## Introdução à Inteligência Artificial

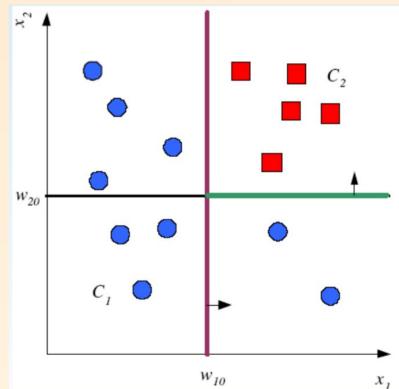
- Árvores de Decisão
- Exemplos
- Algoritmos ID3, C4.5 e variações
- Exemplos



Imagem: pixabay.com

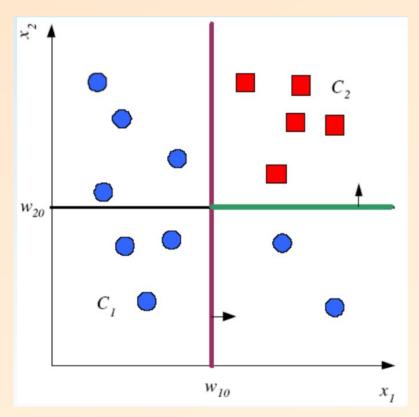
# Árvores para tomada de decisão

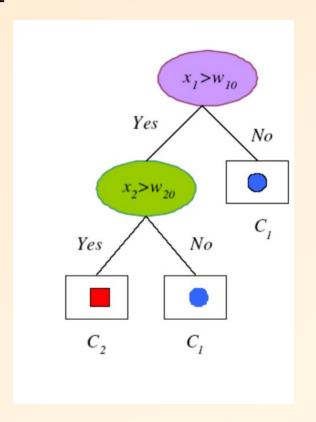
- Sejam exemplos com duas variáveis x₁ e x₂.
- Como decidir entre classes  $C_1 e C_2$ ?



# Árvores para tomada de decisão

Como decidir entre classes C₁ e C₂?





- Dados:
  - X é um espaço de instâncias definido como  $\{X_i\}_{i \in I}$  vone  $X_i$  é uma variável discreta/contínua.

- Dados:
  - X um espaço de instâncias definido como  $\{X_i\}_{i \in I}$  vone  $X_i$  é uma variável discreta/contínua.
  - Y um conjunto finito de classe.

- Dados:
  - X um espaço de instâncias definido como  $\{X_i\}_{i \in I}$  vone  $X_i$  é uma variável discreta/contínua.
  - Y um conjunto finito de classe.
  - Dados de treinamento  $D \subseteq X \times Y$ .

- Dados:
  - X um espaço de instâncias definido como  $\{X_i\}_i$  $\in_{1..N}$  one  $X_i$  é uma variável discreta/contínua.
  - Y um conjunto finito de classe.
  - Dados de treinamento  $D \subseteq X \times Y$ .
- Encontrar:
  - Classe  $y \in Y$  de uma instância  $x \in X$ .

- Uma árvore de decisão é uma árvore onde:
  - Cada nó interior testa uma variável.

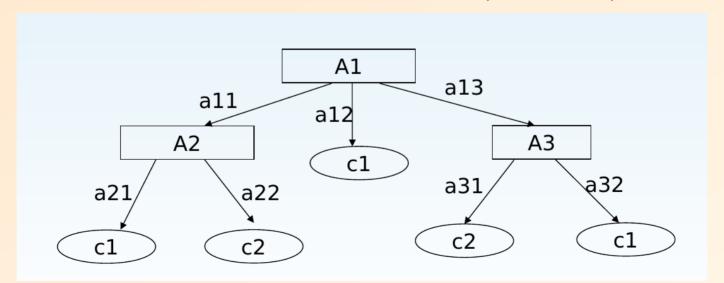
dibio @ unb.br

- Uma árvore de decisão é uma árvore onde:
  - Cada nó interior testa uma variável.
  - Cada ramo corresponde a um valor de variável.

- Uma árvore de decisão é uma árvore onde:
  - Cada nó interior testa uma variável.
  - Cada ramo corresponde a um valor de variável.
  - Cada nó folha é rotulado com uma classe (nó classe).

dibio @ unb.br

- Uma árvore de decisão é uma árvore onde:
  - Cada nó interior testa uma variável.
  - Cada ramo corresponde a um valor de variável.
  - Cada nó folha é rotulado com uma classe (nó classe).



# Para pensar e resolver (pegar lápis e papel)

• Em um teste de classificação (binária), de 165 amostras a tabela a seguir mostra os resultados de acertos das 2 classes do sistema.

n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

Calcule Precisão, Revocação e medida F1.

## Para pensar e resolver (pegar lápis e papel)

n=165	Predicted: NO	Predicted: YES
Actual: NO	TN 50	FP 10
Actual: YES	<sub>5</sub> FN	TP 100

$$Precision = \frac{TP}{TP + FP}$$
  $TP = True positive$   $TN = True negative$   $TP = True negative$   $TN = True negative$   $TP = True negative$   $TP = True negative$   $TP = True negative$   $TP = True positive$   $TP = True positive$ 

Seja o seguinte conjunto de dados (Mitchell, 1997)

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

15

Qual seria uma árvore de decisão possível?

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

- Qual seria uma árvore de decisão possível?
- Raiz?
- Nós folhas?

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

Qual atributo (raiz) escolher?



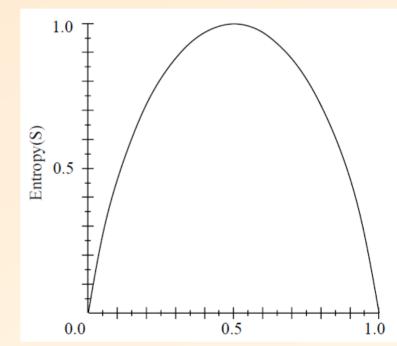
- Qual atributo (raiz) escolher?
  - Aquele que proporcionar a menor árvore

dibio @ unb.br

- Qual atributo (raiz) escolher?
  - Aquele que proporcionar a menor árvore
  - Aquele que oferecer a maior informação para decisão

## Entropia para medir informação

- Seja S uma amostra de exemplos de treinamento
- p+ a proporção de exemplos +
- p- a proporção de exemplos -
- Entropia mede a impureza de S



dibio @ unb.br

$$Entropy(S) = -p_{+} \log(p_{+}) - p_{-} \log(p_{-})$$

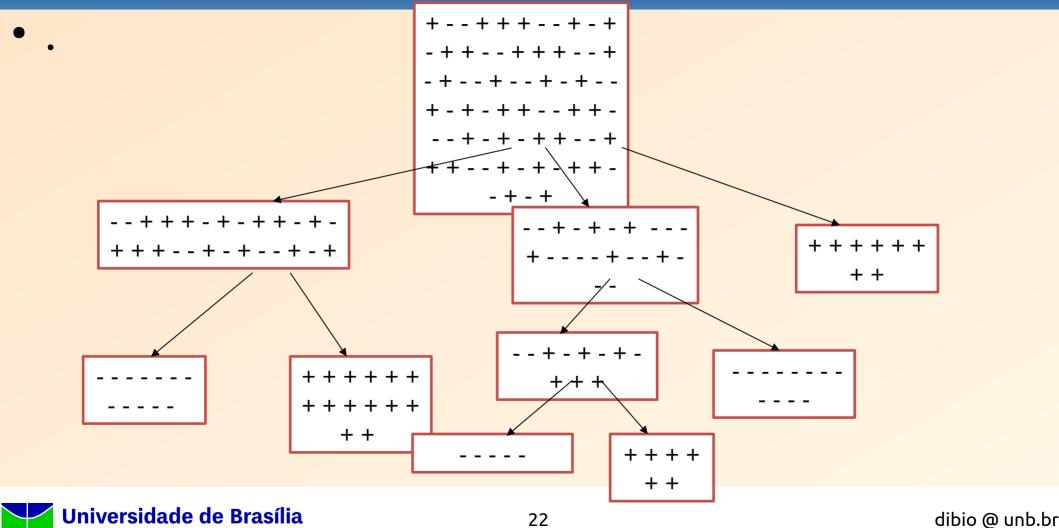


## Entropia para medir informação

- Entropia alta, muita desorganização, requer mais informação
- Entropia baixa, muita organização, requer pouca informação

dibio @ unb.br

# Entropia para medir informação



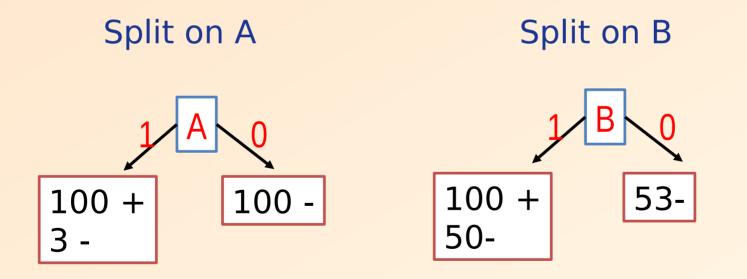
## Ganho de Informação

 Ganho de informação seria uma redução esperada na entropia, mudando-se a escolha do nó

dibio @ unb.br

## Ganho de Informação

 Ganho de informação seria uma redução esperada na entropia, mudando-se a escolha do nó



## Ganho de Informação

 Ganho de informação seria uma redução esperada na entropia, mudando-se a escolha do nó

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

Entropia (S)

$$-\frac{9}{14}\log(\frac{9}{14})$$

$$-\frac{5}{14}\log(\frac{5}{14})$$

= 0.94

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	No
Day1 Day2	Sunny	Hot	High	Strong	No No
_	-		· ·		
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

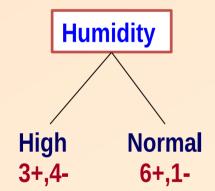
 Testando atributos para maior ganho

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

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

 Testando atributos para maior ganho

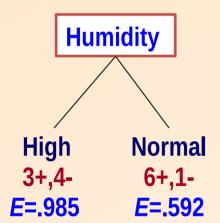
$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$



Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

 Testando atributos para maior ganho

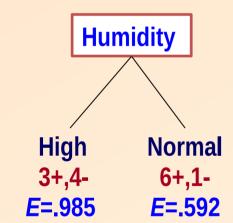
$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$



Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

 Testando atributos para maior ganho

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

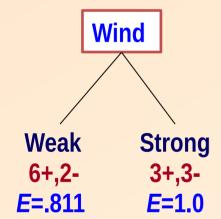


Day	Outlook	Temperature	Humidity	Wind	Play
					Tennis
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

*Gain(S,* Humidity) = .94 - 7/14 \* 0.985 - 7/14 \*.592 = **0.151** 

 Testando atributos para maior ganho

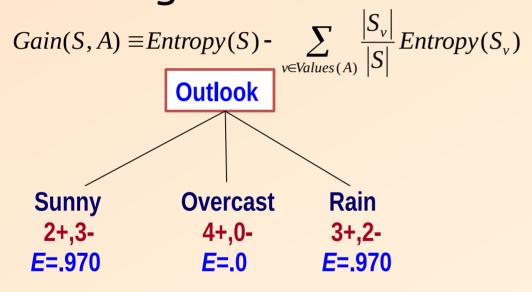
$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$



Day	Outlook	Temperature	Humidity	Wind	Play
					Tennis
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

Gain(S, Wind) = .94 - 8/14 \* 0.811 - 6/14 \*1.0 = 0.048

 Testando atributos para maior ganho

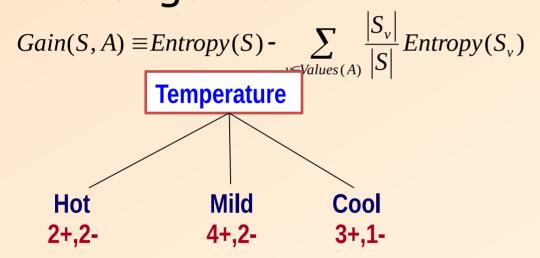


Day	Outlook	Temperature	Humidity	Wind	Play
					Tennis
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

*Gain(S, Outlook) = 0.246* 

33

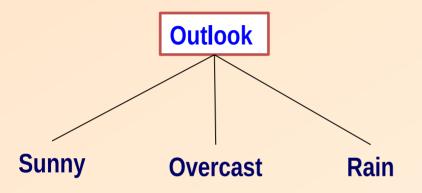
 Testando atributos para maior ganho



Day	Outlook	Temperature	Humidity	Wind	_Play
					Tennis
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

*Gain(S,* Outlook) = 0.029

Qual usar? Maior ganho de informação



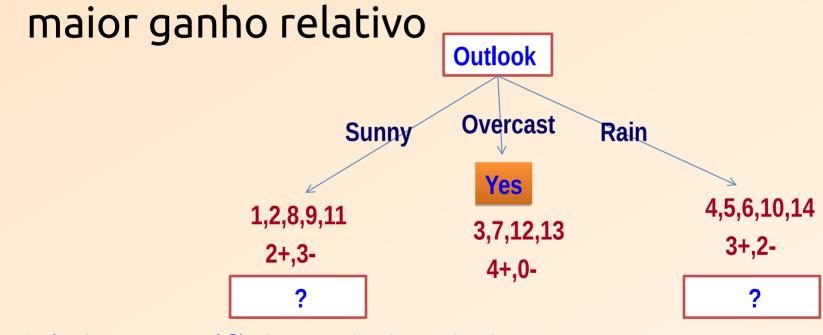
**Gain(S, Humidity) = 0.151** 

Gain(S, Wind) = 0.048

Gain(S, Temperature) = 0.029

Gain(S, Outlook) = 0.246

• Depois, continuar incluindo próximos atributos com



Gain ( $S_{sunny}$ , Humidity) = .97-(3/5) \* 0-(2/5) \* 0 = .97 Gain ( $S_{sunny}$ , Temp) = .97- 0-(2/5) \*1 = .57

Gain ( $S_{\text{sunny}}$ , Wind) = .97-(2/5) \*1 - (3/5) \*.92 = .02



• Até folhas... **Outlook Overcast** Sunny Rain Yes **Humidity** High Normal No Yes

 Até folhas... **Outlook Overcast** Sunny Rain Yes Wind **Humidity** Strøng Weak High Normal Yes No No Yes

dibio @ unb.br

# Algoritmos de Árvores de decisão

• Este apresentado é o ID3 (Iterative Dichotomiser 3)

Quinlan, J. R. 1986. Induction of Decision Trees. *Machine Learning* 1, 1 (Mar. 1986), 81–106

dibio @ unb.br

# Algoritmos de Árvores de decisão

• Este apresentado é o ID3 (Iterative Dichotomiser 3)

Quinlan, J. R. 1986. Induction of Decision Trees. *Machine Learning* 1, 1 (Mar. 1986), 81–106

- 1) Calcular a entropia de cada atributo a do conjunto S.
- 2) Particionar o conj. S em subconjuntos de forma a obter um ganho de informação maior possível (por pares)
- 3) Escolher o nó equivalente com o maior ganho.
- 4) Fazer recursivamente nos nós/atributos restantes até folhas.

#### ID3

ID3(Examples, Target\_attribute, Attributes)

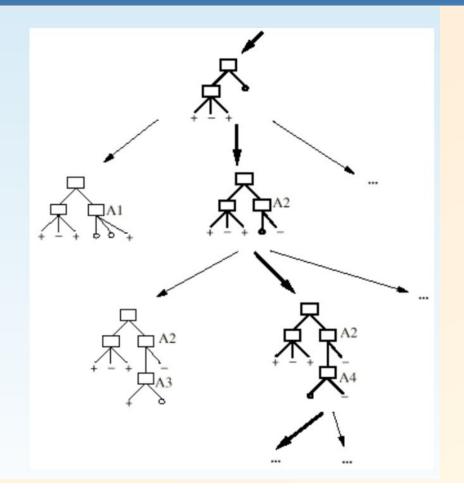
Examples are the training examples. Target\_attribute is the attribute whose value is to be predicted by the tree. Attributes is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given Examples.

- · Create a Root node for the tree
- If all Examples are positive, Return the single-node tree Root, with label = +
- If all Examples are negative, Return the single-node tree Root, with label = -
- If Attributes is empty, Return the single-node tree Root, with label = most common value of Target\_attribute in Examples
- · Otherwise Begin
  - A ← the attribute from Attributes that best\* classifies Examples
  - The decision attribute for Root ← A
  - For each possible value, vi, of A,
    - Add a new tree branch below Root, corresponding to the test A = v<sub>i</sub>
    - Let Examples<sub>vi</sub> be the subset of Examples that have value v<sub>i</sub> for A
    - · If Examples, is empty
      - Then below this new branch add a leaf node with label = most common value of Target\_attribute in Examples
      - Else below this new branch add the subtree
         ID3(Examples<sub>v.</sub>, Target\_attribute, Attributes {A}))

- End
- · Return Root

#### ID3

- A função de avaliação é o ganho de informação.
- ID3 mantém somente uma árvore de decisão.
- ID3 usa todos os exemplos de treinamento a cada passo da busca.
- · Não faz backtracking.





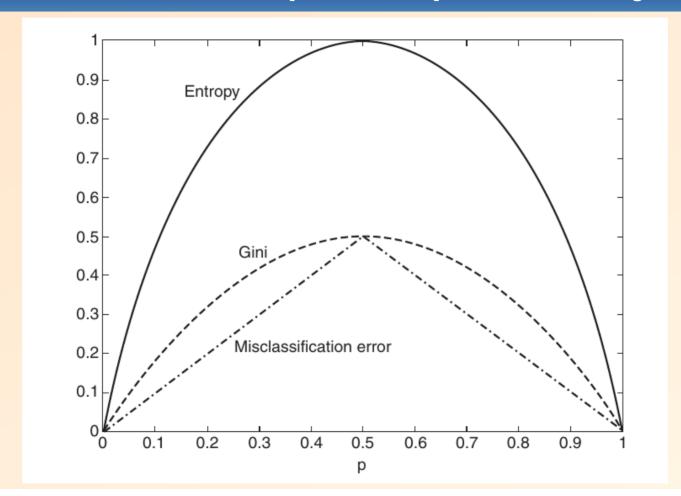
41

# Medidas de impureza/informação

$$\operatorname{Entropy}(t) = -\sum_{i=0}^{c-1} p(i|t) \log_2 p(i|t),$$
 
$$\operatorname{Gini}(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2,$$
 
$$\operatorname{Classification\ error}(t) = 1 - \max_i [p(i|t)],$$

where c is the number of classes and  $0 \log_2 0 = 0$  in entropy calculations.

# Medidas de impureza/informação





# Algoritmos de Árvores de decisão

 O próprio Quinlan lançou melhorias, como o C4.5 em 1993

Quinlan, J. R. C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993

- Permitindo uso de atributos numéricos
- Lidando com valores incompletos
- Poda em atributos com pouco ganho e com ruído

## Podas em árvores de decisão/classificação

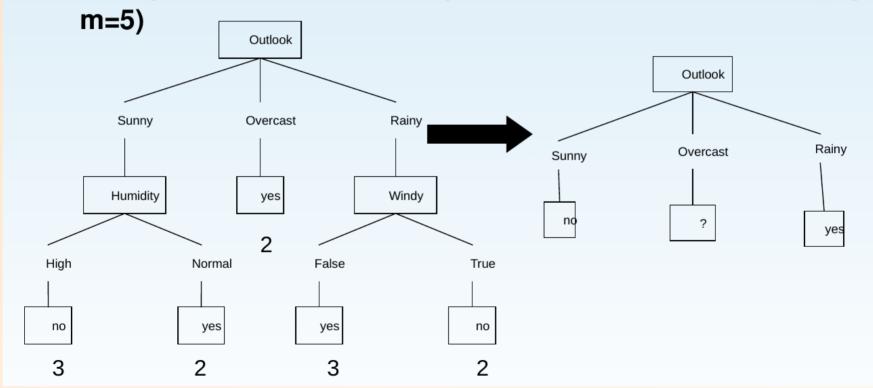
Podas em geral são realizadas para evitar sobreajuste (overfitting)

45

- Anterior (Pré)
- Posterior (Pós)

# Exemplo: Poda anterior (pré)

 Parar crescimento quando os nós folhas receberem menos exemplos de treinamento que um determinado valor. (e.g.



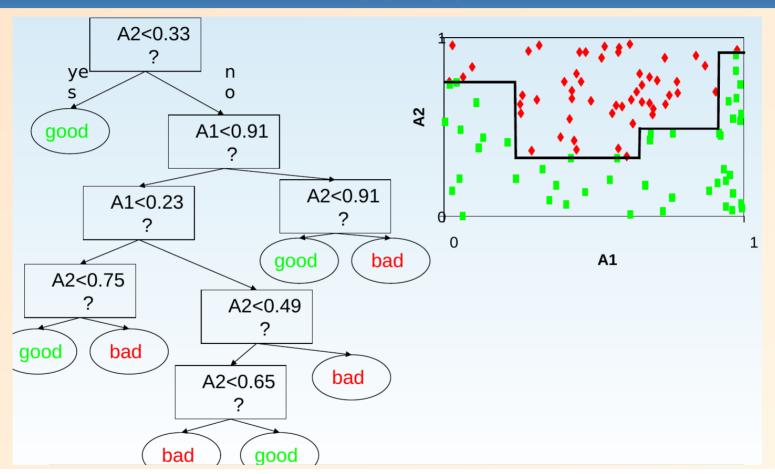


#### C4.5

 O C4.5 usa como critério de escolha do nó a razão de ganho

$$\begin{aligned} & Gain(S,A) \equiv \frac{Gain(S,A)}{SplitInformation(S,A)} \\ & SplitInformation(S,A) \equiv -\sum_{i=1}^{c} \frac{\left|S_{i}\right|}{\left|S\right|} \log_{2} \frac{\left|S_{i}\right|}{\left|S\right|} \end{aligned}$$

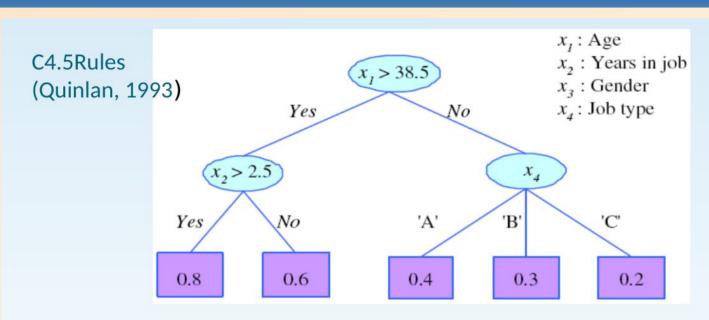
# Arvores de decisao sao classificadores nao lineares



# Árvores de decisão (ID3, C4.5, CART)

- Estratégia dividir e conquistar
  - Nós internos de decisão
    - Uma variável: Usa um único atributo, x,
      - Numérico  $x_i$ : Divisão binária :  $x_i > w_m$
      - Discreto x; : divisão n-way para n valores possíveis
    - Multivariada: Usa todos atributos, x
  - Folhas
    - Classificação: Rótulos das classes, ou propoções
    - Regressão: Numérico; r médio, ou ajuste local
  - Aprendizagem é gulosa; encontrar a melhor divisão recursivamente (Breiman et al, 1984; Quinlan, 1986, 1993)

## Regras de produção extraídas de árvores



- R1: IF (age>38.5) AND (years-in-job>2.5) THEN y = 0.8
- R2: IF (age>38.5) AND (years-in-job  $\leq$  2.5) THEN y = 0.6
- R3: IF (age  $\leq$  38.5) AND (job-type='A') THEN y = 0.4
- R4: IF (age  $\leq$  38.5) AND (job-type='B') THEN y = 0.3
- R5: IF (age  $\leq$  38.5) AND (job-type='C') THEN y = 0.2

## Aprendizagem de regras

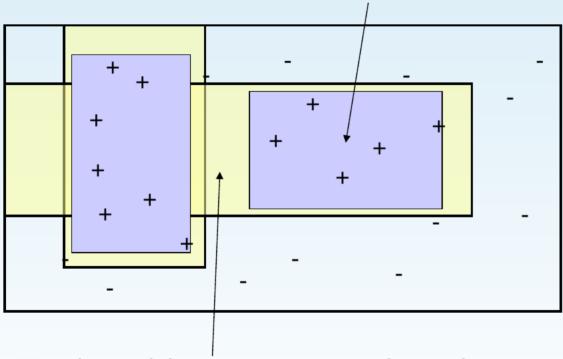
- Indução de regras é semelhante à indução em árvores, mas
  - Indução em árvores é BFS,
  - Indução de regras é DFS; uma regra por vez
- Conjunto de regras contém regras; regras são conjunções de termos
- Regra cobre um exemplo se todos os termos da regra avaliam para T (True) para o exemplo
- IREP (Fürnkrantz and Widmer, 1994), Ripper (Cohen, 1995)

## Aprendizagem de regras

- Duas estratégias:
  - Top-down:
    - Inicie com regra mais geral
    - Adicione literais (variáveis) uma por uma
  - Bottom-up:
    - -Inicie com a mais específica
    - -Remova literais (variáveis) uma por uma

# Aprendizagem de regras

Bottom-up: tipicamente regras mais específicas



Top-down: tipicamente regras mais gerais

- 1. Alternate: whether there is a suitable alternative restaurant nearby.
- 2. Bar: whether the restaurant has a comfortable bar area to wait in.
- 3. Fri/Sat: true on Fridays and Saturdays.
- 4. Hungry: whether we are hungry right now.
- 5. Patrons: how many people are in the restaurant (values are None, Some, and Full).
- 6. *Price*: the restaurant's price range (\$, \$\$, \$\$\$).
- 7. Raining: whether it is raining outside.
- 8. *Reservation*: whether we made a reservation.
- 9. *Type*: the kind of restaurant (French, Italian, Thai, or burger).
- 10. *WaitEstimate*: host's wait estimate: 0–10, 10–30, 30–60, or >60 minutes.

A set of 12 examples, taken from the experience of one of us (SR), is shown in Figure 19.2. Note how skimpy these data are: there are  $2^6 \times 3^2 \times 4^2 = 9,216$  possible combinations of values for the input attributes, but we are given the correct output for only 12 of them; each of the other 9,204 could be either true or false; we don't know. This is the essence of induction: we need to make our best guess at these missing 9,204 output values, given only the evidence of the 12 examples.

dibio @ unb.br

Example	Input Attributes										Output
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Туре	Est	WillWait
$\mathbf{x}_1$	Yes	No	No	Yes	Some	\$\$\$	No	Yes	French	0–10	$y_1 = Yes$
$\mathbf{x}_2$	Yes	No	No	Yes	Full	\$	No	No	Thai	<i>30–60</i>	$y_2 = No$
<b>X</b> 3	No	Yes	No	No	Some	\$	No	No	Burger	0 - 10	$y_3 = Yes$
$\mathbf{x}_4$	Yes	No	Yes	Yes	Full	\$	Yes	No	Thai	10-30	$y_4 = Yes$
$\mathbf{x}_5$	Yes	No	Yes	No	Full	\$\$\$	No	Yes	French	>60	$y_5 = No$
$\mathbf{x}_6$	No	Yes	No	Yes	Some	\$\$	Yes	Yes	Italian	0 - 10	$y_6 = Yes$
<b>X</b> 7	No	Yes	No	No	None	\$	Yes	No	Burger	0 - 10	$y_7 = No$
$\mathbf{x}_8$	No	No	No	Yes	Some	\$\$	Yes	Yes	Thai	0 - 10	$y_8 = Yes$
<b>X</b> 9	No	Yes	Yes	No	Full	\$	Yes	No	Burger	>60	$y_9 = No$
$\mathbf{x}_{10}$	Yes	Yes	Yes	Yes	Full	\$\$\$	No	Yes	Italian	10-30	$y_{10} = No$
$\mathbf{x}_{11}$	No	No	No	No	None	\$	No	No	Thai	0 - 10	$y_{11} = No$
$\mathbf{x}_{12}$	Yes	Yes	Yes	Yes	Full	\$	No	No	Burger	<i>30–60</i>	$y_{12} = Yes$

Figure 19.2 Examples for the restaurant domain.

#### Leitura

Ler capítulo 19 do livro (Russell & Norvig, 2020)

dibio @ unb.br

#### Referências

- Russell, S. & Norvig, P. Artificial Intelligence: a modern approach (4th ed), Pearson, 2020.
- Mitchell, T. Machine Learning, McGraw-Hill, 1997.
- Géron, A. Hands-On Machine Learning with Scikit-Learn & TensorFlow, O'Reilly Inc., 2017.