



# ALGORITMO ID3

Ingeniería del Conocimiento

Guion de la asignatura de Ingeniería del Conocimiento de cuarto de carrera del grado de Ingeniería del Software de la Universidad Complutense de Madrid

Iker Burgoa  
4ºE INGENIERÍA DEL SOFTWARE

## INDICE

INTRODUCCIÓN .....	2
DETALLES DEL FUNCIONAMIENTO .....	4
DETALLES DE LA IMPLEMENTACIÓN.....	4
MANUAL DE USUARIO .....	6
CASOS DE PRUEBA.....	6
BIBLIOGRAFÍA.....	6



UNIVERSIDAD  
COMPLUTENSE  
MADRID

# INTRODUCCIÓN

El algoritmo ID3 fue creado por J.R.Quinlan, emplea un procedimiento de “arriba abajo” haciendo un recorrido voraz por el espacio de las posibles ramificaciones sin back tracking. Para ello, el algoritmo usa conceptos como la de “entropía” y “ganancias”.

El algoritmo ID3 es un árbol de decisión que construye un modelo de regresión o de clasificación en forma de estructura de árbol. Divide el conjunto de datos en conjuntos cada vez más pequeños mientras van construyendo el árbol de decisión asociado de una forma recursiva.

El árbol final está formado por nodos de decisión y por nodos hojas. Cada nodo decisión tiene dos o más ramas que son las posibilidades de esa decisión (el atributo), y los nodos hojas son la clasificación correspondiente de la propiedad que se quiere clasificar, representado por el nodo raíz del árbol.

Ejemplo:

Paciente	Presión Aterial	Urea en sangre	Gota	Hipotiroidismo	Administrar Tratamiento
1	Alta	Alta	Sí	No	No
2	Alta	Alta	Sí	Sí	No
3	Normal	Alta	Sí	No	Sí
4	Baja	Normal	Sí	No	Sí
5	Baja	Baja	No	No	Sí

En cada iteración se debe calcular el mérito para cada atributo, cuando tengamos el mérito de todos, debemos elegir el que menos valor tenga. Será el nodo actual del árbol. Para ello primero debo calcular la **ENTROPIA**:

**Fórmula Entropía:**

$$infor(p,n) = p \log_2(p) + n \log_2(n)$$

Donde p es el tanto por ciento de casos con una solución posible:

$$p : \%ejemplos + = \frac{|E+|}{|E+|+|E-|}$$

Y n es el tanto por ciento de casos con una solución posible:

$$n : \%ejemplos - = \frac{|E-|}{|E+|+|E-|}$$

```
soleado
Repeticiones: 5
Positivos: 2
Negativos: 3
Valor de R :0.35714285714285715
lluvioso
Repeticiones: 5
Positivos: 3
Negativos: 2
Valor de R :0.35714285714285715
nublado
Repeticiones: 4
Positivos: 4
Negativos: 0
Valor de R :0.2857142857142857
Merito de la variable principal TiempoExterior: 0.6935361388961918
```

Ejemplo donde desgloso cada variable y saco sus repeticiones, positivos, negativos y el valor de cada r.

Cuando ya tenemos los valores de la entropía, necesito calcular ahora el mérito para cada atributo:

**Mérito (Am) =  $\sum_{i=1}^n r_i \times \text{infor}(p_i, n_i)$ , donde  $p_i$  y  $n_i$  son el número de ejemplos positivos y negativos en el ejemplo actual.**

**El valor  $r_i$  es el resultado de hacer  $(a_i / N)$  donde  $a_i$  es el número de veces que se repite el atributo en todos los casos y  $N$  es el número total de ejemplos.**

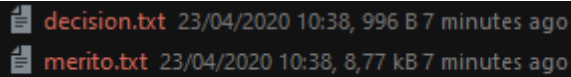
```
Merito de soleado : 0.3467680694480959
Merito de lluvioso : 0.3467680694480959
Merito de nublado : 0.0
Merito de la variable principal TiempoExterior: 0.6935361388961918
Merito de caluroso : 0.2857142857142857
Merito de frio : 0.23179374984546652
Merito de templado : 0.39355535745192405
Merito de la variable principal Temperatura: 0.9110633930116763
Merito de normal : 0.29583638929116374
Merito de alta : 0.4926140680171258
Merito de la variable principal Humedad: 0.7884504573082896
Merito de verdad : 0.42857142857142855
Merito de falso : 0.46358749969093305
Merito de la variable principal Viento: 0.8921589282623617
La cabeza del arbol es: TiempoExterior con un merito de : 0.6935361388961918
```

Este es un ejemplo de mi aplicación , donde muestro el mérito total de cada atributo y a continuación el mérito total de la variable principal.

Al final se puede observar que, tras obtener los méritos de todas las variables principales, se saca el menor mérito de los atributos.

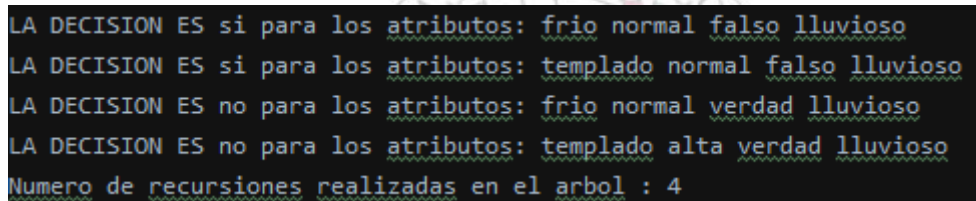
## DETALLES DEL FUNCIONAMIENTO

El programa al ejecutarse va a generar dos .txt, dentro del src:



decision.txt 23/04/2020 10:38, 996 B 7 minutes ago  
merito.txt 23/04/2020 10:38, 8,77 kB 7 minutes ago

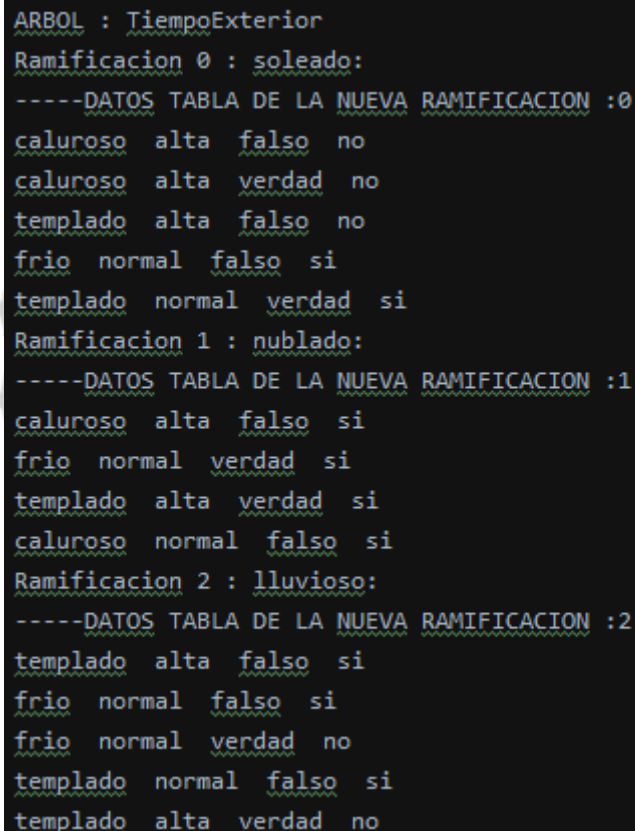
El fichero **decisión.txt** obtiene el resultado final de todas las variables con sus decisiones desglosadas y el número de recursiones hechas:



```
LA DECISION ES si para los atributos: frio normal falso lluvioso
LA DECISION ES si para los atributos: templado normal falso lluvioso
LA DECISION ES no para los atributos: frio normal verdad lluvioso
LA DECISION ES no para los atributos: templado alta verdad lluvioso
Numero de recursiones realizadas en el arbol : 4
```

El fichero **merito.txt** engloba todo el proceso hecho:

- 1) Méritos de las variables:  
(Segunda imagen)
- 2) El árbol desglosado con las nuevas variables de cada nueva rama con los datos nuevos para cada una:



```
ARBOL : TiempoExterior
Ramificacion 0 : soleado:
-----DATOS TABLA DE LA NUEVA RAMIFICACION :0
caluroso alta falso no
caluroso alta verdad no
templado alta falso no
frio normal falso si
templado normal verdad si
Ramificacion 1 : nublado:
-----DATOS TABLA DE LA NUEVA RAMIFICACION :1
caluroso alta falso si
frio normal verdad si
templado alta verdad si
caluroso normal falso si
Ramificacion 2 : lluvioso:
-----DATOS TABLA DE LA NUEVA RAMIFICACION :2
templado alta falso si
frio normal falso si
frio normal verdad no
templado normal falso si
templado alta verdad no
```

En este ejemplo es la primera vuelta del algoritmo, donde te escoge como nodo principal TiempoExterior y se divide en 3 ramas: soleado, nublado y lluvioso.

Cada rama tiene las nuevas variables con las que se usará la recursión y se volvería a repetir el proceso haciendo el árbol cada vez más pequeño hasta llegar a la decisión final.

Al ejecutar, por consola muestra cada resultado de las variables (Primera imagen)

## DETALLES DE LA IMPLEMENTACIÓN

El algoritmo ID3 lo he programado con el lenguaje de programación Java.

El algoritmo cuenta con 4 clases principales:

**CargarDatos:** En esta clase, tengo como atributos privados el nombre de los ficheros donde tengo guardado el nombre de las columnas y las variables.

Me encargo de ver si existen dichos ficheros, si es así tengo otros atributos donde guardo en ellos los datos de los textos. En caso de que no existan los ficheros saltaría un mensaje de error.

**Elementos:** En esta clase, represento cada valor de todos los datos, donde guardo el número de repeticiones, el número de positivos y negativos, el valor  $r$  y el mérito individual de cada variable.

**Nodo:** En esta clase es donde guardo cada nodo nuevo que me surja durante la recursión. Mi nodo es cada variable nueva que tenga, con sus datos nuevos, su antecesor (el padre) y el nombre de las columnas guardadas. Hasta que no estén los nodos hechos por completo. La recursión continuará.

**ID3:** La clase principal, en ella realizo todo el algoritmo. En esta clase tengo:

**Dos variables String** con el nombre de los ficheros donde quiero que me los guarde

**Tres HashMap** donde guardo el mérito y valores de cada variable y también otro donde guardo para cada nombre de columna, los datos de esa columna.

**Varios arrayList** donde voy guardado los nodos, los datos y las respuestas.

En esta clase contengo **dos métodos**, **primeraVuelta()**, con el nodo en null, el cual es el encargado en primero hacer una vuelta a los datos y en sacar el padre principal del árbol y sus ramificaciones.

Después tenemos el método **recursión()**, el cual aparte de contabilizar las veces que se hace la recursividad, vuelve a llamar a la función primeraVuelta y va simplificando el árbol hasta acabar con un único nodo final que es la solución.



# MANUAL DE USUARIO

El funcionamiento de la práctica es sencillo.

Es necesario tener instalado Java y una máquina en donde poder ejecutar el src, para esta práctica no se dispone de un ejecutable.

Se le dará dos .txt iniciales con las pruebas del enunciado y también he añadido otros modelos de prueba para comprobar el funcionamiento mismo del algoritmo.

Al ejecutar, en consola le irán saliendo ciertos pasos que está haciendo el algoritmo, pero en los .txt creados saldrá lo esencial del algoritmo, méritos y soluciones.

Para poder cambiar los modelos de prueba debe ir a la clase CargarDatos e insertar en:

**\_ficheroAtributos:** El .txt donde contengan el nombre de las columnas.

**\_ficheroDatos:** El .txt donde contengan los datos de cada columna.

A la derecha, en comentarios le saldrán los otros .txt disponibles en la carpeta **pruebas** por si desea hacer alguna prueba más con otros .txt.

## BIBLIOGRAFÍA

<http://www.cs.us.es/~fsancho/?e=38>

[https://www.nebrija.es/~cmalagon/inco/apuntes\\_mios/ejemplo\\_ID3\\_clase.pdf](https://www.nebrija.es/~cmalagon/inco/apuntes_mios/ejemplo_ID3_clase.pdf)

<https://es.slideshare.net/FernandoCaparrini/arboles-decision-id3>

[https://es.wikipedia.org/wiki/Algoritmo\\_ID3](https://es.wikipedia.org/wiki/Algoritmo_ID3)

[https://ceal.ingenieria.uncuyo.edu.ar/data\\_mining/Algoritmos/algoritmo1.pdf](https://ceal.ingenieria.uncuyo.edu.ar/data_mining/Algoritmos/algoritmo1.pdf)

[https://www.researchgate.net/figure/Figura-3-Pseudocodigo-del-algoritmo-ID3-La-implementacion-de-algoritmo-ID3-se-realizo\\_fig2\\_307777124](https://www.researchgate.net/figure/Figura-3-Pseudocodigo-del-algoritmo-ID3-La-implementacion-de-algoritmo-ID3-se-realizo_fig2_307777124)

**Material de clase**