Aprendizado de Máquina II

Árvores de Decisão

Profª. Renata De Paris

Especialização em Ciência de Dados

UCS
UNIVERSIDADE
DE CAXIAS DO SUL

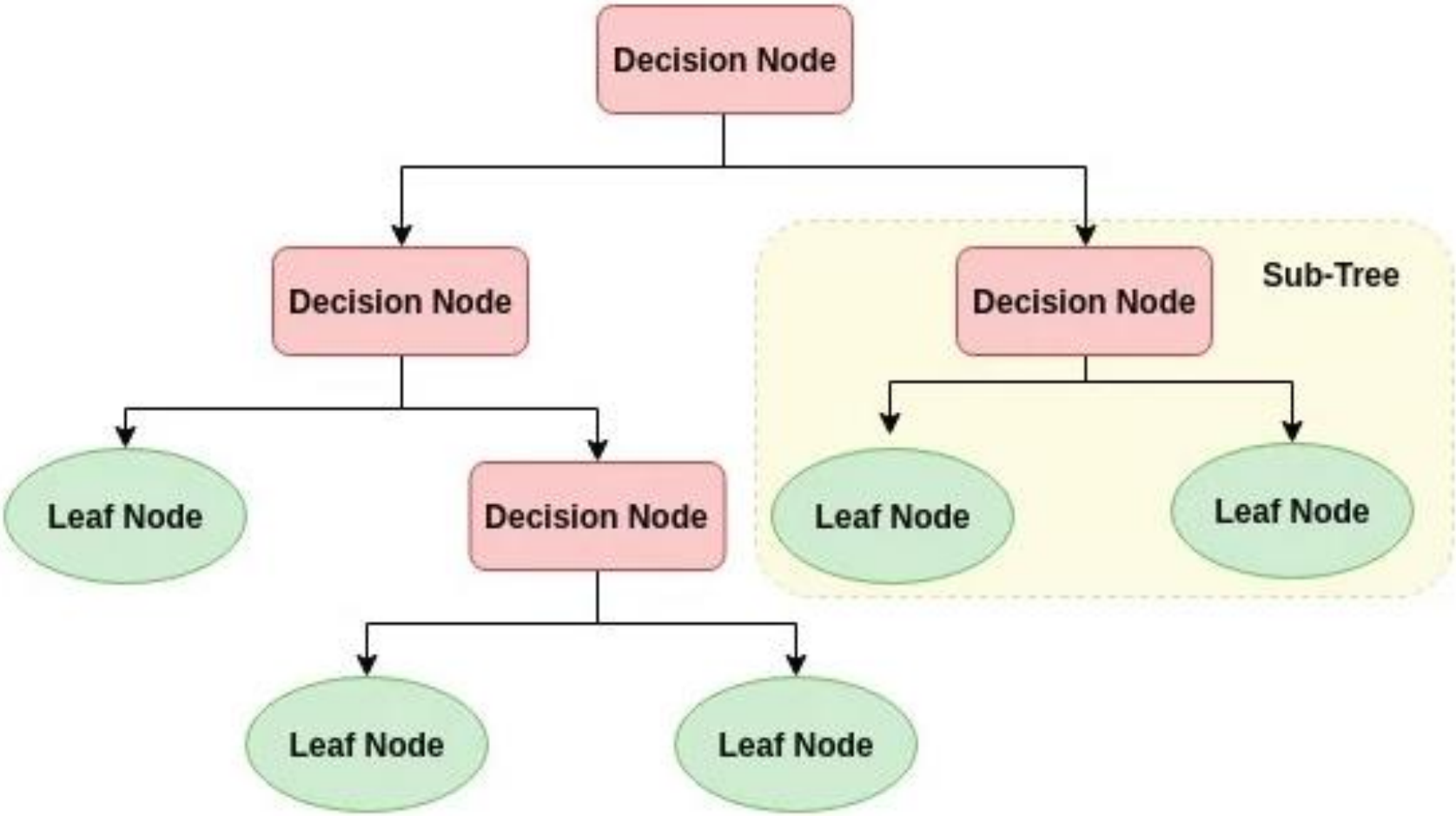
Roteiro da Aula

- Definição
- Exemplo de árvore de decisão
- Geração da árvore de decisão
- Métricas utilizadas para selecionar a melhor divisão
- Avaliação do desempenho em Árvore de Decisão
- Atividade

Árvores de Decisão

- **Definição**
  - Utiliza a estratégia dividir para conquistar.
  - Um problema complexo é dividido em problemas mais simples (subproblemas).
  - Para cada subproblema é aplicada uma mesma estratégia recursivamente
- **Representação**
  - Grafo acíclico direcionado que possuem dois tipos de nós:
    - Nó de decisão: nó de divisão com dois ou mais sucessores.
    - Nó folha: nó terminal, rotulado como uma função.
- **Algortimos**
  - ID3 (Quilan, 1979).
  - CART (Breiman et al., 1984).
  - C4.5 (J48 no Weka) (Quilan, 1993) .

Árvore de Decisão



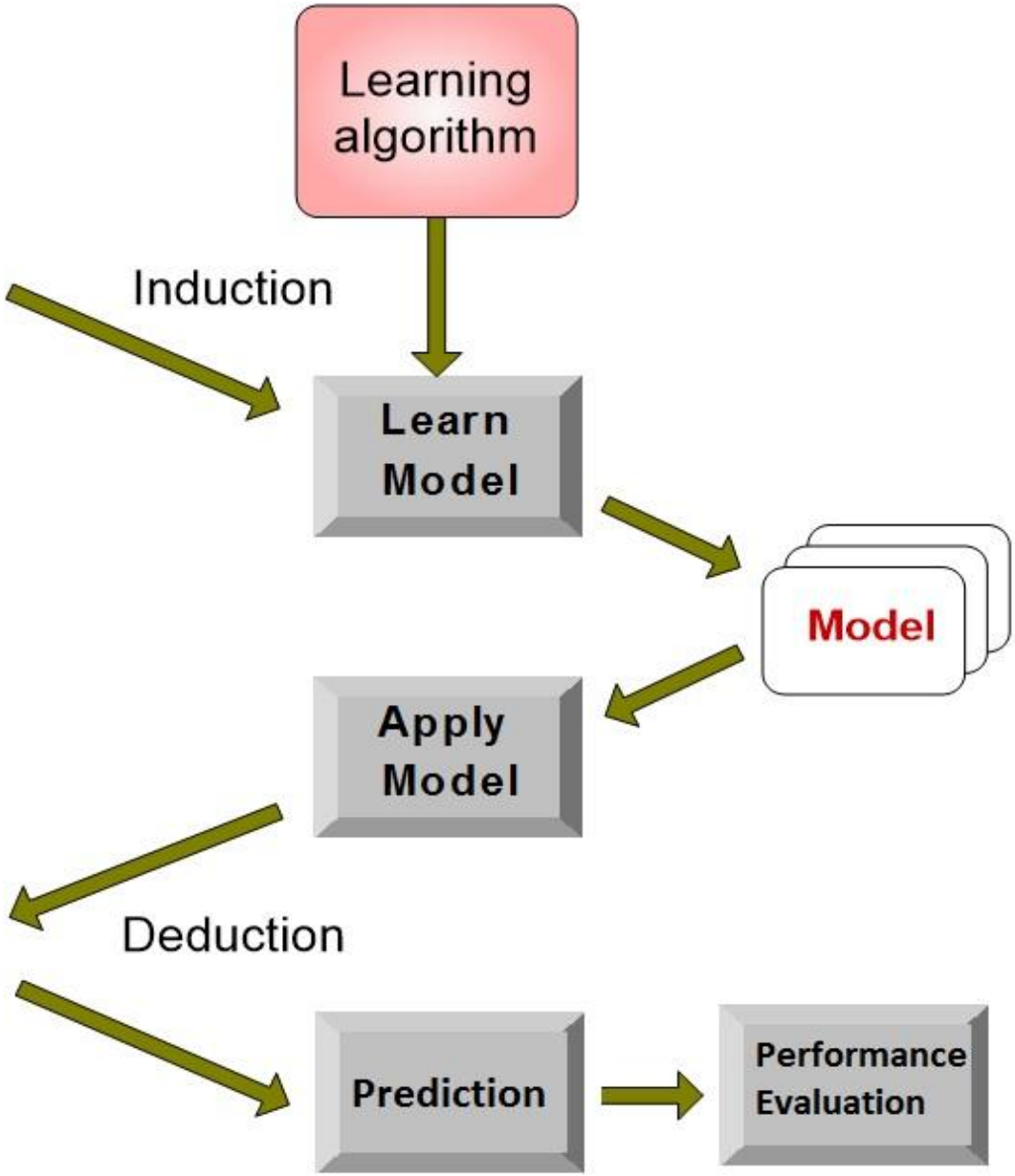
Esquema da Tarefa de Classificação

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	?

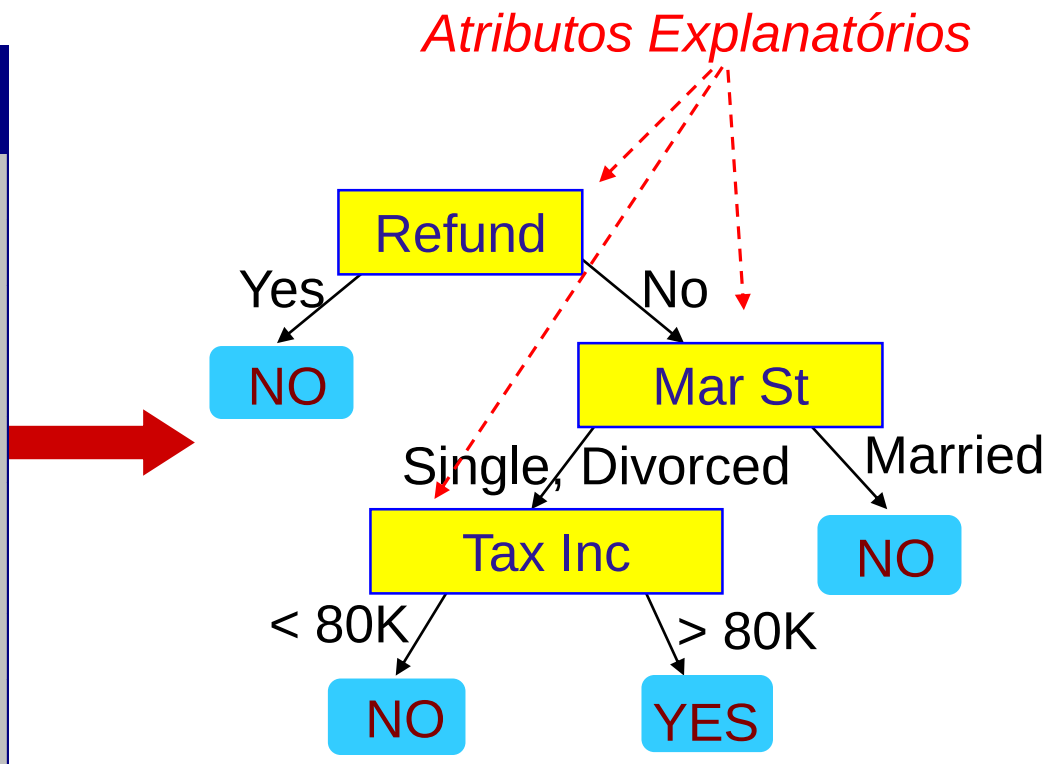
Test Set



Exemplos de Árvore de Decisão

categórico categórico contínuo 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



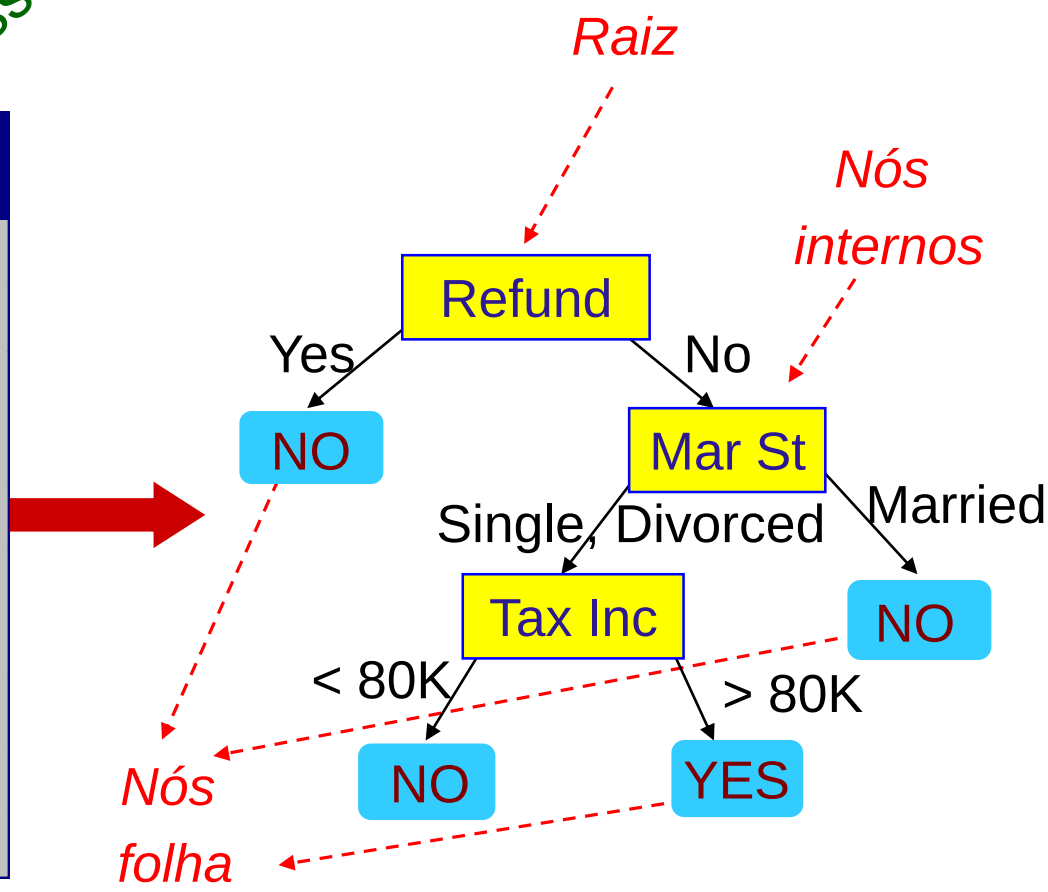
Dados de Treino

Modelo: Árvore de Decisão

Exemplos de Árvore de Decisão

categórico categórico contínuo 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



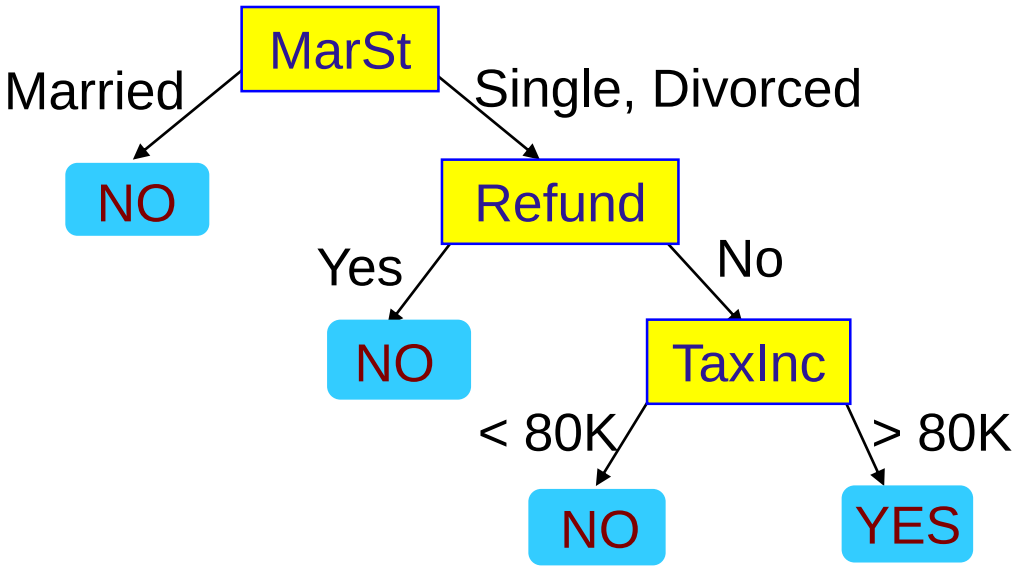
Dados de Treino

Modelo: Árvore de Decisão

Exemplos de Árvore de Decisão

categórico categórico contínuo classe

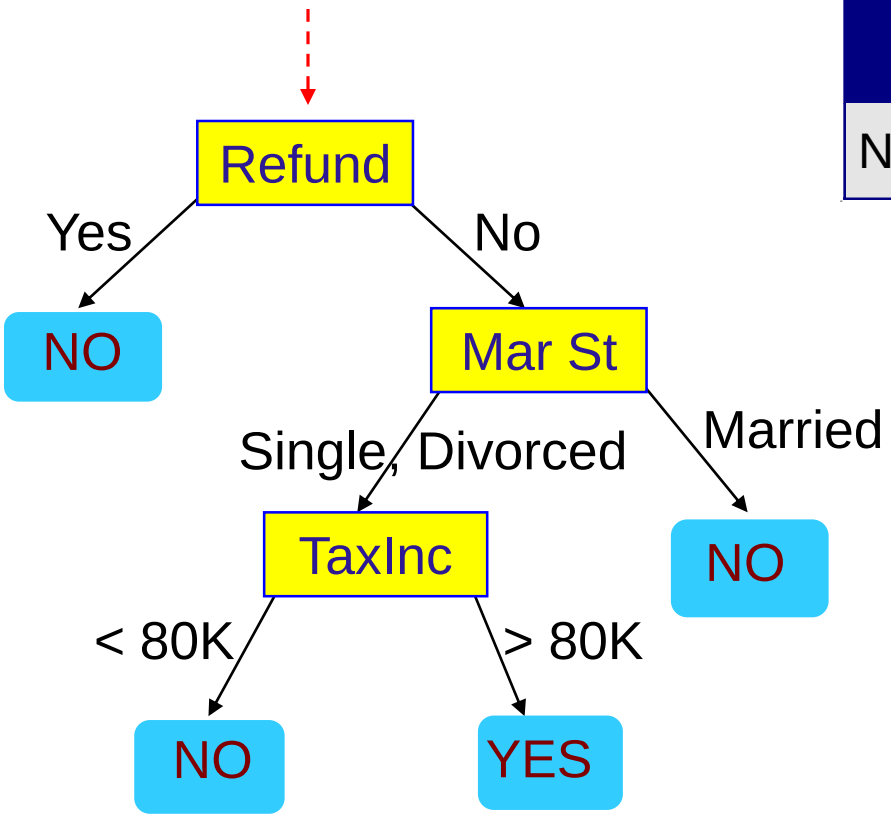
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**



Pode existir mais de uma árvore de decisão adequada para os mesmos dados!

Aplicando o Modelo aos Dados de Teste

Início na raiz da árvore.



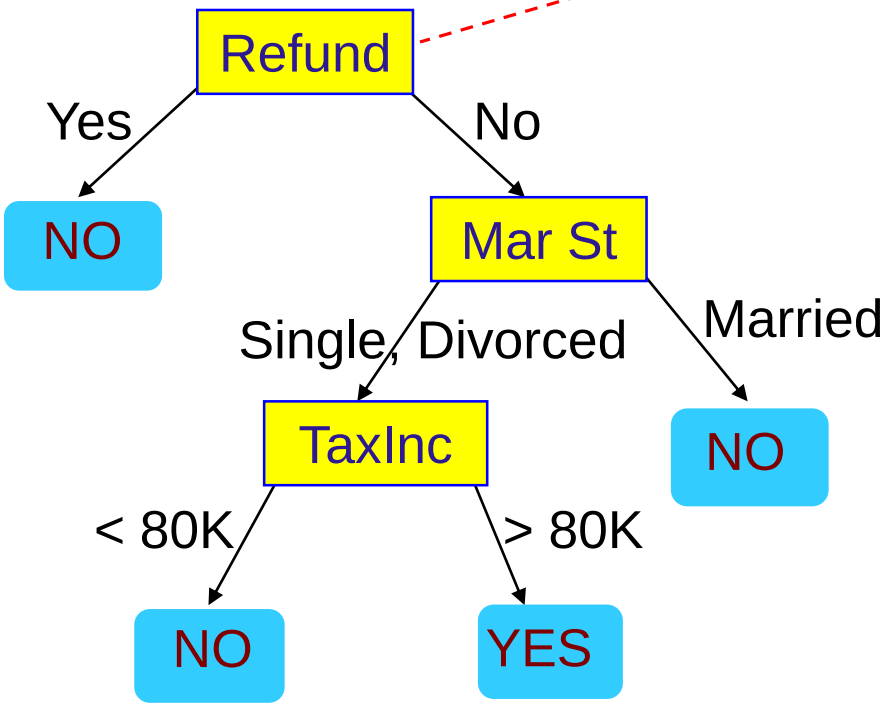
Dados de Teste

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Aplicando o Modelo aos Dados de Teste

Dados de Teste

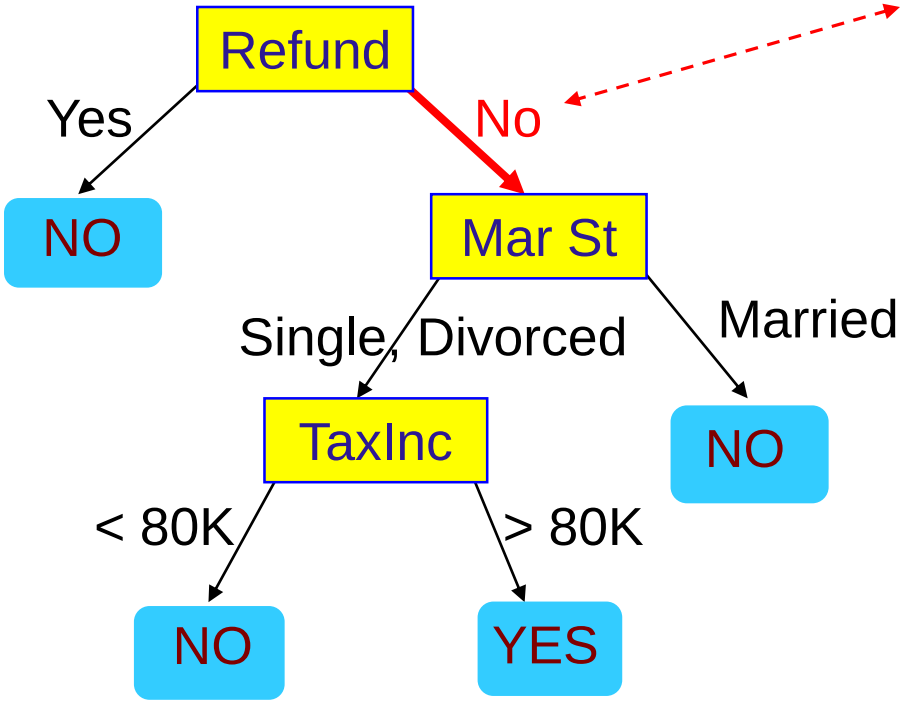
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Aplicando o Modelo aos Dados de Teste

Dados de Teste

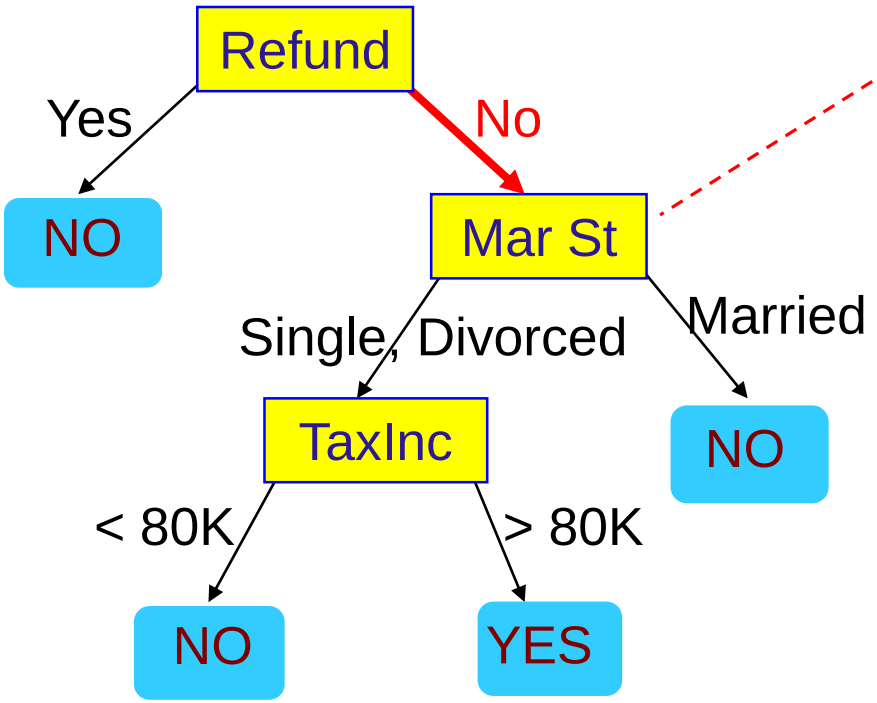
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Aplicando o Modelo aos Dados de Teste

Dados de Teste

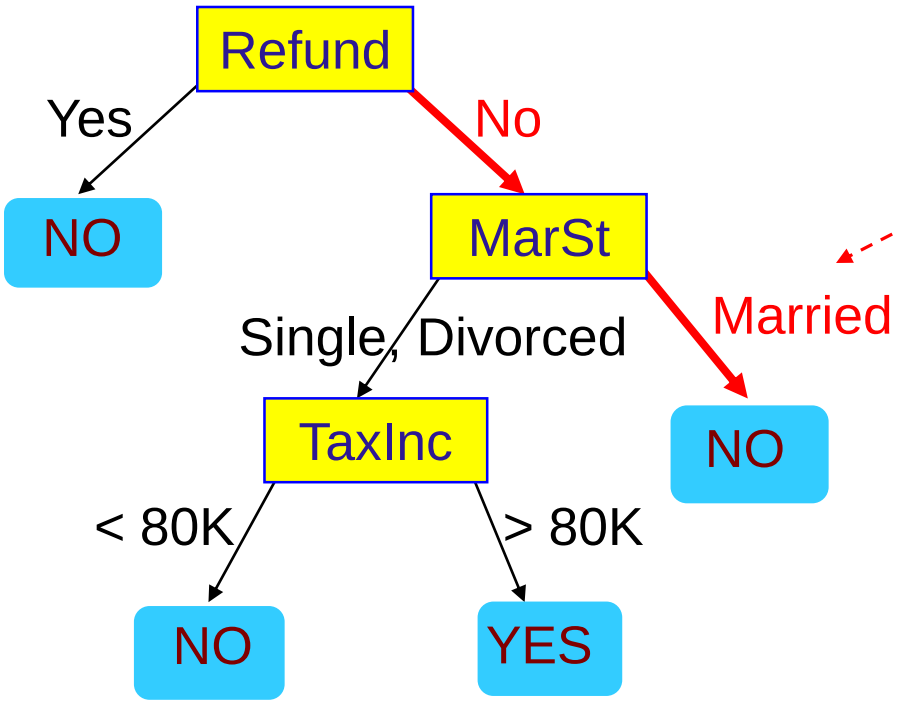
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Aplicando o Modelo aos Dados de Teste

Dados de Teste

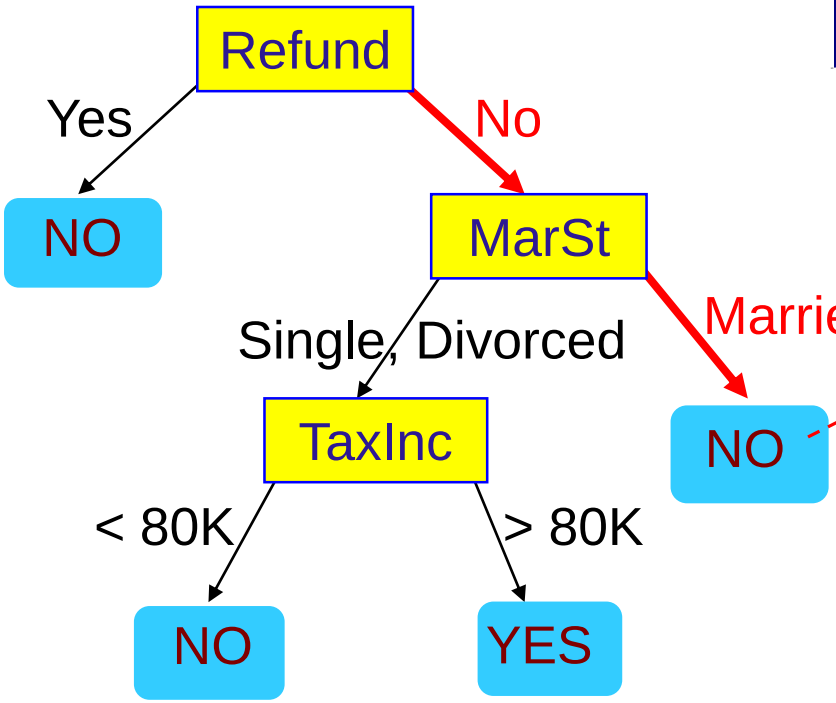
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Aplicando o Modelo aos Dados de Teste

Dados de Teste

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Atribuir “NO” para Calote.

Indução de Árvores de Decisão

Uma árvore de decisão abrange **todo o espaço de instâncias**, permitindo executar **predições** para **qualquer** exemplo de **entrada**.

- **Características**
  - Utiliza uma abordagem **não paramétrica** para construir modelos de classificação.
  - Encontrar uma árvore de decisão adequada é um problema NP-completo devido a estratégia de particionamento **recursiva**, **gulosa** e de **cima para baixo (top-down)** usada para aumentar a árvore.
  - Pequenas árvores de decisão são muito fáceis de **interpretar**.

Indução de Árvores de Decisão

- **Características da Indução (Continuação)**

  - A **construção** de árvores é computacionalmente **barata** mesmo para uma grande quantidade de dados.
  - A **classificação** dos dados de testes em uma árvore de decisão é extremamente **rápida**.
  - Os algoritmos de árvores de decisão são bastante **robustos** para a presença de ruídos, especialmente quando possuem métodos para evitar o *overfitting*.
  - A presença de atributos **redundantes** afeta negativamente a acurácia de árvores de decisão.

Indução de Árvores de Decisão

- Estratégia Gulosa (Greedy).
  - Particionar os registros baseado no teste de um atributo que otimiza um certo critério.

- Problemas:
  - Determinar como particionar os registros.
    - Como especificar a condição de teste para o atributo?
    - Como determinar qual é o melhor particionamento?
  - Determinar quando parar de particionar.

Como Especificar a Condição de Teste?

- Depende do tipo do atributo
  - Nominal
    - cor, identificação, profissão, ….
  - Ordinal
    - gosto (ruim, médio, bom), dias da semana, …
  - Contínuo (numérico)
    - peso, tamanho, idade, temperatura, …

- Depende do número de ramos para particionar
  - Particionamento em 2 ramos.
  - Particionamento em n ramos.

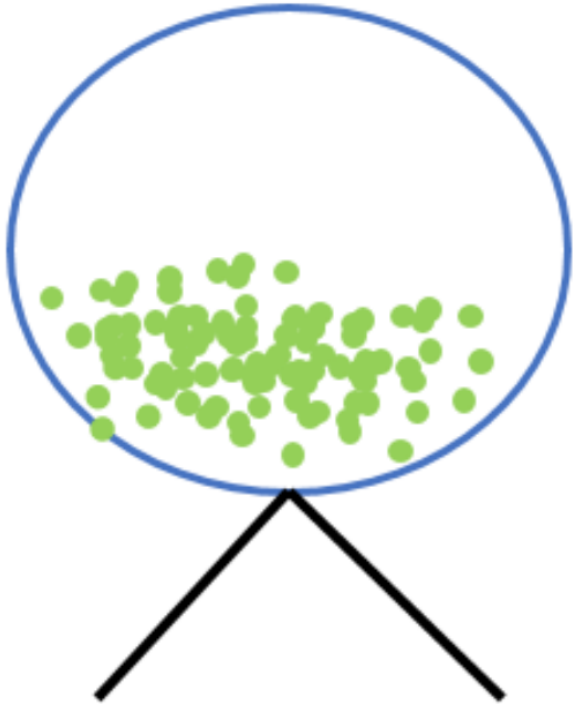
Particionamento em Atributos Contínuos

- Diferentes maneiras de tratar:
  - Discretização para transformar em um atributo categórico ordinal.
    - Estático – discretizado uma vez no início.
    - Dinâmico – intervalos podem ser achados por particionamento em intervalos iguais, em frequências iguais, ou agrupamento.
  - Teste Binário: $(A < v)$ ou $(A \geq v)$
    - Considera todos os possíveis pontos de corte e procura o melhor.
    - Pode ser computacionalmente dispendioso.

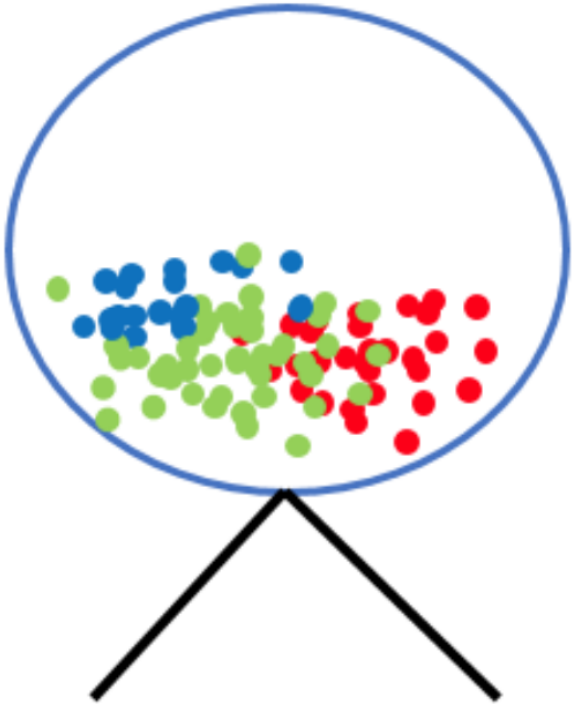
Para Determinar o Melhor Ponto de Particionamento

- **Grau de impureza**

Totalmente puro



Maior grau de impureza



Para Determinar o Melhor Ponto de Particionamento

- **Abordagem Gulosa**:
  - Nodos com distribuição homogênea de classes são preferidos.
- Necessita de uma métrica para medir a impureza do nodo:

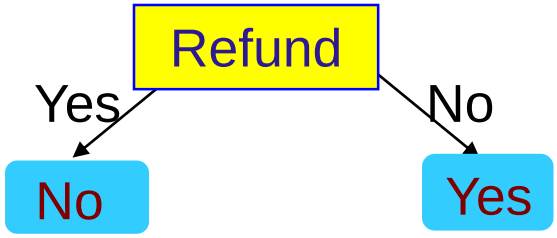
C0: 5 C1: 5	C0: 9 C1: 1
Não-homogêneo	Homogêneo
Alto grau de impureza	Baixo grau de impureza

Geração da Árvore de Decisão baseada no Algoritmo Hunt

Qual o melhor atributo?

Tid	Refund (categórico)	Marital Status (categórico)	Taxable Income (contínuo)	Cheat (classe)
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



Acertos = 3
Erros = 0

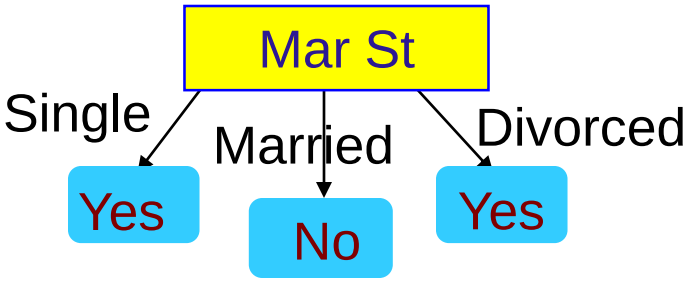
Acertos = 3
Erros = 4

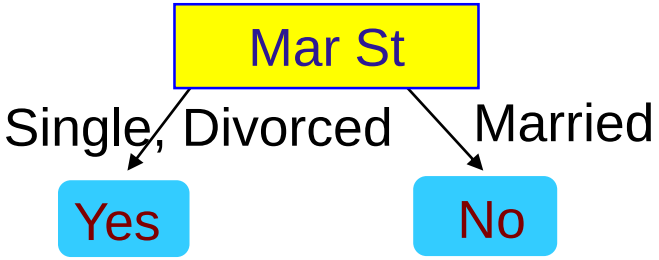
Acertos = 60%

Qual o melhor atributo?

categórico categórico contínuo 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





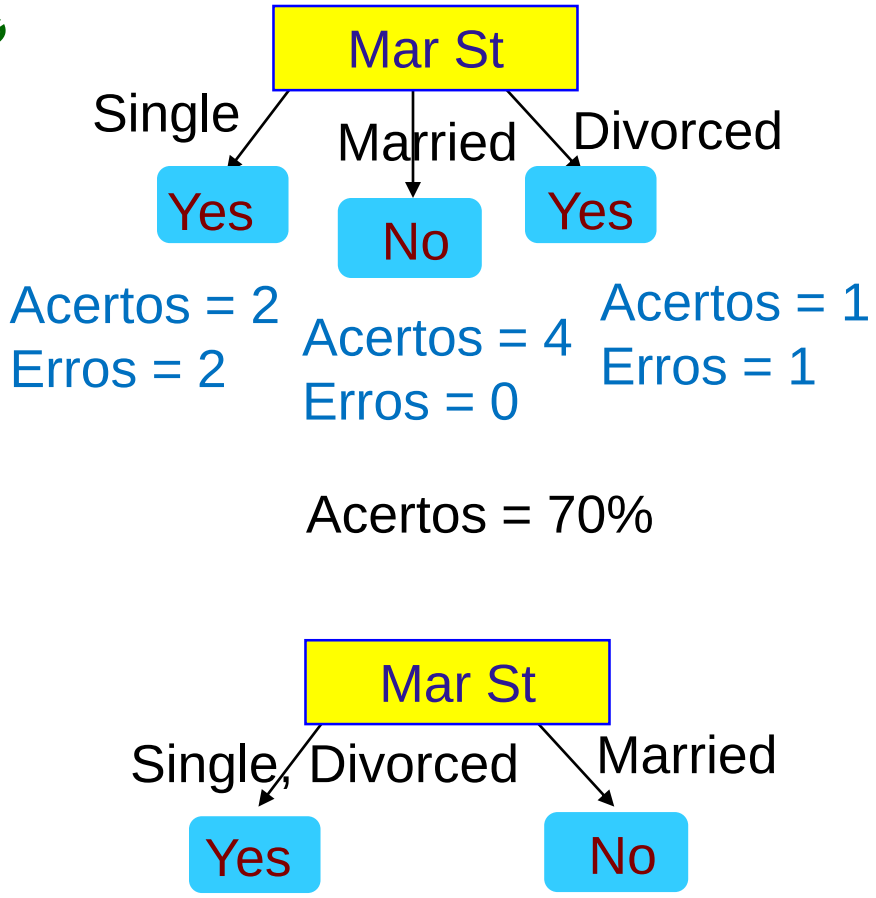
Dados de Treino

Qual o melhor atributo?

categórico | categórico | contínuo | 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

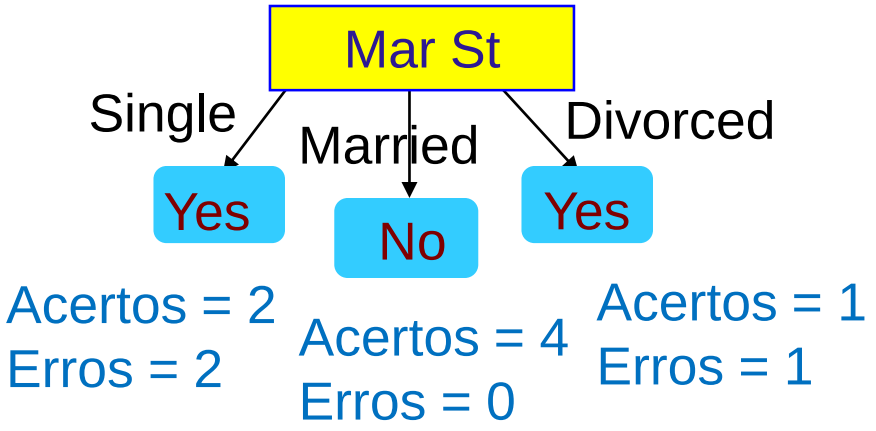
Dados de Treino



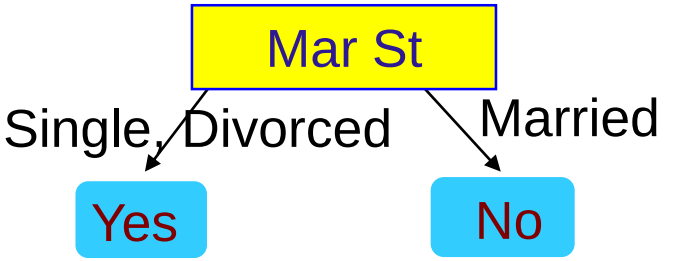
Qual o melhor atributo?

Tid	Refund (categórico)	Marital Status (categórico)	Taxable Income (contínuo)	Cheat (classe)
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



Acertos = 70%



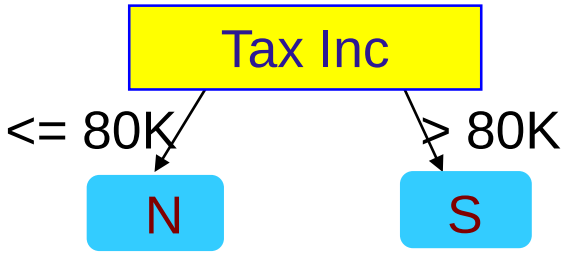
Acertos = 70%

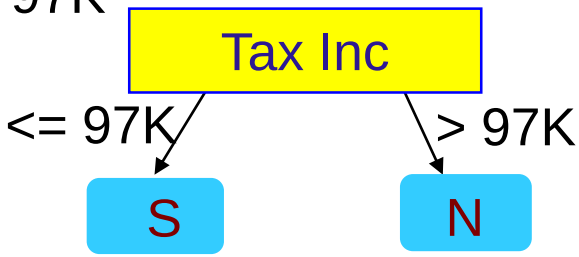
Qual o melhor atributo?

categórico categórico contínuo 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

Taxable Income	Cheat	
60K	No	
70K	No	
75K	No	← 80K
85K	Yes	
90K	Yes	
95K	Yes	← 97K
100K	No	
120K	No	
125K	No	
220K	No	





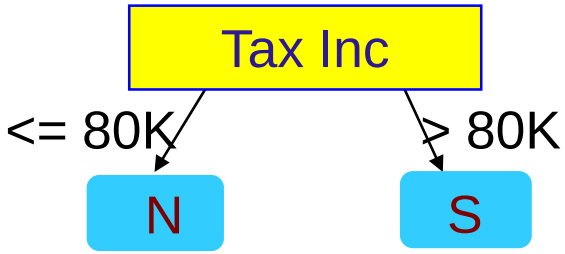
Dados de Treino

Qual o melhor atributo?

categórico categórico contínuo classe

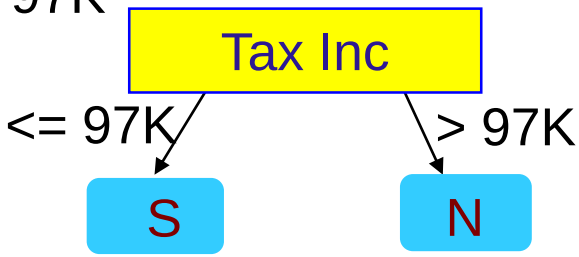
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Taxable Income	Cheat	
60K	No	
70K	No	
75K	No	
85K	Yes	← 80K
90K	Yes	
95K	Yes	← 97K
100K	No	
120K	No	
125K	No	
220K	No	



Acertos = 3 Acertos = 3
Erros = 0 Erros = 4

60%



Dados de Treino

Qual o melhor atributo?

categórico categórico contínuo classe

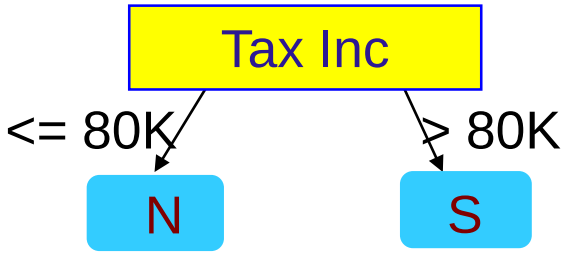
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Dados de Treino

Taxable Income	Cheat
60K	No
70K	No
75K	No
85K	Yes
90K	Yes
95K	Yes
100K	No
120K	No
125K	No
220K	No

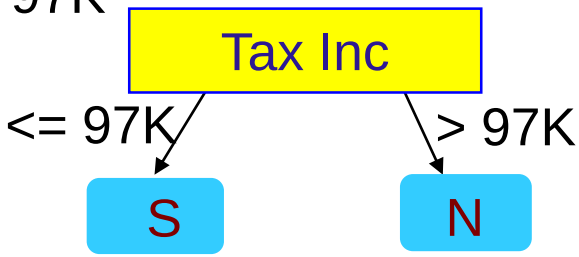
← 80K

← 97K



Acertos = 3 Acertos = 3
Erros = 0 Erros = 4

60%



Acertos = 3 Acertos = 4
Erros = 3 Erros = 0

70%

Funcionamento do Algoritmo Hunt

- D_t é o conjunto de treino que obtém um nodo t.
  - Se D_t possui registros que pertencem a mesma classe y_t, então t é um nodo folha rotulado como y_t.
  - Se D_t é um conjunto vazio, então t é um nodo folha rotulado pela classe padrão y_d.
  - Se D_t possui registros que pertencem a mais do que uma classe, utilize um atributo teste para dividir os dados em subárvores menores. Aplique recursivamente o procedimento para cada subárvore.

Funcionamento do Algoritmo Hunt

- Quando os registros de treino devem ser divididos?
  - Quando cada etapa recursiva do algoritmo encontra uma condição de teste de atributo que divide os registros em pequenas subárvores.

- Quando o procedimento de divisão deve parar?
  - Quando todos os registros pertencerem a uma mesma classe ou todos os registros possuírem valores de atributos idênticos.

Métricas Utilizadas para Selecionar a Melhor Divisão

Métricas para Avaliar a Impureza de Nodos

- Índice Gini

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

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

- Entropia

$$Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)$$

- Erros de classificação

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

Métricas de Impureza: GINI

- Índice Gini para um dado nodo t :

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

(NOTA: $p(j \mid t)$ é a frequência relativa da classe j no nodo t).

  - Máximo (1 - $1/n_c$) quando registros são igualmente distribuidos entre todas as classes, implicando na informação menos interessante.
  - Mínimo (0.0) quando todos os registros pertencem a uma única classe, implicando na informação mais interessante.

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	

Calculando o GINI

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

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

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Gini = 1 – P(C1)² – P(C2)² = 1 – 0 – 1 = 0

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

P(C1) = 1/6 P(C2) = 5/6

Gini = 1 – (1/6)² – (5/6)² = 0.278

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

P(C1) = 2/6 P(C2) = 4/6

Gini = 1 – (2/6)² – (4/6)² = 0.444

Árvore elementar: Calculando o Índice GINI

	categórico	categórico	contínuo	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

No

Acertos = 7
Erros = 3

70%

$Gini = 1 – (7/10)^2 – (3/10)^2$

$Gini = 1 – 49/100 – 9/100$

$Gini = (100 – 49 – 9)/100$

$Gini = 0,42$

Particionamento baseado no Índice GINI

- Usado pelos algoritmos CART, SLIQ, SPRINT.
- Quando um nodo p é particionado em k partições (filhos), a qualidade do particionamento é calculado por,

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

onde, n_i = número de registros no filho i,

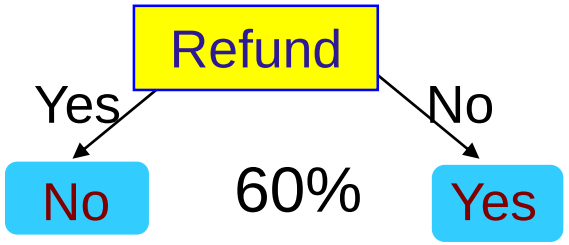
n = número de registros no nodo p.

Atributos Categóricos: Calculando o GINI

categórico | categórico | contínuo | 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



Acertos = 3
Erros = 0

Acertos = 3
Erros = 4

$Gini = 1 – (3/3)^2 – (0/3)^2$
$Gini = 1 – 1 – 0$
$Gini = 0,0$

$Gini = 1 – (3/7)^2 – (4/7)^2$
$Gini = 1 – 9/49 – 16/49$
$Gini = (49 – 9 – 16)/49$
$Gini = 0,49$

$Gini_{split} = (3/10)*0,0 + (7/10)*0,49$

$Gini_{split} = 0 + 0,34$

$Gini_{split} = 0,34$

Atributos Categóricos: Calculando Índice GINI

- Para cada valor distinto, apurar população em cada classe do conjunto de dados
- Usar a matriz com populações para tomar a decisão

Particionamento em n ramos

	TipoVeículo		
	Familiar	Esportivo	Luxo
C1	1	2	1
C2	4	1	1
Gini	0.393		

Particionamento em 2 ramos
(busca pela melhor divisão de valores)

	TipoVeículo	
	{Esportivo, Luxo}	{Familiar}
C1	3	1
C2	2	4
Gini	0.400	

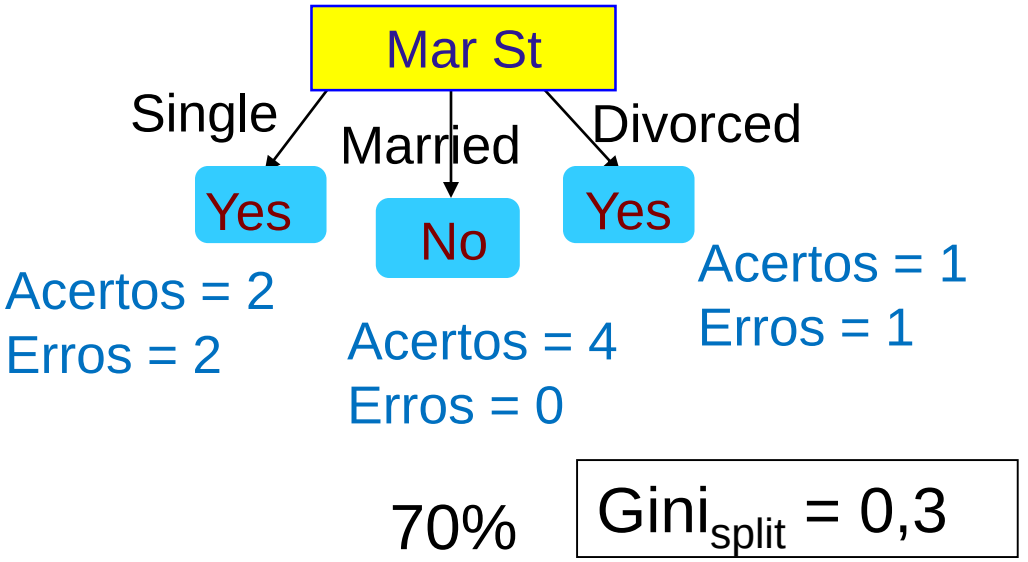
	TipoVeículo	
	{Esportivo}	{Familiar,Luxo}
C1	2	2
C2	1	5
Gini	0.419	

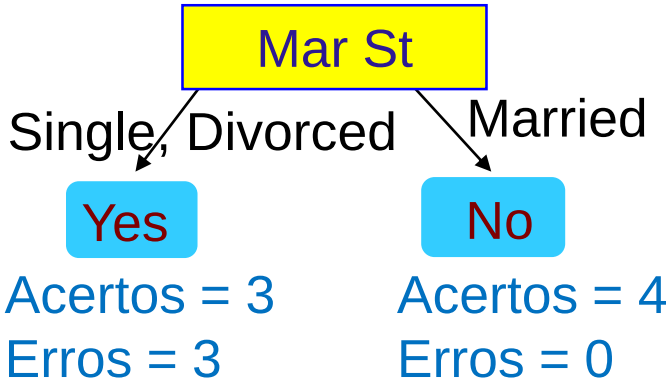
Atributos Categóricos: Calculando Índice GINI

categórico | categórico | contínuo | 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





Atributos Contínuos: Calculando Índice GINI

- Classificar valores existentes.
- Pesquisar linearmente estes valores, apurando a população envolvida, e calculando o índice GINI.
- Escolher a posição de particionamento que apresenta o menor índice GINI.

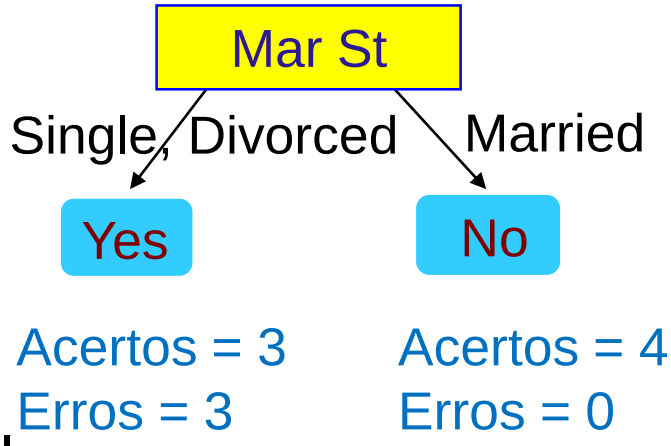
Calote	N		N		N		S		S		S		N		N		N		N	
	Rendim. Tributáveis																			
Valores Ordenados →	60		70		75		85		90		95		100		120		125		220	
Posições de Particionamento →	55		65		72		80		87		92		97		110		122		172		230	
	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
S	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0
N	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	

Induzindo o 2º Nível da Árvore de Decisão

categórico categórico contínuo classe

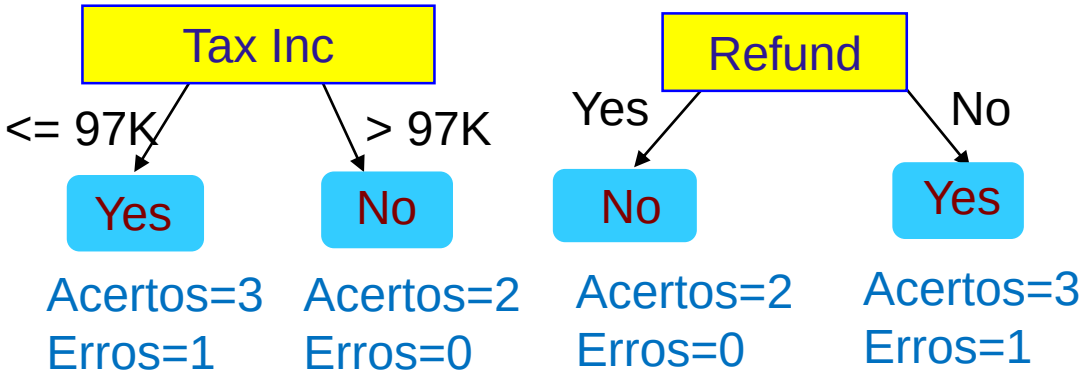
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Dados de Treino



1º Nível

2º Nível



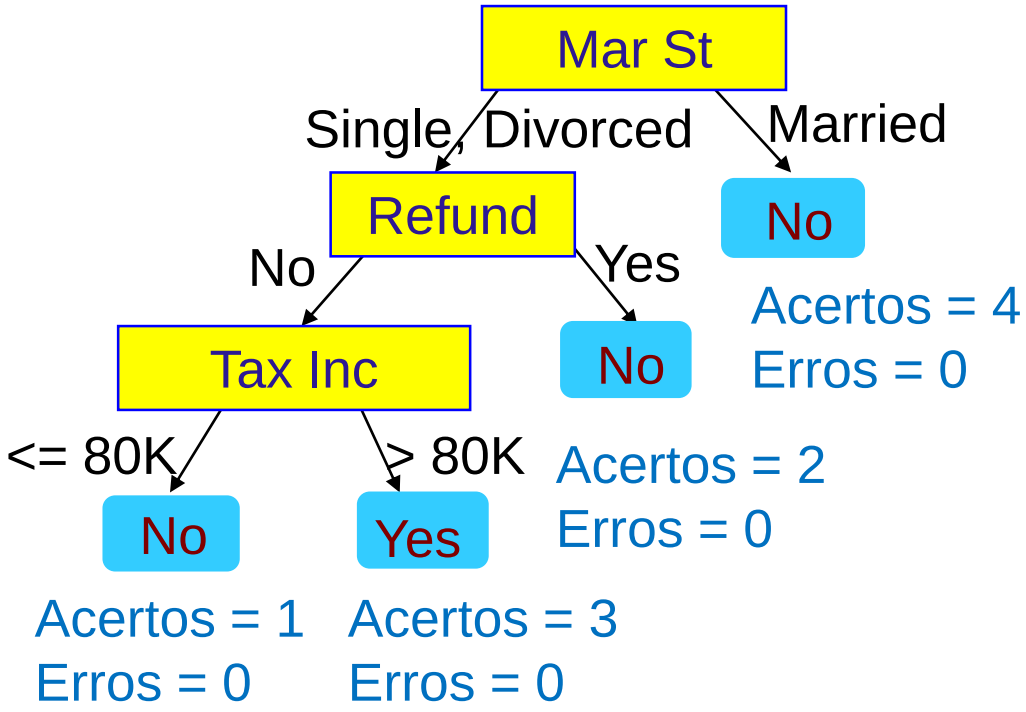
$Gini_{split} = 0,25$

$Gini_{split} = 0,25$

Induzindo o 3º Nível da Árvore de Decisão

categórico | categórico | contínuo | 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



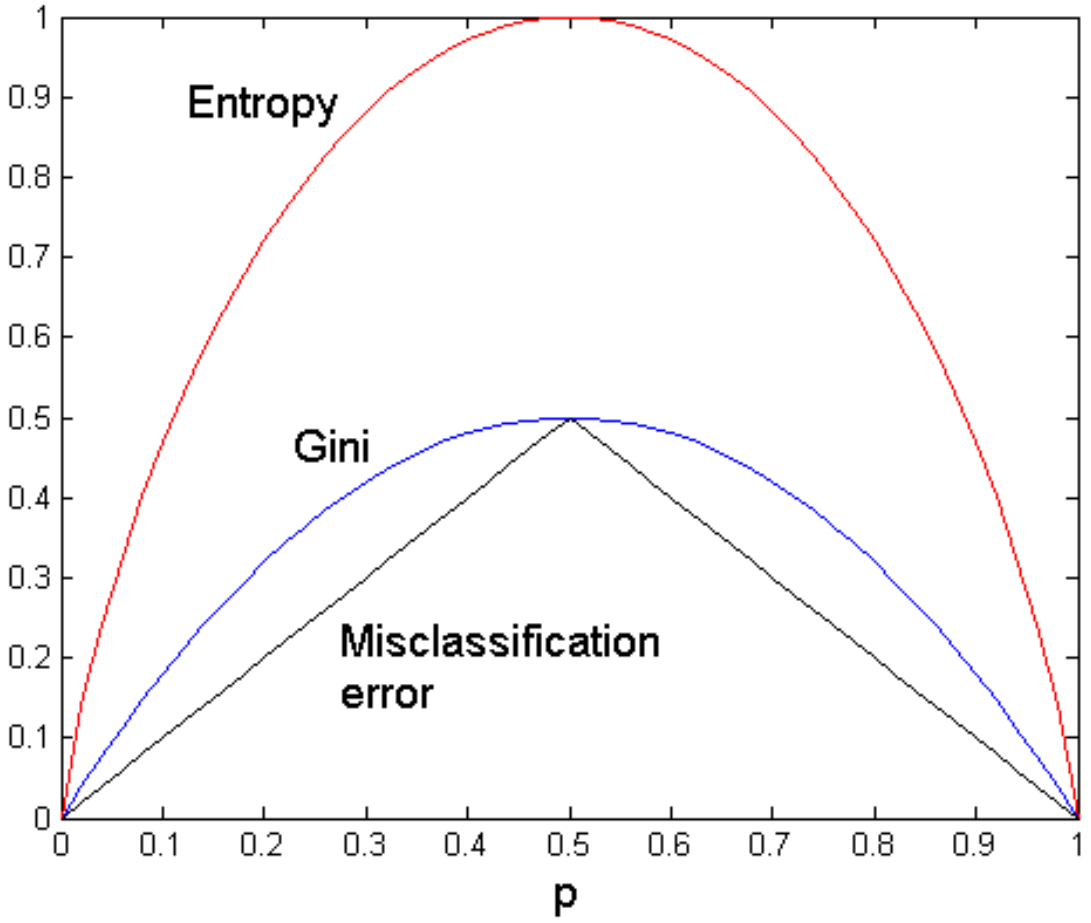
Dados de Treino

Ginisplit = 0,0

Medidas para Selecionar a Melhor Divisão

- As medidas são baseadas no grau de impureza dos nodos filhos.
- Quanto menor o grau de impureza mais distorcida será a distribuição da classe.
- Por exemplo:
  - Um nodo com classe de distribuição uniforme (0,1) tem impureza zero.
  - Um nodo com distribuição de classe uniforme (0.5,0.5) possui uma impureza mais alta.

Comparação entre as medidas de impurezas para problemas de classificação binária



Medidas para Selecionar a Melhor Divisão

- Determine o Gini, a Entropia e o Erro dos nodos abaixo.

Nodo N_1	**Quant**
Classe=0	0
Classe=1	6

Gini =

Entropy =

Error =

Nodo N_1	**Quant**
Classe=0	1
Classe=1	5

Gini =

Entropy =

Error =

Nodo N_1	**Quant**
Classe=0	3
Classe=1	3

Gini =

Entropy =

Error =

Medidas para Selecionar a Melhor Divisão

- Determine o Gini, Entropia e Erro dos nodos abaixo.

Nodo N_1	Quant
Classe=0	0
Classe=1	6

Gini = $1 - (0/6)^2 - (6/6)^2 = 0$

Entropy = $-(0/6)\log_2(0/6) - (6/6)\log_2(6/6) = 0$

Error = $1 - \max[0/6, 6/6] = 0$

Nodo N_1	Quant
Classe=0	1
Classe=1	5

Gini = $1 - (1/6)^2 - (5/6)^2 = 0.278$

Entropy = $-(1/6)\log_2(5/6) - (1/6)\log_2(5/6) = 0.650$

Error = $1 - \max[1/6, 5/6] = 0.167$

Nodo N_1	Quant
Classe=0	3
Classe=1	3

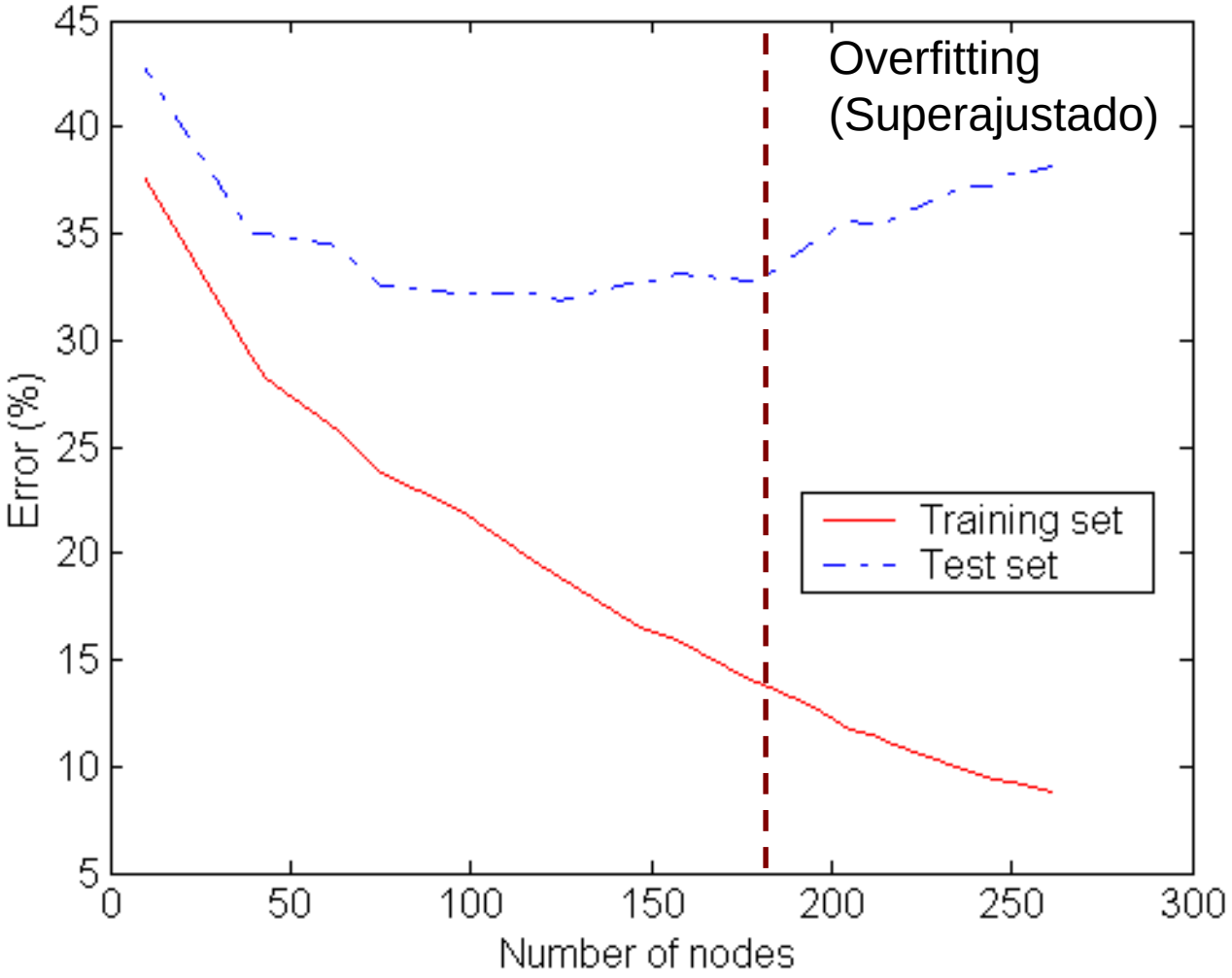
Gini = $1 - (3/6)^2 - (3/6)^2 = 0.5$

Entropy = $-(3/6)\log_2(3/6) - (3/6)\log_2(3/6) = 1$

Error = $1 - \max[3/6, 3/6] = 0.5$

Qual Nodo possui a menor impureza?

Underfitting versus Overfitting



- *Underfitting*: quando o modelo é muito simples, os erros tanto na base de treino quanto na de teste são expressivos.

Overfitting

- Resulta em árvores de decisão mais complexas do que o necessário.
- Erro de treino não fornece uma boa estimativa de quão bem a árvore irá comporta-se sobre novos registros.
- Necessita novas formas para estimar o erro.
- Manipulando Overfitting em Indução de Árvores de Decisão.

Árvores de Decisão

- **Vantagens**
  - Simples de visualizar e entender
  - Não necessita muita preparação para os dados (pre-processamento), tais como normalização
    - Apenas não aceita valores faltantes
  - O custo é logaritmo a quantidade de dados usados para treinar a árvore
  - Suporta dados numéricos e categóricos.
  - Modelo caixa branca: fácil interpretação
  - Possível reproduzir o modelo utilizando testes estatísticos

Árvores de Decisão

- **Desvantagens**
  - Indutores de árvores de decisão podem criar modelos muito complexos que não generalizam bem todos os dados (**overfitting**)
    - Para evitar esse problema deve ser definido um número mínimo de objetos nos nodos folhas ou um número máximo de profundidade da árvore
  - Pequenas variações no dataset podem gerar modelos instáveis
    - Esse problema pode ser atenuado usando árvore de decisão em conjuntos menores.
  - Podem criar modelos tendenciosos a classes dominantes (**bias**)
    - Recomenda-se equilibrar o conjunto de dados ante de ajustar a árvore de decisão.

Avaliando o Desempenho de um Classificador

Avaliação de Desempenho

- Matriz de Confusão:

		CLASSE PREVISTA	
CLASSE REAL		Classe=SIM	Classe=NAO
	Classe=SIM	a (TP)	b (FN)
	Classe=NAO	c (FP)	d (TN)

a: **TP** (true positive)
verdadeiro positivo

b: **FN** (false negative)
falso negativo

c: **FP** (false positive)
falso positivo

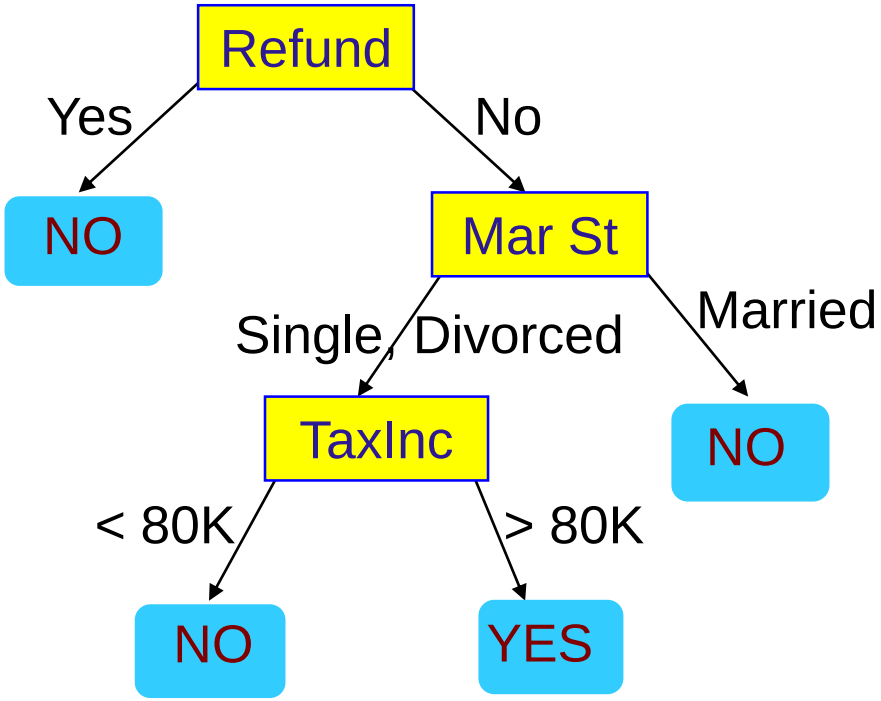
d: **TN** (true negative)
verdadeiro negativo

Métricas para Avaliação de Desempenho

categórico categórico contínuo 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



		CLASSE PREVISTA	
		Classe=S	Classe=N
CLASSE REAL	Classe=S	3	0
	Classe=N	0	7

Acurácia = 100%

Métricas para Avaliação de Desempenho

	CLASSE PREVISTA		
CLASSE REAL		Classe=S	Classe=N
	Classe=S	3 (TP)	0 (FN)
	Classe=N	4 (FP)	3 (TN)

- Acuracy: $\frac{TP+TN}{TP+FN+FP+TN}$ = 60%
  - Percentual de acertos.
- Recall (sensibilidade): $\frac{TP}{TP+FN}$ = 100%
  - Representa as instâncias que deveriam ser da classe **S** mas foram classificadas na classe **N**. Mais direcionado para a classe real.

Métricas para Avaliação de Desempenho

	CLASSE PREVISTA		
CLASSE REAL		Classe=S	Classe=N
	Classe=S	3 (TP)	0 (FN)
	Classe=N	4 (FP)	3 (TN)

- Precision (especificidade): $\frac{TP}{TP+FP}$ = 43%
  - Representa as instâncias que deveriam ser da classe **N** mas foram classificadas na classe **S**. Direcionado para a classe prevista
- F1-Score: $\frac{2\times(Recall\times Precision)}{Recall+Precision}$ = 60,14%
  - Equilíbrio entre Precision e Recall
  - Representa a distribuição de classe desigual

Métricas para Avaliação de Desempenho

- Acuracy:
  - Percentual de acertos.
  - $\frac{TP+TN}{TP+FN+FP+TN}$ = 60%

	CLASSE PREVISTA		
CLASSE REAL		Classe=S	Classe=N
	Classe=S	3 (TP)	0 (FN)
	Classe=N	4 (FP)	3 (TN)

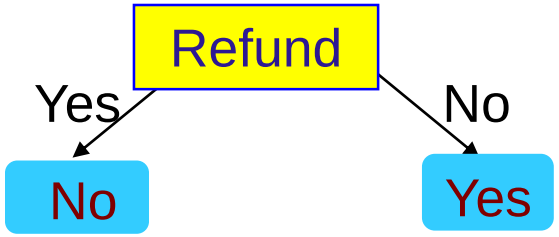
- Recall (sensibilidade):
  - $\frac{TP}{TP+FN}$ = 100%
- Precision (especificidade):
  - $\frac{TP}{TP+FP}$ = 43%
- F1-Score: $\frac{2\times(Recall\times Precision)}{Recall+Precision}$ = 60,14%

Métricas para Avaliação de Desempenho

categórico | categórico | contínuo | 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



		CLASSE PREVISTA	
		Classe=S	Classe=N
CLASSE REAL	Classe=S	3	0
	Classe=N	4	3

Acurácia = 60%

Sensibilidade (*Recall*) = 100%
(percentual de positivos pegos)

Especificidade (*Precision*) = 43%
(percentual de negativos pegos)

Exercício

- Utilize o mesmo dataset fornecido pela professora e execute o algoritmo utilizando o método holdout, reservando apenas 20% dos dados para teste.

- Gere a matriz de confusão para os datasets de treino e teste.
- Compute as métricas (precision, recall e f1-score) para os conjutos de treino e teste.
- ***Avalie*** e ***compare*** os resultados obtidos pelo dataset de treino e teste e identifique se houve *overfitting*, *underfitting* ou se o modelo induzido gerado é adequado para utilizar em dados não vistos.

- Caso os resultados não estejam bons, altere os parâmetros do algoritmo para tentar melhorar o desempenho do modelo.

Créditos

- Adaptação dos slides de Pang-Ning Tan
  - Michigan State University
  - http://www.cse.msu.edu/~ptan/
  - ptan@cse.msu.edu
- Adaptação dos slides de Eamon Keogh
  - University of California at Riverside
  - http://www.cs.ucr.edu/~eamonn/
  - eamonn@cs.ucr.edu
- Adaptação dos slides de Ricardo Campello e Eduardo Hruschka
  - Universidade de São Paulo (ICMC)
- Adaptação dos slides de Rodrigo Barros
  - Pontifícia Universidade Católica do Rio Grande do Sul (PPGCC)

Referências

- Breiman, L., Freidman, J., Olshen, R. e Stone, C. (1984). *Classification and Regression Trees.* Wadsworth International Group., USA.
- Faceli, K.; Lorena, A.C.; Gama, J.; de Carvalho, A.C.P.L.F. *Inteligência Artificial: Uma abordagem de aprendizado de máquina.* LTC, Rio de Janeiro, 2011.
- Quilan, R. (1979). *Discovering rules by induction from large collections of examples.* In: Michie, D. (Ed.) Expert Systems in the Microelectronic Age, p. 168-201. Edinburgh University Press.
- Quilan, J.R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Mateo, CA, USA.
- TAN, P-N; STEINBACH, M.; KUMAR,V. *Introduction to Data Mining.* Pearson, 2006.