Práctica 0: Aprendizaje Automático: Clasificación con Scikit-learn

Objetivo:

Práctica: Introducción al Aprendizaje Automático

Objetivo: Esta práctica tiene como objetivo comprender los conceptos fundamentales de aprendizaje supervisado , así como el uso de conjuntos de datos, atributos, entrenamiento, validación y prueba. Usaremos la biblioteca `scikit-learn` para poner en práctica estos conceptos en Python.

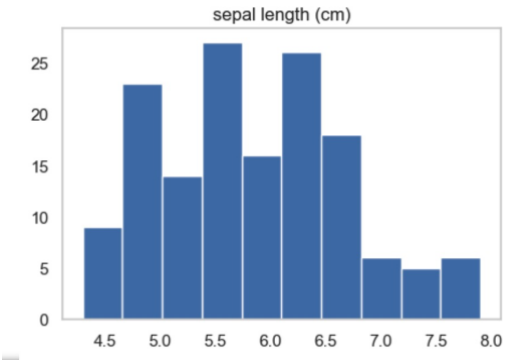
Apartado 1: Introducción y Preparación del Entorno

1. **Entender qué es el aprendizaje supervisado y no supervisado**:
    - **Aprendizaje supervisado**: Se utilizan datos etiquetados (es decir, conjuntos de datos que incluyen tanto entradas como salidas esperadas). El objetivo es que el modelo aprenda a predecir la salida correcta para nuevas entradas.
    - **Aprendizaje no supervisado**: Los datos no están etiquetados, y el objetivo es descubrir patrones o estructuras subyacentes en los datos sin guiarse por una salida esperada.
2. **Bibliotecas necesarias**:
    - `scikit-learn` para cargar datasets, preprocesar datos y entrenar modelos.
    - `numpy y pandas` para manipulación de datos.
    - `matplotlib` o `seaborn` para la visualización de resultados.
    - `DecisionTreeClassifier` para el árbol
    - `Kmeans` para los clusters

Apartado 2: Dataset, Atributos y Datos de Entrenamiento/Prueba

En esta sección, usaremos un dataset que viene integrado en `scikit-learn` fácil de cargar y contiene atributos y etiquetas listos para su uso.

    - Vamos a usar el dataset **Iris** que contiene 150 muestras de flores con 4 atributos (longitud y anchura del sépalo y pétalo) y 3 clases de salida (especies de Iris).
2. **Definiciones**:
    - **Atributos (features)**: Variables independientes que usamos para entrenar el modelo. En el dataset Iris, son las dimensiones del sépalo y el pétalo.
    - **Datos de Entrenamiento**: El subconjunto del dataset usado para entrenar el modelo.
    - **Datos de Validación**: Se usa para ajustar el modelo y evitar el sobreajuste (overfitting).
    - **Datos de Test**: Un subconjunto separado para evaluar el rendimiento del modelo.
3. **Carga del Dataset**: Se cargará el dataset Iris y se mostraran histogramas de cada atributo:



Explicar que muestra un histograma y el efecto que puede tener en una clasificación usando el sentido común.

Mostrar los valores de la clase y cuantas instancias hay de cada clase en el data set :

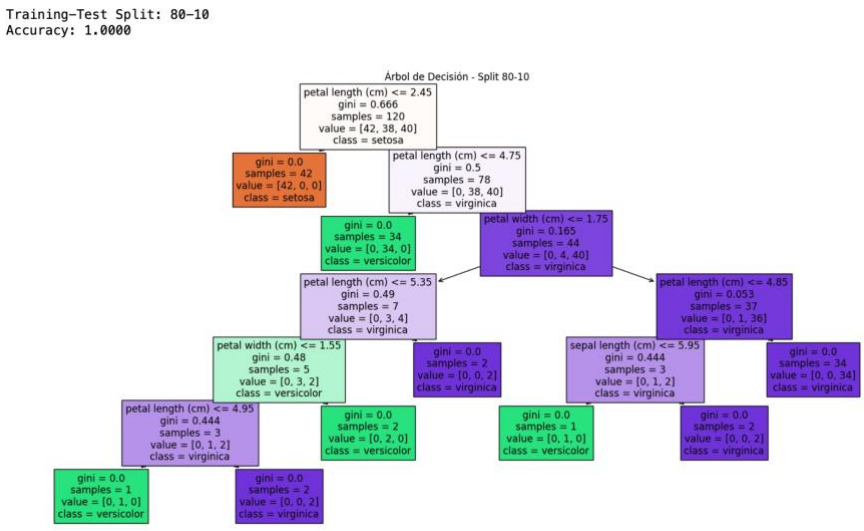
```
setosa          XX
versicolor      XX
virginica       XX
```

.

Apartado 3: Entrenamiento y Validación

En este apartado, vamos a entrenar un modelo de clasificación supervisado utilizando un **árbol de decisión** y validaremos su rendimiento con el accuracy (grado de acierto total).

1. **División de los datos**:
   - Usaremos la función `train_test_split` de `sklearn` para dividir el dataset en entrenamiento (70%) y test (30%).
2. **Entrenamiento del modelo**:
   - Entrenaremos un **Árbol de Decisión** utilizando los datos de entrenamiento. Mediante la función
3. Mostraremos la salida en accuracy de cada árbol y el árbol producido (plot_tree) , así como cuantas instancias hay de cada clase en el data set de entrenamiento :



¿Qué se puede deducir de los resultados? Y de los arboles generados?
¿Salen los mismos resultados si se ejecuta varias veces el script? ¿ porqué?