



國立中山大學  
National Sun Yat-sen University

Program in Interdisciplinary Studies  
Sustainable Innovative Laboratory



# Ch 11多元線性迴歸分析： scikit-learn

國立中山大學人文暨科技跨領域學士學位學程

楊政融老師

# 課程大綱

- 11-0 使用 scikit-learn 並匯入測試資料集
- 11-1 訓練並評估多元線性迴歸模型
- 11-2 評估模型的表現 (performance)
- 11-3 用真實世界的資料做迴歸分析：共享單車與天氣



# 11-0 使用 scikit-learn 並匯入測試資料集

- 前言:
- 第八章我們介紹過簡單線性迴歸，也就是以兩筆資料X和Y的關係來求出一條預測方程式。
- 但是現實生活中影響資料的因素很可能不只一個。
- 譬如第八章提到的黃金價格或許不只會受到美金匯率影響，甚至也跟歐元匯率、美國聯準會利率、石油價格、關稅甚至國際政治跟軍事情勢皆有關係。
- 所以影響變數Y的因子就會有 $X_1$ 、 $X_2$ 、 $X_3...$ 等變數，用方程式表示則為:

$$Y = A_1X_1 + A_2X_2 + A_3X_3 + \cdots + B$$



# 11-0 使用 scikit-learn 並匯入測試資料集

- 總結:
- 這樣有多個變數的模型稱之為多元迴歸(multiple)或複迴歸。
- 多元線性迴歸模型的原理跟簡單線性迴歸是一樣，但NumPy並沒有提供這方面功能。
- 要借助Python專為機器學習模型而設計的資料科學套件scikit-learn。



# 11-0 使用 scikit-learn 並匯入測試資料集

- 案例示範:
- 先練習scikit-learn提供的測試資料集。
- 引用在1978年Harrusin與Rubinfeld對於波士頓房價的研究，資料集分析空汙及其他居住條件對該市房價的影響。
- 此資料及共有506筆資料，每筆資料有14個欄位。
- 最後一欄位MEDV就是我們想要預測的房價，其他前面13欄位則是可能會影響房價的因子。

欄位英文名稱	意義
CRIM	城鎮人均犯罪率
ZN	住宅用地超過 25000 平方呎 (702.5 坪) 的比例
INDUS	城鎮內非零售業商業用地的比例
CHAS	土地是否鄰近查爾斯河 (在河邊 = 1, 否則 = 0)
NOX	一氧化氮濃度
RM	住宅平均房間數
AGE	1940 年前建成的自用房屋比例
DIS	與波士頓五個就業中心地區的加權距離
RAD	與重要幹道的距離指數
TAX	每 10000 美元的全值不動產稅率
PTRATIO	城鎮師生比例
B	公式為 $1000 * (B_k - 0.63)^2$ , 其中 $B_k$ 指城鎮中的黑人比例 (按：此資料集源自種族主義更明顯的時代)
LSTAT	低下階級人口的比例
MEDV	自用住宅的房價中位數, 以千美元計



# 11-0 使用 scikit-learn 並匯入測試資料集

- 取出自變數與應變數資料
- 上面前13個欄位資料就是迴歸模型中的 $X_1$ 、 $X_2$ 、 $X_3$ ...變數，又稱為自變數 (independent variable)或特徵值(feature)。
- 至於MEDV(迴歸模型中的Y，要預測的對象)則稱為應變數 (dependent variable)或依變數。
- 在預測模型中應變數又稱為目標變數(target variable)簡稱目標(target)。
- 後續我們統稱目標為目標變數，目標變數的內容則稱為目標值。





# 11-0 使用 scikit-learn 並匯入測試資料集

- 先來印出一筆波士頓房價的資料看看。

IN

```
from sklearn import datasets
```

```
data = datasets.load_boston().data ← 取出自變數欄位
```

```
target = datasets.load_boston().target ← 取出目標變數欄位
```

```
print(data[0], target[0]) ← 印出索引 0 的資料來看看
```



```
[6.320e-03 1.800e+01 2.310e+00 0.000e+00 5.380e-01 6.575e+00 6.520e+01  
4.090e+00 1.000e+00 2.960e+02 1.530e+01 3.969e+02 4.980e+00] 24.0
```



# 11-0 使用 scikit-learn 並匯入測試資料集

- 前一頁的數值結果對應到的欄位:

欄位	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE
值	0.00632	18	2.31	0	0.538	6.575	65.2
欄位	DIS	RAD	TAX	PTRATIO	B	LSTAT	<b>MEDV</b>
值	4.09	1	296	15.3	396.9	4.98	<b>24.0</b>

[6.320e-03 1.800e+01 2.310e+00 0.000e+00 5.380e-01 6.575e+00 6.520e+01  
4.090e+00 1.000e+00 2.960e+02 1.530e+01 3.969e+02 4.980e+00] 24.0





# 11-0 使用 scikit-learn 並匯入測試資料集

- 也可以將所有資料印出來。
- 可以看見scikit-learn的資料已經將自變數與目標變數分開。
- 後續會展示如何將報表中資料切開。

```
In [2]: print(data)  
        print(target)
```

target  
(目標變數)

```
[[6.3200e-03 1.8000e+01 2.3100e+00 ... 1.5300e+01 3.9690e+02 4.9800e+00]  
 [2.7310e-02 0.0000e+00 7.0700e+00 ... 1.7800e+01 3.9690e+02 9.1400e+00]  
 [2.7290e-02 0.0000e+00 7.0700e+00 ... 1.7800e+01 3.9283e+02 4.0300e+00]  
 ...  
 [6.0760e-02 0.0000e+00 1.1930e+01 ... 2.1000e+01 3.9690e+02 5.6400e+00]  
 [1.0959e-01 0.0000e+00 1.1930e+01 ... 2.1000e+01 3.9345e+02 6.4800e+00]  
 [4.7410e-02 0.0000e+00 1.1930e+01 ... 2.1000e+01 3.9690e+02 7.8800e+00]]
```

data  
(自變數)

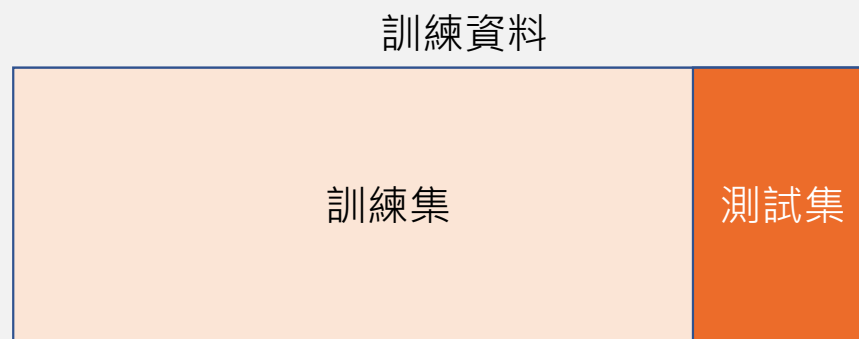
```
[24. 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 15. 18.9 21.7 20.4  
 18.2 19.9 23.1 17.5 20.2 18.2 13.6 19.6 15.2 14.5 15.6 13.9 16.6 14.8  
 18.4 21. 12.7 14.5 13.2 13.1 13.5 18.9 20. 21. 24.7 30.8 34.9 26.6  
 25.3 24.7 21.2 19.3 20. 16.6 14.4 19.4 19.7 20.5 25. 23.4 18.9 35.4  
 24.7 31.6 23.3 19.6 18.7 16. 22.2 25. 33. 23.5 19.4 22. 17.4 20.9  
 24.2 21.7 22.8 23.4 24.1 21.4 20. 20.8 21.2 20.3 28. 23.9 24.8 22.9
```

```
19.5 18.5 20.6 19. 18.7 32.7 16.5 23.9 31.2 17.5 17.2 23.1 24.5 26.6  
 22.9 24.1 18.6 30.1 18.2 20.6 17.8 21.7 22.7 22.6 25. 19.9 20.8 16.8  
 21.9 27.5 21.9 23.1 50. 50. 50. 50. 50. 13.8 13.8 15. 13.9 13.3  
 13.1 10.2 10.4 10.9 11.3 12.3 8.8 7.2 10.5 7.4 10.2 11.5 15.1 23.2  
 9.7 13.8 12.7 13.1 12.5 8.5 5. 6.3 5.6 7.2 12.1 8.3 8.5 5.  
 11.9 27.9 17.2 27.5 15. 17.2 17.9 16.3 7. 7.2 7.5 10.4 8.8 8.4  
 16.7 14.2 20.8 13.4 11.7 8.3 10.2 10.9 11. 9.5 14.5 14.1 16.1 14.3  
 11.7 13.4 9.6 8.7 8.4 12.8 10.5 17.1 18.4 15.4 10.8 11.8 14.9 12.6  
 14.1 13. 13.4 15.2 16.1 17.8 14.9 14.1 12.7 13.5 14.9 20. 16.4 17.7  
 19.5 20.2 21.4 19.9 19. 19.1 19.1 20.1 19.9 19.6 23.2 29.8 13.8 13.3  
 16.7 12. 14.6 21.4 23. 23.7 25. 21.8 20.6 21.2 19.1 20.6 15.2 7.  
 8.1 13.6 20.1 21.8 24.5 23.1 19.7 18.3 21.2 17.5 16.8 22.4 20.6 23.9  
 22. 11.9]
```



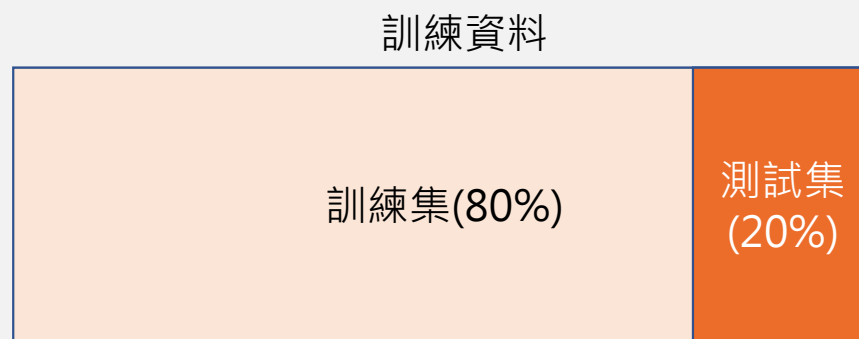
# 11-0 使用 scikit-learn 並匯入測試資料集

- 資料分割：訓練資料集與測試資料集
- 為了得到迴歸模型，我們得先用資料將其訓練出來，就像第八章那樣。
- 然而，光做訓練還不夠，也得確定模型在預測新資料時也能表現得一樣好。
- 因此我們會把訓練資料分割成訓練集(training set)和測試集(testing set)兩塊。
- 若模型對這兩者預測能力差不多，那麼把它拿去預測新資料時，就能得到類似且穩定的表現。



# 11-0 使用 scikit-learn 並匯入測試資料集

- 在機器學習中訓練和測試模型的流程如下：
  1. 將資料分割成訓練集和測試集(後者通常占20%到25%)
  2. 拿訓練集來訓練模型。
  3. 訓練完成後，用模型對測試集的自變數資料來做預測，然後跟測試集的實際目標變數比較看看。



# 11-0 使用 scikit-learn 並匯入測試資料集

- 在scikit-learn中有個功能可讓我們將資料切成訓練集和測試集:

	訓練集(80%)	測試集(20%)
X {	data_train 訓練集自變數	data_test 測試集自變數
Y {	target_train 訓練集目標變數	target_test 測試集目標變數



# 11-0 使用 scikit-learn 並匯入測試資料集

- 程式輸入:

IN

```
from sklearn.model_selection import train_test_split  
  
data_train, data_test, target_train, target_test = train_test_split(data, target, test_size=0.2)
```

匯入分割資料集的函式

從資料中隨機選出 20% 當作測試集



# 11-0 使用 scikit-learn 並匯入測試資料集

- 來檢視看這些資料集裡面有多少項資料。

IN

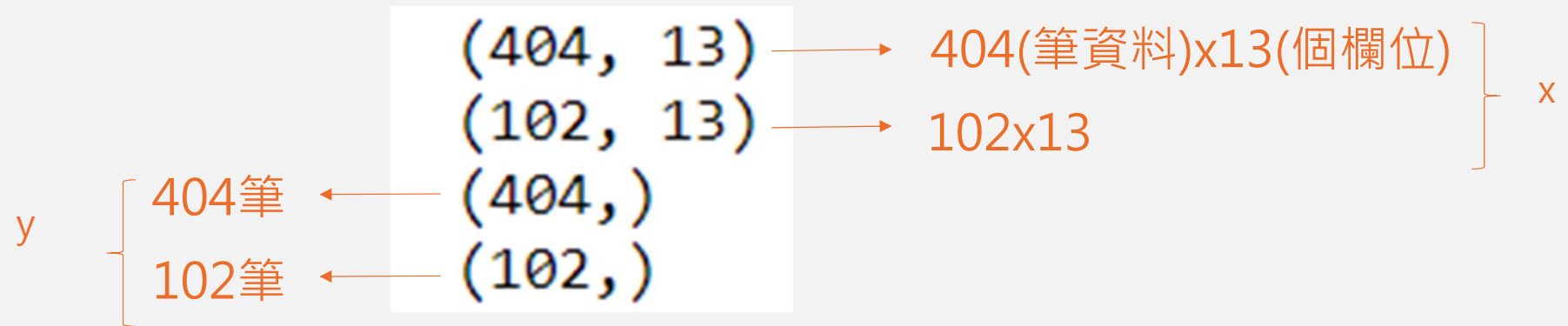
```
print(data_train.shape) ← 它們都是 ndarray, 用 shape  
print(data_test.shape) 屬性會顯示陣列的維度  
print(target_train.shape)  
print(target_test.shape)
```





# 11-0 使用 scikit-learn 並匯入測試資料集

- 來檢視看這些資料集裡面有多少項資料。



- 可看到訓練集跟測試集差不多是4:1(80%對20%)的比例。



# 11-0 使用 scikit-learn 並匯入測試資料集

- 知識補充
- 實務上機器學習還會用到所謂驗證集(validation dataset)。
- 當你要比較多個模型的預測能力或是單一模型在不同參數設定下的表現狀況時，就可以先用驗證集來選出最佳者。
- 此外，驗證集也可以確保模型沒有被過度訓練狀況。
- scikit-learn驗證集的取得是採用所謂的交叉驗證(cross-validation)，將訓練集切成幾塊，每次取一塊當驗證集，其餘則是訓練集，反覆訓練模型後算出平均分數。

後續內容不討論驗證集與交叉驗證議題!!



# 11-0 使用 scikit-learn 並匯入測試資料集

- 使用訓練集產生模型
- 訓練迴歸模型只要從匯入線性迴歸模型，然後呼叫訓練功能即可。

IN

匯入 *scikit-learn* 的線性迴歸模型

```
from sklearn.linear_model import LinearRegression
```

```
regr_model = LinearRegression()
```

← 建立模型物件

```
regr_model.fit(data_train, target_train)
```

← 用訓練集來訓練模型

```
out[7]: LinearRegression()
```

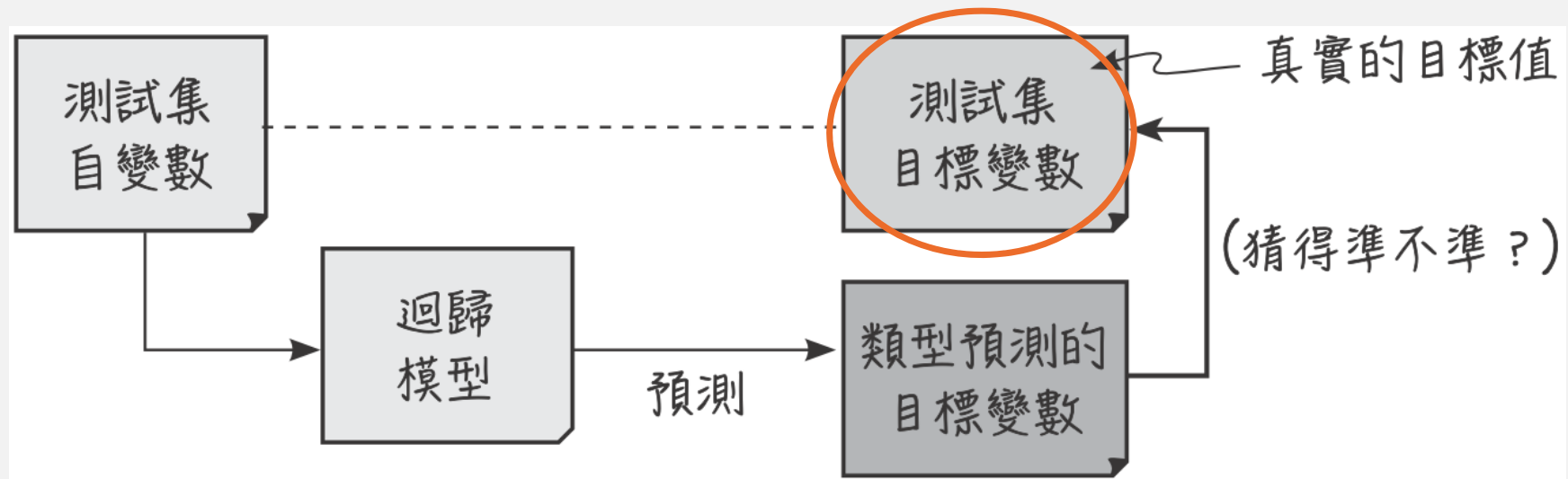


- 現在 **regr\_model** 就是訓練好的 **LinearRegression** (線性迴歸模型) 了。



# 11-1 訓練並評估多元線性迴歸模型

- 產生測試集的預測目標值
- 模型訓練好後就換測試集上場。要讓模型根據測試集的自變數來做預測，看模型對測試集的預測能力是否跟訓練時差不多。



# 11-1 訓練並評估多元線性迴歸模型

- 程式輸入/輸出結果:
- 注意! 由於每次會隨機分割出不同的**訓練集**和**測試集**，所以每次看到的結果很可能不同。
- 比較測試集與預測值的差異會發現，看起來差距不大，但其他筆資料則蠻明顯。
- 下一節將介紹幾種評估預測模型好壞的方法。

第1筆資料

IN

```
predictions = regr_model.predict(data_test)
print(predictions.round(1))
print(target_test)
```

把測試集的自變數資料  
套入模型，產生預測值

← 印出預測值 (四捨五入到小數第1位)

← 印出測試集真實的目標值

[36.3]	20.8	19.9	10.2	31.3	20.7	34.6	31.7	8.3	14.5	21.2	17.1	31.	13.9
23.4	28.9	13.5	19.1	35.2	23.4	30.5	20.	22.	14.3	32.3	23.9	23.1	-1.6
22.2	22.5	20.2	18.4	26.8	34.	17.7	15.5	9.	20.2	32.	18.8	12.2	25.1
17.1	21.1	32.9	25.6	21.6	25.6	15.6	15.3	22.4	10.4	37.3	37.	21.6	16.9
23.8	27.7	20.8	19.8	30.9	32.6	21.3	29.8	36.1	1.4	20.2	34.5	18.	20.5
36.5	20.6	16.6	35.1	22.3	20.2	17.8	31.7	23.	28.	29.1	20.8	19.9	23.
27.6	21.1	27.7	20.8	19.5	22.5	15.8	30.8	43.6	6.2	18.5	28.1	6.5	21.4
14.2	26.1	24.6	27.3]										
[43.1]	20.1	19.3	8.5	29.8	20.9	34.6	35.4	7.	15.4	21.4	17.2	32.9	13.6
24.3	26.6	10.5	19.9	39.8	20.5	31.5	18.4	20.3	13.5	29.	23.	20.1	7.
20.3	21.	24.3	16.6	24.3	33.8	19.9	16.6	27.5	21.5	27.	12.5	12.3	24.7
20.6	25.	27.5	26.5	19.8	22.3	18.4	14.9	19.2	13.2	44.	50.	20.9	19.5
24.4	23.9	21.7	22.2	28.7	37.2	19.7	29.1	46.7	17.9	27.5	35.1	15.4	20.5
38.7	20.	19.9	33.4	19.4	19.2	16.1	31.6	21.7	24.1	24.	21.7	16.7	21.2
26.6	22.	25.	21.2	21.8	22.2	18.9	32.7	50.	8.4	18.3	23.3	11.9	19.6
7.5	22.	22.9	23.9]										

預測目標值

真實目標值



國立中山大學

National Sun Yat-sen University



## 11-2 評估模型的表現 (performance)

- 評估模型表現 1：決定係數(coefficient of determination)
- 有很多統計指標可以用來評估模型的預測能力(到底準不準)，當中最常用的叫做決定係數或判定係數。

IN

```
print(regr_model.score(data_train, target_train).round(3))  
print(regr_model.score(data_test, target_test).round(3))
```

訓練集的決定係數  
(四捨五入到小數第 3 位)

測試集的決定係數  
(四捨五入到小數第 3 位)



0.721  
0.765





## 11-2 評估模型的表現 (performance)

- 決定係數的意思就是自變數資料對目標變數的解釋能力。
- 從結果顯示模型在使用訓練集的自變數時可以解釋訓練集目標變數的74.4%變化，而改用測試集則能解釋目標值72.2%的變化。
- 這兩者數值接近，代表模型沒有被過度訓練。



## 11-2 評估模型的表現 (performance)

- 知識補充:
- 對線性迴歸模型來說，還記得第八章有提到的相關係數 $r$ 平方後剛好會等於決定係數( $R^2$ )。
- 但~這兩者數字涵義不同!!!!
- 相關係數代表的是資料之間的關聯度。
- 決定係數代表自變數對目標變數的解釋力(影響程度)。



## 11-2 評估模型的表現 (performance)

- 知識補充:
- 決定係數介於0到1之間，越接近1代表模型的預測能力越好。
- 但這個值究竟要多高才算有效??? →這並沒有標準答案。
- 在許多領域內 $R^2$ 就算落在0.4~0.6也被認為是有效的。
- 假設你的模型已經將所有重要的變數納入考量，那麼若 $R^2$ 僅有0.2也具有參考性。
- 反之，若不確定模型是否合適，則要去反思是否漏了其他重要因素，或者某些變數是否影響不大等問題....



# 11-2 評估模型的表現 (performance)

- 評估模型表現 2：殘差圖
- 可從視覺化的角度來看模型的預測能力。

IN

```
import numpy as np
import matplotlib.pyplot as plt
```

```
x = np.arange(predictions.size)
y = x * 0
```

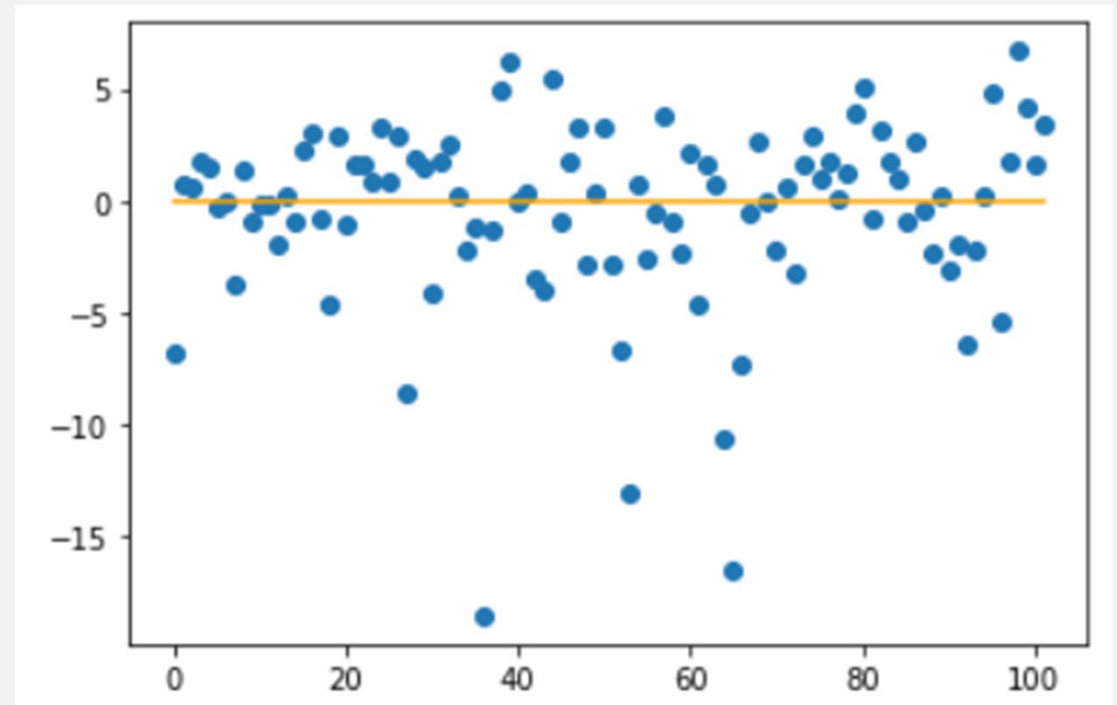
} 根據資料數量產生  
X 軸, Y 軸則為 0

```
plt.scatter(x, predictions - target_test) ← 畫出殘差值
plt.plot(x, y, color='orange') ← 畫出  $y = 0$  的基準線
plt.show()
```



## 11-2 評估模型的表現 (performance)

- 殘差圖運算結果如右:
- 殘差集是預測值跟實際值的差距，只要將所有殘差畫出來，就能看出模型的預測效果有多好。
- 若模型的預測能力越好，那預測值就會越接近實際目標值，使得這些散布點更靠近 $Y=0$ 的水平線。
- 這種圖可用來比較不同模型的效能，相當好用。



## 11-2 評估模型的表現 (performance)

- 評估模型表現 3：平均絕對誤差
- 如果能把殘差值量化，最常用的是平均絕對誤差(mean absolute error, MAE)。
- 意即預測值與實際值差距的絕對值(absolute value)的平均。
- 因此這個值越接近0表示差距越小，預測能力越好。
- scikit-learn的metrics模組提供了mean\_absolute\_error( )可計算兩組資料的MAE:

IN

```
from sklearn.metrics import mean_absolute_error  
print(mean_absolute_error(target_test, predictions).round(3))
```

輸入目標值和預測值來算 MAE



2.809






## 11-2 評估模型的表現 (performance)

- 取得模型的係數
- 若想知道模型的各系數( $A_1$ 、 $A_2$ 、 $A_3...$ 、與B)、以便寫下模型的方程式，你可以用以下程式碼來檢視：

IN

```
print(regr_model.coef_.round(2)) ← 各變數的係數 ( $A_1, A_2, A_3...$ )  
print(regr_model.intercept_.round(2)) ← 截距 (B)
```



```
[-1.000e-02  4.000e-02  3.000e-02  3.980e+00 -1.568e+01  4.970e+00  
-1.000e-02 -1.250e+00  2.100e-01 -1.000e-02 -8.000e-01  1.000e-02  
-4.100e-01]  
23.64
```



## 11-2 評估模型的表現 (performance)

- 這模型即為:

$$\text{MEDV} = -0.01 \times \text{CRIM} + 0.04 \times \text{ZN} + 0.03 \times \text{INDUS} + \dots + 23.64$$



```
[-1.000e-02  4.000e-02  3.000e-02  3.980e+00 -1.568e+01  4.970e+00  
-1.000e-02 -1.250e+00  2.100e-01 -1.000e-02 -8.000e-01  1.000e-02  
-4.100e-01]  
23.64
```

- 同樣取決於`train_test_split()`分割出來的資料集數值，每一次訓練出來的模型係數也會有些差距。



## 11-3 用真實世界的資料做迴歸分析：共享單車與天氣

- 下載韓國首爾市的共享單車及氣象、假日資料
- 此資料為2017-2018年每小時紀錄的共享單車租借次數，外加各種氣象及季節、假日資訊(出自2020年3月研究都會區單車租用需求的論文)。
- 我們將用來探討氣候與假日等條件對人們租借單車的意願有多大的影響。
- 資料集的下載網址如下，請將檔案下載到電腦的下載資料夾中：  
<https://archive.ics.uci.edu/ml/machine-learning-databases/00560/SeoulBikeData.csv>
- 註解:此份資料包含超過8700筆資料。



## 11-3 用真實世界的資料做迴歸分析：共享單車與天氣

- 每筆資料的欄位如右:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Date	Rented Bi	Hour	Temperatu	Humidity(	Wind spee	Visibility	Dew point	Solar Radi	Rainfall(r	Snowfall (	Seasons	Holiday	Functioning Day	
2	01/12/2017	254	0	-5.2	37	2.2	2000	-17.6	0	0	0	Winter	No Holida	Yes	
3	01/12/2017	204	1	-5.5	38	0.8	2000	-17.6	0	0	0	Winter	No Holida	Yes	
4	01/12/2017	173	2	-6	39	1	2000	-17.7	0	0	0	Winter	No Holida	Yes	
5	01/12/2017	107	3	-6.2	40	0.9	2000	-17.6	0	0	0	Winter	No Holida	Yes	
6	01/12/2017	78	4	-6	36	2.3	2000	-18.6	0	0	0	Winter	No Holida	Yes	
7	01/12/2017	100	5	-6.4	37	1.5	2000	-18.7	0	0	0	Winter	No Holida	Yes	
8	01/12/2017	181	6	-6.6	35	1.3	2000	-19.5	0	0	0	Winter	No Holida	Yes	
9	01/12/2017	460	7	-7.4	38	0.9	2000	-19.3	0	0	0	Winter	No Holida	Yes	
10	01/12/2017	930	8	-7.6	37	1.1	2000	-19.8	0.01	0	0	Winter	No Holida	Yes	
11	01/12/2017	490	9	-6.5	27	0.5	1928	-22.4	0.23	0	0	Winter	No Holida	Yes	

⋮



## 11-3 用真實世界的資料做迴歸分析：共享單車與天氣

- 每筆資料的欄位如右:

欄位	Date	Rented Bike Count	Hour	Temperature (°C)	Humidity (%)
意義	日期	單車租借次數 (目標值)	當天第幾小時	溫度 (攝氏)	濕度
欄位	Wind speed (m/s)	Visibility (10m)	Dew point temperature (°C)	Solar Radiation (MJ/m2)	Rainfall (mm)
意義	風速	能見度	露點溫度	陽光輻射量	降雨量
欄位	Snowfall (cm)	Seasons	Holiday	Functioning Day	
意義	降雪量	季節	是否為假日	單車服務是否可用	



## 11-3 用真實世界的資料做迴歸分析：共享單車與天氣

- 匯入資料集到 pandas 的 DataFrame
- DataFrame 容器是處理報表資料的好幫手，在此利用它來讀取跟整理資料：

IN

```
import pandas as pd
```

```
df = pd.read_csv(r'C:\Users\使用者名稱\Downloads\SeoulBikeData.csv', encoding='gbk', index_col=['Date'])
```

df

設定讀取時的編碼

設定 Date 欄位為索引





## 11-3 用真實世界的資料做迴歸分析：共享單車與天氣

- 讀取結果
- 注意:pandas在讀取報表時，預設會使用UTF-8編碼，對於某些中文或亞洲語系的檔案會發生錯誤。
- 這時可以嘗試指定編碼為gbk看看。

Date	Rented Bike Count	Hour	Temperature(度)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(度)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	Seasons	Holiday	Functioning Day
01/12/2017	254	0	-5.2	37	2.2	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
01/12/2017	204	1	-5.5	38	0.8	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
01/12/2017	173	2	-6.0	39	1.0	2000	-17.7	0.0	0.0	0.0	Winter	No Holiday	Yes
01/12/2017	107	3	-6.2	40	0.9	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
01/12/2017	78	4	-6.0	36	2.3	2000	-18.6	0.0	0.0	0.0	Winter	No Holiday	Yes
...	...	...	...	...	...	...	...	...	...	...	...	...	...
30/11/2018	1003	19	4.2	34	2.6	1894	-10.3	0.0	0.0	0.0	Autumn	No Holiday	Yes
30/11/2018	764	20	3.4	37	2.3	2000	-9.9	0.0	0.0	0.0	Autumn	No Holiday	Yes
30/11/2018	694	21	2.6	39	0.3	1968	-9.9	0.0	0.0	0.0	Autumn	No Holiday	Yes
30/11/2018	712	22	2.1	41	1.0	1859	-9.8	0.0	0.0	0.0	Autumn	No Holiday	Yes
30/11/2018	584	23	1.9	43	1.3	1909	-9.3	0.0	0.0	0.0	Autumn	No Holiday	Yes

8760 rows x 13 columns



## 11-3 用真實世界的資料做迴歸分析：共享單車與天氣

- 資料清理 (Data cleaning)
- 現實世界的資料並非像我們之前用load\_xxx那樣一下載就立即可以使用。
- 資料科學的工作有許多時間都是花在資料的蒐集和清理上。
- 所以拿到資料集的第一時間，要先看資料是否有殘缺或不合適的部分。
- 此份資料中可發現單車服務未開放(欄位Functioning Day的值為No時)，租用次數就會是0。
- 這時不管天氣狀況如何，都不可能有人租借單車!
- 因此我們要進行篩選，只保留單車有開放租用時的資料。



## 11-3 用真實世界的資料做迴歸分析：共享單車與天氣

- 程式輸入

IN

```
data = df.copy() ← 先複製一份 DataFrame, 好保留原始資料  
data = data[data['Functioning Day'] == 'Yes'] ←  
將 Functioning Day 欄位的值為 Yes 的部分篩選出來
```

- 在這之後，我們就不需要Functioning Day這欄位，可將這行資料去掉～



## 11-3 用真實世界的資料做迴歸分析：共享單車與天氣

- 丟掉內含有NaN的資料
- 有時資料內會有真正的缺漏(空白、被pandas視為NaN)。
- 若你拿有缺漏的資料做多元迴歸分析，就會產生錯誤。
- 以下敘述將可以把含有NaN的資料(列)全部去掉:

```
In [7]: data.pop('Functioning Day')
```



## 11-3 用真實世界的資料做迴歸分析：共享單車與天氣

- 重新命名欄位名稱
- 此外，你或許會注意到有的欄位名稱有亂碼，是特殊字元存檔後造成的。
- 透過重新命名這些欄位名稱：



IN

從報表複製包含亂碼的欄位名稱

傳入一個字典, 鍵是原本的名稱

值是欄位的新名稱

```
data = data.rename(columns={'Temperature(癡)': 'Temperature(*C)',  
                           'Dew point temperature(癡)': 'Dew point(*C)'})
```

data[['Temperature(\*C)', 'Dew point(\*C)']] ← 用更改後的名稱查詢欄位

Out[14]:

	Temperature(*C)	Dew point(*C)
Date		
01/12/2017	-5.2	-17.6
01/12/2017	-5.5	-17.6
01/12/2017	-6.0	-17.7
01/12/2017	-6.2	-17.6
01/12/2017	-6.0	-18.6
...	...	...
30/11/2018	4.2	-10.3
30/11/2018	3.4	-9.9
30/11/2018	2.6	-9.9
30/11/2018	2.1	-9.8
30/11/2018	1.9	-9.3

8465 rows x 2 columns



## 11-3 用真實世界的資料做迴歸分析：共享單車與天氣

- 將文字的資料『編碼』為數字
- 若資料集中有些資料是文字，無法餵入迴歸模型做計算，譬如Seasons(季節)和Holiday(假日)。以上這些欄位是重要的資料，但要如何轉換成數字來讓迴歸模型做計算呢？
- 可透過標籤編碼器 (label encoder) 將文字資料變成對應的數字。

欄位 Seasons	編碼值	欄位 Holiday	編碼值
Spring (春)	0	Holiday (假日)	0
Summer (夏)	1	No Holiday (非假日)	1
Autumn (秋)	2		
Winter (冬)	3		



## 11-3 用真實世界的資料做迴歸分析：共享單車與天氣

- 用數字來代表四季、假日與非假日，迴歸模型就有辦法可以處理了。
- 我們可用scikit-learn提供的標籤編碼器來完成這任務。

IN

```
from sklearn.preprocessing import LabelEncoder ← 匯入編碼器類別  
le = LabelEncoder() ← 建立編碼器物件  
data['Seasons'] = le.fit_transform(data['Seasons']) ← 將欄位 Seasons 內容轉換成數值  
data['Holiday'] = le.fit_transform(data['Holiday']) ← 將欄位 Holiday 內容轉換成數值
```





## 11-3 用真實世界的資料做迴歸分析：共享單車與天氣

- 現在來檢視一下這兩欄的值，看看轉換結果為何。

Out[14]:

	Seasons	Holiday
Date		
01/12/2017	3	1
01/12/2017	3	1
01/12/2017	3	1
01/12/2017	3	1
01/12/2017	3	1
...	...	...
30/11/2018	0	1
30/11/2018	0	1
30/11/2018	0	1
30/11/2018	0	1
30/11/2018	0	1

8465 rows × 2 columns



## 11-3 用真實世界的資料做迴歸分析：共享單車與天氣

- 抽出目標值
- 最後，我們要將目標值(單車租用次數)從資料集中抽離出來。

IN

```
target = data.pop('Rented Bike Count')
```

- pop()函式會將某欄位的資料從data物件內丟掉，並傳給target變數。



## 11-3 用真實世界的資料做迴歸分析：共享單車與天氣

- `pop()`和`dropna()`有何不同?
- `pop()`會根據指定的名稱刪除(或抽出)DataFrame中的一個欄。
- 而`dropna()`會刪除一個列或一筆資料(假如該筆資料中含有NaN值的話)。
- 如果你在檢視data與target的內容，就會發現data的Rented Bike Count欄位不見了，同時該欄位的值已經存到target了。



## 11-3 用真實世界的資料做迴歸分析：共享單車與天氣

- 開始訓練迴歸模型

將資料分割成訓練集和測試集，  
各自又分為自變數值與目標值

IN

```
data_train, data_test, target_train, target_test = train_  
test_split(data.values, target.values, test_size=0.2)
```

注意這裡得寫 `data.values`,  
只取出值, 以免把索引一併  
放進資料集

同理, 這裡得寫  
`target.values`

取 20% 當 `test set`

```
regr = LinearRegression()  
regr.fit(data_train, target_train) ← 訓練迴歸模型  
predictions = regr.predict(data_test) ← 產生預測值
```

接下行



## 11-3 用真實世界的資料做迴歸分析：共享單車與天氣

- 評估預測成果
- 首先來看訓練集跟測試集的決定係數：

IN

```
print(regr.score(data_train, target_train).round(3))  
print(regr.score(data_test, target_test).round(3))
```

第一次訓練

0.542  
0.533

第二次訓練

0.537  
0.554

- 這兩個數字十分接近，代表模型沒有過度訓練。
- 然而，決定係數只有50%多，代表模型只能解釋目標值的50%變化。
- 但至少我們得知天氣以及假日因素確實能影響人們租用單車的部分意願。



## 11-3 用真實世界的資料做迴歸分析：共享單車與天氣

- 結語
- 在線性迴歸之外，scikit-learn也提供了好幾種非線性迴歸模型。
- 事實上，在研究首爾單車這篇論文中就有提到非線性模型可得到更好的預測效果。

EUROPEAN JOURNAL OF REMOTE SENSING  
2020, VOL. 53, NO. S1, 166–183  
<https://doi.org/10.1080/22797254.2020.1725789>



OPEN ACCESS Check for updates

### A rule-based model for Seoul Bike sharing demand prediction using weather data

Sathishkumar V E and Yongyun Cho

Department of Information and Communication Engineering, Suncheon National University, Suncheon, Republic of Korea

#### ABSTRACT

This research paper presents a rule-based regression predictive model for bike sharing demand prediction. In recent days, Public rental bike sharing is becoming popular because of increased comfortableness and environmental sustainability. Data used include Seoul Bike and Capital Bikeshare program data. Both data have weather data associated with it for each hour. For both the dataset, five statistical models were trained with optimized hyperparameters using a repeated cross validation approach and testing set is used for evaluation: (a) CUBIST (b) Regularized Random Forest (c) Classification and Regression Trees (d) K Nearest Neighbour (e) Conditional Inference Tree. Multiple evaluation indices such as  $R^2$ , Root Mean Squared Error, Mean Absolute Error and Coefficient of Variation were used to measure the prediction performance of the regression models. The results show that the rule-based model CUBIST was able to explain about 95 and 89% of the Variance ( $R^2$ ) in the testing set of Seoul Bike data and Capital Bikeshare program data respectively. An analysis with variable importance was carried to analyse the most significant variables for all the models developed with the two datasets considered. The variable importance results have shown that Temperature and Hour of the day are the most influential variables in the hourly rental bike demand prediction.

#### ARTICLE HISTORY

Received 11 December 2019  
Revised 16 January 2020  
Accepted 1 February 2020

#### KEYWORDS

Data Mining; GIS; predictive analytics; intelligent transport system



國立中山大學

National Sun Yat-sen University



# 下課前，請帶走它~

- scikit-learn是專為機器學習(machine learning)設計的資料科學套件。
- 多元線性迴歸(linear multiple regression)模型是使用多個自變數來預測目標變數。
- 為了確保機器學習模型在訓練完後對新資料也能有類似的預測效果，我們會使用的將資料集分割成訓練集(train dataset)及測試集(test dataset)，並用測試集來測試模型的預測能力。
- LinearRegression( )可用來建立多元線性迴歸模型。建立模型後，先以訓練集為參數執行訓練它，再以測試集為參數執行來產生測試集的預測能力，以評估訓練成效。





# 下課前，請帶走它~

- 決定係數(coefficient of determination,  $R^2$ )代表迴歸模型對目標變數的解釋能力，也就是預測能力。
- 殘差圖(residual plot)可以让你以視覺化方式檢視迴歸模型的預測值與真實目標值的差距。
- 在對真實報表資料做迴歸分析時，可用的來清理資料、並將自變數與目標變數分開。
- 標籤編碼器(label encoder)可將資料中有意義的非數值資料轉換成數字，以利作迴歸分析。

