



Formação Inteligência Artificial



Introdução à Inteligência Artificial





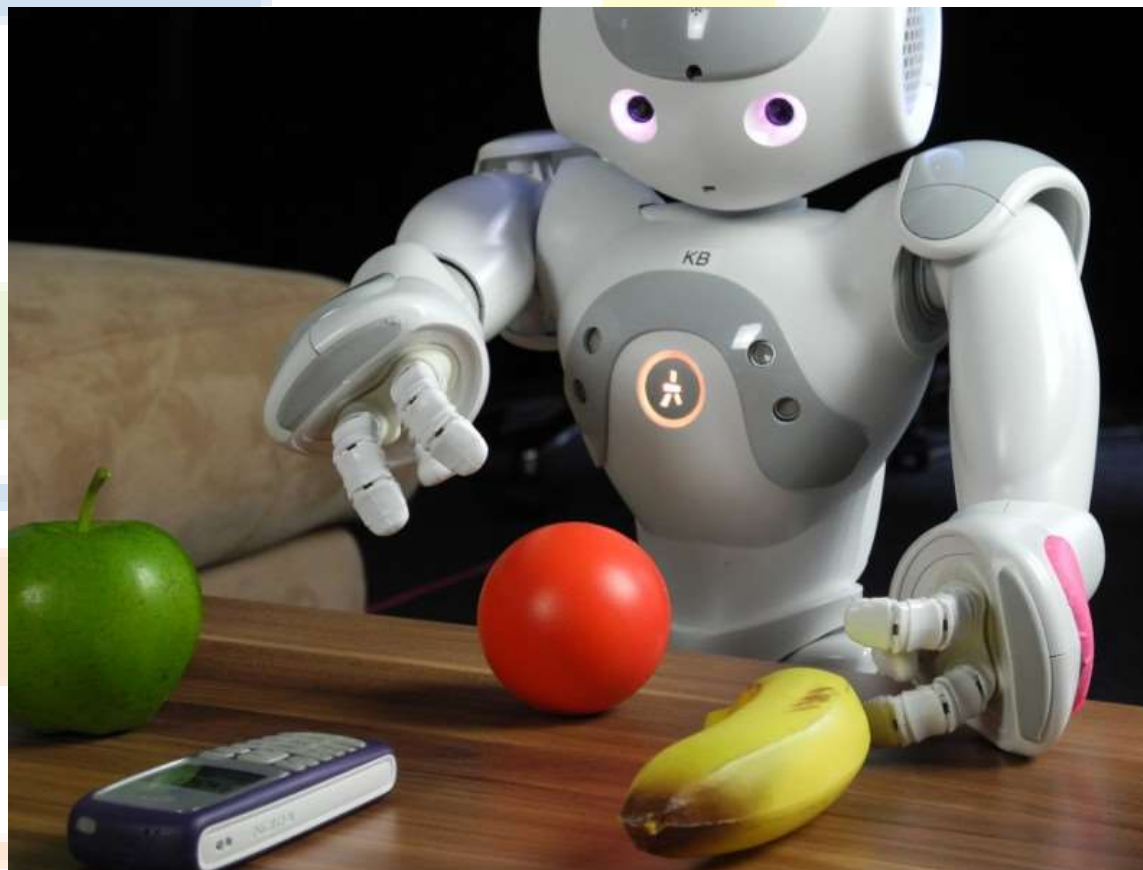
Aprendizagem





Introdução

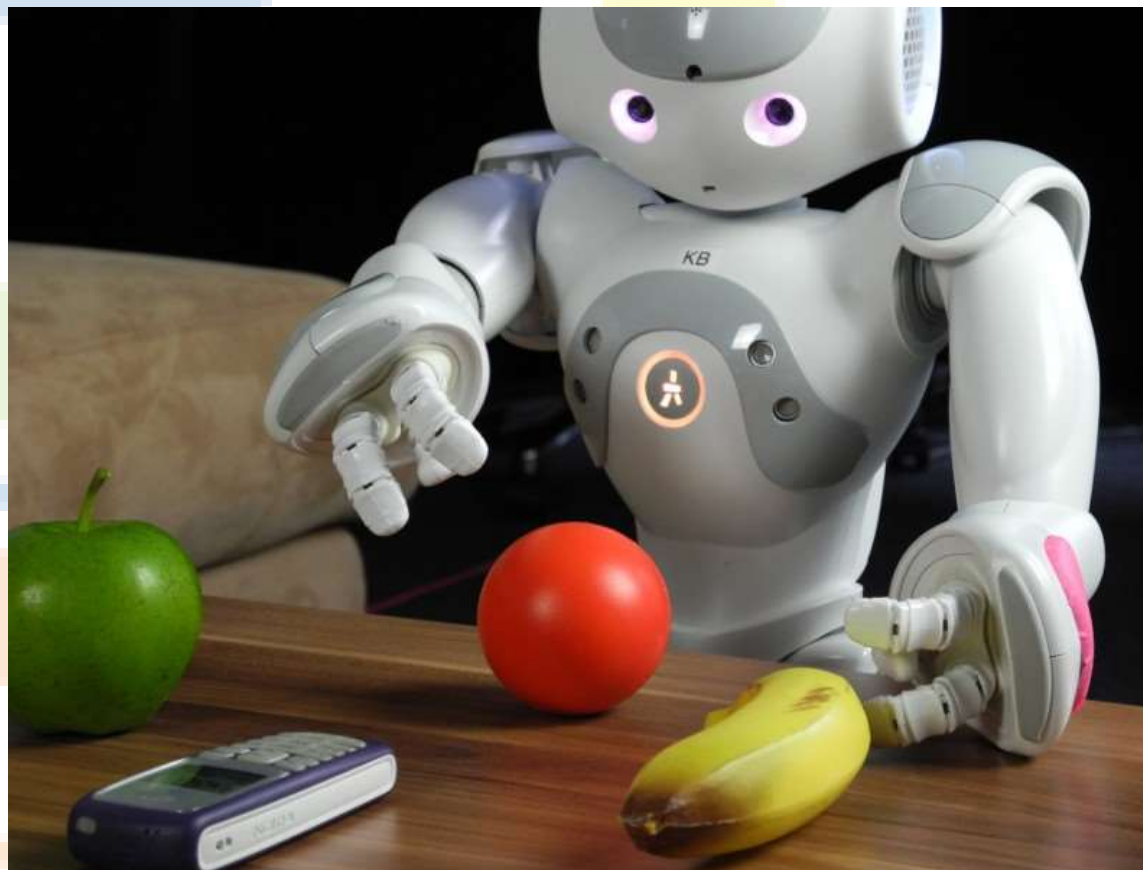
A partir de uma coleção de pares de entrada e saída, aprender uma função que prevê a saída para novas entradas.





Introdução

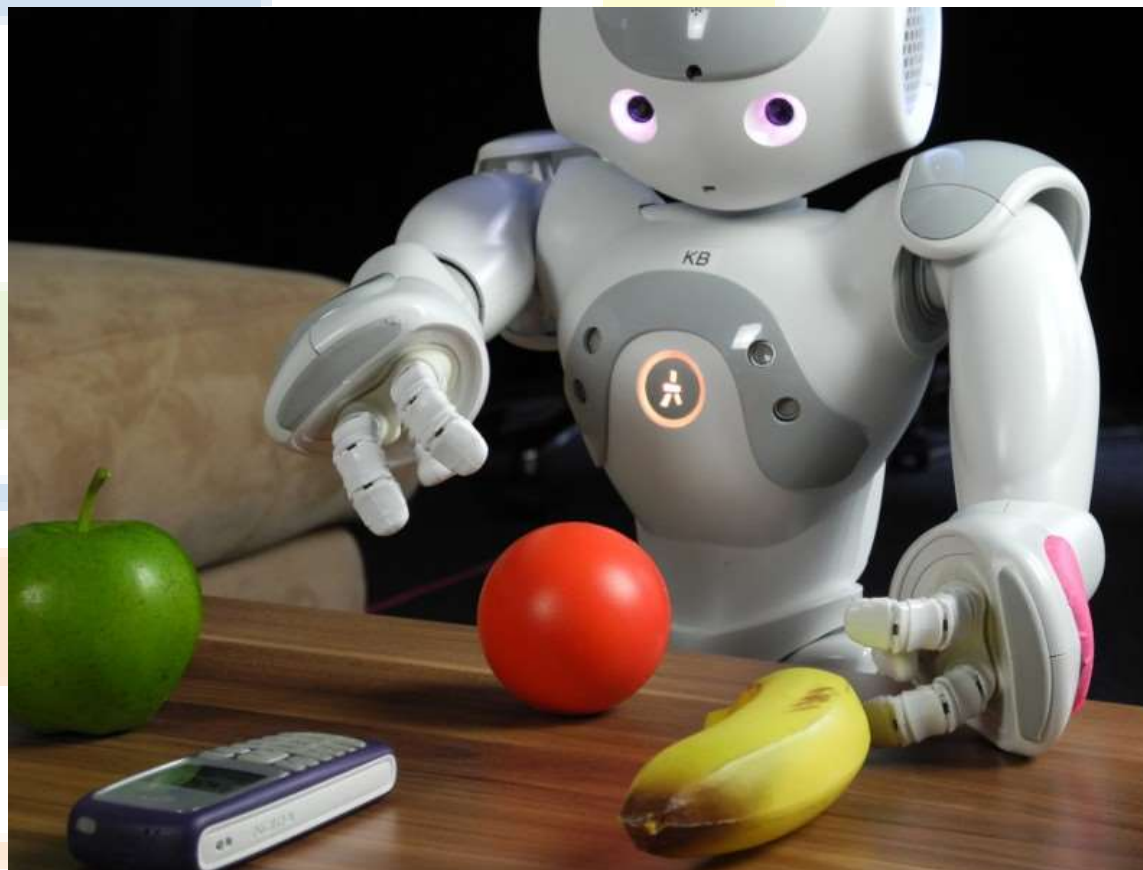
Por que queremos que um agente aprenda?





Introdução

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







Introdução

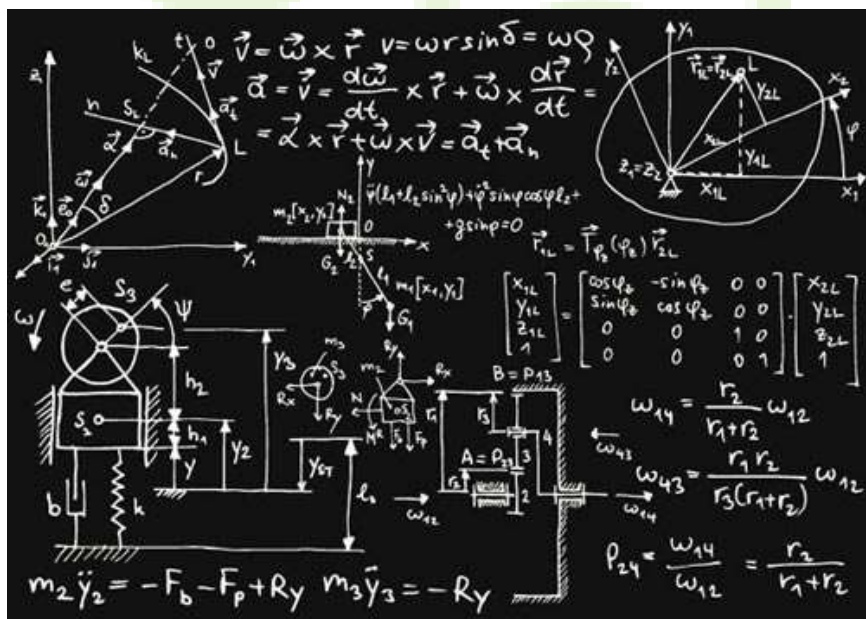


Processo de Aprendizagem





Introdução

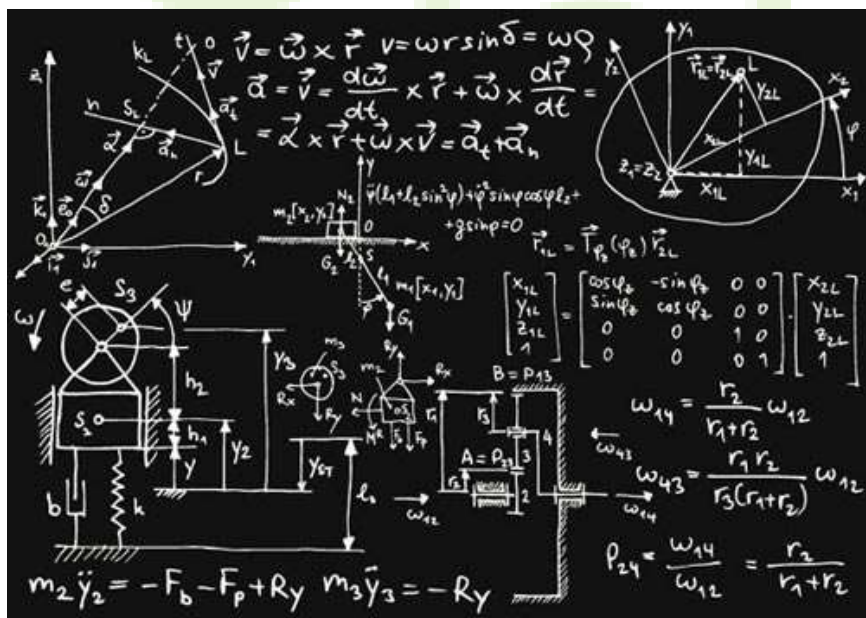


Um algoritmo constrói suas capacidades cognitivas através da criação de uma formulação matemática que inclui todas as características dadas sobre um determinado fenômeno.





Introdução

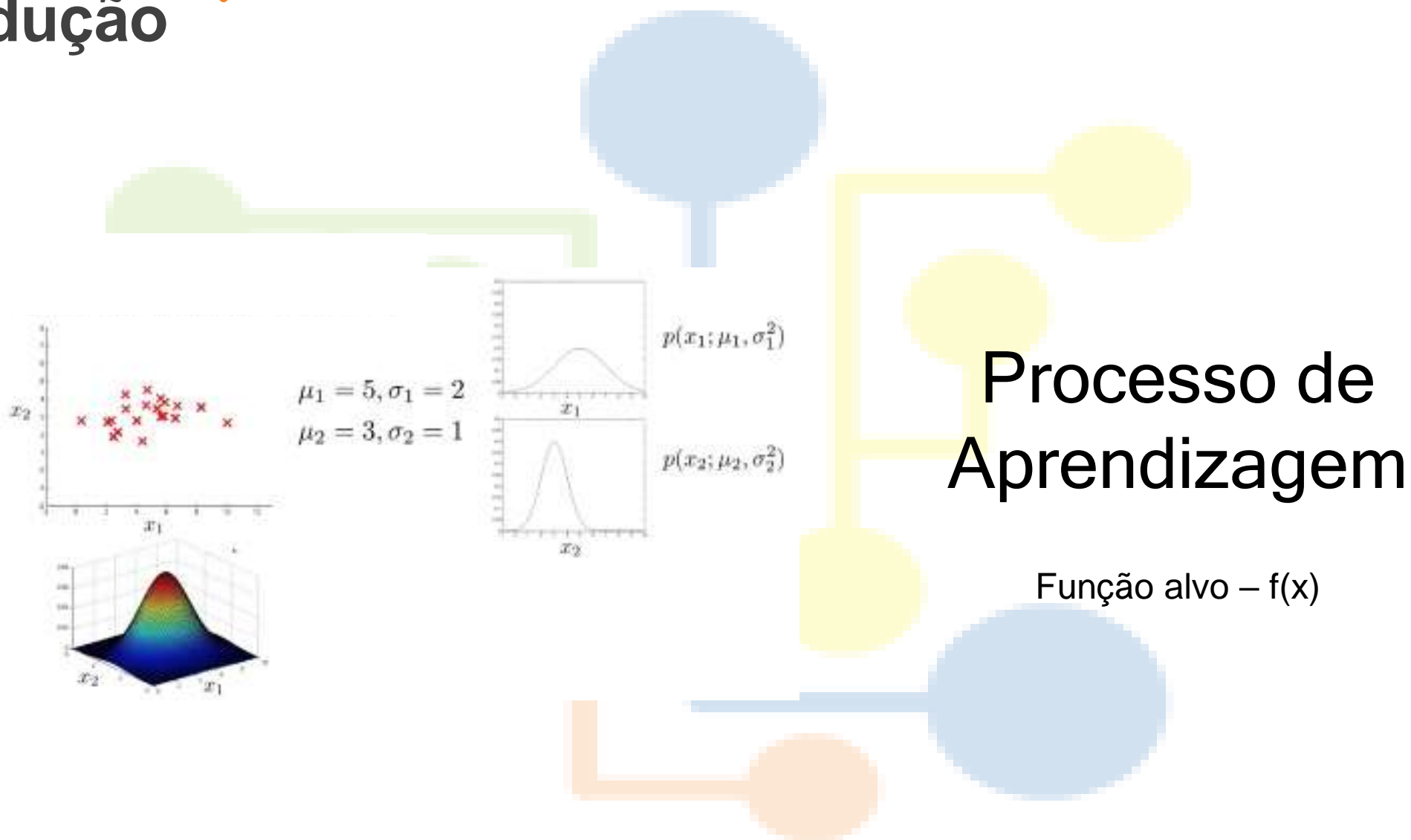


Função alvo – $f(x)$





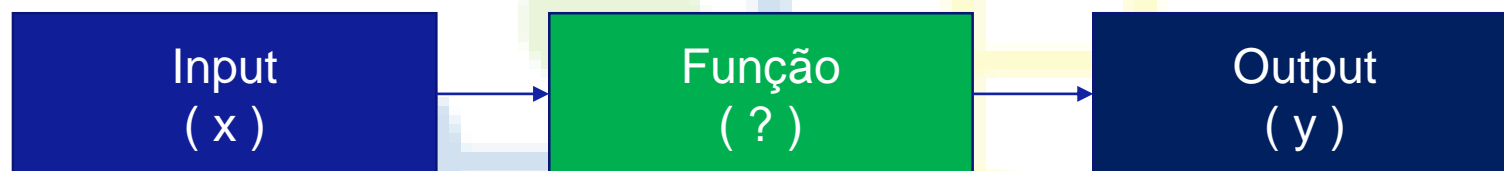
Introdução





Introdução

Processo de Aprendizagem



E como um algoritmo encontra a função matemática que descreve este relacionamento?





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 Apreendidos

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.





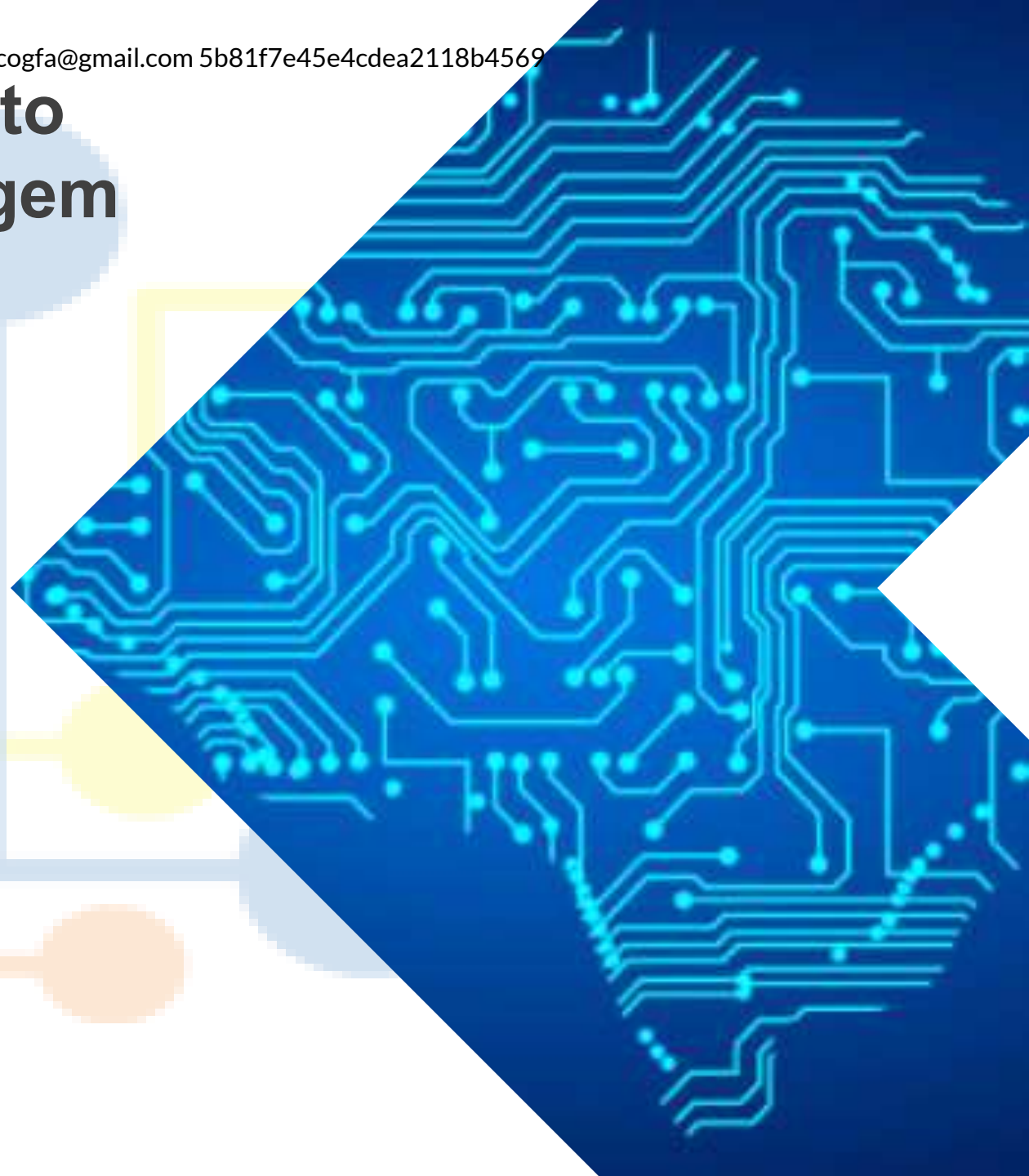
Representação, Conhecimento Prévio e Tipos de Aprendizagem





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





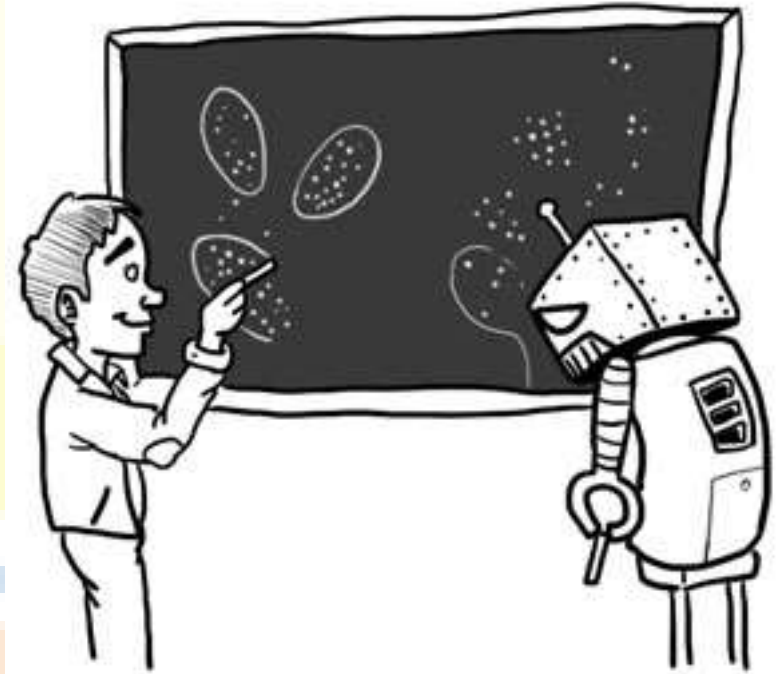
Representação, Conhecimento Prévio e Tipos de Aprendizagem





Representação, Conhecimento Prévio e Tipos de Aprendizagem

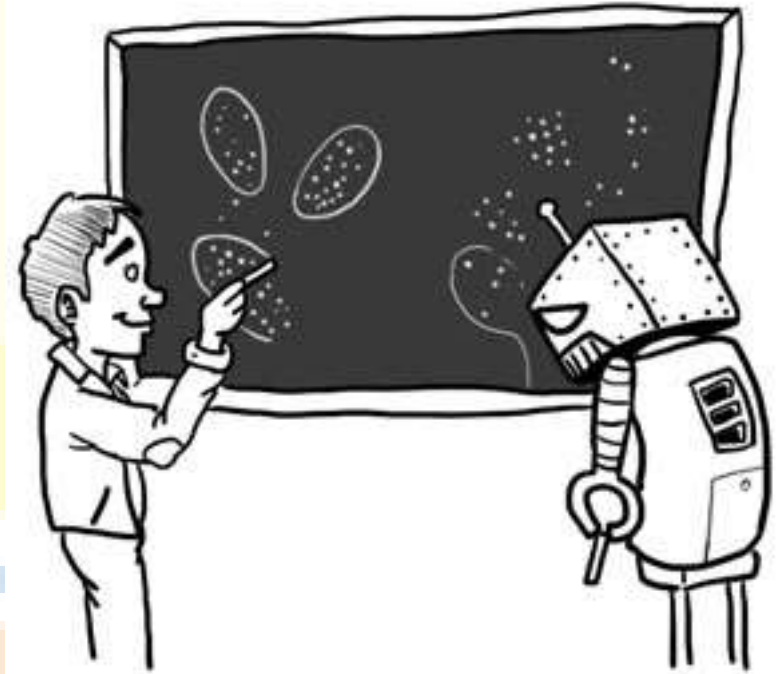
No **Aprendizado por Exemplos**, o aprendiz induz a descrição de um conceito formulando uma regra geral a partir dos exemplos e dos contra-exemplos fornecidos pelo professor ou pelo ambiente.





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

Aprendizagem Por
Reforço

reforço

AGENTE

estado

AMBIENTE

ação





Representação, Conhecimento Prévio e Tipos de Aprendizagem

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





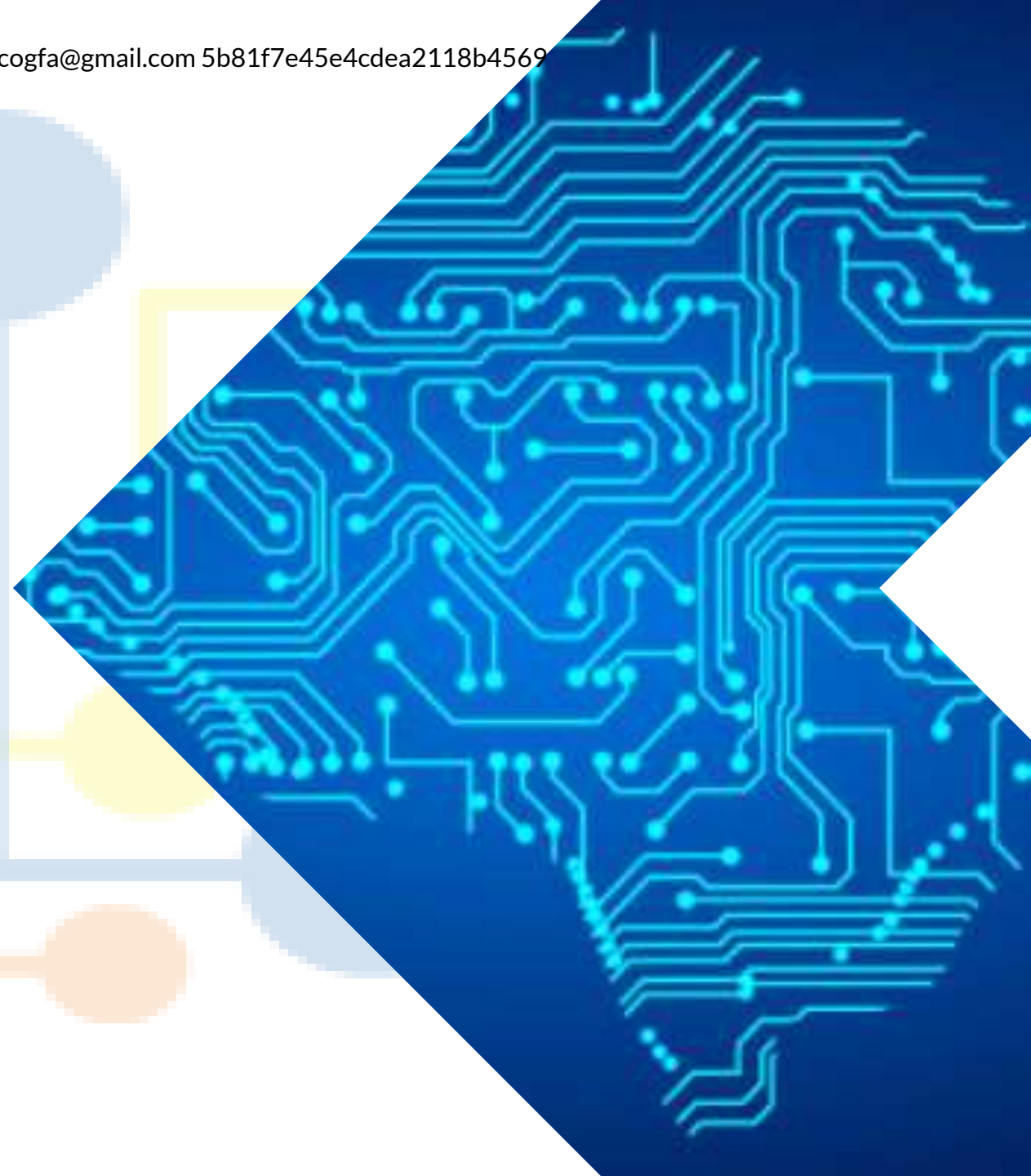
Teoria da Aprendizagem





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:

$$y = f(x),$$

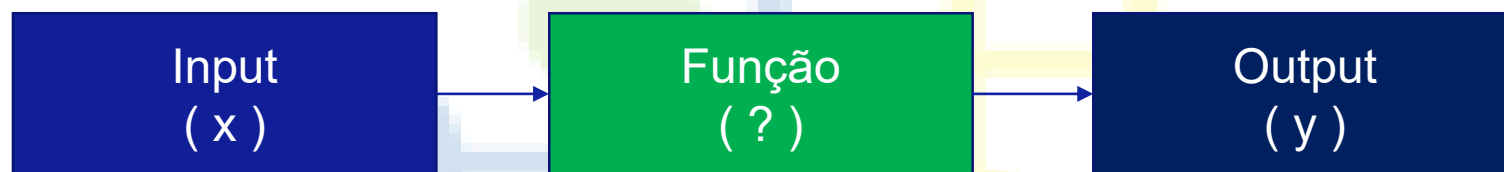
o objetivo da aprendizagem é descobrir uma função h (hipótese) que se aproxime da função verdadeira f .





Teoria da Aprendizagem

Processo de Aprendizagem



$$y = f(x)$$

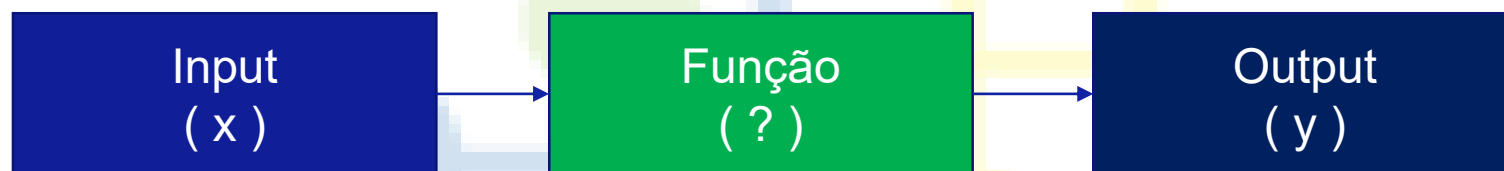
A função h é uma hipótese. Aprendizagem é uma busca através do espaço de hipóteses possíveis, por aquela que terá um bom desempenho, mesmo em novos exemplos além do conjunto de treinamento.





Teoria da Aprendizagem

Processo de Aprendizagem



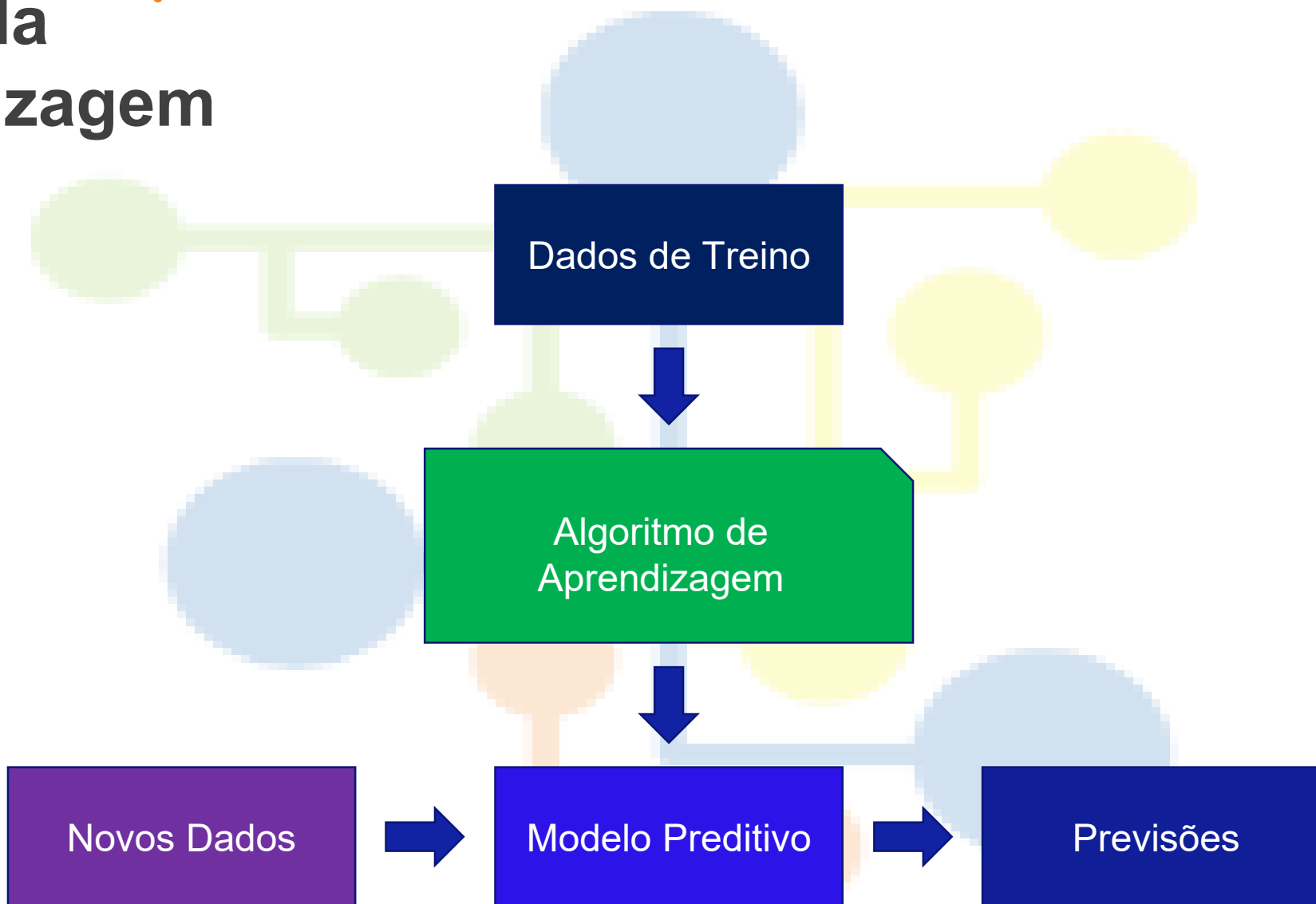
$$y = f(x)$$

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, $P(Y|x)$.





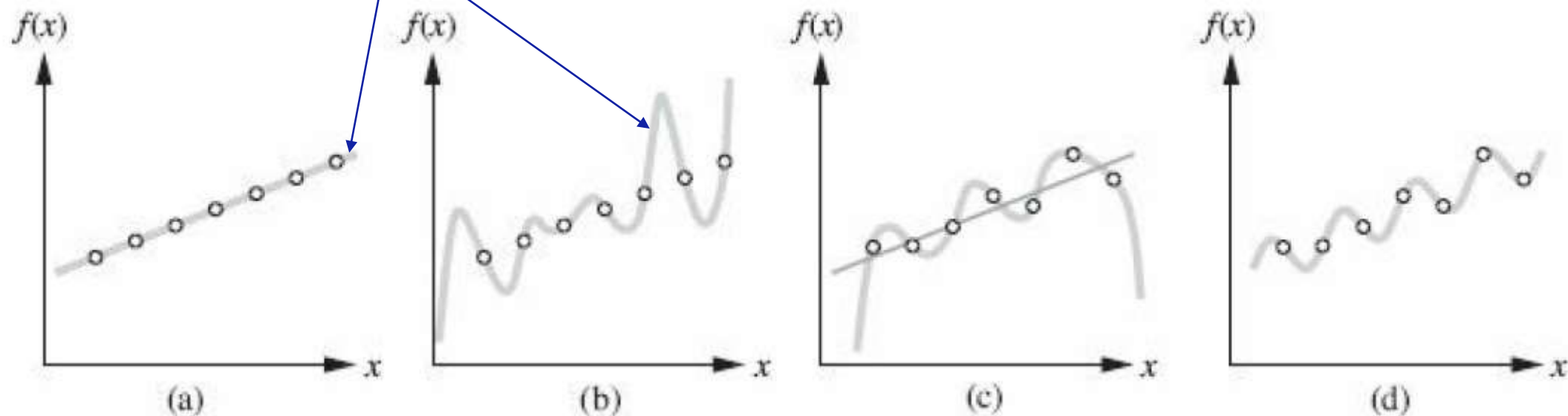
Teoria da Aprendizagem





Teoria da Aprendizagem

h (hipóteses)

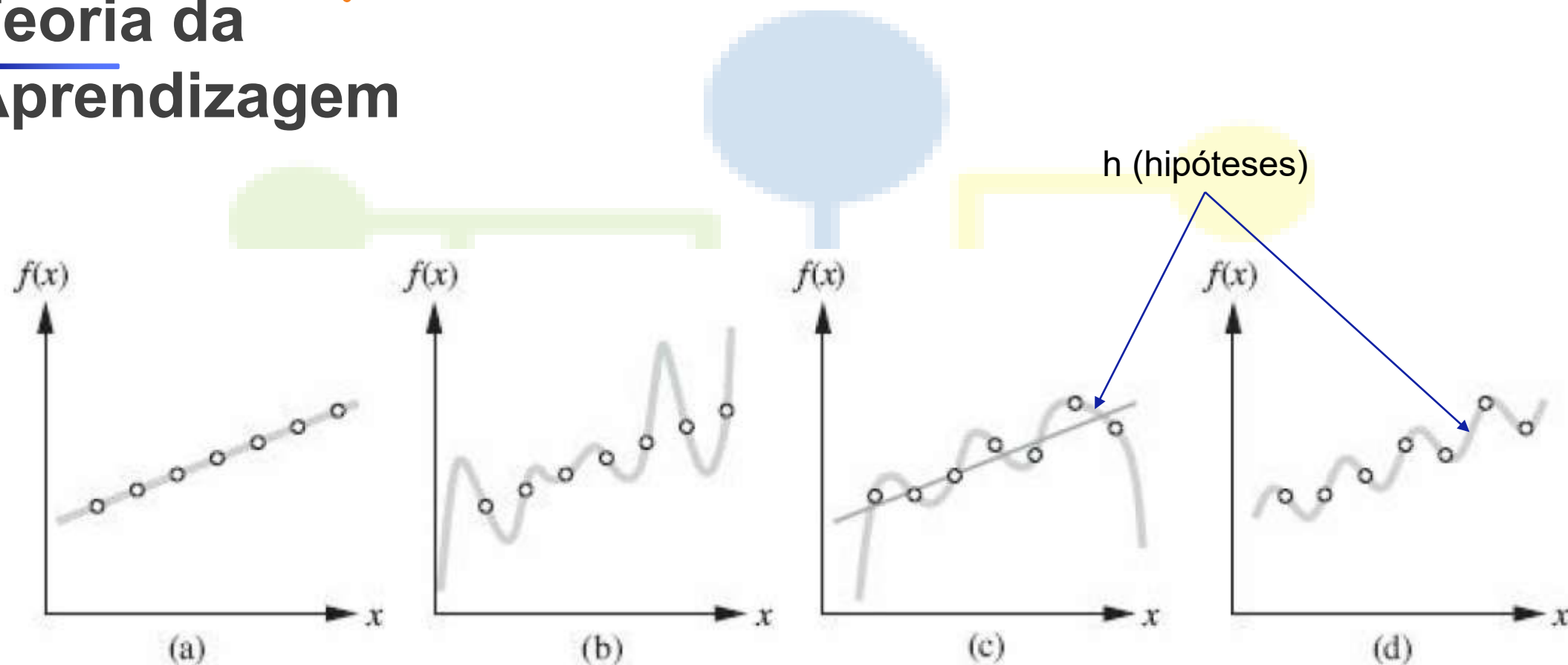


Os exemplos são pontos no plano (x, y) , onde $y = f(x)$.





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)

X

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

y

Recebeu crédito	Sim
-----------------	-----





Elementos do Processo de Aprendizagem

Input

x

{Dados do cliente}

Output

y

{Decisão \rightarrow 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

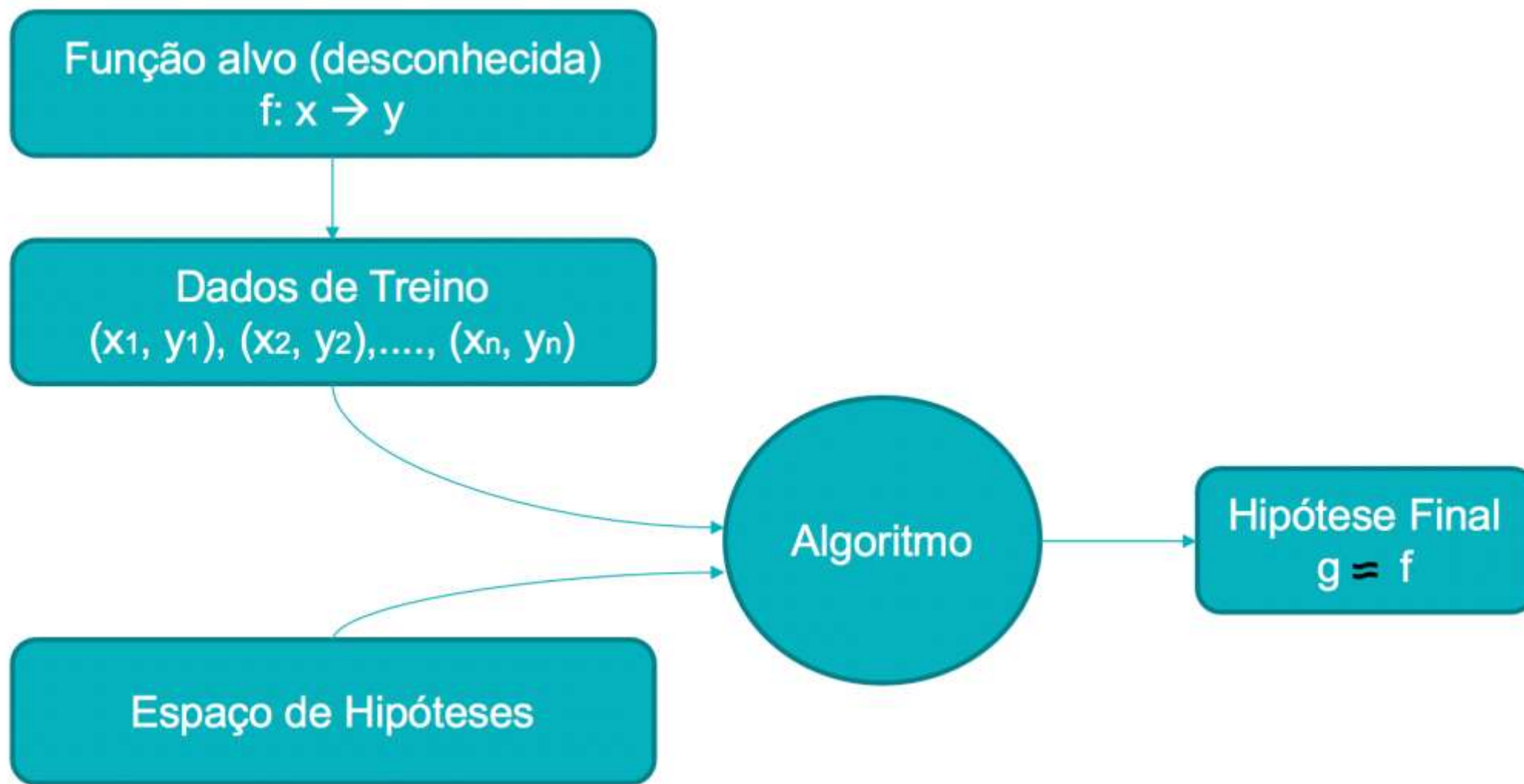
$g: x \rightarrow y$

{Fórmula a ser usada}





Elementos do Processo de Aprendizagem



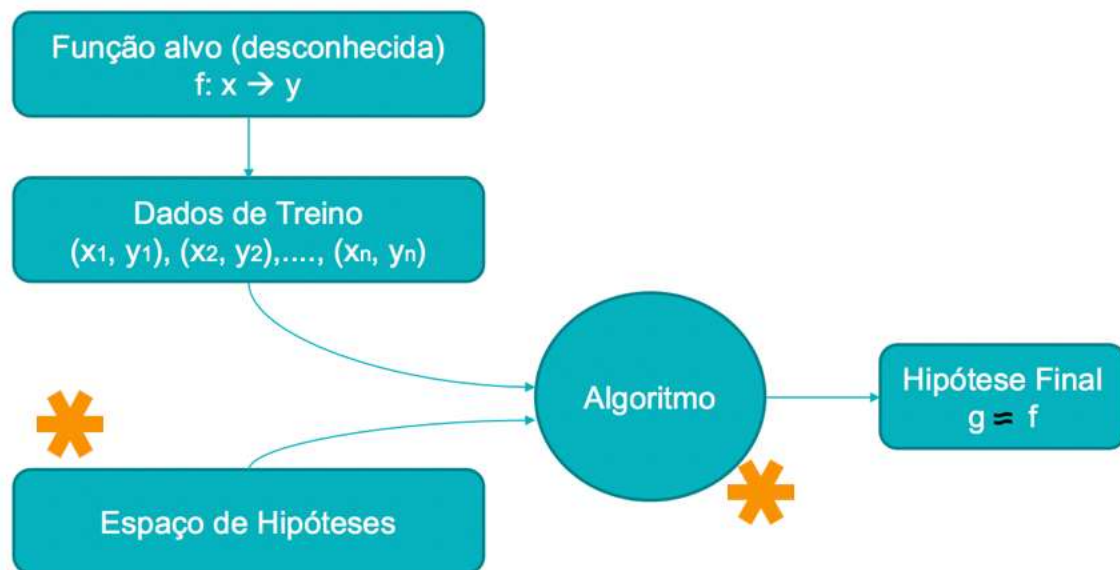


Modelo de Aprendizagem e Espaço de Hipóteses





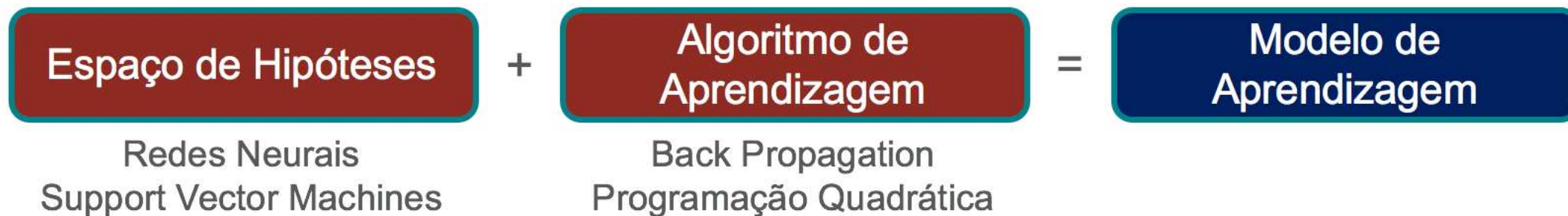
Modelo de Aprendizagem e Espaço de Hipóteses



- Espaço de Hipóteses

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

- Algoritmo de Aprendizagem





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

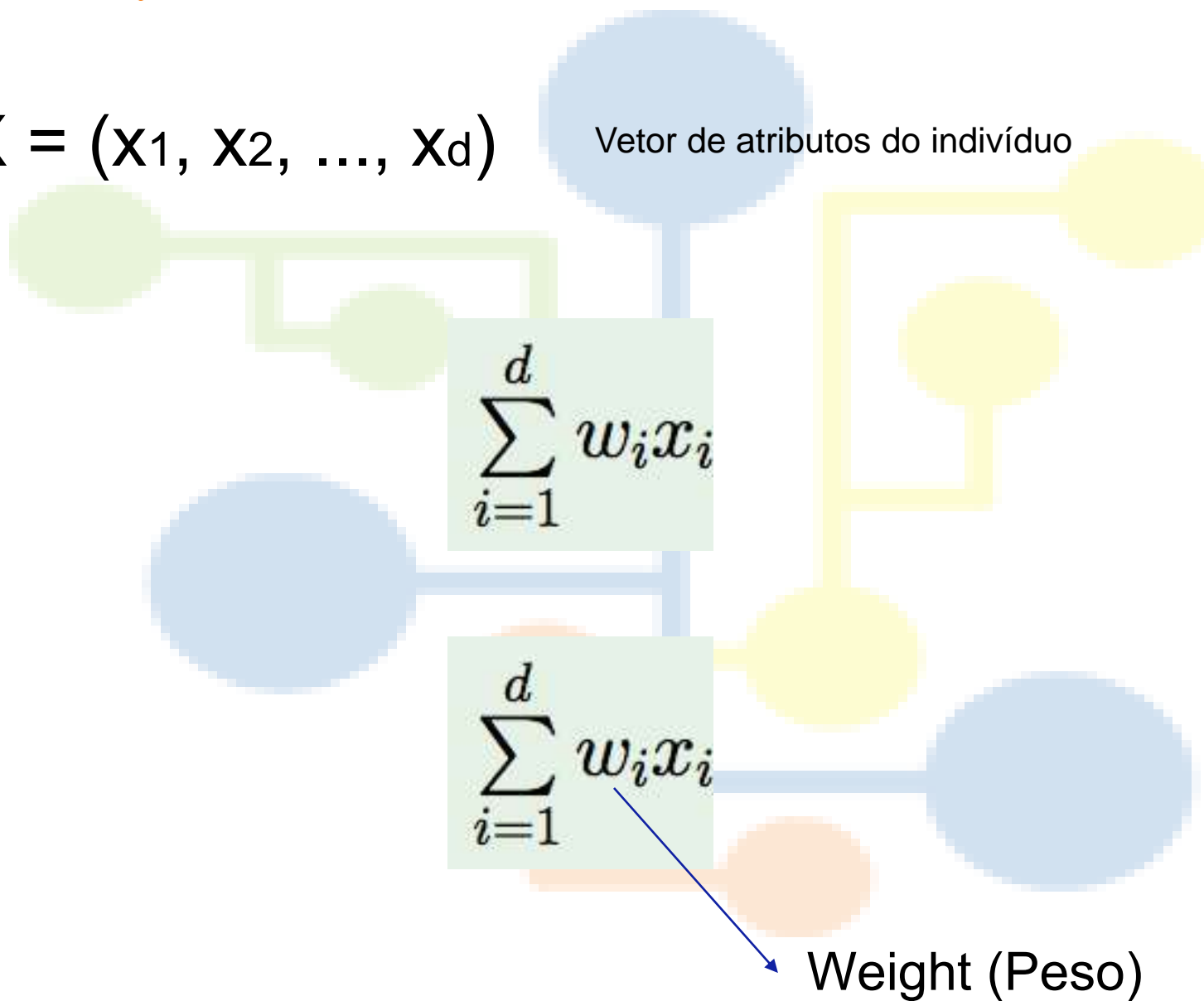
$$h(\mathbf{x}) = \text{sign} \left(\left(\sum_{i=1}^d w_i x_i \right) - \text{threshold} \right)$$

As diferentes combinações weight/threshold vão formar diferentes hipóteses





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



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



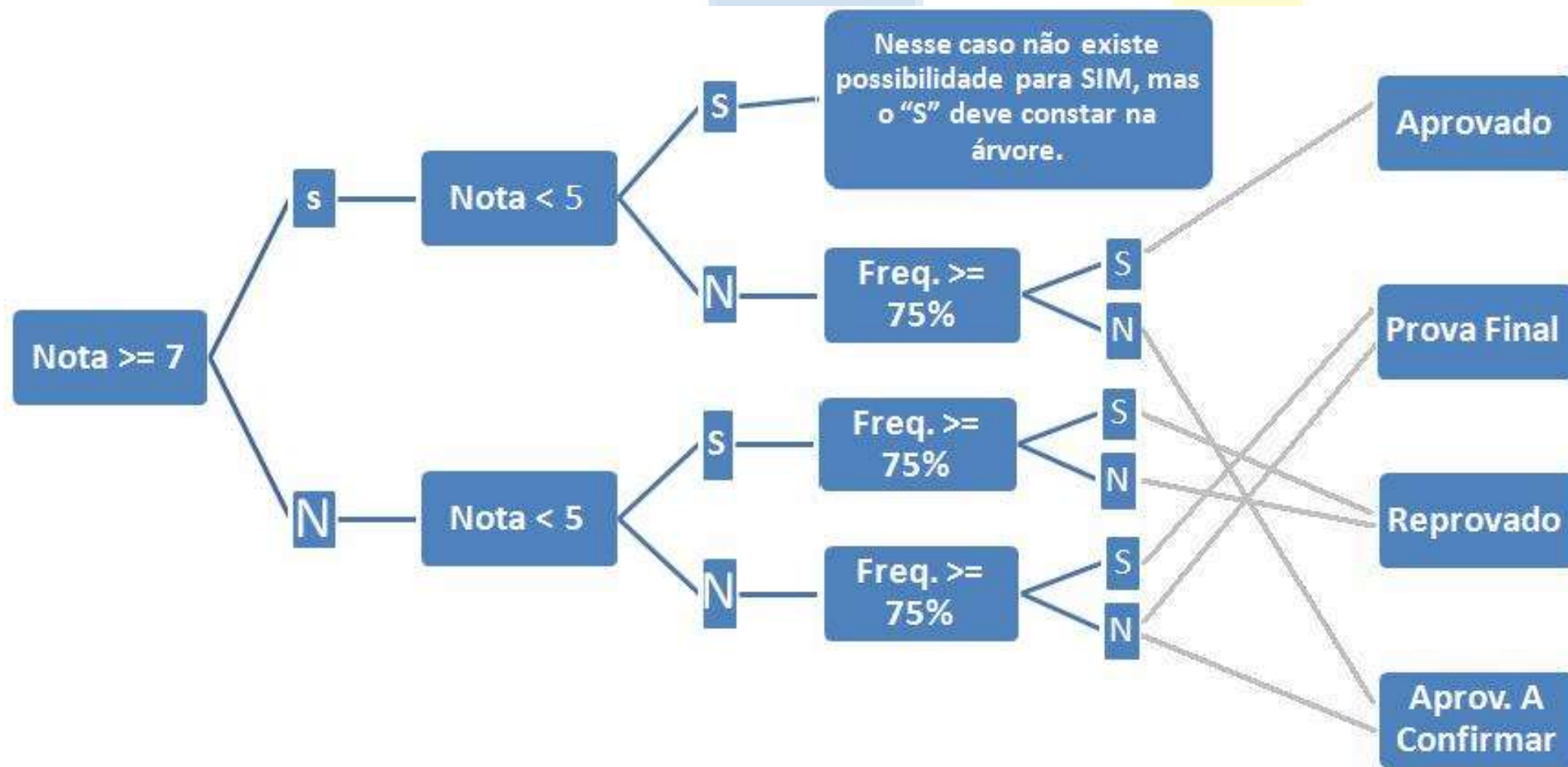


Aprendizagem em Árvores de Decisão





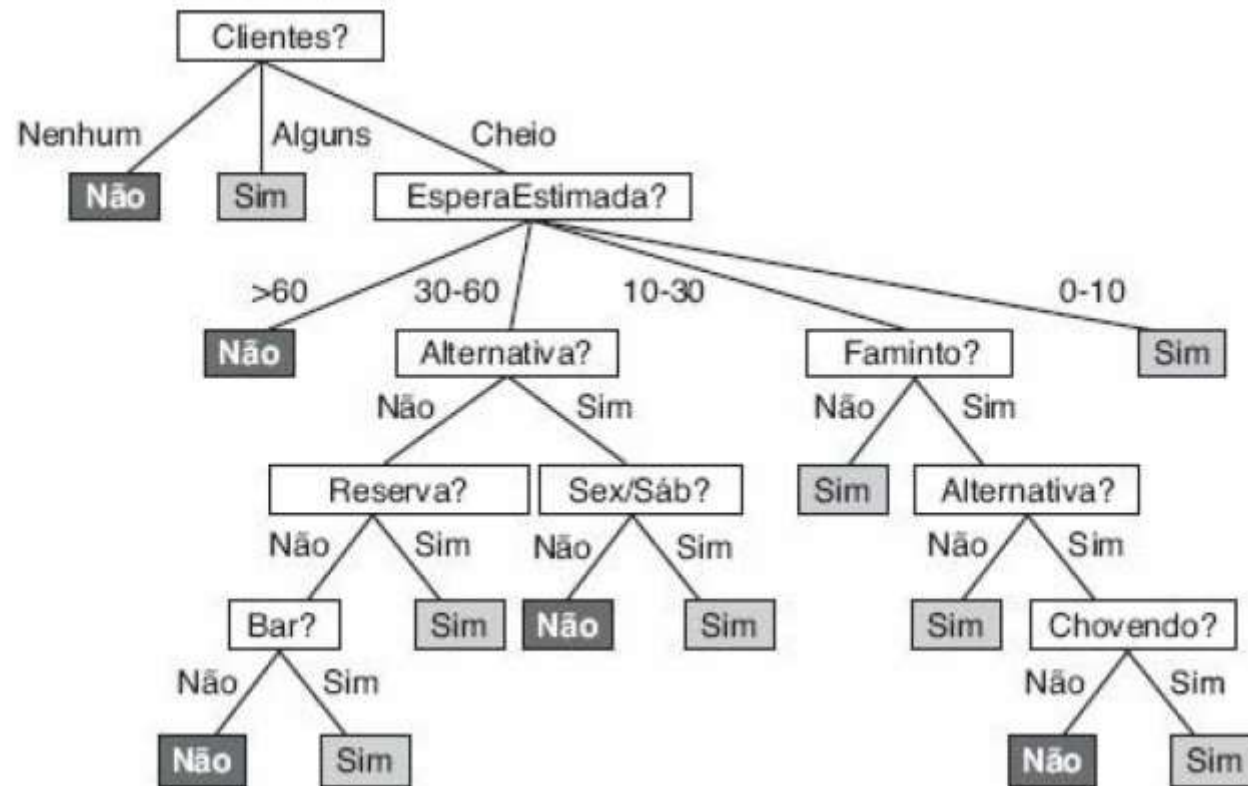
Aprendizagem em Árvores de Decisão





Aprendizagem em Árvore de Decisão

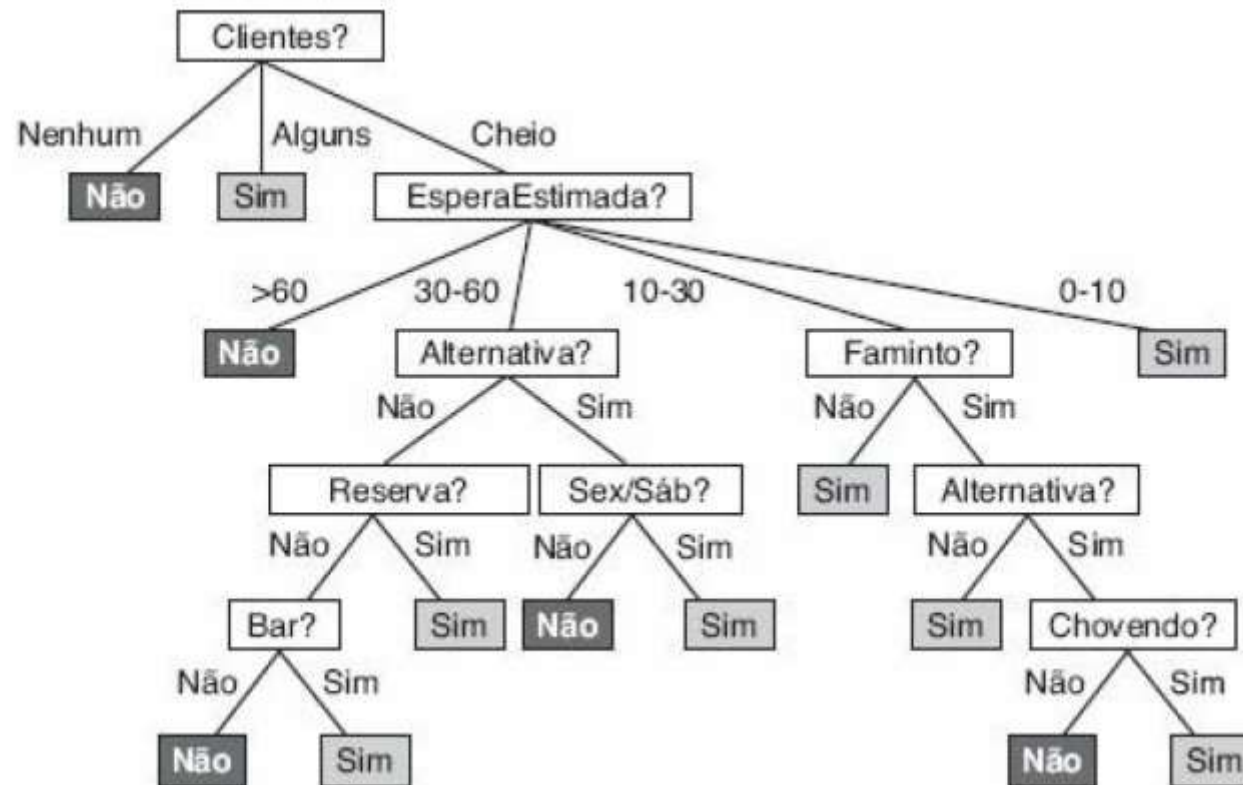
Uma árvore de decisão representa uma função que toma como entrada um vetor de valores de atributos e retorna uma “decisão” — um valor de saída, único.





Aprendizagem em Árvores de Decisão

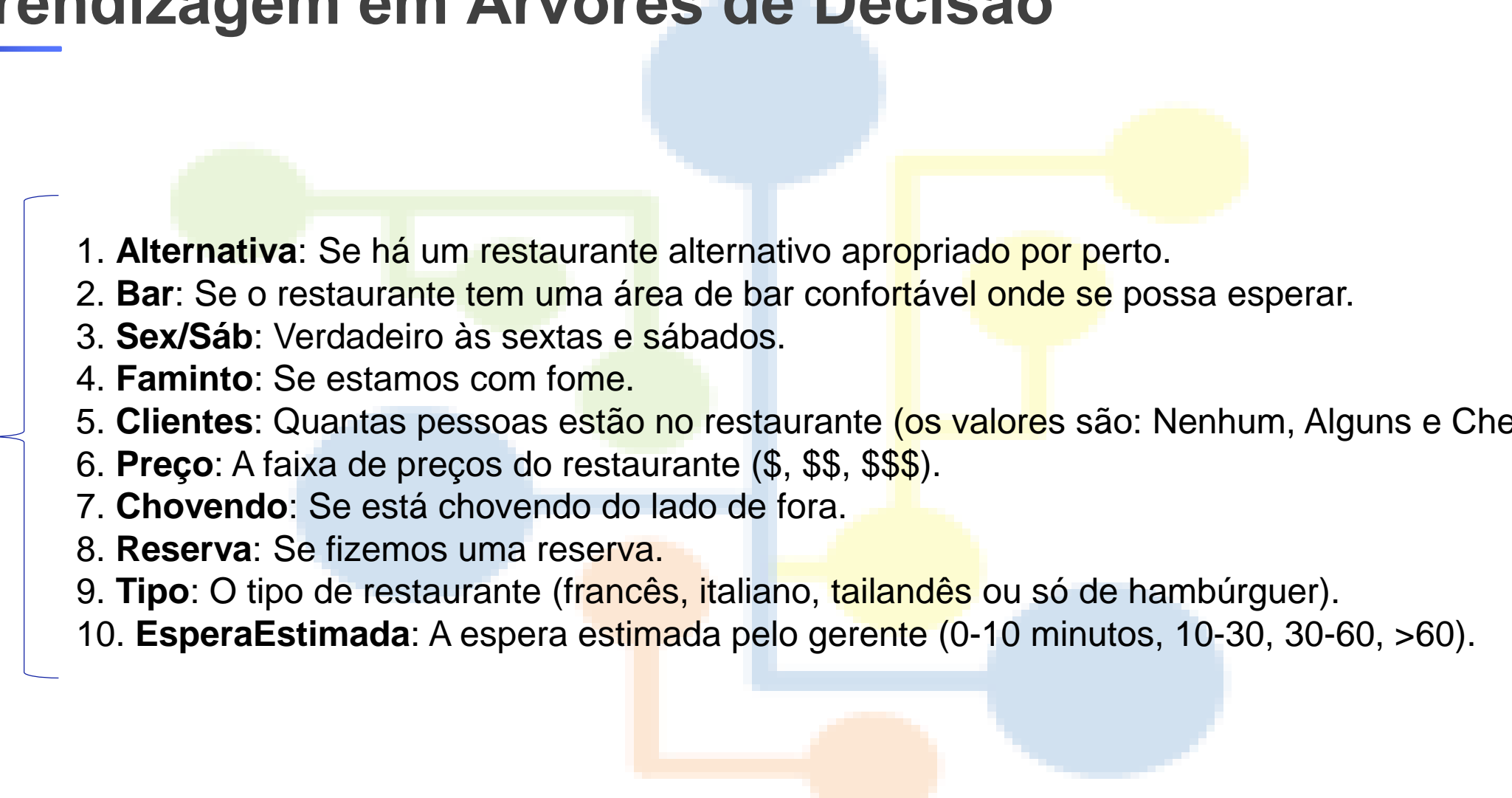
Uma árvore de decisão alcança sua decisão executando uma sequência de testes.





Aprendizagem em Árvores de Decisão

X

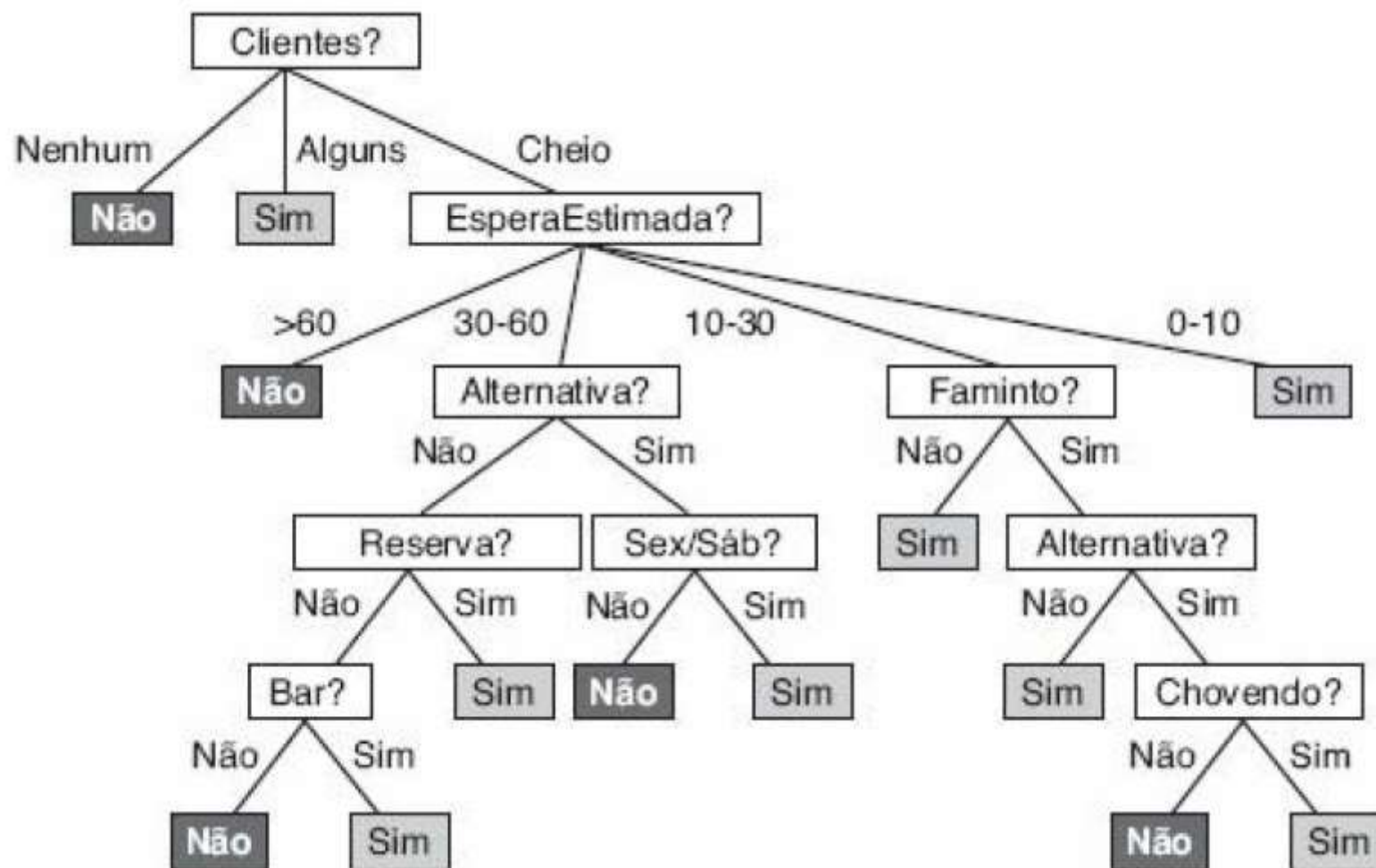
- 
1. **Alternativa:** Se há um restaurante alternativo apropriado por perto.
 2. **Bar:** Se o restaurante tem uma área de bar confortável onde se possa esperar.
 3. **Sex/Sáb:** Verdadeiro às sextas e sábados.
 4. **Faminto:** Se estamos com fome.
 5. **Cientes:** Quantas pessoas estão no restaurante (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 Árvore de Decisão

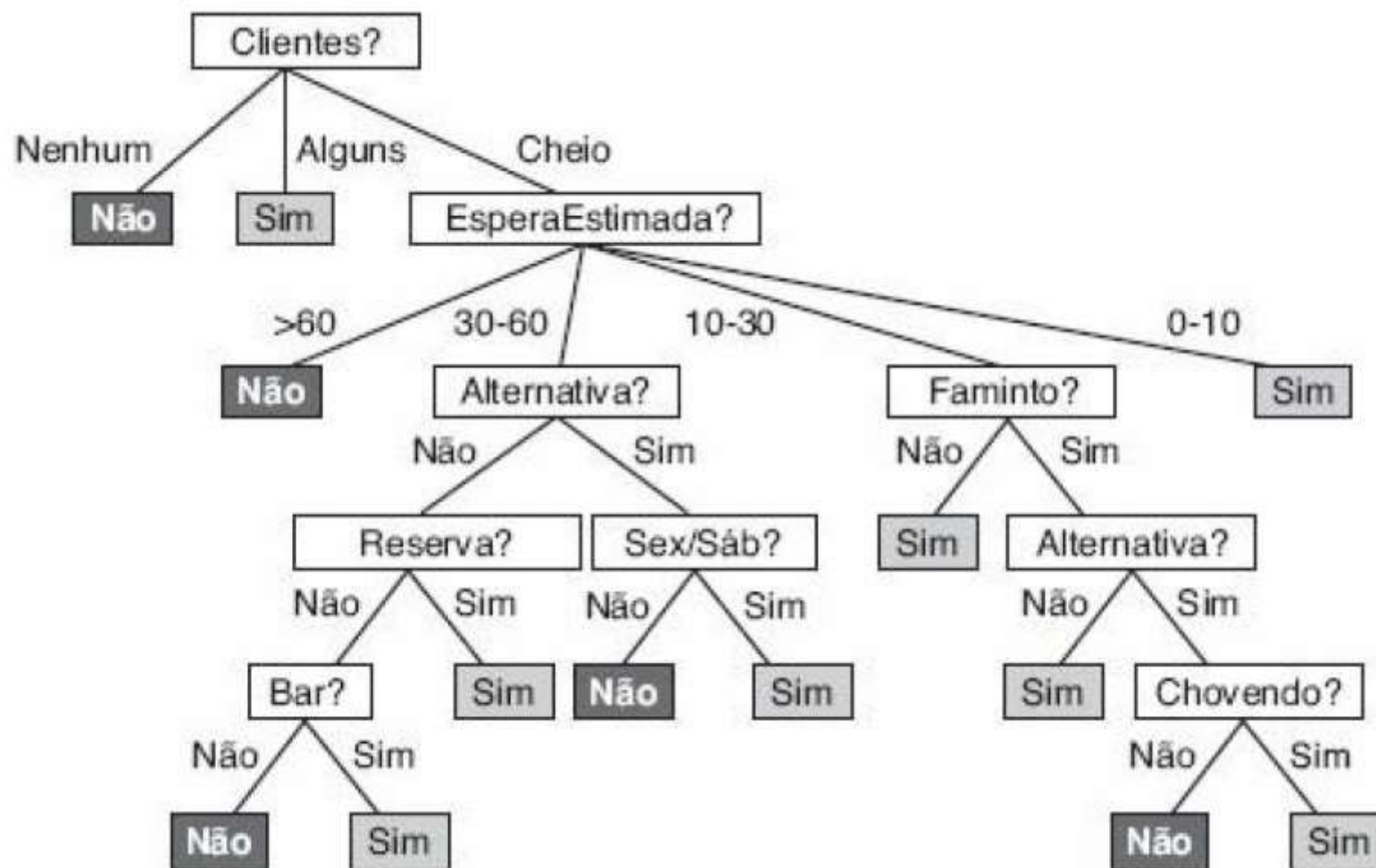
Exemplo	Atributos										Meta
	<i>Alt</i>	<i>Bar</i>	<i>Sex</i>	<i>Fam</i>	<i>Cli</i>	<i>Preço</i>	<i>Chuva</i>	<i>Res</i>	<i>Tipo</i>	<i>Estim</i>	<i>VaiEsperar</i>
x_1											$y_1 = \text{Sim}$
x_2	<i>Sim</i>	<i>Não</i>	<i>Não</i>	<i>Sim</i>	<i>Alguns</i>	<i>\$\$\$</i>	<i>Não</i>	<i>Sim</i>	<i>Francês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

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



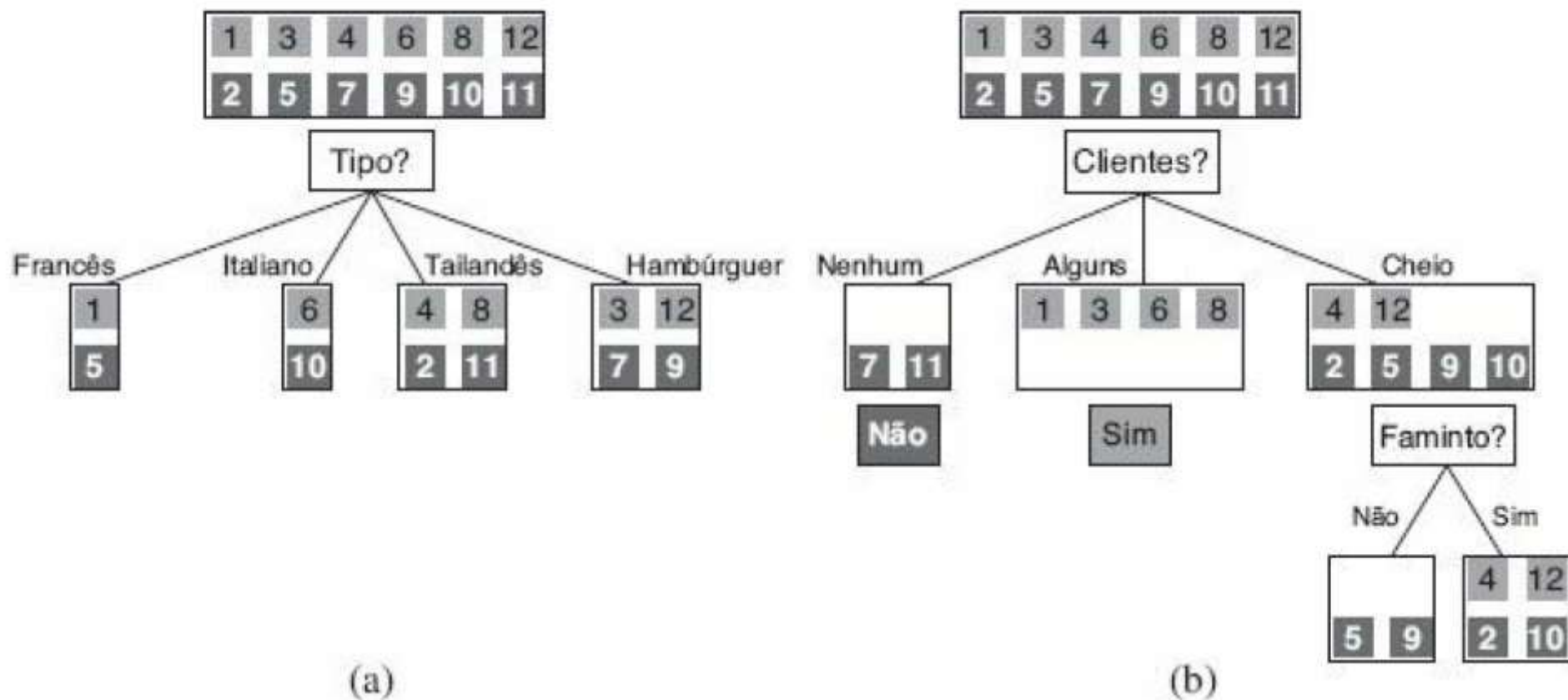


Aprendizagem em Árvores de Decisão





Aprendizagem em Árvores de Decisão





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

$A \leftarrow \operatorname{argmax}_{a \in \text{atributos}} \text{IMPORTÂNCIA}(a, \text{exemplos})$

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

para cada valor v_k de A **faça**

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

$\text{subárvore} \leftarrow \text{APRENDIZAGEM-EM-ÁRVORE-DE-DECISÃO}(\text{exs}, \text{atributos} - A, \text{exemplos})$

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



Dados



Regras



Partições



Decisão

(Regra de Parada)

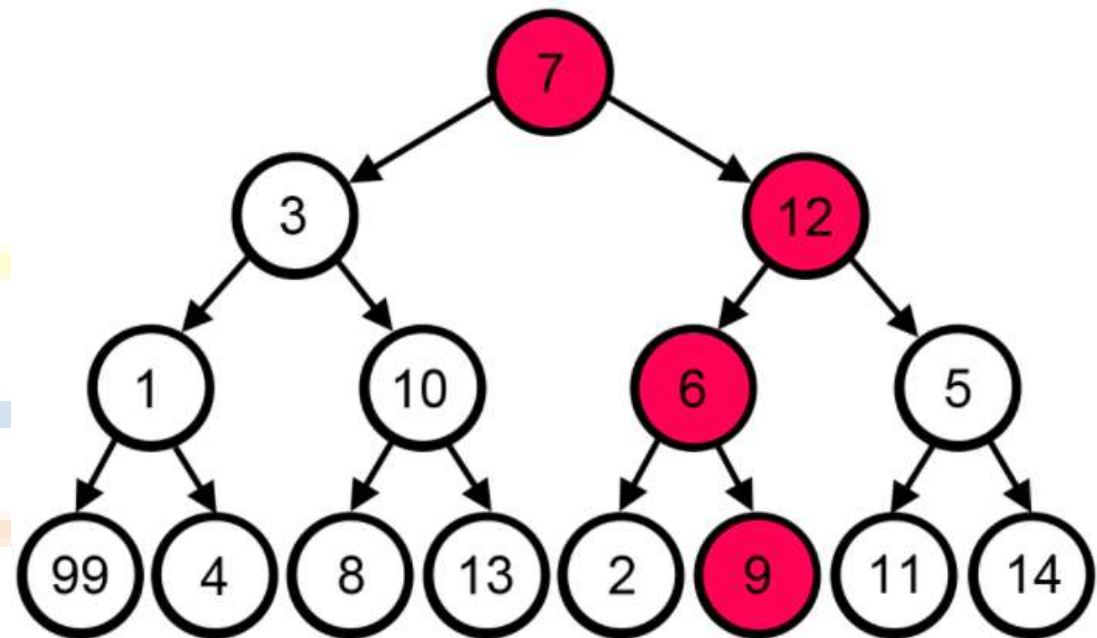




Aprendizagem em Árvores de Decisão

Greedy Search (Busca Gananciosa ou Gulosa)

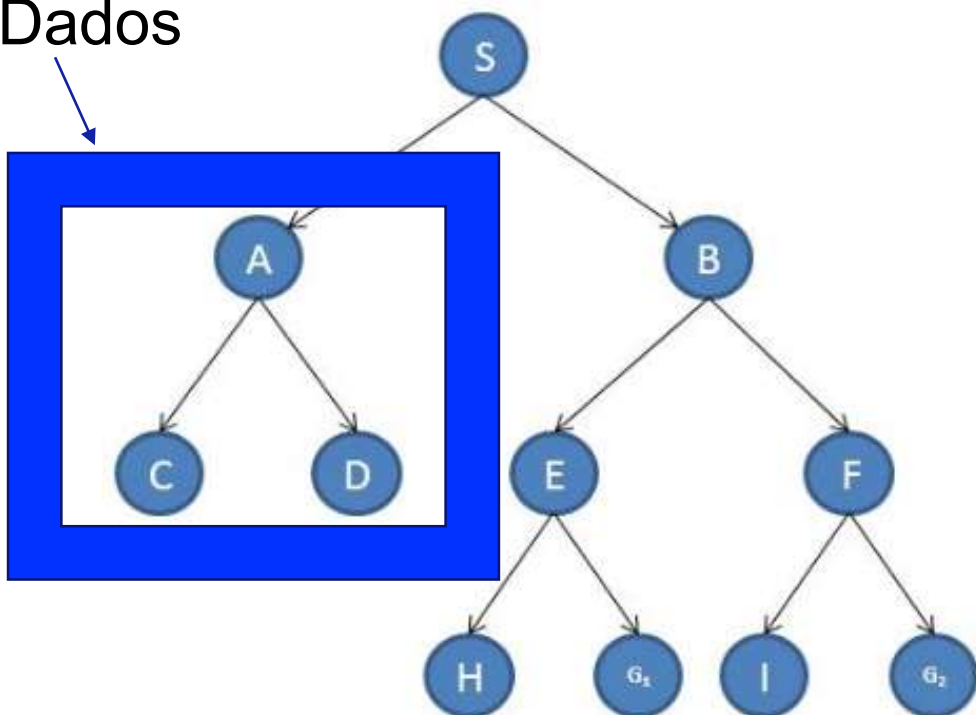
O algoritmo procura maximizar o passo atual sem olhar para o passo seguinte, a fim de alcançar uma otimização global.





Aprendizagem em Árvores de Decisão

Partição
de Dados



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

S → Estado inicial
G1, G2 → 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

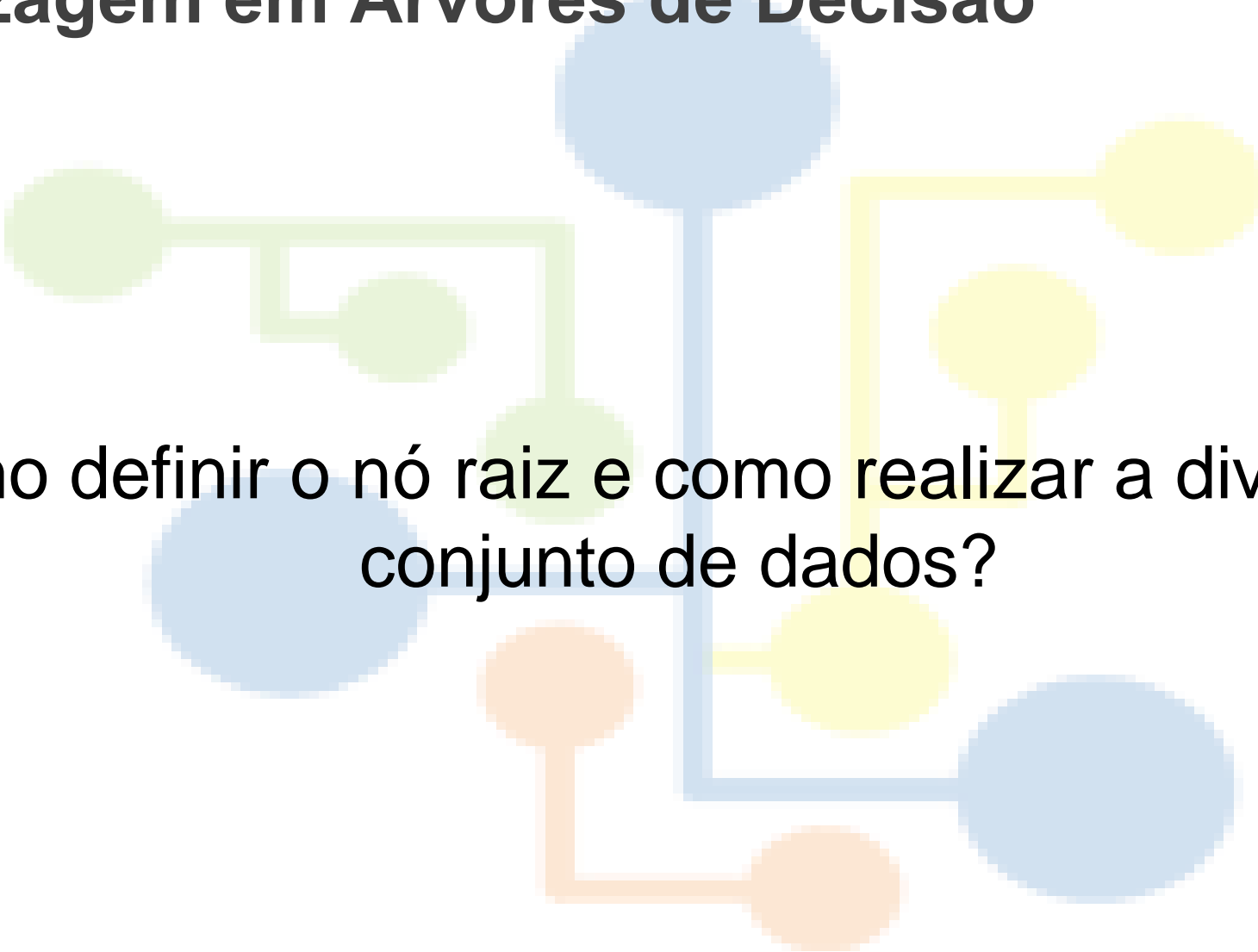
Ross Quinlan \rightarrow (ID3) \rightarrow C4.5 \rightarrow C5.0





Aprendizagem em Árvores de Decisão

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



Training Set

Test Set

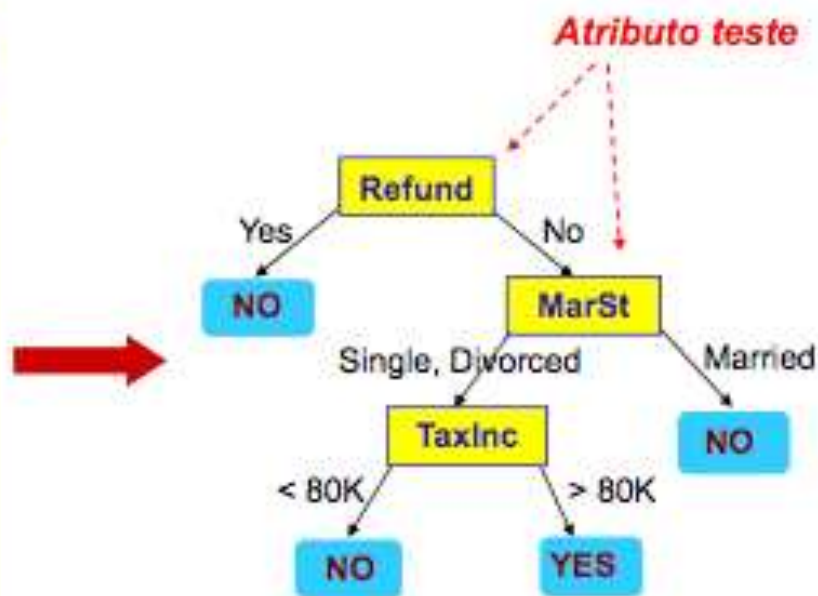
Classificação



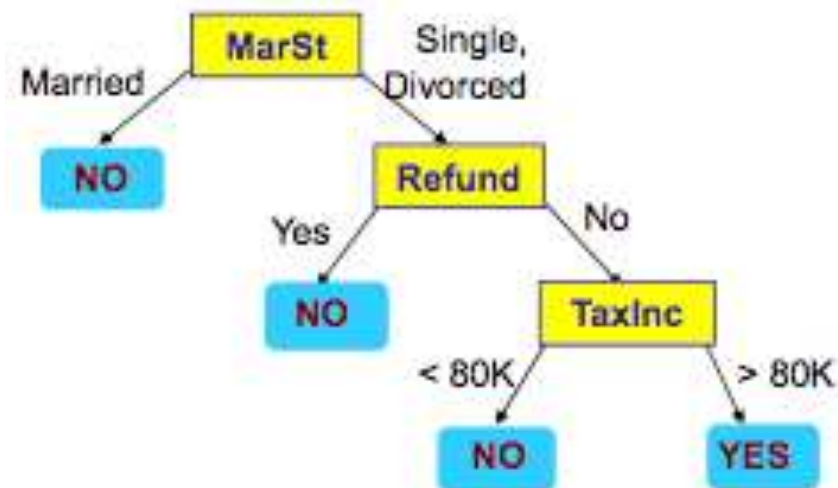
Aprendizagem em Árvore de Decisão

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 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

- Entropia
- Índice de Gini
- Erro de Classificação

C0: 5
C1: 5

Não-homogênea,
Alto grau de impureza

C0: 9
C1: 1

Homogêneo,
baixo grau de impureza





Aprendizagem em Árvores de Decisão

Entropia é a medida da incerteza nos dados

Ganho de Informação é a redução da Entropia

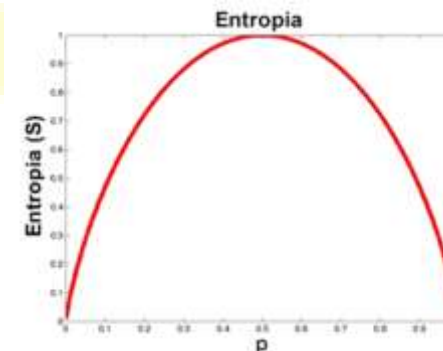




Aprendizagem em Árvores de Decisão

Entropia

$$Entropy = \sum -p_i \log_2 p_i$$



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

$$Entropy = -0.5 \cdot \log_2(0.5) - 0.5 \cdot \log_2(0.5) = 1.0$$

Entropia considerando duas classes com distribuição 40/60:

$$Entropy = -0.4 \cdot \log_2(0.4) - 0.6 \cdot \log_2(0.6) = 0.97$$





Aprendizagem em Árvores de Decisão



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.





Data Science
Academy

Data Science Academy angelicogfa@gmail.com 5b81f7e45e4cdea2118b4569

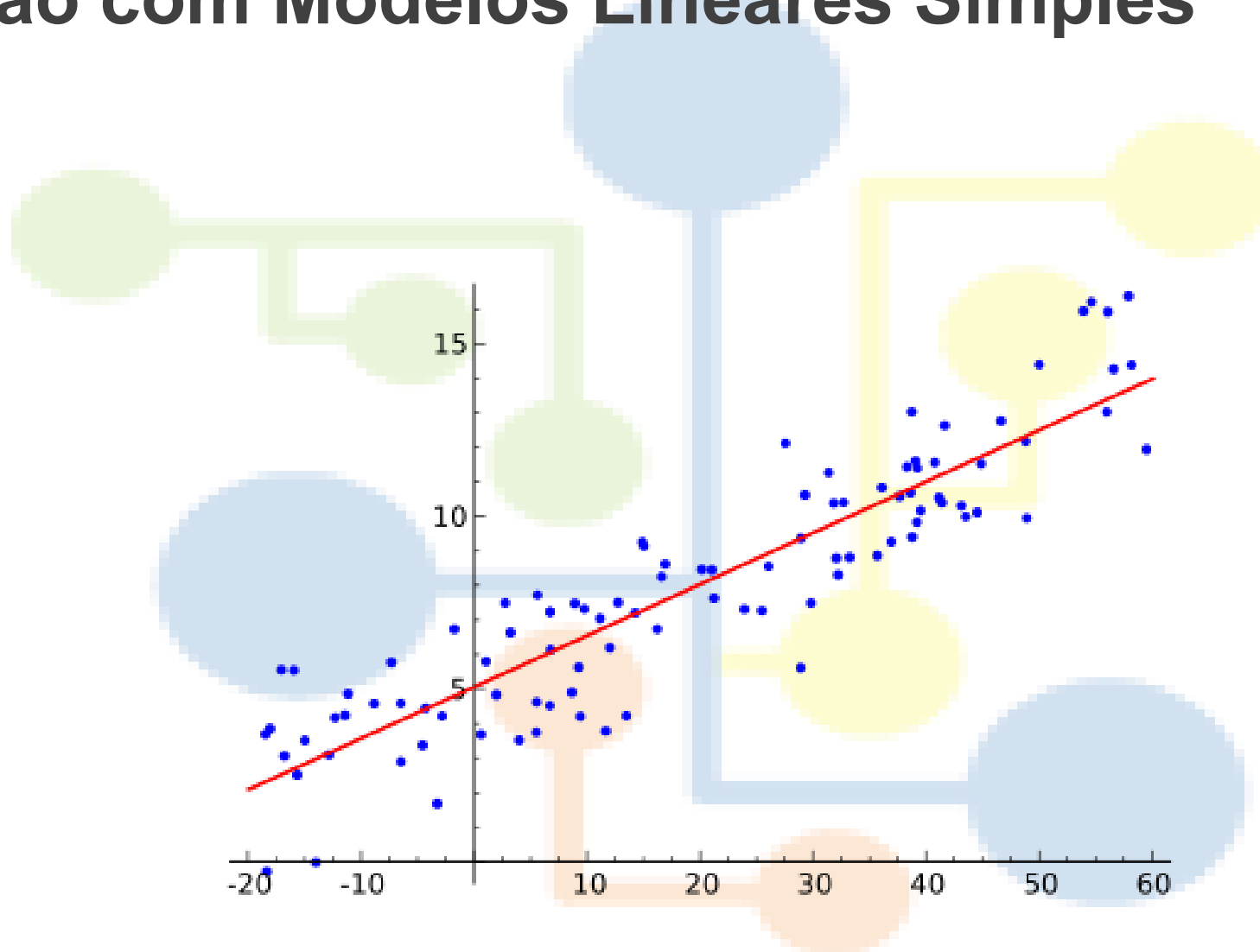
Regressão com Modelos Lineares Simples



Data Science Academy



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)

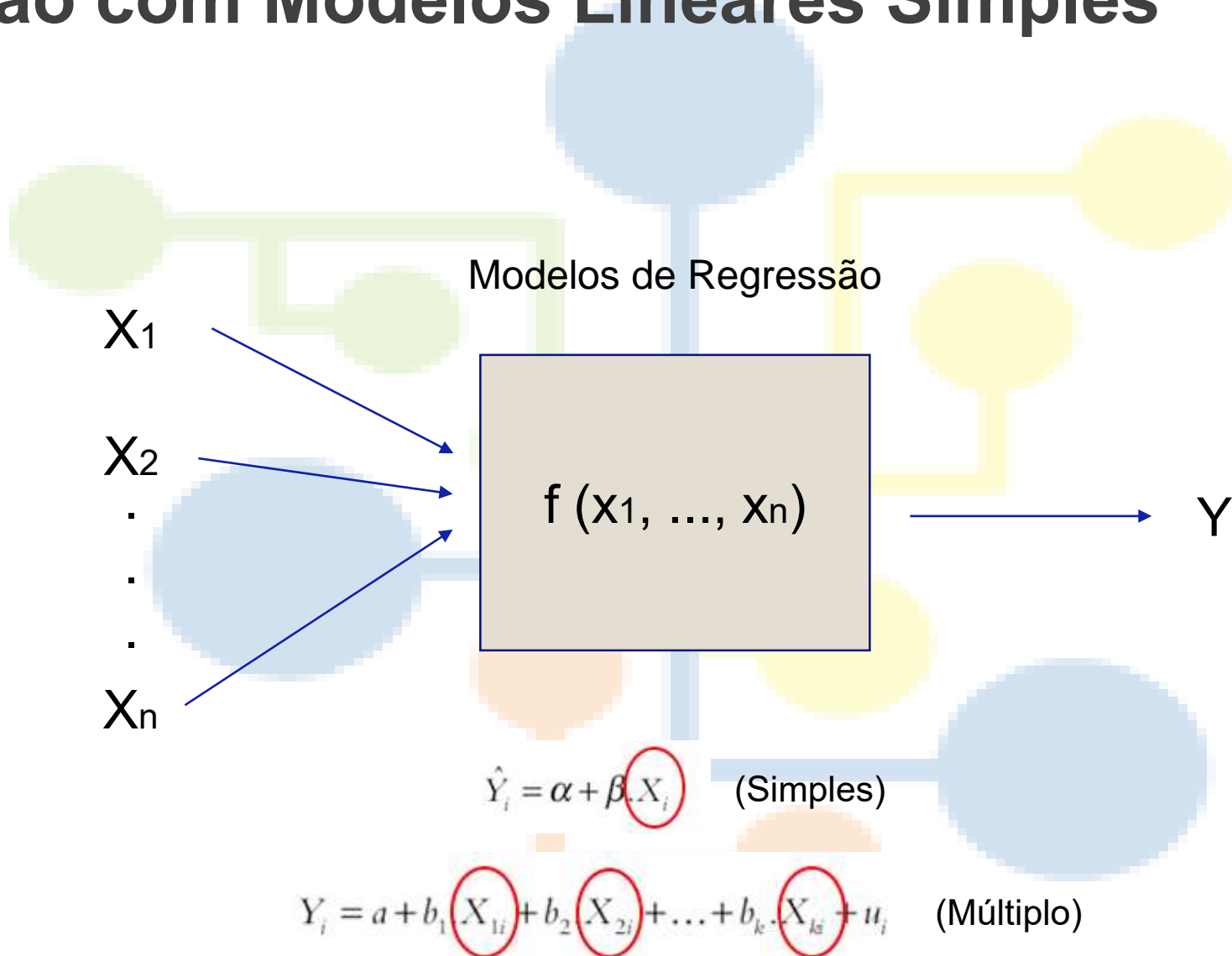
Regressão

- 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) \rightarrow variável dependente (y)





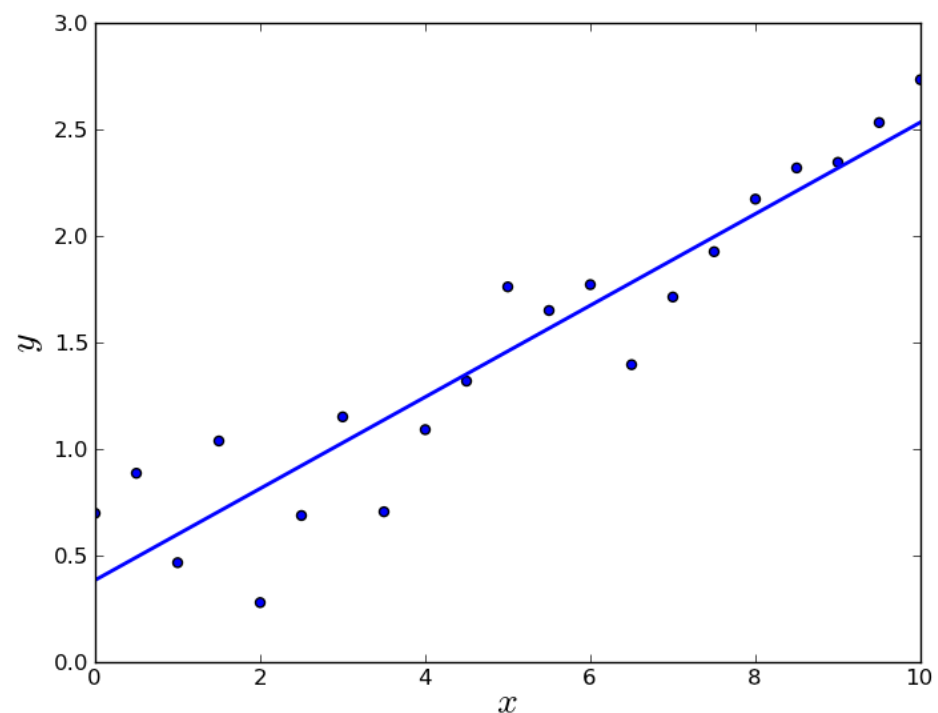
Regressão com Modelos Lineares Simples

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

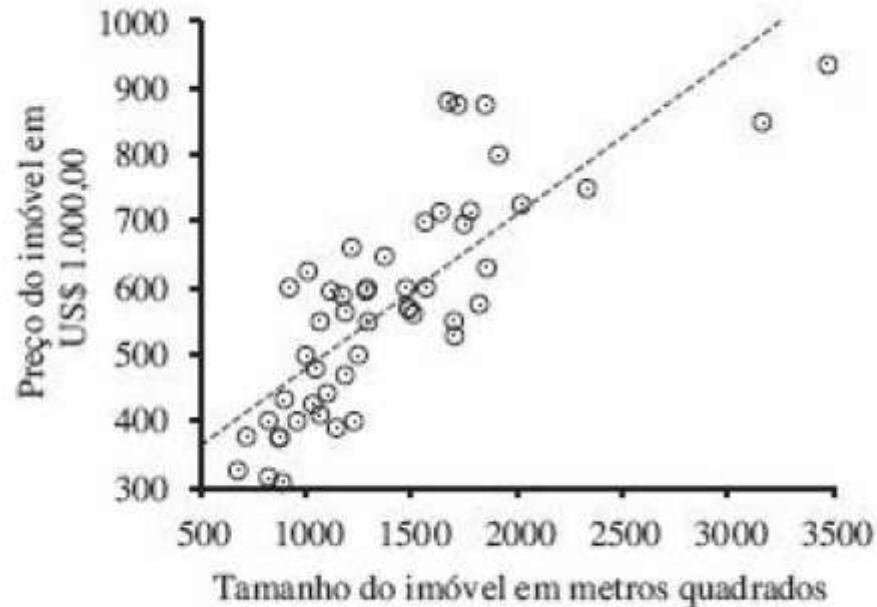


$$h_{\mathbf{w}}(x) = w_1x + w_0$$





Regressão com Modelos Lineares Simples



(a)

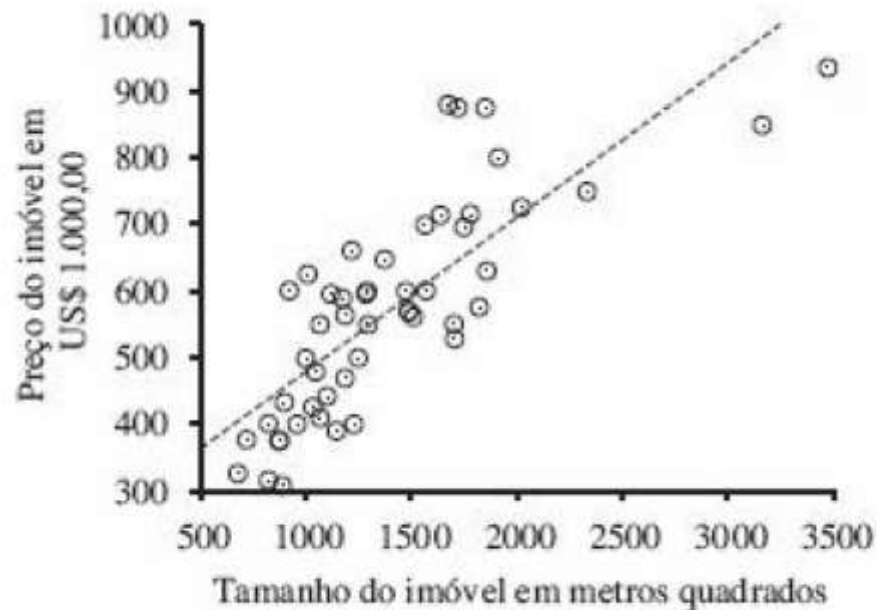
A tarefa de encontrar o **hw** que melhor se encaixe nesses dados é chamada de regressão linear.

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

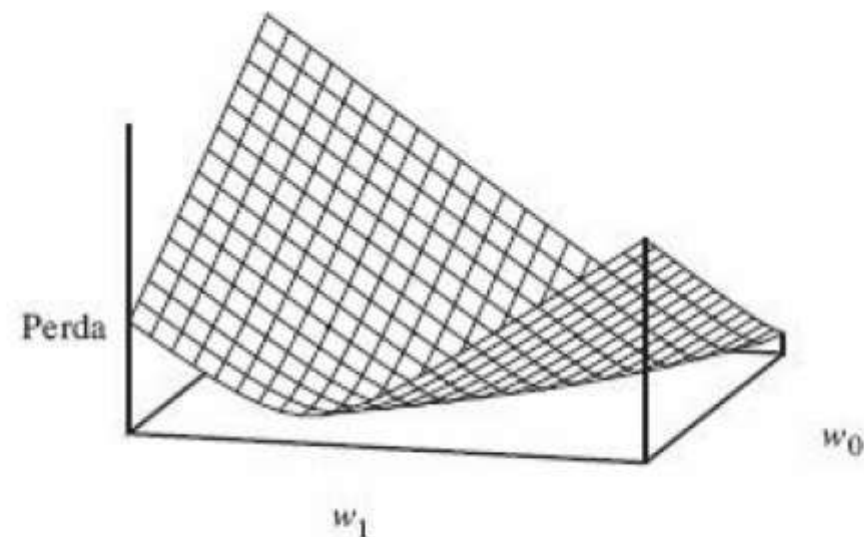




Regressão com Modelos Lineares Simples



(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

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

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





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

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

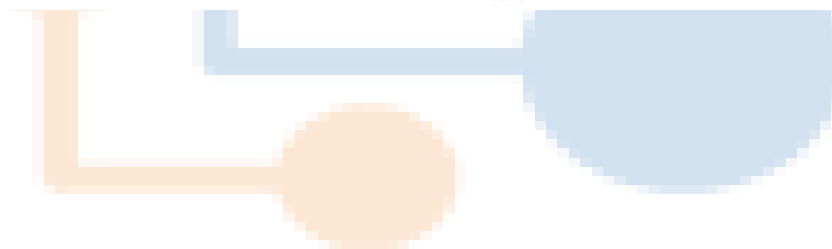




Regressão com Modelos Lineares Simples

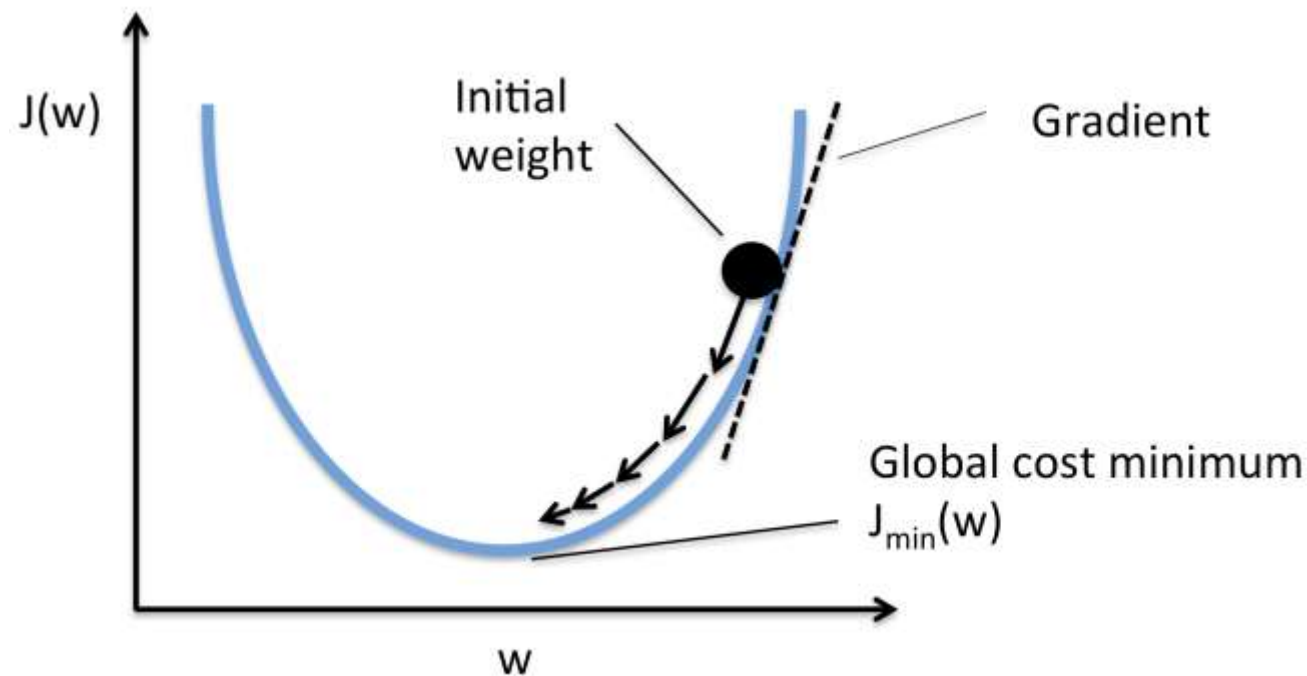
A derivada de uma soma é a soma das derivadas, por isso temos:

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





Regressão com Modelos Lineares Simples



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



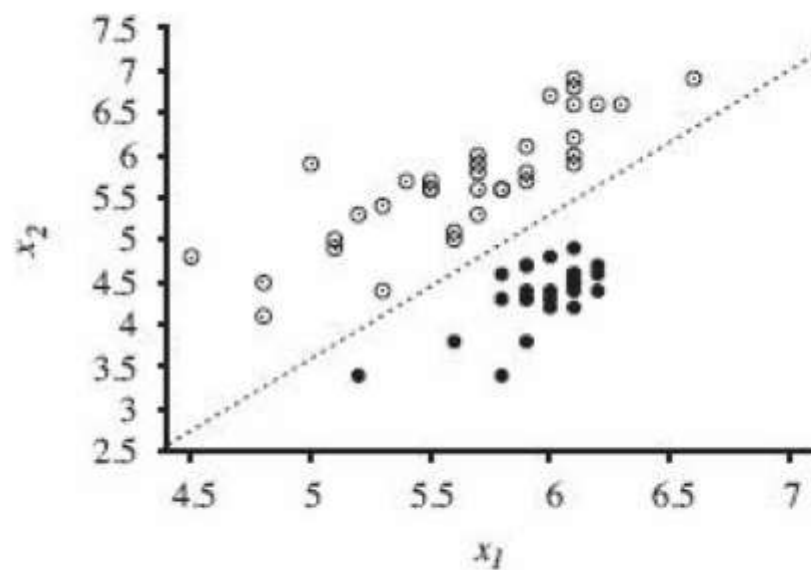


Classificação com Modelos Lineares

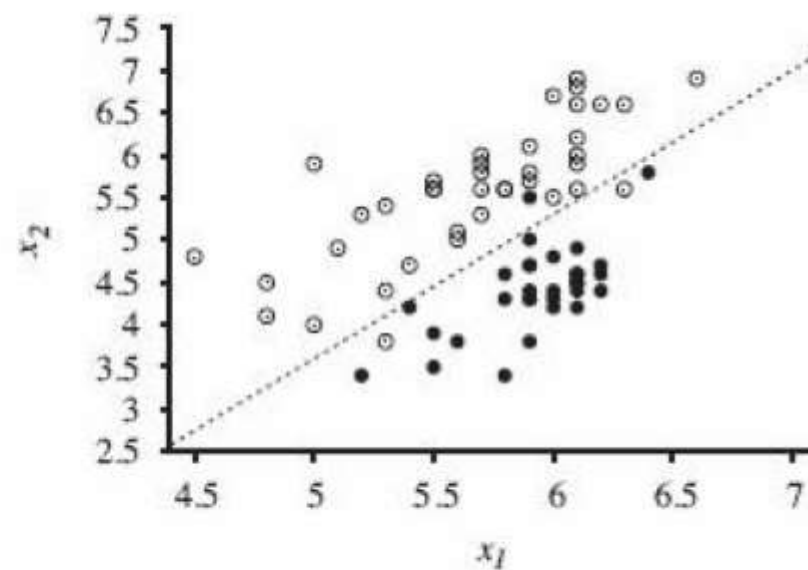




Classificação com Modelos Lineares



(a)



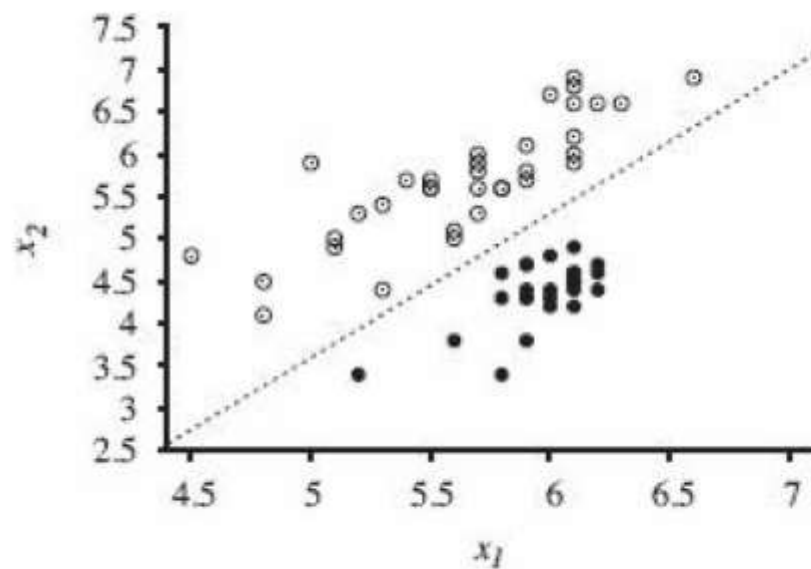
(b)



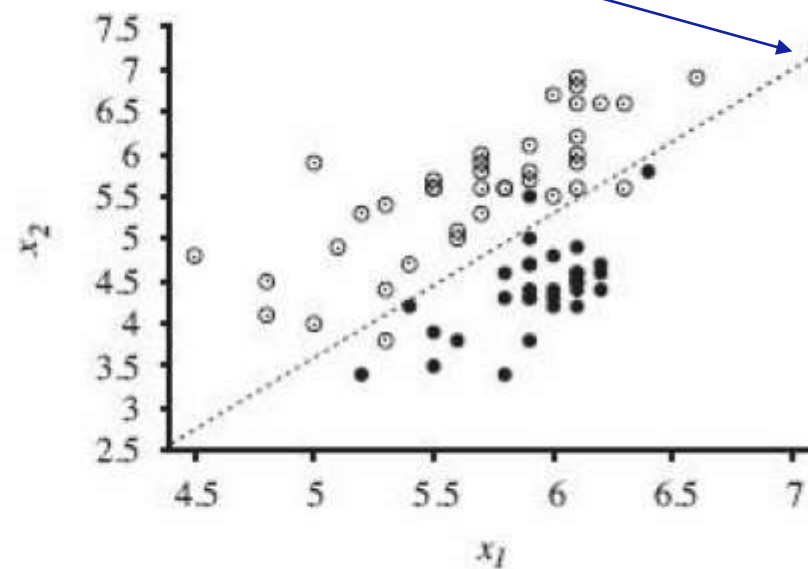


Classificação com Modelos Lineares

Fronteira de Decisão



(a)

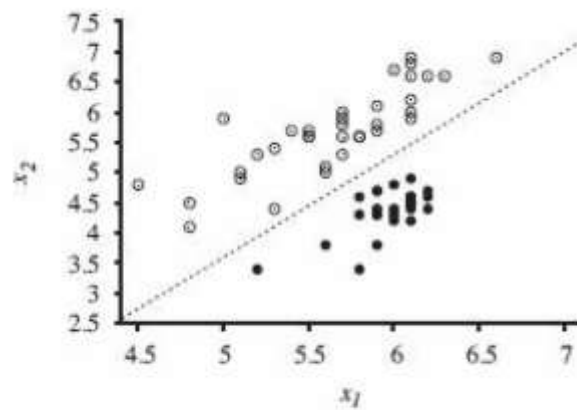


(b)

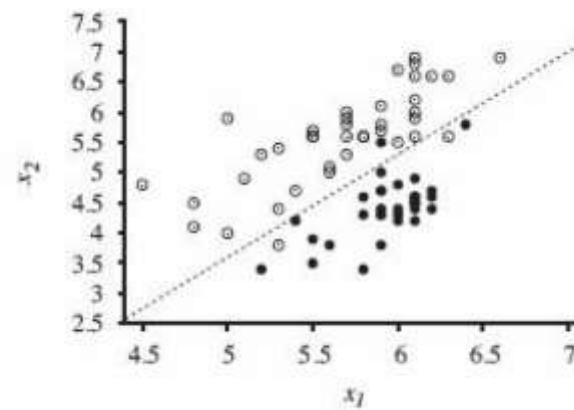




Classificação com Modelos Lineares



(a)



(b)

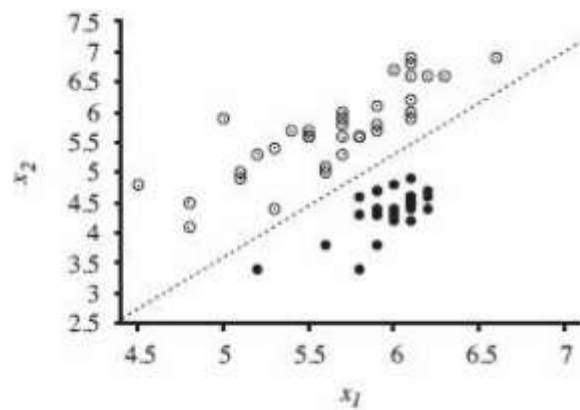
$hw(x) = 1$ se $w \cdot x \geq 0$ e 0 caso contrário

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

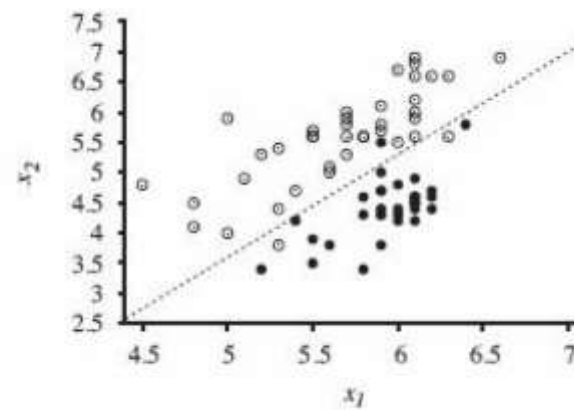




Classificação com Modelos Lineares



(a)



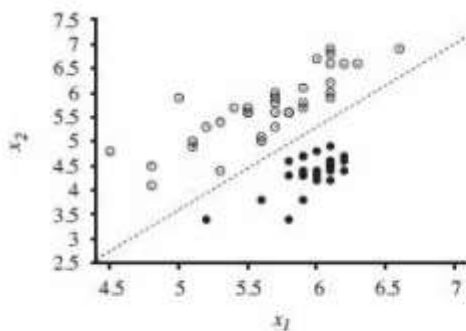
(b)

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

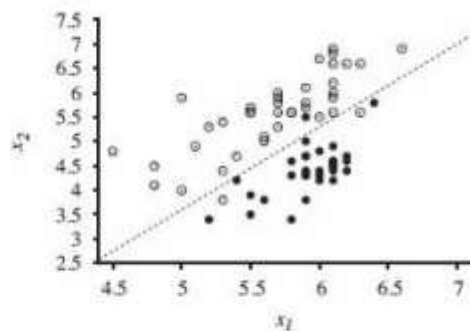




Classificação com Modelos Lineares



(a)



(b)

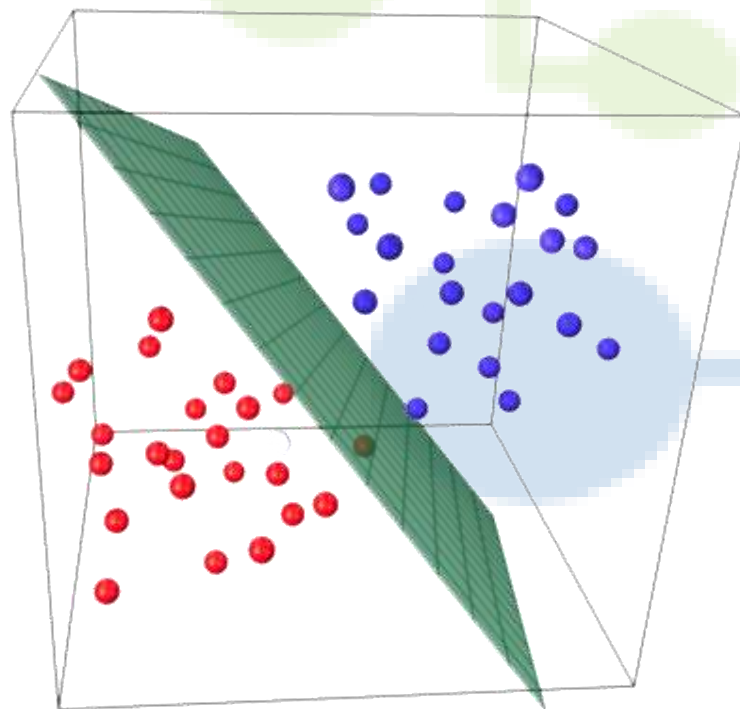
$$w_i \leftarrow w_i + \alpha (y - h_w(\mathbf{x})) \times x_i$$

- Se a saída está correta, ou seja, $y = h_w(\mathbf{x})$, os pesos não são alterados.
- Se y for 1, mas $h_w(\mathbf{x})$ 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 $h_w(\mathbf{x})$ gere um 1.
- Se y for 0, mas $h_w(\mathbf{x})$ 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 $h_w(\mathbf{x})$ gere um 0.





Classificação com Modelos Lineares

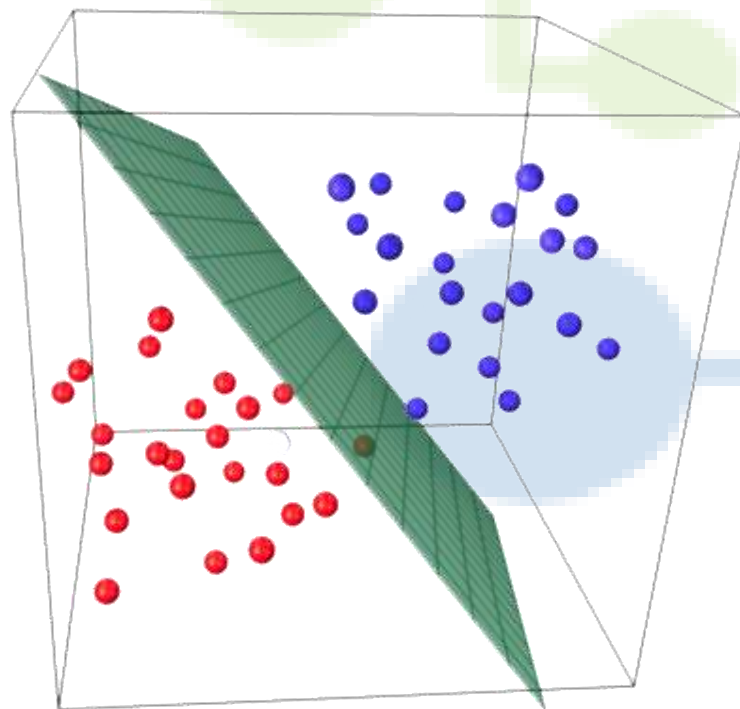


Além disso, o classificador linear sempre anuncia uma previsão completamente confiante de 1 ou 0, mesmo para exemplos que estão muito perto da fronteira (o que pode levar a classificações incorretas).





Classificação com Modelos Lineares



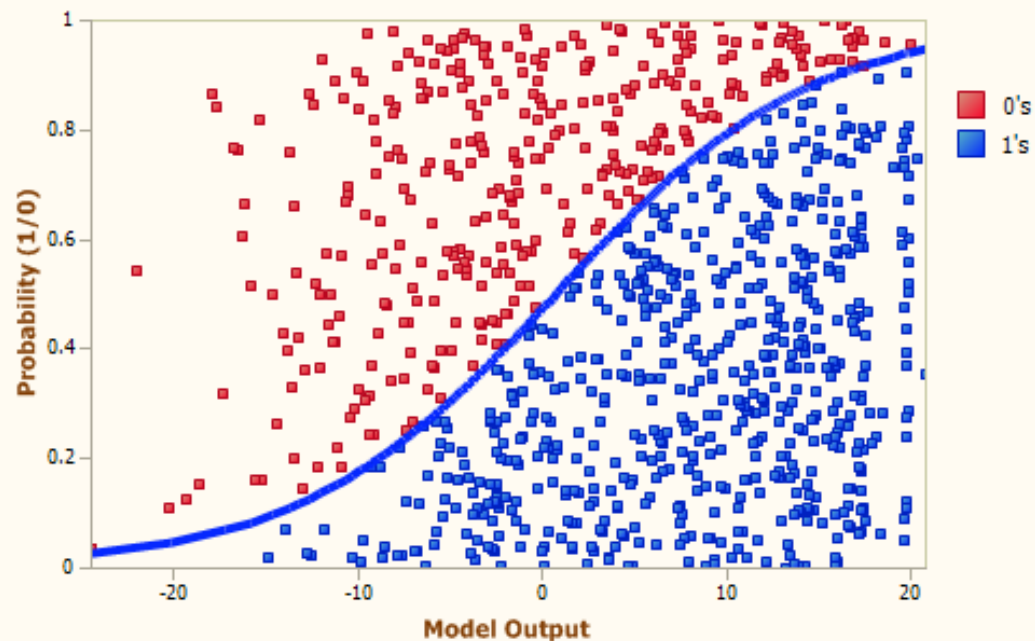
No modelo logístico a variável resposta é binária. Uma variável binária assume dois valores, como por exemplo, $Y = 0$ e $Y = 1$ denominados "fracasso" e "sucesso", respectivamente.

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





Classificação com Modelos Lineares

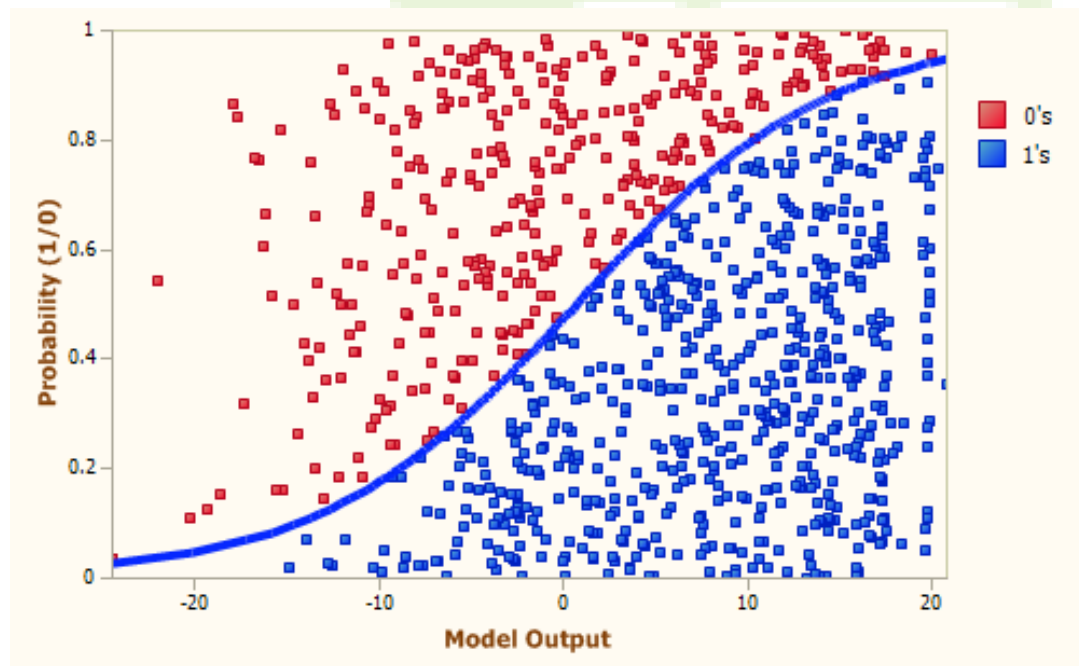


Os modelos de regressão constituem uma das ferramentas estatísticas mais importantes na análise estatística de dados, quando se pretende modelar relações entre variáveis.





Classificação com Modelos Lineares

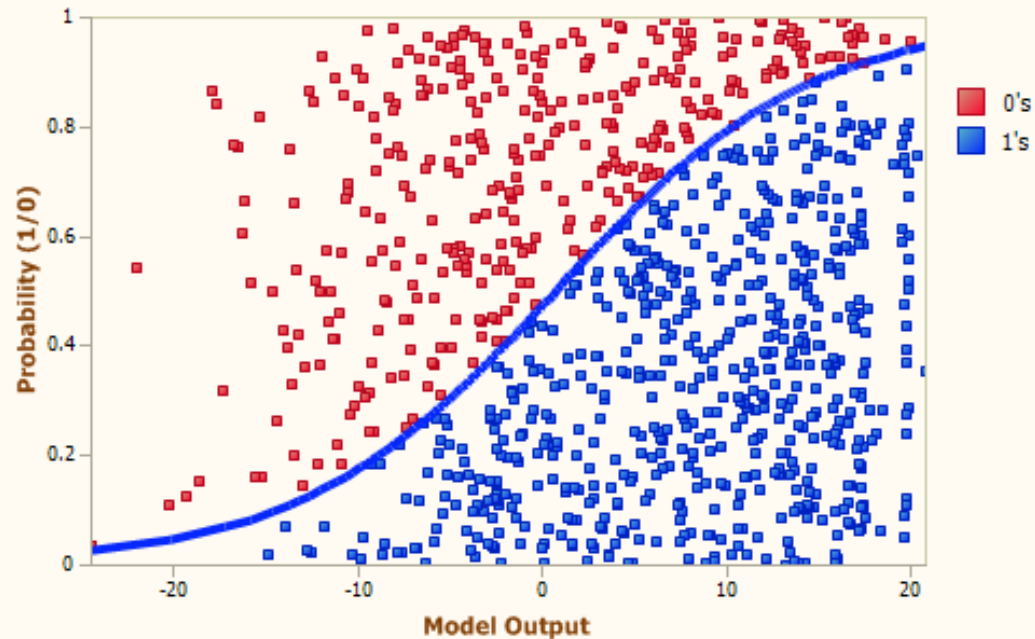


A regressão logística é uma técnica estatística que tem como objetivo modelar, a partir de um conjunto de observações, a relação “logística” entre uma variável resposta e uma série de variáveis explicativas numéricas (contínuas, discretas) e/ou categóricas.





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

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

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

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

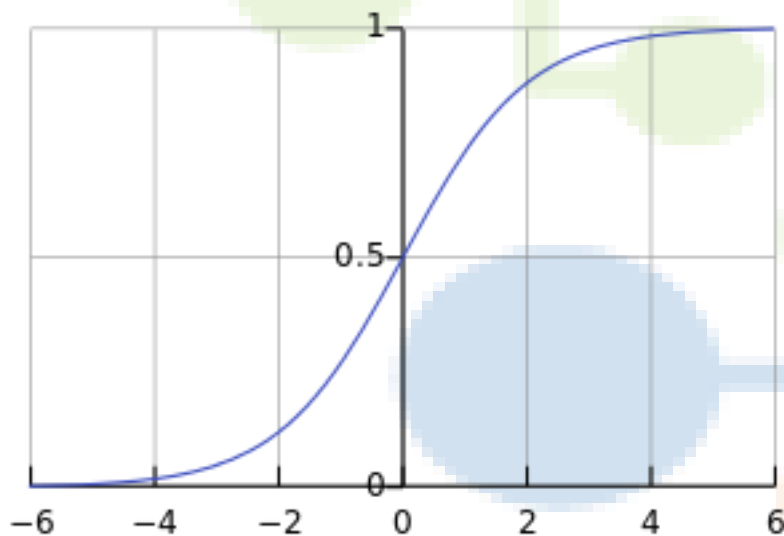
Logaritmo

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







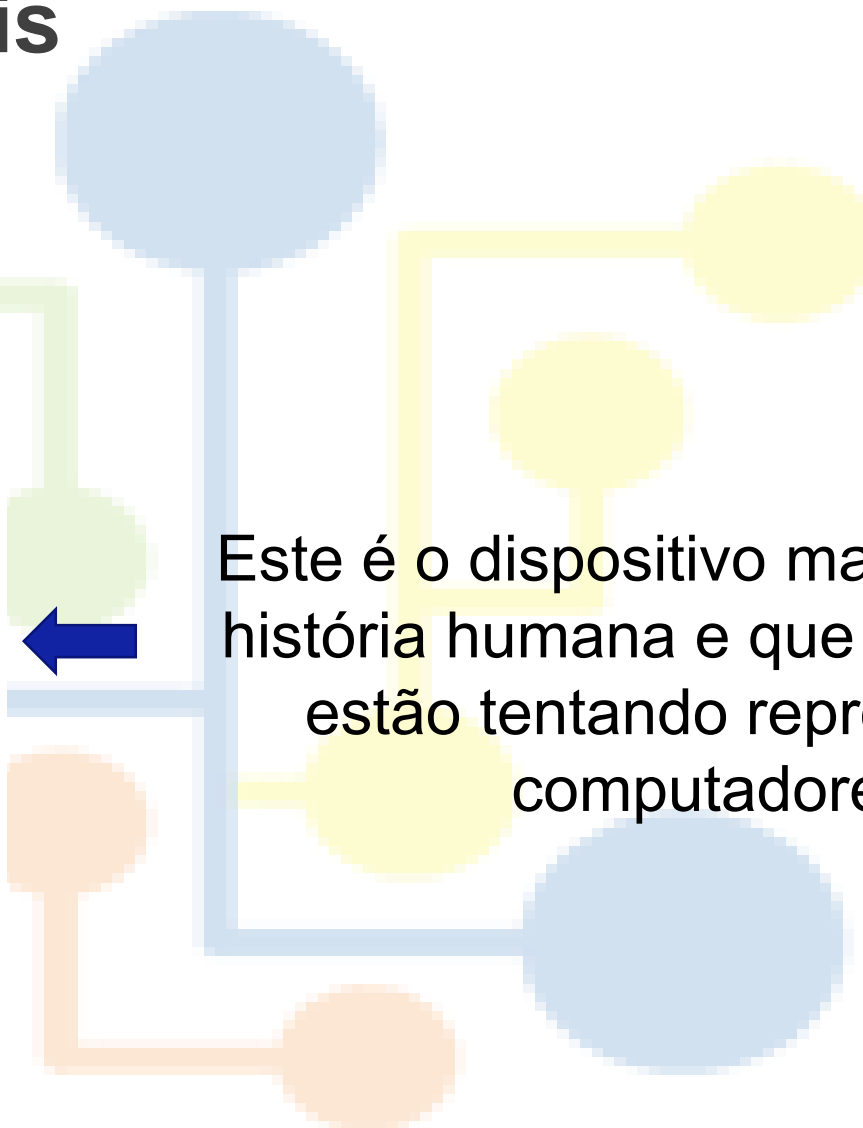
Redes Neurais Artificiais

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





Redes Neurais Artificiais

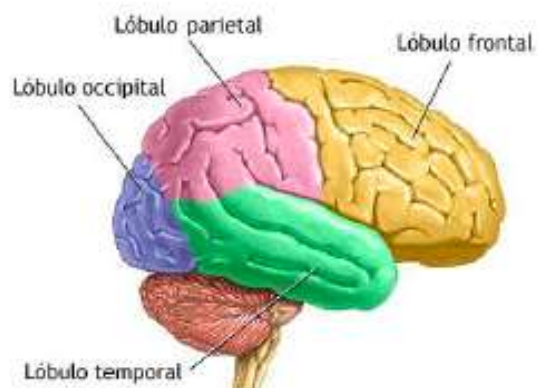


Este é o dispositivo mais incrível da história humana e que os cientistas estão tentando reproduzir em computadores!





Redes Neurais Artificiais

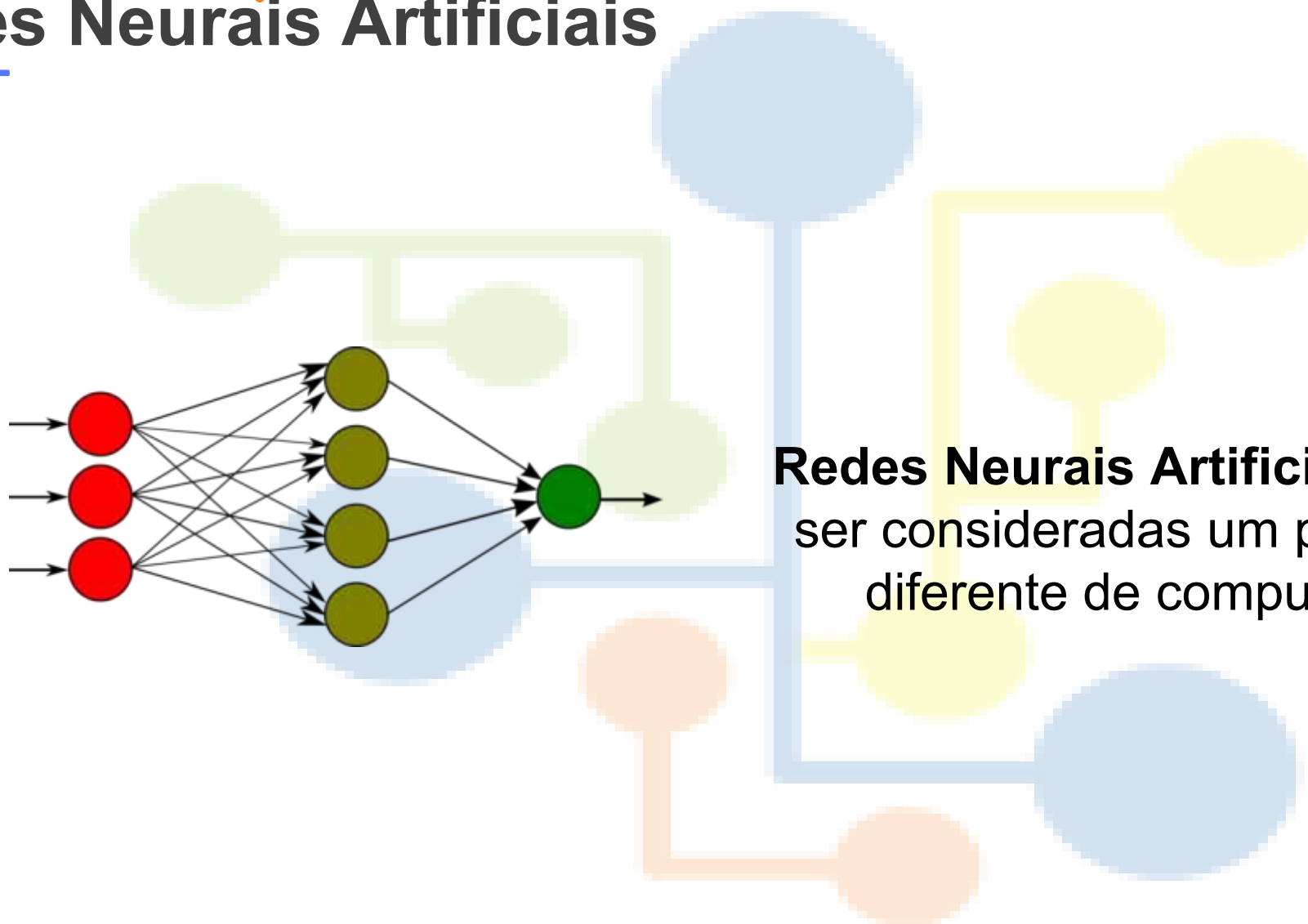


O **cérebro humano** tem sido extensamente estudado, mas ainda não somos capazes de **entender completamente** o seu funcionamento.





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**

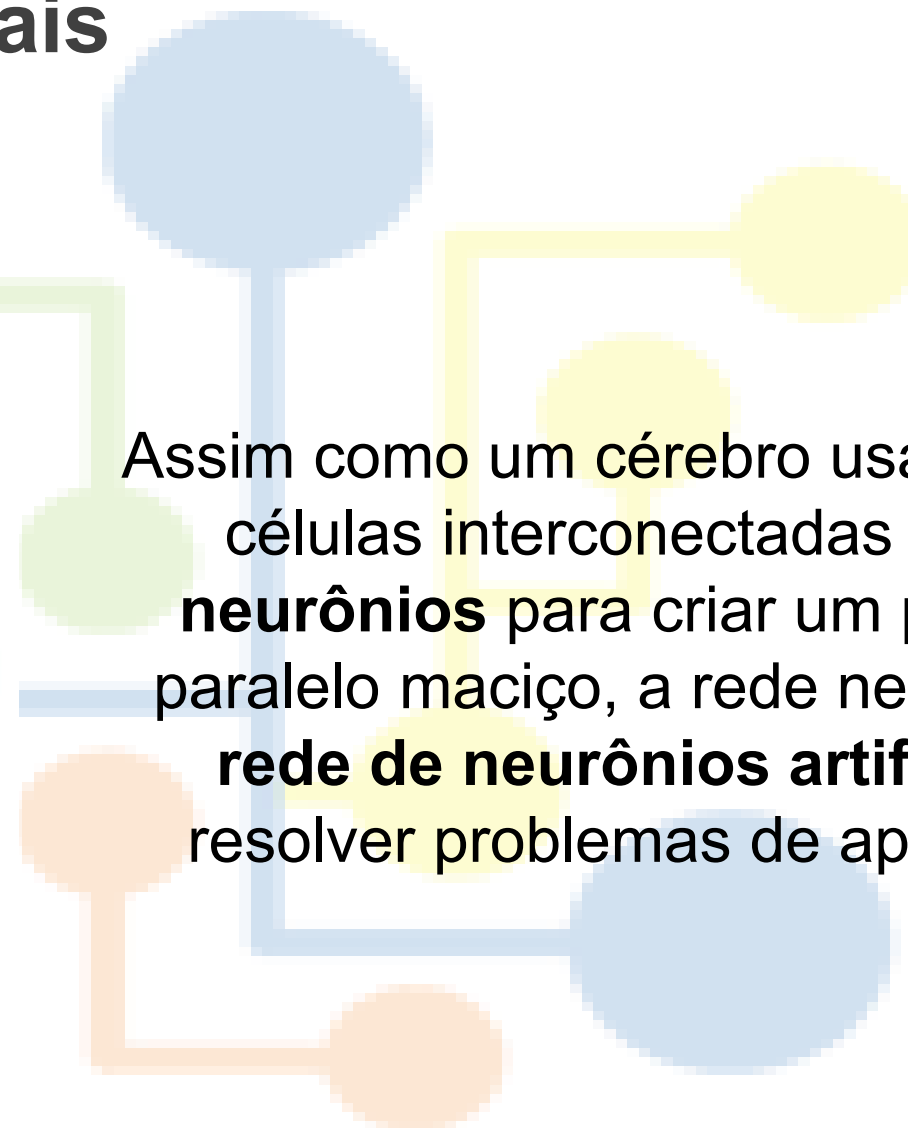




Redes Neurais Artificiais



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





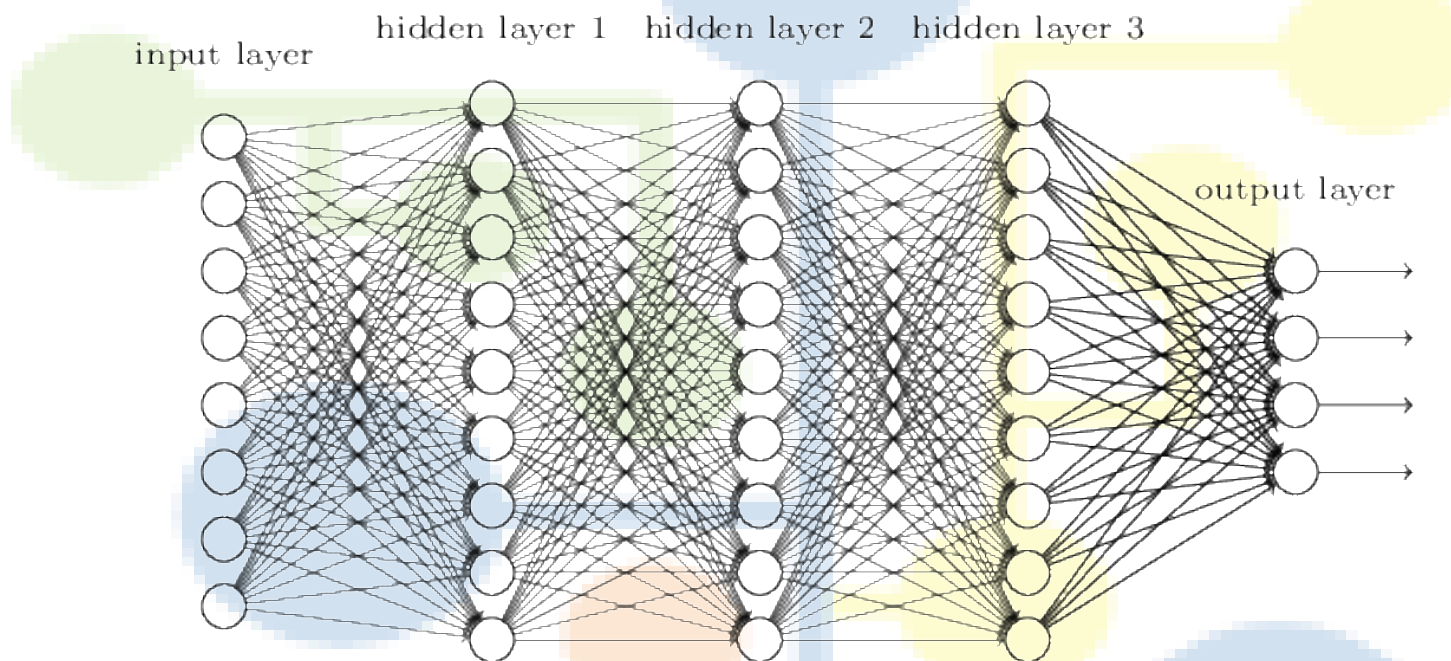
Redes Neurais Artificiais

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





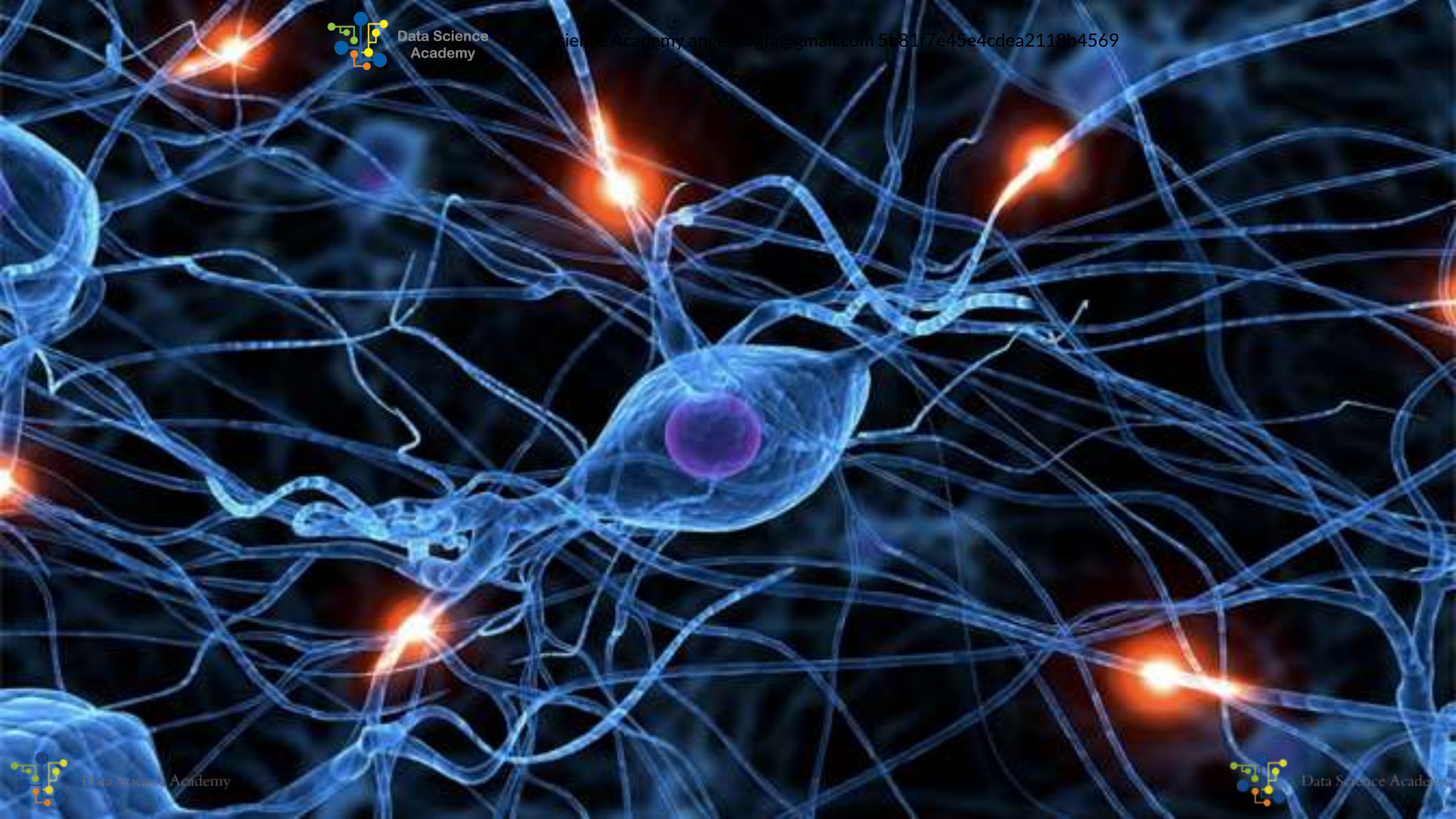
Redes Neurais Artificiais





Data Science
Academy

Data Science Academy angellogfa@gmail.com 5b81f7e45e4cdea2118b4569



Data Science Academy



Data Science Academy



Redes Neurais Artificiais



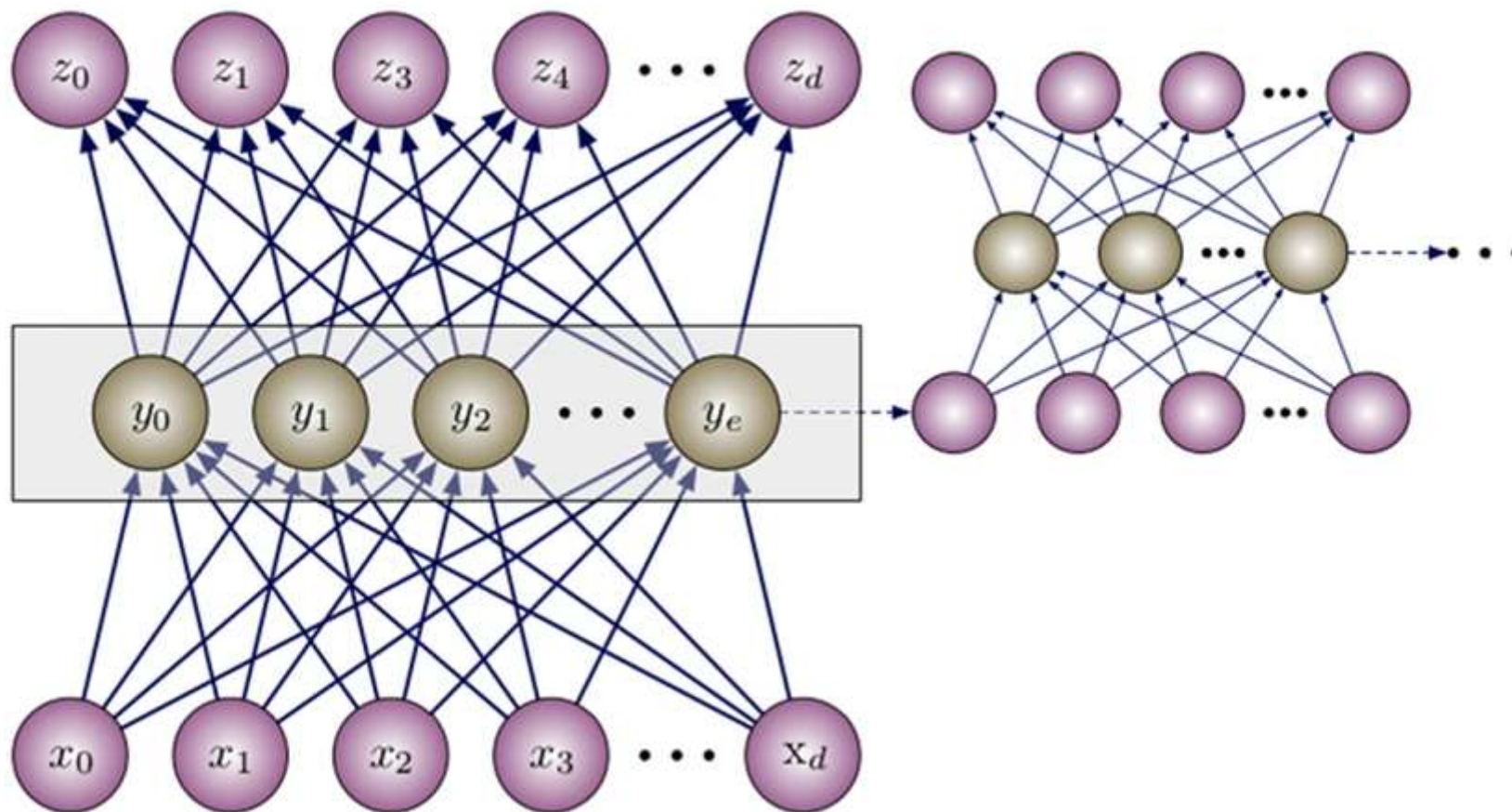
Warren McCulloch



Walter Pitts

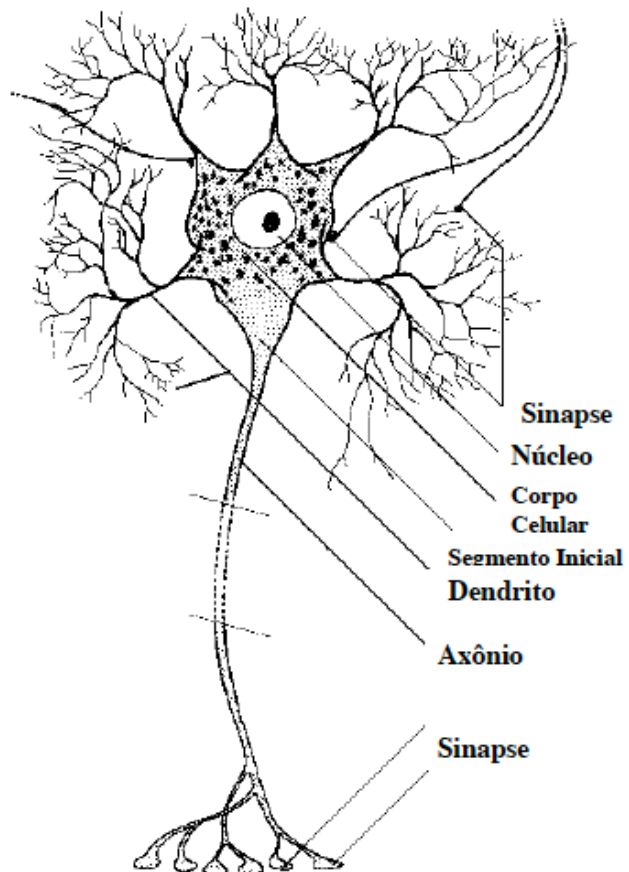


Redes Neurais Artificiais



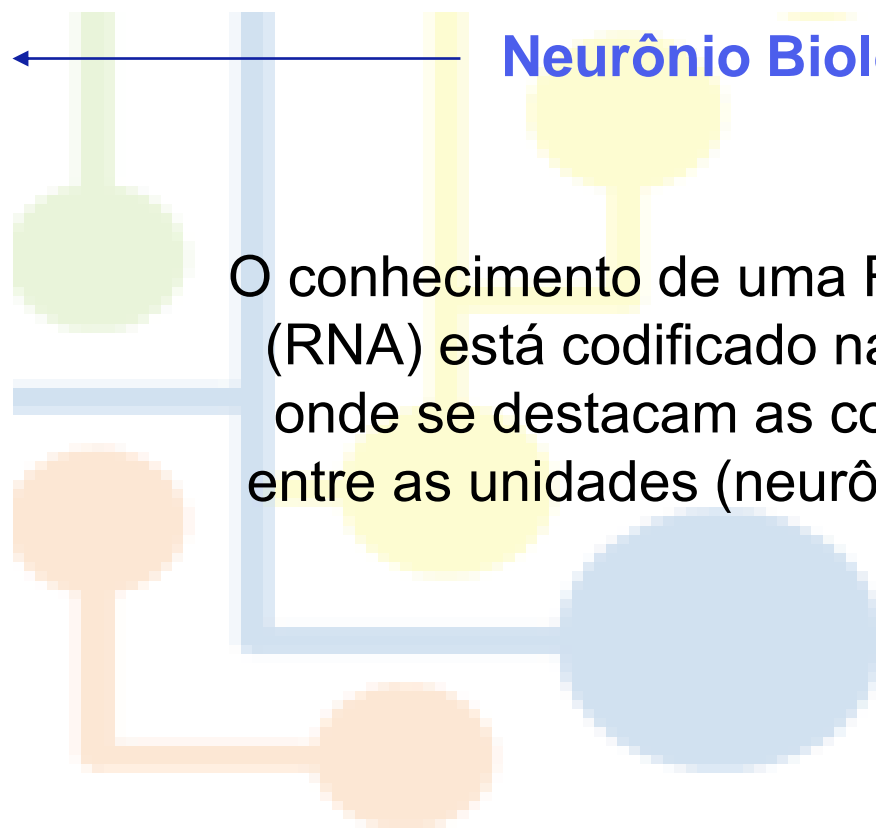


Redes Neurais Artificiais



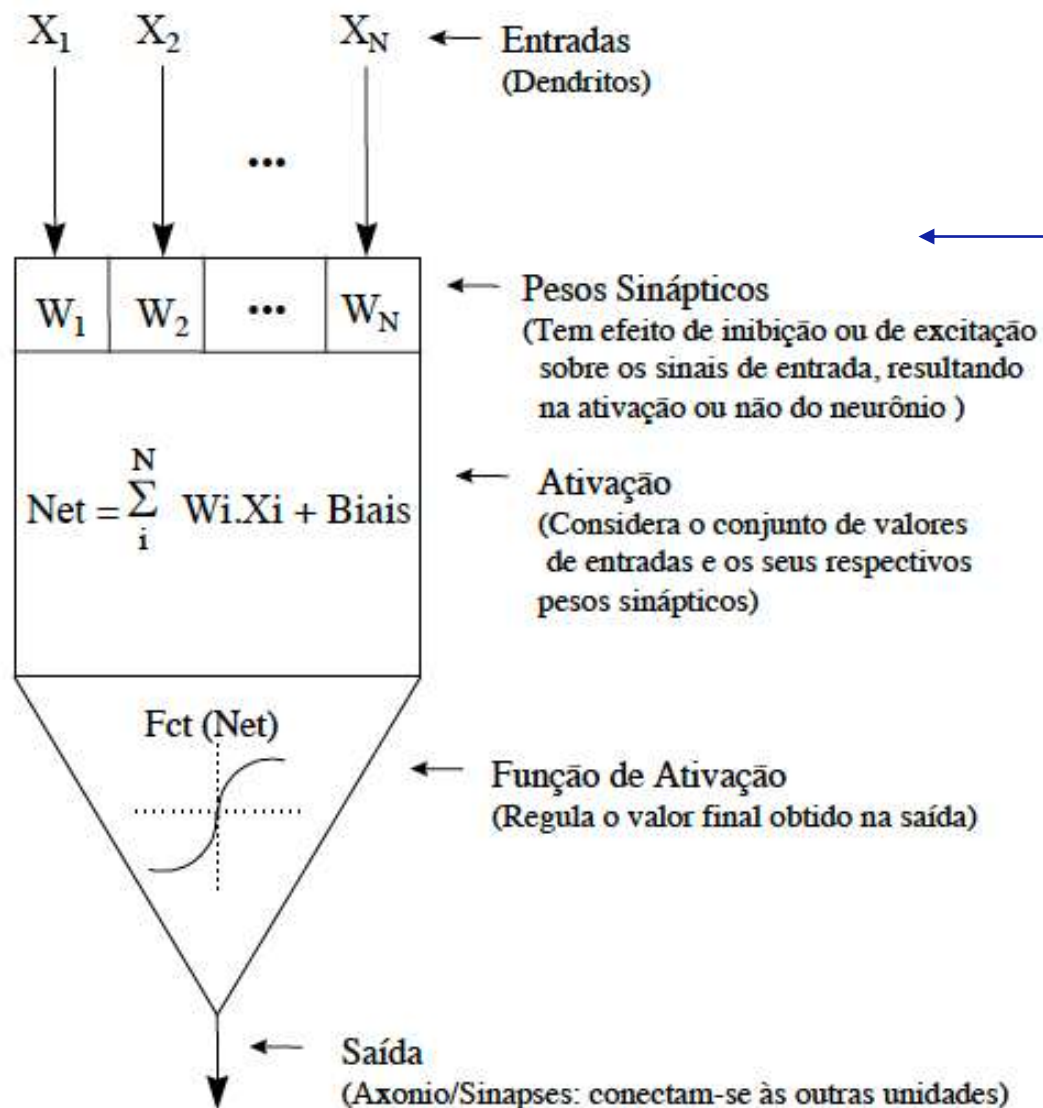
Neurônio Biológico

O conhecimento de uma Rede Neural Artificial (RNA) está codificado na estrutura da rede, onde se destacam as conexões (sinapses) entre as unidades (neurônios) que a compõe



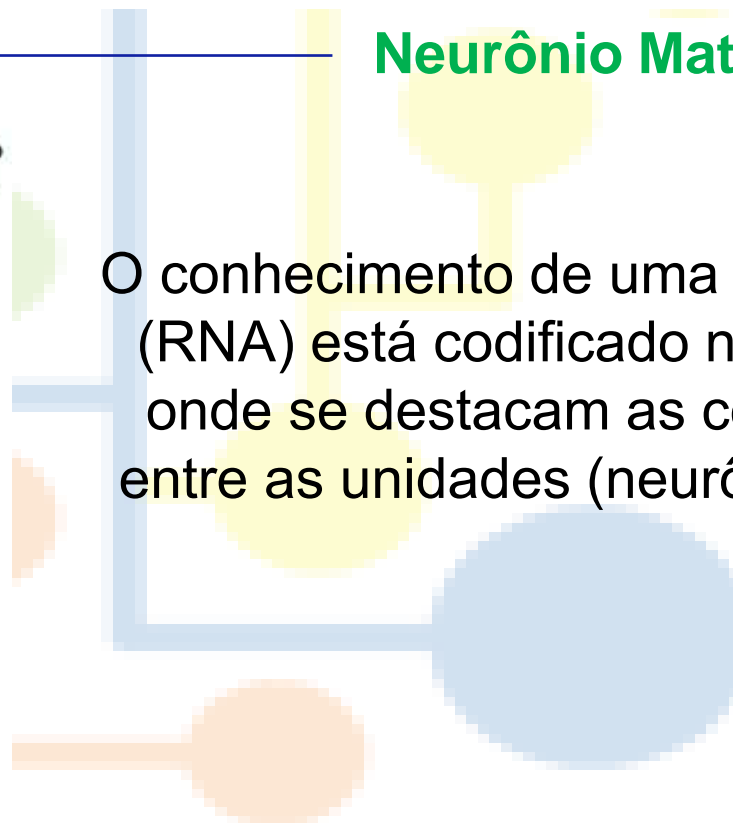
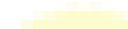


Redes Neurais Artificiais



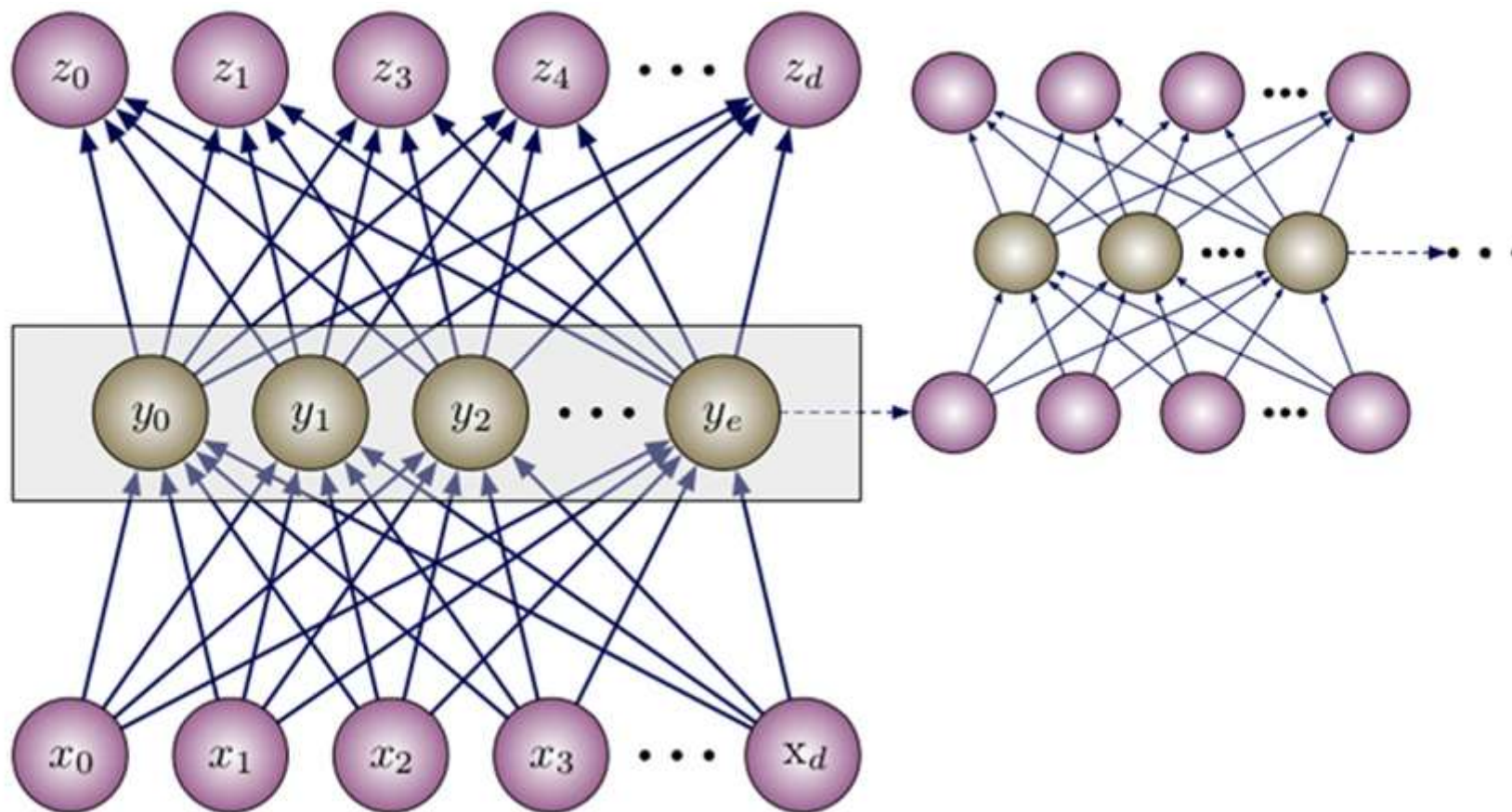
Neurônio Matemático

O conhecimento de uma Rede Neural Artificial (RNA) está codificado na estrutura da rede, onde se destacam as conexões (sinapses) entre as unidades (neurônios) que a compõe



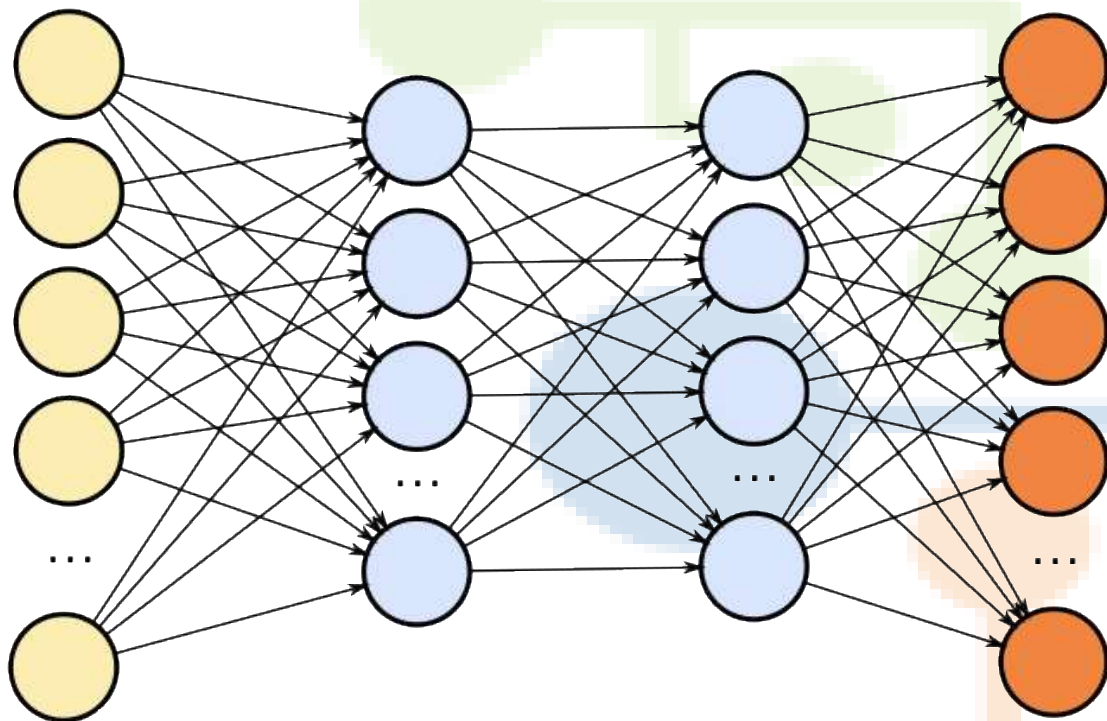


Redes Neurais Artificiais





Redes Neurais Artificiais

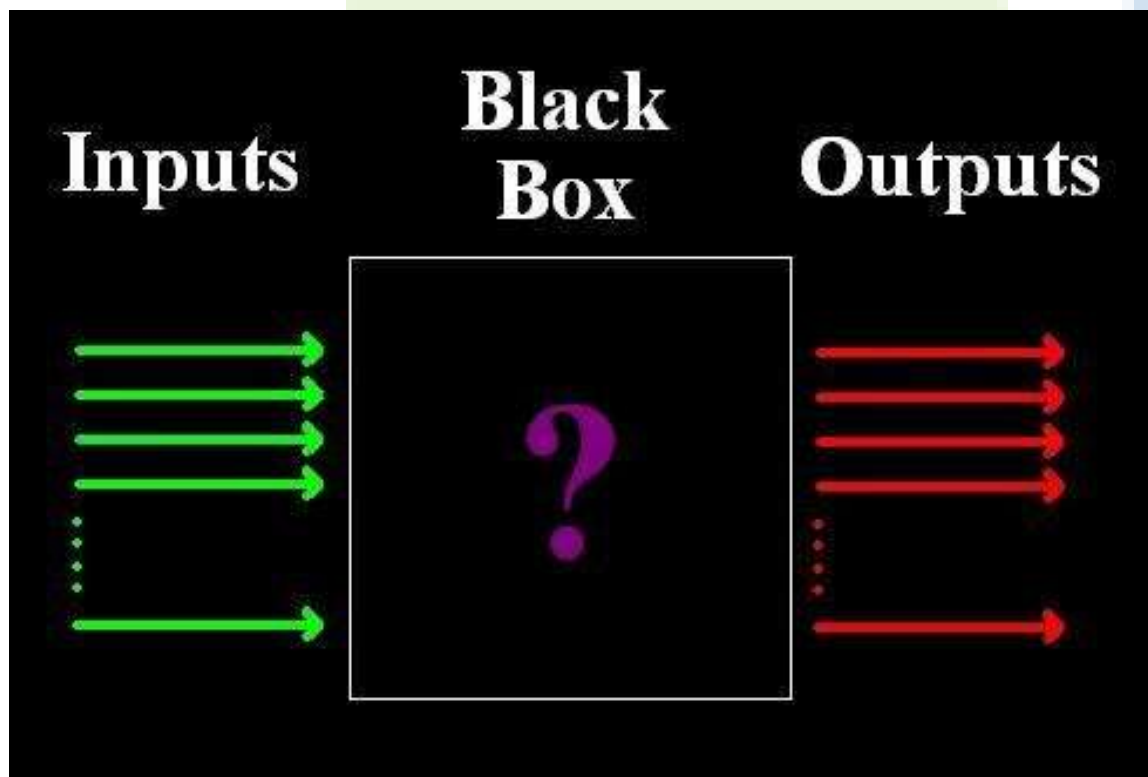


Um dos benefícios das redes diz respeito ao tratamento de um problema clássico da Inteligência Artificial, que é a representação de um universo não-estacionário (onde as estatísticas mudam com o tempo)





Redes Neurais Artificiais

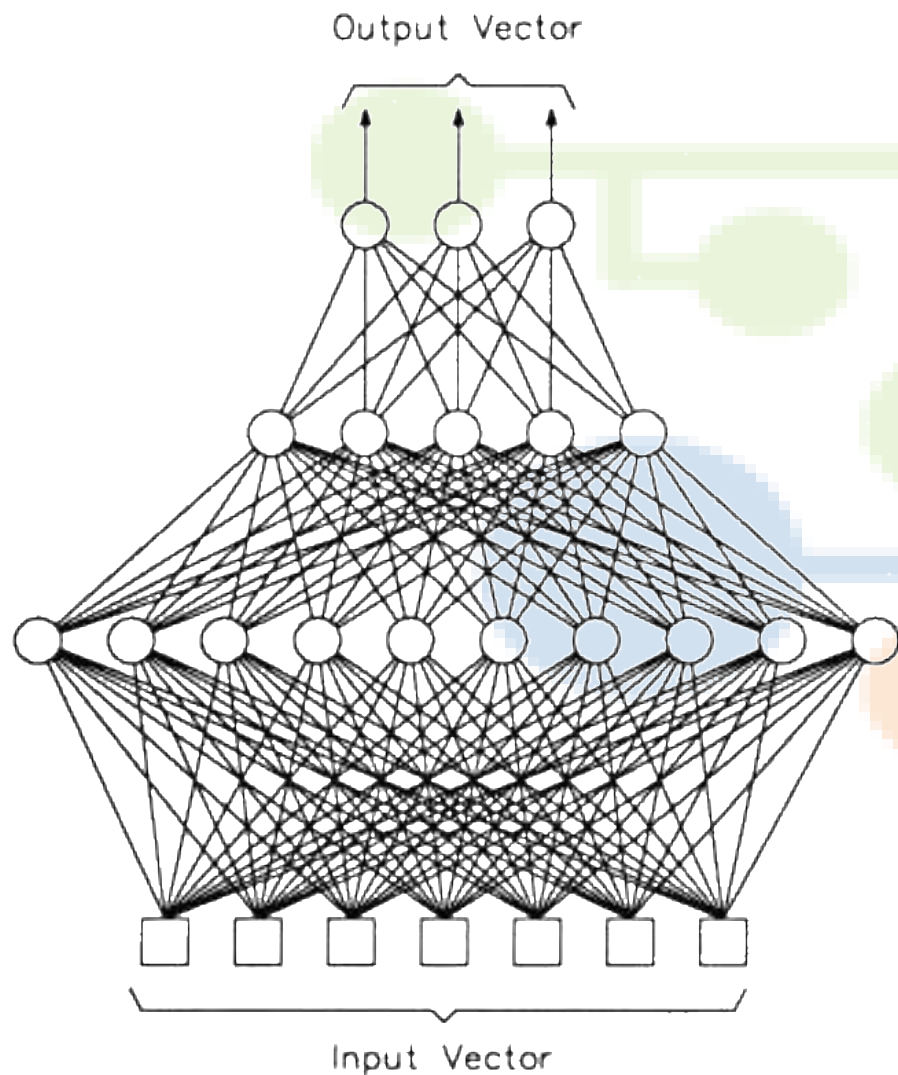


Uma desvantagem das redes neurais é o fato delas , normalmente, serem uma "caixa preta"





Redes Neurais Artificiais

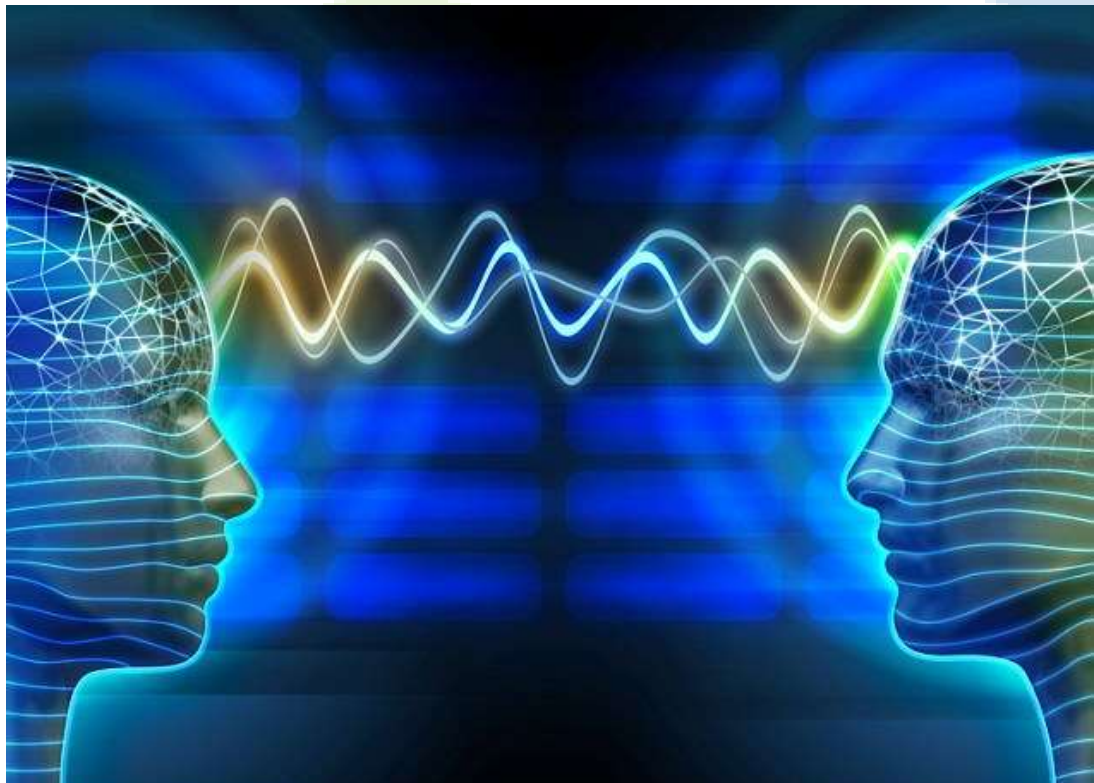


A solução de problemas através das RNAs é bastante atrativa, pois o paralelismo constitui-se na característica principal das RNAs, onde esta cria a possibilidade de um desempenho superior em relação a solução de problemas baseados nos modelos convencionais.





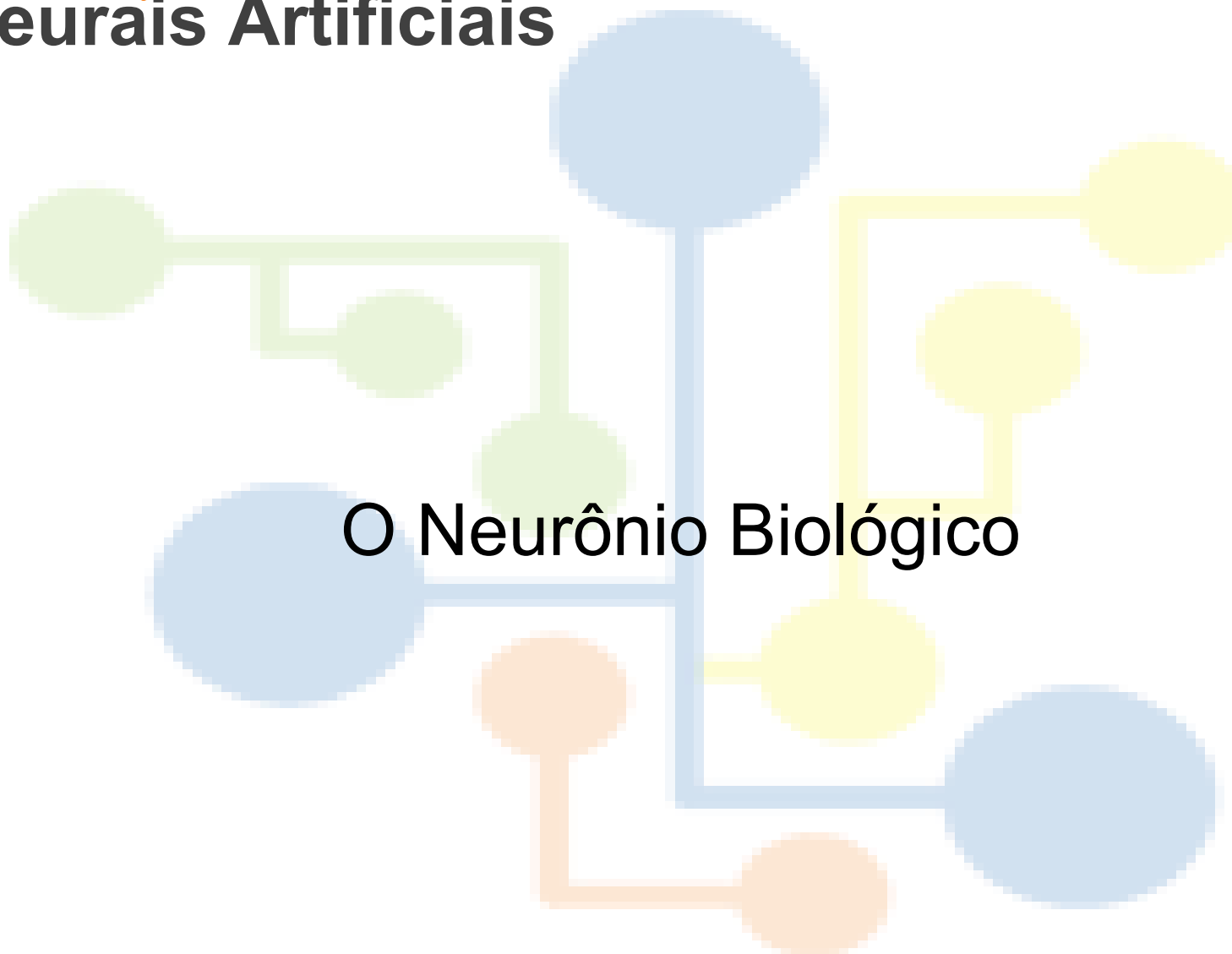
Redes Neurais Artificiais



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



O Neurônio Biológico



Redes Neurais Artificiais

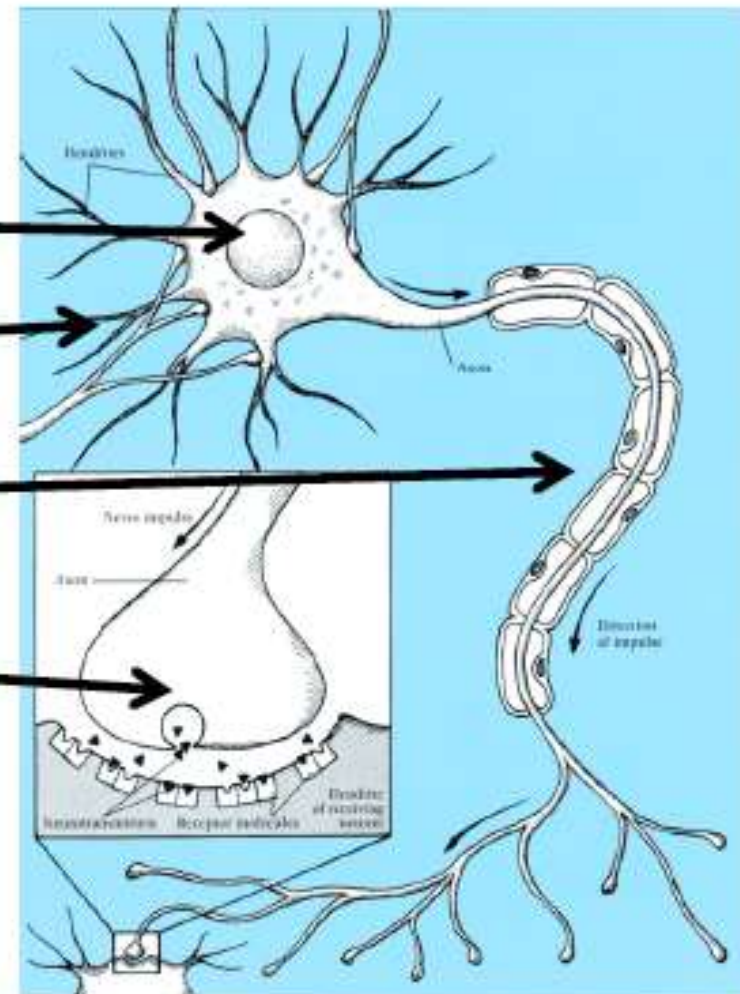
Estrutura de um Neurônio:

Corpo celular

Dendritos

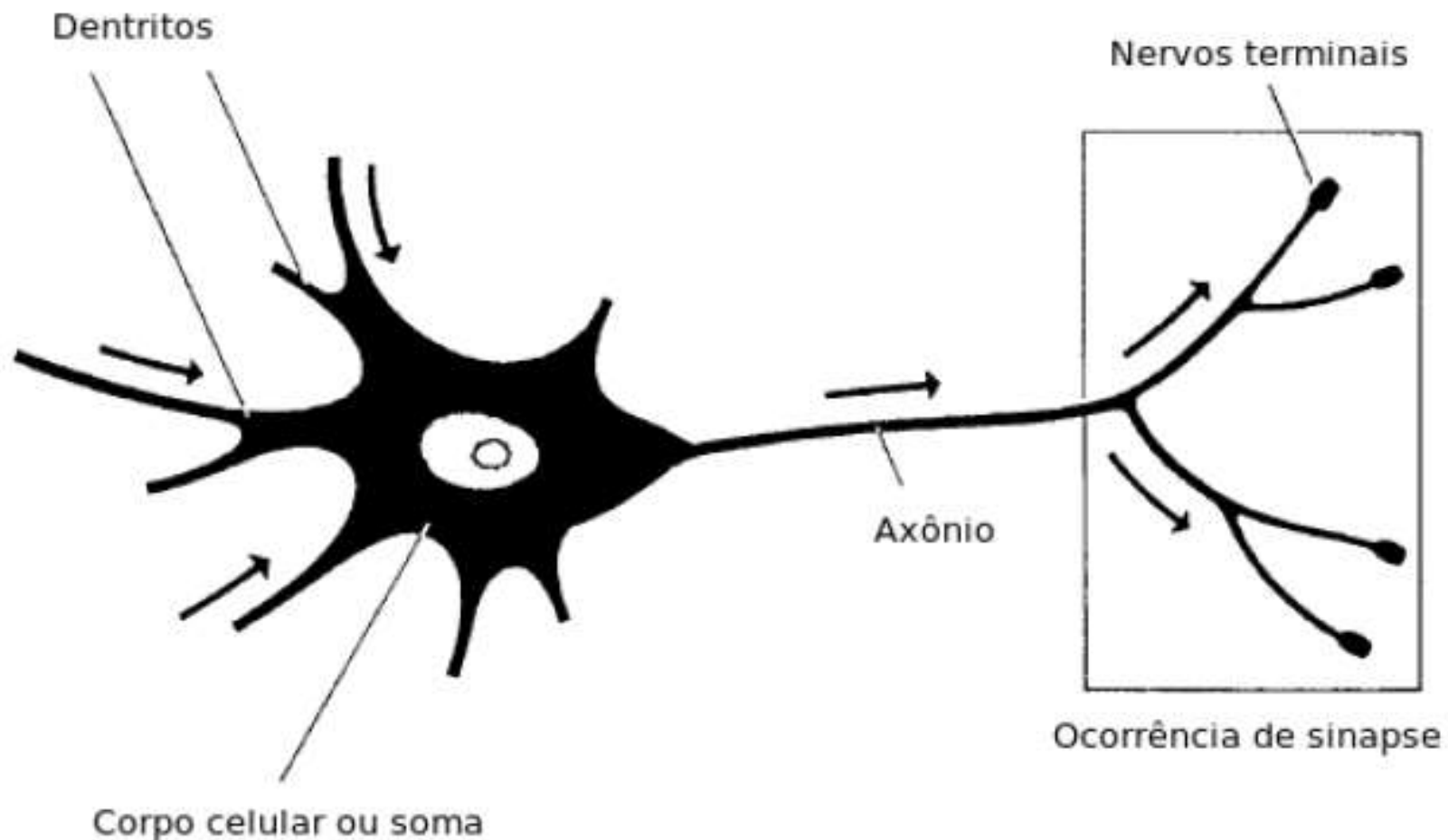
Axônio

Terminais sinápticos



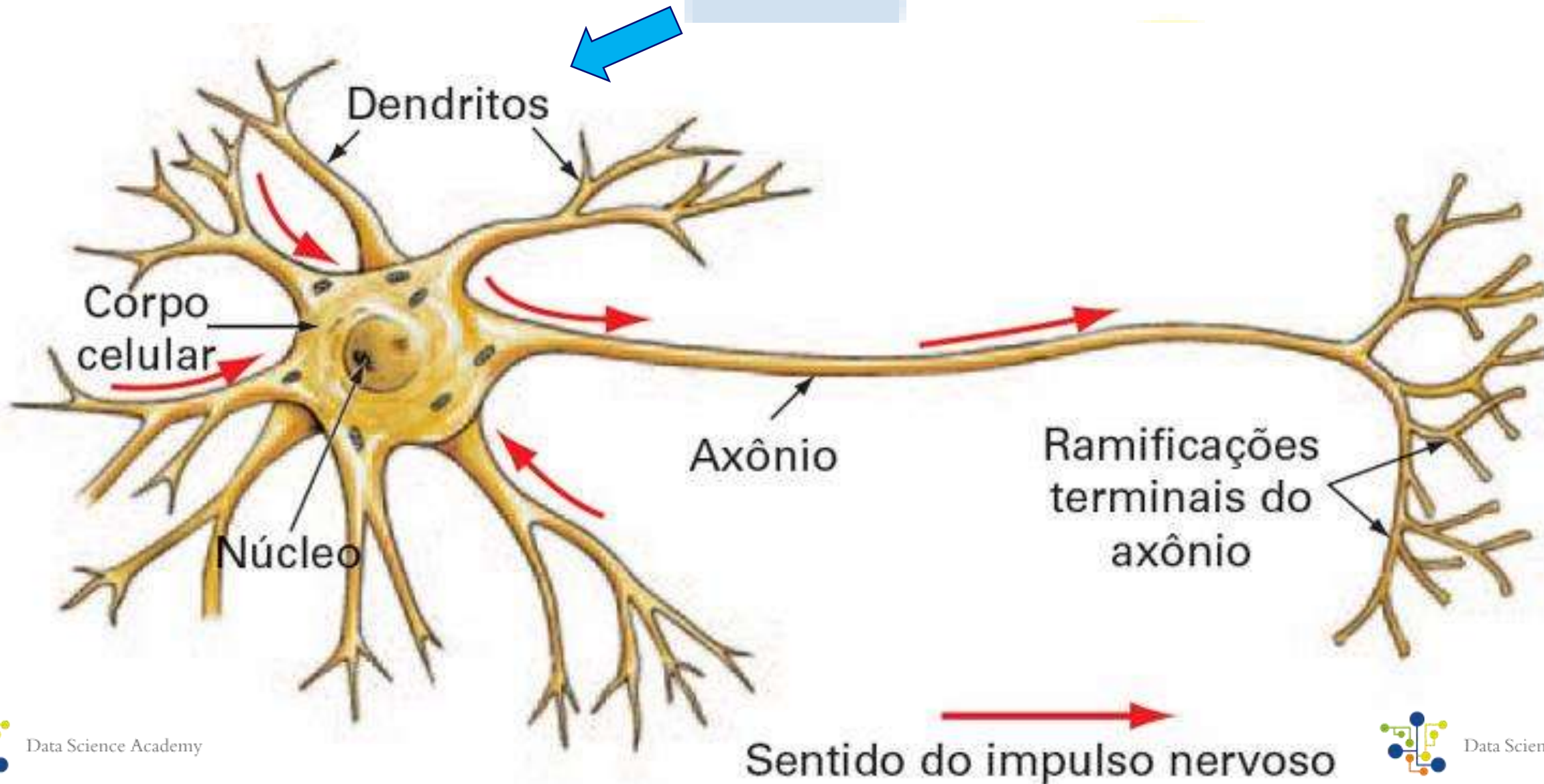


Redes Neurais Artificiais



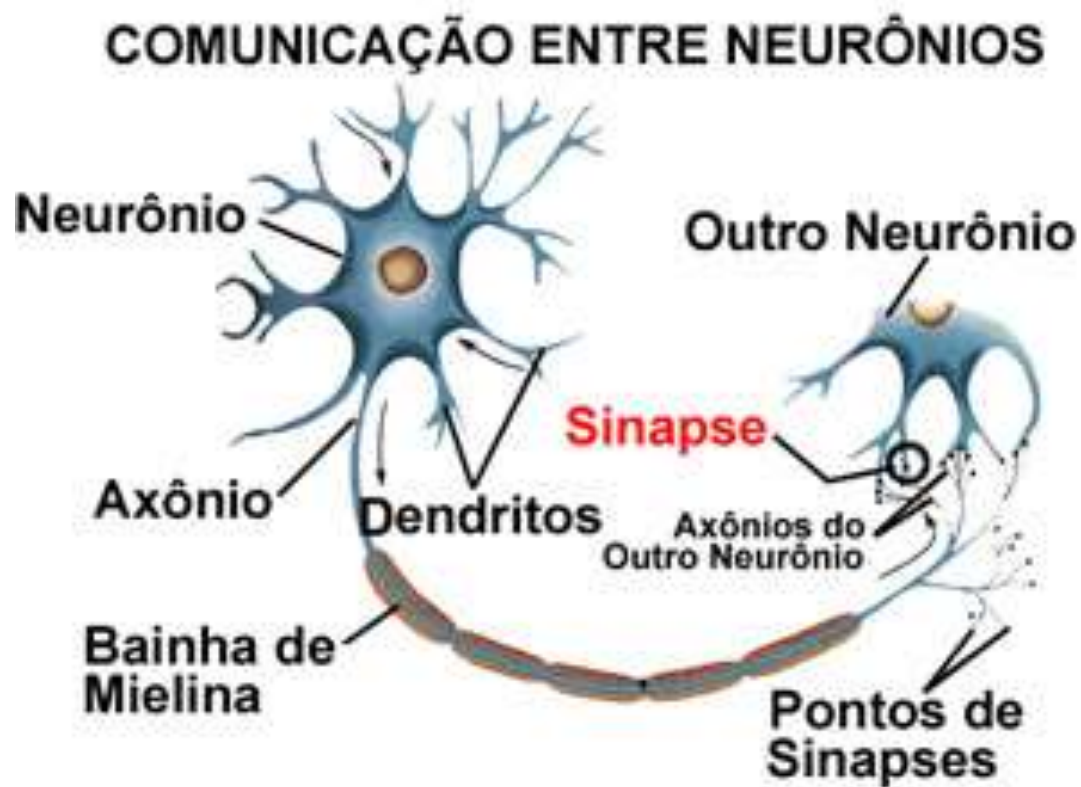


Redes Neurais Artificiais

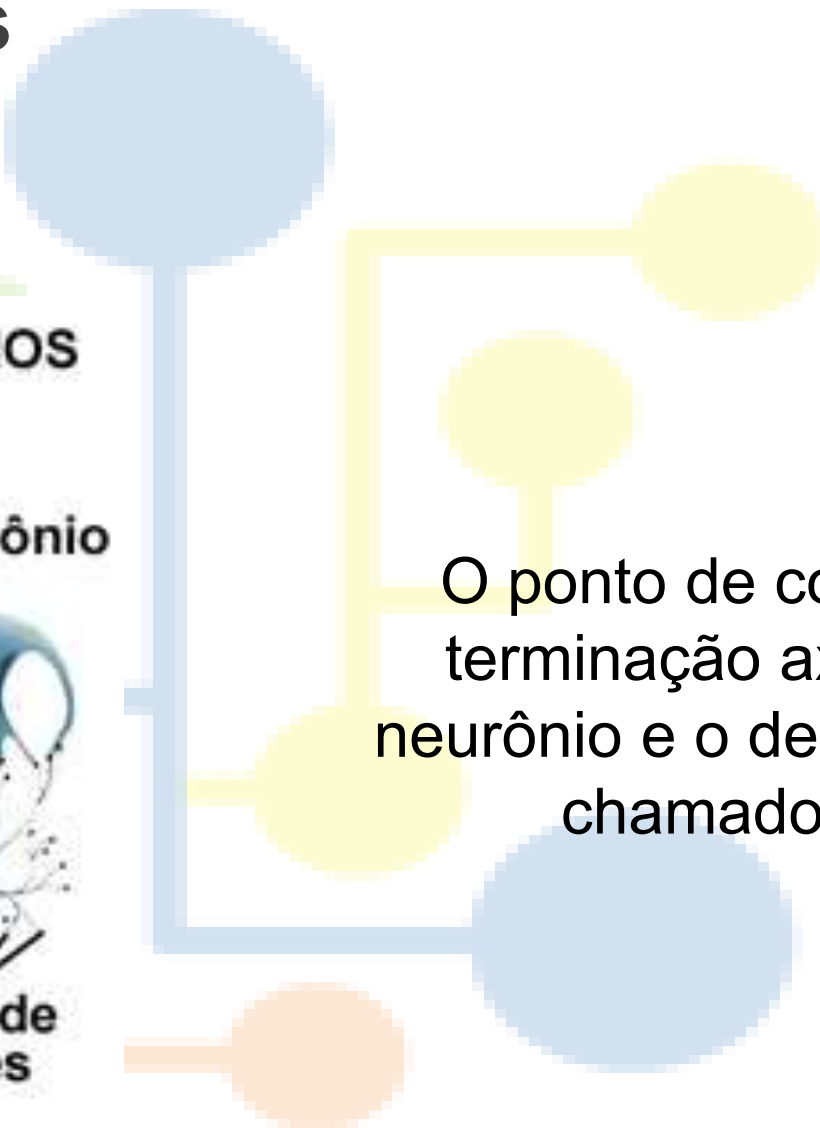




Redes Neurais Artificiais

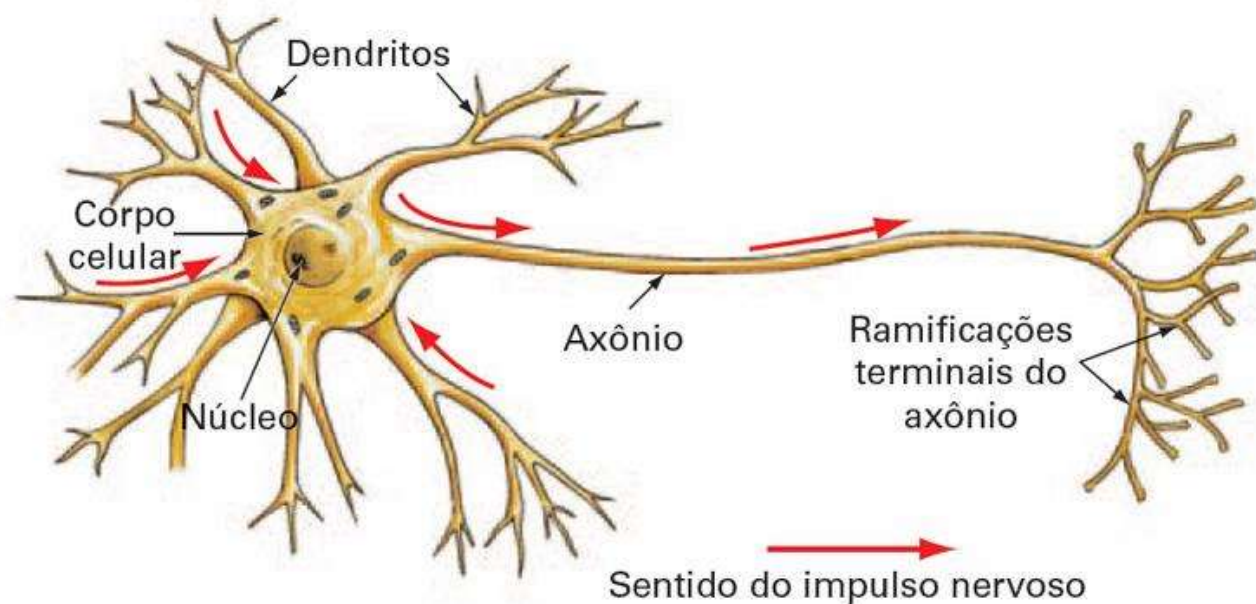


O ponto de contato entre a terminação axônica de um neurônio e o dendrito de outro é chamado **sinapse**





Redes Neurais Artificiais



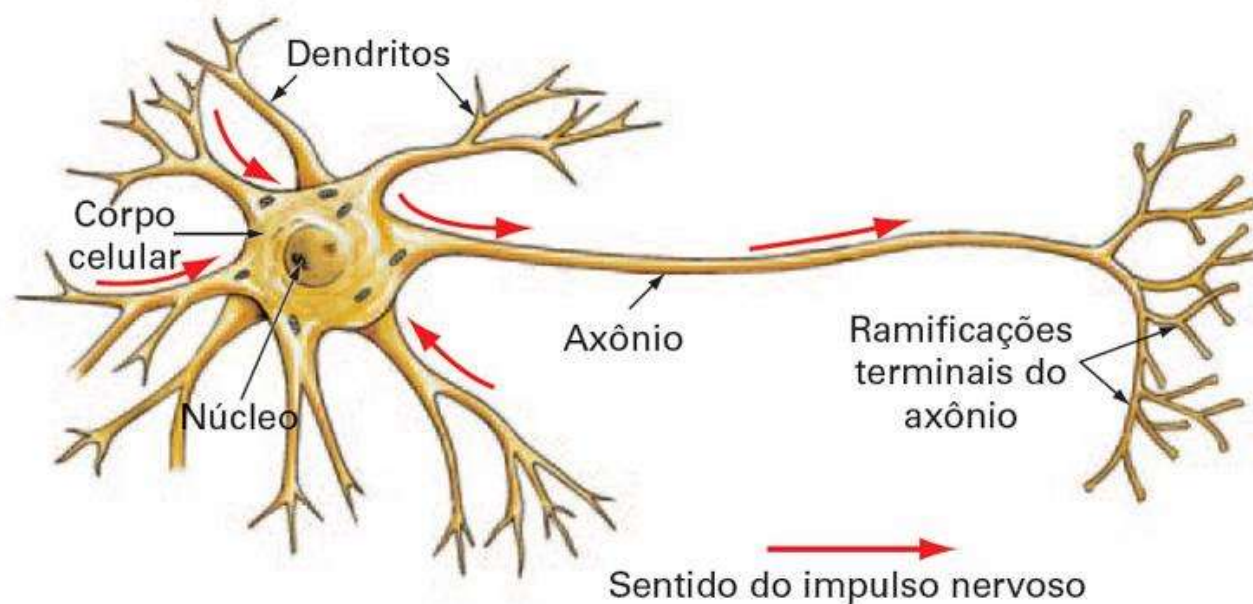
Se esses sinais forem superiores a aproximadamente 50mV (limiar do disparo), seguem pelo axônio. Caso contrário, são bloqueados e não prosseguem (são considerados irrelevantes).

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





Redes Neurais Artificiais



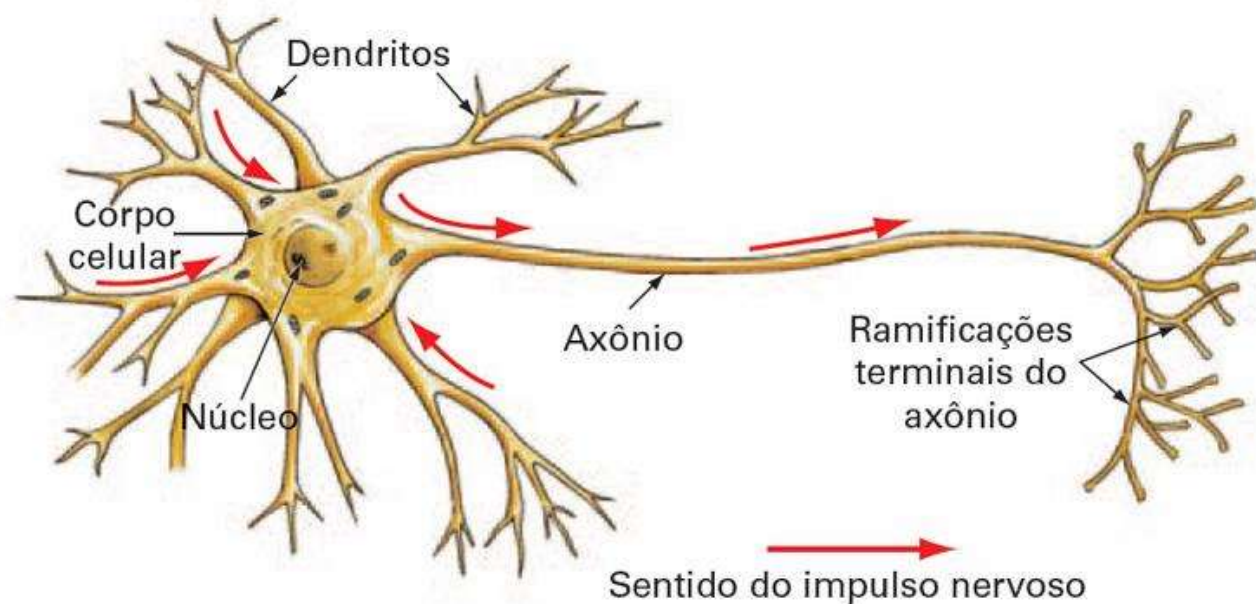
Um neurônio recebe sinais através de inúmeros dendritos, os quais são **ponderados** e enviados para o axônio, podendo ou não seguir adiante (**threshold**)

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...

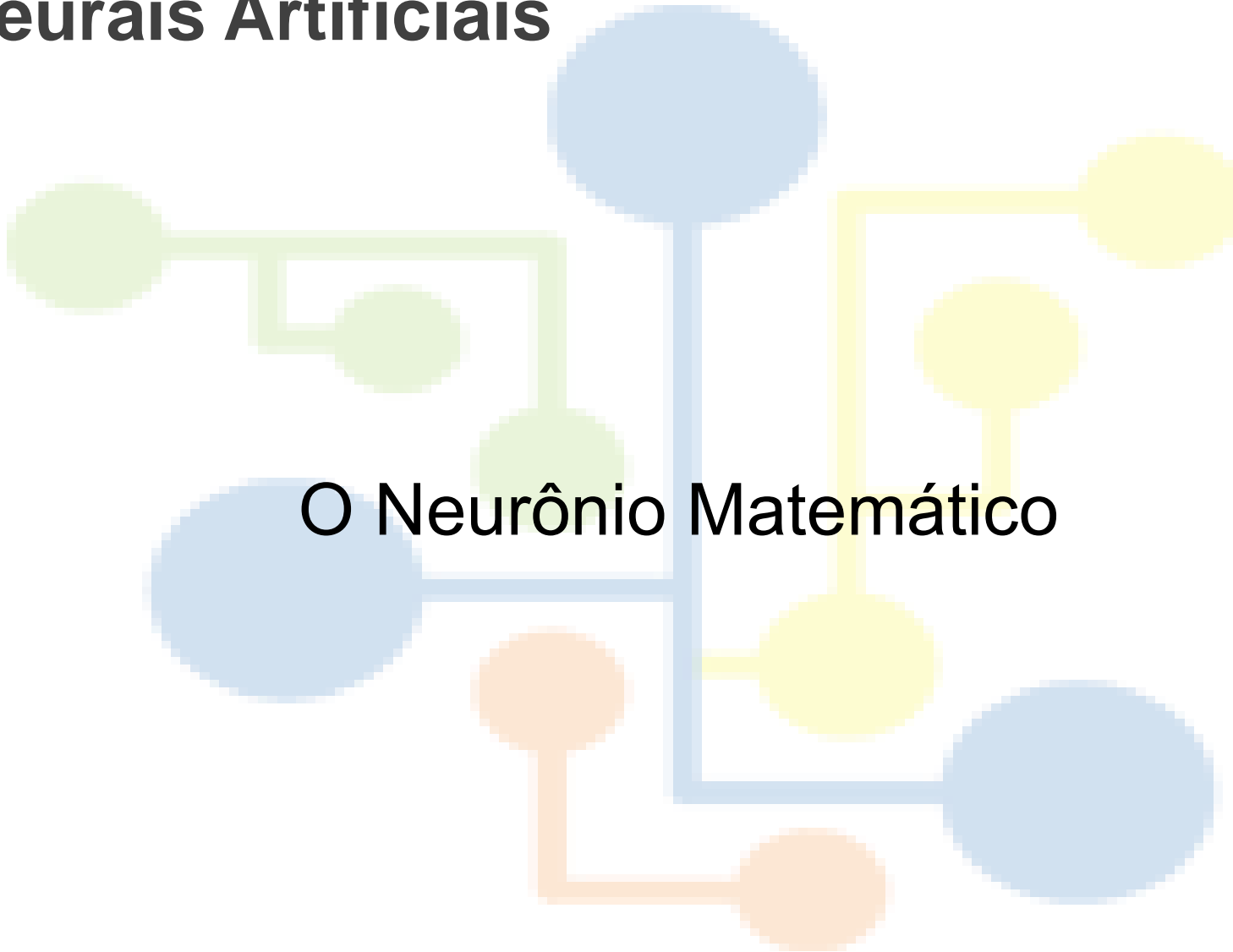
Os valores dos pesos são estabelecidos por meio de treinamento recebido pelo cérebro durante a vida útil. É a memorização.





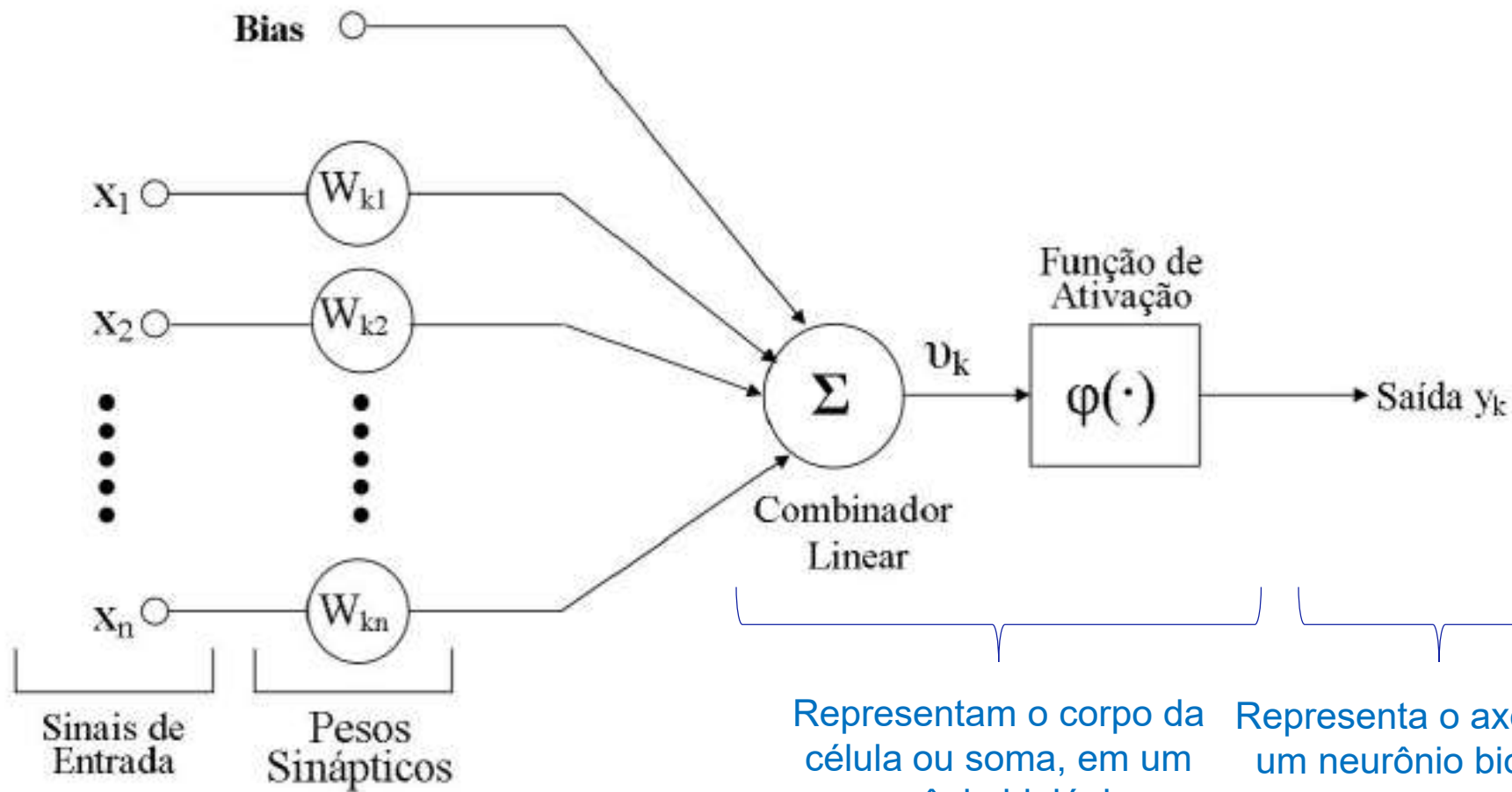
Redes Neurais Artificiais

O Neurônio Matemático





Redes Neurais Artificiais



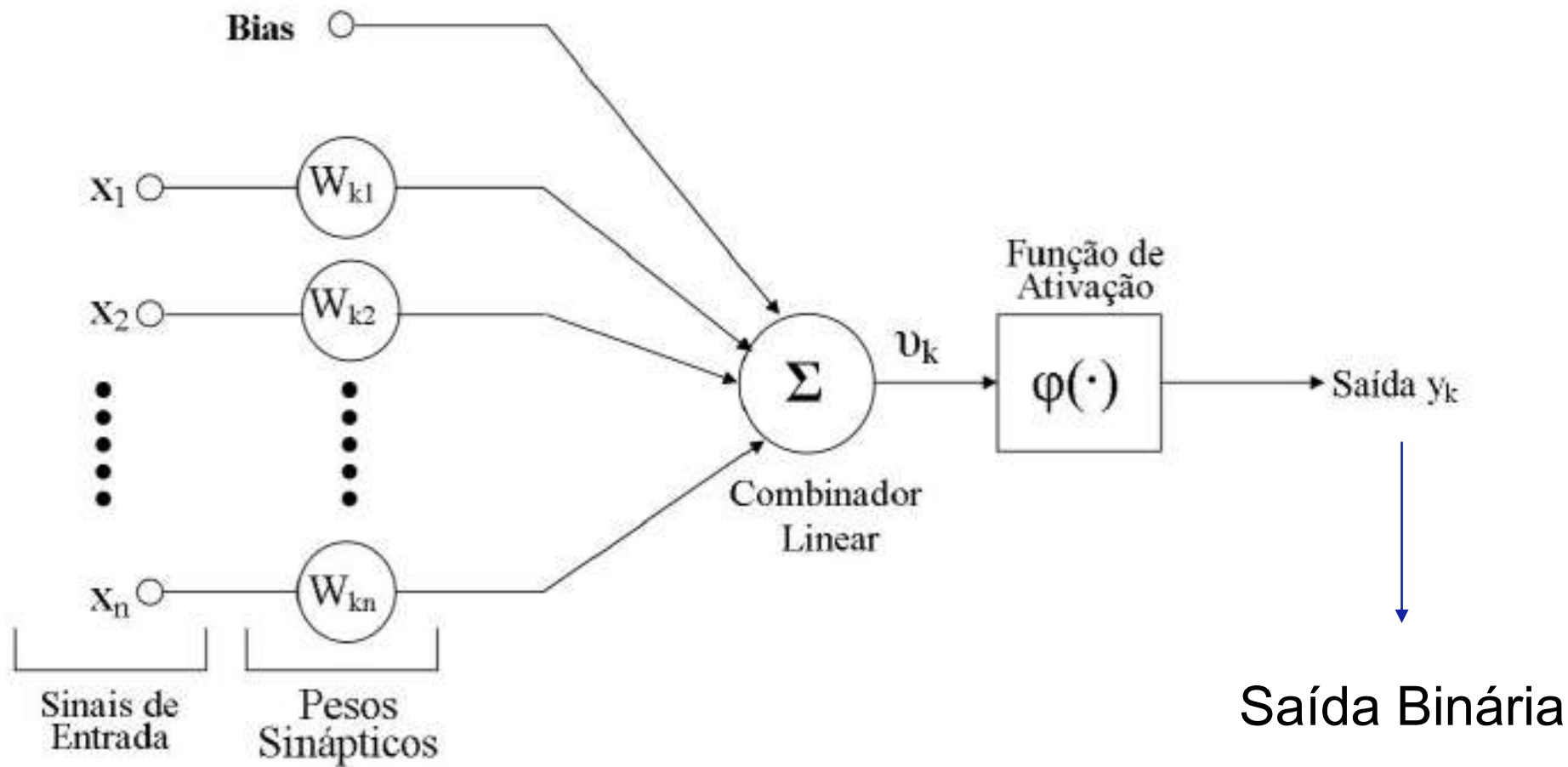
Representam o corpo da
célula ou soma, em um
neurônio biológico

Representa o axônio em
um neurônio biológico



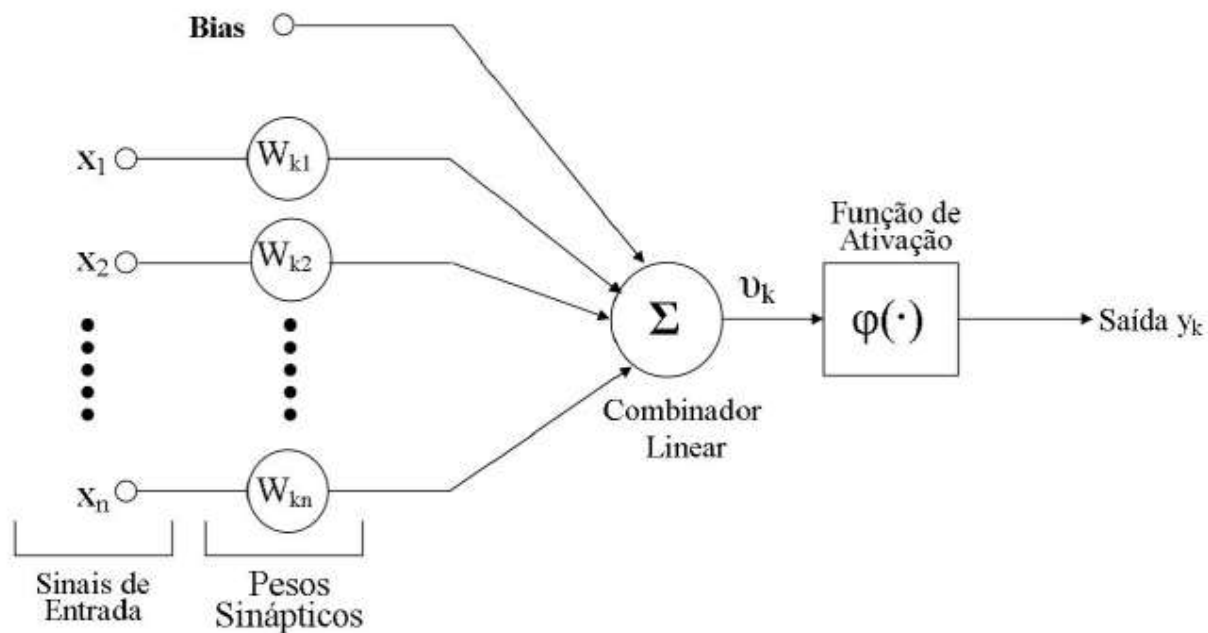


Redes Neurais Artificiais





Redes Neurais Artificiais



$$X_1W_1, X_2W_2, \dots, X_nW_n$$

$$v = \sum_{i=0}^n w_i x_i$$

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

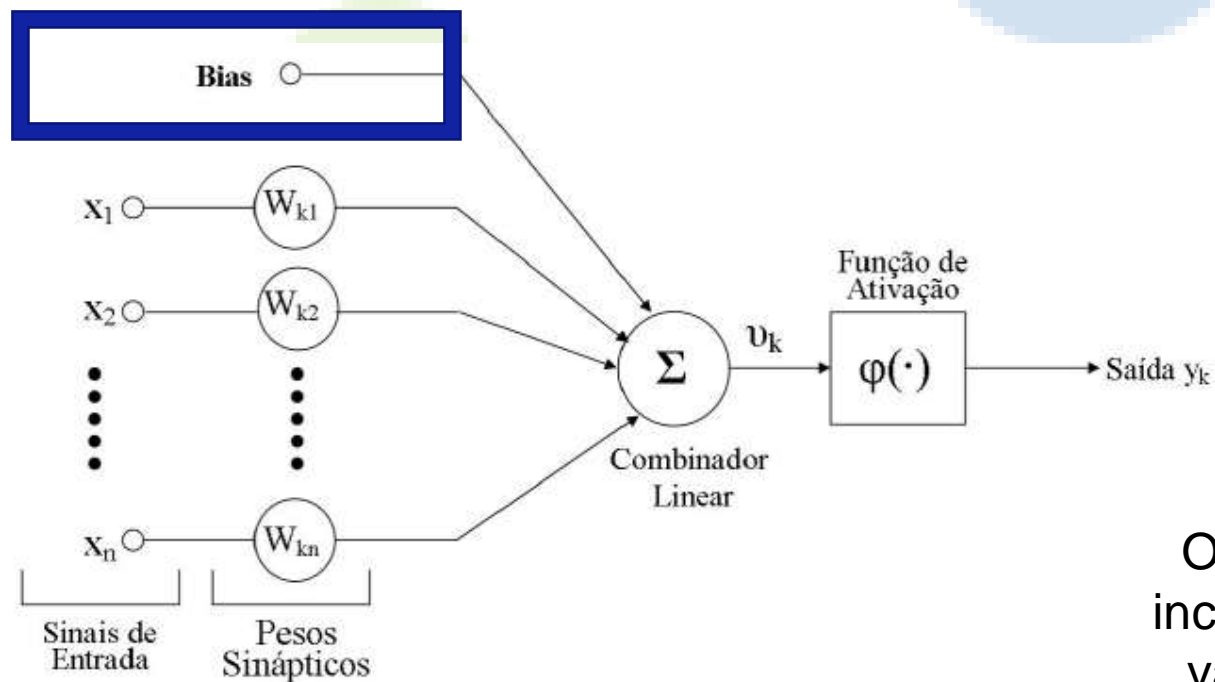
Dentre as funções de ativação utilizadas, podemos destacar:

	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, conseqüentemente, 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.

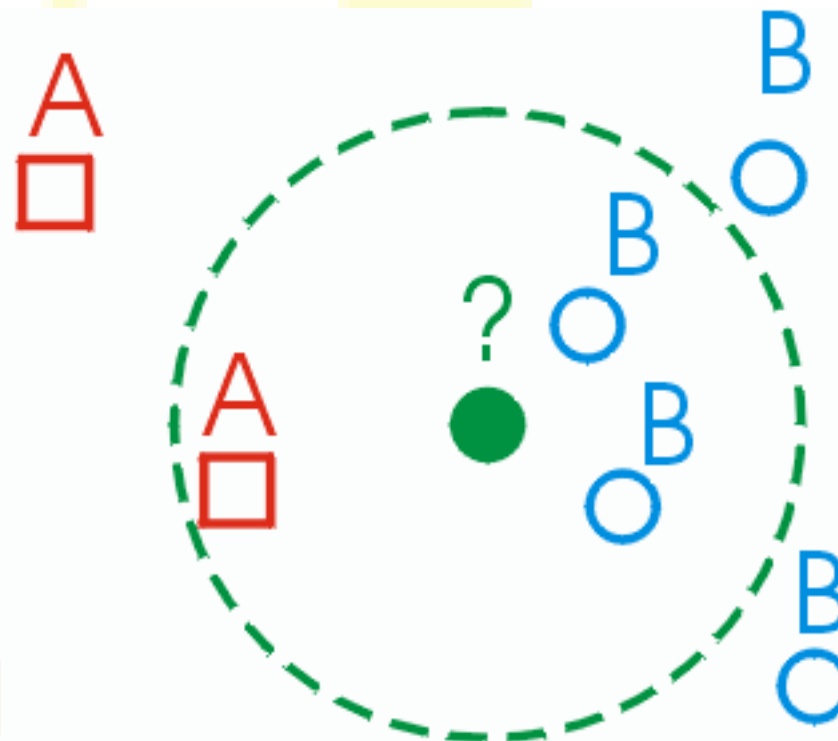




Modelos Não Paramétricos

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

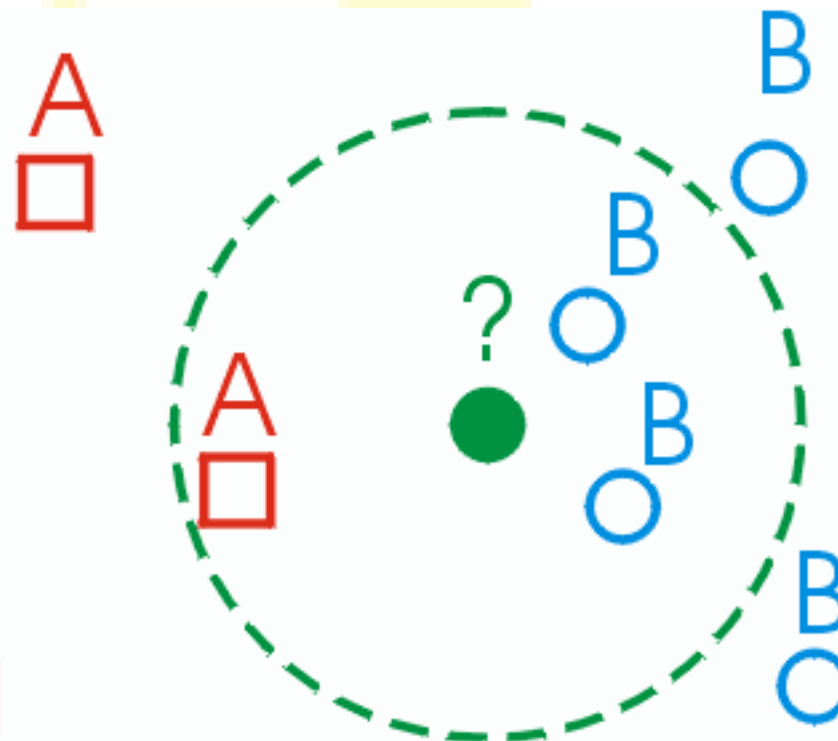
Isso é chamado de pesquisa de k -vizinhos mais próximos





Modelos Não Paramétricos

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





Modelos Não Paramétricos

Como funciona o KNN

$\{x_i, y_i\}$

Atributos (variáveis preditoras)

Classe (variável target)

Como classificar um novo ponto de dado X

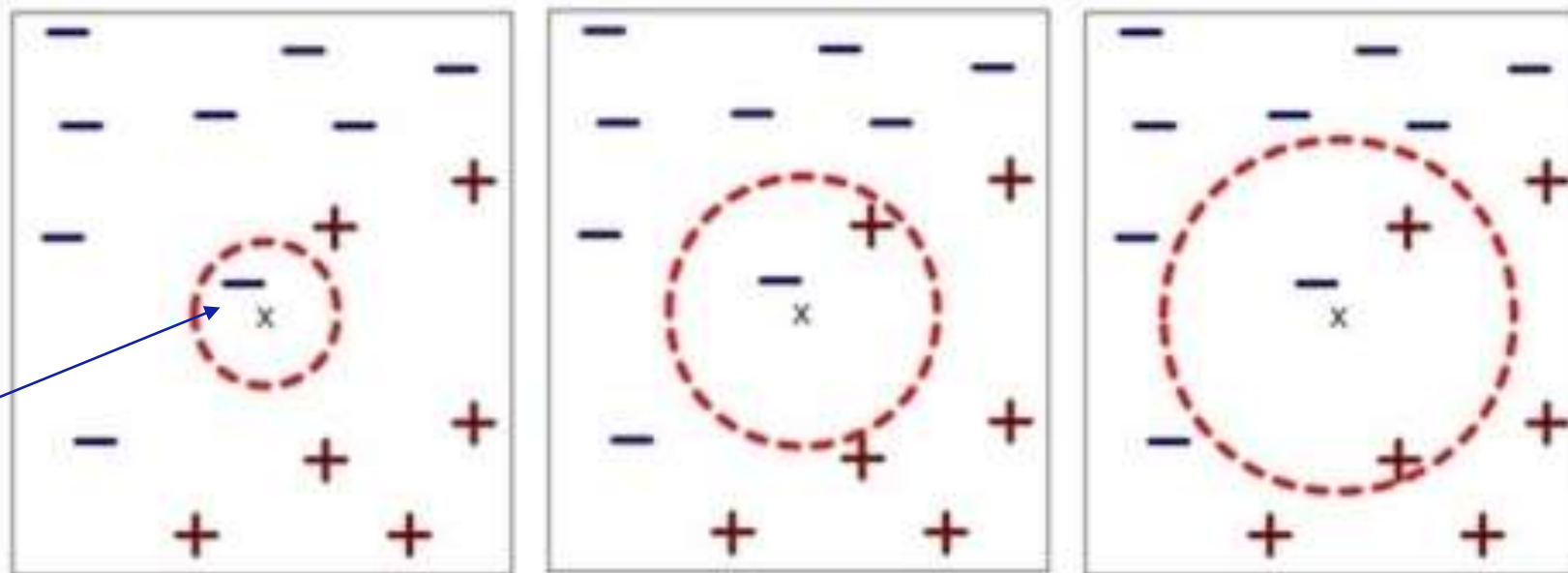
- 1- A distância é computada entre X e x_i para cada valor de x_i .
- 2- É escolhido o k-vizinho mais próximo x_{in} e sua respectiva classe.
- 3- Retorna-se o valor de y mais frequente na lista $y_{i1}, y_{i2}, \dots, y_{in}$.





Modelos Não Paramétricos

Classificação do
novo ponto de
dato



1NN

2NN

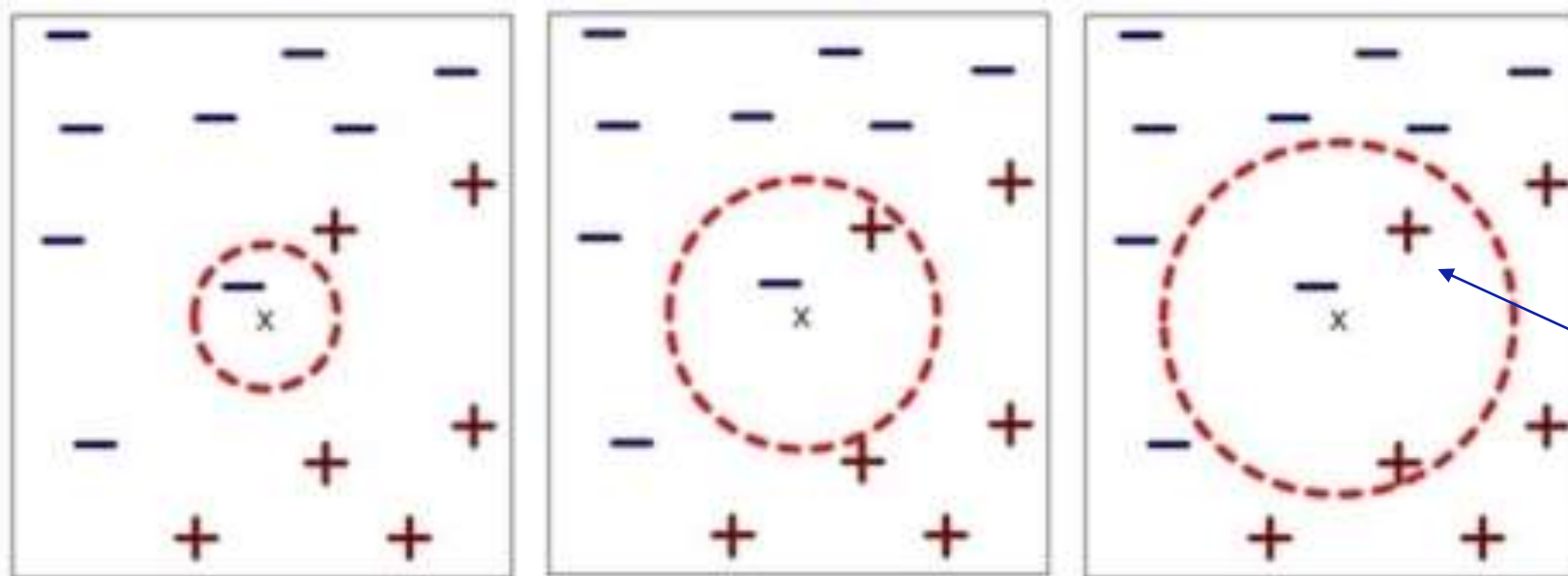
3NN

Valor de K





Modelos Não Paramétricos



1NN

2NN

3NN

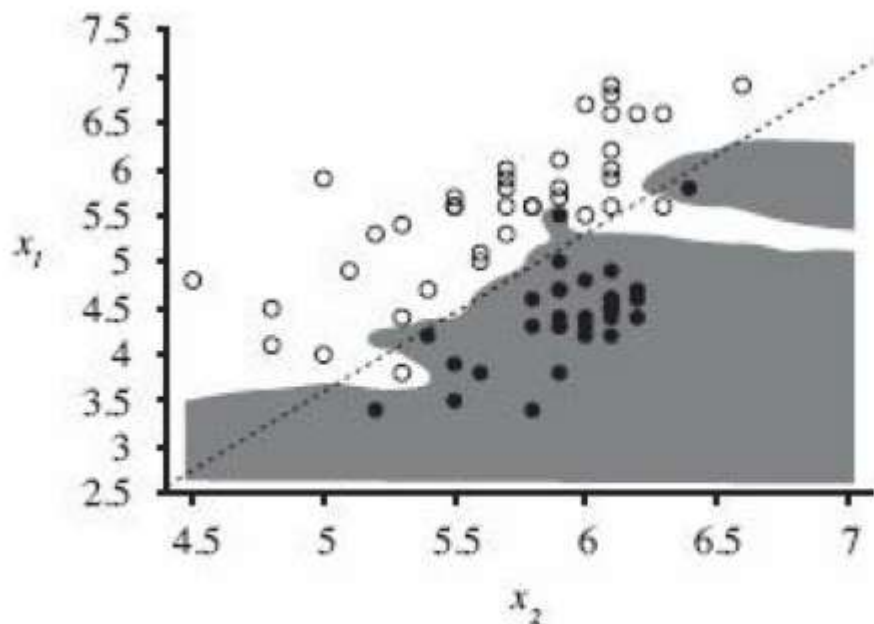
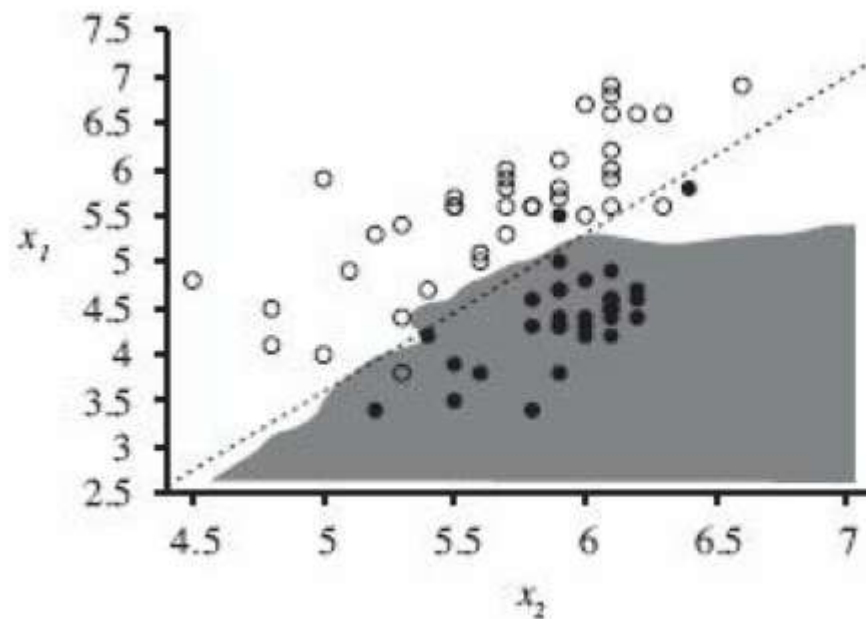
Classificação do
novo ponto de
dado

Valor de K





Modelos Não Paramétricos

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

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





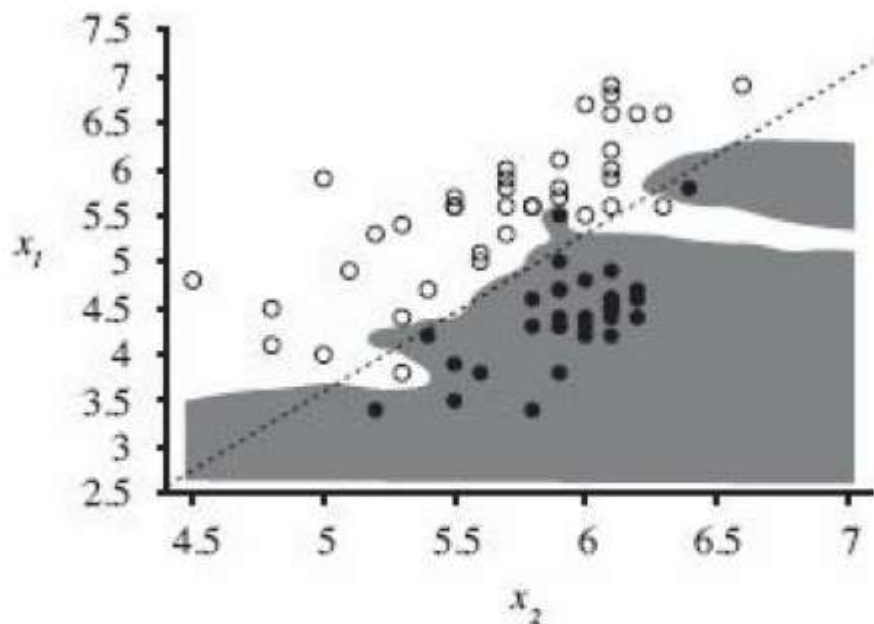
Modelos Não Paramétricos

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.

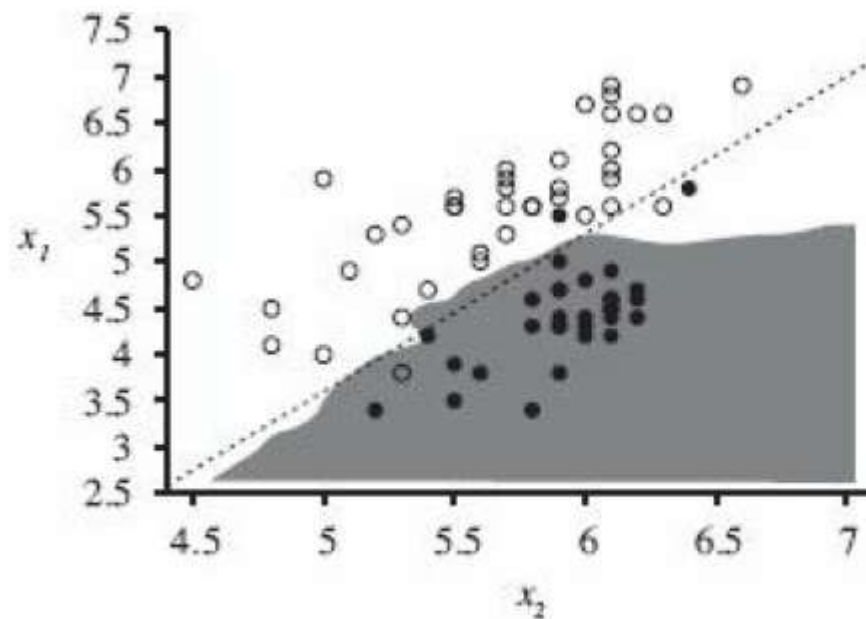




Modelos Não Paramétricos



(a) ($k = 1$)



(b) ($k = 5$)

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



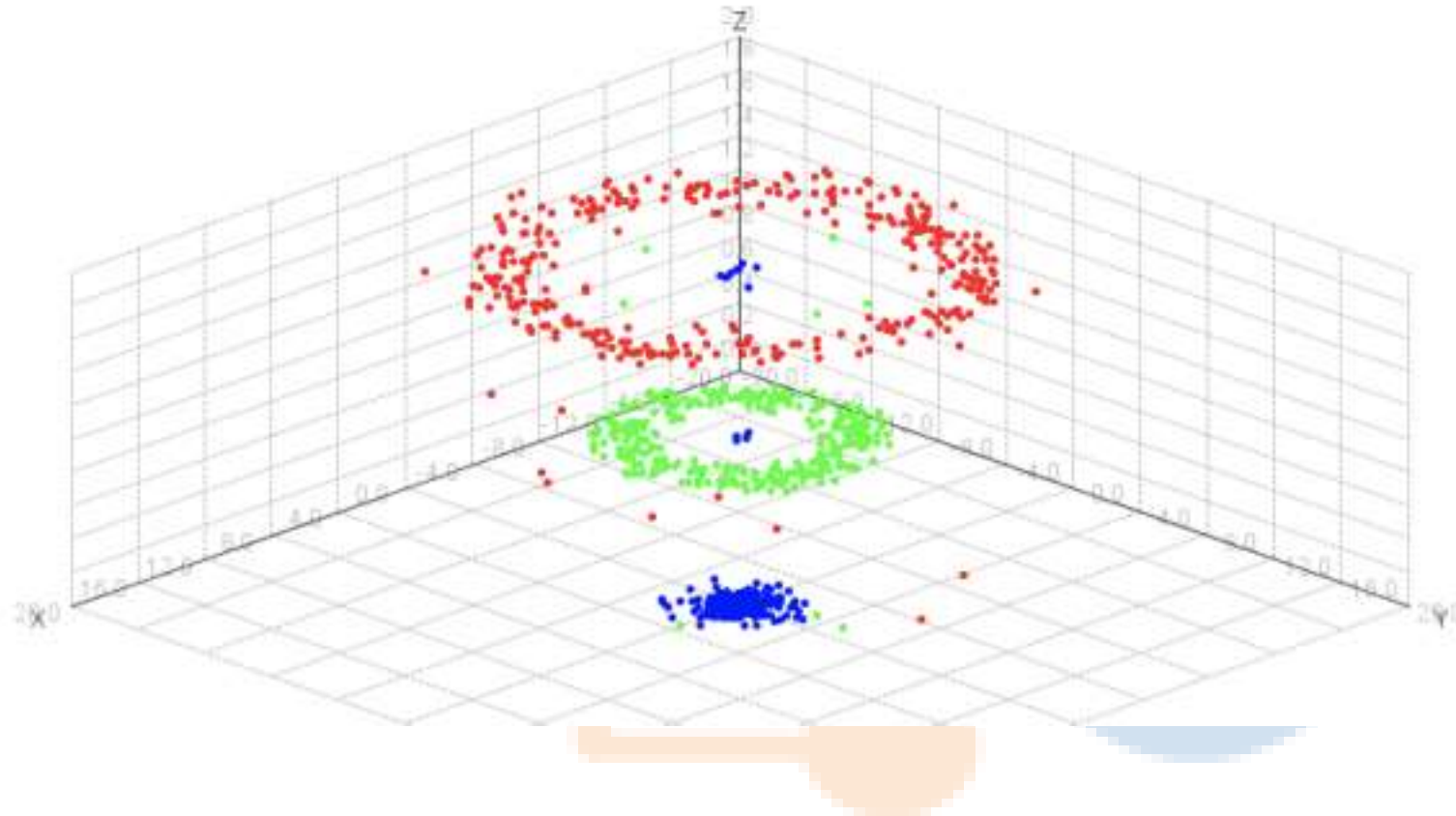


Máquinas de Vetores de Suporte





Máquinas de Vetores de Suporte





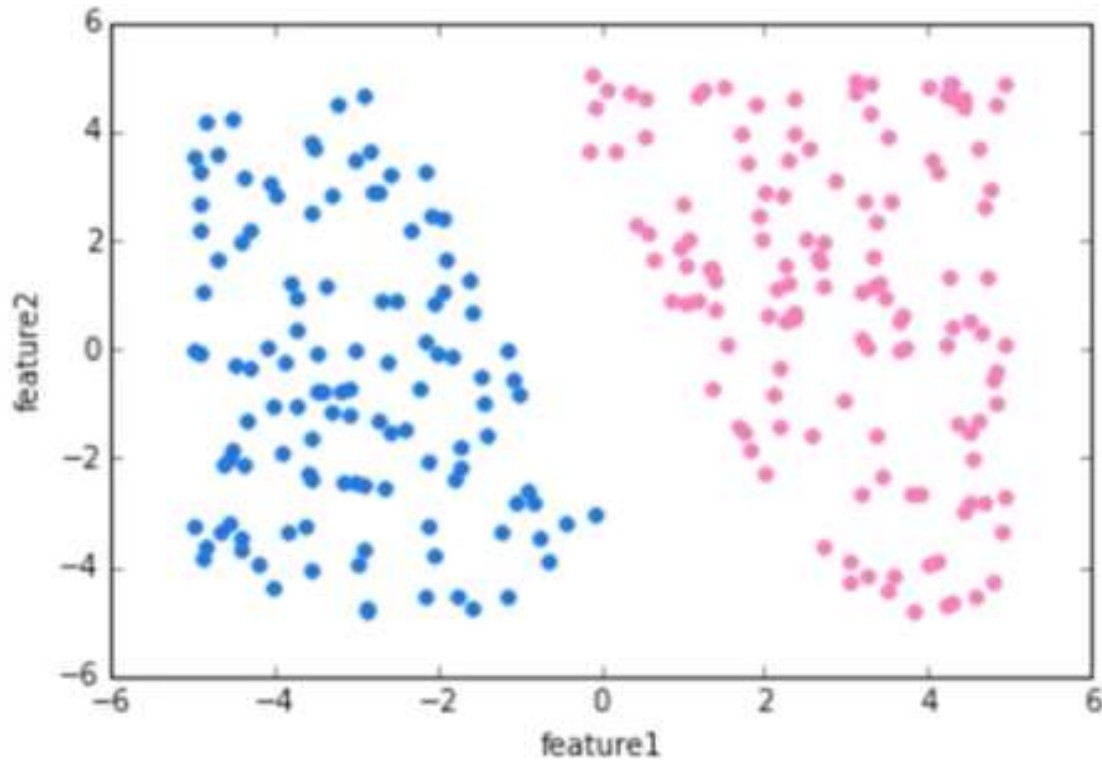
Máquinas de Vetores de Suporte

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

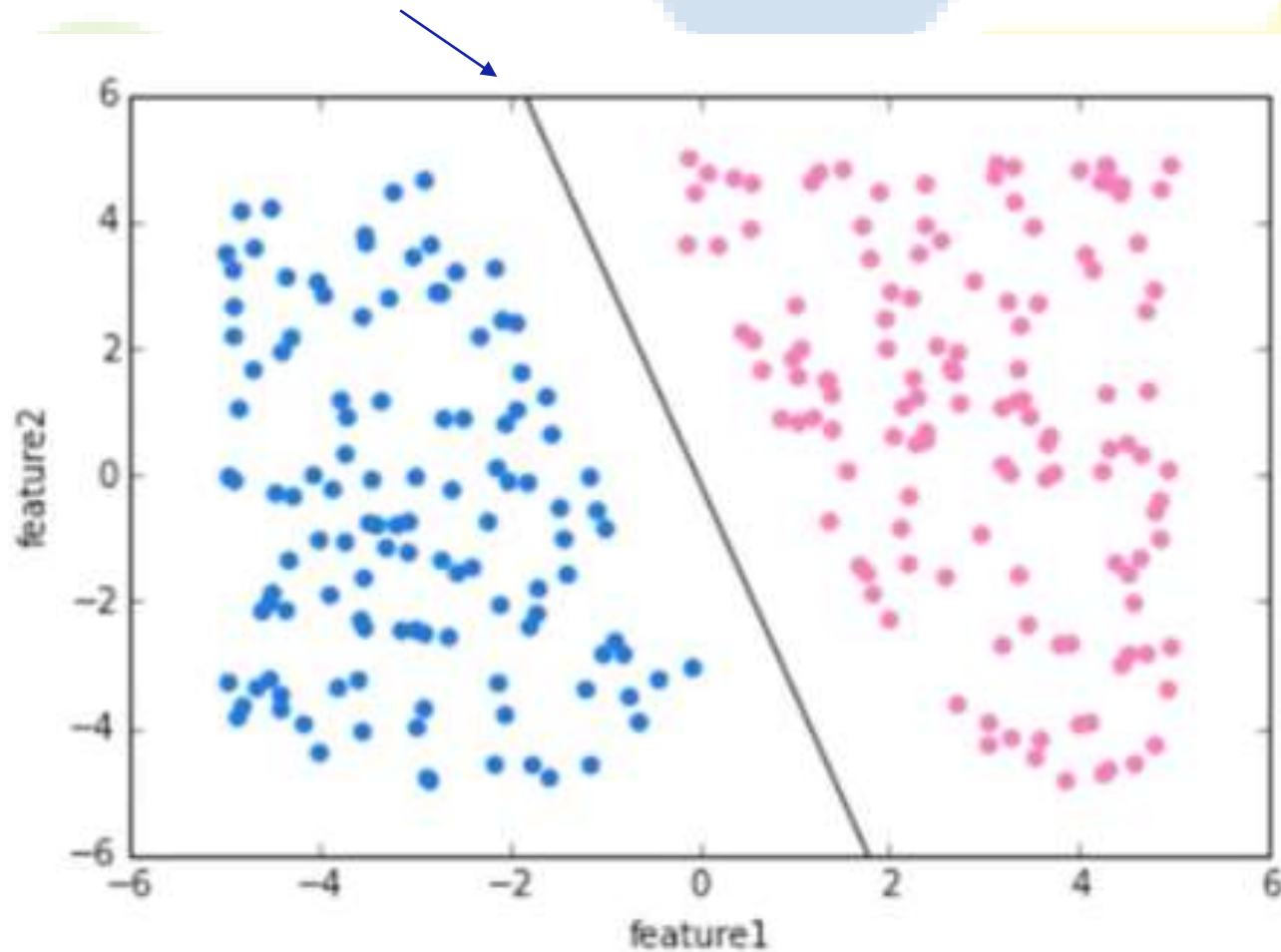


O algoritmo SVM cria um classificador binário linear não-probabilístico



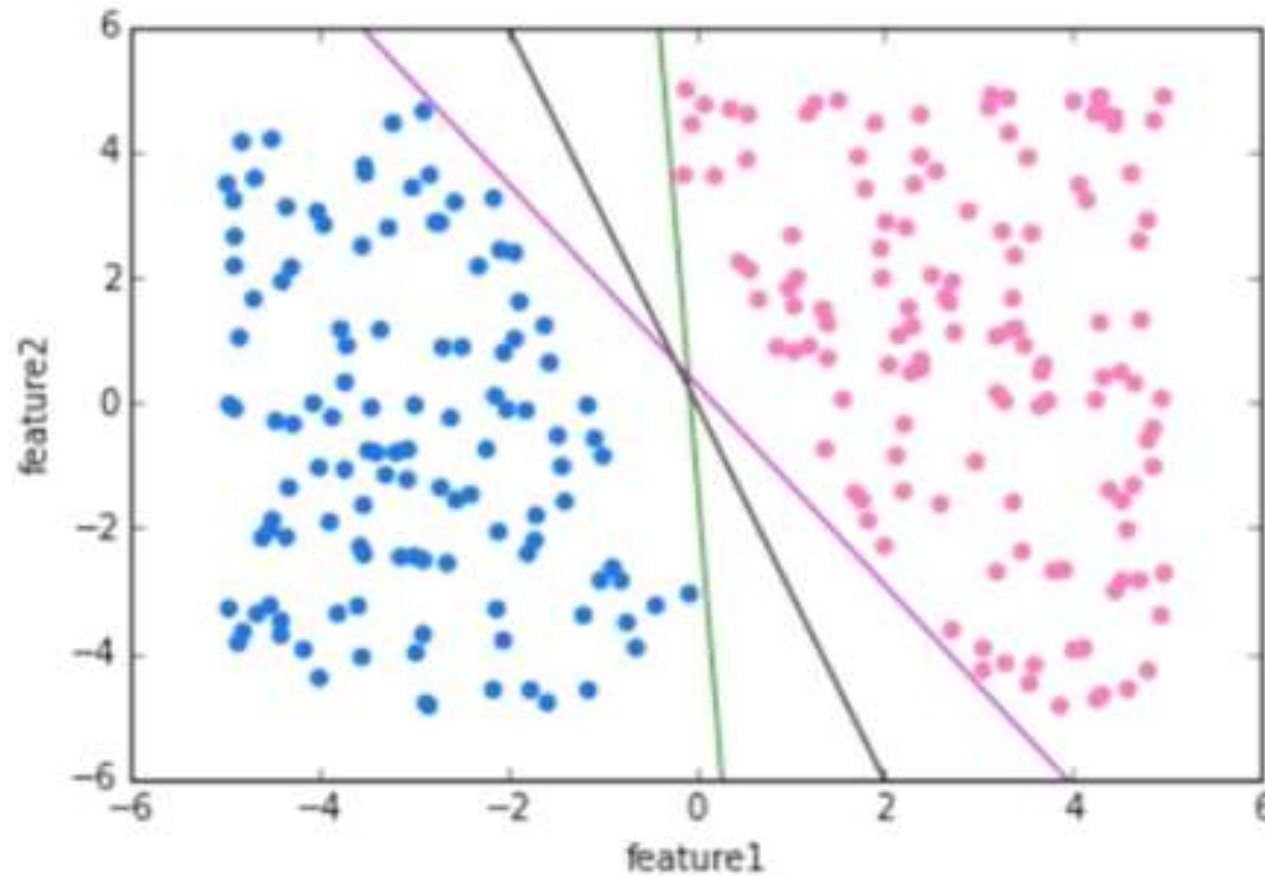
Máquinas de Vetores de Suporte

Hiperplano





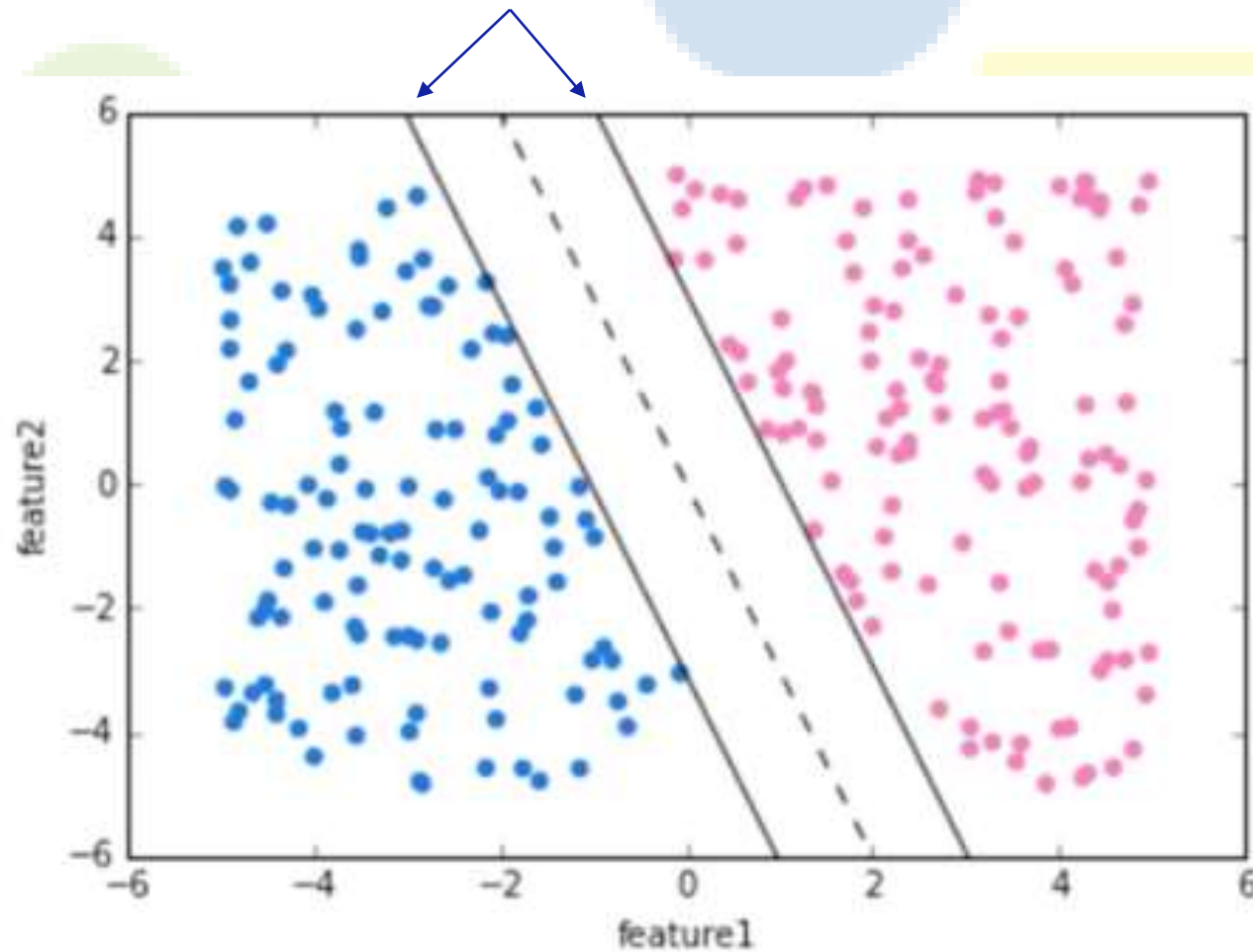
Máquinas de Vetores de Suporte





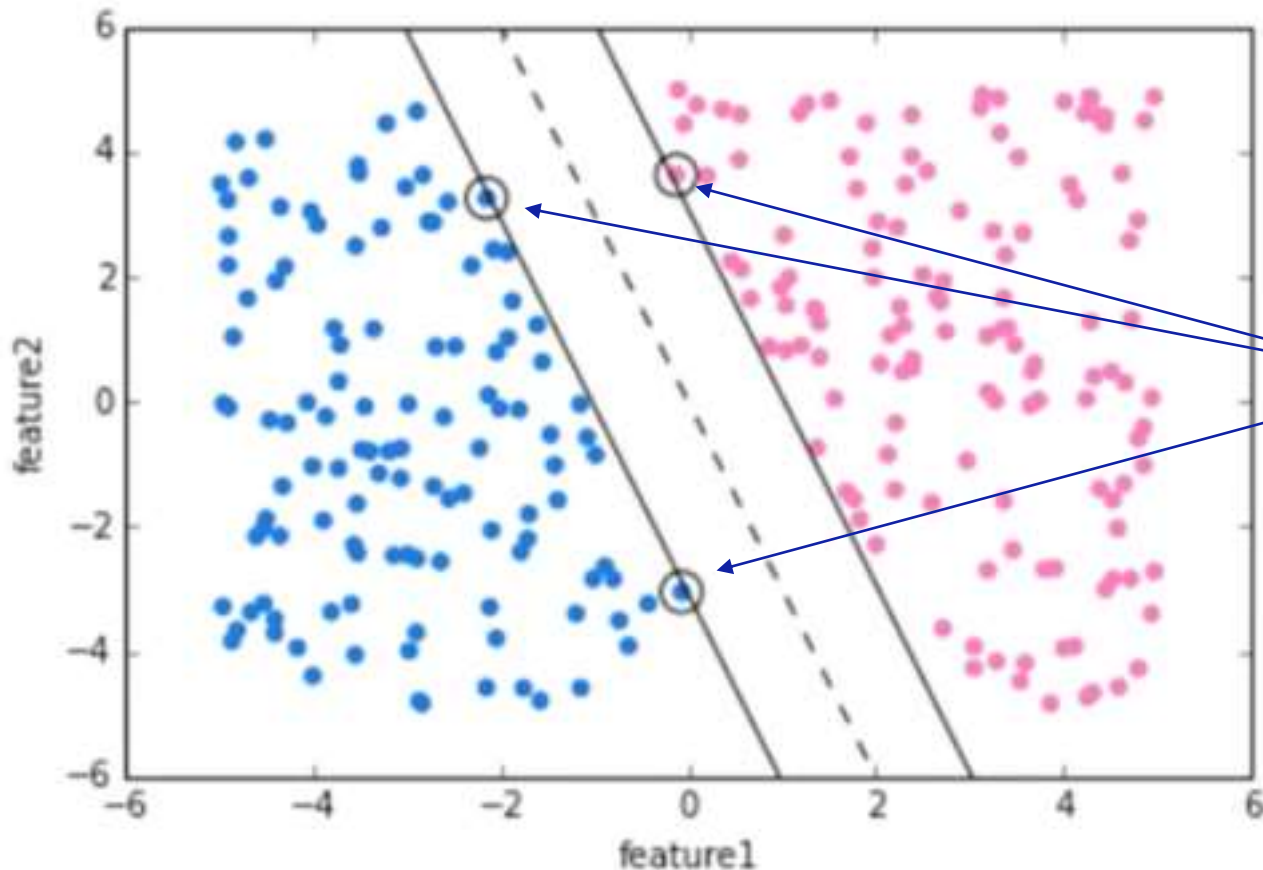
Máquinas de Vetores de Suporte

Margens





Máquinas de Vetores de Suporte

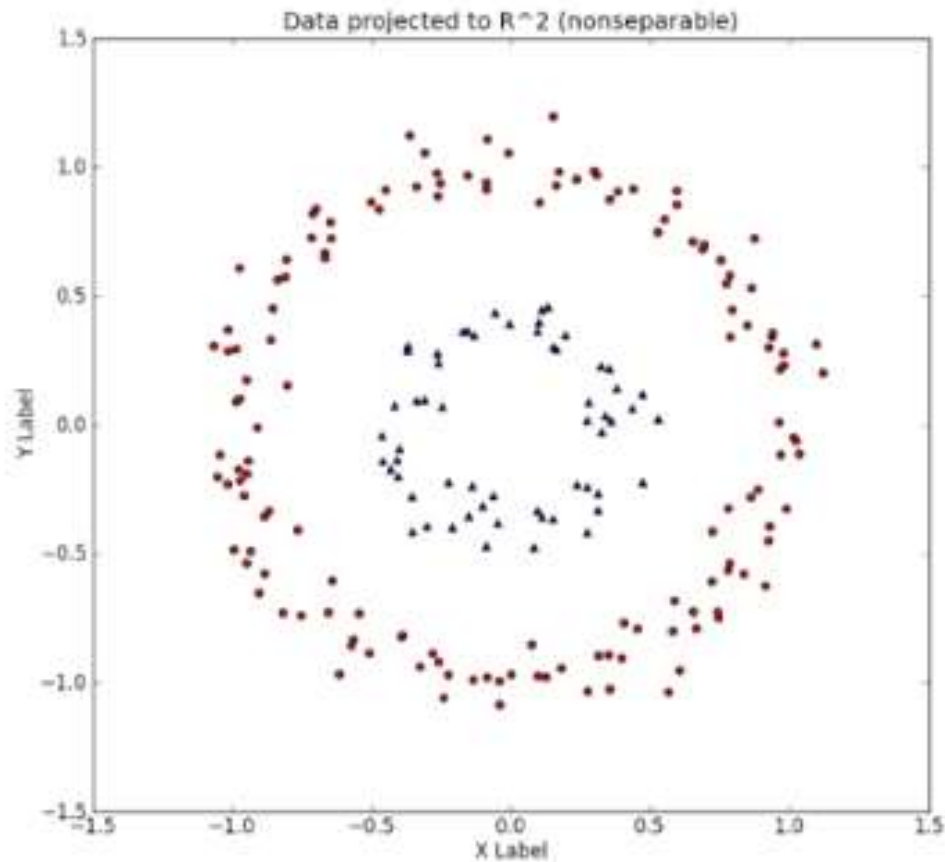


Vetores de
Suporte





Máquinas de Vetores de Suporte

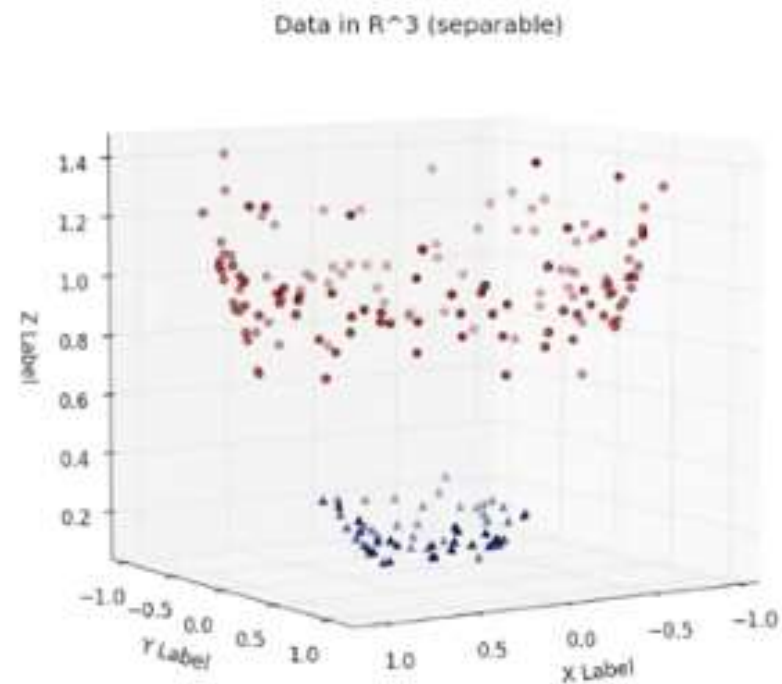
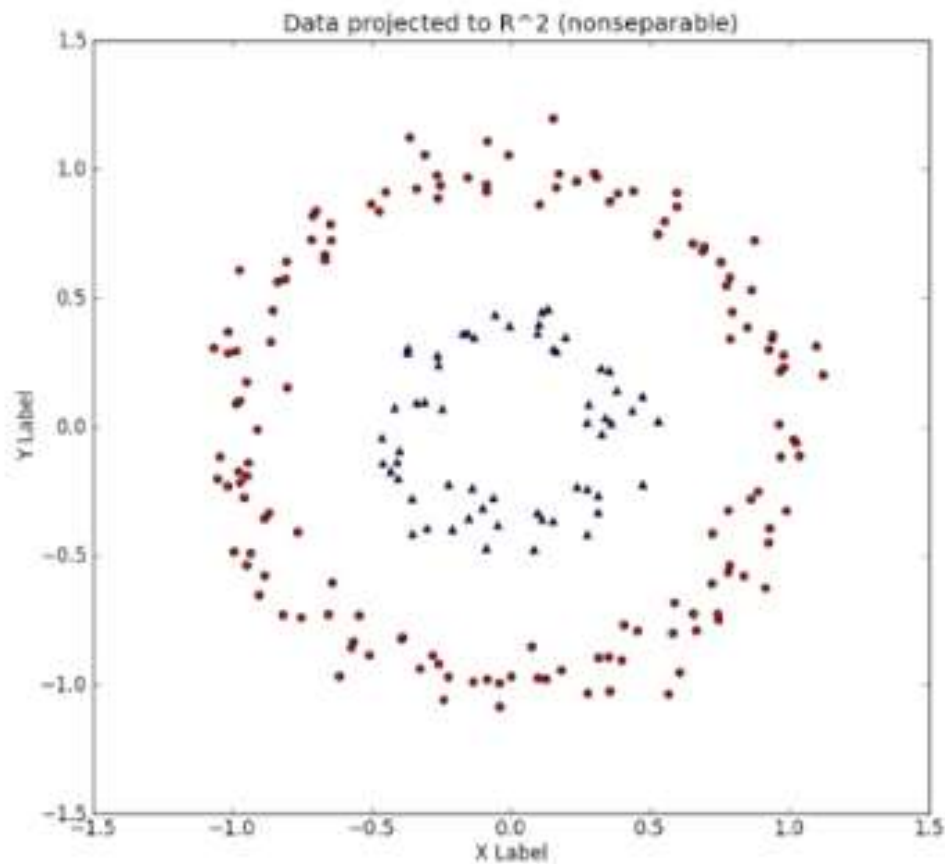


Mas e quando os dados são **não linearmente separáveis**?





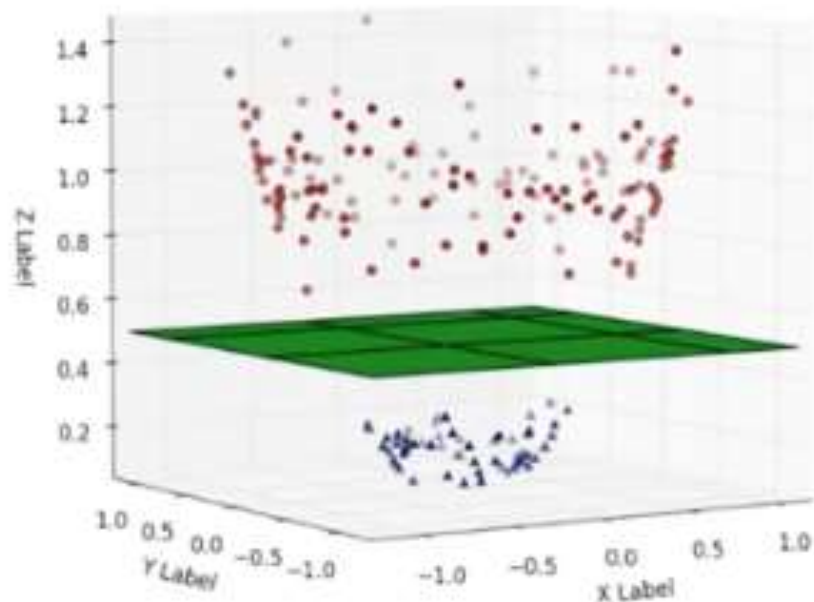
Máquinas de Vetores de Suporte



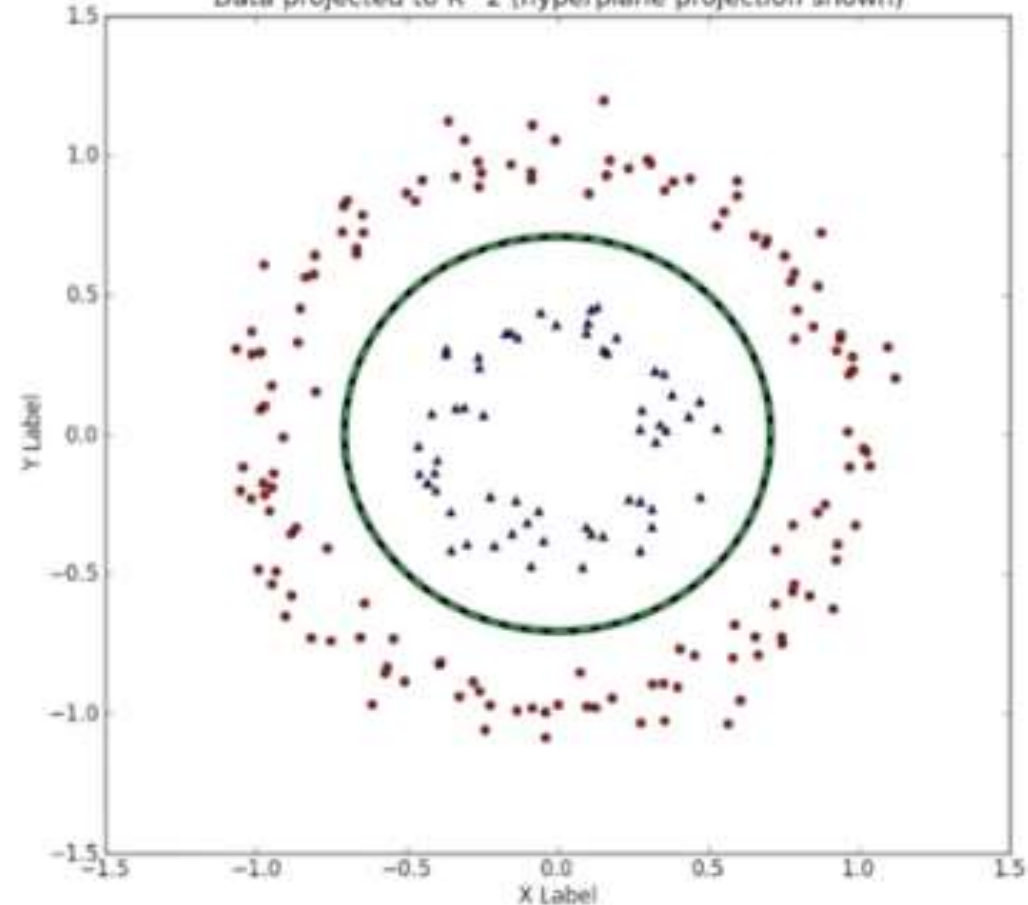


Máquinas de Vetores de Suporte

Data in R^3 (separable w/ hyperplane)

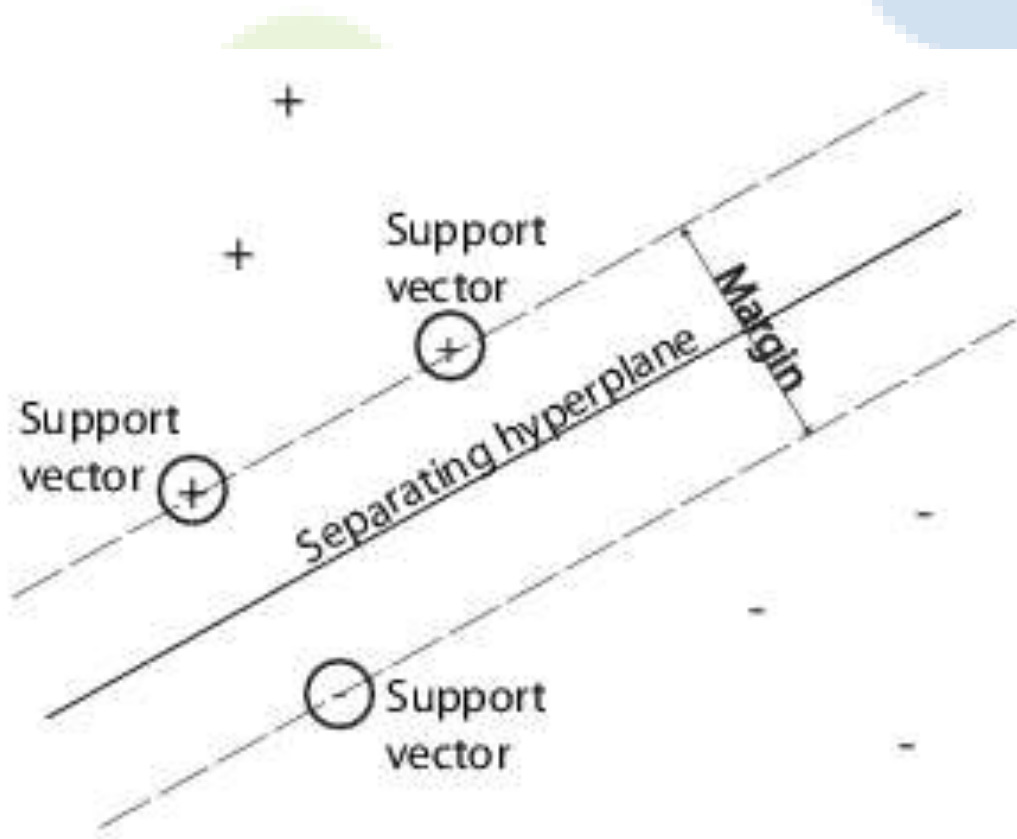


Data projected to R^2 (hyperplane projection shown)





Máquinas de Vetores de Suporte



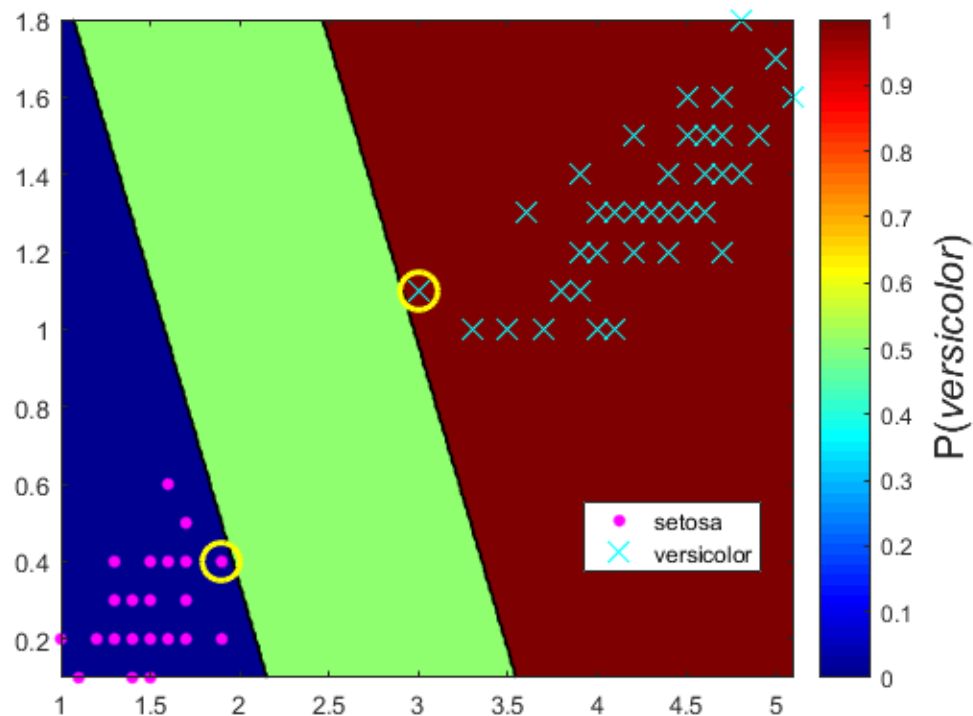
SVM's Lineares

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





Máquinas de Vetores de Suporte



SVM's Lineares

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

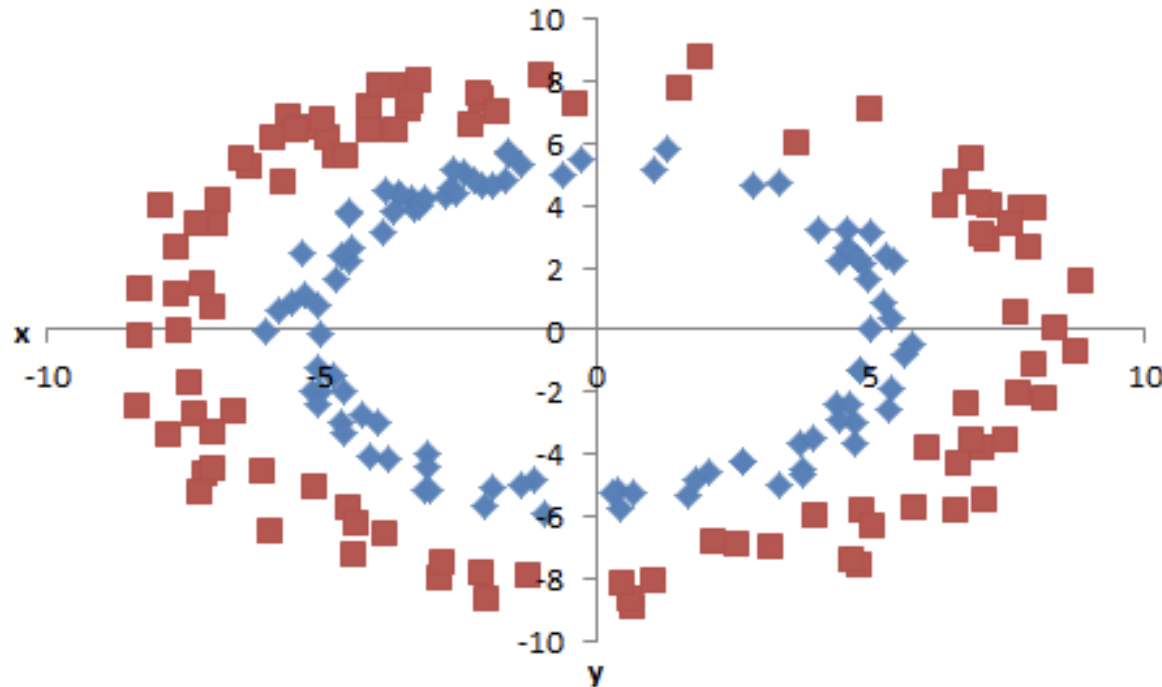




Máquinas de Vetores de Suporte

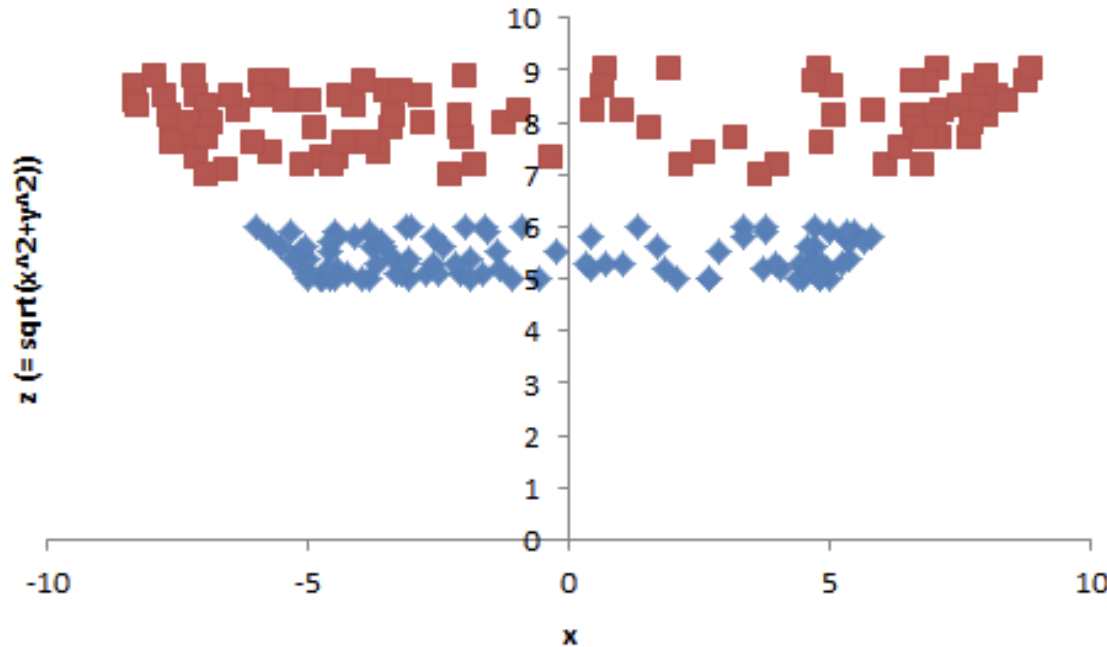
SVM's Não Lineares

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 $z = \sqrt{x^2 + y^2}$

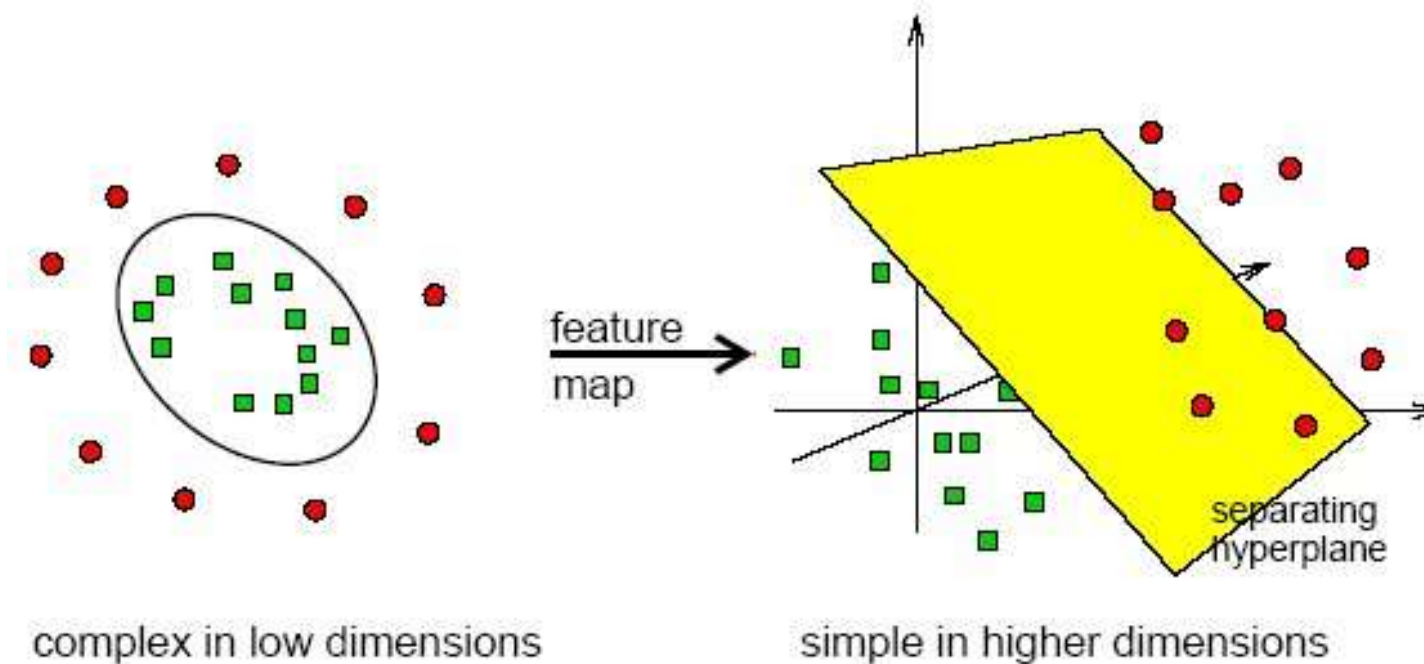




Máquinas de Vetores de Suporte

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!





Aprendizagem Por Agrupamento





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

Construção de Ensembles

Construir Vários Modelos

Combinar suas Estimativas





Aprendizagem Por Agrupamento

Métodos
Ensemble

Random Forest

Bagging

AdaBoost

Gradient Boosting





Aprendizagem Por Agrupamento

Métodos Ensemble

Bagging

Boosting

Voting





Aprendizagem Por Agrupamento

Estado da Arte em Machine Learning





Aprendizagem Por Agrupamento

Estado da Arte em Machine Learning

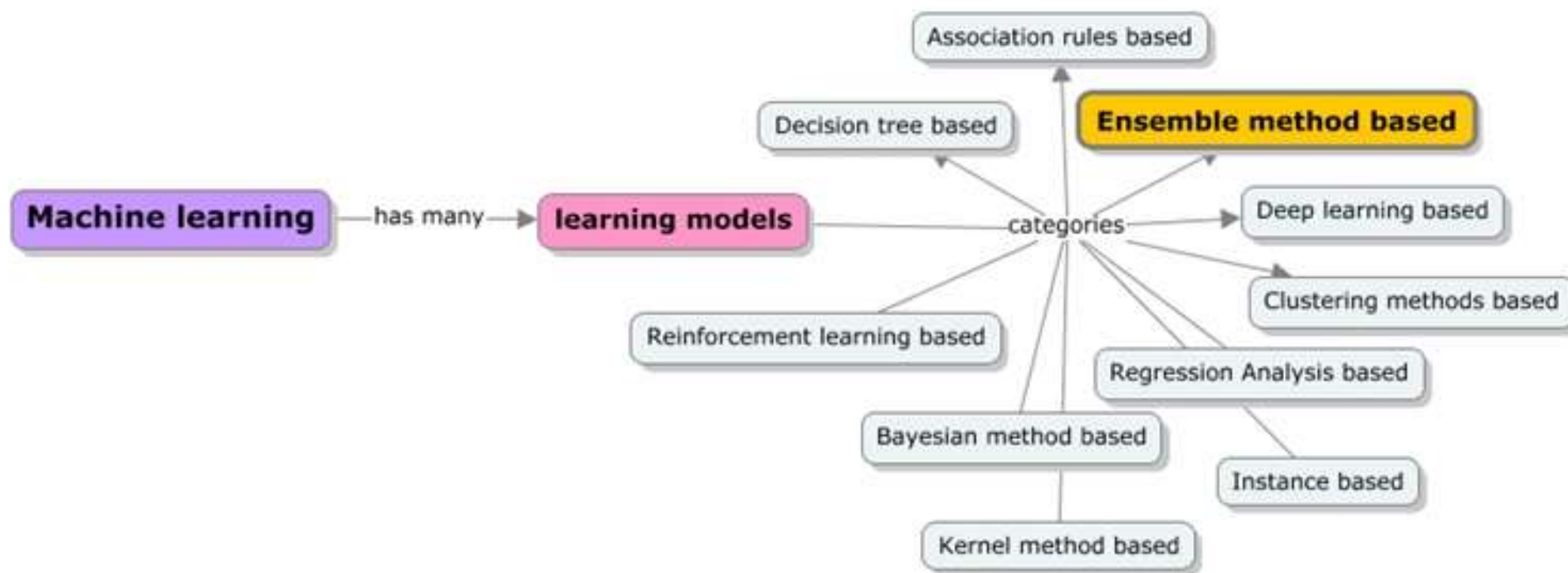
Acurácia e Simplicidade





Aprendizagem Por Agrupamento

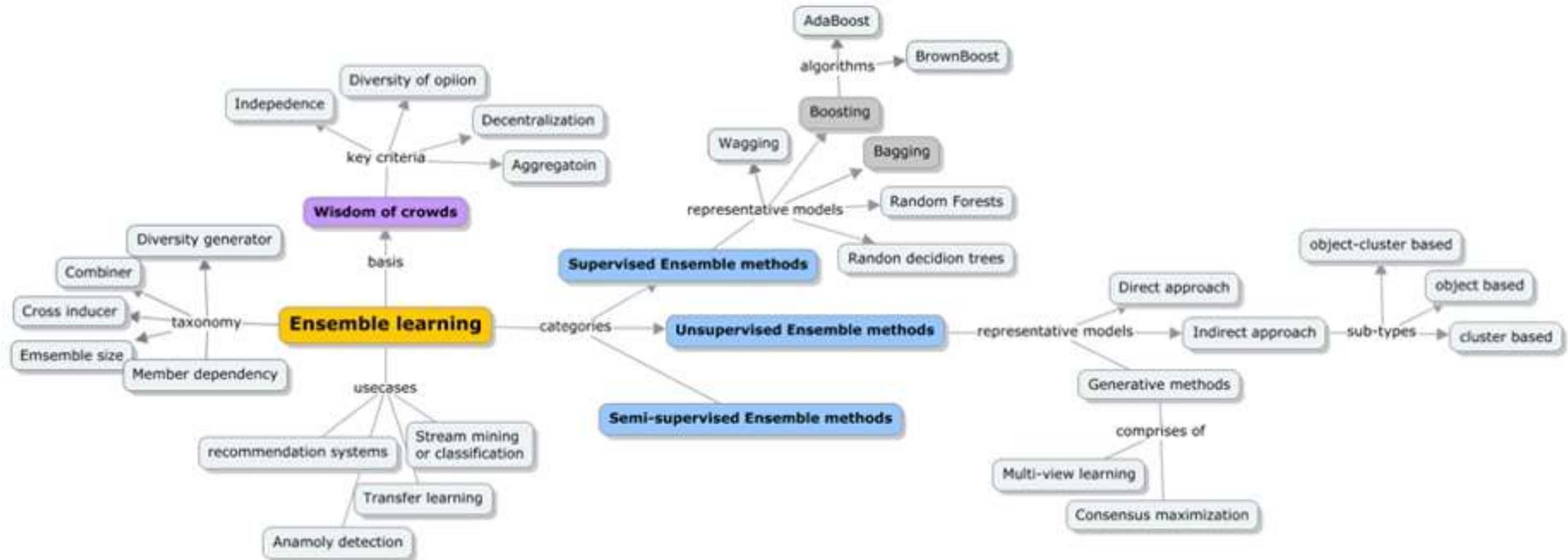
Métodos Ensemble são uma categoria de Algoritmos de Machine Learning





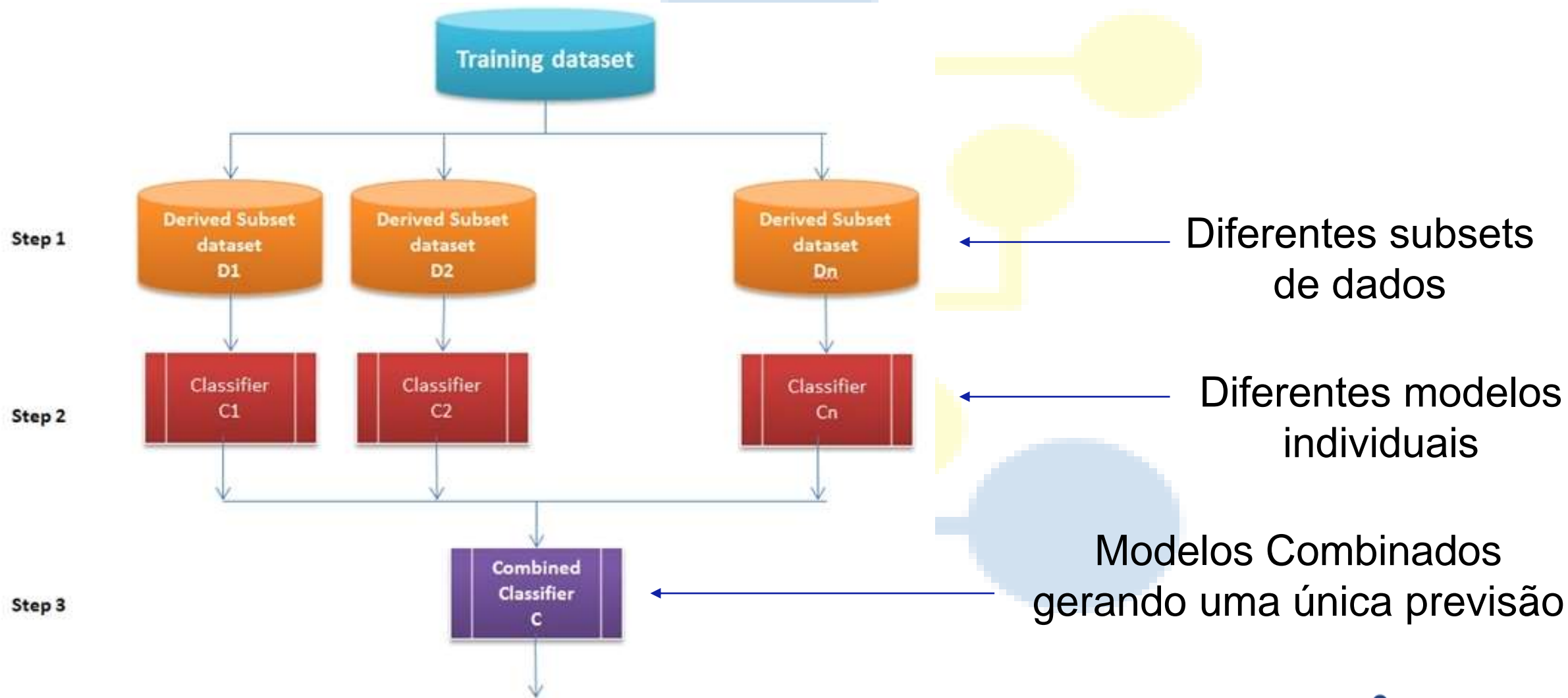
Aprendizagem Por Agrupamento

Ensemble Learning





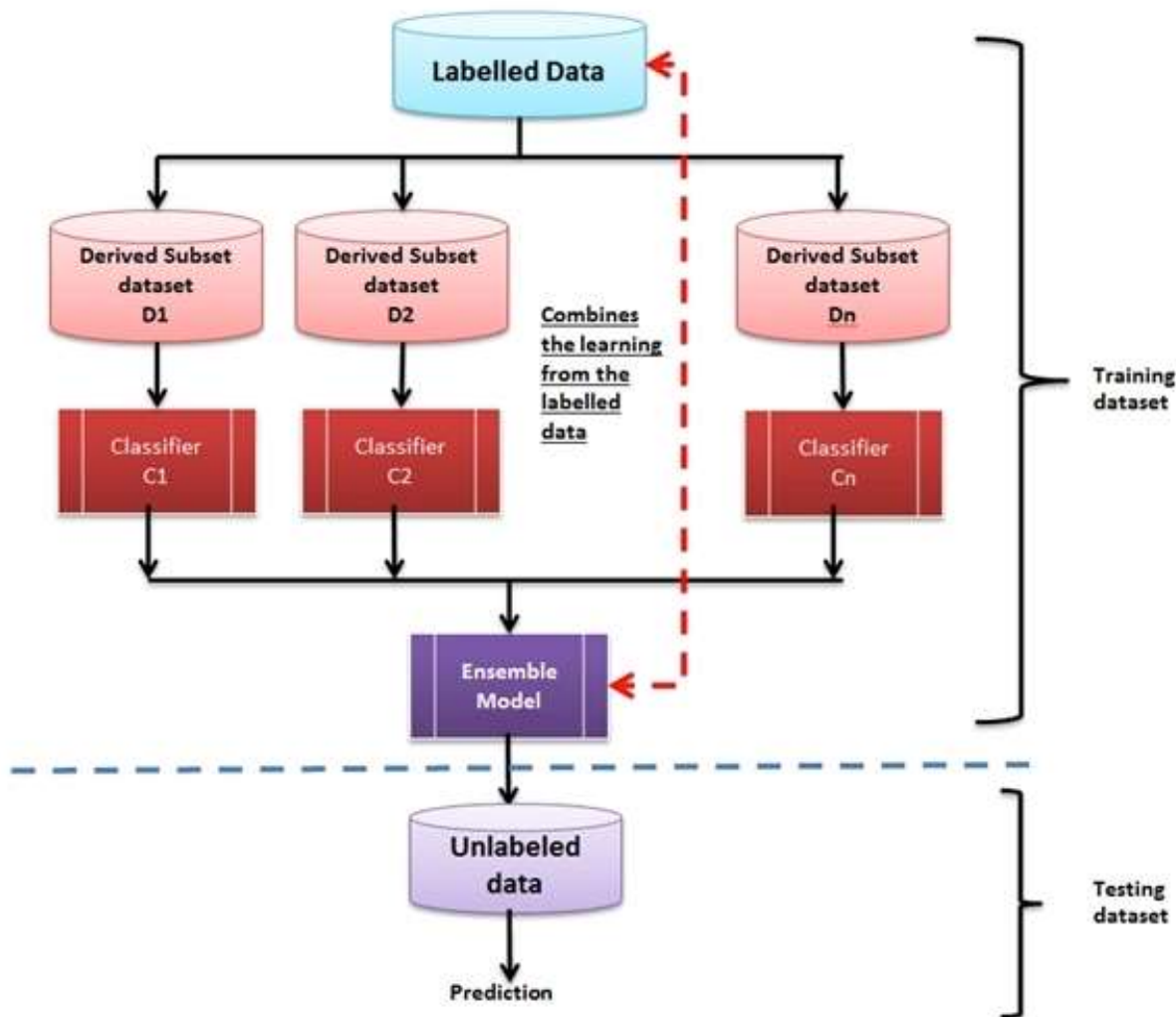
Aprendizagem Por Agrupamento





Aprendizagem Por Agrupamento

Ensemble method for Supervised learning
Combining the "learning" technique



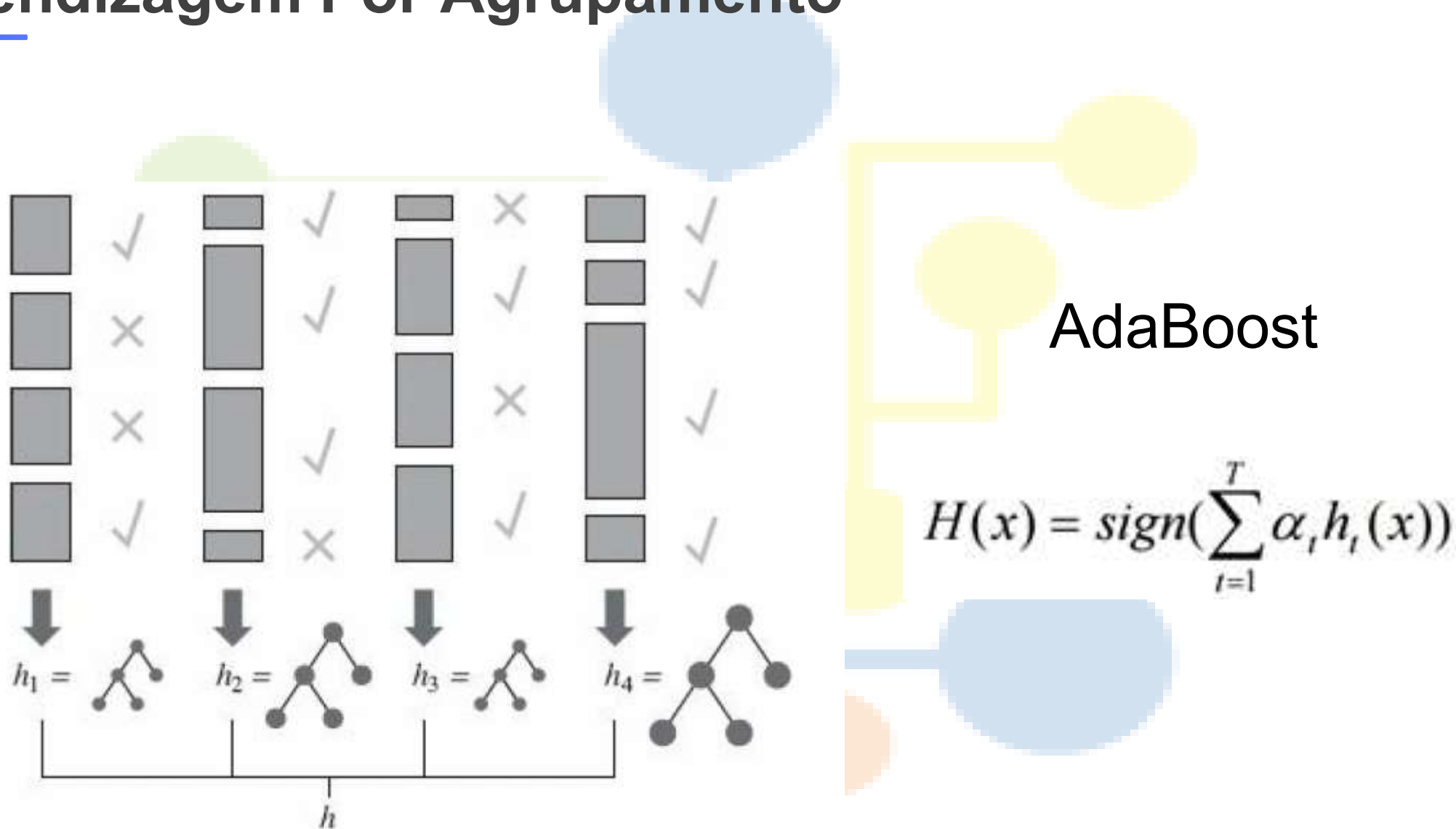
Aqui ocorre a criação de diferentes modelos

Aqui ocorre a validação do modelo final





Aprendizagem Por Agrupamento



AdaBoost

$$H(x) = \text{sign}\left(\sum_{i=1}^T \alpha_i h_i(x)\right)$$





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: \mathbf{w} , um vetor de N pesos de exemplo, inicialmente $1/N$

\mathbf{h} , um vetor de K hipóteses

\mathbf{z} , um vetor de K pesos de hipóteses

para $k = 1$ **até** K **faça**

$\mathbf{h}[k] \leftarrow L(\textit{exemplos}, \mathbf{w})$

$\textit{erro} \leftarrow 0$

para $j = 1$ **até** N **faça**

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

para $j = 1$ **até** N **faça**

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

$\mathbf{w} \leftarrow \text{NORMALIZAR}(\mathbf{w})$

$\mathbf{z}[k] \leftarrow \log(1 - \textit{erro}) / \textit{erro}$

retornar MAIORIA-PONDERADA(\mathbf{h}, \mathbf{z})





Data Science
Academy

Data Science Academy angelicogfa@gmail.com 5b81f7e45e4cdea2118b4569

Obrigado



Data Science Academy



Data Science Academy