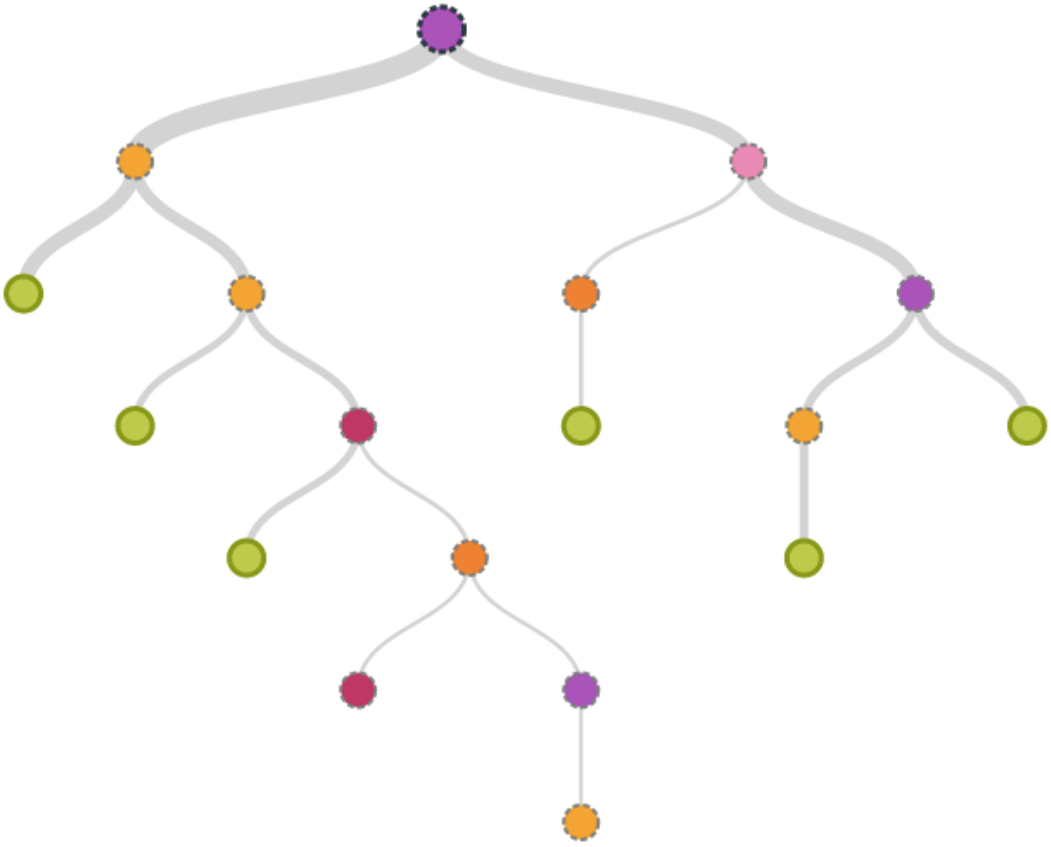
Decision Trees

-Anil Kumar 2013CSB1008

-Ananya Kirti 2013CSB1048



Introduction

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm.[1]

In our experiment we created a decision tree using the ID3 algorithm, using information gain to select the best attribute. While traversing the data, we stored the possible values of all attributes, and stored them. This was used in creating the tree.

And ran various tests on it. Each experiment was run 10 times, and all the data used to create this report (included in the source files) is the average of the said experiments.

The code is well documented, along with all the relevant java docs.

How to run the code?

Code is written in Java. Compile the code, and pass the file name as the 1st argument, and experiment number as the 2nd argument.

```
java DecisionTree data experiment_number
eg. java DecisionTree ticdata2000.txt  1
```

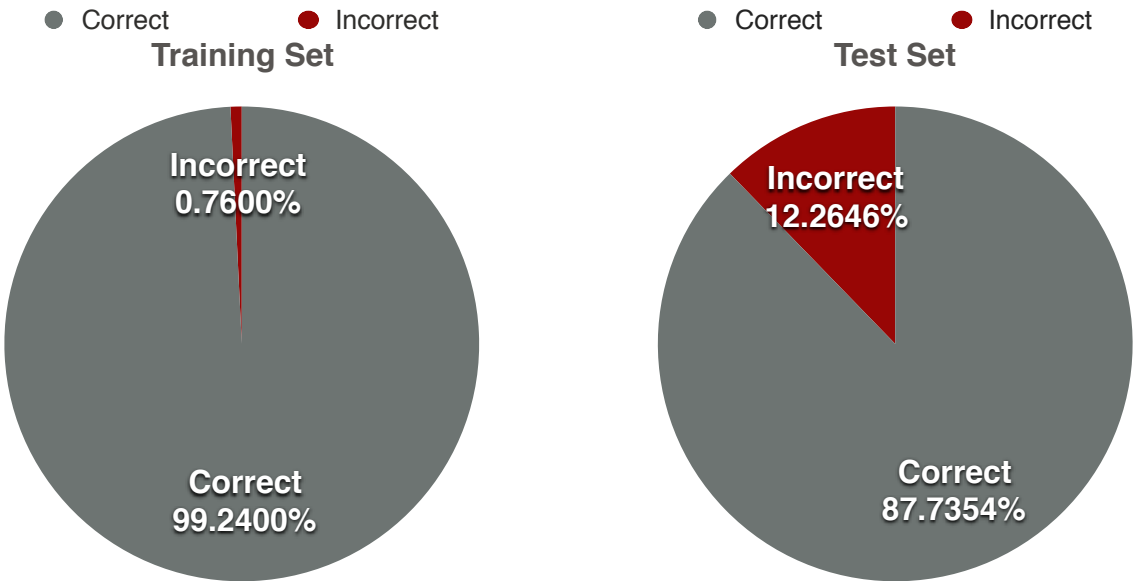
There are 4 experiments to be done, pass the appropriate experiment_number for each experiment.

You can change the parameters such as maxLevel, noise (as a percentage), numberOfForest in Test1.java Test2.java Test3.java

The main function calls these functions, depending on the experiment_number

Results and Methodology

We randomly selected 1000 instances from the data set, and these became the **Training Set,** the rest became the **Test Set**. We then created the ID3 tree using information gain to select the best attribute to create the Decision Tree using the Training Set.

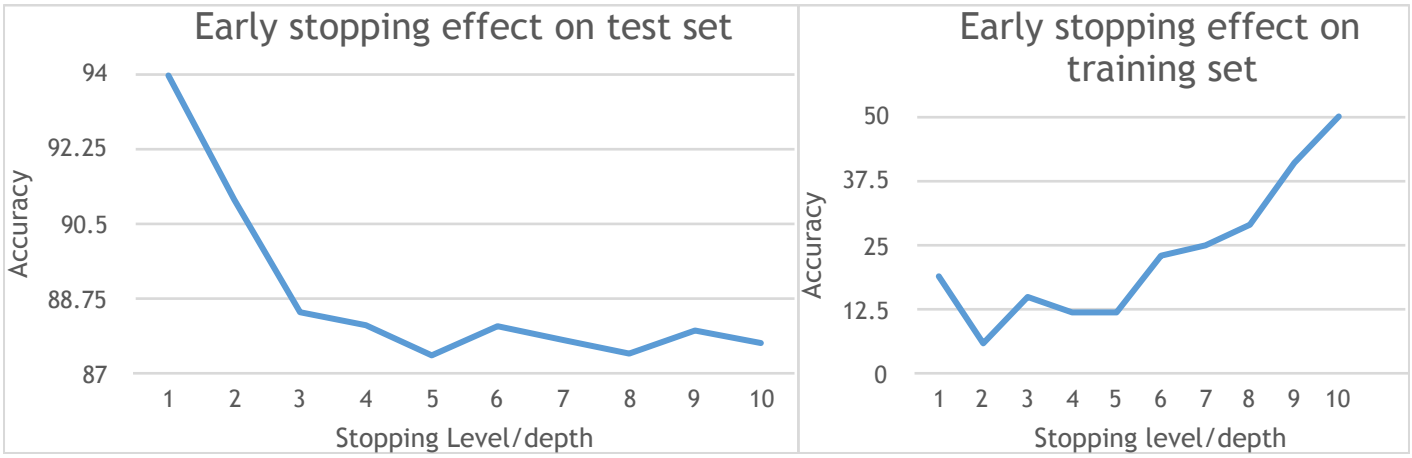


For ID3 on Training set- 99.240005% was the average accuracy.
For ID3 on Test set- 87.735374% was the average accuracy.

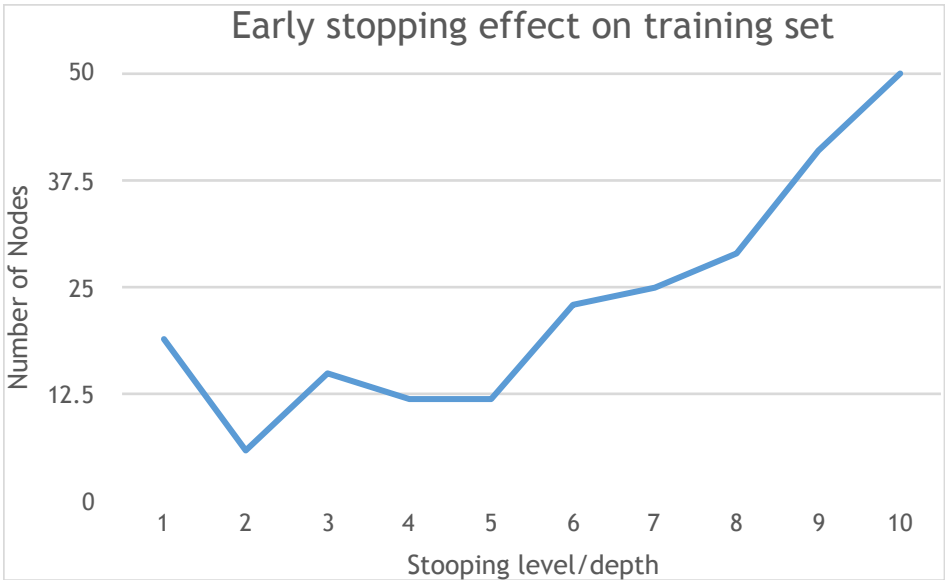
Early Stopping

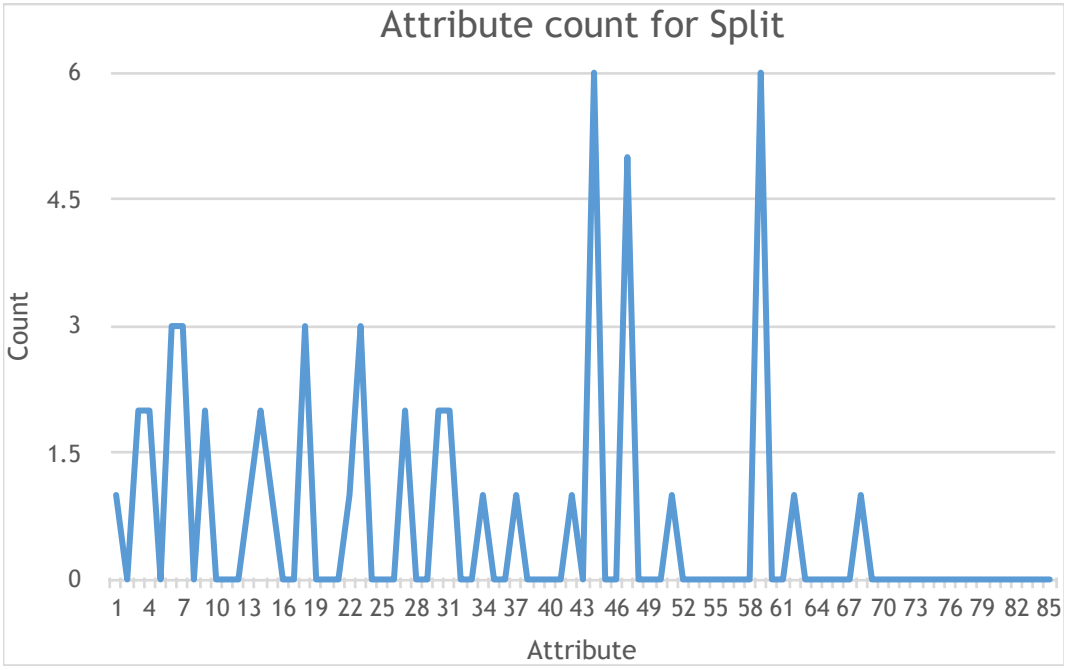
We then studied the effect of early stopping. We did this be using a maximum depth of the tree. However it is an overhead to generate another tree with less depth, so we used the same tree generated by ID3, instead we changed our classifier which now used a static maxLevel, and would not check the nodes after the maxLevel (as would be the case if we created a new tree, this saves space and time.)

Effects of Early Stopping



As it can be seen from the above graph, that on increasing the levels, the accuracy of the Test Set decrease, while that of the Training Set increases.. It means that the tree is now **Overfitting** the data, as the levels increase. Early stopping prevents the data from being overfitted. On Increasing the level, the number of nodes increase.

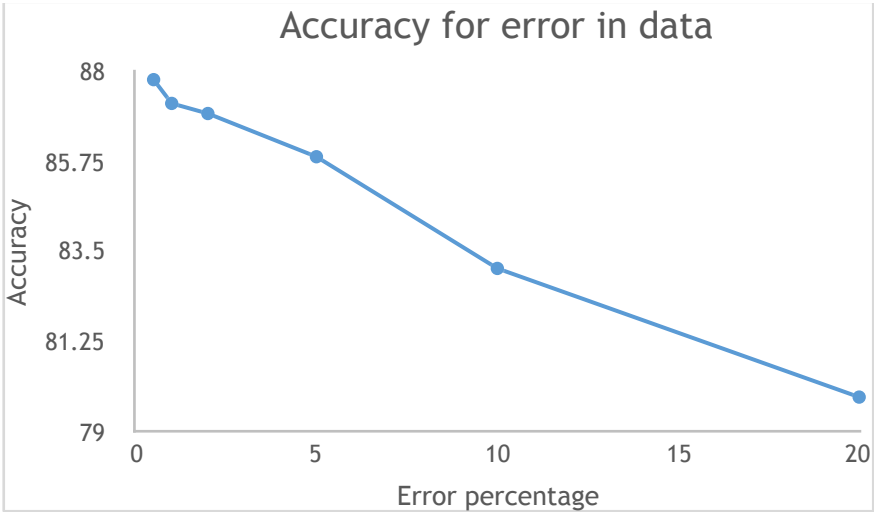




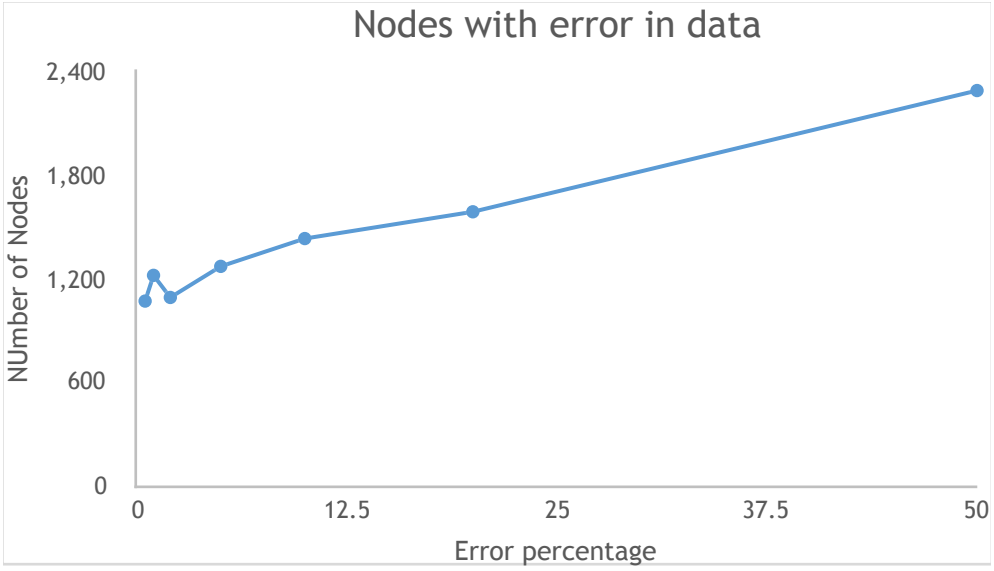
The graph for attribute count is shown, and clearly some attribute are preferred over others, this is because Information Gain has a bias for those attributes which have more number of possible values.

Effect of Addition of Noise

Noise was randomly added to the **training set.** This was done by randomly changing the class label of the training set.



It can be seen that as noise is added to the training set, the accuracy of the tree decreases. This is because we are using incorrect labels to generate the tree. Hence the tree wrongly classifies the test set.

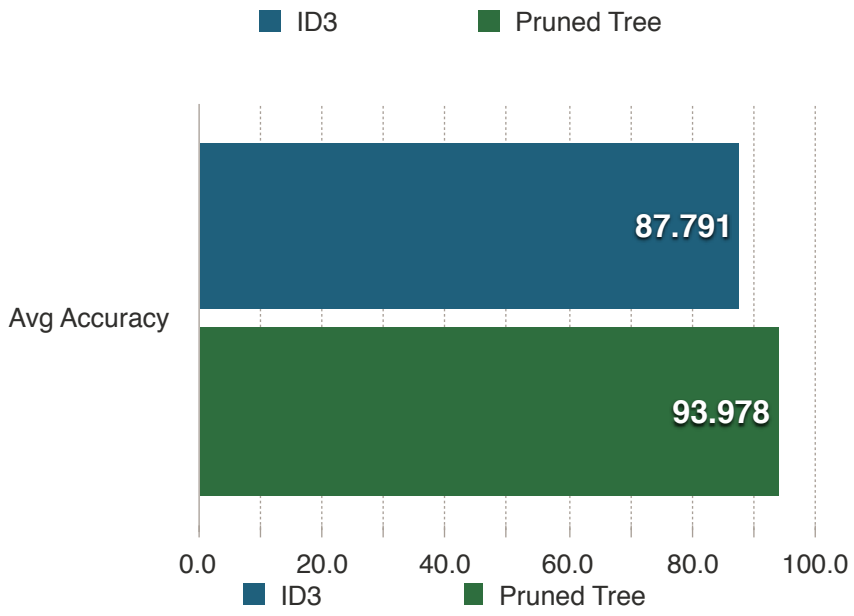


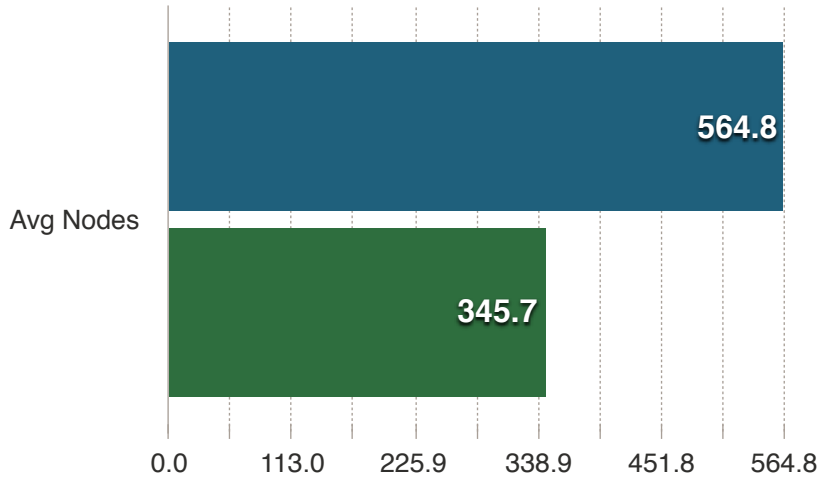
Effect of Pruning the Tree.

We used a bottom up approach to prune the tree, as it seemed most logical and efficient method to prune the tree. We have used the entire test data as validation set.

```
function Prune(node){
     Prune(childNode) // prune all the child nodes
     Flag current Node // don't check the subtree for
classification
     getNewAccuracy()
     if(newAccuracy is better){
          remove the subTree at node
     }
}
```

Instead of deleting a node, and then checking for improvement, a flag is used. If a flag is set, then the classifier ignores the subTree rooted at the node. (Instead of removing and then adding again in case there is no betterment on removal).

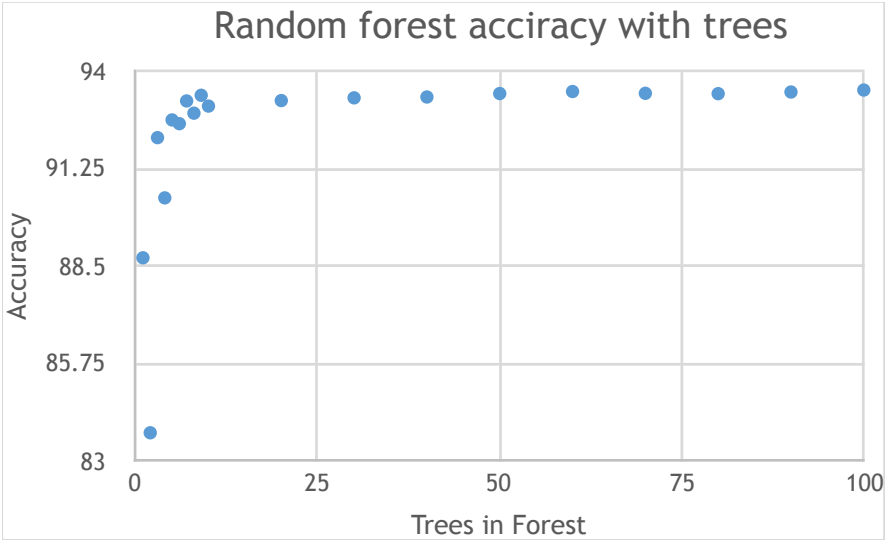




After Pruning the tree the accuracy of the pruned tree on test data increases. Since it was the criteria used to prune the tree. The accuracy increases because we are reducing the overfitting of the data.

The number of nodes decreased after pruning the tree, since some nodes were removed.

Random Forrest



A random forrest was generated using feature bagging, D = 9 was used. This feature bagging was used to generate a multiple trees, hence creating a forest. Each tree then classified the data, and the majority function was used to then classify the data.

As the number of trees increase, the classification accuracy increased, then became constant with respect to the number of trees.

Conclusion

Decision Trees were studied, and various experiments were conducted on them.

References

- Image:https://blog.bigml.com/2013/04/19/a-new-way-to-visualize-decision-trees/
- https://en.wikipedia.org/wiki/Decision_tree