CS 3481
Fundamentals of Data Science

Assignment-2
Random Forest and Naive Bayes Classifier

Table of Contents

Name: Avi Malhotra
Student ID: 55773896

Part A

Construct random forest models using different numbers of component trees based on the default training set/test set partition, and analyze the resulting change in classification performance. (25%)

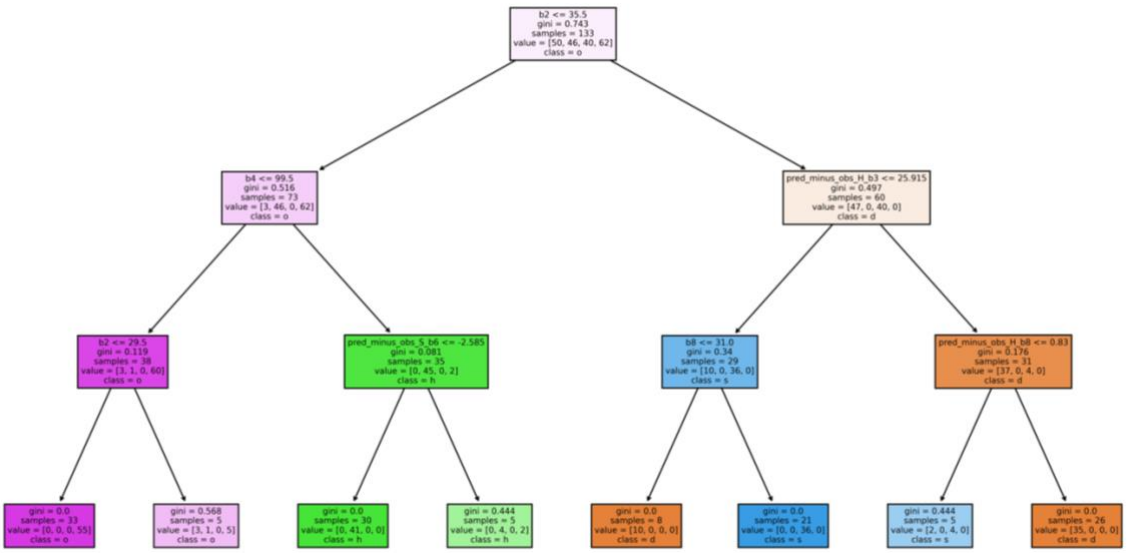
The number of component trees in the random forest models were varied by altering the value of n_estimators in the random forest classifier function. For effective comparison, all other filters were fixed as follows: criterion="gini", max_depth=5, min_samples_leaf=5.

For practicality and the purpose of this assignment, the value of n_estimators was kept in the range of 2 to 10, with increments of 2. While commendable random forest trees may be constructed with higher number of component trees, an analysis of such a random forest would be nearly impossible for subsequent sections of this assignment (such as part b).

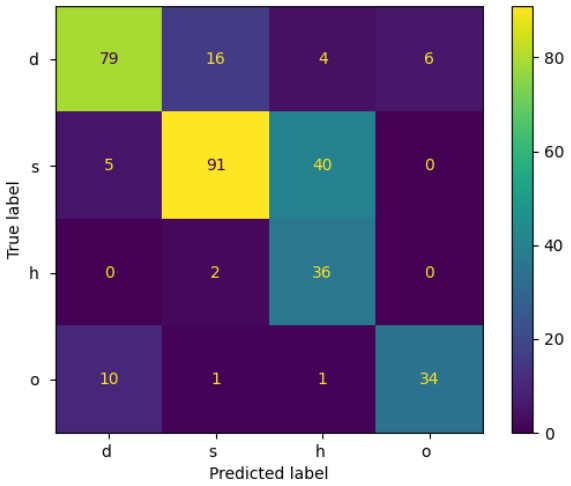
The following are the random forest models and their corresponding confusion matrices:

n_estimators = 2

Accuracy: 73.8%
Random Forest Model:



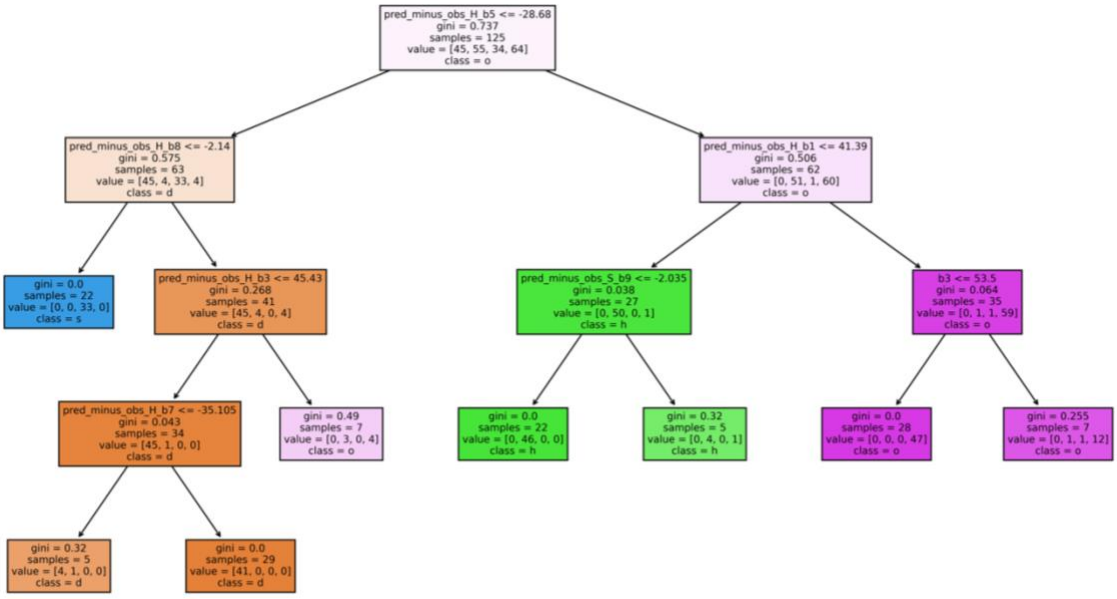
Confusion Matrix:



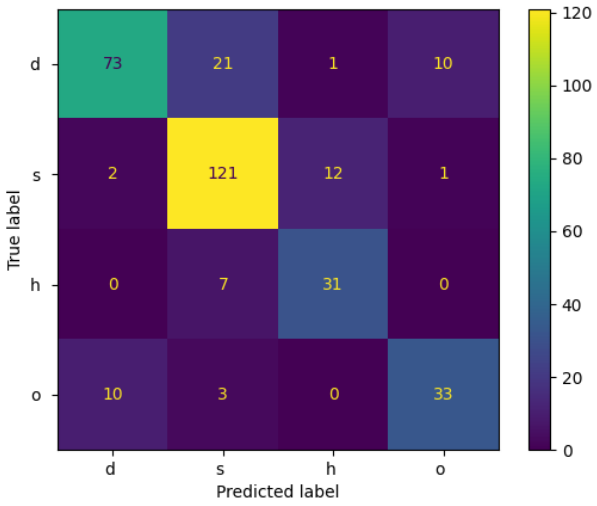
Analysis:
This random forest model has the lowest accuracy of the set, as evidenced by the absurdly high count for the confusion pair s-d (i.e. node ‘s’ is mislabeled as node ‘h’ by the model). This model is also too simplistic, as it utilizes a depth of only 3, as opposed to the maximal permissible depth of 5.

n_estimators = 4

Accuracy: 79.4%
Random Forest Model:



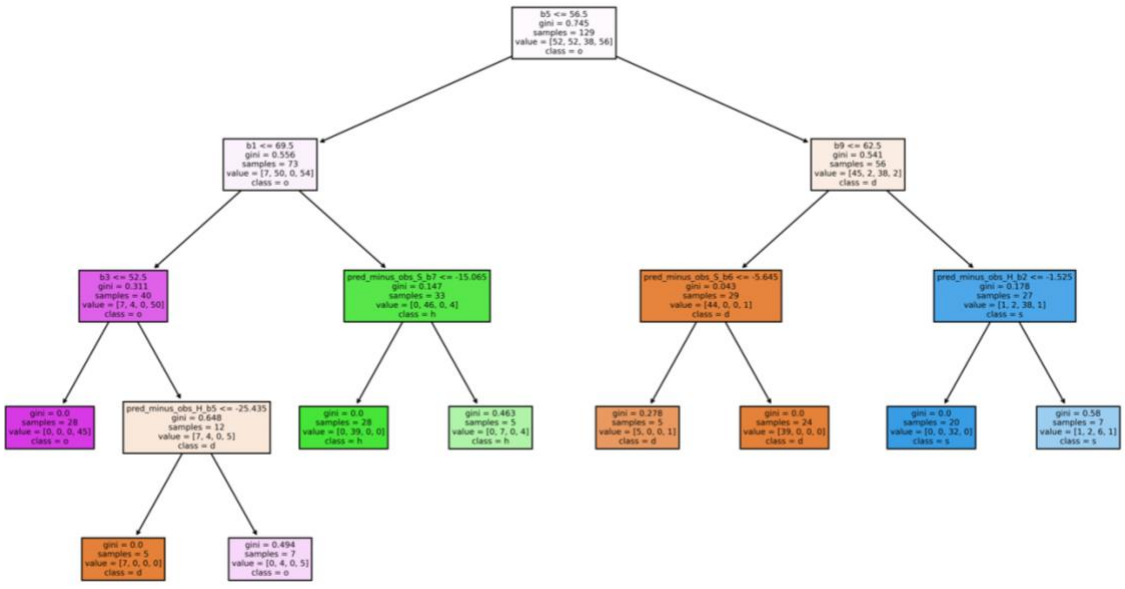
Confusion Matrix:



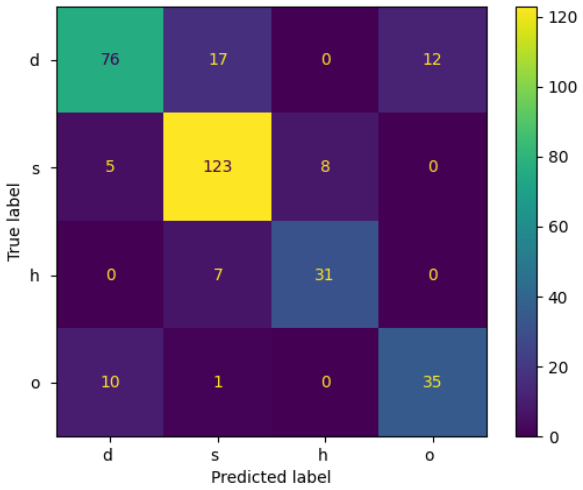
Analysis:
This model is considerably better than the first, given the significant increase in the overall accuracy. Node ‘d’ being mislabeled as node ‘s’ is the primary noticeable confusion pair.

n_estimators = 6

Accuracy: 81.5%
Random Forest Model:



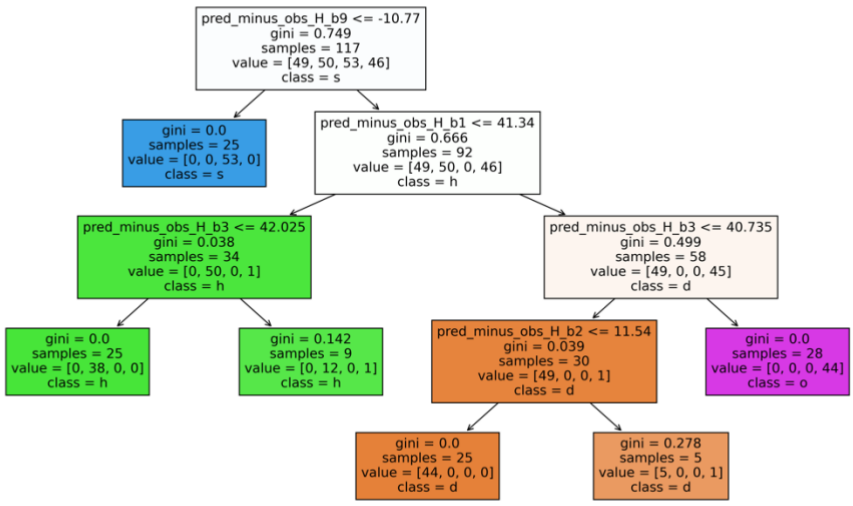
Confusion Matrix:



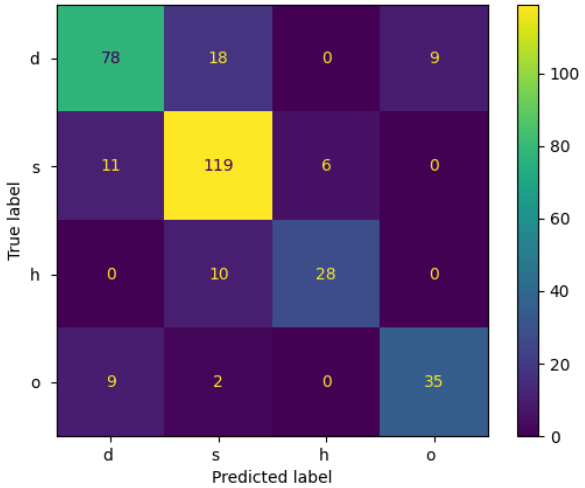
Analysis:
This model is hindered by the same d-s confusion pair as the previous one. Nonetheless, increasing the number of component sub-trees here results in an increase in overall accuracy. In fact, this model yields the highest accuracy comparative to the set.

n_estimators = 8

Accuracy: 80.0%
Random Forest Model:



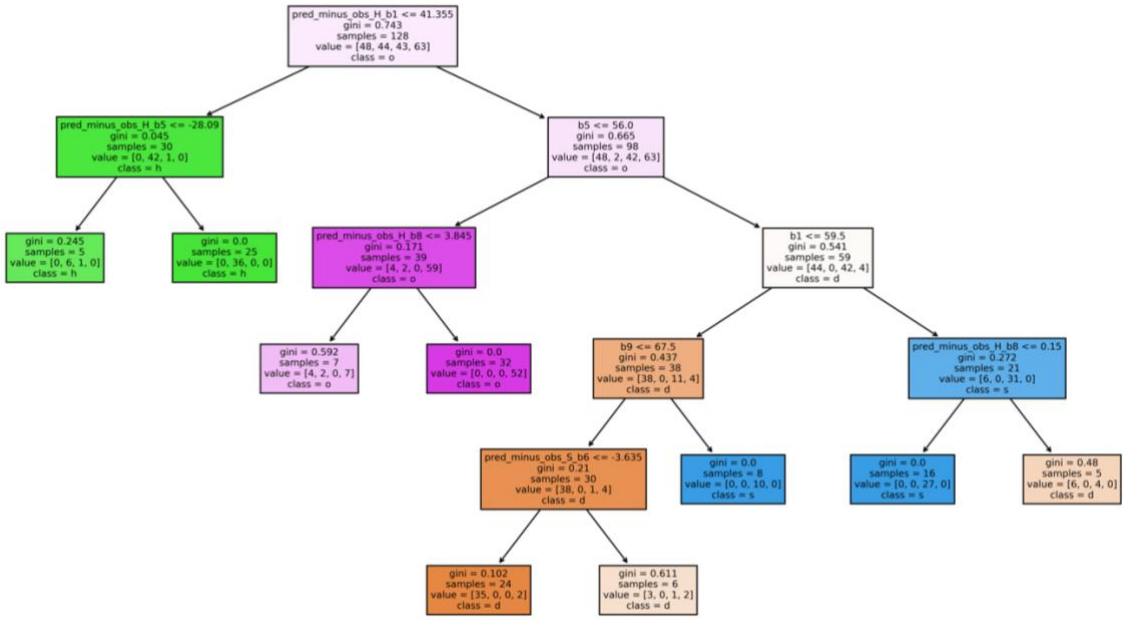
Confusion Matrix:



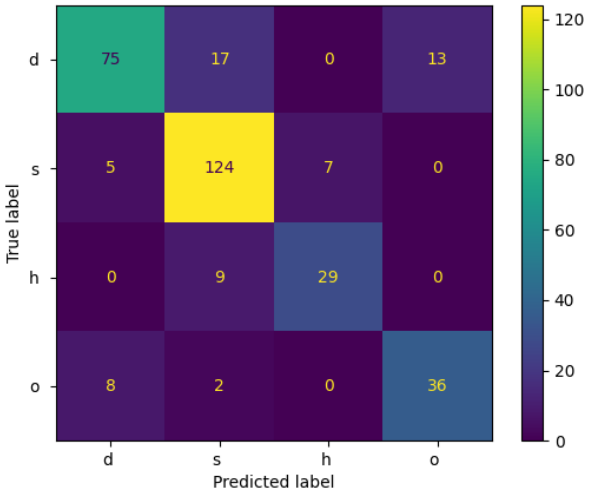
Analysis:
Again, the confusion pair ‘d’ to ‘s’ is prevalent here, though this model yields a slightly lower accuracy as compared to the previous one. A reasonable explanation for this is the fact that this model is less complicated than its predecessor.

n_estimators = 10

Accuracy: 81.2%
Random Forest Model:



Confusion Matrix:



Analysis:

This model yields the second highest accuracy of the set. It might seem that given the general trend of increasing accuracy with an increasing number of n_estimators, this model would be the best of the set. Nonetheless, the model with n_estimators has a higher accuracy. This could be due to random_state filter (defaulted to none) used in the construction of all random forest models.

Part B

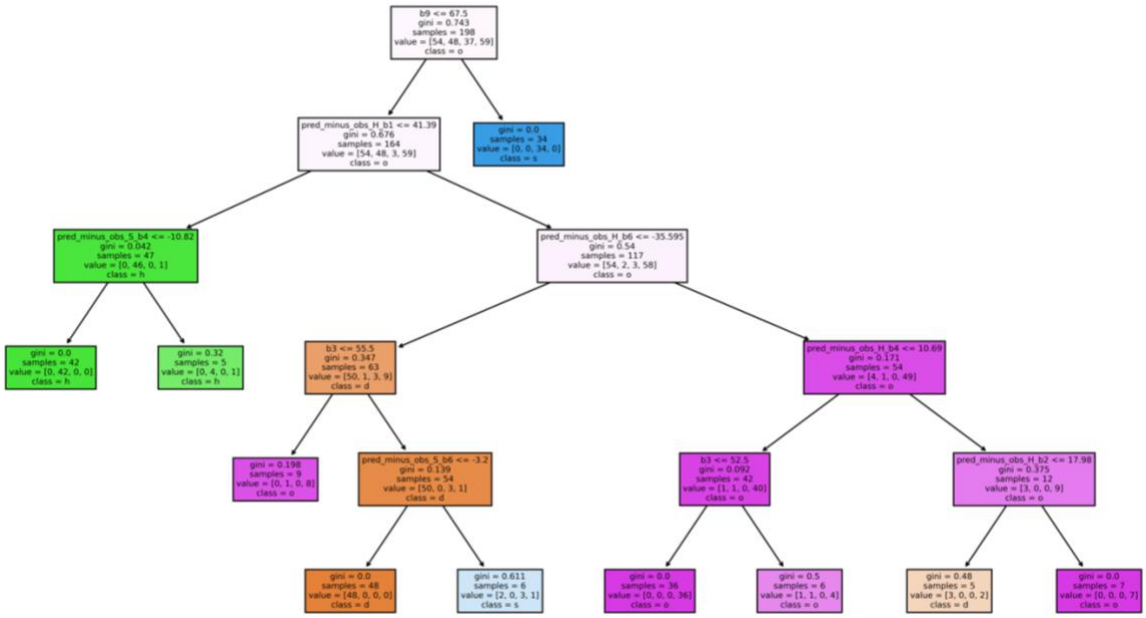
For the random forest model corresponding to the best classification performance, select different component decision trees in the model, and compare the classification performances of these trees with that of the original random forest model. (25%)

As seen from the analysis in Part A, the random forest model with 6 component decision trees (i.e. n_estimators=6) is the most accurate. Please refer to the random forest model and confusion matrix in Part A for additional details.

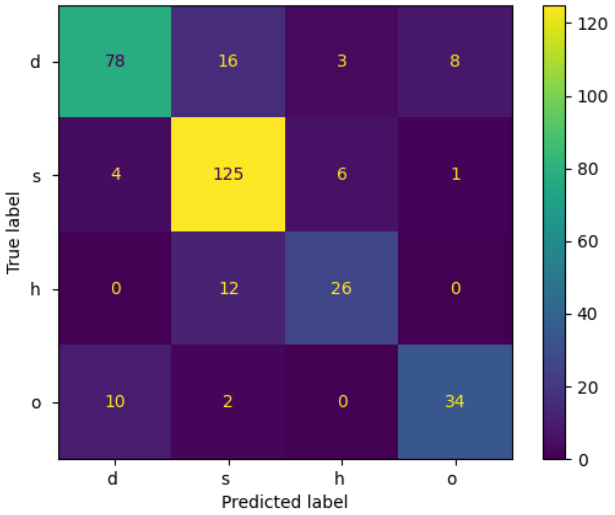
The information of its subtrees is listed below:

Component Decision Tree 1:

Accuracy: 80.9%
Decision Tree:



Confusion Matrix:



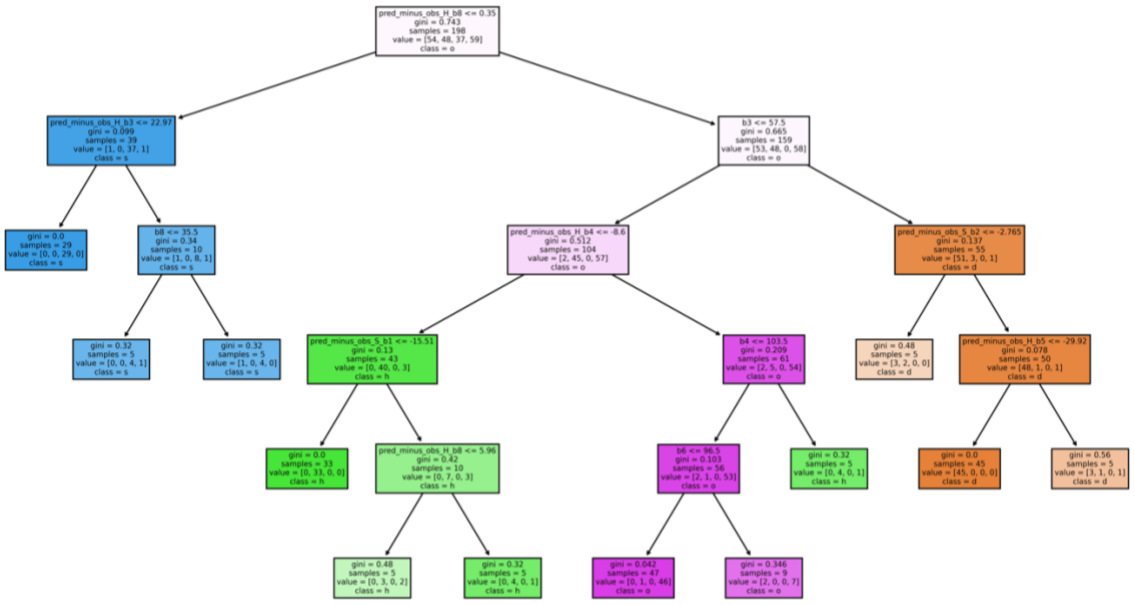
Analysis:

This decision subtree perhaps most closely models the overarching random forest model, especially in the context of the Confusion Matrix. The only visible discrepancies between the two matrices are in the case when node ‘h’ is mislabeled as ‘s’ and when node ‘d’ is mislabeled as ‘o’. Even in these cases, the counts are barely off by a few digits and hence comparatively insignificant.

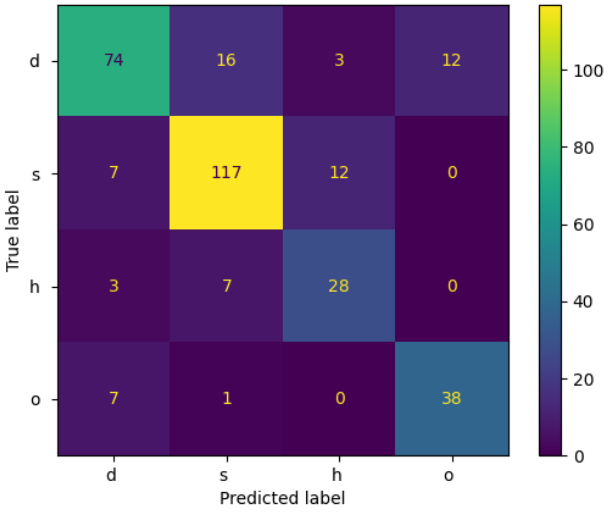
Component Decision Tree 2:

Accuracy: 79.1%

Decision Tree:



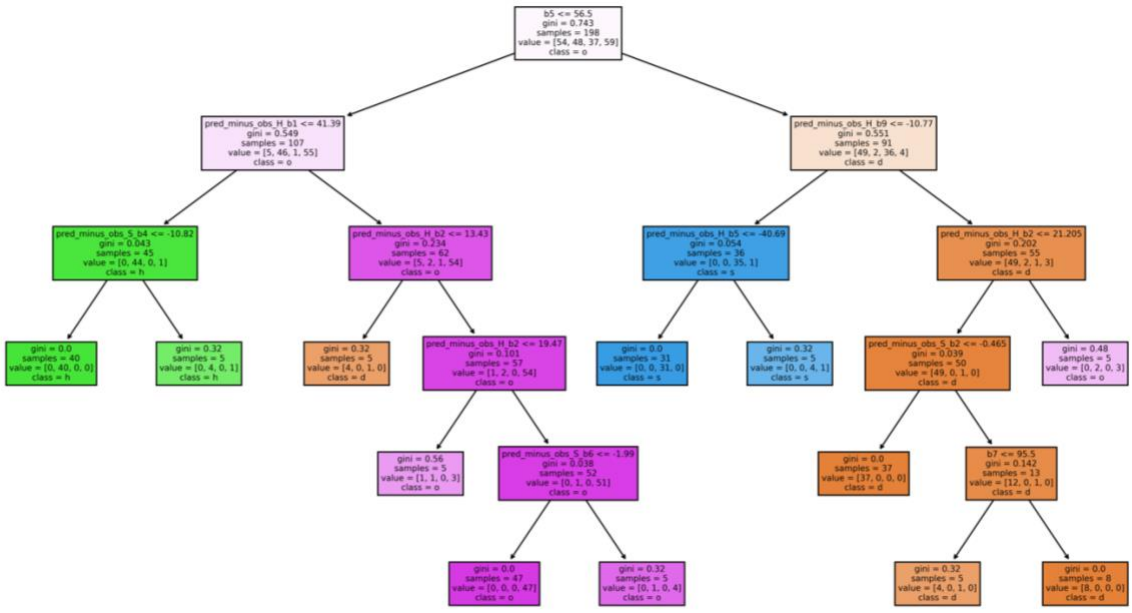
Confusion Matrix:



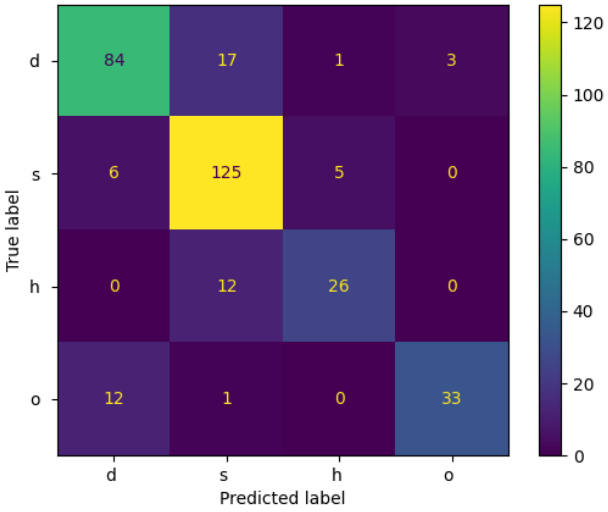
Analysis:
This decision subtree also approximates the random forest model to a great extent, almost as well as the previous one in fact when evaluating the confusion matrices. Moreover, as seen from the decision tree, many of the higher-level depths share the same parent nodes for almost every class represented. It is worth mentioning that this tree might be a suitable for post-pruning.

Component Decision Tree 3:

Accuracy: 82.5%
Decision Tree:



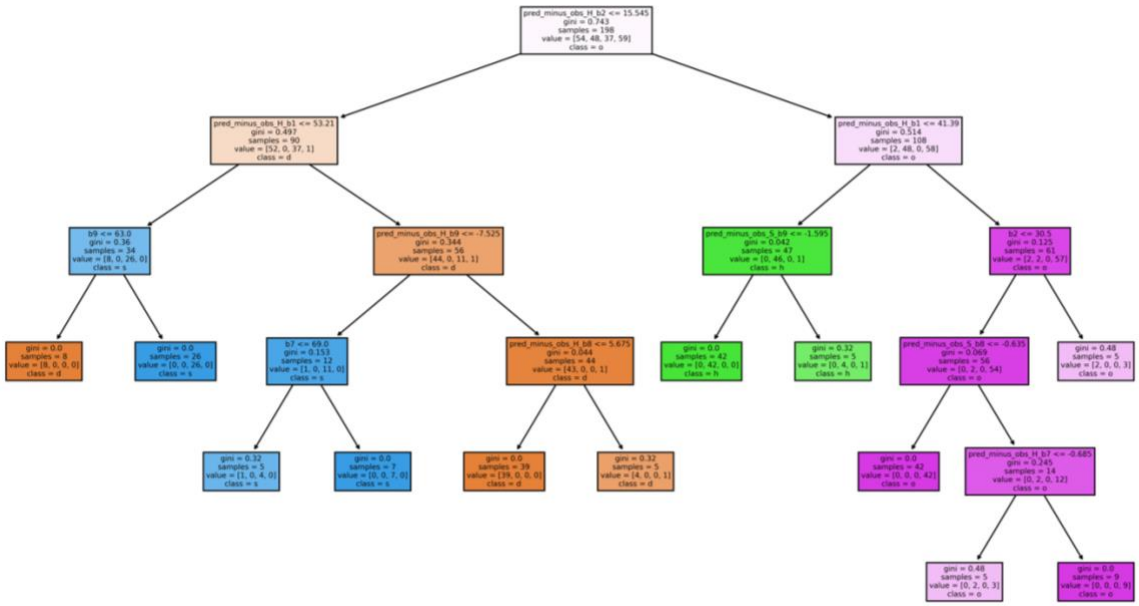
Confusion Matrix:



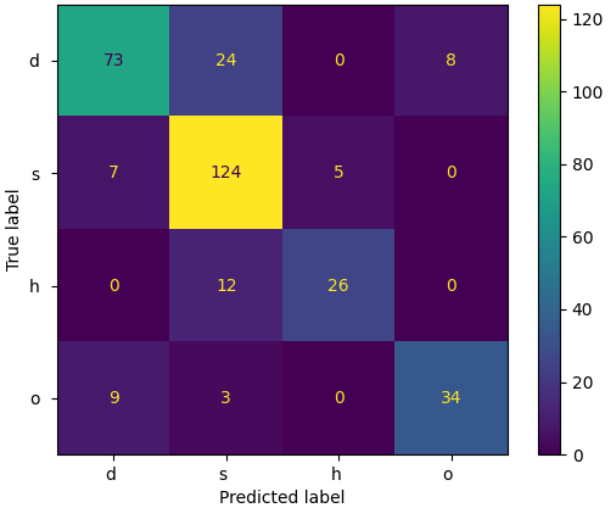
Analysis:
This decision tree has the highest accuracy of all component trees. Its accuracy is higher than the random forest tree's primarily because it identifies a higher percentage of class 'd' nodes accurately. A closer inspection reveals that this is accomplished by minimizing the number of d-o confusion pairs (12) prevalent in the random forest model's confusion matrix (vis-à-vis only 3 pairs in the subtree).

Component Decision Tree 4:

Accuracy: 79.7%
Decision Tree:



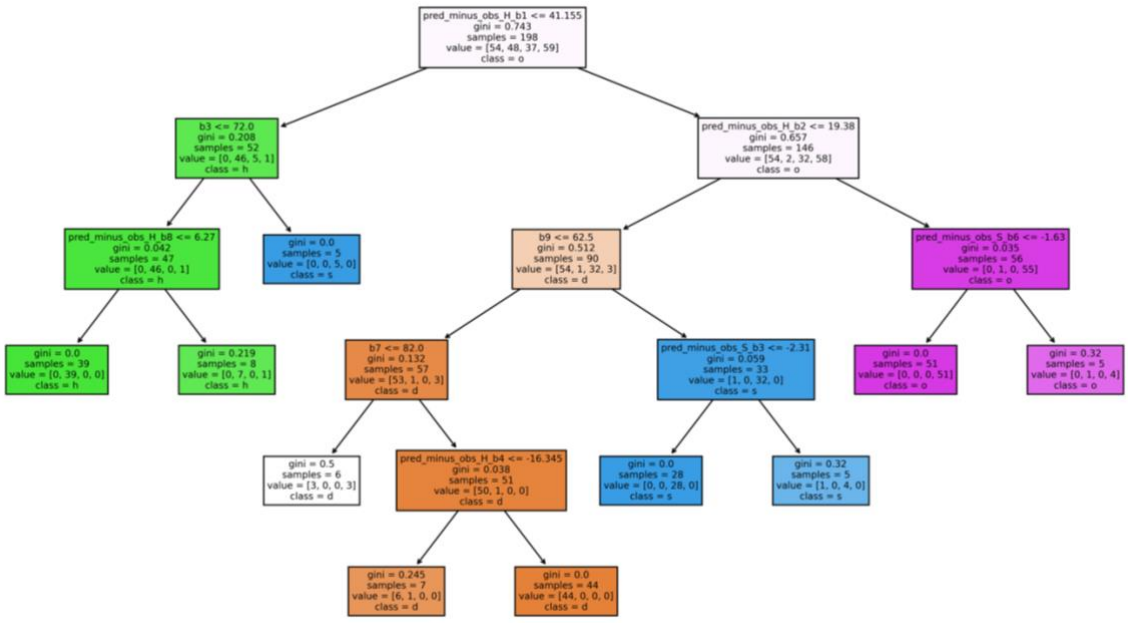
Confusion Matrix:



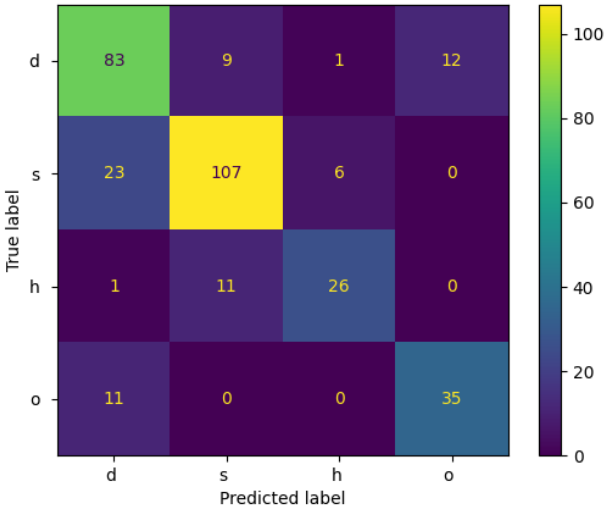
Analysis:
Even though the accuracy of this component tree is lower than that of the random forest model, the only confusion pair that most likely diminished the accuracy of this tree is the d-s pair (with a count of 24 here, as opposed to the original's count of 17).

Component Decision Tree 5:

Accuracy: 78.2%
Decision Tree:



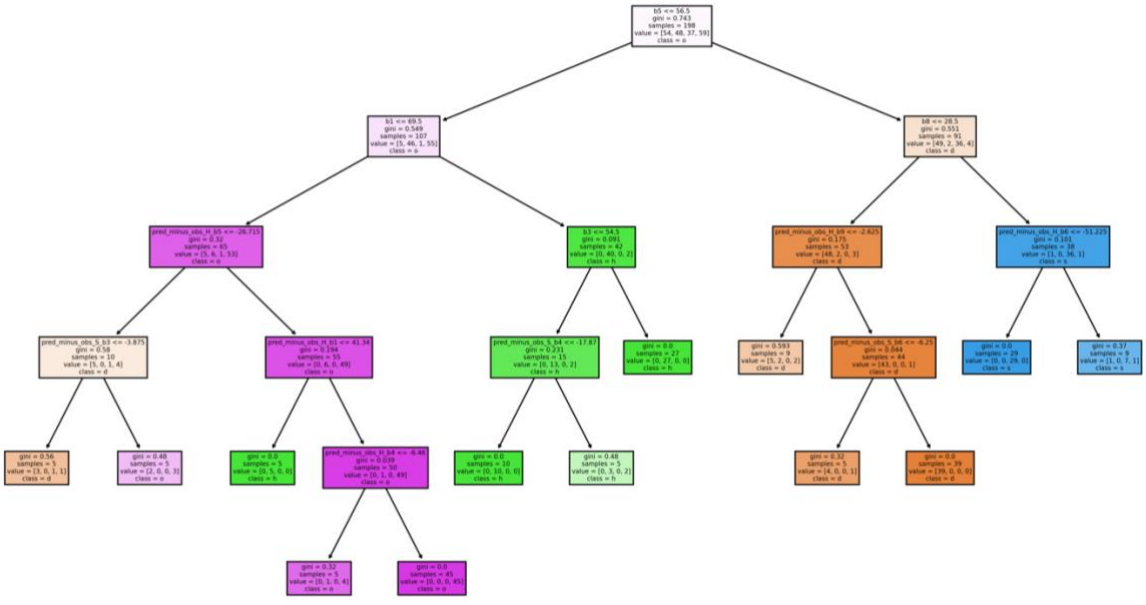
Confusion Matrix:



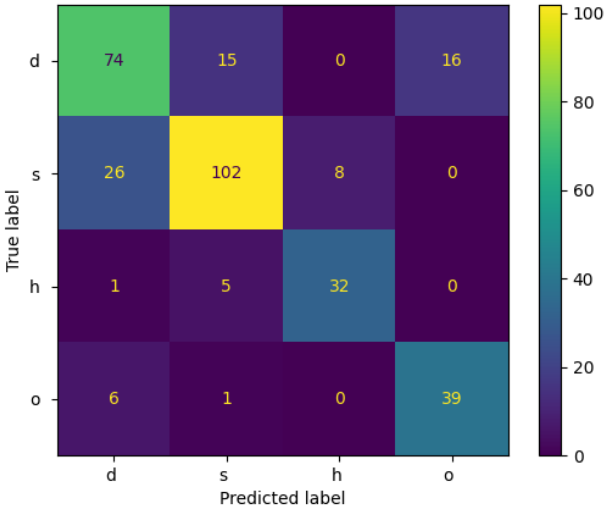
Analysis:
This subtree's classification performance, evaluated via confusion matrix, offers mixed results. While this subtree identifies a higher number of class 'd' nodes (83 as compared to 76 in the random forest), it also labels class 's' nodes less accurately (107 compared to 123). Moreover, this tree's 's-d' confusion pair count of 23 is considerably higher than the original matrix's count of 5. On the contrary, the confusion pair d-s' count of 9 is much lower than the count of 17 in the random forest model.

Component Decision Tree 6:

Accuracy: 77.0%
Decision Tree:



Confusion Matrix:



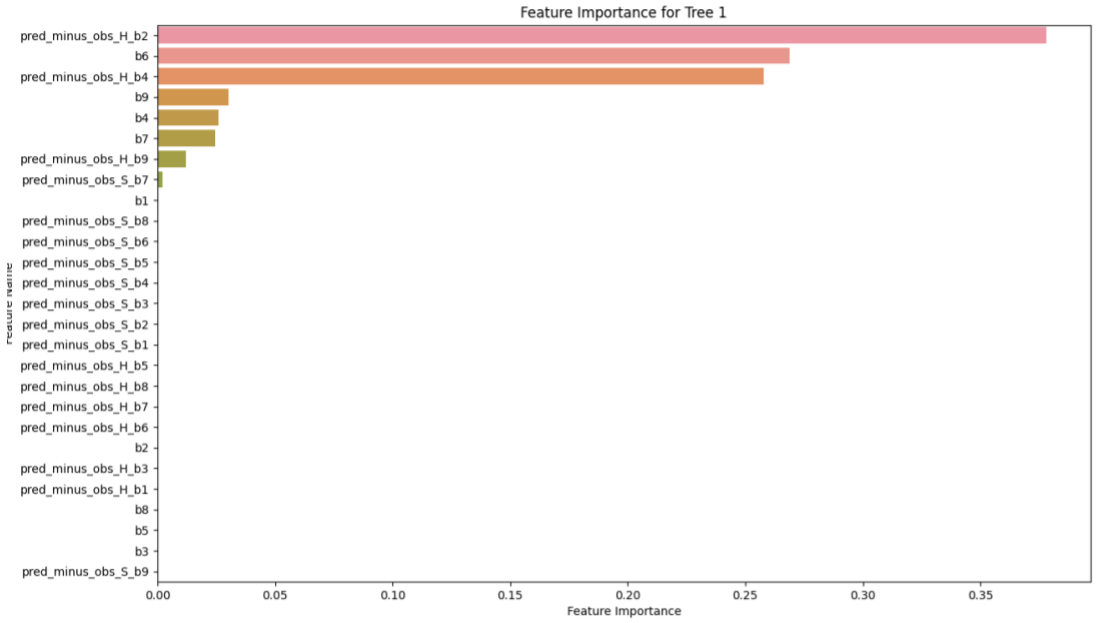
Analysis:

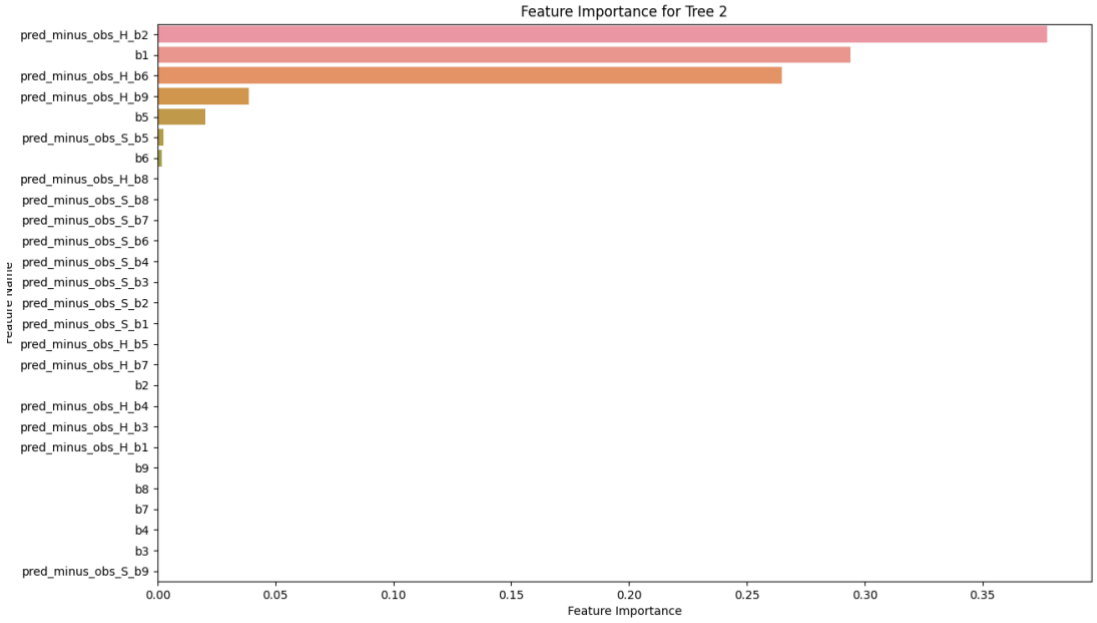
This component tree has the lowest accuracy of the set, and it is most likely due to a lower count in classifying class ‘s’ nodes accurately. The decision tree classifies 102 instances correctly, compared to 123 in the original random tree classifier. The resulting inaccuracies are largely due to the 26 s-d confusion pairs prevalent here.

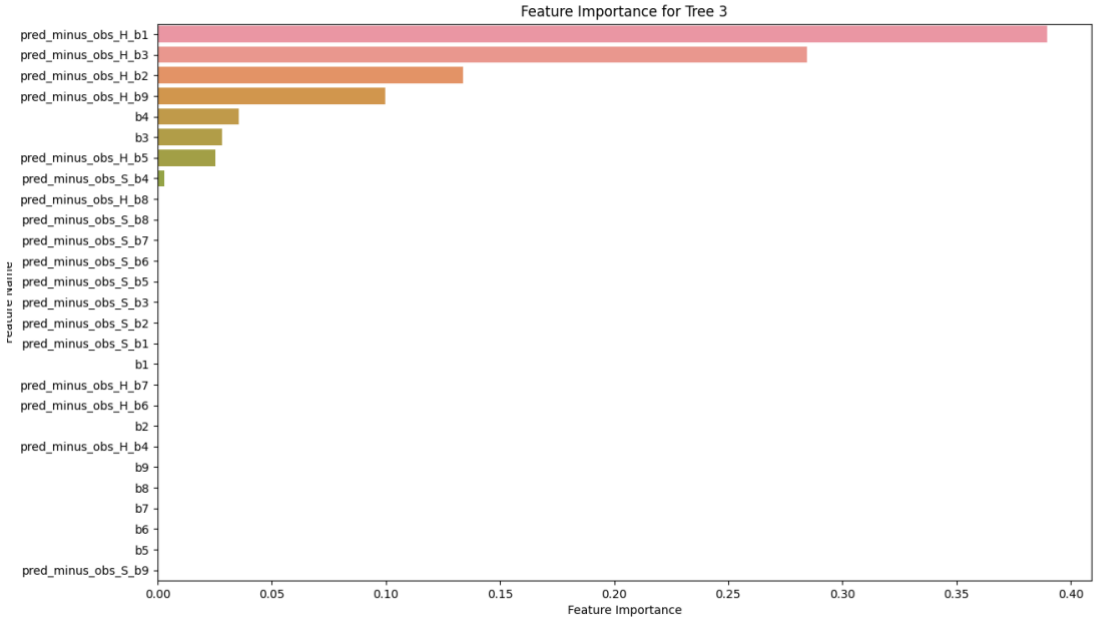
Part C

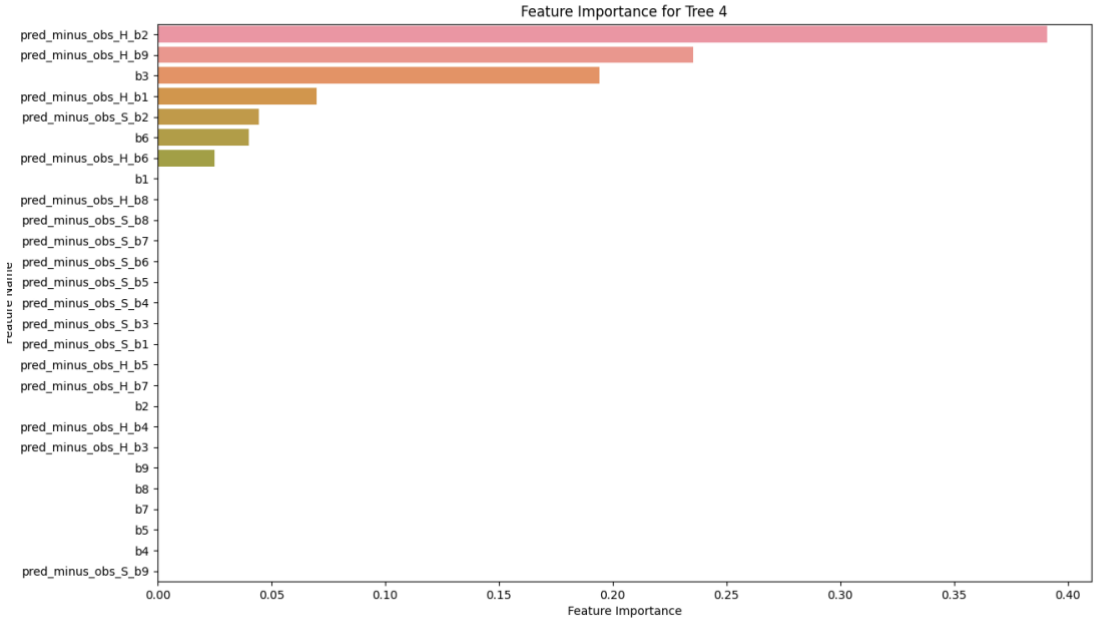
For a random forest classifier (or one of its component trees), the relative importance of the attributes can be measured through the `feature_importances_` *field of the classifier. For selected component trees in (b), compare their associated lists of relative attribute importance values. (25%)*

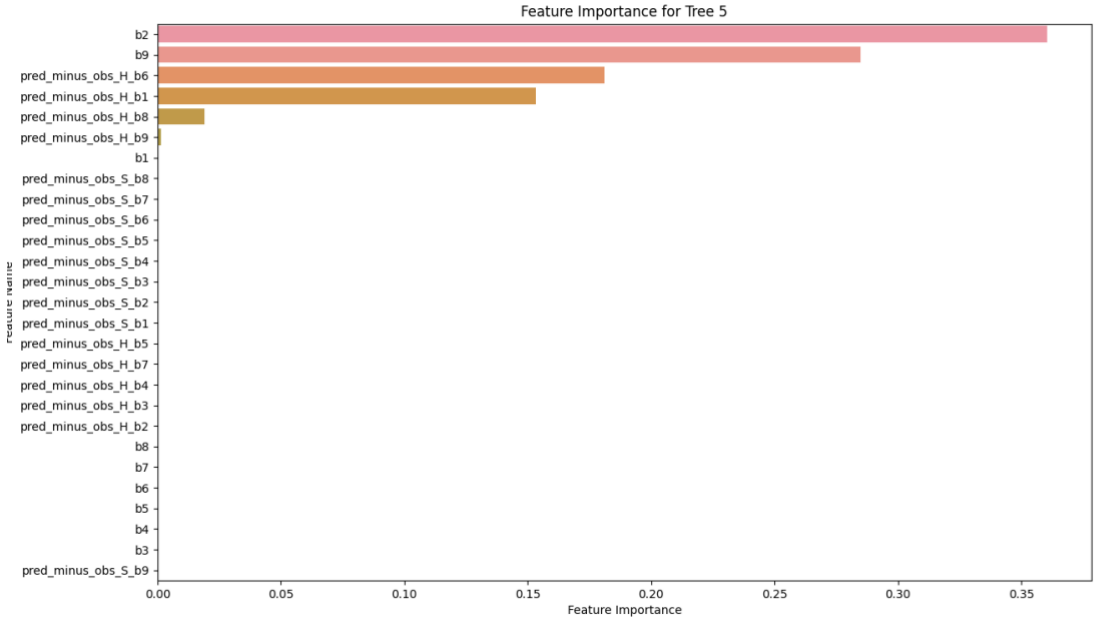
The feature importance vs feature name diagrams for all 6 component trees are shown below:

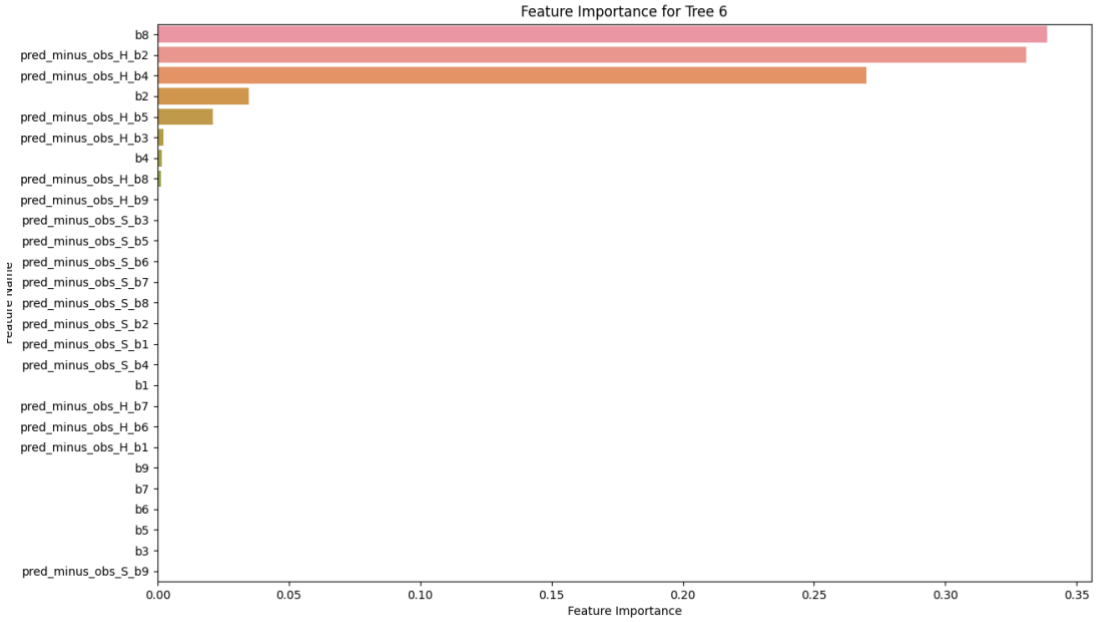




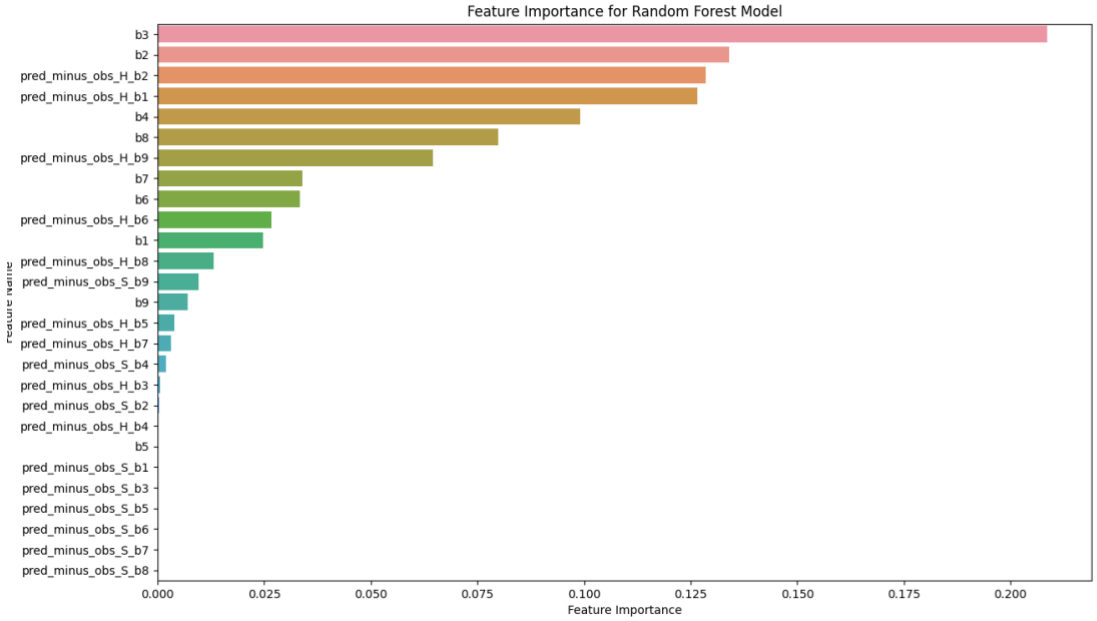

Feature Importance for Tree 3
pred_minus_obs_H_b1
pred_minus_obs_H_b3
pred_minus_obs_H_b2
pred_minus_obs_H_b9
b4
b3
pred_minus_obs_H_b5
pred_minus_obs_S_b4
pred_minus_obs_H_b8
pred_minus_obs_S_b8
pred_minus_obs_S_b7
pred_minus_obs_S_b6
pred_minus_obs_S_b5
pred_minus_obs_S_b3
pred_minus_obs_S_b2
pred_minus_obs_S_b1
b1
pred_minus_obs_H_b7
pred_minus_obs_H_b6
b2
pred_minus_obs_H_b4
b9
b8
b7
b6
b5
pred_minus_obs_S_b9
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
Feature Importance


Feature Importance for Tree 4
pred_minus_obs_H_b2
pred_minus_obs_H_b9
b3
pred_minus_obs_H_b1
pred_minus_obs_S_b2
b6
pred_minus_obs_H_b6
b1
pred_minus_obs_H_b8
pred_minus_obs_S_b8
pred_minus_obs_S_b7
pred_minus_obs_S_b6
pred_minus_obs_S_b5
pred_minus_obs_S_b4
pred_minus_obs_S_b3
pred_minus_obs_S_b1
pred_minus_obs_H_b5
pred_minus_obs_H_b7
b2
pred_minus_obs_H_b4
pred_minus_obs_H_b3
b9
b8
b7
b5
b4
pred_minus_obs_S_b9
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
Feature Importance





The same diagram for the main random forest model is illustrated below:

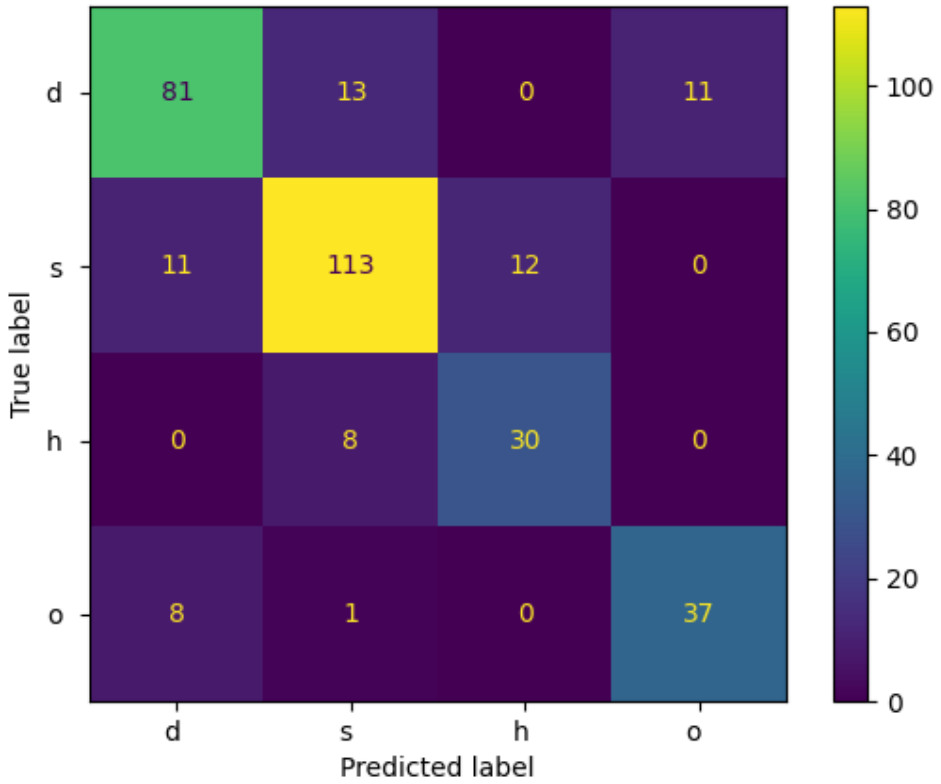


Interestingly, the diagram for the random forest classifier is quite different to either of the component trees' diagrams. First of all, the degree of feature importance for the most important feature for the random forest classifier barely crosses the 0.2 mark. On the contrary, the most important features in all 6-component tree's scale a much larger value: approximately in the range of 0.35 to 0.40, almost twice the value. Moreover, component trees 1,2, and 6 have 3 features that cross the threshold feature importance of 0.05 while component trees 3,4, and 5 have 4 features that cross the same threshold. Nonetheless, the random forest model is not as overly dependent on a scare list of features given that there are 7 features that cross the established threshold.

Part D

Construct a naïve Bayes classifier model based on our data set, and compare the classification performance with that of the random forest model. (25%)

To generate this model in particular, the sklearn module's "GaussianNB()" Python function was utilized. The function call was made entirely without passing in any additional parameters, i.e., the classifier was constructed on the default settings. The confusion matrix for the naïve Bayes classifier model is depicted below:



Compared to the Random Forest classifier's matrix, the only noticeable difference is in the classification of node 's'. The above Bayes classifier predicts 113 instances correctly whereas the Random Forest classifier predicts 121 instances correctly. The discrepancy can be explained by the mislabeled for class 's' as 'd' and 'h' (with 11 and 12 instances respectively).