

成為Python數據分析達人的第一堂課

高一下學期自主學習計畫

11006 林亭妏

2022

CONTENTS

簡述

主題簡介、計畫動機、自主計畫成果表

學習內容簡介

基本語法、套件、線性回歸、機器學習

感想

計劃疏失、心得、期望

主題說明

上ewant網站報名線上課程，
課程主要以影片方式呈現。

影片內容主要包含python程式語言的
基本語法(包含條件判斷、迴圈...)，
以及引入套件進行資料數據的分析處理，
最後延伸到機器學習的部分

計畫動機

對程式有些好奇，且對資工相關科系也
有些興趣，想趁早探索與此相關的程式
學習，增加對自己未來的規劃

The screenshot shows the Ewant website interface. At the top is a blue navigation bar with the Ewant logo, links for '課程探索', '合作夥伴', and '最新消息', a search bar, and a user profile for '林亭妏' with a language dropdown set to '繁體中文 (zh_tw)'. Below the navigation bar is a white section titled '課程資訊'. On the left is a course card for '成為Python數據分析達人的第一堂課 (110高中自主學習)' by 蔡炎龍 from 國立政治大學. The card includes the dates '2021/07/01~2022/06/30', a duration of '7小時/16週 (報名結束)', and social media sharing icons. A red '進入課程' button is at the bottom of the card. To the right of the card is a video player showing the Ewant logo and the text 'education you want!'. At the bottom of the page is a dark navigation bar with tabs: '摘要', '課程目標', '授課教師', '課程進度表', '上課形式', '評分標準', and '證書資訊'. There are also 'TOP' and '消息' buttons on the right.

國立新竹女中彈性學習時間學生自主學習成果表

2022/07/20 10:21

申請人	林亭玟	班級/座號	1 年 10 班 06 號
申請學期	11002	申請時數	16
共學同學	無		
計畫名稱	成為 Python 數據分析達人的第一堂課		
學習類型	學科課程延伸	對應學科屬性	資訊
設備需求	筆電		
指導教師			
自主學習內容概述	對程式設計有興趣，希望能藉由線上觀看這門課程學 Python 的基本語法以及學習基礎的程式撰寫，並且學習計算思維，進而發展將各種問題程式化，並有效解決的能力。		
預期效益	1. 建立 Python 程式應用與多面向之思考與學習，增進程式語法與數據分析相關知識、實例解析與應用之能力 2. 實踐及運用課堂知識，培養問題理解、思辨分析之素養，訓練規劃與執行計畫之能力		
與十二年國教核心素養之關聯	A2 系統思考與解決問題、A3 規劃執行與創新應變、B2 科技資訊與媒體素養		
成果展示	同意於校內學習平台提供自主學習成果與資料給其他同學參考		

週次	日期	自學內容	檢核進度	學習心得	自學場地
3	111/02/24(四)	Python 程式基礎 I：Python 的開發環境、串列與繪圖	完全達標	看了影片，但無法如當初規畫將章節影片在一節課內全部看完	
4	111/03/03(四)	複習上週內容與做練習題	稍有落後	有在下週進新章節前將進度影片看完，但無時間複習上週內容內容	
5	111/03/10(四)	Python 程式基礎 II：迴圈、條件判斷	完全達標	有利用一些課外時間將本章節影片看完，覺得函數部分比較需要時間理解	
6	111/03/17(四)	複習上週內容與做練習題	稍有落後	沒複習到上週內容，先看了一些下週章節的影片	
8	111/03/31(四)	Python 強大的秘密：numpy、向量化與各式套件	完全達標	將影片看完，numpy 和套件部分開始覺得有些難度	
9	111/04/07(四)	複習上週內容與做	待改進	因為之前無法一章	

		練習題		節一章節複習所以直接一起下週期末考範圍	
10	111/04/14(四)	python 期中考	完全達標	考了 92 分，但考試過程有看了一下影片筆記	
11	111/04/21(四)	處理與展示資料的技巧：Python 的字典檔與 jupyter 互動功能	完全達標	有將影片看完	
13	111/05/05(四)	複習上上週內容與做練習題	待改進	沒有複習上次內容直接先看下週影片	
14	111/05/12(四)	試算表：pandas 數據分析	完全達標	有將影片進度看完	
15	111/05/19(四)	複習上週內容與做練習題	待改進	沒有複習上次內容直接先看下週影片	
16	111/05/26(四)	預測未來：用線性迴歸做預測	完全達標	有將影片進度看完	
17	111/06/02(四)	複習上週內容與做練習題	待改進	沒有複習上次內容直接先看下週影片	
18	111/06/09(四)	成為機器學習專家：機器學習概要	完全達標	有將影片進度看完	
19	111/06/16(四)	複習上週內容與做練習題	無	恢復實體課後事務繁雜沒有複習到	
20	111/06/23(四)	python 期末考	無	恢復實體課後事務繁雜因此沒有考到試	

成果說明：文字

在一邊看線上影片同時也有跟著影片內容打程式碼，在自主學習課程中漸漸認識 python 語言的基礎語法、迴圈、條件...接著學習到引進套件的部分像是 numpy、pandas...並綜合以上學得進行資料的數據分析和處理的程式編寫，最後則是藉由程式認識機器學習的理念。

成果說明：照片

說明：	說明：
-----	-----

影片中的基本語法

數字的計算
變數
List
字串
Input互動
條件判斷
迴圈
函數
亂數

```
繼續  
狗狗  
睡著
```

```
In [6]: for i in range(1,11):  
        print(i*2)
```

```
2  
4  
6  
8  
10  
12  
14  
16  
18  
20
```

for迴圈

拍拍機器人 v1.0

```
In [ ]: message = ""  
while message != "bye":  
    message = input(">> ")  
    print("拍拍")
```

```
>> 今天很難過  
拍拍  
>> 功課好難  
拍拍  
>> 同學都笑我  
拍拍
```

while迴圈
條件判斷
Input

字串的格式化

```
In [2]: name = "亭紋"  
        place = "新竹"
```

```
In [3]: message = "你好,來自"+place+"的"+name+"!"
```

```
In [4]: print(message)
```

```
你好,來自新竹的亭紋!
```

```
In [5]: message = "你好,來自{}的{}!".format(place,name)
```

```
In [6]: print(message)
```

```
你好,來自新竹的亭紋!
```

字串

```
In [2]: name = "亭紋"  
        place = "新竹"
```

```
In [3]: message = "你好,來自{}的{}!".format(place,name)
```

```
In [4]: message
```

```
Out[4]: '你好,來自新竹的亭紋!'
```

```
In [5]: def say_hi(name,place): #函數名字不能打空格  
        message = "你好,來自{}的{}!".format(place,name)  
        return message
```

```
In [6]: say_hi("依晨","羅東")
```

```
Out[6]: '你好,來自羅東的依晨!'
```

```
In [7]: say_hi("琦貞","台北")
```

```
Out[7]: '你好,來自台北的琦貞!'
```

函數

```
%pylab inline
```

Populating the interactive namespace from numpy and matplotlib

```
[2]: randint(3,21) #3~20
```

```
Out[2]: 9
```

```
In [3]: for i in range(10):  
        k = randint(1,50)  
        print(" *k + " ")
```

亂數

學習方式 在看影片同時，照著老師的操作自己編寫課程程式

重點套件

Numpy

支援高階大量的維度陣列與矩陣運算，也針對陣列運算提供大量的數學函數函式庫。

```
In [2]: import numpy
In [3]: numpy.sin(3)
Out[3]: 0.1411200080598672
In [4]: numpy.cos(3)
Out[4]: -0.9899924966004454
In [5]: numpy.pi
Out[5]: 3.141592653589793
In [6]: import numpy as np
In [7]: np.sin(3)
Out[7]: 0.1411200080598672
In [8]: np.cos(3)
Out[8]: -0.9899924966004454
```

學習運用

- 處理list數字
- 數學函數應用

Array

陣列(Array)是線性且同質的資料結構，使用零或正整數為索引來存取其中元素

```
In [2]: %matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
In [3]: grades = np.array([85,70,82])
In [6]: weights = np.array([0.3,0.4,0.3])
In [7]: g = grades * weights
In [8]: g
Out[8]: array([25.5, 28. , 24.6])
In [9]: g.sum()
Out[9]: 78.1
In [10]: np.dot(grades,weights) #np.dot(array1,array2)將array1跟array2做內積
Out[10]: 78.1
In [11]:
```

學習應用

- 用Array算成績(加權和)
- Array畫圖
- Array快篩法(索引)

Pandas

「Python界的Excel」，結合NumPy的特性，以及試算表和關連式資料庫（SQL）的資料操作能力，可以用來對資料進行重構、切割、聚合及選擇子集合等操作

```
In [1]: %matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
In [3]: df = pd.read_csv("grades.csv")
In [4]: df.head()
Out[4]:
```

	姓名	國文	英文	數學	自然	社會
0	劉俊安	9	10	15	10	13
1	胡曉華	10	10	10	8	9
2	黃淑婷	13	15	8	11	14
3	陳上紫	10	10	8	9	14
4	崔靜成	13	12	14	12	13

```
In [5]: df.國文.mean()
Out[5]: 11.0
In [6]: df.國文.std()
Out[6]: 1.8708286933869707
```

學習運用

- 用熊貓做基本數據分析
- DataFrame
- loc定位法

1.

```
In [1]: %matplotlib inline
import numpy as np
import matplotlib.pyplot as plt

In [2]: from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

#波士頓房價數據集

In [3]: from sklearn.datasets import load_boston

In [5]: boston = load_boston()

可以用feature_names查看數據集中所有的features - 包含像總面積等共13個features

In [6]: boston.feature_names
Out[6]: array(['CRIM', 'INDUS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD',
              'TAX', 'PTRATIO', 'B', 'LSTAT'], dtype=object)

In [7]: X = boston.data
Y = boston.target

In [8]: len(X)
Out[8]: 506

In [9]: X_train, X_test, y_train, y_test = train_test_split(X, Y,
                                                          test_size=0.3,
                                                          random_state=0)

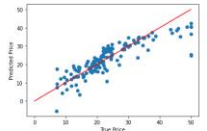
In [10]: regr = LinearRegression()

In [11]: regr.fit(X_train, y_train)
Out[11]: LinearRegression()

In [12]: y_predict = regr.predict(X_test)

In [14]: plt.scatter(y_test, y_predict)
plt.plot([0, 50], [0, 50], 'r')
plt.xlabel('True Price')
plt.ylabel('Predicted Price')

Out[14]: Text(0, 0.5, 'Predicted Price')
```



```
波士頓數據內容

In [15]: 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
- INDUS   proportion of residential land zoned for lots over 25,000 sq.ft.
- NOX     proportion of non-retail business acres per town.
- TAX     full-value property-tax rate per $10,000
- PTRATIO pupil-teacher ratio by town
- B       lower middle class of black people 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 VEC 18: housing dataset.
https://archive.ics.utexas.edu/machine-learning-databases/housing/

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

The Boston house-price data of Harrison, D. and Rubinfeld, D.L. "Hedonic
prices and the demand for clean air", J. Service, Economics & Management,
vol.5, 81-102, 1978. Used in Belsley, Kuh & Welsch, "Regression diagnostics
...", Wiley, 1980. B-2. 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 regression
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 of the Tenth International Conference of Machine Learning, 236-243, University of Massachusetts, Amherst. Morgan Kaufmann.

[小技巧] 善用enumerate

In [16]: L = ['a', 'b', 'c']

In [17]: for i in L:
print(i)

a
b
c
```

2.

```
In [18]: for i in range(3):
print(L[i], L[i])

1 a
2 b
3 c

In [19]: list(enumerate(L))
Out[19]: [(0, 'a'), (1, 'b'), (2, 'c')]

In [20]: for i in enumerate(L):
print(i)

(0, 'a')
(1, 'b')
(2, 'c')

In [21]: for i, x in enumerate(L):
print(i, x)

1 a
2 b
3 c

[小技巧] 遍多維度

In [24]: x = np.linspace(-10, 10, 200)

In [27]: plt.plot(x, np.sin(x))
plt.plot(x, np.cos(x))

Out[27]: [matplotlib.lines.Line2D at 0x22af9cd10b]

In [28]: plt.subplot(2,1,1)
plt.plot(x, np.sin(x))

plt.subplot(2,2,2)
plt.plot(x, np.cos(x))

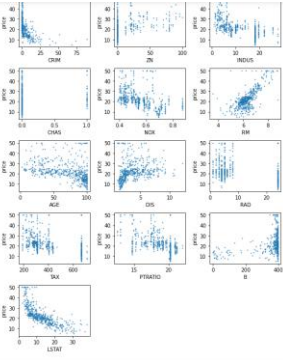
plt.subplot(2,3,3)
plt.plot(x, x)

plt.subplot(2,4,4)
plt.plot(x, x**2)

Out[28]: [matplotlib.lines.Line2D at 0x22af9cd20b]

In [29]: plt.figure(figsize=(6,10))
for i, feature in enumerate(boston.feature_names):
plt.subplot(2,3,i+1)
plt.scatter(L[i], Y, s=1)
plt.xlabel('price')
plt.ylabel(feature)
plt.tight_layout()

In [30]:
```



```
In [31]:
```

線性回歸



—找出符合資料規律的直線

章節學習重點

- 過度擬合overfitting
- 訓練資料和測試資料

用線性迴歸預測波士頓房價& 畫出各 feature 和輸出關係


```
In [5]: %matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
```

用SVM來做分類

先做個簡單的資料

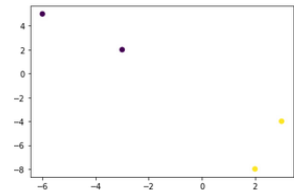
假設我們有四個點，有兩個類別

```
In [7]: x = np.array([[ -3,2],[ -6,5],[3,-4],[2,-8]])
y = np.array([1,1,2,2])
```

我們要畫圖時，需要把 x 中點 x-座標、y-座標 分成兩個 list(array)。記得我們要 x 全部是這樣叫出來的

而 x 中前面的元素(x-座標)是這樣。當然 y-座標也是類似方法叫出來。

```
In [11]: plt.scatter([ -3,-6,3,2],[2,5,-4,-8],c=y)
Out[11]: <matplotlib.collections.PathCollection at 0x1afc3395ea0>
```



```
In [12]: x
Out[12]: array([[ -3, 2],
               [ -6, 5],
               [ 3, -4],
               [ 2, -8]])
```

```
In [13]: x[2,1]
Out[13]: -4
```

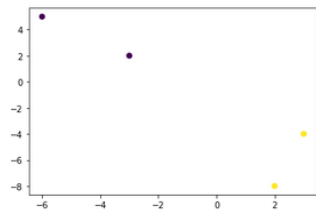
```
In [14]: x[:,0]
Out[14]: array([ -3, -6,  3,  2])
```

```
In [15]: x[:,1]
Out[15]: array([ 2,  5, -4, -8])
```

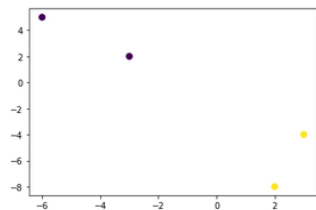
1.

2.

```
In [16]: plt.scatter(x[:,0],x[:,1],c=y)
Out[16]: <matplotlib.collections.PathCollection at 0x1afc33f8fd0>
```



```
In [17]: plt.scatter(x[:,0],x[:,1],s=50,c=y)
Out[17]: <matplotlib.collections.PathCollection at 0x1afc34673d0>
```



```
In [18]: x
Out[18]: array([[ -3,  2],
               [ -6,  5],
               [  3, -4],
               [  2, -8]])
```

```
In [19]: y
Out[19]: array([1, 1, 2, 2])
```

SVM 支持向量機

一個用曲線把資料分隔的辦法。在高維度的時候自然就是曲面(超曲面)分隔資料的方法。

```
In [20]: from sklearn.svm import SVC
```

```
In [22]: clf = SVC() #支持向量機簡合的方法 機器的變數名稱 = SVC()
```

```
In [23]: clf.fit(x,y) #支持向量機訓練的方法 機器的變數名稱 .fit(輸入資料,正確答案)
```

```
Out[23]: SVC()
```

預測結果

```
In [24]: clf.predict([[-3,2]])
```

```
Out[24]: array([1])
```

```
In [25]: clf.predict(x)
```

```
Out[25]: array([1, 1, 2, 2])
```

```
In [26]: clf.predict([[2.5,3]])
```

```
Out[26]: array([1])
```

meshgrid

```
In [27]: xx = [1,2,3,4]
yy = [5,6,7,8]
```

```
In [28]: X, Y = np.meshgrid(xx,yy) #用 meshgrid 找格點 X,Y = np.meshgrid(X格點,Y格點)
```

```
In [29]: X
```

```
Out[29]: array([[1, 2, 3, 4],
               [1, 2, 3, 4],
               [1, 2, 3, 4],
               [1, 2, 3, 4]])
```

```
In [30]: Y
```

```
Out[30]: array([[5, 5, 5, 5],
               [6, 6, 6, 6],
               [7, 7, 7, 7],
               [8, 8, 8, 8]])
```

```
In [34]: X, Y = np.meshgrid(np.linspace(-6,3,30), np.linspace(-8,5,30))
```

```
In [35]: X
```

```
Out[35]: array([[ -6.         , -5.68965517, -5.37931034, -5.06896552, -4.75862069,
               -4.44827586, -4.13793103, -3.82758621, -3.51724138, -3.20689655,
               -2.89655172, -2.5862069 , -2.27586207, -1.96551724, -1.65517241,
               -1.34482759, -1.03448276, -0.72413793, -0.4137931 , -0.10344828,
               0.20689655, 0.51724138, 0.82758621, 1.13793103, 1.44827586,
               1.75862069, 2.06896552, 2.37931034, 2.68965517, 3.         ],
               [ -6.         , -5.68965517, -5.37931034, -5.06896552, -4.75862069,
               -4.44827586, -4.13793103, -3.82758621, -3.51724138, -3.20689655,
               -2.89655172, -2.5862069 , -2.27586207, -1.96551724, -1.65517241,
               -1.34482759, -1.03448276, -0.72413793, -0.4137931 , -0.10344828,
               0.20689655, 0.51724138, 0.82758621, 1.13793103, 1.44827586,
               1.75862069, 2.06896552, 2.37931034, 2.68965517, 3.         ],
               [ -6.         , -5.68965517, -5.37931034, -5.06896552, -4.75862069,
               -4.44827586, -4.13793103, -3.82758621, -3.51724138, -3.20689655,
               -2.89655172, -2.5862069 , -2.27586207, -1.96551724, -1.65517241,
               -1.34482759, -1.03448276, -0.72413793, -0.4137931 , -0.10344828,
               0.20689655, 0.51724138, 0.82758621, 1.13793103, 1.44827586,
               1.75862069, 2.06896552, 2.37931034, 2.68965517, 3.         ],
               [ -6.         , -5.68965517, -5.37931034, -5.06896552, -4.75862069,
```

3.

機器學習—

資料分類

4.

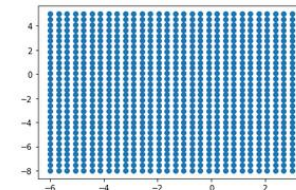
```
-1.34482759, -1.03448276, -0.72413793, -0.4137931 , -0.10344828,
0.20689655, 0.51724138, 0.82758621, 1.13793103, 1.44827586,
1.75862069, 2.06896552, 2.37931034, 2.68965517, 3.         ],
[-6.         , -5.68965517, -5.37931034, -5.06896552, -4.75862069,
```

reve拉平法

```
In [54]: X = X.ravel()
Y = Y.ravel()
```

```
In [37]: plt.scatter(X,Y)
```

```
Out[37]: <matplotlib.collections.PathCollection at 0x1afe57de7e0>
```



zip 高級組合法

```
In [50]: xx = [1,2,3,4]
yy = [5,6,7,8]
```

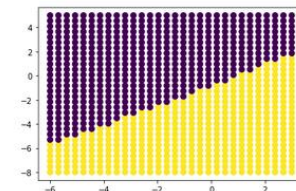
```
In [51]: list(zip(xx,yy))
```

```
Out[51]: [(1, 5), (2, 6), (3, 7), (4, 8)]
```

```
In [52]: Z = clf.predict(list(zip(X,Y)))
```

```
In [58]: plt.scatter(X,Y,s=50,c=Z)
```

```
Out[58]: <matplotlib.collections.PathCollection at 0x1afe5e153a0>
```



自主學習的疏失

這是我第一次自己規劃自主學習，然而度過這一學期後，我發現了不少計畫安排上的問題點，甚至最後還有一點進度沒有做完(線上期末考沒考到)

- 與計劃的落差

一邊看影片一邊照著打程式，無法在預期的時間內將影片看完，導致原本下週原本要複習和做練習題的時間改成繼續把上週沒看完的內容看完

- 學習效率低

只跟著影片照樣打程式，也沒有額外練型的情況下，我發覺我根本無法自己打出一個完整程式，在課程後期也更容易出現看不懂程式的情況



(._.)



心得

這次藉由學習Python，雖然語法上還沒到精熟，不過也有大致認識了它的架構。另外，在學習期間，有時發現跟著影片步驟走執行結果卻錯誤的狀況時，我會上網找尋解決方法，而當方法找到並問題解決時，這往往令我感到舒心且很有成就感。在意識到這次計畫安排上的不足後，這些不管是好是壞的經歷也激起我繼續學習程式的心。



期許

雖然這次的自主學習不如預期，不過有了這次經驗後我也對下次的自主學習也有了一些發想，像是：

- 要學的「扎實」，即使進度可能比較慢，但至少要把這部分的東西弄懂再繼續往下學
- 作練習很重要，只是一味地跟著影片打一模一樣的程式碼並無法培養真正寫程式的能力

希望下次在學習程式時，能夠吸取這些教訓，並安排出更有效率的學習！

jupyter			Quit	Logout
<input type="checkbox"/>	Videos	1 個月前		
<input type="checkbox"/>	0407 國語的互動之 over fitting.ipynb	Running 1 天前	10 KB	
<input type="checkbox"/>	0606 訓練資料和測試資料.ipynb	Running 1 個月前	39.1 KB	
<input type="checkbox"/>	0607 用線性迴歸預測波士頓房價& 0608 畫出各 feature 和輸出關係.ipynb	Running 1 個月前	183 KB	
<input type="checkbox"/>	0103 Jupyter 筆記本環境.ipynb	Running 5 個月前	1.21 KB	
<input type="checkbox"/>	0104 Jupyter 的魔術.ipynb	5 個月前	58.9 KB	
<input type="checkbox"/>	0105 markdown.ipynb	3 個月前	1.19 KB	
<input type="checkbox"/>	0106 變數好好用.ipynb	5 個月前	3.85 KB	
<input type="checkbox"/>	0108 串列 List 是數據分析第一步.ipynb	5 個月前	13.4 KB	
<input type="checkbox"/>	0109 字串 Hello World.ipynb	5 個月前	4.72 KB	
<input type="checkbox"/>	0110 互動的開始 input.ipynb	Running 5 個月前	5.95 KB	
<input type="checkbox"/>	0201 條件判斷.ipynb	Running 4 個月前	4.01 KB	
<input type="checkbox"/>	0202 用迴圈刪除資料.ipynb	Running 4 個月前	3.44 KB	
<input type="checkbox"/>	0203 字串完全解我們整理格式化.ipynb	Running 4 個月前	2.64 KB	
<input type="checkbox"/>	0204 函數的魔法初探.ipynb	Running 4 個月前	3.26 KB	
<input type="checkbox"/>	0205 函數練習.ipynb	Running 4 個月前	2.36 KB	
<input type="checkbox"/>	0206 List 的初探與解.ipynb	Running 4 個月前	4.65 KB	
<input type="checkbox"/>	0207 字串混合 list.ipynb	4 個月前	1.87 KB	
<input type="checkbox"/>	0208 List 快速生成法.ipynb	4 個月前	2.2 KB	
<input type="checkbox"/>	0209 for 迴圈.ipynb	Running 4 個月前	2.42 KB	
<input type="checkbox"/>	0210 我們要訪的資料在哪裡嗎.ipynb	Running 4 個月前	2.71 KB	
jupyter			Quit	Logout
<input type="checkbox"/>	0305 處理一個 ndarray.ipynb	4 個月前	1.99 KB	
<input type="checkbox"/>	0306 Array 真是太炫了.ipynb	Running 4 個月前	2.36 KB	
<input type="checkbox"/>	0307 用 Array 完成總加權和().ipynb	Running 4 個月前	2.37 KB	
<input type="checkbox"/>	0308 實地玩陣列.ipynb	4 個月前	2.5 KB	
<input type="checkbox"/>	0309 Array 大變身.ipynb	4 個月前	17.2 KB	
<input type="checkbox"/>	0310 Array 快速生成法.ipynb	4 個月前	4.46 KB	
<input type="checkbox"/>	0311 Array 畫圖.ipynb	4 個月前	60.4 KB	
<input type="checkbox"/>	0312 Array 快翻法.ipynb	3 個月前	34.6 KB	
<input type="checkbox"/>	0402 字串資料型態.ipynb	3 個月前	2.26 KB	
<input type="checkbox"/>	0403 超炫的互動技能之一.ipynb	Running 1 天前	4.75 KB	
<input type="checkbox"/>	0404 超炫的互動技能之二.ipynb	3 個月前	2.8 KB	
<input type="checkbox"/>	0405 互動應用範例.ipynb	3 個月前	2.53 KB	
<input type="checkbox"/>	0406 國語的互動之一.ipynb	3 個月前	24.5 KB	
<input type="checkbox"/>	0407 國語的互動之 over fitting.ipynb	3 個月前	2.24 KB	
<input type="checkbox"/>	0502 什麼是 Python 的 Excel.ipynb	Running 2 個月前	37.5 KB	
<input type="checkbox"/>	0503 用 pandas 做基本數據分析.ipynb	Running 2 個月前	24.5 KB	
<input type="checkbox"/>	0504 組織自己的 DataFrame.ipynb	Running 2 個月前	15.3 KB	
<input type="checkbox"/>	0505 用 Groupby 找到美國最棒且最 UFO 的州.ipynb	Running 2 個月前	61.5 KB	
<input type="checkbox"/>	0506 Pandas 的 loc 定位法.ipynb	2 個月前	14.9 KB	
<input type="checkbox"/>	0507 玩玩真的超風資料.ipynb	Running 2 個月前	84.7 KB	
<input type="checkbox"/>	0603 機器學習的數據 & 0604 真的資料真的迴歸.ipynb	Running 2 個月前	44.1 KB	
<input type="checkbox"/>	0605 快速混合 over fitting.ipynb	Running 1 天前	36.9 KB	

THANK YOU

感謝觀看

THANK YOU

Elva-Design