Ridge Regression Using UCI Dataset

In [1]:
```
import pandas as pd
import numpy as np

# Communities and Crime dataset for regression
# https://archive.ics.uci.edu/ml/datasets/Communities+and+Crime+Unnormalized

crime_datafile = 'E:\Protfolio\Ridge Regression Using UCI Dataset/CommViolPredUnnormalizedData.txt'
crime = pd.read_table(crime_datafile, sep=',', na_values='?')
crime.head()
```

Out[1]:

	communityname	state	countyCode	communityCode	fold	population	householdsize	racepctblack	racePctWhite	racePctAsian
0	BerkeleyHeightstownship	NJ	39.0	5320.0	1	11980	3.10	1.37	91.78	6.50
1	Marpletownship	PA	45.0	47616.0	1	23123	2.82	0.80	95.57	3.44
2	Tigardcity	OR	NaN	NaN	1	29344	2.43	0.74	94.33	3.43
3	Gloversvillecity	NY	35.0	29443.0	1	16656	2.40	1.70	97.35	0.50
4	Bemidjicity	MN	7.0	5068.0	1	11245	2.76	0.53	89.16	1.17

5 rows × 147 columns

In [2]:
```
# remove features with poor coverage or lower relevance, and keep ViolentCrimesPerPop target column
columns_to_keep = ([5, 6] + list(range(11,26))+ list(range(32, 103)) + [145])
crime = (crime[crime.columns[columns_to_keep]].dropna().reset_index(drop=True))
crime.head()
```

Out[2]:

	population	householdsize	agePct12t21	agePct12t29	agePct16t24	agePct65up	numbUrban	pctUrban	medIncome	pctWWage	...	Me
0	11980	3.10	12.47	21.44	10.93	11.33	11980	100.0	75122	89.24	...	
1	23123	2.82	11.01	21.30	10.48	17.18	23123	100.0	47917	78.99	...	
2	29344	2.43	11.36	25.88	11.01	10.28	29344	100.0	35669	82.00	...	
3	16656	2.40	12.55	25.20	12.19	17.57	0	0.0	20580	68.15	...	
4	140494	2.45	18.09	32.89	20.04	13.26	140494	100.0	21577	75.78	...	

5 rows × 89 columns

In [3]:
```
#splitting features
X=crime.iloc[:,0:88]
X
```

Out[3]:

	population	householdsize	agePct12t21	agePct12t29	agePct16t24	agePct65up	numbUrban	pctUrban	medIncome	pctWWage	...
0	11980	3.10	12.47	21.44	10.93	11.33	11980	100.00	75122	89.24	...
1	23123	2.82	11.01	21.30	10.48	17.18	23123	100.00	47917	78.99	...
2	29344	2.43	11.36	25.88	11.01	10.28	29344	100.00	35669	82.00	...
3	16656	2.40	12.55	25.20	12.19	17.57	0	0.00	20580	68.15	...
4	140494	2.45	18.09	32.89	20.04	13.26	140494	100.00	21577	75.78	...
...	...	...	...	...	...	...	...	...	...	...	...
1989	56216	3.07	15.46	30.16	14.34	8.08	56216	100.00	24727	75.05	...
1990	12251	2.68	17.36	31.23	16.97	12.57	12251	100.00	20321	75.06	...
1991	32824	2.46	11.81	20.96	9.53	20.73	32824	100.00	27182	59.79	...
1992	13547	2.89	17.16	30.01	14.73	10.42	0	0.00	19899	71.67	...
1993	28898	2.61	12.99	25.21	11.63	12.12	28664	99.19	23287	68.89	...

1994 rows × 88 columns

In [4]:
```
#splitting level
y=crime['ViolentCrimesPerPop']
y
```

Out[4]:
```
0        41.02
1       127.56
2       218.59
3       306.64
4       442.95
         ...  
1989    545.75
1990    124.10
1991    353.83
1992    691.17
1993    918.89
Name: ViolentCrimesPerPop, Length: 1994, dtype: float64
```

In [5]:
```
#plitting train and test set
from sklearn.linear_model import Ridge
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,y,random_state=0)
```

In [6]:
```
#fit rdge regreesion on train set
linridge = Ridge(alpha = 20.0)
linridge.fit(X_train, y_train)
```

Out[6]:
```
Ridge(alpha=20.0)
```

In [7]:
```
print('Crime dataset')
print('ridge regression linear model intercept: {}'
     .format(linridge.intercept_))
print('ridge regression linear model coeff:\n{}'
     .format(linridge.coef_))
print('R-squared score (training): {:.3f}'
     .format(linridge.score(X_train, y_train)))
print('R-squared score (test): {:.3f}'
     .format(linridge.score(X_test, y_test)))
print('Number of non-zero features: {}'
     .format(np.sum(linridge.coef_ != 0)))
```

```
Crime dataset
ridge regression linear model intercept: -3352.4230358457785
ridge regression linear model coeff:
[ 1.95091438e-03  2.19322667e+01  9.56286607e+00 -3.59178973e+01
  6.36465325e+00 -1.96885471e+01 -2.80715856e-03  1.66254486e+00
 -6.61426604e-03 -6.95450680e+00  1.71944731e+01 -5.62819154e+00
  8.83525114e+00  6.79085746e-01 -7.33614221e+00  6.70389803e-03
  9.78505502e-04  5.01202169e-03 -4.89870524e+00 -1.79270062e+01
  9.17572382e+00 -1.24454193e+00  1.21845360e+00  1.03233089e+01
 -3.78037278e+00 -3.73428973e+00  4.74595305e+00  8.42696855e+00
  3.09250005e+01  1.18644167e+01 -2.05183675e+00 -3.82210450e+01
  1.85081589e+01  1.52510829e+00 -2.20086608e+01  2.46283912e+00
  3.29328703e-01  4.02228467e+00 -1.12903533e+01 -4.69567413e-03
  4.27046505e+01 -1.22507167e-03  1.40795790e+00  9.35041855e-01
 -3.00464253e+00  1.12390514e+00 -1.82487653e+01 -1.54653407e+01
  2.41917002e+01 -1.32497562e+01 -4.20113118e-01 -3.59710660e+01
  1.29786751e+01 -2.80765995e+01  4.38513476e+01  3.86590044e+01
 -6.46024046e+01 -1.63714023e+01  2.90397330e+01  4.15472907e+00
  5.34033563e+01  1.98773191e-02 -5.47413979e-01  1.23883518e+01
  1.03526583e+01 -1.57238894e+00  3.15887097e+00  8.77757987e+00
 -2.94724962e+01 -2.33551881e-04  3.13528914e-04 -4.13071930e-04
 -1.80407541e-04 -5.74054527e-01 -5.17742507e-01 -4.20670930e-01
  1.53383594e-01  1.32725423e+00  3.84863158e+00  3.03024594e+00
 -3.77692644e+01  1.37933464e-01  3.07676522e-01  1.57128807e+01
  3.31418306e-01  3.35994414e+00  1.61265911e-01 -2.67619878e+00]
R-squared score (training): 0.671
R-squared score (test): 0.494
Number of non-zero features: 88
```

Ridge Regression with feature normalization

Here using MinMaxScaler for the preprocessing

In [12]:
```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
#normalize the train and test set
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.fit_transform(X_test)

#fit ridge model in scaled train set
linridge2 = Ridge(alpha = 20.0).fit(X_train_scaled,y_train)
```

In [13]:
```
print('Crime dataset')
print('ridge regression linear model intercept: {}'
     .format(linridge2.intercept_))
print('ridge regression linear model coeff:\n{}'
     .format(linridge2.coef_))
print('R-squared score (training): {:.3f}'
     .format(linridge2.score(X_train_scaled, y_train)))
print('R-squared score (test): {:.3f}'
     .format(linridge2.score(X_test_scaled, y_test)))
print('Number of non-zero features: {}'
     .format(np.sum(linridge2.coef_ != 0)))
```

```
Crime dataset
ridge regression linear model intercept: 933.390638504413
ridge regression linear model coeff:
[  88.68827454   16.48947987  -50.30285445  -82.90507574  -65.89507244
   -2.27674244   87.74108514  150.94862182   18.8802613   -31.05554992
  -43.13536109 -189.44266328   -4.52658099  107.97866804  -76.53358414
    2.86032762   34.95230077   90.13523036   52.46428263  -62.10898424
  115.01780357    2.66942023    6.94331369   -5.66646499 -101.55269144
  -36.9087526    -8.7053343    29.11999068  171.25963057   99.36919476
   75.06611841  123.63522539   95.24316483 -330.61044265 -442.30179004
 -284.49744001 -258.37150609   17.66431072 -101.70717151  110.64762887
  523.13611718   24.8208959     4.86533322  -30.46775619   -3.51753937
   50.57947231   10.84840601   18.27680946   44.11189865   58.33588176
   67.08698975  -57.93524659  116.1446052    53.81163718   49.01607711
   -7.62262031   55.14288543  -52.08878272  123.39291017   77.12562171
   45.49795317  184.91229771  -91.35721203    1.07975971  234.09267451
   10.3887921    94.7171829   167.91856631  -25.14025088   -1.18242839
   14.60362467   36.77122659   53.19878339  -78.86365997   -5.89858411
   26.04790298  115.1534917    68.74143311   68.28588166   16.5260514
  -97.90513652  205.20448474   75.97304123   61.3791085   -79.83157049
   67.26700741   95.67094538  -11.88380569]
R-squared score (training): 0.615
R-squared score (test): 0.620
Number of non-zero features: 88
```

In [ ]: