



ELDORADO

# *Trilha Python*

## Aula 13

# *Chamada*



**Faísca**

# Atividades de hoje

## Introdução

- ◆ Definição
- ◆ Exemplo de aplicações
- ◆ Como funciona os algoritmos supervisionados

## Algoritmo KNN

## Avaliação de desempenho

- ◆ Matriz de confusão
- ◆ Validação do conjunto de teste



**Mundo 3**

Fortaleza das IAs

**Mundo 2**

Labirinto das Ilusões

Gráficas

**Mundo 1**

Planicie da Pythonlândia



# Boss Final

# Projeto Final



# *Introdução*

# Classificação: Definição

Dado um conjunto de registros (**conjunto de treino – usados para treinar o modelo =D** )

- ◆ Cada registro contém um conjunto de atributos, e um dos atributos é o atributo classe.

In [3]: `zoo.head()`

Out[3]:

	animal	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed	backbone	breathes	venomous	fins	legs	tail	domestic	catsize	type
0	aardvark	1	0	0	1	0	0	1	1	1	1	0	0	4	0	0	1	mammal
1	antelope	1	0	0	1	0	0	0	0	1	1	1	0	0	4	1	0	mammal
2	bass	0	0	1	0	0	1	1	1	1	0	0	0	1	0	1	0	fish
3	bear	1	0	0	1	0	0	0	1	1	1	1	0	0	4	0	0	mammal
4	boar	1	0	0	1	0	0	0	1	1	1	1	0	0	4	1	0	mammal

In [3]:

```
1 #Separa o conjunto de dados em treino e teste
2 #Eliminate the first and last attributes
3 X = zoo.iloc[:, 1:-1].values
4 #Take the last attribute as a target
5 y = zoo.iloc[:, -1].values
6 print(X)
```

```
[[1 0 0 ... 0 0 1]
 [1 0 0 ... 1 0 1]
 [0 0 1 ... 1 0 0]
 ...
 [1 0 0 ... 1 0 1]
 [0 0 1 ... 0 0 0]
 [0 1 1 ... 1 0 0]]
```

## ***Classificação: Definição***

**Objetivo:** aos registros previamente desconhecidos devem ser atribuídas classes, com a maior precisão possível.

- ◆ O conjunto de registros inicial é dividido em conjuntos de **treino** e **teste** (Não colocamos a classes, o Y, eu quero que ele descubra – **inferindo** uma classe).
- ◆ O conjunto de treino é usado para construir o modelo e, o de teste, para validá-lo.
- ◆ Um conjunto de teste é usado para determinar a precisão do modelo.

# Classificação: Aplicação

AlphaGo - Lee Sedol



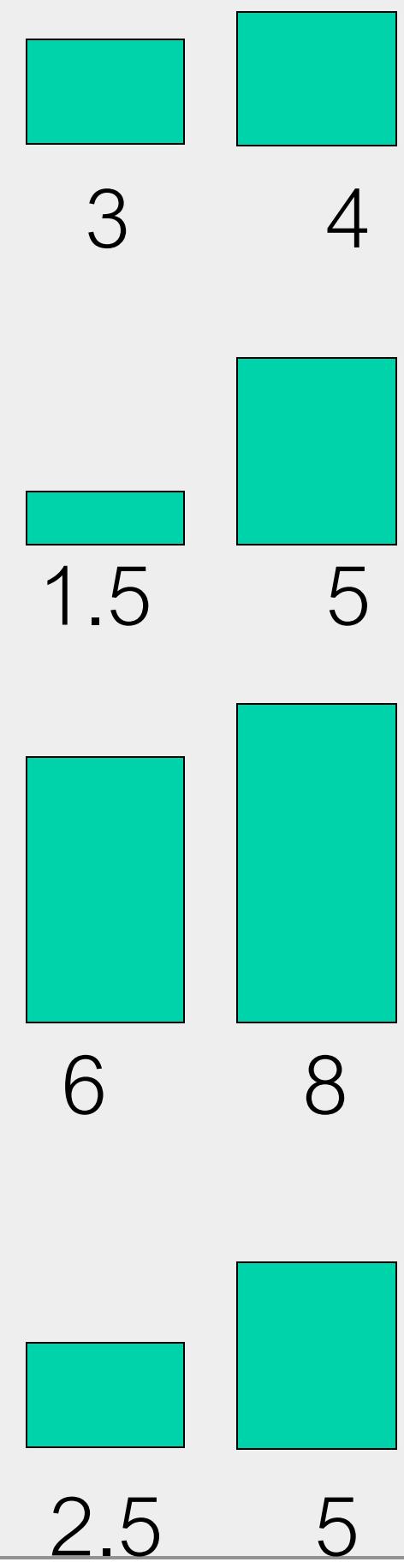
Máquina ganhou a partida por 4 X 1.

# **Classificação: Aplicação**

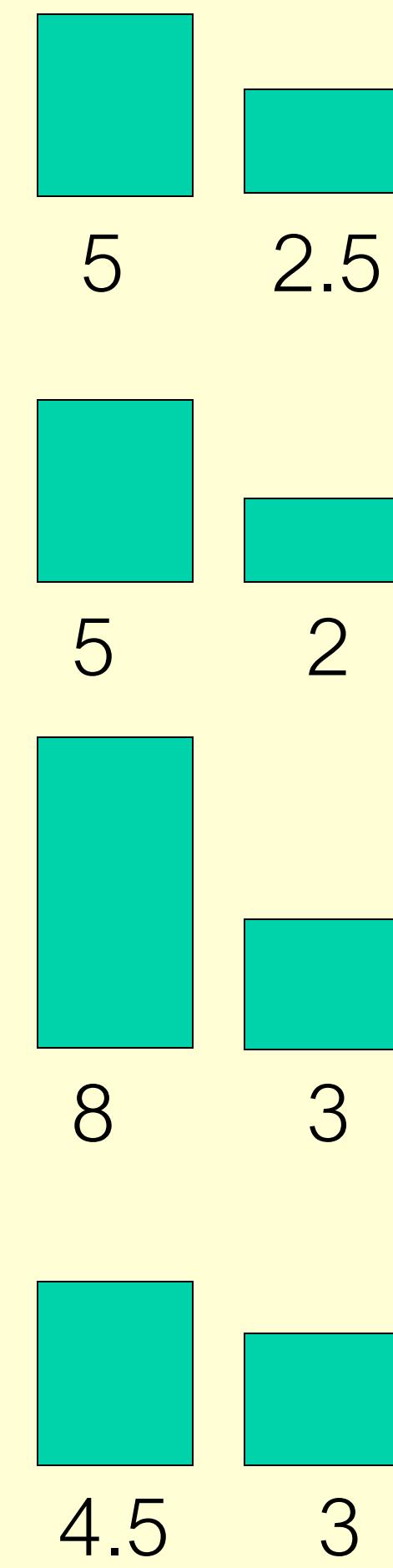


# **Classificação: Problema do Pombo**

Exemplos da classe  
A

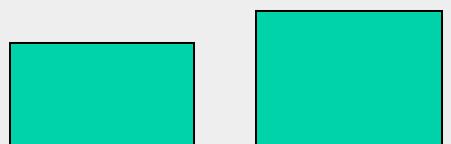


Exemplos da  
classe B

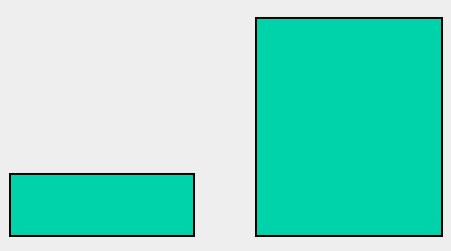


# Classificação: Problema do Pombo

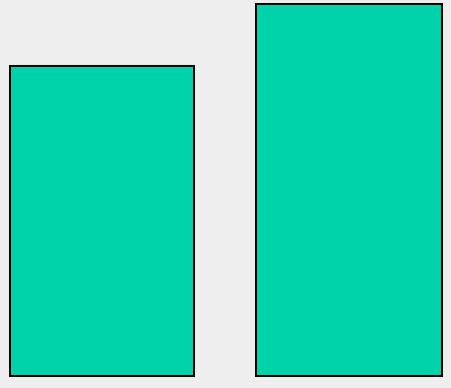
Exemplos da classe  
A



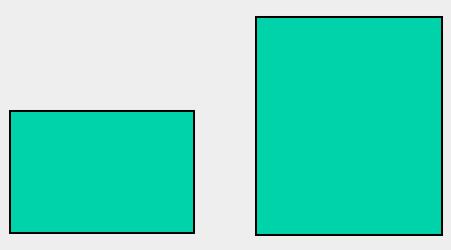
3      4



1.5      5

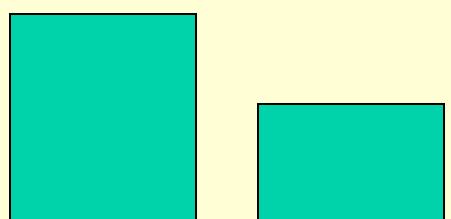


6      8

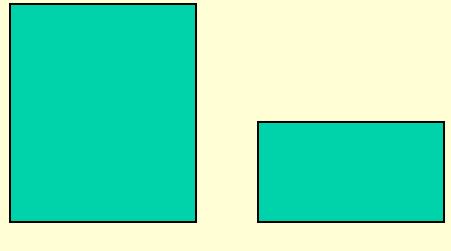


2.5      5

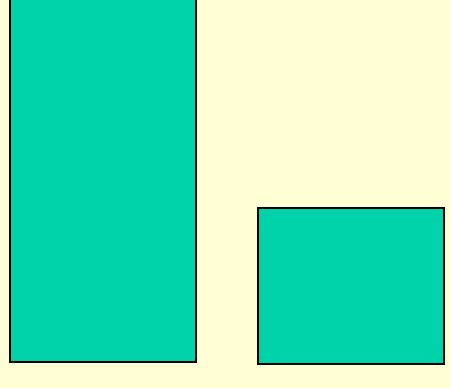
Exemplos da  
classe B



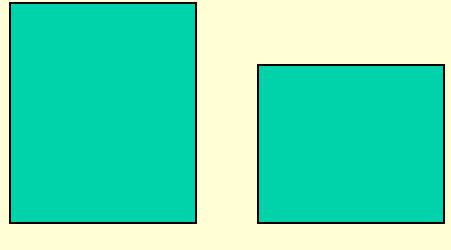
5      2.5



5      2



8      3



4.5      3

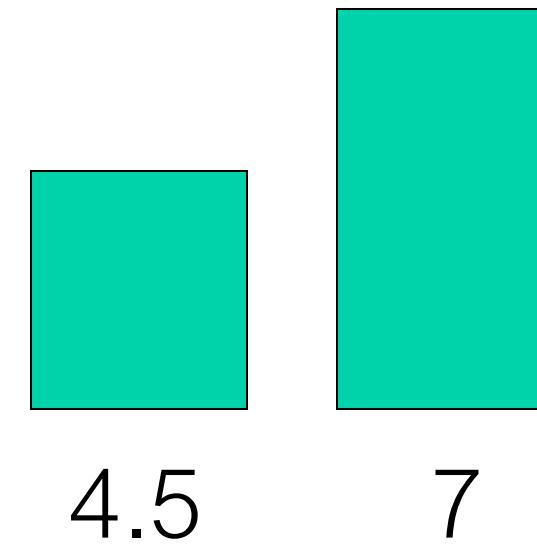


De qual classe é  
este objeto?

8      1.5

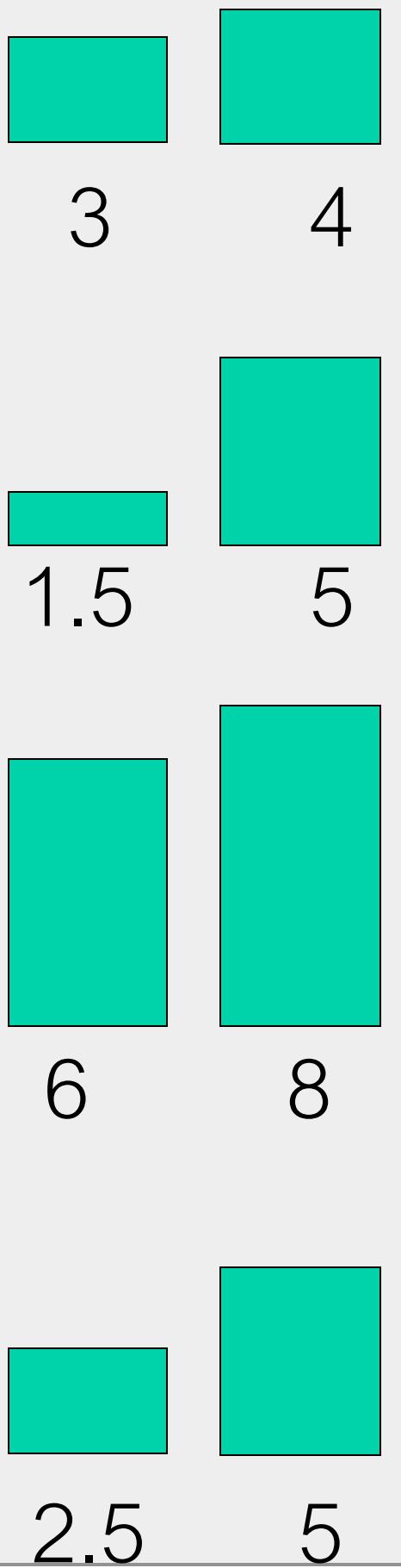


Que tal este, A  
ou B?

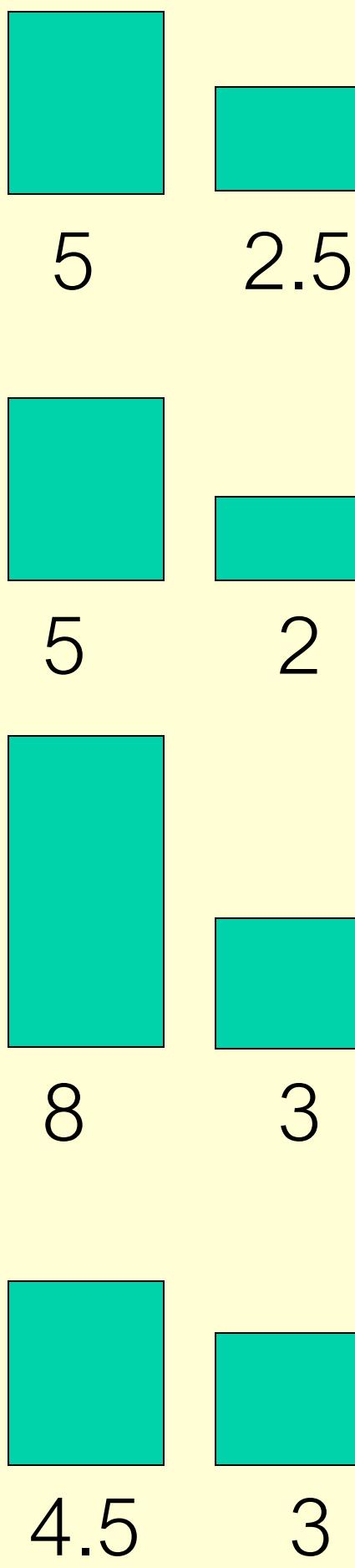


# Classificação: Problema do Pombo

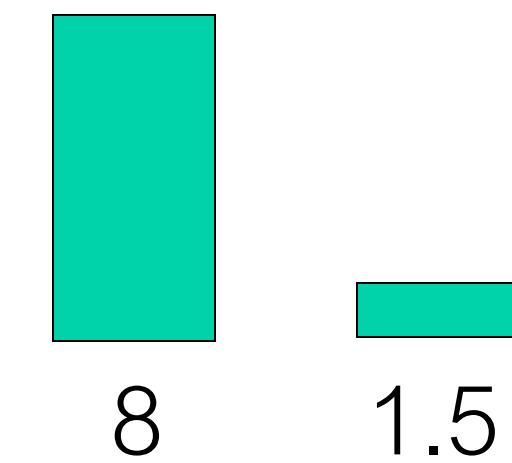
Exemplos da classe  
A



Exemplos da  
classe B



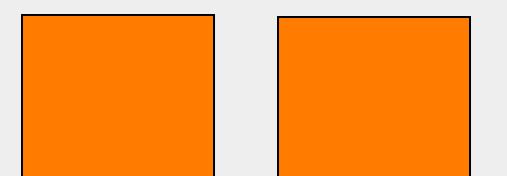
Este é um B!



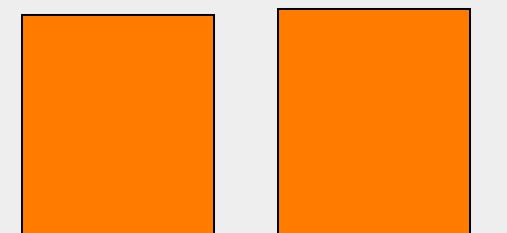
Regra: se a barra  
esquerda é menor que a  
direita, é um A, caso  
contrário é um B.

# Classificação: Problema do Pombo 2

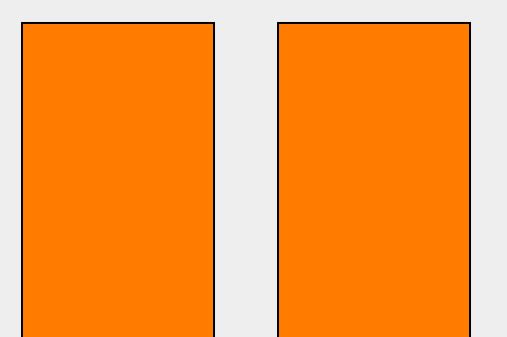
Exemplos da  
classe A



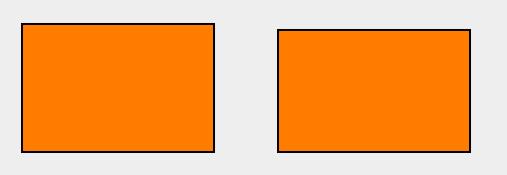
4      4



5      5

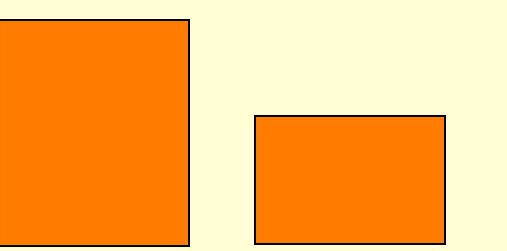


6      6

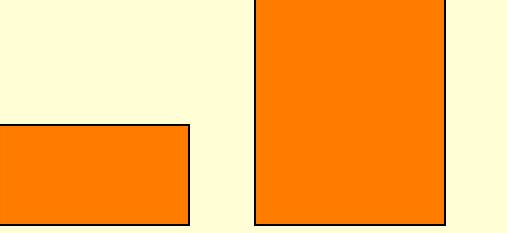


3      3

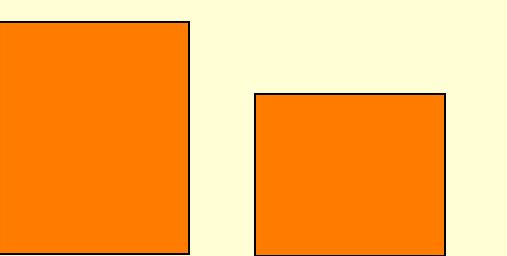
Exemplos da  
classe B



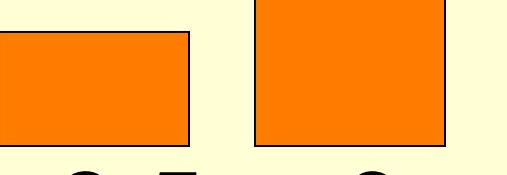
5      2.5



2      5



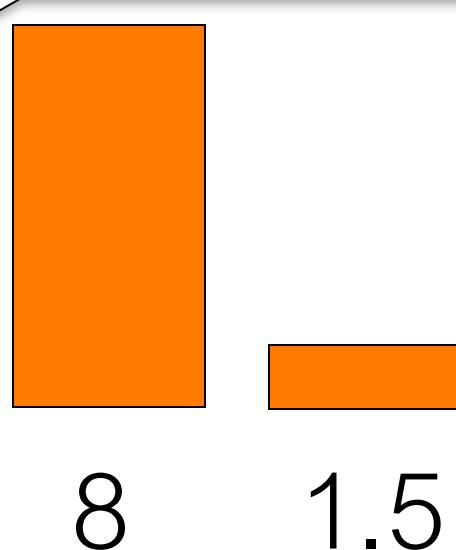
5      3



2.5    3



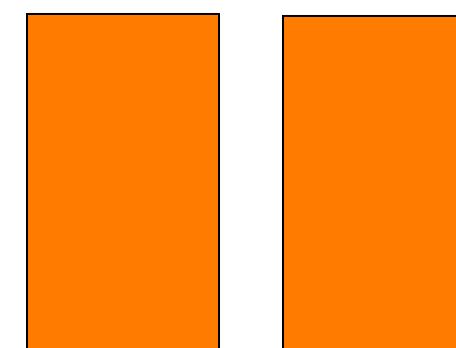
Oh! Este aqui é  
difícil!



8      1.5



Até eu sei este!

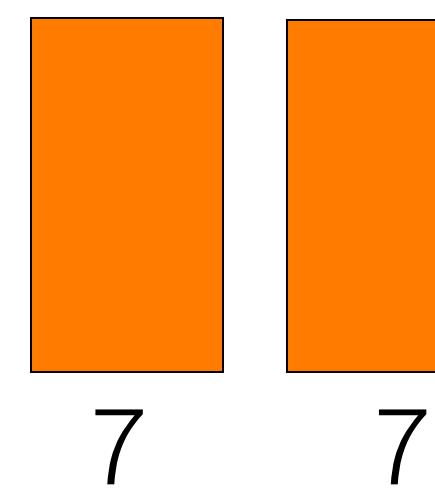


7      7

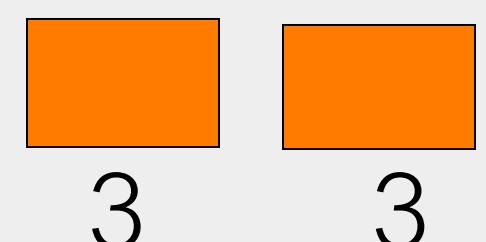
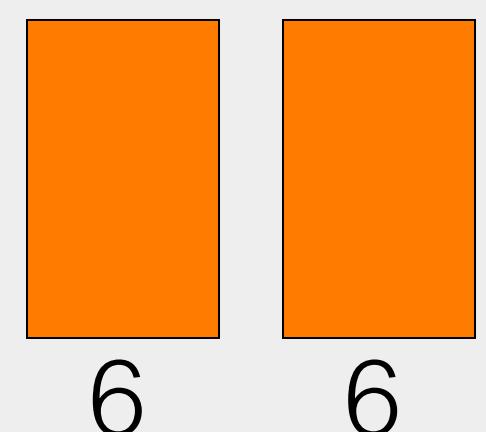
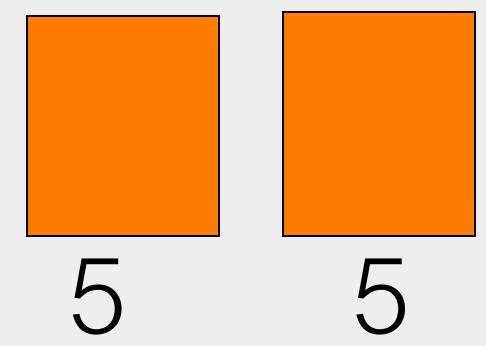
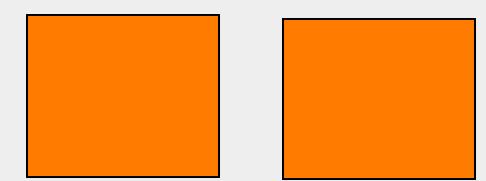
# Classificação: Problema do Pombo 2



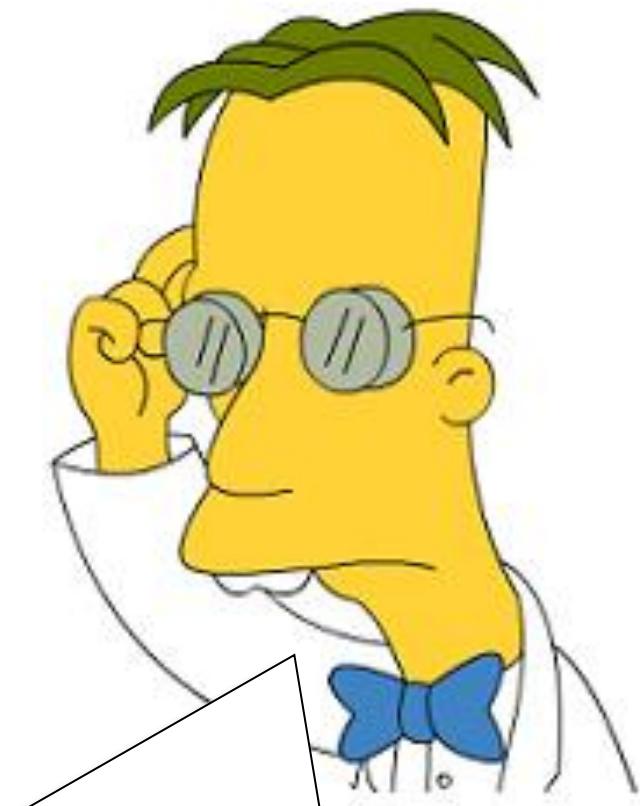
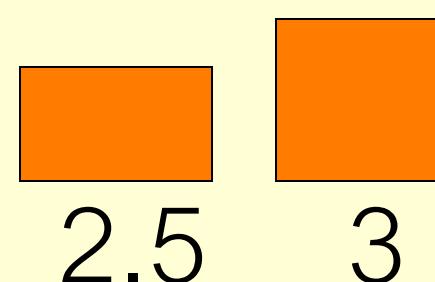
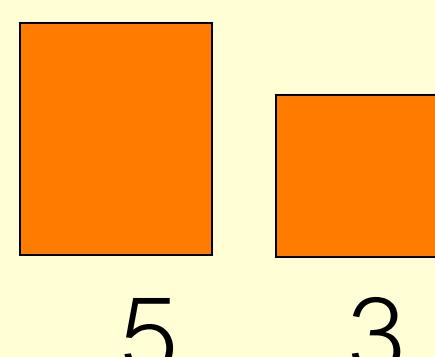
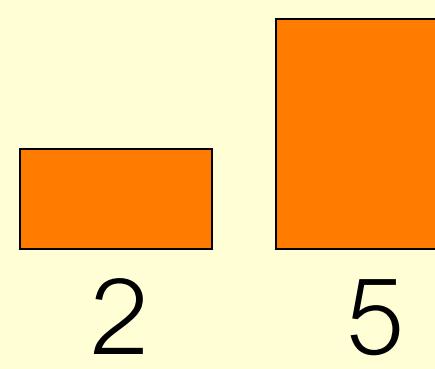
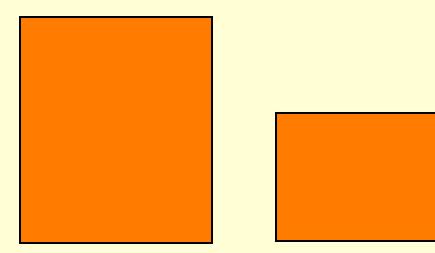
Então este é um  
**A.**



Exemplos da  
classe A



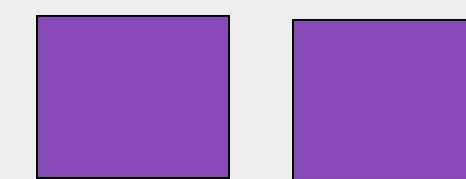
Exemplos da  
classe B



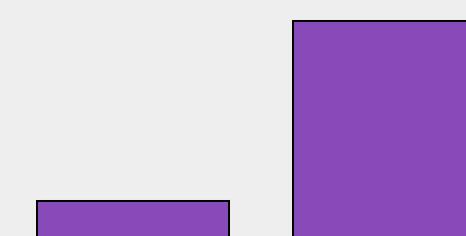
A regra é: se duas barras  
são iguais em tamanho é  
um **A**. Caso contrário é um  
**B.**

# Classificação: Problema do Pombo 3

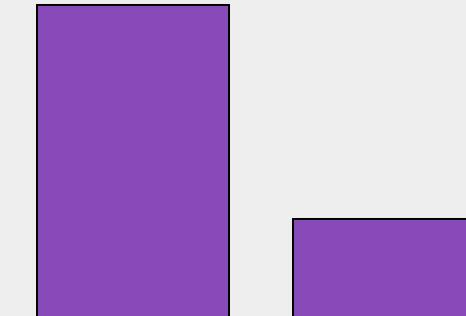
Exemplos da  
classe A



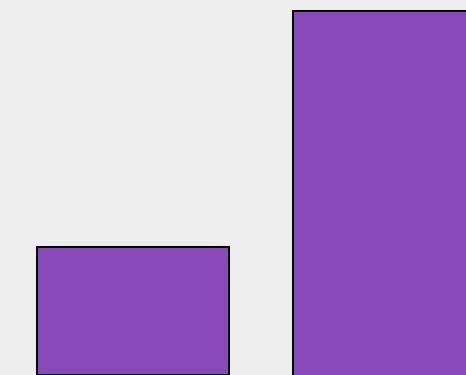
4      4



1      5

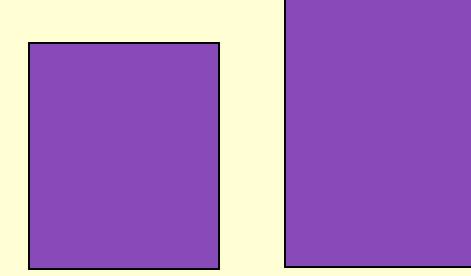


6      3

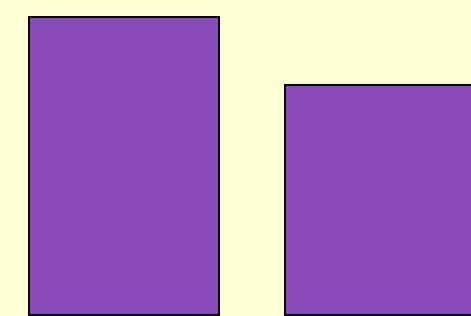


3      7

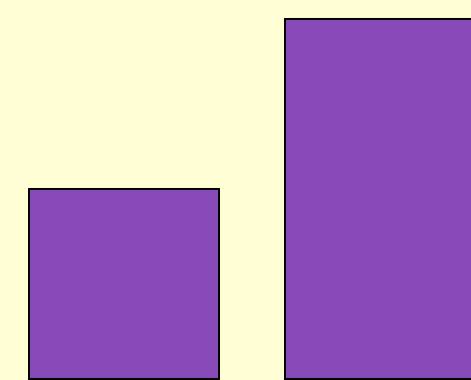
Exemplos da  
classe B



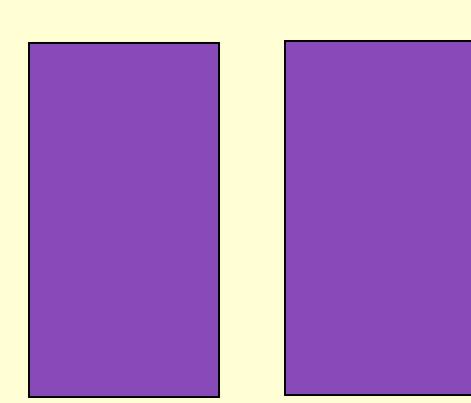
5      6



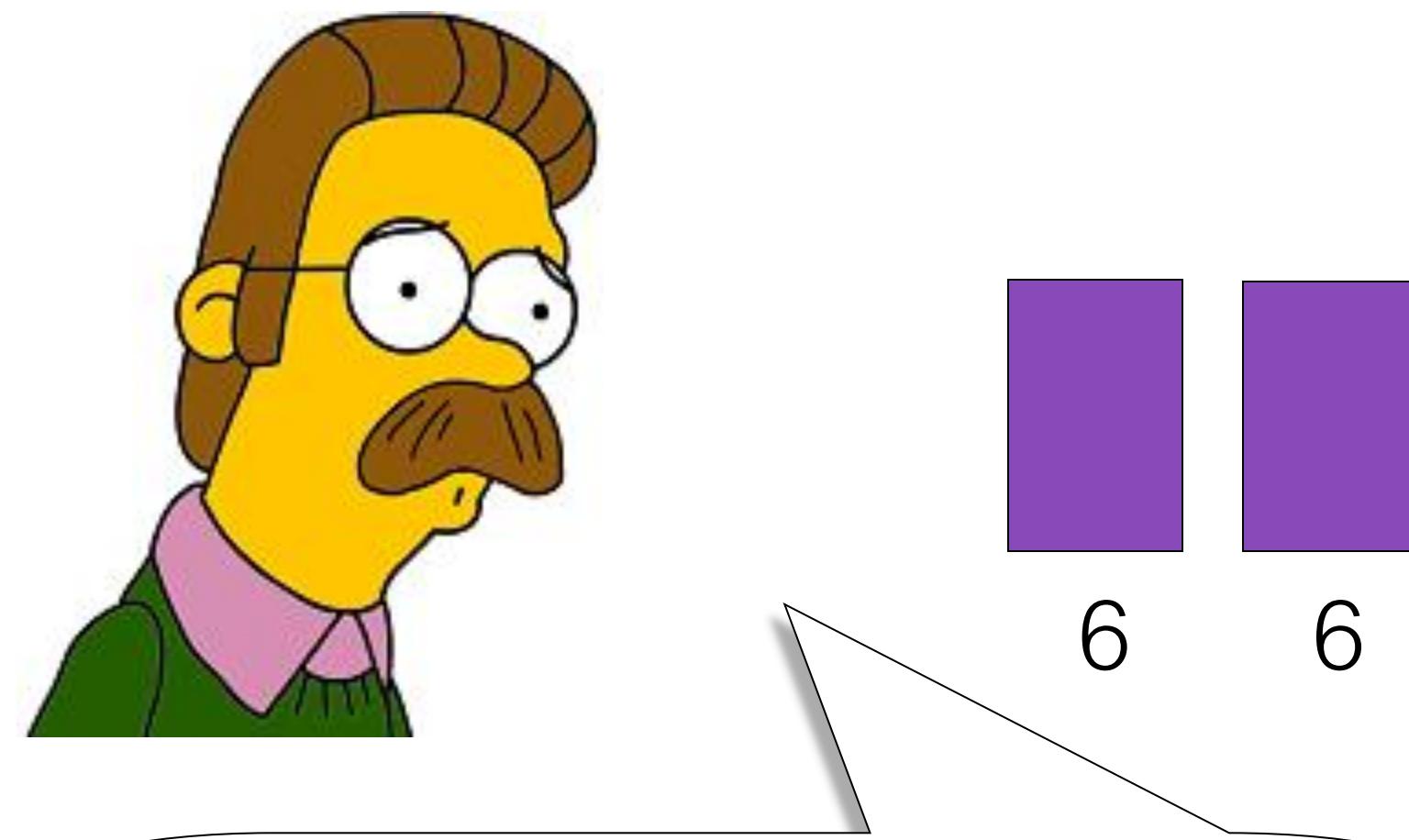
7      5



4      8



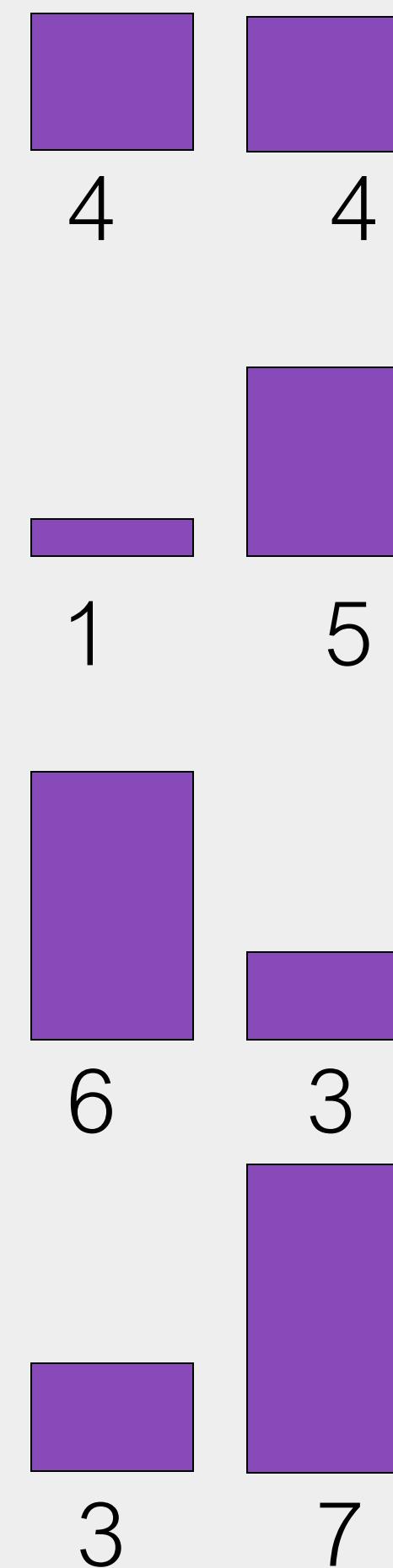
7      7



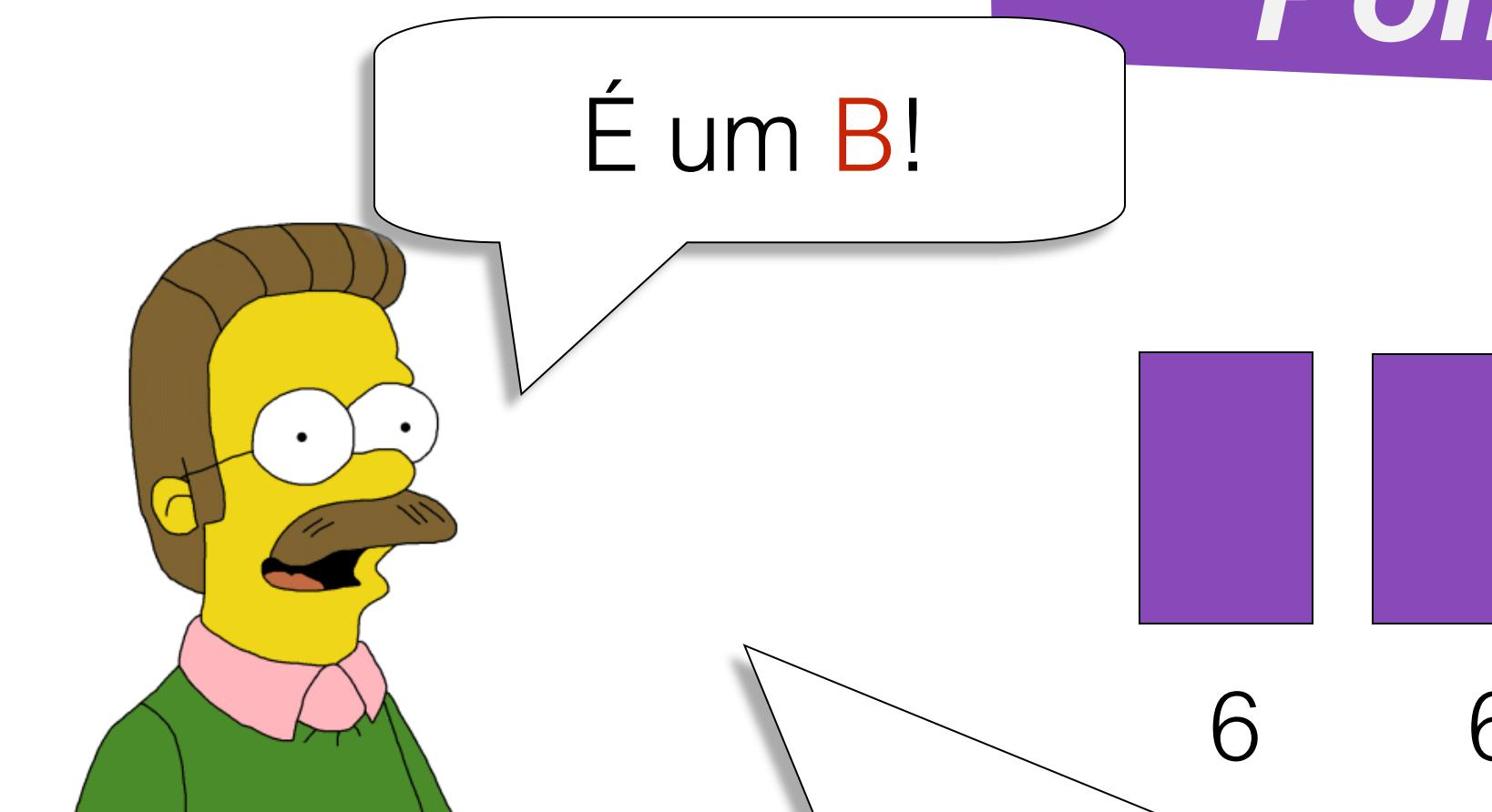
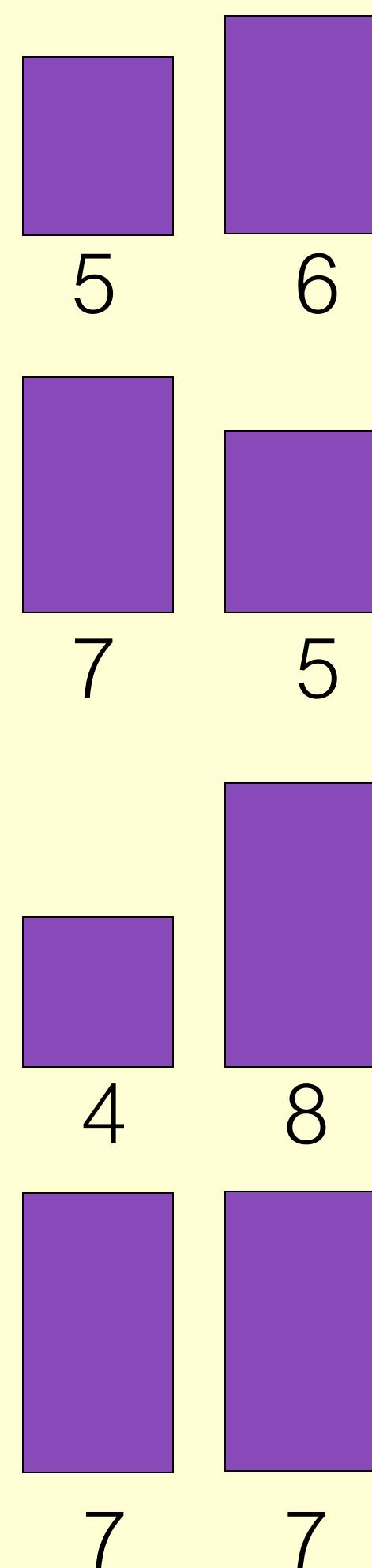
Este é muito difícil!  
Qual é este, A ou B?

# Classificação: Problema do Pombo 3

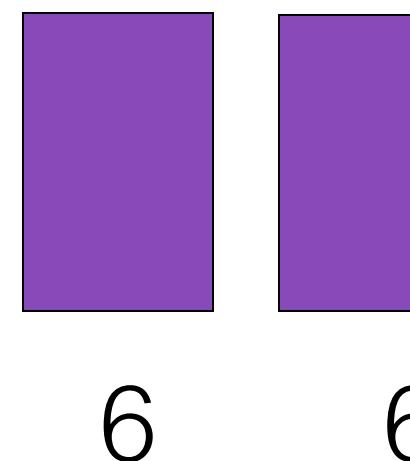
Exemplos da classe A



Exemplos da classe B



A regra é a seguinte: se o quadrado da soma das duas barras é menor ou igual a 100, é um A. Caso contrário é um B.

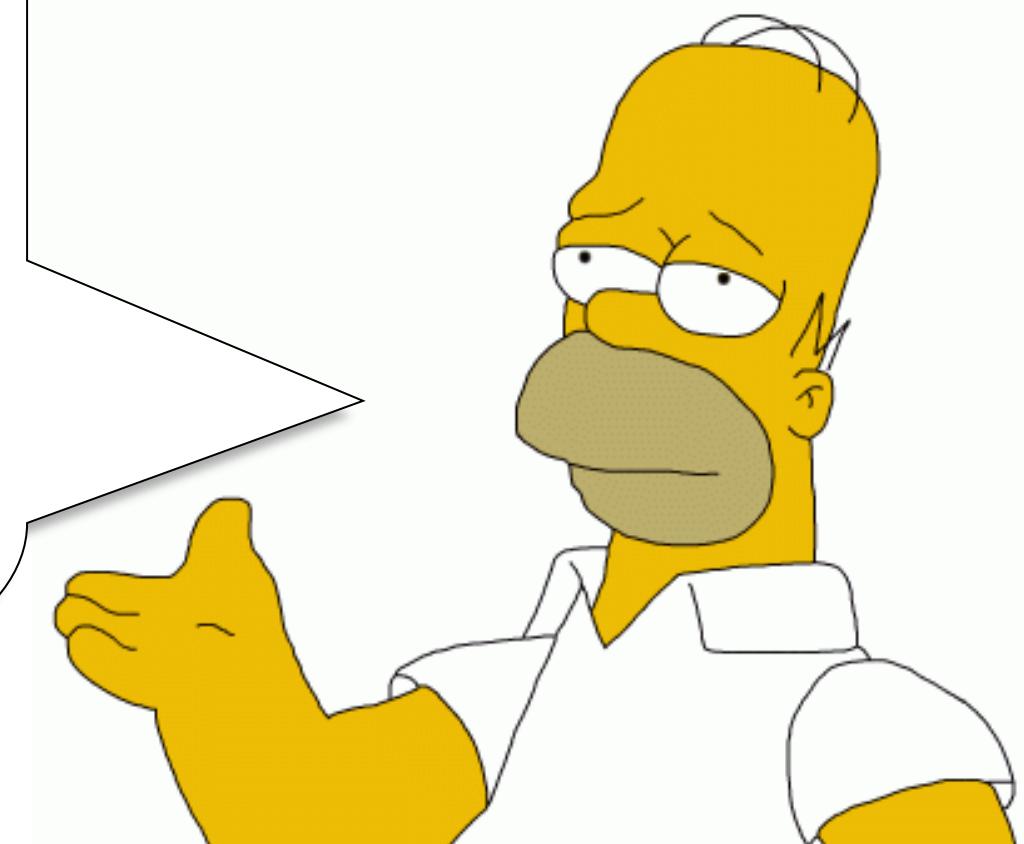


# Classificação

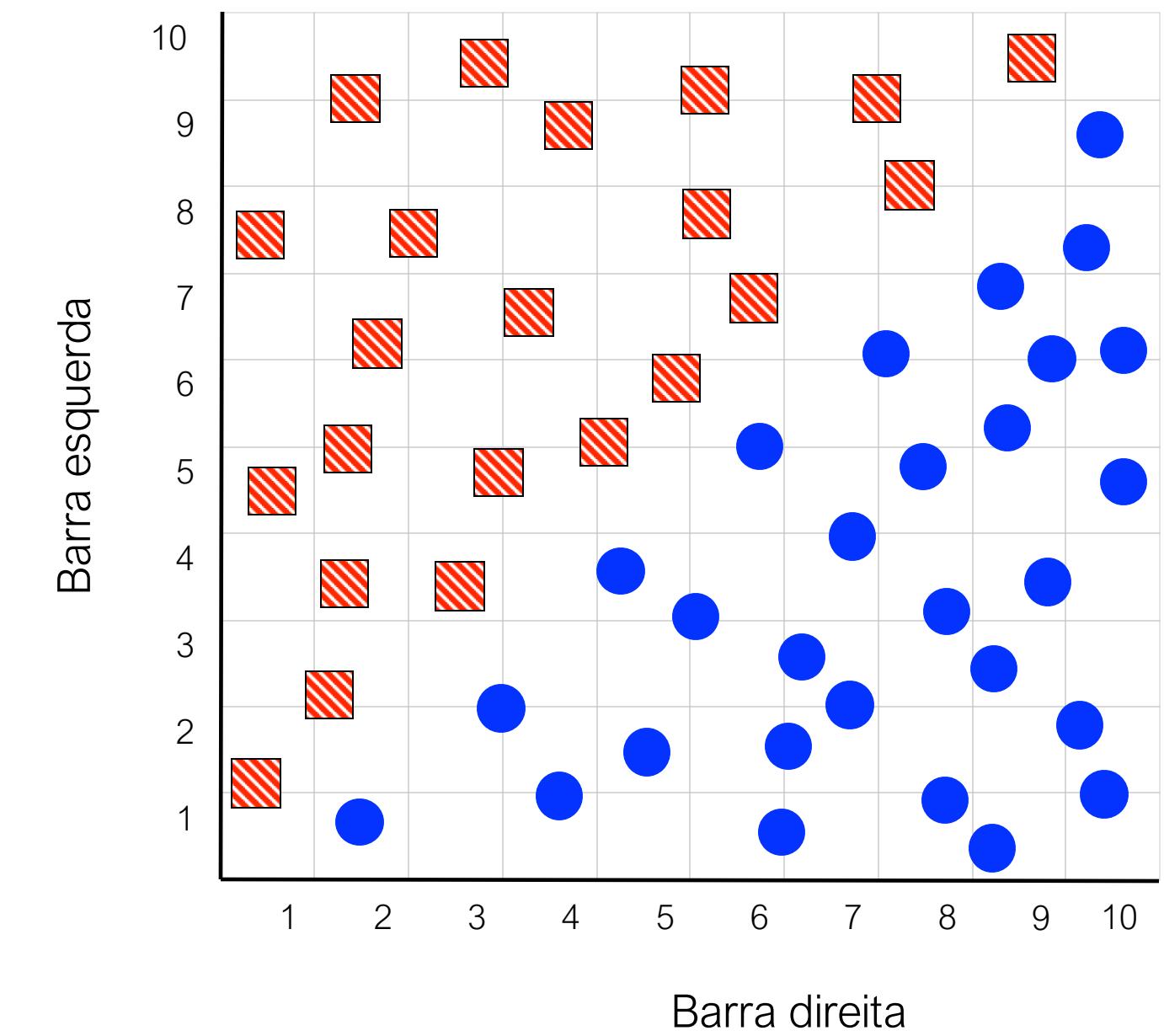
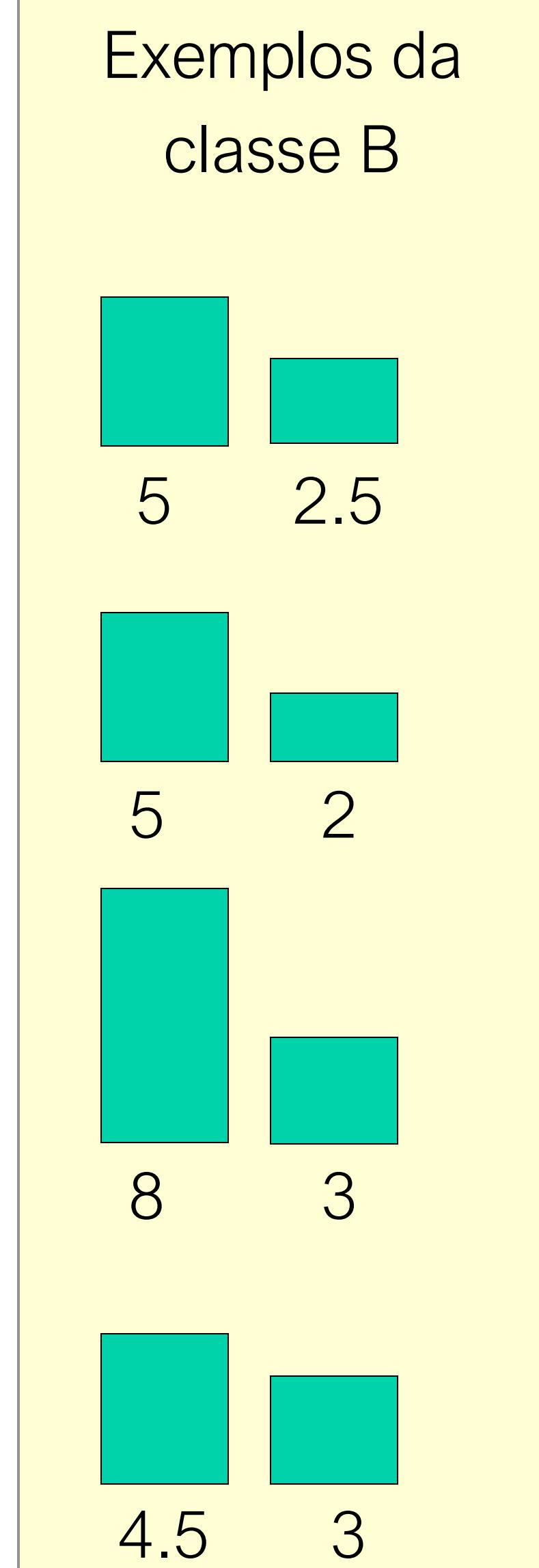
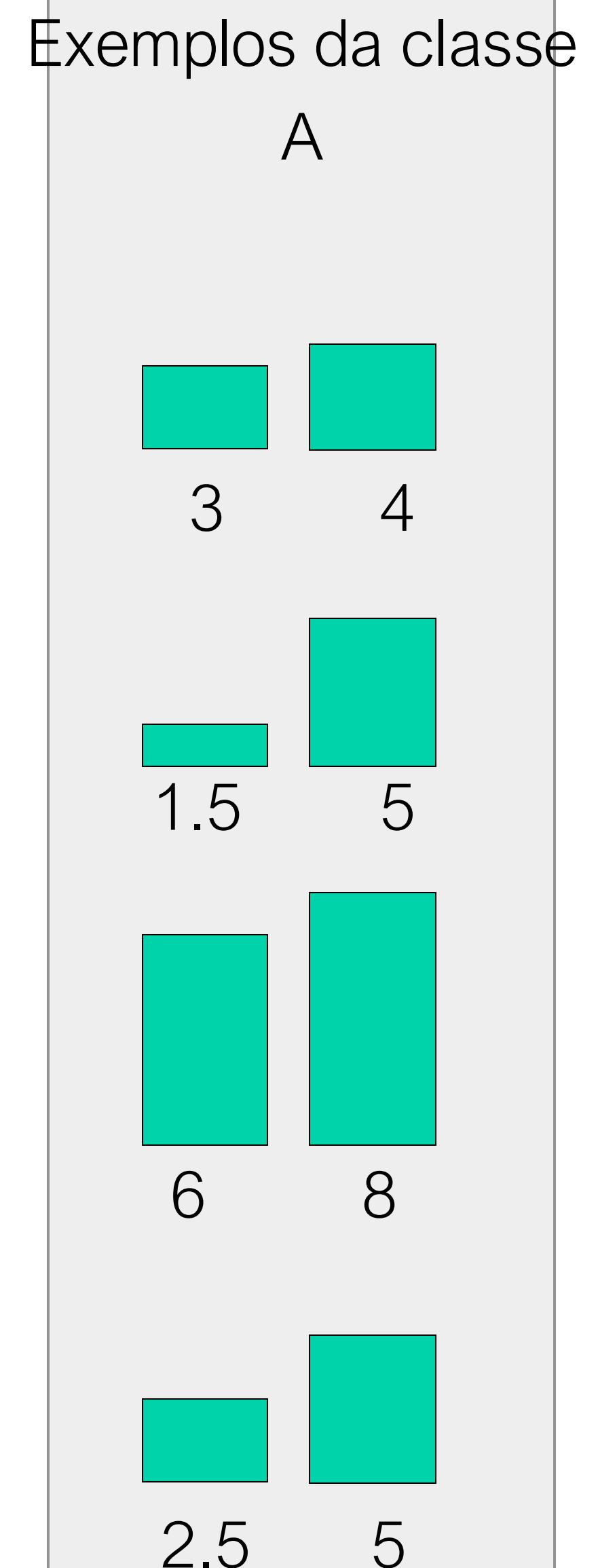


Por que gastamos tanto tempo com este  
joguinho?

Porque quero mostrar que quase todos  
os problemas de classificação tem uma  
interpretação geométrica. Confira os  
próximos 3 slides...



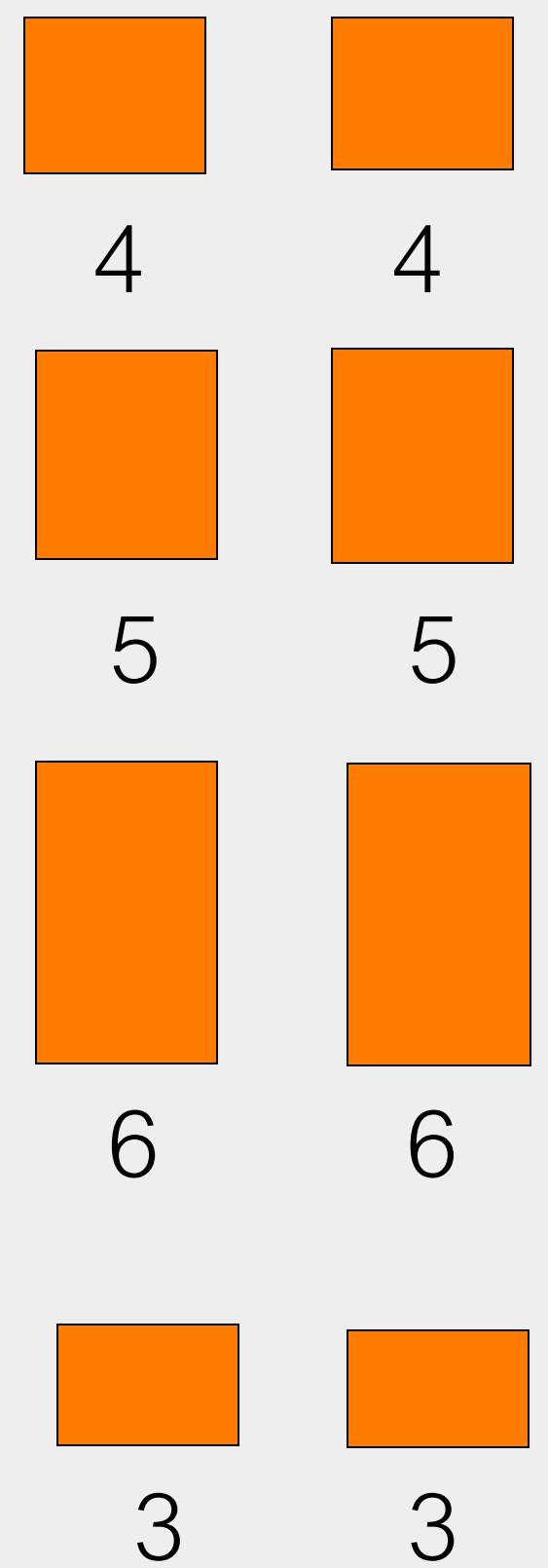
# Classificação: Problema do Pombo



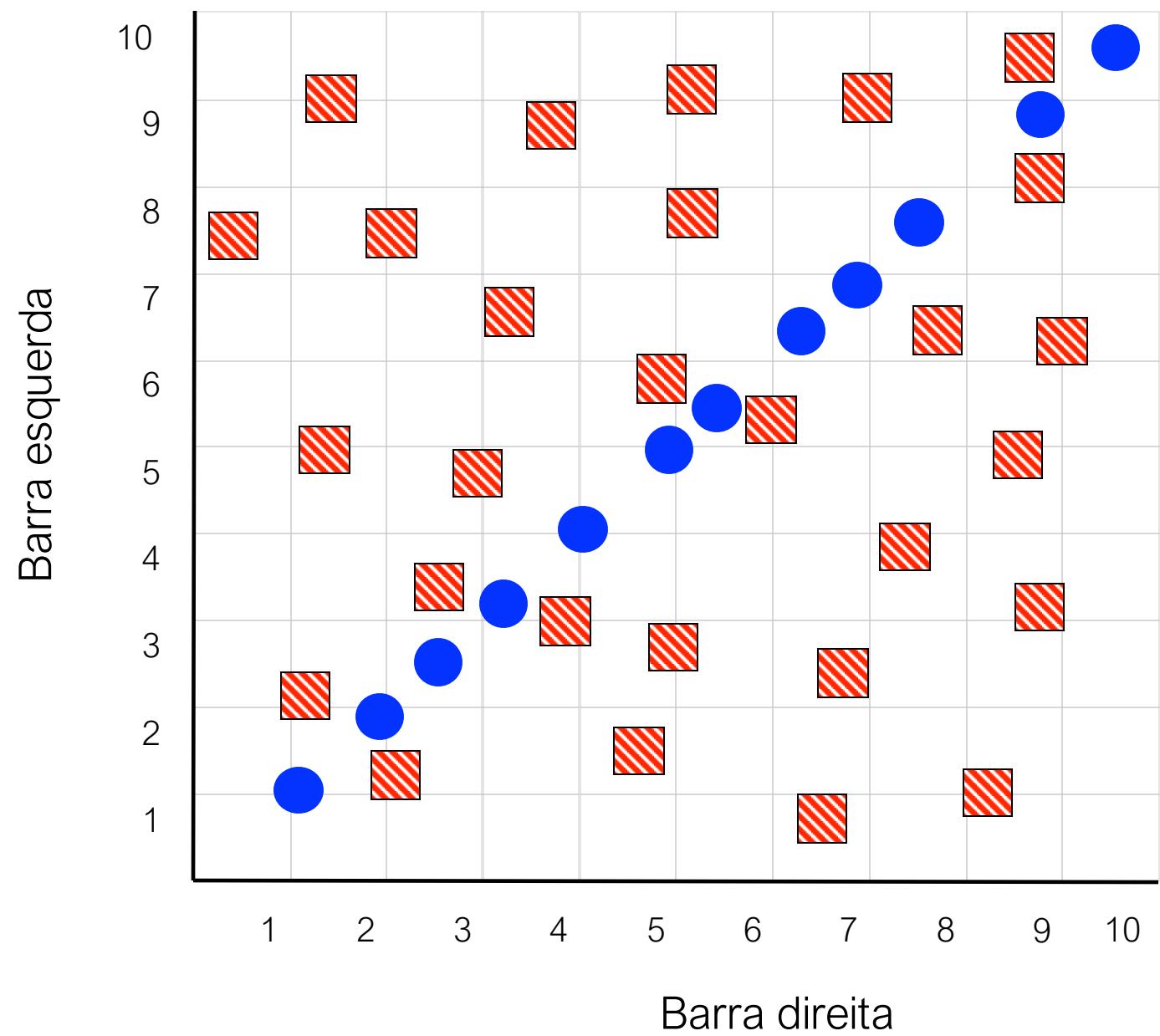
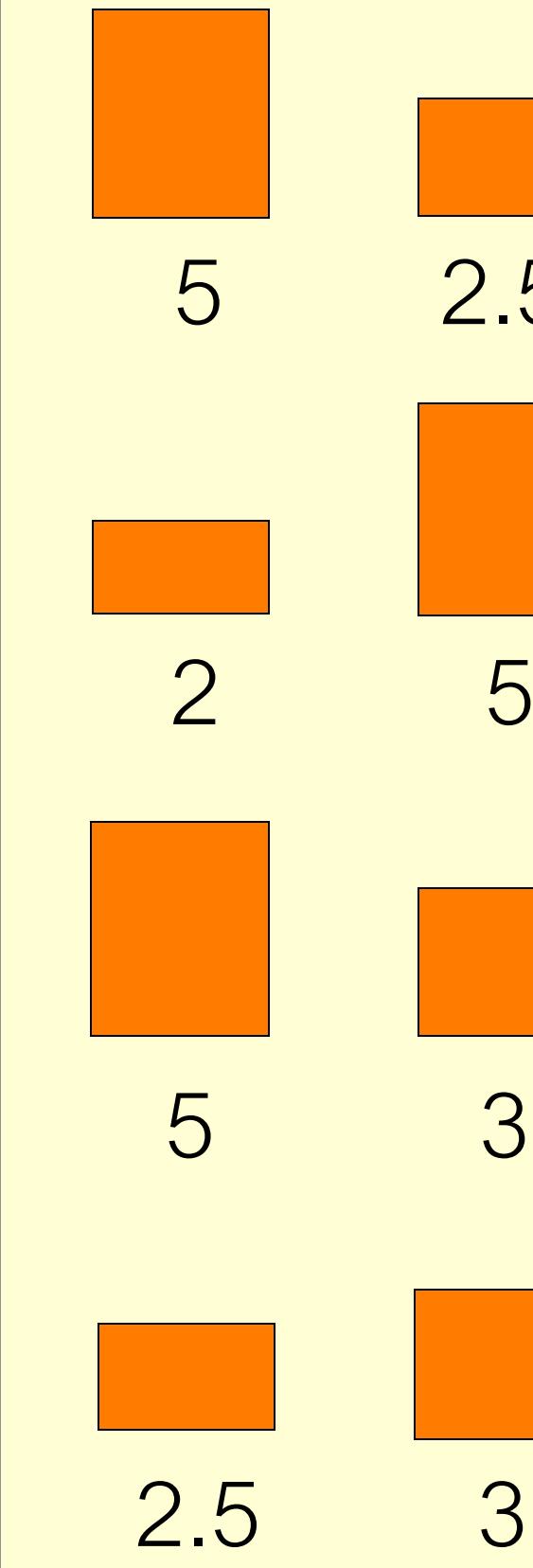
Eis a regra novamente.  
Se a barra esquerda é  
menor que a direita, é  
um A, caso contrário é  
um B.

# Classificação: Problema do Pombo 2

Exemplos da classe  
A



Exemplos da classe B

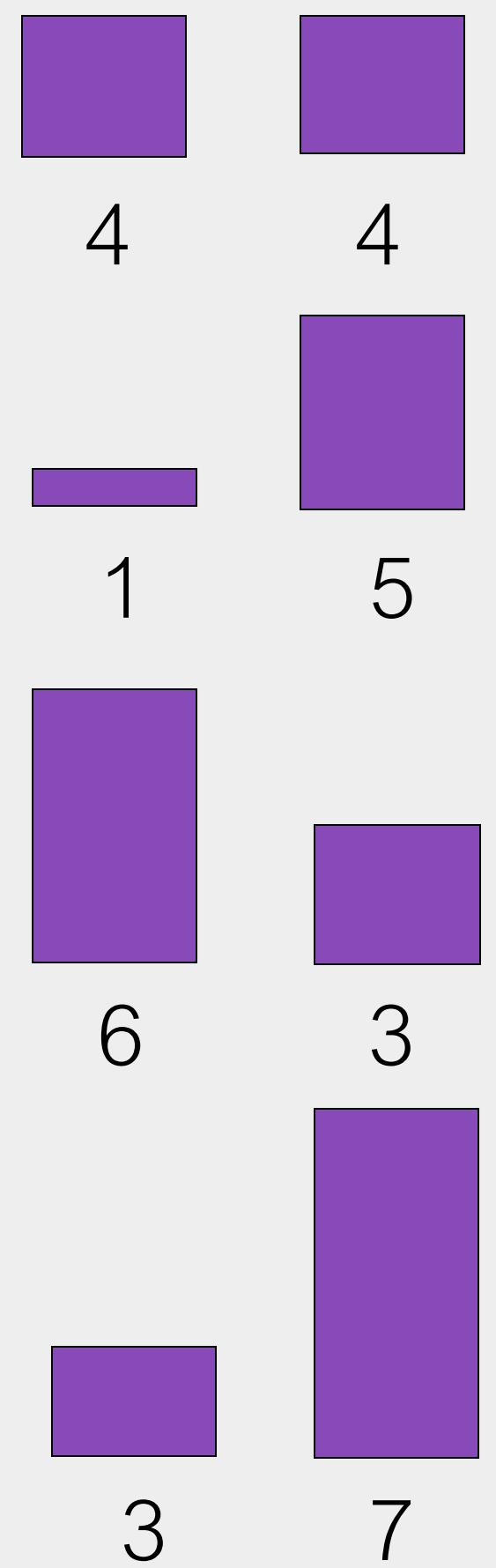


A regra é, se as duas barras têm tamanhos iguais, é um A. Senão é um B.

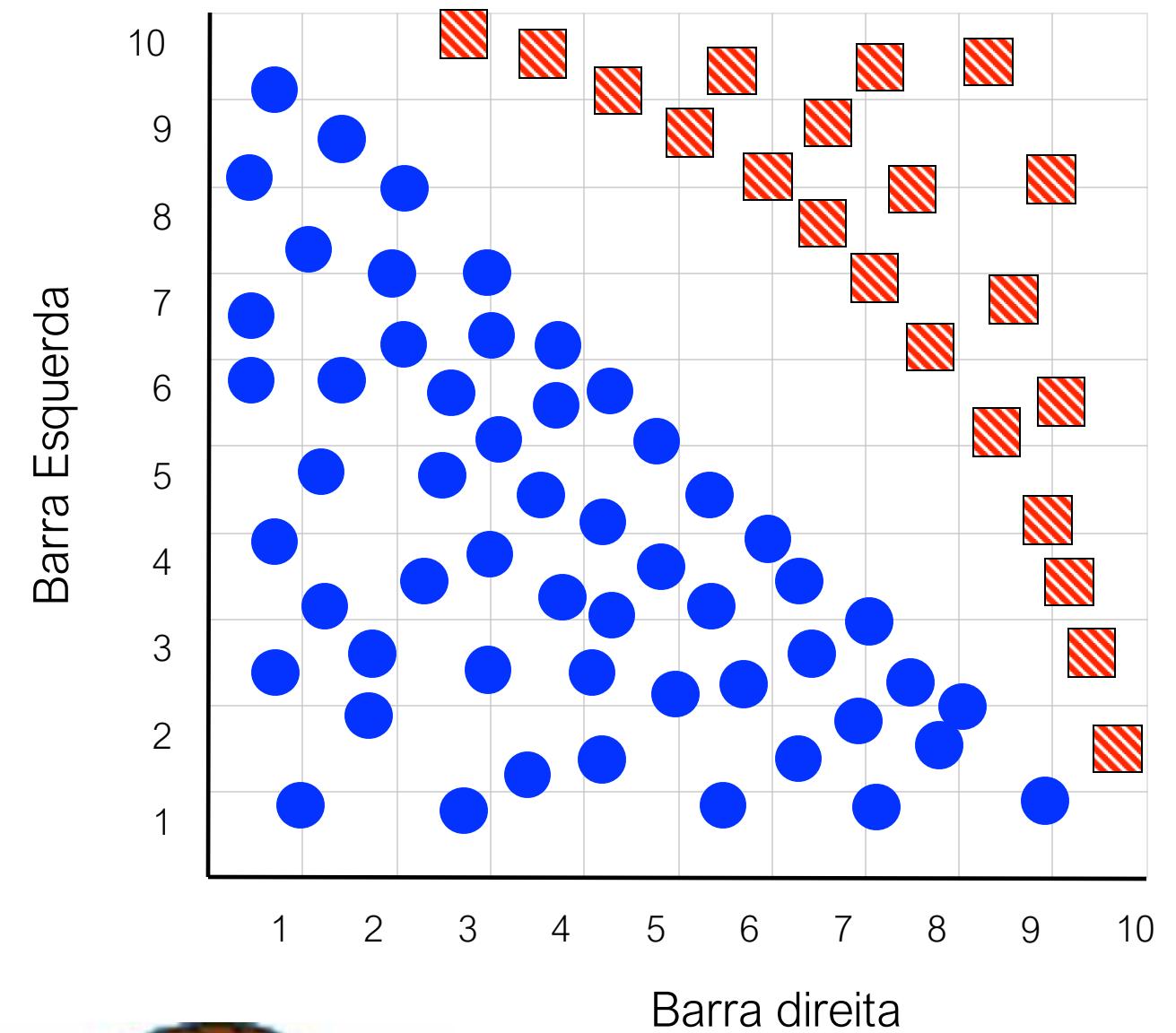
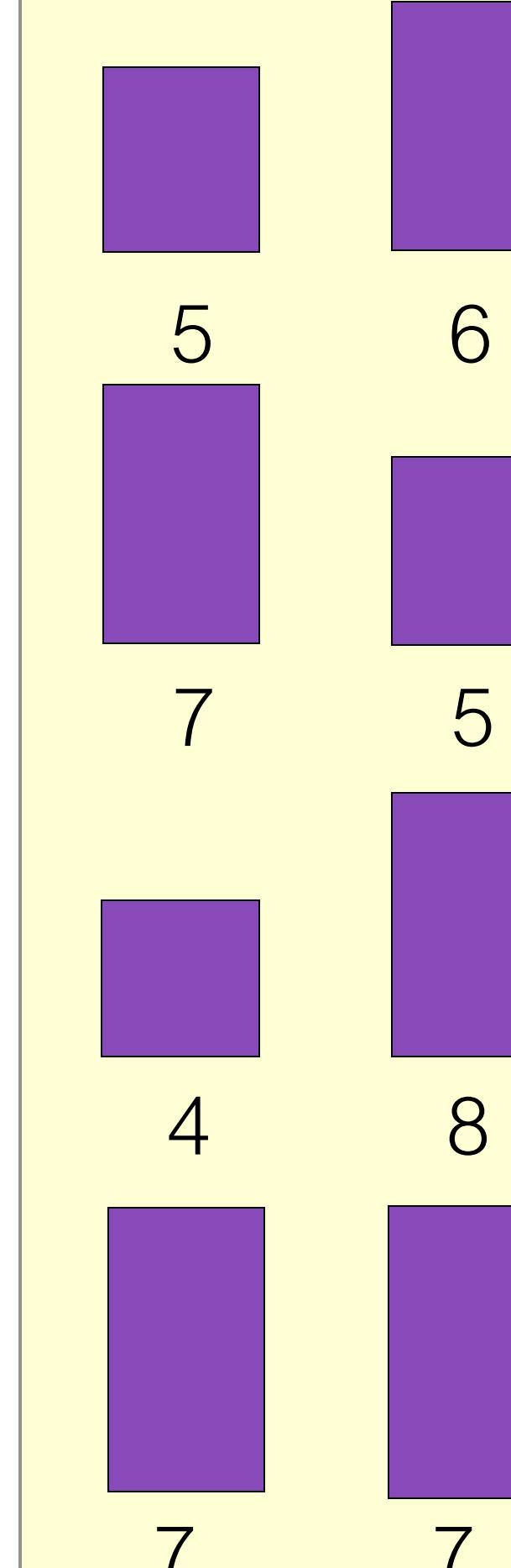


# Classificação: Problema do Pombo 3

Exemplos da classe  
A



Exemplos da classe  
B



A regra novamente:  
Se o quadrado da soma das duas  
barras é menor ou igual a 100, é  
um A. Senão é um B.

# Métodos Preditivos

## Métodos Baseados em Distância

**Algoritmo  $k$ -NN**

## Métodos Baseados em Procura

Árvores de Decisão e Regressão

Regras de Decisão

## Métodos Probabilísticos

Classificador *Naive Bayes*

Redes Bayesianas para Classificação

## Métodos Baseados em Otimização

Redes Neurais

SVM: máquinas de vetores de suporte

## Outros Métodos

Algoritmos Genéticos

Conjuntos Fuzzy

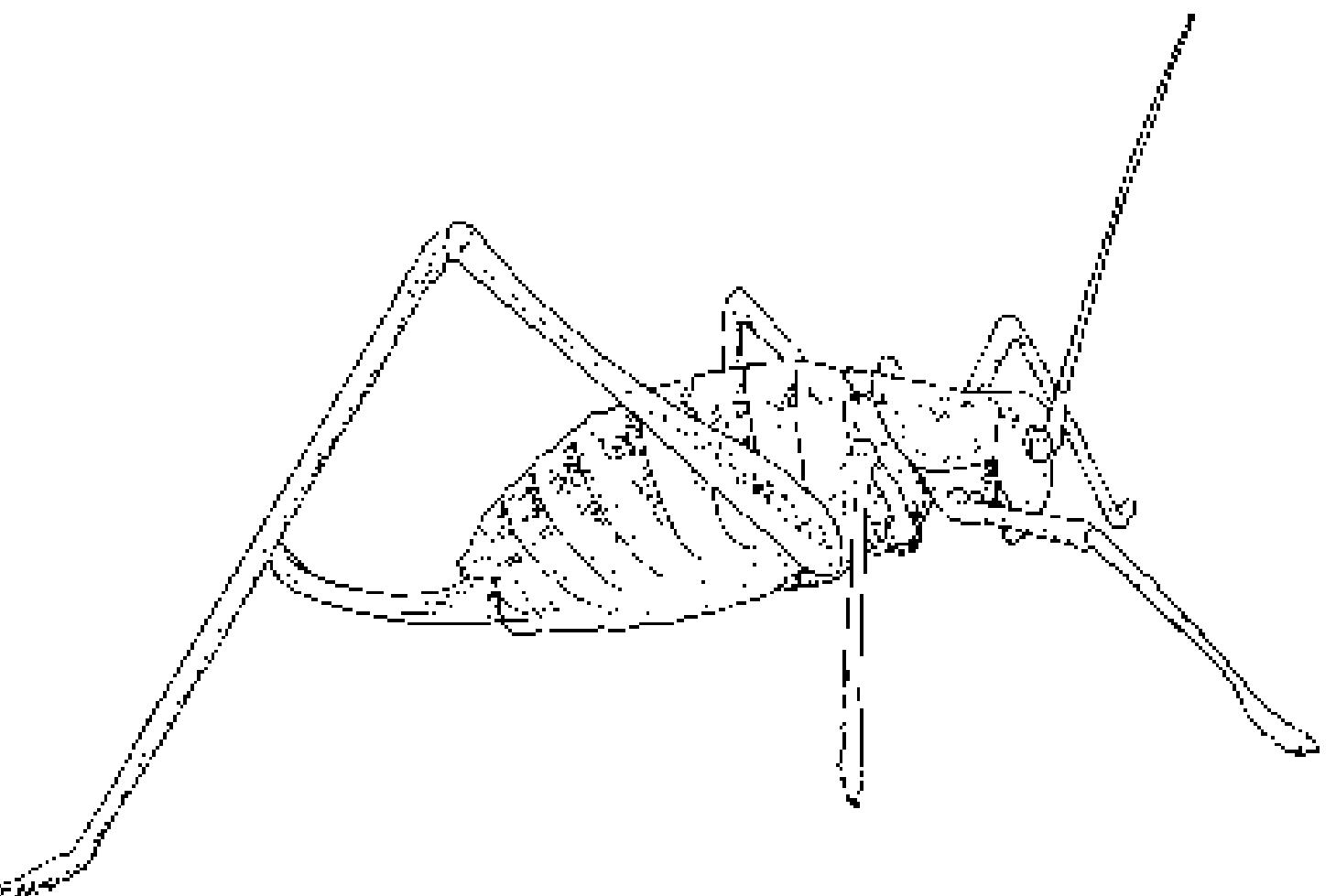
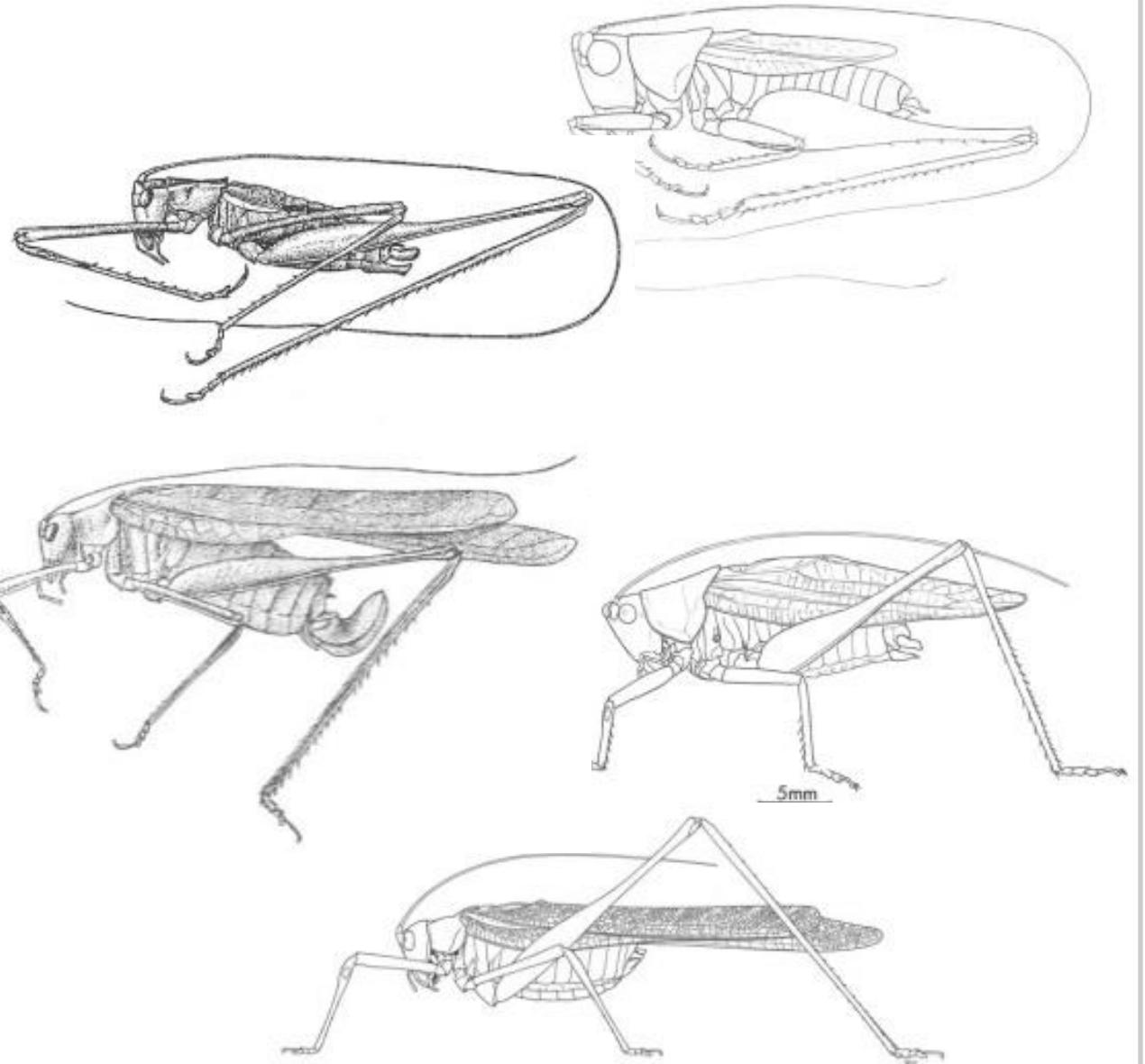
# Classificação: O Problema (definição informal)

- Dada uma coleção de dados detalhados (neste caso 5 exemplos de **Esperança** e 5 do **Gafanhoto**), decida a qual tipo de inseto o exemplo não rotulado pertence.

Obs: **Esperança** = tipo de gafanhoto verde

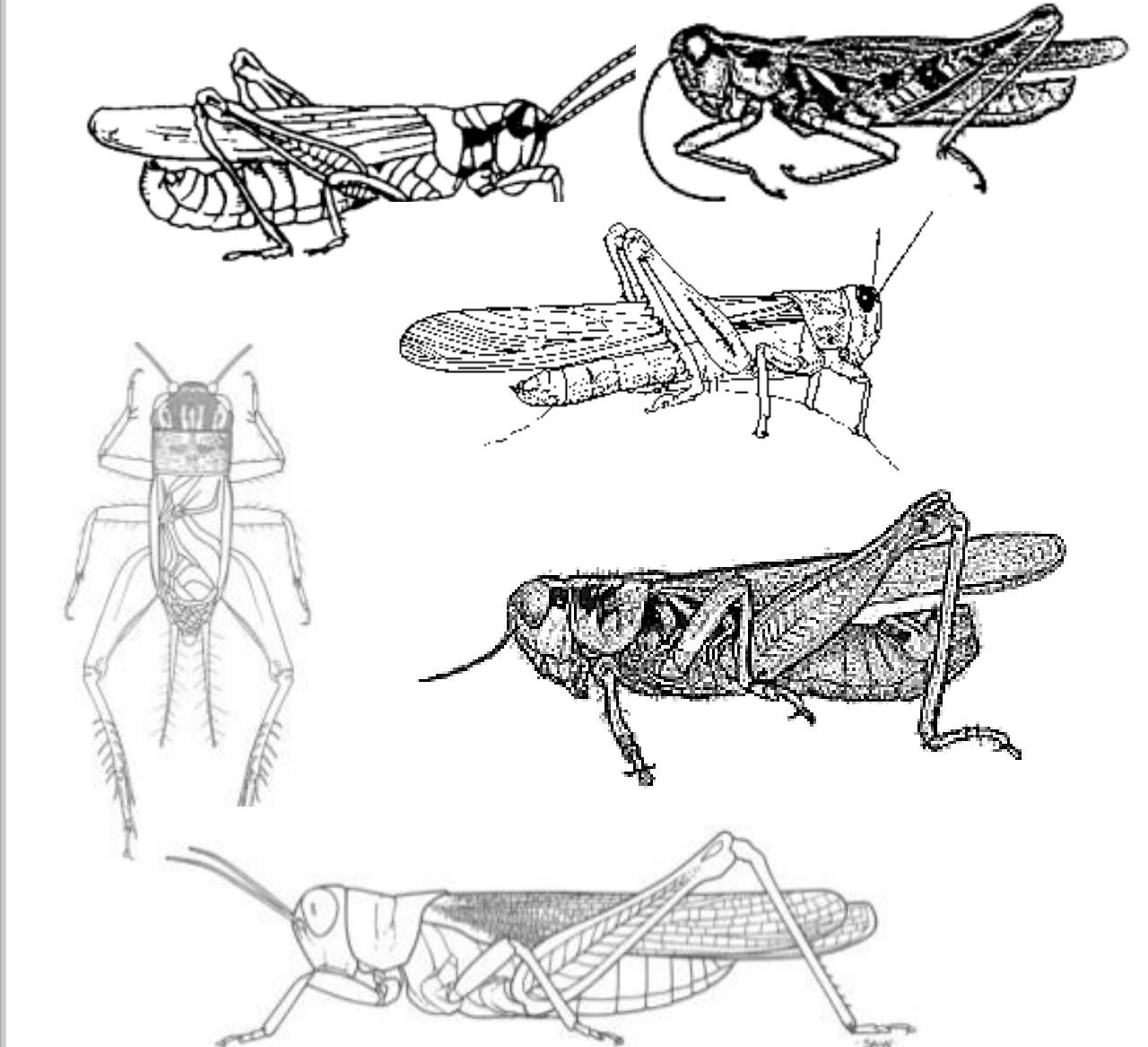
# Classificação: O Problema (definição informal)

Esperança



Esperança ou Gafanhoto?

Gafanhoto

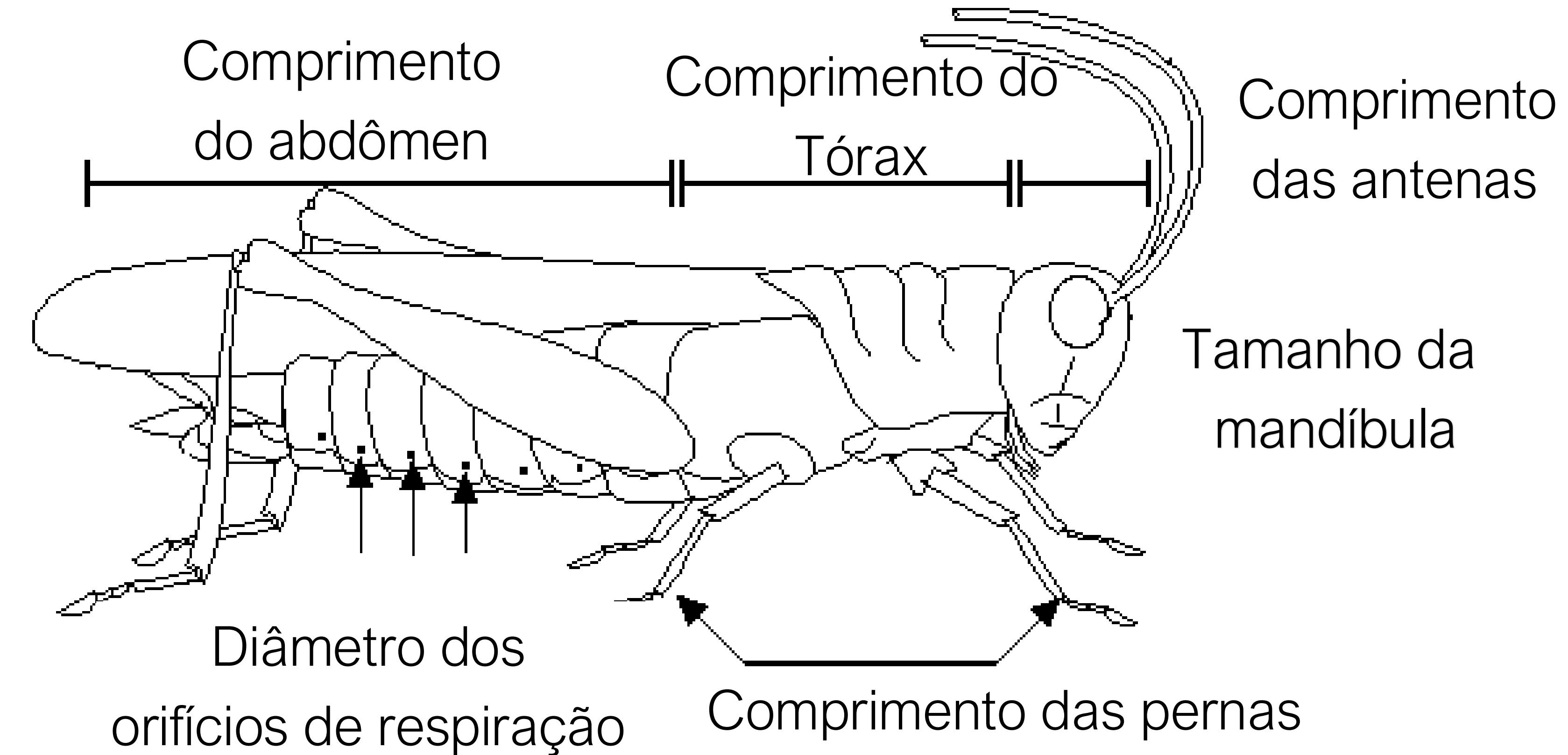


# Classificação: Domínio de Interesse

Cor: {Verde, Marrom, Cinza, Outra}

## Medir Características

Tem asas?



# Armazenamento de Características

O problema de classificação agora pode ser expresso da seguinte forma:

- Dada uma base de treino (Base), preveja o rótulo da classe dos exemplos ainda não vistos

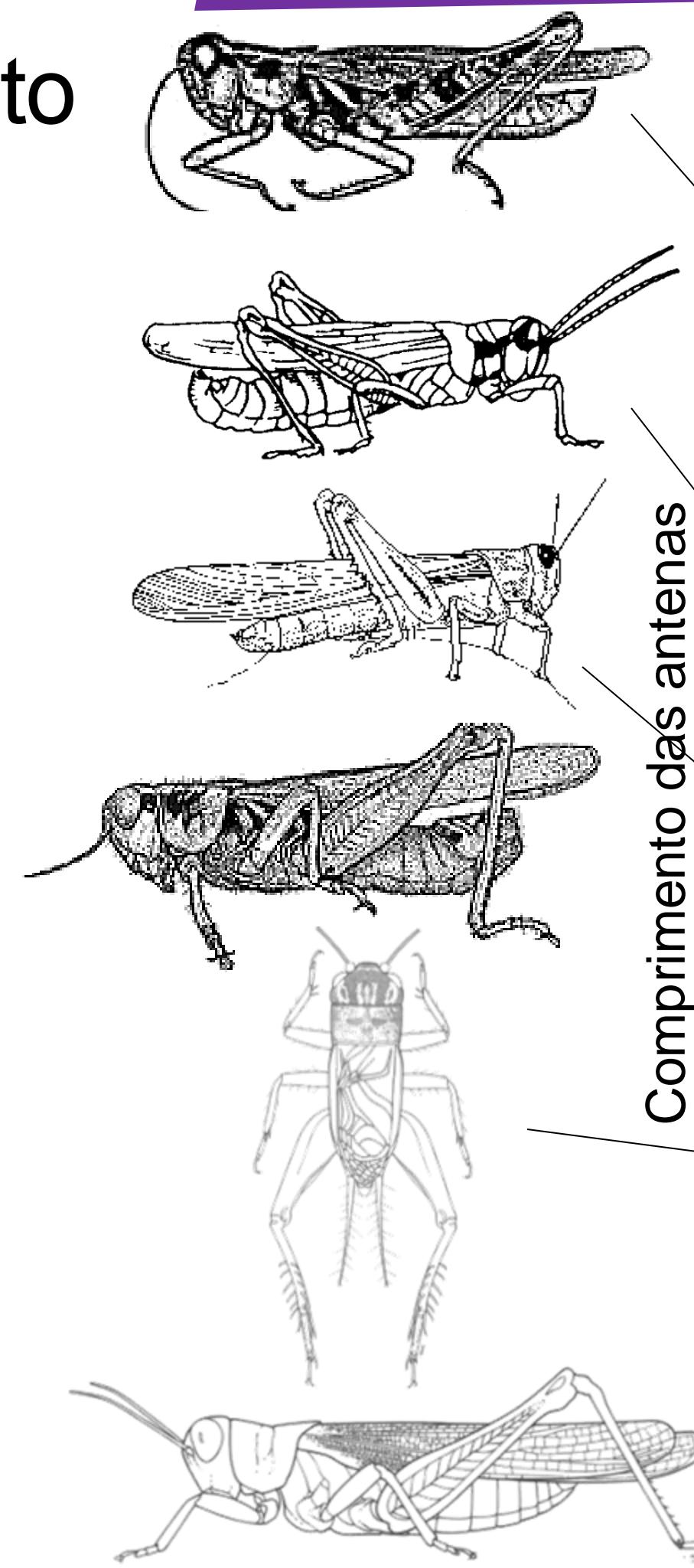
ID do inseto	Comp. do abdômen	Comp. das antenas	Classe do inseto
1	2.7	5.5	Gafanhoto
2	8.0	9.1	Esperança
3	0.9	4.7	Gafanhoto
4	1.1	3.1	Gafanhoto
5	5.4	8.5	Esperança
6	2.9	1.9	Gafanhoto
7	6.1	6.6	Esperança
8	0.5	1.0	Gafanhoto

Exemplo não visto =

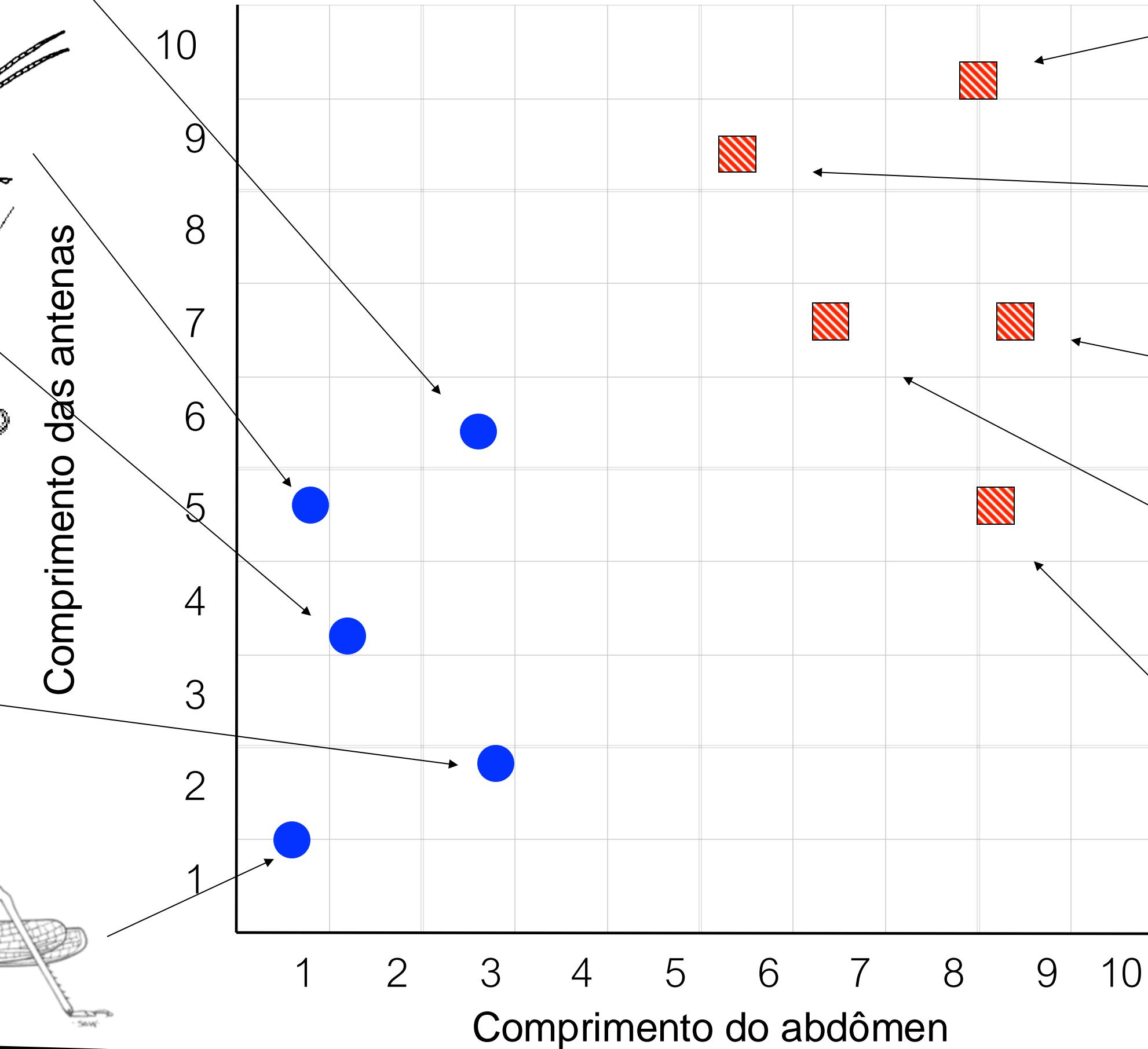
11	5.1	7.0	?????????
----	-----	-----	-----------

# Representação de Características

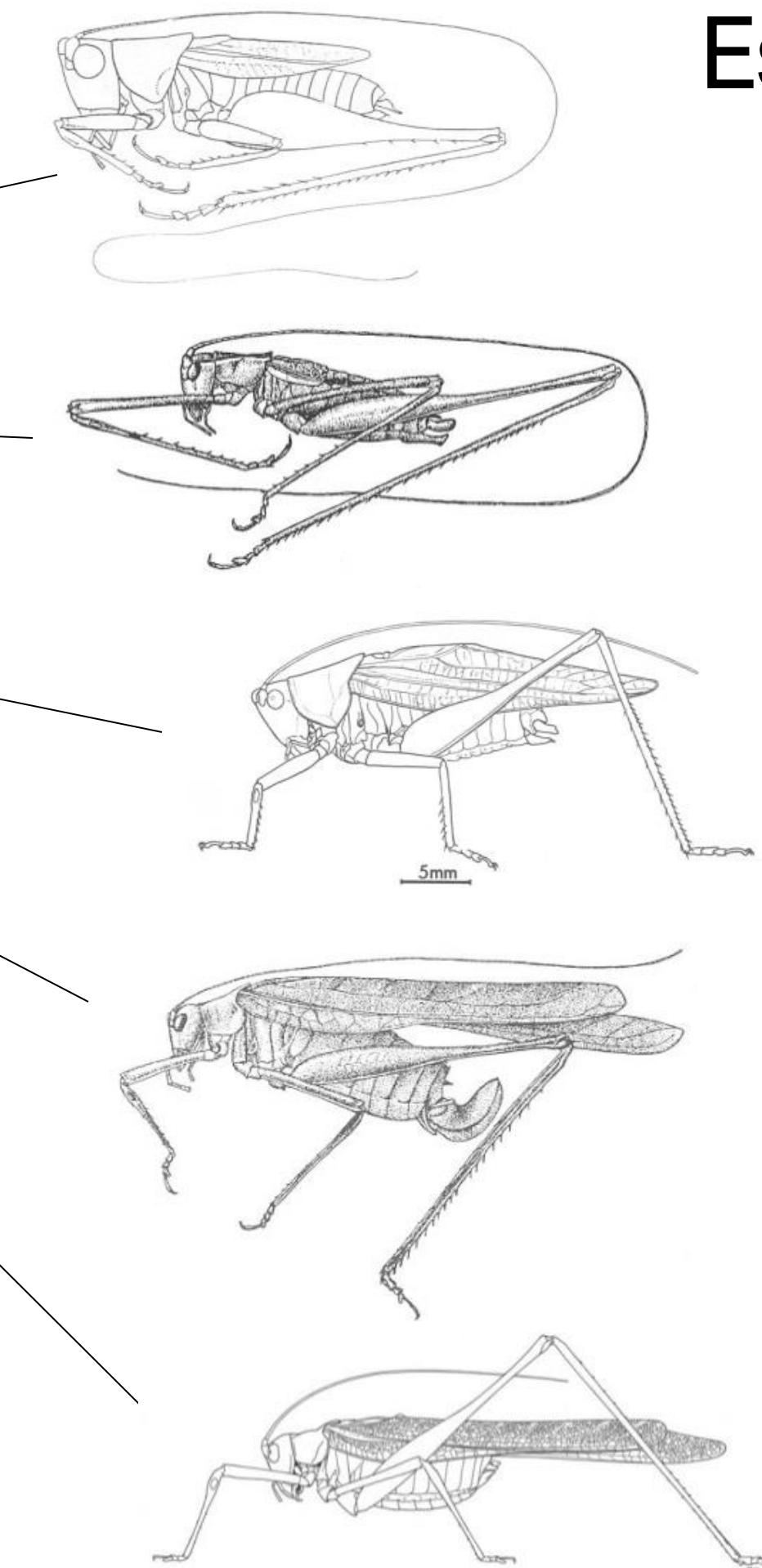
Gafanhoto



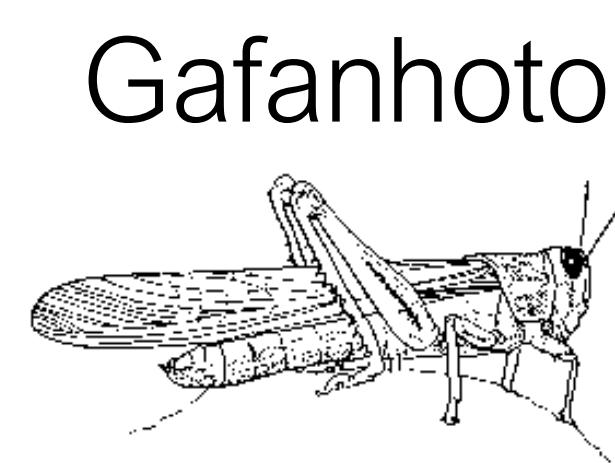
Comprimento das antenas



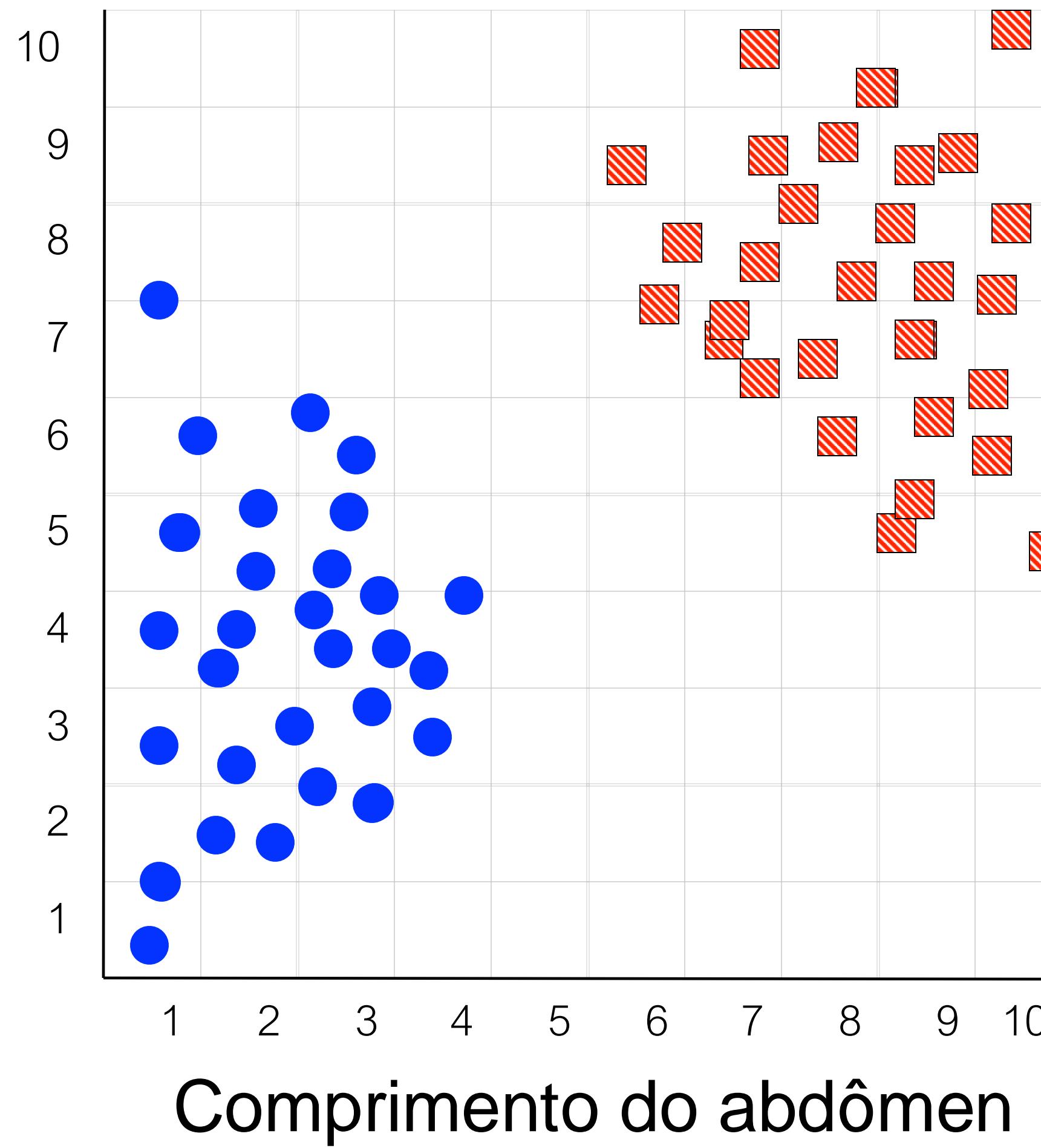
Esperança



# Mais exemplos na Base

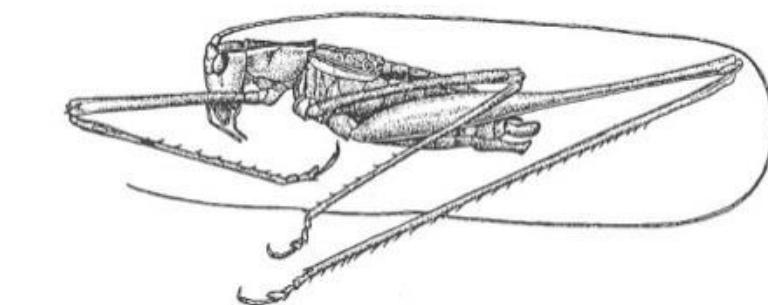


Comprimento das antenas

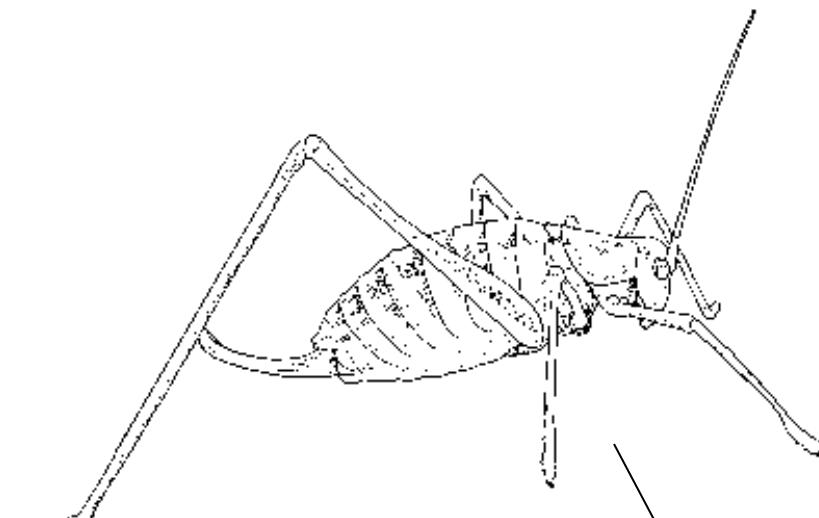


Suponha que a base tenha mais exemplos, como apresentado abaixo

Esperança



- Cada um destes objetos de dados é chamado de:
- exemplo (de treino)
  - instância
  - linha
  - tupla
  - exemplar
  - objeto



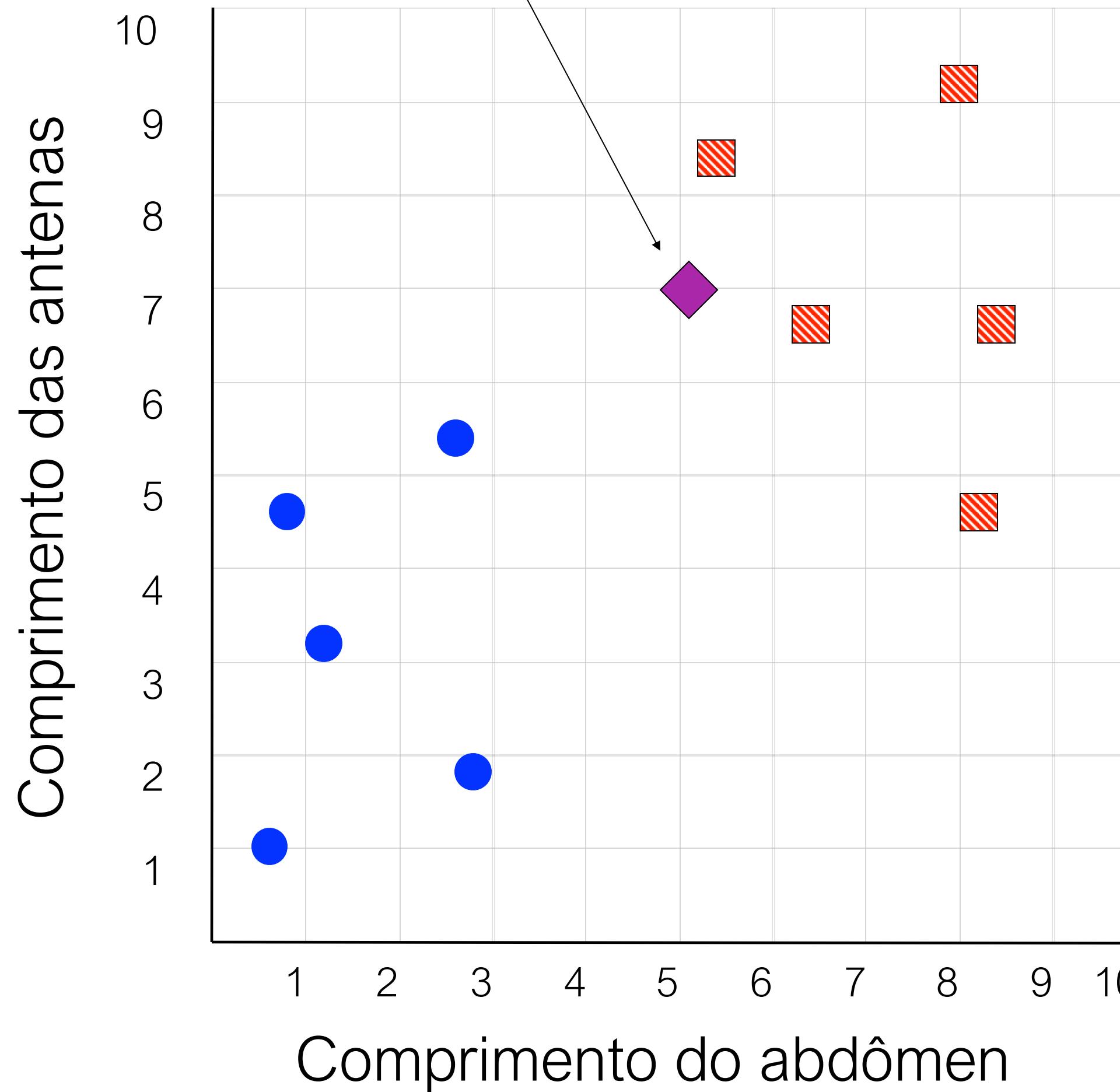
Exemplo não visto antes =

11

5.1

7.0

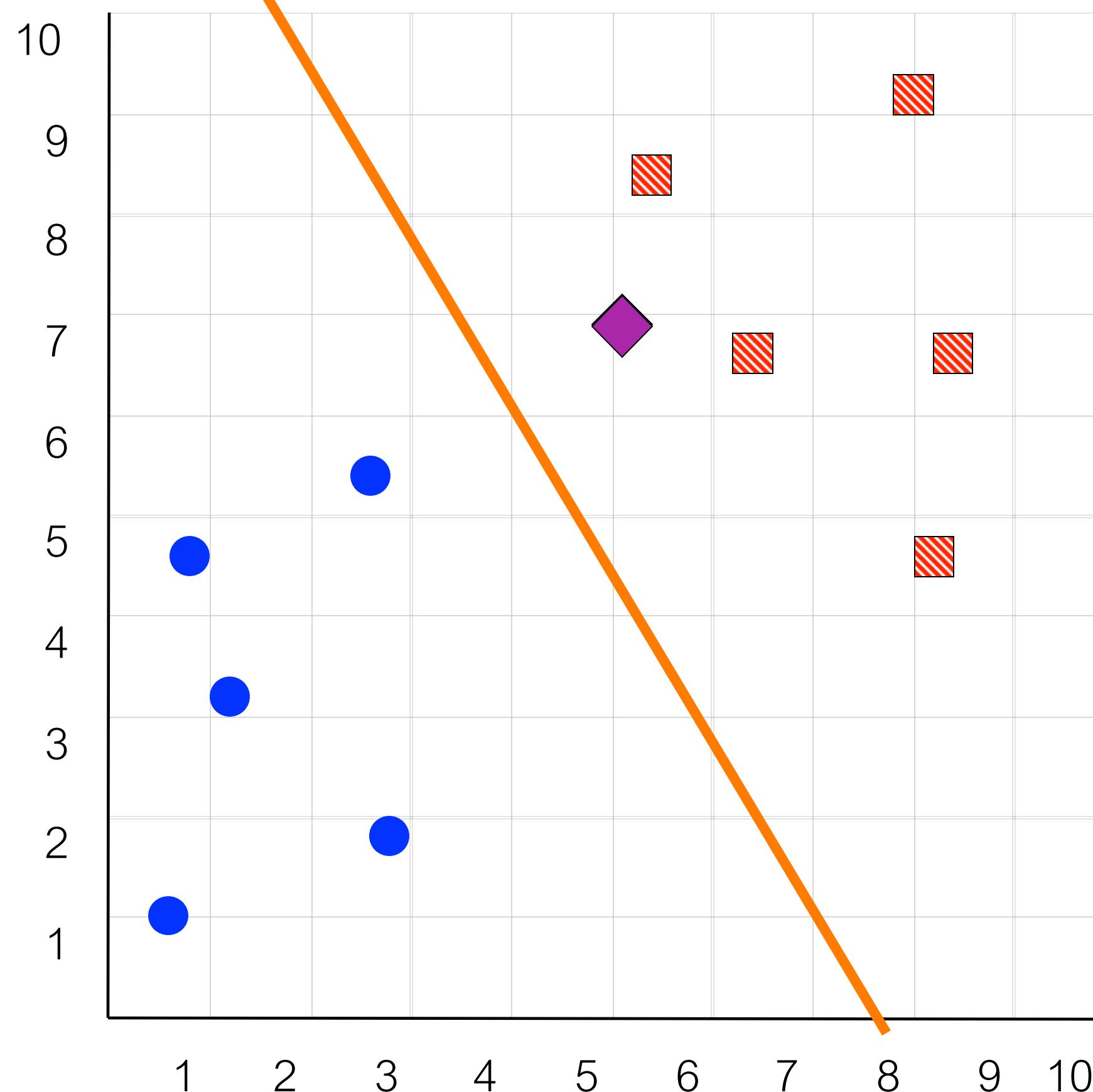
???????



- Esperança
- Gafanhoto

- ◆ Podemos “projetar” o exemplo não visto antes dentro do mesmo espaço que a base de dados.
- ◆ Acabamos de abstrair os detalhes do nosso problema particular. Será muito mais fácil falar de pontos no espaço.

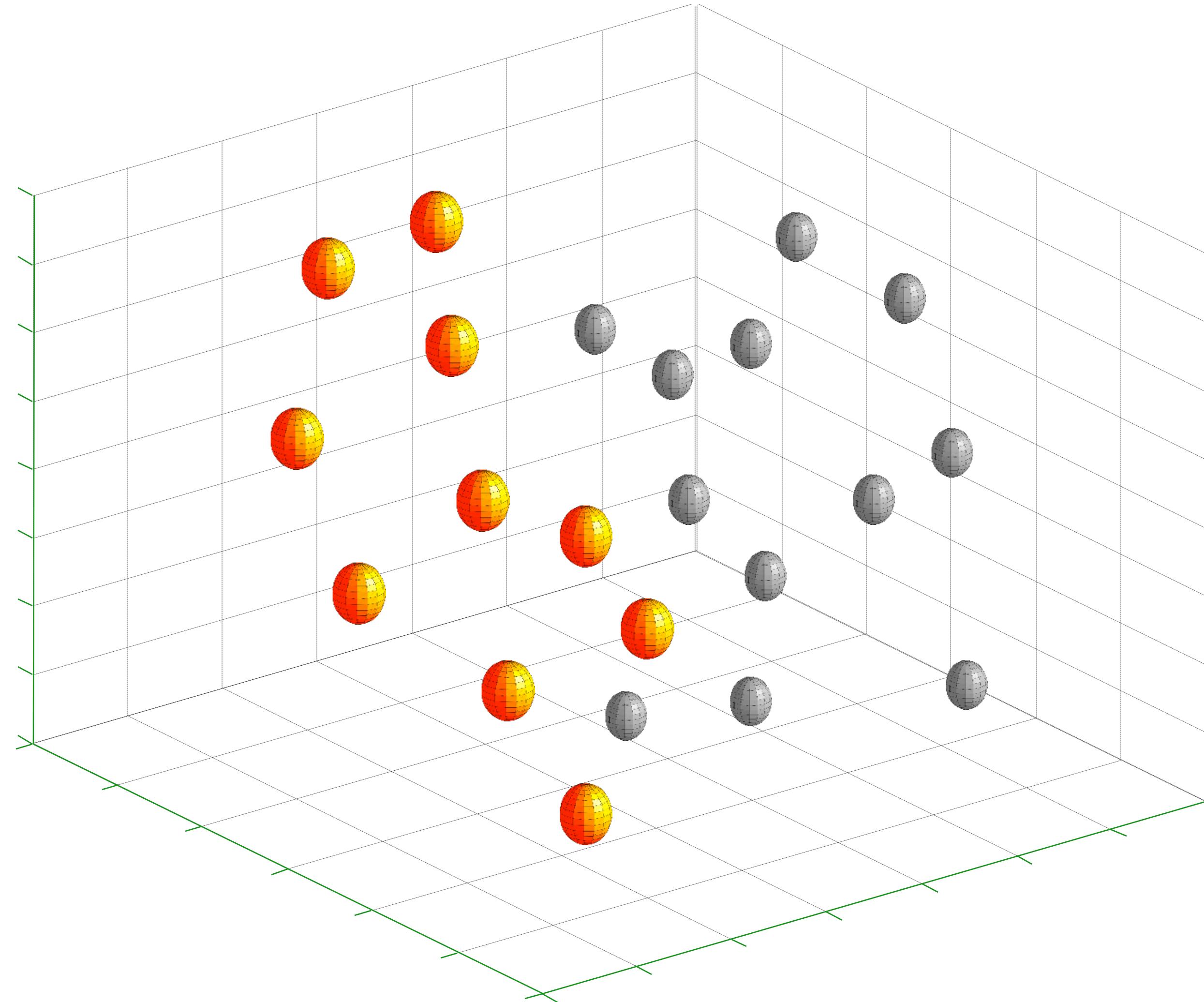
# Classificador Linear Simples



se exemplo desconhecido está acima  
da linha  
então  
classe é Esperança  
senão  
classe é Gafanho

R.A. Fisher  
1890-1962  
1936

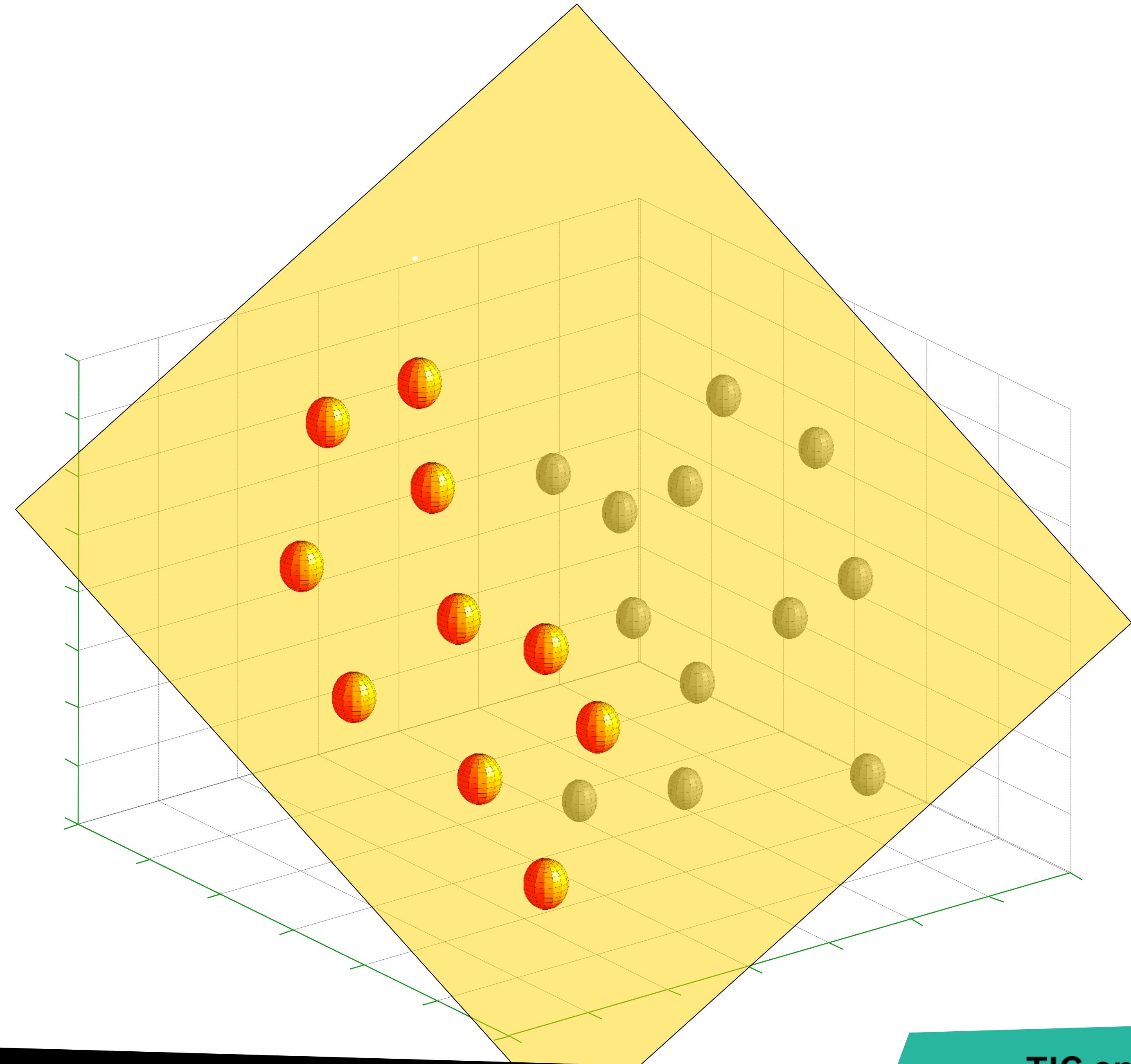
# Classificador Linear Simples



Está definido para  
espaços  
dimensionais  
maiores

# Classificador Linear Simples

Podemos visualizá-lo  
como sendo um  
hiperplano n-dimensional

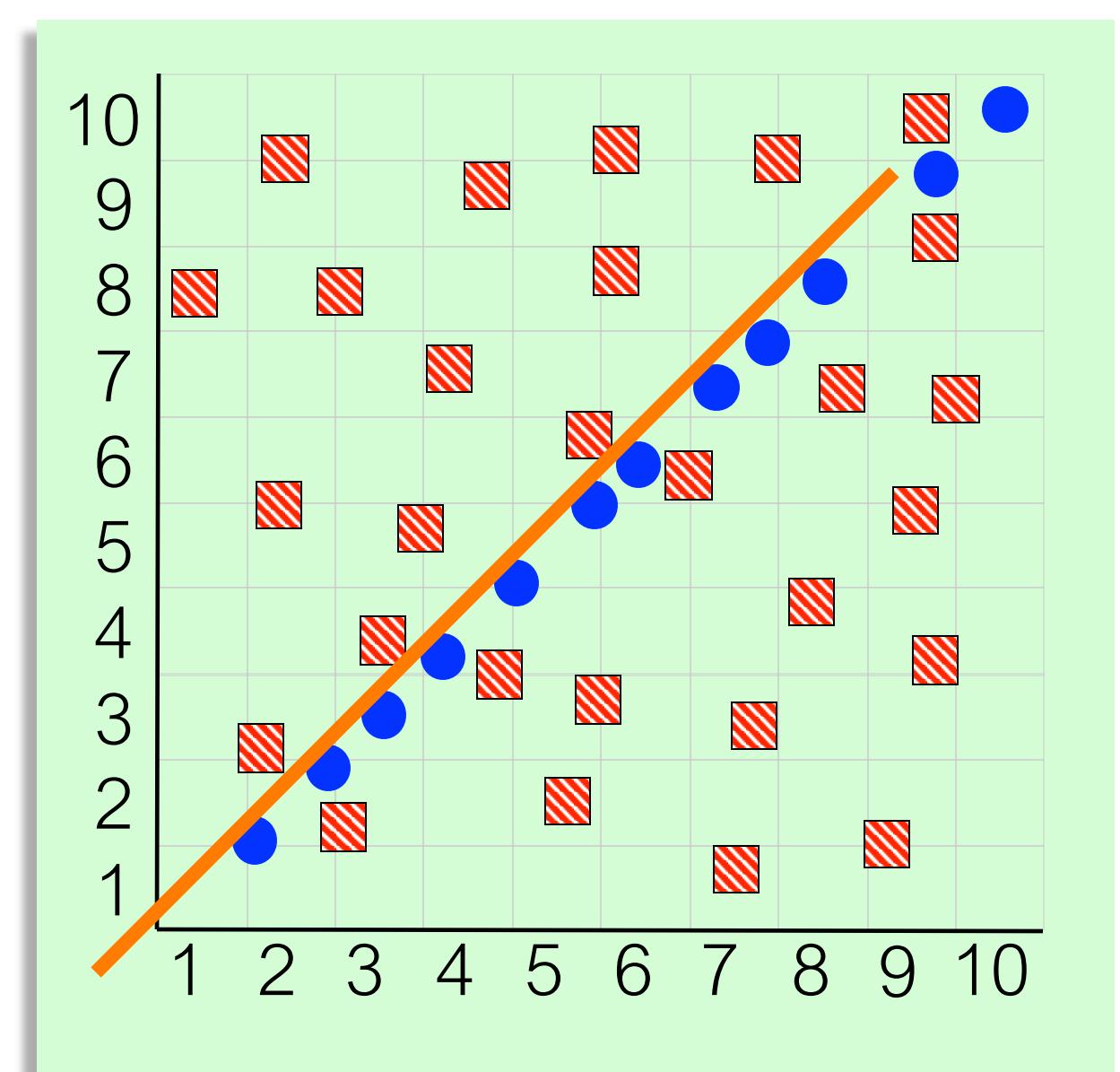
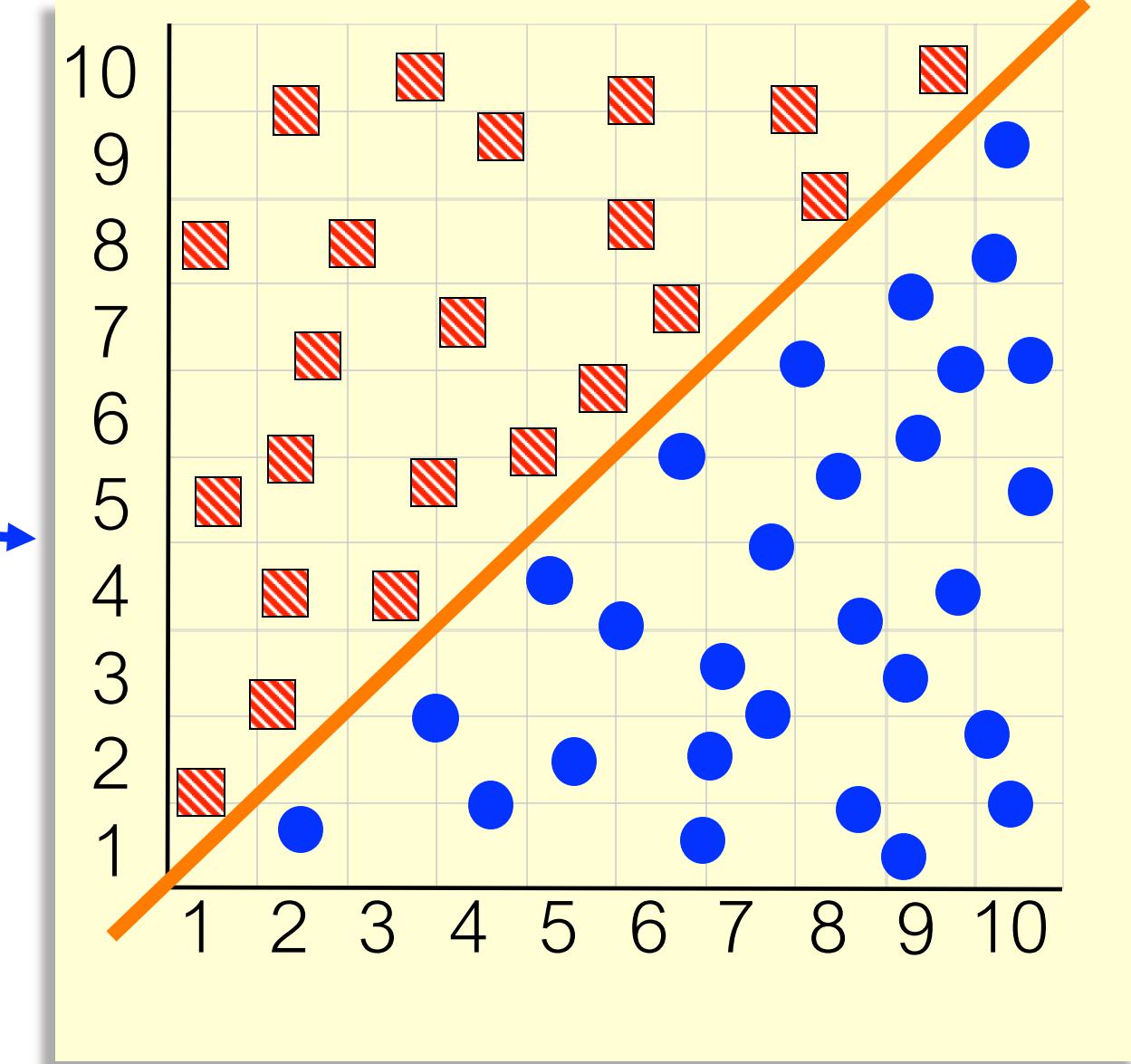
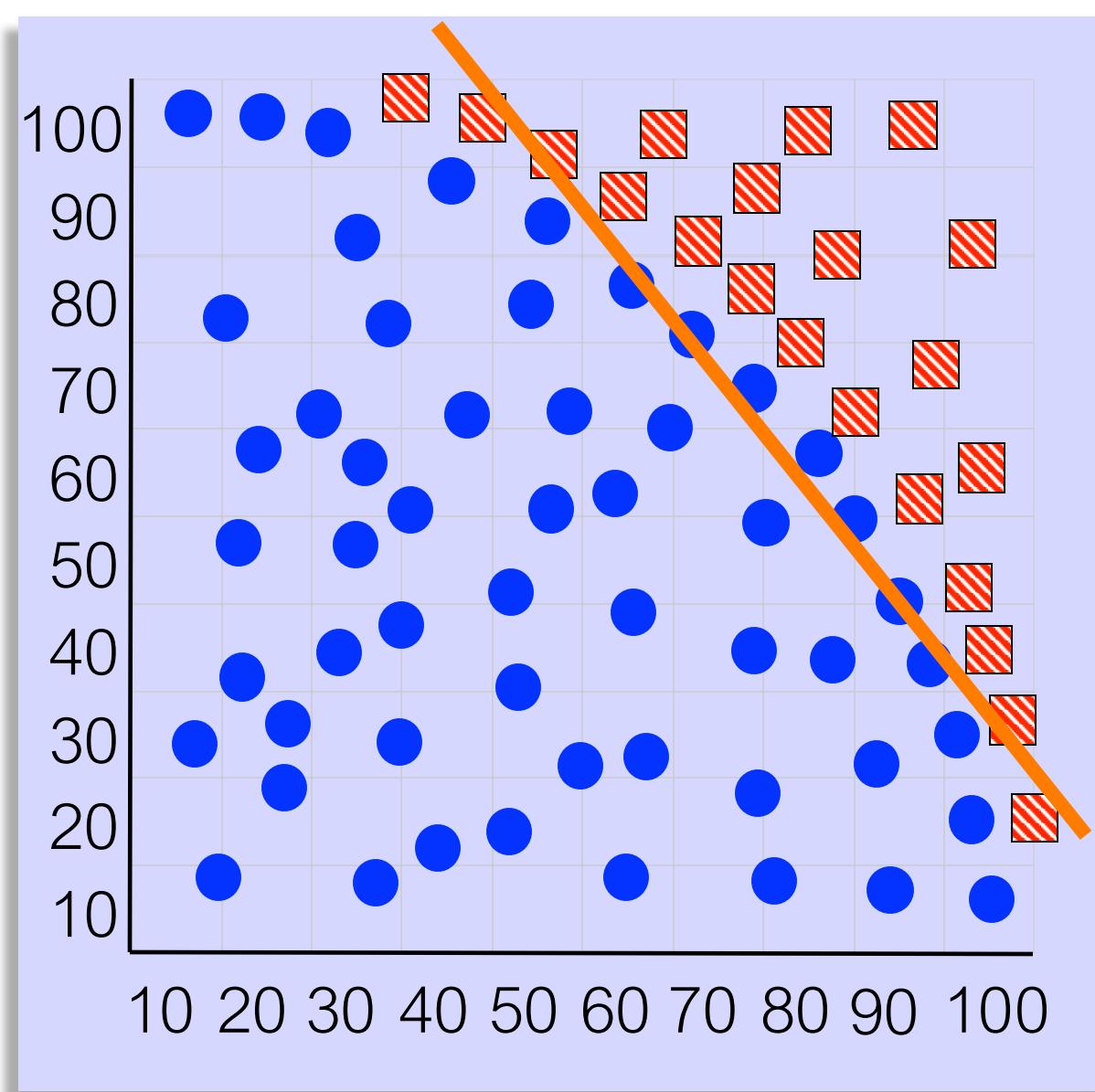


# Classificação: Problema do Pombo

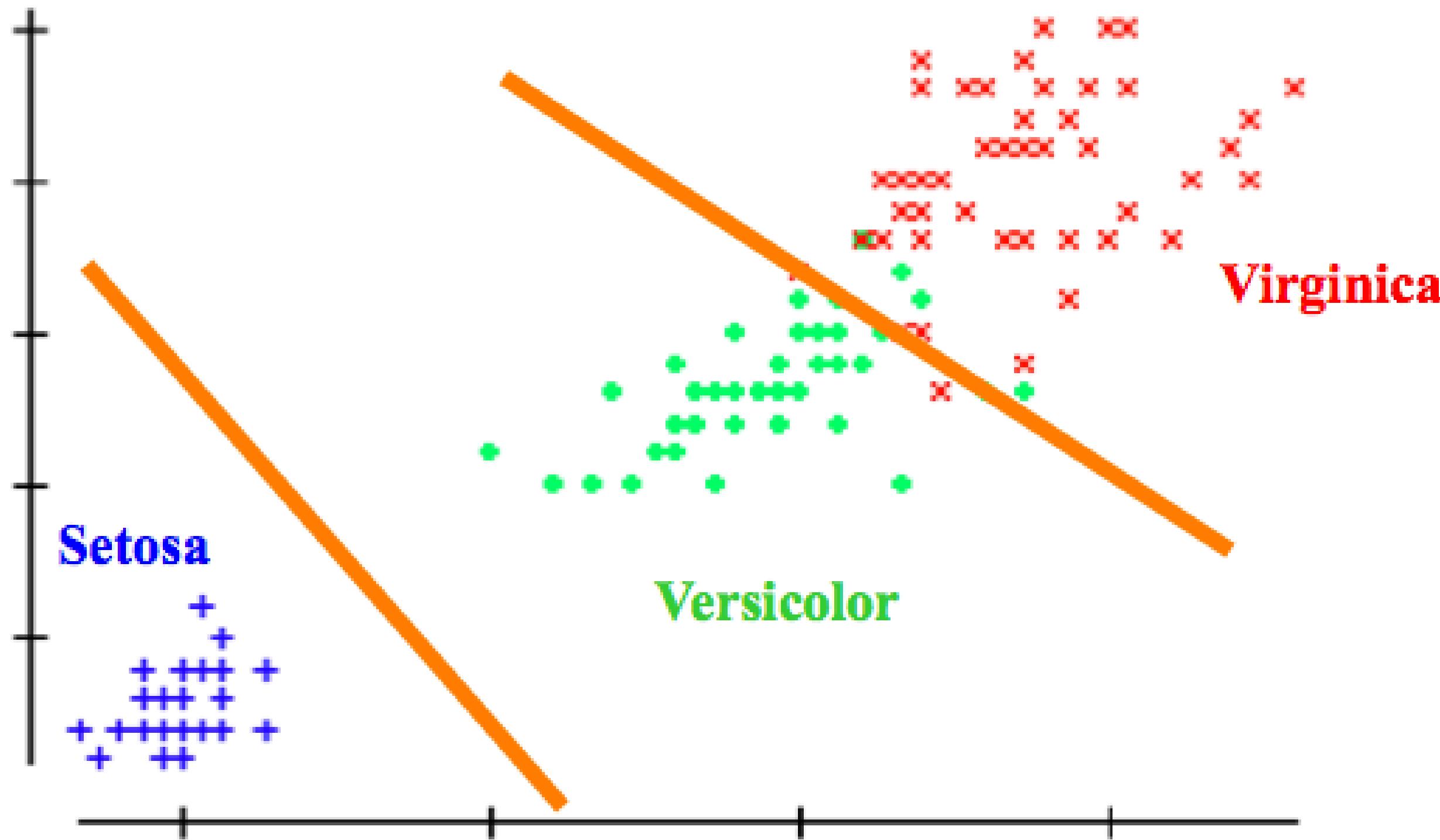
**Quais dos “Problemas do Pombo”  
podem ser resolvidos pelo  
Classificador Linear Simples?**

Problemas que podem ser  
resolvidos por um classificador  
linear são chamados de linearmente  
separáveis.

- 1) Perfeito
- 2) Inútil
- 3) Muito bom



# Classificação : Um Problema Famoso



se largura\_petala > 3.72 - (0.325 \* comprimento\_petala)

então classe = Virginica

senão

se largura\_petala...



Iris Setosa



Iris Versicolor



Iris Virginica

K

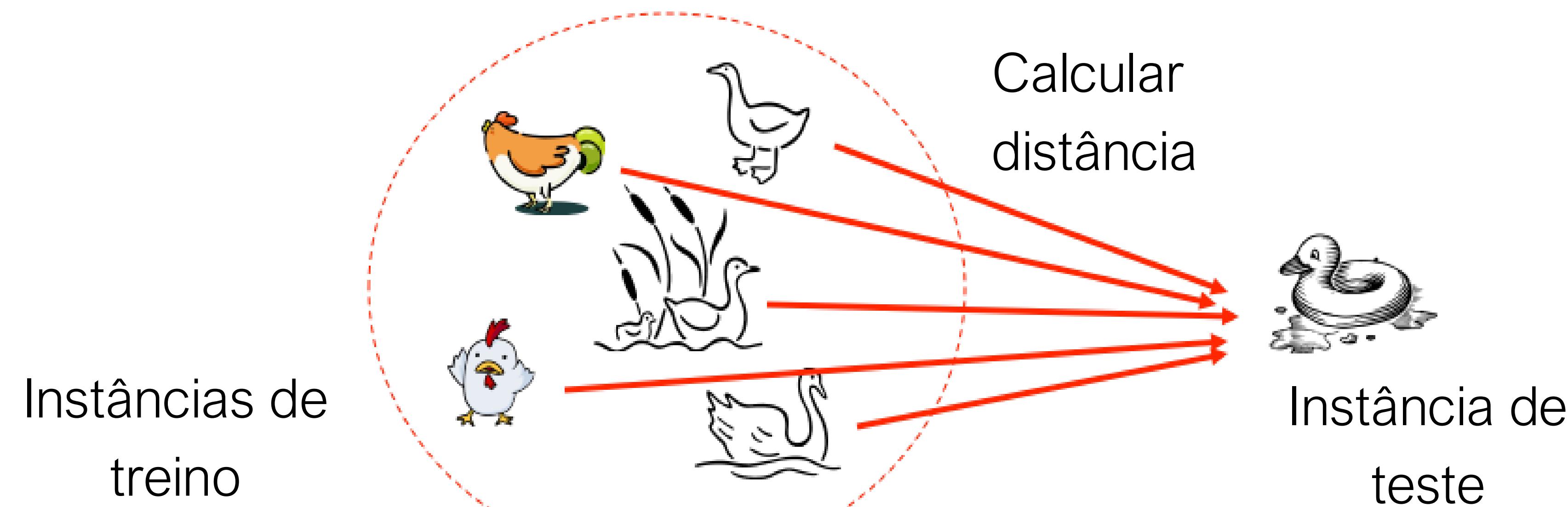
# Nearest Neighbors (KNN)

# O que é?

- ◆ O classificador dos k vizinhos mais próximos. Baseado em instâncias.

## Idéia básica

- ◆ Se caminha como um pato, faz “quack” como um pato, então provavelmente é um pato!



# O que é?

- ◆ O classificador dos k vizinhos mais próximos.

## Idéia básica

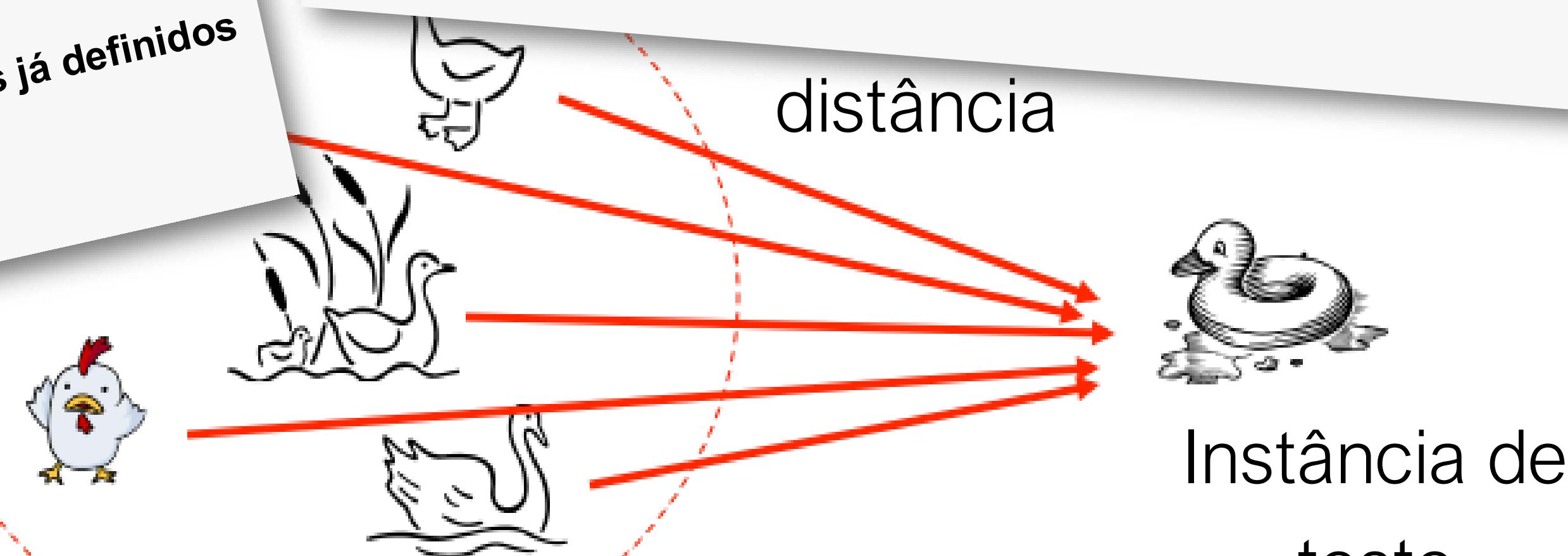
- ◆ Se caminha como um pato, faz “qu



Gian 1 second ago  
São os exemplos conhecidos, com rótulos já definidos

REPLY

Instâncias de  
treino



Diego 1 second ago

"instâncias" refere-se aos dados de entrada usados para o treinamento e teste do algoritmo.

REPLY

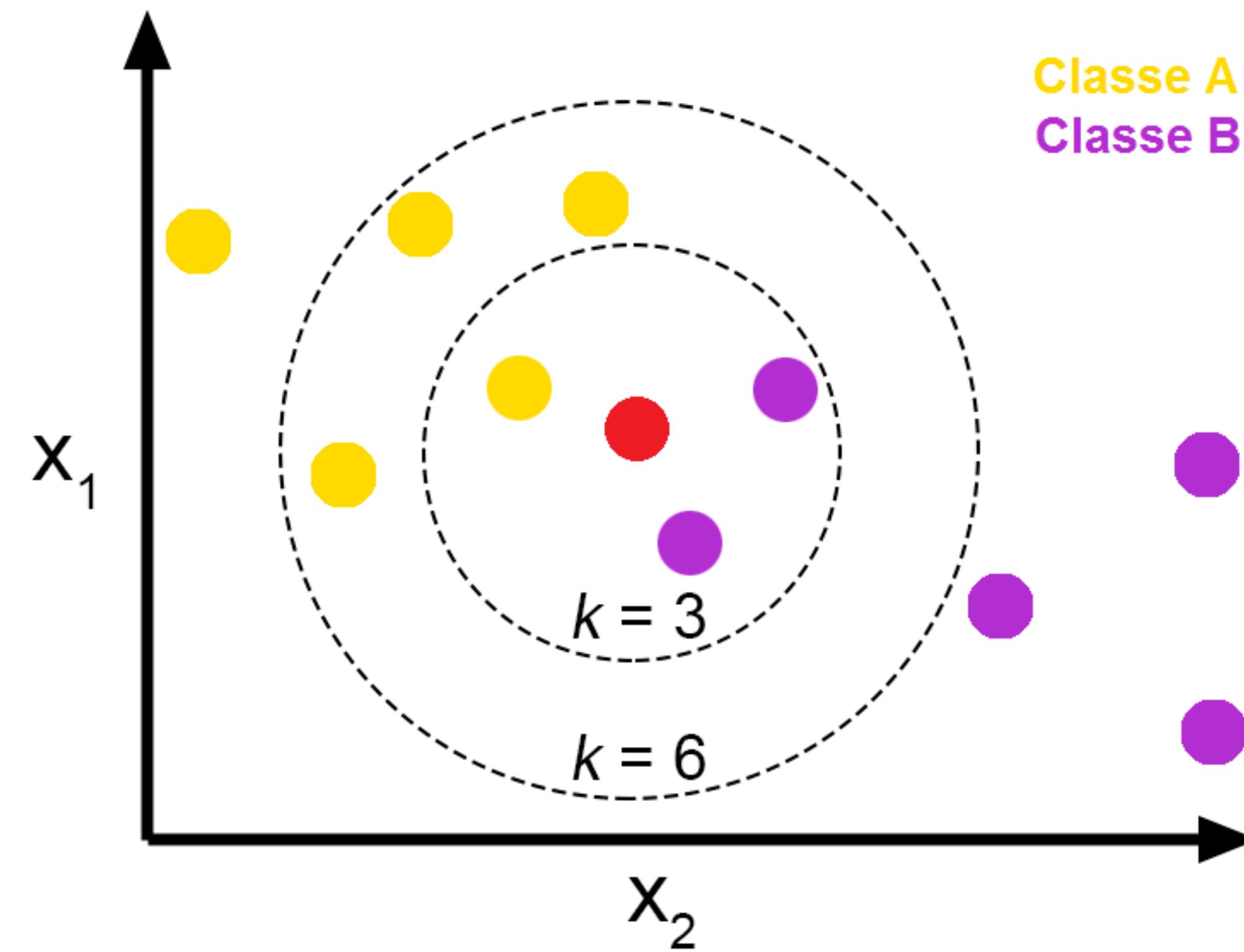


Joaquim 1 second ago

O KNN compara o novo dado de teste com os dados de treino, calculando a distância para achar os mais próximos.

REPLY

# O que significa K?



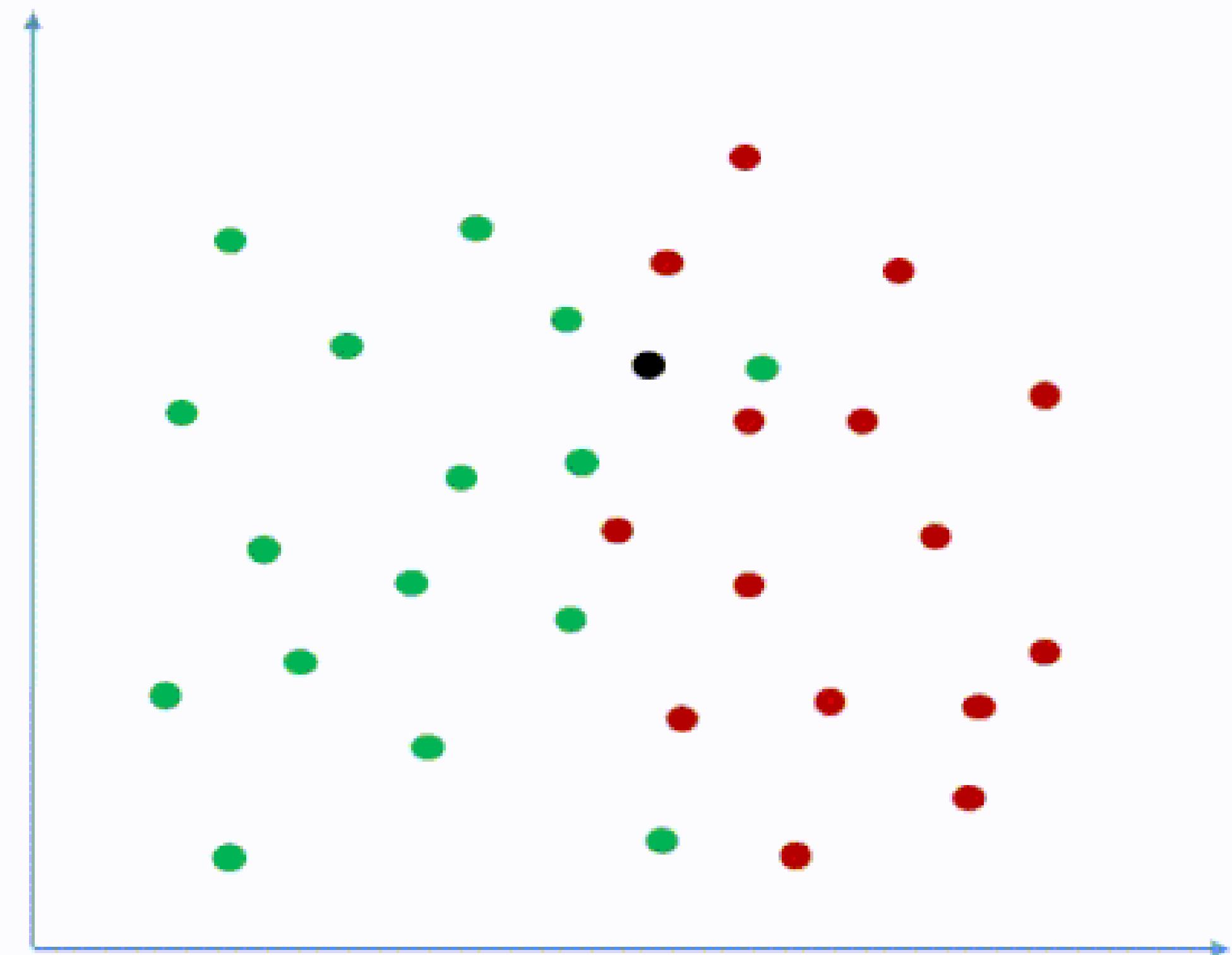
<https://medium.com/brasil-ai/knn-k-nearest-neighbors-1-e140c82e9c4e>

- ◆ K representa o número de pontos de dados próximos ao ponto de dados de interesse que usaremos para encontrar a classe.

# Como?

- ◆ O dado incluído é classificado na região com mais dados próximos de uma mesma classe.
- ◆ Ele não “compara” o dado não classificado com todos os outros de fato, mas ele executa um cálculo matemático para medir a distância entre os dados para fazer sua classificação

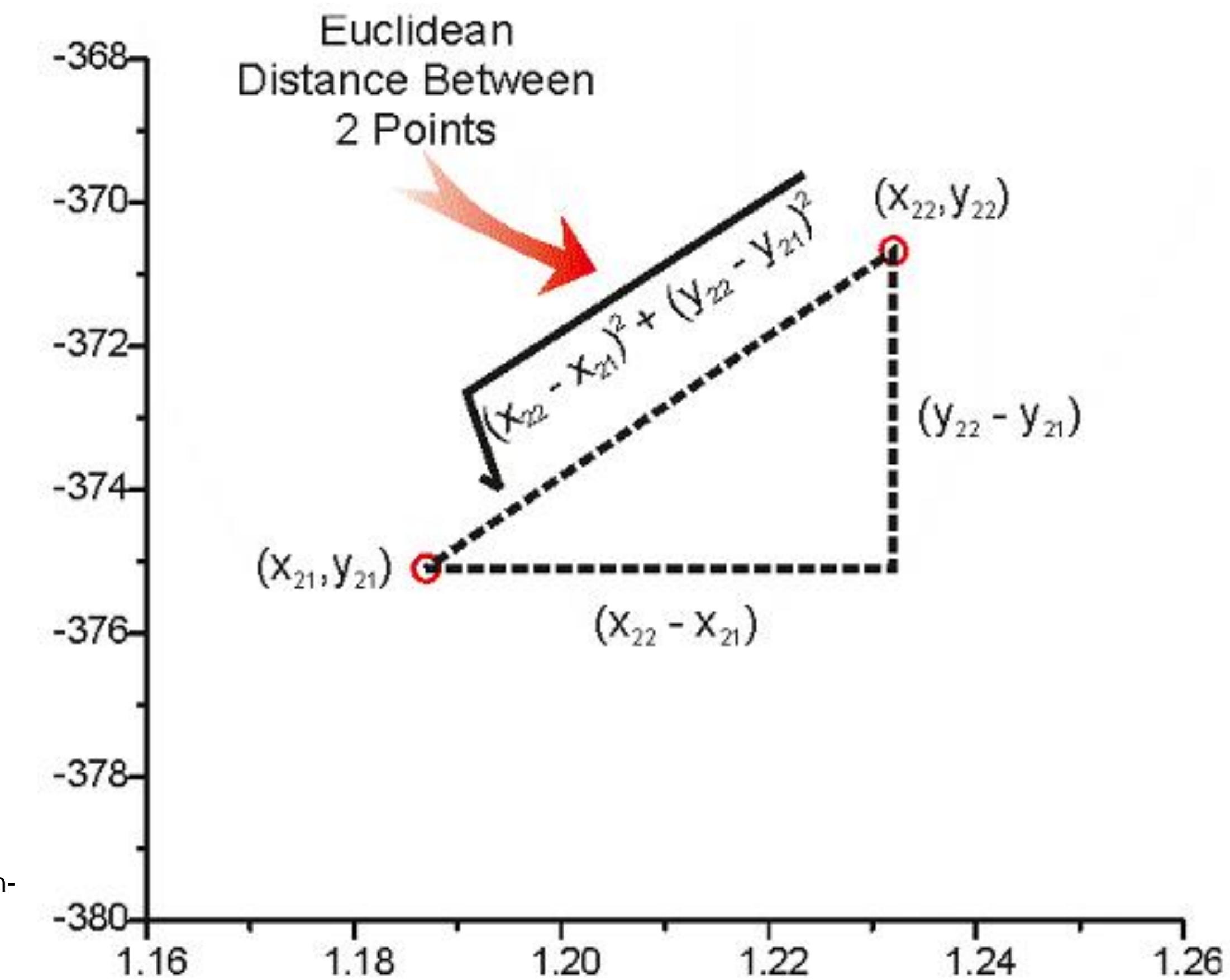
K-Nearest Neighbors Classification



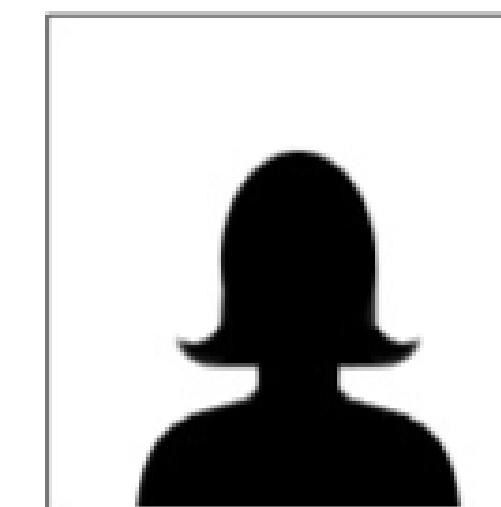
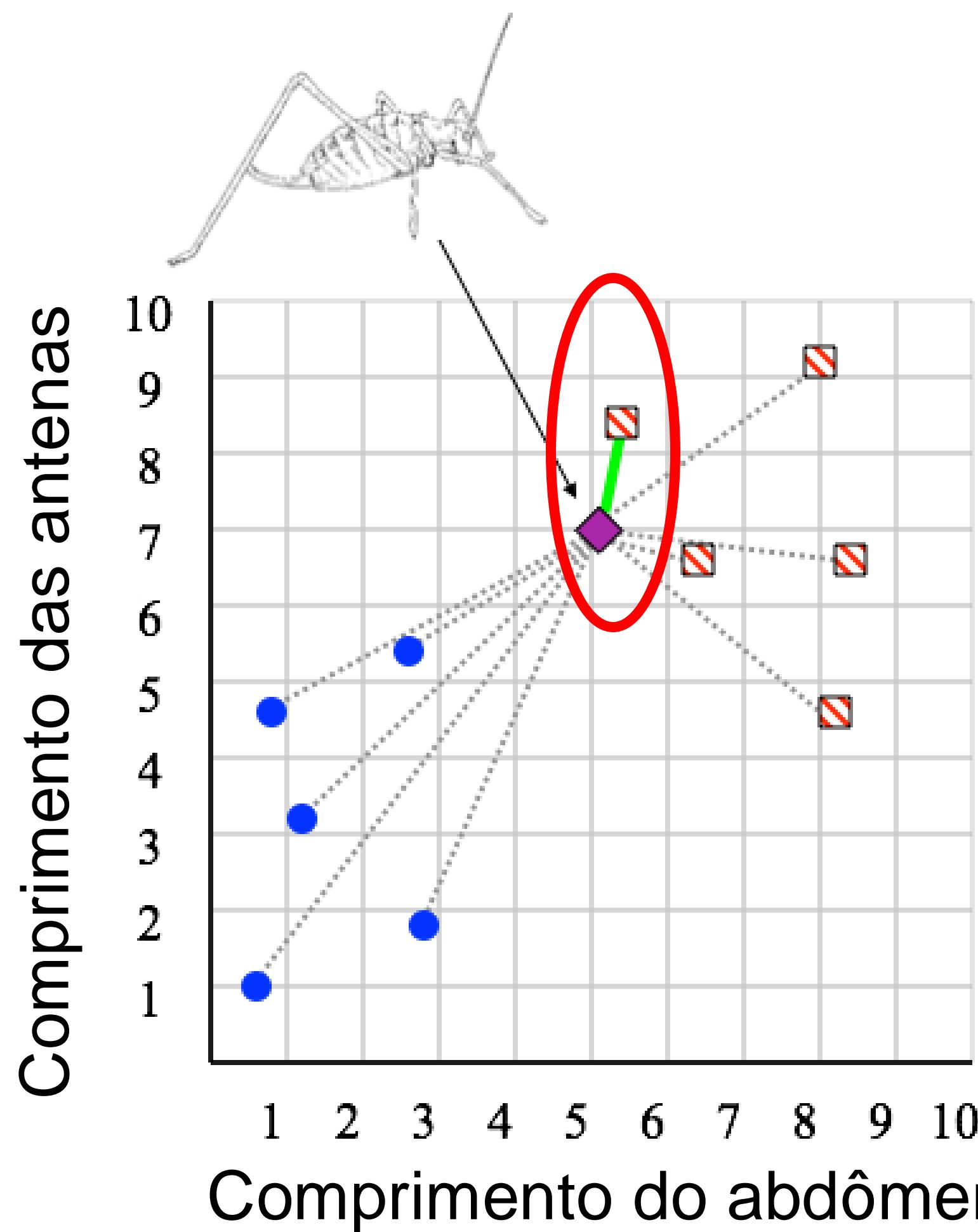
# Como?

- ◆ Esse cálculo pode ser qualquer um que meça a distância entre dois pontos, como por exemplo: Euclidiana, Manhattan, Minkowski, Ponderada e etc.

<https://medium.com/@tpreethi/introduction-to-k-nearest-neighbors-knn-algorithm-python-implementation-9c387915f31d>



# KNN com K=1



Evelyn Fix  
1904-1965



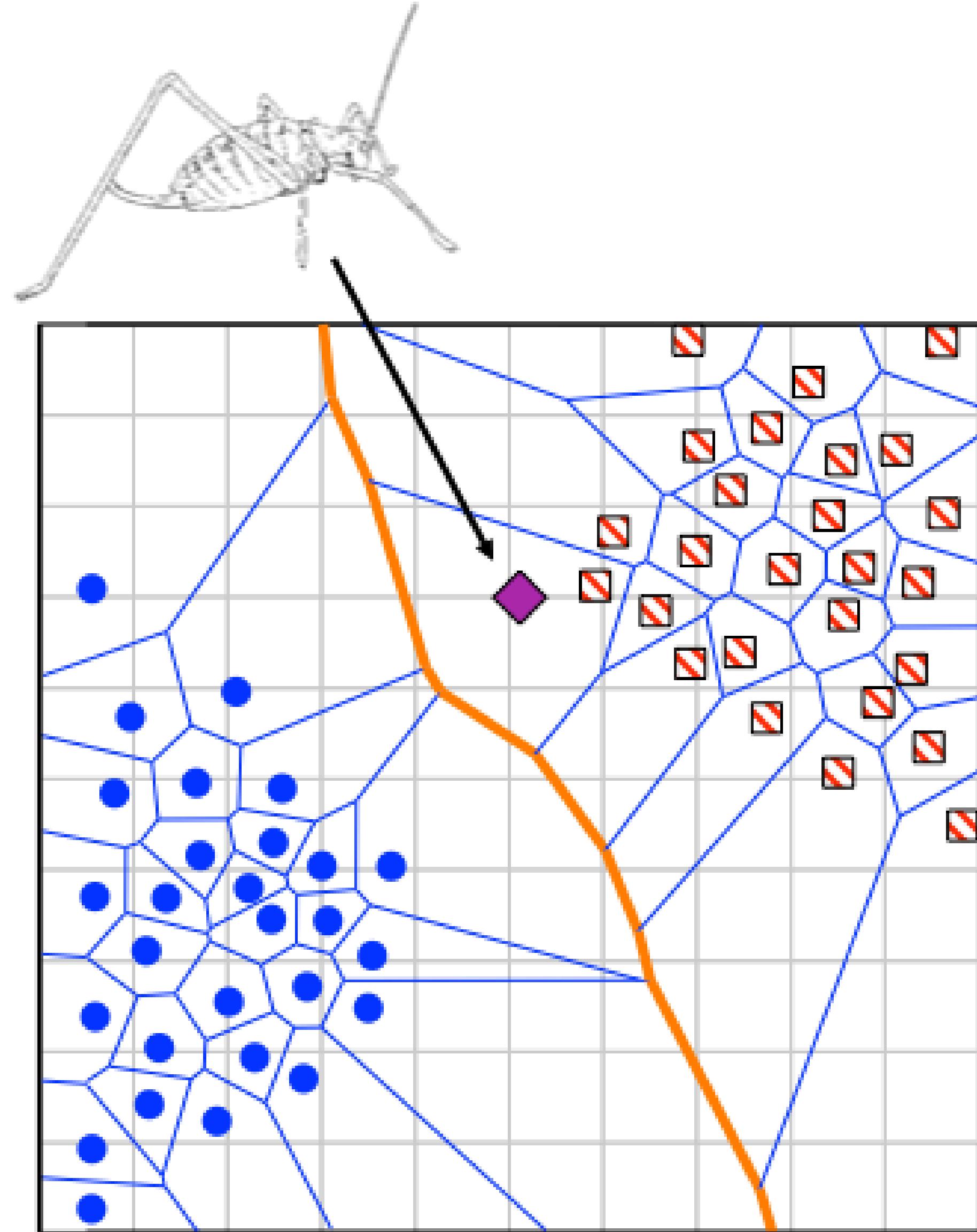
Joseph Hodges  
1922-2000

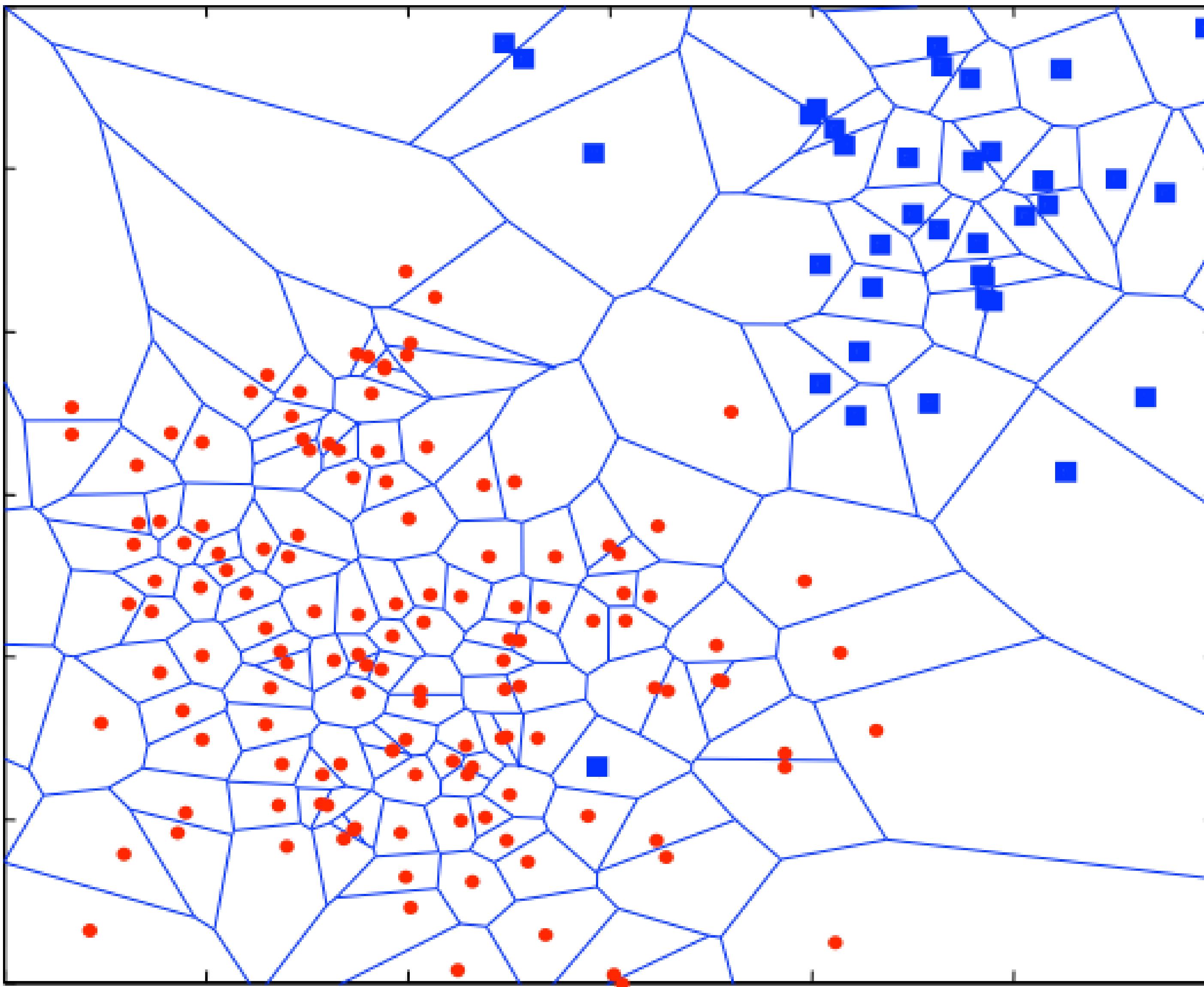
Criado em  
1951

se o exemplo mais próximo ao exemplo desconhecido é da classe Esperança  
então  
classe é Esperança  
senão  
classe é Gafanhoto

# KNN com K=1

- É possível visualizar o K-NN (com  $k=1$ ) em termos de uma fronteira de decisão!
- Fronteiras implícitas que delimitam as zonas pertencentes a cada exemplo de treino
- Esse tipo de divisão é chamada de **Diagrama de Voronoi** compostas por poliedros convexos.





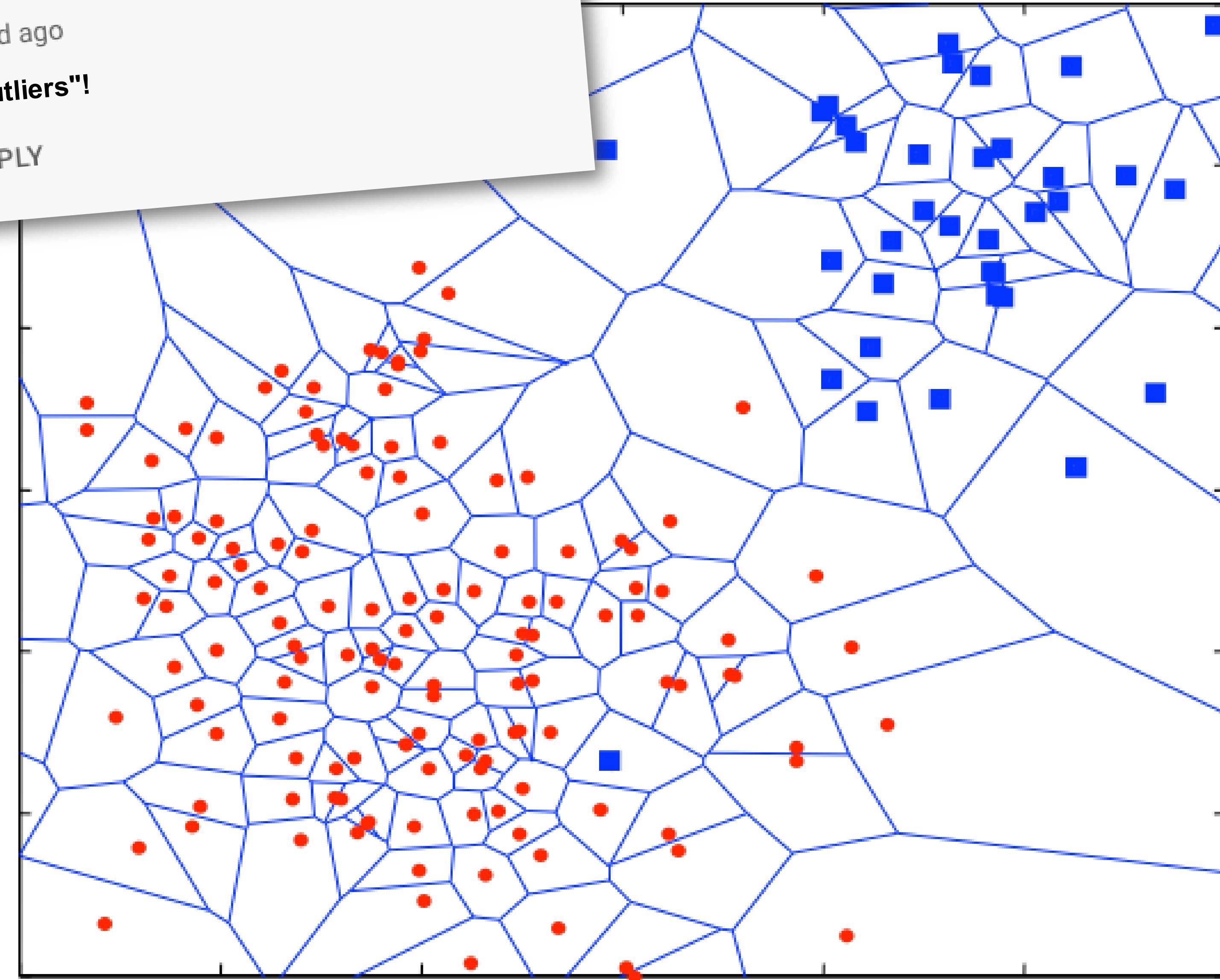


Diego 1 second ago

É sensível a “outliers”!

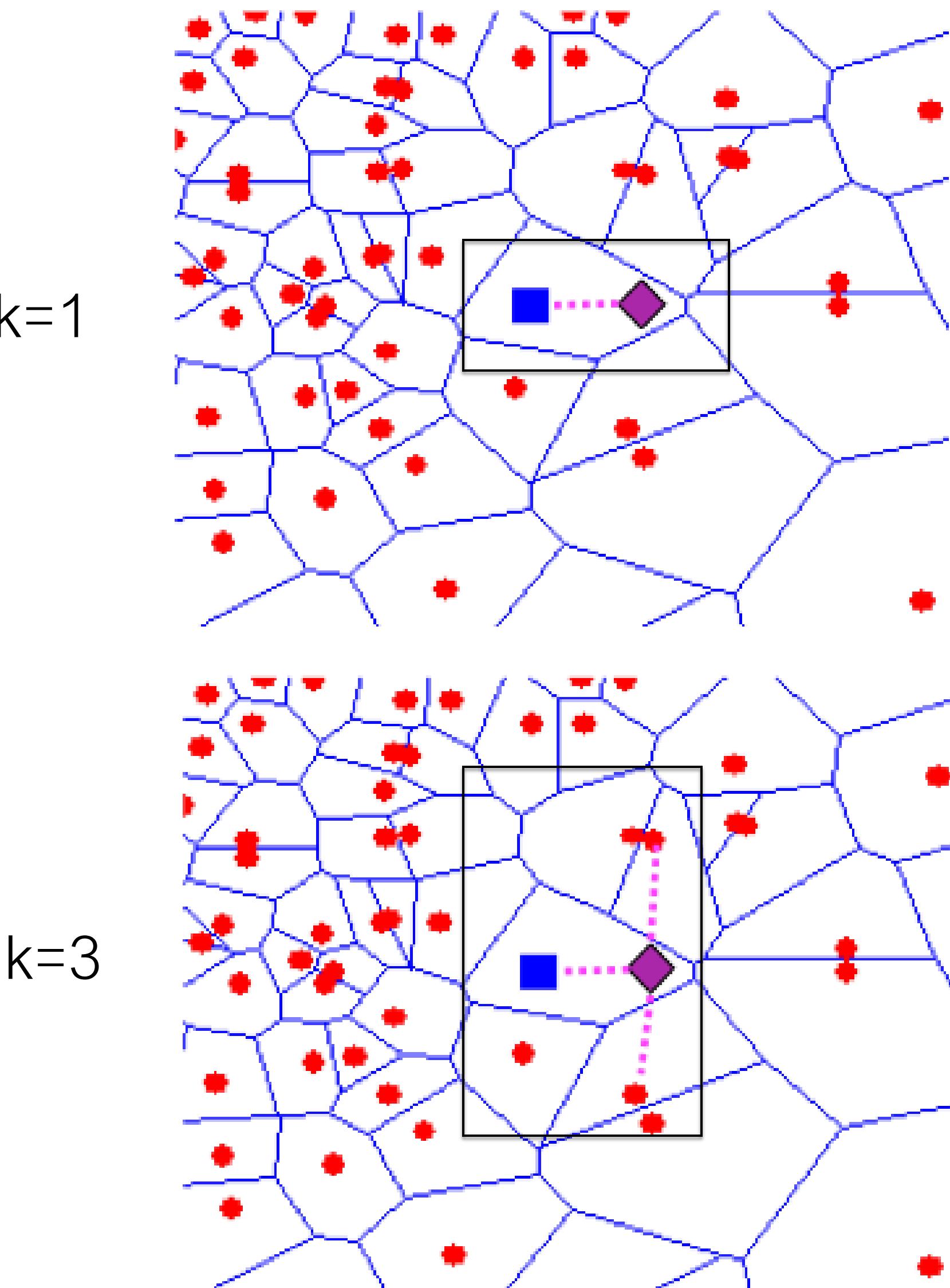


REPLY



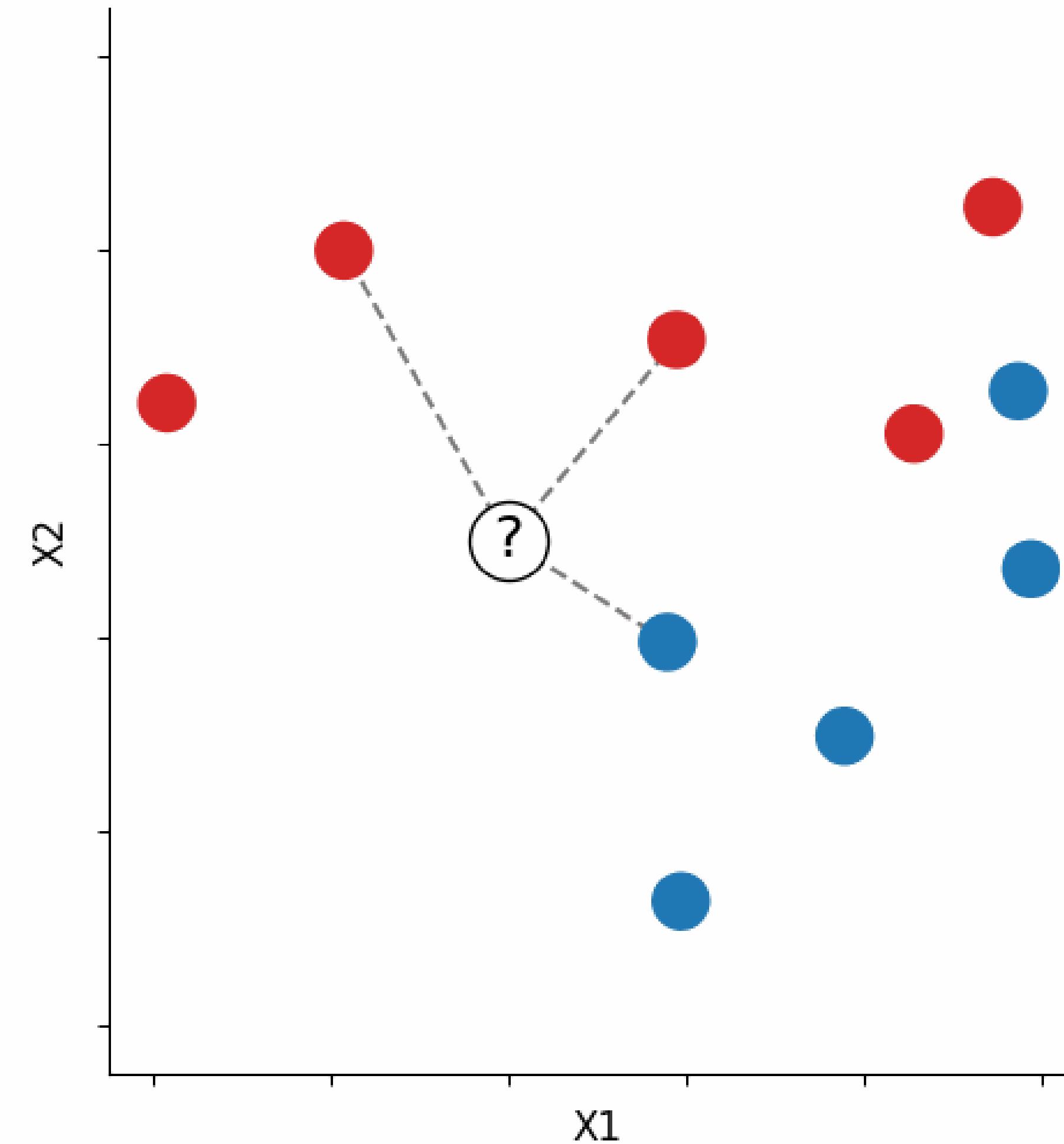
# Solução

- ◆ Aumentar o **valor de K!**
- ◆ Mede-se a distância para os **k** exemplos mais próximos.
- ◆ Computa o voto da maioria para definição da classe do exemplo desconhecido.
- ◆ O objeto de teste é classificado na classe mais votada.

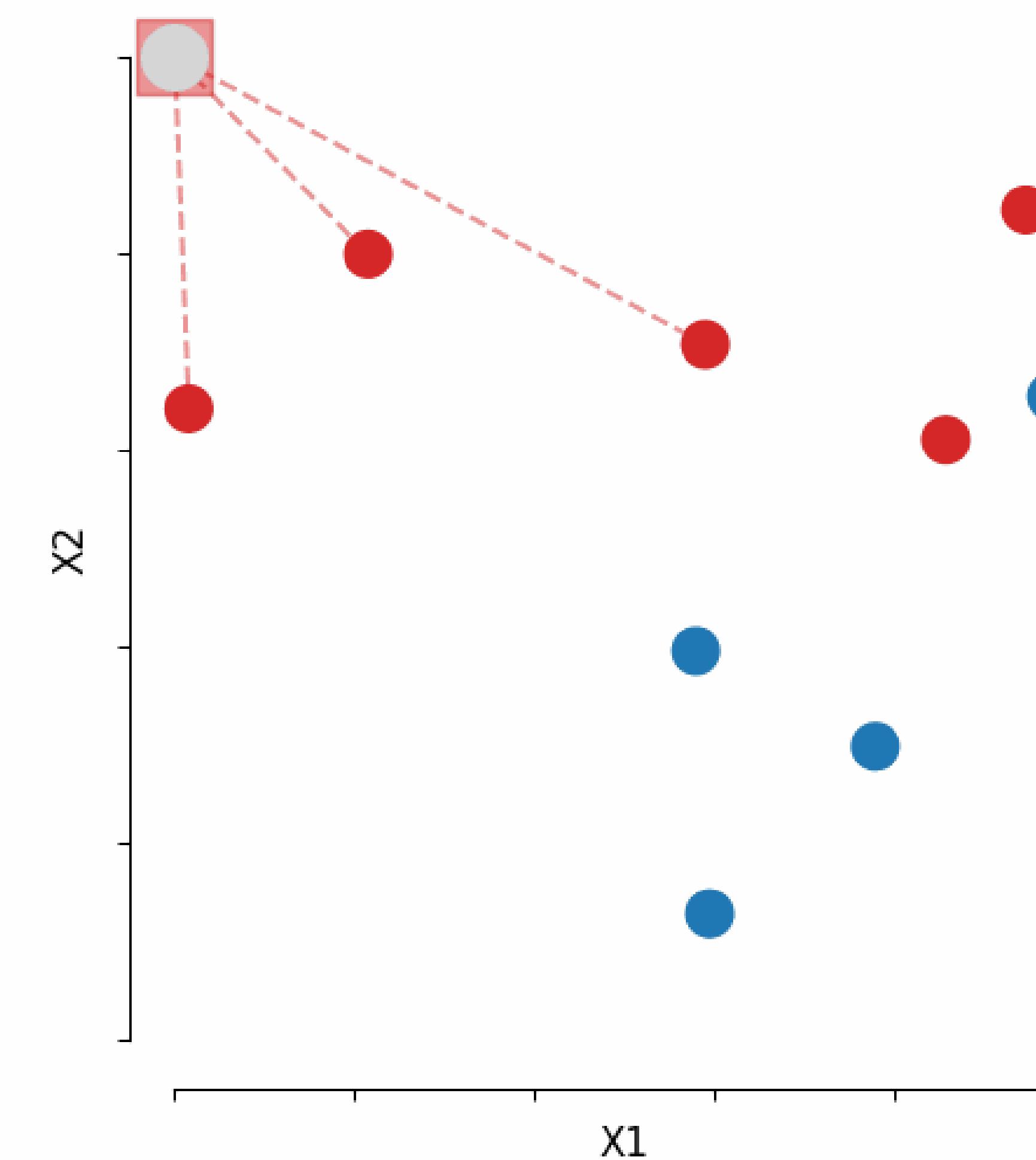


# $k$ NN classifier | $k = 3$

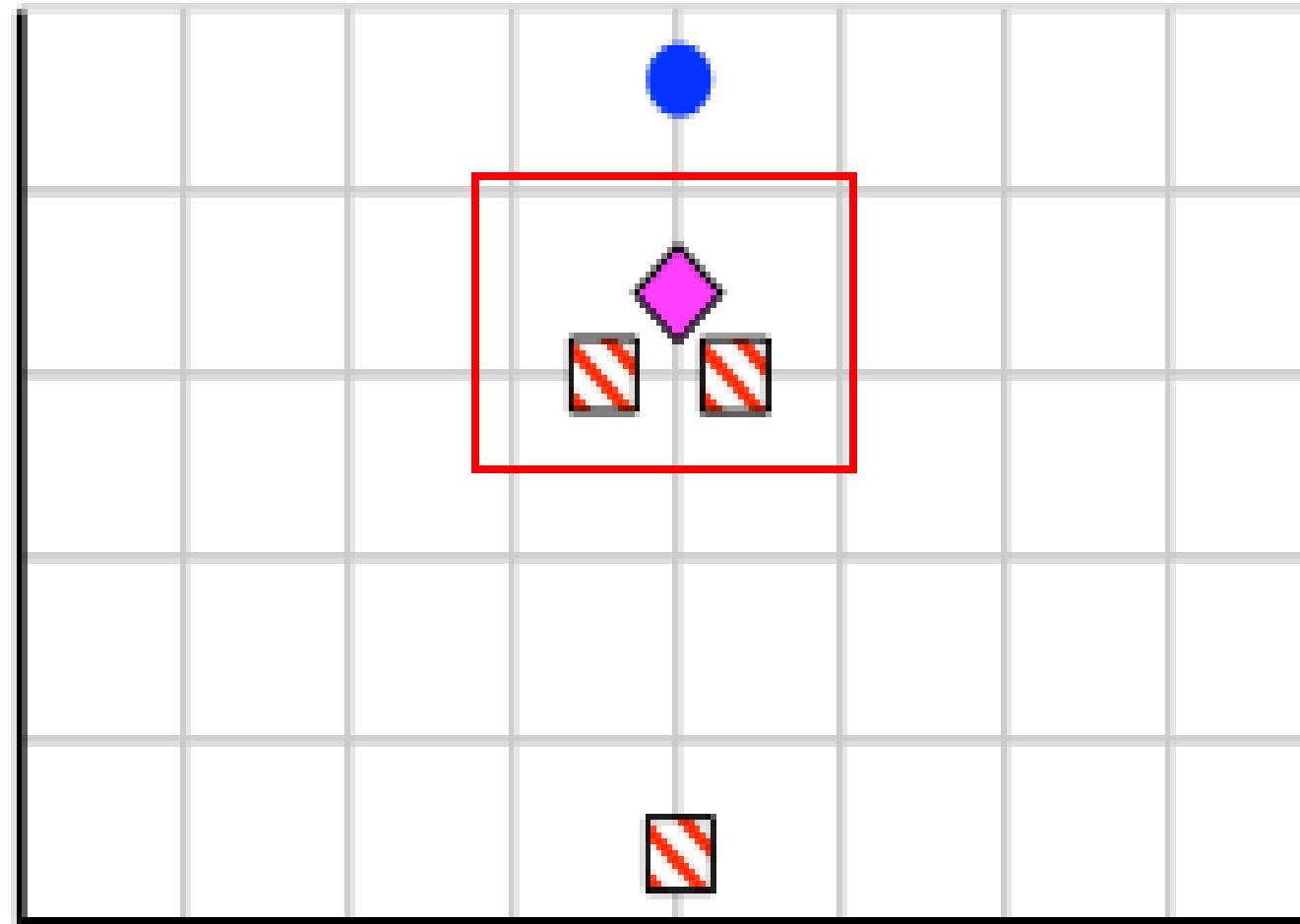
Classifying a single test point



Drawing the decision surface



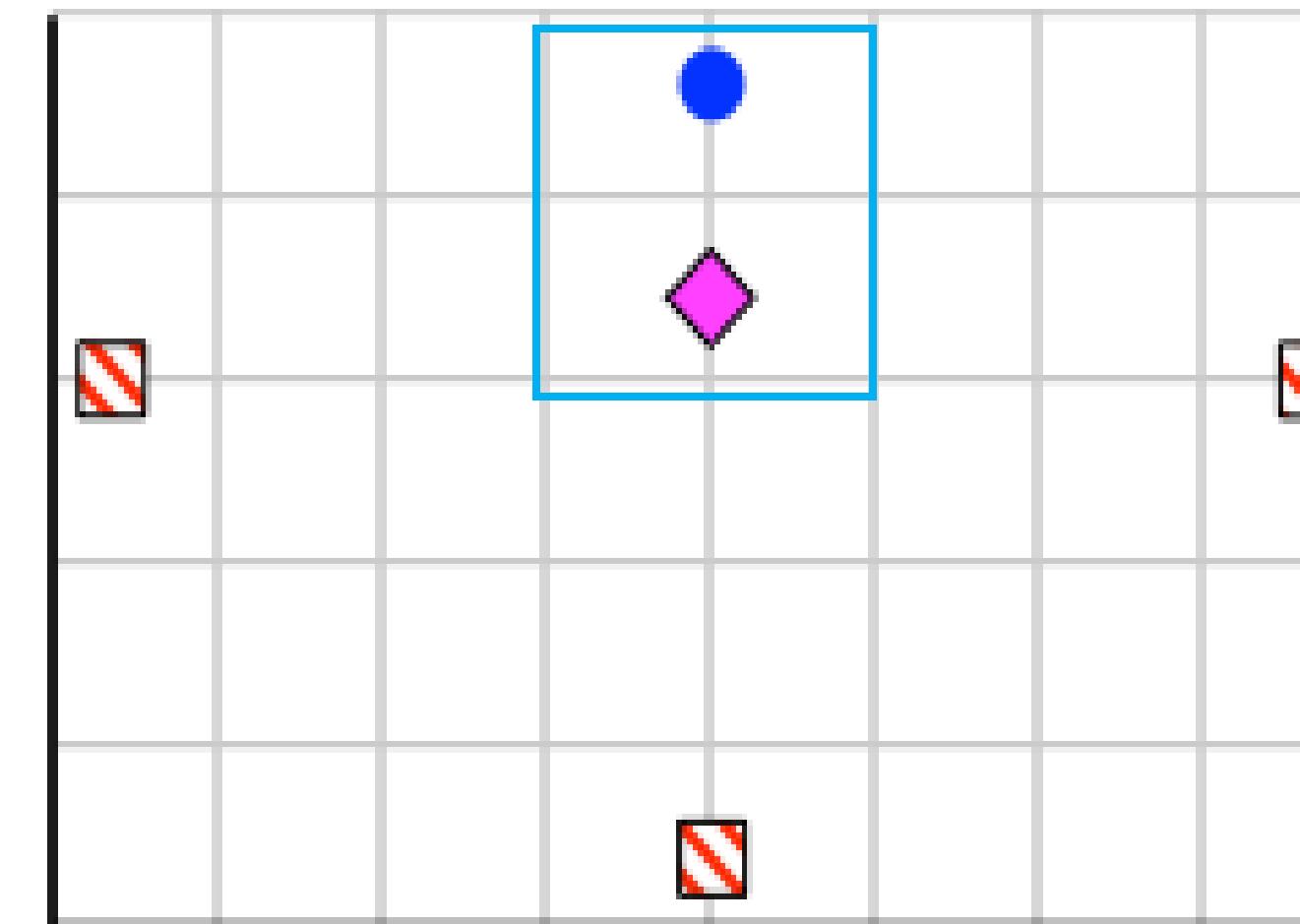
# Sensível às unidades de medidas utilizadas.



Eixo x em centímetros.

Eixo y em reais.

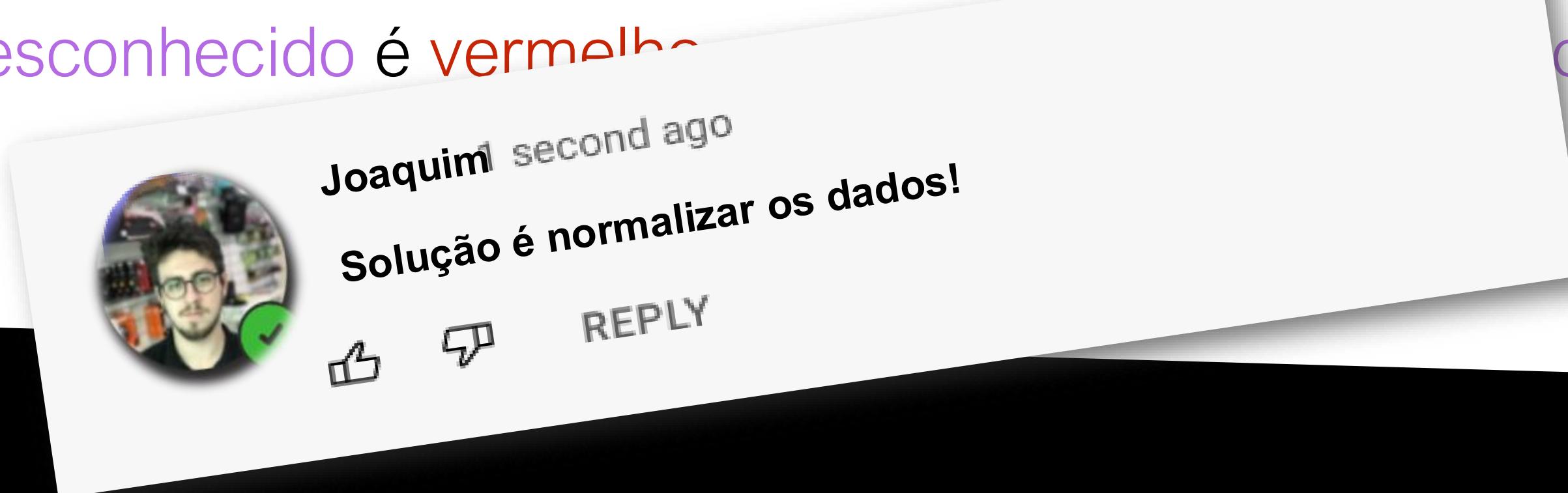
Exemplo mais próximo do rosa  
desconhecido é vermelho



Eixo x em milímetros.

Eixo y em reais.

Exemplo mais próximo do rosa  
desconhecido é azul.

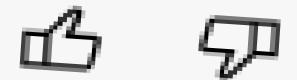


- ◆ A escolha do valor de k mais apropriado para um problema de decisão específico pode não ser trivial.
- ◆ Qual seria a classe do objeto desconhecido abaixo se o K fosse igual a 3?
- ◆ E se K fosse igual a 5?



Joaquim 1 second ago

Normalmente o valor de k é pequeno e ímpar ( $k = 3, 5, 7, \dots$ )



REPLY

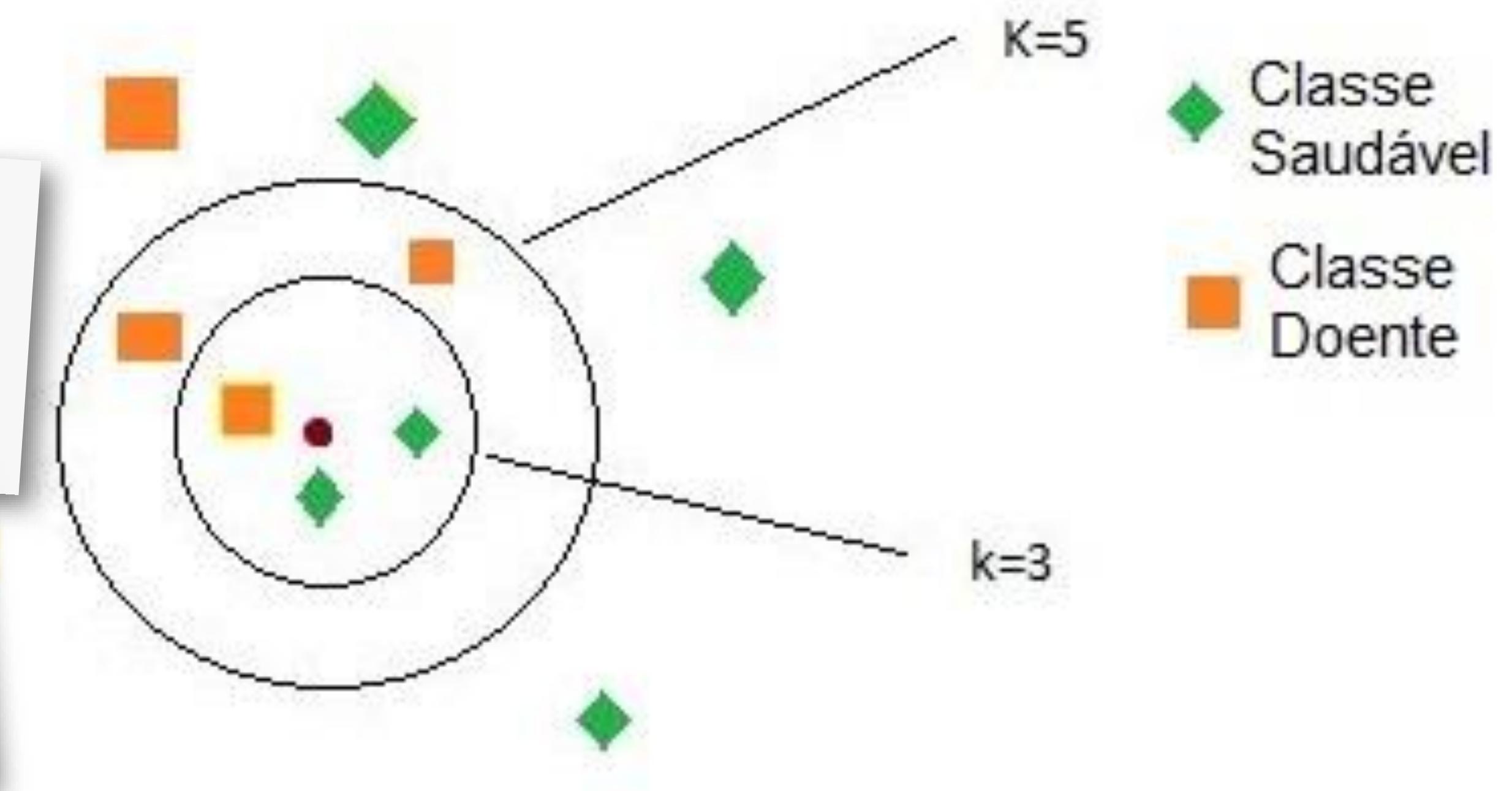


Gian 1 second ago

Para evitar empates em problemas de classificação, não é usual utilizar  $k=2$  ou valores pares



REPLY



# Os Pros e Contras

## Aspectos Positivos

- ◆ Algoritmo de treinamento é **simples**.
- ◆ Não gera um modelo.
- ◆ Algoritmo é incremental: quando novos exemplos de treinamento estão disponíveis, basta armazená-los na memória.
- ◆ Existem apenas dois parâmetros necessários para a implementação, o **valor de K** e a **função de distância**

# Os Pros e Contras

## Aspectos Negativos

- ◆ Algoritmo **lazy**.
- ◆ **Predição custosa**: classificar um objeto de teste requer calcular a distância desse objeto a todos os objetos de treinamento.
- ◆ Afetado pela presença de atributos redundantes e de atributos irrelevantes.
- ◆ O número de atributos define o número de dimensões do espaço, o qual **cresce exponencialmente**.
- ◆ Não aceita atributos categóricos

# *Matriz de Confusão*

# O que é?

- Tabela que avalia o desempenho de modelos de classificação.
- Compara valores reais com as previsões do modelo.  
**Estrutura Básica (Problema Binário)**

Algoritmo de treinamento é **simples**.

- ◆ **True Positive (TP)**: Previsões corretas de positivo.
- ◆ **True Negative (TN)**: Previsões corretas de negativo.
- ◆ **False Positive (FP)**: Falsos positivos (erro tipo I).
- ◆ **False Negative (FN)**: Falsos negativos (erro tipo II).

# Métricas

$$Acurácia = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precisão = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$f-score = 2 * \frac{precision * recall}{precision + recall}$$

## Scikit Learn

- ◆ temos acessos a várias métricas que podem nos ajudar a entender melhor nossos conjunto de dados e a matriz de confusão.

<https://scikit-learn.org/stable/>

```
from sklearn.metrics import confusion_matrix  
from sklearn.metrics import classification_report
```

- ◆ Utilizando o método “classification\_report” da biblioteca sklearn.metrics.

# Métricas

## Acurácia

- ◆ É como o modelo se sai no geral. Ele mede a quantidade de acertos totais (tanto as previsões de quem gosta quanto de quem não gosta de chocolate) sobre todas as tentativas de prever. Então, se o modelo acerta em 90% das vezes (gosta ou não gosta), ele tem 90% de acurácia.

## Exemplo:

- ◆ De 100 pessoas, o modelo acerta quem gosta e quem não gosta de chocolate em 90 vezes. Acurácia de 90%.

# Métricas

## Precisão

- ◆ Foca só nas previsões positivas — ou seja, quando o modelo diz que alguém gosta de chocolate. A precisão nos diz quantas dessas previsões de "gosta" estão realmente corretas.

### Exemplo:

- ◆ O modelo diz que 20 pessoas gostam de chocolate, mas só 15 realmente gostam. Nesse caso, a precisão é  $15/20 = 75\%$ .

# Métricas

## Em resumo:

- ◆ Acurácia é o desempenho geral do modelo, incluindo todos os acertos (gosta e não gosta).
- ◆ Precisão mostra o quanto podemos confiar quando o modelo diz que alguém gosta de chocolate.

Cada uma é útil dependendo do que você quer dar mais importância: o desempenho geral (acurácia) ou a confiabilidade dos positivos (precisão).

## Sensibilidade

- ◆ Proporção de verdadeiros positivos.
- ◆ Capacidade do sistema em predizer corretamente a condição para casos que realmente a têm.

## Especificidade

- ◆ Proporção de verdadeiros negativos.
- ◆ Capacidade do sistema em predizer corretamente a ausência da condição para casos que realmente não a têm.

# Métricas para Avaliação de Desempenho

==== Confusion Matrix ====

	a	b	c	d	e	---	classified as
a	12	0	0	3	0		a = eleito
b	1	0	0	2	0		b = media
c	0	0	8	4	0		c = naoeleito
d	1	0	0	10	0		d = suplente
e	0	0	0	0	0		e = naoinformado

**classificados corretamente**  
**classificados erroneamente**

# Limitações de Precisão

- Considere um problema de 2 classes
  - Número de exemplos da Classe 0 = 9990
  - Número de exemplos da Classe 1 = 10
- Se o modelo prevê que todos os objetos são da classe 0, a acurácia é  $9990/10000 = 99.9\%$ 
  - Acurácia é enganosa porque o modelo não detectou nenhum exemplo da classe 1.

# Métodos de Amostragem

- ◆ Calcular o erro estimado do modelo durante o treino.
- ◆ O cálculo do erro estimado auxilia o algoritmo de aprendizagem a fazer uma seleção do modelo.

**Objetivo:** encontrar o modelo com uma complexidade precisa que não é suscetível a overfitting.

**Métodos comumente utilizados:**

- ◆ Holdout
- ◆ Validação Cruzada (Cross-Validation)
- ◆ Bootstrap

# Métodos de Amostragem

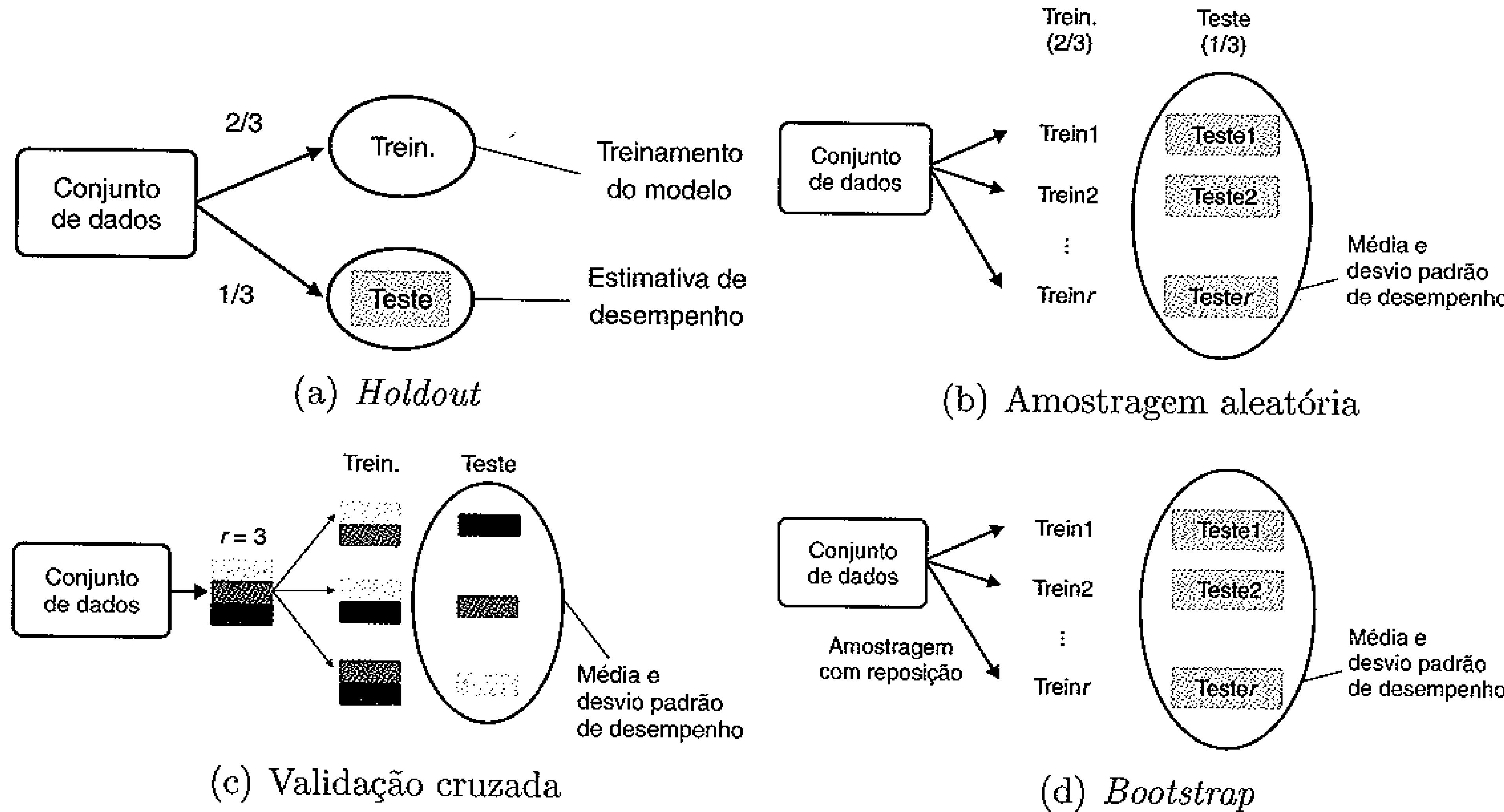




Foto de Gælle Marcel,  
disponível na Unsplash.  
Editada pelo autor.

# *Mão na Massa*

# K-NN - Exercício

Descubra a classe do exemplo desconhecido com k=1 e com k=3

Objeto	X	Y	Classe
1	5	9	□
2	7	9	□
3	8	9	□
4	9	8	□
5	5	7	□
6	7	7	□
7	8	7	□
8	7	6	□
9	9	6	□
10	9	4	□
11	6	2	□
12	2	5	○
13	4	4	○
14	1	3	○
15	2	3	○
16	3	3	○
17	6	3	○
18	3	2	○
19	1	1	○
20	4	1	○
21	8	1	○
22	9	1	○
23	5	2	?



$$D(1) = \sqrt{(5 - 5)^2 + (2 - 9)^2} = \sqrt{49} = 7$$

$$D(2) \approx 7$$

$$D(3) \approx 7$$

$$D(4) \approx 7$$

$$D(5) = 7$$

$$D(6) \approx 5$$

$$D(7) \approx 5$$

$$D(8) \approx 4$$

$$D(9) \approx 5$$

$$D(10) \approx 4$$

$$D(11) = 1$$

$$D(12) \approx 4$$

$$D(13) \approx 2$$

$$D(14) \approx 4$$

$$D(15) \approx 3$$

$$D(16) \approx 2$$

$$D(17) = 1$$

$$D(18) = 2$$

$$D(19) \approx 4$$

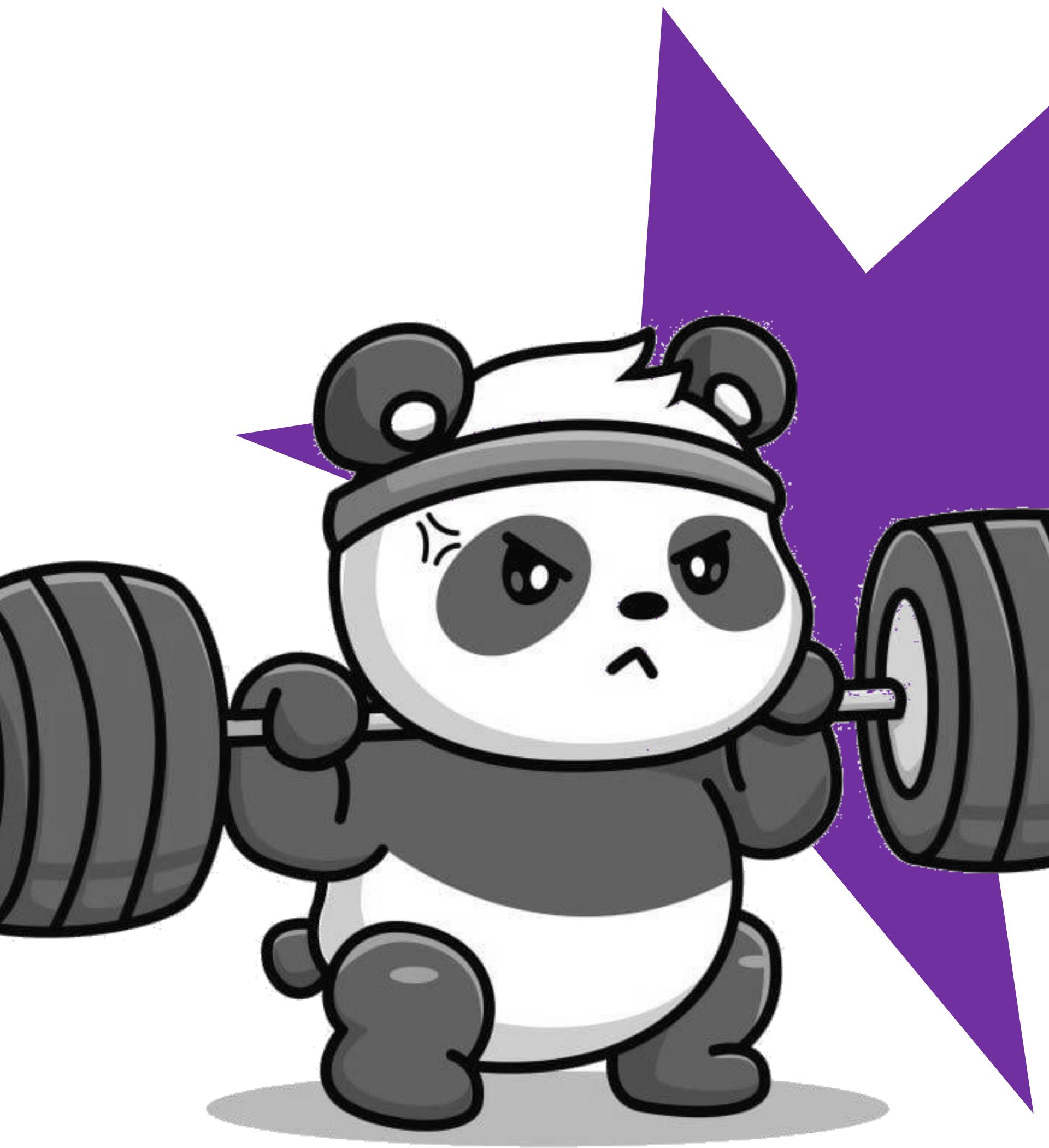
$$D(20) = 1$$

$$D(21) \approx 3$$

$$D(22) \approx 4$$

$$D(23) ????$$

Objeto	X	y	Classe
1	5	9	□
2	7	9	□
3	8	9	□
4	9	8	□
5	5	7	□
6	7	7	□
7	8	7	□
8	7	6	□
9	9	6	□
10	9	4	□
11	6	2	□
12	2	5	○
13	4	4	○
14	1	3	○
15	2	3	○
16	3	3	○
17	6	3	○
18	3	2	○
19	1	1	○
20	4	1	○
21	8	1	○
22	9	1	○
23	5	2	?



# Matriz de Confusão - Exercíc

Descubra a classe do exemplo desconhecido com k=1 e com k=3

ObjetoX	y	Classe
1	5	9
2	7	9
3	8	9
4	9	8
5	5	7
6	7	7
7	8	7
8	7	6
9	9	6
10	9	4
11	6	2
12	2	5
13	4	4
14	1	3
15	2	3
16	3	3
17	6	3
18	3	2
19	1	1
20	4	1
21	8	1
22	9	1
23	5	2
		?



# Avaliação de Desempenho

## ■ Matriz de Confusão:

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

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

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

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

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

# Métricas para Avaliação de Desempenho

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

■ Métrica mais usada: Acurácia (Accuracy)

$$\text{Acurácia} = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Sensibilidade} \quad (Recall) = \frac{TP}{TP + FN}$$

$$\text{Especificidade} \quad (Precision) = \frac{TN}{TN + FP}$$

# Atividade

•A partir do dataset escolhido para trabalhar com os métodos supervisionados e a biblioteca *scikitlearn* do Python, realize as seguintes tarefas:

1. Identifique o atributo alvo.
2. Execute o algoritmo K-NN aplicando *cross-validation* (validação cruzada) para 10 *KFolds*.
3. Execute o algoritmo K-NN aplicando *Holdout*, dividindo o dataset em 30% para teste.
4. Altere o número de k para obter uma melhor acurácia na validação.
5. Analise e compare os resultados obtidos, utilizando a matriz de confusão e computando a acurácia.

# *Trilha de* *Análise de dados*

**Módulo 1: Terminar até dia 15/10**

**Módulo 2: Terminar até dia 22/10**

**Módulo 3: Terminar até dia 29/10**

# ***Trilha de***

# ***Intro Inteligência Artificial***

**Módulo 1: Terminar até dia 25/10**

**Módulo 2: Terminar até dia 01/11**

**Módulo 3: Terminar até dia 08/11**

**Módulo 4: Terminar até dia 15/11**

**Módulo 5: Terminar até dia 22/11**

# ***Trilha de***

# ***Intro Machine Learning***

**Módulo 1: Terminar até dia 08/11**

**Módulo 2: Terminar até dia 15/11**

**Módulo 3: Terminar até dia 22/11**

**Módulo 4: terminar até dia 29/11**

**Até a próxima aula...**

**30/11/24**

# Créditos

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

## Referências

Breiman, L., Freidman, J., Olshen, R. e Stone, C. (1984). **Classification and Regression Trees.** Wadsworth International Group., USA.

Faceli, K.; Lorena, A.C.; Gama, J.; de Carvalho, A.C.P.L.F. **Inteligência Artificial: Uma abordagem de aprendizado de máquina.** LTC, Rio de Janeiro, 2011.

Quilan, R. (1979). **Discovering rules by induction from large collections of examples.** In: Michie, D. (Ed.) Expert Systems in the Microelectronic Age, p. 168-201. Edinburgh University Press.

## Referências

Quilan, J.R. (1993). **C4.5: Programs for Machine Learning.** Morgan Kaufmann Publishers Inc., San Mateo, CA, USA.

TAN, P-N; STEINBACH, M.; KUMAR,V. **Introduction to Data Mining.** Pearson, 2006.