1. Training and testing set has been used to train. The result has been summarized as below:

|  |  |  |
| --- | --- | --- |
| n\_estimators | Training Set  Accuracy score | Testing Set  Accuracy Score |
| 1 |  |  |
| 12 |  |  |
| 100 |  |  |

We can see the trend easily for both training and testing data set. When *n\_estimators=1*, the accuracy score is the lowest as it just generates 1 tree from large data set and it has low reliability. The accuracy score starts to increase when *n\_estimators=12* as more trees were generated as more attributes were considered. However, when *n\_estimators=100* the score goes down slightly as there are too much trees and it may make the outliers become significant.

Apparently, testing data set gets higher score than training because testing set contains more data. Although it may contain more outliers, since random forest won’t overfit, it won’t influence the accuracy score.

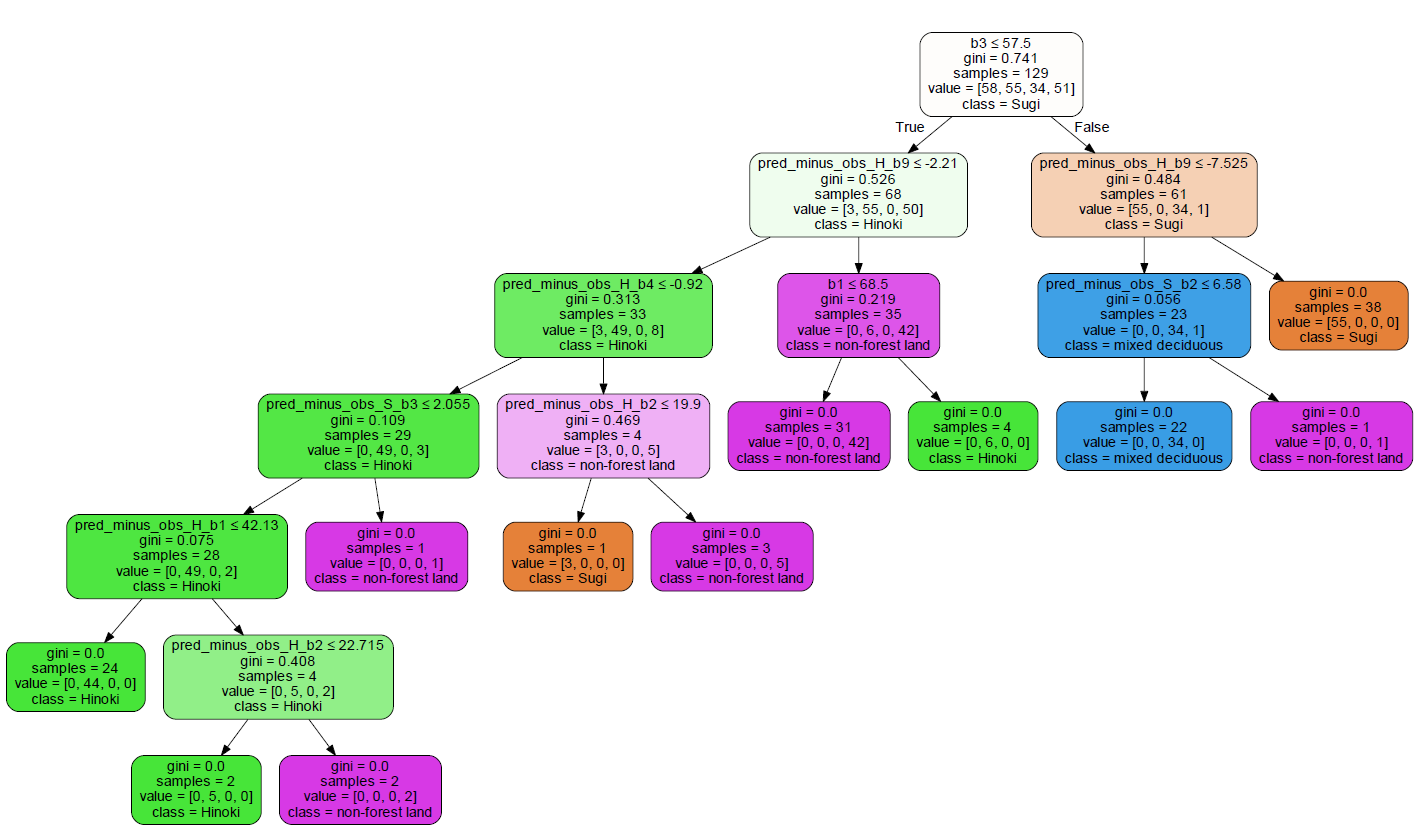
Random forest doesn’t have the overfit problem; however, This comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance in the final model.

The accuracy score for original optimal tree and each tree has been summarized as below:

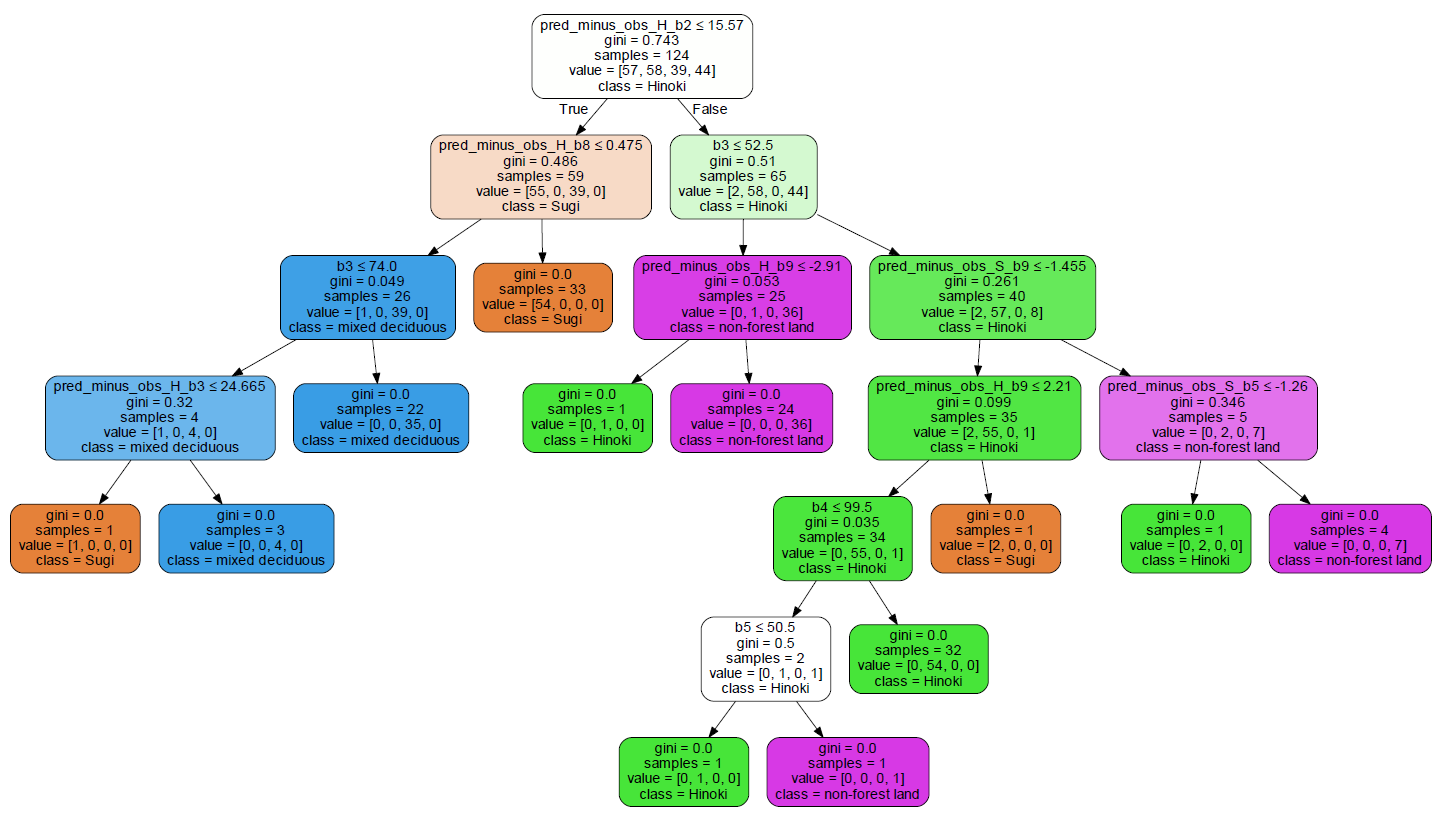
|  |  |
| --- | --- |
|  | Training Set  Accuracy score |
| Original Random forest model |  |
| Component Trees | |
| A |  |
| B |  |
| C |  |

By select *n\_estimators=12* with the highest performance, it shown that all the component trees that used to form the final random forest model has lower accuracy score than the final model. Looking more closer to each individual component trees, it is found that all these component trees contain redundant roots and are overfitting the model.

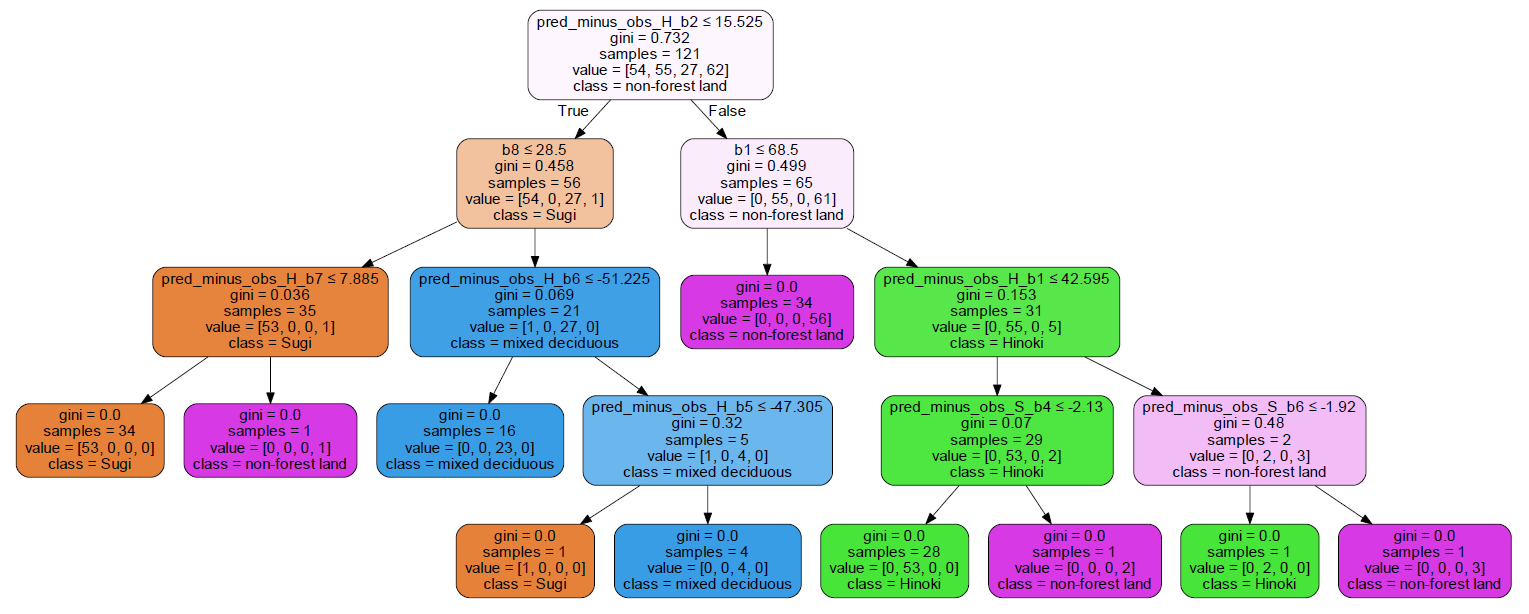
Below are the component trees A, B and C which is selected:



Tree A



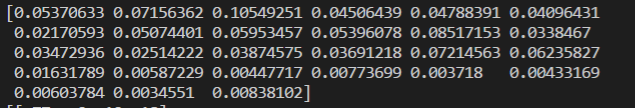
Tree B



Tree C

The roots that are circled red are redundant, because they are overfitting the particular randomly selected data set. These roots will be removed when they are combined to for the final random forest model.

(c)



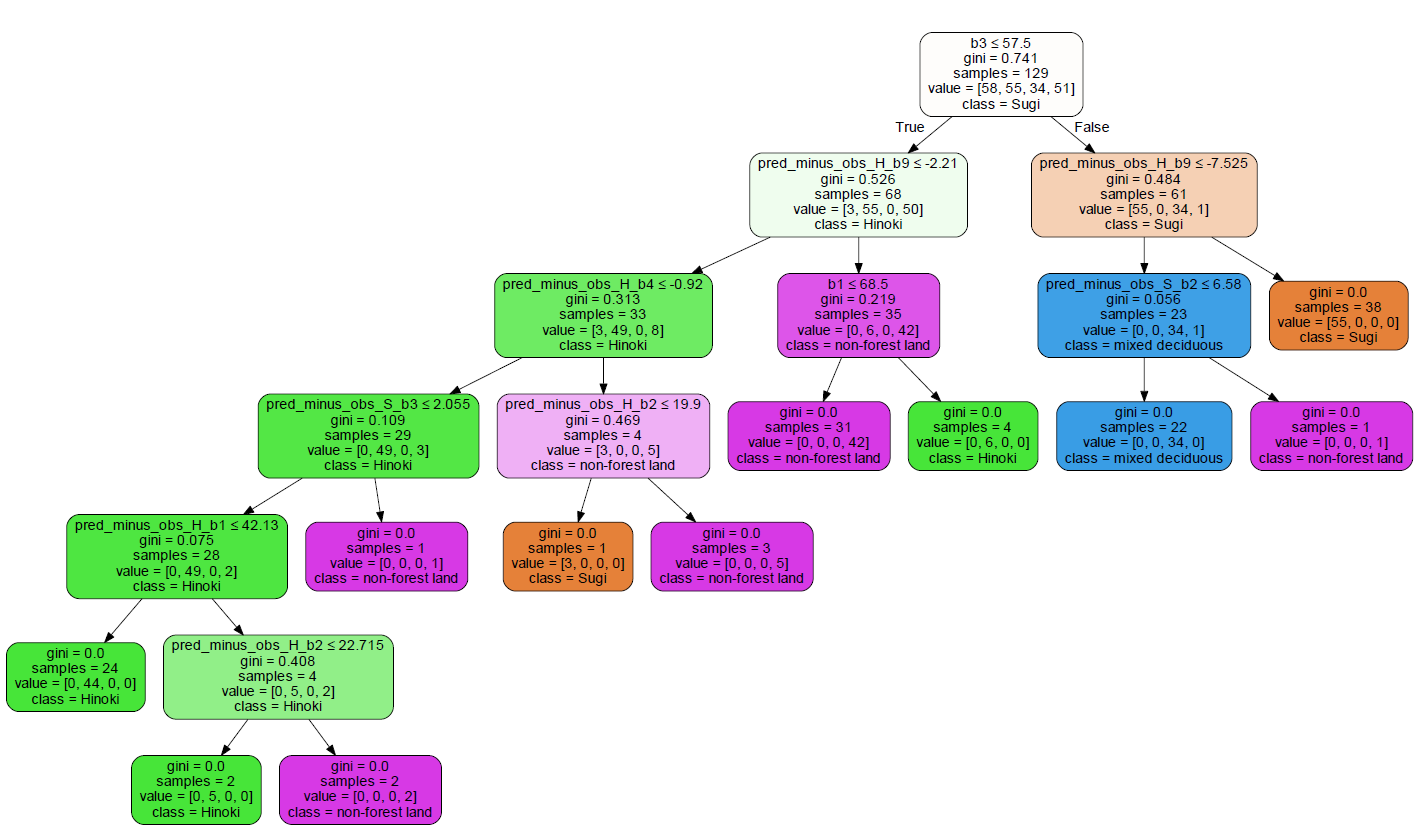
This corresponding array values are the feature importance of the final random forest model. The corresponding attribute is:

[b1,b2,b3,b4,b5,b6,b7,b8,b9,pred\_minus\_obs\_H\_b1,pred\_minus\_obs\_H\_b2,pred\_minus\_obs\_H\_b3,pred\_minus\_obs\_H\_b4,pred\_minus\_obs\_H\_b5,pred\_minus\_obs\_H\_b6,pred\_minus\_obs\_H\_b7,pred\_minus\_obs\_H\_b8,pred\_minus\_obs\_H\_b9,pred\_minus\_obs\_S\_b1,pred\_minus\_obs\_S\_b2,pred\_minus\_obs\_S\_b3,pred\_minus\_obs\_S\_b4,pred\_minus\_obs\_S\_b5,pred\_minus\_obs\_S\_b6,pred\_minus\_obs\_S\_b7,pred\_minus\_obs\_S\_b8,pred\_minus\_obs\_S\_b9]

The feature importance which have its’ score larger than 0.6

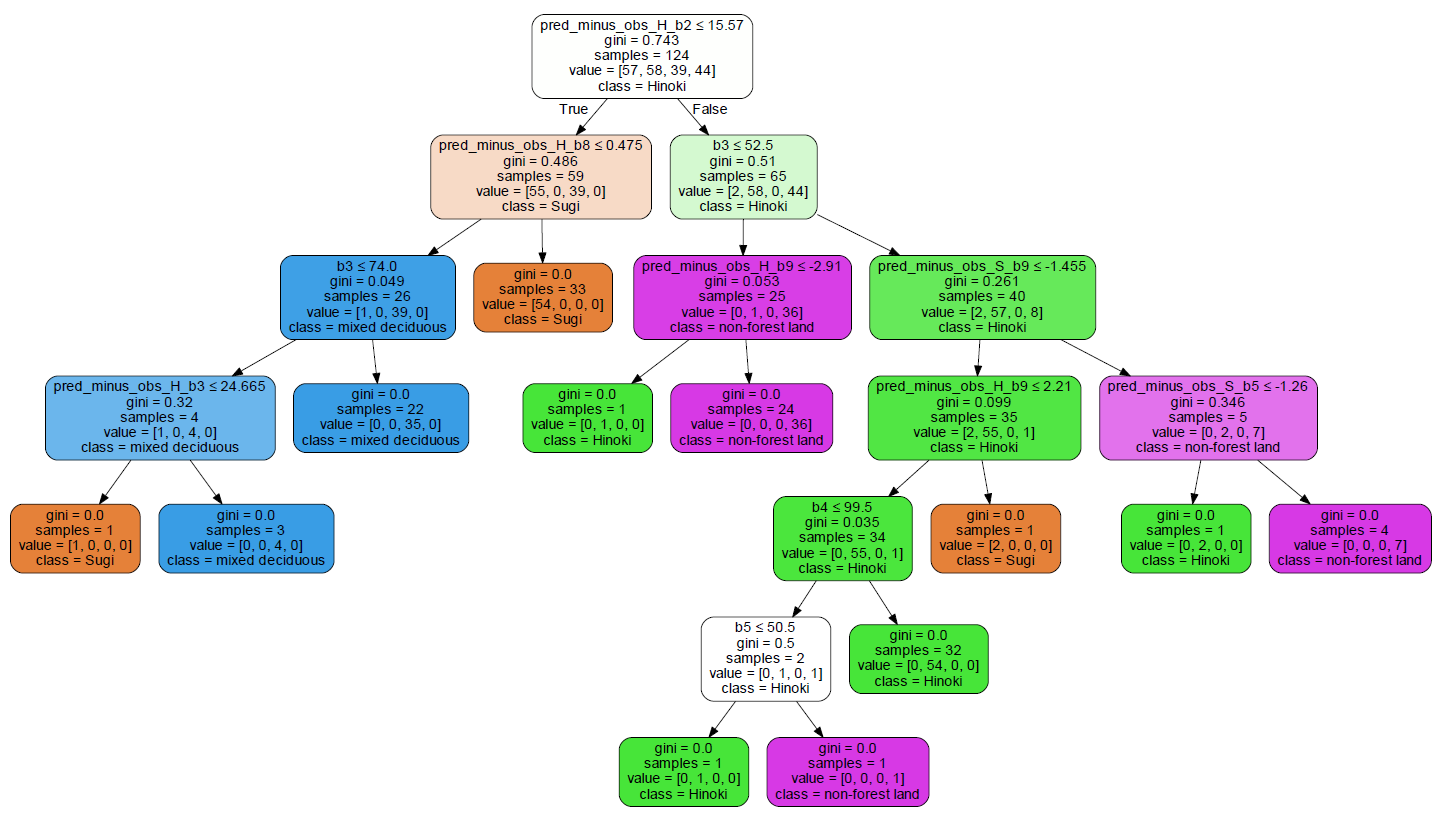
|  |  |
| --- | --- |
| Feature | Importance value |
| B3 | 0.10549251 |
| pred\_minus\_obs\_H\_b2 | 0.08517153 |
| pred\_minus\_obs\_H\_b8 | 0.07214563 |
| B2 | 0.07156362 |
| pred\_minus\_obs\_H\_b9 | 0.06235827 |

Having a closer look on the Test A, B and C:



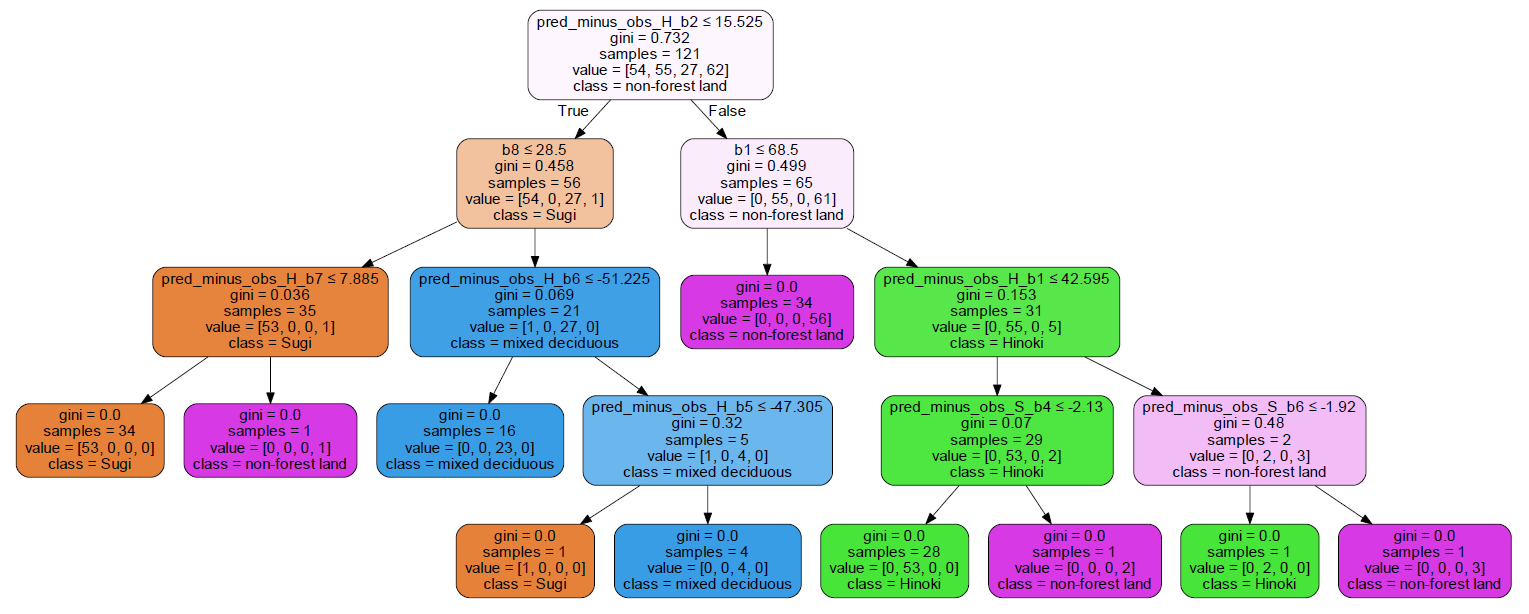
Test A

The most important features are: b3, pred\_minus\_obs\_H\_b9



Test B

The most important features are: pred\_minus\_obs\_H\_b2, pred\_minus\_obs\_H\_b8, b3



Test C

The most important features are: pred\_minus\_obs\_H\_b2, b8, b1

By comparing the data from the test component trees and most important features identified from the random forest model, we found out that the most important features of the component trees are usually the same as the random forest model. This is because the random forest model is formed by combining all these trees, these features also generalised the decision tree for the whole data set, as they are the most important features.

Another thing to note from the component trees is that, although the head of the tree (the starting point of the tree) is very different when all trees are compared to each other, some tends to repeat. The more the repetitions of the trees, the more important it is when it is used to combined. The frequency of appearance of an attribute on the starting point, signify the importance of the attribute.

Looking at the previous discussion, the trees who have all those redundant roots (overfitted roots) also contribute to the lost of the classification performance. These roots is unique from component trees to component trees, as the data set is randomly picked from the larger data set, these roots tends to be even unique. When combining, these parts will be removed as their frequency tend to be too low, these contribute to higher classification performance of the final model.