資料科學概論 — 期末專題
主題：利用雨量預測空氣品質指標

隊伍名稱

熬夜寫 Code 來杯 Java

組員

F74114037 江曉明

F74116275 陳柏淮

F74111071 楊承翰

壹、 主題介紹

期末專題主題為「利用雨量預測空氣品質指標」

專題目的：

本專題旨在探討降雨量對空氣品質的影響，分析降雨是否能夠洗滌空氣、改善空氣品質。此外，我們期望透過機器學習的方式建構預測模型，以有效預測空氣品質指標 AQI。

問題敘述：

本專題聚焦於以歷史資料進行預測，透過輸入包含前一日或數日的空氣品質指標 AQI、各項空氣污染物濃度數值，以及降雨量等相關數據，來預測隔日的空氣品質指標 AQI。

輸入：降雨量、AQI 及其他空氣污染物指標

輸出：隔日的 AQI 預測值

貳、 資料蒐集與處理

我們資料來源有三處：

1. 環境部-統計查詢網
2. 環境部-空氣品質指標(AQI)(歷史資料)
3. 中央氣象署-每日降雨量

利用 Python 的 Selenium 套件從環境部-統計查詢網[1]中爬取從 2017 年 1 月 1 日至 2024 年 9 月 30 日測站的 AQI 數值（共有 60 個測站）以及從中央氣象署網站[3]中，爬取那 60 個測站所對應縣市的降雨量。

最後再從環境部-空氣品質指標(AQI)(歷史資料)[2]中，下載相關的相關檔案，以補充空氣汙染物指標的資料。

在資料處理方面，我們發現爬取的 AQI 資料中缺少 2021 年 5 月份的數據，為確保資料完整性，我們利用從下載的資料中進行補充。此外，由於下載的資料是以每小時為單位記錄 AQI 及相關空氣污染物指標（註 1），我們將每個測站當天的空氣污染物指標進行平均處理，並將該平均值視為當日的空氣污染物指標。

值得注意的是，下載的資料中包含超過 60 個測站的數據，但由於爬取的網站僅涵蓋 60 個測站，為了確保資料的一致性，我們僅保留這 60 個測站的資料，並刪除其他無關的測站數據。

最終，我們將 2017 年 1 月 1 日至 2024 年 9 月 30 日的資料作為模型的訓練集與驗證集，而 2024 年 10 月與 11 月的資料則作為測試集，用以評估模型的預測效果。

註1：下載資料內的欄位分別如下

測站名稱 (sitename)	縣市 (county)	空氣品質指標 (AQI)	主要污染物 (pollutant)	空氣品質狀態 (status)
二氧化硫 (SO_2)	一氧化碳 (CO)	臭氧 (O_3)	臭氧8小時平均 (O_3_8hr)	懸浮微粒 (pm10)
細懸浮微粒 (pm2.5)	二氧化氮 (NO_2)	氮氧化物 (NOx)	一氧化氮 (NO)	風速 (windspeed)
風向 (winddirec)	資料建立日期 (datacreationdate)	單位 (unit)	一氧化碳 8小時平均 (CO_8hr)	細懸浮微粒 平均 (pm2.5_avg)
懸浮微粒平均 (pm10_avg)	二氧化硫平均 (SO_2_avg)	經度 (longitude)	緯度 (latitude)	測站編號 (siteid)

在初步資料處理中，已將部分欄位去除（以紅色作為記號）

測站名稱 (sitename)	縣市 (county)	空氣品質指標 (AQI)	主要污染物 (pollutant)	空氣品質狀態 (status)
二氧化硫 (SO_2)	一氧化碳 (CO)	臭氧 (O_3)	臭氧8小時平均 (O_3_8hr)	懸浮微粒 (pm10)
細懸浮微粒 (pm2.5)	二氧化氮 (NO_2)	氮氧化物 (NOx)	一氧化氮 (NO)	風速 (windspeed)
風向 (winddirec)	資料建立日期 (datacreationdate)	單位 (unit)	一氧化碳 8小時平均 (CO_8hr)	細懸浮微粒 平均 (pm2.5_avg)
懸浮微粒平均 (pm10_avg)	二氧化硫平均 (SO_2_avg)	經度 (longitude)	緯度 (latitude)	測站編號 (siteid)

去除二氧化硫平均欄位是由於大部分資料此欄位均為 nan，因此，決定直接刪除此欄位。

參、 模型建構

我們認為模型建構可以從以下幾個面向進行探討：

1. **基於不同模型**：使用 Regressioin Neural Network 或 LSTM 進行預測。
2. **基於不同預測輸出**：預測隔日的 AQI，或先預測隔日的空氣污染指標，進而推算 AQI（註 2）。
3. **基於區域劃分**：針對每個測站分別建構模型，或以縣市為單位建構模型（註 3）。

由於時間有限，我們決定每位組員各自嘗試建構一種組合，最後相互比較彼此表現最好的模型，以找出最佳解決方案。

一、 江曉明

採用「針對每個測站建構一個 Regression NN 模型預測隔日的 AQI」的方式。

輸入特徵如下：

月份 (Month)	前一日降雨量 (Precipitation)	空氣品質指標 (AQI)	一氧化碳 (CO)	一氧化碳 8 小時平均 (CO_8hr)
一氧化氮 (NO)	二氧化氮 (NO_2)	氮氧化物 (NOx)	臭氧 (O_3)	臭氧 8 小時平均 (O_3_8hr)
懸浮微粒 (pm10)	懸浮微粒平均 (pm10_avg)	細懸浮微粒 (pm2.5)	細懸浮微粒平均 (pm2.5_avg)	二氧化硫 (SO_2)

先建構一個 Regression NN 模型和 XGBoost 模型基於上述輸入特徵做訓練，訓練結束後，用 Regression NN 模型和 XGBoost 模型的結果，納入輸入特徵，再訓練一個新的 Regression NN 模型（即 Ensemble Model），此模型的結果即為預測的隔日 AQI 指標。

Regression NN 架構如下（Ensemble NN 亦使用相同類別建構）

```
class RegressionNN(nn.Module):

    def __init__(self, input_size, output_size, hidden_size = [64, 128, 256, 1024, 1024, 512, 128,
128, 64]):
        super(RegressionNN, self).__init__()

        self.hidden_layer_cnt = len(hidden_size)
        self.fc = nn.ModuleList()
        self.bn = nn.ModuleList()

        self.fc.append(nn.Linear(input_size, hidden_size[0]))
        self.bn.append(nn.BatchNorm1d(hidden_size[0]))

        for i in range(1, len(hidden_size)):
            self.fc.append(nn.Linear(hidden_size[i - 1], hidden_size[i]))
            self.bn.append(nn.BatchNorm1d(hidden_size[i]))


        self.fc.append(nn.Linear(hidden_size[-1], output_size))

        self.relu = nn.ReLU()                           # ReLU function
        self.dropout = nn.Dropout(0.3)                  # Dropout to avoid overfitting

    def forward(self, x):

        for i in range(self.hidden_layer_cnt):
            x = self.fc[i](x)
            x = self.bn[i](x)
            x = self.relu(x)
            x = self.dropout(x)
        x = self.fc[-1](x)

        return x
```

損失函數為 MSE、優化器為 Adam，並且使用 StepLR 作為學習率調整器使模型容易收斂。最後，在每次 Epoch 訓練結束時，令模型在驗證集上測試，並保留在驗證集

上最低 MSE 時的模型權重。

二、 陳柏淮

採用「針對每個縣市和每個汙染物指標建構一個模型預測隔日汙染物指標，並將預測結果按照公式計算 AQI」，分別測試 XGB 和 Neural Network 的結果。

輸入特徵如下：

當日降雨量 (Precipitation)	前日降雨量 (Precipitation)	空氣品質指標 (AQI)	一氧化碳 (CO)	一氧化碳 8 小時平均 (CO_8hr)
一氧化氮 (NO)	二氧化氮 (NO_2)	氮氧化物 (NOx)	臭氧 (O_3)	臭氧 8 小時平均 (O_3_8hr)
懸浮微粒 (pm10)	細懸浮微粒 (pm2.5)	風速 (Windspeed)	風向 (Winddirec)	二氧化硫 (SO_2)
當前縣市的各個測站	年份 (Year)	月份 (Month)	日 (Day)	

將同一縣市的各個測站使用獨熱編碼作為特徵，並且將時間（年、月、日）做為數值特徵。

模型結構如下：

1. XGBoost

```
xgb = XGBRegressor(
    verbosity=1,
    objective='reg:absoluteerror',
    random_state=42
)
# Define parameter distributions for random search
param_distributions = {
    "n_estimators": [100, 200, 300, 400, 500],
    "max_depth": [3, 5, 7, 9, 11],
    "learning_rate": [0.01, 0.05, 0.1, 0.2],
    "subsample": [0.6, 0.8, 1.0],
    "colsample_bytree": [0.6, 0.8, 1.0],
    "gamma": [0, 0.1, 0.2, 0.3],
}
```

```
# Perform random search
rand_search = RandomizedSearchCV(
    xgb,
    param_distributions,
    n_iter=50,
    scoring='neg_mean_absolute_error',
    cv=5,
    verbose=0,
    n_jobs=-1,
    random_state=42,
)
```

使用 RandomizedSearchCV 以及交叉驗證找出最佳超參數，並使用 XGBoost 提供的回歸模型，設定目標函數為平均絕對誤差（MAE）。

2. Regression Neural Network

```
class RegressionModel(nn.Module):
    def __init__(self, input_dim):
        super(RegressionModel, self).__init__()
        self.fc = nn.Sequential(
            nn.Linear(input_dim, 64),
            nn.ReLU(),
            nn.Linear(64, 32),
            nn.ReLU(),
            nn.Linear(32, 1)
        )

    def forward(self, x):
        return self.fc(x)
```

模型架構如上圖所示，損失函數為 MSE、優化器為 Adam，評分方式使用 MAE。

每個縣市和每個汙染物都建構一個模型，輸入相同特徵來預測下一天的各項汙染物指標，而不直接預測 AQI，最後再將各項汙染物指標的預測結果代入公式計算出 AQI。

三、 楊承翰

採用「針對每個測站建構一個 LSTM 模型預測隔日空氣狀況，並對照 AQI 換算表」的方式。

輸入特徵如下(LSTM 一共會採用過去 15 天份的資料)：

前一日降雨量 (Precipitation)	空氣品質指標 (AQI)	一氧化碳 (CO)	一氧化碳 8 小時平均 (CO_8hr)	一氧化氮 (NO)
二氧化氮 (NO_2)	氮氧化物 (NOx)	臭氧 (O_3)	臭氧 8 小時平均 (O_3_8hr)	懸浮微粒 (pm10)
懸浮微粒平均 (pm10_avg)	細懸浮微粒 (pm2.5)	細懸浮微粒平均 (pm2.5_avg)	二氧化硫 (SO_2)	

模型結構如下：

```
def create_sequences(data, seq_length):
    X_seq = []
    for i in range(len(data) - seq_length):
        X_seq.append(data[i : i + seq_length])      You, 1 秒前 • Uncommitted changes
    return np.array(X_seq)

model = Sequential(
    [
        LSTM(128, input_shape=(seq_length, len(features)), return_sequences=True),
        LSTM(64, return_sequences=True),
        LSTM(32),
        Dense(len(features)),
    ]
)

model.compile(optimizer="adam", loss="mse", metrics=["mae"])

early_stopping = EarlyStopping(monitor="val_loss", patience=5, restore_best_weights=True)
```

使用多層 LSTM 模型來提高訓練效果，損失函數為 MSE、優化器為 Adam，評分方式使用 MAE，並且加入 early stoping 的方式加快訓練過程。

一次輸入過去 15 天的資料來預測下一天的空氣狀

況，而不直接預測 AQI，這邊我測試了兩種方式，分別加入或不加入降雨量，但兩者的結果相近，有加入降雨量的部分需要後置處理負值，並且加入 early stoping 的方式加快訓練過程。

最後，透過 AQI 的轉換公式將相關的特徵經由程式插值轉換，並取當日影響最大的特徵當作當日 AQI。

註 2：AQI 是由多個空氣污染指標推算而成，因此可透過先預測空氣污染指標，再推算 AQI，達到預測目的。

註 3：由於部分測站位於同一縣市中，若以縣市為單位建構模型，即可將該縣市內所有測站的資料合併進行訓練，提升資料利用效率。

肆、 結果比較

比較這 60 個測站的 AQI 預測結果（註 4）可以得知，陳柏淮的模型為我們三人中表現最好的。

我們認為這其中的理由有兩個

1. **以縣市為單位訓練模型**：

透過以縣市為單位進行模型訓練，可以結合多個測站的資料作為訓練樣本，進一步擴大訓練數據的規模。如

此一來，不僅能提升模型的泛化能力，還能大幅改善預測結果的精確性。

2. **採用先預測空氣汙染物指標再計算隔日 AQI 的方式：**

由於 AQI 的計算方法比較特殊，是採用「空氣汙染物指標取最大」的方式，因此，若是基於數個空氣汙染指標來直接預測 AQI 的方式（即江曉明的方式），可能導致某一空氣汙染指標明明超標嚴重，卻因為其他汙染物指標良好，從而降低預測的 AQI 數值，例如：江曉明的預測結果圖中 2024 年 11 月 10 日至 2024 年 11 月 15 日的桃園地區。

除此之外，雖然我們彼此輸入特徵稍微不太一樣，但透過觀察特徵與隔日 AQI 的散佈圖與測試使用特徵的效果後，我們認為彼此模型輸入特徵的差異對於預測結果的差異並沒有太大的影響，例如：風向、年份等。

值得注意的是，江曉明與楊承翰的模型在對於 AQI 劇烈變化時，預測結果往往不太好（預測過於保守）。我們認為

- 江曉明的模型是由於基於數個空氣汙染指標來直接預測 AQI 的方式。（具體解釋已於上述提及）

- 楊承翰的模型則是由於使用的是 LSTM，導致其容易受過去資料影響，從而當 AQI 短時間劇烈變化的時候，無法正確預測 AQI。

最後，僅管陳柏淮的模型效果已經很不錯了，但我們認為，若是還想再進一步提升預測效果，在模型的選擇上，可以考慮將 XGBoost 跟 NN 模型的結果結合起來，使用 Ensemble Model 的方式來預測，效果或許會更好。

註 4：60 個測站的 AQI 預測結果放置於 Predict_Result.pdf 內。

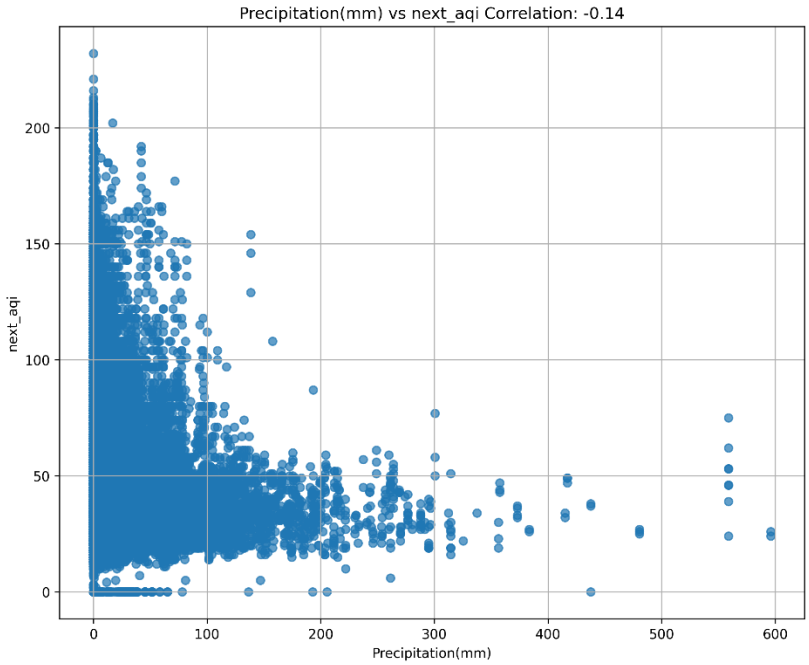
伍、 總結

本次專題的目的為「探討雨量對空氣品質的影響」與「建構能夠預測空氣品質的模型」。經過多次實驗與觀察，我們得到以下幾個結論：

- 雨量對空氣品質的影響：

  透過觀察雨量對隔日 AQI 的分布圖（下圖），我們發現雨量確實對空氣品質有正面影響，且降雨量越多，改善效果越明顯。然而，在進一步的模型測試中，將雨量作為特徵預測空氣品質時，其效果並不突出，我們推測可能

有降雨的樣本量不足，也有可能因為降雨的強度不足，附近的空氣汙染並不明顯的原因造成降雨量並不能成為有效預測空氣汙染指數的重要指標。



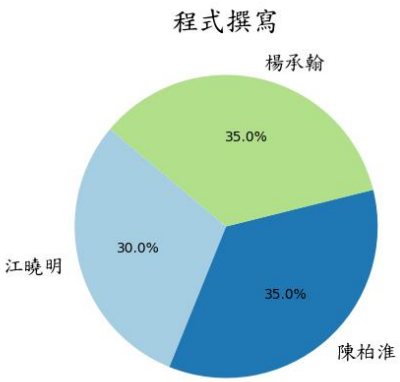
- 我們測試出來預測空氣品質指標的最佳方法為：

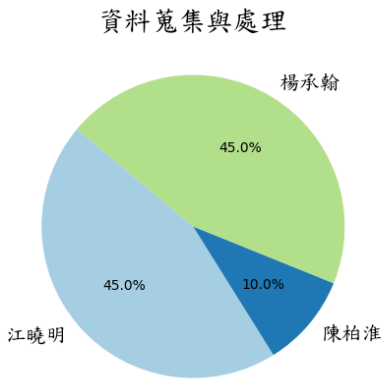
  依照各個縣市，結合位於同一縣市測站的資料，使用前一天空氣品質指標、相關的空氣汙染物指標、降雨量與日期作為輸入特徵，並將測站使用獨熱編碼做為新的輸入特徵值，預測隔日的各個空氣汙染物指標，最後基於 AQI 計算規則，推算出隔日的 AQI。

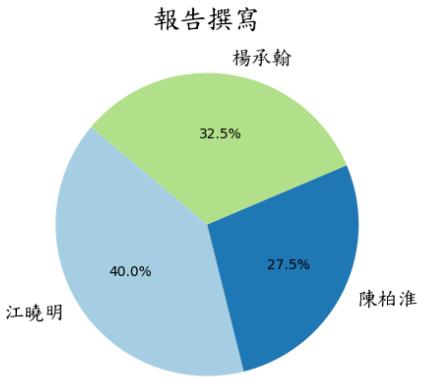
  模型選擇方面則可以採用 XGBoost 模型。

陸、 其他

分工







GitHub Repository
https://github.com/MingMinNa/IDS-Final-Project

報告錄影

https://www.youtube.com/watch?v=8ZV1o1nt2oU