lab1_linear_regression

November 15, 2022

LINEAR REGRESSION

[ ]:
```
import numpy as np
import pandas as pd
from sklearn import linear_model
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error,accuracy_score
```

[ ]:
```
traindata=pd.read_csv("/content/crime_train.csv")
traindata.describe()
traindata.head()
```

[ ]:

	ID	population	householdsize	agePct12t21	agePct12t29	agePct16t24	\
0	1	14985	2.56	16.55	34.42	22.54	
1	2	30843	2.83	15.45	35.12	18.14	
2	3	74991	2.52	10.48	20.43	9.11	
3	4	45061	2.44	10.59	24.97	11.61	
4	5	12863	2.45	12.02	22.51	10.49	

	agePct65up	numbUrban	pctUrban	medIncome	…	MedOwnCostPctInc	\
0	10.13	0	0.0	35545	…	23.3	
1	4.70	0	0.0	32033	…	21.6	
2	20.68	73342	97.8	31177	…	23.6	
3	16.34	45061	100.0	39822	…	24.0	
4	18.46	0	0.0	23044	…	16.0	

	MedOwnCostPctIncNoMtg	NumInShelters	NumStreet	PctForeignBorn	\
0	13.5	0	0	3.32	
1	10.9	68	41	4.98	
2	11.9	0	0	6.51	
3	16.3	0	1	13.13	
4	11.7	0	0	1.08	

	PctBornSameState	PctSameHouse85	PctSameCity85	PctSameState85	\
0	60.94	47.28	66.65	82.41	
1	25.24	28.17	52.73	59.20	
2	13.69	34.49	60.88	66.97	
3	67.79	65.19	84.28	91.32	

```
4             80.22           53.17          86.56           93.15

   ViolentCrimesPerPop
0                428.64
1                742.54
2                303.72
3                373.88
4                108.07

[5 rows x 90 columns]
```

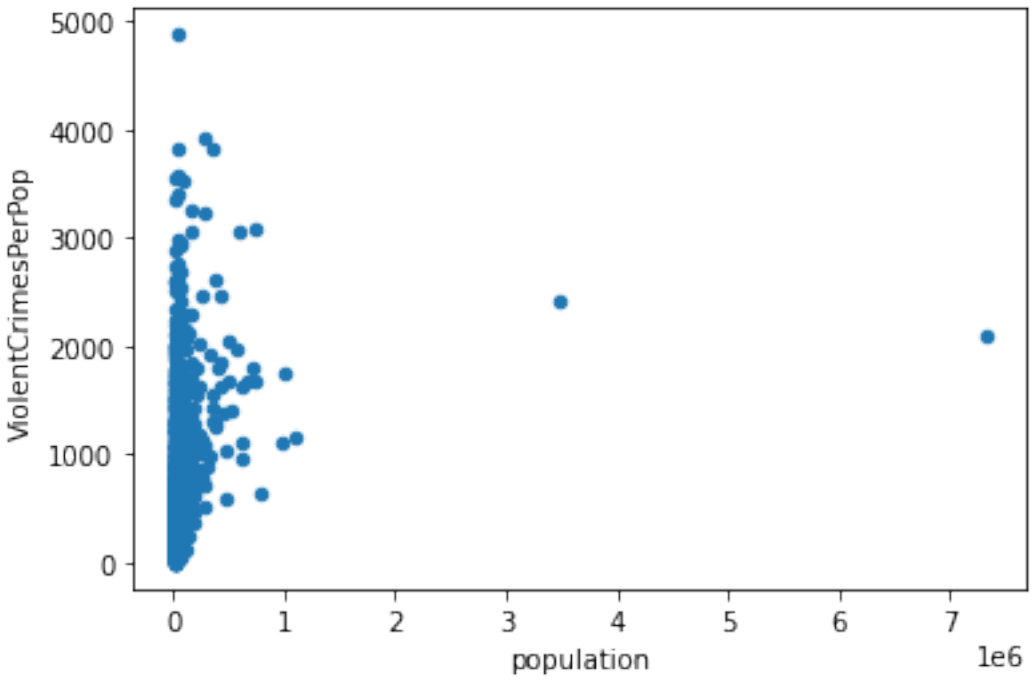
[ ]:
```
y=traindata["ViolentCrimesPerPop"]
traindata=traindata.drop(["ID"],axis=1)
X=traindata.drop(["ViolentCrimesPerPop"],axis=1)
x=traindata["population"]
```

[ ]:
```
rl=linear_model.LinearRegression()
rl.fit(X,y)
rl.coef_
```

[ ]:
```
array([ 7.14919936e-04,  1.23800100e+01, -8.28000652e+00, -3.58840708e+01,
        3.34549393e+01, -1.30305450e+01, -8.54932516e-04,  1.41109567e+00,
       -2.27552750e-02, -7.00821090e+00,  2.95727019e+01, -6.67902639e+00,
        1.15458646e+01,  8.82161707e+00, -8.30115048e+00,  1.83594413e-02,
       -4.98509797e-03, -2.78001000e-03, -2.94824284e+00, -1.43136672e+01,
        4.97783566e+00,  1.75724402e+00, -1.30374690e+00,  1.20918533e+01,
       -4.06281211e+00,  2.15366190e+00, -4.99653775e-01,  4.22547268e+00,
        2.00271997e+02, -2.07777399e+00,  1.43816056e+02, -3.32259909e+02,
       -4.05926926e+02, -2.43207388e+00, -1.93158681e+01,  2.92604083e+00,
       -7.24481971e-01,  3.29895767e+00, -1.01925764e+01, -1.42096878e-03,
        5.64015457e+01,  8.53412360e-04,  2.72361194e+00, -1.48151566e+00,
       -1.37678352e-03,  1.12487361e+00, -4.05246604e+01, -1.47028063e+01,
        1.30649975e+01,  6.46421199e-01,  2.32781012e+00, -1.30391028e+01,
        4.72734257e+01, -7.20762077e+01,  3.65621264e+02,  6.07187018e+02,
       -3.58006983e+02, -5.18526869e+01,  2.25870300e+01,  3.33424415e+00,
        5.68650282e+01,  2.26150440e-02, -1.94529481e+00,  5.00177702e+01,
        1.42080174e+01, -2.04950217e+00,  1.00205869e+00,  9.08003736e-01,
       -1.86145026e+01,  4.24723319e-04, -1.78351018e-04, -7.01954979e-05,
       -4.94948554e-04, -5.84075526e-01, -5.50825347e-01, -4.51515791e-01,
        1.32559734e-01,  1.68001115e+00, -4.96248438e+00, -7.16664364e+00,
       -1.91599715e+01,  4.50683044e-02, -2.25101784e-02,  1.48603114e+01,
        1.16333653e+00, -8.25052651e-01,  2.72834875e+00, -3.01647322e+00])
```

[ ]:
```
traindata.plot(y="ViolentCrimesPerPop",x="population",kind="scatter")
```

[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd3f208f790>
```



[ ]:
```
testdata=pd.read_csv("/content/crime_test.csv")
traindata.info()
testdata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1595 entries, 0 to 1594
Data columns (total 89 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   population            1595 non-null   int64
 1   householdsize         1595 non-null   float64
 2   agePct12t21           1595 non-null   float64
 3   agePct12t29           1595 non-null   float64
 4   agePct16t24           1595 non-null   float64
 5   agePct65up            1595 non-null   float64
 6   numbUrban             1595 non-null   int64
 7   pctUrban              1595 non-null   float64
 8   medIncome             1595 non-null   int64
 9   pctWWage              1595 non-null   float64
 10  pctWFarmSelf          1595 non-null   float64
 11  pctWInvInc            1595 non-null   float64
 12  pctWSocSec            1595 non-null   float64
 13  pctWPubAsst           1595 non-null   float64
 14  pctWRetire            1595 non-null   float64
 15  medFamInc             1595 non-null   int64
```

```
16  perCapInc              1595 non-null   int64
17  NumUnderPov            1595 non-null   int64
18  PctPopUnderPov         1595 non-null   float64
19  PctLess9thGrade        1595 non-null   float64
20  PctNotHSGrad           1595 non-null   float64
21  PctBSorMore            1595 non-null   float64
22  PctUnemployed          1595 non-null   float64
23  PctEmploy              1595 non-null   float64
24  PctEmplManu            1595 non-null   float64
25  PctEmplProfServ        1595 non-null   float64
26  PctOccupManu           1595 non-null   float64
27  PctOccupMgmtProf       1595 non-null   float64
28  MalePctDivorce         1595 non-null   float64
29  MalePctNevMarr         1595 non-null   float64
30  FemalePctDiv           1595 non-null   float64
31  TotalPctDiv            1595 non-null   float64
32  PersPerFam             1595 non-null   float64
33  PctFam2Par             1595 non-null   float64
34  PctKids2Par            1595 non-null   float64
35  PctYoungKids2Par       1595 non-null   float64
36  PctTeen2Par            1595 non-null   float64
37  PctWorkMomYoungKids    1595 non-null   float64
38  PctWorkMom             1595 non-null   float64
39  NumKidsBornNeverMar    1595 non-null   int64
40  PctKidsBornNeverMar    1595 non-null   float64
41  NumImmig               1595 non-null   int64
42  PctImmigRecent         1595 non-null   float64
43  PctImmigRec5           1595 non-null   float64
44  PctImmigRec8           1595 non-null   float64
45  PctImmigRec10          1595 non-null   float64
46  PctRecentImmig         1595 non-null   float64
47  PctRecImmig5           1595 non-null   float64
48  PctRecImmig8           1595 non-null   float64
49  PctRecImmig10          1595 non-null   float64
50  PctSpeakEnglOnly       1595 non-null   float64
51  PctNotSpeakEnglWell    1595 non-null   float64
52  PctLargHouseFam        1595 non-null   float64
53  PctLargHouseOccup      1595 non-null   float64
54  PersPerOccupHous       1595 non-null   float64
55  PersPerOwnOccHous      1595 non-null   float64
56  PersPerRentOccHous     1595 non-null   float64
57  PctPersOwnOccup        1595 non-null   float64
58  PctPersDenseHous       1595 non-null   float64
59  PctHousLess3BR         1595 non-null   float64
60  MedNumBR               1595 non-null   int64
61  HousVacant             1595 non-null   int64
62  PctHousOccup           1595 non-null   float64
63  PctHousOwnOcc          1595 non-null   float64
```

```
 64  PctVacantBoarded       1595 non-null   float64
 65  PctVacMore6Mos         1595 non-null   float64
 66  MedYrHousBuilt         1595 non-null   int64
 67  PctHousNoPhone         1595 non-null   float64
 68  PctWOFullPlumb         1595 non-null   float64
 69  OwnOccLowQuart         1595 non-null   int64
 70  OwnOccMedVal           1595 non-null   int64
 71  OwnOccHiQuart          1595 non-null   int64
 72  OwnOccQrange           1595 non-null   int64
 73  RentLowQ               1595 non-null   int64
 74  RentMedian             1595 non-null   int64
 75  RentHighQ              1595 non-null   int64
 76  RentQrange             1595 non-null   int64
 77  MedRent                1595 non-null   int64
 78  MedRentPctHousInc      1595 non-null   float64
 79  MedOwnCostPctInc       1595 non-null   float64
 80  MedOwnCostPctIncNoMtg  1595 non-null   float64
 81  NumInShelters          1595 non-null   int64
 82  NumStreet              1595 non-null   int64
 83  PctForeignBorn         1595 non-null   float64
 84  PctBornSameState       1595 non-null   float64
 85  PctSameHouse85         1595 non-null   float64
 86  PctSameCity85          1595 non-null   float64
 87  PctSameState85         1595 non-null   float64
 88  ViolentCrimesPerPop    1595 non-null   float64
dtypes: float64(67), int64(22)
memory usage: 1.1 MB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 399 entries, 0 to 398
Data columns (total 89 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   ID                     399 non-null    int64
 1   population             399 non-null    int64
 2   householdsize          399 non-null    float64
 3   agePct12t21            399 non-null    float64
 4   agePct12t29            399 non-null    float64
 5   agePct16t24            399 non-null    float64
 6   agePct65up             399 non-null    float64
 7   numbUrban              399 non-null    int64
 8   pctUrban               399 non-null    float64
 9   medIncome              399 non-null    int64
 10  pctWWage               399 non-null    float64
 11  pctWFarmSelf           399 non-null    float64
 12  pctWInvInc             399 non-null    float64
 13  pctWSocSec             399 non-null    float64
 14  pctWPubAsst            399 non-null    float64
 15  pctWRetire             399 non-null    float64
```

```
16  medFamInc              399 non-null    int64
17  perCapInc              399 non-null    int64
18  NumUnderPov            399 non-null    int64
19  PctPopUnderPov         399 non-null    float64
20  PctLess9thGrade        399 non-null    float64
21  PctNotHSGrad           399 non-null    float64
22  PctBSorMore            399 non-null    float64
23  PctUnemployed          399 non-null    float64
24  PctEmploy              399 non-null    float64
25  PctEmplManu            399 non-null    float64
26  PctEmplProfServ        399 non-null    float64
27  PctOccupManu           399 non-null    float64
28  PctOccupMgmtProf       399 non-null    float64
29  MalePctDivorce         399 non-null    float64
30  MalePctNevMarr         399 non-null    float64
31  FemalePctDiv           399 non-null    float64
32  TotalPctDiv            399 non-null    float64
33  PersPerFam             399 non-null    float64
34  PctFam2Par             399 non-null    float64
35  PctKids2Par            399 non-null    float64
36  PctYoungKids2Par       399 non-null    float64
37  PctTeen2Par            399 non-null    float64
38  PctWorkMomYoungKids    399 non-null    float64
39  PctWorkMom             399 non-null    float64
40  NumKidsBornNeverMar    399 non-null    int64
41  PctKidsBornNeverMar    399 non-null    float64
42  NumImmig               399 non-null    int64
43  PctImmigRecent         399 non-null    float64
44  PctImmigRec5           399 non-null    float64
45  PctImmigRec8           399 non-null    float64
46  PctImmigRec10          399 non-null    float64
47  PctRecentImmig         399 non-null    float64
48  PctRecImmig5           399 non-null    float64
49  PctRecImmig8           399 non-null    float64
50  PctRecImmig10          399 non-null    float64
51  PctSpeakEnglOnly       399 non-null    float64
52  PctNotSpeakEnglWell    399 non-null    float64
53  PctLargHouseFam        399 non-null    float64
54  PctLargHouseOccup      399 non-null    float64
55  PersPerOccupHous       399 non-null    float64
56  PersPerOwnOccHous      399 non-null    float64
57  PersPerRentOccHous     399 non-null    float64
58  PctPersOwnOccup        399 non-null    float64
59  PctPersDenseHous       399 non-null    float64
60  PctHousLess3BR         399 non-null    float64
61  MedNumBR               399 non-null    int64
62  HousVacant             399 non-null    int64
63  PctHousOccup           399 non-null    float64
```

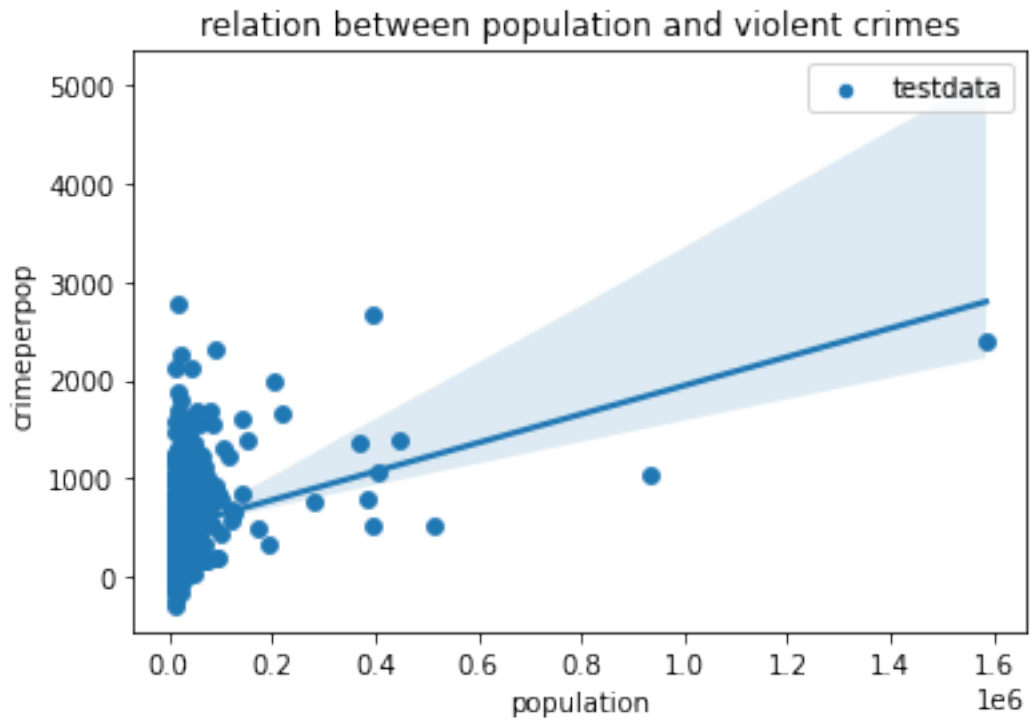
```
 64  PctHousOwnOcc           399 non-null    float64
 65  PctVacantBoarded        399 non-null    float64
 66  PctVacMore6Mos          399 non-null    float64
 67  MedYrHousBuilt          399 non-null    int64
 68  PctHousNoPhone          399 non-null    float64
 69  PctWOFullPlumb          399 non-null    float64
 70  OwnOccLowQuart          399 non-null    int64
 71  OwnOccMedVal            399 non-null    int64
 72  OwnOccHiQuart           399 non-null    int64
 73  OwnOccQrange            399 non-null    int64
 74  RentLowQ                399 non-null    int64
 75  RentMedian              399 non-null    int64
 76  RentHighQ               399 non-null    int64
 77  RentQrange              399 non-null    int64
 78  MedRent                 399 non-null    int64
 79  MedRentPctHousInc       399 non-null    float64
 80  MedOwnCostPctInc        399 non-null    float64
 81  MedOwnCostPctIncNoMtg   399 non-null    float64
 82  NumInShelters           399 non-null    int64
 83  NumStreet               399 non-null    int64
 84  PctForeignBorn          399 non-null    float64
 85  PctBornSameState        399 non-null    float64
 86  PctSameHouse85          399 non-null    float64
 87  PctSameCity85           399 non-null    float64
 88  PctSameState85          399 non-null    float64
dtypes: float64(66), int64(23)
memory usage: 277.6 KB
```

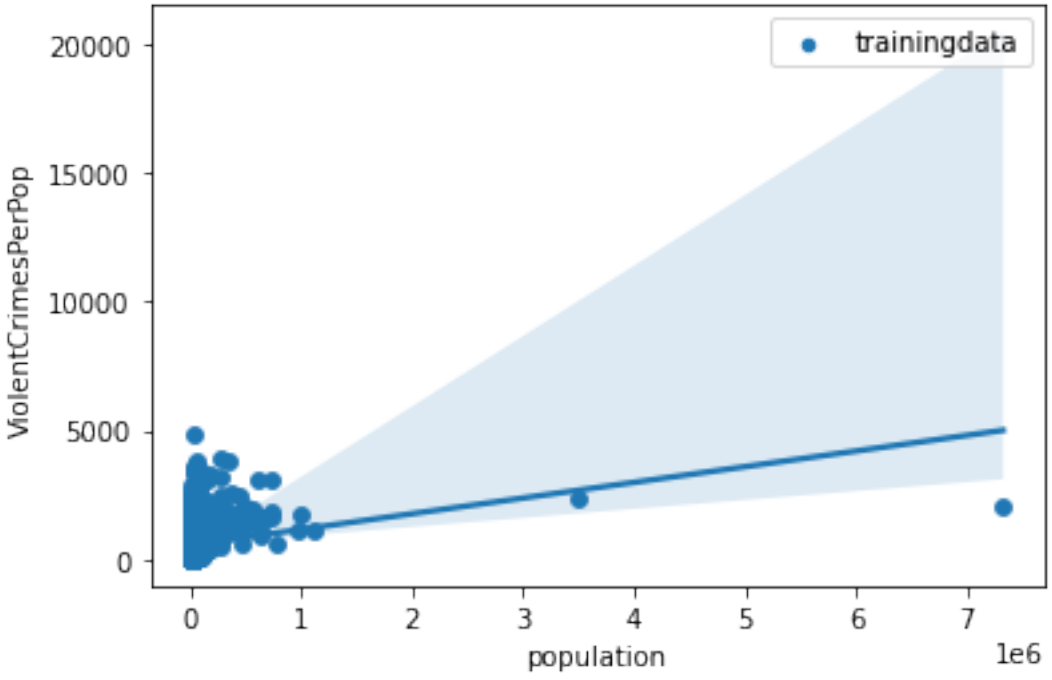
[ ]:
```
Xtest=testdata.drop(["ID"],axis=1)
ytest=rl.predict(Xtest)
```

[ ]:
```
import seaborn as sb
ytrain=rl.predict(X)
print("the mean squared error of the dataset is:␣
 ↪",mean_squared_error(y,ytrain,squared=False))
testdata["crimeperpop"]=ytest
testdata.
 ↪plot(y="crimeperpop",x="population",kind="scatter",label='testdata',title="relation␣
 ↪between population and violent crimes")
sb.regplot(x = "population",y = "crimeperpop", data = testdata)
traindata.
 ↪plot(y="ViolentCrimesPerPop",x="population",kind="scatter",label='trainingdata')
sb.regplot(x = "population",y = "ViolentCrimesPerPop", data = traindata)
```

```
the mean squared error of the dataset is:  354.6109622573772
```

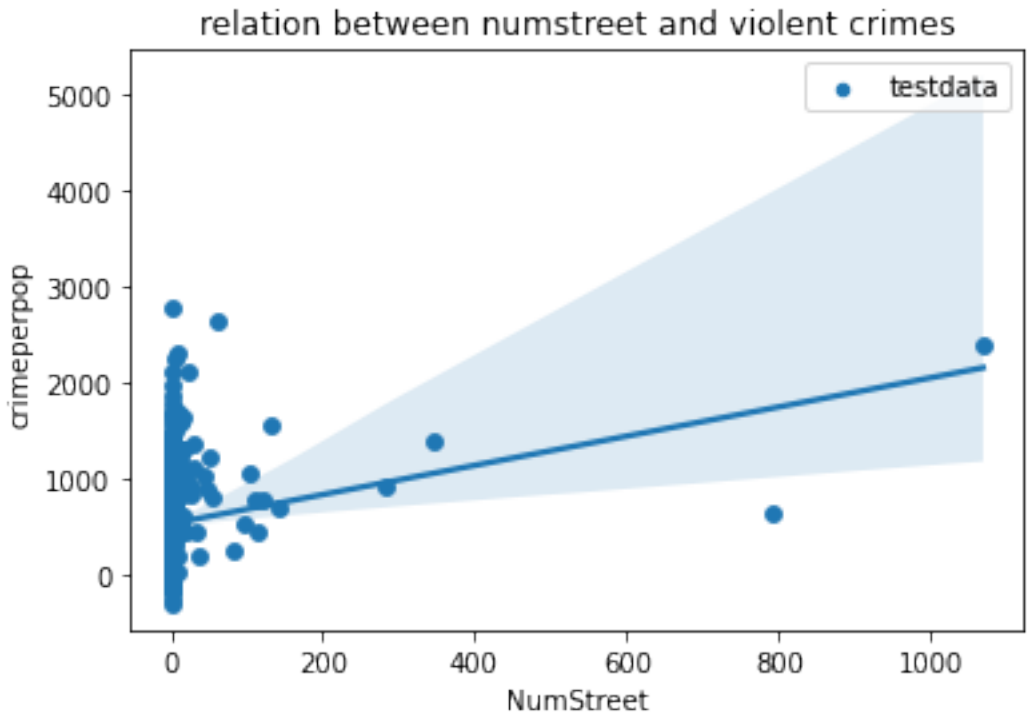
[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd3f1ddb110>
```

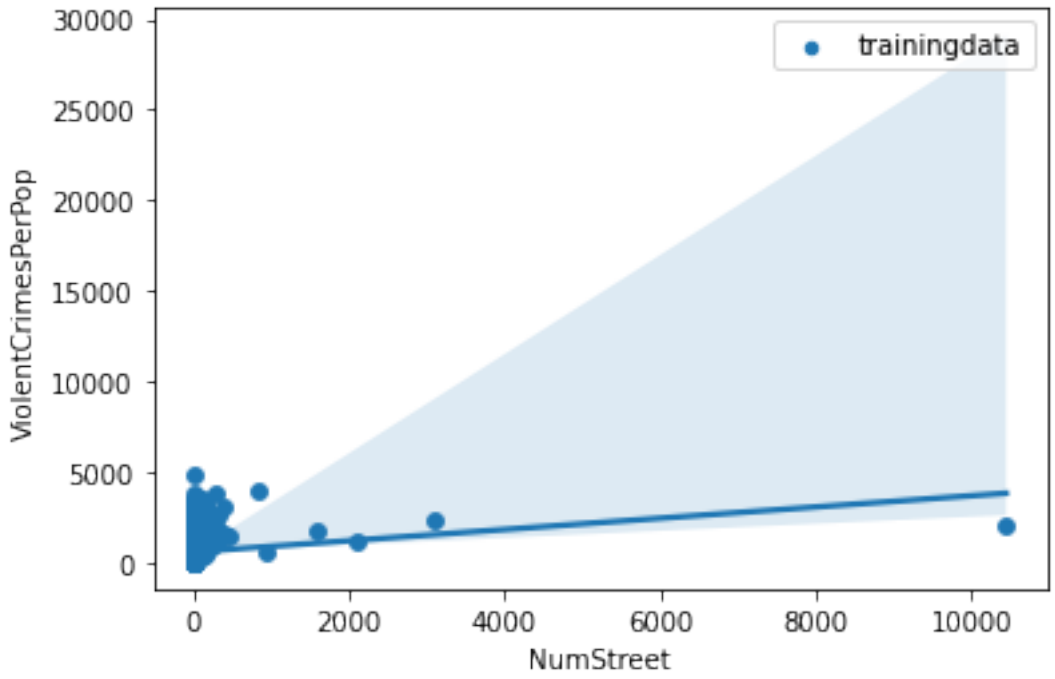

relation between population and violent crimes
testdata
crimeperpop
5000
4000
3000
2000
1000
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
population
1e6


trainingdata
ViolentCrimesPerPop
20000
15000
10000
5000
0
0
1
2
3
4
5
6
7
population
1e6

[ ]:
```
testdata["crimeperpop"]=ytest

testdata.
 ↪plot(y="crimeperpop",x="NumStreet",kind="scatter",label='testdata',title="relation␣
 ↪between numstreet and violent crimes")
sb.regplot(x = "NumStreet",y = "crimeperpop", data = testdata)
traindata.
 ↪plot(y="ViolentCrimesPerPop",x="NumStreet",kind="scatter",label='trainingdata')
sb.regplot(x = "NumStreet",y = "ViolentCrimesPerPop", data = traindata)
```

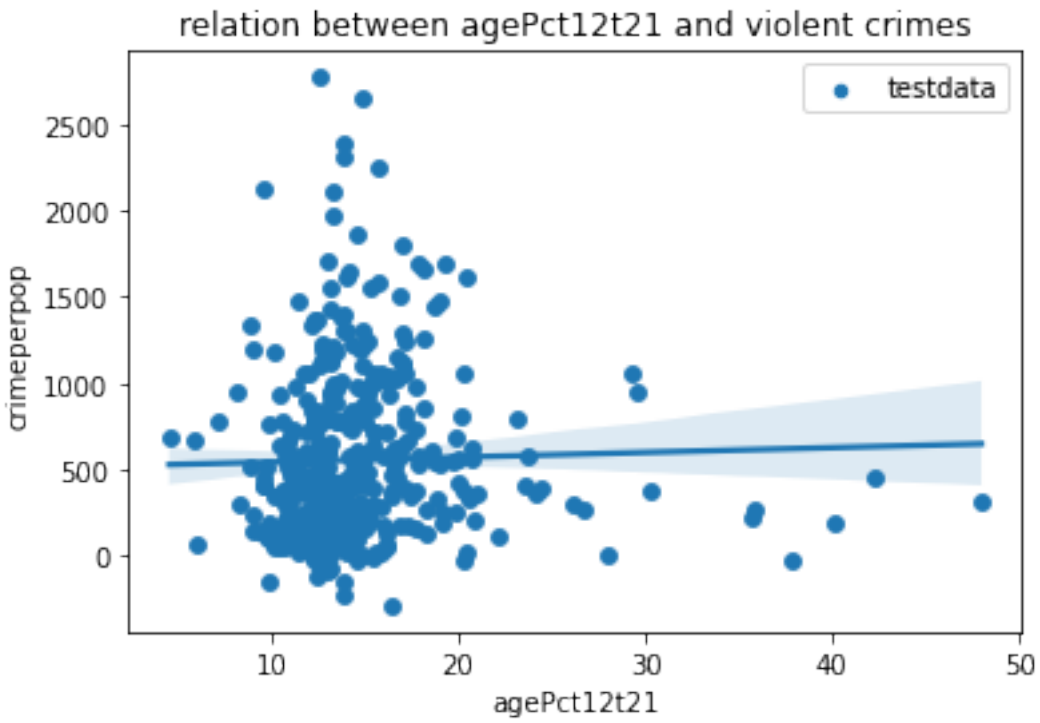
[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd3f3117910>
```

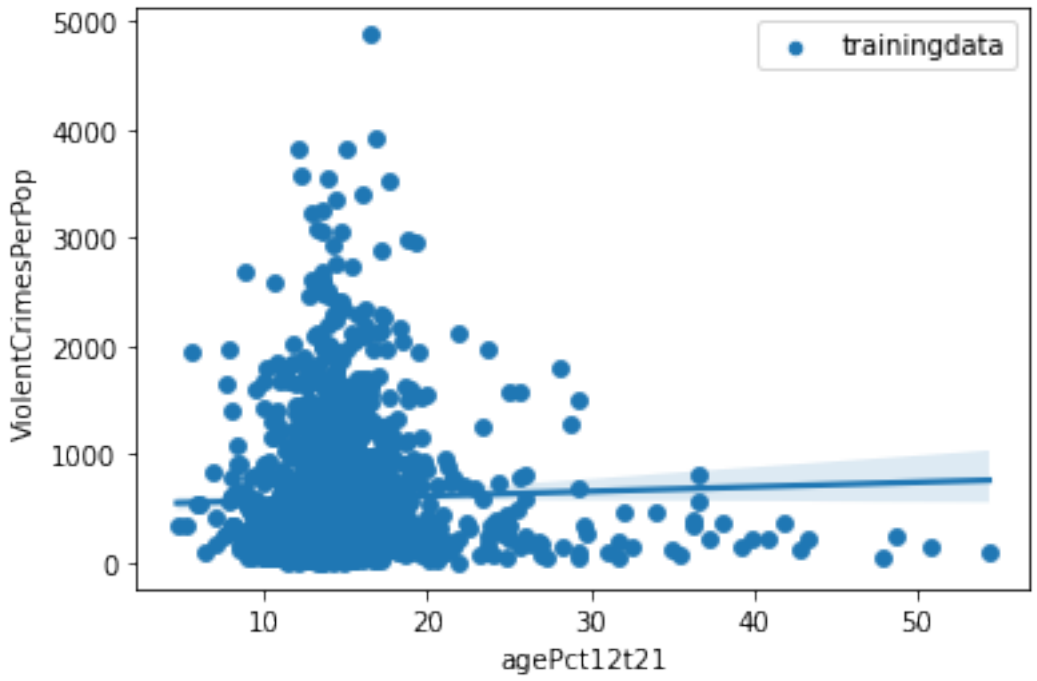




[ ]:
```
testdata["crimeperpop"]=ytest

testdata.
 ↪plot(y="crimeperpop",x="agePct12t21",kind="scatter",label='testdata',title="relation␣
 ↪between agePct12t21 and violent crimes")
sb.regplot(x = "agePct12t21",y = "crimeperpop",data = testdata)
traindata.
 ↪plot(y="ViolentCrimesPerPop",x="agePct12t21",kind="scatter",label='trainingdata')
sb.regplot(x = "agePct12t21",y = "ViolentCrimesPerPop", data = traindata)
```

[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd3f1e68ad0>
```



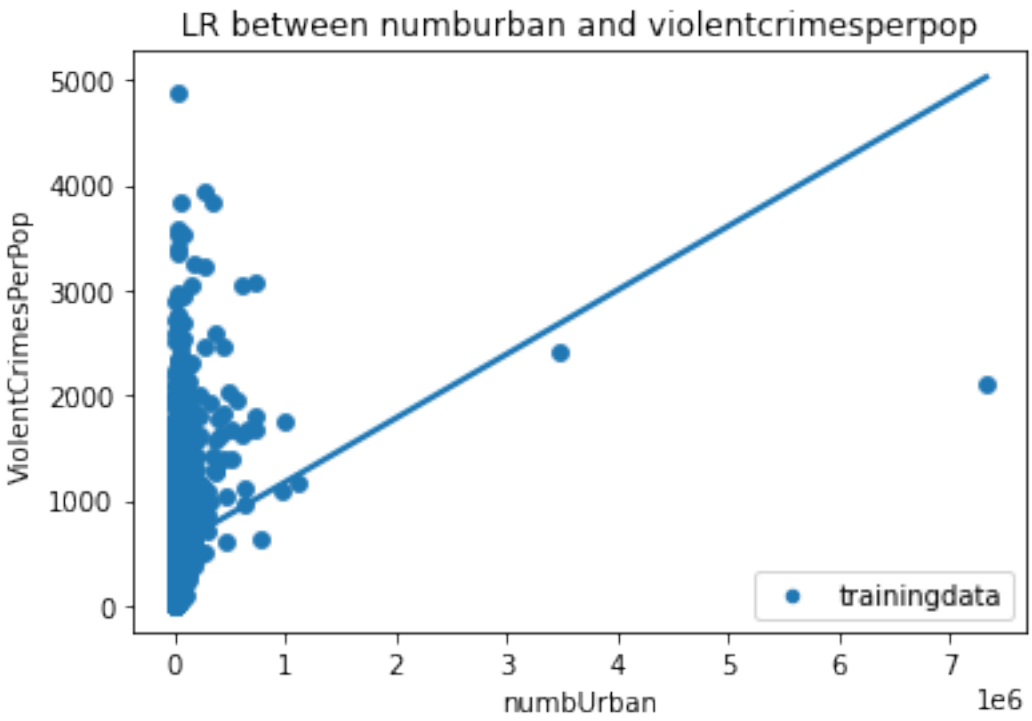


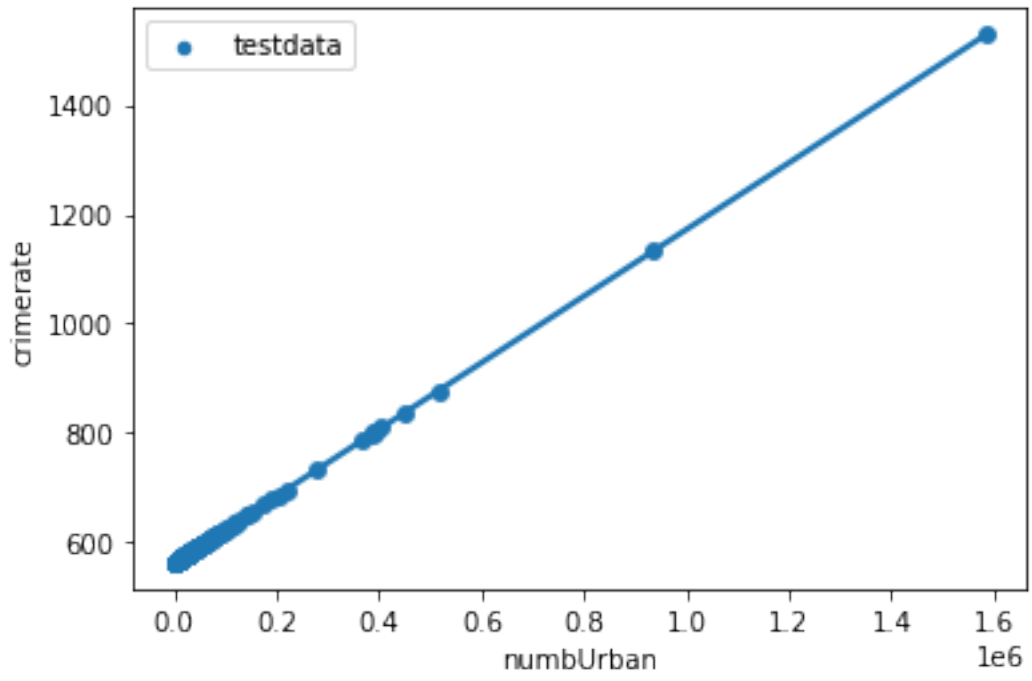

```
[ ]: print('the score of the training dataset:',rl.score(X,y))
```

```
the score of the training dataset: 0.6647700427492149
```

[ ]:
```
traindata=pd.read_csv("/content/crime_train.csv")
x=traindata["numbUrban"]
ytrain=traindata["ViolentCrimesPerPop"]
rl.fit(np.array(x).reshape(-1, 1),y)
print("the coefficient for the LR between numurban and violentcrimesperpop is:␣
 ↪",rl.coef_[0])

traindata.
 ↪plot(y="ViolentCrimesPerPop",x="numbUrban",kind="scatter",label='trainingdata',title="LR␣
 ↪between numburban and violentcrimesperpop")
sb.regplot(x = "numbUrban",y = "ViolentCrimesPerPop", ci = None,data =␣
 ↪traindata)

testdata=pd.read_csv("/content/crime_test.csv")
xtest=testdata["numbUrban"]
ytest=rl.predict(np.array(xtest).reshape(-1, 1))
testdata["crimerate"]=ytest
testdata.plot(y="crimerate",x="numbUrban",kind="scatter",label='testdata')
sb.regplot(x = "numbUrban",y = "crimerate", ci = None,data = testdata)

ypred=rl.predict(np.array(x).reshape(-1, 1))
print("the mean squared error of the dataset is:␣
 ↪",mean_squared_error(ytrain,ypred,squared=False))
```

```
the coefficient for the LR between numurban and violentcrimesperpop is:
0.000609853508308507
the mean squared error of the dataset is:  597.527237506273
```


LR between numburban and violentcrimesperpop
ViolentCrimesPerPop
numbUrban
1e6
trainingdata
0
1000
2000
3000
4000
5000
0
1
2
3
4
5
6
7


testdata
crimerate
numbUrban
1e6
600
800
1000
1200
1400
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6