

Instituto Tecnológico de Costa Rica - TEC

Inteligencia Artificial

Docente: Kenneth Obando Rodríguez

✓ Trabajo Corto 3: Árboles de Decisión

Estudiantes:

- Estudiante 1
- Estudiante 2
- Estudiante 3

Link del Cuaderno (recuerde configurar el acceso a público):

- [Link de su respuesta](#)

Nota: Este trabajo tiene como objetivo promover la comprensión de la materia y su importancia en la elección de algoritmos. Los alumnos deben evitar copiar y pegar directamente información de fuentes externas, y en su lugar, demostrar su propio análisis y comprensión.

Entrega

Debe entregar un archivo comprimido por el TecDigital, incluyendo un documento pdf con los resultados de los experimentos y pruebas. La fecha de entrega es el domingo 17 de setiembre, antes de las 10:00pm.

Instrucciones:

Las alternativas se rifarán en clase utilizando números aleatorios. Deberá realizar la asignación propuesta. Si realiza ambos ejercicios, recibirá 20 puntos en **la nota porcentual de la actividad**, para aplicar a la totalidad de los puntos extra es necesario que ambas actividades se completen al 100%

Actividad - Taller

1. Cree una clase nodo con atributos necesarios para un árbol de decisión: feature, umbral, gini, cantidad_muestras, valor, izquierda, derecha
2. Crea una clase que implementa un árbol de decisión, utilice las funciones presentadas en clase, además incluya los siguientes hyperparámetros:

- `max_depth`: Cantidad máxima de variables que se pueden explorar
 - `min_split_samples`: Cantidad mínima de muestras que deberá tener un nodo para poder ser dividido
 - `criterio`: función que se utilizará para calcular la impuridad.
3. Divida los datos en los conjuntos tradicionales de entrenamiento y prueba, de forma manual, sin utilizar las utilidades de `sklearn` (puede utilizar índices de `Numpy` o `Pandas`)
 4. Implemente una función que se llame `validacion_cruzada` que entrene k modelos y reporte las métricas obtenidas:
 - a. Divida el conjunto de entrenamiento en k subconjuntos excluyentes
 - b. Para cada uno de los k modelos, utilice un subconjunto como validación
 - c. Reporte la media y la desviación estándar para cada una de las métricas, todo debe realizarse solo usando `Numpy`:
 - Accuracy
 - Precision
 - Recall
 - F1
 5. Entrene 10 combinaciones distintas de parámetros para su implementación de Arbol de Decisión y utilizando su implementación de `validacion_cruzada`.
 6. Utilizando los resultados obtenidos analice cuál y porqué es el mejor modelo para ser usado en producción.
 7. Compruebe las métricas usando el conjunto de prueba y analice el resultado

Rúbrica para la Implementación de un Árbol de Decisión

Nota: Esta rúbrica se basa en la calidad de la implementación y los resultados obtenidos, no en la cantidad de código.

1. Creación de la Clase `Nodo` (10 puntos)

- ☐ Se crea una clase `Nodo` con los atributos mencionados en las especificaciones (`feature`, `umbral`, `gini`, `cantidad_muestras`, `valor`, `izquierda`, `derecha`).
- ☐ Los atributos se definen correctamente y se asignan de manera apropiada.

2. Creación de la Clase `Árbol de Decisión` (20 puntos)

- ☐ Se crea una clase que implementa un árbol de decisión.
- ☐ La clase utiliza las funciones presentadas en el cuaderno.
- ☐ Se implementan los hiperparámetros solicitados (`max_depth`, `min_split_samples`, `criterio`).

- ☐ La clase es capaz de entrenar un árbol de decisión con los hiperparámetros especificados.

3. División de Datos (10 puntos)

- ☐ Los datos se dividen en conjuntos de entrenamiento y prueba de forma manual.
- ☐ Se utiliza Numpy o Pandas para realizar esta división.
- ☐ Se garantiza que los conjuntos sean excluyentes.

4. Implementación de Validación Cruzada (20 puntos)

- ☐ Se implementa la función `validacion_cruzada` correctamente.
- ☐ Los datos de entrenamiento se dividen en k subconjuntos excluyentes.
- ☐ Se entrena y evalúa un modelo para cada subconjunto de validación.
- ☐ Se calculan y reportan las métricas de accuracy, precision, recall y F1.
- ☐ Se calcula la media y la desviación estándar de estas métricas.

5. Entrenamiento de Modelos (20 puntos)

- ☐ Se entrenan 10 combinaciones distintas de parámetros para el árbol de decisión.
- ☐ Cada combinación se entrena utilizando la función `validacion_cruzada`.
- ☐ Los resultados de las métricas se registran adecuadamente.

6. Análisis de Modelos (10 puntos)

- ☐ Se analizan los resultados obtenidos y se selecciona el mejor modelo para ser utilizado en producción.
- ☐ Se proporciona una justificación clara y fundamentada sobre por qué se eligió ese modelo.

7. Prueba en el Conjunto de Prueba (10 puntos)

- ☐ Se comprueban las métricas del modelo seleccionado utilizando el conjunto de prueba.
- ☐ Se analizan los resultados y se comentan las conclusiones.

General (10 puntos)

- ☐ El código se documenta de manera adecuada, incluyendo comentarios que expliquen las secciones clave.
- ☐ El código se ejecuta sin errores y sigue buenas prácticas de programación.
- ☐ La presentación de los resultados es clara y fácil de entender.
- ☐ Se cumple con todos los requisitos y las especificaciones proporcionadas.

Puntuación Total: 100 puntos

