Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Estado de México

TC3006C
Inteligencia Artificial Avanzada para la Ciencia de Datos I

Módulo 2
Aprendizaje automático

Implementación de una técnica de aprendizaje máquina sin el uso de un framework.

Evidencia del portafolio de implementación

Profesores

Mtro. Alberto Michel Pérez Domínguez

Dra. Andrea Torres Calderón

Mtro. David Higuera Rosales

Dra. Elisabetta Crescio

Dr. Jorge Adolfo Ramirez Uresti

Dr. Julio Guillermo Arriaga Blumenkron

Dr. Victor Adrián Sosa Hernández

Profesor del módulo
Dr. Jorge Adolfo Ramírez Uresti

Grupo 501

Maximiliano De La Cruz Lima A01798048

5 – Septiembre – 2025

1. Introducción

El objetivo de esta evidencia es la implementación de un algoritmo de aprendizaje automático sin el uso de frameworks avanzados que permitan importar algoritmos ya implementados, utilizando únicamente NumPy y Pandas para dicha labor. Para ello, se optó por el desarrollo de un clasificador basado en Árboles de Decisión, empleando el criterio de entropía y ganancia de información para la selección de los mejores puntos de división en cada nodo.

Las pruebas de desempeño del algoritmo se realizaron con el dataset Iris, un conjunto de datos ampliamente utilizado en el ámbito del aprendizaje automático, el cual contiene 150 muestras de flores distribuidas en tres clases: *Setosa*, *Versicolor* y *Virginica*. Cada muestra está descrita por cuatro características: la longitud y el ancho de los sépalos, así como la longitud y el ancho de los pétalos.

2. Metodología

a) Dataset empleado

En primer lugar, se dividió el dataset de manera aleatoria con una proporción del 15% para validación y otro 15% para prueba, contando con la siguiente cantidad de muestras para cada conjunto:

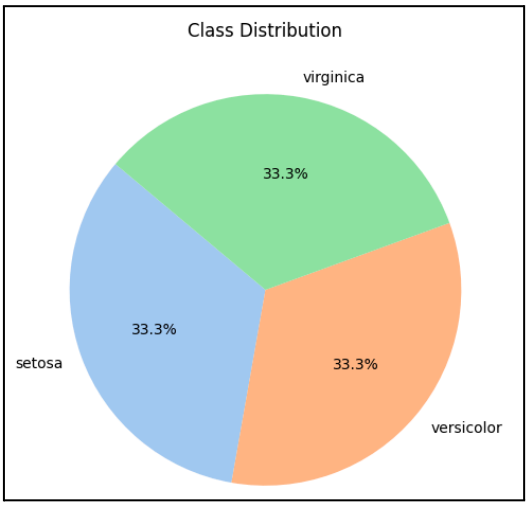
- Conjunto de entrenamiento: 107 muestras
- Conjunto de validación: 20 muestras
- Conjunto de pruebas: 23 muestras

b) Preprocesamiento

Antes de entrenar el modelo se revisó la distribución de las clases en el dataset, con el fin de identificar posibles desbalances entre ellas. Este análisis mostró que las tres clases están distribuidas de manera perfectamente balanceada.

Figura 1

Distribución de las clases en el dataset



Este balance garantiza que la métrica de accuracy sea una medida adecuada para evaluar el desempeño del modelo ya que, de haber un claro desbalance, esta medida puede resultar engañosa, ya que basta con predecir la clase mayoritaria para obtener valores altos. Sin embargo, en este caso el balance asegura un accuracy que refleje la calidad de las predicciones.

Por otro lado, no fue necesario aplicar alguna estandarización de características, esto debido a que los árboles de decisión dividen el espacio de características en función del orden relativo de los valores, por lo que no se ven afectados por la escala de de los datos. Por lo tanto, normalizar o escalar las variables no aporta beneficio alguno en el desempeño del modelo..

De igual manera, tampoco se requirió emplear algún método de codificación numérica de etiquetas dado que, en el caso particular del dataset de Iris, en scikit-learn ya se cuenta con las clases en formato numérico (0 = *Setosa*, 1 = *Versicolor*, 2 = *Virginica*).

c) Modelo implementado

Como se mencionó al principio, el modelo implementado corresponde a a un clasificador de árbol de decisión desarrollado sin emplear librerías especializadas en modelos de aprendizaje automático o estadśitica avanzada, estableciendo los siguientes parámetros:

- *min_sample_split:* número mínimo de muestras requeridas para dividir un nodo.
- *max_depth:* profundidad máxima permitida para el árbol.
- *Entropia* como criterio de impureza.
- *Ganancia de información* como criterio de selección de umbral.

d) Métricas de evaluación

El desempeño del modelo implementado se evaluó haciendo uso de las siguientes métricas:

- *Matriz de confusión:* para visualizar los aciertos y errores por clase.
- *Precision, Recall y F1-score:* para medir equilibrio entre falsos positivos y negativos.
- *Accuracy:* para medir el desempeño general en los conjuntos de validación y prueba.

e) Criterios de parada

Asimismo, se establecieron los siguientes criterios de parada para controlar la complejidad del modelo y evitar sobreajuste:

- *Número mínimo de muestras para dividir* ($min\_sample\_split = 3$) → evitando que nodos con pocos datos generen divisiones poco representativas.
- *Profundidad máxima del árbol* ($max\_depth = 5$) → limita el crecimiento excesivo del árbol, previniendo que memorice los datos de entrenamiento.
- *Pureza del nodo* → si todas las muestras de un nodo pertenecen a la misma clase, se detienen la división y se asigna esa clase como hojas.

3. Fundamentos matemáticos

Este tipo de clasificador basado en árboles de decisión divide el espacio de característica de manera recursiva, de tal manera que para cada división se selecciona en función de la métrica de *entropía* y su correspondiente *ganancia de información*, con el objetivo de encontrar nodos cada vez más "puros", es decir, con la mayor proporción de ejemplos de una sola clase.

a) Entropía

La entropía permite medir el grado de impureza de un conjunto de ejemplos en un nodo:

$$H(S) = -\sum_{c=1}^{K} p_c log_2(p_c) \text{ - ec. (1)}$$

donde:

- $p_c \rightarrow$ proporción de instancias de las clase c en el subconjunto S.
- $K \rightarrow$ número de clases.
- Si $H(S) = 0 \rightarrow$ todas las instancias pertenecen a una sola clase
- Si $H(S) = máxima\ ganancia \rightarrow$ las instancias están distribuidas uniformemente.

Por ejemplo, véase el caso de un nodo donde hay 10 muestras distribuidas como 4 *Setosa*, 3 *Versicolor*, 3 *Virginica:*

$$H = -\left(\frac{4}{10} log_2(\frac{4}{10}) + \frac{3}{10} log_2(\frac{3}{10}) + \frac{3}{10} log_2(\frac{3}{10})\right) \approx 1.57$$

b) Ganancia de información

Cuando se divide un nodo padre P en dos hijos L y R usando un umbral t, se calcula la ganancia de información como:

$$IG(P,t) = H(P) - \left(\frac{|L|}{|P|} H(L) + \frac{|R|}{|P|} H(R)\right) \text{ - ec. (2)}$$

donde.

- $|L|, |R| \rightarrow$ número de ejemplos en los nodos hijo.
- $H(L), H(R) \rightarrow$ entropías respectivas de los nodos hijo.

Por lo tanto, el algoritmo busca un umbral t y la característica que maximicen IG

Continuando con el ejemplo anterior donde el nodo padre tiene una entropía $H(P) = 1.57$, al dividirse los los hijos tienen una entropía de $H(L) = 0.0$ y $H(R) = 0.97$ puesto que todos los ejemplos corresponden a Setosa, entonces la ganancia de información queda:

$$IG = 1.57 - \left((\frac{4}{10}) \cdot 0.0 + (\frac{6}{10}) \cdot 0.97\right) = 0.988$$

Donde este valor positivo indica que la división reduce la impureza del nodo y, por lo tanto, se considera un buen split.

4. Resultados

a) Conjunto de validación

Figura 2

Matriz de confusión para el conjunto de validación

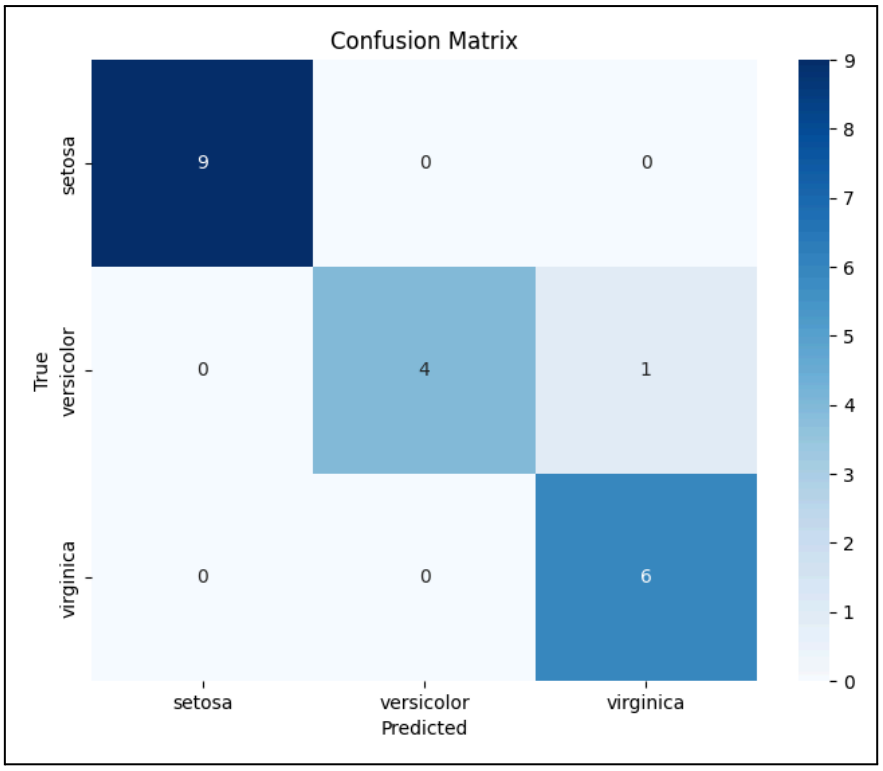


Tabla 1

Reporte de clasificación con el conjunto de validación

	precision	recall	f1-score	support	accuracy
setosa	1.0	1.0	1.0	9	
versicolor	1.0	0.80	0.89	5	
virginica	0.86	1.0	0.92	6	
					95.00%

b) Conjunto de prueba

Figura 3

Matriz de confusión para el conjunto de prueba

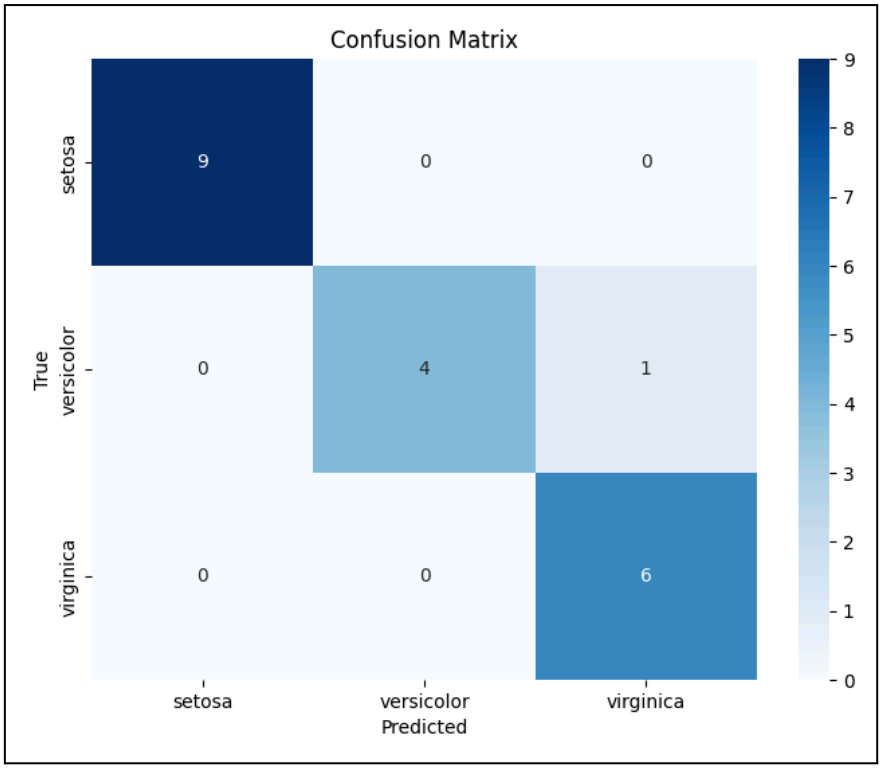


Tabla 2

Reporte de clasificación con el conjunto de prueba

	precision	recall	f1-score	support	accuracy
setosa	1.0	1.0	1.0	8	
versicolor	0.75	1.0	0.86	6	
virginica	1.0	0.78	0.89	9	
					91.30%

5. Análisis de resultados

- El modelo obtuvo un desempeño sobresaliente en la clase Setosa, clasificando correctamente todas las instancias tanto en validación como prueba.
- En contraste, los errores del modelo se concentraron entre Versicolor y Virginica.
- El modelo logró 95% de exactitud en validación y 91.30% en prueba, indicando una buena capacidad de generalización.
- Los cálculos de la entropía y la ganancia de información fueron efectivos en la construcción de nodos de alta pureza, reduciendo progresivamente la incertidumbre en la clasificación.
- Los criterios de parada ayudaron a mantener un balance adecuado entre sesgo y varianza:
    - El árbol no fue tan simple para caer en un subajuste (alto sesgo).
    - Tampoco fue tan complejo para memorizar los datos de entrenamiento (alta varianza).
- El modelo muestra un sesgo bajo (buen desempeño promedio) y una varianza controlada (pequeña diferencia entre validación y prueba).

6. Conclusiones

- El árbol de decisión implementado alcanzó un 91.30% de exactitud en el conjunto de prueba. mostrando un desempeño sobresaliente.
- El dataset balanceado validó el uso de accuracy como métrica principal.
- No fue necesario aplicar estandarización ni codificación adicional debido a la naturaleza del algoritmo de aprendizaje automático y del dataset.
- La entropía y la ganancia de información guiaron las divisiones del árbol, mejorando la pureza de los nodos en cada iteración.
- Los criterios de parada fueron fundamentales para lograr un modelo con buen equilibrio entre sesgo y varianza, evitando así el subajuste y el sobreajuste.
- A futuro, sería conveniente experimentar con métodos más complejos de implementar como el ensamble mediante Random Forest o Bagging para reducir los errores residuales y mejorar aún más la capacidad de generalización.

7. Referencias

Elmenshawii, F. (2023). *Decision Tree From Scratch*. kaggle. https://www.kaggle.com/code/fareselmenshawii/decision-tree-from-scratch

Goel, S. (2024, May0 25). *Building Decision Tree from Scratch (Without Scikit-Learn Toolkit)*. Medium. https://medium.com/@wriath18/building-decision-tree-from-scratch-without-scikit-learn-toolkit-a96a14cfc6f1

Tripathi, M. (2022). *Machine Learning Algorithms from Scratch*. kaggle. https://www.kaggle.com/code/milan400/machine-learning-algorithms-from-scratch