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

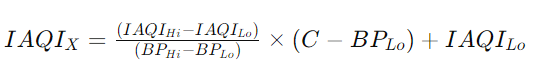


圖 1 子指標計算公式

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。

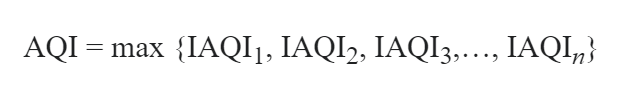


圖 2 各種濃度算完取最大值

最終可以得到一個對應的分數範圍，如圖 3。

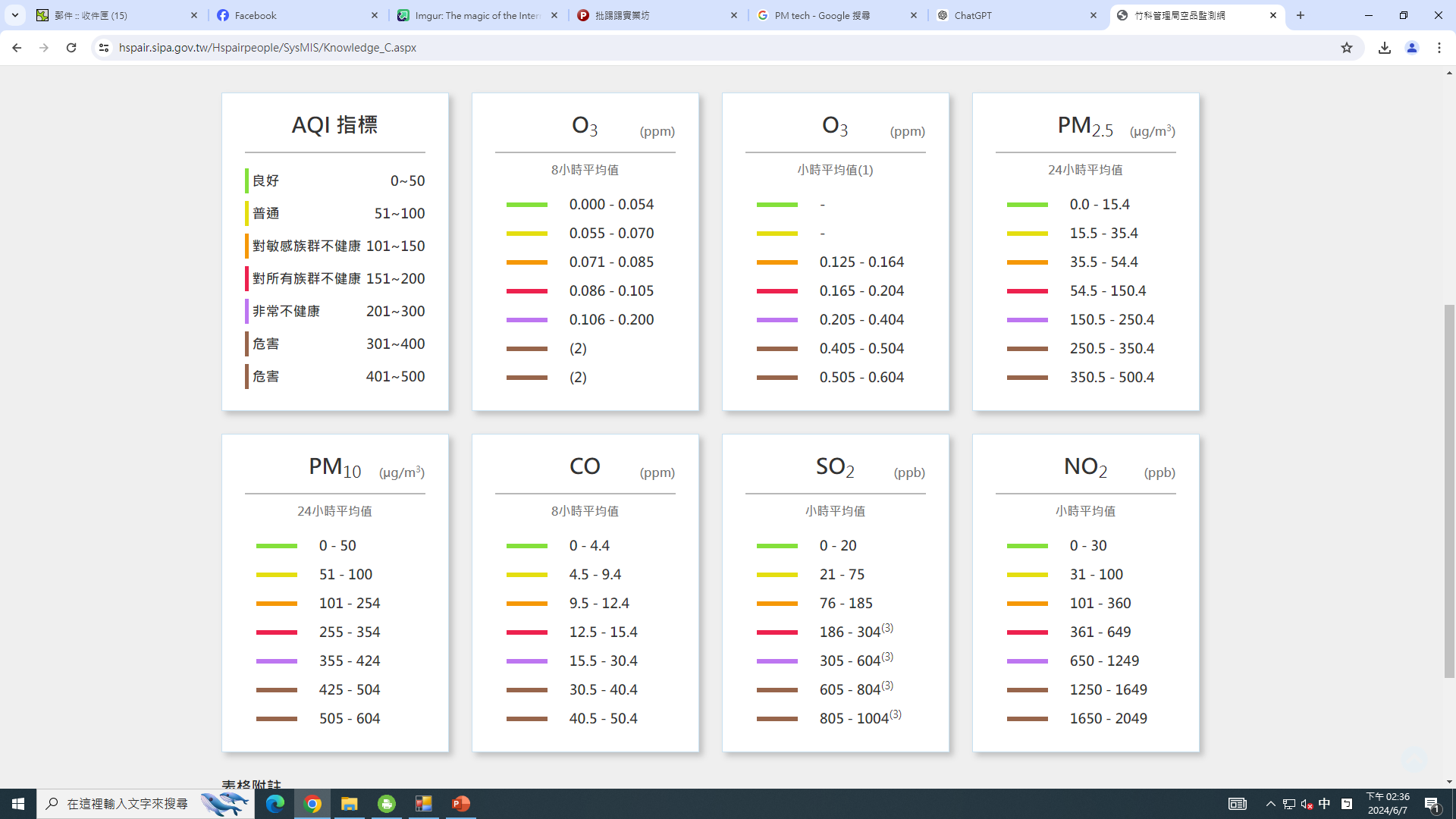


圖 3 各種汙染物對應分數

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**

**2-1 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

**2-1 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

**2-2 方法說明\_LSTM**

全名為長短期記憶(Long short-term memory)，會使用記憶來更新訊息，並控制訊息流動，使模型能夠更優化捕捉長期依賴關係。相較於簡易的 RNN，LSTM 透過多個可學習的 gate 較能夠觀察長距離的資訊而不會遺忘較早輸入的訊息。

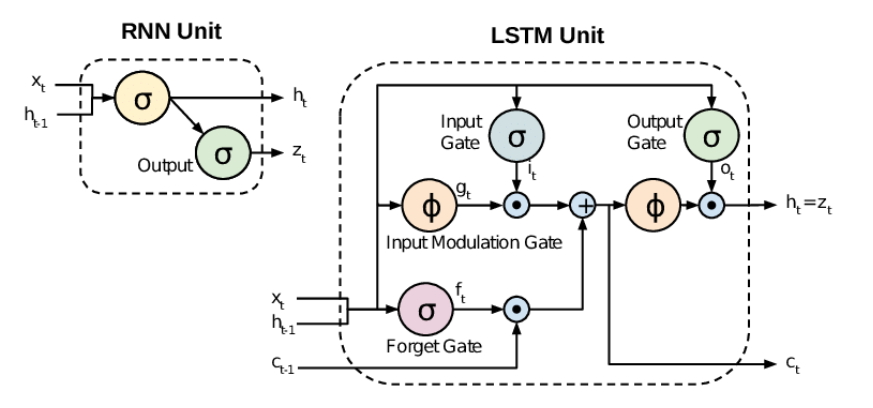


圖 4 LSTM示意圖

**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 6

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

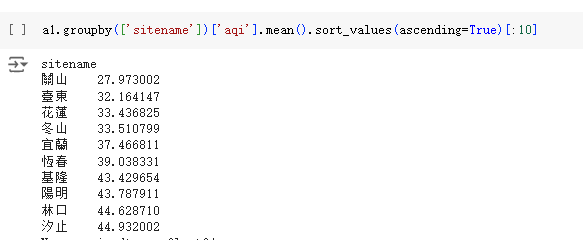
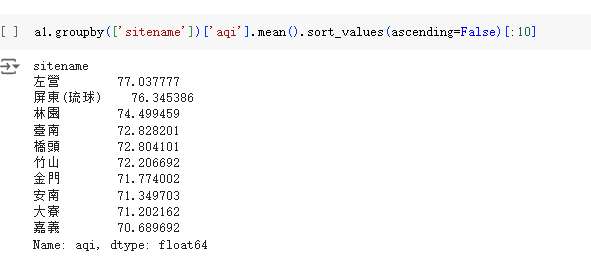


圖 5 用groupby篩選出最高五個及最低五個

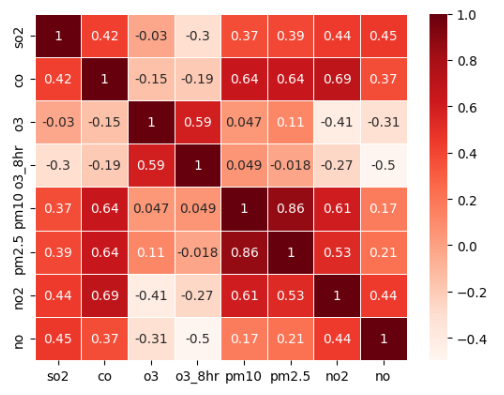
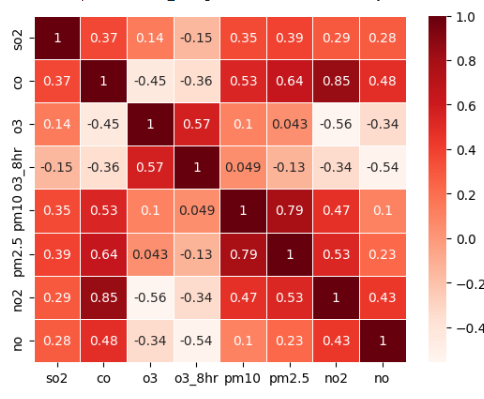


圖 6 左邊為左營右邊為林園

其中越深色表示越有關聯如圖 6，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。

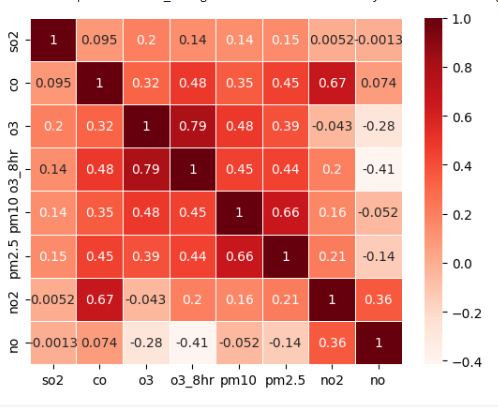
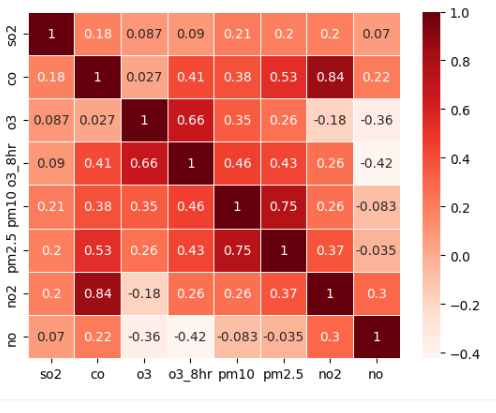


圖 7 左邊為花蓮右邊為台東

圖 7為AQI較小之區域，關聯係數與圖 6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):#計算Pearson係數

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'])

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

temp\_sitename=a[a['sitename']=='左營']#篩選出左營這個地區所有數據

temp\_sitename=temp\_sitename.sort\_values(by=['datetime'])#以時間做排序

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']))#計算AQI與so2之關聯係數

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'])) #計算AQI與co之關聯係數

# 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'])) #計算AQI與o3之關聯係數

# 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'])) #計算AQI與pm10之關聯係數

# 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'])) #計算AQI與pm2.5之關聯係數

# 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'])) #計算AQI與nox之關聯係數

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})#將上述數據汙染物、時間、關聯係數做成dataframe進行畫圖

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

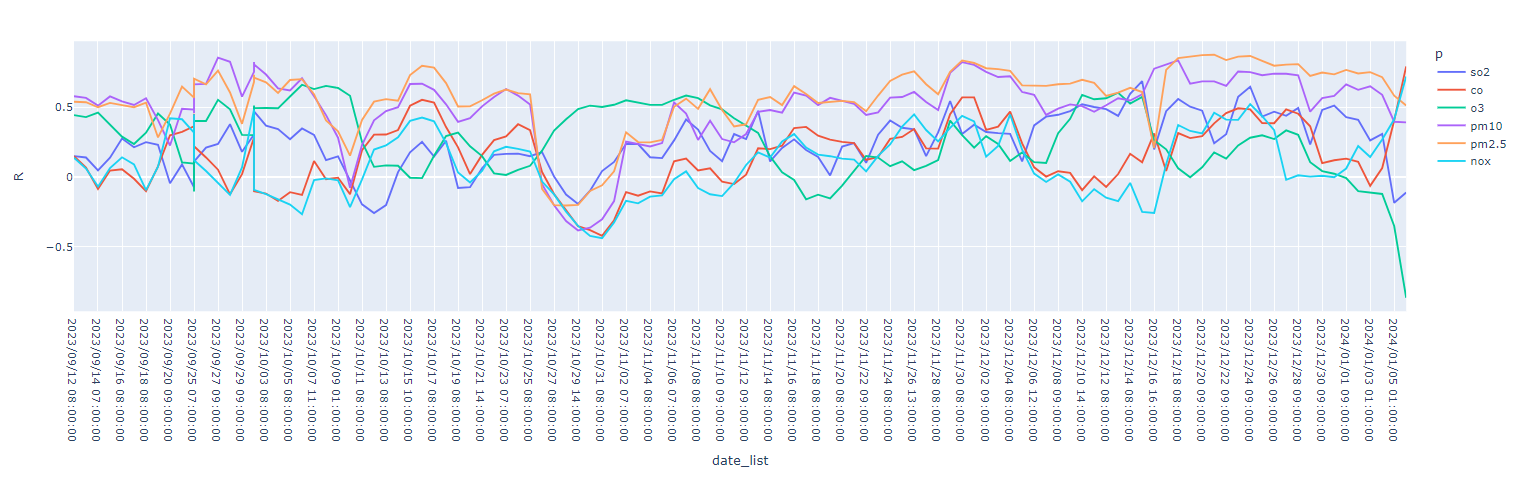


圖 8 左營關聯係數隨時間變化圖

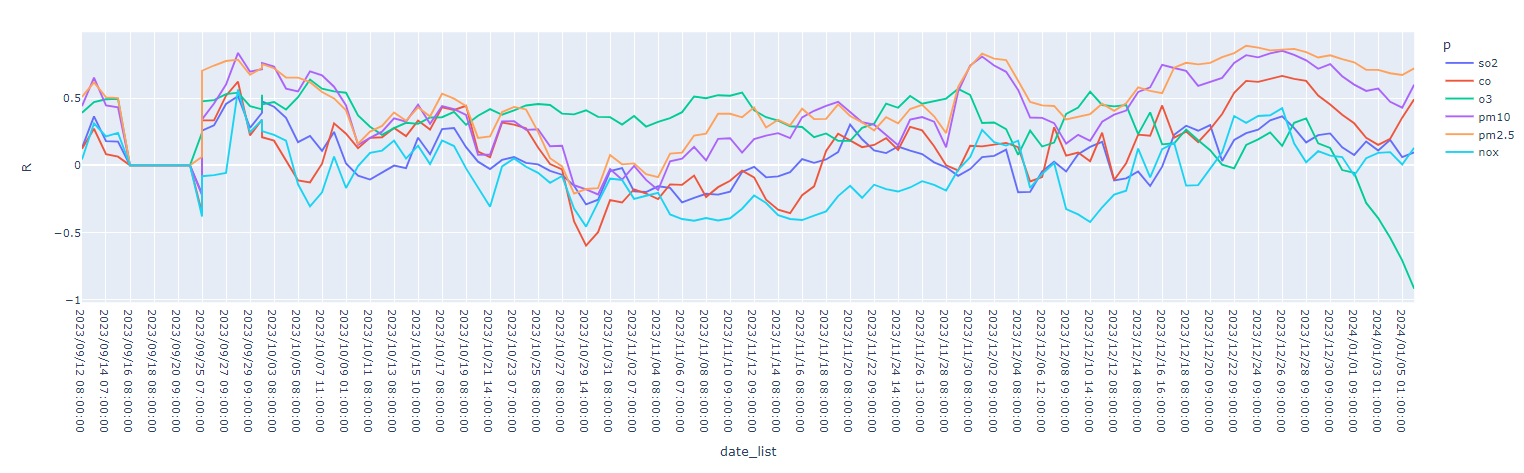


圖 9 林園關聯係數隨時間變化圖



圖 10 仁武關聯係數隨時間變化圖

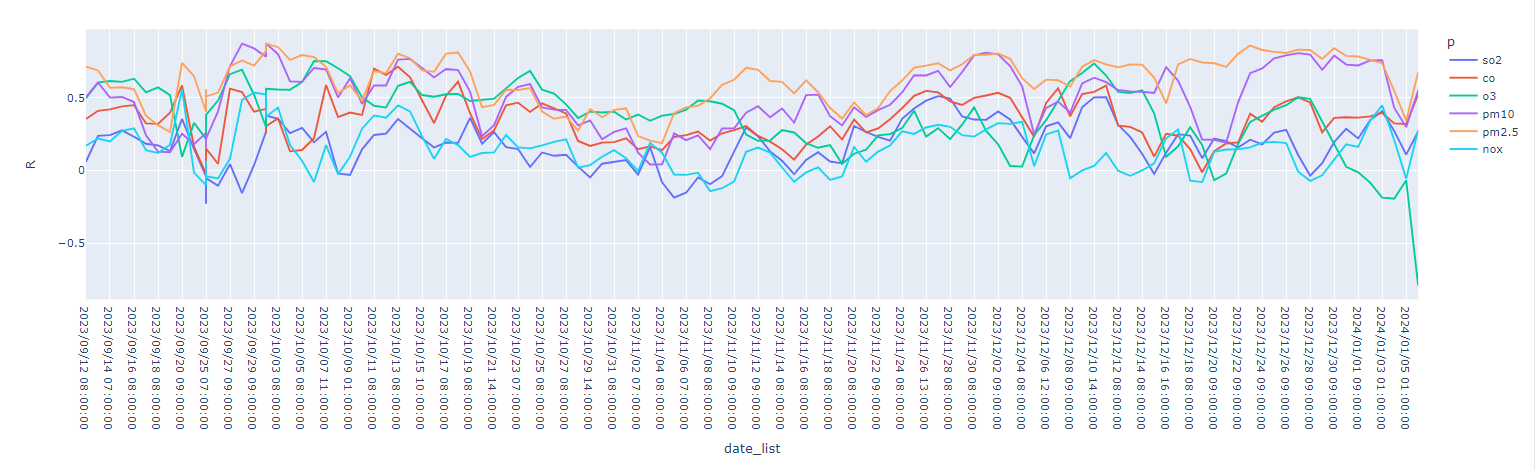


圖 11 琉球關聯係數隨時間變化圖

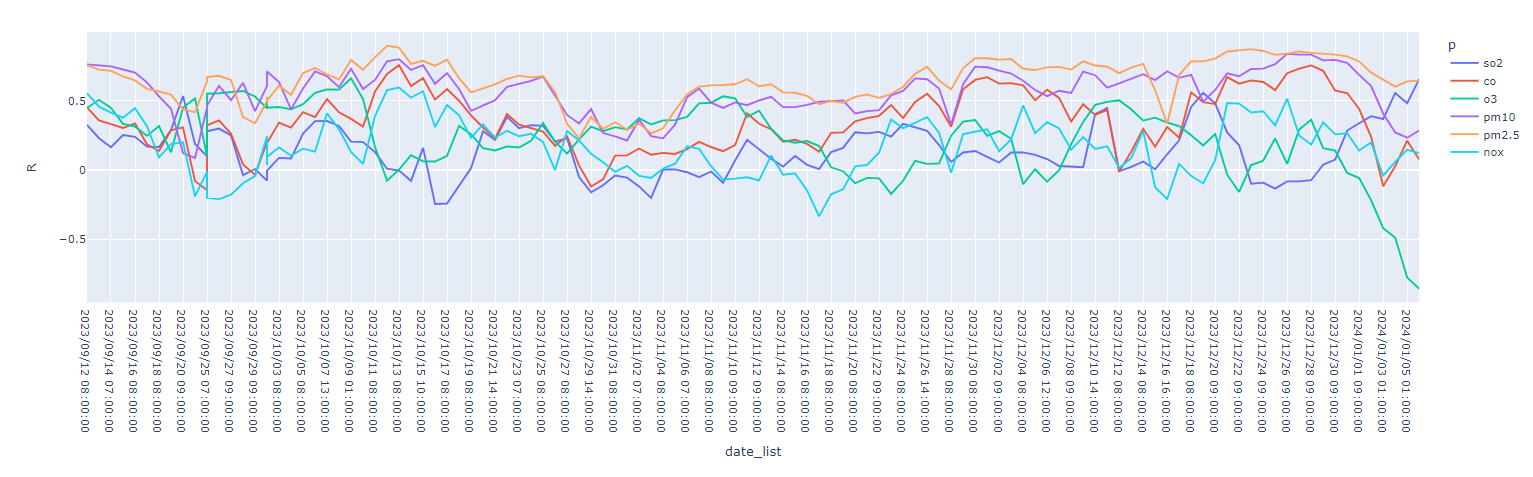


圖 12 大寮關聯係數隨時間變化圖

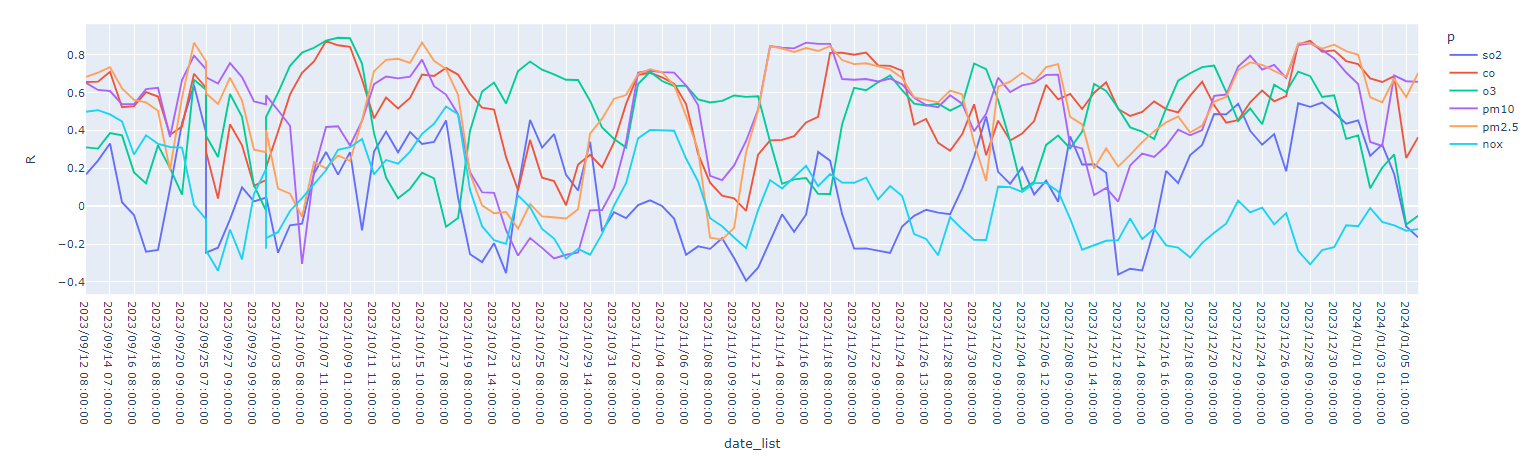


圖 13 關山關聯係數隨時間變化圖

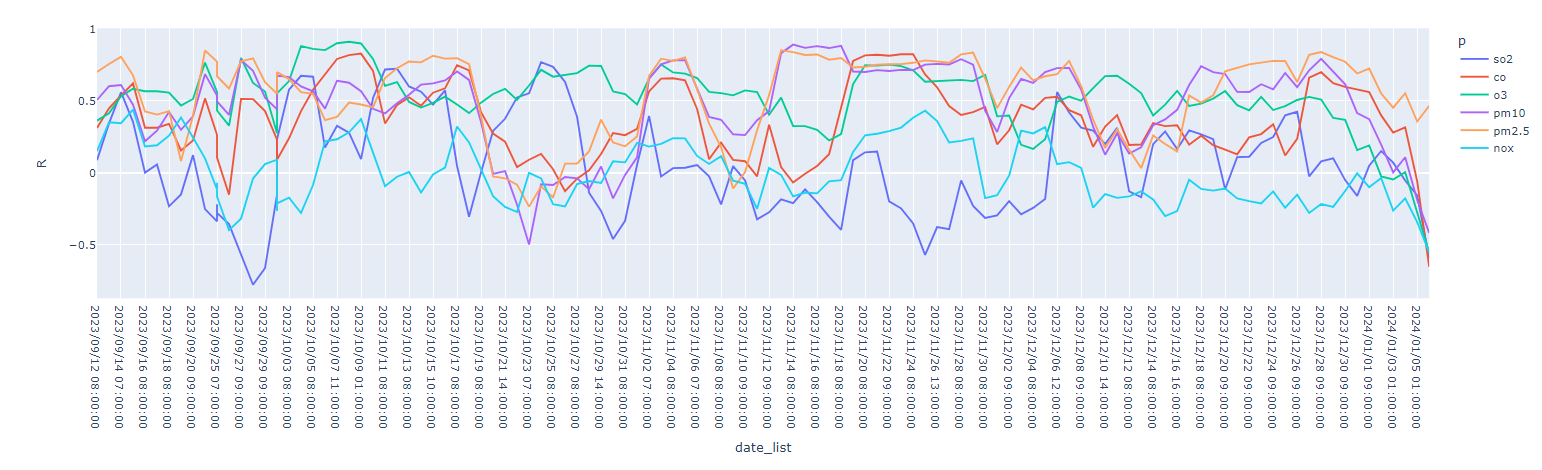


圖 14 台東關聯係數隨時間變化圖



圖 15 花蓮關聯係數隨時間變化圖

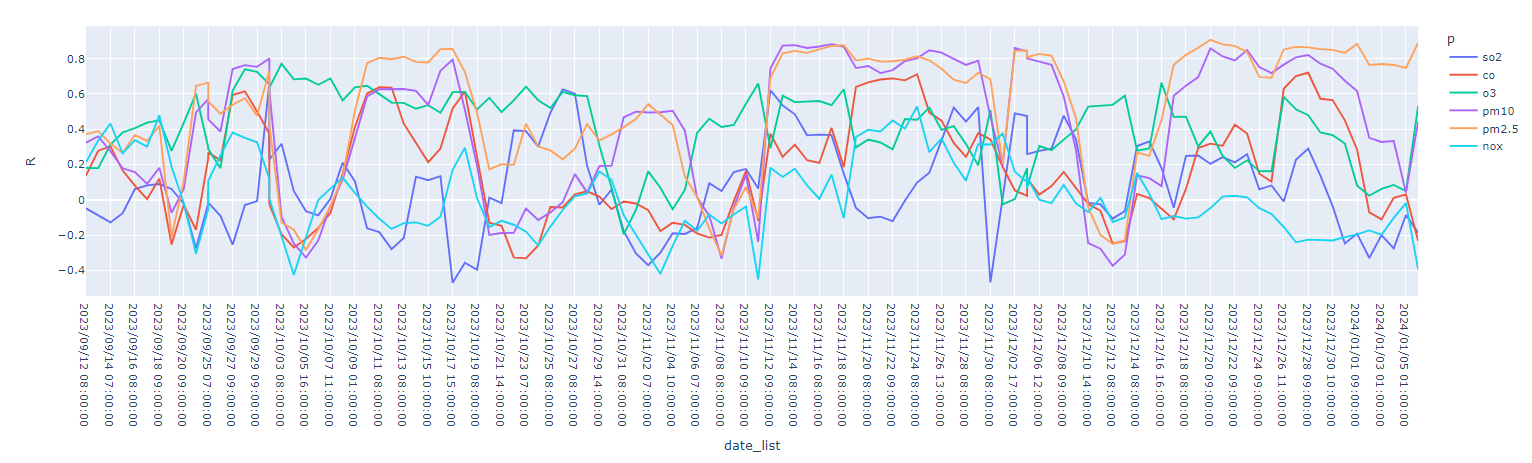


圖 16 冬山關聯係數隨時間變化圖

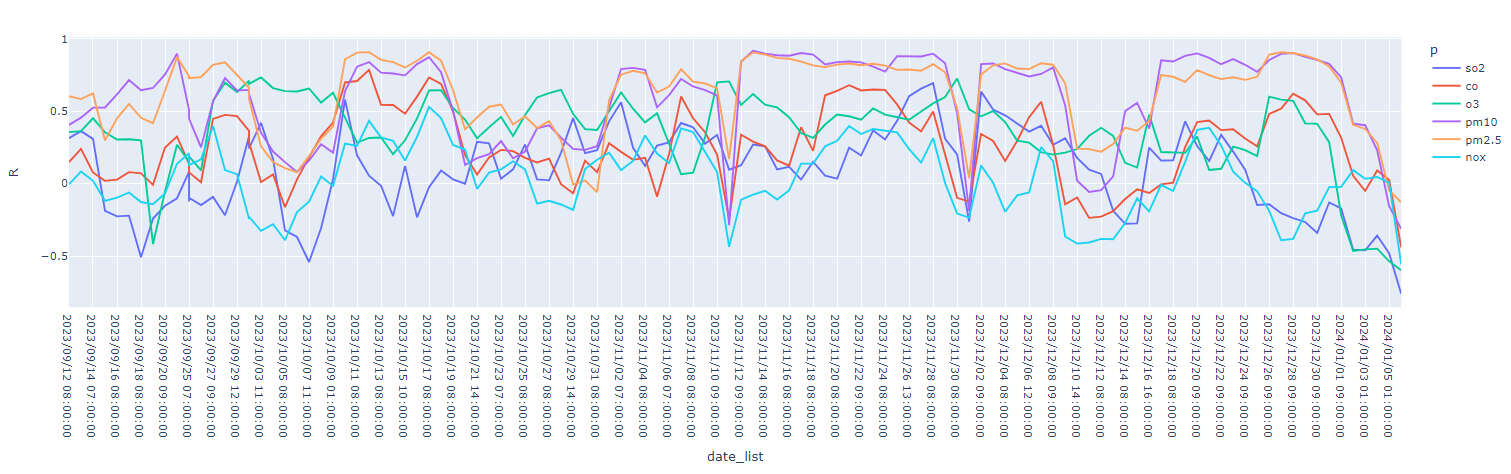


圖 17 宜蘭關聯係數隨時間變化圖

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 17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 18，輸入的數據由圖 19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 20說明隨著不同epoch其loss變化，程式碼如下

air = pd.read\_csv('/content/drive/MyDrive/ppt/air\_total.csv')

air=air.drop(['pollutant','status'],axis=1)

air=air[air['sitename']=='臺東'].dropna()

air=air.sort\_values(by=['publishtime'])

air=air.reset\_index(inplace=False,drop=True)

lookback\_window = 12

air\_v=air.iloc[:,2:11]

air\_v1=air\_v.values

x = []

y = []

for i in tqdm(range(lookback\_window, len(air\_v1))):

x.append(air\_v1[i - lookback\_window:i,1:])

y.append(air\_v1[i,0])

#以上是製作導入模型訓練之數據，會以前12筆資料預測第13筆

def root\_mean\_squared\_error(y\_true, y\_pred):

return K.sqrt(K.mean(K.square(y\_pred - y\_true), axis=-1))

x = np.array(x)

x = np.asarray(x).astype('float32')

y = np.array(y)

y = np.asarray(y).astype('float32')

lookback\_window=12

i = Input(shape=(lookback\_window, 8))

m = LSTM(units=100)(i)

m = Dense(1, activation='linear')(m)

model = Model(inputs=[i], outputs=[m])

#建構LSTM模型

model.compile('adam',loss = root\_mean\_squared\_error)

h=model.fit(x[:round(len(x)\*0.7)], y[:round(len(x)\*0.7)], epochs=100,batch\_size=30,validation\_split = 0.1)#以前70%數據進行訓練

keras.utils.plot\_model(model, to\_file='/content/1.jpg', show\_shapes=True)#繪製模型架構圖

plt.plot(h.history['loss'])

plt.plot(h.history['val\_loss'])

plt.title('model accuracy')

plt.ylabel('loss')

plt.xlabel('epoch')

plt.legend(['train', 'val'], loc='upper left')

plt.show()#繪製loss隨時間變化圖

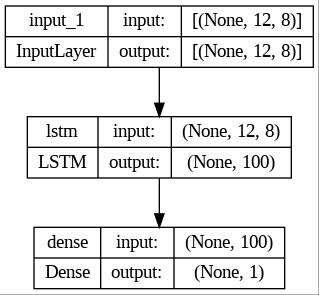


圖 18

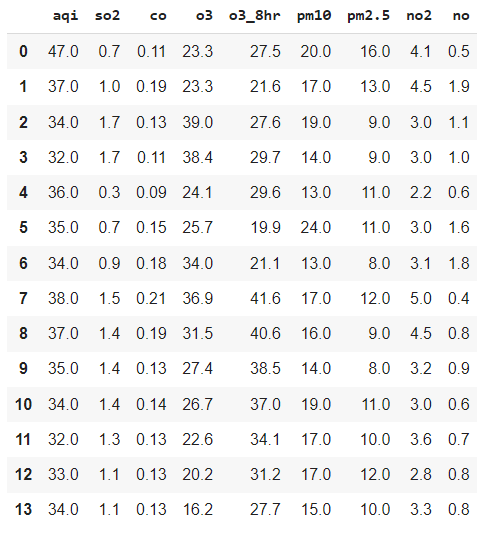


圖 19

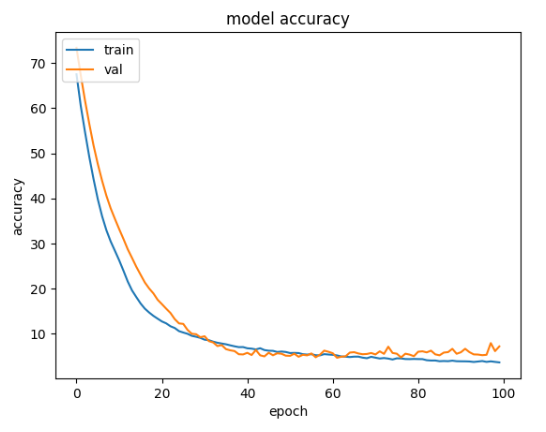


圖 20 不同epoch其loss變化

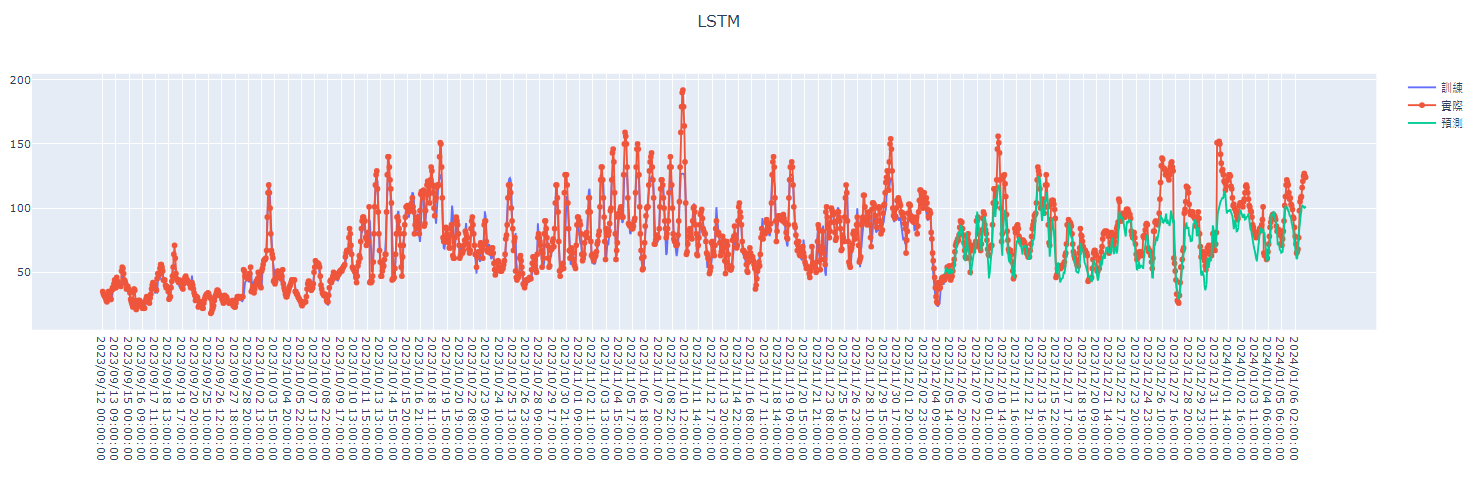


圖 21 林園數據跑LSTM之結果，綠色為模型預測測試集之結果

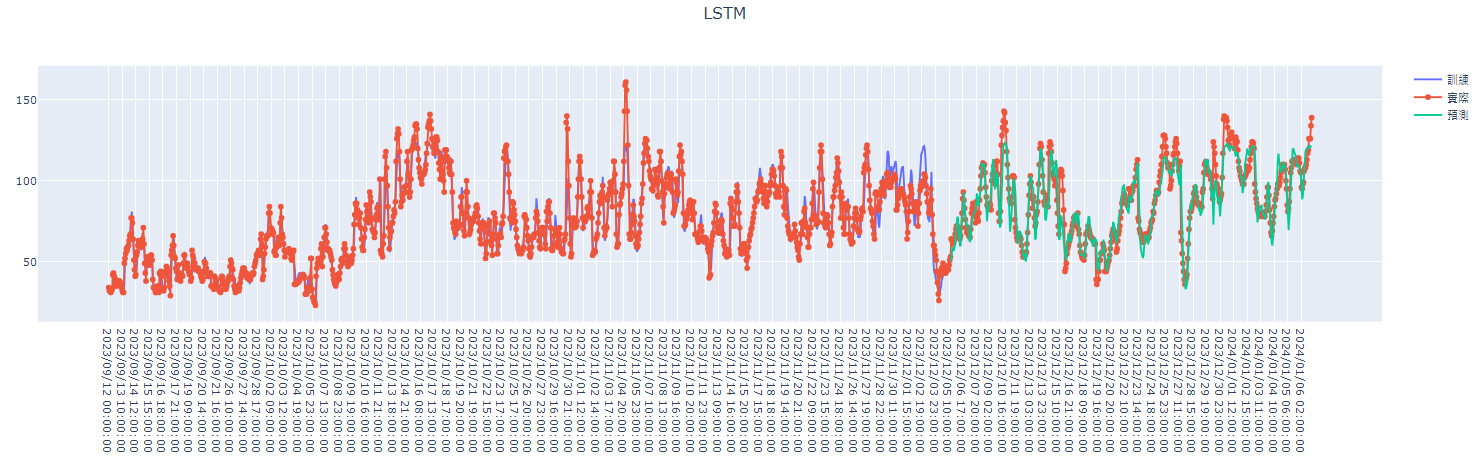


圖 22 左營數據跑LSTM之結果，綠色為模型預測測試集之結果

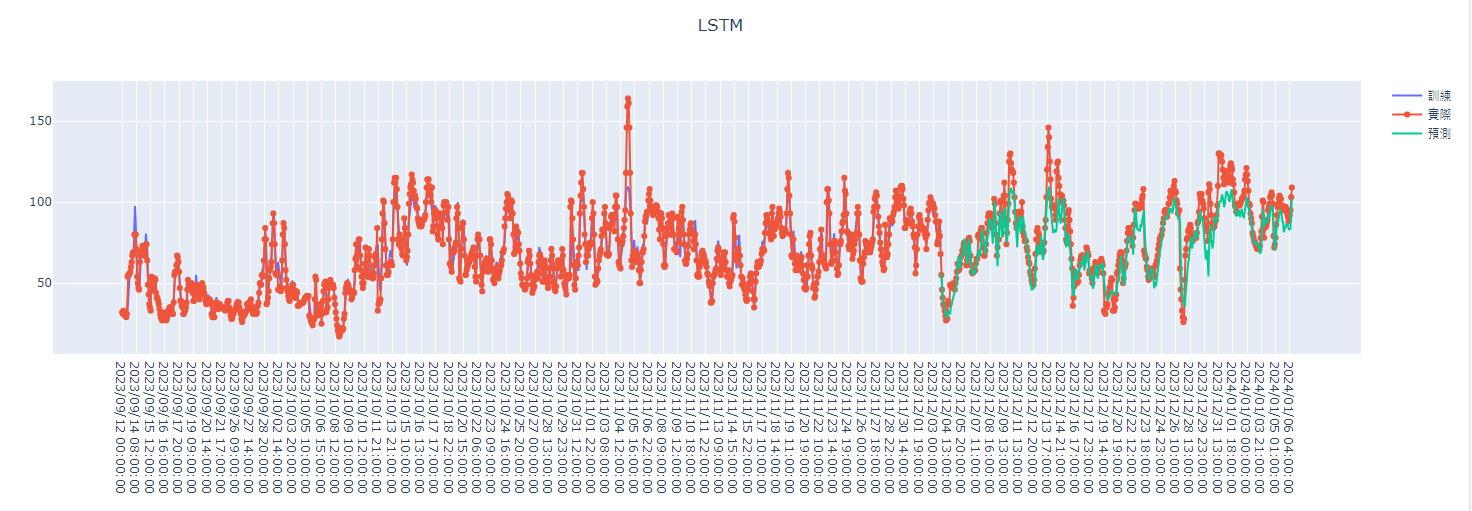


圖 23 仁武數據跑LSTM之結果，綠色為模型預測測試集之結果

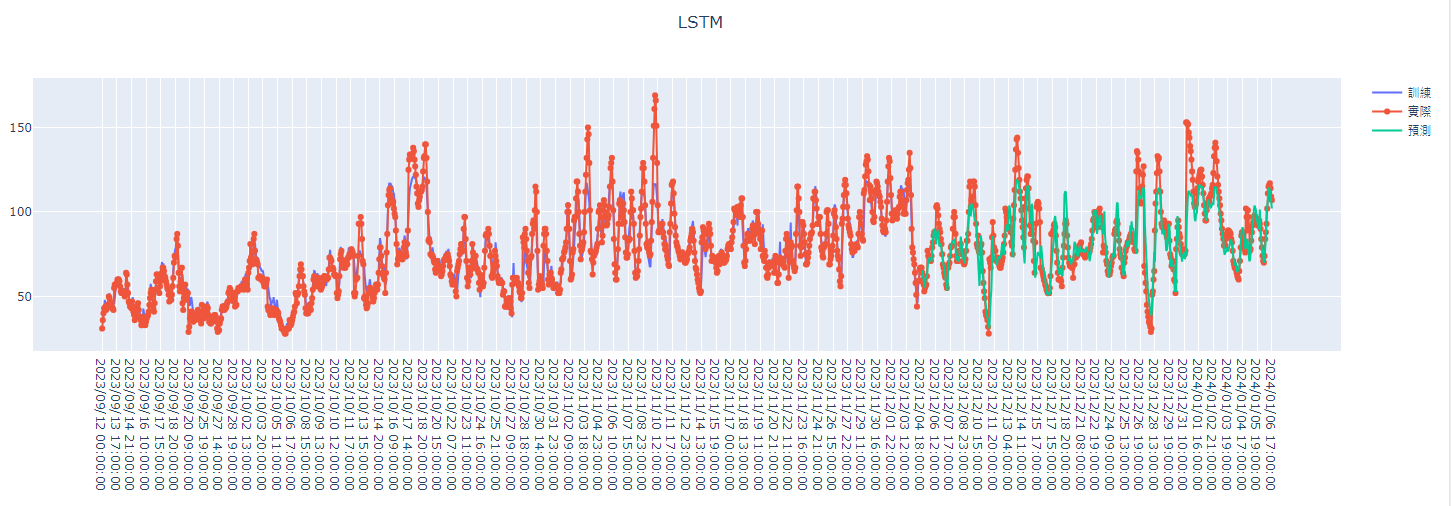


圖 24 琉球數據跑LSTM之結果，綠色為模型預測測試集之結果

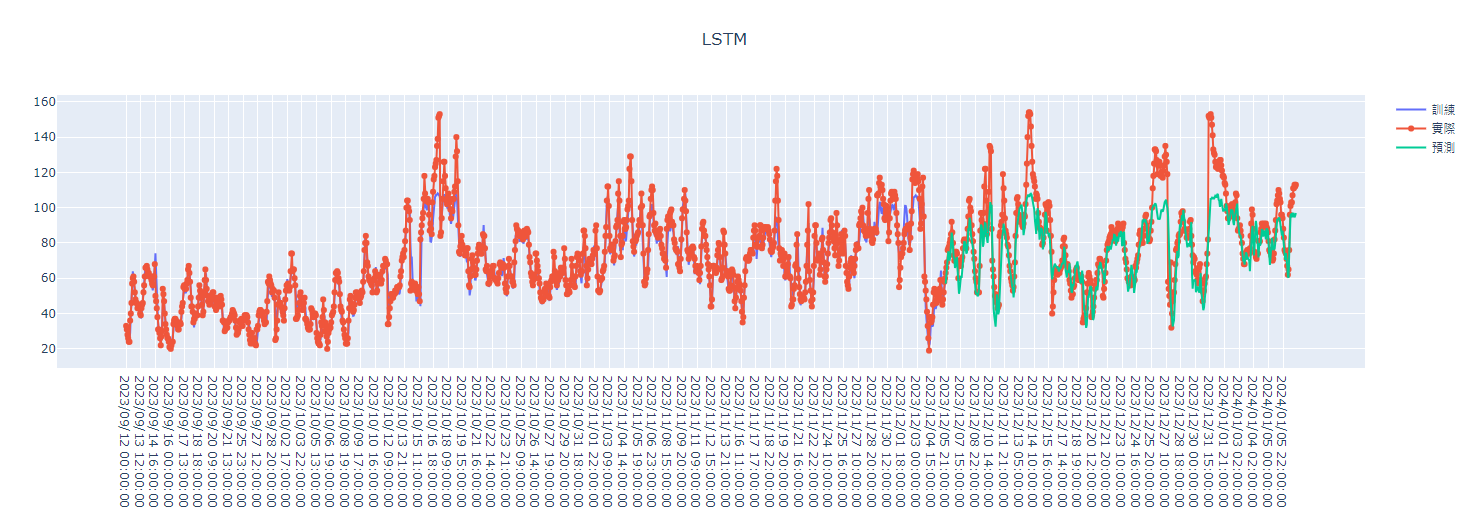


圖 25 大寮數據跑LSTM之結果，綠色為模型預測測試集之結果

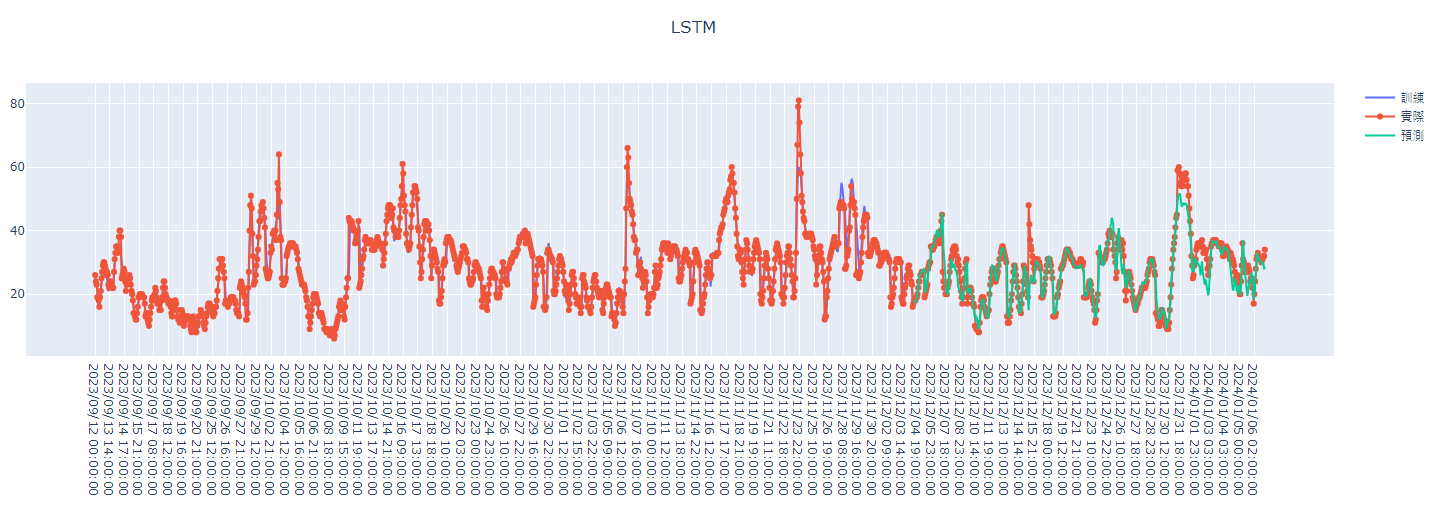


圖 26 關山數據跑LSTM之結果，綠色為模型預測測試集之結果

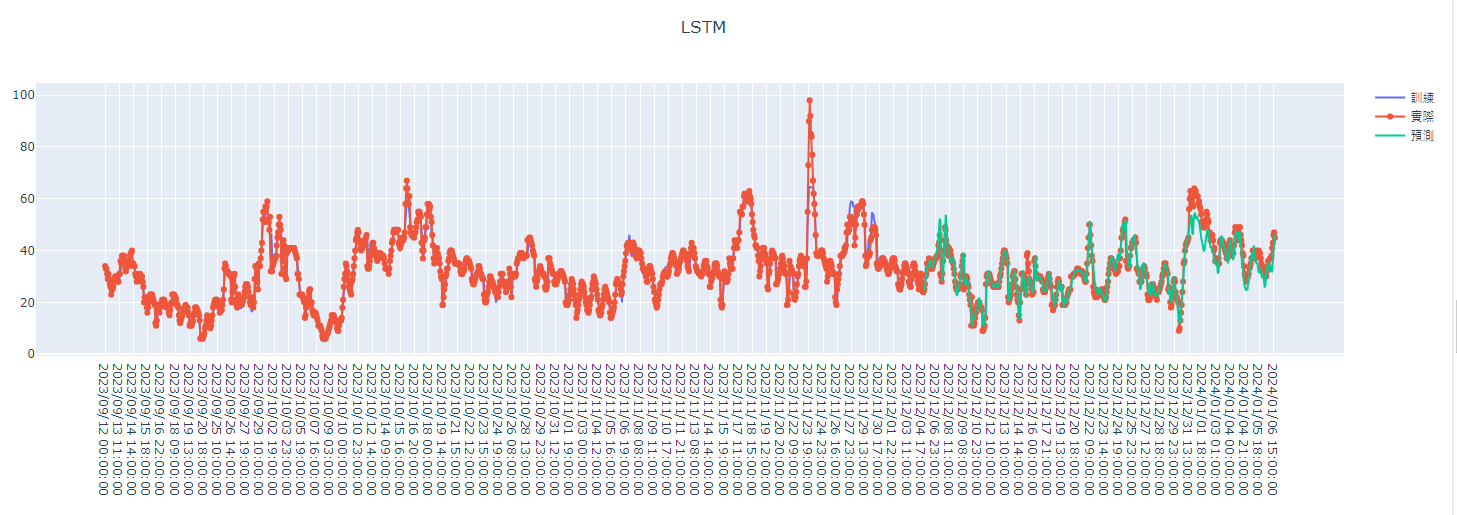


圖 27 台東數據跑LSTM之結果，綠色為模型預測測試集之結果

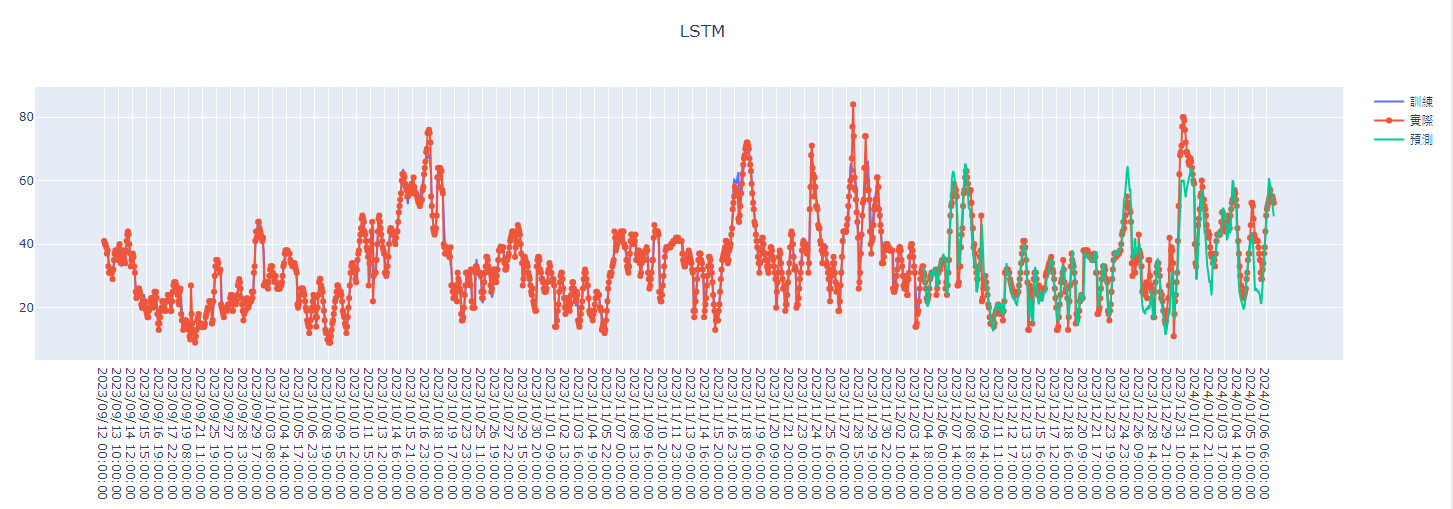


圖 28 花蓮數據跑LSTM之結果，綠色為模型預測測試集之結果

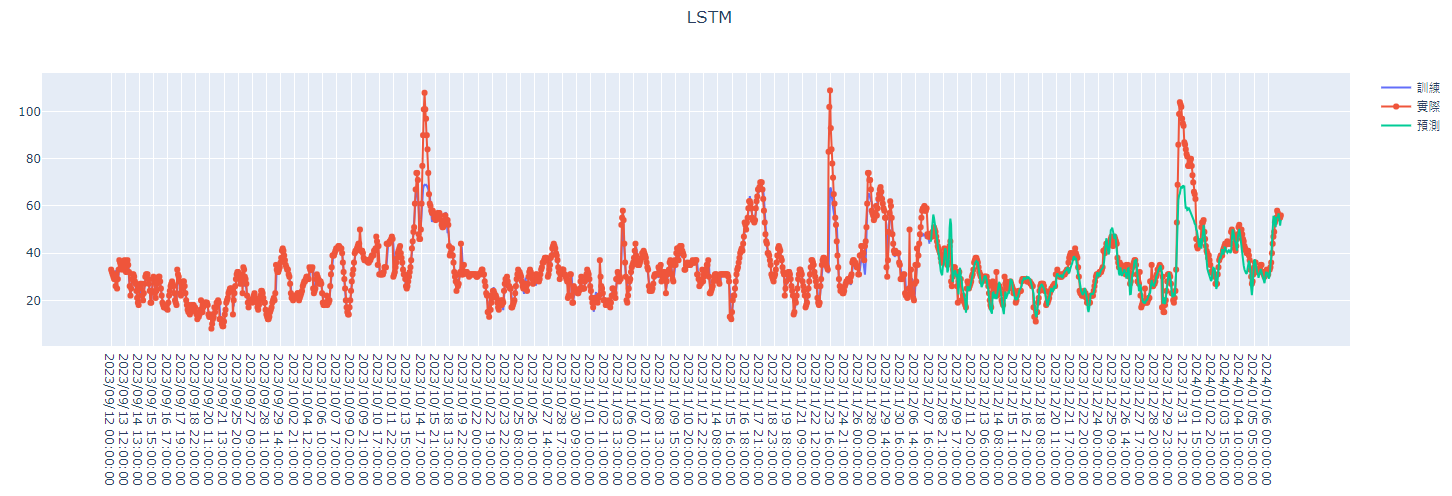


圖 29 冬山數據跑LSTM之結果，綠色為模型預測測試集之結果

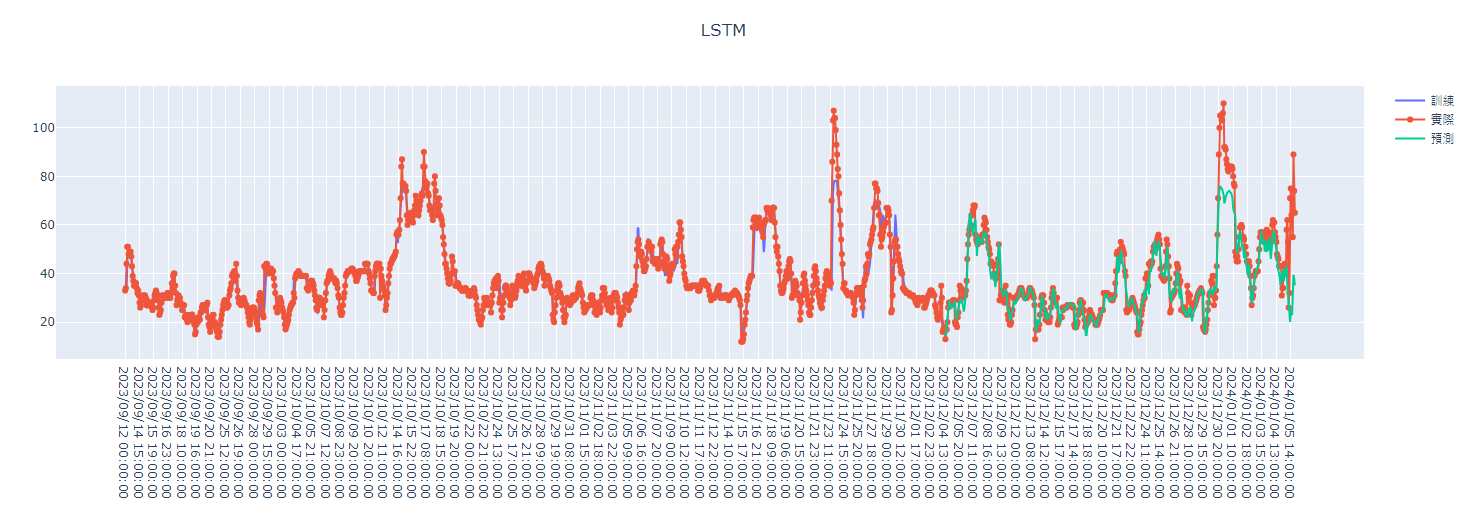


圖 30 宜蘭數據跑LSTM之結果，綠色為模型預測測試集之結果

**3-3 癌症與空氣品質關聯分析**

圖 31為不同地區癌症人數之統計表，再以地理數據做結合，對於高雄地區畫出不同地區期癌症人數，其中顏色越深表示人數越多，與空氣品質資料相比，可以發現深色地區其空氣品質相對也較差。程式碼如下

cancer\_death = pd.read\_csv('/content/drive/MyDrive/data/cancer105.csv')#讀入癌症人數

county\_code = pd.read\_csv('/content/drive/MyDrive/data/County\_Code\_100.csv')

#讀入不同縣市其代號

cancer\_death\_new = pd.merge(left=cancer\_death, right=county\_code, left\_on='county',right\_on='county')#癌症人數與縣市代號資料作合併

cancer\_code = pd.read\_csv('/content/drive/MyDrive/data/CancerCode\_97.csv')#讀入癌症代碼資料表

cancer\_code=cancer\_code.drop(['ICD-10'], axis=1)

cancer\_death\_new1=pd.merge(left=cancer\_death\_new, right=cancer\_code, left\_on='cause',right\_on='97年以後cause')

Kaohsiung\_cancer\_death=cancer\_death\_new1[(cancer\_death\_new1['city']=='高雄市')]#篩選出高雄之數據

town\_shp = gpd.read\_file('/content/drive/MyDrive/TW97 shp/TOWN\_MOI\_1081121.shp',encoding='utf-8') #呼叫台灣圖資

town\_shp = town\_shp.to\_crs(epsg=3826)

North\_town\_shp=town\_shp[(town\_shp['COUNTYNAME']=='高雄市')]

North\_town\_shp.head()

Kaohsiung\_cancer\_death\_M = Kaohsiung\_cancer\_death[Kaohsiung\_cancer\_death['sex']==1]

Kaohsiung\_cancer\_death\_M=Kaohsiung\_cancer\_death\_M.drop(['cause','county','sex','97年以後cause','city'],axis=1)

Kaohsiung\_Colorectal\_cancer\_M=Kaohsiung\_cancer\_death\_M[Kaohsiung\_cancer\_death\_M['死因分類']=='氣管、支氣管和肺癌']

Kaohsiung\_Colorectal\_cancer\_M=Kaohsiung\_Colorectal\_cancer\_M.drop(Kaohsiung\_Colorectal\_cancer\_M.columns[[0,1,5]], axis=1)

Kaohsiung\_Colorectal\_cancer\_M\_sum=Kaohsiung\_Colorectal\_cancer\_M.groupby('100年~鄉鎮市區').sum()

Kaohsiung\_Colorectal\_cancer\_M\_sum['District']=Kaohsiung\_Colorectal\_cancer\_M\_sum.index

Kaohsiung\_Colorectal\_cancer\_M\_sum['District']=Kaohsiung\_Colorectal\_cancer\_M\_sum['District'].apply(lambda t: t[3:])

Kaohsiung\_Colorectal\_cancer\_M\_sum.head()

Kaohsiung\_cancer\_death\_M[Kaohsiung\_cancer\_death\_M['死因分類'].str.contains("肺")]

Kaohsiung\_Colorectal\_cancer\_M\_sum.sort\_values(by='N',ascending=False)

left = North\_town\_shp.set\_index('TOWNNAME') #這個應該是 geopandas 的 GeoDataFrame 物件

right = Kaohsiung\_Colorectal\_cancer\_M\_sum.set\_index('District') #這應該是 pandas 的 DataFrame 物件

Kaohsiung\_ColorectalCancerDistrict = left.join(right)

Kaohsiung\_ColorectalCancerDistrict['N'] = Kaohsiung\_ColorectalCancerDistrict['N'].replace(np.nan, 0)#用取代的功能將nan補0

fig, ax = plt.subplots(1,1, figsize=(10,10)) # 定義畫布大小，其中(1,1)代表一行一列，即為一組座標

Kaohsiung\_ColorectalCancerDistrict.plot(column='N',cmap='Reds', ax=ax)#以N在地圖上以顏色標示，數字越大顏色越深

ax.set\_xlim([140000, 270000]) # 這行設定 X 範圍

ax.set\_ylim([ 2480000, 2600000]) # 這行設定 Y 範圍

plt.tight\_layout()#繪圖

****

圖 31 癌症人數資料

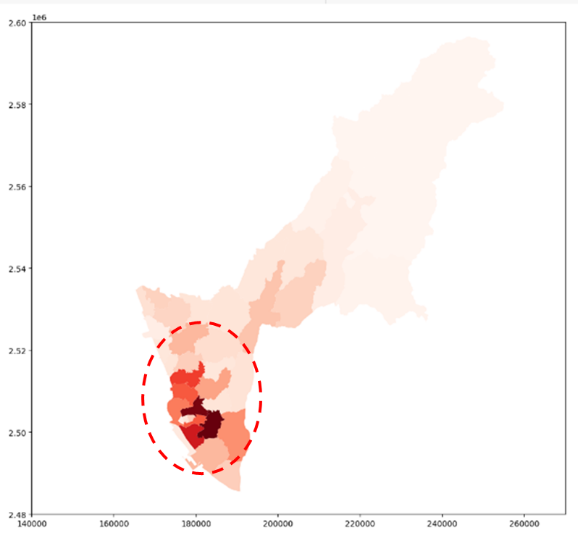
****

圖 32 高雄癌症人數之地圖分布

由圖 32可以看出，左營、林園、仁武、大寮這幾個地方其空氣各項汙染濃度皆比高雄其他高很多，推測可能是空氣品質越嚴重導致罹患肺癌人數越高。

本研究依照不同年份，將選出汙染程度高的五個地點左營、林園、仁武、琉球、大寮，畫癌症人口隨時間變化如圖 33，右側列出其斜率值，其程式碼如下

for index, i in enumerate(data2['area'].unique()):#data2['area']內放入左營、林園、仁武、#琉球、大寮

model = LinearRegression()#初始化線性模型

temp = data2[data2['area'] == i]

X = temp.iloc[:, [0, 4]]

y = temp.iloc[:, 1]

model.fit(X, y)#進行訓練

y\_pred = model.predict(X)#產出預測結果

temp['pred'] = y\_pred

rmse = np.sqrt(mean\_squared\_error(y, y\_pred))

rmse\_list.append(rmse)

print("RMSE for each area:", rmse\_list)

# Calculating MSE

mse = mean\_squared\_error(y, y\_pred)

mse\_list.append(mse)

# Calculating MAE

mae = mean\_absolute\_error(y, y\_pred)

mae\_list.append(mae)

# Calculating R-squared

ssr = np.sum((y\_pred - np.mean(y)) \*\* 2)

sst = np.sum((y - np.mean(y)) \*\* 2)

r\_squared = ssr / sst

r\_squared\_list.append(r\_squared)

print("R-squared for", i, ":", r\_squared)

# Calculating Adjusted R-squared

n = len(y)

p = X.shape[1]

adjusted\_r\_squared = 1 - (1 - r\_squared) \* (n - 1) / (n - p - 1)

adjusted\_r\_squared\_list.append(adjusted\_r\_squared)

print("Adjusted R-squared for", i, ":", adjusted\_r\_squared)

color = colors[index % len(colors)] # 使用不同顏色畫線

fig.add\_trace(go.Scatter(x=temp.year, y=temp['N'],

mode='lines+markers',

name=i + '測站' + '(\_real)',

line=dict(color=color)))

fig.add\_trace(go.Scatter(x=temp.year, y=temp['pred'],

mode='lines+markers',

name=i + '測站' + '(\_predict)',

line=dict(color=lighten\_color(color, 0.7))))

# 使用update\_layout更新圖例文字大小

fig.update\_layout(legend=dict(

font=dict(

family="Courier",

size=24, # 將圖例文字大小設為18

)

))

fig.update\_layout(

title={

'text': "民國87年到民國111年五個空氣汙染嚴重地區肺癌死亡人口數趨勢訓練圖",

'x': 0.5,

'y': 0.95,

'xanchor': 'center',

'yanchor': 'top',

'font': dict(

size=30

)

},

xaxis\_title={

'text': "年份",

'font': dict(

size=30 # 設定X軸標題文字大小為 18

)

},

yaxis\_title={

'text': "肺癌死亡人口數",

'font': dict(

size=30 # 設定Y軸標題文字大小為 18

)

}

)

fig.show()

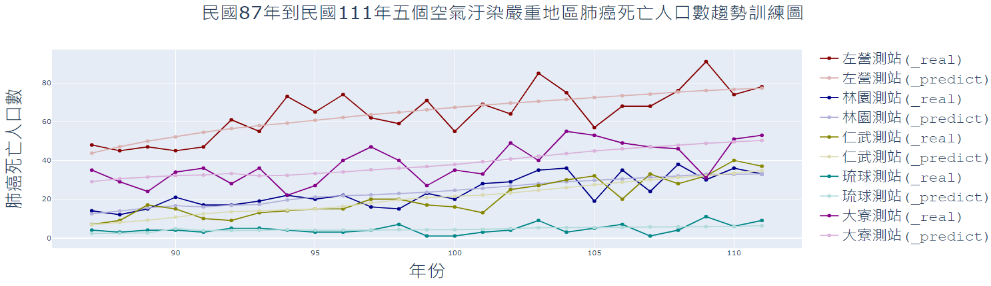
****

圖 33 癌症人口隨時間變化

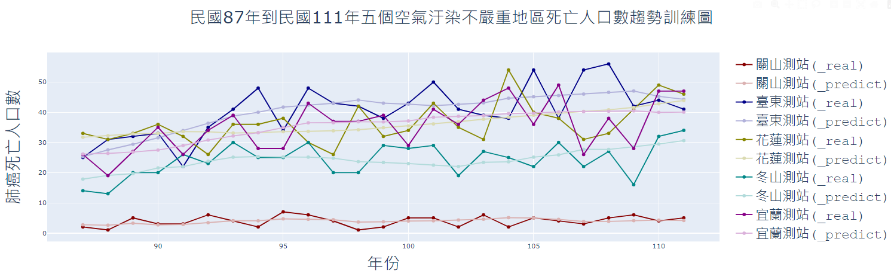
**** ****

圖 34 癌症人口隨時間變化

圖 34為關山、台東、花蓮、冬山、宜蘭之癌症人口隨時間變化圖

**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)