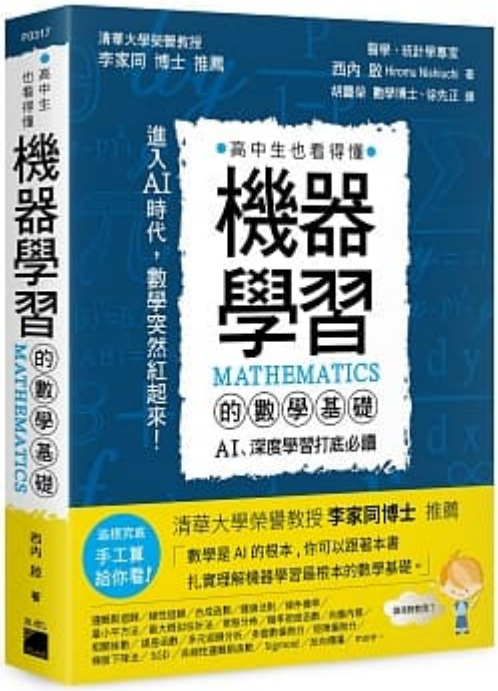
用sklearn做波士頓房價線性回歸

不錯的參考資料

政治大學線上課程<成為python數據分析達人的第一門課>

<機器學習的數學>



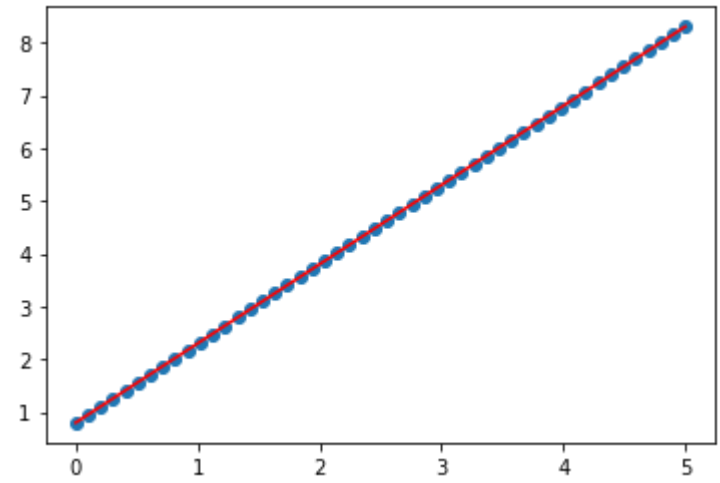
1. 前置練習--畫直線

先自己創建一組線性二維數據，且自己設定y=mx+b的m與b
再用matplotlib做直線

```
import  numpy  as  np       #引入numpy資源庫
import  matplotlib.pyplot  as  plt    #引入matplotlib資源庫
x=np.linspace(0,5,50)     #x軸在0-50之間產生50個點
y=1.5*x+0.8     #斜率設1.5
```

```
plt.scatter(x,y)  #  下圖藍色點狀圖
plt.plot(x,y,'r')  #  直線形式(紅色)
```

```
[<matplotlib.lines.Line2D at 0x7f76c2dbb5d0>]
```

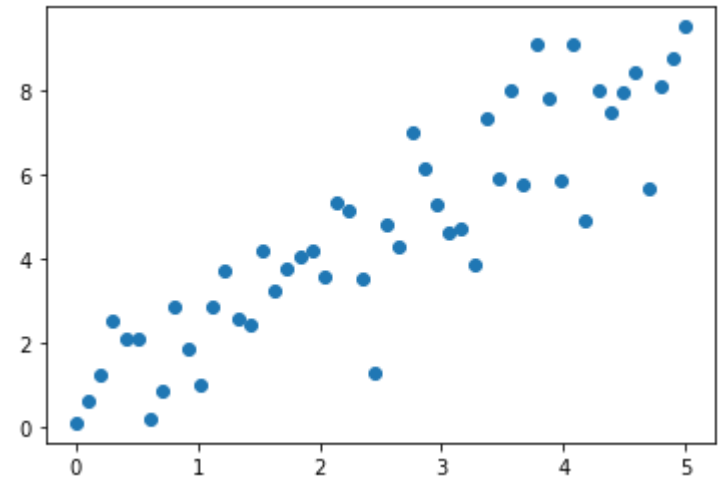


2. 前置練習--加入噪點

因為真實世界的數據不會如上圖，加入噪點產生隨機數據

```
y1=1.5*x+0.8+np.random.randn(50)  #  因為x軸有50個點，每個點都要加上偏移，所以加上50個隨機偏移
plt.scatter(x,y1)
```

```
<matplotlib.collections.PathCollection at 0x7f76bd12d8d0>
```



按兩下 (或按 Enter 鍵) 即可編輯

3. Sklearn 把X軸數據從50x1改成1x50

```
from  sklearn.linear_model  import  LinearRegression
regr  =  LinearRegression()
X=  x.reshape(50,1)
```

```
x
```

```
array([0.        , 0.10204082, 0.20408163, 0.30612245, 0.40816327,
       0.51020408, 0.6122449 , 0.71428571, 0.81632653, 0.91836735,
       1.02040816, 1.12244898, 1.2244898 , 1.32653061, 1.42857143,
       1.53061224, 1.63265306, 1.73469388, 1.83673469, 1.93877551,
       2.04081633, 2.14285714, 2.24489796, 2.34693878, 2.44897959,
       2.55102041, 2.65306122, 2.75510204, 2.85714286, 2.95918367,
       3.06122449, 3.16326531, 3.26530612, 3.36734694, 3.46938776,
       3.57142857, 3.67346939, 3.7755102 , 3.87755102, 3.97959184,
       4.08163265, 4.18367347, 4.28571429, 4.3877551 , 4.48979592,
       4.59183673, 4.69387755, 4.79591837, 4.89795918, 5.        ])
```

```
X
```

```
array([[0.        ],
       [0.10204082],
       [0.20408163],
       [0.30612245],
       [0.40816327],
       [0.51020408],
       [0.6122449 ],
       [0.71428571],
       [0.81632653],
       [0.91836735],
       [1.02040816],
       [1.12244898],
       [1.2244898 ],
       [1.32653061],
       [1.42857143],
       [1.53061224],
       [1.63265306],
       [1.73469388],
       [1.83673469],
       [1.93877551],
       [2.04081633],
       [2.14285714],
       [2.24489796],
       [2.34693878],
       [2.44897959],
       [2.55102041],
       [2.65306122],
       [2.75510204],
       [2.85714286],
       [2.95918367],
       [3.06122449],
       [3.16326531],
       [3.26530612],
       [3.36734694],
       [3.46938776],
       [3.57142857],
       [3.67346939],
       [3.7755102 ],
       [3.87755102],
       [3.97959184],
       [4.08163265],
       [4.18367347],
       [4.28571429],
```

```
       [4.3877551 ],
       [4.48979592],
       [4.59183673],
       [4.69387755],
       [4.79591837],
       [4.89795918],
       [5.        ]])
```

4. 利用fit功能進行線性回歸

```
regr.fit(X,y1)
#regr為前面步驟調用的線性回歸方法，就有點像自己給有線性回歸功能的寶可夢自己取名字
#  X為輸入資料，y為正確答案
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

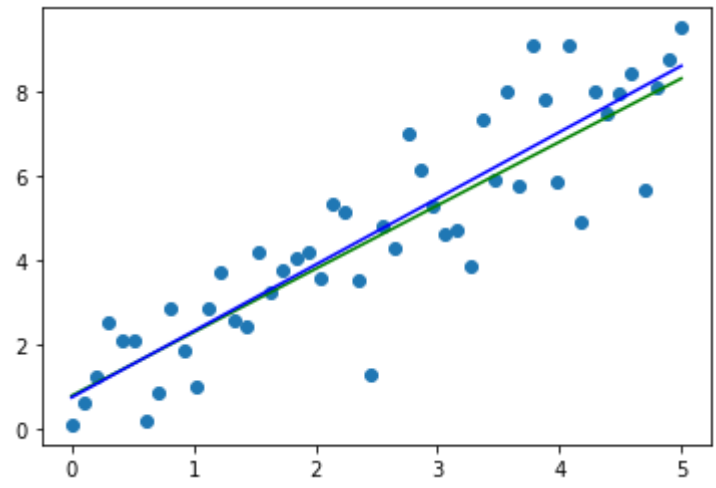
5. 利用建立好的regr開始進行預測

```
Y  =  regr.predict(X)
#  f(x)  =  y1  預設的輸入x和正確答案y
#  x  進行矩陣行列轉換變成X
#  線性回歸尋找X和y1關係,把建立關係式模型寫入regr
#  利用建立好的regr模型，預測X輸入後的結果，並把結果命為Y，regr(X)=Y
```

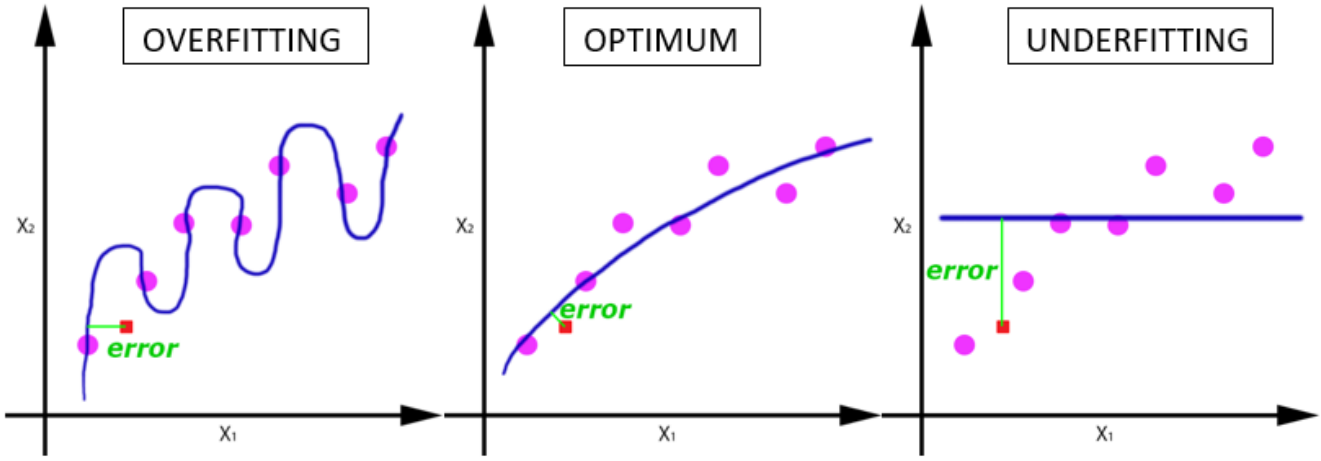
6. 對比正確答案和線性預測結果

```
plt.scatter(x,y1)#  正確答案
plt.plot(x,1.5*x+0.8,'green')#  綠色線為預設線性方程
plt.plot(x,Y,'blue')  #藍色線為線性回歸結果
```

```
[<matplotlib.lines.Line2D at 0x7f76abc42190>]
```



7. 避免overfit 進行數據分割練習



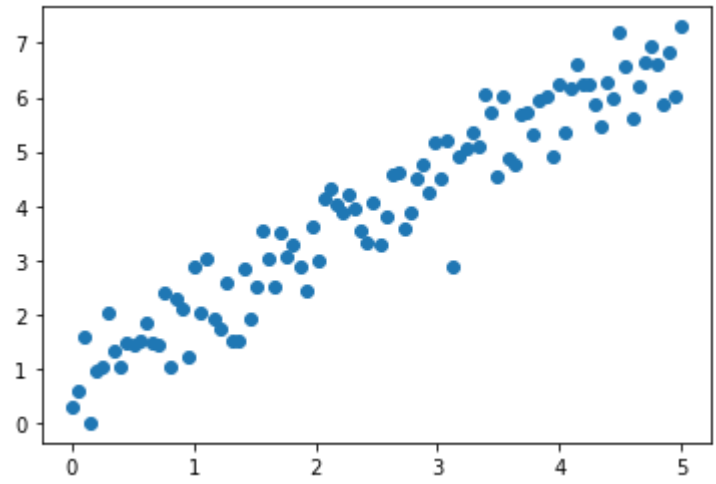
圖片來源:Sagar Sharma / Towards Data Science

overfit就像考試作弊背答案沒有真正理解，
雖然劃出來預測線涵蓋所有數據，但一旦脫離預測的數據
輸入其他數據進行預測時跑出來的就會失真
為了避免被答案，就要分割數據，把一部分的數據拿來考建立好的模型

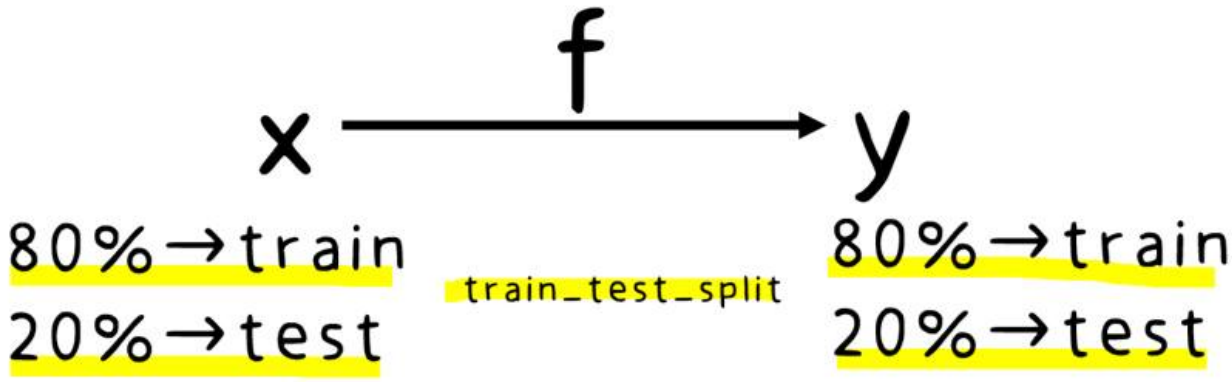
8. 數據分割練習

```
x=np.linspace(0,5,100)
y=1.2*x+0.9+0.5*np.random.randn(100)
plt.scatter(x,y)
#  先產生100筆含有噪點的數據
```

```
<matplotlib.collections.PathCollection at 0x7f76ab8f3a50>
```



```
from  sklearn.model_selection  import  train_test_split
#  引入數據分割模組
```




```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.2,random_state=87)
#rain_test_split預設將資料分為 x_train訓練用, x_test測試用, y_train訓練用, y_test測試用
#,0.2表示20%分給test，random state為讓隨機狀態產生標籤，指定隨便一個數字，避免每次跑出來不一樣
```

```
len(x_train) # 確認訓練數據分得80筆
```

```
80
```

```
len(x_test) #確認測試數據分得20筆
```

```
20
```

```
x_train = x_train.reshape(80,1)
# 把80筆變成一維陣列 80列*1行
```

```
x_test = x_test.reshape(20,1)
#把20筆變成一維陣列 20列*1行
```

```
x_test
#確認一下
```

```
array([[0.80808081],
       [4.09090909],
       [4.29292929],
       [1.61616162],
       [1.96969697],
       [1.26262626],
       [1.31313131],
       [1.46464646],
       [3.28282828],
       [0.35353535],
       [4.24242424],
       [5.        ],
       [2.92929293],
       [3.53535354],
```

```
       [4.64646465],
       [3.73737374],
       [0.15151515],
       [0.        ],
       [2.22222222],
       [1.66666667]])
```

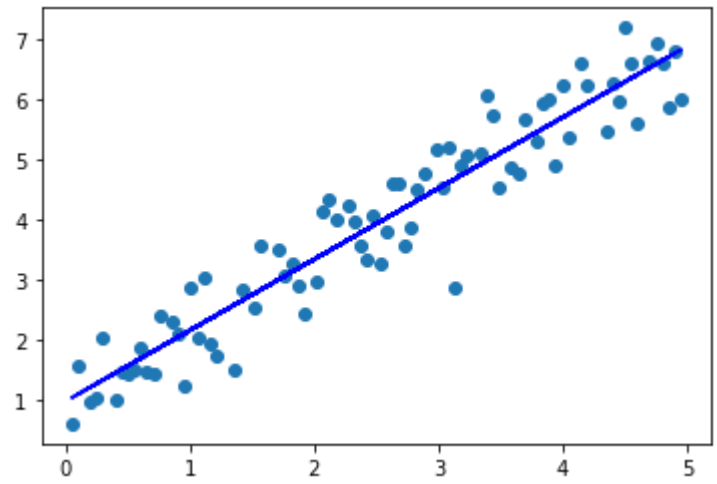
```
regr  =  LinearRegression()
regr.fit(x_train,y_train)
#  一樣創建線性回歸模組，並訓練該80筆數據
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

9. 查看regr線性回歸 預測結果(藍線)

```
plt.scatter(x_train,y_train)
plt.plot(x_train,regr.predict(x_train),'blue')
```

```
[<matplotlib.lines.Line2D at 0x7f76ab884ed0>]
```



10. x_test(考題)輸入建立好的regr模型

```
plt.scatter(x_test,y_test)  #  問題與正確答案(藍點)
plt.plot(x_test,  regr.predict(x_test),'r')  #  預測出來的回歸線(考試結果)
plt.plot(x,1.2*x+0.9,'green')  #  正確答案回歸線
```

```
[<matplotlib.lines.Line2D at 0x7f76ab779b10>]
```

brief summary

經過數據分割後，跑出來的預測回歸線幾乎貼合正確答案

實際數據_波士頓房價分析

1. 使用sklearn進行線性回歸，並調用內含的房價數據庫

```
from  sklearn.linear_model  import  LinearRegression
from  sklearn.model_selection  import  train_test_split
from  sklearn.datasets  import  load_boston
```

按兩下 (或按 Enter 鍵) 即可編輯

```
boston  =  load_boston()
```

2. 查看數據庫內的標籤

```
boston.feature_names
```

```
array(['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD',
       'TAX', 'PTRATIO', 'B', 'LSTAT'], dtype='<U7')
```

CRIM 城鎮人均犯罪率
ZN 住宅用地超過 25000 sq.ft. 的比例
AGE 1940年之前建成的自用房屋比例
DIS 到波士頓5個中心區域的加權距離
INDUS 城鎮非零售商用土地的比例
RAD 輻射性公路的靠近指數
TAX 每10000美元的全值財產稅率
CHAS 邊界是河流為1，否則0
NOX 一氧化氮濃度
PTRATIO 城鎮師生比例
RM 住宅平均房間數
LSTAT 人口中地位低下者的比例

3. 將要訓練的數據命為X，真實房價為Y

```
X = boston.data
Y = boston.target  #  內含正確答案
```

```
len(X)  #查看數據筆數
```

```
506
```

```
len(Y)
```

```
506
```

4. 將數據分割20%給Test

```
x_train, x_test, y_train, y_test = train_test_split(X,Y,test_size = 0.2,random_state=87)
```

5. 線性回歸待訓練數據

```
regr = LinearRegression()
regr.fit(x_train,y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

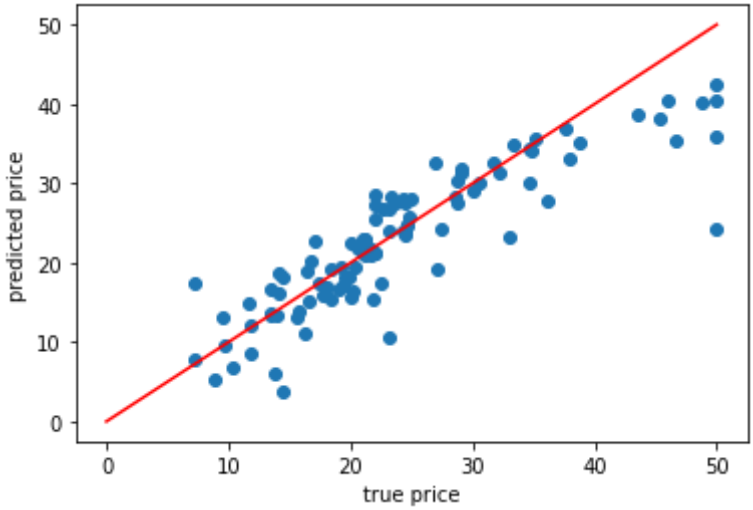
6. 訓練後將測試用x輸入regr模型，輸出predict

```
y_predict  =  regr.predict(x_test)
```

7. 若預測出來的y_predict 和真實的y_test數據極為相近 散佈圖會呈現對角線

```
#  畫紅色對角線
```

```
[<matplotlib.lines.Line2D at 0x7f769b6fa7d0>]
```



8. 列表編號技巧 enumerate

```
L  =['A','B','C','D']
list(enumerate(L))
```

```
[(0, 'A'), (1, 'B'), (2, 'C'), (3, 'D')]
```

```
for  i,s  in  enumerate(L):
    print(i+1,s)
```

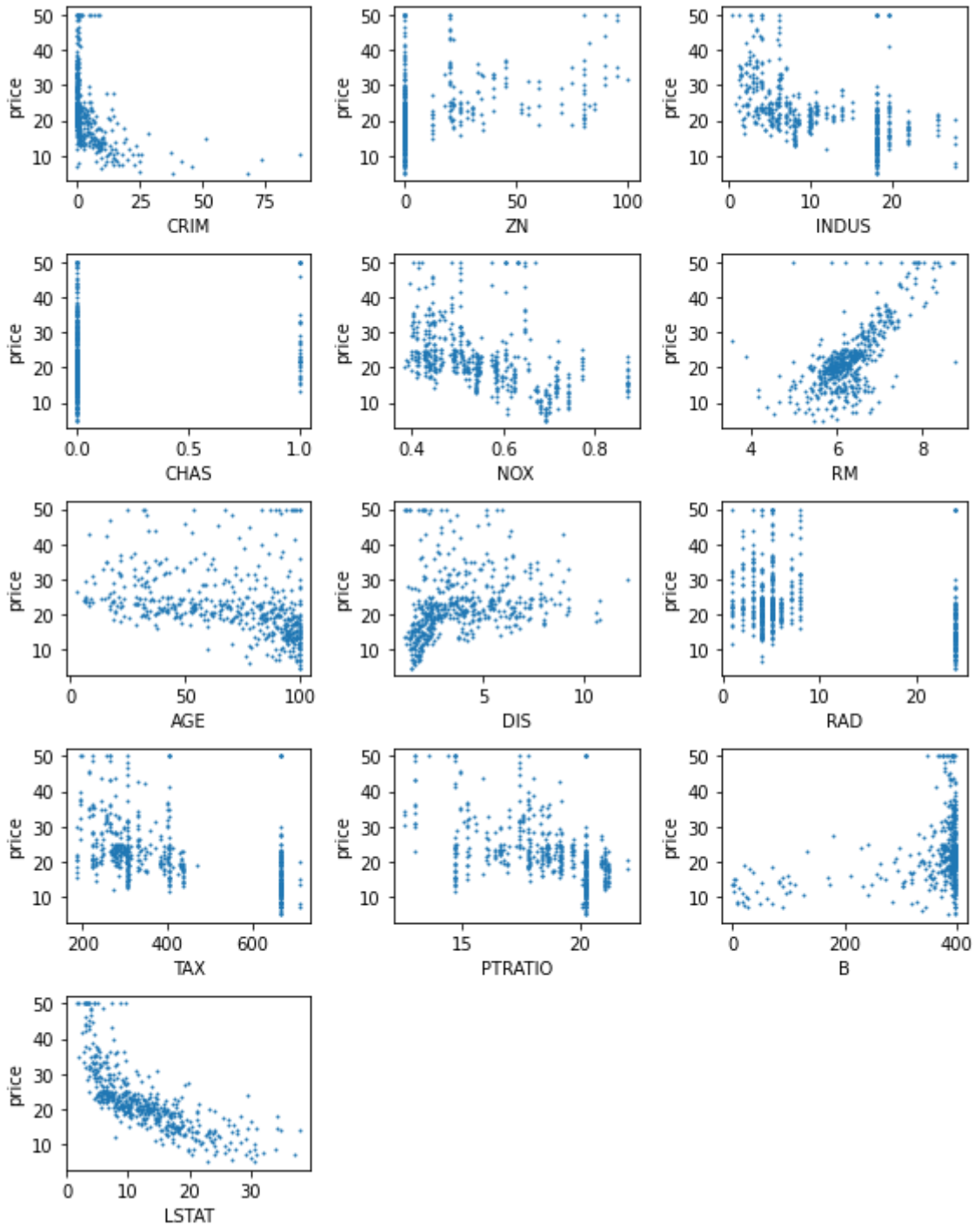
```
1 A
2 B
3 C
4 D
```

9. 畫多張圖技巧 subplot(a,b,i) 一次要畫a列b行共a*b 張圖 i 為第幾張圖

10. 利用subplot和enumberate技巧來一次對比各項特徵和房價之間的關係

```
plt.figure(figsize=(8,10))
```

```
for i,feature in enumerate(boston.feature_names):
    plt.subplot(5,3,i+1)
    plt.scatter(X[:,i],Y,s=1)
    plt.ylabel("price")
    plt.xlabel(feature)
    plt.tight_layout()
```



按兩下 (或按 Enter 鍵) 即可編輯