0.25->sea
born) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\sowmika\appdata\local
\programs\python\python310\lib\site-packages (from python-dateutil>=2.7->m
atplotlib!=3.6.1,>=3.1->seaborn) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [3]:

```
!pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in c:\users\sowmika\appdata\lo
cal\programs\python\python310\lib\site-packages (1.2.2)
Requirement already satisfied: numpy>=1.17.3 in c:\users\sowmika\appdata\l
ocal\programs\python\python310\lib\site-packages (from scikit-learn) (1.2
4.3)
Requirement already satisfied: scipy>=1.3.2 in c:\users\sowmika\appdata\lo
cal\programs\python\python310\lib\site-packages (from scikit-learn) (1.10.
1)
Requirement already satisfied: joblib>=1.1.1 in c:\users\sowmika\appdata\l
ocal\programs\python\python310\lib\site-packages (from scikit-learn) (1.2.
0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\sowmika\ap
pdata\local\programs\python\python310\lib\site-packages (from scikit-lear
n) (3.1.0)
```

In [4]:

```
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 [5]:

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

```
C:\Users\sowmika\AppData\Local\Temp\ipykernel_19392\259309552.py:1: DtypeW
arning: Columns (47,73) have mixed types. Specify dtype option on import o
r set low_memory=False.
  df=pd.read_csv(r"C:\Users\sowmika\Downloads\bottle.csv.zip")
```

Out[5]:

l_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat	...	R_PHAEO
1	054.0 056.0	19-4903CR-HY-060-0930-05400560-0000A-3	0	10.500	33.4400	NaN	25.64900	NaN	...	NaN
2	054.0 056.0	19-4903CR-HY-060-0930-05400560-0008A-3	8	10.460	33.4400	NaN	25.65600	NaN	...	NaN
3	054.0 056.0	19-4903CR-HY-060-0930-05400560-0010A-7	10	10.460	33.4370	NaN	25.65400	NaN	...	NaN
4	054.0 056.0	19-4903CR-HY-060-0930-05400560-0019A-3	19	10.450	33.4200	NaN	25.64300	NaN	...	NaN
5	054.0 056.0	19-4903CR-HY-060-0930-05400560-0020A-7	20	10.450	33.4210	NaN	25.64300	NaN	...	NaN
...	...	...	...	...	...	...	...	...	...	...
64859	093.4 026.4	20-1611SR-MX-310-2239-09340264-0000A-7	0	18.744	33.4083	5.805	23.87055	108.74	...	0.18
64860	093.4 026.4	20-1611SR-MX-310-2239-09340264-0002A-3	2	18.744	33.4083	5.805	23.87072	108.74	...	0.18
64861	093.4 026.4	20-1611SR-MX-310-2239-09340264-0005A-3	5	18.692	33.4150	5.796	23.88911	108.46	...	0.18
64862	093.4 026.4	20-1611SR-MX-310-2239-09340264-0010A-3	10	18.161	33.4062	5.816	24.01426	107.74	...	0.31

In [6]:

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

In [7]:

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

Out[7]:

	sal	Temp
0	33.440	10.50
1	33.440	10.46
2	33.437	10.46
3	33.420	10.45
4	33.421	10.45
5	33.431	10.45
6	33.440	10.45
7	33.424	10.24
8	33.420	10.06
9	33.494	9.86

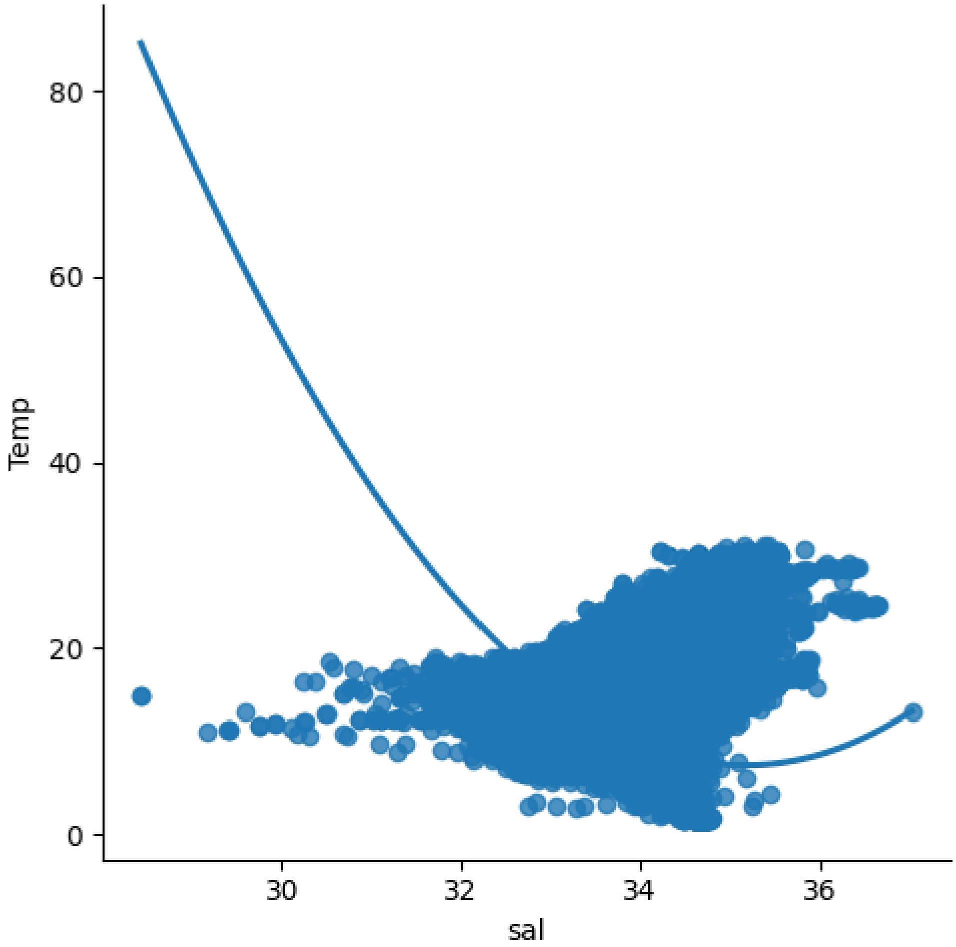
l_Cnt	Sta_ID	Depth_ID	Depthm	T_degC	Salnty	O2ml_L	STheta	O2Sat	...	R_PHAEO
54863	093.4 026.4	20- 1611SR- MX-310- 2239- 09340264-	15	17.533	33.3880	5.774	24.15297	105.66	...	0.61

In [8]:

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

Out[8]:

```
<seaborn.axisgrid.FacetGrid at 0x1f5dceae950>
```



In [9]:

```
df.describe()
```

Out[9]:

	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 [10]:

```
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 [29]:

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

```
C:\Users\sowmika\AppData\Local\Temp\ipykernel_19392\3028625988.py:1: Setti
ngWithCopyWarning: 
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 [12]:

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

In [30]:

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

```
C:\Users\sowmika\AppData\Local\Temp\ipykernel_19392\1791587065.py:1: Setti
ngWithCopyWarning: 
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)
```

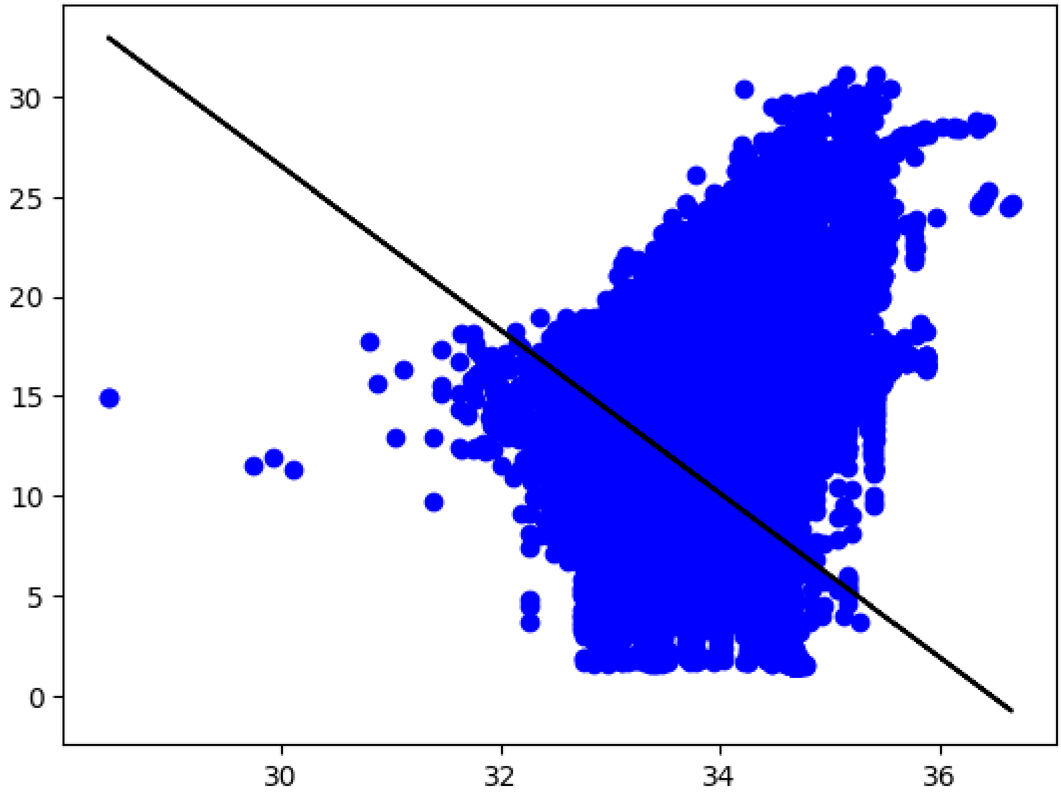
In [14]:

```
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))
```

```
0.2036387277835181
```

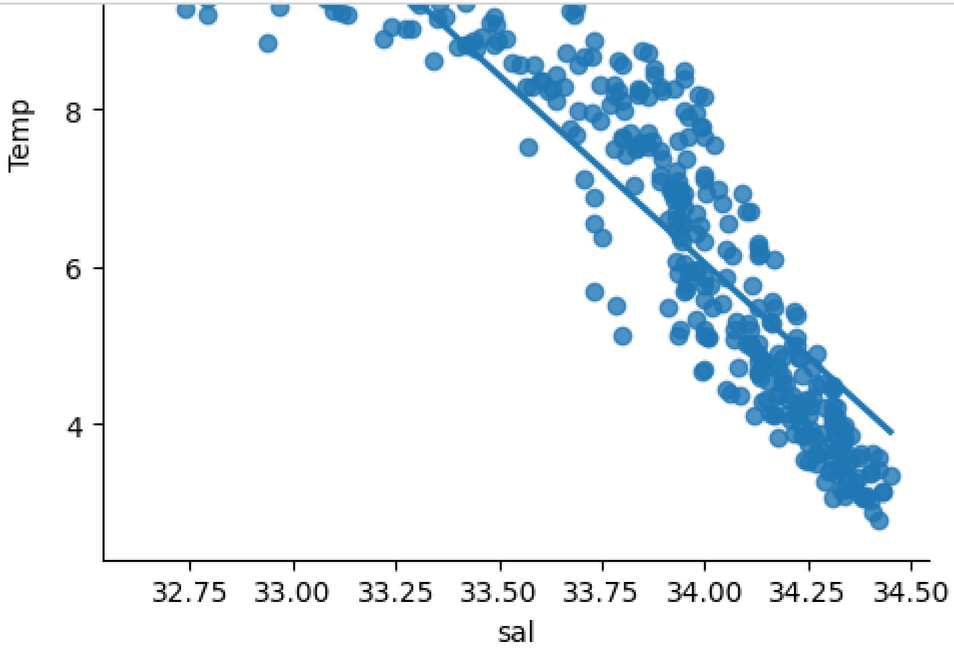
In [15]:

```
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 [16]:

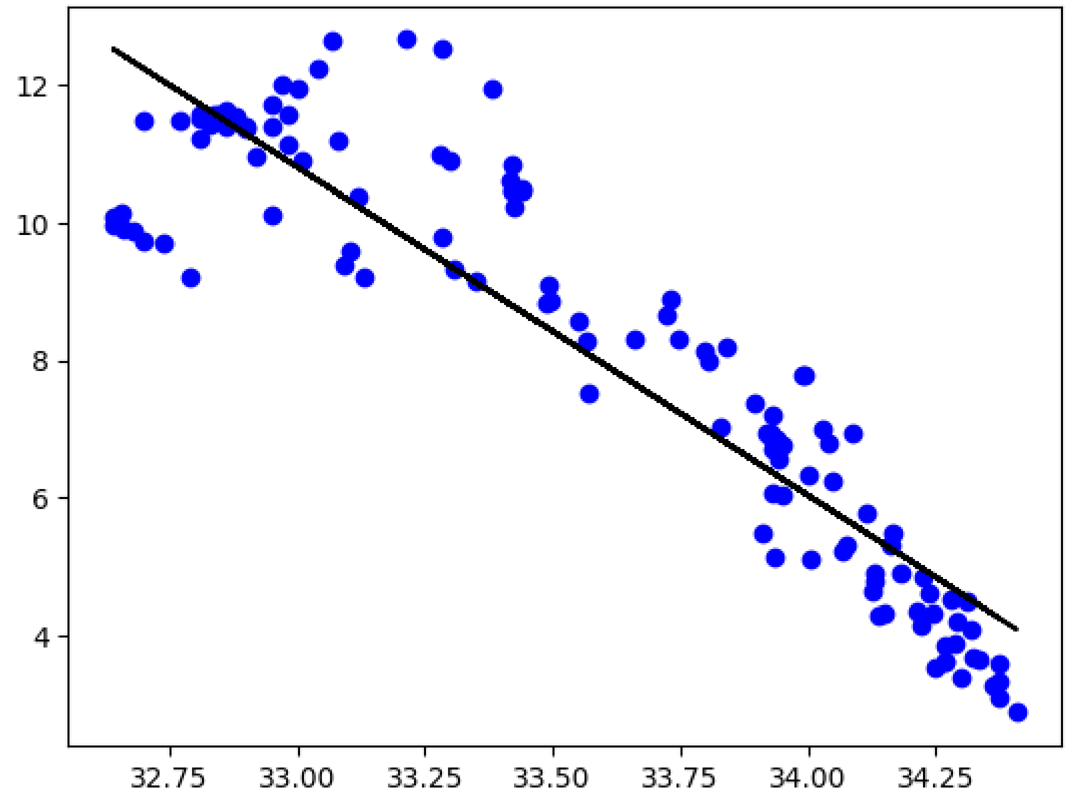
```
df500 = df[:][:500]
sns.lmplot(x ="sal",y ="Temp", data = df500,order = 1,ci = None)
```



In [17]:

```
df500.fillna(method = 'ffill',inplace = True)
X = np. array(df500['sal']).reshape(-1, 1)
y = np.array(df500['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: 0.8429543309817631
```



In [18]:

```
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: 0.8429543309817631
```

vehicle selection

In [19]:

```
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 [47]:

```
df=pd.read_csv(r"C:\Users\sowmika\Downloads\fiat500_VehicleSelection_Dataset (1).csv")
df
```

Out[47]:

```
        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 [48]:

```
df=df[['age_in_days','km']]
df.columns=['age','km']
```

In [49]:

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

Out[49]:

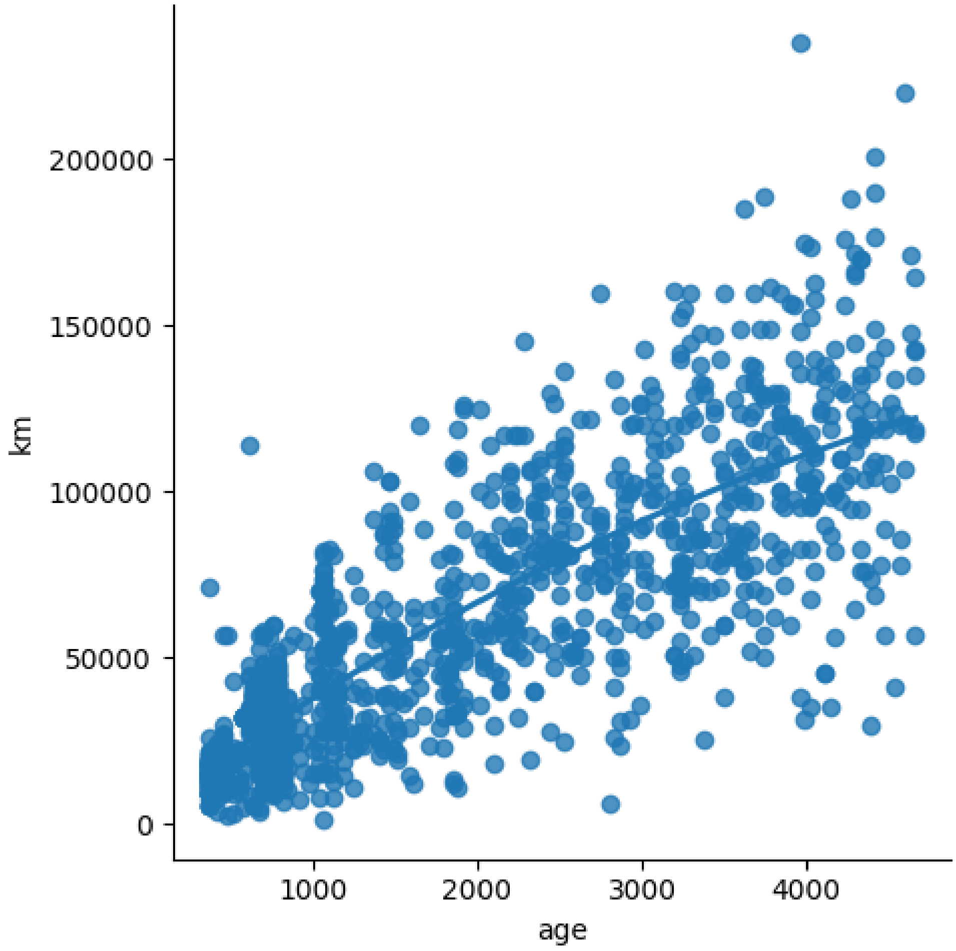
	age	km
0	882	25000
1	1186	32500
2	4658	142228
3	2739	160000
4	3074	106880
5	3623	70225
6	731	11600
7	1521	49076
8	4049	76000
9	3653	89000

In [50]:

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

Out[50]:

```
<seaborn.axisgrid.FacetGrid at 0x1f58a24eaa0>
```



In [51]:

```
df.describe()
```

Out[51]:

	age	km
count	1538.000000	1538.000000
mean	1650.980494	53396.011704
std	1289.522278	40046.830723
min	366.000000	1232.000000
25%	670.000000	20006.250000
50%	1035.000000	39031.000000
75%	2616.000000	79667.750000
max	4658.000000	235000.000000

In [52]:

```
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   km      1538 non-null   int64
dtypes: int64(2)
memory usage: 24.2 KB
```

In [53]:

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

```
C:\Users\sowmika\AppData\Local\Temp\ipykernel_19392\3028625988.py:1: Setti
ngWithCopyWarning:
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 [54]:

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

In [55]:

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

```
C:\Users\sowmika\AppData\Local\Temp\ipykernel_19392\1791587065.py:1: Setti
ngWithCopyWarning:
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)
```

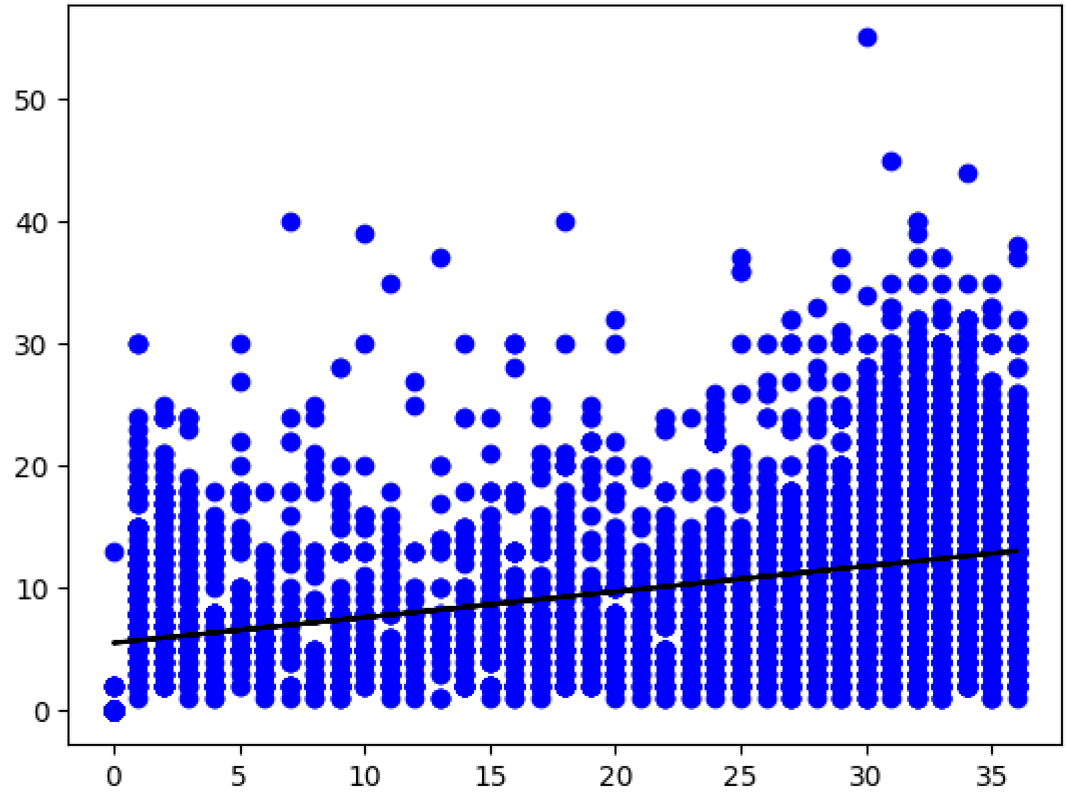
In [56]:

```
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.25)
regr = LinearRegression()y_pred=regr.predict(X_test)
plt.scatter(X_test, Y_test,color='b')
plt.plot(X_test,y_pred,color='k')
plt.show()
regr.fit(X_train, Y_train)
print(regr.score(X_test, Y_test))
```

```
0.6885339929552072
```

In [32]:

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

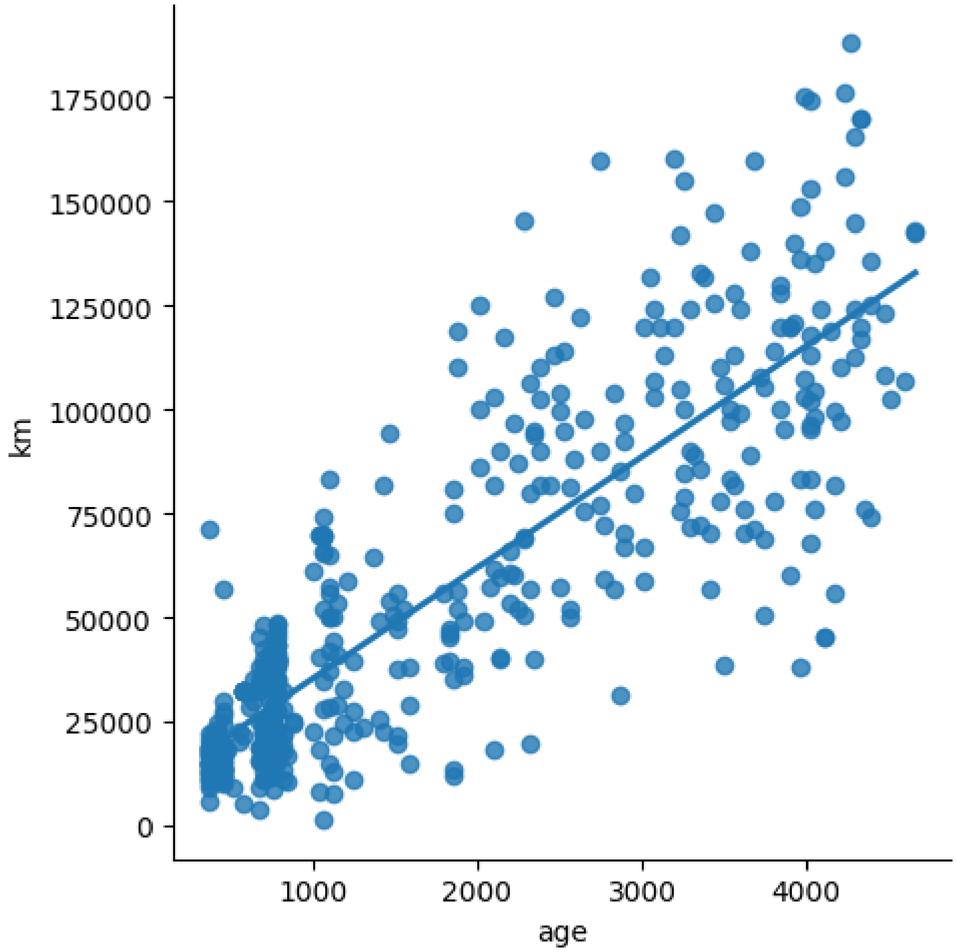


In [61]:

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

Out[61]:

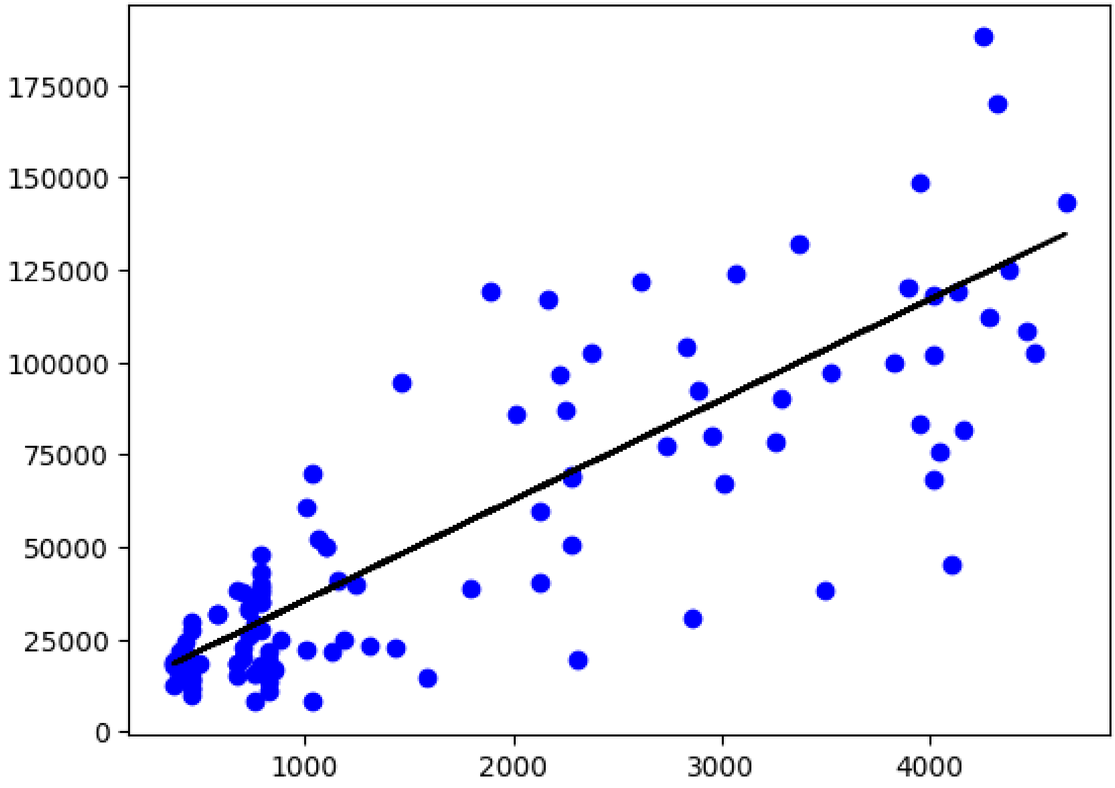
```
<seaborn.axisgrid.FacetGrid at 0x1f58a516fb0>
```



In [62]:

```
df500.fillna(method = 'ffill',inplace = True)
X = np. array(df500['age']).reshape(-1, 1)
y = np.array(df500['km']).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: 0.7069135527596802
```



In [59]:

```
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: 0.7670566623468238
```

House price prediction

In [63]:

```
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 [64]:

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

Out[64]:

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront
0	2014-05-02 00:00:00	3.130000e+05	3.0	1.50	1340	7912	1.5	0
1	2014-05-02 00:00:00	2.384000e+06	5.0	2.50	3650	9050	2.0	0
2	2014-05-02 00:00:00	3.420000e+05	3.0	2.00	1930	11947	1.0	0
3	2014-05-02 00:00:00	4.200000e+05	3.0	2.25	2000	8030	1.0	0
4	2014-05-02 00:00:00	5.500000e+05	4.0	2.50	1940	10500	1.0	0
...	...	...	...	...	...	...	...	...
4595	2014-07-09 00:00:00	3.081667e+05	3.0	1.75	1510	6360	1.0	0
4596	2014-07-09 00:00:00	5.343333e+05	3.0	2.50	1460	7573	2.0	0
4597	2014-07-09 00:00:00	4.169042e+05	3.0	2.50	3010	7014	2.0	0
4598	2014-07-10 00:00:00	2.034000e+05	4.0	2.00	2090	6630	1.0	0
4599	2014-07-10 00:00:00	2.206000e+05	3.0	2.50	1490	8102	2.0	0

4600 rows × 18 columns

In [65]:

```
df=df[['sqft_living','sqft_lot']]
df.columns=['living','lot']
```

In [66]:

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

Out[66]:

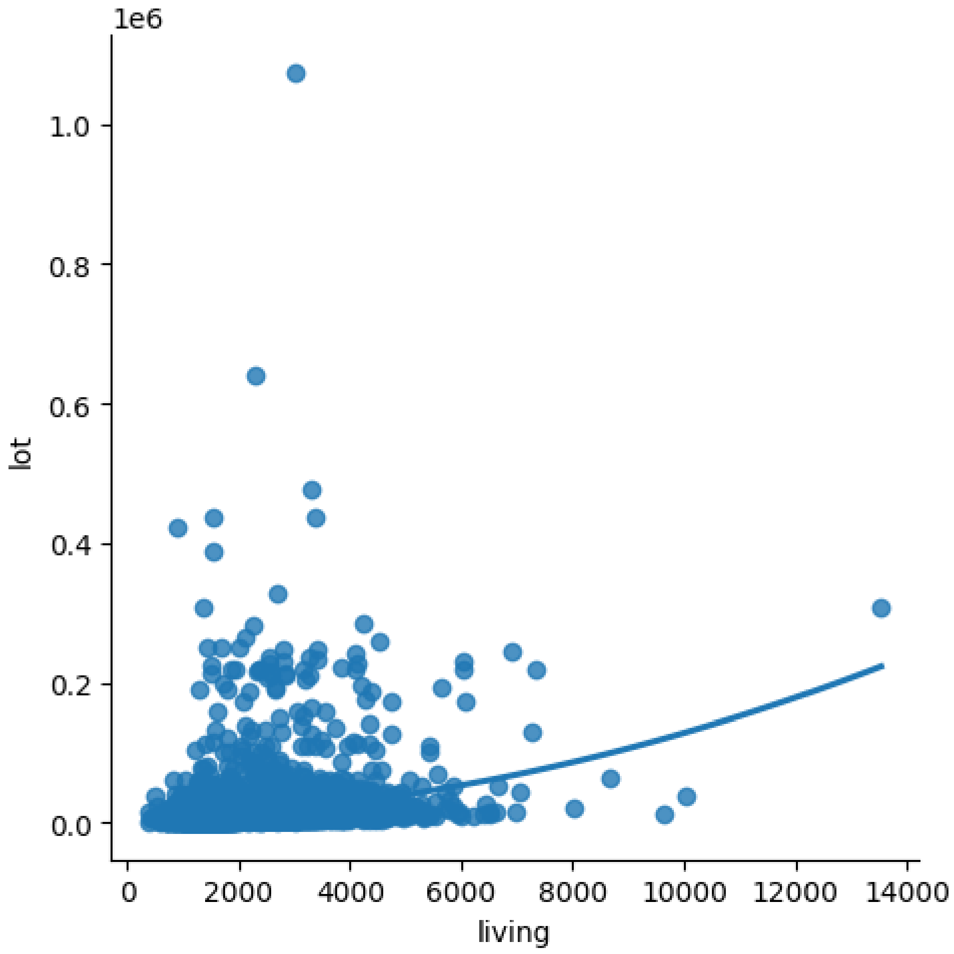
	living	lot
0	1340	7912
1	3650	9050
2	1930	11947
3	2000	8030
4	1940	10500
5	880	6380
6	1350	2560
7	2710	35868
8	2430	88426
9	1520	6200

In [67]:

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

Out[67]:

```
<seaborn.axisgrid.FacetGrid at 0x1f58a758f70>
```



In [69]:

```
df.describe()
```

Out[69]:

	living	lot
count	4600.000000	4.600000e+03
mean	2139.346957	1.485252e+04
std	963.206916	3.588444e+04
min	370.000000	6.380000e+02
25%	1460.000000	5.000750e+03
50%	1980.000000	7.683000e+03
75%	2620.000000	1.100125e+04
max	13540.000000	1.074218e+06

In [70]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4600 entries, 0 to 4599
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   living  4600 non-null   int64
 1   lot     4600 non-null   int64
dtypes: int64(2)
memory usage: 72.0 KB
```

In [71]:

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

```
C:\Users\sowmika\AppData\Local\Temp\ipykernel_19392\3028625988.py:1: Setti
ngWithCopyWarning:
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 [72]:

```
X = np. array(df['living']).reshape(-1, 1)
Y = np.array(df['lot']).reshape(-1, 1)
```

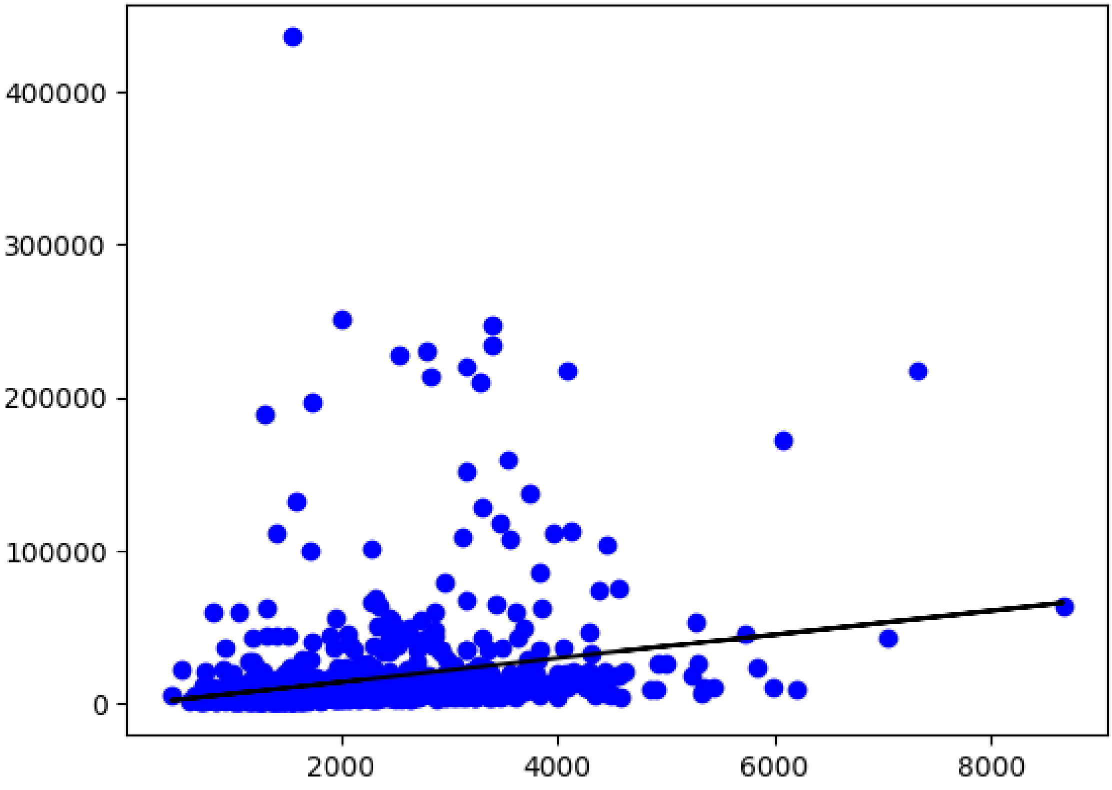
In [73]:

```
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))
```

```
0.06680986187711502
```

In [74]:

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

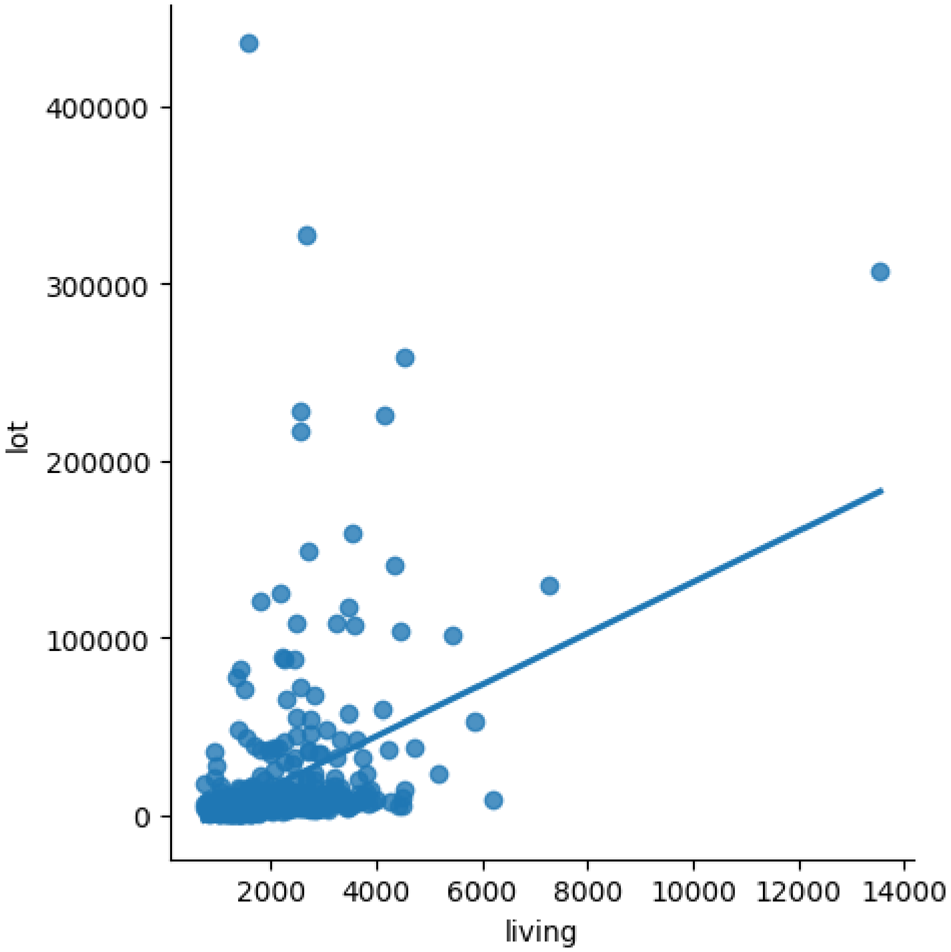


In [75]:

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

Out[75]:

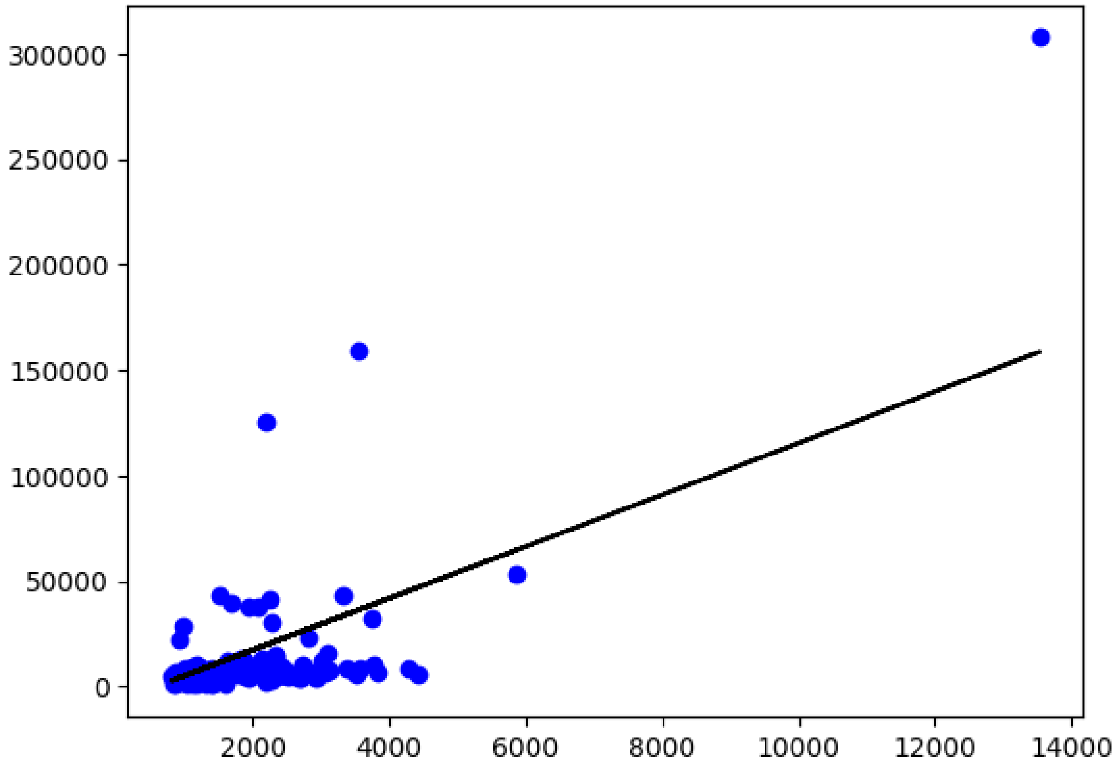
```
<seaborn.axisgrid.FacetGrid at 0x1f58a75a590>
```



In [79]:

```
df500.fillna(method = 'ffill',inplace = True)
X = np. array(df500['living']).reshape(-1, 1)
y = np.array(df500['lot']).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: 0.45239880007632904
```



In [80]:

```
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: 0.45239880007632904
```

In [ ]: