GD Regression:

Step 0: The data and the loss function

Input Data $\{(x_i, y_i)\}_{i=1}^{n}$ and a differentiable Loss Function $L(y_i, F(x))$

- x = Features
- y = Target
- n = Number of rows
- y_i = Observed Value (Target)
- F(x) = Predicted Value (Target)

Loss Function = $\frac{1}{2}\ (Observed - Predicted)^2$

$\frac{d}{d\ Predicted}\ \frac{1}{2}\ (Observed - Predicted)^2$ = $-(Observed - Predicted)$

Step 1: Initialize model with a constant value:

$F_0(x)$ = $argmin_\gamma \sum_{i=1}^{n} L(y_i, \gamma)$

- γ = Predicted Value (Target)
- argmin over gamma means we need to find a Predicted Value that minimizes this sum
- $F_0(x)$ = Initial Predicted Value that predicts that all sample will equal $F_0(x)$
- Given this Loss Function, $F_0(x)$ = Average of all the Observed Values

Step 2: Built M trees

For m = 1 to M

- m = index tree
- M = number of trees

(A) Calculate Residuals

Compute $r_{im} = -[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}]_{F(x)=F_{m-1}(x)} \; For \; i = 1, \ldots, n$

- $[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}]_{F(x)=F_{m-1}(x)}$ = the derivative of the Loss Function
- r = residual
- i = sample number
- r_{im} = residual for each sample in a specific tree (pseudo residual)

r_{im} = (Observed - Predicted)

(B) Fit a regression tree to the residuals

Fit a regression tree to the r_{im} values and create terminal region R_{jm} for j = 1…J_m

- j = index leaf
- J_m = Total number of leaves
- R_{jm} = Leaves

(C) Optimize leaf output values

For j = 1…J_m compute $\gamma_{jm} = argmin_\gamma \sum_{x \epsilon R_{jm}} L(y_i, F_{m-1}(x) + \gamma)$

$\gamma_{jm} = argmin_\gamma \sum_{x \epsilon R_{jm}} \frac{1}{2} (y_i - (F_{m-1}(x) + \gamma))^2$

The Output Values for γ_{jm} is always the average of the Residuals that end up in the same leaf

(D) Update predictions with the new tree

Update $F_{(m)}(x) = F_{m-1}(x) + \eta \sum_{j=1}^{J_m} \gamma_{jm} I(x \epsilon R_{jm})$

Step 3: Output the final prediction

Output $F_M(x) = F_0(x) + \eta \sum_{j=1}^{J_m} \gamma_{j1} I(x \epsilon R_{j1}) + \eta \sum_{j=1}^{J_m} \gamma_{j2} I(x \epsilon R_{j2}) + \eta \sum_{j=1}^{J_m} \gamma_{jM} I(x \epsilon R_{jM})$

GB Classification:

Step 0: The data and the loss function

Input Data $\{(x_i, y_i)\}_{i=1}^{n}$ and a differentiable Loss Function $L(y_i, F(x))$

- x = Features
- y = Target
- n = Number of rows
- y_i = Observed Value (Target)
- F(x) = Predicted Value (Target)

Log(likelihood) = $\sum_{i=1}^{N} y_i \ \log(p) + (1 - y_i) \ \log(1 - p)$

Log Loss = - Log(likelihood) or negative log(likelihood) =
$-\sum_{i=1}^{N} y_i \ \log(p) + (1 - y_i) \ \log(1 - p)$

odds = $\frac{P}{1-P}$

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

Loss Function =$-y_i \ \log(odds) + \ \log(1 + e^{\log(odds)})$

$\frac{d}{d \log(odds)} - y_i \ \log(odds) + \ \log(1 + e^{\log(odds)})$

= $-y_i + \frac{e^{\log(odds)}}{1+e^{\log(odds)}}$ or $-y_i + p$

Step 1: Initialize model with a constant value:

$F_0(x) = \ argmin_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$

- γ = log(odds) value
- argmin over gamma means we need to find a log(odds) that minimizes this sum
- $F_0(x)$ = Initial Predicted Value that predicts that all sample will equal $F_0(x)$
- Given this Loss Function, $F_0(x) = \log \frac{Number\ of\ Yes}{Number\ of\ No}$

Step 2: Built M trees

For m = 1 to M

- m = index tree
- M = number of trees

(A) Calculate Residuals

Compute $r_{im} = -[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}]_{F(x)=F_{m-1}(x)}\ For\ i = 1, \dots, n$

- $[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}]_{F(x)=F_{m-1}(x)}$ = the derivative of the Loss Function
- r = residual
- i = sample number
- r_{im} = residual for each sample in a specific tree (pseudo residual)

r_{im} = (Observed - p)

(B) Fit a regression tree to the residuals

Fit a regression tree to the r_{im} values and create terminal region R_{jm} for j = 1…J_m

- j = index leaf
- J_m = Total number of leaves
- R_{jm} = Leaves

(C) Optimize leaf output values

For j = 1…J_m compute $\gamma_{jm} = argmin_\gamma \sum_{x \epsilon R_{ij}} L(y_i, F_{m-1}(x) + \gamma)$

$\gamma_{jm} = argmin_\gamma \sum_{x \epsilon R_{ij}} -y_i \left[F_{m-1}(x_i) + \gamma\right] + \log(1 + e^{F_{m-1}(x_i)+\gamma})$

$\gamma_{jm} = \frac{\sum_{j=1}^{Jm} Residual}{\sum_{j=1}^{Jm} p(1-p)} I(j \in R_{jm})$

(D) Update predictions with the new tree

Update $F_{(m)}(x) = F_{m-1}(x) + \eta \sum_{j=1}^{J_m} \gamma_{jm} I(x \epsilon R_{jm})$

Step 3: Output the final prediction

Output $F_M(x) = F_0(x) + \eta \sum_{j=1}^{J_m} \gamma_{j1} I(x \epsilon R_{j1}) + \eta \sum_{j=1}^{J_m} \gamma_{j2} I(x \epsilon R_{j2}) + \eta \sum_{j=1}^{J_m} \gamma_{jM} I(x \epsilon R_{jM})$

The process:

1. Initial predicted value
2. Loss function
3. Residual (negative gradient of loss function) for each observation
4. Build a first weak learner
5. Added into a model: $F(x) = F(x0) + \eta$ (first weak learner)
6. Calculate new prediction from the model for each observation
7. Calculate new residual for each observation
8. Build a second weak learner
9. Added into a model: $F(x) = F(x0) + \eta$ (first weak learner) + η (second weak learner)