MSAI 349-MACHINE LEARNING GROUP 7 - HW 1

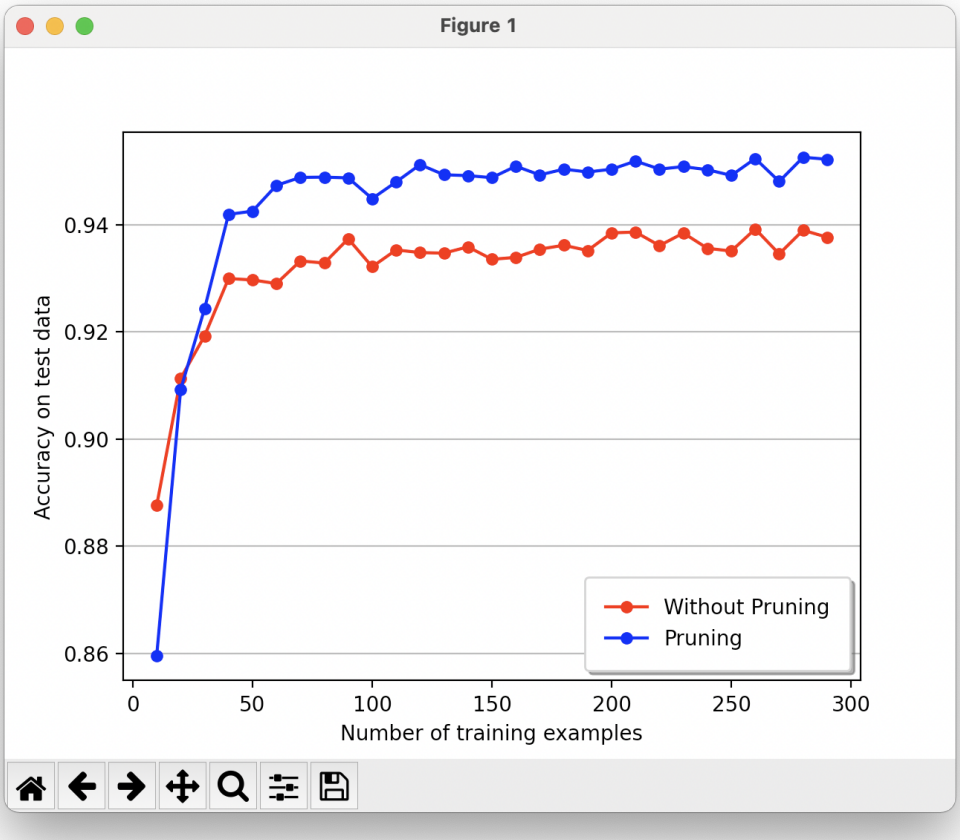
1. Did you alter the Node data structure? If so, how and why?
   Ans: We used label to track the classification of leaf nodes and added a_star to note the best attribute of the internal nodes.

2. How did you handle missing attributes, and why did you choose this strategy?
   Ans:We decided to consider the missing values as another attribute category(ex: others).

   Initially, we decided to take the mode of the missing attribute. But after analyzing the data, this approach makes the data skewed and degrades the accuracy. Hence we chose to consider it as a separate category. This approach is simple yet produced better accuracy.

3. How did you perform pruning, and why did you choose this strategy?
   Ans: We used **Reduced error pruning** because of it is computationally efficient and easy to implement.
   In our pruning algorithm, we traverse through the decision tree through the leaf nodes. We evaluate its accuracy with the effects of pruning the leaf; if the accuracy is equal or surpasses the original decision tree accuracy, pruning will take place.

4. The figure being referred to answer this question is as follows:



a) What is the general trend of both lines as training set size increases, and why does this make sense?
Ans: From the figure above, the accuracy for both lines steadily increases when the training size increases up to 100-150 examples. Beyond that point, increase in training size isn't affecting the accuracy as a whole since the curve seems to flatten out.

This makes sense because it is indicative of the fact that our model training has reached a saturation point in terms of training size having an effect on its accuracy. A training size of 100-150 examples seems enough for our model to learn all that it has to and achieve good accuracy.

b) How does the advantage of pruning change as the data set size increases? Does this make sense, and why or why not?

Ans: From the figure, the intercept of pruning is slightly less than that of without pruning for training size of around 30-40 examples. After that the accuracy for pruning increases and becomes better than the tree without pruning. This makes sense because pruning helps in avoiding

overfitting of the data that can cause the model to perform poorly against the validation/test dataset even if it performs well against the training dataset.

We have used the Reduced Error Pruning method. However as we can see the accuracy difference for both cases isn't very staggering thereby indicating that our original model (without pruning) is good and there isn't a lot of overfitting happening. Once the saturation point hits around 100-150 examples the curve seems to have flattened out for both cases thereby indicating the model has learnt all that it has to and achieve good accuracy.

5. Our Random Forest classifier design details are as below:

We decided to have **350 random ID3 decision trees** in our ensemble of decision trees.
Algorithm - We employed **bootstrap aggregating (or bagging)** to generate our training sub-dataset from candy.data.
Train-test split- For every sub-dataset, we randomly select **80% from the original dataset** to train.
Method - We will subsequently perform ID3 training on the 350 generated sub-dataset.
For every evaluation, we will run the evaluation on all 350 instances of our ID3 decision trees and use the majority vote to decide the final answer.
Bootstrap aggregating is used because it is simple and increases model performance by reducing variance of the model without increasing bias. With the use of majority rules, we reduce the risk of results being very sensitive to noise or bad data in our training dataset.