期末專題報告

巨量資料分析

期末專題報告 –
台灣鐵路人流預測

C109110146 呂彥鎒
C109110138 温翔旭
C109110135 趙韋誠

目錄

1 動機

2 目的

3 資料處理

4 實作方法

5 LSTM實作過程

6 實作成果

7 心得

8 參考資料

期末專題報告

01

動機

現況分析

- **歷史悠久**
  - 擁有241座車站
  - 車站環繞全台
- **運量龐大(民國110年)**
  - 載客1億5,493萬人次
  - 貨運噸數661萬公噸
- **近期議題**
  - 假期罷工(2022.05.01)
  - 摺疊門莒光號停駛(2022.06.29)
  - 復興號停駛(2022.12.27)

動機

台灣鐵路從1887年到現今已有一百多年的歷史了，現在每個縣市都可以看到火車站，全台的火車站數量達到了241座。
根據111年統計，110年的載客1億5,493萬人次，貨運噸數661萬公噸，這還是因為疫情的影響，導致載客人次較109年降低了23.9%，貨運噸數減少了8.9%。

近期又因為台鐵轉民營這個議題台鐵又有罷工舉動，如：五一勞動節，同時因為要節省開支所以將一些較為冷門的列車停駛。

所以我們希望利用巨量資料分析課堂所學來更進一步分析與預測台灣鐵路搭乘人次，希望可以用於降低營運(人力)成本與更加精準地進行班次排班。

02

目的

目的

01 資料視覺化

- 利用**資料視覺化**來觀察各車站進出人次
- 加上時間序列**以動畫的方式顯示**每日車站進出人次
- 以**熱力圖**統計整年全台車站總搭乘人次

02 長短期記憶模型(LSTM)預測人流量

- 蒐集2018~2021共1,461天的人流資料訓練模型
- 使用2022年共365天當作測試集
- 透過模型來進行**人流預測**

03 資料處理

資料處理(1/7) – 原始資料

下圖為政府資料開放平臺提供之台灣鐵路每日各站點進出站人次(原始資料,擷取部分)

trnOpDate	staCode	gateInComingCnt	gateOutGoingCnt
20220101	900	8645	8526
20220101	910	996	1243
20220101	920	1369	1711
20220101	930	3663	4535
20220101	940	1532	1845
20220101	950	1319	1343
20220101	960	7138	8228
20220101	970	4616	5277
20220101	980	9418	8806
20220101	990	12266	10553
20220101	1000	66442	48815
20220101	1010	3443	3330
20220101	1020	26366	23999
20220101	1030	1958	2059
20220101	1040	12337	14532
20220101	1050	1038	1170
20220101	1060	1405	1679
20220101	1070	7345	8169
20220101	1080	24991	31095
20220101	1090	6704	8271
20220101	1100	24577	28729
20220101	1110	4329	5086
20220101	1120	3601	4256
20220101	1130	1513	1870
20220101	1140	67	173
20220101	1150	465	548
20220101	1160	2946	3188
20220101	1170	5107	6152
20220101	1180	3733	4032
20220101	1190	1341	1552
20220101	1191	263	278
20220101	1192	1295	1431
20220101	1193	704	819
20220101	1194	3093	2345
20220101	1201	62	84
20220101	1202	147	125
20220101	1203	777	755
20220101	1204	97	96
20220101	1205	40	56
20220101	1206	299	448
20220101	1207	51	110
20220101	1208	1761	1554

▼原始資料欄位說明

欄位名稱	欄位說明
trnOpDate	乘車日期
staCode	車站代碼
gateInComingCnt	進站人次
gateOutGoingCnt	出站人次

為了方便後續處理,我們將進站與出站人次分別建立csv檔

資料處理(2/7) – VLOOKUP抓取資料

而建立3D 地圖要使用資料時，若直接使用地名去偵測，部分資料會誤判到其他具有相同地名的錯誤位置，所以我們最後使用經緯度去定位正確位置。

另staCode是車站的編號，若要轉換為車站名稱或經緯度，手動取代的速度過慢，所以我們使用**VLOOKUP**來完成，詳細步驟如下：

- Step1：將車站基本資料集.json轉為.csv檔。

▼原始資料欄位說明

欄位名稱	欄位說明
stationCode	車站代碼
stationName	中文站名
stationEName	英文站名
name	網站中文站名
ename	網站英文站名
stationAddrTw	車站地址
stationAddrEn	車站英文地址
stationTel	車站電話

資料處理(3/7) – VLOOKUP抓取資料

➢ Step2：先刪除不必要的欄位(英文站名、網站中文站名、網站英文站名、車站地址、車站英文地址、車站電話)，再將車站基本資料集複製到每日各站點進出站人次的資料中。

trnOpDate	staCode	gateInComingCnt	gateOutGoingCnt		stationCode	stationName	stationCode	gps
20220101	900	8645	8526		900	基隆	900	25.13411 121.73997
20220101	910	996	1243		910	三坑	910	25.12305 121.74202
20220101	920	1369	1711		920	八堵	920	25.10843 121.72898
20220101	930	3663	4535		930	七堵	930	25.09294 121.71415
20220101	940	1532	1845		940	百福	940	25.07803 121.69373
20220101	950	1319	1343		950	五堵	950	25.07784 121.66764
20220101	960	7138	8228		960	汐止	960	25.06894 121.66244
20220101	970	4616	5277		970	汐科	970	25.06261 121.64659
20220101	980	9418	8806		980	南港	980	25.05348 121.60706
20220101	990	12266	10553		990	松山	990	25.04936 121.57807
20220101	1000	66442	48815		1000	臺北	1000	25.04771 121.51784
20220101	1010	3443	3330		1001	臺北-環島	1001	25.04811 121.51793
20220101	1020	26366	23999		1010	萬華	1010	25.03348 121.49984
20220101	1030	1958	2059		1020	板橋	1020	25.01469 121.46352

資料處理(4/7)

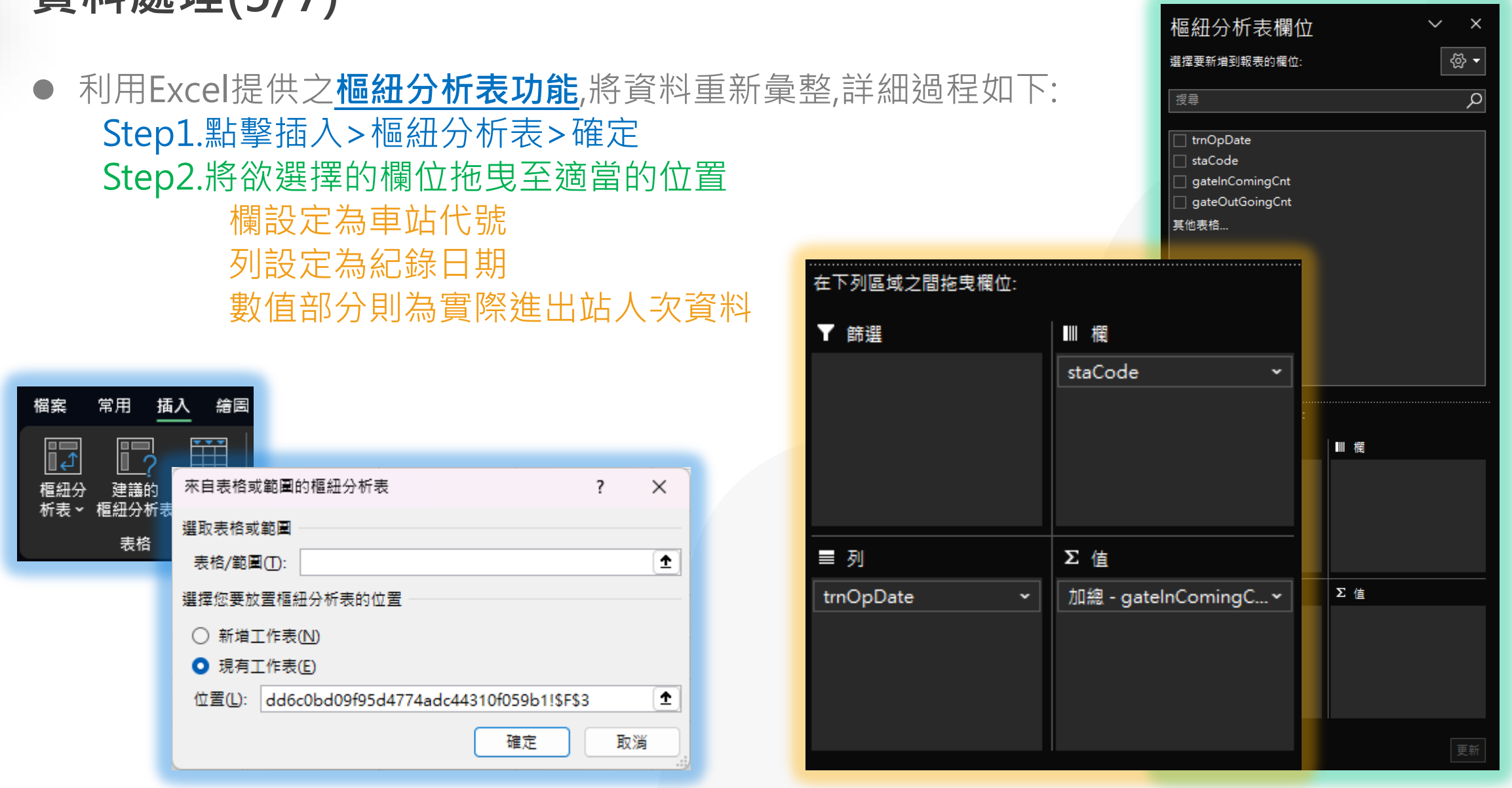
➢ Step3：在staCode後面插入兩個column分別為stationName(地區)、location(經緯度)並在stationName的第一筆使用VLOOKUP公式，最後將設定值套用到C所有的欄位，location也是一樣，差別在於選取的資料表為J、K兩行。

- **VLOOKUP(欲查詢變數, 資料表範圍, 行數, TRUE『不完全相同』/FALSE『完全相同』)**

	A	B	C	D	E	F	G	H	I	J	K
1	trnOpDate	staCode	stationName	location	gateInComingCnt	gateOutGoingCnt		stationCod	stationNam	stationCode	gps
2	20220101	900	FALSE)		8645	8526		900	基隆	900	25.13411 121.73997
3	20220101	910	三坑		996	1243		910	三坑	910	25.12305 121.74202
4	20220101	920	八堵		1369	1711		920	八堵	920	25.10843 121.72898
5	20220101	930	七堵		3663	4535		930	七堵	930	25.09294 121.71415
6	20220101	940	百福		1532	1845		940	百福	940	25.07803 121.69373
7	20220101	950	五堵		1319	1343		950	五堵	950	25.07784 121.66764
8	20220101	960	汐止		7138	8228		960	汐止	960	25.06894 121.66244
9	20220101	970	汐科		4616	5277		970	汐科	970	25.06261 121.64659
10	20220101	980	南港		9418	8806		980	南港	980	25.05348 121.60706
11	20220101	990	松山		12266	10553		990	松山	990	25.04936 121.57807
12	20220101	1000	臺北		66442	48815		1000	臺北	1000	25.04771 121.51784
13	20220101	1010	萬華		3443	3330		1001	臺北-環島	1001	25.04811 121.51793
14	20220101	1020	板橋		26366	23999		1010	萬華	1010	25.03348 121.49984
15	20220101	1030	浮洲		1958	2059		1020	板橋	1020	25.01469 121.46352
16	20220101	1040	樹林		12337	14532		1030	浮洲	1030	25.00419 121.44477
17	20220101	1050	南樹林		1038	1170		1040	樹林	1040	24.99121 121.42481

資料處理(5/7)

- 利用Excel提供之**樞紐分析表功能**,將資料重新彙整,詳細過程如下:
  - Step1.點擊插入>樞紐分析表>確定
  - Step2.將欲選擇的欄位拖曳至適當的位置
    - 欄設定為車站代號
    - 列設定為紀錄日期
    - 數值部分則為實際進出站人次資料



資料處理(6/7)

Step3.資料內的車站代號以實際車站名稱取代,下圖為最終完成之結果 (部分擷取)
利用相同手法分別將進站人次與出站人次分別建立一份csv檔案。

	基隆	三坑	八堵	七堵	百福	五堵	汐止	汐科	南港	松山	臺北	萬華	板橋	浮洲	樹林	南樹林	山佳	鶯歌	桃園	內壢	中壢	埔心	楊梅	富岡
20220101	8645	996	1369	3663	1532	1319	7138	4616	9418	12266	66442	3443	26366	1958	12337	1038	1405	7345	24991	6704	24577	4329	3601	1513
20220102	6744	1011	1494	3691	1526	1134	6320	4123	8348	9334	47310	2886	21477	1812	11987	816	1378	6509	25728	6703	25927	4402	4264	1642
20220103	6677	1337	2271	6042	2549	2153	9674	11506	11145	15665	46642	4296	19246	3039	13315	1104	1595	7845	22729	8557	20276	5932	4587	1607
20220104	6481	1319	2216	5848	2483	2186	9797	11778	10820	15353	43776	4026	17327	3027	12329	1089	1565	7688	21015	8279	17874	5622	4311	1444
20220105	6772	1375	2260	5958	2510	2193	9910	11773	10928	15466	44969	4140	17746	3013	12219	1134	1600	7763	21336	8227	18341	5829	4335	1473
20220106	6522	1306	2271	5867	2533	2147	9762	11687	10898	15301	45509	3927	17571	3021	12440	1122	1486	7597	21026	8086	17773	5341	4222	1397
20220107	7457	1348	2443	6203	2727	2281	10449	12141	12208	16741	57177	4228	21232	3411	13709	1213	1539	8158	24377	9165	22319	5872	5296	1509
20220108	5948	978	1447	3766	1872	1407	7540	4797	8224	10328	44323	2989	17617	2297	11032	880	1450	6030	19368	5990	17609	4190	3486	1228
20220109	5509	924	1304	3342	1433	1111	6255	3904	7534	8226	39394	2821	16753	1736	10160	971	1277	5438	21099	5695	20084	3723	3406	1379
20220110	6433	1366	2189	5954	2546	2174	9588	11636	10742	15318	44981	4009	17773	2968	12683	1128	1478	7723	20888	8018	17987	5495	4302	1548
20220111	6021	1306	2098	5723	2456	2210	9516	11660	10684	14678	43044	3696	16690	2971	12077	1064	1421	7328	19532	7506	15814	5160	3856	1304
20220112	6275	1314	2210	5633	2535	2190	9556	11631	10520	14723	42964	3875	16610	2953	12243	1074	1424	7357	19406	7612	15828	5195	3852	1380
20220113	6259	1260	2190	5765	2570	2111	9622	11520	10619	14815	44210	3770	17119	2855	11975	1042	1376	7314	19328	7435	15669	4993	3846	1305
20220114	7321	1364	2407	6164	2685	2271	10203	12073	11700	16630	55293	4147	20811	3231	13369	1137	1498	7984	22411	8346	19204	5368	4817	1403
20220115	5914	865	1444	3610	1734	1358	7240	4651	7006	9989	41659	2635	16462	2239	9881	872	1232	5237	15931	4857	13478	3140	2651	972
20220116	5239	945	1294	3300	1404	1141	5838	3689	6813	8017	37270	2553	15678	1786	9627	754	1144	5021	16991	4687	14259	3040	2665	1021
20220117	5964	1247	2096	5726	2446	2070	9143	11153	10452	15088	42972	3779	17630	2601	11813	1017	1375	7039	18306	6730	13730	3796	3392	1099
20220118	5972	1238	2085	5634	2419	2028	9373	11489	10244	14266	41161	3635	16047	2509	11277	992	1331	6757	17438	6615	12962	4648	3405	1150
20220119	6223	1289	2102	5786	2441	2060	9486	11413	10392	14417	42296	3719	16166	2462	11416	1018	1401	7056	17326	6591	12999	4176	3322	1150
20220120	6791	1278	2157	5813	2559	2162	10004	11779	10937	14741	44598	3825	16993	2810	12357	1062	1404	7459	17544	6604	13038	3341	3296	971
20220121	6619	1224	1861	5252	2378	1990	9248	11559	10518	15161	48987	3335	18442	2570	11822	938	1296	6510	18020	6018	13134	3189	2908	907
20220122	5866	1104	1546	4142	2041	1670	7924	6983	8797	11103	41259	2768	15416	2274	10132	809	1122	5465	14876	4704	11233	2427	2355	781
20220123	4619	779	1212	2921	1315	1072	5576	3442	6368	7055	33874	2207	13285	1659	8328	651	957	4101	12491	3808	10244	2177	1988	780
20220124	5934	1180	1875	5025	2225	1953	8616	10817	9601	13852	42622	3150	15558	2164	11079	911	1274	6171	15976	5316	11593	2927	2658	934
20220125	6094	1224	1844	4962	2312	1949	8782	10920	9571	13788	42260	3322	15463	2241	11169	916	1310	6356	16246	5328	11547	2872	2603	921
20220126	6221	1180	1884	4906	2336	1946	9099	10951	9573	13711	42936	3392	15774	2288	11214	921	1271	6494	16234	5535	11960	3017	2753	897
20220127	6571	1230	2093	5129	2239	1984	8866	10684	9416	14200	45522	3448	16409	2265	11138	941	1244	6228	16039	5340	12112	2821	2671	870
20220128	6379	1215	1812	5352	2280	2006	8949	10785	9939	14975	54007	3338	19723	2129	12506	916	1274	6122	18566	5503	14491	3185	3218	1001
20220129	4274	764	1063	2980	1191	1046	5652	3629	6020	8899	39857	2089	16859	1410	9511	637	887	3801	13325	3585	11174	2176	1771	736
20220130	3290	645	972	2475	944	758	4350	2803	4824	7681	33832	1582	15174	1122	9131	544	809	3023	12593	3037	10300	1888	1449	727
20220131	3283	627	1068	2386	827	741	3804	2516	4096	6676	28623	1951	14667	1030	9121	573	808	3047	12769	3529	11080	2264	1714	921

資料處理(7/7) – 問題

在上述資料處理的過程中，我們發現了以下兩點問題：

01

部分人流資料遺失

我們推測可能的原因有:

該站為**簡易站**(僅派站員未派站長之車站)
或**招呼站**(僅設有候車月台而無站員)

02

部分車站資料遺失

我們推測可能的原因有:

該站**已廢站**或站務人員撤離(沒有站務員統計該站人流數據)

04

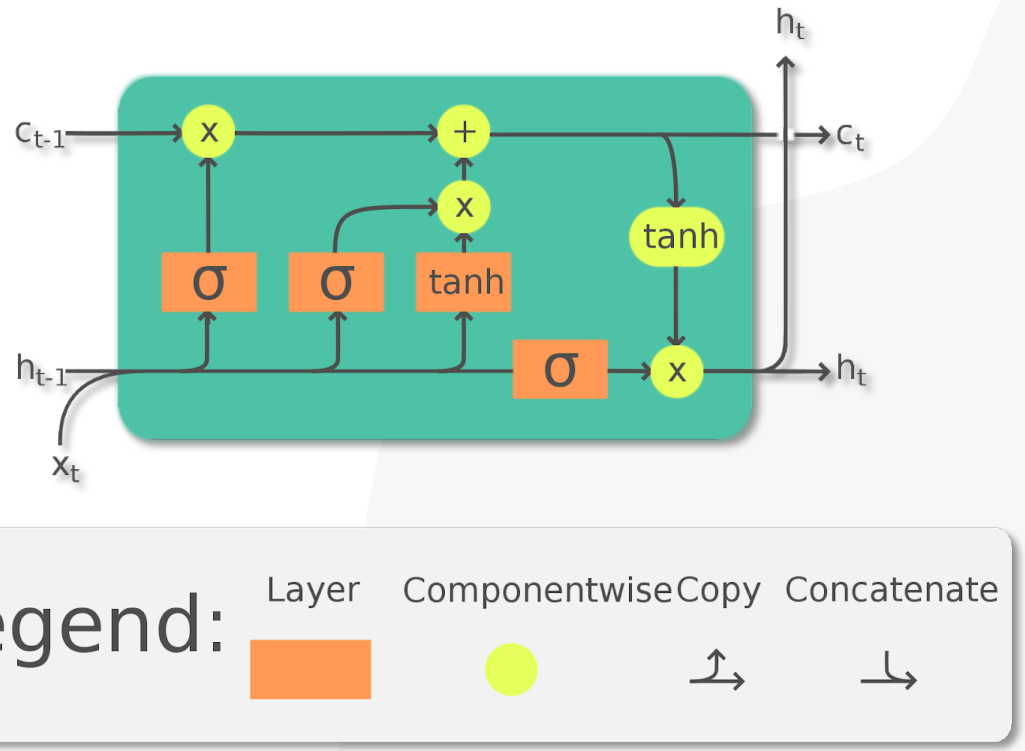
實作方法-LSTM

LSTM實作方法(1/2)

- ✓ 使用政府資料開放平台的每日各站點進出站人次作為訓練資料
- ✓ 透過**Tensorflow2.0搭建單變量LSTM模型**進行預測
- ✓ 以2018年-2021年每天進站人次作為訓練集，切分2022年作為測試集，以單一車站預測人流
- ✓ 本次報告以台北車站為例進行實作

LSTM實作方法(2/2)

- ✓ LSTM(Long Short-Term Memory)是一種循環神經網路模型，主要由輸入門(Input Gate)，遺忘門(Forget Gate)，輸出門(Output Gate)，記憶單元(Memory Cell)組成，可透過遺忘門判斷從上個節點輸入的資訊是否捨棄，以此減少不重要的資料在長期遞歸下對模型的影響，並透過記憶單元儲存需要的資訊，最後將其輸出，因此LSTM可以用於時間序列的預測。



圖片來自於維基百科[參考資料6]

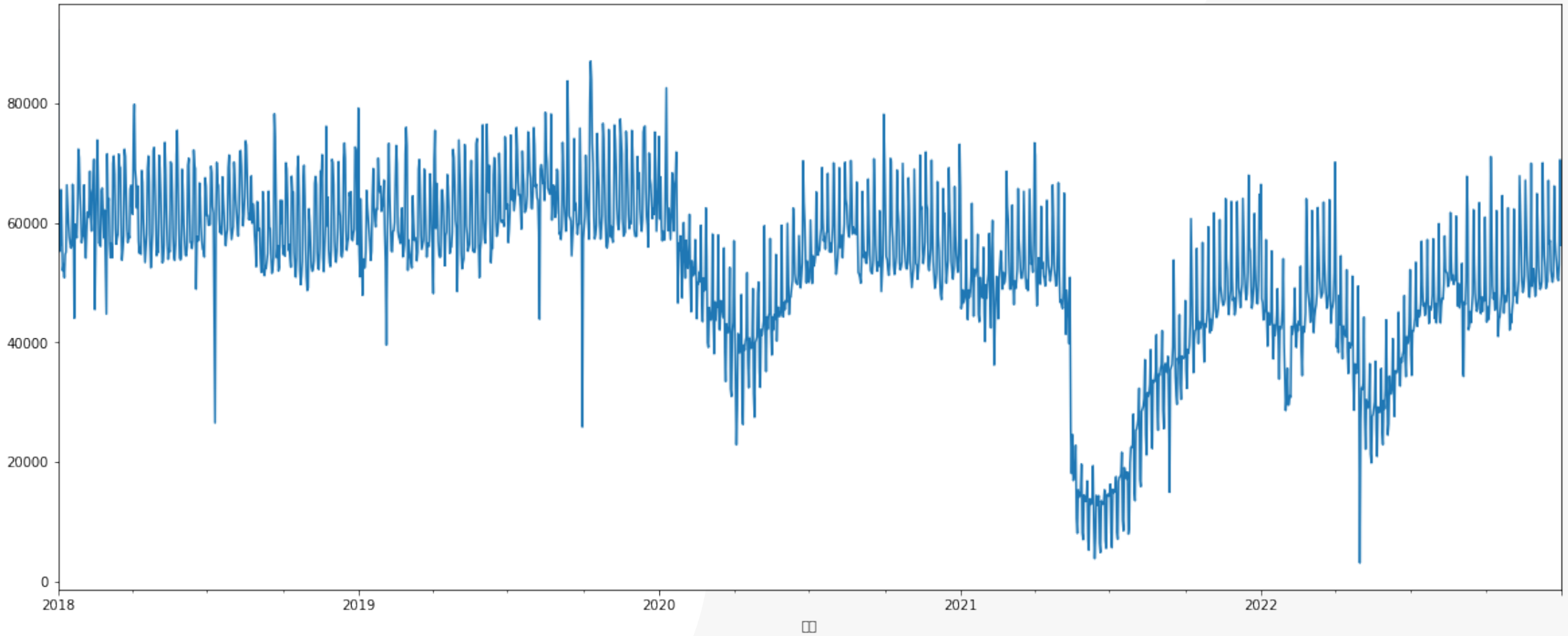
05

LSTM實作過程

LSTM實作過程(1/8) – 導入數據

✓ 首先導入數據，並將人流數據進行可視化顯示從2018年到2022年的人流變化。

▼歸一化前數據(以台北車站為例)

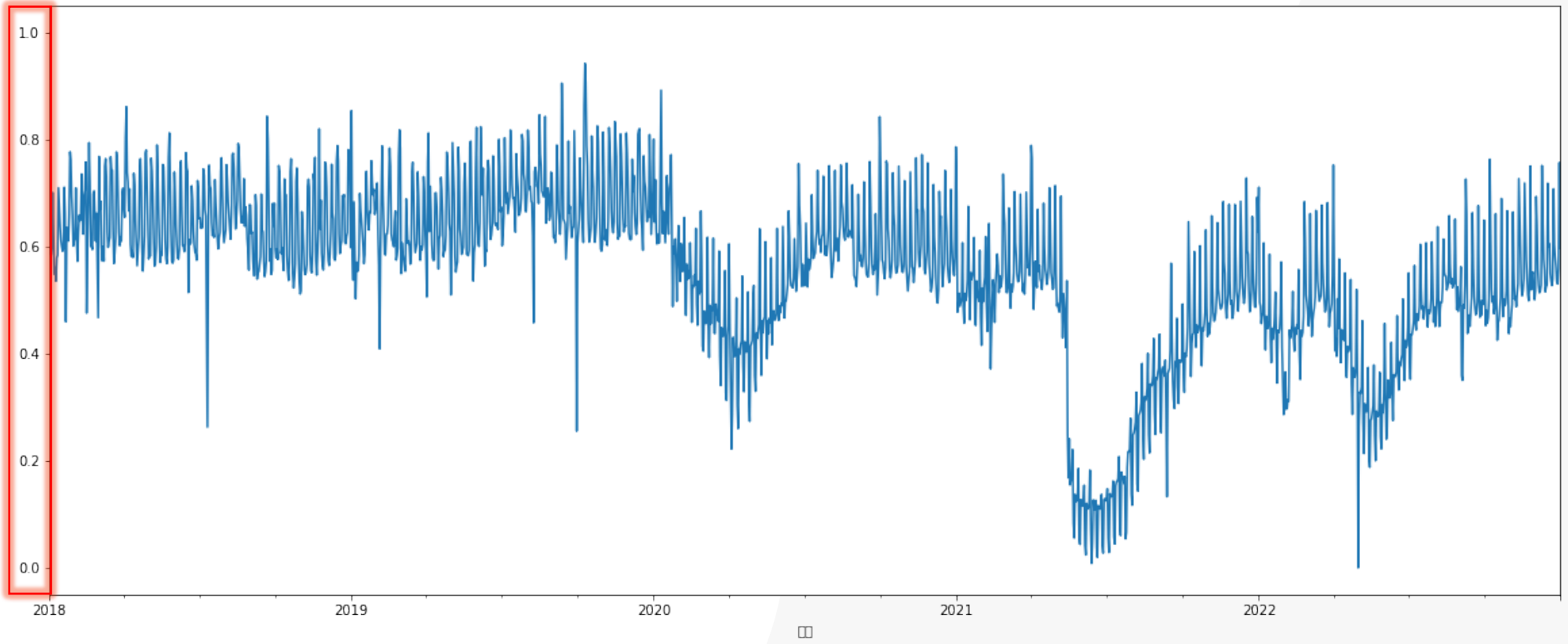


LSTM實作過程(2/8) – 數據歸一化

✓ 將數據歸一化，以便後續模型訓練。

```
# 數據歸一化
scaler = MinMaxScaler()
train['進站人次'] = scaler.fit_transform(train_df['進站人次'].values.reshape(-1,1))
```

▼歸一化後數據



LSTM實作過程(3/8) – 建立特徵數據

✓ 接著將資料建立特徵數據集與標籤集，Timesteps設定7天為一組。

```
# 建立特徵數據集與標籤集
def create_dataset(dataset, seq_len = 7):
    X = []
    y = []
    start = 0
    end = dataset.shape[0] - seq_len
    for i in range(start, end):
        sample = dataset[i : i+seq_len] #創建樣本，如i=0則以0:7為樣本
        label = dataset[i+seq_len]      #創建標籤，如i=0則將第8筆數據做為標籤
        X.append(sample)                #保存樣本數據
        y.append(label)                 #保存標籤數據

# 將特徵集與標籤集以numpy array格式返回
return np.array(X), np.array(y)

# 將x,y分別定義為數據集與標籤集，時間以7天做切分
x,y = create_dataset(train.values,seq_len=7)
```

LSTM實作過程(4/8) – 訓練與測試集

✓ 建立完成後再分為訓練與測試集。

```
# 將建立好的特徵數據與標籤集切分為訓練集與測試集
def split_dataset(X, y, train_ratio=0.8):

    X_len = len(X)
    train_data_len = int(X_len * train_ratio)

    X_train = X[:train_data_len]
    y_train = y[:train_data_len]

    X_test = X[train_data_len:]
    y_test = y[train_data_len:]

    return X_train, X_test, y_train, y_test
```

LSTM實作過程(5/8) – TensorFlow

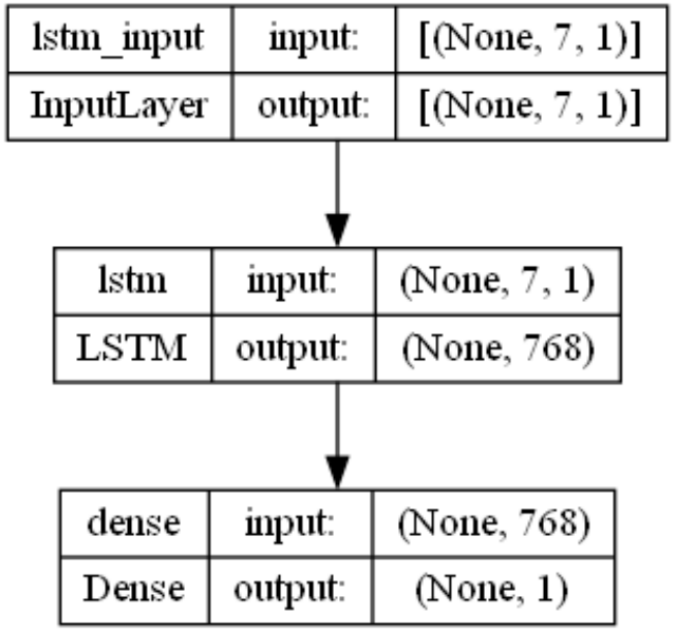
✓ 建立batch_dataset，將資料封裝為tensorflow能接受的資料類型。

```
# 將X_train, X_test, y_train, y_test建立batch dataset
def create_batch(X, y, batch_size=128, data_type=1):

    # data_type區分,測試集為1,訓練集為2
    if data_type == 1:

        #將x,y封裝為tensor類型
        dataset = tf.data.Dataset.from_tensor_slices((tf.constant(X), tf.constant(y)))
        test_batch_data = dataset.batch(batch_size)

        return test_batch_data
    else:
        dataset = tf.data.Dataset.from_tensor_slices((tf.constant(X), tf.constant(y)))
        train_batch_data = dataset.cache().shuffle(1000).batch(batch_size)

        return train_batch_data
```

LSTM實作過程(6/8) – LSTM神經元

✓ 建構模型，定義LSTM神經元，資料輸入，並選用目前較常用的Adam作為優化器，mse(mean_squared_error)作為損失函數。

▼顯示模型結構

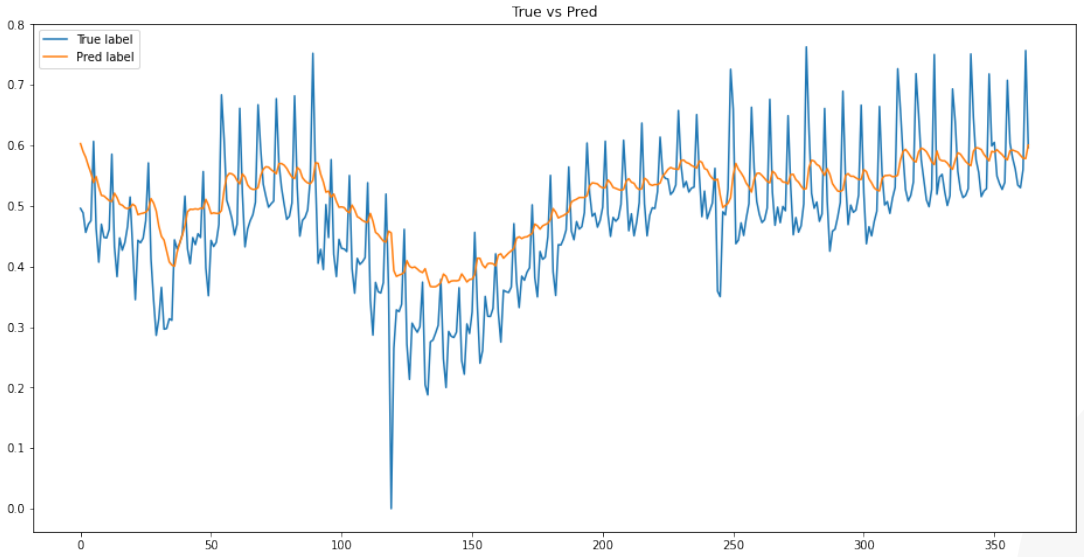



```
# 模型構建
model = Sequential()
model.add(LSTM(768,input_shape=(7,1)))
model.add(Dense(1))
model.compile(optimizer='adam',loss = 'mse’)

# 顯示模型結構
plot_model(model,to_file='model.png',show_shapes=True)
```

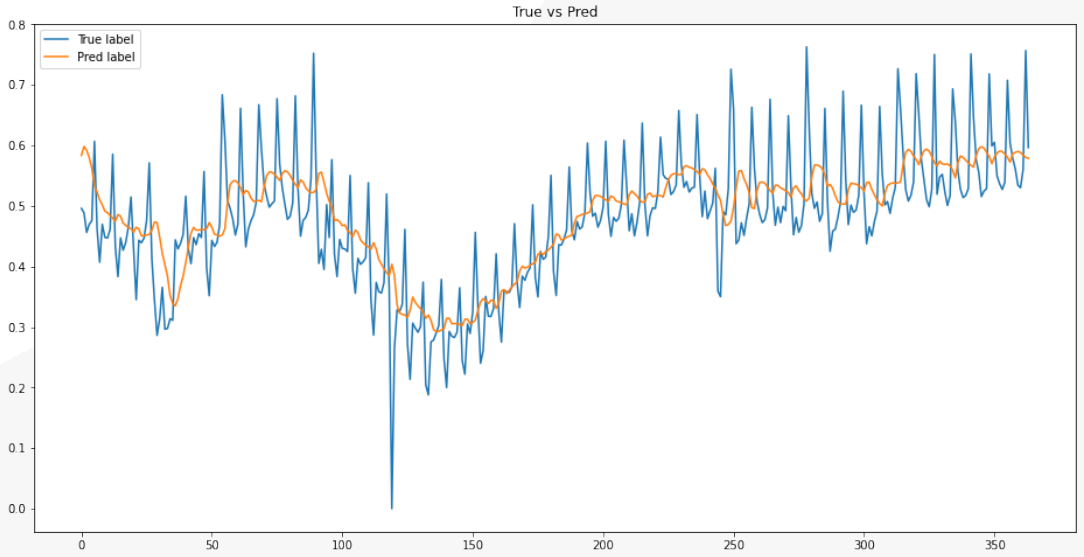
LSTM實作過程(7/8) – 參數設定

✓ 訓練時可以發現，在LSTM神經元數量較多的情況下，預測結果與實際結果會更加吻合。

```
# 模型構建
model.add(LSTM(8,input_shape=(7,1)))
```



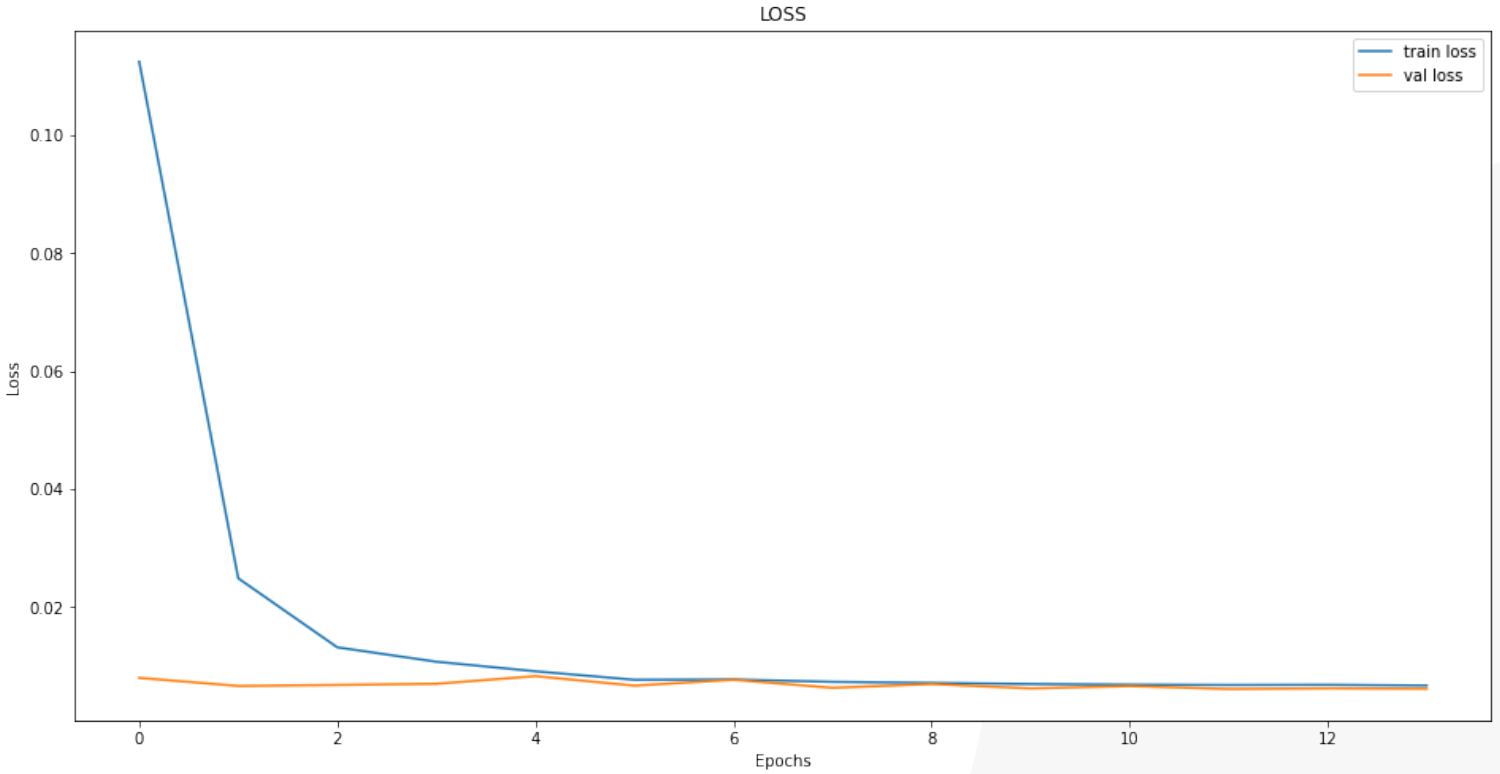

```
# 模型構建
model.add(LSTM(768,input_shape=(7,1)))
```



LSTM實作過程(8/8) – loss圖形

✓ 模型建構好後進行模型訓練，訓練完成後顯示loss圖形，利用train_loss及val_loss判斷訓練是否過擬合。

```
# 模型訓練
history = model.fit(train_batch_dataset, epochs=12, validation_data=test_batch_dataset,
          callbacks=[checkpoint_callback])
```

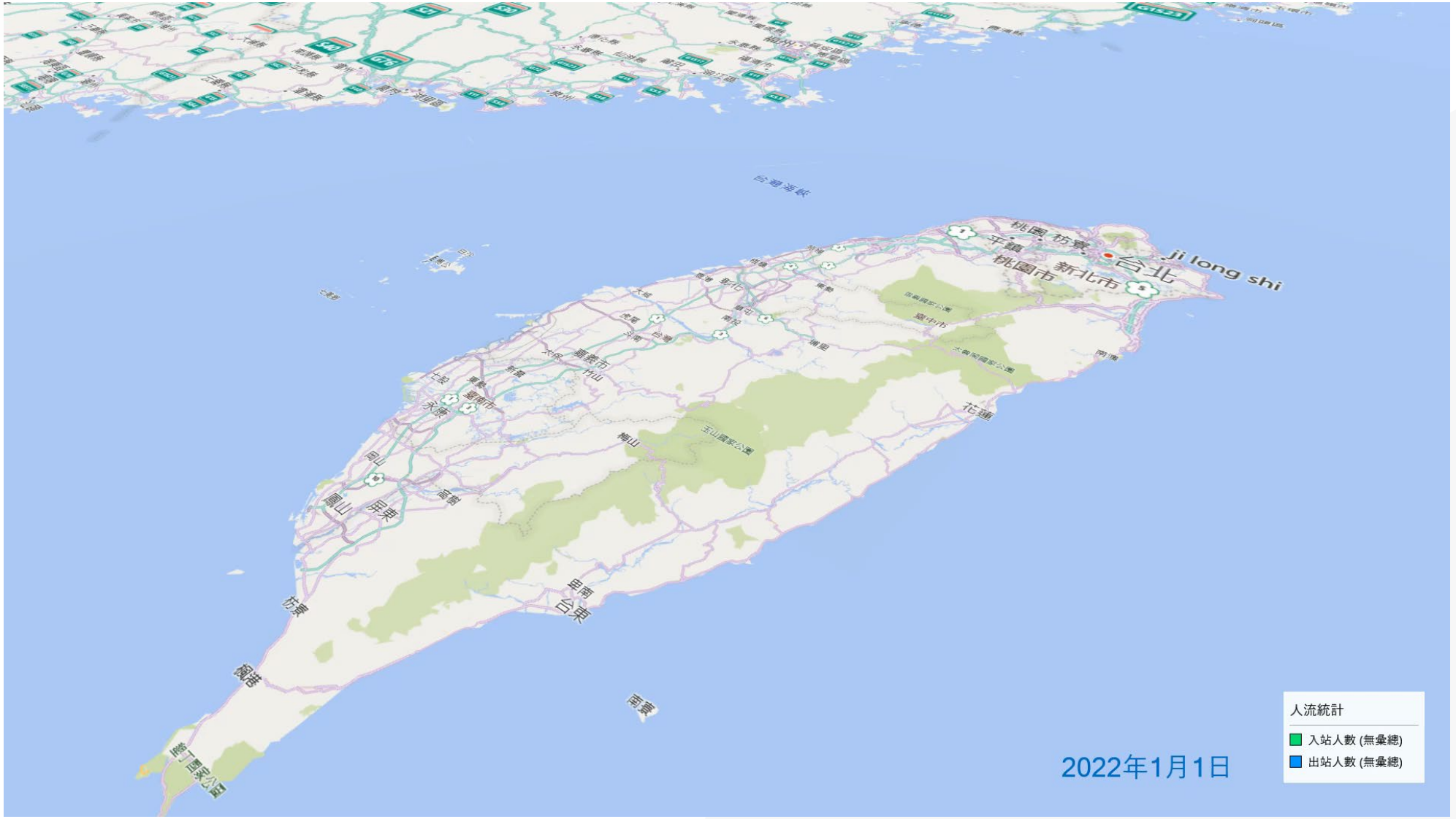


▼訓練迭代過程

```
Epoch 1/12 6/6 [==============] - 0s 27ms/step - loss: 0.0066 - val_loss: 0.0060
Epoch 2/12 6/6 [==============] - 0s 21ms/step - loss: 0.0065 - val_loss: 0.0060
Epoch 3/12 6/6 [==============] - 1s 205ms/step - loss: 0.0065 - val_loss: 0.0060
Epoch 4/12 6/6 [==============] - 0s 22ms/step - loss: 0.0064 - val_loss: 0.0059
Epoch 5/12 6/6 [==============] - 0s 29ms/step - loss: 0.0064 - val_loss: 0.0059
Epoch 6/12 6/6 [==============] - 0s 20ms/step - loss: 0.0064 - val_loss: 0.0058
Epoch 7/12 6/6 [==============] - 0s 19ms/step - loss: 0.0063 - val_loss: 0.0058
Epoch 8/12 6/6 [==============] - 0s 21ms/step - loss: 0.0063 - val_loss: 0.0057
Epoch 9/12 6/6 [==============] - 0s 23ms/step - loss: 0.0063 - val_loss: 0.0057
Epoch 10/12 6/6 [==============] - 0s 21ms/step - loss: 0.0062 - val_loss: 0.0059
Epoch 11/12 6/6 [==============] - 0s 22ms/step - loss: 0.0062 - val_loss: 0.0056
Epoch 12/12 6/6 [==============] - 0s 21ms/step - loss: 0.0061 - val_loss: 0.0055
```

06 實作成果

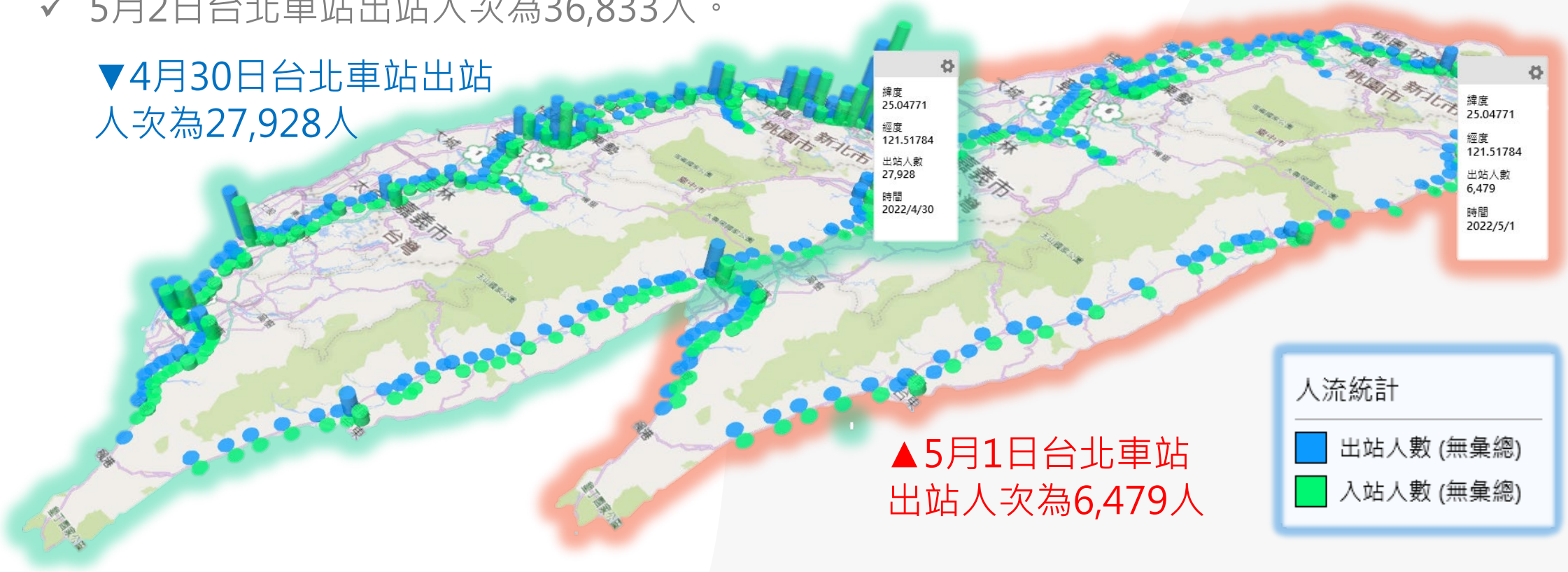
實作成果(1/4) - 搭乘人次視覺化



實作成果(2/4) – 從視覺化資訊找出問題

- ✓ 5月1日的人次明顯比前後兩天少了很多，透過這個現象我們尋找當天有什麼事件，搜尋到的結果為：
  因為台鐵工會對於台鐵公司化議題有不同的立場，所以發起「五一勞動節不加班」，所以導致當日搭乘台鐵的人次驟減。
- ✓ 5月2日台北車站出站人次為36,833人。

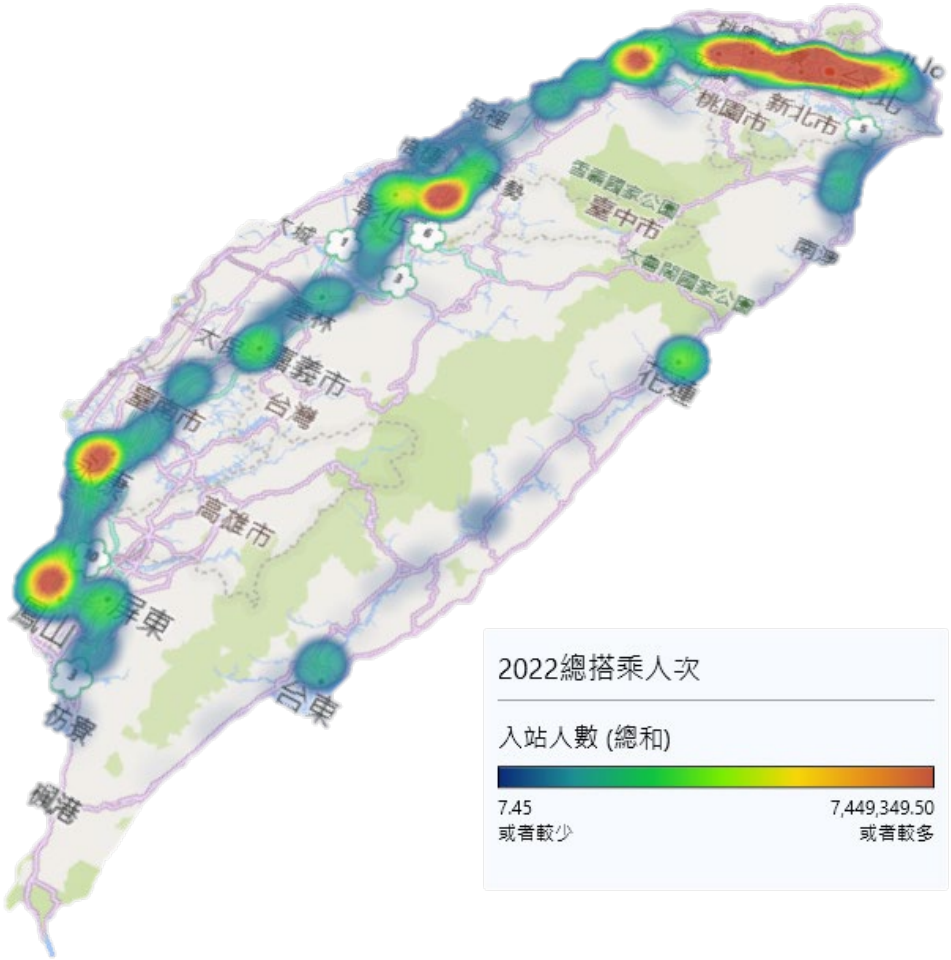
▼4月30日台北車站出站人次為27,928人

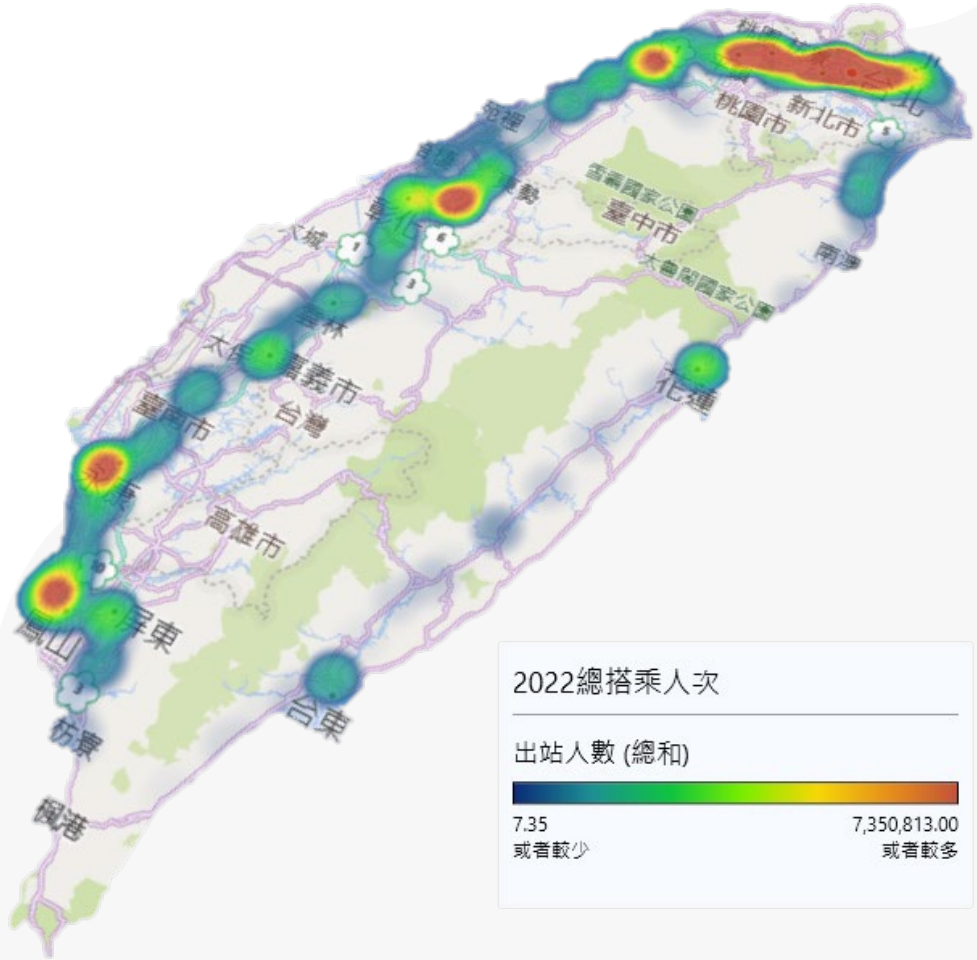


▲5月1日台北車站出站人次為6,479人

實作成果(3/4) – 熱力圖

- ✓ 這兩張圖分別為2022年總入站人次以及總出站人次，從這張圖可以明顯看到**台北**、**桃園**、**台中**、**高雄**、**屏東**這幾個地區搭乘人次眾多，而東部則是**花蓮**、**台東**較多。

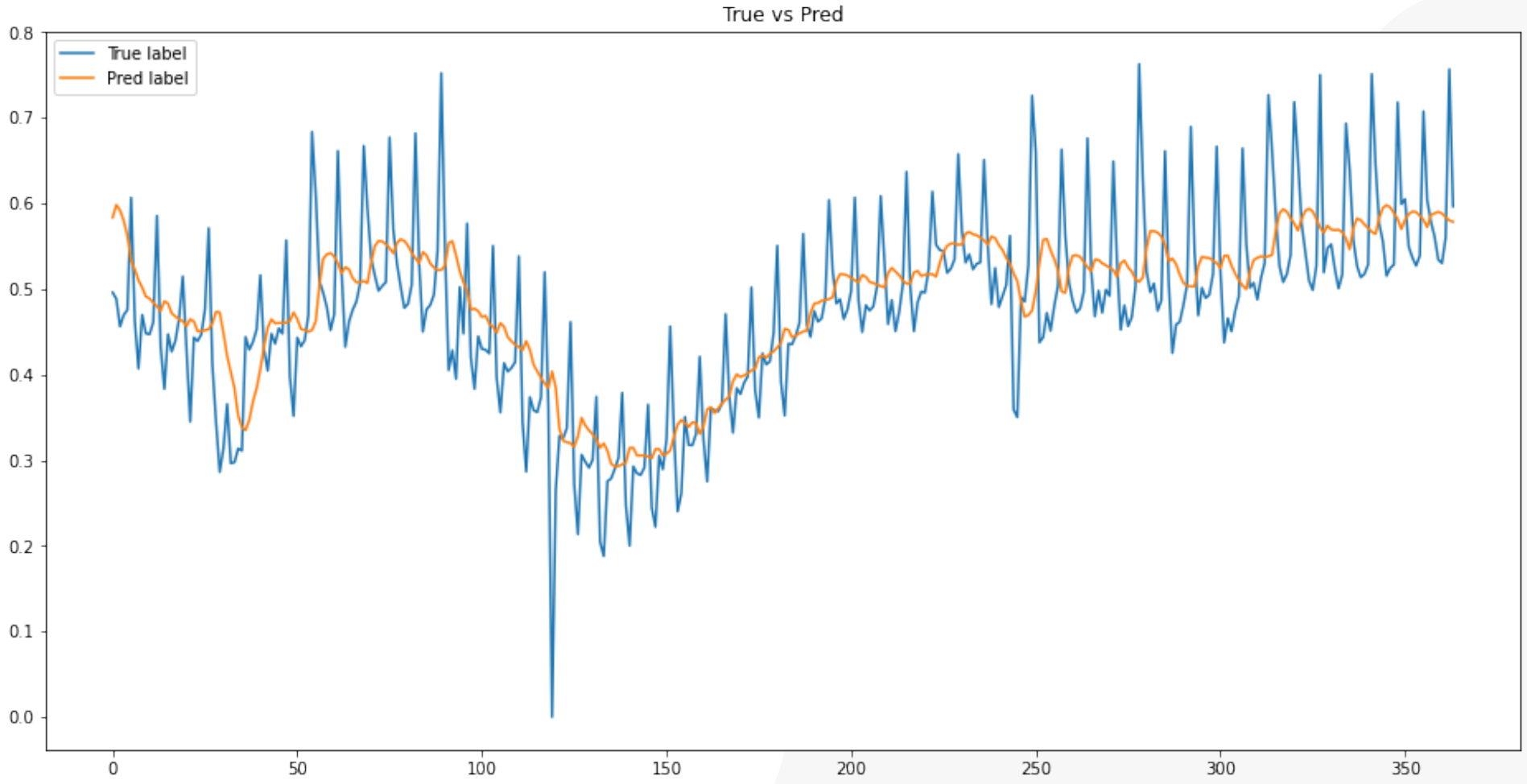




實作成果(4/4) – LSTM預測搭乘人次

✓ 最終顯示真實資料與預測結果圖形，預測出搭乘人次，藍線為**實際值**，橘色為**預測值**。

▼下圖顯示了2022年共365天，每天的人流量預測結果與實際結果對比



07 心得

心得

最初在處理資料時會不知道如何下手，因為第一次處理這麼大量的資料，且原始的檔案、車站是使用車站編號，以至於無法直接使用。

一開始嘗試使用土法煉鋼的方法，直接複製車站名稱，但在貼到第一天的資料時我們發現資料對不上。(如下圖)

238	20220101	7362	猴硐		367	468
239	20220101	7380	瑞芳		442	482
240	20220101	7390	海科館		278	357
241	20220102	900	八斗子		6744	6897
242	20220102	910	四腳亭		1011	1007
243	20220102	920	暖暖		1494	1421

我們當時猜測可能是有些車站已經沒有在使用了，所以將多出來的那幾筆車站刪除，最後第一天的資料都對上了，但在將已經對上的資料繼續往下貼時又發現問題了，新的問題是這整個資料集不會把出入人次為0的資料寫入裡面，所以導致有些資料缺失，所以我們只能另尋他法。

在網路上搜尋過後，最後確定使用VLOOKUP函示，它可以一次滿足我們的問題，至於實作資料視覺化是因為在【救命大數據】中，人口數量即是以資料視覺化的方式呈現的。

08 參考資料

參考資料

1. 交通部公共運輸整合資訊流通服務平臺. (2013, April 19). 臺灣鐵路營運路線圖. 臺灣鐵路營運路線圖. https://ptx.transportdata.tw/PTX/
2. 行政院. *台鐵 (國情簡介-交通運輸)*. 行政院 國情簡介. https://www.ey.gov.tw/state/A44E5E33CDA7E738/1f34ef94-2ff5-40ab-b526-e2fedd358a36
3. 交通部臺灣鐵路管理局. (2021, June 27). 車站基本資料集. 車站基本資料集 | 政府資料開放平臺. https://data.gov.tw/dataset/33425
4. 交通部臺灣鐵路管理局. (2023, January 6). 每日各站點進出站人次. 每日各站點進出站人數 | 政府資料開放平臺. https://data.gov.tw/dataset/8792
5. 維基百科. (2022, January 20). *分類:台灣鐵路廢站 - 維基百科*. 維基百科. https://zh.wikipedia.org/wiki/Category:台灣鐵路廢站
6. *Long Short-Term Memory - Wikipedia*. (2022, December 8). Long Short-Term Memory. https://en.wikipedia.org/wiki/Long_short-term_memory
7. 唐国梁. (2021, June 16). TensorFlow 2.0基於LSTM單變量預測_電力消耗案例. https://www.bilibili.com/video/BV1f5411K7qD/?spm_id_from=333.1007.top_right_bar_window_history.content.click&vd_source=f317c61eb30bfb5accf016154d324788

20230113

謝　謝　觀　看

期末專題報告 –
台灣鐵路人流預測