

資料科學程式設計 第五組-YouBike 資料分析應用發想

1. 應用概述

本應用專為 YouBike 使用者設計，提供了即時的站點狀態查詢、導航、使用者位置顯示及車輛預測等功能，旨在改善使用者在站點間借還車輛的便捷性，解決地圖顯示車輛數量充足但實際到站卻無車可借的問題。

1.1 應用流程概述

- **進入應用**：使用者進入應用介面後，需首先點擊「連線」按鈕，應用會與 YouBike 即時資料 API 連線，並取得車輛數量、站點位置等相關資訊。
- **選擇行政區與站點**：使用者選擇台北市的行政區，應用會自動篩選並顯示該行政區的所有 YouBike 站點，使用者可在清單中選擇具體站點查看詳情。



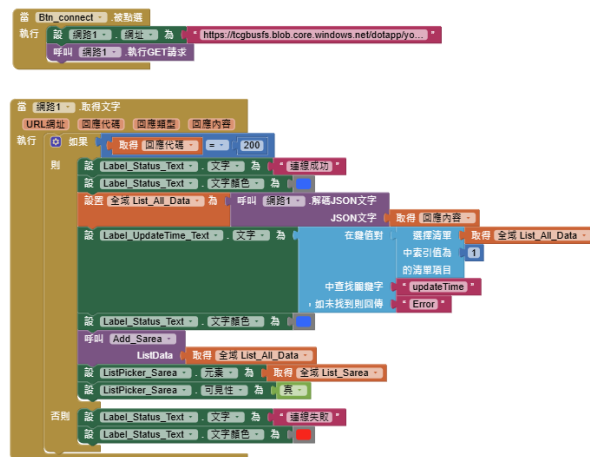
- **查看站點狀態**：在選擇具體站點後，應用會顯示該站點的詳細資訊，包括目前可借車輛數、可還車位數等，方便使用者決定是否前往該站點借車。
 - **地圖顯示與導航**：應用將該站點標示在地圖上，使用者可以點擊站點標記來進行導航，獲得從當前位置前往站點的最佳路線。
 - **查看使用者位置**：使用者可以隨時在地圖上查看當前位置，以了解距離站點的實際距離。
 - **預測功能**：應用包含車輛預測功能，根據站點過去的借還車紀錄，預測未來幾分鐘的車輛供應情況，幫助使用者避免因無車可借而浪費時間。
- 查看站點位置、可借車數量



2. App Inventor 邏輯 (程式區塊實在太多, 只截代表性的區塊做展示)

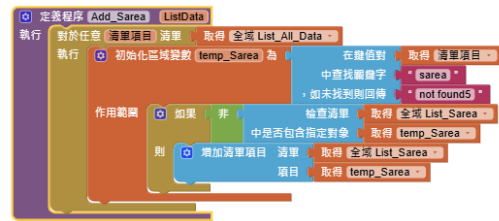
1. 連接資料源:

- 使用者點擊「連線」按鈕後, 應用會透過 API 取得最新的 YouBike 資料, 更新站點清單及地圖資訊。

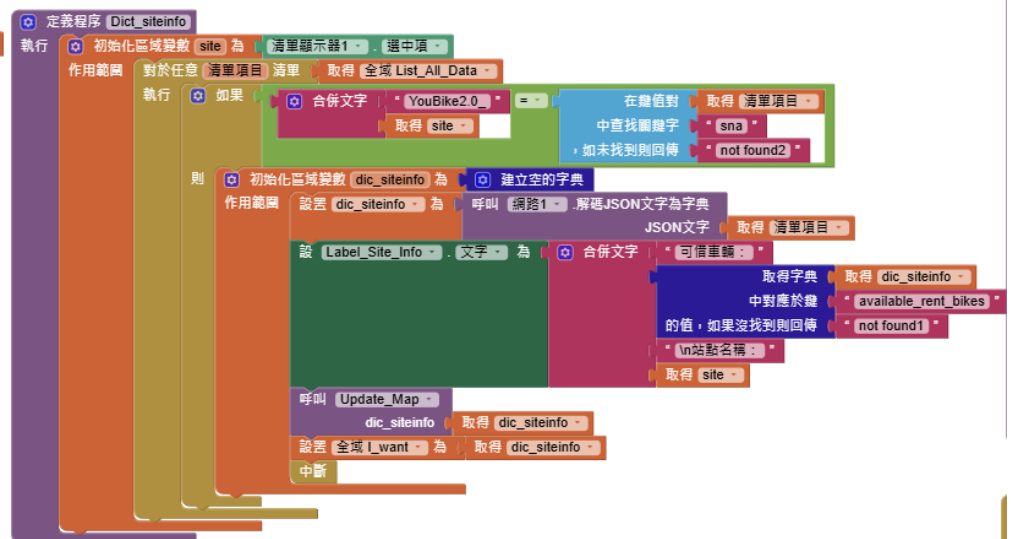


2. 選擇行政區和站點:

- 使用者選擇台北市的行政區後, 應用自動顯示該區域內所有的 YouBike 站點清單。

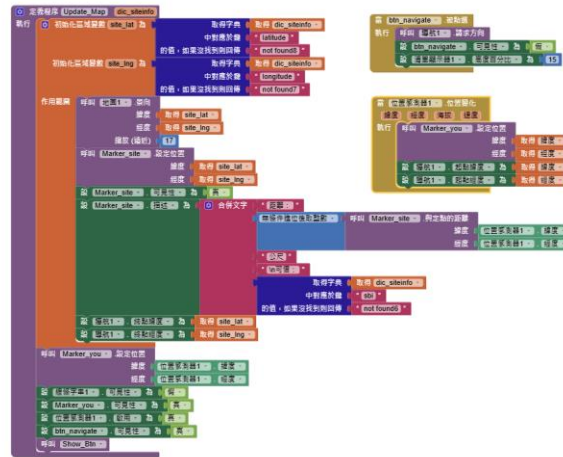


- 使用者從清單中選擇一個具體站點，應用會顯示站點的即時狀態。



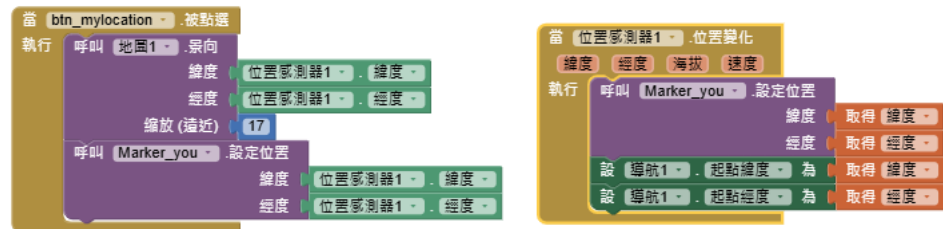
3. 查看即時資訊與導航：

- 當使用者選定站點後，應用會顯示該站點的可借車輛數。
- 若使用者決定前往該站點，可點擊地圖上的標記以啟動導航功能，系統會自動規劃出路徑，指引使用者前往。



4. 查看使用者位置：

- 使用者可隨時啟用位置查詢功能，確認當前位置，方便對比自己與站點之間的距離。



5. 預測車輛供應：

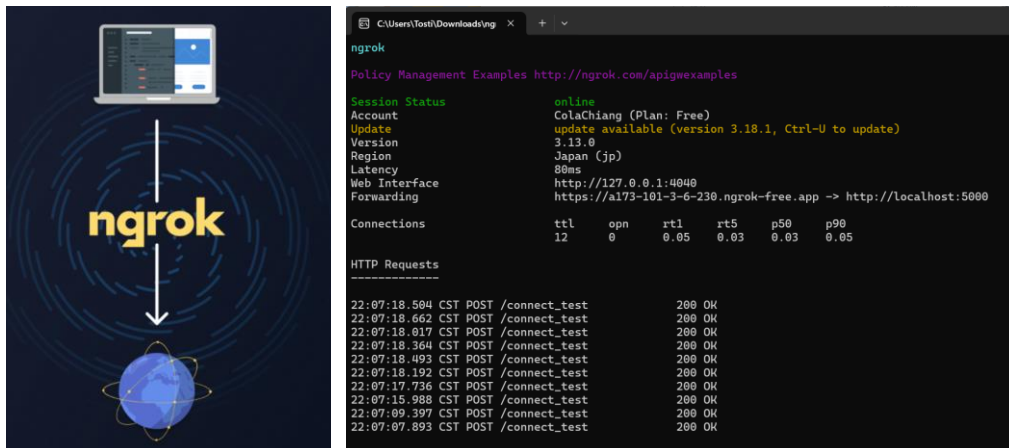
- 使用者點擊「預測」按鈕後，應用會將站點 ID 傳遞到後端的預測 API，後端使用 ARIMA 模型計算該站點未來數分鐘的車輛供應情況，並將預測結果回傳顯示在應用上。
- 若預測結果顯示未來幾分鐘內車輛會被借光，使用者可以根據預測考慮前往其他站點。

3. 技術架構設計與運行流程

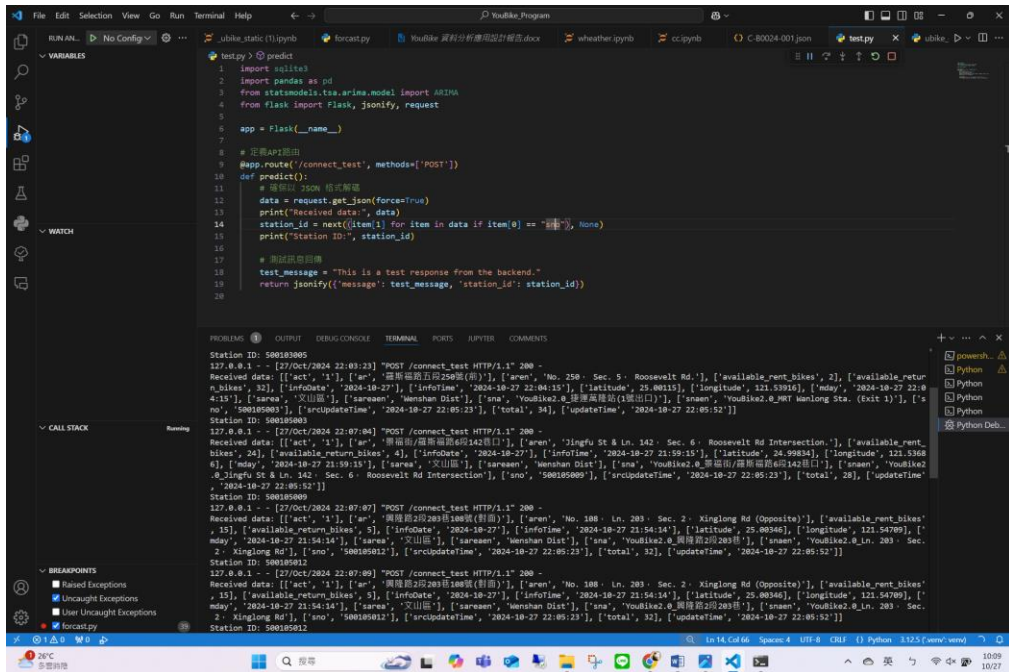
3.1 後端架構設計 (Flask 與 ngrok)

1. **Flask 伺服器：**應用的後端使用 Flask 搭建，負責處理即時資料的查詢和預測模型的計算。
 - **車輛供應預測 API：**接收站點 ID，調用 ARIMA 模型預測該站點未來幾分鐘的車輛供應狀態，並將結果返回。
2. **ngrok：**Flask 伺服器在本地運行，需要透過 ngrok 將本地服務公開至網路上，以便 App Inventor 進行調用。

App Inventor 前端與後端 Flask 連動，已經測試成功！



可從 App Inventor 獲取站點資料，已經測試成功！



可以將後端的文字回傳至前端顯示，已經測試成功！



3.2 前端介面設計 (App Inventor)

1. GET request (連線按鈕): 使用者點擊「連線」按鈕後, App Inventor 會調用 YouBike 的即時資料介接網址, 取得即時站點資訊。
2. POST request (預測按鈕): 當使用者點擊「車數預測」按鈕後, App Inventor 將站點 ID 發送到 /predict API, 獲取未來幾分鐘的車輛供應預測結果。

常見問答區

為何現在只做測試、不做完整功能呢?

因為現在 YouBike 資料量不夠, 我寫了一個程式動態在電腦中抓取 YouBike 的即時資料到 Database 中 (如最下面的圖), 不過資料量要做預測是遠遠不夠, 已經有執行過幾次模型預測皆回傳錯誤, ChatGPT 分析錯誤是資料量不足。在這過程中我們也去對我們認為會影響 YouBike 租借量的變因, 也就是天氣, 做一些資料清洗及資料視覺化。由於擔心說明檔案過於混亂, 因此分成另一個檔案做講解, 可以看另外一個檔案 (天氣資料概述) 參考

為何選擇 App Inventor 作為前端呢?

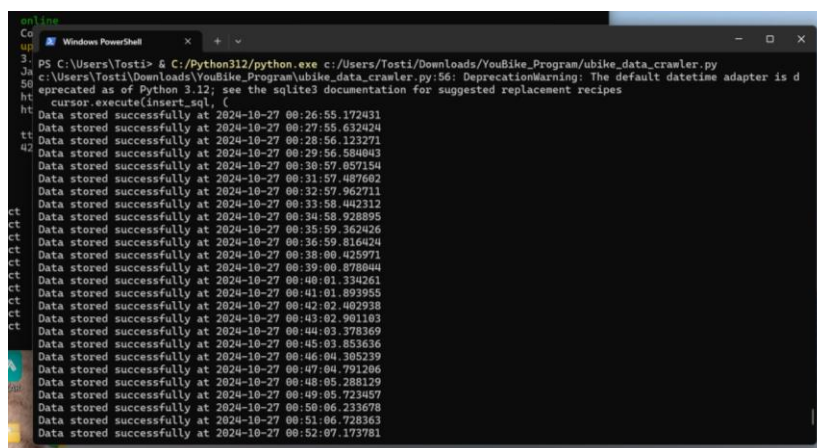
因為老師有提到這個專案要設計使用者操作介面、資料呈現頁面。感覺用 App Inventor 的介面最方便展示我們想要的成果, 但沒想到根本沒人做過 App Inventor 與後端整合, 要靠自己不斷測試、不斷想辦法修正, 光是完成整合就花了我很多時間 (@_@)

為何預測中想使用 ARIMA 模型呢?

使用 ARIMA 模型進行預測的原因是它在處理 時間序列數據 (例如隨時間變化的 YouBike 租借數量) 時非常有效。簡單來說, ARIMA 模型可以幫助我們根據過去的數據趨勢, 預測未來的數量變化, 適合短期預測 (重點!)。

為何想要做車輛數量預測呢?

當在做功能發想的時候, 從我的需求出發開始思考, 感覺似乎 YouBike 相關的功能做的已經很完整了, 都沒什麼新意, 因此我想到之前在補習的時候, 很常需要借 YouBike, 當時看到介面上明明還有腳踏車, 但是走到的時候卻沒車了, 因此我想結合導航估算使用者到達的時間, 並做簡單的預測數量功能。



```
PS C:\Users\Tosti> & C:\Python312\python.exe c:\Users\Tosti\Downloads\YouBike_Program\ubike_data_crawler.py
c:\Users\Tosti\Downloads\YouBike_Program\ubike_data_crawler.py:56: DeprecationWarning: The default datetime adapter is d
eprecated as of Python 3.12; see the sqlite3 documentation for suggested replacement recipes
  cursor.execute(insert_sql, (
Data stored successfully at 2024-10-27 00:26:55.172431
Data stored successfully at 2024-10-27 00:27:55.632424
Data stored successfully at 2024-10-27 00:28:56.123271
Data stored successfully at 2024-10-27 00:29:56.584043
Data stored successfully at 2024-10-27 00:30:57.057154
Data stored successfully at 2024-10-27 00:31:57.487602
Data stored successfully at 2024-10-27 00:32:57.962711
Data stored successfully at 2024-10-27 00:33:58.442312
Data stored successfully at 2024-10-27 00:34:58.928895
Data stored successfully at 2024-10-27 00:35:59.362426
Data stored successfully at 2024-10-27 00:36:59.816424
Data stored successfully at 2024-10-27 00:38:00.425971
Data stored successfully at 2024-10-27 00:39:00.878044
Data stored successfully at 2024-10-27 00:40:01.334261
Data stored successfully at 2024-10-27 00:41:01.693955
Data stored successfully at 2024-10-27 00:42:02.402938
Data stored successfully at 2024-10-27 00:43:02.981103
Data stored successfully at 2024-10-27 00:44:03.378369
Data stored successfully at 2024-10-27 00:45:03.853636
Data stored successfully at 2024-10-27 00:46:04.305239
Data stored successfully at 2024-10-27 00:47:04.791286
Data stored successfully at 2024-10-27 00:48:05.288129
Data stored successfully at 2024-10-27 00:49:05.723457
Data stored successfully at 2024-10-27 00:50:06.233678
Data stored successfully at 2024-10-27 00:51:06.728363
Data stored successfully at 2024-10-27 00:52:07.173781
```

←執行動態儲存