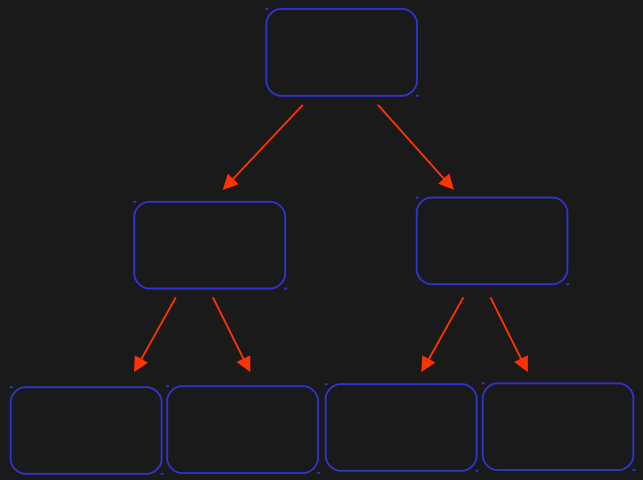
Árboles de Decisión

Alain Raymond

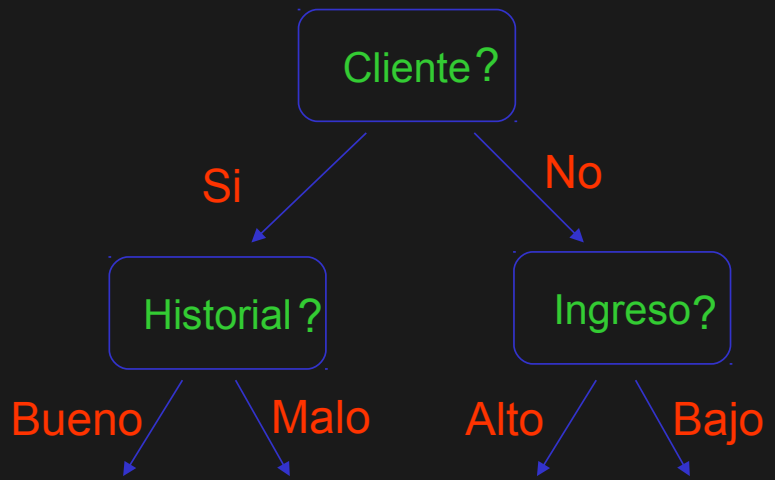
Computer Science Department, PUC

- Una técnica de clasificación.
- Como su nombre lo dice, consiste en un árbol.



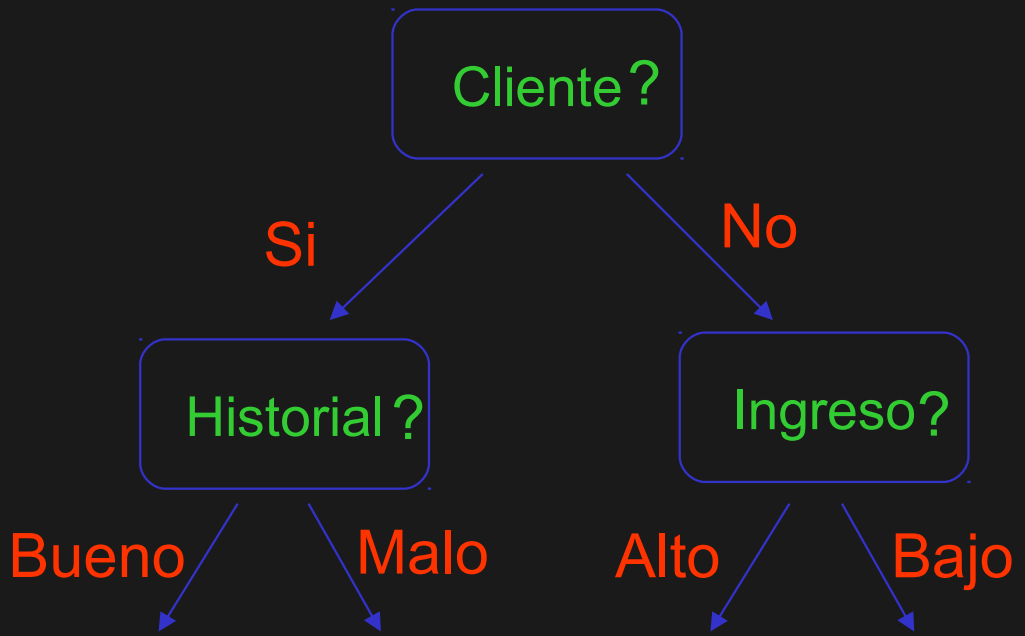
- Por tanto contiene nodos y uniones dirigidas entre los nodos (links).
- Lo anterior permite una fácil visualización del clasificador.

- Cada nodo interno representa un atributo.
- En cada nodo interno se realiza un test basado en los valores del atributo.
- Los links representan el resultado del test.



- Los nodos hoja representan el resultado de la clasificación.
- Así, para clasificar un registro se debe recorrer el árbol desde el nodo raíz a la hoja resultante.
- El camino recorrido dependerá de los valores del registro.
- Ejemplo: ¿Cuál es la clasificación para el crédito de un cliente con Cliente==Si e Historial=Bueno e Ingreso=Alto?

- Un detalle importante es que el árbol sólo puede implementar decisiones discretas (categóricas), ¿Por qué?
- Por ende, en el caso de atributos continuos será necesario discretizarlos o utilizar un criterio discreto de decisión, ¿cómo?.



- Los árboles de decisión son una técnica de *Aprendizaje Supervisado*.
- Por tanto, su entrenamiento necesita de un set de registros rotulados.
- El set de entrenamiento permite ajustar el modelo, i.e., encontrar una estructura apropiada para el árbol, i.e., explorar el espacio de hipótesis buscando un buen clasificador.
- ¿Cuál es el espacio de hipótesis de los árboles de decisión?

Espacio de Hipótesis

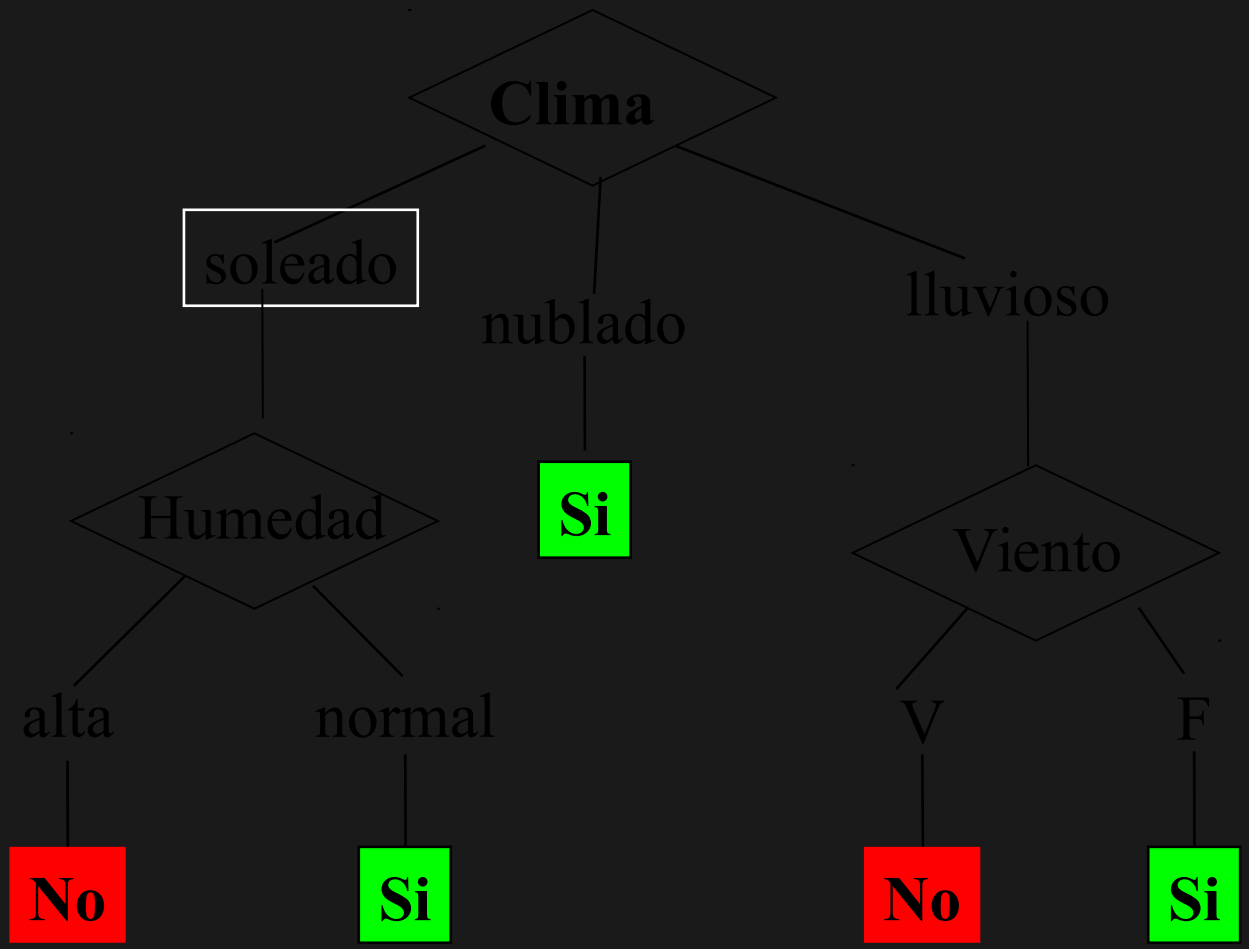
- ¿Cuál es el espacio de hipótesis de los árboles de decisión?
  - El espacio de hipótesis corresponde a las posibles disyunciones de conjunciones de los atributos disponibles.

$$[(Cliente == Si \wedge Historial == Bueno) \implies Clase = Aprobado] \vee$$
$$[(Cliente == Si \wedge Historial == Malo) \implies Clase = Rechazado] \vee$$
$$[(Cliente == No \wedge Ingreso == Alto) \implies Clase = Aprobado] \vee$$
$$[(Cliente == No \wedge Ingreso == Bajo) \implies Clase = Rechazado]$$

Ej. Jugar Tenis

Clima	Temperatura	Humedad	Viento	Jugar
soleado	alta	alta	F	No
soleado	alta	alta	V	No
nublado	alta	alta	F	Si
lluvioso	Agradable	alta	F	Si
lluvioso	frio	alta	F	Si
lluvioso	frio	alta	V	No
nublado	frio	alta	V	Si
soleado	Agradable	alta	F	No
soleado	frio	alta	F	Si
lluvioso	Agradable	alta	F	Si
soleado	Agradable	alta	V	Si
nublado	Agradable	alta	V	Si
nublado	alta	alta	F	Si
lluvioso	Agradable	alta	V	No

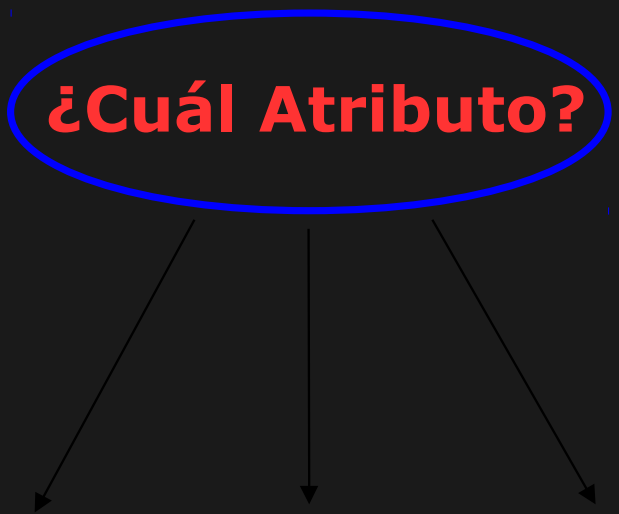
Ej. Jugar Tenis



¿Cómo determinar el árbol?

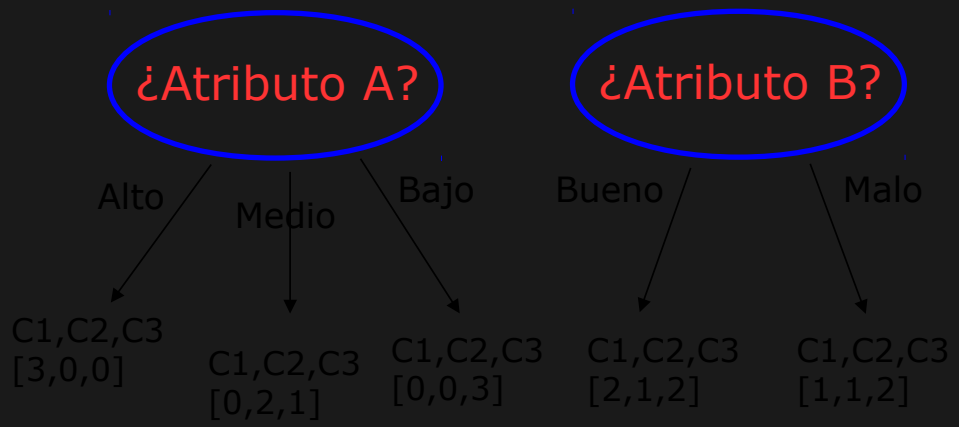
- La construcción de un árbol de decisión es incremental partiendo por el nodo raíz.
- Por tanto, el desafío inicial es decidir qué atributo utilizar para el nodo raíz, alguna idea?:

Clima	Temperatura	Humedad	Viento	Jugar?
soleado	alta	alta	F	No
soleado	alta	alta	V	No
nublado	alta	alta	F	Si
lluvioso	Agradable	alta	F	Si
lluvioso	frio	normal	F	Si
lluvioso	frio	normal	V	No
nublado	frio	normal	V	Si
soleado	Agradable	alta	F	No
soleado	frio	normal	F	Si
lluvioso	Agradable	normal	F	Si
soleado	Agradable	normal	V	Si
nublado	Agradable	alta	V	Si
nublado	alta	normal	F	Si
lluvioso	Agradable	alta	V	No



¿Cuál atributo?

Atributo A	Atributo B	Clase
Alto	Bueno	C1
Alto	Malo	C1
Bajo	Malo	C3
Medio	Malo	C2
Alto	Bueno	C1
Bajo	Malo	C3
Bajo	Bueno	C3
Medio	Bueno	C3
Medio	Bueno	C2

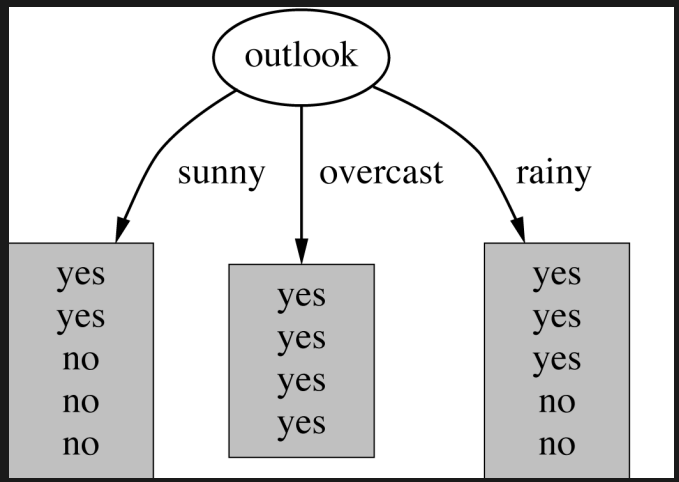


El algoritmo de construcción selecciona el atributo que mejor separa los registros de acuerdo al valor de las clases, i.e., el atributo más discriminativo.

¿Cómo determinar el árbol?

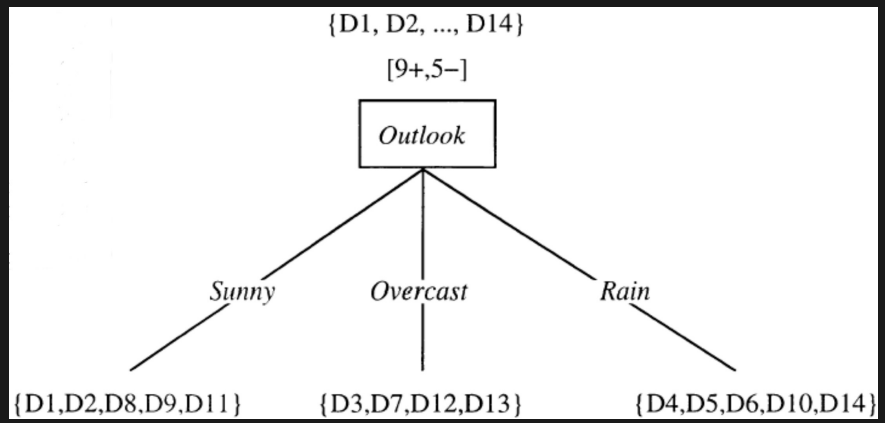
- Más adelante veremos un criterio matemático para determinar el atributo más discriminativo.
- Por ahora, asumamos que para el set de datos de la figura, el atributo más discriminativo es: *'outlook'*.

Day	Outlook	Temperature	Humidity	Wind	PlayTenn
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



¿Cómo determinar el árbol?

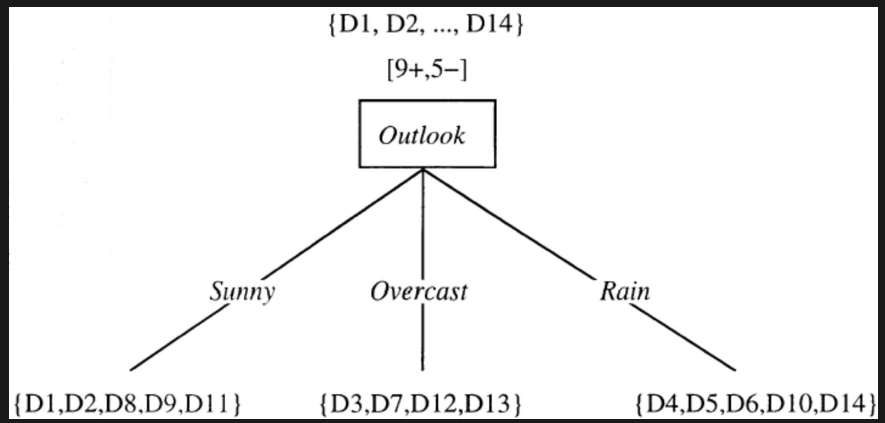
Day	Outlook	Temperature	Humidity	Wind	PlayTenn
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



- Al analizar las ramas resultantes es posible apreciar que cada una de ellas recibe un subconjunto de los datos originales.
- ¿Cómo podemos seguir construyendo el árbol?

¿Cómo determinar el árbol?

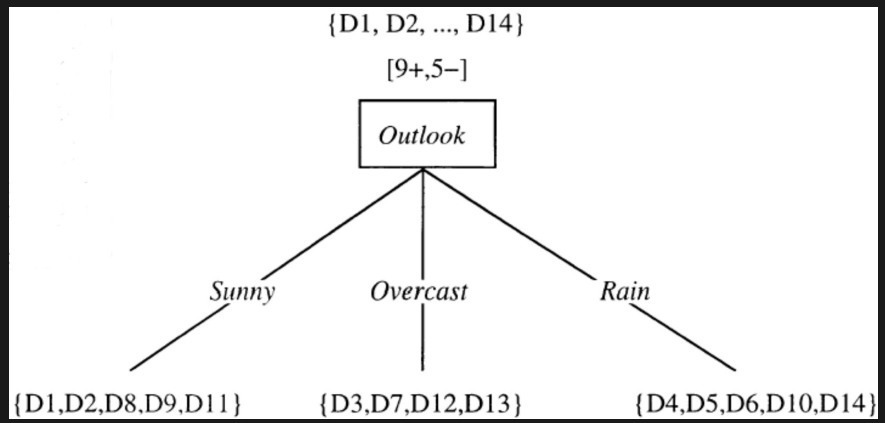
Day	Outlook	Temperature	Humidity	Wind	PlayTenn
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



- Al analizar las ramas resultantes es posible apreciar que cada una de ellas recibe un subconjunto de los datos originales.
- ¿Cómo podemos seguir construyendo el árbol?
  - Volver a identificar el atributo más discriminativo para cada uno de los subconjuntos formados

¿Cómo determinar el árbol?

Day	Outlook	Temperature	Humidity	Wind	PlayTenn
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



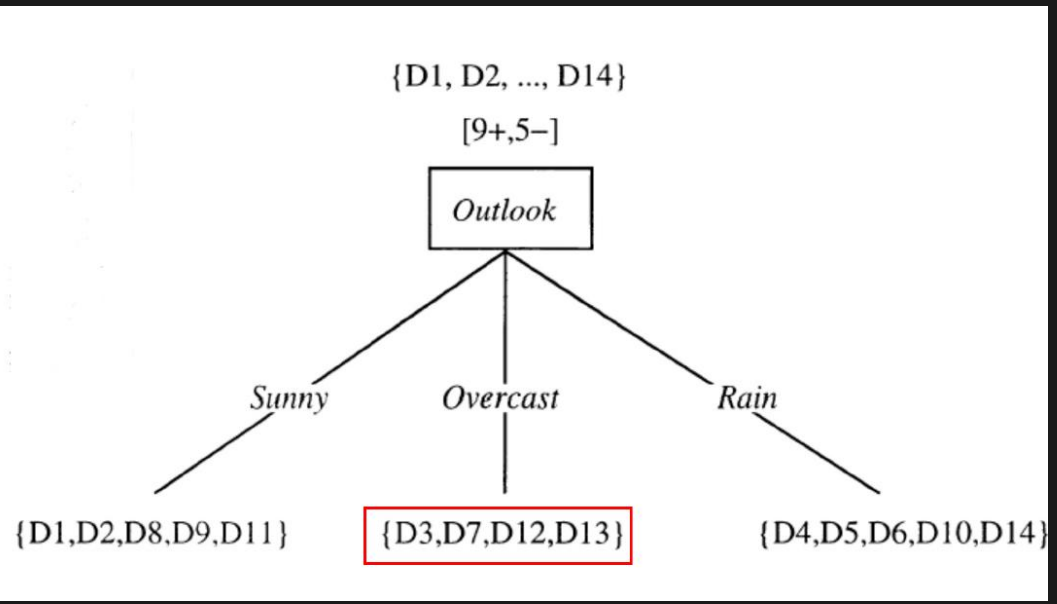
- Al analizar las ramas resultantes es posible apreciar que cada una de ellas recibe un subconjunto de los datos originales.
- ¿Cómo podemos seguir construyendo el árbol?
  - Volver a identificar el atributo más discriminativo para cada uno de los subconjuntos formados
  - ¿Cuándo para de iterar?

Árbol de Decisión: Algoritmo de Construcción

1. De los atributos disponibles, seleccionar el que “mejor” separa los registros de las distintas clases.
2. Usar ese atributo como nodo raíz y dividir el set de entrenamiento de acuerdo a este atributo.
3. Para cada subgrupo resultante:
   - **IF(** pertenecen todos los registros a la misma clase **)**:
     - Retornar marcando el nodo hoja con la clase respectiva.
   - **IF(** tienen todos los registros el mismo valor para todos los atributos que determinan su clase **)**:
     - Retornar marcando nodo hoja con la clase más común.
4. Para cada rama sin nodo hoja, volver al paso 1 considerando sólo los datos en la rama.

Árbol de Decisión: Algoritmo de Construcción

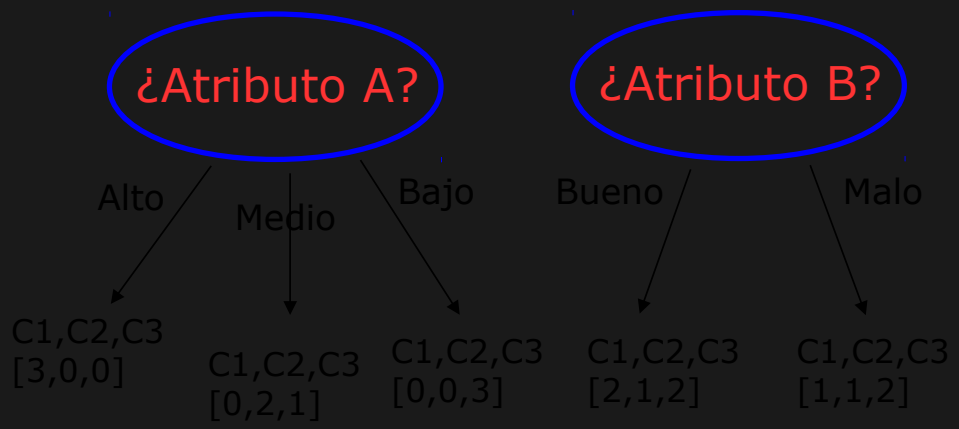
Day	Outlook	Temperature	Humidity	Wind	PlayTenn
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



¿Cómo determinar el atributo más discriminativo?

¿Cuál atributo?

Atributo A	Atributo B	Clase
Alto	Bueno	C1
Alto	Malo	C1
Bajo	Malo	C3
Medio	Malo	C2
Alto	Bueno	C1
Bajo	Malo	C3
Bajo	Bueno	C3
Medio	Bueno	C3
Medio	Bueno	C2



- **Objetivo:** obtener subgrupos homogéneos respecto a la clase.
- **Necesitamos:** una métrica de homogeneidad de cada subgrupo.
- Dos métricas usuales:
  - Entropía $\rightarrow$ *¡Nos enfocaremos en ésta!*
  - Gini Impurity

Métrica de homogeneidad: Entropía

- **Entropía** es una buena manera de medir homogeneidad.
- En teoría de la información, entropía H(S) mide el número de bits promedio que se necesita para codificar en forma óptima un conjunto de datos S.

$$H(S) = -\sum_{c_i} p_i \ log_2 \ p_i$$

$H(S)$: entropía del set S.

c_i : set de clases posibles.

p_i : fracción de registros en S con clase c_i.

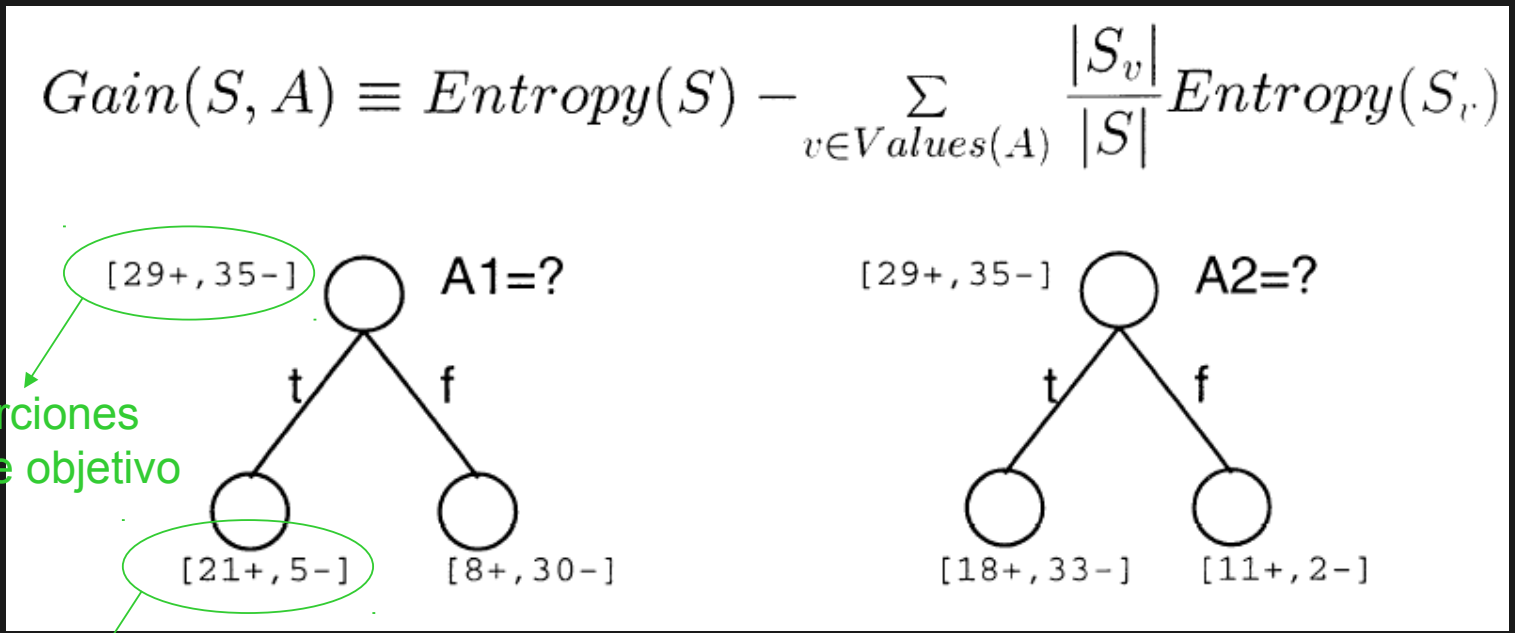
Intuición: Mientras más incerteza existe sobre el posible valor de un símbolo $S = s_i$ mayor es la entropía. Por tanto, si el grupo S es totalmente homogéneo, i.e., todos los elementos de S son iguales, la entropía es mínima. Revisar: `https://www.youtube.com/watch?v=ErfnhcEV1O8`

- Ejemplo 1:
  - 10 registros con clase A
  - 20 registros con clase B
  - 30 registros con clase C
  - 40 registros con clase D
  - Entropía= -[(.1 log .1) + (.2 log .2) + (.3 log .3) + (.4 log .4)]
  - Entropía = 1.85
- Ejemplo 2:
  - 100 registros con clase A
  - 0 registros con clase B
  - 0 registros con clase C
  - 0 registros con clase D
  - Entropía= -[(1 log 1)]
  - Entropía = 0

Ganancia de información

- En la construcción de un árbol de decisión, cada atributo divide los datos en varios grupos, por ende, debemos considerar la entropía de todos los sub-grupos resultantes.
- La ganancia de información $Gain(S, A)$ es la reducción esperada en entropía al separar el set original S según cierto atributo A:

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

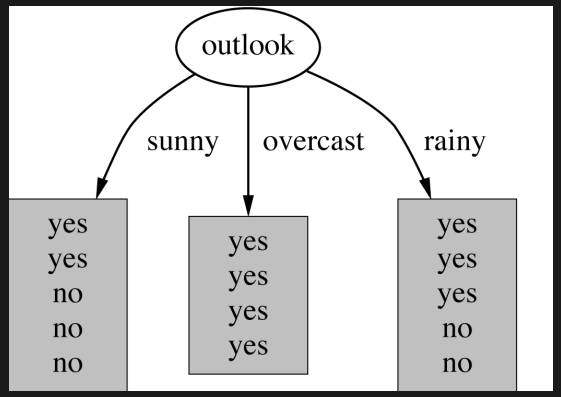


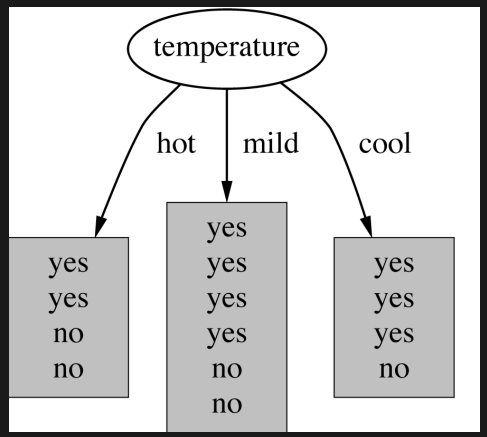
Ganancia de información

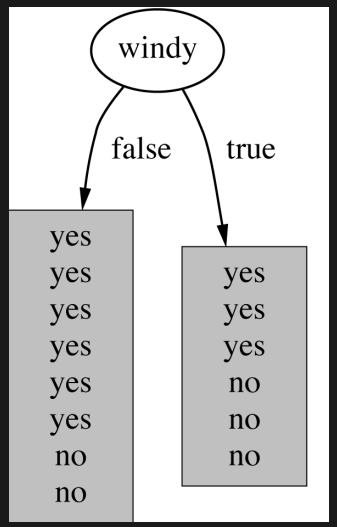
Clima	Temperatura	Humedad	Viento	Jugar
soleado	alta	alta	F	No
soleado	alta	alta	V	No
nublado	alta	alta	F	Si
lluvioso	Agradable	alta	F	Si
lluvioso	frio	alta	F	Si
lluvioso	frio	alta	V	No
nublado	frio	alta	V	Si
soleado	Agradable	alta	F	No
soleado	frio	alta	F	Si
lluvioso	Agradable	alta	F	Si
soleado	Agradable	alta	V	Si
nublado	Agradable	alta	V	Si
nublado	alta	alta	F	Si
lluvioso	Agradable	alta	V	No

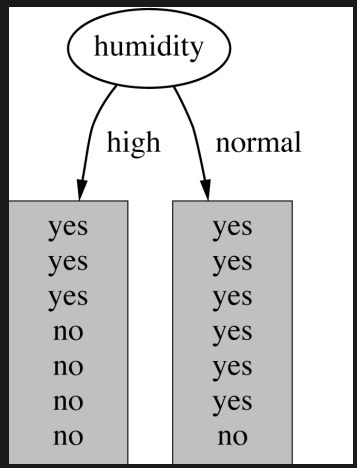
¿Qué atributo seleccionar?

Day	Outlook	Temperature	Humidity	Wind	PlayTenn
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

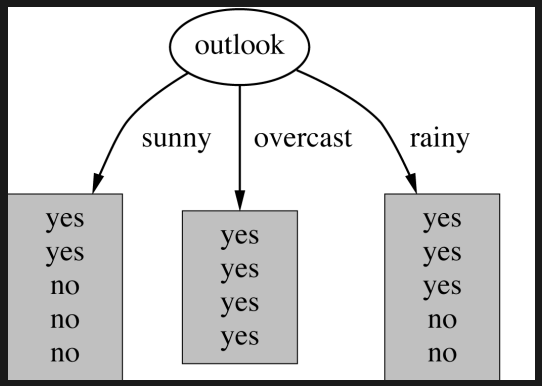








¿Qué atributo seleccionar?



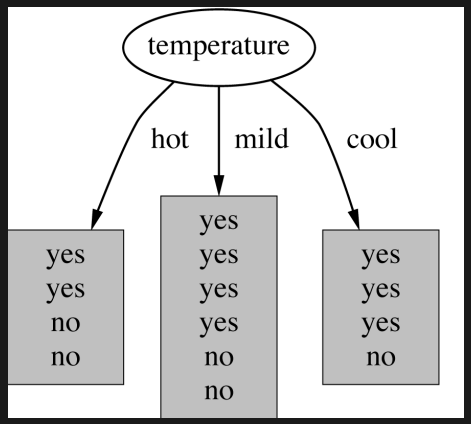
S:[9+,5-]
E=0.940

Outlook

Sunny **Overc.** **Rainy**

[2+,3-] E=0.971 | [4+,0-] E=0 | [3+,2-] E=0.971

Gain(S, Outlook)=0.940-(5/14)0.971-0-(5/14)0.971 =0.266



S:[9+,5-]
E=0.940

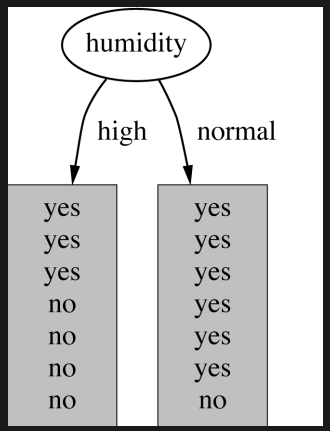
Temperat.

Hot **Mild** **Cool**

[2+,2-] E=1 | [4+,2-] E=0.918 | [3+,1-] E=0.811

Gain(S, Temp.)=0.940-(4/14)-(6/14)0.918 - (4/14)0.811 =0.029

¿Qué atributo seleccionar?



S:[9+,5-]
E=0.940

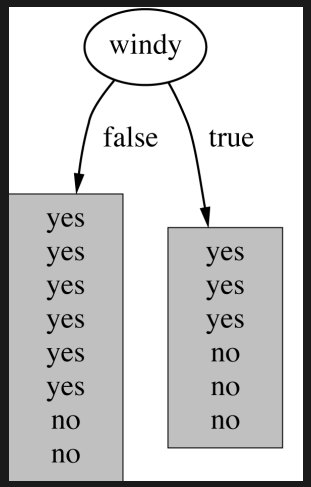
Humidity

High **Normal**

[3+,4-]
E=0.985

[6+,1-]
E=0.592

Gain(S, Humidity)=0.940-(7/14)0.985-(7/14)0.592
=0.151



S:[9+,5-]
E=0.940

Windy

False **True**

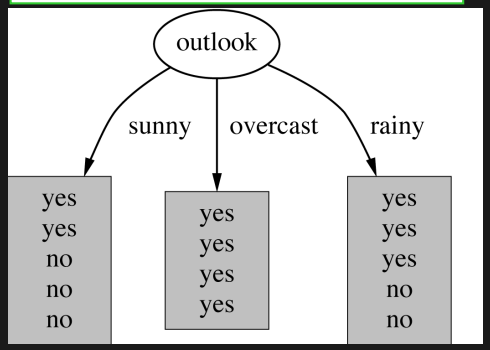
[6+,2-]
E=0.811

[3+,3-]
E=1

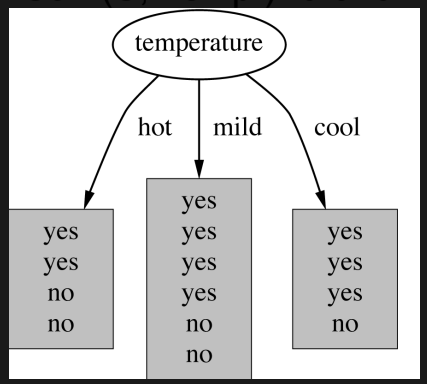
Gain(S, Windy)=0.940-(8/14)0.985-(6/14)
=0.048

¿Qué atributo seleccionar?

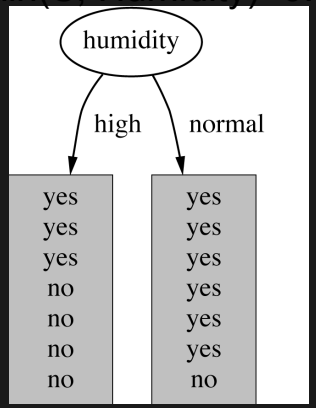
Gain(S, Outlook)=0.266



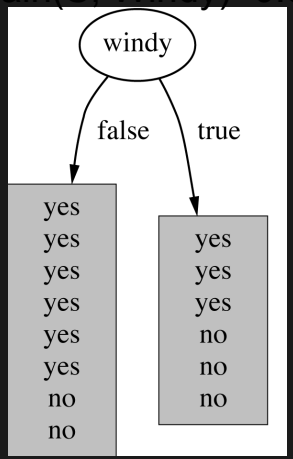
Gain(S, Temp.)=0.029



Gain(S, Humidity)=0.151

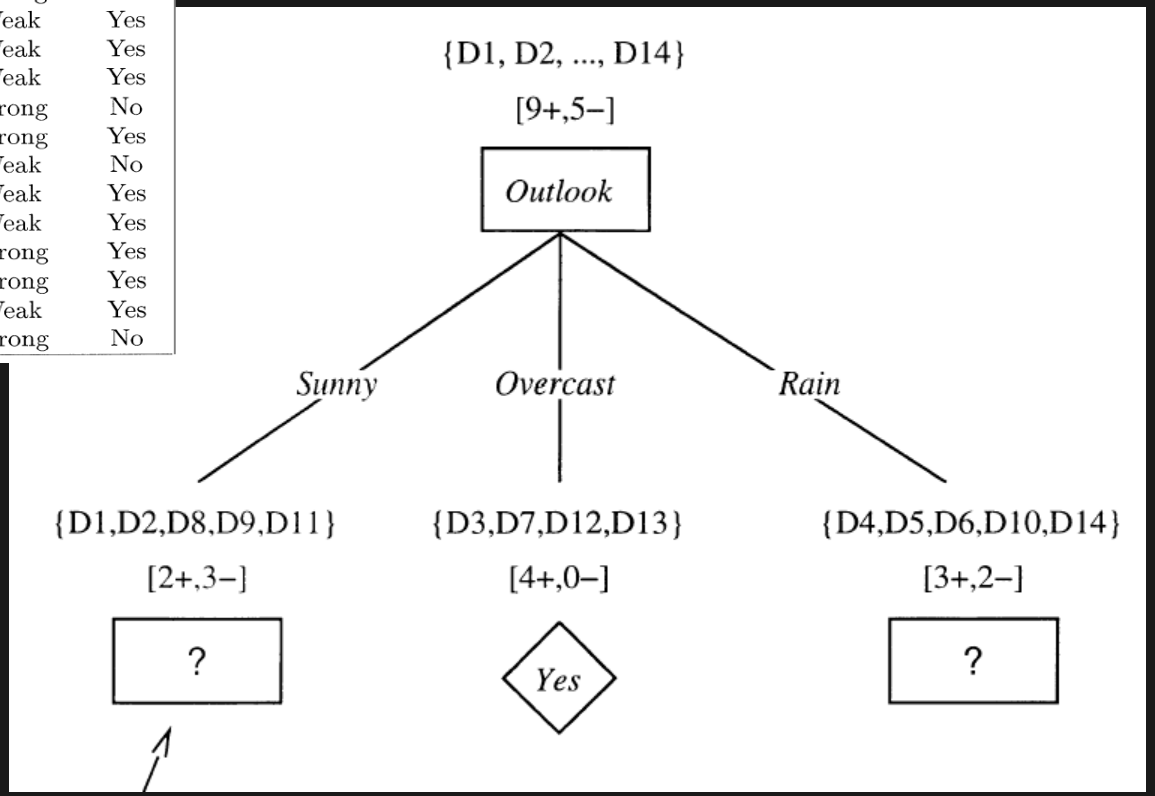


Gain(S, Windy)=0.048



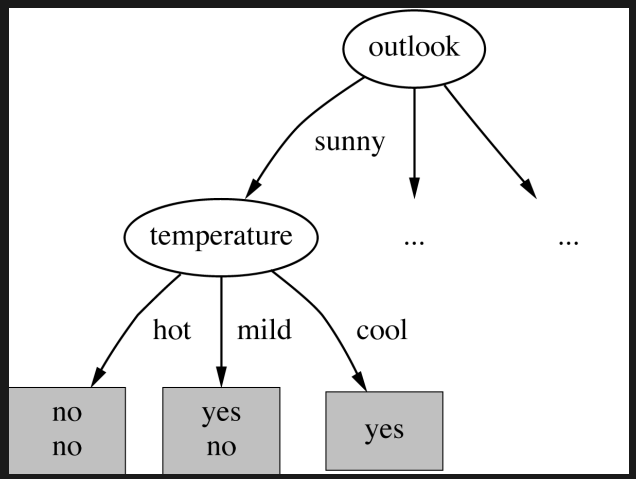
¿Qué atributo seleccionar?

Day	Outlook	Temperature	Humidity	Wind	PlayTenn
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

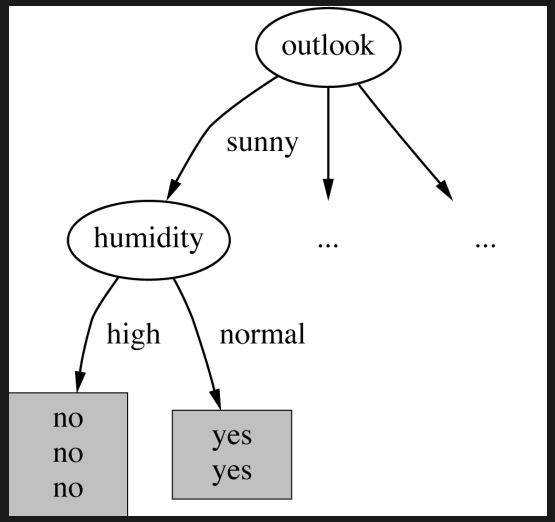


¿Cuál atributo?

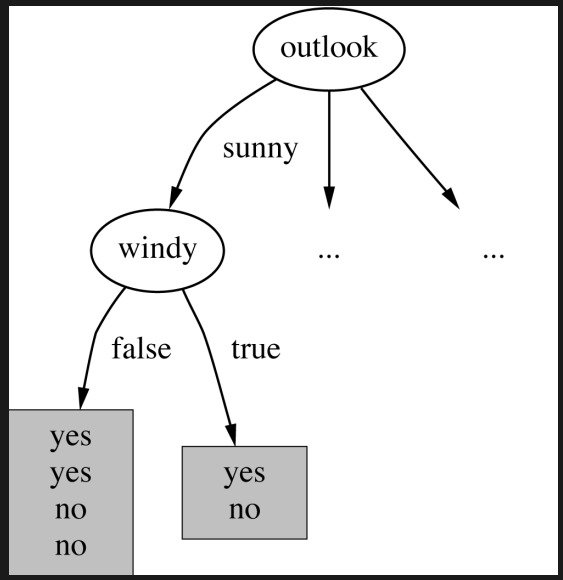
¿Qué atributo seleccionar?



Gain(S,Temp.)=?

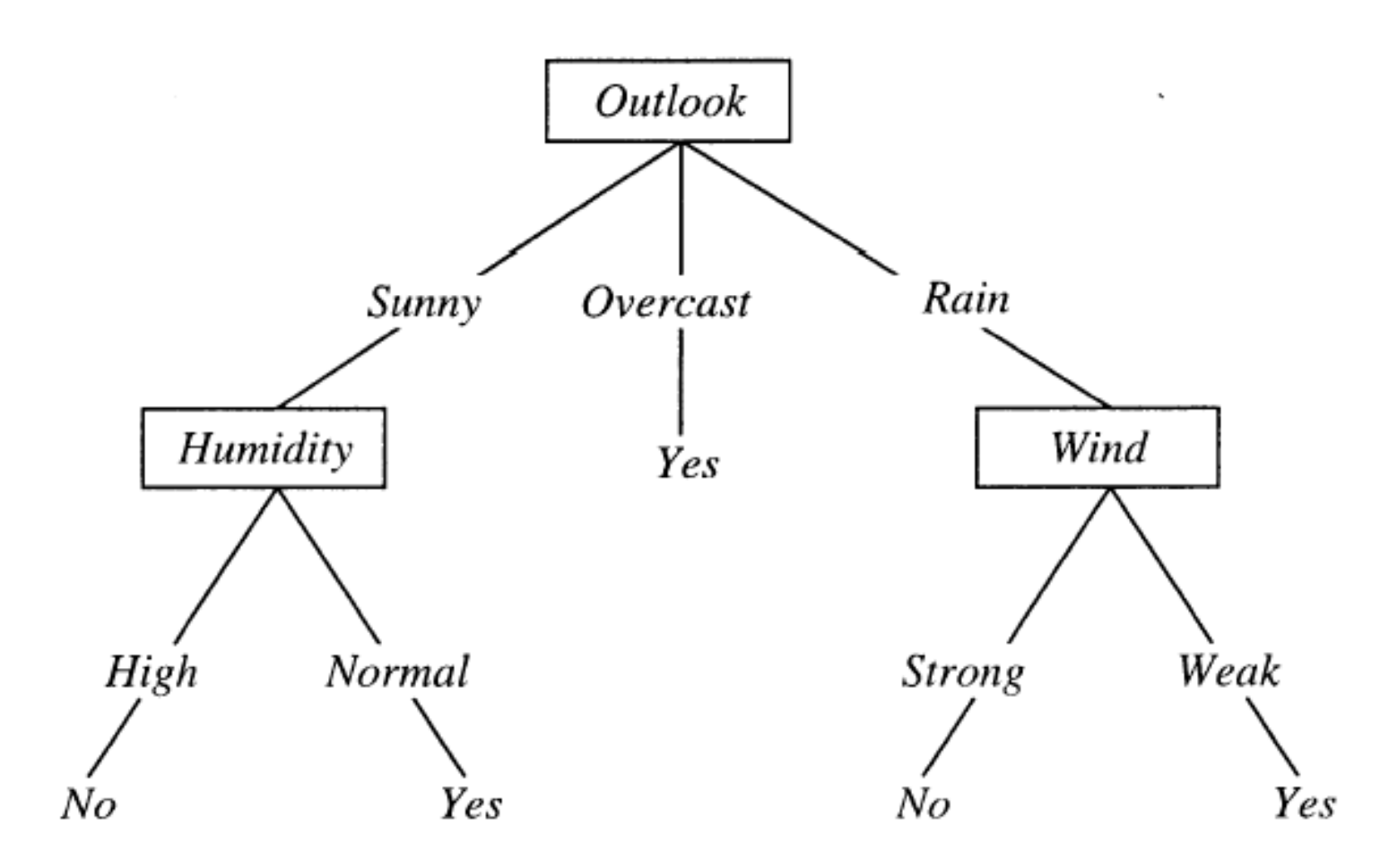


Gain(humid.,Temp.)=?



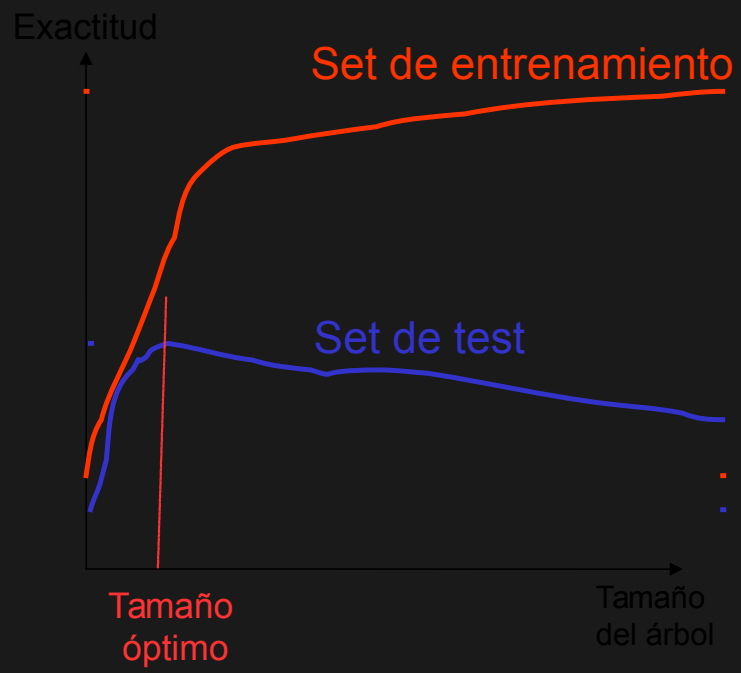
Gain(S,Windy)=?

Árbol final


Outlook
Sunny
Overcast
Rain
Humidity
Yes
Wind
High
Normal
Strong
Weak
No
Yes
No
Yes

Sobreajuste (*Overfitting*)

- El gráfico muestra el comportamiento típico en el set de entrenamiento:
  - Exactitud en set de entrenamiento crece a medida que el árbol de decisión crece.
  - Existe un momento en que la exactitud en el set de test comienza a disminuir, ¿Por qué?.
  - El árbol sobremodela ejemplos de entrenamiento, perdiendo su capacidad de generalización.
  - Atributos menos predictivos, agregados al final, introducen ruido.



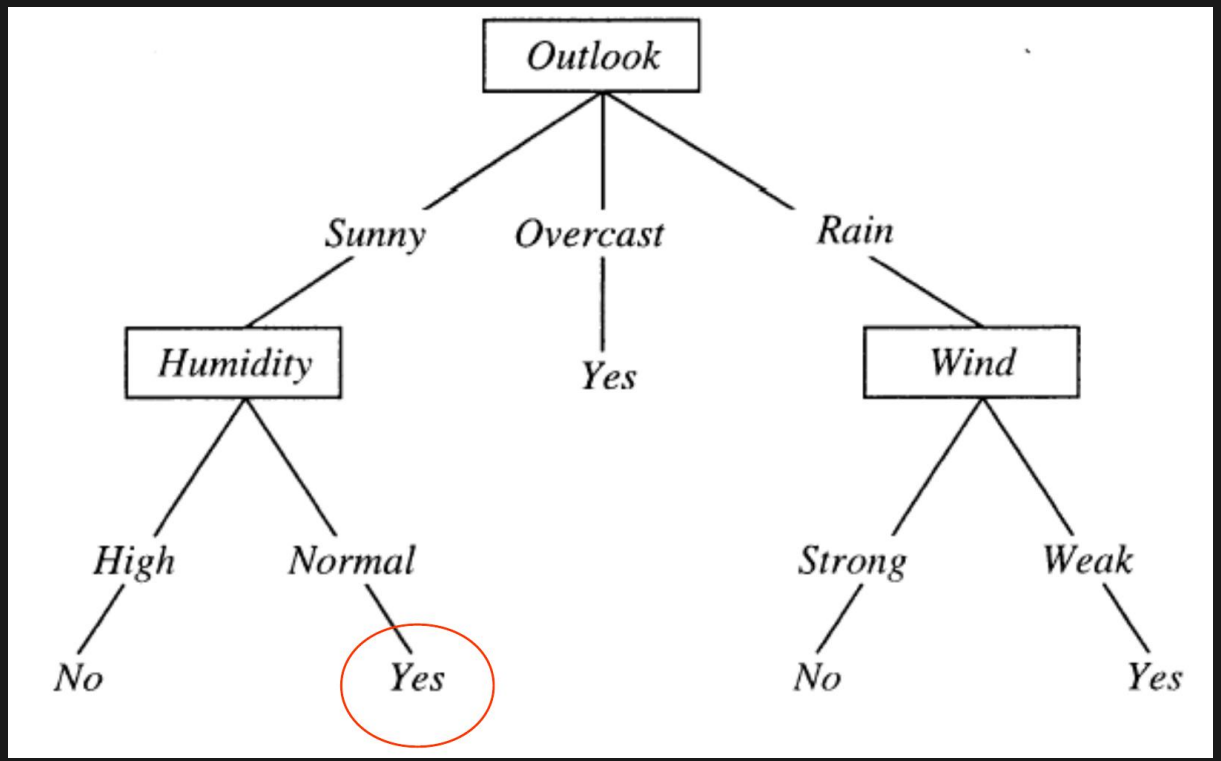
Escenario Deseado

Detener la construcción del árbol antes que se produzca *overfitting*.

Sobreajuste (*Overfitting*)

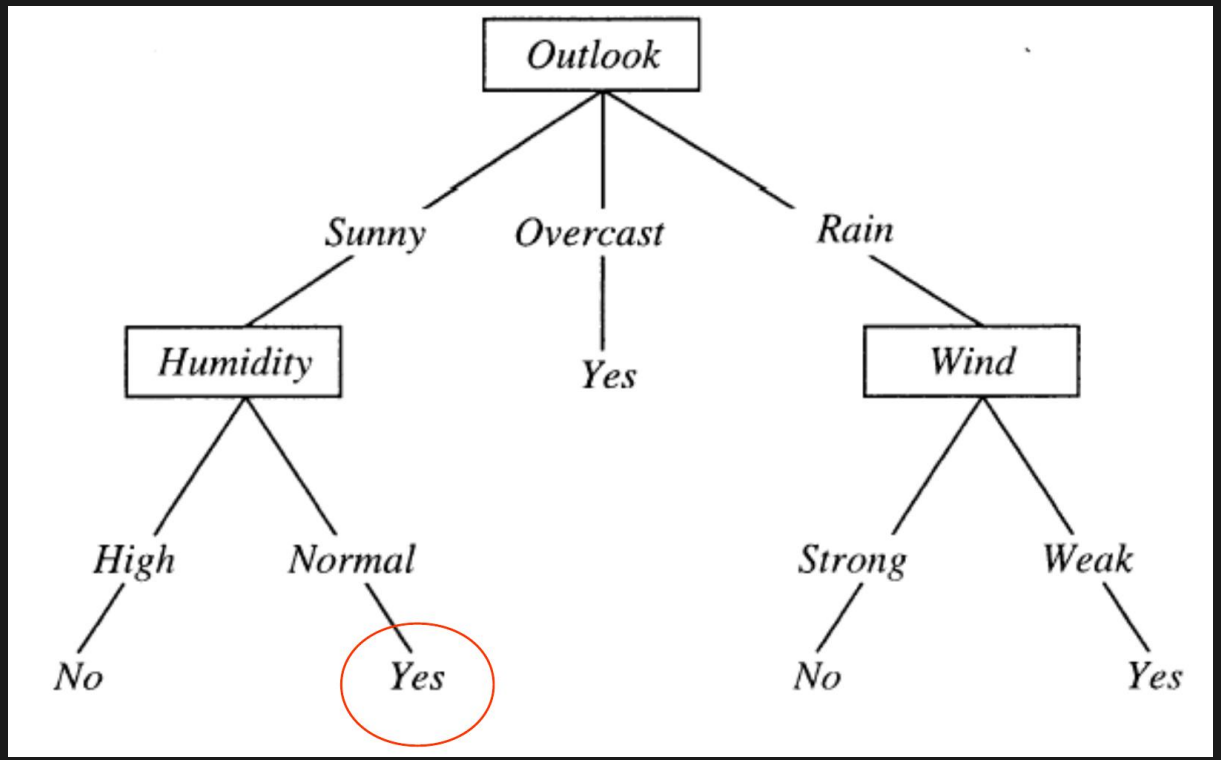
Por ejemplo, que sucede si agregamos el siguiente registro al ejemplo sobre jugar tenis:

Sunny, Hot, Normal, Strong, PlayTennis = NO



¿Cómo evitamos el *overfitting*?

- Parar la construcción del árbol cuando el número de registros restante no es estadísticamente significativo.
- Construir el árbol sin considerar restricciones de tamaño y luego podar usando rendimiento en set de validación.



Ganancia de Información: Problema

- En cada paso de la construcción del árbol, GI es usada para cuantificar el atributo que presenta mayor discriminatividad, medida como homogeneidad de grupos resultantes respecto a los valores de la clase.
- GI entrega buenos resultados y es muy usado en la práctica. Sin embargo, un inconveniente de GI es que no funciona bien con atributos que toman muchos valores, por qué?

Atributos con muchos valores

- Si un atributo tiene muchos valores, probablemente la métrica de ganancia de información lo seleccionará ¿Por qué?
  - Ej. Día=Julio 7 2005
- Una forma de solucionar el problema es usar la razón de ganancia (*GainRatio*):

$$GainRatio(S, A) \equiv \frac{Gain(S, A)}{SplitInformation(S, A)}$$

$$SplitInformation(S, A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} log_2 \frac{|S_i|}{|S|}$$

- *SplitInformation*: mide entropía respecto al número de instancias $|S_i|$ en cada subgrupo. Mientras más uniforme sea el número de registros que sigue cada link, mayor será el *SplitInformation*.

Árboles de decisión ¿Cuándo?

- Problemas de clasificación.
- Necesitamos generar reglas de decisión entendibles por personas. Un árbol de decisión es una disyunción de conjunciones.
- Atributos son discretos o discretizables sin gran pérdida de información.

Árboles de decisión: Regresión

- Es *posible* usarlos para Regresión haciendo que los nodos hojas entreguen el resultado promedio de los ejemplos.
- En vez de ocupar Information Gain para hacer *split* usamos Reducción de Varianza (VR) como métrica.


Random Forest:
Uniendo muchos árboles de decisión

La Sabiduría del Grupo...

Random Forest

- Se entrenan L árboles de decisión usando en cada caso N datos de entrenamiento.
- Los N datos de entrenamiento corresponden a muestras con reemplazo del set original de datos, usualmente también de tamaño N.
- Clasificación está dada por la predicción más común entre los L árboles resultantes (majority vote).

Fomentando diversidad en el set de árboles del ensamble (why?) :

- Cada árbol de decisión es entrenado usando sólo un subconjunto m de los atributos disponibles.
- Cada árbol es construído hasta convergencia sin poda (fully grown tree, no pruning).

Random Forest

Algoritmo

Sea:

N = número de instancia entrenamiento,

M = número de atributos,

m = número atributos para cada árbol, $m \ll M$:

For (1:L),

- Muestrear N ejemplos del set de entrenamiento con reemplazo.
- Seleccionar aleatoriamente m features.
- Entrenar árbol de decisión usando las N instancias y m atributos.
- Entrenar hasta convergencia sin poda.

End;

- Clasificar nuevas instancias usando votación por mayoría.

Random Forest: Diversidad

- La forma más directa de regular la diversidad viene del valor de m.
- Breiman et al. 1996, muestra que el error de generalización (GE) de un RF depende de la exactitud (*strength*) de cada árbol y el grado de correlación entre ellos.
- $GE \leq \phi \frac{1-s^2}{s^2}$, donde ϕ es la correlación media entre pares de árboles en el bosque (*forest*) y s es el *strength* del ensamble (i.e., el valor esperado de la exactitud de cada árbol respecto a todo el ensamble).
- Al mover m nos enfrentamos a un *tradeoff* entre diversidad y *strength* de los árboles. (¿Por qué?)

Resultado Experimental

$$m = \sqrt{M} \quad \text{(“Rule of thumb”)}$$

Random Forest: Microsoft Kinect

Real-Time Human Pose Recognition in Parts from Single Depth Images

Jamie Shotton Andrew Fitzgibbon Mat Cook Toby Sharp Mark Finocchio
Richard Moore Alex Kipman Andrew Blake
Microsoft Research Cambridge & Xbox Incubation

Abstract

We propose a new method to quickly and accurately predict 3D positions of body joints from a single depth image, using no temporal information. We take an object recognition approach, designing an intermediate body parts representation that maps the difficult pose estimation problem into a simpler per-pixel classification problem. Our large and highly varied training dataset allows the classifier to estimate body parts invariant to pose, body shape, clothing, etc. *Finally we generate confidence-scored 3D proposals of several body joints by reprojecting the classification result and finding local modes.*

The system runs at 200 frames per second on consumer