
RESULTADO DE APRENDIZAJE

RdA de la asignatura:

- **RdA 2:** Aplicar modelos de aprendizaje automático supervisado y no supervisado, así como su validación y optimización, en la resolución de problemas tanto reales como simulados.

RdA de la clase:

- Comprender los fundamentos teóricos de los Árboles de Decisión y su uso en clasificación y regresión.
- Analizar los criterios principales asociados con la construcción y optimización de árboles de decisión (parada, selección, clasificación, partición y poda).
- Implementar un Árbol de Decisión en Python mediante un cuaderno de Jupyter.

INTRODUCCIÓN

Pregunta inicial: ¿Por qué es importante la interpretabilidad en los modelos de aprendizaje automático y cómo los Árboles de Decisión nos ayudan a lograrla?

DESARROLLO

Actividad 1: Retroalimentación de la clase invertida

Se conecta lo aprendido en casa con la utilidad práctica de los Árboles de Decisión en tareas de clasificación y regresión. Esta actividad busca reforzar los conceptos clave y vincularlos con aplicaciones prácticas para destacar la importancia de los Árboles de Decisión como modelos interpretables y efectivos.

¿Cómo lo haremos?

- **Resumen rápido:** Breve recapitulación de conceptos clave:
 - Interpretabilidad de los Árboles de Decisión mediante reglas basadas en características.
 - Entropía e información ganada como métricas para la construcción del árbol.
- **Discusión:** Reflexión grupal sobre aplicaciones reales y decisiones de diseño:
 - ¿Cuándo es preferible un Árbol de Decisión frente a otros modelos?

- ¿Cómo afecta la poda al rendimiento y la interpretabilidad del árbol?
- Identificación de ejemplos prácticos de su uso, como la clasificación de clientes o predicciones en salud.

Verificación de aprendizaje:

- ¿Qué representa la ganancia de información en un Árbol de Decisión?
- ¿Qué ventajas tiene la interpretabilidad de un Árbol de Decisión?

Actividad 2: Clase magistral sobre criterios y poda**¿Cómo lo haremos?****• Explicación teórica:**

- Criterio de parada: ¿Cuándo detener la división?
- Criterio de selección: Métricas como ganancia de información, índice Gini.
- Criterio de clasificación: Cómo asignar etiquetas finales a las hojas.
- Criterio de partición: Estrategias para dividir datos en subconjuntos.

- **Discusión guiada:** Comparación de criterios según el tipo de problema (clasificación vs regresión).

Verificación de aprendizaje:

- ¿Por qué es importante el criterio de parada?

Actividad 3: Implementación práctica de Árboles de Decisión

Esta actividad se enfoca en aplicar Árboles de Decisión utilizando Python en un cuaderno de Jupyter. Los estudiantes experimentarán con criterios de partición, poda y parámetros como la profundidad máxima, conectando teoría con aplicaciones prácticas en tareas de clasificación.

¿Cómo lo haremos?

- **Implementación en Python:** Los estudiantes trabajarán en un cuaderno de Jupyter previamente preparado.

Enlace al cuaderno: [16-Arboles-Decision.ipynb](#).

- **Experimentación guiada:** Los estudiantes modificarán parámetros del Árbol de Decisión:
 - Cambiar el criterio de partición ('gini' a 'entropy').
 - Ajustar la profundidad máxima (max_depth).
 - Probar con diferentes tamaños de conjuntos de datos.

Verificación de aprendizaje:

- ¿Cómo afectan los diferentes criterios de partición a la estructura del árbol?
- ¿Qué impacto tiene limitar la profundidad máxima en el modelo?
- ¿Cómo podemos interpretar los resultados visuales del árbol?

CIERRE

Tarea: Implementar un Árbol de Decisión utilizando un conjunto de datos real y comparar los resultados al variar los criterios de partición, poda y profundidad máxima.

Pregunta de investigación:

1. ¿Qué significa podar un árbol de decisión? ¿Cómo se poda un árbol?

Para la próxima clase: Visualizar el siguiente video sobre Bosques Aleatorios:

Enlace al video: [StatQuest: Random Forests Part 1](#).