



# DAY 23 — Gradient Boosting

## 1 Why Gradient Boosting Exists

Random Forest:

- Trees are **independent**
- Errors are averaged out
- Strong baseline

But...

Trees do NOT learn from previous mistakes

Hard examples stay hard

**Gradient Boosting fixes this**

## 2 What is Gradient Boosting?

**Gradient Boosting builds trees Sequentially.**

**Each new tree focuses on correcting the mistakes of the previous trees.**

Think like this:

1. First model → makes mistakes
2. Second model → learns those mistakes
3. Third model → learns remaining mistakes
4. Final model = sum of all models

This is why it's called **boosting**

## 3 Key Difference: Random Forest vs Gradient Boosting

Concept	Random Forest	Gradient Boosting
Tree training	Parallel	Sequential
Learning	Independent	Error-correcting
Overfitting	Less	More risk

Power	Medium	Very high
Speed	Faster	Slower

## 4 How Gradient Boosting Works

### Step-by-step (Regression example)

1. Predict mean of target
2. Calculate errors (residuals)
3. Train tree on residuals
4. Add tree predictions to previous model
5. Repeat

Final Prediction:

$$\text{Prediction} = \text{Tree1} + \text{Tree2} + \text{Tree3} + \dots$$

📌 Each tree is **small and weak**, but together they are powerful.

## 5 Learning Rate

### What is Learning Rate?

It controls **how much each tree contributes**.

Low learning rate → slow but accurate  
 High learning rate → fast but risky

📌 **Small steps = better generalization**

Typical values:

0.01, 0.05, 0.1

## 6 Number of Estimators (Trees)

Trees	Effect
Few	Underfitting
Many	Powerful but slow

✖ Learning rate & number of trees **must be balanced**.

---

## 7 Tree Depth

Gradient Boosting uses:

- **Shallow trees** (depth 1-3)
- Called **weak learners**

Why?

- Prevents overfitting
  - Forces gradual learning
- 

## 8 Gradient Boosting for Classification

Instead of predicting values:

- Predicts **probabilities**
- Uses **log loss**

Final decision:

Probability > threshold → class 1

## 9 Overfitting in Gradient Boosting

Gradient Boosting can:

- Overfit easily
- Memorize noise

How to control it:

- Small `learning_rate`
  - Small `max_depth`
  - Fewer estimators
  - Early stopping (later)
-