本次報告所用的資料分別是iris, wdbc, glass, wine

**有無用attribute bagging的時間和精準度差別:**

Iris(4 attribute):





Wbdc(31 attribute):

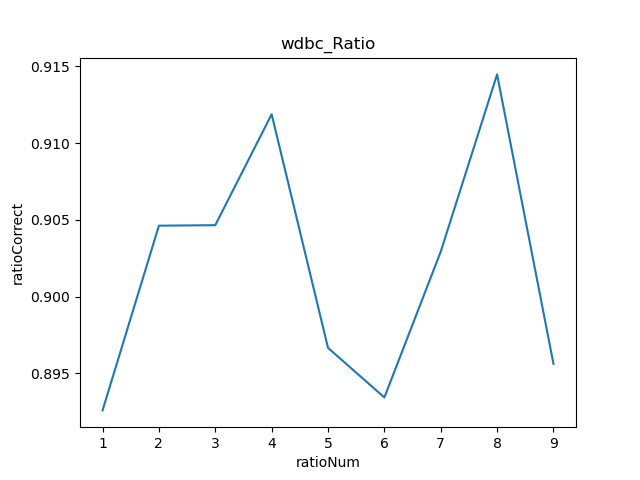
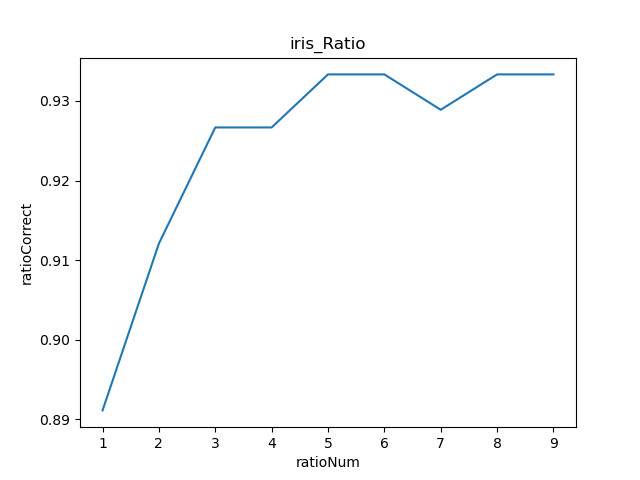


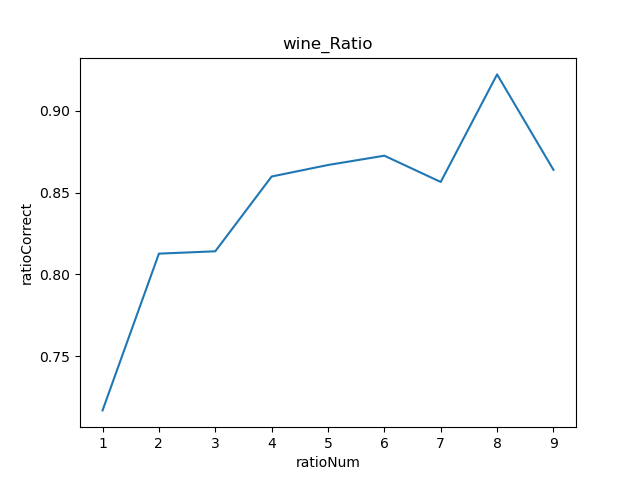
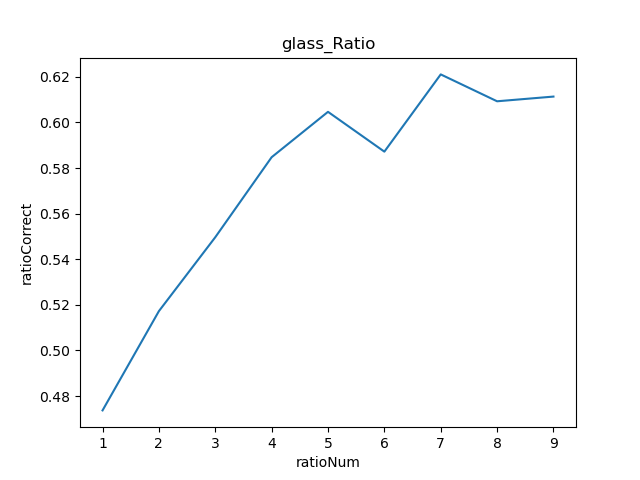


在attribute多的情況下，wbdc的速度有著非常明顯的加速，但精準度基本上卻沒什麼差別，因此之後的實驗皆會採用bagging。

**Relative sizes of the training and validation subsets:**

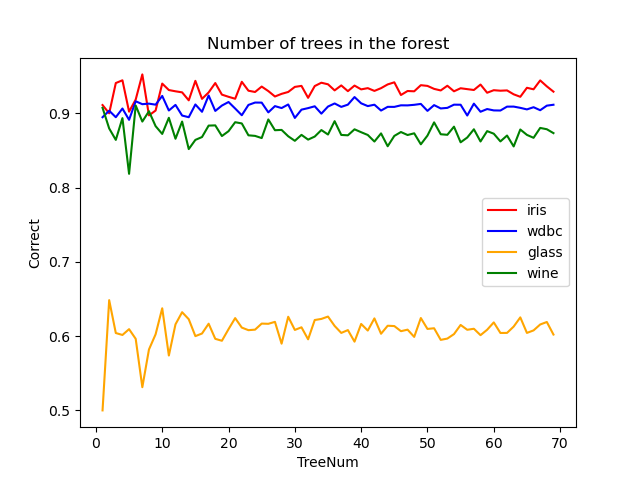
※Random forest tree number=20





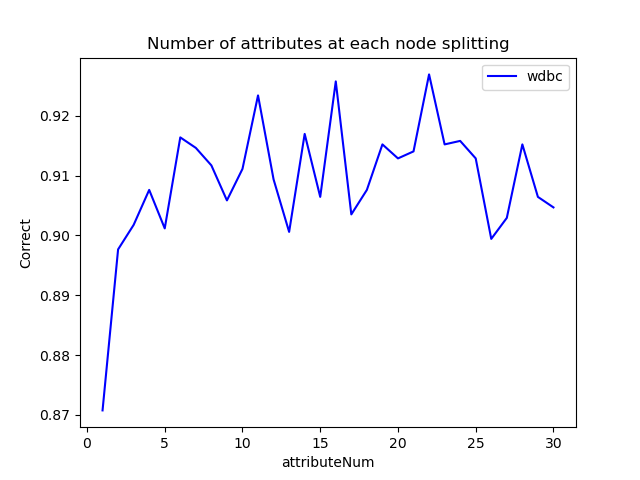
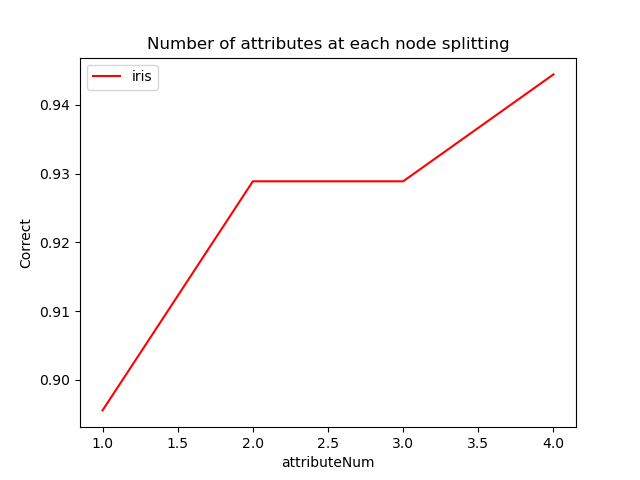
從兩種不同的data可以看得出來，當Relative sizes of the training and validation subsets大概到7:3的時候準度是最高的，而之後隨著比例愈高，準度很不穩定的原因我覺得是validation subset 的數量太少了，很容易就overfitting。之後測試的比例皆為7:3。

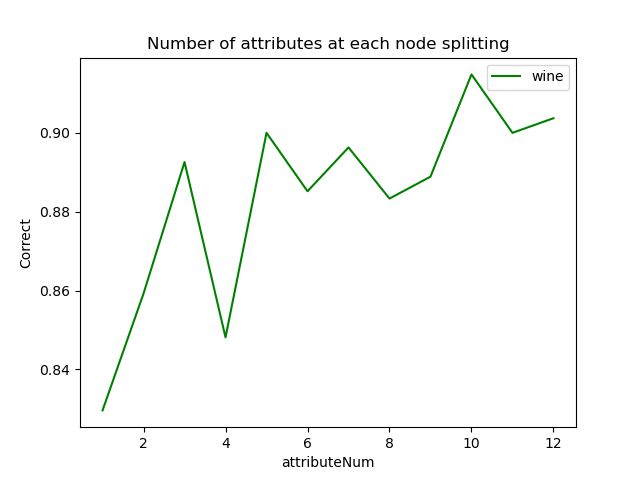
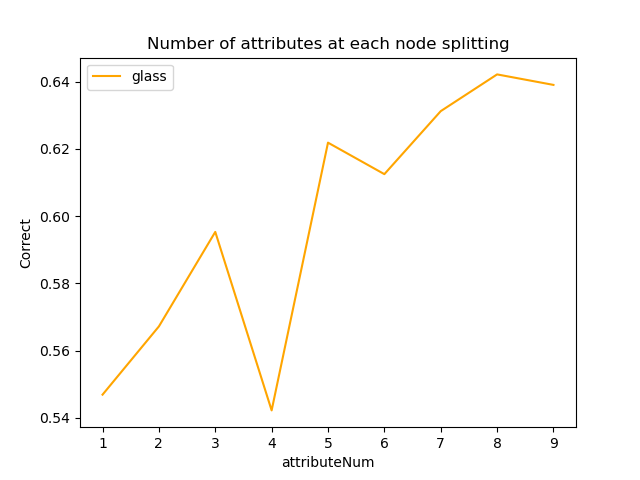
**Number of trees in the forest:**



可以看出，隨著森林的數目越多，準確度越趨近於穩定。準確度沒有隨著樹的數量而提高，我覺得是因為data的數目太少了，所以即使樹再多，抽到同一批data的機率也蠻高的。因為TreeNum在這裡的影響不大所以下一個實驗就使用TreeNum=10。

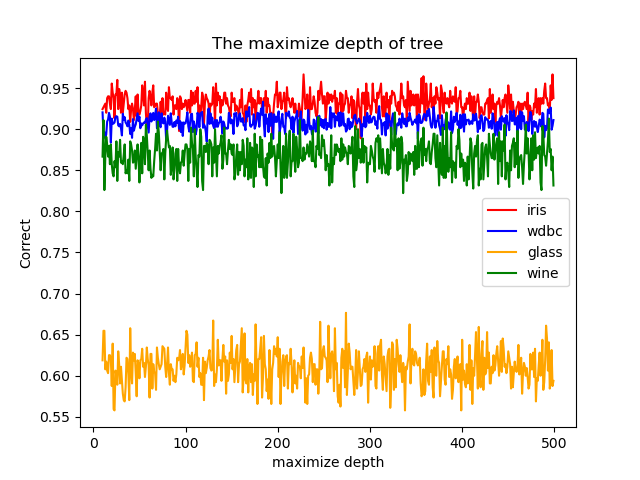
**Number of attributes at each node splitting:**





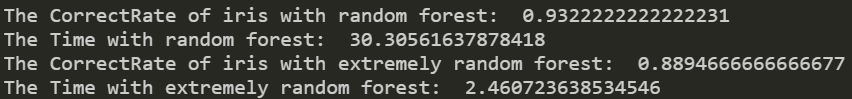
在直覺上，attribute所採用的數量愈多應該會越準，而實驗出來的結果也在我的預期內，雖然每一個data幾乎都在採用所有的attribute時精準度都有稍微下降，但大致上來看，attribute的確採用的越多準度越高。但即使如此，這所付出的時間代價太大了，從第一個實驗就可以知道，採用根號的數量和全部的數量，準度其實不會差太大，然而時間卻是平方倍，所以在準度與效率的取捨上，取根號還是最優的方法。

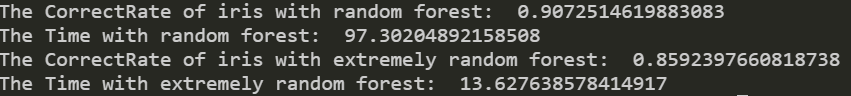
**Methods that limit a tree's size: upper bound on the tree's depth:**

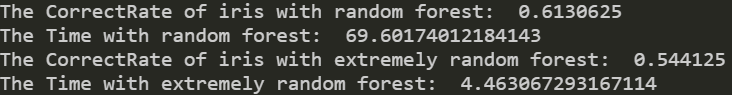


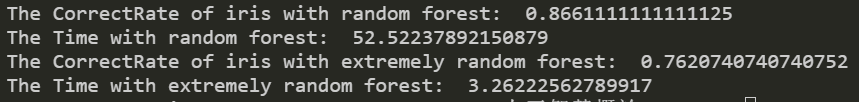
做出來的實驗結果跟自己預期的差距蠻大的…，本來想說深度越淺時，準確度越低，結果實驗顯示是差不多的，我想了一會兒，後來我試試看那些資料通常深度會到多深，結果發現幾乎都在15以內，所以最大深度的測試測到500是完全沒意義的...(跑了半小時以上)

**Extremely random forest:**









由上而下依序是iris、wdbc、glass、wine的結果。

結果顯示，extremely random forest的精準度並沒有比random forest差太多，反而在時間上是random forest的好幾倍(畢竟省下了算gini的時間)，所以在取捨上，如果不是精準度要求很高，我覺得選擇extremely random forest的方法也是一個不錯的選擇。

**Summary:**

經過了上面6個實驗，大概可以得出結論，就是forest 的tree number不需要太高大概是資料量的10%就好了，而如果是追求運算效率的可以選擇extremely random forest，而如果是追求準度則可以選擇random forest，還有train data : validation data=7:3最好，最後選擇在attribute的數量上還是建議根號就好。

這次的作業雖然是我寫過最多行的作業，但也是我覺得最有趣的(除了group project)，我想是因為寫起來沒有遇到太多困難，像之前幾次的作業，光是Debug的時間就比寫Code的時間還要來的多了…，這學期下來學到了很多人工智慧的東西，我覺得比起讀那些理論，實際做過一次印象才深刻，讓理論走出課本讓這門課變得十分有趣!

**附錄:**

我的code:

<https://github.com/KaivinC/AI_HW4>