

銘 傳 大 學

資 訊 工 程 學 系

專 題 研 究 總 審 文 件

本校一一二學年度 資訊工程學系

組員： 林佳歆、蕭育君
洪玉珍、苗程皓

所提專題研究： 基於紐約計程車軌跡數據之載客熱點分析

指導教授： 李御璽

中華民國 一 一 二 年 十 二 月 七 日

基於紐約計程車軌跡數據之載客熱點分析

摘要

在城市交通中，計程車起到重要的作用，提供乘客便捷、舒適和快速的交通選擇，特別滿足偏遠地區和弱勢族群的出行需求。但計程車行業面臨的一個重要問題是空車率，在沒有乘客的情況下，計程車無法增加收入，還會增加燃油成本。為了解決這個問題，我們計畫提高計程車司機的載客率，縮短空車和乘客等待時間。通過地圖熱點視覺化，我們可以清晰瞭解乘客需求的分佈情況，並識別出高需求的密集區域。面對這種情況，我們可以建議計程車司機前往這些密集區域，以滿足潛在乘客需求，提高載客率，減少計程車空車駕駛，降低成本，提高整體出行效率。乘客方面，希望可以用一個網頁完成出行規劃和行程建議。

本研究計畫是利用 Streamlit 和 Flask 創建一個互動式平臺，視覺化紐約地區的熱點分佈，供司機、乘客使用。通過分析大數據，我們將為司機提供最佳路線建議，人流聚集地，以減少空車時間，為乘客提供計程車聚集地，以及有關下車地點周邊景點和行程時間及費用的資訊，讓乘客出行前可以規畫路徑行程的工具，提高乘客的出行體驗。

關鍵詞：計程車空車率、乘客需求熱點

目錄

摘要.....	I
圖目錄.....	IV
第一章 緒論.....	1
1-1 研究背景	1
1-2 研究動機	1
1-3 研究問題	2
第二章 文獻回顧與探討	3
2-1 DBSCAN 聚類算法	3
2-2 軌跡路徑	6
2-3 地圖可視化	6
第三章 研究方法	8
3-1 提取數據集	8
3-2 數據預處理	9
3-3 資料視覺化方法及工具	9
3-4 用戶身份選擇與其相對功能	10
3-4-1 司機.....	10
3-4-2 乘客.....	10
3-5 資料視覺化和交互功能	11
3-5-1 數據視覺化.....	11
3-5-2 交互功能.....	11
第四章 系統實作及測試	13
4-1 資料前處理	13
4-1-1 異常值刪除及格式處理.....	13

4-1-2 資料分區	13
4-2 系統功能與實測結果	15
4-2-1 Streamlit.....	15
4-2-2 Flask.....	25
4-3 模型預測	28
第五章 結論.....	30
參考文獻.....	31

圖目錄

圖 2-1 Sample databases.....	3
圖 2-2 聚類過程表示圖	5
圖 3-1 系統架構圖	8
圖 3-2 刪除有異常值的資料	9
圖 4-1 異常值刪除及格式處理	13
圖 4-2 分區程式碼	14
圖 4-3 密度熱點資料合併程式碼	14
圖 4-4 Streamlit 時段熱點分析頁面	15
圖 4-5 Streamlit 區域收入頁面	16
圖 4-6 Streamlit 區域收入頁面環狀圖	16
圖 4-7 Streamlit 最短路徑參數調整	17
圖 4-8 Streamlit 最短路徑路線規劃及導航	18
圖 4-9 Streamlit 最短路徑計算結果	18
圖 4-10 Streamlit 油資計算頁面	19
圖 4-11 Streamlit 人流聚集地參數調整	20
圖 4-12 Streamlit 人流聚集地標點圖	20
圖 4-13 Streamlit 附近觀光景點頁面	21
圖 4-14 Streamlit 行程費用提示參數調整	22
圖 4-15 Streamlit 最短路徑路線規劃	22
圖 4-16 Streamlit 最短路徑計算結果	23

圖 4-17 Streamlit 行程時間計算參數調整	23
圖 4-18 Streamlit 行程時間計算結果	24
圖 4-19 Streamlit 計程車聚集地參數調整	24
圖 4-20 Streamlit 計程車聚集地標點	25
圖 4-21 Flask 角色選擇頁面	25
圖 4-22 Flask 司機路線規劃	26
圖 4-23 Flask 司機收入計算	26
圖 4-24 Flask 路徑規劃及導航	27
圖 4-25 Flask 區域熱點圖	27
圖 4-26 Flask 區域熱視圖放大	28
圖 4-27 Streamlit 模擬預測熱點分析圖	28
圖 4-28 Flask 模擬預測熱點分析圖	29

第一章 緒論

1-1 研究背景

在城市交通中，計程車扮演著不可或缺的角色，為乘客提供了方便、舒適且快速的交通選擇。尤其對於偏遠地區和弱勢族群而言，計程車成為了滿足其出行需求的主要方式。然而，計程車行業面臨著一項重要問題，即空車率。在沒有乘客的情況下，計程車無法增加收入，同時還需要應對不必要的燃油成本。為了有效解決這個問題，我們計畫提高計程車司機的載客率，以縮短空車和乘客等待時間。

1-2 研究動機

計程車行業的空車率問題對於行業可持續性和司機收入有著直接影響。為了改進這一現象，我們希望利用現代技術和數據分析方法，創建一個互動式平臺，幫助計程車司機更有效地適應市場需求，提高載客率，同時為乘客提供更好的出行體驗。通過這項研究，我們也有望降低計程車行業的成本，提高整體出行效率。

1-3 研究問題

本研究將聚焦於解決計程車行業面臨的空車率問題，具體而言，我們將致力於以下幾個方面的研究問題：

1. 空車率分析

如何準確測定計程車在不同區域的空車率，以了解市場需求的變化和司機載客效能？

2. 地圖熱點識別

如何透過地圖熱點視覺化，清晰瞭解乘客需求的分佈情況，特別是識別高需求的密集區域？

3. 調度策略

如何制定有效的調度策略，提前引導計程車司機前往高需求密集區域，以最大程度地滿足潛在乘客需求，降低空車率？

4. 使用者體驗提升

如何通過分析大數據，為計程車司機提供最佳路線建議，同時提供乘客有關下車地點周邊景點和行程費用的資訊，以提高乘客的整體出行體驗？

通過對這些研究問題的深入探討，我們期望能夠為計程車行業的可持續發展和城市出行體驗做出實質性的貢獻。

第二章 文獻回顧與探討

2-1 DBSCAN 聚類算法

在 A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise[1]提到的 DBSCAN (Density-Based Spatial Clustering of Applications with Noise) 是一種聚類算法，用於將數據點分組成具有高密度區域的簇，同時能夠檢測並排除低密度區域的雜訊點。它是一種基於密度的聚類方法，與傳統的 K 均值等算法相比，DBSCAN 不需要事先指定簇的數量，並且能夠發現任意形狀和大小的簇。DBSCAN 的主要思想是通過測量每個數據點周圍的密度，將數據分為三類：核心點、邊界點和雜訊點。核心點是指在指定半徑內具有指定數量以上的鄰居的點，邊界點是在指定半徑內的鄰居數量未達到指定數量的點，而雜訊點則是既不是核心點也不是邊界點的點。我們可以根據資料點近鄰數來定義空間密度的方式，先指定群的數量，並且我們也可以找出空間中形狀不規則的群，如圖 2-1。

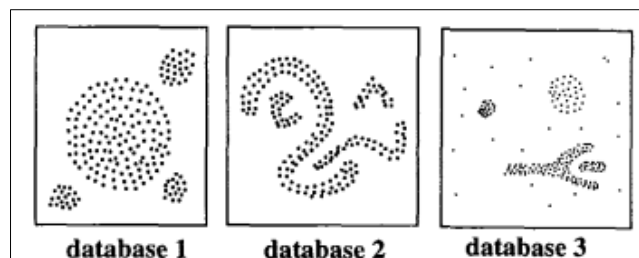


圖 2-1 Sample databases

資料來源：Martin Ester 等[1]

DBSCAN 聚類算法使用參數(ϵ , minPts)描述密度，其中 ϵ 為選定點的空間半徑閾值，minPts 是半徑 ϵ 下最小包含樣本數閾值。核心思

想包括點的 Eps 鄰域、核心點定義，以及密度可達概念。優勢包括不需預先聲明聚類數量、處理任意形狀數據集、對異常點不敏感，結果無偏倚。

1. 一個點的 Eps 鄰域：需要至少有一個比距離 Eps 更靠近的點，即群內至少有一個點距離小於閾值。

$$N_{Eps}(p) = \{q \in D \mid \text{dist}(p, q) < Eps\} \quad (\text{式 2-1})$$

2. 設定 p 為核心點： p 與所有可達點構成一個群，包含核心點和非核心點，每個群至少有一個核心點，非核心點可以位於群的邊界。

3. 核心點是密度可達：存在一系列點，可以直接密度到達該核心點，即密度可達。

4. 兩個邊界點屬於同一集群：兩邊界點需共享特定核心點，否則彼此無法密度到達。

5. 存在樣本點 o ，使得 p 和 q 均由樣本點 o 密度可達，即樣本點 p 和樣本點 q 對於參數 $\{Eps, \text{MinPts}\}$ 是密度相連的。

聚類過程，從這些大量數據集中，需要先找到任意一個核心點，從這個核心點去出發的所有密度相連的數據點，接著去找與這些數據點密度相連的點，一直到沒有可以增加的數據點為止。當聚類成的群集的邊界點都是非核心數據點時，就可以再重新在尋找一次數據集，去尋找沒有被聚類的核心點，然後必須一直重複上面的步驟，直到數據集中沒有新的核心點為止，如圖 2-2。

(數據集中沒有包含在任何群中的數據點就會構成異常點)

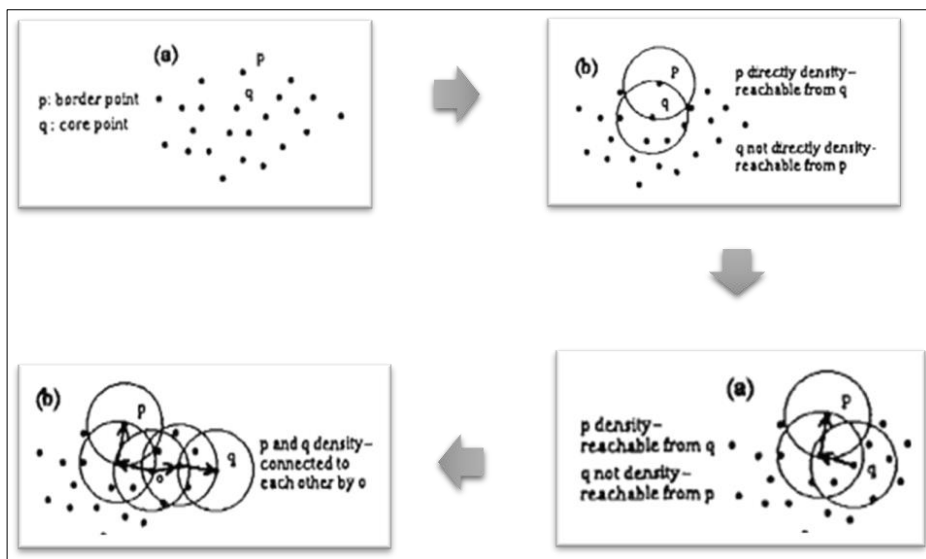


圖 2-2 聚類過程表示圖

資料來源：Martin Ester 等[1]

馮琦森[2]等人通過對出租車軌跡和點數據的聚類方法研究，發現不同時間居民出行的路徑熱點和出租車上下客區域。他們提出了一種基於最長公共子序列的車輛軌跡相似性度量方法，結合 DBSCAN 聚類形成 LCS-BASED DBSCAN。透過軌跡的海拔高度改進相似度計算，並應用 LCS-BASED DBSCAN 對有效軌跡進行聚類。再通過 LCS-BASED HRE 熱點路徑提取算法，提取不同時間段的居民出行熱點路徑。同時，基於歐式距離的 DBSCAN 聚類算法用於出租車上下客軌跡點的分析，獲取早中晚不同時間段的熱點區域。

李文傑、閔世強、蔣瑩、張松芝、王成良等人[3]在 DBSCAN 演算法基礎上引入密度閾值，提出了一種自我調整確定 Eps 和 MinPts 參數的算法。通過 K-平均最近鄰法生成候選 Eps 參數，利用數學期望法獲得 MinPts 參數，自動尋找聚類結果簇數的變化穩定區間。最終以雜訊最少的簇數變化穩定區間內的 Eps 和 MinPts 值作為最優參數，實現全自動化聚類，無需人為干預。

2-2 軌跡路徑

郭哲倫等人[4]做出一套模擬系統，模擬司機登入營運過程，之後會在模擬出好幾位乘客在不同的位置叫車服務，並從後台系統運算出最符合司機乘載的三位客人與司機配對（若無三位適合，則會以兩位為主），透過「最長共同子路徑」演算法篩選至多 3 位乘客共乘，並據以決定計程車司機的載客起點與終點，並且觀察由最長共同子路徑演算法配對出的乘客是否對於司機為最佳排程。

吳俊廷等人[5]介紹了一種名為 Flexi-Sharing 的出租車共享系統，其主要目的是透過個性化配對服務和有效路徑計算，降低出租車費用和行車距離，提供乘客更靈活的共享路線。系統透過考慮乘客的個人偏好，包括步行距離、等待時間、行車距離和交通費用，以最大程度地滿足共享需求，並在實驗中顯示出在城市中可提供卓越的共享體驗和迅速的系統回應時間。

2-3 地圖可視化

劉琳等人[6]計程車 GPS 軌跡資料，通過 Python 語言在 Spark 環境下，對其進行資料預處理，為下文研究內容奠定資料基礎。挖掘計程車載客行為的時空特徵，通過數學統計方法計算計程車各個時段內的載客時長、載客量及其變化趨勢，對計程車載客行為進行時序特徵分析，同時利用地理資訊系統 ArcGIS 軟體的核密度分析，對計程車載客行為的空間分佈進行視覺化展示，分析其載客空間分佈特徵。

許龍哲等人[7]將深度學習與智慧交通結合，並且建置一個的計程車載客熱點預測平台作為示範，將計程車上專用行車紀錄器所記錄的車用資訊以及 GPS 定位座標的資料進行處理，並且結合座標資

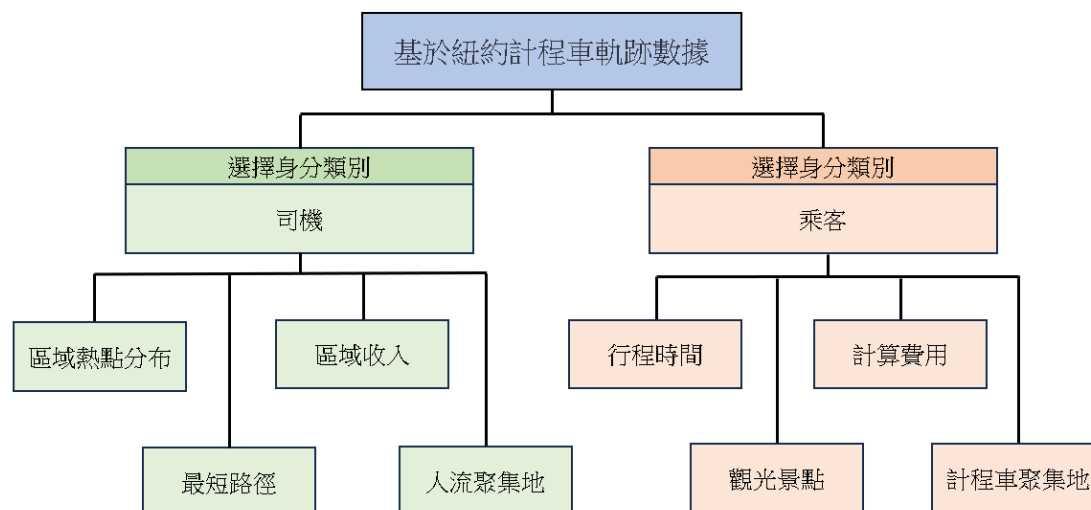
料與過去載客熱點的資料，把地理座標資料進行處理將其轉換成地圖能運用的向量資料，利用深度學習的模型去進行未來熱點的預測，完成後與向量資料結合繪製地圖，最後放上 PythonAnywhere 執行，資料在地圖上將熱點需求視覺化，並且以視覺化的方式在網頁顯示，提供計程車司機參考，希望提供有價值的熱點需求分布以達到派遣車輛之目的。

王貝貝等人[8]利用出租車 GPS 數據挖掘出租車需求時空分布特徵，通過對出租車 GPS 數據進行預處理，利用地圖匹配算法，將出租車軌跡點映射到實際路網中，通過 C#語言編程提取上下客點，計算出租車出行需求量，分析出租車出行需求時空分布特徵，結合 ArcGIS 工具繪製出租車出行需求空間分布圖，基於空間聚類算法挖局出租車載客熱點區域。

林娜、鄭亞男等人[9]透過對出租車收集的 GPS 數據進行統計和分析，研究了出租車司機在道路選擇上的經驗和規律。為了提高路徑規劃的效率，文中提出了一種基於經驗等級路網的層次搜索算法。與傳統的最短路徑規劃算法和常用的 Logit 模型相比，該算法具有兩方面的優勢：一是利用層次搜索策略縮減了搜索空間，提高了算法效率；二是融合了出租車司機的智能，得到的是動態路網，能夠真實地反映不同時段的路況，使得結果更加合理且具有實時性。

第三章 研究方法

本研究使用 Streamlit 和 Flask 創建的互動式應用程式，用於基於紐約計程車軌跡資料提供不同功能和資訊，如圖 3-1。



3-1 提取數據集

目前是透過網路找到紐約 Taxi & Limousine Commission(TLC)官方網站發布的紐約市計程車軌跡數據集，時間為 2013 年的數據。

`pickup_datetime` 上車的日期和時間

`dropoff_datetime` 停表的日期和時間

`trip_time_in_secs` 行程時長

`trip_distance` 行程持續時間（秒）

`pickup_longitude` 上車的經度

`pickup_latitude` 上車的緯度

`dropoff_longitude` 下車經度

`dropoff_latitude` 下車的緯度

3-2 數據預處理

計程車的數據集都有很多行駛的軌跡點，每條載客紀錄都有行駛點，數據集裡有基本的計程車資訊，包括行程的編號、上下車的日期和時間、下車的經緯度、行程的持續時間、停表的日期和時間、車輛中的乘客數量，一開始會先進行了一定程度的預處理，將計程車數據裡的異常點、噪音點、格式錯誤、重複、不完整的數據等一些有問題的資料先進行數據清理，獲取計程車的上車地點以及下車地點、若不清理就直接使用可能導致結果的準確性，如圖 3-2。

```
In [1]: import pandas as pd
        from collections import Counter
        url = 'trip_data_1.csv'
        data = pd.read_csv(url,encoding='latin1',low_memory=False)
        print(data.isnull().sum())
```

medallion	0
hack_license	0
vendor_id	0
rate_code	0
store_and_fwd_flag	7326207
pickup_datetime	0
dropoff_datetime	0
passenger_count	0
trip_time_in_secs	0
trip_distance	0
pickup_longitude	0
pickup_latitude	0
dropoff_longitude	86
dropoff_latitude	86
dtype: int64	

圖 3-2 刪除有異常值的資料

3-3 資料視覺化方法及工具

為了分析基於紐約計程車軌跡資料的各種資訊，我們選擇使用了 Streamlit 和 Flask 兩種工具。我們選擇 Streamlit，因為它是一個專為資料應用程式開發而設計的開源 Python 庫。Streamlit 的主要目標是使資料科學家和分析師能夠快速、簡單地構建資料應用程式，無需深入瞭解 Web 開發。該庫內置了許多常用元件，如按鈕、單選框

、圖表等，這使使用者能夠輕鬆創建使用者友好的介面。同時，我們也選擇了 Flask，因為它提供了更大的靈活性，通常用於建構訂製的 Web 應用程式。Flask 允許開發者完全控制應用程式的路由、資料存儲和範本等方面，這在構建複雜的 Web 應用程式時非常有用。

3-4 用戶身份選擇與其相對功能

基於紐約計程車軌跡資料的應用程式旨在為計程車司機和乘客提供定制化的服務和資訊。

3-4-1 司機

計程車司機需要即時的路況資訊，以選擇最佳路線，以及瞭解熱門乘車時段，以提高收入。此外，他們還需要計算油費以優化成本。因此，我們提供了以下功能：

1. 時段熱點分析：幫助司機瞭解不同時間段的需求高峰。
2. 收入查詢：允許司機查看特定日期的收入情況。
3. 最短路徑建議：為司機提供最佳路線建議。
4. 行程搭乘時間建議：幫助司機避開交通高峰，提供出行時間建議。
5. 油費計算：協助司機估算油費成本。

3-4-2 乘客

乘客尋找方便、經濟和愉快的乘車體驗。他們通常關心附近的觀光地點、行程費用、計程車聚集地點和行程時間。因此，我們提供了以下功能：

1. 附近觀光地點：展示乘客附近的美食、景點等資訊，以增強旅行體驗。

2. 行程費用提示：允許乘客估算行程費用，並了解周末或尖峰時段的搭乘費用。
3. 計程車聚集地點提示：幫助乘客找到附近的計程車聚集點。
4. 行程時間路線：提供估計的行程時間與路線，幫助乘客計畫行程。

3-5 資料視覺化和交互功能

3-5-1 數據視覺化

1. 熱點密度圖

根據使用者的選擇，應用程式將創建一個熱點密度圖，用來展示在選定時間段內，上車地點的密度。這個熱點密度圖使用 Plotly Express 庫來繪製。

2. 地圖視覺化

使用者使用了 folium 庫來創建地圖，同時在地圖上添加標記、路線等元素，以視覺化地展示路線和地點資訊。

3. 社區收入環形圖

使用者可以使用 Matplotlib 庫繪製一個環形圖，顯示不同社區的平均美元收入，圓形圖中的每個部分表示一個社區，其面積與社區的收入相關。

3-5-2 交互功能

1. 區域選擇框：使用者可以選擇要查看的區域（例如，布魯克林、曼哈頓等）。這些選擇將用於載入相應的資料檔案。通過選擇不同的區域和時間段，使用者可以展示所選的資料。

2. 載入數據：根據使用者的選擇，選擇要載入的資料檔案，

這些資料包含計程車的上/下車位置資訊和時間戳記。用戶可以通過這個功能來切換不同的資料來源，從而查看不同時間段內的上車地點密度。

3. 選擇時間段：用戶可以在視覺化中選擇時間段，例如，選擇上午8點到9點之間的時間。這允許用戶比較不同時間段內的上車地點密度，以便觀察變化。

4. 地圖交互性：使用者可以在生成的熱點密度圖上進行交互。這包括縮放（放大和縮小地圖）和拖動地圖以查看不同區域的細節。這增加了用戶的自由度，使他們能夠更深入地探索資料，查看不同位置的密度情況。

5. 石油價格查詢

使用 `scrape_oil_prices` 函數，<https://www2.moeaea.gov.tw/oil111/> 中抓取石油 92、95、98 的價格再轉換成美國標示，87、89、92。它使用 BeautifulSoup 庫來解析網頁內容，找到相應的價格資訊。

6. Google Maps Directions API 獲取路線資訊

使用 `googlemaps` 用戶端物件 (`gmaps`) 向 Directions API 發送請求，獲取從起點到終點的駕車路線資訊，在接收到 Directions API 的回應後，提取了路線中每個步驟的起始座標，並將這些座標用於在地圖上繪製導航路徑。

7. DBSCAN 算法顯示計程車的聚集地點


使用 DBSCAN 演算法對資料進行聚類，找到並顯示計程車的聚集地點，我們選擇適當的 DBSCAN 參數，包括 `epsilon (eps)` 和最小樣本數 (`min_samples`)，這些參數的選擇取決於所研究的區域和資料的特性，最後將聚類結果在地圖上視覺化，並進一步呈現計程車的聚集地點。

第四章 系統實作及測試

4-1 資料前處理

4-1-1 異常值刪除及格式處理

如圖 4-1。



```
In [7]: #刪除缺失值行
data = data.dropna()
#旅行時間異常值
data = data[data['trip_time_in_secs'] >= 10]
#更改日期變數格式
data['pickup_datetime'] = pd.to_datetime(data.pickup_datetime)
data.loc[:, 'pickup_datetime'] = data['pickup_datetime'].dt.date
data['dropoff_datetime'] = pd.to_datetime(data.dropoff_datetime)
data.head(5)
```

index	pickup_datetime	dropoff_datetime	passenger_count	trip_time_in_secs	trip_distance	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude
0	2013-01-01	2013-01-01 15:18:10	4	382	1.0	-73.978165	40.757977	-73.989838	40.751171
1	2013-01-06	2013-01-06 00:22:54	1	259	1.5	-74.006683	40.731781	-73.994499	40.750660
2	2013-01-05	2013-01-05 18:54:23	1	282	1.1	-74.004707	40.737770	-74.009834	40.726002
3	2013-01-07	2013-01-07 23:58:20	2	244	0.7	-73.974602	40.759945	-73.984734	40.759388
4	2013-01-07	2013-01-07 23:34:24	1	560	2.1	-73.976250	40.748528	-74.002586	40.747868

圖 4-1 異常值刪除及格式處理

4-1-2 資料分區

原始資料區分為月份，總共十二份檔案，為了讀取資料及網頁顯示更快速還有直覺性，先將不同區域提取出來，利用經緯度去圍成一個矩形提取，以下為分區程式碼，如圖 4-2。

```

1 import pandas as pd
2
3
4 df = pd.read_csv(r'C:\Users\Jasmine\Downloads\trip_data\trip_data_12.csv')
5
6
7 Bronx_longitude_range = (-73.933518, -73.782310)
8 Bronx_latitude_range = (40.796639, 40.915254)
9
10
11 pickup_longitude_in_range = df['pickup_longitude'].between(*Bronx_longitude_range)
12 pickup_latitude_in_range = df['pickup_latitude'].between(*Bronx_latitude_range)
13 dropoff_longitude_in_range = df['dropoff_longitude'].between(*Bronx_longitude_range)
14 dropoff_latitude_in_range = df['dropoff_latitude'].between(*Bronx_latitude_range)
15
16
17 df['is_Bronx'] = pickup_longitude_in_range & pickup_latitude_in_range & dropoff_longitude_in_range & dropoff_latitude_in_range
18
19 Bronx_data = df[df['is_Bronx']]
20
21 Bronx_data.to_csv('Bronx_taxi_data_12.csv', index=False)
22
23
24 print("成功將數據保存為新的csv文件!")

```

圖 4-2 分區程式碼

再將各區域 1 到 6 月、7 到 12 月，隨機取前 10000 筆資料合併成為上下半年兩個檔案，為 Streamlit 密度熱點頁面資料來源，如圖 4-3。

```

1 import pandas as pd
2 import random
3
4 # 要從每個檔案中隨機抽取的筆數
5 sample_size = 10000
6
7 # 創建一個空的DataFrame來儲存合併的結果
8 combined_data = pd.DataFrame()
9
10 # 迴圈處理每個檔案
11 for i in range(1, 7):
12     # 讀取CSV檔案
13     file_path = f'Queens_taxi_data_{i}.csv'
14     df = pd.read_csv(file_path)
15
16     # 隨機抽取指定筆數的資料
17     sampled_data = df.sample(n=sample_size, random_state=random.seed(i))
18
19     # 將抽取的資料合併到結果中
20     combined_data = pd.concat([combined_data, sampled_data], ignore_index=True)
21
22 # 儲存合併後的資料
23 combined_data.to_csv('Queens_taxi_data_123456.csv', index=False)
24

```

圖 4-3 密度熱點資料合併程式碼

4-2 系統功能與實測結果

4-2-1 Streamlit

司機功能：

1. 時段熱點分析頁面

網頁設計：互動式地圖，視覺化顯示熱點區域。

功能：使用者可以選擇時間段，查看熱點區域，地圖上標記熱點位置，如圖 4-4。



圖 4-4 Streamlit 時段熱點分析頁面

2. 區域收入查詢頁面

網頁設計：輸入日期，選擇區域等，顯示收入資訊。

功能：使用者輸入日期和區域，系統返回該區域的收入情況，

如圖 4-6、4-5。



圖 4-5 Streamlit 區域收入頁面

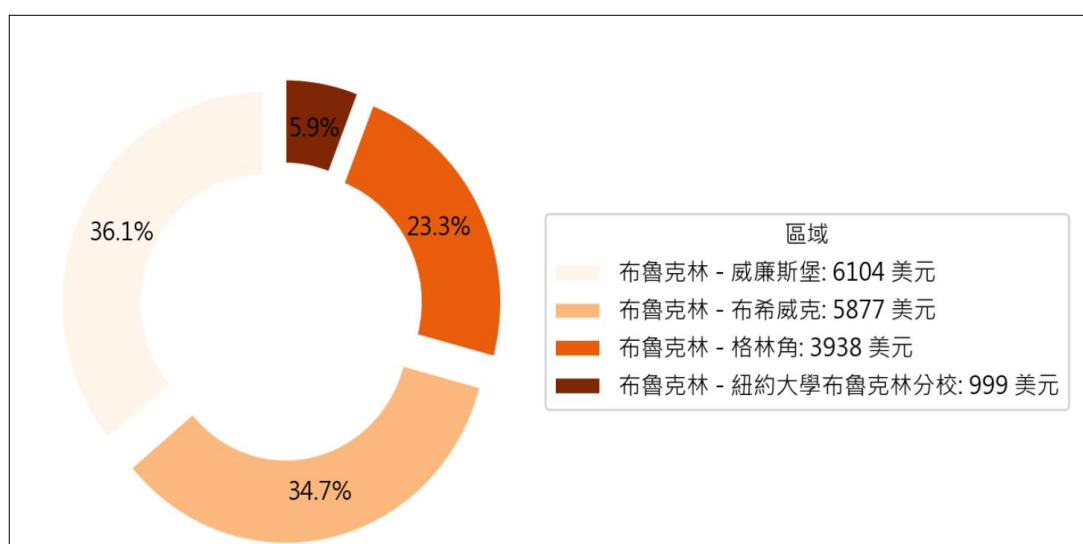


圖 4-6 Streamlit 區域收入頁面環狀圖

3. 最短路徑建議頁面

網頁設計：使用者輸入起點和終點，顯示最短路徑建議。

功能：提供使用者輸入起點和終點，顯示最短路徑在地圖上，並包括交通情況、時間和距離信息，以幫助使用這進行到航和路線規劃，如圖 4-7、4-8、4-9。

基於紐約計程車軌跡數據 🚕

最短路徑

輸入或選擇起點地點的名稱:

選擇起點地點:

Brooklyn Bridge

輸入或選擇終點地點的名稱:

選擇終點地點:

Park Slope

時間設定

選擇時間幾點

9

0

23

請選擇今天是星期幾:

星期一

選擇時間幾分

22

0

59

請輸入尖峰時間的倍率

1.5

輸入附加費費用

請輸入夜間附加費 (美元)

0.50

請輸入尖峰時間附加費 (美元)

1.00

顯示費用/時間

圖 4-7 Streamlit 最短路徑參數調整

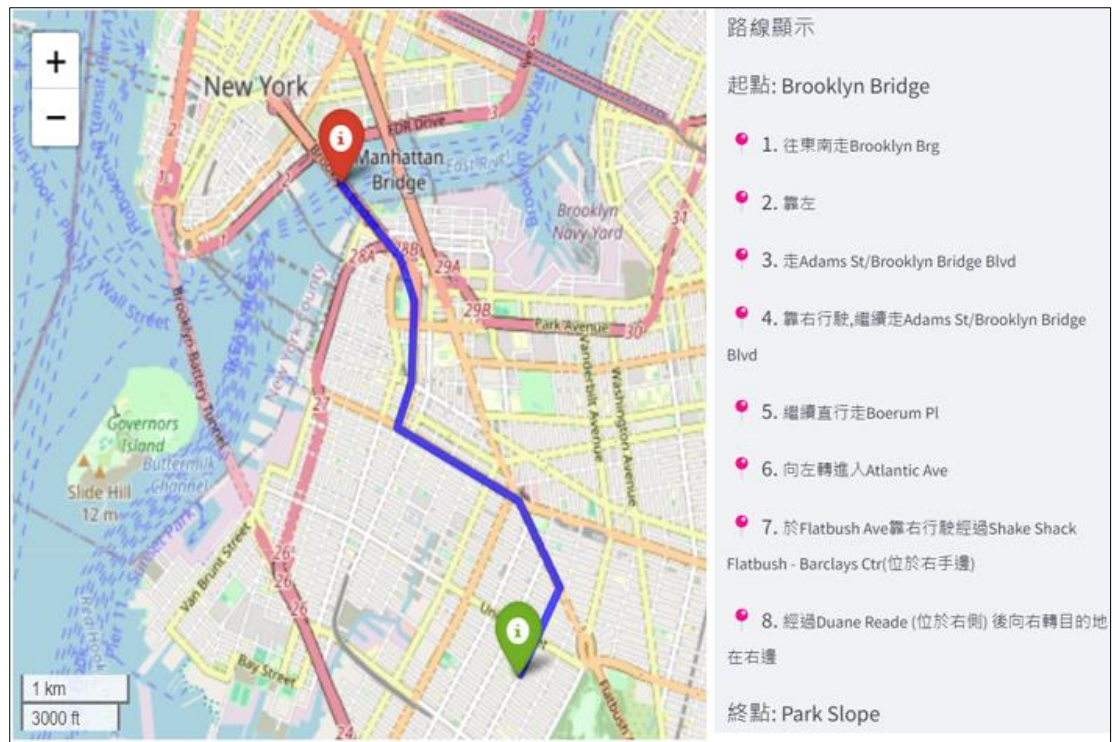


圖 4-8 Streamlit 最短路徑路線規劃及導航

	項目	結果
0	起點	Brooklyn Bridge (40.7061, -73.9969)
1	終點	Park Slope (40.6721, -73.9776)
2	起點和終點之間的距離	4.11 公里
3	時間分類	離峰时间
4	估計行車時間	8.23 分鐘
5	費用(美元)	11.97 美元
6	費用(台幣)	366.00 台幣

圖 4-9 Streamlit 最短路徑計算結果

4. 油費計算頁面

網頁設計：使用者用戶可以輸入要查詢的石油類型（例如：87、89、92），以及行駛的距離和汽車的油耗，顯示油費估算。

功能：使用者輸入相關資訊，系統計算油費估算，如圖 4-10。

基於紐約計程車軌跡數據 🚕

油資查詢

請輸入要查詢的美國石油類型 (例如 : 87、89、92)

87

請輸入行駛的距離 (英里)

1.00 - +

請輸入汽車的油耗 (英里/加侖)

1.00 - +

計算

87 石油價格: \$1.04 美元/加侖

預計的油資為: \$1.04 美元

圖 4-10 Streamlit 油資計算頁面

5. 人流聚集地

網頁設計：幫助計程車司機瞭解人流聚集地點，以便更好地規劃其路線和服務。

功能：使用者可以自訂查看人流聚集點的時機，選擇特定的區域以查看該地區的人流聚集點，如圖 4-11、4-12。

基於紐約計程車軌跡數據 🚕

人流聚集地

選擇要顯示的月份

1月

選擇星期

星期一

選擇時間



你選擇時間是：06:00

選擇想看的區域

布魯克林

計程車聚集地點地圖（布魯克林）：

圖 4-11 Streamlit 人流聚集地參數調整

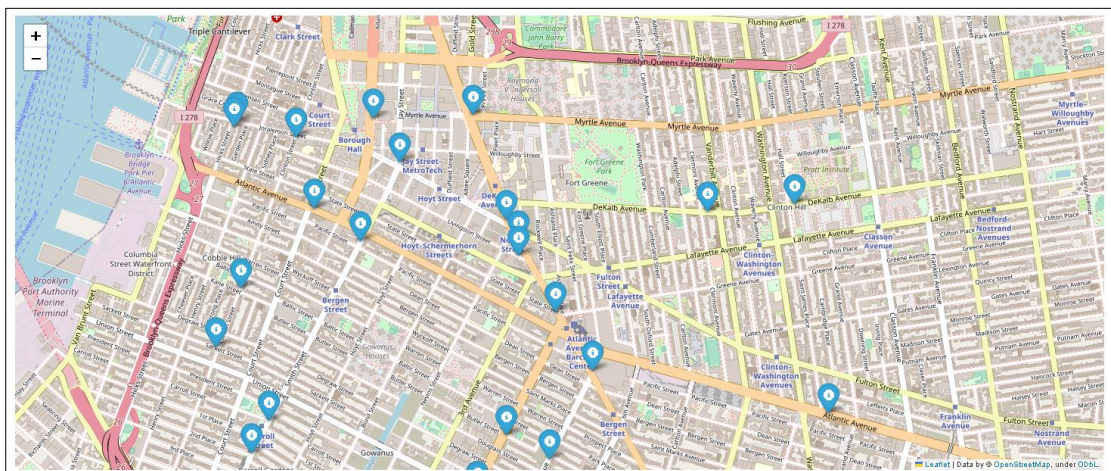


圖 4-12 Streamlit 人流聚集地標點圖

乘客功能：

1. 附近觀光地點頁面

網頁設計：顯示附近的美食、景點等資訊，地圖展示。

功能：根據乘客當前位置，顯示附近的觀光地點資訊，可以通過地圖查看位置，如圖 4-13。

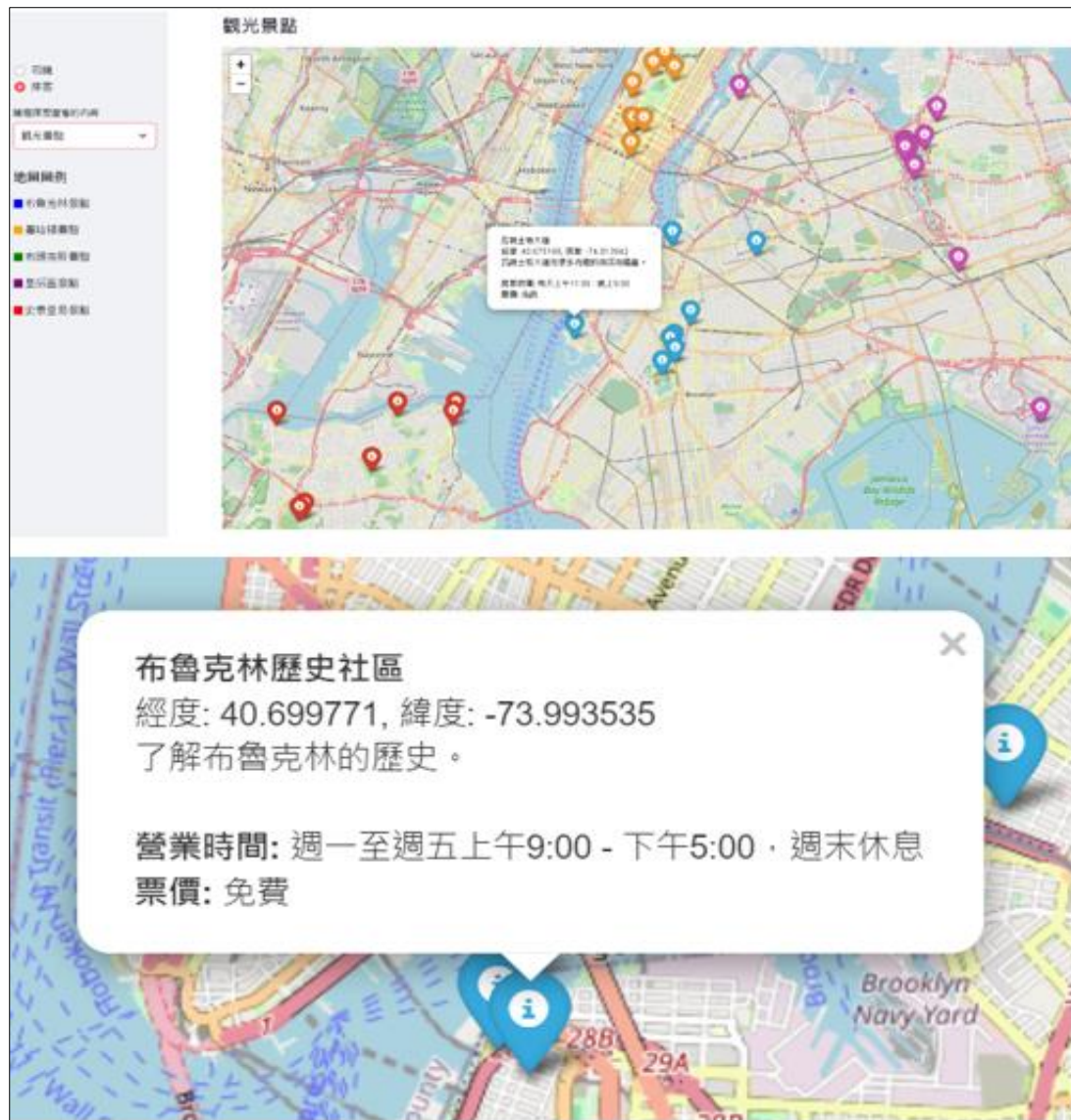


圖 4-13 Streamlit 附近觀光景點頁面

2. 行程費用提示頁面

網頁設計：使用者輸入行程相關資訊，顯示費用提示。

功能：使用者可以在網頁上輸入起點、終點、選擇時間，然後點擊"顯示費用/時間"按鈕，系統將計算行程費用、顯示地圖和路線，並在頁面中和側邊欄上展示相關資訊，如圖 4-14、4-15、4-16。

基於紐約計程車軌跡數據 🚕

最短路徑

輸入或選擇起點地點的名稱:

選擇起點地點:

Brooklyn Bridge

輸入或選擇終點地點的名稱:

選擇終點地點:

Park Slope

時間設定

選擇時間幾點

0

23

9

選擇時間幾分

0

59

22

請選擇今天是星期幾:

星期一

請輸入尖峰時間的倍率

1.5

輸入附加費費用

請輸入夜間附加費 (美元)

0.50

請輸入尖峰時間附加費 (美元)

1.00

顯示費用/時間

圖 4-14 Streamlit 行程費用提示參數調整

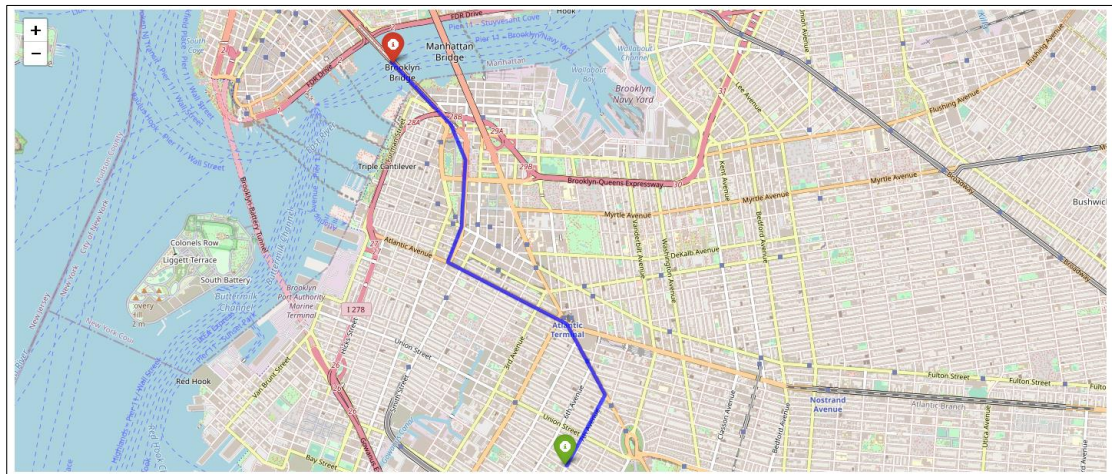


圖 4-15 Streamlit 最短路徑路線規劃

	項目	結果
0	起點	Brooklyn Bridge (40.7061, -73.9969)
1	終點	Park Slope (40.6721, -73.9776)
2	起點和終點之間的距離	4.11 公里
3	時間分類	離峰时间
4	估計行車時間	8.23 分鐘
5	費用(美元)	11.97 美元

圖 4-16 Streamlit 最短路徑計算結果

3. 行程時間頁面

網頁設計：使用者輸入起點和終點，顯示預計的行程時間。

功能：使用者輸入起點和終點，系統返回預計的行程時間，如圖 4-17、4-18。

基於紐約計程車軌跡數據 🚕

行程時間計算

選擇你想要的方式

在地圖上選擇點

輸入或選擇上車地點的名稱:

選擇上車地點:

Brooklyn Bridge

輸入或選擇下車地點的名稱:

選擇下車地點:

Park Slope

時間設定

請選擇現在時間(幾時)

8

0 23

請選擇今天是星期幾:

星期一

請選擇現在時間(幾分)

20

0 59

請輸入尖峰時間的倍率

1.5

- +

計算行程時間

圖 4-17 Streamlit 行程時間計算參數調整

	項目	結果
0	上車地點	Brooklyn Bridge (40.7061, -73.9969)
1	下車地點	Park Slope (40.6721, -73.9776)
2	現在時間	08:20
3	距離	4.11 公里
4	時間分類	尖峰时间
5	估計行車時間	12.34 分鐘

圖 4-18 Streamlit 行程時間計算結果

4. 計程車聚集地點

網頁設計：幫助乘客輕鬆找到附近的計程車聚集點，根據使用者的時間和日期選擇最佳的搭乘時機

功能：使用者可以自訂查看計程車聚集點的時機，選擇特定的區域以查看該地區的計程車聚集點，如圖 4-19、4-20。

計程車聚集地

選擇要顯示的月份

1月

選擇星期

星期一

選擇時間

0

6

23

你選擇時間是：06:00

選擇我想看的區域

布魯克林

計程車聚集地點地圖（布魯克林）：

圖 4-19 Streamlit 計程車聚集地參數調整

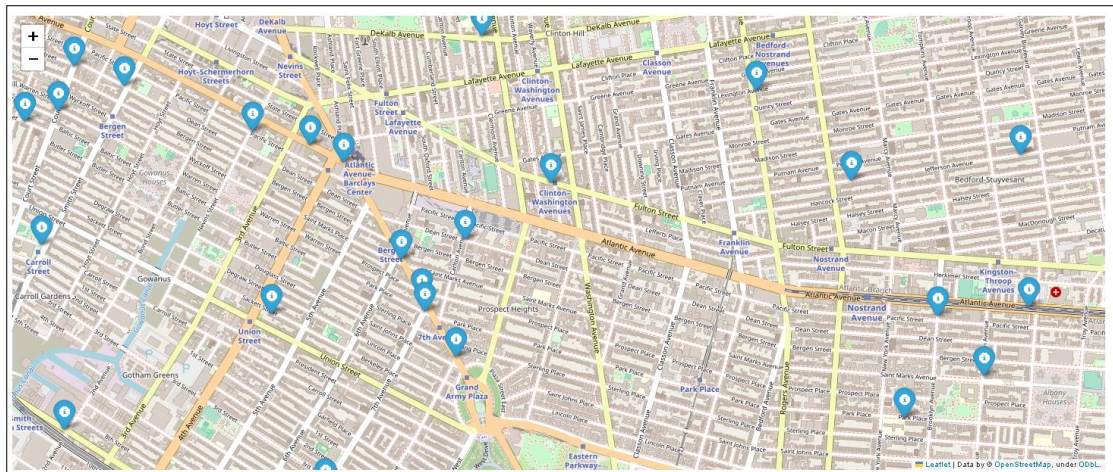


圖 4-20 Streamlit 計程車聚集地標點

4-2-2 Flask

1. 角色選擇，如圖 4-21。

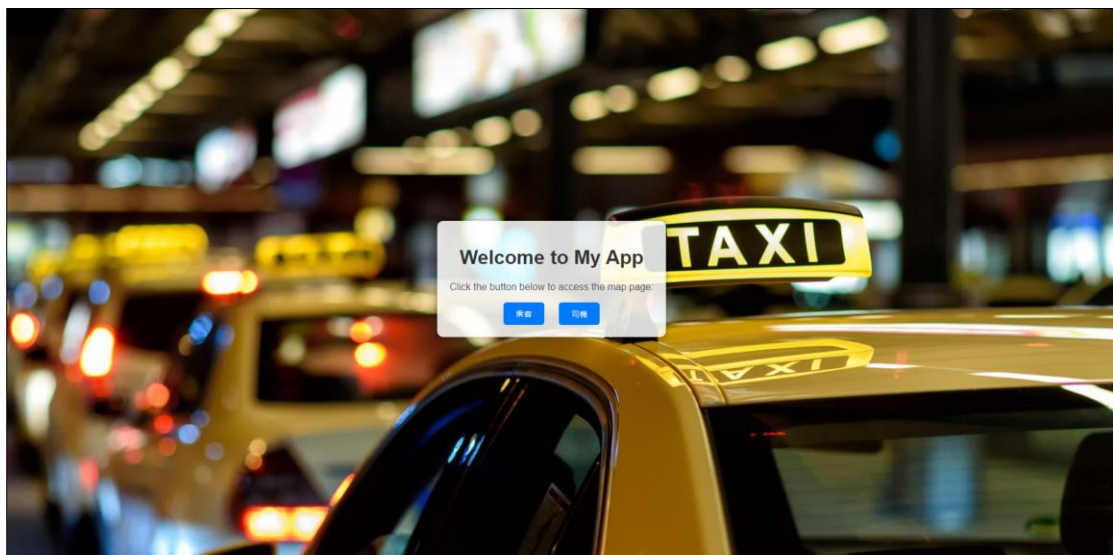


圖 4-21 Flask 角色選擇頁面

司機：最佳路徑

網頁設計：使用者移動圖案標示起點和終點，顯示預計的行程時間與路徑。

功能：移動圖案標示起點和終點，並且可以計算此趟行程的收入以及油耗量，如圖 4-22。



圖 4-22 Flask 司機路線規劃

2. 司機：指定區間，取得平均收入以及總收入，如圖 4-23。



圖 4-23 Flask 司機收入計算

3. 乘客：出遊時，可利用此功能規劃路線、車資以及推薦紐約五區分別的美食景點，如圖 4-24。

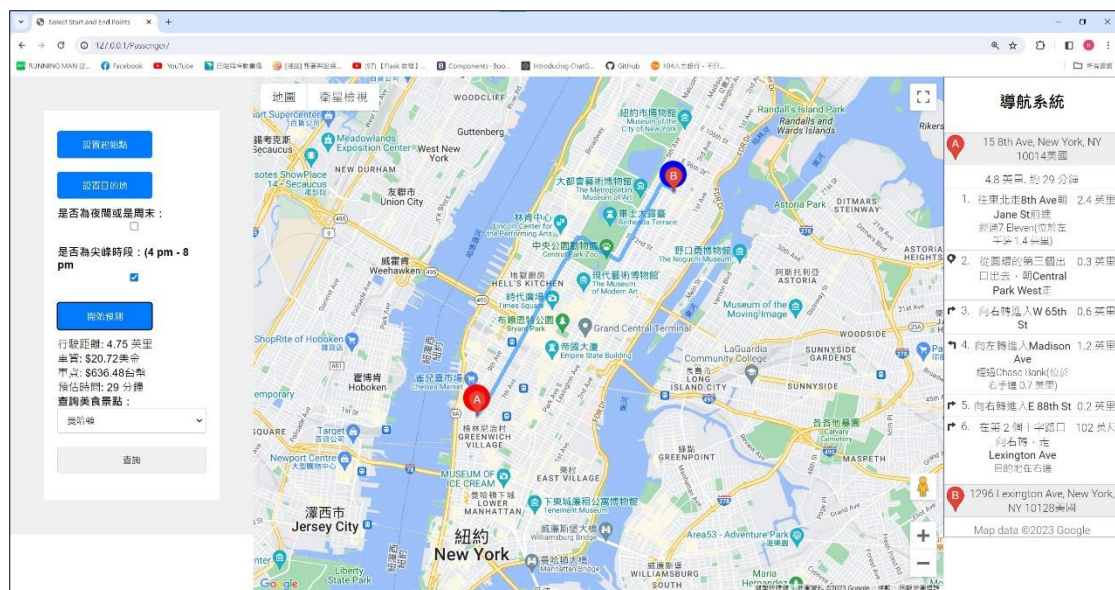


圖 4-24 Flask 路徑規劃及導航

4. 熱點可視化：司機可以了解到人流聚集地在哪，乘客可以了解到哪裡是最佳上車點，是雙向熱點整合的熱點圖。點擊那區域的熱點將會更詳細的放大釋出更多地點資訊，如圖 4-25、4-26。

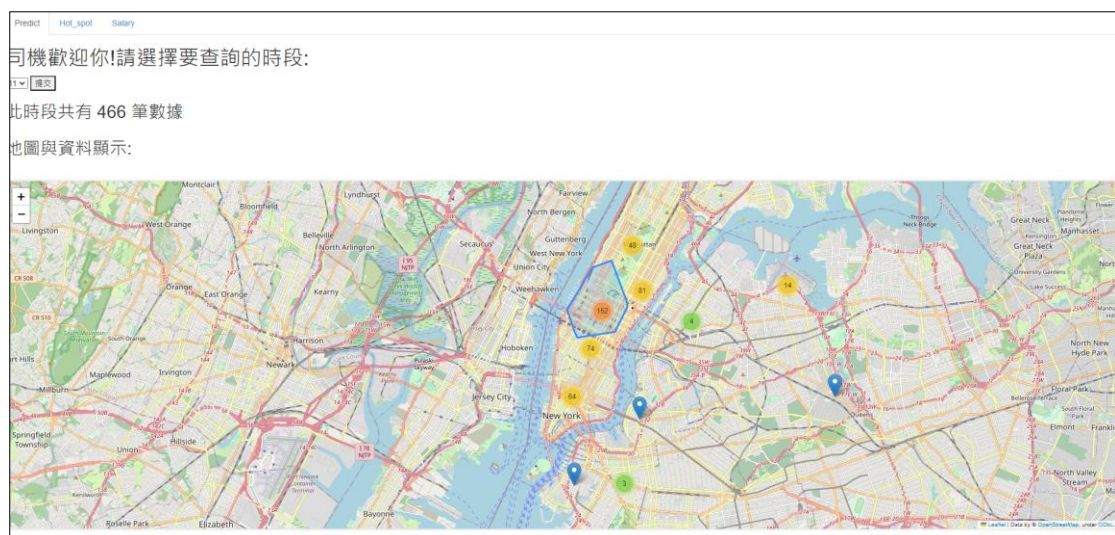


圖 4-25 Flask 區域熱點圖



圖 4-26 Flack 區域熱視圖放大

4-3 模型預測

我們的模擬預測主要旨在應對若有更即時的數據，我們將預先處理該數據集，並直接上傳至我們的互動式平台進行資料更新。這樣，我們的平台將能夠呈現更加即時的數據，提供用戶更準確和實用的信息。以下是採用於紐約市 2014 年 1 月份的數據集進行模擬預測，如圖 4-27、4-28。

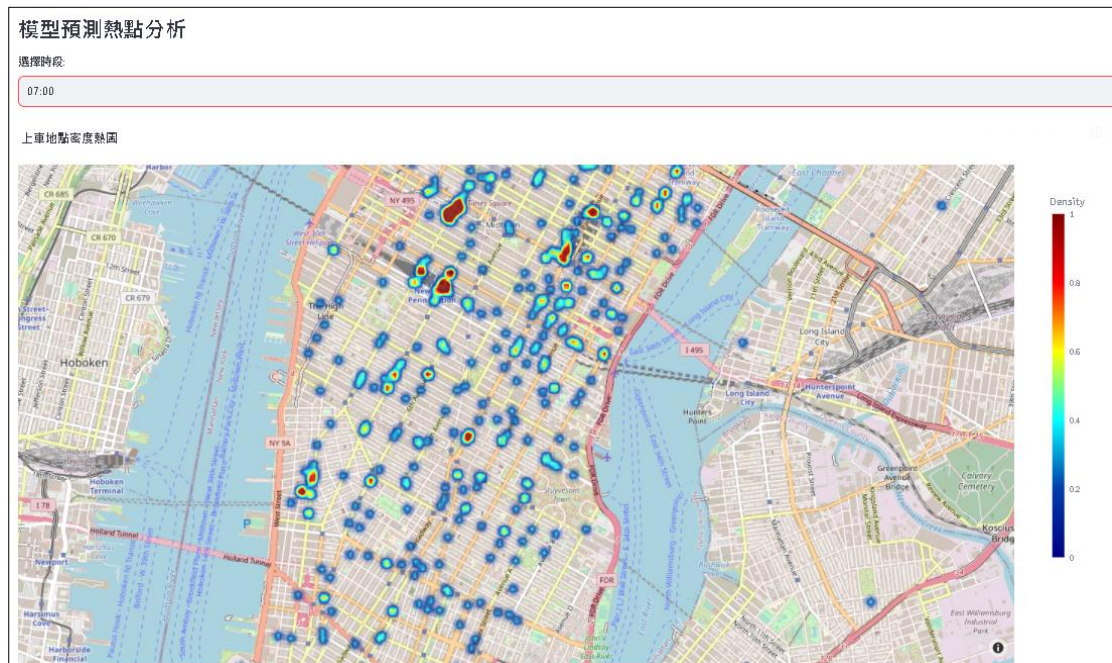


圖 4-27 Streamlit 模擬預測熱點分析圖

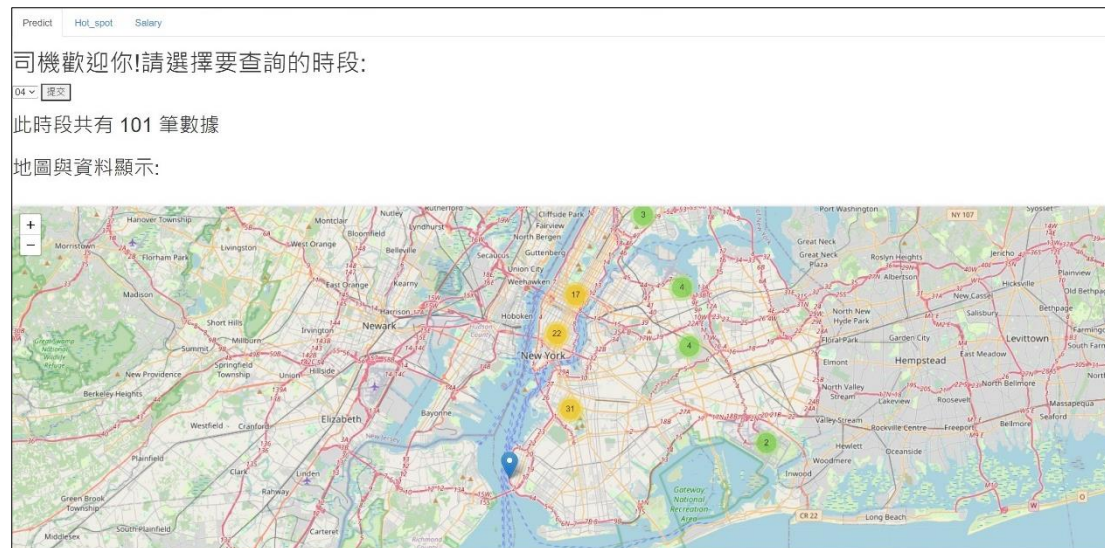


圖 4-28 Flask 模擬預測熱點分析圖

第五章 結論

使用 Streamlit 和基於 Flask 的應用程式的好處在於提供了一個直觀且互動性高的平台，讓使用者能夠深入了解紐約市計程車相關資訊，並進行自定義的資料查詢和視覺化分析。以下是使用這兩種工具的好處：

1. 直觀且互動性高的介面：

Streamlit 提供了一個簡單而強大的框架，讓開發者能夠輕鬆建立互動性高的 Web 應用程式。使用者可以透過選擇日期範圍、區域和社區等條件，即時查看計程車收入的相關資訊。這種互動性的介面使用者能夠更輕鬆地探索和理解資料。

2. 多功能的資料查詢和視覺化分析：

基於 Flask 的應用程式提供了計程車費用計算、油價資訊爬取、薪水計算和熱點分析等多功能，使得使用者可以根據個人需求進行更深入的資料查詢。這樣的機能性結合了不同方面的計程車相關資訊，讓用戶能夠從多個角度進行分析。

最後我們結合了 Streamlit 的互動式介面和基於 Flask 的多功能應用程式，專注於深入研究紐約市計程車相關資訊。這個平台不僅能夠提供詳盡的計程車收入趨勢分析，包括平均收入和總收入的展示，還能讓用戶透過選擇日期範圍、區域和社區來自訂其感興趣的資料視圖。同時，我們也整合了美化視覺化效果、提供更詳細的資料細節，以及增加更多過濾選項等改進，以進一步提升用戶體驗。透過持續聆聽用戶回饋，我們確保平台不斷演進和改進，滿足用戶不斷變化的需求，希望最後能成為他們在探索和理解紐約市計程車相關資訊時的首選工具。

參考文獻

- [1] Martin Ester、Hans-Peter Kriegel、Jiirg Sander、Xiaowei Xu(1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Institute for Computer Science, University of Munich Oettingenstr. 67, D-80538 Miinchen, Germany.
- [2] 馮琦森(2017)。基於計程車軌跡的居民出行熱點路徑和區域挖掘中國優秀碩士學位論文全文資料庫資訊科技輯。
- [3] 李文傑、閔世強、蔣瑩、張松芝、王成良(2019)。自我調整確定 DBSCAN 演算法參數的演算法研究。計算機工程與應用 2019 年第 5 期。
- [4] 郭哲倫(2019)。最佳化計程車司機總收入之車聯網單一終點共乘機制排程探討與系統實作。國立聯合大學電機資訊學院物聯網產業碩士專班論文。
- [5] 吳俊廷(2021)。整合式交通共用系統規劃與設計。國立屏東大學資訊科學系碩士論文。
- [6] 劉琳(2020)。基於出租車軌跡數據的載客高峰時段和熱點。河北工程大學碩士論文。
- [7] 許龍哲(2020)。以計程車大數據為基礎之載客熱點預測平臺之建置。亞洲大學資訊工程碩士論文。
- [8] 王貝貝(2018)。基於北京市載客熱點區的出租車出行需求研究。電腦應用與軟件第 40 卷第 10 期。
- [9] 林娜、鄭亞男(2016)。基於出租車軌跡數據的路徑規劃方法。電腦應用與軟件第 33 卷第 1 期。