GROUP 4: MSAI 349 Machine Learning Assignment HW#1

1. Did you alter the Node data structure? If so, how and why?

Yes, we altered the node data structure by adding three elements:

- Information Gain (float) to track and store the information gain,
- Label Name (string) for easier understanding, and
- Leaf Node flag (boolean) to indicate whether the node is a leaf or a parent node.

These modifications were made to enhance the clarity and efficiency of the decision tree process.

2. How did you handle missing attributes, and why did you choose this strategy?

Initially, we tried dropping rows with missing values, but this resulted in losing 50% of the data, which was not ideal. Instead, we used mode-based imputation i.e. selecting categorical values based on the maximum frequency of the column, as the data consisted of categorical values. This allowed us to retain more data while accurately handling the missing values.

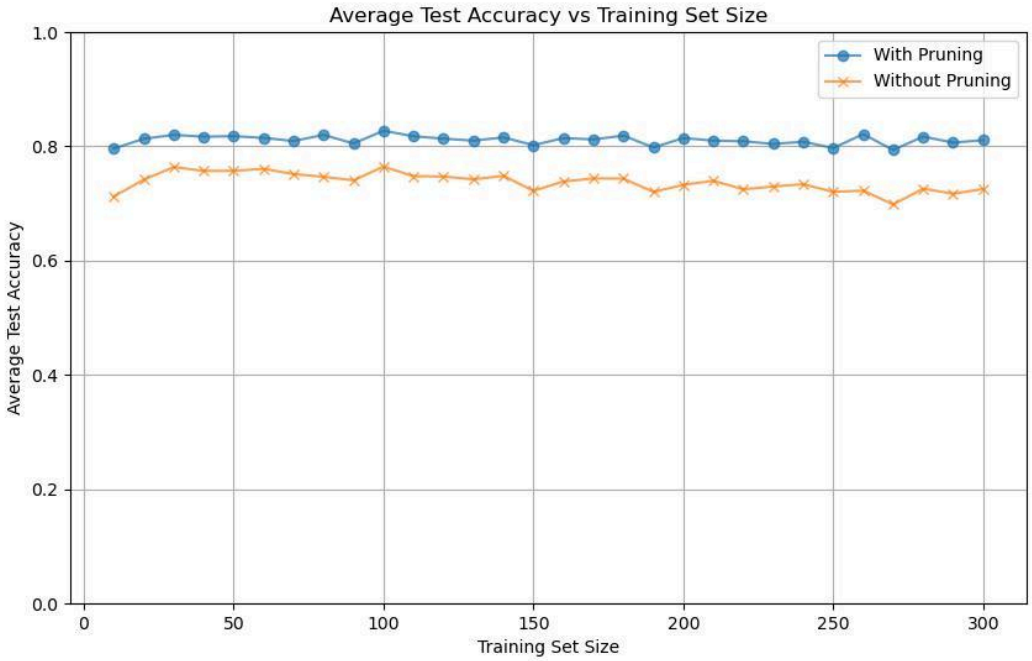
3. How did you perform pruning, and why did you choose this strategy?

We experimented with two pruning strategies: **Critical Value Pruning** and **Reduced Error Pruning**. We ultimately selected Reduced Error Pruning as it increased the accuracy by 3-4% when tested on the cars dataset.

Reduced Error Pruning is a post-pruning process where the individual nodes are replaced by majority class nodes of that branch and evaluated to calculate the accuracy from validation dataset, based on which the individual node is replaced with pruned node.

Critical Value Pruning is a pre-pruning process which restricts the building of trees i.e. adding nodes whose information gain is below a certain value.

4.
a) What is the general trend of both lines as training set size increases, and why does this make sense?
b) How does the advantage of pruning change as the dataset size increases? Does this make sense, and why or why not?



a) As the training set increases it can be observed that the average test accuracy increases. This is logical because pruning eliminates insignificant information that could cause overfitting, resulting in a more generalized model. By simplifying the model in this way, it performs more effectively on test data.

b) Since the dataset is relatively small, pruning has a significant effect on the decision tree model. This is understandable, as pruning in smaller datasets helps prevent the model from overfitting to individual training instances, which is crucial for enhancing its ability to generalize better.

5) Use your ID3 code to learn a decision tree on cars_train.data. Report accuracy on the cars_train.data, cars_valid.data and cars_test.data datasets. If your accuracy on cars_train.data is less than 100%, explain how this can happen. Prune the decision tree learned on cars_train.data using cars_valid.data. Run your pruned decision tree on cars_test.data, and explain the resulting accuracy on all three datasets.

Here is the accuracy report of our ID3 algorithm on cars dataset:

- Train Dataset: 100%
- Valid Accuracy: 71%
- Test Accuracy: 60%

The 100% accuracy is most likely a sign of overfitting. Consequently, the validation accuracy dropped significantly to approximately 71%.

Post pruning the trained decision tree with Reduced Error Pruning technique by using the validation dataset, we were able to observe a significant (20%) increase in the test accuracy scores. Therefore indicating that pruning has definitely improved the performance or generalization of the model.

```
(hitech) jnk789@Narasimhas-MacBook-Pro HW#1 % python test.py


Accuracy on Cars Test Dataset Before Pruning: 0.6
Accuracy on Cars Test Dataset After Pruning: 0.8
```

Dataset Observations:

- Training data accuracy: 1.0 (before pruning)
- Validation data accuracy: 0.7142 (before pruning)
- Test accuracy before pruning: 0.6
- Test accuracy after pruning: 0.8

6) Use your ID3 code to construct a Random Forest classifier using the candy.data dataset. You can construct any number of random trees using methods of your choosing. Justify your design choices and compare results to a single decision tree constructed using the ID3 algorithm.

We implemented a random forest consisting of 10 decision trees. For each tree, we randomly selected a subset of features and constructed the rest of the tree using our custom ID3 algorithm. Increasing the number of trees beyond 10 did not lead to significant changes in test accuracy, as the results remained within the same range. In nearly all test runs, the random forest outperformed individual decision trees, likely due to its superior ability to generalize. Furthermore, pruning had a notable positive effect on performance; random forests built with pruned trees consistently outperformed those using unpruned decision trees.