Introduction to Machine Learning
Coursework 1

Georgi Angelov, Aleksandar Limonov, Dennis Duka, Razvan Rusu

Link to repository: https://gitlab.doc.ic.ac.uk/dn321/intro-to-machine-learning-cw1

1. Visualisation

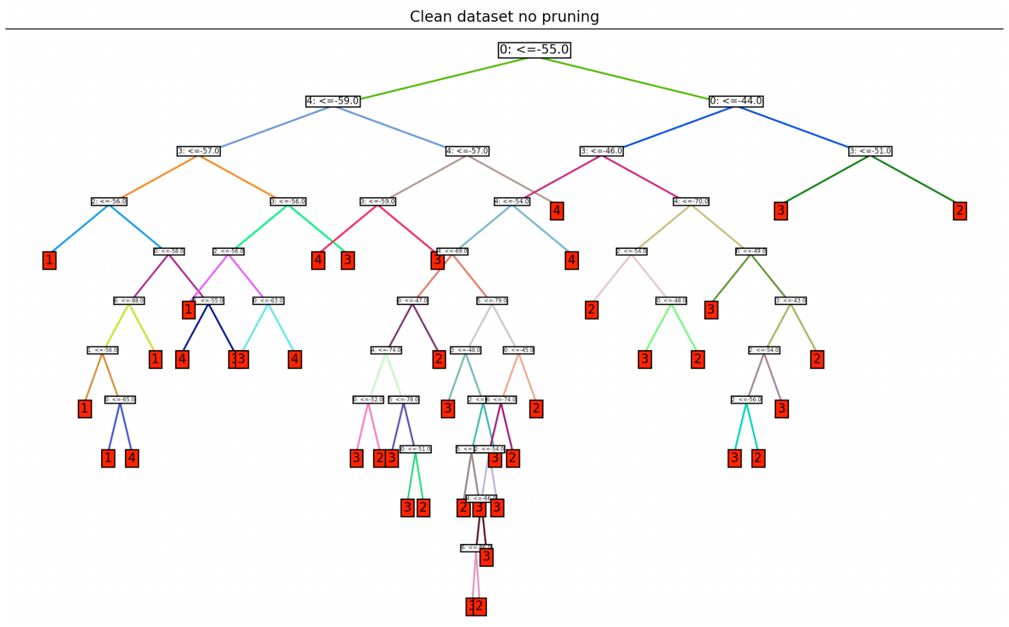
The results of the decision tree algorithm can be plotted using the matplotlib library. The tree is mapped using in-order, depth first traversal. This means that when a node is visited, it is added to the graph and then the left and right branches from that node are explored where applicable.

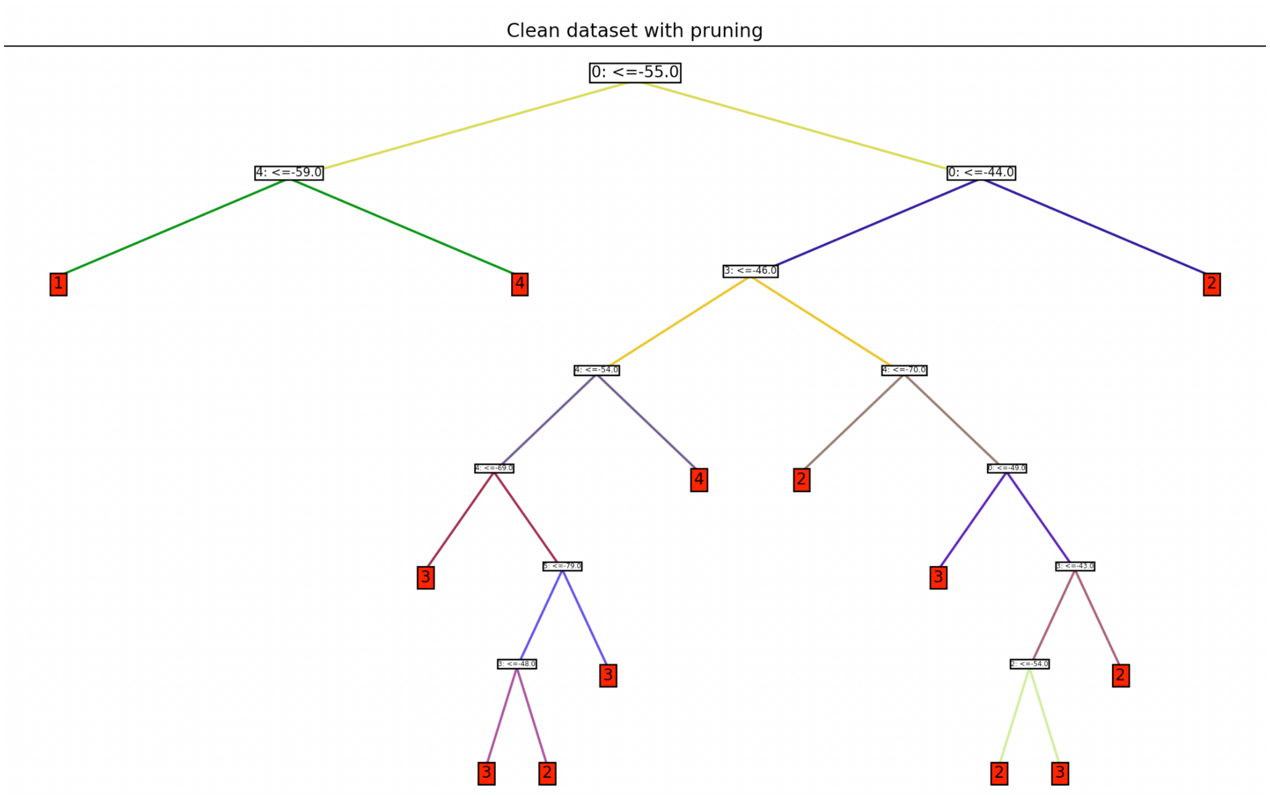
Decisions to make based on attributes are in white boxes labelled as: '*attribute*: <= *threshold*' whereas final decisions are in red boxes labelled as '*room number*'.

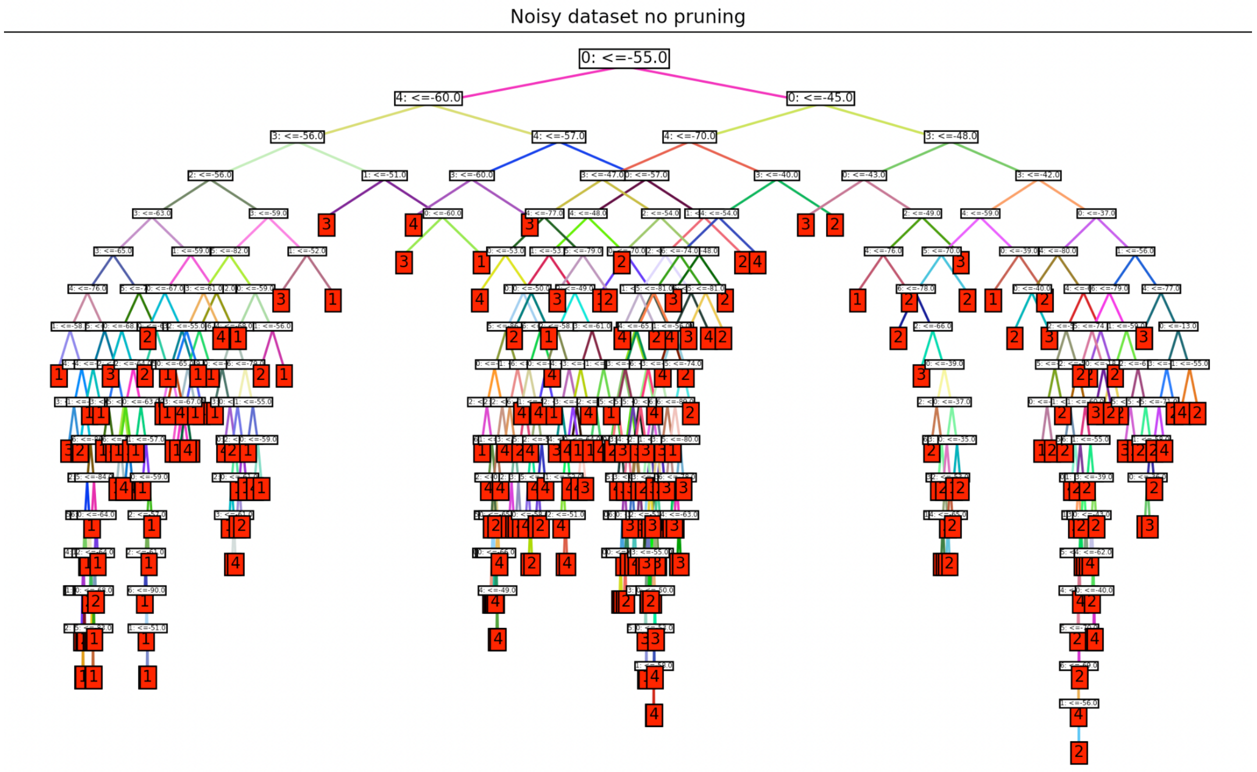
The visualisation function is called recursively from the highest-level node and works its way down exploring the left and right trees. At each node, it checks for the base case where the node is a leaf by checking whether the 'threshold' property is defined. If it is not, there are no further branches to explore, and a decision has been made.

Especially before pruning, the tree can be quite deep and intricate – making it difficult to plot clearly and can sometimes overlap depending on the shape of the tree. To try and mitigate this, each branch from a node is assigned a random colour to differentiate them from other nearby branches. In addition, how far child nodes span horizontally is exponentially inversely proportional to how deep in the tree the parent node is. These work to reduce overlaps in most cases.

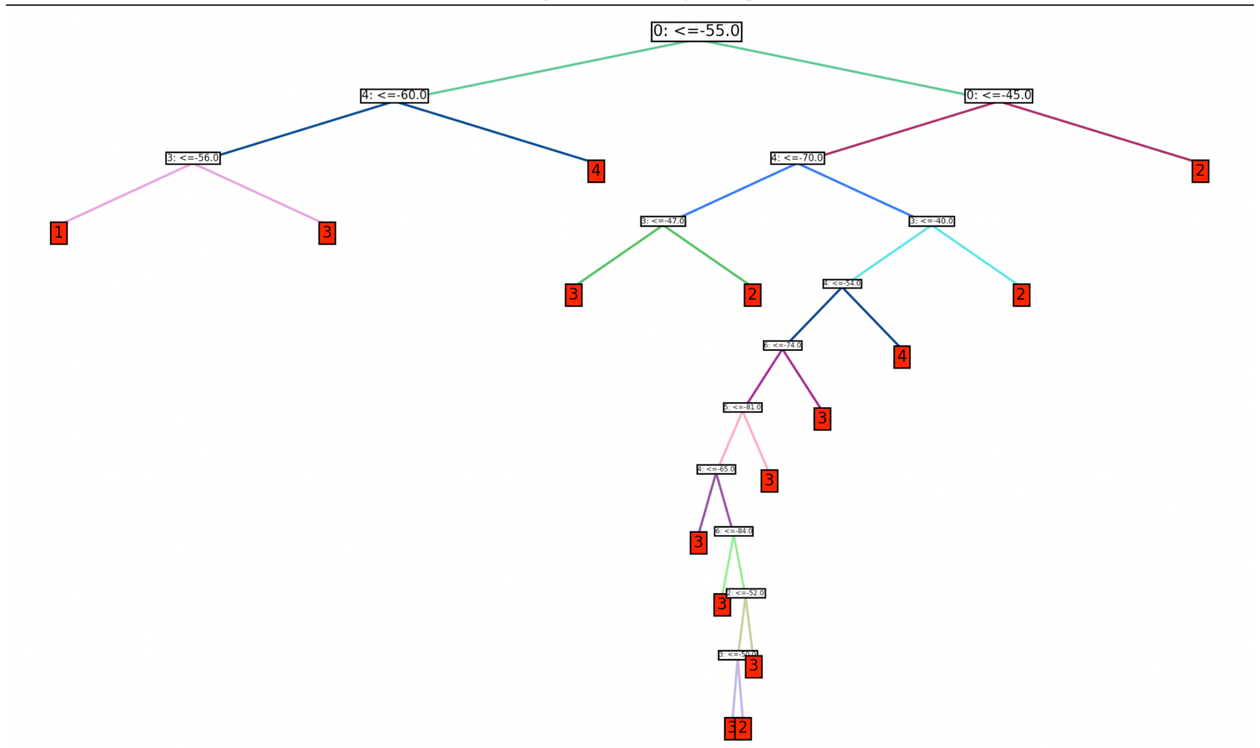
Here are some examples of visualised trees:




Clean dataset with pruning
0: <=-55.0
4: <=-59.0
0: <=-44.0
1
4
3: <=-46.0
2
4: <=-54.0
4: <=-70.0
4
2
3
3
3
3
2
2
2
3


Noisy dataset no pruning
0: <=-55.0
4: <=-60.0
0: <=-45.0
3: <=-56.0
4: <=-57.0
4: <=-70.0
3: <=-48.0
2: <=-56.0
1: <=-51.0
3: <=-60.0
3: <=-47.0
0: <=-57.0
3: <=-40.0
0: <=-43.0
3: <=-42.0

Noisy dataset with pruning



2. Evaluation

Result analysis

Figure 2 shows the confusion matrix, accuracy, recall, precision and F1-measures for the noisy dataset. It can be deduced from the confusion matrix that the room class with the highest precision is room 2 whilst the room with the lowest precision is room 3. However, the differences between the results for each class are very small, this could be because the dataset is not perfectly balanced, having 490 instances of the first class, 497 instances of the second class, 515 instances of the third class and 498 instances of the fourth class.

Cross-validation classification metrics before pruning

Average Performance for k=10 fold

Confusion Matrix

Averaged (5 d.p.)

Clean Dataset	Class 1 Predicted	Class 2 Predicted	Class 3 Predicted	Class 4 Predicted
Class 1 Actual	49.32222	0	0.36667	0.31111
Class 2 Actual	0	48.16667	1.83333	0
Class 3 Actual	0.16667	1.94444	47.63333	0.25556
Class 4 Actual	0.44444	0	0.13333	49.42222

Noisy Dataset	Class 1 Predicted	Class 2 Predicted	Class 3 Predicted	Class 4 Predicted
Class 1 Actual	38.5	3.22222	3.38889	3.88889
Class 2 Actual	2.83333	39.81111	4.12222	2.93333
Class 3 Actual	2.71111	3.81111	41.53333	3.44444
Class 4 Actual	3.78889	2.61111	3.26666	40.13333

Accuracy

(5 .s.f)

	Clean Dataset	Noisy Dataset
Accuracy	0.97272	0.79988

Recall and Precision per class

(5 s.f.)

Noisy Data	Class 1	Class 2	Class 3	Class 4
Recall	0.78571	0.80103	0.80647	0.80589
Precision	0.80488	0.80499	0.79397	0.79630

Clean Data	Class 1	Class 2	Class 3	Class 4
Recall	0.98644	0.96333	0.95267	0.98844
Precision	0.98776	0.96120	0.95330	0.98867

F1 measures

(5 s.f.)

Clean Data	Class 1	Class 2	Class 3	Class 4
F1	0.98710	0.96226	0.95298	0.98855

Noisy Data	Class 1	Class 2	Class 3	Class 4
F1	0.79518	0.80300	0.80017	0.80106

When considering the clean dataset, Figure 1, the class with the highest precision is room 4 and the room with the lowest precision is room 3. Again the differences between the results for each class are very small, albeit marginally larger.

Dataset differences

As the figures above show, there are big differences between the performances of both datasets. For instance, the accuracy of the algorithm when using the clean dataset is around 18% more than when using the noisy dataset and that pattern is seen also for the other measurements. When looking at the depth of both trees, it is clear that the noisy one has a much bigger depth. This means that there are much more noisy instances in the dataset, which could be the reason for the overall lower performance of the algorithm. Maybe the noise impacts different classes to a different extent.

3. Pruning

Cross-validation classification metrics after pruning

Average Performance for k=10 fold

Confusion Matrix

Averaged (5 d.p.)

Clean Dataset	Class 1 Predicted	Class 2 Predicted	Class 3 Predicted	Class 4 Predicted
Class 1 Actual	49.77778	0	0.15556	0.06667
Class 2 Actual	0	47.14444	2.85556	0.00000
Class 3 Actual	0.52222	1.73333	47.45556	0.28889
Class 4 Actual	0.48889	0	0	49.26667

Noisy Dataset	Class 1 Predicted	Class 2 Predicted	Class 3 Predicted	Class 4 Predicted
Class 1 Actual	44.07778	1.03333	1.77778	2.11111
Class 2 Actual	1.85556	43.21111	3.40000	1.23333
Class 3 Actual	2.25556	3.06666	44.21111	1.96667
Class 4 Actual	2.31111	1.34444	1.68889	44.45556

Accuracy

(5 .s.f)

	Clean Dataset	Noisy Dataset
Accuracy	0.96822	0.87978

Recall and Precision per class

(5 s.f.)

Clean Data	Class 1	Class 2	Class 3	Class 4
Recall	0.99556	0.94289	0.94911	0.98533
Precision	0.98009	0.96454	0.93580	0.99283

Noisy Data	Class 1	Class 2	Class 3	Class 4
Recall	0.89955	0.86944	0.85847	0.89268
Precision	0.87282	0.88810	0.86556	0.89330

F1 measures

(5 s.f.)

Clean Data	Class 1	Class 2	Class 3	Class 4
F1	0.98776	0.95359	0.94241	0.99283

Noisy Data	Class 1	Class 2	Class 3	Class 4
F1	0.88599	0.87867	0.86200	0.89298

Result analysis after pruning

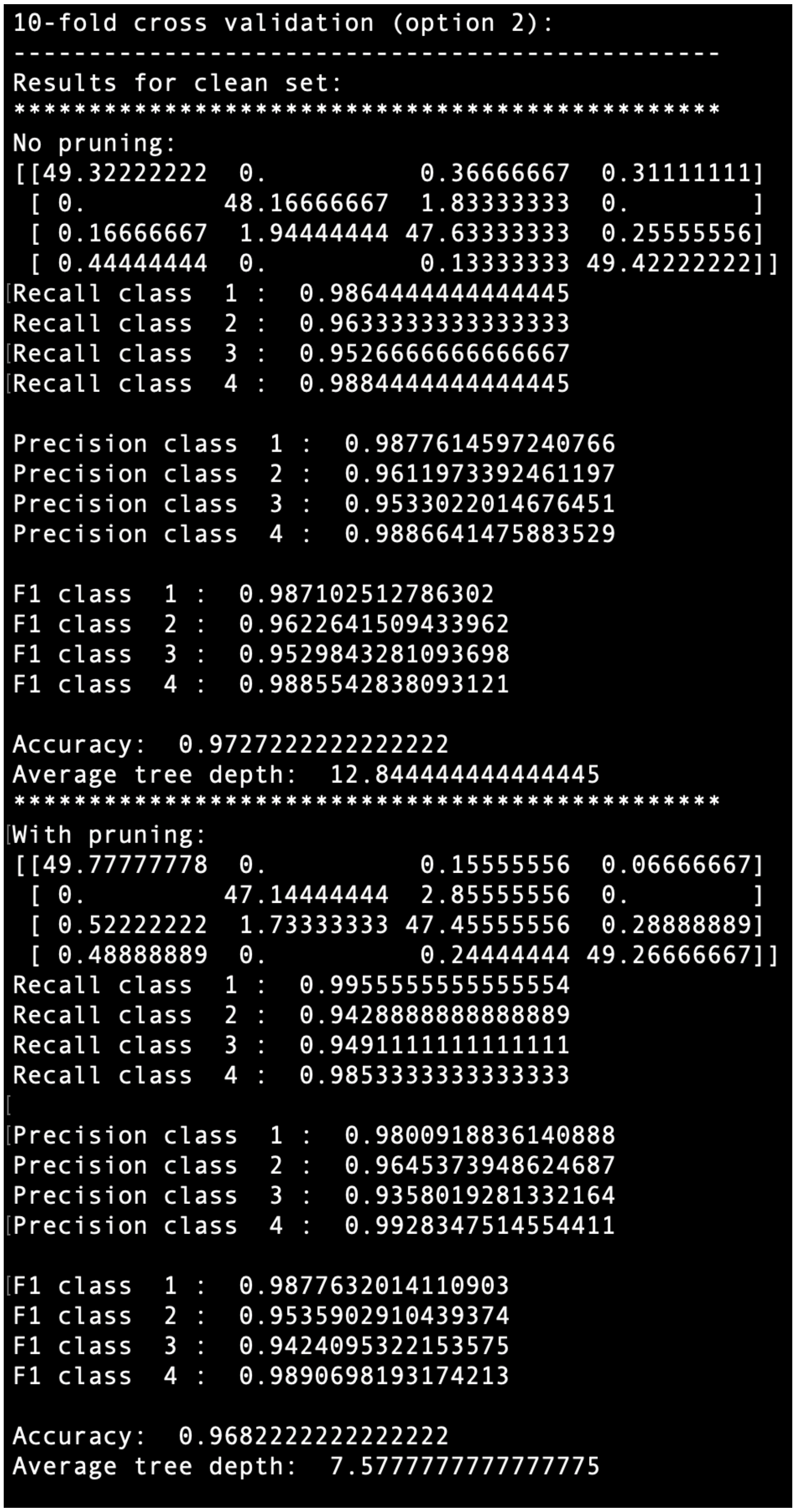
For the clean dataset, pruning did not improve any metrics; the accuracy would decrease by up to 1%. We believe that during pruning, because of the much smaller size compared to the training set, the validation set may be less indicative of the real accuracy of the decision tree model. For the noisy dataset, pruning turned out to be very effective, showing very promising results throughout, with an increase in accuracy of about 8% (± 1%).

In terms of the depth of the trees created, for the clean data, the depth for the clean dataset was decreased by 6 (± 1), which means that generalisation has been achieved, although performance hasn't improved. For the noisy data, the depth decreased quite a lot as well - about

9 (± 1). With such a significant increase in performance and decrease in depth, this shows pruning has been rather effective.

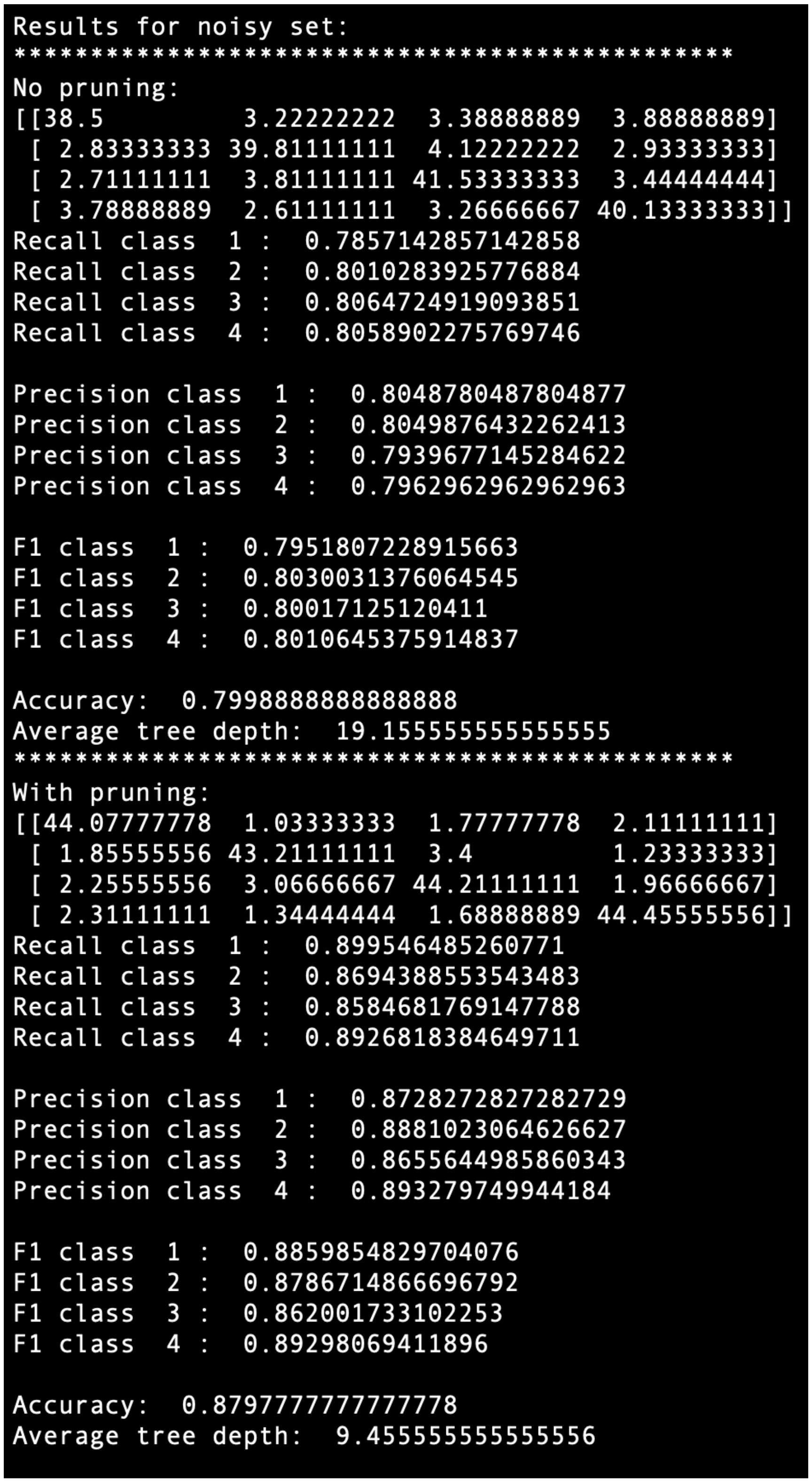
4. Additional images

Below we have shown screenshots of the full results of our 10-fold cross-evaluation for both the clean and the noisy datasets.



```
10-fold cross validation (option 2):
-------------------------------------------
Results for clean set:
*******************************************
No pruning:
[[49.32222222  0.          0.36666667  0.31111111]
 [ 0.         48.16666667  1.83333333  0.        ]
 [ 0.16666667  1.94444444 47.63333333  0.25555556]
 [ 0.44444444  0.          0.13333333 49.42222222]]
Recall class  1 :  0.9864444444444445
Recall class  2 :  0.9633333333333333
Recall class  3 :  0.9526666666666667
Recall class  4 :  0.9884444444444445

Precision class  1 :  0.9877614597240766
Precision class  2 :  0.9611973392461197
Precision class  3 :  0.9533022014676451
Precision class  4 :  0.9886641475883529

F1 class  1 :  0.987102512786302
F1 class  2 :  0.9622641509433962
F1 class  3 :  0.9529843281093698
F1 class  4 :  0.9885542838093121

Accuracy:  0.9727222222222222
Average tree depth:  12.844444444444445
*******************************************
With pruning:
[[49.77777778  0.          0.15555556  0.06666667]
 [ 0.         47.14444444  2.85555556  0.        ]
 [ 0.52222222  1.73333333 47.45555556  0.28888889]
 [ 0.48888889  0.          0.24444444 49.26666667]]
Recall class  1 :  0.9955555555555554
Recall class  2 :  0.9428888888888889
Recall class  3 :  0.9491111111111111
Recall class  4 :  0.9853333333333333

Precision class  1 :  0.9800918836140888
Precision class  2 :  0.9645373948624687
Precision class  3 :  0.9358019281332164
Precision class  4 :  0.9928347514554411

F1 class  1 :  0.9877632014110903
F1 class  2 :  0.9535902910439374
F1 class  3 :  0.9424095322153575
F1 class  4 :  0.9890698193174213

Accuracy:  0.9682222222222222
Average tree depth:  7.5777777777777775
```

Figure 1: Cross-Validation Clean Dataset



```
Results for noisy set:
*********************************************
No pruning:
[[38.5         3.22222222  3.38888889  3.88888889]
 [ 2.83333333 39.81111111  4.12222222  2.93333333]
 [ 2.71111111  3.81111111 41.53333333  3.44444444]
 [ 3.78888889  2.61111111  3.26666667 40.13333333]]
Recall class  1 :  0.7857142857142858
Recall class  2 :  0.8010283925776884
Recall class  3 :  0.8064724919093851
Recall class  4 :  0.8058902275769746

Precision class  1 :  0.8048780487804877
Precision class  2 :  0.8049876432262413
Precision class  3 :  0.7939677145284622
Precision class  4 :  0.7962962962962963

F1 class  1 :  0.7951807228915663
F1 class  2 :  0.8030031376064545
F1 class  3 :  0.80017125120411
F1 class  4 :  0.8010645375914837

Accuracy:  0.7998888888888888
Average tree depth:  19.155555555555555
*********************************************
With pruning:
[[44.07777778  1.03333333  1.77777778  2.11111111]
 [ 1.85555556 43.21111111  3.4         1.23333333]
 [ 2.25555556  3.06666667 44.21111111  1.96666667]
 [ 2.31111111  1.34444444  1.68888889 44.45555556]]
Recall class  1 :  0.899546485260771
Recall class  2 :  0.8694388553543483
Recall class  3 :  0.8584681769147788
Recall class  4 :  0.8926818384649711

Precision class  1 :  0.8728272827282729
Precision class  2 :  0.8881023064626627
Precision class  3 :  0.8655644985860343
Precision class  4 :  0.893279749944184

F1 class  1 :  0.8859854829704076
F1 class  2 :  0.8786714866696792
F1 class  3 :  0.862001733102253
F1 class  4 :  0.89298069411896

Accuracy:  0.8797777777777778
Average tree depth:  9.455555555555556
```

Figure 2: Cross-Validation Noisy Dataset Run