HOMEWORK 9083410275

YangGao
9083410275

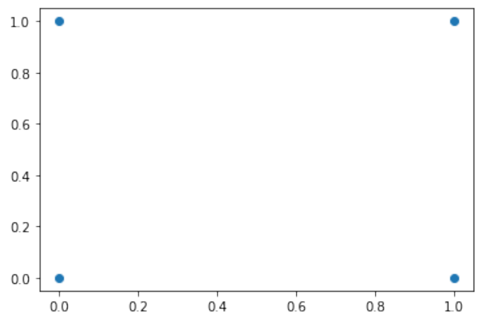
Solution 2

Solution 2.1

Intuitively, the node is a leaf as the same prediction should be made on all samples, and there's no need to split it further. Mathematically, the entropy of the node would be 0, and the node is terminated to become a leaf. $Entropy = -class0log(class0) - class1log(class1)$
class0 = fraction of 0 predictions
class1 = fraction of 1 predictions
if all predictions are 0, class0=1, class1 = 0 and log(class1)=0
if all predictions are 1, class0=0, log(class0)=0, class1 = 0
Therefore entropy = 0 if all predictions are the same

Solution 2.1

(1) Example dataset:
df=pd.DataFrame('feature':[1,0,0,1], 'target' :[0,1,0,1])



2.2

(2) The algorithm refuses to split as the information gain is 0 for this dataset, as the points are perfectly random against the target. If the split is forced, then the classes are no longer perfectly random and the algorithm will split the data further.

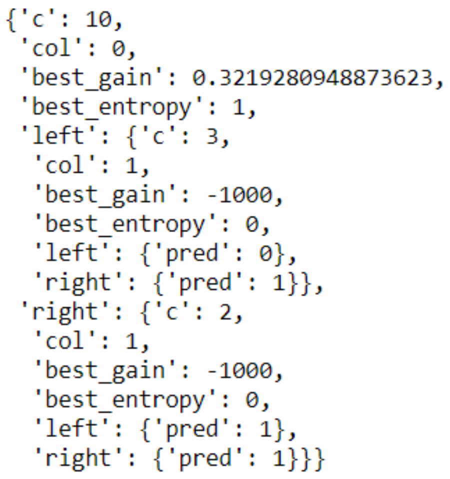
Solution 2.3

These are the candidate splits for root node:

c	column	entropy	IG
0.1	0	0.801174	0.044177
-1	1	0.801174	0.044177
0	1	0.807076	0.038275
1	1	0.840465	0.004886
2	1	0.844269	0.001082
3	1	0.829038	0.016313
4	1	0.795899	0.049452
5	1	0.740155	0.105196
6	1	0.645764	0.199587
7	1	0.807076	0.038275
8	1	0.656298	0.189053

Solution 2.4

The tree logic is shown below. Note that the best information gain is initialized at -1000 and it is not evaluated if entropy = 0

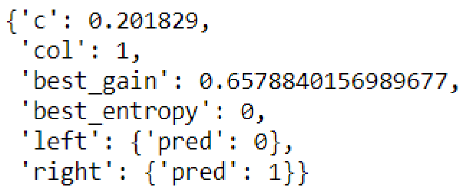


```
{'c': 10,
 'col': 0,
 'best_gain': 0.3219280948873623,
 'best_entropy': 1,
 'left': {'c': 3,
  'col': 1,
  'best_gain': -1000,
  'best_entropy': 0,
  'left': {'pred': 0},
  'right': {'pred': 1}},
 'right': {'c': 2,
  'col': 1,
  'best_gain': -1000,
  'best_entropy': 0,
  'left': {'pred': 1},
  'right': {'pred': 1}}}
```

2.4

Solution 2.5

Decision tree for D1:



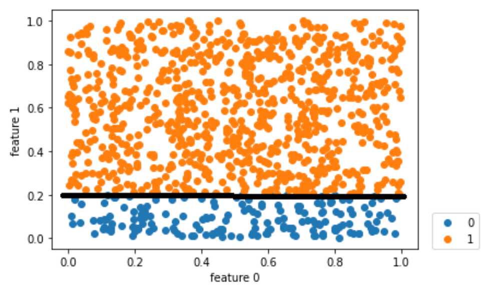
```
{'c': 0.201829,
 'col': 1,
 'best_gain': 0.6578840156989677,
 'best_entropy': 0,
 'left': {'pred': 0},
 'right': {'pred': 1}}
```

2.5 D1

The decision boundary is at feature 1 = 0.201829. When feature 1 is smaller than the boundary, the prediction is 0, otherwise it is 1.
Decision tree for D2:
{’c’: 0.533076, ’col’: 0, ’best_gain’: 0.22357293600240136, ’best_entropy’: 1, ’left’: {’c’: 0.639018, ’col’: 1, ’best_gain’: 0.34999649471050315, ’best_entropy’: 1, ’left’: {’c’: 0.534979, ’col’: 1, ’best_gain’: 0.0757438417334903, ’best_entropy’: 1, ’left’: {’c’: 0.002848, ’col’: 0, ’best_gain’: -1000, ’best_entropy’: 0, ’left’: {’pred’: 0}, ’right’: {’pred’: 0}}, ’right’: {’c’: 0.409972, ’col’: 0, ’best_gain’: 0.5694479740460766, ’best_entropy’: 1,

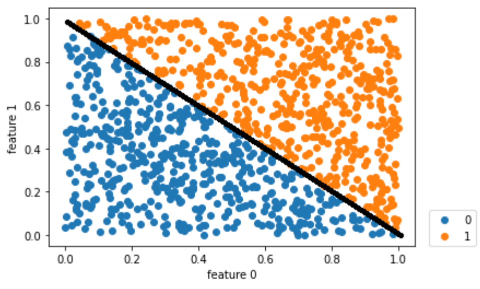
’left’: {’c’: 0.030679, ’col’: 0, ’best_gain’: -1000, ’best_entropy’: 0, ’left’: {’pred’: 0}, ’right’: {’pred’: 0}}, ’right’: {’c’: 0.426073, ’col’: 0, ’best_gain’: 0.2576788051033316, ’best_entropy’: 1, ’left’: {’c’: 0.417579, ’col’: 0, ’best_gain’: -1000, ’best_entropy’: 0, ’left’: {’pred’: 1}, ’right’: {’pred’: 0}}, ’right’: {’c’: 0.446889, ’col’: 0, ’best_gain’: -1000, ’best_entropy’: 0, ’left’: {’pred’: 1}, ’right’: {’pred’: 1}}}}}, ’right’: {’c’: 0.111076, ’col’: 0, ’best_gain’: 0.2755449824041122, ’best_entropy’: 1, ’left’: {’c’: 0.964767, ’col’: 1, ’best_gain’: 0.4460610799523433, ’best_entropy’: 0, ’left’: {’pred’: 0}, ’right’: {’pred’: 1}}, ’right’: {’c’: 0.861, ’col’: 1, ’best_gain’: 0.19938026829205352, ’best_entropy’: 1, ’left’: {’c’: 0.33046, ’col’: 0, ’best_gain’: 0.4277709381256626, ’best_entropy’: 1, ’left’: {’c’: 0.745406, ’col’: 1, ’best_gain’: 0.3212981208250194, ’best_entropy’: 1, ’left’: {’c’: 0.186339, ’col’: 0, ’best_gain’: -1000, ’best_entropy’: 0, ’left’: {’pred’: 0}, ’right’: {’pred’: 0}}, ’right’: {’c’: 0.254049, ’col’: 0, ’best_gain’: 0.5471904269481032, ’best_entropy’: 1, ’left’: {’c’: 0.191915, ’col’: 0, ’best_gain’: 0.3178113757536236, ’best_entropy’: 1, ’left’: {’c’: 0.147915, ’col’: 0, ’best_gain’: -1000, ’best_entropy’: 0, ’left’: {’pred’: 0}, ’right’: {’pred’: 0}}, ’right’: {’c’: 0.792752, ’col’: 1, ’best_gain’: 0.4591479170272448, ’best_entropy’: 0, ’left’: {’pred’: 0}, ’right’: {’pred’: 1}}}, ’right’: {’c’: 0.270046, ’col’: 0, ’best_gain’: -1000, ’best_entropy’: 0, ’left’: {’pred’: 1}, ’right’: {’pred’: 1}}}}, ’right’: {’c’: 0.340793, ’col’: 0, ’best_gain’: -1000, ’best_entropy’: 0, ’left’: {’pred’: 1}, ’right’: {’pred’: 1}}}, ’right’: {’c’: 0.118056, ’col’: 0, ’best_gain’: -1000, ’best_entropy’: 0, ’left’: {’pred’: 1}, ’right’: {’pred’: 1}}}}}, ’right’: {’c’: 0.383738, ’col’: 1, ’best_gain’: 0.35833190797692255, ’best_entropy’: 1, ’left’: {’c’: 0.761423, ’col’: 0, ’best_gain’: 0.3192484516444669, ’best_entropy’: 1, ’left’: {’c’: 0.301105, ’col’: 1, ’best_gain’: 0.2689955935892812, ’best_entropy’: 1, ’left’: {’c’: 0.541216, ’col’: 0, ’best_gain’: -1000, ’best_entropy’: 0, ’left’: {’pred’: 0}, ’right’: {’pred’: 0}}, ’right’: {’c’: 0.66337, ’col’: 0, ’best_gain’: 0.7394468924503993, ’best_entropy’: 0, ’left’: {’pred’: 0}, ’right’: {’pred’: 1}}}, ’right’: {’c’: 0.191206, ’col’: 1, ’best_gain’: 0.35601142235001465, ’best_entropy’: 1, ’left’: {’c’: 0.90482, ’col’: 0, ’best_gain’: 0.40624428894524356, ’best_entropy’: 1, ’left’: {’c’: 0.169053, ’col’: 1, ’best_gain’: 0.3064509395127767, ’best_entropy’: 1, ’left’: {’c’: 0.768663, ’col’: 0, ’best_gain’: -1000, ’best_entropy’: 0, ’left’: {’pred’: 0}, ’right’: {’pred’: 0}}, ’right’: {’c’: 0.850316, ’col’: 0, ’best_gain’: -1000, ’best_entropy’: 0, ’left’: {’pred’: 0}, ’right’: {’pred’: 1}}}, ’right’: {’c’: 0.037708, ’col’: 1, ’best_gain’: 0.49962073521348704, ’best_entropy’: 1, ’left’: {’c’: 0.941524, ’col’: 0, ’best_gain’: -1000, ’best_entropy’: 0, ’left’: {’pred’: 0}, ’right’: {’pred’: 0}}, ’right’: {’c’: 0.930371, ’col’: 0, ’best_gain’: 0.14865258200778284, ’best_entropy’: 1, ’left’: {’c’: 0.927522, ’col’: 0, ’best_gain’: 0.2516291673878229, ’best_entropy’: 0, ’left’: {’pred’: 1}, ’right’: {’pred’: 0}}, ’right’: {’c’: 0.933425, ’col’: 0, ’best_gain’: -1000, ’best_entropy’: 0, ’left’: {’pred’: 1}, ’right’: {’pred’: 1}}}}}, ’right’: {’c’: 0.762561, ’col’: 0, ’best_gain’: -1000, ’best_entropy’: 0, ’left’: {’pred’: 1}, ’right’: {’pred’: 1}}}}, ’right’: {’c’: 0.550364, ’col’: 0, ’best_gain’: 0.029045593760420537, ’best_entropy’: 1, ’left’: {’c’: 0.474971, ’col’: 1, ’best_gain’: 0.21229006661701388, ’best_entropy’: 0, ’left’: {’pred’: 0}, ’right’: {’pred’: 1}}, ’right’: {’c’: 0.551028, ’col’: 0, ’best_gain’: -1000, ’best_entropy’: 0, ’left’: {’pred’: 1}, ’right’: {’pred’: 1}}}}}}
For D2, the decision tree is very hard to interpret, as there are so many nodes. I think it is possible to interpret if I spend a lot of time studying the tree’s logic, but this is not practical.

Solution 2.6



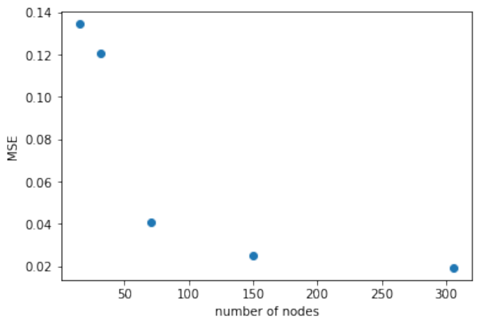
2.6 D1

The way decision trees make boundaries or is splitting along one feature, visually this would be either a horizontal line or vertical line in a 2 D feature space. The hypothesis space is all possible combinations of these splits, only allowing for straight splits along one feature at a time. Since D1’s data can be split by one horizontal line, the tree only needs one split. D2’s data needs to be split diagonally, which results in numerous horizontal and vertical lines (zig-zag line) to approximate the diagonal line.

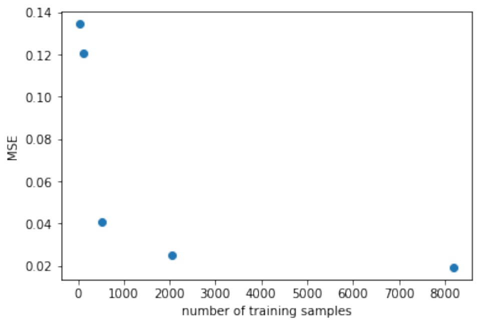


2.6 D2

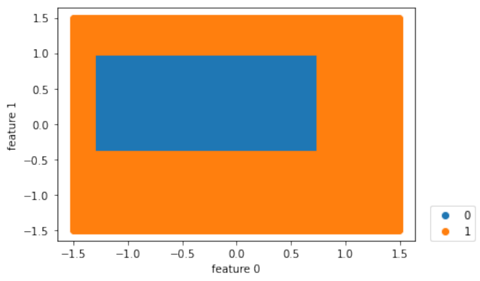
Solution 2.7



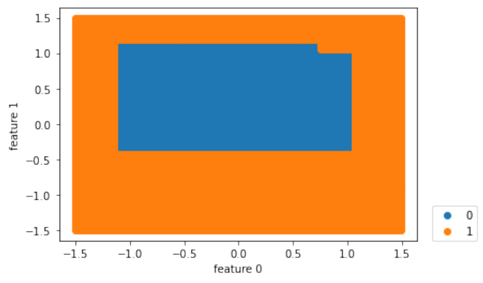
2.7(1) MSE vs nodes



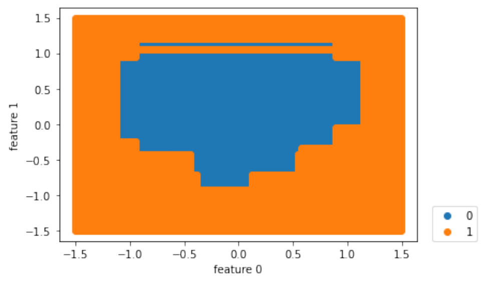
2.7(2) MSE vs number of training data



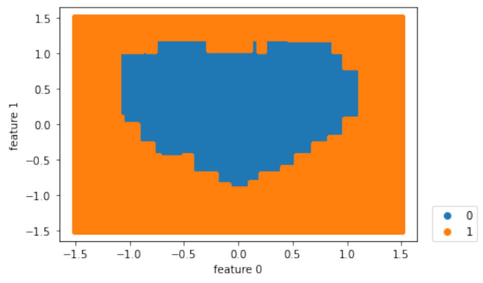
2.7(3) Decision boundary for D32 tree



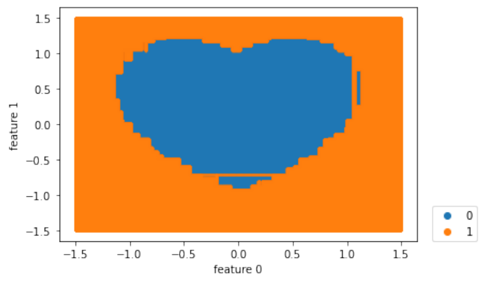
2.7(3) Decision boundary for D128 tree



2.7(3) Decision boundary for D512 tree



2.7(3) Decision boundary for D2048 tree



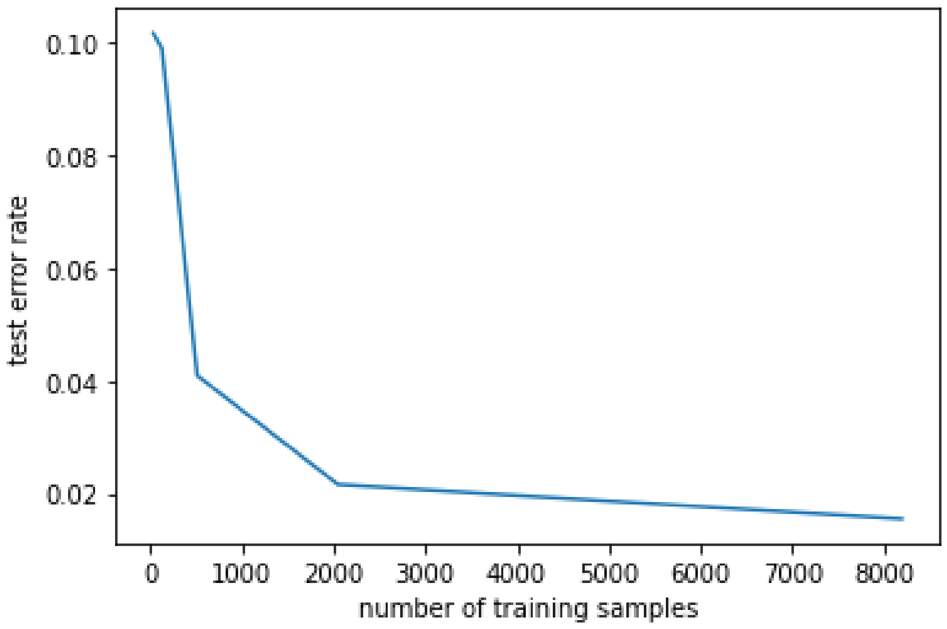
2.7(3) Decision boundary for D8192 tree

Solution 3

(1)

n	nodes	MSE
32	9	0.101770
128	21	0.107301
512	59	0.034845
2048	105	0.021571
8192	229	0.016593

(2)



Q3 (2)

Solution 4

Model train MSE: 0.0
Model test MSE: 3.81e35
I observe that the model test error is much larger than the train error. This would imply that the model is overfitting the training data and doesn't generalize well.
Adding Gaussian Noise

Standard deviation	MSE
0.01	1.51e37
1	5.19e71
10	6.47e197

I see that the mean squared error increases as I increase the standard deviation of the Gaussian noise. This makes sense, as a larger variance in noise should result in higher error.