

網路與資料庫期末專題

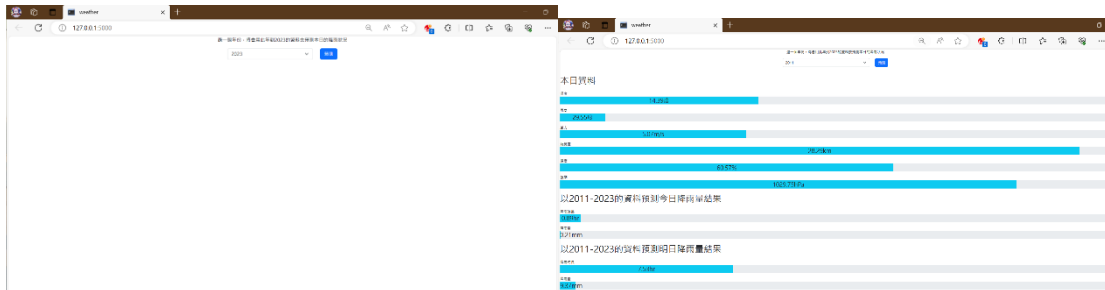
110502516 許尚軒

一、 題目發想:

有一次的作業是以決策樹來分析天氣的狀況，可是因為決策樹的輸入只能是少數離散化的資料，且不是真實資料，加上我家在基隆，是個常常下雨的城市，我就想去氣象局找過去的真實資料，使用 knn 來預測今日的降雨狀況，來看看今天這種天氣在過去的降雨表現。

二、 功能說明

左圖是起始畫面，可以選擇要用來預測的年份，按下按鈕後等待一段時間會出現右圖，包括了今日的天氣資料以及分析的降雨結果。



三、 實作技術

使用了 flask 來寫網站，在收到按下預測按鈕傳來的 post 請求後，會用 predict.py 的 pre()函式來預測，再將結果傳回前端。

```
6 @app.route('/', methods=['GET', 'POST'])
7 def pred():
8     if request.method == 'POST':
9         y = request.values['sel']
10
11         now, p = pre(int(y))
12
13         data = {'now': now, 'p': p, 'y': int(y)}
14         return jsonify(data)
15     else:
16         return render_template('main.html')
```

在 `pre()` 函式中，會去資料庫抓取需要的資料，然後將那些資料標準化後丟給 `knn`。

(補充 1: 上述提到的資料庫是將位於氣象局網站中分散的資料一個一個下載下來後用 `python` 存入資料庫，`data.py` 就是負責此工作)

(補充 2: `n_neighbor` 設為 3*用來預測的年份數量)

```
24     scalerx = StandardScaler()
25     X = scalerx.fit_transform(X)
26     scalery = StandardScaler()
27     y = scalery.fit_transform(y)
```

之後再利用爬蟲去氣象局網站將今日天氣抓取下來，因為氣象局網站中的天氣資料是在進入網站時才會用 `js` 產生的，所以只用 `bs4` 抓取不到資料，就使用了 `selenium` 來模擬實際造訪網站情形，再用 `bs4` 分析獲得的 `html`，最後再將爬到的資料標準化後就可以拿去預測了。

```
92     new_data_point = np.array(now_data).mean(axis=0)
93     print(new_data_point)
94     new_data_point_sca = scalerx.transform(new_data_point.reshape(1, 7)).reshape(7)
95     k = 3*request_year
```

因為氣象局網站上的資料是過去 24 小時的，因此 `knn` 分析出來的結果只能表示過去 24 小時的降雨預測，如果要取得未來 24 小時的預測資料就得去算 `knn` 結果加上 1 天的平均。

```
102     nearest_neighbor_ids += request_year
103     nearest_neighbor_ids %= len(X)
```

最終再將原始資料與反初始化的預測結果回傳。

```
111     return list(new_data_point), list(scalery.inverse_transform(prediction.reshape(1, -1))[0]), \
112           list(scalery.inverse_transform(prediction2.reshape(1, -1))[0])
```

四、 自我評估

此專題使用到了之前教的的 `sqlite` 和 `flask` 以及這次的 `knn`, 爬蟲, 標準化等等的技術, 我覺得算是統整運用了蠻多上課所學的技術。只不過本次可以運用的技術偏少, 其實找不到任何一個適合的模型來分析天氣, 像是決策樹有離散問題, `gpt` 或分群也幫不上忙, 就連最適合的 `knn` 都有個問題是每個輸入的權重都相等, 導致預測出來的結果不慎精準, 只能代表在過去相似資料下的平均降雨情形。如果要延伸的話還可以結合其他資料視覺化的圖表分析那些過去資料, 或是使用其他更適合的機器學習模型, 我覺得都是可行的未來目標。

五、 檔案說明

執行方法: 執行 `main.py` 後進入 `127.0.0.1:5000`

`main.py`: 主程式, 用來架設 `flask` 網站

`templates/main.html`: `html` 檔

`predict.py`: 連接資料庫、以 `knn` 預測天氣、爬蟲抓取現在天氣

`data.db`: 儲存過去天氣資料

`data.py`: 將下載下來的 `csv` 檔存入 `data.db`

`csv`: 放下載下來的 `csv` 檔

`.env`: 需要用到的套件