

# **Practica 3**

## **Inteligencia Artificial**

José Miguel Avellana López 18068091G

Yassine El Kihal X8592934L

## **T9 - Construcción del árbol de forma recursiva**

El constructor recursivo lo que hace es recorrer todas las columnas (menos el:-1 porque es el de la columna final y de cada una recorrer todos sus valores con cada valor, con el método “divideset” calculamos el descenso de impureza o desorden que produciría.

Después de recorrer todos los valores del dataset nos quedamos con el descenso, la columna, el valor y los dos datasets de la mejor pregunta que podemos hacer desde ese nodo.

Si el mayor descenso no supera la variable determinada beta habremos llegado a un nodo hoja, y retornaremos un `decissionnode` que posea tan solo el diccionario de nuestro dataset.

De lo contrario, retornaremos un “`decissionnode`” con la columna, el valor y los dos “`datasets`” con el mejor descenso.

## **T10 - Construcción del árbol de forma iterativa.**

En esta versión del método se da uso de las clases “`Stack`” y “`Node`”, con los nodos frontera y su respectivo “`dataSet`”. Cuando toca crear los nodos true y false, los inserta en la lista de nodos frontera y, además, los coloca en el nodo actual como hijos, (en Python las dos variables apuntan a la misma dirección de memoria, por lo que si se modificará el árbol).

## **T12 - Función de clasificación.**

El método de clasificación consiste en determinar a qué hoja de un árbol ya construido (entrenado) llega un nuevo elemento. El método se dedica a, de forma recursiva, recorrer el árbol de la siguiente forma:

Desde el nodo raíz, aplica su pregunta (columna y valor) al elemento, y de cumplirse o no se dirige a una rama u otra. Desde este nuevo nodo nos preguntamos si es hoja, si no lo es volvemos a aplicar la pregunta de este nuevo nodo. Si se trata de un nodo hoja, retorna su diccionario “*result*”.

## **T13/14 - Evaluación del árbol.**

Con el fin de evaluar la calidad del algoritmo introducimos como parámetros dos data sets, uno que sirva para construir el árbol y otro para probarlo.

Construimos el árbol con trainingset. Después recorreremos cada elemento del testset y, con cada uno, daremos uso del método classify con el cual obtendremos si respectivo diccionario y, si el valor de clase (elem[:-1]) se encuentra en el diccionario (según el valor de beta durante la construcción, el diccionario será homogéneo o no) aumentaremos el contador de correctas en uno ya que significa que el elemento fue bien asignado.

Una vez recorrido todo el testset, retornamos el porcentaje de elementos correctos en función de la longitud total del testset.

## **T15 - Missing data.**

Otra manera de asignarle un valor a un campo vacío, aparte de introducir el valor que más se repite de ese atributo, puede ser, en caso de un valor numérico, calcular la media de ese atributo.

O bien aplicar un árbol de decisiones en función de todos los atributos del dataset menos del que nos falta el dato, y este usarlo como clase (atributo que queremos adivinar). Y entrenar el árbol con todos los elementos de nuestro dataset menos del elemento donde nos falta el dato (suponiendo que es el único valor que nos falta).

## **16 - Poda del árbol.**

La idea principal de esta función es dejar crecer el árbol hasta que no llegar a las hojas, es decir, a una hoja. Además, también juntamos los nodos hijos con el padre en caso de que la reducción de la entropía sea poca. También vemos que medimos la impureza de las dos hojas juntadas y las separamos en caso de que no supere threshold.

## Mejora de K-Clusters

Bestmatches es una lista de listas que contiene los ítems asignados al clúster correspondiente con la variable “j” recorreremos los clústeres, y con la variable “i” recorreremos cada uno de los ítems. De esta manera obtenemos los ítems y sabemos el clúster al que pertenecen.

Una vez sabemos la asignación de los ítems, calculamos la suma de las distancias al cuadrado de todos los ítems a sus respectivos centroides. Mediante una función auxiliar llamada “mínim\_total”, de la cual minimizamos la suma de las distancias.

A la función le pasaremos como parámetros el valor de “k” (número de clusters) y el número de veces que queremos llamar a “kcluster”. Lo que hará este método será devolver la mejor configuración.