CS 3481

Fundamentals of Data Science

Assignment-1: Decision Tree

Table of Contents

Name: Avi Malhotra
Student ID: 55773896

Part A

Construct multiple decision trees based on the default training set/test set partition using different parameter settings. Compare the structures and classification performances of these decision trees. (25%)

Decision Trees 1A, 2A, 3A and 4A (please refer to the Appendix) are based on the default training sets (training.csv).

3 parameter settings were chosen for these trees:

- Criterion – defines the impurity using the Gini or Entropy values.
- Splitter – defines the strategy used to split each node.
- Max Depth – defines the maximum depth of the tree.

The following table surmises the different parameter settings employed for these trees:

Trees based on the training set			
Tree	**Criterion**	**Splitter**	**Max Depth**
1A	Gini	Best	3
2A	Entropy	Random	3
3A	Gini	Random	5
4A	Entropy	Best	5

Since there are 27 features in the entire dataset, constraining the maximum number of features in the decision tree classifier could result in underfitting. Thus, the features parameter was not controlled. A decision was also made to restrict the maximum depth to a smaller value to ensure large trees do not form as doing so would have resulted in misclassifications that would be much harder to identify by the naked eye.

The prediction accuracy for the decision trees is as follows:

- Tree 1A: 78%
- Tree 2A: 66%
- Tree 3A: 75%
- Tree 4A: 76%

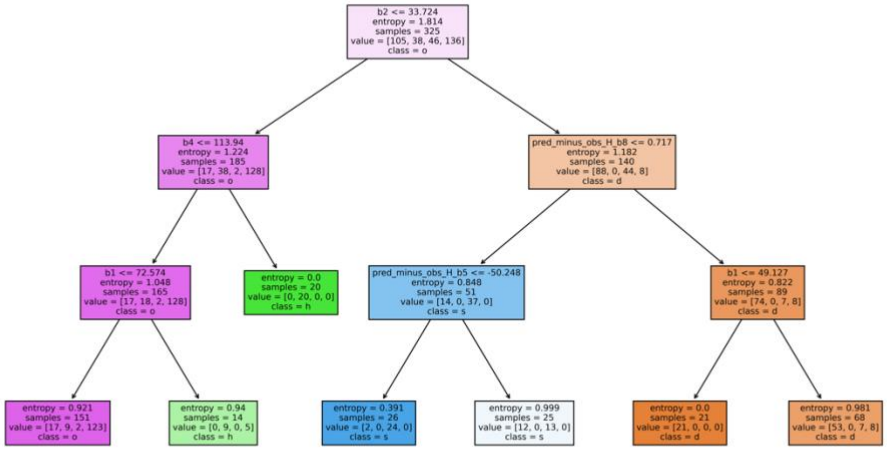
It can be seen that Tree 2A, with an accuracy of only 66%, is a clear outlier. Changing the criterion from Gini to Entropy in Trees 1A and 4A respectively does not yield a noticeable change in accuracy, and similarly, increasing the maximum depth does not affect the accuracy as well.

The decision trees 1A and 2A are almost perfectly balanced, i.e., the depth of most leaf nodes is equal to the constraint specificized by the max depth parameter. This pattern is not prevalent in the case of Tree 4A, wherein the max depth is 4 (1 below the constraint of 5). In fact, most leaf nodes have a depth of 3. The structural formation of these trees suggests overfitting for the

decision tree 3A. Even though it has a reasonable accuracy of 75%, this tree would not be as effective for larger datasets.

Tree 2A's astonishingly low accuracy can be explained by the splitter parameter, which is set to random. In such a filter, the strategy to choose a split at the node is dependent on the "best random split". Hence, Tree 3A's accuracy that appears to be relatively consistent with 1A's and 4A's is also a matter of chance. This can be confirmed by remaking the decision trees for 2A and 3A, which may result in decision trees of vastly different accuracies.

For the sake of curiosity, I firsthand remade Tree 3A for multiple trial runs, one of which even yielded an accuracy of a whopping 91%. The decision tree for this trial run is shown below:



Part B

Exchange the training and test set and repeat the tasks in (a). (25%)

Decision Trees 1B, 2B, 3B and 4B (please refer to the Appendix) are based on the testing sets (testing.csv). It is worth mentioning that all 3 parameter settings for these decision trees were unaltered; only the data sets were interchanged:

Trees based on the testing set			
Tree	**Criterion**	**Splitter**	**Max Depth**
1A	Gini	Best	3
2A	Entropy	Random	3
3A	Gini	Random	5
4A	Entropy	Best	5

The prediction accuracy for the decision trees is as follows:

- Tree 1A: 88%
- Tree 2A: 86%
- Tree 3A: 92%
- Tree 4A: 85%

Generally speaking, the accuracy of the decision trees based on the testing data is much higher than those in part A that are based on the training data. Such a difference is primarily due to the varying sizes of the datasets: the testing dataset is much larger than the training one. As a result, switching the datasets yields decision trees that are not only more complex, but more accurate as well.

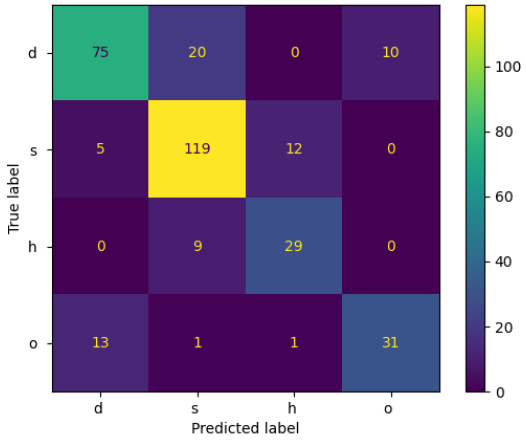
When comparing the decision trees, it can be seen that Trees 1B and 2B, and Trees 3B and 4B are similar. On the contrary, these two sets are quite different than each other. The aforementioned sets are not only different in terms of the features/filters used – namely the max depth of 3 vis-à-vis 5 – but also in terms of their structure. Trees 1B and 2B are fully filled as all leaf nodes in both the trees extend to the maximum permissible tree depth. In comparison with Trees 3B and 4B, the model depicted in Trees 1B and 2B is underfitting. In these cases, one may construct better decision trees by increasing the maximum depth, though an observable difference in accuracy in my examples is only observed in the case of Tree 3A, where a Gini criterion is employed.

Part C

For selected trees in (a) and (b), observe the classification performance associated with the different classes, and determine which pair(s) of classes are likely to be confused with each other. (25%)

Note that confusion pairs are determined using the confusion matrices below wherein pairs are selected based on whether their count has been greater than or equal to 10.

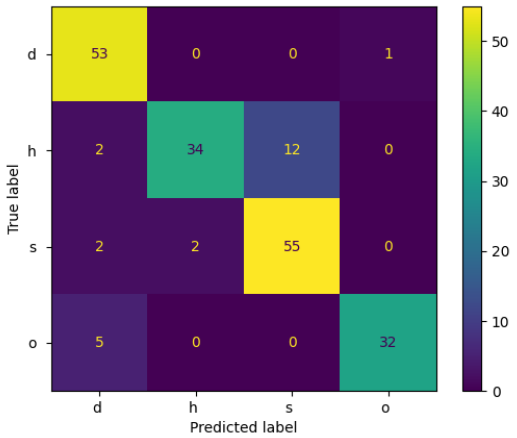
Tree 1A's Confusion Matrix:



Based on the above matrix, the following confusion pairs may arise:

- class d mistaken for class s
- class s mistaken for class h
- class o mistaken for class d
- class d mistaken for class o

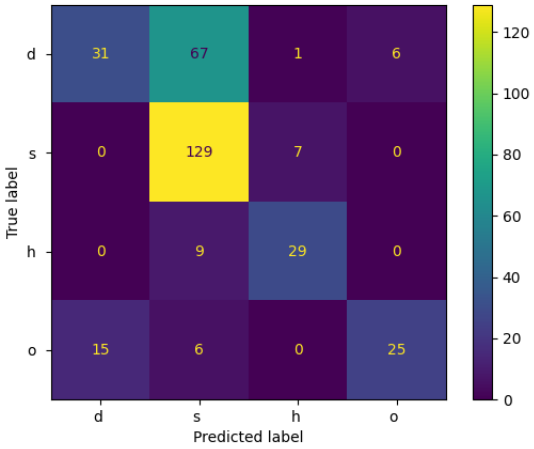
Tree 1B's Confusion Matrix:



Based on the above matrix, the following confusion pairs may arise:

- class h mistaken for class o

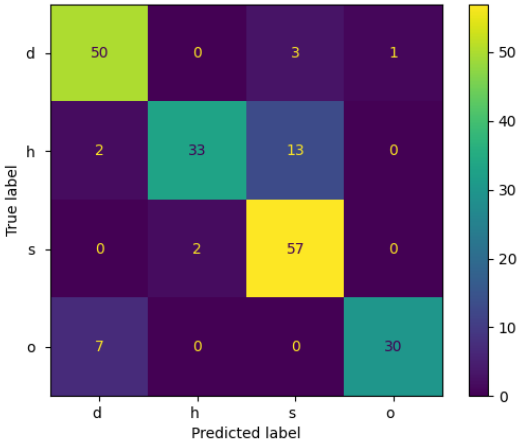
Tree 2A's Confusion Matrix:



Based on the above matrix, the following confusion pairs may arise:

- class d mistaken for class s
- class o mistaken for class d

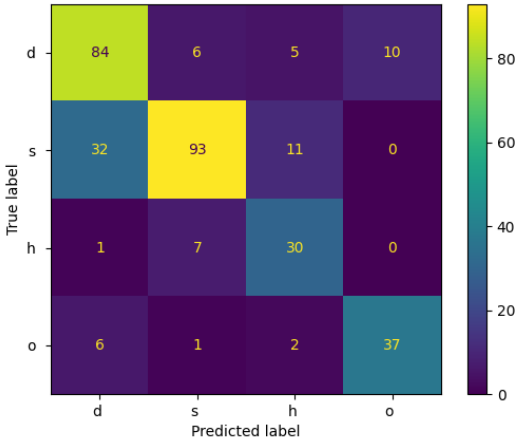
Tree 2B's Confusion Matrix:



Based on the above matrix, the following confusion pairs may arise:

- class h mistaken for class s

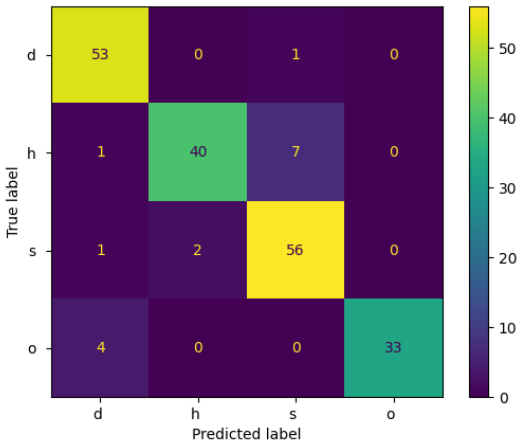
Tree 3A's Confusion Matrix:



Based on the above matrix, the following confusion pairs may arise:

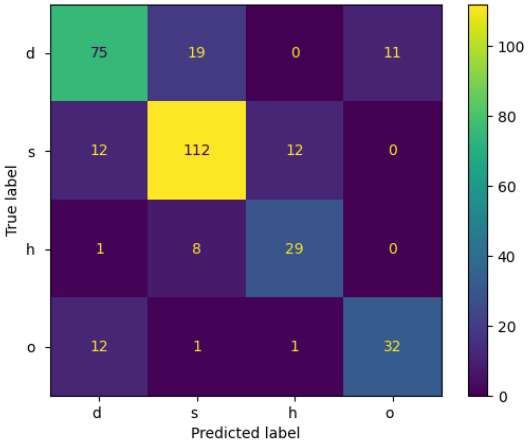
- class s mistaken for class d
- class s mistaken for class h
- class d mistaken for class o

Tree 3B's Confusion Matrix:



The maximum count for a confused pair (class h mistaken for class s) is 7, which is not that significant.

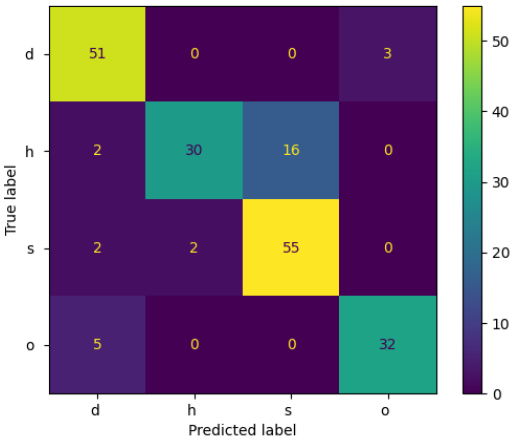
Tree 4A's Confusion Matrix:



Based on the above matrix, the following confusion pairs may arise:

- class d mistaken for class s
- class s mistaken for class d
- class s mistaken for class h
- class o mistaken for class d
- class d mistaken for class o

Tree 4B's Confusion Matrix:



Based on the above matrix, the following confusion pairs may arise:

- class h mistaken for s

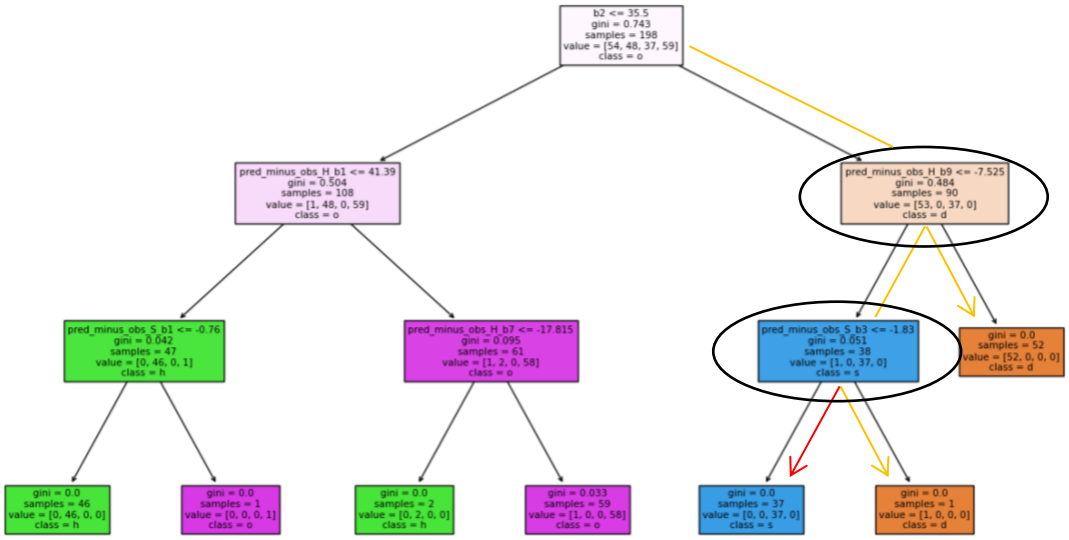
Part D

For selected confused class pairs in (c), identify the corresponding leaf node(s) and analyze the sequence of decisions that lead to the misclassification. (25%)

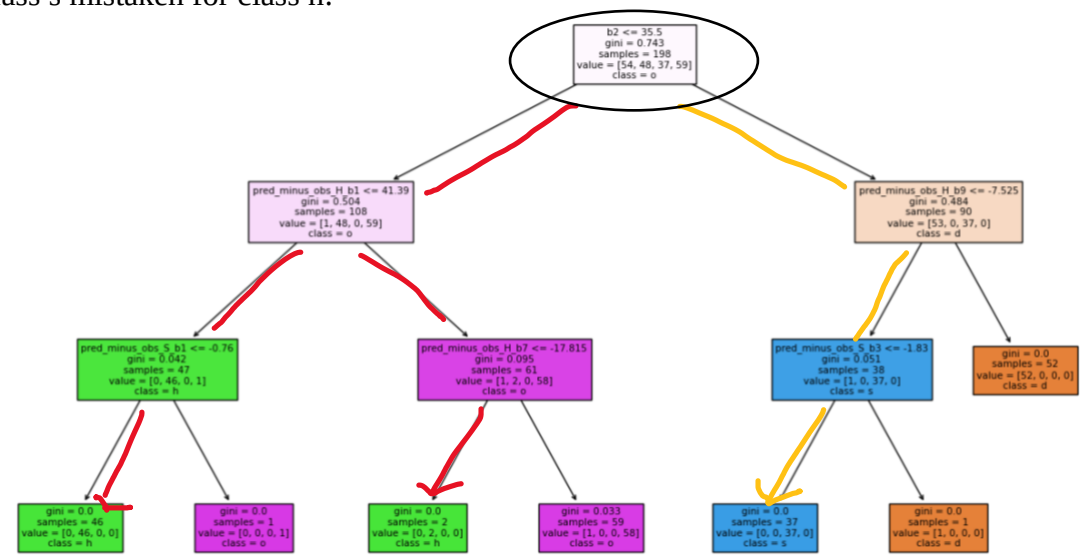
Note that the golden arrows denote the sequence of decisions that should be taken for correct classification whereas the red arrows denote the sequence of decisions that resulted in the misclassification. The node(s) at which the misclassification occurs have been circled.

Tree 1A:

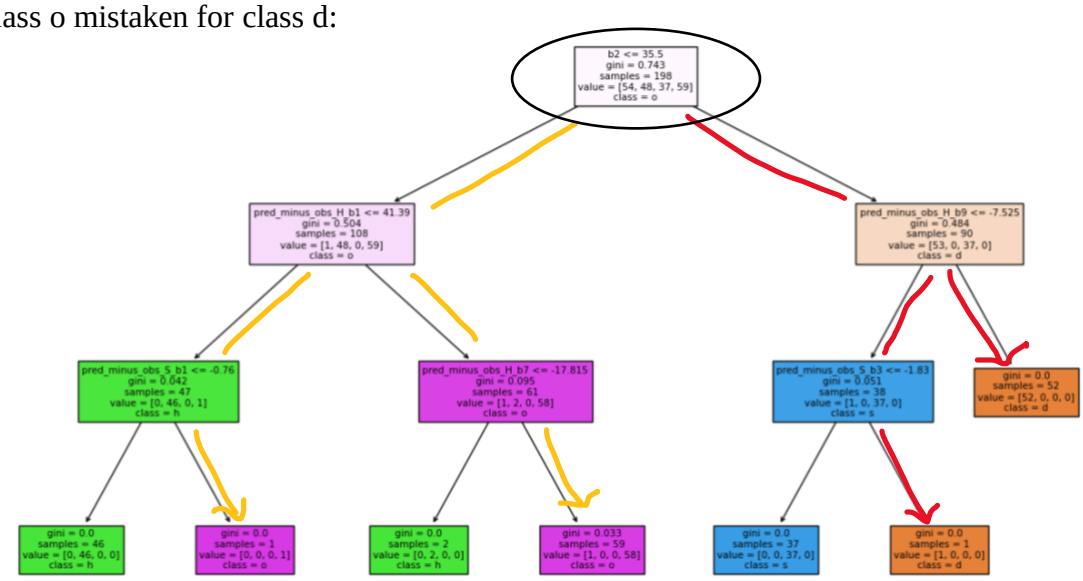
For class d mistaken as class s:



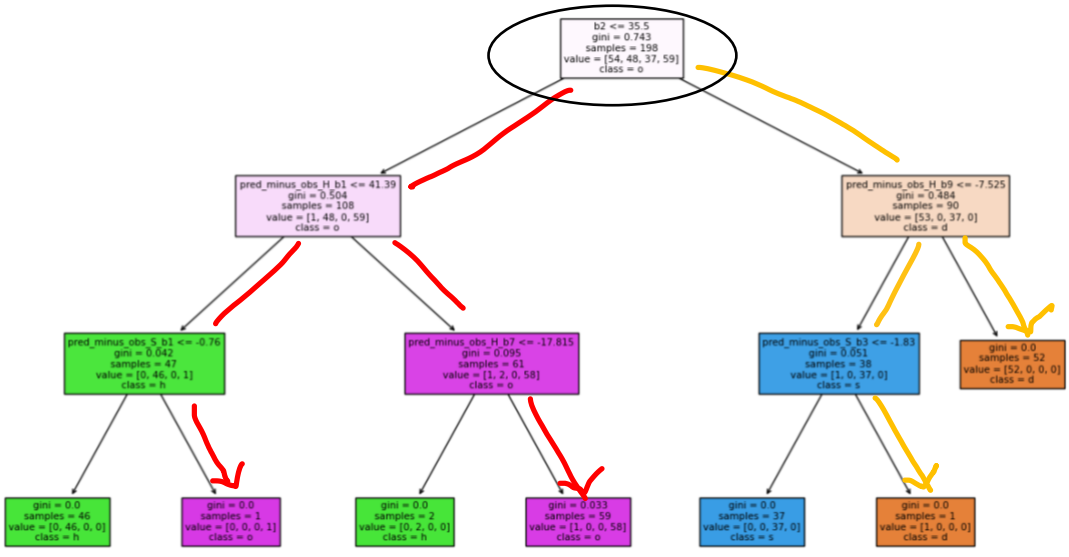
For class s mistaken for class h:



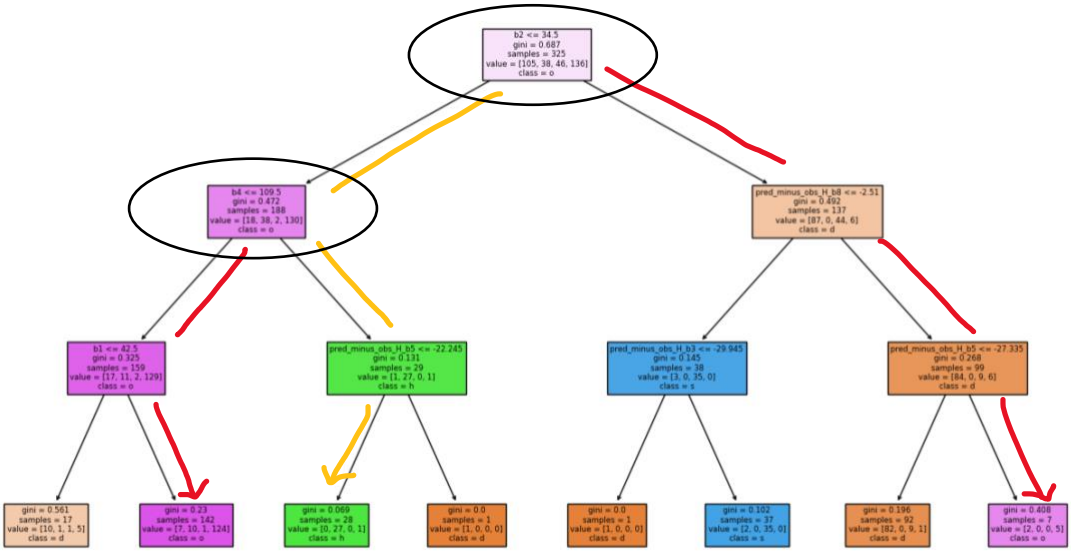
For class o mistaken for class d:



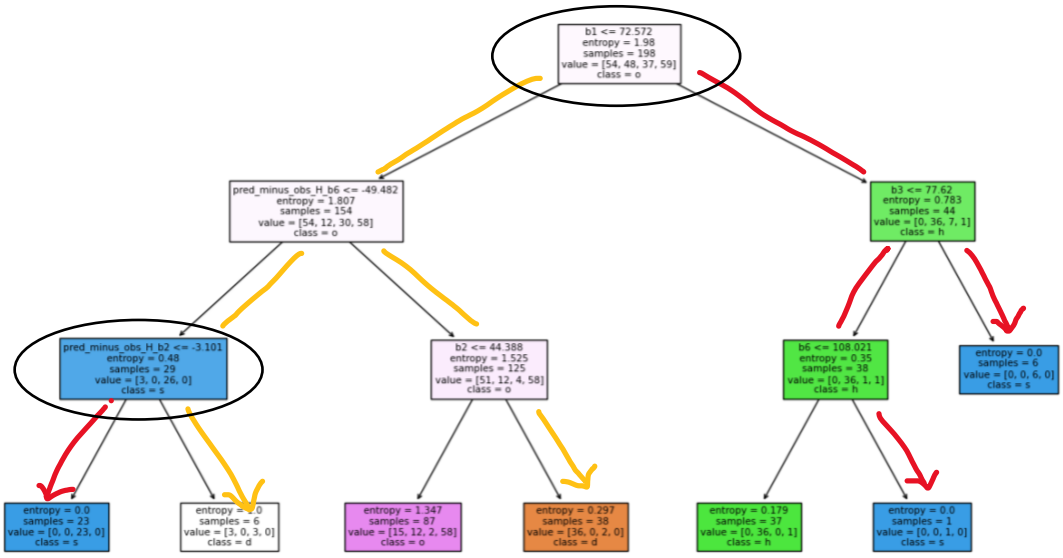
For class d mistaken for class o:



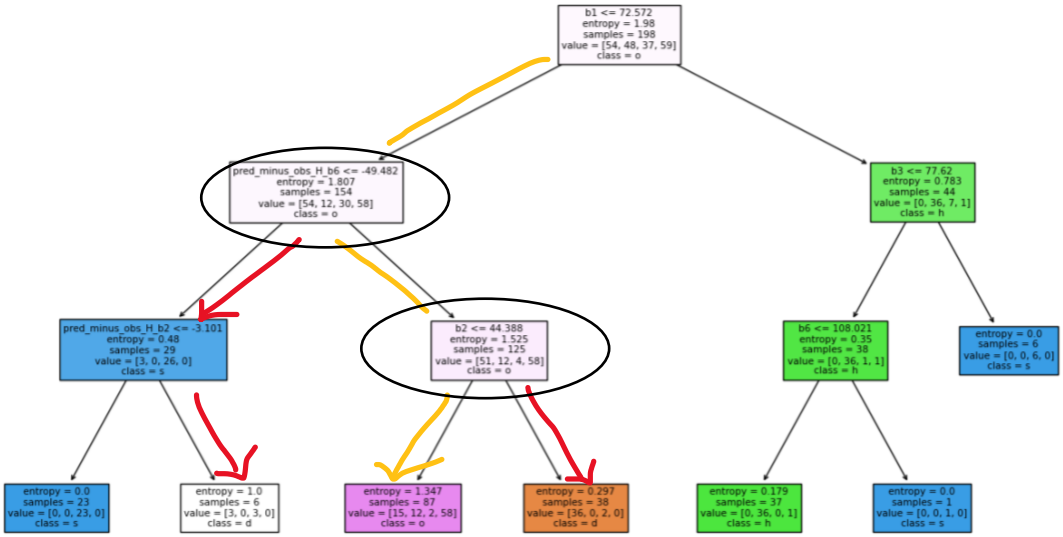
Tree 1B:
class h mistaken for class o



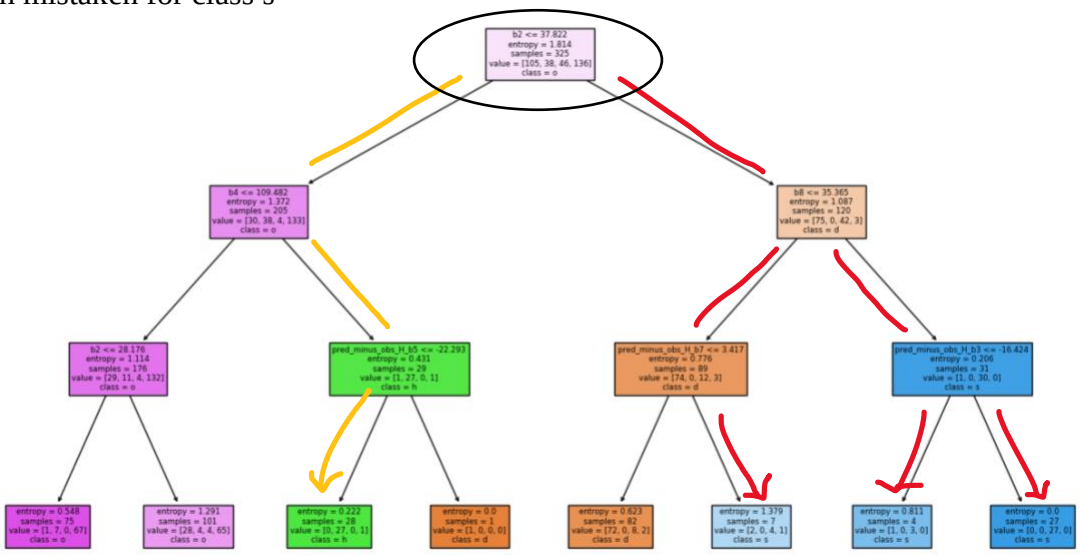
Tree 2A:
class d mistaken for class s



class o mistaken for class d

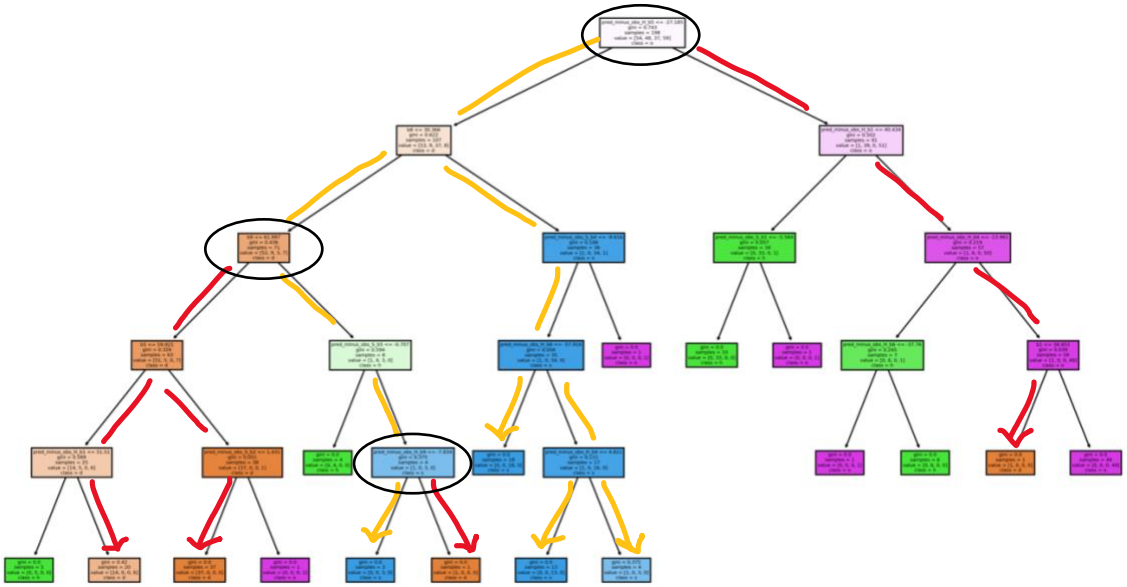


Tree 2B:
class h mistaken for class s

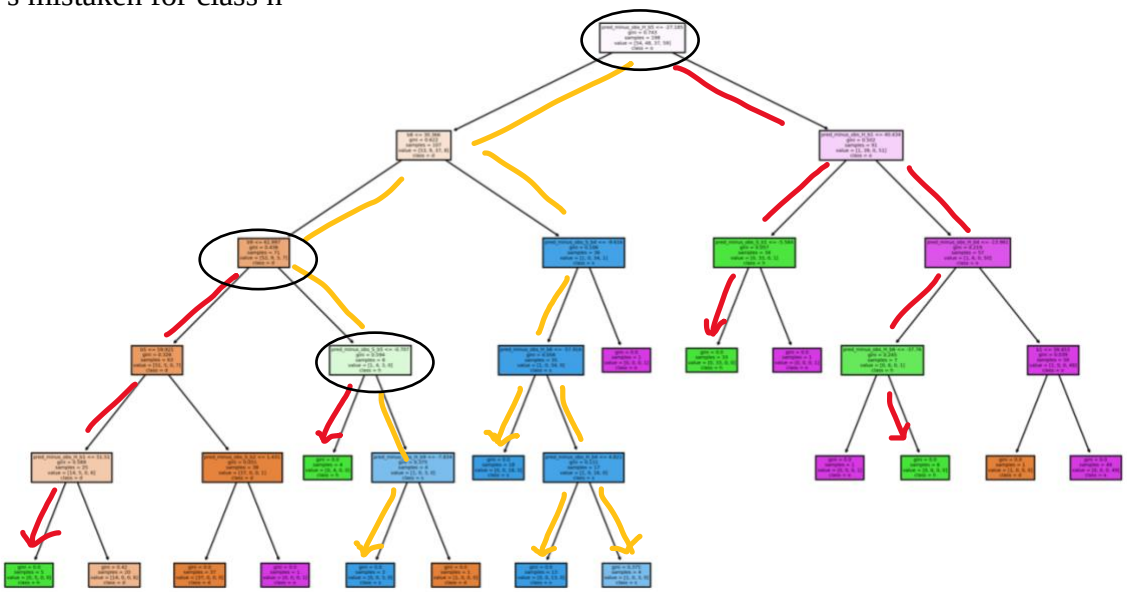


Tree 3A:

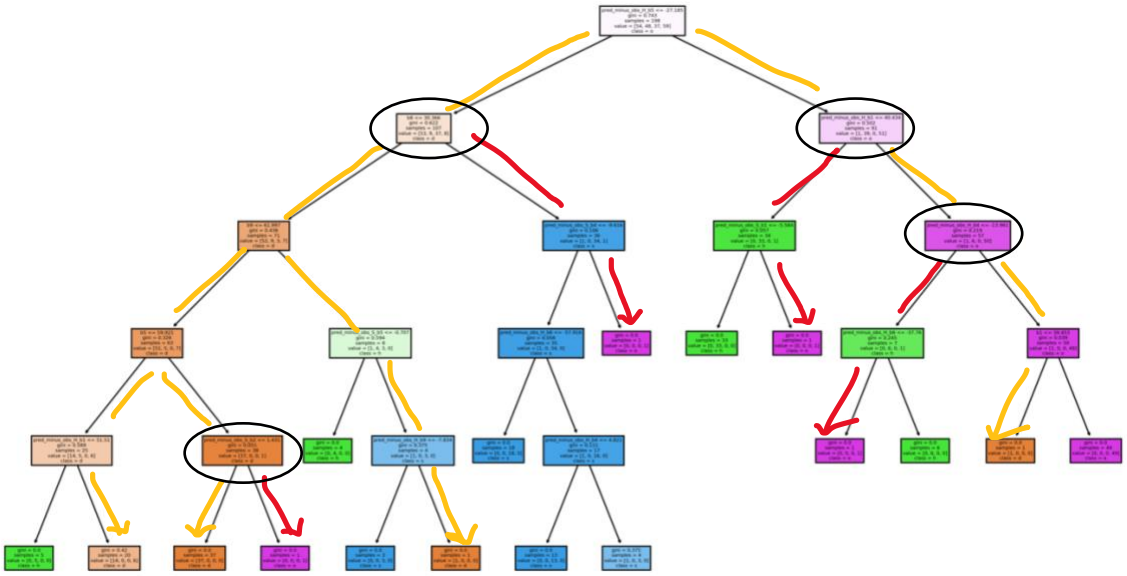
class s mistaken for class d



class s mistaken for class h

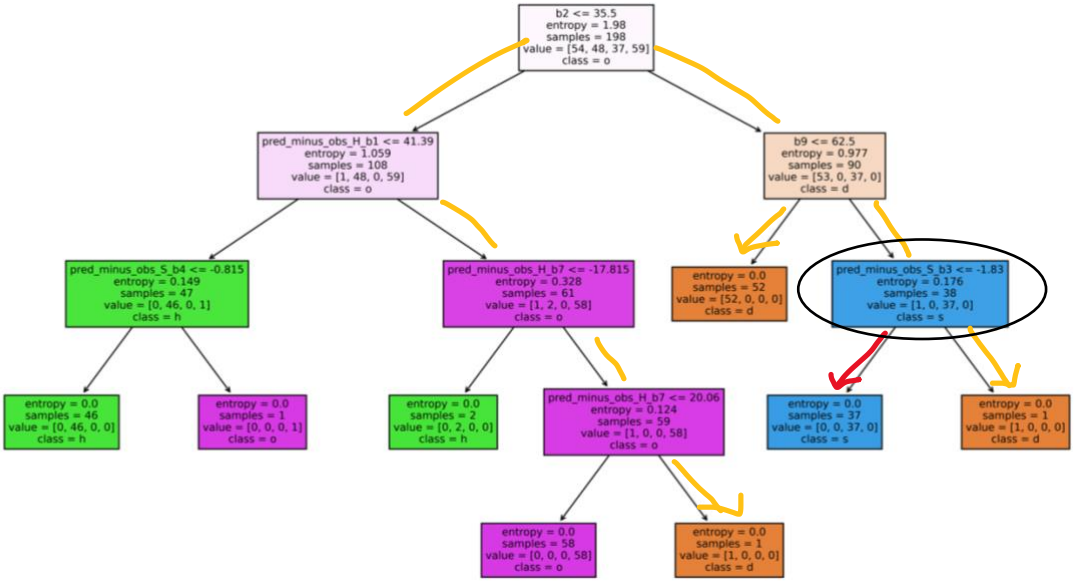


class d mistaken for class o

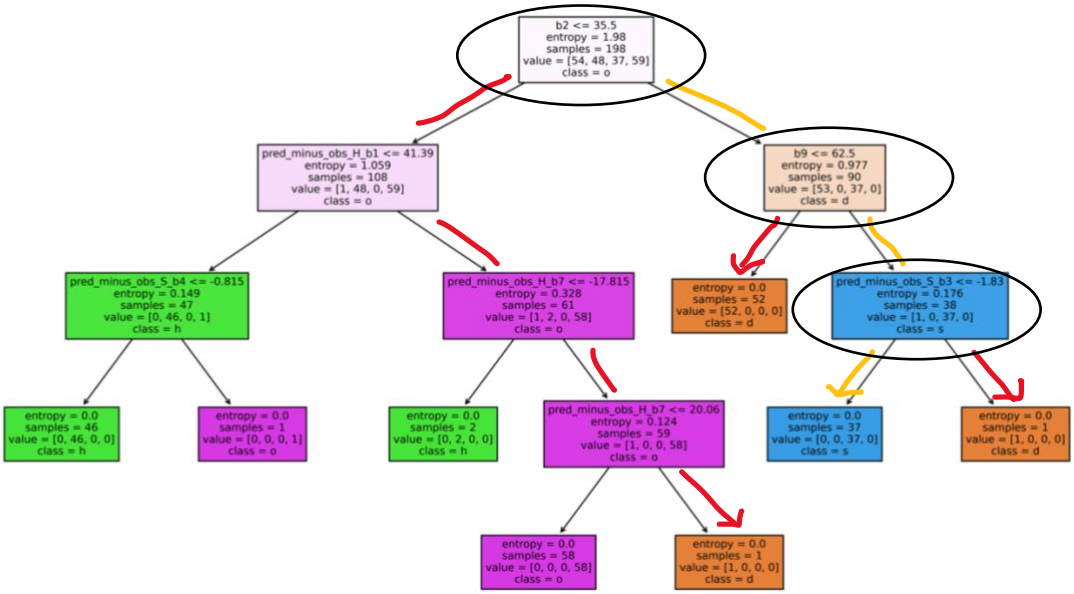


Tree 3B:
N/A (See Part C for more info)

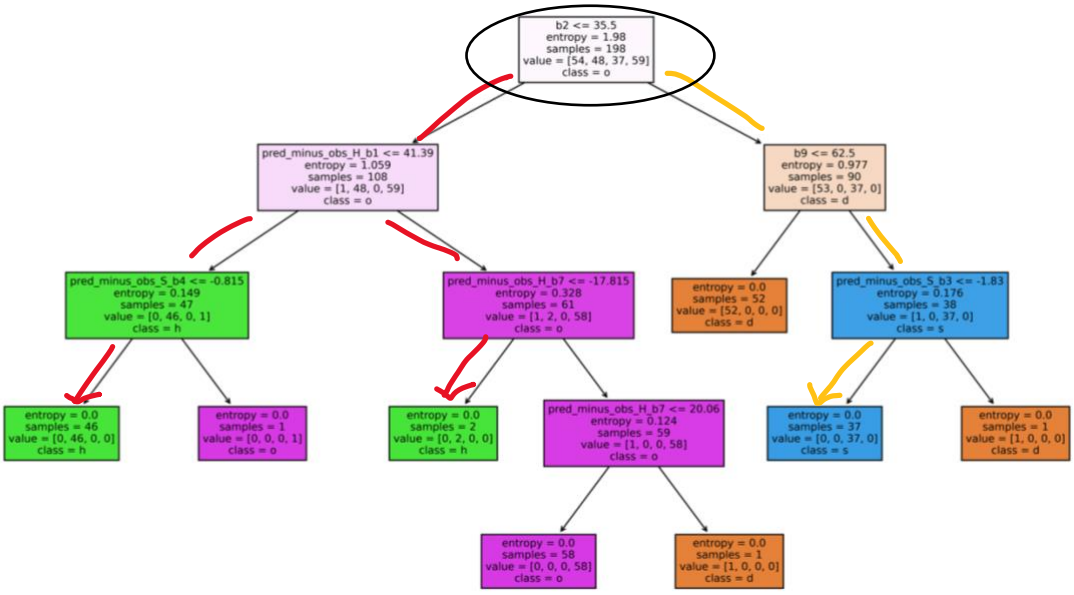
Tree 4A:
class d mistaken for class s



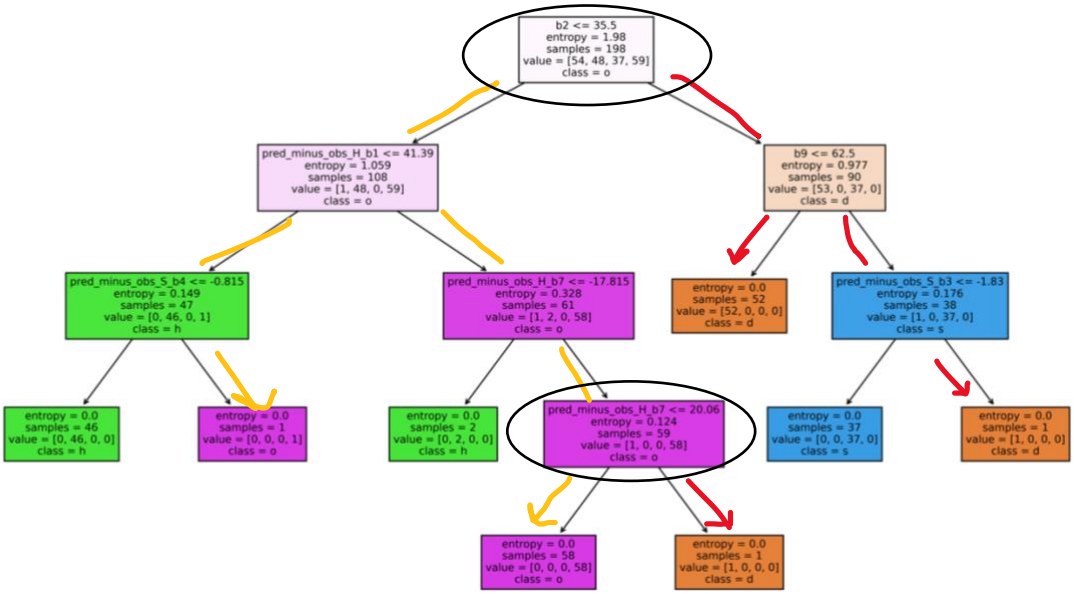
class s mistaken for class d



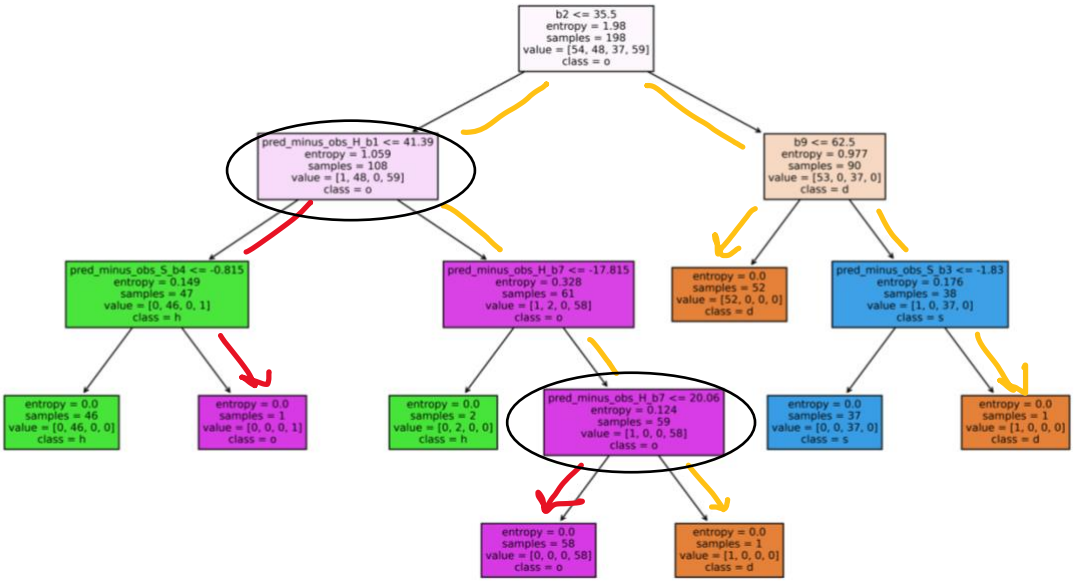
class s mistaken for class h



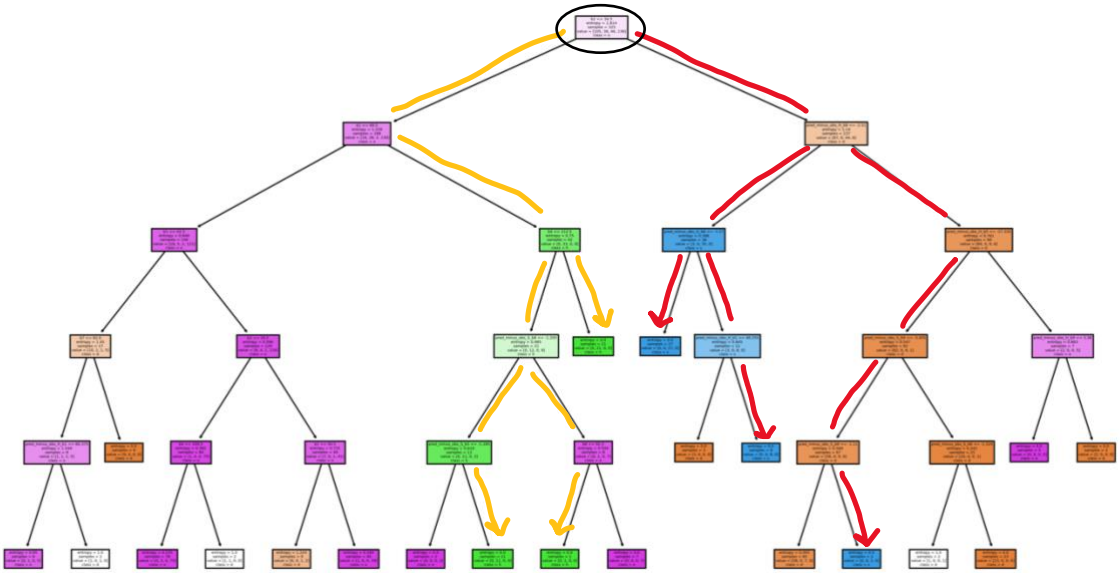
class o mistaken for class d



class d mistaken for class o



Tree 4B:
class h mistaken for s



Appendix: Tree 1A

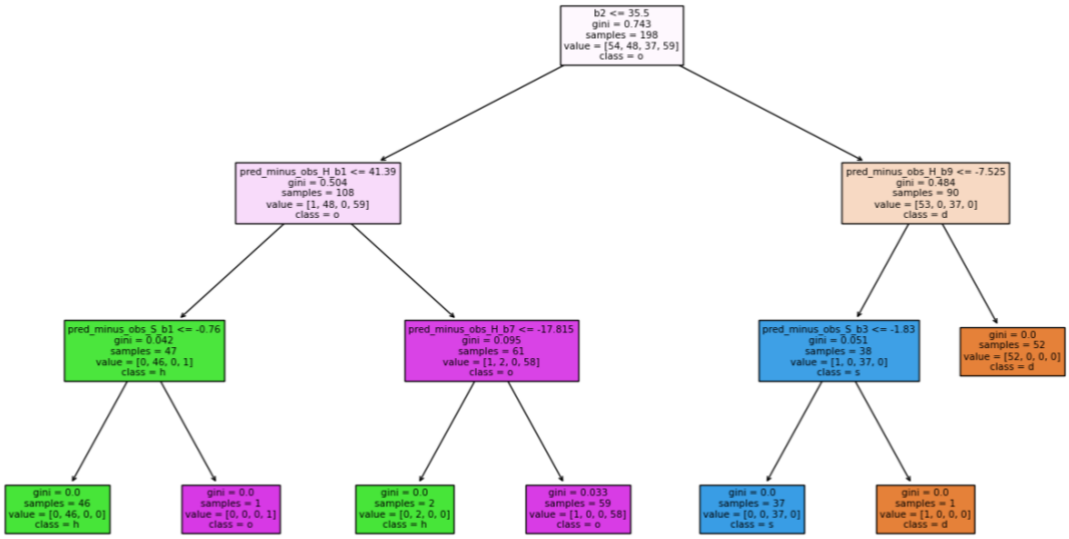
Accuracy: 78%

Prediction on: Testing data

Filters:

- Criterion: Gini
- Splitter: Best
- Max Depth: 3

Decision Tree:



Appendix: Tree 1B

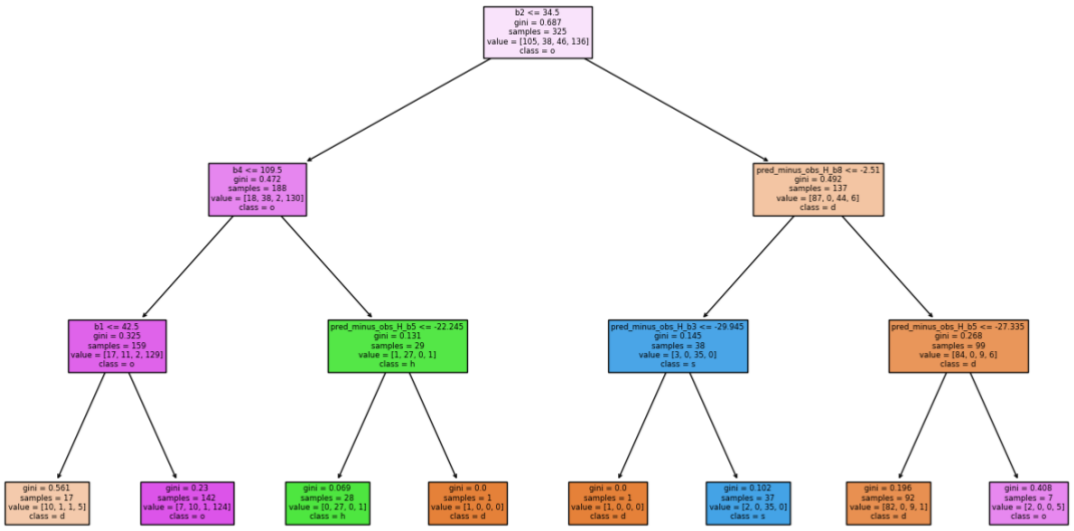
Accuracy: 88%

Prediction on: Training data

Filters:

- Criterion: Gini
- Splitter: Best
- Max Depth: 3

Decision Tree:



Appendix: Tree 2A

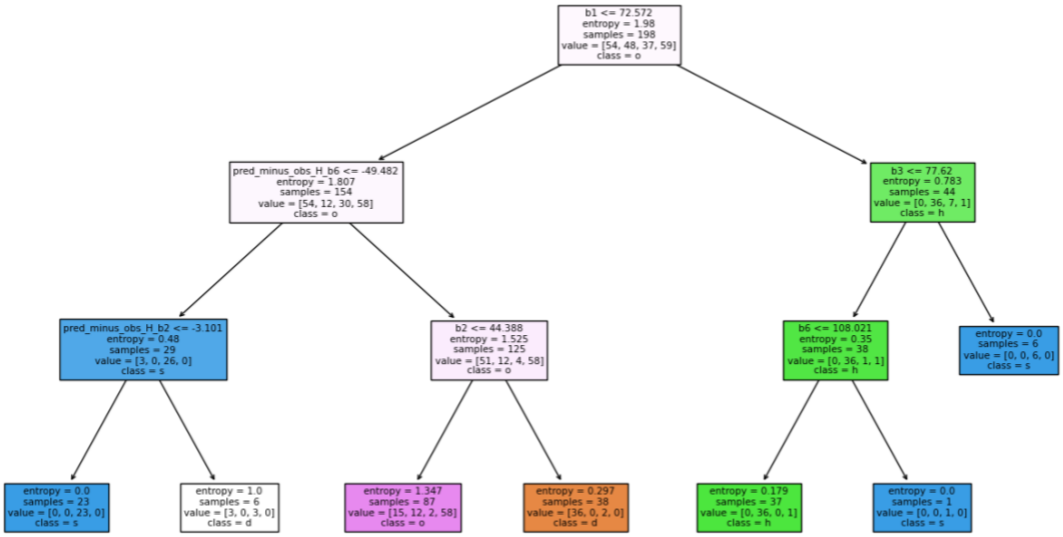
Accuracy: 66%

Prediction on: Testing data

Filters:

- Criterion: Entropy
- Splitter: Random
- Max Depth: 3

Decision Tree:



Appendix: Tree 2B

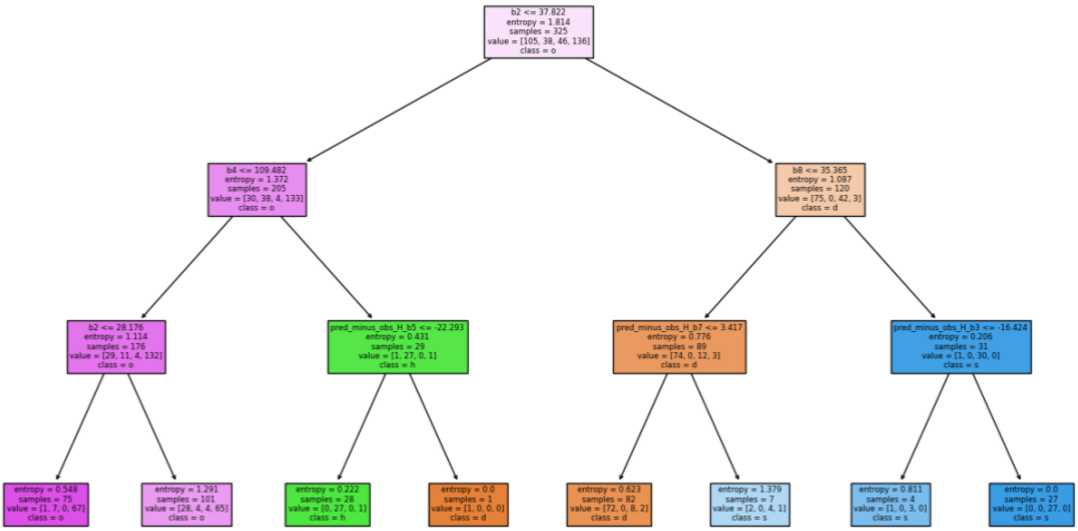
Accuracy: 86%

Prediction on: Training data

Filters:

- Criterion: Entropy
- Splitter: Random
- Max Depth: 3

Decision Tree:



Appendix: Tree 3A

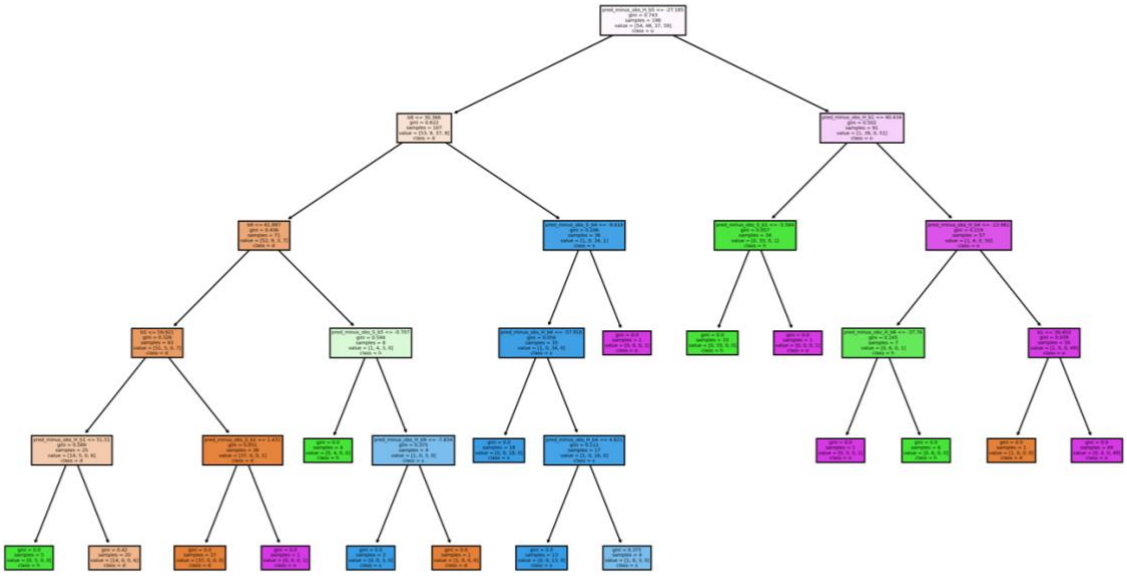
Accuracy: 75%

Prediction on: Training data

Filters:

- Criterion: Gini
- Splitter: Random
- Max Depth: 5

Decision Tree:



Appendix: Tree 3B

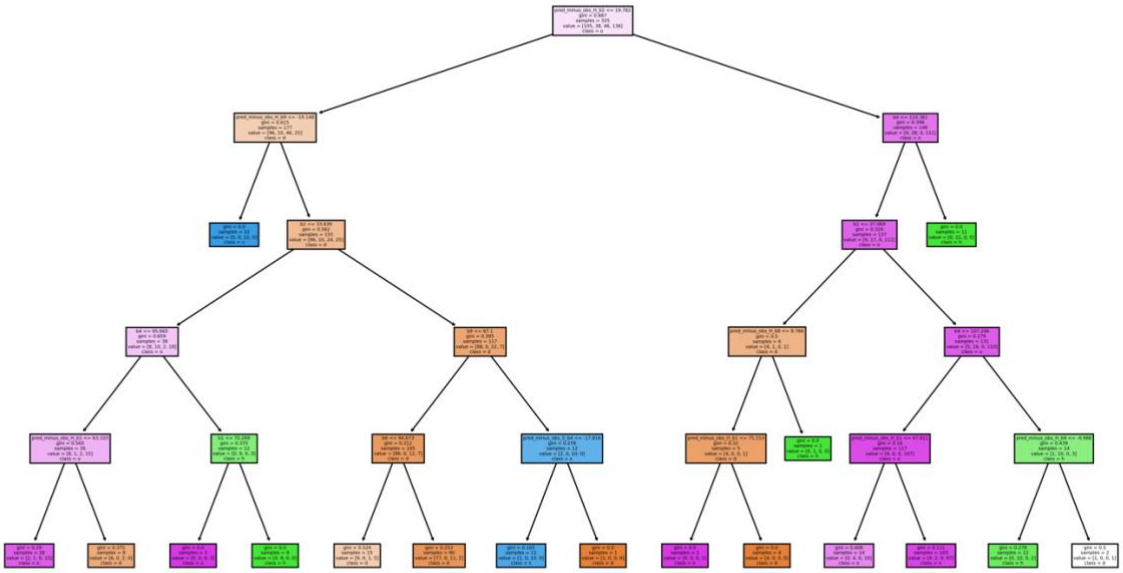
Accuracy: 92%

Prediction on: Testing data

Filters:

- Criterion: Gini
- Splitter: Random
- Max Depth: 5

Decision Tree:



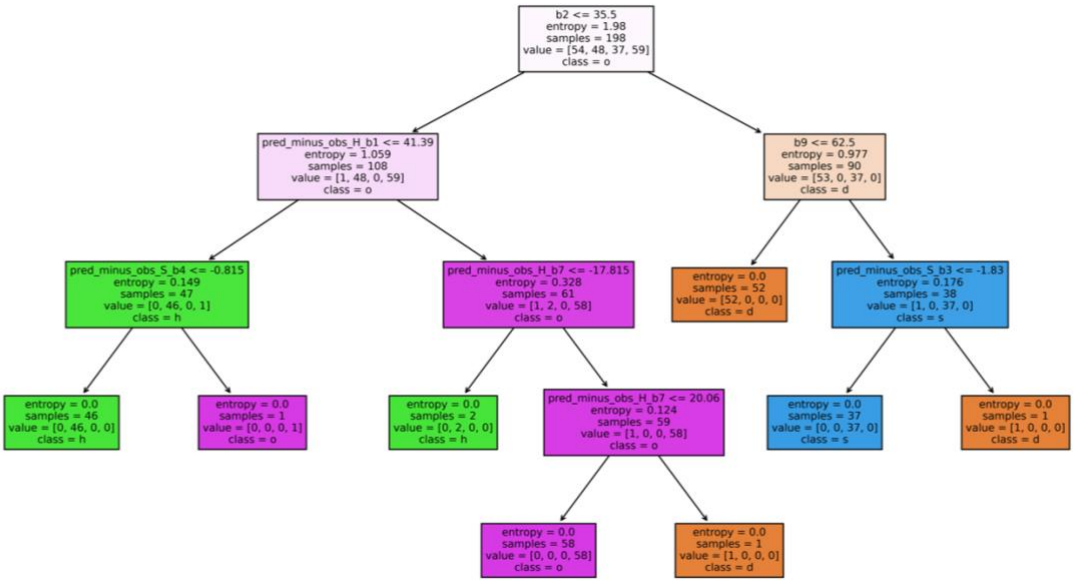
Appendix: Tree 4A

Accuracy: 76%

Prediction on: Testing data

Filters:

- Criterion: Entropy
- Splitter: Best
- Max Depth: 5



Appendix: Tree 4B

Accuracy: 85%

Prediction on: Training data

Filters:

- Criterion: Entropy
- Splitter: Best
- Max Depth: 5

Decision Tree: