HOMEWORK 2

Jackson Hellmers
9075444662

Completed On: 09/24/2021

Solution 2

Solution 2.1

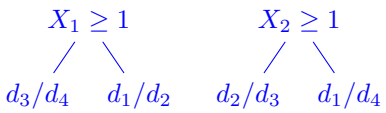
When having different features all leading to the same label we know our decision tree will stop. By calculating the info gain/gain ratio we can see that no matter which way we split we get an info gain of 0.
InfoGain= $H_D(Y) - H_D(Y|S)$ where Y is our set of output labels and S is the decided split.
When all labels in the remaining set are the same $H_D(Y) = H_D(Y|S)$ this is because Y is now independent of S. This leaves the resulting InfoGain= $H_D(Y) - H_D(Y) = 0$. Which matches the trees stopping criteria of "All splits have zero gain ratio".

Solution 2.2

If we consider the following data set.

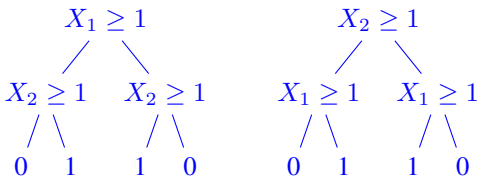
D	X_1	X_2	Y
d_1	0	0	0
d_2	0	1	1
d_3	1	1	0
d_4	1	0	1

There is no split that provides an info gain larger than another.



(1)

For both X_1 and X_2 a split will result in both decisions to contain a single 0 and 1 label. Looking at the info gain of the split at the root node we find.
InfoGain= $H_D(Y) - H_D(Y|X_1 \geq 1) = -\log_2(\frac{1}{2}) + \log_2(\frac{1}{2}) = 0$
InfoGain= $H_D(Y) - H_D(Y|X_2 \geq 1) = -\log_2(\frac{1}{2}) + \log_2(\frac{1}{2}) = 0$
So at the root node the tree has already met stopping criteria. However if we force a split we find that we can achieve a 100% accuracy split.



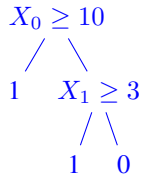
(2)

Solution 2.3

$X_0 \geq 0.0$ Info Gain: 0.0
$X_0 \geq 0.1$ Gain Ratio: 0.10051807676021828
$X_1 \geq -2.0$ Info Gain: 0.0
$X_1 \geq -1.0$ Gain Ratio: 0.10051807676021828
$X_1 \geq 0.0$ Gain Ratio: 0.055953759631263686
$X_1 \geq 1.0$ Gain Ratio: 0.00578004220515232
$X_1 \geq 2.0$ Gain Ratio: 0.0011443495172767494
$X_1 \geq 3.0$ Gain Ratio: 0.016411136842102023
$X_1 \geq 4.0$ Gain Ratio: 0.0497490641817785466

$X_1 \geq 5.0$ Gain Ratio: 0.11124029586339801
$X_1 \geq 6.0$ Gain Ratio: 0.23609960614360798
$X_1 \geq 7.0$ Gain Ratio: 0.05595375963126384
$X_1 \geq 8.0$ Gain Ratio: 0.4301569161309807
Where X_0 corresponds to the first feature column vector and X_1 is the second.

Solution 2.4



The tree generated from 'D3leaves.txt' uses the classification rules: If $X_0 < 10$ and $X_1 < 3$ then $y = 0$, otherwise $y = 1$.

Solution 2.5

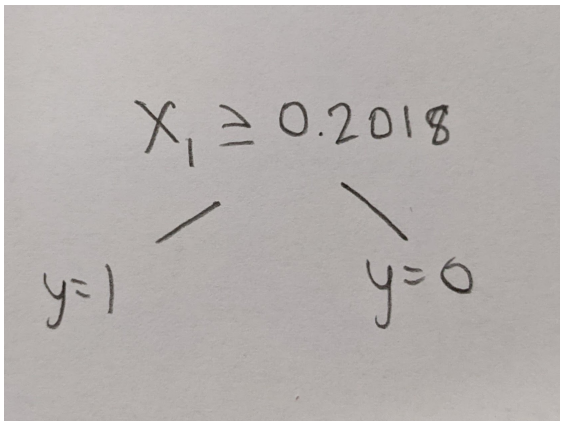


Figure 1: Decision Tree for D1.txt

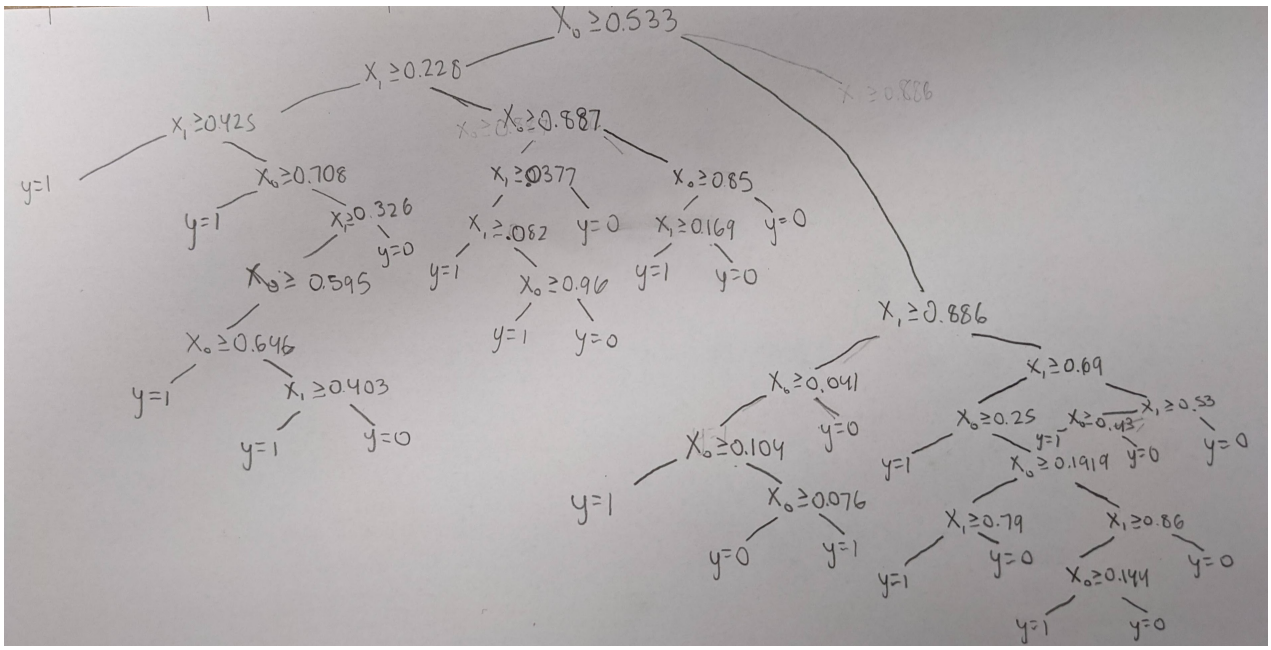
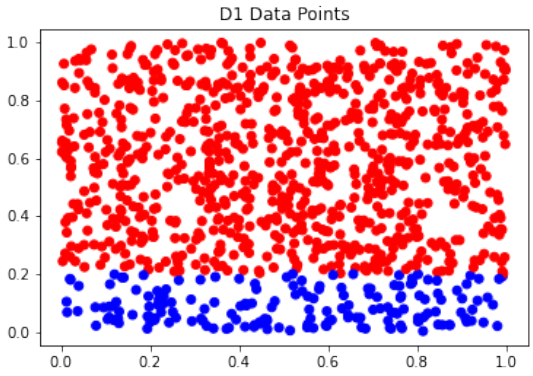


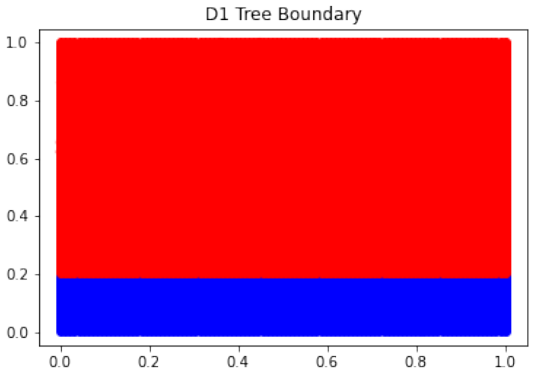
Figure 2: Decision Tree for D2.txt

For the D1 tree we can easily explain that the output just relies on the input crossing some threshold; however, without knowing what the data or the features represent it is hard to interpret any decision tree. This problem becomes even more apparent when the complexity of the tree increases as it does for D2.txt.

Solution 2.6

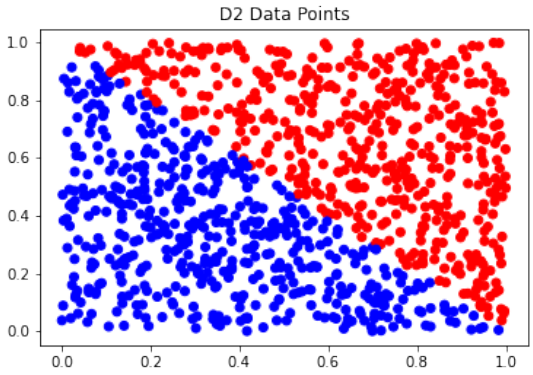


(a) Scatter Plot

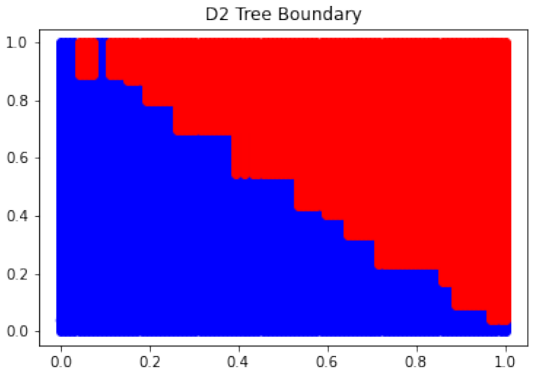


(b) Decision Boundary

Figure 3: D1 Dataset



(a) Scatter Plot



(b) Decision Boundary

Figure 4: D2 Dataset

The reason for the difference in size of each tree comes from the complexity of the decision boundary. Since each node of the tree checks either $X_0 \geq c$ OR $X_1 \geq c$, each decision split can only capture a single dimension. Since the boundary line of D2 clearly has a non-zero slope (therefore higher than single dimensional) we would need a tree node that checks both $X_0 \geq c$ AND $X_1 \geq c$ Becuase our D2 tree does not, it must switch back and forth betwen checking X_0 and X_1 causing the jagged line we see.

Solution 2.7

Size	nodes	err
32	9	0.10896
128	21	0.0945796
512	43	0.0575221
2048	111	0.0392699
8192	285	0.0254425

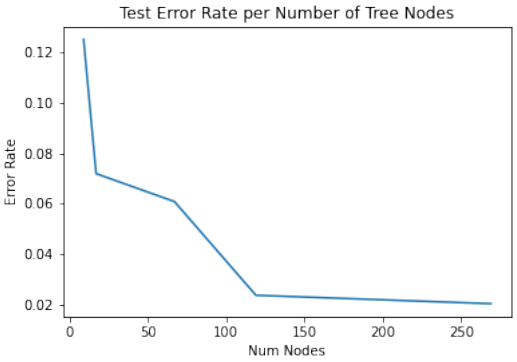


Figure 5: Learning Curve

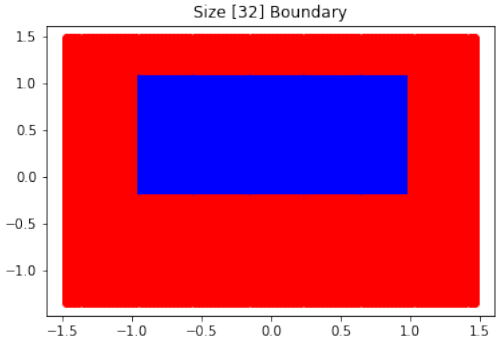


Figure 6

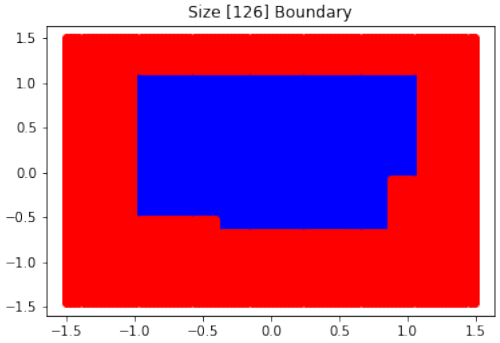


Figure 7

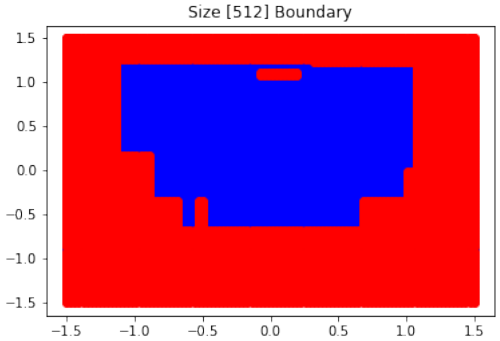


Figure 8

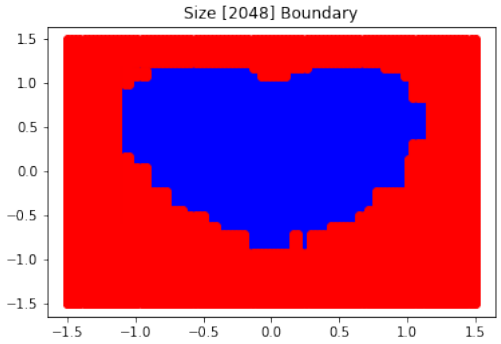


Figure 9

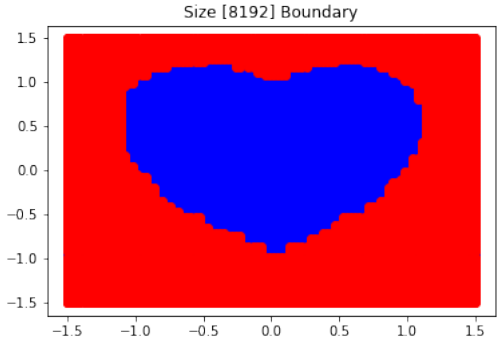


Figure 10

Solution 3

Size	nodes	err
32	9	0.153208
128	17	0.069137
512	53	0.048673
2048	97	0.022124
8192	219	0.015487

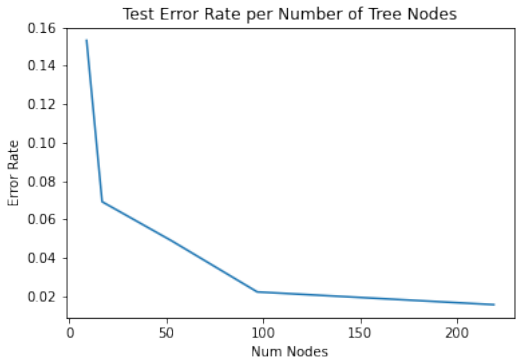
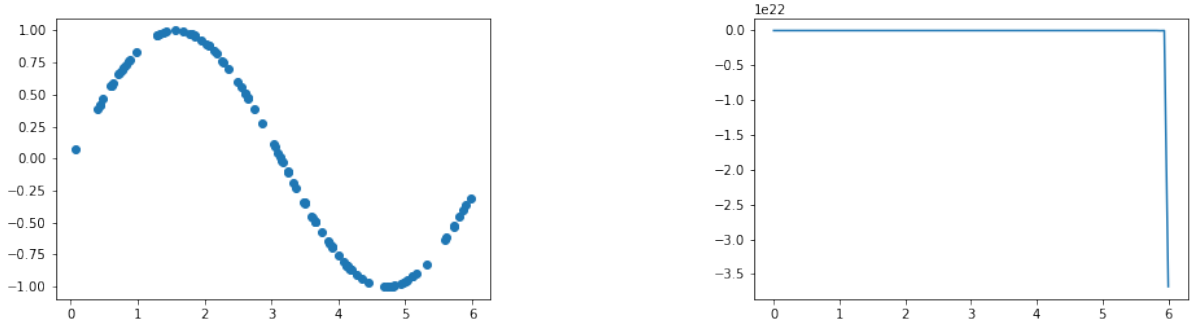


Figure 11

Solution 4



(a) Scatter Plot (b) Lagrange Interpolation

Figure 12: 100 Point Uniform $\sin(x)$ Distribution

100 Point Errors - Train RMSE: 0.0, Test RMSE: $3.18 * 10^{18}$

We can see that the interpolation using 100 points causes a great deal of instability near either endpoint. However, if we zoom in and limit the y-axis from $[-1, 1]$ we find that near center of the range ($x = [1, 4]$) the interpolation matches the $\sin(x)$ function moderately well.

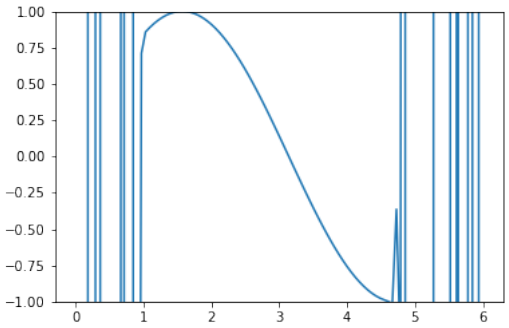
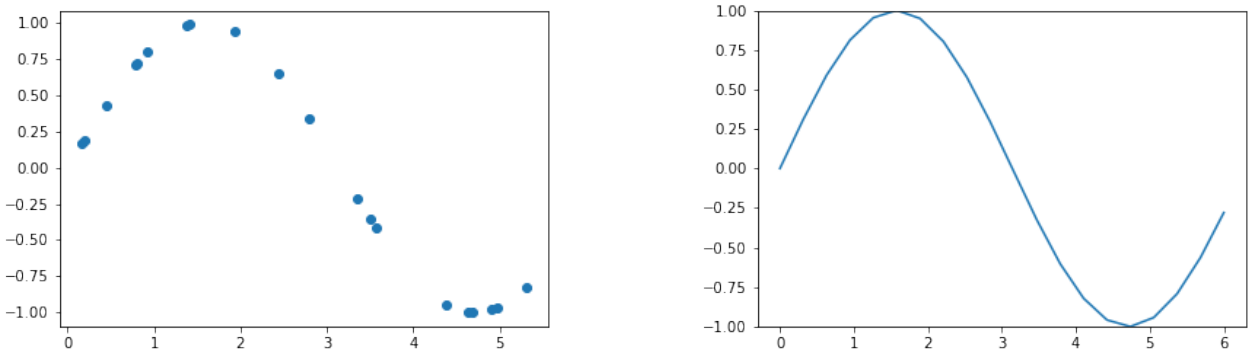


Figure 13: 100 Point Interpolation Zoomed In

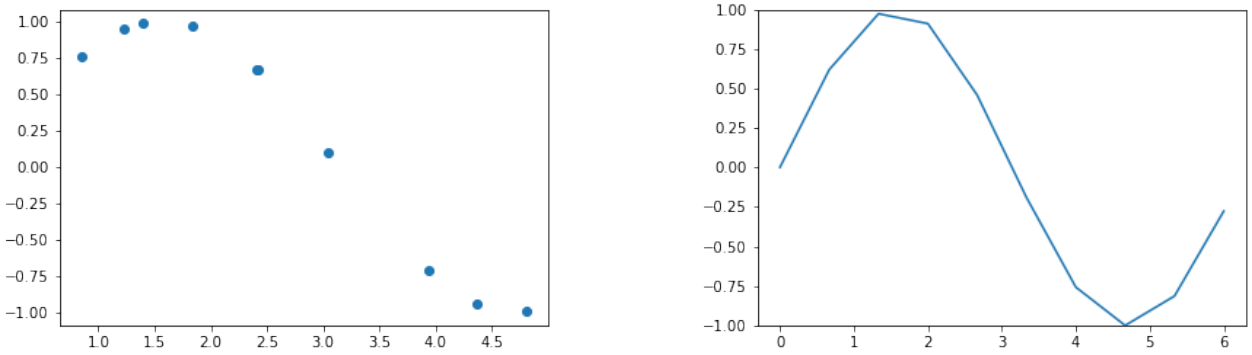
We also find that when we reduce the number of interpolation points the stability near the endpoints is greatly increased.



(a) Scatter Plot (b) Lagrange Interpolation

Figure 14: 20 Point Uniform $\sin(x)$ Distribution

20 Point Errors - Train RMSE: 0.0, Test RMSE: $8.82 * 10^{-6}$

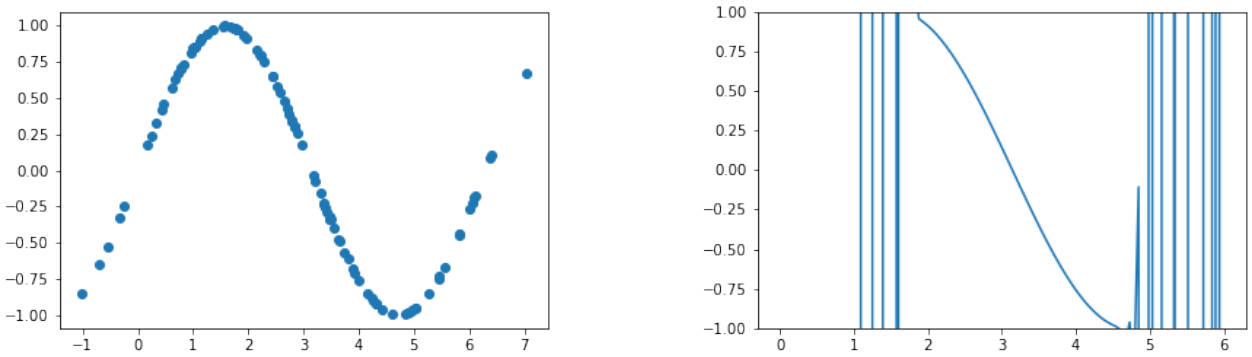


(a) Scatter Plot (b) Lagrange Interpolation

Figure 15: 10 Point Uniform $\sin(x)$ Distribution

10 Point Errors - Train RMSE: 0.0, Test RMSE $1.8 * 10^{-3}$

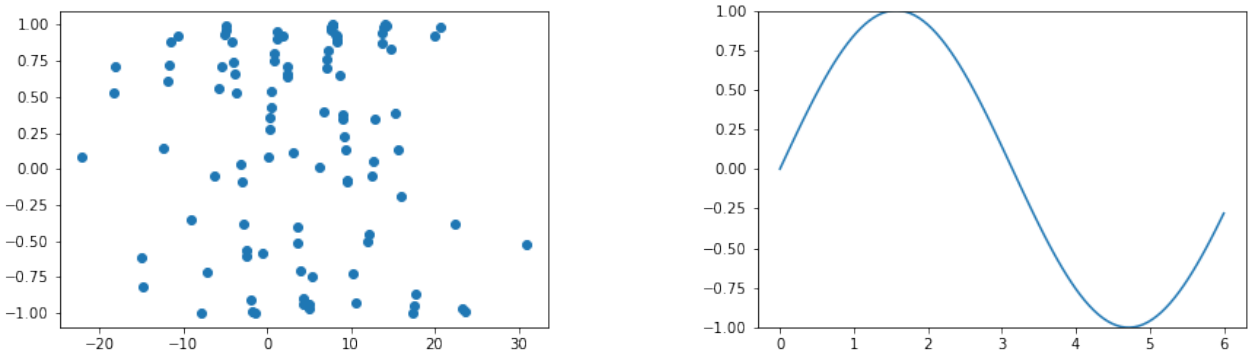
Since all of our models have little to no training error, having too many points can easily cause over-fitting. This is why reducing the number of points increased the test set accuracy.
Next we added zero mean gaussian noise to the interpolation points, altering the variance of the normal distribution to see its impact on the output.



(a) Scatter Plot (b) Lagrange Interpolation

Figure 16: 100 Point Uniform $\sin(x)$ Distribution w/ $\sigma = 1$ AWGN

100 Point $\sigma = 1$ Errors - Train RMSE: 0.0, Test RMSE: $2.40 * 10^{12}$



(a) Scatter Plot (b) Lagrange Interpolation

Figure 17: 100 Point Uniform $\sin(x)$ Distribution w/ $\sigma = 10$ AWGN

100 Point $\sigma = 10$ Errors - Train RMSE: 0.0, Test RMSE: $7.00 * 10^{-10}$

It is clear that adding a higher variance zero mean gaussian noise helps to combat over-fitting.