Classificação: conceitos básicos e árvores de decisão

Prof. Luis Otavio Alvares

INE/UFSC

Parte desta apresentação é baseada
no livro Introduction to Data Mining (Tan, Steinbach, Kumar) e
em material do prof. José Todesco (UFSC)

Classificação: Introdução

Classificação: é uma das técnicas mais utilizadas na mineração, por exemplo são comuns as tarefas de classificação de clientes em baixo, médio ou alto risco de empréstimo bancário.

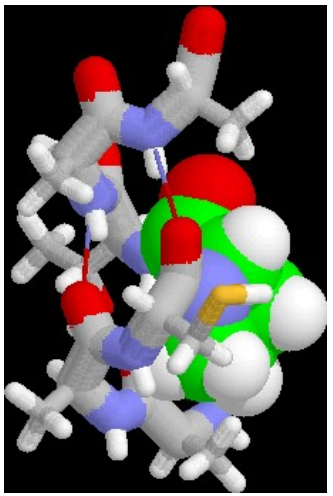
“ Classificar um objeto (registro, amostra, exemplo) é determinar com que grupo de entidades, já classificadas anteriormente, esse objeto apresenta mais semelhanças ”

Exemplos de Tarefas de Classificação

- Predizer se um tumor é **benigno** ou **maligno**

- Classificar transações de cartões de crédito como **legítimas** ou **fraudulentas**

- Classificar estruturas secundárias de proteínas como alpha-helix, beta-sheet, or random coil



- Categorizar textos como da área de finanças, previsão de tempo, esportes, cultura, etc.

Exemplos reais de uso de classificação

- *Upgrade* de pacotes de TV por assinatura (Net)
- Cancelamento de assinaturas (RBS)
- Análise para concessão de empréstimos bancários (instituições financeiras)
- Softwares de correio eletrônico como Outlook e Firefox usam classificadores para filtrar (marcar) emails que seriam *spam*

Classificação: Definição

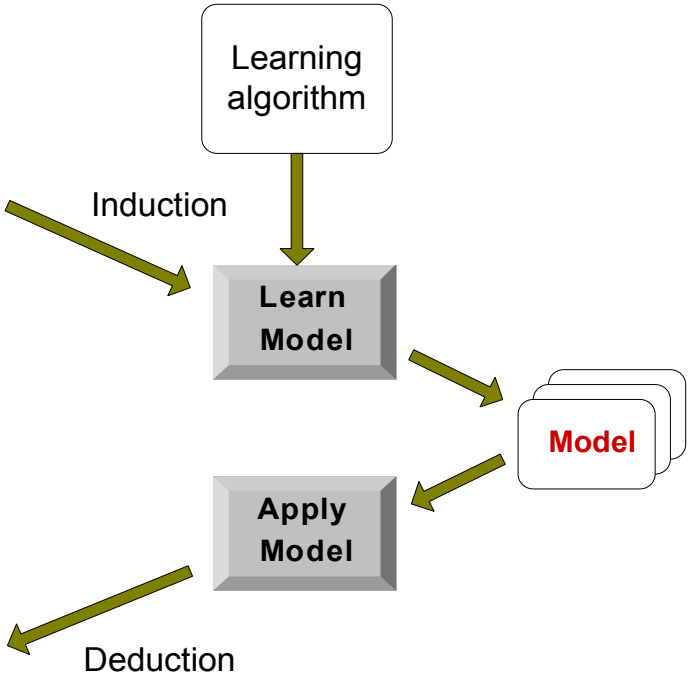
- Dada uma coleção de registros (*conjunto de treinamento*)
  - cada registro contém um conjunto de *atributos*, e um dos atributos é a *classe*.
- Encontrar um *modelo* para determinar o valor do atributo classe em função dos valores de outros atributos.
- Objetivo: definir a classe de novos registros
  - a classe deve ser atribuída o mais corretamente possível
  - Um *conjunto de DADOS de teste* é usado para avaliar o modelo
  - Geralmente o conjunto de dados é dividido em conjunto de treinamento (usado para gerar o modelo) e conjunto de teste.

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Métodos de Classificação

- **Classificadores *eager (espertos)***
  - *A partir da amostragem inicial (conjunto de treinamento), constroem um modelo de classificação capaz de classificar novos registros.*
  - *Uma vez pronto o modelo, o conjunto de treinamento não é mais utilizado na classificação de novos objetos (registros)*
    - **Árvores de Decisão**
    - **Redes Neurais**
    - **Redes Bayesianas e Naïve Bayes**
    - **Máquinas de Vetores de Suporte**
    - **Regras de Decisão**
- **Classificadores *lazy* (preguiçosos)**
  - *Cada novo registro é comparado com todo o conjunto de treinamento e é classificado segundo a classe do registro que é mais similar.*
    - **Método kNN (k-nearest-neighbor)**
- **Outros Métodos**
    - **Algoritmos Genéticos**
    - **Conjuntos Fuzzy**

Árvores de Decisão

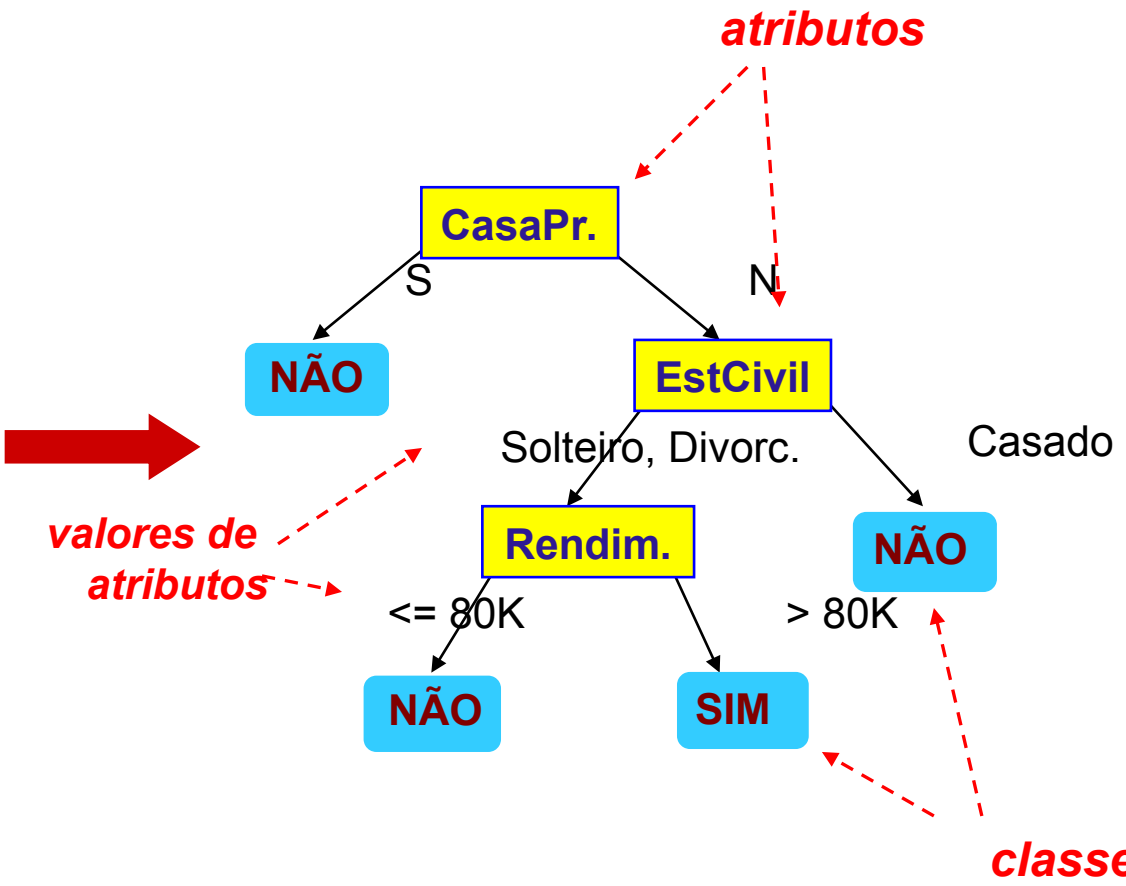
As árvores de decisão são representações gráficas que consistem:

- de nodos que representam os atributos;
- de arcos que correspondem ao valor de um atributo;
- de nodos folha que designam uma classificação.

Exemplo de uma árvore de decisão

	categórico	categórico	continuo	classe
Id	Casa própria	EstCivil	Rendim.	Mau Pagador
1	S	Solteiro	125K	NÃO
2	N	Casado	100K	NÃO
3	N	Solteiro	70K	NÃO
4	S	Casado	120K	NÃO
5	N	Divorc.	95K	SIM
6	N	Casado	60K	NÃO
7	S	Divorc.	220K	NÃO
8	N	Solteiro	85K	SIM
9	N	Casado	75K	NÃO
10	N	Solteiro	90K	SIM

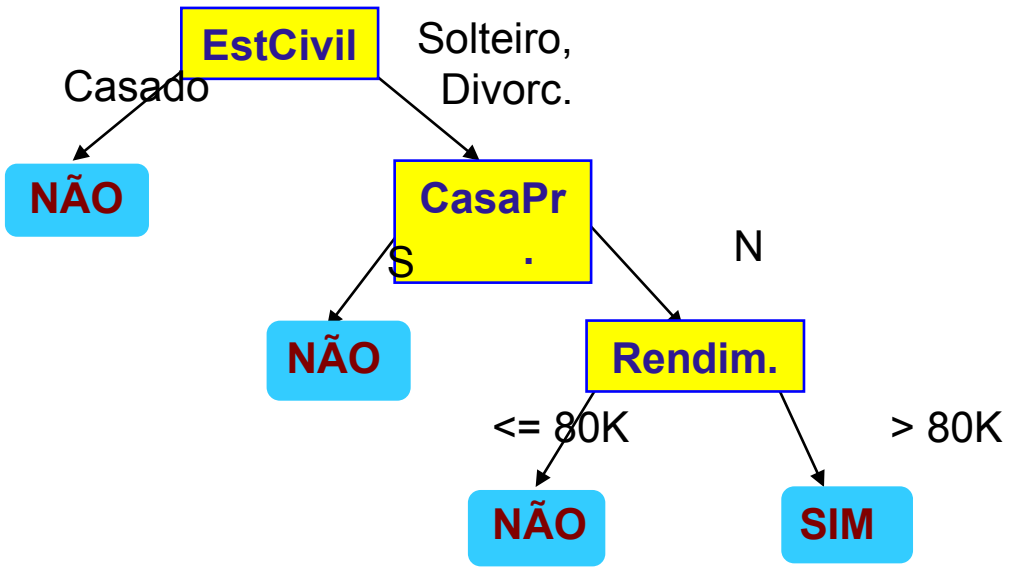
Dados de treinamento



Modelo: árvore de decisão

Outro exemplo de árvore de decisão

	categórico	categórico	continuo	classe
Id	Casa própria	EstCivil	Rendim.	Mau Pagador
1	S	Solteiro	125K	NÃO
2	N	Casado	100K	NÃO
3	N	Solteiro	70K	NÃO
4	S	Casado	120K	NÃO
5	N	Divorc.	95K	SIM
6	N	Casado	60K	NÃO
7	S	Divorc.	220K	NÃO
8	N	Solteiro	85K	SIM
9	N	Casado	75K	NÃO
10	N	Solteiro	90K	SIM



Pode haver mais de uma árvore para o mesmo conjunto de dados!!!

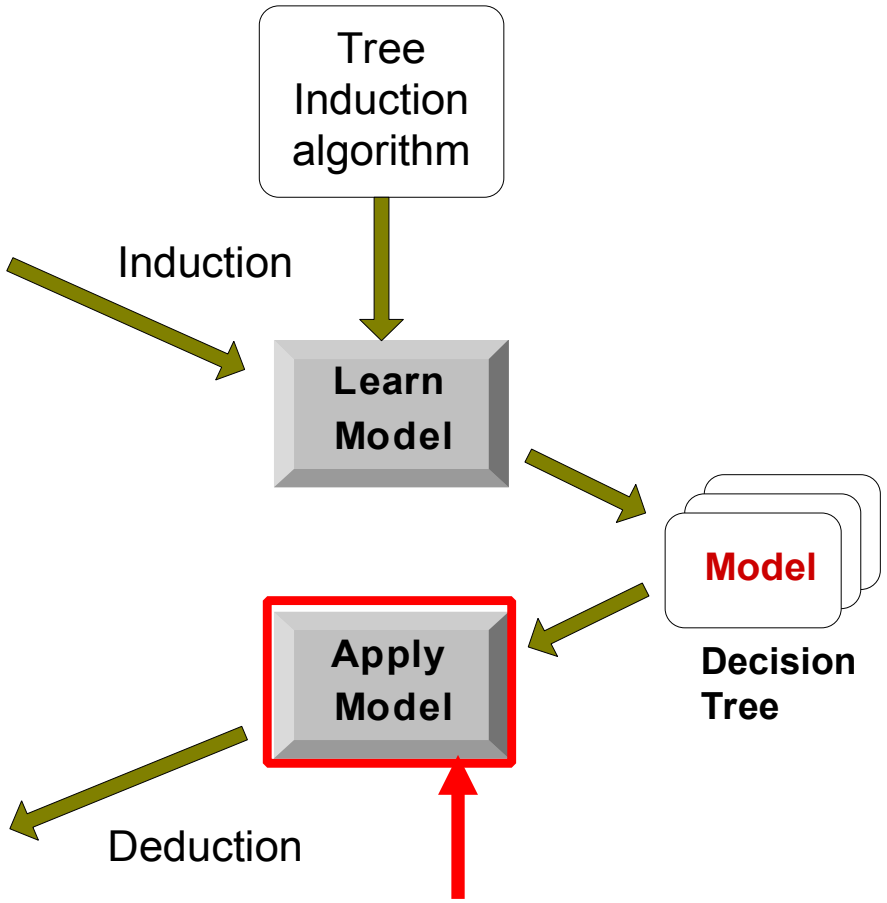
Classificação usando árvores de decisão

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

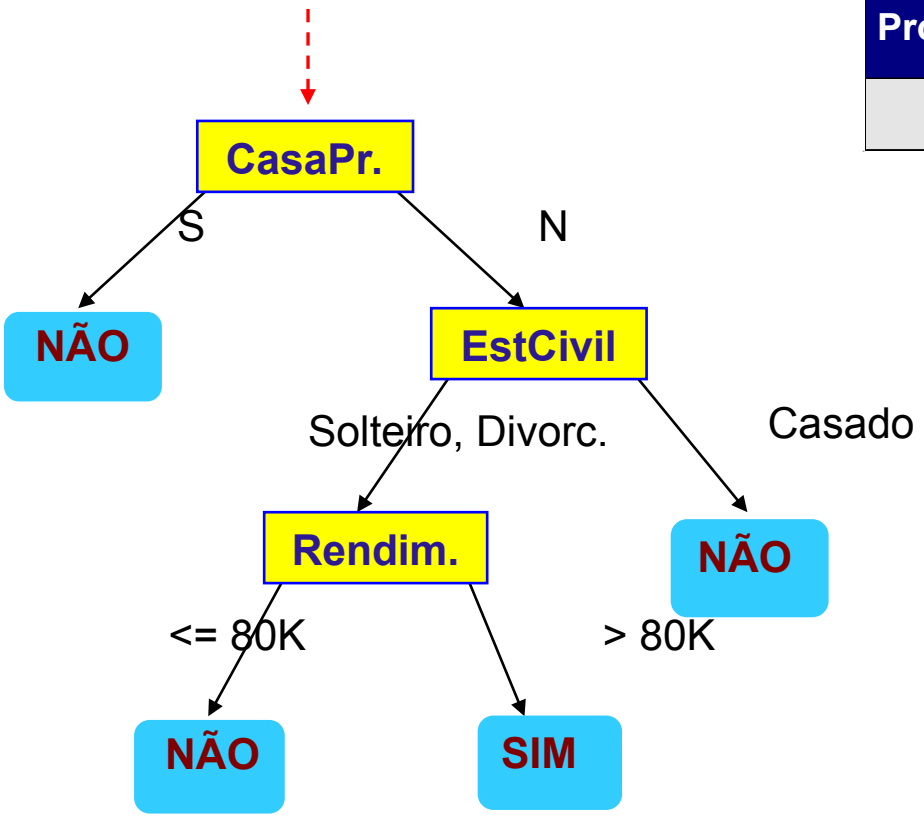
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Aplicando o modelo nos dados de teste

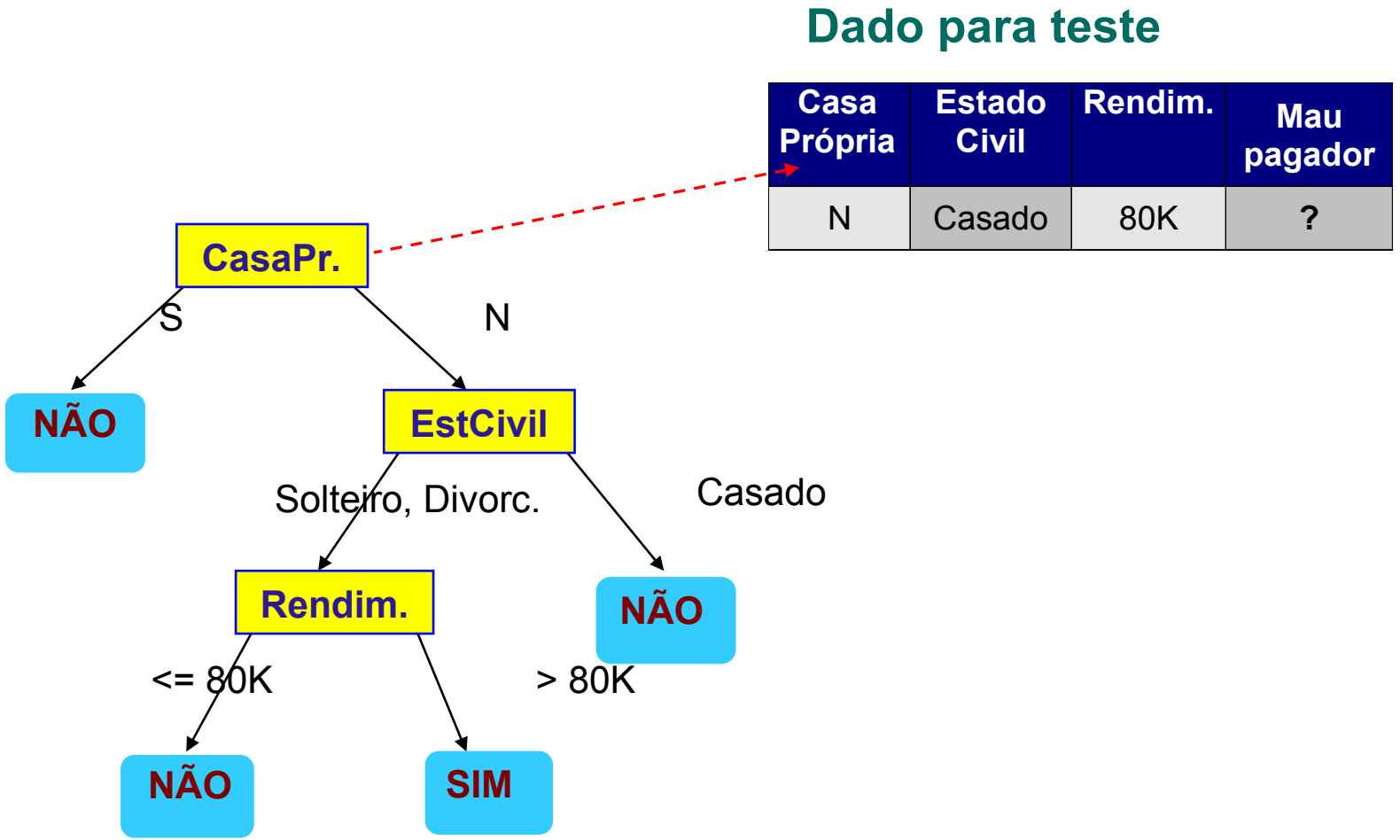
Comece pela raíz da árvore.



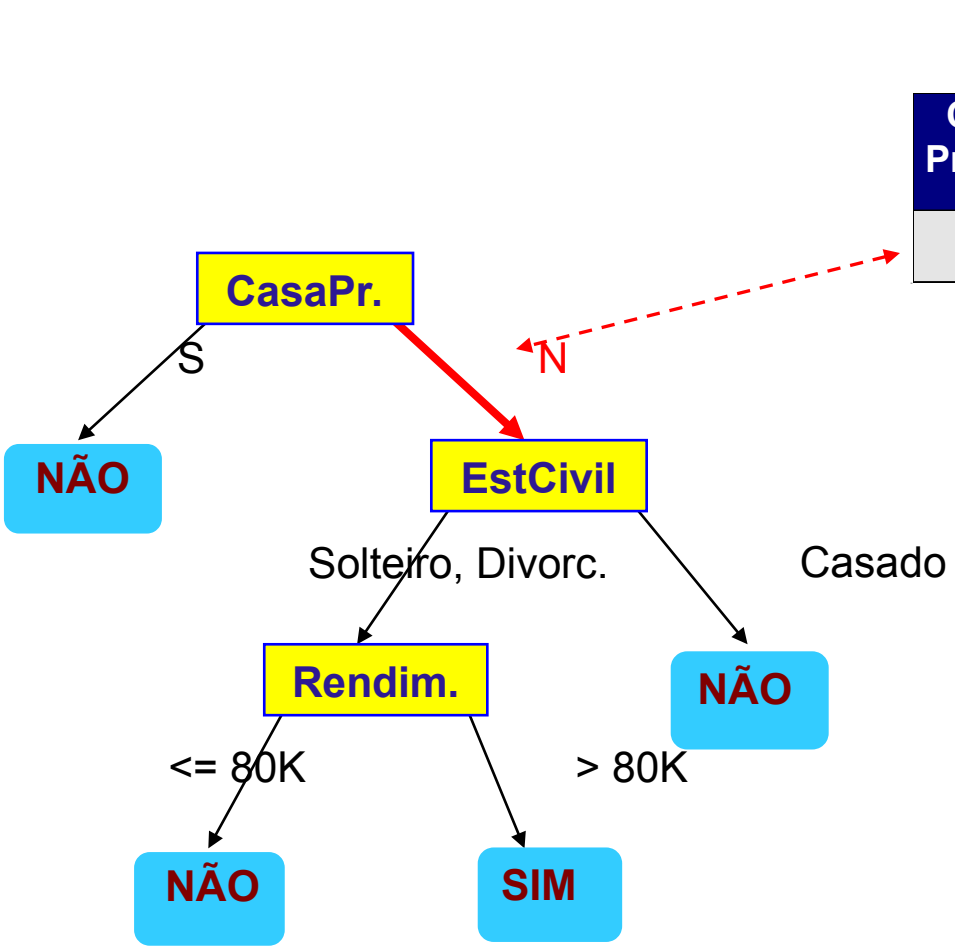
Dado para teste

Casa Própria	Estado Civil	Rendim.	Mau pagador
N	Casado	80K	?

Aplicando o modelo nos dados de teste



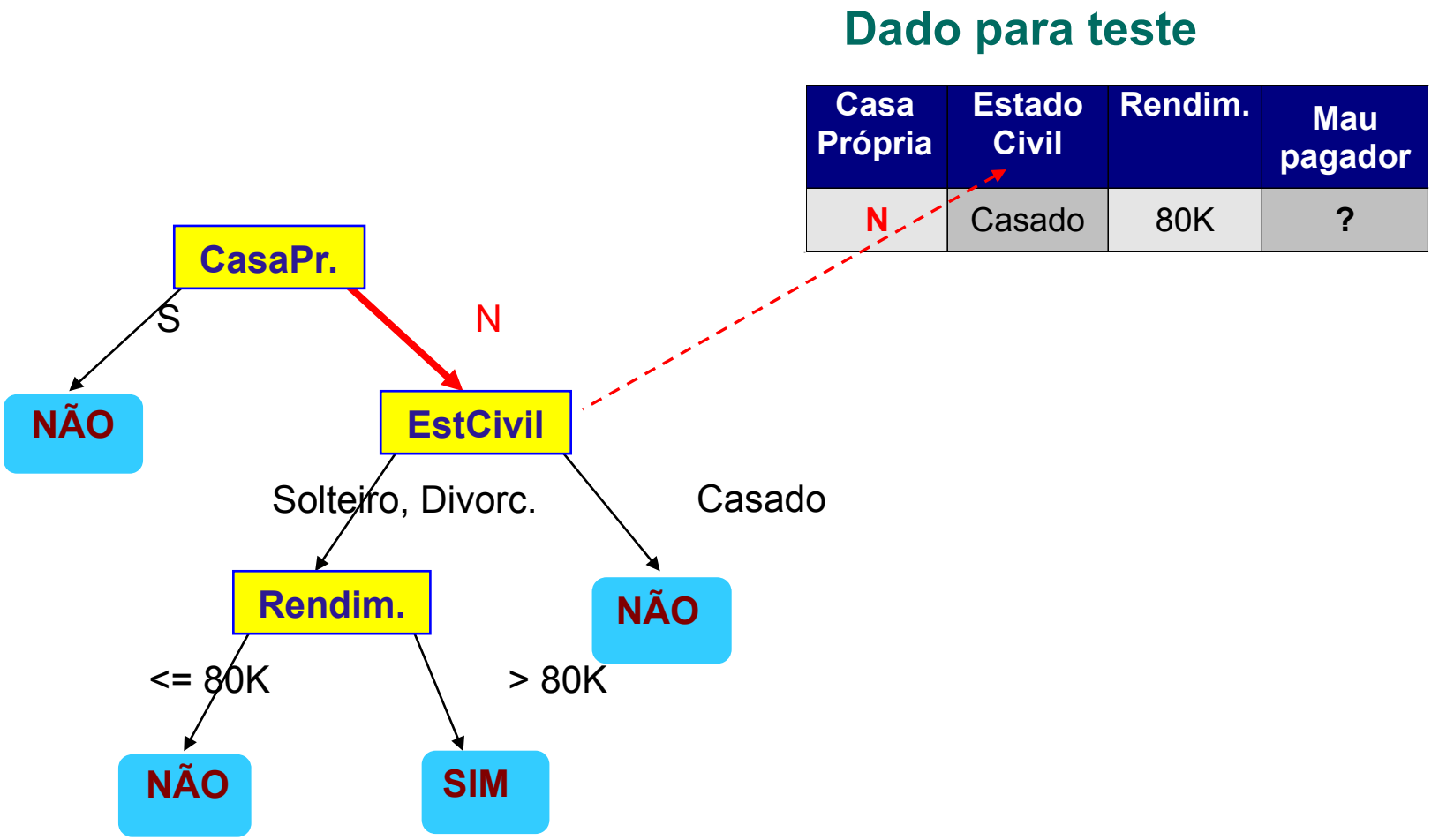
Aplicando o modelo nos dados de teste



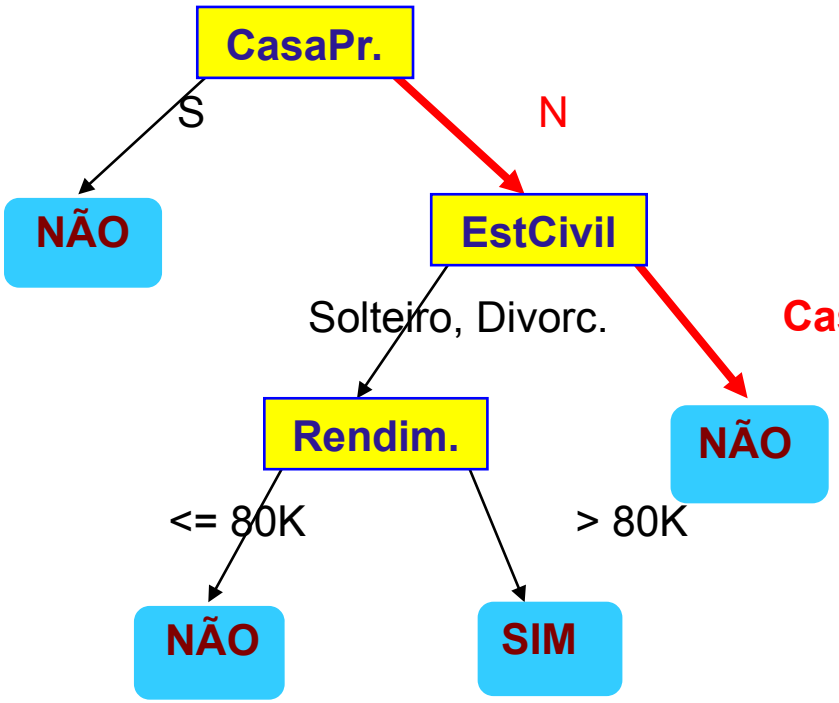
Dado para teste

Casa Própria	Estado Civil	Rendim.	Mau pagador
N	Casado	80K	?

Aplicando o modelo nos dados de teste



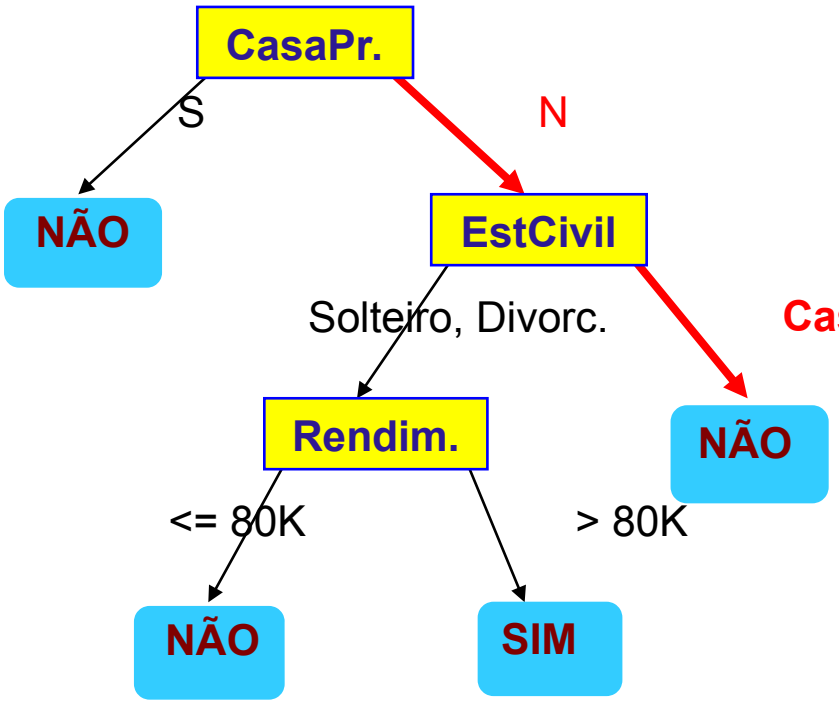
Aplicando o modelo nos dados de teste



Dado para teste

Casa Própria	Estado Civil	Rendim.	Mau pagador
N	Casado	80K	?

Aplicando o modelo nos dados de teste



Dado para teste

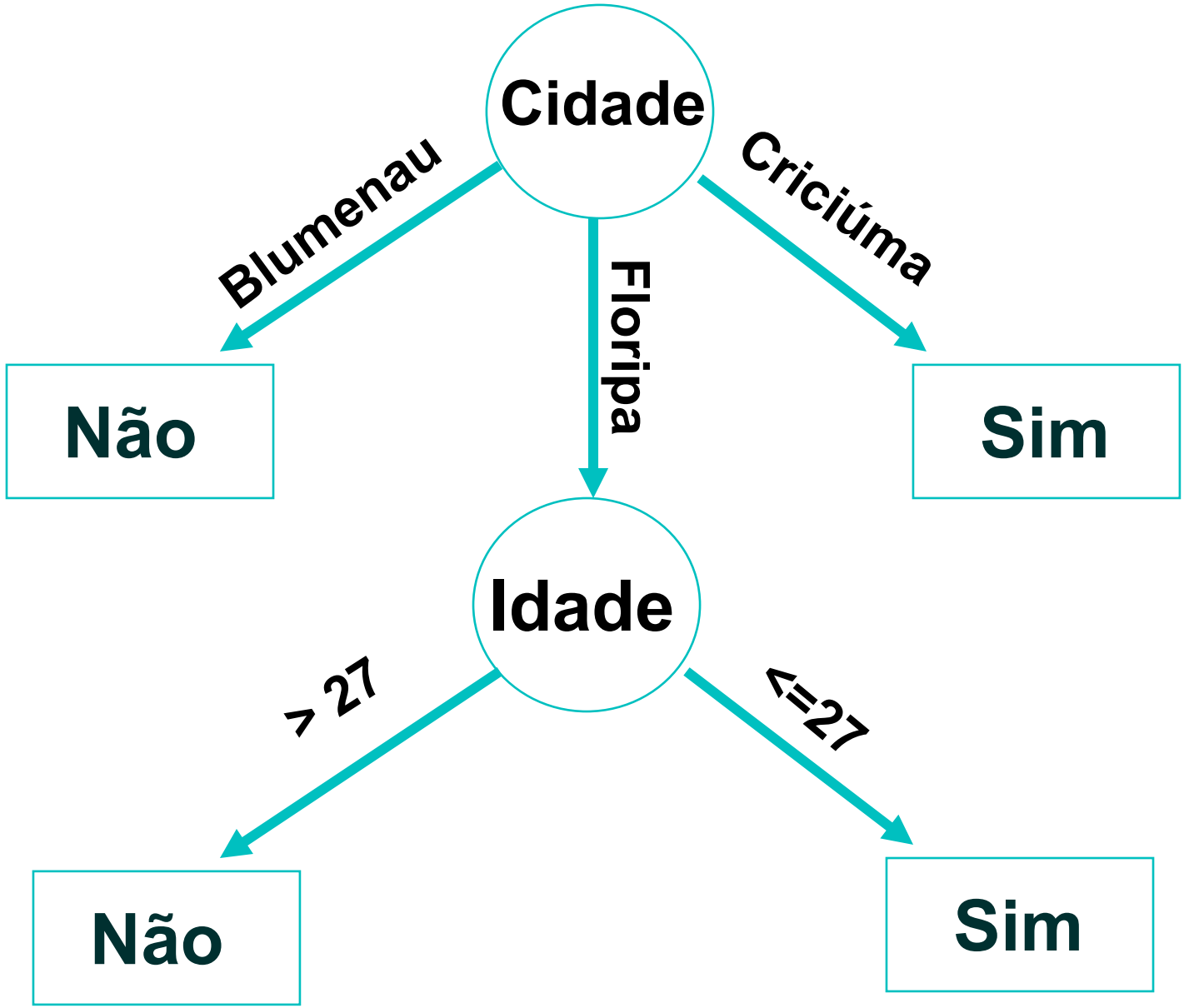
Casa Própria	Estado Civil	Rendim.	Mau pagador
N	Casado	80K	?

Atribua à classe (Mau Pagador) o valor NÃO

Exemplo de conjunto de dados

ID	Sexo	Cidade	Idade	Compra
1	M	Floripa	25	S
2	M	Criciuma	21	S
3	F	Floripa	23	S
4	F	Criciuma	34	S
5	F	Floripa	30	N
6	M	Blumenau	21	N
7	M	Blumenau	20	N
8	F	Blumenau	18	N
9	F	Floripa	34	N
10	M	Floripa	55	N

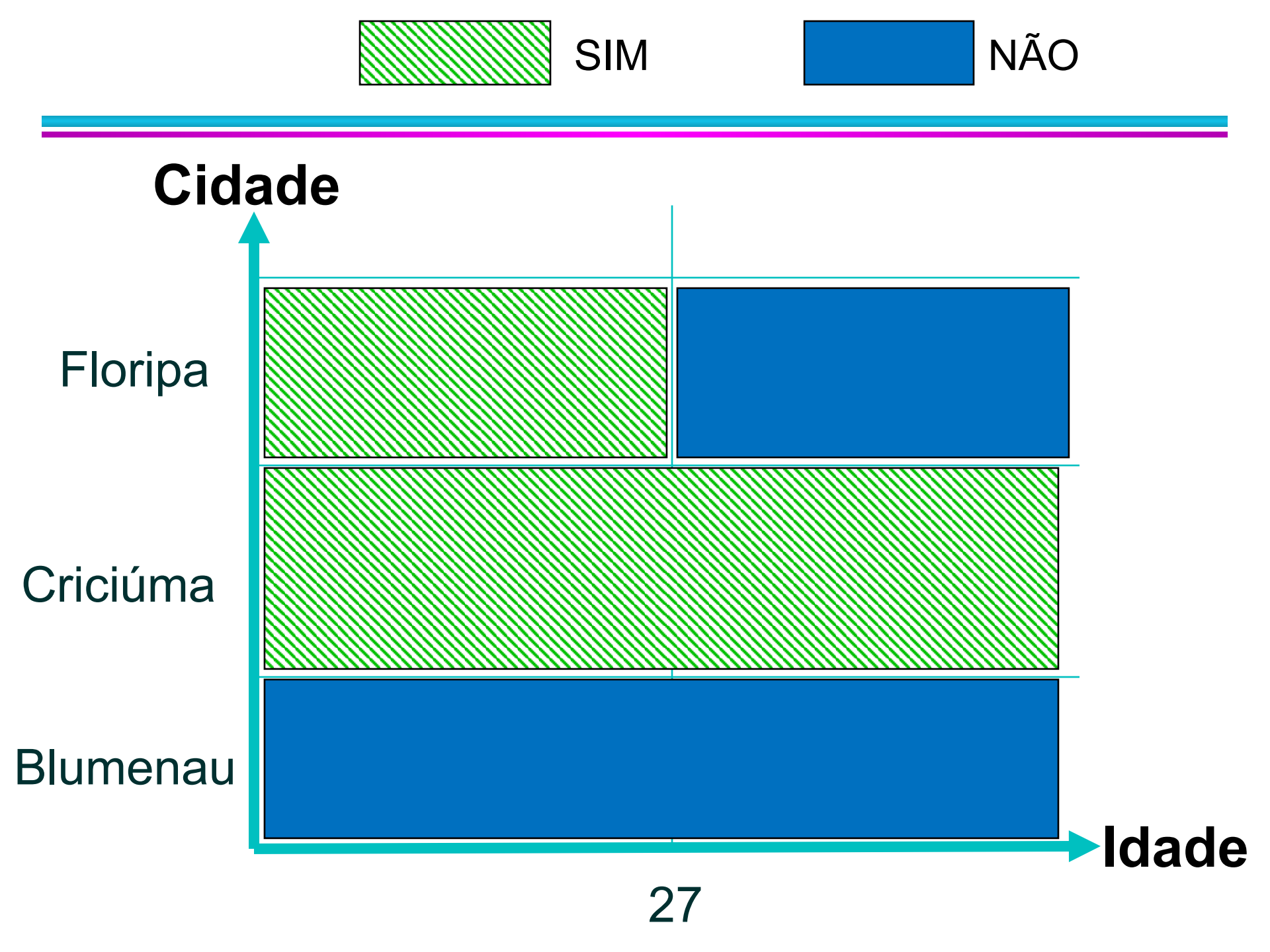
Árvore de Decisão: Exemplo



Árvores de Decisão

Os métodos baseados em árvores, dividem o espaço de entrada em **regiões disjuntas** para construir uma **fronteira de decisão**.

As regiões são escolhidas baseadas em uma otimização heurística onde a cada passo os algoritmos selecionam a variável que provê a melhor separação de classes de acordo com alguma função custo.


SIM
NÃO
Cidade
Floripa
Criciúma
Blumenau
Idade
27

Como criar uma árvore de decisão?

Exemplo usando o Algoritmo ID3

Algoritmo ID3 [Quinlam 1986]

O **ID3** é um algoritmo simples que constrói uma **árvore de decisão** sob as seguintes premissas:

- Cada **vértice** (nodo) corresponde a um **atributo**, e cada **aresta** da árvore a um **valor possível** do atributo.
- Uma **folha** da árvore corresponde ao valor esperado da **decisão** segundo os dados de treino utilizados (**classe**).

A **explicação** de uma determinada decisão está na **trajetória** que vai da raiz até a folha representativa desta decisão.

Algoritmo ID3

Passos para construção da árvore de decisão:

1. Seleciona um atributo como sendo o nodo raiz ;
2. Arcos são criados para todos os diferentes valores do atributo selecionado no passo 1;
3. Se todos os exemplos de treinamento (registros) sobre uma folha pertencerem a uma mesma classe, esta folha recebe o nome da classe. Se todas as folhas possuem uma classe, o algoritmo termina;
4. Senão, o nodo é determinado com um atributo que não ocorra no trajeto da raiz, e arcos são criados para todos os valores. O algoritmo retorna ao passo 3.

Algoritmo ID3

A seleção dos nodos a serem utilizados na árvore é baseada na Teoria da Informação de Shannon, mais especificamente nos conceitos de ***entropia*** *e* ***ganho de informação***

Entropia

Quantidade necessária de informação para identificar a classe de um caso

$$Entropia(S) = -(p_1 \log_2 p_1 + p_2 \log_2 p_2 + \ldots + p_n \log_2 p_n)$$

onde:

S é o conjunto de amostras (registros)

n é o número de valores possíveis da classe

$\mathbf{p_i}$ é a proporção de amostras da classe *i* em relação ao total de amostras

Entropia

Considerando apenas **2 valores possíveis da classe**, a entropia é dada pela fórmula:

Entropia (S) = $- (p_+ \log_2 p_+ + p_- \log_2 p_-)$

Onde:

S é a totalidade de amostras do conjunto

p_+ é a proporção de amostras positivas

p_- é a proporção de amostras negativas

Exemplo:

Se S é uma coleção de 14 exemplos com 9 instâncias positivas (classe=sim) e 5 negativas (classe=não), então:

Entropia (S) = $- (9/14) \text{Log}_2 (9/14) – (5/14) \text{Log}_2 (5/14) = 0.940$

Entropia

Exemplos:

$P= (p_+ ; p_-)$

P = (0.5 ; 0.5) ➔ entropia(P) = 1;

P = (0.67 ; 0.33) ➔ entropia(P) = 0.92;

P = (1.0 ; 0.0) ➔ entropia(P) = 0.0;

Site com applet de logaritmos:

http://www.math.utah.edu/~pa/math/Log.html

Ganho de Informação

Ganho de informação

É a redução esperada da entropia ao utilizarmos um atributo na árvore

O ganho de informação é dado por:

Ganho (S, A) = Entropia (S) - $\sum$ ((|Sv| / |S|) * Entropia (Sv))

Onde:
Ganho (S, A) é o ganho do atributo A sobre o conjunto S
Sv = subconjunto de S para um valor do atributo A
|Sv| = número de elementos de Sv
|S| = número de elementos de S

Exemplo de dados para concessão de empréstimo bancário

caso	montante	idade	salário	conta	empréstimo
1	médio	sênior	baixo	sim	***não***
2	médio	sênior	baixo	não	***não***
3	baixo	sênior	baixo	sim	***sim***
4	alto	média	baixo	sim	***sim***
5	alto	jovem	alto	sim	***sim***
6	alto	jovem	alto	não	***não***
7	baixo	jovem	alto	não	***sim***
8	médio	média	baixo	sim	***não***
9	médio	jovem	alto	sim	***sim***
10	alto	média	alto	sim	***sim***
11	médio	média	alto	não	***sim***
12	baixo	jovem	baixo	não	***sim***
13	baixo	sênior	alto	sim	***sim***
14	alto	média	baixo	não	***não***

ID3: Nodo raiz

Selecionando o melhor atributo:

Entropia(S) = - 9/14 log_2 (9/14) - 5/14 log_2 (5/14) = 0,940

caso	montante	idade	salário	conta	empréstimo
1	médio	sênior	baixo	sim	***não***
2	médio	sênior	baixo	não	***não***
3	baixo	sênior	baixo	sim	***sim***
4	alto	média	baixo	sim	**sim**
5	alto	jovem	alto	sim	**sim**
6	alto	jovem	alto	não	***não***
7	baixo	jovem	alto	não	***sim***
8	médio	média	baixo	sim	***não***
9	médio	jovem	alto	sim	***sim***
10	alto	média	alto	sim	***sim***
11	médio	média	alto	não	***sim***
12	baixo	jovem	baixo	não	***sim***
13	baixo	sênior	alto	sim	***sim***
14	alto	média	baixo	não	***não***

Amarelo = classe ***não***
Verde = classe ***sim***

Selecionando o melhor atributo:

Entropia(montante=médio) = - 2/5 $\log_2$ (2/5) - 3/5 $\log_2$ (3/5) = 0,971

caso	montante	idade	salário	conta	empréstimo
1	médio	sênior	baixo	sim	***não***
2	médio	sênior	baixo	não	***não***
3	baixo	sênior	baixo	sim	***sim***
4	alto	média	baixo	sim	**sim**
5	alto	jovem	alto	sim	**sim**
6	alto	jovem	alto	não	***não***
7	baixo	jovem	alto	não	***sim***
8	médio	média	baixo	sim	***não***
9	médio	jovem	alto	sim	***sim***
10	alto	média	alto	sim	***sim***
11	médio	média	alto	não	***sim***
12	baixo	jovem	baixo	não	***sim***
13	baixo	sênior	alto	sim	***sim***
14	alto	média	baixo	não	***não***

Selecionando o melhor atributo:

Entropia(montante=médio) = - 2/5 $\log_2$ (2/5) - 3/5 $\log_2$ (3/5) = 0,971

Entropia(montante=baixo) = - 4/4 $\log_2$ (4/4) - 0/4 $\log_2$ (0/4) = 0

caso	montante	idade	salário	conta	empréstimo
1	médio	sênior	baixo	sim	***não***
2	médio	sênior	baixo	não	***não***
3	baixo	sênior	baixo	sim	***sim***
4	alto	média	baixo	sim	**sim**
5	alto	jovem	alto	sim	**sim**
6	alto	jovem	alto	não	***não***
7	baixo	jovem	alto	não	***sim***
8	médio	média	baixo	sim	***não***
9	médio	jovem	alto	sim	***sim***
10	alto	média	alto	sim	***sim***
11	médio	média	alto	não	***sim***
12	baixo	jovem	baixo	não	***sim***
13	baixo	sênior	alto	sim	***sim***
14	alto	média	baixo	não	***não***

Selecionando o melhor atributo:

Entropia(montante=médio) = - 2/5 $\log_2$ (2/5) - 3/5 $\log_2$ (3/5) = 0,971

Entropia(montante=baixo) = - 4/4 $\log_2$ (4/4) - 0/4 $\log_2$ (0/4) = 0

Entropia(montante=alto) = - 3/5 $\log_2$ (3/5) - 2/5 $\log_2$ (2/5) = 0,971

caso	montante	idade	salário	conta	empréstimo
1	médio	sênior	baixo	sim	***não***
2	médio	sênior	baixo	não	***não***
3	baixo	sênior	baixo	sim	***sim***
4	alto	média	baixo	sim	**sim**
5	alto	jovem	alto	sim	**sim**
6	alto	jovem	alto	não	***não***
7	baixo	jovem	alto	não	***sim***
8	médio	média	baixo	sim	***não***
9	médio	jovem	alto	sim	***sim***
10	alto	média	alto	sim	***sim***
11	médio	média	alto	não	***sim***
12	baixo	jovem	baixo	não	***sim***
13	baixo	sênior	alto	sim	***sim***
14	alto	média	baixo	não	***não***

Selecionando o melhor atributo:

$$Entropia(S) = - 9/14 \log_2 (9/14) - 5/14 \log_2 (5/14) = 0,940$$

$$Entropia(montante=médio) = - 2/5 \log_2 (2/5) - 3/5 \log_2 (3/5) = 0,971$$

$$Entropia(montante=baixo) = - 4/4 \log_2 (4/4) - 0/4 \log_2 (0/4) = 0$$

$$Entropia(montante=alto) = - 3/5 \log_2 (3/5) - 2/5 \log_2 (2/5) = 0,971$$

$$Entropia\ (idade = senior) = - 2/4 \log_2 (2/4) - 2/4 \log_2 (2/4) = 1$$

$$Entropia\ (idade = média) = - 3/5 \log_2 (3/5) - 2/5 \log_2 (2/5) = 0,971$$

$$Entropia\ (idade = jovem) = - 4/5 \log_2 (4/5) - 1/5 \log_2 (1/5) = 0,722$$

.....

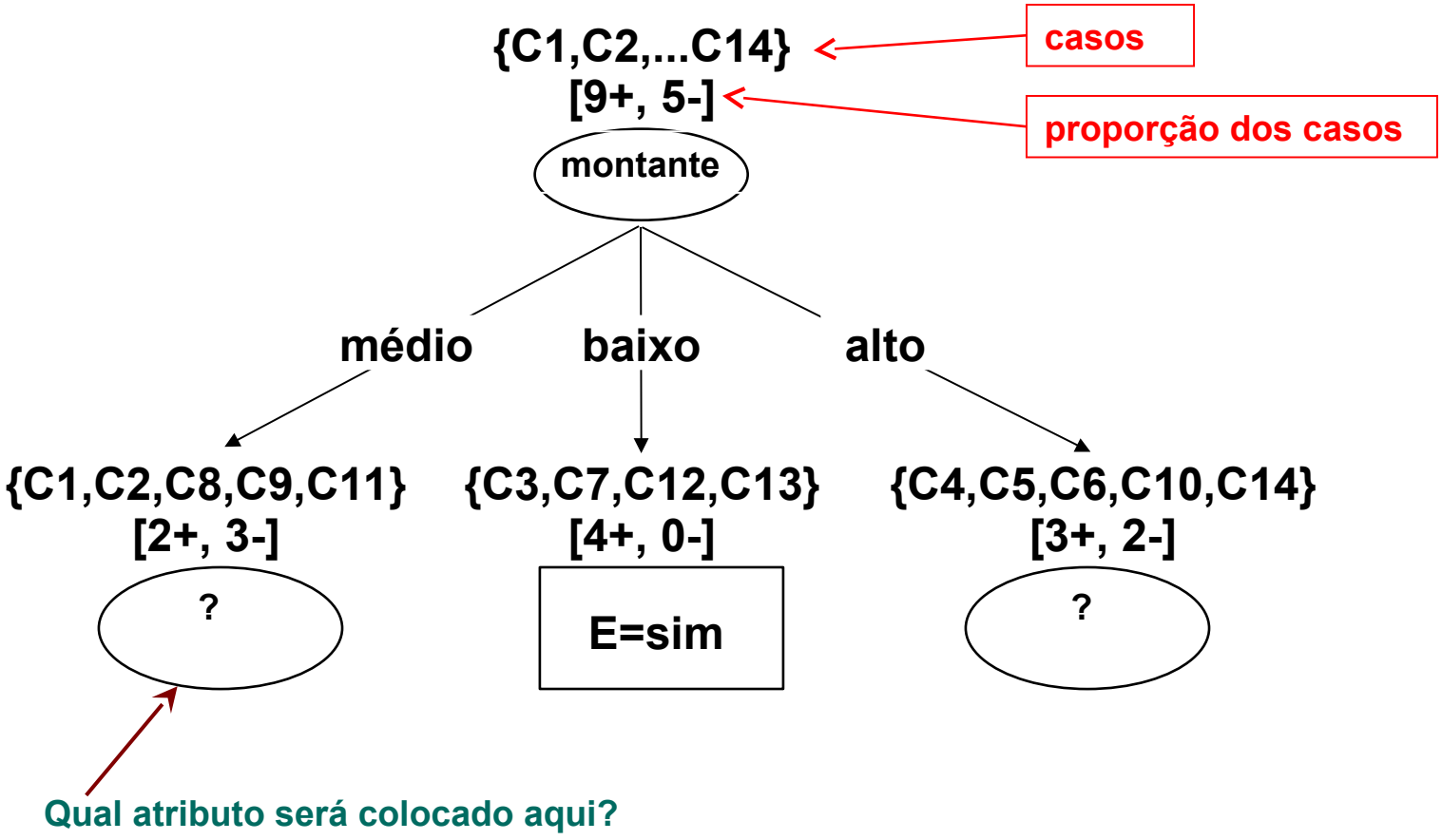
$$Ganho\ (S,montante) = 0,940 - (5/14).\ 0,971 - (4/14).\ 0 - (5/14).\ 0,971 = 0,246$$

$$Ganho\ (S,idade) = 0,940 - (4/14).\ 1 - (5/14).\ 0,971 - (5/14).\ 0,722 = 0,049$$

$$Ganho\ (S,salário) = 0,940 - (7/14).\ 0,592 - (7/14).\ 0,985 = 0,151$$

$$Ganho\ (S,conta) = 0,940 - (8/14).\ 0,811 - (6/14).\ 1 = 0,047$$

Escolha do próximo atributo



caso	montante	idade	salário	conta	empréstimo
1	médio	sênior	baixo	sim	***não***
2	médio	sênior	baixo	não	***não***
3	baixo	sênior	baixo	sim	***sim***
4	alto	média	baixo	sim	***sim***
5	alto	jovem	alto	sim	***sim***
6	alto	jovem	alto	não	***não***
7	baixo	jovem	alto	não	***sim***
8	médio	média	baixo	sim	***não***
9	médio	jovem	alto	sim	***sim***
10	alto	média	alto	sim	***sim***
11	médio	média	alto	não	***sim***
12	baixo	jovem	baixo	não	***sim***
13	baixo	sênior	alto	sim	***sim***
14	alto	média	baixo	não	***não***

Escolha do próximo atributo

Qual é o melhor atributo?

$S_{médio}$ = {C1,C2,C8,C9,C11}

Entropia($S_{médio}$) = - 2/5 $\log_2$ (2/5) - 3/5 $\log_2$ (3/5) = 0,971

Entropia(idade=senior)= 0

Entropia(idade=média)= 1

Entropia(idade=jovem)= 0

Entropia(salário=baixo)= 0

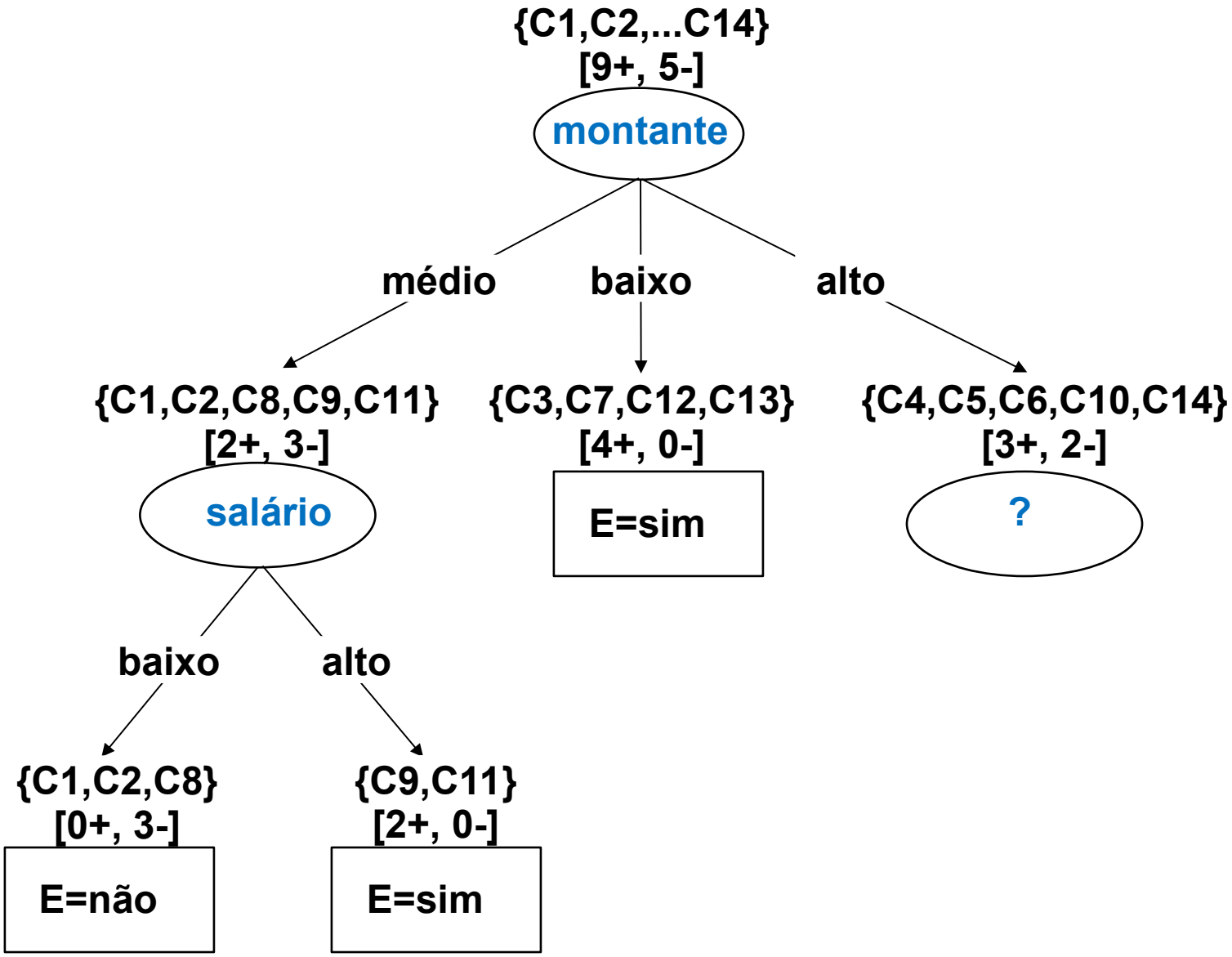
Entropia(salário=alto)= 0

Entropia ….

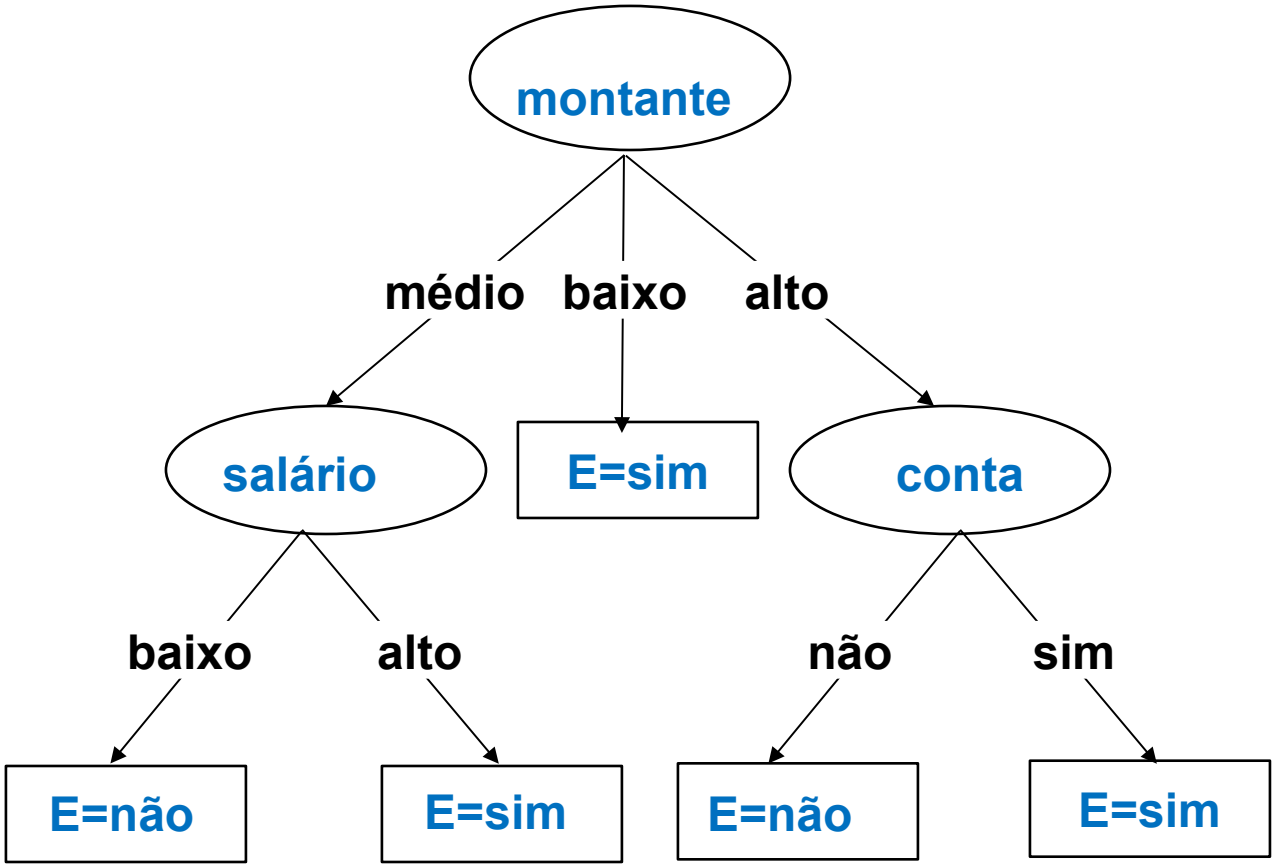
Ganho ($S_{médio}$, idade) = 0,971 - (2/5).0 - (2/5).1 - (1/5).0 = 0,571

Ganho ($S_{médio}$, salário) = 0,971 - (3/5).0 - (2/5).0 = 0,971

Ganho ($S_{médio}$, conta) = 0,971 - (3/5).0,918 - (2/5).1= 0,020


{C1,C2,...C14}
[9+, 5-]
montante
médio
baixo
alto
{C1,C2,C8,C9,C11}
[2+, 3-]
salário
{C3,C7,C12,C13}
[4+, 0-]
E=sim
{C4,C5,C6,C10,C14}
[3+, 2-]
?
baixo
alto
{C1,C2,C8}
[0+, 3-]
E=não
{C9,C11}
[2+, 0-]
E=sim

Resultado: modelo de classificação



Classificação baseada em árvores de decisão

- Construção barata
- Extremamente rápido para classificar novos registros
- Fácil interpretação de árvores pequenas
- A acurácia é comparável a outros métodos de classificação para muitos conjuntos de dados

Algoritmo C 4.5 [Quinlan 1993]

Algoritmo C 4.5

O **C 4.5** é uma extensão do **ID3**:

1. Constrói árvores de decisão, com valores **desconhecidos** para alguns **atributos**.
2. Trabalha com **atributos** que apresentam **valores contínuos**.
3. Utiliza o conceito de **poda (pruning)** de árvores.
4. Gera **regras de decisão** a partir da árvore gerada

Algoritmo C 4.5: valor desconhecido de atributo

Quando existem **valores desconhecidos** para algum atributo, os mesmos são considerados como um novo valor do atributo, por exemplo o valor “desconhecido”.

Algoritmo C 4.5: atributos contínuos

- Quando existem atributos com **valores contínuos**:
  1. os registros são classificados pelo atributo contínuo
  2. o algoritmo cria intervalos segundo as alterações na variável de decisão (classe).
  3. O ganho de informação é calculado para cada intervalo

Algoritmo C 4.5

atributo	classe
64	S
65	N
68	S
69	S
70	S
71	N
72	N
73	S
75	S
75	S
80	N
81	S
83	S
85	N

Algoritmo C 4.5

atributo	classe
64	S
65	N
68	S
69	S
70	S
71	N
72	N
73	S
75	S
75	S
80	N
81	S
83	S
85	N

64.5

66.5

70.5

72.5

77.5

80.5

84

Algoritmo C 4.5

Exemplo:

entropia(S) = - 9/14 log_2 (9/14) - 5/14 log_2 (5/14) = 0,940

entropia(V<=70.5) = - 4/5 log_2(4/5) – 1/5 log_2(1/5)= 0,722

entropia(V>70.5)= - 5/9 log_2(5/9) – 4/9 log_2(4/9)= 0,991

ganho(S,70.5)=0,940 – 5/14 . 0,722 – 9/14 . 0,991 = 0,027

Esse cálculo é repetido para cada um dos intervalos, e escolhe-se o de maior ganho para comparar com os outros atributos, para decidir qual atributo será considerado.

Variável	Decisão
64	S
65	N
68	S
69	S
70	S
71	N
72	N
73	S
75	S
75	S
80	N
81	S
83	S
85	N

Algoritmo C 4.5: pruning (poda da árvore)

Poda da árvore, significa substituir uma **parte da árvore** (sub-árvore) por uma **folha**, com o objetivo de simplificar as regras de decisão.

A poda tem lugar quando o **valor esperado** do **erro da sub-árvore** é **maior do que o erro** da **folha** que fica em seu lugar.

Algoritmo C 4.5 : Pruning

Erro Esperado (Nó) = (N - n + k -1) /(N + k)

onde :

N = Número de exemplos do nó

n = Número de exemplos de N pertencentes à classe com o maior número de elementos

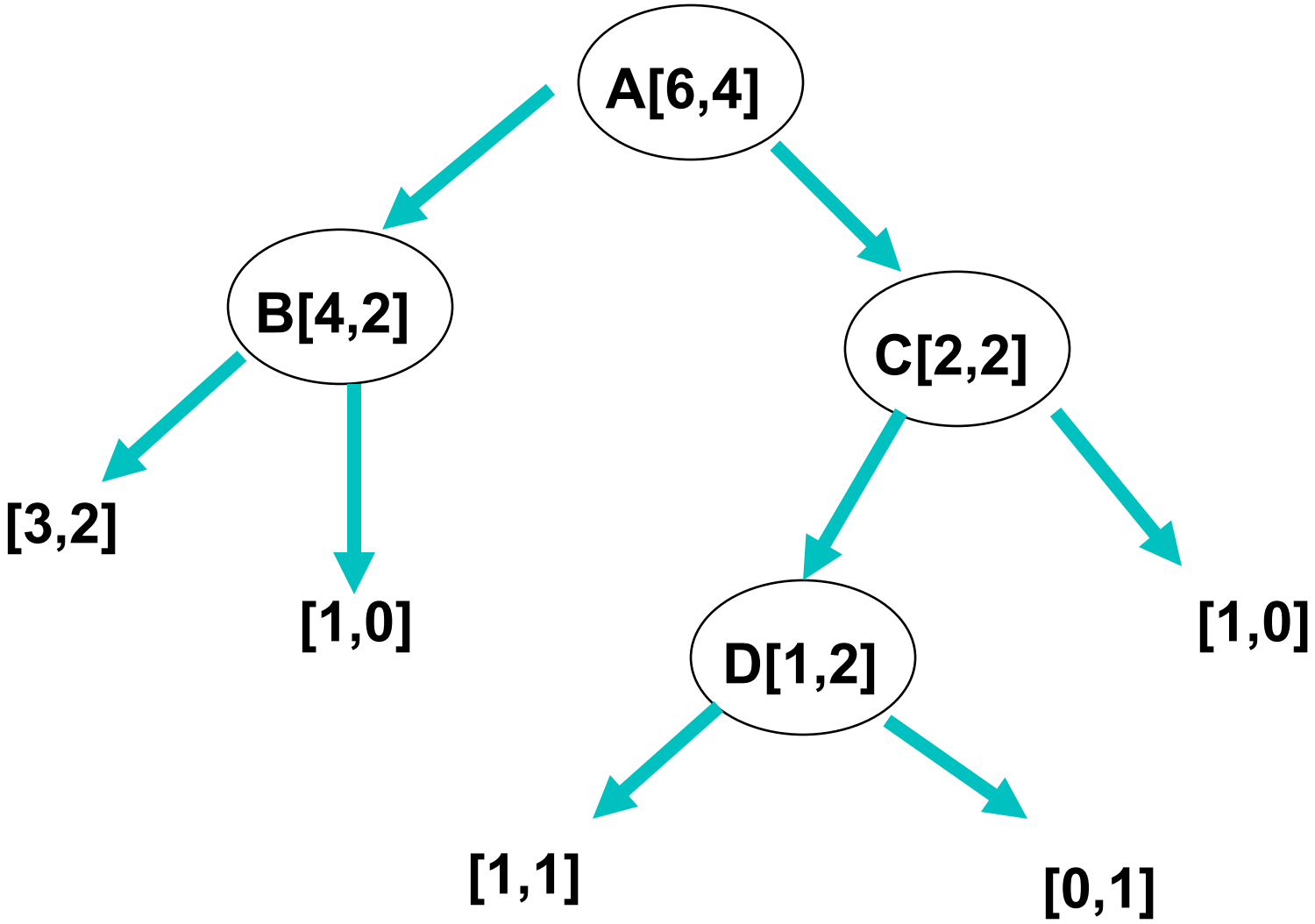
k = Número de classes

Erro (Sub-árvore) = Σ_i (P_i* Erro (Nó i))

onde:

P_i = proporção de exemplos do Nó *i* em relação ao total de exemplos da sub-árvore considerada

Algoritmo C 4.5 : Pruning



Algoritmo C 4.5 : Pruning

Erro Esperado (Nó) = (N - n + k -1) /(N + k)

onde :

N = Número de exemplos do nó

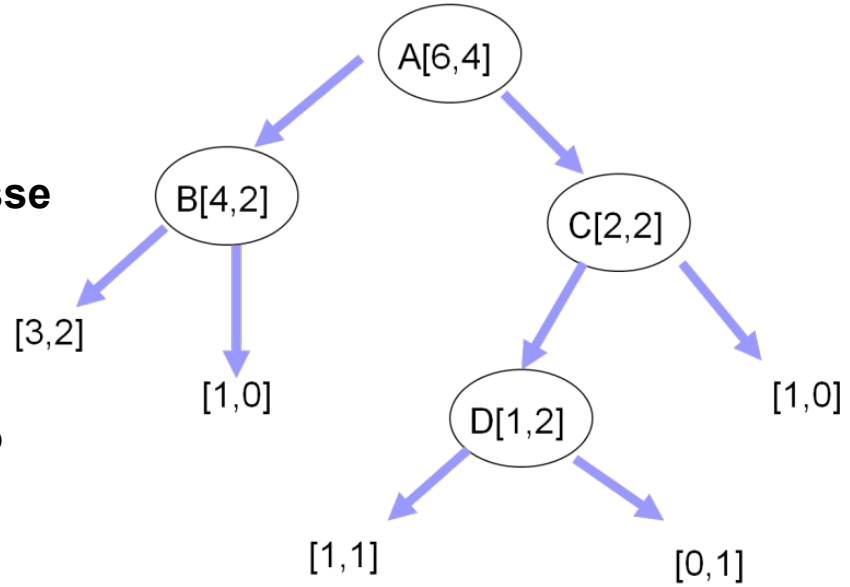
k = Número de classes

n = Número de exemplos de N pertencentes à classe com o maior número de elementos

Erro (Sub-arvore) = Σ_i (P_i* Erro (Nó i))

onde:

P_i = proporção de exemplos do Nó *i* em relação ao total de exemplos da sub-árvore considerada



Erro (B) = (6 - 4 + 2 -1) / (6 + 2) = **0.375**

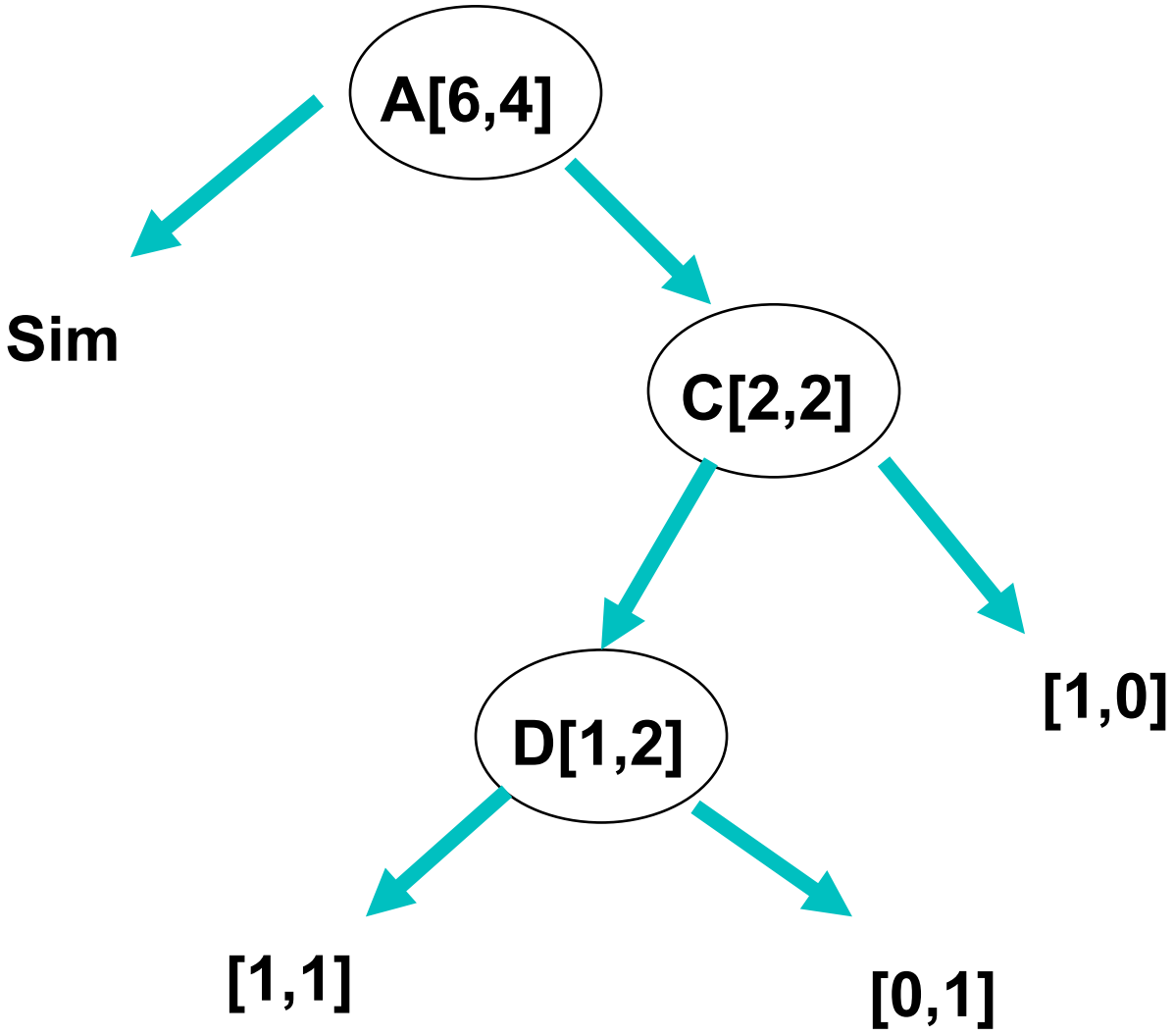
Erro (Filho 1) = (5 - 3 + 2 -1) / (5 + 2) = 0.429

Erro (Filho 2) = (1 - 1 + 2 -1) / (1 + 2) = 0.333

Erro (SubArv) = 5/6 * 0.429 + 1/6 * 0.333 = **0.413**

Erro (B) < Erro (SubArv) então **Poda**

Algoritmo C 4.5 : Pruning



Algoritmo C 4.5 : Pruning

Erro Esperado (Nó) = (N - n + k -1) /(N + k)

onde :

N = Número de exemplos do nó

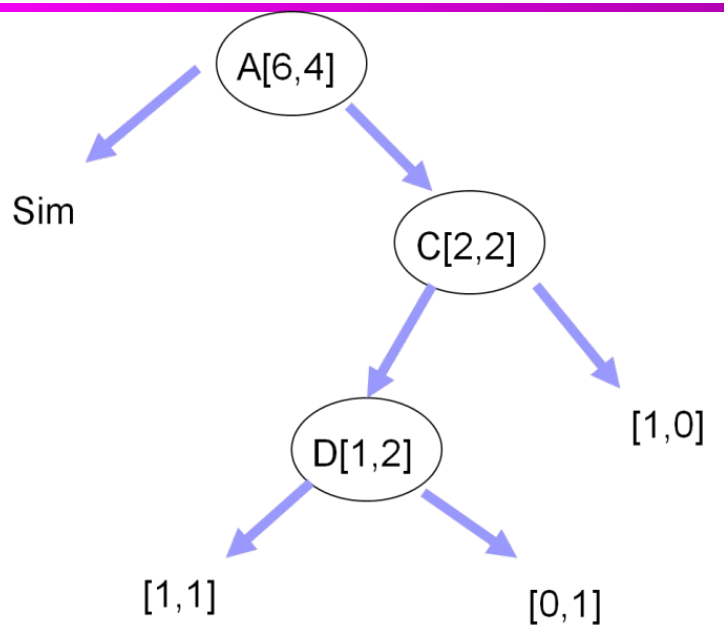
k = Número de classes

n = Número de exemplos de N pertencentes à classe com o maior número de elementos

Erro (Sub-arvore) = Σ_i (P_i* Erro (Nó i))

onde:

P_i = proporção de exemplos do Nó *i* em relação ao total de exemplos da sub-árvore considerada



Erro (D) = (3 - 2 + 2 -1) / (3 + 2) = **0.4**

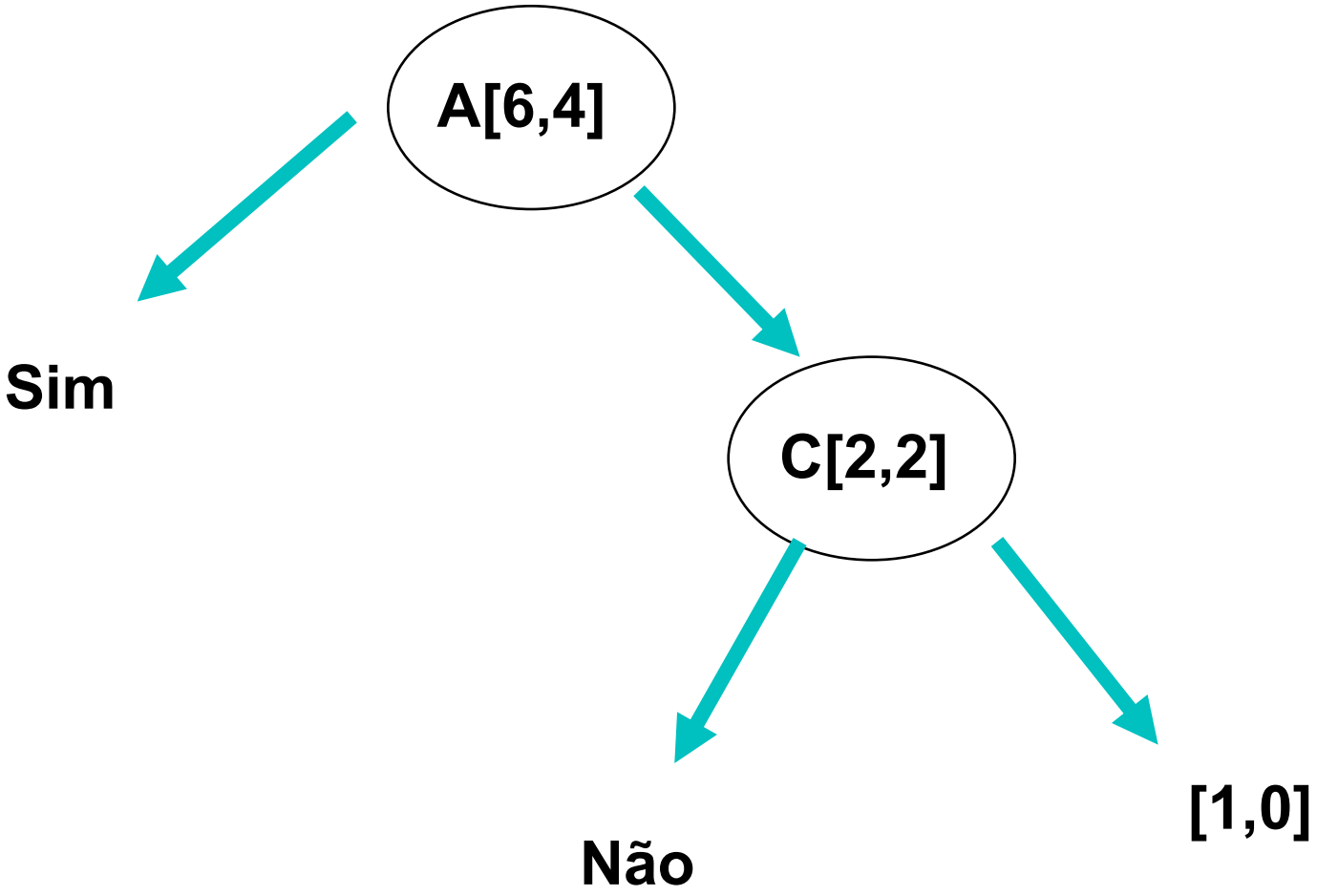
Erro (Filho 1) = (2 - 1 + 2 -1) / (2 + 2) = 0.5

Erro (Filho 2) = (1 - 1 + 2 -1) / (1 + 2) = 0.33

Erro (SubArv) = 2/3 * 0.5 + 1/3 * 0.33 = **0.444**

Erro (D) < Erro (SubArv) então **Poda**

Algoritmo C 4.5 : Pruning



Algoritmo C 4.5 : Pruning

Erro Esperado (Nó) = (N - n + k -1) /(N + k)

onde :

N = Número de exemplos do nó

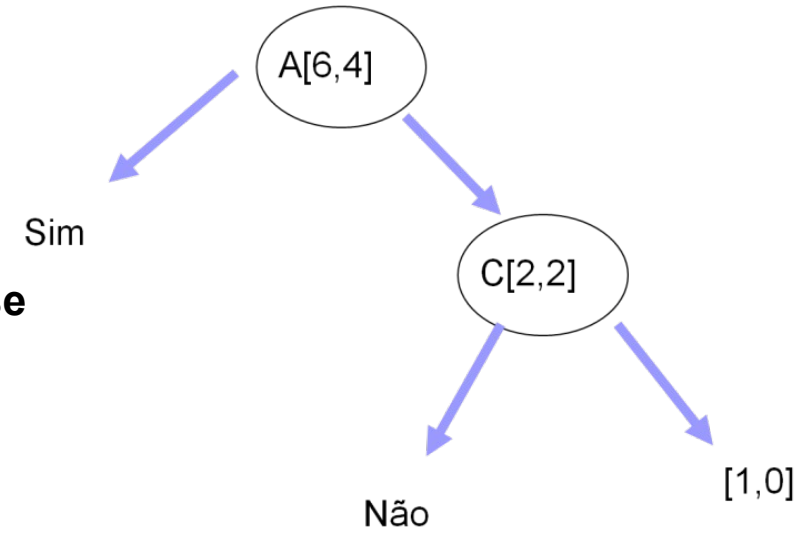
k = Número de classes

n = Número de exemplos de N pertencentes à classe com o maior número de elementos

Erro (Sub-arvore) = Σ_i (P_i* Erro (Nó i))

onde:

P_i = proporção de exemplos do Nó *i* em relação ao total de exemplos da sub-árvore considerada



Erro (C) = (4 - 2 + 2 -1) / (4 + 2) = **0.5**

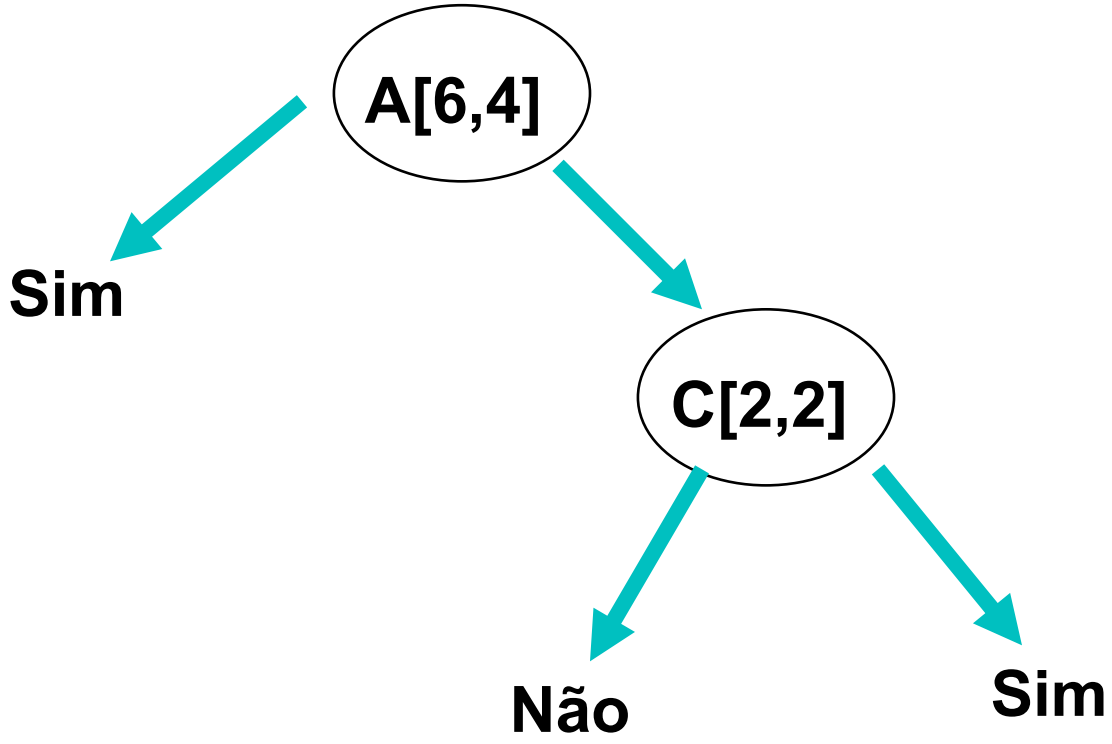
Erro (Filho 1) = Erro (D) = 0.4

Erro (Filho 2) = (1 - 1 + 2 -1) / (1 + 2) = 0.33

Erro (SubArv) = 3/4 * 0.4 + 1/4 * 0.33 = **0.3825**

Erro (C) > Erro(SubArv) Então **Não Poda**

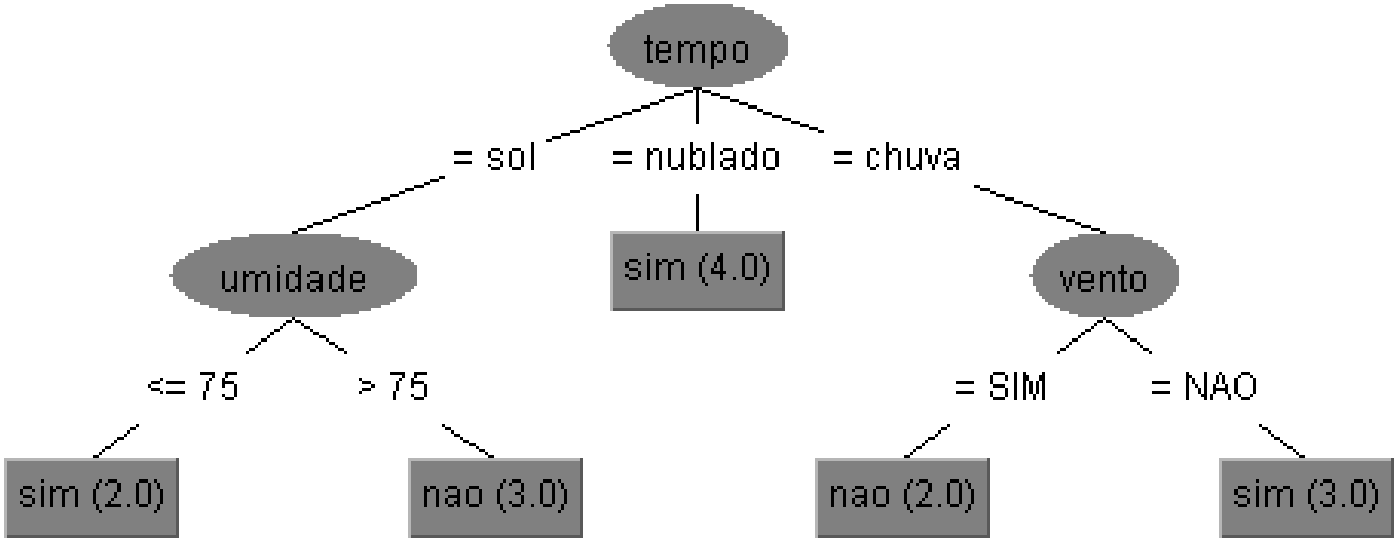
Algoritmo C 4.5 : Pruning



Algoritmo C 4.5: regras

O C 4.5 gera uma regra de decisão para cada caminho que vai do nodo raíz até um nodo folha.

Exemplo:



Se tempo=sol E umidade <=75% então jogo=sim

Se tempo=sol E umidade >75% então jogo=não

Se tempo=nublado então jogo=sim

Se tempo=chuva E vento=sim então jogo=não

Se tempo=chuva E vento=não então jogo=sim

Outras formas de escolher o atributo

- Além da entropia (ganho de informação), outras formas de escolher o próximo atributo a ser considerado na árvore são:
  - Índice de GINI
  - Erro de classificação

Divisão baseada no índice de GINI

- Índice de Gini para um nó t :

$$GINI(t) = 1 - \sum_j [p(j|t)]^2$$

(onde: $p(j \mid t)$ é a freqüência relativa da classe *j* no nó *t*).

- Máximo ($1 - 1/n_c$) quando os registros estão igualmente distribuídos entre todas as classes (pior)
- Mínimo (0.0) quando todos os registros pertencem a uma classe (melhor)

C1	**0**
C2	**6**
Gini=0.000	

C1	**1**
C2	**5**
Gini=0.278	

C1	**2**
C2	**4**
Gini=0.444	

C1	**3**
C2	**3**
Gini=0.500	

Divisão baseada em erro de classificação

- Erro de classificação no nó *t* :

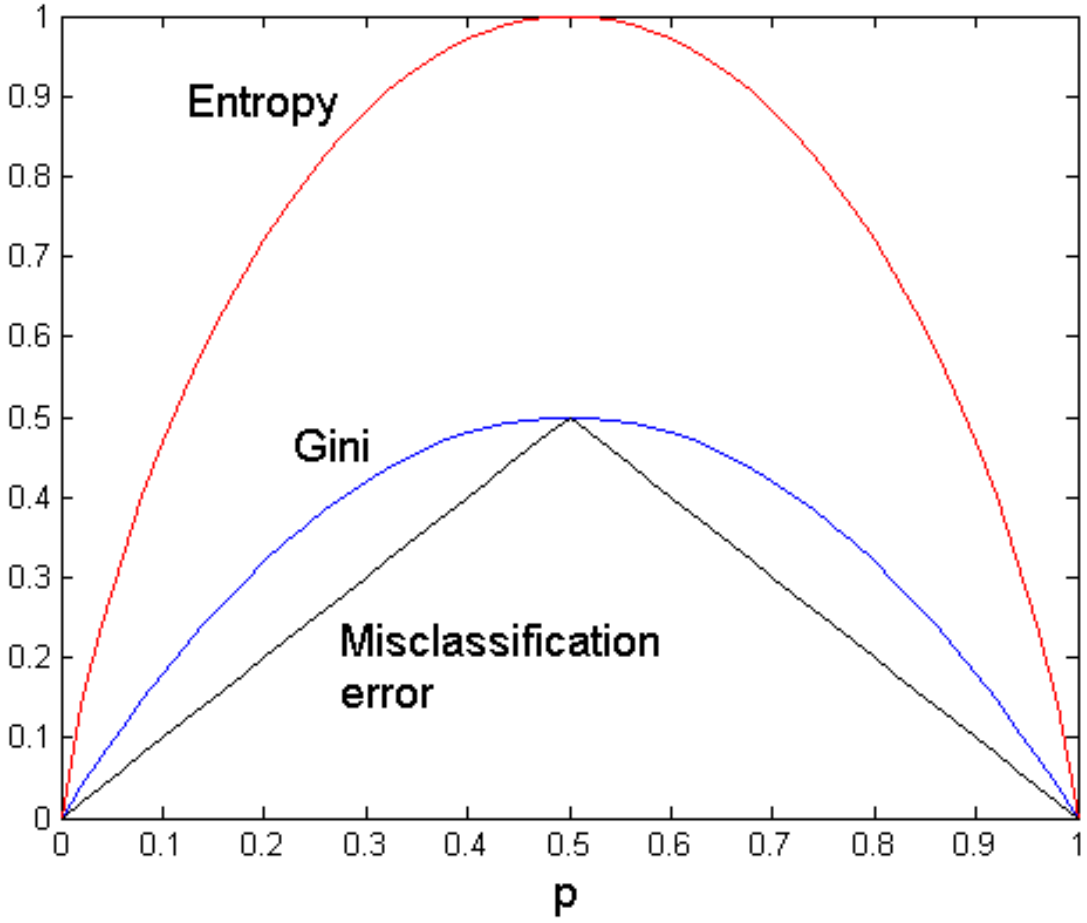
$$Error(t) = 1 - \max_i P(i|t)$$

(onde: $p(i \mid t)$ é a freqüência relativa da classe *i* no nó *t*).

- Mede o erro de classificação em um nó.
  - Máximo (1 - 1/n_c) quando os registros são igualmente distribuídos entre todas as classes (pior)
  - Mínimo (0.0) quando todos os registros pertencem à mesma classe (melhor)

Comparação entre os critérios de divisão

Para problemas com duas classes:



Referências:

Tan,P-N;Steimbach, M; Kumar,V. Introduction to Data Mining. Boston: Addison Wesley, 2006. 769p.

Quinlan, J. R. 1986. Induction of Decision Trees. Machine Learning. v1(1) (Mar. 1986), 81-106.

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