Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2613 - Inteligencia Artificial

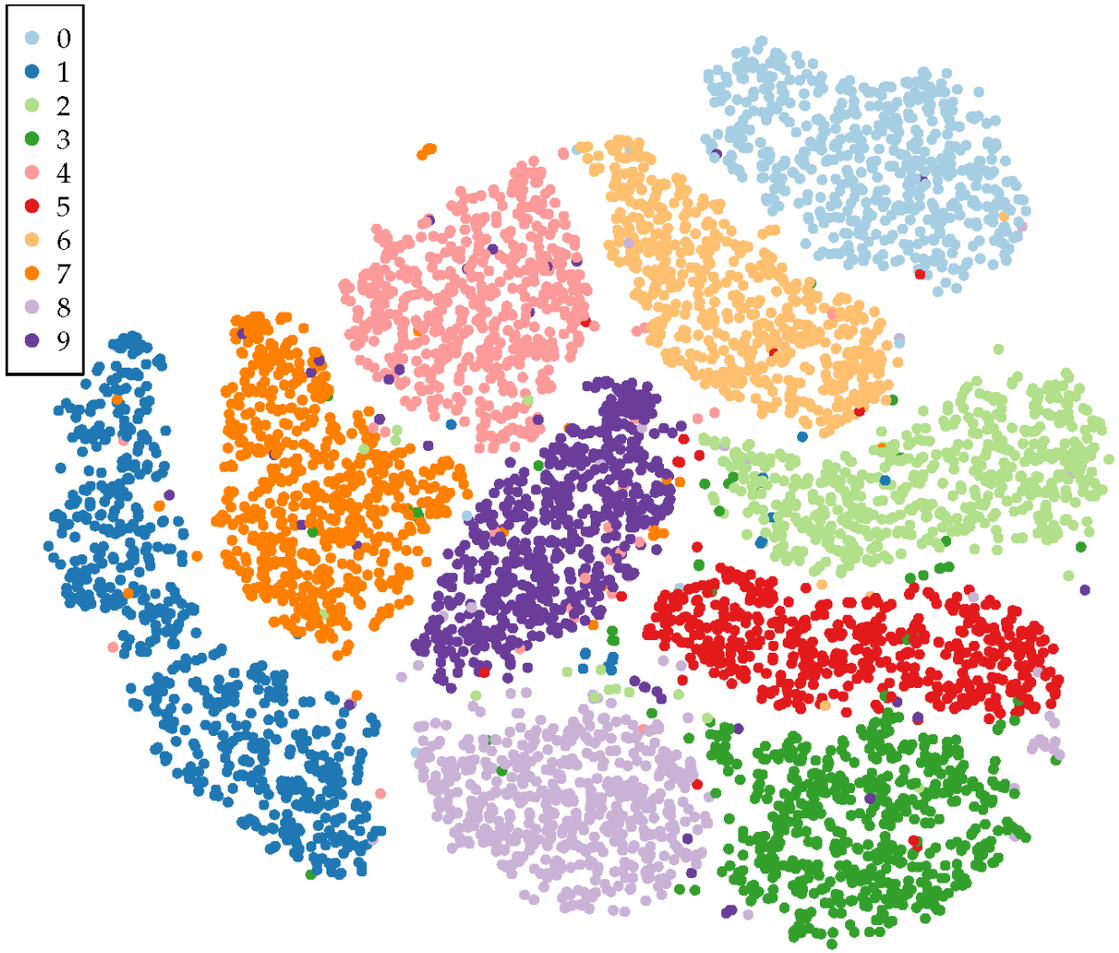
Árboles de decisión

Hans Löbel
Dpto. Ingeniería de Transporte y Logística
Dpto. Ciencia de la Computación

Recapitulemos un poco lo que hemos visto hasta ahora en el curso

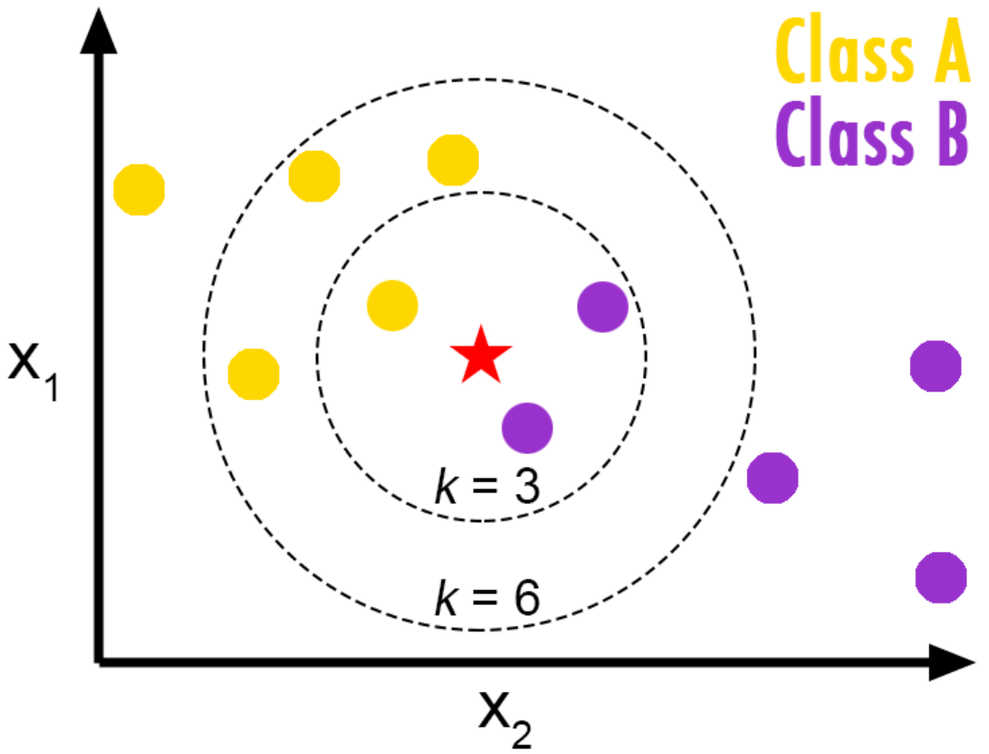
- Discutimos sobre la programación directa vs aprendizaje para resolver problemas computacionalmente
- Continuamos con la relación íntima entre ML y datos, desde el punto de vista del aprendizaje inductivo.
- Luego, vimos los conceptos fundamentales de ML, generalización y representación de datos.
- Ahora comenzaremos con la revisión de modelos y algoritmos de aprendizaje supervisado.

¿Cuál es el esquema más simple que podríamos usar para resolver este problema de clasificación?



Clasificador de k-vecinos más cercanos permite enfrentar este tipo de problema de manera intuitiva

- Entrenamiento consiste únicamente en almacenar los datos de entrenamiento.
- La inferencia sobre un nuevo ejemplo se basa directamente en un voto de mayoría sobre la clase de los k ejemplos más cercanos (similares).
- Supuestos muy fuertes: ejemplos de la misma clase son cercanos (espacio de características semántico) o métrica de distancia especializada (semántica).
- Funcionamiento complejo cuando el espacio de características no es continuo.

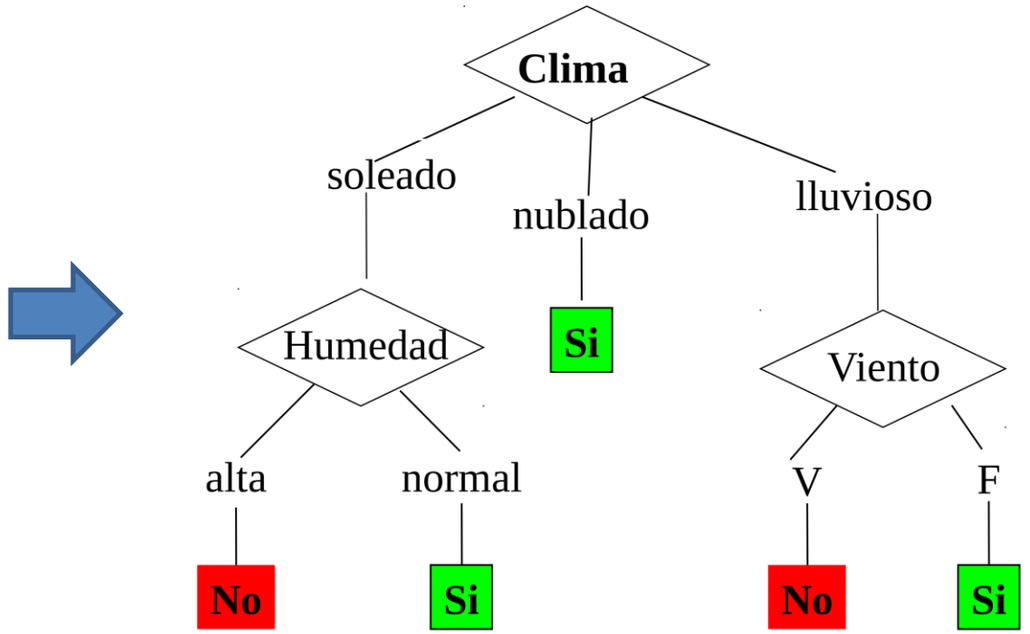


¿Cómo solucionamos este nuevo problema de clasificación?

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

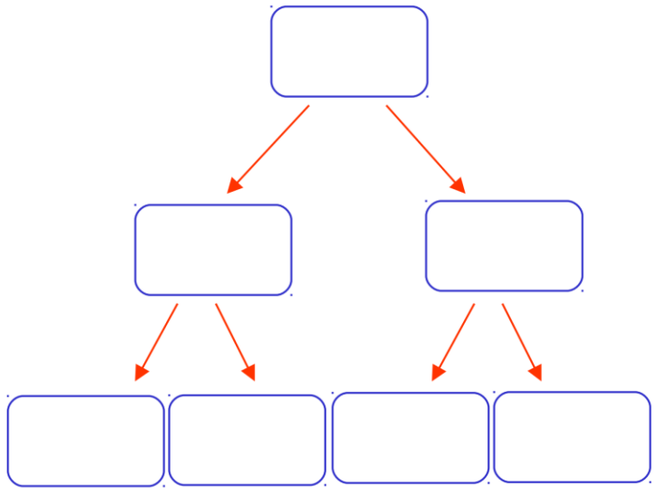
¿Cómo solucionamos el siguiente problema de clasificación?

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



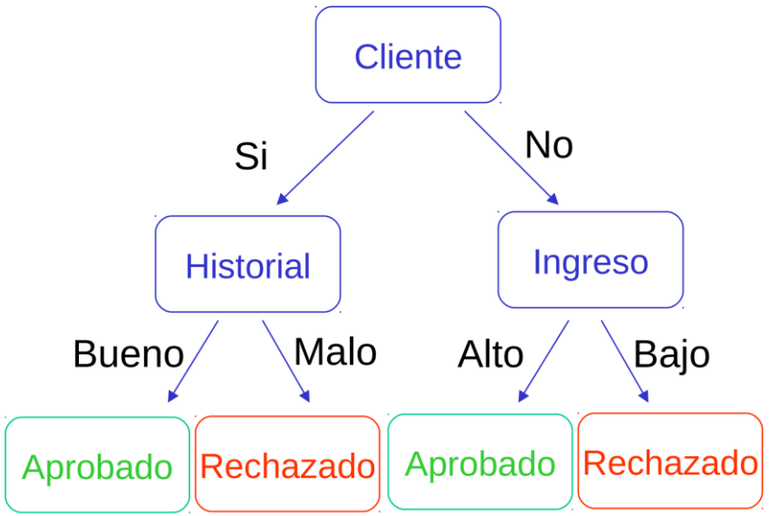
Árboles de decisión pueden solucionar el caso anterior

- Técnica de aprendizaje supervisado.
- Pueden realizar clasificación y regresión.
- Pueden usarse sobre distintos tipos de *features* (binaria, categórica, numérica, etc).

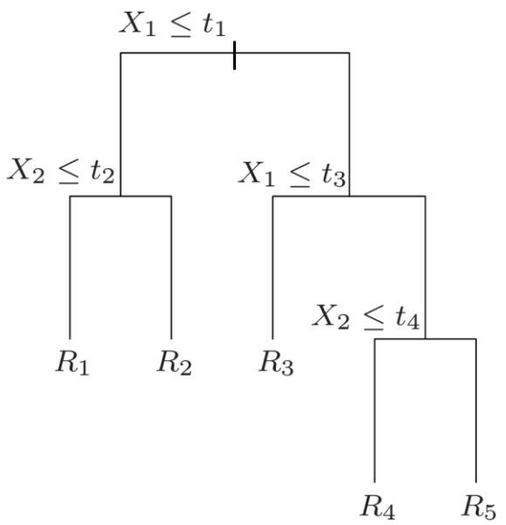


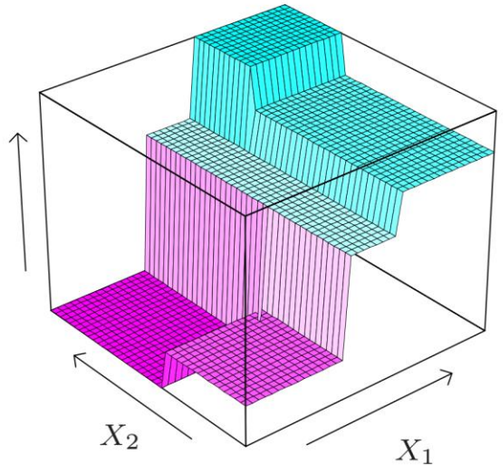
Árboles de decisión son ampliamente utilizados en la práctica

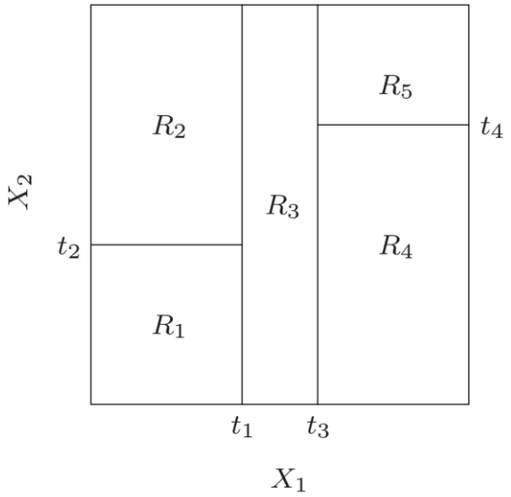
- Cada nodo interno representa una característica/atributo y cada nodo hoja representa una categoría.
- En cada nodo interno, se realiza un test en base a los valores de la característica.
- Aristas representan el resultado del test.
- Para clasificar un registro, se debe pasar desde la raíz hasta alguna hoja.



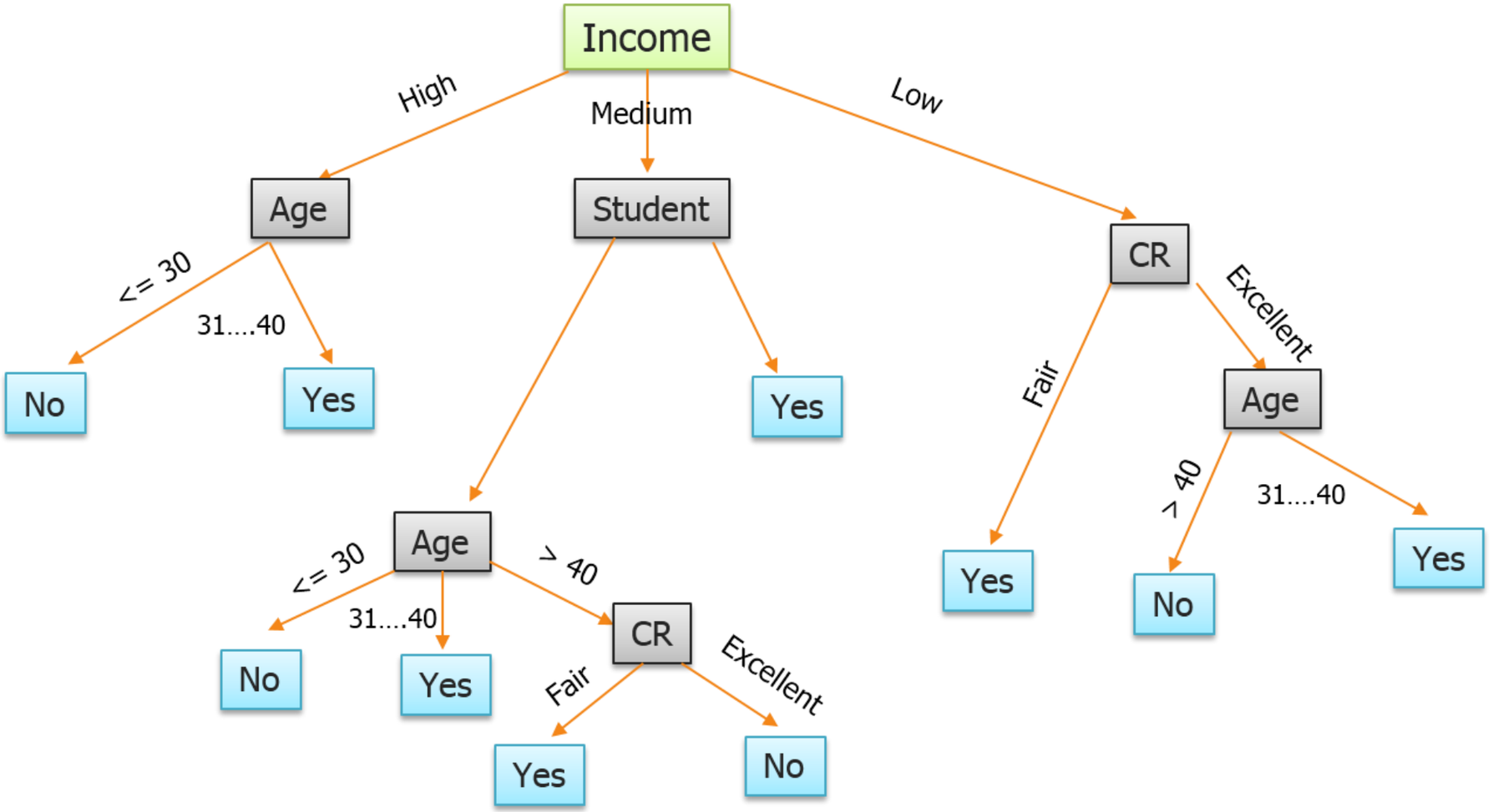
Los árboles de decisión pueden “verse” de múltiples maneras





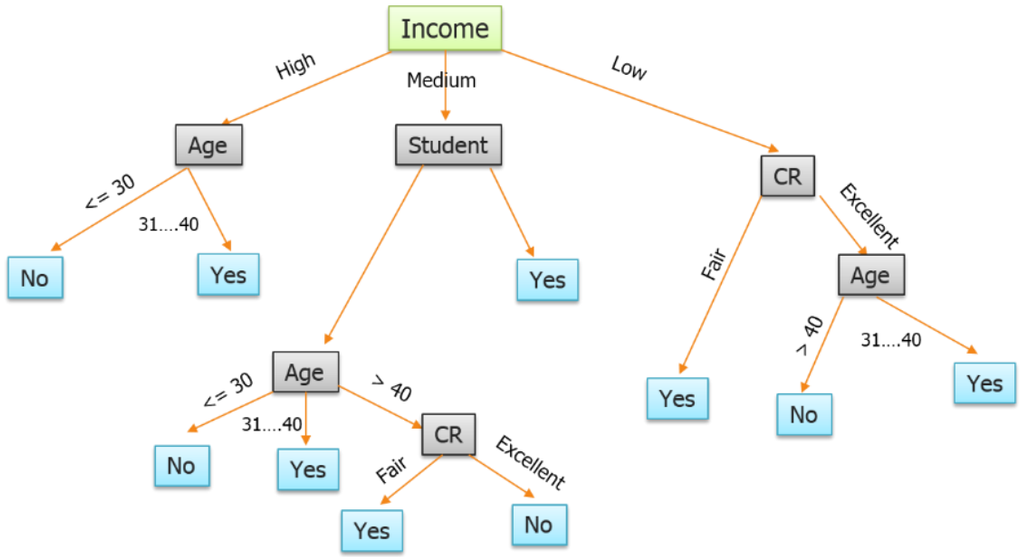


¿Cómo construimos un árbol de decisión?



Podemos usar un algoritmo recursivo para construir un árbol

1. ¿Pertenecen todos los registros a la misma clase?
   - Retornar marcando el nodo hoja con la clase respectiva.
2. ¿Tienen todos los registros el mismo valor para todas las características?
   - Retornar marcando nodo hoja con la clase más común.
3. De lo contrario:
   i. Seleccionar la mejor característica.
   ii. Usar esta característica como nodo raíz.
   iii. Dividir el set de entrenamiento restante de acuerdo a este atributo y para cada rama resultante continuar la construcción del árbol en forma recursiva.



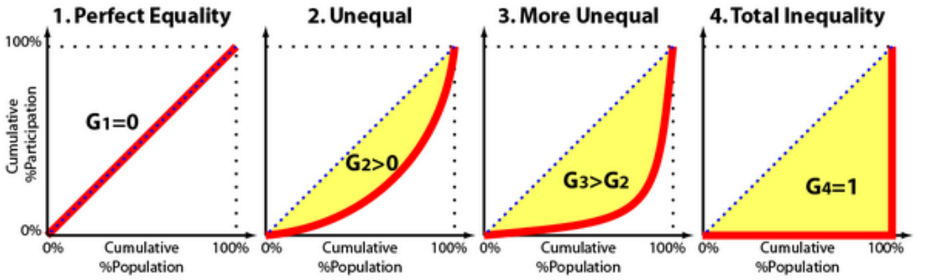
Podemos destilar este paso en dos preguntas más específicas

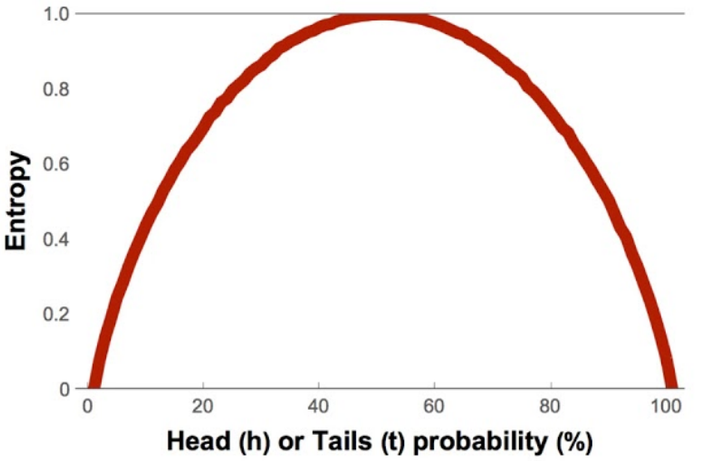
1. ¿Cómo defino cuál es la mejor característica?

2. Si la mejor característica es numérica, ¿en qué parte de su dominio pongo el umbral de separación?

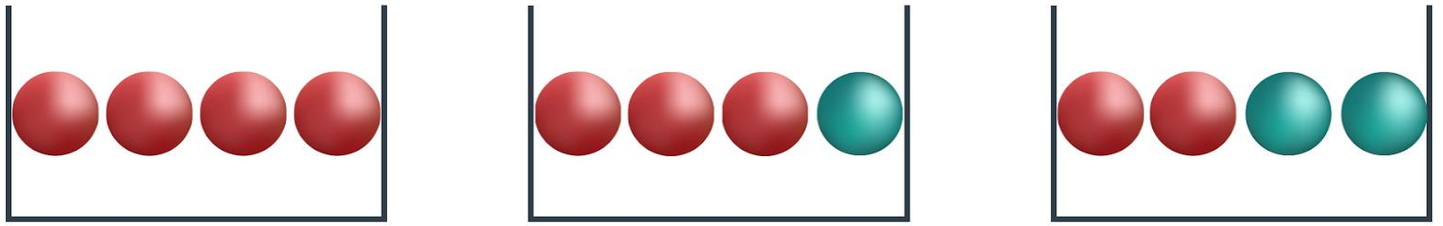
La elección de la mejor característica depende de la tarea que busquemos solucionar

- Si objetivo es clasificar, es razonable que la mejor característica sea aquella que mejor separe las clases.
- Dos maneras típicas de medir esto son:
  - Gini Index: desigualdad (inequidad) sobre distintas categorías.
  - Information Entropy: uniformidad de una distribución o cantidad de información necesaria para codificar la ocurrencia de eventos.





Entropía permite capturar de manera eficiente cuán informativa es una distribución



High Knowledge
Low Entropy

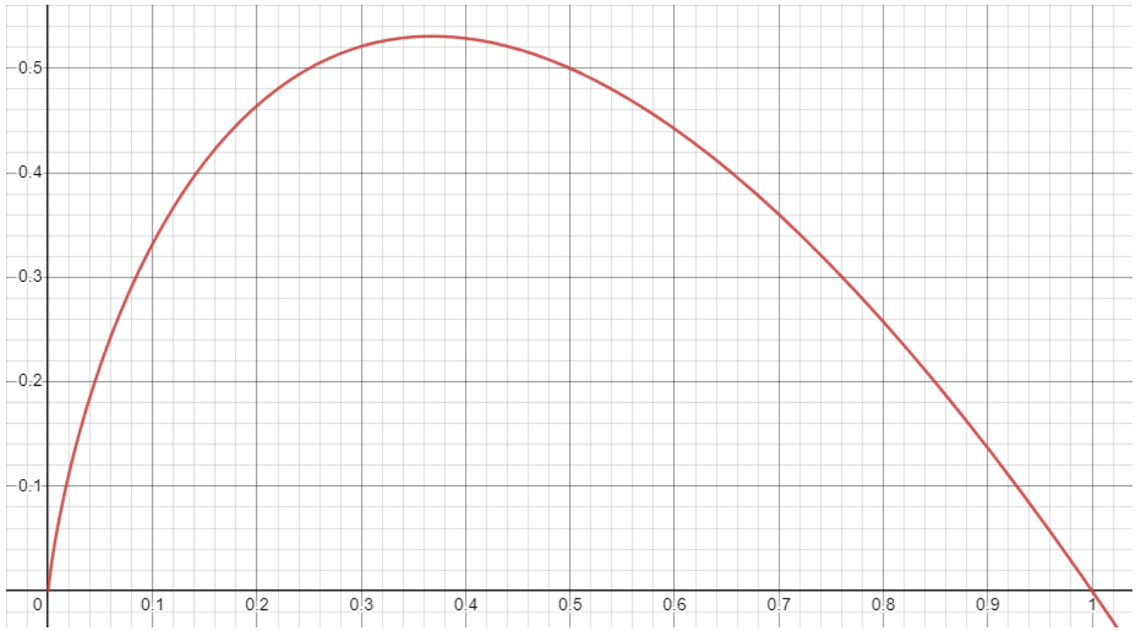
Medium Knowledge
Medium Entropy

Low Knowledge
High Entropy

Entropía permite capturar de manera eficiente cuán informativa es una distribución

- “Intuitivamente”, la entropía puede verse como un promedio ponderado de la información sobre los eventos particulares:

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



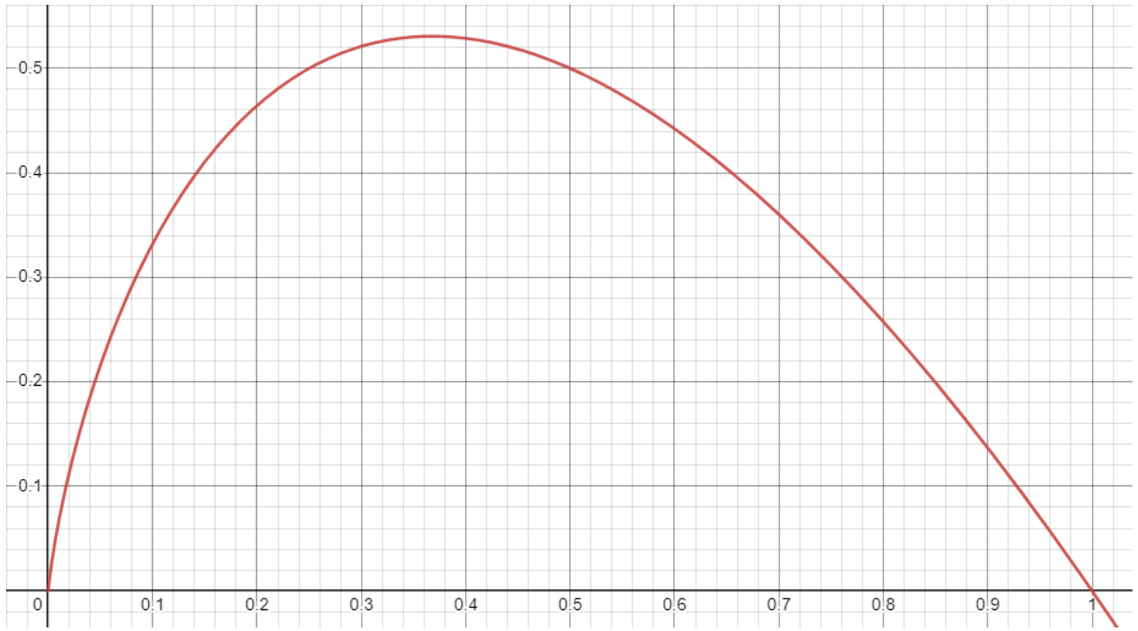
$$f(x) = -x log_2(x)$$

Entropía permite capturar de manera eficiente cuán informativa es una distribución

- "Intuitivamente", la entropía puede verse como un promedio ponderado de la información sobre los eventos particulares:

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

- Por ejemplo:
  - 4 clases (A,B,C,D): 10 registros clase A, 20 clase B, 30 clase C, 40 clase D.
  - Entropía= -[(.1 log .1) + (.2 log .2) + (.3 log .3) + (.4 log .4)] = 1.85 bits.

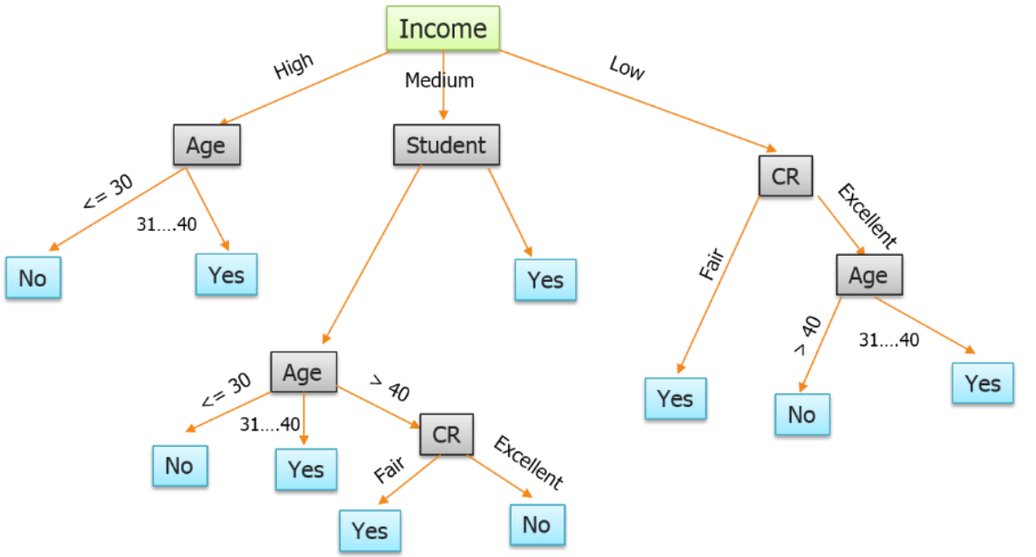


$$f(x) = -x log_2(x)$$

* https://www.khanacademy.org/computing/computer-science/informationtheory/moderninfotheory/v/information-entropy

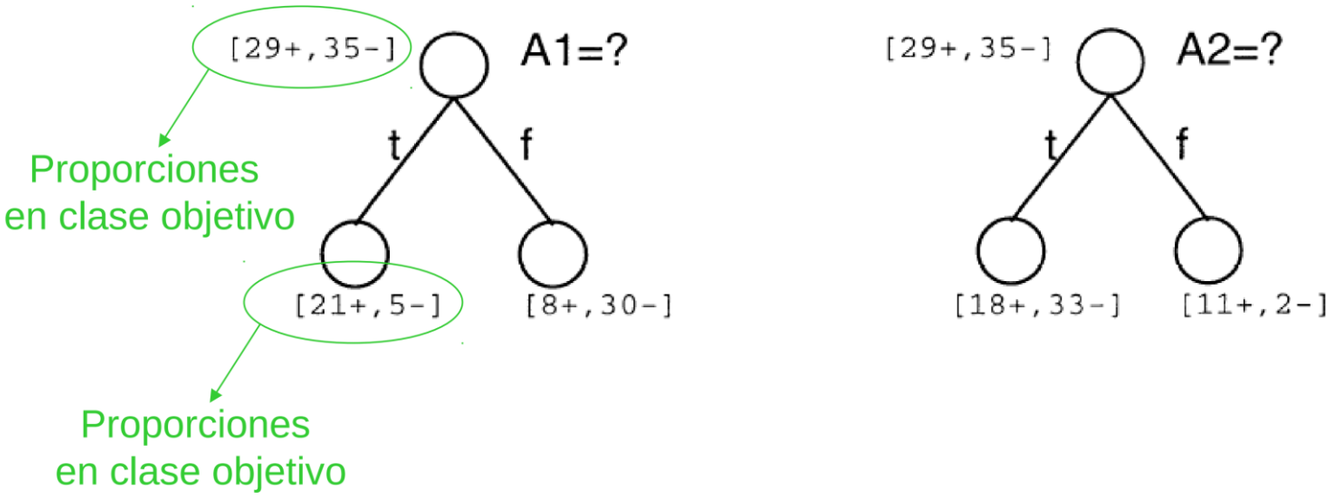
¿Cómo nos ayuda entonces la entropía a elegir la mejor característica?

- Cada valor/umbral de una feature genera una partición distinta.
- Cada una de estas particiones tendrá su propia entropía.
- Ahora sí, intuitivamente, algo como la media ponderada de la entropía de las particiones generadas podría indicar cuán buena o mala es una característica.
- Esta idea es formalizada a través de una métrica llamada Ganancia de Información (Information Gain).



Elegimos la característica que entrega la mayor ganancia de información (mayor reducción de entropía)

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

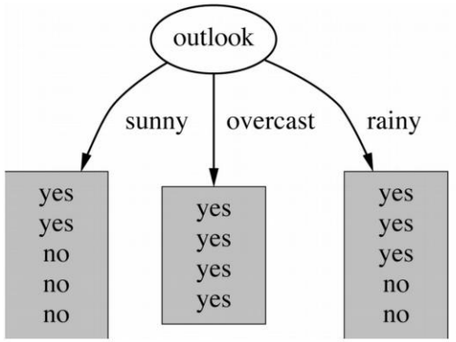


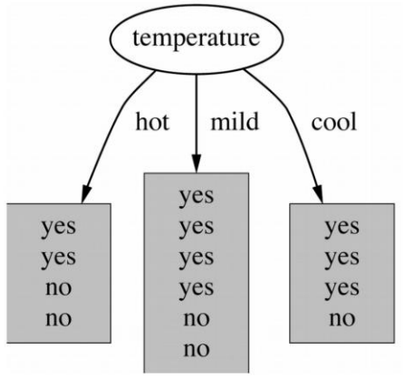
Elegimos la característica que entrega la mayor ganancia de información (mayor reducción de entropía)

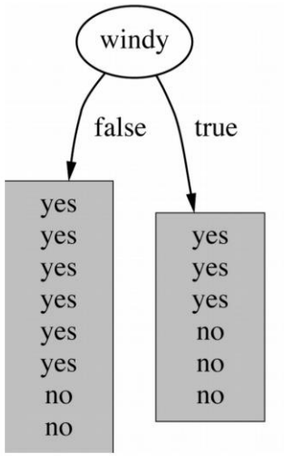
Day	Outlook	Temperature	Humidity	Wind	PlayTenn
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak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

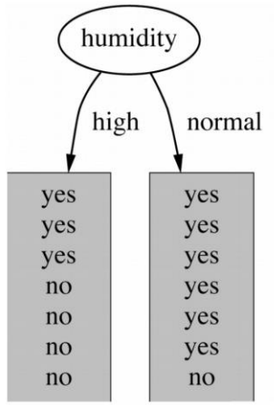
Elegimos la característica que entrega la mayor ganancia de información (mayor reducción de entropía)

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

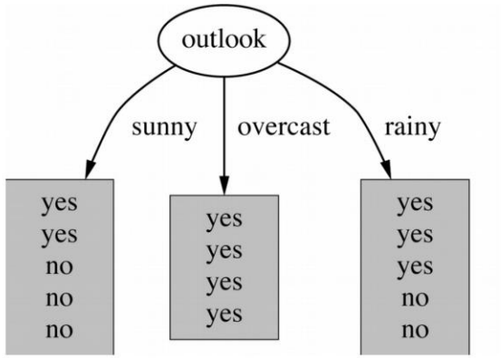








Elegimos la característica que entrega la mayor ganancia de información (mayor reducción de entropía)



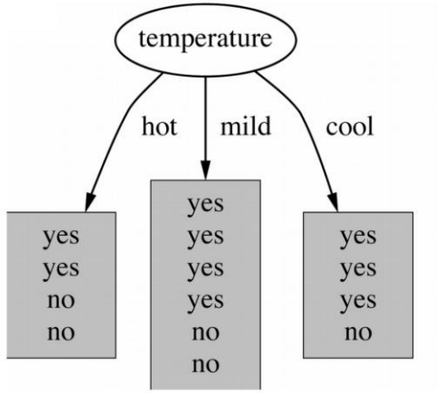
S:[9+,5-]
E=0.940

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

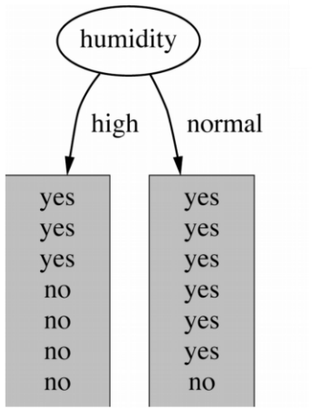
Temperat.

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

Elegimos la característica que entrega la mayor ganancia de información (mayor reducción de entropía)



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

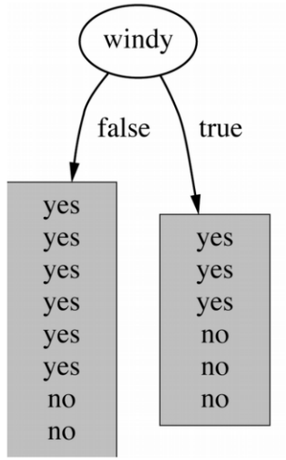
Humidity

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**

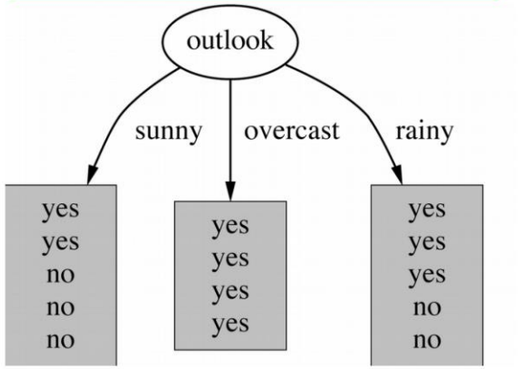
[6+,2-]
E=0.811

[3+,3-]
E=1

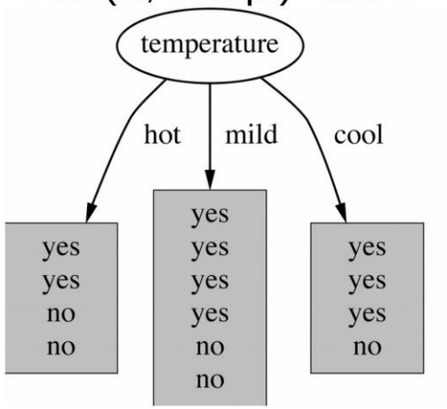
Gain(S, Windy)=0.940-(8/14)0.985-(6/14)
=0.048

Elegimos la característica que entrega la mayor ganancia de información (mayor reducción de entropía)

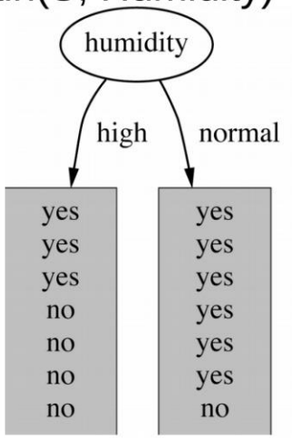
Gain(S, Outlook)=0.266



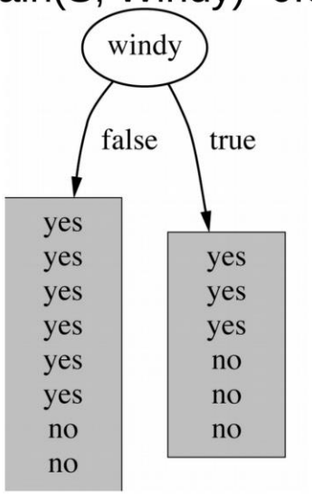
Gain(S, Temp.)=0.029



Gain(S, Humidity)=0.151

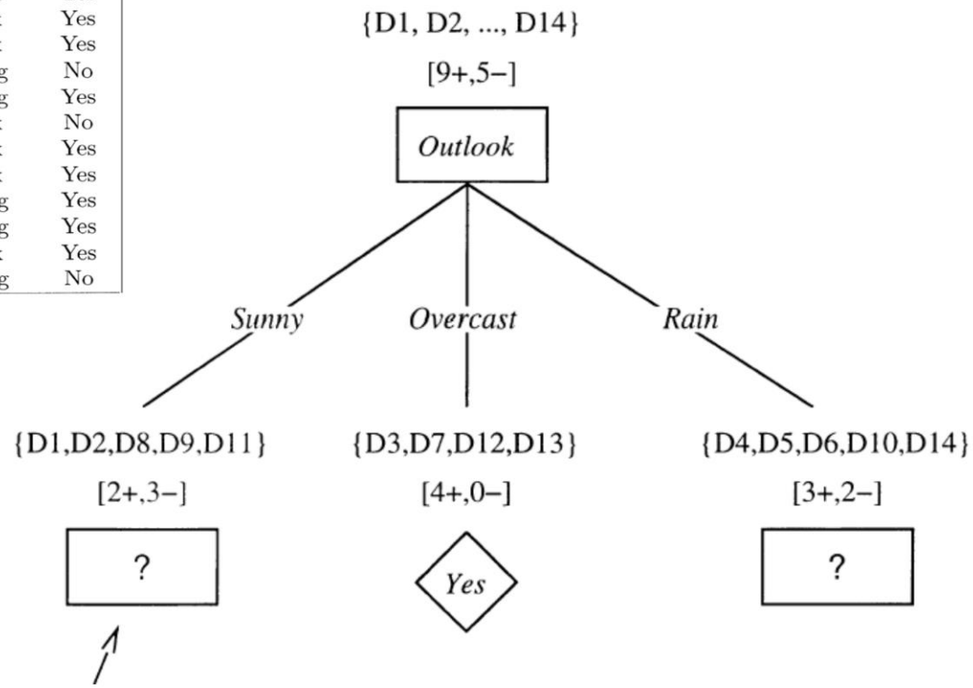


Gain(S, Windy)=0.048



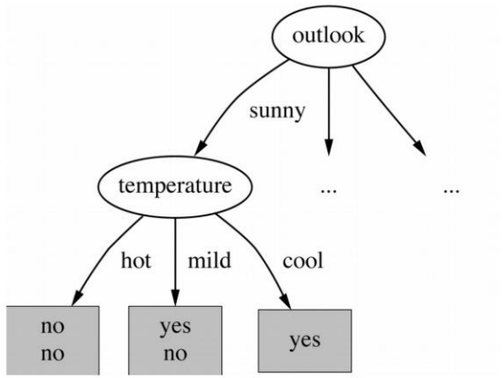
Elegimos la característica que entrega la mayor ganancia de información (mayor reducción de entropía)

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

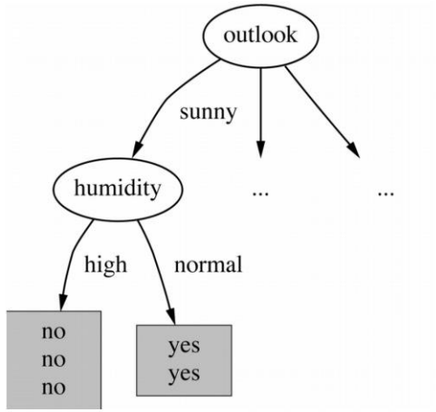


¿Cuál atributo?

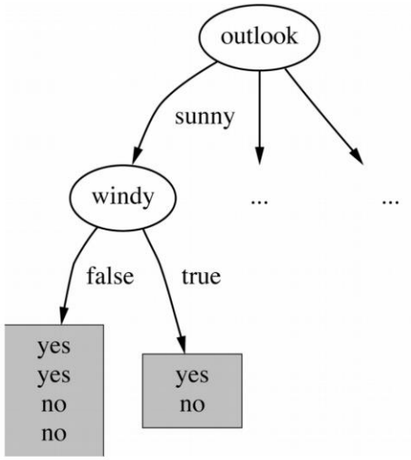
Elegimos la característica que entrega la mayor ganancia de información (mayor reducción de entropía)



Gain(S,Temp.)=?

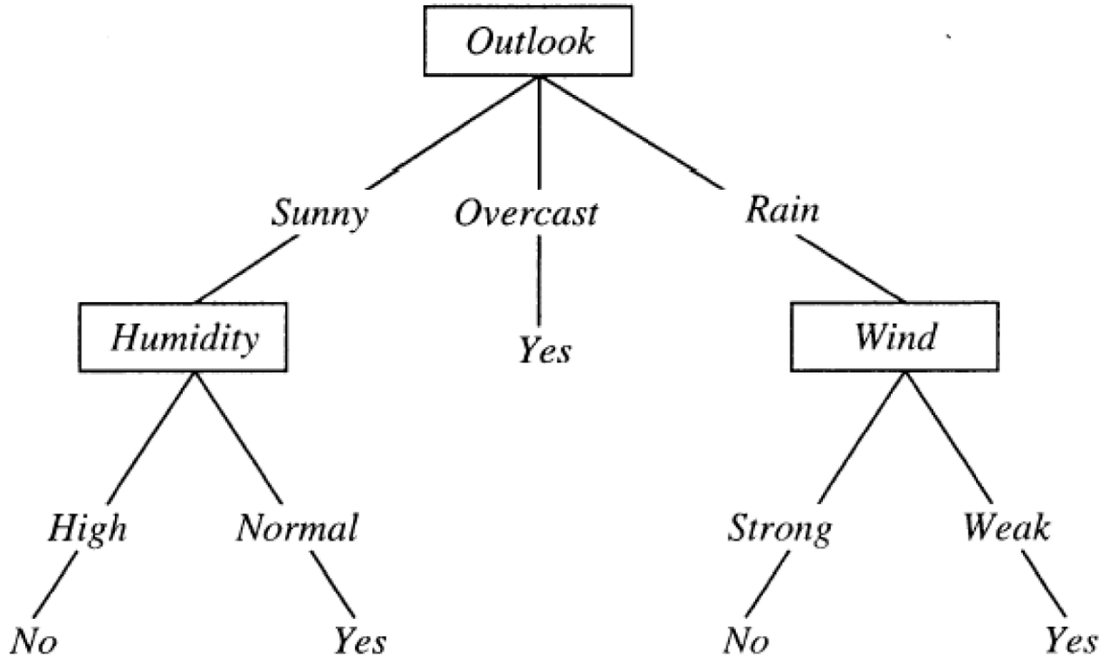


Gain(humid.,Temp.)=?



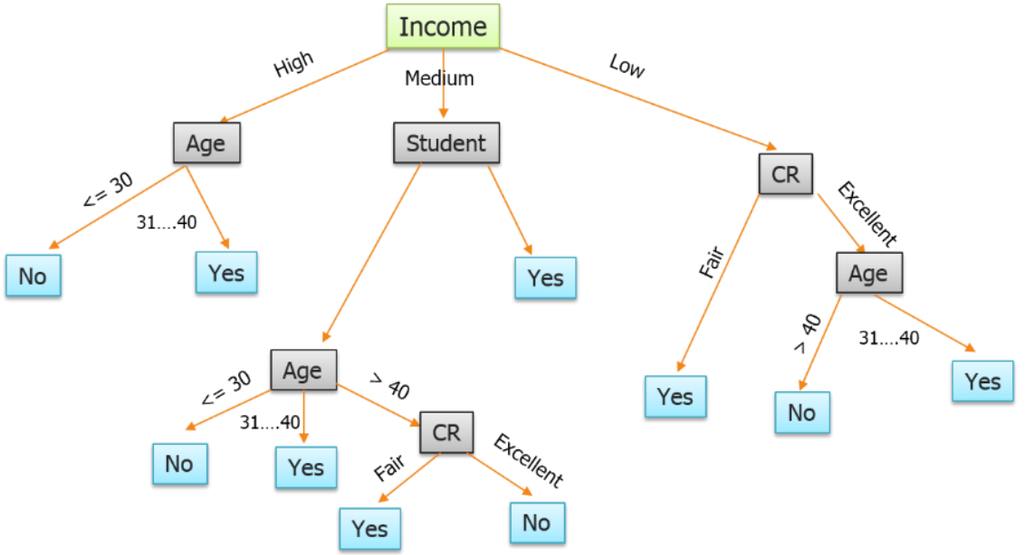
Gain(S,Windy)=?

Elegimos la característica que entrega la mayor ganancia de información (mayor reducción de entropía)



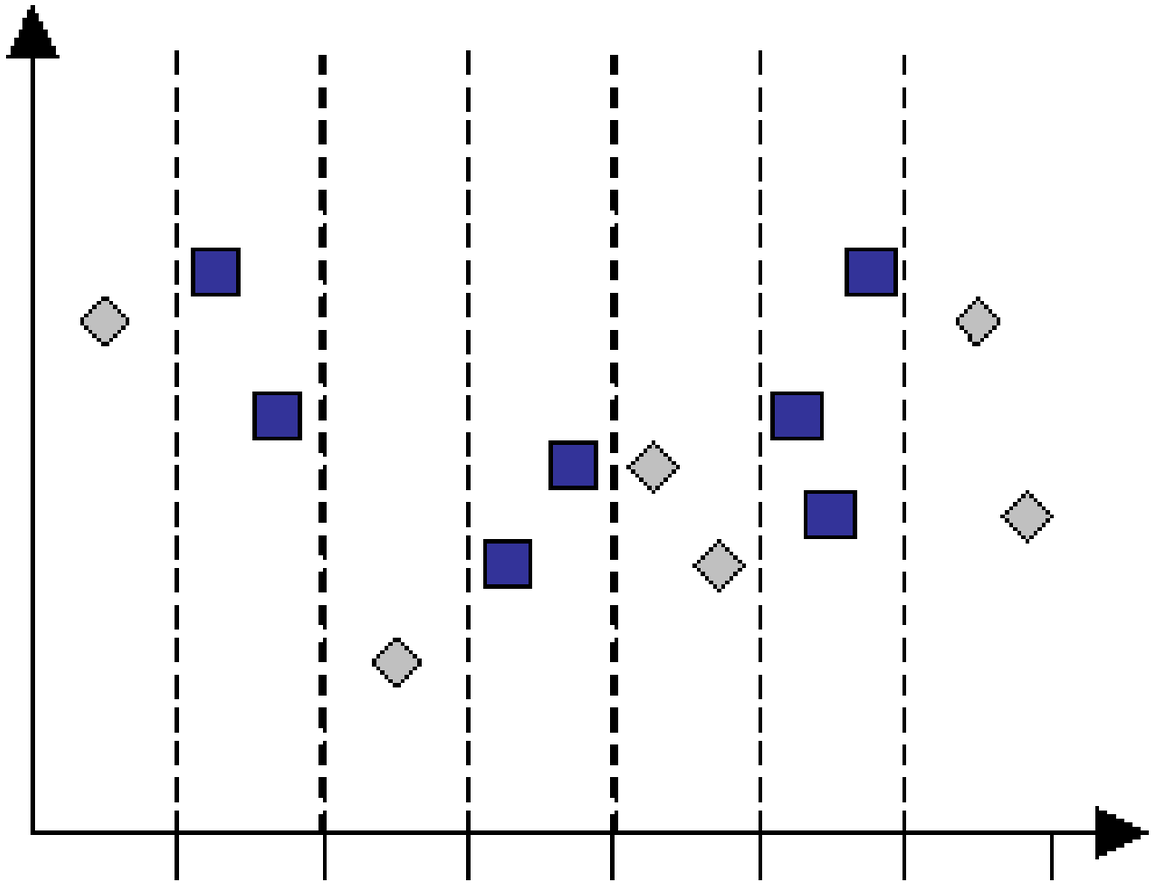
Aún nos queda responder una pregunta más sobre la construcción de árboles de decisión

1. ¿Cómo defino cuál es el mejor atributo?
2. ¿En qué parte del dominio pongo el umbral de separación para los atributos numéricos?



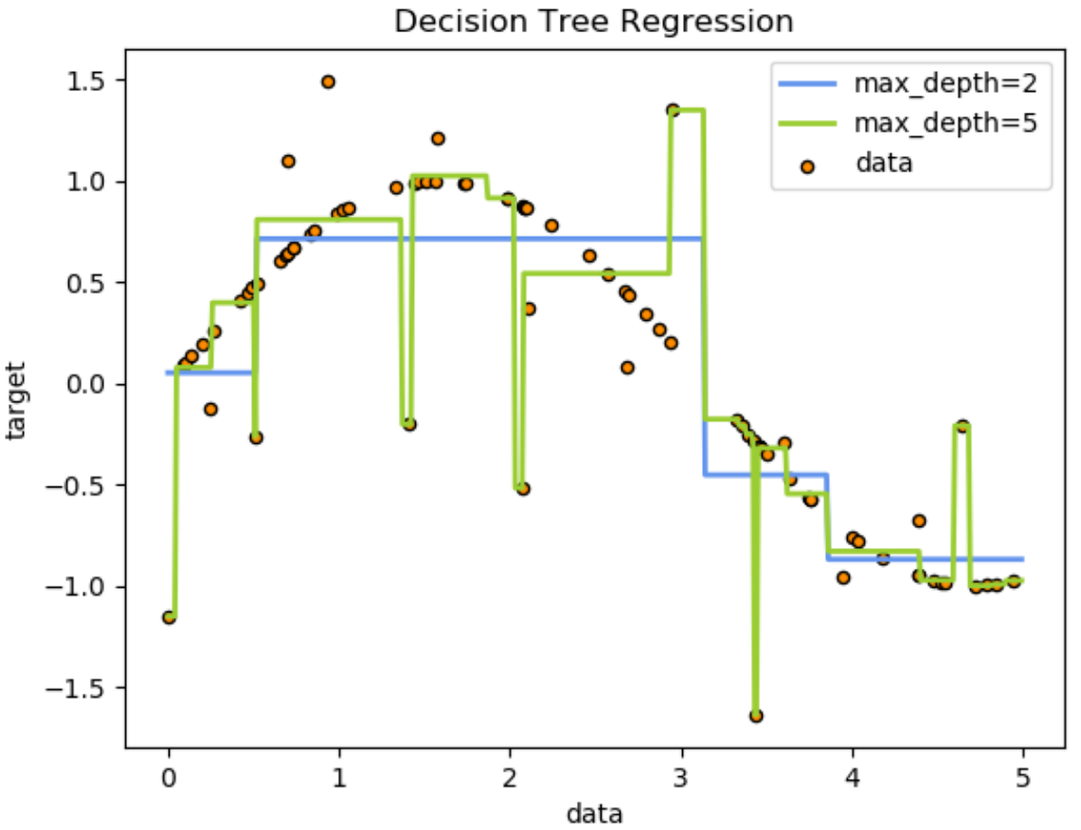
Nuevamente tenemos dos opciones directas

- Fuerza bruta
- Ordenar por dimensión, y evaluar split en cada cambio de categoría.



¿Qué hacemos si la tarea a resolver no es una clasificación?

- Si objetivo es una regresión, es razonable que el mejor atributo sea el que más disminuya el error en la estimación.
- Cada nodo hoja retorna la media de los ejemplos que contiene.
- En vez de medir IG, se calcula la disminución de la varianza (VR).
- Construcción recursiva sigue la misma idea que para clasificación.



Volvamos un poco a overfitting (sobreajuste) y complejidad

¿Qué tipo de árboles prefiere construir el algoritmo que analizamos?

(sesgo inductivo del algoritmo)

Árboles pocos profundos (mientras más arriba aumenta la información, mejor)

¿Tiene esto algo que ver con el sobreajuste?

Occam’s Razor[1] (o la navaja de Occam, claramente una traducción poco afortunada) be my guide

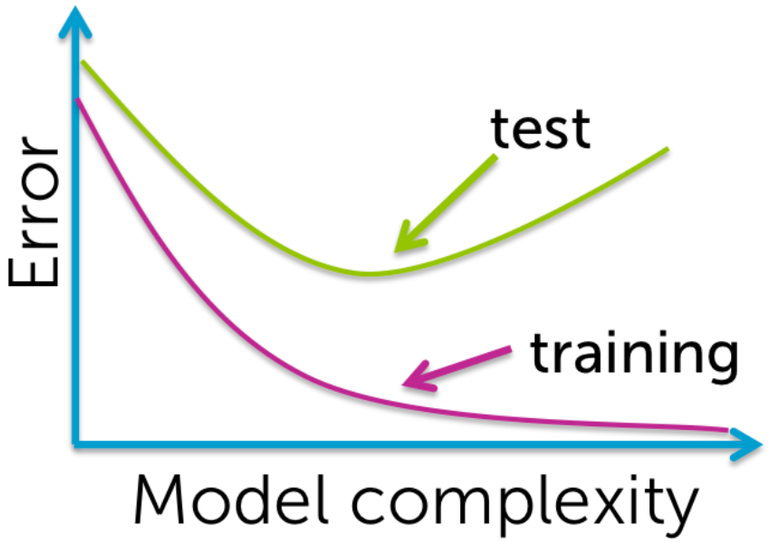
En igualdad de condiciones, la explicación más sencilla suele ser la más probable [2]

[1] Supuestamente fue enunciado mientras se afeitaba.
[2] Esto es un principio filosófico/metodológico, no una ley de la naturaleza.

Sobreajuste es un problema importante para los árboles de decisión

Existen varias técnicas que pueden ayudar reduciendo el sobreajuste.

- Detener construcción del árbol en base a un set de validación.
- Detener construcción cuando registros restantes no son estadísticamente significativos.
- Construir un árbol completo y luego podar ramas.
- Penalizar complejidad en métrica de selección del siguiente atributo.



Cuáles son los conceptos centrales de la clase de hoy

- Árboles son técnicas muy eficientes, pero con gran riesgo de sobreajuste.
- Simpleza en la explicación es uno de los criterios principales para evaluar soluciones a problemas de clasificación/regresión.
- Pueden ser muy útiles como primera solución a un problema, si se tienen features de tipos distintos, donde no es evidente como combinarlas.

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2613 - Inteligencia Artificial

Árboles de decisión

Hans Löbel
Dpto. Ingeniería de Transporte y Logística
Dpto. Ciencia de la Computación