Introduction to Machine Learning
Coursework 1 Report

He Liu, Kejian Shi, Qianyi Li, Shitian Jin

November 4, 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

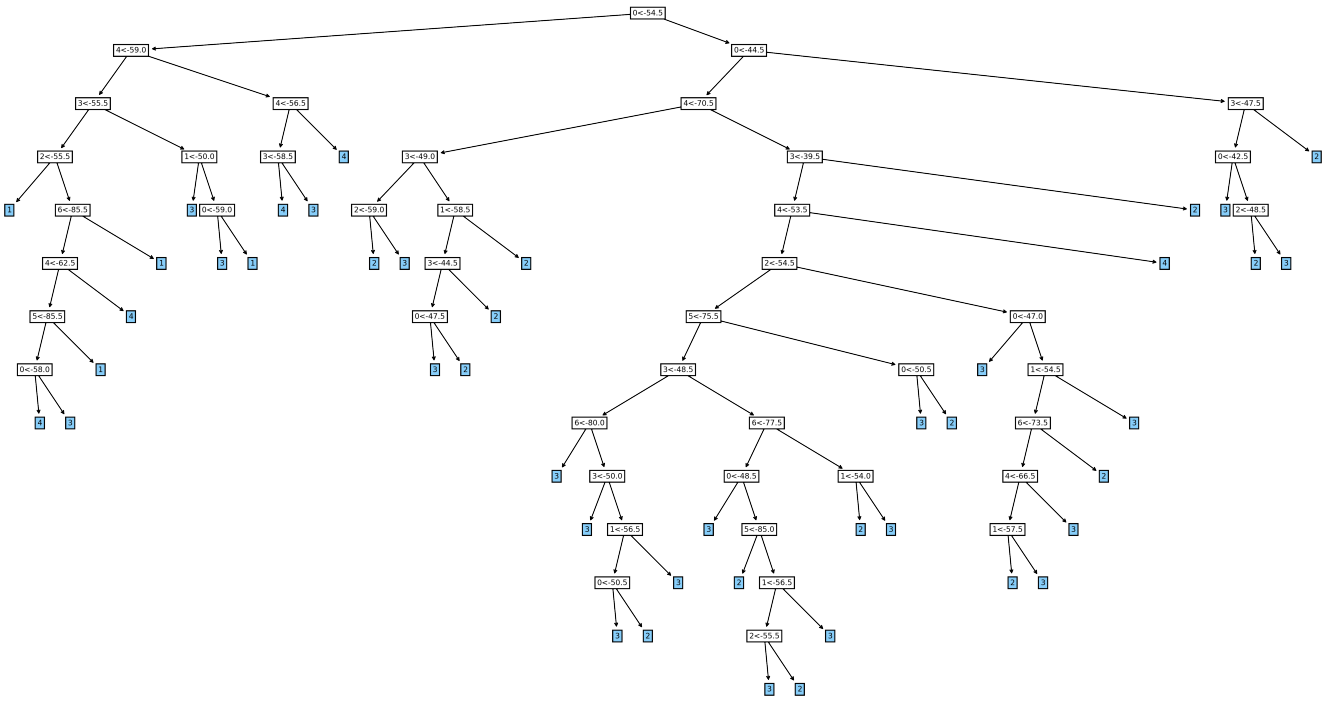

Figure 1: **The tree visualisation generated by matplotlib.**

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 respectively:

2.1 Cross validation classification metrics

- Confusion matrix

Actual \ Predicted	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

Actual \ Predicted	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

Clean Dataset	97.45%
Noisy Dataset	78.55%

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 and 1** had higher recall, precision and F1 score in comparison with **Class 2 and 3**. From Table 1 it can be seen **Class 2 and 3** confuses with each other, and confuses the classifier the most.

For noisy data: **Class 2,3,4** has relatively higher recall, precision and F1 scores while the **Class 1** is associated with the lowest score. **Class 2**(F1 = 0.7897) is the most distinguishable and the **Class 1**(F1 = 0.7755) causes the most confusion because it has the smallest F1-score so more confusion.

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 % (97.45% comparing to 78.55%). This is due to the fact that in the noisy dataset there are some **outliers/noise** and the decision tree tries to fit also the noise, which results in a over-fitted tree with reduced generalisation. Note that Class 2, 3 are confused in clean data but did well in noisy data. This might because more emitters are around Room 2 and 3 while the noise has more effects on further rooms.

Unmarked: More explanation about the last line:
Note a fact that the signal of WiFi will get weaker for further distances.

Because the emitters are situated around Room 2 and 3, there is even on emitter an the wall between these two rooms, these signals can be easily get confused. Relatively speaking, Room 1 and 4 which are further from the emitters can be easier distinguished. This can be justified in the visualized tree on clean data, where Class 1 and 4 was distinguished at shallower leaf nodes, while Class 2 and 3 takes more depth to distinguish. However, when considering noise, This may not be the case. Because the signal for Room 1 was suppose to be weaker and slightly separated, if there were some noisy data for Room 1 with strong signals, this could confuse the classifier so much and may lead to over-fit.

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

Actual \ Predicted	Class 1	Class 2	Class 3	Class 4
Class 1	49.8	0.0	0.2	0.0
Class 2	0.0	48.1	1.9	0.0
Class 3	0.2	1.4	48.3	0.1
Class 4	0.3	0.0	0.3	49.4

Clean Dataset

Actual \ Predicted	Class 1	Class 2	Class 3	Class 4
Class 1	41.8	2.4	1.9	2.9
Class 2	2.8	43	2.1	1.8
Class 3	2.1	2.7	44.7	2.0
Class 4	3.0	1.6	2.7	42.5

Noisy Dataset

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

- The accuracy per class

Clean Dataset	97.8%
Noisy Dataset	86.0%

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	0.99592003	0.96281065	0.96539752	0.98838911
Noisy Dataset	0.8544964	0.86261829	0.86592828	0.85364642

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

	Class 1	Class 2	Class 3	Class 4
Clean Dataset	0.99068281	0.97309447	0.95274396	0.998
Noisy Dataset	0.84413447	0.86531348	0.87071746	0.86310995

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

- The F1-measures

	Class 1	Class 2	Class 3	Class 4
Clean Dataset	0.99322061	0.96750432	0.95838048	0.99308653
Noisy Dataset	0.84749201	0.86364363	0.8672454	0.85790045

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(+ 0.35% on average accuracy). While for the **noisy dataset** after pruning the evaluation metrics are significantly higher(on average + 7.45% on accuracy, + 0.07 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 over-fitted. By pruning, We can fix this and so the performance will improve significantly.

3.3 Depth analysis

Maxumum depth per tree	Average
12 15 14 13 14 13 12 14 11 12	13.0

Clean Dataset

Maximum 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**

Maximum depth per best tree	Average
12 14 10 14 16 11 10 14 12 11	12.4

Clean Dataset

Maximum depth per best tree	Average
16 18 19 18 16 19 18 16 17 19	17.6

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 12.4 after pruning while the accuracy kept almost the same. For **noisy data**, the average depth decreased from 18.9 to 17.6 while the accuracy on average increased from 78.55% to 86.0% after pruning. So in general, for the same dataset, one may say decreasing the maximum depth can increase accuracy, but not guaranteed. While in comparison an extremely deep tree (over-fitted) will not perform well.(As discussed above, it depends on distribution of data and over-fitted or not)