

(b). 我將 thread 開在最後 test data 去跑 random forest 的過程，每個 thread 分別去跑一定數量的 test data。

(c). tree 數量固定在 10001

thread 數量	2	3	4	5	6	7	8	9	10
時間 (sec)	10.55	11.64	10.97	10.52	9.39	8.20	10.36	10.19	13.02

因為我開 thread 的地方不太需要大量的運算時間，於是用多 thread 時能省的時間有限，但是仍可以看出有分別，而在 thread 超出一定數量時可以看到時間明顯變慢，那是因為開 thread 很花效能，若沒有讓 thread 用在很需要大量運算的地方的話反而會變慢。

(d). tree 數量固定在 10001

thread 數量	2	5	10	20	50
instructions 數量	28739494902	25490866920	25275361114	23885093147	28069009756

用多少 thread 出來的 instructions 數量都差不多，因為使用 multi-thread 運行時只是將許多操作平行去處理，但是跟單一 thread 所需要用到的操作數量是一樣的。

(e). thread 數量固定在 7

tree 數量	10	100	1000	10000	100000
instructions 數量	2,292,095,561	3,026,596,348	6,408,339,421	27,814,197,276	38,327,032,947

開越多 tree 會使得需要更多的操作去建立 tree，從圖表中可以看到 tree 的數量每放大 10 倍，instructions 數量都放大 1.4~2.1 倍不等。

(f). 在嘗試每顆 tree 所取的 id 數量時發現，在一定 tree 的數量下，大概取超過 30 個 id 之後得的正確率都差不多，但是速度明顯變許多，可能是因為我計算 gini impurity 時太過繁瑣。於是我們將 id 數控制在 30 左右，然後靠著增加 tree 的數量來提高正確率，發現大約在 5000 以上時正確率會趨於穩定。