



**Departamento de Engenharia Elétrica**

**SEL-0339**

# **Introdução à Visão Computacional**

## **Aula 10**

# **Classificação de Padrões em Imagens**

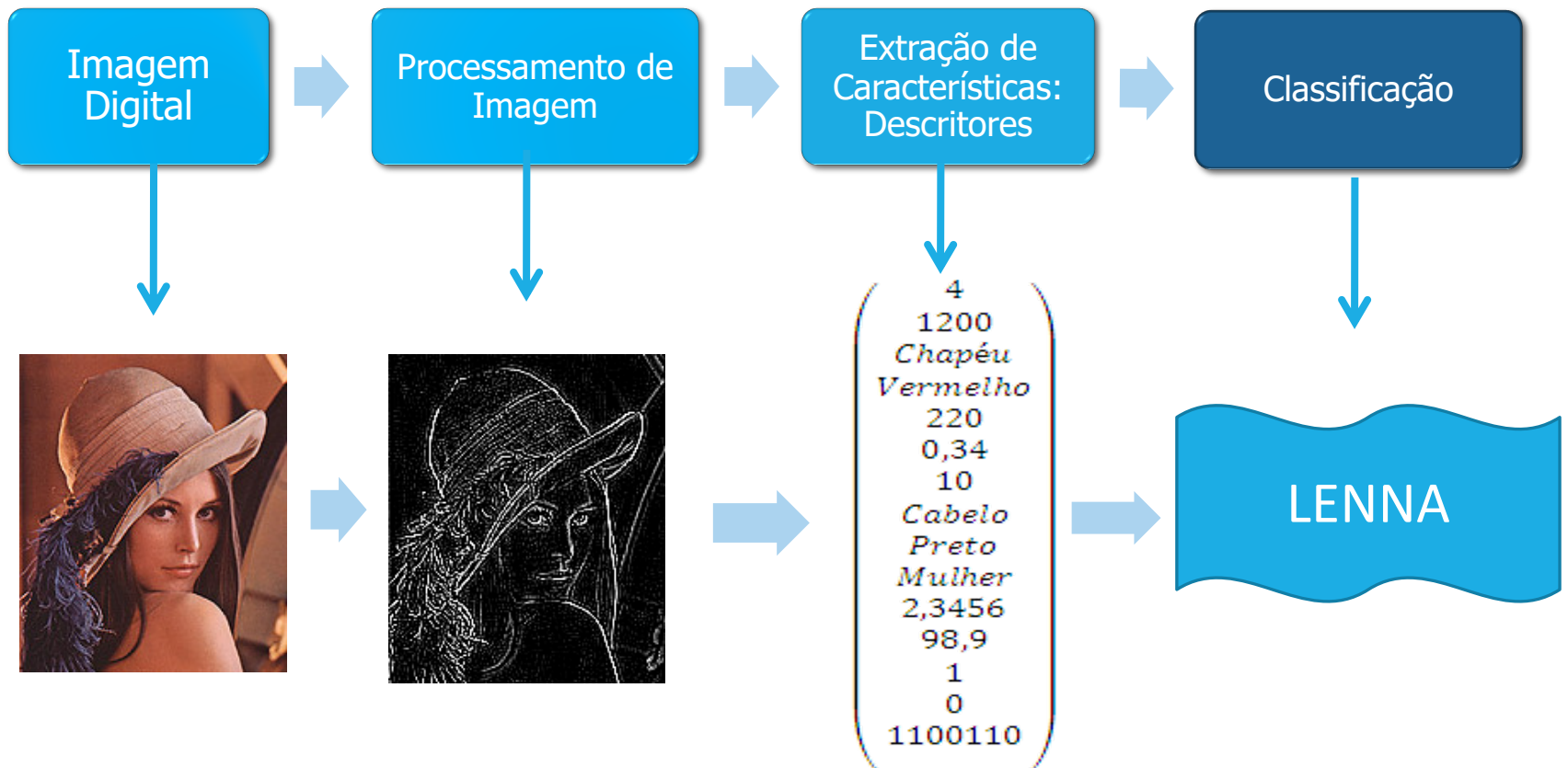
**Prof. Dr. Marcelo Andrade da Costa Vieira**

**Arthur Chaves Costa**

**Renann de Faria Brandão**

[mvieira@sc.usp.br](mailto:mvieira@sc.usp.br)

# Visão Computacional



# Ajuste dos valores de atributos

Os vetores de características devem ser ajustados de alguma maneira para padronizar uma escala de valores no espaço de atributos, desta forma, eliminando discrepâncias entre unidades.

Maneiras comuns de realizar esse ajuste são:

- **Normalização (pela amostra ou pela população):** Ajusta os valores para a faixa de  $[0,1]$  ou  $[-1,1]$  (se possuírem valores negativos)
- **Padronização (z-score):** Ajusta os valores levando em consideração a média e o desvio padrão da variável

É necessário conhecer o seus dados e como o algoritmo de classificação funciona para escolher como fazer o ajuste. Diferentes métodos produzem diferentes resultados dependendo dos fatores.

# Escala dos valores de atributos

Cada uma das raças representa uma **classe**



Bulldog

<b>Altura dorso</b>	35,5 cm
<b>Comprimento</b>	60 cm
<b>Peso</b>	21,5 kg



Beagle

<b>Altura dorso</b>	37 cm
<b>Comprimento</b>	57,5 cm
<b>Peso</b>	10 kg



Whippet

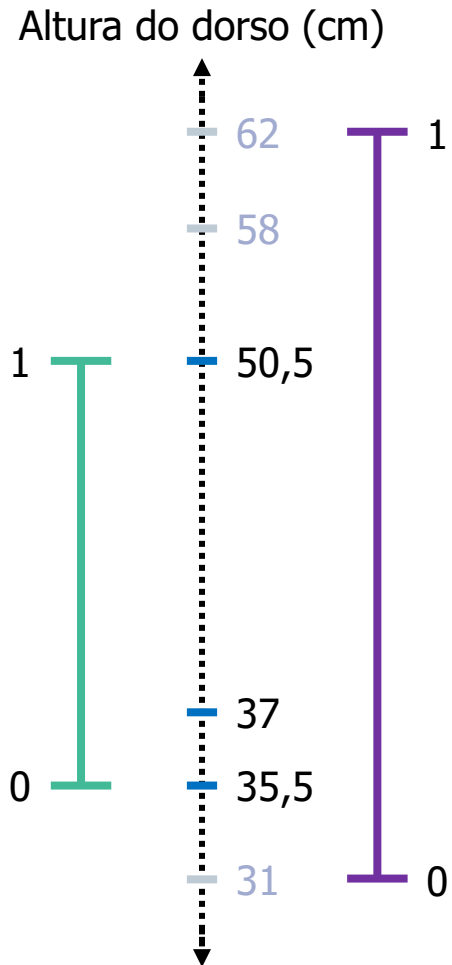
<b>Altura dorso</b>	50,5 cm
<b>Comprimento</b>	68,5 cm
<b>Peso</b>	10,5 kg

# Normalização

Normalização dos atributos do vetor de característica [0,1]:

É necessário conhecer o problema para definir seus limites de normalização. Mesmo ao considerar a população, esta sempre será limitada aos dados disponíveis

Normalização pela amostra



Normalização pela população

É necessário atentar que novas amostras a serem normalizadas podem possuir valores fora da faixa já utilizada, o que pode ocasionar perda de informação

# Normalização pela Amostra

Considerando o grupo amostral, com atributo expresso por  $X$ :

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Bulldog

<b>Altura dorso</b>	35,5 cm
<b>Comprimento</b>	60 cm
<b>Peso</b>	21,5 kg



$$AlturaDorso_{norm} = \frac{35,5 - 35,5}{50,5 - 35,5} = \frac{0}{15} = 0$$

$$Comprimento_{norm} = \frac{60 - 57,5}{68,5 - 57,5} = \frac{2,5}{11} = 0,23$$

$$Peso_{norm} = \frac{21,5 - 10}{21,5 - 10} = \frac{11,5}{11,5} = 1$$



Raça	Beagle	Whippet
<b>Altura dorso</b>	37 cm	50,5 cm
<b>Comprimento</b>	57,5 cm	68,5 cm
<b>Peso</b>	10 kg	10,5 kg



<b>0</b>
<b>0,23</b>
<b>1</b>

<b>0,1</b>
<b>0</b>
<b>0</b>


<b>1</b>
<b>1</b>
<b>0,04</b>

Vetor de características normalizado

# Normalização pela População

Considerando a população, se a tarefa de classificação for abranger outras raças (classes), então deve-se levar em consideração valores máximos e mínimos dos atributos observados em outras raças. Exemplo:

Raça	Chihuahua	Dogue alemão
Altura dorso	16,5 cm	81 cm
Comprimento	24 cm	109 cm
Peso	2,7 kg	79 kg




Bulldog
<b>0,29</b>
<b>0,42</b>
<b>0,25</b>

Vetor de características normalizado

Ou considerar os mínimos e máximos valores observados nas raças (classes) em questão. Com isso se estabelecem novos limites para a normalização.

Raça	Beagle	Whippet	Bulldog
Altura dorso (mín-máx)	33-41 cm	45- <b>56</b> cm	<b>31</b> -40 cm
Comprimento (mín-máx)	<b>51</b> -64 cm	58- <b>79</b> cm	<b>51</b> -69 cm
Peso (mín-máx)	9-11 kg	<b>7</b> -14 kg	18- <b>25</b> kg

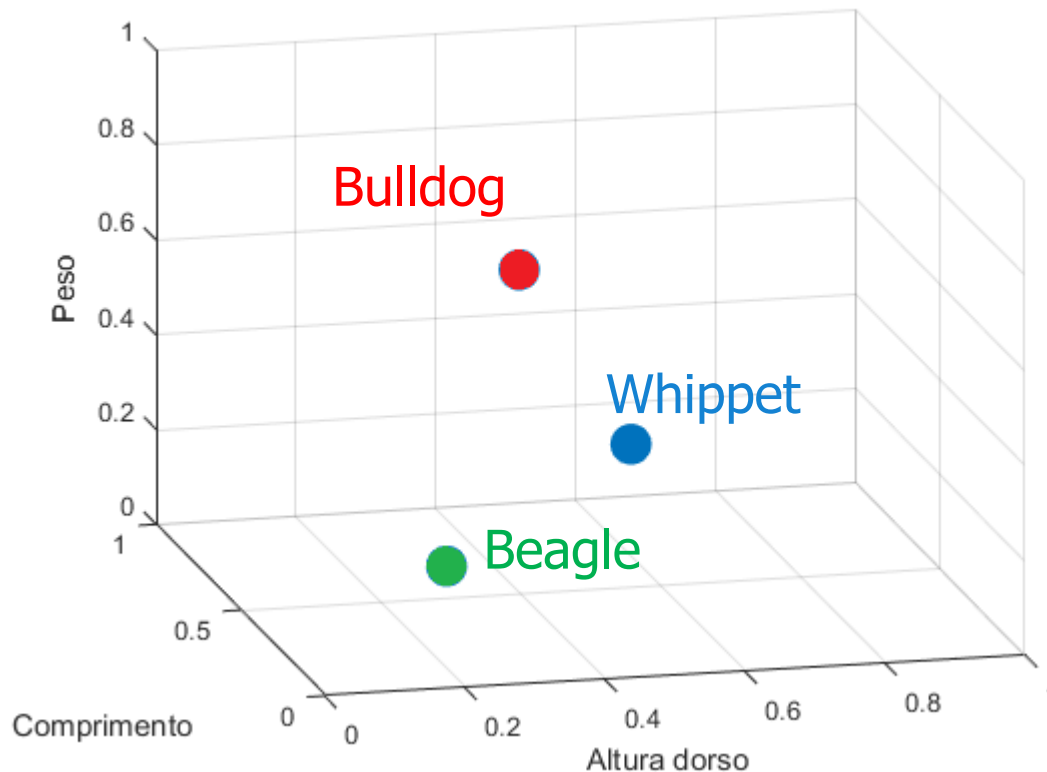


Bulldog
<b>0,18</b>
<b>0,32</b>
<b>0,81</b>

Vetor de características normalizado

# Normalização pela População

Normalização pelos valores registrados da população de cada raça:



Um aspecto importante da normalização é que valores discrepantes do conjunto de amostras (*outliers*) fazem a faixa de valores ser estendida, reduzindo a sensibilidade às diferenças entre os valores que de fato compõem a maioria do grupo estudado



# Padronização

A padronização é realizada de forma a se obter dados com média **0** e desvio padrão igual a **1**. O seu valor irá representar quantos desvios padrão o valor está da média. É conhecida como *Z-score* :

$$Z = \frac{X - \bar{X}}{\sigma_X}$$

Onde  $\bar{X}$  é a média do valor do atributo  $X$  e  $\sigma_X$ , o seu desvio padrão.

Atributo	$\bar{X}$	$\sigma_X$
Altura dorso	41 cm	8,26
Comprimento	62 cm	5,77
Peso	14 kg	6,5

Bulldog

Altura dorso	35,5 cm
Comprimento	60 cm
Peso	21,5 kg



$$Z_{altura\_dorso} = \frac{35,5 - 41}{8,26} = -0,67$$

$$Z_{comprimento} = \frac{60 - 62}{5,77} = -0,35$$

$$Z_{peso} = \frac{21,5 - 14}{6,5} = 1,15$$



Bulldog

**-0,67**

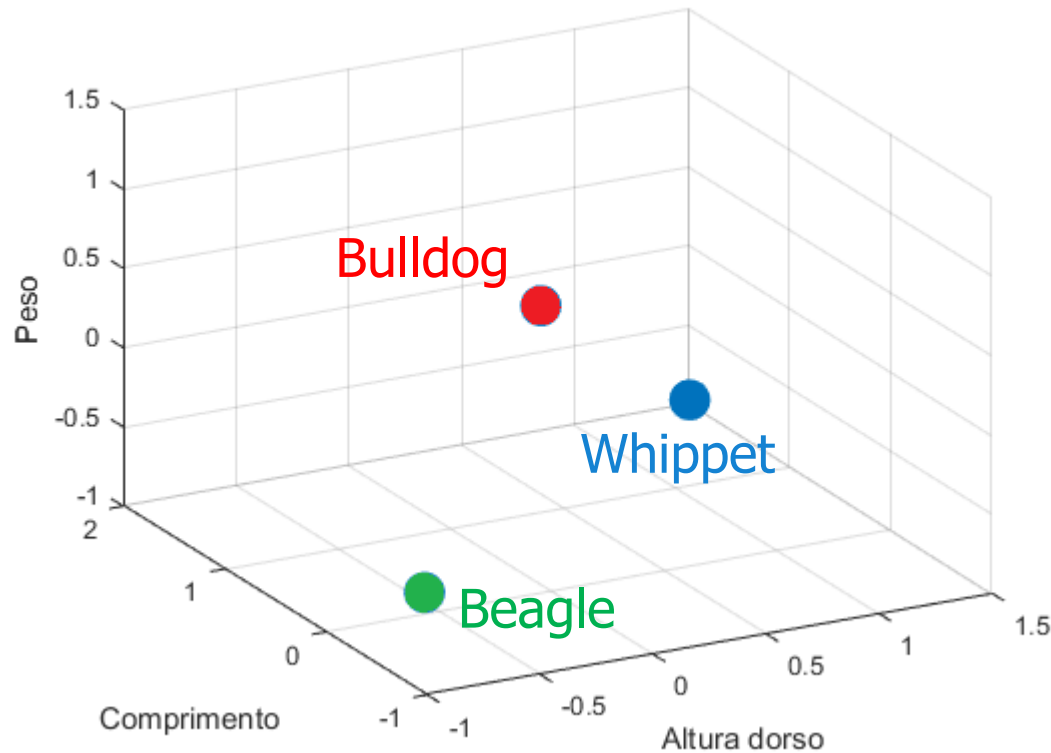
**-0,35**

**1,15**

Vetor de características padronizado

# Padronização

Por z-score:



A padronização permite a comparação de valores entre grupos com média e desvio padrão distintos

Novos elementos a serem padronizados utilizam a média e o desvio padrão do grupo amostral de base, e não acontece de extrapolarem um faixa de valores pré-definida.

# Vetor Protótipo

O **vetor protótipo**, ou **centroide**, de uma classe, é o vetor representante de todos os outros vetores que pertencem àquela classe e assim, a sua vizinhança no espaço de atributos ( $R^N$ , com N sendo a quantidade de atributos).

Vamos considerar que recolhemos as características já citadas de 10 animais de cada uma das três raças:

	Bulldogs										Whippets										Beagles									
	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Altura dorso	39	35	31	34	38	38	36	37	39	31	51	52	48	47	50	49	54	54	52	46	40	40	36	38	40	40	38	35	38	34
Comprimento	56	52	55	64	64	67	61	52	68	65	58	70	64	78	79	64	75	77	71	77	56	60	63	62	57	61	56	64	64	62
Peso	20	22	25	23	24	21	21	22	20	23	14	11	12	11	7	10	12	11	10	14	10	10	9	11	11	11	10	10	9	11

O vetor protótipo das classes-raça pode ser calculado pelo valor médio de cada atributo dos 10 animais.

Após padronização:

**Bulldog**

**-0,79**

**-0,48**

**1,34**

**Whippet**

**1,28**

**0,95**

**-0,58**

**Beagle**

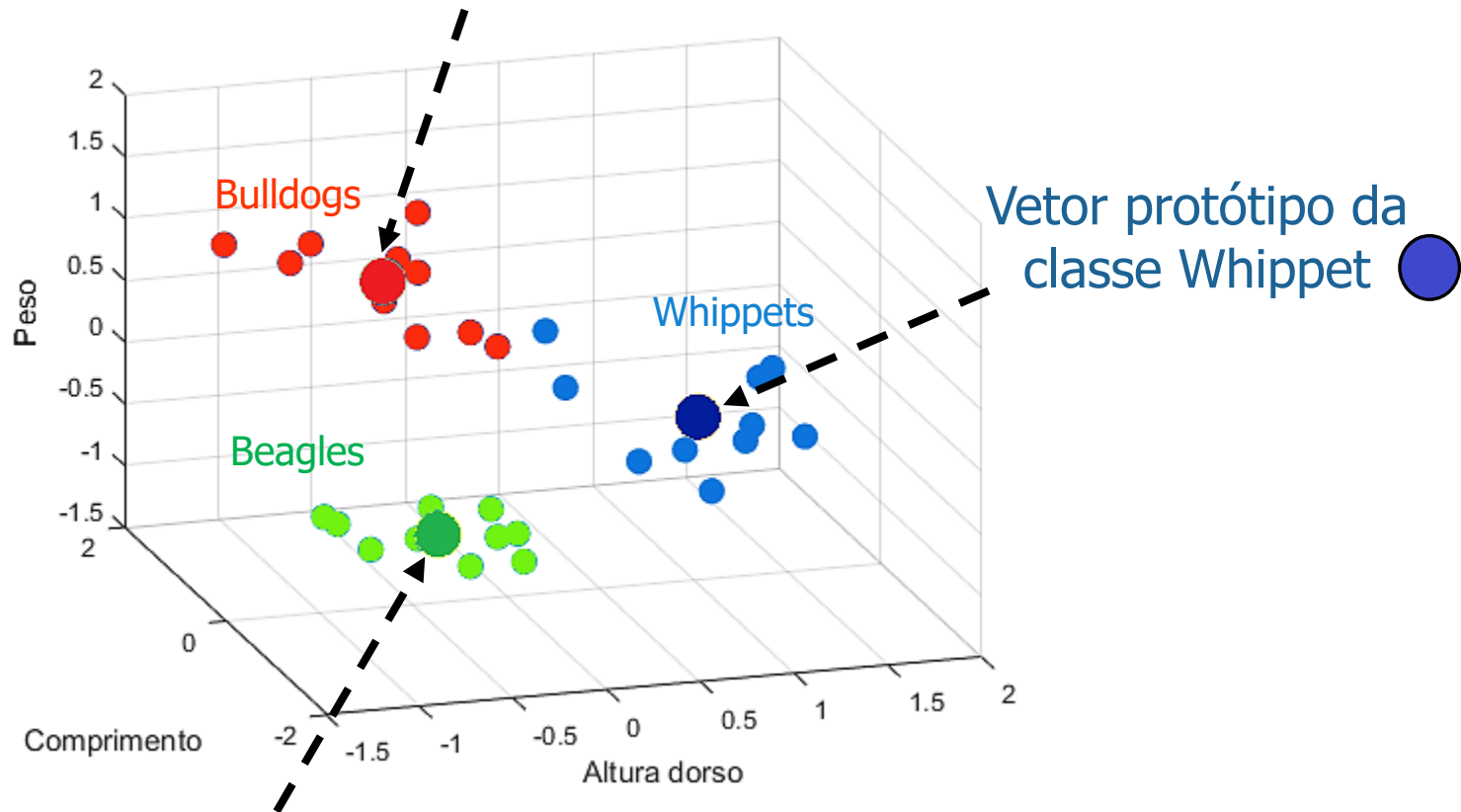
**-0,49**

**-0,47**

**-0,76**

# Vetor Protótipo

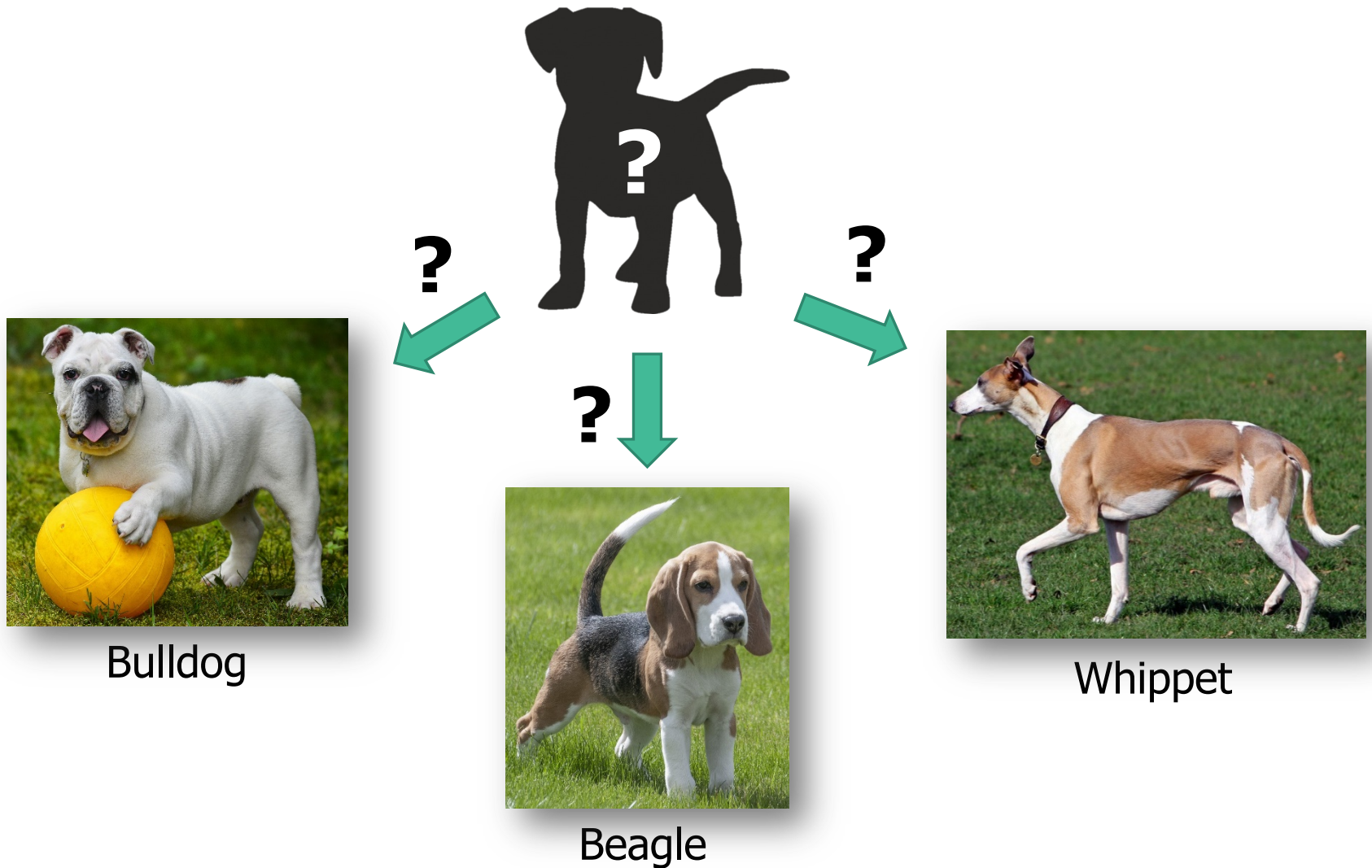
● Vetor protótipo da classe Bulldog



Vetor protótipo da classe Whippet ●

● Vetor protótipo da classe Beagle

# Novo animal, como classificá-lo?



# Novo animal, como classificá-lo?

Novo animal possui:

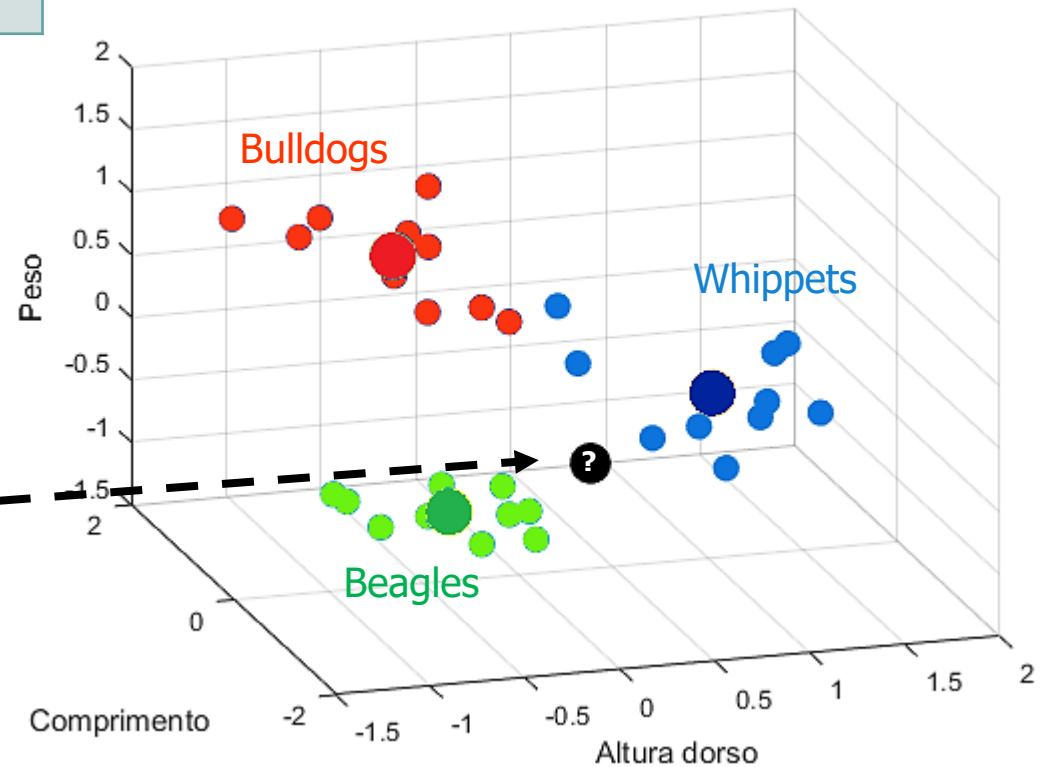


<b>Altura dorso</b>	43 cm
<b>Comprimento</b>	60 cm
<b>Peso</b>	12 kg



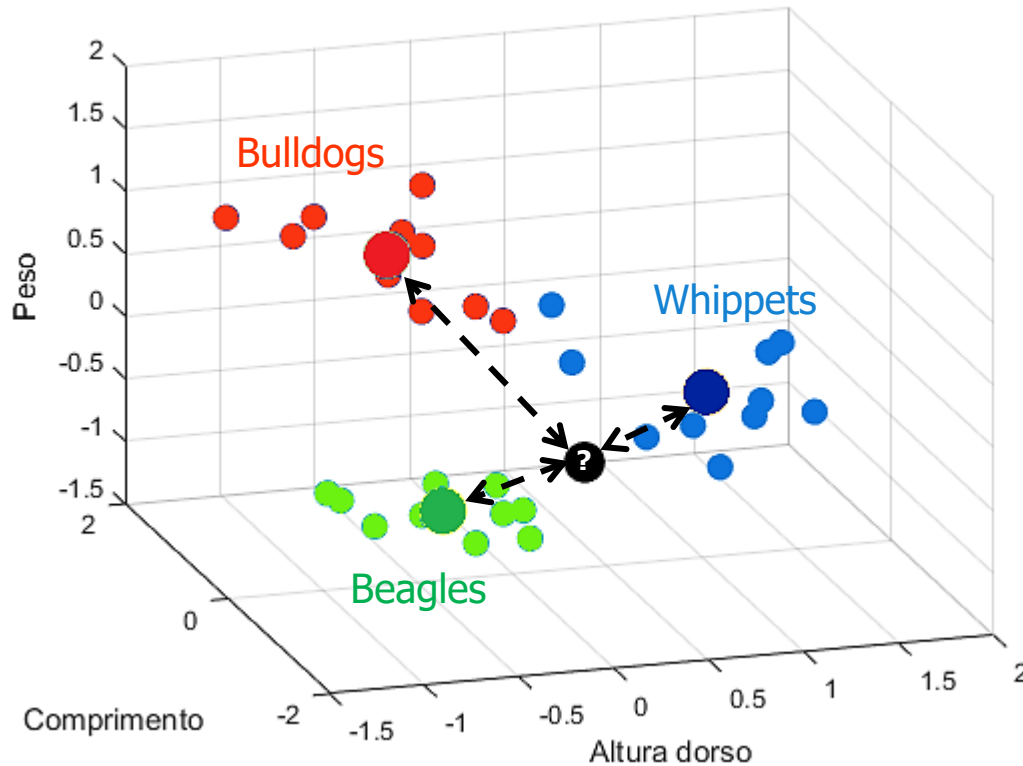
Após padronização:

<b>0,24</b>
<b>-0,53</b>
<b>-0,44</b>



# Novo animal, como classificá-lo?

Uma maneira simples de classificá-lo é associá-lo à classe do vetor protótipo mais próximo. Pode-se para tal, utilizar a **distância euclidiana**, um exemplo de métrica de similaridade.



Distância entre duas instâncias  $\mathbf{p}_i$  e  $\mathbf{p}_j$  definida como:

$$d = \sqrt{\sum_{k=1}^n (p_{ik} - p_{jk})^2}$$

$\mathbf{p}_{ik}$  e  $\mathbf{p}_{jk}$  para  $k = 1, \dots, n$  são os  $n$  atributos que descrevem as instâncias  $\mathbf{p}_i$  e  $\mathbf{p}_j$ , respectivamente

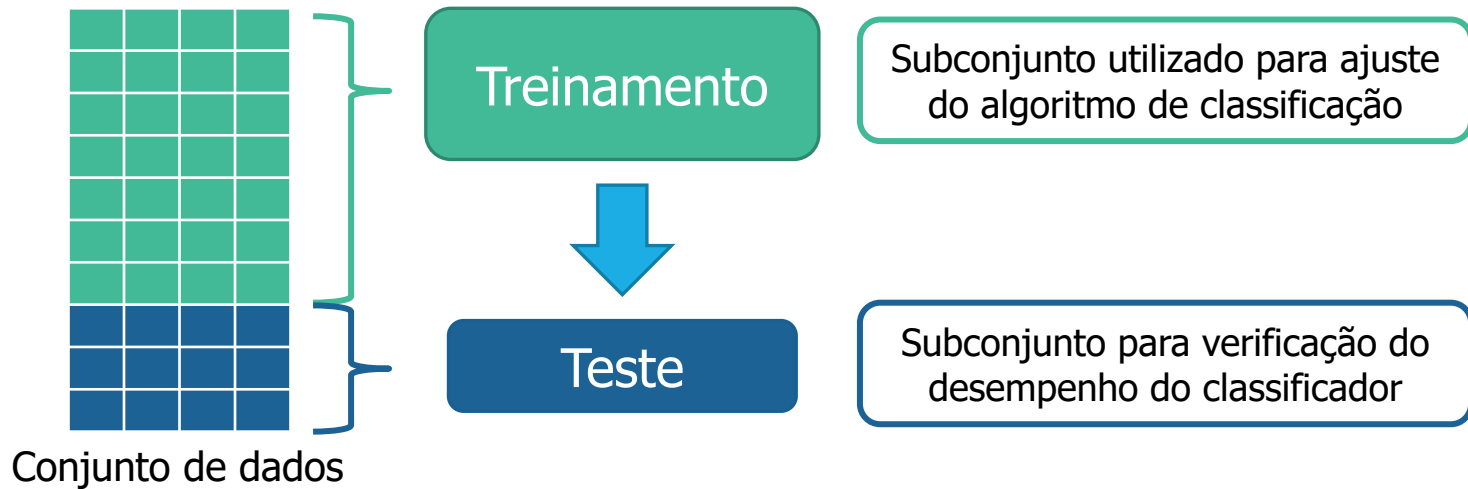
<i>Distância entre os vetores</i>	<i>Distância Euclidiana</i>
V. Protótipo Bulldog	2,05
V. Protótipo Whippet	1,82
V. Protótipo Beagle	0,80



**Classificado como Beagle!**

# Classificação de Padrões em Imagens

Dado um conjunto de dados utilizado para um problema, é necessário dividir o conjunto em subconjuntos que serão utilizados para diferentes tarefas:

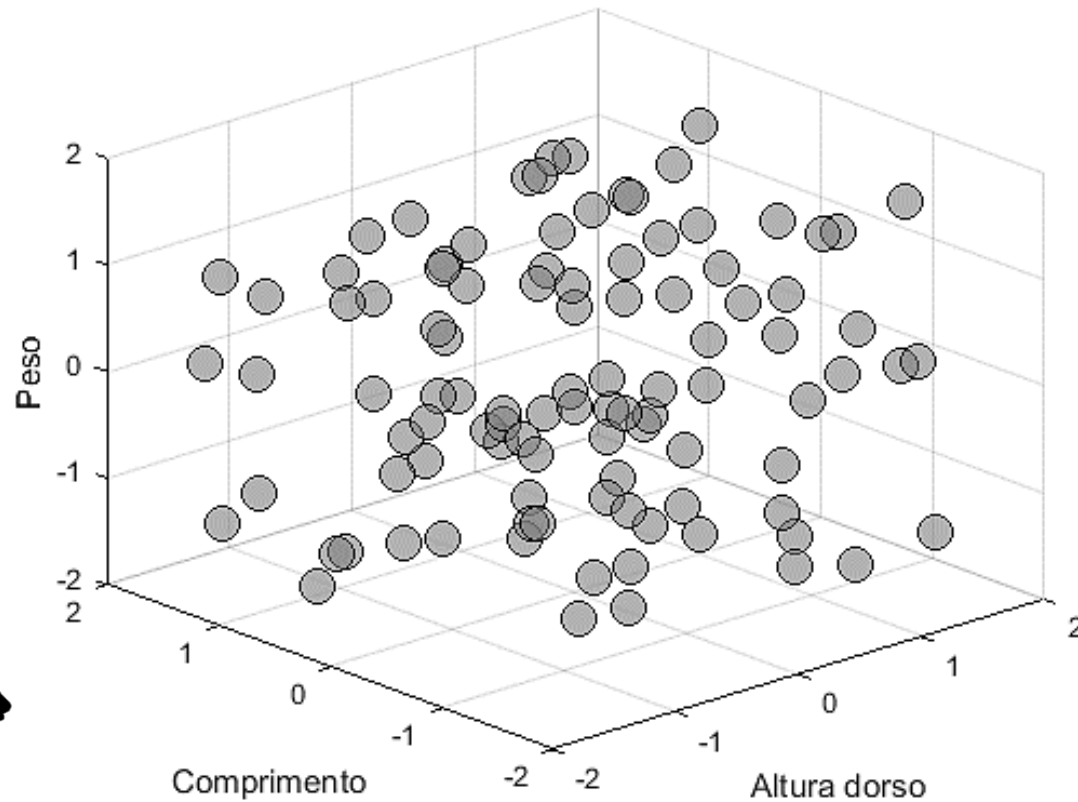


A parcela de dados que é usada em cada parte é arbitrária, porém aconselha-se utilizar ao menos metade deles para o treinamento.

Também é prática comum existir um terceiro subgrupo de **validação** em alguns sistemas, o qual também não é utilizado para treino e serve como referência para estabelecer quando o processo de treinamento deve acabar.



# E quando temos dados sem conhecer as classes?

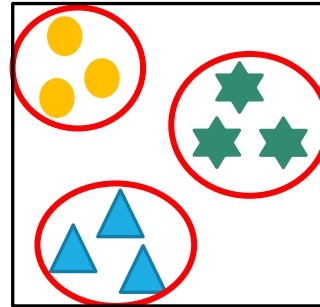


# Reconhecimento de objetos

- Treinamento:
  - Necessidade de se estabelecer vetores protótipos para cada classe;
  - 2 situações possíveis:

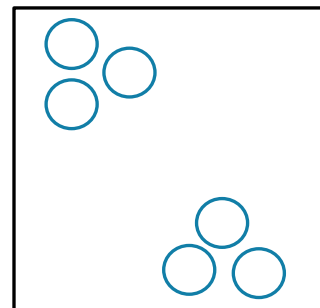
- Conhecimento das amostras e números de classes

↳ Supervisionado

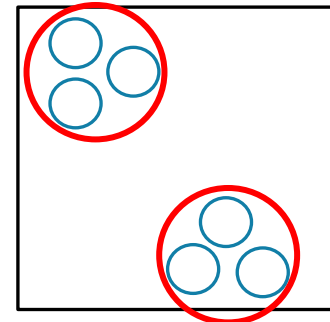


- Não conhecimento das amostras e números de classes:

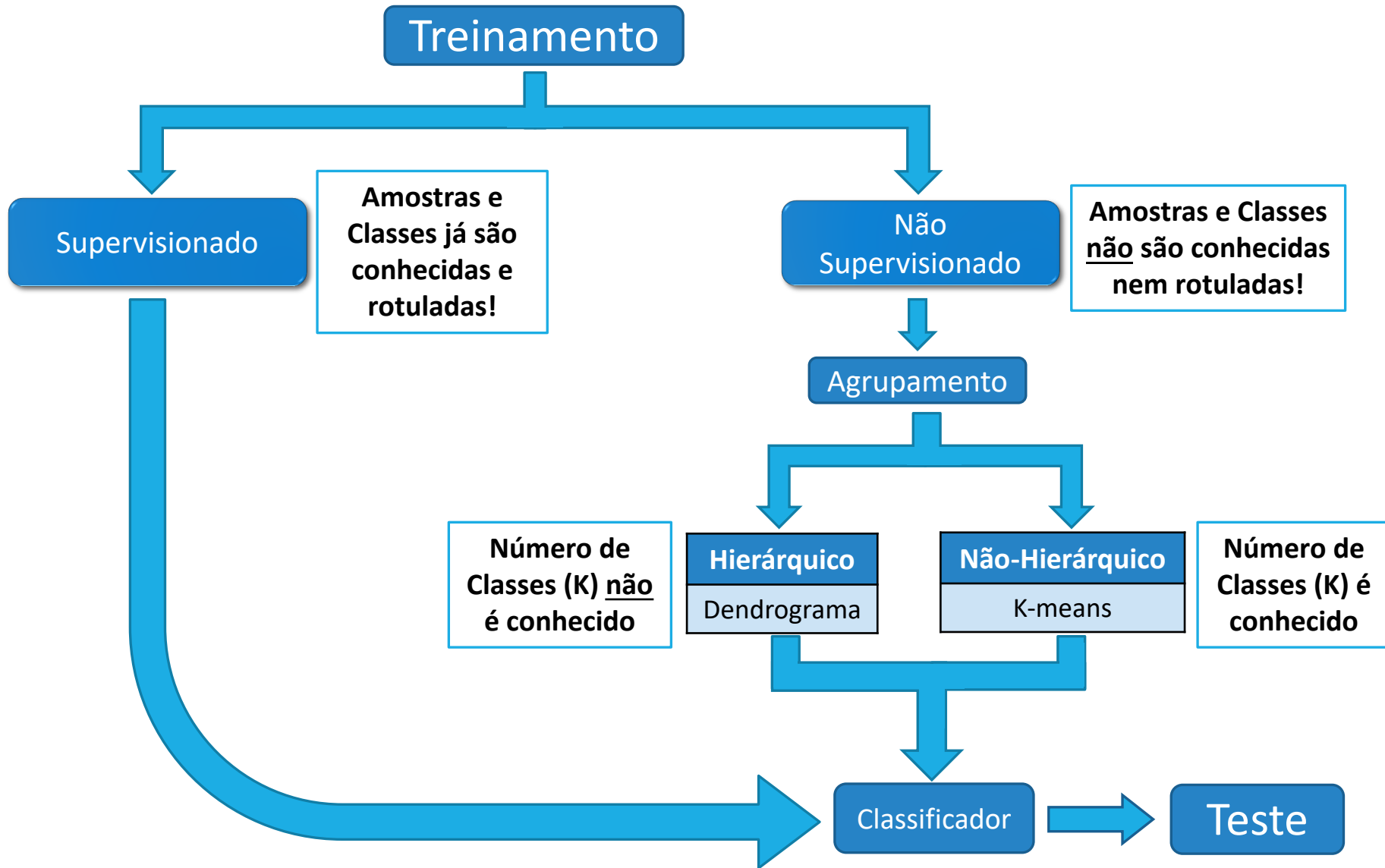
↳ Não Supervisionado



↳ Agrupamento



# Reconhecimento de objetos

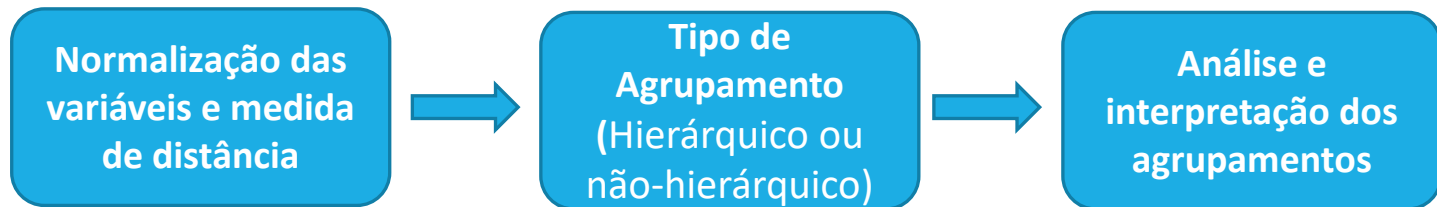


# Treinamento Não Supervisionado

- Diferente da análise supervisionada em que o número de grupos e seus nomes (rótulos) são conhecidos previamente;
- Não se tem informação inicial no número de classes e nem do vetor protótipo de cada uma delas
- Busca identificar os grupos através em características semelhantes entre as amostras analisadas;
- Após o agrupamento, os objetos dentro de um grupo devem ser similares e os objetos entre os diferentes grupos devem ser diferentes em suas características

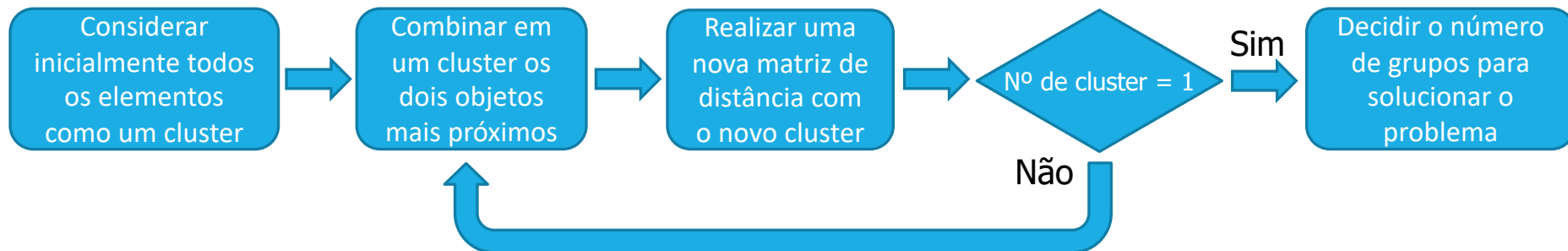
# Análise de agrupamentos

- É um método de Estatística Multivariada que identifica grupos em um grande número de objetos, baseado em suas características.
- O objetivo principal da Análise de Agrupamentos é identificar objetos similares baseado em suas características.
- *Cluster analysis* agrupa objetos similares em grupos tal que os objetos dentro de um grupo são similares e objetos entre os diferentes grupos são significativamente diferentes em suas características.



# Análise de agrupamentos

- Agrupamento hierárquico:
  - Investiga o agrupamento dos dados, simultaneamente em várias escalas, através da geração de uma *Árvore de Grupos* (*Cluster Tree*).
  - A *Árvore de Grupos* não é apenas um simples conjunto de grupos, mas uma Hierarquia em multinível onde grupos em um nível são unidos a grupos em um próximo nível mais alto.
  - Isto permite decidir qual nível ou escala de agrupamento é mais apropriada para cada aplicação.
  - O número de Agrupamentos (“clusters”) e quais são eles é desconhecido.
  - Usa a Matriz de Distâncias ou a Similaridade para construir um gráfico de *Árvore de Grupo* chamado de Dendrograma.



# Análise de agrupamentos

- Agrupamento hierárquico:
- Exemplo:
  - Desempenho de jogadores de futebol em toda a carreira (até novembro de 2019);

	Jogos	Gols	Assistências
<b>Cristiano Ronaldo</b>	817	607	213
<b>Messi</b>	698	612	247
<b>Gabriel Barbosa</b>	175	73	21
<b>Lewandowski</b>	561	379	110
<b>Rafael Moura</b>	312	86	20
<b>Fred</b>	407	188	53
<b>Alexandre Pato</b>	380	154	46
<b>Neymar</b>	383	230	140

# Análise de agrupamentos

- Agrupamento hierárquico:
  - 1ª etapa: Normalização dos dados

	Jogos	Gols	Assistências
<b>Cristiano Ronaldo</b>	<b>817</b>	607	213
<b>Messi</b>	698	<b>612</b>	<b>247</b>
<b>Gabriel Barbosa</b>	175	73	21
<b>Lewandowski</b>	561	379	110
<b>Rafael Moura</b>	312	86	20
<b>Fred</b>	407	188	53
<b>Alexandre Pato</b>	380	154	46
<b>Neymar</b>	383	230	140



	Jogos	Gols	Assistências
<b>Cristiano Ronaldo</b>	1,000	0,992	0,862
<b>Messi</b>	0,854	1,000	1,000
<b>Gabriel Barbosa</b>	0,214	0,119	0,085
<b>Lewandowski</b>	0,687	0,619	0,445
<b>Rafael Moura</b>	0,382	0,141	0,081
<b>Fred</b>	0,498	0,307	0,215
<b>Alexandre Pato</b>	0,465	0,252	0,186
<b>Neymar</b>	0,469	0,376	0,567



# Análise de agrupamentos

- Agrupamento hierárquico:
  - 2ª etapa: Cálculo da Matriz de Distâncias:

Cluster	Jogador	Jogos	Goals	Assistências
J1	Cristiano Ronaldo (J1)	1,000	0,992	0,862
J2	Messi (J2)	0,854	1,000	1,000
J3	Gabriel Barbosa (J3)	0,214	0,119	0,085
J4	Lewandowski (J4)	0,687	0,619	0,445
J5	Rafael Moura (J5)	0,382	0,141	0,081
J6	Fred (J6)	0,498	0,307	0,215
J7	Alexandre Pato (J7)	0,465	0,252	0,186
J8	Neymar (J8)	0,469	0,376	0,567

Distancia  
Euclidiana

Matriz  
Distância



	J1	J2	J3	J4	J5	J6	J7	J8
J1	0,000	0,201	1,408	0,641	1,310	1,068	1,136	0,865
J2	0,201	0,000	1,422	0,694	1,344	1,106	1,172	0,852
J3	1,408	1,422	0,000	0,777	0,169	0,365	0,301	0,603
J4	0,641	0,694	0,777	0,000	0,674	0,431	0,501	0,349
J5	1,310	1,344	0,169	0,674	0,000	0,243	0,174	0,547
J6	1,068	1,106	0,365	0,431	0,243	0,000	0,070	0,360
J7	1,136	1,172	0,301	0,501	0,174	0,070	0,000	0,401
J8	0,865	0,852	0,603	0,349	0,547	0,360	0,401	0,000

# Análise de agrupamentos

- Agrupamento hierárquico:
  - 3ª etapa: Encontrar a menor distância entre os clusters, combinar em um único cluster e atualizar as distâncias;

	J1	J2	J3	J4	J5	J6	J7	J8
J1	0,000	0,201	1,408	0,641	1,310	1,068	1,136	0,865
J2	0,201	0,000	1,422	0,694	1,344	1,106	1,172	0,852
J3	1,408	1,422	0,000	0,777	0,169	0,365	0,301	0,603
J4	0,641	0,694	0,777	0,000	0,674	0,431	0,501	0,349
J5	1,310	1,344	0,169	0,674	0,000	0,243	0,174	0,547
J6	1,068	1,106	0,365	0,431	0,243	0,000	0,070	0,360
J7	1,136	1,172	0,301	0,501	0,174	0,070	0,000	0,401
J8	0,865	0,852	0,603	0,349	0,547	0,360	0,401	0,000



- Menor distância entre o cluster 7 e 6
- Combinar o cluster 7 e o 6 em um único cluster.

$d_{1,(6,7)}$	$\min(d_{1,6}; d_{1,7}) = \min(1,068; 1,136) = 1,068$
$d_{2,(6,7)}$	$\min(d_{2,6}; d_{2,7}) = \min(1,106; 1,172) = 1,106$
$d_{3,(6,7)}$	$\min(d_{3,6}; d_{3,7}) = \min(0,365; 0,301) = 0,301$
$d_{4,(6,7)}$	$\min(d_{4,6}; d_{4,7}) = \min(0,431; 0,501) = 0,431$
$d_{5,(6,7)}$	$\min(d_{5,6}; d_{5,7}) = \min(0,243; 0,174) = 0,174$
$d_{8,(6,7)}$	$\min(d_{8,6}; d_{8,7}) = \min(0,360; 0,401) = 0,360$

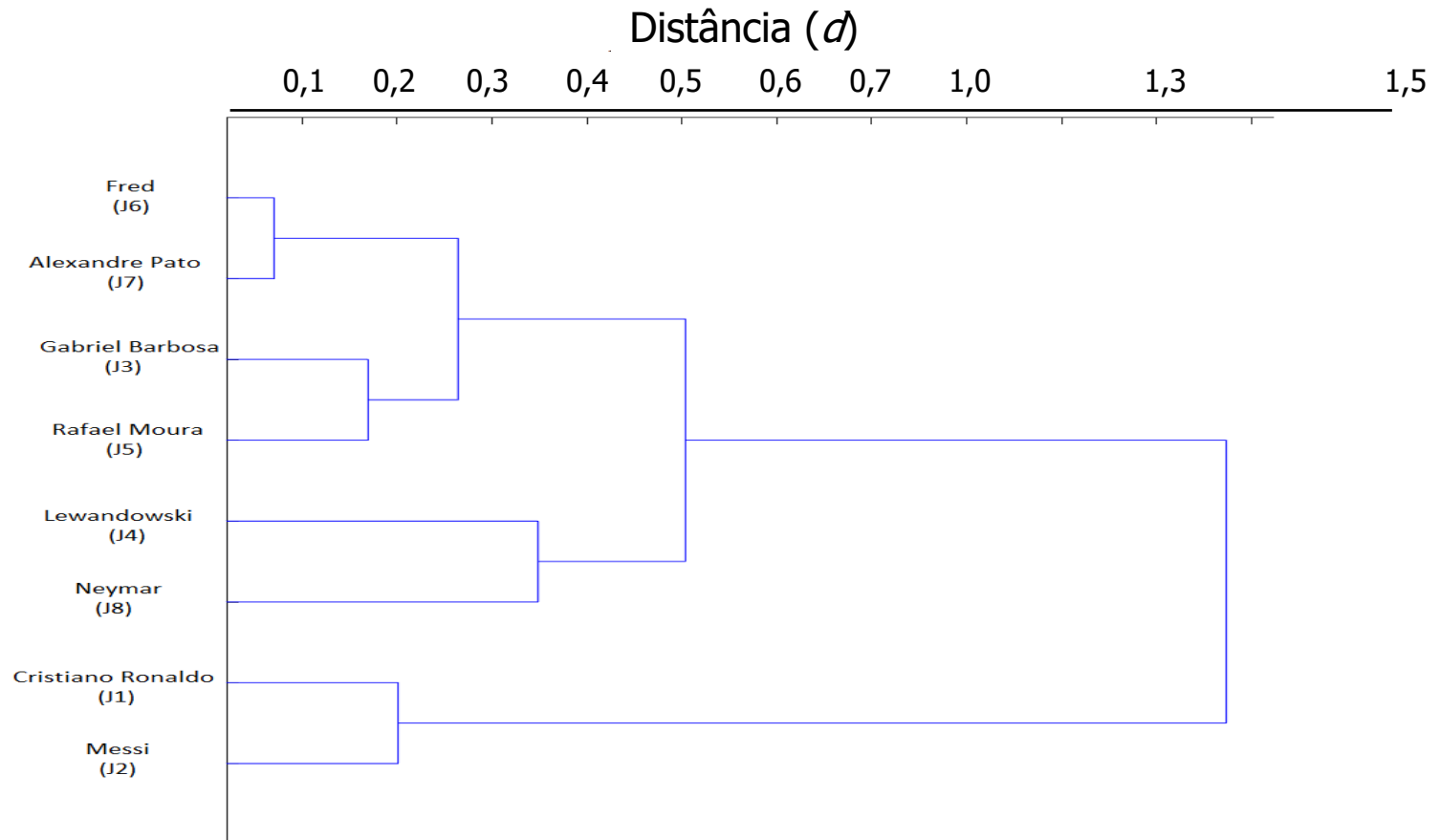
# Análise de agrupamentos

- Agrupamento hierárquico:
  - 3ª etapa: Encontrar a menor distância entre os clusters, combinar em um único cluster e atualizar as distâncias em uma nova matriz de distâncias, que será uma dimensão menor que a anterior;

	J1	J2	J3	J4	J5	J6,7	J8
J1	0,000	0,201	1,408	0,641	1,310	1,068	0,865
J2	0,201	0,000	1,422	0,694	1,344	1,106	0,852
J3	1,408	1,422	0,000	0,777	0,169	0,301	0,603
J4	0,641	0,694	0,777	0,000	0,674	0,431	0,349
J5	1,310	1,344	0,169	0,674	0,000	0,174	0,547
J6,7	1,068	1,106	0,301	0,431	0,174	0,000	0,360
J8	0,865	0,852	0,603	0,349	0,547	0,360	0,000

# Análise de agrupamentos

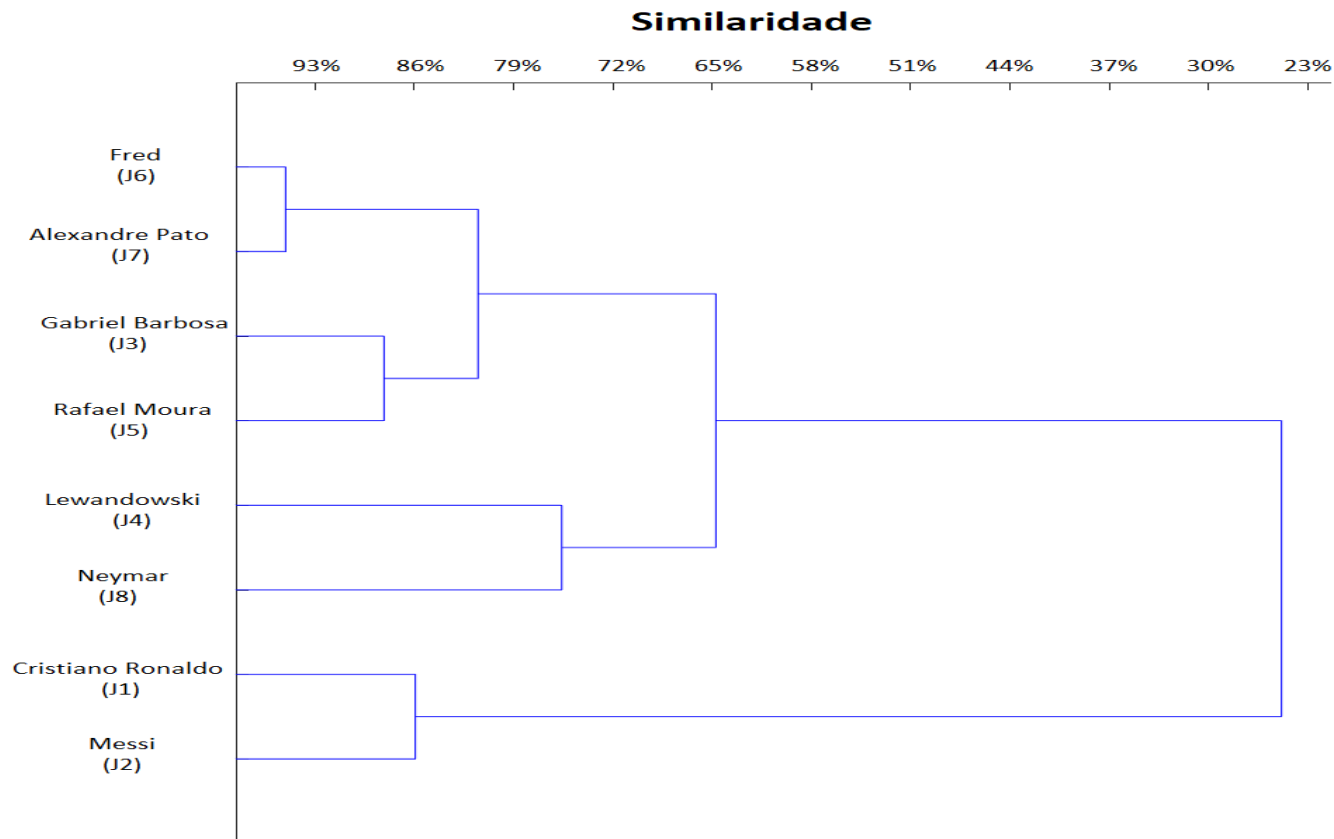
- Agrupamento hierárquico:
  - 4ª etapa: Repetir a etapa 3 até o número de classes ficar igual a 1
  - Estabelecer um Dendrograma: Classes x Distância;



# Dendograma

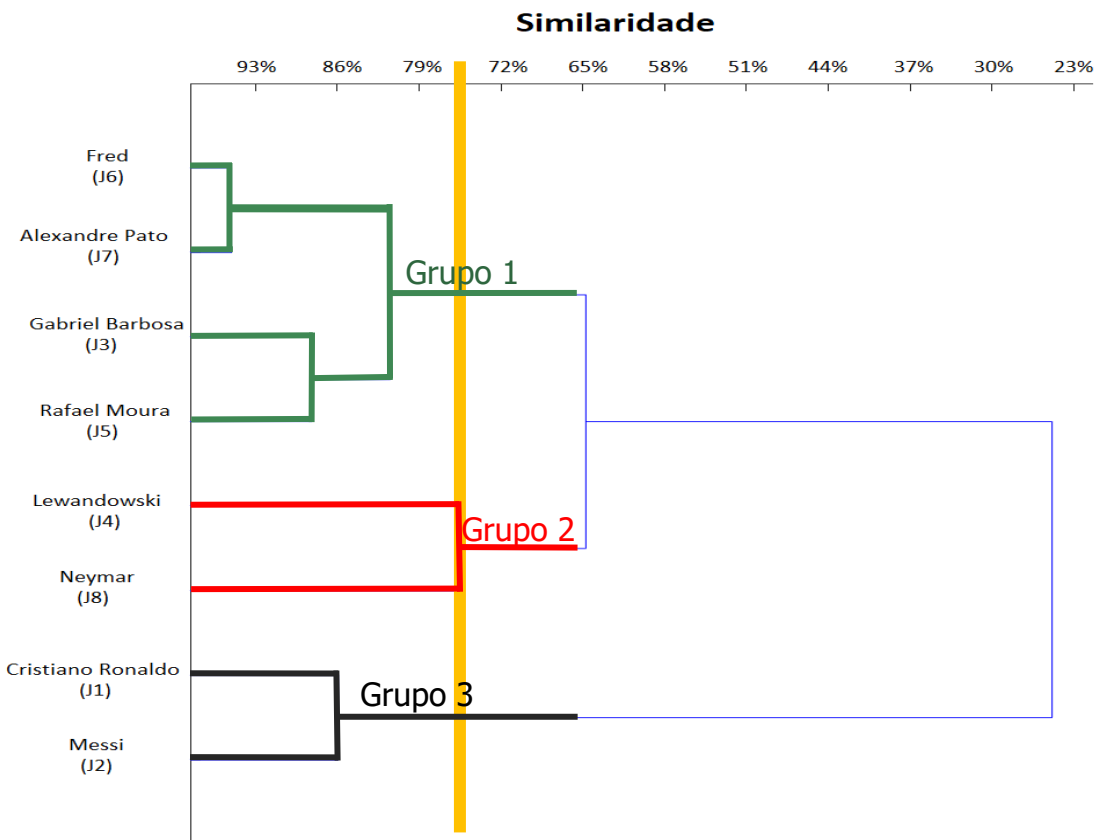
- Também pode-se usar a **Similaridade** ao invés da distância

$$\text{Similaridade: } s_{rs} = 1 - \frac{d_{rs}}{d_{\max}}$$



# Análise de agrupamentos

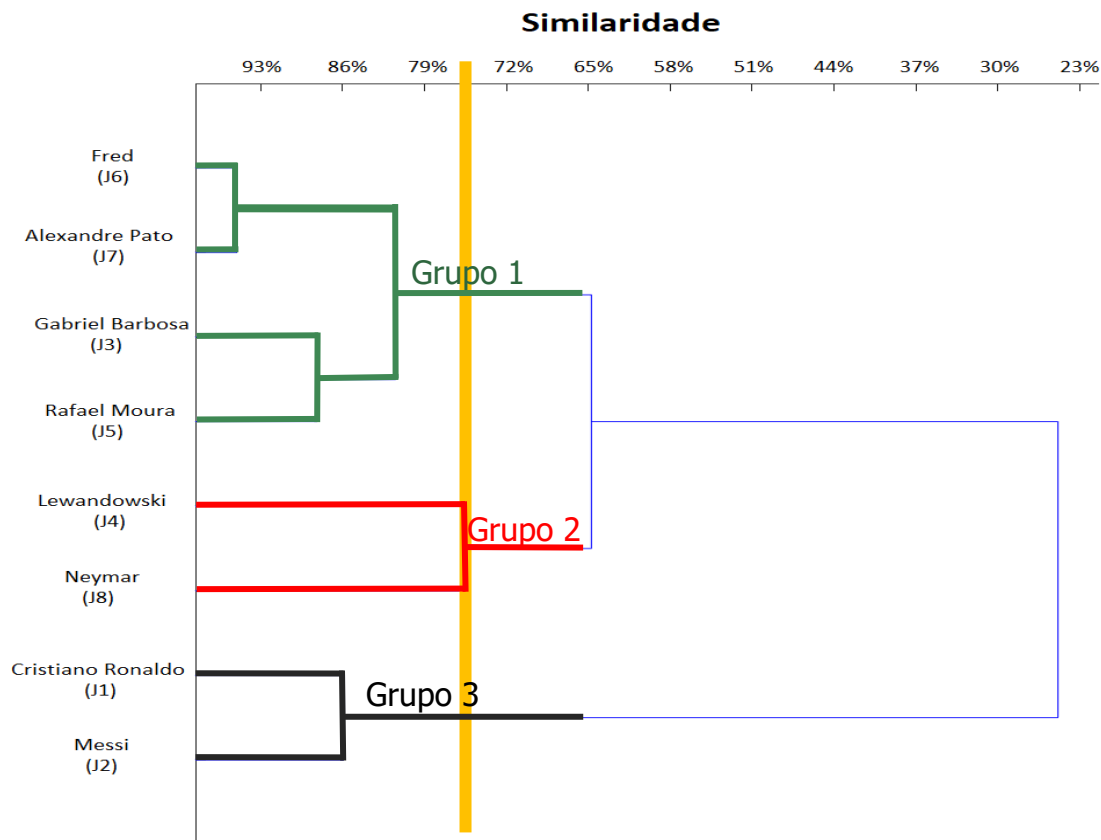
- Agrupamento hierárquico:
  - Para um índice de similaridade de 75% temos 3 grupos de jogadores.



	Jogos	Gols	Assistências
<b>Cristiano Ronaldo</b>	<b>817</b>	<b>607</b>	<b>213</b>
<b>Messi</b>	<b>698</b>	<b>612</b>	<b>247</b>
<b>Gabriel Barbosa</b>	175	73	21
<b>Rafael Moura</b>	312	86	20
<b>Fred</b>	407	188	53
<b>Alexandre Pato</b>	380	154	46
<b>Lewandowski</b>	561	379	110
<b>Neymar</b>	383	230	140

# Análise de agrupamentos

- Agrupamento hierárquico:
  - Para um índice de similaridade de 75% temos 3 grupos de jogadores:



- **Grupo 1:**  
Fred, Alexandre Pato, Gabriel,  
Rafael Moura;
- **Grupo 2 :**  
Lewandowski, Neymar;
- **Grupo 3 :**  
Cristiano Ronaldo, Messi;

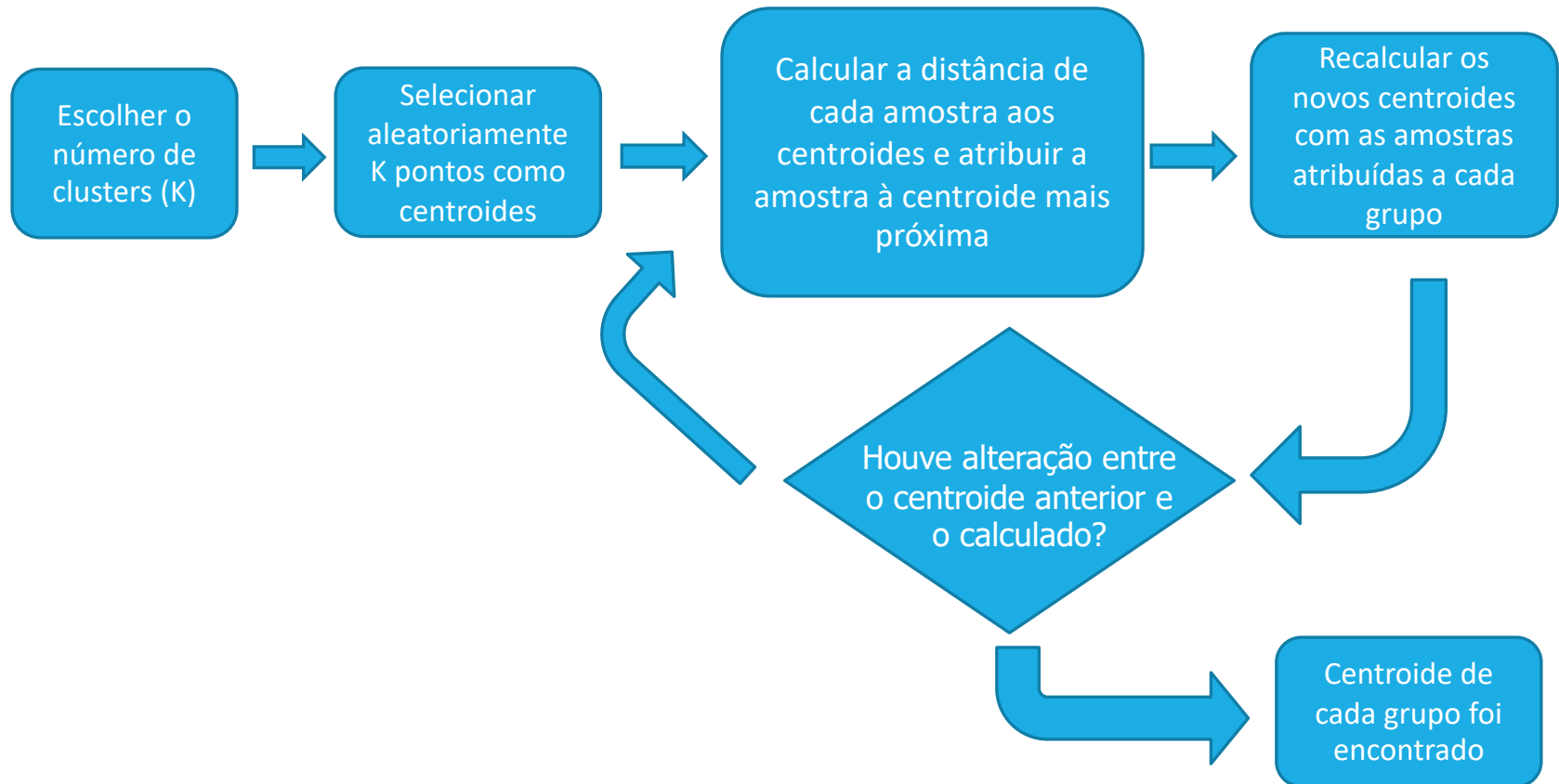
# Análise de agrupamentos

- Agrupamento não hierárquico:
  - Deve-se primeiramente selecionar o número de clusters que se deseja formar;
  - Um dos métodos mais utilizados é o *k-means*
  - A ideia básica é encontrar grupos similares, de maneira a minimizar as distâncias.
  - As distâncias são calculadas entre os pontos e as médias de cada um dos  $k$  grupos, chamadas *centroides*.



# Análise de agrupamentos

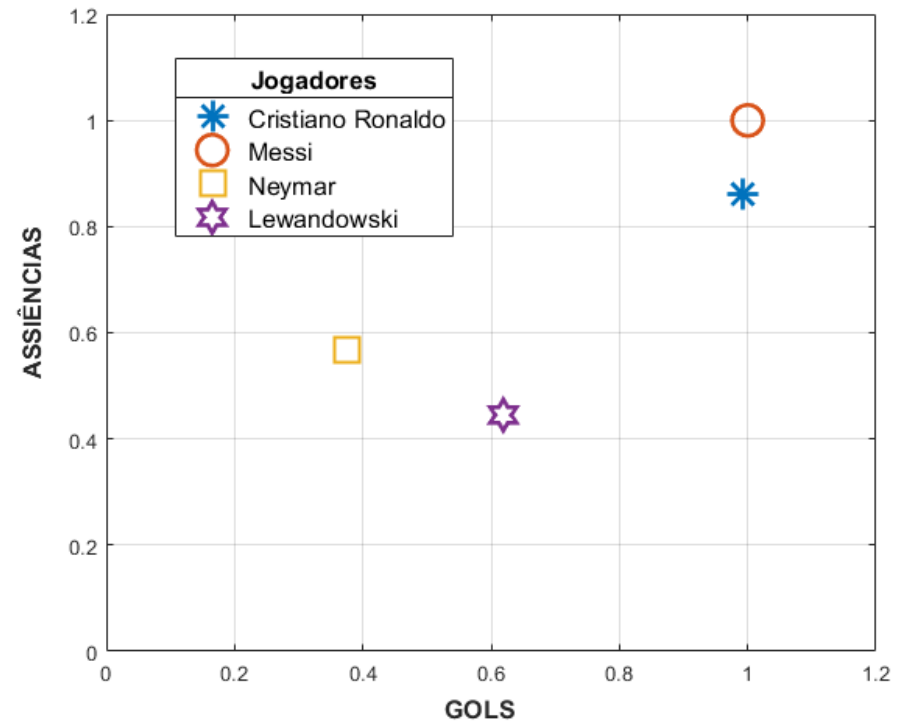
- Agrupamento Não-hierárquico:
  - *K-means*



# Análise de agrupamentos

- K-means
  1. Escolher o número de clusters
    - Exemplo para  $K = 2$ :

	Gols	Assistências
Neymar	0,376	0,567
Lewandowski	0,619	0,445
Cristiano Ronaldo	0,992	0,862
Messi	1,000	1,000



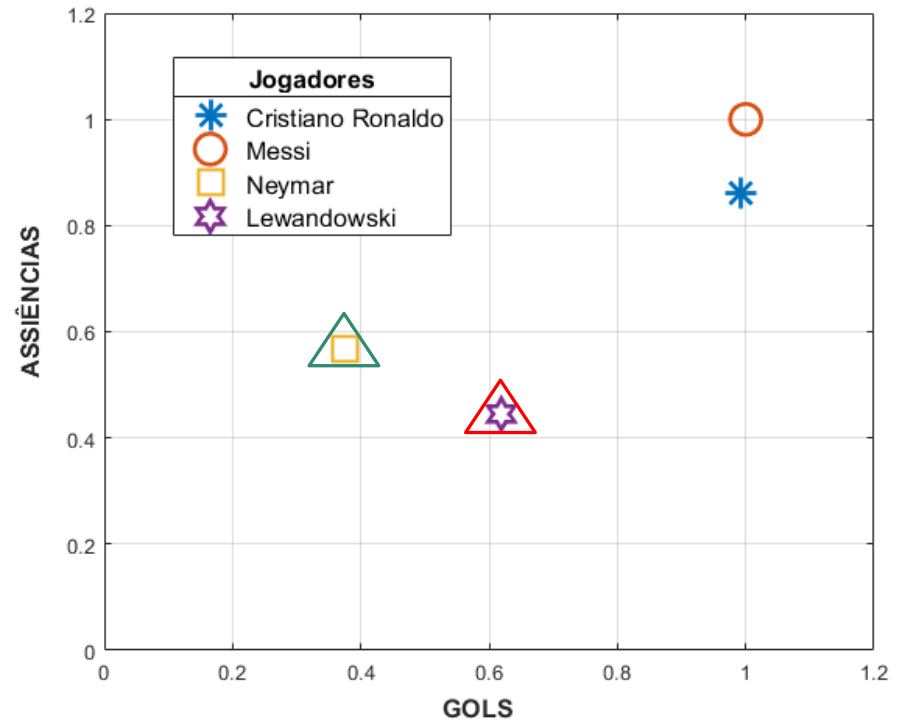
# Análise de agrupamentos

- K-means

2. Escolher posições iniciais do centroide:

Centroides	$X$	$Y$
$C_1$	0,376	0,567
$C_2$	0,619	0,445

	Gols	Assistências
Neymar	0,376	0,567
Lewandowski	0,619	0,445
Cristiano Ronaldo	0,992	0,862
Messi	1,000	1,000



# Análise de agrupamentos

- K-means

3. Cálculo da distância entre o centroide e as amostras (Matriz de Distância):

Centroides	X	Y
$C_1$	0,376	0,567
$C_2$	0,619	0,445

	Gols	Assistências
Neymar (J1)	0,376	0,567
Lewandowski (J2)	0,619	0,445
Cristiano Ronaldo (J3)	0,992	0,862
Messi (J4)	1,000	1,000

Exemplo:

- **Distância Euclidiana** do J3 (Cristiano Ronaldo) até a centroide  $C_1$  e  $C_2$  :

$$DJ3_{C_1} = \sqrt{(0,992 - 0,376)^2 + (0,862 - 0,567)^2}$$

$$DJ3_{C_1} = 0,683$$

$$DJ3_{C_2} = \sqrt{(0,992 - 0,619)^2 + (0,862 - 0,445)^2}$$

$$DJ3_{C_2} = 0,560$$

Matriz de Distância

Distância do centroides com os Jogadores:				
	J1	J2	J3	J4
$C_1$	0	0,269	0,683	0,759
$C_2$	0,272	0	0,560	0,673

# Análise de agrupamentos

- K-means
  4. Criando a Matriz de Grupos
    - Observando-se a Matriz de Distâncias, atribui-se o valor 1 na Matriz de Grupos à posição de menor distância de cada objeto.

Matriz de Distância

Distância do centroides com os Jogadores:				
	J1	J2	J3	J4
$C_1$	0	0,269	0,683	0,759
$C_2$	0,272	0	0,560	0,673



Matriz de Grupo

Distância do centroides com os Jogadores:				
	J1	J2	J3	J4
$C_1$	1	0	0	0
$C_2$	0	1	1	1

# Análise de agrupamentos

- K-means

4. Criando a Matriz de Grupos

- Observando-se a Matriz de Distâncias, atribui-se o valor 1 na Matriz de Grupos à posição de menor distância de cada objeto.
- Calcular os novos centroides de acordo com a Matriz de Grupo encontrada:

Matriz de Grupo

Distância do centroides com os Jogadores:				
	J1	J2	J3	J4
$C_1$	1	0	0	0
$C_2$	0	1	1	1

$$C_{1(\text{n\~{a}o alterou})} = (0,376 ; 0,567)$$

	Gols	Assistências
Neymar (J1)	0,376	0,567
Lewandowski (J2)	0,619	0,445
Cristiano Ronaldo (J3)	0,992	0,862
Messi (J4)	1,000	1,000

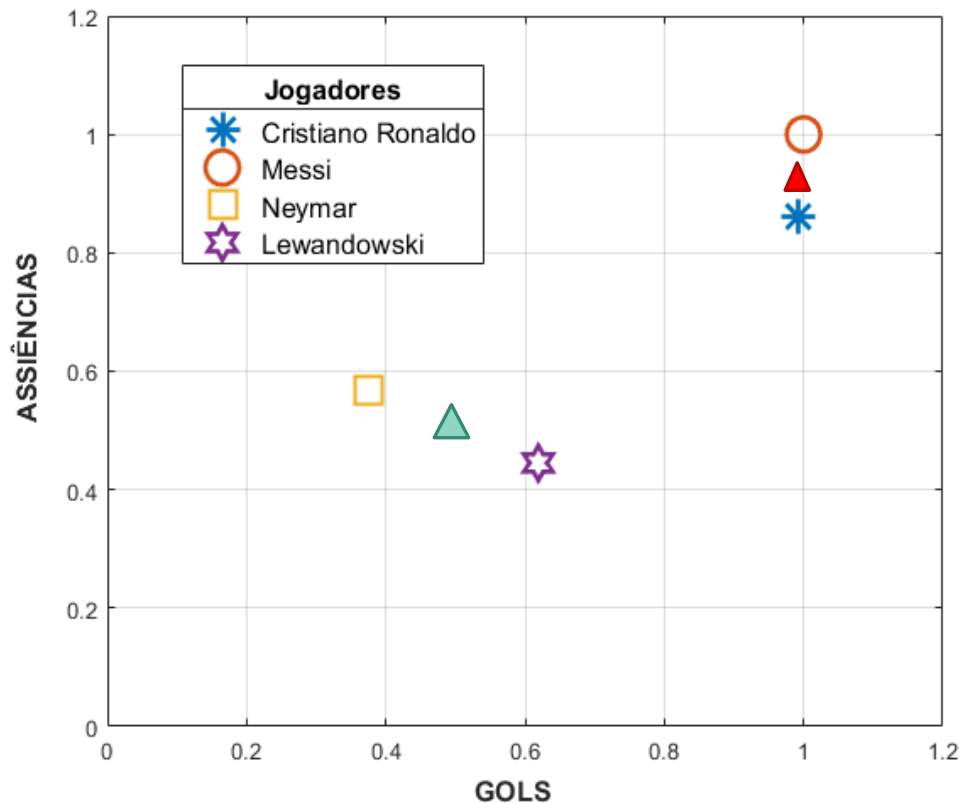
$$C_{2(\text{novo})} = \left( \frac{0,619 + 0,992 + 1}{3} ; \frac{0,445 + 0,862 + 1}{3} \right)$$

$$C_{2(\text{novo})} = (0,870 ; 0,769)$$

# Análise de agrupamentos

- K-means

5. Repetir as etapas 3 e 4 até que não haja alteração entre o centroide anterior e o calculado.



Centroides Finais	X	Y
$C_1(Final)$	0,497	0,506
$C_2(Final)$	0,996	0,931

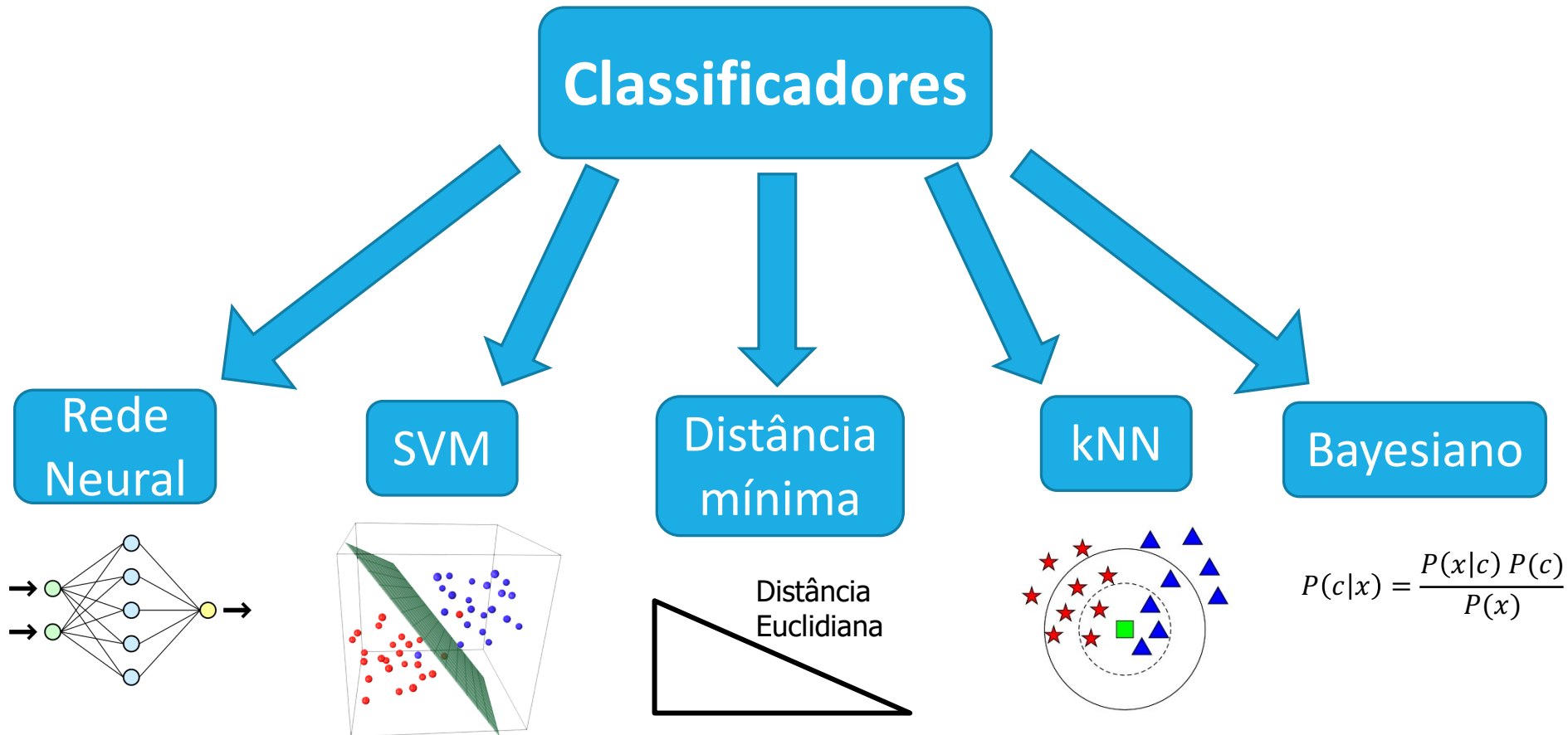
# Classificador

- No momento em que as amostras são agrupadas em classes (tanto pelo método supervisionado quanto pelo não supervisionado), o cálculo do vetor protótipo deve ser feito;
- Com os vetores protótipos é possível fazer o reconhecimento e classificação de uma nova amostra em uma das classes aprendidas.



# Classificador

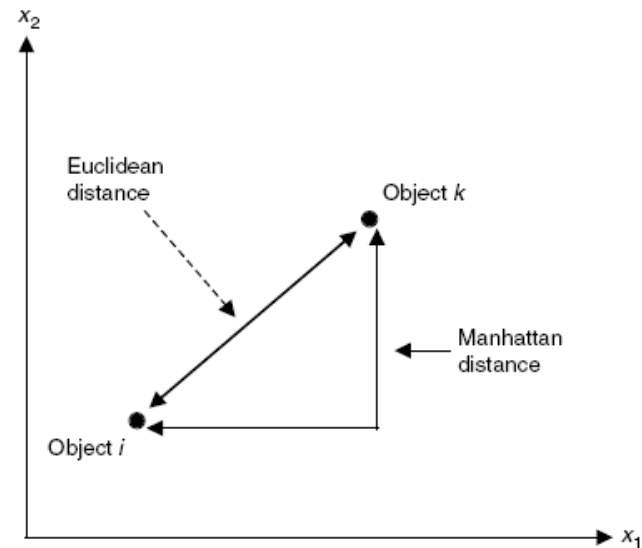
- Existem diversos tipos de classificadores



# Classificadores de distância mínima

- Tem como objetivo determinar se um Vetor característica desconhecido se assemelha a algum protótipo analisado;
- O classificador de distância é uma técnica que analisa de distância dos pixels da imagem como determinada classe já estabelecida;
- Classificadores de distância mais utilizados:

- *Distância City Block*
- *Distância Euclidiana*
- *Distância Chessboard*



# Classificadores de distância mínima

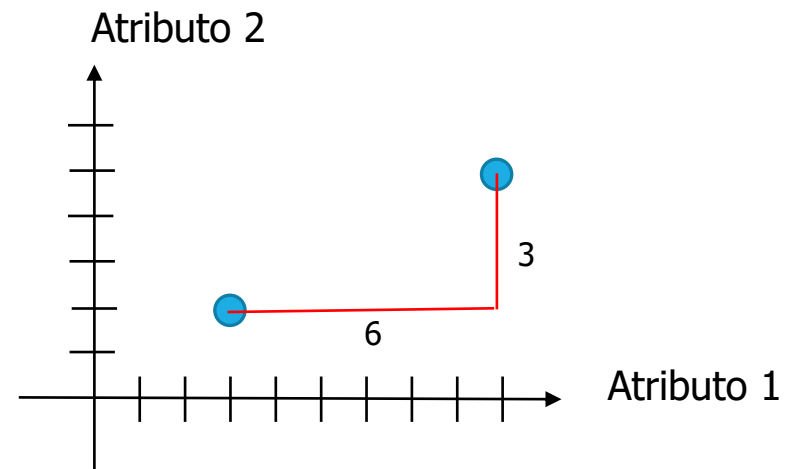
- Distância *City Block* (norma 1 ou  $L_1$ ):
  - A *distância City Block* (ou *Manhattan*) é uma medida de dissimilaridade que avalia a distância absoluta entre dois vetores.

$$DM_{ik} = \sum_{j=1}^p |x_{ij} - x_{kj}|$$

Onde  $DM_{ik}$  é distância Manhattan entre o Vetor característica  $i$  e o Vetor característica  $k$

	Amostra 1	Amostra 2
Atributo 1	3	9
Atributo 2	2	5

$$\text{Distância City Block} = |3 - 9| + |2 - 5| = 9$$



# Classificadores de distância mínima

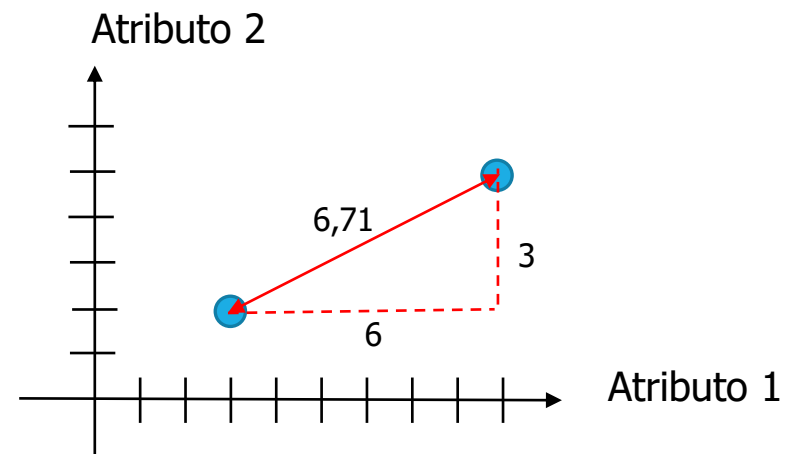
- Distância Euclidiana (norma 2 ou  $L_2$ ):
  - Em matemática, distância euclidiana é a distância entre dois pontos, que pode ser provada pela aplicação repetida do teorema de Pitágoras. Aplicando essa fórmula como distância, o espaço euclidiano torna-se um espaço métrico utilizado para classificação de distância entre dois objetos:

$$DE_{ik} = \sqrt{\sum_{j=1}^p (x_{ij} - x_{kj})^2}$$

Onde  $DE_{ik}$  é distância Euclidiana entre o Vetor característica  $i$  e o Vetor característica  $k$

	Amostra 1	Amostra 2
Atributo 1	3	9
Atributo 2	2	5

$$\text{Distância Euclidiana} = \sqrt{(3 - 9)^2 + (2 - 5)^2}$$



# Classificadores de distância mínima

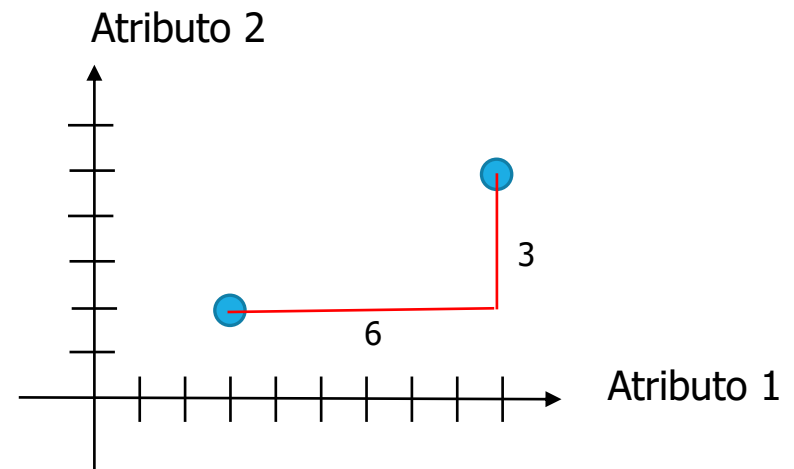
- Distância Chessboard (ou Chebyshev)
  - A distância Chessboard é uma métrica definida em um espaço de vetores onde a distância entre dois vetores é a maior de suas diferenças entre suas dimensões de coordenadas.

$$DCh_{ik} = \max(|x_i - x_k|)$$

	Amostra 1	Amostra 2
Atributo 1	3	9
Atributo 2	2	5

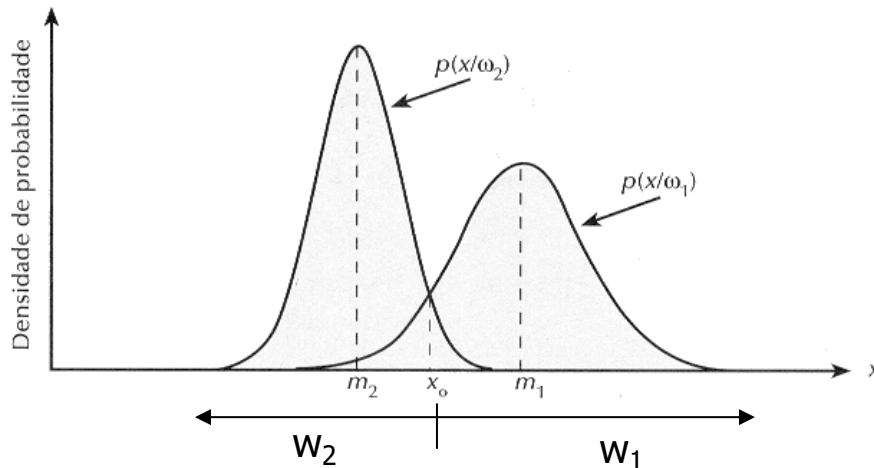
$$\text{Distância Chessboard} = \max(6, 3) = 6$$

Onde  $DCh_{ik}$  é distância Chessboard entre o Vetor característica  $i$  e o Vetor característica  $k$



# Classificadores estatísticos

- Distância Bayesiano:
  - Trabalha com a probabilidade um vetor característica pertencer a uma classe conhecida;
  - Esta métrica faz uso de fórmulas estatísticas e cálculo de probabilidades para realizar a classificação;
  - Bastante utilizados onde o problema abordado envolve duas classes de padrões governados por densidades gaussianas;



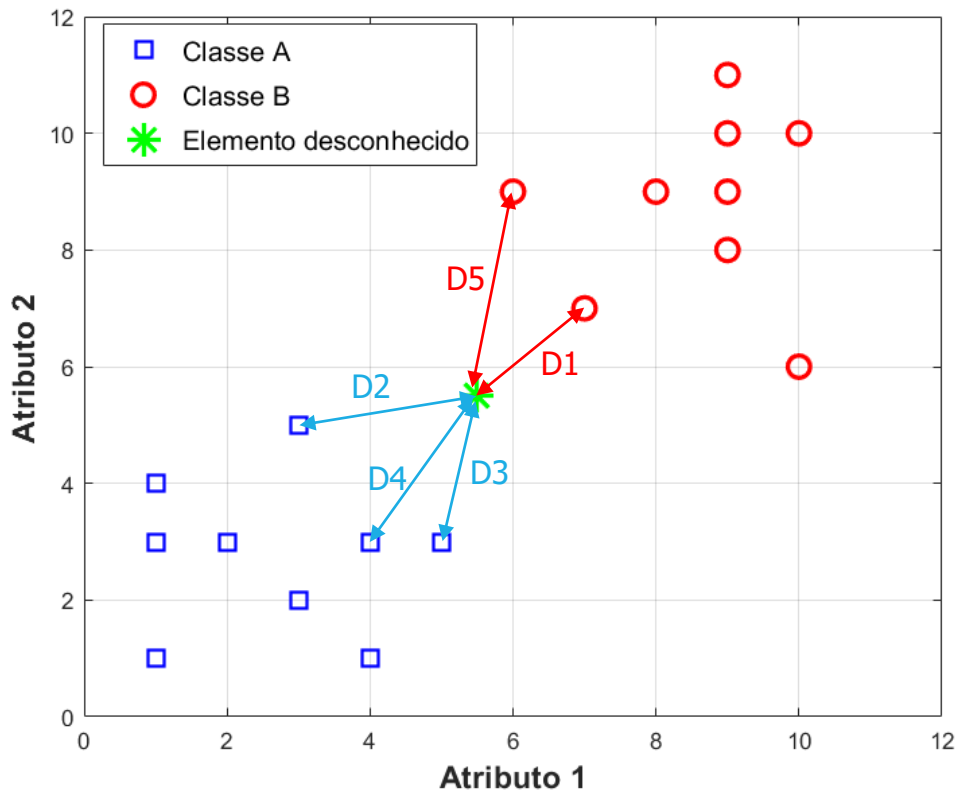
A fronteira entre as classes é o ponto  $x_0$

# Classificador K – Nearest Neighbors (KNN)

- O classificador KNN procura K elementos do conjunto de classes conhecidas que estejam mais próximos deste elemento desconhecido, ou seja, que tenham a menor distância;
- Estes K elementos são chamados de K-vizinhos mais próximos.
- Para a classificação, verifica-se quais são as classes desses K vizinhos e a classe mais frequente será atribuída ao elemento desconhecido.
- A menor distância geralmente é calculado pela distância Euclidiana. No entanto, pode-se usar outras métricas (distância Manhattan ou Chebyshev).

# Classificador K – Nearest Neighbors (KNN)

- Exemplo para  $K = 5$



**Qual classe o novo elemento deve ser classificado?**

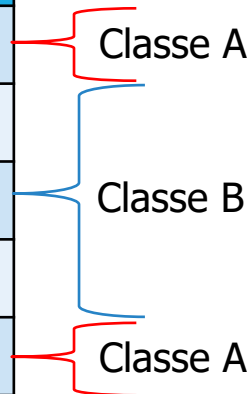
Distância das amostras mais próximas ao elemento desconhecido	
D1	2,12
D2	2,55
D3	2,55
D4	2,92
D5	3,53



# Classificador K – Nearest Neighbors (KNN)

- Exemplo para  $K = 5$

Distância das amostras mais próximas ao elemento desconhecido	
D1	2,12
D2	2,55
D3	2,55
D4	2,92
D5	3,53



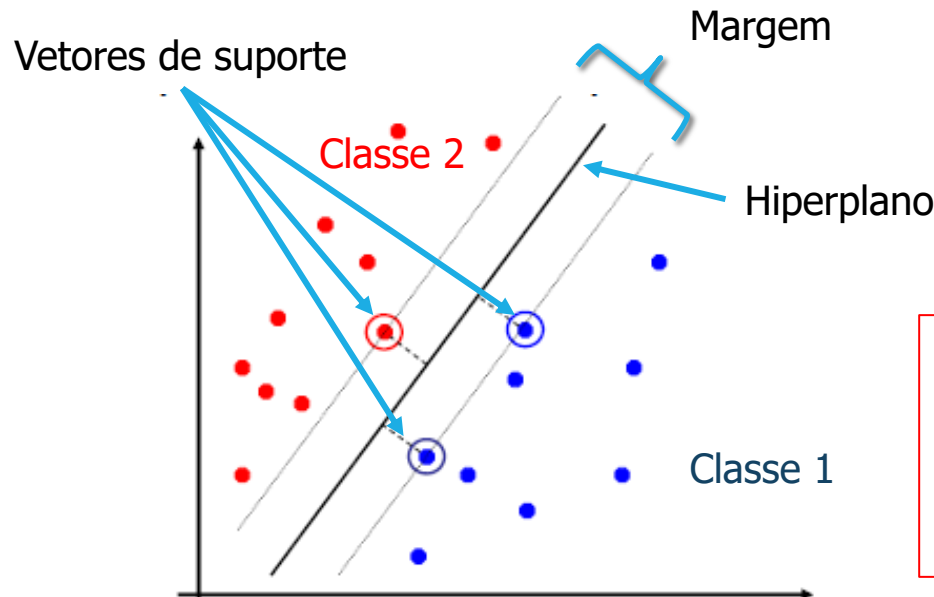
**Qual classe o novo elemento deve ser classificado?**

- Para  $k = 5$ :
  - 2 Classes A próximas;
  - 3 Classes B próximas;
- ✓ Elemento desconhecido atribuído a **Classe B!**

# Classificador por Support Vector Machines

*Support Vector Machines* (SVM) é um algoritmo que busca um hiperplano (em diversas dimensões) ótimo que faz a separação das classes de maneira que entre elas se tenha a maior distância possível.

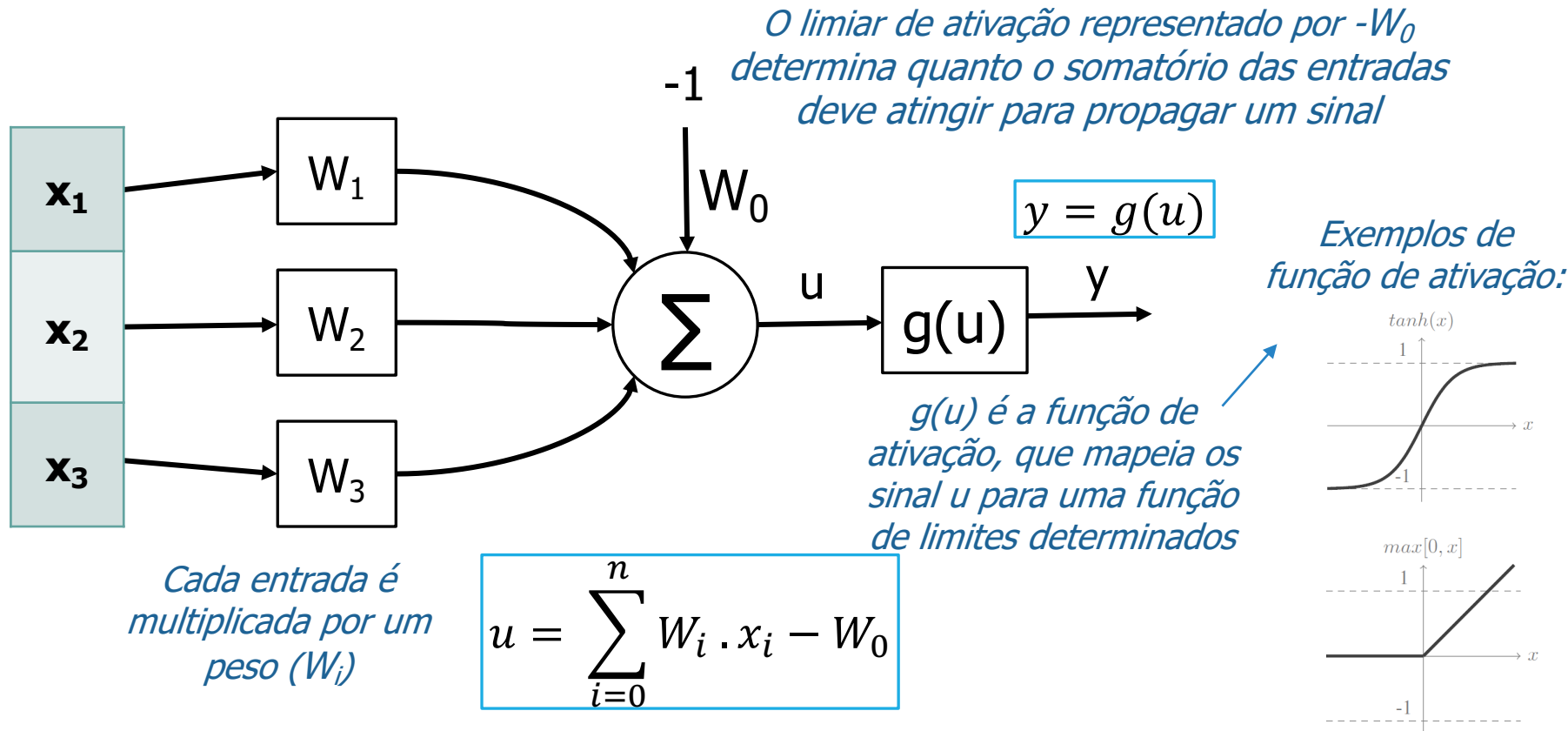
Os **vetores de suporte**, ou seja, aqueles que estão nas extremidades entre uma classe e outra, são os responsáveis por definir as margens.



É um algoritmo mais robusto em relação aos anteriores, como o de distância, pois lidam melhor com *clusters* de grande variância

# Redes Neurais Artificiais

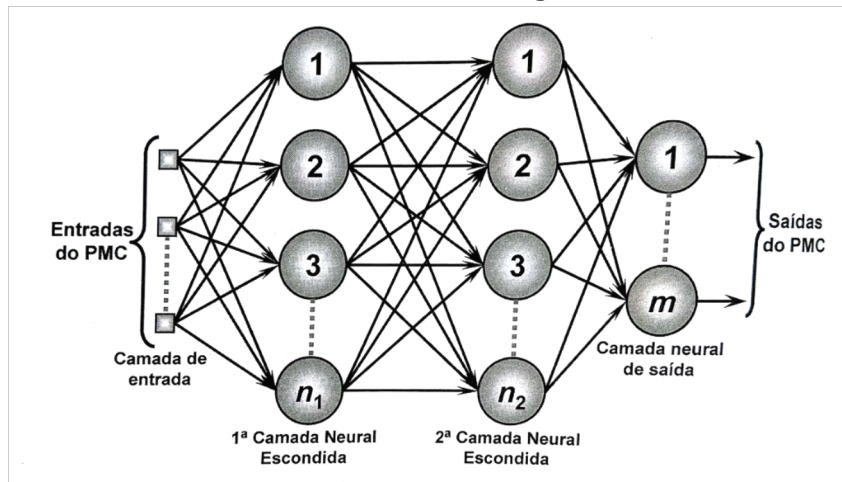
O algoritmo básico que compõe a rede neural artificial é o neurônio artificial, que associados em cadeia permitem realizar separações no espaço de atributos cada vez mais complexas. Sua estrutura é representada como:



# Redes Neurais Artificiais

Durante o treinamento, diversas amostras (vetores de característica) são apresentadas à rede. O erro produzido entre a classe correta e o valor predito pela rede é propagado pelos neurônios, e utilizado para calibrar os valores dos pesos  $W$  de forma a aprender o valor correto a ser entregue.

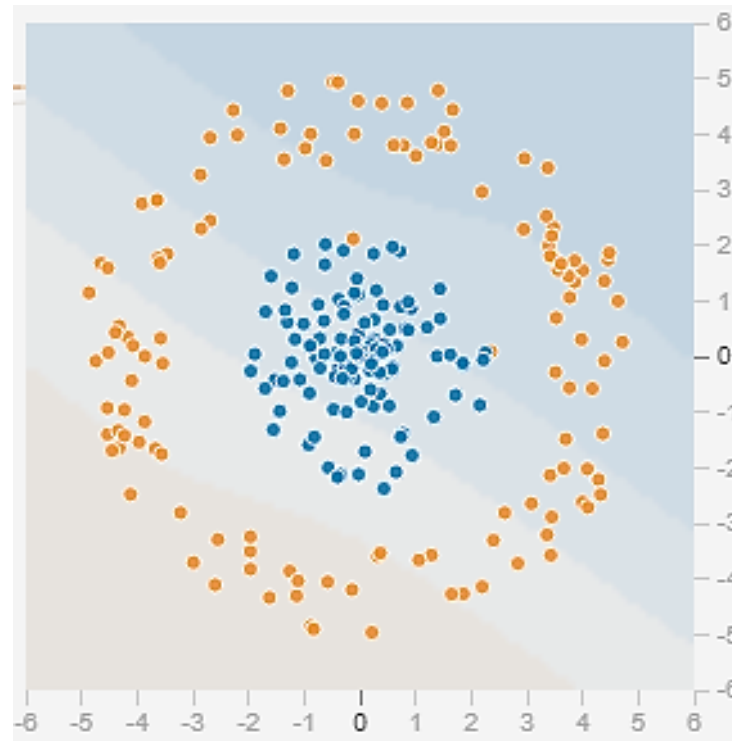
A rede Perceptron Multi-Camadas ou MLP (*Multi Layer Perceptron*) é uma arquitetura com diversas camadas de neurônios e permite a divisão do espaço de atributos de maneira mais complexa. Quanto mais camadas e neurônios mais específica se tornará a classificação.



É necessário balancear a quantidade de neurônios para que a rede seja capaz de generalizar para novas amostras e não fique específica demais em sua tarefa

# Redes Neurais Artificiais

Considerando a seguinte tarefa de classificação:

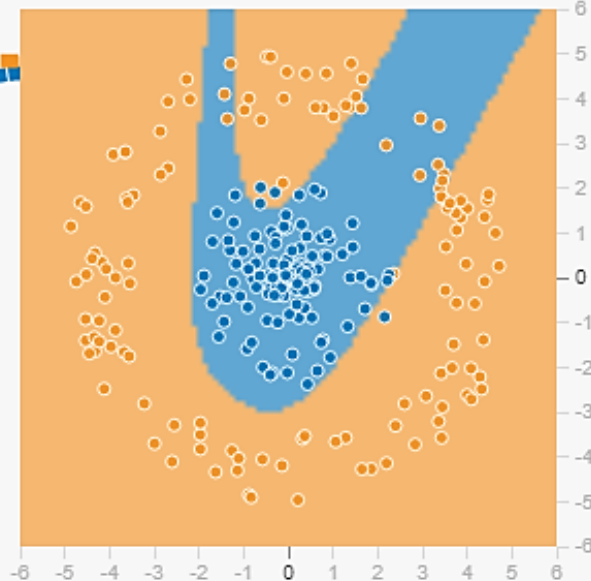
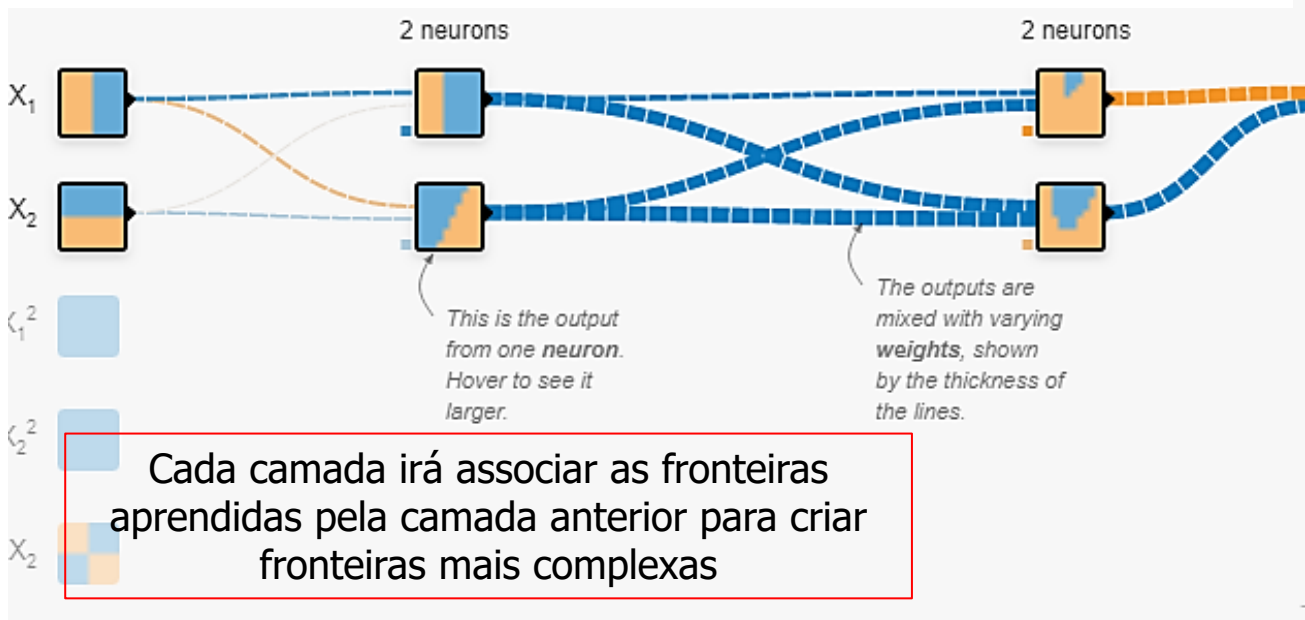


# Redes Neurais Artificiais

Com duas camadas de dois neurônios cada:

É comum o erro de treino ser menor do que o de teste, visto que o algoritmo está otimizando em relação ao conjunto de treino

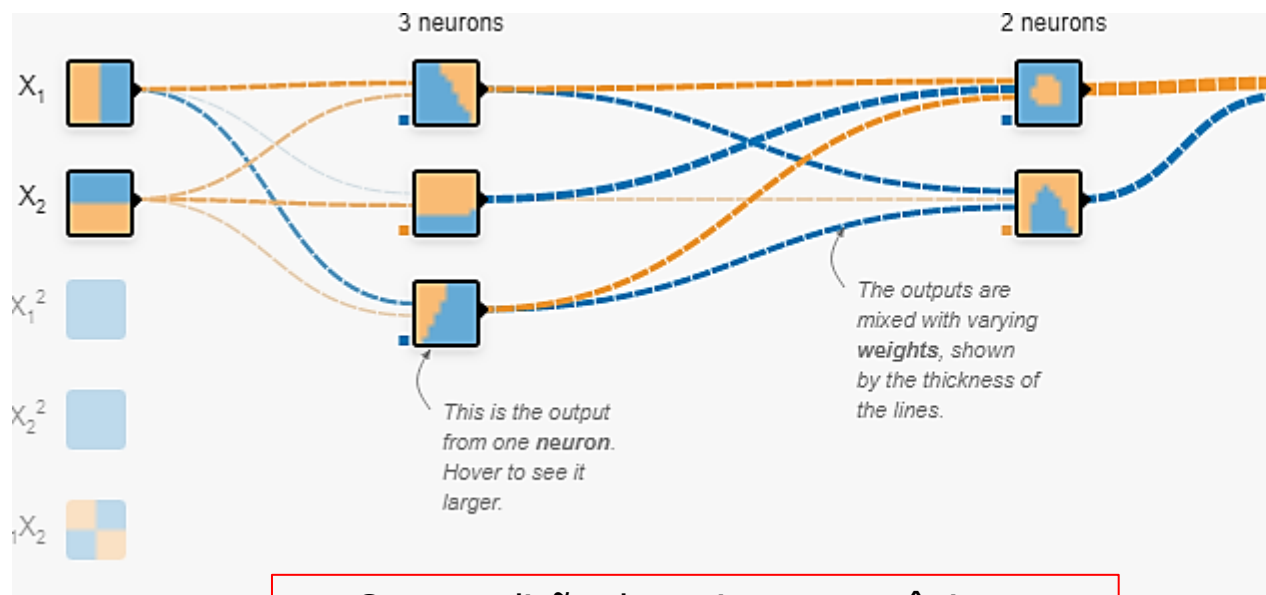
Funções de medida do erro



Com essa configuração as classes ainda não foram devidamente separadas.

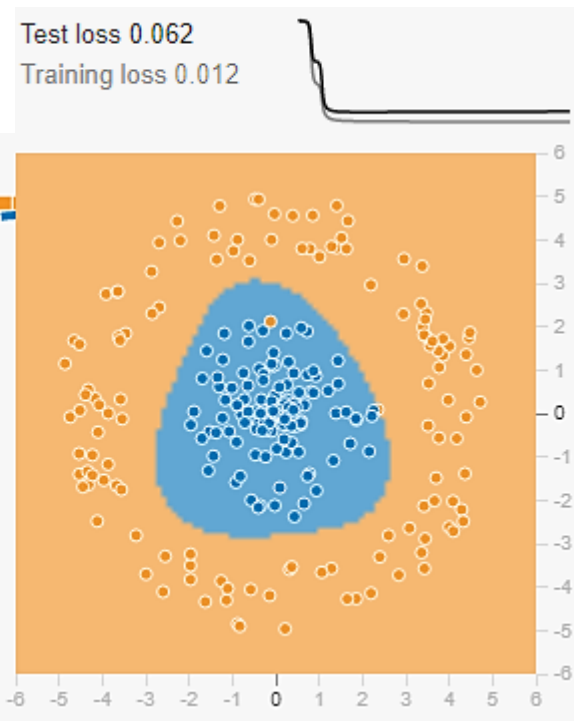
# Redes Neurais Artificiais

Adicionando um neurônio na primeira camada:



Com a adição de mais um neurônio na primeira camada foi possível obter uma melhor separação entre as classes

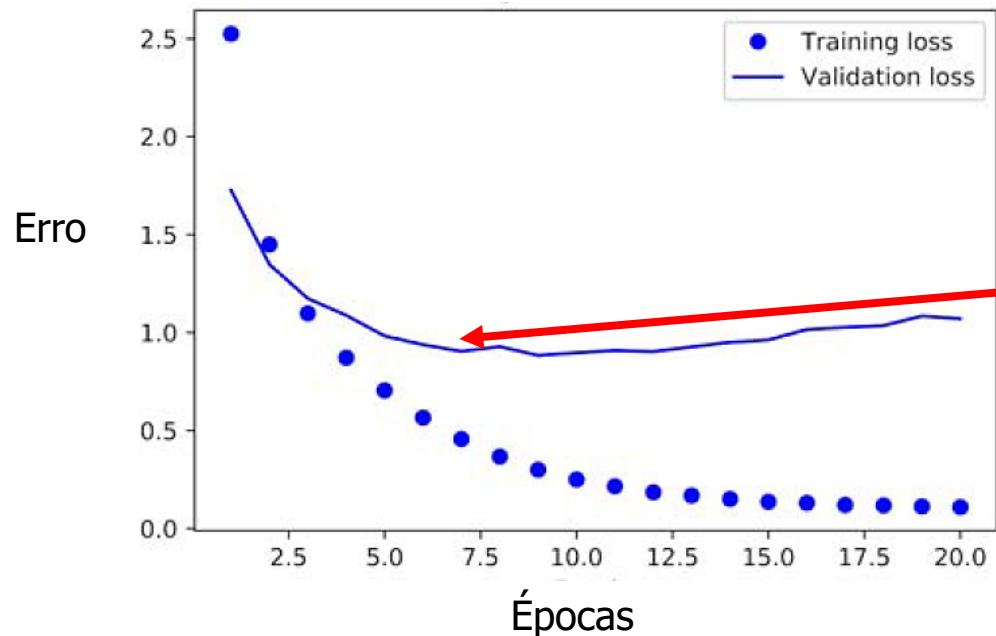
Funções de medida do erro



# Redes Neurais Artificiais

Conjunto de validação é utilizado para detectar quando a rede começa a sobreajustar para as amostras de treinamento (*overfitting*)

Funções de medida do erro de treino e de validação



Erro

Épocas

(quantas vezes todas as amostras de treino são apresentadas para a rede)

Quando o erro da função de custo da validação começa a subir, significa que a rede está sobreajustando para as amostras de treinamento

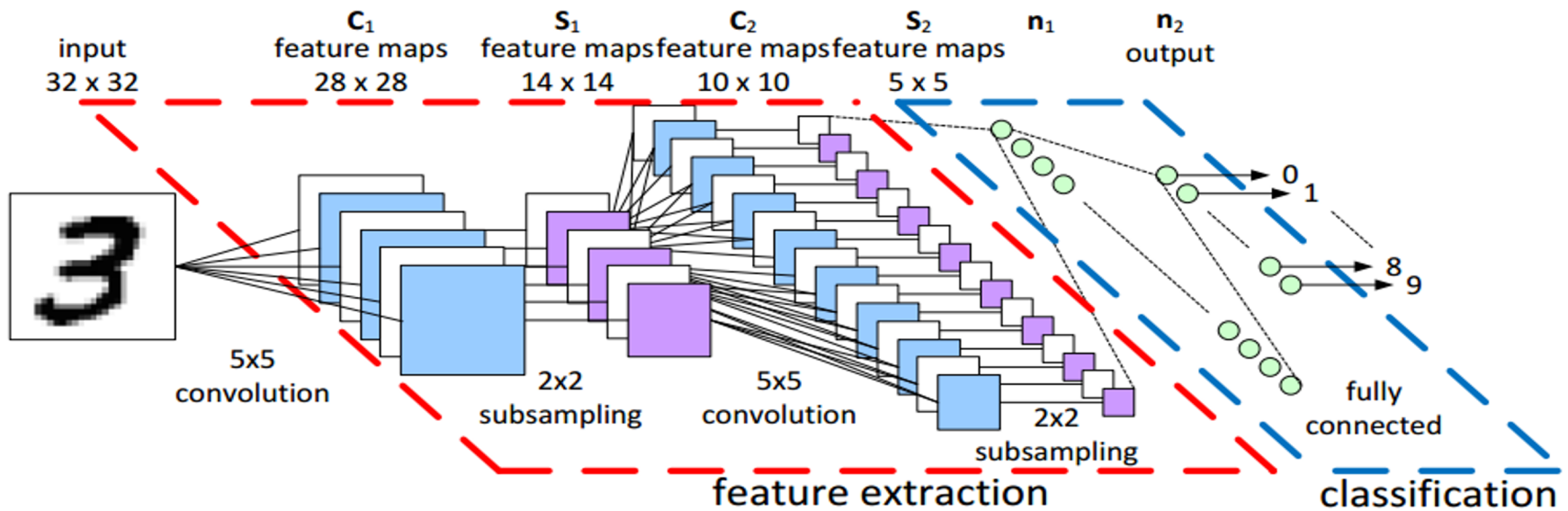


# Redes Neurais Convolucionais

- Redes Neurais Convolucionais (do inglês, CNN) fazem parte do ramo de algoritmos de *Deep Learning* e ganhou mais visibilidade a partir de 2009 quando começou a ganhar várias competições internacionais em reconhecimento de padrões (ImageNet);
- Elas são redes utilizadas para classificação e segmentação em imagens, e utilizam a ideia de cadeias de múltiplas convoluções, que quando associadas conseguem captar padrões nas imagens;
- Necessitam de uma quantidade muito superior de amostras de treinamento para obter um bom desempenho quando comparada a outras redes;

# Redes Neurais Convolucionais

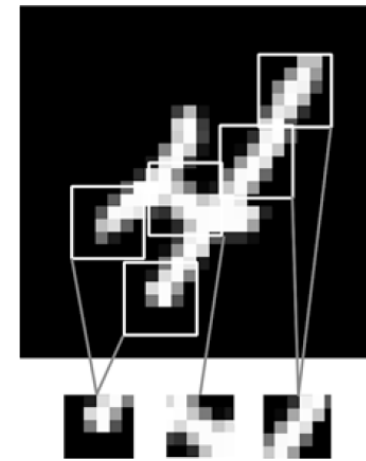
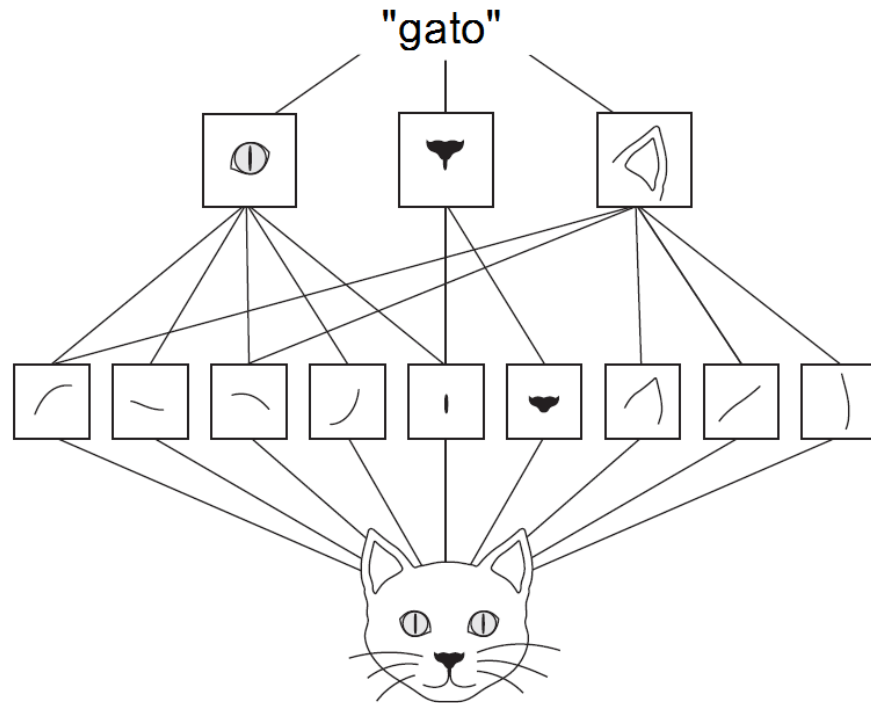
- Existem diversas arquiteturas já consolidadas como:
  - ResNet
  - AlexNet
  - Inception
  - MobileNet



Arquitetura de uma rede CNN para classificação de dígitos [1]

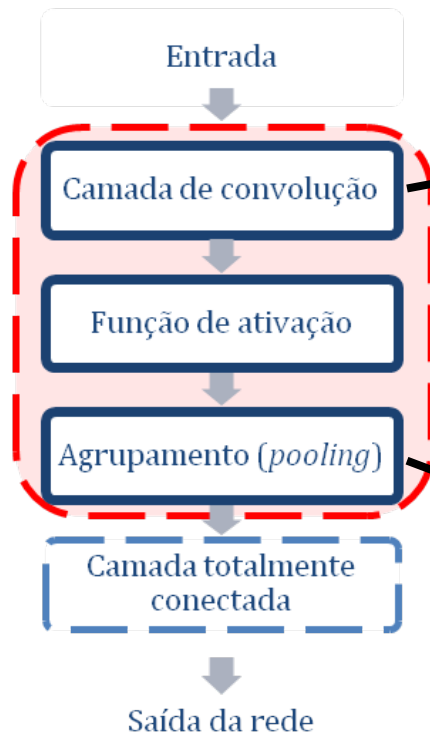
# Redes Neurais Convolucionais

- Alto nível de abstração
- Detectam padrões na imagem de maneira hierárquica



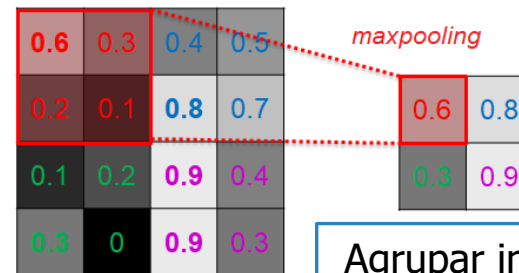
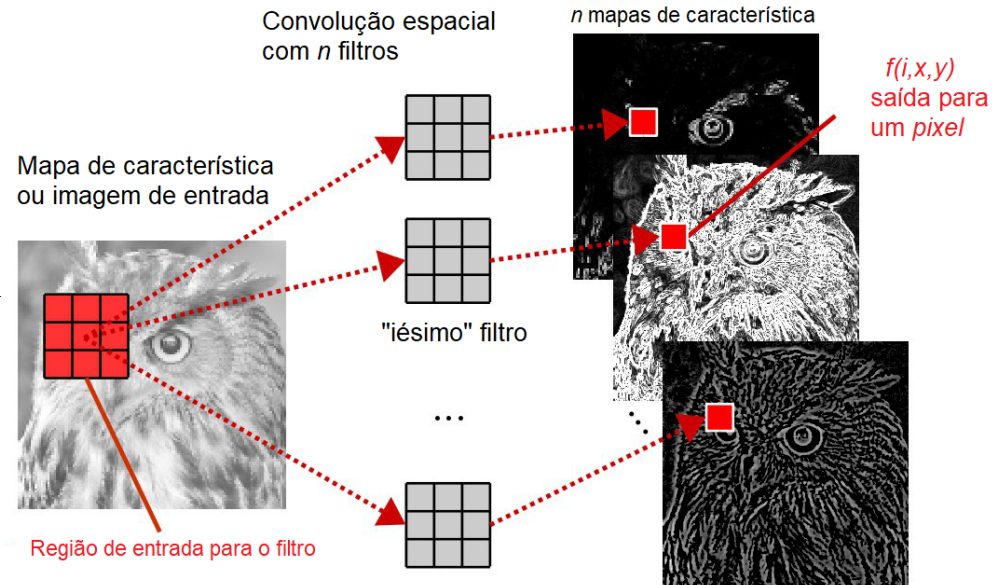
# Redes Neurais Convolucionais

- Estrutura básica:



Estrutura replicada

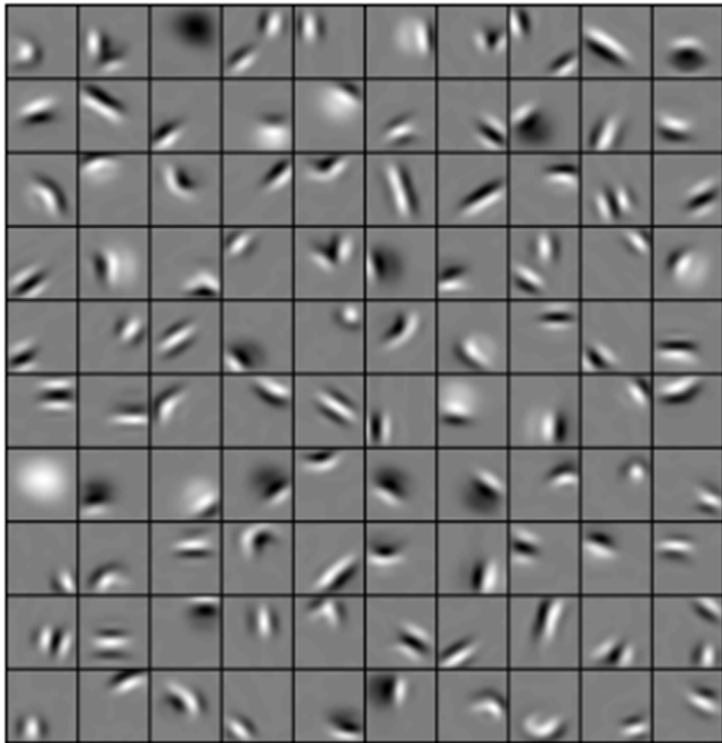
Os filtros são aprendidos durante o processo de treinamento, extraíndo o que é relevante para a classificação



Agrupar informações mais relevantes

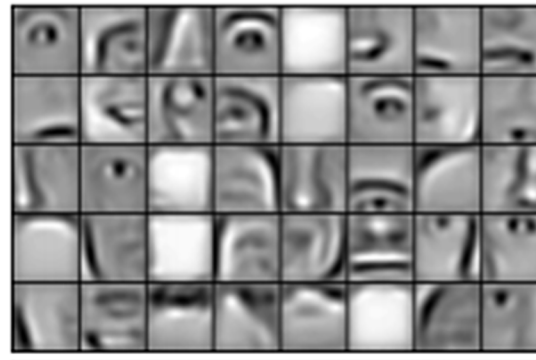
# Redes Neurais Convolucionais

Exemplos de filtros aprendidos na primeira camada

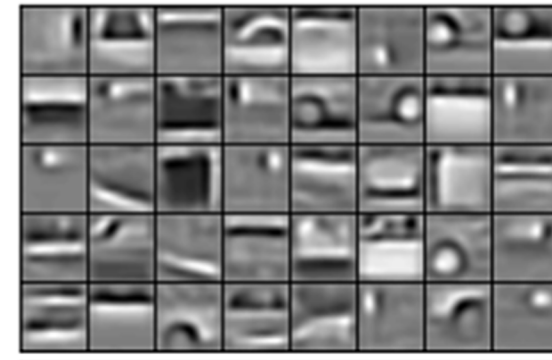


Exemplos de filtros aprendidos na segunda camada para tarefas específicas: faces(esq.) e carros(dir.)

faces



cars

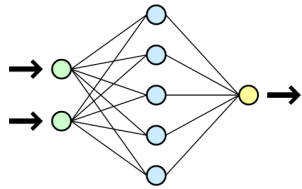


# Classificadores

- Tarefas mais complexas necessitam classificadores mais complexos;
- Se os descritores selecionados são suficientes para separar as classes por uma reta, plano ou hiperplano, um classificador de distâncias é suficiente na etapa de classificação;
- Quando a tarefa de separação é mais complexa, deve-se optar por uma rede neural;
- Em geral, redes neurais necessitam de uma quantidade significativa de amostras na etapa de treinamento;

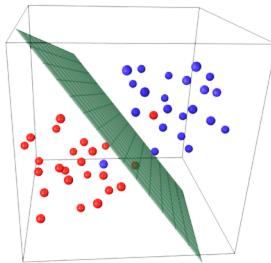
# Resumo - Classificadores

## Rede Neural



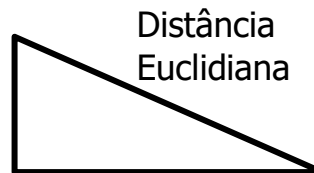
- a. Grande quantidade de dados disponíveis.
- b. Baixa interpretabilidade de dos critérios.
- c. Lento para treinar, rápido para avaliar.

## SVM



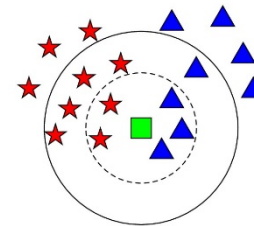
- a. Não necessita de grande número de dados
- b. Baixa interpretabilidade de dos critérios.
- c. Pode lidar com dados não linearmente separáveis

## Distância mínima



- a. Não necessita de grande número de dados
- b. Lida com dados linearmente separáveis
- c. Lento para avaliar

## kNN



- a. Não necessita de grande número de dados
- b. Lento para avaliar

## Bayesiano

$$P(c|x) = \frac{P(x|c) P(c)}{P(x)}$$

- a. Bom para classificação de textos

FIM