2024 年【科學探究競賽-這樣教我就懂】

普高組 成果報告表單

題目名稱： 嘿颱風，你假不假 - 颱風假預測之大數據分析

一、摘要

「颱風假」在每年的七~九月間常成為民眾熱門話題，雖然氣象上有詳細的颱風假判斷標準，但是否放假的決策總是困擾民眾。為了讓民眾有更好的事前規劃，我們跨學科領域，利用資訊科技所學 Python 建立一套預測颱風假的模型，考慮包括雨量、風速、颱風路徑、區域等多變項因素。首先利用爬蟲程式技術收集大量的氣象數據和相關資訊，並整合了颱風的預測路徑。而後，我們使用大數據分析和機器學習工具建立了預測模型，提前預測颱風假的發佈時機。並加入政治變因。旨在創造出具有前瞻功能的預測系統。

二、探究題目與動機

「台灣位於亞熱帶，每年經歷 20 到 30 個颱風生成，其中更有 4 到 5 個會直接影響台灣，造成嚴重的生命財產損害。」（吳俊傑，2021）因此，台灣政府需仔細考慮是否應當放颱風假。 除了取決於氣象局提供的最低標準，台灣地形多變，地區間存在巨大差異，需要更謹慎的應對措施。此外，部分政治因素也可能影響到縣市政府首長的決定。

「近年來，人工智慧與機器學習的快速發展，可以應用於各種科學領域來幫助解決問題，在天氣預報上也有其可應用之處。」（楊天瑞，2021）為了更有效地應對颱風，我們希望能夠運用程式開發，建立一套綜合性的準則，以協助各地區判斷是否應該放颱風假。這個程式可以根據氣象數據、地理特徵、政治因素等多種變因，自動分析並提供建議，來預測是否應該停止上班上課，使颱風災害和損失最小化。

三、探究目的與假設

（一）利用決策樹來剖析各變因對颱風假的影響

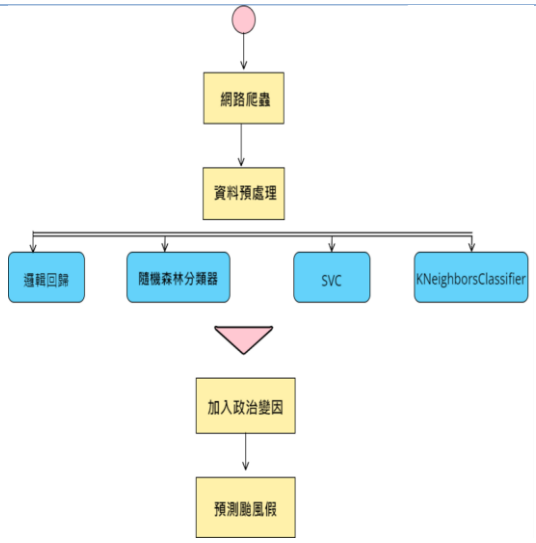
（二）透過機器學習及大數據分析預測是否放颱風假

（三）利用機器學習完成颱風假預測系統的開發

四、探究方法與驗證步驟

我們採用系統性的科學方法進行規劃，實驗按照文獻分析法、定性分析法、實證研究方法進行。如下圖一、二所示。根據下圖一，首先我們利用爬蟲抓取歷年的颱風資料，並輸出成 Excel 檔。接下來我們利用下圖二的四個模型去製作颱風假的預測模型，並測驗其與真實數據的準確度。最後，我們製作加入政治因素的模型，並比對其前後差異進行分析。

圖一、研究流程圖



圖二、系統實作流程圖

（一）爬蟲程式撰寫

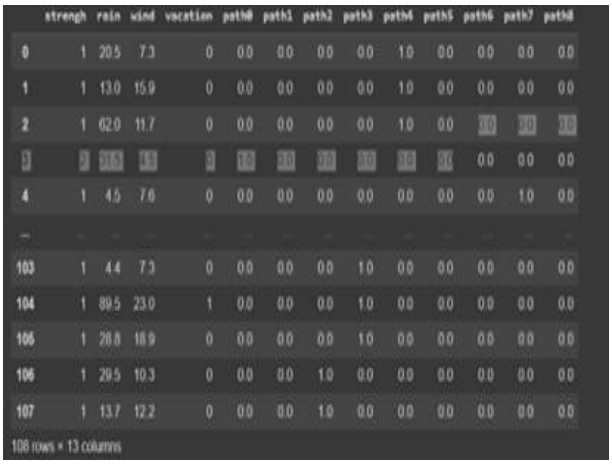
```
soup = BeautifulSoup(html_content, 'html.parser')

# Extract table
table = soup.select_one("#warning_typhoon_list_table")

# Extract headers
headers = [th.text.strip() for th in table.select("thead th")]

# Extract rows
rows = []
for tr in table.select("tbody tr"):
    row_data = [td.text.strip() for td in tr.select("td")]
    rows.append(row_data)

import pandas as pd
pd.DataFrame(rows, columns=headers)
import pandas as pd

# 創建 DataFrame
df = pd.DataFrame(rows, columns=headers)

# 將 DataFrame 保存為 Excel 檔案
df.to_excel("typhoon_data.xlsx", index=False)
```

圖三、爬蟲程式

利用爬蟲將資料從交通部中央氣象署的颱風資料庫中，把歷次颱風的名稱和日期找出來、列出最大風速、累積雨量、侵台路徑等資訊並匯入模型。下圖四為爬蟲出來的資料。

年份	颱風編號	颱風名稱	Column 5	臺路徑分	警報期間	Column 8	近臺強度	最低氣壓	最大風速	風暴風半	風暴風半	報發布報數
2023	202314	小犬(KOI	KOINU	4	2023-10-0	2023-10-0	中度	930	48	250	90	29
2023	202311	海葵(HAI	HAIKUI	4	2023-09-0	2023-09-0	中度	945	43	180	60	29
2023	202309	蘇拉(SAO	SAOLA	---	2023-08-2	2023-08-3	強烈	915	53	200	80	22
2023	202306	卡努(KH	KHANUN	特殊	2023-08-0	2023-08-0	中度	930	48	280	100	22
2023	202305	杜蘇芮(D	DOKSUR	7	2023-07-2	2023-07-2	中度	935	48	300	100	32
2023	202302	瑪娃(MA	MAWAR	---	2023-05-2	2023-05-3	中度	945	43	300	100	16
2022	202220	尼莎(NES	NESAT	---	2022-10-1	2022-10-1	中度	970	33	200	70	11
2022	202212	梅花(MU	MUIFA	---	2022-09-1	2022-09-1	中度	945	43	150	50	20
2022	202211	軒嵐諾(H	HINNAM	6	2022-09-0	2022-09-0	強烈	915	55	300	100	21
2021	202118	圓規(KOI	KOMPAS	---	2021-10-1	2021-10-1	輕度	975	30	250	80	13
2021	202114	璨樹(CH	CHANTH	6	2021-09-1	2021-09-1	強烈	915	58	200	70	24
2021	202109	盧碧(LUI	LUPIT	---	2021-08-0	2021-08-0	輕度	988	20	80	---	10
2021	202106	烟花(IN-	IN-FA	---	2021-07-2	2021-07-2	中度	945	43	200	70	22
2021	202103	彩雲(CH	CHOI-W	---	2021-06-0	2021-06-0	輕度	995	20	100	---	11

圖四、颱風資料



	strengh	rain	wind	vacation	path0	path1	path2	path3	path4	path5	path6	path7	path8
0	1	20.5	7.3	0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
1	1	13.0	15.9	0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
2	1	62.0	11.7	0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
3	2	33.0	8.1	0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
4	1	4.5	7.6	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
103	1	4.4	7.3	0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
104	1	89.5	23.0	1	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
105	1	28.8	18.9	0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
106	1	29.5	10.3	0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
107	1	13.7	12.2	0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0

108 rows × 13 columns

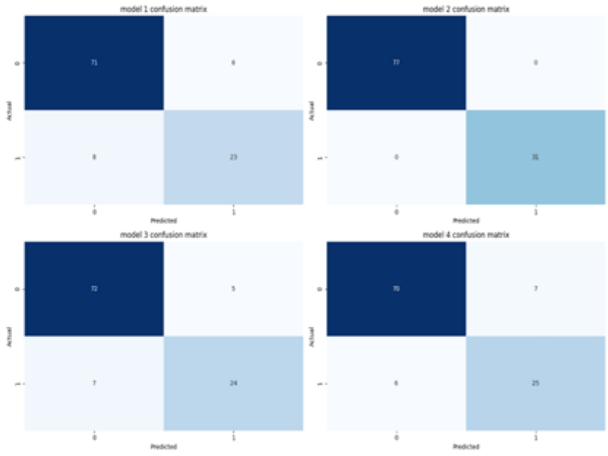
圖五、資料預處理

圖五使用 OneHotEncoding 和 LabelEncoding 進行資料預處理。OneHot Encoding 拆分每個路徑成獨立標籤，而 Label Encoding 將強度等級表示為 0、1、2。

```python
from sklearn.metrics import confusion_matrix
import seaborn as sns

# 用每個模型進行預測
predictions = []
accuracies = []

for i, model in enumerate(models):
    prediction = model.predict(x_scaled)
    accuracy = accuracy_score(y, prediction)

    predictions.append(prediction)
    accuracies.append(accuracy)
    print(f'模型 {i+1} 的準確度: {accuracy}')

# 顯示混淆矩陣
plt.figure(figsize=(15, 10))

for i, (model, prediction) in enumerate(zip(models, predictions), 1):
    plt.subplot(2, 2, i)
    cm = confusion_matrix(y, prediction)
    sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", cbar=False)
    plt.title(f'model {i} confusion matrix')
    plt.xlabel('Predicted')
    plt.ylabel('Actual')

plt.tight_layout()
plt.show()
```

圖六、模型準確度和混淆矩陣之程式

圖六中使用了 Scikit-Learn 庫中的混淆矩（confusion_matrix）和 Seaborn 庫中的熱力圖（heatmap）來評估多個模型的預測效果。 模型的預測結果將被添加到 predictions 列表中並計算該模型的準確度。我們可以藉由此部分得到圖八中每個模型的準確度和圖七中四個模型的混淆矩陣。



圖七、混淆矩陣

左上角為 TP（True Positive）實際為正例，模型預測也為正例。
右上角為 FN（False Negative）實際為正例，但模型預測為負例
左下角為 FP（False Positive）實際為負例，但模型預測為正例。
右下角為 TN（True Negative）實際為負例，模型預測也為負例。
由圖六可以得知跟圖七相通的結論。

```
模型 1 的準確度: 0.8703703703703703
模型 2 的準確度: 1.0
模型 3 的準確度: 0.8888888888888888
模型 4 的準確度: 0.8796296296296297
```

圖八、實測後準確度

由上圖八，我們可以知道模型 2（隨機森林）的準確度最高，模型 3（SVC）的準確度次之，模型 1（邏輯回歸）第三，模型 4（KNeighborsClassifier）為末。在本次實作中並不以單一模型為標準，而是呈現此四模型的判斷結果供民眾參考。

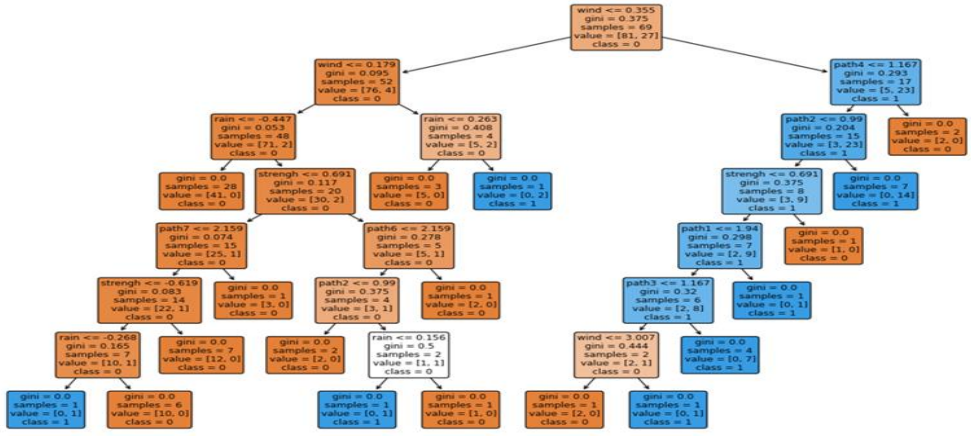
由下表三可得知準確度由大到小分別是：**隨機森林 > SVC > KNeighborsClassifier > 邏輯回歸，且隨機森林的錯誤率非常之低**。

表 三：邏輯回歸、隨機森林分類器、支持向量機分類器、K 最近鄰分類器比較

	TP	FN	FP	TN
邏輯回歸	71	6	8	23
隨機森林	77	0	0	31
SVC	72	5	7	24
K 最近鄰分類器	70	7	6	25

（二）決策樹模型

採用決策樹模型，找出影響颱風假的最大變因。如下圖九每個內部節點表示對數據的一個測試，每個分支代表測試的一個結果。每個節點上，決策樹根據某一個特徵的某個閾值進行分割。並且我們使用特徵重要性排名，排序出影響颱風假的前三大因素。雖決策樹在訓練數據上表現得很好，但有時過於複雜對新數據的泛化能力不足。在接受未接觸過的數據時，準確度會下降，所以**本次實作我們並未使用此方式去呈現，僅使用決策樹剖析各變因對颱風假的影響程度**。如下圖九可知對颱風假影響最大的為最大風速、累積雨量、路徑。



```
Feature: wind, Importance: 0.5042119529278564
Feature: rain, Importance: 0.25855649471556996
Feature: strengh, Importance: 0.06402272977730224
```

圖九、決策樹

	strengh	rain	wind	vacation	election	path0	path1	path2	path3	path4	path5	path6	path7	path8
0	1	139.5	26.4	0	0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
1	1	196.5	31.4	0	0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
2	1	38.0	27.5	0	0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
3	2	43.0	25.9	0	0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	1	7.0	20.4	0	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
103	1	0.5	27.3	0	0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
104	1	1.0	10.2	0	0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
105	1	171.0	24.8	1	0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
106	1	416.0	42.2	1	0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
107	1	19.0	27.1	0	0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0

108 rows × 14 columns

圖十、加入選舉變因

我們想了解縣市政府是否會因地方選舉等因素而去決定颱風假與否。像是：放了假，當天卻無風無雨、達放假標準卻不放，使民眾損失慘重。這些會成社會上的輿論，進而影響民調，於是我們認為選縣市首長那年，颱風假較容易施行。所以我們在原模型基礎上又加了一個變因 - 當年是否為選舉年，根據上圖十我們看見了加入選舉年（election）的資料。

表四、是否為選舉年

path7	強烈	102.5	38	1	0

```
請輸入強度(輕度=0, 中度=1, 強烈=2):1
請輸入累積雨量(mm):102.5
請輸入最大風速(m/s):38
請輸入侵台路徑編號(0到8之間):0
模型 1 的預測結果: [0]
模型 2 的預測結果: [0]
模型 3 的預測結果: [0]
模型 4 的預測結果: [0]
```

```
請輸入強度(輕度=0, 中度=1, 強烈=2):1
請輸入累積雨量(mm):102.5
請輸入最大風速(m/s):38
是否為選舉年(是=1, 否=0):1
請輸入侵台路徑編號(0到8之間):0
模型 1 的預測結果: [1]
模型 2 的預測結果: [1]
模型 3 的預測結果: [1]
模型 4 的預測結果: [0]
```

path5	強烈	57.9	39.9	1	1

```
請輸入強度(輕度=0, 中度=1, 強烈=2):2
請輸入累積雨量(mm):57.9
請輸入最大風速(m/s):39.9
請輸入侵台路徑編號(0到8之間):5
模型 1 的預測結果: [1]
模型 2 的預測結果: [1]
模型 3 的預測結果: [0]
模型 4 的預測結果: [1]
```

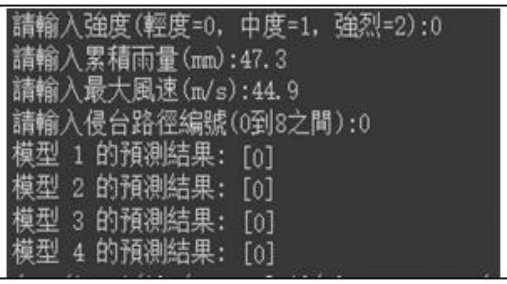
```
請輸入強度(輕度=0, 中度=1, 強烈=2):2
請輸入累積雨量(mm):57.9
請輸入最大風速(m/s):39.9
是否為選舉年(是=1, 否=0):1
請輸入侵台路徑編號(0到8之間):5
模型 1 的預測結果: [1]
模型 2 的預測結果: [1]
模型 3 的預測結果: [0]
模型 4 的預測結果: [0]
```

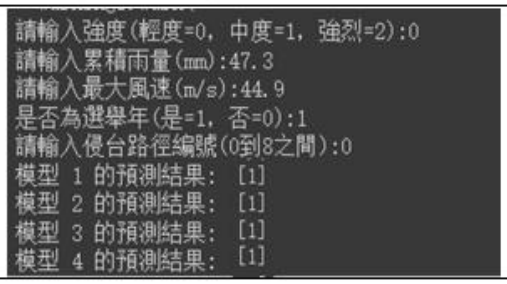
path3	中度	53	40.8	1	1

```
請輸入強度(輕度=0, 中度=1, 強烈=2):1
請輸入累積雨量(mm):53
請輸入最大風速(m/s):40.8
請輸入侵台路徑編號(0到8之間):3
模型 1 的預測結果: [0]
模型 2 的預測結果: [0]
模型 3 的預測結果: [0]
模型 4 的預測結果: [0]
```

```
請輸入強度(輕度=0, 中度=1, 強烈=2):1
請輸入累積雨量(mm):53
請輸入最大風速(m/s):40.8
是否為選舉年(是=1, 否=0):1
請輸入侵台路徑編號(0到8之間):3
模型 1 的預測結果: [1]
模型 2 的預測結果: [1]
模型 3 的預測結果: [0]
模型 4 的預測結果: [0]
```

path0	輕度	47.3	44.9	1	1





根據上表四，**有三個案例在選舉年時呈現較高的放假機率，顯示選舉年受到特殊環境或條件的影響，促使當局傾向休假措施**。然而，其中一個強颱風案例未考慮選舉年變數反而顯示更高的放假機率，趨勢可能受到氣象事件和其他因素的影響，因此解讀仍需謹慎。

（三）研究結果

1、本研究結合氣象和政治因素建立全面模型，迅速預測颱風假。**以隨機森林為最佳預測模型，正確率近 100%。**

2、分析最大風速和累積雨量，加入侵台路徑、颱風強度、選舉年提升預測力。從結果得知，影響颱風假最大的變因是最大風力。

3、從研究結果得知，政治因素影響偏高，**顯示放颱風假的機率仍與政治因素有一定關聯。**

五、結論與生活應用

（一）結論

1、實驗結果可應用於日常生活中，可用於颱風來臨時自行預測放假結果。

2、政府放假準確率平均達八成以上，雖然部分人認為政府放假不準確，但實驗結果顯示政府放假準確率高。

3、政治因素雖會降低些許放假準確度，但總體而言，政府並未亂放假

（二）生活應用

1、可開發颱風防災智能系統，結合氣象數據、政治事件資訊和其他相關資訊。

2、提供更全面的颱風假機率預測和應對建議。

3、這樣的系統有望在颱風來臨時提供更具效益的應急指引。

參考資料

吳俊傑（譯）（2021）。**颱風**（原作者：Kerry Emanuel）。 國立台灣大學出版中心。（原著出版年：2005）

楊天瑞（2021 年 7 月）。利用觀測環境參數與機器學習預報大台北都會區的午後雷陣雨。https://reurl.cc/67OqNO