Adaboost (Solved example)

Dataset

Dataset for the AdaBoost Example

age	likes height	likes goats	go rock climbing
23	0	0	0
31	1	1	1
35	1	0	1
35	0	0	0
42	0	0	0
43	1	1	1
45	0	1	0
46	1	1	1
46	1	0	0
51	1	1	1

Dataset constructed to illustrate how to fit an AdaBoost model in Python using sklearn.

The dataset describes a classification problem, to decide whether or not a person should go rock climbing or not depending on their age, and whether the person likes goats and height.

Assigning weights to the individual data points

Build first Model - Data and Weights

age	likes height	likes goats	go rock climbing	weight
23	0	0	0	0.1
31	1	1	1	0.1
35	1	0	1	0.1
35	0	0	0	0.1
42	0	0	0	0.1
43	1	1	1	0.1
45	0	1	0	0.1
46	1	1	1	0.1
46	1	0	0	0.1
51	1	1	1	0.1

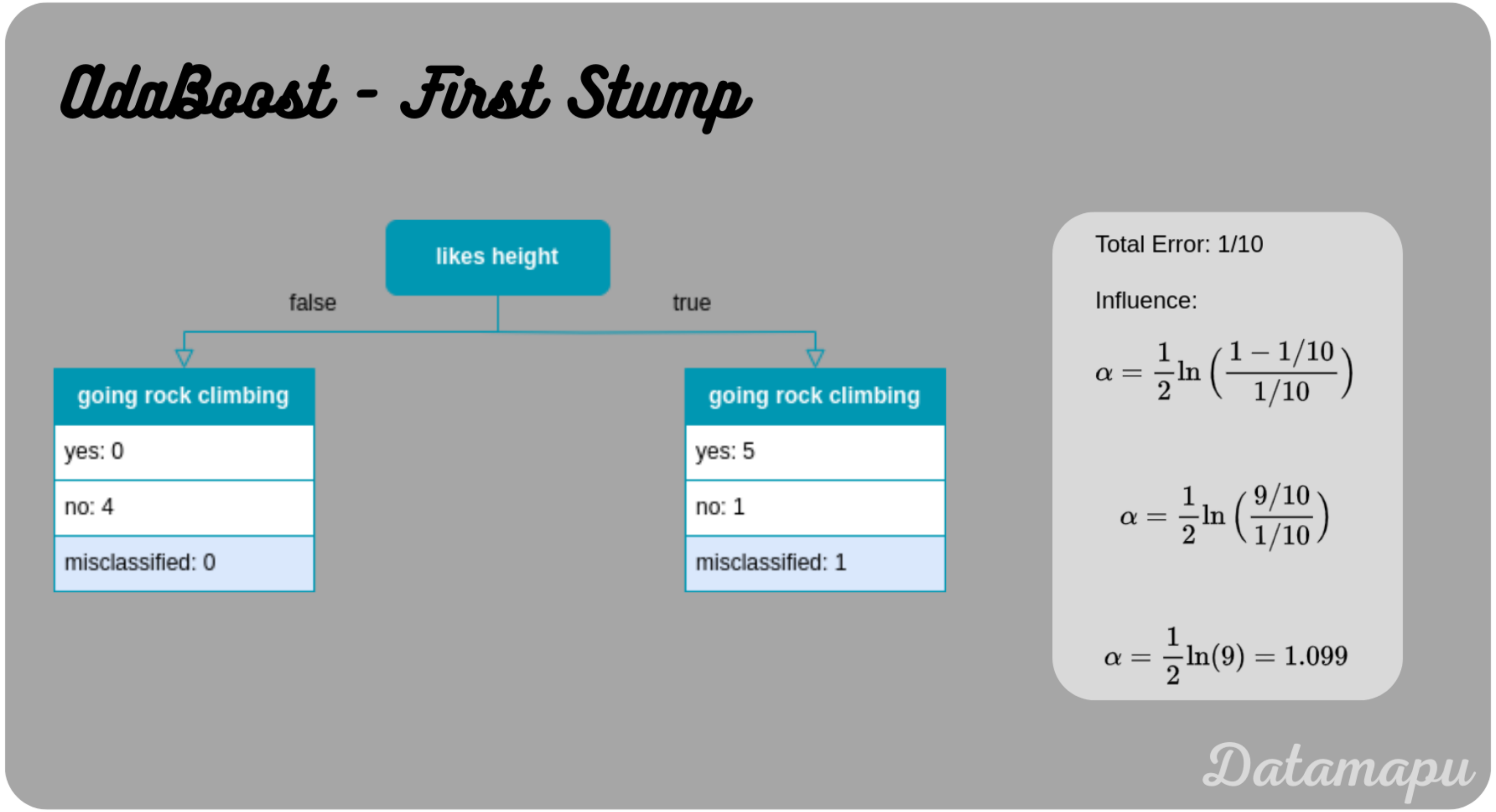
Initially, all data samples get the same weight, which is 1/N, with N the number of data points. In this case N=10. All weights sum up to 1.

Calculating influence

each underlying model - in our case decision stumps - gets a different weight, which is the so-called *influence* α. The influence depends on the *Total Error* of the model, which is equal to the number of wrongly classified samples divided by the total number of samples. The influence is defined as

$$\alpha = \frac{1}{2}\ln\left(\frac{1 - TotalError}{TotalError}\right).$$

The first stump, i.e. the first weak learner decided using Gini Impurity



Updating weights

With the *influence* α calculated, we can determine the new weights for the next iteration.

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

The sign in this equation depends on whether a sample was correctly classified or not. For correctly classified samples, we get

$$w_{new} = 0.1 \cdot e^{-1.099} = 0.0333,$$

and for wrongly classified samples

$$w_{new} = 0.1 \cdot e^{1.099} = 0.3,$$

accordingly. These weights still need to be normalized, so that their sum equals 1. This is done, by dividing by their sum. The next plot shows the dataset with the new weights.

Updating weights

Build second Model - Data and Weights

age	likes height	likes goats	go rock climbing	weight	normalized weight
23	0	0	0	0.033	0.056
31	1	1	1	0.033	0.056
35	1	0	1	0.033	0.056
35	0	0	0	0.033	0.056
42	0	0	0	0.033	0.056
43	1	1	1	0.033	0.056
45	0	1	0	0.033	0.056
46	1	1	1	0.033	0.056
46	1	0	0	0.3	0.5
51	1	1	1	0.033	0.056

Updated weights, based on the results of the first weak learner. All weights sum up to 1.

Creating Bins

The weights are used to create bins. Let's assume we have the weights $w_1, w_2, \ldots, w_N$, the bin for the first sample is $[0, w_1]$, for the second sample, $[w_1, w_1 + w_2]$, etc. In our example, the bin of the first sample is $[0, 0.056]$, for the second $[0.056, 0.112]$, etc.. The following plot shows all samples with their bins.

Creating Bins

Build second Model - Data and Bins

age	likes height	likes goats	go rock climbing	bins
23	0	0	0	[0,0.056]
31	1	1	1	[0.056, 0.111]
35	1	0	1	[0.111, 0.178]
35	0	0	0	[0.178, 0.222]
42	0	0	0	[0.222, 0.278]
43	1	1	1	[0.278, 0.333]
45	0	1	0	[0.333, 0.389]
46	1	1	1	[0.389, 0.444]
46	1	0	0	[0.444, 0.944]
51	1	1	1	[0.944, 1]

Bins for the individual samples based on the weights.

Creating Modified Dataset

Now, some randomness comes into play. Random numbers between 0 and 1 are drawn, then we check in which bin the random number falls, and the according data sample is selected for the new modified dataset. We draw as many numbers as the length of our dataset is, that is in this example we draw 10 numbers. Due to the higher weight of the misclassified example, this example has a larger bin, and the probability of drawing it is higher. Let's assume we draw the numbers $[0.1, 0.15, 0.06, 0.5, 0.65, 0.05, 0.8, 0.7, 0.95, 0.97]$, which leads to the selection of the samples $[1, 2, 1, 8, 8, 0, 8, 8, 9, 9]$. The modified dataset then has the following form.

Creating Modified Dataset

Modified Dataset

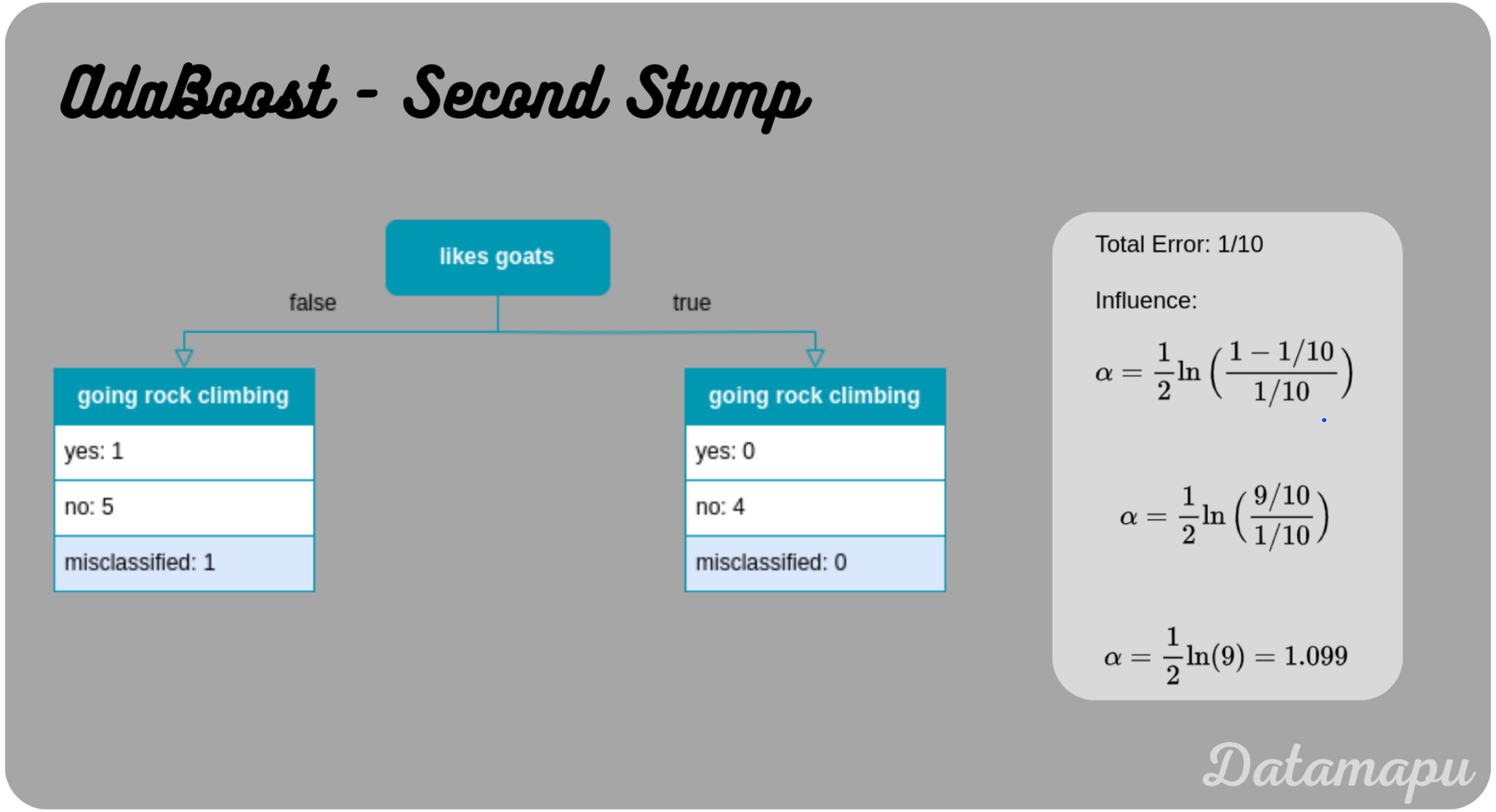
age	likes height	likes goats	go rock climbing
31	1	1	1
35 23	1	0	1
31	1	1	1
46	1	0	0
46	1	0	0
23	0	0	0
46	1	0	0
46	1	0	0
51	1	1	1
51	1	1	1

Modified dataset based on bootstrapping the original dataset using the asigned bins.

Note, that this dataset contains dublicates entries.

Second Stump

(Best spliting node is found using gini index among the three attributes again)



Updating Weights

As in the first stump, one sample is misclassified, so we get the same value for alpha as for the first stump. Accordingly, the weights are updated using the above formula.

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

For the first sample, this results in

$$w_{new} = 0.056 \cdot e^{-1.099} = 0.0185.$$

The second sample was the only one misclassified, so the sign in the exponent needs to be changed

$$w_{new} = 0.056 \cdot e^{1.099} = 0.167.$$

The following plot shows all samples together with their old weights, new weights, and normalized weights.

Updating Weights

Build second Model - Data and Weights

age	likes height	likes goats	go rock climbing	old weight	new weight	normalized weight
31	1	1	1	0.056	0.0185	0.02
35	1	0	1	0.056	0.167	0.18
31	1	1	1	0.056	0.0185	0.02
46	1	0	0	0.5	0.167	0.18
46	1	0	0	0.5	0.167	0.18
23	0	0	0	0.056	0.0185	0.02
46	1	0	0	0.5	0.167	0.18
46	1	0	0	0.5	0.167	0.18
51	1	1	1	0.056	0.0185	0.02
51	1	1	1	0.056	0.0185	0.02

Updated weights, based on the results of the first weak learner. All weights sum up to 1.

Bin Creation and random no generation for modified dataset

Indexing start from 0 for samples

We repeat the bootstrapping and draw 10 random numbers between 0 and 1. Let's assume we draw the numbers [0.3, 0.35, 0.1, 0.4, 0.97, 0.8, 0.9, 0.05, 0.25, 0.05], which refer to the samples [~~4, 4, 1, 4, 9, 7, 7, 1, 3, 1~~], then we get the following modified dataset. 3, 3, 1, 3, 8, 7, 7, 1, 3, 1.

Modified dataset

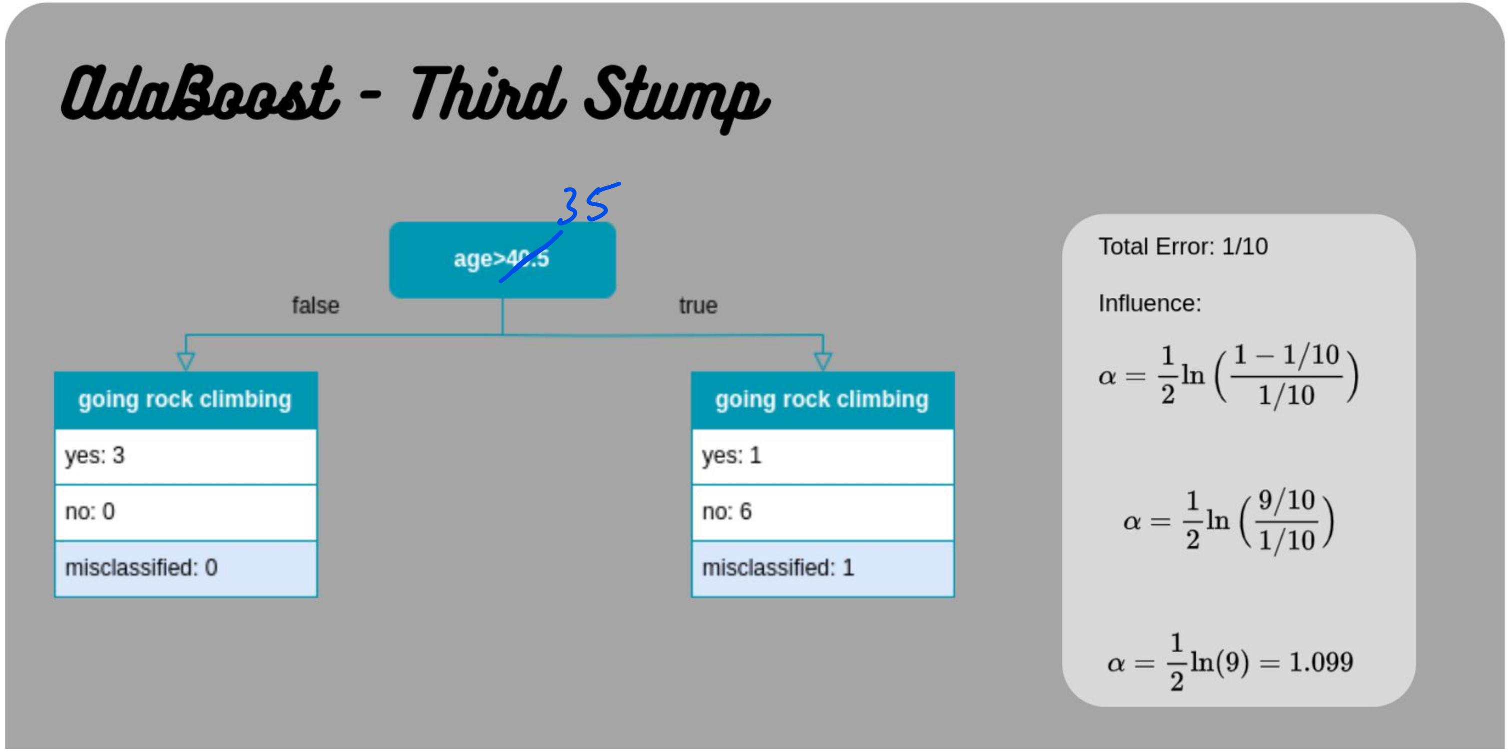
Modified Dataset

age	likes height	likes goats	go rock climbing
46	1	0	0
46	1	0	0
35	1	0	1
46	1	0	0
51	1	1	1
46	1	0	0
46	1	0	0
35	1	0	0
46	1	0	0
35	1	0	1

Modified dataset based on bootstrapping the previous dataset using the assigned bins.

Note, that this dataset contains duplicates entries.

Third stump

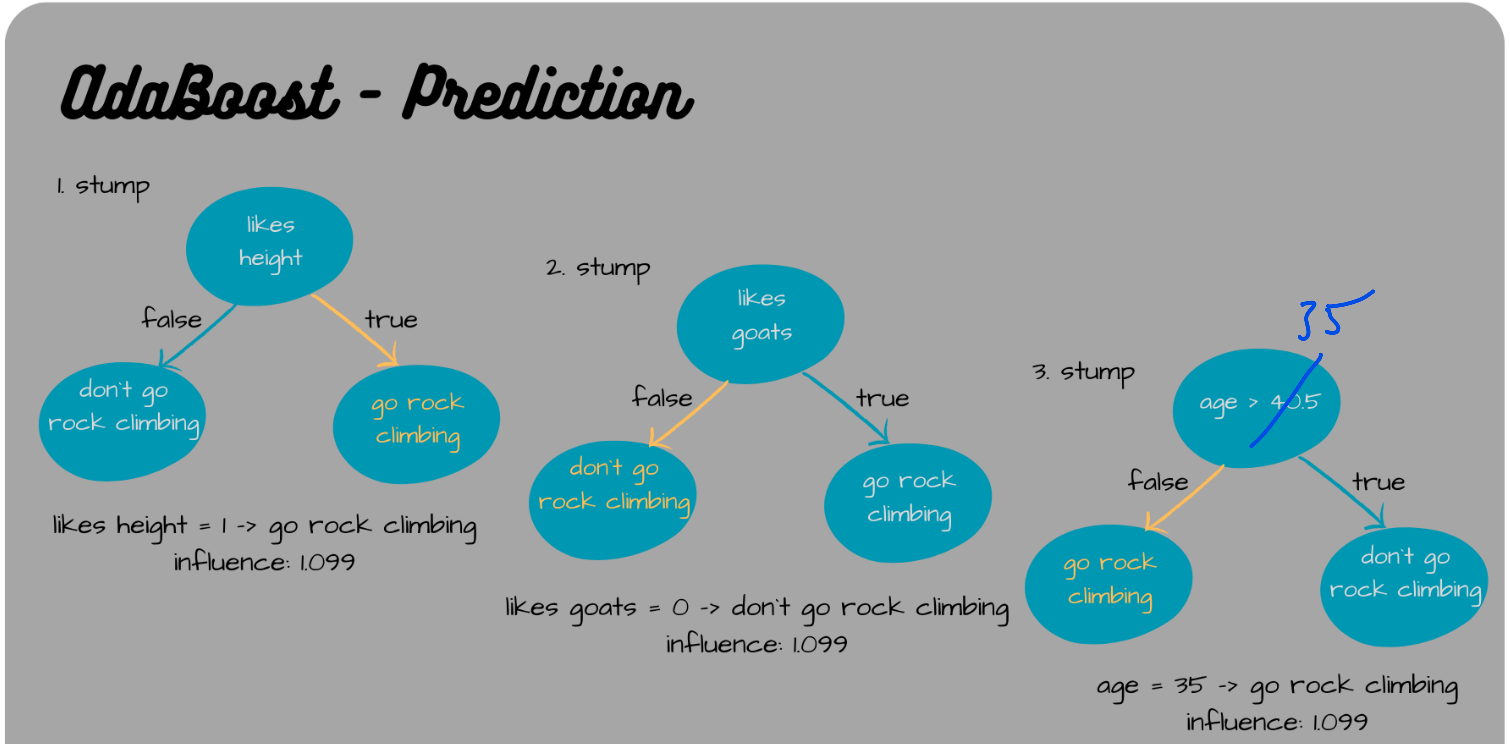

AdaBoost - Third Stump
35
age>~~40.5~~
false
true
going rock climbing
yes: 3
no: 0
misclassified: 0
going rock climbing
yes: 1
no: 6
misclassified: 1
Total Error: 1/10
Influence:
$\alpha = \frac{1}{2}\ln\left(\frac{1-1/10}{1/10}\right)$
$\alpha = \frac{1}{2}\ln\left(\frac{9/10}{1/10}\right)$
$\alpha = \frac{1}{2}\ln(9) = 1.099$

Test sample

Again, one sample is misclassified, which leads to the same influence α as previously. Note, that this is due to the very simplified dataset considered, in a more realistic example the influences of the different models would differ. We now use the individual trees and their calculated values α to determine the final prediction. Let's consider one of the samples in the dataset.

Feature	Value
age	35
likes height	1
likes goats	0

Prediction for sample



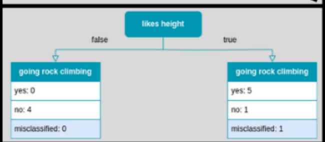
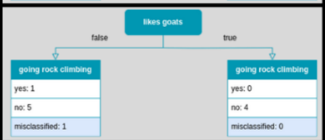
Adding influence

The final prediction is achieved by adding up the influences of each tree for the predicted classes. In this example the first and the third stump predict “go rock climbing” and the second stump predicts “don’t go rock climbing”. The first and the third stump have an influence of $1.099 + 1.099 = 2.198$, and the second stump has an influence of 1.099. That means the influence for the prediction “go rock climbing” is higher and this is thus our final prediction.

Prediction for sample

AdaBoost - Prediction

Sample: age = 35
likes height = 1
likes goats = 0

	predicted class / stump	go rock climbing	don't go rock climbing
ifluence of stump 1		1.099	
ifluence of stump 2			1.099
ifluence of stump 3		1.099	
	sum of influences	2.198	1.099

Final Prediction:
go rock climbing