COLOMBIA POTENCIA DE LA VIDA

TIC

UNIVERSIDAD SERGIO ARBOLEDA

Análisis de Datos
Nivel Integrador

TALENTO TECH

UNIVERSIDAD SERGIO ARBOLEDA

PREPARACIÓN SEMANA 10

MINERÍA DE DATOS

La minería de datos es un proceso de descubrimiento de patrones significativos, tendencias y conocimientos ocultos en grandes conjuntos de datos. Utiliza una combinación de métodos estadísticos, algoritmos de aprendizaje automático y técnicas de visualización para explorar y analizar datos. El objetivo principal de la minería de datos es transformar datos brutos en información útil, proporcionando una comprensión más profunda de los fenómenos subyacentes.

Aspectos Clave:

1. **Extracción de Conocimiento:** La minería de datos busca extraer conocimiento implícito en los datos, permitiendo la toma de decisiones informada.
2. **Identificación de Patrones:** Busca patrones, relaciones o tendencias en los datos que pueden no ser evidentes a simple vista.
3. **Aplicación en Diversas Áreas:** Se aplica en áreas como negocios, ciencia, medicina, finanzas, entretenimiento, etc.
4. **Proceso Iterativo:** Implica un ciclo continuo de exploración, modelado y evaluación.

MINERÍA DE DATOS

Aplicaciones en el Mundo Real

La minería de datos tiene aplicaciones en una amplia variedad de campos, mejorando la eficiencia y proporcionando información valiosa. Algunas aplicaciones prácticas incluyen:

1. **Comercio y Negocios:**
    - Segmentación de clientes para campañas de marketing dirigidas.
    - Predicción de tendencias de ventas.
    - Análisis de patrones de compra.
2. **Ciencia y Investigación:**
    - Identificación de patrones en datos científicos.
    - Descubrimiento de relaciones en grandes conjuntos de datos.

MINERÍA DE DATOS

3. **Medicina y Salud:**
    - Diagnóstico y pronóstico de enfermedades.
    - Personalización de tratamientos médicos.
    - Análisis de datos genéticos.
4. **Finanzas:**
    - Detección de fraudes financieros.
    - Pronóstico de riesgos y movimientos del mercado.
5. **Producción y Manufactura:**
    - Control de calidad y optimización de procesos.
    - Mantenimiento predictivo de maquinaria.
6. **Educación:**
    - Seguimiento del rendimiento académico.
    - Personalización de métodos de enseñanza.

PROCESO DE MINERÍA DE DATOS

El proceso de minería de datos es una secuencia sistemática de pasos diseñados para descubrir patrones y conocimientos significativos a partir de grandes conjuntos de datos. A continuación, se detallan los principales pasos del proceso:

1. Exploración de Datos

Objetivo: Familiarizarse con los datos y comprender su estructura.

Actividades Clave:

- **Descripción de Datos:**
  - Obtener estadísticas descriptivas como media, mediana, desviación estándar.
  - Visualizar la distribución de variables mediante histogramas y gráficos.
- **Identificación de Patrones Preliminares:**
  - Explorar relaciones entre variables.
  - Detectar posibles outliers o valores atípicos.

PROCESO DE MINERÍA DE DATOS

2. Preprocesamiento

Objetivo: Mejorar la calidad de los datos y prepararlos para el modelado.

Actividades Clave:

- **Limpieza de Datos:**
  - Tratar valores faltantes o nulos.
  - Eliminar duplicados y registros irrelevantes.
- **Transformación de Datos:**
  - Normalizar variables para escalas consistentes.
  - Codificar variables categóricas.
- **Selección de Características:**
  - Identificar y mantener las variables más relevantes.

PROCESO DE MINERÍA DE DATOS

3. Modelado

Objetivo: Aplicar algoritmos de minería de datos para descubrir patrones.

Actividades Clave:

- **Selección de Algoritmos:**
  - Elegir algoritmos adecuados según los objetivos y la naturaleza de los datos.
- **Entrenamiento del Modelo:**
  - Utilizar datos de entrenamiento para ajustar el modelo.

PROCESO DE MINERÍA DE DATOS

4. Evaluación

Objetivo: Evaluar el rendimiento del modelo y su capacidad para generalizar.

Actividades Clave:

- **Validación Cruzada:**
  - Dividir los datos en conjuntos de entrenamiento y prueba.
- **Métricas de Evaluación:**
  - Calcular métricas como precisión, recall, F1-score para modelos de clasificación.
  - Utilizar métricas como el error cuadrático medio para modelos de regresión.
- **Ajuste del Modelo:**
  - Realizar ajustes según los resultados de la evaluación.

PROCESO DE MINERÍA DE DATOS

Ejemplo en Python

El código de los ejemplos vistos acá podremos encontrarlo en este enlace: https://acortar.link/SJ3Fjy

PROCESO DE MINERÍA DE DATOS

En Python, bibliotecas como scikit-learn proporcionan herramientas para cada etapa del proceso así:

```python
# 1. Exploración de Datos
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('datos.csv')
print(df.describe())
df['variable'].hist()
plt.show()

# 2. Preprocesamiento
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer

imputer = SimpleImputer(strategy='mean')
df['variable'].fillna(df['variable'].mean(), inplace=True)

scaler = StandardScaler()
df['variable_normalizada'] = scaler.fit_transform(df[['variable']])

# 3. Modelado
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

X_train, X_test, y_train, y_test = train_test_split(df[['feature1', 'feature2']], df['target'], test_size=0.2)
model = RandomForestClassifier()
model.fit(X_train, y_train)

# 4. Evaluación
from sklearn.metrics import accuracy_score

y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy}')
```

PROCESO DE MINERÍA DE DATOS

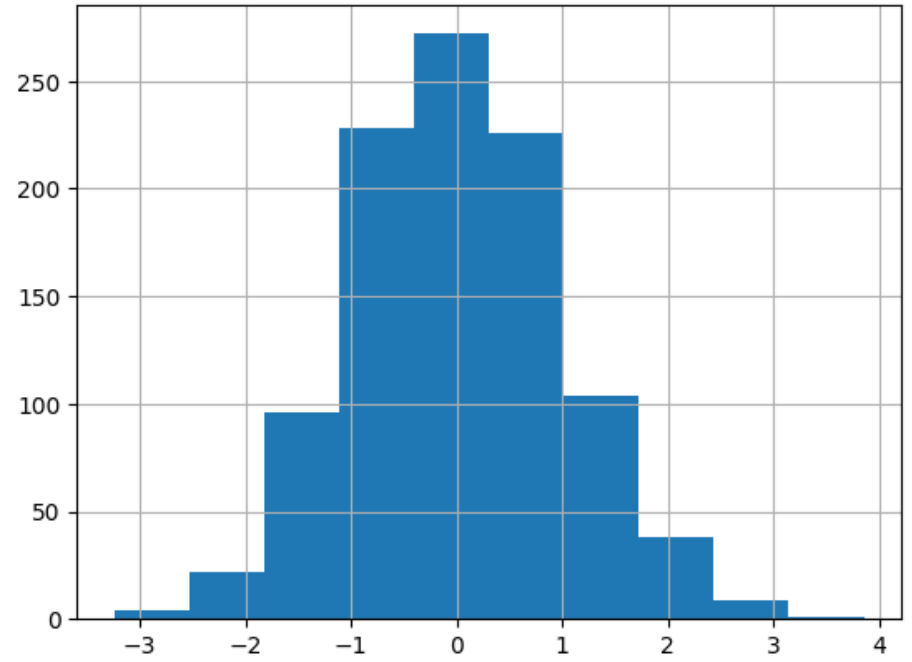
El cual con datos aleatorios(Encuentras el código en el enlace) Donde:

- Genero dos variables (**feature1** y **feature2**) y la variable objetivo (**target**) de manera aleatoria.
- Exploro los datos con **describe()** y visualizo el histograma de **feature1**.
- Escalo la variable **feature1** usando **StandardScaler**.
- Divido los datos en conjuntos de entrenamiento y prueba.
- Entreno un modelo de Bosques Aleatorios (**RandomForestClassifier**).
- Evalúo el modelo utilizando la métrica de precisión (**accuracy**).

PROCESO DE MINERÍA DE DATOS

Quedaría asi:

	feature1	feature2	target
count	1000.000000	1000.000000	1000.00000
mean	0.019332	0.070836	0.49000
std	0.979216	0.997454	0.50015
min	-3.241267	-2.940389	0.00000
25%	-0.647590	-0.606242	0.00000
50%	0.025301	0.063077	0.00000
75%	0.647944	0.728882	1.00000
max	3.852731	3.193108	1.00000



Accuracy: 0.495

TIPOS DE PROBLEMAS EN MINERÍA DE DATOS

En minería de datos nos encontramos con diferentes tipos de problemas según la naturaleza de los datos y los objetivos del análisis. Aquí te presento una descripción de cuatro tipos comunes de problemas en Minería de Datos:

1. **Clasificación:**
    - **Descripción:** En problemas de clasificación, el objetivo es asignar instancias o registros a una categoría o clase predefinida. Estas clases pueden ser binarias (dos clases) o multinarias (más de dos clases).
    - **Ejemplo Práctico:** Clasificación de correos electrónicos como "spam" o "no spam".
2. **Regresión:**
    - **Descripción:** Los problemas de regresión implican predecir un valor numérico continuo en lugar de asignar a una clase. Se busca encontrar la relación entre las variables de entrada y la variable de salida.
    - **Ejemplo Práctico:** Predicción de los precios de las viviendas basándose en características como el número de habitaciones, la ubicación, etc.

TIPOS DE PROBLEMAS EN MINERÍA DE DATOS

1. **Clustering:**
   - **Descripción:** En el clustering, el objetivo es agrupar instancias similares entre sí en conjuntos o clusters. Los miembros de un cluster comparten similitudes y difieren de los miembros de otros clusters.
   - **Ejemplo Práctico:** Agrupación de clientes en segmentos según su comportamiento de compra.
2. **Asociación:**
   - **Descripción:** Los problemas de asociación buscan identificar patrones o relaciones entre variables en grandes conjuntos de datos. Especialmente se utilizan para descubrir reglas de asociación entre diferentes elementos.
   - **Ejemplo Práctico:** Identificación de patrones de compra conjunta en un conjunto de datos de ventas minoristas.

ÁRBOLES DE DECISIÓN

Los arboles de decisión son modelos predictivos que mapean características (o variables) a conclusiones sobre el valor objetivo deseado. Estos modelos toman decisiones en forma de estructuras de árboles, donde cada nodo interno representa una prueba en una característica, cada rama representa el resultado de esa prueba, y cada hoja representa el resultado de la decisión, son atractivos por su interpretabilidad y capacidad para manejar conjuntos de datos complejos y no lineales. Sin embargo, también pueden ser propensos al sobreajuste, y su rendimiento puede mejorar mediante técnicas como la poda y el uso de conjuntos de árboles (como Bosques Aleatorios).

La estructura de un árbol de decisión consta de:

1. **Nodos:**
    - **Nodo Raíz:** El nodo superior que realiza la primera división basada en una característica.
    - **Nodos Internos:** Nodos que realizan divisiones adicionales en el árbol.
    - **Hojas (o Nodos Terminales):** Nodos que no se dividen más y representan las predicciones finales.

ÁRBOLES DE DECISIÓN

1. **Ramas:** Las ramas conectan los nodos y representan los resultados de las pruebas realizadas en los nodos.

Funcionamiento del Algoritmo:

El algoritmo de construcción de árboles de decisión sigue un enfoque de división recursiva. Aquí hay una visión general del proceso:

1. **Selección de Característica:** En cada paso, se selecciona la característica que mejor divide el conjunto de datos en función de algún criterio, como la ganancia de información o la impureza de Gini.
2. **División del Conjunto de Datos:** El conjunto de datos se divide en subconjuntos más pequeños en función de los valores de la característica seleccionada.
3. **Construcción Recursiva:** El proceso se repite recursivamente en cada subconjunto hasta que se alcanza un criterio de parada, como una profundidad máxima o un número mínimo de muestras en una hoja.
4. **Asignación de Etiquetas a Hojas:** Cada hoja se asigna con la etiqueta de la clase predominante en ese subconjunto.
5. **Predicción:** Para predecir, los datos nuevos se siguen por el árbol de decisión desde el nodo raíz hasta llegar a una hoja, donde se hace la predicción.

APRENDIZAJE SUPERVISADO

Árboles de Decisión para Clasificación:

En el contexto de clasificación, los árboles de decisión se utilizan para predecir la pertenencia a una categoría específica. Aquí hay una descripción general de cómo se aplican:

1. **Construcción del Árbol:**
    - Selecciona la característica que mejor divide el conjunto de datos en términos de la medida de impureza (como la entropía o la impureza de Gini).
    - Divide el conjunto de datos en ramas basadas en los valores de la característica seleccionada.
2. **Recorrido del Árbol:** Para hacer una predicción, los datos se recorren por el árbol desde la raíz hasta una hoja siguiendo las decisiones basadas en las características.
3. **Predicción:** La predicción se realiza asignando a la instancia de datos la etiqueta de la clase mayoritaria en la hoja correspondiente.
4. **Interpretación:** La estructura del árbol permite una interpretación sencilla de cómo se toman las decisiones y qué características son más relevantes.

APRENDIZAJE SUPERVISADO

Ejemplo: Se utiliza el conjunto de datos Iris y un árbol de decisión para clasificar las flores en tres clases (setosa, versicolor, virginica).

```python
# Importar librerías
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier, export_text

# Cargar el conjunto de datos Iris
iris = load_iris()
X = iris.data
y = iris.target

# Crear un clasificador de árbol de decisión
clf = DecisionTreeClassifier()

# Entrenar el clasificador
clf = clf.fit(X, y)

# Mostrar el árbol de decisión
tree_rules = export_text(clf, feature_names=iris.feature_names)
print(tree_rules)
```

APRENDIZAJE SUPERVISADO

El resultado muestra las reglas de decisión del árbol.

```
|--- petal length (cm) <= 2.45
|   |--- class: 0
|--- petal length (cm) >  2.45
|   |--- petal width (cm) <= 1.75
|   |   |--- petal length (cm) <= 4.95
|   |   |   |--- petal width (cm) <= 1.65
|   |   |   |   |--- class: 1
|   |   |   |--- petal width (cm) >  1.65
|   |   |   |   |--- class: 2
|   |   |--- petal length (cm) >  4.95
|   |   |   |--- petal width (cm) <= 1.55
|   |   |   |   |--- class: 2
|   |   |   |--- petal width (cm) >  1.55
|   |   |   |   |--- sepal length (cm) <= 6.95
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- sepal length (cm) >  6.95
|   |   |   |   |   |--- class: 2
|   |--- petal width (cm) >  1.75
|   |   |--- petal length (cm) <= 4.85
|   |   |   |--- sepal length (cm) <= 5.95
|   |   |   |   |--- class: 1
|   |   |   |--- sepal length (cm) >  5.95
|   |   |   |   |--- class: 2
|   |   |--- petal length (cm) >  4.85
|   |   |   |--- class: 2
```

APRENDIZAJE SUPERVISADO

Árboles de Decisión para Regresión:

Cuando se trata de regresión, los árboles de decisión predicen valores numéricos en lugar de clases. Aquí está el proceso:

1. **Construcción del Árbol:** Similar a la clasificación, se selecciona la característica que mejor divide el conjunto de datos en términos de alguna métrica de impureza, pero en este caso se busca minimizar la varianza.
2. **Recorrido del Árbol:** Los datos se recorren por el árbol, dividiendo el conjunto en ramas basadas en los valores de la característica seleccionada.
3. **Predicción:** La predicción se realiza asignando a la instancia de datos el valor promedio de la variable objetivo en la hoja correspondiente.
4. **Interpretación:** Al igual que en la clasificación, la interpretación es sencilla, y se puede entender cómo las características afectan a la predicción.

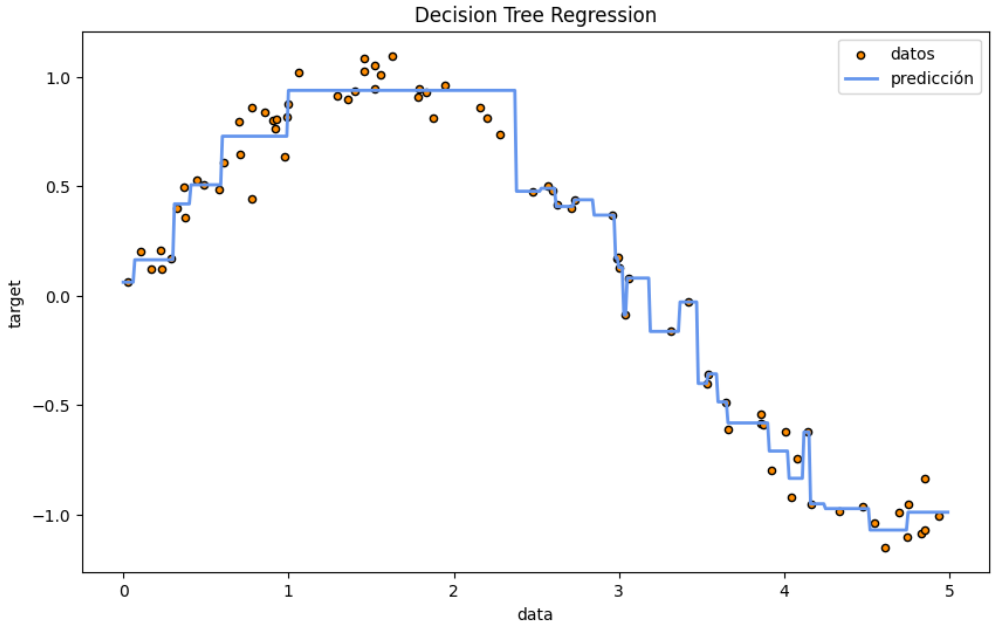
APRENDIZAJE SUPERVISADO

Ejemplo: Aquí se utiliza datos de regresión y un árbol de decisión para predecir una función seno con ruido añadido.

```
# Importar librerías
import numpy as np
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeRegressor

# Crear datos de ejemplo para regresión
np.random.seed(42)
X_reg = np.sort(5 * np.random.rand(80, 1), axis=0)
y_reg = np.sin(X_reg).ravel() + np.random.normal(0, 0.1, X_reg.shape[0])

# Crear un regresor de árbol de decisión
regressor = DecisionTreeRegressor(max_depth=5)

# Entrenar el regresor
regressor.fit(X_reg, y_reg)

# Predecir valores para nuevas instancias
X_new = np.arange(0.0, 5.0, 0.01)[:, np.newaxis]
y_pred = regressor.predict(X_new)

# Visualizar los resultados
plt.figure(figsize=(10, 6))
plt.scatter(X_reg, y_reg, s=20, edgecolor="black", c="darkorange", label="datos")
plt.plot(X_new, y_pred, color="cornflowerblue", label="predicción", linewidth=2)
plt.xlabel("data")
plt.ylabel("target")
plt.title("Decision Tree Regression")
plt.legend()
plt.show()
```

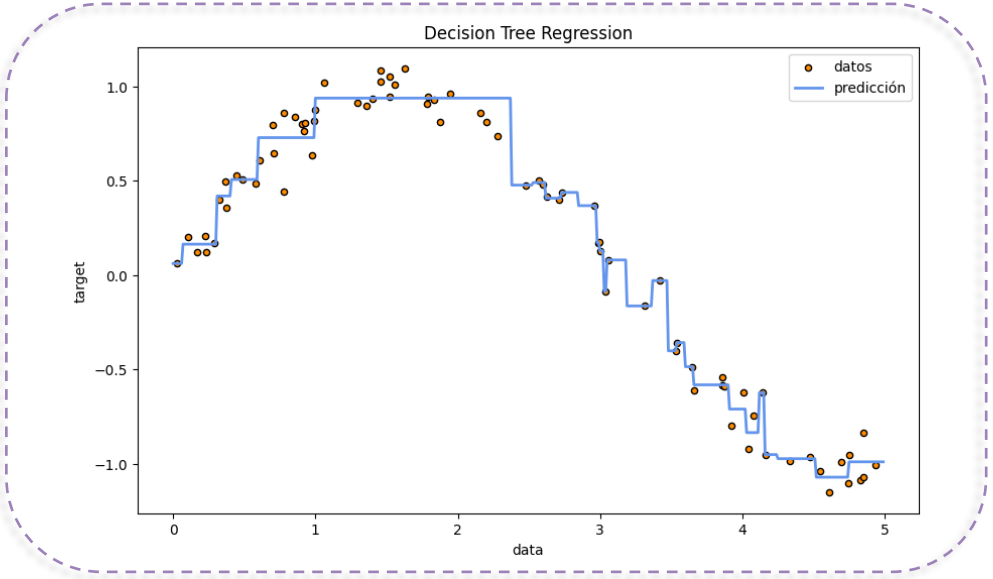
APRENDIZAJE SUPERVISADO

El resultado muestra cómo el árbol de decisión se ajusta a los datos de entrenamiento y realiza predicciones para nuevos valores.



APRENDIZAJE SUPERVISADO

El resultado muestra cómo el árbol de decisión se ajusta a los datos de entrenamiento y realiza predicciones para nuevos valores.



Los árboles de decisión para regresión y clasificación son versátiles y se utilizan en una variedad de aplicaciones. Sin embargo, es esencial considerar el sobreajuste y ajustar parámetros como la profundidad del árbol para obtener un rendimiento óptimo en datos nuevos.

CRITERIOS DE DIVISIÓN

Los criterios de división son fundamentales en la construcción de árboles de decisión. Cada uno mide la impureza de un nodo y se utiliza para determinar cómo se deben realizar las divisiones en el árbol.

1. **Gini (Índice de Gini):**
   - **Definición:** El índice de Gini mide la probabilidad de que un elemento elegido aleatoriamente sea clasificado incorrectamente cuando se le asigna una etiqueta de acuerdo con la distribución de las etiquetas en el nodo.
   - **Fórmula:** $Gini = 1 - \sum_{i=1}^{c} p_i^2$
   - Donde *pi* es la proporción de ejemplos de la clase *i* en el nodo.

CRITERIOS DE DIVISIÓN

2. **Entropía:**

   - **Definición:** La entropía mide la cantidad de incertidumbre o desorden en el nodo. Un nodo con entropía baja significa que sus ejemplos pertenecen principalmente a una clase.

   - **Fórmula:** $Entropía = -\sum_{i=1}^{c} p_i \log_2(p_i)$

   - Donde pi es la proporción de ejemplos de la clase i en el nodo.

CRITERIOS DE DIVISIÓN

3. **Error Cuadrático Medio (MSE - Mean Squared Error):**

   - **Definición:** MSE se utiliza en problemas de regresión y mide la media de los cuadrados de las diferencias entre las predicciones y los valores reales.

   - **Fórmula:** $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2$

   - Donde *n* es el número de ejemplos, *yi* es la predicción para el ejemplo *i*, y ¯*y*¯ es la media de todas las predicciones.

En la implementación de árboles de decisión en bibliotecas como **scikit-learn**, puedes elegir el criterio de división que mejor se adapte a tu problema. En general, no hay una regla estricta sobre cuál usar, y a menudo es recomendable probar varios criterios para ver cuál funciona mejor para tu conjunto de datos específico.

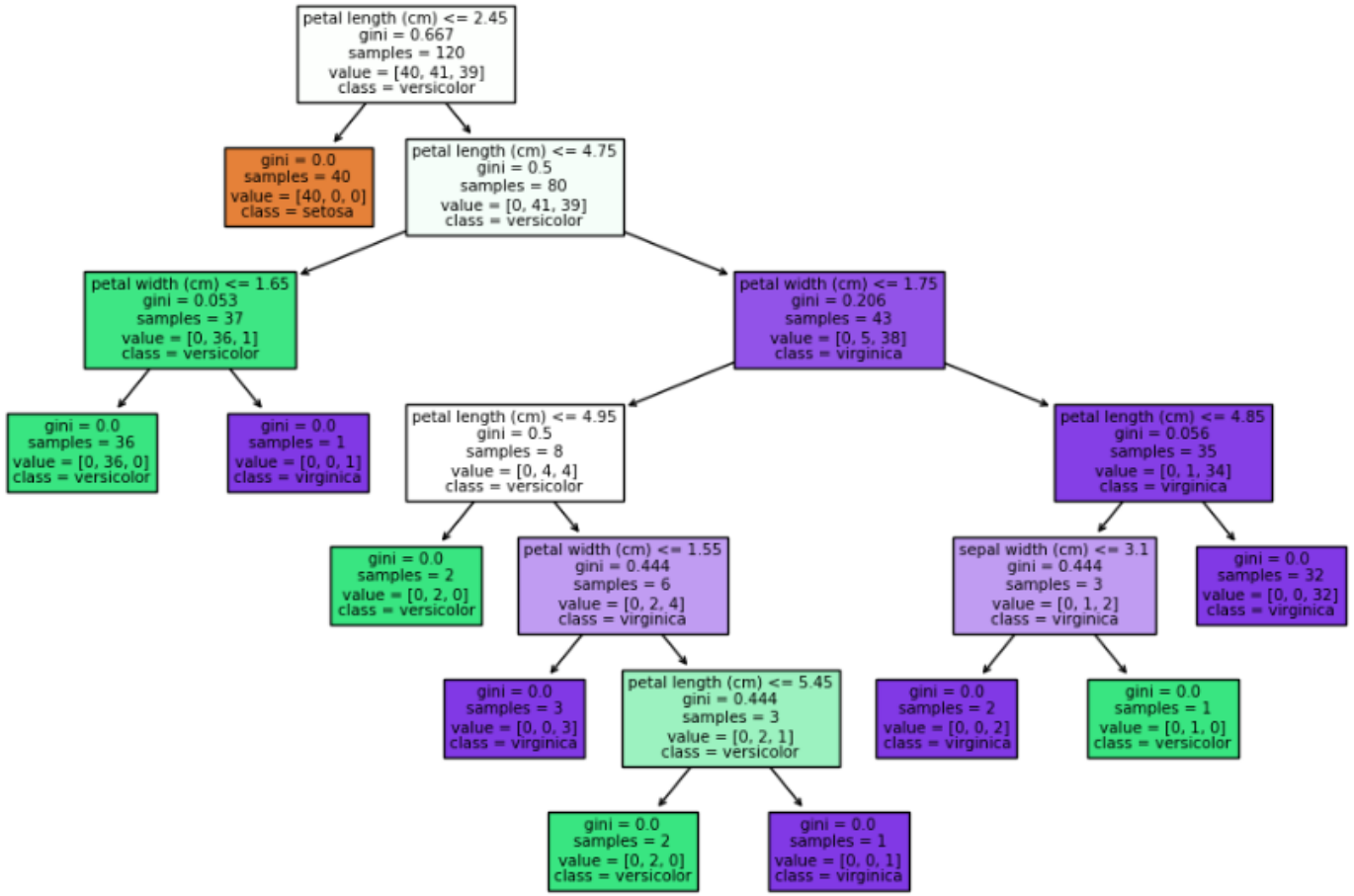
EJEMPLO PRÁCTICO

Aplicación de un Modelo de Árbol de Decisión en Python: Ejemplo y Evaluación de Rendimiento (Ver código en el enlace)

Vamos a utilizar scikit-learn para aplicar un modelo de árbol de decisión a un conjunto de datos de ejemplo y evaluar su rendimiento. En este ejemplo, asumiremos que trabajamos con un problema de clasificación.

```
Accuracy: 1.0
Matriz de Confusión:
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
Informe de Clasificación:
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        10
  versicolor       1.00      1.00      1.00         9
   virginica       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

EJEMPLO PRÁCTICO



¿Preguntas?