

Ayudantía 8

Árboles de decisión, *Ensambles* y Presentación T4

Por Melanie Castillo y Rodrigo Figueroa

25 de Mayo 2025

Contenidos



- *Árboles de decisión.*
- *Ensambles*
- *Presentación Tarea 4*

Árboles de decisión



- Técnica de aprendizaje supervisado
- Usados para clasificación y regresión.
- Organizan los datos para dividirlos en diferentes clases.

Árboles de decisión

Clima	Temperatura	Humedad	Viento	Jugar?
soleado	alta	alta	F	No
soleado	alta	alta	V	No
nublado	alta	alta	F	Si
lluvioso	Agradable	alta	F	Si
lluvioso	frio	normal	F	Si
lluvioso	frio	normal	V	No
nublado	frio	normal	V	Si
soleado	Agradable	alta	F	No
soleado	frio	normal	F	Si
lluvioso	Agradable	normal	F	Si
soleado	Agradable	normal	V	Si
nublado	Agradable	alta	V	Si
nublado	alta	normal	F	Si
lluvioso	Agradable	alta	V	No

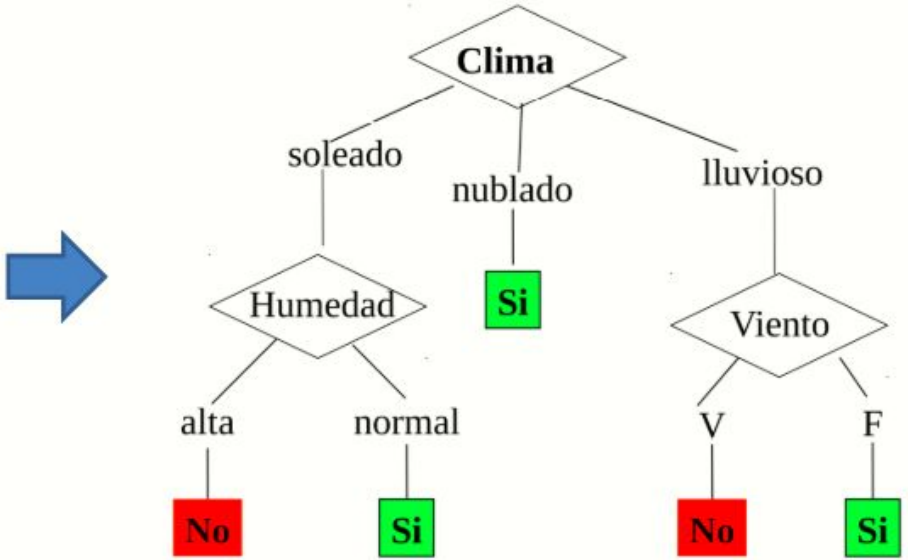


Imagen obtenida de la clase de árboles de decisión del profesor Hans Löbel 2025-1

Árboles de decisión



Estructura del Árbol:

- Raíz: Representa la totalidad de los datos.

- Nodos:
  - Cada nodo (excepto las hojas) es una pregunta o condición sobre alguna característica.
  - Según la respuesta, los datos se dividen en nodos del siguiente nivel.

Árboles de decisión



Estructura del Árbol:

- Para decidir la mejor división en un nodo se utilizan métricas como Gini, Entropía o índice de impureza.
- Elegimos la característica que entrega la mayor ganancia de información (mayor reducción de entropía)
- Estas son calculadas por los modelos automáticamente

Árboles de decisión

DecisionTreeClassifier

```
class sklearn.tree.DecisionTreeClassifier(*, criterion='gini', splitter='best',
max_depth=None, min_samples_split=2, min_samples_leaf=1,
min_weight_fraction_leaf=0.0, max_features=None, random_state=None,
max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0,
monotonic_cst=None)
```

[source]

A decision tree classifier.

Read more in the User Guide.

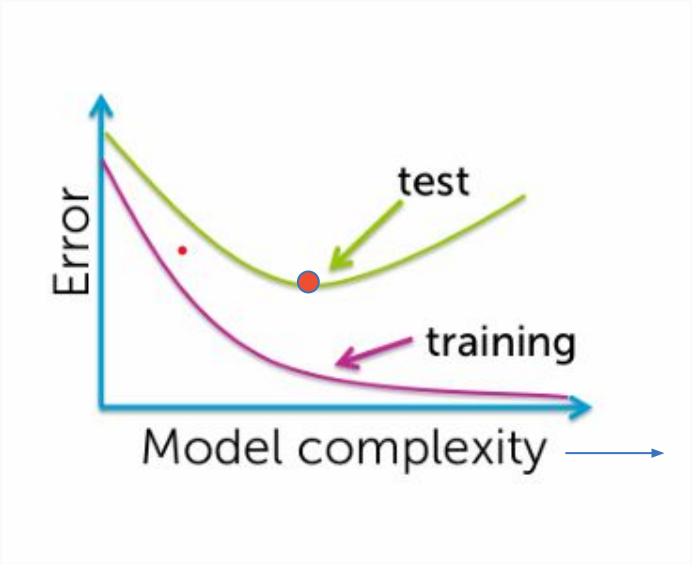
Parameters:

criterion : *{"gini", "entropy", "log_loss"}, default="gini"*

The function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "log_loss" and "entropy" both for the Shannon information gain, see Mathematical formulation.

Fuente: DecisionTreeClassifier. (n.d.). Scikit-learnhttps://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

Sobreajuste es un problema importante para los árboles de decisión



Profundidad del árbol en este caso

Imagen obtenida de la clase de árboles de decisión del profesor Hans Löbel 2025-1

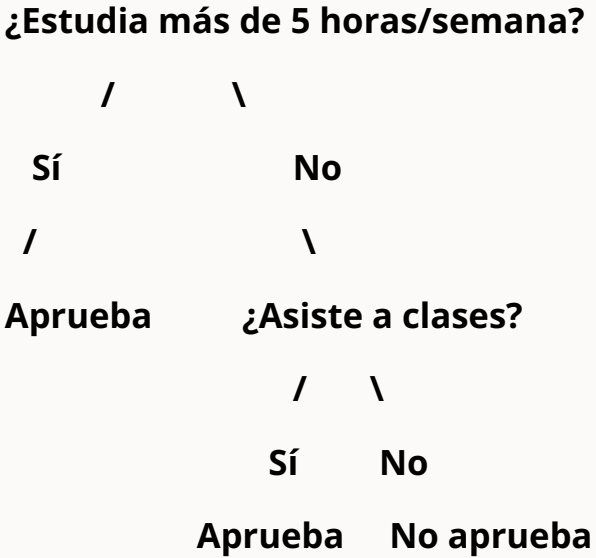
Técnicas para reducir el sobreajuste

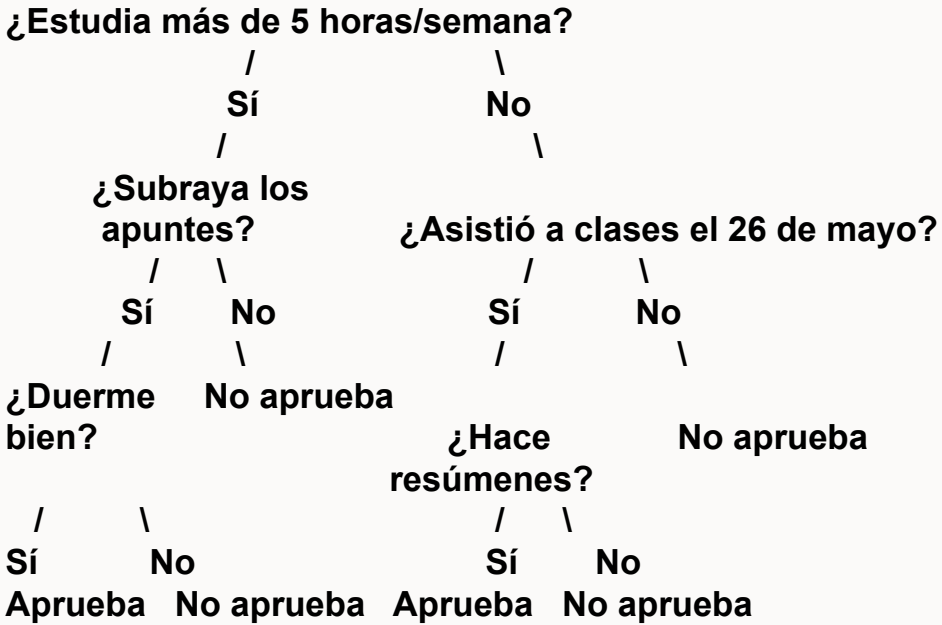


- Usar un conjunto de validación para detener el crecimiento.
- Detener cuando los datos restantes no son estadísticamente relevantes.
- Podar el árbol después de construirlo completamente.
- Penalizar la complejidad al seleccionar atributos.

Ejemplo de sobreajuste en árboles de decisión







Este árbol es poco profundo, tiene reglas generales. Puede que no sea perfecto, pero generaliza bien a nuevos estudiantes.

Este árbol tiene muchas divisiones específicas, como fechas y detalles muy concretos. Esto significa que está "aprendiendo de memoria" los datos del pasado, en vez de aprender patrones generales.



Más modelos

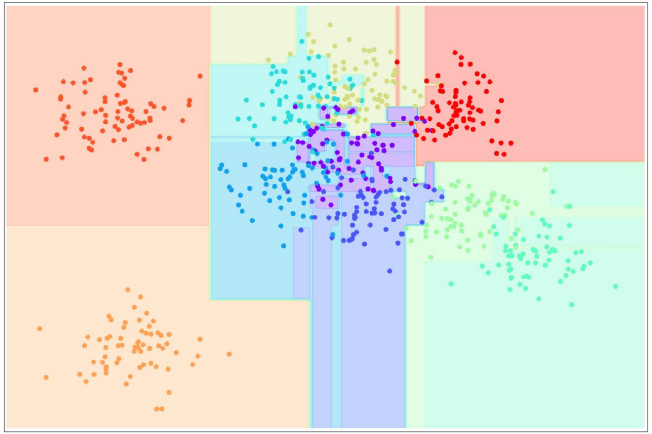


- El **árbol de decisión** tiene una gran ventaja: es simple y fácil de interpretar...
- ... pero puede sufrir serios problemas de sobreajuste

Más modelos

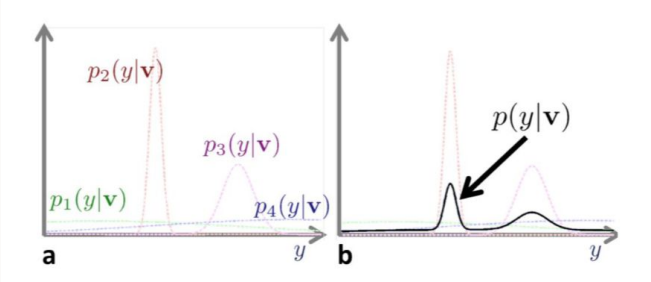
¿De dónde proviene el sobreajuste en un árbol de decisión?

- Al crear nuevos nodos en el entrenamiento (“bajar” en el árbol), la **muestra** se hace cada vez más reducida
- Si hay muchos **atributos**, es altamente probable elegir alguno bueno en los datos de entrenamiento, pero que es “inútil” para generalizar



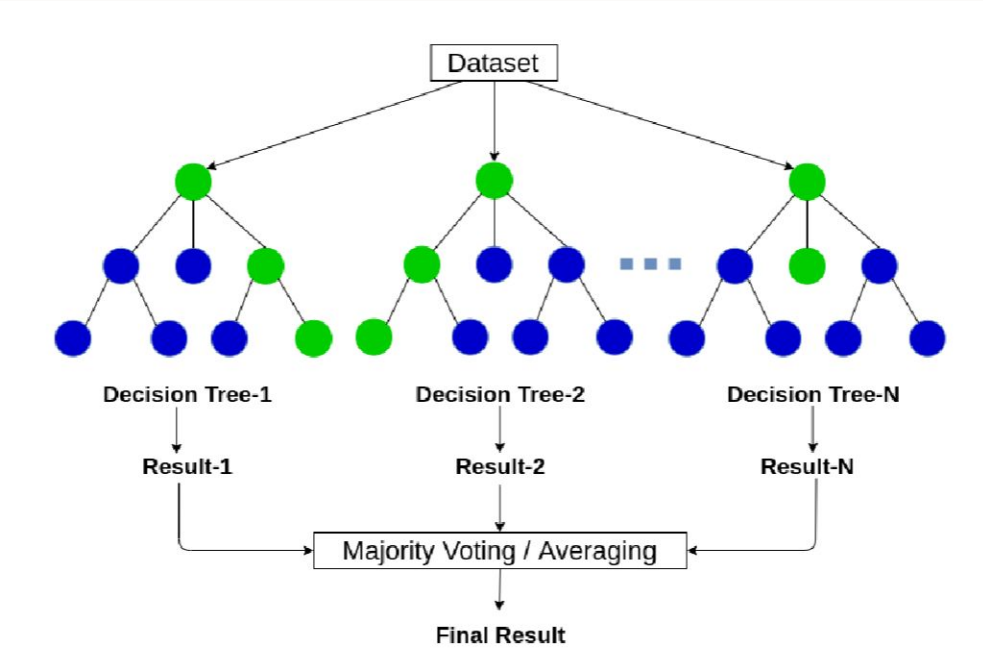
Ensambles de modelos con baja correlación

- Si el **patrón de error** (acierto) es **distinto** para todos, es altamente probable que, en promedio, la respuesta del ensamble sea correcta (los **errores se cancelan**)



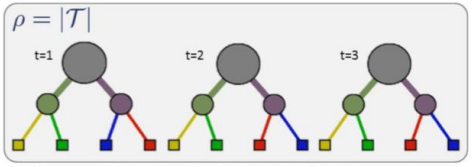
Muestras aleatorias

- Tomemos varios **árboles de decisión** y armemos una gran **“votación”**, donde cada árbol tenga “un voto”.
- Como vimos en la diapositiva pasada, necesitamos que los modelos tengan **baja correlación** para que nuestro plan resulte. **¿Cómo logramos esto?**
- Si cada modelo ve sólo una parte de los datos (elegida aleatoriamente), es probable que la **correlación** entre ellos **disminuya**

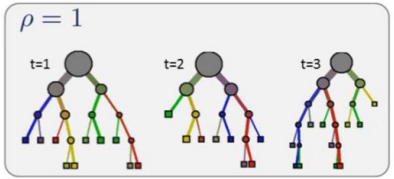


Atributos aleatorias

- Lo anterior no basta: **¿qué pasa si tenemos un atributo muy bueno?**
- La solución es similar: tomamos muestras aleatorias de atributos
- Esto no solo **disminuye la correlación**, sino que también **limita la profundidad del árbol**, reduciendo la complejidad de este



a) Low randomness, high tree correlation



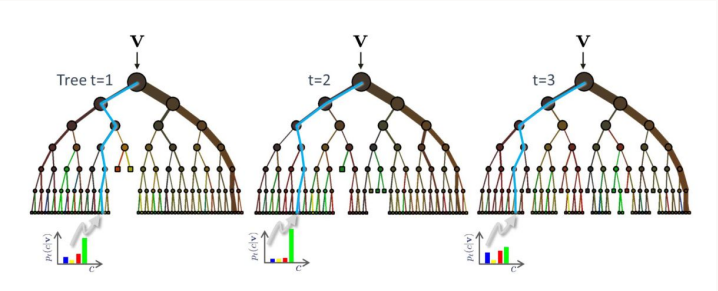
b) High randomness, low tree correlation



Random Forest

Random Forest

- La introducción de todas estas estrategias basadas en **aleatoriedad**, aplicada a **varios árboles de decisión**, convierten esta técnica en un ensamble llamado ***Random Forest***

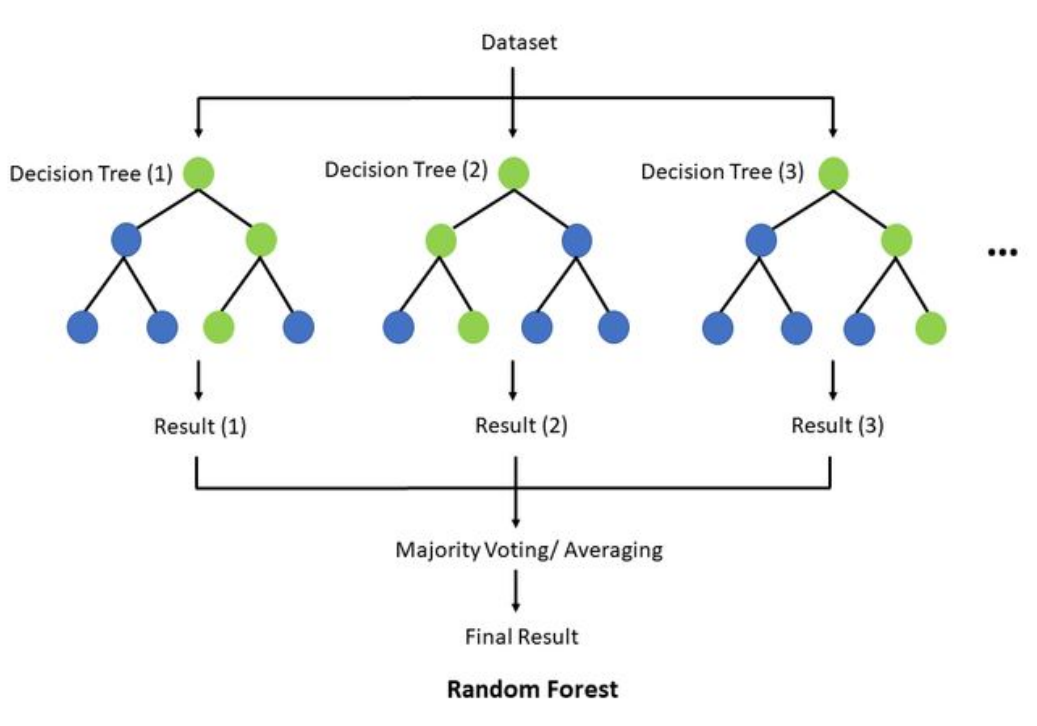


Random Forest



- Un árbol para cada muestra aleatoria (subconjunto) de los datos.
- Árboles de **poca profundidad** para evitar el *overfitting*.
- La predicción final se obtiene de un promedio entre las predicciones de los árboles del bosque

Random Forest



Hiperparámetros: RF



- **Los mismos de un árbol**
- **n_estimators:** número de árboles en el bosque

Ventajas de *Random Forest*



- Al igual que los **árboles de decisión**, son **fácilmente interpretables**
- Rendimiento es **altamente competitivo** con datos **tabulados**
- Gracias a la aleatorización en su construcción, son altamente **resistentes** al ***overfitting***

Ventajas de *Random Forest*



- Todas estas ventajas hacen que ***Random Forest*** sea un modelo **altamente utilizado**, a día de hoy, en el mundo de *machine learning*



Ensambles

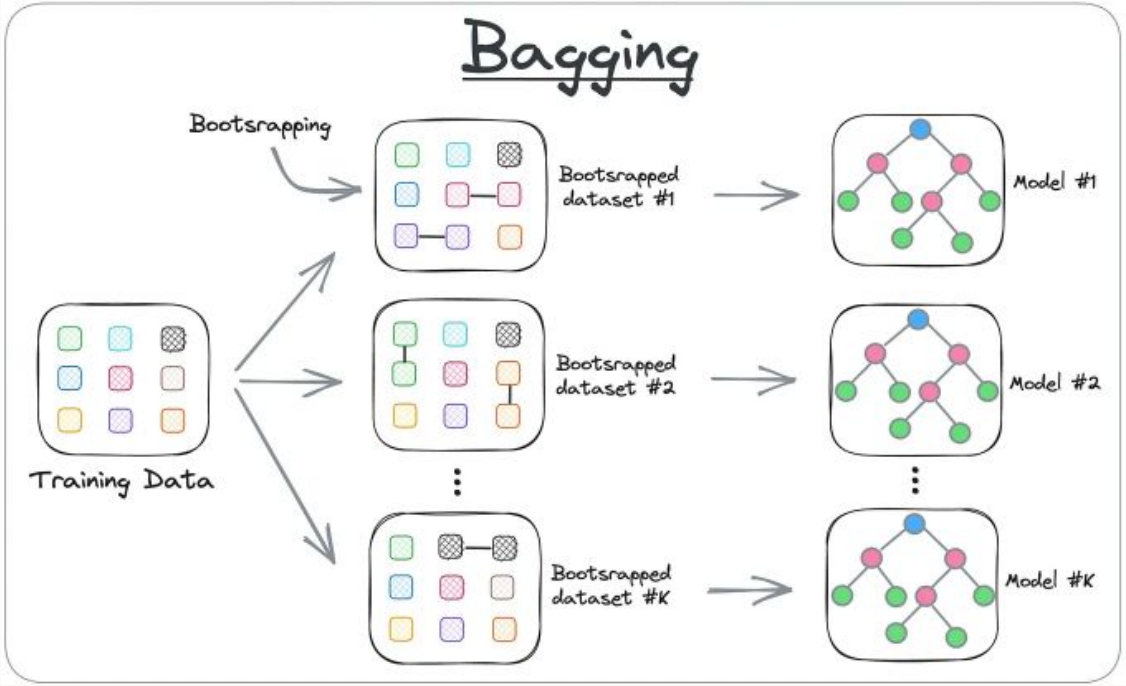
¿Qué son los ensambles?



- Combinación de múltiples modelos con el fin de **crear uno mejor.**

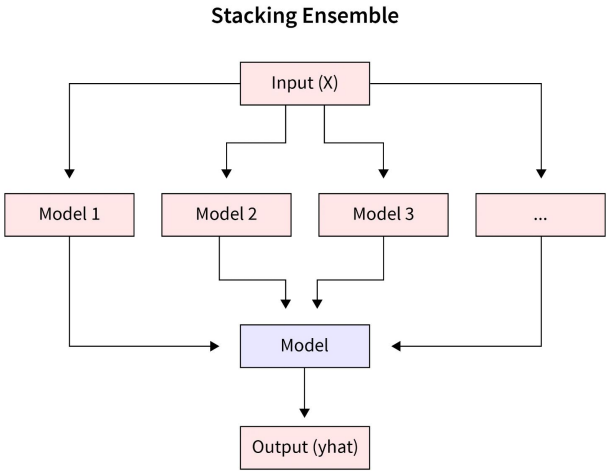
Tipos de ensambles : *Bagging*

- Crear **múltiples datasets** con muestreo **con reemplazo** y entrenar distintos modelos.
- Luego por alguna d**ecisión conjunta**, por ejemplo mayoria de votación se clasifica/predice.
- Random Forest es el clásico de estos modelos



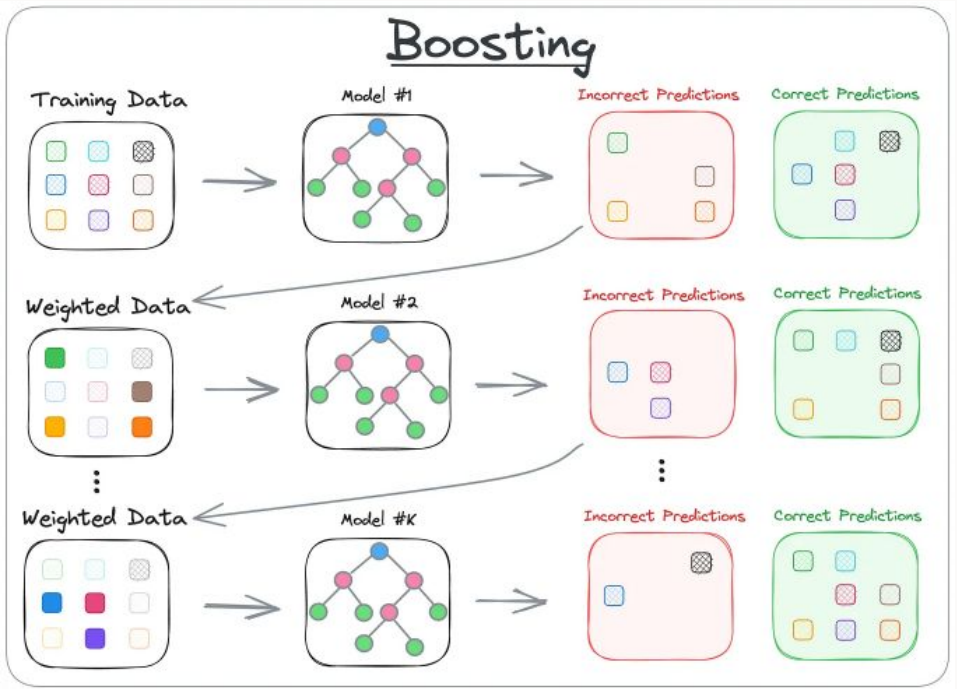
Tipos de ensambles : *Stacking*

- Entrenar varios modelos con el mismo dataset. Se recomienda que sean ***weak learners***.
- Luego ***stackear*** (concatenar) los *outputs* de los modelos y usarlos como *inputs* para entrenar un ***metalearner***.
- Finalmente este *metalearner* genera el *ouput* esperado.

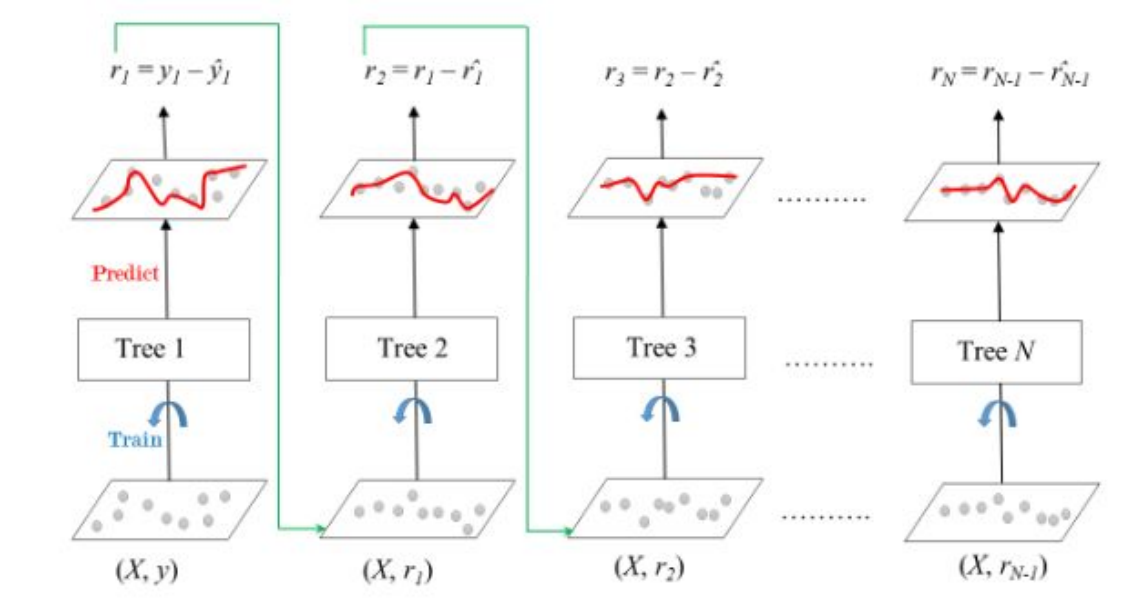


Tipos de ensambles : *Boosting*

- Aprovechar la **información errónea** de una jerarquía de modelos.
- De forma secuencial se entrenan modelos cada vez más débiles, para que estos **aprendan de los errores de los anteriores**.
- Luego se **combinan** para generar el output.



Gradient Boosting





Algorithm 2 Gradient Boosting

Input Data: $x_i \in R^n, y_i \in [-1, +1]$

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

2: **for** $m = 1$ to M **do**

3: Compute the negative gradient

$$r_{i,m} = -\left[\frac{\delta L(y_i, F(x_i))}{\delta F(x_i)}\right]$$

4: Fit tree to targets $r_{i,m}$, with leafs $R_{j,m}$

5: For $j = 1, 2, ...J_m$

$$\gamma_{jm} = argmin_\gamma \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x) + \gamma)$$

6: Update the estimation of F(x)

$$F_m(x) = F_{m-1}(x) + \nu(\gamma_{jm})$$

7: Output $\hat{f}(x) = f_M(x)$

Gradient Boosting : Clasificación



- Si los modelos son de **regresión** el error es simplemente la **diferencia** del valor predicho. ¿Pero qué pasa con los **problemas de clasificación**?

- No podemos usar la diferencia de clases como tal, puesto que no representan algo real*.

- Pero recordando los modelos de clasificación lo que hacen es calcular las **probabilidades de que pertenezcan a una clase**.

XGBoost : eXtreme Gradient Boosting



- Probablemente el algoritmo de boosting más utilizado.
- *Gradient Boosting* **no escala bien para muchos datos.**
- **XGboost:** Es una versión más eficiente, puesto que:
  - Agrega regularizadores L1 y L2 (quitan el overfitting)
  - Paraleliza las tareas
  - Tiene implementado una forma de **parar** con basado en Cross-Validation
  - Corta las zonas menos interesantes de árboles.



Ayudantía 8

Árboles de decisión, *Ensambles* y Presentación T4

Por Melanie Castillo y Rodrigo Figueroa

25 de Mayo 2025