元 智 大 學

資訊工程學系

專題製作成果報告

基於開放資料及社群評論分析之旅遊推薦系統

專題生：莊子毅、楊宗霖

學號：1051524、1051535

指導教授：周志岳

中華民國一○八年十一月

目錄

1. 前言 3
2. 問題描述 4
3. 文獻回顧與探討5
4. 研究方法9

(一)方法架構9

(二)使用流程描述11

(三)程式流程描述19

1. 結果與討論30

(一)研究成果與心得31

(二)創意應用競賽—資料創樂桃桃31

1. 參考文獻32
2. **前言**

現今社會中，人們往往被忙碌的生活步調壓的喘不過氣，會想些方法來舒緩過度緊繃且疲憊的自己。而旅遊，是一種暫時逃離壓力的方式。當你到了一個從沒去過的地方、體驗從沒體驗過的事，一切壓力煙消雲散。旅遊已成為現代人生活中不可或缺的一環了。但在過程中，難免會遇到一些突發狀況，例如到達目的地後才發現天氣狀況不佳、人潮擁擠或是路線上有交通狀況，甚至是踩雷發現根本不好玩。種種原因讓人們不得不敗興而歸。這時就很需要在出門前先進行資訊查詢。例如天氣狀況、人潮、交通、觀光地點的評論評分等等，來進行出遊的地點選擇及評估，完善自己的出遊計畫。

1. **問題描述**

使用者決定好景點後，首先最重要的，要先知道那個地方好不好玩、遊客數量如何；再來是交通工具選擇，公車、計程車、腳踏車或是自駕？選擇自駕則又得考慮是否有停車位、車位多不多的問題；最後，當地天氣如何，是否有下雨或是紫外線過高等等。不過，要一項項查詢實在是花費太多心力與時間，因此我們透過一個「整合所有資訊」的網站去提前知道、了解並且避免掉這些外在因素，進而提升旅遊品質。其中包含：

1. 旅遊網站的網友評分

讓使用者來判斷景點的好玩性和人氣，藉此知道此地是否符合自己的期待。

1. 市政府提供景點開放資料

每天皆會更新100個景點資訊，包含地址、電話、景點描述、營業時間、停車資訊等等，讓使用者找到理想的景點。

1. 自動天氣查詢

使用者決定景點後，根據市政府開放資料所提供座標位置，準確的查詢當地的天氣，讓使用者還未出發前便可了解天氣狀況。

1. **文獻回顧與探討**

關於開放資料的說明，以下分為(1)介紹和(2)優點兩個部分：

1. 介紹

此篇文獻作者將開放資料定義為不受公共資源限制的非機密數據(Janssen, Charalabidis & Zuiderwijk ,2012)，這些數據是用公共資金所產生的，且在使用方面不受任何限制。不包括私人和機密數據，因為此類數據不適合公開發布。

在各領域中公共機構，例如政府，往往是最大的數據創建和收集者。這些數據的範圍從交通、天氣、地理、旅遊訊息、統計數據、業務、公共部門預算和績效水平到各種有關政策和檢查的數據（食品，安全性，教育質量等）都有涵蓋。在此系統中所使用到的開放資料有:

1. 天氣

大桃園的平均天氣及分區詳細天氣:溫度、降雨、紫外線。

1. 旅遊訊息

桃園地區景點的基本介紹、聯絡方式、停車資訊、營業時間。

1. 社群評論統計

在社群上網友對於桃園地區景點的旅遊評論及評分統計。

1. 優點

開放資料本身比銷售數據集創造並產生更多的價值。收益集中在：

1. 政治和社會收益
2. 經濟收益
3. 經營和技術收益

可看到其涵蓋範圍之廣且密不可分，數據的開放有望帶來刺激創新和促進經濟增長的好處。各種公共組織都面臨發布原始數據的壓力。主要原因是因為獲取公共資助資料可以從公共投資中獲得更大的回報進而產出商機財富。

一位受訪者評論說：如果將數據用於政策制定，則對數據和解釋的準確性具有高度的信心。因此，決策者應準備共享他們的數據。通過打開數據，用戶可以驗證和驗證從數據中得出的結論是否正確和合理，並且他們可以分析先前收集的數據以加強決策重點，並且可以提高透明度和知識(Janssen, Charalabidis & Zuiderwijk, 2012)。

在實作此系統時感受到了資料的開放便利性，想要甚麼開放資料大部分都能在網路上取得，並且有即時性的資料，例如天氣，也不用擔心資料的時效性。

而關於旅遊推薦系統的部分，我們從兩個方向來探討，其分別為：

1. 探討過去各旅遊推薦系統的功能與特色
2. 現今旅遊推薦系統所使用的介面與技術

因為網路的無遠弗屆，一個指令就可搜尋到許多內容，也漸漸的造就出整合系統的存在，為了讓使用者更方便存取內容。不過現在網路上的資源過多，對於使用者來說是為複雜且耗時的。

目前生活上已出現很多不同種類、功能的旅遊推薦系統。因此，接下來要探討的是各旅遊系統所提供的功能(Borràs, Moreno & Valls, 2014)。

而以介面來看，大多數的旅遊推薦系統為了要充分呈現資訊，皆是以web的形式在電腦上供瀏覽。但隨著智慧手機越來越普及，做成手機app的趨勢也越來越高。也有些推薦系統是以軟體的方式供人下載至電腦，它的好處是能夠提供更多資訊且不須連上網路(產品更新除外)，但這方式卻沒被廣泛接受，原因是旅客通常手邊不會有可用的電腦，而且對他們來說是複雜且麻煩的。而本系統介面是基於web的形式，能夠及時抓取在網路上的資料(參考第二部分)。

在系統功能的部分，大致上能夠分成四類：

1. 目的地建議

根據遊客所提出的喜好或想法來建議。

1. 特定地點的景點推薦

當遊客到某一地點後，提供可能的景點。

1. 行程規劃

幫助遊客規劃路線、餐廳、飯店等等。

1. 社群媒體

根據網路社交、社群、論壇的評論、評分來做出判斷。

在此系統中，使用到了(2)及(4)，以下簡述其大概的功能：

1. 特定地點的景點推薦

特定地點我們設定在整個桃園市，並且提供100個景點的資訊。

1. 社群媒體

分析且使用旅遊網站上網友對景點的評分。

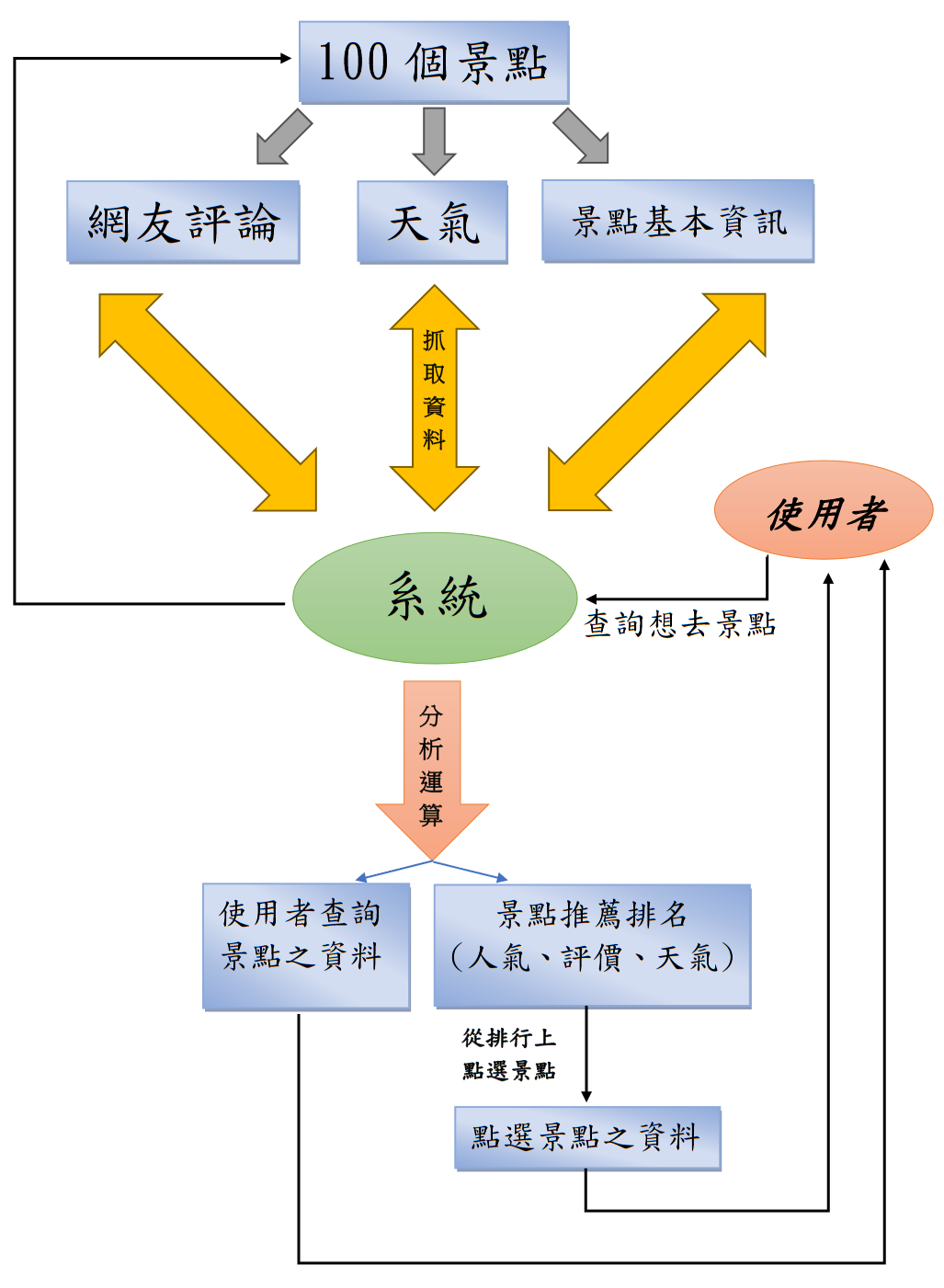
1. **研究方法**
2. **方法架構**

圖1，系統方法架構流程圖

主要架構是利用C#及HTML編寫前端介面，後端搭配Python來抓取開放資料，再將開放資料搭配數種演算法進行分析運算，並且提供讓使用者輸入想去的地方，查詢出該地的各項資訊。系統主要特點如下：

1. 每天皆會更新100個市政府提供的桃園景點資訊。
2. 主要抓取的詳細資料有:天氣(溫度、降雨機率、紫外線)，景點基本資訊(景點描述、地址、聯絡方式、營業時間、停車場資訊)，網友評論、評分及評價。
3. 讓使用者查詢想去景點(提供關鍵字查詢)，系統會提供景點之各項資訊。
4. 將100個景點資料進行分析運算後提供景點排行推薦。
5. **使用流程描述**

**USER**

**輸入地點**

**想去「石門水庫」走走**

**石門水庫**

****圖2，系統使用簡易示意圖

**提供資訊**

C#

1. 首先來到首頁，使用者可以在搜尋框內搜尋景點名稱，系統有提供關鍵字建議搜尋，會將名稱內含有搜尋關鍵字的景點列出來。



圖3，首頁搜尋畫面

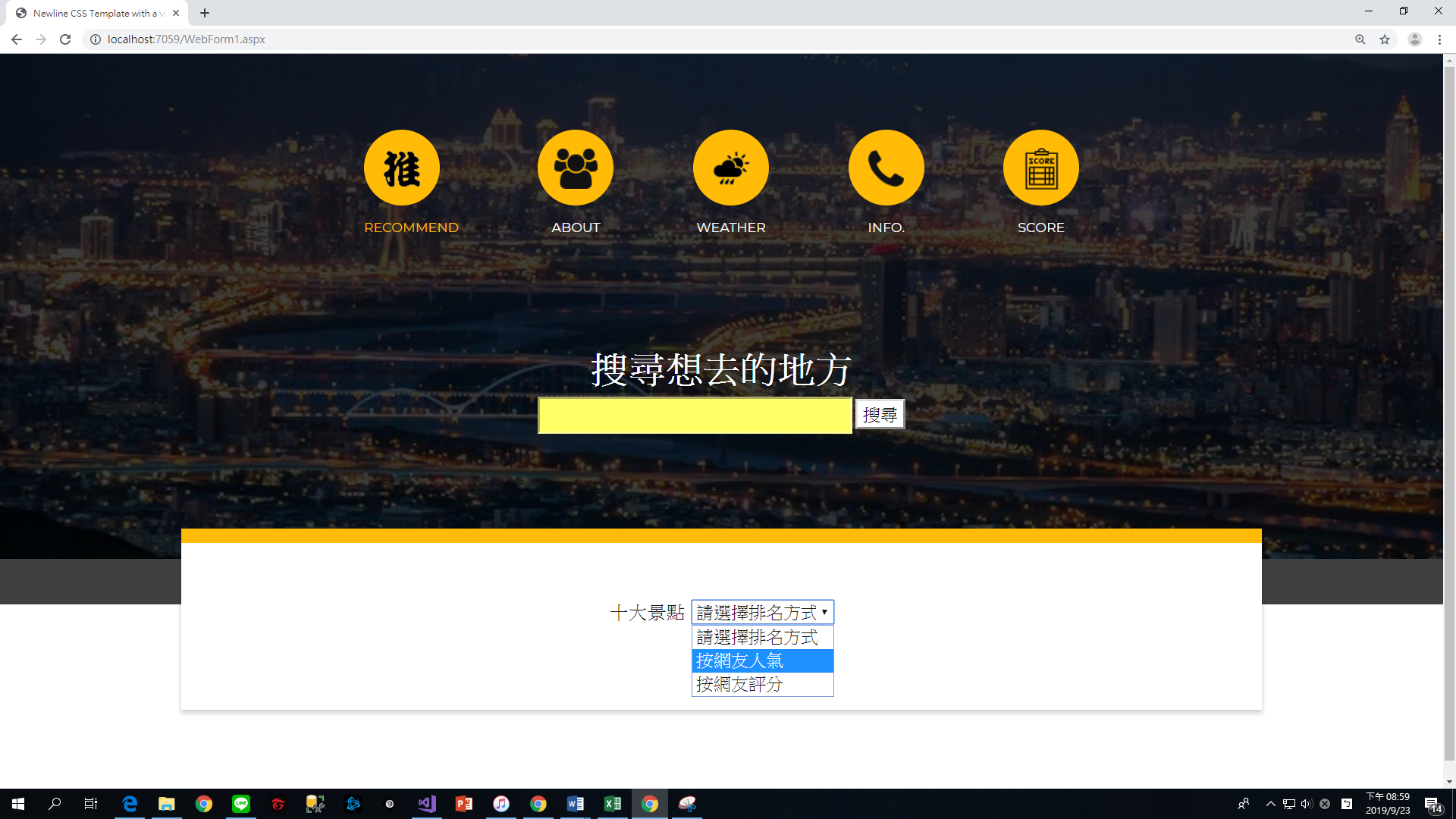
另外，根據綜合評分(詳細請參考程式流程描述)，有提供五個當下最適合出遊的景點。(圖4，圖5)

圖4，當前五大景點

|  |  |
| --- | --- |
|  |  |
| 平均氣溫為26度，晴天 | 平均氣溫為20度，細雨 |

圖5，不同氣溫下的分數變化

1. 首頁下半部(圖6)為今日十大排行景點區，使用者點開下拉式選單可選擇排行依據:網友人氣or網友評分。

圖6，景點排名分類選擇

1. 選擇完排行類型後會列出十大排行表，圖7是人氣排行，同樣也會列出景點的評分，左邊查詢按鍵點選後將會列出景點各項資訊。



圖7，景點十大人氣排行

圖8是評分排行，同樣也會列出人氣值，也有附查詢按鍵。

圖8，景點十大評分排行

1. 當使用者使用搜尋功能或是點選排行表上查詢按鈕後，系統會列出該景點各項資訊，只要透過網頁上端5個類別按鈕來進行資訊切換即可，從左至右是：首頁推薦及搜尋、景點介紹、天氣狀況、相關資訊、旅遊指數點選  即可看到景點介紹(圖9)。



圖9，景點介紹

1. 點選  後會列出景點地區之天氣狀況(圖10)，下圖是當下的天氣：有溫度、天氣概況描述、紫外線。



圖10點簡略天氣狀況

下半部則是有時刻天氣狀況包含溫度、降雨機率、濕度、風力，以及整個桃園市的平均天氣資訊(如圖11)。



圖11，景點詳細天氣狀況

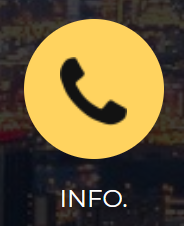
1. 點進  後會有景點的相關資訊，一些基本資料如：地址、停車資訊、連絡電話、營業時間(圖12)。



圖12，景點相關資訊

1. 最後點進  裡則是有該景點的旅遊指數，裡面會將各大項指數依據展示出來，如天氣、人氣(圖14)、評分(圖15)，最後再透過演算法整合出綜合旅遊指數(圖13)，讓使用者可以判斷適不適合出遊。



圖13，旅遊指數呈現頁面

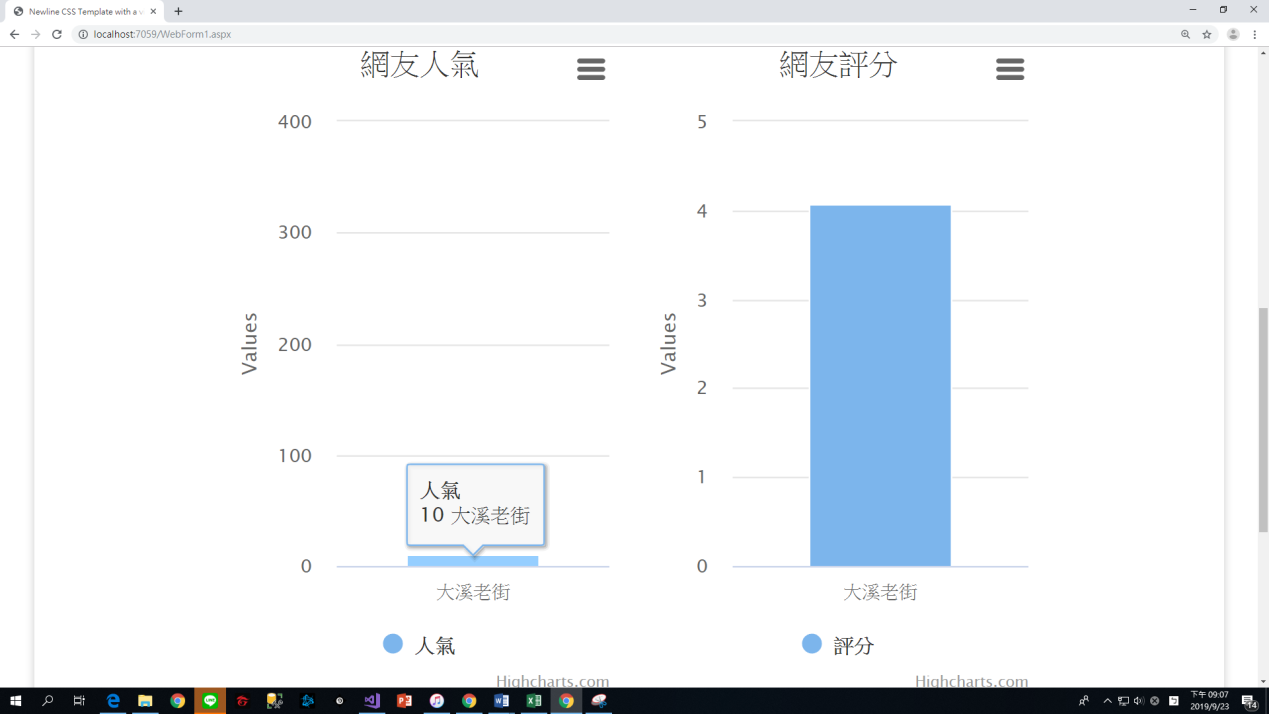
 

圖14，景點人氣長條圖 圖15，景點評分長條圖

1. 結束之後只要點選上面最左邊  即可回到首頁再次進行搜尋或是觀看排行。
2. **程式流程描述**
3. 抓取桃園開放資料(<https://data.tycg.gov.tw/>)提供的桃園市景點資料(https://data.tycg.gov.tw/api/v1/rest/datastore/bd906b29-9006-40ed-8bd7-67597c2577fc?format=json)(如圖16)



圖16，桃園市開放景點資料(部分)

其中包含100個桃園市景點的名稱、簡述、地址、停車資訊、電

話、開放時間、X座標(經度)和Y座標(緯度)，並存成excel檔(如圖17)，方便存取。

圖17，桃園市開放景點資料(只取20個)

編寫的程式碼如下：

1. url = "https://data.tycg.gov.tw/api/v1/rest/datastore/bd906b29-9006-40ed-8bd7-67597c2577fc?format=json"
2. r = requests.get(url)
3. data = json.loads(r.text)
5. Name\_list = []
6. Toldescribe\_list = []
7. Add\_list = []
8. Parkinginfo\_list = []
9. Tel\_list = []
10. Opentime\_list = []
11. Px\_list = []
12. Py\_list = []
14. f = codecs.open("./File/Name.txt", 'w', 'utf-8')
15. **for** i **in** range(100):
16. newname = data['result']['records'][i]['Name']
17. Name\_list.append(newname)
18. Toldescribe\_list.append(data['result']['records'][i]['Toldescribe'])
19. Add\_list.append(data['result']['records'][i]['Add'])
20. Parkinginfo\_list.append(data['result']['records'][i]['Parkinginfo'])
21. Tel\_list.append(data['result']['records'][i]['Tel'])
22. Opentime\_list.append(data['result']['records'][i]['Opentime'])
23. Px\_list.append(data['result']['records'][i]['Px'])
24. Py\_list.append(data['result']['records'][i]['Py'])
25. f.write(newname)
26. f.write("\r\n")
27. f.close()
28. 從貓途鷹(<https://www.tripadvisor.com.tw/>)搜尋所抓取的100個景點的旅客評等(如圖18)。

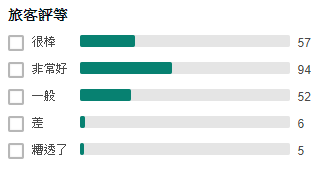


圖18，貓途鷹網站旅客評等圖

此處以大溪老街為例。根據旅客評等，計算出「人氣」和「評分」 兩種結果：

1. 人氣

很棒人數和非常好人數之和，舉例來說，57加上94為151，則代表大溪老街人氣數值為151。

1. 評分

依序設定很棒到糟透了的加權分數為5到1，再把各人數乘上加權分數後做總和，最後再除以總人數，舉例來說：

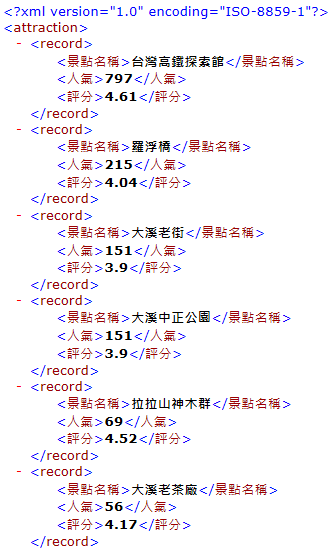
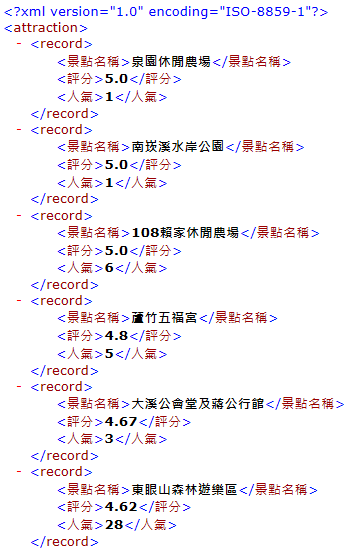
57\*5+94\*4+52\*3+6\*2+5\*1=834

57+94+52+6+5=214

834/214=3.897..，則評分為3.89。

編寫的程式碼如下：

1. # 存入excel的list
2. attraction\_name\_list = []
3. hot\_list=[]
4. score\_list=[]
6. **for** index **in** range(100):
7. # Google 搜尋 URL
8. google\_url = 'https://www.google.com.tw/search'
10. # 查詢參數
11. inputname = data['result']['records'][index]['Name']
12. j = inputname + ' Attraction\_Review'
13. my\_params = {'q': j}
14. r = requests.get(google\_url, params = my\_params)
16. # 確認是否下載成功
17. **if** r.status\_code == requests.codes.ok:
18. soup = BeautifulSoup(r.text, 'html.parser')
19. items = soup.select('a[href^="/url"]')
20. attraction\_url = ''
21. **for** i **in** items:
22. # 在tripadvisor網頁中搜尋景點資訊
23. **if**(i.get('href')[0:55]=='/url?q=https://www.tripadvisor.com.tw/Attraction\_Review'):
24. attraction\_url = i.get('href')[7:]
26. **if** len(attraction\_url)==0:
27. **continue**
28. res = requests.get(attraction\_url)
29. html\_doc = res.text
31. # 以 Beautiful Soup 解析 HTML 程式碼
32. soup = BeautifulSoup(html\_doc, 'html.parser')
34. label\_tags = soup.find\_all('label',class\_="row\_label label",limit=5)
35. span\_tags = soup.find\_all('span',class\_="row\_num is-shown-at-tablet")
36. labellist=[]
37. spanlist=[]
38. **for** label\_tag **in** label\_tags:
39. labellist.append(label\_tag.string)
40. **for** span\_tag **in** span\_tags:
41. spanlist.append(span\_tag.string)
42. dictionary = dict(zip(labellist, spanlist))
44. **if** bool(dictionary) == False:
45. **pass**
46. **else**:
47. # 將各級別評分擷取下來
48. **print**(index,"/100")
49. great\_count = int(dictionary.get("很棒"))
50. verygood\_count = int(dictionary.get("非常好"))
51. good\_count = int(dictionary.get("一般"))
52. poor\_count = int(dictionary.get("差"))
53. terrible\_count = int(dictionary.get("糟透了"))
55. hot = great\_count+verygood\_count
57. # 算出評分
58. score = (great\_count\*5 + verygood\_count\*4 + good\_count\*3+poor\_count\*2+terrible\_count\*1)/(great\_count+verygood\_count+good\_count+poor\_count+terrible\_count)
60. attraction\_name\_list.append(inputname)
61. hot\_list.append(hot)
62. score\_list.append(round(score,2))
64. # 匯出成excel檔
65. data={'景點名稱':attraction\_name\_list,'人氣':hot\_list,'評分':score\_list}
66. df=DataFrame(data)
67. df.to\_excel('排名.xlsx')
68. 將100個景點的「人氣」和「評分」按照大小排序，各取前十名，並存成xml，方便存取。(如圖19、圖20)

 圖19，人氣排名(只擷取部分) 圖20，評分排名(只擷取部分)

編寫的程式碼如下：

1. read = pd.read\_excel('排名.xlsx',index\_col=0)
2. root1 = etree.Element('attraction') #建立xml
3. root2 = etree.Element('attraction') #建立xml
5. resort\_hot = read.sort\_values(by=['人氣'], ascending=False)  #按照人氣度排名
6. **for** i **in** range(10):
7. record1 = etree.SubElement(root1, 'record')
8. name1 = etree.SubElement(record1, '景點名稱')
9. hot = etree.SubElement(record1, '人氣')
10. mainhot\_score = etree.SubElement(record1, '評分')
11. name1.text = resort\_hot.iloc[i,0]
12. hot.text = str(resort\_hot.iloc[i,1])
13. mainhot\_score.text = str(resort\_hot.iloc[i,2])
14. s1 = etree.ElementTree(root1)
15. s1.write('hotsort.xml', pretty\_print=True, xml\_declaration=True, encoding='utf-8')
17. resort\_score = read.sort\_values(by=['評分'], ascending=False)  #按照評分排名
19. **for** j **in** range(10):
20. record2 = etree.SubElement(root2, 'record')
21. name2 = etree.SubElement(record2, '景點名稱')
22. score = etree.SubElement(record2, '評分')
23. mainscore\_hot = etree.SubElement(record2, '人氣')
24. name2.text = resort\_score.iloc[j, 0]
25. score.text = str(resort\_score.iloc[j, 2])
26. mainscore\_hot.text = str(resort\_score.iloc[j, 1])
27. s2 = etree.ElementTree(root2)
28. s2.write('scoresort.xml', pretty\_print=True, xml\_declaration=True, encoding='utf-8')
29. 使用者輸入欲查詢的景點後，系統會顯示此景點的多項資料，包括簡述、地址、停車資訊、電話、開放時間和即時天氣。

即時天氣的運作方式說明如下：

1. 將開放資料得到的X座標(經度)和Y座標(緯度)代入

https://weather.com/zh-TW/weather/today/1/Y座標,X座標?par=apple\_widget&locale=zh\_TW，會得到目前時間的當地天氣資訊(如圖21)

1. 將開放資料得到的X座標(經度)和Y座標(緯度)代入

https://weather.com/zh-TW/weather/hourbyhour/l/Y座標,X座標?par=apple\_widget&locale=zh\_TW，會得到目前時間後若干小時的當地天氣資訊(如圖22)

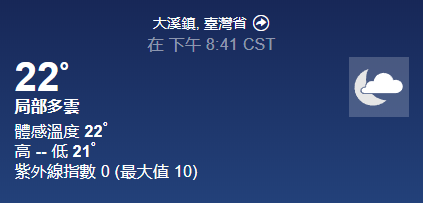


圖21，當地天氣資訊(此處以大溪老街為例)



圖22，當地天氣詳細資訊(此處以大溪老街為例)

1. 將(1)、(2)結果抓取下來並顯示在系統上。

編寫的程式碼如下：

1. **def** now\_weather\_search(X, Y):
2. # 將座標帶入網址，即可查詢當地天氣資訊
3. now\_weather\_url = "https://weather.com/zh-TW/weather/today/l/" + Y + "," + X + "?par=apple\_widget&locale=zh\_TW"
4. # print(now\_weather\_url)
5. now\_weather\_res = requests.get(now\_weather\_url)
6. now\_weather\_html\_doc = now\_weather\_res.text
8. # 以 Beautiful Soup 解析 HTML 程式碼
9. now\_weather\_f = codecs.open('./File/nowweather.txt', 'w', 'utf-8')
10. soup = BeautifulSoup(now\_weather\_html\_doc, 'html.parser')
11. div\_tags = soup.find('div', class\_="today\_nowcard-temp")
12. span\_tags = div\_tags.find('span', class\_="")
13. **for** i **in** span\_tags:
14. # 現在溫度
15. # print(int(i))
16. now\_weather\_f.writelines(i.string)
17. now\_weather\_f.write("\r\n")
18. **break**
20. div\_tags = soup.find('div', class\_="today\_nowcard-phrase")
21. **for** j **in** div\_tags:
22. # 現在天氣狀態(中文敘述)
23. # print(j.string)
24. now\_weather\_f.writelines(j.string)
25. now\_weather\_f.write("\r\n")
27. div\_tags = soup.find('div', class\_="today\_nowcard-hilo")
28. div2\_tags = div\_tags.find('div')
29. span\_tags = div2\_tags.find\_all('span', class\_="")
30. **for** k **in** span\_tags:
31. # 現在紫外線
32. # print(k.string)
33. now\_weather\_f.writelines(k.string)
34. now\_weather\_f.write("\r\n")
35. now\_weather\_f.close()
37. **def** hourbyhour(X,Y):
38. res = requests.get("https://weather.com/zh-TW/weather/hourbyhour/l/" + Y + "," + X + "?par=apple\_TWC&locale=zh\_TW")
39. html\_doc = res.text
40. soup = BeautifulSoup(html\_doc, 'html.parser')
41. table = soup.find('table',class\_='twc-table')
42. columns = [th.text.replace('\n', '') **for** th **in** table.find('tr').find\_all('th')]
44. columns[4] = "降雨機率"
45. trs = table.find\_all('tr')[1:]
46. rows = list()
47. **for** tr **in** trs:
48. rows.append([td.text.replace('\n', '').replace('\xa0', '') **for** td **in** tr.find\_all('td')[1:]])
50. df = pd.DataFrame(data=rows, columns=columns)
51. # print(df.head())
52. df.to\_excel('weather.xlsx')
53. create\_weather\_xml()
54. 計算綜合旅遊指數：

首先，將「人氣」對照以下表格後換算成分數：

|  |  |
| --- | --- |
| 人氣 | 人氣換算後分數 |
| 60以上 | 100 |
| 50-59 | 90 |
| 40-49 | 80 |
| 30-39 | 70 |
| 20-29 | 60 |
| 10-19 | 50 |
| 0-9 | 40 |

並且帶入公式：

，藉以算出綜合旅遊指數。

以大溪老街為例，大溪老街人氣為151，對照表格換算後為100、大溪老街評分為3.89、氣溫為26度。

將上述數值帶入公式：

即可得出綜合旅遊指數為71.05。

1. **結果與討論**
2. **研究成果與心得**

此系統在後端使用了python語言來編寫，其提供了許多套件來完成爬蟲技術上的實作(包括requests、beautifulsoup4)抓取開放資料以及社群上的評論，並加以分析和計算，達到「群眾外包」的概念。「群眾外包」是指一套系統或服務中，利用大量的網路用戶的想法來取得資源，並將此統整、分析，最後產生出適合使用者習慣、強調客觀性的系統。推薦系統則是根據旅遊網站上的網友評分及評論，經過多重比對後給予適當權重及算法，算出理想的數據，而此結果（整體排名）也比較有合理性。

為了讓使用者得到更準確的天氣結果，因此在計算天氣時選擇使用經緯度來當作天氣資訊的位置，而非是大範圍(例如桃園市八德區)，與以往的天氣預報做出差別。此舉能夠將位置鎖定在特定景點上，因此天氣狀況會更加準確。

使用者能自行查詢桃園景點並且獲得各式資訊。在首頁上也有人氣排名、評分排名、綜合排名的推薦，推薦排名也會附上天氣狀況，讓喜好不同天氣的使用者能夠自行決斷是否接受，一切以客觀推薦為主，主要決定權還是落在使用者身上。

目前景點範圍定在桃園市內，未來不排除也可以將範圍提升至北部、全台灣甚至多數國家。

1. **創意應用競賽—資料創意樂桃桃**

此系統參加過由桃園市政府資訊科技局舉辦的第2屆創意應用競賽—資料創意樂桃桃，運用政府提供以及自行抓取的開放資料來進行創意實作系統。

1. 競賽宗旨

運用「資料」解決「桃園在地政府施政與民眾生活圈相關議題」，而藉此競賽使用開放資料發揮各行業專業精神、無限創意及構思設計，提出建議或解決方案以提升公共利益或經濟價值。

1. 反思

當時我們對於爬蟲、數據分析和建置互動式介面的技術等等還略為生疏，因此成果仍然是屬於比較陽春、簡單的。而此版本則是從當時的版本，加入更多新的概念及想法並且實作出來，功能更加完善、齊全。

**六、參考文獻**

1. Janssen, M., Charalabidis, Y., & Zuiderwijk, A. (2012). Benefits, adoption barriers and myths of open data and open government. *Information systems management*, *29*(4), 258-268.
2. Borràs, J., Moreno, A., & Valls, A. (2014). Intelligent tourism recommender systems: A survey. *Expert Systems with Applications*, *41*(16), 7370-7389.
3. 桃園開放資料。檢自<https://od.tycg.gov.tw/creative#sectionB>(Nov.4,2019)