



Inteligencia Artificial Aplicada para la Economía



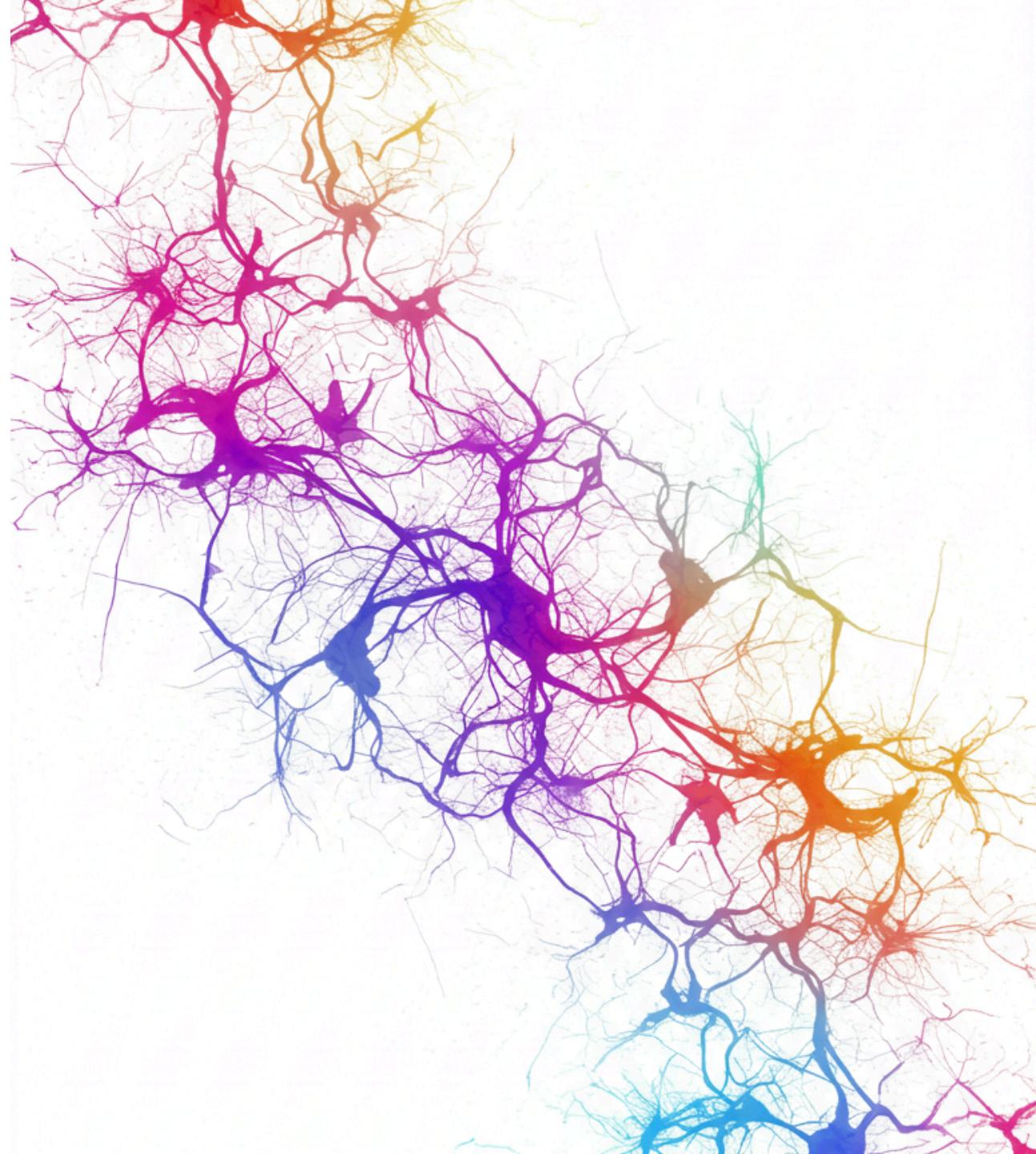
Profesores

Profesor Magistral

Camilo Vega Barbosa

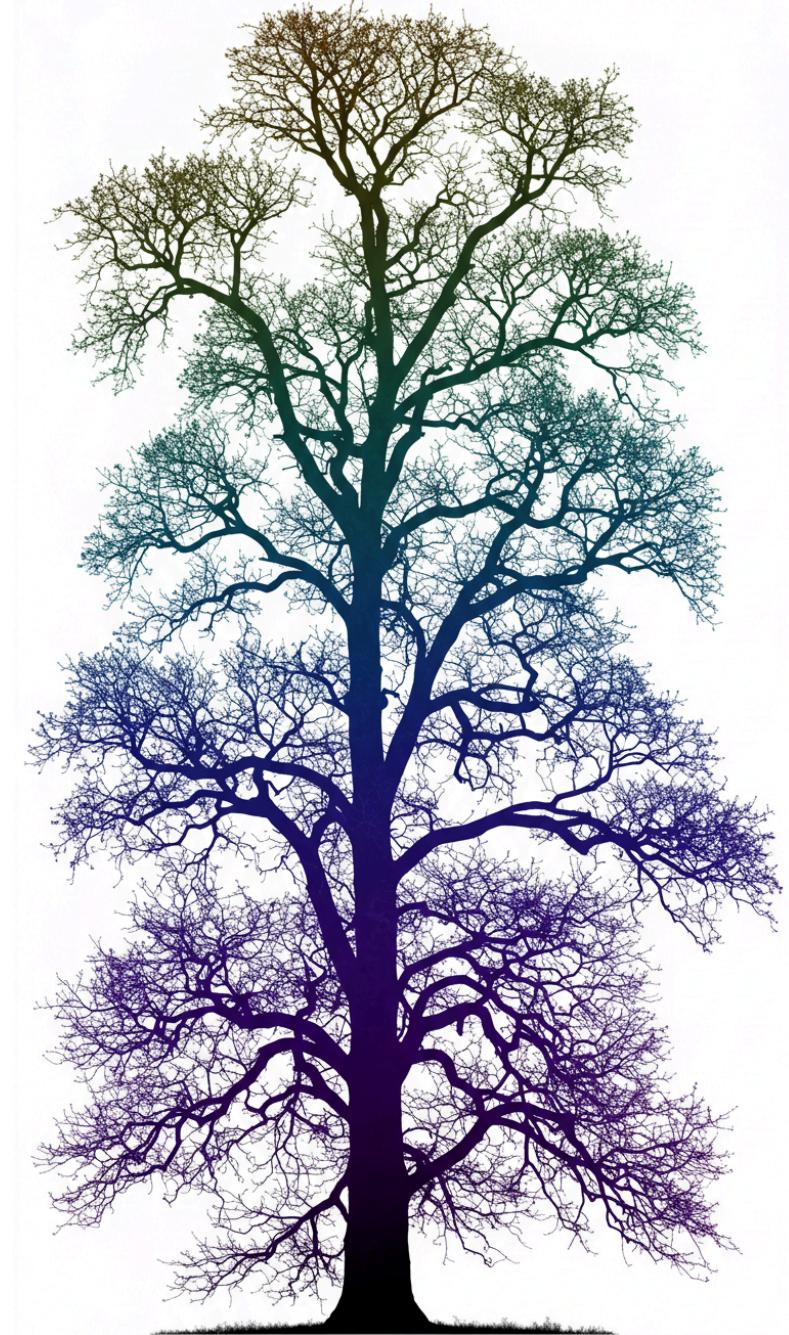
Asistente de Docencia

Sergio Julian Zona Moreno



Árboles y Ensamblés

Los caminos para predecir





Del Árbol al Bosque

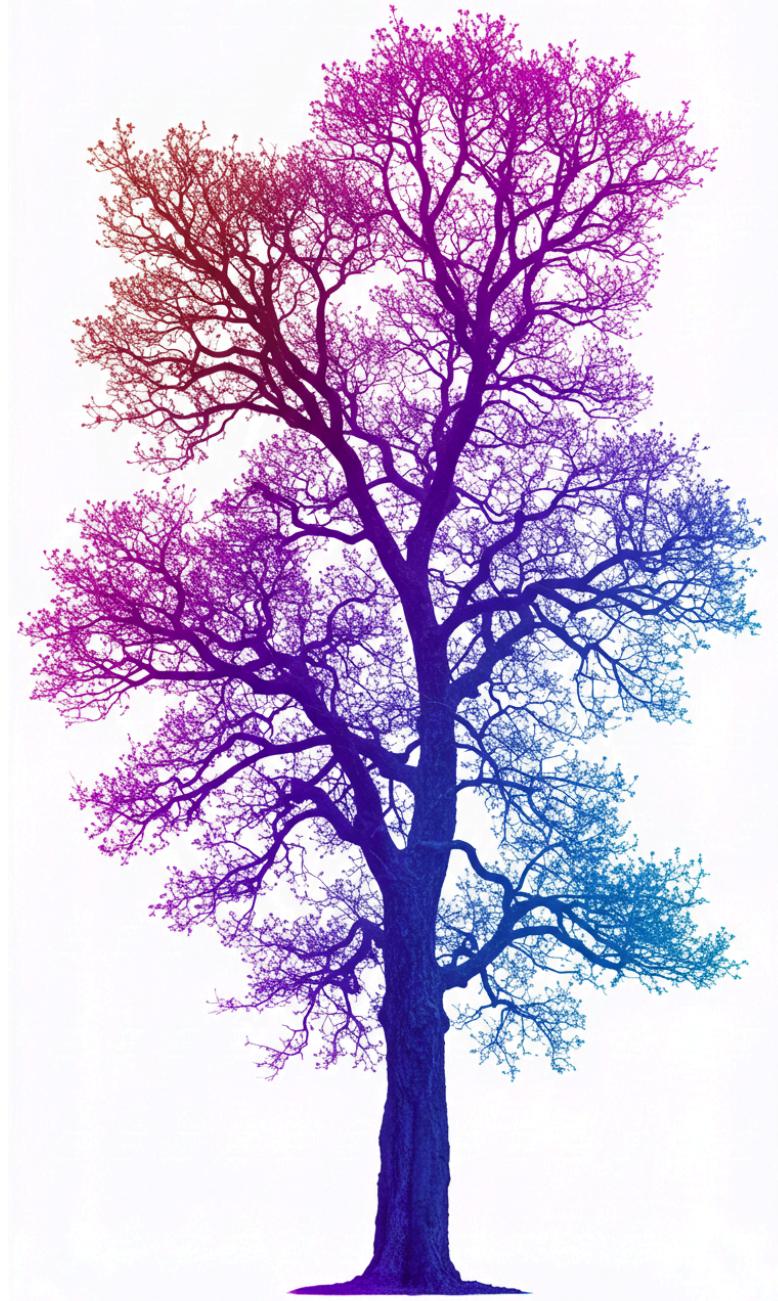
Imagina tomar decisiones como si fueras navegando por las ramas de un árbol: cada decisión te lleva por un camino diferente hasta llegar a una conclusión. Así funcionan nuestros algoritmos:



Nuestro Viaje de Aprendizaje

- **Árboles de Decisión:** El fundamento del aprendizaje basado en reglas
- **Random Forest:** La sabiduría colectiva de múltiples árboles
- **XGBoost:** La evolución y optimización del bosque

Árbol de decisión





Árboles de Decisión: La Intuición

Los árboles de decisión son como un juego de "20 preguntas" donde cada pregunta nos acerca más a la respuesta correcta.

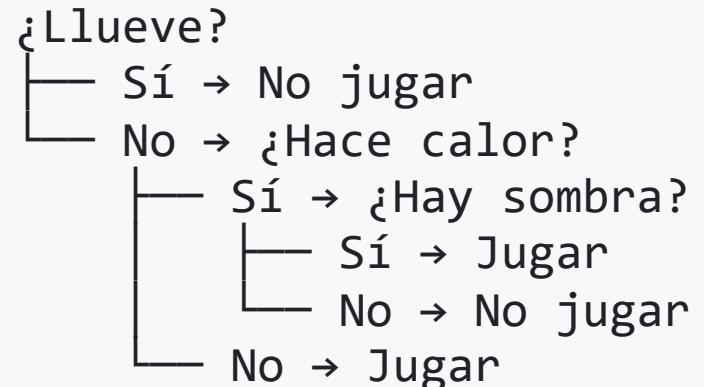


¿Cómo Funciona?

- Divide el problema en preguntas simples
- Cada nodo es una pregunta sobre una característica
- Cada rama es una posible respuesta
- Las hojas son las predicciones finales



Ejemplo Simple: ¿Jugar Tenis?





Árboles de Decisión: Aspectos Técnicos



Matemática del Árbol

- **Entropía:**

$$H(S) = - \sum_i p_i \log_2(p_i)$$

- **Ganancia de Información:**

$$IG(S, A) = H(S) - \sum_v \frac{|S_v|}{|S|} H(S_v)$$

- **Índice Gini:**

$$G = 1 - \sum_i p_i^2$$



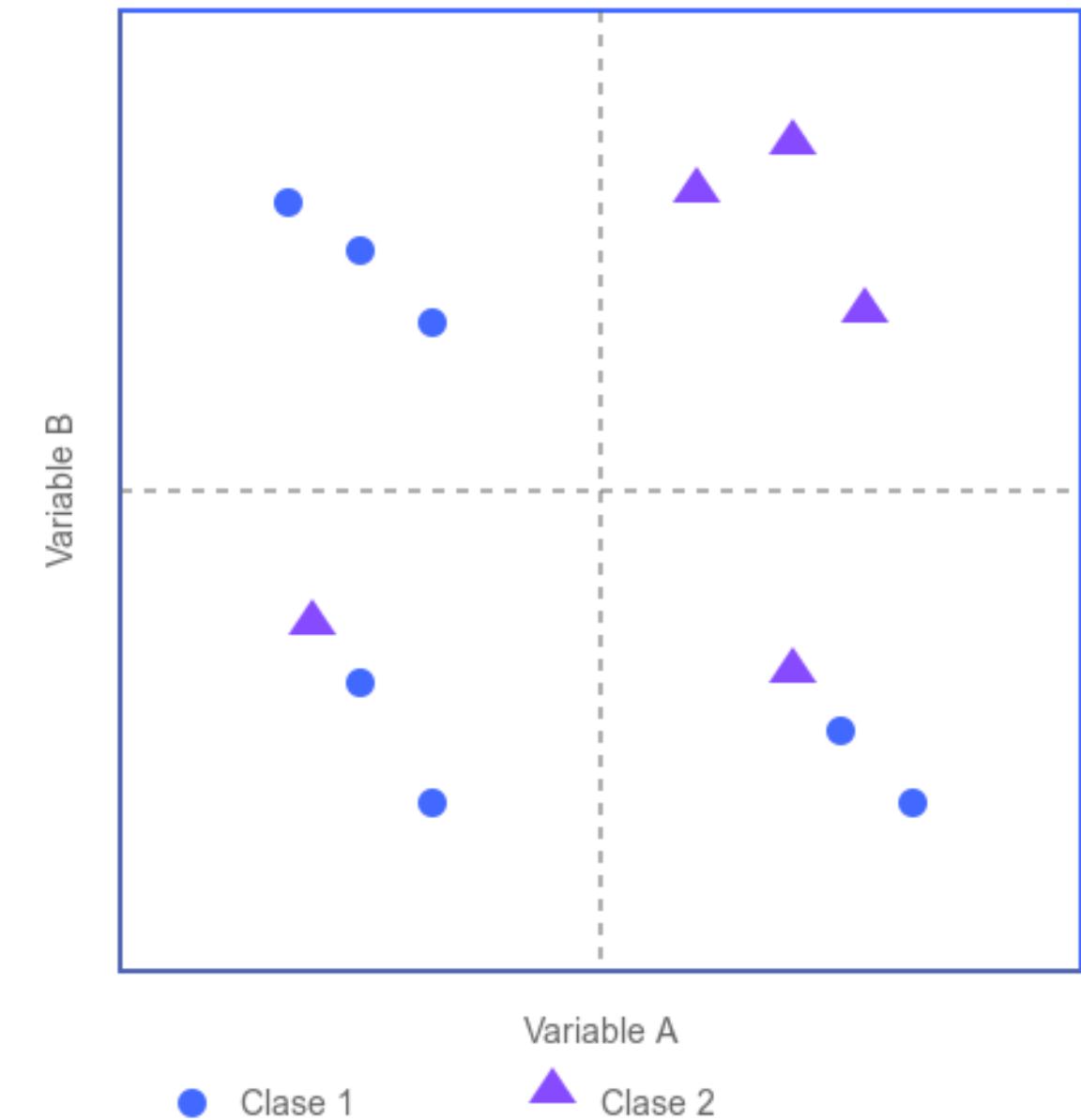
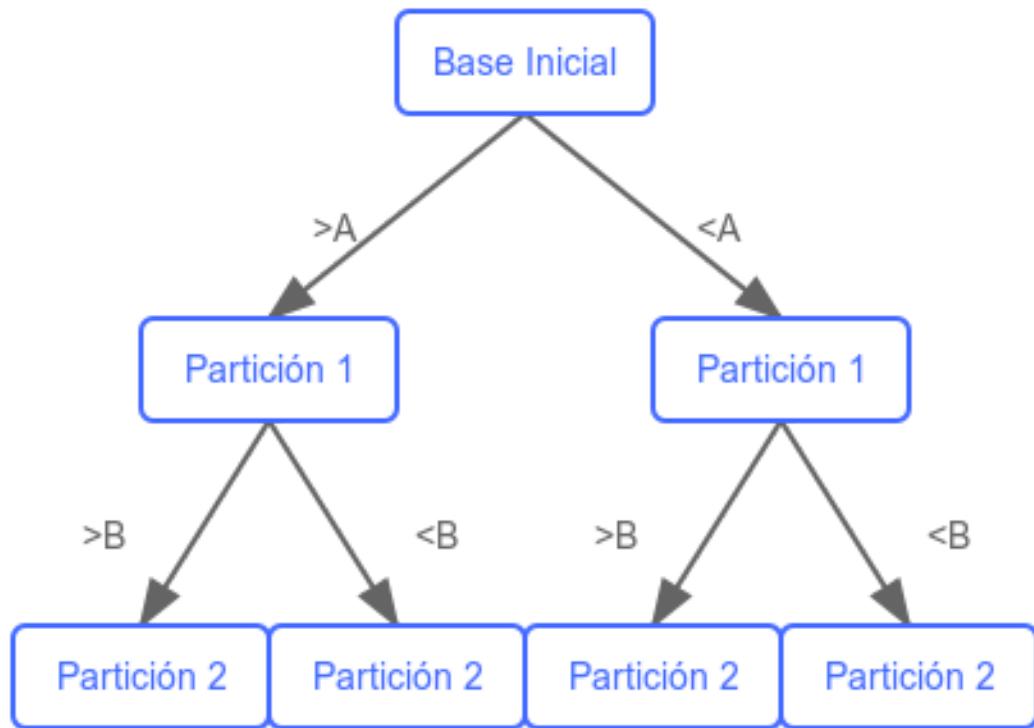
Intuición de las Métricas

- **Entropía:** Mide el "desorden" o incertidumbre en nuestros datos
- **Ganancia de Información:** Cuánto mejora nuestra predicción al usar una variable
- **Índice Gini:** Alternativa que mide la pureza de la separación



El árbol selecciona las variables y puntos de división que maximizan la ganancia de información, buscando las separaciones más eficientes de los datos.

Visualización Árbol de Decisión





Podando el Árbol: El Arte del Equilibrio

Los árboles, como los bonsáis, necesitan poda para crecer de manera saludable y eficiente. El pruning es nuestro jardinero experto en el aprendizaje automático.

Pre-pruning (Forward Pruning)

- Detiene el crecimiento antes del sobreajuste
- Establece reglas como profundidad máxima
- Es como podar mientras el árbol crece
- Previene el desperdicio de recursos

Post-pruning (Backward Pruning)

- Simplifica el árbol después del entrenamiento
- Elimina ramas que no aportan valor
- Reduce la complejidad sin sacrificar rendimiento
- Mejora la interpretabilidad del modelo

La Ciencia Detrás del Pruning



Criterios Técnicos

- Cost Complexity (α):

$$R_\alpha(T) = R(T) + \alpha|T|$$

donde $R(T)$ es el error y $|T|$ el tamaño del árbol

- Error Reduction:

$$\Delta R(t) = R(t) - R(T_t)$$

donde T_t es el subárbol en el nodo t



La Lógica del Pruning

- ¿Cuándo Podar?

Eliminamos una rama cuando el costo de mantenerla supera el beneficio que aporta a la precisión

- Encontrando el Balance

Ajustamos gradualmente cuánto valoramos la simplicidad vs. la precisión hasta encontrar el punto óptimo

Random Forest





Random Forest: El Poder del Conjunto

Los Random Forests son como un consejo de sabios donde cada árbol aporta su opinión basada en su propia experiencia única.



¿Cómo Funciona?

- Crea múltiples árboles con variaciones
- Cada árbol ve datos diferentes (Bootstrap)
- Cada árbol considera variables diferentes
- La decisión final es por votación



Elementos Aleatorios

- **Bootstrap:** Muestreo con reemplazo
- **Feature Sampling:** Subconjunto de variables
- **Agregación:** Promedio o votación mayoritaria



Random Forest: Aspectos Técnicos



Fórmulas Clave

- Out-of-Bag Error:

$$OOB = \frac{1}{n} \sum_{i=1}^n I(\hat{y}_i^{OOB} \neq y_i)$$

- Importancia de Variables:

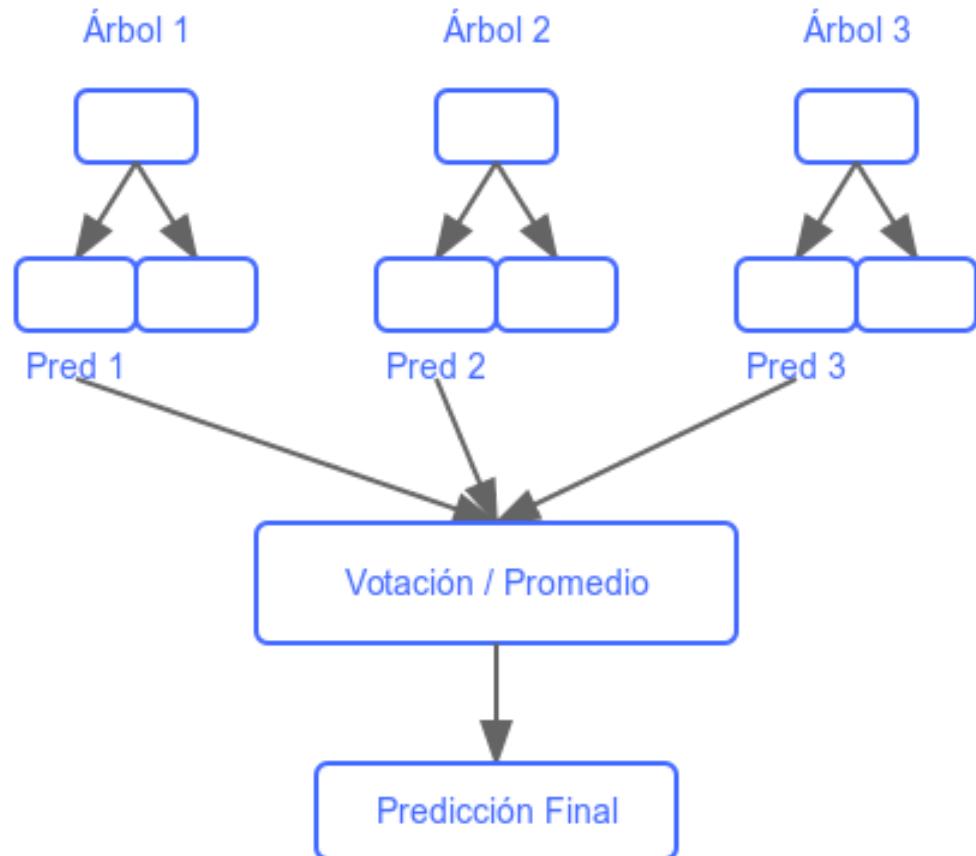
$$Imp(X_j) = \frac{1}{N_t} \sum_{t=1}^{N_t} \Delta I_j^t$$



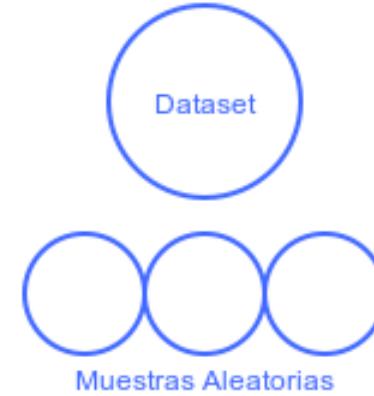
Intuición de las Métricas

- Out-of-Bag Error: Mide el error usando muestras que no se usaron para entrenar cada árbol
- Importancia de Variables: Cuánto empeora el modelo cuando "desordenamos" una variable
- Bootstrap: Aproximadamente 63.2% de los datos originales aparecen en cada muestra

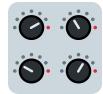
Visualización Random Forest



Características del Random Forest



- Cada árbol usa una muestra aleatoria
- Diferentes variables por árbol
- Reduce el sobreajuste



Los Hiperparámetros Clave

Los dos controles principales de nuestro Random Forest determinan el balance entre aprendizaje y generalización:



n_estimators

- Define el tamaño del "comité votante"
- Más árboles = mayor estabilidad
- Valores típicos: 100-1000
- Aumentarlo no causa overfitting



max_depth

- Controla cuánto puede aprender cada árbol
- Muy bajo → underfitting
- Muy alto → overfitting



Estrategia común: Comenzar con `n_estimators=100, max_depth=None, min_samples_leaf=5` y ajustar según validación cruzada



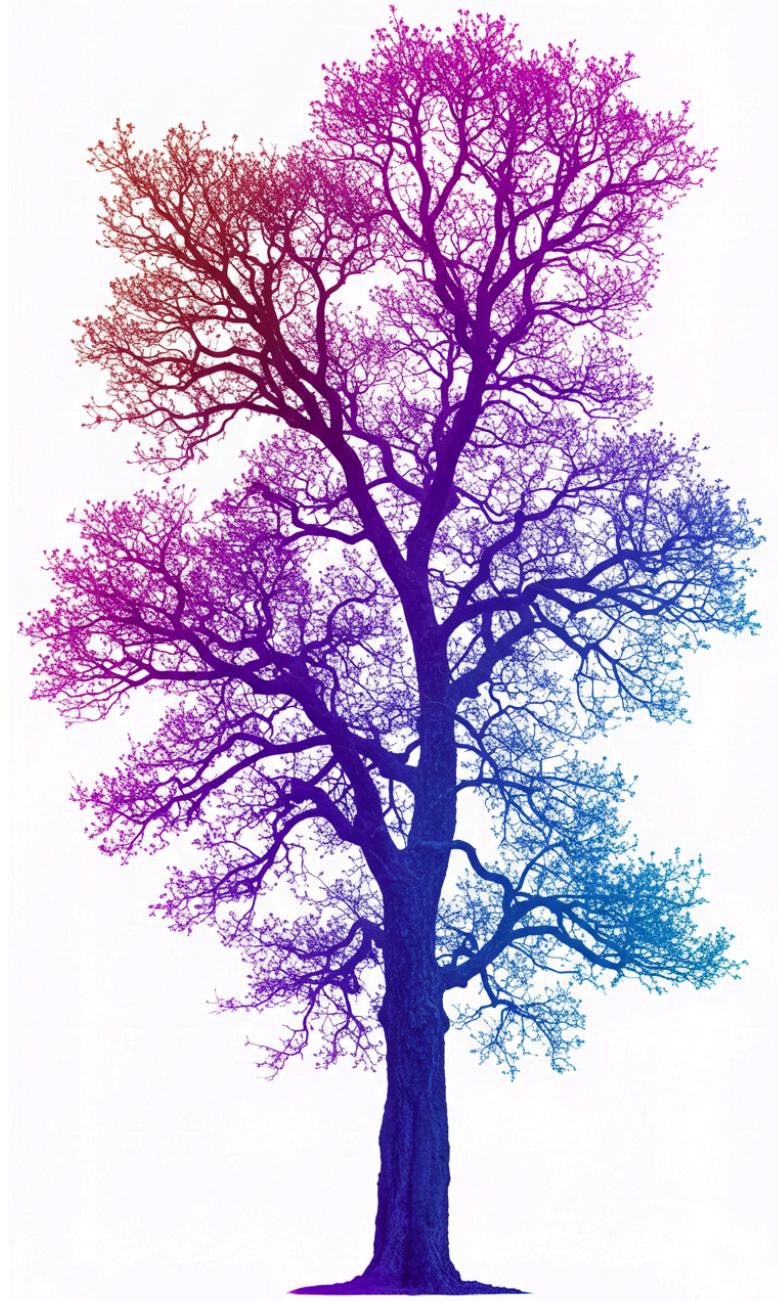
Pruning en Random Forest: La Realidad Práctica

La poda en Random Forest es fundamentalmente diferente a la de los árboles individuales. El propio diseño del algoritmo incorpora **mecanismos naturales contra el sobreajuste** a través de la **agregación de múltiples árboles** y la **selección aleatoria de características** en cada split.

Por esto, nos enfocamos en controlar el crecimiento inicial en lugar de podar:

- Ajustamos `max_features` para las variables por split
- Usamos `min_samples_split` para divisiones significativas
- Priorizamos `n_estimators` sobre técnicas de poda

XGBoost





XGBoost: La Evolución del Bosque

XGBoost lleva el aprendizaje con árboles al siguiente nivel, combinando múltiples modelos con un aprendizaje secuencial e inteligente.

Imagina un equipo de estudiantes donde cada uno se especializa en lo que sus compañeros no dominan:

- El primero aprende lo básico
- El siguiente se enfoca en los temas difíciles
- Cada nuevo estudiante corrige los errores que persisten
- Al final, toman decisiones en conjunto, priorizando a los más precisos

Esta es la esencia de XGBoost: un sistema donde cada árbol nuevo se especializa en corregir los errores de los anteriores, creando un equipo de expertos altamente eficiente.



XGBoost: La Matemática del Boosting



Fórmulas Clave

- Objetivo de Optimización:

$$\mathcal{L} = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k)$$

- Expansión de Taylor sobre los residuos:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^n [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t)$$

donde:

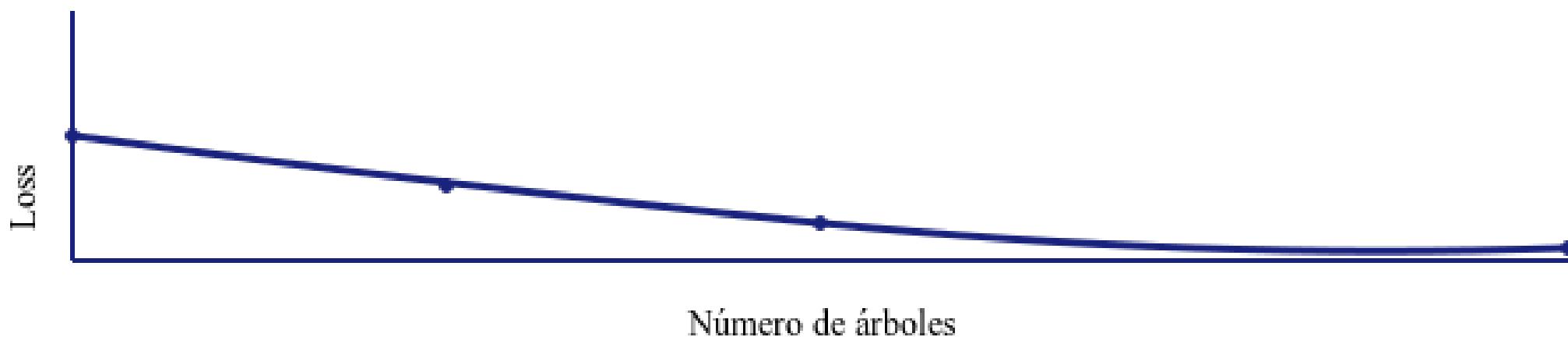
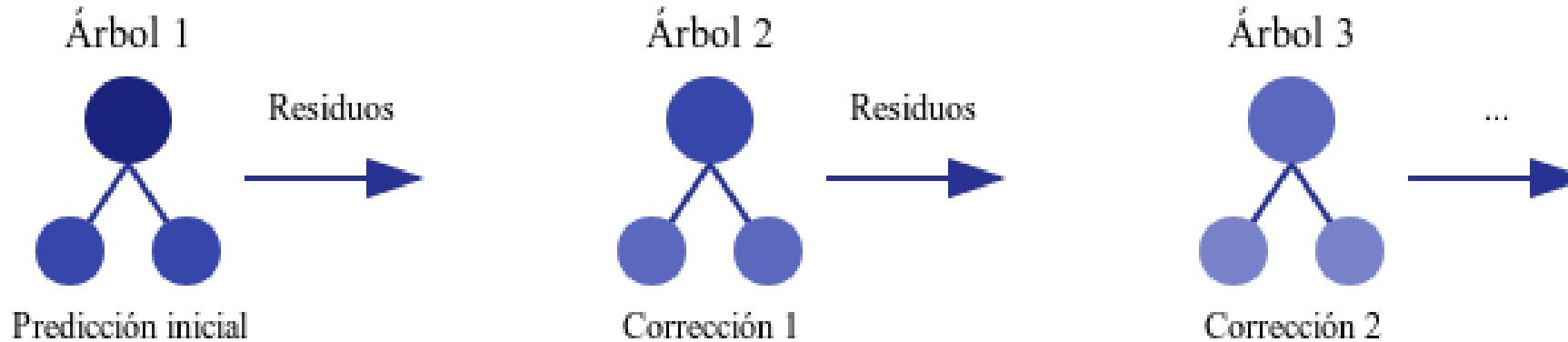
- $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$: gradiente del error
- $h_i = \partial_{\hat{y}^{(t-1)}}^2 l(y_i, \hat{y}^{(t-1)})$: segunda derivada



La Intuición

- La función objetivo equilibra precisión y simplicidad.
- ! La expansión de Taylor guía la construcción de cada árbol para minimizar el error residual (función de pérdida).

Proceso de Aprendizaje Secuencial XGBoost



Función de Pérdida en XGBoost



La función de pérdida es una medida matemática que cuantifica la diferencia entre las predicciones del modelo y los valores reales, junto con un término de regularización que controla la complejidad del modelo. En XGBoost, esta función se optimiza usando una expansión de Taylor de segundo orden, donde la regularización considera específicamente el número de hojas y sus pesos.

Rol en XGBoost

- Guía el aprendizaje: decide cómo ajustar cada nuevo árbol
- Equilibra precisión vs complejidad: evita el sobreajuste
- Optimización por gradiente: cada árbol nuevo minimiza esta función
- Monitoreo: su valor decreciente indica que el modelo mejora

XGBoost: Afinando el Modelo



Hiperparámetros de Aprendizaje

- **learning_rate (η):** 0.01-0.3
 - Ajusta el tamaño de cada paso de aprendizaje
 - Valores más pequeños requieren más árboles
- **n_estimators & early_stopping:**
 - n_estimators: número máximo de árboles
 - early_stopping_rounds: detiene el entrenamiento cuando el error no mejora en N rondas
 - Mejor usar ambos para evitar overfitting y optimizar recursos



XGBoost: La Profundidad del Aprendizaje

La magia de XGBoost está en su uso de la segunda derivada:

$$h_i = \frac{\partial^2 L}{\partial \hat{y}_i^2}$$

Es como un GPS que no solo te dice "gira", sino también "con qué intensidad girar".

Esta información adicional permite a XGBoost:

- Detectar cuando una predicción parece "sospechosa", ajustando automáticamente cuánto aprender de cada error
- Manejar naturalmente los datos atípicos, como un profesor que distingue entre un despiste y un error conceptual

En esencia, XGBoost no solo ve el tamaño del error, sino que entiende su "personalidad" - haciéndolo excepcionalmente bueno con datos del mundo real.

Recursos del Curso

Plataformas y Enlaces Principales

 GitHub del curso

 github.com/CamiloVga/IA_Aplicada

 Asistente IA para el curso

 [Google Notebook LLM](#)