COMP70050

Introduction to Machine Learning

October 2023

Imperial College London

Department of Computing

—*Coursework 1*—

Decision Trees

Team members:

Kyoya Higashino [redacted]

Jack Hau [redacted]

Fadi Zahar [redacted]

Konstantinos Mitsides [redacted]

Plan

Introduction

Decision trees, while intuitive, often overfit to training data, especially in the presence of noise. In this regard, pruning, which involves reducing the size of the tree by removing excessive branching to simplify the model, is frequently employed as a technique to remedy this overfitting. In this report, we examine the efficacy of pruning in the context of clean and noisy datasets. Prior to pruning, trees were evaluated using 10-fold cross-validation (detailed in Section 2 – Evaluation). Post-pruning, performance was assessed via nested 10-fold cross-validation, as discussed in Section 3 – Pruning (and Evaluation Again). A noteworthy mention is our use of **np.random.shuffle** with a seed of 0 for data shuffling before cross-validation. Given the datasets' balanced nature (~500 instances per class, totalling 2000 across 4 classes), this approach was deemed appropriate, negating the need for stratified shuffling, which is pivotal for imbalanced datasets to get similar representation of classes between folds.

Section 1 – Decision Tree Visualisation Function

The plot of the decision tree trained on the entire clean dataset is reported below. The tree has a depth of 14.

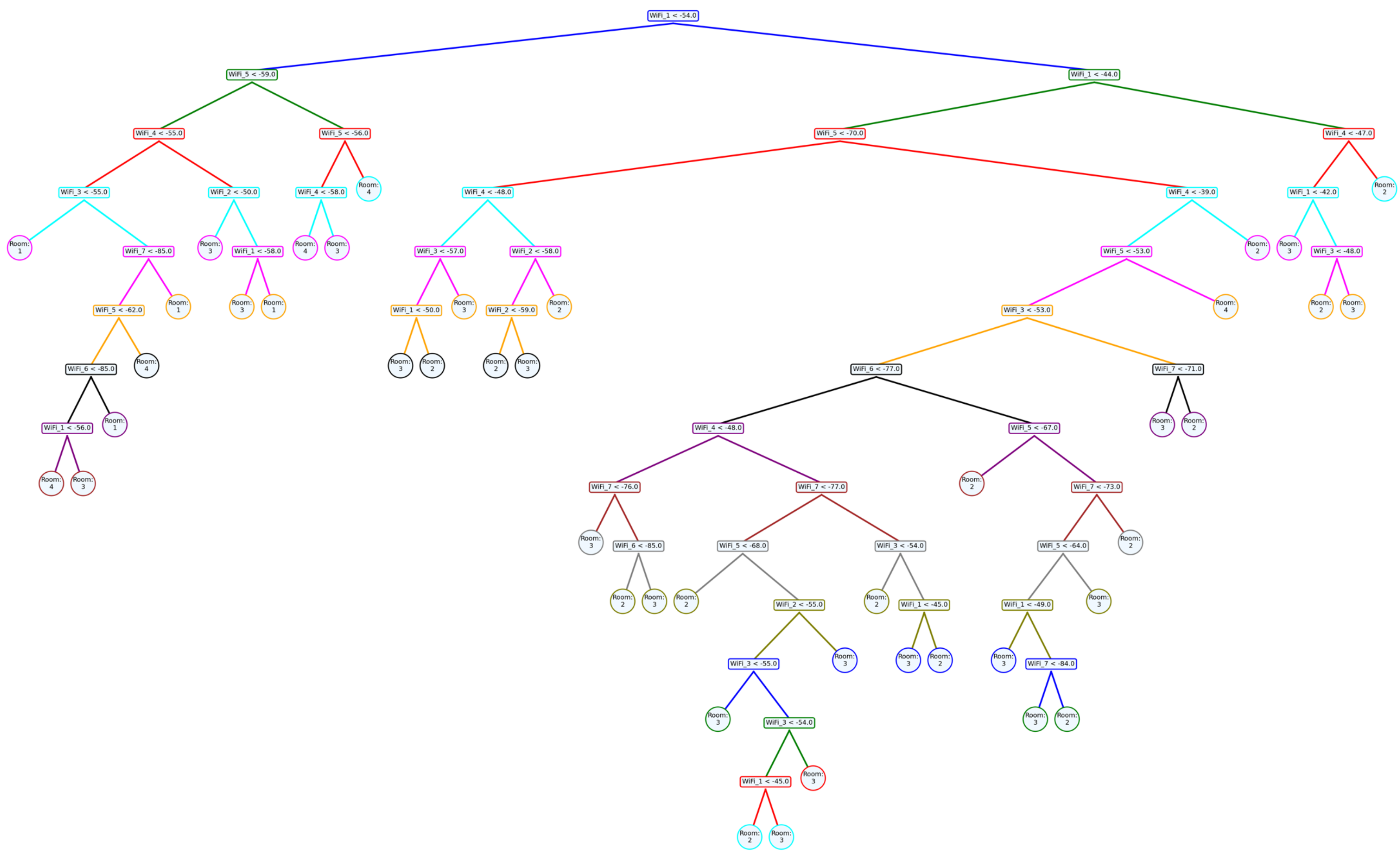

Figure 1: Plot of the decision tree trained on the entire clean dataset.

Section 2 – Evaluation

Cross-validation classification metrics (before pruning)

Table 1: Pre-Pruning Cross-Validation Classification Metrics

<table>
<tr><th></th><th>Clean Data</th><th>Noisy Data</th></tr>
<tr><td>Class Balance</td><td><table><tr><td>Room</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>Number of instances</td><td>500</td><td>500</td><td>500</td><td>500</td></tr></table></td><td><table><tr><td>Room</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>Number of instances</td><td>490</td><td>497</td><td>515</td><td>498</td></tr></table></td></tr>
<tr><td>Depth</td><td>11.8</td><td>18.1</td></tr>
<tr><td>Confusion Matrix</td><td><table><tr><td></td><td></td><td colspan="4">Predicted</td></tr><tr><td></td><td>Room</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td rowspan="4">Actual</td><td>1</td><td>49.60</td><td>0.</td><td>0.10</td><td>0.30</td></tr><tr><td>2</td><td>0.</td><td>47.60</td><td>2.40</td><td>0.</td></tr><tr><td>3</td><td>0.30</td><td>1.90</td><td>47.60</td><td>0.20</td></tr><tr><td>4</td><td>0.50</td><td>0.</td><td>0.20</td><td>49.30</td></tr></table></td><td><table><tr><td></td><td></td><td colspan="4">Predicted</td></tr><tr><td></td><td>Room</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td rowspan="4">Actual</td><td>1</td><td>38.50</td><td>3.</td><td>3.20</td><td>4.30</td></tr><tr><td>2</td><td>3.</td><td>40.30</td><td>4.</td><td>2.40</td></tr><tr><td>3</td><td>2.90</td><td>4.40</td><td>40.60</td><td>3.60</td></tr><tr><td>4</td><td>3.60</td><td>2.90</td><td>3.70</td><td>39.60</td></tr></table></td></tr>
<tr><td>Accuracy</td><td>97.1%</td><td>79.5%</td></tr>
<tr><td>Precision per Room (class)</td><td><table><tr><td>Room</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>Rate</td><td>98.4%</td><td>96.2%</td><td>94.7%</td><td>99.1%</td></tr></table></td><td><table><tr><td>Room</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>Rate</td><td>80.1%</td><td>79.3%</td><td>79.0%</td><td>79.9%</td></tr></table></td></tr>
<tr><td>Recall per Room</td><td><table><tr><td>Room</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>Rate</td><td>99.2%</td><td>95.3%</td><td>95.2%</td><td>98.6%</td></tr></table></td><td><table><tr><td>Room</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>Rate</td><td>78.3%</td><td>81.1%</td><td>79.1%</td><td>79.7%</td></tr></table></td></tr>
<tr><td>F1-Measures per Room</td><td><table><tr><td>Room</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>Rate</td><td>98.8%</td><td>95.7%</td><td>94.9%</td><td>98.8%</td></tr></table></td><td><table><tr><td>Room</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>Rate</td><td>78.8%</td><td>79.8%</td><td>78.8%</td><td>79.6%</td></tr></table></td></tr>
</table>

*Measures averaged over the $k = 10$ folds and rounded to 3 significant figures (except for the confusion matrix, rounded to 2 decimal places, and the class balance)

Result analysis

For the clean dataset, all rooms were accurately discerned with precision and recall rates predominantly above 95%. Rooms 1&4 exhibit metrics surpassing 98.4%, while Rooms 2&3 score slightly lower, between 94.7%-96.2%, which reveals slight confusion between them. Conversely for the noisy dataset, metrics are noticeably lower, ranging from 78.3%-81.1%. While Rooms 2&3 remain the primary source of confusion, misidentifications are more frequent and uniform, with all rooms having similar F1 scores ($\pm$0.5%).

Dataset differences

The clean dataset outperforms the noisy one across all metrics, exhibiting an average accuracy of 97.1% vs. 79.5%. As branching continues until leaf nodes are singularly labelled, decision trees tend to overfit training data, particularly evidenced in the noisy data's increased tree depth (~18.1). This suggests an overly complex model that generalises poorly and has compromised interpretability. Noisy data reduces the discriminatory power of relevant features while over-valuing irrelevant noise, resulting in incorrect tree splits.

Section 3 – Pruning (and Evaluation Again)

Nested Cross-validation classification metrics (after pruning)

Table 2: Post-Pruning Nested Cross-Validation Classification Metrics

	Clean Data	*Noisy Data*
Depth Before Pruning	11.8	18.1
Depth After Pruning	8.13	13.8
Accuracy	96.8%	87.9%

Confusion Matrix — Clean Data

		Predicted			
	Room	1	2	3	4
Actual	1	49.77	0.	0.14	0.09
	2	0.	47.68	2.32	0.
	3	0.77	1.98	46.90	0.36
	4	0.47	0.	0.32	49.21

Confusion Matrix — Noisy Data

		Predicted			
	Room	1	2	3	4
Actual	1	43.91	1.16	1.70	2.23
	2	2.01	43.54	2.90	1.24
	3	2.17	2.98	44.56	1.80
	4	2.60	1.46	1.97	43.78

Precision per Room (class) — Clean Data

Room	1	2	3	4
Rate	97.6%	96.1%	94.6%	99.2%

Precision per Room (class) — Noisy Data

Room	1	2	3	4
Rate	86.3%	88.2%	87.2%	89.4%

Recall per Room — Clean Data

Room	1	2	3	4
Rate	99.5%	95.4%	93.9%	98.4%

Recall per Room — Noisy Data

Room	1	2	3	4
Rate	89.1%	87.6%	86.8%	88.0%

F1-Measures per Room — Clean Data

Room	1	2	3	4
Rate	98.5%	95.7%	94.2%	98.8%

F1-Measures per Room — Noisy Data

Room	1	2	3	4
Rate	87.6%	87.7%	86.8%	88.6%

*Measures averaged over the $k \times (k-1) = 90$ folds and rounded to 3 significant figures (except for the confusion matrix whose entries are rounded to 2 decimal places)

Result analysis after pruning

Post-pruning, the clean dataset's accuracy marginally dropped from 97.1% to 96.8%. This reduction suggests the possibility of over-pruning, where the model might become oversimplified and lose important decision boundaries. In contrast, the noisy dataset showed a noticeable improvement across all metrics, with its accuracy, for example, jumping from 79.5% to 87.9%. This increase underscores the efficacy of pruning in mitigating overfitting, thus enhancing the model's generalisation capability and overall performance[1].

Depth analysis

Before pruning, the tree depths were ~11.8 (clean dataset) and ~18.1 (noisy dataset). After pruning, they decreased to ~8.13 and ~13.8 respectively. The improved accuracy on the noisy data, along with the slight decline in accuracy on the clean data due to these depth reductions, confirms that reducing the tree depth can enhance prediction accuracy. However, a threshold depth exists and going below it can result in decreased prediction accuracy. This phenomenon represents the bias/variance trade-off (underfitting/overfitting)[2].

[1] *[Note: Despite the enhancement, Rooms 2 & 3 are also more frequently confused than other rooms.]*

[2] *[Note: Given the patterns observed, it might be worth exploring tree depth as a specific hyperparameter in pruning to optimise performance.]*

Appendix – Code Output Screenshot

```
PRE-PRUNING EVALUATION METRICS ON CLEAN DATA:

The average confusion matrix is:
[[49.6  0.   0.1  0.3]
 [ 0.  47.6  2.4  0. ]
 [ 0.3  1.9 47.6  0.2]
 [ 0.5  0.   0.2 49.3]]
The average accuracy is: 0.9705
The average precision per class is: [0.9837308  0.96187184 0.94748462 0.99069303]
The average recall per class is: [0.99152995 0.9527838  0.95249331 0.98649587]
The average f1 per class is: [0.98752917 0.95692159 0.9494833  0.98846704]


PRE-PRUNING EVALUATION METRICS ON NOISY DATA:

The average confusion matrix is:
[[38.5  3.   3.2  4.3]
 [ 3.  40.3  4.   2.4]
 [ 2.9  4.4 40.6  3.6]
 [ 3.6  2.9  3.7 39.6]]
The average accuracy is: 0.7950000000000002
The average precision per class is: [0.80086676 0.79256929 0.78953661 0.79948804]
The average recall per class is: [0.78299954 0.8113007  0.79137343 0.79650052]
The average f1 per class is: [0.78844635 0.7984593  0.78797293 0.79580815]


POST-PRUNING EVALUATION METRICS ON CLEAN DATA:

The average pre-pruning tree depth is: 11.844444444444445
The average post-pruning tree depth is: 8.133333333333333
The average confusion matrix is:
[[49.76666667  0.          0.14444444  0.08888889]
 [ 0.         47.67777778  2.32222222  0.        ]
 [ 0.76666667  1.97777778 46.9         0.35555556]
 [ 0.46666667  0.          0.32222222 49.21111111]]
The average accuracy is: 0.9677777777777777
The average precision per class is: [0.97585489 0.96078112 0.94592022 0.9916509 ]
The average recall per class is: [0.99515631 0.95370299 0.93930262 0.98400783]
The average f1 per class is: [0.98522888 0.95655387 0.94160724 0.98772751]


POST-PRUNING EVALUATION METRICS ON NOISY DATA:

The average pre-pruning tree depth is: 18.066666666666666
The average post-pruning tree depth is: 13.833333333333334
The average confusion matrix is:
[[43.91111111  1.15555556  1.7         2.23333333]
 [ 2.01111111 43.54444444  2.9         1.24444444]
 [ 2.16666667  2.97777778 44.55555556  1.8       ]
 [ 2.6         1.45555556  1.96666667 43.77777778]]
The average accuracy is: 0.8789444444444446
The average precision per class is: [0.86267918 0.88216176 0.87179236 0.89351553]
The average recall per class is: [0.89106128 0.87615766 0.86845989 0.87994243]
The average f1 per class is: [0.87571793 0.87746006 0.86835809 0.8855732 ]
```