Gradient Boosting for Classification

Let's use a synthetic dataset to predict **student exam results** (**Pass = 1**, Fail = 0) using **hours studied** and **previous exam grade**. We'll apply gradient boosting with **log loss** (binary cross-entropy) and a **learning rate** of $\gamma = 0.1$.

Dataset

Hours Studied	Previous Grade	Pass (Target)
2	65	0
3	80	1
4	85	1

Goal: Predict "Pass" using gradient boosting (2 iterations, tree depth = 1).

Key Concepts for Classification

1. **Initial Prediction**: Log-odds of the target class (e.g., log-odds = $\ln\left(\frac{\text{Pass count}}{\text{Fail count}}\right)$).
2. **Probability Conversion**: Use the logistic function $p = \frac{1}{1+e^{-F(x)}}$.
3. **Residuals (Gradients)**: $r_i = y_i - p_i$ (difference between actual and predicted probability).
4. **Tree Outputs**: For each leaf, compute Output = $\frac{\sum \text{residuals}}{\sum p_i(1-p_i)}$.
5. **Update Model**: Add tree outputs scaled by γ to log-odds.

Step 1: Initial Model (F_0)

- **Log-odds initialization**: Pass count = 2, Fail count = 1. $F_0(x) = \ln\left(\frac{2}{1}\right) \approx 0.693$. All samples start with $F_0 = 0.693$.
- **Initial probabilities** (via logistic function):

$$p_0 = \frac{1}{1 + e^{-0.693}} \approx 0.667 \quad \text{for all samples.}$$

- **Residuals** ($r_i = y_i - p_i$):

Residuals	-0.667	0.333	0.333

Step 2: First Tree (h_1)

Fit a regression tree to predict residuals. Assume the best split is **Hours Studied 2.5**:

- **Left leaf** (Sample 1: Hours = 2): Residual $= -0.667$. Output $= \frac{-0.667}{p(1-p)} = \frac{-0.667}{0.667 \times 0.333} \approx -3.0$. Scaled by $\gamma = 0.1$: $-3.0 \times 0.1 = -0.3$.
- **Right leaf** (Samples 2 & 3: Hours ¿ 2.5): Residuals $= 0.333 + 0.333 = 0.666$. Output $= \frac{0.666}{0.667 \times 0.333 \times 2} \approx 1.5$. Scaled by $\gamma = 0.1$: $1.5 \times 0.1 = 0.15$.
- **Updated log-odds** ($F_1 = F_0$+tree outputs): Sample 1: $0.693 - 0.3 = 0.393$. Samples 2 & 3: $0.693 + 0.15 = 0.843$.
- **New probabilities** (p_1): Sample 1: $\frac{1}{1+e^{-0.393}} \approx 0.596$. Samples 2 & 3: $\frac{1}{1+e^{-0.843}} \approx 0.699$.

- **New residuals**:

Residuals	-0.596	0.301	0.301

Step 3: Second Tree (h_2)

Fit a tree to updated residuals. Assume the best split is **Previous Grade 75**:

- **Left leaf** (Sample 1: Grade $= 65$): Residual $= -0.596$. Output $= \frac{-0.596}{0.596 \times 0.404} \approx -2.47$. Scaled by $\gamma = 0.1$: $-2.47 \times 0.1 = -0.247$.
- **Right leaf** (Samples 2 & 3: Grade ¿ 75): Residuals $= 0.301 + 0.301 = 0.602$. Output $= \frac{0.602}{0.699 \times 0.301 \times 2} \approx 1.43$. Scaled by $\gamma = 0.1$: $1.43 \times 0.1 = 0.143$.
- **Updated log-odds** ($F_2 = F_1 +$tree outputs): Sample 1: $0.393 - 0.247 = 0.146$. Samples 2 & 3: $0.843 + 0.143 = 0.986$.
- **Final probabilities** (p_2): Sample 1: $\frac{1}{1+e^{-0.146}} \approx 0.536$. Samples 2 & 3: $\frac{1}{1+e^{-0.986}} \approx 0.728$.
- **Final residuals**:

Residuals	-0.536	0.272	0.272

Key Observations

1. **Residuals Decrease**: Errors shrink with each iteration (e.g., Sample 1's residual improved from -0.667 to -0.536).
2. **Probability Adjustment**: The model shifts probabilities toward actual labels (e.g., Sample 2's probability increased from 0.667 to 0.728).

3. **Feature Importance**: The first tree used **hours studied**, and the second used **previous grade**.

How Gradient Boosting Works for Classification

1. **Initial Log-Odds**: Start with the log-odds of the target class.
2. **Residual Calculation**: Compute residuals as $y_i - p_i$.
3. **Tree Fitting**: Trees predict residuals, with outputs scaled by $\frac{\sum \text{residuals}}{\sum p_i(1-p_i)}$.
4. **Model Update**: Log-odds are adjusted by tree outputs multiplied by γ.
5. **Iteration**: Repeat until residuals are minimized.

Conclusion

After two iterations, gradient boosting improves predictions by correcting residuals using weak trees. The final probabilities (0.536, 0.728, 0.728) better reflect the true labels (0, 1, 1) than the initial guess (0.667 for all). With more trees, the model would further refine predictions.