中**華大學資訊工程學系**

**專題製作報告**

**大數據分析-空氣品質與癌症之間的關聯**

指導教授：歐陽雯

組員：陳楷勳 B10902202

中 華 民 國 114年 6 月 21 日

**目 錄**

**第一章 前言**

**1-1. 研究動機 ………………....................03**

**1-2. 研究目標 ………………....................03**

**第二章 系統流程與動作流程**

**2-1. 系統流程 ………………....................04**

**2-2. 流程說明 ………………....................05**

**第三章 成果**

**3-1. 成果 ………………........................13**

**第四章** **問題與討論**

**4-1. 問題 ………………….........................22**

**第五章 結論**

**5-1. 結語 ………………….........................23**

**5-2. 參考資料 ………………....................23**

**第一章 前言**

**1.1 研究動機**

空氣品質之評估，需要根據多項空氣污染物濃度來計算的，主要包括PM2.5、PM10、臭氧(O3)、二氧化硫(SO2)、二氧化氮(NO2)和一氧化碳(CO)，最終得到AQI，以此作為衡量空氣品質重要指標，AQI衡量方式並非為一線性公式，每個污染物的濃度可以用以下公式來計算如圖 1

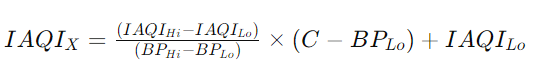


圖 1 子指標計算公式

其中IAQIX​ 是污染物X的IAQI，C 是污染物X的濃度，BPHi​ 是大於或等於C的最高濃度參數，BPLo​ 是小於或等於C的最低濃度參數，IAQIHi​ 和 IAQILo 分別是對應於 BPHi和 BPLo的IAQI，再來各種濃度算完取最大值如圖 2。

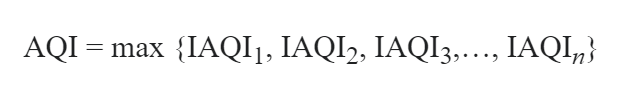


圖 2 各種濃度算完取最大值

最終可以得到一個對應的分數範圍，如圖 3。

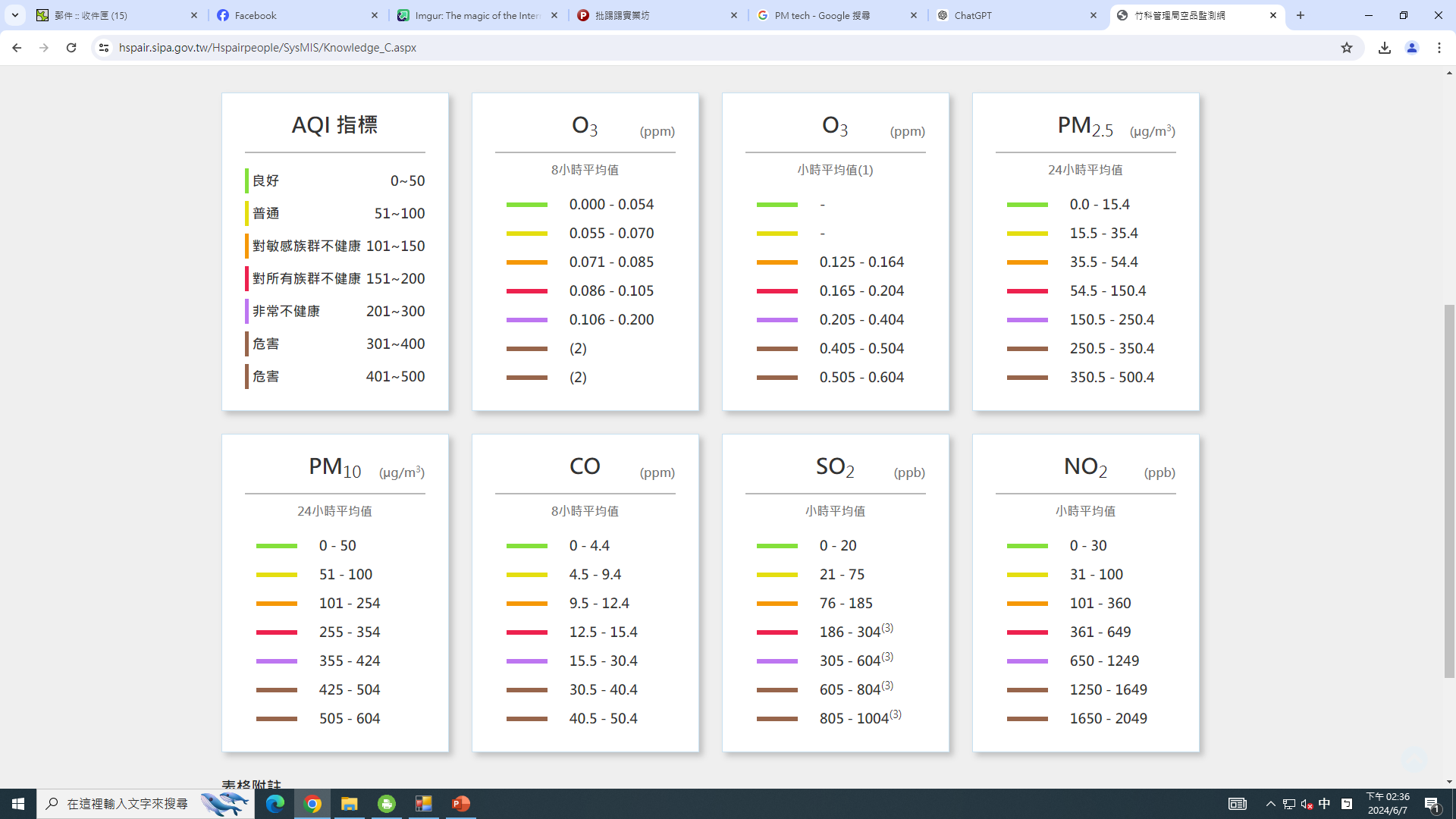


圖 3 各種汙染物對應分數

因此本研究希望直接用原始數據進行基礎統計以及建立預測模型，並將原始數據與肺癌人口做比較，嘗試尋求找到空氣品質數據與罹癌人口是否存在一定關聯性。

**1.2 研究目標**

本研究會將數據整理後，預計完成以下：

1. 做一系列可視化之圖表解釋可能之關聯。
2. 建立AQI預測模型。
3. 運用地圖資訊及罹癌人數繪製分布圖。

**第二章 系統流程**

**2-2 流程說明\_爬蟲**

本專題所使用的資料為政府資料開放平臺中的資訊。

網址為https://data.moenv.gov.tw/api/v2/aqx\_p\_432?api\_key=ca6e7784-9cec-4631-b038-ce4ab6e075d1&limit=1000&sort=ImportDate%20desc&format=CSV，該資料每日抓取一份csv檔案，一份CSV大約含有85筆資料，日期區間為2023.9.12至2024.1.6

**2-2 流程說明\_資料分析**

收集數據

模型評估

資料清理及統計

分train/test

訓練模型/評估

**第三章 成果**

**3-1統計資料分析**

將AQI依照大小分出最大前五名縣市為左營、林園、仁武、琉球、大寮，最小前五名縣市為關山、台東、花蓮、冬山、宜蘭，挑選其中幾個資料作關聯係數矩陣。以下為程式碼

df=pd.read\_csv('/content/drive/MyDrive/ppt/air\_total.csv')#引入數據

st1='花蓮'#指定地地

df=df[df['sitename']==st1]#將該地點之數據篩出

binary\_columns = df[['so2','co', 'o3', 'o3\_8hr', 'pm10','pm2.5', 'no2', 'no']]

correlation\_matrix = binary\_columns.corr()#利用corr這個函式算出兩兩之間的

#關聯係數

sns.heatmap(correlation\_matrix, annot=True, cmap='Reds', linewidths=0.5)#繪製

#熱力圖如圖 5

由於AQI數據是每一天所有地區的值記錄在一份csv檔案，因此將資料合併後要使用groupby去算各個地區平均值，才能知道前五後五名之地區程式碼如下

add=glob.glob("/content/drive/MyDrive/ppt/去年資料/\*.csv")#記錄所有檔案位置

a1=pd.read\_csv(add[0],encoding='big5')

for i in add[1:]:

temp=pd.read\_csv(i,encoding='big5')

a1=pd.concat([a1,temp])#該for 迴圈將所有資料合併

a1.groupby(['sitename'])['aqi'].mean().sort\_values(ascending=False).to\_csv("aqi\_mean.csv",encoding='big5')#將不同地區分別算平均值

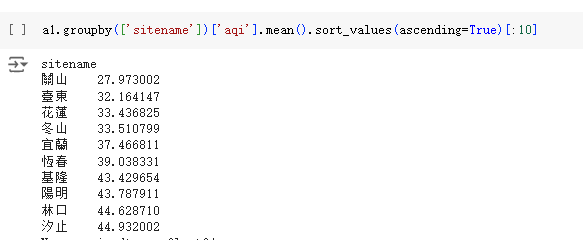
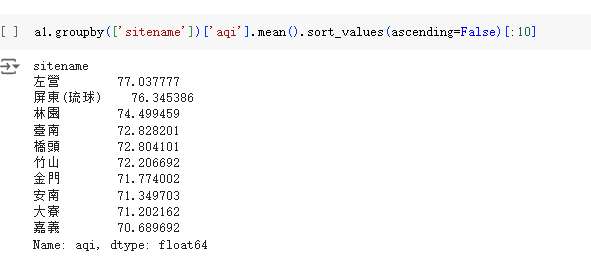


圖 4 用groupby篩選出最高五個及最低五個

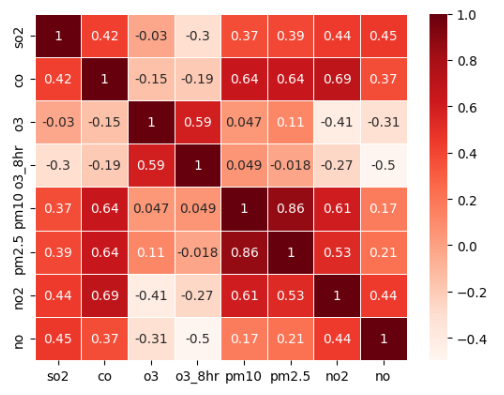
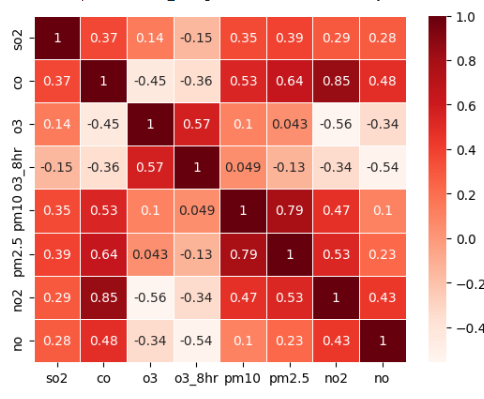


圖 5 左邊為左營右邊為林園

其中越深色表示越有關聯如圖 5，大部分情況關聯係數小於0.6，其中比較特殊的為PM2.5與PM10，關聯係數大約為0.8，推測可能是PM2.5和PM10都是空氣中的顆粒物，可能來自於類似的污染源，因此才會有同時升高減少的現象發生，另外是NO2與CO也是有比較高的相關性，由於這兩者是常見的汽機車廢氣，可能也是因為同為一個汙染源才導致相關係數較高。

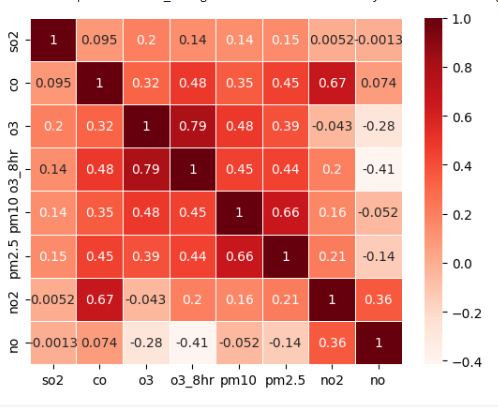
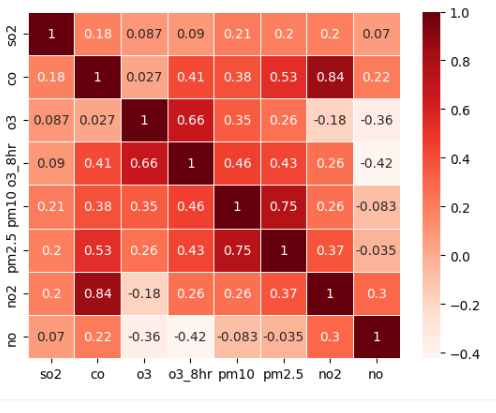


圖 6 左邊為花蓮右邊為台東

圖 6為AQI較小之區域，關聯係數與圖 5雷同。

以上數據是考慮單一地區全部時間所繪製的關聯係數矩陣，若是以日為單位，針對每一日AQI與各種汙染物(SO2 CO O3 PM10 PM2.5)算相關係數，再以日期做折線圖程式碼如下

def Pearson\_correlation(X,Y):

if len(X)==len(Y):

Sum\_xy = sum((X-X.mean())\*(Y-Y.mean()))

Sum\_x\_squared = sum((X-X.mean())\*\*2)

Sum\_y\_squared = sum((Y-Y.mean())\*\*2)

corr = Sum\_xy / np.sqrt(Sum\_x\_squared \* Sum\_y\_squared)

return corr

else:

return 0

a=a.dropna(subset=[ 'so2','co','o3','o3\_8hr', 'pm10', 'pm2.5', 'no2', 'nox', 'no'])

# nox pm2.5 pm10

a['datetime']=a['publishtime'].apply(pd.to\_datetime)

temp\_sitename=a[a['sitename']=='左營']

temp\_sitename=temp\_sitename.sort\_values(by=['datetime'])

R=[]

pollutant=[]

date\_list=[]

start=temp\_sitename.iloc[0,24]

while(start<temp\_sitename.iloc[-1,24]):

temp\_r=temp\_sitename[(temp\_sitename['datetime']>start)&(temp\_sitename['datetime']<start+datetime.timedelta(days=6))]

if not temp\_r.empty:

R.append(Pearson\_correlation(temp\_r['so2'],temp\_r['aqi']))

pollutant.append('so2')

date\_list.append(temp\_r.iloc[0,16])

start=start+datetime.timedelta(days=1)

else:

start=start+datetime.timedelta(days=1)

start=temp\_sitename.iloc[0,24]

while(start<temp\_sitename.iloc[-1,24]):

temp\_r=temp\_sitename[(temp\_sitename['datetime']>start)&(temp\_sitename['datetime']<start+datetime.timedelta(days=6))]

if not temp\_r.empty:

R.append(Pearson\_correlation(temp\_r['co'],temp\_r['aqi']))

# site.append(temp\_sitename.iloc[0,0])

pollutant.append('co')

date\_list.append(temp\_r.iloc[0,16])

start=start+datetime.timedelta(days=1)

else:

start=start+datetime.timedelta(days=1)

start=temp\_sitename.iloc[0,24]

while(start<temp\_sitename.iloc[-1,24]):

temp\_r=temp\_sitename[(temp\_sitename['datetime']>start)&(temp\_sitename['datetime']<start+datetime.timedelta(days=6))]

if not temp\_r.empty:

R.append(Pearson\_correlation(temp\_r['o3'],temp\_r['aqi']))

# site.append(temp\_sitename.iloc[0,0])

pollutant.append('o3')

date\_list.append(temp\_r.iloc[0,16])

start=start+datetime.timedelta(days=1)

else:

start=start+datetime.timedelta(days=1)

start=temp\_sitename.iloc[0,24]

while(start<temp\_sitename.iloc[-1,24]):

temp\_r=temp\_sitename[(temp\_sitename['datetime']>start)&(temp\_sitename['datetime']<start+datetime.timedelta(days=6))]

if not temp\_r.empty:

R.append(Pearson\_correlation(temp\_r['pm10'],temp\_r['aqi']))

# site.append(temp\_sitename.iloc[0,0])

pollutant.append('pm10')

date\_list.append(temp\_r.iloc[0,16])

start=start+datetime.timedelta(days=1)

else:

start=start+datetime.timedelta(days=1)

start=temp\_sitename.iloc[0,24]

while(start<temp\_sitename.iloc[-1,24]):

temp\_r=temp\_sitename[(temp\_sitename['datetime']>start)&(temp\_sitename['datetime']<start+datetime.timedelta(days=6))]

if not temp\_r.empty:

R.append(Pearson\_correlation(temp\_r['pm2.5'],temp\_r['aqi']))

# site.append(temp\_sitename.iloc[0,0])

pollutant.append('pm2.5')

date\_list.append(temp\_r.iloc[0,16])

start=start+datetime.timedelta(days=1)

else:

start=start+datetime.timedelta(days=1)

start=temp\_sitename.iloc[0,24]

while(start<temp\_sitename.iloc[-1,24]):

temp\_r=temp\_sitename[(temp\_sitename['datetime']>start)&(temp\_sitename['datetime']<start+datetime.timedelta(days=6))]

if not temp\_r.empty:

R.append(Pearson\_correlation(temp\_r['nox'],temp\_r['aqi']))

# site.append(temp\_sitename.iloc[0,0])

pollutant.append('nox')

date\_list.append(temp\_r.iloc[0,16])

start=start+datetime.timedelta(days=1)

else:

start=start+datetime.timedelta(days=1)

dataframe=pd.DataFrame({'p':pollutant,"date\_list":date\_list,'R':R})

# print(dataframe)

fig = px.line(dataframe, x="date\_list", y="R", color='p')

fig.show()

，以AQI依照大小分出最大前五名縣市為左營、林園、仁武、琉球、大寮，最小前五名縣市為關山、台東、花蓮、冬山、宜蘭作圖，可以得到如下

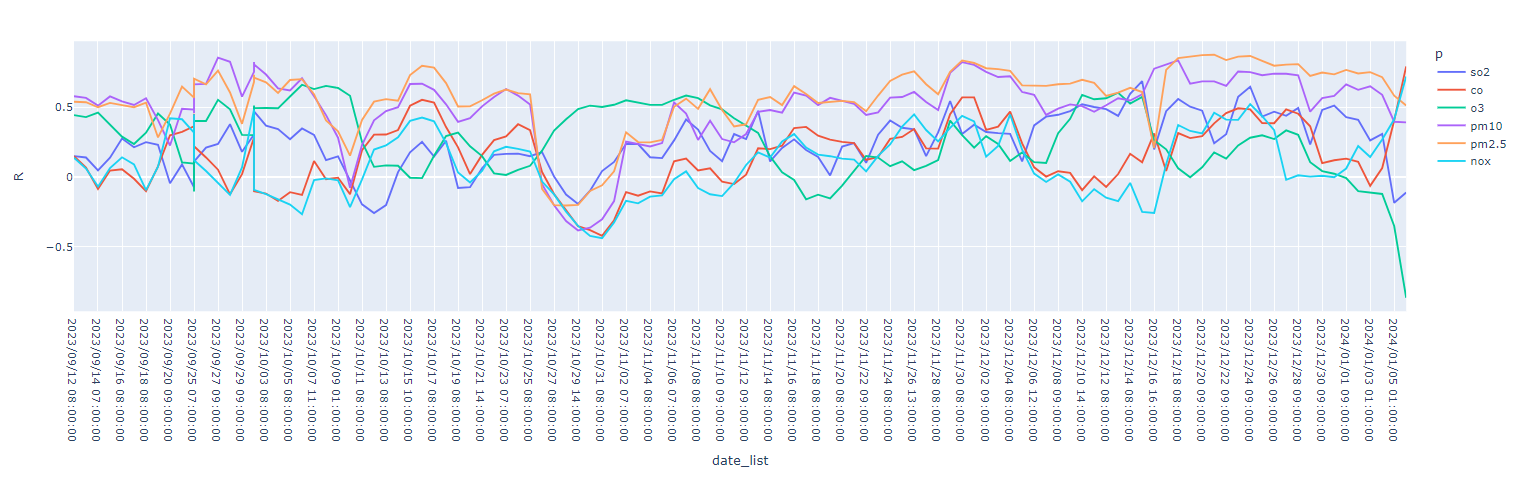


圖 7 左營關聯係數隨時間變化圖

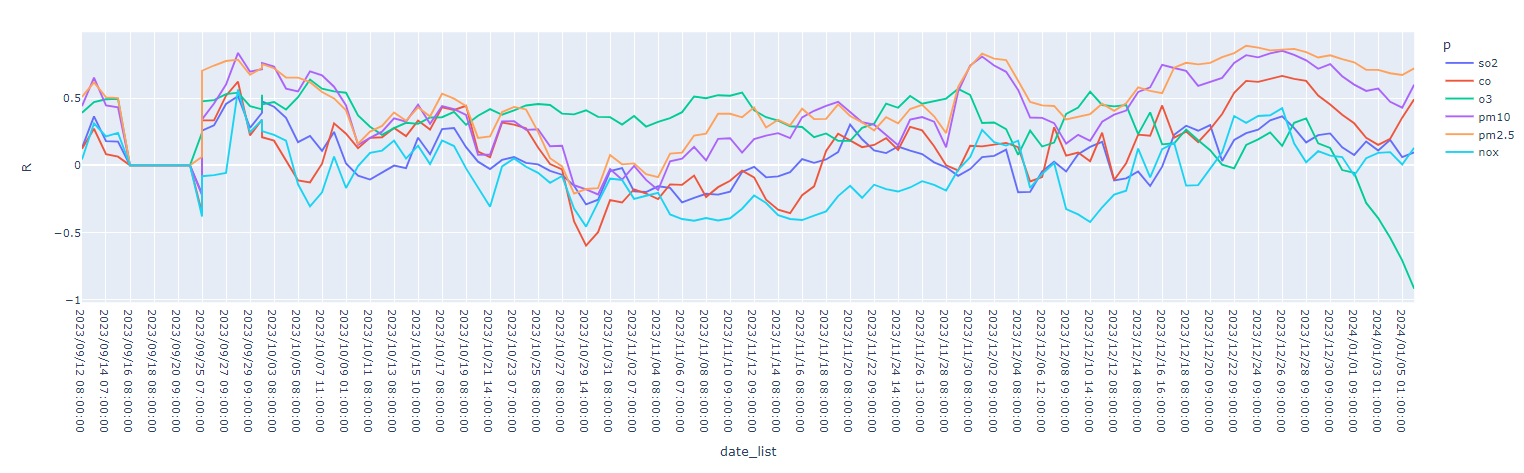


圖 8 林園關聯係數隨時間變化圖



圖 9 仁武關聯係數隨時間變化圖

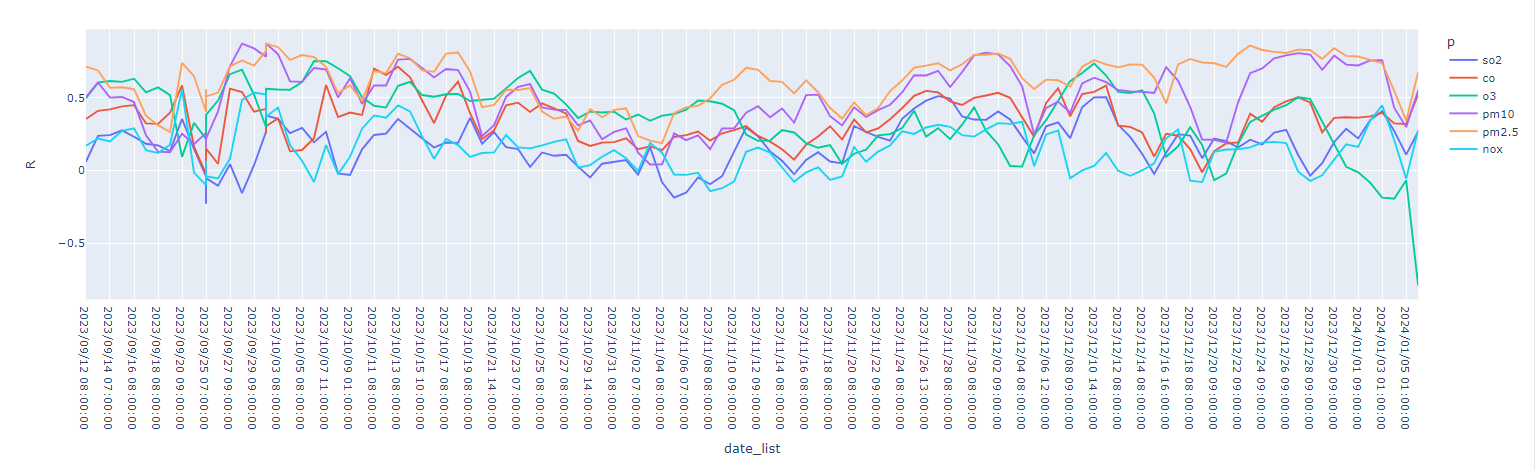


圖 10 琉球關聯係數隨時間變化圖



圖 11 大寮關聯係數隨時間變化圖

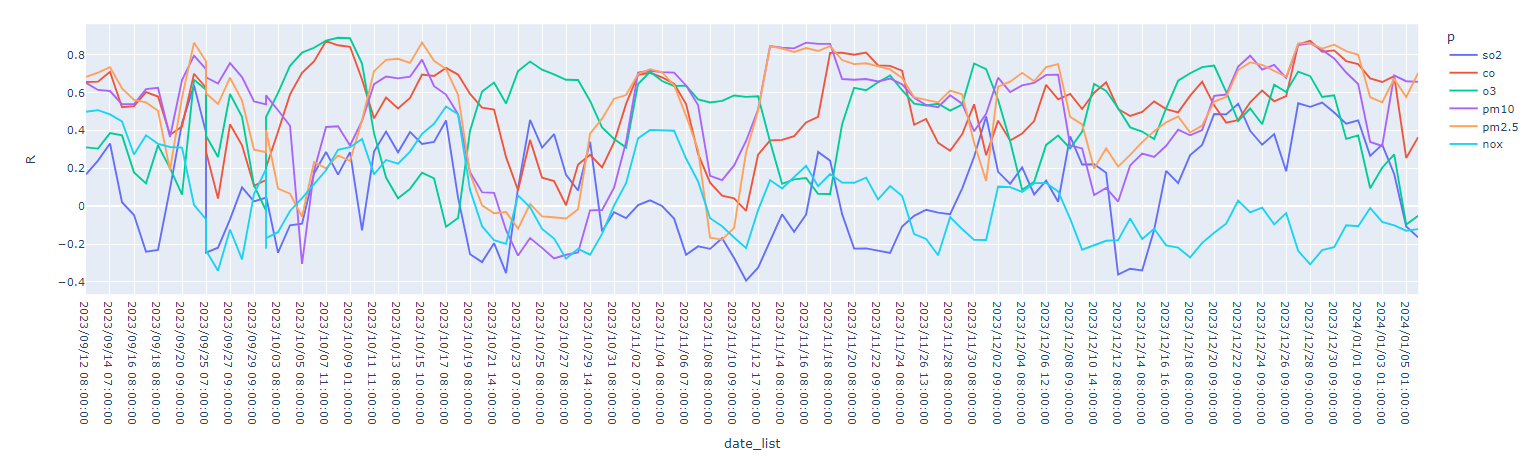


圖 12 關山關聯係數隨時間變化圖

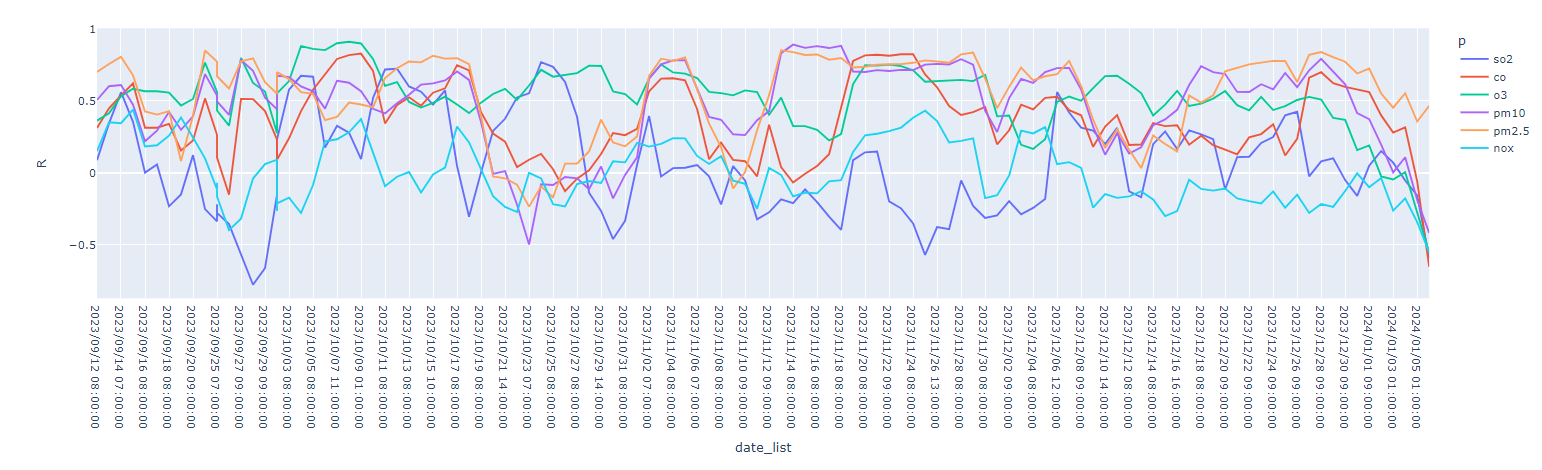


圖 13 台東關聯係數隨時間變化圖



圖 14 花蓮關聯係數隨時間變化圖

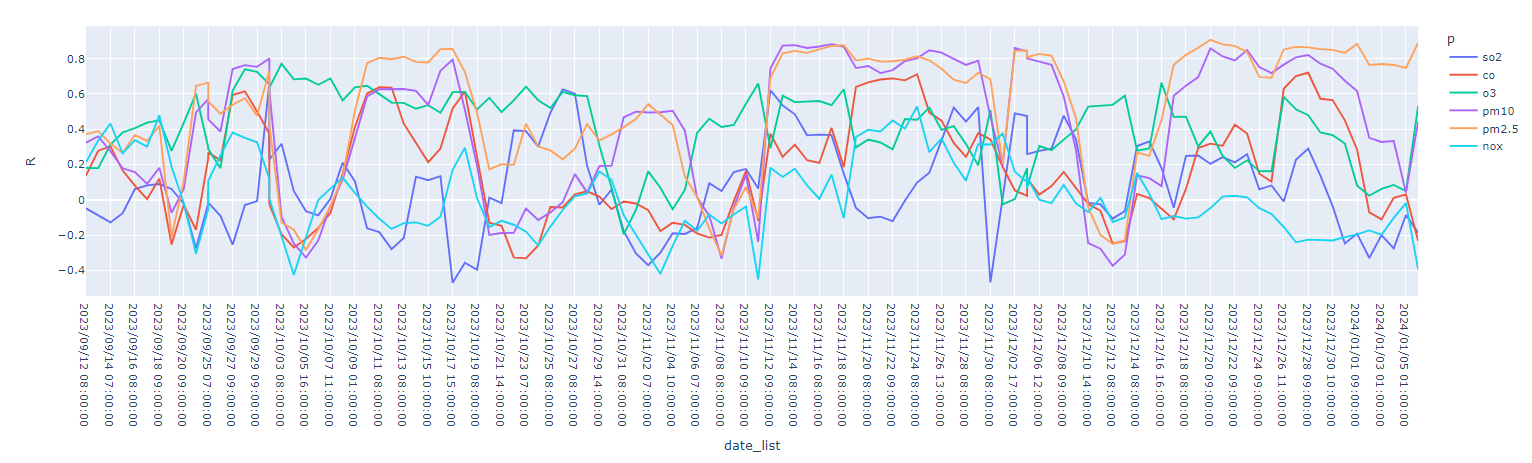


圖 15 冬山關聯係數隨時間變化圖

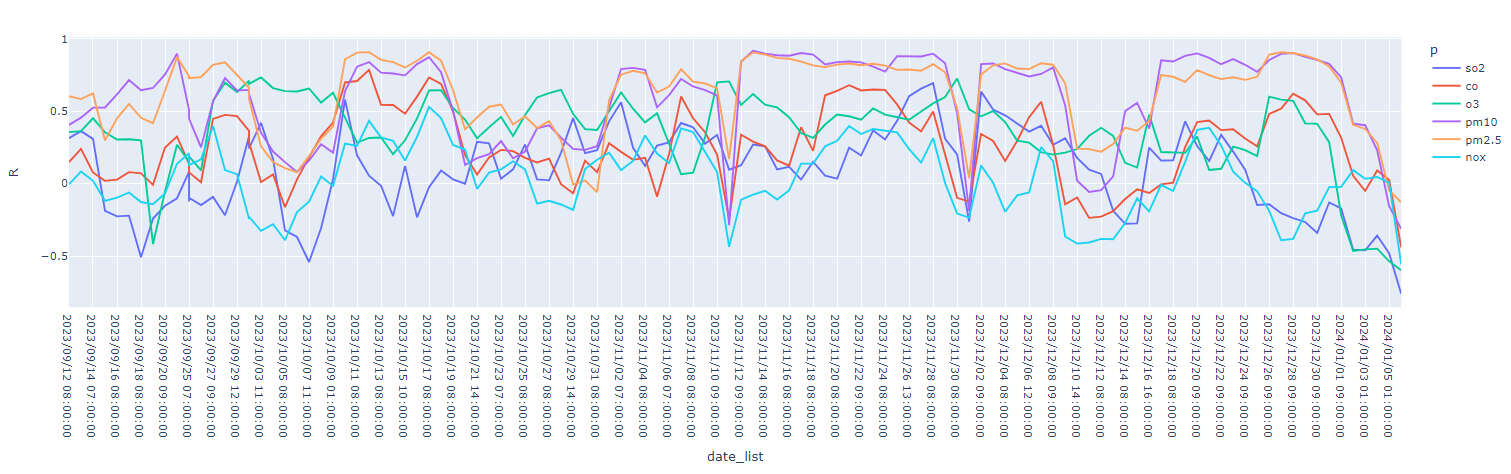


圖 16 宜蘭關聯係數隨時間變化圖

由於AQI本身是將各種汙染物指標取其最大值，因此總會有某些時間某種汙染物會與AQI最有關聯，例如以左營與林園為例子，在2023.12.4~2024.1.6這段期間，PM2.5與AQI會最有關聯，而PM2.5與PM10也有強相關，所以一樣可以在這個圖中發現PM10也是處於相對高值。汙染嚴重區域與污染不嚴重之區域有一個明顯差異在於，汙染嚴重的區域其關聯係數曲線大部分情況是偏上邊，但污染不嚴重之區域其曲線上下變化較為劇烈，推測原因是汙染物較低，一點波動就會讓相關係數變化劇烈，但因為不是常態高或低值，所以重要汙染物一直變換。圖 16有一個比較特殊的現象，其關聯係數在2024.1.5每種汙染物都下降，若攤開原始數據可以發現此時各種汙染物變化很小，沒有汙染物作為先行指標，所以顯得AQI值與任何一種汙染物都無關。

**3-2 預測模型**

本研究將使用LSTM作為預測之模型其結構如圖 17，輸入的數據由圖 18做舉例，其中AQI為要預測的對象，輸入的數據會用0~11筆數據，共12筆包含AQI、SO2、CO、O3、O3(8hr)、PM10、PM2.5、NO2、NO這些欄位共108個數據，去預測第12筆也就是33，再輸入會用1~12筆數據來預測第13筆也就是34，使用資料數為1841筆，參數調整epochs設100，batch\_size設30。圖 19說明隨著不同epoch其loss變化，

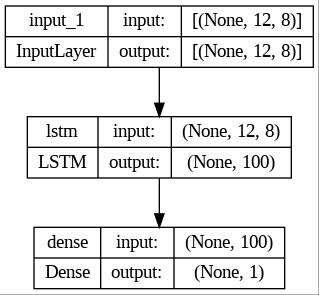


圖 17

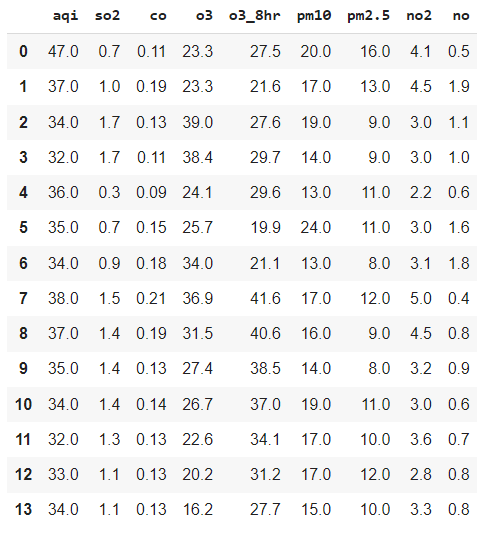


圖 18

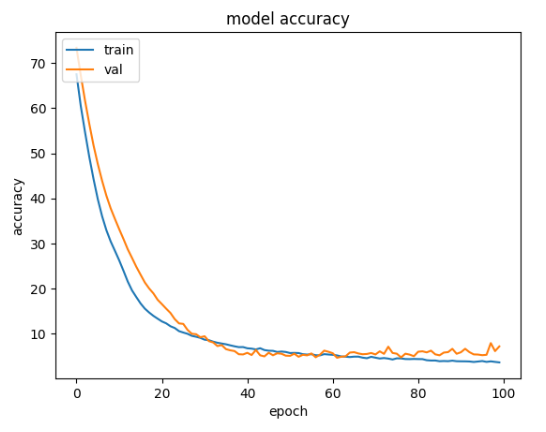


圖 19 不同epoch其loss變化

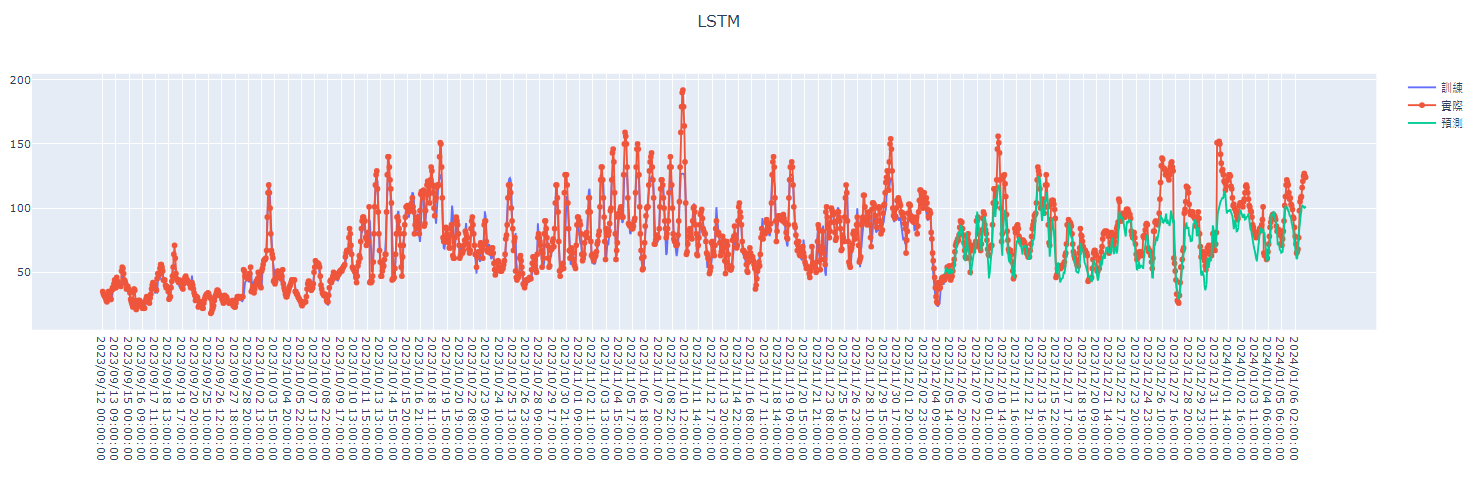


圖 20 林園數據跑LSTM之結果，綠色為模型預測測試集之結果

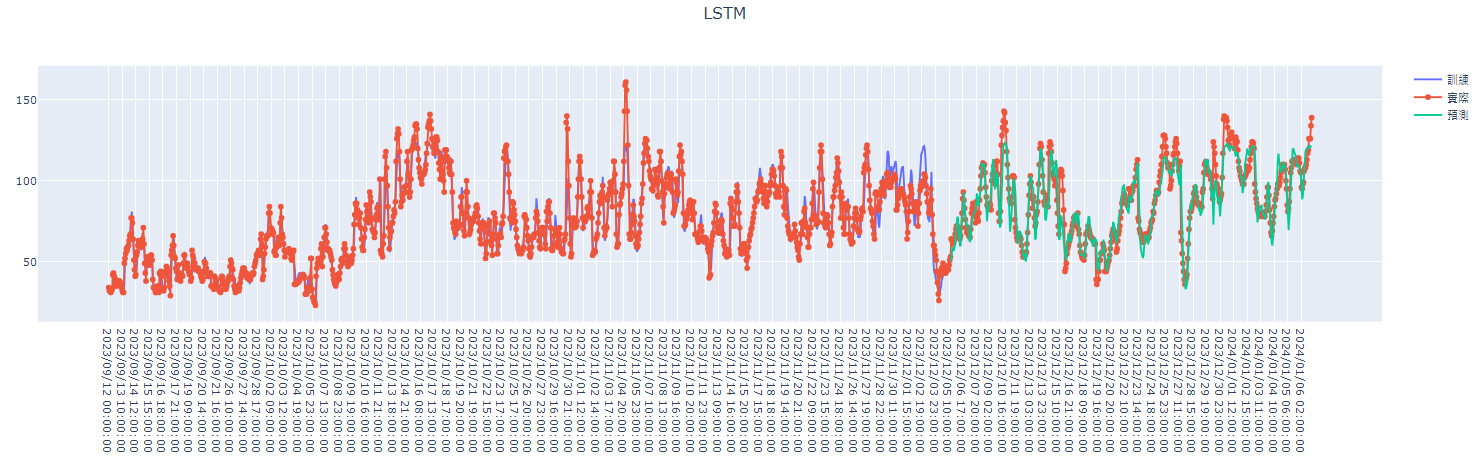


圖 21 左營數據跑LSTM之結果，綠色為模型預測測試集之結果

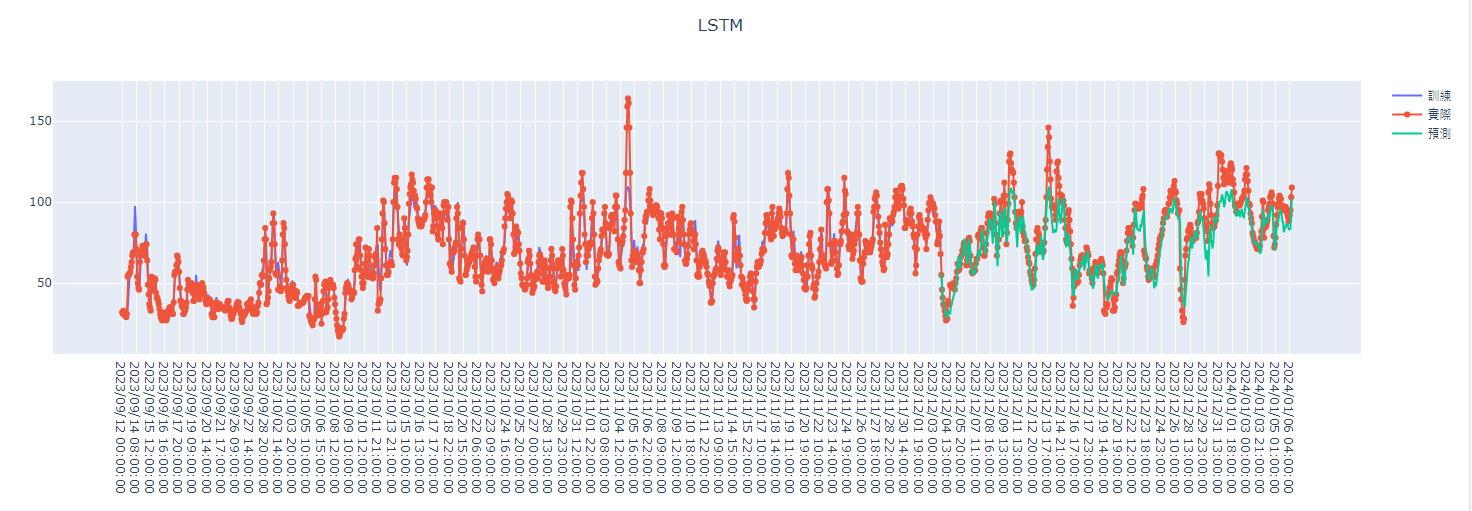


圖 22 仁武數據跑LSTM之結果，綠色為模型預測測試集之結果

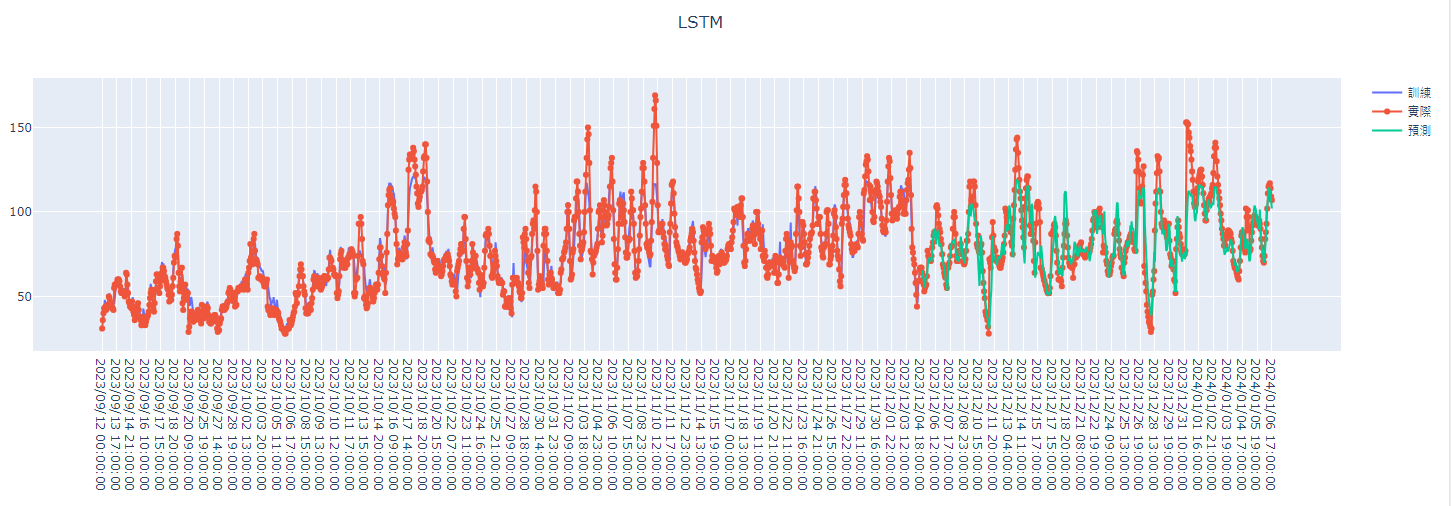


圖 23 琉球數據跑LSTM之結果，綠色為模型預測測試集之結果



圖 24 大寮數據跑LSTM之結果，綠色為模型預測測試集之結果

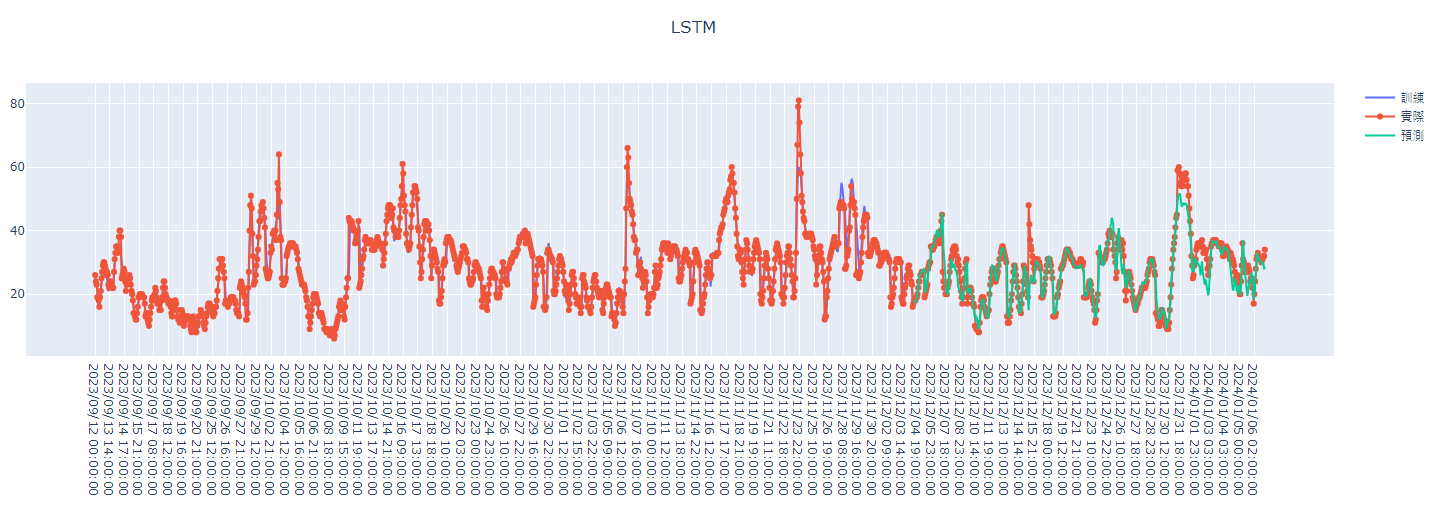


圖 25 關山數據跑LSTM之結果，綠色為模型預測測試集之結果

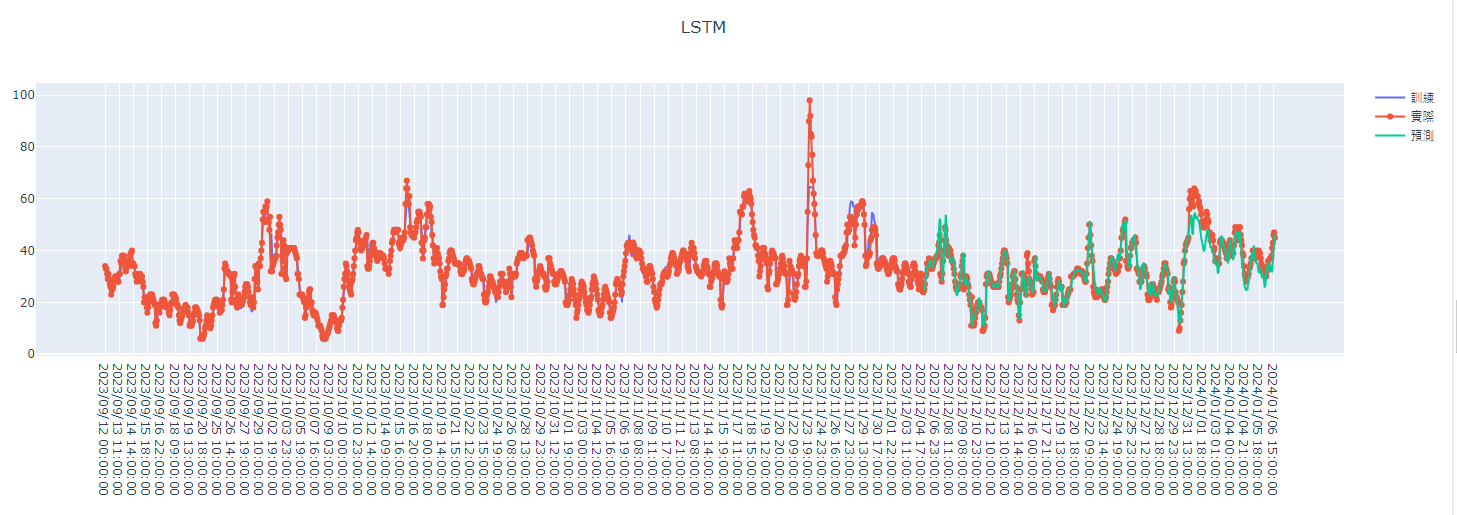


圖 26 台東數據跑LSTM之結果，綠色為模型預測測試集之結果

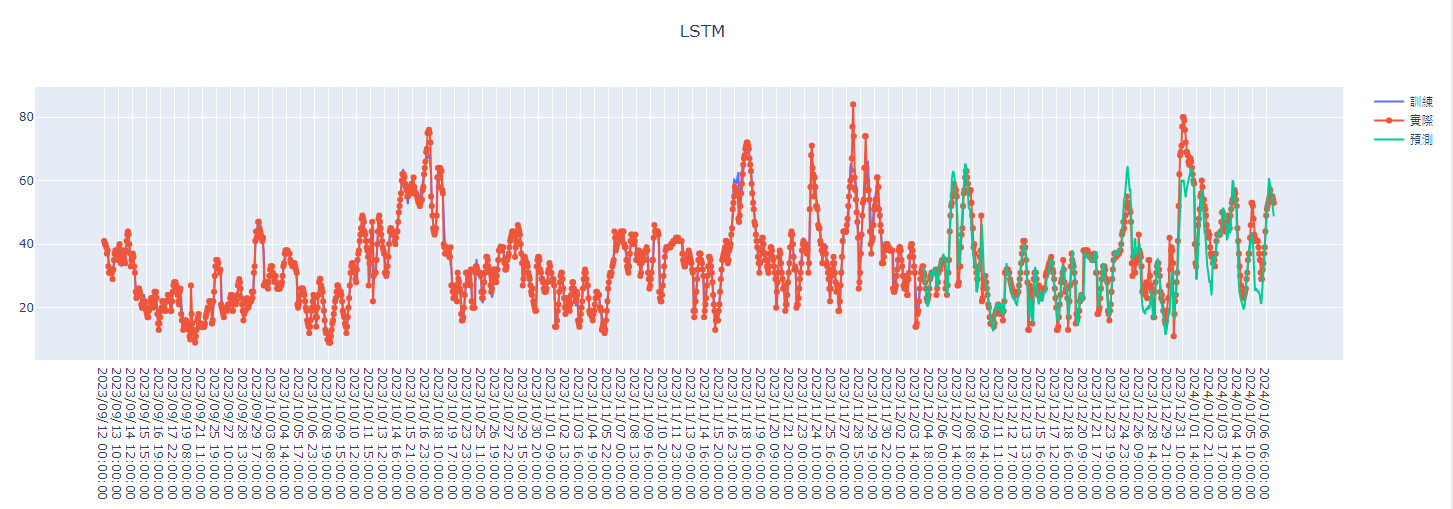


圖 27 花蓮數據跑LSTM之結果，綠色為模型預測測試集之結果

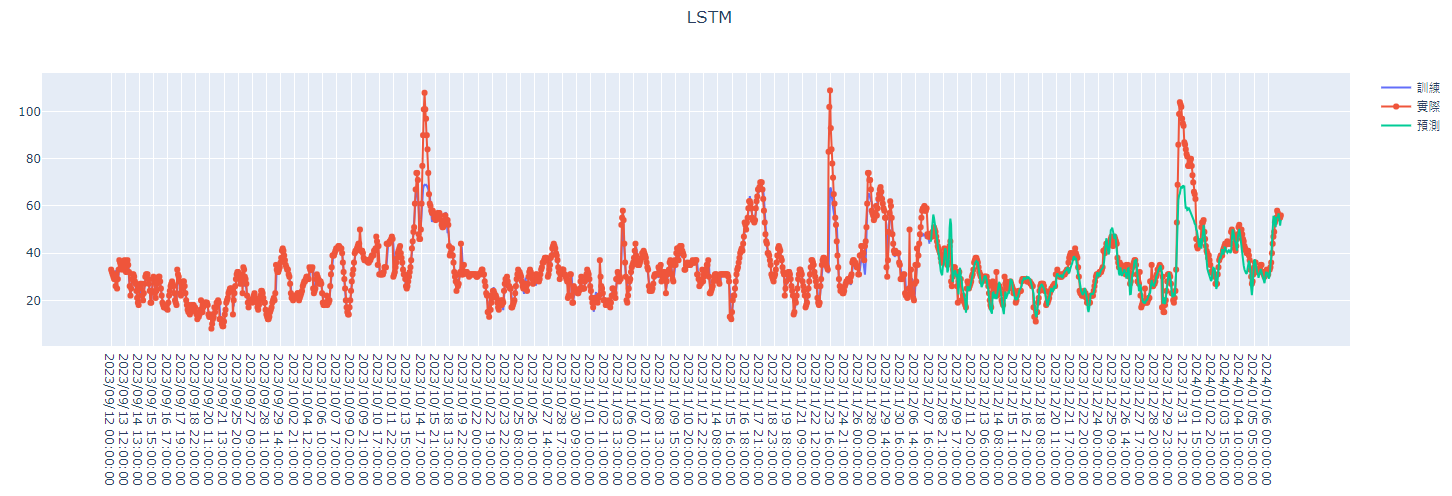


圖 28 冬山數據跑LSTM之結果，綠色為模型預測測試集之結果



圖 29 宜蘭數據跑LSTM之結果，綠色為模型預測測試集之結果

**3-3 癌症與空氣品質關聯分析**

****

圖 30 癌症人數資料

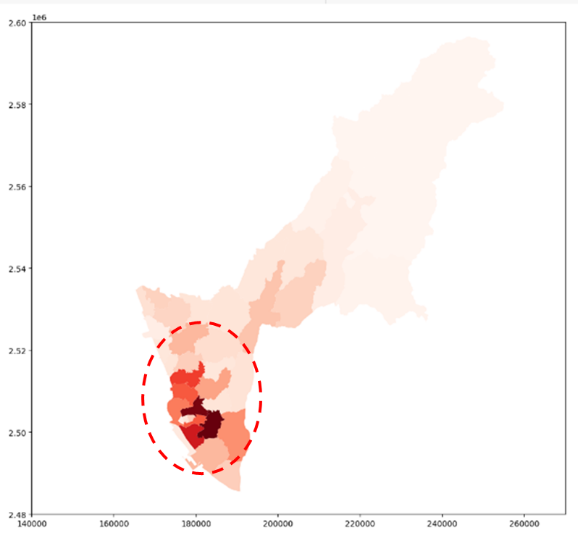
****

圖 31 高雄癌症人數之地圖分布

程式碼 : 透過下拉式選單的選項去查詢資料庫裡指定資料

一張含有 文字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述

程式碼 : 繪製黑色的折線圖

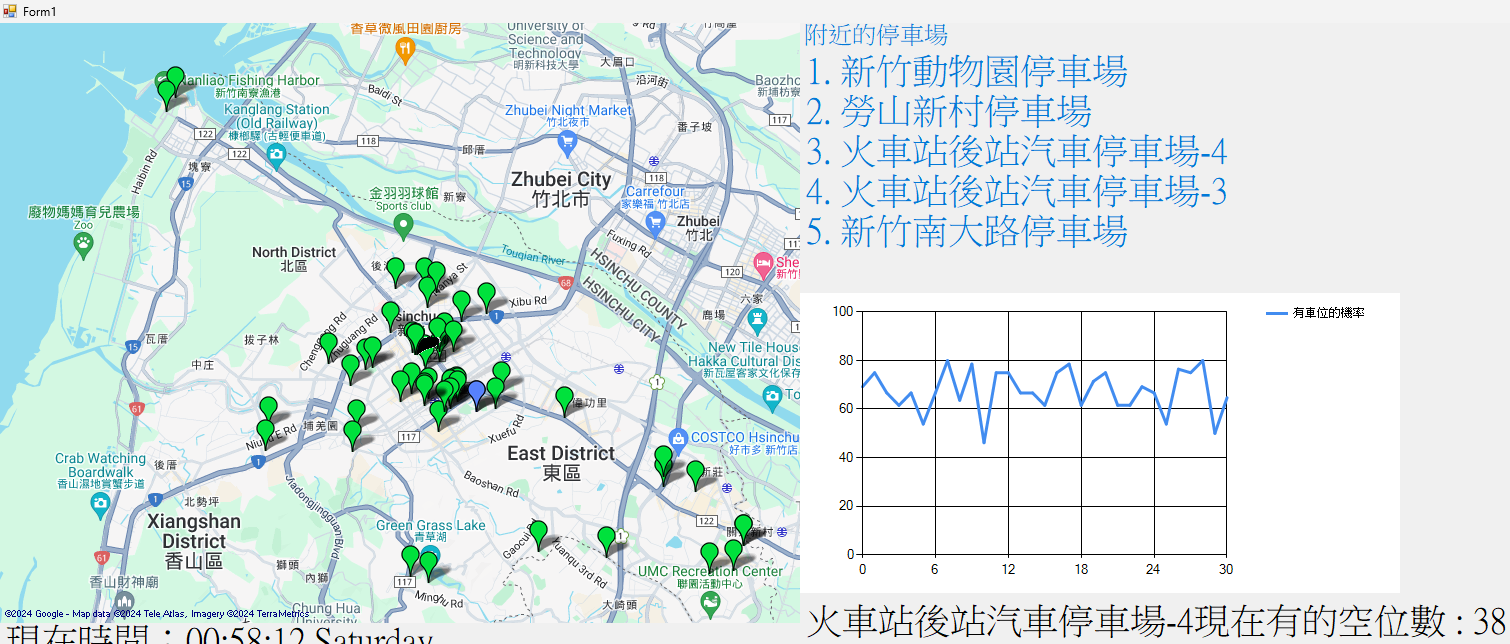
一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

**3-1 專案\_PIN**

**一張含有 文字, 地圖, 地圖集, 字型 的圖片

自動產生的描述**

在地圖上點擊兩下會產生藍色標點，此時旁邊的附近的停車場會顯示離該藍點前五近的停車場。

此時點擊其中一個停車場後(這裡點擊了火車後站停車場-4)，就會產生現在時間到後30分鐘的有剩餘空位的機率的折線圖，以及折現圖下方會透過即時爬蟲所抓下來的該停車場現在的空位數。

程式碼 : 標記藍點

程式碼 : 產生前五近的停車場

**第四章 問題與討論**

**4-1. 問題**

本來想要使用機器學習去預測數據，也進行了許多嘗試，所使用的是LSTM，在這不斷嘗試的期間，調整了許多不同種的數值(learning rate,epochs,batch\_size,units)，

也改過許多次fetcher內容，但成果皆沒有起色，於是只能把此問題已人心莫測畫下結尾。

**第五章 結論**

**5-1. 結語**

這次的專題還有許多進步的空間，例如程式碼太繁瑣，其中能夠省下許多步驟，也能結合其他資料進行更多元的功能，例如:停車場附近的餐廳OR景點。我相信這個視窗能夠在某些時候給予幫助吧。

**5-2. 參考資料**

[<https://data.gov.tw/dataset/129136>](https://data.gov.tw/dataset/129136)