XGBoost: A Beginner's Tutorial

EV Battery Life Estimation Example

What is XGBoost?

- Gradient Boosting = Sequentially improve weak learners
- XGBoost = Efficient, regularized implementation
- Combines prediction trees using gradient descent principles
- Excellent for tabular data (e.g. EV sensor data)

Example: EV Battery Life Estimation

Features:

- X_1: Charge cycles
- X_2: Average temperature (°C)
- X_3: Battery age (years)

Data:

Sample	X_1	X_2	X_3
1	500	35	2
2	600	30	3
3	700	40	4

Target: $y = [85, 80, 75]$

Step 1: Initial Prediction

$$\hat{y}^{(0)} = \frac{1}{3}(85 + 80 + 75) = 80$$

$$\hat{y}^{(0)} = \begin{bmatrix} 80 \\ 80 \\ 80 \end{bmatrix}$$

Step 2: Compute Gradients and Hessians

Squared Error Loss:

$$L = (y - \hat{y})^2$$

Gradient:

$$g_i = \hat{y}_i - y_i \quad \Rightarrow \quad \begin{bmatrix} -5 \\ 0 \\ 5 \end{bmatrix}$$

Hessian:

$$h_i = 1 \quad \Rightarrow \quad \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Step 3: Try Tree Split

Split on $X_1 < 600$:

- Left Node: Sample 1 $\rightarrow g = -5,\ h = 1$
- Right Node: Samples 2, 3 $\rightarrow g = 5,\ h = 2$

$$\text{Gain} = \frac{1}{2}\left[\frac{(-5)^2}{1+1} + \frac{5^2}{2+1} - \frac{(0)^2}{3+1}\right] = 10.42$$

Step 4: Compute Leaf Weights

$$w_j = -\frac{\sum g_j}{\sum h_j + \lambda}$$

- Left Leaf: $w = \frac{5}{2} = 2.5$
- Right Leaf: $w = \frac{-5}{3} \approx -1.67$

Learning rate: $\eta = 0.1$

$$\hat{y}^{(1)} = \hat{y}^{(0)} + \eta \cdot f_1(x) \Rightarrow \begin{bmatrix} 80.25 \\ 79.83 \\ 79.83 \end{bmatrix}$$

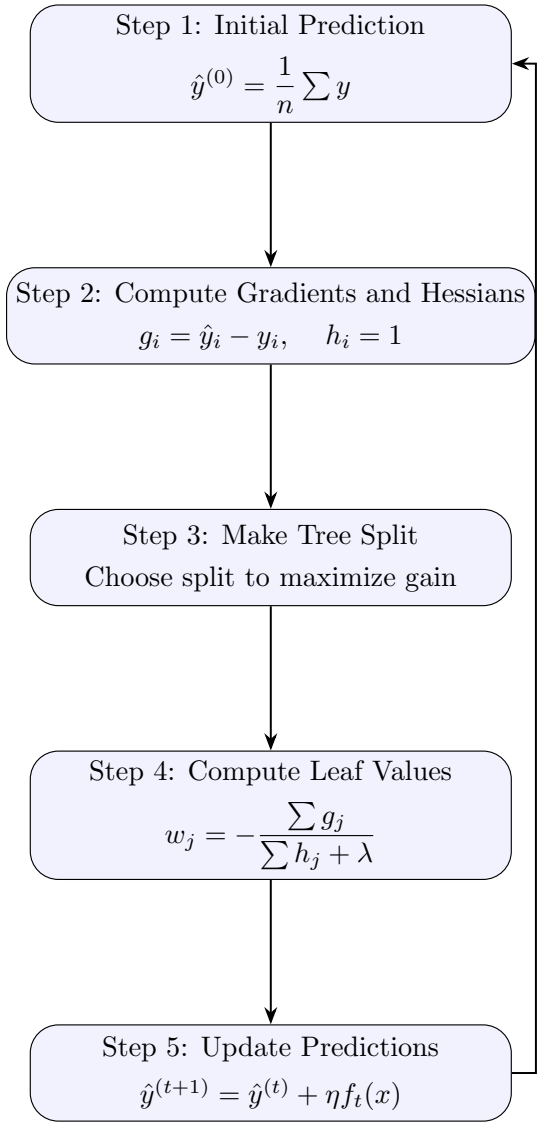
Summary of Step 1

Sample	True y	Init $\hat{y}$	Update	New $\hat{y}$
1	85	80	$+0.25$	80.25
2	80	80	-0.167	79.83
3	75	80	-0.167	79.83

Each tree improves on previous predictions using gradient-based splits.

XGBoost Flowchart

XGBoost Flowchart: EV Battery Life Estimation



Key Takeaways

- XGBoost builds trees sequentially using gradient descent.
- Uses Taylor expansion: gradient + Hessian.
- Learns leaf values analytically from gradients.
- Extremely effective on tabular prediction tasks.