Machine Learning Final Report

- **題目：DengAI: Predicting Disease Spread**

- **Team name：**

  NTU_b03505027_隨便不 care

  Members：

  劉亦浚 李育嘉 曾耕森 楊其昇

  Work division：

  劉亦浚 楊其昇負責 DengAI: Predicting Disease Spread

  李育嘉 曾耕森負責 Pump it Up: Data Mining the Water Table

- **Preprocessing/Feature Engineering**

在 dengue_features_train.csv 中，先將 sj 與 iq 兩城市的 features 分別存成 sj_train 與 iq_train 兩個 array（皆不包含 city、year、week_strat_date 的資料），因考慮到時間的連續性，我們覺得遇到空格時，則取前一個有資料的值來填入，其效果也比採用取平均填入還要更好。比較特別的是，在 sj_train 裡，我們 drop 掉了 ndvi_ne 的 feature，原因為空格太多，不丟掉會影響最後的結果。

在 dengue_labels_train.csv 中，也將 sj 與 iq 兩城市的 labels 分別存成 sj_label 與 iq_label 兩個 array，再來對兩城市的 features、label 做標準化。最後，我們覺得登革熱在爆發時，會有一段潛伏期（也就是跟之前的 features 會有相關），所以我們決定將前三周的 features 併入當周的 features 來做預測，也就是一次取四周來預測，此時 sj_train 與 iq_train 的 features 維度分別為 80、84

（sj_train 中有拿掉四周的 ndvi_ne，因此維度會比 iq_train 的維度少 4），取 features 的概念如下圖所示。

1	reanalysis	reanalysis	reanalysis	reanalysis	station	avg station	diu station	ma station	mir station	precip_mm	total_cases
2	73.36571	12.42	14.01286	2.628571	25.44286	6.9	29.4	20	16		4
3	77.36857	22.82	15.37286	2.371429	26.71429	6.371429	31.7	22.2	8.6		5
4	82.05286	34.54	16.84857	2.3	26.71429	6.485714	32.2	22.8	41.4		4
5	80.33714	15.36	16.67286	2.428571	27.47143	6.771429	33.3	23.3	4		3
6	80.46	7.52	17.21	3.014286	28.94286	9.371429	35	23.9	5.8		6
7	79.89143	9.58	17.21286	2.1	28.11429	6.942857	34.4	23.9	39.1		2
8	82	3.48	17.23429	2.042857	27.41429	6.771429	32.2	23.3	29.7		4
9	83.37571	151.12	17.97714	1.571429	28.37143	7.685714	33.9	22.8	21.1		5
10	82.76857	19.32	17.79	1.885714	28.32857	7.385714	33.9	22.8	21.1		10
11	81.28143	14.41	18.07143	2.014286	28.32857	6.514286	33.9	24.4	1.1		

- **Model Description**

以下列出三種 model：

Model1：城市 sj（XGBRegressor）、城市 iq（XGBRegressor）

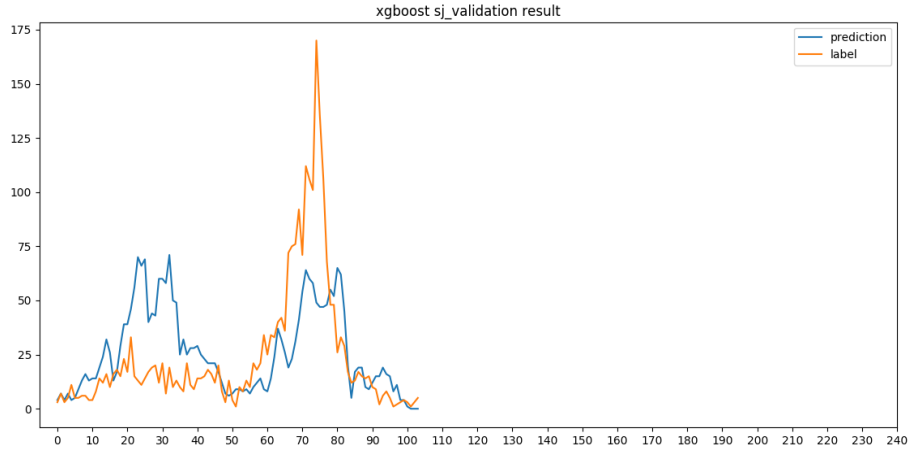
Model2：城市 sj（GradientBoostingRegressor）、城市 iq（BaggingRegressor）

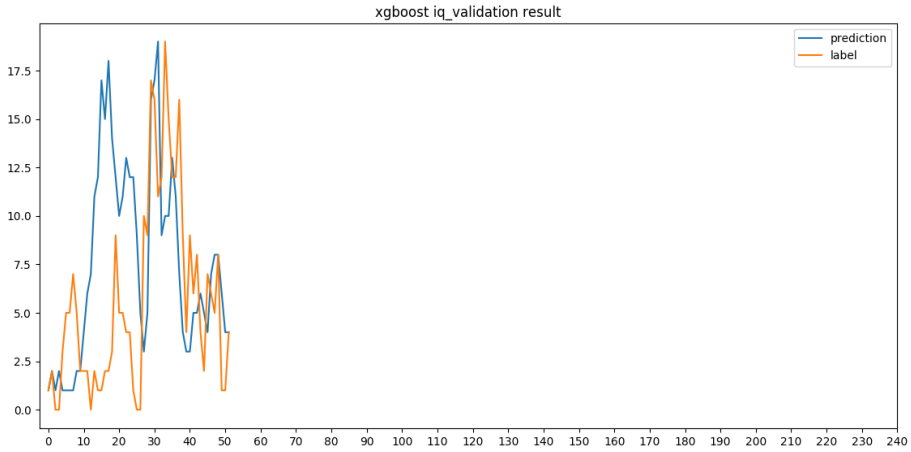
Model3：城市 sj（ExtraTreesRegressor）、城市 iq（ExtraTreesRegressor）

Model1：兩者 XGBRegressor 的參數皆為 n_estimators = 250, subsample = 0.4, max_depth = 5, colsample_bytree = 0.7。

下兩張圖為 Model1 validation data 的 predict 與真實 label 的比較：

（城市 sj 切了最後 104 筆當作 validation data；城市 iq 則為 52 筆。）



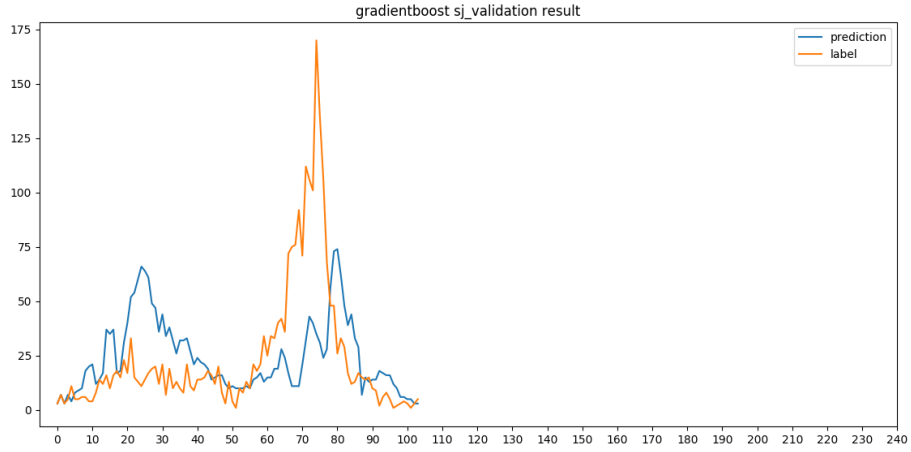


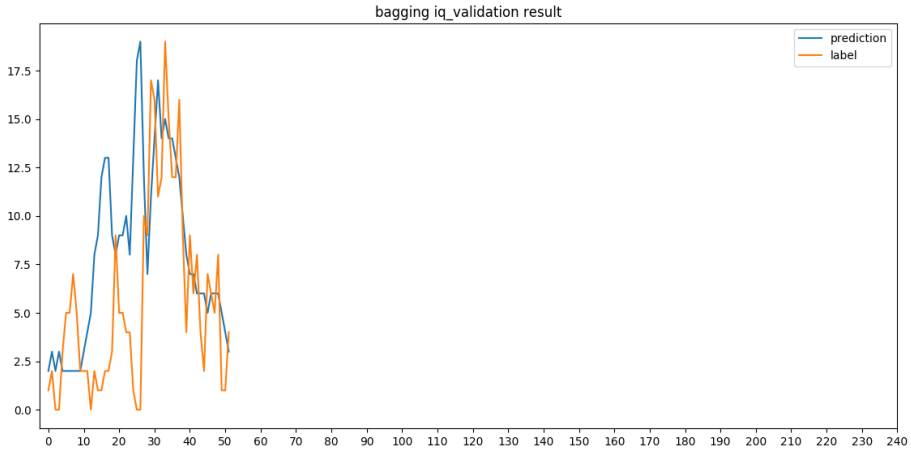
Validation data 的 mae 為 13.49（兩個城市合併），在 sj 的 mae 較大，由圖也可看出 sj 在真實高峰時，會達到 170 左右，但預估的值約在 75，低估了許多也造成很大的誤差，在較低的值時，也是預測高估了真實許多，造成誤差；在 iq 則 mae 較小，沒有高峰，預估的趨勢與真實也滿符合。

Model2：GradientBoostingRegressor 的參數為 loss = 'huber', learning_rate = 0.1, n_estimators = 1000, max_depth = 5, criterion = 'mae'；BaggingRegressor 的參數為 n_estimators = 100, max_features = 0.6, max_samples = 0.6。

下兩張圖為 Model2 validation data 的 predict 與真實 label 的比較：

（城市 sj 切了最後 104 筆當作 validation data；城市 iq 則為 52 筆。）



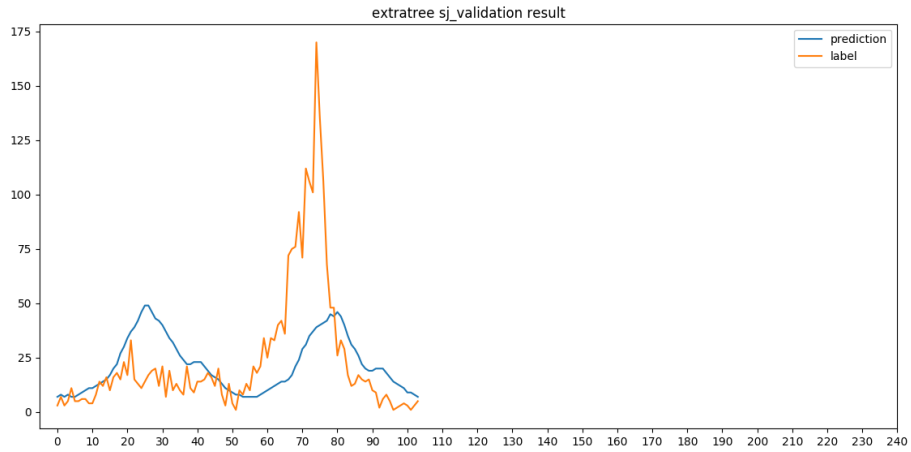


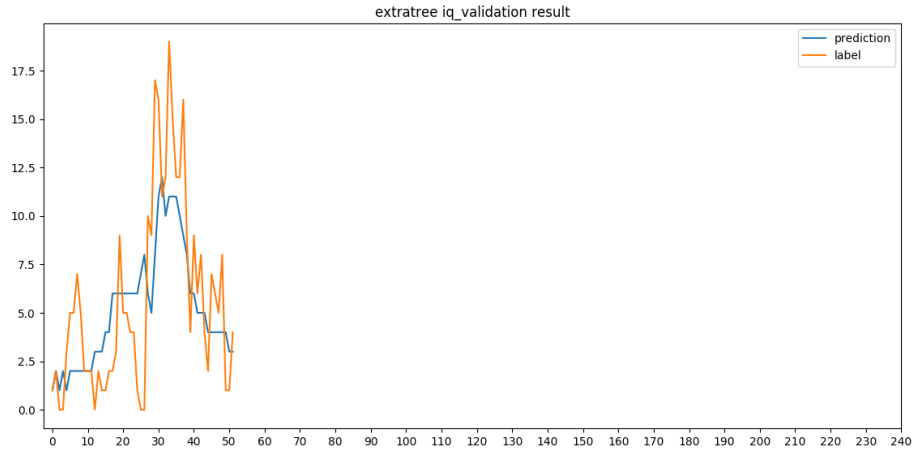
Validation data 的 mae 為 14.78（兩個城市合併），依然是 sj 的 mae 較大，預估高峰的值約在 75，低估了真實許多，在較低的值預估約為 70，則是高於真實；在 iq 也是 mae 較小，沒有高峰，預估的趨勢與真實也滿符合。

Model3：兩者 ExtraTreesRegressor 的參數皆為 n_estimators = 2000, criterion = 'mae', max_depth = 3。

下兩張圖為 Model3 validation data 的 predict 與真實 label 的比較：

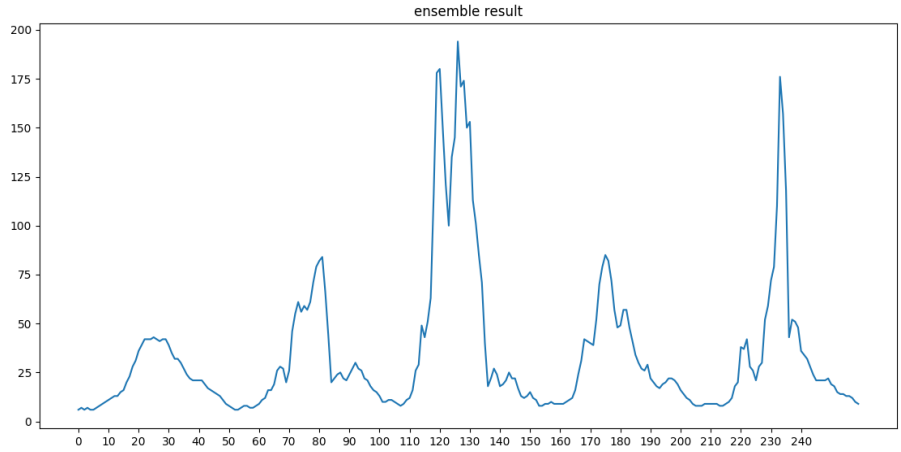
（城市 sj 切了最後 104 筆當作 validation data；城市 iq 則為 52 筆。）





Validation data 的 mae 為 12.76（兩個城市合併），整體的預估線較為平緩，且值比起前面的 model 偏低一點，在 sj 中，雖然預估高峰的值約在 50，低估了真實許多，但在預估較低的值時，與真實 label 較為相近，也是 mae 比前面的 model 較低的原因；在 iq 中，雖預估的趨勢與真實沒那麼吻合，但因為真實的值都沒有很大，所以對 mae 的誤差影響並不大。

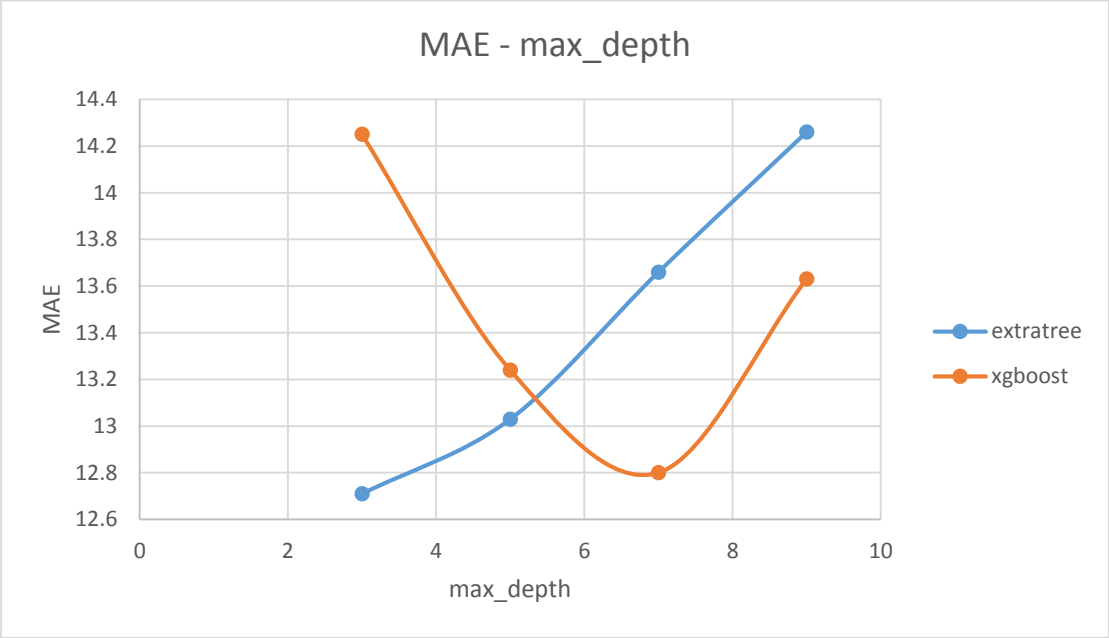
總和上述三個 model，可發現 ExtraTreesRegressor 所預估出的值較低，且浮動沒有很劇烈；在 GradientBoostingRegressor、BaggingRegressor、XGBRegressor 中，預估的值較高，且改變會較劇烈。了解這些 model 特性後，在 ensemble 中是非常有用的，我們可以先用某一個 model 看出 test prediction 的大致分布（local peak 值的大小、趨勢的波動程度），再決定哪些區段適合用甚麼 model 來做 ensemble，如 local peak 的值偏低時（下圖的第一個 peak），ExtraTreesRegressor 預估的值就可以佔較高的比例；若 local peak 的值適中時（下圖的第二、四個 peak），ExtraTreesRegressor 與其他 Regressor 所預估的值，兩者佔的比例可大約相等；若 local peak 的值偏高時（下圖的第三、五個 peak），ExtraTreesRegressor 預估的值佔的比例可較低，或者甚至不包含在 ensemble 中。藉此來調整每一個區段所需的 model，將不同 model 的優點運用在恰當的地方，可以大大的提升準確率。



- Experiments and Discussion

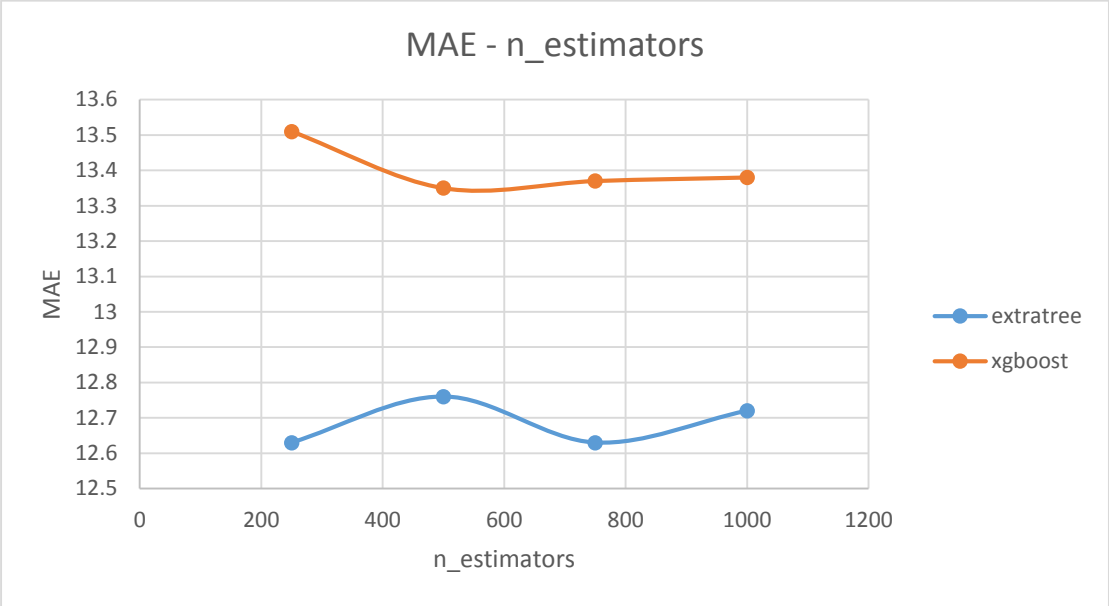
實驗我們採用 ExtraTreesRegressor 和 XGBRegressor 兩種 model，分別觀察不同 max_depth、n_estimators、shift（預測值的位移）在 validation data 上的表現差異。

實驗一：max_depth

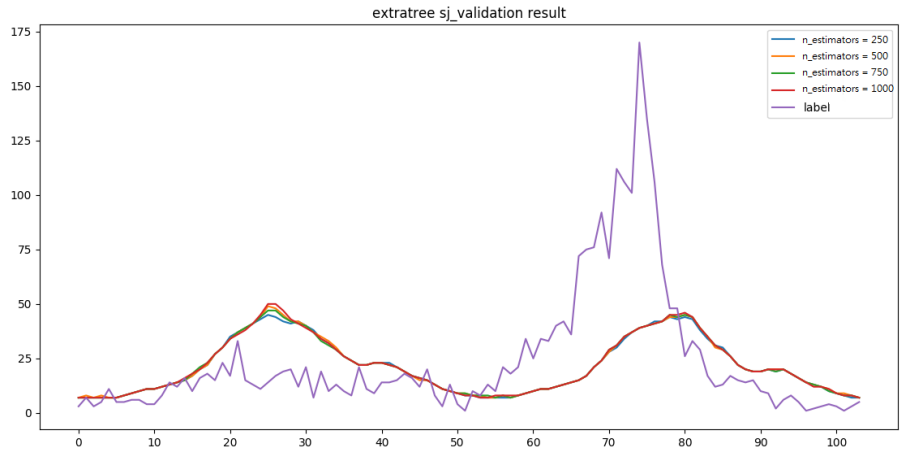


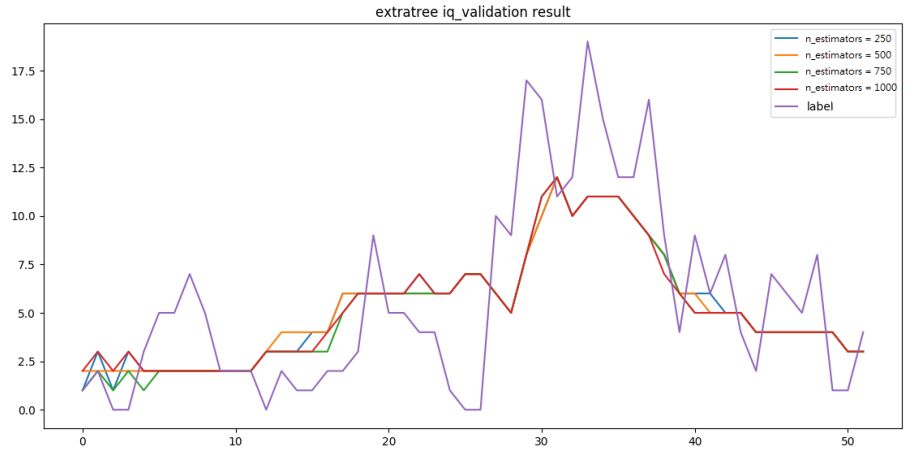
在 ExtraTreesRegressor 上，決策樹的深度愈深，預測結果反而不好，我們估計是 overfitting 的緣故。在 XGBRegressor 上，決策樹稍微有點深度，結果仍然會不錯，最好的情況發生在 max_depth = 7。

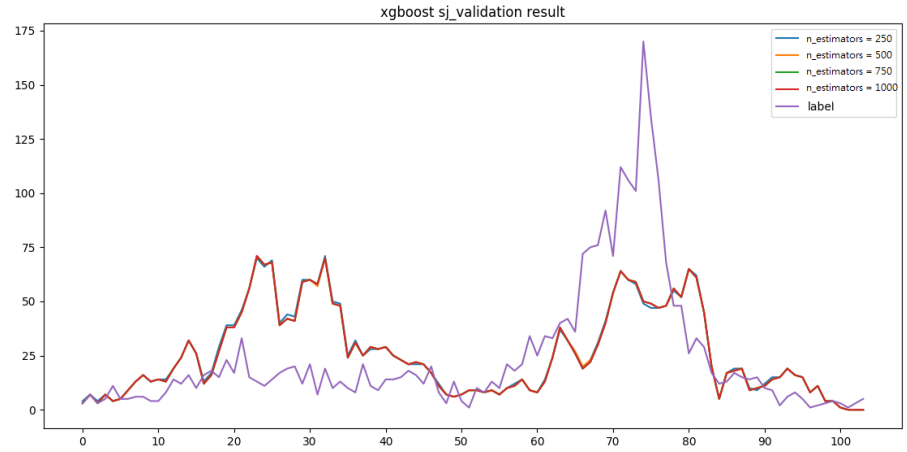
實驗二：n_estimators



由這張圖可觀察出，在只改變 n_estimators 的參數下，同一個 model 的 MAE 與該 model 的平均相差不超過 0.1，所以這個參數影響算是非常小，唯一影響大的地方應該是在 training 所花的時間，而且 ExtraTreesRegressor 的表現比 XGBRegressor 略好。

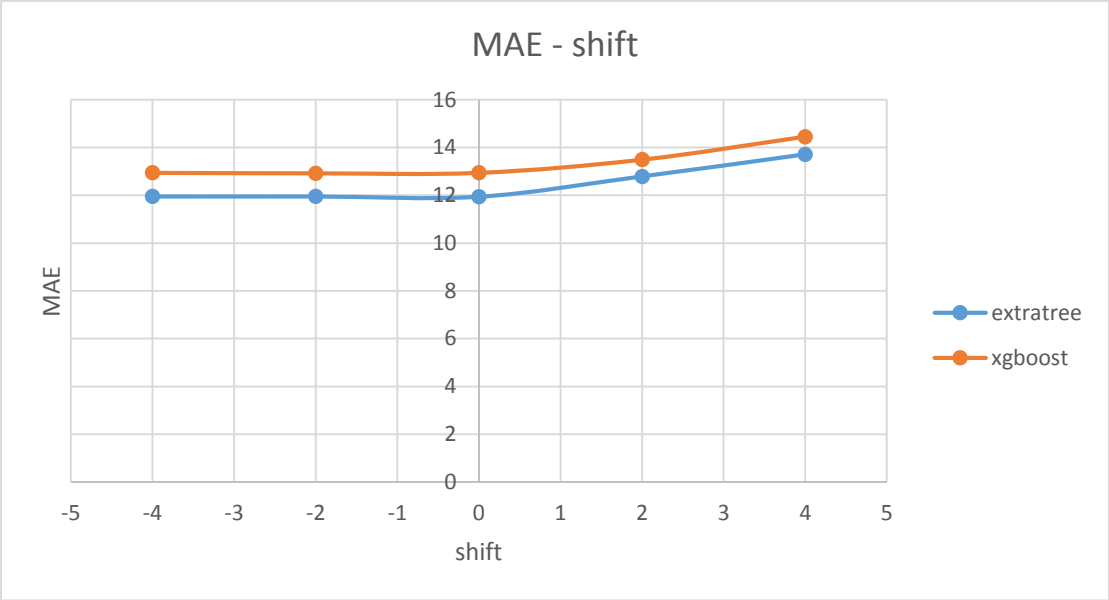






以上四張圖是 ExtraTreesRegressor 和 XGBRegressor 分別在城市 sj 和 iq 的 validation 上的預測結果，可看出每張圖上，除了 label 以外的四條曲線，重疊程度相當高。

實驗三：shift



由於我們發現，sj 城市的預測波形，local peak 出現的位置普遍都會比 label 後面一點點，於是我們決定在預測結果上再做一個 shift 的動作。此圖表橫軸代表我們將 output 向後 shift 多少週，負值則是向前 shift，由於預測 output 的 peak 比 label 晚出現，所以理論上將 output 向前 shift 的結果會比較好，圖表也證實了這個說法，shift 為正值的時候 MAE 會略大（差異幅度並非像實驗二一樣小到可以忽略）。然而我們在 platform rank deadline 前發現 testing data 上，往後 shift 的結果較完全不 shift 的結果好，於是我們選擇 shift = 2。然而死線之後，我們才發現這個現象，而將 shift 改為 -2 之後，drivendata 上的成績更由 22.6202 進步到 21.4471，算是一個非常大的進步，對於我們無法提早觀察到此現象，我們感到非常可惜。