# Árvores de Decisão

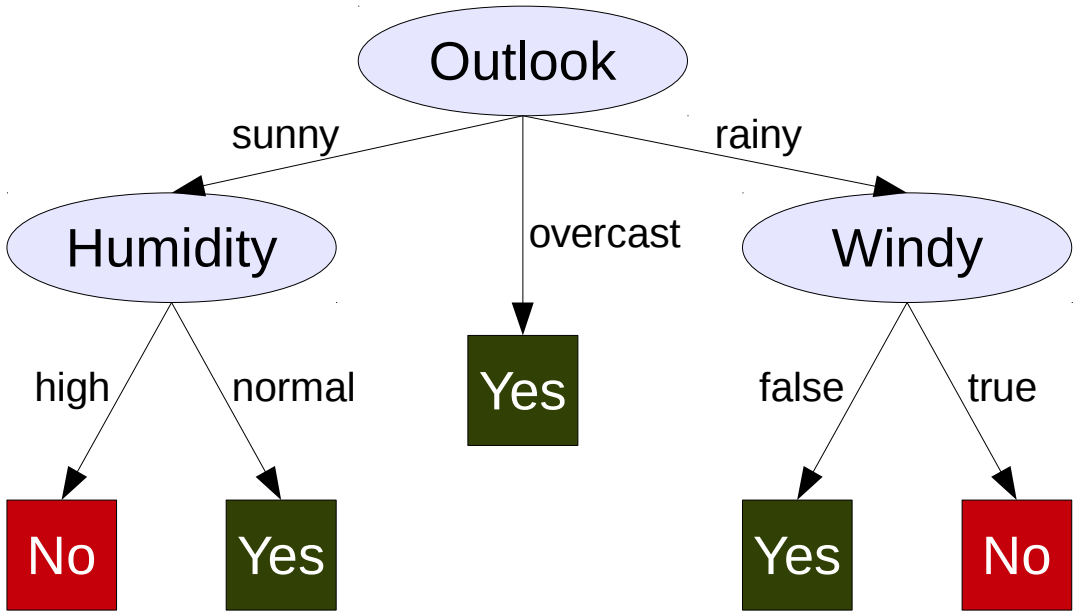
## SCC0244 - Mineração a partir de Grandes Bases de Dados

Rafael Geraldeli Rossi
Solange Oliveira Rezende

Slides baseados em [Han et al., 2011], [Tan et al., 2005] e [Witten and Frank, 2005]

# Introdução

- Uma árvore de decisão é uma estrutura na qual cada nó interno corresponde a um teste em um atributo, cada ramificação representa a saída de um teste, e cada nó folha representa um rótulo de classe



- Paradigma de aprendizado **simbólico**, ou seja, os padrões gerados são facilmente interpretados
- Algoritmos para a indução de árvores de decisão
  - ID3 (Iterative Dichotomiser) [Quinlan, 1986]
  - C4.5 (sucessor do ID3) [Quinlan, 1993]
  - Classification and Regression Trees (CART) [Breiman et al., 1984]
- Ambos os algoritmos adotam uma estratégia gulosa
  - As árvores são construídas de maneira *top-down*
  - Uma da estratégia de dividir e conquistar de maneira recursiva - o conjunto de treinamento é recursivamente particionado em subconjuntos conforme a árvore é construída

- Uma árvore de decisão gera hiperplanos de separação perpendiculares aos eixos

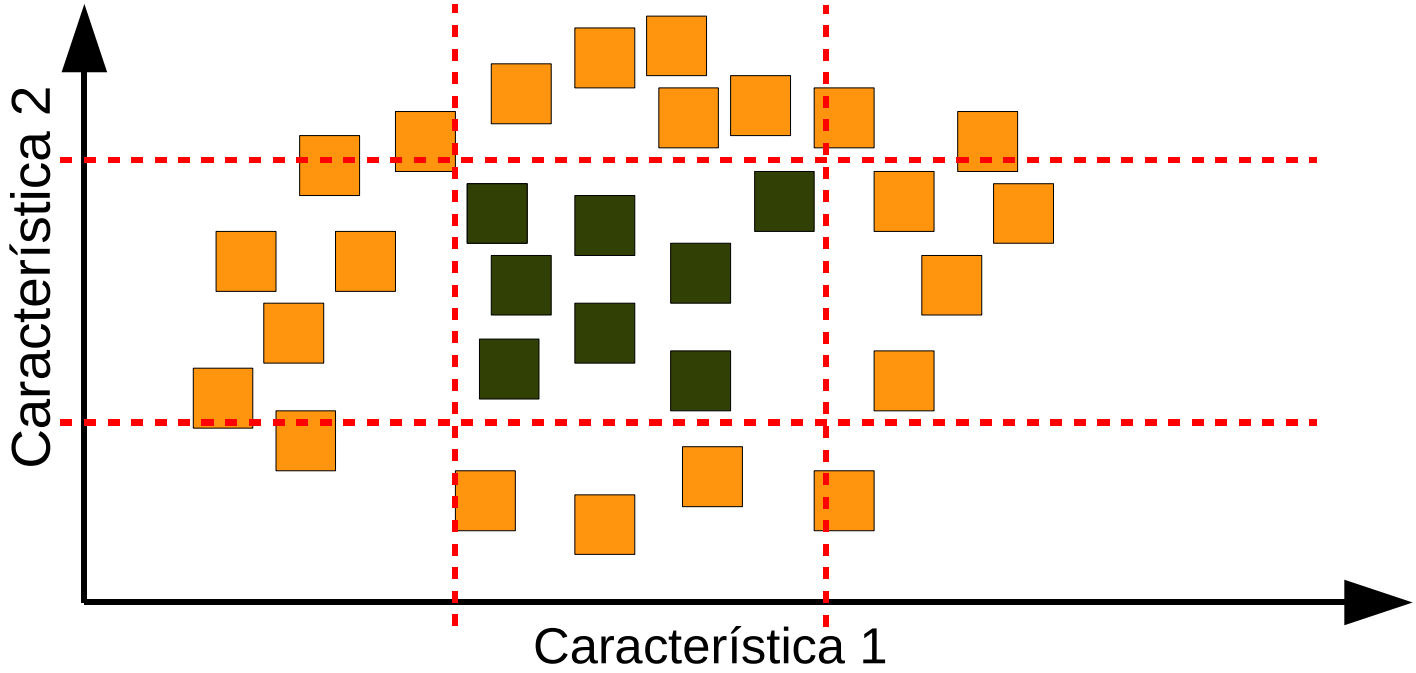



Figura : Exemplo da separação em hiperplanos gerada por uma árvore de decisão

- As divisões, ou testes, podem dividir o conjunto de exemplos de acordo com cada valor de um atributo (discreto), se é maior ou menor que um valor de um atributo (numérico), ou se pertence ou não a um subconjunto de valores de um atributo (discreto)

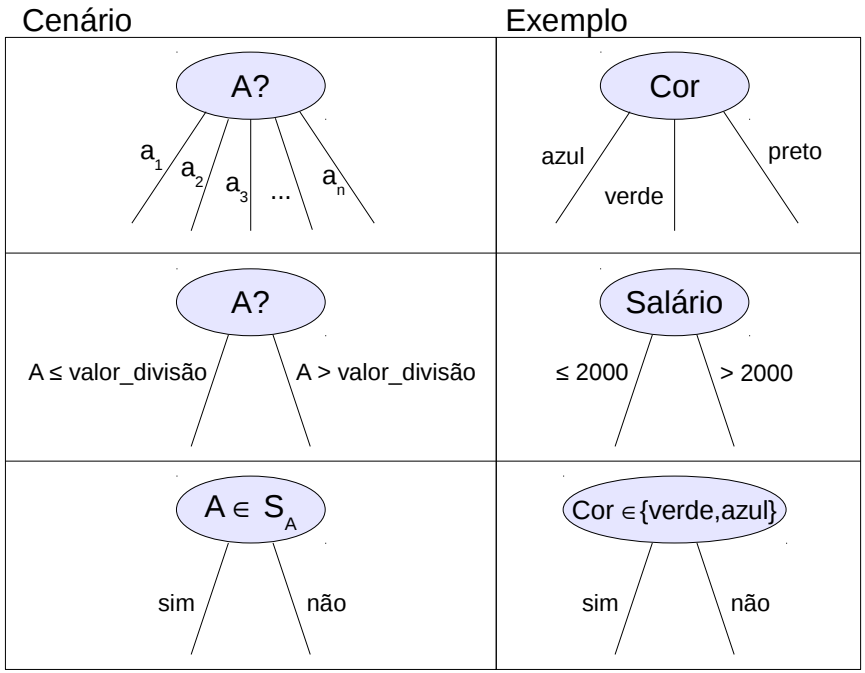



Figura : Exemplos de divisões dos nós de uma árvore

# Indução

Algoritmo **Gerar_árvore_decisão**

- **Entrada**
  - $D$: conjunto de exemplos de treinamento
  - *lista_atributos*: conjunto de atributos candidatos
  - *método_seleção_atributos*: determina o critério de divisão que melhor particiona o conjunto de treinamento em relação às classes
- **Saída**: Árvore de decisão
- **Método**
  1. crie um nó $N$
  2. **se** os exemplos em $D$ são todos da mesma classe $C$ **então**
  3. retorne $N$ como um nó folha rotulado com a classe $C$
  4. **se** *lista_atributos* está vazia **então**
  5. retorne $N$ como um nó folha rotulado com a classe majoritária em $D$

- **Método**
  6. aplique *Método_seleção_atributo*($D$, *lista_atributos*) para **encontrar** o melhor *critério_divisão*
  7. rotule o nó $N$ com o *critério_divisão*
  8. **se** *atributo_divisão* é discreto **e** divisões múltiplas são permitidas **então**
  9. $\quad$ *lista_atributos* $\leftarrow$ *lista_atributos* - *atributo_divisão*
  10. **para cada** valor $j$ do *critério_divisão*
  11. $\quad$ seja $D_j$ os conjunto de exemplos em $D$ que contém $j$
  12. $\quad$ **se** $D_j$ está vazio **então**
  13. $\quad\quad$ atribua um nó folha rotulado com a classe majoritária em $D$ ao nó $N$
  14. $\quad$ **senão**
  15. $\quad\quad$ anexa o nó retornado por *Gerar_árvore_decisão*($D_j$, *lista_atributos*) ao nó $N$
  16. **fim para**
  17. retorne $N$

- A complexidade computacional dado um conjunto de exemplos de treinamento $D$ é $O(n \times |D| \times log(|D|))$, na qual $n$ é o número de atributos e $|D|$ é o número de exemplos de treinamento
- A função `Método_seleção_atributo` determina o critério de divisão
  - Qual atributo será utilizado como teste em um nó da árvore
  - É escolhido o atributo que melhor separa os exemplos considerando as classes
  - Idealmente, após as partições os nós folhas devem ser os mais "puros" possíveis, ou seja, devem haver apenas exemplos de um única classe
  - No caso de atributos contínuos, cada ponto de divisão deve ser considerado

# Critérios para Seleção de Atributos

- Uma medida de seleção de atributos é uma heurística para selecionar um critério de divisão que melhor separa os dados considerando suas classes
- As medidas geram um *ranking* para os atributos, na qual o melhor atributo no *ranking* é escolhido como critério de divisão
- Em um cenário ideal, as partições devem ser "puras"
- O melhor critério de divisão é aquele mais próximo ao cenário ideal

- Para entender as medidas de seleção de atributos apresentadas à seguir, serão utilizados as seguintes notações
  - $D$: conjunto de treinamento
  - $|D|$: quantidade de exemplos do conjunto de treinamento
  - $C$: conjunto de classes, tal que $C = (C_1, C_2, \ldots, C_m)$
  - $C_{i,D}$: conjunto de tuplas da classe $C_i$ em $D$
  - $|C_{i,D}|$: quantidade de exemplos da classe $C_i$ em $D$

# Ganho de Informação

- Usado no algoritmo ID3
- O atributo com maior ganho de informação é escolhido como atributo de divisão para um nó da árvore
- Este atributo é o que minimiza a informação necessária para classificar as tuplas nas partições resultantes – maior "pureza"
- A informação necessária para classificar uma tupla em $D$ é dada por (Entropia)

$$Info(D) = -\sum_{i=1}^{m} p_i log_2(p_i) \quad (1)$$

na qual $p_i$ é a probabilidade, diferente de zero, de uma tupla em $D$ pertencer a classe $C_i$, ou seja $|C_{i,D}|/|D|$

- Caso $p_i = 0$, assume-se que $p_i log_2(p_i) = 0$
- DICA: para quem não têm calculadora que realize cálculos com $log_2$

$$log_2 a = \frac{log_{10} a}{log_{10} 2} = \frac{log_{10} a}{0,301}$$

- Suponha que queremos particionas os exemplos em $D$ considerando um atributo $A$ com $v$ valores distintos, ou seja, $A = \{a_1, a_2, \ldots, a_v\}$
- Se $A$ possui valores discretos, cada um dos valores corresponde as saídas de um nó na árvore de decisão
- O atributo $A$ será utilizado para separar $D$ em $v$ partições, $\{D_1, D_2, \ldots, D_v\}$, na qual $D_j$ contém as tuplas em $D$ cujo valor do atributo $A$ é $a_j$

- Para verificar a informação necessária para classificar um exemplo em $D$ baseado na partição gerada pelo atributo $A$ usa-se

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j) \quad (2)$$

- O termo $\frac{|D_j|}{|D|}$ age como peso para a $j$-ésima partição
- Quanto menor a informação "ainda" requerida, maior a puridade das partições

- O *ganho de informação* é definido como a diferença entre a informação original, isto é, baseada apenas nas proporções das classe, e a informação obtida após o particionamento utilizando o atributo $A$

$$Gain(A) = Info(D) - Info_A(D) \tag{3}$$

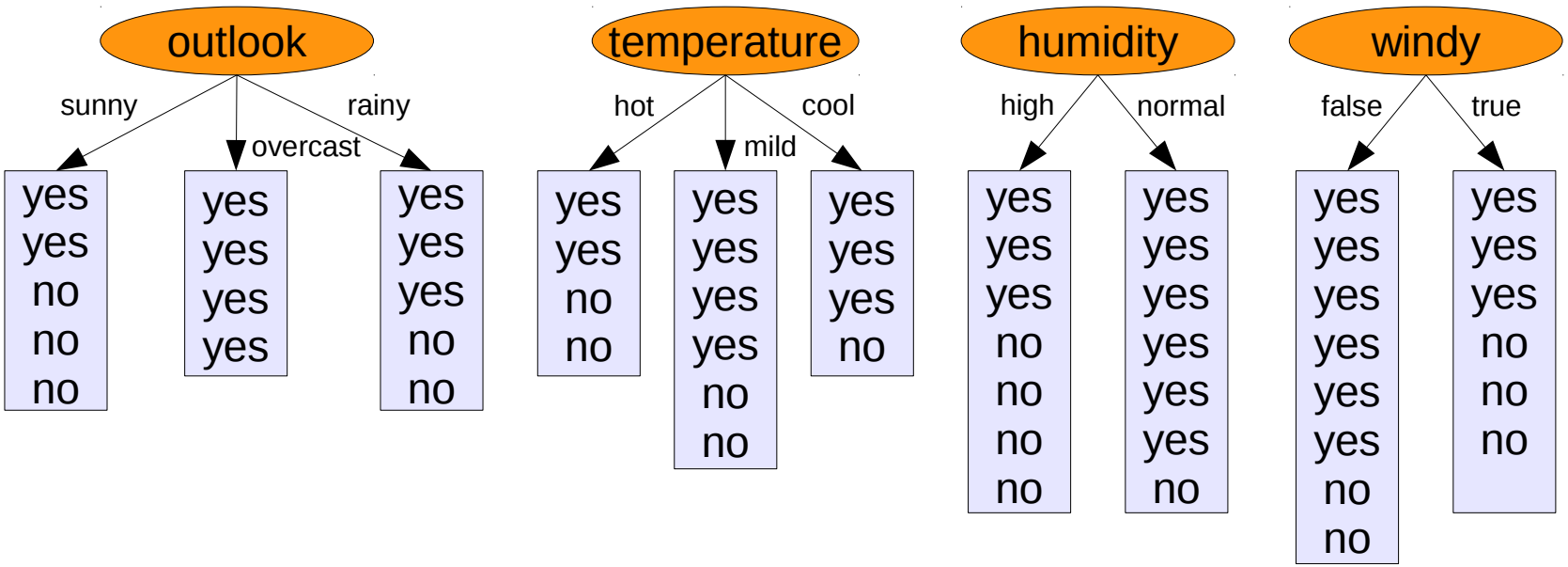
- O atributo $A$ que produz o maior ganho de informação ($Gain(A)$) é escolhido como critério de divisão para o nó $N$

# Exemplo

- Vamos considerar o conjunto de dados *Weather*

Tabela : Conjunto de dados *Weather* [Witten and Frank, 2005].

| **Outlook** | **Temp** | **Humidity** | **Windy** | **Play** |
|---|---|---|---|---|
| Sunny | Hot | High | False | **No** |
| Sunny | Hot | High | True | **No** |
| Overcast | Hot | High | False | **Yes** |
| Rainy | Mild | High | False | **Yes** |
| Rainy | Cool | Normal | False | **Yes** |
| Rainy | Cool | Normal | True | **No** |
| Overcast | Cool | Normal | True | **Yes** |
| Sunny | Mild | High | False | **No** |
| Sunny | Cool | Normal | False | **Yes** |
| Rainy | Mild | Normal | False | **Yes** |
| Sunny | Mild | Normal | True | **Yes** |
| Overcast | Mild | High | True | **Yes** |
| Overcast | Hot | Normal | False | **Yes** |
| Rainy | Mild | High | True | **No** |


outlook
sunny
overcast
rainy
yes yes no no no
yes yes yes yes
yes yes yes no no
temperature
hot
mild
cool
yes yes no no
yes yes yes yes no no
yes yes yes no
humidity
high
normal
yes yes yes no no no no
yes yes yes yes yes yes no
windy
false
true
yes yes yes yes yes yes no no
yes yes yes no no no

- $info(D) = info([9,5]) = -(9/14)log_2(9/14) - (5/14)log_2(5/14) = 0,92$

- $info_{outlook}(D) = info_{outlook}([2,3],[4,0],[3,2]) =$
  $(5/14)info([2,3]) + (4/14)info([4,0]) + (5/14)info([3,2])$
- $info_{outlook}(D) = (5/14)0,94 + (4/14)0 + (5/14)0,94 = 0,64$
- $gain_{outlook}(D) = info(D) - info_{outlook}(D) = 0,92 - 0,64 = 0,28$

- $info_{temperature}(D) = info_{temperature}([2,2],[4,2],[3,1]) =$
  $(4/14)info([2,2]) + (6/14)info([4,2]) + (4/14)info([3,1])$
- $info_{temperature}(D) = (4/14)1 + (6/14)0,9 + (4/14)0,78 = 0,89$
- $gain_{temperature}(D) = info(D) - info_{temperature}(D) = 0,92 - 0,89 = 0,03$

- $info_{humidity}(D) = info_{humidity}([3,4],[6,1]) = (7/14)info([3,4]) + (7/14)info([6,1])$
- $info_{humidity}(D) = (7/14)0,96 + (7/14)0,58 = 0,77$
- $gain_{humidity}(D) = info(D) - info_{humidity}(D) = 0,92 - 0,77 = 0,15$

- $info_{windy}(D) = info(windy)([6,2],[3,3]) = (8/14)info([6,2]) + (6/14)info[3,3]$
- $info_{windy}(D) = (8/14)0,78 + (6/14)1 = 0,87$
- $gain_{windy}(D) = info(D) = info_{windy} = 0,92 - 0,87 = 0,05$

- Portanto, o atributo *outlook* é selecionado como atributo de divisão
- Como não foram selecionados nós anteriormente, o atributo *outlook* será a raiz da árvore de decisão
- Vale ressaltar que a partição gerada por *outlook* = *overcast* gera uma partição pura, portanto, pode-se gerar um nó folha para este ramo

Tabela : Conjunto de dados *Weather* considerando *outlook* = *sunny*

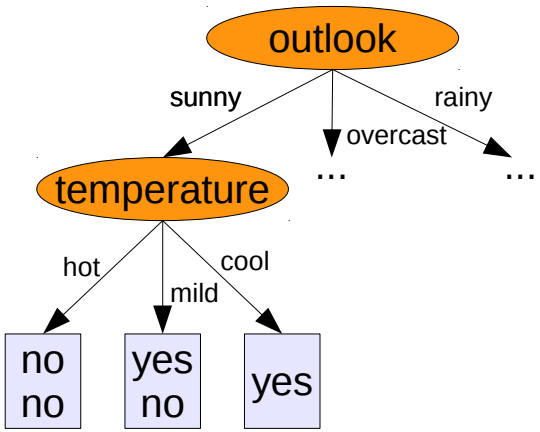
| Outlook | Temp | Humidity | Windy | Play |
|---|---|---|---|---|
| Sunny | Hot | High | False | **No** |
| Sunny | Hot | High | True | **No** |
| Sunny | Mild | High | False | **No** |
| Sunny | Cool | Normal | False | **Yes** |
| Sunny | Mild | Normal | True | **Yes** |

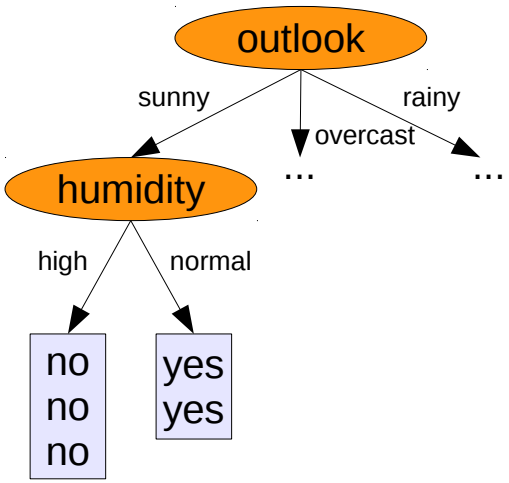
Tabela : Conjunto de dados *Weather* considerando *outlook* = *overcast*.

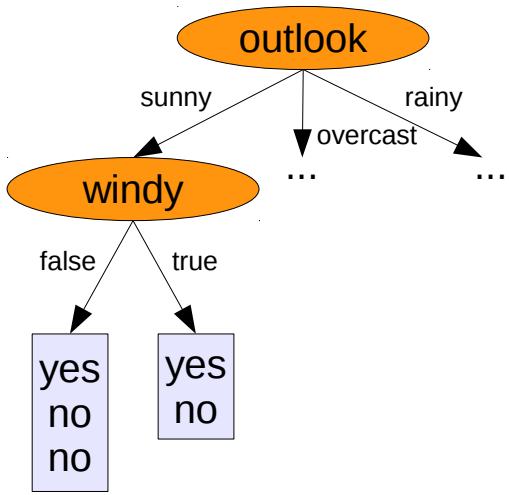
| Outlook | Temp | Humidity | Windy | Play |
|---|---|---|---|---|
| Overcast | Hot | High | False | **Yes** |
| Overcast | Cool | Normal | True | **Yes** |
| Overcast | Mild | High | True | **Yes** |
| Overcast | Hot | Normal | False | **Yes** |

Tabela : Conjunto de dados *Weather* considerando *outlook* = *rainy*.

| Outlook | Temp | Humidity | Windy | Play |
|---|---|---|---|---|
| Rainy | Mild | High | False | **Yes** |
| Rainy | Cool | Normal | False | **Yes** |
| Rainy | Cool | Normal | True | **No** |
| Rainy | Mild | Normal | False | **Yes** |
| Rainy | Mild | High | True | **No** |


outlook
sunny
overcast
rainy
...
...
temperature
hot
mild
cool
no
no
yes
no
yes

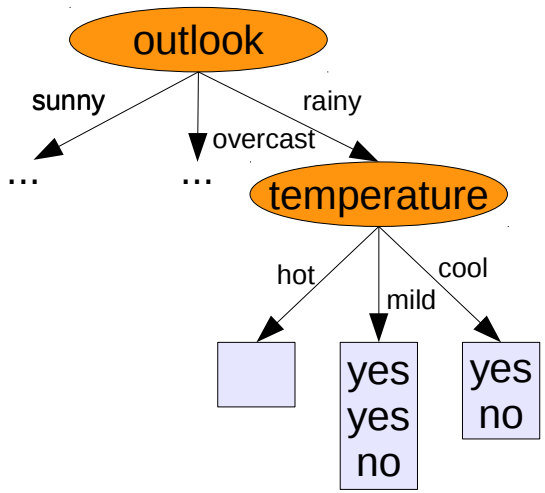

outlook
sunny
overcast
rainy
...
...
humidity
high
normal
no
no
no
yes
yes

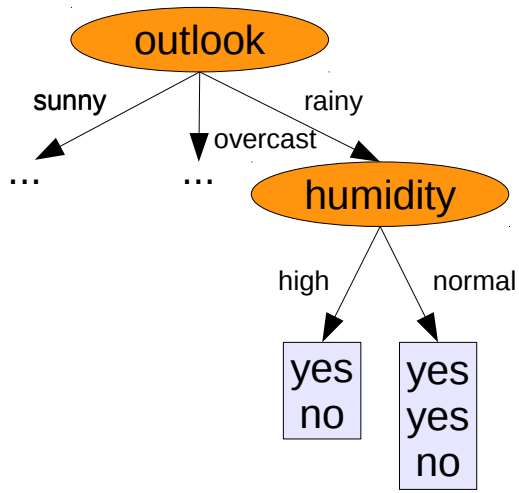

outlook
sunny
overcast
rainy
...
...
windy
false
true
yes
no
no
yes
no

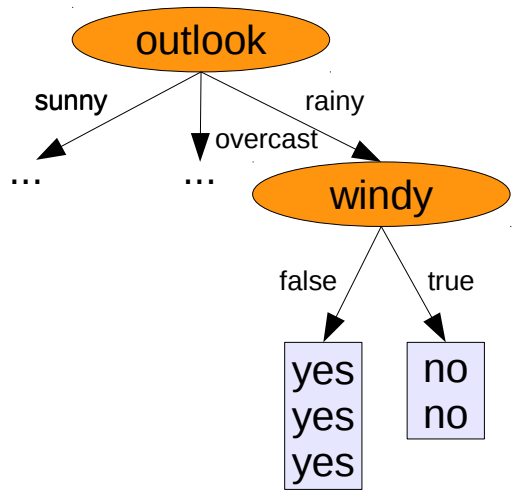
- $info(D_{outlook=sunny}) = info([2,3]) = 0,94$
- $info_{temperature}(D_{outlook=sunny}) = info_{temperature}([0,2],[1,1],[1,0]) =$ $(2/5)info([0,2]) + (2/5)info([1,1]) + (1/5)info(1,0)$
- $info_{temperature}(D_{outlook=sunny}) = (2/5)0 + (2/5)1 + (1/5)0 = 0,4$
- $gain_{temperature}(D_{outlook=sunny}) = info(D_{outlook=sunny}) - gain_{temperature}(D_{outlook=sunny}) = 0,54$
- $info_{humidity}(D_{outlook=sunny}) = info_{humidity}([0,3],[2,0])$
- $info_{humidity}(D_{outlook=sunny}) = (3/5)info([0,3]) + (2/5)info([2,0]) = 0$
- $gain_{humidity}(D_{outlook=sunny}) = info(D_{outlook=sunny}) - gain_{humidity}(D_{outlook=sunny}) = 0,94$

...

- O atributo *humidity* apresenta o maior ganho de informação dado $outlook = sunny$
- Vale ressaltar que para $outlook = sunny$, as divisões do atributo *humidity* geram partições puras, portanto, criam-se nós folhas para estes ramos

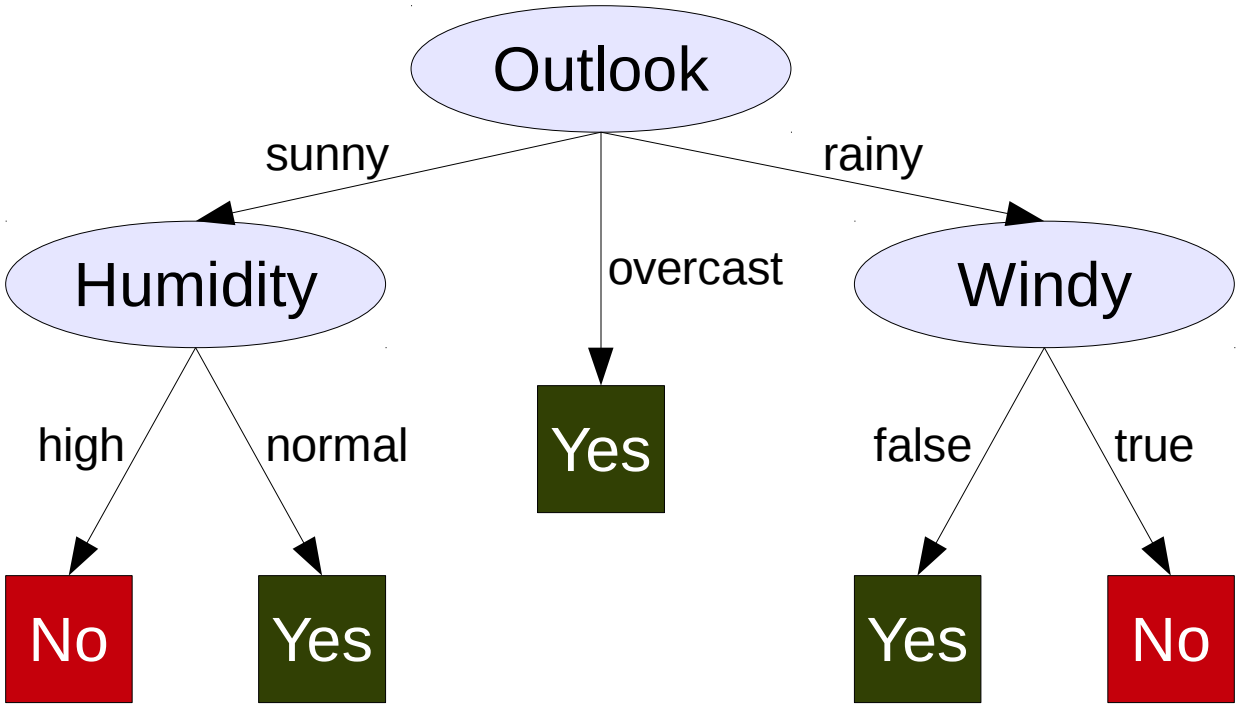

outlook
sunny
overcast
rainy
...
...
temperature
hot
mild
cool
yes
yes
no
yes
no


outlook
sunny
overcast
rainy
...
...
humidity
high
normal
yes
no
yes
yes
no


outlook
sunny
overcast
rainy
...
...
windy
false
true
yes
yes
yes
no
no

- O atributo *windy* apresenta o maior ganho de informação dado *outlook* = *rainy*
- Vale ressaltar que para *outlook* = *rainy*, as divisões do atributo *windy* geram partições puras, portanto, criam-se nós folhas para estes ramos

# Árvore de decisão final

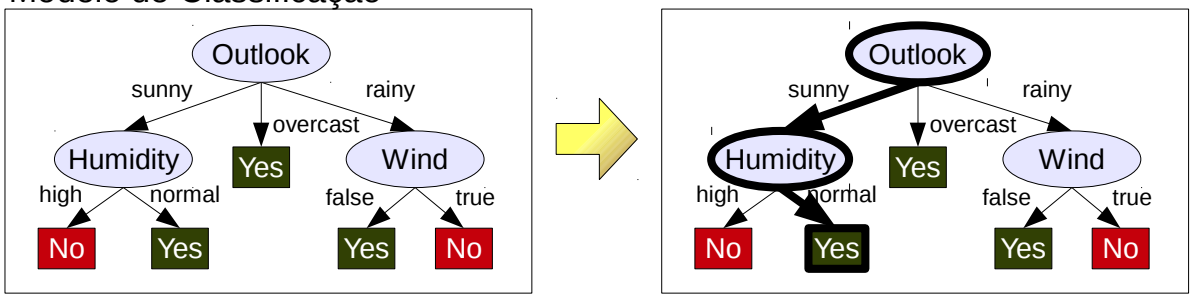


# Classificação

- Dado uma instância $X$, na qual a classe é desconhecida, os valores dos atributos são testado nos nós das árvore de decisão
- Um caminho é traçado da raiz à um nó folha, que representa a classe predita pela árvore de decisão

| Outlook | Temp | Humidity | Windy | Classe |
|---|---|---|---|---|
| Sunny | Mild | Normal | True | ???? |

Modelo de Classificação




| | | | | |
|---|---|---|---|---|
| Sunny | Mild | Normal | True | Yes |

# Referências Bibliográficas I


Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984).
*Classification and Regression Trees*.
Wadsworth.


Han, J., Kamber, M., and Pei, J. (2011).
*Data Mining: Concepts and Techniques*.
The Morgan Kaufmann Series in Data Management Systems.
Elsevier.


Quinlan, J. R. (1986).
Induction of decision trees.
*Mach. Learn.*, 1(1):81–106.

# Referências Bibliográficas II


Quinlan, J. R. (1993).
*C4.5: programs for machine learning*.
Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.


Tan, P.-N., Steinbach, M., and Kumar, V. (2005).
*Introduction to Data Mining*.
Addison-Wesley.


Witten, I. H. and Frank, E. (2005).
*Data Mining: Practical machine learning tools and techniques*.
Morgan Kaufmann, 2 edition.