Regression equation in basic:

y_hat[i] = w[0]*x[i] + w[1]*y[i-1] + w[2]*y[i-2] + w[3]*y[i-3] + w[4]*y[i-4] + w[5]*y[i-5] + b

（預測 y = w[0]*溫度 + w[1]*前一天病例 + w[2]前兩天病例 + w[3]前三天病例 + w[4]*前四天病例 + w[5]*前五天病例 + bias)

Advance Variables:

在 advance part，我多加入了一個 feature 即 precipitation 降雨量，將 model 的 equation 修改成（預測 y = w[0]*溫度 + w[1]*前一天病例 + w[2]前兩天病例 + w[3]前三天病例 + w[4]*前四天病例 + w[5]*前五天病例 +w[6]*降雨量 + b)，隨後 training model 使它可以學到降雨量與病例之間的關係，從而降低預測的失誤率。

Difficulty：

- Python 語言及 numpy 函式庫的不熟悉：
- 對 Machine Learning 的觀念不清晰
- Hyperparameters 的選擇及取值
- MAPE 率難以壓下

Summarize :

一開始在實作的時候，對眾多事物的不熟悉讓整個實作過程進行的非常緩慢。在不斷上網查閱資料及在 codelab 上做實驗反復來回中，一點一滴的增進自己對 python 及 numpy 的熟悉度，並且也補足了很多 linear regression 的知識，而後初步的 model training 也就這麼笨拙的完成了，相當然而，training 出來的結果 MAPE 很高。

隨後我便對一些 hyperparameters 的初始值去做一些改變，例如 learning rate、iteration 的次數等等，因為是初次接觸 machine learning，這些 hyperparameters 的值不太知道該如何變化會比較符合 model training，導致在很多次嘗試裡面，gradient 反而往上跑導致 loss 的值直接溢位，然而在一次又一次的嘗試以及配合網路資料的建議及經驗下，model training 出來的結果是有使 MAPE 下降到 40~50%，但這仍不能滿足作業的條件。

於是我對 data 做了一些處理，因為僅靠溫度去預測登革熱病例，這兩者的關聯性並不大，所以我加入了前五天的病例當成 input feature 給 model 去學習兩者的關係。這一改變確實使 MAPE 率下降不少，使得 train data 的 bias 有降至 15~25%，但 valid data 的 variance 卻仍居高不下。

最後我加入了 Regularization，使得 variance 可以下降至 bias 的基準，甚至表現更佳，而後的 training 過程也是一直在調整 hyperparameters 的值使 model 可以更加精準的預測，但這一過程確實也是最花費時間的。

經由此次作業，不僅讓我更加熟悉 python 和 numpy，更讓我正式踏入了 AI 時代，期待以後的作業可以學到更多 machine learning 的知識