Gradient Boost

1. Gradient Boost crea una cadena de árboles de profundidad fija.

2. Comienza con un solo valor, un nodo hoja.
Y luego calcula árboles para calcular el error cometido por el anterior, por este primer estimador.

3. Cada árbol está ponderado por un factor constante llamado **learning rate** o tasa de aprendizaje.

a. Los árboles están restringidos en su crecimiento (no son tocones, pero tampoco son arboles que van a llegar a unas profundidades completas)

Gradient Boost puede usarse para problemas de regresión o de clasificación.

En este ejemplo, queremos determinar un valor en un rango continuo, el peso (Weight), por lo tanto estamos ante un problema de regresión.

Height (m)	Favorite color	Gender	Weight (kg)
1.6	Blue	Male	88
1.6	Green	Female	76
1.5	Blue	Female	56
1.8	Red	Male	73
1.5	Green	Male	77
1.4	Blue	Female	57

Paso 1: Calculamos una sola hoja de un árbol, que clasifica todos los casos. Esta hoja es el promedio

Estimador → Promedio = suma de todos los weight / 6 = **71,2**

Height (m)	Favorite color	Gender	Weight (kg)	Prediction (kg)	Residual
1.6	Blue	Male	88	71.2	16.8
1.6	Green	Female	76	71.2	4.8
1.5	Blue	Female	56	71.2	-15.2
1.8	Red	Male	73	71.2	1.8
1.5	Green	Male	77	71.2	5.8
1.4	Blue	Female	57	71.2	-14.2

71.2

Paso 2:
Calculamos el error cometido por el estimador hasta ahora.
Se lo llama: **Residuo**

Residuo = (Peso observado - peso predicho)

NOTA: El término correcto sería **pseudo-residuo**. Los **residuos** son los errores en regresión lineal, acá se toma la palabra por similitud, pero se le agrega el **pseudo** para entender que es otro modelo.

Paso 3:

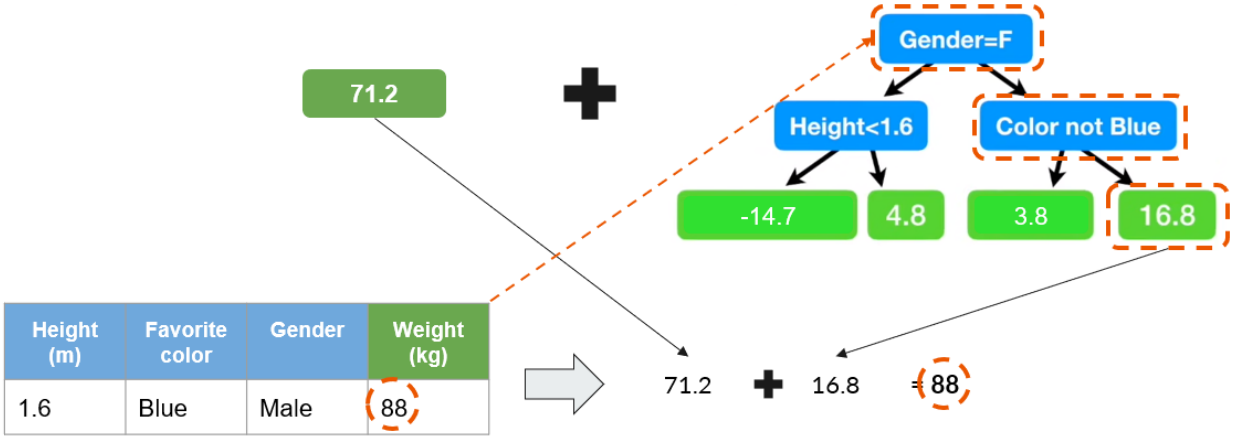
Construimos un árbol para los residuos. El árbol estará limitado a solo 4 hojas.

En casos reales se usa un valor entre 8 y 32 hojas.

Unificamos los valores de a dos con el promedio

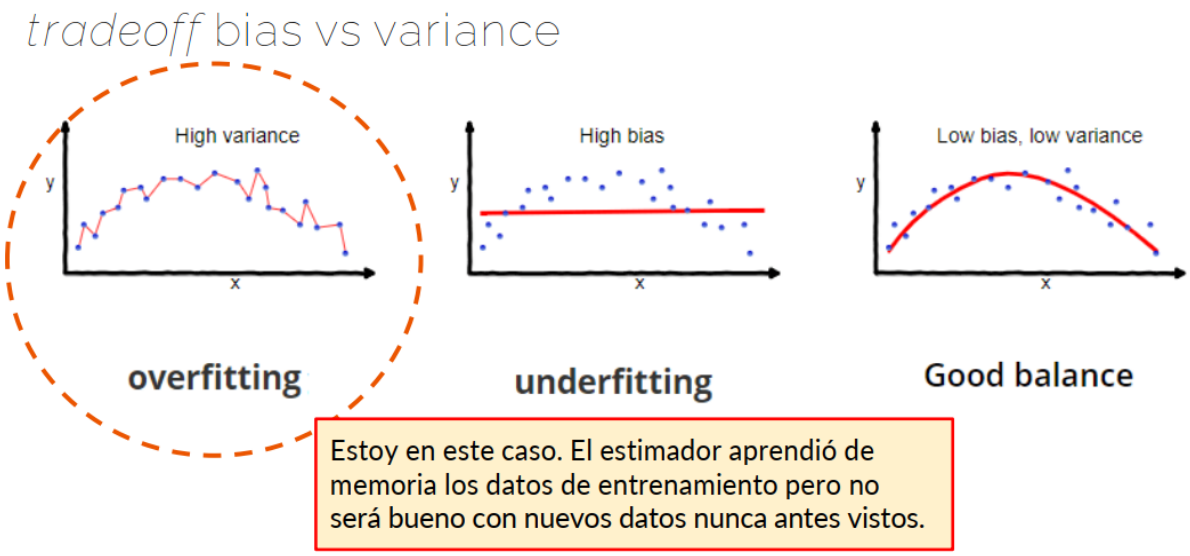
$$\frac{-14.2 + (-15.2)}{2} = -14.7$$

$$\frac{1.8 + 5.8}{2} = 3.8$$



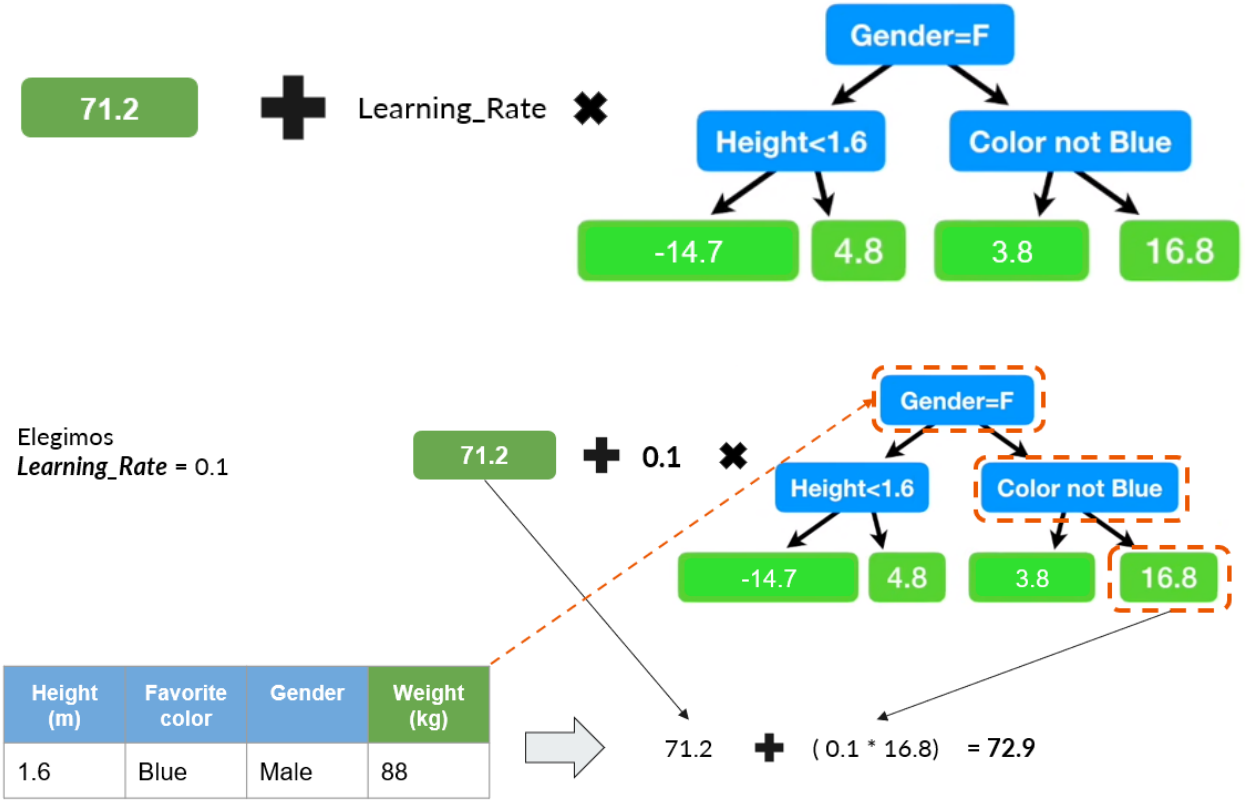
El modelo predice sospechosamente bien…

Esto es porque lo estamos evaluando en el mismo conjunto de entrenamiento.



Gradient Boost maneja el overfitting usando un parámetro llamado **Learning Rate**, para escalar la contribución de cada árbol.

Es un valor entre 0 y 1.



72.9 no es tan buena predicción como 88, pero es mejor que la anterior: 71.2.

Se trata de dar pequeños pasos en la dirección correcta.

La evidencia empírica sugiere que tomar muchos pasos pequeños en la dirección correcta resulta en una mejor predicción en un conjunto de datos de prueba.

Es decir: tendrá una varianza baja

Generamos un nuevo árbol

Volvemos al Paso 2

Calculamos el error cometido por el estimador hasta ahora.

Residuo = (Peso observado - peso predicho)

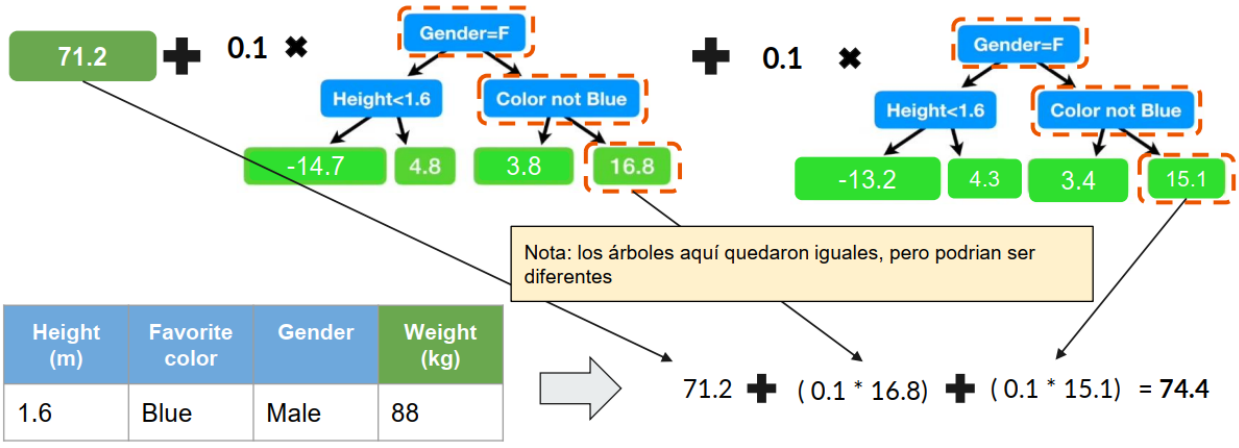
Residual
16.8
4.8
-15.2
1.8
5.8
-14.2

Residuos originales

Residual
15.1
4.3
-13.7
1.4
5.4
-12.7

Residuos nuevos

Podemos ver acá que todos los errores (residuos) se redujeron.
Dimos un paso en la dirección correcta

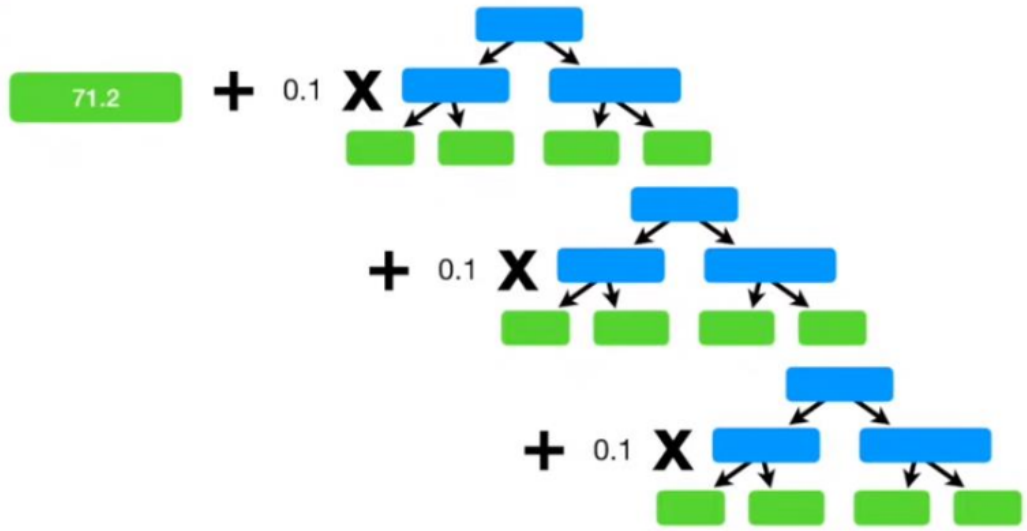


Height (m)	Favorite color	Gender	Weight (kg)
1.6	Blue	Male	88

Seguimos añadiendo árboles, hasta que los residuos no cambien significativamente o bien alcanzamos un número preestablecido de árboles seteado como parámetro.

Finalmente para estimar un nuevo valor, cuando el método termina de entrenar, procedemos como antes, usando el valor inicial y sumando los residuos ponderados de cada árbol.

Con 3 arboles quedaría asi:


71.2
+
0.1
X
+
0.1
X
+
0.1
X