

Gradient Boosting

Un Ejemplo de Cálculo Manual con Demostraciones e Inferencia

July 22, 2025

1 Introducción

Este doc es un ejemplo detallado y paso a paso del algoritmo de Gradient Boosting para una tarea de clasificación binaria. El objetivo es construir un modelo que prediga si una persona es obesa basándose en su altura y peso. Vamos a realizar manualmente tres iteraciones completas, demostrando matemáticamente cómo y por qué se elige cada corte en los árboles de decisión, y finalmente usaremos el modelo entrenado para hacer una nueva predicción (inferencia).

1.1 Los Datos de Entrenamiento

Nuestro set de datos consiste en cuatro personas. Vamos a usar un objetivo binario donde **Obeso** = 1 y **No Obeso** = 0. La tasa de aprendizaje (learning rate) para este ejemplo se fija en $\eta = 0.5$.

Table 1: Datos Iniciales de Entrenamiento

Nombre	Altura (m)	Peso (kg)	Objetivo (y)
Matías	1.81	73	0 (No Obeso)
Juan	1.78	90	1 (Obeso)
Santiago	1.90	90	0 (No Obeso)
Lucas	1.91	121	1 (Obeso)

2 Recorrido del Algoritmo de Entrenamiento

2.1 Inicialización (m=0)

El primer paso es crear una predicción inicial y simple. Esta se basa en la proporción general de la clase positiva (“Obeso”) en los datos. Con 2 personas obesas de un total de 4, la probabilidad inicial es:

$$p_0 = \frac{\text{Cantidad de Obesos}}{\text{Cantidad Total de Personas}} = \frac{2}{4} = 0.5$$

El modelo trabaja con log-odds, así que convertimos esta probabilidad para obtener nuestro modelo inicial, $F_0(x)$:

$$F_0(x) = \log\left(\frac{p_0}{1 - p_0}\right) = \log\left(\frac{0.5}{0.5}\right) = \log(1) = 0$$

En $m = 0$, el modelo predice un log-odds de **0** para todos, lo que corresponde a una probabilidad del **50%** de ser obeso.

2.2 Iteración 1 (m=1)

2.2.1 Paso 1a: Calcular los Pseudo-Residuos

Calculamos el error del modelo inicial. El residuo es el valor real (y) menos la probabilidad predicha ($p_0 = 0.5$). La fórmula es $r_{i1} = y_i - p_0$.

$$\text{Matías } (y = 0): r_1 = 0 - 0.5 = -\mathbf{0.5}$$

$$\text{Juan } (y = 1): r_1 = 1 - 0.5 = \mathbf{0.5}$$

$$\text{Santiago } (y = 0): r_1 = 0 - 0.5 = -\mathbf{0.5}$$

$$\text{Lucas } (y = 1): r_1 = 1 - 0.5 = \mathbf{0.5}$$

2.2.2 Paso 1b: Ajustar un Árbol Débil (h_1) y Justificación del Corte

El árbol de decisión elige el corte que más reduce la impureza de los residuos. Para un objetivo continuo como los residuos, esto equivale a maximizar la ****Reducción de Varianza****. Los residuos a separar son $[-0.5, 0.5, -0.5, 0.5]$, cuya varianza es ****0.25****. Se comparan los posibles cortes y se determina que la regla “**¿El Peso es > 105.5 kg?**” es la que más reduce la varianza (reducción de ****0.083****), por lo que el algoritmo la elige.

2.2.3 Paso 1c: Calcular los Valores de las Hojas (γ_1)

Con el corte ya decidido, calculamos los valores de salida (γ) para cada hoja usando la fórmula:

$$\gamma = \frac{\sum \text{residuos}}{\sum [p_{prev}(1 - p_{prev})]}$$

Para esta primera iteración, la probabilidad previa (p_{prev}) es $p_0 = 0.5$ para todos, por lo que el término del denominador para cada persona es $0.5 \times (1 - 0.5) = 0.25$.

Cálculo Detallado para la Hoja 2 (Peso ≤ 105.5 kg): Esta hoja contiene a Matías, Juan y Santiago.

- **Numerador (Suma de Residuos):** $(-0.5)_{\text{Matías}} + (0.5)_{\text{Juan}} + (-0.5)_{\text{Santiago}} = -0.5$.
- **Denominador (Suma de $p(1 - p)$):** $(0.25)_{\text{Matías}} + (0.25)_{\text{Juan}} + (0.25)_{\text{Santiago}} = 0.75$.

El valor de la hoja es $\gamma_{21} = \frac{-0.5}{0.75} = -\mathbf{0.667}$.

Cálculo para la Hoja 1 (Peso > 105.5 kg): Esta hoja solo contiene a Lucas.

- **Numerador:** 0.5.
- **Denominador:** 0.25.

El valor de la hoja es $\gamma_{11} = \frac{0.5}{0.25} = \mathbf{2.0}$.

2.2.4 Paso 1d: Actualizar el Modelo (F_1)

Usamos $F_1(x) = F_0(x) + \eta \cdot h_1(x)$:

- **Matías, Juan, Santiago:** $F_1 = 0 + (0.5 \times -0.667) = -0.3335 \implies p_1 = 0.417$
- **Lucas:** $F_1 = 0 + (0.5 \times 2.0) = 1.0 \implies p_1 = 0.731$

2.3 Iteración 2 ($m = 2$)

2.3.1 Paso 2a: Calcular los Pseudo-Residuos

Los nuevos errores ($r_{i2} = y_i - p_1$) son: $[-0.417, 0.583, -0.417, 0.269]$.

2.3.2 Paso 2b: Ajustar un Árbol Débil (h_2) y Justificación del Corte

Se repite el proceso de buscar el mejor corte. La regla “¿El Peso es > 81.5 kg?” es la que más minimiza la Suma de Errores al Cuadrado (SEC) de las hojas resultantes y es elegida.

2.3.3 Paso 2c: Calcular los Valores de las Hojas (γ_2)

Para la Hoja 1 (Peso > 81.5 kg): $\gamma_{12} = \frac{0.435}{0.683} = 0.637$.

Para la Hoja 2 (Peso ≤ 81.5 kg): $\gamma_{22} = \frac{-0.417}{0.243} = -1.716$.

2.3.4 Paso 2d: Actualizar el Modelo (F_2)

Matías: $F_2 = -0.3335 + (0.5 \times -1.716) = -1.1915 \implies p_2 = 0.233$

Juan y Santiago: $F_2 = -0.3335 + (0.5 \times 0.637) = -0.015 \implies p_2 = 0.496$

Lucas: $F_2 = 1.0 + (0.5 \times 0.637) = 1.3185 \implies p_2 = 0.789$

2.4 Iteración 3 ($m = 3$)

2.4.1 Paso 3a: Calcular los Pseudo-Residuos

Los nuevos errores ($r_{i3} = y_i - p_2$) son: $[-0.233, 0.504, -0.496, 0.211]$.

2.4.2 Paso 3b: Ajustar un Árbol Débil (h_3) y Justificación Rigurosa del Corte

En este punto, Juan y Santiago tienen el mismo peso. Para separarlos, es necesario usar la altura. Comparamos los costes (SEC) de los posibles cortes:

Opción A: Corte por ‘Peso > 81.5 kg’. Coste Total = **0.5286**.

Opción B: Corte por ‘Altura > 1.85 m’. Coste Total = **0.5216**.

Decisión Numérica: El algoritmo elige el corte por **altura** porque su coste total (0.5216) es menor que el coste del corte por peso (0.5286).

2.4.3 Paso 3c: Calcular los Valores de las Hojas (γ_3)

Para la Hoja 1 (Altura > 1.85 m): $\gamma_{13} = \frac{-0.285}{0.416} = -0.685$.

Para la Hoja 2 (Altura ≤ 1.85 m): $\gamma_{23} = \frac{0.271}{0.429} = 0.632$.

2.4.4 Paso 3d: Actualizar el Modelo (F_3)

Matías: $F_3 = -1.1915 + (0.5 \times 0.632) = -0.8755 \implies p_3 = 0.294$

Juan: $F_3 = -0.015 + (0.5 \times 0.632) = 0.301 \implies p_3 = 0.575$

Santiago: $F_3 = -0.015 + (0.5 \times -0.685) = -0.3575 \implies p_3 = 0.412$

Lucas: $F_3 = 1.3185 + (0.5 \times -0.685) = 0.976 \implies p_3 = 0.726$

3 Ejemplo de Inferencia: Evaluación de un Nuevo Caso

Ahora usamos el modelo entrenado para predecir la probabilidad de obesidad de una nueva persona, Carlos, que no estaba en los datos de entrenamiento.

3.1 Datos de Inferencia

- **Persona:** Carlos
- **Altura:** 1.75 m
- **Peso:** 110 kg

3.2 Paso 1: Recorrido a Través de los Árboles

Hacemos pasar los datos de Carlos por cada regla que creamos:

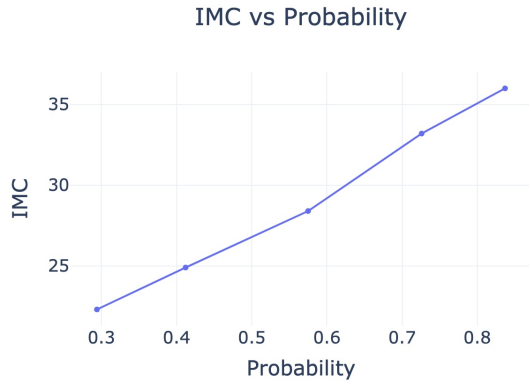
- **Árbol 1** ($\text{¿Peso} > 105.5 \text{ kg?}$): Sí. Cae en la Hoja 1. Valor (γ_1): **2.0**.
- **Árbol 2** ($\text{¿Peso} > 81.5 \text{ kg?}$): Sí. Cae en la Hoja 1. Valor (γ_2): **0.637**.
- **Árbol 3** ($\text{¿Altura} > 1.85 \text{ m?}$): No. Cae en la Hoja 2. Valor (γ_3): **0.632**.

El camino de Carlos (Hoja 1, Hoja 1, Hoja 2) es único y diferente al de las personas del entrenamiento.

3.3 Paso 2: Cálculo del Log-Odds Final (F_3)

Sumamos la predicción inicial y las contribuciones de cada árbol, escaladas por $\eta = 0.5$:

$$\begin{aligned} F_3(\text{Carlos}) &= F_0 + (\eta \times \gamma_1) + (\eta \times \gamma_2) + (\eta \times \gamma_3) \\ &= 0 + (0.5 \times 2.0) + (0.5 \times 0.637) + (0.5 \times 0.632) \\ &= 1.0 + 0.3185 + 0.316 = \mathbf{1.6345} \end{aligned}$$



3.4 Paso 3: Conversión a Probabilidad

Convertimos el log-odds final a una probabilidad con la función sigmoide:

$$p_3(\text{Carlos}) = \frac{1}{1 + e^{-F_3}} = \frac{1}{1 + e^{-1.6345}} \approx \mathbf{0.837}$$

La probabilidad de que Carlos sea considerado obeso por el modelo es del **83.7%**.

4 Resumen Final y Comparativa

La tabla final resume la evolución de las probabilidades para los datos de entrenamiento y el caso de inferencia.

Table 2: Evolución Completa de Predicciones (Entrenamiento e Inferencia)

Nombre	Tipo	y	p_0	p_1	p_2	p_3	IMC (kg/m ²)
Matías	Entrenamiento	0	0.5	0.417	0.233	0.294	22.3
Santiago	Entrenamiento	0	0.5	0.417	0.496	0.412	24.9
Juan	Entrenamiento	1	0.5	0.417	0.496	0.575	28.4
Lucas	Entrenamiento	1	0.5	0.731	0.789	0.726	33.2
Carlos	Inferencia	N/A	0.5	0.731	0.789	0.837	36.0

El modelo final, compuesto por tres árboles simples, es capaz de generar predicciones con matices para casos nuevos y complejos, demostrando el poder del ensamblado iterativo.