冰水主機

黃世豪

目錄

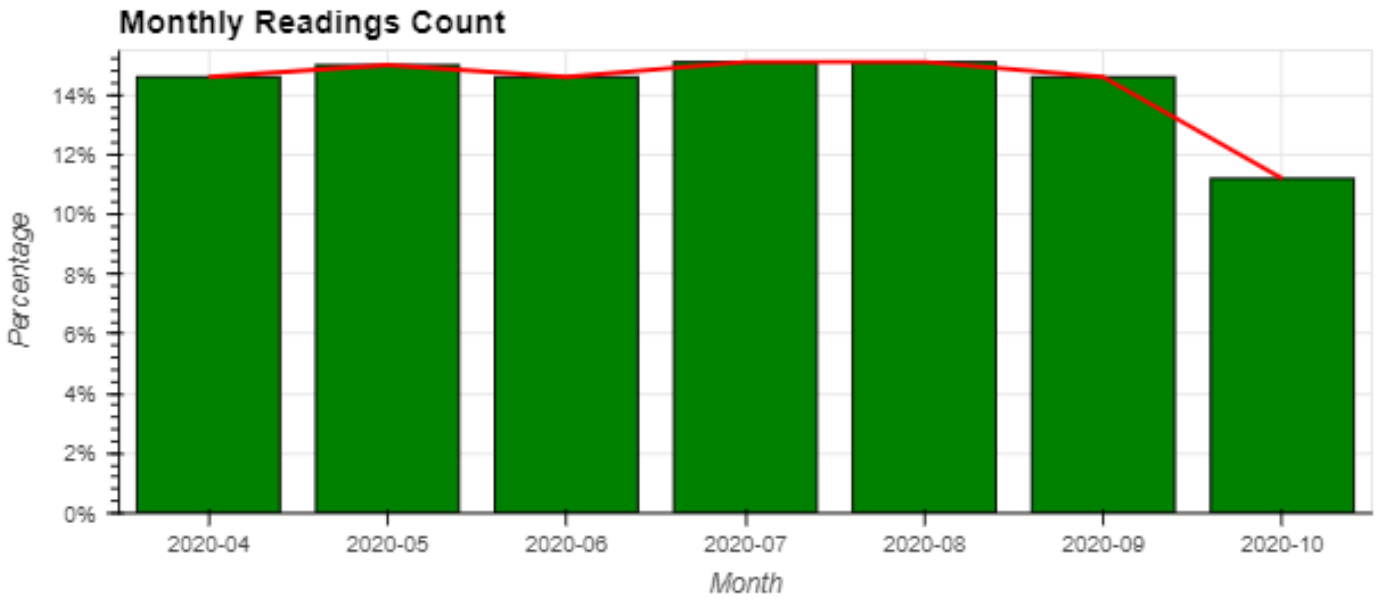
數據圖表

- 呈現冰水主機 CHS_TT11, CHR_TT12, CHR_TT15 在 2020.04.01 ~ 2020.10.23 數據表現的情形

Unnamed: 0	machine	timestamp	avgvalue	lastvalue	year	month	day	weekday	weekofyear	hour	minute	season	timing	daily
0	CHS_TT11	2020-04-01 00:00:00	5.825000	5.825000	2020	4	1	Wednesday	14	0	0	Spring	Night	2020-04-01
1	CHR_TT15	2020-04-01 00:00:00	8.600000	8.600000	2020	4	1	Wednesday	14	0	0	Spring	Night	2020-04-01
2	CHR_TT12	2020-04-01 00:00:00	11.812501	11.812501	2020	4	1	Wednesday	14	0	0	Spring	Night	2020-04-01
3	CHS_TT11	2020-04-01 00:00:02	5.837500	5.837500	2020	4	1	Wednesday	14	0	0	Spring	Night	2020-04-01
4	CHR_TT15	2020-04-01 00:00:02	8.550000	8.550000	2020	4	1	Wednesday	14	0	0	Spring	Night	2020-04-01

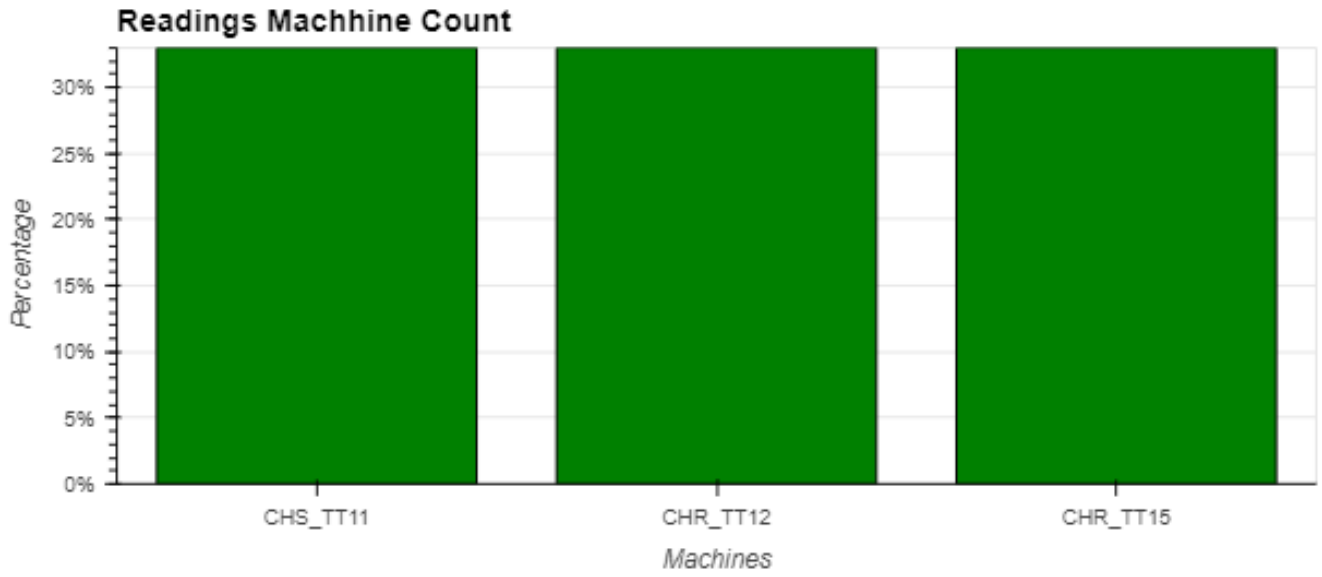
數據圖表

•統計每個月收集的，資料點，因為每個月的天數不同而有起伏，且十月只記錄到23號，因此較少



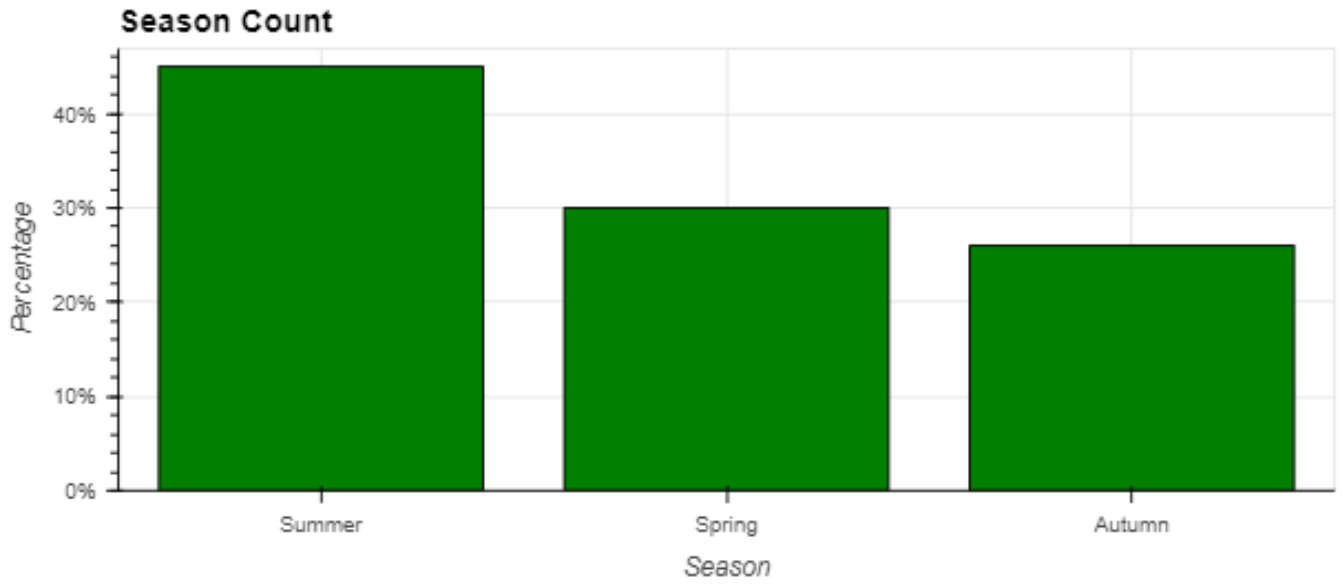
數據圖表

- 統計三台機器數據點比例，可以發現各暫 1/3 ，並無分別



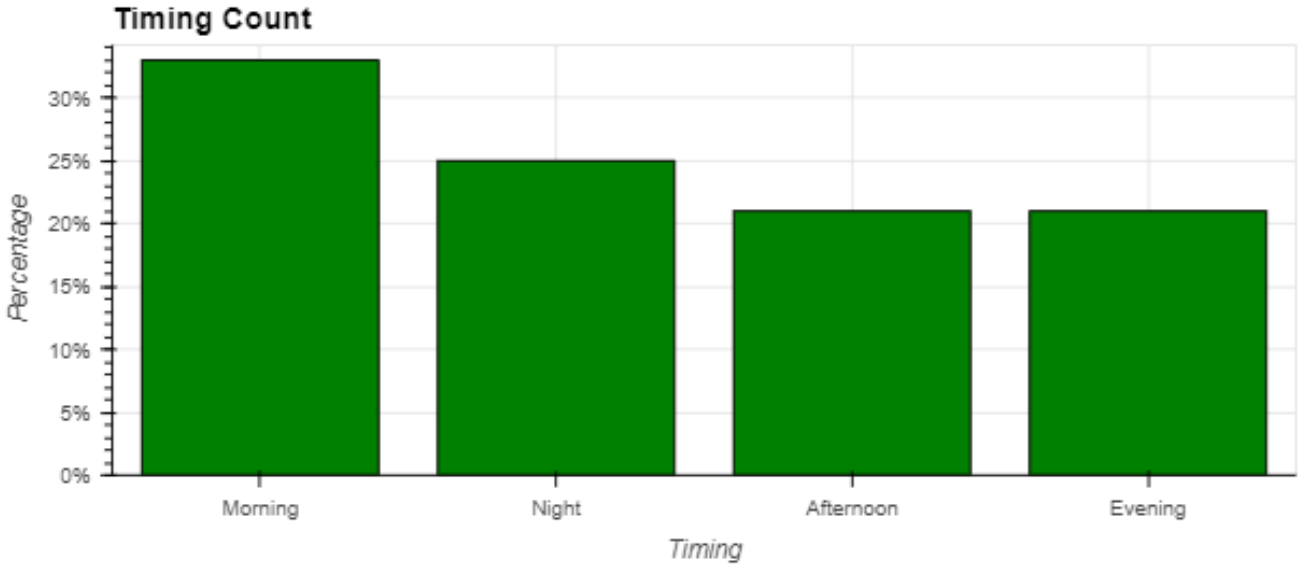
數據圖表

- 不同季節下收集的點，可以看到夏天佔的跨度比較大，因此收集的點比較多



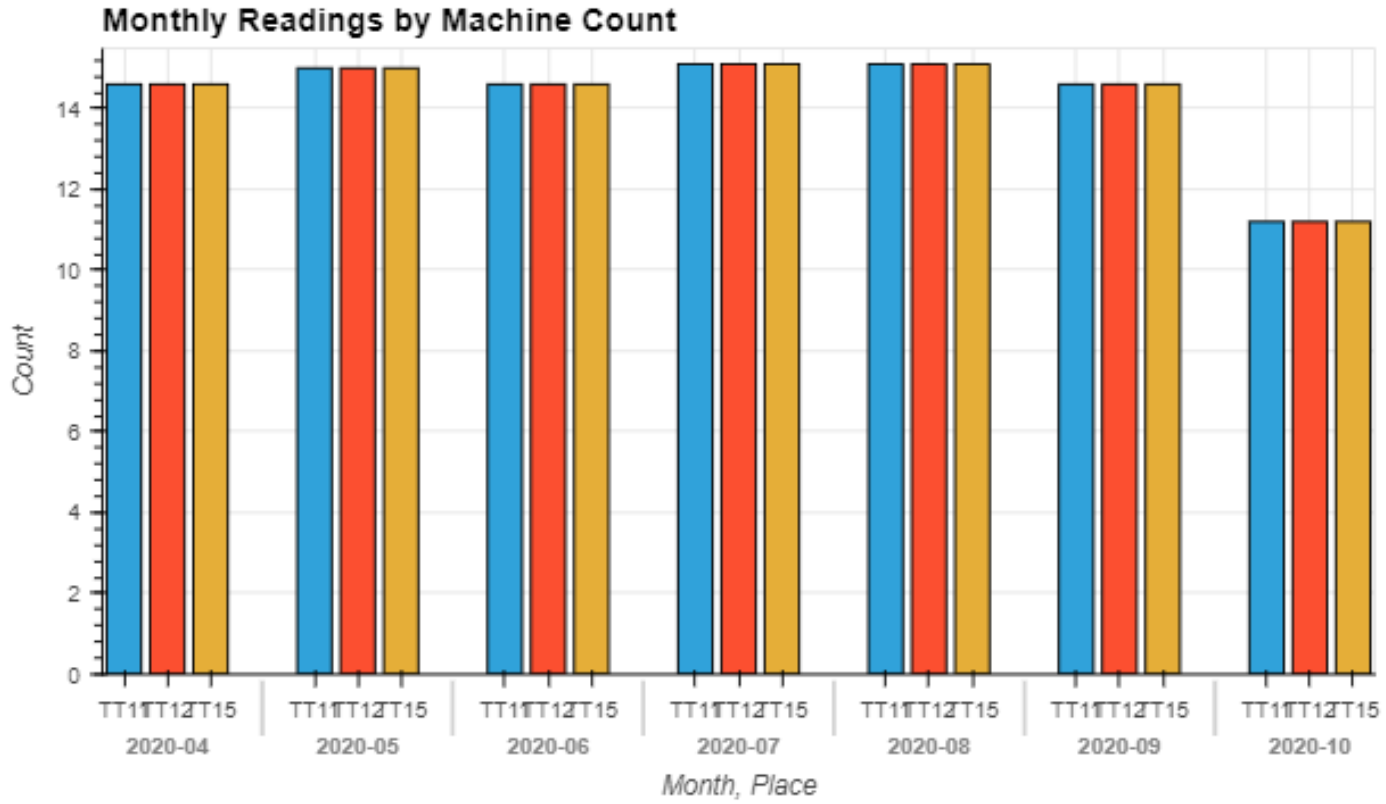
數據圖表

- 不同時間段收集的點，因為它的點是每2秒採集一次的，而我設的白天跨度又比較大，因此收集的點也比較多



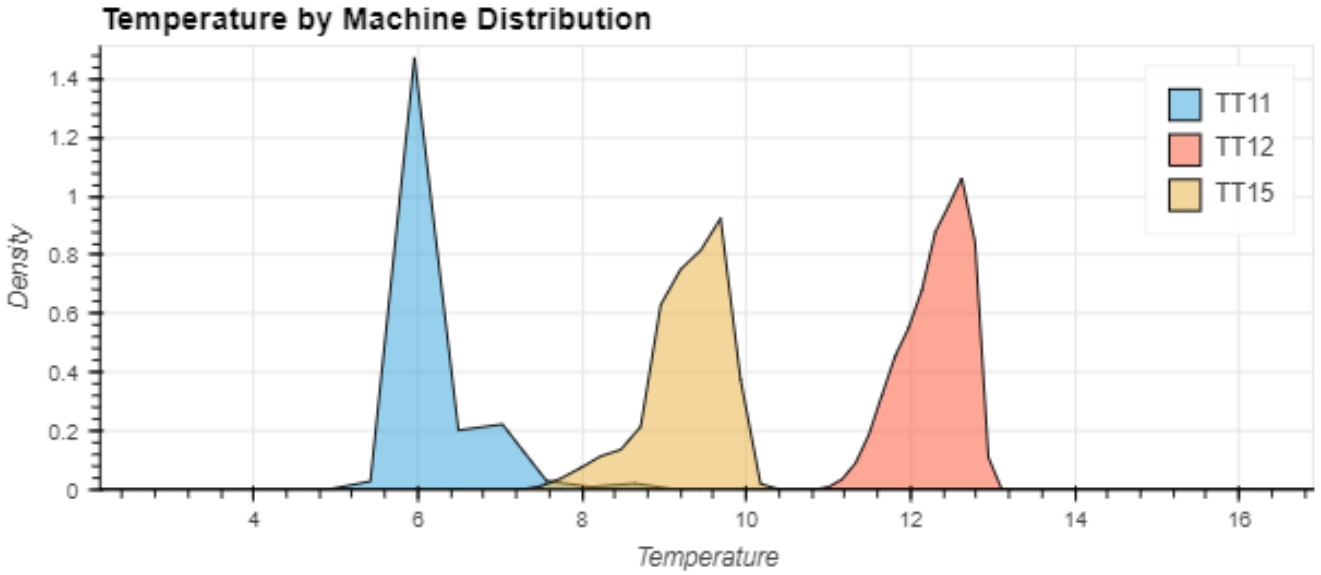
數據圖表

- 查看不同月份下，不同機器收集的數據點數量，看起來並無不同



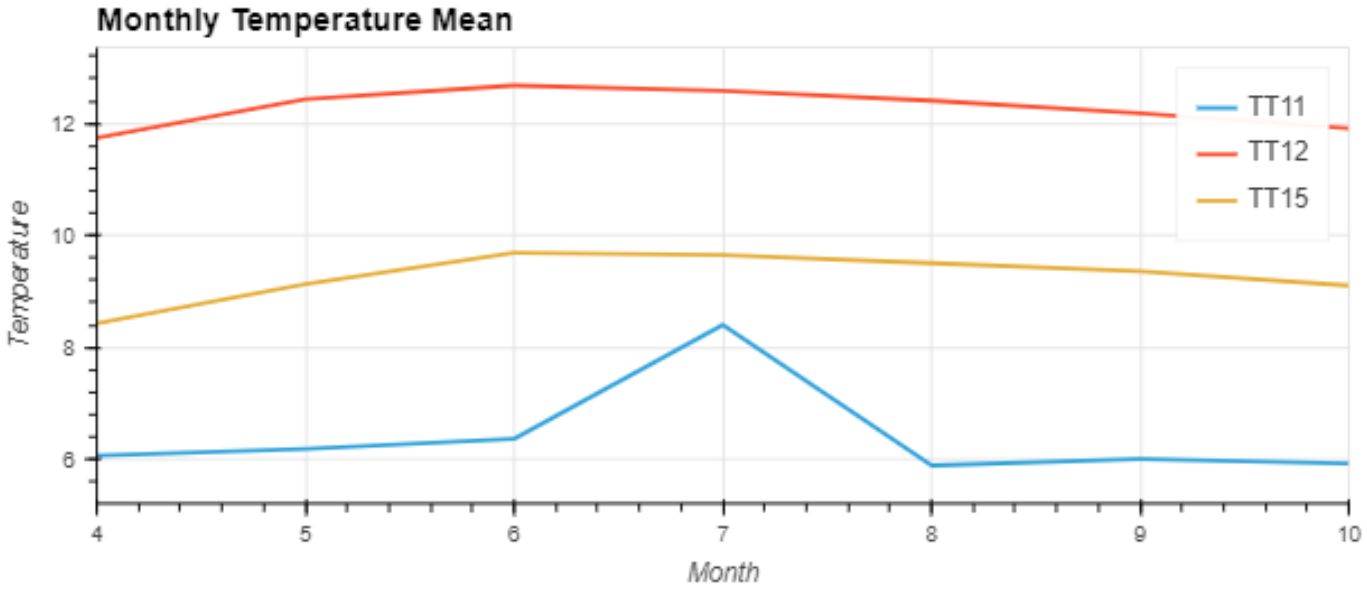
數據圖表

- 不同機器所佔的數據範圍



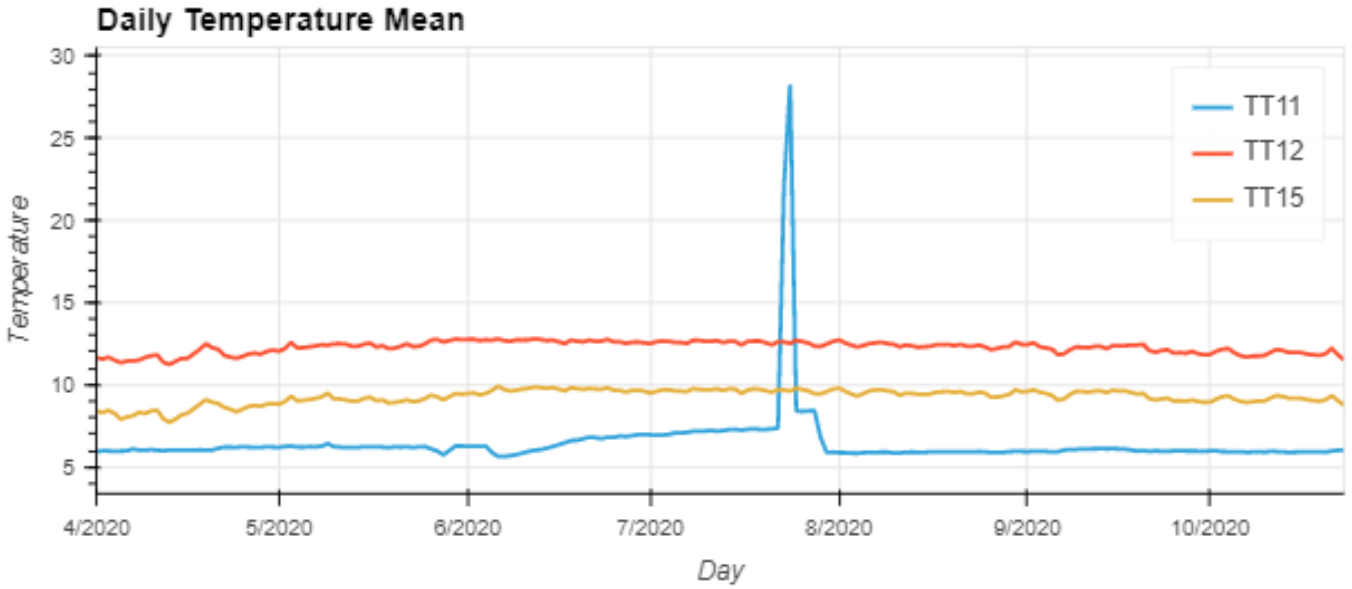
數據圖表

- 不同機器的月平均值，CHS_TT11, 7月的值似乎有比較不正常的高值



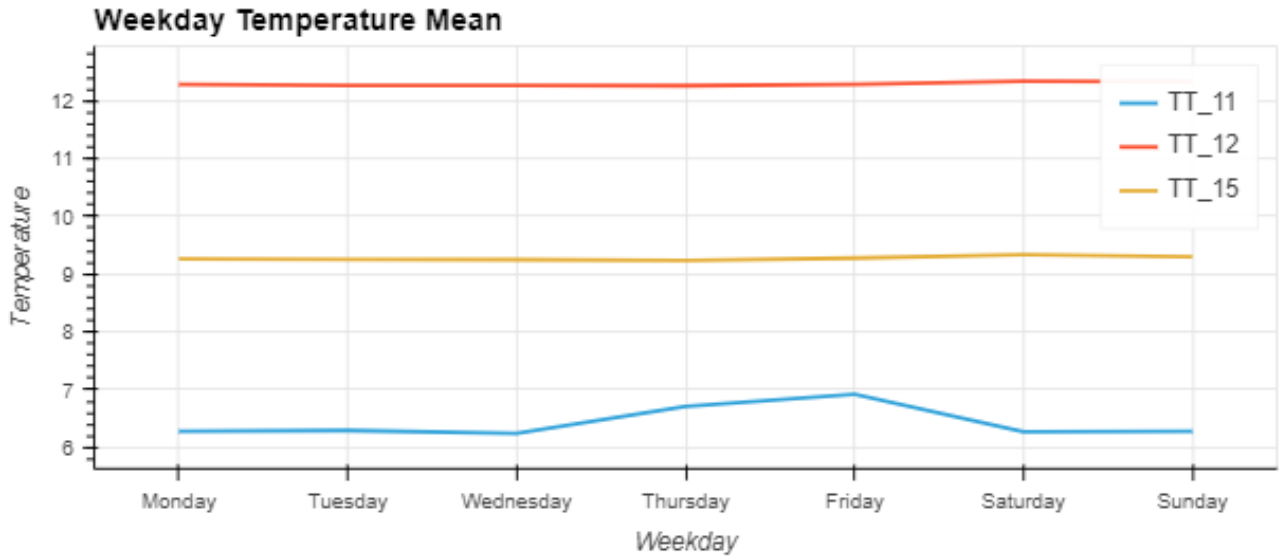
數據圖表

- 各項機器日平均溫度值，可以看到在接近8月的時候有不正常的高峰。



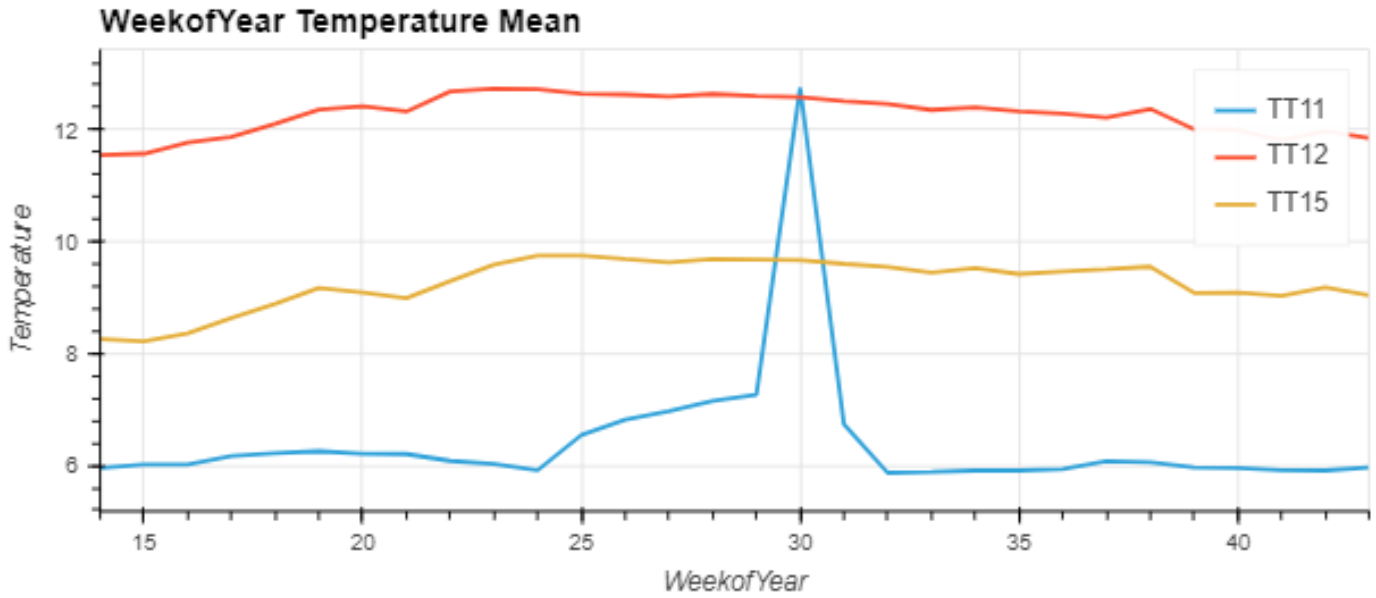
數據圖表

- 每星期日子的平均溫度，照理說應該要是平順的線，因此有高低起伏可能需要注意一下



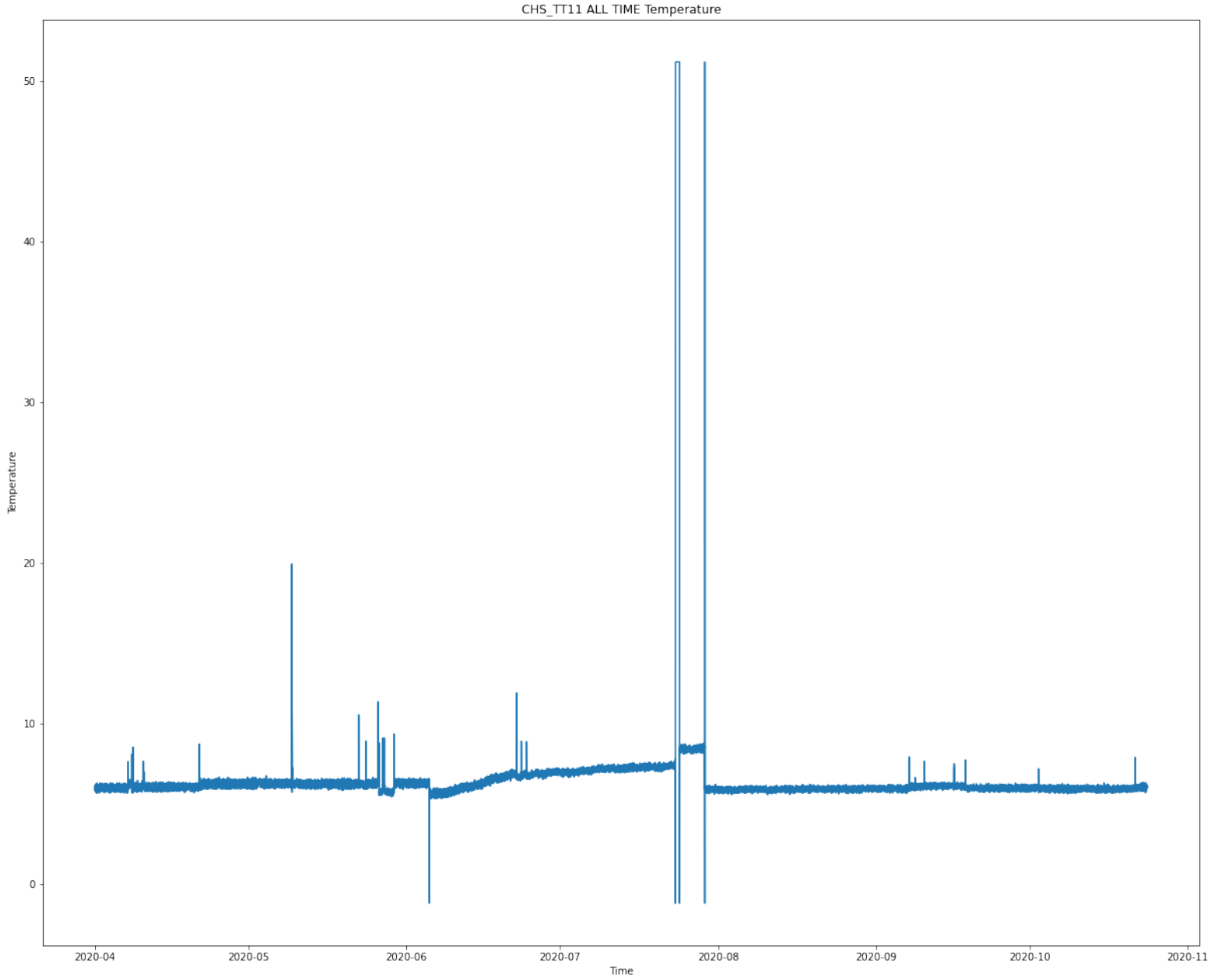
數據圖表

- 一年中的周平均溫度



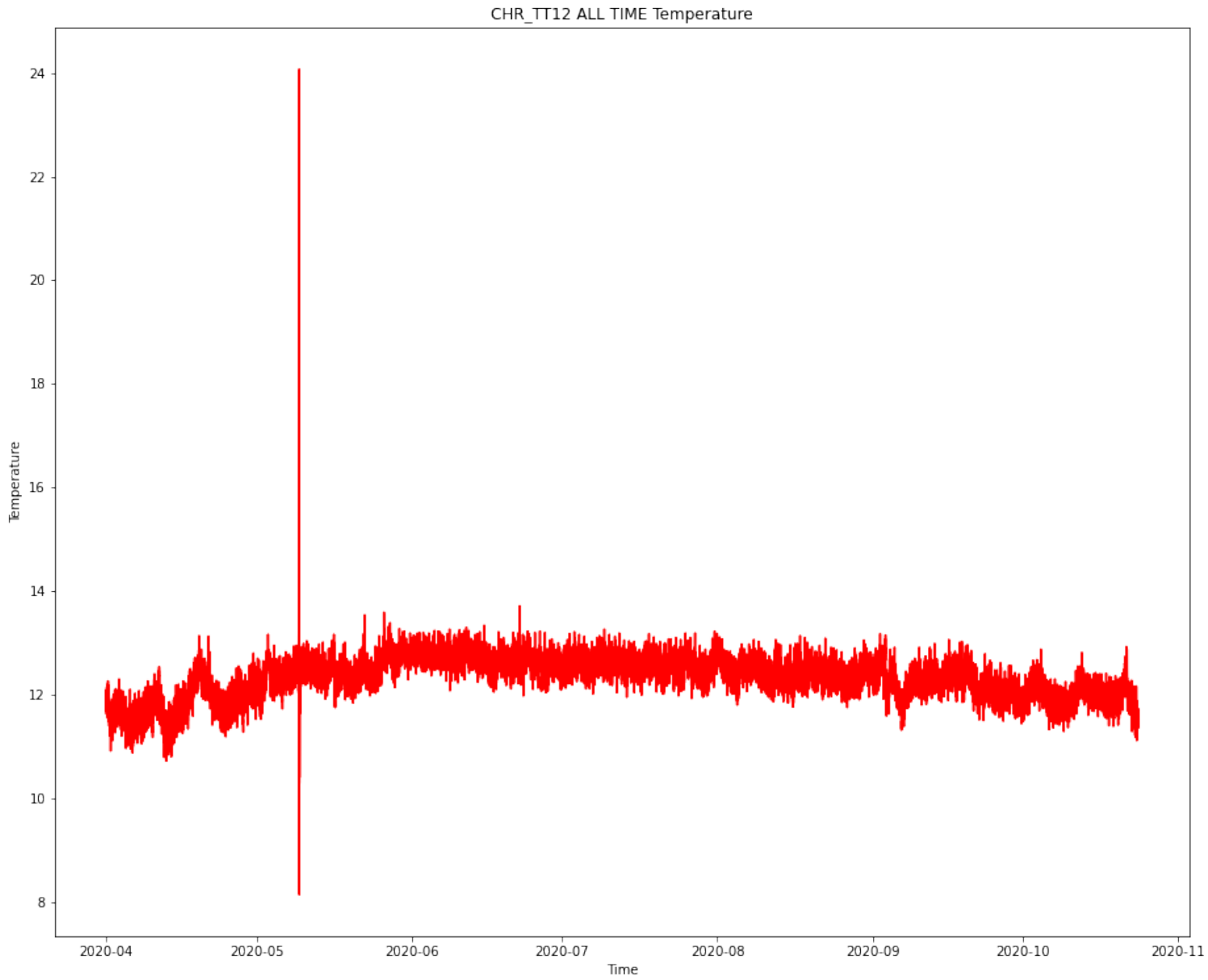
數據圖表

- 機器CHS_TT11溫度隨時間圖



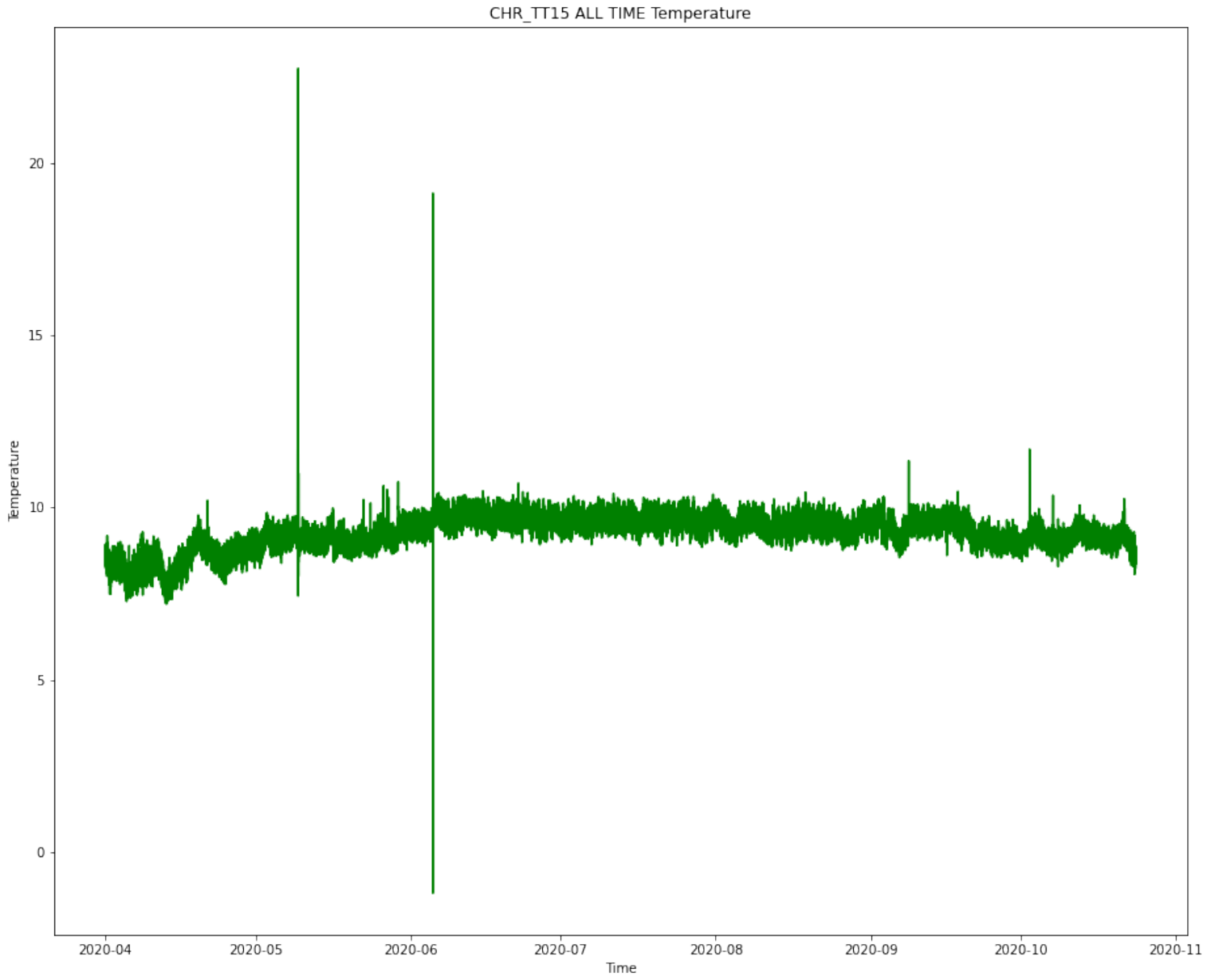
數據圖表

- 機器CHR_TT12 溫度隨時間圖



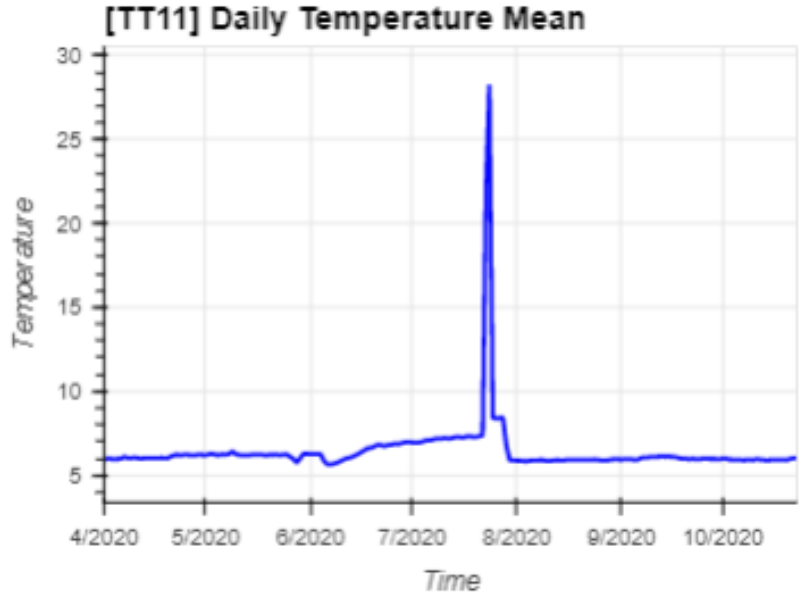
數據圖表

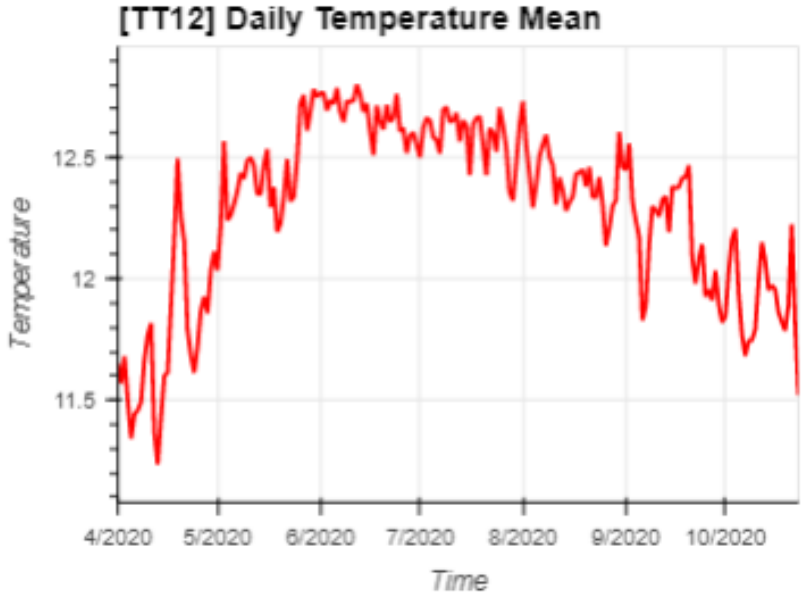
- 機器CHR_TT15溫度隨時間圖

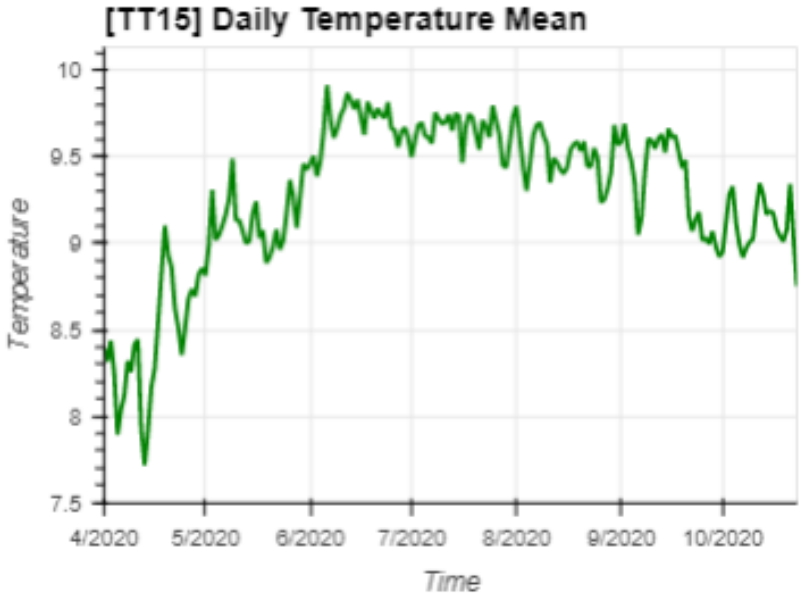


數據圖表

- 日平均溫度總結







小總結

- 透過EDA可以看到資料分布的情形，可以評估哪些值是異常的或是正常的高低起伏，假如是異常的值，這在機器學習稱為離群值，表示這些值是離開群體的值，那麼，這樣的離群值就會影響機器學習的成效，移除它們，並以合理的統計手段補齊，才是正確的作法。

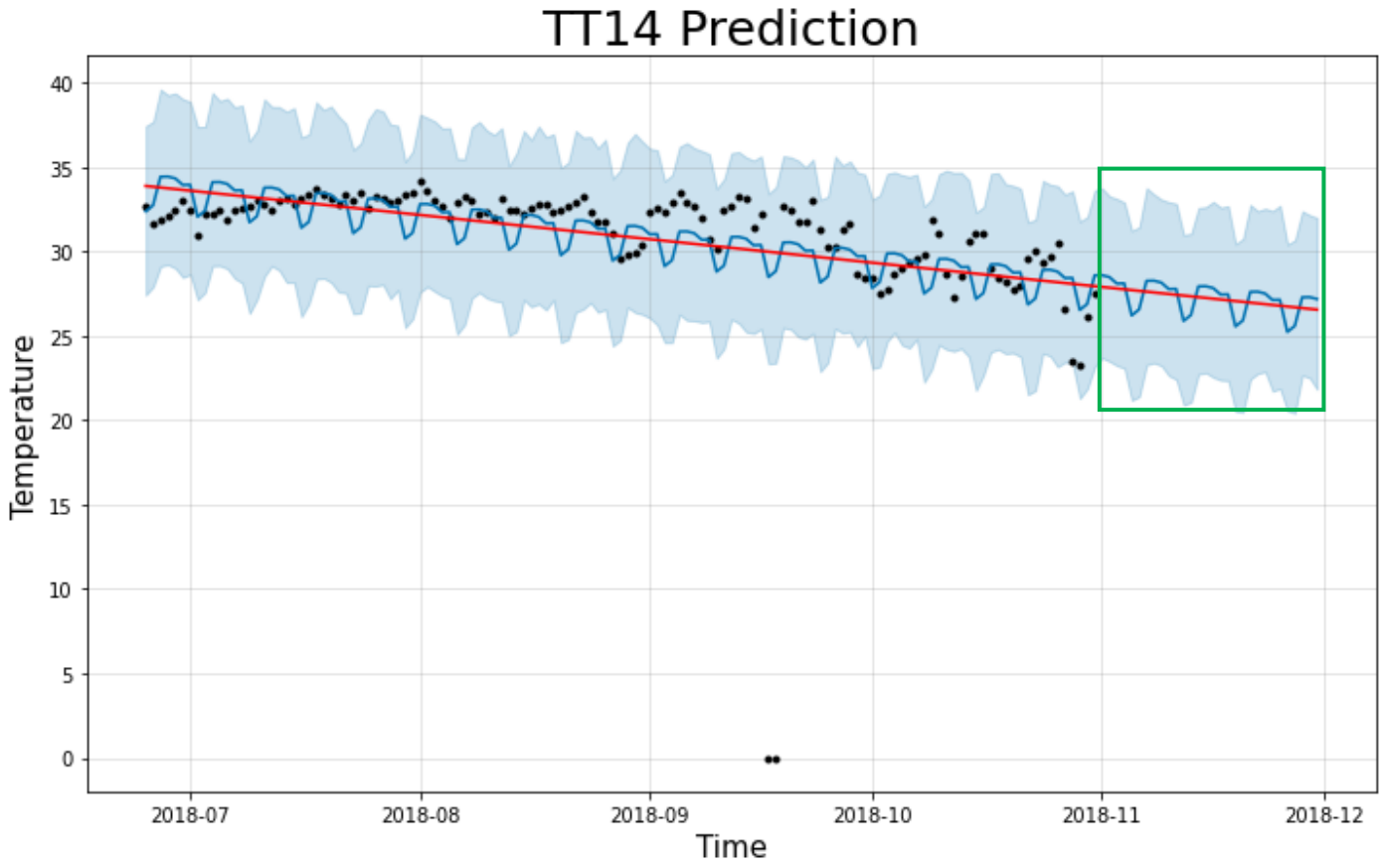
模型預測(日計)

•說明:

模型: fbprophet

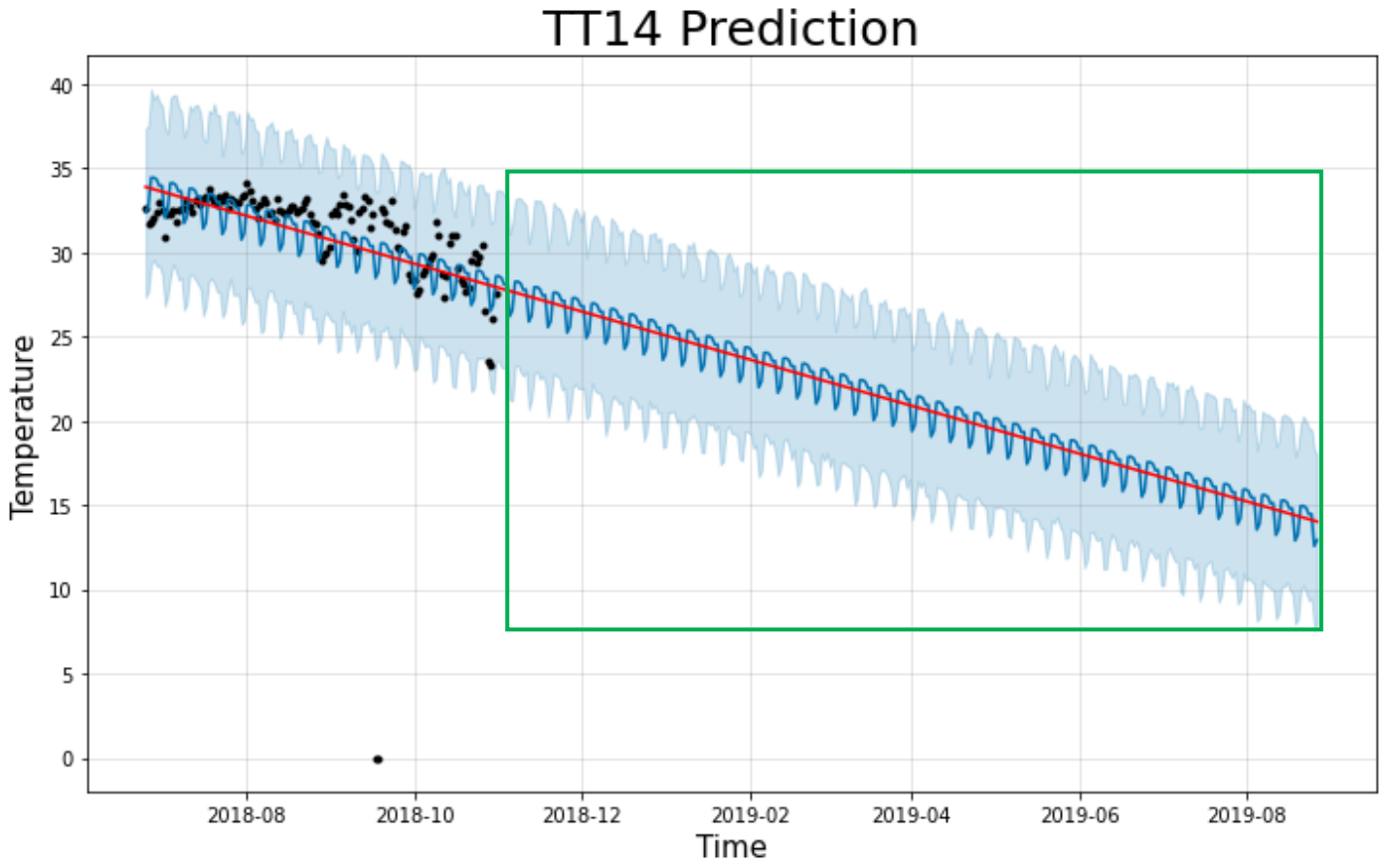
黑點: 實際數據分布
藍色實線: 模型預測值
藍色覆蓋範圍: 模型認為可能的數值分布
紅色實線: 趨勢
綠色框: 未來的預測，這裡是30點，也就是30天

此次的區間是2018.6.25~2018.11.1，資料點其實沒那麼多，加上沒什麼起伏，預測效果沒那麼好



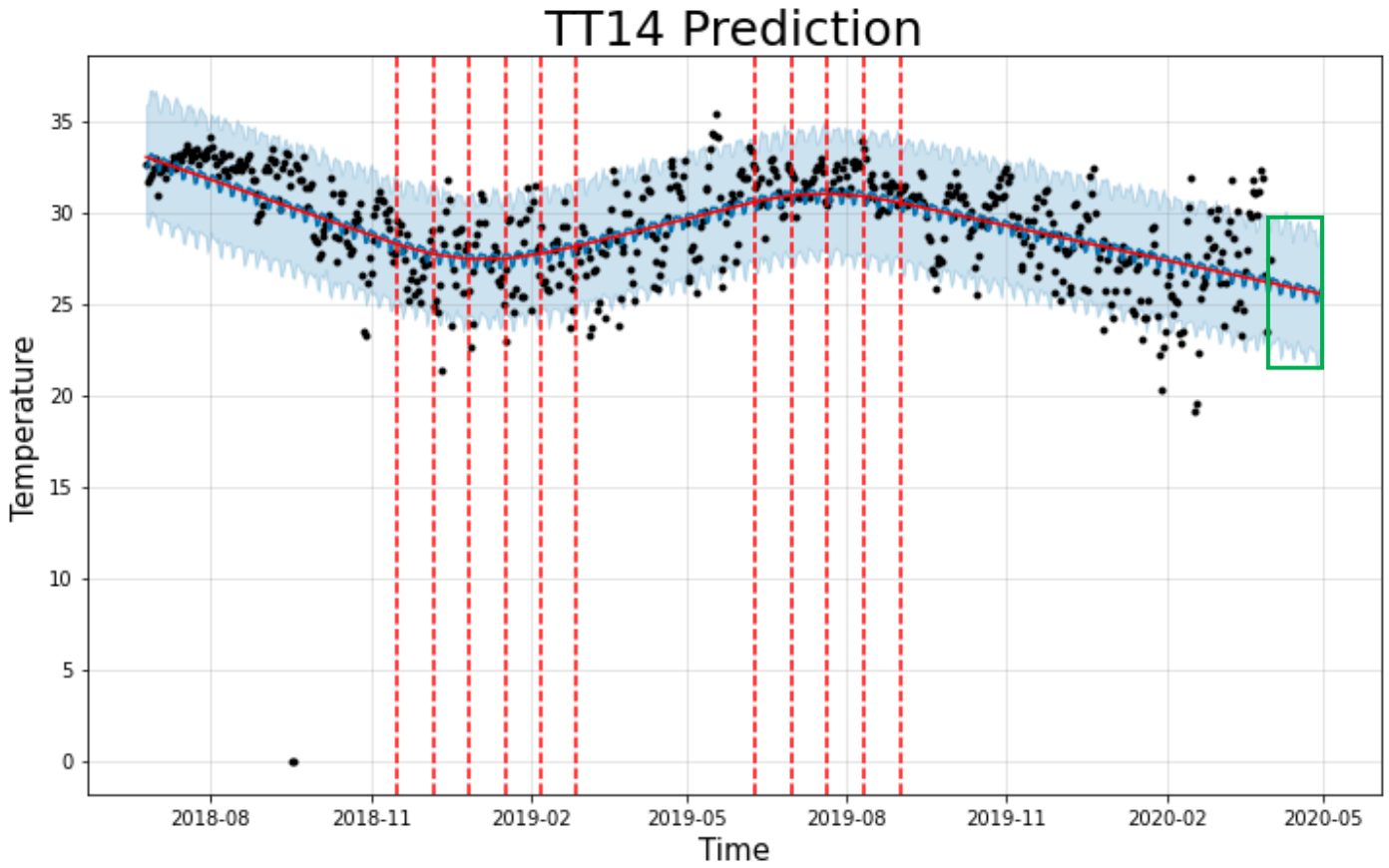
模型預測(日計)

與上一頁取的資料時間區間相同，可以看到模型預測如果取300天的話，模型只是單調的預測一個方向，而且明顯是錯誤的，因此，當數據量不足的話，模型是很難學習和預測的。



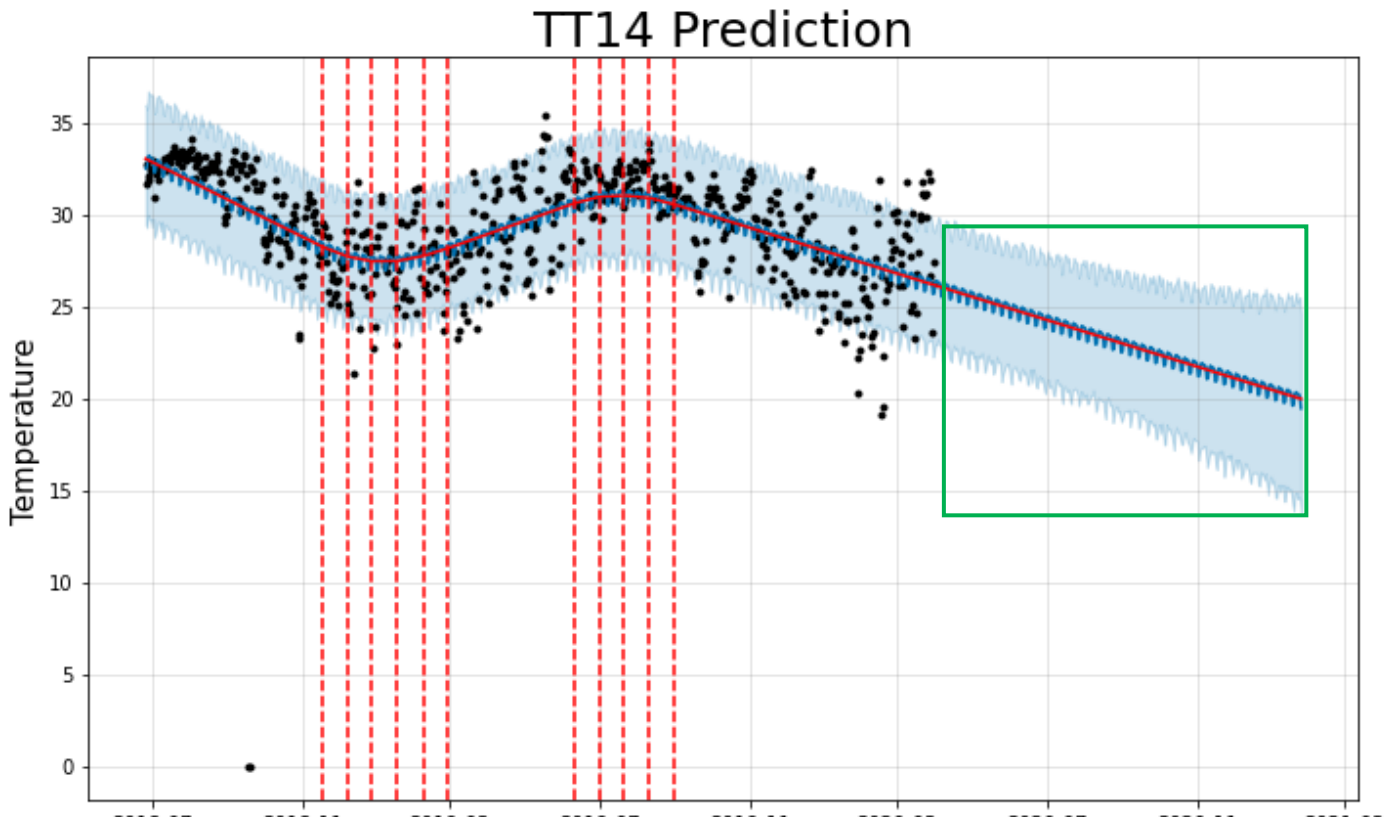
模型預測(日計)

數據點增多，區間變為2018.6.25~2020.4.1，預測未來值30天，可以發現這次模型學習到更多東西，使他的趨勢線一再的改變，這是模型認識到數據分布不再是單調方向分布的證明。



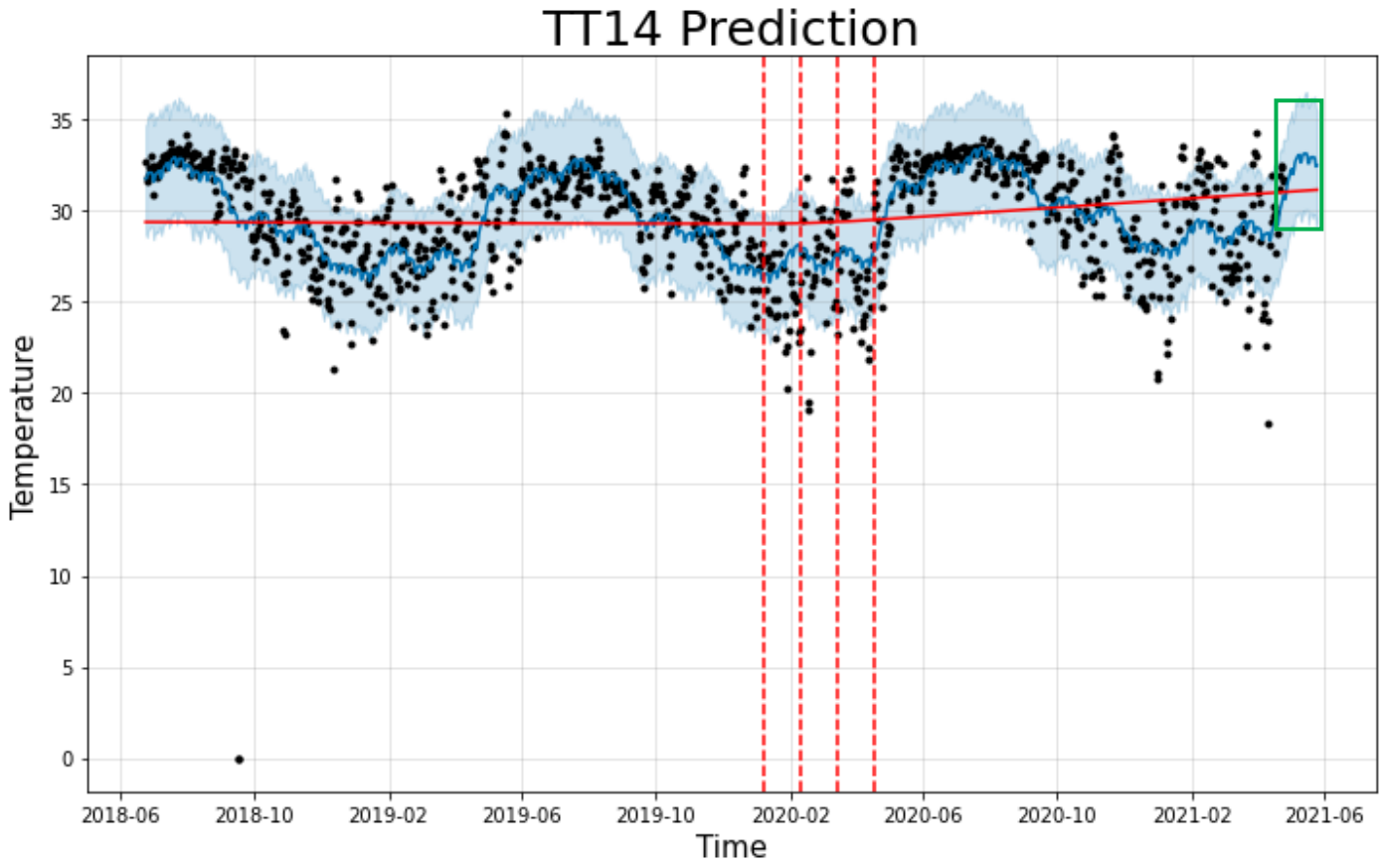
模型預測(日計)

資料區間和上一頁相同，但預測變為300日，可以看到模型的預測變為有點發散，這是好事，這代表模型學到未來不只是單調的方向。



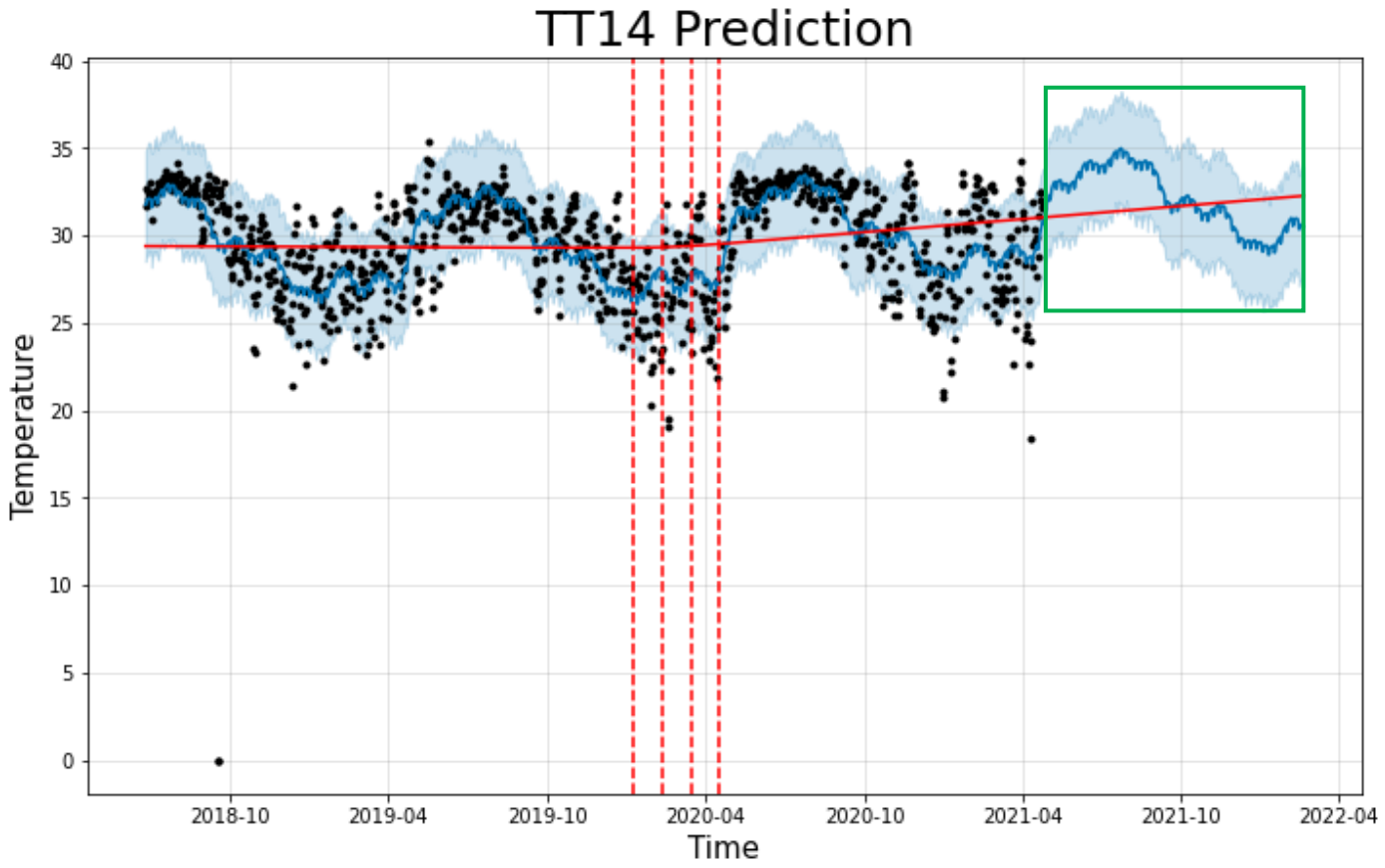
模型預測(日計)

這是我手上所有資料了，
區間2018.6.25~2021.4.25，
可以看到模型似乎已抓到
一定的趨勢了，在它預測
的30日間可以看到它先預
測走向往上，再慢慢往下，
我認為這是很合理的推算。



模型預測(日計)

與上面一樣的時間段，預測未來300日，時間點大概接近2022.3，雖然不知道走勢如何，但有可能就是機器預測的這條路



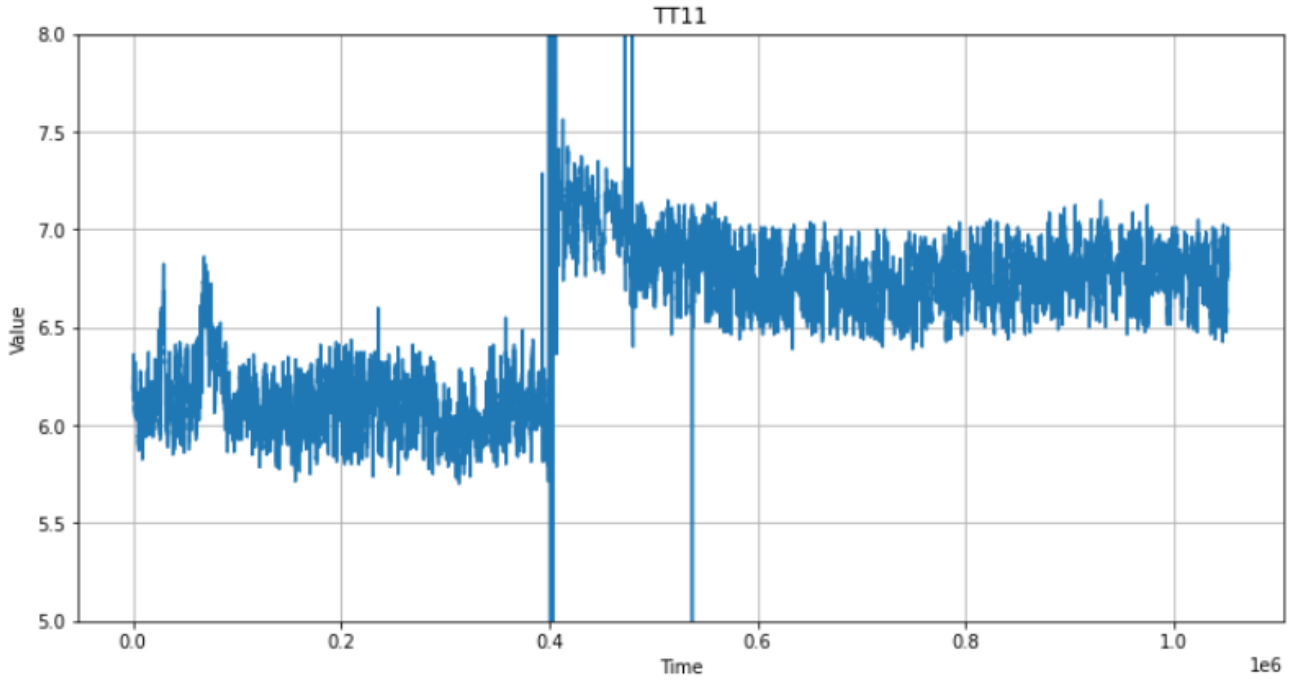
小總結

- 想要在一個機器學習任務中得到好成果，大方向分為兩種，一個是以模型為中心的model-centric view，另一個是以資料為中心的data-centric view。
- 若是model centric view，那就是要努力調整參數，設計模型，來讓計算模型表現的loss值低到可以接受的地步。
- 若是data centric view，那就是盡量改善資料品質，比如說在經過資料清洗後，透過EDA來把離群值清掉，或是增加數據的量也能讓模型學習能力增強。

上面6頁模型預測(日計)基本上示範了如何以data centric view的方向來增強模型預測能力(因為我沒有動任何參數調整)，效果也不錯。

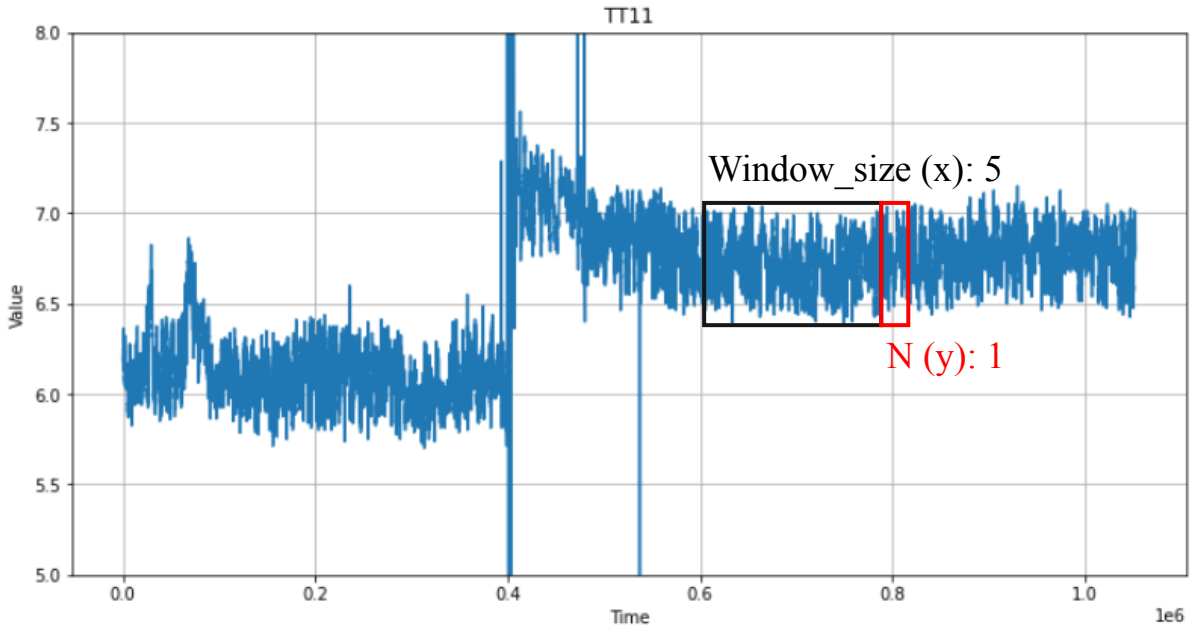
模型預測(秒計)

機器CHS_TT11在2021.4.1 ~ 2021.4.25之間溫度分布的情形，每兩秒採計一次的話，理論上會有1,080,000個點，但實繼數據點為1,052,499個點，因此數據有缺失的情形，但我假定這情形沒發生，時間軸以次序點的形式表現，且數據中很明顯的離群值我也不動，直接將數據丟到深度學習模型中做學習。



模型預測(秒計)

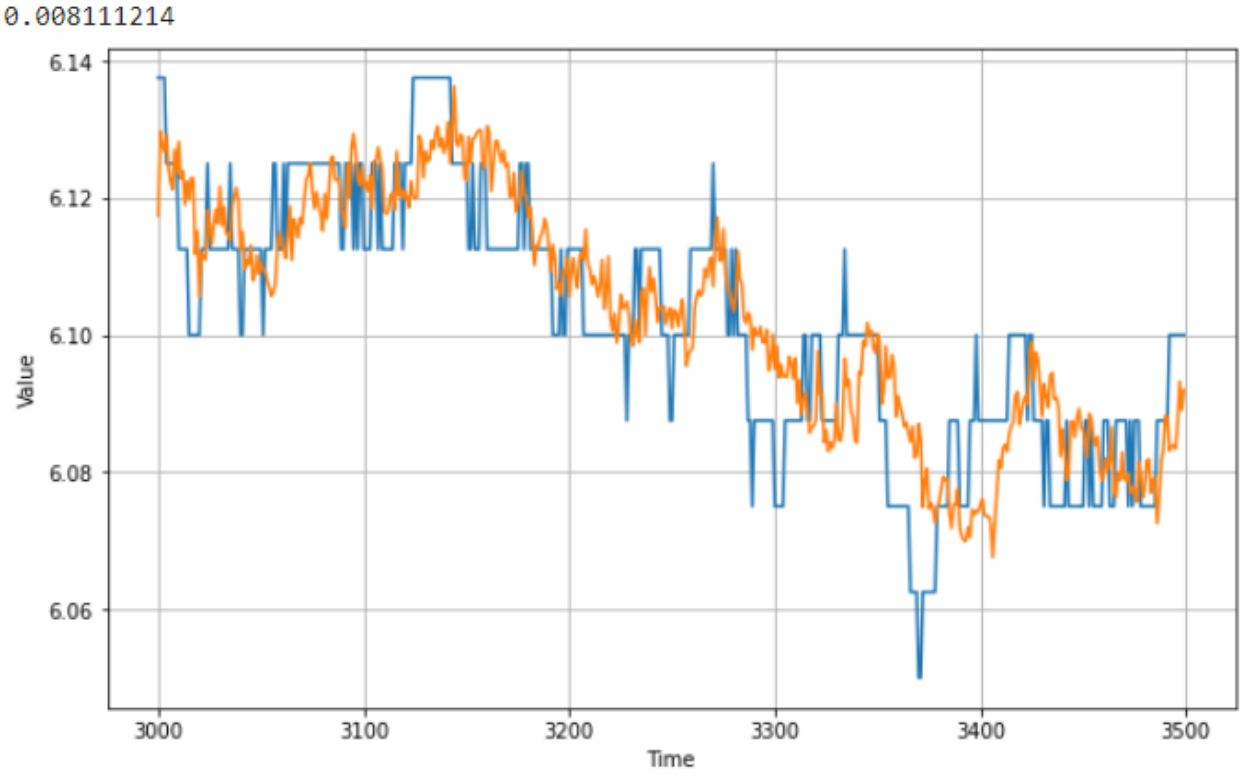
簡單的全連接神經網路，通常不推薦拿來學習有時序性資料的任務，但拿來當開頭當問路石還是不錯的，而資料標記的方法是一次學一個window_size的資料，預測出未來N個數據點，此處window_size是5，且N是1




```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 20)                120
_________________________________________________________________
dense_1 (Dense)              (None, 10)                210
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 11
=================================================================
Total params: 341
Trainable params: 341
Non-trainable params: 0
_________________________________________________________________
```

模型預測(秒計)

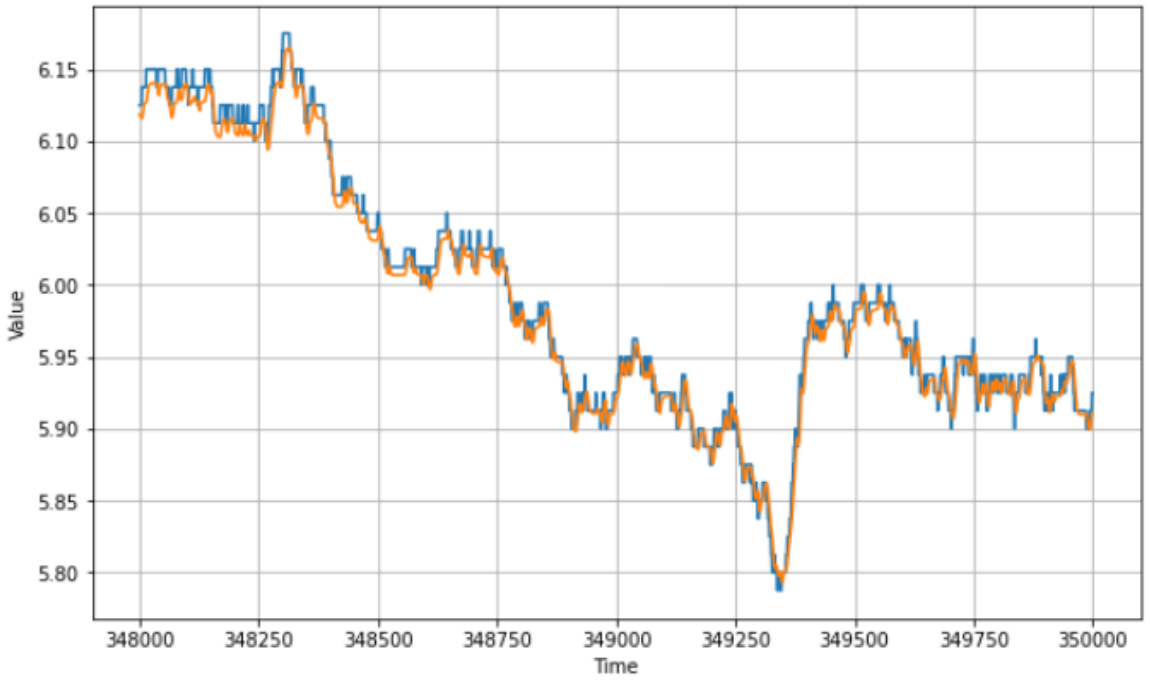
我把資料的前3500點分割出來，3000點作為訓練資料，500點拿來驗證誤差率，即使是拿最簡單的全連結神經網路，成效看來也是不錯，可以看到藍線(實際)和橘線(預測)走勢相當接近，且平均絕對誤差只有0.008，相當低，我想原因大概在於這不是一個很困難的任務，未來的預測只有1點，且數據的起伏並不是很大，造就了簡單模型也能出色的預測結果。



模型預測(秒計)

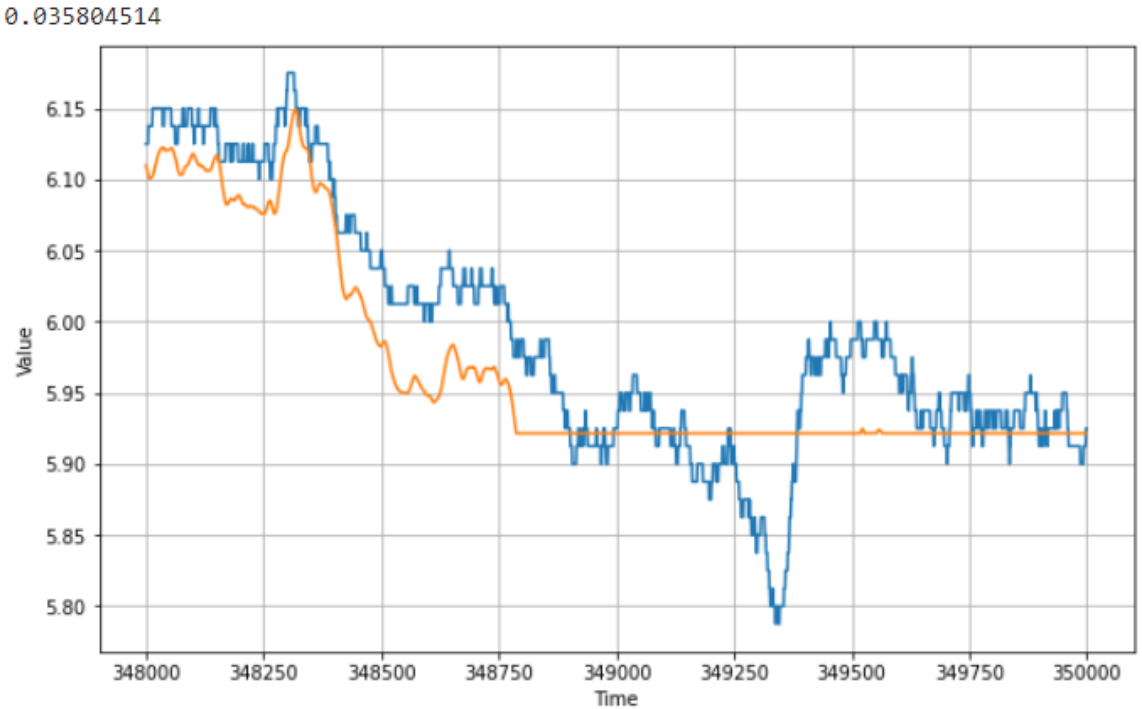
這次我把資料量加到35000點，比先前多了10倍資料，我把34800點資料當作訓練集，這次使用LSTM model，這是適合時序型資料的模型，window_size 是64, learning rate 是10^{-5}，一樣是預測未來1點，可以看到結果相當不錯。

0.0077311154




```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, None, 32)          192
_________________________________________________________________
lstm (LSTM)                  (None, None, 64)          24832
_________________________________________________________________
lstm_1 (LSTM)                (None, None, 64)          33024
_________________________________________________________________
dense (Dense)                (None, None, 30)          1950
_________________________________________________________________
dense_1 (Dense)              (None, None, 10)          310
_________________________________________________________________
dense_2 (Dense)              (None, None, 1)           11
_________________________________________________________________
lambda (Lambda)              (None, None, 1)           0
=================================================================
Total params: 60,319
Trainable params: 60,319
Non-trainable params: 0
_________________________________________________________________
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_1 (Conv1D)            (None, None, 32)          192
_________________________________________________________________
lstm_2 (LSTM)                (None, None, 64)          24832
_________________________________________________________________
lstm_3 (LSTM)                (None, None, 64)          33024
_________________________________________________________________
dense_3 (Dense)              (None, None, 30)          1950
_________________________________________________________________
dense_4 (Dense)              (None, None, 10)          310
_________________________________________________________________
dense_5 (Dense)              (None, None, 5)           55
_________________________________________________________________
lambda_1 (Lambda)            (None, None, 5)           0
=================================================================
Total params: 60,363
Trainable params: 60,363
Non-trainable params: 0
_________________________________________________________________
```



模型預測(秒計)

一樣的資料點個數，只是預測的值變為5個點，可以發現預測的效果，明顯沒那麼好了。

總結

- 當預測超過一個值時，模型可能會預測失敗，如果不考慮改善數據品質的話，那就要從模型參數下手，比如說window size，可以考慮將他調小一點，或是學習率(learning rate)也可以調整看看，若是都沒幫助的話，那可能就要考慮變更模型或演算法。
- 根據EDA的結果，可以知道有許多離群值，調整這些離群值，絕對對訓練的結果有幫助，但我沒有做的原因是，我現在採用的方法是深度學習，就像我透過看資料知道這些點是離群值，同樣的，機器也能透過深度學習知道這些點是離群值而忽略，因此，假如我們要使用淺層學習的話，那麼，就必須去除這些離群值了。
- 時序型資料可以採用方法: simple RNN, LSTM, seq2se2, fbprophet, xgboost.