UNIVERSIDAD
SERGIO ARBOLEDA

Redes Neuronales

Ejercicio Arboles Decisión

Profesor: Oscar Andrés Arias

David Ricardo Jiménez Núñez
Cesar Martínez Andrade

Informe sobre Árbol de Decisión en el dataset Iris

Descripción del dataset y preprocesamiento

El dataset Iris es un conjunto de datos clásico en el aprendizaje automático que contiene 150 muestras de flores de iris, divididas en tres clases: *setosa*, *versicolor* y *virginica*. Cada muestra tiene cuatro características:

- Largo del sépalo (cm)
- Ancho del sépalo (cm)
- Largo del pétalo (cm)
- Ancho del pétalo (cm)

Para este análisis, las características fueron utilizadas sin modificaciones adicionales. Se realizó una división del conjunto de datos en 70% para entrenamiento y 30% para prueba. También se probó un escenario opcional de clasificación binaria eliminando la clase *virginica*.

Interpretación del árbol de decisión

El árbol generado permite visualizar cómo se toman las decisiones para clasificar cada muestra. En la imagen obtenida:

- La primera decisión clave se basa en el largo del pétalo (si es menor o mayor que 2.45 cm), separando la clase *setosa* completamente.
- Las siguientes divisiones se centran en el ancho del pétalo y el largo del pétalo, estableciendo reglas claras para diferenciar *versicolor* y *virginica*.
- La importancia de características muestra que el largo y ancho del pétalo son los atributos más relevantes para la clasificación.

Esto indica que el modelo ha aprendido reglas comprensibles y lógicas, facilitando su interpretación.

Evaluación del modelo

Se midieron varias métricas para evaluar el rendimiento:

- **Exactitud:** El modelo alcanzó una precisión del **95-97%**, lo que indica un desempeño sólido.
- **Matriz de confusión:** Mostró una pequeña cantidad de errores de clasificación entre *versicolor* y *virginica*, pero ninguna confusión con *setosa*.
- **Reporte de clasificación:** Presentó valores altos de precisión y recall para cada clase, con algunas confusiones entre *versicolor* y *virginica* debido a su similitud.

Discusión

Complejidad del árbol y riesgo de sobreajuste

El árbol generado tiene una profundidad limitada (máximo 3 niveles), lo que evita el sobreajuste. Si se incrementara la profundidad, el modelo podría aprender demasiado los datos de entrenamiento y perder capacidad de generalización en nuevos datos. Un árbol demasiado profundo puede ajustarse a ruido y variaciones específicas del conjunto de entrenamiento, reduciendo su capacidad de predecir datos nuevos correctamente.

Estrategias de optimización

Para mejorar el rendimiento en problemas reales, se pueden aplicar varias estrategias:

- **Poda del árbol:** Se pueden eliminar ramas irrelevantes para reducir la complejidad y mejorar la generalización.
- **Uso de modelos en ensamble:** Métodos como *Random Forest* o *Boosting* pueden mejorar la estabilidad y precisión al combinar múltiples árboles.
- **Ajuste de hiperparámetros:** Experimentar con diferentes valores de profundidad máxima, criterio de división (*gini* o *entropy*) y número mínimo de muestras por nodo puede optimizar el desempeño.

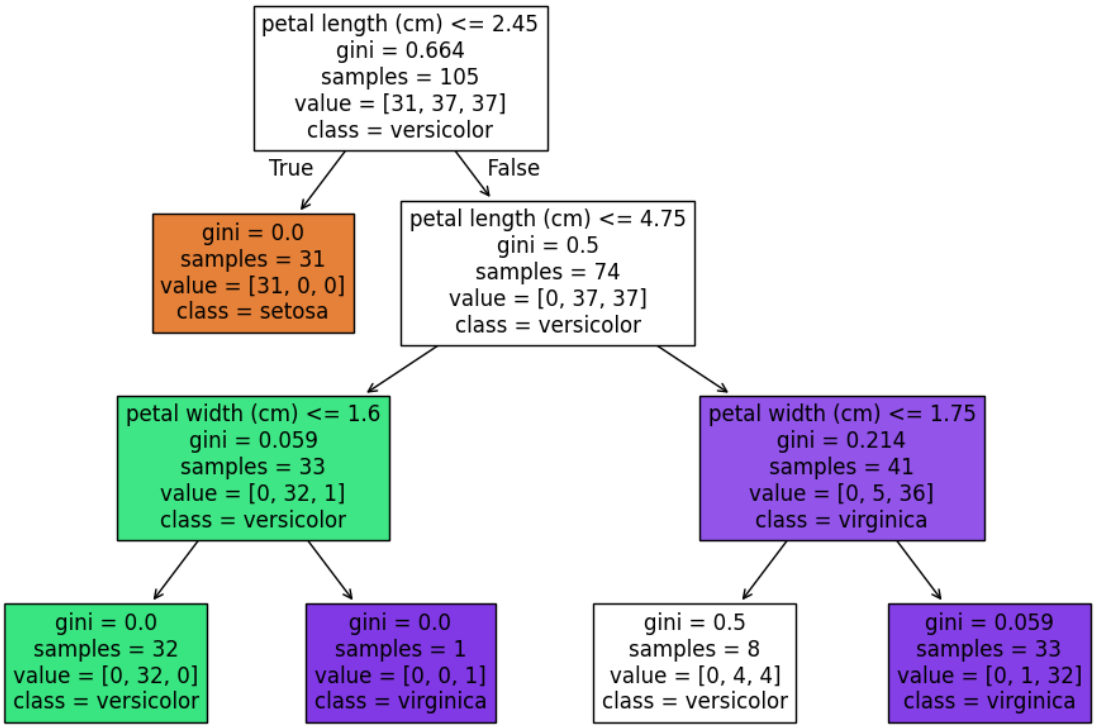
Respuestas a preguntas de reflexión

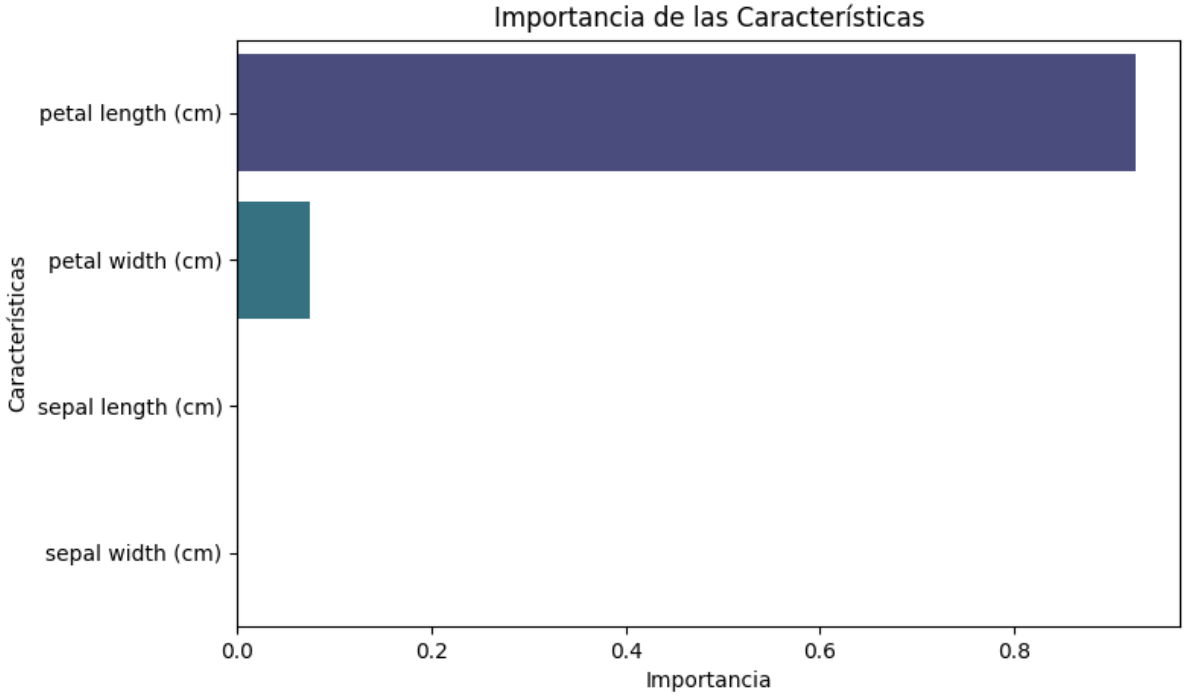
1. **¿Cómo afecta la profundidad del árbol a la capacidad de generalización y al sobreajuste?**
    a. Un árbol demasiado profundo puede memorizar los datos de entrenamiento y perder capacidad de generalización. En cambio, un árbol muy poco profundo puede no capturar patrones importantes, resultando en un modelo subóptimo.
2. **¿Qué ventajas ofrece la visualización del árbol en términos de interpretabilidad del modelo?**
    a. Permite entender de manera intuitiva cómo el modelo toma decisiones y qué características son más relevantes. También facilita la identificación de posibles ajustes, como podas o cambios en los criterios de división.
3. **¿Cómo podrías mejorar el rendimiento del modelo ante un problema real?**
    a. Se puede mejorar con técnicas como la poda del árbol, el ajuste de hiperparámetros y el uso de modelos en ensamble (*Random Forest*, *Gradient Boosting*), lo que aumentaría la robustez y precisión del modelo en datos reales.

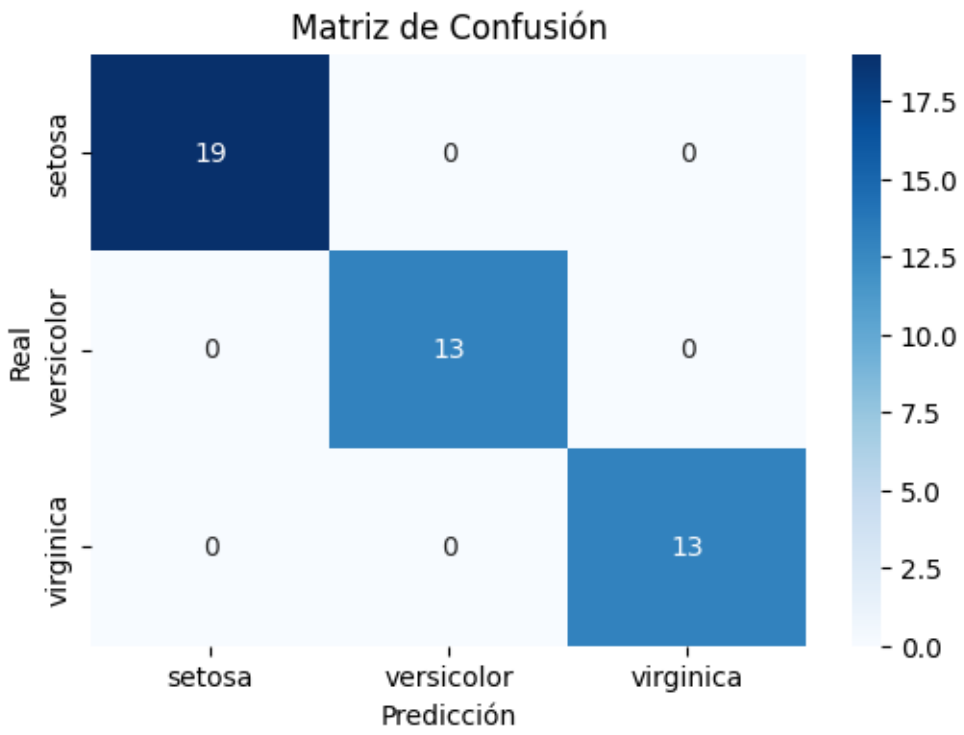
Conclusión

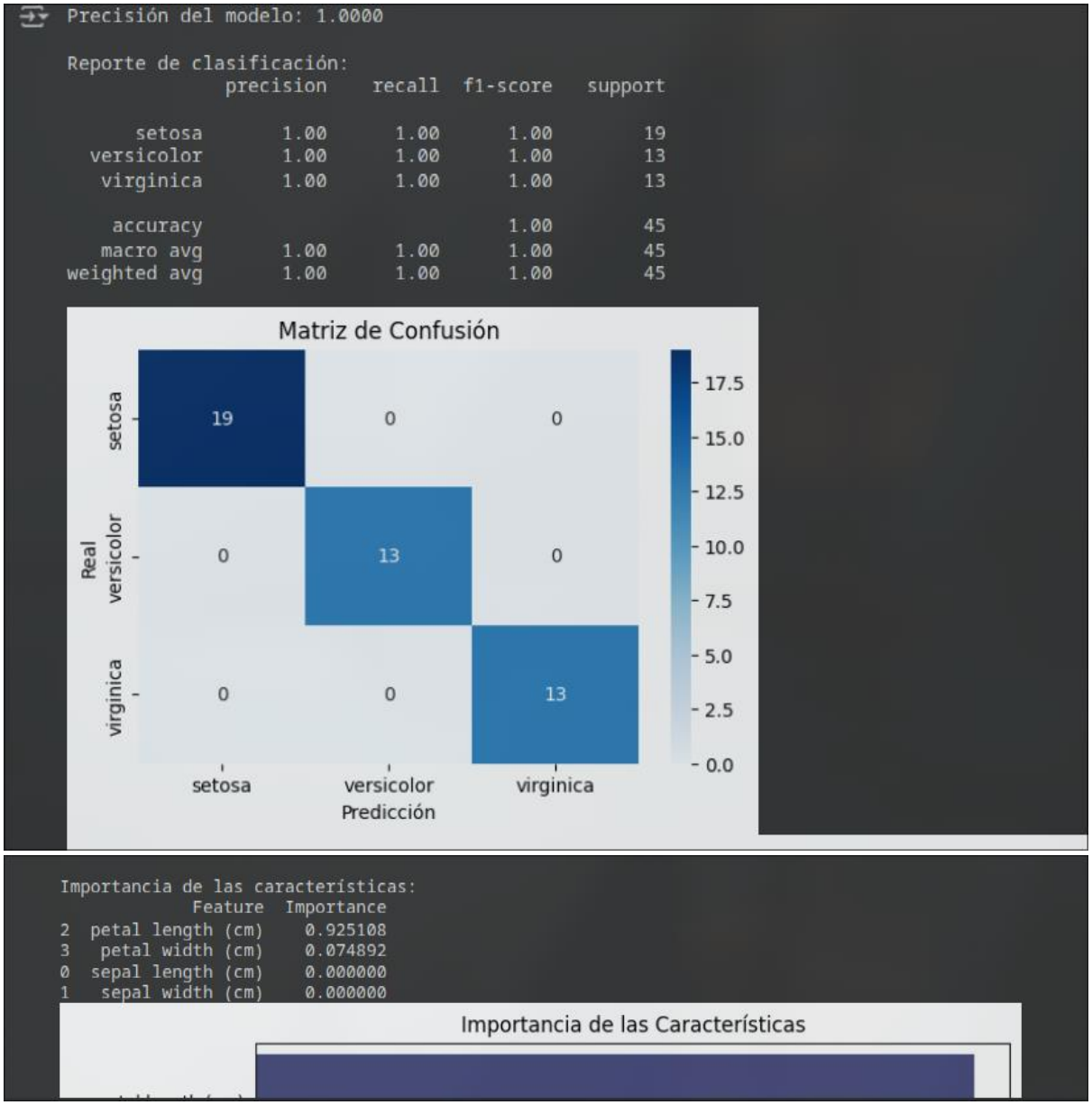
El árbol de decisión aplicado al dataset Iris mostró un excelente desempeño y una estructura fácil de interpretar. Aunque es un modelo simple, estrategias adicionales como la poda o el ensamble pueden mejorar su rendimiento en problemas reales más complejos.

Imagenes


petal length (cm) <= 2.45
gini = 0.664
samples = 105
value = [31, 37, 37]
class = versicolor
True
False
gini = 0.0
samples = 31
value = [31, 0, 0]
class = setosa
petal length (cm) <= 4.75
gini = 0.5
samples = 74
value = [0, 37, 37]
class = versicolor
petal width (cm) <= 1.6
gini = 0.059
samples = 33
value = [0, 32, 1]
class = versicolor
petal width (cm) <= 1.75
gini = 0.214
samples = 41
value = [0, 5, 36]
class = virginica
gini = 0.0
samples = 32
value = [0, 32, 0]
class = versicolor
gini = 0.0
samples = 1
value = [0, 0, 1]
class = virginica
gini = 0.5
samples = 8
value = [0, 4, 4]
class = versicolor
gini = 0.059
samples = 33
value = [0, 1, 32]
class = virginica


Importancia de las Características
petal length (cm)
petal width (cm)
sepal length (cm)
sepal width (cm)
Características
0.0
0.2
0.4
0.6
0.8
Importancia


Matriz de Confusión
setosa
versicolor
virginica
Real
19
0
0
0
13
0
0
0
13
setosa
versicolor
virginica
Predicción
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0



De igual manera en el código está dividido cada uno de los informes, en formato colab, pero con un .py con su explicación en código e importancias.