

# **Inteligência Artificial**

## **Aprendizado supervisionado 2**

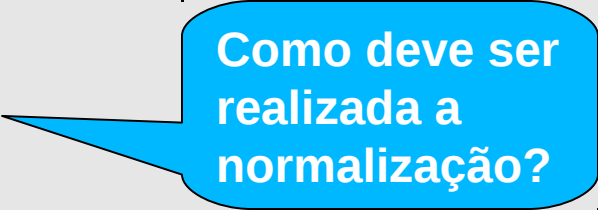
# **Avaliação de algoritmos**

# Avaliação de Algoritmos

- Como já mencionado, o conjunto de dados é usualmente dividido em:
  - Conjunto de treinamento
    - Erro aparente
  - Conjunto de teste
    - Erro verdadeiro (se o conjunto de teste é grande o suficiente)
- \* Utilizar toda a base de dados para o treinamento do classificador que será utilizado na prática

# Avaliação de Algoritmos

- Como já mencionado, o conjunto de dados é usualmente dividido em:
    - Conjunto de treinamento
      - Erro aparente
    - Conjunto de teste
      - Erro verdadeiro (se o conjunto de teste é grande o suficiente)
- \* Utilizar toda a base de dados para o treinamento do classificador que será utilizado na prática

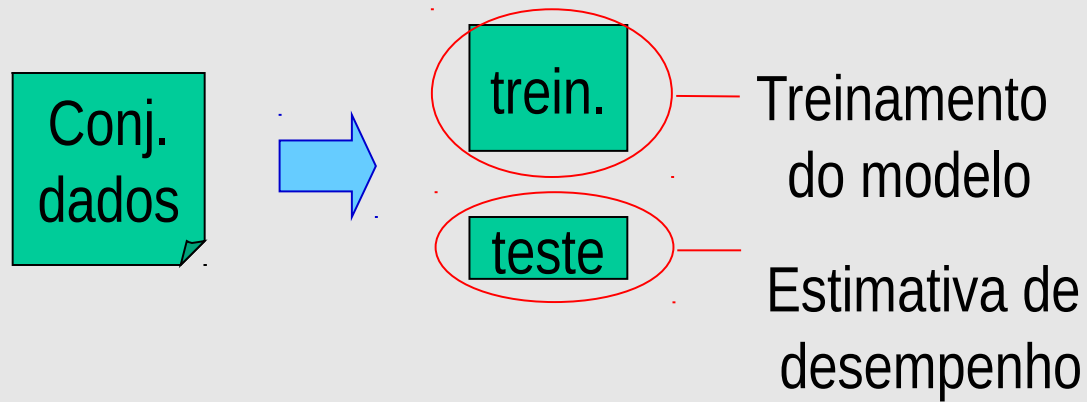


Como deve ser realizada a normalização?

# Holdout

- O conjunto de dados é dividido aleatoriamente em:
  - $p\%$  para treinamento
  - $(1-p)\%$  para teste
- Para tornar os resultados menos dependentes da partição feita, faz-se diversas partições e se dá uma média de desempenho em holdout

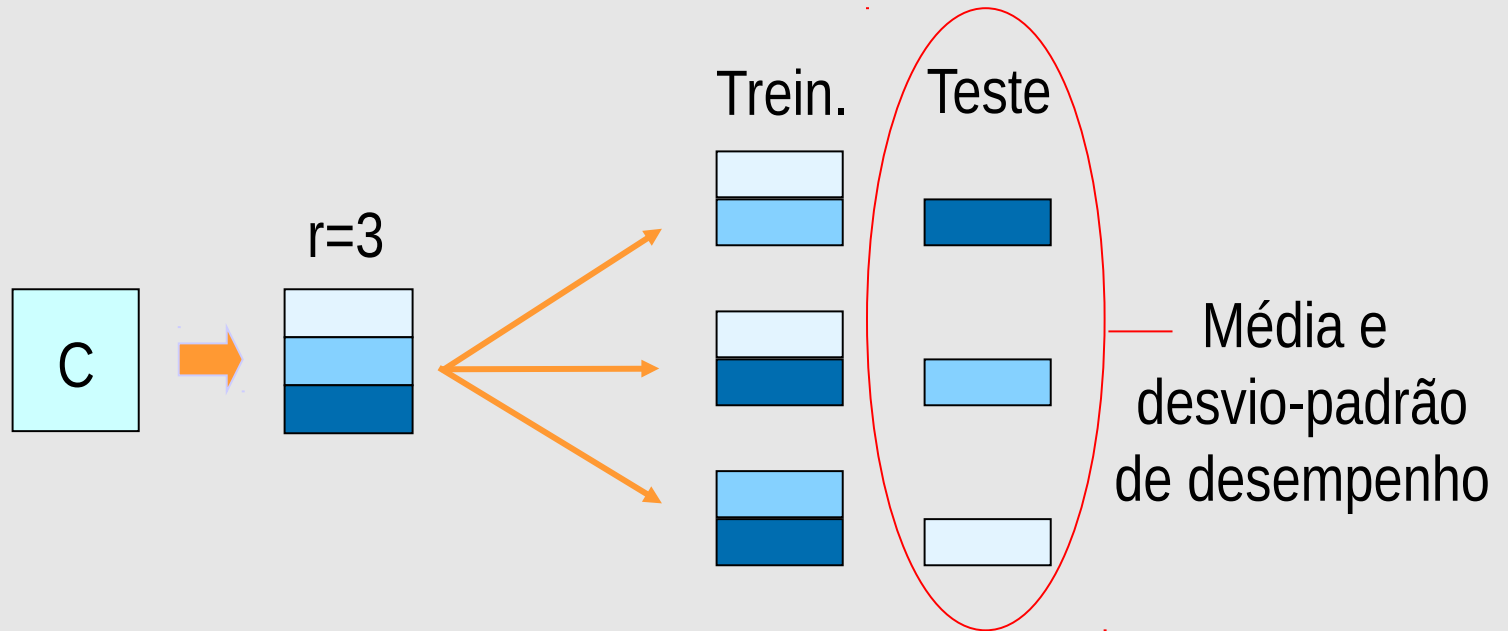
# Holdout



# Validação cruzada

- Validação cruzada – *k-fold cross-validation*
  - Conjunto de dados é aleatoriamente dividido em  $k$  partições exclusivas de tamanho aproximadamente igual
  - Os dados de  $k-1$  partições são usados para treinamento e a partição restante é usada para teste
  - Este processo é repetido  $k$  vezes
    - Considerando cada partição para teste
  - Erro é dado pela média nas partições

# Validação cruzada





# Validação cruzada

- Valor de  $k$  mais usado: 10
- Variação: estratificada
  - Manter a distribuição de classes em cada partição
  - Exemplo: se conjunto de dados original tem 20% na classe  $C_1$  e 80% na classe  $C_2$ , cada partição também deve manter essa proporção

# Leave-one-out

- Caso especial de CV em que  $k = N$ 
  - É computacionalmente caro
    - Usado geralmente para conjuntos de dados pequenos
  - Erro final é a soma dos erros cometidos para cada dado de teste individual

# Matriz de confusão

- Oferece uma medida da efetividade do modelo de classificação  $h$ 
  - Mostra o número de classificações reais contra as classificações preditas, em cada classe
  - Os resultados são sumarizados em uma matriz de duas dimensões
    - Classes verdadeiras x Classes preditas

# Matriz de confusão

Class Label	predicted $C_1$	predicted $C_2$	$\cdots$	predicted $C_k$
true $C_1$	$M(C_1, C_1)$	$M(C_1, C_2)$	$\cdots$	$M(C_1, C_k)$
true $C_2$	$M(C_2, C_1)$	$M(C_2, C_2)$	$\cdots$	$M(C_2, C_k)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
true $C_k$	$M(C_k, C_1)$	$M(C_k, C_2)$	$\cdots$	$M(C_k, C_k)$

$$M(C_i, C_j) = \sum_{\{ \forall (x,y) \in T : y = C_i \}} |h(x) = C_j|$$

# Matriz de confusão

- Número de predições corretas: diagonal da matriz
  - Outros elementos correspondem números de erros

# Média e desvio-padrão

- Seja um dos métodos de amostragem
  - $k$ -fold CV, por ser o mais utilizado
- Um indutor  $A$  gerará  $k$  hipóteses  $h_1, h_2, \dots, h_r$ 
  - E cada hipótese terá uma taxa de erro, medida em no  $i$ -ésimo fold
  - A média e desvio-padrão do desempenho de  $A$  são então dados por:

$$med(A) = \frac{1}{r} \sum_{i=1}^r er(h_i) \quad dp(A) = \sqrt{\left[ \frac{1}{r-1} \sum_{i=1}^r (er(h_i) - med(A))^2 \right]}$$

# Média e desvio-padrão

## ■ Exemplo:

- Em 10-fold CV, A obteve os erros:
  - (5,5; 11,40; 12,70; 5,20; 5,90; 11,30; 10,90; 11,20; 4,90; 11,00)
  - Temos então:

$$med(A) = \frac{90}{10} = 9 \qquad dp(A) = \sqrt{\frac{1}{9} 90,30} = 3,17$$

# Desvio-padrão

- O **desvio-padrão** pode ser visto como uma imagem da **robustez do algoritmo**
  - Se os erros dos preditores produzidos pelo algoritmo A **variam muito**, o algoritmo **não é robusto** a mudanças no conjunto de treinamento
  - Pelo desvio-padrão se consegue também **comparar inicialmente** dois algoritmos com mesma média
    - O de menor desvio-padrão é mais robusto



# Referências

## ■ Slides de:

- ❑ Profa Dra Ana Carolina Lorena
- ❑ Prof Dr André C. P. L. F. de Carvalho
- ❑ Prof Dr Ricardo Campello
- ❑ Prof Dr Marcilio Carlos Pereira de Souto
- ❑ Livro: A. P. Braga, A. C. P. L. F. Carvalho, T. B. Ludermir, Redes Neurais Artificiais: teoria e aplicações, 2007, Ed LTC
- ❑ Cap 7 livro Inteligência Artificial: uma Abordagem de Aprendizado de Máquina

# Árvores de Decisão

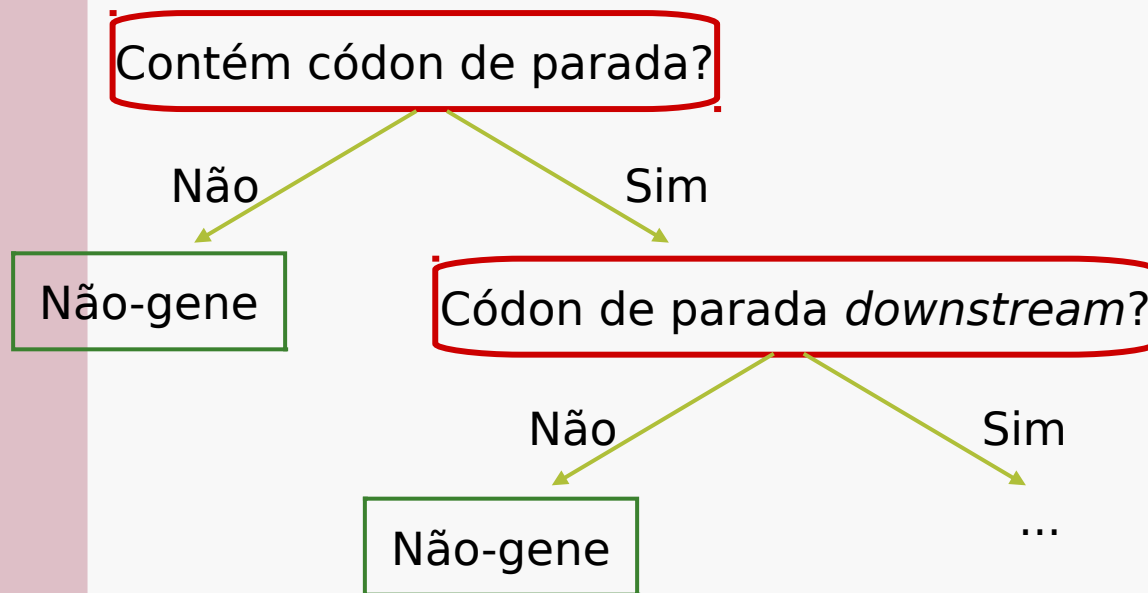
# Árvores de Decisão

- **Árvore de Decisão**: usa estratégia **dividir para conquistar** para resolver problema de decisão
  - Problema complexo é dividido em problemas mais simples, aos quais a mesma estratégia é usada
  - Soluções dos subproblemas são então combinadas
    - *Na forma de uma **árvore***

Em problemas de regressão são denominadas **Árvores de Regressão**, mas, dadas suas semelhanças, usaremos o termo **Árvore de Decisão** de maneira genérica

# Árvores de Decisão

- Estrutura da árvore é determinada por processo de aprendizado
  - Ex. caracterizar genes



# Árvore de Decisão

- **Formalmente:** grafo direcionado acíclico em que cada nó é:

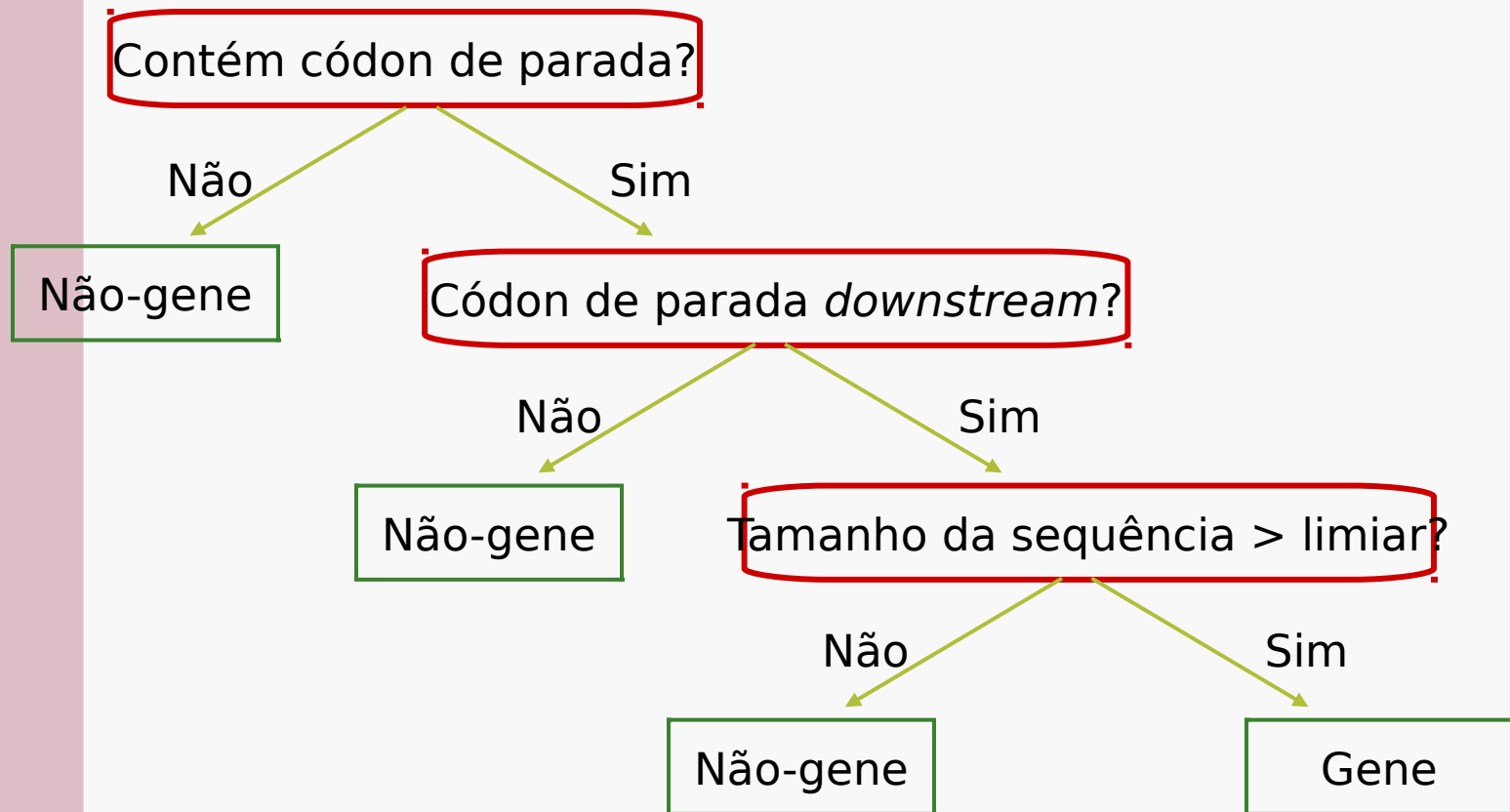
## Nó de divisão:

- Possui dois ou mais sucessores
- Contém *teste condicional* baseado nos valores de atributos
- **Padrão:** testes univariados e um atributo
- *Ex: Idade > 18, Profissão*  $\in \{\text{professor, estudante}\}$ ,  
 $0,3 + 0,2 x^1 - 0,5 x^2 \leq 0$

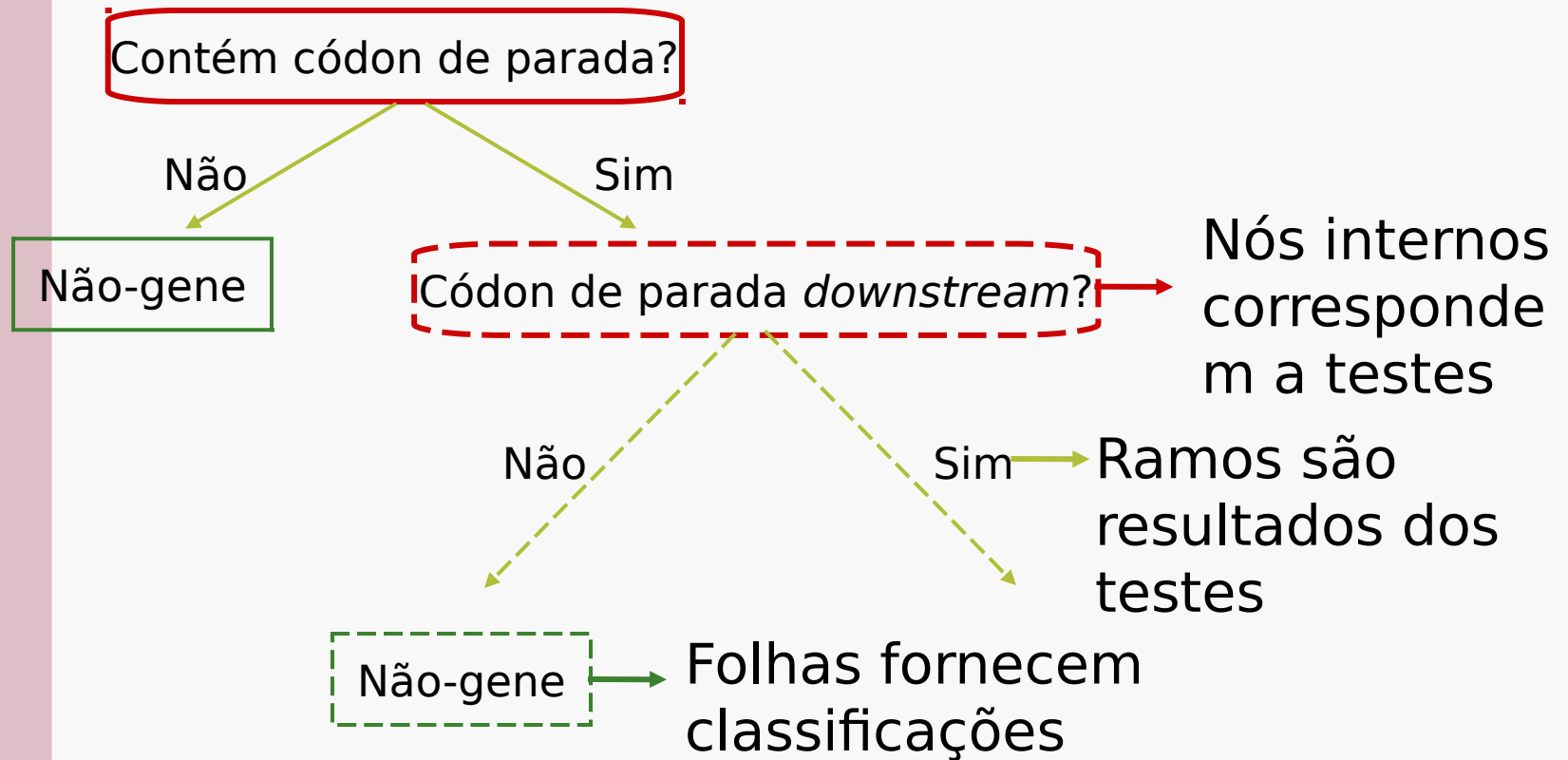
## Nó folha:

- É rotulado com uma *função* que considera valores da variável alvo dos **exemplos que chegam na folha**
- **Classificação:** moda
- **Regressão:** média

# Exemplo: determinar gene

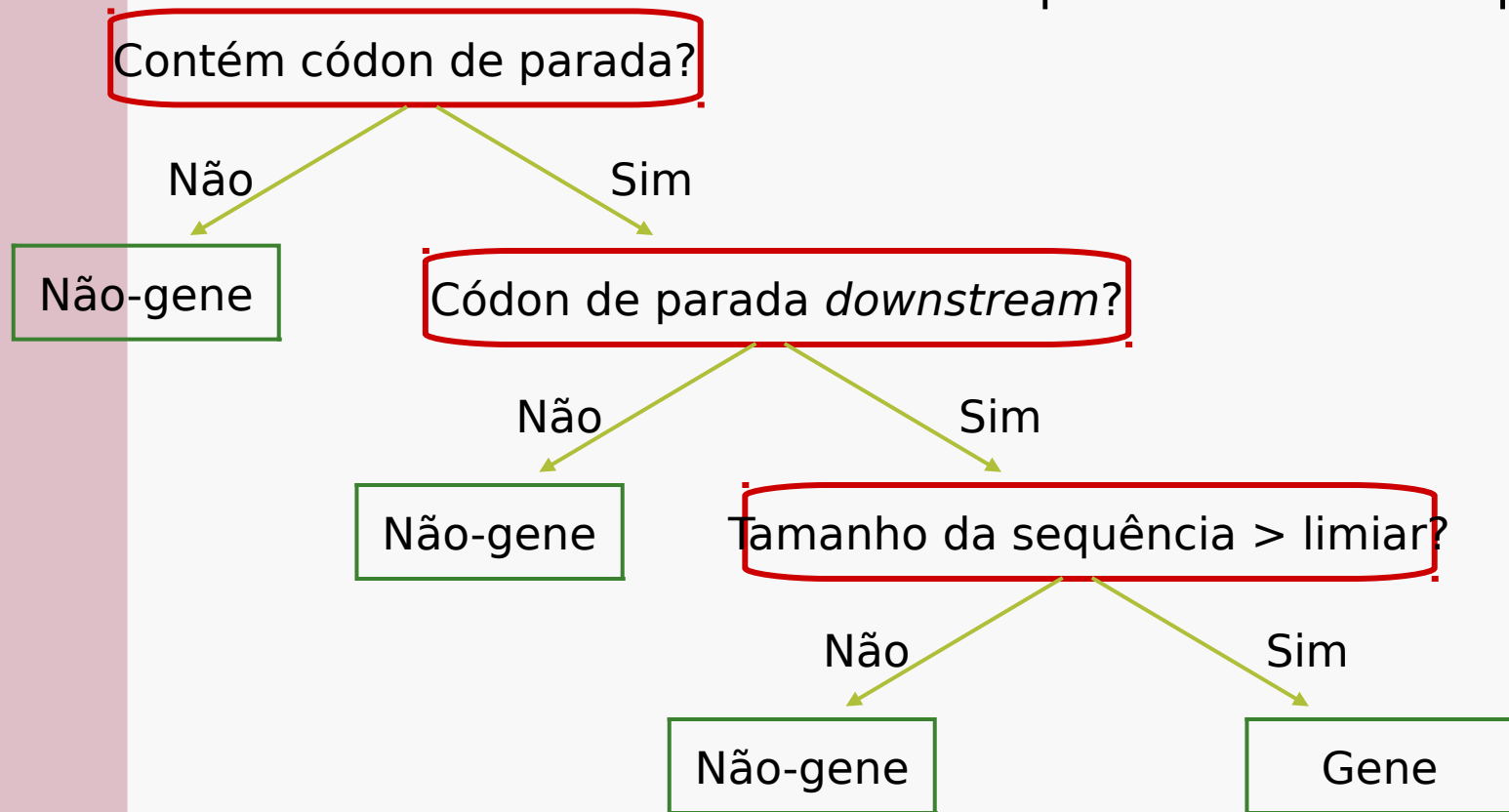


# Exemplo: determinar gene



# Exemplo: determinar gene

**Novo dado:** Contém códon de parada downstream e tamanho da sequência é menor que limiar





# Exemplo: determinar gene

**Novo dado:** Contém códon de parada downstream e tamanho da sequência é menor que limiar

Contém códon de parada?

Não

Não-gene

Sim

Códon de parada *downstream*?

Não

Não-gene

Sim

Tamanho da sequência > limiar?

Não

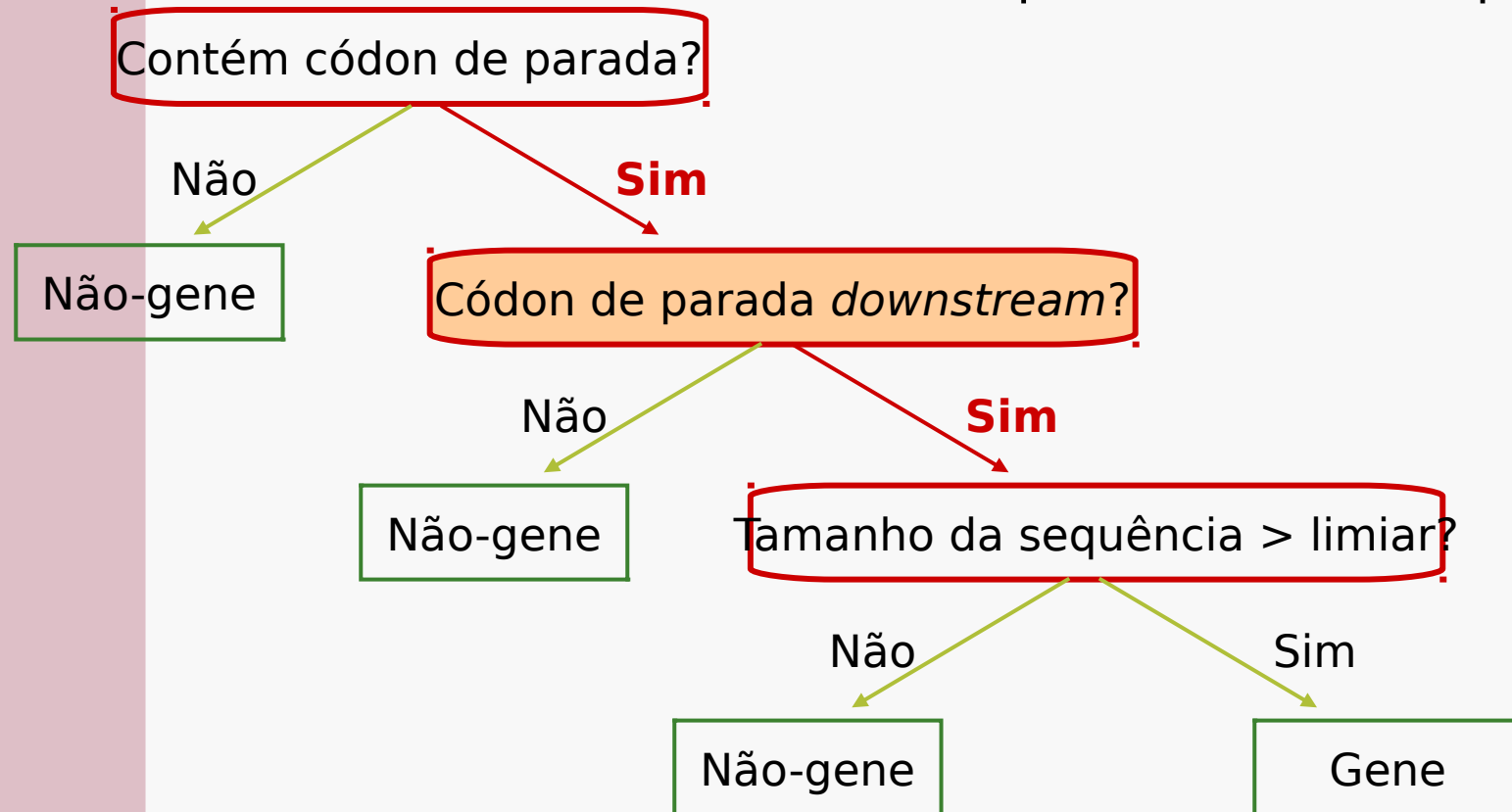
Não-gene

Sim

Gene

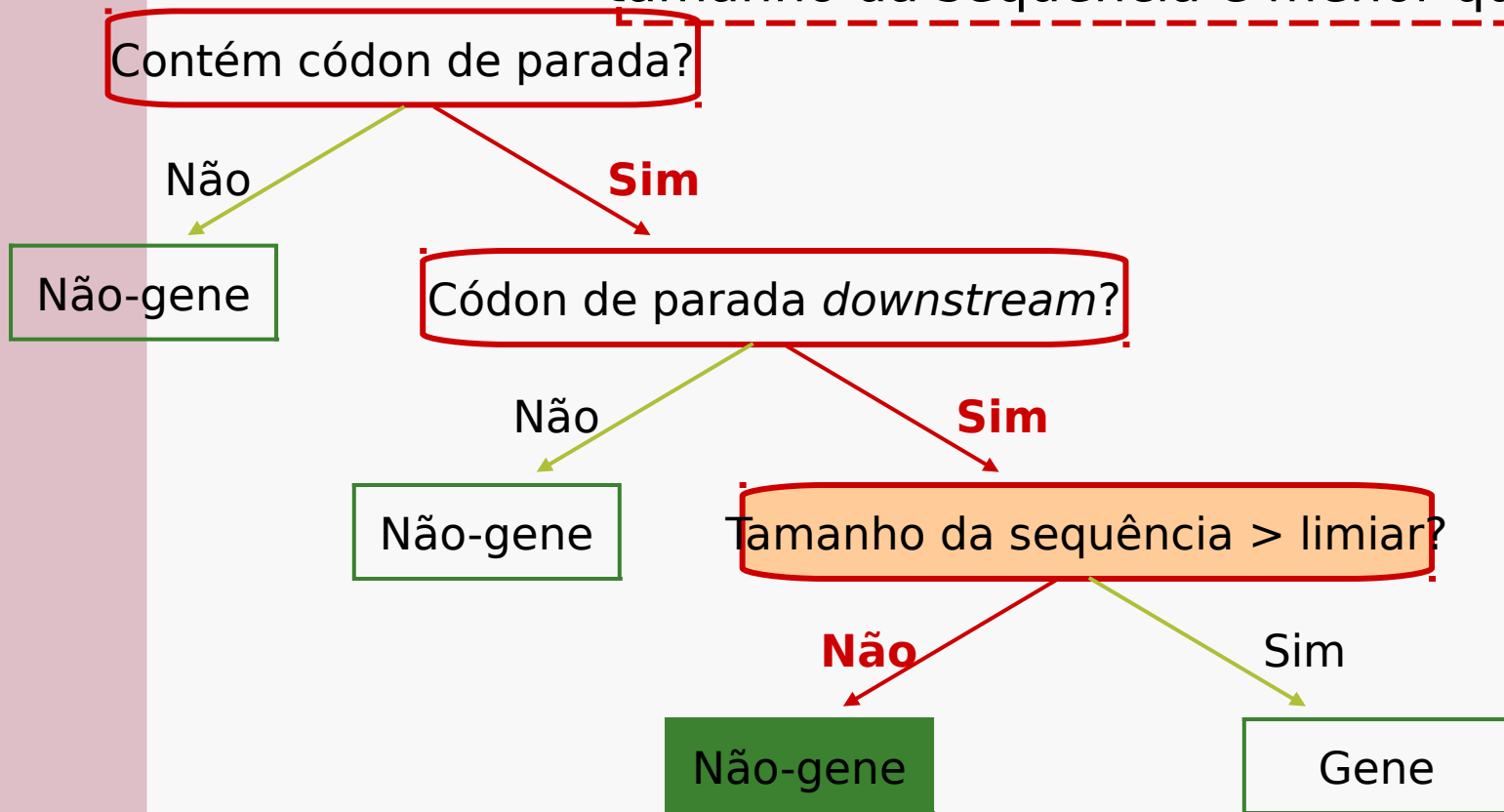
# Exemplo: determinar gene

**Novo dado:** Contém códon de parada downstream e tamanho da sequência é menor que limiar



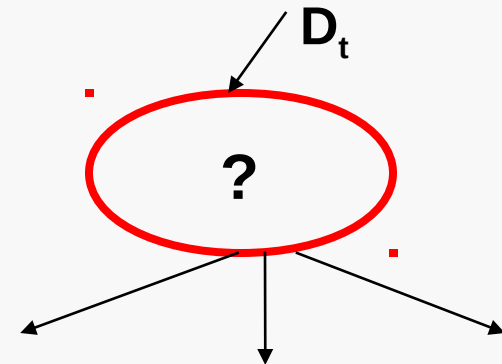
# Exemplo: determinar gene

**Novo dado:** Contém códon de parada downstream e tamanho da sequência é menor que limiar



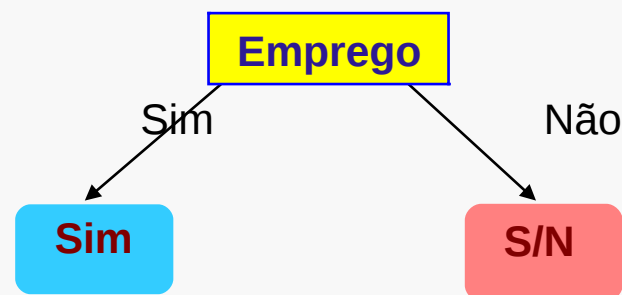
# Algoritmo de Hunt

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Sim
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Sim
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Não	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	500	Não
Não	Divorciado	8000	Não



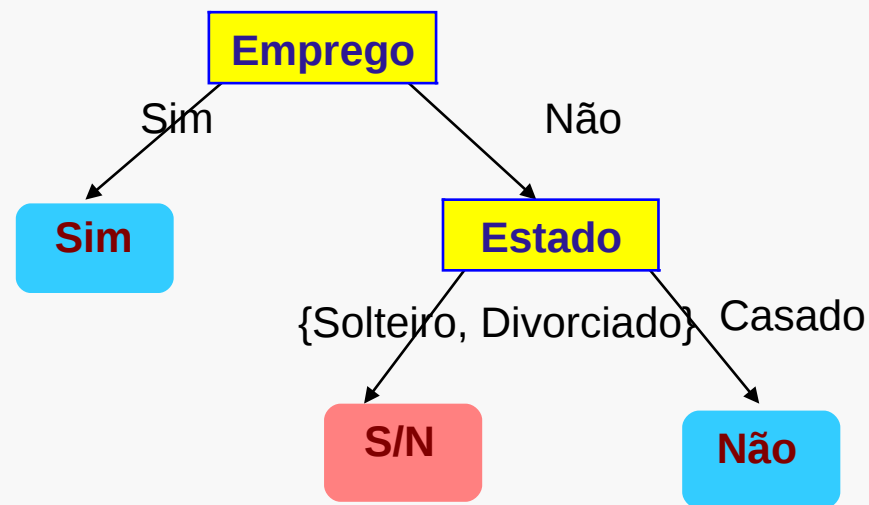
# Algoritmo de Hunt

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Sim
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Sim
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Sim
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Divorciado	8000	Não



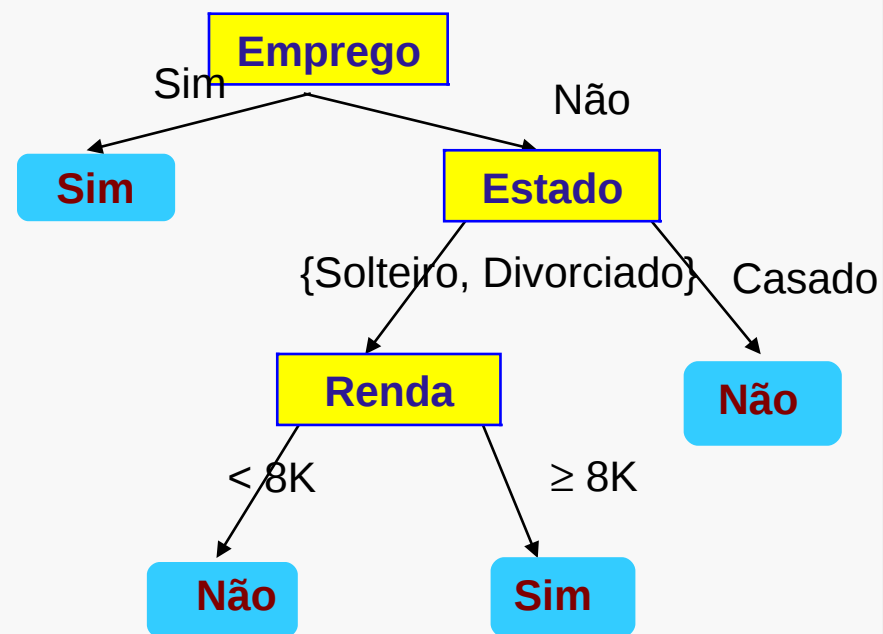
# Algoritmo de Hunt

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Sim
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Sim
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Sim
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Divorciado	8000	Não



# Algoritmo de Hunt

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Sim
Não	Casado	8000	Não
<b>Não</b>	<b>Solteiro</b>	<b>7000</b>	<b>Não</b>
Sim	Casado	12000	Sim
<b>Não</b>	<b>Divorciado</b>	<b>9000</b>	<b>Sim</b>
Não	Casado	6000	Não
Sim	Divorciado	4000	Sim
<b>Não</b>	<b>Solteiro</b>	<b>8500</b>	<b>Sim</b>
Não	Casado	7500	Não
<b>Não</b>	<b>Divorciado</b>	<b>8000</b>	<b>Não</b>



# Indução de Árvore de Decisão

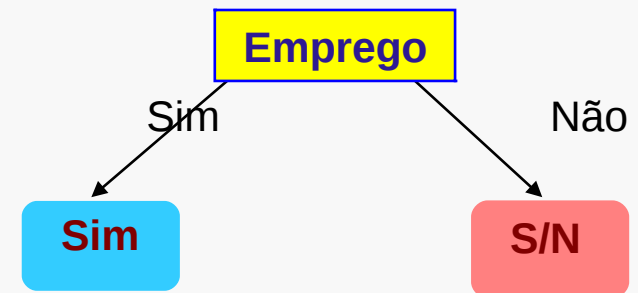
- Decisões importantes
  - Como dividir os objetos
    - *Método para escolha do atributo de teste*
      - Medida para avaliar qualidade de atributo escolhido
  - Quando parar de dividir os objetos



# Como dividir os objetos?

- Valores de atributos particionam os objetos
  - Como divisão é feita depende:
    - *Do tipo do atributo*
    - *Do número de divisões suportada pelo algoritmo*

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Sim
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Sim
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Sim
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim

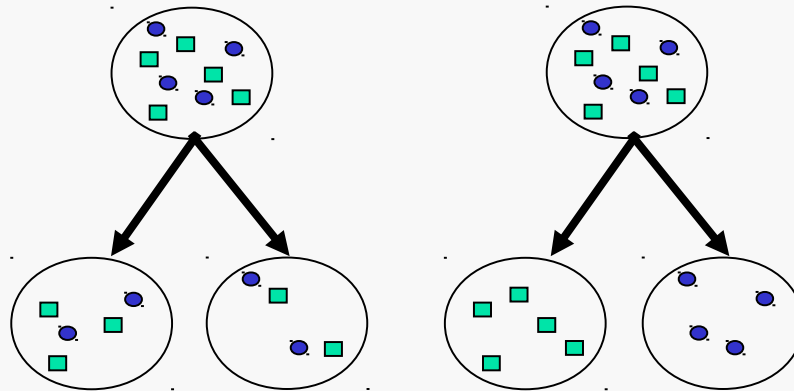


# Como dividir os atributos

- **Qualitativos:** usualmente
  - $\# \text{ramos} = \# \text{possíveis valores}$
- **Quantitativos:** usualmente
  - Comparação ( $A < \text{valor}$ )
  - Escolher posição (valor) que gera melhor partição
    - *Ponto de referência*

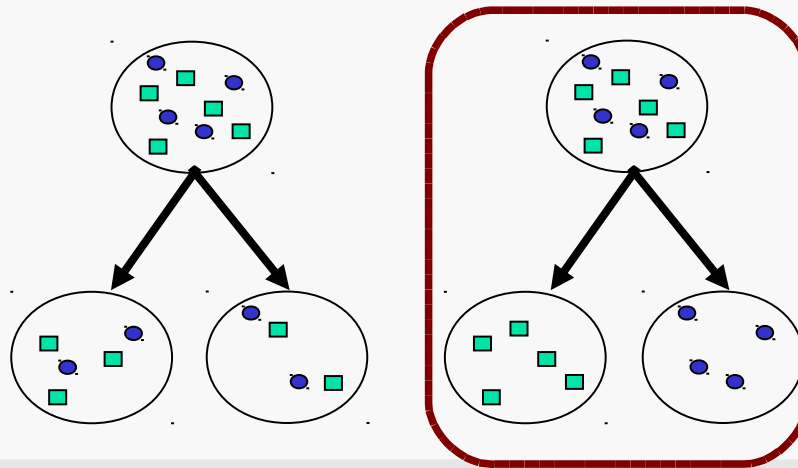
# Que atributo escolher para divisão?

- Regras de divisão para **classificação**:
  - Guiada por medida de *goodness of split*



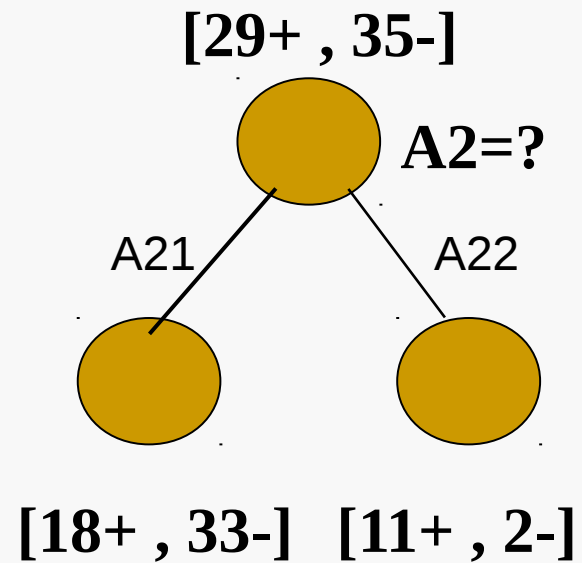
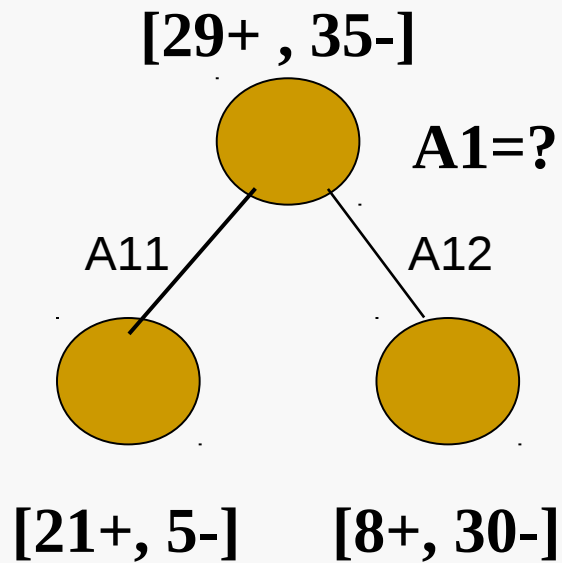
# Que atributo escolher para divisão?

- Regras de divisão para **classificação**:
  - Guiada por medida de *goodness of split*
    - *Indica quão bem um atributo discrimina as classes*
    - *Selecionar atributo que maximiza a medida*
  - Funciona como heurística que olha um passo para frente



# Exemplo

- Qual é o melhor atributo?



# Entropia em AD

- Usada como medida de impureza para medir a **aleatoriedade** (dificuldade para prever) do **atributo alvo**
  - Para caso binário:
    - *A entropia é 0 se todos os elementos pertencem à mesma classe*
      - Pureza máxima
    - *A entropia é 1 quando a coleção contém número igual de exemplos positivos e negativos*
  - $\Rightarrow$  A cada nó de decisão, o atributo que mais reduz a aleatoriedade do alvo é escolhido para divisão

# Ganho de informação

- Ganho de informação mede redução na entropia nas partições obtidas de acordo com os valores do atributo
  - Diferença entre entropia do conjunto de exemplos e a soma ponderada da entropia das partições

AD é guiada a **reduzir entropia** (aleatoriedade/dificuldade de predizer) da variável alvo

- Equação de redução de impureza usando a entropia recebe o nome de ganho de informação

# Entropia em AD

- Sejam  $p$  e  $q$  o número de objetos de duas classes diferentes em um conjunto de dados  $D$

$$H(D) = -\frac{p}{p+q} \log\left(\frac{p}{p+q}\right) - \frac{q}{p+q} \log\left(\frac{q}{p+q}\right)$$

Probabilidade é computada a partir do conjunto de treinamento  $D$



# Entropia em AD

- Entropia pode ser usada em problemas com mais que duas classes (k classes):

$$H(D) = \sum_{i=1}^k -p_i \log_2(p_i)$$

# Entropia em AD

- Se atributo A com v valores é selecionado, a árvore resultante tem um conteúdo de informação esperado de:

$$H(A, D) = \sum_{i=1}^v \frac{p_i + q_i}{p + q} H(D_i)$$

- $p_i$  e  $q_i$ : números de objetos em cada classe na partição  $D_i$

# Ganho de informação

- Ganho de informação alcançado selecionando A para divisão:

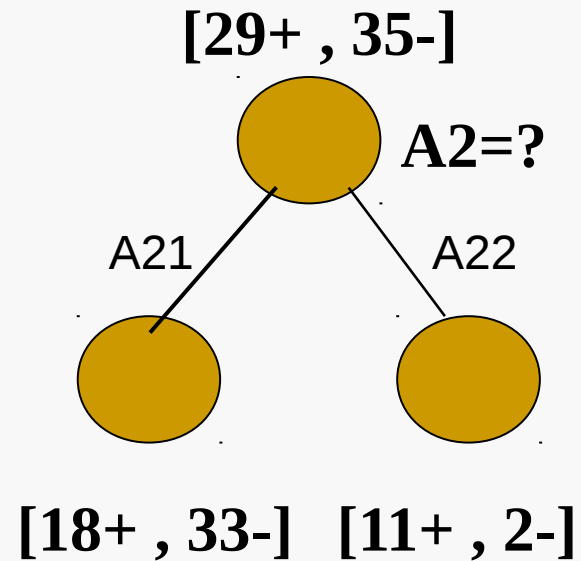
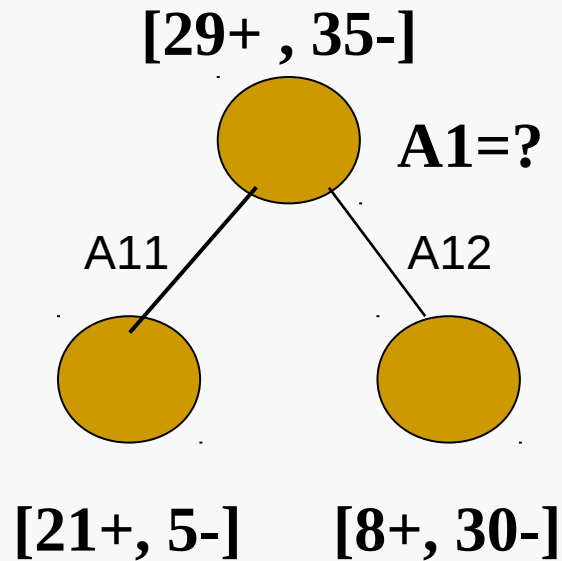
$$IG(A, D) = H(D) - H(A, D)$$

*Ganho(A)* = redução esperada da entropia devido à “classificação” de acordo com o atributo A

$$IG(A, D) \equiv H(D) - \sum_{v \in \text{Valores}(A)} \frac{|D_v|}{|D|} H(D_v)$$

# Exemplo: que atributo escolher?

- Usar o critério de ganho de informação para decidir!



# Exemplo: que atributo escolher?

- Ganho de informação:

- $D = \{29 +, 35 -\}$

- $H(D) = -(29/64) \cdot \log_2(29/64) - (35/64) \cdot \log_2(35/64) = 0,994$

- *De acordo com A1:*

- $D_{A11} = \{21 +, 5 -\}$

- $H(D_{A11}) = -(21/26) \cdot \log_2(21/26) - (5/26) \cdot \log_2(5/26) = 0,706$

- $D_{A12} = \{8 +, 30 -\}$

- $H(D_{A12}) = -(8/38) \cdot \log_2(8/38) - (30/38) \cdot \log_2(30/38) = 0,742$

- $IG(A1, D) = 0,994 - ((26/64) \cdot 0,706 + (38/64) \cdot 0,742) = 0,266$

# Exemplo: que atributo escolher?

- Ganho de informação:

- *De acordo com A2:*

- $D_{A21} = \{18 +, 33 -\}$

- $H(D_{21}) = -(18/51) \cdot \log_2(18/51) - (33/51) \cdot \log_2(33/51) = 0,937$

- $D_{A22} = \{11 +, 2 -\}$

- $H(D_{A22}) = -(11/13) \cdot \log_2(11/13) - (2/13) \cdot \log_2(2/13) = 0,619$

- $IG(A2, D) = 0,994 - ((51/64) \cdot 0,937 + (13/64) \cdot 0,619) = 0,121$

A1 traz maior ganho de informação, então ele é escolhido

# Exemplo ilustrativo

- Conjunto de dados play
  - Decidir quando jogar dadas condições de tempo

Tempo	Temperatura	Umidade	Vento	Joga
Chuvoso	71	91	Sim	Não
Ensolarado	69	70	Não	Sim
Ensolarado	80	90	Sim	Não
Nublado	83	86	Não	Sim
Chuvoso	70	96	Não	Sim
Chuvoso	65	70	Sim	Não
Nublado	64	65	Sim	Sim
Nublado	72	90	Sim	Sim
Ensolarado	75	70	Sim	Sim
Chuvoso	68	80	Não	Sim
Nublado	81	75	Não	Sim
Ensolarado	85	85	Não	Não
Ensolarado	72	95	Não	Não
Chuvoso	75	80	Não	Sim

# Exemplo ilustrativo

## ■ Conjunto de dados play

Tempo	Temperatura	Umidade	Vento	Joga
Chuvoso	71	91	Sim	Não
Ensolarado	69	70	Não	Sim
Ensolarado	80	90	Sim	Não
Nublado	83	86	Não	Sim
Chuvoso	70	96	Não	Sim
Chuvoso	65	70	Sim	Não
Nublado	64	65	Sim	Sim
Nublado	72	90	Sim	Sim
Ensolarado	75	70	Sim	Sim
Chuvoso	68	80	Não	Sim
Nublado	81	75	Não	Sim
Ensolarado	85	85	Não	Não
Ensolarado	72	95	Não	Não
Chuvoso	75	80	Não	Sim

Entropia da classe para todo o conjunto de exemplos:

$$p(\text{Joga} = \text{Sim}) = 9/14 = 0,64$$

$$p(\text{Joga} = \text{Não}) = 5/14 = 0,36$$

$$H(\text{Joga}) = -9/14 \log_2(9/14) - 5/14 \log_2(5/14) = 0,94 \text{ bit}$$



# Exemplo ilustrativo

## ■ Conjunto de dados play /

Tempo	Temperatura	Umidade	Vento	Joga
Chuvoso	71	91	Sim	Não
Ensolarado	69	70	Não	Sim
Ensolarado	80	90	Sim	Não
Nublado	83	86	Não	Sim
Chuvoso	70	96	Não	Sim
Chuvoso	65	70	Sim	Não
Nublado	64	65	Sim	Sim
Nublado	72	90	Sim	Sim
Ensolarado	75	70	Sim	Sim
Chuvoso	68	80	Não	Sim
Nublado	81	75	Não	Sim
Ensolarado	85	85	Não	Não
Ensolarado	72	95	Não	Não
Chuvoso	75	80	Não	Sim

Joga	Ensolarado	Nublado	Chuvoso
Sim	2	4	3
Não	3	0	2

## IG para atributos nominais:

Ex. atributo **Tempo**: três partições

1º passo: estimar probabilidades de observar classes dado cada valor

$$p(\text{Joga}|\text{Ensolarado}) = 2/5$$

$$p(\neg\text{Joga}|\text{Ensolarado}) = 3/5$$

$$H(\text{Joga}|\text{Ensolarado}) = -2/5 \cdot \log_2(2/5) - 3/5 \cdot$$

$$\log_2(3/5) = 0,971 \text{ bit}$$

$$p(\text{Joga}|\text{Nublado}) = 4/4$$

$$p(\neg\text{Joga}|\text{Nublado}) = 0/4$$

$$H(\text{Joga}|\text{Nublado}) = 0,0 \text{ bit}$$

$$p(\text{Joga}|\text{Chuvoso}) = 3/5$$

$$p(\neg\text{Joga}|\text{Chuvoso}) = 2/5$$

$$H(\text{Joga}|\text{Chuvoso}) = 0,971 \text{ bit}$$

# Exemplo ilustrativo

## ■ Conjunto de dados play

Tempo	Temperatura	Umidade	Vento	Joga
Chuvoso	71	91	Sim	Não
Ensolarado	69	70	Não	Sim
Ensolarado	80	90	Sim	Não
Nublado	83	86	Não	Sim
Chuvoso	70	96	Não	Sim
Chuvoso	65	70	Sim	Não
Nublado	64	65	Sim	Sim
Nublado	72	90	Sim	Sim
Ensolarado	75	70	Sim	Sim
Chuvoso	68	80	Não	Sim
Nublado	81	75	Não	Sim
Ensolarado	85	85	Não	Não
Ensolarado	72	95	Não	Não
Chuvoso	75	80	Não	Sim

### IG para atributos nominais:

Ex. atributo **Tempo**: três partições

**2º passo:** calcular a entropia ponderada para o atributo Tempo

$$H(D, \text{Tempo}) = 5/14 * 0,971 + 4/14 * 0 + 5/14 * 0,971 = 0,693 \text{ bit}$$

**3º passo:** calcular o ganho de informação em dividir o conjunto de acordo com os valores do atributo Tempo

$$\begin{aligned} IG(\text{Tempo}) &= H(D) - H(D, \text{Tempo}) \\ &= 0,940 - 0,693 = \mathbf{0,247 \text{ bit}} \end{aligned}$$

⇒ *Conhecendo o valor do atributo Tempo, precisamos de menos bits para codificar o valor do atributo alvo*

# Exemplo ilustrativo

## ■ Conjunto de dados play

Tempo	Temperatura	Umidade	Vento	Joga
Nublado	64	65	Sim	Sim
Chuvoso	65	70	Sim	Não
Chuvoso	68	80	Não	Sim
Ensolarado	69	70	Não	Sim
Chuvoso	70	96	Não	Sim
Chuvoso	71	91	Sim	Não
Nublado	72	90	Sim	Sim
Ensolarado	72	95	Não	Não
Chuvoso	75	80	Não	Sim
Ensolarado	75	70	Sim	Sim
Ensolarado	80	90	Sim	Não
Nublado	81	75	Não	Sim
Nublado	83	86	Não	Sim
Ensolarado	85	85	Não	Não

### IG para atributos contínuos:

Buscar partição binária dos valores

- Atributo  $\leq$  valor
- Atributo  $>$  valor

E aplicar as equações a essas partições

Ex. atributo **Temperatura**

**1º passo:** definir ponto de corte

- Ordena-se os valores do atributo
- Pega a média de dois valores consecutivos: *candidato* a ponto de corte
- Avalia mérito (ex. IG) do ponto de corte
- Escolhe ponto que maximiza mérito

No exemplo, 1º ponto de corte = 64,5 e último ponto de corte = 84

# Exemplo ilustrativo

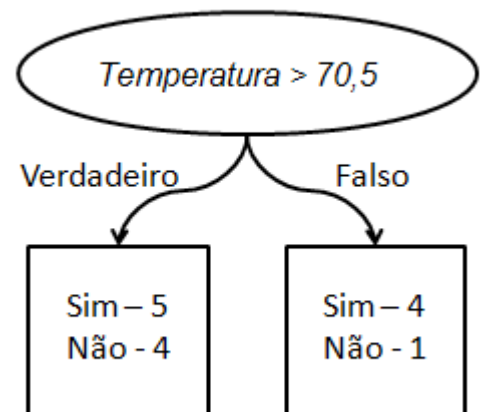
## ■ Conjunto de dados play

Tempo	Temperatura	Umidade	Vento	Joga
Nublado	64	65	Sim	Sim
Chuvoso	65	70	Sim	Não
Chuvoso	68	80	Não	Sim
Ensolarado	69	70	Não	Sim
Chuvoso	70	96	Não	Sim
Chuvoso	71	91	Sim	Não
Nublado	72	90	Sim	Sim
Ensolarado	72	95	Não	Não
Chuvoso	75	80	Não	Sim
Ensolarado	75	70	Sim	Sim
Ensolarado	80	90	Sim	Não
Nublado	81	75	Não	Sim
Nublado	83	86	Não	Sim
Ensolarado	85	85	Não	Não

## IG para atributos contínuos:

Ex. atributo **Temperatura**

2º passo: Escolhido ponto de corte, fazer cálculos de IG correspondentes  
Considerando o ponto 70,5:



# Exemplo ilustrativo

## ■ Conjunto de dados play

Tempo	Temperatura	Umidade	Vento	Joga
Nublado	64	65	Sim	Sim
Chuvoso	65	70	Sim	Não
Chuvoso	68	80	Não	Sim
Ensolarado	69	70	Não	Sim
Chuvoso	70	96	Não	Sim
Chuvoso	71	91	Sim	Não
Nublado	72	90	Sim	Sim
Ensolarado	72	95	Não	Não
Chuvoso	75	80	Não	Sim
Ensolarado	75	70	Sim	Sim
Ensolarado	80	90	Sim	Não
Nublado	81	75	Não	Sim
Nublado	83	86	Não	Sim
Ensolarado	85	85	Não	Não

### IG para atributos contínuos:

Ex. atributo **Temperatura**

2º passo: considerando o ponto 70,5:

$$p(\text{Joga} | \text{Temperatura} \leq 70,5) = 4/5$$

$$p(\neg \text{Joga} | \text{Temperatura} \leq 70,5) = 1/5$$

$$p(\text{Joga} | \text{Temperatura} > 70,5) = 5/9$$

$$p(\neg \text{Joga} | \text{Temperatura} > 70,5) = 4/9$$

$$H(\text{Joga} | \text{Temperatura} \leq 70,5) = -4/5 \log_2(4/5)$$

$$-1/5 \log_2(1/5) = 0,721 \text{ bit}$$

$$H(\text{Joga} | \text{Temperatura} > 70,5) = -5/9 \log_2(5/9)$$

$$-4/9 \log_2(4/9) = 0,991 \text{ bit}$$

$$E(\text{Temperatura}) = 5/14 * 0,721 + 9/14 * 0,991 = 0,895 \text{ bit}$$

$$IG(\text{Temperatura}) = 0,940 - 0,895 = 0,045 \text{ bit}$$

# Divisão de atributos contínuos

- Método pode ser acelerado
  - Considerar apenas pontos entre dois objetos adjacentes com classes diferentes
    - *Não – Sim ou Sim - Não*
    - *Redução do número de pontos de corte candidatos para o atributo renda?*

<b>Emprego</b>	<b>Estado</b>	<b>Renda</b>	<b>Crédito</b>
Sim	Solteiro	9500	Sim
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Sim
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Sim
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim

# Divisão de atributos contínuos

- Método pode ser acelerado
  - Considerar apenas pontos entre dois objetos adjacentes com classes diferentes
    - *Não – Sim ou Sim - Não*
    - *Reduz de de 11 para 2 o número de pontos de corte candidatos no exemplo anterior*

# Exercício

- Conjunto de dados play
  - Com atributos quantitativos discretizados

Dia	Tempo	Temperatura	Umidade	Vento	Joga
D1	Ensolarado	Quente	Alta	Fraco	Não
D2	Ensolarado	Quente	Alta	Forte	Não
D3	Nublado	Quente	Alta	Fraco	Sim
D4	Chuvoso	Média	Alta	Fraco	Sim
D5	Chuvoso	Fria	Normal	Fraco	Sim
D6	Chuvoso	Fria	Normal	Forte	Não
D7	Nublado	Fria	Normal	Forte	Sim
D8	Ensolarado	Média	Alta	Fraco	Não
D9	Ensolarado	Fria	Normal	Fraco	Sim
D10	Chuvoso	Média	Normal	Fraco	Sim
D11	Ensolarado	Média	Normal	Forte	Sim
D12	Nublado	Média	Alta	Forte	Sim
D13	Nublado	Quente	Normal	Fraco	Sim
D14	Chuvoso	Média	Alta	Forte	Não



# Exercício

- Que atributo deve ser selecionado para ser a raiz da árvore?

# Poda

- **Poda**: troca de nós profundos por folhas
  - Utilizada para lidar com dados com ruído e evitar super-ajustes
    - *Estatísticas calculadas em nós mais profundos da árvore têm nível mais baixo de importância*
      - Poucos exemplos chegam a esses nós
    - *Árvore grande é difícil para compreender*

# Métodos de poda

## ■ Dois grupos principais:

### Pré-poda

Param a construção da árvore quando algum critério é satisfeito

### Pós-poda

Constroem árvore completa e podam posteriormente

Todos mantêm ponto de equilíbrio entre tamanho da árvore e estimativa de erro

# Vantagens ADs

- Flexibilidade
  - Fornecem cobertura exaustiva do espaço de entradas
- Robustez
  - São invariantes a transformações monótonas de variáveis de entrada
    - *Ex. usar  $x$ ,  $\log x$ ,  $e^x$  produz mesma árvore*
- Seleção de atributos embutida
  - Seleciona atributos mais relevantes em sua construção
  - Robustas a atributos irrelevantes e redundantes



# Vantagens ADs

- Interpretabilidade

- Eficiência

- Algoritmo guloso top-down, com estratégia dividir-para-conquistar



# Desvantagens ADs

- Atributos contínuos
  - Operação de ordenação consome muito tempo
  - Alguns autores recomendam discretização prévia
- Instabilidade
  - Pequenas variações no conjunto de treinamento podem produzir grandes variações na árvore final

# Referências

- Slides de:

- Profa Dra Ana Carolina Lorena, UNIFESP
- Prof Dr André C. P. L. F. Carvalho, ICMC-USP
- Prof Dr Marcilio C. P. Souto, UFPE

Livro Inteligência Artificial: uma Abordagem de Aprendizado de Máquina, capítulo 6