In [4]:
```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.datasets import load_boston
```

In [5]:
```
boston = load_boston()
print(boston.DESCR)
```

```
.. _boston_dataset:

Boston house prices dataset
---------------------------

**Data Set Characteristics:**

    :Number of Instances: 506

    :Number of Attributes: 13 numeric/categorical predictive. Median Value (attribute 14) is
usually the target.

    :Attribute Information (in order):
        - CRIM     per capita crime rate by town
        - ZN       proportion of residential land zoned for lots over 25,000 sq.ft.
        - INDUS    proportion of non-retail business acres per town
        - CHAS     Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
        - NOX      nitric oxides concentration (parts per 10 million)
        - RM       average number of rooms per dwelling
        - AGE      proportion of owner-occupied units built prior to 1940
        - DIS      weighted distances to five Boston employment centres
        - RAD      index of accessibility to radial highways
        - TAX      full-value property-tax rate per $10,000
        - PTRATIO  pupil-teacher ratio by town
        - B        1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
        - LSTAT    % lower status of the population
        - MEDV     Median value of owner-occupied homes in $1000's

    :Missing Attribute Values: None

    :Creator: Harrison, D. and Rubinfeld, D.L.

This is a copy of UCI ML housing dataset.
https://archive.ics.uci.edu/ml/machine-learning-databases/housing/


This dataset was taken from the StatLib library which is maintained at Carnegie Mellon Univer
sity.

The Boston house-price data of Harrison, D. and Rubinfeld, D.L. 'Hedonic
prices and the demand for clean air', J. Environ. Economics & Management,
vol.5, 81-102, 1978.   Used in Belsley, Kuh & Welsch, 'Regression diagnostics
...', Wiley, 1980.   N.B. Various transformations are used in the table on
pages 244-261 of the latter.

The Boston house-price data has been used in many machine learning papers that address regres
sion
problems.

.. topic:: References

   - Belsley, Kuh & Welsch, 'Regression diagnostics: Identifying Influential Data and Sources
of Collinearity', Wiley, 1980. 244-261.
   - Quinlan,R. (1993). Combining Instance-Based and Model-Based Learning. In Proceedings on
the Tenth International Conference of Machine Learning, 236-243, University of Massachusetts,
Amherst. Morgan Kaufmann.
```

In [6]:
```
dataset = boston.data
```

In [7]:
```
for name,index in enumerate(boston.feature_names):
    print(index,name)
```

```
CRIM 0
ZN 1
INDUS 2
CHAS 3
NOX 4
RM 5
AGE 6
DIS 7
RAD 8
TAX 9
PTRATIO 10
B 11
LSTAT 12
```

In [10]:
```
data = dataset[:,12].reshape(-1,1)
```

In [12]:
```
np.shape(dataset)
```

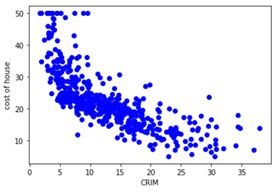
Out[12]:
```
(506, 13)
```

In [13]:
```
target = boston.target.reshape(-1,1)
```

In [14]:
```
np.shape(target)
```

Out[14]:
```
(506, 1)
```

In [19]:
```
%matplotlib inline
plt.scatter(data,target,color='blue')
plt.xlabel('CRIM')
plt.ylabel('cost of house')
plt.show()
```

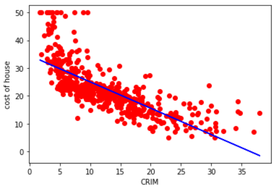


In [49]:
```
from sklearn.linear_model import LinearRegression

reg =  LinearRegression()

reg.fit(data, target)
```

Out[49]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [50]:
```
pred =  reg.predict(data)
```

In [51]:
```
%matplotlib inline
plt.scatter(data,target,color='red')
plt.plot(data,pred,color='blue')
plt.xlabel('CRIM')
plt.ylabel('cost of house')
plt.show()
```



In [61]:
```
from sklearn.preprocessing import PolynomialFeatures

from sklearn.pipeline import make_pipeline
```

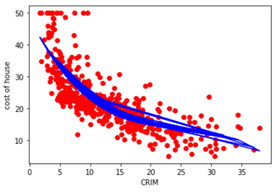
In [96]:
```
model = make_pipeline(PolynomialFeatures(3),reg)
```

In [97]:
```
model.fit(data,target)
```

Out[97]:
```
Pipeline(memory=None,
         steps=[('polynomialfeatures',
                 PolynomialFeatures(degree=3, include_bias=True,
                                    interaction_only=False, order='C')),
                ('linearregression',
                 LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
                                  normalize=False))],
         verbose=False)
```

In [98]:
```
pred = model.predict(data)
```

In [99]:
```
%matplotlib inline
plt.scatter(data,target,color='red')
plt.plot(data,pred,color='blue')
plt.xlabel('CRIM')
plt.ylabel('cost of house')
plt.show()
```



In [100]:
```
from sklearn.metrics import r2_score
```

In [101]:
```
r2_score(pred,target)
```

Out[101]:
```
0.4798911810275662
```