# Gradient boost (classification)

How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 |
|---|---|---|---|
| Yes | 12 | Blue | Yes |
| Yes | 87 | Green | Yes |
| No | 44 | Blue | No |
| Yes | 19 | Red | No |
| No | 32 | Green | Yes |
| No | 14 | Blue | Yes |

If we have the above dataset, and we want to predict if a person love to watch “Troll2” or not

# How gradient booting works

Step 1: we start from a leaf (the “initial prediction” for all samples)

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 |
|---|---|---|---|
| Yes | 12 | Blue | Yes |
| Yes | 87 | Green | Yes |
| No | 44 | Blue | No |
| Yes | 19 | Red | No |
| No | 32 | Green | Yes |
| No | 14 | Blue | Yes |

How gradient booting works

Step 1: we start from a leaf (the "initial prediction" for all samples)

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 |
|---|---|---|---|
| Yes | 12 | Blue | Yes |
| Yes | 87 | Green | Yes |
| No | 44 | Blue | No |
| Yes | 19 | Red | No |
| No | 32 | Green | Yes |
| No | 14 | Blue | Yes |

In gradient boosting for classification, the original leaf is the "log(odds)" of "Yes"

How gradient booting works

Step 1: we start from a leaf (the "initial prediction" for all samples)

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 |
|---|---|---|---|
| Yes | 12 | Blue | Yes |
| Yes | 87 | Green | Yes |
| No | 44 | Blue | No |
| Yes | 19 | Red | No |
| No | 32 | Green | Yes |
| No | 14 | Blue | Yes |

In gradient boosting for classification, the original leaf is the "log(odds)" of "Yes"

- 4 people in the dataset "Loves Trolls == True"

# How gradient booting works

Step 1: we start from a leaf (the "initial prediction" for all samples)

In gradient boosting for classification, the original leaf is the "log(odds)" of "Yes"

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 |
|---|---|---|---|
| Yes | 12 | Blue | Yes |
| Yes | 87 | Green | Yes |
| No | 44 | Blue | No |
| Yes | 19 | Red | No |
| No | 32 | Green | Yes |
| No | 14 | Blue | Yes |

- 4 people in the dataset "Loves Trolls == True"
- 2 people in the dataset "Loves Trolls == False"

How gradient booting works

Step 1: we start from a leaf (the "initial prediction" for all samples)

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 |
|---|---|---|---|
| Yes | 12 | Blue | Yes |
| Yes | 87 | Green | Yes |
| No | 44 | Blue | No |
| Yes | 19 | Red | No |
| No | 32 | Green | Yes |
| No | 14 | Blue | Yes |

In gradient boosting for classification, the original leaf is the "log(odds)" of "Yes"

- 4 people in the dataset "Loves Trolls == True"
- 2 people in the dataset "Loves Trolls == False"

Then "log(odds)" is log(4/2) = 0.69

How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 |
|---|---|---|---|
| Yes | 12 | Blue | Yes |
| Yes | 87 | Green | Yes |
| No | 44 | Blue | No |
| Yes | 19 | Red | No |
| No | 32 | Green | Yes |
| No | 14 | Blue | Yes |

Step 1: we start from a leaf (the "initial prediction" for all samples)

In gradient boosting for classification, the original leaf is the "log(odds)"

- 4 people in the dataset "Loves Trolls == True"
- 2 people in the dataset "Loves Trolls == False"

Then "log(odds)" is log(4/2) = 0.69

So the first/original leaf is Log(odds) = 0.69

# How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 |
|---|---|---|---|
| Yes | 12 | Blue | Yes |
| Yes | 87 | Green | Yes |
| No | 44 | Blue | No |
| Yes | 19 | Red | No |
| No | 32 | Green | Yes |
| No | 14 | Blue | Yes |

Step 1: we start from a leaf (the "initial prediction" for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create "probability" and use it to do the classification

# How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 |
|---|---|---|---|
| Yes | 12 | Blue | Yes |
| Yes | 87 | Green | Yes |
| No | 44 | Blue | No |
| Yes | 19 | Red | No |
| No | 32 | Green | Yes |
| No | 14 | Blue | Yes |

Step 1: we start from a leaf (the "initial prediction" for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create "probability" and use it to do the classification

The probability is estimated by the "Logistic Function", which is

$$Probability_{Yes} = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}$$

How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 |
|---|---|---|---|
| Yes | 12 | Blue | Yes |
| Yes | 87 | Green | Yes |
| No | 44 | Blue | No |
| Yes | 19 | Red | No |
| No | 32 | Green | Yes |
| No | 14 | Blue | Yes |

Step 1: we start from a leaf (the "initial prediction" for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create "probability" and use it to do the classification

The probability is estimated by the "Logistic Function", which is

$$Probability_{Yes} = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}} = \frac{e^{0.69}}{1 + e^{0.69}} = 0.67$$

So we got 0.67 as the probability of "loving Troll2" (or "Yes")

How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 |
|---|---|---|---|
| Yes | 12 | Blue | Yes |
| Yes | 87 | Green | Yes |
| No | 44 | Blue | No |
| Yes | 19 | Red | No |
| No | 32 | Green | Yes |
| No | 14 | Blue | Y |

Step 1: we start from a leaf (the "initial prediction" for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create "probability" and use it to do the classification

The probability is estimated by the "Logistic Function", which is

$$Probability_{Yes} = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}} = \frac{e^{0.69}}{1 + e^{0.69}} = 0.67$$

So we got 0.67 as the probability of "loving Troll2" (or "Yes")

Normally, the probability of "Yes" can be calculated as 4/6 = 0.67

# How gradient booting works

The probability of “Yes” (initial guess)

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | Prob (Yes) |
|---|---|---|---|---|
| Yes | 12 | Blue | Yes | |
| Yes | 87 | Green | Yes | |
| No | 44 | Blue | No | 0.67 |
| Yes | 19 | Red | No | |
| No | 32 | Green | Yes | |
| No | 14 | Blue | Yes | |

Step 1: we start from a leaf (the “initial prediction” for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create “probability” and use it to do the classification

The probability is estimated by the “Logistic Function”, which is

$$Probability_{Yes} = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}} = \frac{e^{0.69}}{1 + e^{0.69}} = 0.67$$

So we got 0.67 as the probability of “loving Troll2” (or “Yes”)

## How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 |
|---|---|---|---|
| Yes | 12 | Blue | Yes |
| Yes | 87 | Green | Yes |
| No | 44 | Blue | No |
| Yes | 19 | Red | No |
| No | 32 | Green | Yes |
| No | 14 | Blue | Yes |

Step 1: we start from a leaf (the "initial prediction" for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create "probability" and use it to do the classification

We got 0.67 as the predicted probability of "loving Troll2"

Step 3: calculate Residual

# How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 |
|---|---|---|---|
| Yes | 12 | Blue | Yes |
| Yes | 87 | Green | Yes |
| No | 44 | Blue | No |
| Yes | 19 | Red | No |
| No | 32 | Green | Yes |
| No | 14 | Blue | Yes |

100%

Step 1: we start from a leaf (the "initial prediction" for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create "probability" and use it to do the classification

We got 0.67 as the predicted probability of "loving Troll2"

Step 3: calculate Residual

For example for the first sample

People loves Troll2 from the observation, so the observation is 100%
And the predicted probability is 0.67 Therefore the residual is:
100% - 67% = 33%

# How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | Residual |
|---|---|---|---|---|
| Yes | 12 | Blue | Yes | 0.3 |
| Yes | 87 | Green | Yes | 0.3 |
| No | 44 | Blue | No | -0.7 |
| Yes | 19 | Red | No | -0.7 |
| No | 32 | Green | Yes | 0.3 |
| No | 14 | Blue | Yes | 0.3 |

100% - 67% = 33%

Step 1: we start from a leaf (the “initial prediction” for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create “probability” and use it to do the classification

We got 0.67 as the predicted probability of “loving Troll2”

Step 3: calculate Residual

For example for the first sample

People loves Troll2 from the observation, so the observation is 100%
And the predicted probability is 0.67 Therefore the residual is:
100% - 67% = 33%

By doing this over all samples, we can have the Residuals column as the left

How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | Residual |
|---|---|---|---|---|
| Yes | 12 | Blue | Yes | 0.3 |
| Yes | 87 | Green | Yes | 0.3 |
| No | 44 | Blue | No | -0.7 |
| Yes | 19 | Red | No | -0.7 |
| No | 32 | Green | Yes | 0.3 |
| No | 14 | Blue | Yes | 0.3 |

100% - 67% = 33%

Step 1: we start from a leaf (the "initial prediction" for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create "probability" and use it to do the classification

We got 0.67 as the predicted probability of "loving Troll2"

Step 3: calculate Residual

For example for the first sample

People loves Troll2 from the observation, so the observation is 100%
And the predicted probability is 0.67 Therefore the residual is:
100% - 67% = 33%

By doing this over all samples, we can have the Residuals column as the left

The purpose of gradient boosting is to grow trees that gives the smallest "Residuals"

## How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | Residual |
|---|---|---|---|---|
| Yes | 12 | Blue | Yes | 0.3 |
| Yes | 87 | Green | Yes | 0.3 |
| No | 44 | Blue | No | -0.7 |
| Yes | 19 | Red | No | -0.7 |
| No | 32 | Green | Yes | 0.3 |
| No | 14 | Blue | Yes | 0.3 |

Step 1: we start from a leaf (the "initial prediction" for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create "probability" and use it to do the classification

We got 0.67 as the predicted probability of "loving Troll2"

Step 3: calculate Residual

Step 4: Build the first tree and predict the Residual

# How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | Residual |
| --- | --- | --- | --- | --- |
| Yes | 12 | Blue | Yes | 0.3 |
| Yes | 87 | Green | Yes | 0.3 |
| No | 44 | Blue | No | -0.7 |
| Yes | 19 | Red | No | -0.7 |
| No | 32 | Green | Yes | 0.3 |
| No | 14 | Blue | Yes | 0.3 |

Step 1: we start from a leaf (the "initial prediction" for all samples)

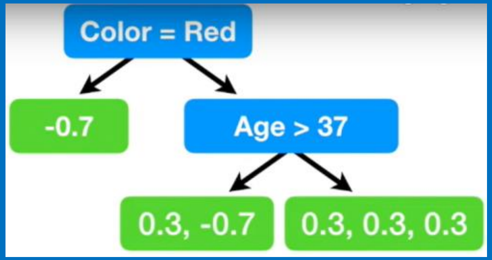
The first/original leaf is **Log(odds) = 0.69**

Step 2: Create "probability" and use it to do the classification

We got 0.67 as the predicted probability of "loving Troll2"

Step 3: calculate Residual

Step 4: Build the first tree and predict the Residual




Following the regular tree building process, we can have the tree as left

## How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | Residual |
|---|---|---|---|---|
| Yes | 12 | Blue | Yes | 0.3 |
| Yes | 87 | Green | Yes | 0.3 |
| No | 44 | Blue | No | -0.7 |
| Yes | 19 | Red | No | -0.7 |
| No | 32 | Green | Yes | 0.3 |
| No | 14 | Blue | Yes | 0.3 |

Step 1: we start from a leaf (the "initial prediction" for all samples)

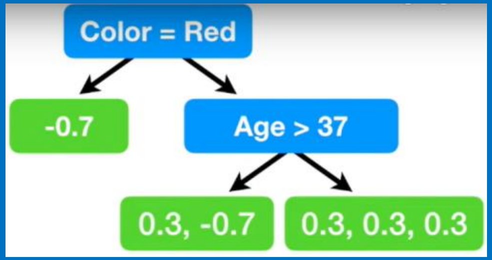
The first/original leaf is Log(odds) = 0.69

Step 2: Create "probability" and use it to do the classification

We got 0.67 as the predicted probability of "loving Troll2"

Step 3: calculate Residual

Step 4: Build the first tree and predict the Residual




Following the regular tree building process, we can have the tree as left

Step 5: Combine the tree and the first/original leaf to predict "log(odds)"

# How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | Residual |
|---|---|---|---|---|
| Yes | 12 | Blue | Yes | 0.3 |
| Yes | 87 | Green | Yes | 0.3 |
| No | 44 | Blue | No | -0.7 |
| Yes | 19 | Red | No | -0.7 |
| No | 32 | Green | Yes | 0.3 |
| No | 14 | Blue | Yes | 0.3 |

Step 1: we start from a leaf (the "initial prediction" for all samples)

The first/original leaf is **Log(odds) = 0.69**

Step 2: Create "probability" and use it to do the classification

We got 0.67 as the predicted probability of "loving Troll2"

Step 3: calculate Residual

Step 4: Build the first tree and predict the Residual


Following the regular tree building process, we can have the tree as left

Step 5: Combine the tree and the first/original leaf to predict "log(odds)"

Let's look at the example for the 2nd and 3rd sample, which go through the three like above

# How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | Residual |
|---|---|---|---|---|
| Yes | 12 | Blue | Yes | 0.3 |
| Yes | 87 | Green | Yes | 0.3 |
| No | 44 | Blue | No | -0.7 |
| Yes | 19 | Red | No | -0.7 |
| No | 32 | Green | Yes | 0.3 |
| No | 14 | Blue | Yes | 0.3 |

Step 1: we start from a leaf (the "initial prediction" for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create "probability" and use it to do the classification

We got 0.67 as the predicted probability of "loving Troll2"

Step 3: calculate Residual

Step 4: Build the first tree and predict the Residual


Following the regular tree building process, we can have the tree as left

However, here the output for the tree is "Residual", we need to convert it to "log(odds)"

Step 5: Combine the tree and the first/original leaf to predict "log(odds)"

Let's look at the example for the 2nd and 3rd sample, which go through the three like above

The "log(odds)" is calculated based on:

$$\frac{\sum \text{Residual}_i}{\sum [\text{Previous Probability}_i \times (1 - \text{Previous Probability}_i)]}$$

# How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | Residual |
|---|---|---|---|---|
| Yes | 12 | Blue | Yes | 0.3 |
| Yes | 87 | Green | Yes | 0.3 |
| No | 44 | Blue | No | -0.7 |
| Yes | 19 | Red | No | -0.7 |
| No | 32 | Green | Yes | 0.3 |
| No | 14 | Blue | Yes | 0.3 |

Step 1: we start from a leaf (the “initial prediction” for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create “probability” and use it to do the classification

We got 0.67 as the predicted probability of “loving Troll2”

Step 3: calculate Residual

Step 4: Build the first tree and predict the Residual


Following the regular tree building process, we can have the tree as left

Step 5: Combine the tree and the first/original leaf to predict “log(odds)”

Let’s look at the example for the 2nd and 3rd sample, which go through the three like above

The prediction is calculated base 0.3 + (-0.7)

$$\frac{\sum \text{Residual}_i}{\sum\left[\text{Previous Probability}_i \times (1 - \text{Previous Probability}_i)\right]}$$

0.67 1- 0.67

# How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | Residual |
|---|---|---|---|---|
| Yes | 12 | Blue | Yes | 0.3 |
| Yes | 87 | Green | Yes | 0.3 |
| No | 44 | Blue | No | -0.7 |
| Yes | 19 | Red | No | -0.7 |
| No | 32 | Green | Yes | 0.3 |
| No | 14 | Blue | Yes | 0.3 |

Step 1: we start from a leaf (the “initial prediction” for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create “probability” and use it to do the classification

We got 0.67 as the predicted probability of “loving Troll2”

Step 3: calculate Residual

Step 4: Build the first tree and predict the Residual


Following the regular tree building process, we can have the tree as left

Step 5: Combi the tree and the first/original leaf to predict “log(odds)”

-1.0

Let’s loo he example for the 2nd and 3rd sample, which go through the three like above

# How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | Residual |
|---|---|---|---|---|
| Yes | 12 | Blue | Yes | 0.3 |
| Yes | 87 | Green | Yes | 0.3 |
| No | 44 | Blue | No | -0.7 |
| Yes | 19 | Red | No | -0.7 |
| No | 32 | Green | Yes | 0.3 |
| No | 14 | Blue | Yes | 0.3 |

Step 1: we start from a leaf (the “initial prediction” for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create “probability” and use it to do the classification

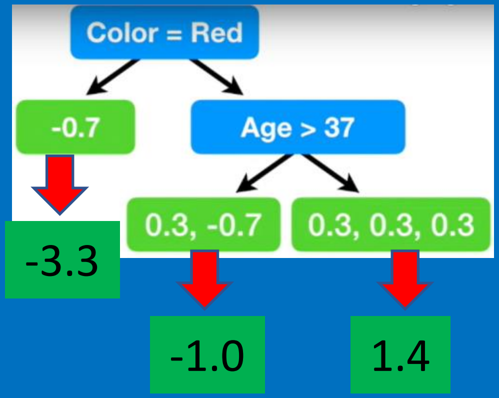
We got 0.67 as the predicted probability of “loving Troll2”

Step 3: calculate Residual

Step 4: Build the first tree and predict the Residual

Step 5: Combine the tree and the first/original leaf to predict “log(odds)”

Similarly, we can postprocess the tree and get the log(odds) prediction as below



# How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | Residual |
|---|---|---|---|---|
| Yes | 12 | Blue | Yes | 0.3 |
| Yes | 87 | Green | Yes | 0.3 |
| No | 44 | Blue | No | -0.7 |
| Yes | 19 | Red | No | -0.7 |
| No | 32 | Green | Yes | 0.3 |
| No | 14 | Blue | Yes | 0.3 |

Step 1: we start from a leaf (the "initial prediction" for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create "probability" and use it to do the classification

We got 0.67 as the predicted probability of "loving Troll2"

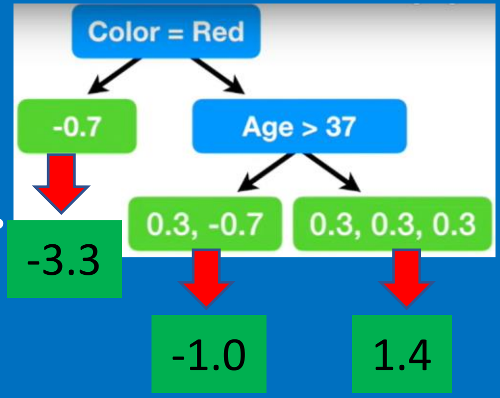
Step 3: calculate Residual

Step 4: Build the first tree and predict the Residual

Step 5: Combine the tree and the first/original leaf to predict "log(odds)"

Log(odds) = 0.69 + 0.8

*Original leaf* *Learning rate*



# How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | Residual |
|---|---|---|---|---|
| Yes | 12 | Blue | Yes | 0.3 |
| Yes | 87 | Green | Yes | 0.3 |
| No | 44 | Blue | No | -0.7 |
| Yes | 19 | Red | No | -0.7 |
| No | 32 | Green | Yes | 0.3 |
| No | 14 | Blue | Yes | 0.3 |

Step 1: we start from a leaf (the "initial prediction" for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create "probability" and use it to do the classification

We got 0.67 as the predicted probability of "loving Troll2"

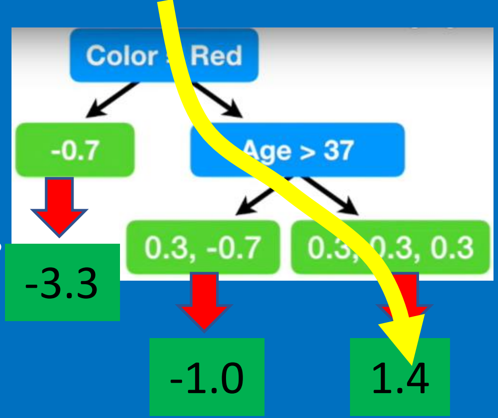
Step 3: calculate Residual

Step 4: Build the first tree and predict the Residual

Step 5: Combine the tree and the first/original leaf to predict "log(odds)"

Log(odds) = 0.69 + 0.8

*Original leaf* *Learning rate*




So for the first sample, we can have the predicted Log(odds) as 0.69 + 0.8 x 1.4 = 1.8

# How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | Residual |
|---|---|---|---|---|
| Yes | 12 | Blue | Yes | 0.3 |
| Yes | 87 | Green | Yes | 0.3 |
| No | 44 | Blue | No | -0.7 |
| Yes | 19 | Red | No | -0.7 |
| No | 32 | Green | Yes | 0.3 |
| No | 14 | Blue | Yes | 0.3 |

Log(odds) = 1.8

Step 1: we start from a leaf (the “initial prediction” for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create “probability” and use it to do the classification

We got 0.67 as the predicted probability of “loving Troll2”

Step 3: calculate Residual

Step 4: Build the first tree and predict the Residuals

Step 5: Combine the tree and the first/original leaf to predict “log(odds)”

So for the first sample, we can have the predicted Log(odds) as 0.69 + 0.8 x 1.4 = 1.8

# How gradient booting works

Step 1: we start from a leaf (the “initial prediction” for all samples)

The first/original leaf is Log(odds) = 0.69

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | Residual |
|---|---|---|---|---|
| Yes | 12 | Blue | Yes | 0.3 |
| Yes | 87 | Green | Yes | 0.3 |
| No | 44 | Blue | No | -0.7 |
| Yes | 19 | Red | No | -0.7 |
| No | 32 | Green | Yes | 0.3 |
| No | 14 | Blue | Yes | 0.3 |

Log(odds) = 1.8

Step 2: Create “probability” and use it to do the classification

We got 0.67 as the predicted probability of “loving Troll2”

Step 3: calculate Residual

Step 4: Build the first tree and predict the Residuals

Step 5: Combine the tree and the first/original leaf to predict “log(odds)”

So for the first sample, we can have the predicted Log(odds) as 0.69 + 0.8 x 1.4 = 1.8

Following this, we can predict “log(odds)” for all other samples

# How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | Residual |
|---|---|---|---|---|
| Yes | 12 | Blue | Yes | 0.3 |
| Yes | 87 | Green | Yes | 0.3 |
| No | 44 | Blue | No | -0.7 |
| Yes | 19 | Red | No | -0.7 |
| No | 32 | Green | Yes | 0.3 |
| No | 14 | Blue | Yes | 0.3 |

Log(odds) = 1.8

Step 1: we start from a leaf (the "initial prediction" for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create "probability" and use it to do the classification

We got 0.67 as the predicted probability of "loving Troll2"

Step 3: calculate Residual

Step 4: Build the first tree and predict the Residuals

Step 5: Combine the tree and the first/original leaf to predict "log(odds)"

So for the first sample, we can have the predicted Log(odds) as 0.69 + 0.8 x 1.4 = 1.8

Following this, we can predict "log(odds)" for all other samples

Step 6: Convert the predicted "log(odds)" to probability

# How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | Residual |
|---|---|---|---|---|
| Yes | 12 | Blue | Yes | 0.3 |
| Yes | 87 | Green | Yes | 0.3 |
| No | 44 | Blue | No | -0.7 |
| Yes | 19 | Red | No | -0.7 |
| No | 32 | Green | Yes | 0.3 |
| No | 14 | Blue | Yes | 0.3 |

Step 1: we start from a leaf (the “initial prediction” for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create “probability” and use it to do the classification

Log(odds) = 1.8

P=0.9

We got 0.67 as the predicted probability of “loving Troll2”

Step 3: calculate Residual

Step 4: Build the first tree and predict the Residuals

Step 5: Combine the tree and the first/original leaf to predict “log(odds)”

So for the first sample, we can have the predicted Log(odds) as 0.69 + 0.8 x 1.4 = 1.8

Following this, we can predict “log(odds)” for all other samples

Step 6: Convert the predicted “log(odds)” to probability

For example, for the first sample,

We can convert log(odds) to probability as $\frac{e^{1.8}}{1+e^{1.8}} = 0.9$

# How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | First prediction | Update prediction 1 |
|---|---|---|---|---|---|
| Yes | 12 | Blue | Yes | 0.67 | 0.9 |
| Yes | 87 | Green | Yes | | |
| No | 44 | Blue | No | | |
| Yes | 19 | Red | No | | |
| No | 32 | Green | Yes | | |
| No | 14 | Blue | Yes | | |

Step 1: we start from a leaf (the "initial prediction" for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create "probability" and use it to do the classification

We got 0.67 as the predicted probability of "loving Troll2"

Step 3: calculate Residual

Step 4: Build the first tree and predict the Residuals

Step 5: Combine the tree and the first/original leaf to predict "log(odds)"

Step 6: Convert the predicted "log(odds)" to probability

For example, for the first sample,

We can convert log(odds) to probability as $\frac{e^{1.8}}{1+e^{1.8}} = 0.9$

# How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | First prediction | Update prediction 1 |
|---|---|---|---|---|---|
| Yes | 12 | Blue | Yes | 0.67 | 0.9 |
| Yes | 87 | Green | Yes | | |
| No | 44 | Blue | No | | |
| Yes | 19 | Red | No | | |
| No | 32 | Green | Yes | | |
| No | 14 | Blue | Yes | | |

The updated prediction (0.9) is clearly better than the original one (0.67)

Step 1: we start from a leaf (the "initial prediction" for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create "probability" and use it to do the classification

We got 0.67 as the predicted probability of "loving Troll2"

Step 3: calculate Residual

Step 4: Build the first tree and predict the Residuals

Step 5: Combine the tree and the first/original leaf to predict "log(odds)"

Step 6: Convert the predicted "log(odds)" to probability

For example, for the first sample,

We can convert log(odds) to probability as $\frac{e^{1.8}}{1+e^{1.8}} = 0.9$

# How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | First prediction | Update prediction 1 |
|---|---|---|---|---|---|
| Yes | 12 | Blue | Yes | 0.67 | 0.9 |
| Yes | 87 | Green | Yes | | 0.5 |
| No | 44 | Blue | No | | 0.5 |
| Yes | 19 | Red | No | | 0.1 |
| No | 32 | Green | Yes | | 0.9 |
| No | 14 | Blue | Yes | | 0.9 |

Step 1: we start from a leaf (the "initial prediction" for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create "probability" and use it to do the classification

We got 0.67 as the predicted probability of "loving Troll2"

Step 3: calculate Residual

Step 4: Build the first tree and predict the Residuals

Step 5: Combine the tree and the first/original leaf to predict "log(odds)"

Step 6: Convert the predicted "log(odds)" to probability

For example, for the first sample,

We can convert log(odds) to probability as $\frac{e^{1.8}}{1+e^{1.8}} = 0.9$

We do these for all the samples, and can get the updated predictions for all of them

# How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | First prediction | Update prediction 1 |
|---|---|---|---|---|---|
| Yes | 12 | Blue | Yes | 0.67 | 0.9 |
| Yes | 87 | Green | Yes | | 0.5 |
| No | 44 | Blue | No | | 0.5 |
| Yes | 19 | Red | No | | 0.1 |
| No | 32 | Green | Yes | | 0.9 |
| No | 14 | Blue | Yes | | 0.9 |

Step 1: we start from a leaf (the "initial prediction" for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create "probability" and use it to do the classification

We got 0.67 as the predicted probability of "loving Troll2"

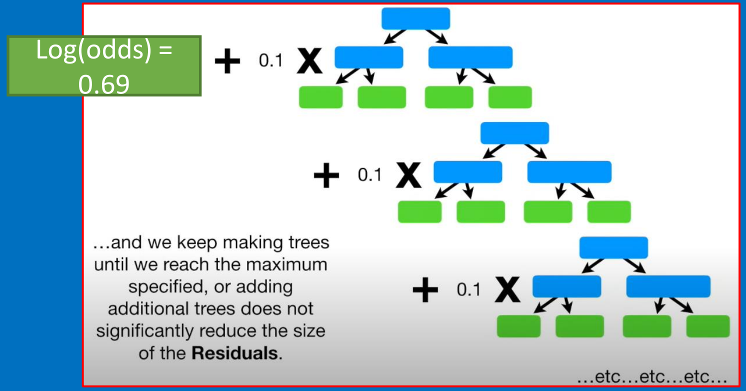
Step 3: calculate Residual

Step 4: Build the first tree and predict the Residuals

Step 5: Combine the tree and the first/original leaf to predict "log(odds)"

Step 6: Convert the predicted "log(odds)" to probability

Step 7: We then repeat Step 3-6 with the updated predictions, until the calculated residuals do not change much



# How gradient booting works

| Likes Popcorn | Age | Favorite Color | Loves Troll 2 | First prediction | Update prediction 1 |
|---|---|---|---|---|---|
| Yes | 12 | Blue | Yes | | 0.9 |
| Yes | 87 | Green | Yes | | 0.5 |
| No | 44 | Blue | No | 0.67 | 0.5 |
| Yes | 19 | Red | No | | 0.1 |
| No | 32 | Green | Yes | | 0.9 |
| No | 14 | Blue | Yes | | 0.9 |

Step 1: we start from a leaf (the "initial prediction" for all samples)

The first/original leaf is Log(odds) = 0.69

Step 2: Create "probability" and use it to do the classification

We got 0.67 as the predicted probability of "loving Troll2"

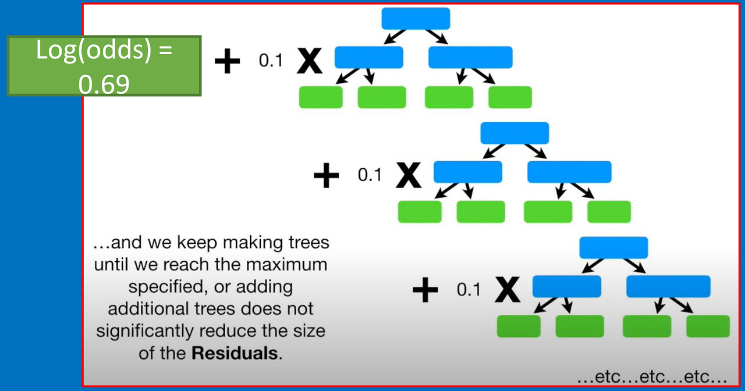
Step 3: calculate Residual

Step 4: Build the first tree and predict the Residuals

Step 5: Combine the tree and the first/original leaf to predict "log(odds)"

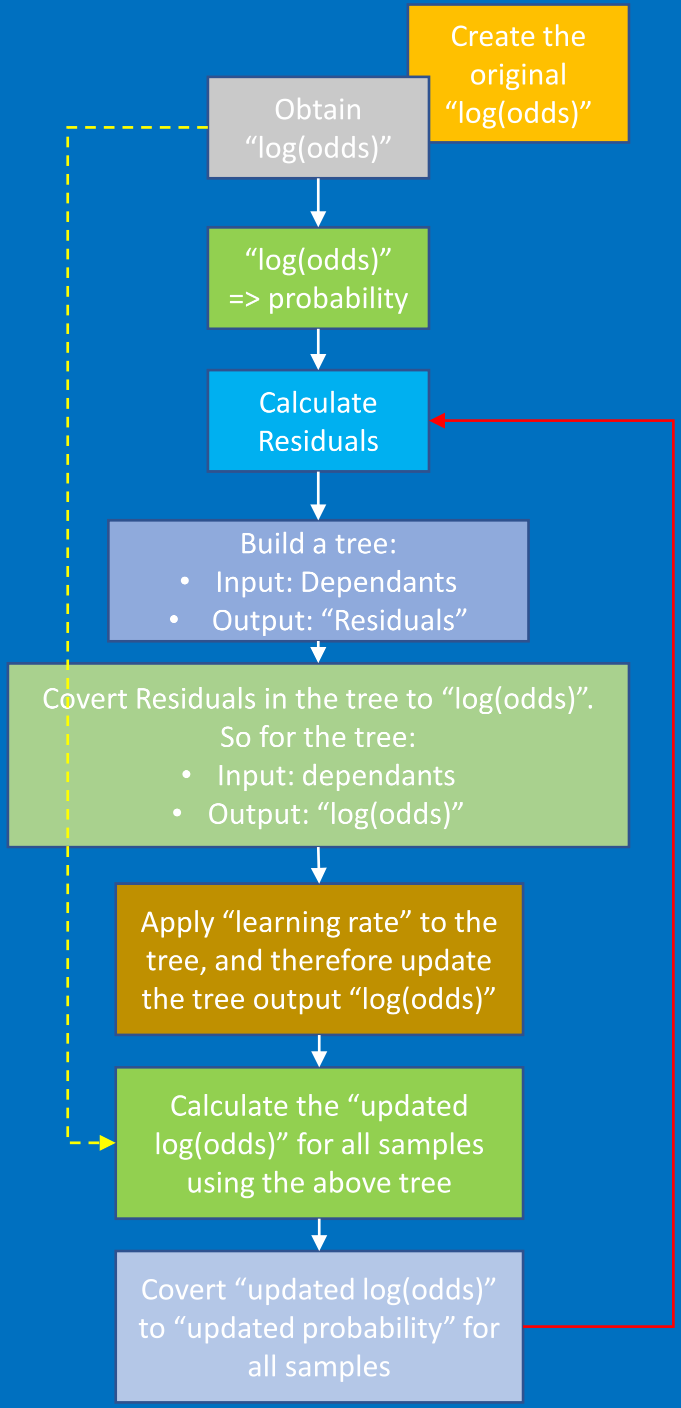
Step 6: Convert the predicted "log(odds)" to probability

Step 7: We then repeat Step 3-6 with the updated predictions, until the calculated residuals do not change much




In prediction, we just need to use the test data to go through all the trees and add them up, the results are predicted Log(odds), which can be converted to probability and indicate the classified category

# Summary




## For example