XGBoost Part1:

XGBoost trees for regression:

The most common way to build XGBoost tree for Regression

1. Initial prediction
2. Residuals
3. All residuals go to a leaf
4. Choose a split
5. $$\text{Similarity Score} = \frac{\text{Sum of Residuals, Squared}}{\text{Number of Residuals} + \lambda}$$
6. $$\text{Gain} = \text{Left}_{\text{Similarity}} + \text{Right}_{\text{Similarity}} - \text{Root}_{\text{Similarity}}$$
7. Find the best split that gives the most gain
8. $$\text{Gain} - \gamma = \begin{cases} \text{If positive, then do not prune.} \\ \text{If negative, then prune.} \end{cases}$$
9. $$\text{Output Value} = \frac{\text{Sum of Residuals}}{\text{Number of Residuals} + \lambda}$$
10. When Regularization parameter > 0 results in more pruning

XGBoost trees for classification:

1. Initial prediction
2. Residuals
3. All residuals go to a leaf
4. Choose a split

5. $$\frac{\left(\sum \text{Residual}_i\right)^2}{\sum \left[\text{Previous Probability}_i \times (1 - \text{Previous Probability}_i)\right] + \lambda}$$
6. $$\text{Gain} = \text{Left}_{\text{Similarity}} + \text{Right}_{\text{Similarity}} - \text{Root}_{\text{Similarity}}$$
7. Find the best split that gives the most gain
8. $$\text{Gain} - \gamma = \begin{cases} \text{If positive, then do not prune.} \\ \text{If negative, then prune.} \end{cases}$$
9. Cover = Number of Residuals in Regression and p(1-p) in Classification
10. $$\frac{\left(\sum \text{Residual}_i\right)}{\sum \left[\text{Previous Probability}_i \times (1 - \text{Previous Probability}_i)\right] + \lambda}$$
11. F(x0) = $\log\left(\frac{p}{1-p}\right) = \log(\text{odds})$
12. P = $\frac{e^{\log(odds)}}{1+e^{\log(odds)}}$
13. When Regularization parameter > 0 results in more pruning

XGBoost Mathematical Details:

1. Initialize predicted value
2. Calculate residuals
3. Putting all of the residual in a single leaf
4. Objective Function:

$$\boxed{\left[\sum_{i=1}^{n} L(y_i, p_i^0 + O_{value})\right] + \frac{1}{2}\lambda O_{value}^2}$$

	Regression		Classification
Similarity Score =	$\frac{\text{Sum of Residuals, Squared}}{\text{Number of Residuals} + \lambda}$	VS	$\frac{(\sum \text{Residual}_i)^2}{\sum [P_i \times (1 - P_i) + \lambda]}$
Output Value =	$\frac{\text{Sum of Residuals}}{\text{Number of Residuals} + \lambda}$	VS	$\frac{(\sum \text{Residual}_i)}{\sum [P_i \times (1 - P_i) + \lambda]}$

$$\text{Similarity Score} = \frac{(g_1 + g_2 + \cdots + g_n)^2}{(h_1 + h_2 + \cdots + h_n + \lambda)}$$

$$O_{value} = \frac{-(g_1 + g_2 + \cdots + g_n)}{(h_1 + h_2 + \cdots + h_n + \lambda)}$$

$$\text{Cover} = h_1 + h_2 + \cdots + h_n \boxed{= \text{Number of Residuals}}$$

$$\text{Cover} = h_1 + h_2 + \cdots + h_n \boxed{= \sum [p_i \times (1 - p_i)]}$$

XGBoost Crazy cool optimizations:

Approximation Greedy Algorithm

Parallel Learning

Weighted Quantile Sketch

Sparsity-Aware Split Finding

Cache-Aware Access

Blocks for Out-of-Core Computation