Introduction to Machine Learning
Coursework 1 Report

He Liu, Kejian Shi, Qianyi Li, Shitian Jin

November 3, 2021

In this project we are implementing the decision tree classifier using numpy and matplotlib only. This report will be represented into 3 parts, which are:

- 1. Visualisation of the tree
- 2. To-be-marked Questions: Step3: Evaluation
- 3. To-be-marked Questions: Step4: Pruning

1 Output of the tree visualisation function

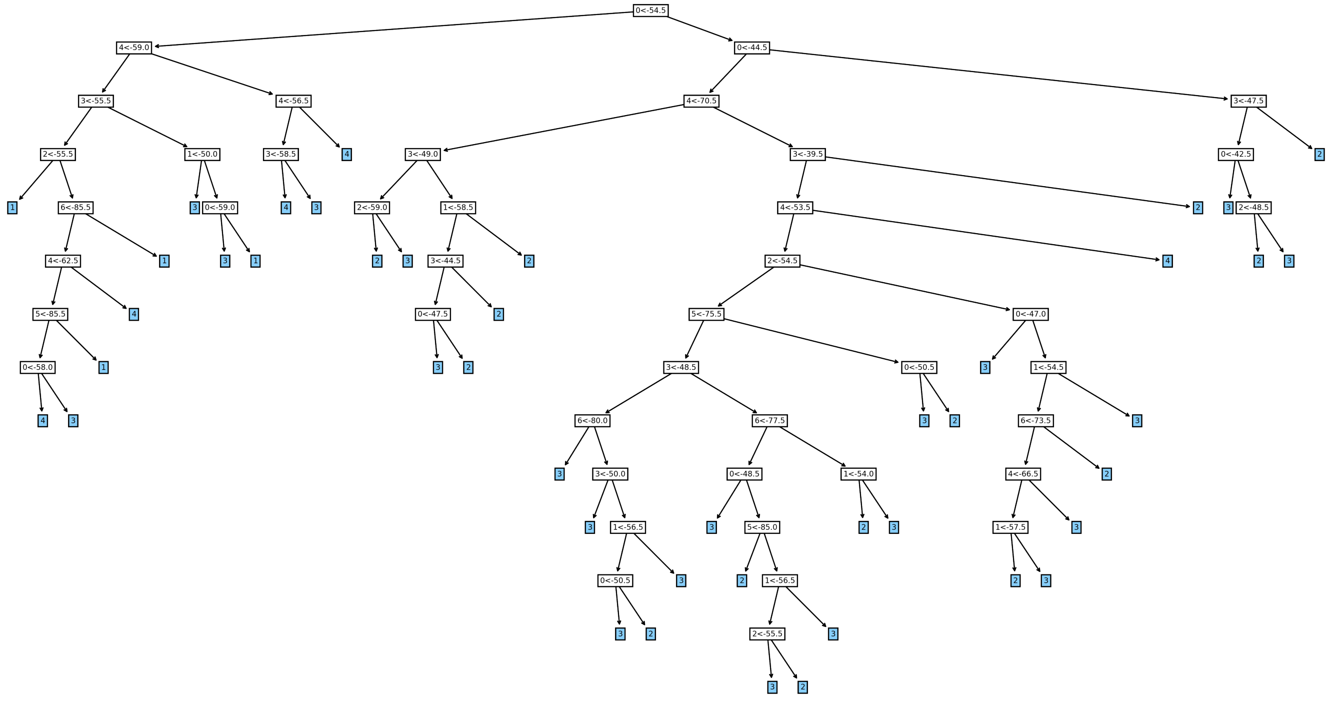

Figure 1: **The tree visualisation generated by matplotlib.**

2 Step 3 - Evaluation

In this part we evaluated the decision tree using 10 folds cross validation on both of the clean and noisy datasets. The evaluation metrics and analysis are displayed below, for convenient we address class 1,2,3,4 to rooms 1,2,3,4:

2.1 Cross validation classification metrics

- Confusion matrix

Predicted **Actual**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 1	49.3	0.0	0.4	0.3
Class 2	0.0	48.4	1.6	0.0
Class 3	0.3	1.7	47.9	0.1
Class 4	0.4	0.0	0.3	49.3

Clean Dataset

Predicted **Actual**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 1	36.9	4.3	3.5	4.3
Class 2	3.0	40.1	4.1	2.5
Class 3	3.0	3.9	41.0	3.6
Class 4	3.3	3.2	4.2	39.1

Noisy Dataset

Table 1: **The confusion matrices for 'Clean data' and 'Noisy data' in Step 3 - Evaluation**

- The accuracy per class

	Class 1	**Class 2**	**Class 3**	**Class 4**
Clean Dataset	0.993	0.9835	0.978	0.9945
Noisy Dataset	0.893	0.895	0.8885	0.8945

Table 2: **The accuracy per class in Step 3 - Evaluation**

- The recall and precision per class

	Class 1	**Class 2**	**Class 3**	**Class 4**
Clean Dataset	0.98522877	0.96841919	0.95955980	0.98617087
Noisy Dataset	0.75351678	0.80426339	0.79420843	0.78531581

Table 3: **The recall per class in Step 3 - Evaluation**

	Class 1	**Class 2**	**Class 3**	**Class 4**
Clean Dataset	0.98624147	0.96564716	0.95616882	0.99182311
Noisy Dataset	0.80036021	0.77753642	0.77968647	0.79048354

Table 4: **The precision per class in Step 3 - Evaluation**

- The F1-measures

	Class 1	**Class 2**	**Class 3**	**Class 4**
Clean Dataset	0.98556396	0.96679811	0.95748980	0.98889247
Noisy Dataset	0.77551450	0.78970809	0.78573464	0.78667981

Table 5: **The F1-measures per class in Step 3 - Evaluation**

2.2 Result analysis

From the above evaluation metrics one can conclude that:

For **clean data**: **Class 4**(which is room 4) has the highest accuracy(0.9945) and F1 score(0.989), while the room 3 has the lowest accuracy(0.978) and F1 score(0.957), which means signal data from **class 3 confuses** the classifier most. **For noisy data**: It is observed that the accuracy for **class 2** is the highest(0.895) while the **class 3** is associated with the lowest accuracy(0.8885). However, it is noted that the **class 2**(f1 = 0.7898) is the most distinguishable and the **class 1** causes the most confusion(f1 = 0.7755) because it has the smallest F1-score.

2.3 Dataset differences

A significant degradation of the performance is observed when using **noisy dataset** as compare to **clean dataset**. Specifically, the performance dropped by 10-20 % when tested for all fours classes. This is due to the fact that in the noisy dataset there are some **outliers/noise** and the decision tree tries to fit every data point including the noise, which results in a over-fitted tree with reduced generalisation. Also, the data distribution of noisy dataset is slightly worse than the clean dataset, which may also cause the classifier fits a class worse than the other classes.

3 Step 4 - Pruning (and evaluation again)

In this section we are using the nested cross validation(option 2) to evaluate our decision tree and do pruning.
We divided one dataset into 10 folds and for each episode we choose one fold as a test set. In that episode we use the rest 9 folds as training and validation sets and do the cross validation. Therefore, for each of 10 episode we generates 9 trees and we do pruning on the tree. The evaluation metrics and analysis after pruning are displayed below:

3.1 Cross validation classification metrics after pruning

- Confusion matrix

Predicted **Actual**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 1	49.6	0.0	0.2	0.2
Class 2	0.0	47.9	2.1	0.0
Class 3	0.1	1.2	48.5	0.2
Class 4	0.4	0.0	0.4	49.2

Clean Dataset

Predicted **Actual**	**Class 1**	**Class 2**	**Class 3**	**Class 4**
Class 1	40.1	3.0	2.1	3.8
Class 2	3.2	41.9	2.6	2.0
Class 3	2.1	2.6	44.5	2.3
Class 4	3.2	2.2	3.1	41.3

Noisy Dataset

Table 6: **The confusion matrices of clean and dirty dataset in Step 4 - Pruning**

- The accuracy per class

	Class 1	**Class 2**	**Class 3**	**Class 4**
Clean Dataset	0.9955	0.9835	0.979	0.994
Noisy Dataset	0.913	0.922	0.926	0.917

Table 7: **The accuracy per class in Step 4 - Pruning**

- The recall and precision per class

	Class 1	**Class 2**	**Class 3**	**Class 4**
Clean Dataset	0.99145925	0.95931288	0.96927651	0.98408754
Noisy Dataset	0.81804413	0.83980374	0.86208070	0.82931121

Table 8: **The recall per class in Step 4 - Pruning**

	Class 1	**Class 2**	**Class 3**	**Class 4**
Clean Dataset	0.99095458	0.97595065	0.94730617	0.99249113
Noisy Dataset	0.82719188	0.84244238	0.85157969	0.83720018

Table 9: **The precision per class in Step 4 - Pruning**

- The F1-measures

	Class 1	**Class 2**	**Class 3**	**Class 4**
Clean Dataset	0.99112272	0.96735285	0.95782628	0.98807954
Noisy Dataset	0.82093856	0.84024389	0.85589562	0.83253703

Table 10: **The F1-measures per class in Step 4 - Pruning**

3.2 Result analysis after pruning

From Table 6-10, on **clean data** the evaluations look similar to Table 1-5(e.g. smaller than 0.005 difference on accuracy). While for the **noisy dataset** after pruning the evaluation metrics are significantly higher(on average + 0.27 on accuracy, + 0.05 on F1 score, etc.). This is because if the data was clean, the tree was trained well so even after pruning, the performance didn't change much. However, for the **noisy data**, because of noise it is more possible for the tree to get overfitted. By pruning, We can fix this and so the performance will improve significantly.

3.3 Depth analysis

Depth per tree	**Average**
12 15 14 13 14 13 12 14 11 12	13.0

Clean Dataset

Depth per tree	**Average**
19 18 19 20 18 23 19 17 17 19	18.9

Noisy Dataset

Table 11: **The depth result in Step 3 without pruning**

Depth per best tree	**Average**
12 14 11 14 16 11 11 14 11 11	12.5

Clean Dataset

Depth per best tree	**Average**
16 18 18 18 16 22 18 16 18 19	17.9

Noisy Dataset

Table 12: **The depth result in Step 4 after pruning**

It is shown above that for **clean data** the average depth decreased from 13 to 12.5 after pruning while the accuracy kept almost the same. For **noisy data**, the average depth decreased from 18.9 to 17.9 while the accuracy on average increased from 0.893 to 0.920 after pruning. So in general, for the same dataset, decreasing the maximum depth by pruning can increase the accuracy, but not guaranteed. While in comparison an extremely deep tree will not perform well.(As discussed above, it depends on the performance of the data distribution and overfit or not)