



Formação Inteligência Artificial

Introdução à Inteligência Artificial





Data Science Academy raphaelbsfontenelle@gmail.com 615c1fdde32fc361b30c9ec2

Aprendizagem

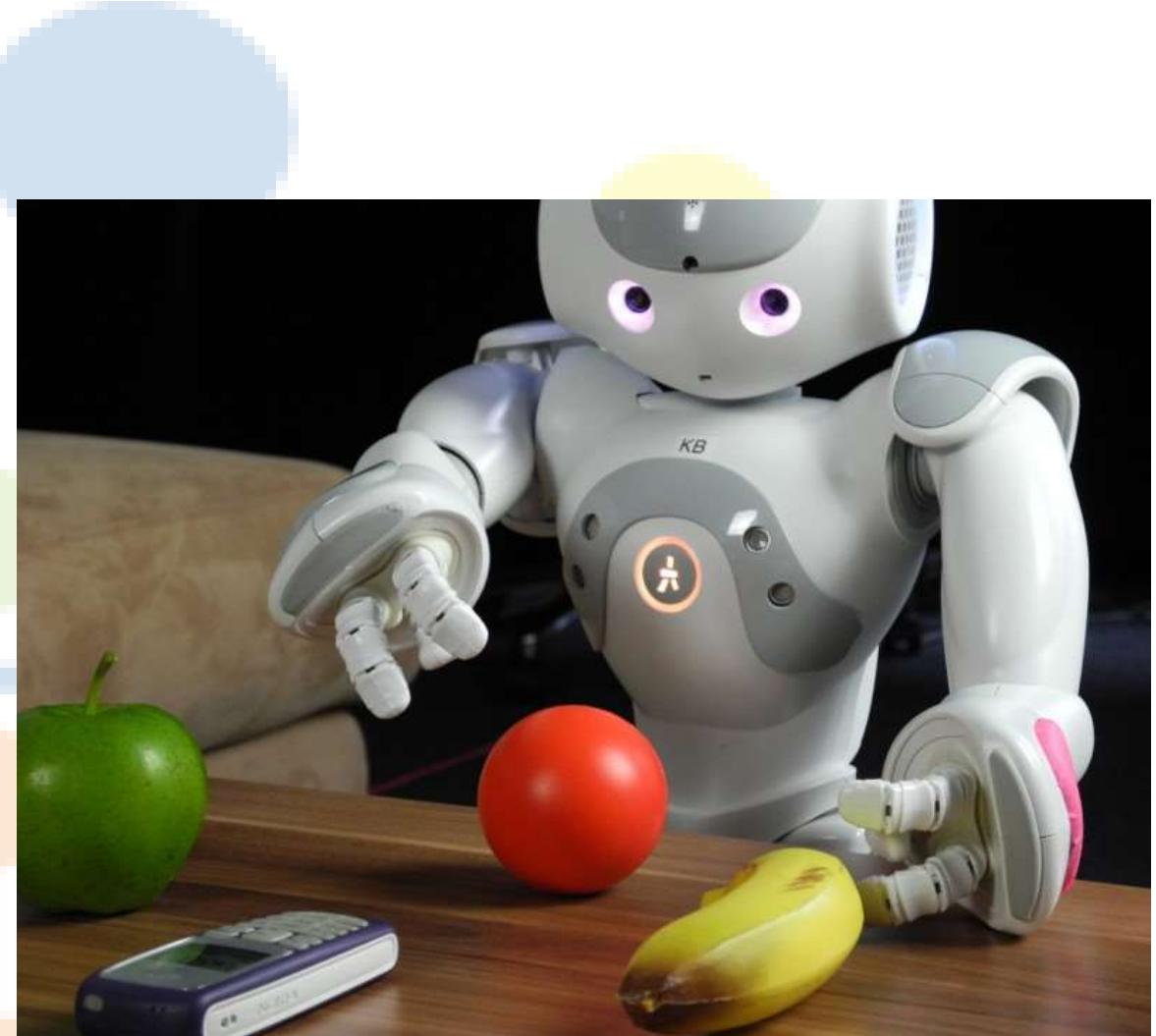


Data Science Academy

Introdução



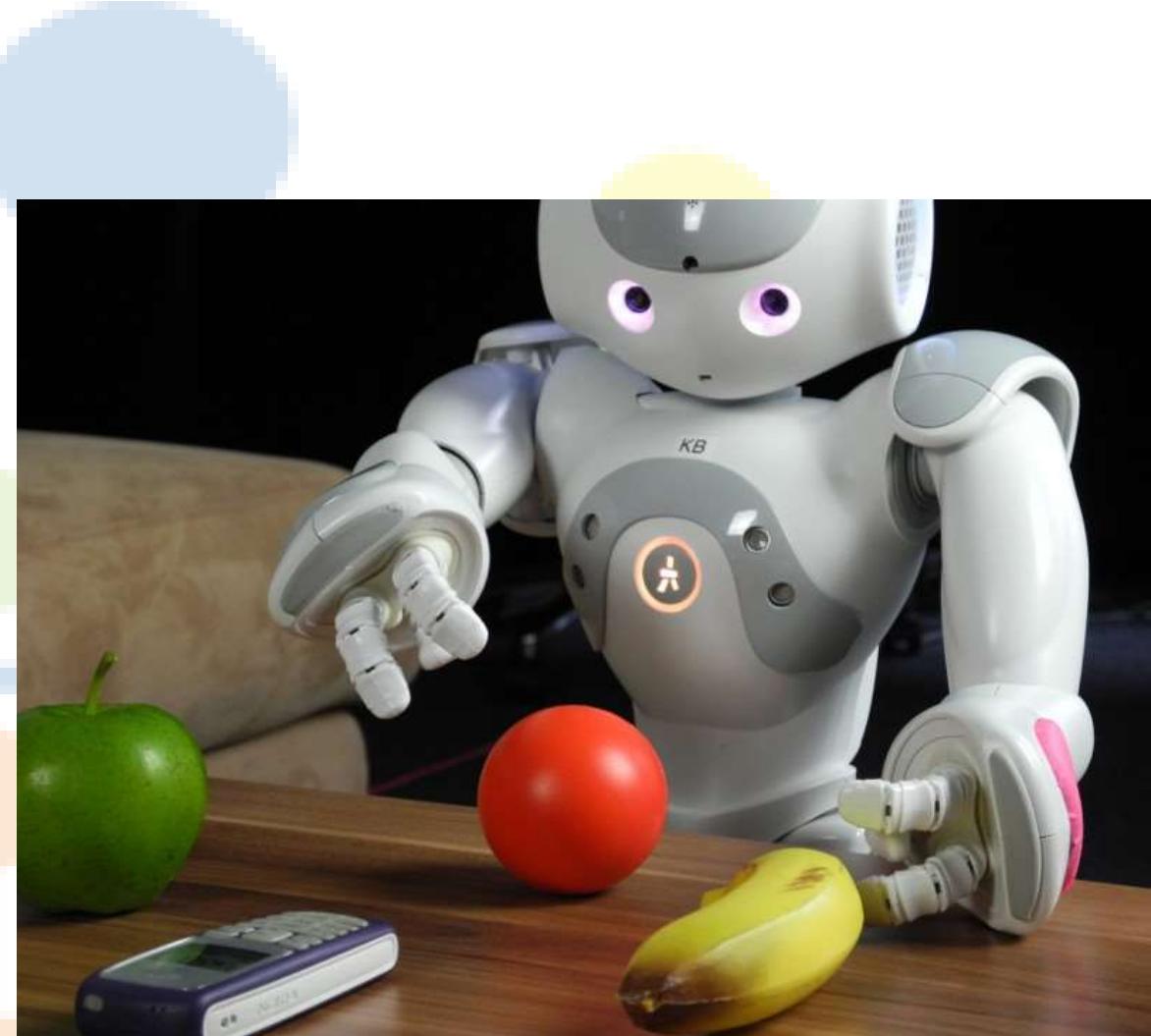
A partir de uma coleção de pares de entrada e saída, aprender uma função que prevê a saída para novas entradas.



Introdução



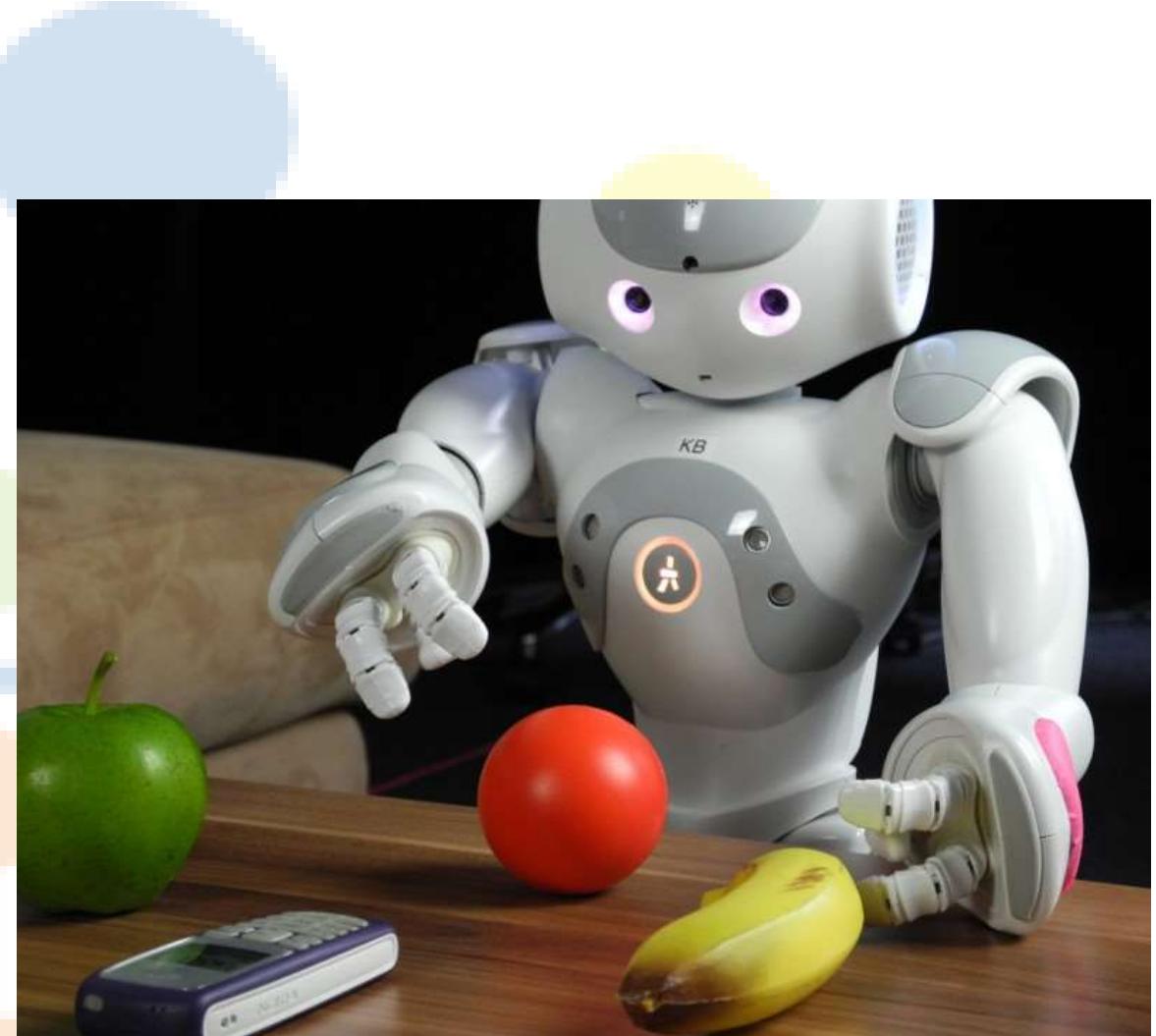
Por que queremos que um agente aprenda?



Introdução



Este capítulo dará a você uma visão geral das diferentes formas de aprendizagem.





Data Science
Academy

Data Science Academy raphaelbsfontenelle@gmail.com 615c1fdde32fc361b30c9ec2



Data Science Academy



Data Science Academy

Introdução



Data Science
Academy

Data Science Academy raphaelbsfontenelle@gmail.com 615c1fdde32fc361b30c9ec2



Processo de
Aprendizagem



Data Science Academy



Data Science Academy

Introdução



The diagram illustrates a mechanical system consisting of two horizontal disks and a vertical spring-mass system. The top disk rotates with angular velocity ω around its center. A point on the circumference of the top disk has position vector \vec{r} and velocity $\vec{v} = \vec{\omega} \times \vec{r}$. The angle between the vertical axis and the radius \vec{r} is δ . The acceleration \vec{a} is given by $\vec{a} = \vec{v} = \frac{d\vec{\omega}}{dt} \times \vec{r} + \vec{\omega} \times \frac{d\vec{r}}{dt} = \vec{\alpha} \times \vec{r} + \vec{\omega} \times \vec{v} = \vec{a}_t + \vec{a}_n$. The bottom disk is connected to the top one by a horizontal rod. The center of the bottom disk has position vector \vec{r}_{1L} and velocity $\vec{v}_{1L} = \vec{T}_{P_2}(\varphi_2) \vec{v}_{2L}$. The angle between the vertical axis and the radius \vec{r}_{1L} is φ . The vertical spring connects the center of the bottom disk to a fixed wall. The mass of the bottom disk is m_2 , and its center has coordinates (x_2, y_2) . The mass of the spring is m_3 , and its center has coordinates (x_3, y_3) . The spring constant is k , and the damping coefficient is b . The displacement of the spring center from its equilibrium position is \bar{y}_3 . The equations of motion are:

$$m_2 \ddot{y}_2 = -F_b - F_p + R_y$$
$$m_3 \ddot{y}_3 = -R_y$$
$$\vec{v} = \vec{\omega} \times \vec{r} \quad v = \omega r \sin \delta = \omega \delta$$
$$\vec{a} = \vec{v} = \frac{d\vec{\omega}}{dt} \times \vec{r} + \vec{\omega} \times \frac{d\vec{r}}{dt} = \vec{\alpha} \times \vec{r} + \vec{\omega} \times \vec{v} = \vec{a}_t + \vec{a}_n$$
$$\vec{a} = \vec{\alpha} \times \vec{r} + \vec{\omega} \times \vec{v} = \vec{a}_t + \vec{a}_n$$
$$m_1[x_1, y_1] + \ddot{y}(l_1 + l_2 \sin^2 \varphi) + \ddot{\varphi} \sin \varphi \cos \varphi l_2 + g \sin \varphi = 0$$
$$\vec{r}_{1L} = \vec{T}_{P_2}(\varphi_2) \vec{v}_{2L}$$
$$\begin{bmatrix} x_{1L} \\ y_{1L} \end{bmatrix} = \begin{bmatrix} \cos \varphi_2 & -\sin \varphi_2 & 0 & 0 \\ \sin \varphi_2 & \cos \varphi_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{2L} \\ y_{2L} \\ z_{2L} \\ 1 \end{bmatrix}$$
$$\omega_{14} = \frac{r_2}{r_1 + r_2} \omega_{12}$$
$$\omega_{43} = \frac{r_1 r_2}{r_3(r_1 + r_2)} \omega_{12}$$
$$P_{24} = \frac{\omega_{14}}{\omega_{12}} = \frac{r_2}{r_1 + r_2}$$

Um algoritmo constrói suas capacidades cognitivas através da criação de uma formulação matemática que inclui todas as características dadas sobre um determinado fenômeno.

Introdução

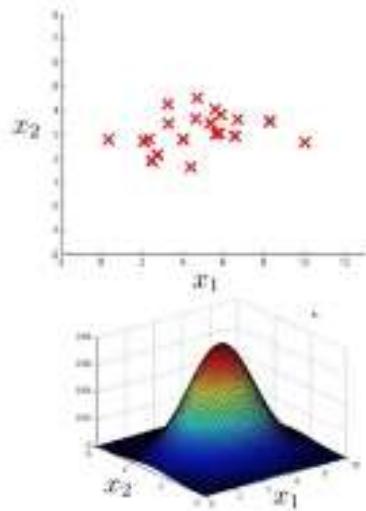


The diagram illustrates a mechanical system consisting of two horizontal disks and a vertical spring-mass system. The top disk rotates with angular velocity ω around its center. A point on the circumference of the top disk has position vector \vec{r} , velocity $\vec{v} = \vec{\omega} \times \vec{r}$, and acceleration $\vec{a} = \frac{d\vec{v}}{dt} = \vec{\omega} \times \vec{v} + \vec{\omega} \times \frac{d\vec{r}}{dt} = \vec{\alpha} + \vec{\omega} \times \vec{v}$. The bottom disk is connected to the top one by a horizontal rod. The vertical spring connects the center of the bottom disk to a fixed wall. The mass of the bottom disk is m_2 , and the spring constant is k . The displacement of the bottom disk from its equilibrium position is denoted by y . The system is subject to forces F_b and F_p , and reaction forces R_x and R_y .

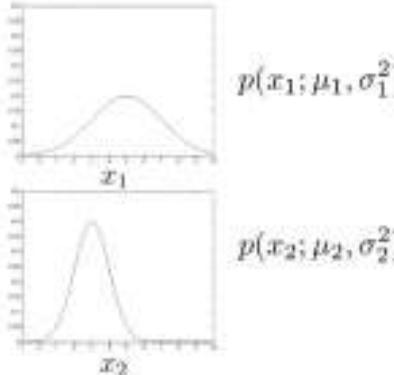
$$\vec{v} = \vec{\omega} \times \vec{r} \quad v = \omega r \sin \delta = \omega \delta$$
$$\vec{a} = \frac{d\vec{v}}{dt} = \vec{\omega} \times \vec{v} + \vec{\omega} \times \frac{d\vec{r}}{dt} = \vec{\alpha} + \vec{\omega} \times \vec{v} = \vec{\alpha}_t + \vec{\alpha}_n$$
$$m_2 \ddot{y}_2 = -F_b - F_p + R_y$$
$$m_3 \ddot{y}_3 = -R_y$$
$$\omega_{14} = \frac{r_2}{r_1 + r_2} \omega_{12}$$
$$\omega_{43} = \frac{r_1 r_2}{r_3(r_1 + r_2)} \omega_{12}$$
$$\rho_{24} = \frac{\omega_{14}}{\omega_{12}} = \frac{r_2}{r_1 + r_2}$$

Função alvo – $f(x)$

Introdução



$$\begin{aligned}\mu_1 &= 5, \sigma_1 = 2 \\ \mu_2 &= 3, \sigma_2 = 1\end{aligned}$$

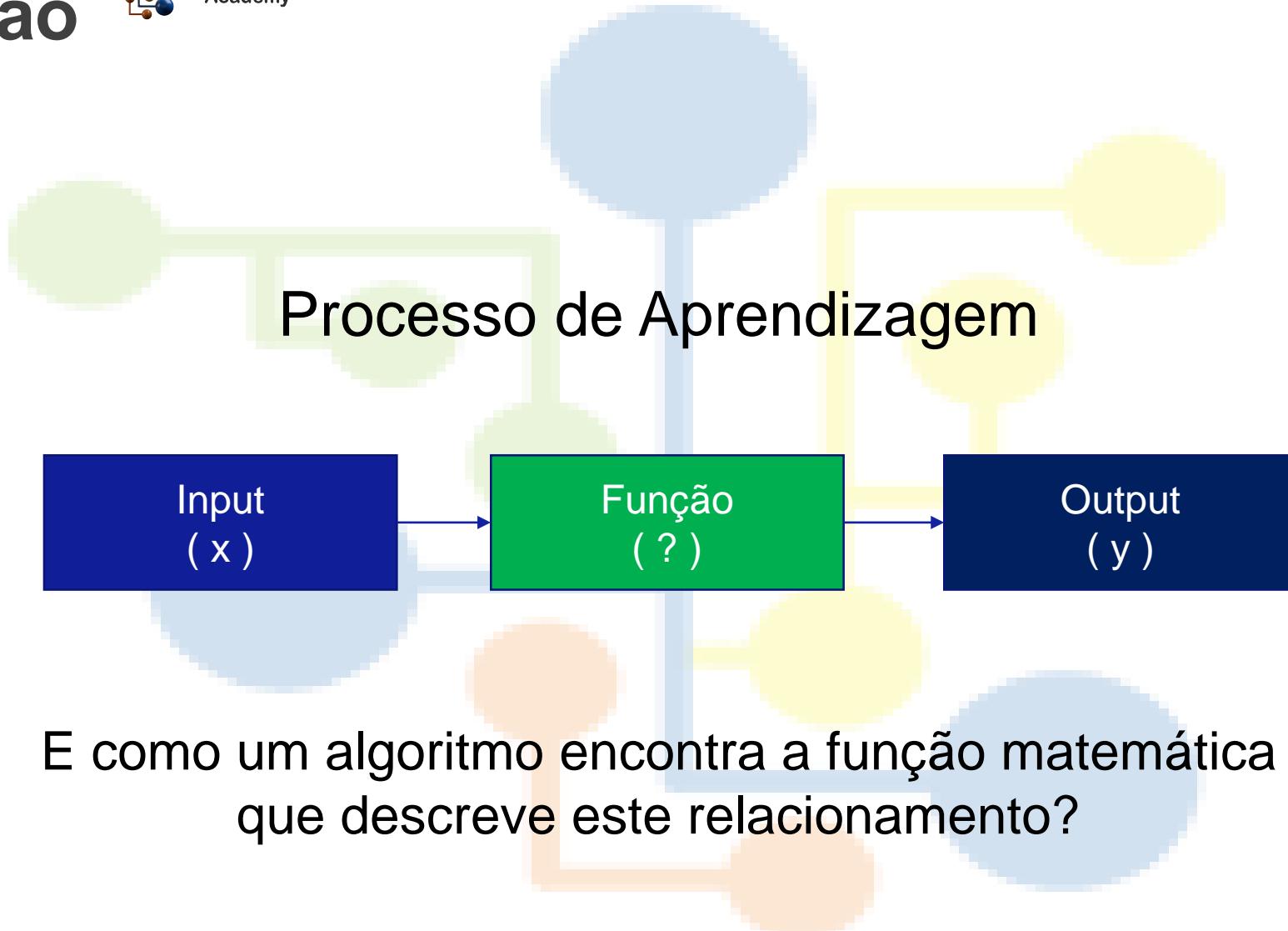


Processo de Aprendizagem

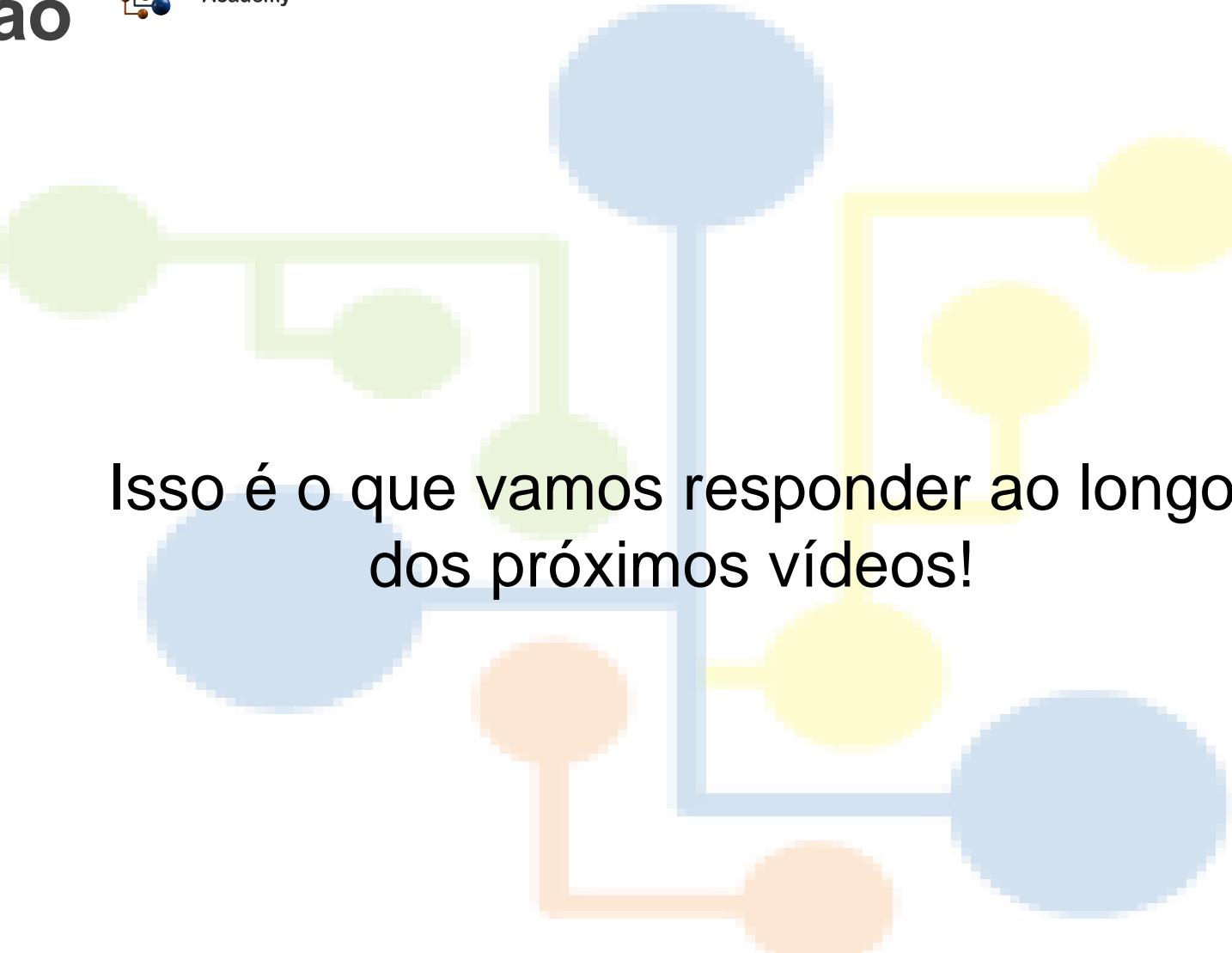
Função alvo – $f(x)$



Introdução



Introdução



Isso é o que vamos responder ao longo
dos próximos vídeos!



Componentes a Serem Aprendidos





⚙ Que componente deve ser melhorado

🛠 O conhecimento prévio que o agente já tem

🏆 Que representação é usada para os dados e para os componentes

🏷 Que feedback está disponível para aprendizagem



Componentes a Serem Aprendidos

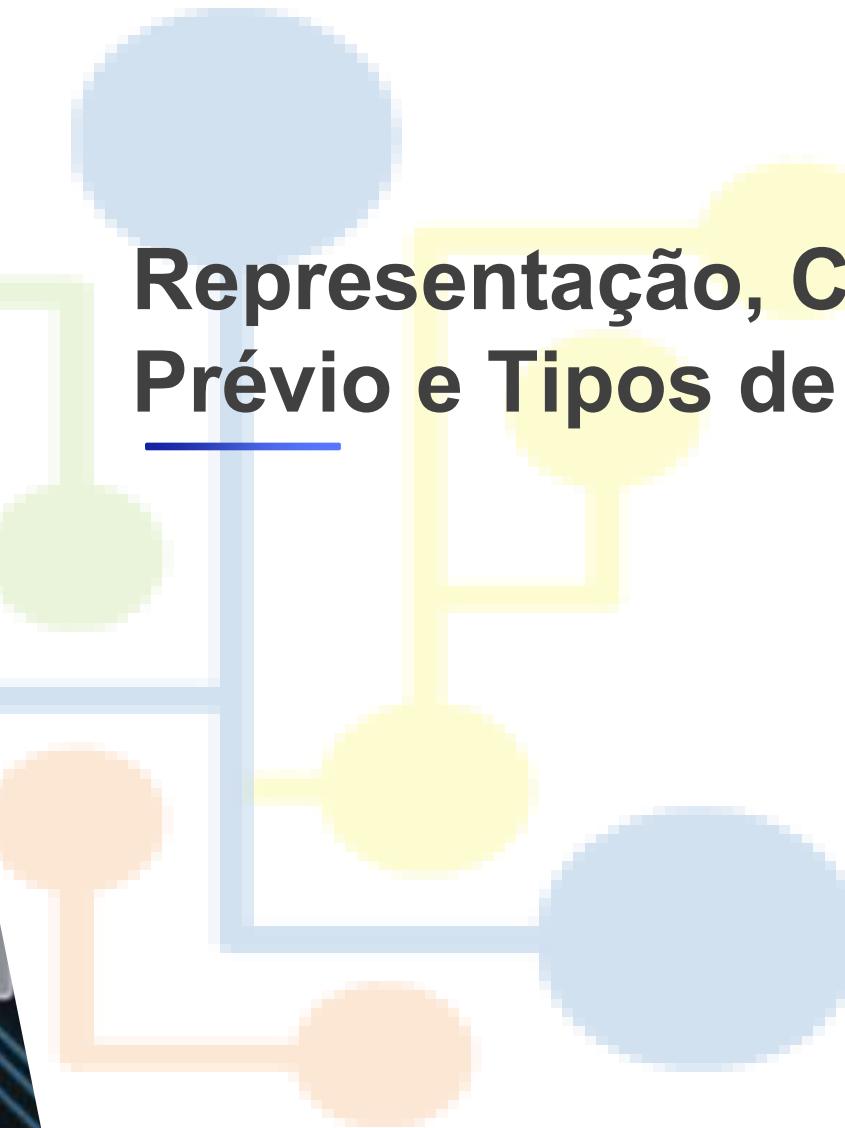


Componentes a Serem Aprendidos

1. Um mapeamento direto de condições no estado atual para ações.
2. Um meio para deduzir propriedades relevantes do mundo a partir da sequência de percepções.
3. Informações sobre o modo como o mundo evolui e sobre os resultados de ações possíveis que o agente pode executar.
4. Informações de utilidade indicando a desejabilidade de estados do mundo.
5. Informações de valores de ações indicando a desejabilidade de ações.
6. Metas que descrevem classes de estados cuja realização maximiza a utilidade do agente.

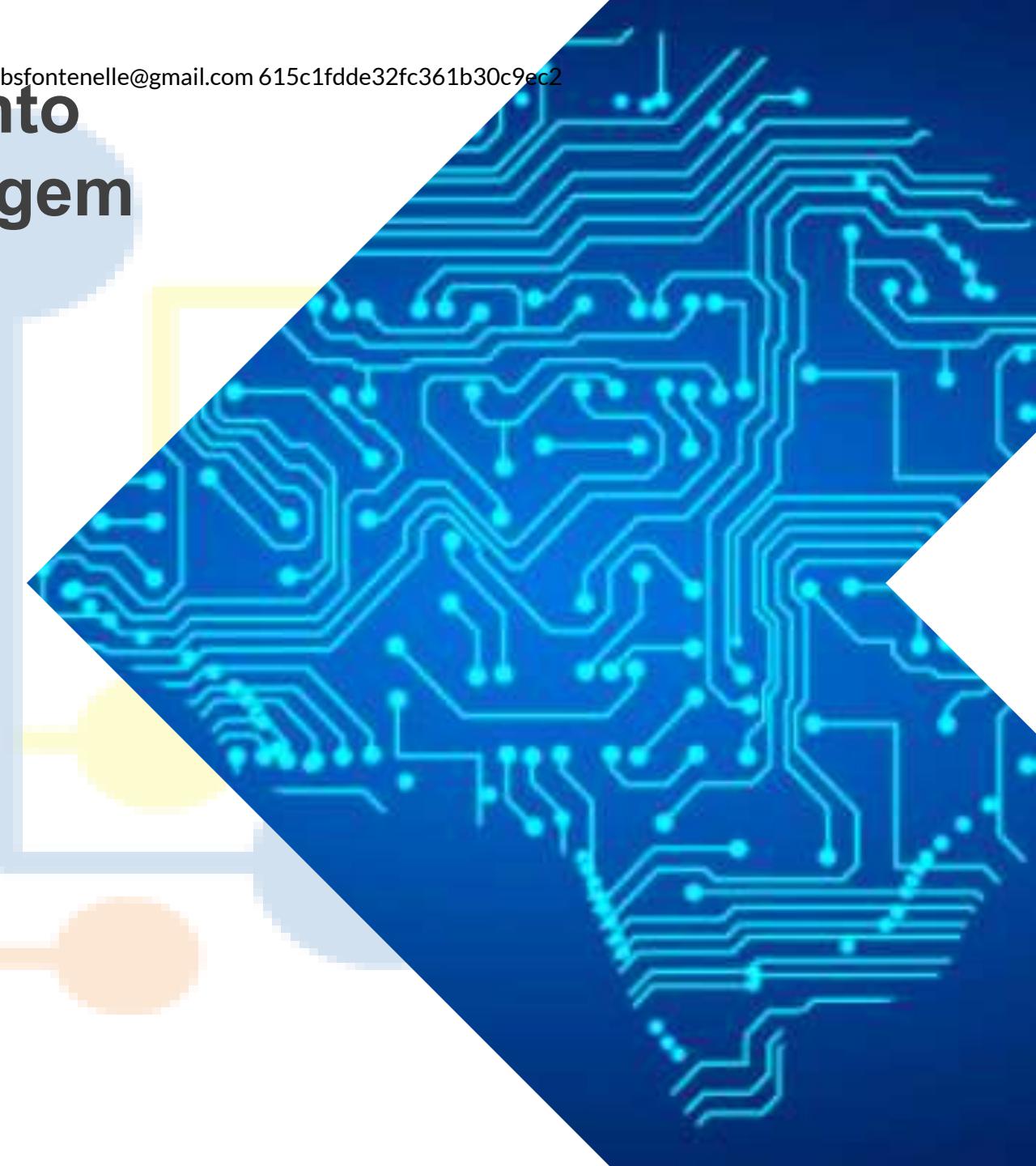


Representação, Conhecimento Prévio e Tipos de Aprendizagem

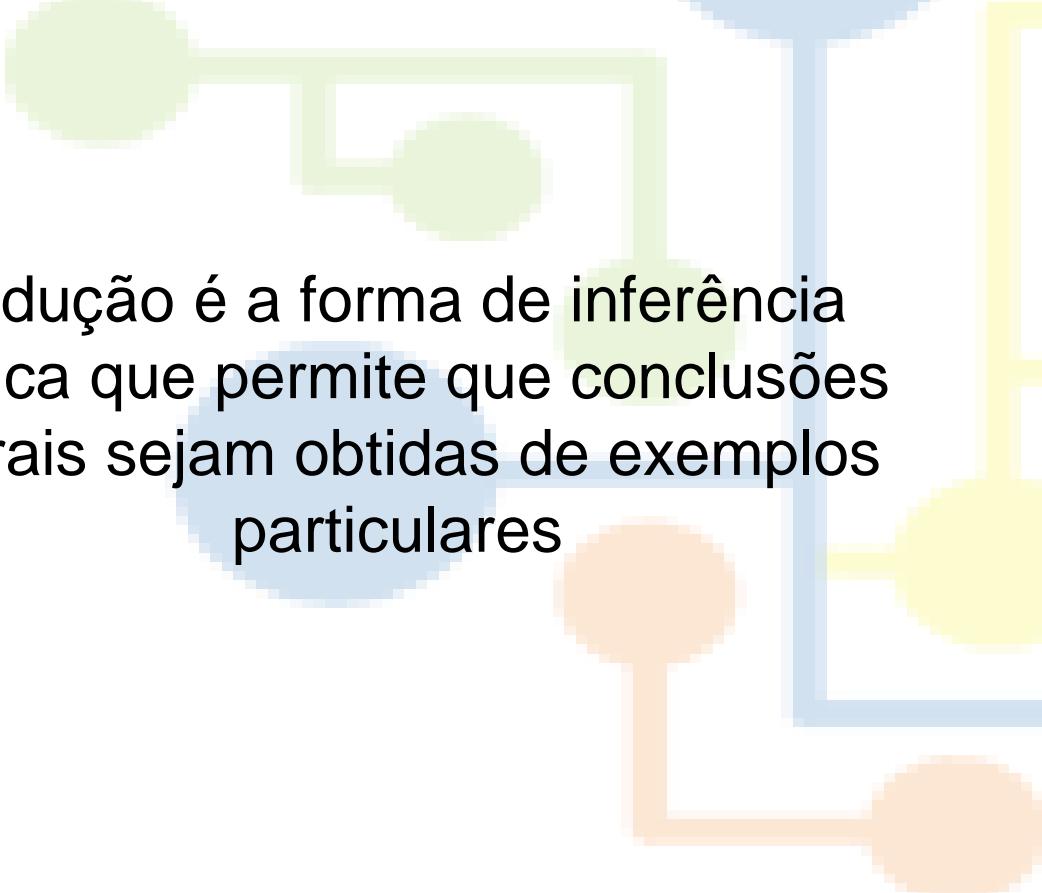


Representação, Conhecimento Prévio e Tipos de Aprendizagem

Ao longo deste capítulo, veremos as entradas que formam uma representação fatorada — **um vetor de valores e atributos** — e saídas que podem ser tanto um valor contínuo numérico como um valor discreto.



Representação, Conhecimento Prévio e Tipos de Aprendizagem



Indução é a forma de inferência lógica que permite que conclusões gerais sejam obtidas de exemplos particulares



Representação, Conhecimento Prévio e Tipos de Aprendizagem

O processo de indução é indispensável ao ser humano, pois é um dos principais meios de criar novos conhecimentos e prever eventos futuros

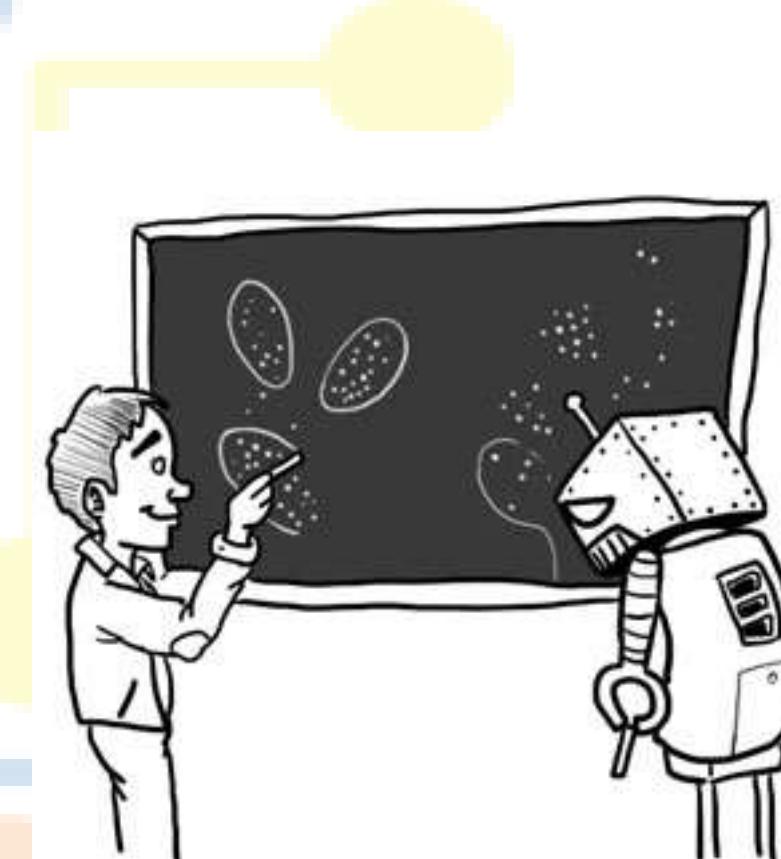


Representação, Conhecimento Prévio e Tipos de Aprendizagem

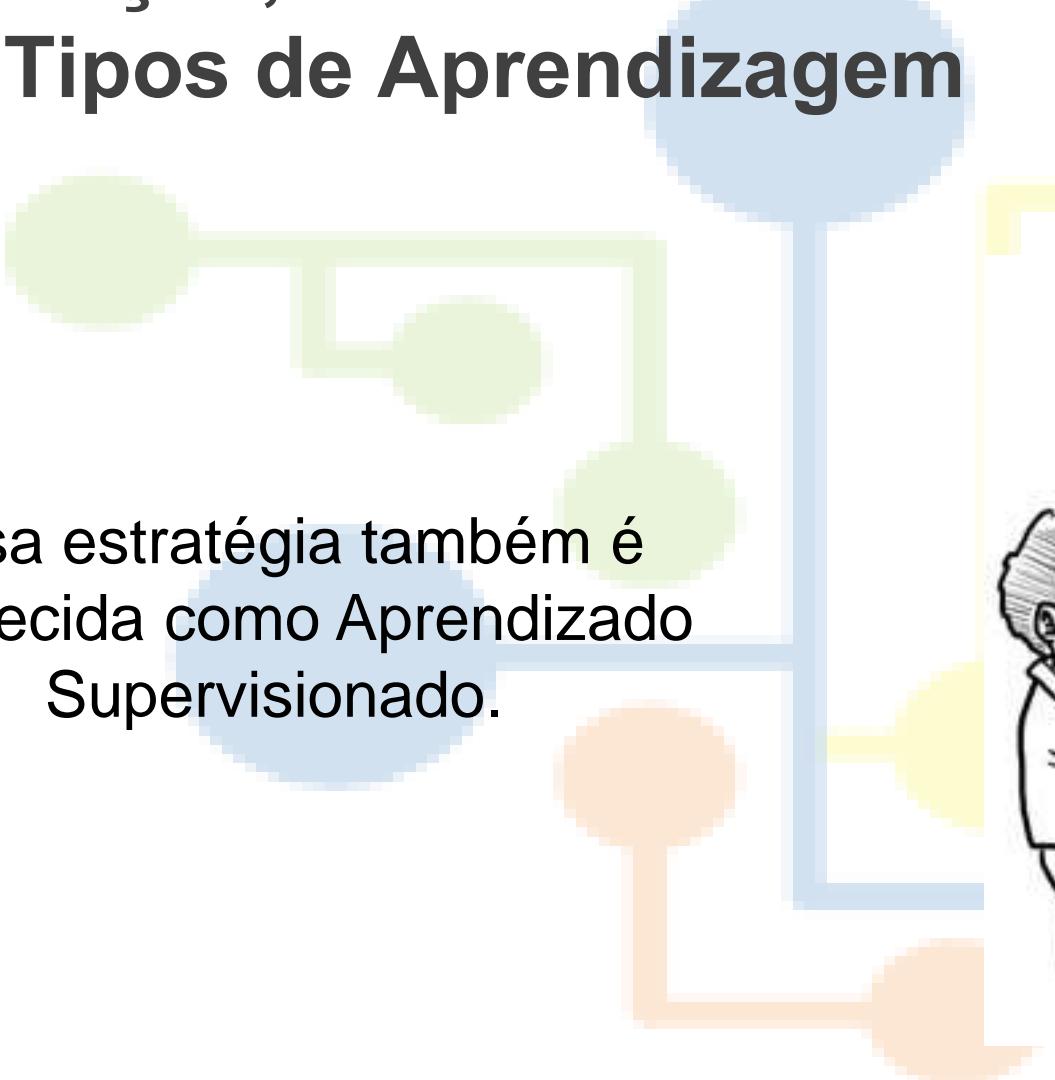


Representação, Conhecimento Prévio e Tipos de Aprendizagem

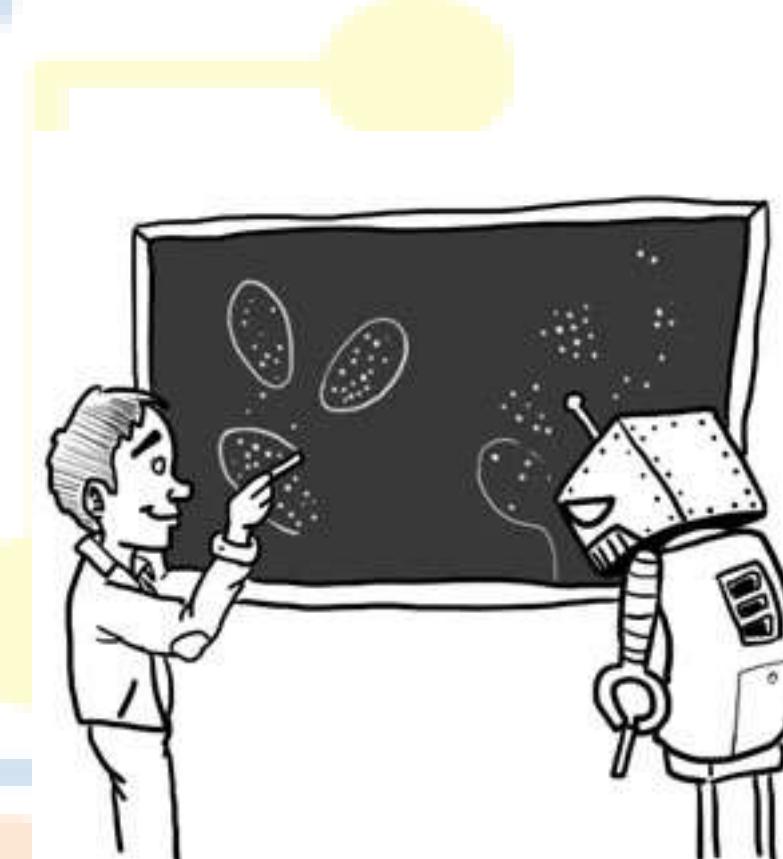
No **Aprendizado por Exemplos**, o aprendiz induz a descrição de um conceito formulando uma regra geral a partir dos exemplos e dos contra-exemplos fornecidos pelo professor ou pelo ambiente.



Representação, Conhecimento Prévio e Tipos de Aprendizagem

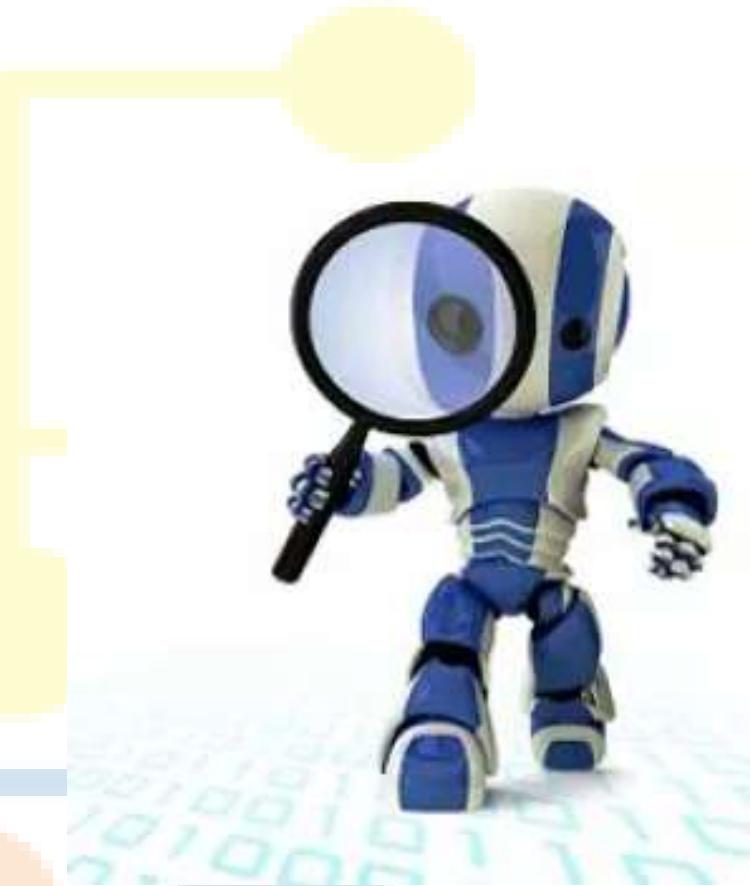


Essa estratégia também é conhecida como Aprendizado Supervisionado.

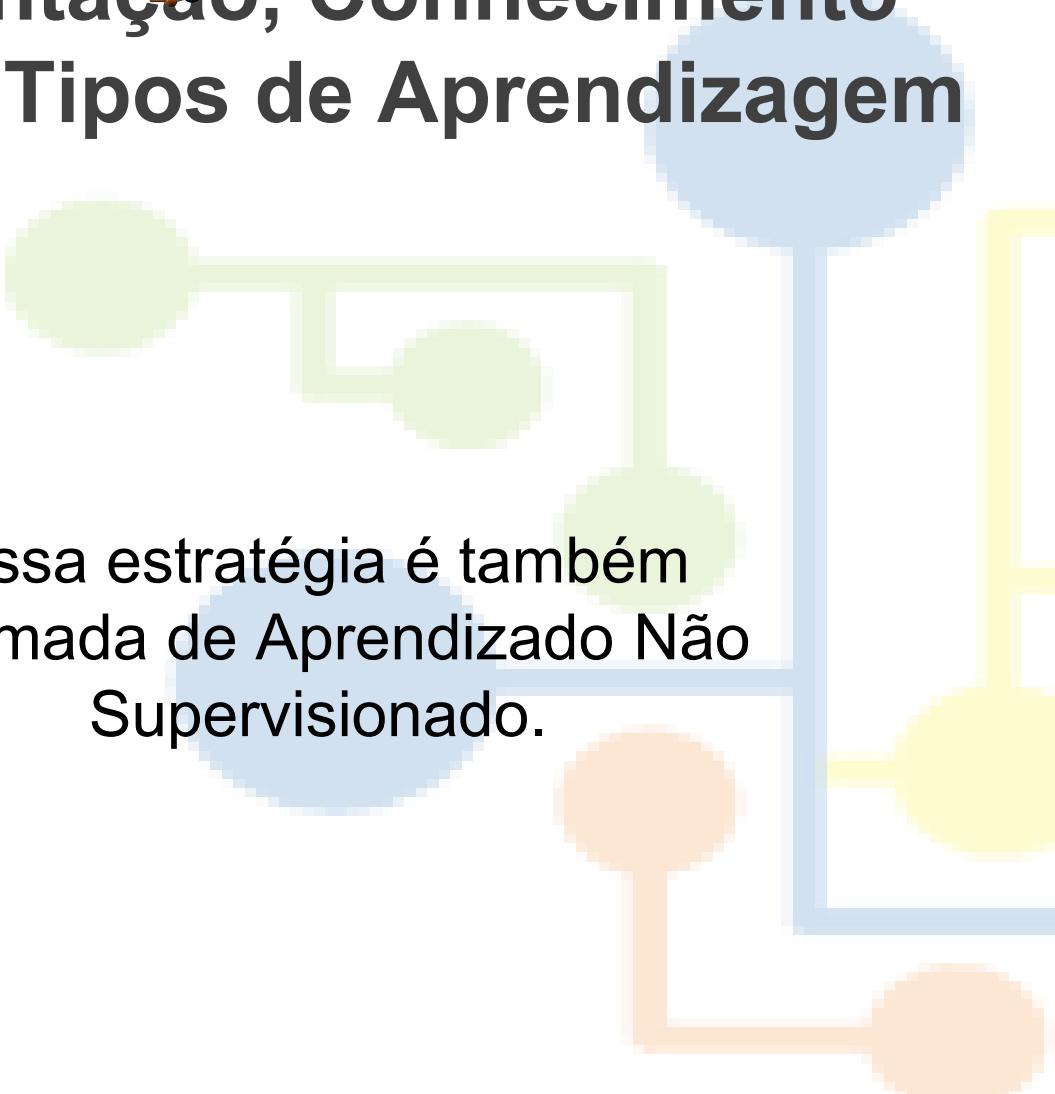


Representação, Conhecimento Prévio e Tipos de Aprendizagem

No Aprendizado por Observação, o aprendiz analisa entidades fornecidas ou observadas e tenta determinar se alguns subconjuntos dessas entidades podem ser agrupados em certas classes de maneira útil.



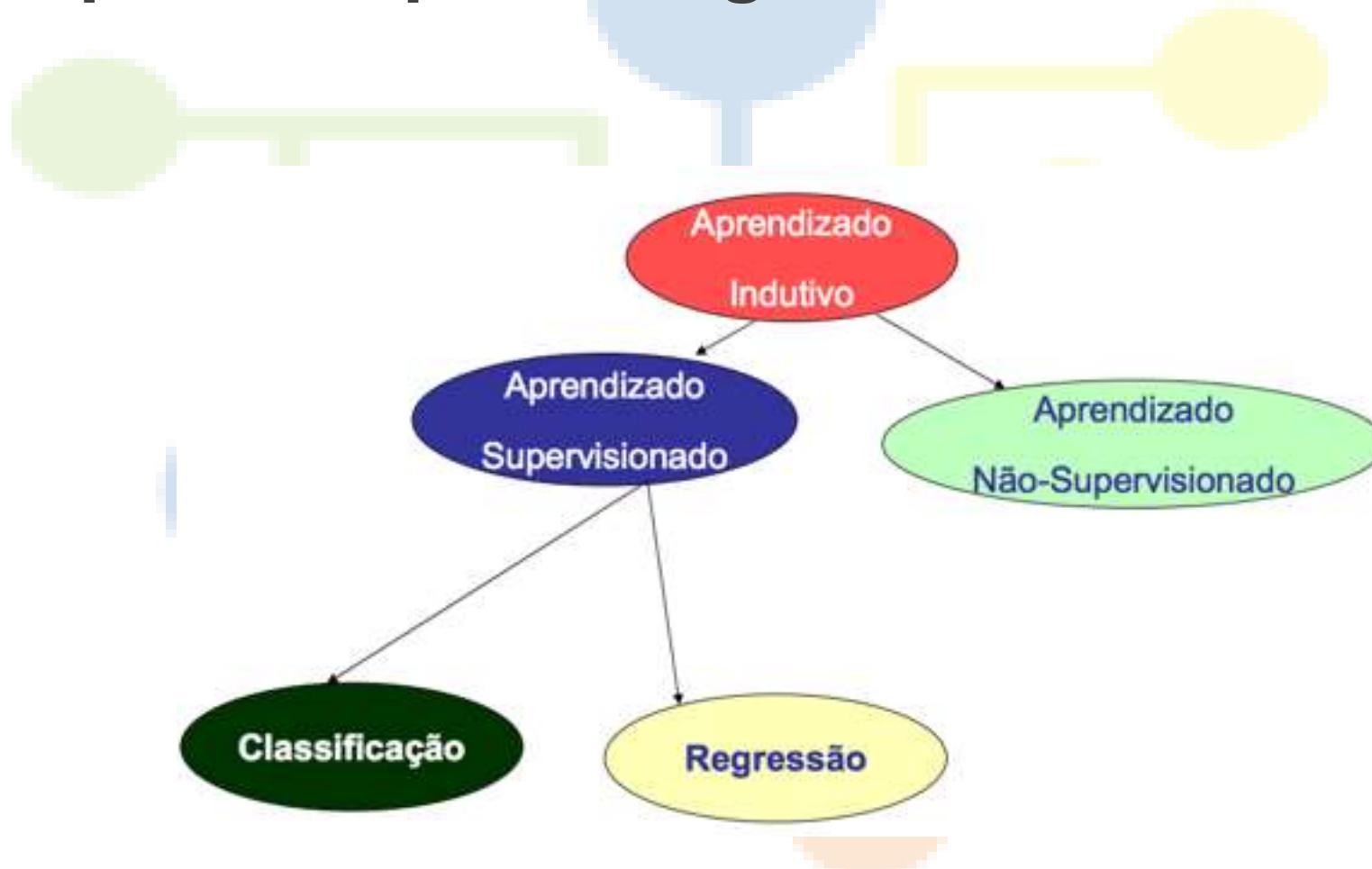
Representação, Conhecimento Prévio e Tipos de Aprendizagem



Essa estratégia é também
chamada de Aprendizado Não
Supervisionado.



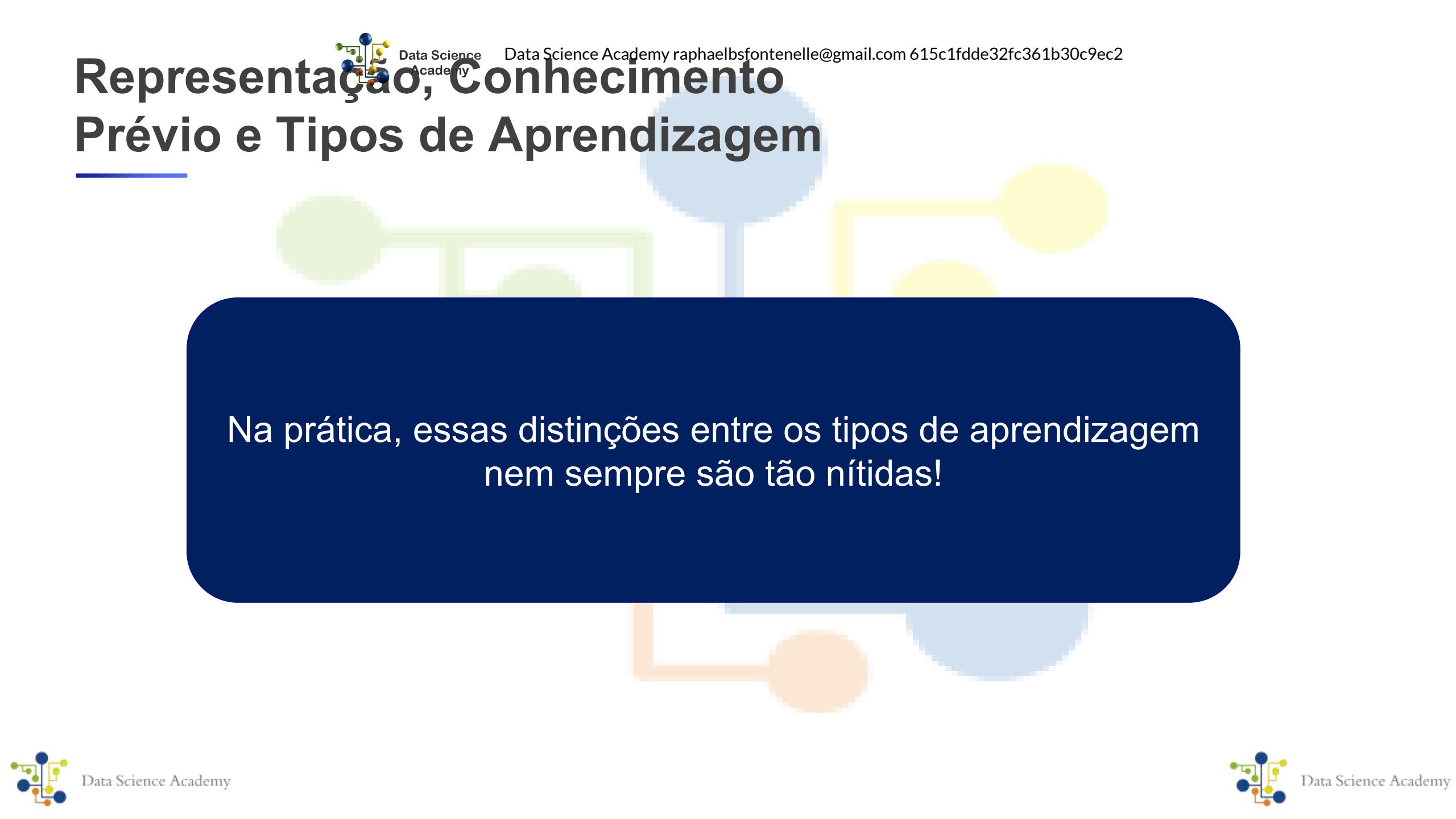
Representação, Conhecimento Prévio e Tipos de Aprendizagem



Representação, Conhecimento Prévio e Tipos de Aprendizagem



Representação, Conhecimento Prévio e Tipos de Aprendizagem



Na prática, essas distinções entre os tipos de aprendizagem nem sempre são tão nítidas!



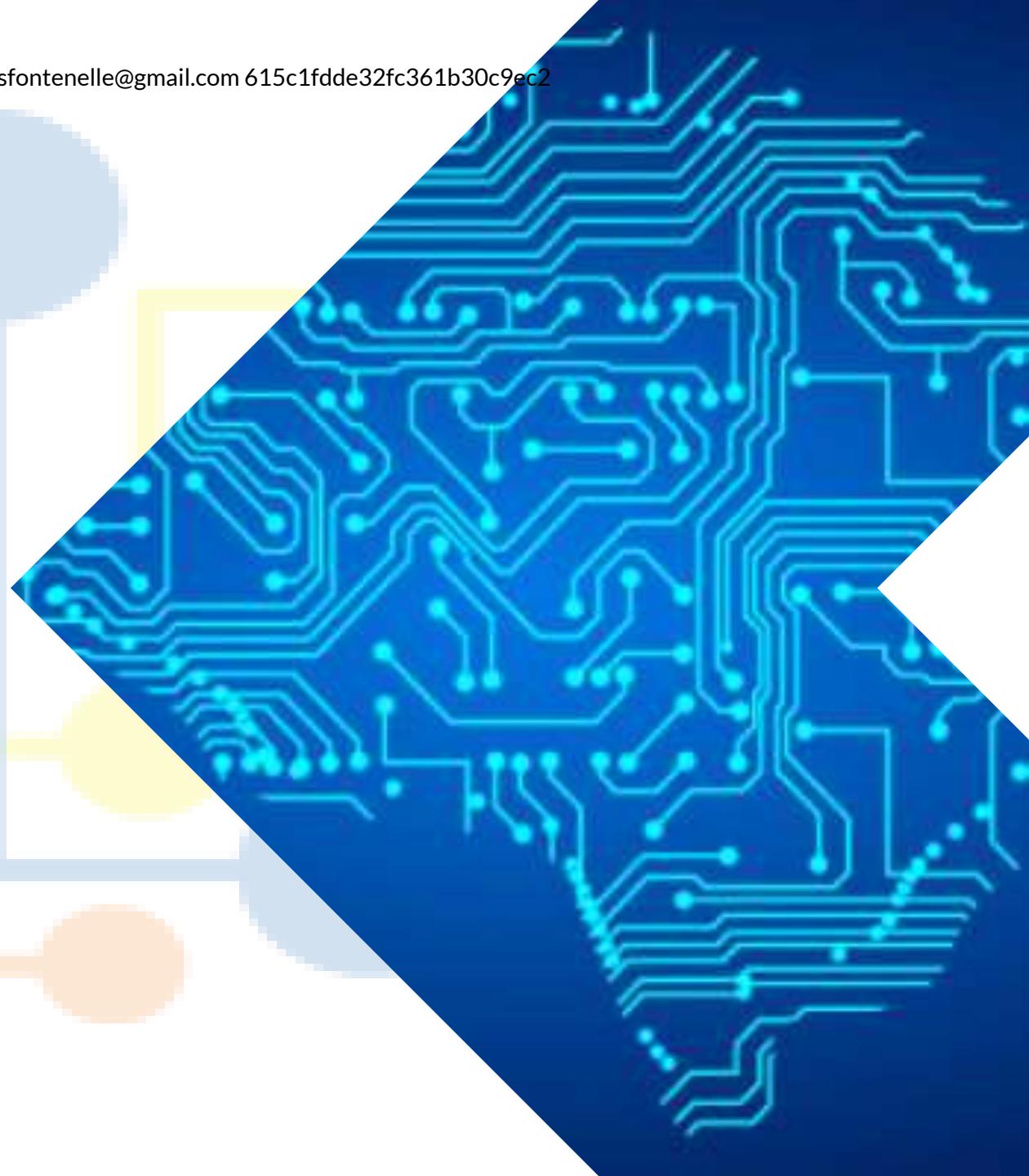
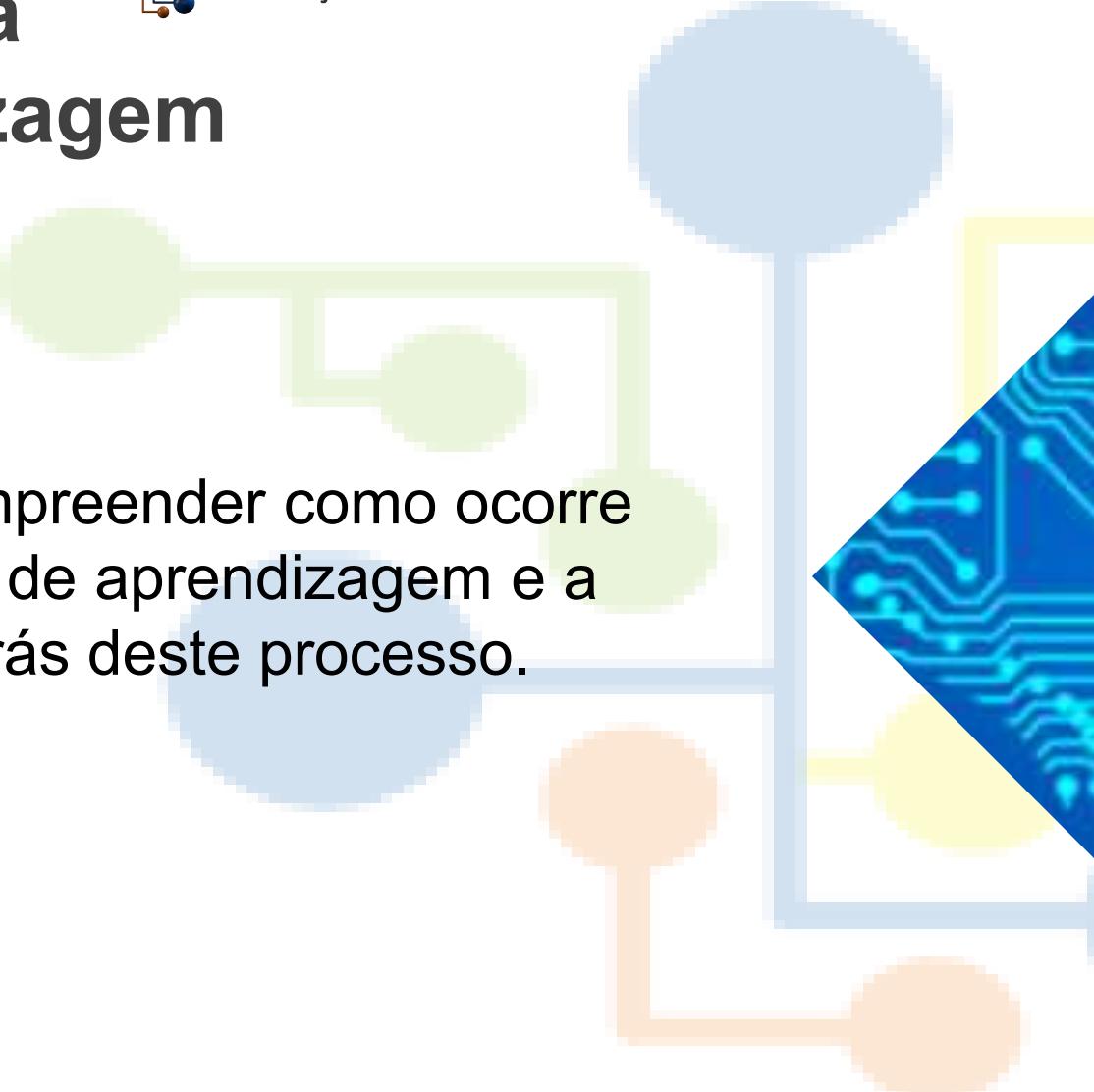
Teoria da Aprendizagem





Teoria da Aprendizagem

Vamos compreender como ocorre o processo de aprendizagem e a teoria por trás deste processo.



Teoria da Aprendizagem

Dado um conjunto de treinamento de N pares de exemplos de entrada e saída

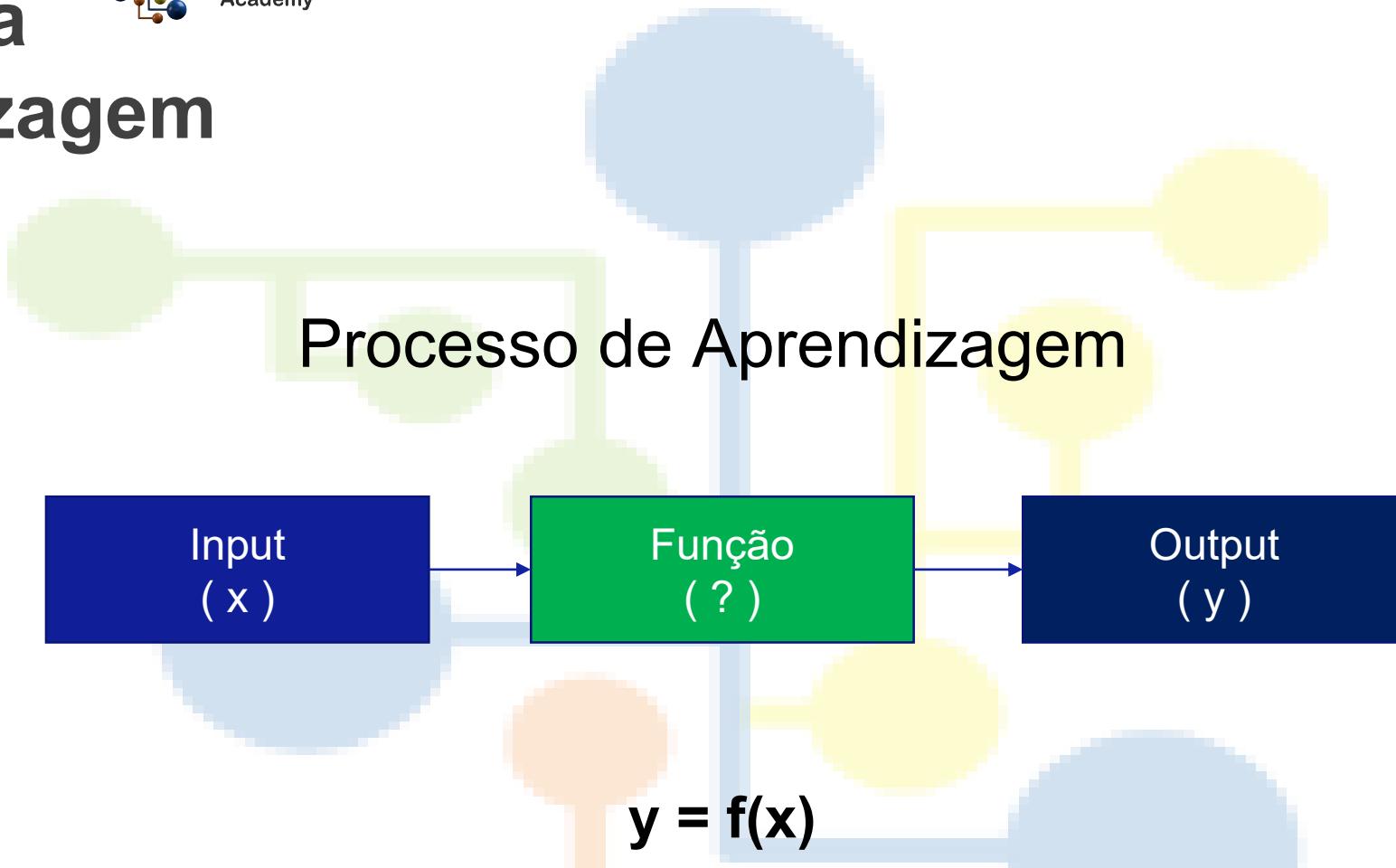
$$(x_1, y_1), (x_2, y_2), \dots (x_n, y_n),$$

onde cada valor de y pode ser encontrado por uma função desconhecida:

$$y = f(x),$$

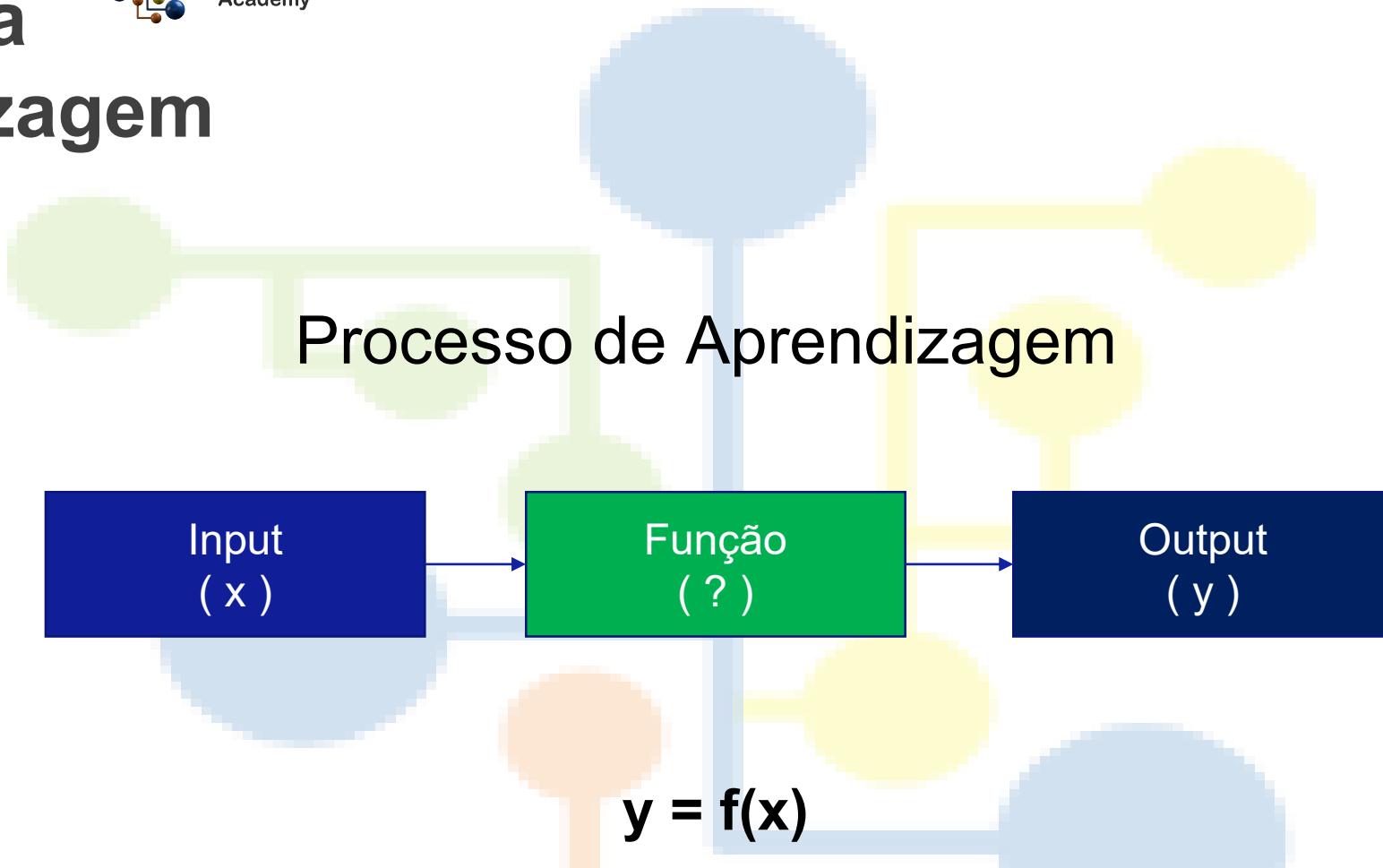
o objetivo da aprendizagem é descobrir uma função h (hipótese) que se aproxime da função verdadeira f .

Teoria da Aprendizagem



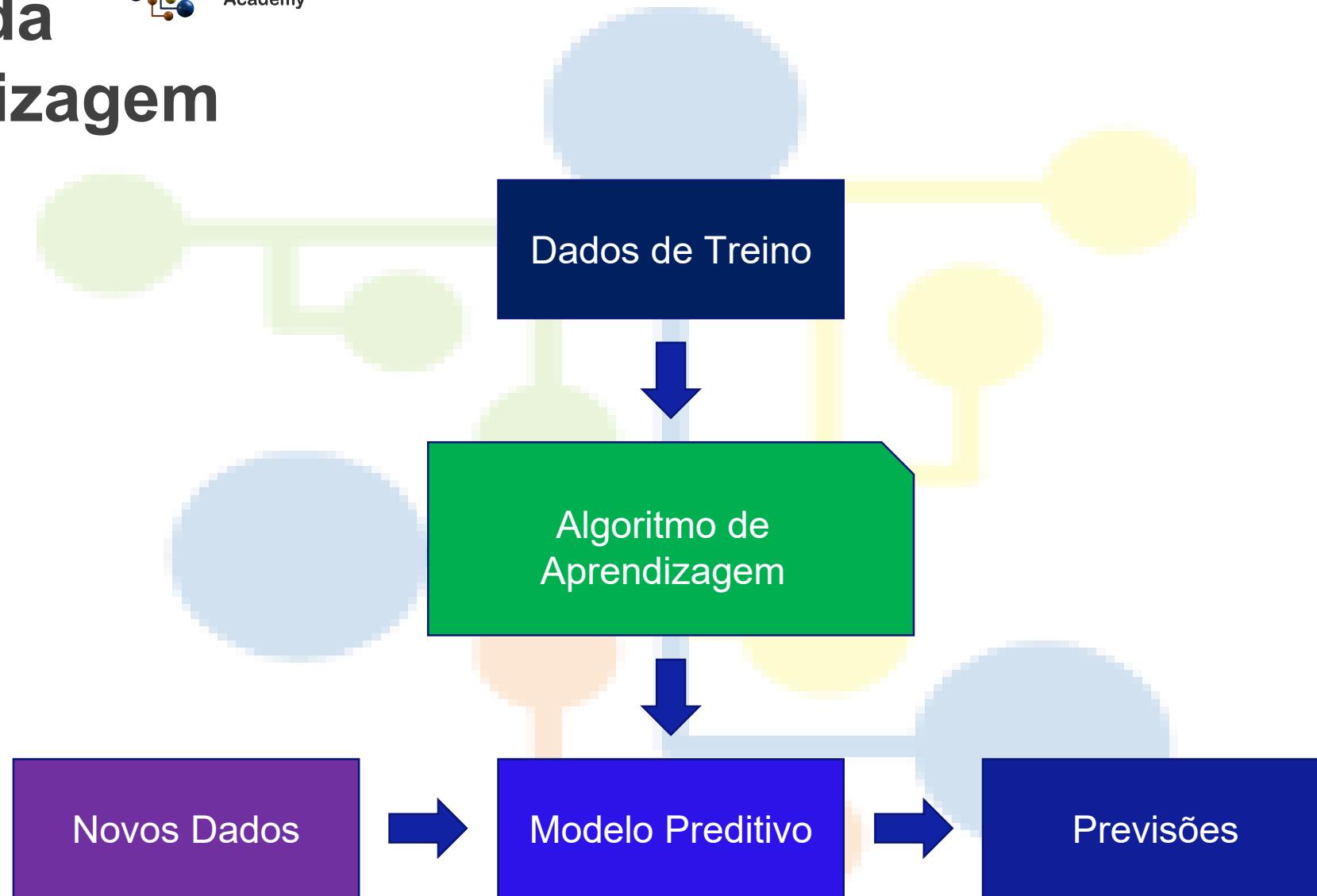
A função h é uma hipótese. Aprendizagem é uma busca através do espaço de hipóteses possíveis, por aquela que terá um bom desempenho, mesmo em novos exemplos além do conjunto de treinamento.

Teoria da Aprendizagem



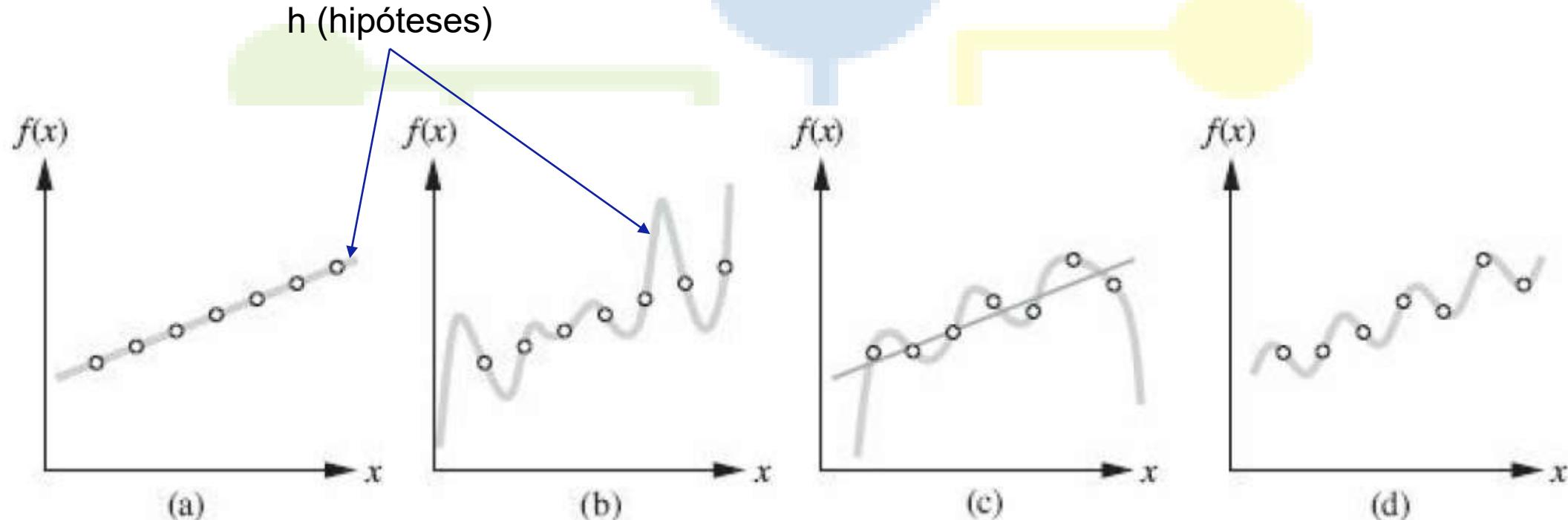
Às vezes, a função f é estocástica — não é estritamente uma função de x , e o que temos de aprender é uma distribuição de probabilidade condicional, $P(Y|x)$.

Teoria da Aprendizagem



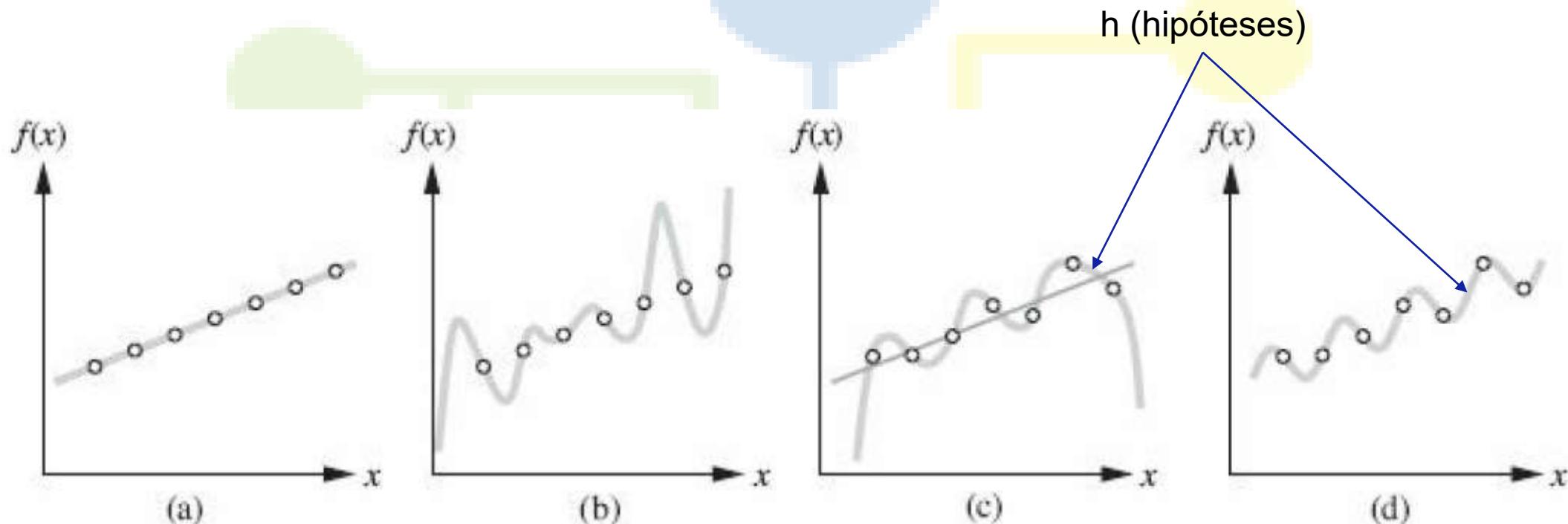


Teoria da Aprendizagem



Os exemplos são pontos no plano (x, y) , onde $y = f(x)$.

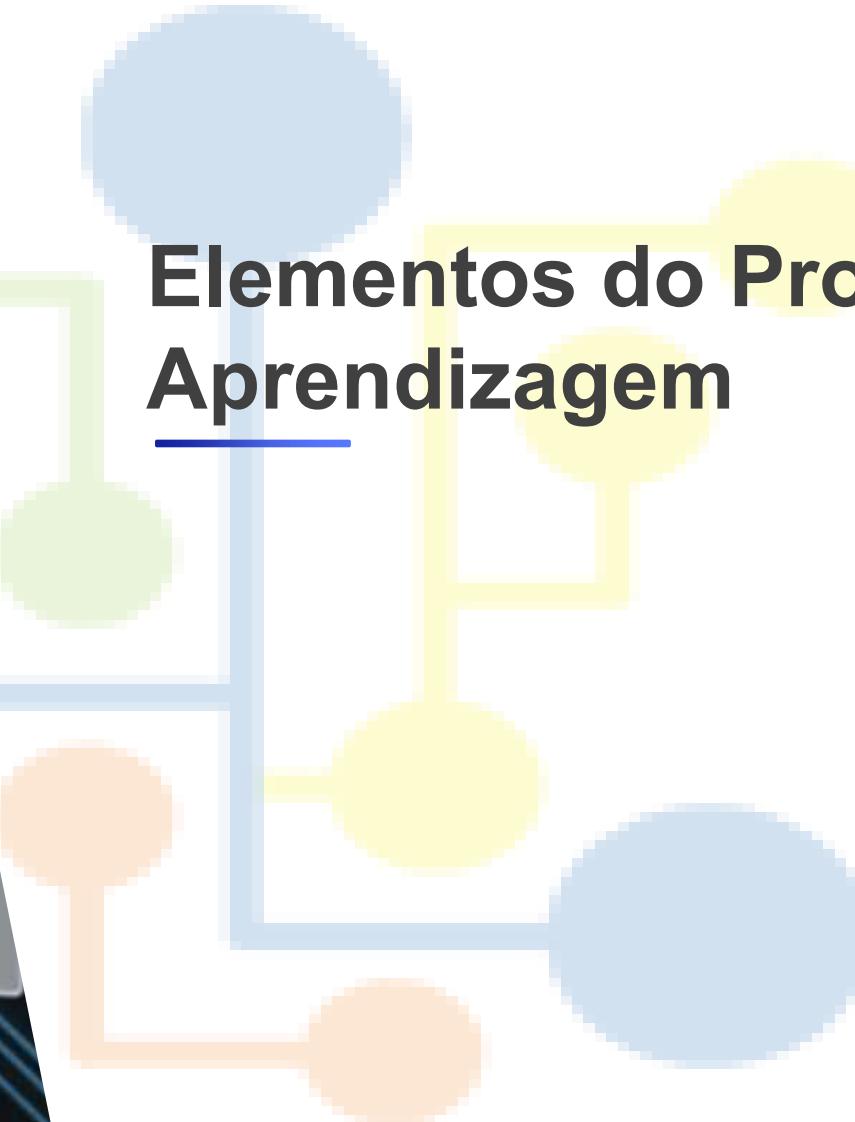
Teoria da Aprendizagem



Os exemplos são pontos no plano (x, y) , onde $y = f(x)$.



Elementos do Processo de Aprendizagem



Elementos do Processo de Aprendizagem



Aprovação de
Crédito

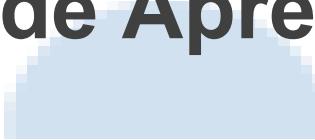


Elementos do Processo de Aprendizagem

Aprovação de Crédito de um Indivíduo (dados históricos)



Atributo	Valor
Sexo	Masculino
Idade	34
Salário Mensal	R\$ 15.000,00
Anos no Emprego Atual	4
Anos de Residência	9
Saldo Bancário	R\$ 49.781,23
Recebeu crédito	
Sim	



Elementos do Processo de Aprendizagem

Input x

{Dados do cliente}

Output y

{Decisão → Crédito: Sim/Não}

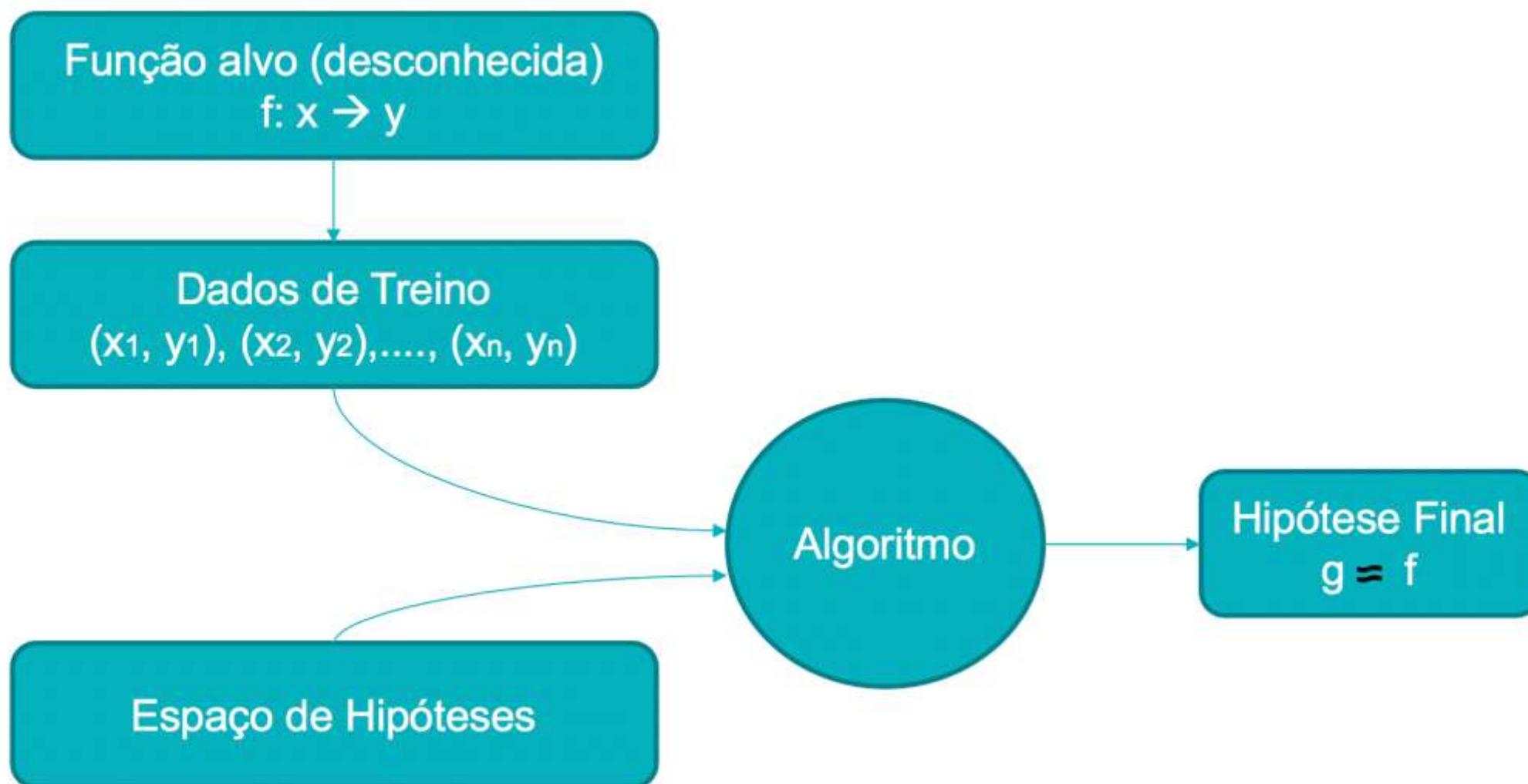
Função alvo $f: x \rightarrow y$ { {Representação do relacionamento} }
{ {Fórmula matemática desconhecida} }**Dados** $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$

{Dados históricos}

Hipótese $g: x \rightarrow y$

{Fórmula a ser usada}

Elementos do Processo de Aprendizagem

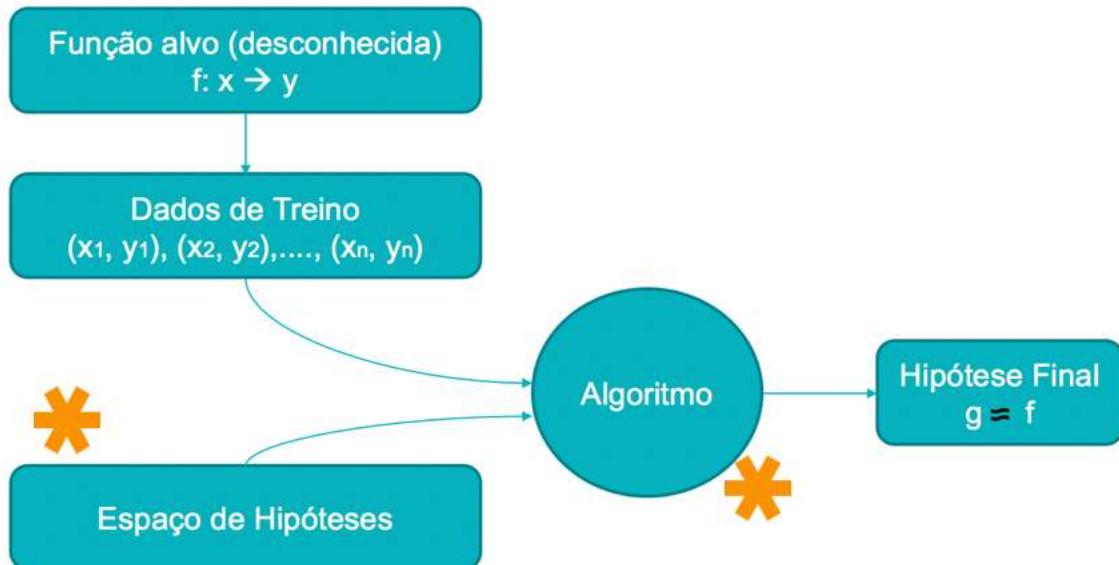




Modelo de Aprendizagem e Espaço de Hipóteses



Modelo de Aprendizagem e Espaço de Hipóteses



- Espaço de Hipóteses

$$\mathcal{H} = \{h\} \quad g \in \mathcal{H}$$

- Algoritmo de Aprendizagem

Espaço de Hipóteses

Redes Neurais
Support Vector Machines

**Algoritmo de
Aprendizagem**

Back Propagation
Programação Quadrática

**Modelo de
Aprendizagem**

Modelo de Aprendizagem e Espaço de Hipóteses



O Espaço de Hipóteses contém os recursos com os quais podemos trabalhar. O Algoritmo de Aprendizagem recebe os dados e navega pelo Espaço de Hipóteses a fim de encontrar a melhor hipótese que gera o resultado desejado.



Fórmula que Define as Hipóteses no Espaço de Hipóteses

$$h(\mathbf{x}) = \text{sign} \left(\left(\sum_{i=1}^d w_i x_i \right) - \text{threshold} \right)$$

As diferentes combinações weight/threshold vão formar diferentes hipóteses



Input → $X = (x_1, x_2, \dots, x_d)$

Vetor de atributos do indivíduo

$$\sum_{i=1}^d w_i x_i$$

$$\sum_{i=1}^d w_i x_i$$

Weight (Peso)





Input $\rightarrow X = (x_1, x_2, \dots, x_d)$

Crédito é **aprovado** se

$$\sum_{i=1}^d w_i x_i$$

$> \text{threshold}$

Crédito é **negado** se

$$\sum_{i=1}^d w_i x_i$$

$< \text{threshold}$



Fórmula que Define as Hipóteses no Espaço de Hipóteses

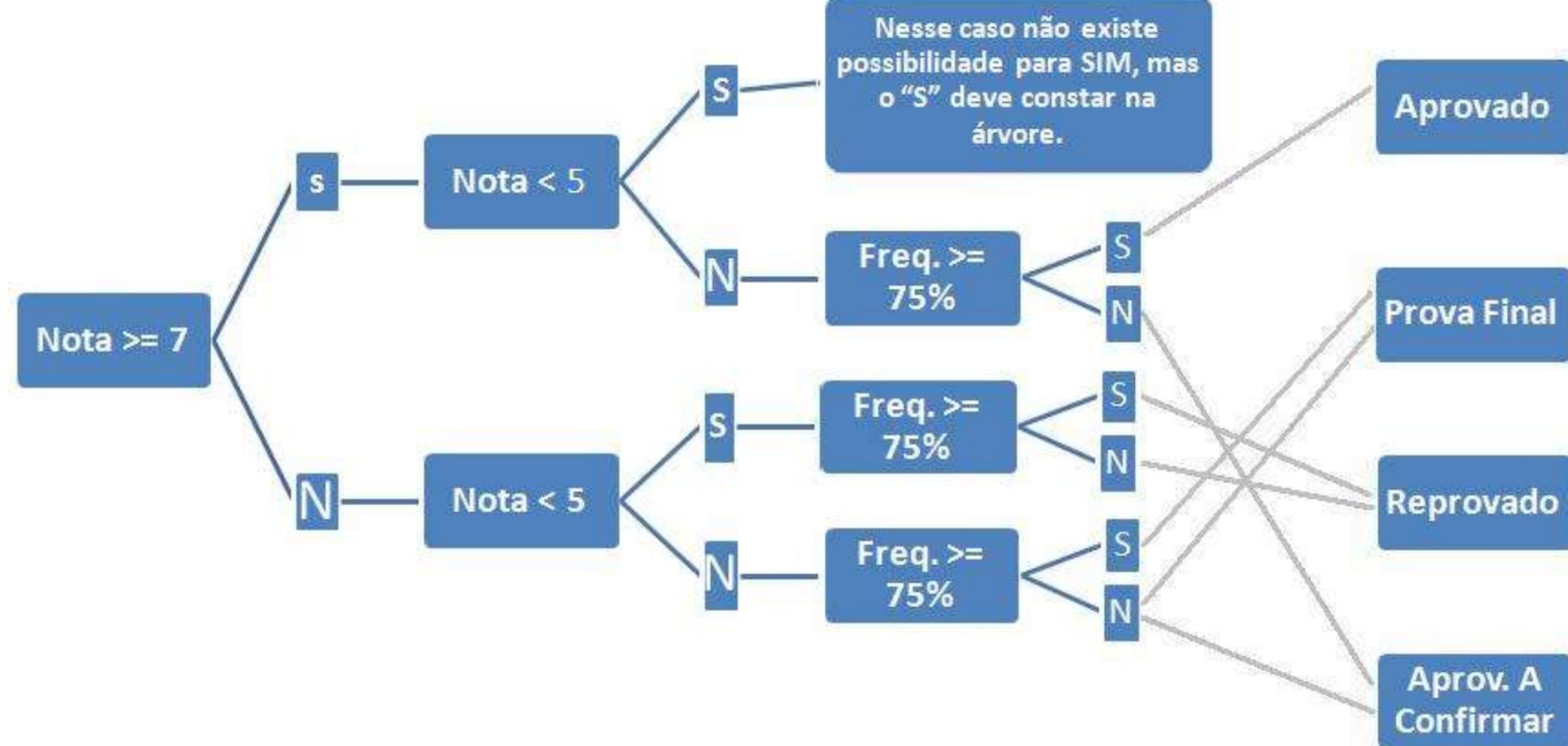
$$h(\mathbf{x}) = \text{sign} \left(\left(\sum_{i=1}^d w_i x_i \right) - \text{threshold} \right)$$

As diferentes combinações weight/threshold vão formar diferentes hipóteses

Aprendizagem em Árvores de Decisão

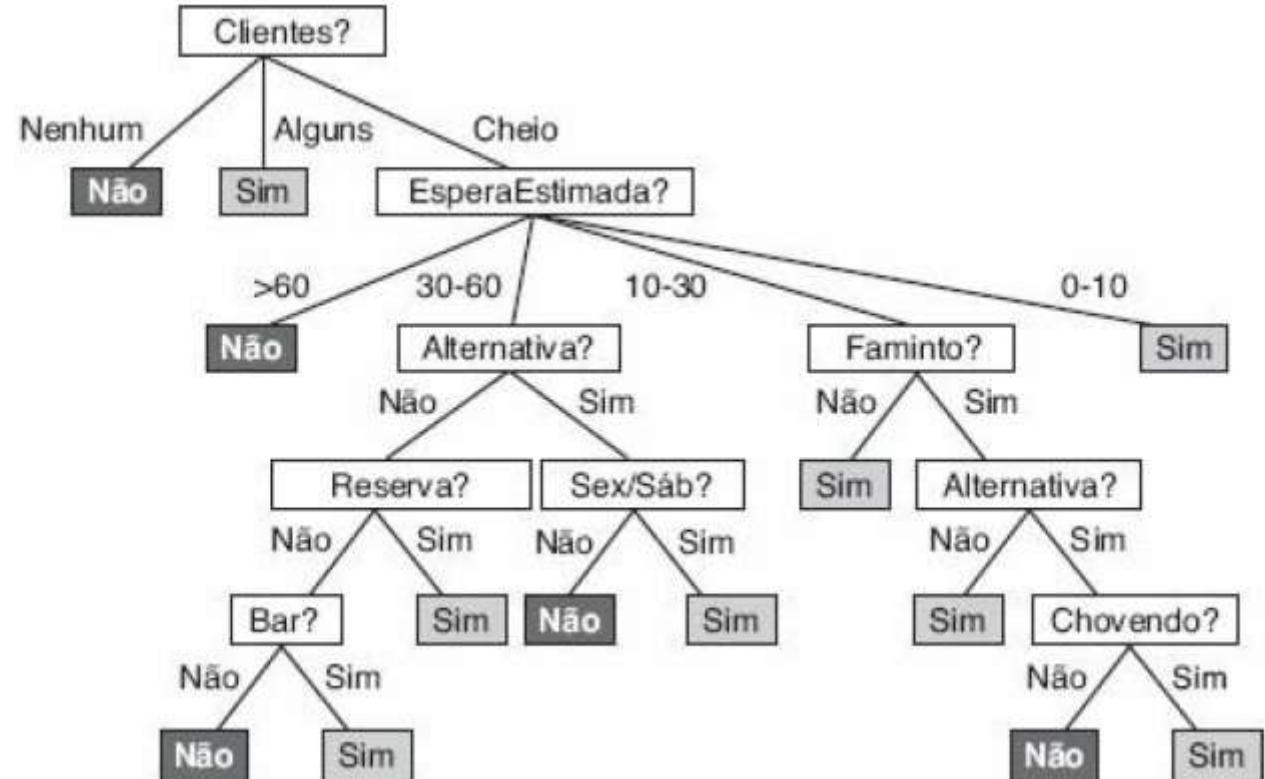


Aprendizagem em Árvores de Decisão



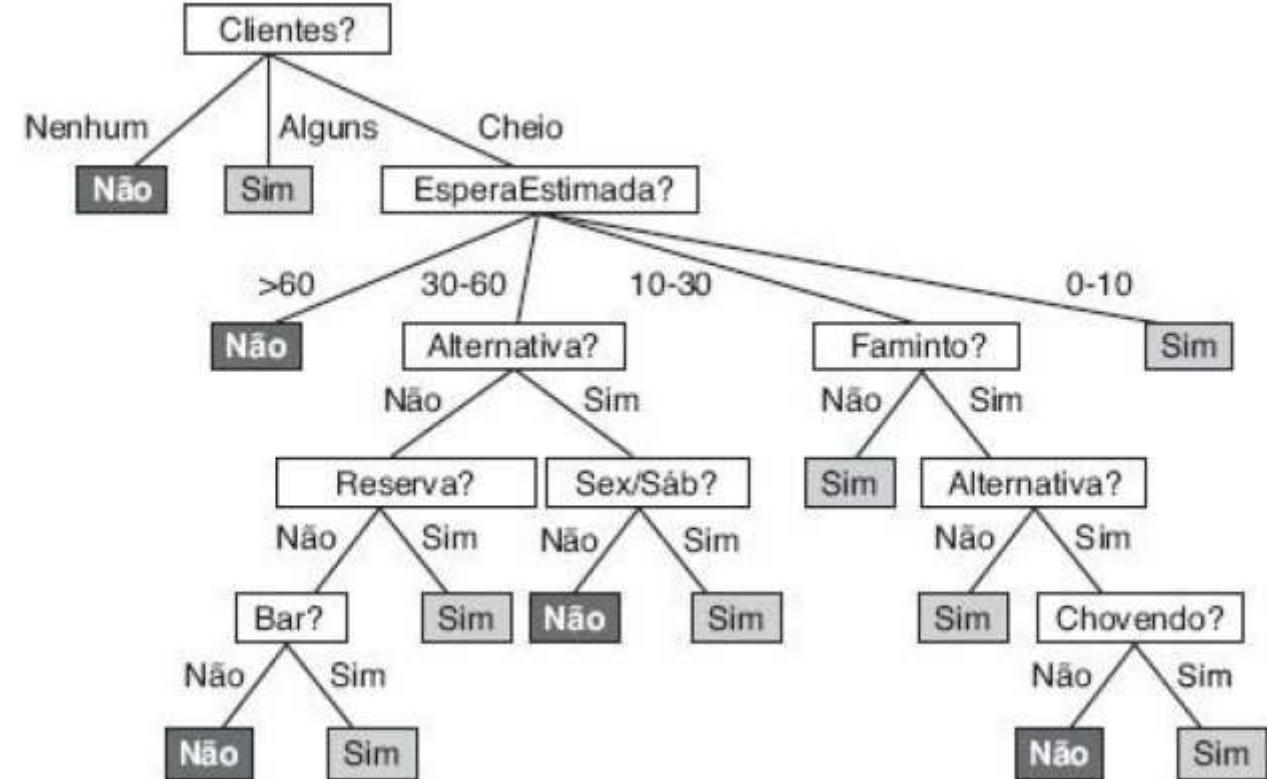
Aprendizagem em Árvores de Decisão

Uma árvore de decisão representa uma função que toma como entrada um vetor de valores de atributos e retorna uma “decisão” — um valor de saída, único.



Aprendizagem em Árvores de Decisão

Uma árvore de decisão alcança sua decisão executando uma sequência de testes.



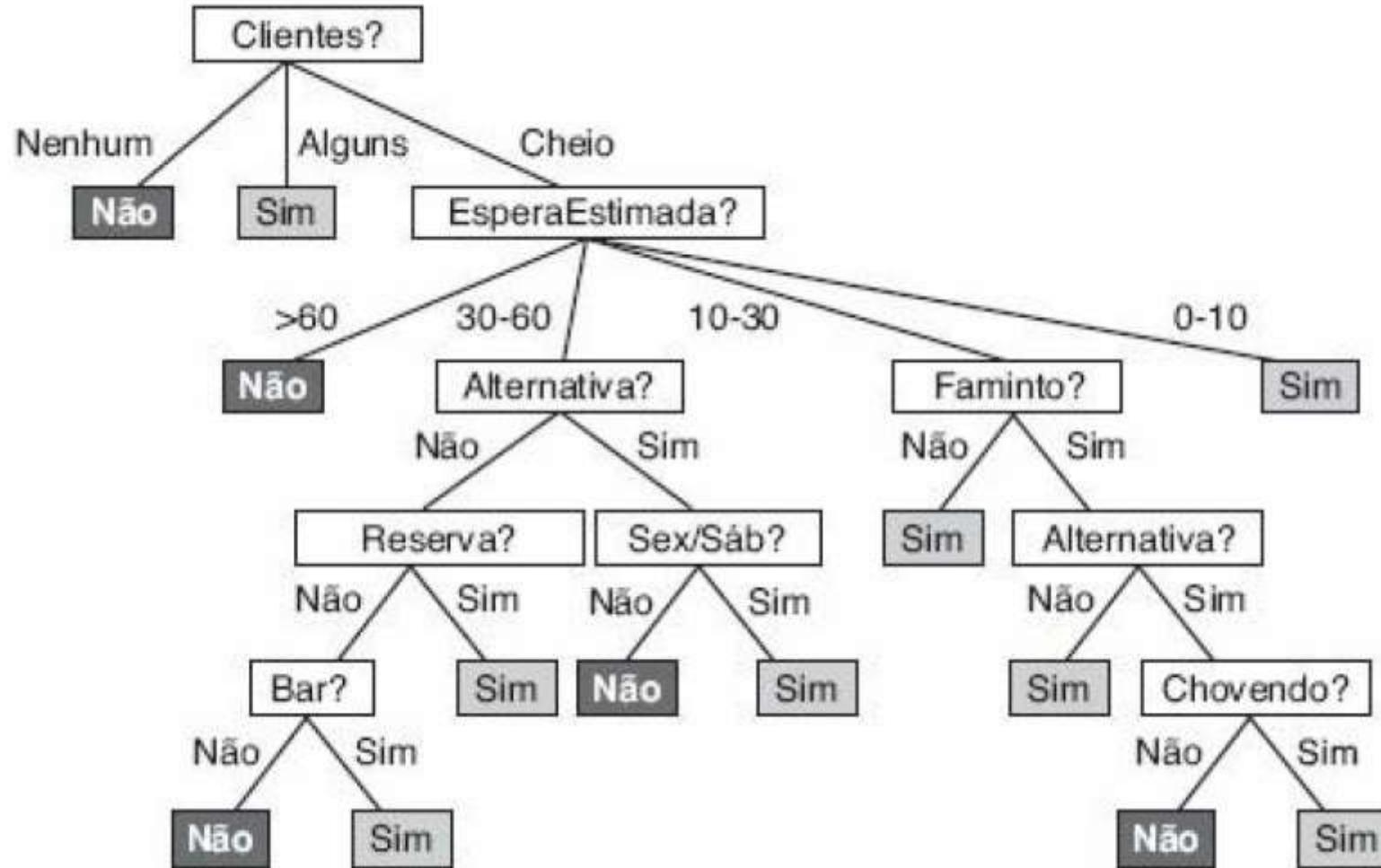
Aprendizagem em Árvores de Decisão

X

- 1. **Alternativa:** Se há um restaurante alternativo apropriado por perto.
- 2. **Bar:** Se o restaurante tem uma área de bar confortável onde se possa esperar.
- 3. **Sex/Sáb:** Verdadeiro às sextas e sábados.
- 4. **Faminto:** Se estamos com fome.
- 5. **Clientes:** Quantas pessoas estão no restaurante (os valores são: Nenhum, Alguns e Cheio).
- 6. **Preço:** A faixa de preços do restaurante (\$, \$\$, \$\$\$).
- 7. **Chovendo:** Se está chovendo do lado de fora.
- 8. **Reserva:** Se fizemos uma reserva.
- 9. **Tipo:** O tipo de restaurante (francês, italiano, tailandês ou só de hambúrguer).
- 10. **EsperaEstimada:** A espera estimada pelo gerente (0-10 minutos, 10-30, 30-60, >60).

Variável target (y): **VaiEsperar** (label)

Aprendizagem em Árvores de Decisão

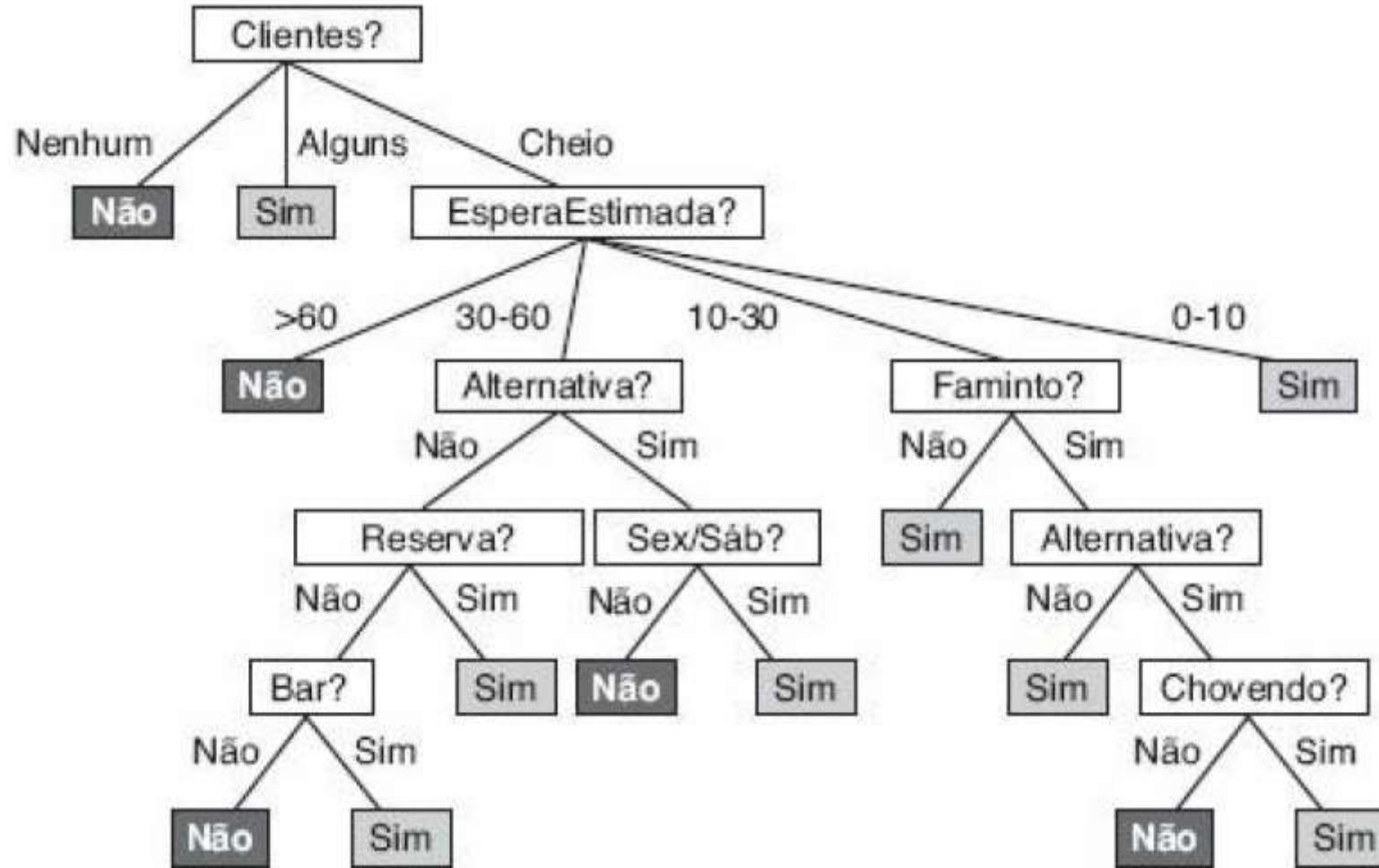


Aprendizagem em Árvores de Decisão

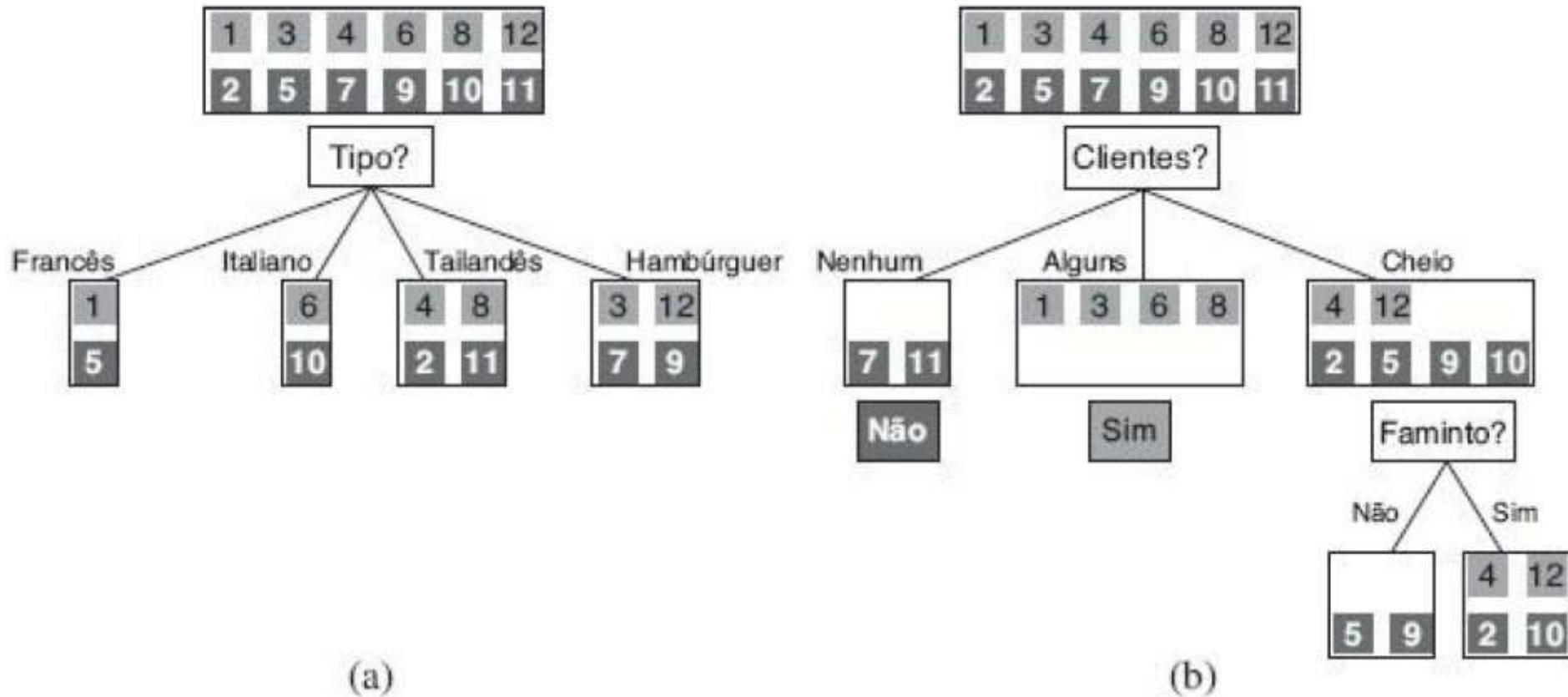
Exemplo	Atributos										Meta <i>VaiEsperar</i>
	Alt	Bar	Sex	Fam	Cli	Preço	Chuva	Res	Tipo	Estim	
x_1											
x_2	Sim	Não	Não	Sim	Alguns	\$\$\$	Não	Sim	Francês	0-10	$y_1 = Sim$
x_3	Sim	Não	Não	Sim	Cheio	\$	Não	Não	Tailandês	30-60	$y_2 = Não$
x_4	Não	Sim	Não	Não	Alguns	\$	Não	Não	Hambúrguer	0-10	$y_3 = Sim$
x_5	Sim	Não	Sim	Não	Cheio	\$\$\$	Não	Sim	Francês	>60	$y_4 = Sim$
x_6	Não	Sim	Não	Sim	Alguns	\$\$	Sim	Sim	Italiano	0-10	$y_5 = Não$
x_7	Não	Sim	Não	Não	Nenhum	\$	Sim	Não	Hambúrguer	0-10	$y_6 = Sim$
x_8	Não	Não	Não	Sim	Alguns	\$\$	Sim	Sim	Tailandês	0-10	$y_7 = Não$
x_9	Não	Sim	Sim	Não	Cheio	\$	Sim	Não	Hambúrguer	>60	$y_8 = Sim$
x_{10}	Sim	Sim	Sim	Sim	Cheio	\$\$\$	Não	Sim	Italiano	10-30	$y_9 = Não$
x_{11}	Não	Não	Não	Não	Nenhum	\$	Não	Não	Tailandês	0-10	$y_{10} = Não$
x_{12}	Sim	Sim	Sim	Sim	Cheio	\$	Não	Não	Hambúrguer	30-60	$y_{11} = Não$
											$y_{12} = Sim$

Um exemplo de árvore de decisão booleana consiste em um par (x, y) , onde x é um vetor de valores para os atributos de entrada e y é um valor único de saída booleano

Aprendizagem em Árvores de Decisão



Aprendizagem em Árvores de Decisão



(a)

(b)

Aprendizagem em Árvores de Decisão

função APRENDIZAGEM-EM-ÁRVORE-DE-DECISÃO(*exemplos, atributos, exemplos-pais*)

retorna uma árvore de decisão

se *exemplos* é vazio **então retornar** VALOR-DA-MAIORIA (*exemplos_pais*)

senão se todos os *exemplos* têm a mesma classificação **então retornar** a classificação

senão se *atributos* é vazio **então retornar** VALOR-DA-MAIORIA(*exemplos*)

senão

A $\leftarrow \operatorname{argmax}_a \square \text{atributes} \text{IMPORTÂNCIA}(a, \text{exemplos})$

árvore \leftarrow uma nova árvore de decisão com teste de raiz *A*

para cada valor v_k de *A* **faça**

exs $\leftarrow \{e : e \square \text{exemplos} \text{ e } e.A = v_k\}$

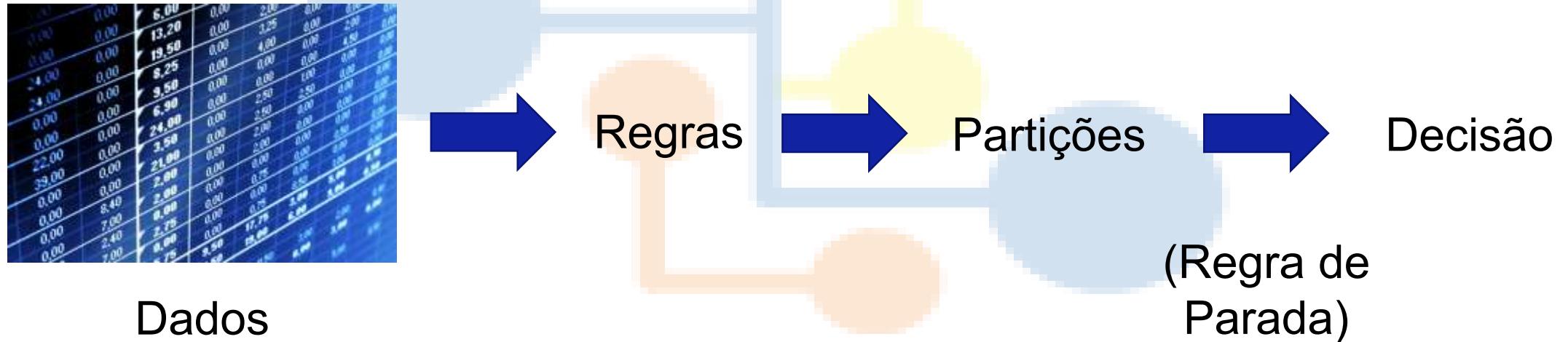
subárvore \leftarrow APRENDIZAGEM-EM-ÁRVORE-DE-DECISÃO (*exs, atributos — A, exemplos*)

adicionar uma ramificação à árvore com rótulo ($A = v_k$) e subárvore *subárvore*

retornar árvore

Aprendizagem em Árvores de Decisão

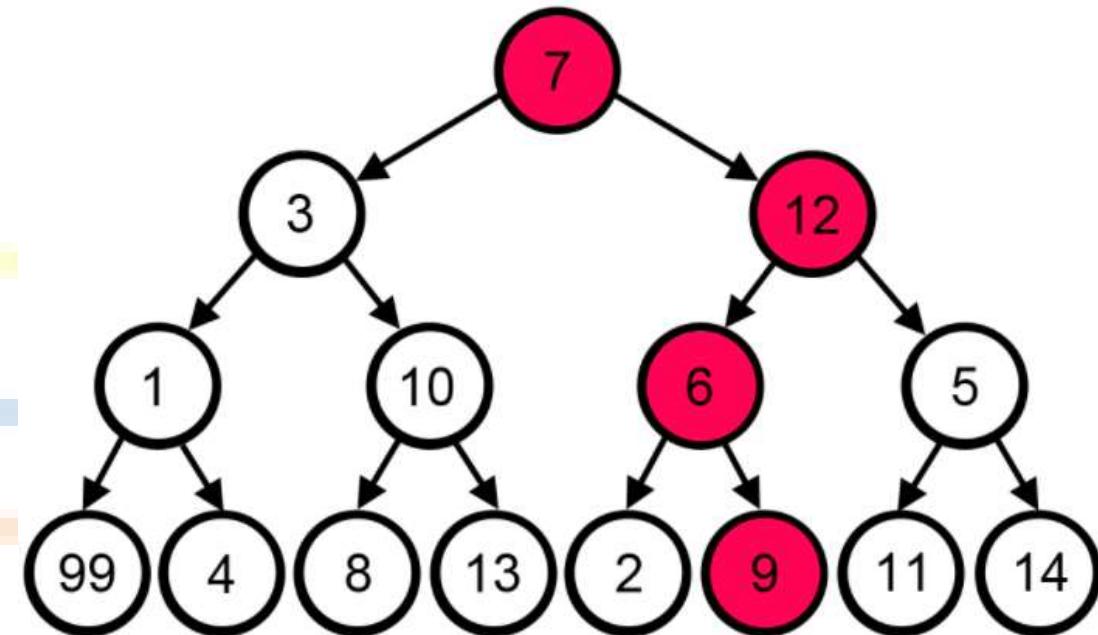
Processo de Aprendizado dos Algoritmos de Árvore de Decisão



Aprendizagem em Árvores de Decisão

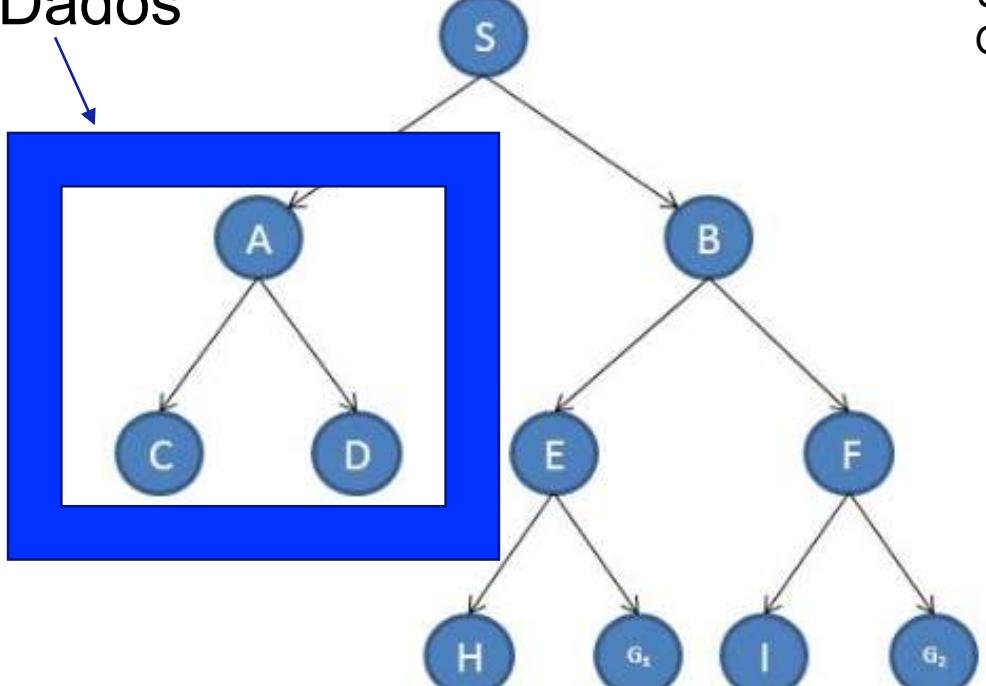
Greedy Search (Busca Gananciosa ou Gulosa)

O algoritmo procura maximizar o passo atual sem olhar para o passo seguinte, a fim de alcançar uma otimização global.



Aprendizagem em Árvores de Decisão

Partição
de Dados

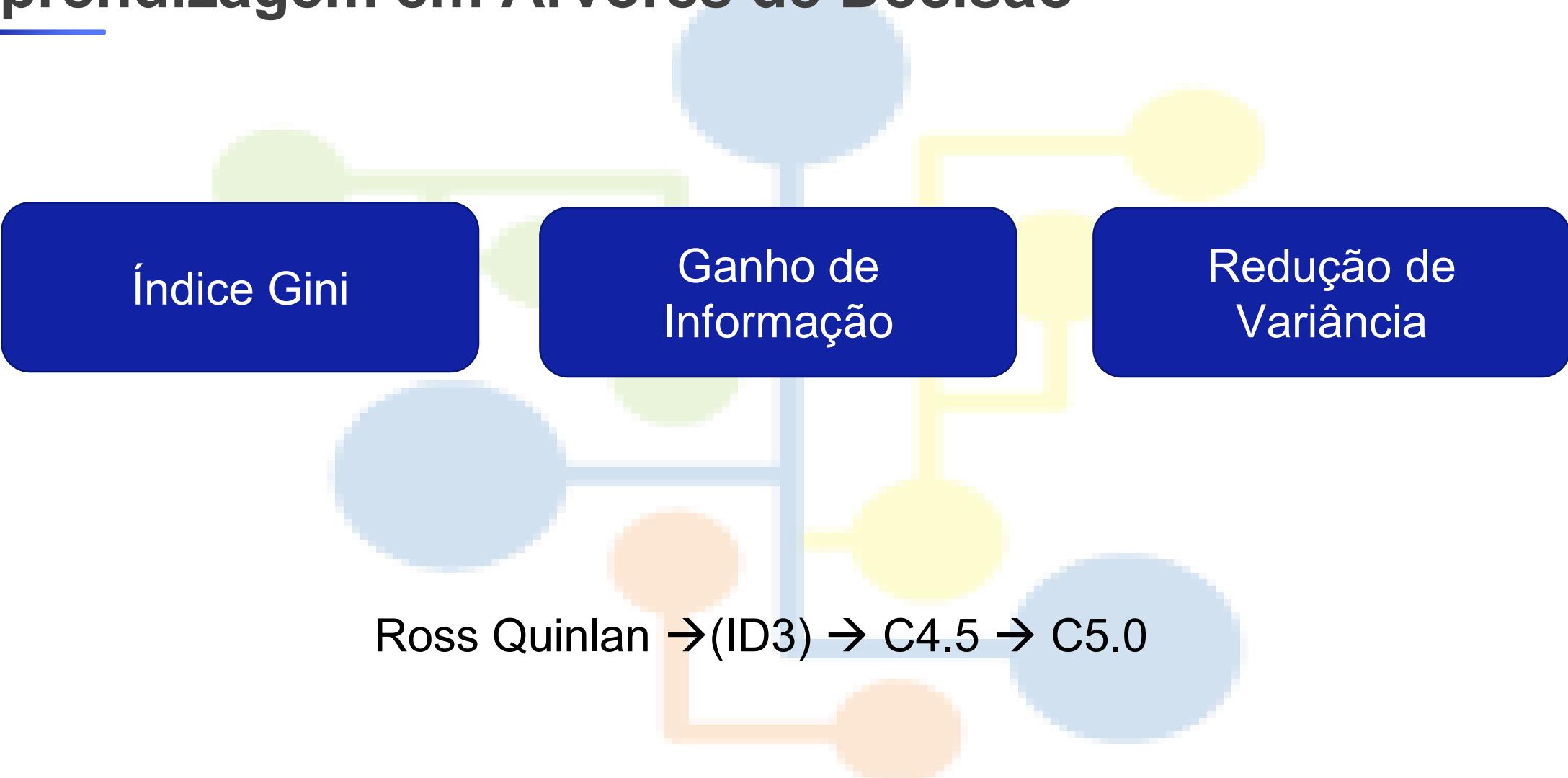


Greedy Search utiliza uma heurística estimada $h(n)$

$S \rightarrow$ Estado inicial
 $G1, G2 \rightarrow$ Objetivo

Node	$h(n)$
A	11
B	5
C	9
D	8
E	4
F	2
H	7
i	3

Aprendizagem em Árvores de Decisão



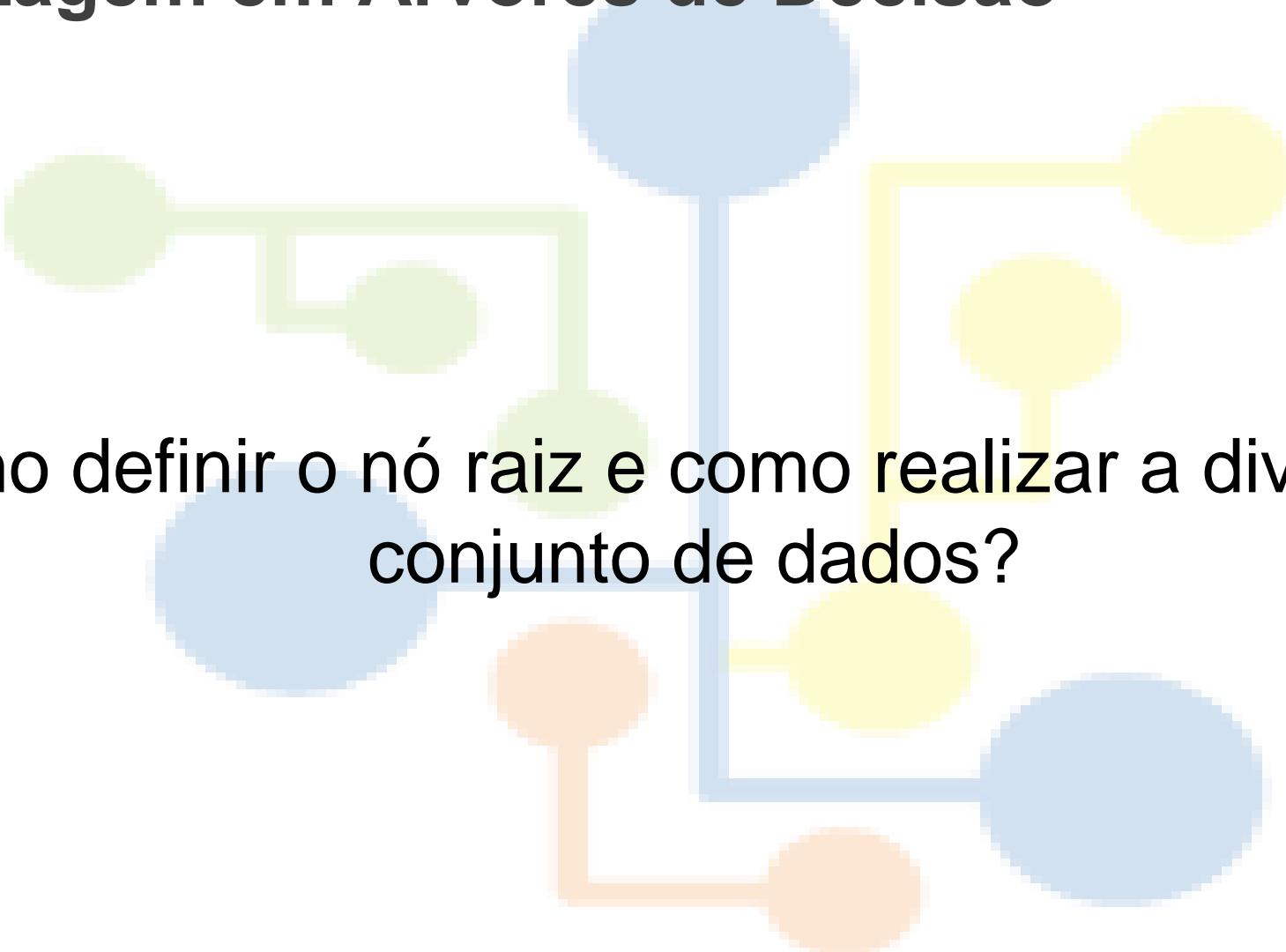
Índice Gini

Ganho de
Informação

Redução de
Variância

Ross Quinlan →(ID3) → C4.5 → C5.0

Aprendizagem em Árvores de Decisão



Como definir o nó raiz e como realizar a divisão do conjunto de dados?

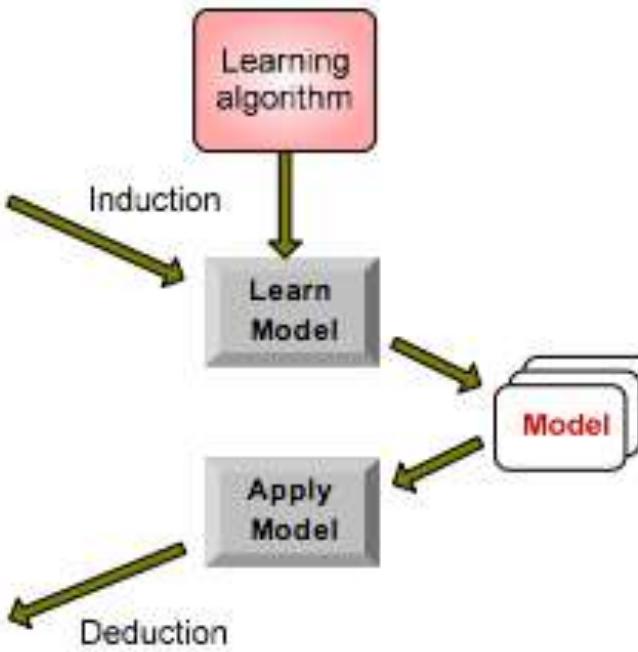
Aprendizagem em Árvores de Decisão

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	80K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

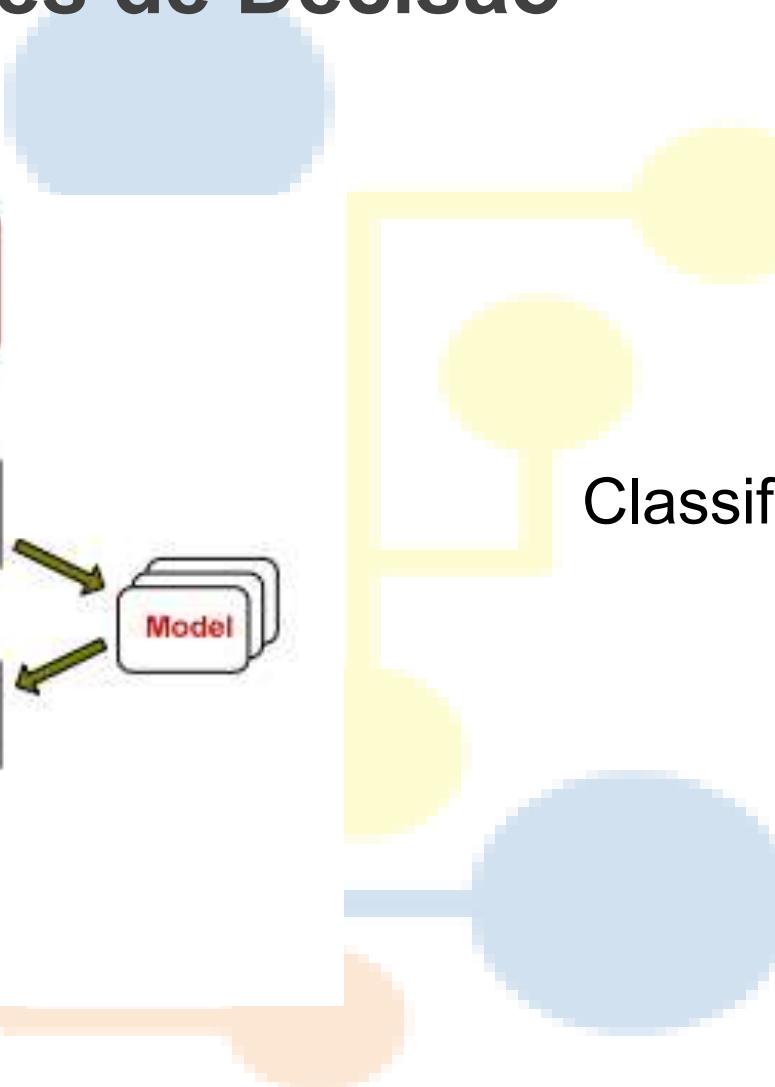
Training Set

TId	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	65K	?
15	No	Large	67K	?

Test Set



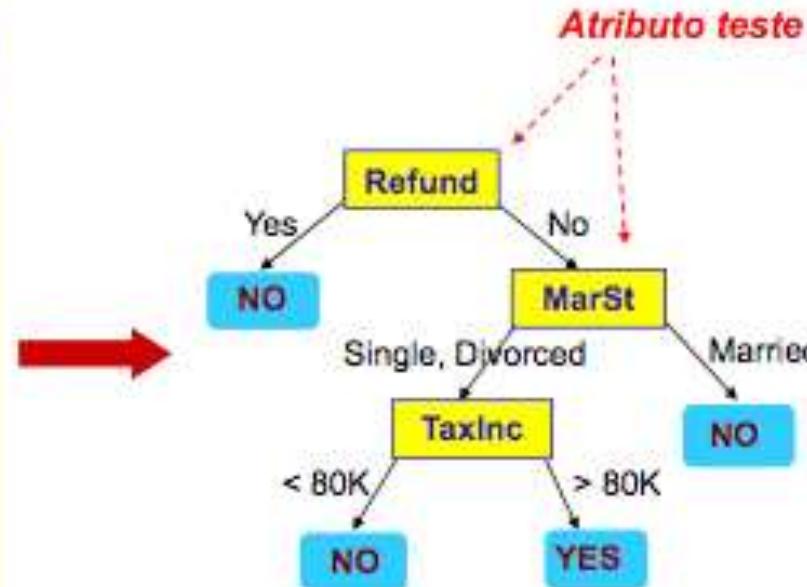
Classificação



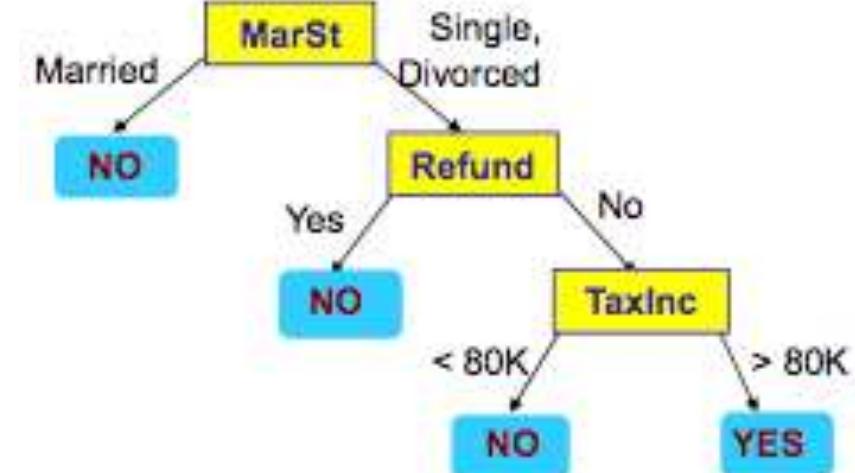
Aprendizagem em Árvores de Decisão

Tid	Refund	Marital Status	Taxable Income	Cheat	categorical	categorical	continuous	class
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 treinamento



Modelo: árvore de decisão



Pode haver mais de um árvore para o mesmo conjunto de dados

Aprendizagem em Árvores de Decisão

Como definir o nó raiz e como realizar a divisão do conjunto de dados?

- Estratégia Gulosa (Greedy Selection)
- Divisão baseada em atributos nominais
 - Divisão Binária
 - Divisão Múltipla
- Divisão baseada em atributos contínuos
 - Decisão Binária
 - Discretização
 - Estática
 - Dinâmica

Aprendizagem em Árvores de Decisão

Como definir o nó raiz e como realizar a divisão do conjunto de dados?

Estratégia Gulosa (Greedy Selection)

Necessita da medida da “impureza” do nó

C0: 5
C1: 5

Não-homogênea,
Alto grau de impureza

C0: 9
C1: 1

Homogêneo,
baixo grau de impureza

Aprendizagem em Árvores de Decisão

Como definir o nó raiz e como realizar a divisão do conjunto de dados?

Estratégia Gulosa (Greedy Selection)

Necessita da medida da “impureza” do nó

C0: 5
C1: 5

Não-homogênea,
Alto grau de impureza

C0: 9
C1: 1

Homogêneo,
baixo grau de impureza

- Entropia
- Índice de Gini
- Erro de Classificação

Aprendizagem em Árvores de Decisão

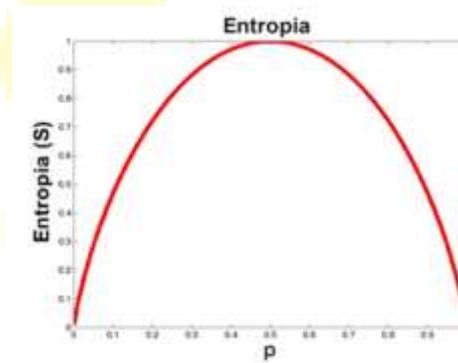
Entropia é a medida da incerteza nos dados

Ganho de Informação é a redução da Entropia

Aprendizagem em Árvores de Decisão

Entropia

$$\text{Entropy} = \sum -p_i \log_2 p_i$$



Entropia máxima considerando duas classes com a mesma probabilidade (distribuição 50/50):

$$\text{Entropy} = -0.5 \log_2 (0.5) - 0.5 \log_2 (0.5) = 1.0$$

Entropia considerando duas classes com distribuição 40/60:

$$\text{Entropy} = -0.4 \log_2 (0.4) - 0.6 \log_2 (0.6) = 0.97$$

Aprendizagem em Árvores de Decisão

Importante!

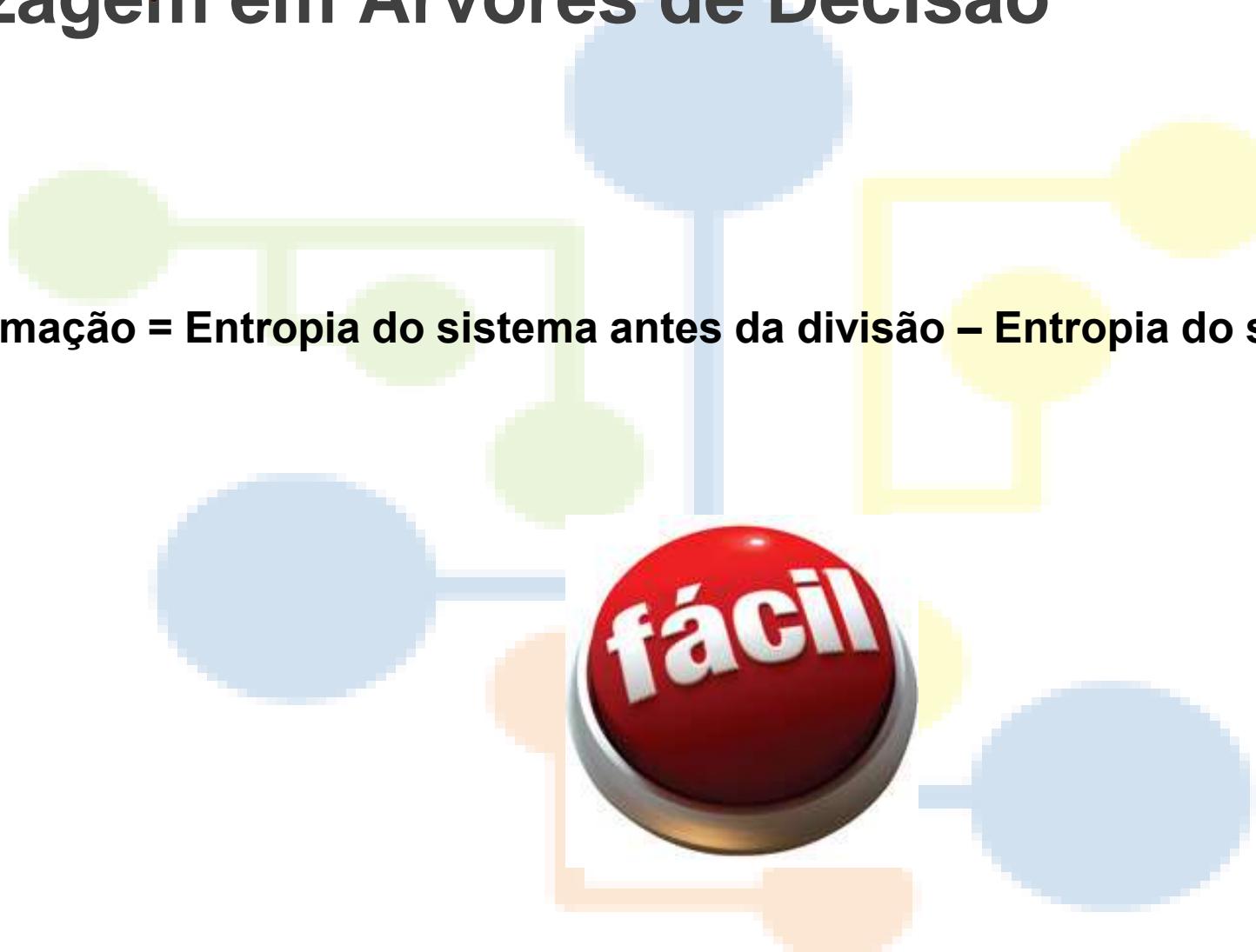


Nos algoritmos ID3, C4.5 e C5.0, o nó raiz é escolhido com base em quanto do total da Entropia é reduzido, se aquele nó é escolhido

Isso é chamado de Ganho de Informação!

Aprendizagem em Árvores de Decisão

Ganho de Informação = Entropia do sistema antes da divisão – Entropia do sistema após a divisão



Aprendizagem em Árvores de Decisão



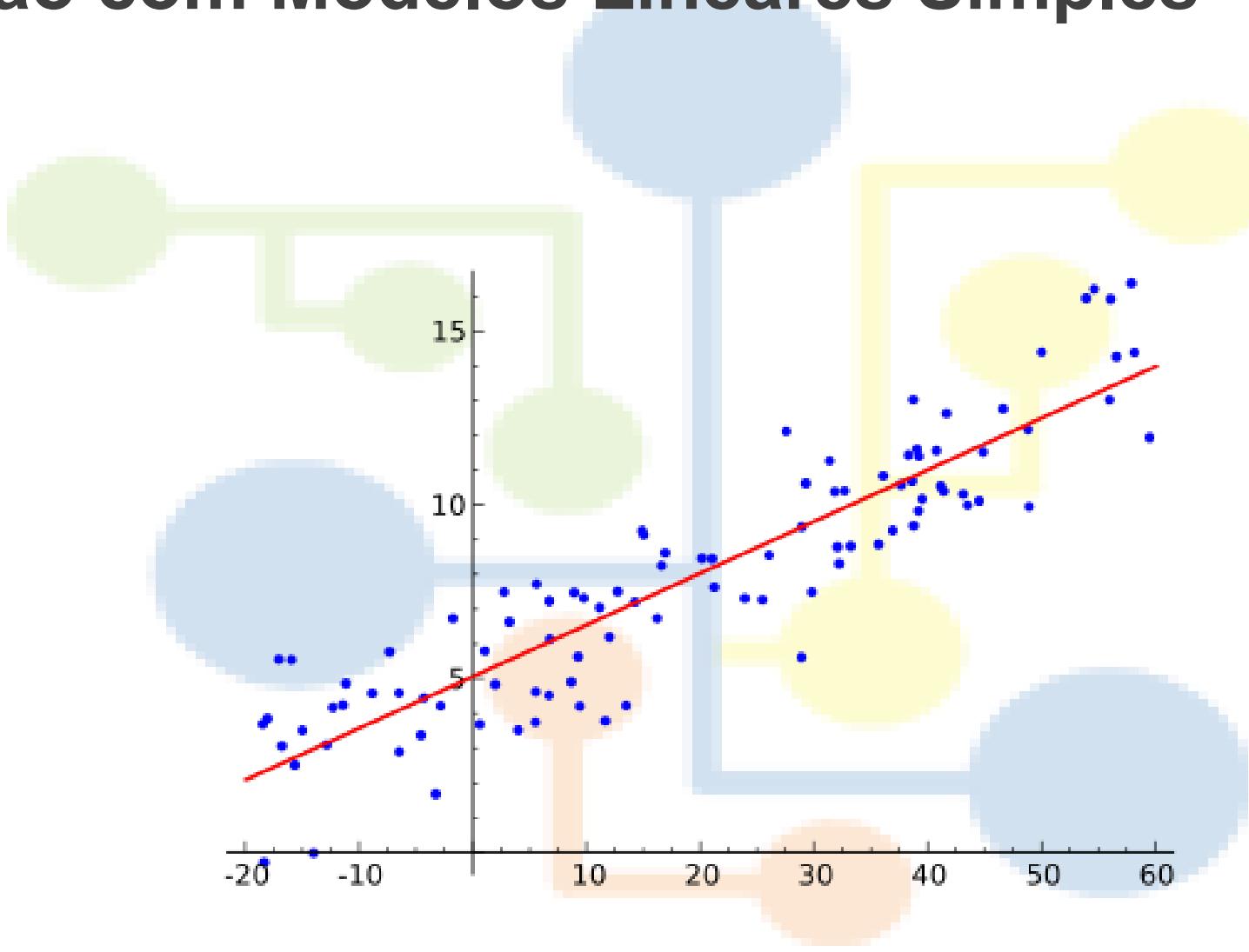
Esta metodologia (Entropia) é aplicada para computar o ganho de informação para todos os atributos. É escolhido o atributo com o **mais alto ganho de informação**. Isso é testado para cada nó a fim de escolher o melhor nó.





Regressão com Modelos Lineares Simples

Regressão com Modelos Lineares Simples



Regressão com Modelos Lineares Simples

Aprovação de Crédito de um Indivíduo

Atributo	Valor
Sexo	Masculino
Idade	37
Salário Mensal	R\$ 15.000,00
Anos no Emprego Atual	3
Anos de Residência	7
Saldo Bancário	R\$ 43.671,94

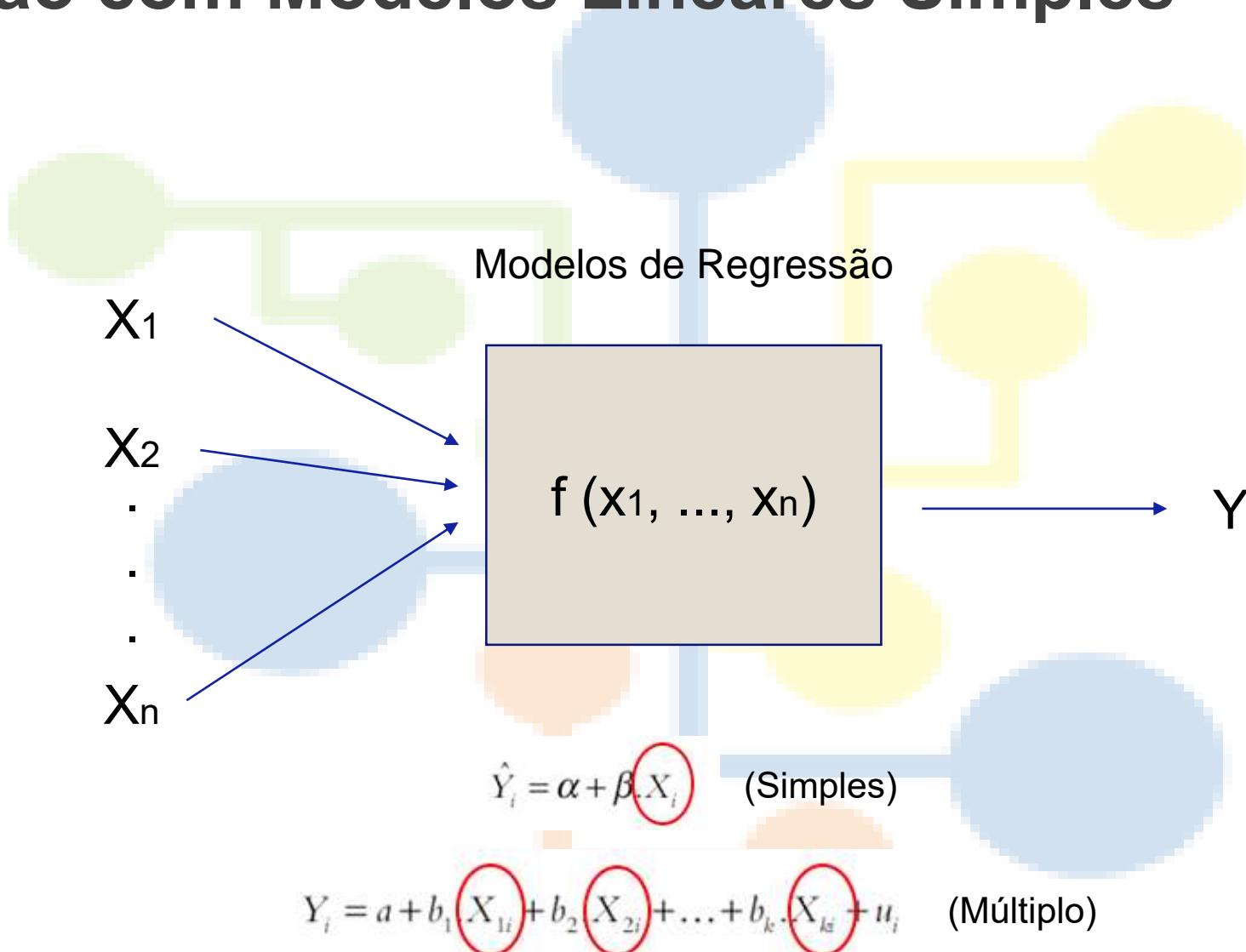
Classificação

- Decisão de crédito (Sim/Não)

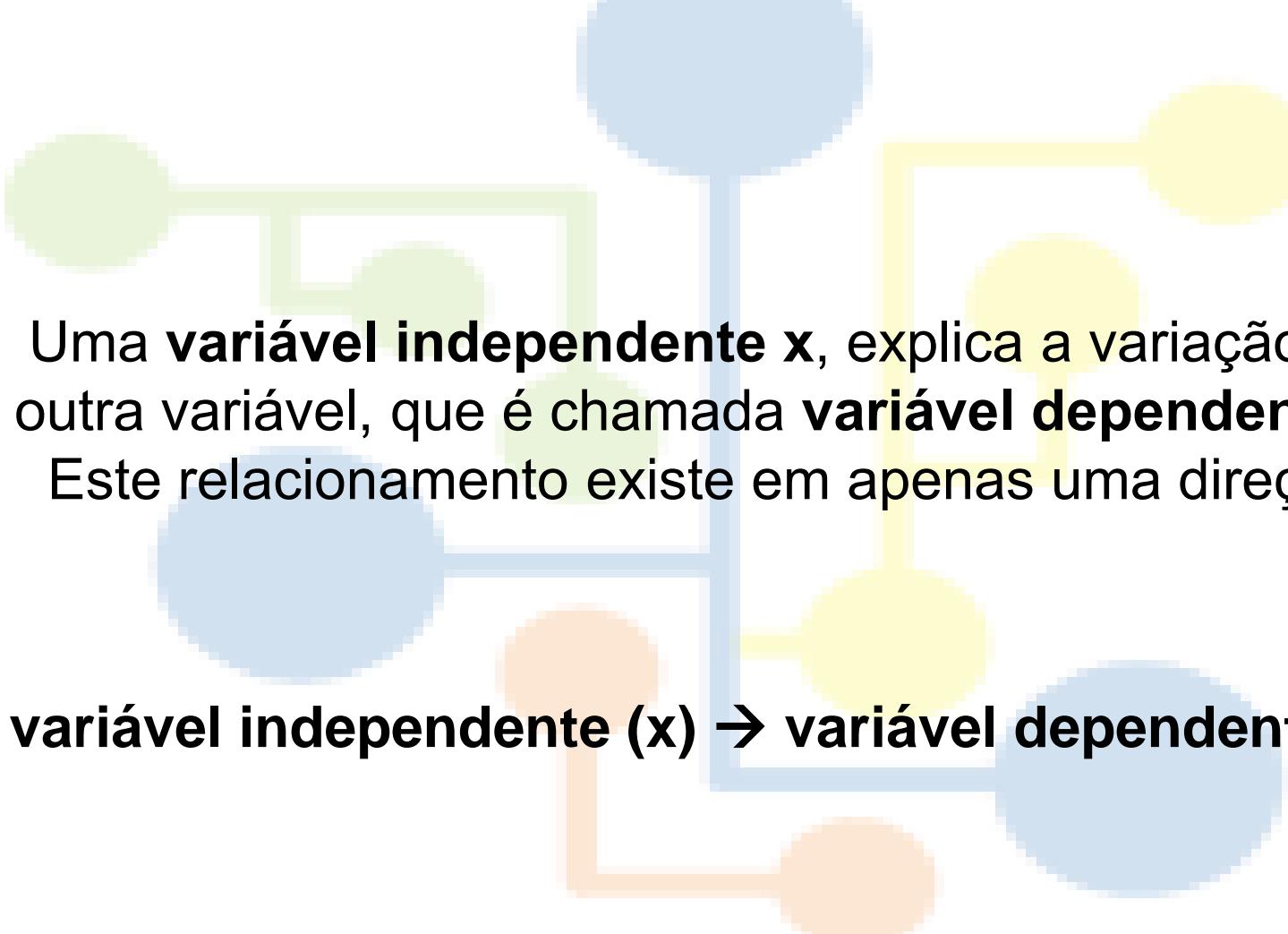
Regressão

- Quantidade de crédito (dinheiro)

Regressão com Modelos Lineares Simples



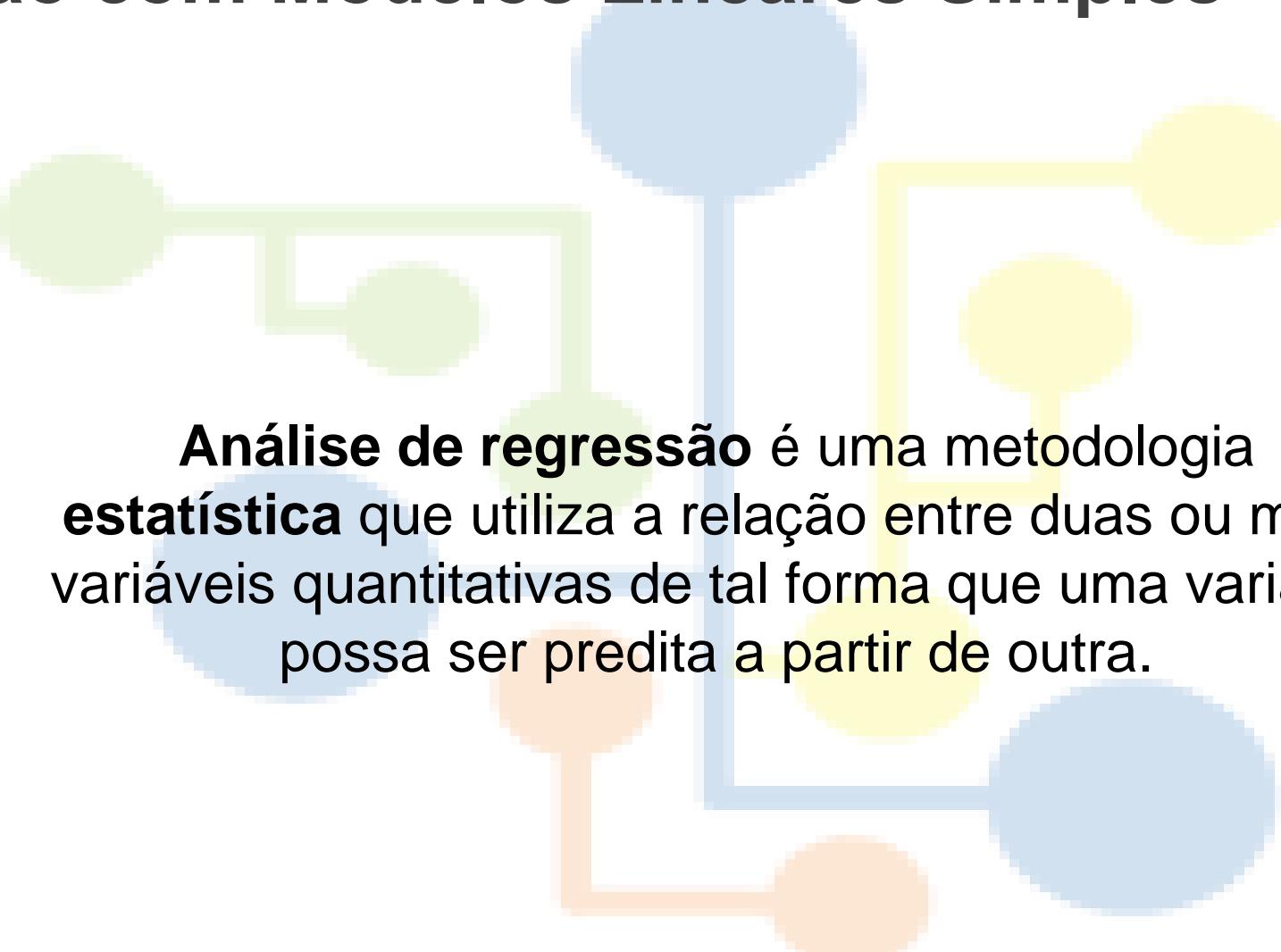
Regressão com Modelos Lineares Simples



Uma **variável independente** x, explica a variação em outra variável, que é chamada **variável dependente** y. Este relacionamento existe em apenas uma direção:

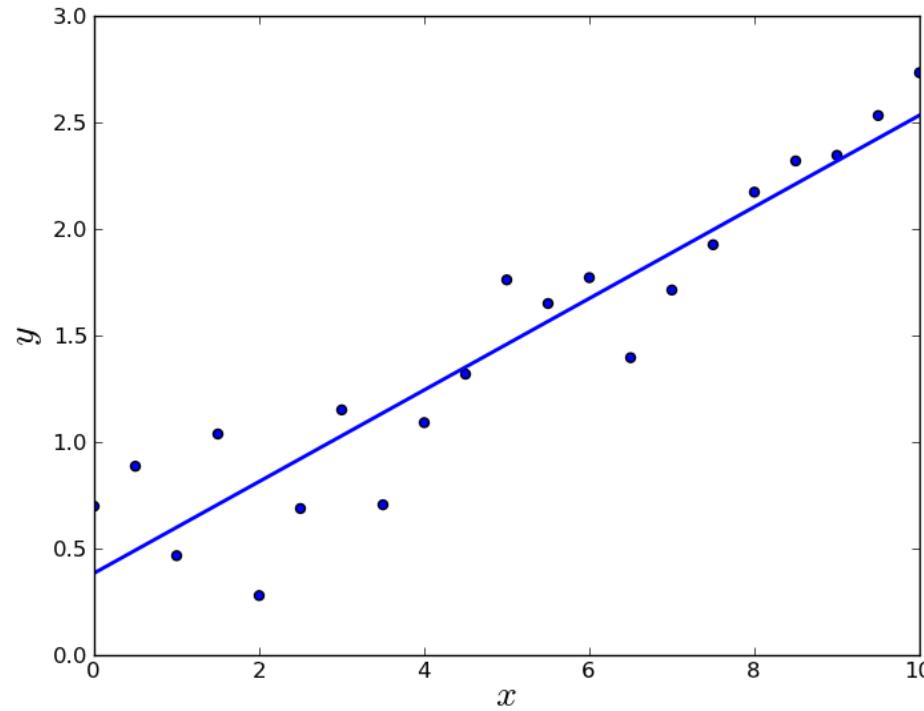
variável independente (x) → variável dependente (y)

Regressão com Modelos Lineares Simples



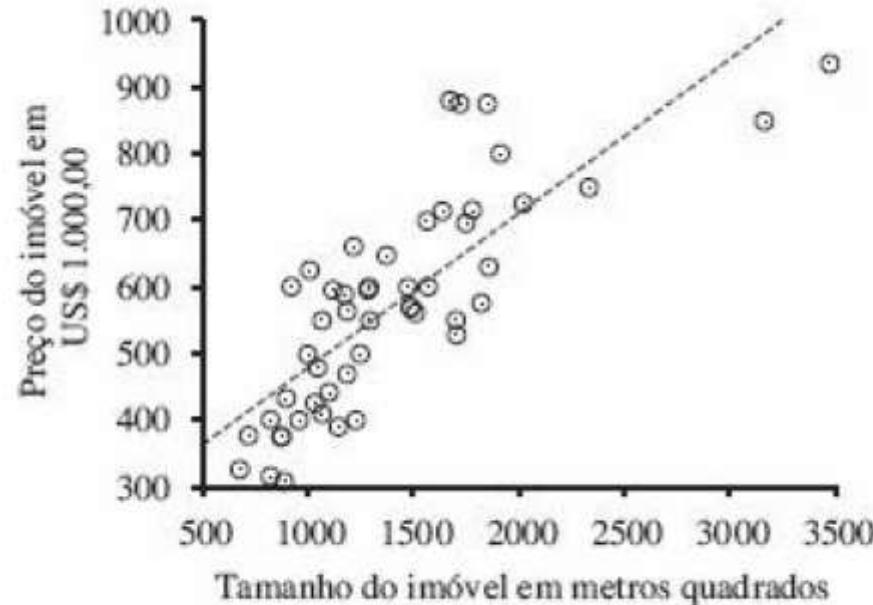
Análise de regressão é uma metodologia **estatística** que utiliza a relação entre duas ou mais variáveis quantitativas de tal forma que uma variável possa ser predita a partir de outra.

Regressão com Modelos Lineares Simples



$$h_w(x) = w_1x + w_0$$

Regressão com Modelos Lineares Simples

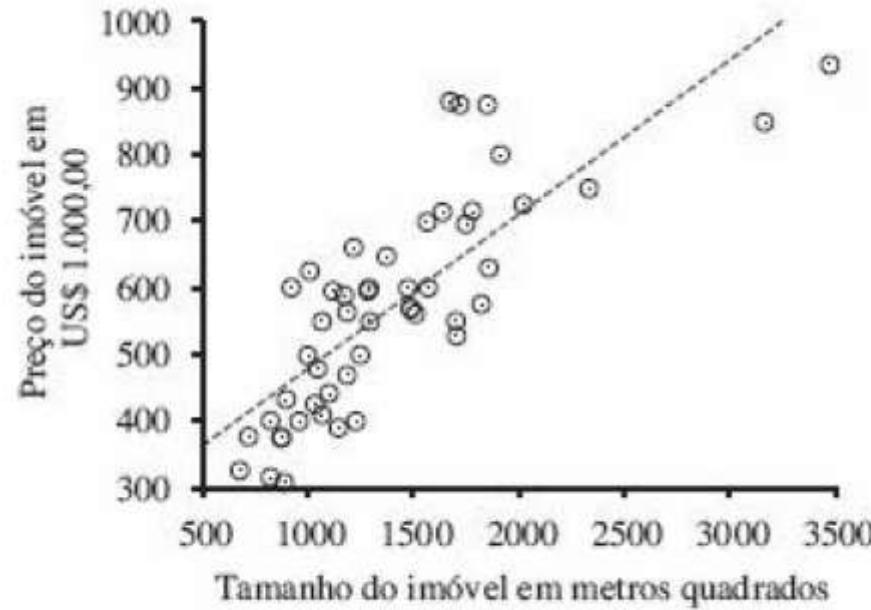


(a)

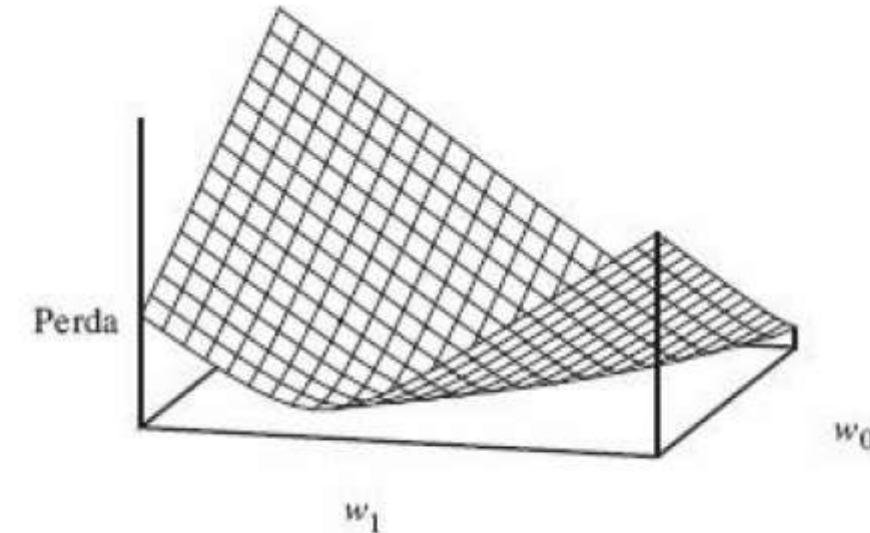
A tarefa de encontrar o **hw** que melhor se encaixe nesses dados é chamada de regressão linear.

$$\text{Perda}(h_w) = \sum_{j=1}^N L_2(y_j, h_w(x_j)) = \sum_{j=1}^N (y_j - h_w(x_j))^2 = \sum_{j=1}^N (y_j - (w_1 x_j + w_0))^2$$

Regressão com Modelos Lineares Simples



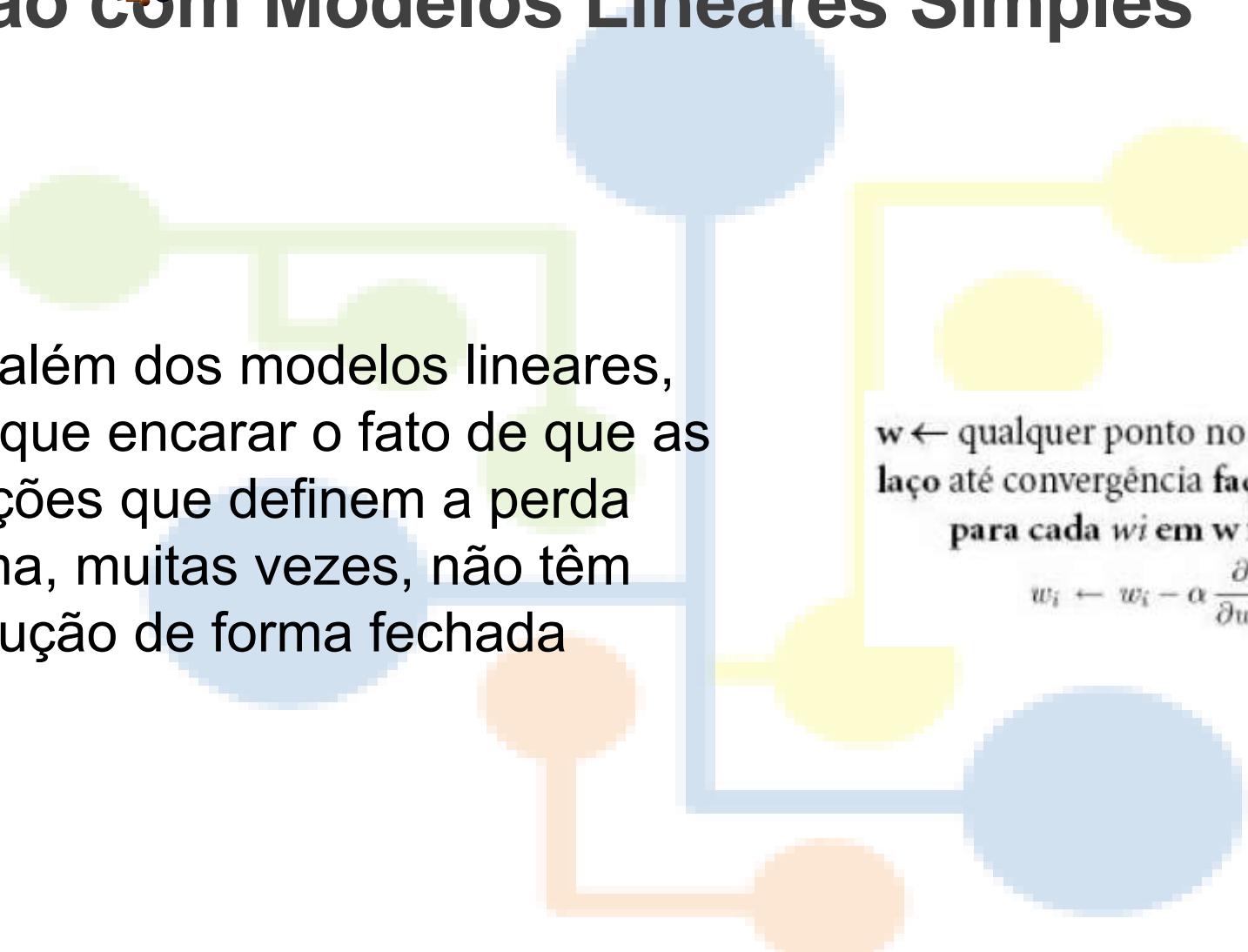
(a)



(b)

$$Perda(h_w) = \sum_{j=1}^N L_2(y_j, h_w(x_j)) = \sum_{j=1}^N (y_j - h_w(x_j))^2 = \sum_{j=1}^N (y_j - (w_1 x_j + w_0))^2$$

Regressão com Modelos Lineares Simples



Para ir além dos modelos lineares, teremos que encarar o fato de que as equações que definem a perda mínima, muitas vezes, não têm solução de forma fechada

w ← qualquer ponto no espaço de parâmetros
laço até convergência **faça**
 para cada w_i em **w** **faça**

$$w_i \leftarrow w_i - \alpha \frac{\partial}{\partial w_i} \text{Perda}(w)$$

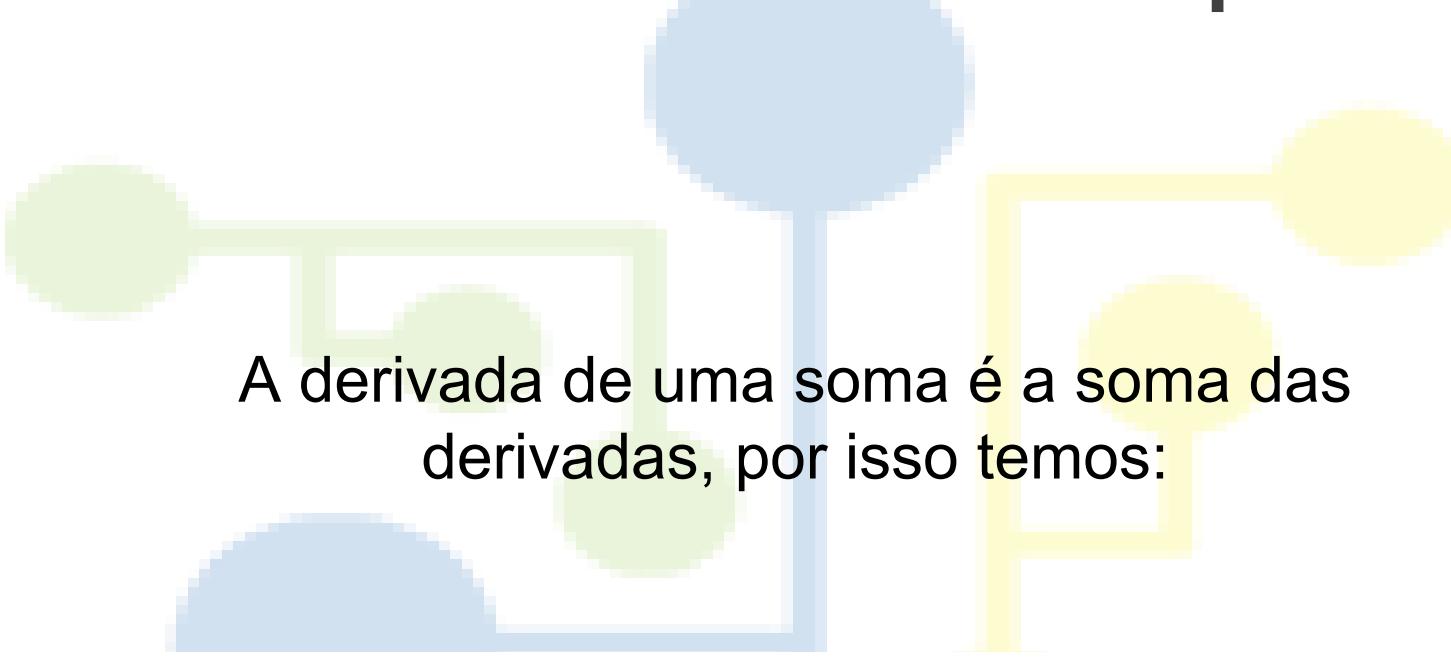
Regressão com Modelos Lineares Simples

O parâmetro, que chamamos de tamanho do passo, é geralmente chamado de **taxa de aprendizagem** quando estamos tentando minimizar a perda em um problema de aprendizagem.

$w \leftarrow$ qualquer ponto no espaço de parâmetros
laço até convergência faça
para cada w_i em w faça

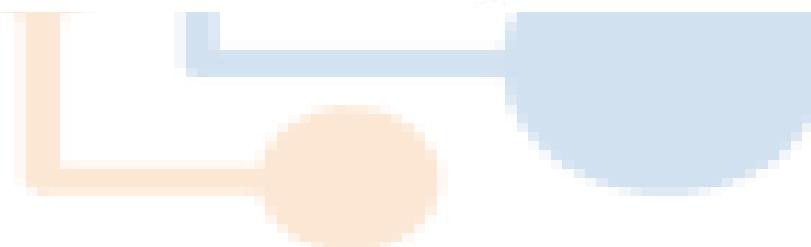
$$w_i \leftarrow w_i - \alpha \frac{\partial}{\partial w_i} Perda(w)$$

Regressão com Modelos Lineares Simples

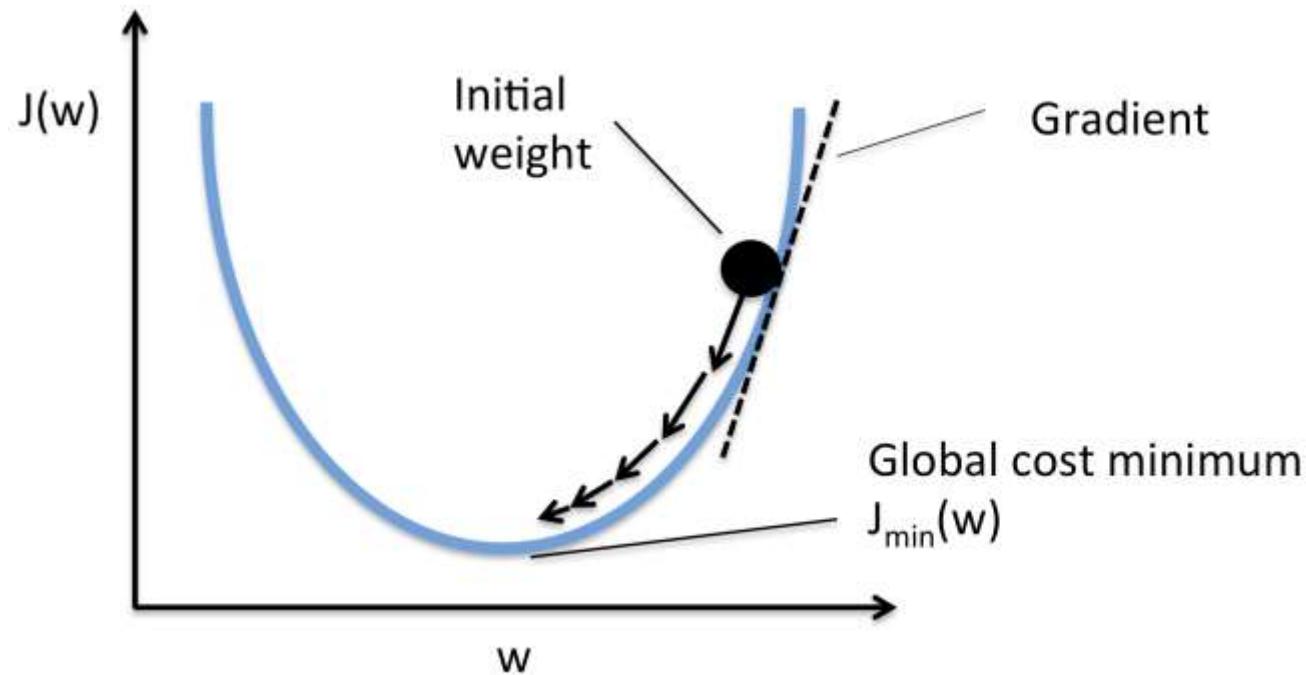


A derivada de uma soma é a soma das derivadas, por isso temos:

$$w_0 \leftarrow w_0 + \alpha \sum_j (y_j - h_{\mathbf{w}}(x_j)); \quad w_1 \leftarrow w_1 + \alpha \sum_j (y_j - h_{\mathbf{w}}(x_j)) \times x_j$$



Regressão com Modelos Lineares Simples

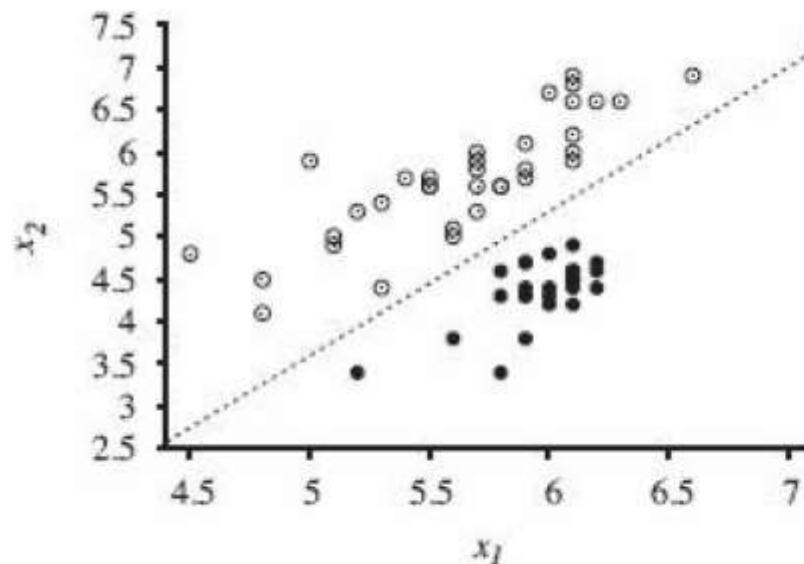


Essas atualizações constituem a regra de aprendizagem da descida pelo gradiente em lotes para regressão linear simples

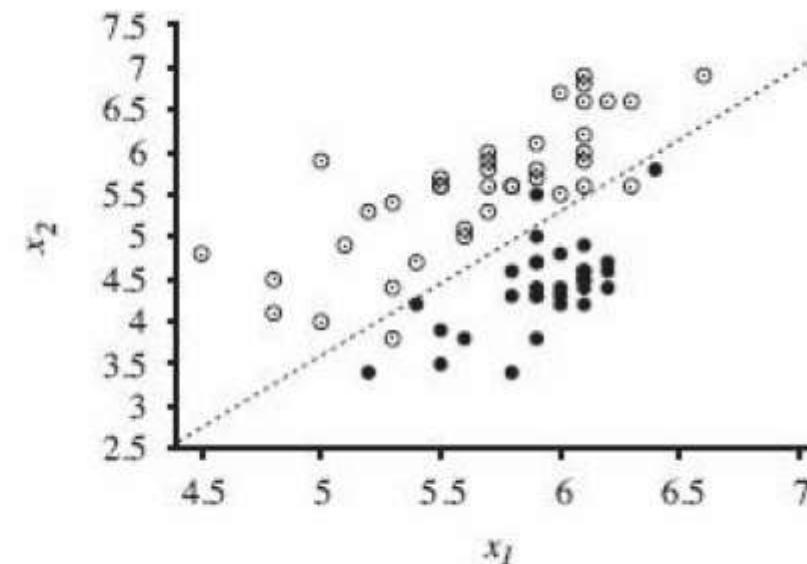


Classificação com Modelos Lineares

Classificação com Modelos Lineares

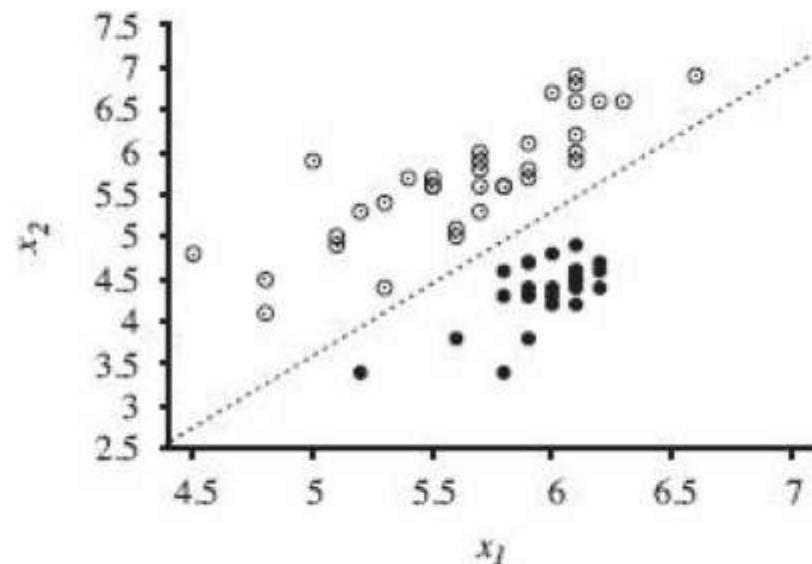


(a)

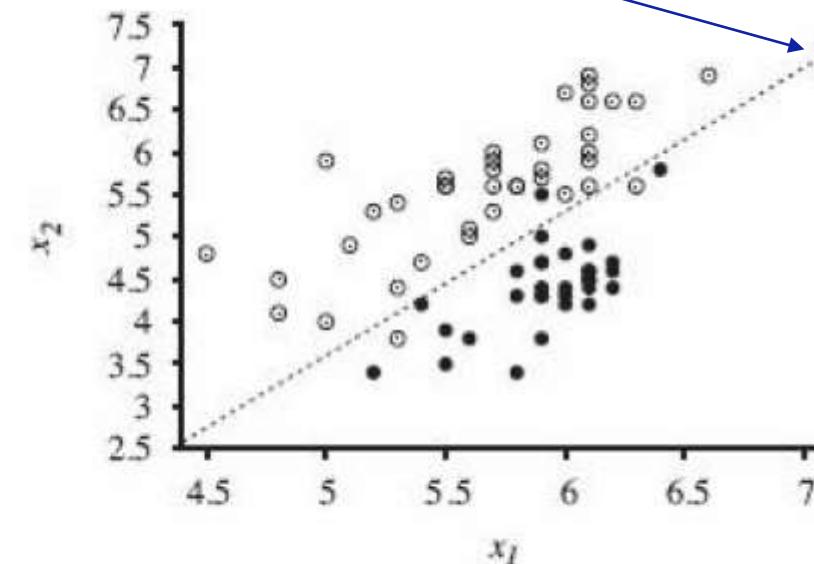


(b)

Classificação com Modelos Lineares

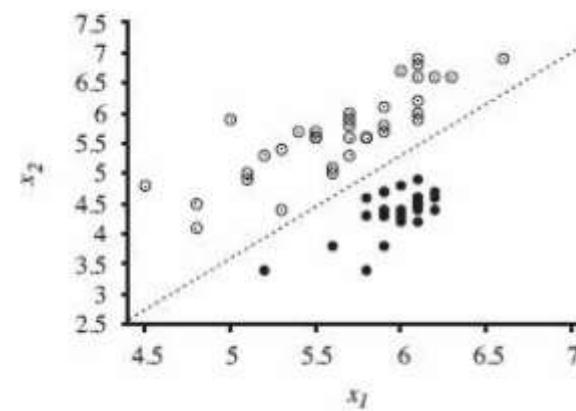


(a)

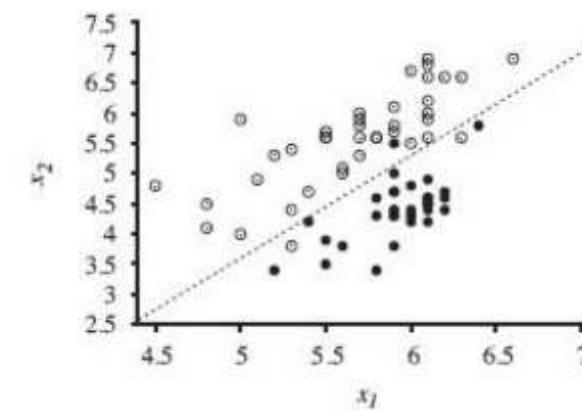


(b)

Classificação com Modelos Lineares



(a)

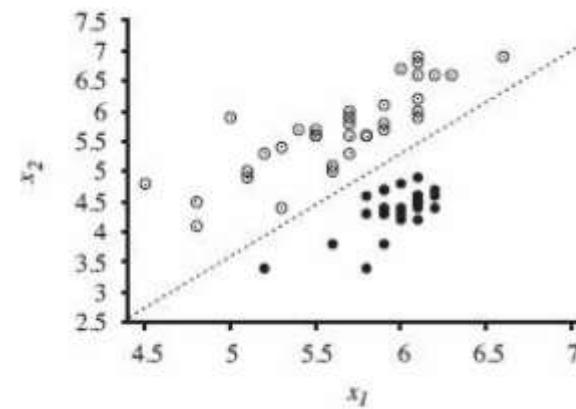


(b)

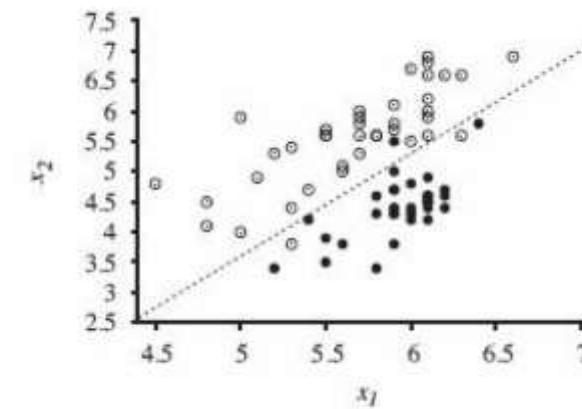
$h_w(x) = 1$ se $w \cdot x \geq 0$ e 0 caso contrário

$h_w(x) = \text{Limiar}(w \cdot x)$, onde $\text{Limiar}(z) = 1$ se $z \geq 0$ e 0 caso contrário

Classificação com Modelos Lineares



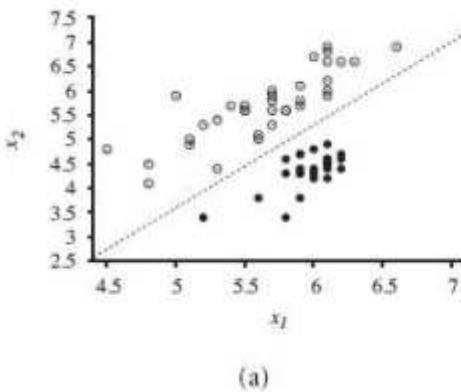
(a)



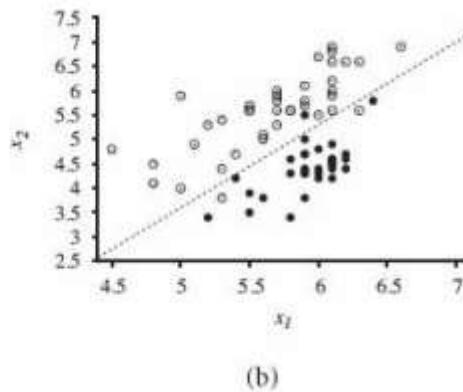
(b)

$$w_i \leftarrow w_i + \alpha (y - h_w(\mathbf{x})) \times x_i$$

Classificação com Modelos Lineares



(a)

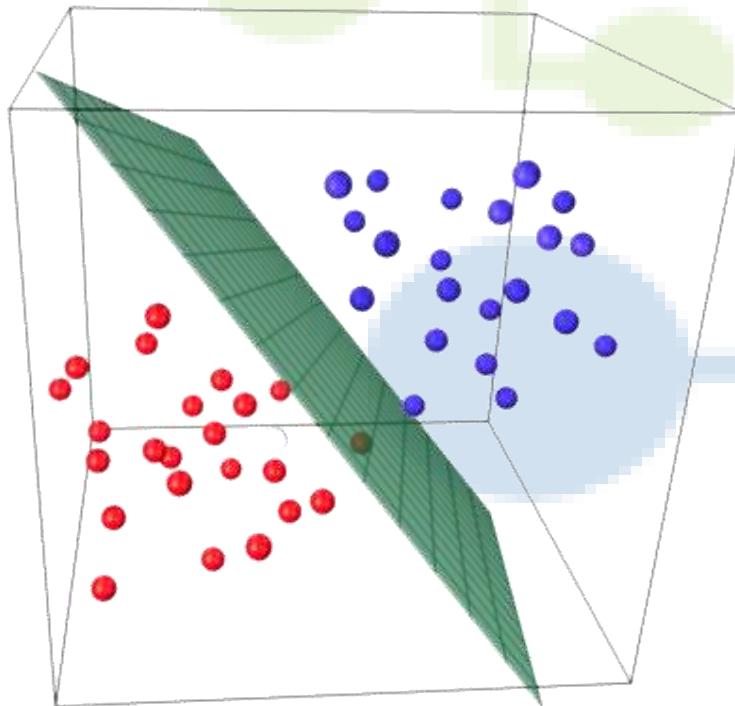


(b)

$$w_i \leftarrow w_i + \alpha (y - h_w(x)) \times x_i$$

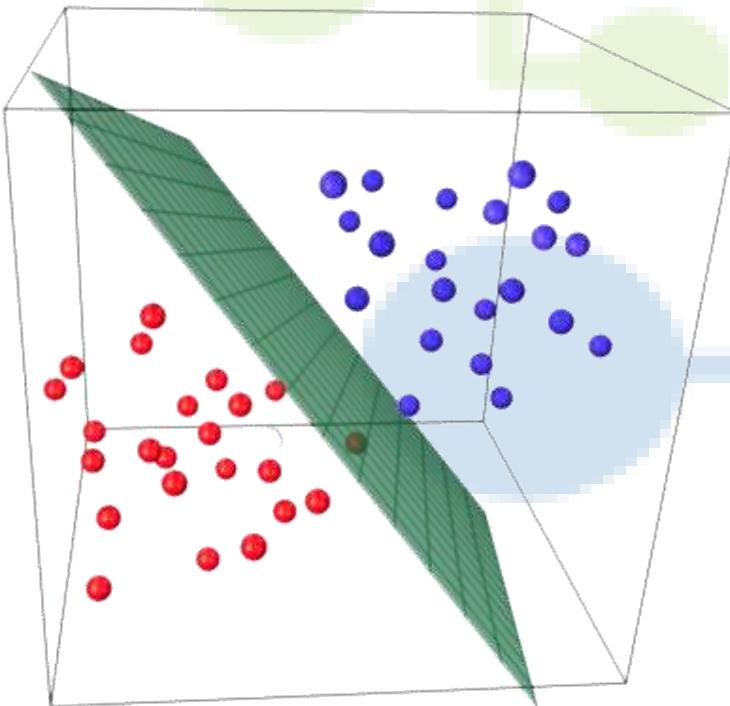
- Se a saída está correta, ou seja, $y = h_w(x)$, os pesos não são alterados.
- Se y for 1, mas $h_w(x)$ for 0, w_i será aumentado quando a entrada correspondente x_i for positiva e diminuído quando x_i for negativo. Isso faz sentido porque queremos fazer $w \cdot x$ maior para que $h_w(x)$ gere um 1.
- Se y for 0, mas $h_w(x)$ for 1, w_i será diminuído quando a entrada correspondente x_i for positiva e aumentado quando x_i for negativo. Isso faz sentido porque queremos fazer $w \cdot x$ menor para que $h_w(x)$ gere um 0.

Classificação com Modelos Lineares



Além disso, o classificador linear sempre anuncia uma previsão completamente confiante de 1 ou 0, mesmo para exemplos que estão muito perto da fronteira (o que pode levar a classificações incorretas).

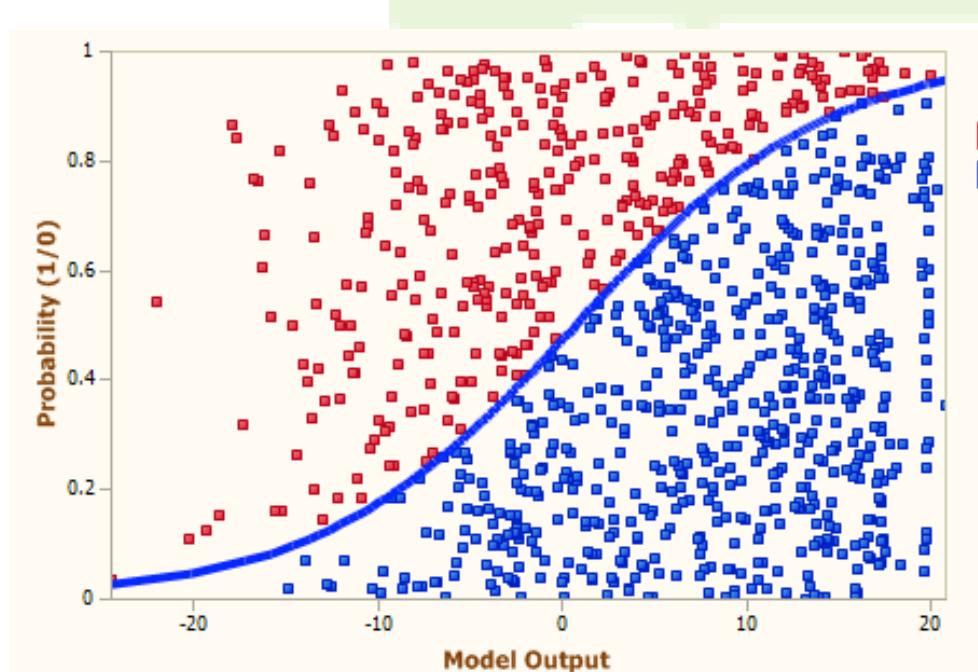
Classificação com Modelos Lineares



No modelo logístico a variável resposta é binária. Uma variável binária assume dois valores, como por exemplo, $Y = 0$ e $Y = 1$ denominados "fracasso" e "sucesso", respectivamente.

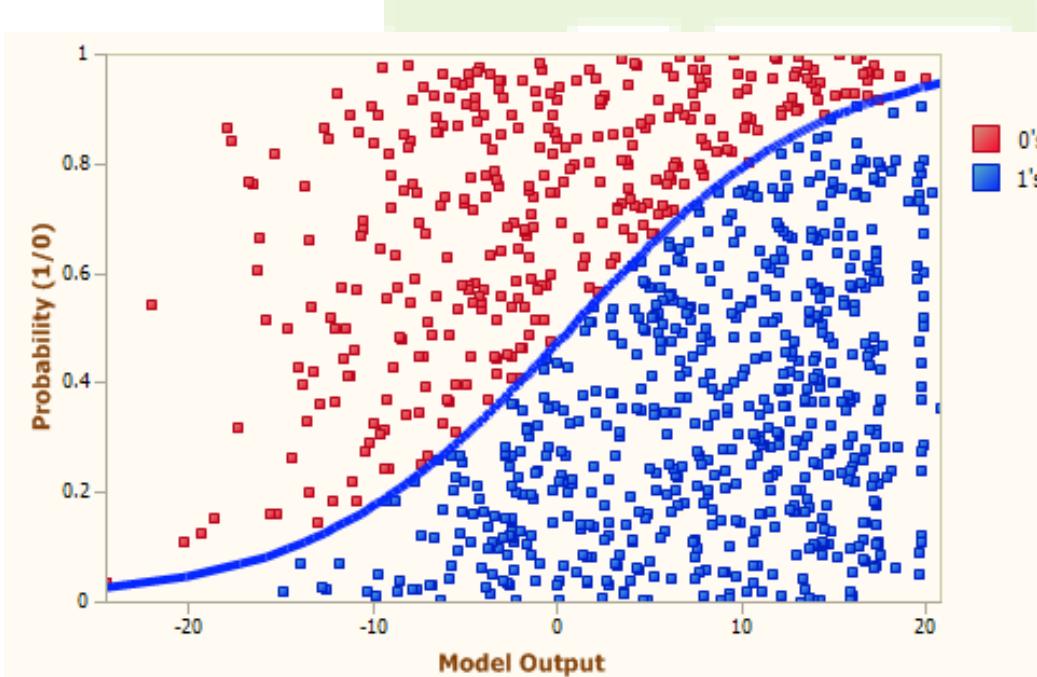
Neste caso, "sucesso" é o evento de interesse.

Classificação com Modelos Lineares



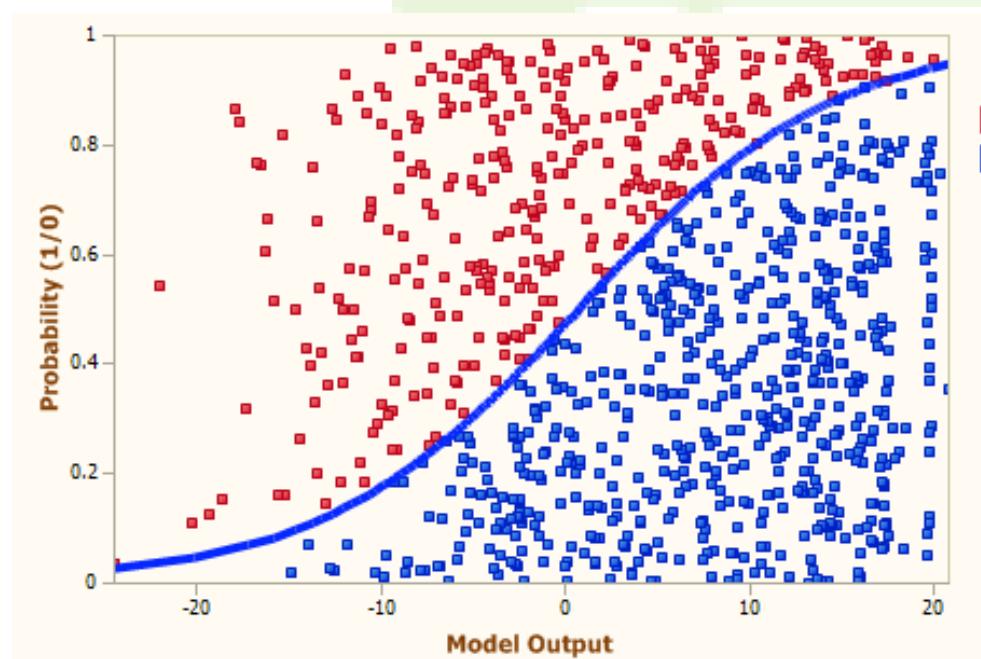
Os modelos de regressão constituem uma das ferramentas estatísticas mais importantes na análise estatística de dados, quando se pretende modelar relações entre variáveis.

Classificação com Modelos Lineares



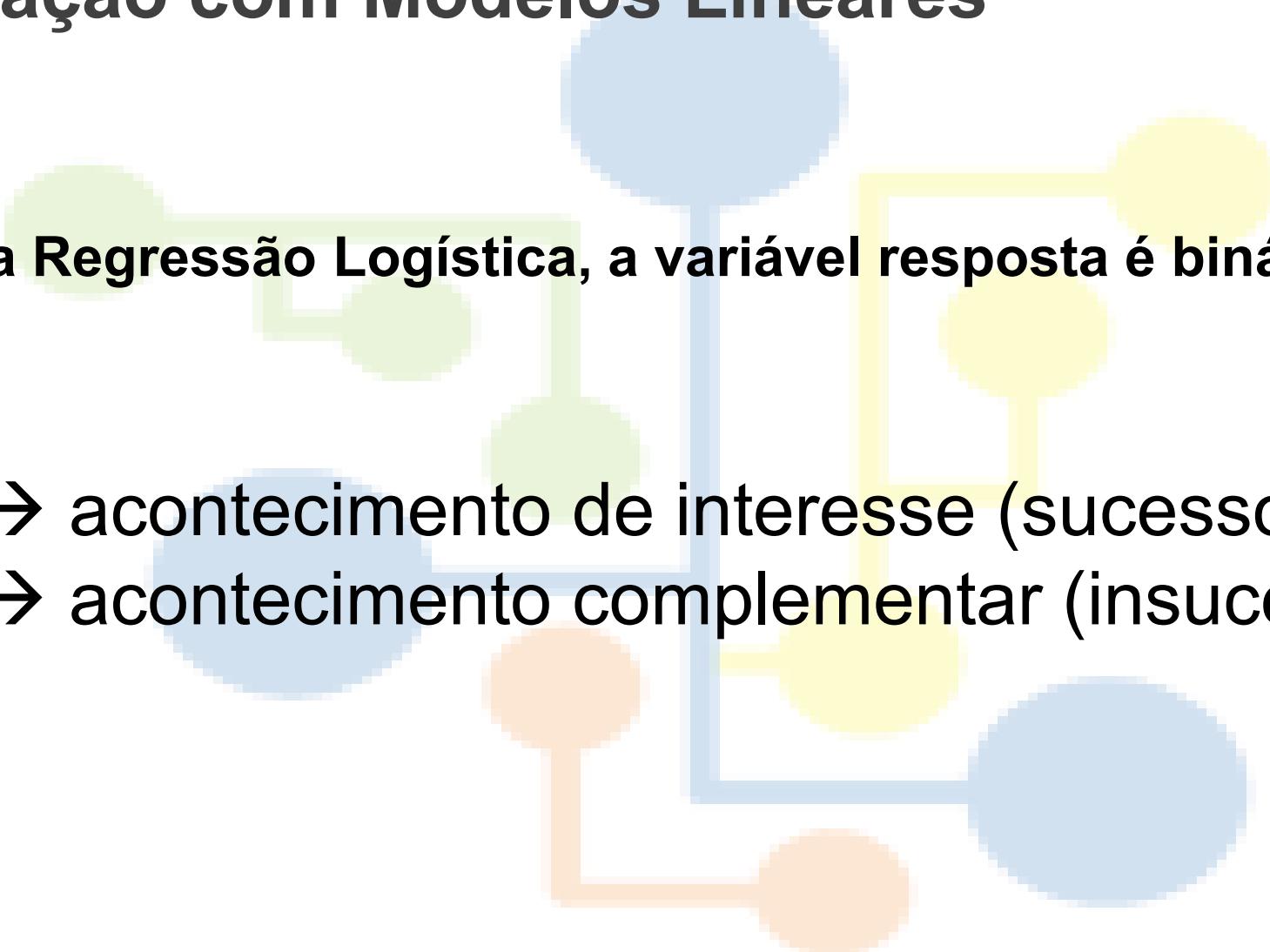
A regressão logística é uma técnica estatística que tem como objetivo modelar, a partir de um conjunto de observações, a relação “logística” entre uma variável resposta e uma série de variáveis explicativas numéricas (contínuas, discretas) e/ou categóricas.

Classificação com Modelos Lineares



A regressão logística é amplamente usada em ciências médicas e sociais, e tem outras denominações, como **modelo logístico**, **modelo logit**, e **classificador de máxima entropia**.

Classificação com Modelos Lineares



Na Regressão Logística, a variável resposta é binária

1 → acontecimento de interesse (sucesso)

0 → acontecimento complementar (insucesso)

Classificação com Modelos Lineares

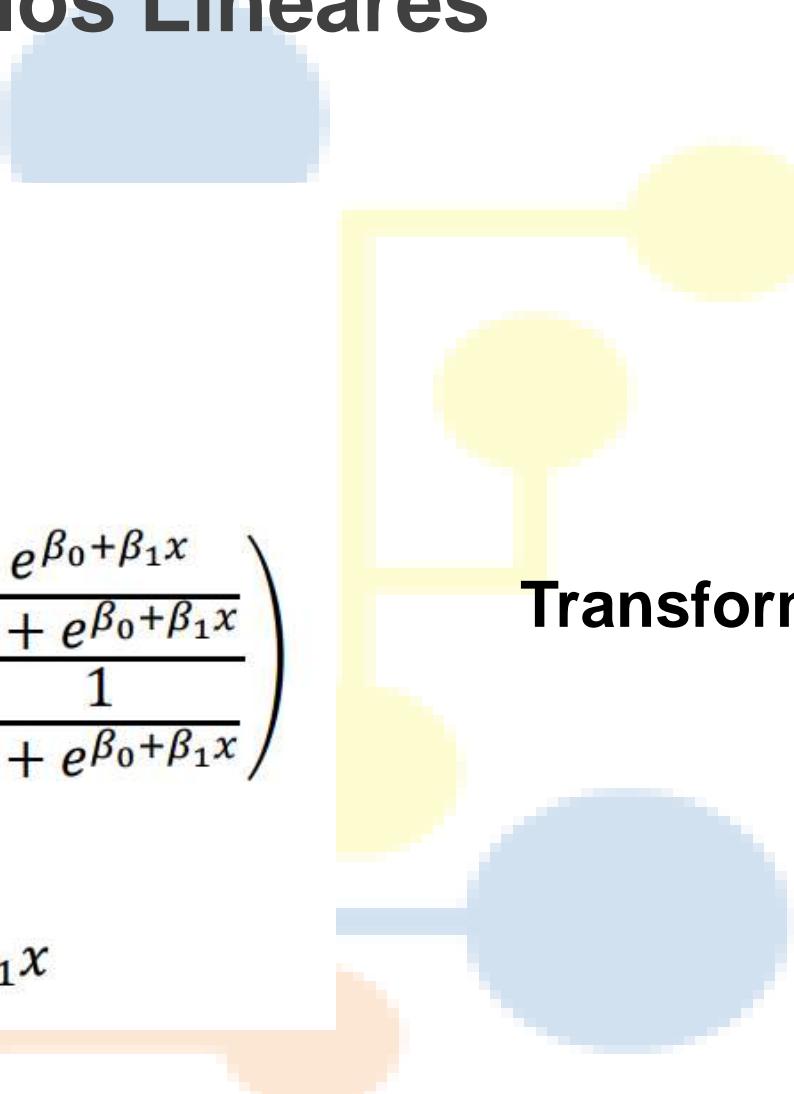
$$g(x) = \ln\left(\frac{\pi(x)}{1 - \pi(x)}\right)$$

$$g(x) = \ln\left(\frac{\frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}}{1 - \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}}\right) = \ln\left(\frac{\frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}}{\frac{1}{1 + e^{\beta_0 + \beta_1 x}}}\right)$$

$$g(x) = \ln(e^{\beta_0 + \beta_1 x}) = \beta_0 + \beta_1 x$$

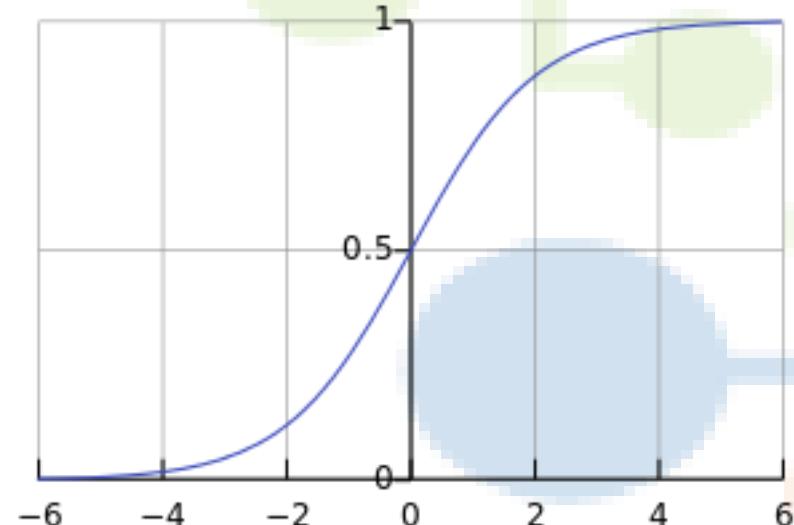


Logaritmo



Transformação logit

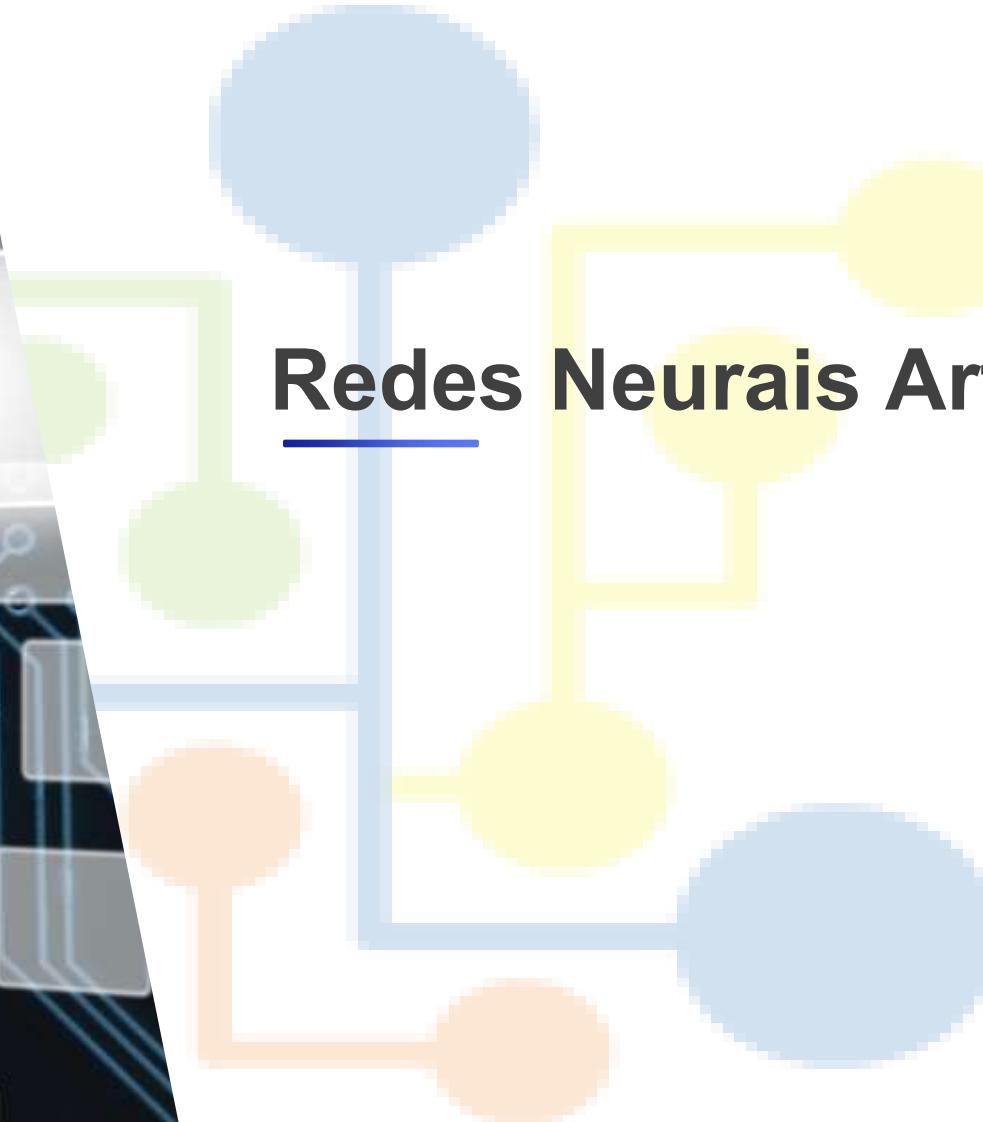
Classificação com Modelos Lineares

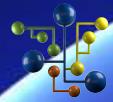


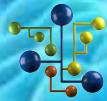
Regressão Logística é útil para modelar a probabilidade de um evento ocorrer como função de outros fatores. É um modelo linear generalizado que usa como função de ligação a função logit.



Redes Neurais Artificiais





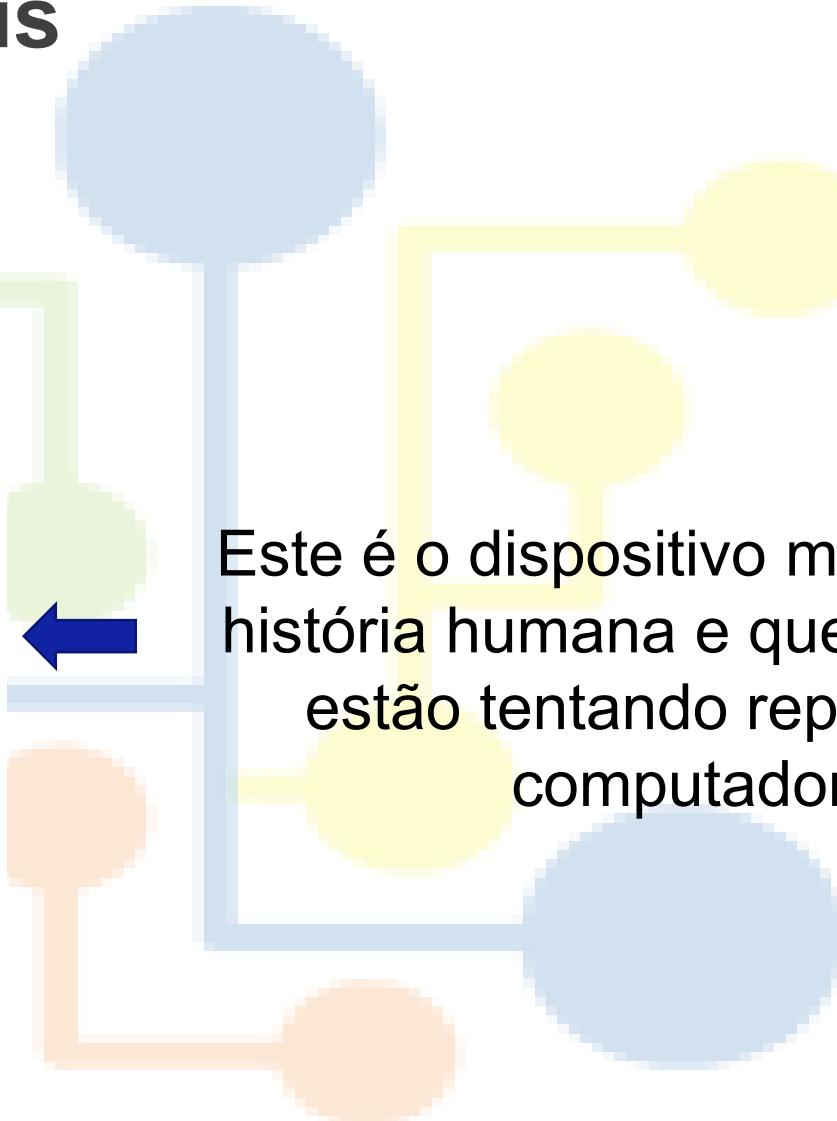
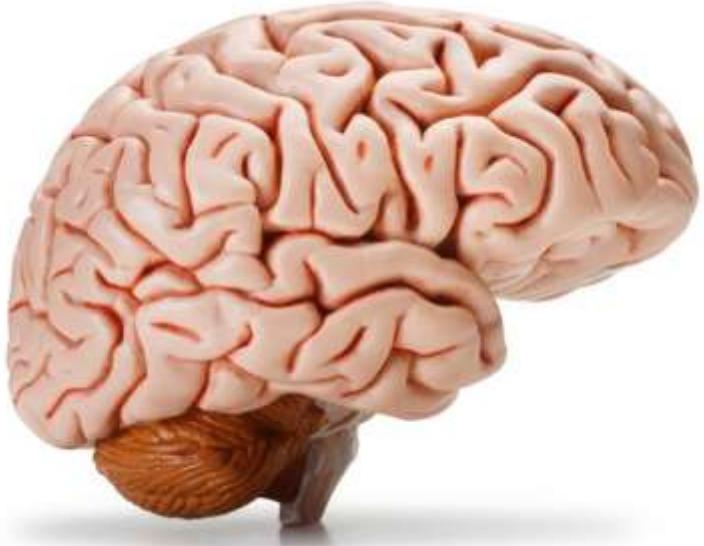


Redes Neurais Artificiais

Como resultados destas pesquisas surgiram o modelo do neurônio articial e posteriormente um sistema com vários neurônios interconectados, a chamada Rede Neural Artificial.

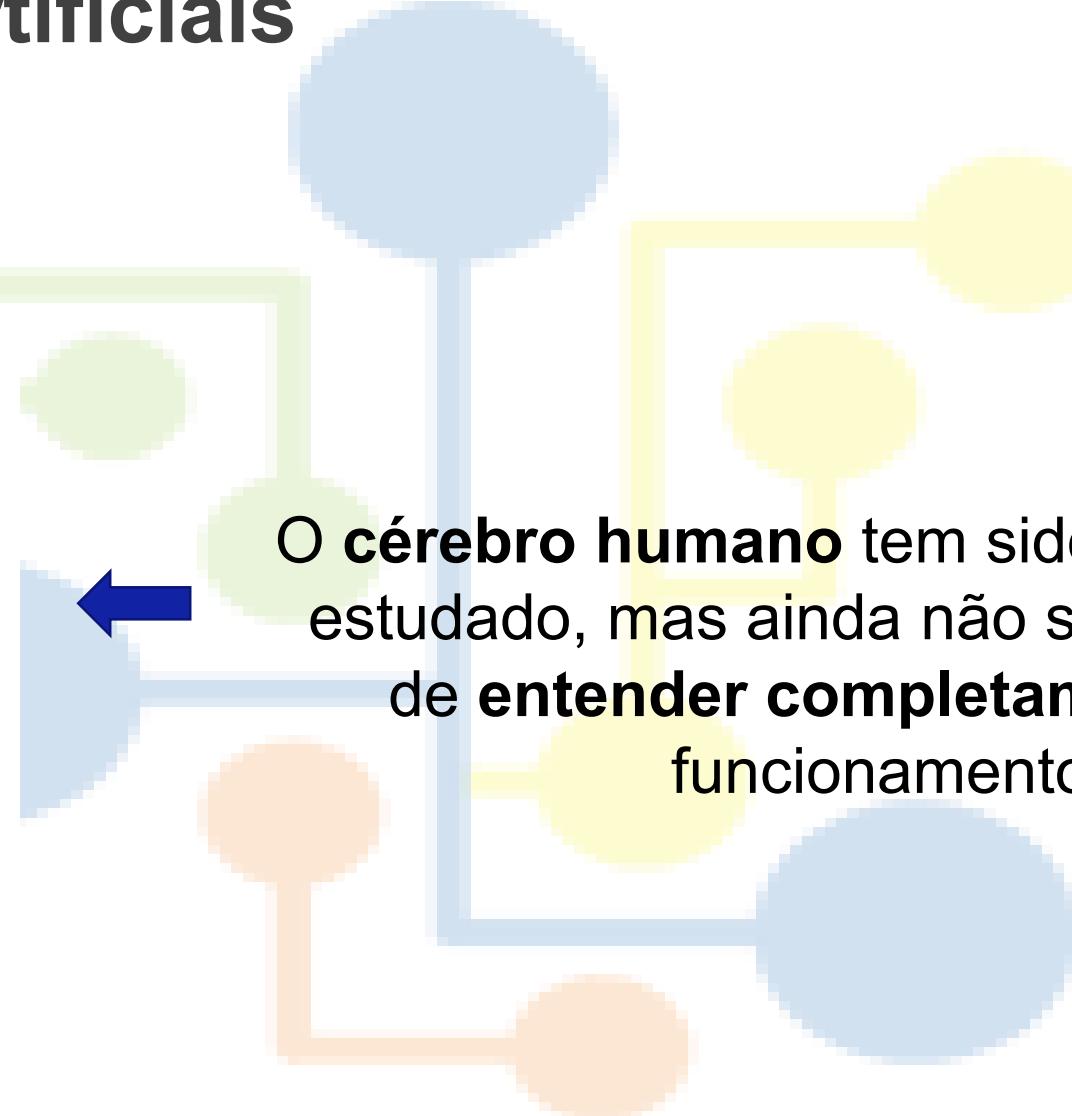
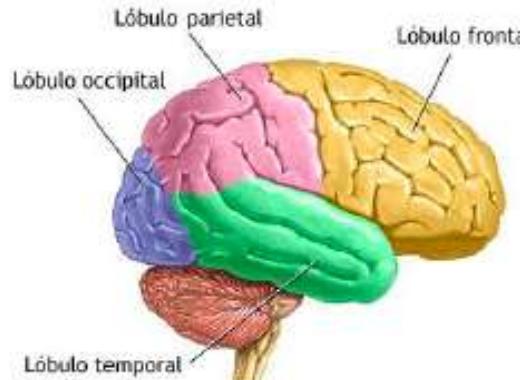


Redes Neurais Artificiais

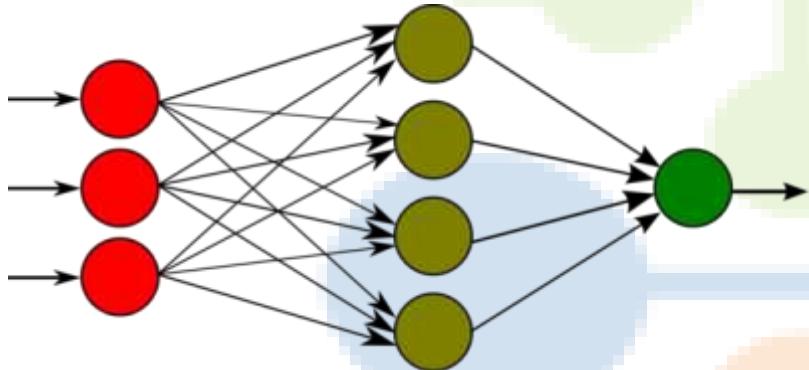


Este é o dispositivo mais incrível da história humana e que os cientistas estão tentando reproduzir em computadores!

Redes Neurais Artificiais



Redes Neurais Artificiais



Redes Neurais Artificiais podem ser consideradas um paradigma diferente de computação.

Redes Neurais Artificiais



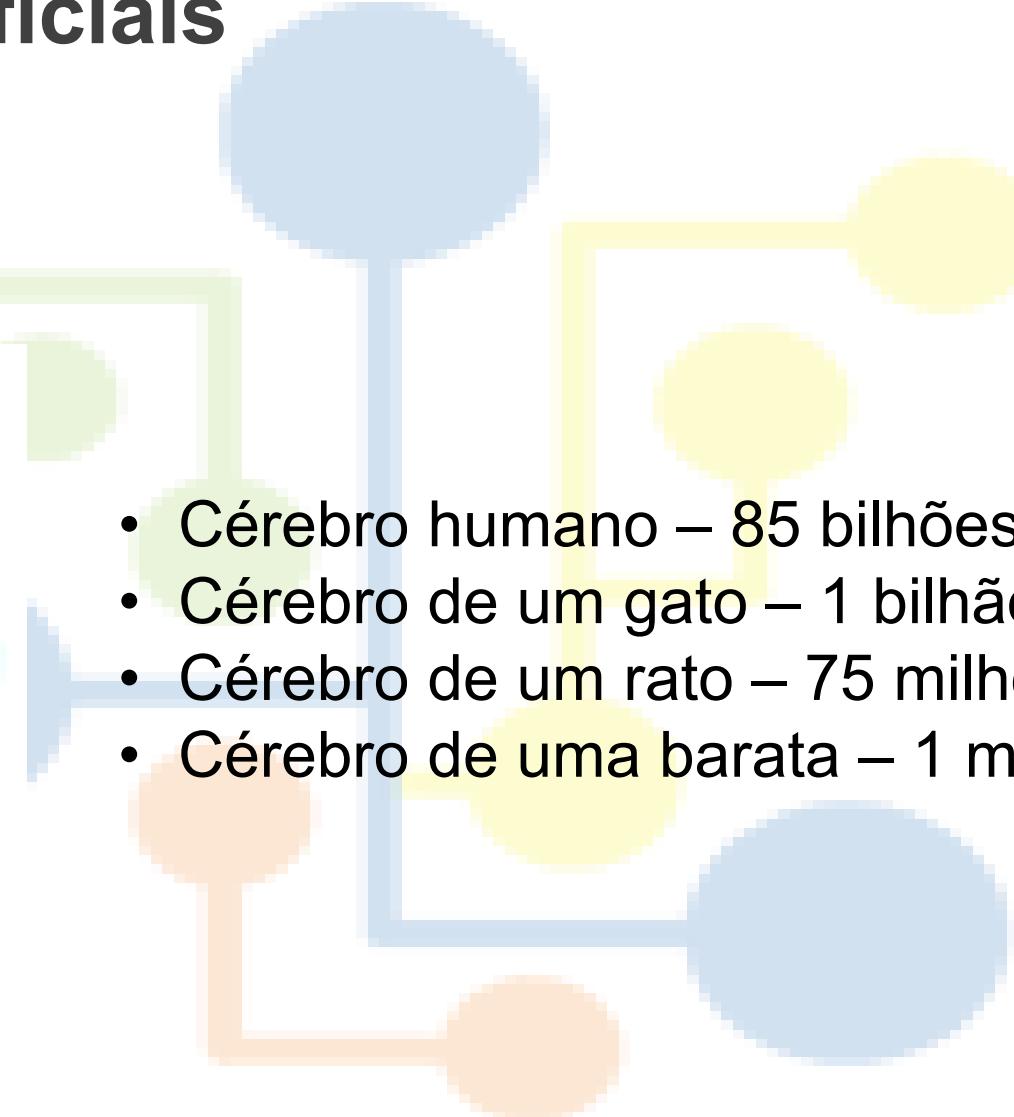
Redes Neurais Artificiais consistem em um modo de abordar a solução de problemas de **Inteligência Artificial**

Redes Neurais Artificiais



Assim como um cérebro usa uma rede de células interconectadas chamadas **neurônios** para criar um processador paralelo maciço, a rede neural usa uma **rede de neurônios artificiais** para resolver problemas de aprendizagem

Redes Neurais Artificiais



- Cérebro humano – 85 bilhões de neurônios
- Cérebro de um gato – 1 bilhão de neurônios
- Cérebro de um rato – 75 milhões de neurônios
- Cérebro de uma barata – 1 milhão de neurônios

Redes Neurais Artificiais



Agora fica mais claro porque a computação paralela em GPU's está acelerando o desenvolvimento de sistemas inteligentes, pois somos capazes de processar cada vez mais dados em redes neurais artificiais com cada vez mais neurônios

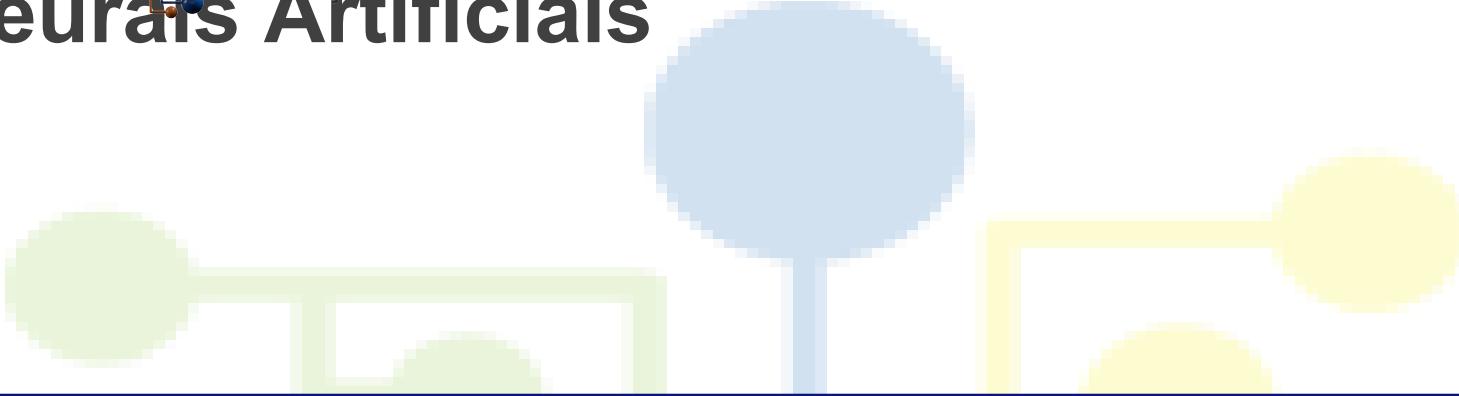
Redes Neurais Artificiais

Programas de reconhecimento de voz e escrita

Automação de dispositivos inteligentes

Modelos sofisticados de padrões climáticos

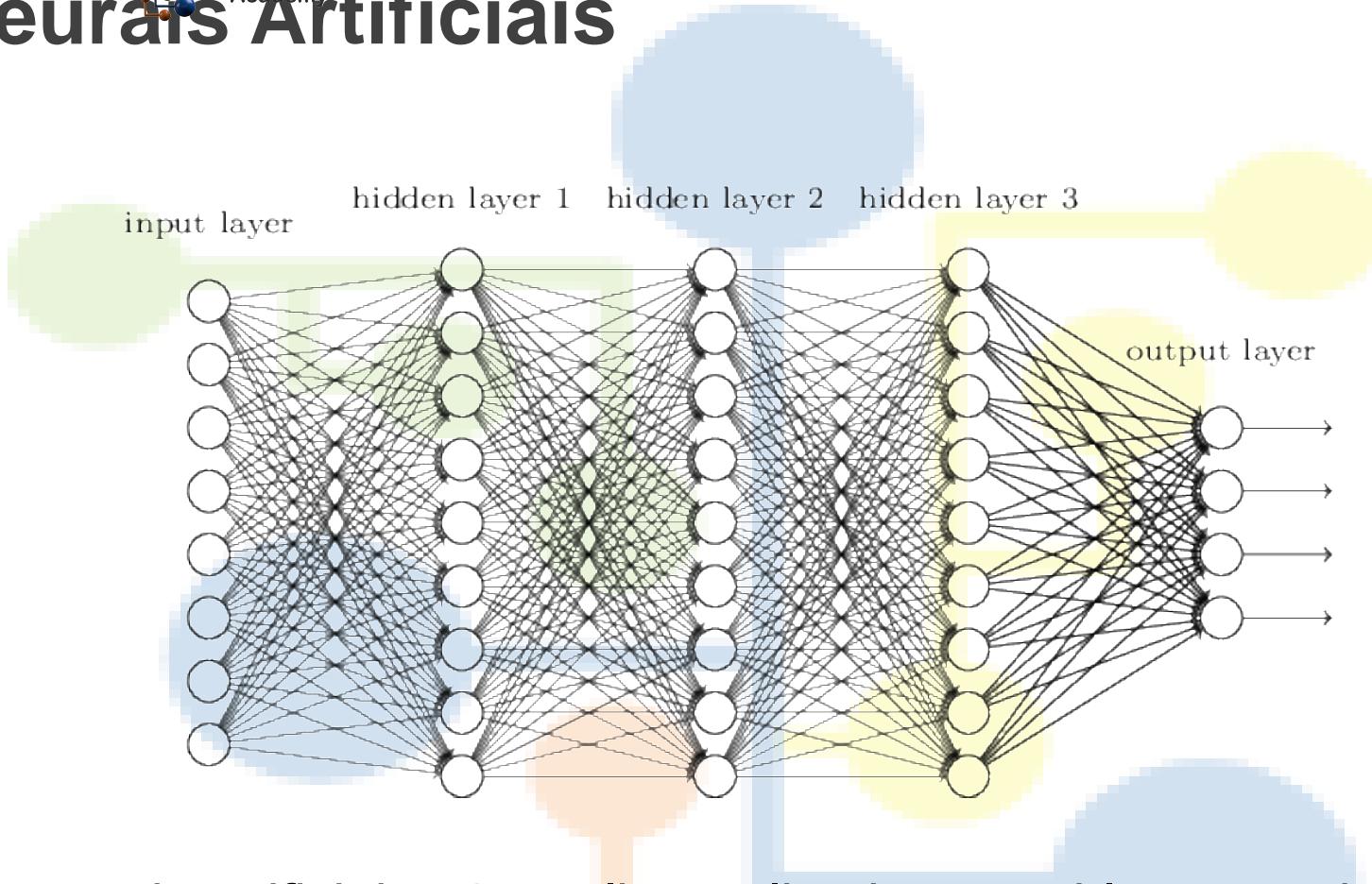
Redes Neurais Artificiais



As Redes Neurais Artificiais são modelos versáteis que podem ser aplicadas a quase todas as tarefas de aprendizagem: classificação, previsão numérica e mesmo reconhecimento não supervisionado de padrões



Redes Neurais Artificiais

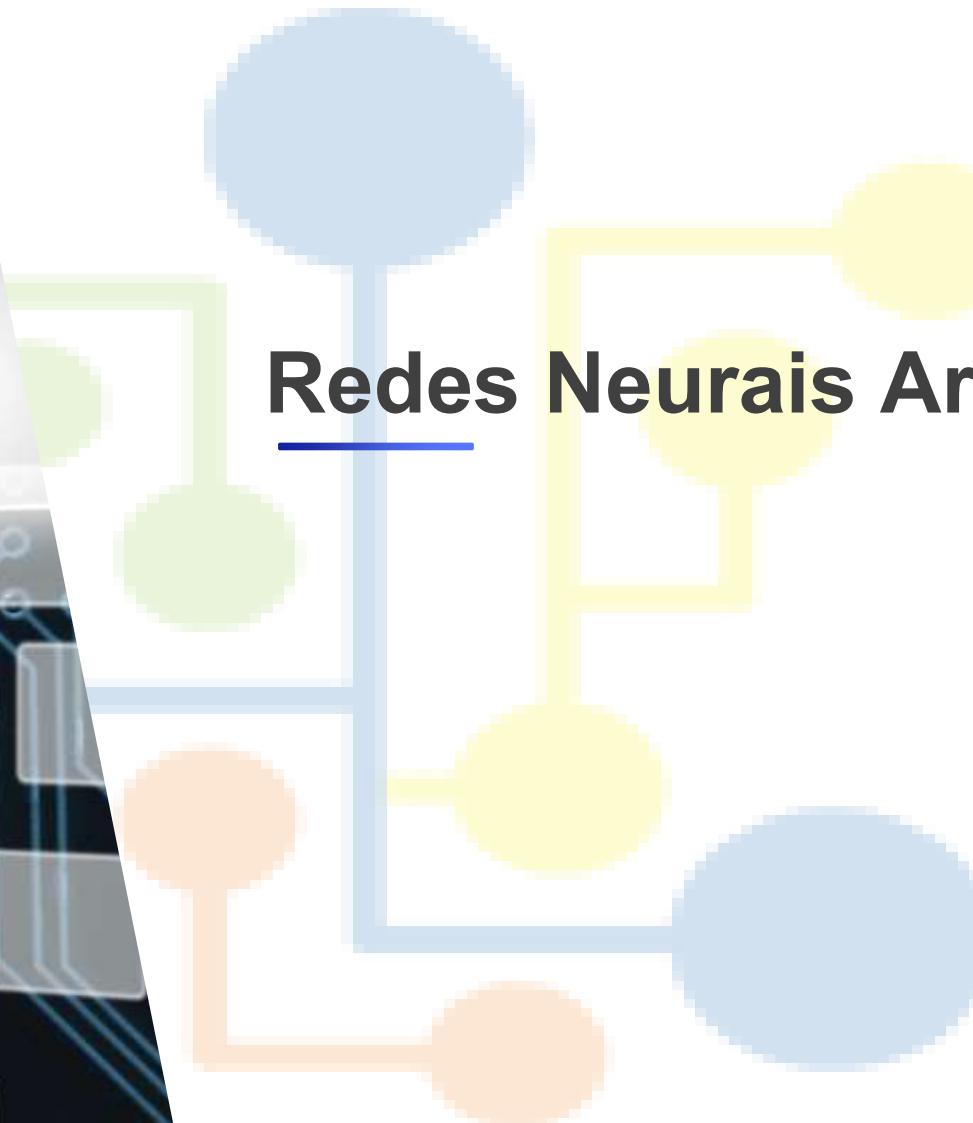


As redes neurais artificiais são melhor aplicadas a problemas onde os dados de entrada e os dados de saída são bem definidos ou, pelo menos, bastante simples, mas o processo que relaciona a entrada com a saída é extremamente complexo



Data Science Academy raphaelbsfontenelle@gmail.com 615c1fdde32fc361b30c9ec2

Redes Neurais Artificiais



Data Science Academy



Redes Neurais Artificiais

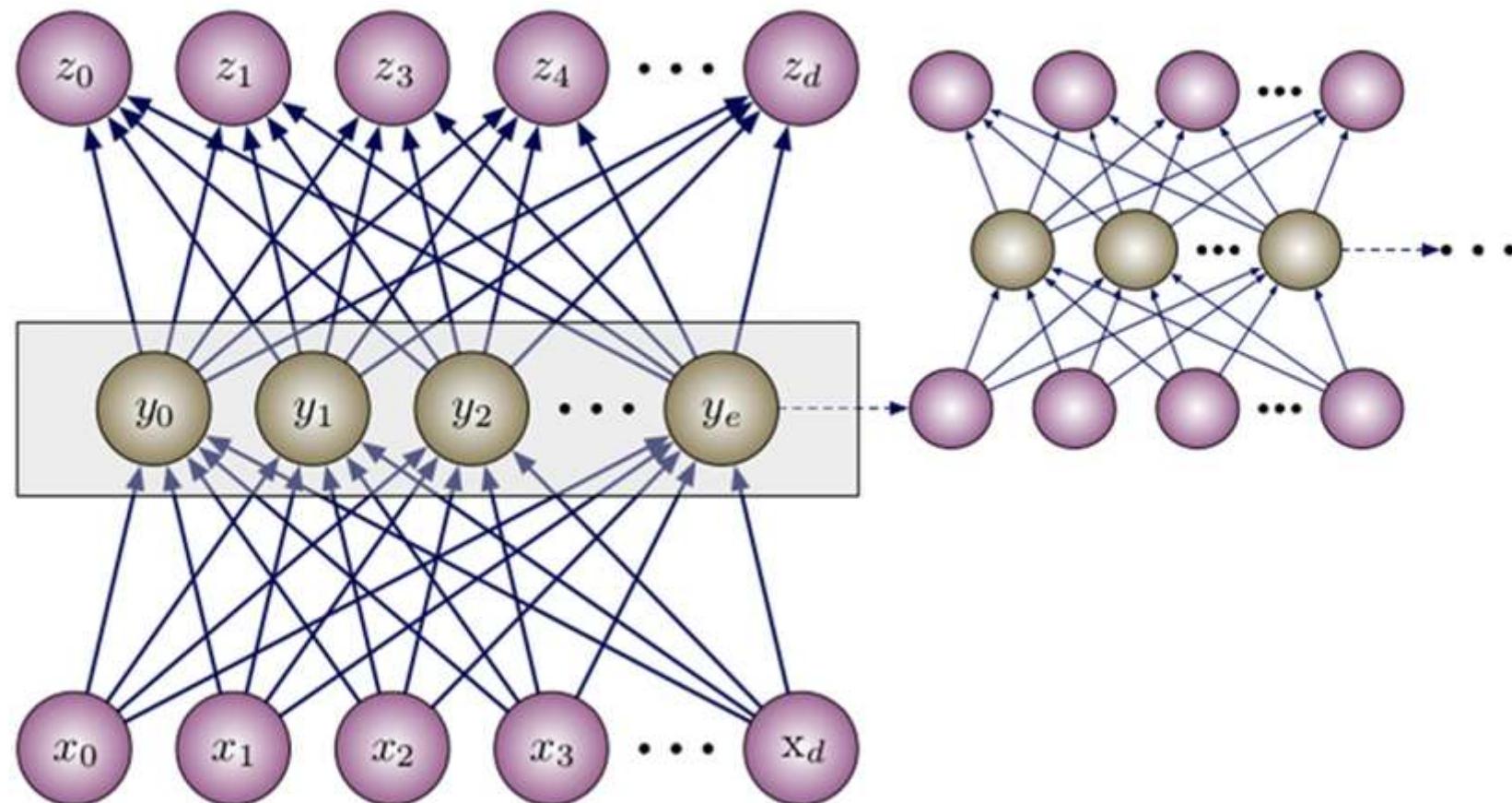


Warren McCulloch

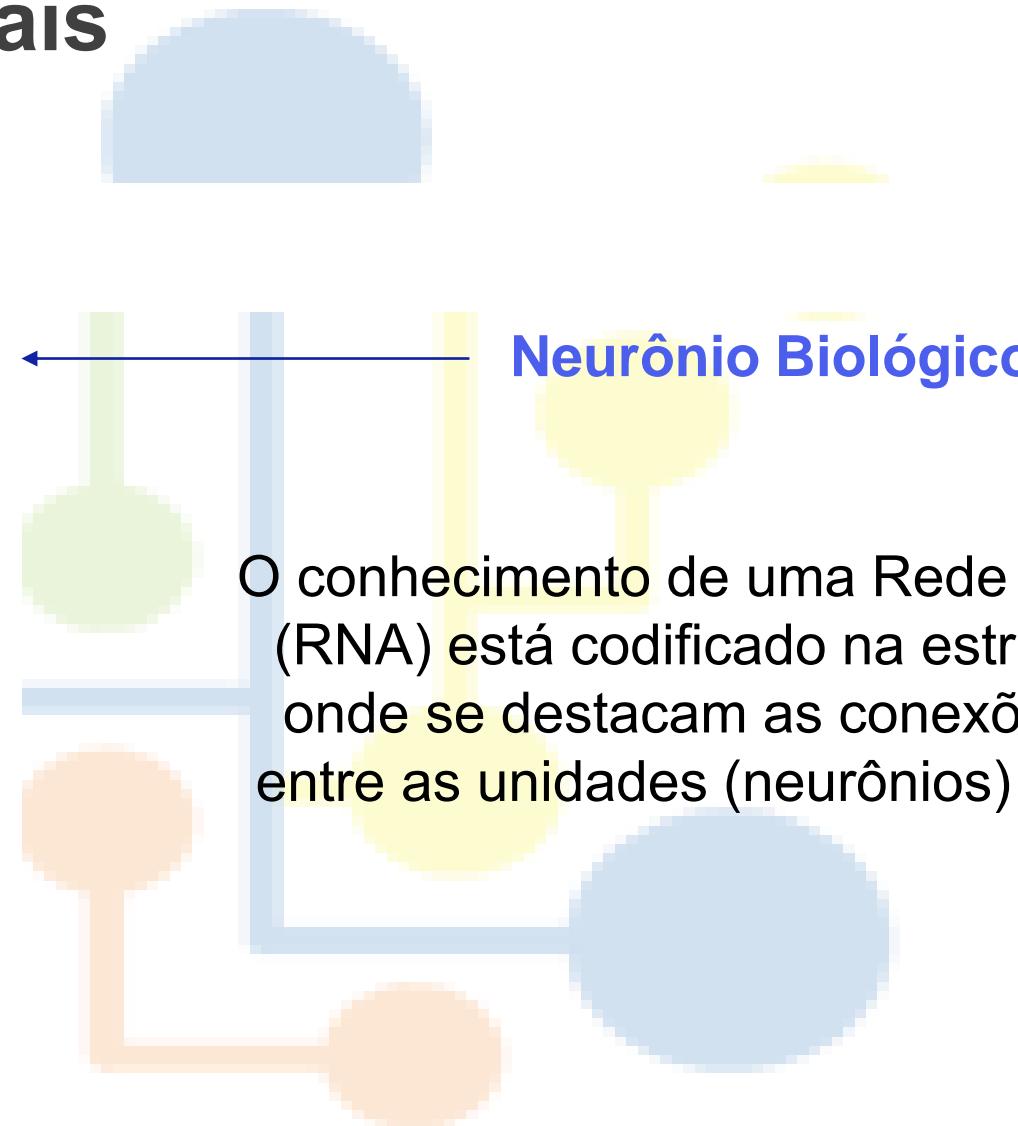
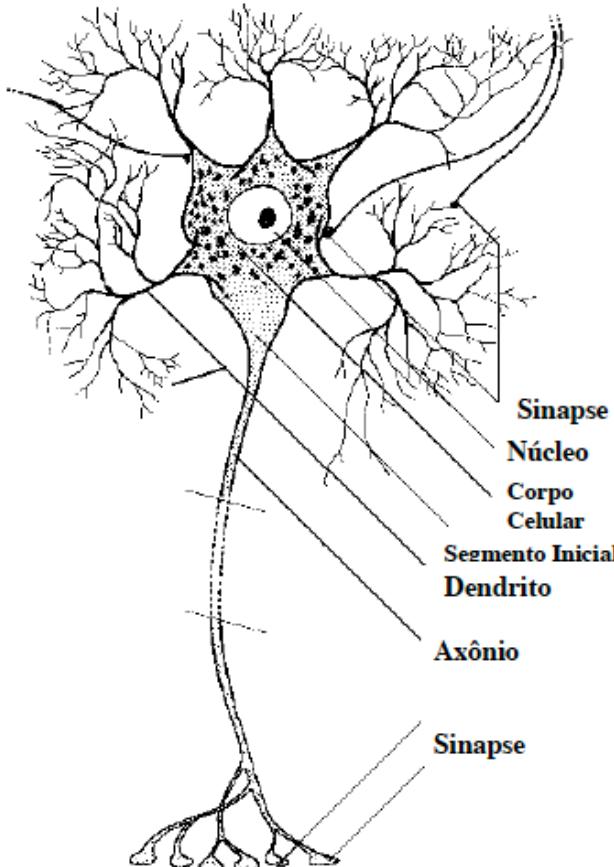


Walter Pitts

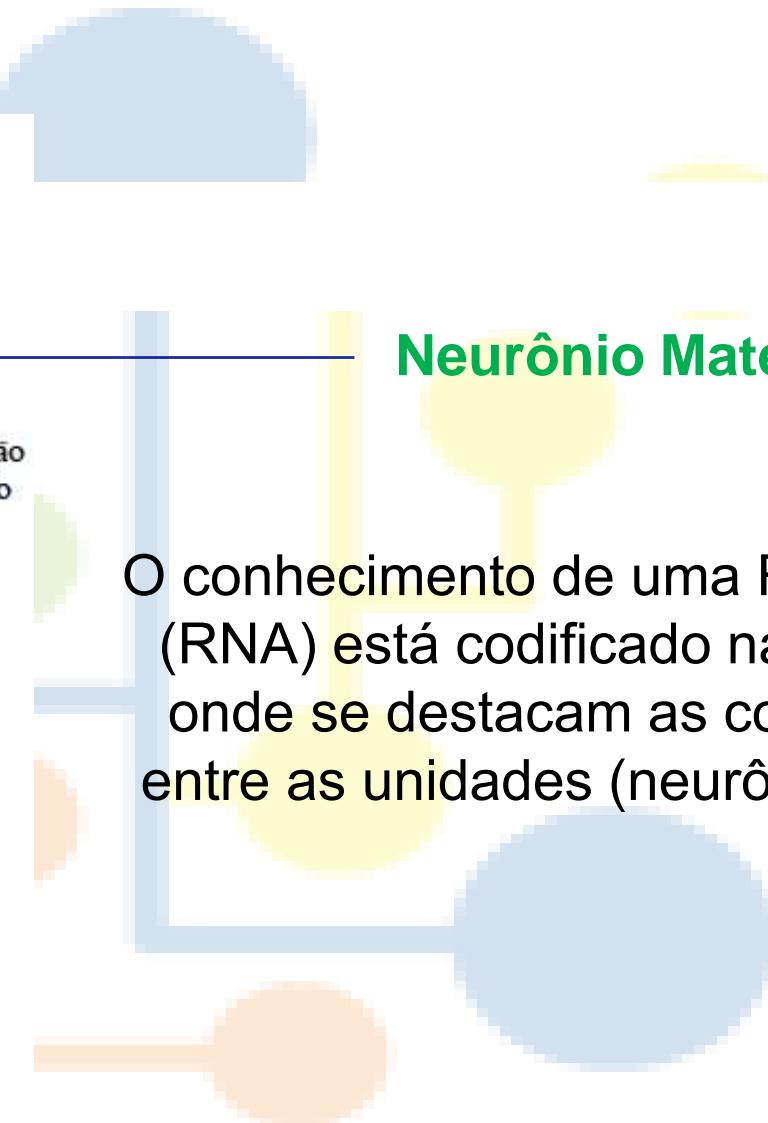
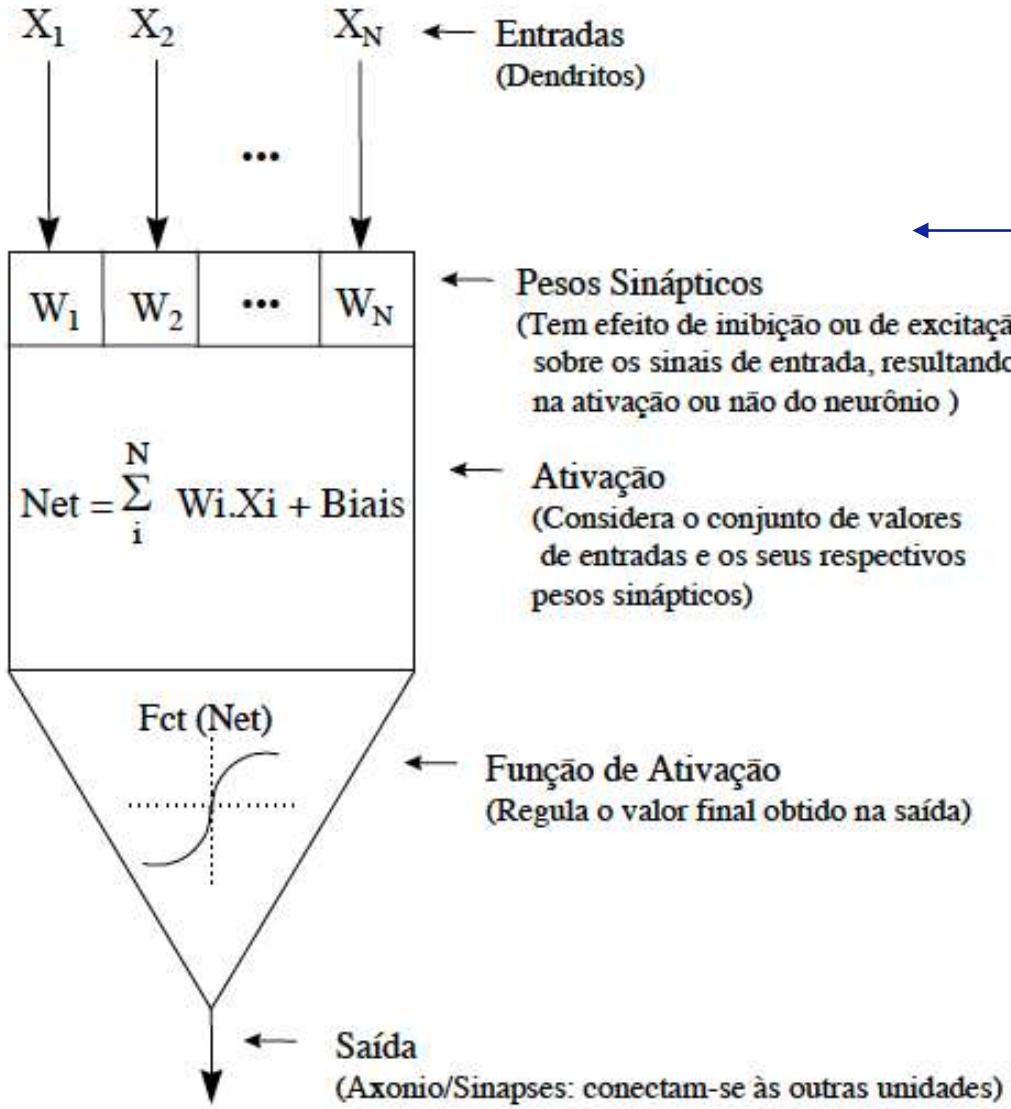
Redes Neurais Artificiais



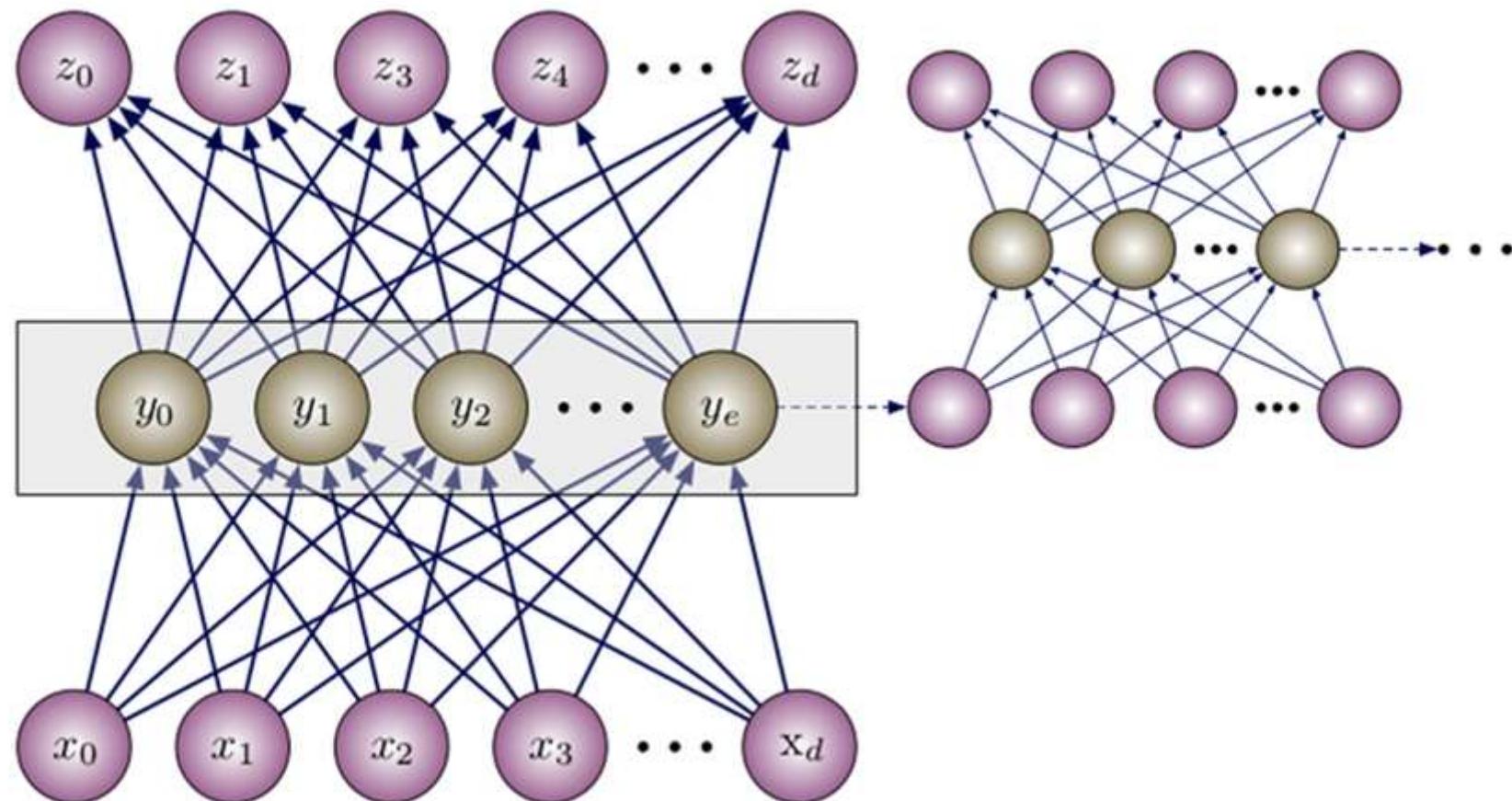
Redes Neurais Artificiais



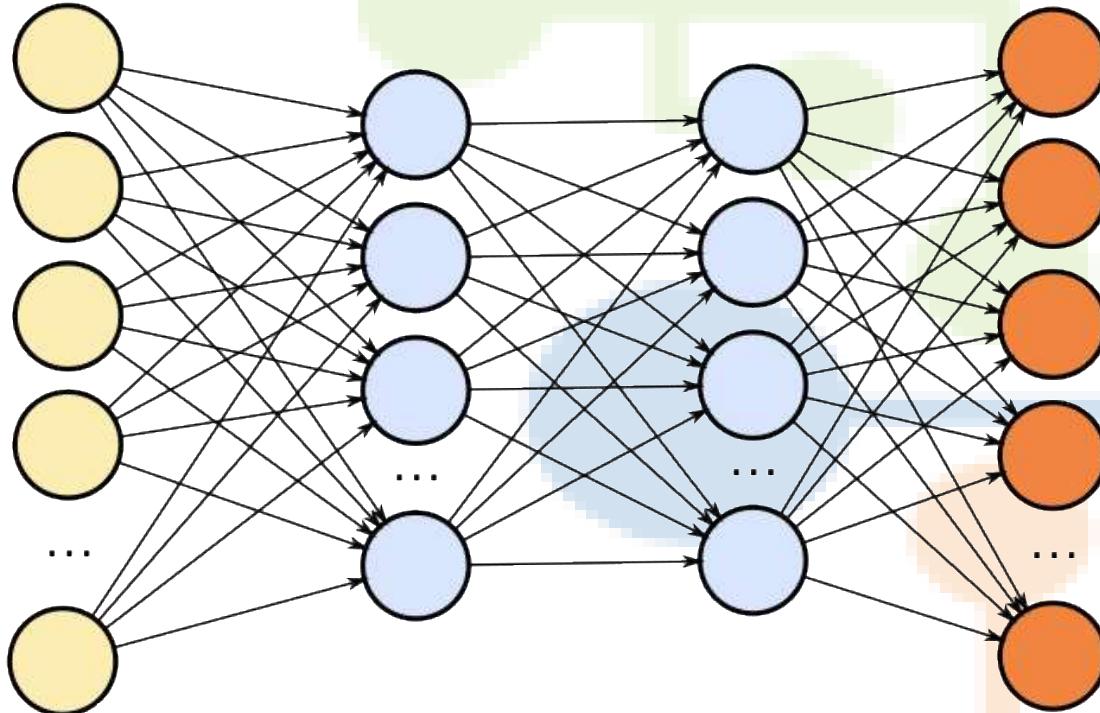
Redes Neurais Artificiais



Redes Neurais Artificiais

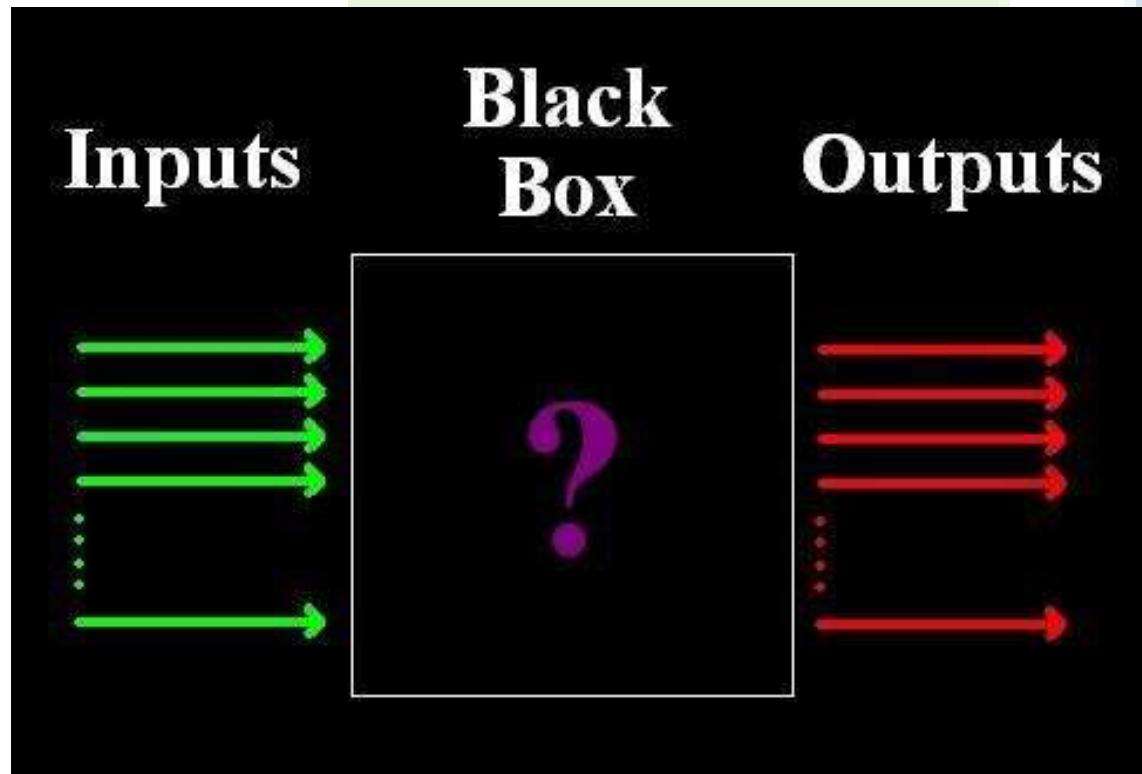


Redes Neurais Artificiais



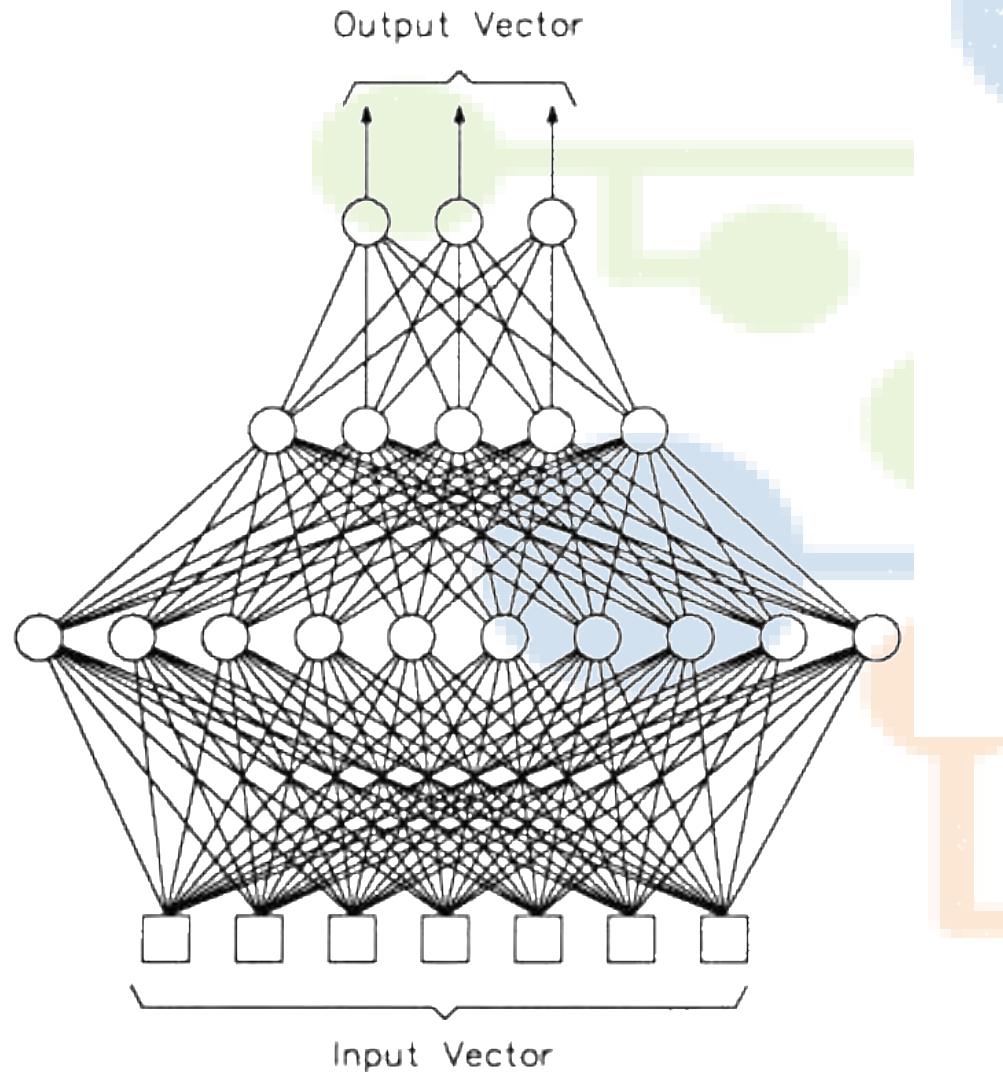
Um dos benefícios das redes diz respeito ao tratamento de um problema clássico da Inteligência Artificial, que é a representação de um universo não-estacionário (onde as estatísticas mudam com o tempo)

Redes Neurais Artificiais



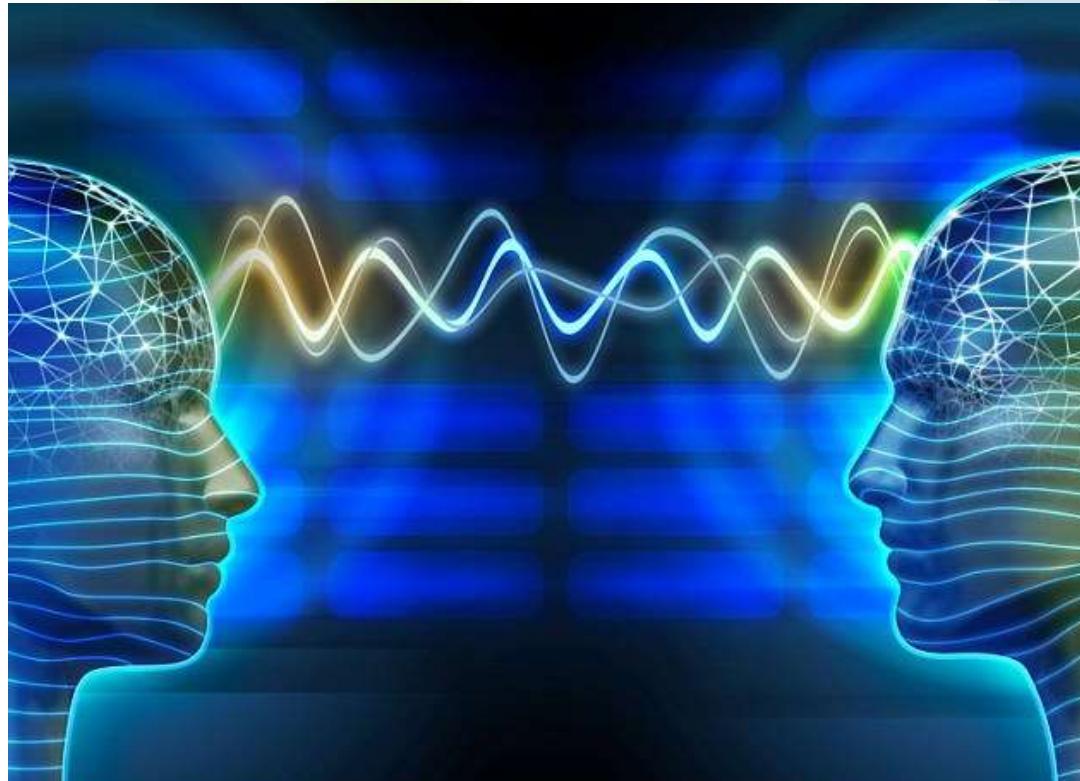
Uma desvantagem das redes neurais é o fato delas, normalmente, serem uma "caixa preta"

Redes Neurais Artificiais



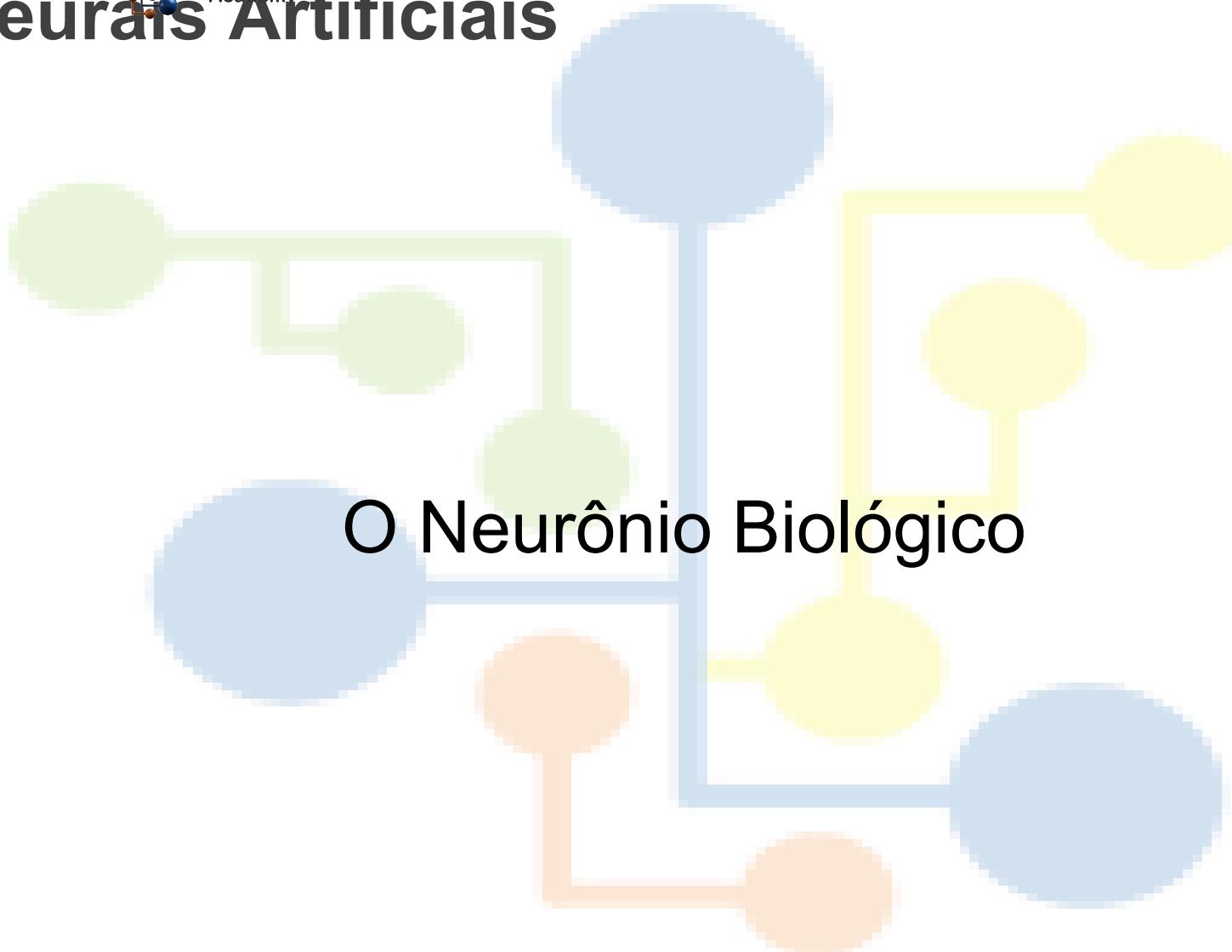
A solução de problemas através das RNAs é bastante atrativa, pois o paralelismo constitui-se na característica principal das RNAs, onde esta cria a possibilidade de um desempenho superior em relação a solução de problemas baseados nos modelos convencionais.

Redes Neurais Artificiais



A generalização está associada à capacidade da rede em aprender através de um conjunto reduzido de exemplos, e posteriormente, dar respostas coerentes a dados não apresentados a rede.

Redes Neurais Artificiais



Redes Neurais Artificiais

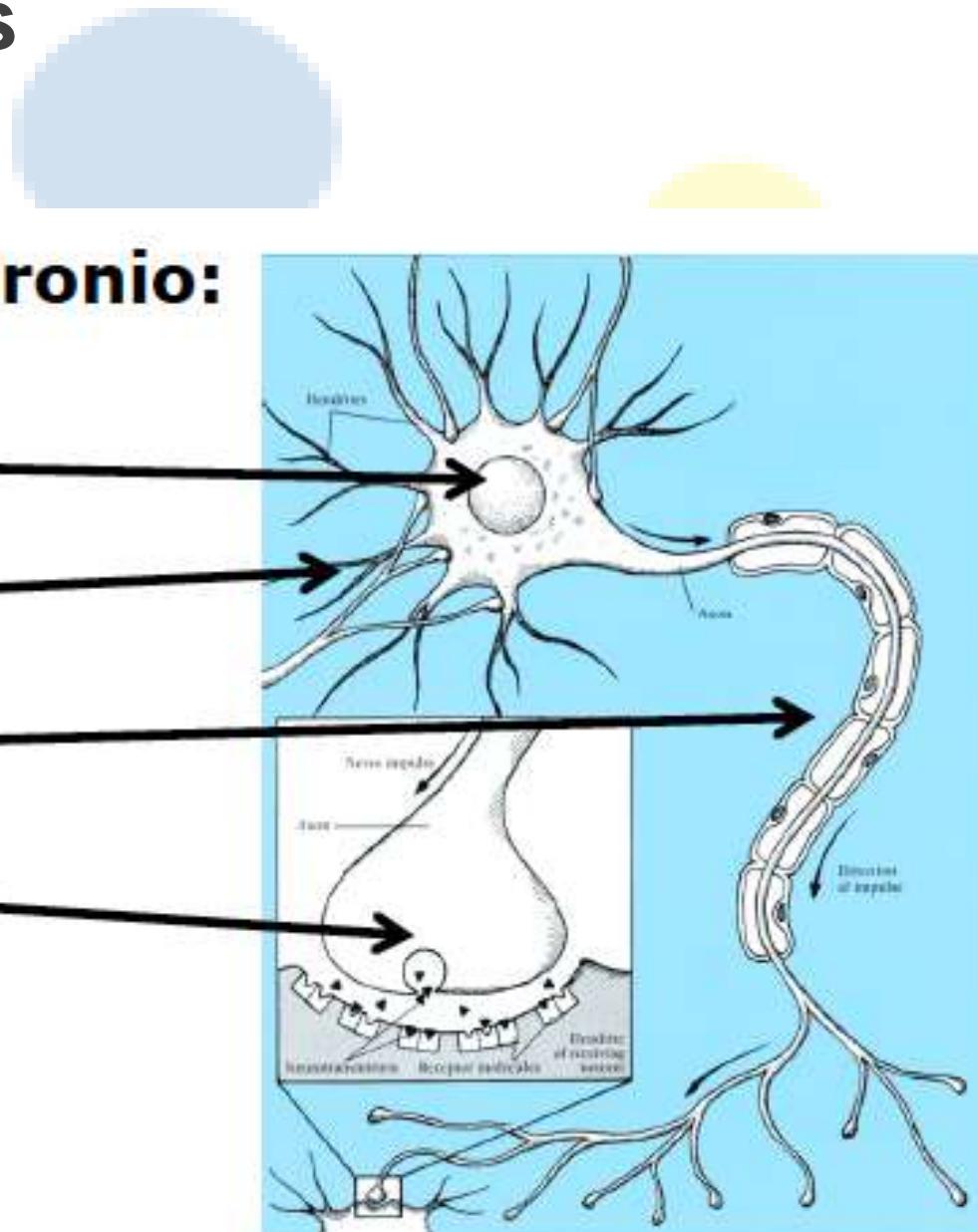
Estrutura de um Neurônio:

Corpo celular

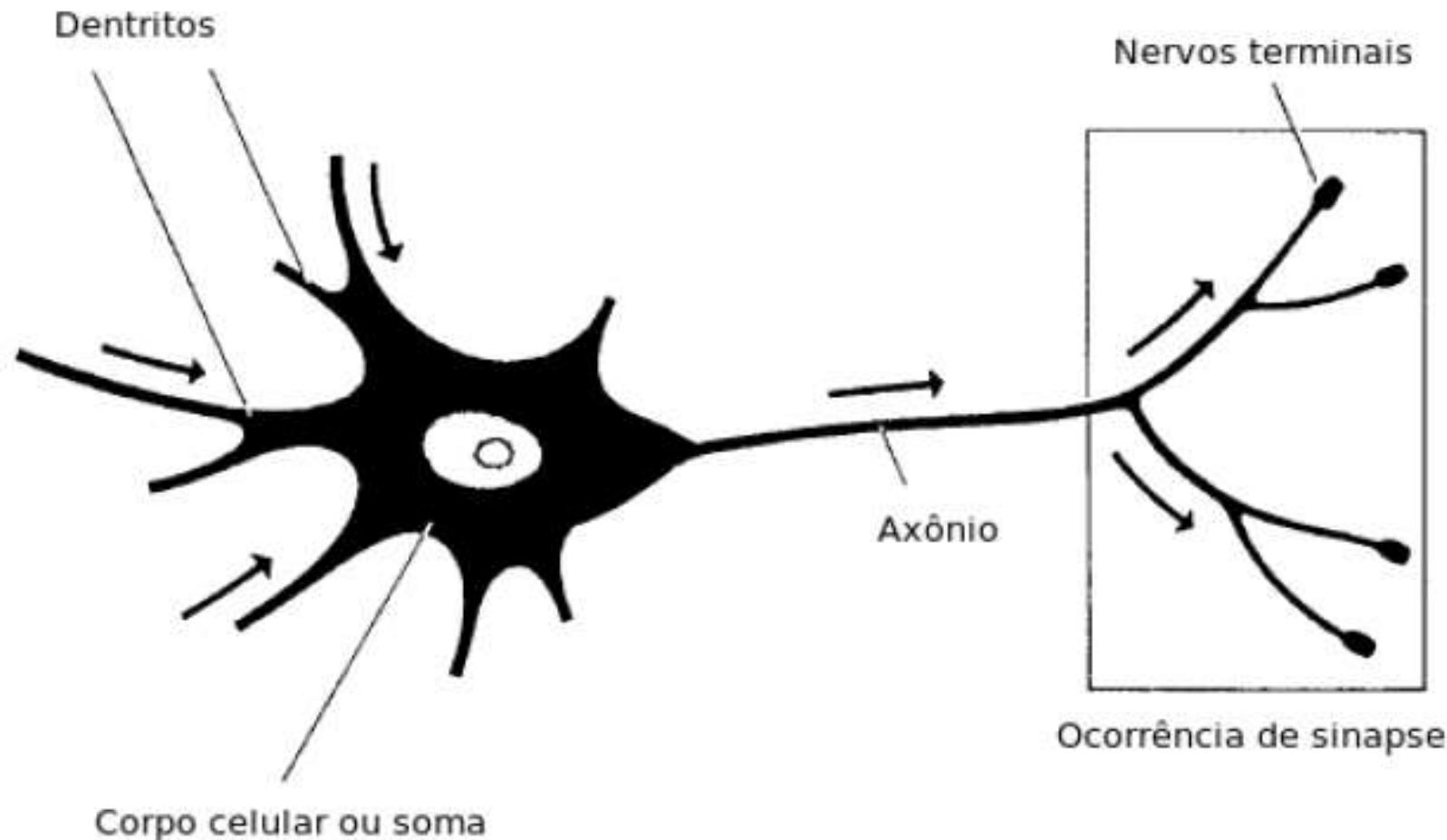
Dendritos

Axônio

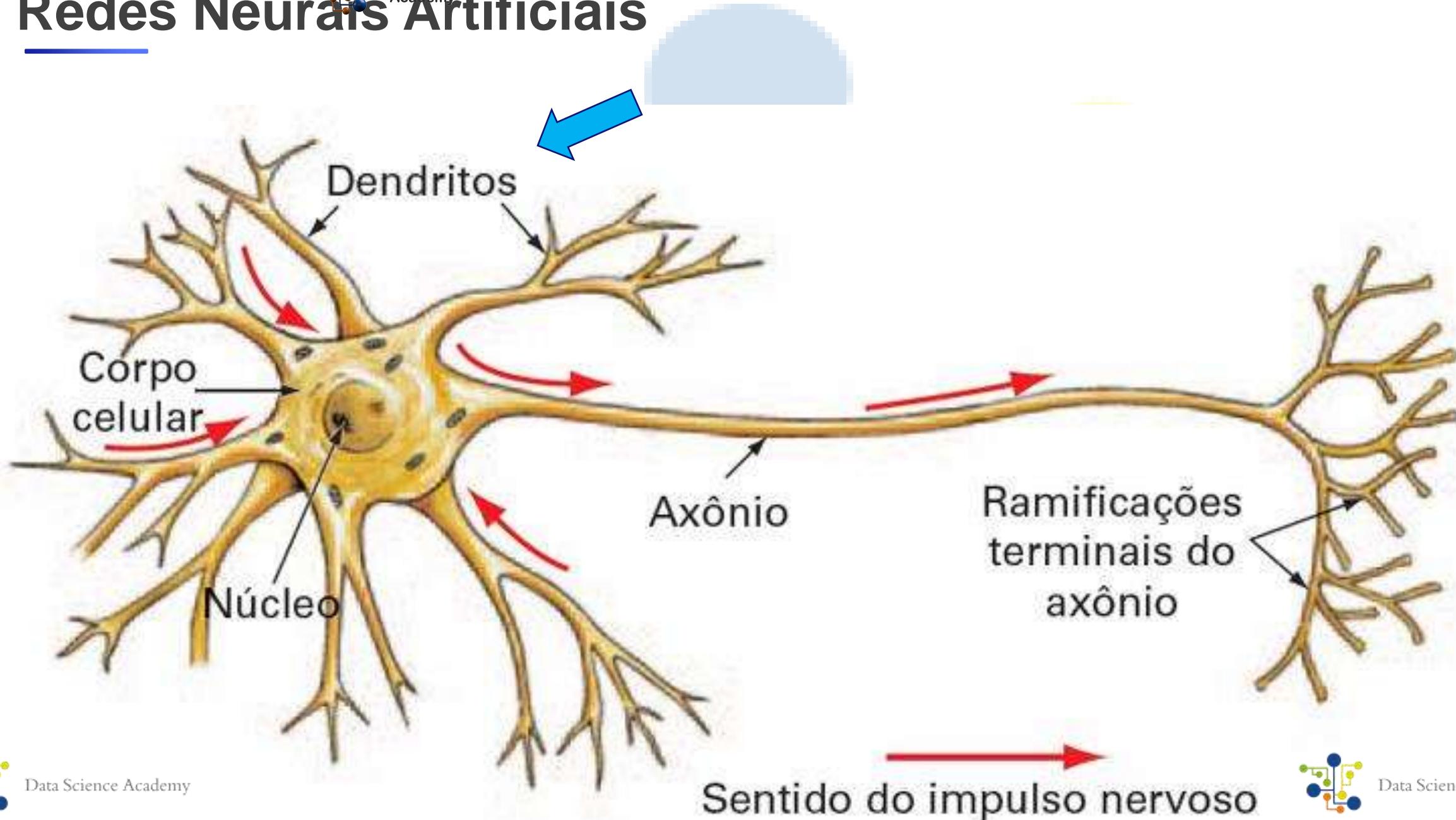
Terminais sinápticos



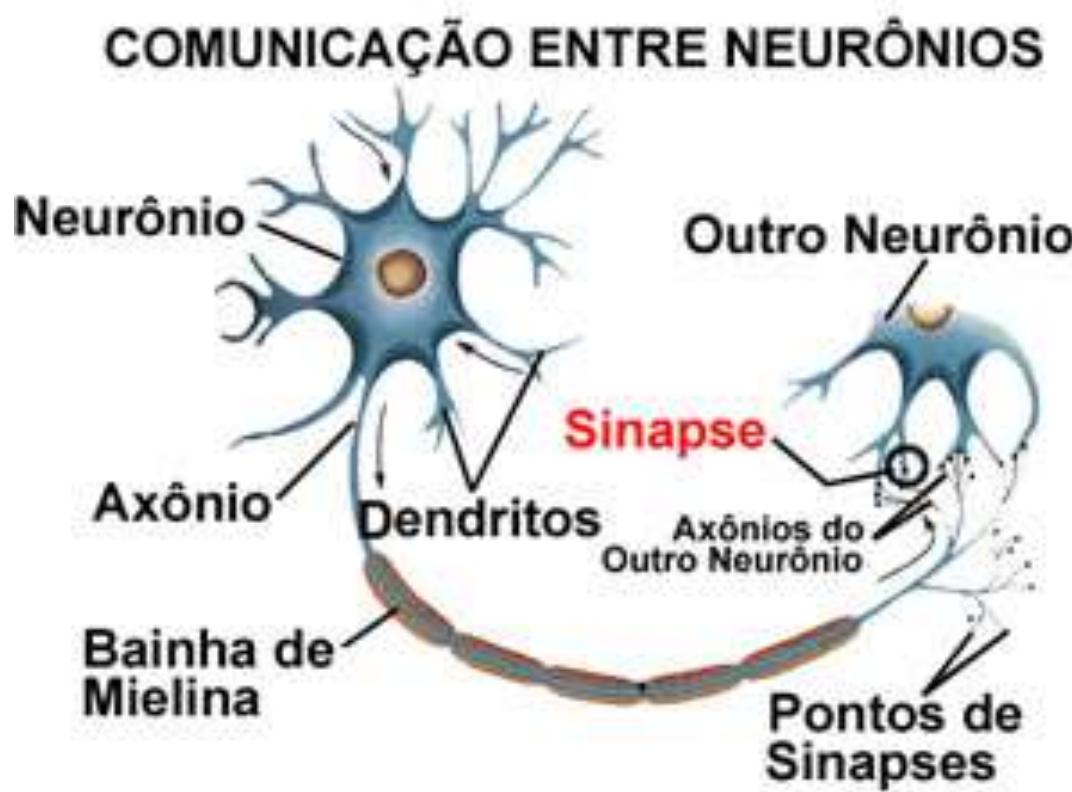
Redes Neurais Artificiais



Redes Neurais Artificiais

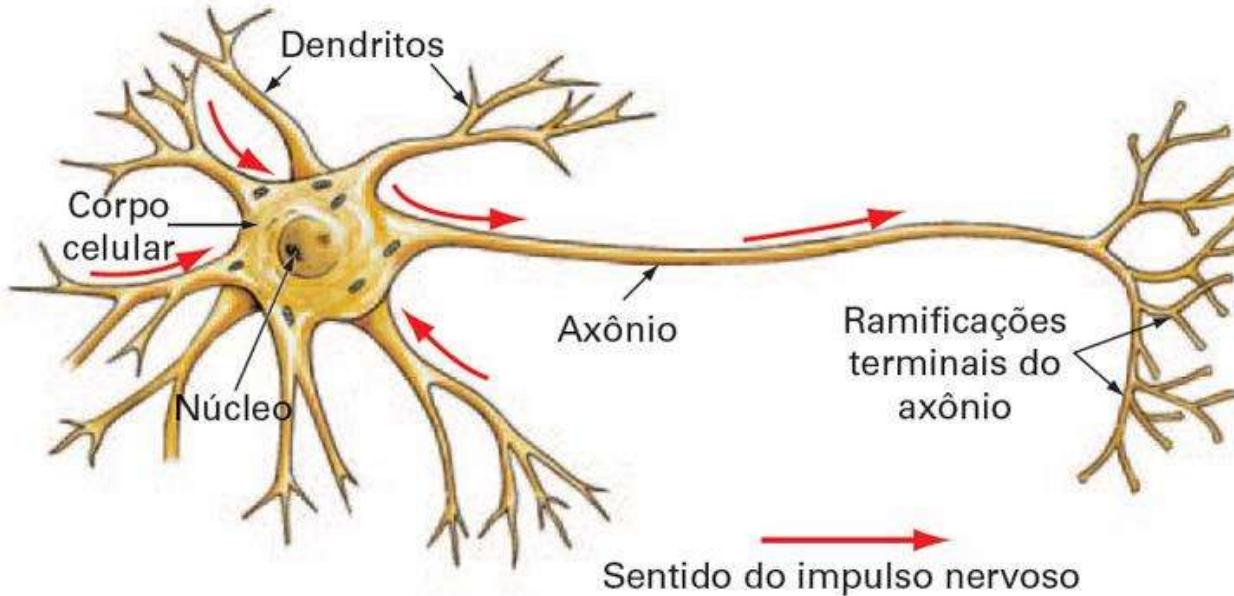


Redes Neurais Artificiais



O ponto de contato entre a terminação axônica de um neurônio e o dentrito de outro é chamado **sinapse**

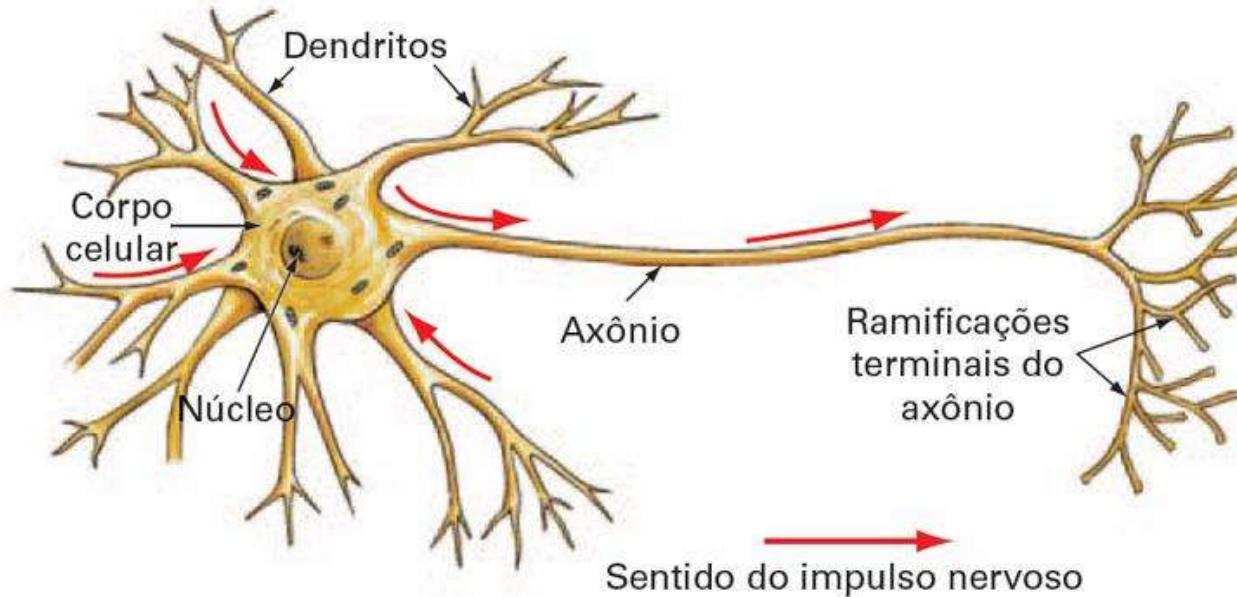
Redes Neurais Artificiais



Se esses sinais forem superiores a aproximadamente 50mV (limiar do disparo), seguem pelo axônio. Caso contrário, são bloqueados e não preosseguem (são considerados irrelevantes).

Se o sinal for superior a certo limite (***threshold***), vai em frente; caso contrário é bloqueado e não segue.

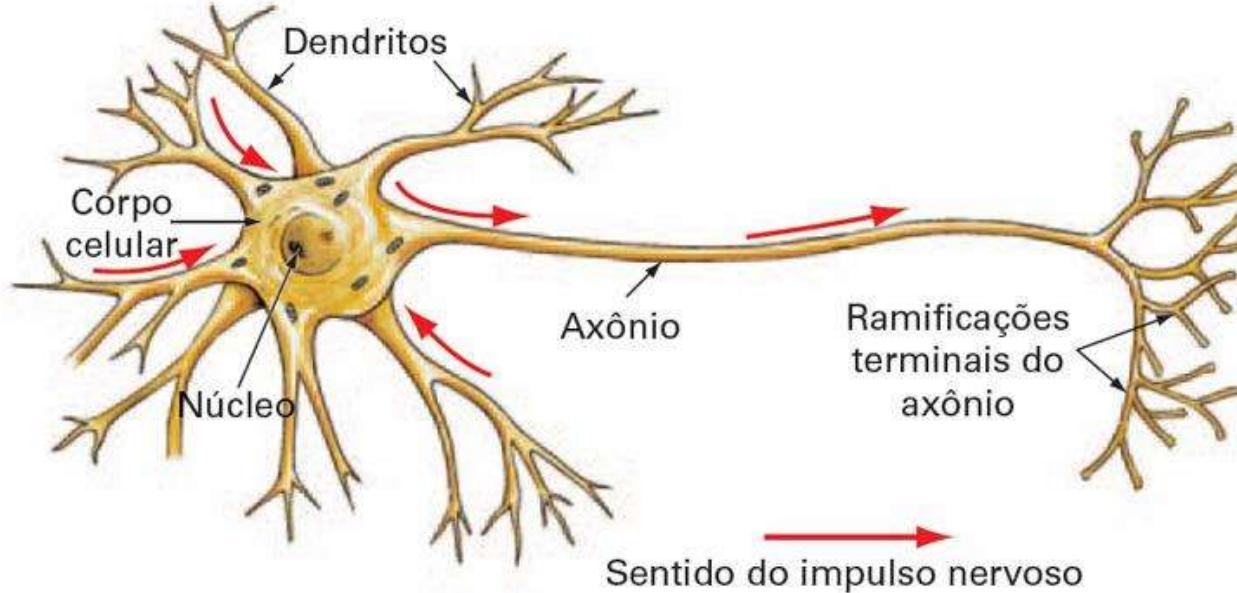
Redes Neurais Artificiais



Um neurônio recebe sinais através de inúmeros dendritos, os quais são **ponderados** e enviados para o axônio, podendo ou não seguir adiante (threshold)

Cada condutor, está associado um **peso** pelo qual o sinal é multiplicado.
A memória são os pesos.

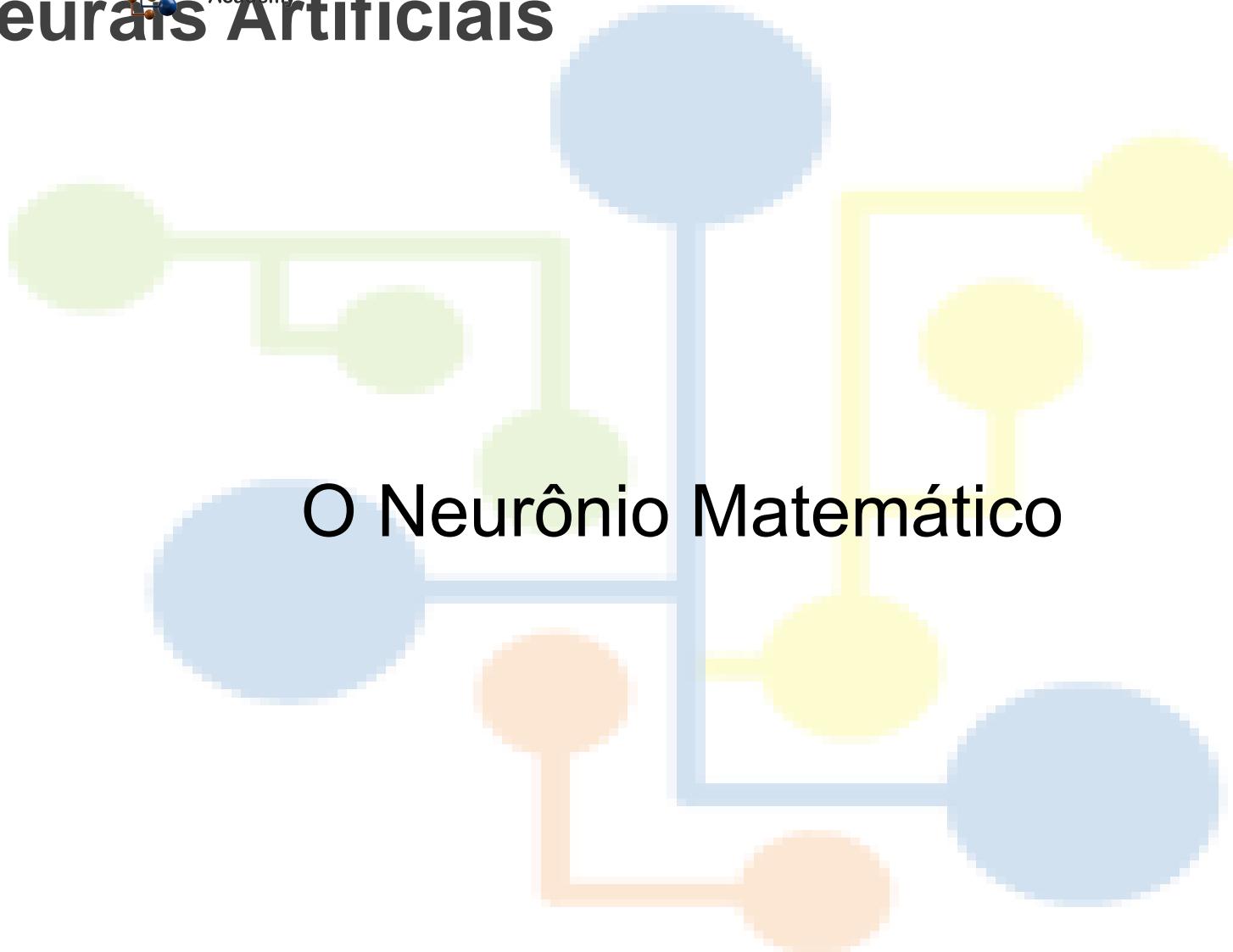
Redes Neurais Artificiais



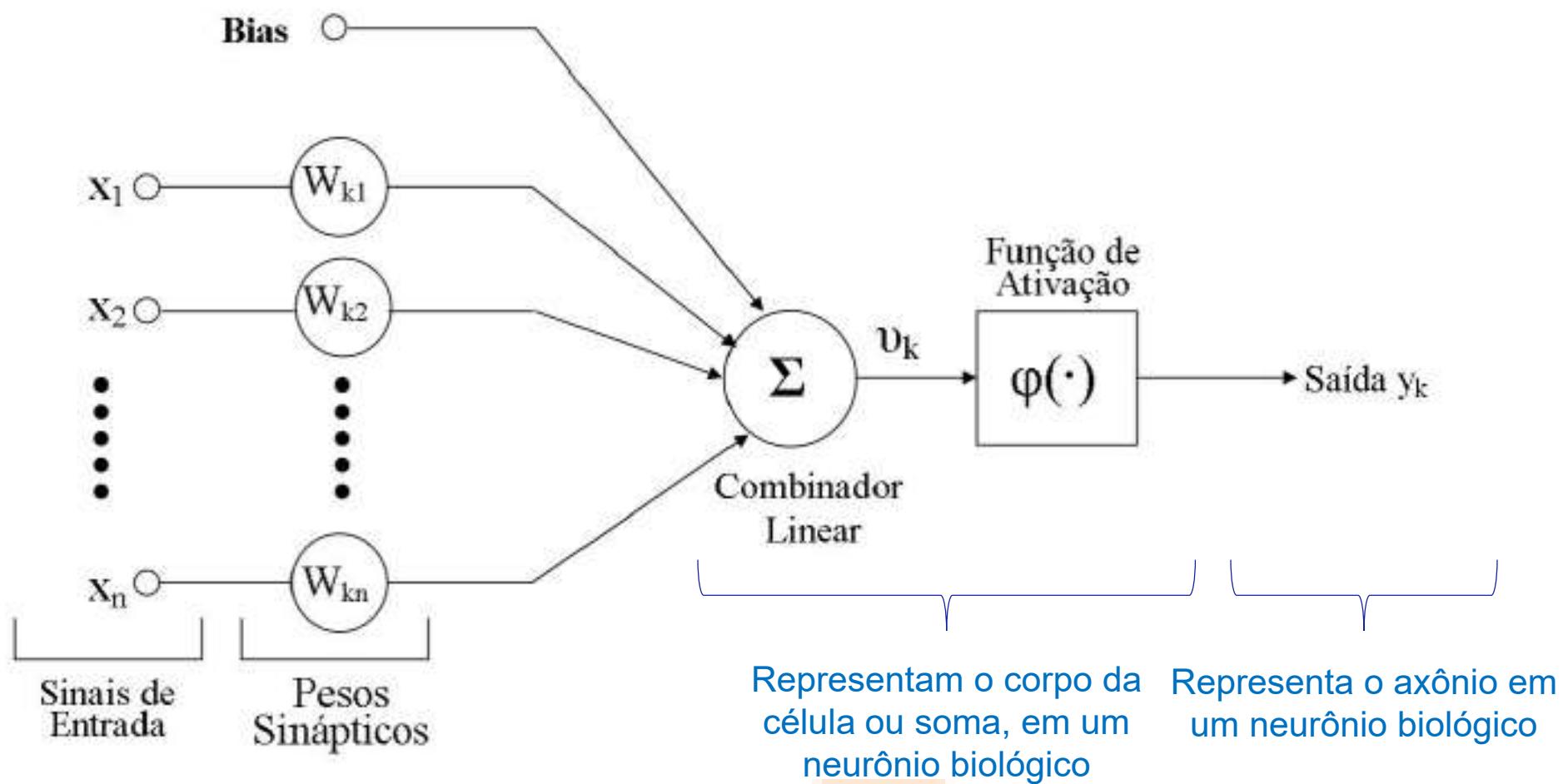
Cada região do cérebro possui uma arquitetura de rede diferente: varia o número de neurônios, de sinapses por neurônio, valor dos thresholds e dos pesos, etc...

Os valores dos pesos são estabelecidos por meio de treinamento recebido pelo cérebro durante a vida útil.
É a memorização.

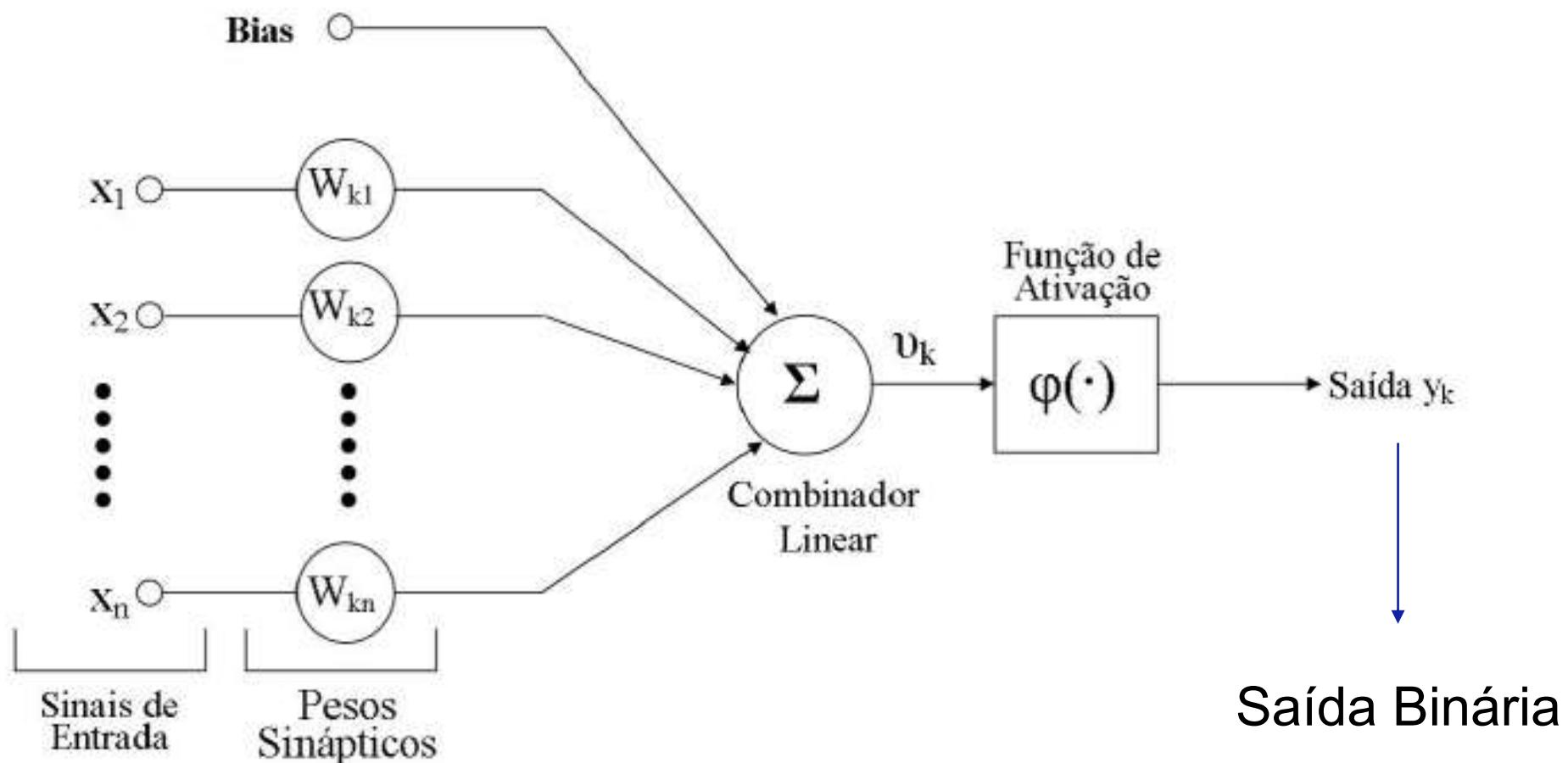
Redes Neurais Artificiais



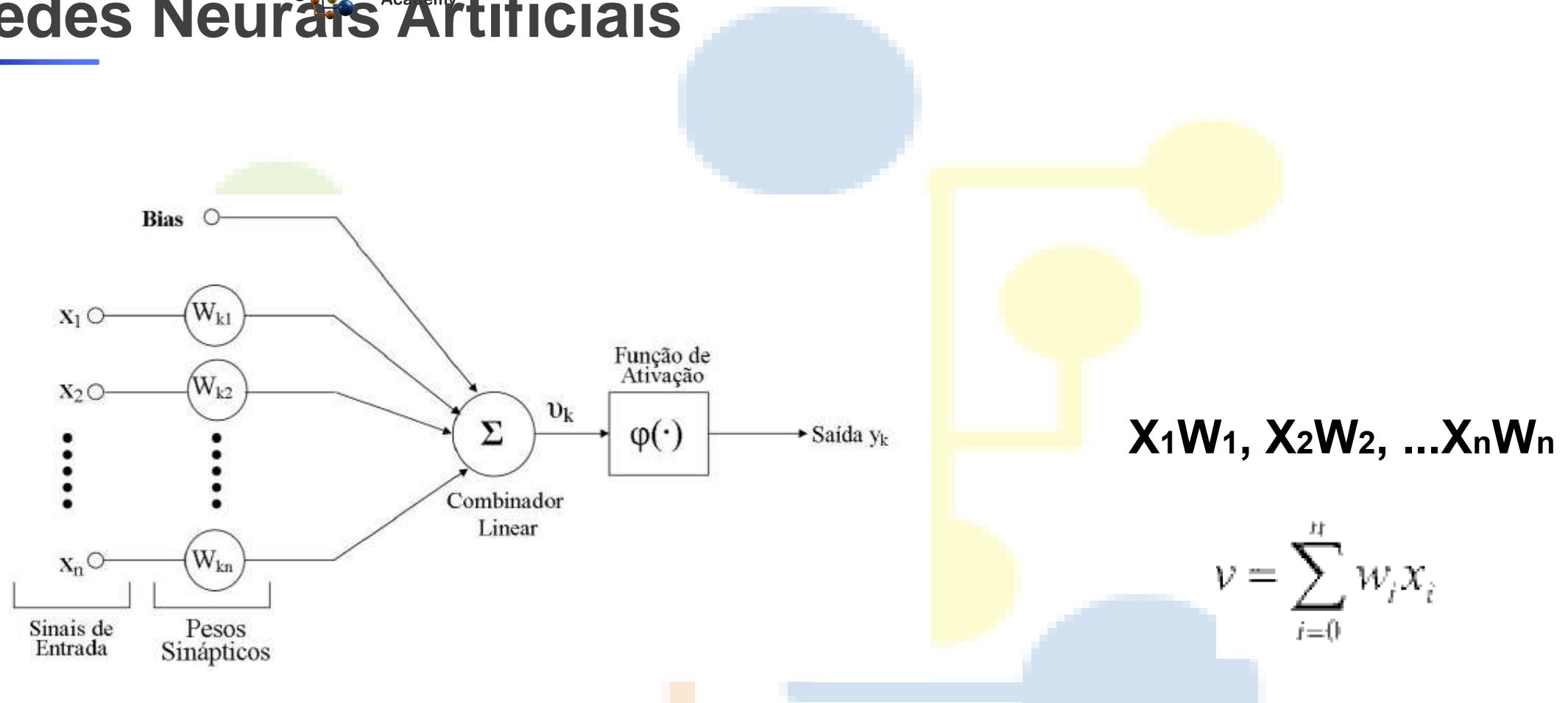
Redes Neurais Artificiais



Redes Neurais Artificiais



Redes Neurais Artificiais



Um neurônio dispara quando a soma dos impulsos que ele recebe ultrapassa o seu limiar de excitação chamado de **threshold**

Redes Neurais Artificiais



Note que este modelo matemático simplificado de um neurônio é estático, ou seja, não considera a dinâmica do neurônio natural. No neurônio biológico, os sinais são enviados em pulsos e alguns componentes dos neurônios biológicos, a exemplo do axônio, funcionam como filtros de frequência.



Redes Neurais Artificiais

$$u_k = \sum_{j=1}^m w_{kj} \cdot x_j$$

Fórmula do Neurônio Artificial

$$y_k = \varphi(u_k)$$

Fórmula da Função de Ativação

$$\varphi(u) = \begin{cases} 1 & , \text{ se } u \geq 0 \\ 0 & , \text{ se } u < 0 \end{cases}$$

Redes Neurais Artificiais

Dentre as funções de ativação utilizadas, podemos destacar:

$$\varphi(v) = \frac{1}{1 + e^{-v}}$$

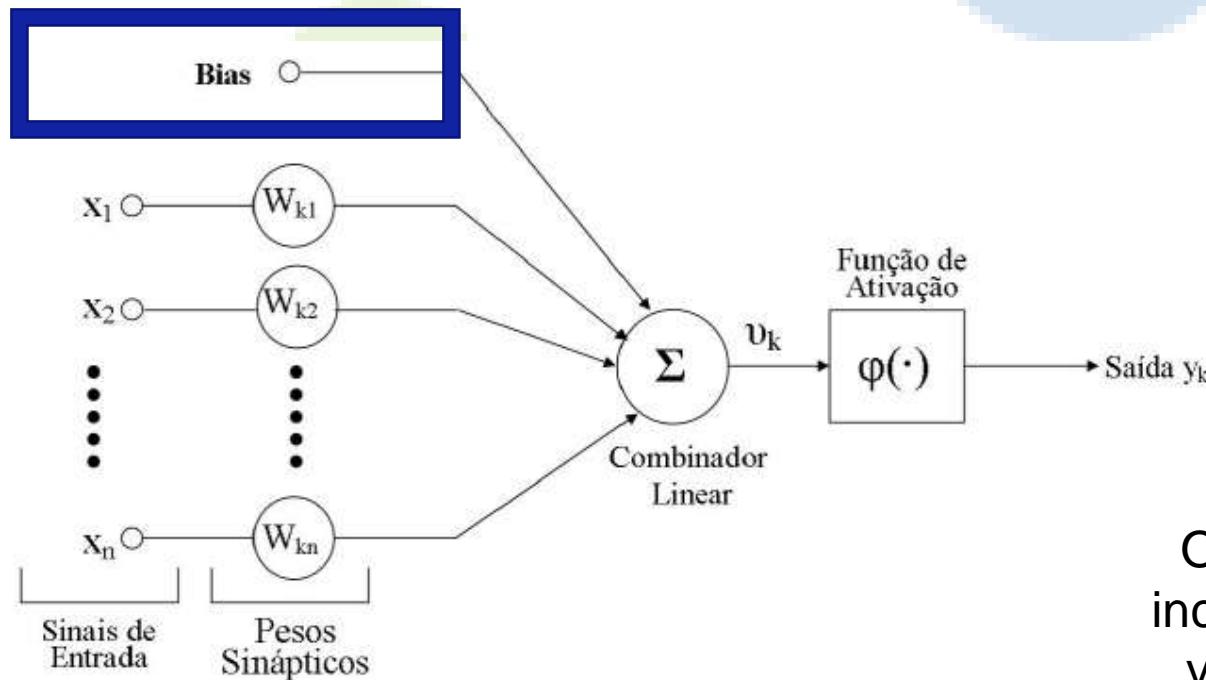
Função Sigmóide

Redes Neurais Artificiais

Dentre as funções de ativação utilizadas, podemos destacar:

	Propagation	Back-propagation
Sigmoid	$y_s = \frac{1}{1+e^{-x_s}}$	$\left[\frac{\partial E}{\partial x} \right]_s = \left[\frac{\partial E}{\partial y} \right]_s \frac{1}{(1+e^{x_s})(1+e^{-x_s})}$
Tanh	$y_s = \tanh(x_s)$	$\left[\frac{\partial E}{\partial x} \right]_s = \left[\frac{\partial E}{\partial y} \right]_s \frac{1}{\cosh^2 x_s}$
ReLU	$y_s = \max(0, x_s)$	$\left[\frac{\partial E}{\partial x} \right]_s = \left[\frac{\partial E}{\partial y} \right]_s \mathbb{I}\{x_s > 0\}$
Ramp	$y_s = \min(-1, \max(1, x_s))$	$\left[\frac{\partial E}{\partial x} \right]_s = \left[\frac{\partial E}{\partial y} \right]_s \mathbb{I}\{-1 < x_s < 1\}$

Redes Neurais Artificiais



O modelo neuronal matemático também pode incluir uma polarização ou **bias** de entrada. Esta variável é incluída ao somatório da função de ativação, com o intuito de aumentar o grau de liberdade desta função e, consequentemente, a capacidade de aproximação da rede



Modelos Não Paramétricos



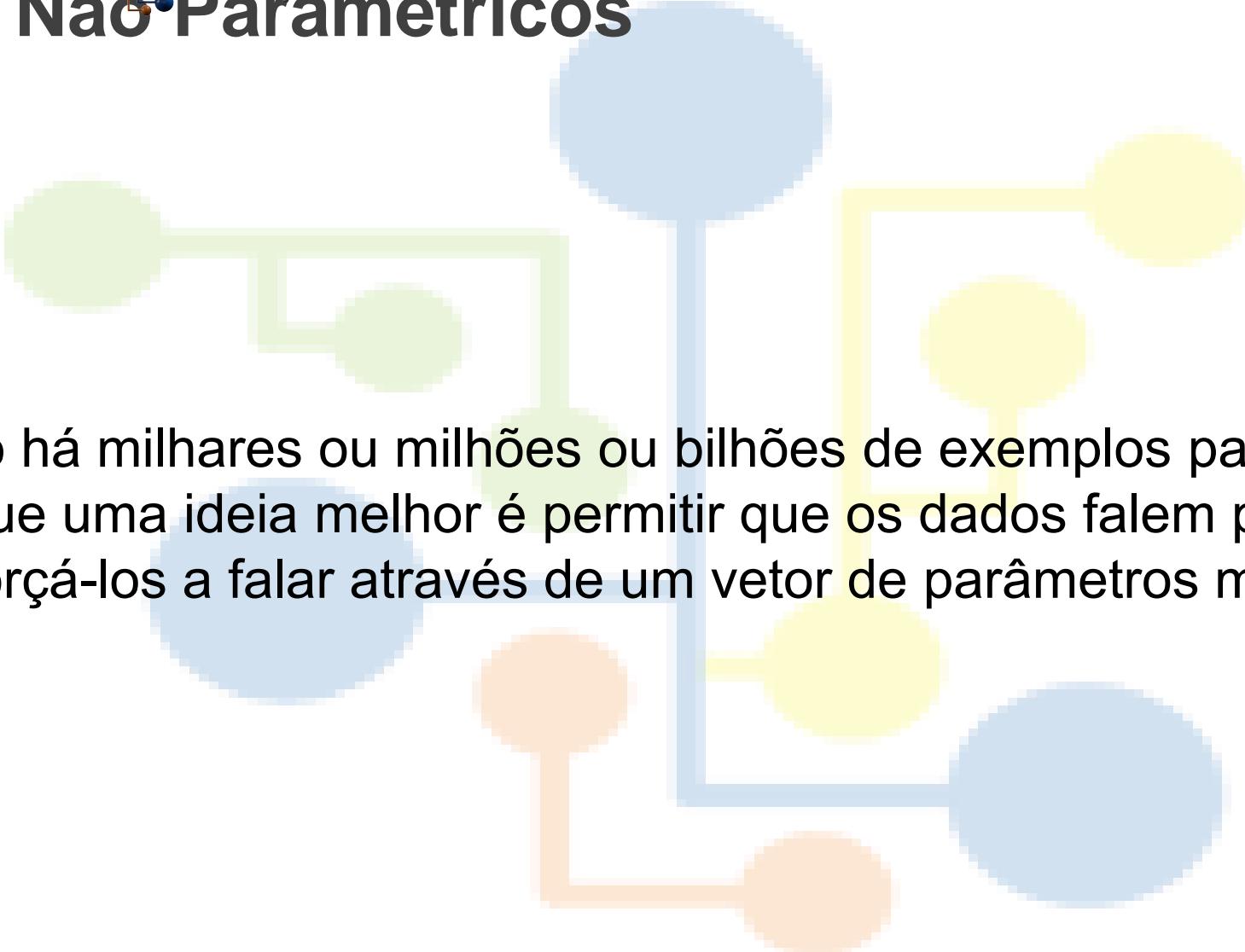
Modelos Não-Paramétricos

A regressão linear e as redes neurais utilizam dados de treinamento para estimar um conjunto fixo de parâmetros w . Isso define a nossa hipótese $hw(x)$, e nesse ponto podemos jogar fora os dados de treinamento porque todos eles estão resumidos por w .

Modelos Não-Paramétricos

Um modelo de aprendizagem que resume os dados com um conjunto de parâmetros de tamanho fixo (independentemente do número de exemplos de treinamento) é chamado de modelo paramétrico.

Modelos Não-Paramétricos



Quando há milhares ou milhões ou bilhões de exemplos para aprender, parece que uma ideia melhor é permitir que os dados falem por si ao invés de forçá-los a falar através de um vetor de parâmetros minúsculo.

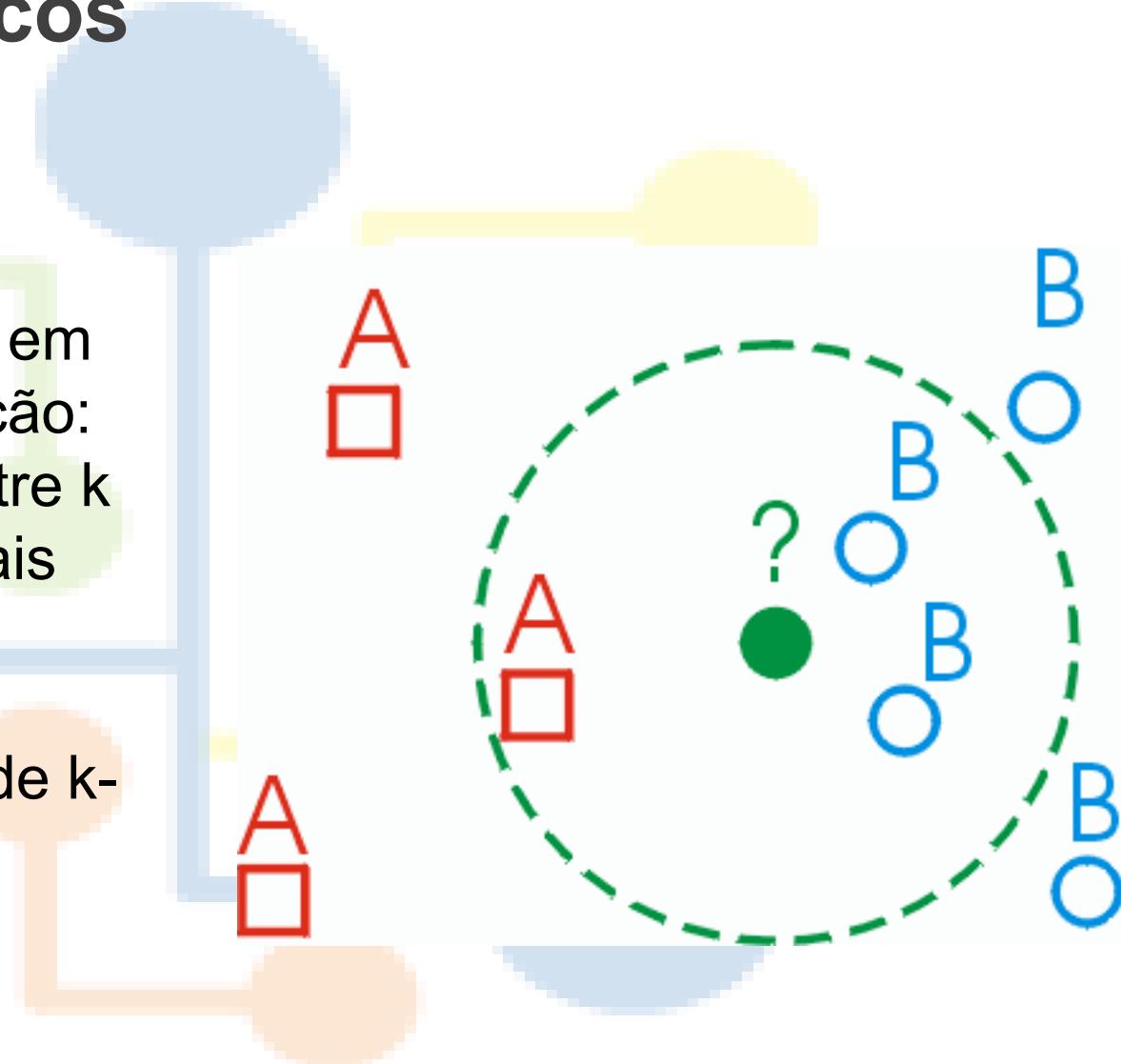
Modelos Não Paramétricos

Um modelo não paramétrico é aquele que não pode ser caracterizado por um conjunto limitado de parâmetros.

Modelos Não-Paramétricos

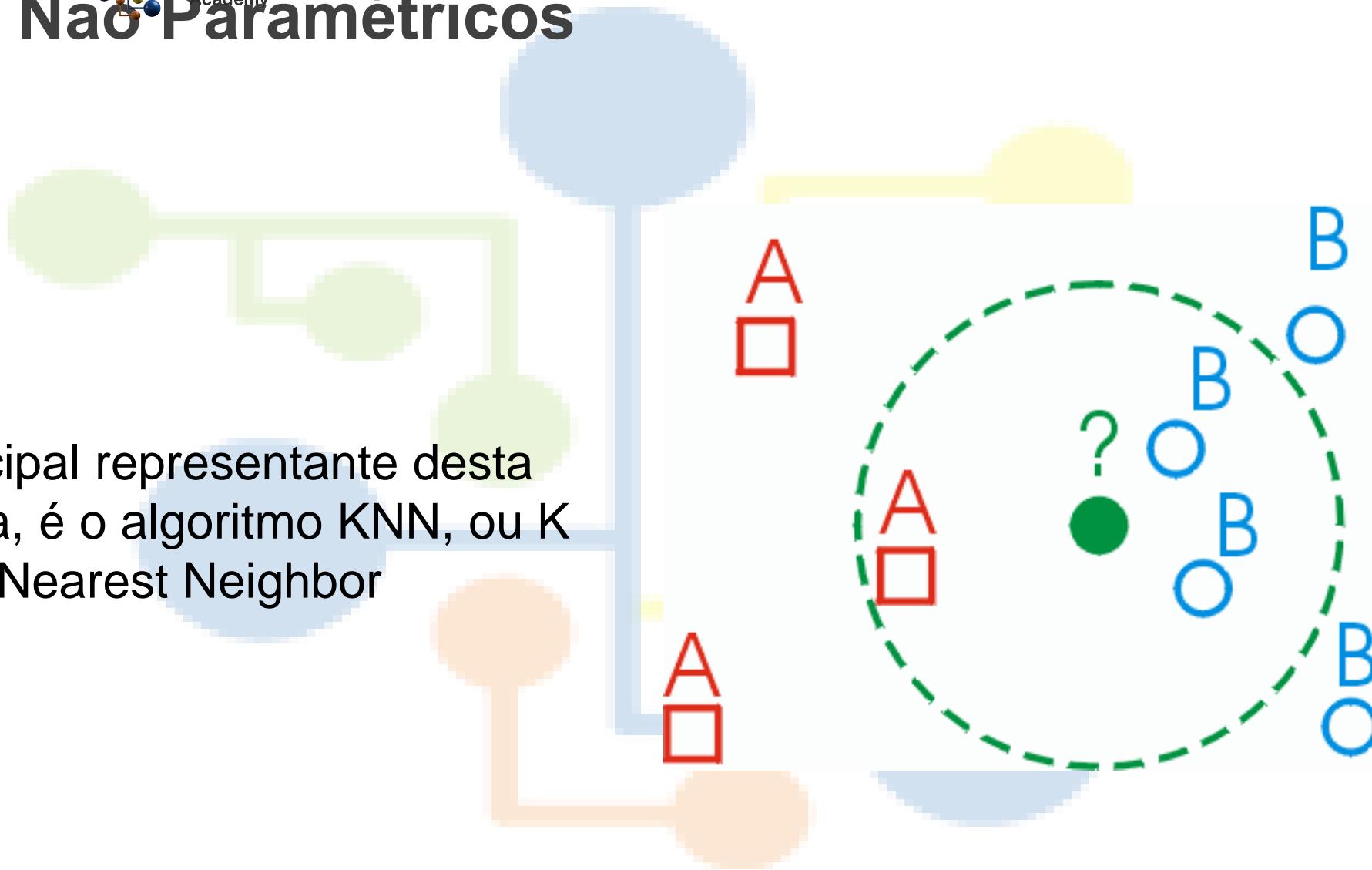
Podemos melhorar pesquisa em tabela com uma ligeira variação: dada uma consulta x_q , encontre k exemplos que estiverem mais próximas de x_q .

Isso é chamado de pesquisa de k -vizinhos mais próximos

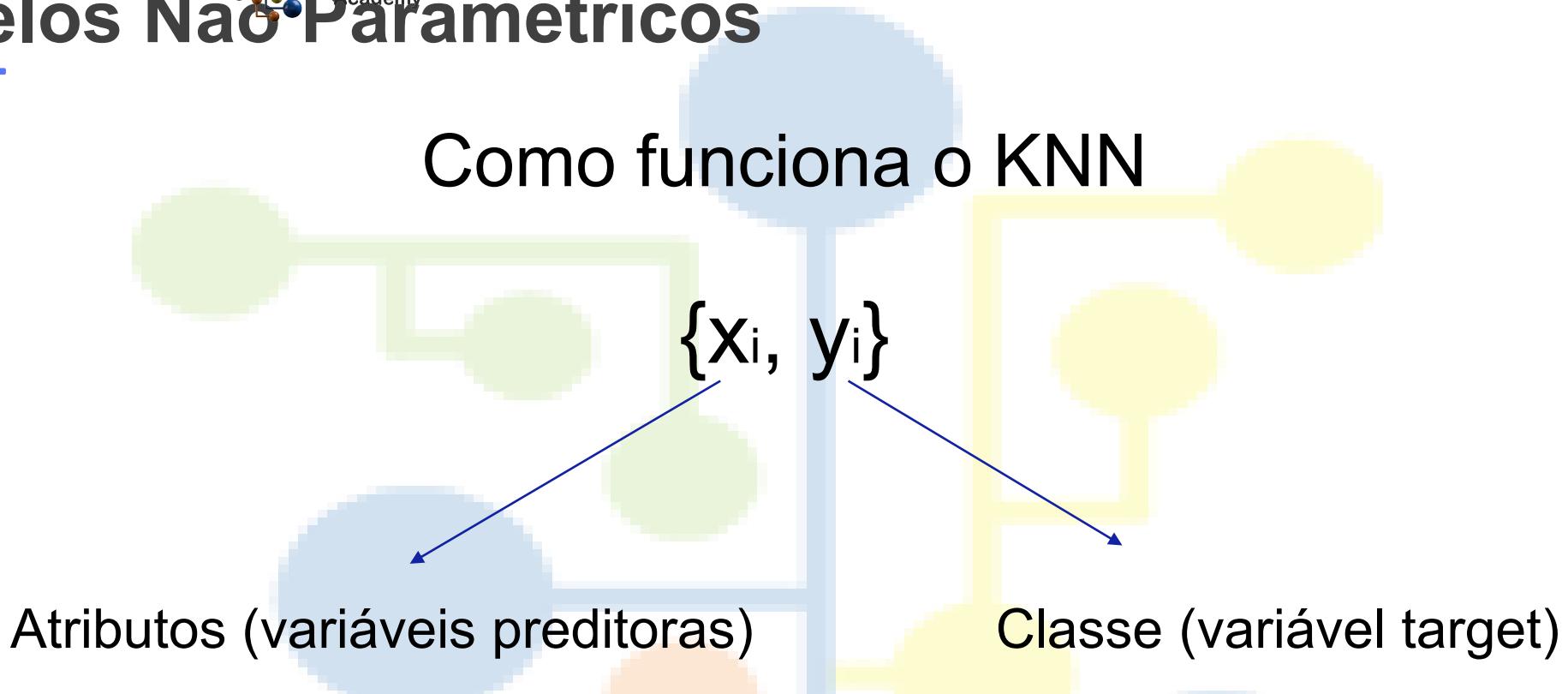


Modelos Não-Paramétricos

O principal representante desta categoria, é o algoritmo KNN, ou K Nearest Neighbor



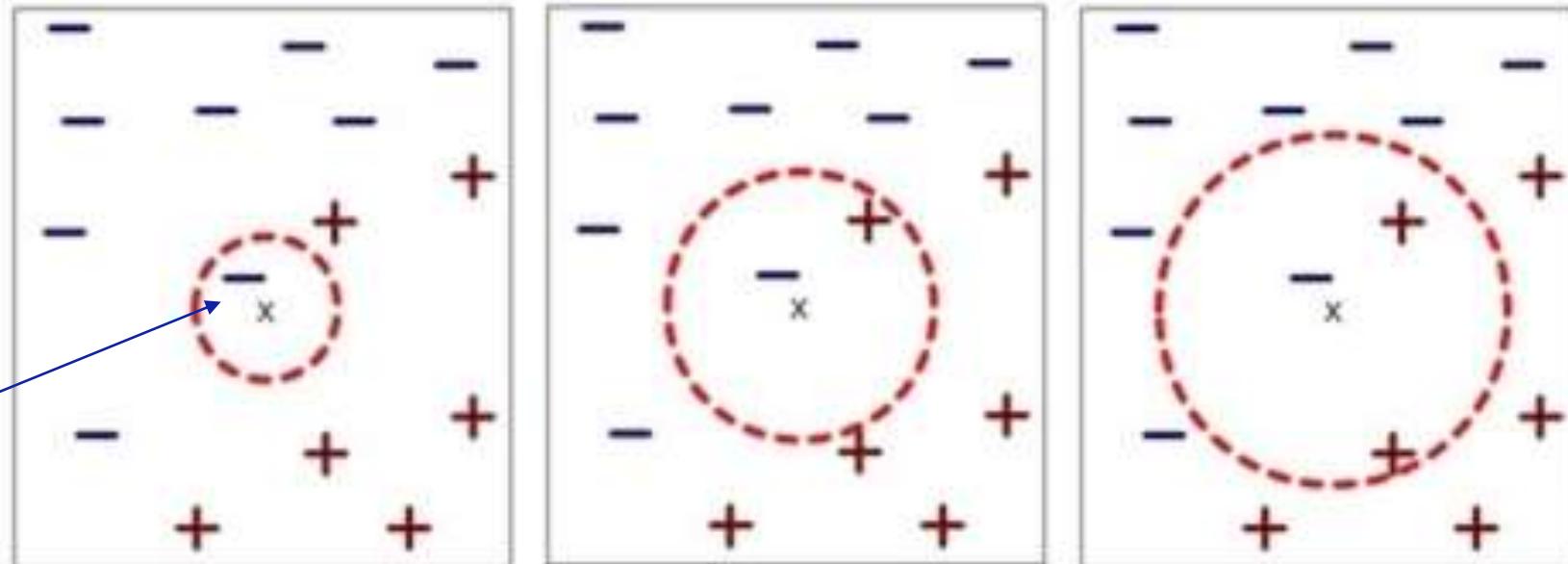
Como funciona o KNN



Como classificar um novo ponto de dado X

- 1- A distância é computada entre X e X_i para cada valor de X_i .
- 2- É escolhido o k-vizinho mais próximo X_{in} e sua respectiva classe.
- 3- Retorna-se o valor de y mais frequente na lista $y_{i1}, y_{i2}, \dots, y_{in}$.

Modelos Não-Paramétricos



Classificação do
novo ponto de
dado

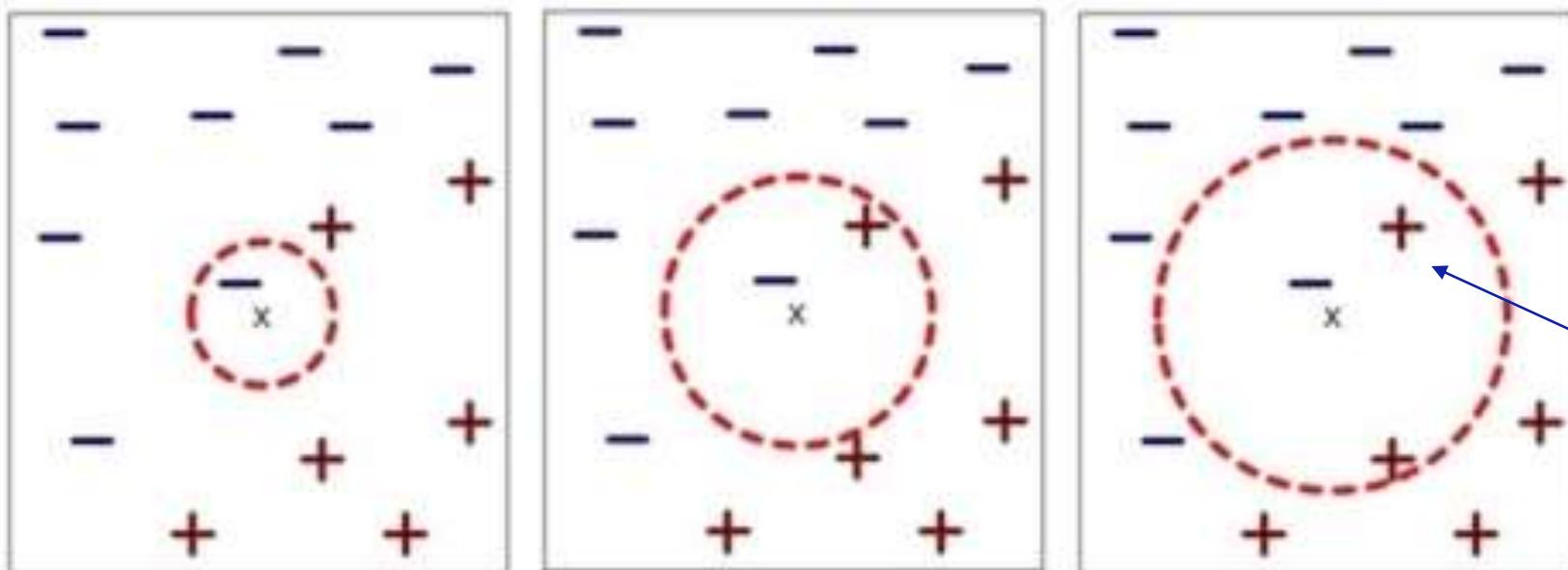
1NN

2NN

3NN

Valor de K

Modelos Não-Paramétricos



1NN

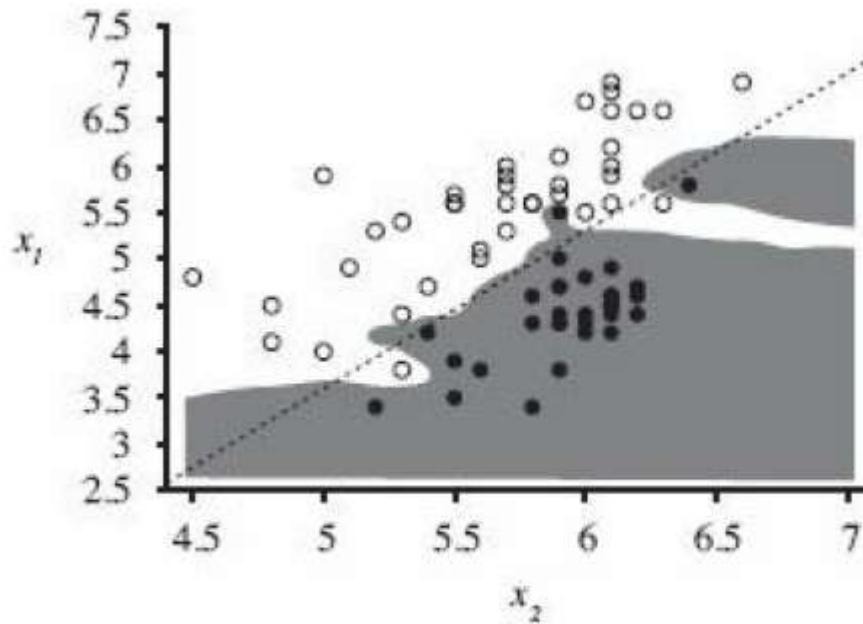
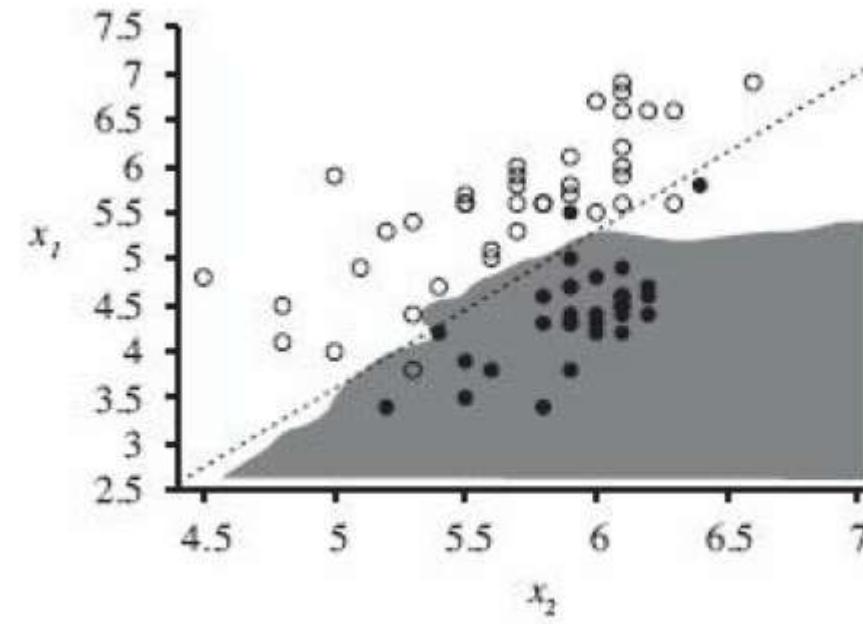
2NN

3NN

Valor de K

Classificação do
novo ponto de
dado

Modelos Não-Paramétricos

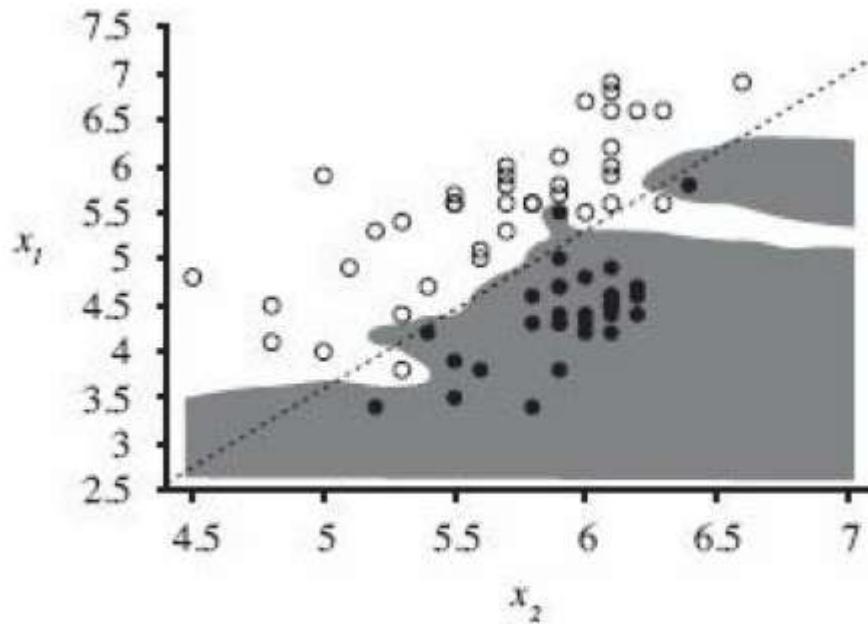
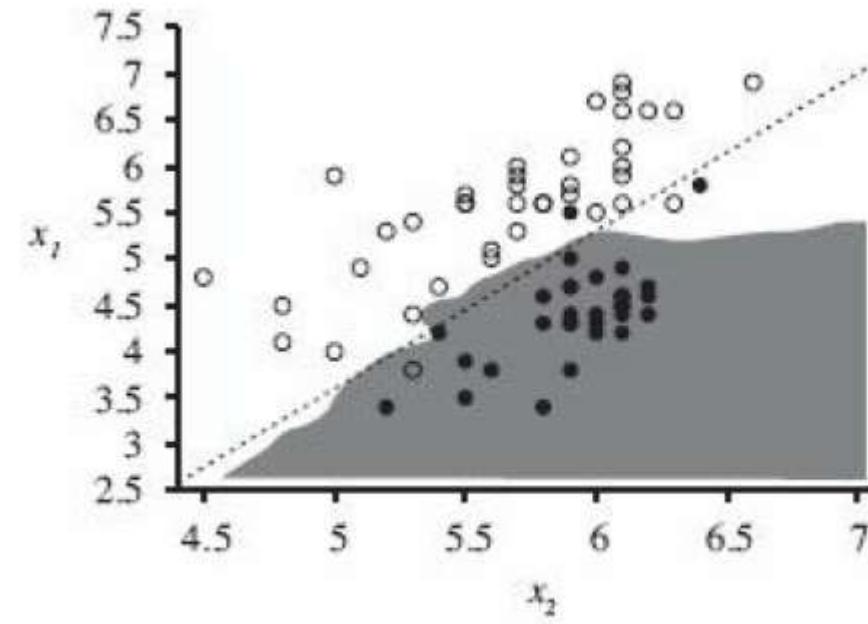
(a) ($k = 1$)(b) ($k = 5$)

$$L^p(\mathbf{x}_j, \mathbf{x}_q) = \left(\sum_i |x_{j,i} - x_{q,i}|^p \right)^{1/p}$$



Existem diversas medidas de distância disponíveis. O principal propósito da medida de distância é identificar os dados que são similares e que não são similares.

Modelos Não-Paramétricos

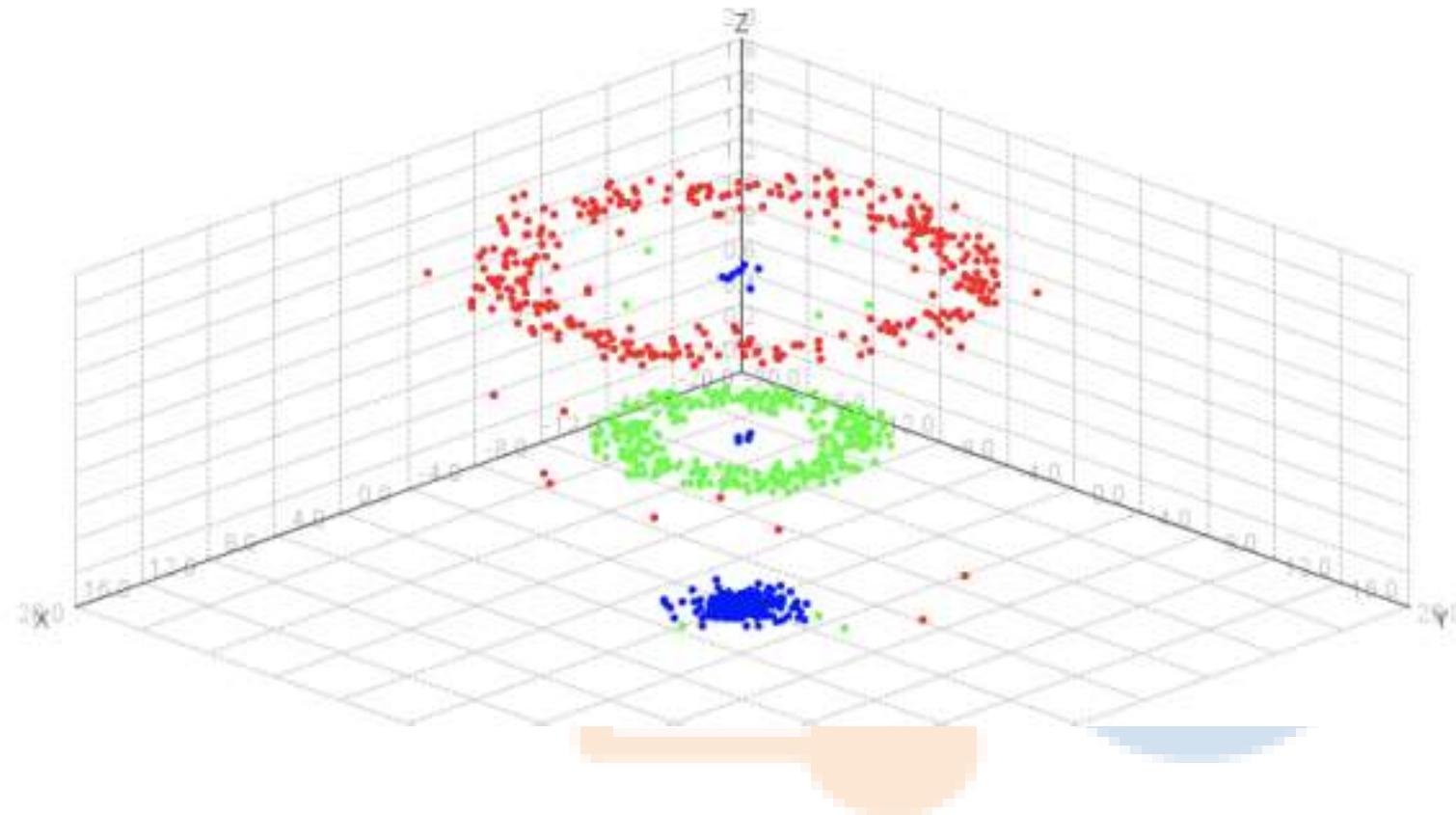
(a) ($k = 1$)(b) ($k = 5$)

$$L^p(\mathbf{x}_j, \mathbf{x}_q) = \left(\sum_i |x_{j,i} - x_{q,i}|^p \right)^{1/p}$$

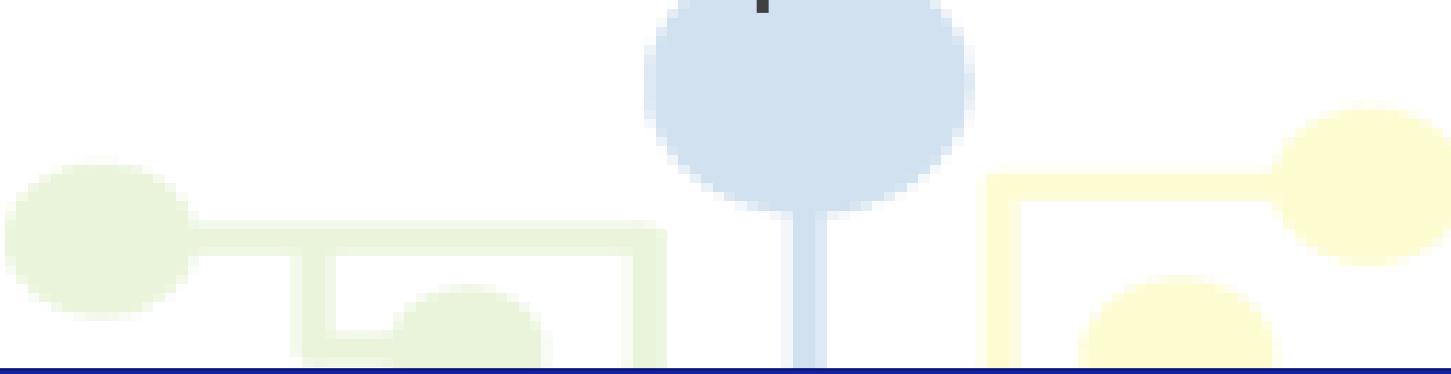
Máquinas de Vetores de Suporte



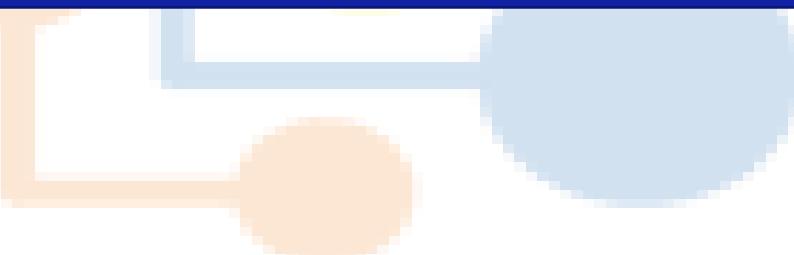
Máquinas de Vetores de Suporte



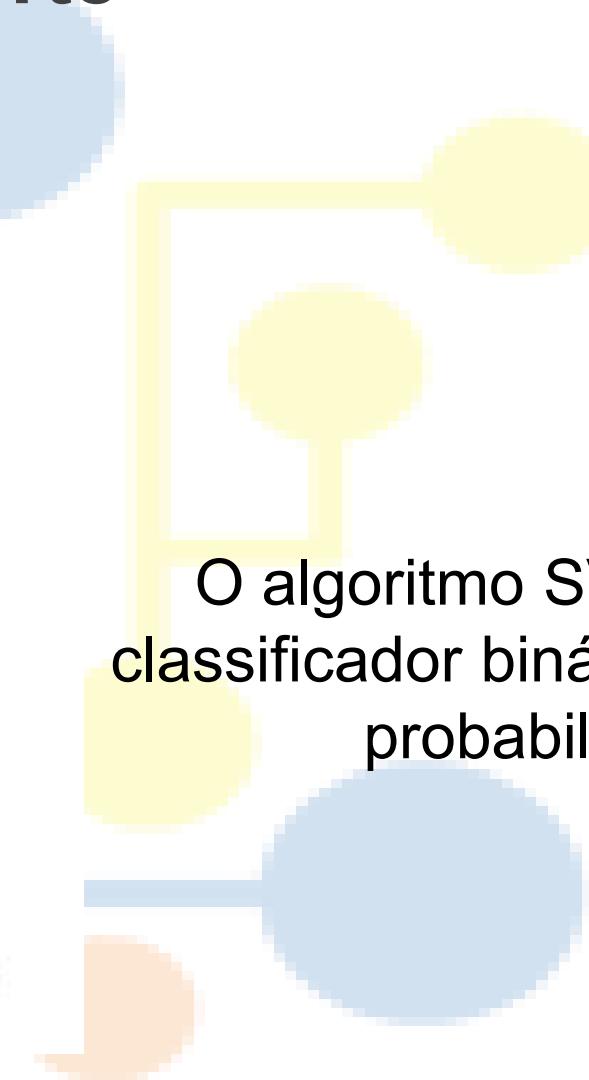
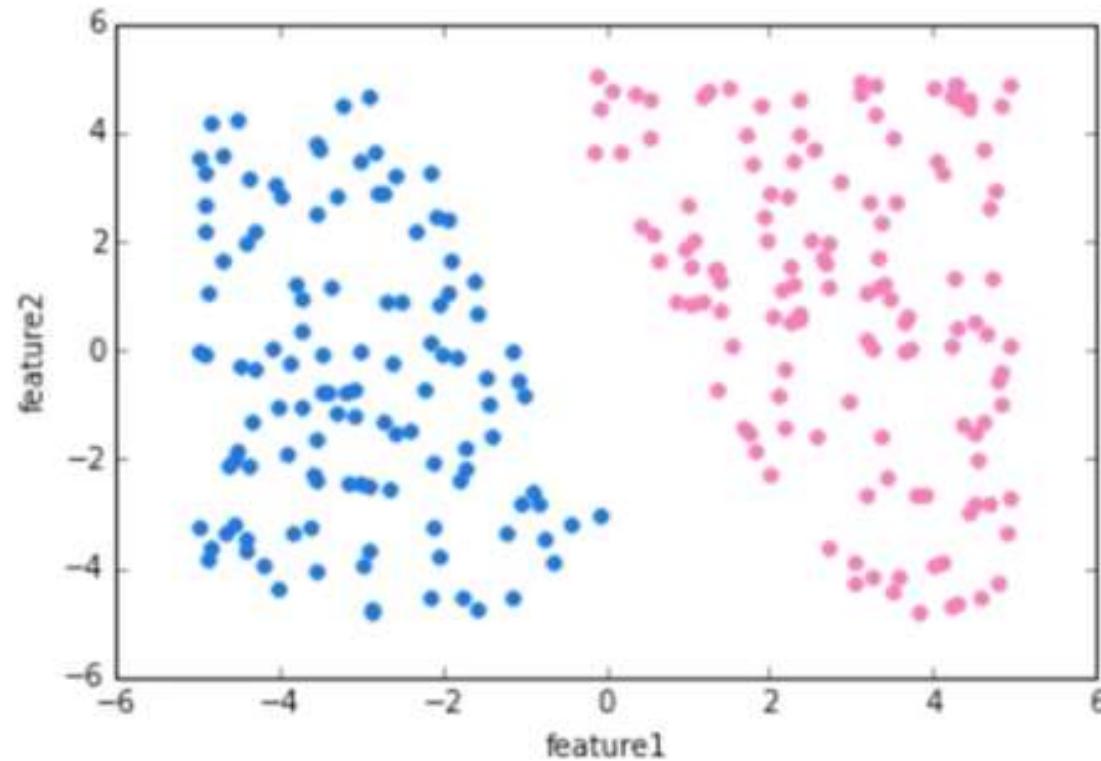
Máquinas de Vetores de Suporte



Support Vector Machines (SVM's) são modelos de aprendizagem supervisionada, que possuem algoritmos de aprendizagem que analisam dados e reconhecem padrões, utilizados para classificação e análise de regressão.

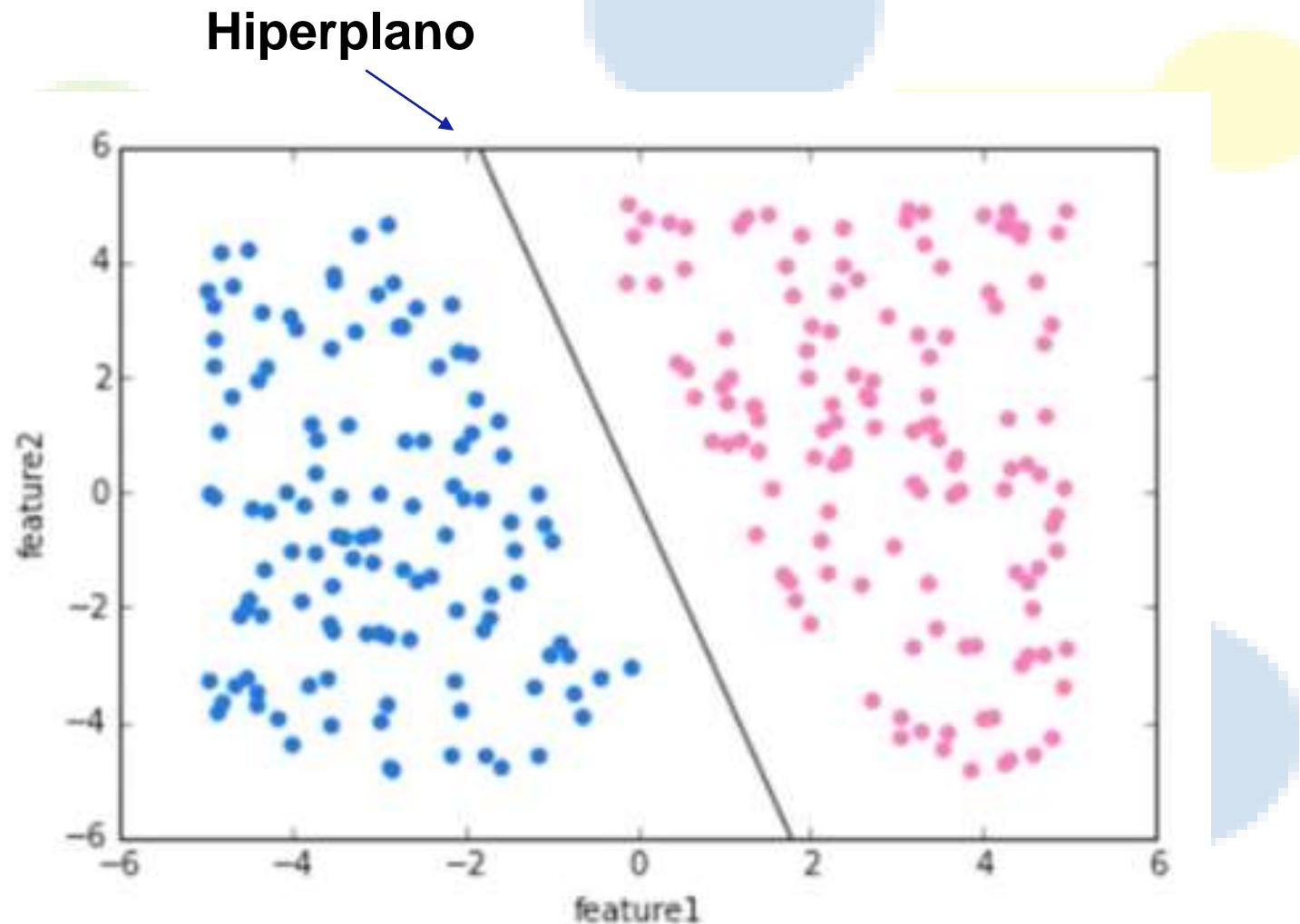


Máquinas de Vetores de Suporte

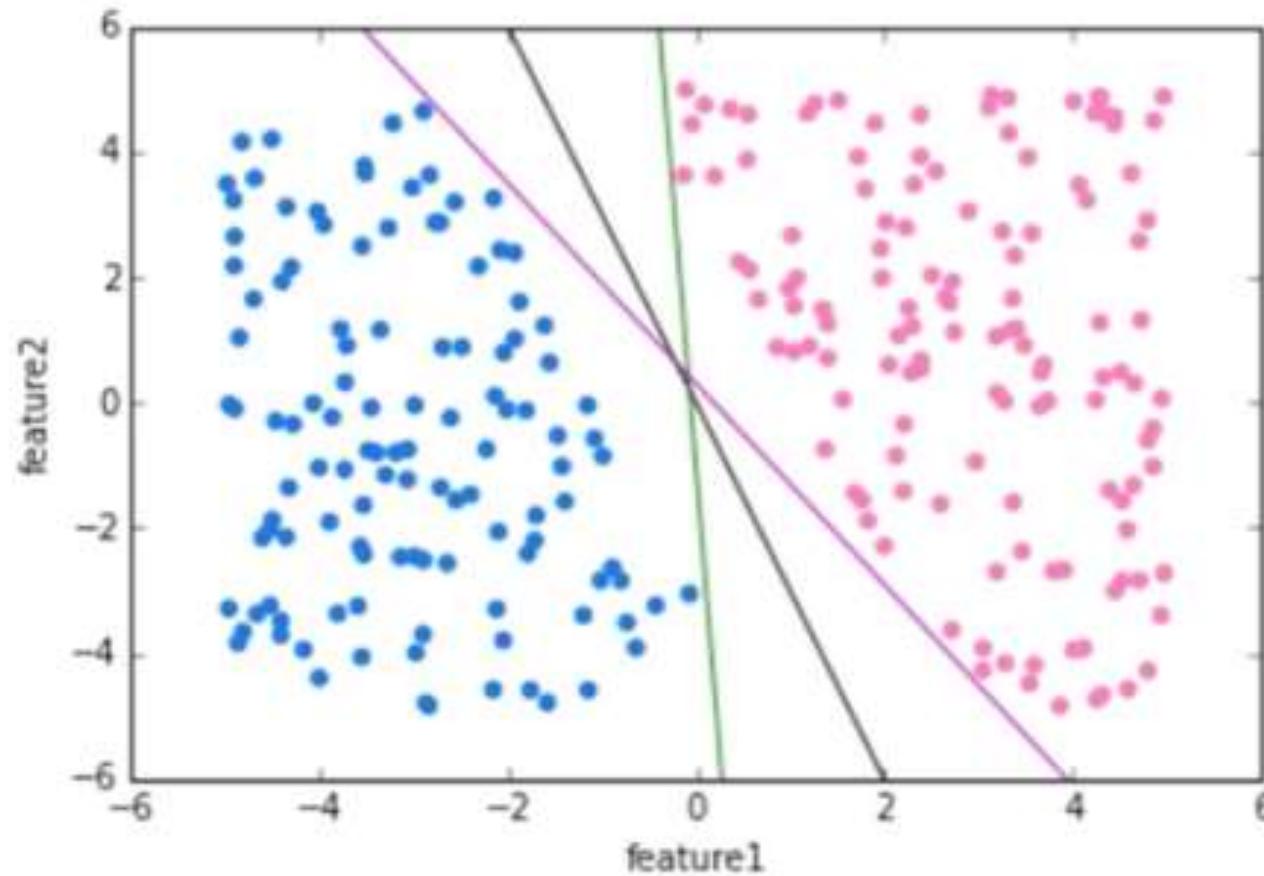


O algoritmo SVM cria um classificador binário linear não-probabilístico

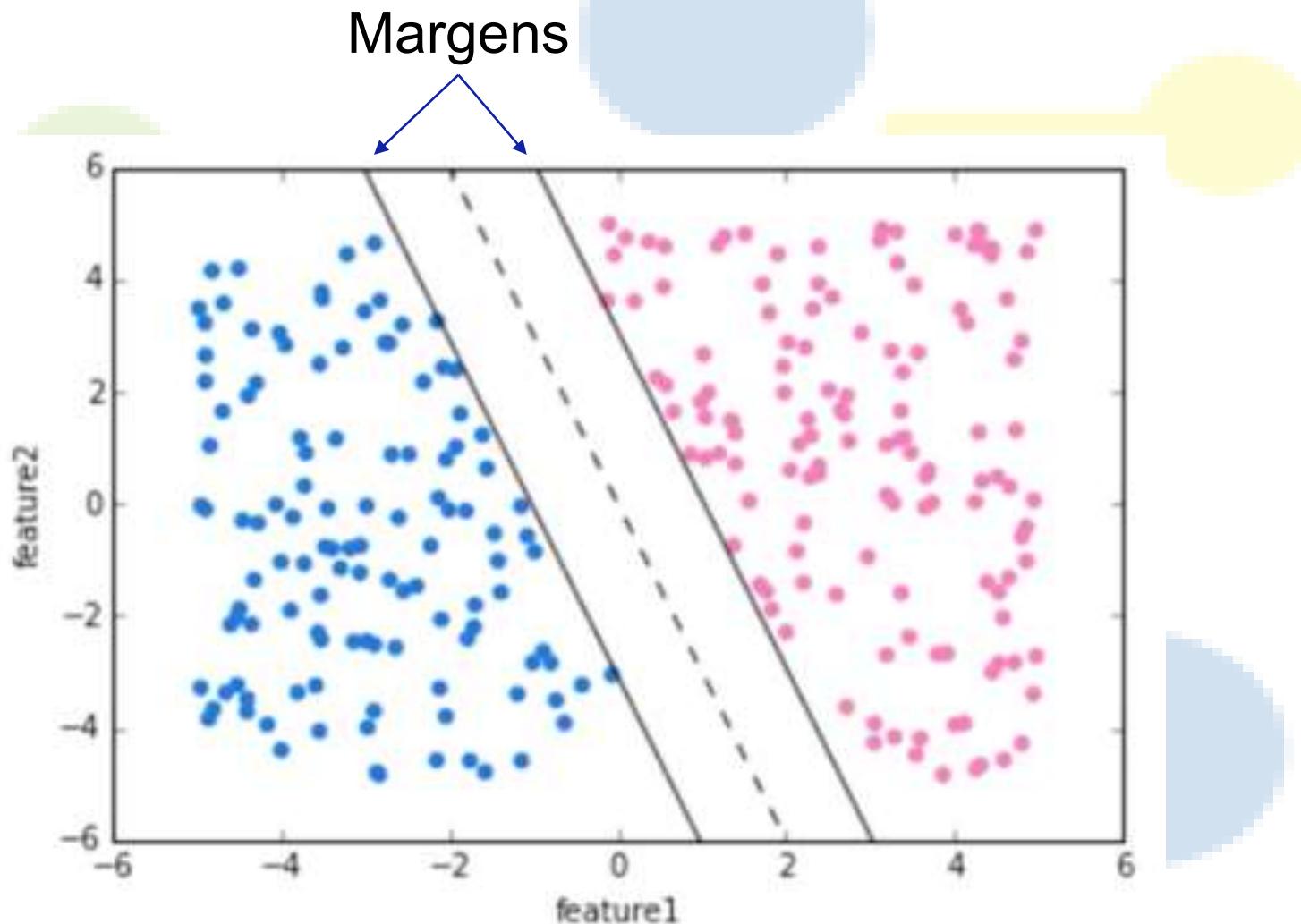
Máquinas de Vetores de Suporte



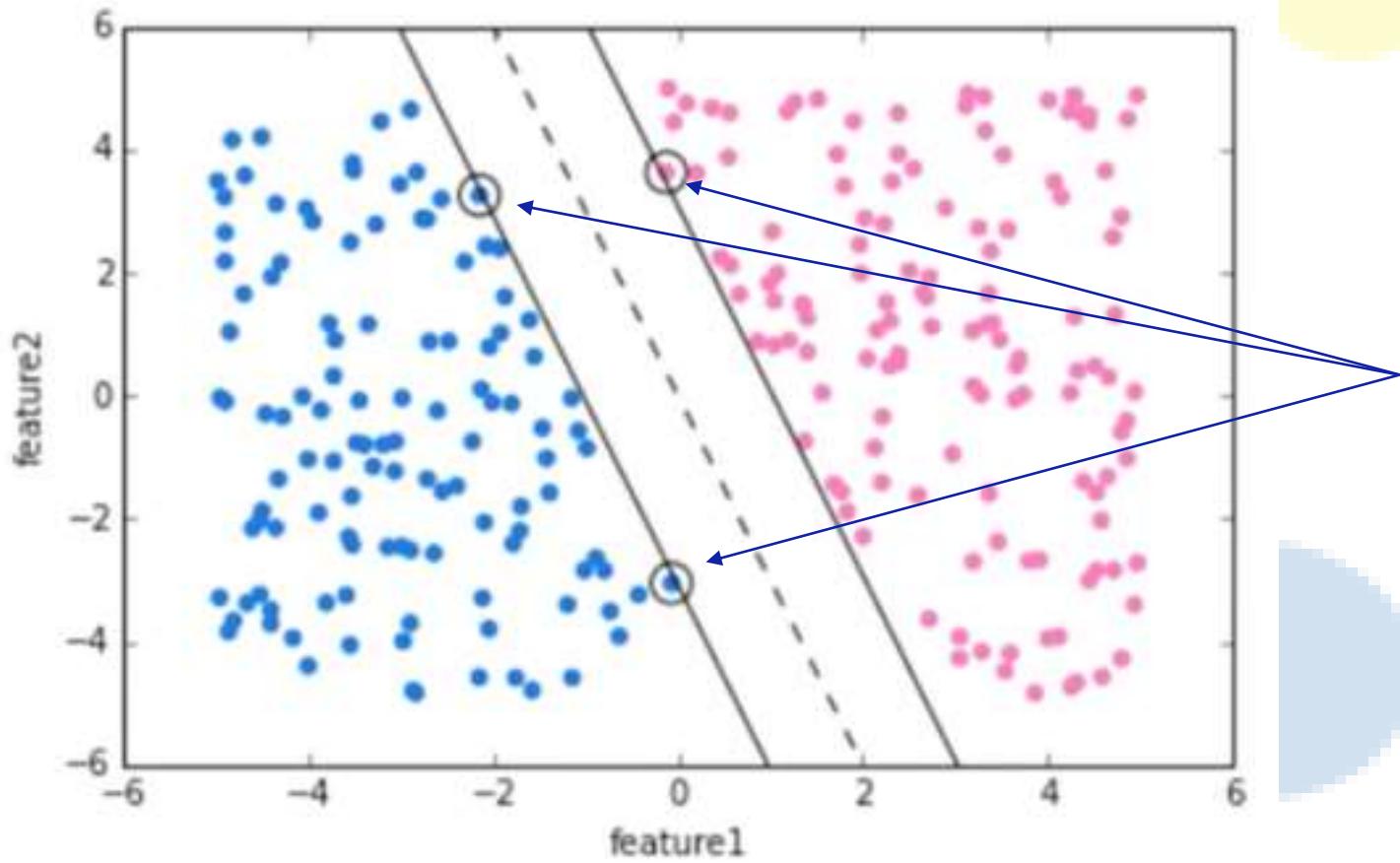
Máquinas de Vetores de Suporte



Máquinas de Vetores de Suporte

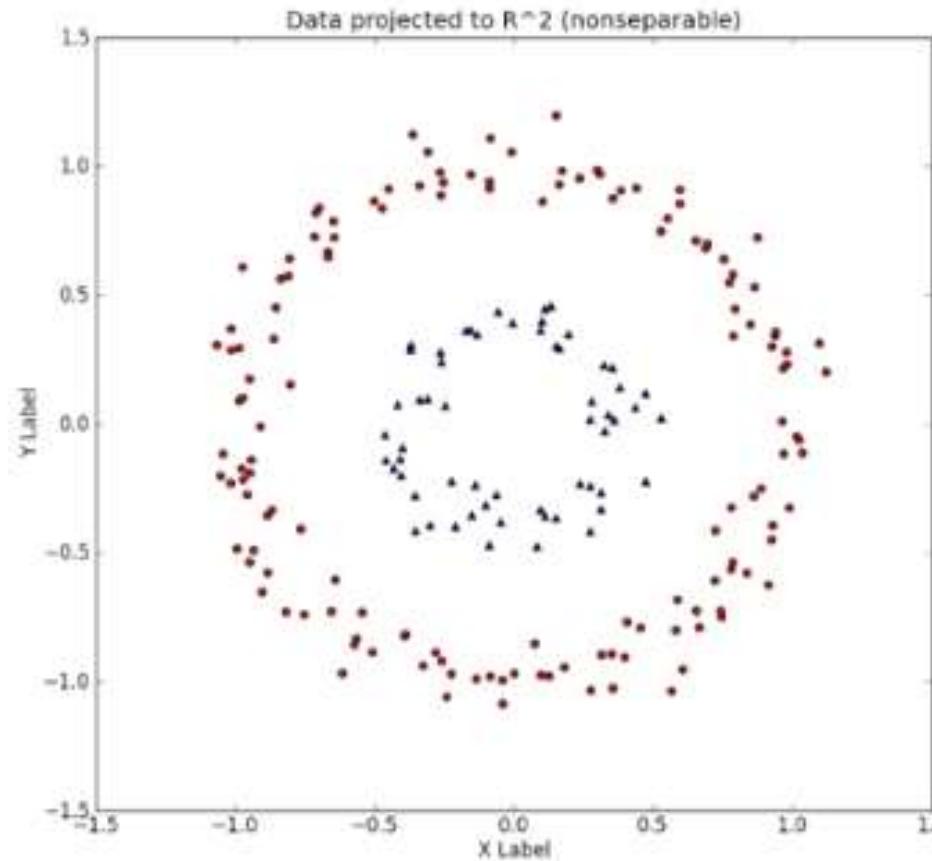


Máquinas de Vetores de Suporte



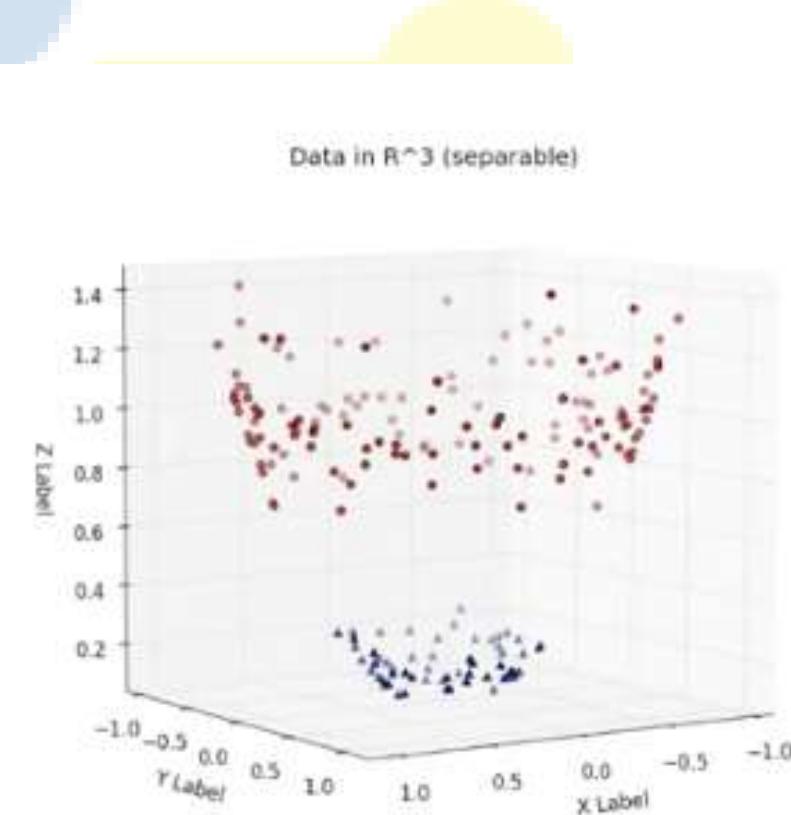
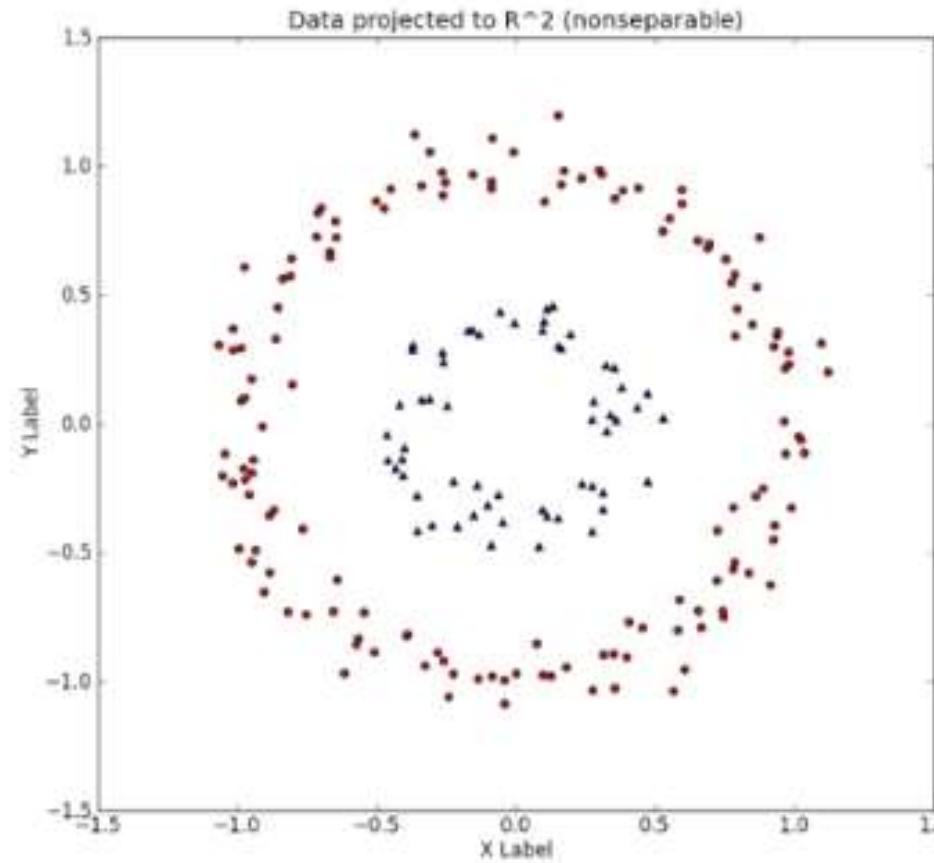
Vetores de
Suporte

Máquinas de Vetores de Suporte

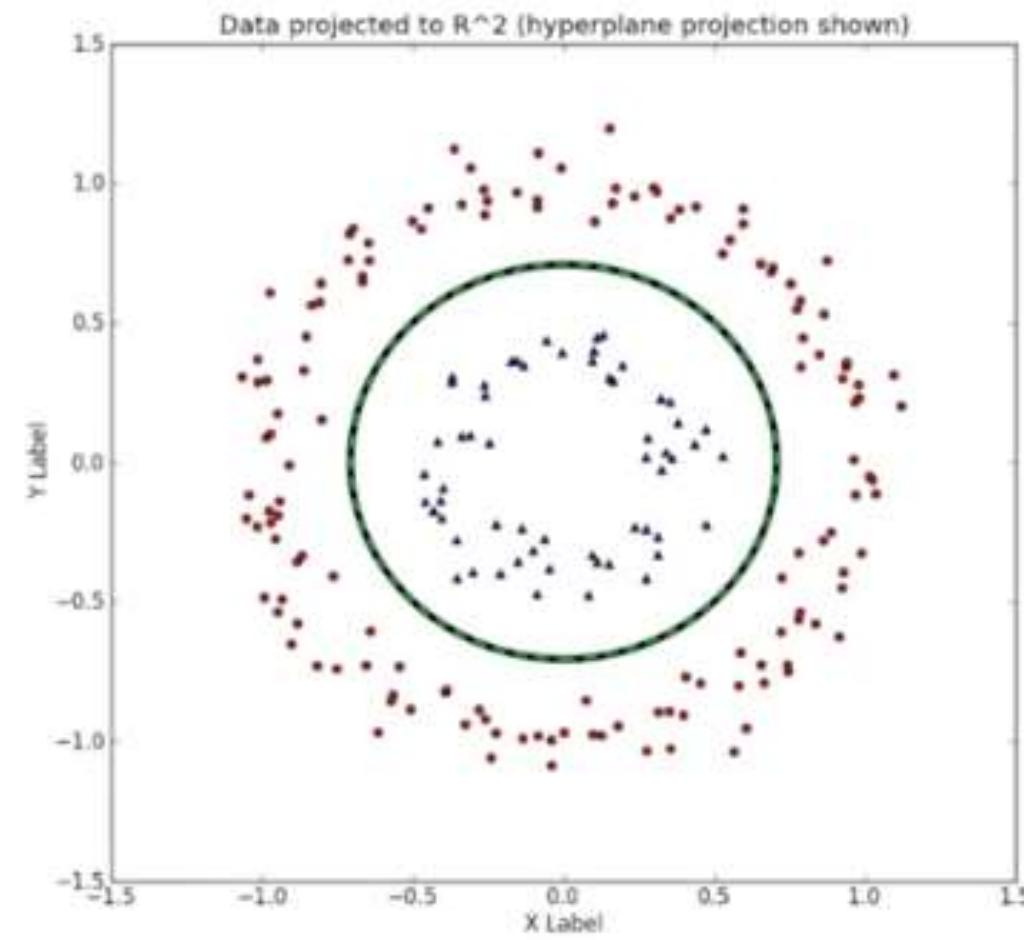
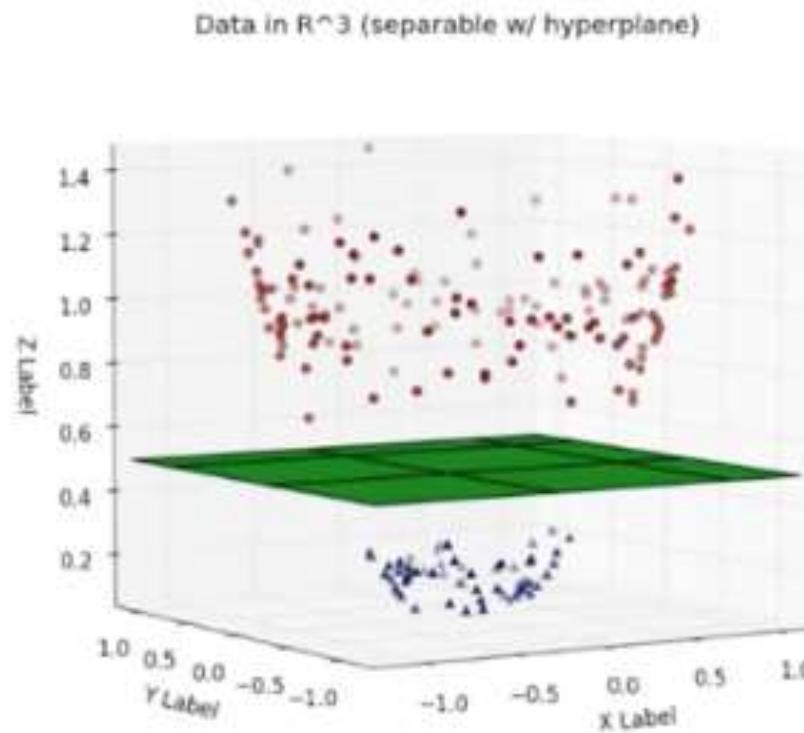


Mas e quando os dados são **não**
linearmente separáveis?

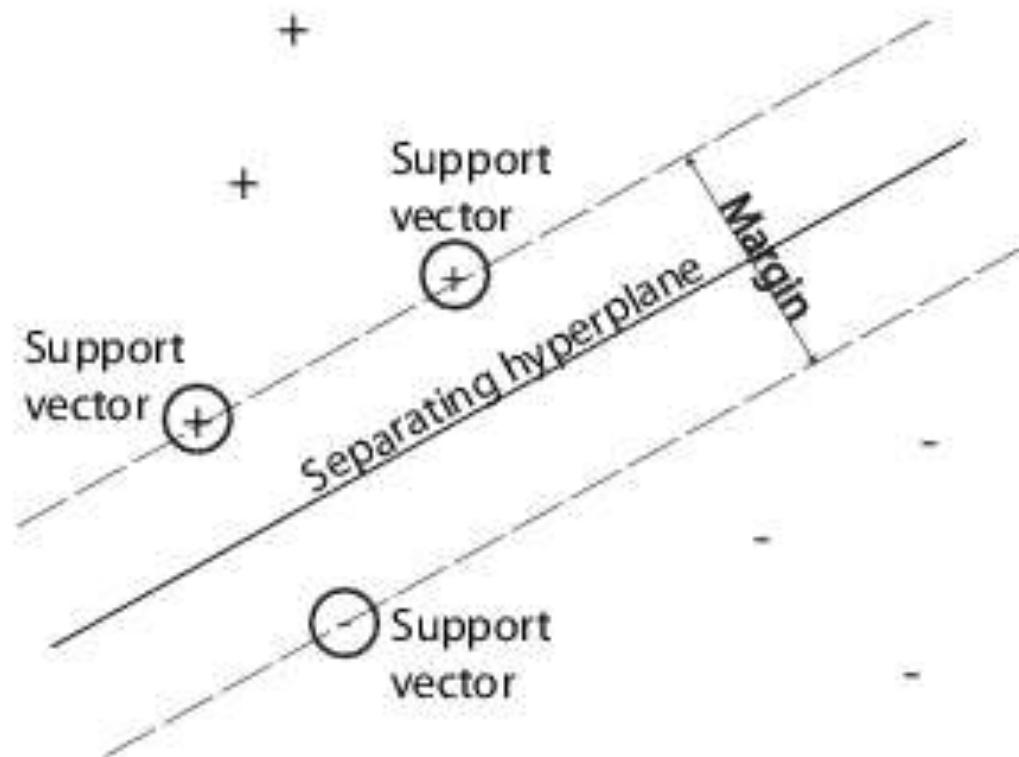
Máquinas de Vetores de Suporte



Máquinas de Vetores de Suporte



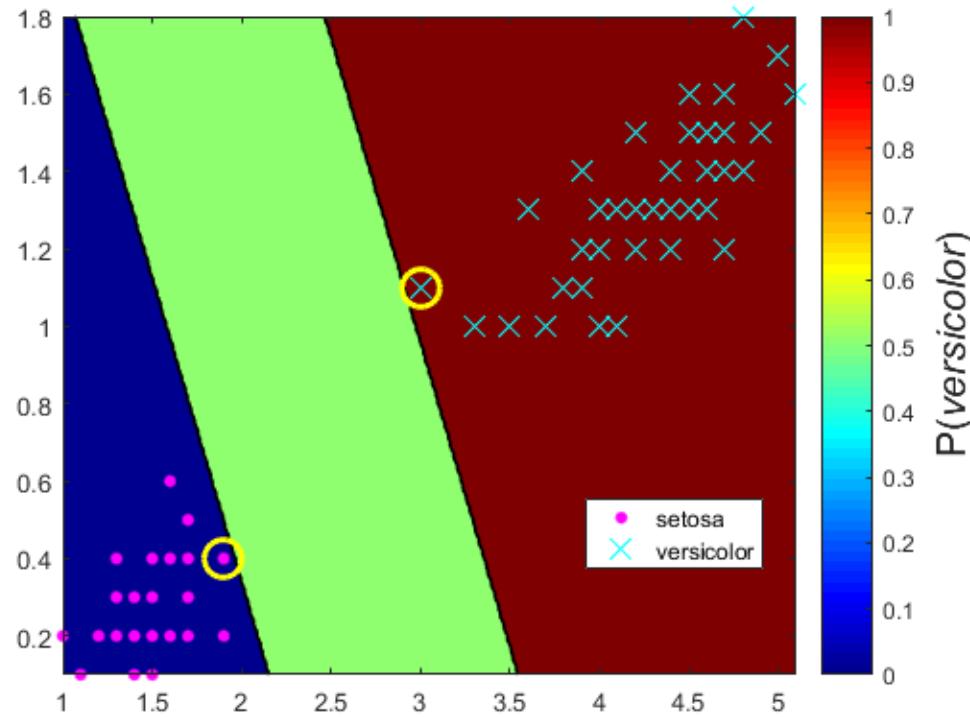
Máquinas de Vetores de Suporte

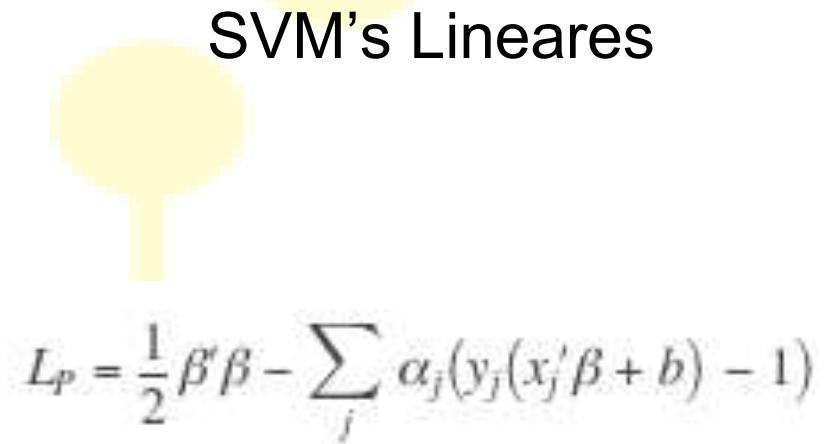


SVM's Lineares

O melhor hiperplano para uma SVM significa aquele com a maior margem entre as duas classes

Máquinas de Vetores de Suporte

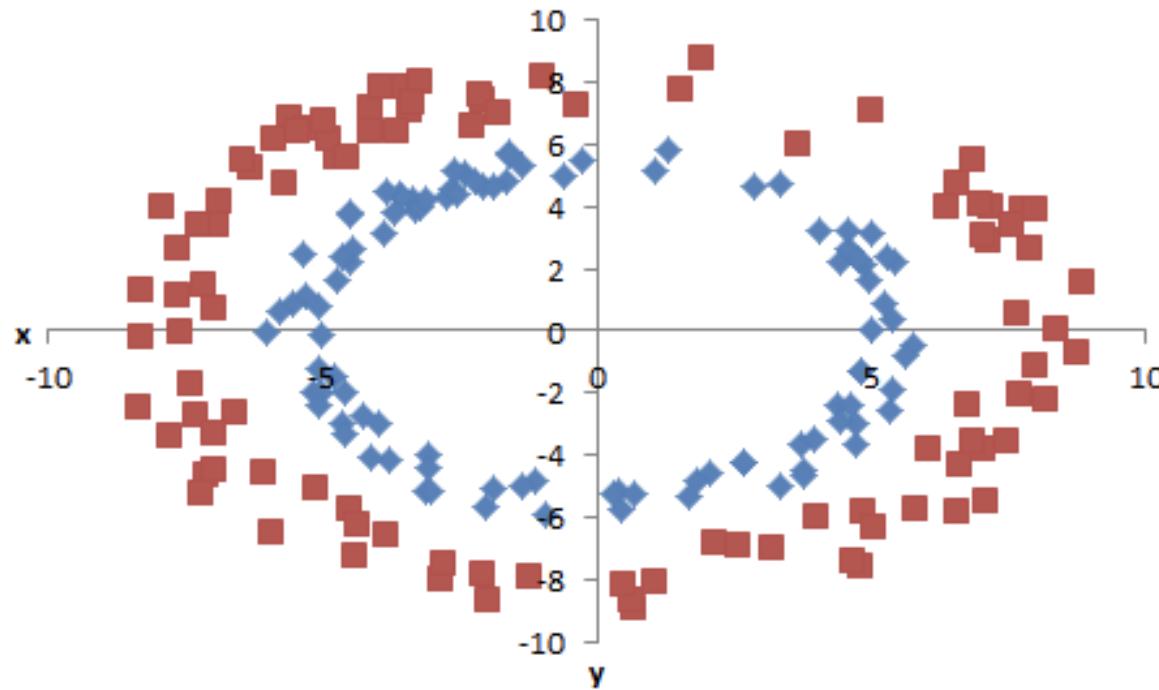




SVM's Lineares

$$L_P = \frac{1}{2} \beta' \beta - \sum_j \alpha_j (y_j (\beta' x_j + b) - 1)$$

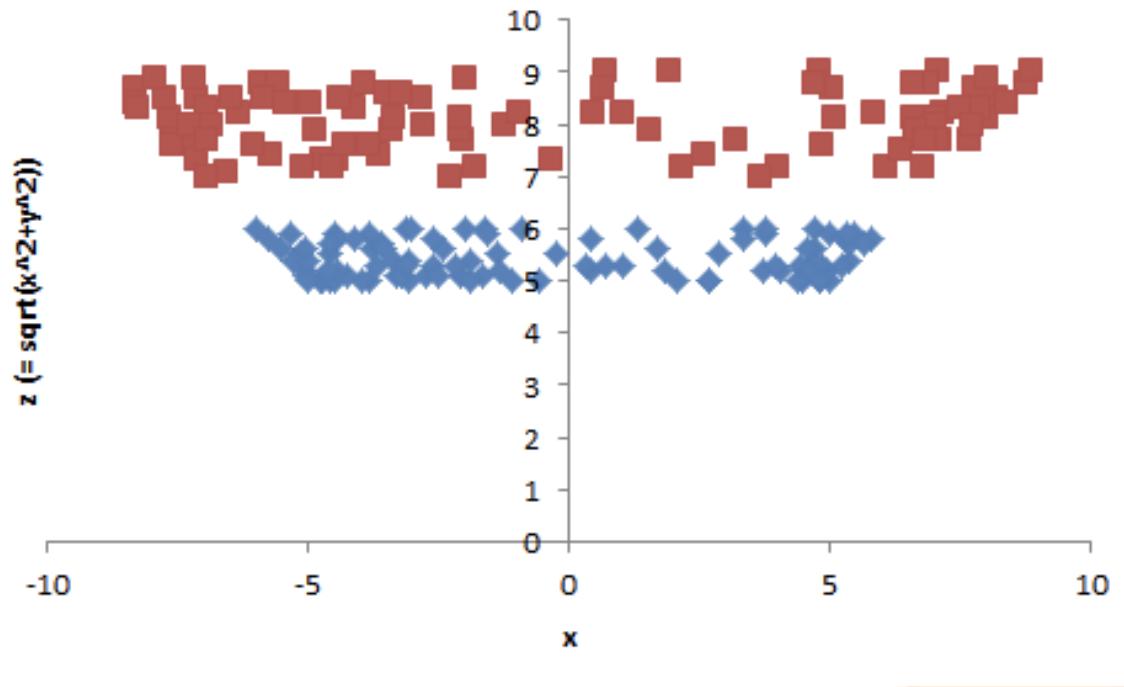
Máquinas de Vetores de Suporte



SVM's Não Lineares

Alguns problemas de classificação binária não têm um hiperplano simples como um critério de separação útil

Máquinas de Vetores de Suporte



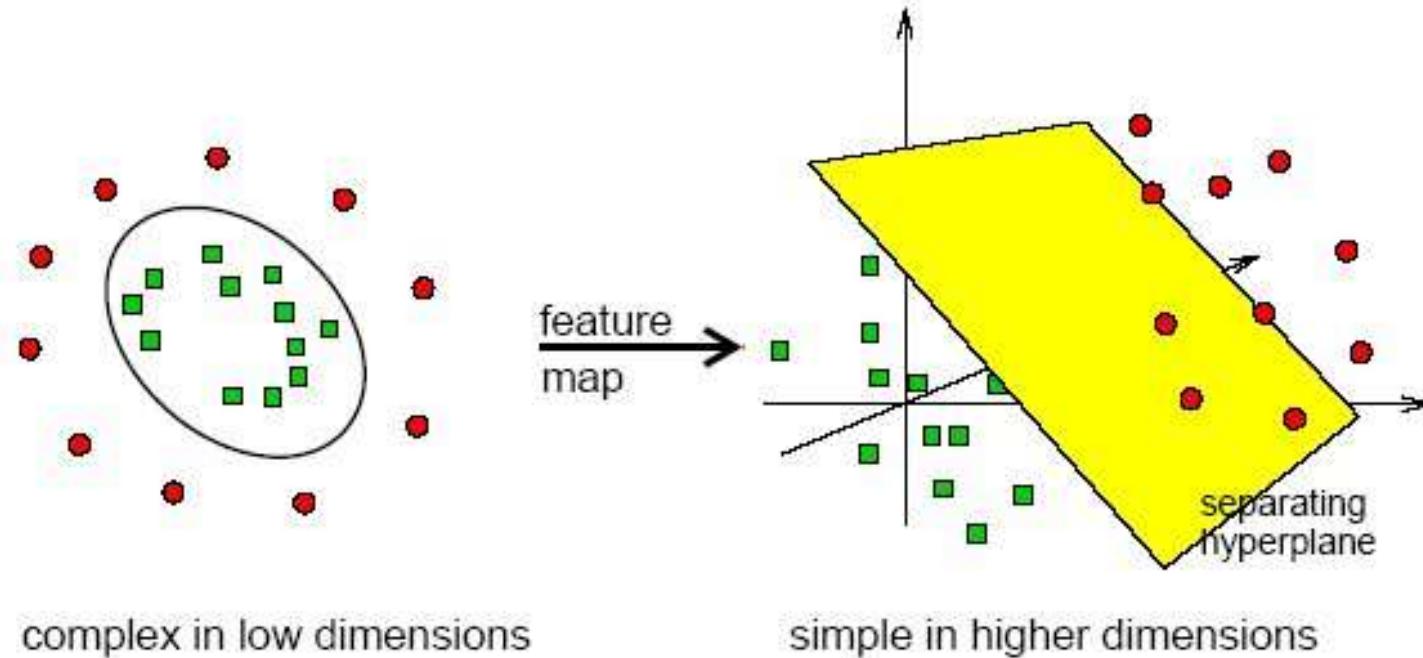
SVM's Não Lineares

Um truque simples seria transformar as duas variáveis x e y em um novo espaço de característica envolvendo x (ou y) e uma nova variável z definida como $z = \sqrt{x^2 + y^2}$

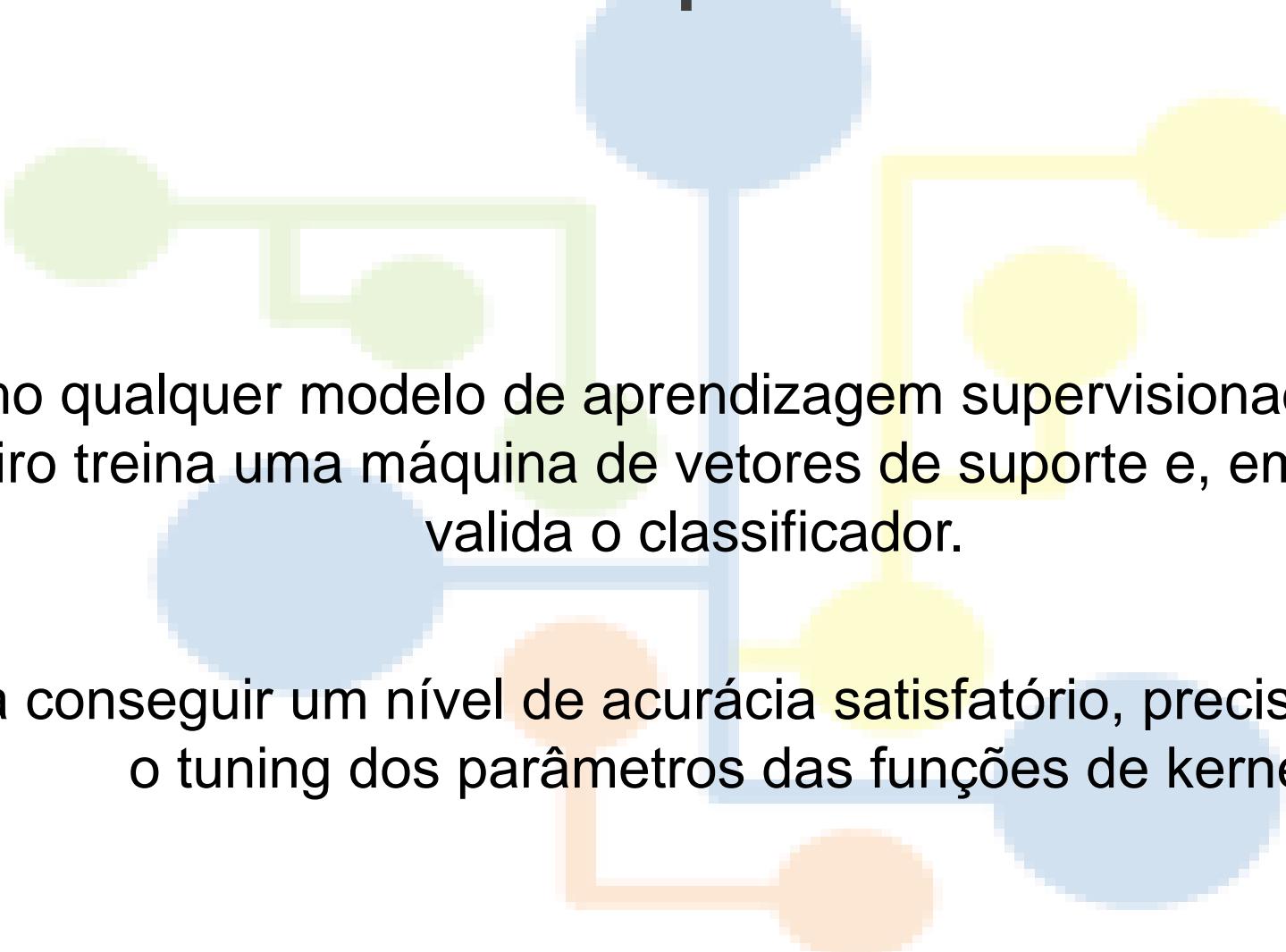
Máquinas de Vetores de Suporte

Função Kernel

Separation may be easier in higher dimensions



Máquinas de Vetores de Suporte



Como qualquer modelo de aprendizagem supervisionado, você primeiro treina uma máquina de vetores de suporte e, em seguida, valida o classificador.

Para conseguir um nível de acurácia satisfatório, precisamos fazer o tuning dos parâmetros das funções de kernel!



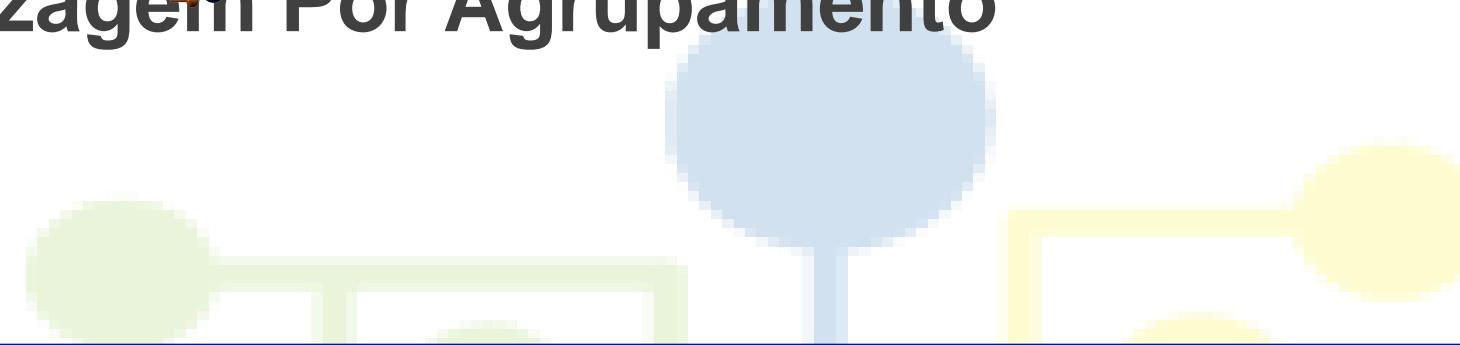
Data Science Academy raphaelbsfontenelle@gmail.com 615c1fdde32fc361b30c9ec2

Aprendizagem Por Agrupamento



Data Science Academy

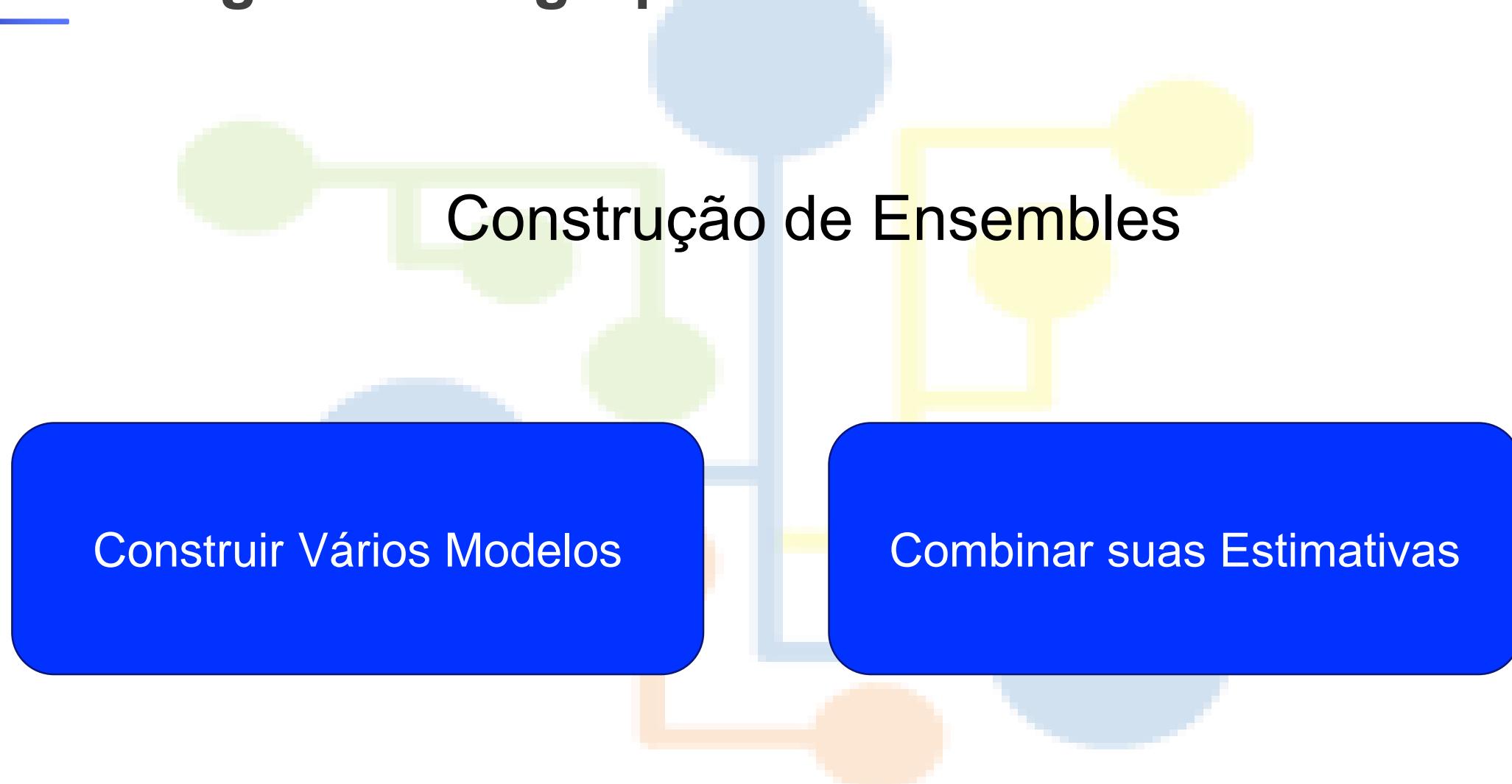
Aprendizagem Por Agrupamento



A ideia de métodos de aprendizagem por agrupamento (também chamada método ensemble) é selecionar uma coleção inteira ou um agrupamento de hipóteses, a partir do espaço de hipóteses, e combinar suas previsões

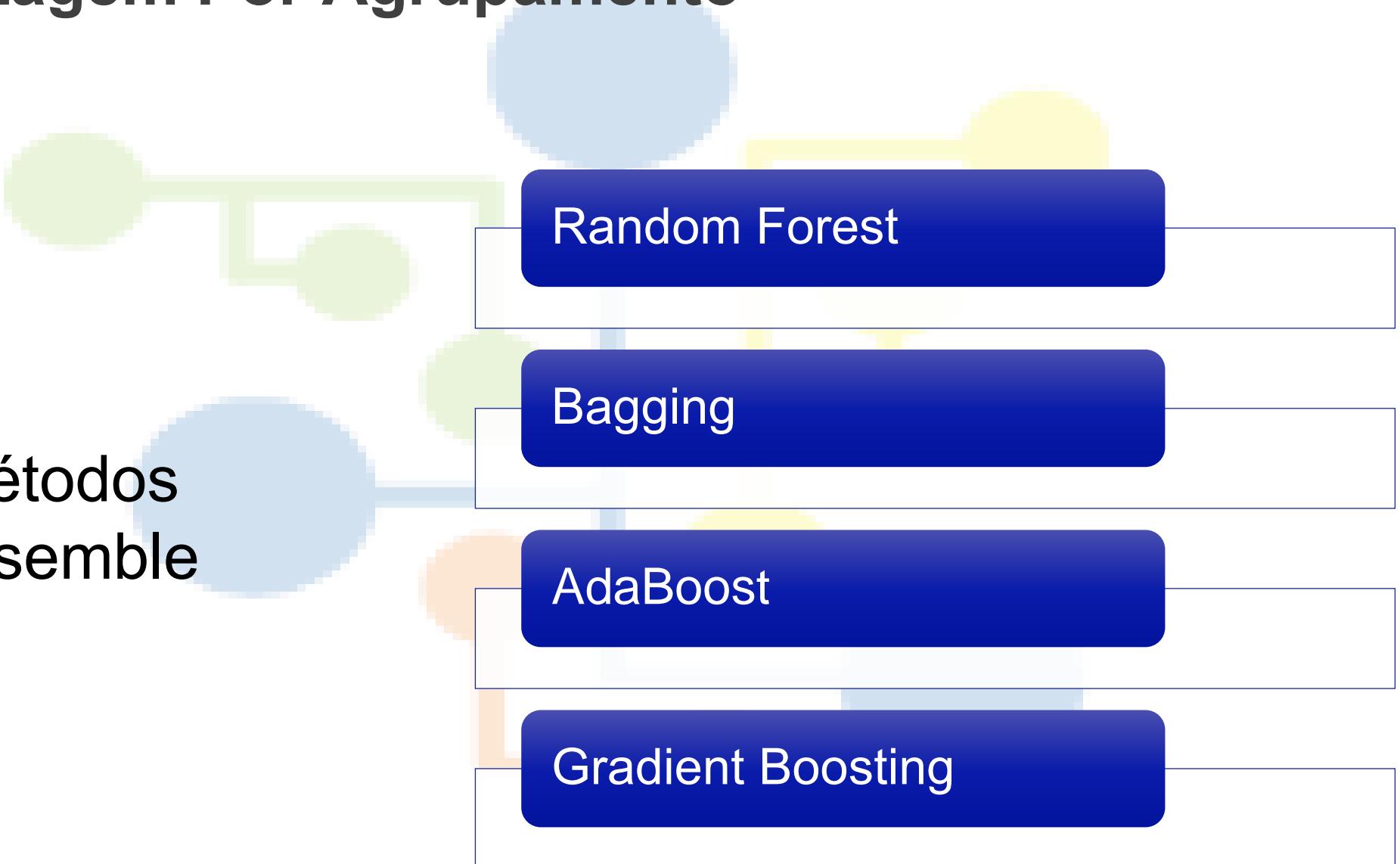


Aprendizagem Por Agrupamento



Aprendizagem Por Agrupamento

Métodos
Ensemble



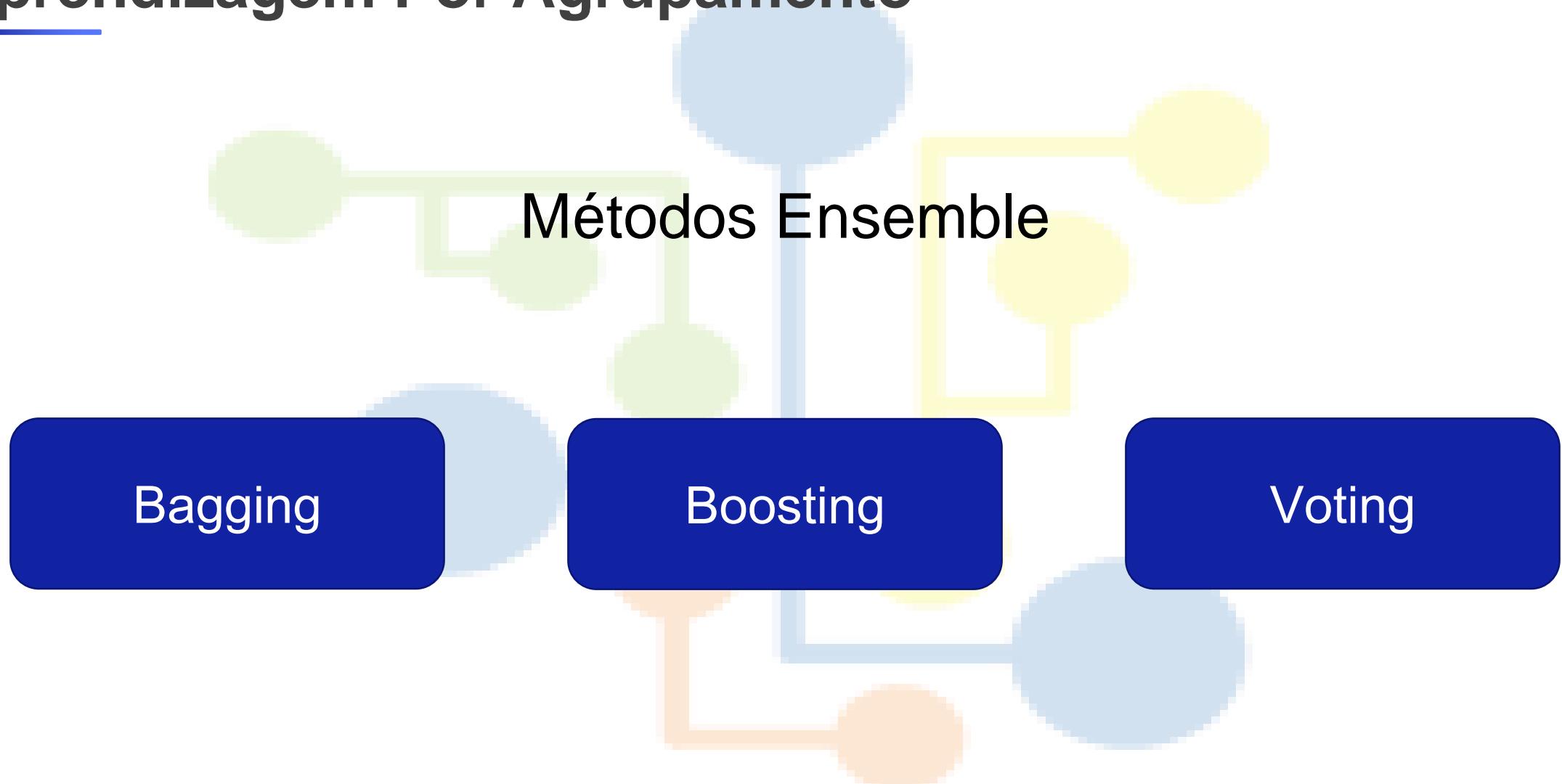
Random Forest

Bagging

AdaBoost

Gradient Boosting

Aprendizagem Por Agrupamento



Aprendizagem Por Agrupamento

Estado da Arte em Machine Learning



Aprendizagem Por Agrupamento



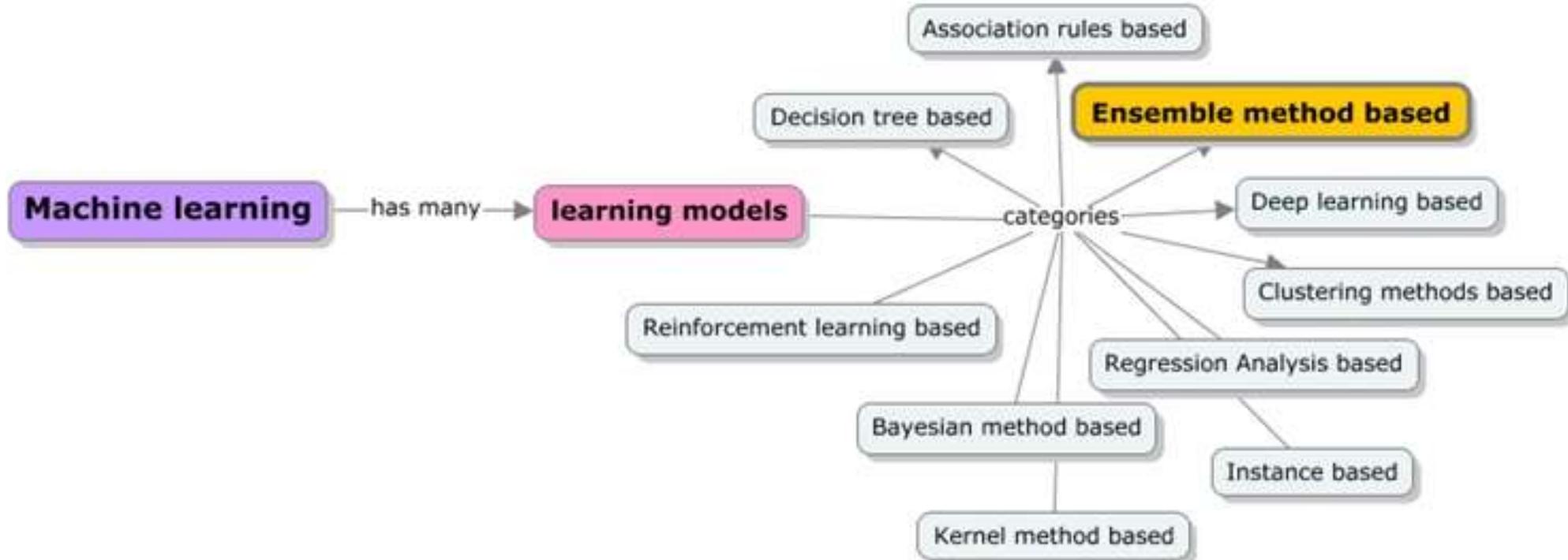
Estado da Arte em Machine Learning



Acurácia e Simplicidade

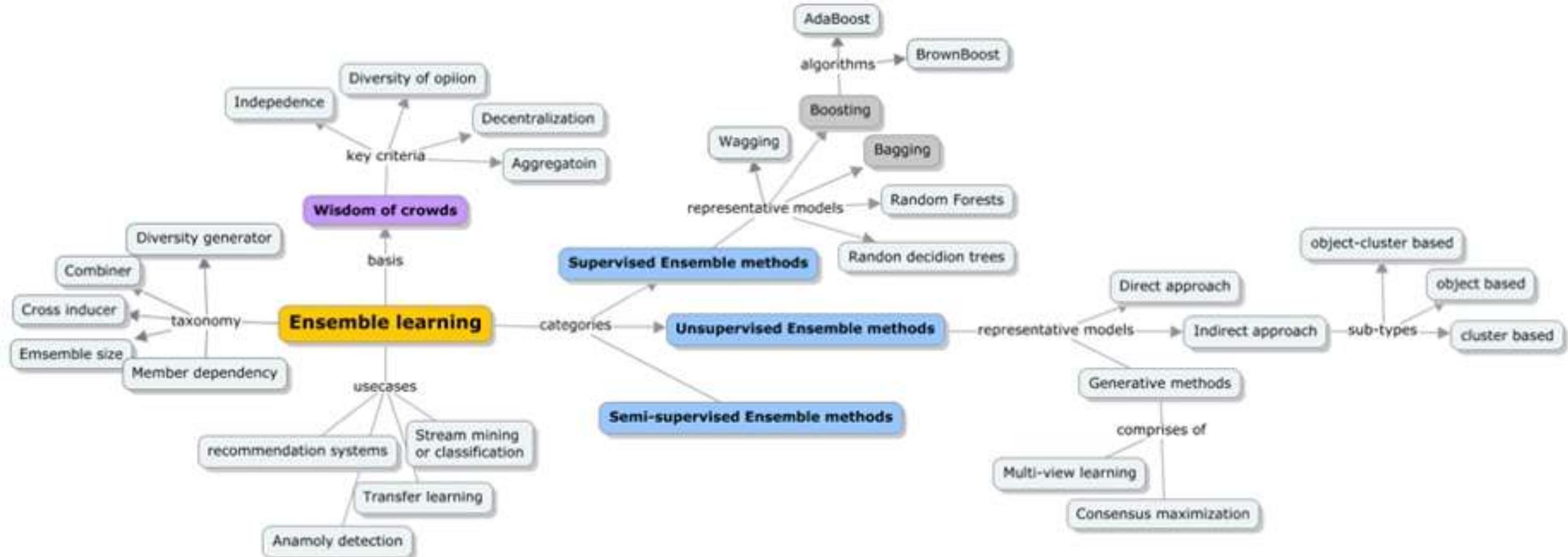
Aprendizagem Por Agrupamento

Métodos Ensemble são uma categoria de Algoritmos de Machine Learning

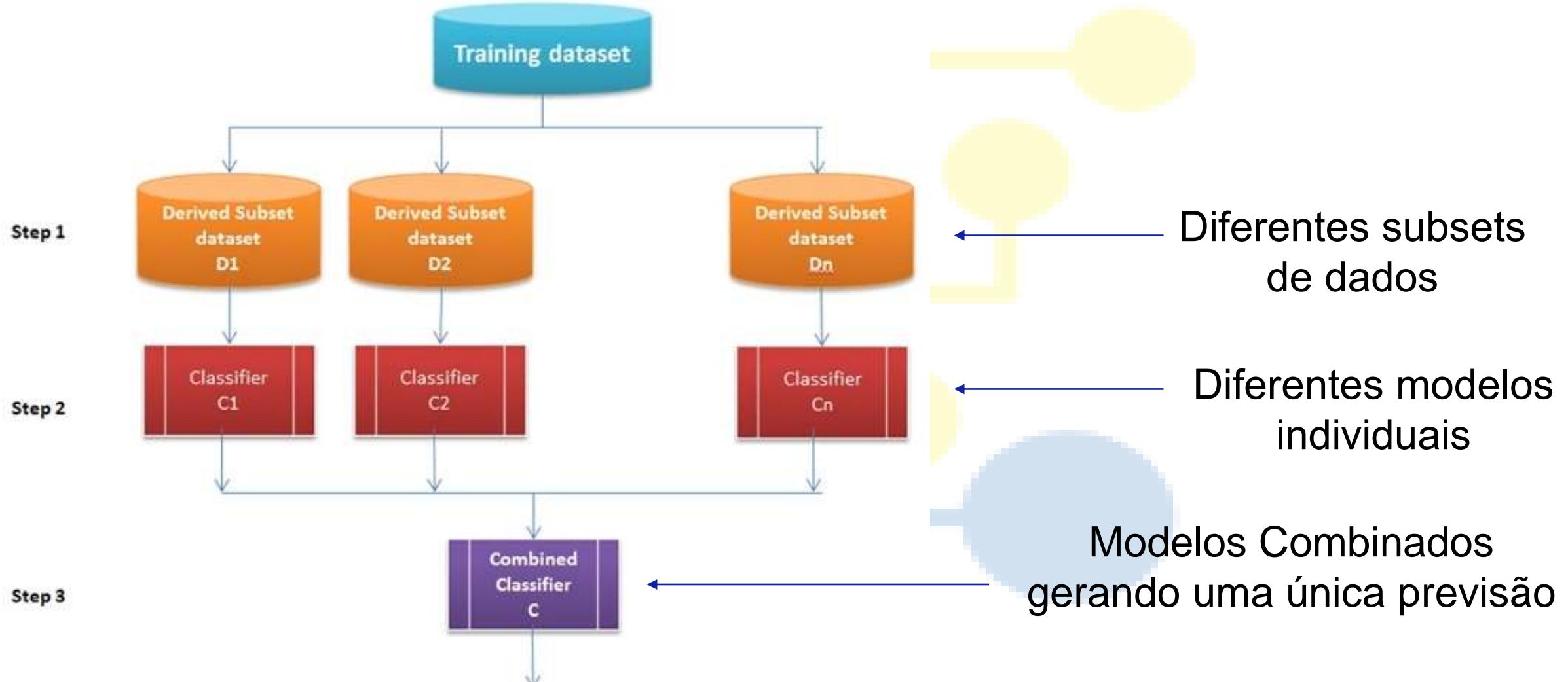


Aprendizagem Por Agrupamento

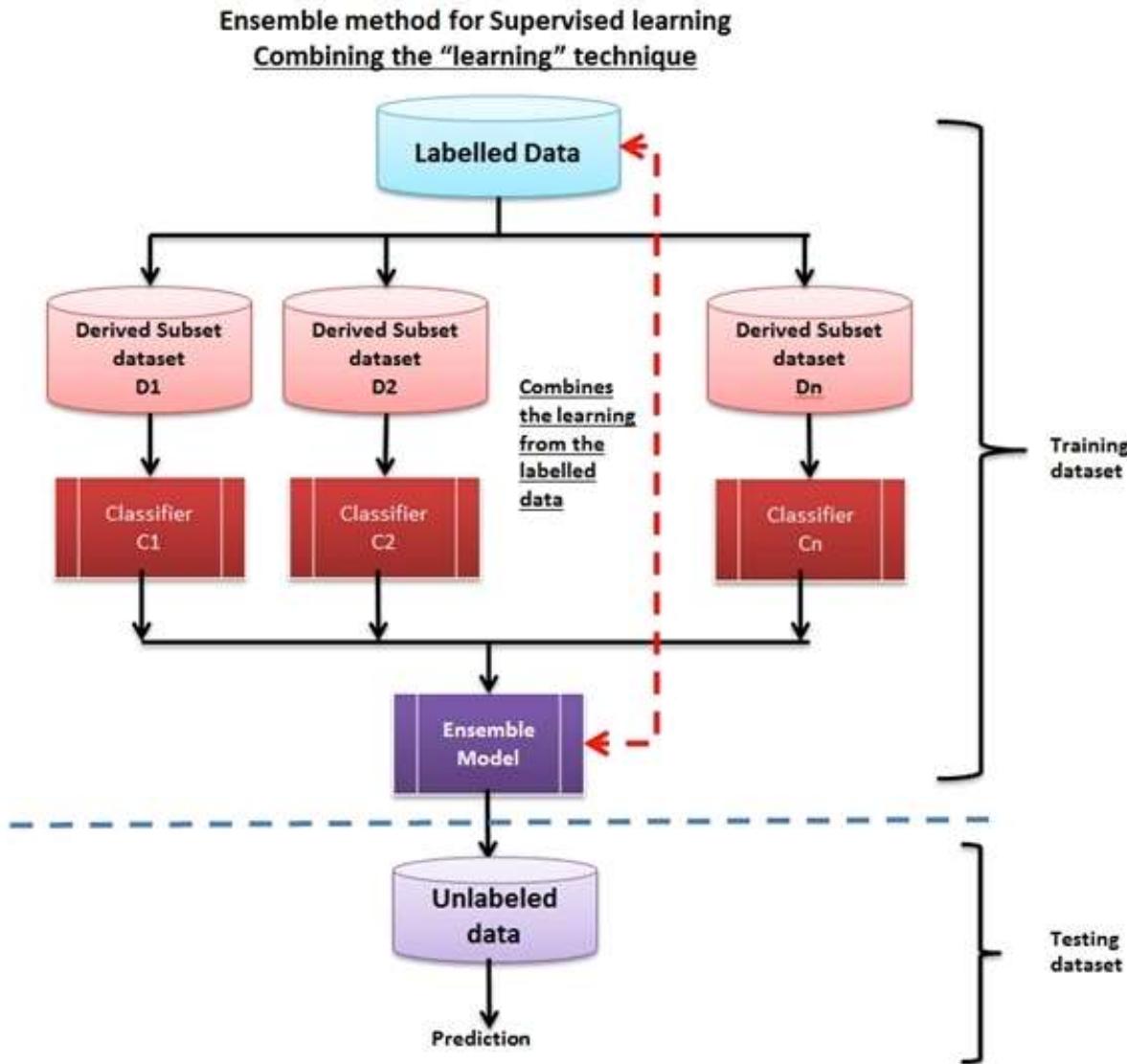
Ensemble Learning



Aprendizagem Por Agrupamento



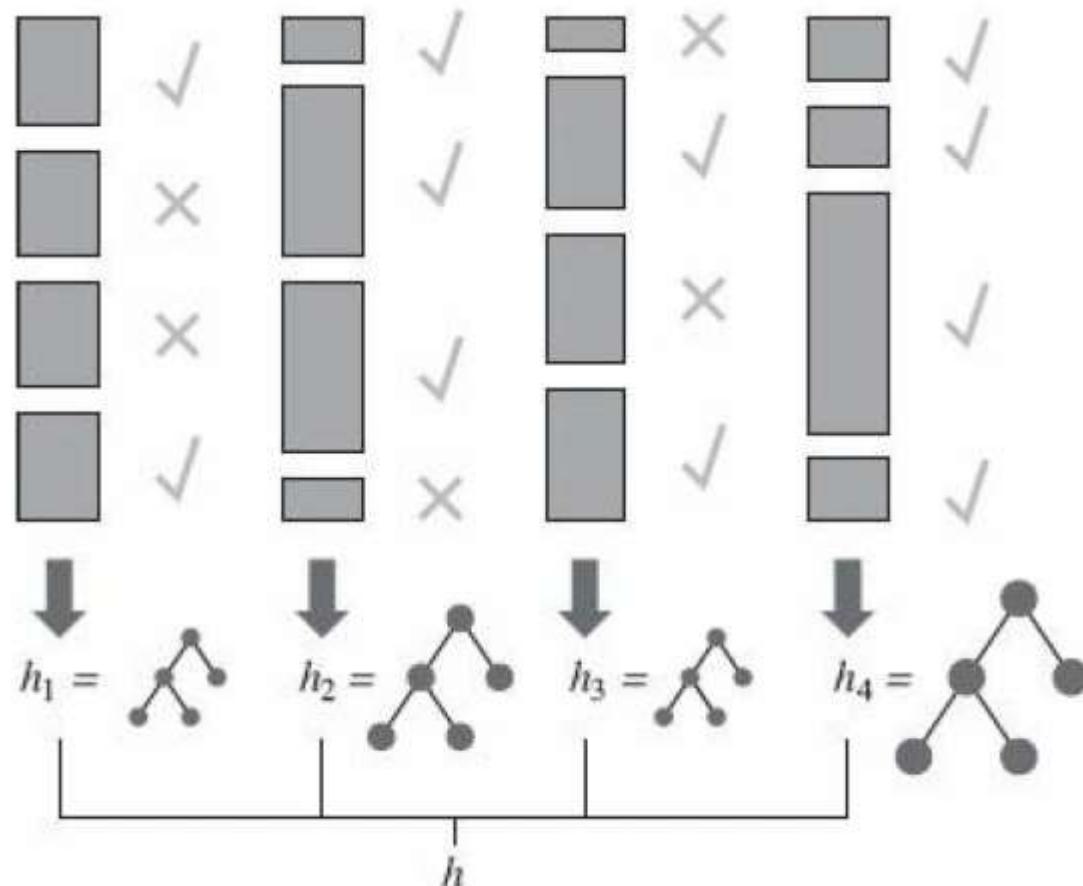
Aprendizaje en Por Agrupamento



Aqui ocorre a criação de diferentes modelos

Aqui ocorre a validação do modelo final

Aprendizagem Por Agrupamento



AdaBoost

$$H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right)$$

Aprendizagem Por Agrupamento

função ADABOOST(*exemplos, L, K*) **retorna** uma hipótese de maioria ponderada
entradas: *exemplos*, conjunto de N exemplos identificados $(x_1, y_1), \dots, (x_N, y_N)$

L, um algoritmo de aprendizagem

K, o número de hipóteses no conjunto

variáveis locais: **w**, um vetor de N pesos de exemplo, inicialmente $1/N$

h, um vetor de K hipóteses

z, um vetor de K pesos de hipóteses

para $k = 1$ **até** K **faça**

h[k] $\leftarrow L(exemplos, w)$

erro $\leftarrow 0$

para $j = 1$ **até** N **faça**

se **h**[k](x_j) $\neq y_j$ **então** *erro* $\leftarrow erro + w[j]$

para $j = 1$ **até** N **faça**

se **h**[k](x_j) $= y_j$ **então** **w**[j] $\leftarrow w[j] \cdot erro / (1 - erro)$

w \leftarrow NORMALIZAR(**w**)

z[k] $\leftarrow \log(1 - erro) / erro$

retornar MAIORIA-APONDERADA(**h, z**)



Obrigado