Machine Learning Lab 2
Jordan Harris
Nov 9th 2021

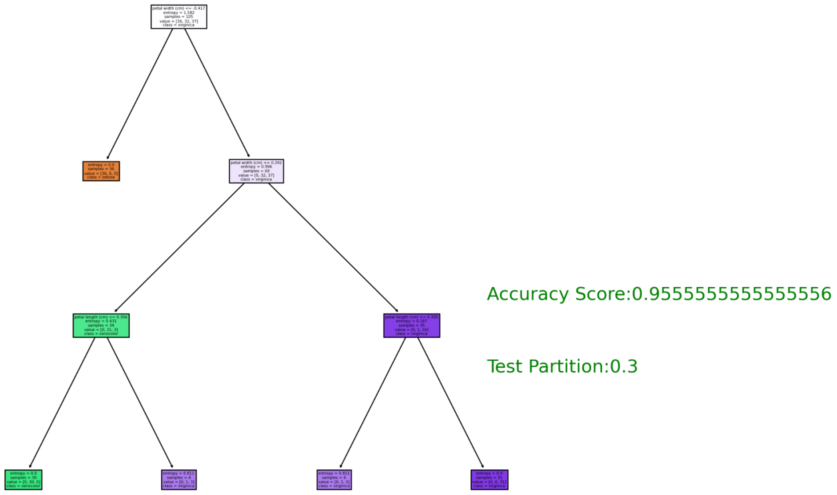
Decision Trees

I ran this test with many different configurations:

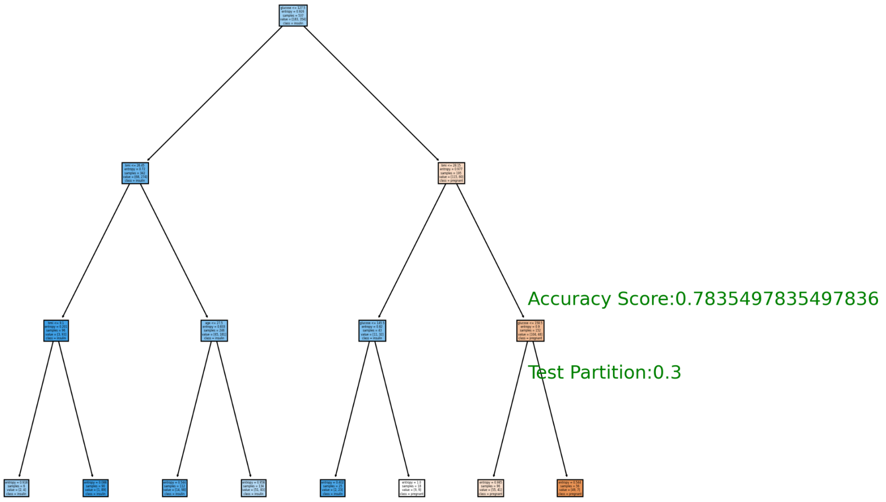
- Max Depth was seen to be optimized at 3 for both the Iris & Diabetes Data Set
- Training Partitions were done at .30, .50 & .70 for the test size.
- I also ran the test for ‘best’ split instead of the ‘random’ split approach. Which chooses the root node to be the feature with the highest ‘importance’ versus choosing one at random.
- Finally I used the “entropy” attribute to try and maximize the information gain/minimize the cross-entropy.

1.1. *Partition the dataset into a training and a testing set. Run a decision tree learning algorithm using the training set. Test the decision tree on the testing dataset and report the total classification error (i.e. 0/1 error).*

Iris

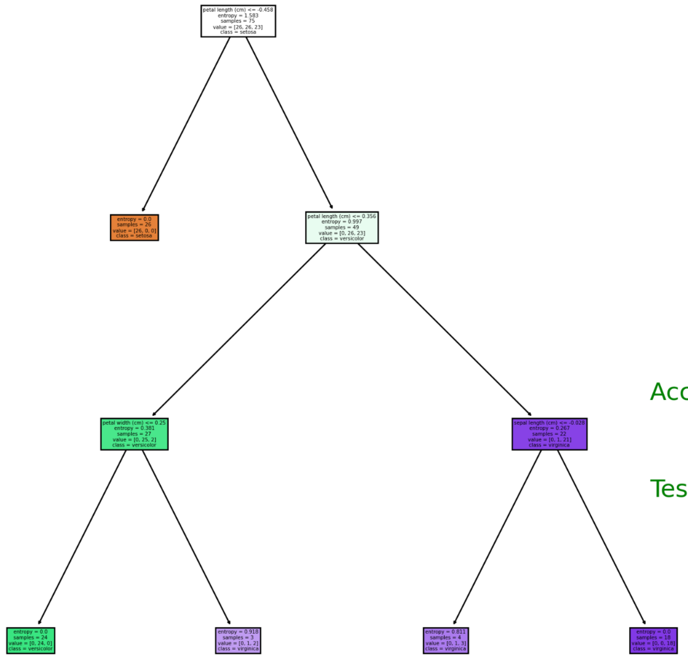


Diabetes



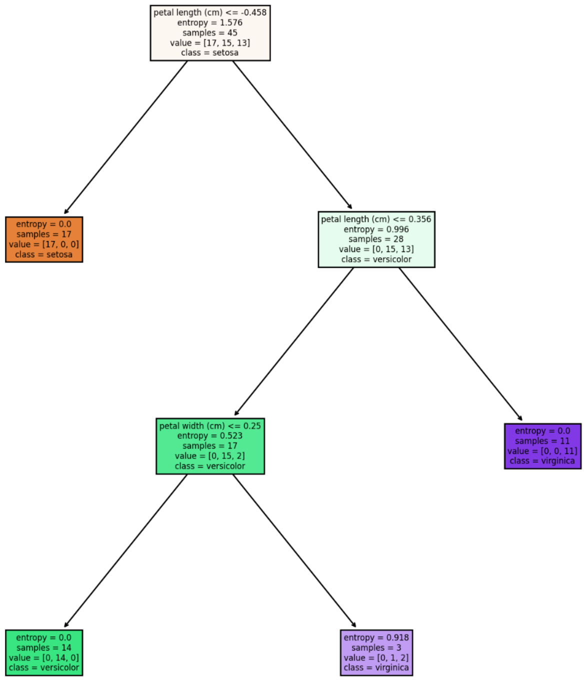
1.2. *Repeat the experiment with a different partition of the data. Plot the resulting trees.*

Iris



Accuracy Score:0.9466666666666667

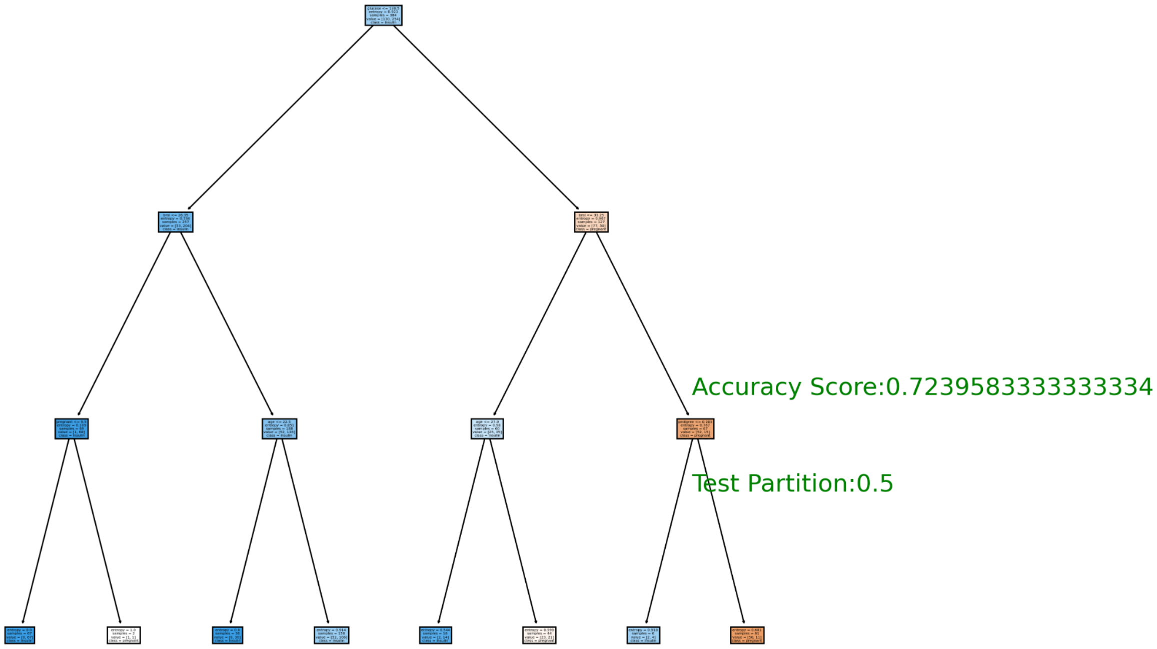
Test Partition:0.5

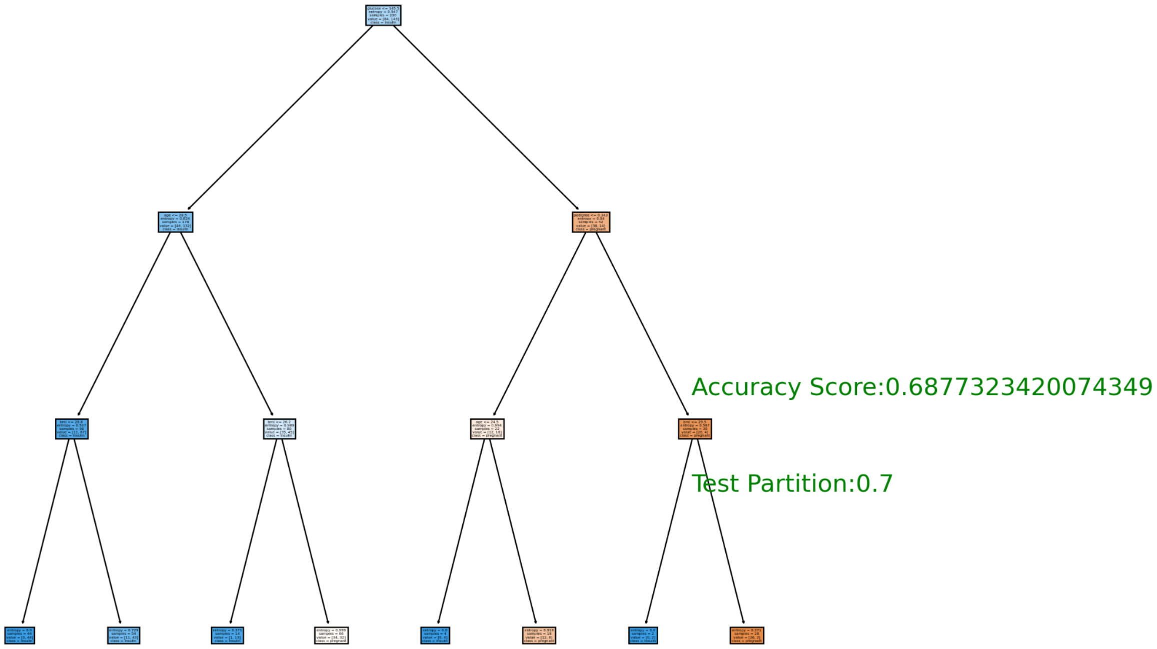


Accuracy Score:0.9523809523809523

Test Partition:0.7

Diabetes





1.3 *Are the previous trees very similar, or very different? Explain why.*

In regards to the Iris data set, the Accuracy Score only saw minimal changes with the different test partitions. But the structure of the graph did become more unbalanced as the test set grew larger, i.e. Test Partition = .70. Which indicates an unbalanced ratio of attribute decisions being made overall and leads to a higher specificity to the data set at hand which can be problematic and unstable and is a tell-ta;e sign of over-fitting.

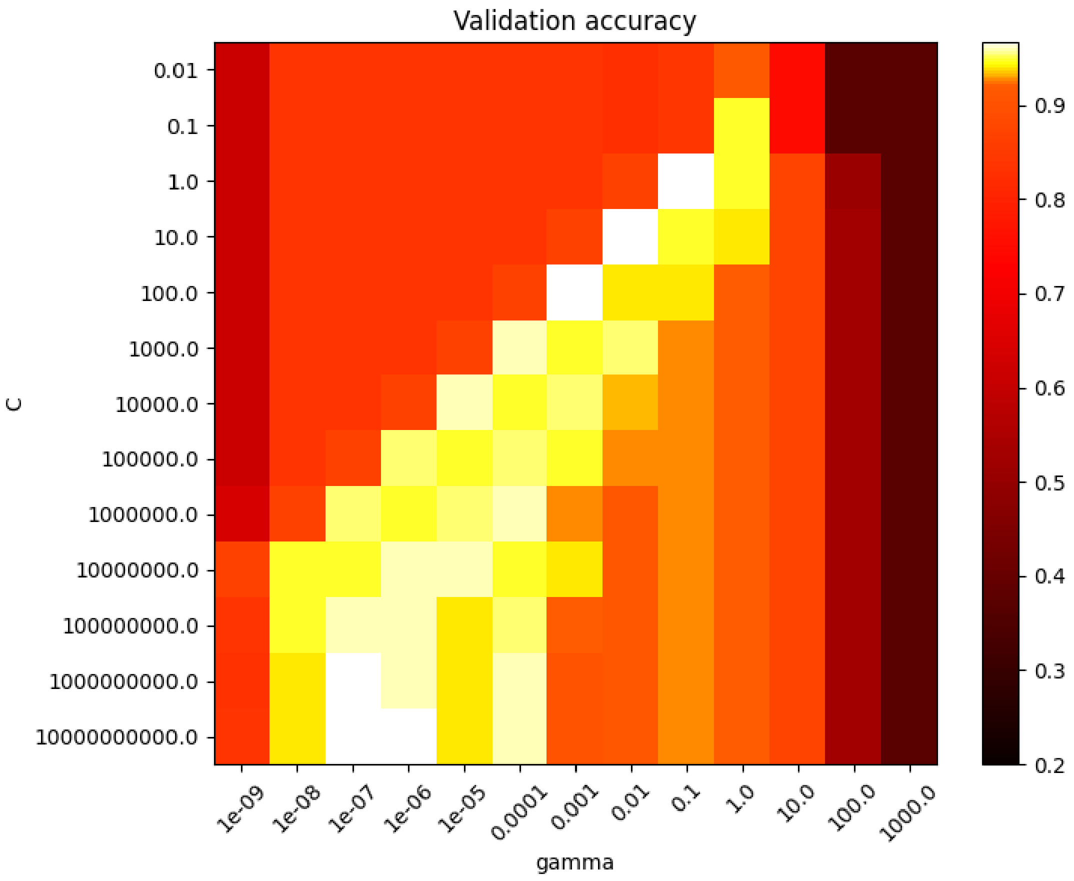
In regards to the Diabetes data set, the Accuracy Score was optimized at the lowest test partition = .30. But saw no change in the structure in the resultant tree. Meaning that the tree was able to reach an optimal configuration with only a very small sample of training/test data resulting in a very balanced tree.In the case of decision trees there is a tendency toward overfitting training data. Which I think is mitigated by keeping the depth low to underfit and counteract this issue.

Support Vector Machines

I ran this test with many different configurations:

- Max Depth was seen to be optimized at 3 for both the Iris & Diabetes Data Set
- Training Partitions were done at .30, .50 & .70 for the test size.
- I also ran the test for 'best' split instead of the 'random' split approach. Which chooses the root node to be the feature with the highest 'importance' versus choosing one at random.
- Finally I used the "entropy" attribute to try and maximize the information gain/minimize the cross-entropy.

2.1 *Run SVM to train a classifier, using radial basis as kernel function. Apply cross-validation to evaluate different combinations of values of the model hyper-parameters (box constraint C and kernel parameter γ).*



2.2. *How sensitive is the cross-validation error to changes in C and \gamma\ Choose the combination of C and \gamma\ that minimizes the cross-validation error, train the SVM on the entire dataset and report the total classification error.*

The gamma value has a very direct and sensitive effect on the Validation Error. When gamma is too small then the model becomes too focused on the minutia of each datapoint and actually just becomes equal to the whole training set itself. C effectively acts as a regularization metric for the model but if gamma grows too large then no regularized value of C can prevent the effects of overfitting as shown with the dark red colors in the right hand sector of the graph. The diagonal of well positioned C and γ values is caused by low gamma values becoming more accurate with increasing values of C which puts more and more emphasis on classifying each point correctly. The slope of correct values also shows that some of the ‘intermediate’ values of gamma produce pretty good validation error as well because as C grows it is not as necessary to regularize by enforcing a larger margin.