Practica 3
Inteligencia Artificial

José Miguel Avellana López 18068091G

Jassine El Kihal X8592934L

T9 - Construcción del árbol de forma recursiva

El constructor recursivo lo que hace es recorrer todos los elementos de *part*, y de cada elemento recorrer todos sus valores (menos el :-1 porque es el de la columna final), con cada valor y el método divideset calculamos el descenso de impureza o desorden que produciria. Despues 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 version del metodo se da uso de una lista de tuplas, 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, ademas, 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 que 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 raiz, 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. Despues 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 (segun el valor de beta durante la construccion, el diccionario será homogeneo 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 funcion de la longitud total del testset.

T15 - Missing data.

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

O bien aplicar un arbol de decisiones en funcion 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, esdecir, 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

Para la mejora sobre esta función, declaramos una serie de variables que intentan guardar la posición que hace menos iteraciones, es decir, primero se empieza poniendo la posición en una posición aleatoria y al final se guardará la posición que hace menos iteraciones . La finalidad de este proceso es guardar la mejor posición inicial por el que se repite varias veces el proceso. Para hacer el reset de clústeres, hicimos que cada 3 iteraciones se vaya cambiando los clústeres que movieron en la última iteración.