**分頁 1**

傳染病風險追蹤平台



組員：陳映良、黃偉智、陳彥儒、王巧如

指導老師：簡志璁

網址：

<https://infectiousdisease-efebc629aa2d.herokuapp.com>

[**一、緒論 1**](#_90jyukv6hbt)

[動機 1](#_nnw5r6q7gv7f)

[主要功能 1](#_3in20k6gyfay)

[**二、環境工具與規劃 2**](#_c1ott4uxzjb)

[開發工具 2](#_e9dwuapd8puu)

[**三、地圖定位與LINEBOT 4**](#_fzl8plhnh7hy)

[1、 資料處理: 4](#_ggfy4zddhcvd)

[2、 網頁地圖功能: 6](#_cz15ed3ct1of)

[3、 Line Bot 9](#_bl7cnjg9bxop)

[**四、首頁圖表呈現 11**](#_t9yrciep5nac)

[首頁圖表流程圖 11](#_49ntv93hd1zj)

[近五年病例數 11](#_ljtltzi4ajoa)

[近三年月病例數 12](#_2akiho5is7z8)

[去年病例年齡層 14](#_5nnouxz6jaob)

[今年縣市病例數 15](#_2p1fz2i0x84b)

[更新圖表數據 16](#_5w37xbj2z97)

[**五、會員系統 18**](#_fyj1ttdiinv)

[工具介紹 18](#_sa4wzxx9o98u)

[會員資料架構圖 19](#_4g0gp6cklmkw)

[功能介紹 19](#_i7afdzwnmats)

[**六、風險預測系統 23**](#_s4u71a5dt7jw)

[介紹 23](#_sf9ibxvov3sw)

[系統架構與流程 24](#_4soicqjmopyw)

[風險預測UI 26](#_19omquamxujd)

[資料來源 27](#_fow81f7eogwn)

[數據處理 28](#_5n80gq1wet6v)

[Kmeans分群分析 29](#_eoqx0o17h6hu)

[線性迴歸 & 風險預測 31](#_o2om9e25d6gc)

[模型評估 32](#_kwa69n7fzn38)

[**七、未來展望 33**](#_gm771r7v7fso)

# 一、緒論

## 動機

鑒於西元2020-2023年COVID-19嚴重特殊傳染性肺炎流行，全體國民防疫意識提高，對不指是COVID-19的疾病也開始注重風險愈預防，並且臺灣國土面積不大，但交通方便且發達，國民不論是就學、工作或是旅遊都有很高的機率向其他的縣市移動，因此我們的組別志在開發架構一個方便判讀以及能隨身獲得疾病資訊的平台。

## 

## 主要功能

視覺化

使用 Google Map API 實現地圖功能

使用 AJAX 串接後台資料呈現 Javascript echart 圖表

定位功能

Google Map APII 定位

Line 定位功能

即時查詢

Line Chat Bot 即時通訊

風險預測

sklearn linear regression

會員管理

firebase authenticatin

firebase Realtime database

# 二、環境工具與規劃

## 開發工具

環境

程式語言：Python

框架：Django

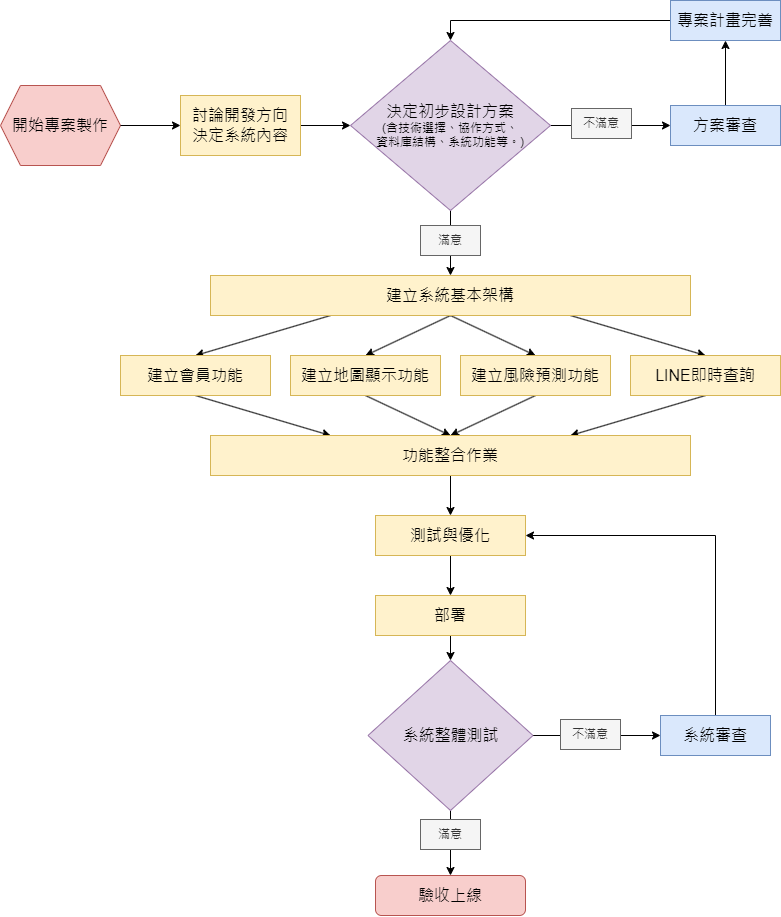
前端語法：Html, Javascript, css, echart, AJAX

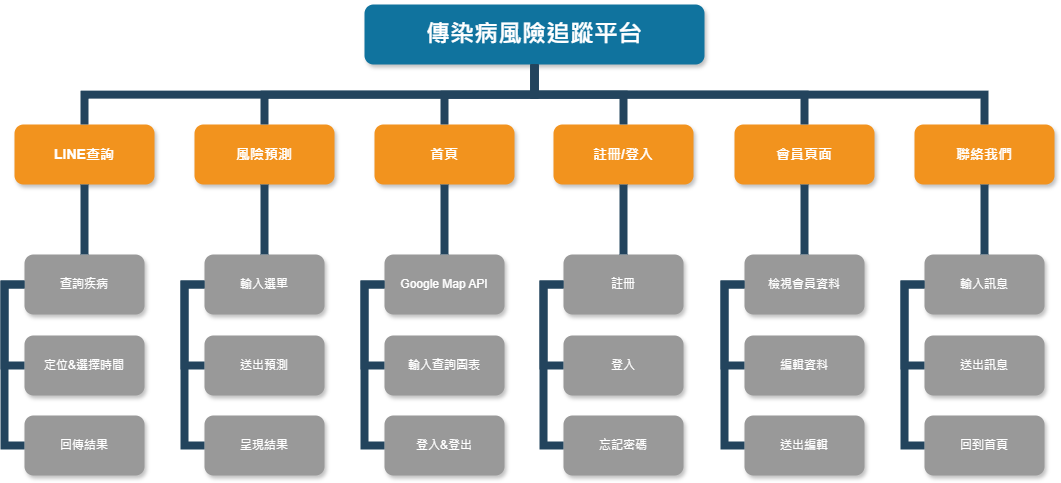
資料庫：Firebase Realtime Datebase

部屬：WSGI, Heruko

套件：

Google Map API, Line Chat Bot, pyrebase, sklearn, numpy, pandas, datetime, os, re, request, threading

**流程圖**

**結構圖**

# 三、地圖定位與LINEBOT

## 資料處理:

到疾病管制署資料開放平台抓取專案所需的資料(流感併發重症、腸病毒、登革熱、侵襲性肺炎鏈球菌)、中華郵政網站抓取臺灣各行政區經緯度資料。

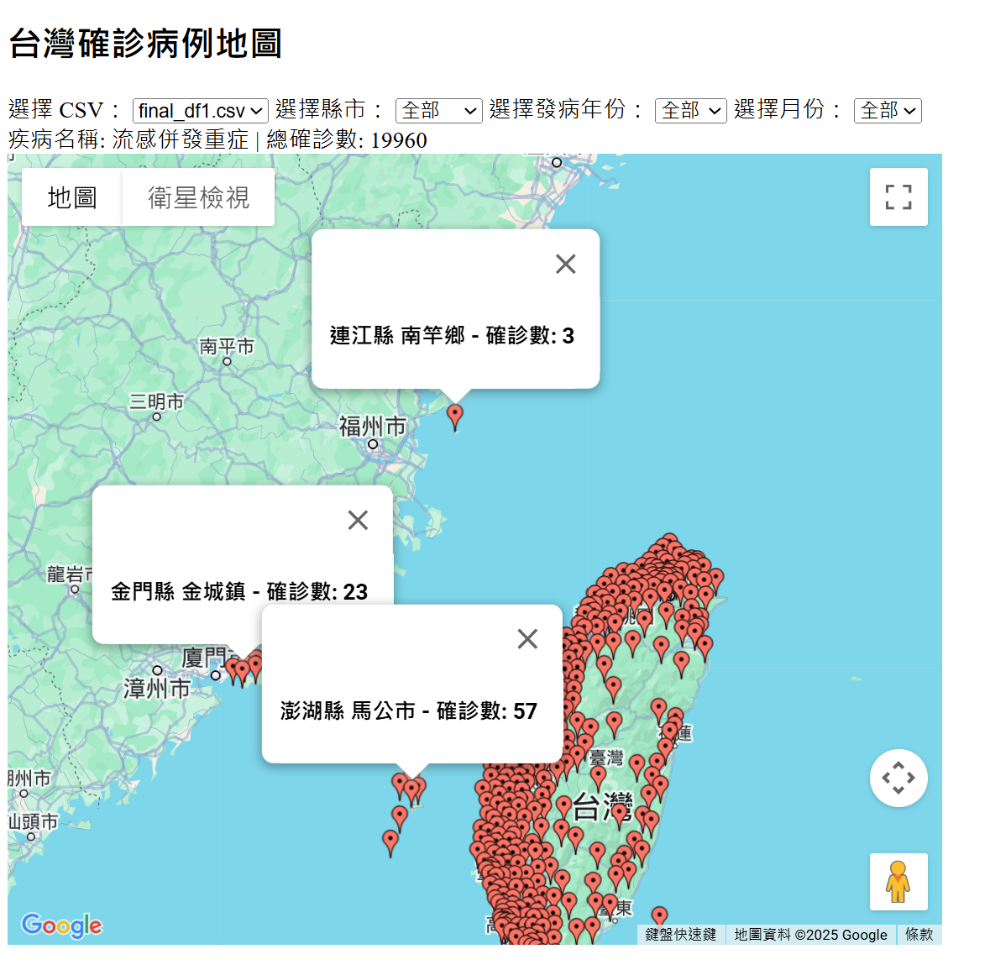
定義一個 JSON 資料來源的對應表，每個 CSV 檔案名稱對應一個 JSON URL；定義 XML 檔案的下載網址，該檔案包含行政區經緯度資料；建立一個名為DISEASE\_MAPPING的字典，用以將疾病代碼與疾病名稱進行對照。

由於需要將臺灣各個行政區的病例數以標記的方式在Google Map中顯示，所以要將各行政區經緯度資料的中心經緯度資料添加到疾病資料中。為了方便資料的處理會將XML 檔案與JSON檔利用Pandas轉換成Data Frame格式；在各行政區經緯度資料中需要對行政區的欄位的值進行拆分，將其拆成與疾病資料一樣的縣市加鄉鎮的格式；在疾病資料的縣市的欄位值可能出現高雄縣(舊)、台中市(舊)和台南市(舊)等，需要轉換成升格後的直轄市名稱，才不會在將各行政區經緯度資料的經緯度資料插入到疾病資料時產生無法匹配的情況。

需要處裡的資料

## 網頁地圖功能:

為了在網頁上顯示地圖，要到Google Cloud平台申請一組 Google Map API 金鑰。取得金鑰後，在網頁的 HTML 中引用 Google Maps JavaScript API。

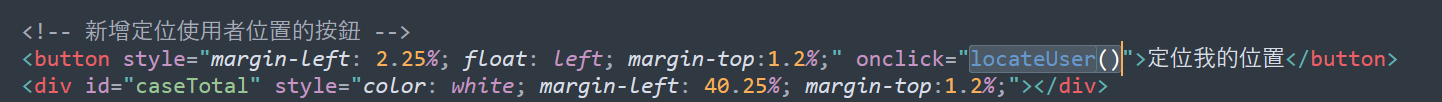
將先前處理過後的疾病資料送回前端，讓使用者根據所選條件下顯示的各個行政區的確診數，在網頁中的地圖顯示。

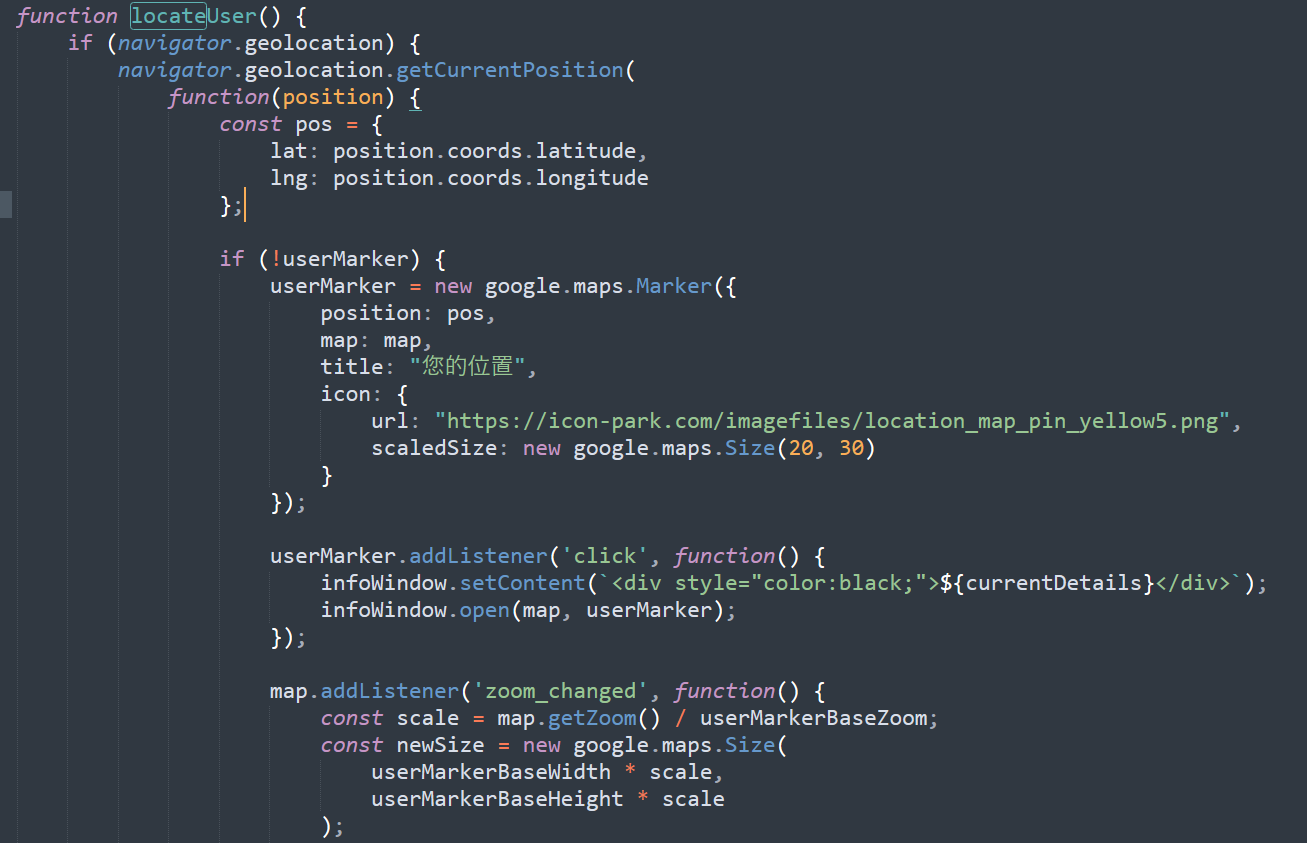
台灣確診病例地圖

由於原始的臺灣確診病例地圖中，總確診數為0 時在下拉式選單中依然有選項可以選取，且此總情況還會導致疾病名稱的欄位顯示未知疾病，因此邀將選擇縣市、選擇發病年份以及選擇月份進行分組求和，再利用總確診數 > 0 來篩選出總確診數大於 0 的選項。

在下述的圖片中的代碼中的三個函式(**get\_counties**, **get\_years**, **get\_months**)皆是從請求中獲取相關參數，並依據指定條件（**縣市**、**發病年份**、**發病月份**）讀取並篩選 CSV 檔案資料，並針對指定的條件進行分組後針對**確診病例數**求和，接著保留**總確診數大於 0** 的條件的資料轉換成列表並排序，最終將篩選後的結果（各個縣市、年份或月份的列表）以 JSON 格式回傳，前端的 map.js 可以透過 AJAX 請求來取得這些資料，用來更新下拉選單、地圖標記。

在Google Map中新增定位功能，讓使用者能知道定位附近行政區的確診數，網頁中的定位我的位置的按鈕為避免使用者在地一次載入網頁時無法順利抓取到定位的備用方案。

台灣確診病例地圖(增加定位功能)

新增定位功能

定位功能(map.js)

## Line Bot

利用Line Bot讓使用者查詢所在地點或是指定行政區的病例各數，以下兩張圖分別為Line Bot 查詢疾病流程圖和使用者在Line Bot的實際操作流程圖。

在Line Bot的操作流程中，若使用者在第三個聊天室畫面出現下列情況，在點選「使用者目前位置」但超過 10 秒未傳送位置資訊時，系統將自動改為「選取縣市」查詢，以確保流程能順利執行完畢。

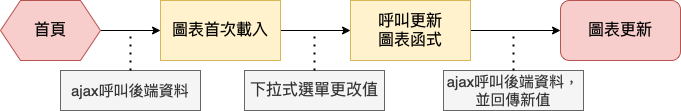
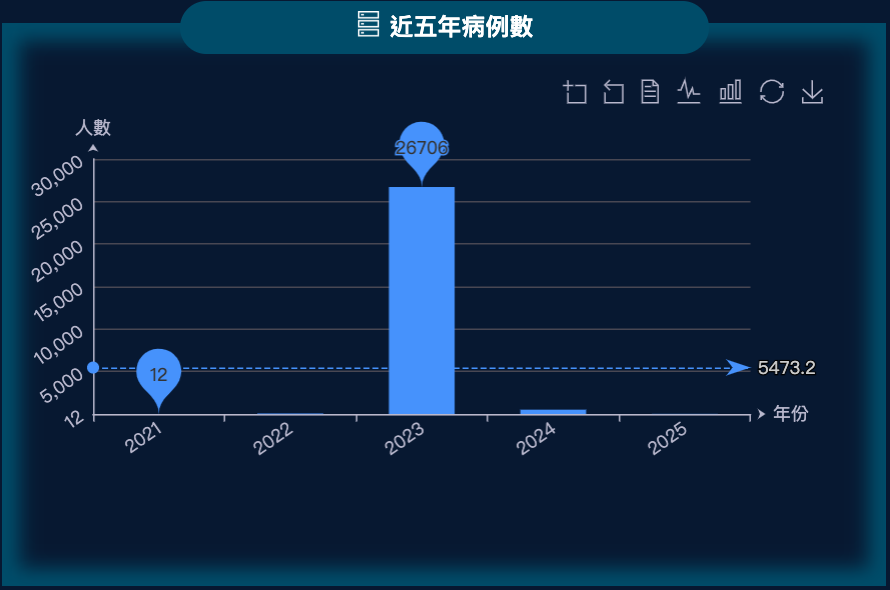


下圖中的代碼就是用於處理在使用者點選「使用者目前位置」但超過 10 秒未傳送位置資訊時的詳細的方法以及處理方式，LINE\_USER\_SESSION 全域字典用來儲存每個用戶的Session資料，get\_user\_session的功能為根據用戶ID取得Session資料，set\_user\_session的功能為將指定用戶的Session資料存入全域字典中；schedule\_location\_fallback的功能為在用戶點選「傳送位置」後開始計時等待，如果10秒內未收到有效位置資訊，則執行fallback。其作用在於啟動一個 10 秒後觸發的計時器，在到達10秒時檢查是否收到位置資訊，若等待超時會透過推播的方式通知用戶並自動切換到「選取縣市」的流程，讓用戶繼續後續操作。同時，移除用戶的等待記錄，防止重複處理的情況發生。



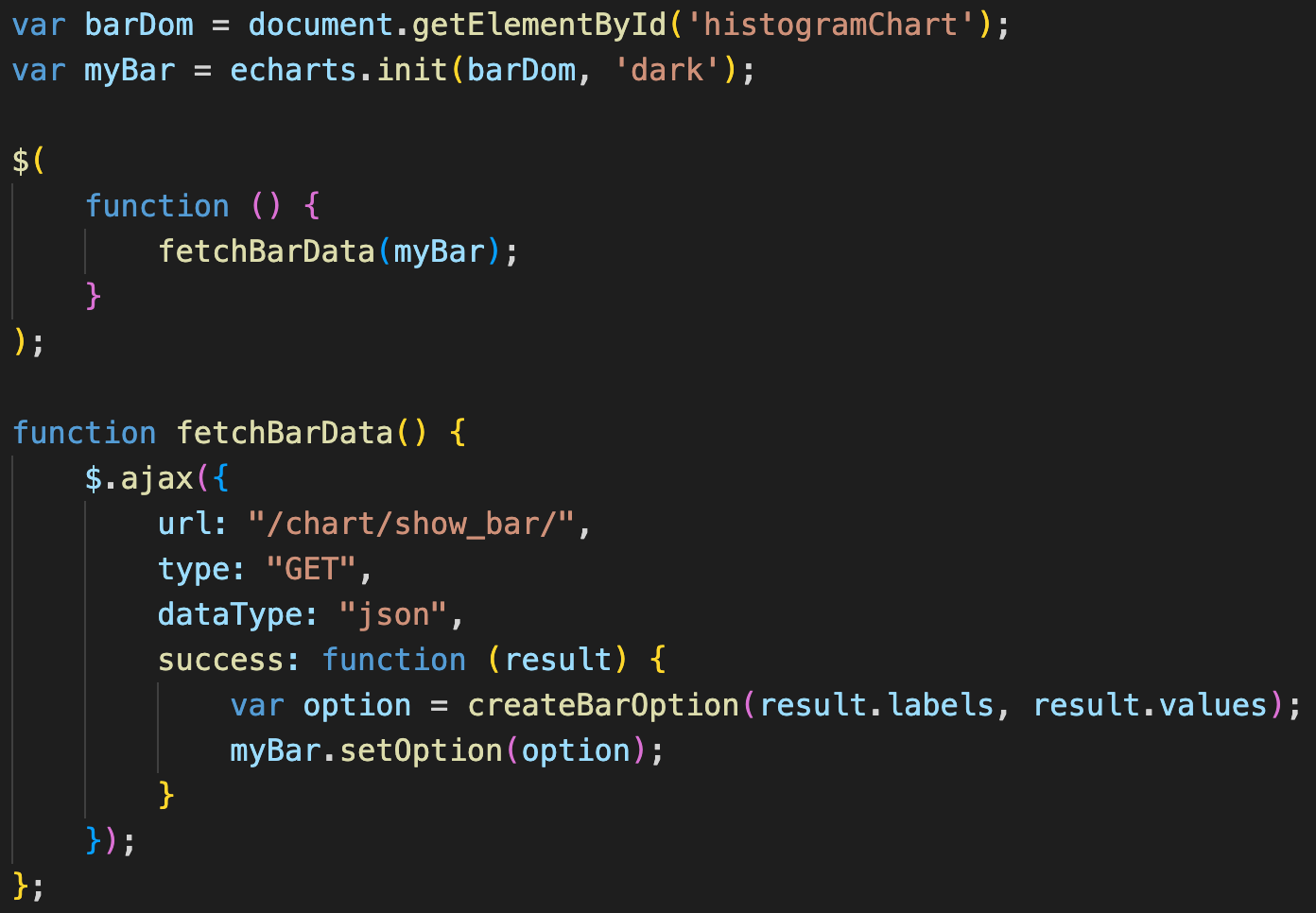
# 四、首頁圖表呈現

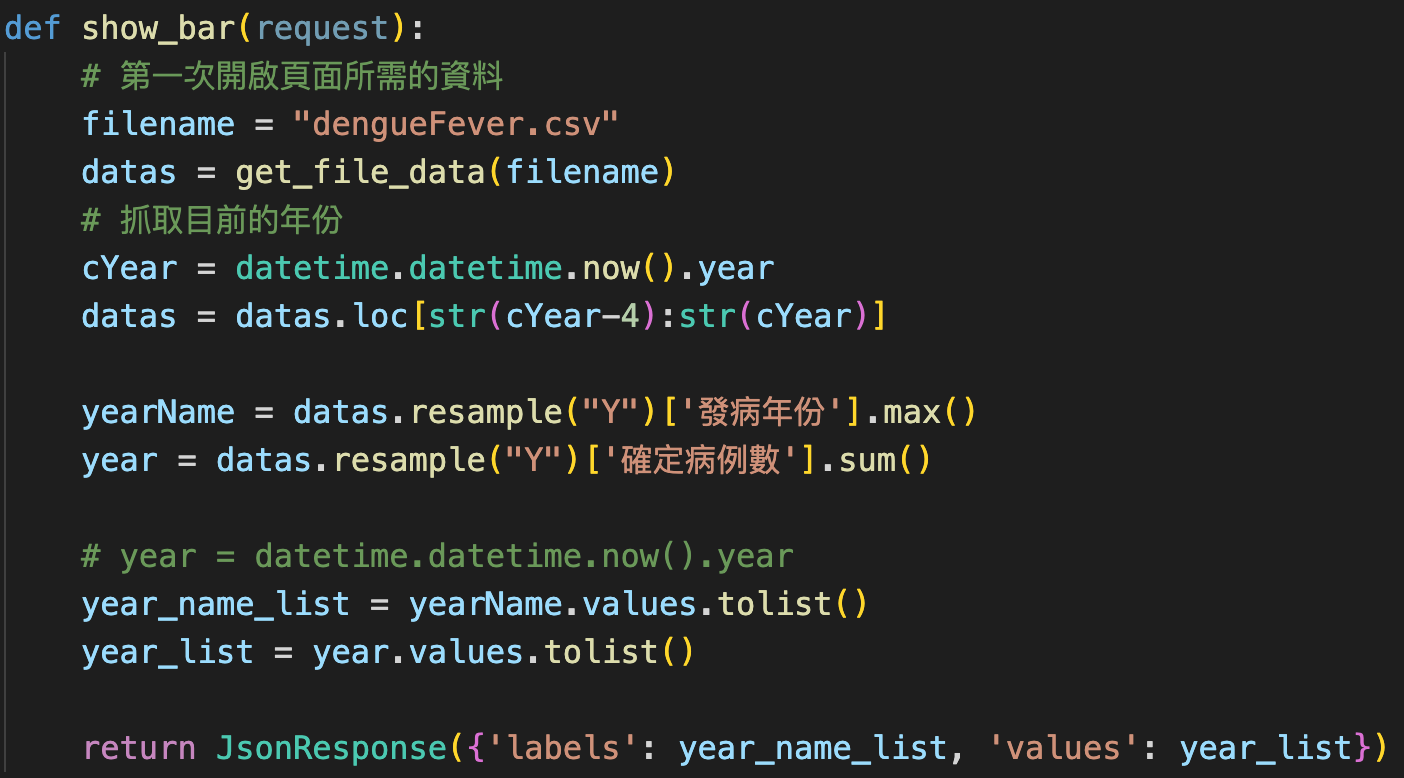
## 首頁圖表流程圖



## 近五年病例數

長條圖適合展示時間趨勢，因此被選用來呈現近五年內各時間段的數據變化，並突顯變化最顯著的時期。

當首頁載入時，JavaScript 會初始化 **ECharts** 長條圖，並透過 **AJAX** 向後端請求數據，以動態更新圖表內容，確保即時顯示最新的趨勢分析。

前端透過 AJAX 請求 /chart/show\_bar/，調用後端函數 show\_bar 以獲取數據。後端預設讀取登革熱病例數據檔案（dengueFever.csv），並使用 get\_file\_data 函數解析 CSV，處理日期資訊。系統擷取當前年份的數據，篩選出適用於長條圖的資料，轉換為 **ECharts** 所需的 列表格式，最終回傳對應的 標籤（**labels**） 與 數值（**values**） 供前端繪製圖表。



## 近三年月病例數

折線圖適用於顯示趨勢變化，因此被選用來呈現近三年內每月的病例數變化，以視覺化方式展示疫情趨勢。

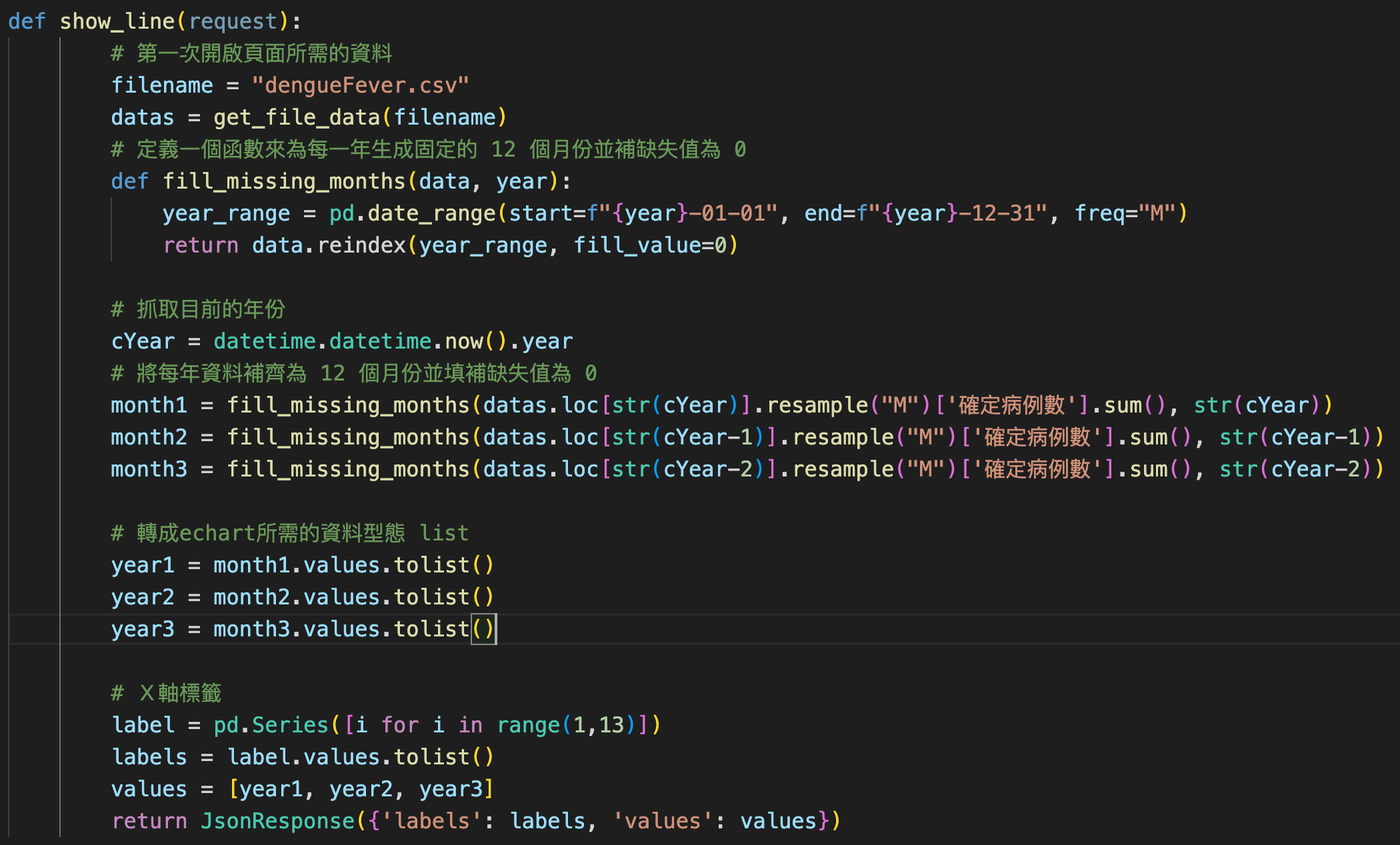
當首頁載入時，JavaScript 會初始化 **ECharts** 折線圖，並透過 **AJAX** 請求後端數據，以動態更新圖表，確保即時呈現每月病例數的變化情況。

前端透過 AJAX 請求 /chart/show\_line/，調用後端函數 show\_line，負責提供近三年每月的登革熱病例數據，並將數據轉換為JSON 格式，供前端 ECharts 折線圖使用。

函數首先讀取 登革熱病例數據檔案，並透過 get\_file\_data 進行數據處理。為確保每年的數據包含完整 **12** 個月份，函數定義 fill\_missing\_months，將缺失的月份填充為 0，避免因數據缺失導致圖表顯示異常。

接著，程式擷取當前年份及前兩年的病例數據，並使用 .resample("M") 來按月份彙總確定病例數，再透過 fill\_missing\_months 確保每年資料完整。經過處理後，這些數據被轉換為列表格式，並存入數值（**values**），其中包含近三年的每月病例數。

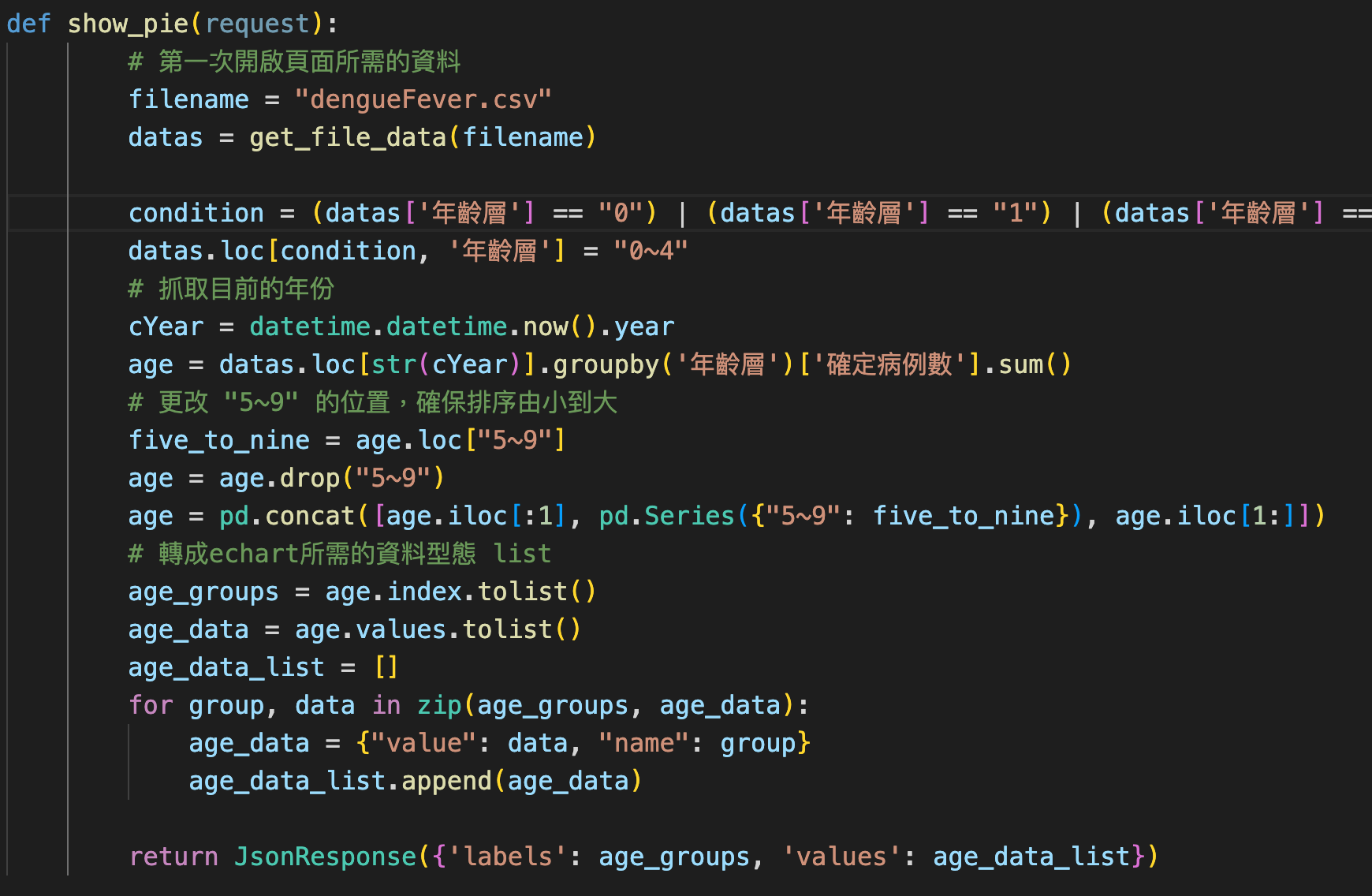
此外，為了確保 X 軸標籤的一致性，程式將 labels 設為 [1, 2, ..., 12]，代表 **1** 月至 **12** 月，確保不同年份的數據能夠對齊比較。最後，函數將標籤（**labels**） 與 數值（**values**）以 JSON 格式回傳，讓前端可以使用這些數據繪製折線圖，清楚呈現 近三年內每月的病例變化趨勢，幫助分析疫情發展。



## 去年病例年齡層

年齡層分佈以圓餅圖方式呈現。前端使用 JavaScript 透過 AJAX 從後端請求數據，並將取得的數據套用至 ECharts 圖表，實現了即時數據獲取與視覺化，使使用者能直觀地了解不同年齡層的疫情狀況。

使用 圓餅圖（Pie Chart）的主要原因是它適合用來 表示比例分佈，尤其當我們關心 各個年齡層的確診病例數佔比 時，圓餅圖能夠清楚地呈現 不同年齡層的相對影響，方便視覺化比較各群體之間的比例大小。

後端讀取預設登革熱病例數據檔案，整理並統計去年年齡層的確診病例數，特別處理 **0~4** 歲 和 **5~9** 歲的年齡區間，確保資料排序正確，最後將結果轉為 JSON 格式回傳給前端。

## 今年縣市病例數

透過圓餅圖呈現各縣市的確診病例分佈，讓使用者能夠清楚看到哪些縣市的疫情較為嚴重，進而協助相關單位制定有效的防疫策略。

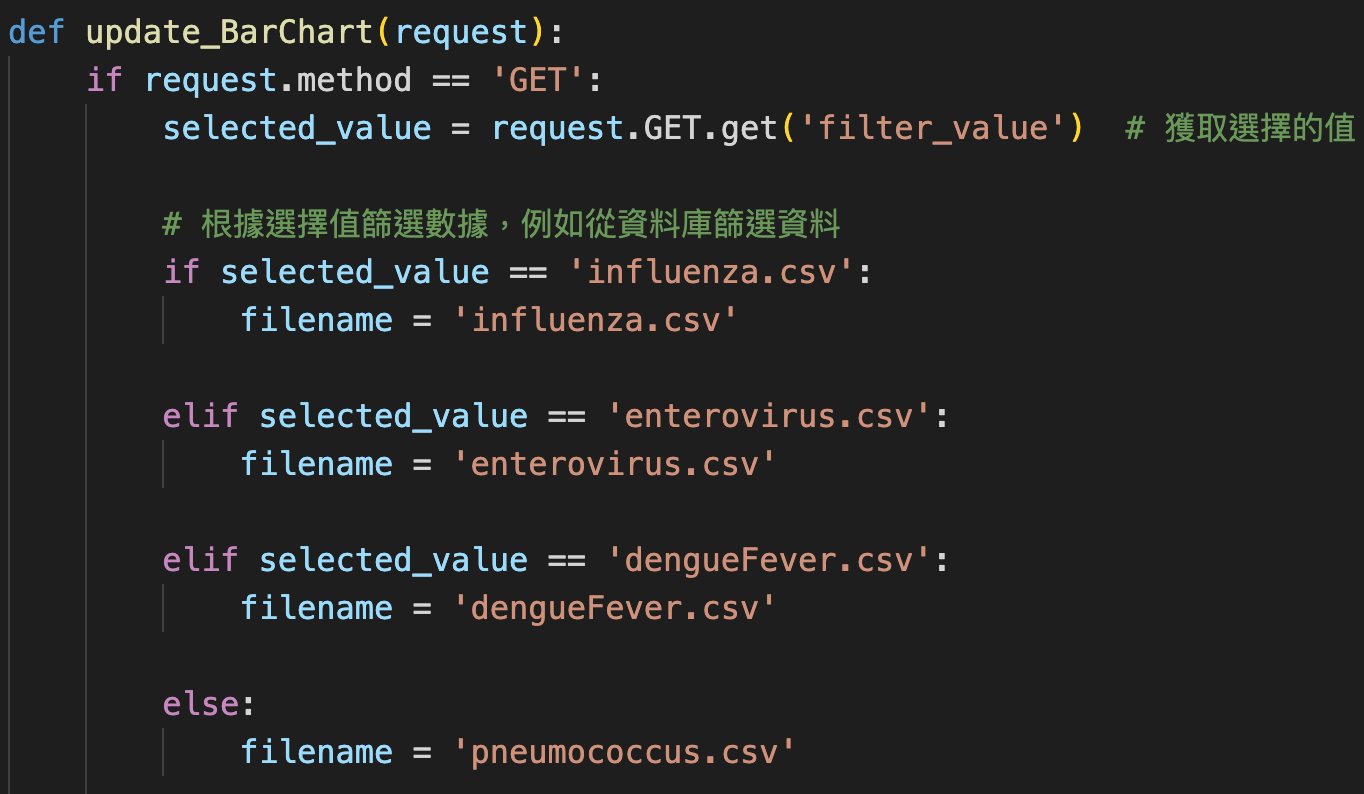
使用 圓餅圖 的主要原因是 它適合表達比例關係，特別是當我們比較 各縣市確診病例的占比時，圓餅圖可以直觀地呈現 不同縣市的疫情嚴重程度。

載入首頁後，透過 **AJAX** 向後端請求 縣市別確診數據，回傳的資料包含 各縣市名稱與對應的確診數。後端讀取預設登革熱病例數據的檔案資料，篩選當前年份的數據，並按照縣市名稱進行分組統計確診數量，最後將整理後的數據轉換為 **ECharts** 需要的 **JSON** 格式 回傳給前端，讓使用者可以透過圓餅圖直觀了解各縣市的疫情分佈情況。

## 更新圖表數據

在 HTML 的 <select> 標籤上使用 onchange 事件來呼叫對應的函式，當使用者選擇不同的選項時，四個圖表會同步更新。

根據使用者選擇的檔案篩選數據，並更新對應圖表，使其顯示對應疾病的確診病例趨勢或比例分佈。



當使用者選擇不同的檔案（如 influenza.csv, dengueFever.csv）時，觸發 updateBarChart()與其他對應函式，此時函示會取得選擇的值，並透過 GET 請求對應的網址（如 /chart/update\_BarChart/），將 filter\_value（選擇的檔案名稱）傳送到 Django 後端。後端相對應的函數取得選擇的檔案名稱CSV 檔案，並篩選數據，轉換為 ECharts 所需的 JSON 格式後，再回傳到前端。前端接收數據並更新相對應的圖表。讓使用者能夠根據不同疾病類型篩選數據，並即時更新圖表，藉此觀察不同疾病在近幾年的趨勢變化。這種互動方式讓數據更加靈活，使用者可以輕鬆切換查看不同疾病的病例變化情況。

# 五、會員系統

## 工具介紹

**1.Firebase**

Firebase 是 Google 提供的雲端開發平台Web 開發、後端服務。它的主要功能為後端服務 (如資料庫、認證)。

Firebase Admin SDK(Software Development Kit)是其官方推出的軟體開發工具套件。

這次主要使用的功能為Realtime Database, Firebase Authentication

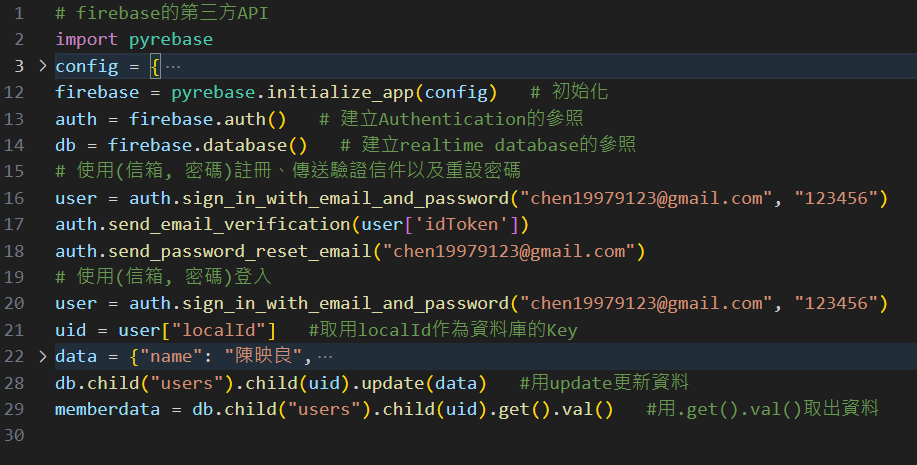
Realtime Database 為Json格式，適合資料量少或是文字類型儲存的應用；另外同為Firebase資料庫的Firestore則適合資料量大，並包含文件 (Document) + 集合 (Collection)的應用

Firebase Authentication 是Firebase提供的驗證方式，支援 Google、Facebook、Apple、Email/密碼等登入方式。

Authentication也提供了會員系統 (登入、註冊、驗證信箱、密碼重設)的應用。

有個第三方的API，pyrebase發音跟firebase相像，令人忍俊不禁。

**2. Pyrebase**  
為Firebase的第三方 API，適合配合 Django 或 Flask 專案，含有 Firebase Authentication, Realtime Database, Firebase Storage，沒有 Firestore 的功能，不適合開發資料庫使用量過大的應用。

Pyrebase功能簡易、易懂且齊全，適合開發周期短或是做為測試使用的應用。

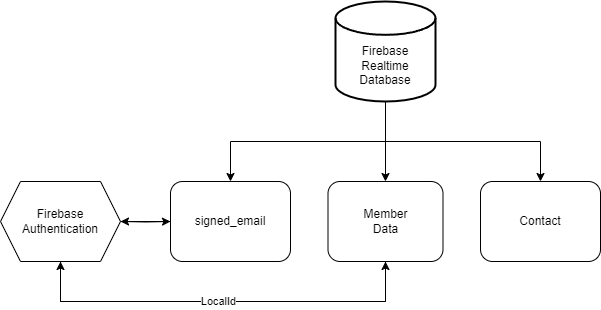
firebase = pyrebase.initialize\_app(config) 給予所需配置，初始化firebase auth =

firebase.auth()

Authentication的參照

db = firebase.database() Realtime Database的參照

## 會員資料架構圖



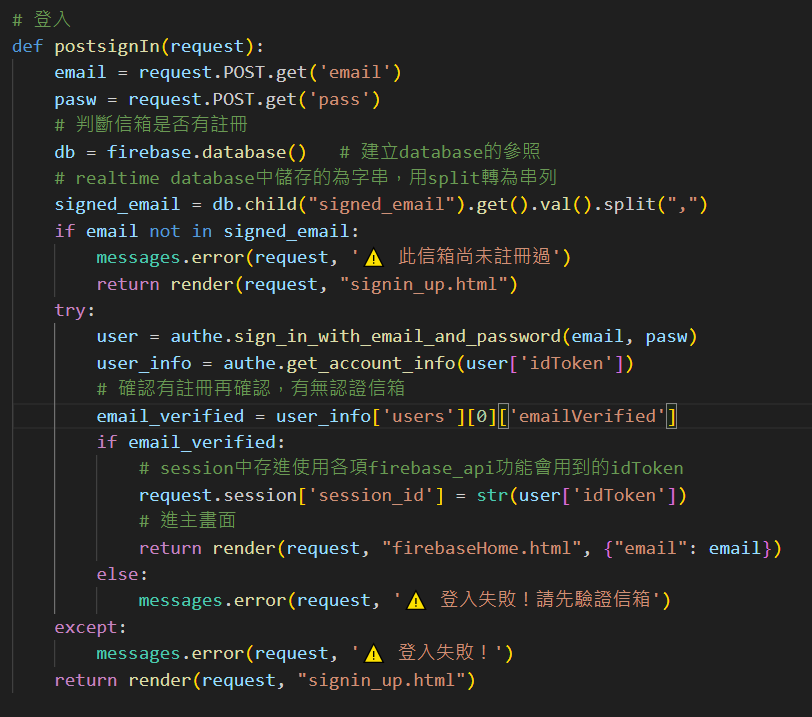
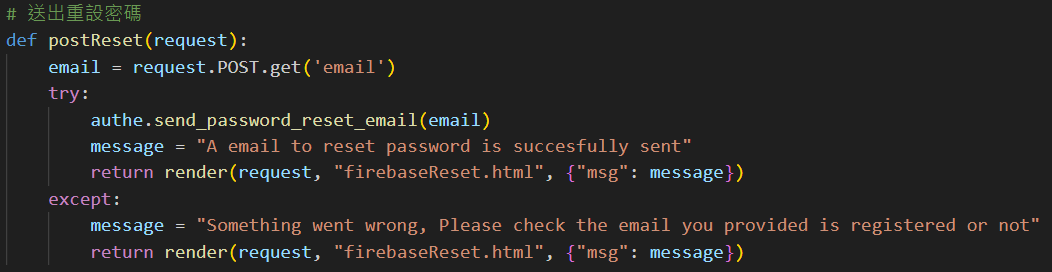
## 功能介紹

**1.註冊&登入**

仰賴於css，進入此頁面可以做到同一個頁面，同時有註冊與登入功能，並且互不影響使用。在註冊的時後會送出，預設的會員資料(name, email, gender=male, birthday= “”, phone=“”)

使用 user = auth.sign\_in\_with\_email\_and\_password(email, password) 後使用

auth.send\_email\_verification(user['idToken'])發送信箱驗證信件，並且將註冊成功的email紀錄於database 的 signed email 中，可以確保檢測重複註冊，或是登入時有無此人帳號。

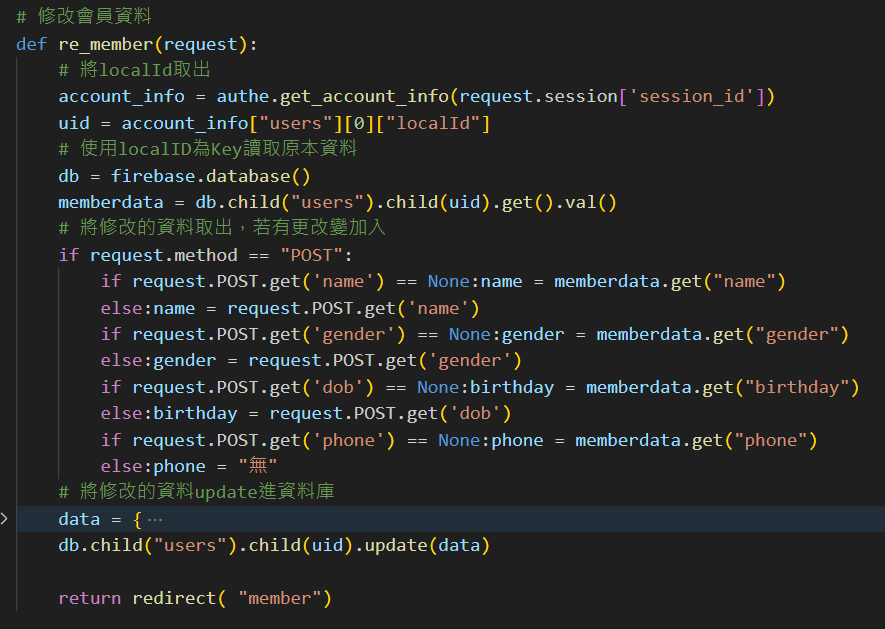
登入使用 user = auth.sign\_in\_with\_email\_and\_password(email, password)  
將 (user['idToken']) 轉成 session(session\_id)作為登入後的通行證，登出也只需要將此session刪除即可。

使用auth.send\_password\_reset\_email(email)重設密碼。



**2.會員資料的檢視與編輯**

使用sript切換編輯、儲存以及取消按鍵的顯示，同時在切換成編輯時，原先無法改動的內容便可以改動了。

使用系統中的 session id 取的會員的localId，並以其叫出該會員的資料。  
會員資料的修改需要先判斷前端是否有更改input的內容，有則將其加入要更新的資料中，若無則使用原資料。

**3.聯絡我們**

傳送簡易的資料到 Realtime Database 以供檢閱，並且會紀入時間戳來確保每則訊息都能分開儲存。



# 六、風險預測系統

## 介紹

本系統基於政府開放數據，設計並實現一套疾病風險預測系統，目標是通過整合流感、腸病毒、登革熱及侵襲性肺炎鏈球菌等四類常見疾病問題的歷史數據，結合人口統計資料，構建機器學習模型以預測區域性感染風險。系統採用分層架構（View層、Model層、Templates層），結合KMeans分群分析與線性回歸模型，實現高效數據處理與透明化的風險評估。最終輸出包含風險機率（每萬人確診率）及分風險評價（輕度、中度、高度風險），為公共衛生政策提供數據支持。

## 

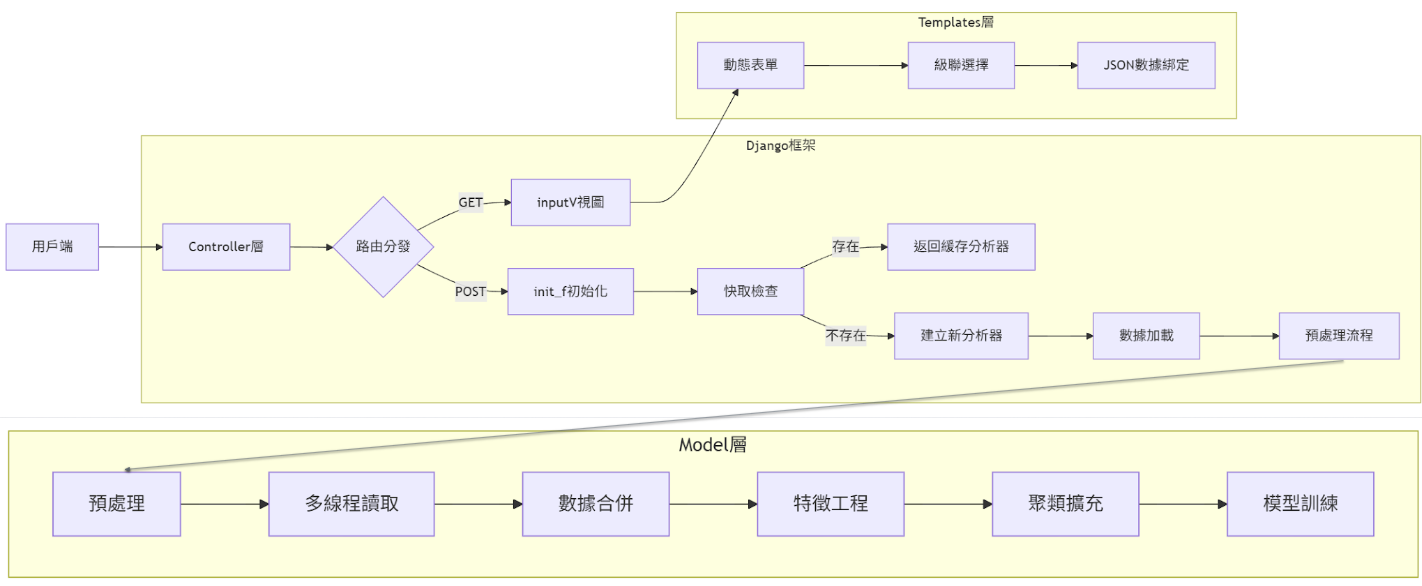
系統特點：

* + 快速預測：基於用戶輸入的縣市、鄉鎮等參數，即時計算風險值。
  + 透明化分析：模型特徵權重清晰可解釋，支持政策制定。
  + 效能優化：利用多線程處理與快取機制，提升巨量數據運算效率。

## 系統架構與流程

**架構圖**

系統分為三層：

* + View層：接收用戶請求（GET/POST），初始化數據集與表單，管理快取。
  + Model層：執行數據讀取、預處理、分群分析、模型訓練與預測。
  + Templates層：動態渲染表單與結果頁面，支持縣市-鄉鎮聯動選單。

**流程圖**

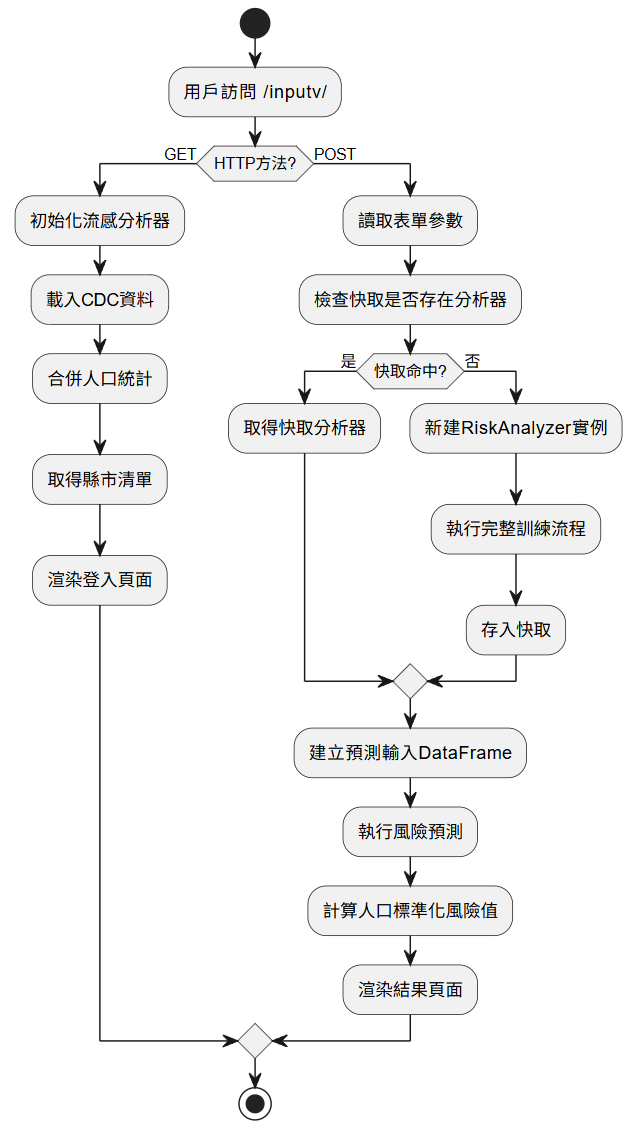
GET請求：

初始化載入預設疾病（流感）數據，提取縣市與鄉鎮列表，生成動態表單。

快取初始化：以疾病類型為鍵，保存分析器（1小時失效）。

POST請求：

必填欄位：疾病類型、縣市；自動填充欄位：年、月、性別等（取最常出現值）。

若選擇流感會直接預測，否則重新加載數據並訓練模型。

## 風險預測UI

**動態表單設計**

* 聯動選單：縣市選單變更時，自動加載對應鄉鎮選項。
* 必填驗證：疾病類型與縣市為必填欄位，未填寫時禁止提交。
* 自動填充：性別、年齡等欄位與會員系統綁定，減少用戶輸入負擔。
* 鄉鎮、是否境外移入為選填。



## 資料來源

疾病數據：來自衛生福利部疾病管制署開放平臺，包含年齡、縣市、性別等維度。

流感：

<https://od.cdc.gov.tw/eic/Age_County_Gender_487a.csv>

腸病毒：

<https://od.cdc.gov.tw/eic/Age_County_Gender_0749.csv>

登革熱：

<https://od.cdc.gov.tw/eic/Age_County_Gender_061.csv>

侵襲性肺炎鏈球菌：

<https://od.cdc.gov.tw/eic/Age_County_Gender_0412.csv>

人口數據：<https://data.moi.gov.tw/MoiOD/System/DownloadFile.aspx?DATA=9DDF0127-95EE-484E-A439-F98E2C30CA0F>

以上數據以上四種均是較為常見的疾病，侵襲性肺炎鏈球菌會引發包括敗血症、肺炎、腦膜炎、關節炎等，其中肺炎是常見且嚴重的疾病

資料均來自：政府開放平台: https://data.gov.tw/

## 數據處理

**效能提升：**以提升程式效能為目的。pandas讀取疾病數據，以.to\_parquet將數據壓縮將CSV轉為Parquet格式，再分割為4塊用ThreadPoolExecutor並行處理後合併。

讀取人口數據使用向量化分析利用Pandas Dataframe的行、列，一次性對數據進行處理，.str.extract批量對數據進行匹配和分割，.str.replace對數據進行字元轉換。這邊會將人口數據跟疾病數據合併。

**特徵工程：**將數據劃分為特徵和目標變數。建立 make\_pipeline 把 TargetEncoder（目標編碼）、StandardScaler（標準化）進行串聯，可以一次性調用 fit 和 transform 方法，依序對資料進行多階段處理。

TargetEncoder：將類別型資料轉換成數值形式。依據每個類別與目標變數（確診病例數）之間的關係來計算一個統計數值（例如均值），從而進行數值映射。

StandardScaler：會對資料進行標準化處理，消除特徵單位差異。

## Kmeans分群分析

* 算法選擇：採用MiniBatchKMeans，隨機取小批量的數據進行計算，從而大幅提高效能，同時保持較高的聚類準確性。
* 最佳K值：採用 WSS（Within-Cluster Sum of Squares，組內平方和）比較 2~10 個中心點之間的差異，找到最佳的數量K值，即Ｋ個中心點。同樣用ThreadPoolExecutor進行多線程處理。
* 特徵擴充：

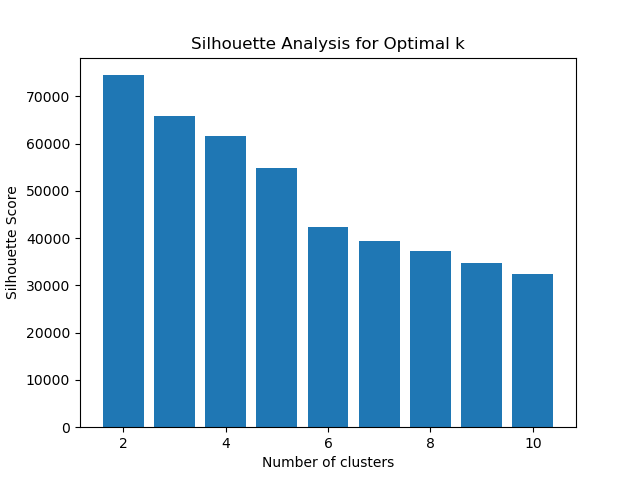
有了K值就有最佳K個資料群，可以得到分群標籤、分群距離作為新的特徵合併入原始資料提升模型性能，增強對空間分佈模式的捕捉能力 。

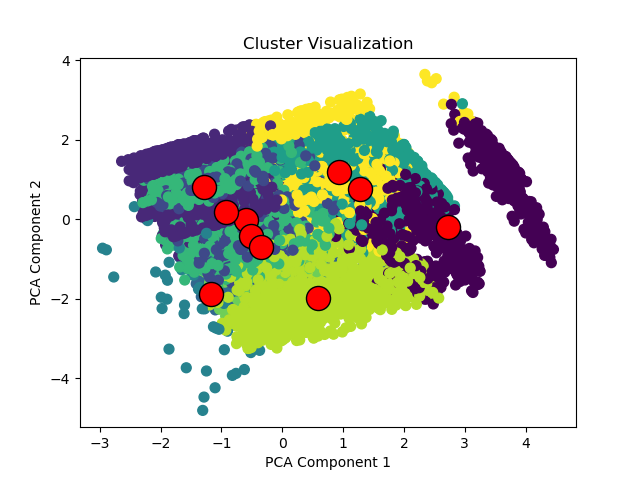
1.分群標籤（cluster）：將樣本分群。反映區域潛在風險結構。

2.分群距離（dist\_centroid\_i）：樣本到各群中心的距離。提供了樣本與各分群中心的相似性資訊；同時分群距離作為相對位置，能削弱絕對位置座標的影響，捕捉空間分佈模式。

* WSS：一種評估 K-means 分群效果的方法，為每個點與其所在群中心的平方距離總和，用於衡量聚類內的緊密程度。組內差異小，代表每個群內是越聚合、越分明。找尋WSS曲線拐點又稱肘部點即最佳K。
* 數據清洗：離群點偵測，基於KMeans的分類距離來計算離群閾值，參照離群閾值分出離群點，資料清洗離群點。

**流感數據的最佳K個點：**WSS後發現最佳K為10。

**流感分為10群：**



## 模型預測

**線性迴歸：**

* 優勢：

1.特徵權重透明（如人口數、分群距離的影響），容易解釋各個特徵對風險的影響，符合政策透明化需求。

2.標準化和分群特徵生成，使得特徵之間的關係更接近線性，從而適合線性回歸。

3.線性回歸訓練速度快，適合巨量資料集。

4.分群特徵已編碼空間關係，降低非線性需求。

* 模型解讀：

cluster：代表不同分群區域的基礎風險水平。影響資源分區調配依據。

dist\_centroid\_0：代表與高風險群中心的距離影響程度。影響疾病熱點監控。

人口數：每萬人感染率的邊際變化。影響疫苗分配優先級。

**風險評估：**

可預測單一鄉鎮或整個縣市的疾病發生率，支援多鄉鎮批量預測。

模型的輸出值為單一地區的確診病例數。

* 風險機率：每萬人確診的機率，。
* 百分位數評估：計算地區平均確診數及其百分位數。
  + - 輕度風險：<25%百分位數。
    - 中度風險：25%~75%百分位數。
    - 高度風險：≥75%百分位數

## 模型評估

流感數據：19162筆資料，時間橫跨2007~2025。

RMSE為0.33309，數據取完對數為0.30662。

腸病毒：1125筆資料，時間橫跨2007~2025。

RMSE為0.23431，數據取完對數為0.23514。

登革熱：39362筆資料，時間橫跨2007~2025。

RMSE為18.26947，數據取完對數為1.67235。

侵襲性肺炎鏈球菌：8923筆資料，時間橫跨2007~2025。

RMSE為0.11058，數據取完對數為0.31675。

故數據為前三類，會對目標變量取對數。

**僅對目標變量取對數：**

在流行病學模型中，病例數（如登革熱病例）通常呈現右偏分佈（大量低值，少數高爆發值）。

所以壓縮高值區間的尺度，減少極端值對模型（如線性回歸）的影響，使分佈更接近常態分佈，符合模型假設。

**不對特徵變量取對數：**

「縣市、鄉鎮、性別、年齡層」編碼後的數值已標準化，且不具備原始病例數的偏態問題。「是否為境外移入」特徵則無右偏問題。

# 七、未來展望

本平台通過整合開放數據、視覺化、各項API套用與機器學習技術，實現了高效、透明的疾病風險預測網站。未來可擴展方向包括：

* 讓 LineBot 能夠即時回報與呈現風險預測的數據
* 改成使用 Firebase Admin SDK，串接Django後臺管理所需的資料庫
* 增加疾病類型：納入更多傳染病數據（如COVID-19）
* 增加不同預測模型或長期趨勢預測,讓系統更加靈活與準確
* 模型優化：嘗試集成學習或深度學習模型以捕捉複雜特徵關係
* 疾病預防資訊查詢,豐富平台功能
* 實時數據更新：串接API實現數據自動更新，提升預測即時性
* 響應式網頁設計