VNIVERSITAT DE VALÈNCIA

Laboratorio de ESTRUCTURAS DE DATOS Y ALGORITMOS

Grado en Ciencia de Datos (1º)

Curso 2024-25

Práctica Nº 5: Árboles binarios

Periodo de realización: Del **07 al 11/04/2025**

Metodología de trabajo

Fases	Tipo	Dedicación
1. Resolver las tareas planteadas en este guion antes de iniciar la sesión presencial en el laboratorio.	No presencial	Máx. 4,5 h.
2. Resolver en el laboratorio un nuevo ejercicio basado en la resolución de las tareas previas.	Presencial	Máx. 3 h.

Introducción

Los árboles de decisión son un modelo de clasificación que permite especificar una serie de condiciones que ocurren de forma sucesiva y que permiten resolver un problema (tomar una decisión) a partir de las características de los datos de entrada. Por ejemplo, un árbol de decisión puede especificar el esquema de clasificación para construir un sistema automático de diagnóstico médico, que permita determinar a partir de los síntomas observados o medidos en una persona la enfermedad que padece.

Los nodos internos de un árbol de decisión siempre evalúan características de la entrada y sus diferentes subárboles especifican los distintos resultados que se pueden obtener de esa evaluación y conducen hacia la evaluación de otras características o hacia la solución del problema. Por ejemplo, el árbol de decisión de la figura 1 especifica la forma de asignar una calificación cualitativa a un estudiante en función de la nota obtenida en un examen:

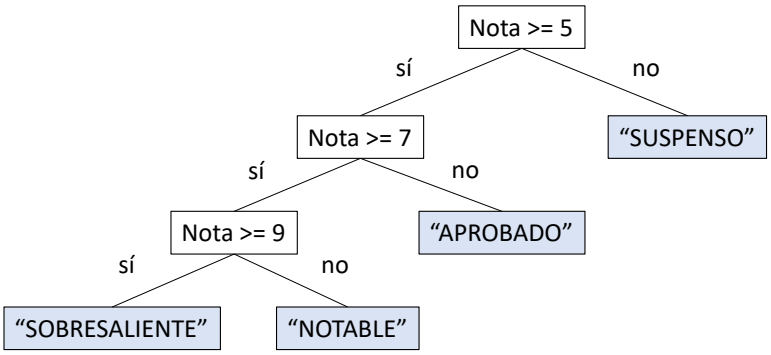

Figura 1. Ejemplo de árbol de decisión para determinar la calificación de un estudiante.

Cada nodo interno especifica una evaluación de las características de la nota. Si el resultado de la evaluación es afirmativo (sí), entonces se debe seguir evaluando la rama izquierda del árbol, si, por el contrario, la evaluación es negativa (no), se debe continuar la evaluación por la rama derecha. De esta manera, se determina la calificación cuando se llega a una hoja del árbol (solución del problema).

Si la evaluación de características especificada en cada nodo interno solo admite dos posibles valores, como en el caso del ejemplo de la figura 1, entonces el árbol de decisión será binario. Este va a ser el tipo de árbol de decisión planteado en esta práctica. Sin embargo, en general, los árboles de decisión no siempre son binarios y en muchos tienen grado superior a 2.

Objetivo de la práctica

En esta práctica se trata de realizar un programa que implemente el conocido juego de mesa *"Quién es quién"*[1], en el que participan 2 jugadores con un tablero de 24 cartas con caras de personajes y en el que tienen que adivinar un personaje seleccionado por el oponente. Para realizar el proceso de deducción, cada jugador debe, por turno, realizar una pregunta al oponente sobre las características físicas del personaje seleccionado. Estas preguntas solo pueden admitir como respuesta "Sí" o "No" (posee o no posee la característica indicada). Cada pregunta y la consecuente respuesta permite al jugador descartar un grupo de personajes. La partida finaliza cuando un jugador adivina el personaje del oponente.

Figura 2. Ejemplo de tableros para jugar a "¿Quién es quién?"

El programa de esta práctica debe simular el razonamiento seguido por un jugador (en este caso, el ordenador) para adivinar el personaje seleccionado por el otro jugador (el usuario del programa). De manera que, el usuario elegirá un personaje de entre los 24 posibles y el programa irá realizando preguntas hasta deducir el personaje seleccionado por el usuario.

Para realizar este programa se necesita la siguiente información:

- El conjunto de 24 personajes y sus características. Esta información se proporciona en el archivo "`personajes.pdf`", donde se incluye una tabla con los nombres de los personajes y las características físicas de cada uno de ellos. No se muestran imágenes para evitar posibles ambigüedades o dudas al responder a las preguntas.
- Un árbol de decisión que determine cómo se clasifican los personajes en función de sus características. Este es el modelo de razonamiento (clasificación) que seguirá el programa para resolver el problema. La información de este árbol está contenida en el archivo "`arbol_personajes.dat`".

[1] También se conoce por su nombre original en inglés, "*Guess who".*

Ejercicios (Fase 1)

Tarea 0: Clase Árbol Binario

Como material para realizar esta práctica se proporciona la implementación de la clase Árbol Binario (`ArbolBinario`) en el archivo "`clase_arbol.py`", que representa el concepto general de árbol binario. La implementación está completa y no debe ser modificada.

Como esta clase va a ser utilizada durante la práctica, debes conocer la interpretación de las operaciones y la forma de uso. Lee detenidamente la documentación proporcionada y el propio código para interpretar correctamente los elementos de la clase.

Tarea 1: Generar el árbol de decisión

Se debe escribir el programa "`pr5_fase1.py`" y su primera tarea deberá ser generar en memoria la estructura del árbol de decisión que va a permitir realizar el proceso de deducción de los personajes. El árbol de decisión ya está diseñado y la información que deben contener sus nodos y la estructura que debe tener está contenida en el archivo "`arbol_personajes.da`". Este archivo almacena secuencialmente la información del árbol, de acuerdo con los siguientes criterios:

- Cada línea contiene la información de un nodo y siempre es un texto (`string`).
- La secuencia de líneas ha sido generada realizando un recorrido *prefijo* del árbol original (diseñado en papel) y trasladando al archivo el contenido de cada nodo por el que pasa el recorrido. Los subárboles vacíos se indican explícitamente en el archivo con el símbolo "`.`". Como ejemplo del formato se muestra la secuencia correspondiente al árbol de la figura 1:

```
Nota >= 5
SUSPENSO
.
.
Nota >= 7
APROBADO
.
.
Nota >= 9
NOTABLE
.
.
SOBRESALIENTE
.
.
```

Para generar el árbol de decisión se debe definir una función que lea e interprete, de acuerdo con el formato descrito, una línea de un archivo de datos `f`, que debe estar previamente abierto para lectura[2], y devolverá un árbol binario con el árbol de decisión generado. Esta función tendrá una estructura recursiva y se ha visto en clase de teoría. En cada llamada a la función solo se lee una línea y se construye el árbol que corresponde a la información que contiene.

Para verificar el correcto funcionamiento de la generación del árbol puedes hacer que el programa imprima por pantalla, en orden prefijo, el contenido del árbol al finalizar su generación. Se deberá mostrar en pantalla la misma secuencia de líneas contenida en el archivo de entrada (excepto los '.' que indican árboles vacíos en la entrada). Si no es así, la generación ha sido incorrecta.

[2] Obviamente, fuera de esta función.

Tarea 2: Deducir del personaje

Como ya se dispone del árbol de decisión, ahora el programa está en condiciones de "adivinar" un personaje seleccionado por el usuario. Para ello, se debe definir una función que recorra el árbol de decisión desde su raíz, si el árbol por el que se pasa no corresponde a una hoja, debe realizar la pregunta contenida en su raíz y si la respuesta del usuario a la pregunta es afirmativa se deberá desplazar al subárbol izquierdo, pero si es negativa se deberá desplazar al subárbol derecho. Cuando se alcance una hoja se deberá mostrar la respuesta al juego, que será el nombre del personaje contenido en la raíz de esa hoja.

Verifica el correcto funcionamiento del programa jugando con él. Selecciona un personaje del archivo de personajes y responde a cada una de las preguntas realizadas por el programa de acuerdo con las características indicadas en el archivo para el personaje seleccionado. Si la respuesta del programa no es la esperada la función realizada en esta tarea será incorrecta y deberás corregir el error.

Tarea 3: Analizar el árbol

Se dispone de un árbol de decisión bien construido, con el que se ha comprobado que se puede "razonar". En esta última tarea se debe analizar de manera sencilla el árbol. En primer lugar, se debe incluir una función que calcule el número de nodos terminales del árbol de decisión. Este valor será el del número de personajes que puede "adivinar" el programa. En este caso concreto, el valor a obtener será 24.

Con objeto de poder visualizar en pantalla el árbol generado, se debe incluir en el programa una función que muestre el contenido del árbol binario usando el criterio prefijo y mostrando los datos con una tabulación proporcional al nivel de los nodos que lo componen y si se trata de una rama afirmativa (izquierda) o negativa (derecha). Como la forma de imprimir cada nodo depende del nivel en el que se encuentra, es probable que necesites pasar esta información como argumento de la función que realice esta tarea. La salida proporcionada debe reproducir la indicada en la página siguiente: *(…)*

Las líneas verticales se han incluido para hacer más fácil la lectura, pero no se deben imprimir en el ejercicio.

```
Tiene el pelo corto?
|  [SÍ]Es hombre?
|  |  [SÍ]Tiene pelo rojo?
|  |  |  [SÍ]Tiene barba?
|  |  |  |  [SÍ]Miguel
|  |  |  |  [NO]Sonrie?
|  |  |  |  |  [SÍ]Luis
|  |  |  |  |  [NO]Pablo
|  |  |  [NO]Tiene pelo oscuro?
|  |  |  |  [SÍ]Tiene ojos marrones?
|  |  |  |  |  [SÍ]Samuel
|  |  |  |  |  [NO]David
|  |  |  |  [NO]Tiene bigote?
|  |  |  |  |  [SÍ]Marc
|  |  |  |  |  [NO]Victor
|  |  [NO]Tiene ojos marrones?
|  |  |  [SÍ]Tiene pelo rojo?
|  |  |  |  [SÍ]Laura
|  |  |  |  [NO]Maria
|  |  |  [NO]Tiene pelo rojo?
|  |  |  |  [SÍ]Carla
|  |  |  |  [NO]Tiene pelo oscuro?
|  |  |  |  |  [SÍ]Katherine
|  |  |  |  |  [NO]Olga
|  [NO]Tiene ojos marrones?
|  |  [SÍ]Tiene pelo oscuro?
|  |  |  [SÍ]Esta calvo?
|  |  |  |  [SÍ]Andres
|  |  |  |  [NO]Guillermo
|  |  |  [NO]Es hombre?
|  |  |  |  [SÍ]Tiene piel blanca?
|  |  |  |  |  [SÍ]Tomas
|  |  |  |  |  [NO]Jaime
|  |  |  |  [NO]Ana
|  |  [NO]Es hombre?
|  |  |  [SÍ]Tiene el pelo rojo?
|  |  |  |  [SÍ]Tiene bigote?
|  |  |  |  |  [SÍ]Pedro
|  |  |  |  |  [NO]Jorge
|  |  |  |  [NO]Esta calvo?
|  |  |  |  |  [SÍ]Ivan
|  |  |  |  |  [NO]Daniel
|  |  |  [NO]Tiene ojos verdes?
|  |  |  |  [SÍ]Aitana
|  |  |  |  [NO]Tiene pelo gris?
|  |  |  |  |  [SÍ]Laila
|  |  |  |  |  [NO]Monica
```

Fase 2: Tarea final

Resolución de un nuevo ejercicio planteado durante la sesión presencial en el laboratorio.