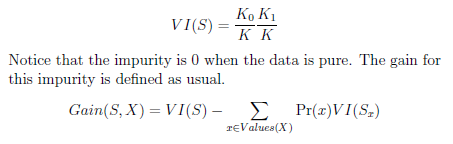
Madeline Jiang

Mxj121230

1.**Which impurity heuristic (Entropy/Variance) yields the best classification accuracy?**

Out of the two impurity heuristics, Entropy yielded the best overall classification accuracy, as best seen in larger data sets. There were only 3 cases where variance outperformed entropy in the initial tree generation in terms of accuracy (valid c300d100, valid c1000d1000, test c1500d100). An explanation is that since variance, as given:



relies on calculating the squared ratio between training examples that have class=0,1 over the total cases, larger sample sizes adversely affect determining highest gain as with larger denominators, VI(S) approaches 0. Since Entropy uses logarithms applied to these fractions, Entropy reduces the chance of underflow and thus allows for more selectivity in gain selection.

For post-pruning via reduced error, the difference between entropy and variance accuracy became more balanced with 4 test set accuracies variance outperforming entropy out of 15 total sets. Overall entropy was the better heuristic, especially for pre-pruning accuracy (Table 3).

**How does increasing the number of examples and/or the number of clauses impact the (accuracy of the) two impurity heuristics. Explain your answer.**

Increasing the number of examples and number of clauses both increased the accuracies of the impurity heuristics. A larger sample size means more observation were able to contribute to the decision of each leaf outcome, making the tree more accurate and increasing the confidence for a correct classification. For example, we can be surer of a classification for a majority value if there were 1 datapoint vs 6 datapoints assigned to a leaf node. Thus, increasing the number of examples reduces the chances of overfitting. Increasing the number of examples also resulted in larger trees, as more combinations and patterns can be generalized with larger sample sizes than from smaller sets of data.

Increasing the number of clauses resulted in smaller trees and higher accuracies. As seen in the Table 4, with increasing number of clauses on the same sample size resulted in decreasing trend of the number of inner nodes counted given the same number of observations. This is to say that an increasing number of clauses resulted in simpler trees which is what we prefer for decision trees following the principles of Occam’s razor, reducing the chances of overfitting, resulting in better accuracies.

2. **Which overfitting avoidance method (reduced error pruning/ depth- based pruning) yields the best accuracy? Again, how does increasing the number of examples and/or the number of clauses impact the (accuracy of the) two overfitting avoidance methods. Explain your answer.**

In all reduced error pruning outperformed depth-based pruning for accuracy. Clearly reduced error pruning yielded best accuracies as each inner node was iteratively optimized for the best accuracy for pruning vs the brute force nature of depth-based pruning. Furthermore, depth-based pruning was also lacking in that many of the decision trees did not reach those depths. Not even the largest example sets reached a max depth of beyond 15.

Since larger trees performed better in terms of accuracy pre-pruning, there was less improvement in accuracy after reduced error pruning for larger datasets given the same number of clauses (see Table 5). Thus, reduced error pruning improved the accuracy of smaller sample-sized trees, as pruning would address over-fitting.

For both methods increasing the number of clauses impacted pruning results by reducing the number of nodes that needed to be pruned, as tree size was inherently smaller. By increasing number of clauses, data was inherently distributed in a more skewed manner as seen in 2 variable input vs 3 variable input OR-Boolean algebra tables, where the former only has 25% False class and the latter has 12.5% False class, and thus accuracy was improved in post-prune data as the number of clauses increased.

**3. Are random forests much better in terms of classification accuracy**

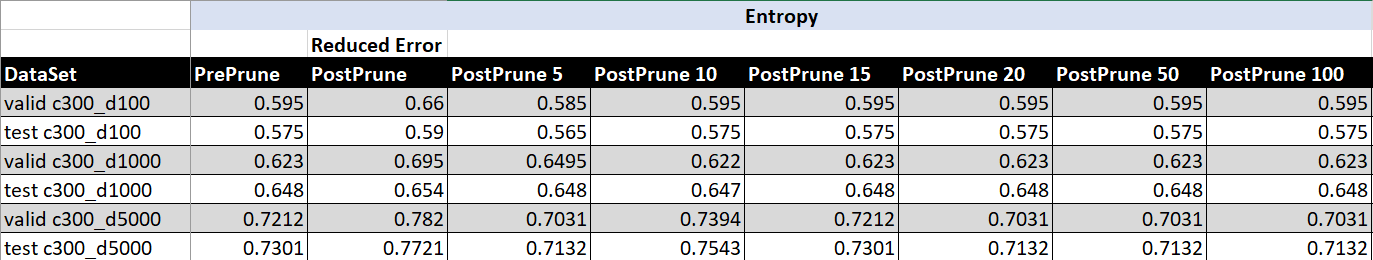
**than your decision tree learners? Why? Explain your answer.**

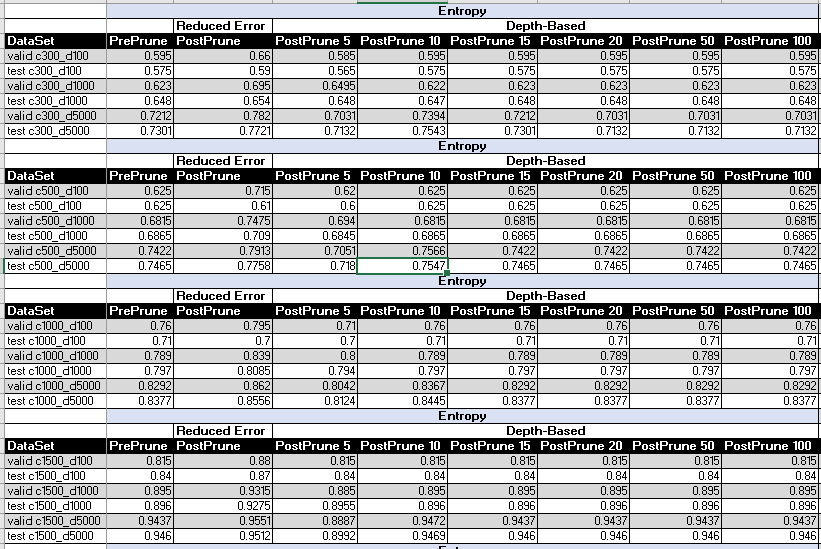
Random forests address overfitting to the training set by averaging multiple decision trees. By averaging multiple decision trees, there is a smoothing of irregular classifications and outliers to our tree. Compared to our single decision tree learner that can only generate one opinion given a set of feature instances, a random forest allows for higher accuracy by taking the votes from multiple decision trees to classify a datapoint. Since a random forest constructs trees by evaluating subsets of features each construction, it allows for more accurate prioritization of features when considering classification.

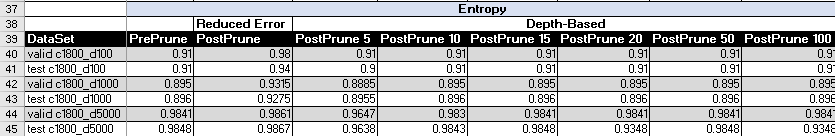
**Notes on methods and assumptions:**

* In the case of equal number of 0’s and 1’s in a leaf node, majority class was set to 1 for leaf nodes.
* GridSearch was performed with parameters n\_estimators=500,750,1000 max\_depth=[5,10,15,20] (as our decision trees did not go beyond 15 in depth). Training and validation sets were merged for GridSearch/Random Forest and compared to test data.
* Additional basic decision tree visualization functionality included
* During Reduced Error pruning, conservatively chose the max accuracy pruned node starting from the deeper nodes by performing a BFS on inner nodes, then reversing the order of the list. Thus deeper node accuracies would be listed earlier in the list and the deeper node would be pruned if two or more nodes shared the same accuracies.

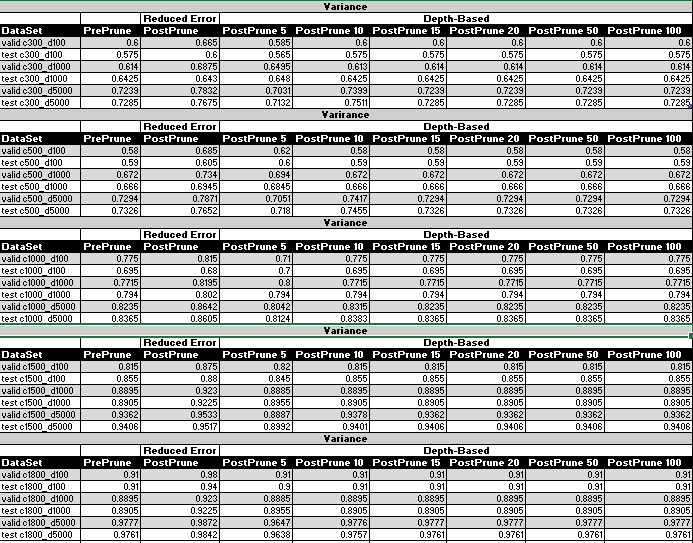
**Table 1**: Decision Tree Learner Accuracy data for clauses={300,500,100,1500,1800} and examples = {100,1000,5000} using Entropy for node selection and reduced error as well as depth max={5,10,15,20,50,100} to prune tree







**Table 2:** Decision Tree Learner Accuracy data for clauses={300,500,100,1500,1800} and examples = {100,1000,5000} using Variance for node selection and reduced error as well as depth max={5,10,15,20,50,100} to prune tree



**Table 3: Comparison of Entropy and Variance heuristic accuracy values for pre and post- reduced error pruning for test set data. Highlighted in yellow are 4 sets where variance outperformed entropy out of total test sets=15.**



**Table 4:Comparison of pre and post-prune accuracies for a given sample size given increasing number of clauses.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Entropy** | |  |  | **Variance** | |
| **Sample Size=100** | **Number of inner nodes** | |  | **Sample Size=100** | **Number of inner nodes** | |
| **Clauses** | **Pre-Prune** | **Post-Prune** |  | **Clauses** | **Pre-Prune** | **Post-Prune** |
| 300 | 27 | 22 |  | 300 | 27 | 21 |
| 500 | 28 | 19 |  | 500 | 27 | 12 |
| 1000 | 20 | 13 |  | 1000 | 20 | 13 |
| 1500 | 13 | 9 |  | 1500 | 14 | 11 |
| 1800 | 10 | 5 |  | 1800 | 10 | 5 |
|  |  |  |  |  |  |  |
| **Sample Size=1000** | **Number of inner nodes** | |  | **Sample Size=1000** | **Number of inner nodes** | |
| **Clauses** | **Pre-Prune** | **Post-Prune** |  | **Clauses** | **Pre-Prune** | **Post-Prune** |
| 300 | 256 | 65 |  | 300 | 265 | 42 |
| 500 | 225 | 76 |  | 500 | 240 | 61 |
| 1000 | 156 | 90 |  | 1000 | 165 | 68 |
| 1500 | 86 | 64 |  | 1500 | 86 | 67 |
| 1800 | 31 | 26 |  | 1800 | 35 | 25 |
|  |  |  |  |  |  |  |
| **Sample Size=5000** | **Number of inner nodes** | |  | **Sample Size=5000** | **Number of inner nodes** | |
| **Clauses** | **Pre-Prune** | **Post-Prune** |  | **Clauses** | **Pre-Prune** | **Post-Prune** |
| 300 | 962 | 277 |  | 300 | 1004 | 321 |
| 500 | 896 | 293 |  | 500 | 987 | 310 |
| 1000 | 617 | 294 |  | 1000 | 646 | 253 |
| 1500 | 241 | 166 |  | 1500 | 255 | 161 |
| 1800 | 241 | 124 |  | 1800 | 255 | 114 |
|  |  |  |  |  |  |  |

**Table 5:** Accuracy Difference for Reduced Error Pruning for Entropy and Variance heuristic trees for given dataset. Highlighted in yellow are 16 out of 30 sets where reduced error resulted in a greater difference in variance accuracy than entropy accuracy. Note that these columns compare pre and post prune accuracy values, not entropy post prune and variance post prune accuracy values



**Table 6: Random Forest results for criterion=Gini Index, Entropy criteria using GridSearchCV to select for best values for n\_estimators=500,750,1000, max\_depth=[5,10,15,20]**

|  |  |  |
| --- | --- | --- |
| **Random Forest** | **Gini** | **Entropy** |
| test c300\_d100 | 0.86 | 0.875 |
| test c300\_d1000 | 0.9115 | 0.9155 |
| test c300\_d5000 | 0.9402 | 0.9409 |
| test c500\_d100 | 0.93 | 0.95 |
| test c500\_d1000 | 0.965 | 0.9655 |
| test c500\_d5000 | 0.9677 | 0.966 |
| test c1000\_d100 | 0.995 | 0.995 |
| test c1000\_d1000 | 0.9935 | 0.9925 |
| test c1000\_d5000 | 0.9967 | 0.9964 |
| test c1500\_d100 | 1.0 | 1.0 |
| test c1500\_d1000 | 1.0 | 1.0 |
| test c1500\_d5000 | 0.9999 | 0.9999 |
| test c1800\_d100 | 1.0 | 1.0 |
| test c1800\_d1000 | 1.0 | 1.0 |
| test c1800\_d5000 | 1.0 | 1.0 |