SEng 474 - Assignment 1

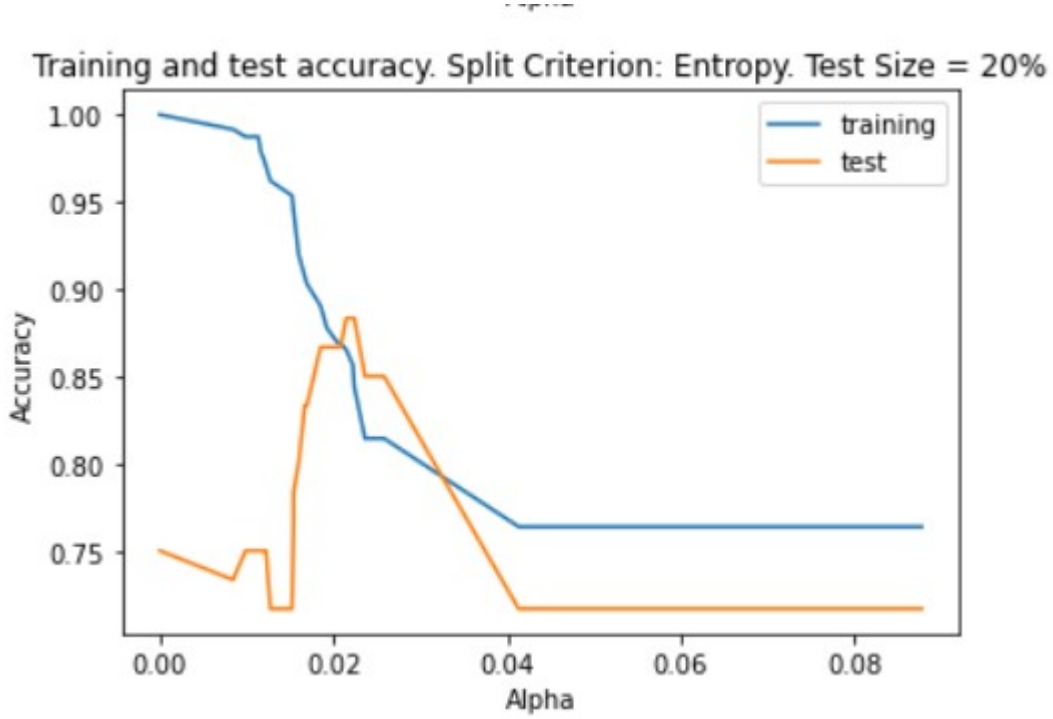
Kenil Shah

V00903842

Cleveland Dataset

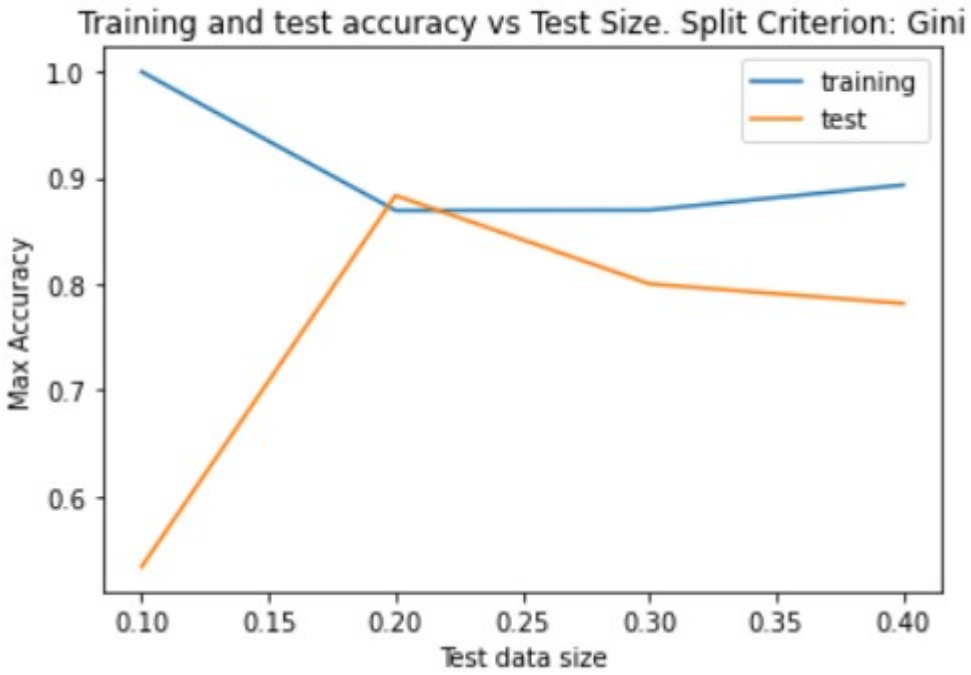
1. Decision Tree

The first classification method we use on our cleveland dataset is the a decision tree with cost complexity pruning. For the analysis of our classification we compare the complexity parameter α with training and test accuracy of the decision tree as it continues to get pruned. The test size used for the initial analysis was 0.2. Two split criterion were used - Entropy and Gini index.



In this case both entropy and gini had equal accuracy for a test size split of 20%, of approximately 88.4%.

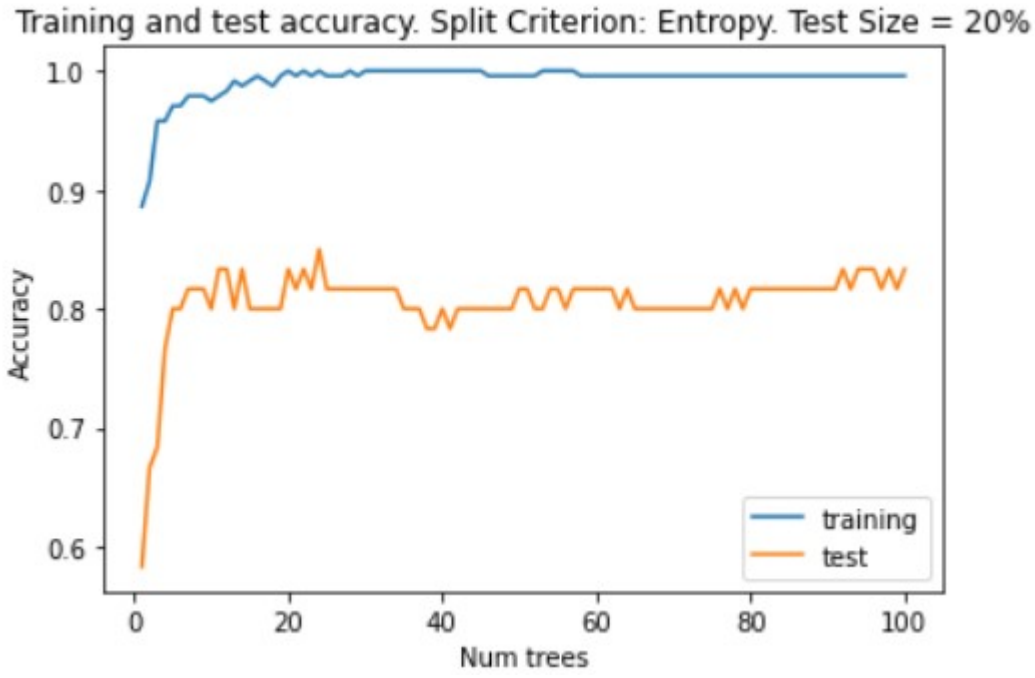
Then the training and test accuracy was measured against a variation of test sizes ranging from 0.1 to 0.4. Again the split critetion used were Entropy and Gini index.



In both cases above, a 80-20 training-test split gives the best accuracy.

2. Random Forests

The second classification method we use on our cleveland dataset is the random forests. We use a test size of 0.2 for the analysis of this method. The number of trees per forest varied form 1 to 100, for a total of 100 forests. The split criterion used were - Entropy and Gini index. We plot training and test accuracy against the number of trees in a forests.



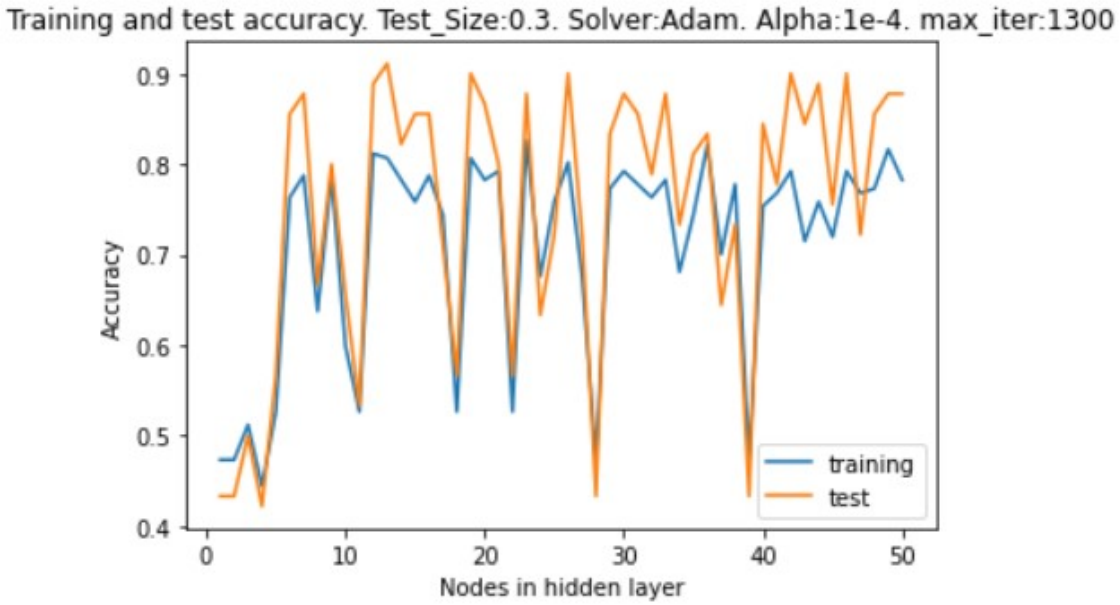
In case of entropy, we get an accuracy of 85% in a forest size of 24. While in the case of gini index, we get a slighly higher accuracy of 88.4% in a forest size of 30.

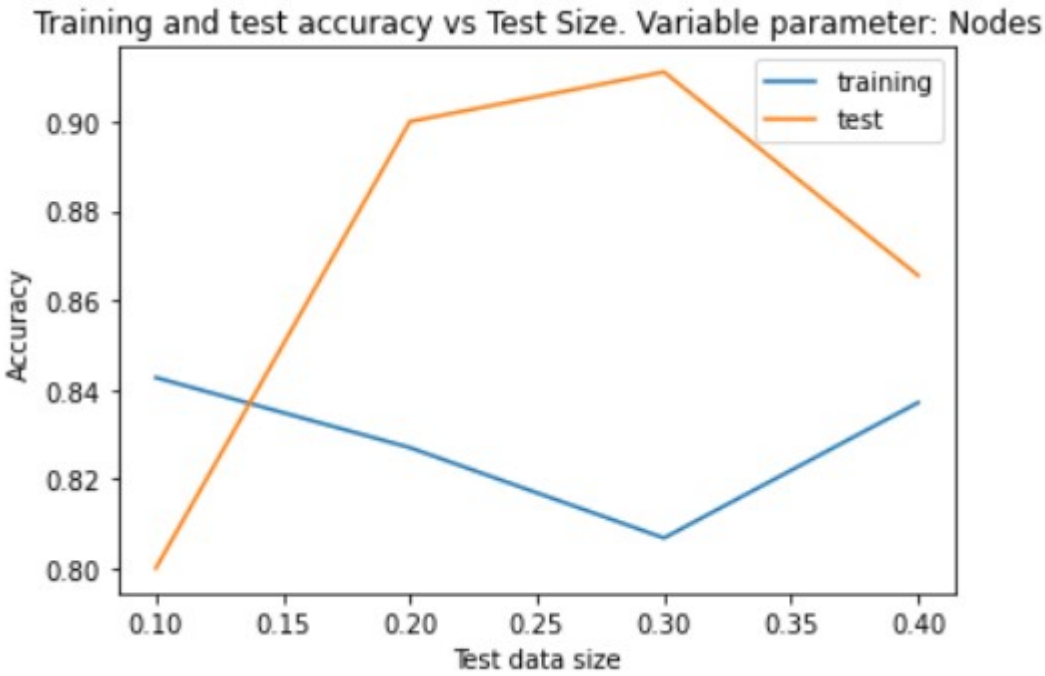
Then we add another parameter max depth to our random forests. Max depth varies form 1 - 10. Now we run the experiment again with the new depth parameter and the previous number of trees parameter and use the split criterion gini index.

In this case, we end up getting an accuracy of 91.7% with a forest size of 26 with max depth 2.

3. Neural Network

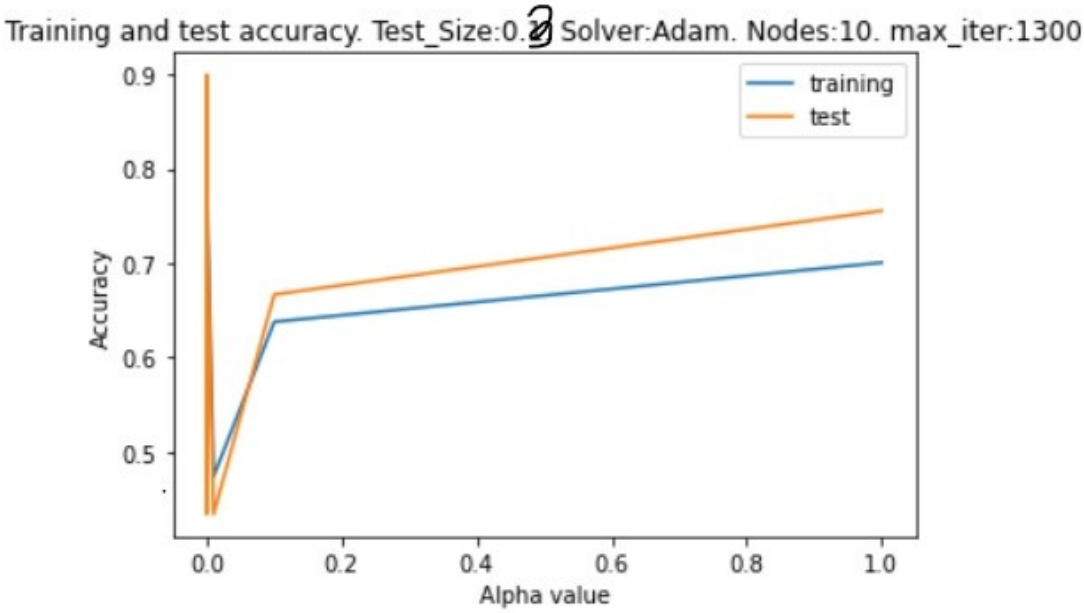
The second classification method we use on our cleveland dataset is a neural network. We first vary the number of nodes in the 1 hidden layer from 1-50. We use the default adam solver, with alpha=1e-4 and max of 1300 iterations. We experimented with test size splits of 0.1 to 0.4.

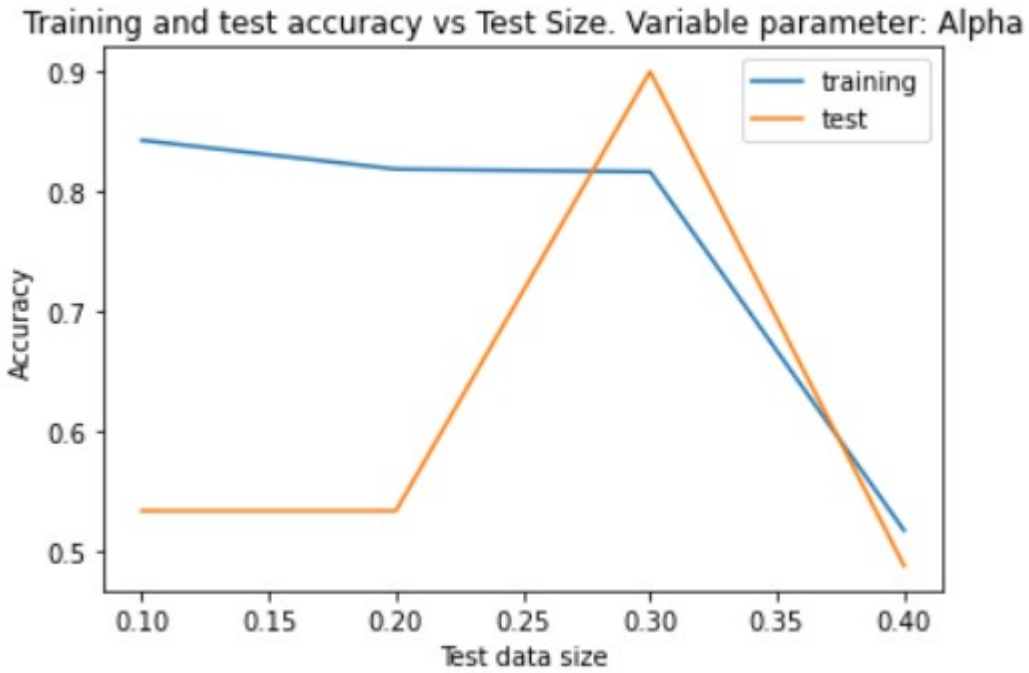




We found that a split of 70-30 training-test gives the best accuracy of approx 92% with 13 nodes in the hidden layer.

Now we vary the value of alpha from 1 to 1e-9. We use 10 nodes in our hidden layer and keep the rest of the parameters same as before. Again we experimented with test size splits of 0.1 to 0.4.





We again see that a split of 70-30 training-test gives the best accuracy of 90% with an alpha value of 0.0001

References

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

https://www.w3schools.com/python/pandas/pandas_csv.asp

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

https://scikit-learn.org/stable/auto_examples/neural_networks/plot_mlp_alpha.html#sphx-glr-

auto-examples-neural-networks-plot-mlp-alpha-py