Práctica 2

ID3

Ingeniería del Software

UCM

Andrei Ionut Vaduva

Lenguaje utilizado, procedimientos utilizados, partes opcionales y detalles necesarios.

Para la implementación de esta práctica se ha utilizado JavaScript, Bootstrap, HTML, Css y una librería para dibujar el árbol, Treant.

El proyecto se ha estructurado según:

1. Public:
    a. Javascripts
    b. Stylesheets

La carpeta Javascripts contiene todos los scripts de la práctica, tanto los de Bootstrap, como JQuery y los scripts que contienen el algoritmo id3. En cuanto a los scripts de la práctica, tenemos dos.

- reader.js: Es un script que se encarga de leer tanto el fichero de atributos como el fichero de datos. Si el número de atributos no coincide con el número de datos por fila, se lanza un error que avisa al usuario de la incorrecta especificación de los datos.
- Index.js: Es el script que se encarga de llamar a las funciones de lectura del "reader.js" y de llamar a la función recursiva id3, que es la encargada de crear un árbol de decisión, haciendo sendas llamadas recursivas hasta completar el árbol. Posteriormente se imprime las tablas con las fórmulas de la ganancia y por último el árbol de decisión y la opción de introducir una serie de datos y obtener una respuesta.

Función ID3:

Como ya se ha mencionado, esta función es la que se encarga de crear el árbol de decisión. Tiene dos parámetros: "datos" y "atributos". El parámetro de datos contiene los datos del fichero que elegimos y tiene tantas palabras por línea como atributos hay. Si esto no se cumple, el algoritmo para y se avisa al usuario.

La estructura de estos dos parámetros ha de ser la siguiente:

- "datos": Ha de tener los valores de los atributos organizados de la siguiente manera: dato1, dato2, dato3, dato4….

- “atributos”: Es un array que contiene los atributos de la siguiente forma: atrib1, atrib2, atrib3, atrib4 ….

De esta forma el programa interpreta que el dato1 es el valor del atrib1, y que el dato2 es el valor del atrib2 y así sucesivamente.

Una vez recibidos esto parámetros correctamente, el algoritmo comienza calculando el dominio de cada atributo, de forma que obtenemos un array de arrays donde cada posición corresponde al dominio de un atributo.

Después se recorre este array y se empiezan a calcular los ejemplos positivos y se recolecta cierta información que se guarda en un objeto.

Concluido este bucle, se calculan los méritos y se imprimen las tablas y los cálculos realizados.

Al tener los méritos calculados el algoritmo recoge la información del atributo que mejor ganancia tiene y mira si se tienen que seguir procesando datos, pero esta vez en función de ese atributo.

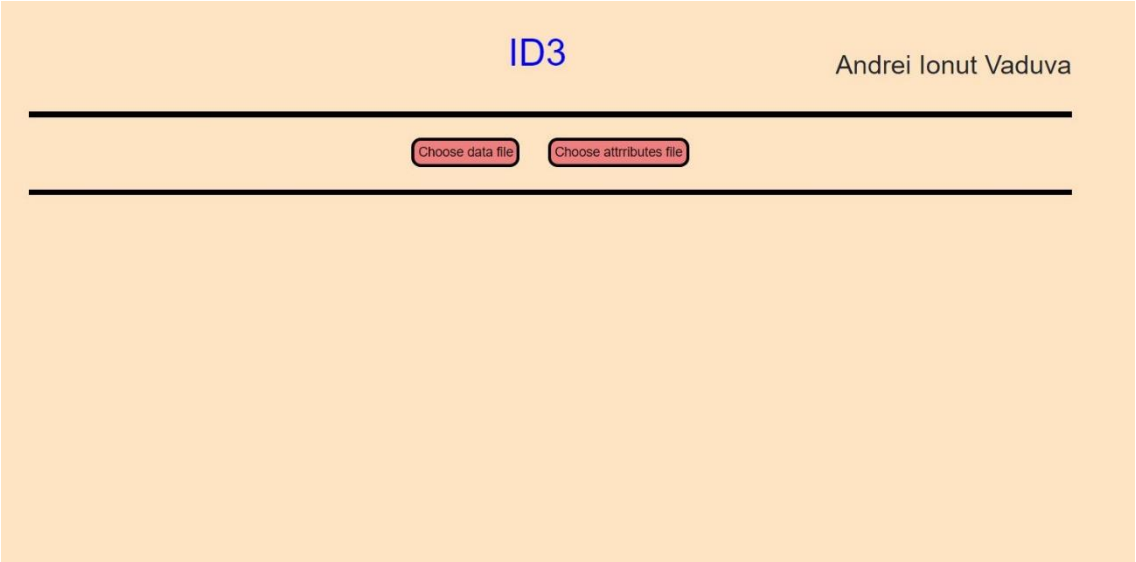
De esta forma, si un atributo ha sido decisivo en el árbol, se guarda su correspondiente rama.

Si, por el contrario, al elegir ese atributo se tienen que comprobar otros atributos, la función id3 prepara los nuevos datos y se hacen llamadas recursivas hasta que se llegue a un atributo decisivo en cada rama del árbol.

Al finalizar la función id3, se llama a la función que dibuja el árbol y ofrece la posibilidad de introducir valores y ver la solución del árbol.

Manual de usuario

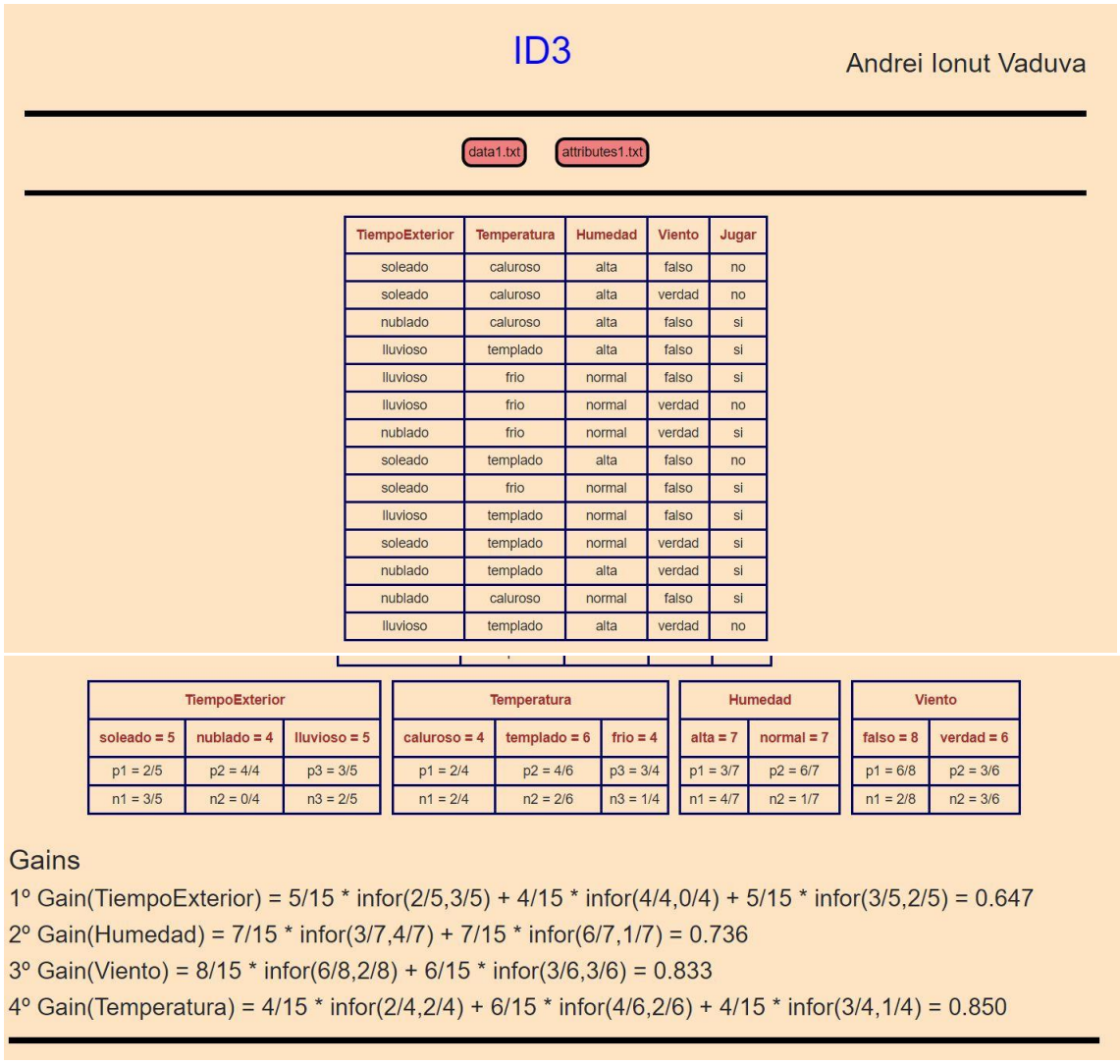
Lo primero que tenemos que hacer es abrir el archivo index.html y se nos abre la siguiente página:



Como podemos observar, en la parte central tenemos dos botones, que nos piden dos ficheros, uno de datos y otro de atributos.

MUY IMPORTANTE: primero hay que elegir el archivo de datos, y después el archivo de atributos.

Automáticamente al elegir el archivo de atributos (ambos correctos), se ejecuta el algoritmo id3, y muestra toda la información en la página:



ID3

Andrei Ionut Vaduva

data1.txt attributes1.txt

TiempoExterior	Temperatura	Humedad	Viento	Jugar
soleado	caluroso	alta	falso	no
soleado	caluroso	alta	verdad	no
nublado	caluroso	alta	falso	si
lluvioso	templado	alta	falso	si
lluvioso	frio	normal	falso	si
lluvioso	frio	normal	verdad	no
nublado	frio	normal	verdad	si
soleado	templado	alta	falso	no
soleado	frio	normal	falso	si
lluvioso	templado	normal	falso	si
soleado	templado	normal	verdad	si
nublado	templado	alta	verdad	si
nublado	caluroso	normal	falso	si
lluvioso	templado	alta	verdad	no

TiempoExterior			Temperatura			Humedad		Viento	
soleado = 5	nublado = 4	lluvioso = 5	caluroso = 4	templado = 6	frio = 4	alta = 7	normal = 7	falso = 8	verdad = 6
p1 = 2/5	p2 = 4/4	p3 = 3/5	p1 = 2/4	p2 = 4/6	p3 = 3/4	p1 = 3/7	p2 = 6/7	p1 = 6/8	p2 = 3/6
n1 = 3/5	n2 = 0/4	n3 = 2/5	n1 = 2/4	n2 = 2/6	n3 = 1/4	n1 = 4/7	n2 = 1/7	n1 = 2/8	n2 = 3/6

Gains

1º Gain(TiempoExterior) = 5/15 * infor(2/5,3/5) + 4/15 * infor(4/4,0/4) + 5/15 * infor(3/5,2/5) = 0.647

2º Gain(Humedad) = 7/15 * infor(3/7,4/7) + 7/15 * infor(6/7,1/7) = 0.736

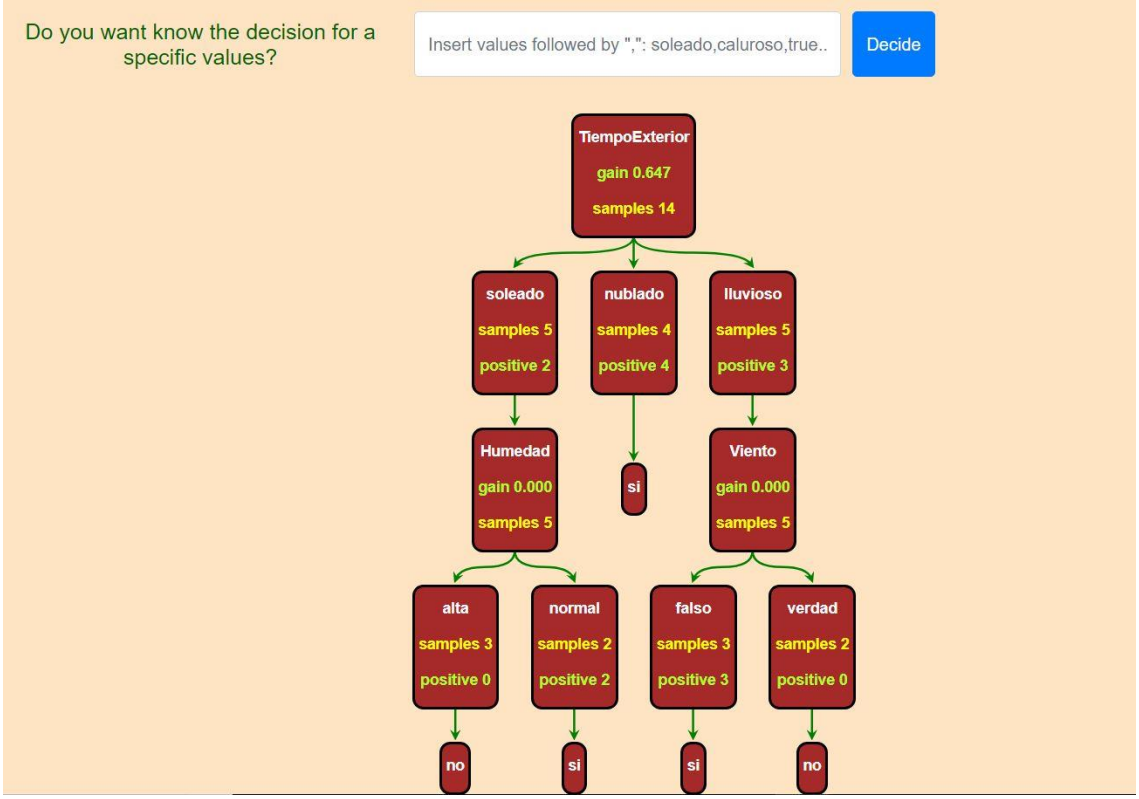
3º Gain(Viento) = 8/15 * infor(6/8,2/8) + 6/15 * infor(3/6,3/6) = 0.833

4º Gain(Temperatura) = 4/15 * infor(2/4,2/4) + 6/15 * infor(4/6,2/6) + 4/15 * infor(3/4,1/4) = 0.850

La estructura es sencilla, por cada paso que da el algoritmo se muestra la información, de esta forma, la primera tabla corresponde a los atributos y a los valores de los datos. Las siguientes subtablas contienen información por cada atributo.

Por último, tenemos las fórmulas que se han utilizado para calcular los méritos y los valores de estos.

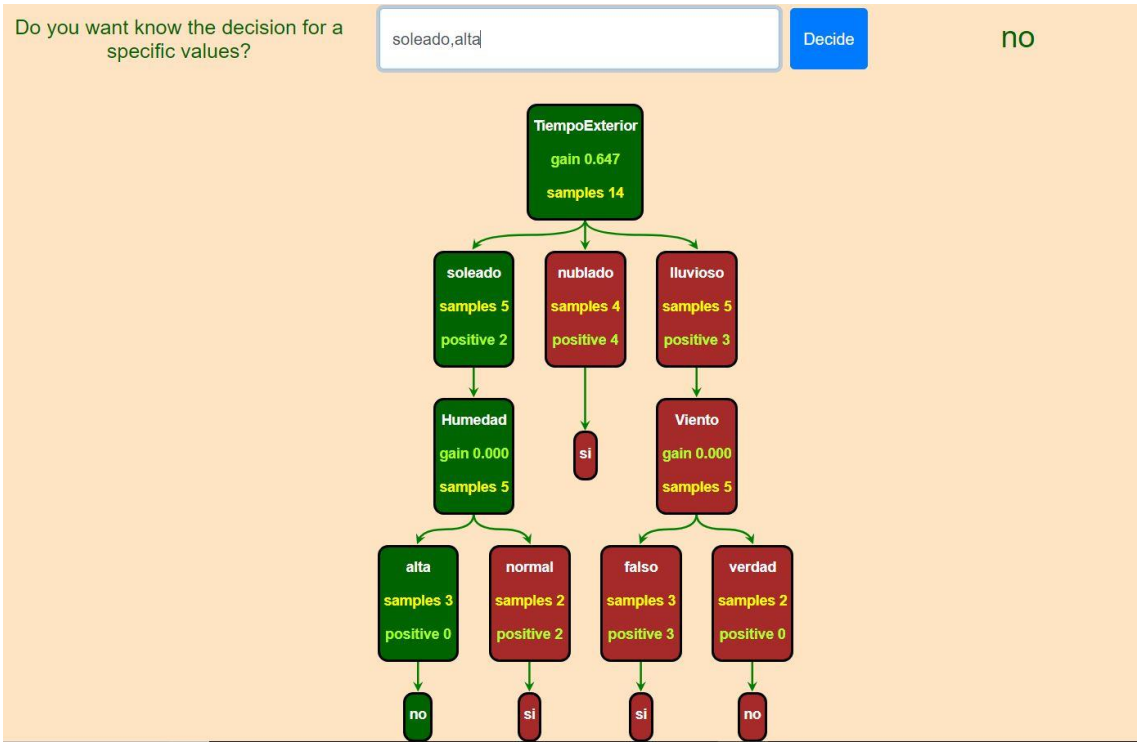
Al hacer scroll hasta abajo del todo, nos encontramos con el árbol de decisión creado y con la opción de introducir valores para calcular la decisión.



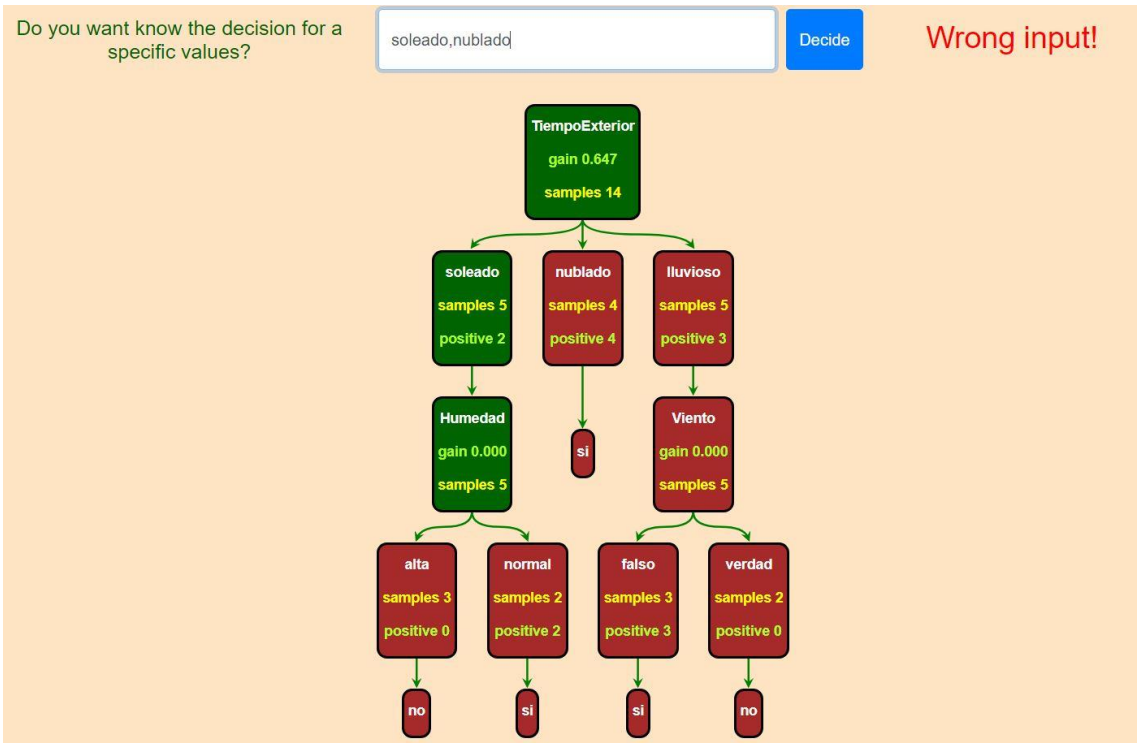
Como podemos observar el árbol de decisión tiene en cada nodo diferentes tipos de información dependiendo de si es un nodo padre o un nodo hijo. En las raíces de este árbol encontramos siempre las decisiones finales.

En cuanto a la parte de averiguar la respuesta en función de los valores introducidos:

- Si los valores introducidos son correctos, el programa nos dirá la respuesta y pintará el camino en el árbol.



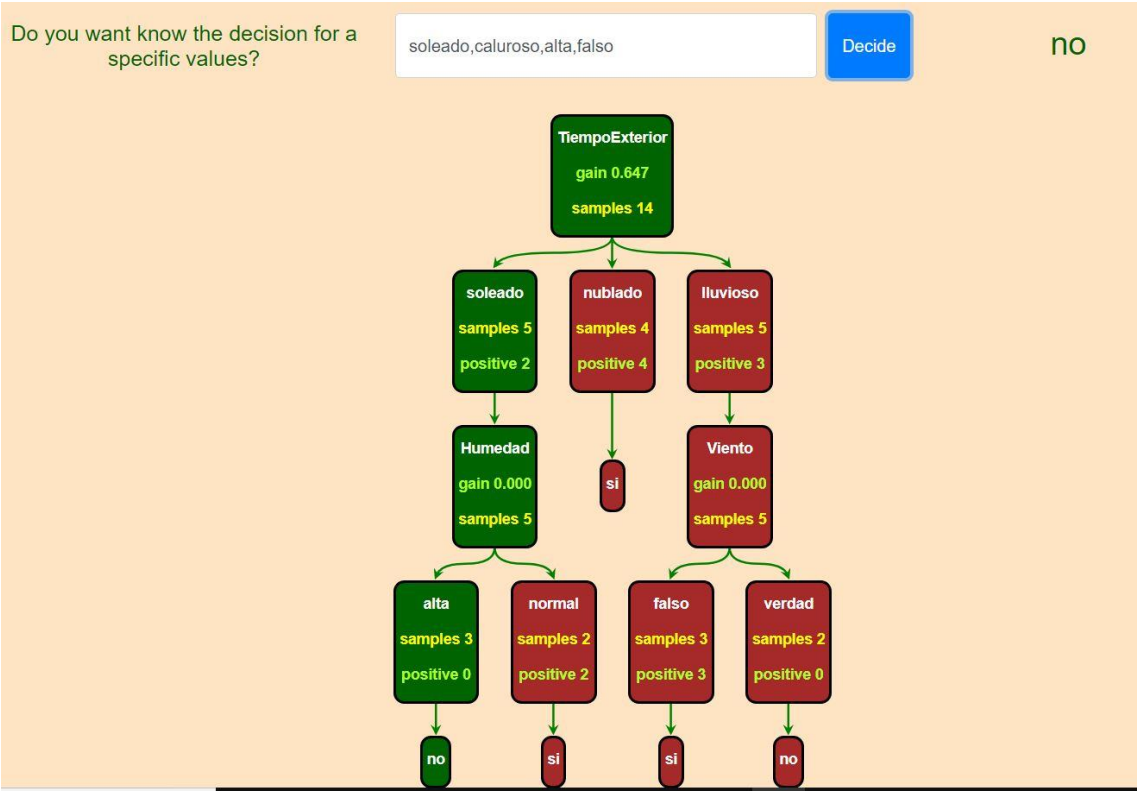
- Si, por el contrario, los valores introducidos no son suficientes o simplemente no corresponden al árbol, se muestra un mensaje de error y se dibuja el árbol hasta la última posición correcta.



Como podemos ver en la imagen de arriba, los valores se tienen que introducir en el orden original, de forma que el primer valor tiene que corresponder al primer atributo, el segundo al atributo dos y así sucesivamente. Si un atributo no esta en la rama, se pueden obviar, es decir, supongamos que una entrada de valores completa sería: “soleado, caluroso,alta,falso”, donde “soleado” corresponde al atr1, TemperaturaExterior; “caluroso ” corresponde al atr2, Temperatura, .. etc.

Sin embargo, si introducimos solamente: “soleado, alta”, obtenemos respuesta ya que el programa encuentra la rama que corresponde a esos atributos.

Si introducimos todos los atributos y hay una rama que los cumple, se mostrará la decisión al obviar los que no sean necesarios:



Los valores tienen que ser introducidos seguidos de comas.

Esta práctica funciona para cualquier archivo, siempre que sigan la estructura descrita. Es configurable a otras estructuras, pero modificando ciertas variables del código, como por ejemplo la variable de decisión, que en este caso tiene que ser un “si”.

Junto a la entrega, se adjuntan otros cuatro ficheros, dos de datos y dos de atributos para que se puedan utilizar para comprobar el funcionamiento.