INTRODUCTION TO MACHINE LEARNING

DECISION TREES REPORT

IMPERIAL COLLEGE LONDON

Authors:
Sidharth Jain, Aadhithya Ashok Naarayan, Ibrahim Ibrahim

Date:
1st November, 2022

1 Visualization of the Decision Tree

These figures show the visualization of our decision tree trained on an entirely clean dataset.

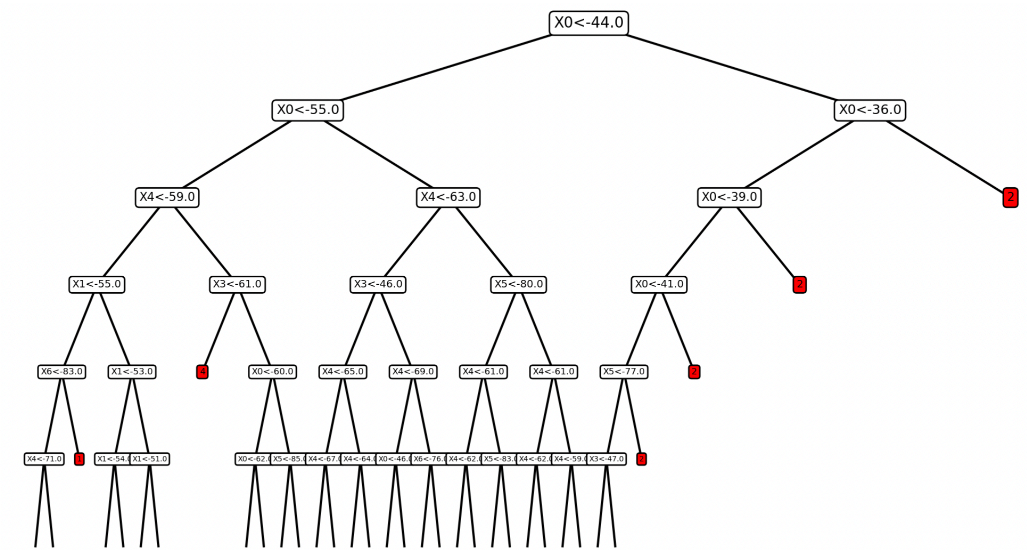

Figure 1: Decision tree trained on clean data set ; depth 5

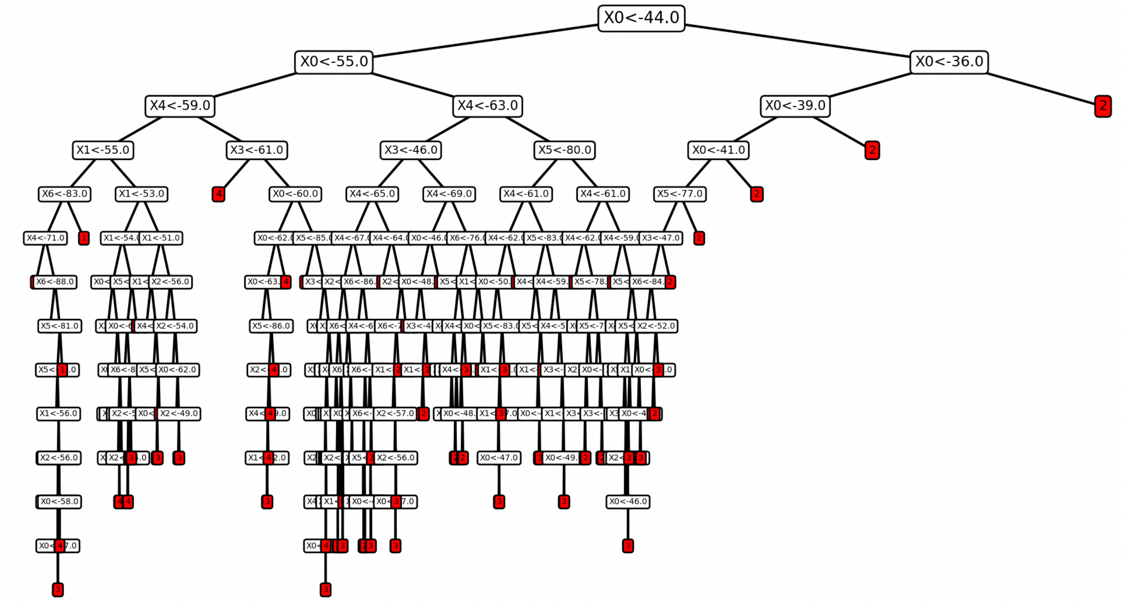

Figure 2: Decision tree trained on clean data set ; with maximum depth 13

2 Method of Implementation

2.1 Decision Tree Algorithm

We defined 5 main functions to implement the decision tree for the given dataset. These functions split the dataset into left and right datasets by maximizing its gain value through the concept of information gain.

Our implementation worked as follows :

- We defined a function to compute the entropy value of our data, denoted by the formula: $H(dataset) = -\sum_{k=1}^{k=K} p(k) * log_2(p_k)$.

  Here, K is the total number of labels in our data and p_k denotes the proportion of label k in our data.

- Then, we calculate the gain for all attributes and values to split the dataset into 2 parts. We compare the information gain for each attribute and pick the attribute and value with the highest gain.

- We sort the data in ascending order with respect to the value of the attribute. Then, we split the data at every unique value of the attribute and choose the split that maximizes the information gain denoted by the formula: $Gain(S_{all}, S_{right}, S_{left}) = H(S_{all}) - Remainder(S_{left}, S_{right})$.
  Here, $Remainder$ is the sum of the information gain of the left and right datasets.

- At the end, we split the data recursively by calculating the highest information gain in each created dataset to generate the decision tree.

2.2 Cross Validation using 10-folds

The generated dataset was split into 10 equal size folds. One fold was set aside as the validation data and a decision tree was trained on the other 9 using the algorithm mentioned in 2.1.

After this, the validation data is evaluated using the trained tree and confusion matrix. From the confusion matrix, we can calculate:

- Accuracy: $(correctly\ classified\ examples)/(total\ examples)$

- Recall: $(correctly\ classified\ positive\ examples)/(total\ positive\ examples)$

- Precision: $(correctly\ classified\ positive\ examples)/(total\ predicted\ positive\ examples)$

- F-1 Score: $2 * (Precision * Recall)/(Precision + Recall)$

For each fold, the confusion matrix and the specific measures are calculated.

2.3 Pruning Algorithm

We used a depth first search approach to find the nodes in the tree which need to be pruned. Pruning prevents our decision tree from over-fitting to our dataset.

Our implementation worked as follows :

- We find a parent node of two unchecked leaf nodes. This node is converted to a leaf with a label determined by the majority label of samples in the subset of the training set used to build that node's sub-tree structure.

- A market is put on the node we just pruned to prevent that node from being pruned again in the next iteration.

- The classification rate of the unpruned and pruned tree are compared and the best tree is chosen. Pruned tree reduces the complexity.

- This process is repeated to find the next node that can be pruned. We carry forward any prune that performs better than the previous iteration of the tree.

We then use the 10-fold cross validation with pruning technique to evaluate the effect of pruning on accuracy of the decision trees.

3 Evaluation

3.1 Evaluation - Clean Dataset

3.1.1 Cross Validation Classification Metrics

	Room 1 Predicted	**Room 2 Predicted**	**Room 3 Predicted**	**Room 4 Predicted**
Room 1 Actual	49.6	0.0	0.4	0.0
Room 2 Actual	0.0	47.2	2.8	0.0
Room 3 Actual	0.8	2.4	46.1	0.7
Room 4 Actual	0.3	0.0	0.5	49.2

Table 1: Average confusion matrix for clean data

3.1.2 Other Metrics

Metrics	Room 1	Room 2	Room 3	Room 4	Average
Recall	0.993	0.944	0.922	0.985	
Precision	0.978	0.951	0.927	0.986	
F1	0.985	0.947	0.923	0.985	
Accuracy					0.961

Table 2: Metrics for clean data

3.1.3 Result Analysis

Table 1 shows the cross validation results for clean data. The averages of the performance metrics (recall, precision, F1, accuracy) are shown in Table 2. The results are accurate for both, the overall score and the label specific measures.

From table 1, we see that only room 2 and room 3 are confused for each other on more than one instance on average.

3.2 Evaluation - Noisy Dataset

3.2.1 Cross Validation Classification Metrics

	Room 1 Predicted	Room 2 Predicted	Room 3 Predicted	Room 4 Predicted
Room 1 Actual	38.8	3.0	3.2	4.0
Room 2 Actual	2.3	40.3	4.3	2.8
Room 3 Actual	2.7	4.1	41.6	3.1
Room 4 Actual	4.2	1.6	3.3	40.7

Table 3: Average confusion matrix for noisy data

3.2.2 Other Metrics

Metrics	**Room 1**	**Room 2**	**Room 3**	**Room 4**	**Average**
Recall	0.785	0.810	0.807	0.809	
Precision	0.788	0.816	0.795	0.809	
F1	0.785	0.811	0.799	0.808	
Accuracy					0.803

Table 4: Metrics for noisy data

3.2.3 Result Analysis

Table 3 shows the cross validation results for clean data. The averages of the performance metrics (recall, precision, F1, accuracy) are shown in Table 4. The results are much more confused compared to those depicted in the clean dataset.

From table 3 and 4 we see that there is a general performance decrease in the prediction of all rooms.

3.3 Dataset Differences (Clean vs. Noisy)

As depicted by the tables above, the performance results of our decision tree on the noisy data is worse than on clean data. This is consistent over all rooms and performance measures. This is proved by the accuracy % of clean data, which is 16% better than the accuracy % of noise data.

Noise may cause samples with different labels to have similar attribute values. This causes contradictions in labels of similar samples and creates further splits in the data set to classify the training data. Since the noise is random, it causes problems in generalizing the classification to unseen data. The additional splitting of our data due to noise in the training data causes an increased rate of miss-classification on unseen data from the same distribution.

4 Pruning Evaluation

4.1 Pruned Clean Data Metrics

	Room 1 Predicted	**Room 2 Predicted**	**Room 3 Predicted**	**Room 4 Predicted**
Room 1 Actual	49.8	0.0	0.1	0.1
Room 2 Actual	0.0	47.1	2.9	0.0
Room 3 Actual	0.7	1.1	47.7	0.5
Room 4 Actual	0.4	0.0	1.0	48.6

Table 5: Average confusion matrix for pruned clean data

Metrics	**Room 1**	**Room 2**	**Room 3**	**Room 4**	**Average**
Recall	0.996	0.943	0.950	0.968	
Precision	0.978	0.971	0.918	0.993	
F1	0.987	0.956	0.933	0.980	
Accuracy					0.965

Table 6: Metrics for pruned clean data

4.2 Pruned Noisy Data Metrics

	Room 1 Predicted	**Room 2 Predicted**	**Room 3 Predicted**	**Room 4 Predicted**
Room 1 Actual	44.2	1.2	1.3	2.3
Room 2 Actual	1.9	43.7	3.0	1.0
Room 3 Actual	2.2	3.3	44.1	1.8
Room 4 Actual	2.1	1.3	2.5	43.8

Table 7: Average confusion matrix for pruned clean data

Metrics	**Room 1**	**Room 2**	**Room 3**	**Room 4**	**Average**
Recall	0.899	0.871	0.863	0.885	
Precision	0.875	0.891	0.867	0.893	
F1	0.887	0.880	0.864	0.888	
Accuracy					0.881

Table 8: Metrics for pruned clean data

4.3 Result Analysis after Pruning

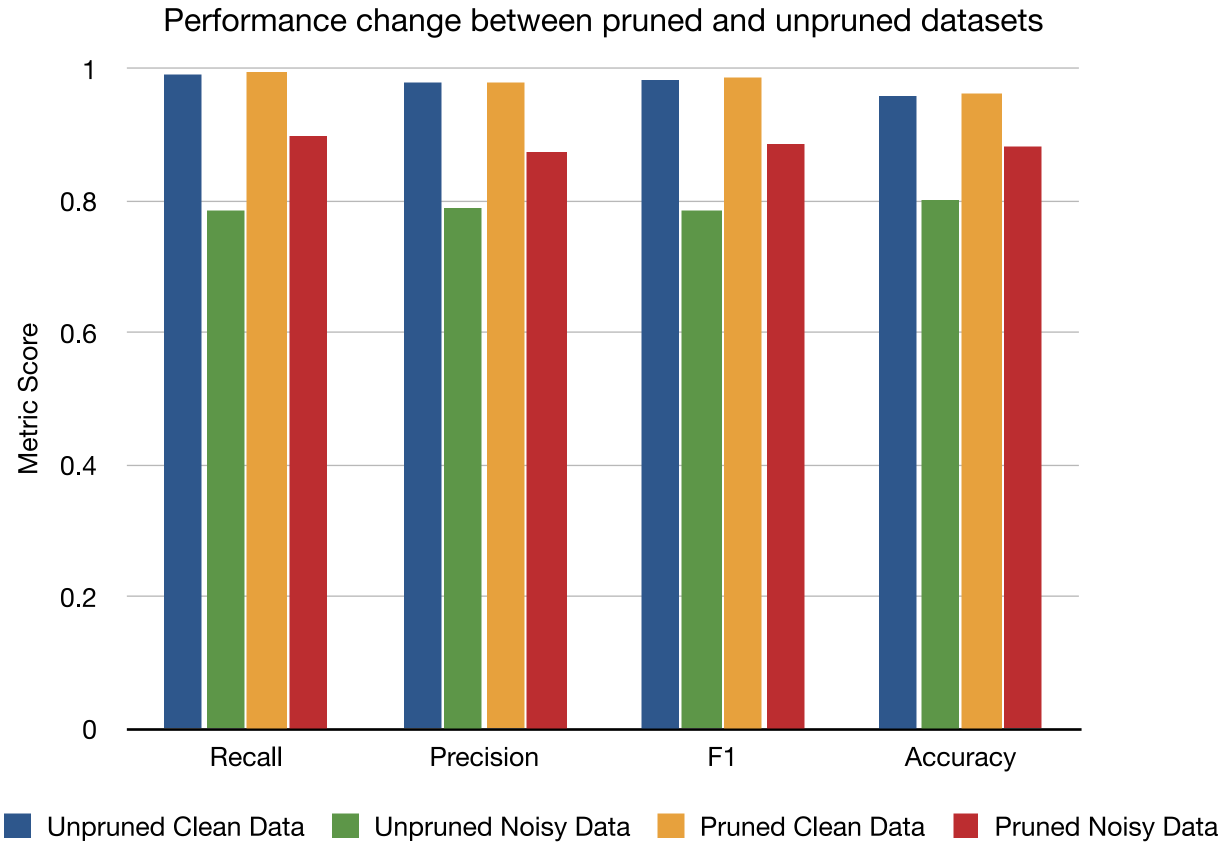

Figure 3: Influence of pruning on clean and noisy data

The influence of pruning our decision tree is different when applied to clean and noisy datasets. Pruning has a negligible effect on the clean data with an average accuracy increase of just 0.01% on the test data whereas a significant effect on the noisy data with an average accuracy increase of around 8%. The significant increase of the average accuracy on noisy datasets is because of the noisy examples (the training data distribution might be different from the test data). Hence, the decision tree has a higher chance of over-fitting the training data.

This significant effect of pruning is depicted in Figure 3 above.

4.4 Depth Analysis

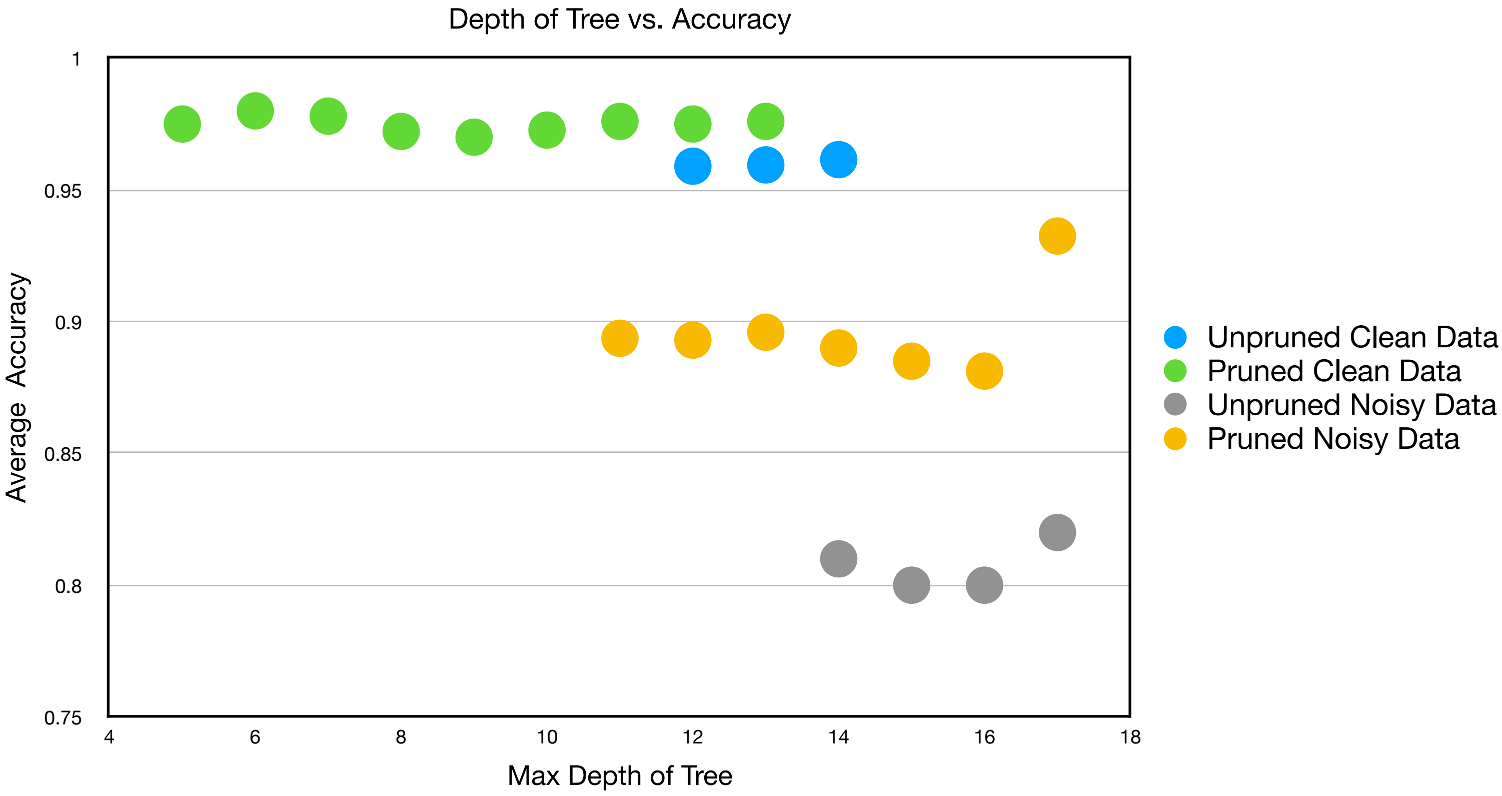

Figure 4: Depth vs. Accuracy of Different Decision Trees

Figure 4 shows the depth vs. average accuracy of all generated decision trees of the same depth from the k-fold evaluation algorithm. As depicted in the figure, pruning the datasets, irrespective of if they are clean or noisy, improves the accuracy. In conclusion, for any real application using decision trees where noise is inevitable, pruning is important. This arises from the fact that the accuracy is significantly better when reducing the depth of trees trained on noisy data.