1. Preprocessing/Feature Engineering

我處理feature的方式主要核心是利用前10周的feature和label來預測第11周的結果，而我抽取的feature分別是：「reanalysis\_dew\_point\_temp\_k」, 「reanalysis\_specific\_humidity\_g\_per\_kg」, 「station\_min\_temp\_c」, 「station\_avg\_temp\_c」以及上面列舉這4項各別的三次方，所以對於每周來說，我取的feature共有上面8個以及每周的登革熱案例數目，總共有9筆feature，因此，每10周為單位來取的話就相當於一筆完整預測的feature共有9\*10=90筆資料，而在從training set取得很多上述型態的90筆feature set後，會將這些data都標準化。  
總結來說，我對feature的抽取就是利用上述方式獲得前10周的90筆資料後，利用這些feature來預測第11周的登革熱案例數。然而，針對testing set的部分，因為它的第一周剛好對應到training set結束後的下一周，因此我的做法就是將training set的最後10周抽出來的feature用來預測testing set第一周的案例數，接著就將training set的最後九周加上testing set第一周的feature來預測testing set第二周的案例數，以此類推。

1. Model Description

我的model是利用RNN和Random forest regressor ensemble預測而成的，RNN的架構請見另外一則介紹，而Random forest的部分就是利用每筆feature set共90個feature當作training input，分成預測sj city和iq city的model，sj city的random forest model是利用250個tree來合併預測，iq city的部份是利用200個tree合併預測。  
在獲得validation accuracy不錯的model後，我們將RNN的model預測出來的案例數和Random forest預測出來的案例數以8:2的比例作ensemble來得到testing set的預測值。

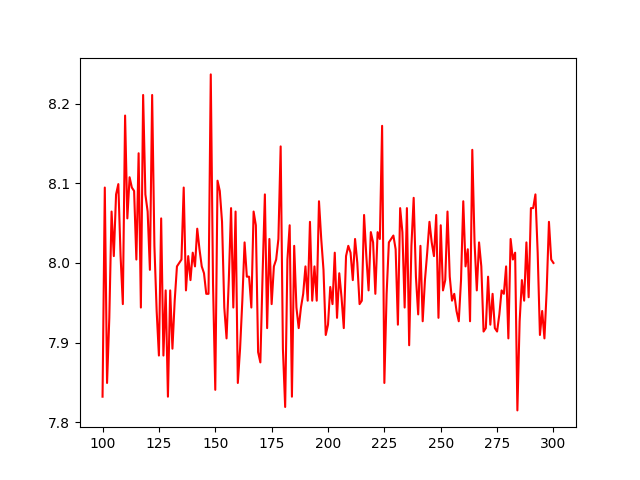
1. Experiments and Discussion

針對random forest的model，我有做了如下的幾種試驗：  
1. 調整sj和iq city合併預測的tree的數目

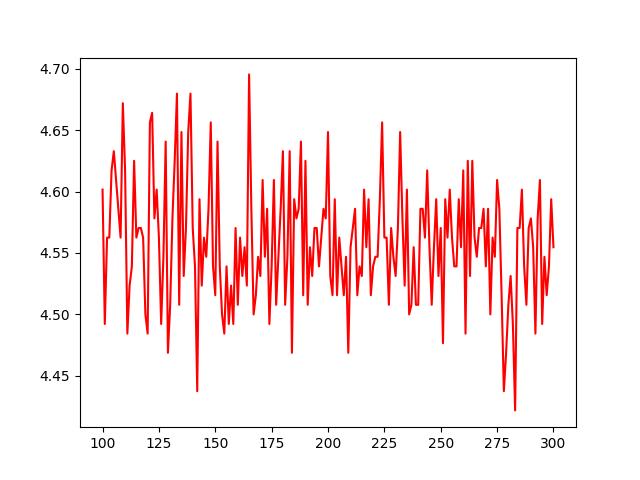
2. 將輸入feature中的三次方項改成一次、二次方

3. 調整RNN model和random forest ensemble的比例

在第一種試驗下，針對sj和iq兩個城市不同tree的數目得到下圖的結果 (橫軸是tree的數目，縱軸是validation loss)，我們可以發現其實不管對於sj或是iq city來說，tree的數目對於loss的影響不大，因此我的作法是選擇loss變異數相對小的區間來確保結果的一致性，因此sj city就是大概在250個tree，而iq city就是大概在150個tree的地方。

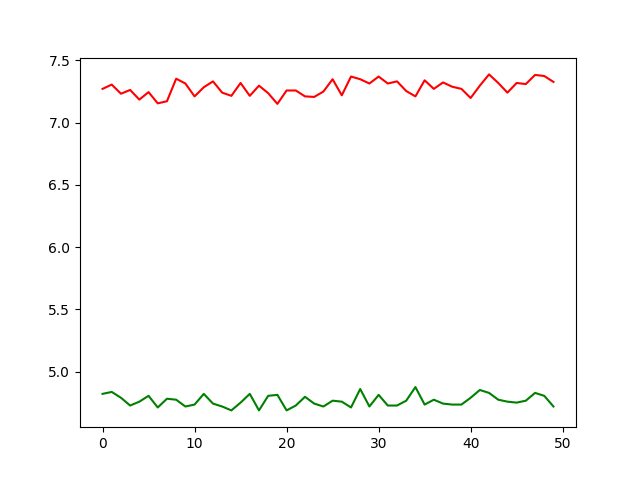


Sj city

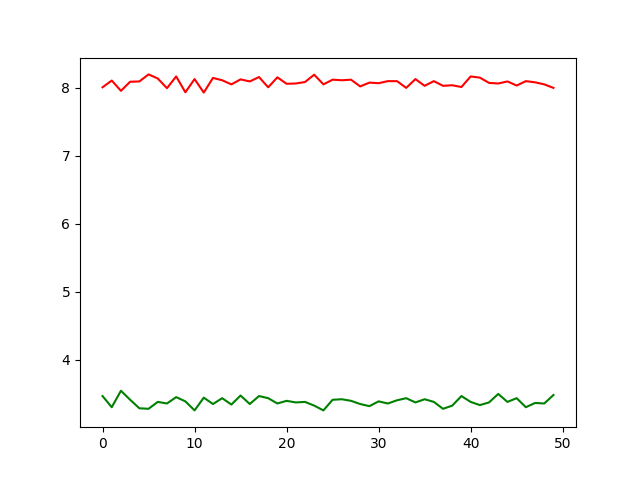


Iq city

在第二種試驗中，主要就是調整我選擇的4種feature在新增feature中的次方項，下圖是用實驗次數當橫軸，validation loss當縱軸的作圖，紅線是代表sj city預測出來的validation loss，綠線則是iq city的，從下圖可以發現用三次方項當作feature和用二次方項當作feature的差別在於，平均來說三次方feature所預測出來的sj city loss是比較穩定的介於7~7.5之間，但是二次方的sj city loss卻在8左右變動，而考量到sj city對於結果的影響較大，所以我選擇以3次方當作新增的feature。



三次方feature



二次方feature

至於最後一項試驗，我選擇以0.8倍的rnn搭配0.2倍的random forest進行ensemble的理由是因為觀察我們上傳最好成績中的結果變化，我們可以觀察下圖，縱軸是sj city的testing set的預測案例數，橫軸則是週數，綠線是單獨利用rnn上傳最好的結果，藍線則是rnn和ramdom forest以8:2的比例ensemble的結果，而紅線則是7:3，當然我還有試過其他的比例，不過為了方便作圖就取這三種來比較。

從圖中可以發現，單獨利用rnn來預測的問題在於峰值的部分會衝的過高，導致整體的loss會稍微偏高，然而它的優點是在其餘的部分相對準確。接著我們觀察藍線，它在峰值的地方在跟rnn ensemble之後做了一些修正，使得峰值不那麼高，但在其他的地方又不和單純rnn的預測結果相當接近，對比紅線的部分，也就是7:3的比例所做的ensemble，它在峰值得地方衝的稍微高一點，不過仍然有修正到單獨rnn的缺失，可是在其他部分卻和rnn的預測結果有點差距，這導致在上傳之後，它所呈現的loss相對不理想，因此，在綜合觀察評判後，我們選擇以8:2的比例將rnn的model和ramdom forest進行ensemble。

