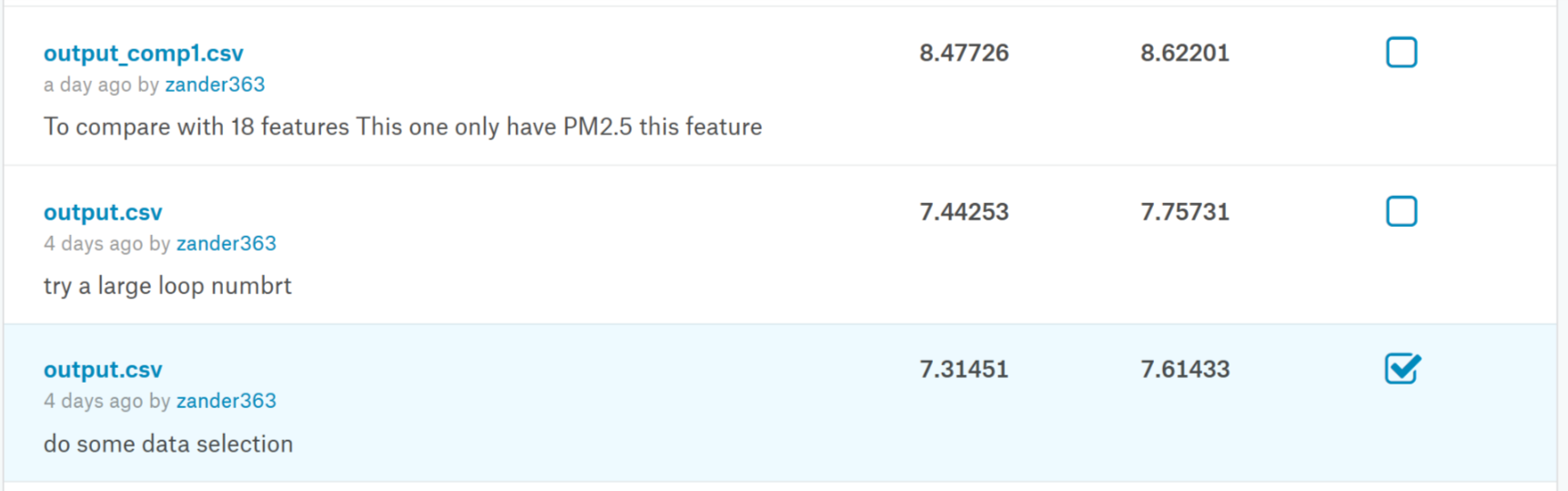
Homework 1 Report - PM2.5 Prediction

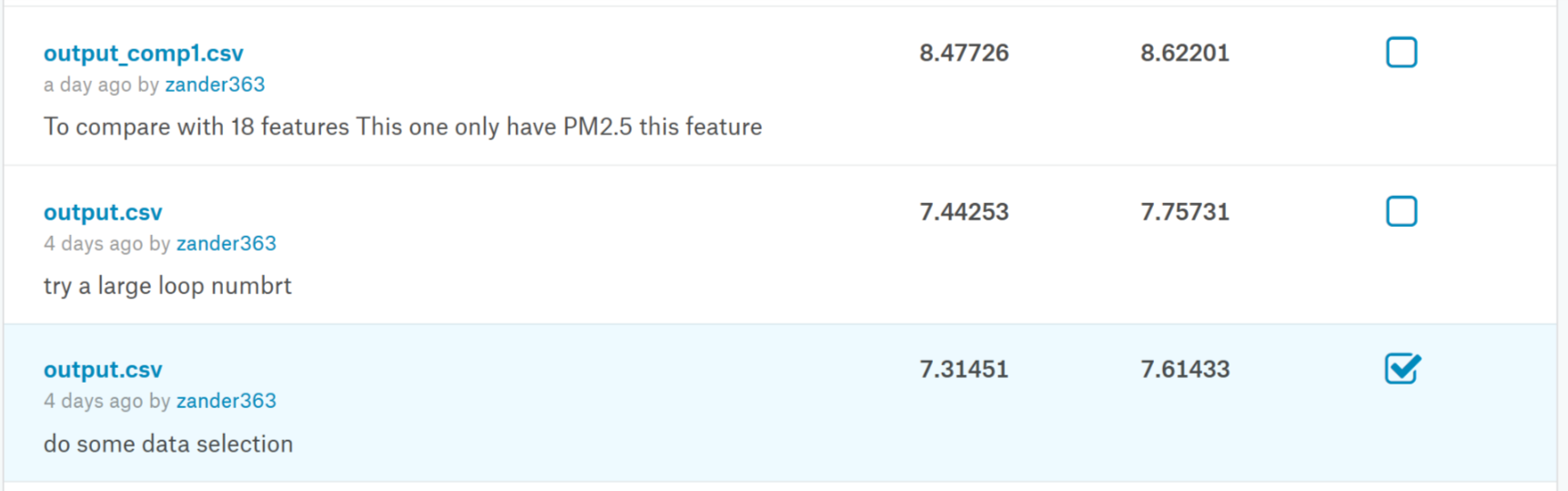
學號：b04505021 系級：工海三 姓名：黃廉弼

1. (1%) 請分別使用每筆data9小時內所有feature的一次項（含bias項）以及每筆data9小時內PM2.5的一次項（含bias項）進行training，比較並討論這兩種模型的root mean-square error（根據kaggle上的public/private score）。  
  
在固定learning rate為0.5

Iteration number 為 100000的情況下

用全部feature:  


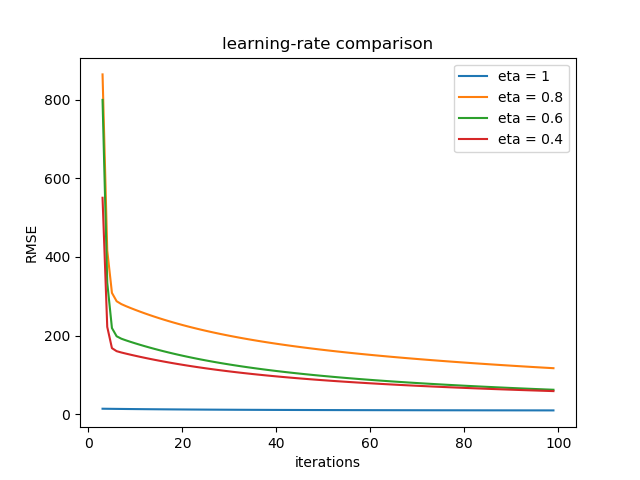
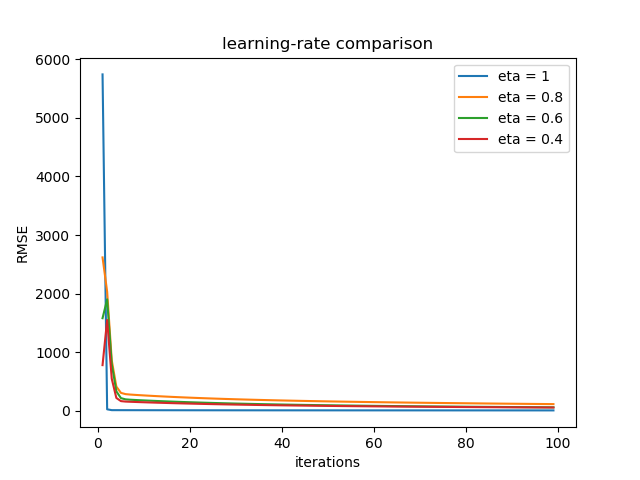
僅用PM2.5:



很明顯可以看出 僅用PM2.5下去train出來的結果 不論是public score還是private score都硬生生比用所有feature的表現差了一截。  
這樣的結果 可說是合情合理，畢竟PM2.5的數值可能受很多不同feature影響，因此直接排除其他feature勢必會降低預測的準確性。  
但話又說回來，在捨棄了17項的feature後，在RMSE上卻只掉1左右，其實也算是表現非常不俗了。可見PM2.5本身對後續PM2.5值的影響是十分可觀的。

2. (2%) 請分別使用至少四種不同數值的learning rate進行training（其他參數需一致），作圖並且討論其收斂過程。

執行結果如下圖，討論請見下頁 :



左圖為取learning rate在1, 0.8 ,0.6 ,0.4 等四種不同learning rate在100個iteration中所得到的iteration-RMSE折線圖。

從圖中可以看出，最一開始4個不同線各自散佈在不同的點，接下來在5個iteration 內 eta=1的線便掉到最低，看似趨於收斂的位置。

而為了方便觀察，我從第3個iteration開始作圖得到右圖，可看出在經過一翻抖動後eta=0.4的紅線的error竟小於eta=0.8與eta=0.6，但好景不常，然而在差不多經過80次iteration時，出現了0.4與0.6的黃金交叉，這點就展現出了learning較大的優勢，但0.8那條可能是一開始噴得太可怕了，因此還要好一段時間才能超英趕美。

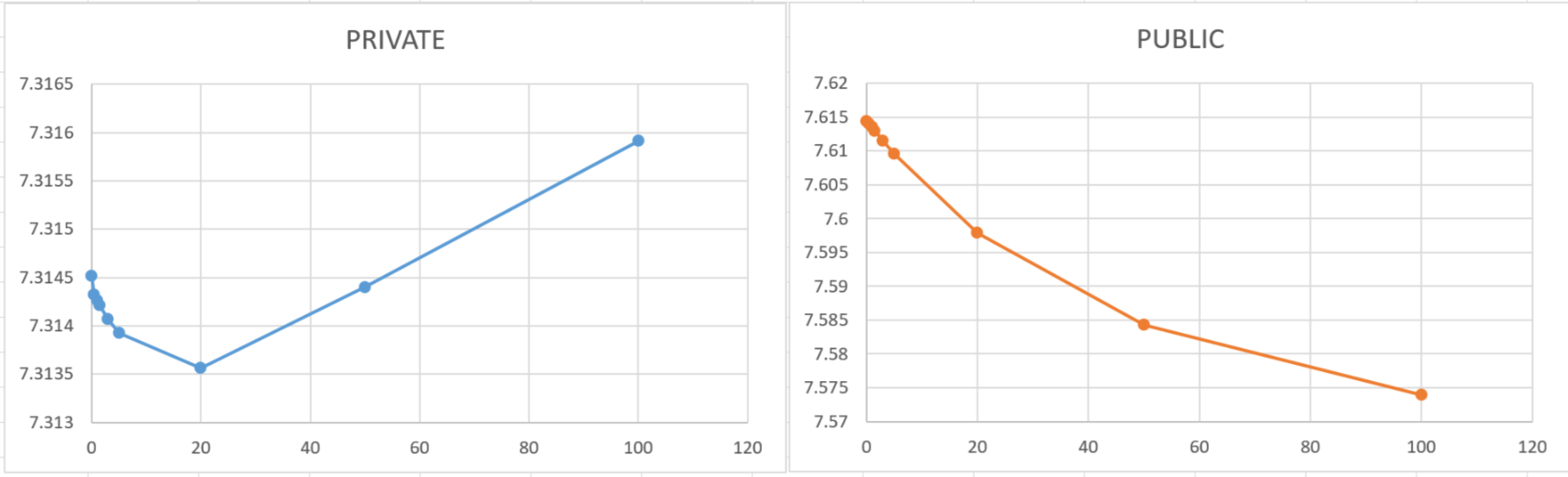
因此可以得出兩個結論:

* Eta較小 一開始暴增的範圍也比較小
* Eta較大Loss decay的速度較快

3. (1%) 請分別使用至少四種不同數值的regulization parameter λ進行training（其他參數需一致），討論其root mean-square error（根據kaggle上的public/private score）。

由於取點較多，故不逐一附圖，僅表列結果並無條件捨去至小數點後四位

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| lambda | 0 | 0.5 | 1 | 1.5 | 3 | 5 | 20 | 50 | 100 |
| Private | 7.3145 | 7.3143 | 7.3142 | 7.3142 | 7.3140 | 7.3139 | 7.3135 | 7.3144 | 7.3159 |
| Public | 7.6143 | 7.6140 | 7.6135 | 7.6130 | 7.6114 | 7.6095 | 7.5978 | 7.5842 | 7.5739 |



由圖可觀察出隨lambda增加，RMSE會逐漸下降，但下降幅度也漸緩。

而private score在lambda大於20後會暴增，而public socre卻不會。看起來是public的最低點還要再往後才會碰到。

此外，雖說regulization會幫助減少Error，但很明顯其減少的幅度相當有限，因此此方法僅能當作優化的一部份，最其根本還是要設定一個有效的model才能使Error有效降低。

4. (1%) 請這次作業你的best\_hw1.sh是如何實作的？（e.g. 有無對Data做任何Preprocessing？Features的選用有無任何考量？訓練相關參數的選用有無任何依據？）

這次實作主要是用課堂上教到的linear regression

此外，原本為了判斷各feature的order，在圖像化顯示出各項feature與PM2.5關係時，意外發現似乎有一些點是不太正常的。因此借了老師上課提到的3-folder cross的觀念，來去除包含不正常數據的資料，雖然會因此損失掉能training的data量，但在用training data測試 以及kaggle評估結果都有亮眼的表現，大大的降低了error。

此外，原本從散布圖看來O3的分布集中在兩條線上，原本判斷該用二次項，但實際run起來似乎沒有什麼幫助，因此猜想或許是O3實際上是更高次方項。