



# AI CUP 2024

教育部大專校院人工智慧競賽

**讓綠能觸手可及 -根據區域微氣候資料預測發電量**

主持人:東華大學資工系 顏士淨教授

協同主持人:東華大學光電系 白益豪教授

助理:東華大學資工系彭英茜

兼任助理:古李傑驛、黃聖文



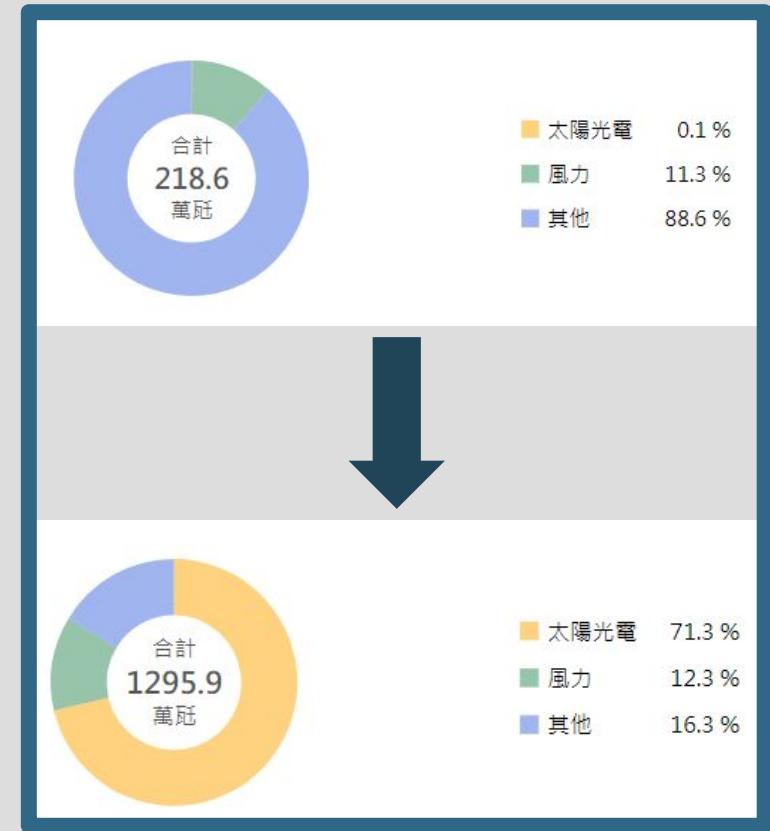
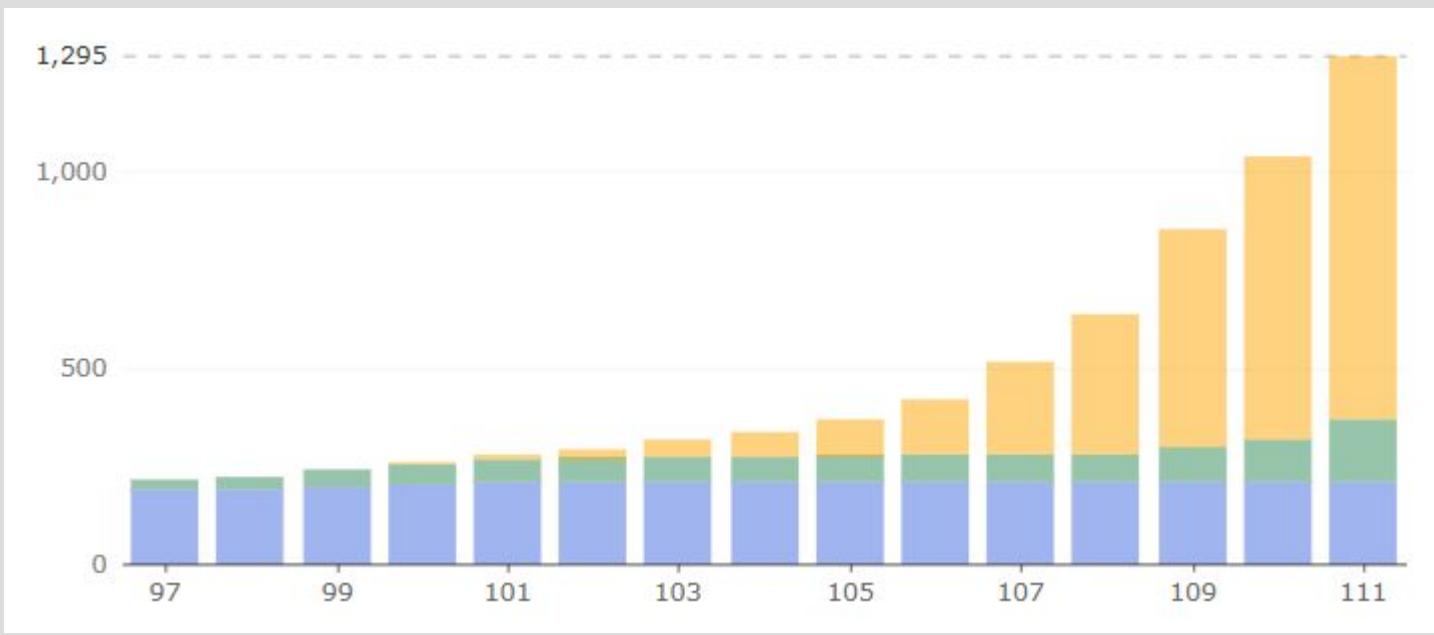
## 競賽背景與目標

- 太陽光電裝置的效能不斷提升，綠能科技已經成為世界未來發展的共識，AI發展也突飛猛進，AI綠能為人才培育一個好題目
- 本競賽根據能源科技相關文獻，選擇出微氣候環境因子進行量測，收集資料，架設太陽能板量測發電量作為ground truth
- 符合綠能科技徵件主題，可以讓AI CUP參賽學生了解綠能科技產業，幫助台灣綠能科技發展
- 主題為預測太陽能當天發電量，有助於台灣電網每天電力調控
- 內容為提供每個裝置每分鐘的氣象數據，利用預測前一天至預測當天早上9:00之前的各項數據，預測當天9:00之後的每10分鐘發電瓦數，持續預測直至當天晚上5:00，共有48筆預測，誤差總和為總分數



# 緣起

太陽能從民國97年代至111年，逐漸成為再生能源的翹楚



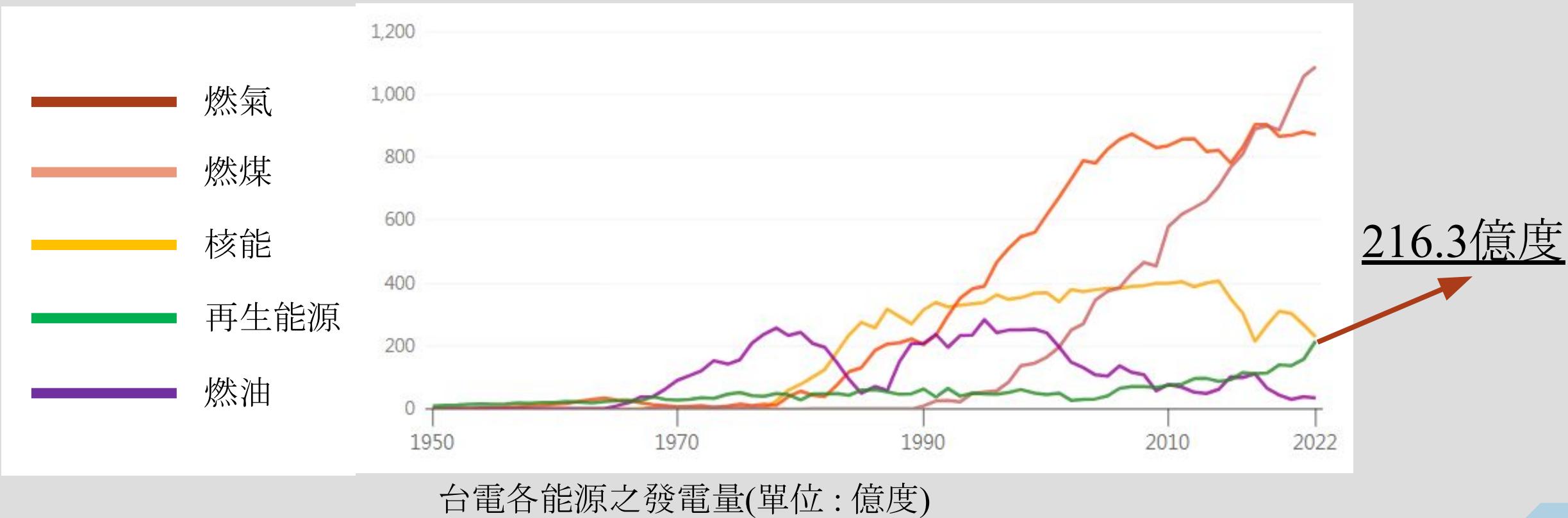
台電再生能源歷年發電佔比(單位：萬瓩)

台電再生能源發電詳細佔比



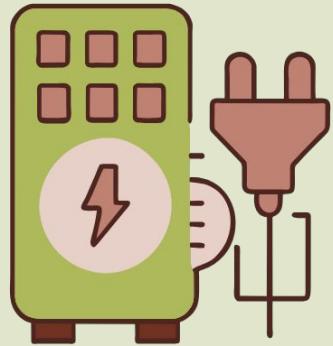
# 緣起

在永續節能的意識抬頭下再生能源的發電量逐漸增加



# 情境

- 太陽能源作為一個佔比極高的再生能源，在發電系統中佔有一席之地，發電的不穩定勢必影響電網的配電機制，增加額外的成本，因此預測太陽能發電量極為重要。
- 預測當天的太陽能發電量，可提供當天電網的配電機制參考**

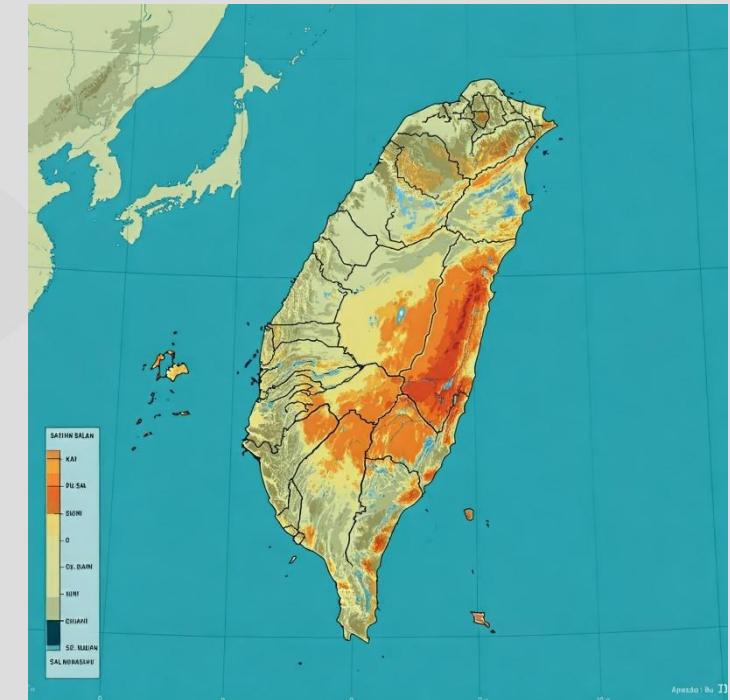
太陽能	
優點 □ 節能減碳	
缺點 □ 發電不穩定	

機種	啟動時間
燃氣	$\geq 10\text{ min}$
燃煤-冷啟動	$\geq 300\text{ min}(5\text{ h})$
燃煤-熱啟動	$\geq 30\text{ min}$
核能	$\geq 1\text{ day}$



# 使用微氣候資料預測當天太陽光電發電量

- 競賽內容介紹：
- 如何標註：
  - 包括自行架設太陽能板與感測器裝置量測，收集氣象局公開氣候資料等
- public & private資料量
  - Training set : 每一天為一筆，約2000筆
  - Test set : 每兩天為一筆，約600筆



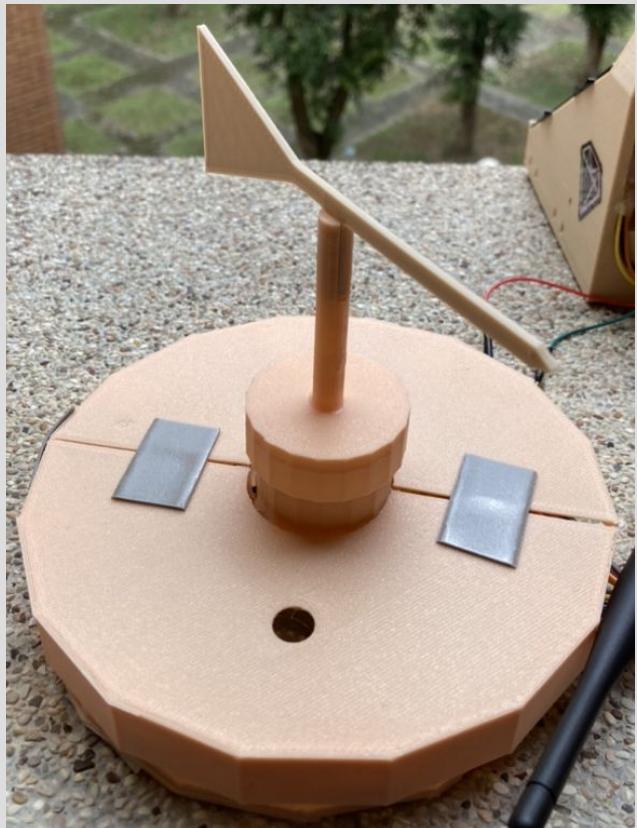
## 資料來源或獲取資料方式

- 量測地點在花蓮東華大學校本部(花蓮溪木瓜溪沖積平原, 農業區)、美崙校區(花蓮市區, 石材工廠附近, 太平洋海岸海邊)、鳳林光電發展特區(花東縱谷山區)等三個地區, 每個地區的不同地點, 如高樓樓頂或地面, 通風處或森林旁邊等等。

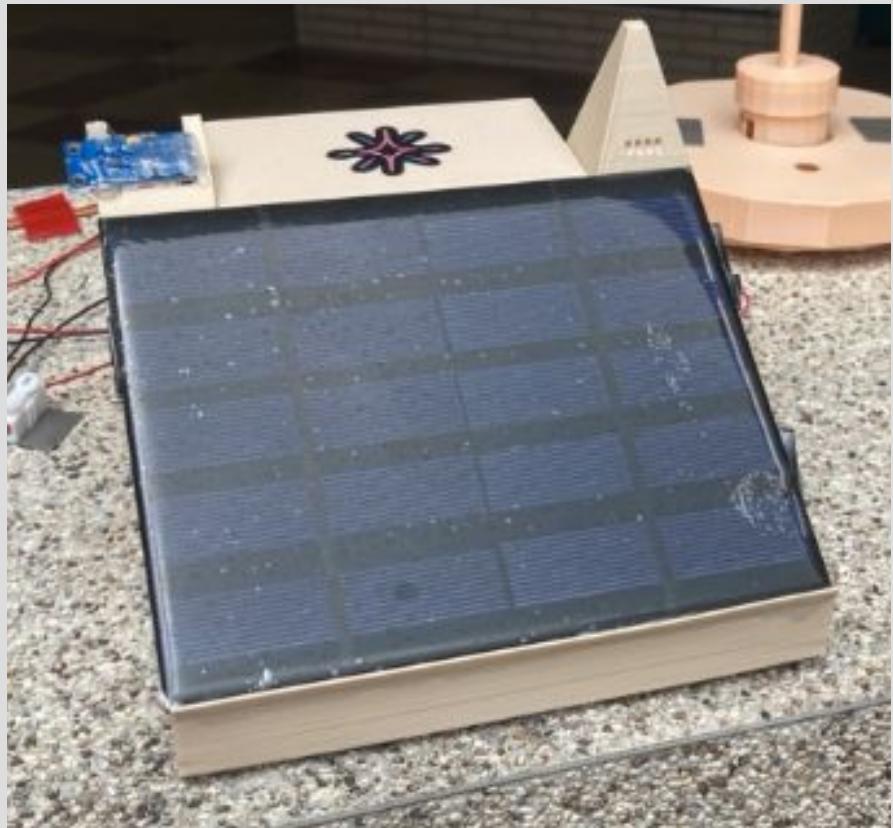


# 太陽能數據收集裝

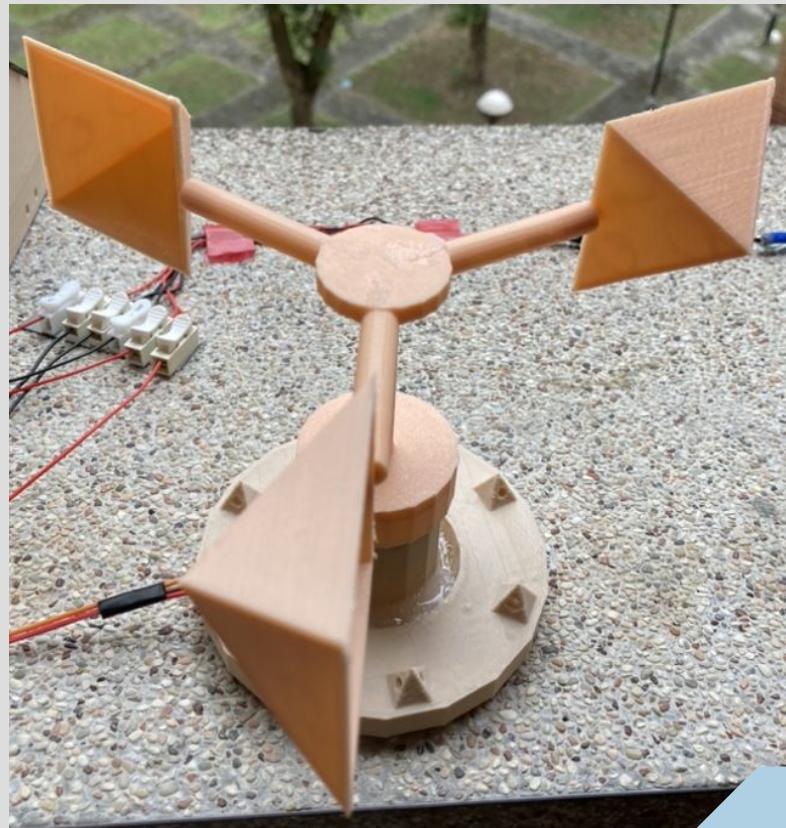
目前已上線之裝置	東華理工頂樓 一號機	東華理工頂樓 二號機	東華理二 3樓教室陽台	東華理二 1樓	花蓮市 自家住宅
預計上線之裝置	東華美崙校區	小雨蛙生態民宿 農田	七星潭	德興棒球場	新北市 住宅



8方位風向感測儀器



太陽能數據收集裝置主體



風速感測儀器

# 太陽能數據收集裝置



裝置內部感測模組

ESP32S	DHT22	BH1750	BMP280	INA219
MCU	溫濕度	光照度	大氣壓力	電源監測



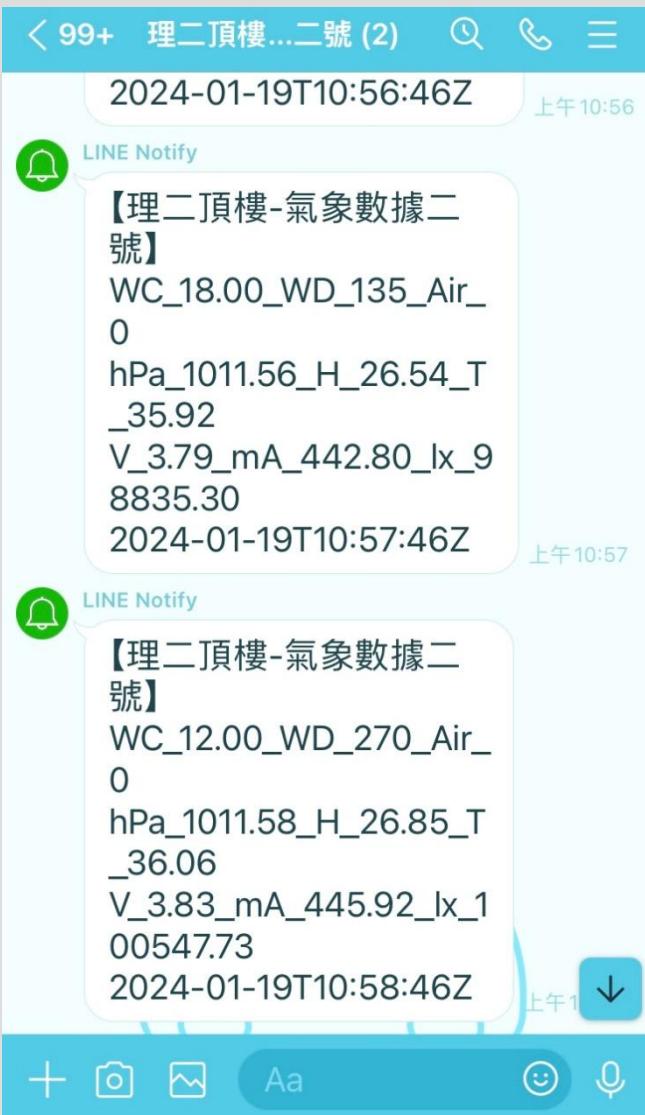
# 資料儲存

存

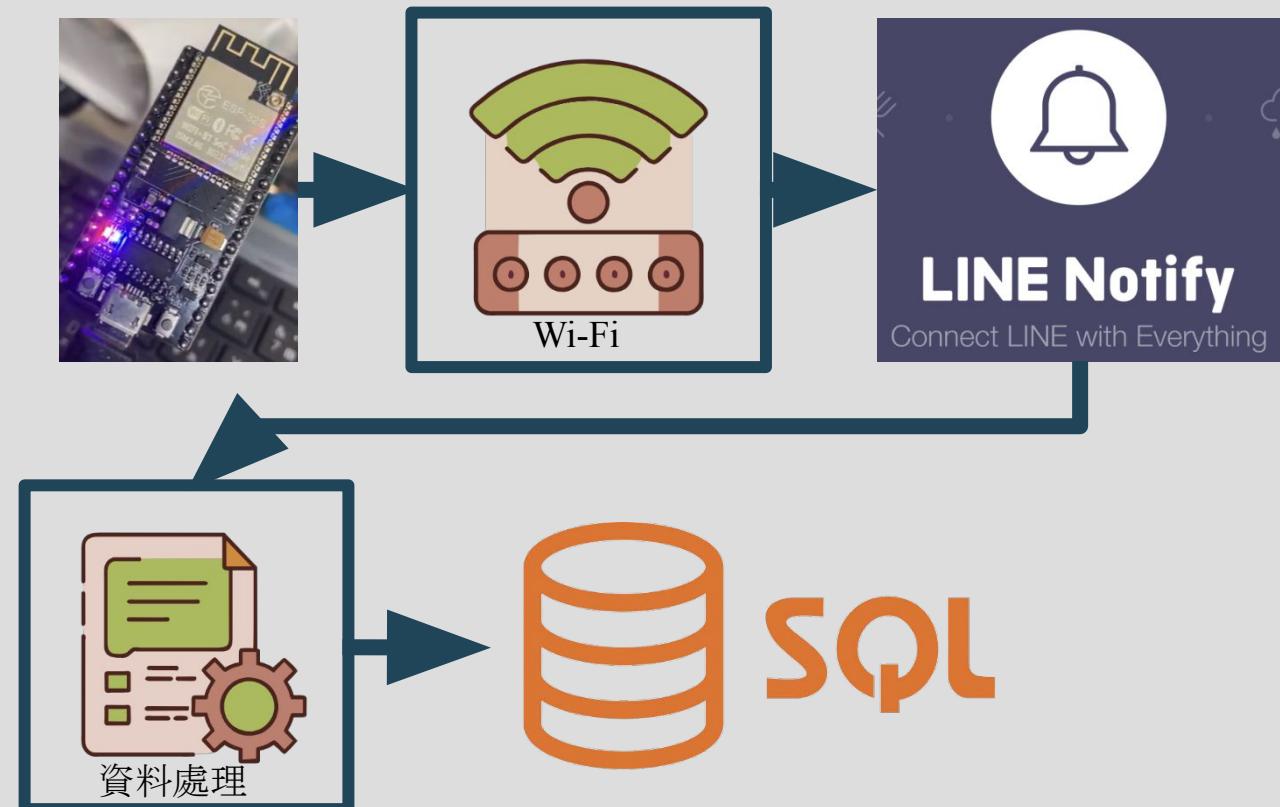
使用Line Notify優點



- 1.能夠快速得知裝置周圍天氣數據
- 2.方便維護
- 3.可將訊息打包成 TXT 檔 或 串接Script



Line Notify 畫面

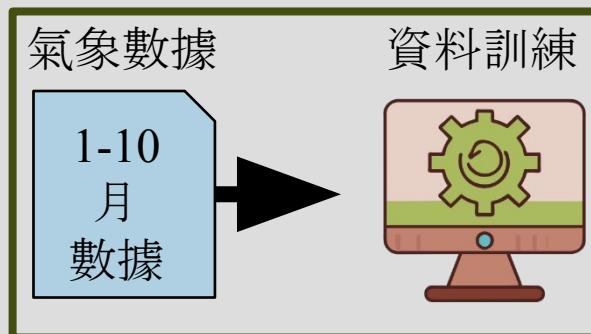


訊息傳遞示意圖

# 比賽方

## 比賽規則

	題目 : 使用微氣候資料預測當天太陽光電發電量
文字說明	<p>提供每個裝置每分鐘的氣象數據，利用<u>預測前一天至預測當天早上9:00之前</u>的各項數據，預測當天9:00之後的每10分鐘發電瓦數，持續預測直至當天晚上5:00，共有48筆預測。</p> <p>※預測日的下一天需要跳過，從預測日往後算的第二天起算為新的一次 ※部分裝置在光照度低於20勒克斯時會進入休眠模式(約為晚上5:00~5:30)</p>
排名計算	將每一筆資料(為每10分鐘的平均瓦數)與正確資料相比並計算誤差(取絕對值)，最後將所有誤差加總在一起， <b>加總誤差值越小名次排序則越前面</b> 。



公開資料訓練示意圖

時間 (整點)	9/1	9/2	9/3	9/4	9/5	9/6	10/27	10/28
07~09	提供	提供						
09~17	預測	預測						

正式比賽階段預測示意圖



# 比賽資料數 據

資料數據	
裝置數量	約20台
資料月份	113/01 ~ 113/11
資料時序	每分鐘
提供數據	<ol style="list-style-type: none"><li>風速(m/s)</li><li>大氣壓力(百帕)</li><li>溫度(攝氏)</li><li>相對濕度(百分比)</li><li>光照度(勒克斯)</li><li>發電量(瓦特)</li></ol>
氣象局資料(每小時)	使用者自行網路下載



# 競賽獎勵方式(以網頁公告為準)

## 教育部獎項(含獎狀和獎金)

獎項名稱	名額	獎金
金牌	1	6萬元整
銀牌	1	4萬元整
銅牌	1	3萬元整
優等	2	2萬元整
佳作	8	1萬元整

## 計畫辦公室獎項(僅電子獎狀)

## 趨勢科技贊助獎項

獎項名稱	名額	獎金
創意獎	5	2萬元整

- Private Score超過Baseline (Final Score > 0.?) 之隊伍(Top 25%), 依規定繳交報告後經主辦單位之評審委員審定, 將獲頒教育部人工智慧競賽計畫辦公室電子獎狀。



# 競賽時程規劃(以網頁公告為準)

## 競賽

日期	事件	
2024/9/2– 2024/11/20	報名時程	
項目	時程	注意事項與說明
報名時程	2024/9/2– 2024/11/20	開放報名
競賽訓練集 Training Dataset 下載	2024/9/23 11:00 AM – 2024/11/28	競賽隊伍可下載競賽訓練集 Training Dataset
比賽正式開始 競賽訓練集 Training Dataset 更新、競賽測試集 Test Dataset 下載與訓練模型預測	2024/11/18 11:00 AM – 2024/11/28	<ul style="list-style-type: none"><li>此期間預測結果每日提交之上限為5次，每日上傳次數以檔案計算。</li><li>競賽者需於規定時間內上傳檔案，答案檔案需依照規定之格式，以避免上傳失敗的問題。</li></ul>
結果發佈	2024/12/2 11:00 AM	公佈 Private Leaderboard 成績
上傳報告	2024/12/2 – 2024/12/9	優勝隊伍提交預測模型的說明文件、自製之訓練資料集、與程式碼
公佈最後名次	2025/1/7 16:00 AM	公佈本次比賽的最終名次
頒獎典禮	2025年初(暫訂第一季)	頒獎典禮將與初階賽及進階賽一同辦理，細節將另行公佈





**AI CUP  
2024**

教育部大專校院人工智慧競賽

**讓綠能觸手可及 -根據區域微氣候資料預測發電量**

主持人:東華大學資工系 顏士淨教授

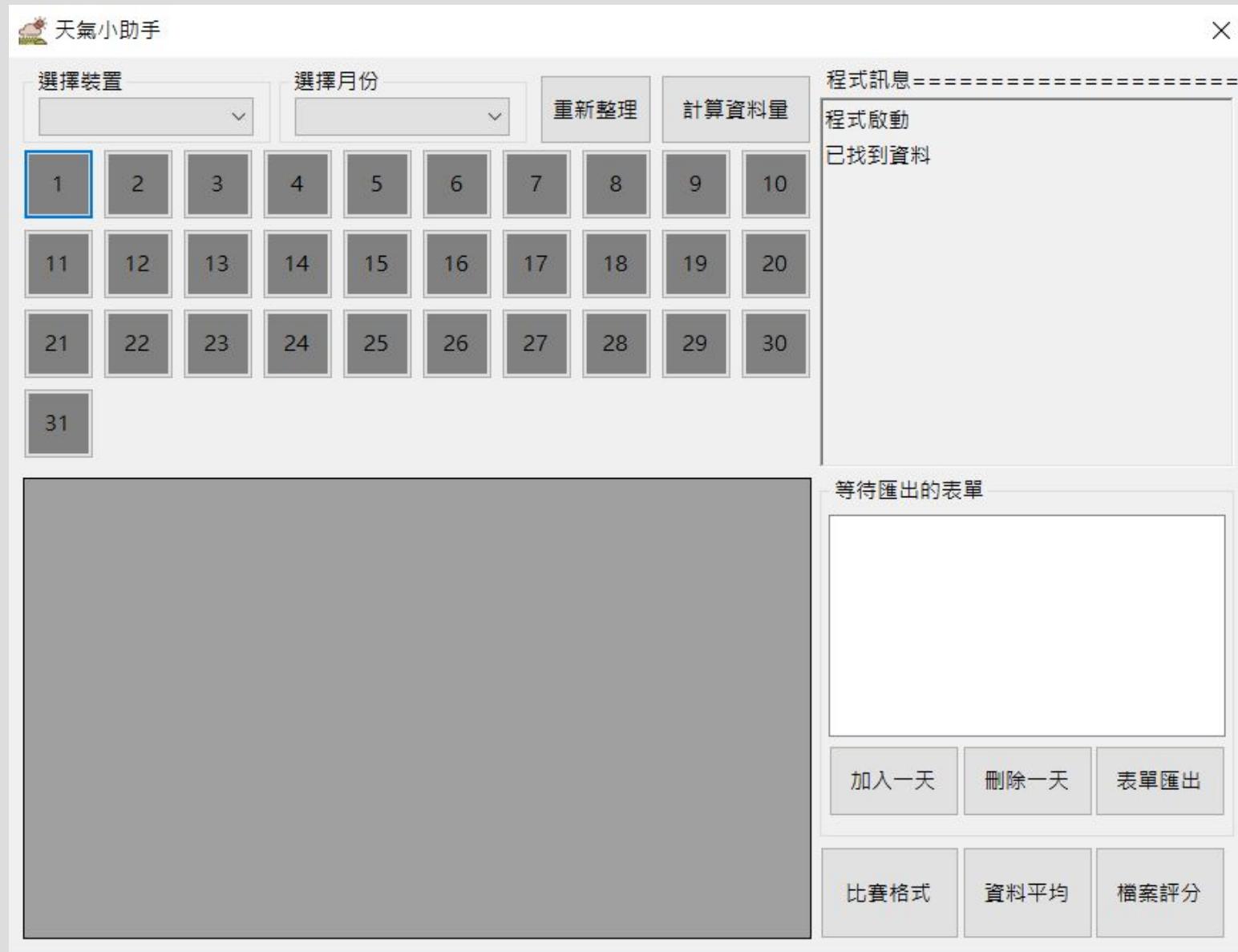
協同主持人:東華大學光電系 白益豪教授

助理:東華大學資工系彭英茜

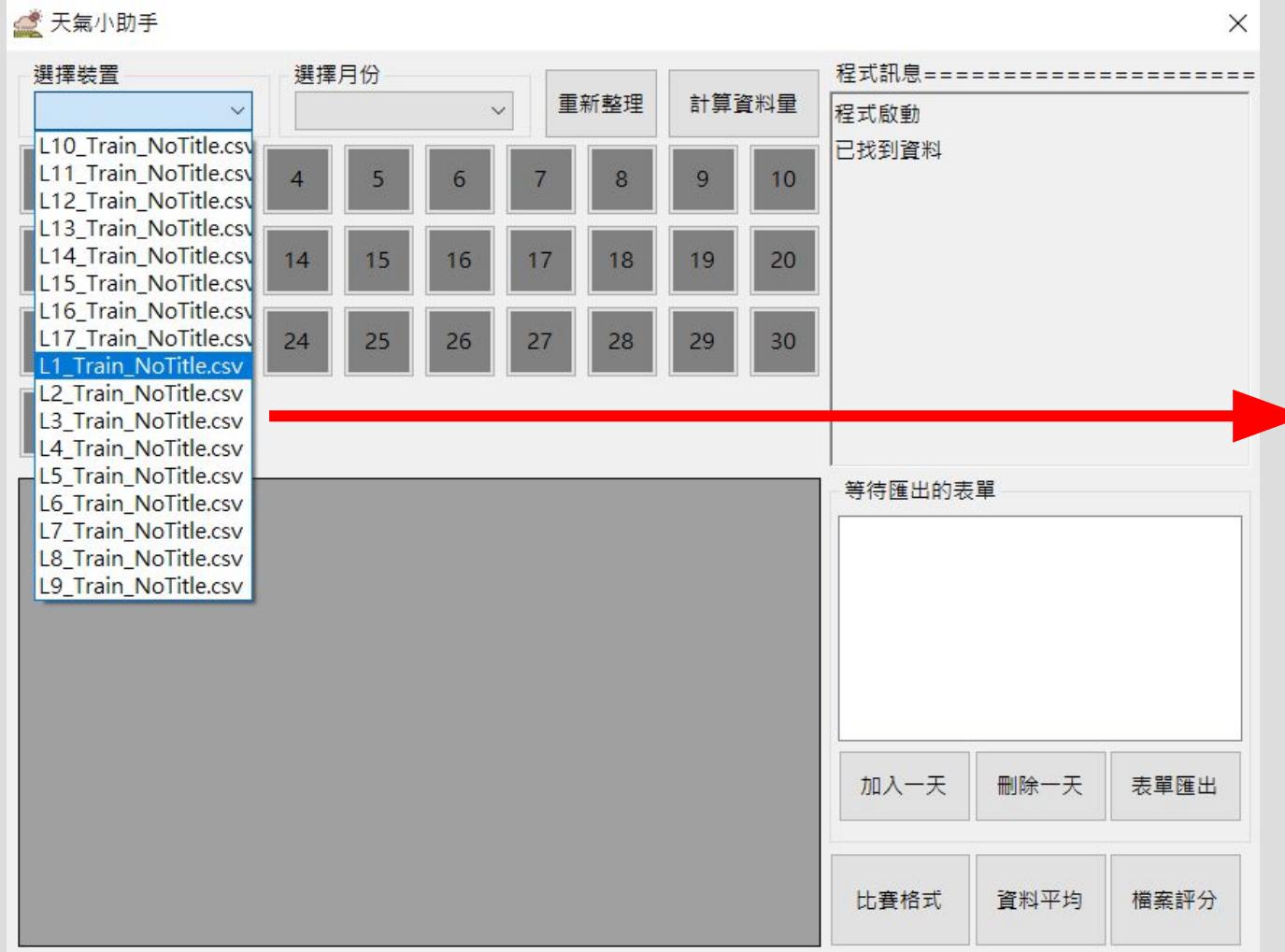
兼任助理:古李傑驛、黃聖文



# 微氣候數據處理-程式介紹



# 微氣候數據處理-選擇裝置



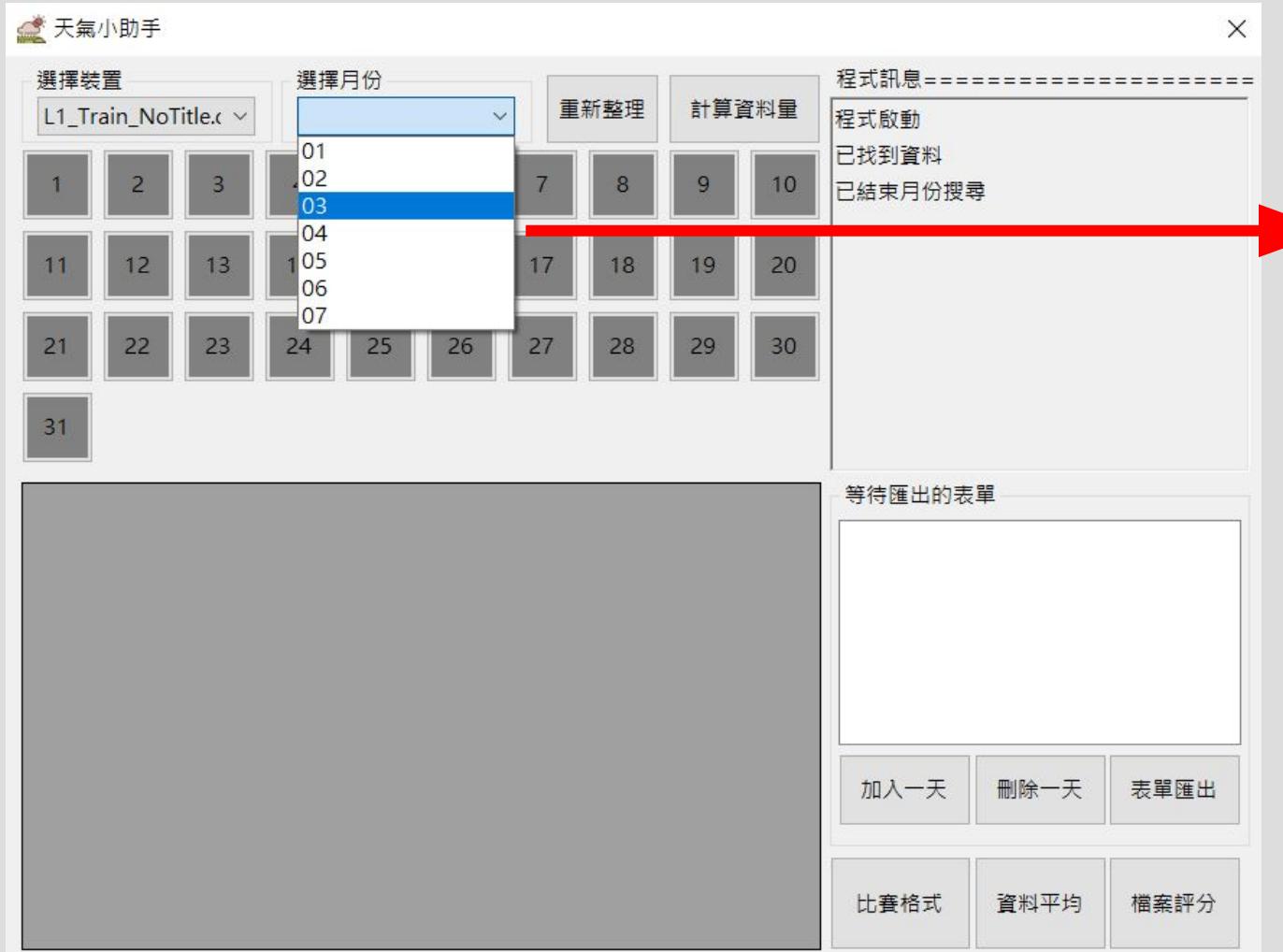
透過「選擇裝置」的下拉式選單，  
選擇要查看的裝置。

**\*文件不能有標題**

Use the "選擇裝置" dropdown menu to  
choose the device you want to view.

**\*The document must not contain headers.**

# 微氣候數據處理-選擇月份



透過「選擇月份」的下拉式選單，  
選擇要查看的月份。

Use the "Select Month" dropdown menu to choose  
the month you want to view.

# 微氣候數據處理-選擇日期

天氣小助手

選擇裝置 L1\_Train\_NoTitle.c  
選擇月份 03  
重新整理 計算資料量

程式訊息-----  
程式啟動  
已找到資料  
已結束月份搜尋  
已結束日期搜尋  
顯示日期: 2024-03-17

Location Code	DateTime	WindSpeed(m/s)	Pressure(hpa)
1	2024-03-17 05...	0	1011.77
1	2024-03-17 05...	0	1011.8
1	2024-03-17 05...	0	1011.82
1	2024-03-17 05...	0	1011.86
1	2024-03-17 05...	0	1011.84
1	2024-03-17 05...	0	1011.82
1	2024-03-17 05...	0	1011.87
1	2024-03-17 05...	0	1011.83
1	2024-03-17 05...	0	1011.83

等待匯出的表單  
加入一天 刪除一天 表單匯出  
比賽格式 資料平均 檔案評分

The screenshot shows a weather data processing application. At the top, there's a toolbar with '選擇裝置' (Select Device) set to 'L1\_Train\_NoTitle.c', '選擇月份' (Select Month) set to '03', '重新整理' (Refresh), and '計算資料量' (Calculate Data Volume). Below the toolbar is a grid of 31 boxes, each representing a day of the month. Days 1, 3, 11, 13, 21, 22, 23, 24, 26, 27, 28, 29, and 31 are orange, indicating complete data collection. Days 2, 4, 5, 6, 7, 8, 9, 10, 12, 14, 15, 16, 18, 19, 20, and 30 are green, indicating incomplete data collection. Days 17 and 25 are gray, indicating no data. A red arrow points from the green box for day 8 to the explanatory text. Another red arrow points from the orange box for day 17 to the explanatory text. A third red arrow points from the gray box for day 25 to the explanatory text. To the right of the grid, a message box displays '程式訊息-----', '程式啟動', '已找到資料', '已結束月份搜尋', '已結束日期搜尋', and '顯示日期: 2024-03-17'. Below the grid is a table for March 17, 2024, with columns for Location Code, DateTime, WindSpeed(m/s), and Pressure(hpa). The table shows 10 rows of data. At the bottom right are buttons for '等待匯出的表單' (Pending Export Form) labeled '加入一天' (Add Day), '刪除一天' (Delete Day), '表單匯出' (Form Export), and other options like '比賽格式' (Competition Format), '資料平均' (Data Average), and '檔案評分' (File Score).

綠色方框代表著資料「不完整」收集的一天

The green box represents a day where data was "incompletely" collected.

橘色方框代表著資料「完整」收集的一天

The orange box represents a day where data was "completely" collected.

灰色方框代表著無資料

The gray box represents a day with no data.

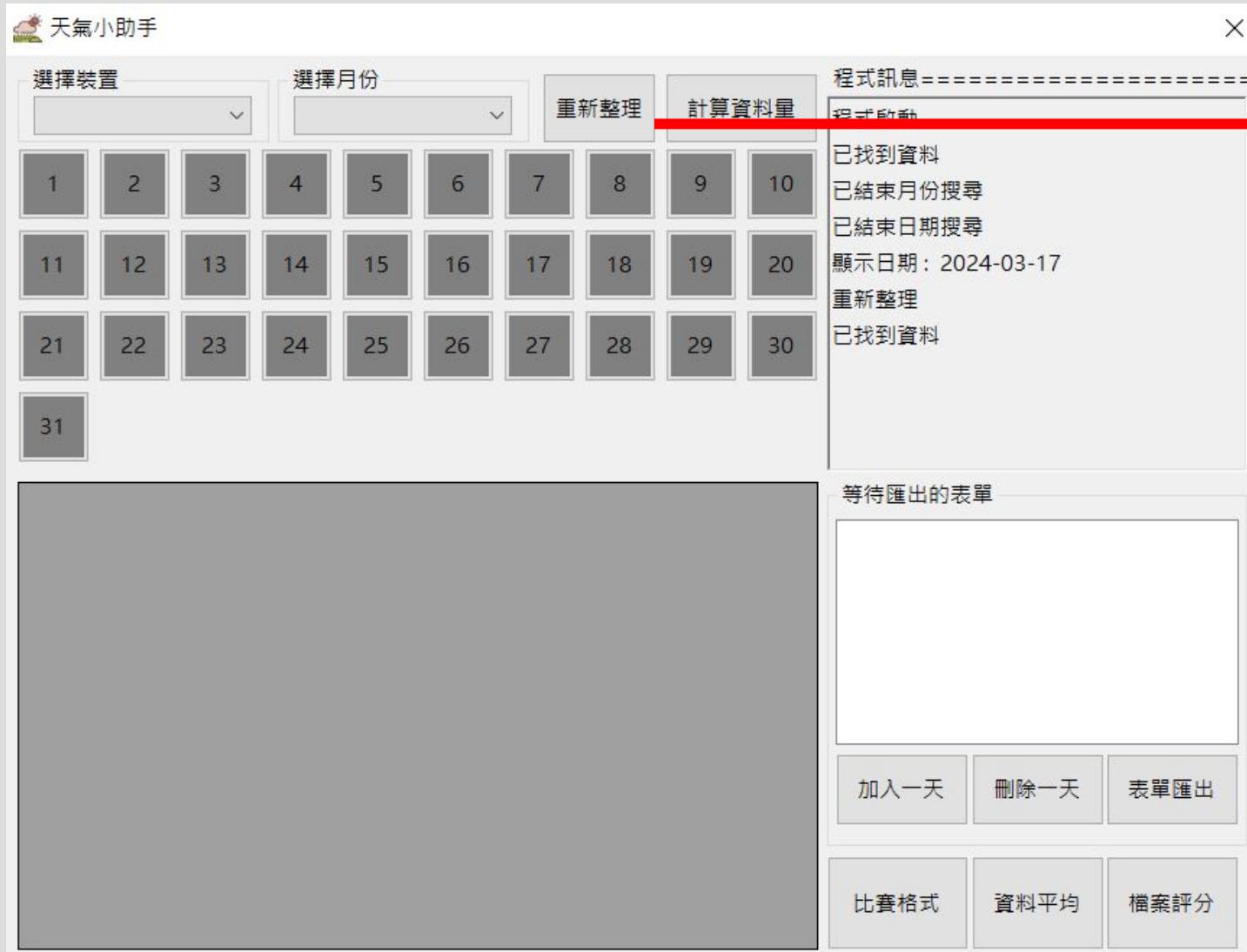
當方框被點擊後，可顯示當天資料

When the box is clicked, the data for that day will be displayed.

\*「完整」收集的一天：此天資料包含著 7:00~16:59(含)，且每10分鐘都至少會有一筆氣候數據

\*「無完整」收集的一天：此天有收集到資料，但氣候數據有遺失

# 微氣候數據處理-重新整理



按下「重新整理」按鈕可清空畫面，並重新  
讀取氣候數據

Clicking the "重新整理" button will clear the  
screen and reload the climate data.

# 微氣候數據處理-計算資料量



按下「計算資料量」按鈕會開始計算資料的完整天數有幾天、不完整的天數有幾天  
**\*計算時間稍久**

Clicking the "計算資料量" button will start calculating how many days have complete data and how many days have incomplete data.  
**\*The calculation may take some time.**

# 微氣候數據處理-加入一天

天氣小助手

選擇裝置	選擇月份	重新整理	計算資料量						
L1_Train_NoTitle.c	03	重新整理	計算資料量						
1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31									

程式訊息=====

已結束日期搜尋  
顯示日期：2024-01-02  
成功加入表單  
顯示日期：2024-01-28  
成功加入表單  
已結束日期搜尋  
顯示日期：2024-03-26  
成功加入表單  
顯示日期：2024-03-30  
成功加入表單

等待匯出的表單

L1\_Train\_NoTitle.csv,2024-01-02  
L1\_Train\_NoTitle.csv,2024-01-28  
L1\_Train\_NoTitle.csv,2024-03-26  
L1\_Train\_NoTitle.csv,2024-03-30  
目前有：4 筆

加入一天 刪除一天 表單匯出 比賽格式 資料平均 檔案評分

Location Code	DateTime	WindSpeed(m/s)	Pressure(hpa)
1	2024-03-30 05...	0	1009.93
1	2024-03-30 05...	0	1010.02
1	2024-03-30 05...	0	1010
1	2024-03-30 05...	0	1010.07
1	2024-03-30 05...	0	1010.03
1	2024-03-30 05...	0	1010.02
1	2024-03-30 05...	0	1010.04
1	2024-03-30 05...	0	1010.07
1	2024-03-30 05...	0	1010.05

先點擊方框，再按下「加入一天」按鈕會將資料記錄到表單之中，才可以進行表單匯出  
**\*可跨越月份選擇，但不可跨裝置選擇**

First, click on a box, then press the "加入一天" button to record the data into the form, allowing the form to be exported.

**\*You can select across months, but not across devices.**

# 微氣候數據處理-刪除一天

等待匯出的表單

L1_Train_NoTitle.csv,2024-01-02
L1_Train_NoTitle.csv,2024-01-28
<b>L1_Train_NoTitle.csv,2024-03-26</b>
L1_Train_NoTitle.csv,2024-03-30

目前有 : 4 筆

加入一天    刪除一天    表單匯出

程式訊息=====

成功加入表單  
顯示日期: 2024-01-28

成功加入表單  
已結束日期搜尋  
顯示日期: 2024-03-26

成功加入表單  
顯示日期: 2024-03-30

成功加入表單  
刪除一筆資料: L1  
\_Train\_NoTitle.csv,2024-03-26

等待匯出的表單

L1_Train_NoTitle.csv,2024-01-02
L1_Train_NoTitle.csv,2024-01-28
L1_Train_NoTitle.csv,2024-03-30

目前有 : 3 筆

加入一天    **刪除一天**    表單匯出

比賽格式    資料平均    檔案評分

要使用刪除功能，需先點擊表單中要刪除的列，再按下「刪除一天」的按鈕，即可成功刪除

To use the delete function, first click on the row in the form that you want to delete, then press the "刪除一天" button to successfully remove it.

# 微氣候數據處理-表單匯出

程式訊息=====

2024x09x18T02x32x21Z.csv  
=====

「有」在表單裡的數據：  
[REDACTED] 微氣候數據處理  
\source\DataInList\_L1\_Train\_NoTitle\_2024x09x18T02x32x21Z.csv  
=====

「未」在表單裡的數據：  
[REDACTED] 微氣候數據處理  
\source\DataNotInList\_L1

等待匯出的表單

加入一天 刪除一天 表單匯出

比賽格式 資料平均 檔案評分

SolarData	2024/9/18 上午 12:34	檔案資料夾
DataInList_L1_Train_NoTitle_2024x09x18T02x32x21Z.csv	2024/9/18 上午 02:32	CSV 檔案 96 KB
DataNotInList_L1_Train_NoTitle_2024x09x18T02x32x21Z.csv	2024/9/18 上午 02:32	CSV 檔案 5,929 KB
ListLog_2024x09x18T02x32x21Z.csv	2024/9/18 上午 02:32	CSV 檔案 1 KB
WeatherAssistant.deps.json	2024/9/4 上午 02:00	JSON File 1 KB
WeatherAssistant.dll	2024/9/9 上午 04:09	應用程式擴充 59 KB
WeatherAssistant.exe	2024/9/9 上午 04:09	應用程式 140 KB
WeatherAssistant.pdb	2024/9/9 上午 04:09	Program Debug ... 33 KB
WeatherAssistant.runtimeconfig.json	2024/9/3 下午 11:41	JSON File 1 KB

其中有三個檔案，第 1 個檔案是僅有再表單出現過的天數，且資料時間是從9:00至當天收集結束為止；第 2 個檔案是取出指定數據後剩下的氣候數據；第 3 個檔案則是記錄匯出時所選擇的天數

There are three files :

1. The first file contains only the days that appeared in the form, with data ranging from 9:00 to the end of the day's collection.
2. The second file consists of the remaining climate data after the specified data has been extracted.
3. The third file records the days selected at the time of export.

# 微氣候數據處理-更新資料

source	2024/9/18 上午 02:32	檔案資料夾
dotnet-sdk-8.0.401-win-x64.exe	2024/9/4 上午 02:02	應用程式
ReadMe_說明文件.txt	2024/9/8 上午 01:19	文字文件
執行檔	2024/9/18 上午 12:38	捷徑
數據庫	2024/9/18 上午 12:38	捷徑

SolarData	2024/9/18 上午 03:55	檔案資料夾
DataInList_L1_Train_NoTitle_2024x09x...	2024/9/18 上午 02:32	CSV 檔案
DataNotInList_L1_Train_NoTitle_2024x...	2024/9/18 上午 02:32	CSV 檔案
ListLog_2024x09x18T02x32x21Z.csv	2024/9/18 上午 02:32	CSV 檔案
WeatherAssistant.deps.json	2024/9/4 上午 02:00	JSON File
WeatherAssistant.dll	2024/9/9 上午 04:09	應用程式擴充
WeatherAssistant.exe	2024/9/9 上午 04:09	應用程式
WeatherAssistant.pdb	2024/9/9 上午 04:09	Program Debug ...
WeatherAssistant.runtimeconfig.json	2024/9/3 下午 11:41	JSON File

名稱	修改日期	類型	大小
DataNotInList_L1_Train_NoTitle_2024x...	2024/9/18 上午 02:32	CSV 檔案	5,929 KB
L1_Train_NoTitle.csv	2024/9/5 下午 04:17	CSV 檔案	6,025 KB
L2_Train_NoTitle.csv	2024/9/4 下午 11:49	CSV 檔案	5,427 KB
L3_Train_NoTitle.csv	2024/9/4 下午 11:56	CSV 檔案	3,408 KB
L4_Train_NoTitle.csv	2024/9/4 下午 03:15	CSV 檔案	3,530 KB
L5_Train_NoTitle.csv	2024/9/5 上午 05:03	CSV 檔案	3,478 KB
L6_Train_NoTitle.csv	2024/9/4 下午 11:23	CSV 檔案	3,317 KB
L7_Train_NoTitle.csv	2024/9/4 下午 03:44	CSV 檔案	2,674 KB
L8_Train_NoTitle.csv	2024/9/5 下午 04:55	CSV 檔案	7,020 KB
L9_Train_NoTitle.csv	2024/9/4 下午 05:10	CSV 檔案	5,414 KB
L10_Train_NoTitle.csv	2024/9/5 下午 04:37	CSV 檔案	5,636 KB
L11_Train_NoTitle.csv	2024/9/5 上午 04:46	CSV 檔案	3,457 KB
L12_Train_NoTitle.csv	2024/9/5 上午 04:37	CSV 檔案	5,020 KB
L13_Train_NoTitle.csv	2024/9/5 上午 04:55	CSV 檔案	4,325 KB
L14_Train_NoTitle.csv	2024/9/4 下午 04:24	CSV 檔案	4,130 KB
L15_Train_NoTitle.csv	2024/9/4 下午 04:48	CSV 檔案	5,274 KB
L16_Train_NoTitle.csv	2024/9/5 上午 05:13	CSV 檔案	5,135 KB

更新資料時，可以透過「數據庫」(捷徑) 或是 進入「source」資料夾，找到「SolarData」資料夾，把新的氣候數據存入

When updating data, you can use the "數據庫" (shortcut) or enter the "source" folder, find the "SolarData" folder, and save the new climate data there.

# 微氣候數據處理-更新資料

The image displays two side-by-side screenshots of a Windows application titled "Micro Climate Data Processing - Update Data". Both screenshots show a grid of numbered buttons (1-31) representing days of the month, with colors indicating data status. A "Reorganize" button is present in both.

**Screenshot 1 (Left):**

- Top Bar:** "選擇裝置" dropdown set to "L1\_Train\_NoTitle.c", "選擇月份" dropdown set to "01", "重新整理" (Reorganize) button, and "計算資料量" (Calculate Data Volume) button.
- Message Area:** "程式訊息" (Program Message) section shows log entries: "已找到資料" (Data found), "重新整理" (Reorganize), "已找到資料" (Data found), "已結束月份搜尋" (Month search completed), "已結束日期搜尋" (Date search completed), "顯示日期: 2024-01-02" (Display date: 2024-01-02).
- Data Grid:** Shows a 6x5 grid of buttons. Rows 1-5 have columns 1-4 orange, 5 green, 6 orange, 7-10 green. Row 6 has columns 1-5 green, 6 orange, 7-10 green. Column 6 contains buttons for days 11-31.
- Table:** A table titled "等待匯出的表單" (Waiting for export table) with columns "Location Code", "DateTime", and "WindSpeed(m/s)". It lists data from January 2nd, 2024, at 08:59:05.000 to 09:07:05.000. The row for 09:00:05.000 is highlighted in blue.
- Buttons:** "加入一天" (Add one day), "刪除一天" (Delete one day), "表單匯出" (Table export).
- Bottom Buttons:** "比賽格式" (Competition format), "資料平均" (Average data), "檔案評分" (File score).

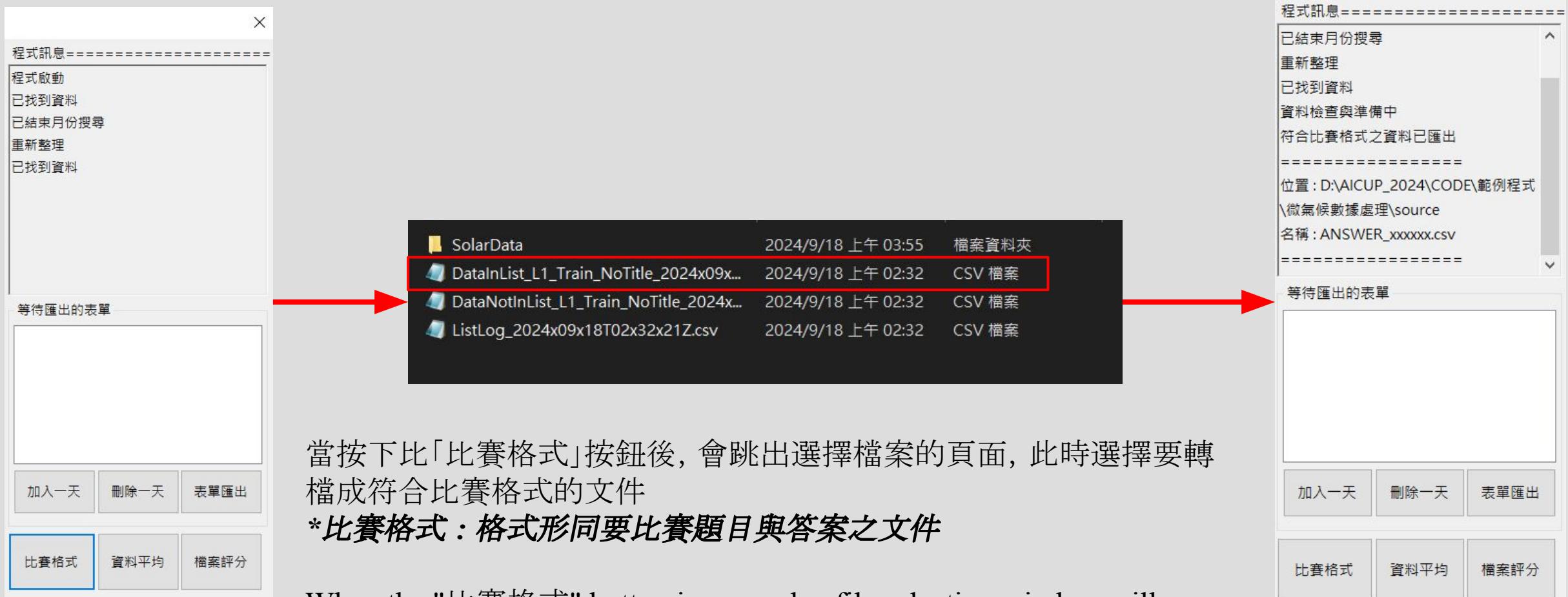
**Screenshot 2 (Right):**

- Top Bar:** "選擇裝置" dropdown set to "DataNotInList\_L1\_1", "選擇月份" dropdown set to "01", "重新整理" button, and "計算資料量" button.
- Message Area:** "程式訊息" (Program Message) section shows log entries: "已找到資料" (Data found), "已結束月份搜尋" (Month search completed), "已結束日期搜尋" (Date search completed), "顯示日期: 2024-01-02" (Display date: 2024-01-02).
- Data Grid:** Similar to Screenshot 1, showing a 6x5 grid of buttons with colors indicating data status.
- Table:** A table titled "等待匯出的表單" (Waiting for export table) with columns "Location Code", "DateTime", "WindSpeed(m/s)", and "Pre". It lists data from January 2nd, 2024, at 08:52:05.000 to 08:59:05.000. The row for 08:59:05.000 is highlighted in blue.
- Buttons:** "加入一天" (Add one day), "刪除一天" (Delete one day), "表單匯出" (Table export).
- Bottom Buttons:** "比賽格式" (Competition format), "資料平均" (Average data), "檔案評分" (File score).

當按下「重新整理」或是重新開啟程式時，新的氣候數據就會被載入；與此同時可以看到被取出的橘色方框資料變成了綠色方塊，資料的時間也是從9:00開始被取出

When the "重新整理" button is pressed or the program is reopened, the new climate data will be loaded. At the same time, the previously extracted orange box data will turn into green boxes, with the data extraction starting from 9:00.

# 微氣候數據處理-比賽格式



當按下比「比賽格式」按鈕後，會跳出選擇檔案的頁面，此時選擇要轉檔成符合比賽格式的文件

**\*比賽格式：格式形同要比賽題目與答案之文件**

When the "比賽格式" button is pressed, a file selection window will appear. At this point, choose the file you want to convert to match the competition format.

**\*Competition Format: The format should match the document containing both the competition questions and answers.**

# 微氣候數據處理-比賽格式

The screenshot shows the application's user interface with two main panes. On the left, a log window displays the progress of the conversion process, including the output file path (D:\AICUP\_2024\CODE\範例程式\微氣候數據處理\source\ANSWER\_xxxxxx.csv) and a section for pending export tables. On the right, a file explorer window shows the generated files: SolarData (CSV, 4 KB), ANSWER\_01\_2024x09x18T14x55x48Z.csv (CSV, 96 KB), DataInList\_L1\_Train\_NoTitle\_2024x09x... (CSV, 5,929 KB), ListLog\_2024x09x18T02x32x21Z.csv (CSV, 1 KB), WeatherAssistant.deps.json (JSON File, 1 KB), WeatherAssistant.dll (Application Extension, 59 KB), WeatherAssistant.exe (Application, 140 KB), WeatherAssistant.pdb (Program Debug, 33 KB), and WeatherAssistant.runtimeconfig.json (JSON File, 1 KB). Red arrows point from the 'ANSWER\_xxxxxx.csv' entry in the log to the generated CSV file in the file explorer, and from the generated CSV file to the detailed list of files.

程式訊息=====

已結束月份搜尋  
重新整理  
已找到資料  
資料檢查與準備中  
符合比賽格式之資料已匯出  
=====

位置 : D:\AICUP\_2024\CODE\範例程式  
\微氣候數據處理\source  
名稱 : ANSWER\_xxxxxx.csv  
=====

等待匯出的表單

加入一天 刪除一天 表單匯出

比賽格式 資料平均 檔案評分

ANSWER\_01\_2024x09x18T14x55x48Z.csv - 記事本

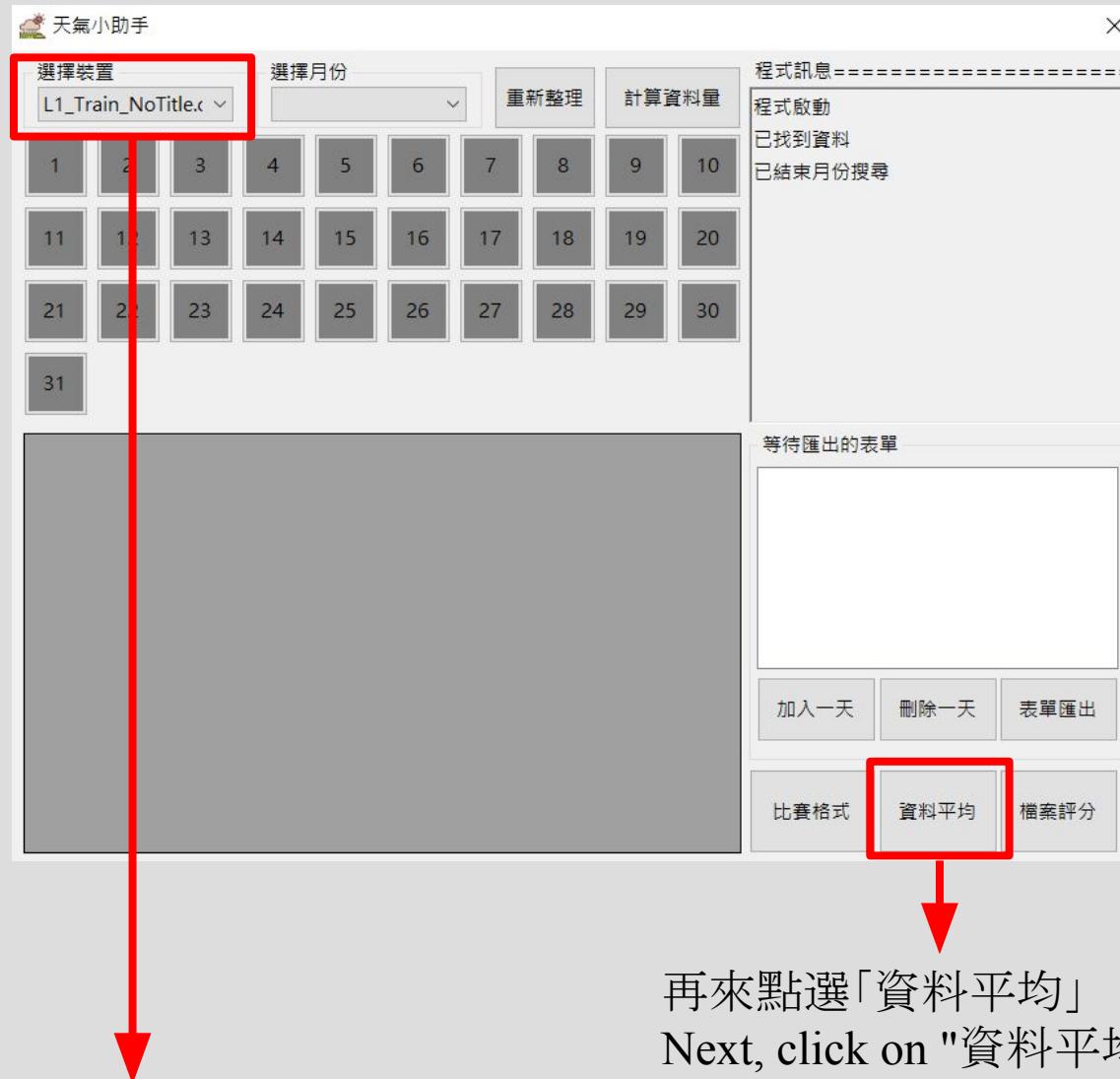
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

ANSWER_01_2024x09x18T14x55x48Z.csv	2024/09/18 上午 03:55	檔案	4 KB
DataInList_L1_Train_NoTitle_2024x09x...	2024/09/18 上午 02:32	CSV 檔案	96 KB
DataNotInList_L1_Train_NoTitle_2024x...	2024/09/18 上午 02:32	CSV 檔案	5,929 KB
ListLog_2024x09x18T02x32x21Z.csv	2024/09/18 上午 02:32	CSV 檔案	1 KB
WeatherAssistant.deps.json	2024/09/04 上午 02:00	JSON File	1 KB
WeatherAssistant.dll	2024/09/09 上午 04:09	應用程式擴充	59 KB
WeatherAssistant.exe	2024/09/09 上午 04:09	應用程式	140 KB
WeatherAssistant.pdb	2024/09/09 上午 04:09	Program Debug ...	33 KB
WeatherAssistant.runtimeconfig.json	2024/09/03 下午 11:41	JSON File	1 KB

資料轉檔後，會生成在「source」資料夾內  
**\*生成的資料未包含標題**

After the data is converted, it will be generated in the "source" folder.  
**\*The generated data will not include headers.**

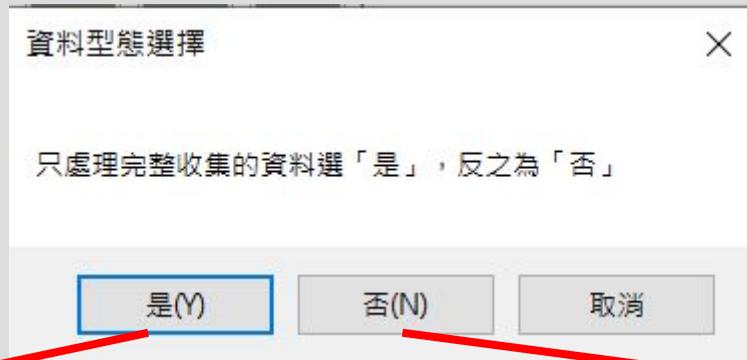
# 微氣候數據處理-資料平均



先選擇要平均處理的裝置

First, select the device you want to process for averaging.

# 微氣候數據處理-資料平均



選擇「是」，代表只處理完整一天的資料，取出9:00 - 16:59，並按照本次比賽之平均規則，製作出新的氣候數據檔案  
Selecting "Y" means only processing data for complete days, extracting data from 9:00 to 16:59, and generating a new climate data file according to the averaging rules of the current competition.

選擇「否」，代表會將資料取出7:00 - 8:59，並按照本次比賽之平均規則，製作出新的氣候數據檔案  
Selecting "N" means that the data from 7:00 to 8:59 will be extracted, and a new climate data file will be generated according to the averaging rules of the current competition.

AvgDATA_01_2024x09x18T16x18x40Z.csv - 記事本	
檔案(F)	編輯(E)
格式(O)	檢視(V) 說明
20240101090001, 1.92, 1017.91, 18.33, 92.29, 8395.25, 17.37	
20240101091001, 0.09, 1017.96, 18.02, 97.37, 7046.5, 12.61	
20240101092001, 0.02, 1018.16, 17.73, 99.93, 5919.58, 8.81	
20240101093001, 0.02, 1018.2, 17.9, 100, 8838.92, 17.56	
20240101094001, 0.04, 1018.13, 18.08, 100, 5774.67, 7.51	
20240101095001, 0.02, 1018.03, 17.83, 100, 5437.75, 7.03	
20240101100001, 0, 1018.04, 17.75, 100, 9312.75, 19.71	
20240101101001, 0.04, 1017.87, 17.99, 100, 9500.67, 20.35	
20240101102001, 0.02, 1017.82, 18.13, 100, 9711.58, 20.43	
20240101103001, 0.01, 1017.82, 18.2, 100, 7952.5, 14.01	
20240101104001, 0.58, 1017.71, 18.11, 95.79, 6422.5, 9.33	
20240101105001, 0.03, 1017.55, 18.21, 93.92, 7941.42, 13.53	
20240101110001, 0.2, 1017.46, 18.64, 92.71, 7700.92, 13.2	

# 微氣候數據處理-檔案評分



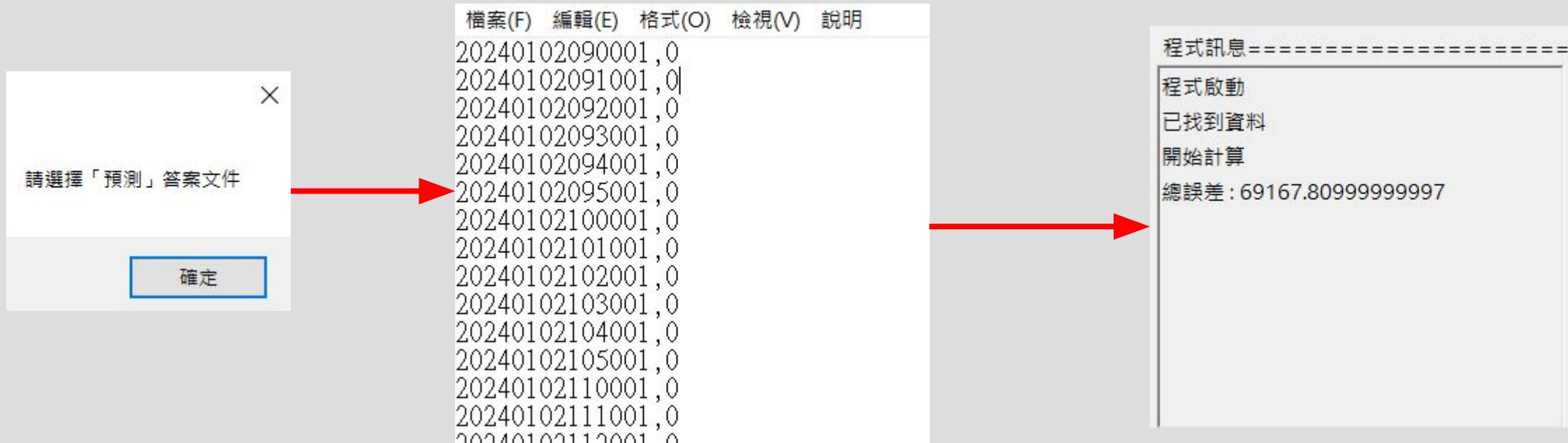
使用「檔案評分」功能，請先選擇標準答案文件

\*匯入檔案時記得把標題刪掉，標準答案文件格式應該是  序號,答案

To use the "File Scoring" function, please first select the standard answer document.

\*When importing the file, remember to remove the headers. The standard answer file format should be: serial number, answer.

# 微氣候數據處理-檔案評分



再來選預測答案文件，選擇完後，系統會給出分數

\*匯入檔案時記得把標題刪掉，預測答案文件格式應該是  序號,答案

\*分數僅供參考

Next, select the predicted answer document. After making your selection, the system will provide a score.

\*When importing the file, remember to remove the headers. The predicted answer file format should be: serial number, answer.

\*The score is for reference only.

# 微氣候數據處理-檔案評分

名稱	修改日期	類型	大小
SolarData	2024/9/18 上午 03:55	檔案資料夾	
ANSWER_01_2024x09x18T14x55x48Z...	2024/9/18 下午 02:55	CSV 檔案	4 KB
ANSWER_USER.csv	2024/9/18 下午 05:39	CSV 檔案	3 KB
AvgDATA_01_2024x09x18T16x18x40Z...	2024/9/18 下午 04:18	CSV 檔案	317 KB
DataInList_L1_Train_NoTitle_2024x09x...	2024/9/18 上午 02:32	CSV 檔案	96 KB
DataNotInList_L1_Train_NoTitle_2024x...	2024/9/18 上午 02:32	CSV 檔案	5,929 KB
ErrorScore_2024x09x18T17x40x16Z.csv	2024/9/18 下午 05:40	CSV 檔案	4 KB
IncompleteAvgDATA_01_2024x09x18T...	2024/9/18 下午 04:19	CSV 檔案	92 KB
ListLog_2024x09x18T02x32x21Z.csv	2024/9/18 上午 02:32	CSV 檔案	1 KB
WeatherAssistant.deps.json	2024/9/4 上午 02:00	JSON File	1 KB
WeatherAssistant.dll	2024/9/9 上午 04:09	應用程式擴充	59 KB
WeatherAssistant.exe	2024/9/9 上午 04:09	應用程式	140 KB
WeatherAssistant.pdb	2024/9/9 上午 04:09	Program Debug ...	33 KB
WeatherAssistant.runtimeconfig.json	2024/9/3 下午 11:41	JSON File	1 KB

ErrorScore_2024x09x18T17x40x16Z.csv - 記事本	
檔案(F)	編輯(E)
20240102090001	,1277.55
20240102091001	,1369.8
20240102092001	,1462.67
20240102093001	,1540.4
20240102094001	,1594.3
20240102095001	,1631.14
20240102100001	,1659.6
20240102101001	,1671.31
20240102102001	,1674.38
20240102103001	,1669.7
20240102104001	,1689.69
20240102105001	,1708.54
20240102110001	,1710.23
20240102111001	,1734.19
20240102112001	,1727.19
20240102113001	,1760.06

檔案評分完後，也會生成一個記錄個別誤差的文件供使用者核對  
**\*分數僅供參考**

After the file scoring is completed, a file recording individual errors will also be generated for the user to verify.  
**\*The score is for reference only.**

# LSTM氣候數據預測-程式介紹

搭建神經網路的結構

搭建全連接層

搭建LSTM層

搭建正則化層

加載已經訓練好的模型

進行科學運算

進行數據處理

管理文件和目錄

執行儲存格 | 執行更低版本 | 偵錯儲存格

#%%

```
from tensorflow.keras.models import Sequential  
from tensorflow.keras.layers import Dense  
from tensorflow.keras.layers import LSTM  
from tensorflow.keras.layers import Dropout  
from tensorflow.keras.models import load_model
```

```
import numpy as np  
import pandas as pd  
import os
```

Python 版本 : 3.9

tensorflow\_gpu 版本 : 2.6.0

# LSTM氣候數據預測-程式介紹

將 Python 代碼劃分為可執行的「代碼區塊」



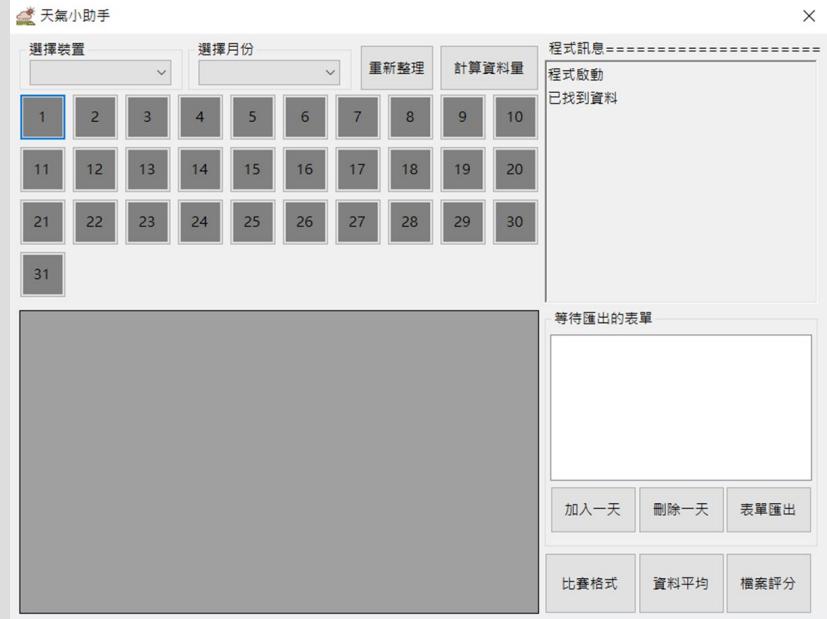
執行儲存格 | 執行更低版本 | 偵錯儲存格

```
#%%
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import LSTM
from tensorflow.keras.layers import Dropout
from tensorflow.keras.models import load_model

import numpy as np
import pandas as pd
import os
```

Python 版本 : 3.9  
tensorflow\_gpu 版本 : 2.6.0

# LSTM氣候數據預測-資料介紹



範例程式用測試資料

範例程式用 平均後的訓練資料

> ExampleTestData  
> ExampleTrainData(AVG)  
> ExampleTrainDataIncompleteAVG  
EasyLSTM\_20240909.py

# LSTM氣候數據預測-資料介紹

範例程式用測試資料

序號	答案
2	20240102090001,1277.55
3	20240102091001,1369.8
4	20240102092001,1462.67
5	20240102093001,1540.4
6	20240102094001,1594.3
7	20240102095001,1631.14
8	20240102100001,1659.6
9	20240102101001,1671.31
10	20240102102001,1674.38
11	20240102103001,1669.7
12	20240102104001,1689.69
13	20240102105001,1708.54
14	20240102110001,1710.23
15	20240102111001,1734.19
16	20240102112001,1727.19

代碼14碼 = 西元年(4碼)+月(2碼)+日(2碼)+時(2碼)+分(2碼)+裝置代碼(2碼)

發電量(四捨五入)

# LSTM氣候數據預測-資料介紹

範例程式用訓練資料  
(9:00 - 當天結束收集為止)

2024-01-01 09:50

2024-01-01 09:59

資料平均

The screenshot shows a file explorer window with the following structure:

- ExampleTestData
  - Title.txt
  - upload.csv
- ExampleTrainData(AVG)
  - AvgDATA\_01.csv (selected)
  - AvgDATA\_02.csv
  - AvgDATA\_03.csv
  - AvgDATA\_04.csv
  - AvgDATA\_05.csv
  - AvgDATA\_06.csv
  - AvgDATA\_07.csv
  - AvgDATA\_08.csv
  - AvgDATA\_09.csv
  - AvgDATA\_10.csv
  - AvgDATA\_11.csv
  - AvgDATA\_12.csv
  - AvgDATA\_13.csv
  - AvgDATA\_14.csv
  - AvgDATA\_15.csv
  - AvgDATA\_16.csv
  - AvgDATA\_17.csv
  - Title.txt
- ExampleTrainData(IncompleteAVG)
- EasyLSTM\_20240909.py

The right pane displays the contents of AvgDATA\_01.csv. A red box highlights the first two rows of the data:

Serial	WindSpeed(m/s)	Pressure(hpa)	Temperature(°C)	Humidity(%)	Sunlight(Lux)	Power(mW)
20240101090001	1.92	1017.91	18.33	92.29	8595.23	17.37
20240101091001	0.09	1017.96	18.02	97.37	7046.5	12.61
20240101092001	0.02	1018.16	17.73	99.93	5919.58	8.81
20240101093001	0.02	1018.2	17.9	100	8838.92	17.56
20240101094001	0.04	1018.13	18.08	100	5774.67	7.51
20240101095001	0.02	1018.03	17.83	100	5437.75	7.03
20240101100001	0	1018.04	17.75	100	9312.75	19.71
20240101101001	0.04	1017.87	17.99	100	9500.67	20.35
20240101102001	0.02	1017.82	18.13	100	9711.58	20.43
20240101103001	0.01	1017.82	18.2	100	7952.5	14.01
20240101104001	0.58	1017.71	18.11	95.79	6422.5	9.33
20240101105001	0.03	1017.55	18.21	93.92	7941.42	13.53
20240101110001	0.2	1017.46	18.64	92.71	7700.92	13.2
20240101110001	0.35	1017.28	18.78	87.1	6558.92	9.47
20240101112001	0.25	1017.09	18.8	86.15	6868.25	10.26
20240101113001	0.17	1016.87	18.94	84.1	8485.08	15.88
20240101114001	0.06	1016.56	19.38	84.1	10389.83	22.82
20240101115001	0.15	1016.36	19.77	80.9	9784.75	20.51
20240101120001	0.43	1016.17	19.87	80.09	11379	27.05
20240101121001	0.97	1015.97	19.96	77.42	9266.17	18.22
20240101122001	0.4	1015.82	19.81	78	9845.92	20.46
20240101123001	1.27	1015.67	20.04	76.07	16415.58	55.61
20240101124001	0.75	1015.51	20.35	74.86	13374.75	37.65
20240101125001	1.37	1015.42	20.29	74.1	10362.5	23.23
20240101130001	0.39	1015.34	19.9	75.45	7872.25	13.66
20240101131001	0.14	1015.27	19.88	77.24	6294.5	8.64

平均後的檔案格式 □ 代碼14碼,風速,大氣壓力,溫度,濕度,光照度,發電量

# LSTM氣候數據預測-資料介紹

範例程式用訓練資料  
(7:00 – 8:59)

2024-01-01 07:50

|  
2024-01-01 07:59  
資料平均

The screenshot shows a file explorer window with the following structure:

- ExampleTrainData (folder)
  - Title.txt
  - upload.csv
- ExampleTrainData(AVG) (folder)
- ExampleTrainData(IncompleteAVG) (folder)
  - IncompleteAvgDATA\_01.csv (selected, highlighted in blue)
  - IncompleteAvgDATA\_02.csv
  - IncompleteAvgDATA\_03.csv
  - IncompleteAvgDATA\_04.csv
  - IncompleteAvgDATA\_05.csv
  - IncompleteAvgDATA\_06.csv
  - IncompleteAvgDATA\_07.csv
  - IncompleteAvgDATA\_08.csv
  - IncompleteAvgDATA\_09.csv
  - IncompleteAvgDATA\_10.csv
  - IncompleteAvgDATA\_11.csv
  - IncompleteAvgDATA\_12.csv
  - IncompleteAvgDATA\_13.csv
  - IncompleteAvgDATA\_14.csv
  - IncompleteAvgDATA\_15.csv
  - IncompleteAvgDATA\_16.csv
  - IncompleteAvgDATA\_17.csv
  - Title.txt
- EasyLSTM\_20240909.py

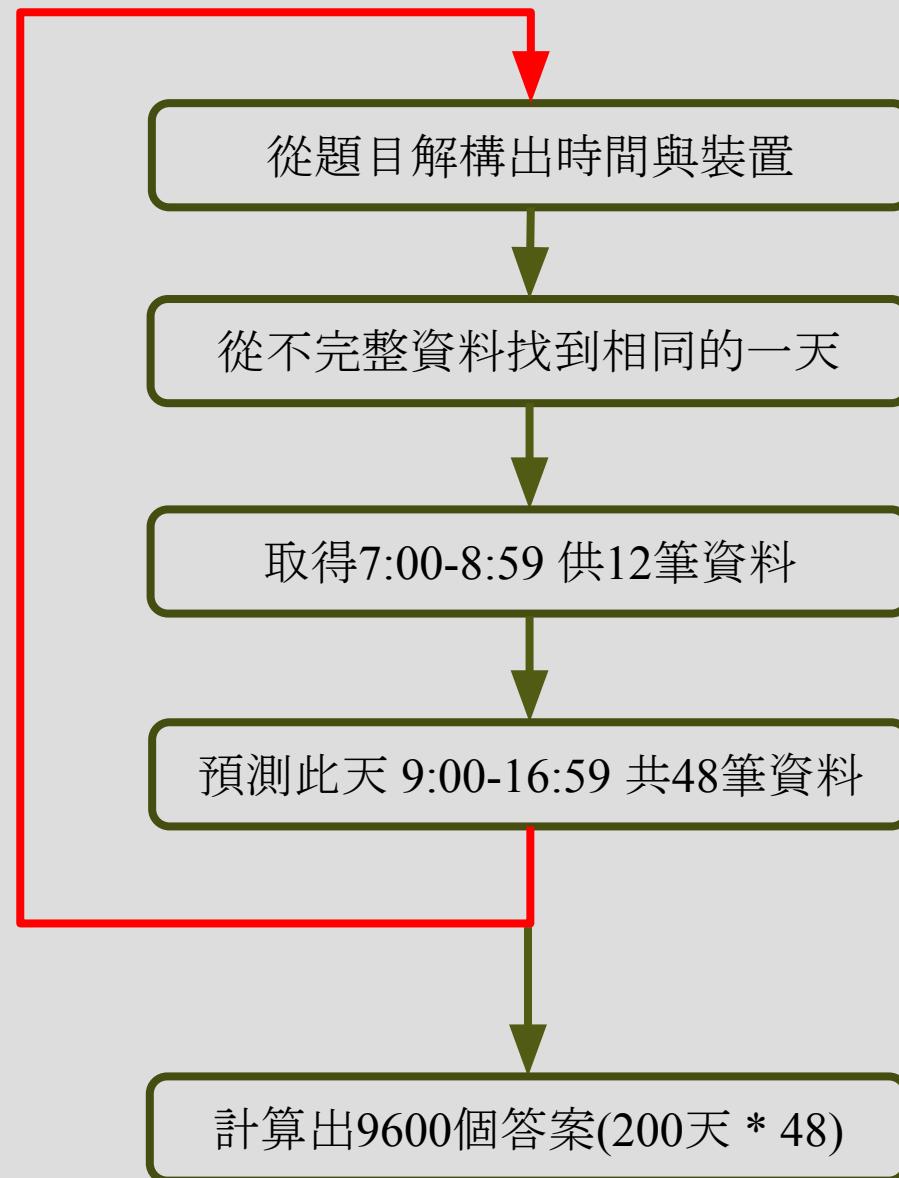
To the right, the content of the selected file "IncompleteAvgDATA\_01.csv" is displayed in a code editor:

```
Serial,WindSpeed(m/s),Pressure(hpa),Temperature(°C),Humidity(%),Sunlight(Lux),Power(mW)
20240101070001,0,1016.73,17.5,80.99,470.83,0.08
20240101071001,0,1016.77,17.57,86.97,659.67,0.14
20240101072001,0.68,1016.92,17.63,86.18,889.67,0.25
20240101073001,0.61,1016.99,17.74,85.41,1263.83,0.45
20240101074001,0,1017.09,17.9,85.41,2136.83,1.13
20240101075001,0,1017.27,18.13,85.19,2598.1.6
20240101080001,0,1017.42,18.29,84.38,2160.17,1.18
20240101081001,0,1017.44,18.3,83.79,2424.67,1.44
20240101082001,0,1017.55,18.28,84.88,3204.5,2.37
20240101083001,1.83,1017.64,18.23,85.92,4376.42,4.18
20240101084001,0.38,1017.75,18.48,85.48,6681.33,9.76
20240101085001,1.43,1017.91,18.77,87.74,6693.83,11.86
20240102070001,0,1014.65,12.99,100,4352.58,8.3
20240102071001,0.07,1014.81,14.44,100,8052.67,35.99
20240102072001,0.39,1014.85,15.55,100,8764,28.9
20240102073001,0.11,1014.92,16.24,100,15173.17,101.45
20240102074001,0.06,1014.98,18.97,100,25676.25,265.02
20240102075001,0.11,1014.91,22.21,90.75,32754.75,386.96
20240102080001,0.29,1014.86,24.48,71.4,39953.71,513.66
20240102081001,0.42,1014.87,25.97,63.99,46366.74,635.5
20240102082001,0.32,1014.81,27.1,59.17,52176.04,767.84
20240102083001,0.72,1014.8,27.78,55.52,57116.18,908.43
20240102084001,0.98,1014.8,28.2,52.96,60843.08,1006.15
20240102085001,0.1,1014.85,29.01,52.95,66731.44,1164.03
20240103070001,0.71,1011.79,15.44,100,1389.42,0.56
20240103071001,1.7,1011.84,15.65,100,2028.33,1.15
```

A red arrow points from the text "平均後的檔案格式" to the first row of the CSV data, highlighting the header. Another red arrow points from the text "資料平均" to the second row of the CSV data.

平均後的檔案格式 □ 代碼14碼,風速,大氣壓力,溫度,濕度,光照度,發電量

# LSTM氣候數據預測-程式介紹



# LSTM氣候數據預測-程式介紹

返回當前工作目錄

載入CSV檔案到程式裡面

```
#設定LSTM往前看的筆數和預測筆數  
LookBackNum = 12 #LSTM往前看的筆數  
ForecastNum = 48 #預測筆數  
  
#載入訓練資料  
DataName = os.getcwd()+'\\ExampleTrainData(AVG)\\AvgDATA_17.csv'  
SourceData = pd.read_csv(DataName, encoding='utf-8')  
  
#選擇要留下來的資料欄位(發電量)  
target = ['Power(mW)']  
AllOutput = SourceData[target].values
```

留下指定的欄位

# LSTM氣候數據預測-程式介紹

Serial	WindSpeed(m/s)	Pressure(hpa)	Temperature(°C)	Humidity(%)	Sunlight(Lux)	Power(mW)
20240101090017	0	1024.42	19.65	3.5758.17	7.23	
20240101091017	0	1024.45	19.01	65.15	5778.75	7.81
20240101092017	0	1024.54	19.15	64.65	7257.17	11.54
20240101093017	0	1024.62	19.3	64.67	7517.17	12.46
20240101094017	0	1024.56	19.45	64.39	7619.08	13.21
20240101095017	0	1024.47	19.65	64.39	11013.58	25.53
20240101100017	0	1024.31	20.26	63.98	15059	47.37
20240101101017	0	1024.25	20.98	62.44	16462.25	58.65
20240101102017	0	1024.19	21.49	61.07	15433.92	54.15
20240101103017	0	1024.13	20.89	60.04	8632.25	16.41
20240101104017	0	1024.16	20.34	61.9	11667.33	30.63
20240101105017	0	1024	20.4	61.13	11000	27.1
20240101110017	0	1023.82	20.47	61.4	11711.83	30.46
20240101110117	0	1023.66	20.61	60.28	11893.67	31.45
202401011102017	0	1023.5	20.55	60.1	9257	19.45
202401011103017	0	1023.25	20.52	61.12	10974.42	26.58
202401011104017	0	1023.08	20.52	61.13	10979.83	27.33
202401011105017	0	1022.82	20.31	62.09	9592.58	20.83
202401011120017	0	1022.62	20.19	63.32	11326.08	29.45
202401011121017	0	1022.47	20.66	63.1	13682.75	41.75
202401011122017	0	1022.24	20.6	61.69	10262.83	23.73
202401011123017	0	1022.19	20.13	62.3	8294.17	15.44
202401011124017	0	1022.02	19.75	62.8	6031.67	8.46
202401011125017	0	1021.88	19.46	63.25	5408	6.87
202401011130017	0	1021.89	19.25	63.38	4619.75	4.9
202401011131017	0	1021.72	19.02	63.83	4267.17	4.06
202401011132017	0	1021.65	18.86	64.22	3849.5	3.36
202401011133017	0	1021.59	18.73	64.68	3835.67	3.35

Interactive-1 AllOutput [DW] X

匯出為檔案 ...

# index	# 0	... 最小值 7.23 最大值 1980.59
0	7.23	
1	7.81	
2	11.54	
3	12.46	
4	13.21	
5	25.53	
6	47.37	
7	58.65	
8	54.15	
9	16.41	
10	30.63	
11	27.1	
12	30.46	
13	31.45	
14	19.45	
15	26.58	
16	27.33	
17	20.83	
18	29.45	
19	41.75	
20	23.73	
21	15.44	

# LSTM氣候數據預測-程式介紹

從 12 開始計數，並且要數整個CSV的長度

資料加入X\_train  
資料加入Y\_train

將python的原生陣列轉換成numpy陣列

```
x_train = []
y_train = []

#設定每i-12筆資料(x_train)就對應到第i筆資料(y_train)
for i in range(LookBackNum,len(AllOutPut)):
    x_train.append(AllOutPut[i-LookBackNum:i, 0])
    y_train.append(AllOutPut[i, 0])

x_train = np.array(x_train)
y_train = np.array(y_train)
```

# LSTM氣候數據預測-程式介紹

LookBackNum = 12

AllOutPut = 14

假設第一次進入迴圈  $\square i = 12$

```
#設定每i-12筆資料(x_train)就對應到第i筆資料(y_train)
for i in range(LookBackNum, len(AllOutPut)):
    x_train.append(AllOutPut[i-LookBackNum:i, 0])
    y_train.append(AllOutPut[i, 0])
```

X\_train 新增一行( AllOutPut [12-12 : 12, 0] )  
行 列

Y\_train 新增一行( AllOutPut [12 , 0] )  
行 列

資料列
0
1
2
3
4
5
6
7
8
9
10
11
12
13

X從這裡開始數 

Y從這裡開始數 

# LSTM氣候數據預測-程式介紹

LookBackNum = 12

AllOutPut = 14

假設第二次進入迴圈  $\square i = 13$

```
#設定每i-12筆資料(x_train)就對應到第i筆資料(y_train)
for i in range(LookBackNum, len(AllOutPut)):
    x_train.append(AllOutPut[i-LookBackNum:i, 0])
    y_train.append(AllOutPut[i, 0])
```

X\_train 新增一行( AllOutPut [13-12 : 13, 0] )  
行 列

Y\_train 新增一行( AllOutPut [13 , 0] )  
行 列

資料列
0
1
2
3
4
5
6
7
8
9
10
11
12
13

X從這裡開始數 

Y從這裡開始數 

# LSTM氣候數據預測-程式介紹

X_trai		Y_trai	
	n		n
0	[ 7.23 7.81 11.54 12.46 13.21 25.53 47.37 58.65 54.15 16.41 30.63 27.1 ]	0	30.46
1	[ 7.81 11.54 12.46 13.21 25.53 47.37 58.65 54.15 16.41 30.63 27.1 30.46]	1	31.45
2	[11.54 12.46 13.21 25.53 47.37 58.65 54.15 16.41 30.63 27.1 30.46 31.45]	2	19.45
3	[12.46 13.21 25.53 47.37 58.65 54.15 16.41 30.63 27.1 30.46 31.45 19.45]	3	26.58
4	[13.21 25.53 47.37 58.65 54.15 16.41 30.63 27.1 30.46 31.45 19.45 26.58]	4	27.33
5	[25.53 47.37 58.65 54.15 16.41 30.63 27.1 30.46 31.45 19.45 26.58 27.33]	5	20.83
6	[47.37 58.65 54.15 16.41 30.63 27.1 30.46 31.45 19.45 26.58 27.33 20.83]	6	29.45
7	[58.65 54.15 16.41 30.63 27.1 30.46 31.45 19.45 26.58 27.33 20.83 29.45]	7	41.75
8	[54.15 16.41 30.63 27.1 30.46 31.45 19.45 26.58 27.33 20.83 29.45 41.75]	8	23.73
9	[16.41 30.63 27.1 30.46 31.45 19.45 26.58 27.33 20.83 29.45 41.75 23.73]	9	15.44
10	[30.63 27.1 30.46 31.45 19.45 26.58 27.33 20.83 29.45 41.75 23.73 15.44]	10	8.46

# LSTM氣候數據預測-程式介紹

```
x_train = np.array(X_train)
y_train = np.array(y_train)

# Reshaping
x_train = np.reshape(x_train,(x_train.shape [0], x_train.shape [1], 1))
```

X\_train.shape[0] □ 行的長度  
X\_train.shape[1] □ 列的長度



LSTM需要的訓練格式為 □ 樣本數(Samples),時間步長(Timesteps),特徵數(Features)

# LSTM氣候數據預測-程式介紹

```
regressor = Sequential ()  
regressor.add(LSTM(units = 128, return_sequences = True, input_shape = (X_train.shape[1], 1)))  
regressor.add(LSTM(units = 64))  
regressor.add(Dropout(0.2))  
  
# output layer  
regressor.add(Dense(units = 1))  
regressor.compile(optimizer = 'adam', loss = 'mean_squared_error')  
  
#開始訓練  
regressor.fit(X_train, y_train, epochs = 500, batch_size = 128)
```

建置 Sequential 模型

第一層 □ LSTM層( 設定單元數量 , 是否輸出整個序列 , 輸入形狀( 時間步長(Timesteps) , 特徵數(Features) ) )

第二層 □ LSTM層( 設定單元數量 )

第三層 □ 正則化層( 設定要丟棄的神經元比率 )

第四層 □ 全連接層( 設定要有幾個輸出 )

編譯模型( 設定優化器 , 設定誤差計算方式 )

擬合模型( 輸入 , 標籤 , 迭代次數 , 批次大小 )

# LSTM氣候數據預測-程式介紹

匯入時間函式  
設定時間格式  
儲存訓練好的模型

```
#保存模型
from datetime import datetime
NowDateTime = datetime.now().strftime("%Y-%m-%dT%H_%M_%SZ")
regressor.save('WheatherLSTM_'+NowDateTime+'.h5')
print('Model Saved')
```

# LSTM氣候數據預測-程式介紹

改成訓練好的模型名稱



```
#載入模型  
regressor = load_model('WheatherLSTM_2024-09-17T20_25_07Z.h5')  
  
#載入測試資料  
DataName = os.getcwd()+r'\ExampleTestData\upload.csv'  
SourceData = pd.read_csv(DataName, encoding='utf-8')  
target = ['序號']  
EXquestion = SourceData[target].values
```

```
inputs = [] #存放參考資料  
PredictOutput = [] #存放預測值  
  
count = 0  
> while(count < len(EXquestion)):
```

題目總共行數



# LSTM氣候數據預測-程式介紹

把題目的裝置代號解析出來

從透過裝置代號找到  
不完整一天的平均資料

從資料中找到相同的一天  
並提取7:00~8:59的發電瓦數  
總共12筆資料

```
while(count < len(EXquestion)):
    print('count : ',count)
    LocationCode = int(EXquestion[count])
    strLocationCode = str(LocationCode)[-2:]
    if LocationCode < 10 :
        strLocationCode = '0'+strLocationCode

DataName = os.getcwd()+'\\ExampleTrainData\IncompleteAVG\IncompleteAvgDATA_'+ strLocationCode +'.csv'
SourceData = pd.read_csv(DataName, encoding='utf-8')
ReferTitle = SourceData[['Serial']].values
ReferData = SourceData[['Power(mW)']].values

inputs = []#重置存放參考資料

#找到相同的一天，把12個資料都加進inputs
for DaysCount in range(len(ReferTitle)):
    if(str(int(ReferTitle[DaysCount]))[:8] == str(int(EXquestion[count]))[:8]):
        inputs = np.append(inputs, ReferData[DaysCount])
```

# LSTM氣候數據預測-程式介紹

要做48次

```
#用迴圈不斷使新的預測值塞入參考資料，並預測下一筆資料  
for i in range(ForecastNum) :
```

```
#print(i)
```

```
#將新的預測值加入參考資料(用自己的預測值往前看)
```

```
if i > 0 :
```

```
    inputs = np.append(inputs, PredictOutput[i-1])
```

```
#切出新的參考資料12筆(往前看12筆)
```

```
X_test = []
```

```
X_test.append(inputs[0+i:LookBackNum+i])
```

```
#Reshaping
```

```
NewTest = np.array(X_test)
```

```
NewTest = np.reshape(NewTest, (NewTest.shape[0], NewTest.shape[1], 1))
```

```
predicted = regressor.predict(NewTest)
```

```
PredictOutput.append(round(predicted[0,0], 2))
```

把X\_test變成3維的形狀

把資料丟進模型進行數值預測  
並以四捨五入的方式儲存預測值

# LSTM氣候數據預測-程式介紹

要做48次

```
#用迴圈不斷使新的預測值塞入參考資料，並預測下一筆資料  
for i in range(ForecastNum) :
```

當每預測完一次  
就把預測值加回inputs

```
#將新的預測值加入參考資料(用自己的預測值往前看)  
if i > 0 :  
    inputs = np.append(inputs, PredictOutput[i-1])
```

設 i = 1  
inputs[1:13]

設 i = 2  
inputs[2:14]

```
#切出新的參考資料12筆(往前看12筆)  
x_test = []  
x_test.append(inputs[0+i:LookBackNum+i])
```

```
#Reshaping  
NewTest = np.array(x_test)  
NewTest = np.reshape(NewTest, (NewTest.shape[0], NewTest.shape[1], 1))  
predicted = regressor.predict(NewTest)  
PredictOutput.append(round(predicted[0,0], 2))
```

```
#每次預測都要預測48個，因此加48個會切到下一天  
#0~47,48~95,96~143...  
count += 48
```

# LSTM氣候數據預測-程式介紹

```
#用迴圈不斷使新的預測值塞入參考資料，並預測下一筆資料
for i in range(ForecastNum) :

    #print(i)

    #將新的預測值加入參考資料(用自己的預測值往前看)
    if i > 0 :
        inputs = np.append(inputs, PredictOutput[i-1])

    #切出新的參考資料12筆(往前看12筆)
    X_test = []
    X_test.append(inputs[0+i:LookBackNum+i])

    #Reshaping
    NewTest = np.array(X_test)
    NewTest = np.reshape(NewTest, (NewTest.shape[0], NewTest.shape[1], 1))
    predicted = regressor.predict(NewTest)
    PredictOutput.append(round(predicted[0,0], 2))

#每次預測都要預測48個，因此加48個會切到下一天
#0~47,48~95,96~143...
count += 48
```

因為每次都預測48筆  
所以結束後conunt + 48  
直至超出題目範圍

# LSTM氣候數據預測-程式介紹

將預測值匯出成csv檔

```
# 寫預測結果寫成新的CSV檔案  
# 將陣列轉換為 DataFrame  
df = pd.DataFrame(PredictOutput, columns=['答案'])  
  
# 將 DataFrame 寫入 CSV 檔案  
df.to_csv('output.csv', index=False)  
print('Output CSV File Saved')
```

	data
1	答案
2	394.47
3	393.35
4	490.59
5	761.18
6	813.07
7	770.11
8	582.05
9	660.12
10	265.05
11	129.94
12	78.14
13	55.65
14	51.94
15	48.14
16	41.1
17	35.25
18	28.48
19	25.87
20	22.06
21	14.55
22	14.42
23	14.79
24	16.55
25	14.59

# LSTM & 迴歸分析氣候數據預測-程式介紹

- 搭建神經網路的結構
- 搭建全連接層
- 搭建LSTM層
- 搭建正則化層
- 加载已經訓練好的模型
  
- 加载Scikit-learn的回歸模型
- 進行正規化
- 儲存與載入模型
  
- 進行科學運算
- 進行數據處理
- 管理文件和目錄

執行儲存格 | 執行更低版本 | 偵錯儲存格

#%%

```
from tensorflow.keras.models import Sequential  
from tensorflow.keras.layers import Dense  
from tensorflow.keras.layers import LSTM  
from tensorflow.keras.layers import Dropout  
from tensorflow.keras.models import load_model
```

```
from sklearn.linear_model import LinearRegression  
from sklearn.preprocessing import MinMaxScaler  
import joblib
```

```
import numpy as np  
import pandas as pd  
import os
```

Python 版本 : 3.9  
tensorflow\_gpu 版本 : 2.6.0

# LSTM & 迴歸分析氣候數據預測-程式介紹

將 Python 代碼劃分為可執行的「代碼區塊」



執行儲存格 | 執行更低版本 | 偵錯儲存格

#%%

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import LSTM
from tensorflow.keras.layers import Dropout
from tensorflow.keras.models import load_model

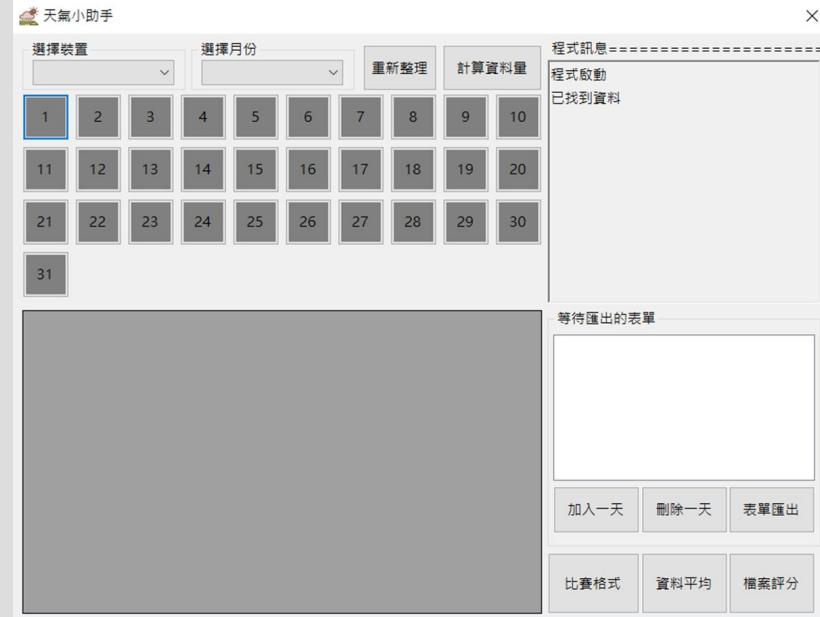
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import MinMaxScaler
import joblib

import numpy as np
import pandas as pd
import os
```

Python 版本 : 3.9

tensorflow\_gpu 版本 : 2.6.0

# LSTM & 迴歸分析氣候數據預測-資料介紹



範例程式用測試資料

範例程式用 平均後的訓練資料

- ExampleTestData
- ExampleTrainData(AVG)
- ExampleTrainDataIncompleteAVG
- Regression+LSTM\_20240917.py

# LSTM & 迴歸分析氣候數據預測-資料介紹

範例程式用測試資料



ExampleTestData > upload.csv > data	
	序號,答案
1	20240102090001,1277.55
2	20240102091001,1369.8
3	20240102092001,1462.67
4	20240102093001,1540.4
5	20240102094001,1594.3
6	20240102095001,1631.14
7	20240102100001,1659.6
8	20240102101001,1671.31
9	20240102102001,1674.38
10	20240102103001,1669.7
11	20240102104001,1689.69
12	20240102105001,1708.54
13	20240102110001,1710.23
14	20240102111001,1734.19
15	20240102112001,1727.19
16	20240102113001,1760.06
17	20240102114001,1766.71
18	20240102115001,1728.95
19	20240102120001,1705.11
20	20240102121001,1703.99
21	20240102122001,1733.1
22	20240102123001,1664.08
23	20240102124001,1791.86
24	20240102125001,1791.78
25	20240102130001,1329.92
26	20240102131001,417.97
27	

代碼14碼 = 西元年(4碼)+月(2碼)+日(2碼)+時(2碼)+分(2碼)+裝置代碼(2碼)

發電量(四捨五入)

# LSTM & 迴歸分析氣候數據預測-資料介紹

範例程式用訓練資料  
(9:00 - 當天結束收集為止)

2024-01-01 09:50

2024-01-01 09:59

資料平均

The screenshot shows a file explorer interface. On the left, under 'ExampleTrainData(AVG)', there is a folder named 'AvgDATA' containing 17 CSV files labeled 'AvgDATA\_01.csv' through 'AvgDATA\_17.csv'. Below these is a 'Title.txt' file. On the right, a preview of 'AvgDATA\_01.csv' is displayed, showing columns for Serial, WindSpeed(m/s), Pressure(hpa), Temperature(°C), Humidity(%), Sunlight(Lux), and Power(mW). The data rows show various values for these parameters over time.

Serial	WindSpeed(m/s)	Pressure(hpa)	Temperature(°C)	Humidity(%)	Sunlight(Lux)	Power(mW)
20240101090001	1.92	1017.91	18.33	92.29	8395.25	17.37
20240101091001	0.09	1017.96	18.02	97.37	7046.5	12.61
20240101092001	0.02	1018.16	17.73	99.93	5919.58	8.81
20240101093001	0.02	1018.2	17.9	100	8838.92	17.56
20240101094001	0.04	1018.13	18.08	100	5774.67	7.51
20240101095001	0.02	1018.03	17.83	100	5437.75	7.03
20240101100001	0	1018.04	17.75	100	9312.75	19.71
20240101101001	0.04	1017.87	17.99	100	9500.67	20.35
20240101102001	0.02	1017.82	18.13	100	9711.58	20.43
20240101103001	0.01	1017.82	18.2	100	7952.5	14.01
20240101104001	0.58	1017.71	18.11	95.79	6422.5	9.33
20240101105001	0.03	1017.55	18.21	93.92	7941.42	13.53
20240101110001	0.2	1017.46	18.64	92.71	7700.92	13.2
20240101111001	0.35	1017.28	18.78	87.1	6558.92	9.47
20240101112001	0.25	1017.09	18.8	86.15	6868.25	10.26
20240101113001	0.17	1016.87	18.94	84.1	8485.08	15.88
20240101114001	0.06	1016.56	19.38	84.1	10389.83	22.82
20240101115001	0.15	1016.36	19.77	80.9	9784.75	20.51
20240101120001	0.43	1016.17	19.87	80.09	11379	27.05
20240101121001	0.97	1015.97	19.96	77.42	9266.17	18.22
20240101122001	0.4	1015.82	19.81	78	9845.92	20.46
20240101123001	1.27	1015.67	20.04	76.07	16415.58	55.61
20240101124001	0.75	1015.51	20.35	74.86	13374.75	37.65
20240101125001	1.37	1015.42	20.29	74.1	10362.5	23.23

平均後的檔案格式 □ 代碼14碼,風速,大氣壓力,溫度,濕度,光照度,發電量

# LSTM & 迴歸分析氣候數據預測-資料介紹

範例程式用訓練資料  
(7:00 – 8:59)

2024-01-01 07:50

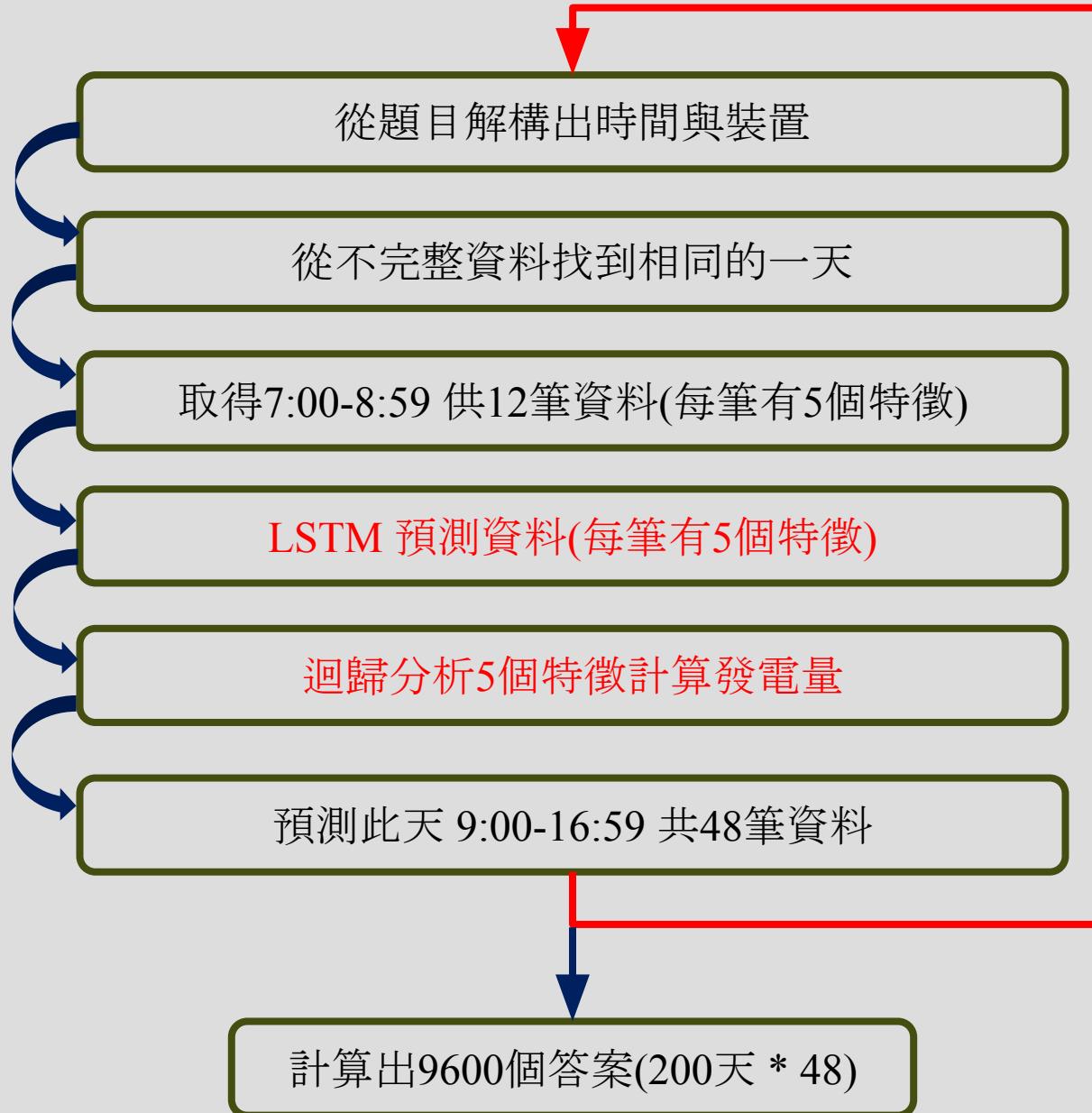
2024-01-01 07:59

資料平均

Serial	WindSpeed(m/s)	Pressure(hpa)	Temperature(°C)	Humidity(%)	Sunlight(Lux)	Power(mW)
20240101070001	0	1016.75	17.5	80.99	470.05	0.00
20240101071001	0	1016.77	17.57	86.97	659.67	0.14
20240101072001	0.68	1016.92	17.63	86.18	889.67	0.25
20240101073001	0.61	1016.99	17.74	85.41	1263.83	0.45
20240101074001	0	1017.09	17.9	85.41	2136.83	1.13
20240101075001	0	1017.27	18.13	85.19	2598.16	
20240101080001	0	1017.42	18.29	84.38	2160.17	1.18
20240101081001	0	1017.44	18.3	83.79	2424.67	1.44
20240101082001	0	1017.55	18.28	84.88	3204.5	2.37
20240101083001	1.83	1017.64	18.23	85.92	4376.42	4.18
20240101084001	0.38	1017.75	18.48	85.48	6681.33	9.76
20240101085001	1.43	1017.91	18.77	87.74	6693.83	11.86
20240102070001	0	1014.65	12.99	100	4352.58	8.3
20240102071001	0.07	1014.81	14.44	100	8052.67	35.99
20240102072001	0.39	1014.85	15.55	100	8764	28.9
20240102073001	0.11	1014.92	16.24	100	15173.17	101.45
20240102074001	0.06	1014.98	18.97	100	25676.25	265.02
20240102075001	0.11	1014.91	22.21	90.75	32754.75	386.96
20240102080001	0.29	1014.86	24.48	71.4	39953.71	513.66
20240102081001	0.42	1014.87	25.97	63.99	46366.74	635.5
20240102082001	0.32	1014.81	27.1	59.17	52176.04	767.84
20240102083001	0.72	1014.8	27.78	55.52	57116.18	908.43
20240102084001	0.98	1014.8	28.2	52.96	60843.08	1006.15
20240102085001	0.1	1014.85	29.01	52.95	66731.44	1164.03
20240103070001	0.71	1011.79	15.44	100	1389.42	0.56

平均後的檔案格式 □ 代碼14碼,風速,大氣壓力,溫度,濕度,光照度,發電量

# LSTM & 迴歸分析氣候數據預測-程式介紹



# LSTM & 迴歸分析氣候數據預測-程式介紹

返回當前工作目錄  
載入CSV檔案到程式裡面

```
#設定LSTM往前看的筆數和預測筆數
LookBackNum = 12 #LSTM往前看的筆數
ForecastNum = 48 #預測筆數

#載入訓練資料
DataName = os.getcwd()+'\ExampleTrainData(AVG)\AvgDATA_17.csv'
SourceData = pd.read_csv(DataName, encoding='utf-8')

#迴歸分析 選擇要留下來的資料欄位
#(風速,大氣壓力,溫度,濕度,光照射)
#(發電量)
Regression_X_train = SourceData[['WindSpeed(m/s)', 'Pressure(hpa)', 'Temperature(°C)', 'Humidity(%)', 'Sunlight(Lux)']].values
Regression_y_train = SourceData[['Power(mW)']].values

#LSTM 選擇要留下來的資料欄位
#(風速,大氣壓力,溫度,濕度,光照射)
AllOutput = SourceData[['WindSpeed(m/s)', 'Pressure(hpa)', 'Temperature(°C)', 'Humidity(%)', 'Sunlight(Lux)']].values
```

留下指定的欄位

# LSTM & 迴歸分析氣候數據預測-程式介紹

Serial	windSpeed(m/s)	Pressure(hpa)	Temperature(°C)	Humidity(%)	Sunlight(Lux)	Power(mW)
20240101090017	0,1024.42,19,65.3,5758.17					
20240101091017	0,1024.45,19.01,65.15,5778.75,7.81					
20240101092017	0,1024.54,19.15,64.65,7257.17,11.54					
20240101093017	0,1024.62,19.3,64.67,7517.17,12.46					
20240101094017	0,1024.56,19.45,64.39,7619.08,13.21					
20240101095017	0,1024.47,19.65,64.39,11013.58,25.53					
20240101100017	0,1024.31,20.26,63.98,15059,47.37					
20240101101017	0,1024.25,20.98,62.44,16462.25,58.65					
20240101102017	0,1024.19,21.49,61.07,15433.92,54.15					
20240101103017	0,1024.13,20.89,60.04,8632.25,16.41					
20240101104017	0,1024.16,20.34,61.9,11667.33,30.63					
20240101105017	0,1024,20.4,61.13,11000,27.1					
20240101110017	0,1023.82,20.47,61.4,11711.83,30.46					
20240101111017	0,1023.66,20.61,60.28,11893.67,31.45					
20240101112017	0,1023.5,20.55,60.1,9257,19.45					
20240101113017	0,1023.25,20.52,61.12,10974.42,26.58					



# LSTM & 迴歸分析氣候數據預測-程式介紹

將陣列內的特徵都壓縮到0與1之間  
並暫存成一個正規化模型

把檔案丟進模型內進行正規化  
並生成一個新的陣列

#正規化

```
LSTM_MinMaxModel = MinMaxScaler().fit(AllOutput)
AllOutput_MinMax = LSTM_MinMaxModel.transform(AllOutput)
```

正規化前

0	0.0	1024.42	19.0	65.3	5758.17
1	0.0	1024.45	19.01	65.15	5778.75
2	0.0	1024.54	19.15	64.65	7257.17
3	0.0	1024.62	19.3	64.67	7517.17
4	0.0	1024.56	19.45	64.39	7619.08
5	0.0	1024.47	19.65	64.39	11013.58

正規化後

0	0.0	0.7958387516254888	0.13421418636995824	0.5254423261555341	0.047172113934023666
1	0.0	0.7968140442132636	0.13444598980064904	0.5233849951995613	0.047347196011191306
2	0.0	0.7997399219765953	0.13769123783031983	0.5165272253463173	0.05992469088612715
3	0.0	0.8023407022106568	0.14116828929068148	0.5168015361404472	0.06213661217201567
4	0.0	0.8003901170351071	0.14464534075104307	0.5129611850226306	0.06300360024218837
5	0.0	0.7974642392717826	0.14928140936485856	0.5129611850226306	0.09188193410737534

# LSTM & 迴歸分析氣候數據預測-程式介紹

從 12 開始計數，並且要數整個CSV的長度

資料加入X\_train  
資料加入Y\_train

將python的原生陣列轉換成numpy陣列

```
x_train = []
y_train = []

#設定每i-12筆資料(X_train)就對應到第i筆資料(y_train)
for i in range(LookBackNum,len(AllOutput_MinMax)):
    x_train.append(AllOutput_MinMax[i-LookBackNum:i, :])
    y_train.append(AllOutput_MinMax[i, :])

x_train = np.array(x_train)
y_train = np.array(y_train)
```

# LSTM & 迴歸分析氣候數據預測-程式介紹

LookBackNum = 12

AllOutPut = 14

假設第一次進入迴圈  $\square i = 12$

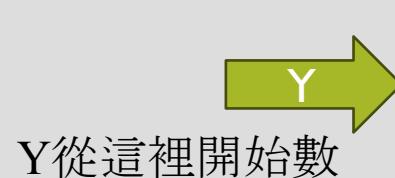
```
#設定每i-12筆資料(x_train)就對應到第i筆資料(y_train)
for i in range(LookBackNum, len(AllOutPut_MinMax)):
    x_train.append(AllOutPut_MinMax[i-LookBackNum:i, :])
    y_train.append(AllOutPut_MinMax[i, :])
```

X\_train 新增一行( AllOutPut [12-12 : 12, :] )  
行 列

Y\_train 新增一行( AllOutPut [12, :] )  
行 列



X從這裡開始數



列1	列2	列3	列4	列5
0	0	0	0	0
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5
6	6	6	6	6
7	7	7	7	7
8	8	8	8	8
9	9	9	9	9
10	10	10	10	10
11	11	11	11	11
12	12	12	12	12
13	13	13	13	13

# LSTM & 迴歸分析氣候數據預測-程式介紹

LookBackNum = 12

AllOutPut = 14

假設第二次進入迴圈  $\square i = 13$

```
#設定每i-12筆資料(x_train)就對應到第i筆資料(y_train)
for i in range(LookBackNum, len(AllOutPut_MinMax)):
    x_train.append(AllOutPut_MinMax[i-LookBackNum:i, :])
    y_train.append(AllOutPut_MinMax[i, :])
```

X\_train 新增一行( AllOutPut [13-12 : 13, :] )  
行 列

Y\_train 新增一行( AllOutPut [13, :] )  
行 列



X從這裡開始數



Y從這裡開始數

列1	列2	列3	列4	列5
0	0	0	0	0
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5
6	6	6	6	6
7	7	7	7	7
8	8	8	8	8
9	9	9	9	9
10	10	10	10	10
11	11	11	11	11
12	12	12	12	12
13	13	13	13	13

# LSTM & 迴歸分析氣候數據預測-程式介紹

```
X_train = np.array(X_train)
y_train = np.array(y_train)

# Reshaping
#(samples 是訓練樣本數量,timesteps 是每個樣本的時間步長,features 是每個時間步的特徵數量)
X_train = np.reshape(X_train,(X_train.shape [0], X_train.shape [1], 5))
```

X\_train.shape[0] □ 行的長度  
X\_train.shape[1] □ 列的長度



LSTM需要的訓練格式為 □ 樣本數(Samples),時間步長(Timesteps),特徵數(Features)

# LSTM & 迴歸分析氣候數據預測-程式介紹

```
regressor = Sequential () ——————→ 建置 Sequential 模型  
regressor.add(LSTM(units = 128, return_sequences = True, input_shape = (x_train.shape[1], 5)))  
regressor.add(LSTM(units = 64))  
regressor.add(Dropout(0.2))  
  
# output layer  
regressor.add(Dense(units = 5))  
regressor.compile(optimizer = 'adam', loss = 'mean_squared_error')  
  
#開始訓練  
regressor.fit(x_train, y_train, epochs = 500, batch_size = 128)
```

第一層□ LSTM層( 設定單元數量 , 是否輸出整個序列 , 輸入形狀( 時間步長(Timesteps) , 特徵數(Features) ) )

第二層□ LSTM層( 設定單元數量 )

第三層□正則化層( 設定要丟棄的神經元比率 )

第四層□全連接層( 設定要有幾個輸出 )

編譯模型( 設定優化器 , 設定誤差計算方式 )

擬合模型( 輸入 , 標籤 , 迭代次數 , 批次大小 )

# LSTM & 迴歸分析氣候數據預測-程式介紹

匯入時間函式  
設定時間格式  
儲存訓練好的模型

```
#保存模型
from datetime import datetime
NowDateTime = datetime.now().strftime("%Y-%m-%dT%H_%M_%SZ")
regressor.save('WheatherLSTM_'+NowDateTime+'.h5')
print('Model Saved')
```

# LSTM & 迴歸分析氣候數據預測-程式介紹

搭建一個回歸模型  
訓練回歸模型

儲存訓練好的模型

```
#開始迴歸分析(對發電量做迴歸)
RegressionModel = LinearRegression()
RegressionModel.fit(LSTM_MinMaxModel.transform(Regression_X_train), Regression_y_train)

#儲存回歸模型
from datetime import datetime
NowDateTime = datetime.now().strftime("%Y-%m-%dT%H_%M_%SZ")
joblib.dump(RegressionModel, 'wheatherRegression_'+NowDateTime)
```

特徵正規化

標籤

# LSTM & 迴歸分析氣候數據預測-程式介紹

$$Y = \alpha_1 X_1 + \alpha_2 X_2 + \alpha_3 X_3 + \alpha_4 X_4 + \alpha_5 X_5 + \beta$$

↑      ↑      ↑      ↑      ↑  
風速    大氣壓力    溫度    濕度    光照度  
↓      ↓      ↓      ↓      ↓  
係數1   係數2   係數3   係數4   係數5   截距

```
#取得截距
print('截距: ', RegressionModel.intercept_)

#取得係數
print('係數 : ', RegressionModel.coef_)

#取得R平方
print('R squared: ', RegressionModel.score(LSTM_MinMaxModel.transform(Regression_X_train), Regression_y_train))
```

$R^2 = 1 \rightarrow$  用特徵可準確的預測出標籤  
 $R^2 = 0 \rightarrow$  用特徵無法的預測出標籤

截距: [-221.39953665]  
係數 : [[ 0.                135.11381615  117.68809059  23.89564382  1761.56873922]]  
R squared: 0.899519098844525

# LSTM & 迴歸分析氣候數據預測-程式介紹

改成訓練好的模型名稱



```
#載入模型  
regressor = load_model('WheatherLSTM_2024-09-17T23_50_15Z.h5')  
Regression = joblib.load('WheatherRegression_2024-09-17T23_51_56Z')  
  
#載入測試資料  
DataName = os.getcwd()+r'\ExampleTestData\upload.csv'  
SourceData = pd.read_csv(DataName, encoding='utf-8')  
target = ['序號']  
EXquestion = SourceData[target].values
```

```
inputs = [] #存放參考資料  
PredictOutput = [] #存放預測值(天氣參數)  
PredictPower = [] #存放預測值(發電量)  
  
count = 0  
while(count < len(EXquestion)): ...
```



題目總共行數

# LSTM & 迴歸分析氣候數據預測-程式介紹

把題目的裝置代號解析出來

從透過裝置代號找到  
不完整一天的平均資料

從資料中找到相同的一天  
並提取7:00~8:59的5個特徵  
總共12筆資料

```
while(count < len(EXquestion)):
    print('count : ',count)
    LocationCode = int(EXquestion[count])
    strLocationCode = str(LocationCode)[-2:]
    if LocationCode < 10 :
        strLocationCode = '0'+LocationCode

    DataName = os.getcwd()+'\\ExampleTrainData\IncompleteAVG\IncompleteAvgDATA_'+ strLocationCode +'.csv'
    SourceData = pd.read_csv(DataName, encoding='utf-8')
    ReferTitle = SourceData[['Serial']].values
    ReferData = SourceData[['WindSpeed(m/s)', 'Pressure(hpa)', 'Temperature(°C)', 'Humidity(%)', 'Sunlight(Lux)']].values

    inputs = []#重置存放參考資料

    #找到相同的一天，把12個資料都加進inputs
    for DaysCount in range(len(ReferTitle)):
        if(str(int(ReferTitle[DaysCount]))[:8] == str(int(EXquestion[count]))[:8]):
            TempData = ReferData[DaysCount].reshape(1,-1)
            TempData = LSTM_MinMaxModel.transform(TempData)
            inputs.append(TempData)
```

# LSTM & 迴歸分析氣候數據預測-程式介紹

```
#用迴圈不斷使新的預測值塞入參考資料，並預測下一筆資料
for i in range(ForecastNum) :
    #print(i)

    #將新的預測值加入參考資料(用自己的預測值往前看)
    if i > 0 :
        inputs.append(PredictOutput[i-1].reshape(1,5))

    #切出新的參考資料12筆(往前看12筆)
    X_test = []
    X_test.append(inputs[0+i:LookBackNum+i])

#Reshaping
NewTest = np.array(X_test)
NewTest = np.reshape(NewTest, (NewTest.shape[0], NewTest.shape[1], 5))

predicted = regressor.predict(NewTest)
PredictOutput.append(predicted)
PredictPower.append(np.round(Regression.predict(predicted),2).flatten())

#每次預測都要預測48個，因此加48個會切到下一天
#0~47, 48~95, 96~143...
count += 48
```

要做48次 ←

把12行的資料加進X\_test ←

把X\_test變成3維的形狀 ←

把資料丟進LSTM模型進行特徵預測  
並加入陣列裡面 ←

把預測的特徵丟進回歸模型進行發電量預測，並以四捨五入的方式儲存預測值 ←

# LSTM & 迴歸分析氣候數據預測-程式介紹

```
#用迴圈不斷使新的預測值塞入參考資料，並預測下一筆資料
for i in range(ForecastNum) :
    #print(i)

    #將新的預測值加入參考資料(用自己的預測值往前看)
    if i > 0 :
        inputs.append(PredictOutput[i-1].reshape(1,5))

    #切出新的參考資料12筆(往前看12筆)
    x_test = []
    x_test.append(inputs[0+i:LookBackNum+i])

#Reshaping
NewTest = np.array(x_test)
NewTest = np.reshape(NewTest, (NewTest.shape[0], NewTest.shape[1], 5))

predicted = regressor.predict(NewTest)
PredictOutput.append(predicted)
PredictPower.append(np.round(Regression.predict(predicted),2).flatten())

#每次預測都要預測48個，因此加48個會切到下一天
#0~47,48~95,96~143...
count += 48
```

要做48次 ←

當每預測完一次  
就把預測值加回inputs ←

設  $i = 1$   
 $\text{inputs}[1:13]$

設  $i = 2$   
 $\text{inputs}[2:14]$

# LSTM & 迴歸分析氣候數據預測-程式介紹

因為每次都預測48筆  
所以結束後count + 48  
直至超出題目範圍

```
#用迴圈不斷使新的預測值塞入參考資料，並預測下一筆資料
for i in range(ForecastNum) :

    #print(i)

    #將新的預測值加入參考資料(用自己的預測值往前看)
    if i > 0 :
        inputs.append(PredictOutput[i-1].reshape(1,5))

    #切出新的參考資料12筆(往前看12筆)
    x_test = []
    x_test.append(inputs[0+i:LookBackNum+i])

    #Reshaping
    NewTest = np.array(x_test)
    NewTest = np.reshape(NewTest, (NewTest.shape[0], NewTest.shape[1], 5))

    predicted = regressor.predict(NewTest)
    PredictOutput.append(predicted)
    PredictPower.append(np.round(Regression.predict(predicted),2).flatten())

#每次預測都要預測48個，因此加48個會切到下一天
#0~47,48~95,96~143...
count += 48
```

# LSTM & 迴歸分析氣候數據預測-程式介紹

將預測值匯出成csv檔

```
# 寫預測結果寫成新的CSV檔案  
# 將陣列轉換為 DataFrame  
df = pd.DataFrame(PredictPower, columns=['答案'])  
  
# 將 DataFrame 寫入 CSV 檔案  
df.to_csv('output.csv', index=False)  
print('Output CSV File Saved')
```

	output.csv > data
1	答案
2	768.36
3	662.75
4	584.73
5	520.45
6	462.02
7	407.21
8	356.23
9	310.26
10	270.0
11	235.3
12	204.95
13	178.22
14	154.81
15	133.9
16	115.82
17	100.72
18	88.71
19	80.17
20	75.24
21	74.06
22	76.51
23	82.54
24	91.81
25	103.57
26	117.38
27	132.31

# AICUP 報名流程教學

Start

步驟一：  
進入報名頁面一  
<https://go.aicup.tw/>

步驟二：  
點選登入 Log In

步驟三：  
選擇 使用google 登入



AI CUP 報名系統 賽事報名 Register for the Competition 賽事報告檢核系統(demo)

AI CUP Registration System 賽事報名 Register for the Competition 賽事報告檢核系統(demo)

請先登入 Log In

步驟二

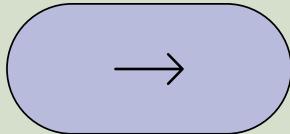
AI CUP 2024 教育部大專校院人工智慧競賽

G 使用 Google 登入

登入 (Login)

步驟三

步驟四：  
點選報名競賽



## 步驟五： 填寫個人資料

僅  
學校名稱(英文)、  
科系名稱(英文)  
非必填  
其餘皆為必填

填寫完畢按  
儲存個人資料



個人資料  
Profile



國立東華大學 資訊工程系  
NDHU CSIE  
學士 Bachelor

儲存個人資料 Save  
登出 Log out

提醒您：  
您所填寫之資料，將作為未來聯絡、製作獎狀等之資料依據，請謹慎填寫。如因填寫錯誤造成權益受損，主辦單位概不負責。

Please note that the information you provide will be used as a basis for future communication and for producing awards. Therefore, please be cautious and make sure to fill in the information accurately. The organizers will not be held responsible for any damages or losses incurred as a result of incorrect information provided by the applicant.

中文姓名  
Chinese Name

xxx

英文姓名  
English Name

xxx

註冊用 Email  
Register Email

[REDACTED]

聯絡用 Email  
Contact Email

[REDACTED]

電話  
Phone Number

038900000

戶籍地址  
Address

花蓮縣壽豐鄉大學路二段一號

身分  
Are you a Student?

我是在學學生 Yes    我是社會人士 No

學校名稱 ( 中文 )  
School Name  
(Chinese)

國立東華大學

學校名稱 ( 英文 )  
School Name  
(English)

NDHU

科系名稱 ( 中文 )  
Department  
(Chinese)

資訊工程系

科系名稱 ( 英文 )  
Department  
(English)

CSIE

系級程度  
Degree

學士 Bachelor

步驟六：  
再次點選報名競賽

步驟七：  
選擇  
根據區域微氣候資料預測發  
電量競賽

步驟八：  
選擇  
立即組隊 Form a team!



AI CUP 報名系統 Register for the Competition

aiabab5051@gmail.com

步驟六

AI CUP Registration System  
Welcome, xxx

目前進行的比賽 Ongoing competitions

提醒您

1. 請在報名競賽前，確認您在本系統所填寫的個資是否正確，以免影響您的權益。如果過去曾參加過AI CUP，請確認資料是否需要更新（例如：前次參賽為在學生，本次參賽已為社會人士則請務必更新您的資料）。

2. 競賽需確實依規定上傳答案，如果獲獎，需繳交報告與程式碼，方具領獎資格，系統排名與賽後報告內容、程式碼同為評審核定最終名次之重要依據，為維持競賽公平，如有謬交、撰寫 / 提交疏漏，不予補交。

Notice

1. Please check whether the personal information you have filled out in this system is correct before registering for the competition, to avoid affecting your rights and interests. If you have participated in AI CUP in the past, please confirm whether your data needs to be updated (for example: if you were a student in the previous competition and now you are a working professional, please be sure to update your information).

2. It is necessary to upload the answers in accordance with the regulations for the competition. If you win, you will need to submit a report and the code to be eligible for the prize. The leaderboard ranking, post-competition report content, and code are important criteria for the judges to determine the final ranking. To maintain a fair competition, late submissions or writing/submission omissions will not be accepted.

AI CUP 2024秋季賽 - 根據區域微氣候資料預測發電量競賽 [ TBrain ]

報名期間 : 2024年9月2日 ~ 2024年11月20日

步驟七

AI CUP 2024秋季賽 - 根據區域微氣候資料預測發電量競賽 [ TBrain ]

報名期間 : 2024年9月2日 23:59 ~ 2024年11月20日 23:59

競賽平台 - T-Brain

步驟八

此為AICUP與TBrain合作之競賽，在AICUP報名系統報名後請去TBrain完成比賽。最終報名截止時間為2024/11/20 23:59。比賽辦法網址：  
<https://tbrain.trendmicro.com.tw/Competitions/Details/36> 註: 此比賽隊伍人數為1~4人(包含隊長)，若超過人數則無法與TBrain連動，無法與TBrain連動的參賽者將視為報名失敗，若報名截止後隊伍超過報名人數上限，則無法做更動，報名失敗者則無參賽資格。

立即組隊 Form a team!

## 步驟九： 填寫表格

指導教授及課程代碼請視自身  
情況填寫。  
基於學校課程要求而參與比賽  
的參賽者，請於報名系統填寫課  
程代碼、並依循該課程之組隊要  
求，以利課堂評分。



AI CUP 2024秋季賽 - 根據區域微氣候資料預測發電量競賽 [ TBrain ]

戰隊編成  
Team Formation

每隊伍至多可填寫兩位指導教授/業師，如未提供，如得獎則視同不領取指導獎狀。  
Each team can provide up to two advisors/instructors. If not provided, the team will be considered ineligible for the advisory award even if they win.

指導教授 1 姓名

指導教授 1 學校

指導教授 1 科系

指導教授 1 Email

指導教授 2 姓名

指導教授 2 學校

指導教授 2 科系

指導教授 2 Email

課程代碼

如何得知本次競賽

(續前頁)

填寫完畢後勾選

我同意參賽者使用條款

並點選 立即組隊

(若需組隊，僅隊長需點擊)

即可完成第一步報名  
請前往報名頁面二

<https://tbrain.trendmicro.com.tw/Competitions/CompetitionList>

點選 Sign In



若是參與和 T-Brain 合作之競賽，報名隊伍每位隊員皆須於『T-Brain AI 實戰吧』以 Google 或 Facebook 帳號註冊會員，來完成整個報名流程 註：在 T-Brain 註冊之 Google 或 Facebook 帳號須與 AI CUP 報名系統一致

※ 報名組隊後「無法合併/拆分/刪除隊伍」，因此請參賽者確認自身隊伍的隊長/隊員分工再報名。

※ 點選「立即組隊」並成功創建隊伍後，該參賽者帳號即自動成為「隊長」。如您為隊員，請由隊長創建隊伍以後，請隊長提供您「戰隊資訊」頁面網址，再行加入。隊伍一旦創建即無法合併/拆分/刪除，且隊長「無法離開」自己創建的隊伍，也「無法再加入其他隊伍」成為隊員。

※ 詳細報名操作請參見「AI CUP 報名系統流程」連結。（請上 AI CUP 官網「相關網站」→ AI CUP 報名系統流程）

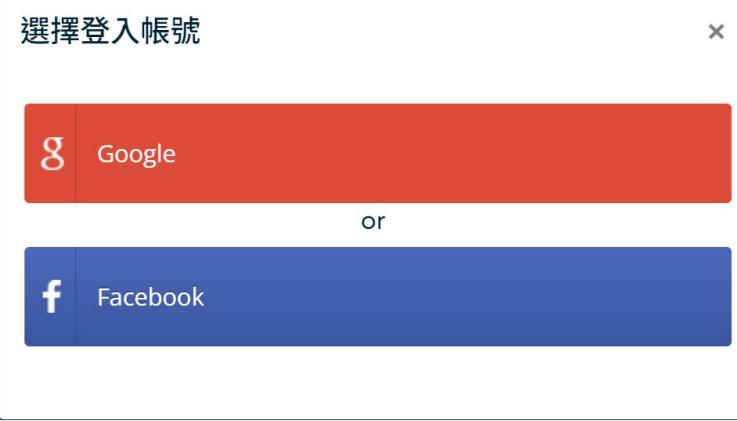
※ 如因報名操作有誤導致隊伍創建有問題，主辦單位「無法協助合併/拆分/刪除隊伍」，敬請見諒。

※ 如有報名相關疑問請來訊：[aicupauto305@gmail.com](mailto:aicupauto305@gmail.com)。

The screenshot shows the T-Brain AI Competition website's homepage. At the top, there is a yellow banner with text about registration requirements. Below the banner, there is a form with a checked checkbox labeled "我同意 參賽者使用條款" (I agree to the participant usage terms). To the right of the form is a large teal button labeled "立即組隊" (Join Team). Further to the right, there is a blue "Sign In" button, which is also highlighted with a red box. The main content area features a dark header with navigation links: Home, Competitions (which is the active page), Discussion, Datasets, Success Story, and a "Sign In" button. The main title "Competitions" is centered below the header. Below the title, there is a section titled "進行中的競賽" (Ongoing Competitions) with a card for the "根據區域微氣候資料預測發電量競賽" (Predicting electricity generation based on regional micro-climate data competition). The card displays the start date (9/2/2024), end date (11/28/2024), total reward (35.00 萬元(NTD)), number of teams (174), and a timer showing 69 days, 11 hours, 56 minutes, and 26 seconds remaining. A small green triangle icon with the text "進行中" (In Progress) is visible on the left side of the card.

## 步驟十：

進行登入(請使用與報名系統  
一相同之帳號登入)



G 使用 Google 帳戶登入

登入

繼續使用「[trendmicro.com.tw](http://trendmicro.com.tw)」

電子郵件地址或電話號碼

忘記電子郵件地址？

如要繼續進行，Google 會將您的姓名、電子郵件地址、語言偏好設定和個人資料相片提供給「[trendmicro.com.tw](http://trendmicro.com.tw)」。

建立帳戶

下一步

繁體中文 ▾

說明 隱私權設定 條款

步驟十一：

填寫資料  
並勾選我同意  
按下確認

確認後即報名成功

步驟十二：

點選根據區域微氣候資料預測  
發電量競賽進入競賽說明頁



請確認是不是與報名系統一所使用之信箱相同

建立帳號

Email: [Redacted]@gms.ndhu.edu.tw

姓 \*

名 \*

【請填寫真實姓名以利頒發獎項】

我是  學生  業界人士

T-Brain AI實戰吧平台服務 使用條款  
2018/1/1 版次1

歡迎您使用「T-Brain AI實戰吧平台服務」，一旦您進入「T-Brain AI實戰吧平台服務」即表示您同意遵守下列條款及細則，與任何不定時提供給您的政策、準則及更新條款及相關比賽規範，包括（但不限於）服務政策和法律聲明（下稱「條款」）如下：

壹、定義

1. 本規範中所稱之「T-Brain AI實戰吧平台服務」，係指由趨勢科技股份有限公司所提供之服務，用以協助「用戶」進行機器學習 Dataset、Script、意見交流與競賽的平台（以下簡稱「本平台」），及達成下列第貳點目的之範圍內，所使用之相關硬體設備、應用軟體及系統等服務。

我同意

確認

Success Story Sign In

## Competitions

進行中的競賽

Hide All

根據區域微氣候資料預測發電量競賽 >

開始 9/2/2024 11/28/2024 結束

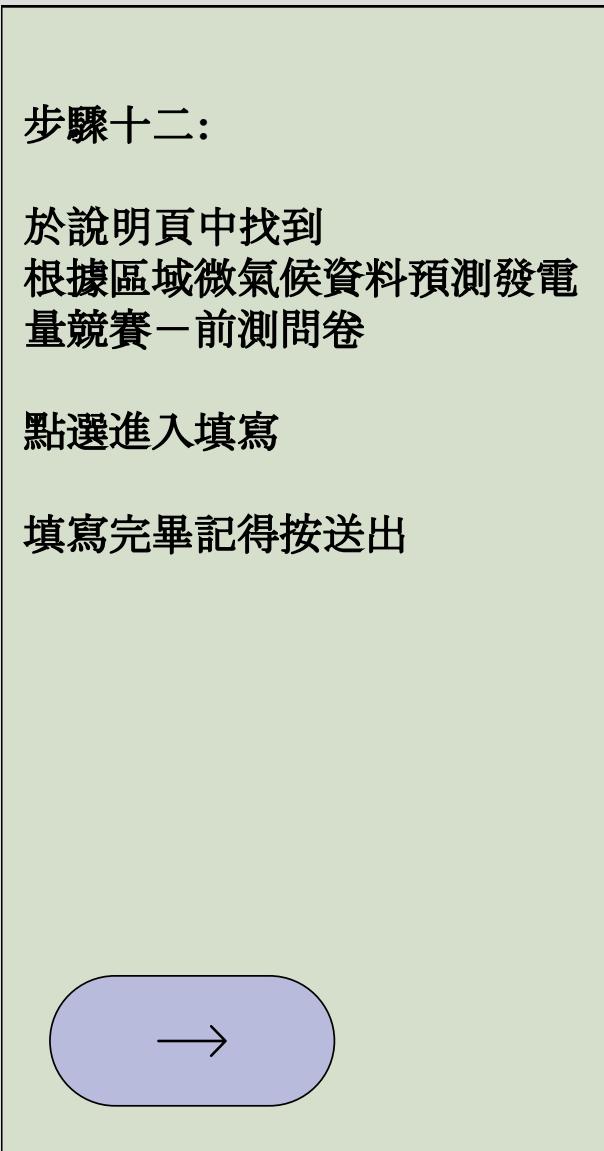
35.00 萬元(NTD) TOTAL REWARD

42 TEAMS

距離比賽結束 79 天 00 時 51 分 38 秒

太陽能光電裝置的發電量與微氣候息息相關，除了太陽輻射之外，溫度與濕度會影響太陽能光電裝置光電效應的進行，而風力又會影響溫度與濕度，另外還需考慮落塵量和雨量的影響。由於花蓮幅員遼...

(More)



TBrain AI實戰吧

Home Competitions Discussion Datasets Success Story Sign In

## 根據區域微氣候資料預測發電量競賽

進行中

Overview Leaderboard Download Dataset

### 競賽說明

太陽能光電裝置的發電量與微氣候息息相關，除了太陽輻射之外，溫度與濕度會影響太陽能光電裝置光電效應的進行，而風力又會影響溫度與濕度，另外還需考慮落塵量和雨量的影響。由於花蓮幅員遼闊，擁有各種複雜的地形地貌，本計畫將在各種地形架設各種感測器與太陽能光電裝置，建立不同地區的微氣候資料集與收集太陽能光電裝置的發電量，做為本次比賽之資料使用。本競賽提供2024年間，在17個地點收集的區域微氣候資料與光電發電量作為訓練資料，比賽內容為預測指定之時間與地點的發電瓦數，並且以誤差值為排名依據。

42 參賽隊伍

總獎金  
新台幣 35.00 萬元

開始 9/2/2024 結束 11/28/2024

### 報名規範

- 報名時具中華民國學籍之在學學生(研究生和高中生皆可)、或社會人士，凡年滿18歲皆可報名參加。若為未滿18歲之未成年人，經法定監護人同意後可報名參加。（趨勢科技公司員工除外）。
- 參賽隊伍人數1至4人。
- 參賽者只能加入一個參賽隊伍，一旦加入隊伍後則無法更換隊伍。
- 報名截止後，不可再變更團隊成員名單及人數。
- 比賽期間，隊伍不可合併與分割。
- 基於學校課程要求而參與比賽的參賽者，請於報名系統填寫課程代碼、並依循該課程之組隊要求，以利課堂評分。

### 報名方式

- 報名隊伍每位隊員皆須於『T-Brain AI實戰吧』(<https://tbrain.trendmicro.com.tw/>) 以 Google 或 Facebook 帳號註冊會員。
- 報名隊伍每位成員須於『AI CUP報名系統』(<https://go.aicup.tw/>) 登入報名，並依照報名頁面所示項目，逐項填寫每位隊員之資料(註1)，並完成組隊流程，以進行競賽登錄。
- 隊伍於『AI CUP報名系統』完成競賽登錄後之下個工作日，各隊伍成員即可於『T-Brain AI實戰吧』參與本次競賽。
- 參賽組別分為學生組、社會人士組。學生組需要全隊皆為學生身分；隊伍內只要有一名非學生身分者，則全隊列為社會人士組。
- 報名後，請每位參賽者協助填寫「**根據區域微氣候資料預測發電量競賽－前測問卷**」



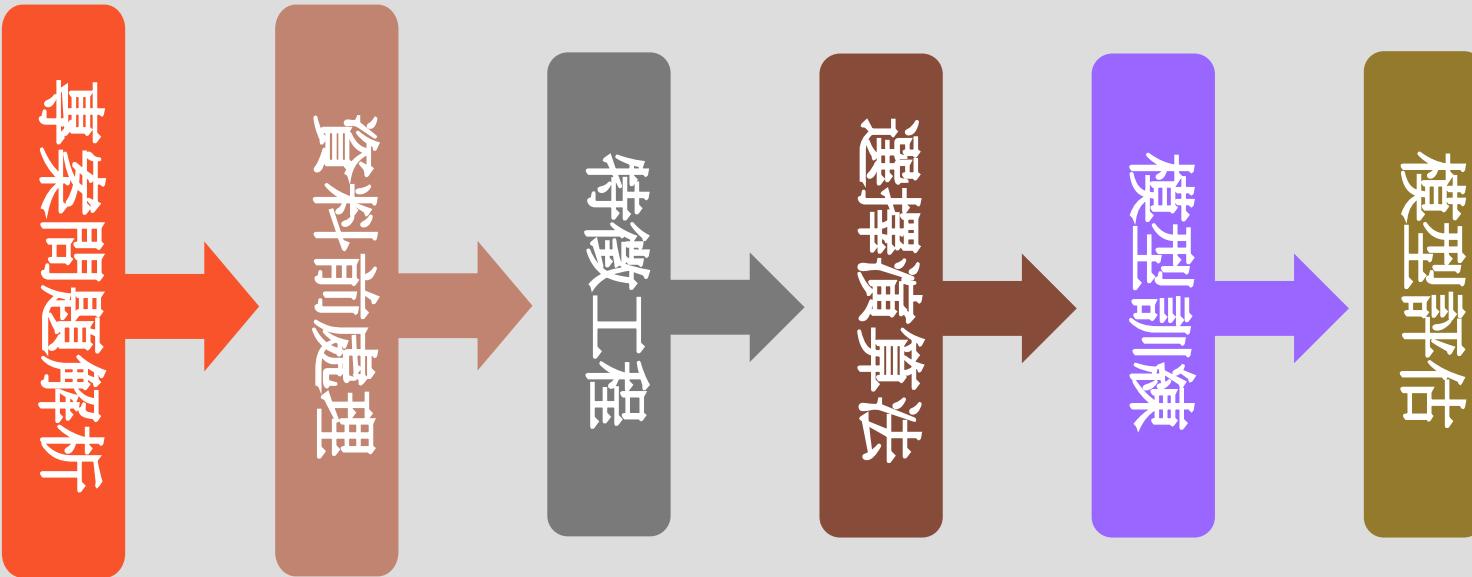
AI CUP  
2024

教育部大專校院人工智慧競賽  
根據區域微氣候資料預測發電量

機器學習步驟



# 機器學習專案建構流程



# 機器學習專案建構流程

專案問題解析

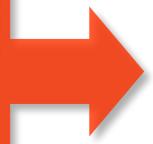
資料前處理

特徵工程

選擇演算法

模型訓練

模型評估



# 專案問題解析

- 確定問題目標



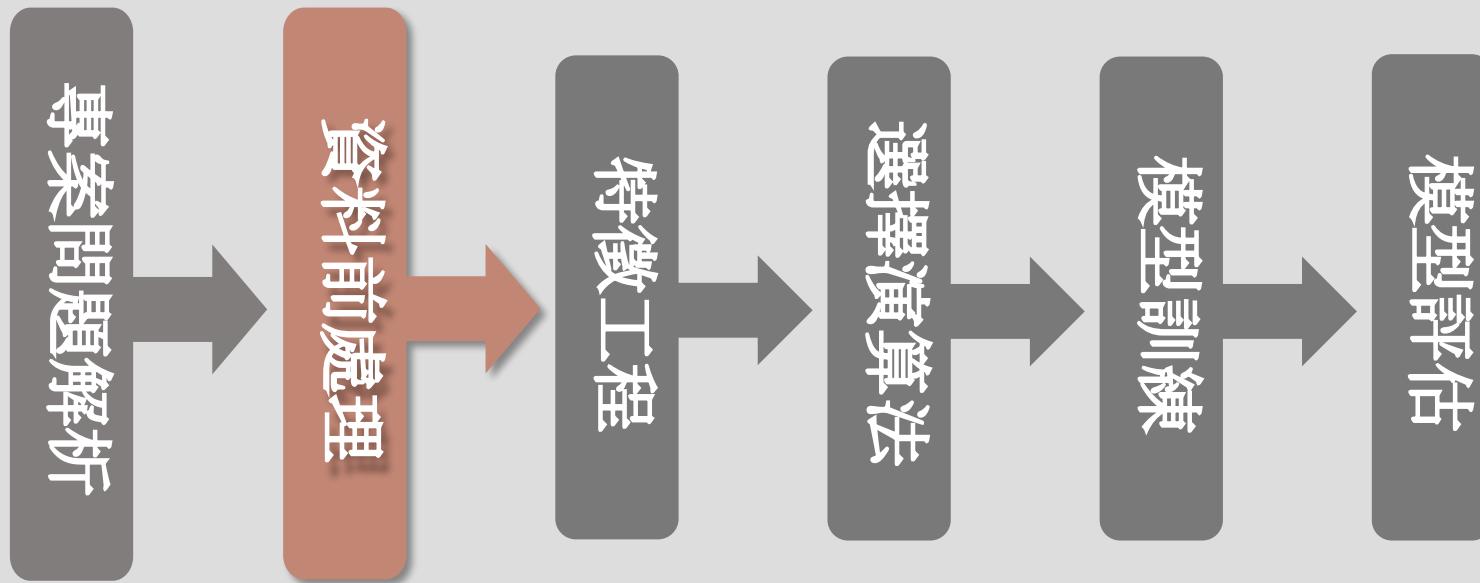
例如：利用保險公司的客戶交易資料，開發機器學習系統達到「提升業績」的目標。

- 可行性評估

考量資料量多寡、軟硬體設備、投入人力資源等，評估專案開發的可行性。



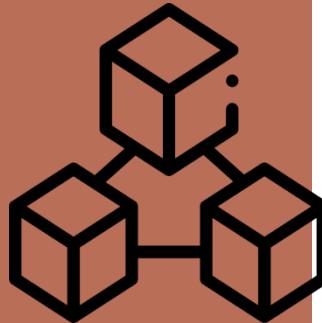
# 機器學習專案建構流程



# 資料前處理

## 資料結構化

將半結構資料和非結構化資料都轉換成結構化資料。



## 資料補齊

將資料表中空缺的欄位進行填補。



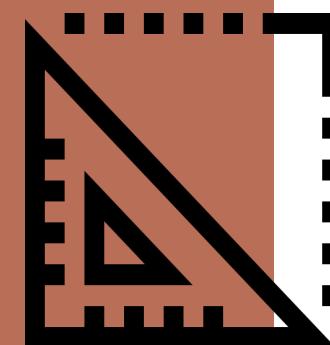
## 資料過濾

將資料中跟結果不相干的特徵資料過濾掉。



## 資料正規化

將資料處理限定於一定範圍內，用於加速演算法的收斂



## 資料結構

- 結構化資料

以矩陣結構儲存在資料庫的資料，可以透過二維表格結構來顯示。

- 半結構化資料

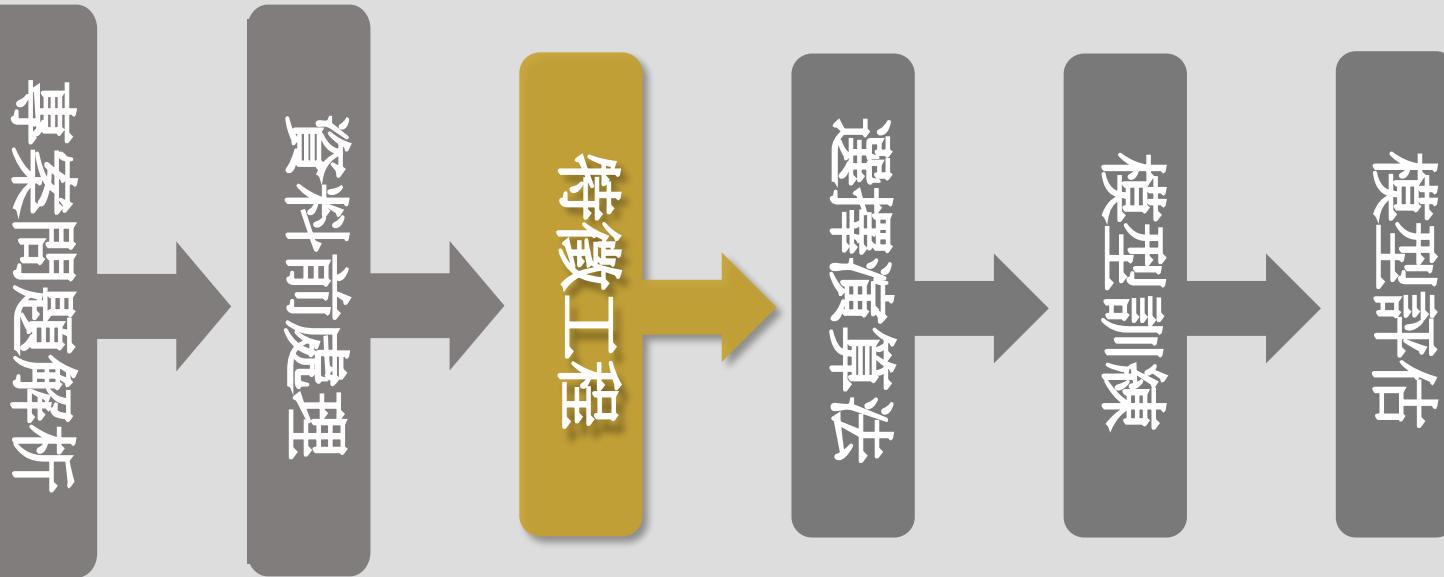
按照一定的結構儲存，但不是二維的表格結構。

- 非結構化資料

未經整理過的資料。例如：圖片、文字、語音等。



# 機器學習專案建構流程



# 特徵工程

特徵向量化

將特徵**量化**，例如ID、性別、區域、職業等。

特徵重要性評估

每個特徵對於目標列的**影響程度**。例如可透過邏輯迴歸的**模型係數**進行判斷。

特徵衍生

利用現有的特徵進行某種組合，**產生具有新含意的特徵**。例如利用產品的銷售記錄和銷售時間，可以組合算出產品熱度。

特徵維度縮減

處理高維度數據時，**挖掘關鍵欄位**，用以**減少輸入維度**。

目的在：

- (1) 確保特徵間的相互獨立
- (2) 減少計算量
- (3) 去除雜訊



# 機器學習專案建構流程

專案問題解析

資料前處理

特徵工程

選擇演算法

模型訓練

模型評估



# 演算法選擇

## 常規演算法

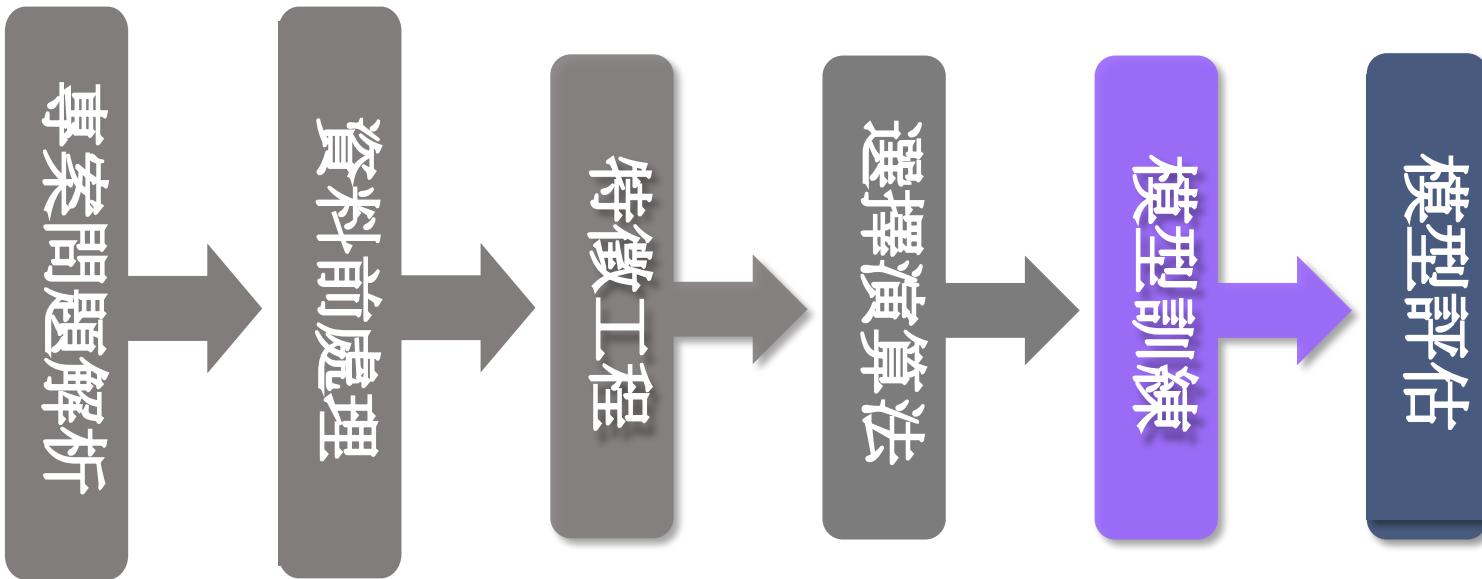
- 「例如：  
分類演算法  
分群演算法  
迴歸演算法  
文字處理演算法」

## 深度學習演算法

- 「例如：  
深度神經網路  
卷積神經網路  
遞歸神經網路」



# 機器學習專案建構流程



# 模型訓練與評估

## 模型參數設定

輸入大小、模型結構、學習率和迭代次數等

## 模型參數設定

尋求效率更好的模型

## 模型參數設定

評估指標包含：  
正確率 (Accuracy)  
精確率 (Precision)  
召回率 (Recall)  
F值 (F-measure) 等。





THE  
—  
END.

