

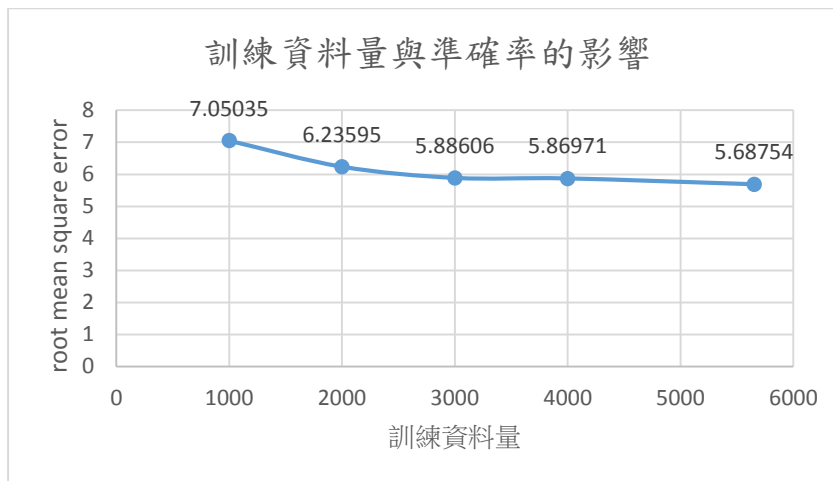
1. 請簡明扼要地闡述你如何抽取模型的輸入特徵 (feature)

答：

在抽取模型 feature 方面，我抽取了 PM2.5、PM10、O3、NO2、RAINFALL、WIND_SPEED 等 feature，其中 PM2.5 包含前九小時的一次項與前四小時的二次項，PM10 包含前四小時的一次項與前兩小時的二次項，O3 取前三小時的一次項，剩餘 NO2、RAINFALL、WIND_SPEED 皆取前兩小時的一次項，共取了 28 筆 data 作為 feature。

2. 請作圖比較不同訓練資料量對於 PM2.5 預測準確率的影響

答：



取了 1000、2000、3000、4000、5652 等不同資料量來 train，其對照的 root mean square error 為 Kaggle 上的分數。可看到 1000 與 2000 的結果差距頗大，而後來結果則漸漸收斂趨於平緩，因此取了太少資料量會造成預估的不準確性；則取大量的資料量雖可增加準確性但確有限且耗費較多時間，所以選取適當的資料量即可。

3. 請比較不同複雜度的模型對於 PM2.5 預測準確率的影響

答：

最開始只取前九小時的 PM2.5 一次項(Kaggle 分數 5.79969)，出來的 training error 就不甚理想，此 model 過於簡單，此為 underfitting 的結果；再來取了全部指標的前九小時一次項(Kaggle 分數 5.84535)，training error 較好，但其 testing error 頗差，model 過於複雜，此為 overfitting 所造成。由前面兩種結果，決定有些指標就不抽取（因為不一定每個指標都與

pm2.5 有關)，其 testing error 也降低許多，再做進一步修正，最後採取有些指標不取滿九個小時（因為有些指標太久以前的數據應與第十小時的 pm2.5 較無關係，會造成不必要的誤差）且與第十小時 pm2.5 較有相關的指標如 pm2.5、pm10 會加上前幾小時的二次項來增加預測的準確率(Kaggle 分數 5.68754)，因此最後所決定的 model 就沒有 underfitting 與 overfitting 的問題，而是處於適中狀態的 model。

4. 請討論正規化(regularization)對於 PM2.5 預測準確率的影響

答：

理論上採取正規化的方法會使回歸線更加平滑且有更好的 result，當我取了全部指標且都九小時的 feature 時，使用正規化（ $\lambda = 1000$ ）時會有不錯的 result(Kaggle 分數 5.78896)；但當我抽取部份指標且只有部份時數的 feature 時，使用正規化並沒有太大的幫助甚至使 result 更差(Kaggle 分數 5.69648)，所以最後我並沒有採取正規化的方法，因此我想在 overfitting 時（model 過於複雜），正規化應該會有不錯的效果。

5. 在線性回歸問題中，假設有 N 筆訓練資料，每筆訓練資料的特徵(feature) 為一向量 x^n ，其標註(label)為一存量 y^n ，模型參數為一向量 w （此處忽略偏權值 b ），則線性回歸的損失函數(loss function)為 $\sum_{n=1}^N (y^n - w \cdot x^n)^2$ 。若將所有訓練資料的特徵值以矩陣 $X = [x^1 \ x^2 \ \cdots \ x^N]$ 表示，所有訓練資料的標註以向量 $y = [y^1 \ y^2 \ \cdots \ y^N]^T$ 表示，請以 X 和 y 表示可以最小化損失函數的向量 w 。

答：

以 y' 表示 train 出來的答案向量，因此 $y' = X^T w$ ，此時 loss function 即可寫成 $L = (y - y')^T (y - y') = (y - X^T w)^T (y - X^T w)$ 。

$$\frac{\delta L}{\delta w} = 2X(y - X^T w) = 2Xy - 2XX^T w$$

$$\text{Let } 2Xy - 2XX^T w = 0, \quad Xy = XX^T w, \quad w = (XX^T)^{-1}Xy$$