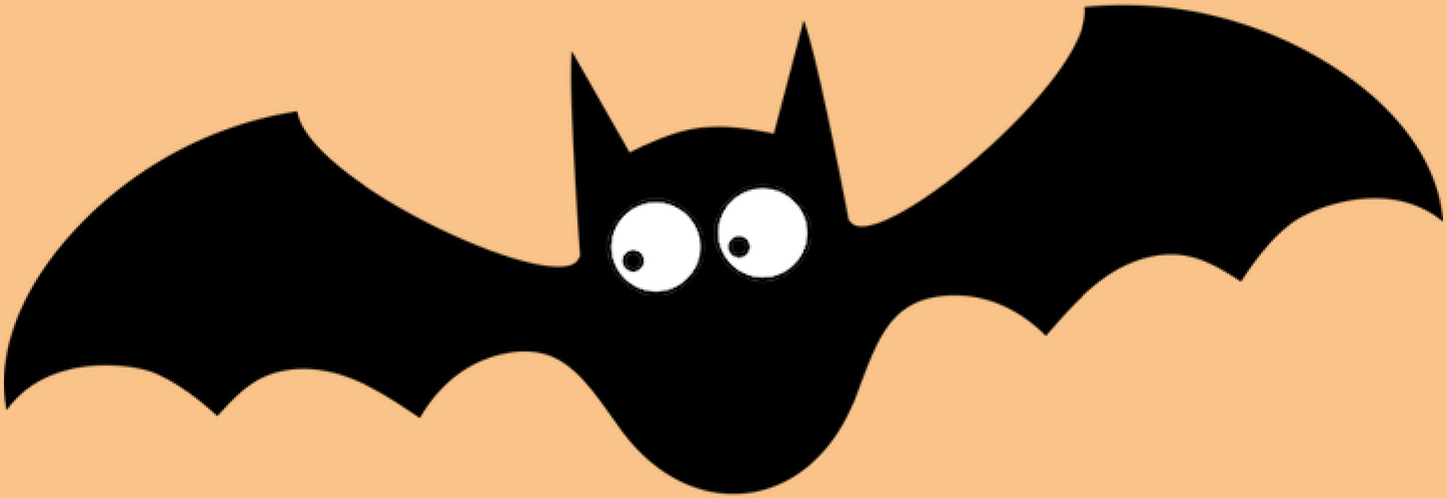


Adaboost

Son Nguyen

Adaboost

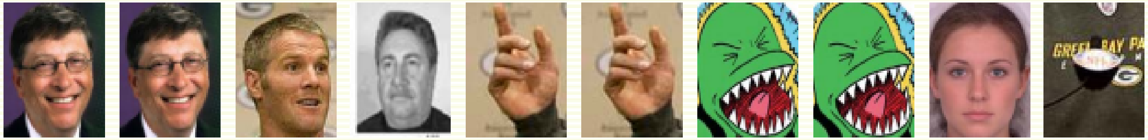
Idea Behind Ada Boost

- Examples of high weight are shown more often at later rounds
- Face/nonface classification problem:

Round 1

	1/7	1/7	1/7	1/7	1/7	1/7	1/7
best weak classifier:	✓	✗	✓	✓	✗	✓	✗
change weights:	1/16	1/4	1/16	1/16	1/4	1/16	1/4

Round 2



best weak classifier:	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓
change weights:		1/8	1/32	11/32		1/2		1/8	1/32	1/32

Adaboost

Idea Behind Ada Boost

Round 3

- out of all available weak classifiers, we choose the one that works best on the data we have at round 3
- we assume there is always a weak classifier better than random (better than 50% error)
-  image is half of the data given to the classifier
- chosen weak classifier **has to** classify this image correctly

Adaboost, Clearly Explained

- Demonstration by StatQuest
- Link

Calculation Example

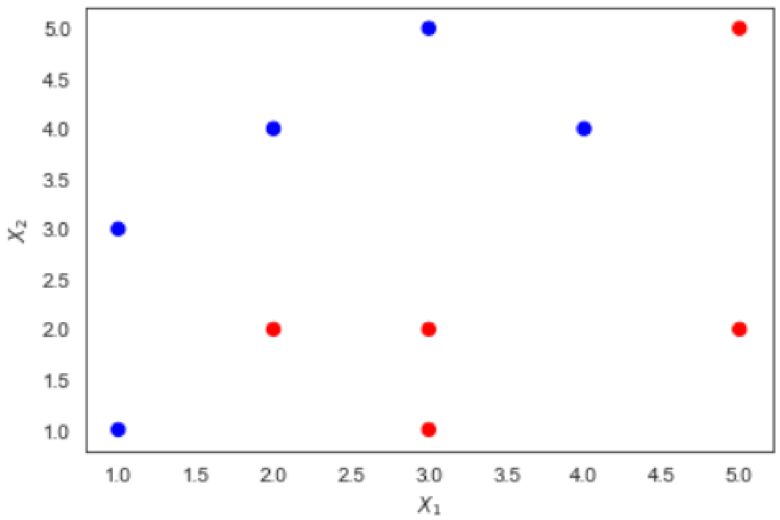
Data

x_1	x_2	**y**
1	1	1
1	3	1
2	2	-1
2	4	1
3	1	-1
3	2	-1
3	5	1
4	4	1
5	2	-1
5	5	-1

Calculation Example

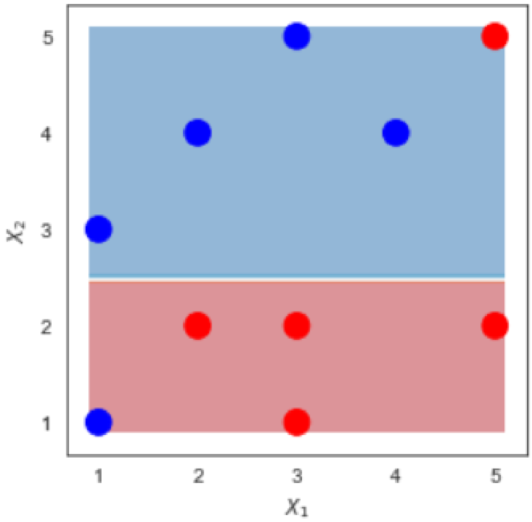
Data

x_1	x_2	**y**
1	1	1
1	3	1
2	2	-1
2	4	1
3	1	-1
3	2	-1
3	5	1
4	4	1
5	2	-1
5	5	-1

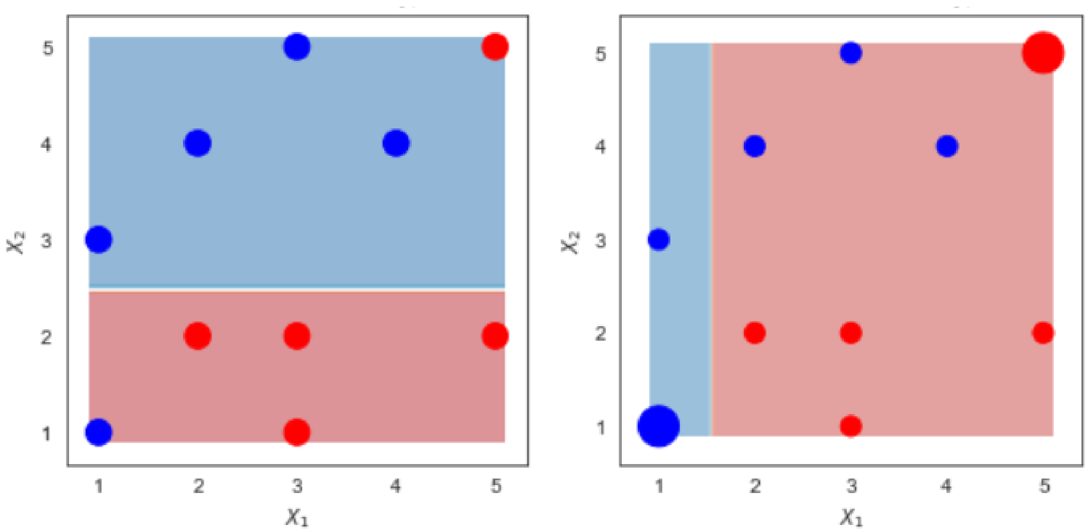


Adaboost in a nutshell

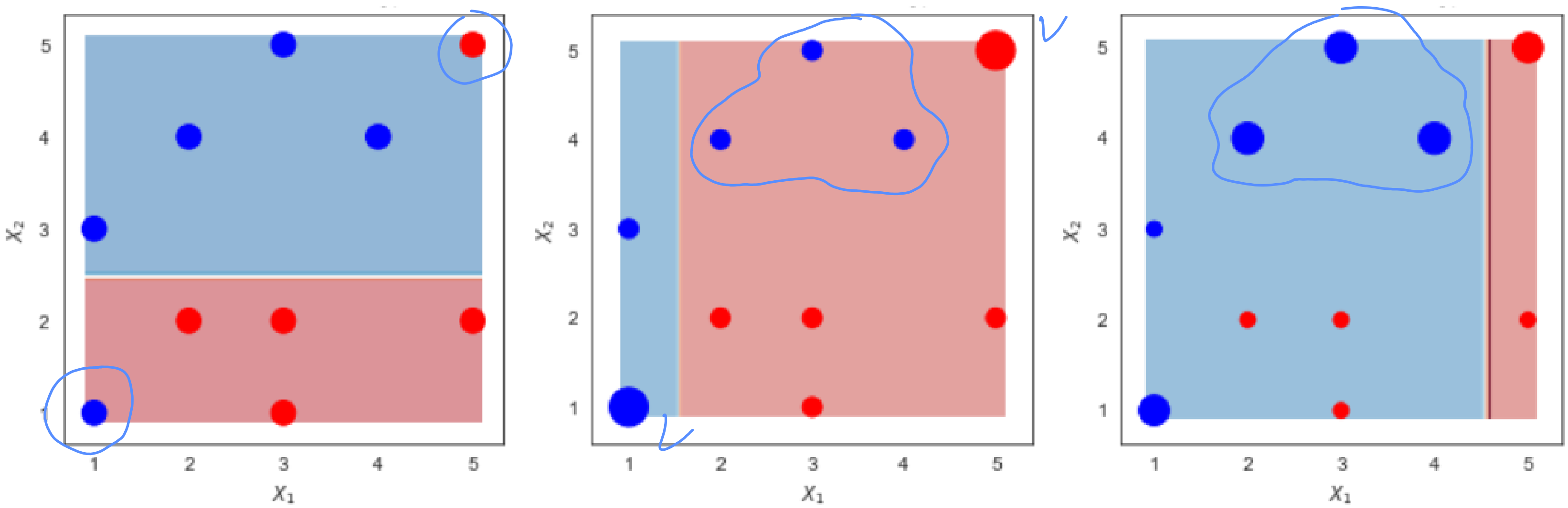
Make Stump 1



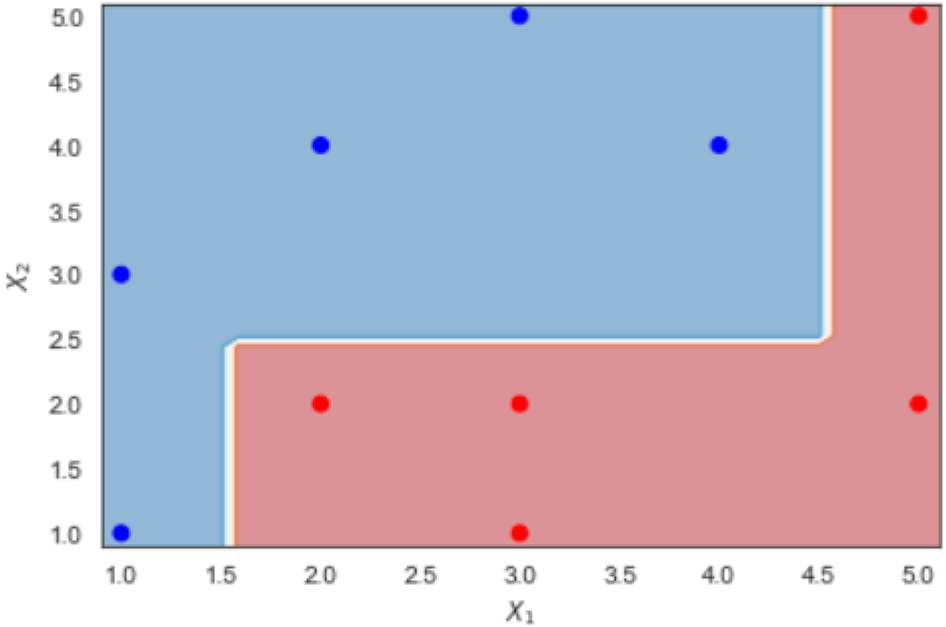
Make Stump 2



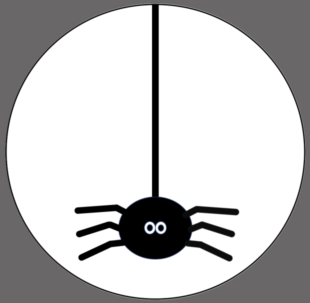
Make Stump 3



Combine the Stumps



Detail Calculation



Make the first stump

Row	x1	x2	y
0	1	1	1
1	1	3	1
2	2	2	-1
3	2	4	1
4	3	1	-1
5	3	2	-1
6	3	5	1
7	4	4	1
8	5	2	-1
9	5	5	-1

Make the first stump

- Assign weights for each row
- Every row has the same weight in the first step

Row	x1	x2	y
0	1	1	1
1	1	3	1
2	2	2	-1
3	2	4	1
4	3	1	-1
5	3	2	-1
6	3	5	1
7	4	4	1
8	5	2	-1
9	5	5	-1

Make the first stump

- Assign weights for each row
- Every row has the same weight in the first step

Row	x1	x2	y	Weight 1
0	1	1	1	0.1
1	1	3	1	0.1
2	2	2	-1	0.1
3	2	4	1	0.1
4	3	1	-1	0.1
5	3	2	-1	0.1
6	3	5	1	0.1
7	4	4	1	0.1
8	5	2	-1	0.1
9	5	5	-1	0.1

Make the first stump

- Use Weighted Gini-Index to calculate the children entropy of all candidate splits

Row	x1	x2	y	Weight 1
0	1	1	1	0.1
1	1	3	1	0.1
2	2	2	-1	0.1
3	2	4	1	0.1
4	3	1	-1	0.1
5	3	2	-1	0.1
6	3	5	1	0.1
7	4	4	1	0.1
8	5	2	-1	0.1
9	5	5	-1	0.1

Make the first stump

- Use Weighted Gini-Index to calculate the children entropy of all candidate splits
- The split with the lowest children impurity is the best split

Row	x1	x2	y	Weight 1
0	1	1	1	0.1
1	1	3	1	0.1
2	2	2	-1	0.1
3	2	4	1	0.1
4	3	1	-1	0.1
5	3	2	-1	0.1
6	3	5	1	0.1
7	4	4	1	0.1
8	5	2	-1	0.1
9	5	5	-1	0.1

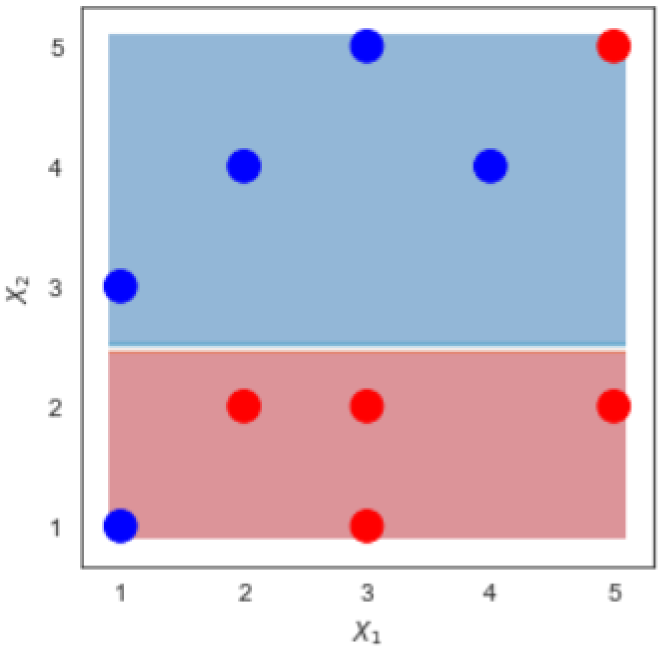
Make the first stump

- Use Weighted Gini-Index to calculate the children entropy of all candidate splits
- The split with the lowest children impurity is the best split
- **NOTE**: You are not required to make the stump. So the stump will be given to you!

Row	x1	x2	y	Weight 1
0	1	1	1	0.1
1	1	3	1	0.1
2	2	2	-1	0.1
3	2	4	1	0.1
4	3	1	-1	0.1
5	3	2	-1	0.1
6	3	5	1	0.1
7	4	4	1	0.1
8	5	2	-1	0.1
9	5	5	-1	0.1

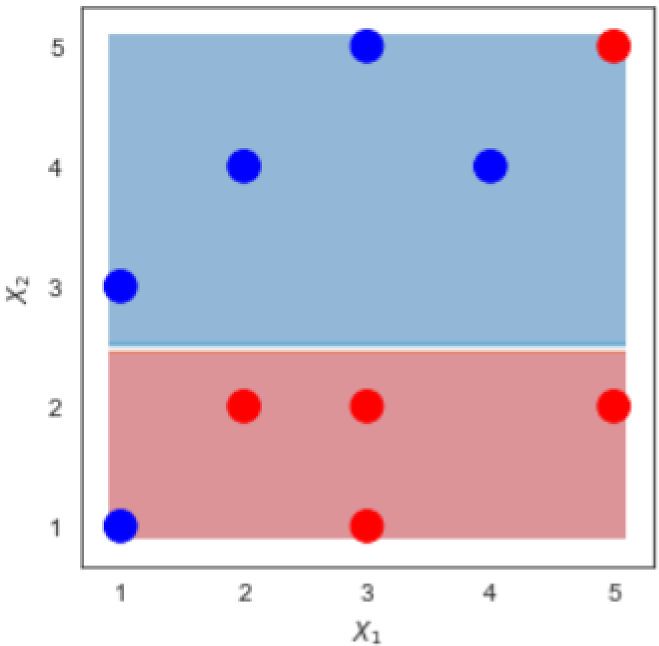
Make the first stump

- Use Weighted Gini-Index to calculate the children entropy of all candidate splits
- The split with the lowest children impurity is the best split
- **NOTE**: You are not required to make the stump. So the stump will be given to you!
- Here is the first stump



Make the first stump

- **Stump 1**: $I(x_2 > 2.5)$



Prediction of Stump 1

- **Stump 1**:

$$I(x_2 > 2.5)$$

- If $x_2 > 2.5$, predicts $y = 1$.
- Otherwise, predicts $y = -1$

Row	x1	x2	y	Stump 1 Predicts
0	1	1	1	-1
1	1	3	1	1
2	2	2	-1	-1
3	2	4	1	1
4	3	1	-1	-1
5	3	2	-1	-1
6	3	5	1	1
7	4	4	1	1
8	5	2	-1	-1
9	5	5	-1	1

Error of the first stump

Row	x1	x2	y	Stump 1 Predicts	Weight 1	
0	1	1	1	-1	0.1	<-
1	1	3	1	1	0.1	
2	2	2	-1	-1	0.1	
3	2	4	1	1	0.1	
4	3	1	-1	-1	0.1	
5	3	2	-1	-1	0.1	
6	3	5	1	1	0.1	
7	4	4	1	1	0.1	
8	5	2	-1	-1	0.1	
9	5	5	-1	1	0.1	<-

Error of the first stump

- Stump 1 has 2 misclassifications at row 0 and 9 (The predictions are NOT the same as the y values). The total weights of these rows are:

$$\epsilon_1 = 0.1 + 0.1 = 0.2$$

Row	x1	x2	y	Stump 1 Predicts	Weight 1	
0	1	1	1	-1	0.1	<-
1	1	3	1	1	0.1	
2	2	2	-1	-1	0.1	
3	2	4	1	1	0.1	
4	3	1	-1	-1	0.1	
5	3	2	-1	-1	0.1	
6	3	5	1	1	0.1	
7	4	4	1	1	0.1	
8	5	2	-1	-1	0.1	
9	5	5	-1	1	0.1	<-

Voting Power of the first Stump

- Stump 1 has 2 misclassifications at row 0 and 9 (The predictions are NOT the same as the y values). The total weights of these rows are:

$$\epsilon_1 = 0.1 + 0.1 = 0.2$$

- Voting Power: (L is the learning rate. $L = 1$ in this example 1)

$$\alpha_1 = L \cdot \frac{1}{2} \cdot \ln(\frac{1 - \epsilon_1}{\epsilon_1}) = 0.693$$

Row	x1	x2	y	Stump 1 Predicts	Weight 1	
0	1	1	1	-1	0.1	<-
1	1	3	1	1	0.1	
2	2	2	-1	-1	0.1	
3	2	4	1	1	0.1	
4	3	1	-1	-1	0.1	
5	3	2	-1	-1	0.1	
6	3	5	1	1	0.1	
7	4	4	1	1	0.1	
8	5	2	-1	-1	0.1	
9	5	5	-1	1	0.1	<-

Calculating the new weights

- For misclassified rows 0 and 9:

$$w_{new} = w_{old} \cdot e^{\alpha} = 0.1 \cdot e^{0.693} = 0.2$$

- For the correctly classified rows:

$$w_{new} = w_{old} \cdot e^{-\alpha} = 0.1 \cdot e^{-0.693} = .05$$

Row	x1	x2	y	Stump 1 Predicts	Weight 1	
0	1	1	1	-1	0.1	<-
1	1	3	1	1	0.1	
2	2	2	-1	-1	0.1	
3	2	4	1	1	0.1	
4	3	1	-1	-1	0.1	
5	3	2	-1	-1	0.1	
6	3	5	1	1	0.1	
7	4	4	1	1	0.1	
8	5	2	-1	-1	0.1	
9	5	5	-1	1	0.1	<-

Calculating the new weights

- For misclassified rows 0 and 9:

$$w_{new} = w_{old} \cdot e^{\alpha} = 0.1 \cdot e^{0.693} = 0.2$$

- For the correctly classified rows:

$$w_{new} = w_{old} \cdot e^{-\alpha} = 0.1 \cdot e^{-0.693} = .05$$

Row	x1	x2	y	Stump 1 Predicts	Weight 1	Weight 2
0	1	1	1	-1	0.1	0.2
1	1	3	1	1	0.1	0.05
2	2	2	-1	-1	0.1	0.05
3	2	4	1	1	0.1	0.05
4	3	1	-1	-1	0.1	0.05
5	3	2	-1	-1	0.1	0.05
6	3	5	1	1	0.1	0.05
7	4	4	1	1	0.1	0.05
8	5	2	-1	-1	0.1	0.05
9	5	5	-1	1	0.1	0.2

Calculating the new weights

- For misclassified rows 0 and 9:

$$w_{new} = w_{old} \cdot e^{\alpha} = 0.1 \cdot e^{0.693} = 0.2$$

- For the correctly classified rows:

$$w_{new} = w_{old} \cdot e^{-\alpha} = 0.1 \cdot e^{-0.693} = .05$$

- Notice how the weights increase for misclassified rows and decrease otherwise.

Row	x1	x2	y	Stump 1 Predicts	Weight 1	Weight 2
0	1	1	1	-1	0.1	0.2
1	1	3	1	1	0.1	0.05
2	2	2	-1	-1	0.1	0.05
3	2	4	1	1	0.1	0.05
4	3	1	-1	-1	0.1	0.05
5	3	2	-1	-1	0.1	0.05
6	3	5	1	1	0.1	0.05
7	4	4	1	1	0.1	0.05
8	5	2	-1	-1	0.1	0.05
9	5	5	-1	1	0.1	0.2

Calculating the new weights

- The total weights has to be 1. We divide the weights by the total $(.2 * 2 + .05 * 8 = .8)$ to achieve this.

Row	x1	x2	y	Stump 1 Predicts	Weight 1	Weight 2
0	1	1	1	-1	0.1	0.2
1	1	3	1	1	0.1	0.05
2	2	2	-1	-1	0.1	0.05
3	2	4	1	1	0.1	0.05
4	3	1	-1	-1	0.1	0.05
5	3	2	-1	-1	0.1	0.05
6	3	5	1	1	0.1	0.05
7	4	4	1	1	0.1	0.05
8	5	2	-1	-1	0.1	0.05
9	5	5	-1	1	0.1	0.2

Calculating the new weights

- The total weights has to be 1. We divide the weights by the total $(.2 * 2 + .05 * 8 = .8)$ to achieve this.
- Divide Weight 2 by 0.8

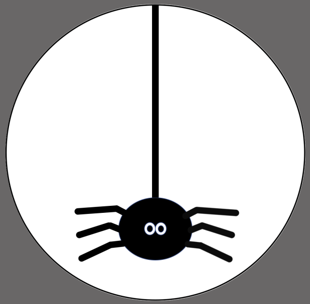
Row	x1	x2	y	Stump 1 Predicts	Weight 1	Weight 2
0	1	1	1	-1	0.1	0.2
1	1	3	1	1	0.1	0.05
2	2	2	-1	-1	0.1	0.05
3	2	4	1	1	0.1	0.05
4	3	1	-1	-1	0.1	0.05
5	3	2	-1	-1	0.1	0.05
6	3	5	1	1	0.1	0.05
7	4	4	1	1	0.1	0.05
8	5	2	-1	-1	0.1	0.05
9	5	5	-1	1	0.1	0.2

Calculating the new weights

- The total weights has to be 1. We divide the weights by the total $(.2 * 2 + .05 * 8 = .8)$ to achieve this.
- Divide Weight 2 by 0.8

Row	x1	x2	y	Stump 1 Predicts	Weight 1	Weight 2
0	1	1	1	-1	0.1	0.25
1	1	3	1	1	0.1	0.0625
2	2	2	-1	-1	0.1	0.0625
3	2	4	1	1	0.1	0.0625
4	3	1	-1	-1	0.1	0.0625
5	3	2	-1	-1	0.1	0.0625
6	3	5	1	1	0.1	0.0625
7	4	4	1	1	0.1	0.0625
8	5	2	-1	-1	0.1	0.0625
9	5	5	-1	1	0.1	0.25

Repeat the process to make the second Stump



Data to make the second Stump

Row	x1	x2	y	Weight 2
0	1	1	1	0.25
1	1	3	1	0.0625
2	2	2	-1	0.0625
3	2	4	1	0.0625
4	3	1	-1	0.0625
5	3	2	-1	0.0625
6	3	5	1	0.0625
7	4	4	1	0.0625
8	5	2	-1	0.0625
9	5	5	-1	0.25

Make the second stump

- Use Weighted Gini-Index to calculate the children entropy of all candidate splits

Row	x1	x2	y	Weight 2
0	1	1	1	0.25
1	1	3	1	0.0625
2	2	2	-1	0.0625
3	2	4	1	0.0625
4	3	1	-1	0.0625
5	3	2	-1	0.0625
6	3	5	1	0.0625
7	4	4	1	0.0625
8	5	2	-1	0.0625
9	5	5	-1	0.25

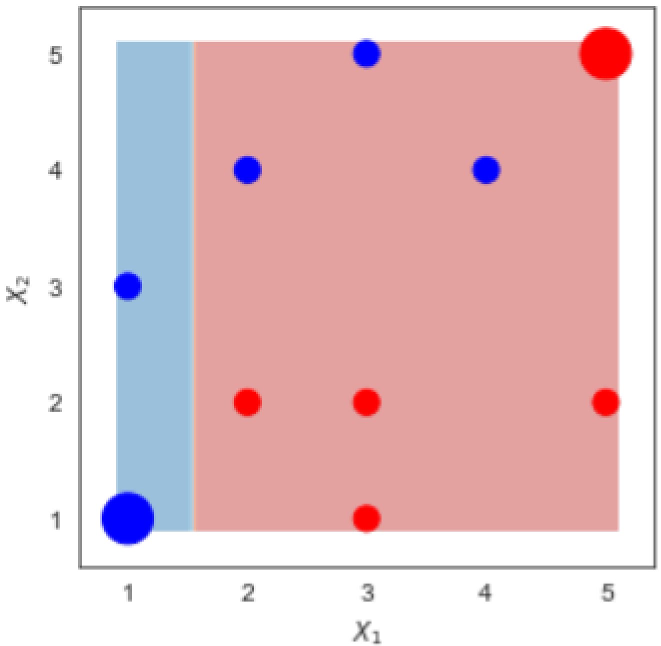
Make the second stump

- Use Weighted Gini-Index to calculate the children entropy of all candidate splits
- The split with the lowest children impurity is the best split

Row	x1	x2	y	Weight 2
0	1	1	1	0.25
1	1	3	1	0.0625
2	2	2	-1	0.0625
3	2	4	1	0.0625
4	3	1	-1	0.0625
5	3	2	-1	0.0625
6	3	5	1	0.0625
7	4	4	1	0.0625
8	5	2	-1	0.0625
9	5	5	-1	0.25

Make the second stump

- Use Weighted Gini-Index to calculate the children entropy of all candidate splits
- The split with the lowest children impurity is the best split



Error of the second stump

Row	x1	x2	y	Stump 2 Predicts	Weight 2	
0	1	1	1	1	0.25	
1	1	3	1	1	0.0625	
2	2	2	-1	-1	0.0625	
3	2	4	1	-1	0.0625	<-
4	3	1	-1	-1	0.0625	
5	3	2	-1	-1	0.0625	
6	3	5	1	-1	0.0625	<-
7	4	4	1	-1	0.0625	<-
8	5	2	-1	-1	0.0625	
9	5	5	-1	-1	0.25	

Error of the second stump

- Stump 2 has misclassifications at row 3, 6, and 7 (The predictions are NOT the same as the y values). The total weights of these rows are: 0.0625 + 0.0625 + 0.0625 = 0.1875

- Error of Stump 2:

$$\epsilon_2 = 0.1875$$

- Voting Power:

$$\alpha_2 = L \cdot \frac{1}{2} \cdot \ln(\frac{1-\epsilon_2}{\epsilon_2}) = 0.733$$

Row	x1	x2	y	Stump 2 Predicts	Weight 2	
0	1	1	1	1	0.25	
1	1	3	1	1	0.0625	
2	2	2	-1	-1	0.0625	
3	2	4	1	-1	0.0625	<-
4	3	1	-1	-1	0.0625	
5	3	2	-1	-1	0.0625	
6	3	5	1	-1	0.0625	<-
7	4	4	1	-1	0.0625	<-
8	5	2	-1	-1	0.0625	
9	5	5	-1	-1	0.25	

Calculating the new weights

- For misclassified rows 3, 6 and 7:

$$w_{new} = w_{old} \cdot e^{\alpha}$$

- For the correctly classified rows:

$$w_{new} = w_{old} \cdot e^{-\alpha}$$

Row	x1	x2	y	Stump 2 Predicts	Weight 2	
0	1	1	1	1	0.25	
1	1	3	1	1	0.0625	
2	2	2	-1	-1	0.0625	
3	2	4	1	-1	0.0625	<-
4	3	1	-1	-1	0.0625	
5	3	2	-1	-1	0.0625	
6	3	5	1	-1	0.0625	<-
7	4	4	1	-1	0.0625	<-
8	5	2	-1	-1	0.0625	
9	5	5	-1	-1	0.25	

Calculating the new weights

- For misclassified rows 3, 6 and 7:

$$w_{new} = w_{old} \cdot e^{\alpha}$$

- For the correctly classified rows:

$$w_{new} = w_{old} \cdot e^{-\alpha}$$

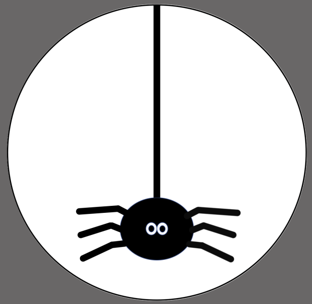
Row	x1	x2	y	Weight 2	Stump 2 Predicts	Weight 3
0	1	1	1	0.25	1	0.12012
1	1	3	1	0.0625	1	0.03003
2	2	2	-1	0.0625	-1	0.03003
3	2	4	1	0.0625	-1	0.13008
4	3	1	-1	0.0625	-1	0.03003
5	3	2	-1	0.0625	-1	0.03003
6	3	5	1	0.0625	-1	0.13008
7	4	4	1	0.0625	-1	0.13008
8	5	2	-1	0.0625	-1	0.03003
9	5	5	-1	0.25	-1	0.12012

Normalize the new weights

- The total weights has to be 1. We divide Weight 3 by the total of current Weight 3, which is 0.780624761 to achieve this.

Row	x1	x2	y	Weight 2	Stump 2 Predicts	Weight 3
0	1	1	1	0.25	1	0.15387
1	1	3	1	0.0625	1	0.03847
2	2	2	-1	0.0625	-1	0.03847
3	2	4	1	0.0625	-1	0.16664
4	3	1	-1	0.0625	-1	0.03847
5	3	2	-1	0.0625	-1	0.03847
6	3	5	1	0.0625	-1	0.16664
7	4	4	1	0.0625	-1	0.16664
8	5	2	-1	0.0625	-1	0.03847
9	5	5	-1	0.25	-1	0.15387

Repeat the process to make the third Stump



Data to Make the third stump

Row	x1	x2	y	Weight 3
0	1	1	1	0.15387
1	1	3	1	0.03847
2	2	2	-1	0.03847
3	2	4	1	0.16664
4	3	1	-1	0.03847
5	3	2	-1	0.03847
6	3	5	1	0.16664
7	4	4	1	0.16664
8	5	2	-1	0.03847
9	5	5	-1	0.15387

Make the third stump

- Use Weighted Gini-Index to calculate the children entropy of all candidate splits

Row	x1	x2	y	Weight 3
0	1	1	1	0.15387
1	1	3	1	0.03847
2	2	2	-1	0.03847
3	2	4	1	0.16664
4	3	1	-1	0.03847
5	3	2	-1	0.03847
6	3	5	1	0.16664
7	4	4	1	0.16664
8	5	2	-1	0.03847
9	5	5	-1	0.15387

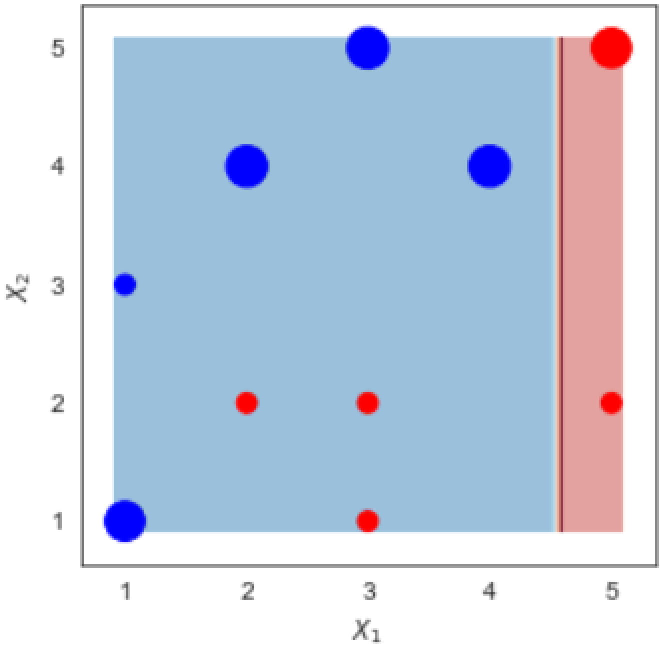
Make the third stump

- Use Weighted Gini-Index to calculate the children entropy of all candidate splits
- The split with the lowest children impurity is the best split

Row	x1	x2	y	Weight 3
0	1	1	1	0.15387
1	1	3	1	0.03847
2	2	2	-1	0.03847
3	2	4	1	0.16664
4	3	1	-1	0.03847
5	3	2	-1	0.03847
6	3	5	1	0.16664
7	4	4	1	0.16664
8	5	2	-1	0.03847
9	5	5	-1	0.15387

Make the third stump

- Use Weighted Gini-Index to calculate the children entropy of all candidate splits
- The split with the lowest children impurity is the best split



Error of the third stump

Row	x1	x2	y	Stump 3 Predicts	Weight 3	
0	1	1	1	1	0.15385	
1	1	3	1	1	0.03846	
2	2	2	-1	1	0.03846	<-
3	2	4	1	1	0.16667	
4	3	1	-1	1	0.03846	<-
5	3	2	-1	1	0.03846	<-
6	3	5	1	1	0.16667	
7	4	4	1	1	0.16667	
8	5	2	-1	-1	0.03846	
9	5	5	-1	-1	0.15385	

Error of the third stump

- Stump 3 has misclassifications at row 2, 4, and 5 (The predictions are NOT the same as the y values). The total weights of these rows are:

$$\epsilon_3 = 0.03846 \cdot 3 = 0.11538$$

- Voting Power:

$$\alpha_3 = L \cdot \frac{1}{2} \cdot \ln(\frac{1-\epsilon_3}{\epsilon_3}) = 1.018$$

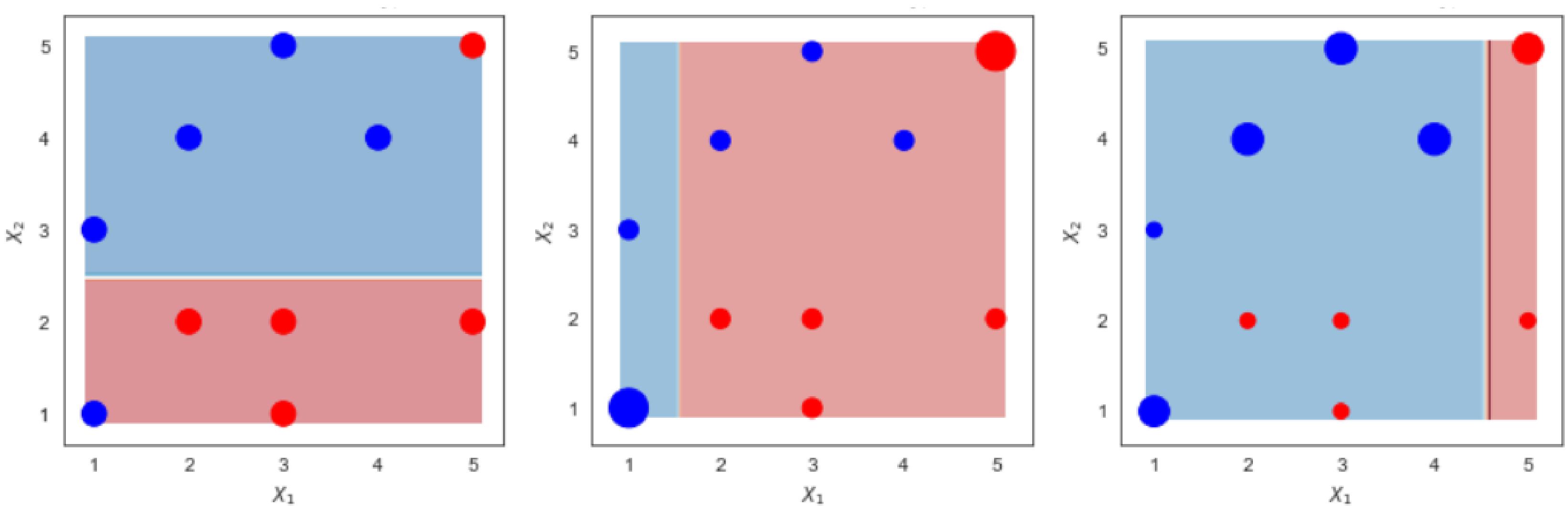
Row	x1	x2	y	Stump 3 Predicts	Weight 3	
0	1	1	1	1	0.15385	
1	1	3	1	1	0.03846	
2	2	2	-1	1	0.03846	<-
3	2	4	1	1	0.16667	
4	3	1	-1	1	0.03846	<-
5	3	2	-1	1	0.03846	<-
6	3	5	1	1	0.16667	
7	4	4	1	1	0.16667	
8	5	2	-1	-1	0.03846	
9	5	5	-1	-1	0.15385	

Summarise the results

Row	x1	x2	y	Stump 1 Predicts	Weight 1	Weight 2	Stump 2 Predicts	Weight 3	Stump 3 Predicts
0	1	1	1	-1	0.1	0.25	1	0.153846	1
1	1	3	1	1	0.1	0.0625	1	0.0384615	1
2	2	2	-1	-1	0.1	0.0625	-1	0.0384615	1
3	2	4	1	1	0.1	0.0625	-1	0.166667	1
4	3	1	-1	-1	0.1	0.0625	-1	0.0384615	1
5	3	2	-1	-1	0.1	0.0625	-1	0.0384615	1
6	3	5	1	1	0.1	0.0625	-1	0.166667	1
7	4	4	1	1	0.1	0.0625	-1	0.166667	1
8	5	2	-1	-1	0.1	0.0625	-1	0.0384615	-1
9	5	5	-1	1	0.1	0.25	-1	0.153846	-1

Combining three Stumps

- Let say we stop making new stumps here.
- We will combine the three stumps to make the final model



voting powers

$\alpha_2 = .733$

$\alpha_3 = 1.018$

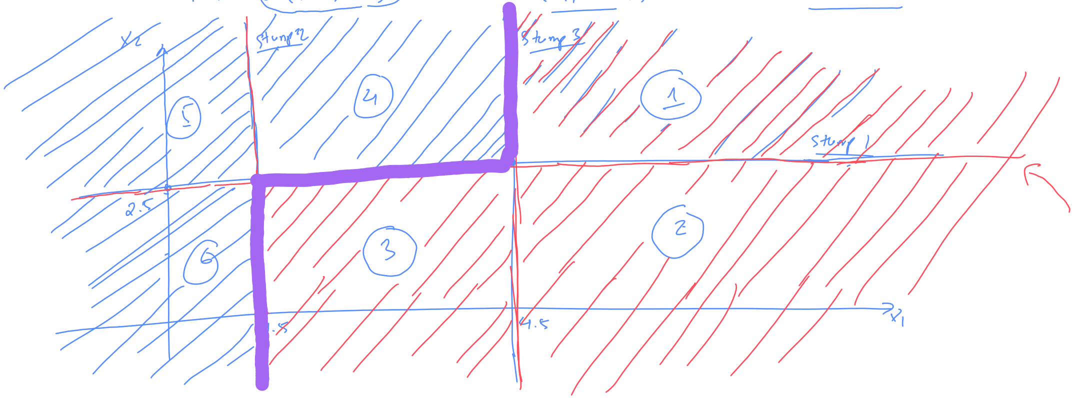
$\alpha_1 = .693$

$$\text{combination} = \alpha_1 \cdot I(X_2 \geq 2.5) + \alpha_2 \, I(X_1 \leq 1.5) + \alpha_3 \, I(X_1 \leq 4.5)$$

$$= .693 \cdot I(x_2 \geq 2.5) + .733 \cdot I(x_1 \leq 1.5) + 1.018 \cdot I(x_1 \leq 4.5)$$

$$C = \alpha_1 \cdot I(x_2 \geq 2.5) + \alpha_2 \, I(x_1 \leq 1.5) + \alpha_3 \, I(x_1 \leq 4.5)$$

$$= .693 \cdot I(x_2 \geq 2.5) + .733 \cdot I(x_1 \leq 1.5) + 1.018 \cdot I(x_1 \leq 4.5)$$



Prediction = Sign of C.

Region 1 :

$$C = .693 \; I(x_2 \geq 2.5) + .733 \cdot I(x_1 \leq 1.5) + 1.018 \; I(x_1 \leq 4.5)$$

$$= .693 \times 1 + .733 \cdot (-1) + 1.018 \times (-1)$$

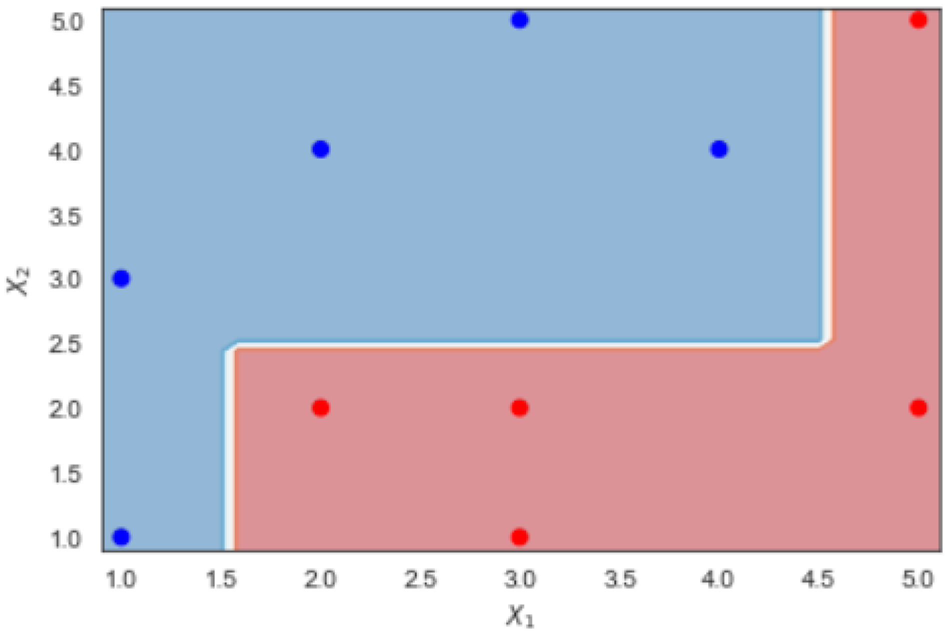
$$= .693 - .733 - 1.018 < 0$$

$$\Rightarrow \text{sign}(C) = -1$$

Region 2 : $C = -.693 - .733 - 1.018 < 0 \Rightarrow \text{sign}(C) = -1$

Region 3 : $C = -.693 - .733 + 1.018 < 0$

Combining three Stumps



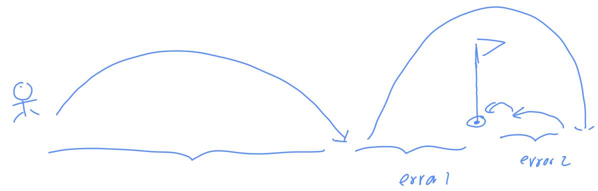
Learning rate

Data 1

X_1	X_2	Y	$\hat{Y}_1$	Error 1 = ε_1 $Y - \hat{Y}_1$
30	10	100K	20K	80 K · L
50	25	250K	500K	− 250 K · L
16	3	120K	10K	110 K · L

original data

Model 1 predicts



Data 2

X_1	X_2	ε_1 (target variable)	$\hat{\varepsilon}_1$	$\varepsilon_1 - \hat{\varepsilon}_1$ (ε_2)
30	10	80	70	10
50	25	−250	−100	−150
16	3	110	150	−40

Training data for Model 2

Data 3 :

X_1	X_2	ε_2	$\hat{\varepsilon}_2$	Final Predictions $= \hat{Y}_1 + \hat{\varepsilon}_1 + \hat{\varepsilon}_2$	Y (true value)
30	10	10	11	101 K	100
50	25	−150	−140	260 K	250
16	3	−40	−46	170 K	120

Training data of Model 3

Final Predictions of 3 models = Prediction of model 1 + Prediction of Model 2 + Prediction of model 3.