Árboles de Decisión y Random Forest

Profesor: Juan Gamarra Moreno

Introducción a los Árboles de Decisión

Árbol de decisión

- Definición: Un árbol de decisión es un modelo predictivo que utiliza una estructura de árbol para tomar decisiones. Cada nodo interno representa una prueba o decisión basada en una característica (feature) específica, y cada rama representa el resultado de esa prueba. Las hojas finales del árbol representan las predicciones o resultados.

Utilidad del árbol de decisión

- En **clasificación**, el árbol asigna una clase a los datos de entrada.
- En **regresión**, el árbol predice un valor numérico.
- Son fáciles de interpretar visualmente y se utilizan en una variedad de áreas como análisis de riesgo, diagnóstico médico y reconocimiento de patrones.

Árbol binario y nodos

- Un **árbol binario** tiene nodos que dividen los datos en dos ramas. En cada nodo, se toma una decisión binaria en función de una característica, por ejemplo, "¿El peso es mayor que 50 kg?".
- Cada nodo del árbol representa una decisión basada en una característica específica de los datos (por ejemplo, la altura, la edad o el precio), dividiendo el conjunto de datos en subconjuntos más pequeños.

Árbol binario y nodos

- **Nodos de decisión**: Estos son los puntos en los que el árbol se ramifica basado en una característica específica. Se realiza una comparación con un valor umbral para decidir qué rama seguir.
- **Hojas**: Son los nodos terminales que no se dividen más, y contienen las predicciones finales.

Construcción de un árbol

- El **algoritmo de construcción** de un árbol de decisión sigue un enfoque recursivo, dividiendo el conjunto de datos en ramas más pequeñas utilizando un criterio de división hasta que no se puede dividir más (cuando los datos están completamente separados o se alcanza un límite predefinido como la profundidad del árbol).
- **Recursividad**: El proceso de dividir los datos continúa hasta que se cumple una condición de parada, como alcanzar una profundidad máxima del árbol o tener muy pocos datos en un nodo.

Construcción de un árbol

Criterios de división:

- **Índice Gini**: Mide la "impureza" de un nodo. Un nodo es puro si todos los datos que contiene pertenecen a una misma clase. El índice Gini se utiliza para determinar la mejor división. Su fórmula es:

$$Gini = 1 - \sum p_i^2$$

  donde p_i es la proporción de observaciones en la clase i.

- **Entropía e información**: La entropía mide la incertidumbre o aleatoriedad en los datos. Se utiliza para seleccionar la mejor división, buscando maximizar la reducción de la entropía (ganancia de información). Su fórmula es:

$$Entropía = -\sum p_i \log_2(p_i)$$

  La **ganancia de información** se calcula restando la entropía promedio de los nodos hijos de la entropía del nodo padre.

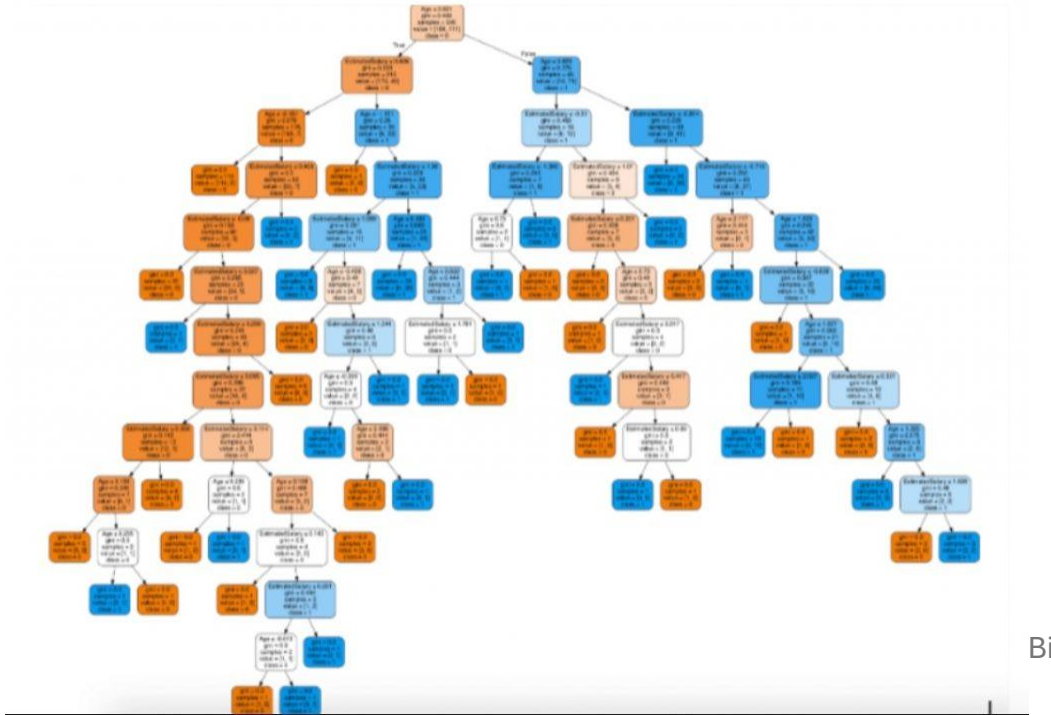
Sobreajuste en Árboles de Decisión y Técnicas para Evitarlo

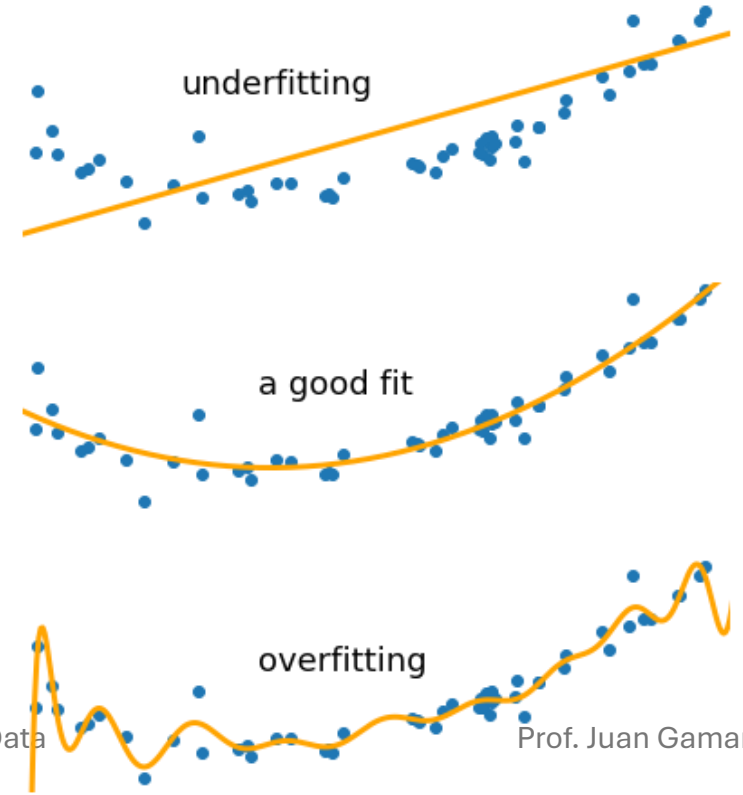
Sobreajuste (overfitting) en árboles de decisión

- **Definición de sobreajuste**: El sobreajuste ocurre cuando un modelo se ajusta demasiado bien a los datos de entrenamiento, capturando tanto los patrones relevantes como el ruido o las particularidades del conjunto de datos. Esto conduce a un bajo error en el conjunto de entrenamiento, pero un alto error en los datos nuevos (validación o prueba), lo que reduce la capacidad de generalización del modelo.
- En **árboles de decisión**, el sobreajuste se manifiesta cuando el árbol crece demasiado, creando ramas que explican en exceso los datos de entrenamiento, incluso los puntos que no son representativos del patrón general. Esto puede ocurrir cuando no hay restricciones en el crecimiento del árbol, dividiendo los datos hasta que cada observación individual es aislada en un nodo hoja.

Sobreajuste (overfitting) en árboles de decisión

- Un árbol sobreajustado tendrá muchas ramas y será altamente específico a los datos de entrenamiento, mientras que un árbol que generaliza mejor será más pequeño y podrá manejar variaciones en datos nuevos.





Poda de árboles

Poda de árboles: Es una técnica que consiste en eliminar ramas del árbol que no aportan valor en la predicción y que probablemente contribuyen al sobreajuste.

Tenemos 2 alternativas de poda:

- Pre-poda
- Post-poda

Poda de árboles

Pre-poda (Pre-pruning): Implica detener el crecimiento del árbol antes de que esté completamente desarrollado. Esto se hace imponiendo restricciones como:

- **Profundidad máxima del árbol**: Limita el número de niveles del árbol.
- **Número mínimo de muestras por nodo**: Requiere que un nodo tenga un número mínimo de datos para que se divida.
- **Máximo número de nodos hoja**: Limita la cantidad de nodos terminales en el árbol.
- Ventaja: Es rápido de implementar.
- Desventaja: Puede detenerse antes de alcanzar un punto óptimo.

Poda de árboles

Post-poda (Post-pruning): Permite que el árbol crezca completamente y luego se eliminan ramas irrelevantes o que agregan complejidad innecesaria.

- **Poda por reducción de error**: Se compara el rendimiento del árbol con y sin ciertas ramas. Si una rama no mejora significativamente el rendimiento del modelo en datos de validación, se elimina.
- Ventaja: Ofrece más control sobre el ajuste del modelo y permite un ajuste más fino.
- Desventaja: Requiere más tiempo de cómputo, ya que el árbol se construye completamente antes de podarlo.

Técnicas para evitar el sobreajuste

- **Poda (Pruning)**:
  - Pre-poda y post-poda son las dos técnicas principales de poda, como se explicó anteriormente. El objetivo es reducir la complejidad del modelo, eliminando nodos que aportan poca o ninguna mejora a la capacidad predictiva del modelo en datos nuevos.
- **Restricción de la profundidad del árbol**:
  - Limitar la **profundidad máxima del árbol** asegura que el árbol no se ramifique excesivamente, lo que ayuda a mantener su capacidad de generalización. Por ejemplo, un árbol con una profundidad limitada a 5 niveles es menos probable que se sobreajuste en comparación con un árbol sin restricciones de profundidad.
  - Scikit-learn permite establecer esta restricción con el hiperparámetro `max_depth`.

Técnicas para evitar el sobreajuste

- **Ajuste del número mínimo de muestras por nodo**:
  - Requiere que un nodo tenga un número mínimo de muestras antes de que pueda dividirse. Esto evita que el árbol se divida en nodos con muy pocas observaciones, lo que normalmente lleva a divisiones específicas y sobreajuste.
  - Este ajuste se controla con los hiperparámetros `min_samples_split` (mínimo de muestras para dividir un nodo) y `min_samples_leaf` (mínimo de muestras requeridas en un nodo hoja) en Scikit-learn.
  - Ejemplo: Si `min_samples_split=10`, un nodo solo se dividirá si contiene al menos 10 muestras, lo que reduce la probabilidad de divisiones que introduzcan ruido en lugar de un patrón general.

Introducción a Random Forest

Ensemble learning

- Es una técnica de aprendizaje automático que combina las predicciones de múltiples modelos para obtener un resultado final más robusto y preciso. La idea es que un conjunto de modelos ("ensemble") pueda corregir los errores que podría cometer un solo modelo.

Random Forest

Random Forest: Es un tipo específico de ensemble que utiliza múltiples **árboles de decisión** como modelos base. A diferencia de un solo árbol de decisión, que es propenso al sobreajuste, Random Forest genera una serie de árboles y promedia sus predicciones para obtener un resultado más estable y preciso.

- Para problemas de **clasificación**, Random Forest toma la decisión final mediante votación mayoritaria: cada árbol "vota" por una clase, y la clase que obtiene más votos se elige como la predicción.
- Para problemas de **regresión**, Random Forest promedia las predicciones numéricas de todos los árboles para obtener la predicción final.

Random Forest

- **Ventaja clave**: Al promediar los resultados de múltiples árboles, Random Forest reduce la varianza, lo que mejora la generalización y disminuye el riesgo de sobreajuste.

Construcción de cada árbol

- **Bootstrap Aggregating (Bagging)**:
  - En **bagging**, se crean diferentes subconjuntos del conjunto de datos original mediante un proceso de muestreo con reemplazo (es decir, algunos datos pueden aparecer más de una vez en un subconjunto mientras que otros pueden no aparecer).
  - Para cada subconjunto, se entrena un árbol de decisión. Debido a que cada árbol ve un conjunto de datos ligeramente diferente, los errores cometidos por un árbol no necesariamente serán los mismos que los de otro árbol. Esto ayuda a mejorar la robustez del modelo global.
  - Una vez que todos los árboles han sido entrenados, las predicciones de todos los árboles se combinan (por votación en clasificación o por promediación en regresión).

Construcción de cada árbol

- **Selección aleatoria de características**:
  - En lugar de utilizar todas las características para entrenar cada árbol de decisión, Random Forest selecciona de manera aleatoria un subconjunto de características en cada división del árbol. Esto asegura que los árboles no estén fuertemente correlacionados entre sí, lo que mejora la diversidad del ensemble y reduce la probabilidad de sobreajuste.
  - En Scikit-learn, este parámetro se ajusta con `max_features`. Por ejemplo, si tienes 10 características, podrías seleccionar 3 al azar para cada división de árbol, evitando que un árbol siempre se incline por las características más informativas del conjunto.

Clasificación con Random Forest

- En clasificación, Random Forest puede manejar conjuntos de datos desequilibrados y resolver problemas con alta dimensionalidad, seleccionando automáticamente las características más relevantes en cada árbol.
- Su capacidad para reducir el sobreajuste lo hace más efectivo que un solo árbol de decisión, especialmente en conjuntos de datos complejos.
- Ventaja clave: Random Forest tiende a tener una alta precisión en clasificación porque puede reducir los errores de generalización que un árbol de decisión único podría cometer.

Regresión con Random Forest

- Para problemas de regresión, Random Forest también es muy efectivo para manejar relaciones no lineales complejas. Promedia las predicciones de múltiples árboles, lo que mejora la precisión al suavizar las predicciones.
- Comparado con los árboles de decisión, Random Forest es menos propenso a generar modelos que se sobreajustan al conjunto de entrenamiento, ya que la variabilidad entre los árboles ayuda a eliminar el impacto del ruido.

Ventajas de Random Forest

- **Reducción del sobreajuste**: Al combinar varios árboles, Random Forest es menos propenso a ajustarse al ruido presente en los datos de entrenamiento.
- **Mayor precisión**: Dado que se promedian las predicciones, el modelo tiene una mejor capacidad para generalizar a nuevos datos.
- **Robustez**: Funciona bien incluso con datos faltantes, ya que cada árbol en el conjunto está entrenado en un subconjunto diferente de los datos.
- **Importancia de características**: Random Forest ofrece un método incorporado para calcular la **importancia de las características** (feature importance), que indica cuáles son las variables más útiles para hacer predicciones.