Historia de las Matemáticas

TRABAJO DE COMPUTACIÓN

TEORÍA DE LA INFORMACIÓN

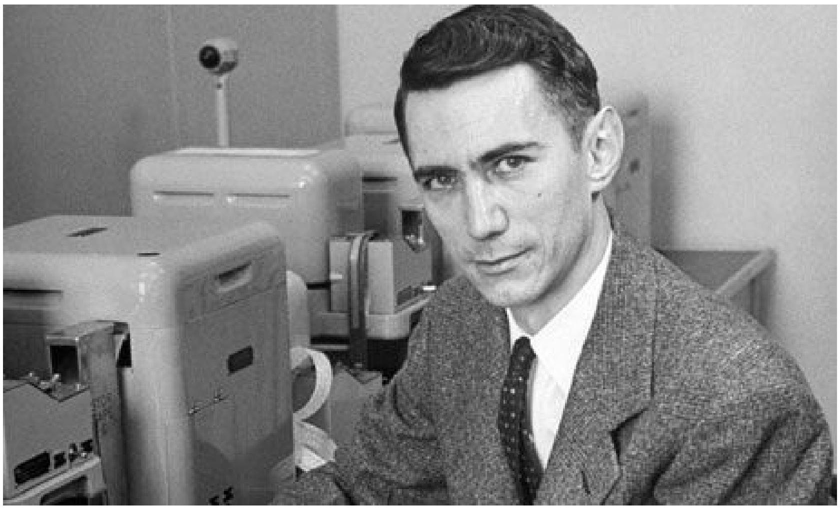
Árboles de decisión

Doble Grado en Ingeniería Informática y Matemáticas

Ramón Casado Gómez
Miguel Sánchez Maldonado
María Oliver Balsalobre

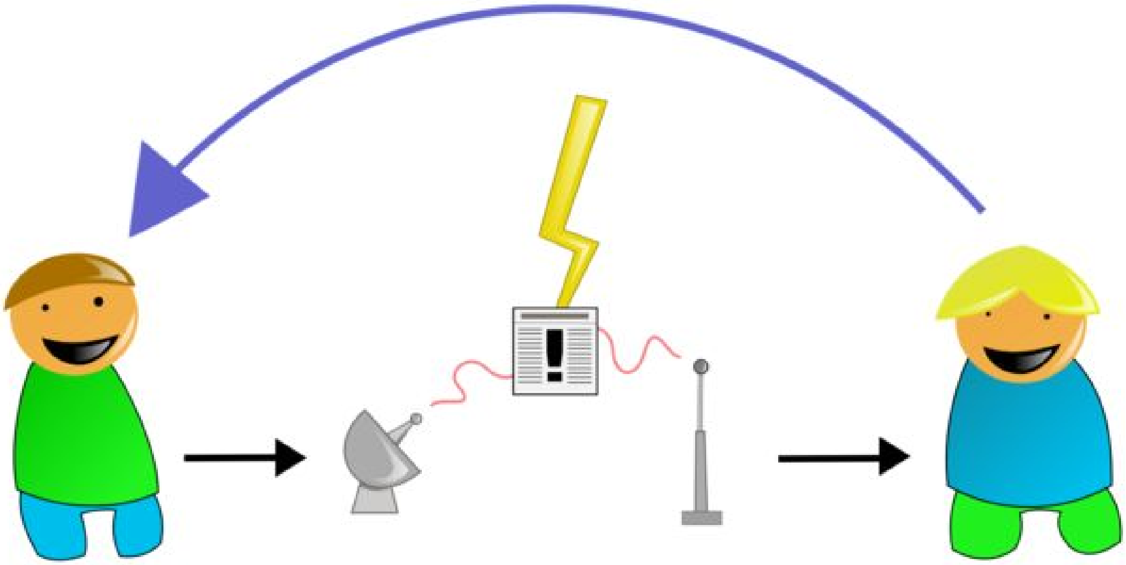
ÍNDICE

1. Introducción



Claude Elwood Shannon (1916-2001)

La teoría de la información surgió a finales de la Segunda Guerra Mundial, en los años cuarenta. En esta época se buscaba la transmisión óptima de los mensajes. Esta teoría es el resultado de trabajos a comienzos del siglo XX. Seguidos por Alan Turing en 1936, el cual realizó el esquema de una máquina capaz de tratar información con emisión de símbolos, y finalmente Claude Elwood Shannon, estadounidense, conocido como "el padre de la teoría de la información", junto a Warren Weaver, contribuyó en la culminación y el asentamiento de la Teoría Matemática de la Comunicación de 1949 con un artículo llamado *Una teoría matemática de la comunicación*. La necesidad de una base teórica para la tecnología de la comunicación surgió del aumento de la complejidad y de la masificación de las vías de comunicación, tales como el teléfono, las redes de teletipo y los sistemas de comunicación por radio. También abarca todas las restantes formas de transmisión y almacenamiento de información, incluyendo la televisión y los impulsos eléctricos que se transmiten en las computadoras y en la grabación óptica de datos e imágenes. Idealmente, los datos se pueden restaurar a su forma original al llegar a su destino. En algunos casos, sin embargo, el objetivo es permitir que los datos de alguna forma se conviertan para la transmisión en masa, se reciban en el punto de destino y sean convertidos fácilmente a su formato original, sin perder ninguna de la información transmitida.



Modelo de comunicación de Claude Elwood Shannon y Warren Weaver (1949)

La transmisión de datos no es tan rápida como quisiéramos, los dispositivos de almacenamiento no tienen capacidad ilimitada, las cantidades muy grandes de datos son poco manejables...Sería interesante reducir la cantidad de datos sin perder información.
Nos muestra, entre otras cosas, el camino a seguir para determinar la cantidad de información útil de unos datos y para comprimir la información de manera que los datos se representan de una manera eficiente. Se desarrolla en términos de probabilidades ya que la información tiene una naturaleza aleatoria. Como en la realidad no disponemos de las probabilidades necesarias, habría que estimarlas de los datos existentes.

Resumiendo, la teoría de la información está relacionada con las leyes matemáticas que rigen la transmisión y el procesamiento de la información y se ocupa de la medición de la información y de la representación de la misma, así como también de la capacidad de los sistemas de comunicación para transmitir y procesar información. Es una rama de la teoría matemática y de las ciencias de la computación que estudia la información y todo lo relacionado con ella.

2. Entropía

La entropía, en la teoría de la información, es una magnitud que mide la información provista por una fuente de datos, es decir, lo que nos aporta sobre un dato o hecho concreto. O lo que es lo mismo, es la medida de la incertidumbre que hay en un sistema. Es decir, ante una determinada situación, la probabilidad de que ocurra cada uno de los posibles resultados.
Frente a grandes cantidades de información se genera cierta incertidumbre que es medida por la entropía, en tal sentido la teoría de la información de Shannon afirma que cuanta menos información haya, mayor será su entropía, es decir, que la información reduce la incertidumbre o la entropía.
El cálculo de la entropía viene dado por la siguiente fórmula:

$$Ent\,(D) \ = (-\frac{|P|}{|D|}log_2\frac{|P|}{|D|} - \frac{|N|}{|D|}log_2\frac{|N|}{|D|})$$

La entropía es una cota inferior para cualquier codificación. O dicho de otro modo, necesitas al menos esa cantidad de bits para codificar ese conjunto de opciones. Con menos bits perderás información.
Cuanto más asimétrica es la distribución de probabilidades menos bits se necesitan. Esto es así porque el conjunto es, en cierto sentido, predecible y podemos aprovechar eso para codificarlo de manera inteligente y ahorrar espacio.

Resumiendo, cuanto más homogénea sea la distribución de probabilidades mayor será la entropía y no podemos encontrar ningún método que codifique información con menos bits de los que marca la entropía.

3. Árboles de decisión.

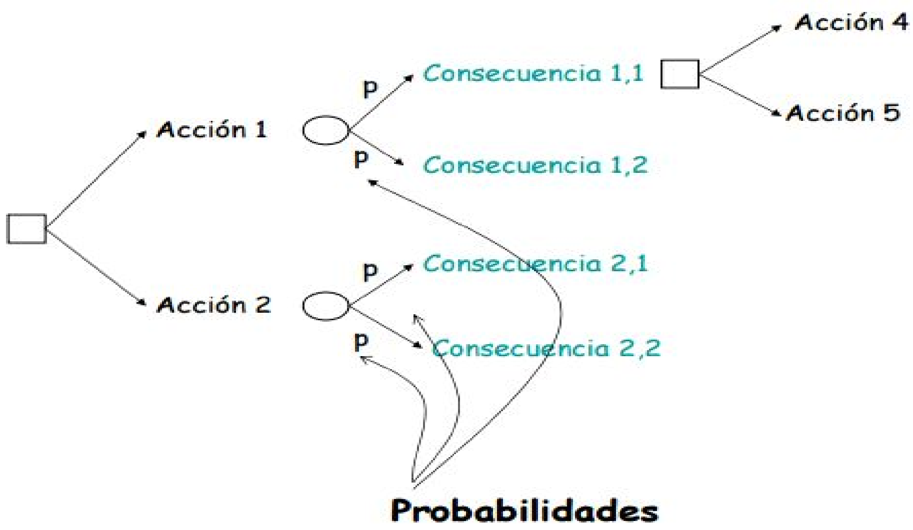
Un árbol de decisión es un modelo de predicción utilizado en el ámbito de la inteligencia artificial. Será una forma gráfica y análitica de representar todos los eventos (sucesos) que pueden surgir a partir de una decisión asumida en cierto momento y por lo tanto nos ayudará a tomar la decisión "más acertada" desde un punto de vista probabilístico, ante posibles decisiones. Además permite desplegar visualmente un problema y organizar el trabajo de cálculos que deben realizarse.
Por tanto dada un base de datos se fabrican diagramas de construcciones lógicas, muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que ocurren de forma sucesiva, para la resolución de un problema.

Un árbol de decisión tiene unas entradas, que pueden ser un objeto o una situación descrita por medio de un conjunto de atributos, y a partir de esto, devuelve una respuesta, esto es, una decisión que es tomada a partir de las entradas y las salidas, que pueden ser valores discretos o continuos.

- Valores discretos denominados clasificación.
- Valores continuos denominados regresión.

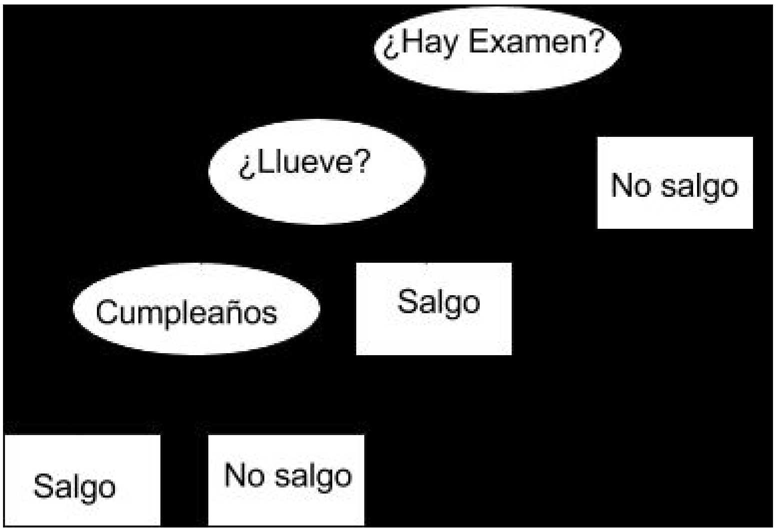
Un árbol de decisión lleva a cabo un test a medida que este se recorre hacia las hojas para alcanzar así una decisión. El árbol de decisión suele contener nodos internos, nodos de probabilidad, nodos hojas y arcos.

Gráficamente:



→ **<u>Pasos para el Análisis del Árbol de decisión</u>**:

1. Definir un problema y dibujar el árbol.
2. Asignar probabilidad a los eventos aleatorios.
3. A continuación, estimar los resultados para cada combinación posible de alternativas.
4. Resolver el problema obteniendo como solución la ruta que proporcione la política óptima.


¿Hay Examen?
¿Llueve?
No salgo
Cumpleaños
Salgo
Salgo
No salgo

4. Algoritmos

Los algoritmos de árboles de decisión construyen modelos de regresión o clasificación en forma de estructura de árbol. Habitualmente, dividen el conjunto de datos en conjuntos cada vez más pequeños mientras van construyendo el árbol de decisión asociado de una forma recursiva. El resultado final es un árbol formado por nodos de decisión y nodos hojas. Cada nodo de decisión tiene dos o más ramas que representan las posibilidades de esa decisión, mientras que los nodos hoja representan la clasificación correspondiente respecto de la propiedad que se quiere clasificar, representado por el nodo raíz del árbol.

4.1 ID3

El algoritmo ID3 (Induction Decision Trees), fue desarrollado por J.Ross Quinlan en 1983. Pertenece a la familia TDIDT (Top-Down Induction of Decision Trees) y su objetivo es construir un árbol de decisión que explique cada instancia de la secuencia de entrada de la manera más compacta posible a partir de una tabla de inducción. En cada momento, elige el mejor atributo dependiendo de una determinada heurística y determina las variables que contienen información relevante para la solución del problema.

Función ID3 (E, A, N): N

E: conjunto de ejemplos
A: conjunto de atributos con sus posibles valores
N: nodo raíz del árbol de decisión

```
Si (A := ∅) ó todos los ejemplos de E pertenecen a la misma clase
Entonces clase-nodo(N)=clase-mayoritaria(E)
Si no A_i=mejor-atributo(A)
      pregunta(N)=A_i
      Para cada valor v de A_i
            H=crear-nodo(A_i, v)
            hijos(N)=hijos(N)+H
            E_i = {e ∈ E | valor(e, A_i) = v}
            ID3(E_i, A − {A_i}, H)
Devolver N
```

Procedimiento (algoritmo) ID3

```
Si todos los ejemplos de E pertenecen a una misma clase C, entonces
   arbol1 <-- nodo etiquetado con C
SiNo
   Si a = f, entonces
      C <-- clase mayoritaria de los ejemplos de E
      arbol1 <-- nodo etiquetado con C
   SiNo
      A <-- mejor atributo de a
      arbol1 <-- nodo etiquetado con A
      Para cada v perteneciente a los valores de A, hacer
         EAv <-- los ejemplos de E que tienen el valor v para el atributo A
         Si EAv = f, entonces
            arbol2 <-- nodo etiquetado con la clase mayoritaria en E
         SiNo
            arbol2 <-- ID3(EAv , a-{A})
         arbol1 <-- añadir a arbol1 el arbol2, a través de una rama etiquetada con v
Devolver arbol1
```

Pseudocódigo del algoritmo ID3

→ <u>Características del Algoritmo ID3</u>

- *Recursividad :* el algoritmo realiza un proceso recursivo sobre todas las ramas del árbol generado. Inicialmente, se le llama con el conjunto de ejemplos, el de atributos, y el nodo raíz del árbol de decisión que estará vacío. El proceso se realiza hasta que todos los ejemplos de la rama pertenecen a una única clase.

- Se pueden presentar 4 casos:

   1. Si el nuevo subconjunto tiene ejemplos positivos y negativos, escoger nuevamente el mejor atributo y bifurcar.
   2. Si el nuevo subconjunto está formado por un solo tipo de ejemplos, se ha llegado a un nodo hoja.
   3. Si en el caso 1 no existiesen ya más atributos (datos incorrectos, ruido...), utilizar el voto de mayoría de los ejemplos del nodo padre.
   4. Si el nuevo subconjunto es vacío (datos no representativos), utilizar el voto de mayoría de los ejemplos del nodo padre.

- Construye el árbol de arriba a abajo, de forma directa, sin hacer uso de backtracking (estrategia para encontrar soluciones a problemas que satisfacen restricciones).

- Utiliza la entropía y se usa el concepto de Ganancia de Información para seleccionar el atributo más útil en cada paso.

→ **Inconvenientes**

- Favorece indirectamente a aquellos atributos con muchos valores, los cuales no tienen que ser los más útiles.
- Genera árboles de decisión a partir de ejemplos de partida.
- Conflictos en la base de conocimientos donde diferentes soluciones se alcanzan con variables con los mismos valores asociados.
- Manejo discreto de los valores de las variables (rangos para discretizar una variable continua).
- Generación de grandes árboles de decisión que no representan garantía de reglas eficientes.
- Puede ocurrir que se hayan agotado todos los atributos y, sin embargo, aún existan ejemplos con distintos valores de clase.
- Puede suceder que, una vez elegido un atributo para un nodo de decisión, no exista ningún ejemplo para una de las ramas generadas por dicho atributo.

→ **Extensiones del algoritmo**

- Atributos con valores continuos.
- Otras medidas para seleccionar atributos.
- Otras estimaciones de error
- Atributos sin valores
- Atributos con coste

Para solucionar estos inconvenientes que presenta el algoritmo, se realizaron una serie de cambios, que dieron lugar a los dos siguientes algoritmos, C4.5 y C5.0 (Quinlan).

4.2 C4.5

Es un algoritmo usado para generar un árbol de decisión desarrollado por Ross Quinlan en 1993, y propone una mejor del algoritmo ID3 descrito anteriormente.
Se basa en la utilización del criterio *ratio de ganancia* definido como I(Xi , C)/H(Xi). De esta manera conseguimos que las variables con mayor número de posibles valores salgan beneficiadas en la selección. Además, el algoritmo incorpora una poda del árbol de clasificación una vez que éste ha sido inducido. La poda está basada en la aplicación de un test de hipótesis que trata de responder a la pregunta de si merece la pena expandir o no un determinada rama. Por tanto, construye árboles de decisión desde un grupo de datos de entrenamiento de la misma forma en que lo hace ID3, usando el concepto de entropía de información. En cada nodo del árbol, C4.5 elige un atributo de los datos que más eficazmente dividen el conjunto de muestras en subconjuntos enriquecidos en una clase u otra. Su criterio es el normalizado para ganancia de información que resulta en la elección de un atributo para dividir los datos.

El algoritmo considera todas las pruebas posibles que pueden dividir el conjunto de datos y selecciona la prueba que resulta en la mayor ganancia de información. Para cada atributo discreto, se considera una prueba con n resultados, siendo n el número de valores posibles que puede tomar el atributo. Para cada atributo continuo, se realiza una prueba binaria sobre cada uno de los valores que toma el atributo en los datos. En cada nodo, el sistema debe decidir cuál prueba escoge para dividir los datos.

→**C4.5 propone tres tipos de pruebas posibles**:

1. La prueba "estándar" para las variables discretas, con un resultado y una rama para cada valor posible de la variable.
2. Una prueba más compleja, basada en una variable discreta, en donde los valores posibles son asignados a un número variable de grupos con un resultado posible para cada grupo, en lugar de para cada valor.
3. Si una variable A tiene valores numéricos continuos, se realiza una prueba binaria con resultados $A \leq Z$ y $A > Z$, para lo cual debe determinarse el valor límite Z.

→ **<u>Mejoras que propone frente a ID3 son</u>**:

- Manejo de datos perdidos. A la hora de construir el árbol se ignoran los campos perdidos, de manera que solo se tienen en cuenta los registros que tienen valor para ese atributo.
- Posibilidad de trabajar con datos continuos. Para poder trabajar con datos continuos, C4.5 divide los datos en rangos en base a los valores encontrados en el conjunto de entrenamiento.
- Propone soluciones para el sobreaprendizaje, pudiendo usar pre-poda (se decide cuando dejar de subdividir el árbol) y post-poda (se construye el árbol y después se poda).
- Mejora la eficiencia computacional.
- Determinar qué tan profundo debe crecer el árbol de decisión.

→ **<u>Características principales del algoritmo serán</u>**:

- Nos permite trabajar con valores continuos para los atributos, separando los posibles resultados en 2 ramas, dependiendo de si supera o no el umbral estimado.
- Los árboles son menos frondosos, ya que cada hoja cubre una distribución de clases.
- Utiliza el método divide y vencerás, para generar el árbol de decisión inicial a partir de un conjunto de datos de entrenamiento.
- Se basa en la utilización del criterio de proporción de ganancia. De esta manera, consigue evitar que las variables con mayor número de categorías salgan beneficiadas en la selección.
- Todas las muestras en la lista pertenecen a la misma clase.
- Ninguna de las característica proporciona ninguna ganancia de información.
- Es recursivo.

→ **Pseudocódigo**

```
Función C4.5
R: conjunto de atributos no clasificadores,
C: atributo clasificador,
S: conjunto de entrenamiento, devuelve un árbol de decisión
Comienzo
     Si S está vacío,
          Devolver un único nodo con Valor Falla;      'para formar el nodo raiz
     Si todos los registros de S tienen el mismo valor para el atributo
clasificador,
          Devolver un único nodo con dicho valor;   'un unico nodo para todos
     Si R está vacío,
          Devolver un único nodo con el valor más frecuente del atributo
          Clasificador en los registros de S [Nota: habrá errores, es decir,
          Registros que no estarán bien clasificados en este caso];
     Si R no está vacío,
          D ←atributo con mayor Proporción de Ganancia (D,S) entre los
atributos de R;
          Sean {dj | j=1,2,...., m} los valores del atributo D;
          Sean {dj | j=1,2,...., m} los subconjuntos de S correspondientes a los
valores de dj respectivamente;
             Devolver un árbol con la raíz nombrada como D y con los arcos
nombrados d1, d2,....,dm, que van respectivamente a los árboles
            C4.5(R-{D}, C, Sl), C4.5(R-{D}, C, S2), C4.5(R-{D}, C, Sm);
Fin
```

Pseudocódigo del algoritmo C4.5

4.3 C5

Quinlan continuó con la creación del C5.0 y el See5 (C5.0 para Unix/Linux, See5 para Windows) con fines comerciales. C5.0 ofrece una serie de mejoras con respecto a C4.5. Algunas de estas son:

- Velocidad - C5.0 es significativamente más rápido que el C4.5 (varios órdenes de magnitud)
- El uso de memoria - C5.0 es más eficiente que el C4.5
- Árboles de decisión más pequeños - C5.0 obtiene resultados similares a C4.5 con árboles de decisión mucho más pequeños.
- Soporte para boosting - Boosting mejora los árboles y les da una mayor precisión.
- Ponderación - C5.0 le permite ponderar los distintos casos y tipos de errores de clasificación.

5. Ejemplo algoritmo ID3

	Cielo	Temperatura	Humedad	Viento	Posibilidad de jugar al tenis
Día 1	Sol	Alta	Alta	Débil	-
Día 2	Sol	Alta	Alta	Fuerte	-
Día 3	Nubes	Alta	Alta	Débil	+
Día 4	Lluvia	Suave	Alta	Débil	+
Día 5	Lluvia	Baja	Normal	Débil	+
Día 6	Lluvia	Baja	Normal	Fuerte	-
Día 7	Nubes	Baja	Normal	Fuerte	+
Día 8	Sol	Suave	Alta	Débil	-
Día 9	Sol	Baja	Normal	Débil	+
Día 10	Lluvia	Suave	Normal	Débil	+
Día 12	Sol	Suave	Normal	Fuerte	+
Día 13	Nubes	Suave	Alta	Fuerte	+
Día 14	Nubes	Alta	Normal	Débil	+
Día 15	Lluvia	Suave	Alta	Fuerte	-

	Cielo	Temperatura	Humedad	Viento	Posibilidad de jugar al tenis
Día 1	Sol	Alta	Alta	Débil	-
Día 2	Sol	Alta	Alta	Fuerte	-
Día 3	Nubes	Alta	Alta	Débil	+
Día 4	Lluvia	Suave	Alta	Débil	+
Día 5	Lluvia	Baja	Normal	Débil	+
Día 6	Lluvia	Baja	Normal	Fuerte	-
Día 7	Nubes	Baja	Normal	Fuerte	+
Día 8	Sol	Suave	Alta	Débil	-
Día 9	Sol	Baja	Normal	Débil	+
Día 10	Lluvia	Suave	Normal	Débil	+
Día 12	Sol	Suave	Normal	Fuerte	+
Día 13	Nubes	Suave	Alta	Fuerte	+
Día 14	Nubes	Alta	Normal	Débil	+
Día 15	Lluvia	Suave	Alta	Fuerte	-

	Cielo	Temperatura	Humedad	Viento	Posibilidad de jugar al tenis
Día 1	**Sol**	**Alta**	**Alta**	**Débil**	**-**
Día 2	**Sol**	**Alta**	**Alta**	**Fuerte**	**-**
Día 3	**Nubes**	**Alta**	**Alta**	**Débil**	**+**
Día 4	**Lluvia**	**Suave**	**Alta**	**Débil**	**+**
Día 5	**Lluvia**	**Baja**	**Normal**	**Débil**	**+**
Día 6	**Lluvia**	**Baja**	**Normal**	**Fuerte**	**-**
Día 7	**Nubes**	**Baja**	**Normal**	**Fuerte**	**+**
Día 8	**Sol**	**Suave**	**Alta**	**Débil**	**-**
Día 9	**Sol**	**Baja**	**Normal**	**Débil**	**+**
Día 10	**Lluvia**	**Suave**	**Normal**	**Débil**	**+**
Día 12	**Sol**	**Suave**	**Normal**	**Fuerte**	**+**
Día 13	**Nubes**	**Suave**	**Alta**	**Fuerte**	**+**
Día 14	**Nubes**	**Alta**	**Normal**	**Débil**	**+**
Día 15	**Lluvia**	**Suave**	**Alta**	**Fuerte**	**-**

Aplicaremos el algoritmo ID3 y utilizaremos la fórmula de la entropía de un conjunto de ejemplos, D (en nuestro caso los Días)

$$Ent\ (D)\ = (-\ \frac{|P|}{|D|} log_2 \frac{|P|}{|D|}\ -\ \frac{|N|}{|D|} log_2 \frac{|N|}{|D|})$$

donde P serán los subconjuntos de ejemplos positivos de D (es decir, los días en que es posible jugar al tenis) y N los subconjuntos de ejemplos negativos de D.
Además, calcularemos cuál será la entropía esperada después de usar un atributo A en el árbol, con la fórmula siguiente:

$$\sum_{v\ \varepsilon\ Valores(A)} \frac{|D_v|}{|D|} Ent(D_v)$$

y la ganancia de información esperada después de usar un atributo A:

$$Ganancia(D,A)\ =\ Ent(D)\ -\ \sum_{v\ \varepsilon\ Valores(A)} \frac{|D_v|}{|D|} Ent(D_v)$$

donde D_v es el subconjunto de ejemplos de D con valor del atributo A igual a v.
Como vamos a emplear, como ya se ha dicho anteriormente, el algoritmo ID3, en cada nodo usamos el atributo con mayor ganancia de información (considerando los ejemplos correspondientes al nodo).

- **Cielo**: $Ent([9+,5-])\ =\ -\frac{9}{14}\cdot log_2(\frac{9}{14}) - \frac{5}{14}\cdot log_2(\frac{5}{14})\ =\ 0'94$
  - Sol: $Ent([2+,3-])\ =\ -\frac{2}{14}\cdot log_2(\frac{2}{14}) - \frac{3}{14}\cdot log_2(\frac{3}{14})\ =\ 0'97$
  - Nubes: $Ent([4+,0-])\ =\ 0$
  - Lluvia: $Ent([2+,3-])\ =\ -\frac{2}{14}\cdot log_2(\frac{2}{14}) - \frac{3}{14}\cdot log_2(\frac{3}{14})\ =\ 0'97$

  $Entropía(D,Cielo)\ =\ \frac{5}{14}\cdot 0'97 + \frac{5}{14}\cdot 0'97\ =\ 0'7$
  $Ganancia(D,Cielo)\ =\ 0'94 - \frac{5}{14}\cdot 0'97 - \frac{5}{14}\cdot 0'97\ =\ 0'24$
- **Temperatura**: $Ent([9+,5-])\ =\ -\frac{9}{14}\cdot log_2(\frac{9}{14}) - \frac{5}{14}\cdot log_2(\frac{5}{14})\ =\ 0'94$
  - Alta: $Ent([2+,2-])\ =\ -\frac{2}{14}\cdot log_2(\frac{2}{14}) - \frac{2}{14}\cdot log_2(\frac{2}{14})\ =\ 1$
  - Suave: $Ent([4+,2-])\ =\ -\frac{4}{14}\cdot log_2(\frac{4}{14}) - \frac{2}{14}\cdot log_2(\frac{2}{14})\ =\ 0'91$
  - Baja: $Ent([3+,1-])\ =\ -\frac{3}{14}\cdot log_2(\frac{3}{14}) - \frac{1}{14}\cdot log_2(\frac{1}{14})\ =\ 0'81$

  $Entropía(D,Temperatura)\ = \frac{4}{14}\cdot 1 + \frac{6}{14}\cdot 0'91 + \frac{4}{14}\cdot 0'81\ =\ 0'92$
  $Ganancia(D,Temperatura)\ =\ 0'94 - \frac{4}{14}\cdot 1 - \frac{6}{14}\cdot 0'91 - \frac{4}{14}\cdot 0'81\ =\ 0'02$

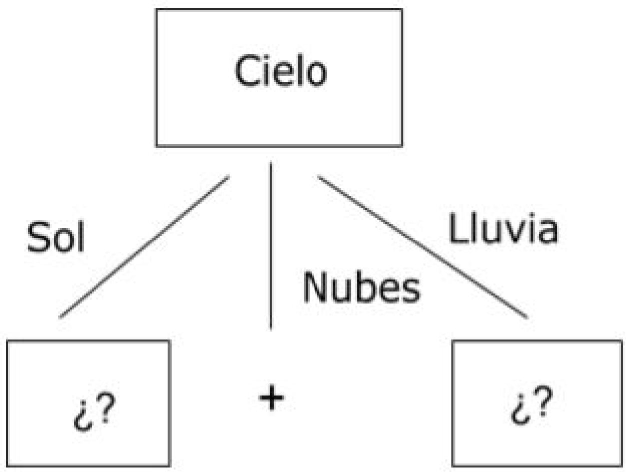
- **Humedad**: $Ent([9+,5-]) \; = \; -\frac{9}{14}\cdot log_2(\frac{9}{14}) - \frac{5}{14}\cdot log_2(\frac{5}{14}) \; = \; 0'94$
    - Alta: $Ent([3+,4-]) \; = \; -\frac{3}{14}\cdot log_2(\frac{3}{14}) - \frac{4}{14}\cdot log_2(\frac{4}{14}) \; = \; 0'99$
    - Normal: $Ent([6+,1-]) \; = \; -\frac{6}{14}\cdot log_2(\frac{6}{14}) - \frac{1}{14}\cdot log_2(\frac{1}{14}) \; = \; 0'59$

$Entropía(D, Humedad) \; = \frac{7}{14}\cdot 0'99 + \frac{7}{14}\cdot 0'59 \; = 0'79$
$Ganancia(D, Humedad) \; = \; 0'94 - \frac{7}{14}\cdot 0'99 - \frac{7}{14}\cdot 0'59 \; = \; 0'15$

- **Viento**: $Ent([9+,5-]) \; = \; -\frac{9}{14}\cdot log_2(\frac{9}{14}) - \frac{5}{14}\cdot log_2(\frac{5}{14}) \; = \; 0'94$
    - Fuerte: $Ent([6+,2-]) \; = \; -\frac{6}{14}\cdot log_2(\frac{6}{14}) - \frac{2}{14}\cdot log_2(\frac{2}{14}) \; = \; 0'81$
    - Débil: $Ent([3+,3-]) \; = \; -\frac{3}{14}\cdot log_2(\frac{3}{14}) - \frac{3}{14}\cdot log_2(\frac{3}{14}) \; = \; 1$

$Entropía(D, Viento) \; = \frac{8}{14}\cdot 0'81 + \frac{6}{14}\cdot 1 \; = \; 0'9$
$Ganancia(D, Viento) \; = \; 0'94 - \frac{8}{14}\cdot 0'81 - \frac{6}{14}\cdot 1 \; = \; 0'04$

Por tanto, seleccionamos el mejor atributo, **Cielo**, ya que es el que produce mayor ganancia de información ($0'24$).



Si recordamos lo anterior, vemos que para el caso en el que haya nubes en el cielo, había 4 subconjuntos de ejemplos positivos y ninguno negativo. Por tanto, es claro que en un día con el cielo nublado, sí es posible jugar al tenis. Pasamos entonces a estudiar los otros dos casos para el nodo Cielo:

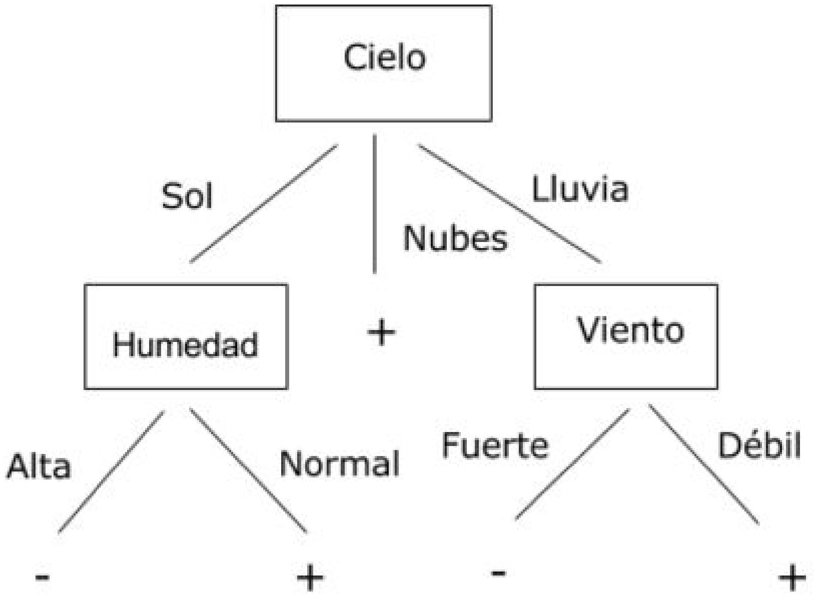
- **Cielo**
    - Sol: $Ent([2+,3-]) \; = \; -\frac{2}{14}\cdot log_2(\frac{2}{14}) - \frac{3}{14}\cdot log_2(\frac{3}{14}) \; = \; 0'97$

- $Entropía(D_{sol}, Temperatura) = \frac{2}{5}{\cdot}0 + \frac{2}{5}{\cdot}1 + \frac{1}{5}{\cdot}0 = 0'4$
- $Ganancia(D_{sol}, Temperatura) = 0'97 - \frac{2}{5}{\cdot}0 - \frac{2}{5}{\cdot}1 - \frac{1}{5}{\cdot}0 = 0'57$
- $Entropía(D_{sol}, Humedad) = \frac{3}{5}{\cdot}0 + \frac{2}{5}{\cdot}0 = 0$
- $Ganancia(D_{sol}, Humedad) = 0'97 - \frac{3}{5}{\cdot}0 - \frac{2}{5}{\cdot}0 = 0'97$ (mejor)
- $Entropía(D_{sol}, Viento) = \frac{2}{5}{\cdot}1 + \frac{3}{5}{\cdot}0'91 = 0'96$
- $Ganancia(D_{sol}, Viento) = 0'97 - \frac{2}{5}{\cdot}1 - \frac{3}{5}{\cdot}0'91 = 0'01$

- Lluvia: $Ent([2+, 3-]) = -\frac{2}{14}{\cdot}log_2(\frac{2}{14}) - \frac{3}{14}{\cdot}log_2(\frac{3}{14}) = 0'97$

  - $Entropía(D_{Lluvia}, Temperatura) = \frac{3}{5}{\cdot}0'91 + \frac{2}{5}{\cdot}1 = 0'15$
  - $Ganancia(D_{Lluvia}, Temperatura) = 0'97 - \frac{3}{5}{\cdot}0'91 - \frac{2}{5}{\cdot}1 = 0'82$
  - $Entropía(D_{Lluvial}, Humedad) = \frac{2}{5}{\cdot}1 + \frac{3}{5}{\cdot}0'91 = 0'15$
  - $Ganancia(D_{Lluvia}, Humedad) = 0'97 - \frac{2}{5}{\cdot}1 - \frac{3}{5}{\cdot}0'91 = 0'82$
  - $Entropía(D_{Lluvia}, Viento) = \frac{3}{5}{\cdot}0 + \frac{2}{5}{\cdot}0 = 0$
  - $Ganancia(D_{Lluvia}, Viento) = 0'97 - \frac{3}{5}{\cdot}0 - \frac{2}{5}{\cdot}0 = 0'97$ (mejor)

Por tanto, ya tenemos información suficiente para construir el árbol minimal que estábamos buscando. Nos quedaría:



6. Aplicaciones

Como hemos visto en los algoritmos ID3, C4.5 y C5, los campos de las aplicaciones pueden ser cualquiera en los que se usen árboles de decisión. Además, tenemos que la teoría de la información tiene múltiples usos como pueden ser la compresión sin pérdida de datos (ej. Archivos ZIP), compresión de datos con pérdida (ej. MP3s), y codificación de canal (ej. Para líneas DSL).
Esta teoría también ha sido crucial en la criptografía, se ha usado en algoritmos como puede ser PGP (Pretty Good Privacy), o RSA, gracias a estos algoritmos existe el internet como se conoce hoy en día.
Por otro lado, ha sido también muy importante para el éxito de las misiones de las sondas espaciales Voyager. La invención del disco compacto y su capacidad están también directamente relacionados con la teoría de la información. La factibilidad de los teléfonos móviles, el estudio de la lingüística y la percepción humana, la comprensión de los agujeros negros, y otros campos numerosos, son ejemplos de usos de la amplias aplicaciones que tiene esta teoría.
Un ejemplo concreto de aplicación en la que se usan árboles de decisión en teoría de la información es Mycin, que es un sistema experto el cual su función principal consiste en el diagnóstico de enfermedades infecciosas de la sangre.

7. Bibliografía

[1] https://es.khanacademy.org/computing/computer-science/informationtheory

[2]http://www.uab.cat/web/estudiar/masteres-oficiales/informacion-general/computo-de-altas-prestaciones-y-tratamiento-de-la-informacion-/-high-performance-computing-and-information-theory-1096480309770.html?param1=1307109711994
[3] Libro: Artificial Intelligence (Winston)
[4] http://www.cise.ufl.edu/~ddd/cap6635/Fall-97/Short-papers/2.htm
[5] http://banzai-deim.urv.net/~riano/teaching/id3-m5.pdf
[6]http://enciclopedia.us.es/index.php/Entrop%C3%ADa_(teor%C3%ADa_de_la_informaci%C3%B3n)
[7]http://cs.uns.edu.ar/~ldm/mypage/data/ss/info/teoria_de_la_informacion1.pdf
[8] Libro: Machine Learning (Michel)
[9]http://www.itnuevolaredo.edu.mx/takeyas/Apuntes/Inteligencia%20Artificial/Apuntes/IA/ID3.pdf
[10]http://banzai-deim.urv.net/~riano/teaching/id3-m5.pdf
[11]http://www.cise.ufl.edu/~ddd/cap6635/Fall-97/Short-papers/2.htm
[12]http://www.ingeba.org/lurralde/lurranet/lur07/07jua/jua07.htm
[13]http://www.dmae.upct.es/~mcruiz/Telem06/Teoria/arbol_decision.pdf
[14]http://tgsentropia.blogspot.com.es/2010/10/shannon-y-la-teoria-de-la-informacion.html