SENG 474 Assignment 1

Performance Analysis of Classification Methods

Ryan Russell

June 25, 2020

1. Second Classification Problem

The second classification problem selected for this assignment involves the evaluation of an authentication procedure for banknotes: the "Banknote Authentication Data Set" [2]. This data was donated to the University of California Irvine's machine learning dataset store by researchers at the University of Applied Sciences, Ostwestfalen-Lippe. The data was extracted from photos (with a resolution of 660 dpi) of both counterfeit and genuine banknotes using a Wavelet transform tool. Each of the 1372 instances has 5 attributes. These 5 attributes are the following:

1. Variance of Wavelet Transformed image (continuous)
2. Skewness of Wavelet Transformed image (continuous)
3. Curtosis of Wavelet Transformed image (continuous)
4. Entropy of image (continuous)
5. Class (integer)

It should be noted that the fifth attribute, Class, is stored as an integer, but can only have two values: 0 and 1. This makes the dataset ideal for a binary classification task without any alterations to the data. Additionally, the data involved in this problem has practical interest to governments and other organizations interested in detecting counterfeit banknotes. Specifically, the results of the decision tree method may be more easily interpreted by non-technical professionals, and therefore is potentially more useful (in the case that overfitting does not occur).

This classification problem satisfies the requirement of a "reasonably small test error" as the test error is consistently well below 50%. This will be shown in the following sections. Furthermore, while some specific parameter combinations can achieve close to 0% test error, most parameter combinations for each method do not. Therefore, the problem is not so simple that 0% test error can always be achieved. Moreover, the relatively large sample size of 1372 instances (compared to the Cleveland dataset's 303 instances) allows for comparison of the three methods with a greater amount of data.

2. Decision Tree Performance Analysis

The first classification method that was implemented and analyzed was the decision tree. The cost complexity pruning rule was used for the decision tree implementation. As part of the analysis, three parameters were varied: the test split (percentage of data used for test rather than training), the split criterion, and the number of features for the tree. For both the Banknote and Cleveland datasets, the results of using test splits of 0.1, 0.2, 0.3, and 0.4 were recorded. As for the split criterion, results were recorded using Gini and Entropy. Finally, the number of features was varied from 1 to 4 for the Banknote data and 1 to 13 for the Cleveland data. The test accuracy plotted against the number of features for the decision tree on the Cleveland data is displayed in Figures 1 and 2. Each coloured line represents a different test split for the decision tree.

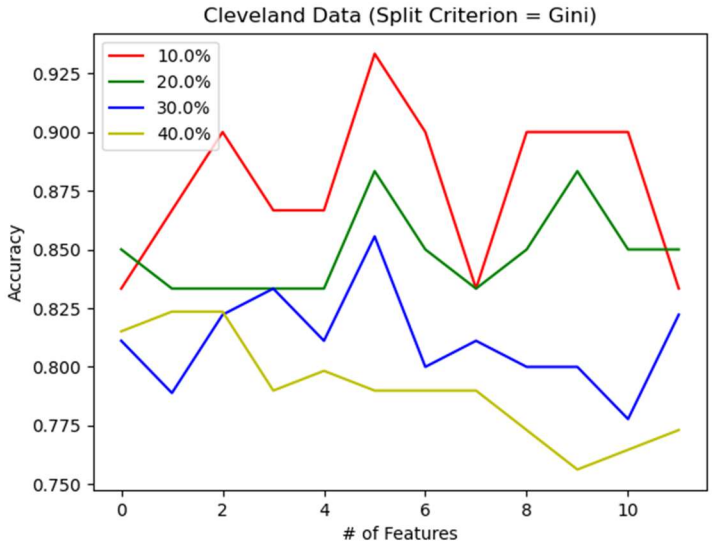

Figure 1: Decision Tree Cleveland Data Using Gini

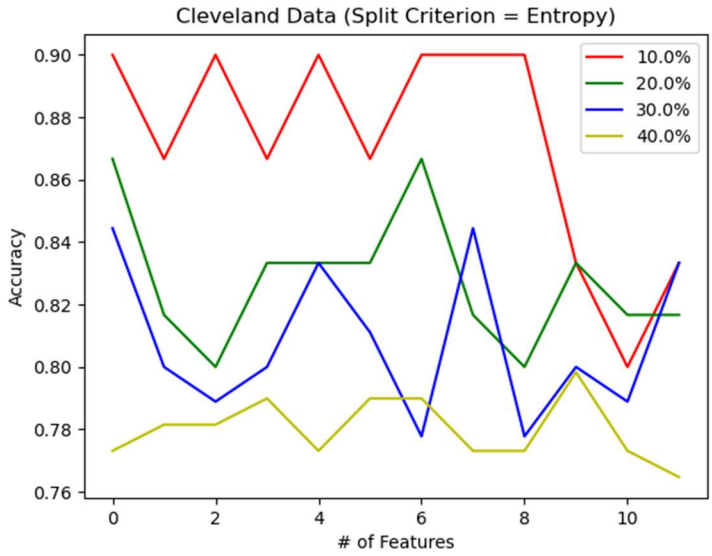

Figure 2: Decision Tree Cleveland Data Using Entropy

The test accuracy plotted against the number of features for the decision tree on the Banknote data is displayed in Figures 3 and 4. Again, each coloured line represents a different test split for the decision tree.

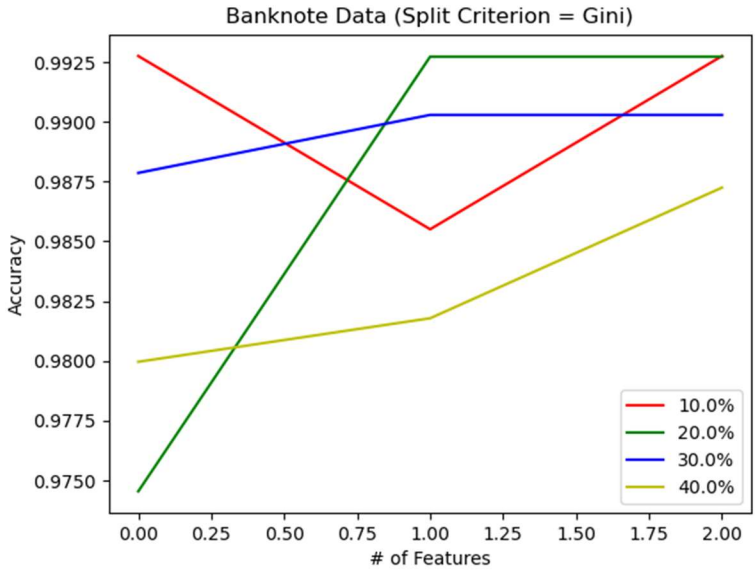

Figure 3: Decision Tree Banknote Data Using Gini

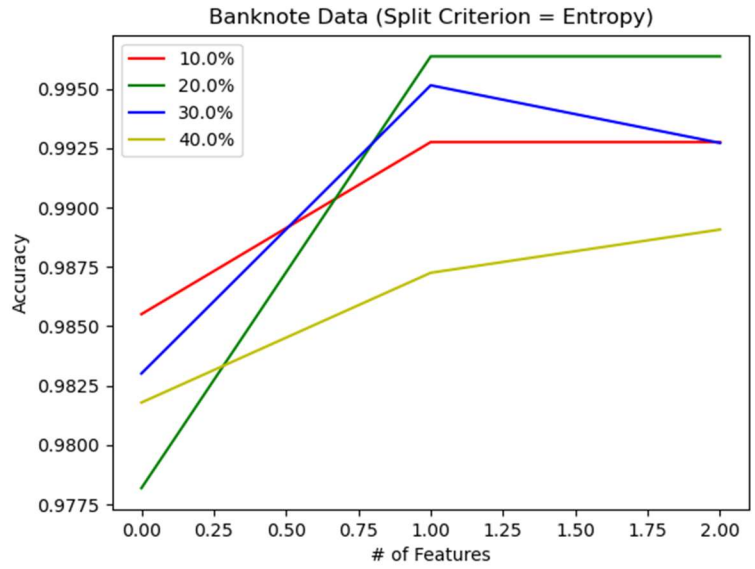

Figure 4: Decision Tree Banknote Data Using Entropy

From examination of the results displayed in Figures 1 and 2, the most accurate decision tree on the Cleveland data has a 10.0% test split, uses Gini as the split criterion, and has 5 features. This tree had an experimental test accuracy of 0.93333. Figure 5 displays the test and training accuracies of this tree plotted against the number of decision tree nodes.

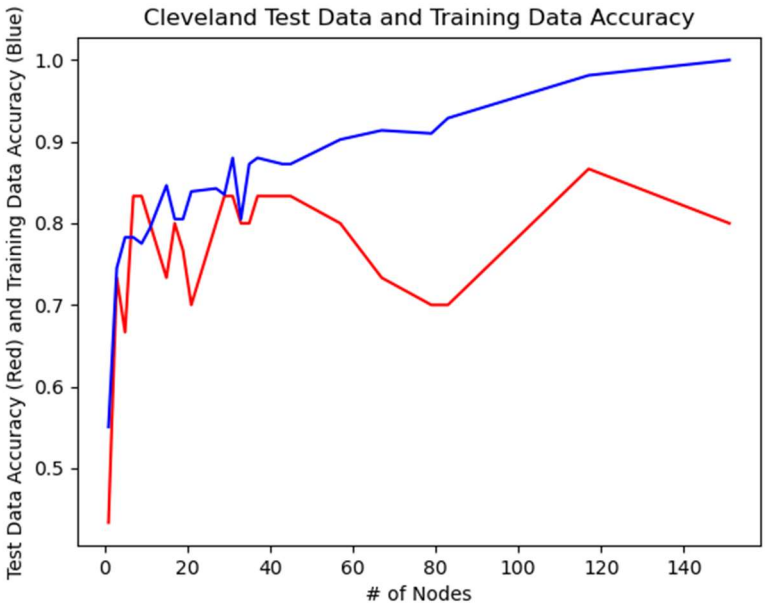

Figure 5: Accuracies for Optimal Cleveland Decision Tree

From examination of the results displayed in Figures 3 and 4, the most accurate decision tree on the Banknote data has a 20.0% test split, uses Entropy as the split criterion, and has 1 feature. This tree had an experimental test accuracy of 0.99636. Figure 6 displays the test and training accuracies of this tree plotted against the number of decision tree nodes.

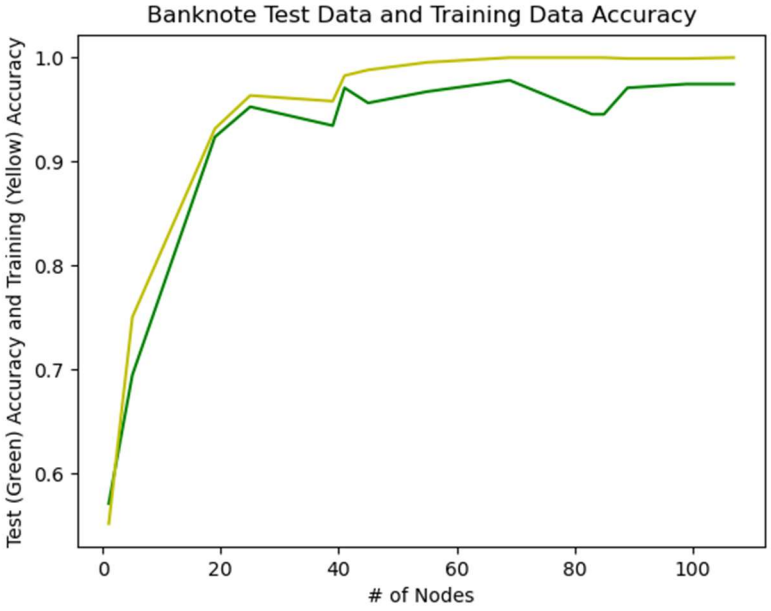

Figure 6: Accuracies for Optimal Banknote Decision Tree

As we can see from Figures 3, 4 and 6, the decision trees produced significantly more accurate classifications on the Banknote data than the Cleveland data, though there was still some variation depending on the values for each parameter. Two potential insights can be drawn from this: the decision tree method benefits from a larger sample size (1372 instances to 303) and fewer attributes (5 attributes to 14). For the Banknote data, it seemed that trees with more features produced more accurate classifications, but the same is not necessarily true for the Cleveland data, which produced a less predictable trend. Additionally, no consensus can be reached from the experiments on whether Gini or Entropy is the optimal split criterion as mixed results were obtained. The optimal decision tree on Cleveland data used Gini while the optimal tree for Banknote data used Entropy.

Furthermore, it can be noted from Figures 5 and 6 that the greater the number of nodes, the higher the decision tree's test and training accuracy. Another insight to be gained from analyzing the results regards the most accurate test split. Experimentally, it seems that test splits of 10.0% and 20.0% are routinely the most accurate. This is evident in Figures 1 through 4.

For future experiments and analysis on the decision tree classification method, it is recommended that additional parameters be varied. For example, while the cost complexity pruning method was used for all tests in this implementation, investigation into the classification accuracy using pre-pruning or another post-pruning rule (such as reduced error pruning) would be useful to future analysis.

3. Random Forest Performance Analysis

The second classification method that was implemented and analyzed was the random forest. No pruning was performed. As part of the analysis, three parameters were varied: the number of trees in the forest, the split criterion, and the number of features. The test split was kept constant at 0.2 for this experiment. For both the Banknote and Cleveland datasets, the results of using 1, 5, 10, 50, 100, and 500 trees were recorded. As for the split criterion, results were recorded using Gini and Entropy. Finally, the number of features was varied from 1 to 4 for the Banknote data and 1 to 13 for the Cleveland data. The test accuracy plotted against the number of features for the random forest on the Cleveland data is displayed in Figures 7 and 9, while the training accuracy plotted against the number of features for the random forest on the Cleveland data is displayed in Figures 8 and 10. Each coloured line represents a different number of trees for the random forest, ranging from 1 to 500.

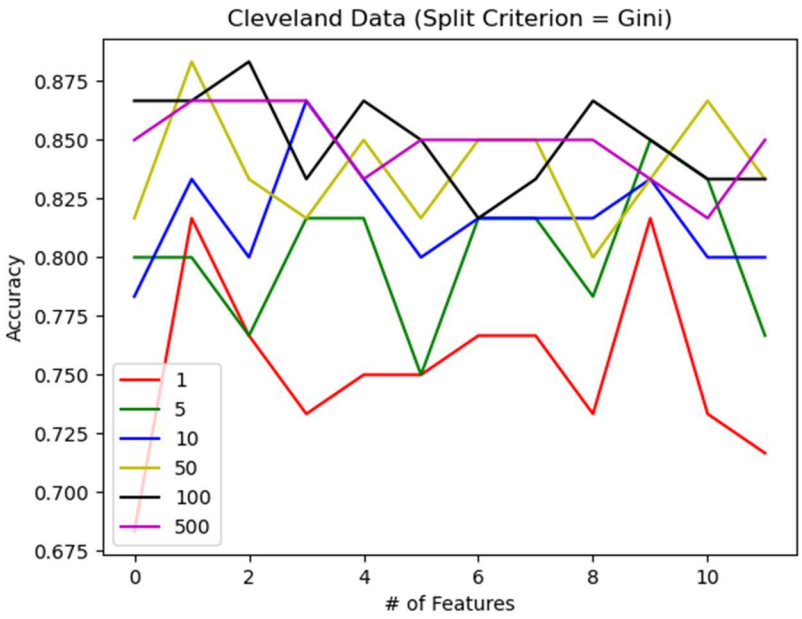

Figure 7: Random Forest Cleveland Test Data Using Gini

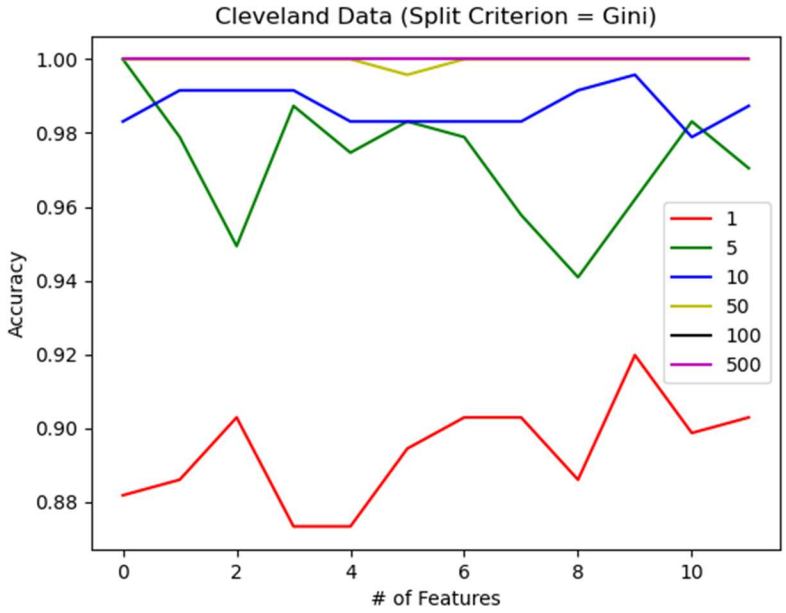

Figure 8: Random Forest Cleveland Training Data Using Gini

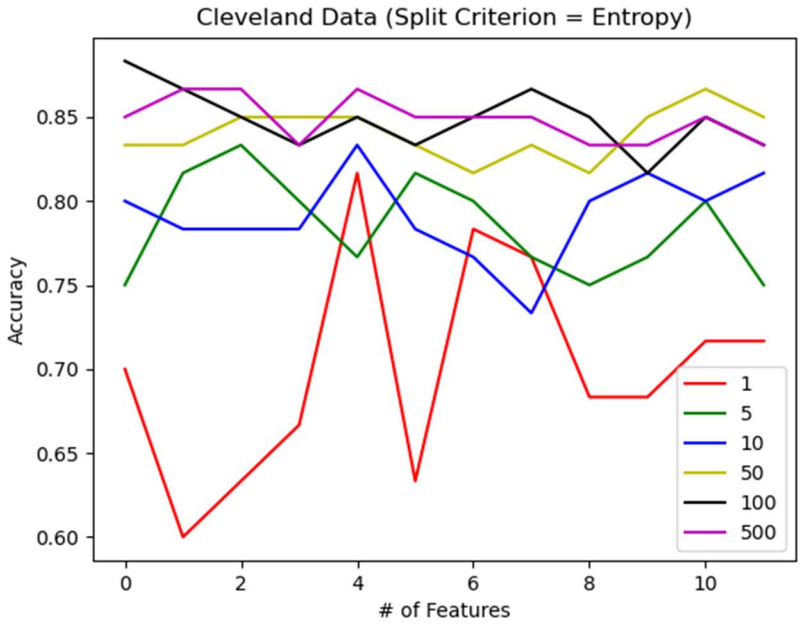

Figure 9: Random Forest Cleveland Test Data Using Entropy

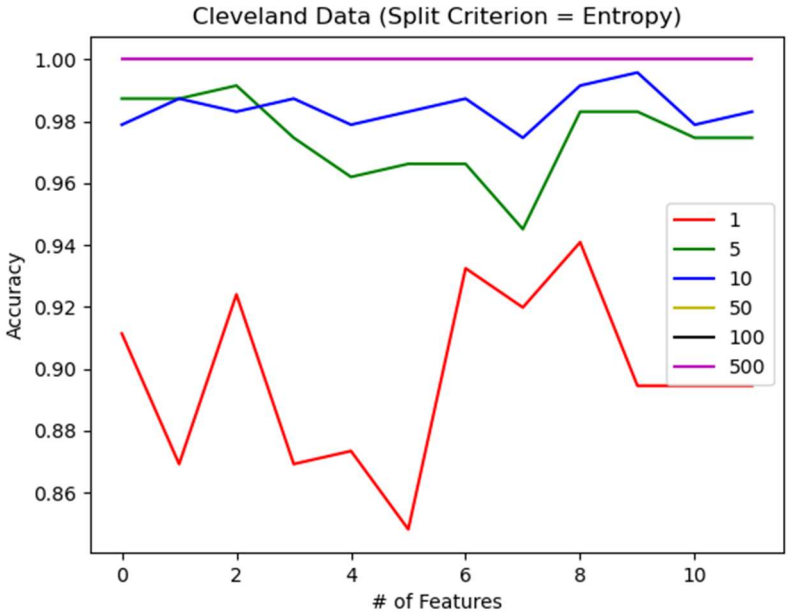

Figure 10: Random Forest Cleveland Training Data Using Entropy

The test accuracy plotted against the number of features for the random forest on the Banknote data is displayed in Figures 11 and 13, while the training accuracy plotted against the number of features for the random forest on the Banknote data is displayed in Figures 12 and 14. Each coloured line represents a different number of trees for the random forest, ranging from 1 to 500.

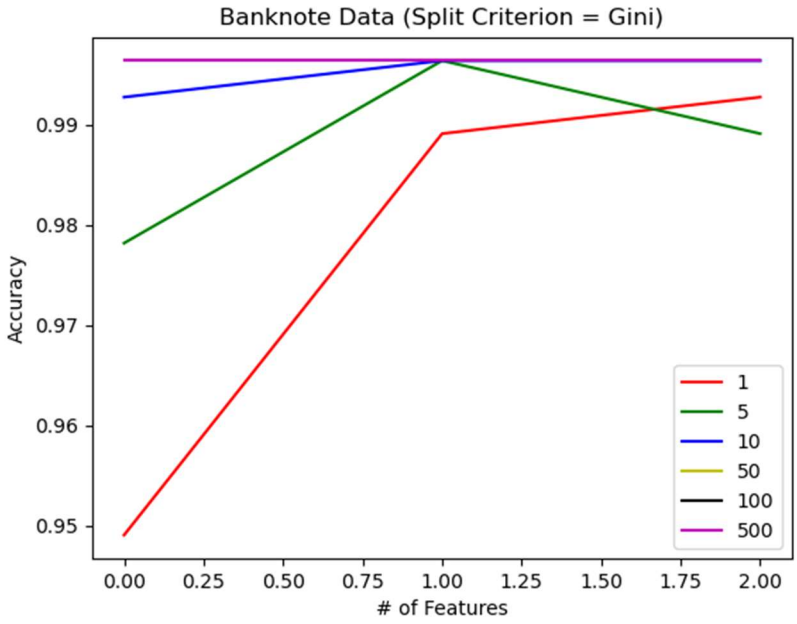

Figure 11: Random Forest Banknote Test Data Using Gini

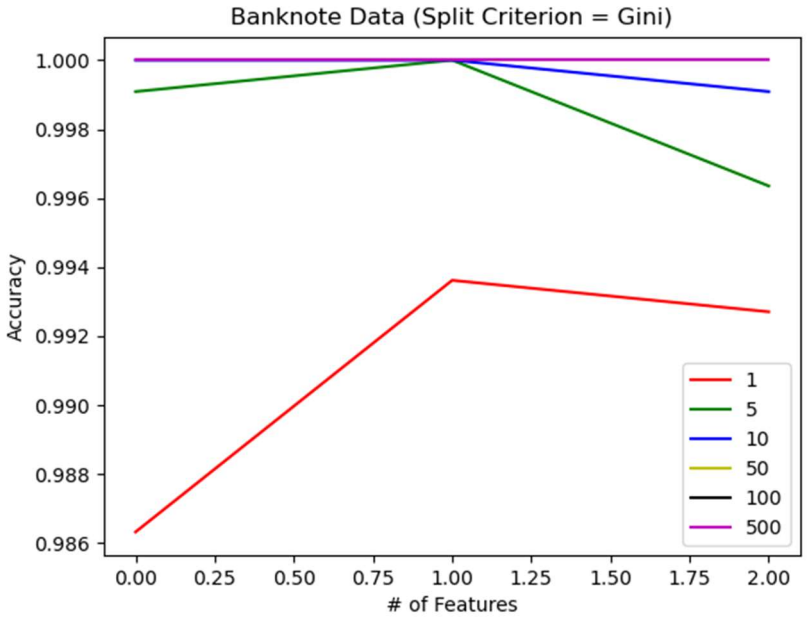

Figure 12: Random Forest Banknote Training Data Using Gini

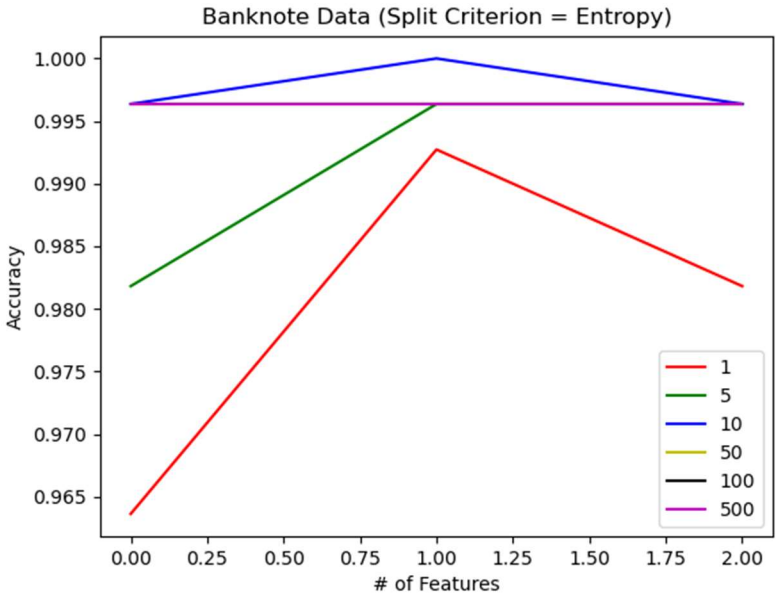

Figure 13: Random Forest Banknote Test Data Using Gini

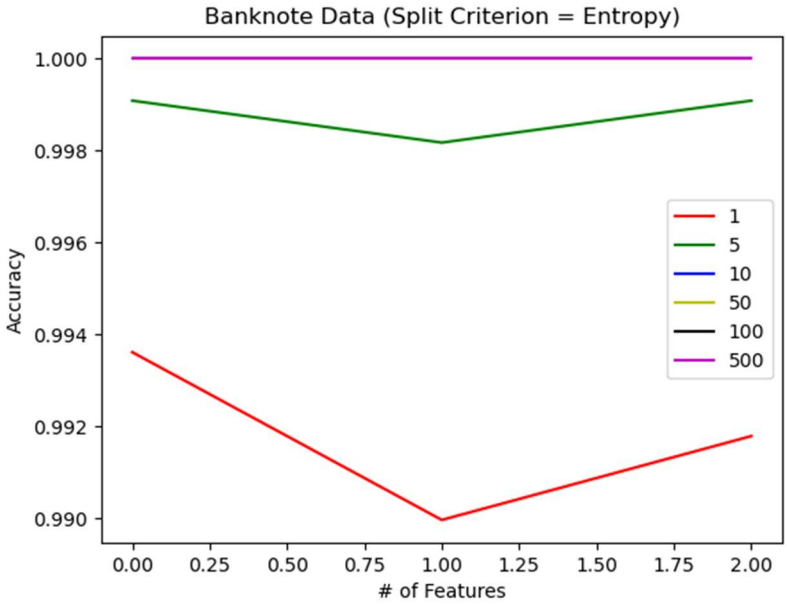

Figure 14: Random Forest Banknote Training Data Using Entropy

From examination of the results displayed in Figures 7 and 9, the most accurate random forest on the Cleveland data has a 20.0% test split, uses Gini as the split criterion, includes 100 trees, and has 2 features. This forest had an experimental test accuracy of 0.883. As for the Banknote data, Figures 11 and 13 show that the most accurate random forest has a 20.0% test split, uses Entropy as the split criterion, includes 10 trees, and has 1 feature. This forest had an experimental test accuracy of 1.00.

Like decision trees, the random forests produced classifications with greater accuracy on the Banknote data. In fact, the accuracy of several parameter combinations achieved 100% accuracy, a marked improvement over the decision tree implementation. Moreover, the training accuracy of every parameter combination for both the Cleveland and Banknote datasets was 100% when including either 100 or 500 trees in the forest. This suggests that a greater number of trees in the forest directly implies a greater classification accuracy. Further to that point, a common inaccuracy appeared when only 1 tree was used in the forest. This is expected as including only 1 tree defeats the purpose of using a random forest.

In contrast to the decision tree results, there seems to be no strong correlation between the number of features and test or training accuracy for random forests. However, in similar fashion to decision trees, no consensus can be reached on which split criterion leads to greater accuracy, as the results for Gini and Entropy were strikingly similar. Additionally, as with decision trees, the larger sample size and fewer attributes appears to benefit the accuracy with which random forests classify the Banknote data.

Because limited insights can be gleaned from this study of random forest classification accuracy, testing additional classification problems on random forests is recommended for future experiments.

4. Neural Network Performance Analysis

The third and final classification method that was implemented and analyzed was the neural network. The neural network implemented for this experiment had one hidden layer and used the default solver for the scikit-learn library's MLPClassifier: the Adam solver for weight optimization. As part of the analysis, three parameters were varied: the learning rate, the size of the hidden layer, and the number of training iterations. The test split was kept constant at 0.2 for this experiment. For both the Banknote and Cleveland datasets, the results of using learning rates of 0.001, 0.01, 0.1, and 0.2 were recorded. As for the hidden layer size, results were recorded using sizes of 5, 10, and 50. Finally, the number of training iterations was varied from 1 to 10 and 1 to 20 (in addition to testing with just 1 iteration) for both the Banknote and Cleveland data.

Because the results for hidden layer sizes of 5 and 10 were so similar, only plots for size 10 will be shown. The test accuracy plotted against the number of iterations for the neural network on the Cleveland data is displayed in Figures 15 and 16, while the training accuracy plotted against the number of iterations for the neural network on the Cleveland data is displayed in Figure 17. Each coloured line represents a different learning rate.

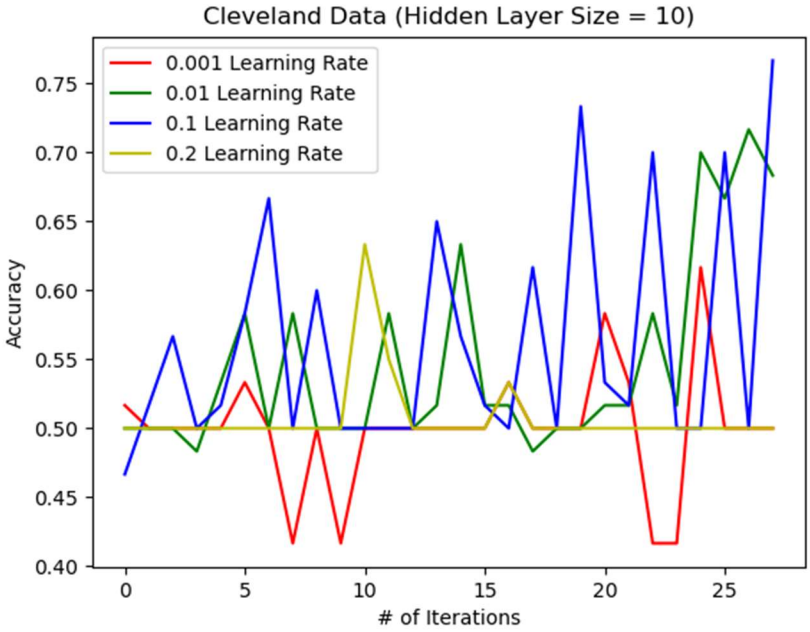

Figure 15: Neural Network Cleveland Test Data Using HL Size 10

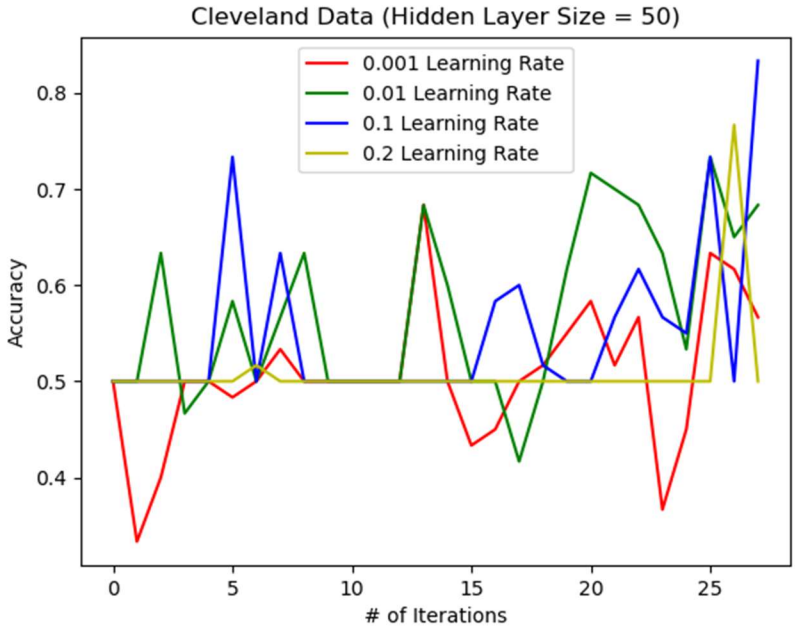

Figure 16: Neural Network Cleveland Test Data Using HL Size 50

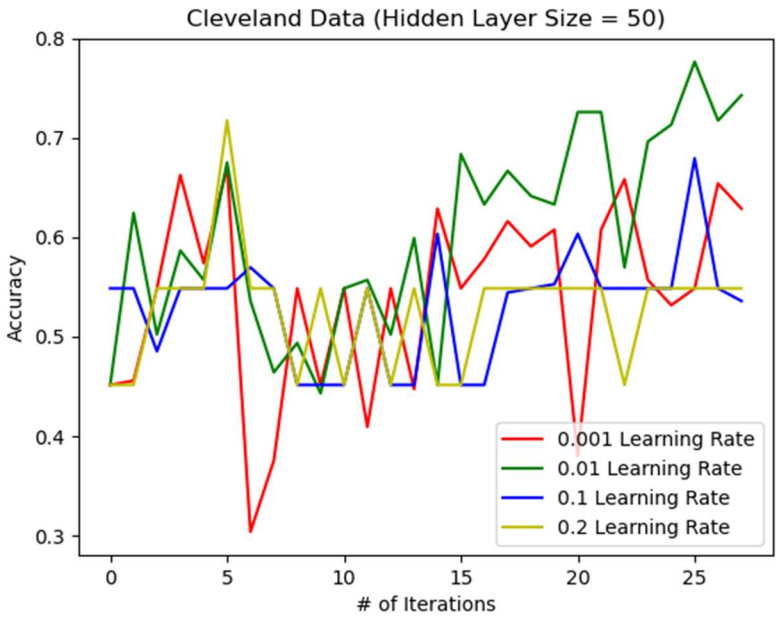

Figure 17: Neural Network Cleveland Training Data Using HL Size 50

The test accuracy plotted against the number of iterations for the neural network on the Banknote data is displayed in Figures 18 and 19, while the training accuracy plotted against the number of iterations for the neural network on the Banknote data is displayed in Figure 20. Each coloured line represents a different learning rate.

Figure 18: Neural Network Banknote Test Data Using HL Size 10

Figure 19: Neural Network Banknote Test Data Using HL Size 50

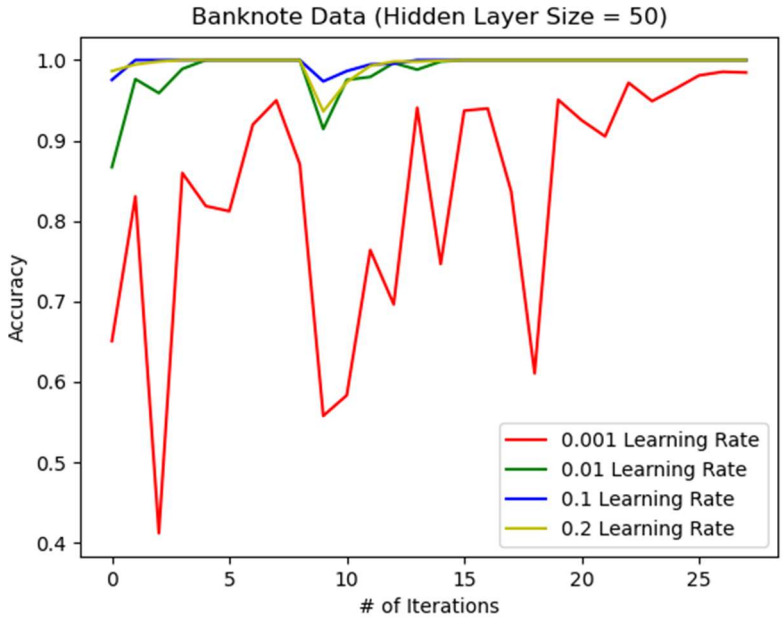

Figure 20: Neural Network Banknote Training Data Using HL Size 50

From examination of the results displayed in Figures 15 and 16, the most accurate neural network on the Cleveland data has a 20.0% test split, a 0.2 learning rate, a hidden layer size of 50, and 18 iterations. This network had an experimental test accuracy of 0.743. As for the Banknote data, Figures 18 and 19 show that the most accurate neural network has a 20.0% test split, a 0.2 learning rate, a hidden layer size of 50, and 5 iterations (though other numbers of iterations achieve the same accuracy). This network had an experimental test accuracy of 1.00.

Neural networks continue the trend of greater accuracy on the Banknote data. Specifically, when learning rates of 0.1 and 0.2 were used, the network implementation consistently achieved 100% test accuracy (regardless of the hidden layer size). Moreover, a trend appeared (especially when observing the 0.001 learning rate) showing that as the number of iterations grew, the accuracy increased. The Cleveland data roughly followed this trend as well, though the accuracy was significantly more erratic when compared to not only the Banknote data, but also to the accuracy of the other two classification methods on the same data. The learning rate appears to be strongly connected to the accuracy, with a higher learning rate leading to more accurate results. This fact is especially evident in Figures 18 and 19.

For future studies, it is recommended that the implementation of the neural network be optimized and cleaned up to reduce inconsistency in the Cleveland data. One way to achieve this may be to alter the solver used from ‘adam’ to ‘sgd’ (sigmoid) or lbfgs. Another option is to move away from the Multi-Layer Perceptron Classifier (MLPClassifier) implementation altogether. Additionally, increasing the number of iterations may assist in providing more readable and consistent results. However, more iterations would also mean much longer execution times.