PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
ESCOLA POLITÉCNICA

Aprendizado de Máquina

Aprendizado Supervisionado III
Paradigma Simbólico

Prof. Me. Otávio Parraga

MALTA

Machine Learning Theory
and Applications Lab

Aula Passada

$$P(V|A) = \frac{P(A|V) \times P(V)}{P(A)}$$

Prior

Likelihood

Evidence

Posterior

Aula de Hoje

- Árvores de Decisão
  - Conceitos
  - Como classificar?
  - Como induzir?
    - Indução top-down
  - Medidas de Impureza
  - Critérios de Parada
  - Vantagens e Desvantagens
  - Árvores de Decisão para Problemas de Regressão

Árvores de Decisão

- Método para aproximar funções discretas ou contínuas, representadas por meio de um grafo acíclico direcionado

- Tal grafo pode ser representado por um conjunto de regras "`SE…ENTÃO`"
  – Compreensibilidade

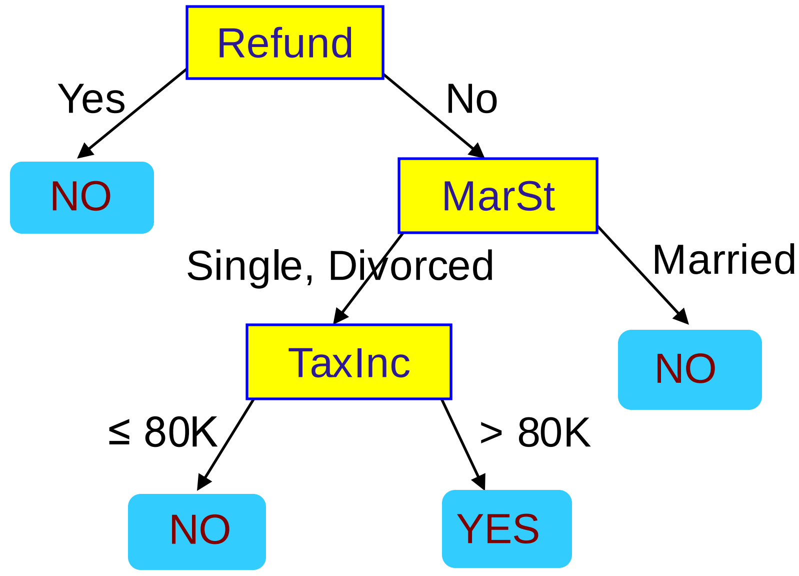
- Amplamente utilizado em aplicações práticas, principalmente em problemas de classificação

Exemplo de Árvore de Decisão

categorico categorico continuo classe

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Dados de Treino



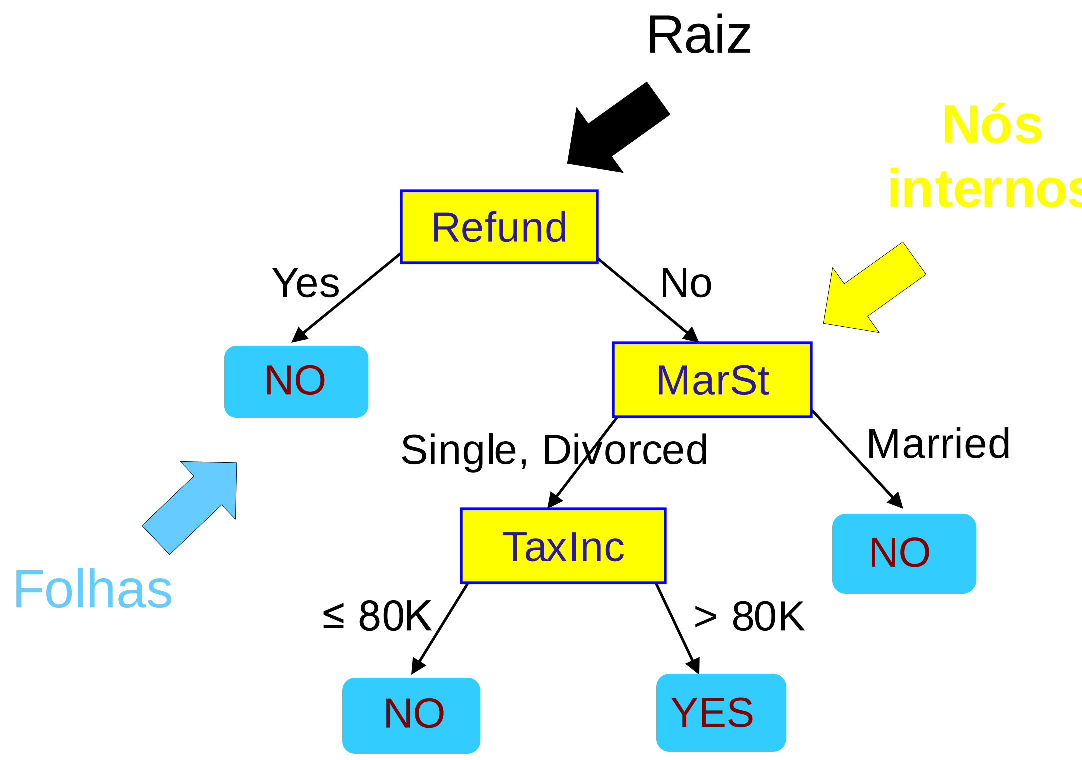
Modelo: Árvore de Decisão

Exemplo de Árvore de Decisão

categorico categorico continuo classe

Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

Dados de Treino



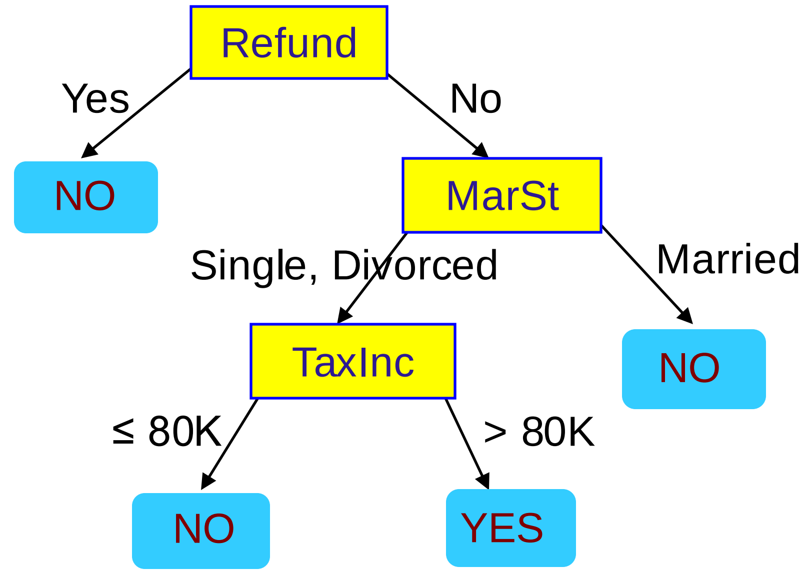
Modelo: Árvore de Decisão

Exemplo de Árvore de Decisão

categorico categorico continuo classe

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Dados de Treino



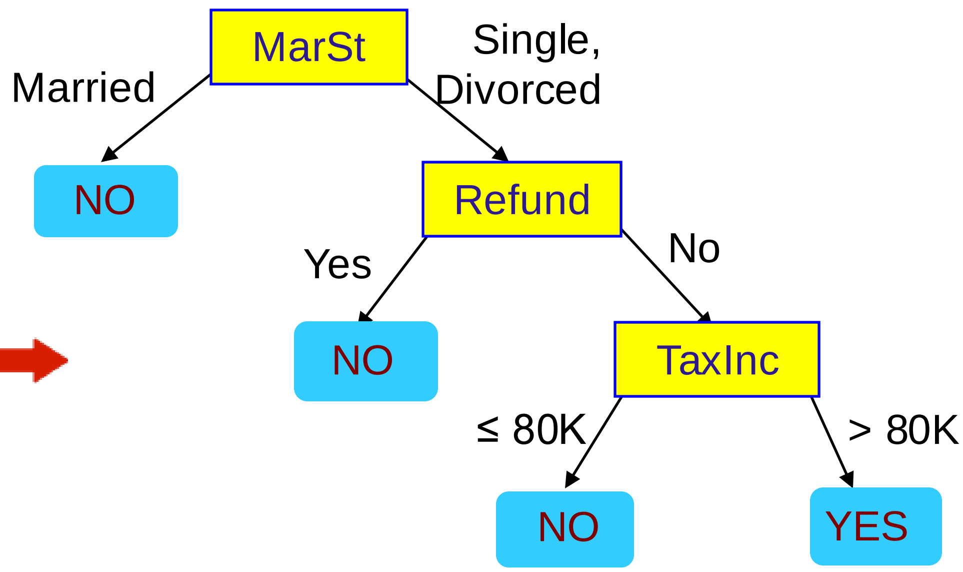
Perfeitamente ajustada aos dados de treino

Exemplo de Árvore de Decisão

categorico categorico continuo classe

Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

Dados de Treino

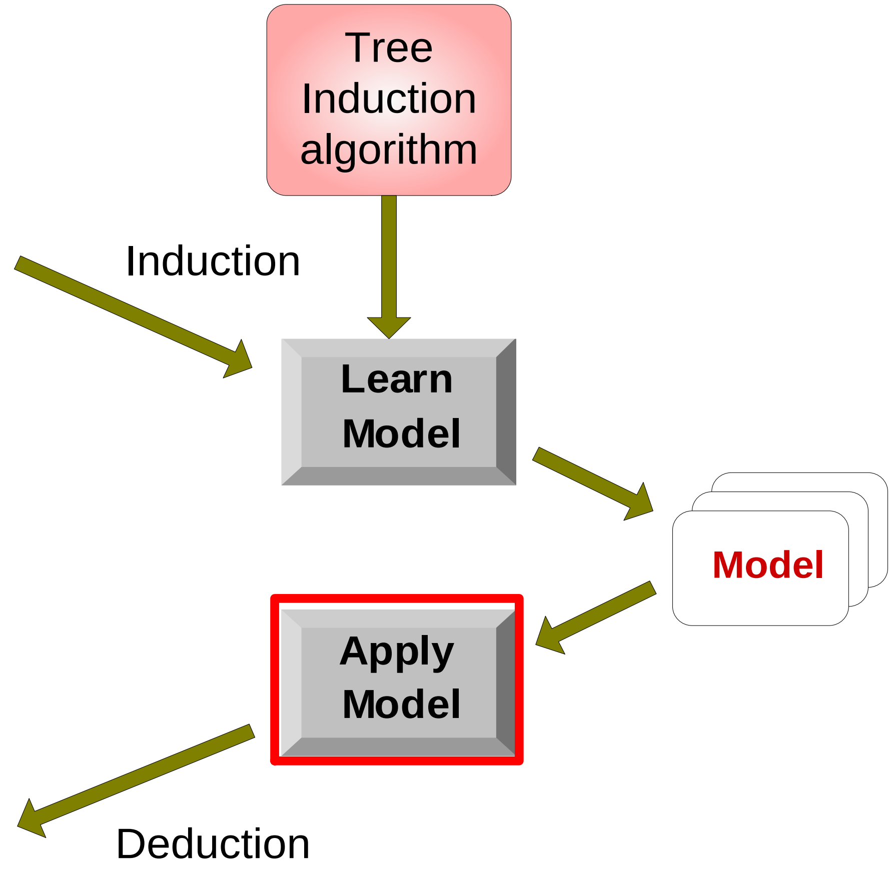


Note que várias árvores podem ser ajustadas aos mesmos dados!

Classificação com Árvore 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

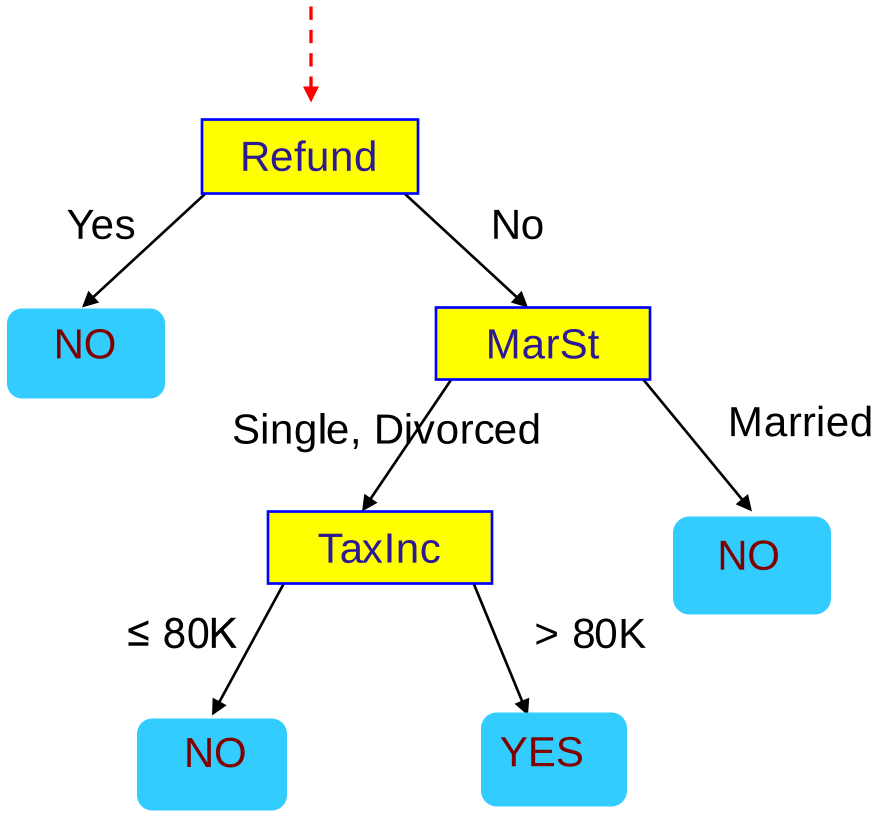


Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Aplicação do Modelo

Começar pela raiz da árvore



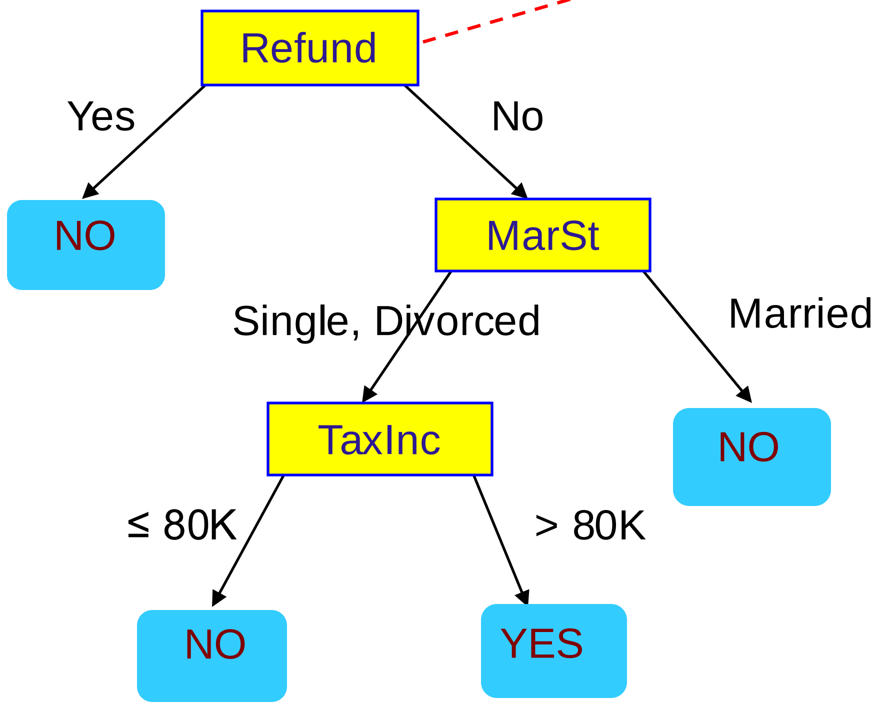
Instância de teste

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Aplicação do Modelo

Instância de teste

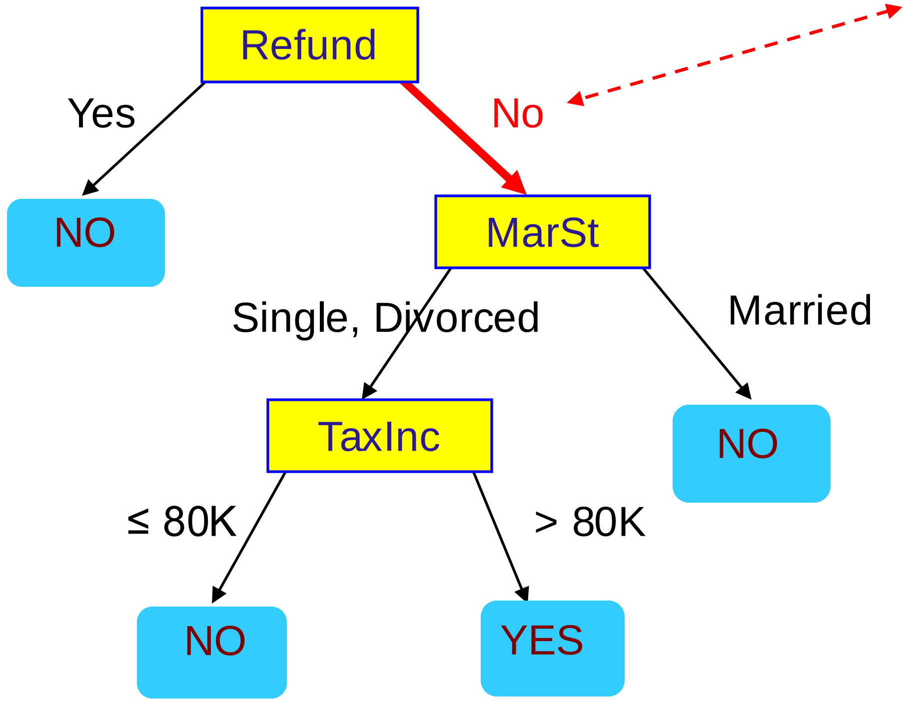
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Aplicação do Modelo

Instância de teste

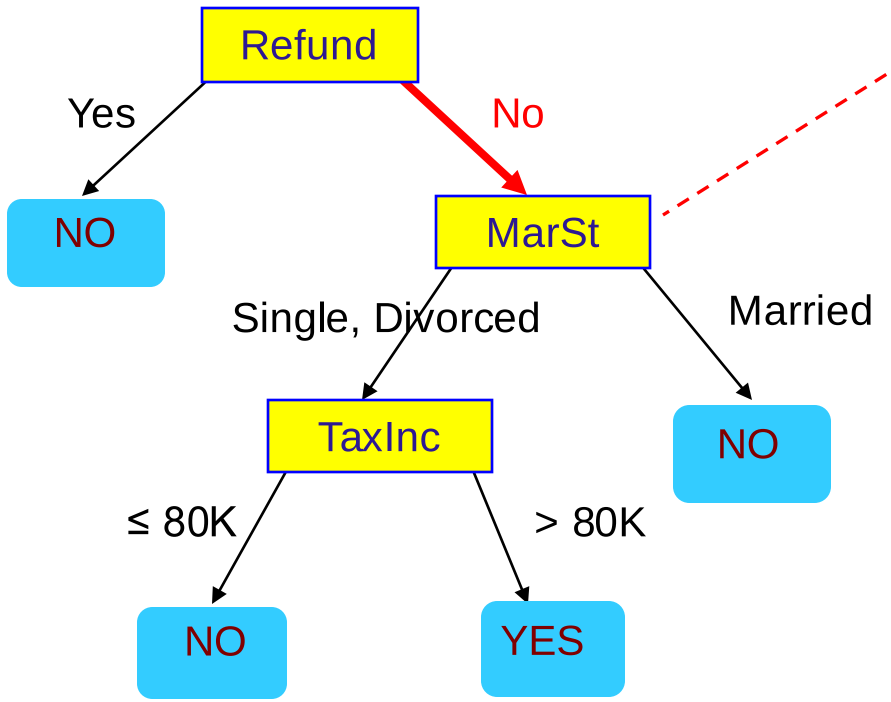
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Aplicação do Modelo

Instância de teste

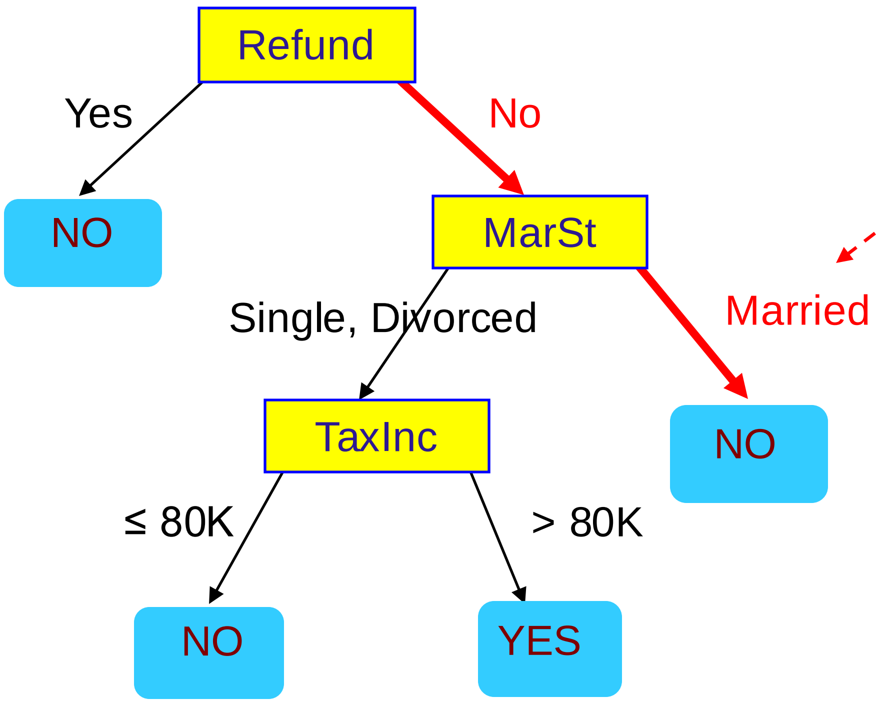
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Aplicação do Modelo

Instância de teste

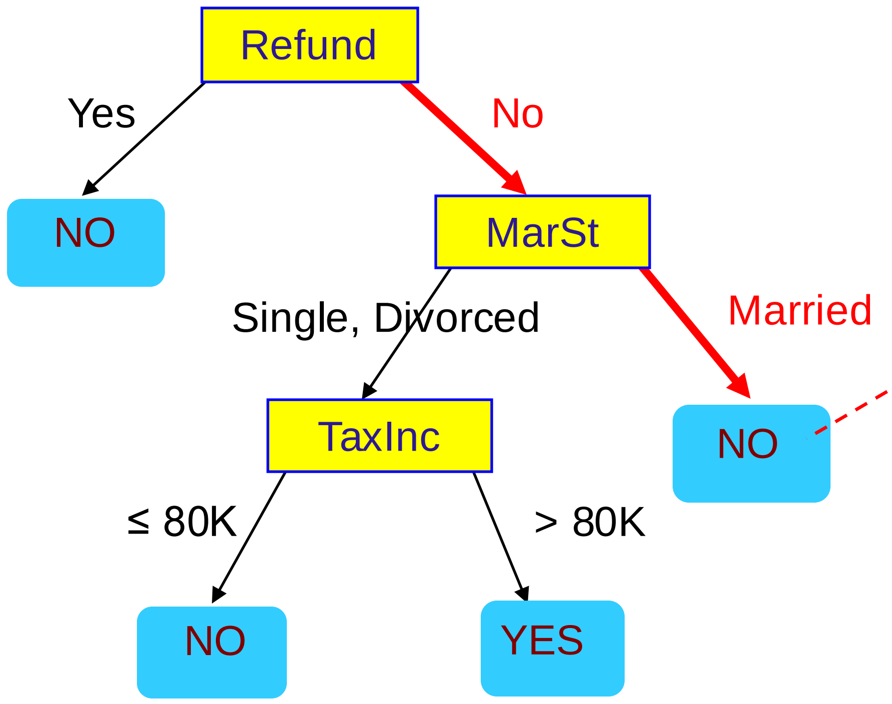
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Aplicação do Modelo

Instância de teste

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Atribuir instância à classe NO

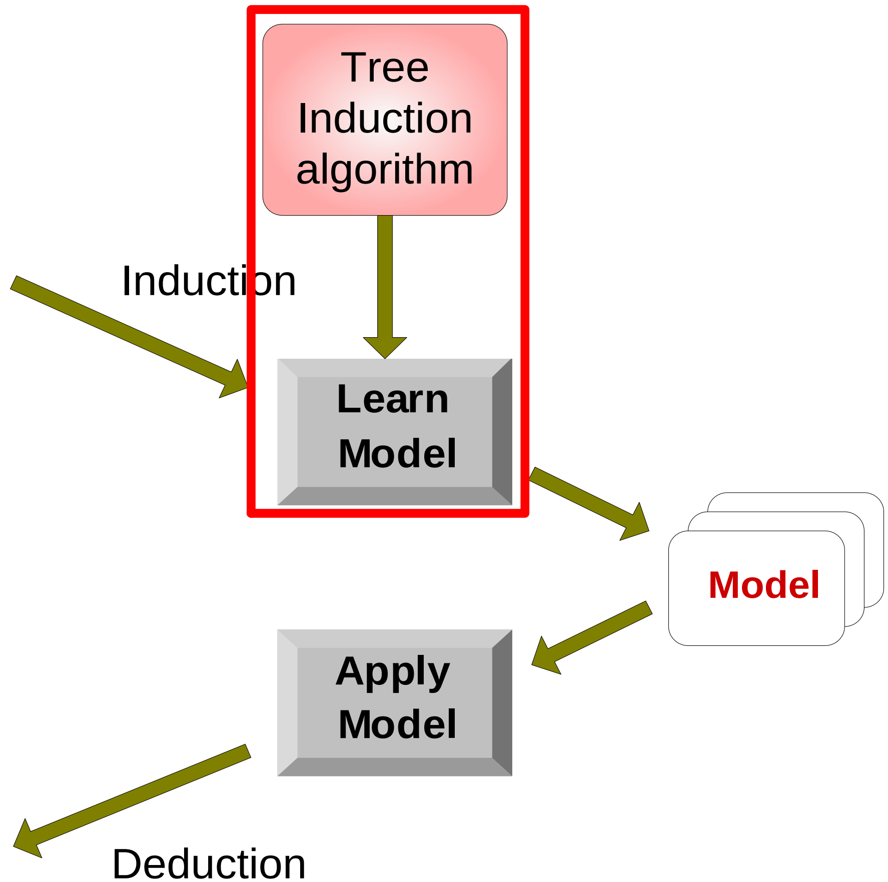
Aplicação do Modelo

- A maior vantagem das árvores é a facilidade de interpretar a tomada de decisão
- Vamos colocar a prova essa facilidade!

Classificação com Árvore 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

Indução de Árvores de Decisão

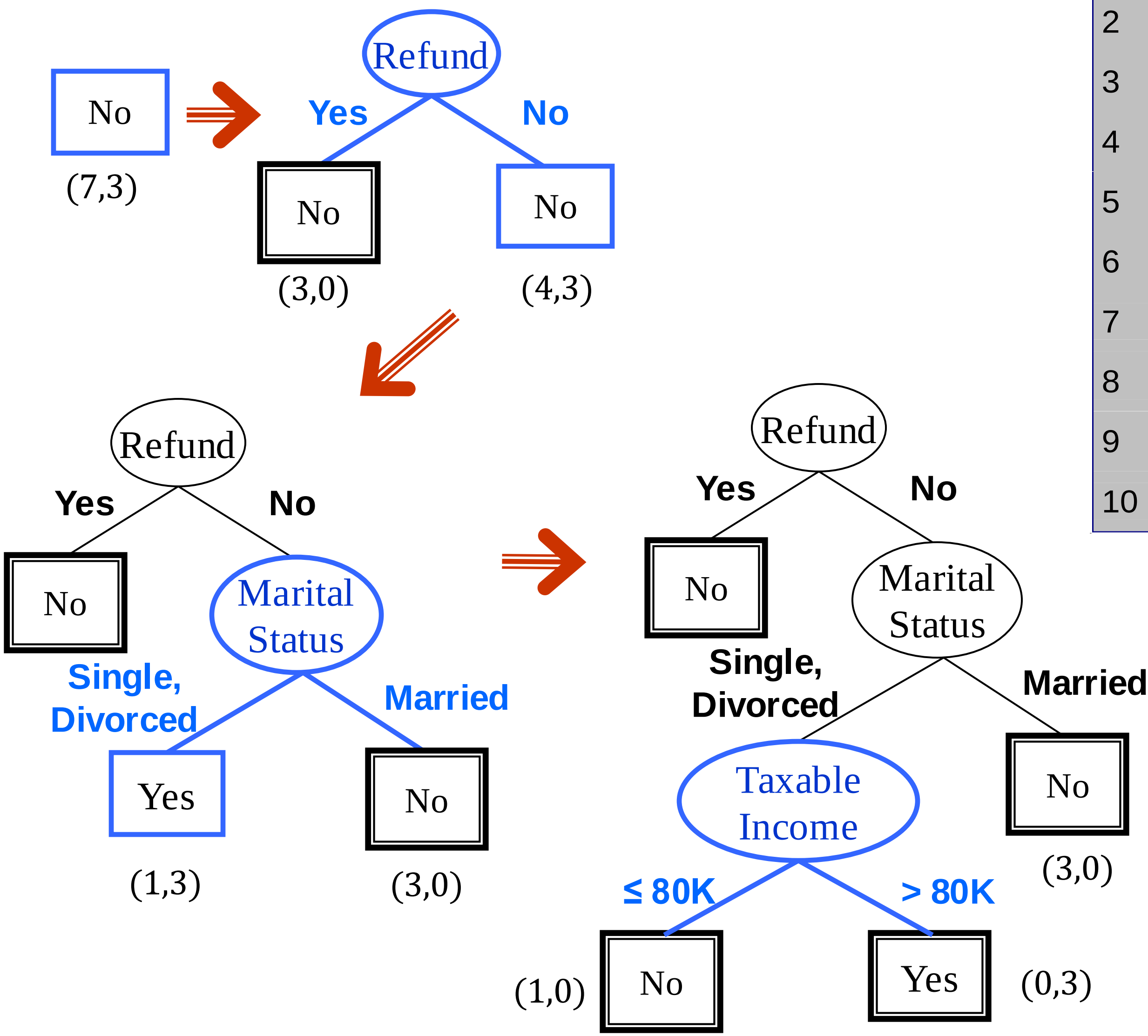
- Descobrir “árvore ótima” é problema NP-Difícil
- Muitas heurísticas para gerar árvores
  - Top-Down
  - Bottom-Up
  - Híbrida
  - Algoritmos Evolutivos
  - etc.

Indução Top-Down

- Algoritmo de Hunt
  - Assuma que D_t é o conjunto de instâncias de treino que chega ao nó t
  - Assuma que $y = \{y_1, \dots, y_c\}$ são os rótulos das classes
  - Passo 1:
    - Se todas instâncias em $\mathrm{D_t}$ pertencem a mesma classe $\mathrm{y_i}$, então t é um nó folha rotulado como $\mathrm{y_i}$
  - Passo 2:
    - Se $\mathrm{D_t}$ contém instâncias de mais de uma classe, um teste sobre determinado atributo é selecionado para particionar os registros em sub-conjuntos menores. Um nó é criado para cada resultado do teste e as instâncias em $\mathrm{D_t}$ são distribuídas por estes nós de acordo com os resultados. Aplicar algoritmo recursivamente para cada nó gerado

Algoritmo de Hunt

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Indução Top-Down

- Estratégia Recursiva
- Estratégia Gulosa (*greedy*)
  - Divide os registros com base em teste sobre atributo que otimiza localmente determinado critério

- Questões de Projeto
  - Determinar como particionar os dados
    - Como filtrar os dados com base em um atributo?
    - Como escolher o atributo a ser utilizado?
  - Determinar quando parar de particionar

Indução Top-Down

- Estratégia Recursiva
- Estratégia Gulosa (*greedy*)
  - Divide os registros com base em teste sobre atributo que <u>otimiza localmente</u> determinado critério

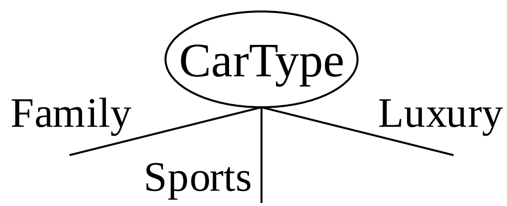
- Questões de Projeto
  - Determinar como particionar os dados
    - Como <u>filtrar os dados</u> com base em um atributo?
    - Como <u>escolher o atributo</u> a ser utilizado?
  - Determinar quando parar de particionar

Como filtrar os dados com base em um atributo?

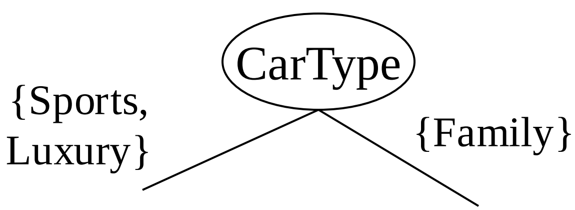
- Depende do tipo de atributo
  - Nominal
  - Ordinal
  - Contínuo
- Depende do número de divisões desejado
  - Binária
  - Múltipla

Divisão para atributos categóricos nominais

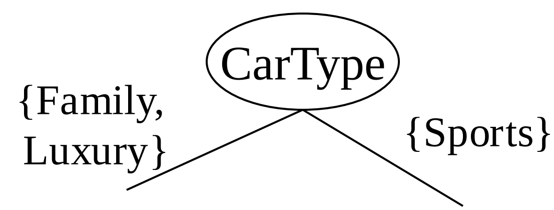
- Múltipla: dividir com base no número de categorias



- Binária: agregar categorias em dois sub-conjuntos. Necessário encontrar a divisão ótima.



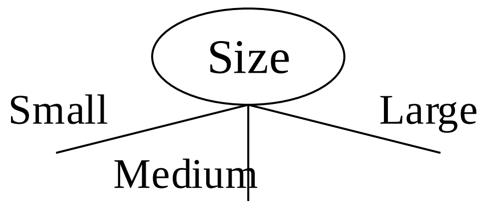
OU



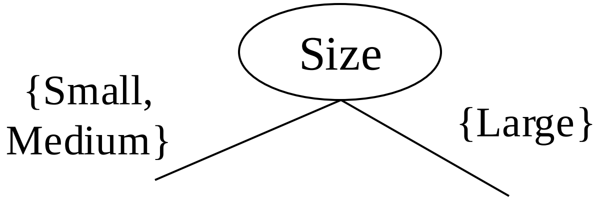
OU ....

Divisão para atributos categóricos ordinais

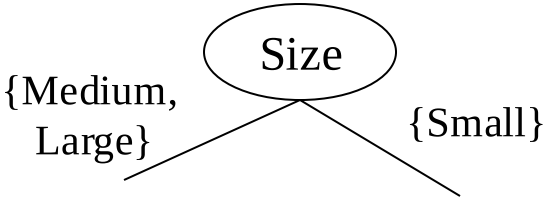
- Múltipla: dividir com base no número de categorias



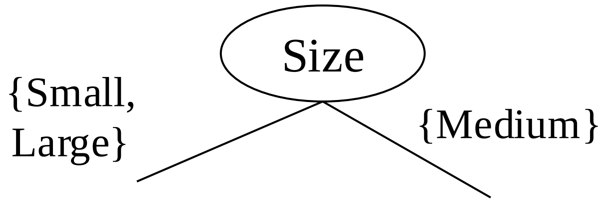
- Binária: agregar categorias em dois sub-conjuntos. Necessário encontrar a divisão ótima.



OU

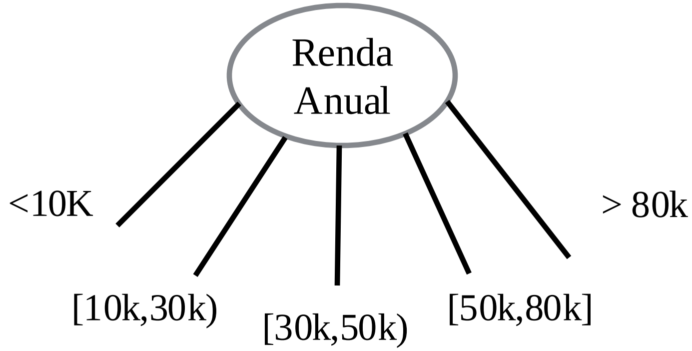


- E esta divisão?

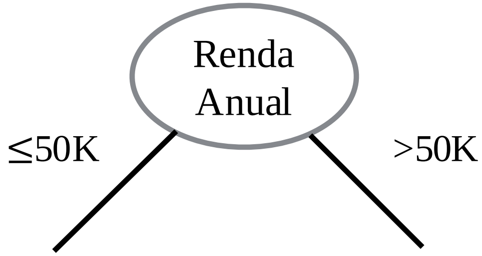


Divisão para atributos contínuos

- Múltipla: discretizar os valores em intervalos



- Binária: definir ponto de divisão



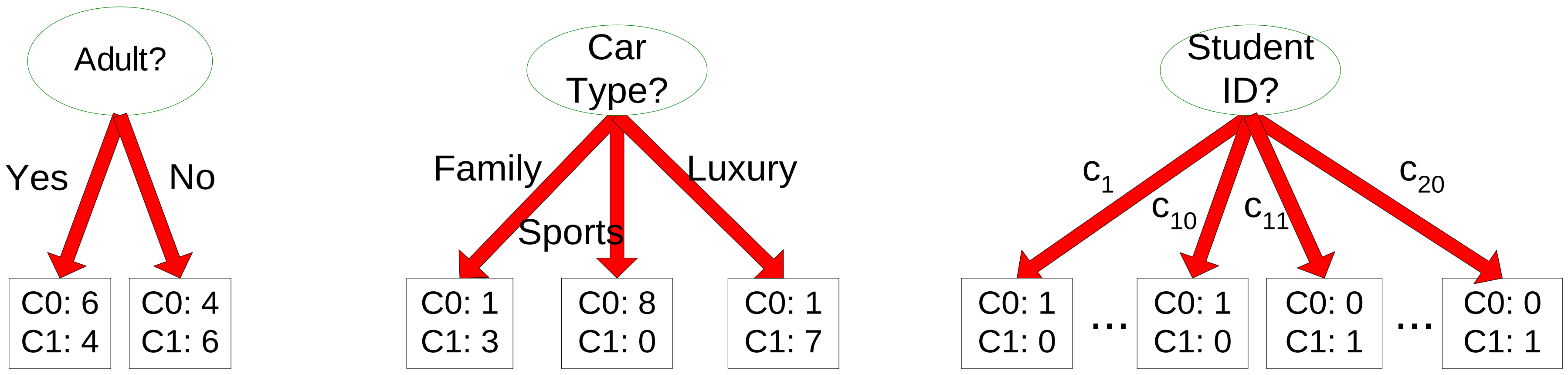
Indução Top-Down

- Estratégia Recursiva
- Estratégia Gulosa (*greedy*)
  - Divide os registros com base em teste sobre atributo que otimiza localmente determinado critério

- Questões de Projeto
  - Determinar como particionar os dados
    - Como filtrar os dados com base em um atributo?
    - Como escolher o atributo a ser utilizado?
  - Determinar quando parar de particionar

Como escolher o atributo?

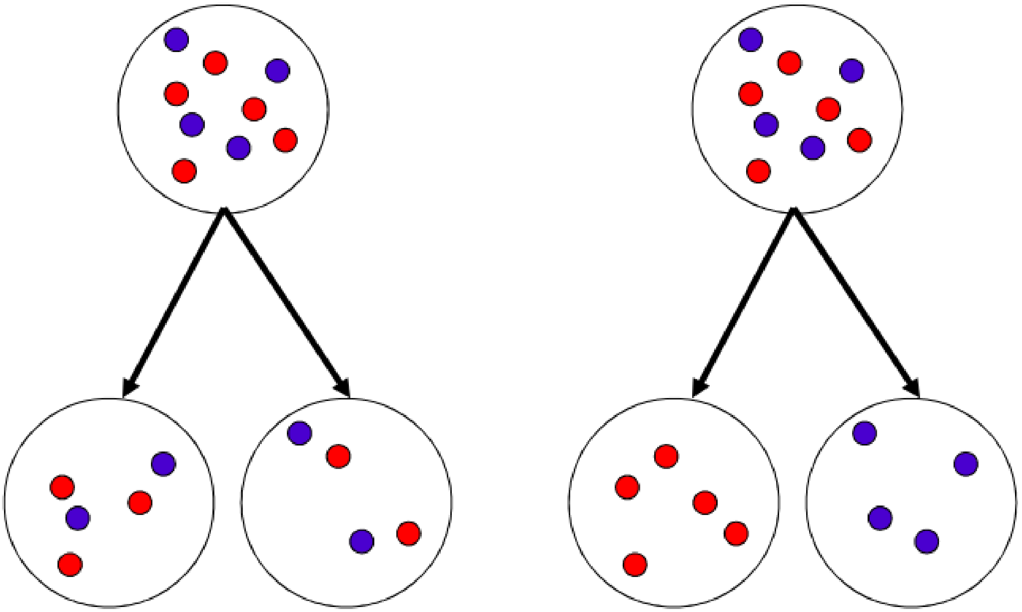
Antes da divisão: 10 instâncias da classe C_0
10 instâncias da classe C_1



Qual atributo é melhor para dividir os dados?

Como escolher o atributo?

- Estratégia gulosa
  - Dar preferência a nós com distribuição de classe homogênea (pura!)
  - Para tanto, precisamos de uma medida para quantificar impureza!



Medidas de Impureza de Nó

- Índice Gini
- Entropia
- Erro de Classificação

Medidas de Impureza de Nó

- **Índice Gini**
- Entropia
- Erro de Classificação

Índice Gini

- Índice Gini para um nó t:

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

$p(j|t)$ é a frequência relativa da classe j no nó t

- Valor máximo: $1 - \frac{1}{c}$ (quando classes forem equiprováveis)
- Valor mínimo: 0 (quando todas instâncias pertencem à mesma classe)

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	

Índice Gini

$$\boxed{GINI(t) = 1 - \sum_j [p(j \mid t)]^2}$$

C1	**0**
C2	**6**

$$p(C_1|t) = \frac{0}{6} = 0$$

$$p(C_2|t) = \frac{6}{6} = 1$$

$$Gini(t) = 1 - [p(C_1|t)^2 + p(C_2|t)^2] = 1 - [0^2 + 1^2] = 0$$

Índice Gini

$$GINI(t) = 1 - \sum_{j} [p(j \mid t)]^2$$

C1	**0**
C2	**6**

$$p(C_1|t) = \frac{0}{6} = 0 \qquad p(C_2|t) = \frac{6}{6} = 1$$

$$Gini(t) = 1 - [p(C_1|t)^2 + p(C_2|t)^2] = 1 - [0^2 + 1^2] = 0$$

C1	**1**
C2	**5**

$$p(C_1|t) = \frac{1}{6} \qquad p(C_2|t) = \frac{5}{6}$$

$$Gini(t) = 1 - [p(C_1|t)^2 + p(C_2|t)^2] = 1 - \left[\left(\frac{1}{6}\right)^2 + \left(\frac{5}{6}\right)^2\right] = 0.278$$

Índice Gini

$$GINI(t) = 1 - \sum_{j} [p(j \mid t)]^2$$

C1	**0**
C2	**6**

$p(C_1|t) = \frac{0}{6} = 0$ $\quad$ $p(C_2|t) = \frac{6}{6} = 1$

$$Gini(t) = 1 - [p(C_1|t)^2 + p(C_2|t)^2] = 1 - [0^2 + 1^2] = 0$$

C1	**1**
C2	**5**

$p(C_1|t) = \frac{1}{6}$ $\quad$ $p(C_2|t) = \frac{5}{6}$

$$Gini(t) = 1 - [p(C_1|t)^2 + p(C_2|t)^2] = 1 - \left[\left(\frac{1}{6}\right)^2 + \left(\frac{5}{6}\right)^2\right] = 0.278$$

C1	**2**
C2	**4**

$p(C_1|t) = \frac{2}{6}$ $\quad$ $p(C_2|t) = \frac{4}{6}$

$$Gini(t) = 1 - [p(C_1|t)^2 + p(C_2|t)^2] = 1 - \left[\left(\frac{1}{3}\right)^2 + \left(\frac{2}{3}\right)^2\right] = 0.444$$

Computando uma divisão com o Índice Gini

- Quando um nó p é dividido em k partições (filhos), a qualidade dessa divisão é dada por:

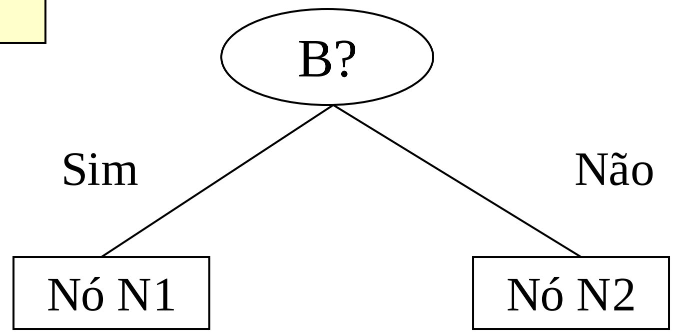
$$\boxed{GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)}$$

onde,

n_i = número de exemplos no filho i
n = número de exemplos no nó pai p

Computando Índice Gini para Atributos Binários

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



	Pai
C1	**6**
C2	**6**
Gini = 0.500	

Gini(N1) = 1 - [$(5/7)^2$ + $(2/7)^2$]
= 0.4082

	N1	N2
C1	**5**	**1**
C2	**2**	**4**

Gini(N2) = 1 - [$(1/5)^2$ + $(4/5)^2$]
= 0.32

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

Gini(divisão) = [(7/12) * 0.4082] + [(5/12) * 0.32]
= 0.37145

Computando Índice Gini para Atributos Categóricos

- Para cada categoria do atributo, faça as contagens por classe para descobrir as probabilidades de classe

Divisão Múltipla

	CarType		
	Family	**Sports**	**Luxury**
C1	1	2	1
C2	4	1	1
Gini	**0.393**		

Divisão Binária (encontre melhor divisão)

	CarType	
	{Sports, Luxury}	**{Family}**
C1	3	1
C2	2	4
Gini	**0.400**	

	CarType	
	{Sports}	**{Family, Luxury}**
C1	2	2
C2	1	5
Gini	**0.419**	

Computando Índice Gini para Atributos Contínuos

- Use decisões binárias baseada em limiar
  - Ordene os valores de forma ascendente
  - Percorrer linearmente os valores, atualizando cada vez a matriz de contagens e computando o índice Gini
  - Escolher o limiar que minimiza o Gini

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No
	Taxable Income									
Valores ordenados →	60	70	75	85	90	95	100	120	125	220

Limiares →	55		65		72		80		87		92		97		110		122		172		230	
	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
Yes	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0
No	0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0
Gini	0.420		0.400		0.375		0.343		0.417		0.400		***0.300***		0.343		0.375		0.400		0.420	

Medidas de Impureza de Nó

- Índice Gini
- **Entropia**
- Erro de Classificação

Entropia

- Entropia de um nó t:

$$Entropy(t) = -\sum_{j} p(j\,|\,t)\log_2 p(j\,|\,t)$$

$p(j|t)$ é a frequência relativa da classe j no nó t

- Valor máximo: $\log_2 c$ (quando classes forem equiprováveis)

- Valor mínimo: 0 (quando todas instâncias pertencem à mesma classe)

$p(j\|t)$	$log(p(j\|t))$	$p(j\|t) * log(p(j\|t))$
0	valor indefinido	assumimos = 0
0.1	-3.32	-0.33
0.2	-2.32	-0.46
0.3	-1.74	-0.52
0.4	-1.32	-0.53
0.5	-1.00	-0.50
0.6	-0.74	-0.44
0.7	-0.51	-0.36
0.8	-0.32	-0.26
0.9	-0.15	-0.14
1	0.00	0.00

Entropia

$$Entropy(t) = -\sum_j p(j|t)\log_2 p(j|t)$$

C1	**0**
C2	**6**

$$p(C_1|t) = \frac{0}{6} = 0 \qquad p(C_2|t) = \frac{6}{6} = 1$$

$$E(t) = -[0\log_2 0 + 1\log_2 1] = -[0+0] = 0$$

C1	**1**
C2	**5**

$$p(C_1|t) = \frac{1}{6} \qquad p(C_2|t) = \frac{5}{6}$$

$$E(t) = -\left[\frac{1}{6}\log_2\frac{1}{6} + \frac{5}{6}\log_2\frac{5}{6}\right] = -[-0.65] = 0.65$$

C1	**2**
C2	**4**

$$p(C_1|t) = \frac{2}{6} \qquad p(C_2|t) = \frac{4}{6}$$

$$E(t) = -\left[\frac{2}{6}\log_2\frac{2}{6} + \frac{4}{6}\log_2\frac{4}{6}\right] = -[-0.92] = 0.92$$

Computando uma divisão com a Entropia

- Ganho de Informação:

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

Nó pai, p, é dividido em k partições

n_i é o número de instâncias na partição i

  - Mede a redução em entropia devido à divisão. Procura-se minimizar a média ponderada das entropias dos nós filhos (equivalente a maximizar o ganho de informação)
  - Utilizado nos algoritmos ID3 e C4.5 de J.R. Quinlan
  - Desvantagem: assim como o índice Gini, é tendencioso àquelas divisões com número grande de partições, cada uma sendo pequena porém pura

Alternativa ao Ganho de Informação

- Gain Ratio:

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO} \qquad SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

Nó pai, p, é dividido em k partições

n_i é o número de instâncias na partição i

- Ajusta o Ganho de Informação pela entropia da distribuição do particionamento (SplitINFO). Quanto maior a entropia do particionamento (número alto de partições pequenas), maior a penalidade ao Ganho de Informação!

- Utilizado no algoritmo C4.5

Medidas de Impureza de Nó

- Índice Gini
- Entropia
- **Erro de Classificação**

Erro de Classificação

- Erro de classificação do nó *t* :

$$\boxed{Erro(t) = 1 - \max_{i} P(i \mid t)}$$

- Mede o erro de classificação feito em um nó
- Valor máximo: $1 - \frac{1}{c}$ (quando classes forem equiprováveis)
- Valor mínimo: 0 (quando todas instâncias pertencem à mesma classe)

Erro de Classificação

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

C1	**0**
C2	**6**

$p(C_1|t) = \frac{0}{6} = 0$ $\quad p(C_2|t) = \frac{6}{6} = 1$

$Erro(t) = 1 - \max(0,1) = 1 - 1 = 0$

C1	**1**
C2	**5**

$p(C_1|t) = \frac{1}{6}$ $\quad p(C_2|t) = \frac{5}{6}$

$Erro(t) = 1 - \max\left(\frac{1}{6}, \frac{5}{6}\right) = 1 - \frac{5}{6} = \frac{1}{6}$

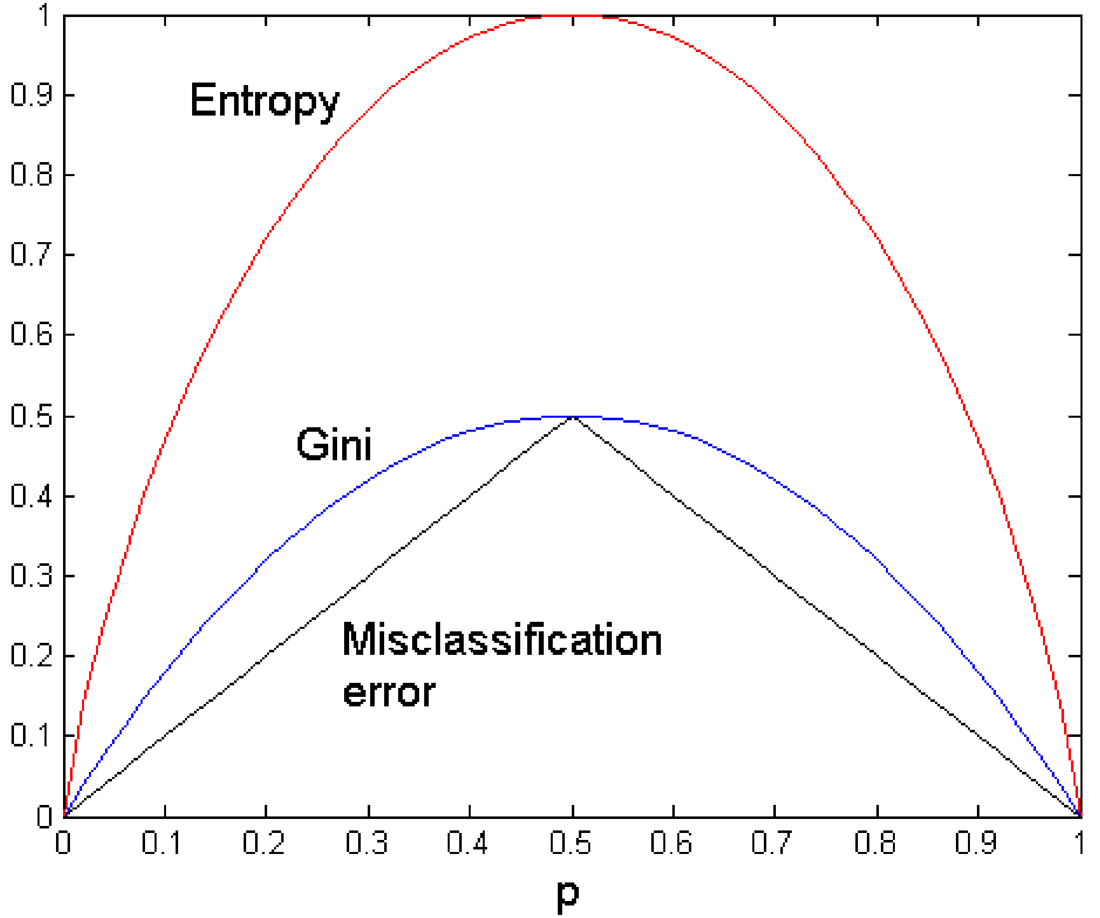
C1	**2**
C2	**4**

$p(C_1|t) = \frac{2}{6}$ $\quad p(C_2|t) = \frac{4}{6}$

$Erro(t) = 1 - \max\left(\frac{2}{6}, \frac{4}{6}\right) = 1 - \frac{4}{6} = \frac{1}{3}$

Comparação entre os critérios de divisão

Para um problema de 2 classes:



Genericamente...

$$\Delta = I(v_{pai}) - \sum_{t=1}^{k} \frac{N(v_t)}{N} I(v_t)$$

Média ponderada

onde:

$I(v)$: mede o grau de impureza do nó v

$N(v_t)$: número de objetos no filho v_t

N: número de objetos no nó pai v_{pai}

Indução Top-Down

- Estratégia Recursiva
- Estratégia Gulosa (*greedy*)
  - Divide os registros com base em teste sobre atributo que otimiza localmente determinado critério

- Questões de Projeto
  - Determinar como particionar os dados
    - Como filtrar os dados com base em um atributo?
    - Como escolher o atributo a ser utilizado?
  - Determinar quando parar de particionar

Critérios de Parada para Indução Top-Down

- Parar de expandir nós quando:
  - Todas instâncias forem da mesma classe (homogeneidade de classe)
  - Todos valores de atributos forem iguais (homogeneidade de instâncias)
  - Atingir valor satisfatório do critério de divisão (parâmetro)
  - Atingir profundidade máxima (parâmetro)
  - ...

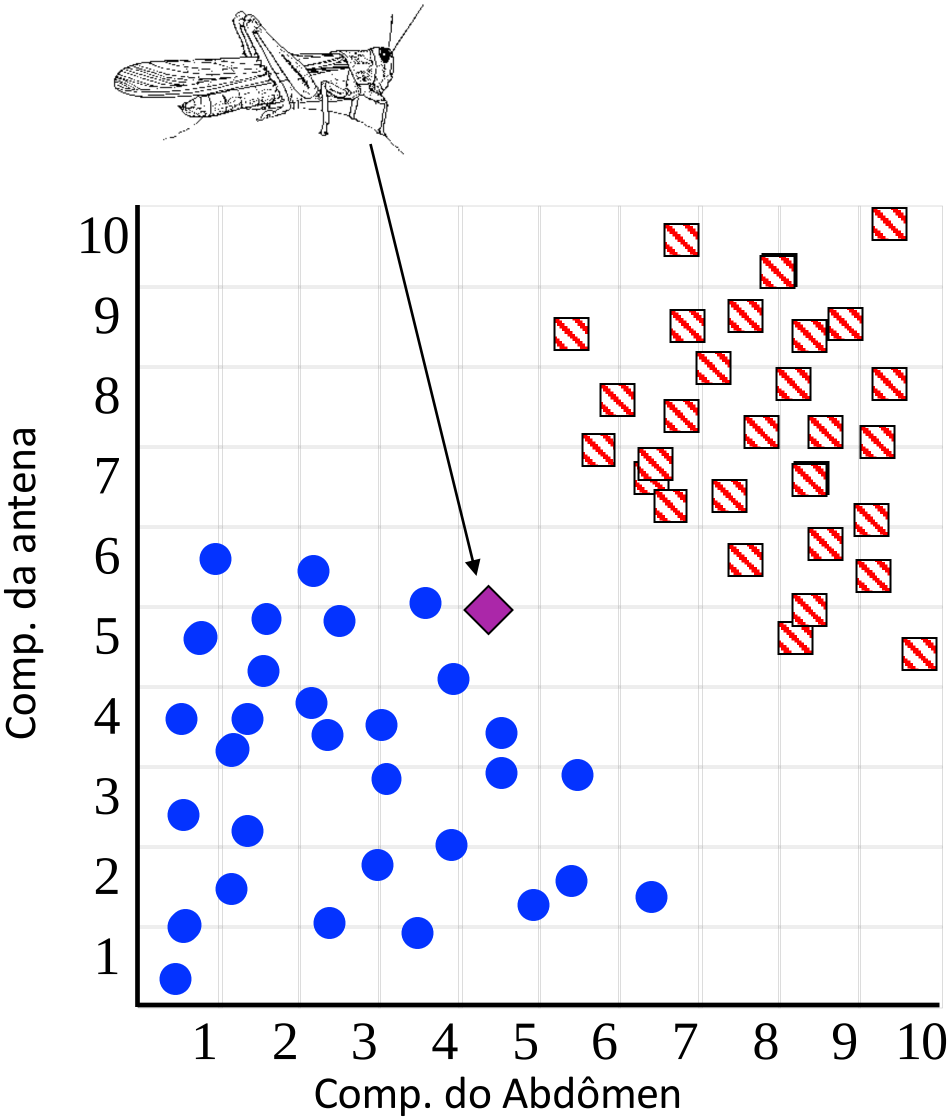
Questões

- Árvores de decisão não possuem bias de restrição (isto é, são capazes de representar qualquer função de classificação de dados). Desta forma, responda:
  - Qual o limite inferior (l*ower bound*) de taxa de erro que árvores construídas a partir do critério de homogeneidade de classes são capazes de atingir nos dados de treinamento?
  - Isso significa que árvores de decisão são mais sujeitas a underfitting ou overfitting?

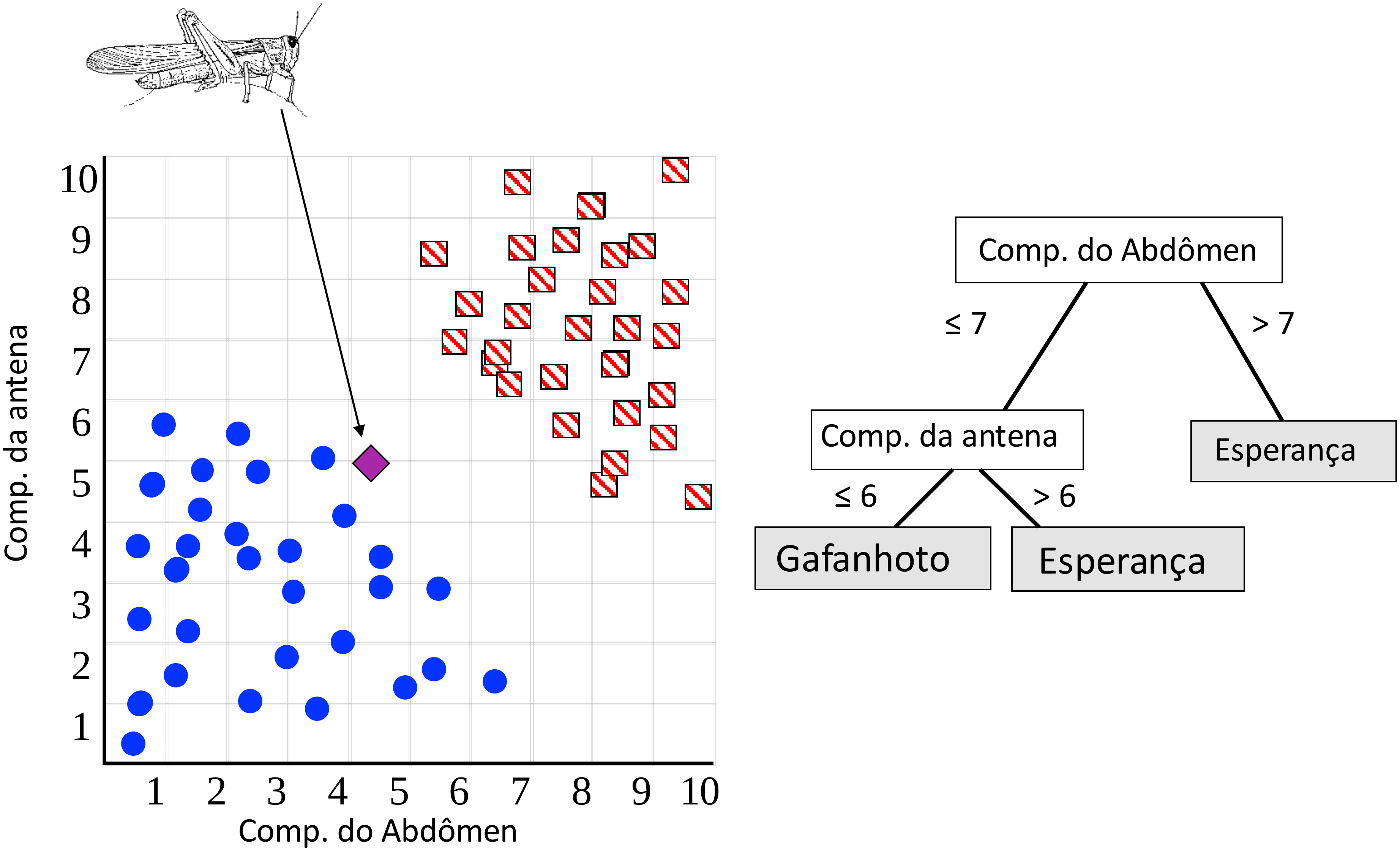
Vantagens e Desvantagens de Árvores de Decisão

- Vantagens:
  - Fácil de compreender (muito utilizadas por médicos!)
  - Possível gerar regras com base nas árvores
  - Custo baixo de geração do modelo
  - Extremamente rápida para classificar novas instâncias
- Desvantagens:
  - Podem tornar-se muito grandes
  - Sujeitas a overfitting (super-ajuste aos dados)
  - Geram apenas hiperplanos paralelos aos eixos
    - Logo, não lidam bem com atributos correlacionados (por quê?)
  - Solução localmente ótima pode estar longe do ótimo global

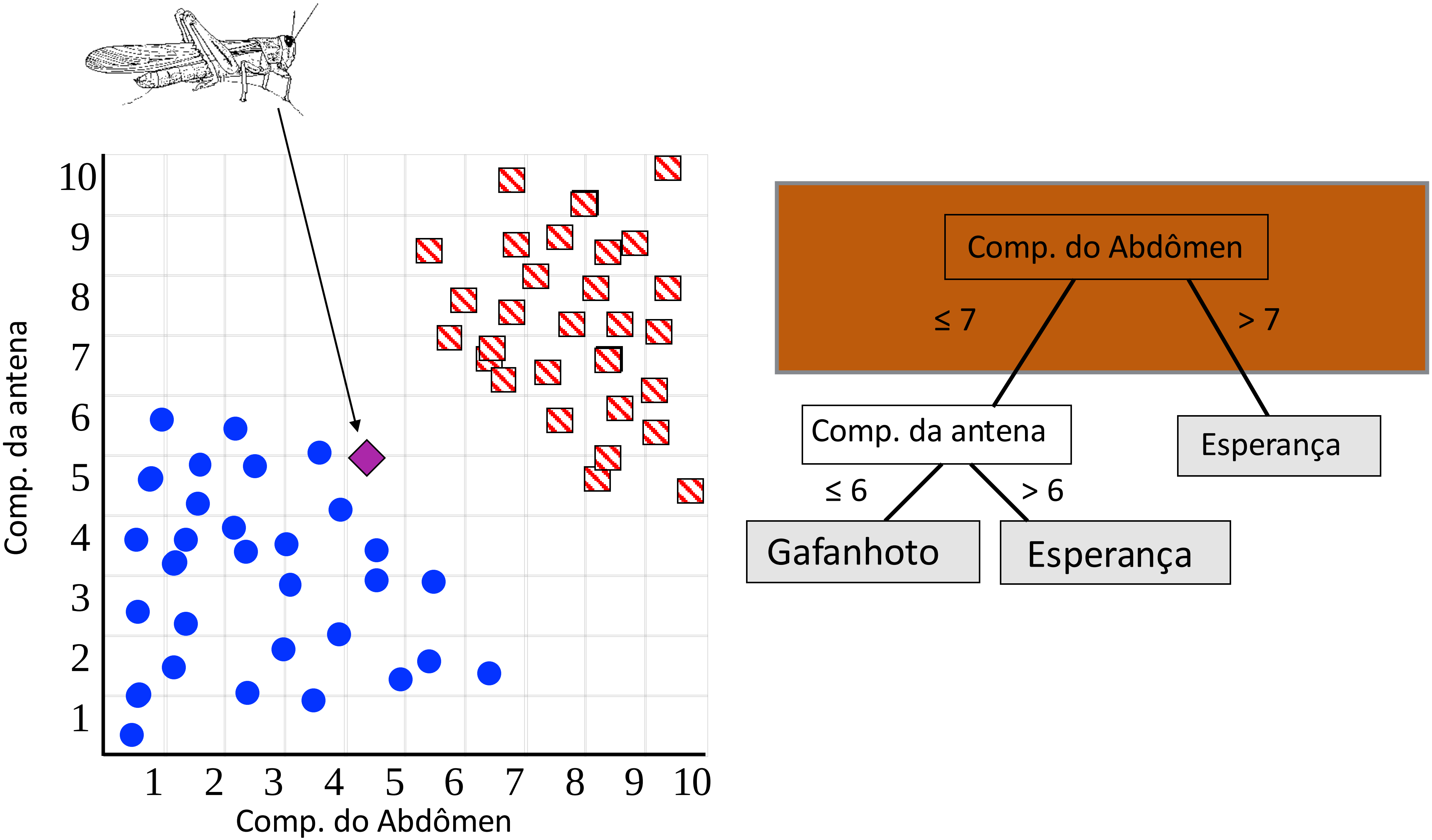
Visualização Geométrica de uma Árvore de Decisão



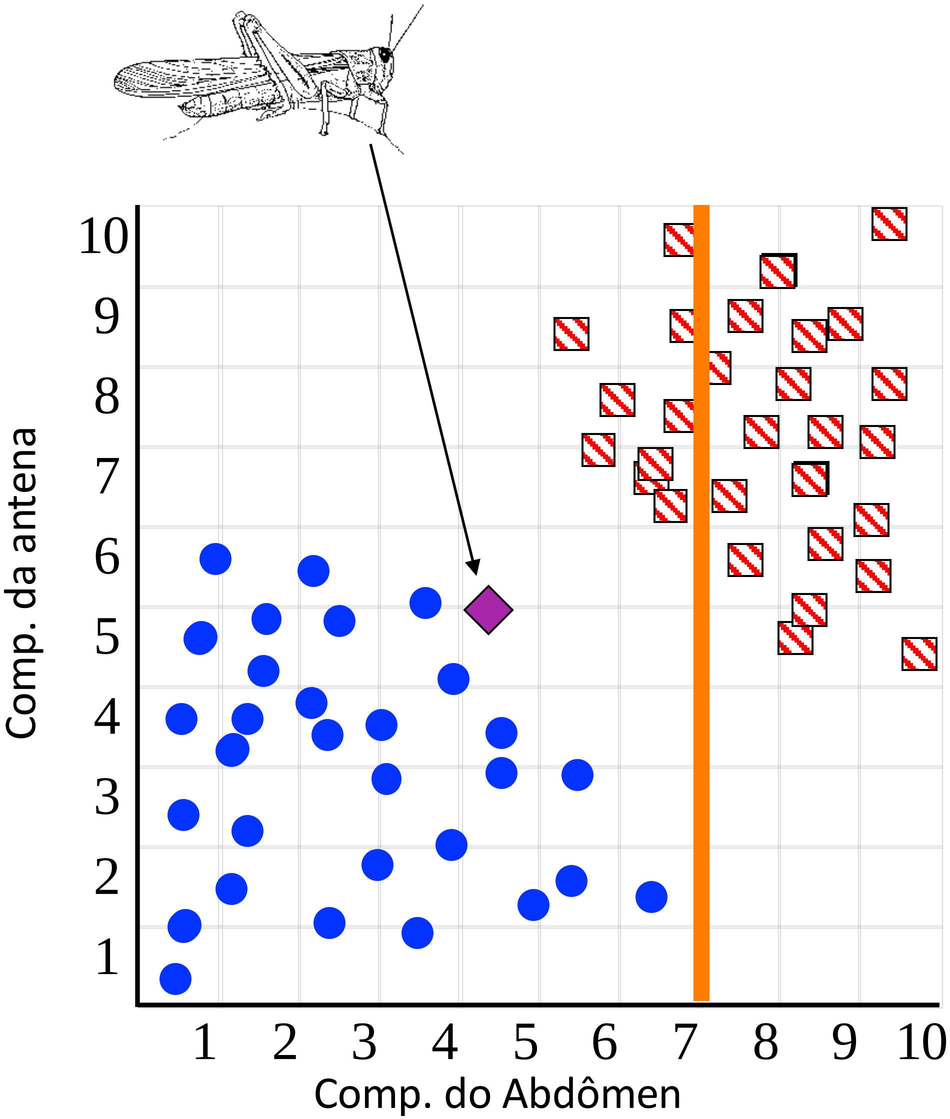
Visualização Geométrica de uma Árvore de Decisão

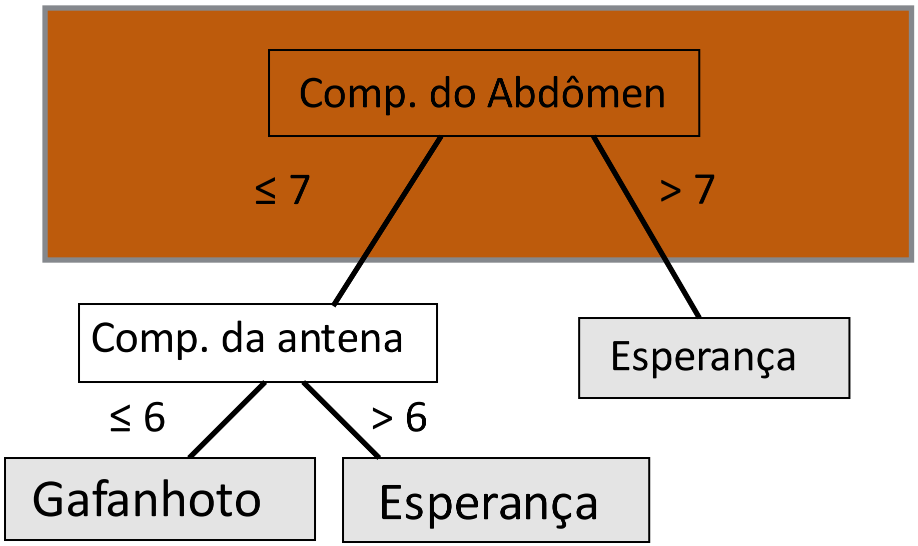


Visualização Geométrica de uma Árvore de Decisão

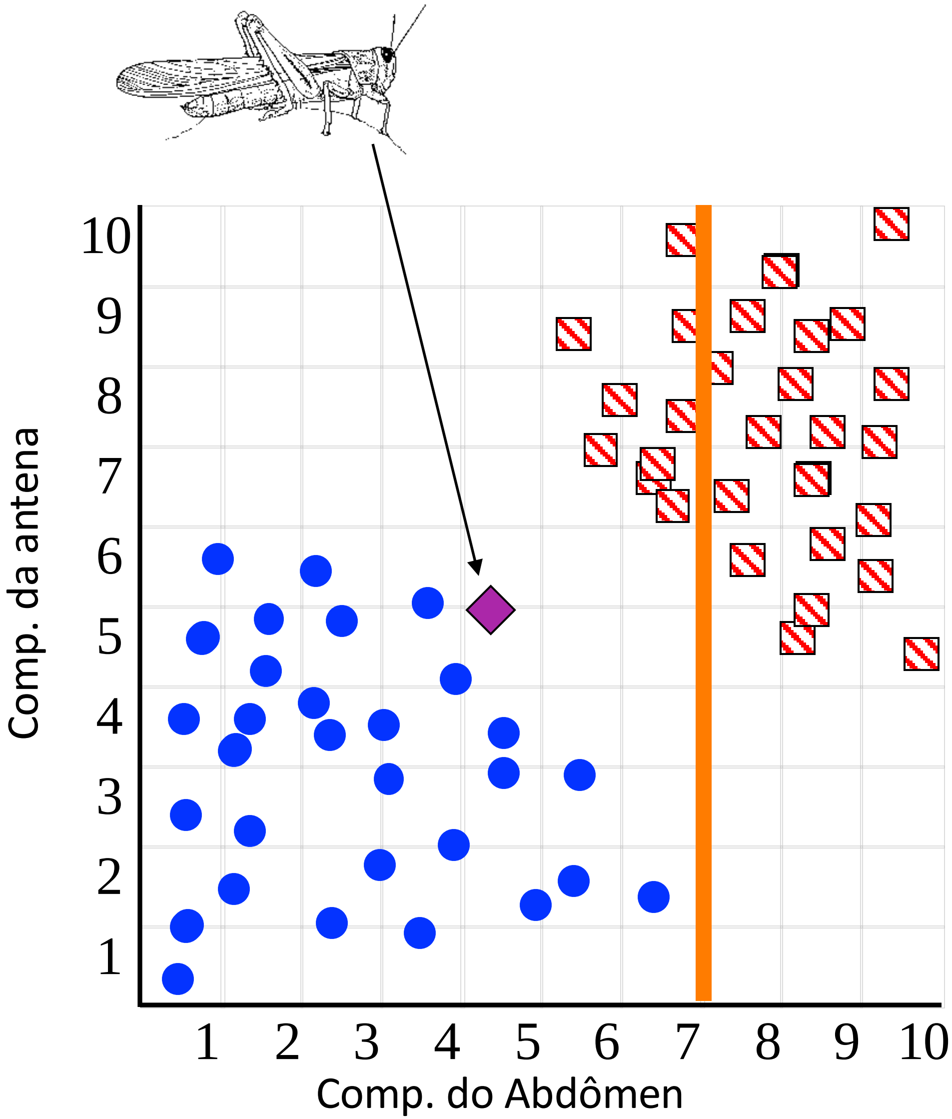


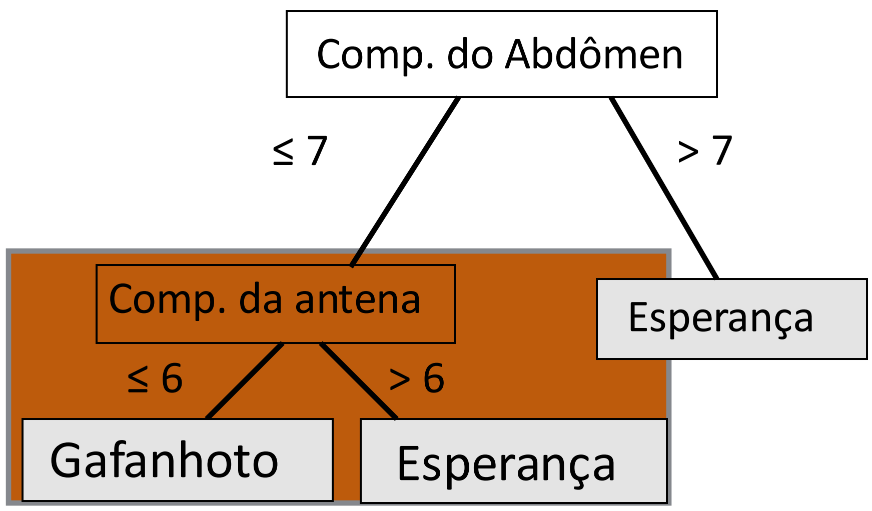
Visualização Geométrica de uma Árvore de Decisão



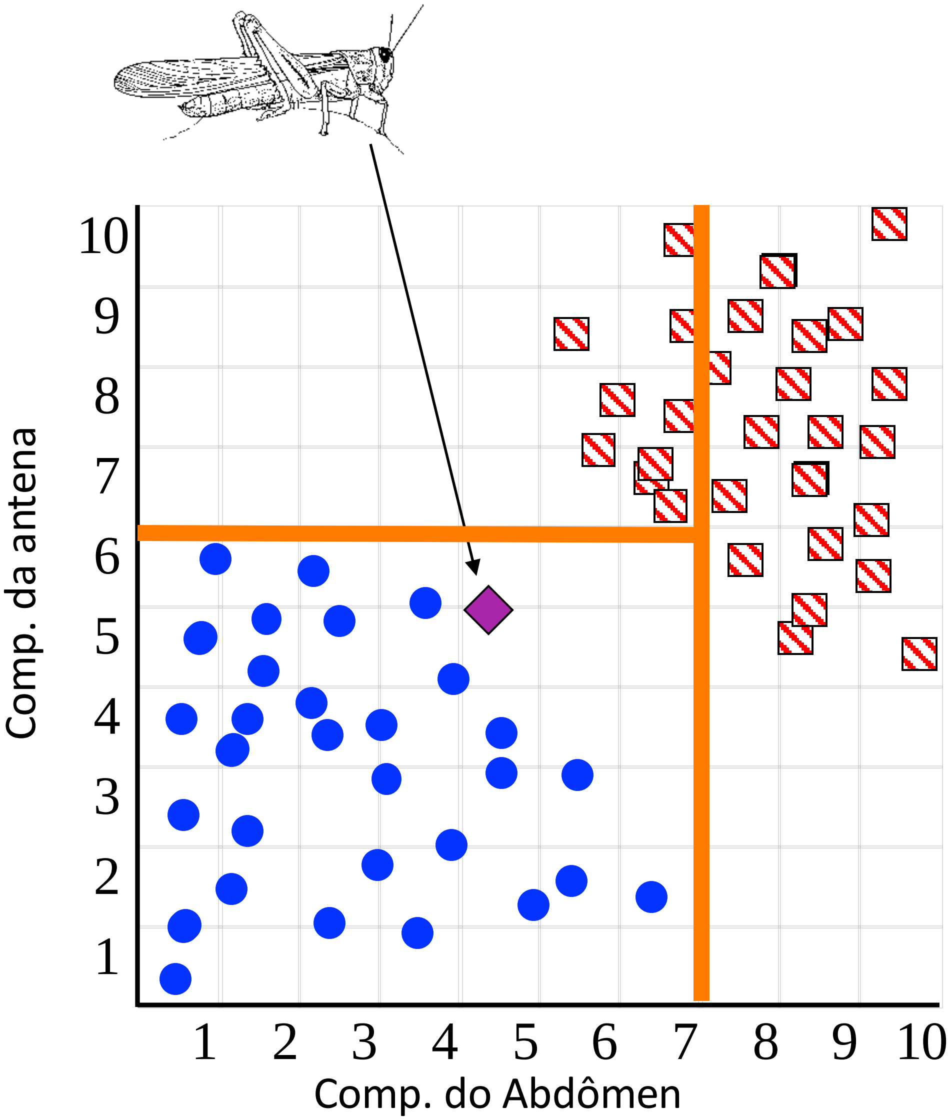


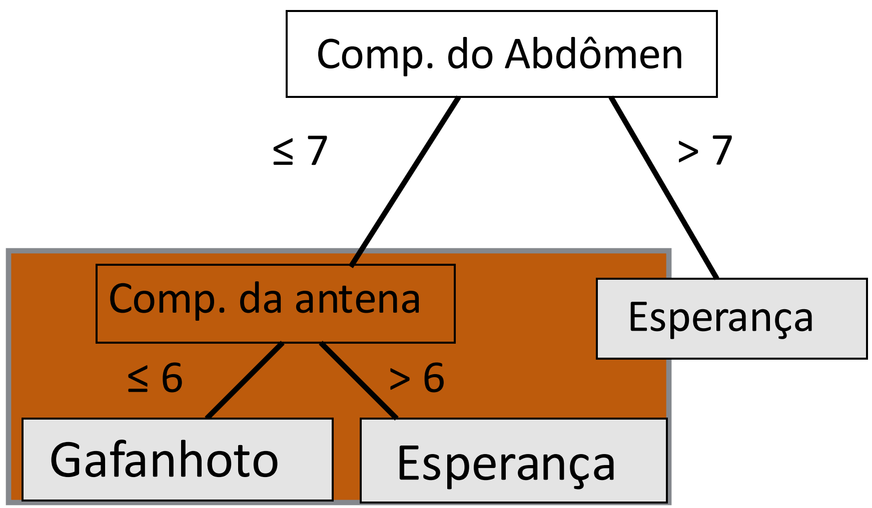
Visualização Geométrica de uma Árvore de Decisão





Visualização Geométrica de uma Árvore de Decisão



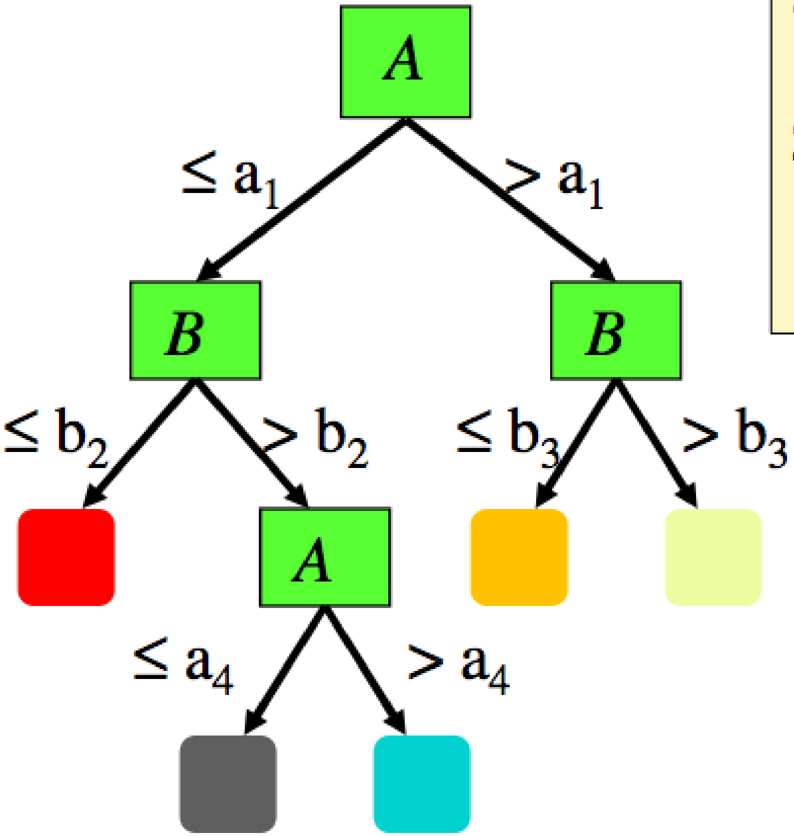


Espaço de Hipóteses

- Cada percurso da raiz até o nó folha representa uma **regra de classificação**

- Cada nó folha
  - Está associado a uma classe
  - Corresponde a uma região do domínio dos atributos
    - Hiper-retângulo
    - Intersecção de hiper-retângulos é vazia
    - União é o espaço total

De árvores para regras



Regras: disjunções de conjunções lógicas

1. **Se** A $\leq a_1$ **E** B $\leq b_2$ **Então** Classe = Vermelha

OU

2. **Se** A $> a_1$ **E** B $\leq b_3$ **Então** Classe = Laranja

OU

...

Exercício: complete as regras !

Busca no Espaço de Hipóteses

- Não há backtracking
  - Impureza é minimizada localmente em cada nó!
    - Suposição: soma dos ótimos locais aproxima bem o ótimo global

- Espaço de hipóteses completo
  - A função objetivo certamente está contida nele
  - Sem bias de restrição
    - Proporcionando chances de *overfitting*
  - Com bias de busca (preferência)
    - Árvores onde atributos que geram maior redução de impureza estão nos níveis superiores
    - Tal bias implica em tendência para árvores mais curtas

Alternativas às Desvantagens

- *Como solucionar overfitting?*
  - Navalha de Occam
  - Poda!

"Se em tudo o mais forem idênticas as várias explicações de um fenômeno, a mais simples é a melhor."

"Entia non sunt multiplicanda praeter necessitatem "

- Pré-poda
  - Interromper crescimento da árvore segundo algum critério (valor de medida de impureza, número mínimo de instâncias atingido, etc.)
- Pós-poda
  - Crescer a árvore até a homogeneidade de classes
  - Cortar os nós de maneira bottom-up
  - Se erro de generalização melhorar após corte, trocar sub-árvore por nó folha

Exemplo 1 de Pós-Poda: Erro Pessimista

- Erro de treinamento é tendencioso, e portanto não pode ser utilizado como medida confiável para avaliar se nós podem ser podados
- Solução: ajustar o erro de treinamento de forma a penalizar pela criação de novos nós

$$\overline{e}(T) = \frac{\sum_{t_i \in T} e(t_i)}{\sum_{t_i \in T} n(t_i)} = \frac{e(T)}{N} \Rightarrow \overline{e}'(T) = \frac{\sum_{t_i \in T} \left[e(t_i) + \Omega(t_i) \right]}{\sum_{t_i \in T} n(t_i)} = \frac{e(T) + \Omega(T)}{N}$$

Valor típico de $\Omega(t_i) = 0.5$

Exemplo 1 de Pós-Poda: Erro Pessimista

Classe = Sim	20
Classe = Não	10
Erro = 10/30	

Erro de treino (pai) = 10/30

Erro pessimista (pai) = (10 + 0.5)/30 = 10.5/30

Exemplo 1 de Pós-Poda: Erro Pessimista

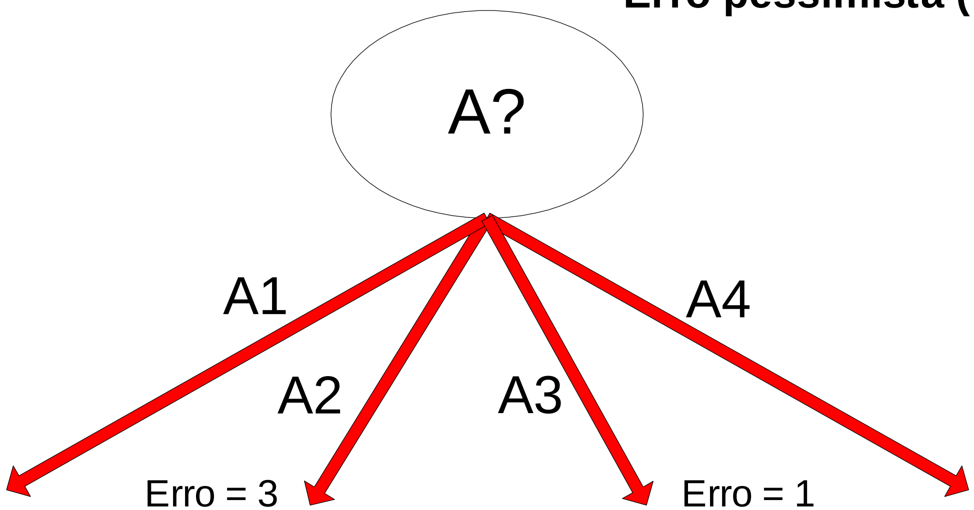
Classe = Sim	20
Classe = Não	10
Erro = 10/30	

Erro de treino (pai) = 10/30

Erro pessimista (pai) = (10 + 0.5)/30 = 10.5/30

Erro de treino (filhos) = 9/30

Erro pessimista (filhos) = (9 + 4 *0.5)/30 = 11/30



Erro = 4

Classe = Sim	8
Classe = Não	4

Erro = 3

Classe = Sim	3
Classe = Não	4

Erro = 1

Classe = Sim	4
Classe = Não	1

Erro = 1

Classe = Sim	5
Classe = Não	1

Exemplo 1 de Pós-Poda: Erro Pessimista

Classe = Sim	20
Classe = Não	10
Erro = 10/30	

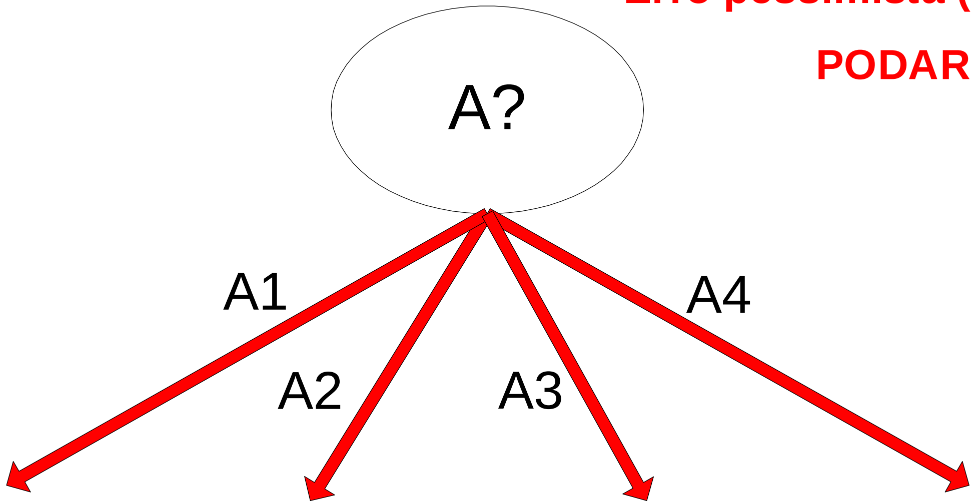
Erro de treino (pai) = 10/30

Erro pessimista (pai) = (10 + 0.5)/30 = 10.5/30

Erro de treino (filhos) = 9/30

Erro pessimista (filhos) = (9 + 4 *0.5)/30 = 11/30

PODAR!



Classe = Sim	8
Classe = Não	4

Classe = Sim	3
Classe = Não	4

Classe = Sim	4
Classe = Não	1

Classe = Sim	5
Classe = Não	1

Exemplo 2 de Pós-Poda: Reduced-Error Pruning

- Separar uma parte dos dados de treino para conjunto de validação
  - Não utilizado para treinamento

- Avaliar de maneira bottom-up se trocar uma sub-árvore por nó folha reduz o erro no conjunto de validação

- Vantagem: complexidade linear

- Desvantagem: reduz o conjunto de treino

Exemplo 3 de Pós-Poda: Cost-Complexity Pruning

- Define um custo associado ao tamanho da árvore
- Logo, considera o erro e o tamanho para estimar a árvore ideal

$$C_a(T) = \sum_{m=1}^{|T|} N_m E_m(T) + a|T|$$

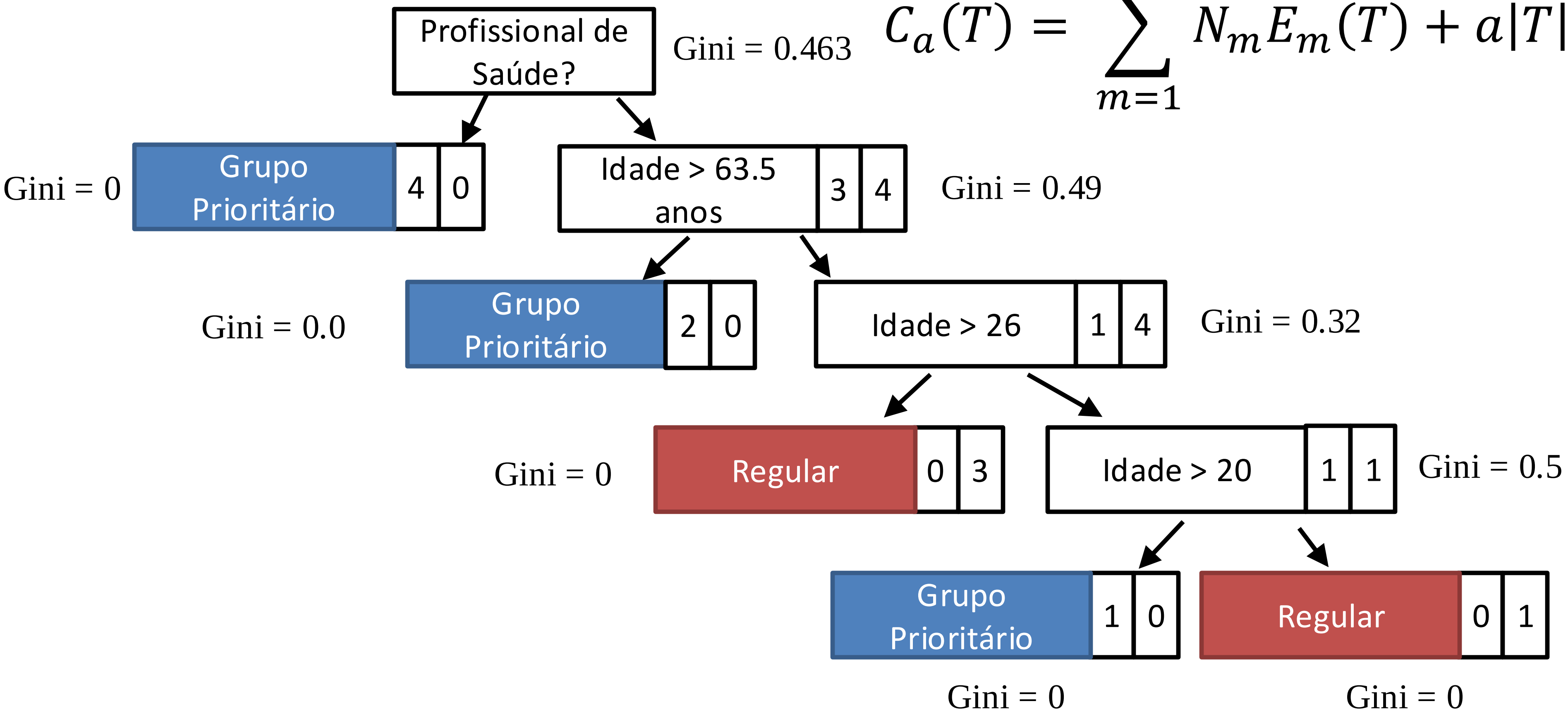
Exemplo 3 de Pós-Poda: Cost-Complexity Pruning

$$C_a(T) = \sum_{m=1}^{|T|} N_m E_m(T) + a|T|$$

- N_m: Quantidade de itens que chegaram ao nó m
- $E_m(T)$: Medida de erro para um nó m
- a: Termo penalizador
- $|T|$: Quantidade de nós folha na árvore

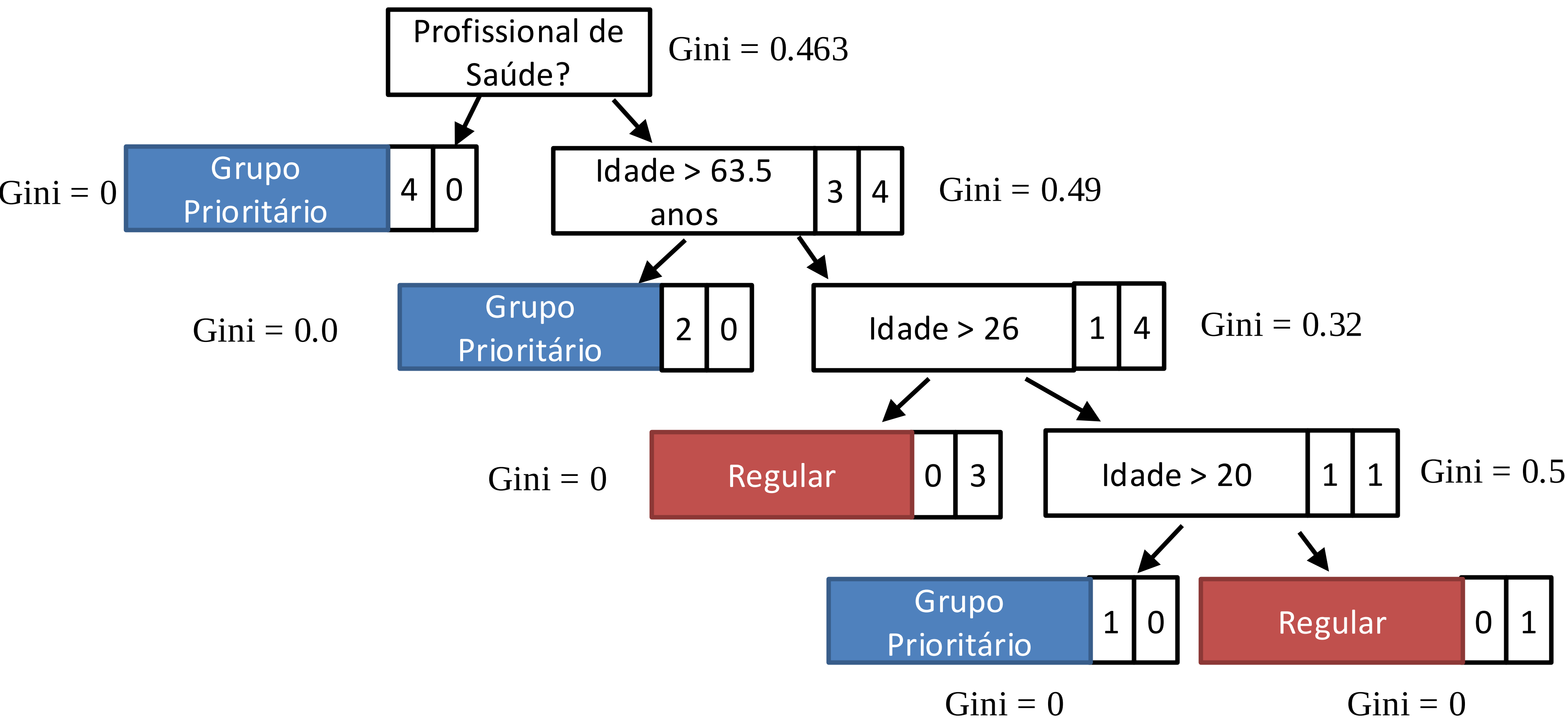
Exemplo 3 de Pós-Poda: Cost-Complexity Pruning

$$C_a(T) = \sum_{m=1}^{|T|} N_m E_m(T) + a|T|$$



$$C_a(T) = \sum_{m=1}^{|T|} N_m E_m(T) + a|T|$$

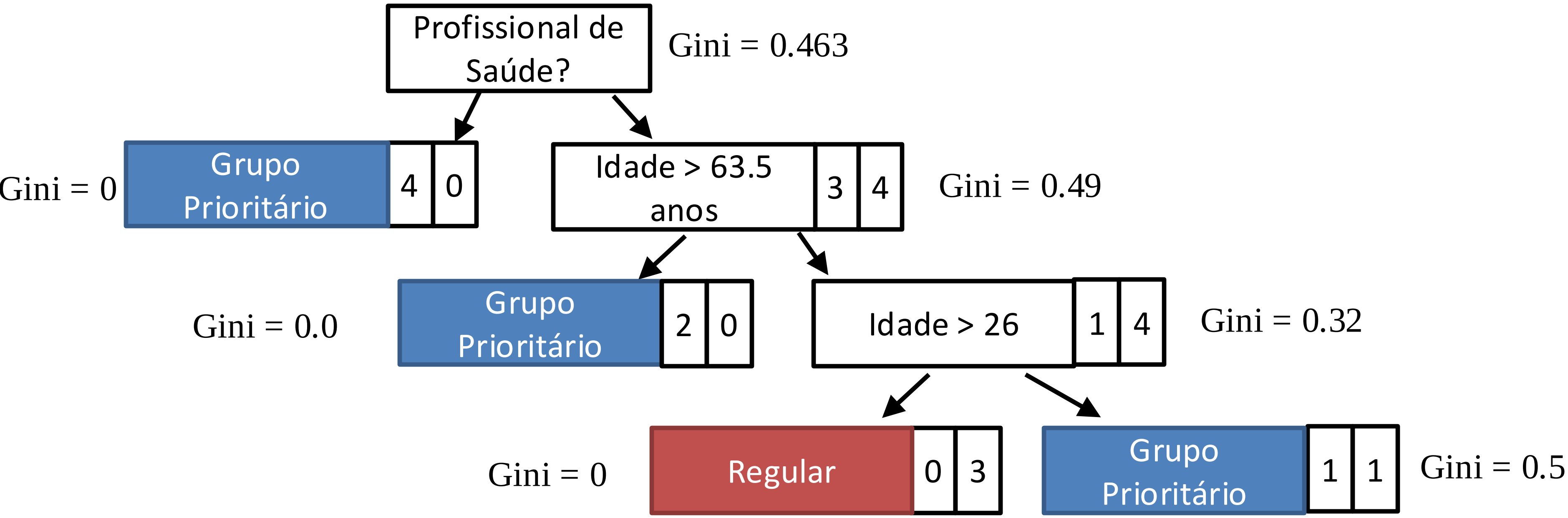
$$C_a(T_0) = \frac{4}{11} \times 0 + \frac{2}{11} \times 0 + \frac{3}{11} \times 0 + \frac{1}{11} \times 0 + \frac{1}{11} \times 0 + a \times 5$$



$$C_a(T) = \sum_{m=1}^{|T|} N_m E_m(T) + a|T|$$

$$C_a(T_0) = \frac{4}{11} \times 0 + \frac{2}{11} \times 0 + \frac{3}{11} \times 0 + \frac{1}{11} \times 0 + \frac{1}{11} \times 0 + a \times 5$$

$$C_a(T_1) = \frac{4}{11} \times 0 + \frac{2}{11} \times 0 + \frac{3}{11} \times 0 + \frac{2}{11} \times 0.5 + a \times 4$$

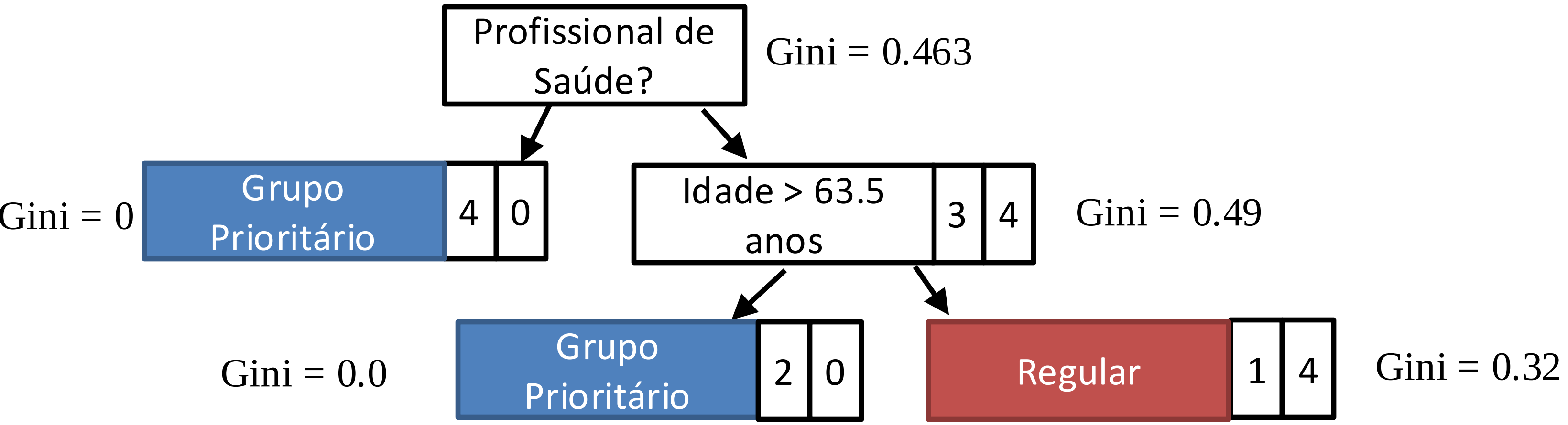


$$C_a(T) = \sum_{m=1}^{|T|} N_m E_m(T) + a|T|$$

$$C_a(T_0) = \frac{4}{11} \times 0 + \frac{2}{11} \times 0 + \frac{3}{11} \times 0 + \frac{1}{11} \times 0 + \frac{1}{11} \times 0 + a \times 5$$

$$C_a(T_1) = \frac{4}{11} \times 0 + \frac{2}{11} \times 0 + \frac{3}{11} \times 0 + \frac{2}{11} \times 0.5 + a \times 4$$

$$C_a(T_2) = \frac{4}{11} \times 0 + \frac{2}{11} \times 0 + \frac{5}{11} \times 0.32 + a \times 3$$



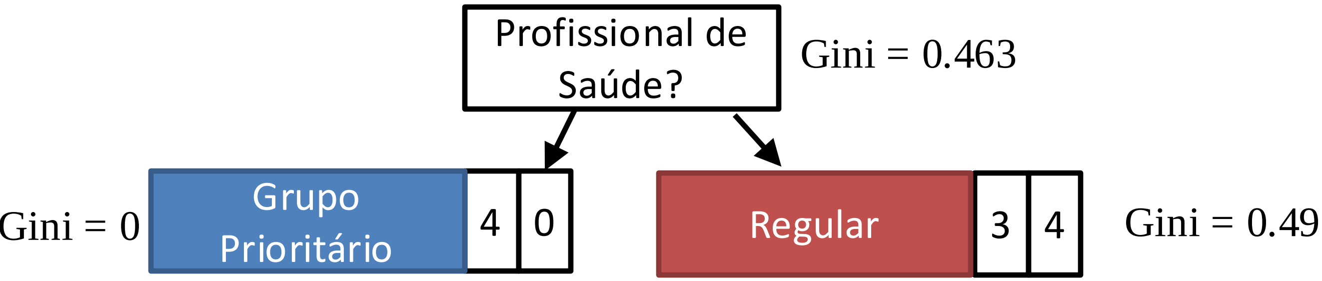
$$C_a(T) = \sum_{m=1}^{|T|} N_m E_m(T) + a|T|$$

$$C_a(T_0) = \frac{4}{11} \times 0 + \frac{2}{11} \times 0 + \frac{3}{11} \times 0 + \frac{1}{11} \times 0 + \frac{1}{11} \times 0 + a \times 5$$

$$C_a(T_1) = \frac{4}{11} \times 0 + \frac{2}{11} \times 0 + \frac{3}{11} \times 0 + \frac{2}{11} \times 0.5 + a \times 4$$

$$C_a(T_2) = \frac{4}{11} \times 0 + \frac{2}{11} \times 0 + \frac{5}{11} \times 0.32 + a \times 3$$

$$C_a(T_3) = \frac{4}{11} \times 0 + \frac{7}{11} \times 0.49 + a \times 2$$



$$C_a(T) = \sum_{m=1}^{|T|} N_m E_m(T) + a|T|$$

$$C_a(T_0) = \frac{4}{11} \times 0 + \frac{2}{11} \times 0 + \frac{3}{11} \times 0 + \frac{1}{11} \times 0 + \frac{1}{11} \times 0 + a \times 5$$

$$C_a(T_1) = \frac{4}{11} \times 0 + \frac{2}{11} \times 0 + \frac{3}{11} \times 0 + \frac{2}{11} \times 0.5 + a \times 4$$

$$C_a(T_2) = \frac{4}{11} \times 0 + \frac{2}{11} \times 0 + \frac{5}{11} \times 0.32 + a \times 3$$

$$C_a(T_3) = \frac{4}{11} \times 0 + \frac{7}{11} \times 0.49 + a \times 2$$

$$C_a(T_4) = \frac{11}{11} \times 0.463 + a \times 1$$

Grupo Prioritário

Gini = 0.463

Exemplo 3 de Pós-Poda: Cost-Complexity Pruning

$$C_a(T) = \sum_{m=1}^{|T|} N_m E_m(T) + a|T|$$

$$C_a(T_0) = \frac{4}{11} \times 0 + \frac{2}{11} \times 0 + \frac{3}{11} \times 0 + \frac{1}{11} \times 0 + \frac{1}{11} \times 0 + a \times 5$$

$$C_a(T_1) = \frac{4}{11} \times 0 + \frac{2}{11} \times 0 + \frac{3}{11} \times 0 + \frac{2}{11} \times 0.5 + a \times 4$$

$$C_a(T_2) = \frac{4}{11} \times 0 + \frac{2}{11} \times 0 + \frac{5}{11} \times 0.32 + a \times 3$$

$$C_a(T_3) = \frac{4}{11} \times 0 + \frac{7}{11} \times 0.49 + a \times 2$$

$$C_a(T_4) = 1 \times 0.463 + a \times 1$$

Exemplo 3 de Pós-Poda: Cost-Complexity Pruning

$$C_a(T) = \sum_{m=1}^{|T|} N_m E_m(T) + a|T|$$

$$C_a(T_0) = 0 + 0 + 0 + 0 + 0 + a \times 5$$

$$C_a(T_1) = 0 + 0 + 0 + \frac{2}{11} \times 0.5 + a \times 4$$

$$C_a(T_2) = 0 + 0 + \frac{5}{11} \times 0.32 + a \times 3$$

$$C_a(T_3) = 0 + \frac{7}{11} \times 0.49 + a \times 2$$

$$C_a(T_4) = 1 \times 0.463 + a \times 1$$

Exemplo 3 de Pós-Poda: Cost-Complexity Pruning

$$C_a(T) = \sum_{m=1}^{|T|} N_m E_m(T) + a|T|$$

$$C_a(T_0) = 0 + a \times 5$$
$$C_a(T_1) = 0.09 + a \times 4$$
$$C_a(T_2) = 0.15 + a \times 3$$
$$C_a(T_3) = 0.31 + a \times 2$$
$$C_a(T_4) = 0.463 + a \times 1$$

Exemplo 3 de Pós-Poda: Cost-Complexity Pruning

$$caso\ a = 0.08$$

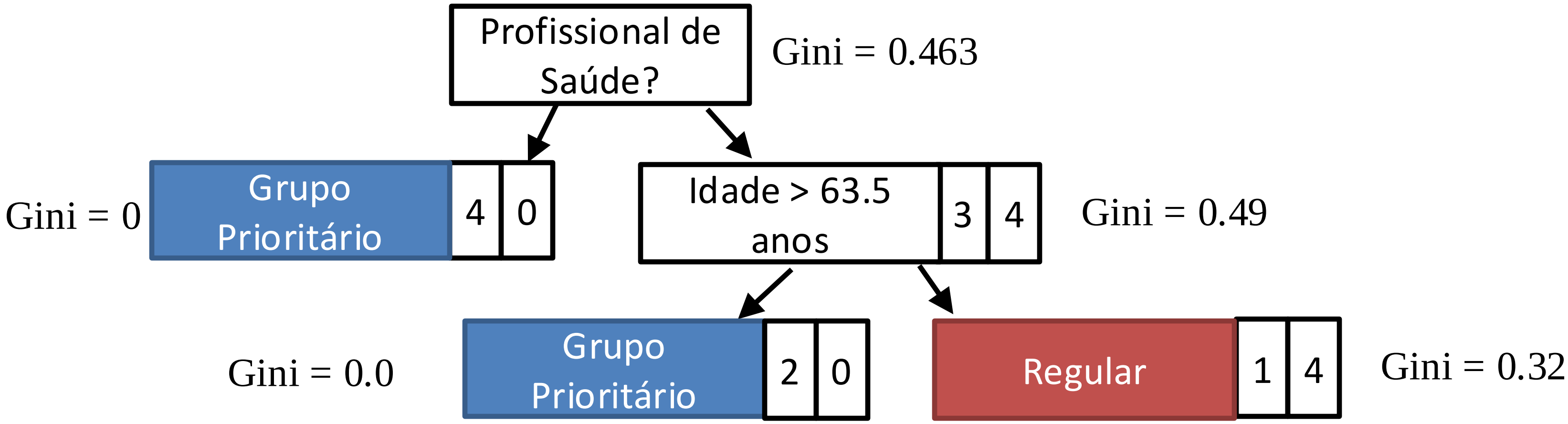
$$C_a(T_0) = 0.08 \times 5 = 0.4$$

$$C_a(T_1) = 0.09 + 0.08 \times 4 = 0.41$$

$$\boldsymbol{C_a(T_2) = 0.15 + 0.08 \times 3 = 0.39}$$

$$C_a(T_3) = 0.31 + 0.08 \times 2 = 0.47$$

$$C_a(T_4) = 0.463 + 0.08 \times 1 = 0.543$$



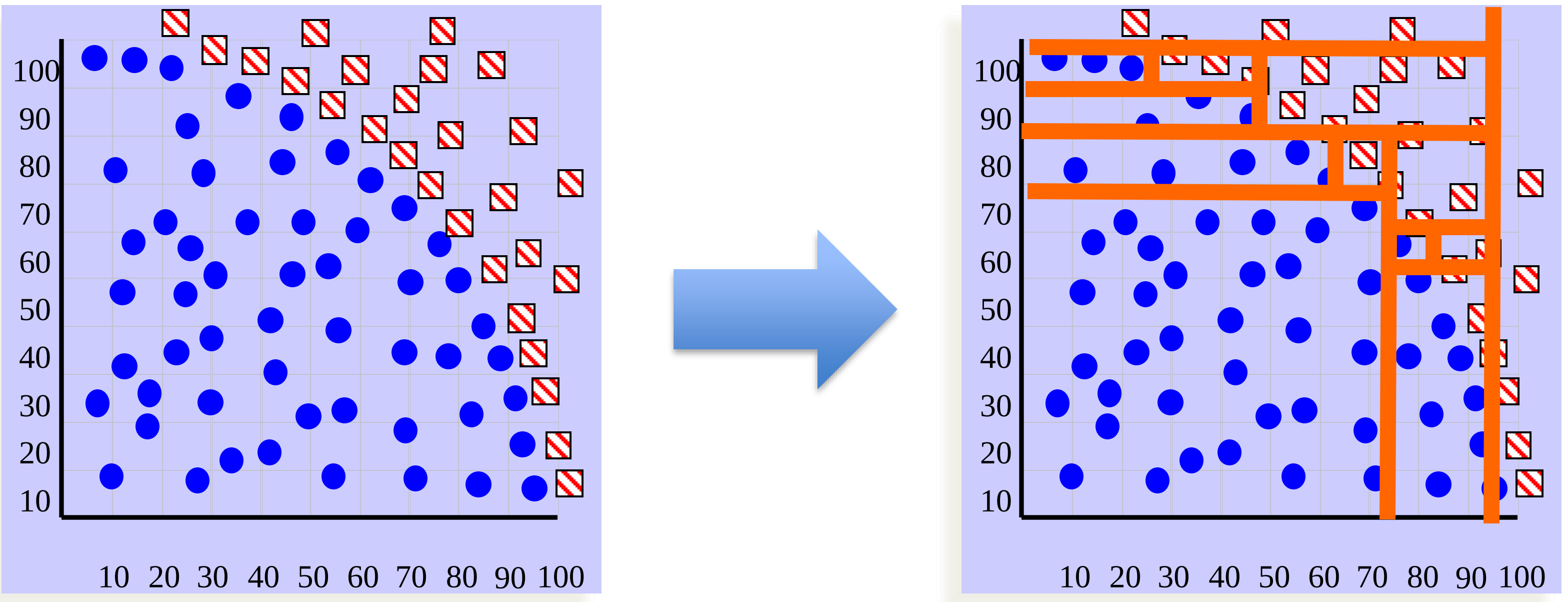
Outros tipos de Pós-Poda

- Pessimistic Error Pruning (Quinlan 1987)
- Error-Based Pruning (Quinlan 1993)
- Minimum-Error Pruning (Niblett e Bratsko 1986)
- Critical-Value Pruning (Mingers 1987)
- Cost-Complexity Pruning (Breiman et al. 1984)

- Sugestão de Leitura:
  - Artigo no moodle (Esposito et al. 1997)

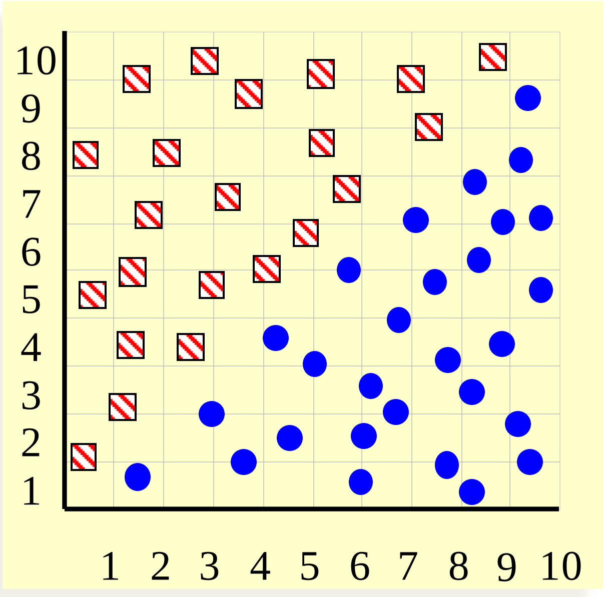
Alternativas às Desvantagens

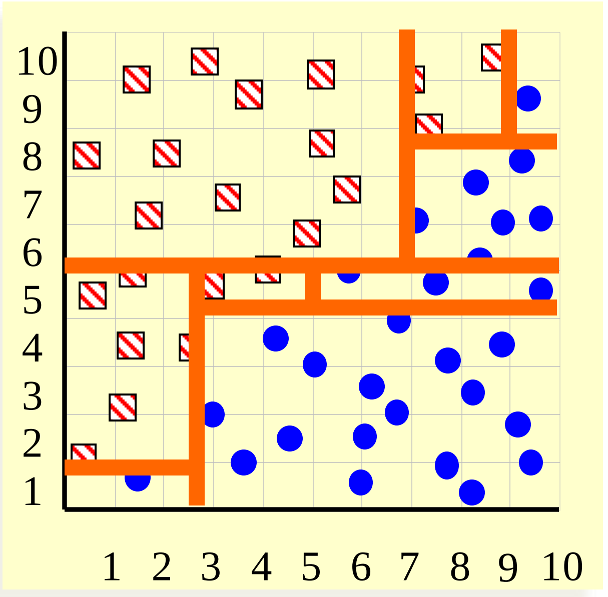
- Hiperplanos paralelos aos eixos
  - Vamos pensar no seguinte exemplo:



Alternativas às Desvantagens

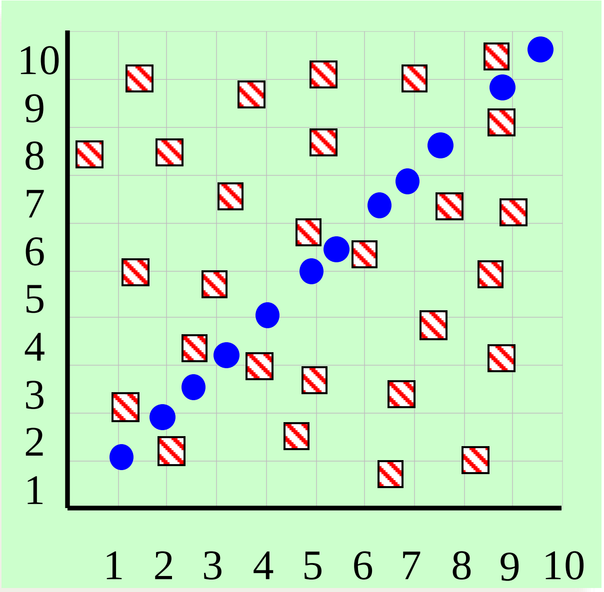
- Hiperplanos paralelos aos eixos
  - Outro exemplo:





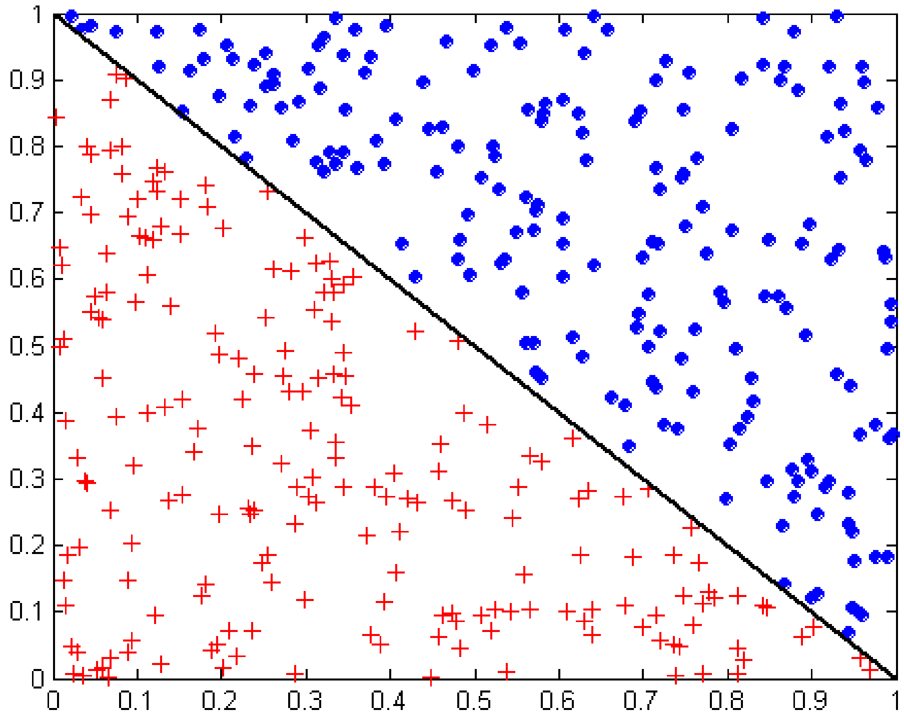
Alternativas às Desvantagens

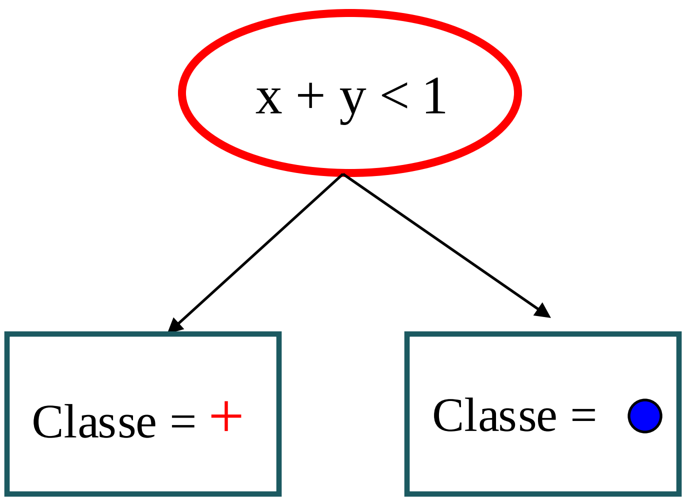
- Hiperplanos paralelos aos eixos
  - Mais um exemplo:



Alternativas às Desvantagens

- Hiperplanos paralelos aos eixos
  - Solução?
    - Árvores Oblíquas!!





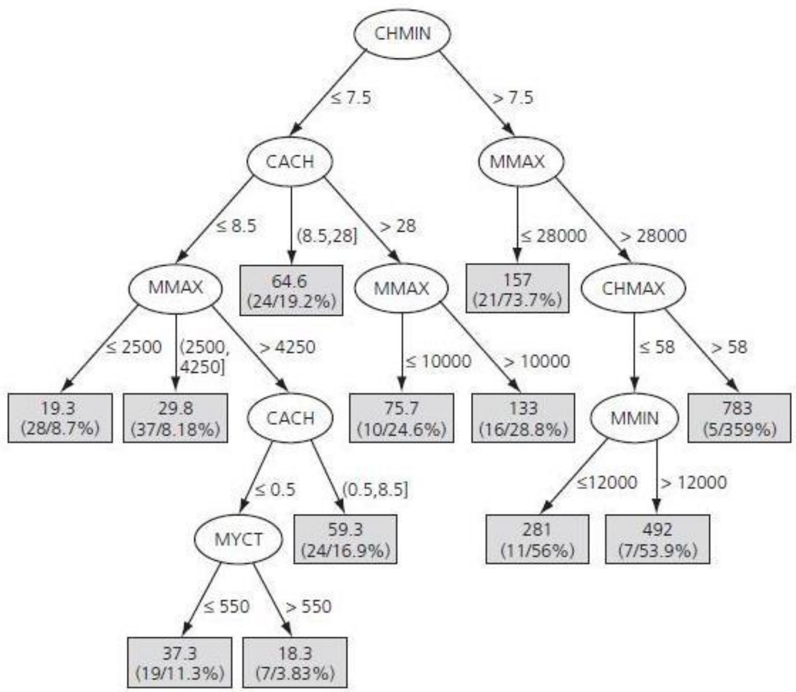
Desvantagens?

Alternativas às Desvantagens

- Solução localmente ótima pode estar longe do ótimo global
  - Solução?
    - Heurísticas que aproximam o ótimo global
      - Ex: computação evolutiva!
        - » Algoritmos Genéticos, Programação Genética
        - » Ver artigo no moodle
          - A Survey of Evolutionary Algorithms for Decision-Tree Induction

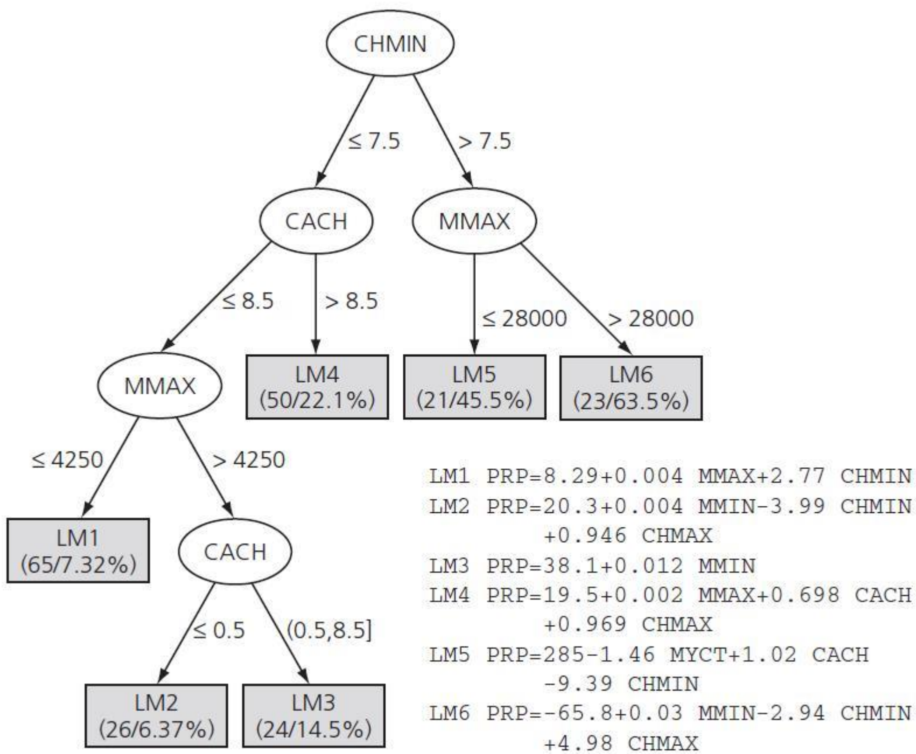
Árvores de Decisão para Problemas de Regressão

- Árvores de Regressão
  - Folha contém média dos valores do atributo alvo dos exemplos de treino que chegam até lá



Árvores de Decisão para Problemas de Regressão

- Árvores de Modelos
  - Folha contém função de regressão (não-)linear calculada sobre as instâncias que chegam até lá



Árvores de Decisão para Problemas de Regressão

- Principal mudança: medida de divisão de nós
  - Exemplo: *standard deviation reduction* (SDR)
    - Mesma fórmula genérica do "ganho"
    - Em vez de entropia ou Gini, apenas calcular o desvio padrão do atributo alvo para as instâncias de cada nó e ponderá-las pelas frequências

$$SDR = SD(v_{pai}) - \sum_{t=1}^{k} \frac{N(v_t)}{N} SD(v_t)$$

Exemplos de Algoritmos

- ID3 (Quinlan 1986)
  - Iterative Dichotomiser 3
  - Lida apenas com atributos nominais
  - Medida de impureza: ganho de informação
  - Tipo de poda: pré-poda (limite de instâncias)
- C4.5 (Quinlan 1993)
  - J48 (Weka), C5.0 (comercial)
  - Atributos discretos e contínuos
  - Medida de impureza: gain ratio
  - Tipo de poda: pós-poda (error-based pruning)

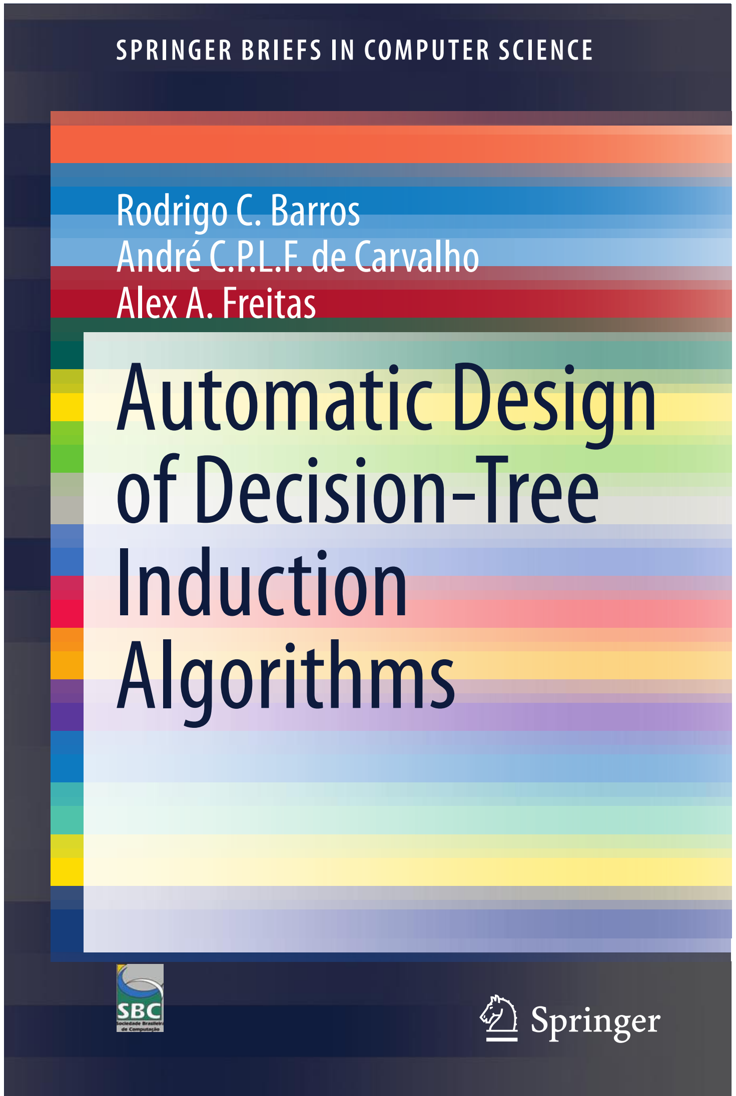
Exemplos de Algoritmos

- CART (Breiman et al. 1984)
  - Classification and Regression Trees
  - Árvores de Classificação e Regressão
  - Atributos discretos e contínuos
  - Divisões sempre binárias (agrega categorias)
  - Medida de impureza: índice Gini / twoing / sum of squares
  - Tipo de poda: pós-poda (cost-complexity pruning)

Exemplos de Algoritmos

- M5 (Quinlan 1992)
  - M5P (Weka)
  - Árvores de Regressão e Árvores de Modelos
  - Atributos discretos e contínuos
  - Medida de impureza: SDR
  - Tipo de poda: erro corrigido (leva em conta o número de parâmetros dos modelos lineares)

Sugestão de Leitura



Capítulo 2! (livro inteiro está no moodle)

Sugestão de Leitura

- Seção 4.3 (Tan et al., 2006)
- Capítulo 6 (Faceli et al., 2011)
- Artigos no Moodle

Créditos e Referências

Slides adaptados dos originais gentilmente cedidos por:

- Prof. Dr. Rodrigo Coelho Barros (PUCRS)
- André Carvalho, Eduardo Hruschka, Ricardo Campello (ICMC-USP)
- Pang-Ning Tan (Michigan State University)
- Eamon Keogh (University of California at Riverside)
  - http://www.cs.ucr.edu/~eamonn/
  - eamonn@cs.ucr.edu

- Tan, P. N., Steinbach, M., Kumar, V. **Introduction to Data Mining.** Addison-Wesley, 2005. 769 p.

- Faceli et al. **Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina**. LTC, 2011. 378 p.