

# APRENDIZADO DE MÁQUINAS

Aprendizado para Classificação  
Árvores de Decisão



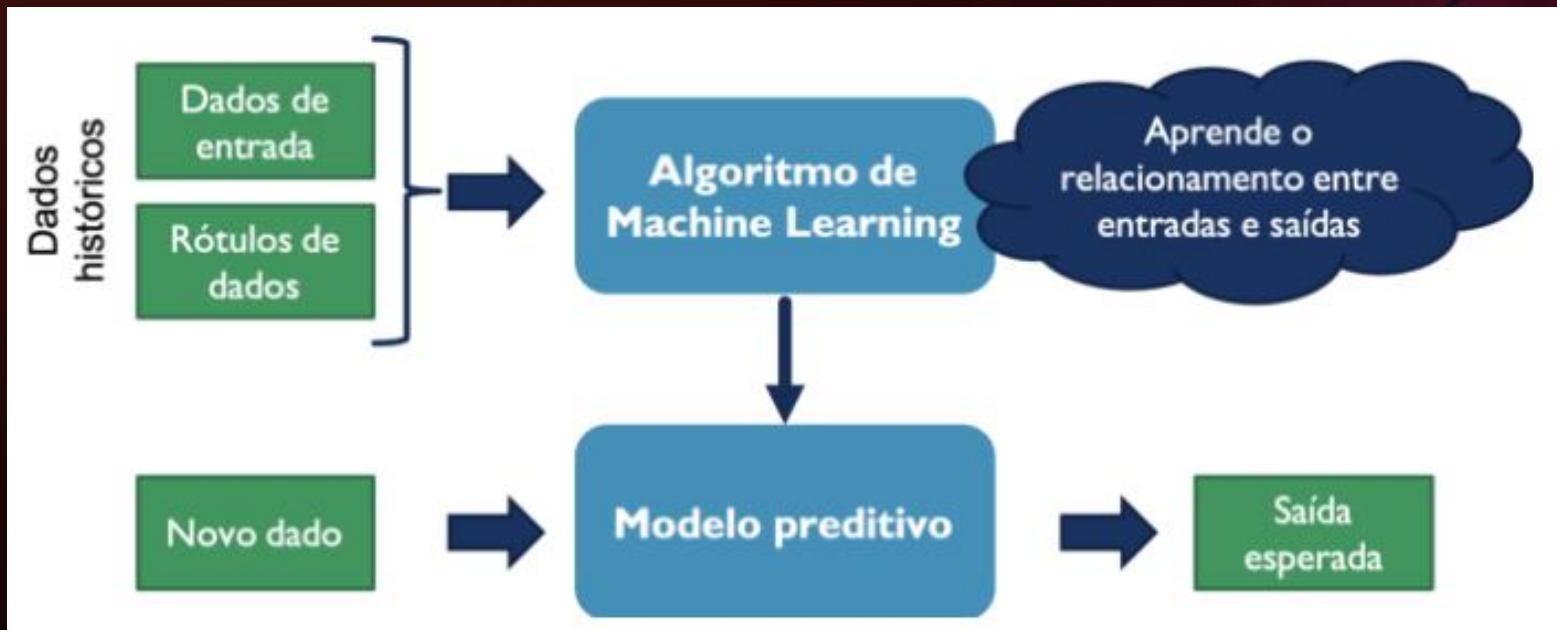
# TÓPICOS

- 1. Métodos Simbólicos**
- 2. Árvores de Decisão**
- 3. Regras de Decisão**
- 4. Regras de Divisão para Classificação**
- 5. Regras de Divisão para Regressão**
- 6. Valores Desconhecidos**
- 7. Estratégias de Poda**
- 8. Vantagens x Desvantagens**
- 9. Regras de Decisão**

# MÉTODOS SIMBÓLICOS

Neste paradigma, um conceito é representado em uma estrutura simbólica e o aprendizado é realizado através da apresentação de exemplos e contraexemplos deste conceito.

Essas estruturas possibilitam uma interpretação mais direta por seres humanos. A vantagem principal desse tipo de método é uma maior comprehensibilidade do processo decisório.

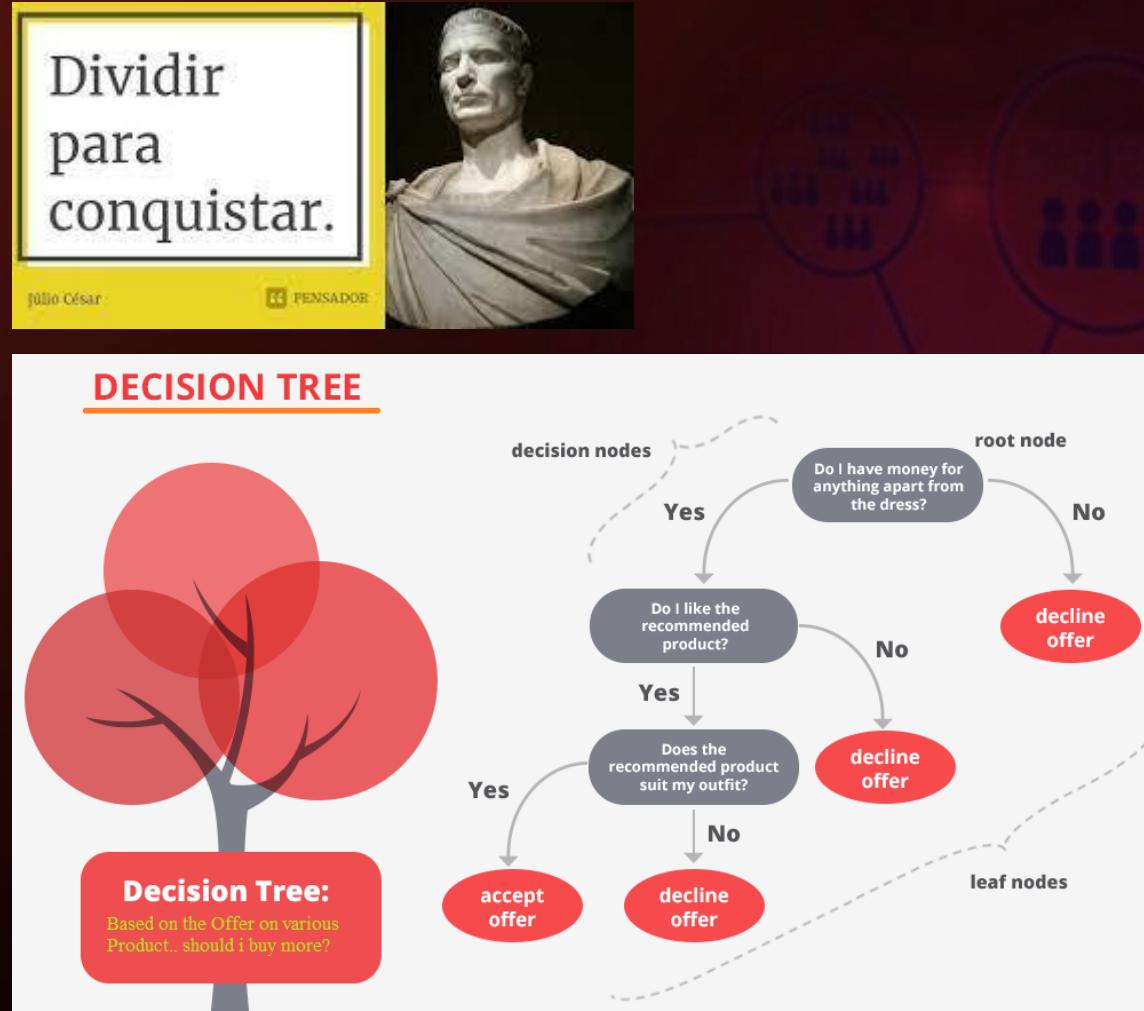


Fonte: Medium.com

# ÁRVORES DE DECISÃO E REGRESSÃO

Uma árvore de decisão usa a estratégia dividir para conquistar, de modo a resolver um problema de decisão.

Formalmente, uma árvore de decisão é um grafo direcionado acíclico em que cada nó ou é um nó de divisão, com dois ou mais sucessores, ou é um nó folha.



Fonte: Data Science Foundation

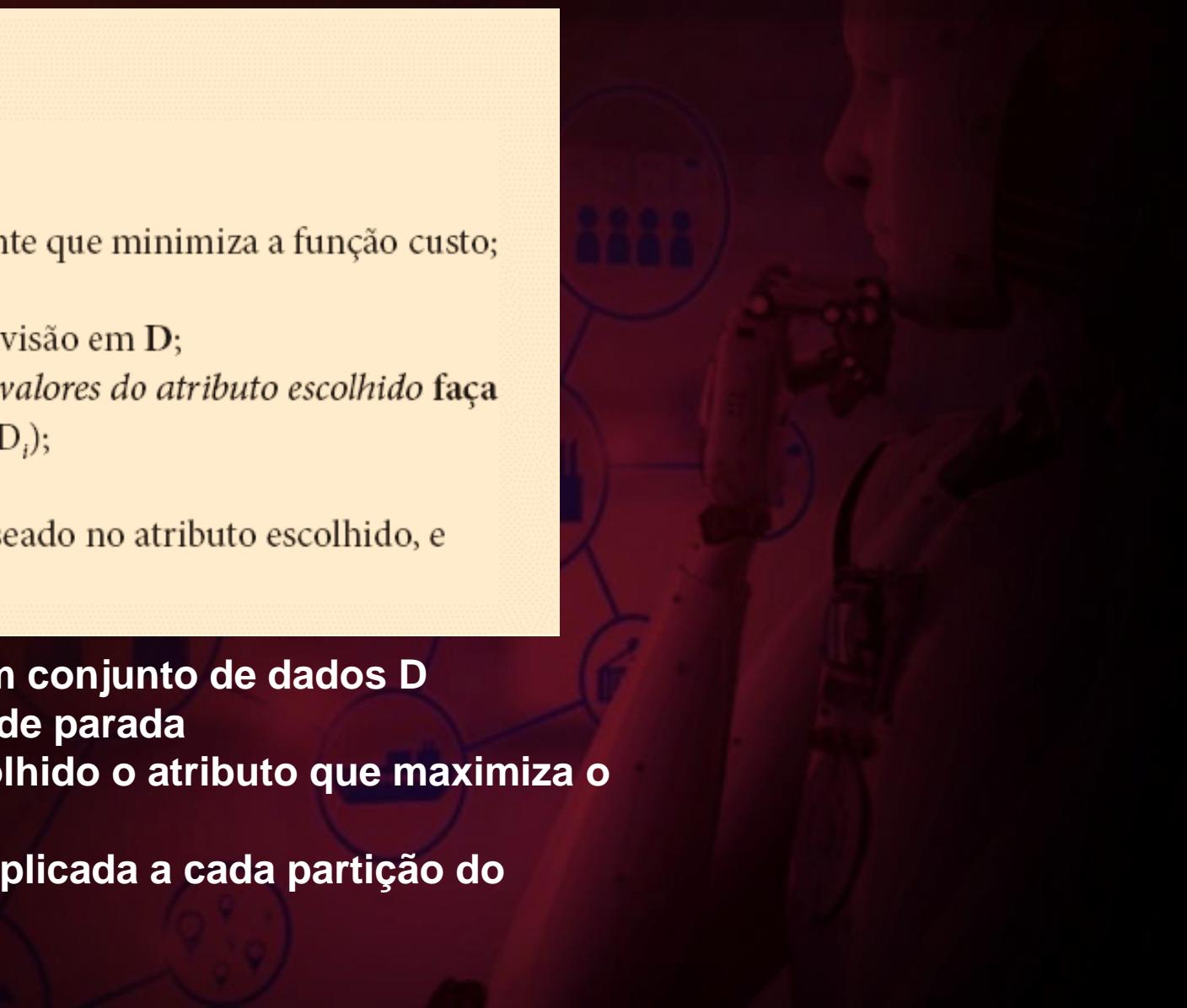
# INDUÇÃO DE ÁRVORES DE DECISÃO E REGRESSÃO

**Entrada:** Um conjunto de treinamento  $D = \{(\mathbf{x}_i, y_i), i = 1, \dots, n\}$

**Saída:** Árvore de Decisão

```
1 /* Função GeraÁrvore(D);*/
2 se critério de parada(D) = Verdadeiro então
3     Retorna: um nó folha rotulado com a constante que minimiza a função custo;
4 fim
5 Escolha o atributo que maximiza o critério de divisão em D;
6 para cada partição dos exemplos  $D_i$ , baseada nos valores do atributo escolhido faça
7     Induz uma subárvore  $\text{Árvore}_i = \text{GeraÁrvore}(D_i)$ ;
8 fim
9 Retorna: Árvore contendo um nó de decisão baseado no atributo escolhido, e
descendentes  $\text{Árvore}_i$ ;
```

- A entrada para a função **GeraÁrvore** é um **conjunto de dados D**
- No passo 3, o algoritmo avalia o critério de parada
- Se mais divisões são necessárias é escolhido o atributo que maximiza o critério de Divisão (passo 5)
- No passo 7, a função é recursivamente aplicada a cada partição do conjunto de dados D



# REGRAS DE DIVISÃO POR CLASSIFICAÇÃO

Deve-se considerar o uso de árvores de decisão em situações onde:

- As instâncias são descritas por pares atributo-valor;
- A função objeto (alvo) é de valor discreto;
- Os exemplos de treino poderão ter erro (*noise*);
- Faltam valores nos atributos;

Exemplos:

- Diagnósticos médicos;
- Análises de risco de crédito;
- Classificação de objetos para um manipulador de *robot* (Tan1993).

# REGRAS DE DIVISÃO POR CLASSIFICAÇÃO

## Algoritmo ID3

O algoritmo ID3 (*inductive decision tree*) é dos mais utilizados para a construção de árvores de decisão. Este algoritmo segue os seguintes passos:

- Começar com todos os exemplos de treino;
- Escolher o teste (atributo) que melhor divide os exemplos, ou seja, agrupar exemplos da mesma classe ou exemplos semelhantes;
- Para o atributo escolhido, criar um nó filho para cada valor possível do atributo;
- Transportar os exemplos para cada filho tendo em conta o valor do filho;
- Repetir o procedimento para cada filho não "puro". Um filho é puro quando cada atributo X tem o mesmo valor em todos os exemplos.

Coloca-se então, uma pergunta muito importante:

Como saber qual o melhor atributo a escolher?

Para lidar com esta escolha são introduzidos dois novos conceitos, a Entropia e o Ganho.

# REGRAS DE DIVISÃO POR CLASSIFICAÇÃO

## Entropia

A entropia de um conjunto pode ser definida como sendo o grau de pureza desse conjunto. Este conceito, emprestado pela Teoria da Informação define a medida de "falta de informação", mais precisamente o número de bits necessários, em média, para representar a informação em falta usando codificação ótima.

Dada uma coleção  $S$  contendo exemplos positivos (+) e negativos (-) de algum conceito alvo, a entropia de  $S$  relativa a essa classificação boolena é:

$$\text{Entropia } (S) \equiv - p_+ \log_2 p_+ - p_- \log_2 p_-$$

Onde:

$P_+$  é a proporção de exemplos positivos em  $S$

$P_-$  é a proporção de exemplos negativos em  $S$

# REGRAS DE DIVISÃO POR CLASSIFICAÇÃO

Exemplo:

Exemplos de Treino

Dia	Aspecto	Temp.	Humidade	Vento	Jogar Tênis
D1	Sol	Quente	Elevada	Fraçoo	Não
D2	Sol	Quente	Elevada	Forte	Não
D3	Nuvens	Quente	Elevada	Fraçoo	Sim
D4	Chuva	Ameno	Elevada	Fraçoo	Sim
D5	Chuva	Fresco	Normal	Fraçoo	Sim
D6	Chuva	Fresco	Normal	Forte	Não
D7	Nuvens	Fresco	Normal	Fraçoo	Sim
D8	Sol	Ameno	Elevada	Fraçoo	Não
D9	Sol	Fresco	Normal	Fraçoo	Sim
D10	Chuva	Ameno	Normal	Forte	Sim
D11	Sol	Ameno	Normal	Forte	Sim
D12	Nuvens	Ameno	Elevada	Forte	Sim
D13	Nuvens	Quente	Normal	Fraçoo	Sim
D14	Chuva	Ameno	Elevada	Forte	Não

Entropia:

$$S = [9+, 5-]$$

$$E(\text{Entropia}) = -\left(\frac{9}{14}\right)\log_2\left(\frac{9}{14}\right) - \left(\frac{5}{14}\right)\log_2\left(\frac{5}{14}\right) = 0,940$$

# REGRAS DE DIVISÃO POR CLASSIFICAÇÃO

**Ganho de Informação:** redução esperada no valor da Entropia, devido à ordenação no conjunto de treino segundo os valores do atributo A.

Exemplo: 
$$Ganho(S, A) = Entropia(S) - \sum_{v \in valores(A)} \frac{|S_v|}{|S|} \cdot Entropia(S_v)$$

Para responder a pergunta: Qual o melhor atributo para iniciar a árvore ?

Resposta: utiliza-se o ganho.

No exemplo dado:

$$Ganho(S, Perspectiva) = 0,246$$

$$Ganho(S, Umidade) = 0,151$$

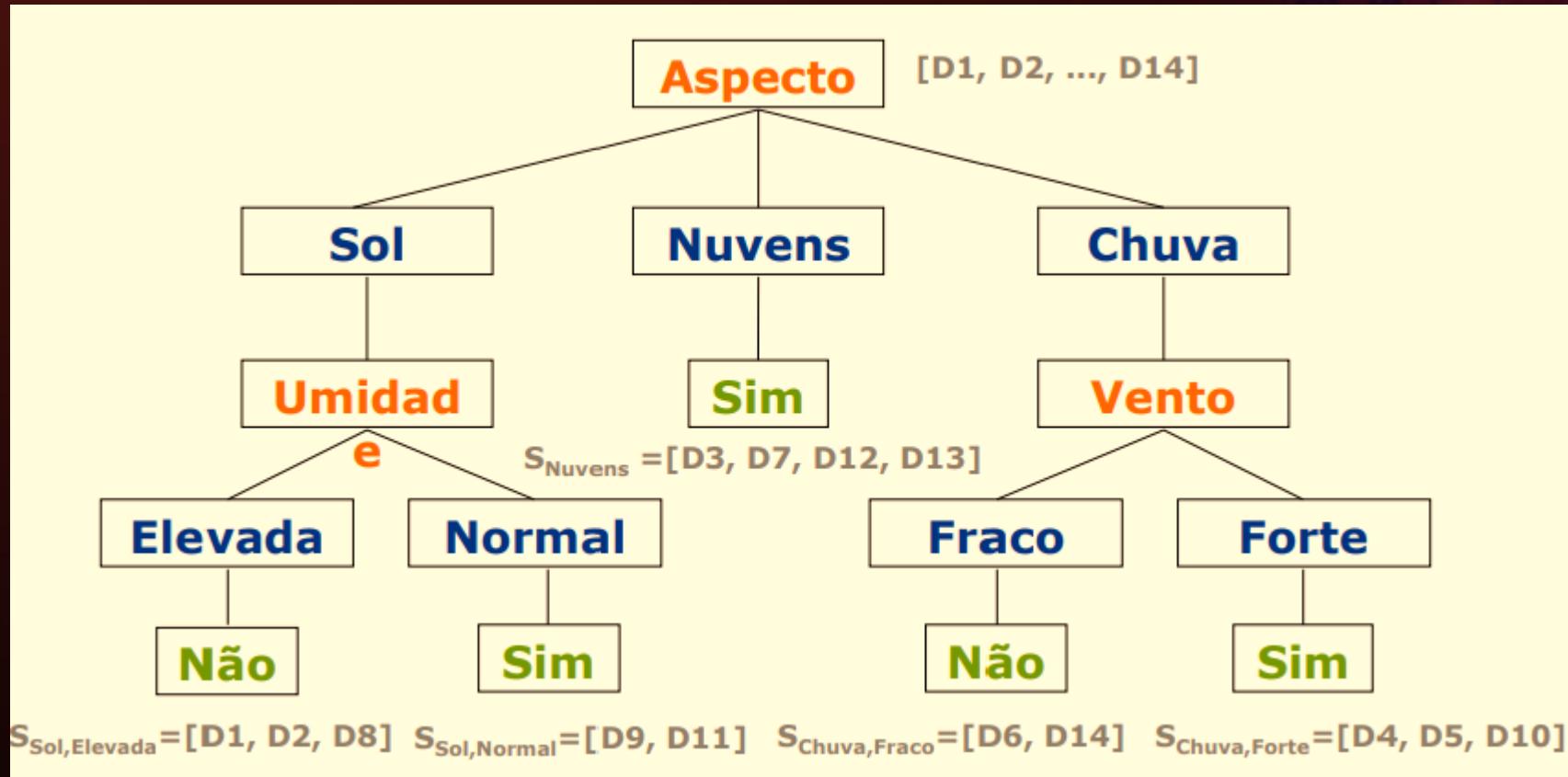
$$Ganho(S, Vento) = 0,048$$

$$Ganho(S, Temperatura) = 0,029$$



# REGRAS DE DIVISÃO POR CLASSIFICAÇÃO

Quando em todos os nós a entropia for nula, o algoritmo para e obtém-se a seguinte árvore de decisão:



# REGRAS DE DIVISÃO POR REGRESSÃO

As árvores de regressão são usadas quando a variável dependente é contínua, diferentemente das árvores de classificação que são usadas quando a variável dependente é categórica.

No caso da árvore de regressão, o valor obtido pelos nós de término nos dados de treinamento é o valor médio das suas observações. Assim, a uma nova observação de dados atribui-se o valor médio correspondente.

Ambas as árvores dividem o espaço preditor (variáveis independentes) em regiões distintas e não sobrepostas.

Para a realização das partições em uma árvore de regressão utiliza-se a métrica de redução da variância, definida pelas fórmulas abaixo:

$$sd(\mathbf{D}, \mathbf{y}) = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2}$$

$$SDR(h_A) = sd(\mathbf{D}, \mathbf{y}) - \frac{n_L}{n} \times sd(\mathbf{D}_L, \mathbf{y}) - \frac{n_R}{n} \times sd(\mathbf{D}_R, \mathbf{y})$$

# VALORES DESCONHECIDOS

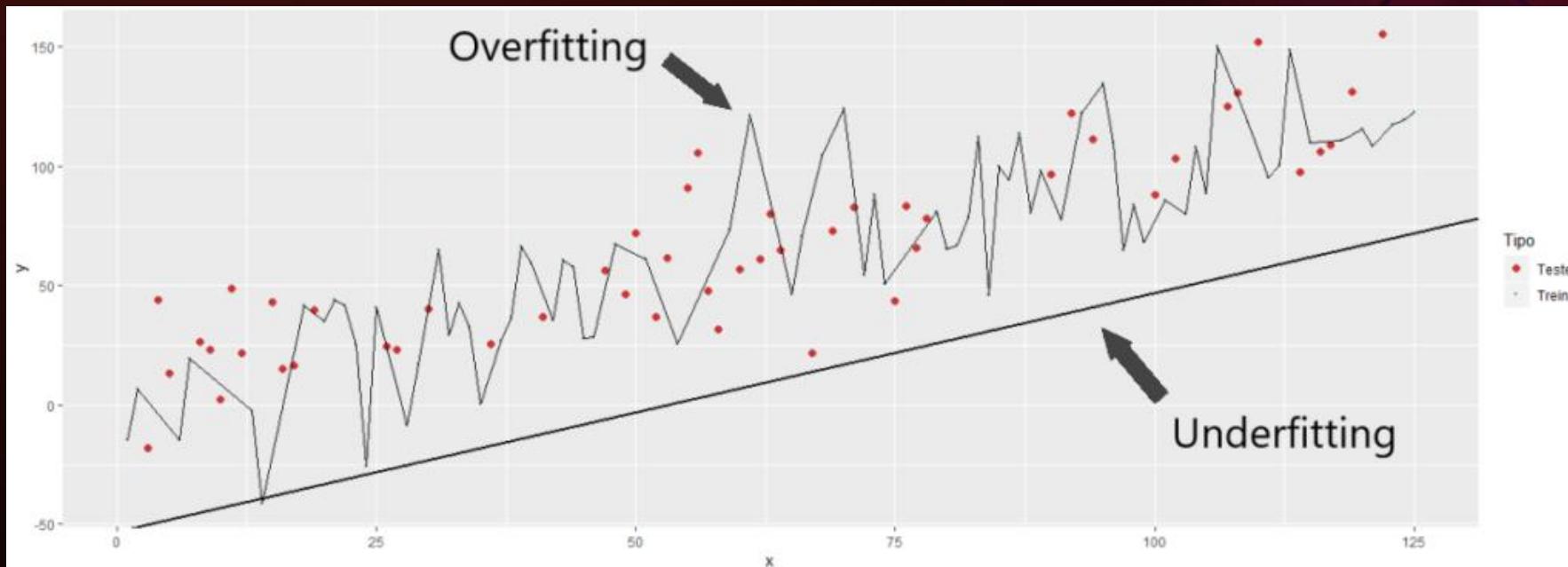
**Se o valor do atributo testado não é conhecido podemos utilizar as seguintes estratégias:**

- **Trocar o valor desconhecido pelo valor mais comum para o atributo encontrado no conjunto de treinamento**
- **Considerar o valor desconhecido como outro valor possível**
- **Associa-se uma probabilidade a cada um dos possíveis valores do atributo (Algoritmo C4.5)**
- **Estratégia da Divisão Substituta (Algoritmo CART)**

# ESTRATÉGIAS DE PODA

**Overfitting:** Um cenário de *overfitting* ocorre quando, nos dados de treino, o modelo tem um desempenho excelente, porém, quando utilizamos os dados de teste o resultado é ruim.

**Underfitting:** No cenário de *underfitting* o desempenho do modelo já é ruim no próprio treinamento. O modelo não consegue encontrar relações entre as variáveis e o teste nem precisa acontecer. Este modelo já pode ser descartado, pois não terá utilidade.



# ESTRATÉGIAS DE PODA

Para melhorar o modelo, utilizam-se métodos de poda (pruning) na árvore, cujo objetivo é melhorar a taxa de acerto do modelo para novas amostras que não foram utilizadas no treinamento.

**Pré-poda:** realizada durante a construção da árvore. Em um certo momento, se o ganho de informação for menor que um valor preestabelecido, então, esse nó vira folha.

**Pós-poda:** realizada após a construção da árvore. Para cada nó interno da árvore, é calculada a taxa de erro caso esse nó vire folha (e tudo abaixo dele seja eliminado). Em seguida, é calculada a taxa de erro caso não haja a poda. Se a diferença entre essas duas taxas de erro for menor que um valor preestabelecido, a árvore é podada; caso contrário, não ocorre a poda.

ESTRATÉGIAS



# ASPECTOS POSITIVOS X NEGATIVOS

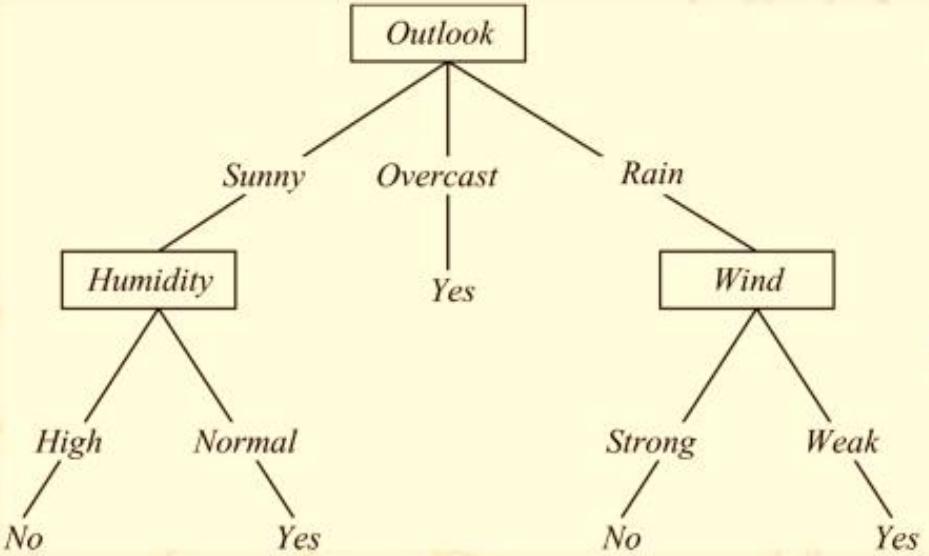
## Vantagens:

1. Flexibilidade
2. Robustez
3. Seleção de atributos
4. Interpretabilidade
5. Eficiência

## Desvantagens:

1. Replicação
2. Valores ausentes
3. Atributos contínuos
4. Instabilidade

# REGRAS DE DECISÃO



IF (*Outlook* = *Sunny*)  $\wedge$  (*Humidity* = *High*)  
THEN *PlayTennis* = *No*

IF (*Outlook* = *Sunny*)  $\wedge$  (*Humidity* = *Normal*)  
THEN *PlayTennis* = *Yes*



# REGRAS DE DECISÃO

**Vantagens de converter uma árvore de decisão em regras antes da poda:**

- Permite distinguir entre os diferentes contextos onde os nós de decisão são utilizados
- Remove a distinção entre atributos de testes que ocorrem próximos da raiz da árvore e aqueles que ocorrem próximos das folhas
- Melhora a leitura humana
- Regras são geralmente mais fáceis para pessoas entenderem

# RESUMO

**Aprendizagem de árvores de decisão fornece um método prático para a aprendizagem de conceito e para a aprendizagem de outras funções de valor discreto.**

**A família de algoritmos ID3 infere árvores de decisão expandindo-as a partir da raiz e descendo, selecionando o próximo melhor atributo para cada novo ramo de decisão adicionado à árvore.**

**Simplicidade para compreensão e interpretação.**

**Os dados não necessitam de pré-processamento.**

**Lidam tanto com dados numéricos quanto categóricos.**

# RESUMO

**Emprega um modelo “caixa branca”.**

**Possibilidade de validar um modelo através de estatísticas.**

**Robustez.**

**Bom desempenho em grandes conjuntos de dados em um tempo curto.**

# ALGORITMOS DE ÁRVORE DE DECISÃO

Onde encontrar ?

<http://aima.cs.berkeley.edu/>

<https://github.com/>

<https://scikit-learn.org/stable/modules/tree.html>