Arboles de decisión

### Software de árbol de decisiones

El software de árbol de decisiones es un tipo de aplicación que se utiliza en [la minería de datos](https://searchsqlserver.techtarget.com/definition/data-mining) para simplificar desafíos estratégicos complejos y evaluar la rentabilidad de las decisiones de investigación y de negocios. La intención es asegurar que un conjunto determinado de datos se describa, categorice y analice con precisión para que se puedan derivar conclusiones significativas.

El software del árbol de decisión utiliza modelos predictivos para lograr resultados, mapeando los valores observados sobre un sujeto a conclusiones sobre el valor objetivo de ese sujeto. Este proceso utiliza [estructuras de árbol](https://searchdatamanagement.techtarget.com/definition/tree-structure) para corresponder a clasificaciones y características, donde las hojas representan clasificaciones y las ramas representan conjunciones de características que conducen a esas clasificaciones.

El software de árbol de decisión se utiliza en una amplia gama de industrias. El análisis de riesgos es una aplicación común en las compañías de seguros, por ejemplo, donde los actuarios utilizan las capacidades de procesamiento de datos y descubrimiento de conocimiento para segmentar y comprender los comportamientos de los clientes. En los [centros de llamadas](https://searchcrm.techtarget.com/definition/call-center) , el software de árbol de decisiones se puede usar para medir los niveles óptimos de personal en diferentes condiciones de tráfico entrante y para evaluar el [ROI](https://searchcio.techtarget.com/definition/ROI) de las inversiones en [análisis de CRM](https://searchcrm.techtarget.com/definition/CRM-analytics) y otras tecnologías relevantes. El software permite la creación de escenarios que muestran todos los costos asociados y el impacto de la tecnología elegida en los gastos generales y la productividad.

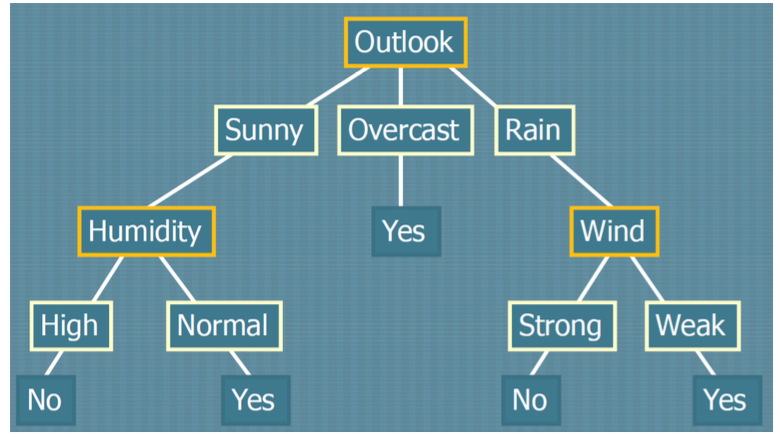
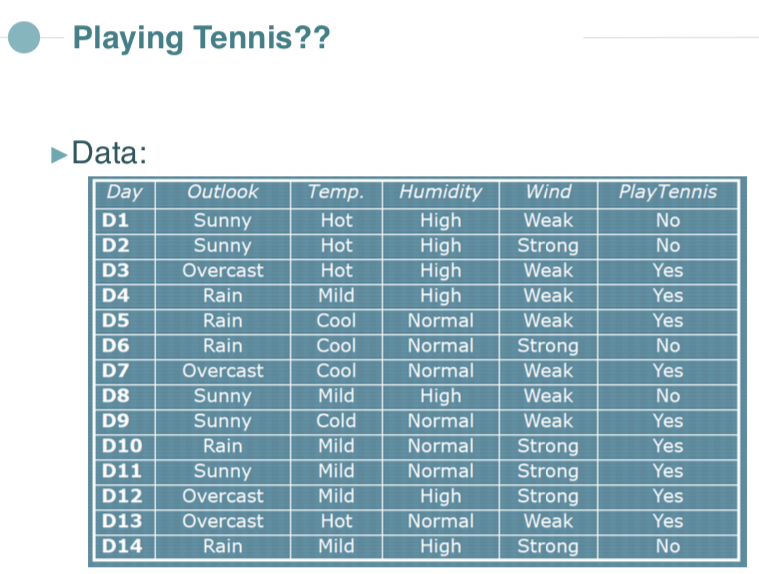
<https://searchcrm.techtarget.com/definition/decision-tree-software>

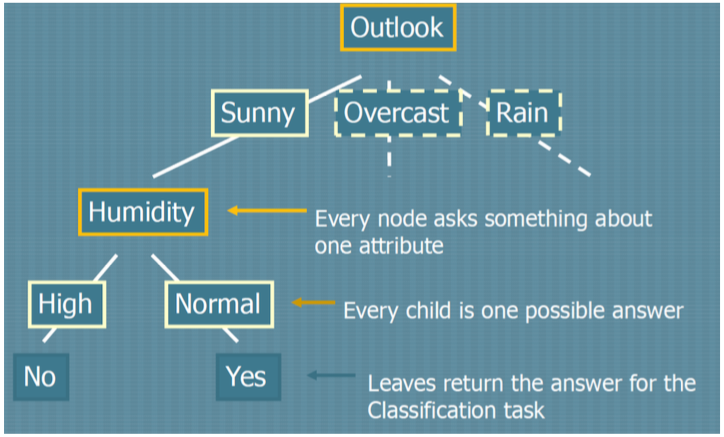
### Decision tree in data mining

* La tarea central en DM es la predicción.
* Clasificación si la respuesta es categórica.
* Regresión si la respuesta es continua.
* Hay muchos métodos (modelos) para realizar la predicción:
* Regresión lineal, regresión logística, análisis discriminante, Naive Bayes, Knn, redes neuronales, SVM, ...
* Los árboles de decisión son un método no paramétrico para realizar predicciones. Su popularidad proviene de la simplicidad de los resultados (todos pueden entenderlos) y de su funcionamiento directo.
* De uso general en minería de datos/ aprendizaje automático.
* Objetivo : crear un modelo que prediga el valor de una variable objetivo en función de varias variables de entrada.
* Ejemplo: Cada nodo interior: variables de entrada;
* Bordes a hijos para cada uno de los valores posibles de esa variable de entrada.
* Hoja: un valor de la variable objetivo dados los valores de las variables de entrada representadas por la ruta desde la raíz a la hoja.
* En el datamining, los árboles de decisión pueden describirse también como la combinación de técnicas matemáticas y computacionales para ayudar a la descripción, categorización y generalización de un conjunto dado de datos.
* Los datos vienen en los registros de la forma:

(X,Y) = (X1,X2,…,XK,Y)   
La variable dependiente, Y, es la variable objetivo que estamos tratando de entender, clasificar o generalizar. El vector X se compone de las características, X1, X2, etc., que se utilizan para esa tarea.

### Playing tennis





### Decision tree: Purpose

**Objetivo**: segmentar los datos para encontrar grupos homogéneos respecto la variable respuesta.

Los árboles difieren según:

* Tipo de variable respuesta (numérica o categórica).
* Tipo de variables explicativas (binarias, nominales, ordinales, numéricas).
* Tipo de árbol (binario, de múltiples vías).
* Criterio de división (información, gini, …)
* Criterio de parada (pre-pruning, post-pruning)

Los arboles de decisión usados en la minería de datos se clasifican en dos grupos principales:

* Árboles de clasificación: es cuando el resultado predicho es la clase a la que pertenecen los datos (por ejemplo; jugar tenis, crédito, tipo de auto, etc.).
* Variable respuesta categórica.
* Árboles de regresión: es cuando el resultado predicho se puede considerar como un número real (por ejemplo; el precio de una casa, la estadía de un paciente en un hospital, etc.).
* Variable respuesta numérica.

### Decision tree learning

* El mejor árbol es el más sencillo/corto.
* Un enfoque: crear todos los árboles y mantener el más corto…
* Pero este enfoque requiere demasiado tiempo de cálculo.
* El algoritmo Gredy construye árboles de decisión cortos de manera eficiente.
* pero no asegura obtener el más corto.

Hay muchos algoritmos específicos de árbol de decisión.

* ID3 (Dichotomiser3 iterativo)
* C4.5 (sucesor de ID3)
* CART (árboles de clasificación y regresión)
* CHAID (Detector Automático de Interacción CHi-cuadrado). Realiza divisiones multinivel al calcular árboles de clasificación.
* MARS: extiende los árboles de decisión para manejar mejor los datos numéricos.
* Árboles de inferencia condicional. Enfoque basado en estadísticas que utiliza pruebas no paramétricas como criterios de división, corregido para múltiples pruebas para evitar el sobreajuste. Este enfoque da como resultado una selección de predictores imparcial y no requiere poda.

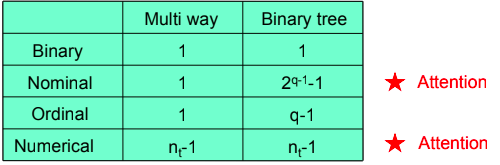
### Decision Tree Learning. Algorithm (top-down)

1. Para establecer todos los individuos en el nodo raíz
2. Evalúe todas las divisiones posibles y encuentre la partición óptima en nodos secundarios.
3. Para cada nodo de usuario: decida si detener el proceso o volvemos al paso 2.

* Necesitamos un criterio de división y un criterio de stop.

### ¿Cuántas divisiones podemos hacer en un nodo?

Las divisiones se definen por el número y el tipo de las variables explicativas y el tipo del árbol

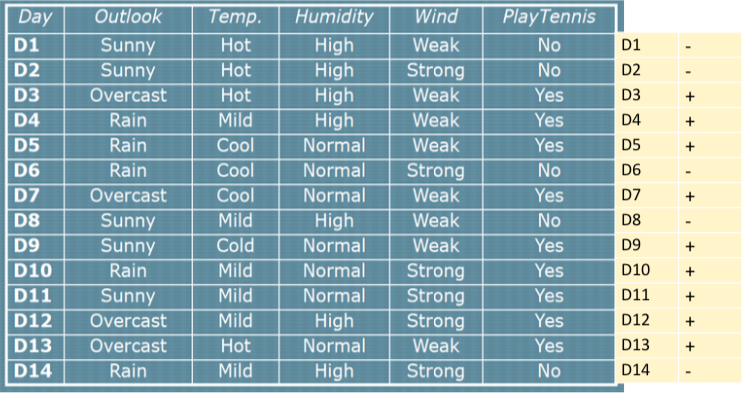
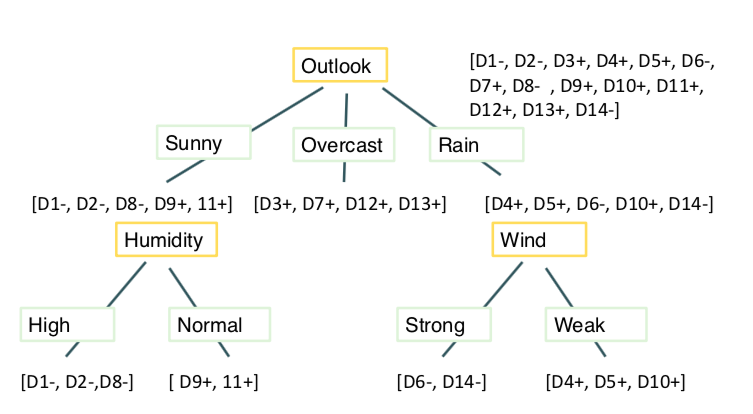


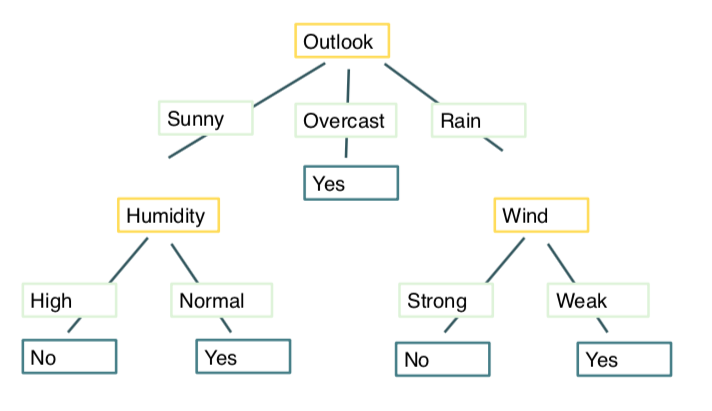
### Algoritmo ID3

* Inventado por Ross Quinlan (1986)
* Categorías o variables discretas.
* El concepto a aprender puede ser descrito por reglas lógicas
* Ejemplos:
* diagnostic médico
* Análisis de riesgo en crédito.
* Clasificación de objetos para su manipulación.
* *Esquema del algoritmo:*

1. Establecer el nodo en los ejemplos de entrenamiento y ponerlo como la raiz del árbol.
2. A: las “mejores” variables de prueba para el nodo actual.
3. Para cada valor de la variable A, cree un Nuevo nodo descendente en el árbol de decisión.
4. Ejemplos separados en nodo entre descendientes dependiendo del valor de la variable para A.
5. Si quedan nodos de hojas en el árbol con ejemplos mixtos positivos y negativos, configure esta hoja en nodo y vuelva al paso 2.
6. En otro caso, etiquete cada hoja del árbol con el signo de los ejemplos en la hoja y termine.

Veremos un ejemplo de este algoritmo en “playing tennis” 🡪



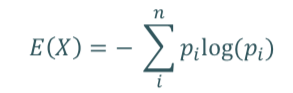


Como escoger las variables?

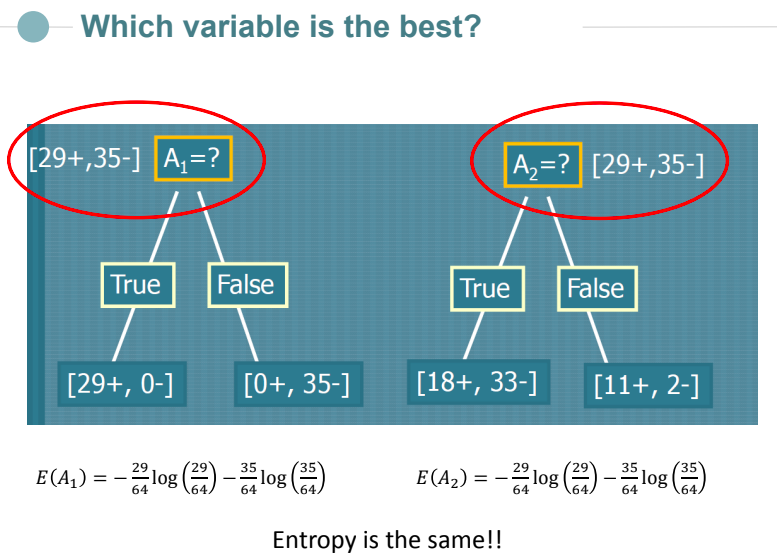
* Objetivo : árbol más corto.
* Equivalente a separar lo más rápido posible los ejemplos positivos (+) de los negativos (-).
* Criterio de división: máximo discriminativo.
* Queremos escoger las variables que sean más discriminativas.
* Ganancia de información.

Teoría de la información, entropía

* Una respuesta nos da una cantidad de información proporcional a la imprevisibilidad de la respuesta

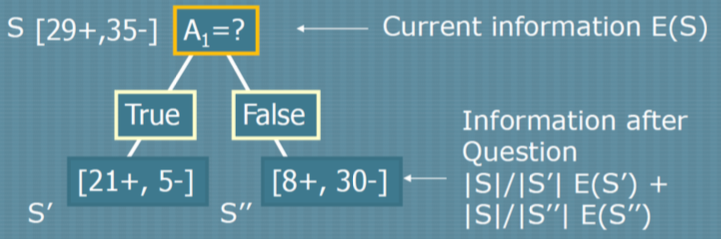


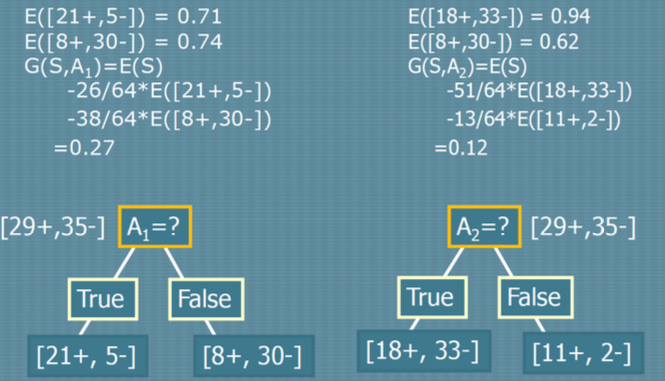
* Donde pi es la probabilidad de la respuesta i, (denota la proporción de instancias que pertenecen a la clase i)
* n es el número de respuestas posibles.
* E(X) – la entropía de X – mide los bits necesarios para comprimir la información sobre la distribución actual + y - ejemplos en X considerándola una variable aleatoria. Es una medida del contenido de información tomada del campo Teoría de la información.
* Medida del desorden… por eso es útil en ID3.
* **Propiedades de la entropía:**
*  
* Maximizado cuando los elementos son heterogéneos (impuros, desordenados):
* Si entonces Entropía =
* Minimizado cuando los elementos son homogeneous (puros, ordenados):
* Si o , entonces Entropía = 0.

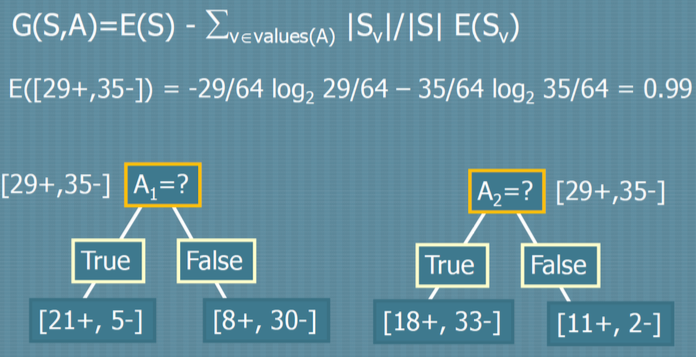


Ganancia de información

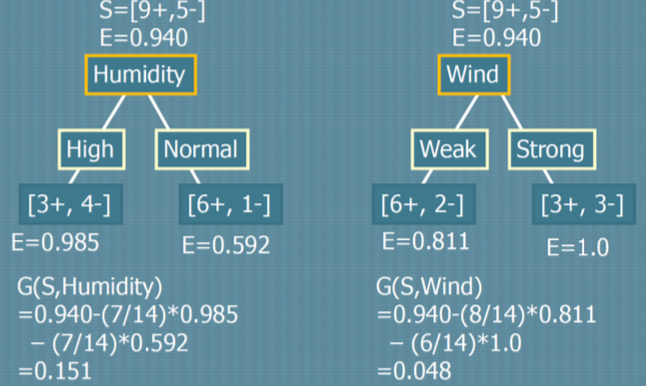
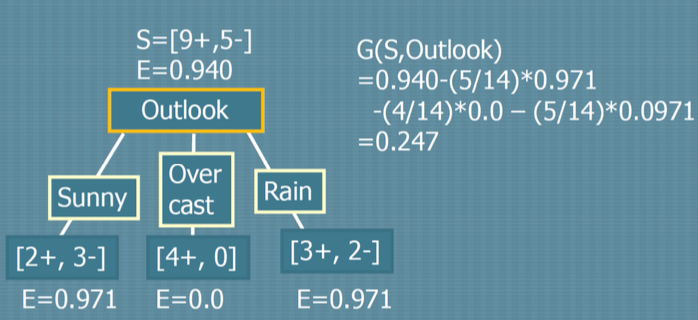
La ganancia de información (IG) obtenida de una respuesta es la diferencia de información antes y después de la respuesta.

****

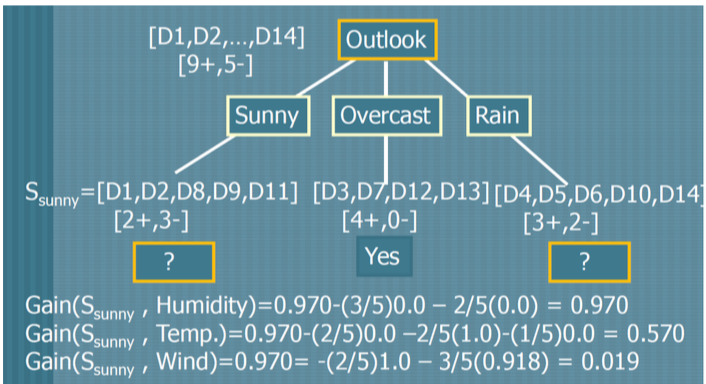
****



* La mejor variable es la que tiene una ganancia de información mayor.
* La ganancia de información mide la cantidad de “información” que nos da una variable sobre la clase.
* Las características que participan perfectamente deben dar información máxima.
* Las características no relacionadas no deben dar información.
* Mide la reducción de la entropía.
* Entropía: (im)pura en una colección arbitraria de ejemplos.



* Ganancia de información para cada función:
* Outlook = 0.247
* Temperature = 0.029
* Humidity = 0.152
* Windy = 0.048
* La división inicial está en Outlook, porque es la característica con la mayor ganancia de información.



Algoritmo ID3

* Para cada nodo, seleccione la mejor variable (criterio de división).
* Utilice todas las variables que no sean la actual o una variable en un nodo principal.
* Esto se hace en todos las ramas hasta que todos los ejemplos en el nodo sean de la misma clase.
* Seleccionando la mejor variable:
* Selecciona la mejor variable usando la entropía y ganancia de información.
* Mejor = Atributo con IG más alto.
* **Posibles problemas:**
* Qué pasa si no hay ejemplos para un valor al expandir un nodo?
* No generar rama
* Al probar, dirígete a la sucursal más popular.
* Cuando no podemos expandir el nodo pero aun hay + y – ejemplos mezclados en el nodo.
* Etiqueta el nodo con la clase de la mayoría.
* Sobreajuste.
* **Problema de Sobreajuste:**
* El algoritmo anterior puede resultar en un árbol con muchos niveles y divisiones.
* Muchas de las hojas pueden contener un número reducido de ejemplos.
* El árbol no es lo suficientemente general y casi hay una regla/ ruta para cada ejemplo.
* Overfitting.
* **Como evitar el sobreajuste**
* No dividir los ejemplos cuando la ganancia de información supere un umbral (límite previo).
* Podar algunas ramas del árbol (post-poda).
* Post-poda en el árbol.
* Post-poda de reglas (C4.5).

### Algoritmo C4.5

* C4.5 desarrollado por Ross Quinlan.
* C4.5 es una extensión del algoritmo ID3 anterior de Quinlan.
* El mismo conjunto de datos que ID3
* Variables categóricas.
* Criterio de división: usar la información de ganancia normalizada (relación de ganancia de información) en su lugar.
* Poda del árbol después de la creación.

Ganancia de información

* La relación de ganancia de información (IGR) es una solución al efecto no deseado: sesgo hacia variables con gran número de valores diferentes.
* Tendencia de sobrealimentación.
* IGR es la normalización de IG por el número de categorías.

(Entropía de la variable divisoria).

* La relación de ganancia de información es una relación entre la ganancia de información y la información intrínseca.
* Propuesto por Ross Quinlan, para reducir el sesgo hacia los atributos de valores múltiples, teniendo en cuenta el número y el tamaño de las sucursales al elegir un atributo.
* La ganancia depende del número de modalidades de la variable de división, por lo que los resultados dependen en gran medida de la codificación real de las variables.
* Remedio. Relativizar respeta la información intrínseca (IV) de la variable de división (que depende de su número de modalidades):

(Entropía de la variable divisoria).

### C4.5

* Criterio de división: asignación de cada nodo a la modalidad de respuesta con la máxima probabilidad.
* Criterio de parada: se detiene cuando la ganancia relativa está por debajo de un cierto umbral.
* Después de la poda de las ramas, si la probabilidad de error de clasificación en los nodos descendentes no mejora la probabilidad de error de clasificación en el nodo actual de manera significativa.

Post Poda de las ramas

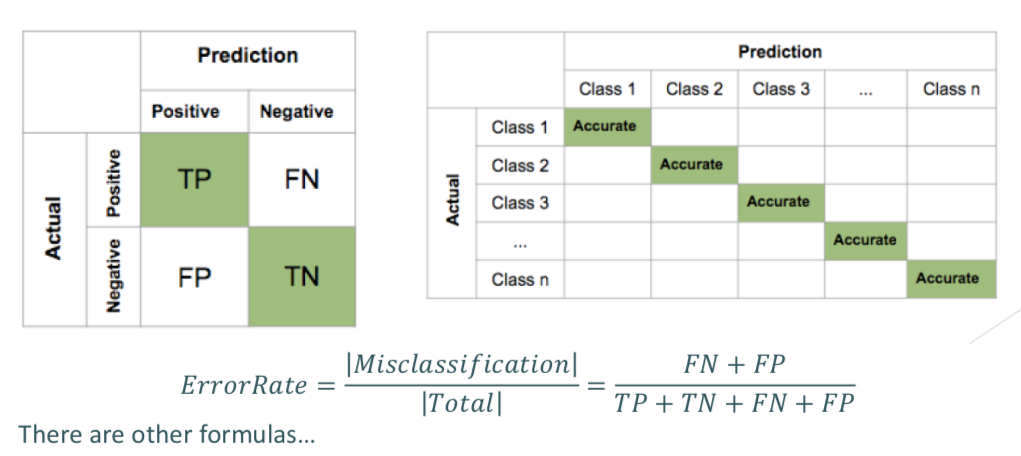
* Datos dividos en subgrupos de entrenamiento y validación.
* Entrenamiento y validación: para obtener un árbol confiable, necesitamos probarlo con datos independientes del que se uso en el paso de aprendizaje.
* Dividir los datos totales al azar en una parte para el aprendizaje (capacitación) y el resto para la prueba (validación)
  + Porcentaje común: 2/3 de entrenamiento y 1/3 de validación.
* Datos divididos en subgrupos de entrenamiento y validación.
* Repita hasta que no haya mejora en el error de subconjunto de validación.
* Para cada nodo, calcular el error en el conjunto de validación al eliminar el nodo (y, por lo tanto, el subárbol debajo del nodo).
* Realice la poda que reduzca al máximo el error en el conjunto de validación.

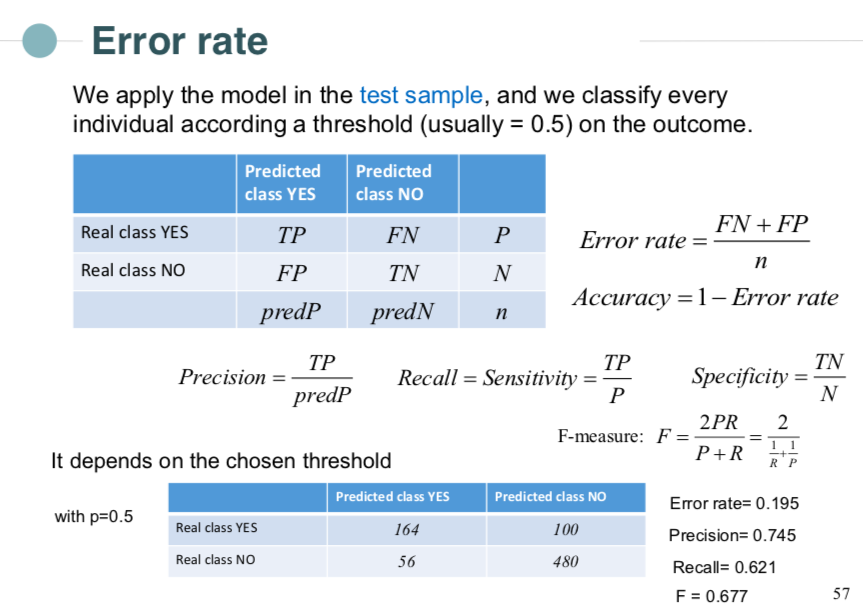
Probabilidad de error de clasificación

* Necesitamos definir qué tan bueno (o malo) es un árbol.
* Criterio obvio = probabilidad de error de clasificación (= coste del árbol).
* Como podemos calcular el coste del árbol?
* Asignamos cada permiso a la clase de respuesta.
* Si hay ejemplos de clases diferentes, tomamos la clase con el número máximo de ejemplos (modo).

Tasa de clasificación errónea

* Matriz de confusión para caracterizar errores de clasificación.
* Matriz de confusión = visualización de resultados predichos frente a resultados reales.
* Bueno si: valores altos a lo largo de la diagonal, valores bajos en otros lugares.



Error rate 🡪 tasa de error.

Aplicamos el modelo en la muestra de prueba y clasificamos a cada individuo de acuerdo con un umbral (generalmente = 0.5) en el resultado.

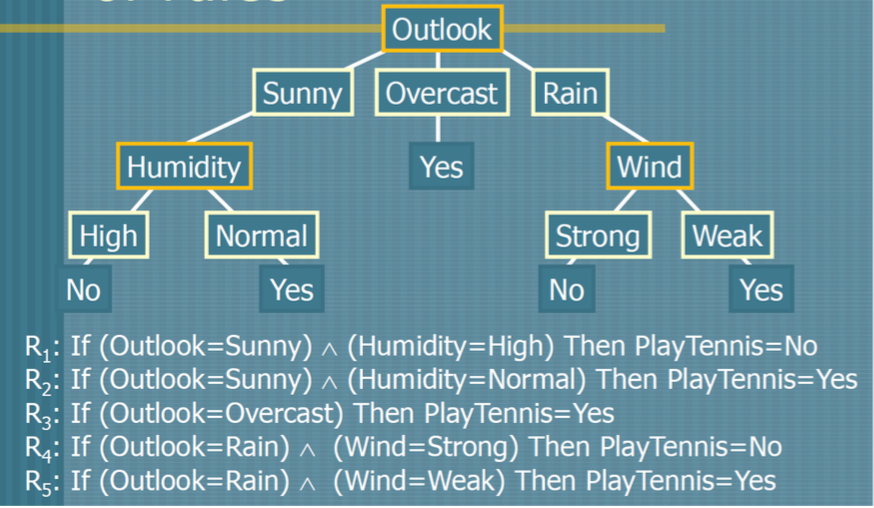
Post poda de las ramas

* Datos divididos en subgrupos de entrenamiento y validación.
* Repetir hasta que no haya mejora en el error de subconjunto de validación.
  1. Para cada nodo, calcular el error en el conjunto de validación al eliminar el nodo (y, por lo tanto, el subárbol debajo del nodo).
  2. Realizar la poda que disminuya al máximo el error en la validación.

Post poda de reglas (C4.5)

* Algoritmo C4.5

1. Construye el árbol de decisión con ID3.
2. Transformar el árbol en reglas.
3. Pode cada regla independientemente eliminando las condiciones que disminuyen el error en el conjunto de validación.
4. Ordenar el conjunto final de reglas para resolver conflictos.
5. Eliminar reglas cuando disminuye el error en el conjunto de validación.



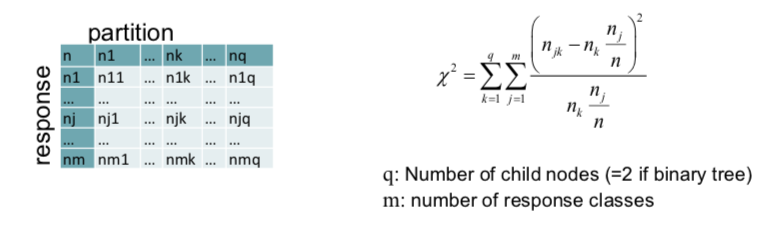
* Traducir el árbol en el conjunto de reglas

### CHAID (Kass, 1980)

CHAID (detección de interacción automática Chi-cuadrado).

**Criterio de división:** basado en la estadística de Chi cuadrado calculada cruzando la variable de respuesta con la partición tentativa definida por una variable explicativa.

El más significativo es el Chi-cuadrado, más diferente es la distribución de la respuesta en el nodo padre con respecto a los nodos hijos.

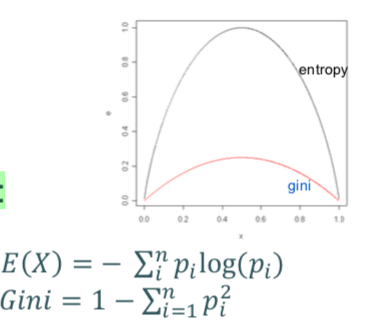


La variable (categorica) que da la X2 más significativa define la división óptima.

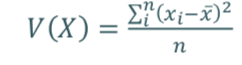
### CART

* Desarrollado 1974-1984 por 4 profesores de estadística : Leo Breiman (Berkeley), Jerry Friedman (Stanford), Charles Stone (Berkeley), Richard Olshen (Stanford). Distribuido por Salford Systems <http://salford-systems.com/>.
* Solo realiza árboles binarios.
* Unifica la respuesta categórica y continua (numérica) bajo el mismo marco.
* Árbol de clasificación.
* Árbol de regresión.
* Cualquier tipo de variable explicativa (numérica y categórica).
* Criterio de división: impureza del nodo.
* Post poda (sin criterio de parada).
* Ofrece estimaciones honestas de la calidad de un árbol.

Criterio de división: impureza de un nodo



* Para respuestas categóricas:
* Entropía (Información).
* Gini.
* Para respuestas continuas:
* Varianza.



Propiedad del índice de Gini

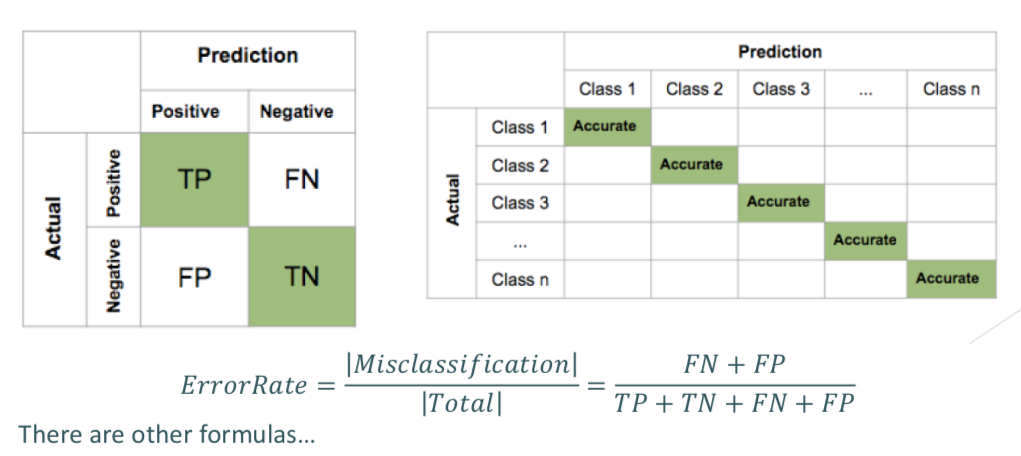
* Maximizar cuando los elementos sean heterogéneos (impuro).
* Si , entonces Gini
* Minimizar cuando los elementos sean homogéneos (puros).
* Si o , entonces Gini = 1-1-0 = 0.

### CART

* Algoritmo codicioso para minimizar una función de coste.
* (No versión original) **Criterio de parada:**
* Fijar un numero mínimo de ejemplo por nodo. Si el nodo tiene menos de este mínimo, entonces el nodo es una hoja.
* Si mínimo = 1, entonces es probable que sea demasiado especifico y excesivo? (o sobre ajustado, no lo sé).
* Post-poda : encuentra nodos que sean removibles.
* Simple: el coste del árbol mejora cuando se elimina el nodo
* Otros: penalización por complejidad.
* Criterio de parada: No hay criterio de parada. CART utiliza la poda posterior, construye un árbol máximo y poda las ramas no interesantes
* Árbol máximo absoluto: árbol con todas las hojas puras.

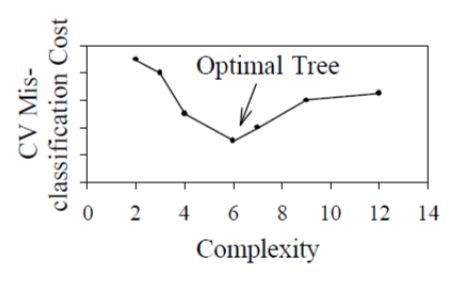
Tasa de clasificación errónea

* Matriz de confusión para caracterizar errores de clasificación.
* Matriz de confusión = visualización de resultados predichos frente a resultados reales.
* Bueno si: valores altos a lo largo de la diagonal, valores bajos en otros lugares.

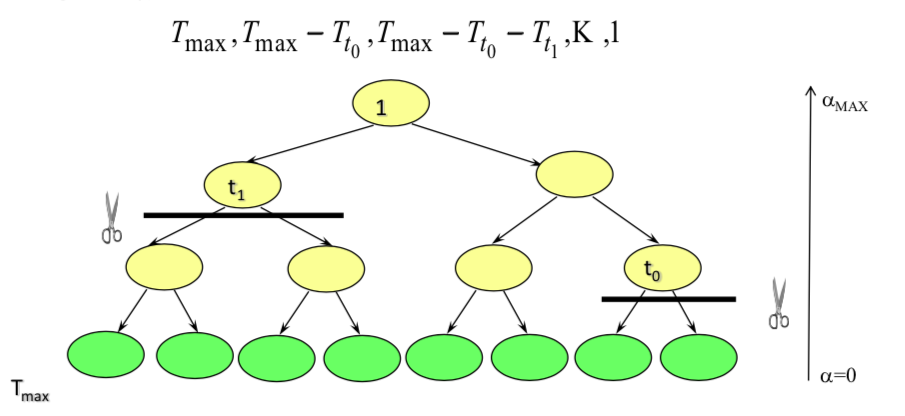


Penalización por complejidad

Criterio para optimizar 🡪 Min ( R(T) + |T| ) , siendo R(T) = coste del árbol.

Donde es el parámetro de complejidad, expresa la penalización de construir árboles grandes.

Podemos construir los árboles para valores crecientes α, obteniendo una secuencia de árboles óptimos (de tamaño creciente) (CART realiza esta poda del árbol máximo de una manera más inteligente).



Cada subárbol es óptimo (min R (T)), dentro de los subárboles de complejidad ITІ

### Ventajas de los árboles de decisión

* Fácil de entender e interpretar.
* Los árboles también se pueden mostrar gráficamente de una manera que sea fácil de interpretar para los no expertos.
* Capaz de manejar datos numéricos y categóricos.
* Otras técnicas suelen estar especializadas en analizar conjuntos de datos que tienen un solo tipo de variable.
* Las reglas de asociación solo pueden usarse con variables categóricas.
* Las redes neuronales se pueden usar solo con variables numéricas o categóricas convertidas a valores de 0-1.
* Requiere poca preparación de datos.
* No hay normalización de datos.
* No es necesario crear variables ficticias.
* Utiliza un modelo de caja blanca.
* Si una situación dada es observable en un modelo, la explicación de la condición se explica fácilmente mediante la lógica booleana.
* Por el contrario, en un modelo de caja negra, la explicación de los resultados suele ser difícil de entender, por ejemplo, con una red neuronal artificial.
* Posible validar un modelo utilizando pruebas estadísticas.
* Posible tener en cuenta la fiabilidad del modelo.
* Enfoque no estadístico que no hace suposiciones de los datos de entrenamiento o los residuos de predicción; por ejemplo, no hay supuestos de distribución, independencia o variación constante.
* Realiza bien con grandes conjuntos de datos.
* Se pueden analizar grandes cantidades de datos utilizando recursos informáticos estándar en un tiempo razonable.

### Limitaciones de los árboles de decisión

* Los arboles pueden ser muy poco robustos.
* Un pequeño cambio en los datos de entrenamiento puede resultar un gran cambio en el árbol y consecuentemente en las predicciones finales.
* Se sabe que el aprendizaje de un árbol de decisiones óptimo está completo en NP bajo varios aspectos de optimalidad e incluso para conceptos simples.
* En consecuencia, la práctica de la decisión de aprender a utilizar el cerebro se basa en las heurísticas, como el algoritmo codicioso, en el que se toman decisiones óptimas a nivel local en cada nodo. Tales algoritmos no pueden garantizar devolver el árbol de decisión óptimo a nivel mundial.
* Los aprendices del árbol de decisión pueden crear árboles demasiado complejos que no se generalizan bien a partir de los datos de capacitación. (Esto se conoce como sobreajuste)
* Mecanismos tales como la eliminación son necesarios para evitar el problema
* Para los datos que incluyen variables categóricas con diferentes números de niveles, la ganancia de información en los árboles de decisión está sesgada a favor de los atributos con más niveles.

### CART – LIBRARY RPART

