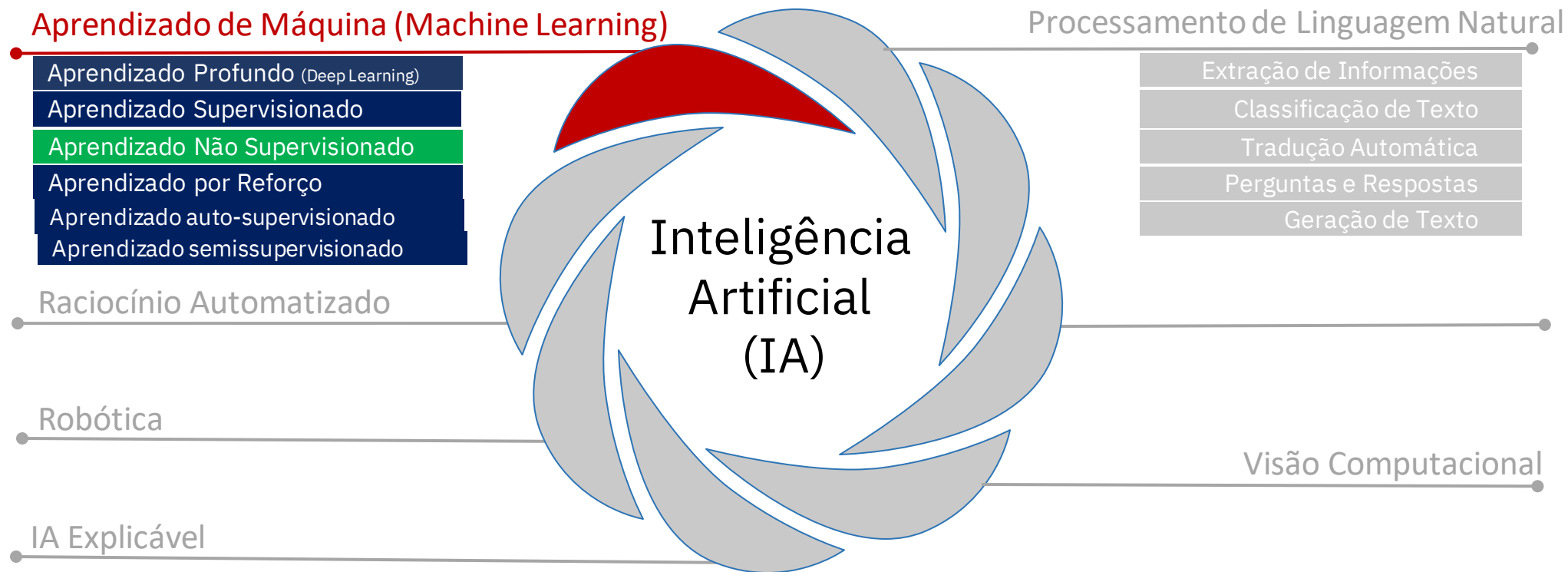
The background of the slide is an abstract composition of numerous circular and irregular splatters. On the left side, there are large, overlapping splatters in shades of orange and brown. These transition into a dense cluster of smaller, bright green and yellow-green splatters on the right side. The overall effect is a vibrant, energetic splash of color against a light, gradient background that transitions from blue on the left to white and then to a warm orange at the bottom right.

# Aprendizado não supervisionado: parte 1

Profa Silvia Moraes

# Subáreas da Inteligência Artificial



# Aprendizado Não Supervisionado

- **Não exige** que os **dados** estejam **rotulados**
- Sem crítica, **usa regularidades e propriedades estatísticas** dos dados no processo de aprendizagem.





# Aprendizado Não Supervisionado

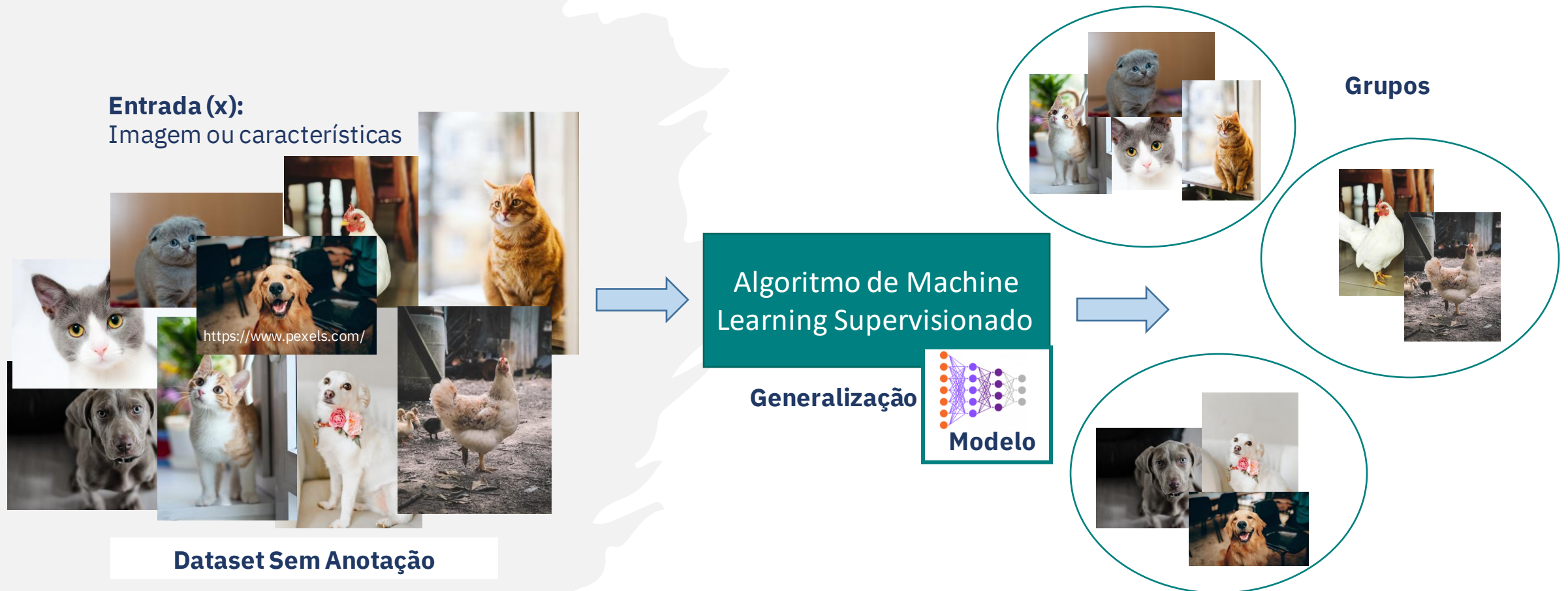
Executa **tarefas descritivas**: explora ou descreve um conjunto de dados.

- **Agrupamento**: divisão em grupos baseada em alguma regularidade ou similaridade
- **Sumarização**: descrição simples e compacta
- **Associação**: relações frequentes entre dados



# Agrupamento

Organiza dados (não classificados, sem rótulos) em grupos de acordo com alguma medida de similaridade.



# Agrupamento

- **Características:**

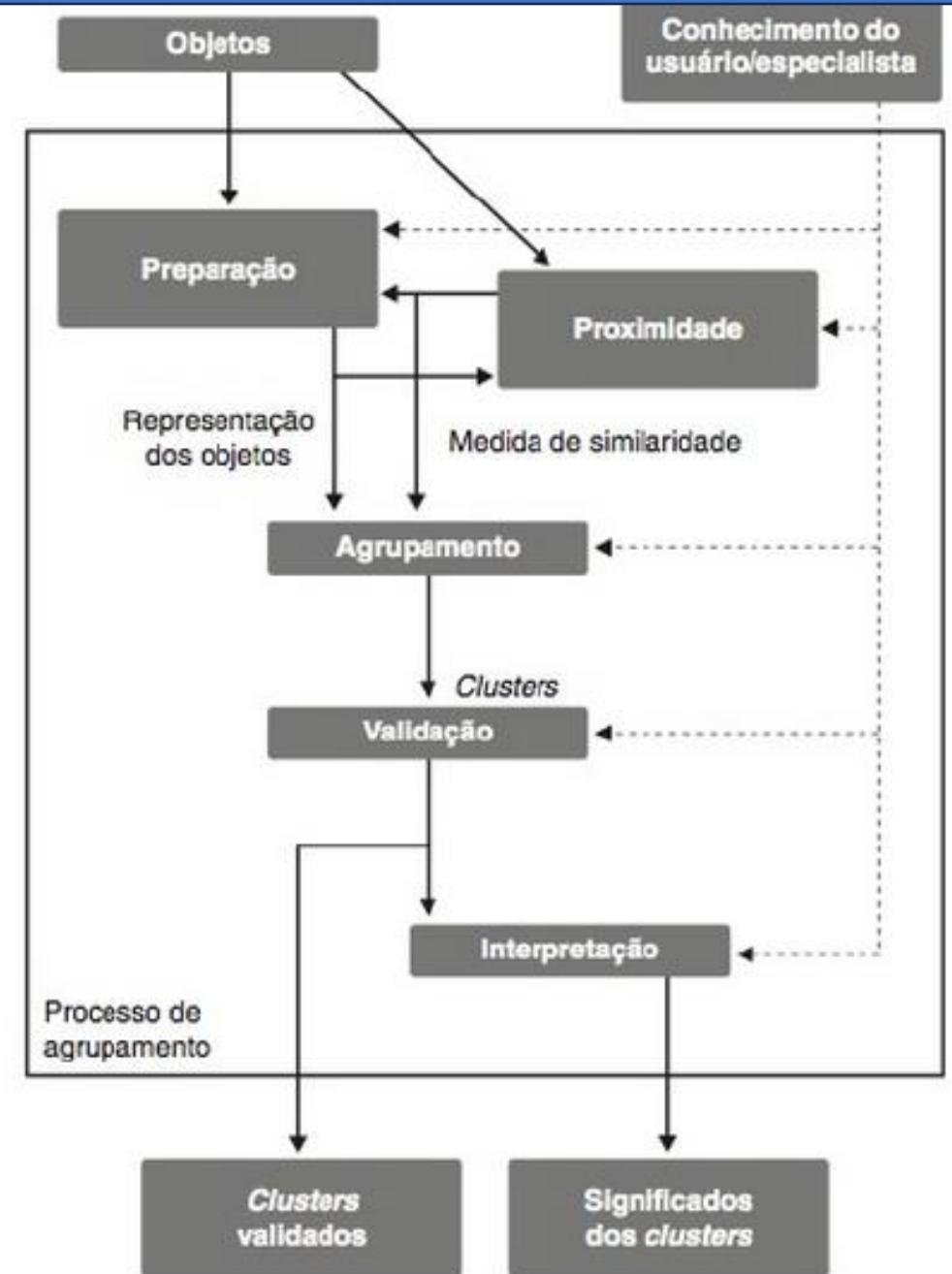
- Técnica aplicada para organizar os dados quando não há classe para predizer.
- Pode ser usado como uma etapa anterior a alguma tarefa, como por exemplo: sumarização.
- **Grupos:** formados por dados (objetos) que compartilham características (podem ser mais genéricos ou mais especializados, diferentes níveis de refinamento).  
Espera-se:
  - Alta similaridade intra-grupo.
  - Baixa similaridade entre grupos.



# Agrupamento

- **Etapas:**

- preparação,
- proximidade,
- agrupamento,
- validação e
- interpretação.



# Agrupamento

## Etapas:

- **Preparação:** inclui pré-processamento (ex: normalizações, conversão de tipos e redução de dimensionalidade) e forma de representação dos dados (ex: matriz de similaridade) para que o algoritmo de agrupamento possa ser usado.
- **Proximidade:** definição de medidas de proximidade apropriadas ao domínio e ao tipo de informação que se deseja extrair dos dados. Existem medidas para atributos quantitativos e qualitativos.



# Agrupamento

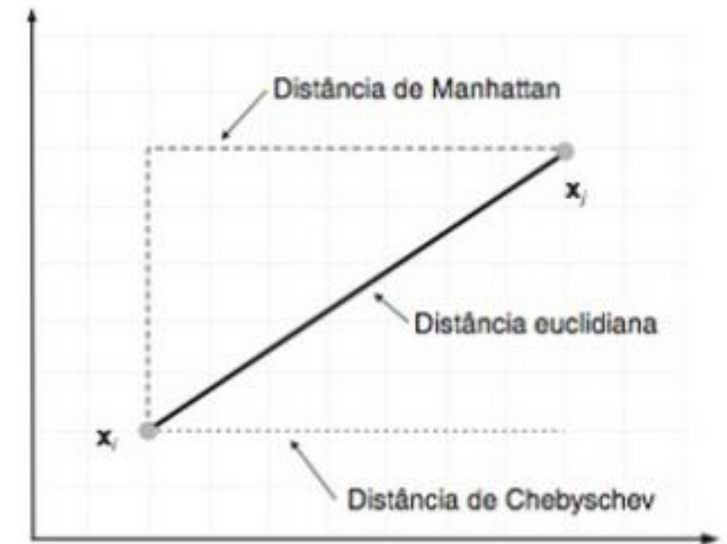
## Etapas: ...

- **Proximidade:** Medidas para atributos quantitativos e qualitativos.
  - Medidas de Distância: atributos contínuos e racionais:

- Manhattan:  $d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$  (usual para binários)

- Euclidiana:  $d(x_i, x_j) = \sqrt{\sum_{l=1}^d (x_i^l - x_j^l)^2}$

- Chebyshev (ou supremum):  $d(x_i, x_j) = \max_{1 \leq l \leq d} |x_i^l - x_j^l|$



# Agrupamento

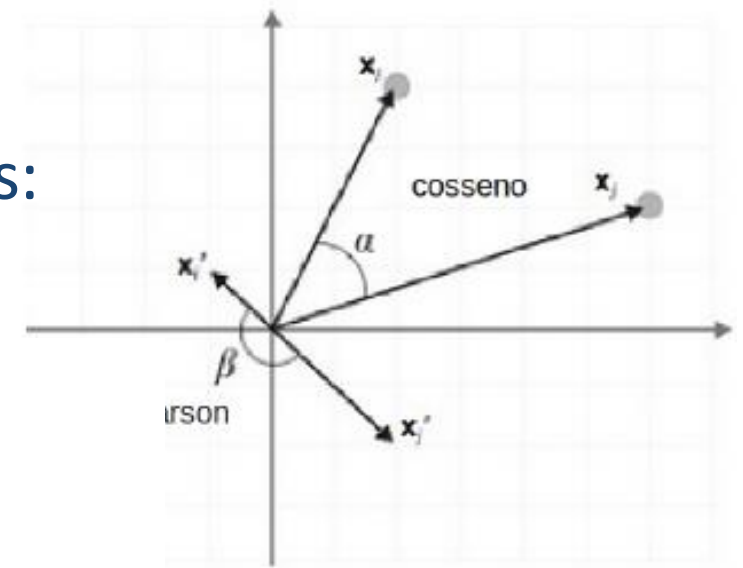
## Etapas: ...

- **Proximidade:** Medidas para Atributos Quantitativos:

- Medidas de Similaridade:

- Separação angular (cosseno):  $\cos(x_i, x_j) = \frac{\sum_{l=1}^d x_i^l x_j^l}{\sqrt{\sum_{l=1}^d x_i^l{}^2 \sum_{l=1}^d x_j^l{}^2}}$

- Pearson:  $p(x_i, x_j) = \frac{\sum_{l=1}^d (x_i^l - \bar{x}_i)(x_j^l - \bar{x}_j)}{\sqrt{\sum_{l=1}^d (x_i^l - \bar{x}_i)^2 \sum_{l=1}^d (x_j^l - \bar{x}_j)^2}}$  (quando magnitude não é importante, mas sim o grau de variação. Ex: Bioinformática)



# Agrupamento

## Etapas: ...

- **Proximidade:** Medidas para Atributos Qualitativos:
  - São obtidas a partir da soma das contribuições individuais de todos os atributos.
  - Para atributos nominais, a distância mais usada é a de Hamming.

$$d(x_i, x_j) = \sum_{l=1}^d a(x_i^l, x_j^l) , \text{ onde } a(x_i^l, x_j^l) = \begin{cases} 1 & \text{se } x_i^l \neq x_j^l \\ 0 & \text{c.c} \end{cases}$$

# Agrupamento

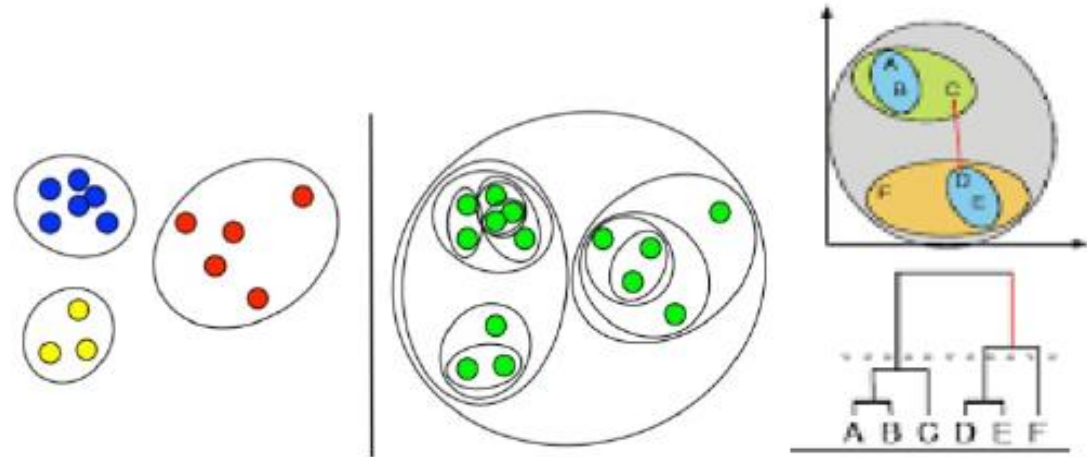
## Etapas: ...

- **agrupamento:** nessa etapa um ou mais algoritmos de agrupamento são usados para gerar os grupos.
- **validação:** etapa que verifica se os grupos gerados são significativos. Ajuda a determinar o número adequado de
- **grupos:** quando esse número não é conhecido.
- **interpretação:** processo de examinar o grupo em relação aos outros com o objetivo de rotulá-lo, indicando a natureza do grupo.



# Agrupamento

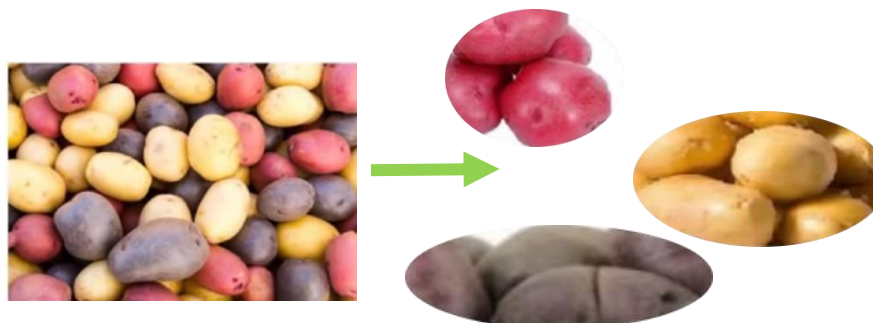
## Tipos:



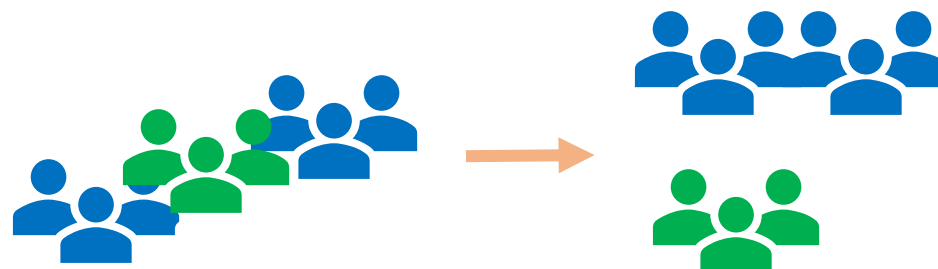
- **Agrupamento Particional:** Divisão dos objetos de dados em subconjuntos (grupos) sem sobreposição tal que cada objeto de dados está em exatamente um único grupo.
- **Agrupamento Hierárquico:** Um conjunto de grupos aninhados na forma de uma árvore hierárquica.

# Exemplos de Agrupamento Particional

**Agrupamento produtos**  
de acordo com as suas  
características.

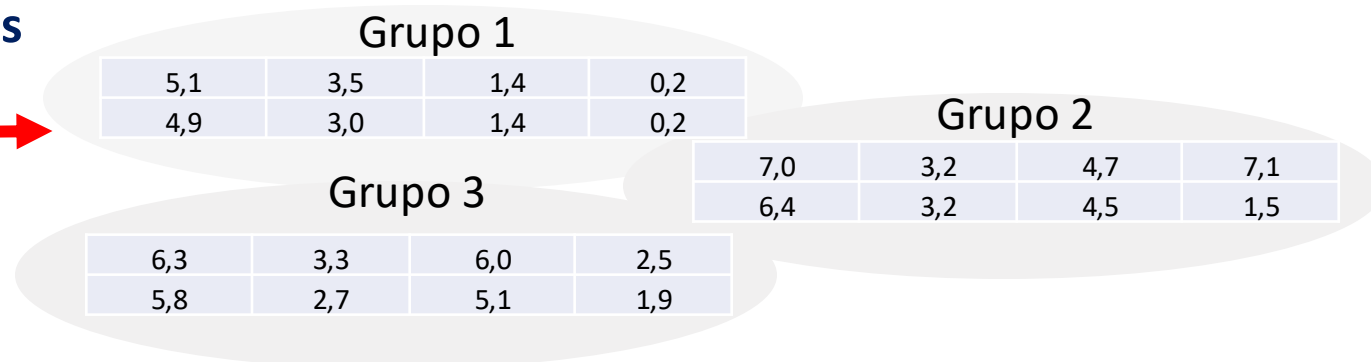


**Agrupamento de clientes**  
Identificação de perfil para  
recomendação de  
produtos

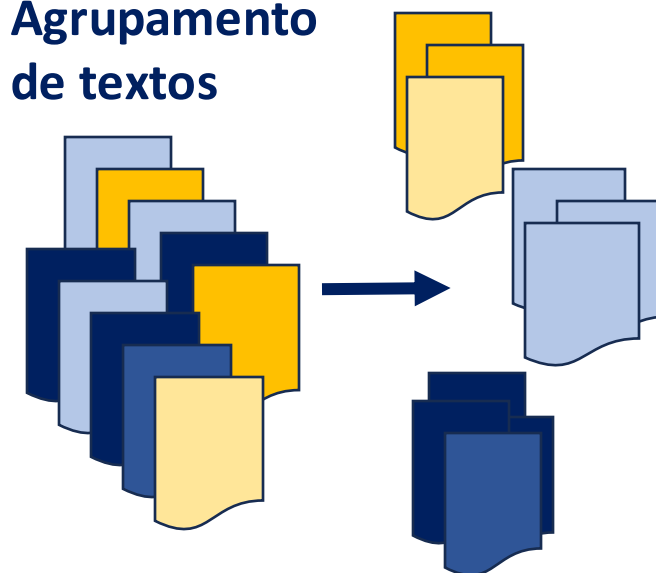


**Agrupamento de dados similares**

sepal length	sepal width	petal length	petal width
5,1	3,5	1,4	0,2
4,9	3,0	1,4	0,2
7,0	3,2	4,7	7,1
6,4	3,2	4,5	1,5
6,3	3,3	6,0	2,5
5,8	2,7	5,1	1,9



**Agrupamento  
de textos**



# Agrupamento Particional



Dados não rotulados

sepal length	sepal width	petal length	petal width
5,1	3,5	1,4	0,2
4,9	3,0	1,4	0,2
7,0	3,2	4,7	7,1
6,4	3,2	4,5	1,5
6,3	3,3	6,0	2,5
5,8	2,7	5,1	1,9

Algoritmo de  
Machine Learning

Resultados

Algoritmo  
k-Means

Tarefa: agrupamento  
(clustering)

Grupo 1

5,1	3,5	1,4	0,2
4,9	3,0	1,4	0,2

Grupo 2

7,0	3,2	4,7	7,1
6,4	3,2	4,5	1,5

Grupo 3

6,3	3,3	6,0	2,5
5,8	2,7	5,1	1,9

# k-means: Algoritmo de agrupamento particional

- **k-means:** Algoritmo de agrupamento particional.
- Características:
  - Cada grupo está associado a um centroide (objeto central).
  - Cada objeto é atribuído ao grupo com o centroide mais próximo.
  - Número de grupos ( $k$ ) deve ser especificado
  - Centroides iniciais são geralmente aleatórios.
  - Agrupamento varia conforme a inicialização.



# k-means: Algoritmo de agrupamento particional

- **k-means:** Algoritmo de agrupamento particional.
- Características:
  - Centroides são (tipicamente) a média de todos os objetos do grupo.
  - A medida de distância geralmente empregada é a distância Euclidiana.
  - k-means geralmente converge com poucas iterações.
  - Complexidade é  $O(n.k.i.d)$ , onde
    - $n$  = número de objetos,
    - $k$  = número de grupos,
    - $i$  = número de iterações e
    - $d$  = número de atributos

# k-means: Algoritmo de agrupamento particional

- **k-means:** Algoritmo de agrupamento particional.
  1. Selecione k objetos como centroides;
  2. Repita
    1. Calcule a distância de cada objeto em relação ao centroide de cada cluster k
    2. O objeto será atribuído ao cluster k cujo centroide está mais próximo
    3. Depois de processar todos os dados, recalcule os centroides, considerando os dados em cada cluster k.
  3. Até que os centroides não mudem mais

# k-means: Algoritmo de agrupamento particional

- k-means: Algoritmo de agrupamento particional. Exemplo:

Padrão	Peso(Atributo1)	Altura(Atributo2)
1	2	8
2	8	2
3	6	8
4	2	7
5	8	4
6	2	6

Manhattan:  $d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$

# k-means: Algoritmo de agrupamento particional

- k-means: Algoritmo de agrupamento particional. Exemplo:

Padrão	Peso(Atributo1)	Altura(Atributo2)
1	2	8
2	8	2
3	6	8
4	2	7
5	8	4
6	2	6

Manhattan:  $d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$

## Cluster 1

Centroide: (2) [8,2]

Tuplas:

## Cluster 2

Centroide: (6) [2,6]

Tuplas:



# k-means: Algoritmo de agrupamento particional

- k-means: Algoritmo de agrupamento particional. Exemplo:

Padrão	Peso(Atributo1)	Altura(Atributo2)
1	2	8
2	8	2
3	6	8
4	2	7
5	8	4
6	2	6

Manhattan: 
$$d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$$

## Cluster 1

Centroide: (2) [8,2]

Tuplas:

## Cluster 2

Centroide: (6) [2,6]

Tuplas:

**Tupla: (1) [2,8]**

Dist(C1,T1)=  $|8-2| + |2-8|=12$

Dist(C2,T1) =  $|2-2| + |6-8|=2$

# k-means: Algoritmo de agrupamento particional

- k-means: Algoritmo de agrupamento particional. Exemplo:

Padrão	Peso(Atributo1)	Altura(Atributo2)
1	2	8
2	8	2
3	6	8
4	2	7
5	8	4
6	2	6

Manhattan:  $d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$

## Cluster 1

Centroide: (2) [8,2]

Tuplas:

## Cluster 2

Centroide: (6) [2,6]

Tuplas: [2,8]

**Tupla: (1) [2,8]**

**Dist(C1,T1)= |8-2|+|2-8|=12**

**Dist(C2,T1) = |2-2| + |6-8|=2**

# k-means: Algoritmo de agrupamento particional

- k-means: Algoritmo de agrupamento particional. Exemplo:

Padrão	Peso(Atributo1)	Altura(Atributo2)
1	2	8
2	8	2
3	6	8
4	2	7
5	8	4
6	2	6

Manhattan:  $d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$

## Cluster 1

Centroide: (2) [8,2]

Tuplas: [8,2]

## Cluster 2

Centroide: (6) [2,6]

Tuplas: [2,8]

**Tupla: (2) [8,2]**

Dist(C1,T2)=  $|8-8| + |2-2| = 0$

Dist(C2,T2) =  $|2-8| + |6-2| = 10$

# k-means: Algoritmo de agrupamento particional

- k-means: Algoritmo de agrupamento particional. Exemplo:

Padrão	Peso(Atributo1)	Altura(Atributo2)
1	2	8
2	8	2
3	6	8
4	2	7
5	8	4
6	2	6

Manhattan:  $d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$

## Cluster 1

Centroide: (2) [8,2]

Tuplas: [8,2]

## Cluster 2

Centroide: (6) [2,6]

Tuplas: [2,8] [6,8]

**Tupla: (3) [6,8]**

**Dist(C1,T3)= |8-6|+|2-8|=8**

**Dist(C2,T3) = |2-6| + |6-8|=6**



# k-means: Algoritmo de agrupamento particional

- k-means: Algoritmo de agrupamento particional. Exemplo:

Padrão	Peso(Atributo1)	Altura(Atributo2)
1	2	8
2	8	2
3	6	8
4	2	7
5	8	4
6	2	6

Manhattan:  $d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$

## Cluster 1

Centroide: (2) [8,2]

Tuplas: [8,2]

## Cluster 2

Centroide: (6) [2,6]

Tuplas: [2,8] [6,8] [2,7]

**Tupla: (4) [2,7]**

**Dist(C1,T4)= |8-2|+|2-7|=11**

**Dist(C2,T4) = |2-2| + |6-7|=1**

# k-means: Algoritmo de agrupamento particional

- k-means: Algoritmo de agrupamento particional. Exemplo:

Padrão	Peso(Atributo1)	Altura(Atributo2)
1	2	8
2	8	2
3	6	8
4	2	7
5	8	4
6	2	6

Manhattan:  $d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$

## Cluster 1

Centroide: (2) [8,2]

Tuplas: [8,2], [8,4]

## Cluster 2

Centroide: (6) [2,6]

Tuplas: [2,8] [6,8] [2,7]

**Tupla: (5) [8,4]**

**Dist(C1,T5)= |8-8|+|2-4|=2**

**Dist(C2,T5) = |2-8| + |6-4|=8**

# k-means: Algoritmo de agrupamento particional

- k-means: Algoritmo de agrupamento particional. Exemplo:

Padrão	Peso(Atributo1)	Altura(Atributo2)
1	2	8
2	8	2
3	6	8
4	2	7
5	8	4
6	2	6

Manhattan:  $d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$

## Cluster 1

Centroide: (2) [8,2]

Tuplas: [8,2], [8,4]

## Cluster 2

Centroide: (6) [2,6]

Tuplas: [2,8] [6,8] [2,7] [2,6]

**Tupla: (6) [2,6]**

**Dist(C1,T6)= |8-2| + |2-6|=10**

**Dist(C2,T6) = |2-2| + |6-6|=0**

# k-means: Algoritmo de agrupamento particional

- k-means: Algoritmo de agrupamento particional. Exemplo:

Padrão	Peso(Atributo1)	Altura(Atributo2)
1	2	8
2	8	2
3	6	8
4	2	7
5	8	4
6	2	6

Manhattan: 
$$d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$$

## Cluster 1

Centroide: ~~(2) [8,2]~~ [8,3]

Tuplas: [8,2], [8,4]

## Cluster 2

Centroide: (6) [2,6]

Tuplas: [2,8] [6,8] [2,7] [2,6]

Recalculando os centroides:

Cluster1:

$[(8+8)/2, (2+4)/2] = [8,3]$

# k-means: Algoritmo de agrupamento particional

- k-means: Algoritmo de agrupamento particional. Exemplo:

Padrão	Peso(Atributo1)	Altura(Atributo2)
1	2	8
2	8	2
3	6	8
4	2	7
5	8	4
6	2	6

Manhattan:  $d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$

## Cluster 1

Centroide: ~~(2) [8,2]~~ [8,3]

Tuplas: [8,2], [8,4]

## Cluster 2

Centroide: ~~(6) [2,6]~~ [3, 7.25]

Tuplas: [2,8] [6,8] [2,7] [2,6]

Recalculando os centroides:

Cluster2:

$[(2+6+2+2)/4, (8+8+7+6)/4] = [3, 7.25]$

# k-means: Algoritmo de agrupamento particional

- **Dinâmica:** Execute 1 iteração e determine o centróide, considerando que as tuplas 2 e 4 foram escolhidas como centróides:

	Atributo 1	Atributo 2	Atributo 3	Atributo 4
1	5	4	3	1
2	1	0	1	2
3	2	1	0	2
4	6	3	6	1
5	3	4	2	3
6	3	3	1	3

# k-means: Algoritmo de agrupamento particional

- **Dinâmica:** Execute 1 iteração e determine o centróide, considerando que as tuplas 2 e 4 foram escolhidas como centróides:

Manhattan:  $d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$

	Atributo 1	Atributo 2	Atributo 3	Atributo 4
1	5	4	3	1
2	1	0	1	2
3	2	1	0	2
4	6	3	6	1
5	3	4	2	3
6	3	3	1	3

Centroide 1 (C1) = [ 1, 0, 1, 2 ] (Tupla 2)

Centroide 2 (C2) = [6, 3, 6, 1 ] (Tupla 4)



# k-means: Algoritmo de agrupamento particional

- **Dinâmica:** Execute 1 iteração e determine o centróide, considerando que as tuplas 2 e 4 foram escolhidas como centróides:

$$\text{Manhattan: } d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$$

	Atributo 1	Atributo 2	Atributo 3	Atributo 4
1	5	4	3	1
2	1	0	1	2
3	2	1	0	2
4	6	3	6	1
5	3	4	2	3
6	3	3	1	3

Centroide 1 (C1) = [ 1, 0, 1, 2 ] (Tupla 2)

- $\text{Dist}(C1, \text{Tupla1}) = |1-5| + |0-4| + |1-3| + |2-1| = 11$
- $\text{Dist}(C1, \text{Tupla2}) = 0$
- $\text{Dist}(C1, \text{Tupla3}) = |1-2| + |0-1| + |1-0| + |2-2| = 3$
- $\text{Dist}(C1, \text{Tupla4}) = |1-6| + |0-3| + |1-6| + |2-1| = 14$
- $\text{Dist}(C1, \text{Tupla5}) = |1-3| + |0-4| + |1-2| + |2-3| = 8$
- $\text{Dist}(C1, \text{Tupla6}) = |1-3| + |0-3| + |1-1| + |2-3| = 6$


Centroide 2 (C2) = [ 6, 3, 6, 1 ] (Tupla 4)

- $\text{Dist}(C2, \text{Tupla1}) = |6-5| + |3-4| + |6-3| + |1-1| = 5$
- $\text{Dist}(C2, \text{Tupla2}) = |6-1| + |3-0| + |6-1| + |1-2| = 14$
- $\text{Dist}(C2, \text{Tupla3}) = |6-2| + |3-1| + |6-0| + |1-2| = 13$
- $\text{Dist}(C2, \text{Tupla4}) = 0$
- $\text{Dist}(C2, \text{Tupla5}) = |6-3| + |3-4| + |6-2| + |1-3| = 10$
- $\text{Dist}(C2, \text{Tupla6}) = |6-3| + |3-3| + |6-1| + |1-3| = 10$

# k-means: Algoritmo de agrupamento particional

- **Dinâmica:** Execute 1 iteração e determine o centróide, considerando que as tuplas 2 e 4 foram escolhidas como centróides:

Manhattan:  $d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$



	Atributo 1	Atributo 2	Atributo 3	Atributo 4
1	5	4	3	1
2	1	0	1	2
3	2	1	0	2
4	6	3	6	1
5	3	4	2	3
6	3	3	1	3

**Centroide 1 (C1) = [ 1, 0, 1, 2 ] (Tupla 2)**

- $\text{Dist}(C1, \text{Tupla1}) = |1-5| + |0-4| + |1-3| + |2-1| = 11$
- $\text{Dist}(C1, \text{Tupla2}) = 0$
- $\text{Dist}(C1, \text{Tupla3}) = |1-2| + |0-1| + |1-0| + |2-2| = 3$
- $\text{Dist}(C1, \text{Tupla4}) = |1-6| + |0-3| + |1-6| + |2-1| = 14$
- $\text{Dist}(C1, \text{Tupla5}) = |1-3| + |0-4| + |1-2| + |2-3| = 8$
- $\text{Dist}(C1, \text{Tupla6}) = |1-3| + |0-3| + |1-1| + |2-3| = 6$


**Centroide 2 (C2) = [ 6, 3, 6, 1 ] (Tupla 4)**

- **$\text{Dist}(C2, \text{Tupla1}) = |6-5| + |3-4| + |6-3| + |1-1| = 5$**
- $\text{Dist}(C2, \text{Tupla2}) = |6-1| + |3-0| + |6-1| + |1-2| = 14$
- $\text{Dist}(C2, \text{Tupla3}) = |6-2| + |3-1| + |6-0| + |1-2| = 13$
- $\text{Dist}(C2, \text{Tupla4}) = 0$
- $\text{Dist}(C2, \text{Tupla5}) = |6-3| + |3-4| + |6-2| + |1-3| = 10$
- $\text{Dist}(C2, \text{Tupla6}) = |6-3| + |3-3| + |6-1| + |1-3| = 10$

# k-means: Algoritmo de agrupamento particional

- **Dinâmica:** Execute 1 iteração e determine o centróide, considerando que as tuplas 2 e 4 foram escolhidas como centróides:

Manhattan:  $d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$



	Atributo 1	Atributo 2	Atributo 3	Atributo 4
1	5	4	3	1
2	1	0	1	2
3	2	1	0	2
4	6	3	6	1
5	3	4	2	3
6	3	3	1	3

Centroide 1 (C1) = [ 1, 0, 1, 2 ] (Tupla 2)

- **Dist(C1, Tupla2) = 0**
- Dist(C1, Tupla3) =  $|1-2| + |0-1| + |1-0| + |2-2| = 3$
- Dist(C1, Tupla4) =  $|1-6| + |0-3| + |1-6| + |2-1| = 14$
- Dist(C1, Tupla5) =  $|1-3| + |0-4| + |1-2| + |2-3| = 8$
- Dist(C1, Tupla 6) =  $|1-3| + |0-3| + |1-1| + |2-3| = 6$

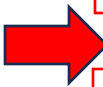
Centroide 2 (C2) = [ 6, 3, 6, 1 ] (Tupla 4)  
[ 5, 4, 3, 1 ]

- Dist(C2, Tupla2) =  $|6-1| + |3-0| + |6-1| + |1-2| = 14$
- Dist(C2, Tupla3) =  $|6-2| + |3-1| + |6-0| + |1-2| = 13$
- Dist(C2, Tupla4) = 0
- Dist(C2, Tupla5) =  $|6-3| + |3-4| + |6-2| + |1-3| = 10$
- Dist(C2, Tupla 6) =  $|6-3| + |3-3| + |6-1| + |1-3| = 10$

# k-means: Algoritmo de agrupamento particional

- **Dinâmica:** Execute 1 iteração e determine o centróide, considerando que as tuplas 2 e 4 foram escolhidas como centróides:

Manhattan:  $d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$



	Atributo 1	Atributo 2	Atributo 3	Atributo 4
1	5	4	3	1
2	1	0	1	2
3	2	1	0	2
4	6	3	6	1
5	3	4	2	3
6	3	3	1	3

Centroide 1 (C1) = [ 1, 0, 1, 2 ] (Tupla 2)  
[1, 0, 1, 2]

- **Dist(C1, Tupla3)** = |1-2|+|0-1|+|1-0|+|2-2| = 3
- Dist(C1, Tupla4) = |1-6|+|0-3|+|1-6|+|2-1| = 14
- Dist(C1, Tupla5) = |1-3|+|0-4|+|1-2|+|2-3| = 8
- Dist(C1, Tupla 6) = |1-3|+|0-3|+|1-1|+|2-3| = 6

Centroide 2 (C2) = [6, 3, 6, 1 ] (Tupla 4)  
[ 5, 4, 3, 1 ]

- Dist(C2, Tupla3) = |6-2|+|3-1|+|6-0|+|1-2| = 13
- Dist(C2, Tupla4) = 0
- Dist(C2, Tupla5) = |6-3|+|3-4|+|6-2|+|1-3| = 10
- Dist(C2, Tupla 6) = |6-3|+|3-3|+|6-1|+|1-3| = 10

# k-means: Algoritmo de agrupamento particional

- **Dinâmica:** Execute 1 iteração e determine o centróide, considerando que as tuplas 2 e 4 foram escolhidas como centróides:

Manhattan:  $d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$

	Atributo 1	Atributo 2	Atributo 3	Atributo 4
1	5	4	3	1
2	1	0	1	2
3	2	1	0	2
4	6	3	6	1
5	3	4	2	3
6	3	3	1	3

Centroide 1 (C1) = [ 1, 0, 1, 2 ] (Tupla 2)

[1, 0, 1, 2]

[2, 1, 0, 2]

- $\text{Dist}(C1, \text{Tupla4}) = |1-6| + |0-3| + |1-6| + |2-1| = 14$
- $\text{Dist}(C1, \text{Tupla5}) = |1-3| + |0-4| + |1-2| + |2-3| = 8$
- $\text{Dist}(C1, \text{Tupla 6}) = |1-3| + |0-3| + |1-1| + |2-3| = 6$

Centroide 2 (C2) = [6, 3, 6, 1] (Tupla 4)

[ 5, 4, 3, 1 ]

- **$\text{Dist}(C2, \text{Tupla4}) = 0$**
- $\text{Dist}(C2, \text{Tupla5}) = |6-3| + |3-4| + |6-2| + |1-3| = 10$
- $\text{Dist}(C2, \text{Tupla 6}) = |6-3| + |3-3| + |6-1| + |1-3| = 10$

# k-means: Algoritmo de agrupamento particional

- **Dinâmica:** Execute 1 iteração e determine o centróide, considerando que as tuplas 2 e 4 foram escolhidas como centróides:

Manhattan:  $d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$

	Atributo 1	Atributo 2	Atributo 3	Atributo 4
1	5	4	3	1
2	1	0	1	2
3	2	1	0	2
4	6	3	6	1
5	3	4	2	3
6	3	3	1	3

Centroide 1 (C1) = [ 1, 0, 1, 2 ] (Tupla 2)

[1, 0, 1, 2]

[2, 1, 0, 2]

- **Dist(C1, Tupla5)** = |1-3|+|0-4|+|1-2|+|2-3| = 8

- **Dist(C1, Tupla 6)** = |1-3|+|0-3|+|1-1|+|2-3| = 6

Centroide 2 (C2) = [6, 3, 6, 1 ] (Tupla 4)

[ 5, 4, 3, 1]

[ 6, 3, 6, 1]

- **Dist(C2, Tupla5)** = |6-3|+|3-4|+|6-2|+|1-3| = 10

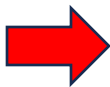
- **Dist(C2, Tupla 6)** = |6-3|+|3-3|+|6-1|+|1-3| = 10

# k-means: Algoritmo de agrupamento particional

- **Dinâmica:** Execute 1 iteração e determine o centróide, considerando que as tuplas 2 e 4 foram escolhidas como centróides:

Manhattan:  $d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$

	Atributo 1	Atributo 2	Atributo 3	Atributo 4
1	5	4	3	1
2	1	0	1	2
3	2	1	0	2
4	6	3	6	1
5	3	4	2	3
6	3	3	1	3



Centroide 1 (C1) = [ 1, 0, 1, 2 ] (Tupla 2)

[1, 0, 1, 2]

[2, 1, 0, 2]

[3, 4, 2, 3]

- **Dist(C1, Tupla 6) = |1-3| + |0-3| + |1-1| + |2-3| = 6**

Centroide 2 (C2) = [6, 3, 6, 1 ] (Tupla 4)

[ 5, 4, 3, 1 ]

[ 6, 3, 6, 1 ]

- **Dist(C2, Tupla 6) = |6-3| + |3-3| + |6-1| + |1-3| = 10**



# k-means: Algoritmo de agrupamento particional

- **Dinâmica:** Execute 1 iteração e determine o centróide, considerando que as tuplas 2 e 4 foram escolhidas como centróides:

Manhattan:  $d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$

	Atributo 1	Atributo 2	Atributo 3	Atributo 4
1	5	4	3	1
2	1	0	1	2
3	2	1	0	2
4	6	3	6	1
5	3	4	2	3
6	3	3	1	3

Centroide 1 (C1) = [ 1, 0, 1, 2 ] (Tupla 2)

[1, 0, 1, 2]

[2, 1, 0, 2]

[3, 4, 2, 3]

[3, 3, 1, 3]

Centroide 2 (C2) = [6, 3, 6, 1 ] (Tupla 4)

[ 5, 4, 3, 1]

[ 6, 3, 6, 1]

# k-means: Algoritmo de agrupamento particional

- **Dinâmica:** Execute 1 iteração e determine o centróide, considerando que as tuplas 2 e 4 foram escolhidas como centróides:

Manhattan:  $d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$

## Recalculando os centróides:

Centroide 1 (C1) = [ 1, 0, 1, 2 ] (Tupla 2)

[1, 0, 1, 2]

[2, 1, 0, 2]

[3, 4, 2, 3]

[3, 3, 1, 3]

[ (1+2+3+3)/4, (0+1+4+3)/4, (1+0+2+1)/4, (3+3+1+3)/4 ] =

[ 9/4, 8/4, 4/4, 10/4 ] = [ 2.25, 2, 1, 2.5 ]

Centroide 2 (C2) = [6, 3, 6, 1 ] (Tupla 4)

[ 5, 4, 3, 1 ]

[ 6, 3, 6, 1 ]

	Atributo 1	Atributo 2	Atributo 3	Atributo 4
1	5	4	3	1
2	1	0	1	2
3	2	1	0	2
4	6	3	6	1
5	3	4	2	3
6	3	3	1	3

# k-means: Algoritmo de agrupamento particional

- **Dinâmica:** Execute 1 iteração e determine o centróide, considerando que as tuplas 2 e 4 foram escolhidas como centróides:

Manhattan:  $d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$

## Recalculando os centróides:

Centroide 1 (C1) = ~~[1, 0, 1, 2]~~ (Tupla 2) **[2.25, 2, 1, 2.5]**

[1, 0, 1, 2]

[2, 1, 0, 2]

[3, 4, 2, 3]

[3, 3, 1, 3]

Centroide 2 (C2) = **[6, 3, 6, 1]** (Tupla 4)

[5, 4, 3, 1]

[6, 3, 6, 1]

$[(5+6)/2, (4+3)/2, (3+6)/2, (1+1)/2] = [11/2, 7/2, 9/2, 2/2]$

**= [5.5, 3.5, 4.5, 1]**

	Atributo 1	Atributo 2	Atributo 3	Atributo 4
1	5	4	3	1
2	1	0	1	2
3	2	1	0	2
4	6	3	6	1
5	3	4	2	3
6	3	3	1	3

# k-means: Algoritmo de agrupamento particional

- **Dinâmica:** Execute 1 iteração e determine o centróide, considerando que as tuplas 2 e 4 foram escolhidas como centróides:

Manhattan:  $d(x_i, x_j) = \sum_{l=1}^d |x_i^l - x_j^l|$

## Recalculando os centróides:

Centroide 1 (C1) = ~~[1, 0, 1, 2]~~ (Tupla 2) **[2.25, 2, 1, 2.5]**

[1, 0, 1, 2]

[2, 1, 0, 2]

[3, 4, 2, 3]

[3, 3, 1, 3]

Centroide 2 (C2) = ~~[6, 3, 6, 1]~~ (Tupla 4) **[5.5, 3.5, 4.5, 1]**

[5, 4, 3, 1]

[6, 3, 6, 1]

	Atributo 1	Atributo 2	Atributo 3	Atributo 4
1	5	4	3	1
2	1	0	1	2
3	2	1	0	2
4	6	3	6	1
5	3	4	2	3
6	3	3	1	3

# K-Means

## `sklearn.cluster.KMeans`

```
class sklearn.cluster.KMeans(n_clusters=8, *, init='k-means++', n_init='warn', max_iter=300, tol=0.0001, verbose=0, random_state=None, copy_x=True, algorithm='lloyd')
```

[\[source\]](#)

- **n\_clusters** : numero de grupos, padrao=8
- **init**{'k-means++', 'random'}: metodo de inicializacao dos centroides: k-means++ (padrao) faz a escolha baseado em distribuicao.
- **n\_init** 'auto' ou int, padrão = 10 : Número de vezes que o algoritmo k-means é executado com sementes para centróides diferentes.
- **max\_iter** int, padrão=300 : Número máximo de iterações do algoritmo k-means para uma única execução.

# K-Means

## `sklearn.cluster.KMeans`

```
class sklearn.cluster.KMeans(n_clusters=8, *, init='k-means++', n_init='warn', max_iter=300, tol=0.0001, verbose=0, random_state=None, copy_x=True, algorithm='lloyd')
```

[\[source\]](#)

- **tolfloat**, padrão = 1e-4 : Tolerância relativa em relação à norma de Frobenius da diferença nos centros do cluster de duas iterações consecutivas para declarar convergência.
- **algorithm**{“lloyd”, “elkan”}, default=”lloyd”. O algoritmo clássico do estilo EM é "lloyd". A variação "elkan" pode ser mais eficiente em alguns conjuntos de dados com clusters bem definidos. No entanto, consome mais memória devido à alocação de uma matriz extra de forma (n\_samples, n\_clusters).

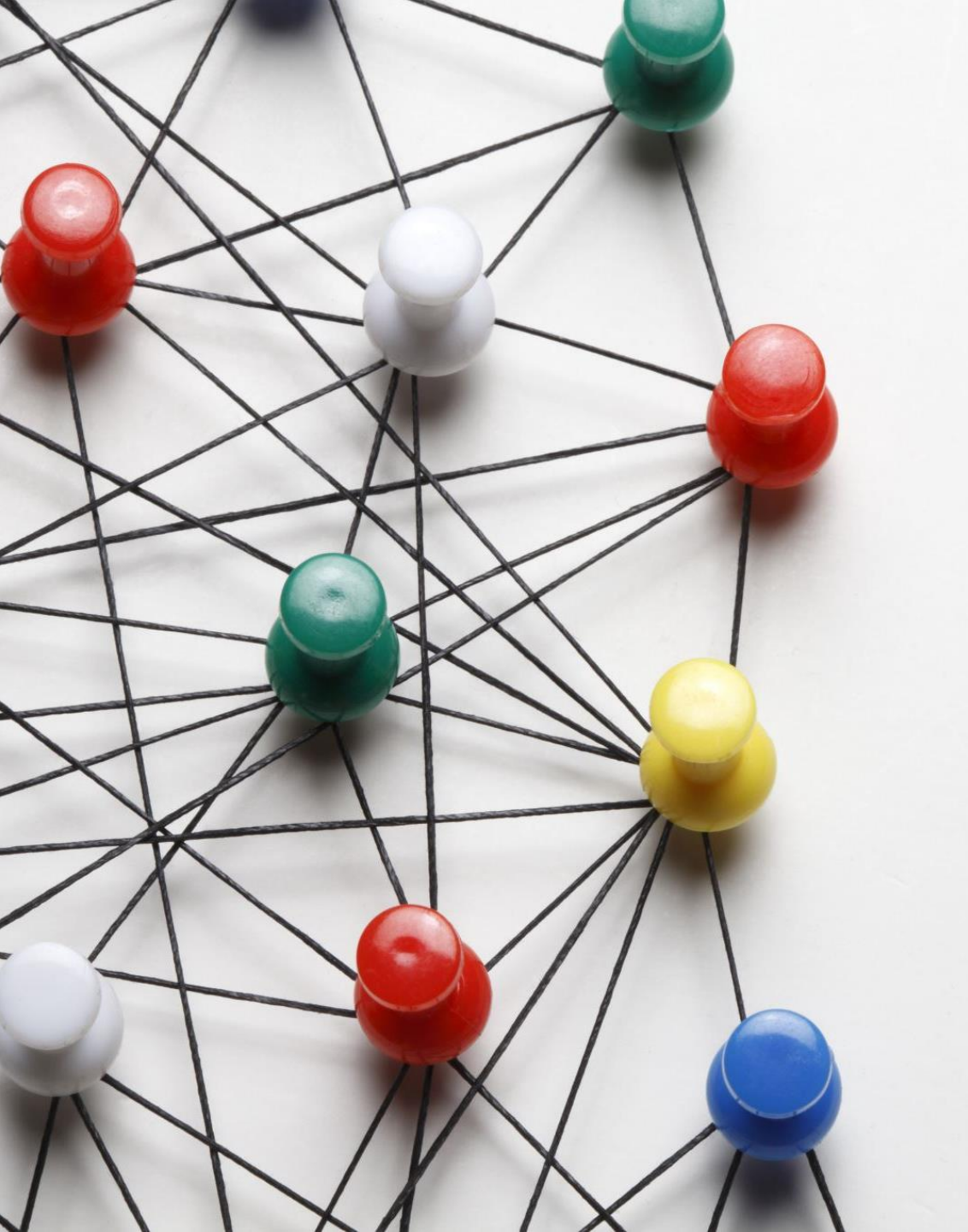
# K-Means

```
# Executando o k-Means para cada valor de k  
kmeanModel = KMeans(n_clusters=k,n_init='auto')  
kmeanModel.fit(X)
```

X contém os dados de entrada







Como determinar a  
quantidade de clusters?  
Qual o melhor  $k$ ?

# K-Means e o método Elbow

- O método elbow é conhecido como método do cotovelo. Basicamente o que o método faz é **testar a variância dos dados em relação os centroides dos clusters**.
- Neste método, para determinar o valor  $k$ ,
  - Iteramos continuamente de  $k = 1$  a  $k = n$  (aqui  $n$  é o hiperparâmetro que escolhemos de acordo com nossos requisitos).
  - Para cada valor de  $k$ , calculamos o valor da soma dos quadrados dentro do cluster (WCSS).
    - WCSS – Mede a coesão. É a soma das distâncias quadradas entre o centróide e cada objeto do cluster.

# K-Means e o método Elbow

No sklearn, usamos a **Distortion** e **Inertia** para medir a relação intracluster e extracluster.

- **Distorção:** É calculada como a média das distâncias quadradas dos centroides até cada objeto do cluster. Normalmente, usa como métrica a distância euclidiana.

$$\text{Distortion} = 1/n * \sum(\text{distance}(\text{point}, \text{centroid})^2)$$

# K-Means e o método Elbow

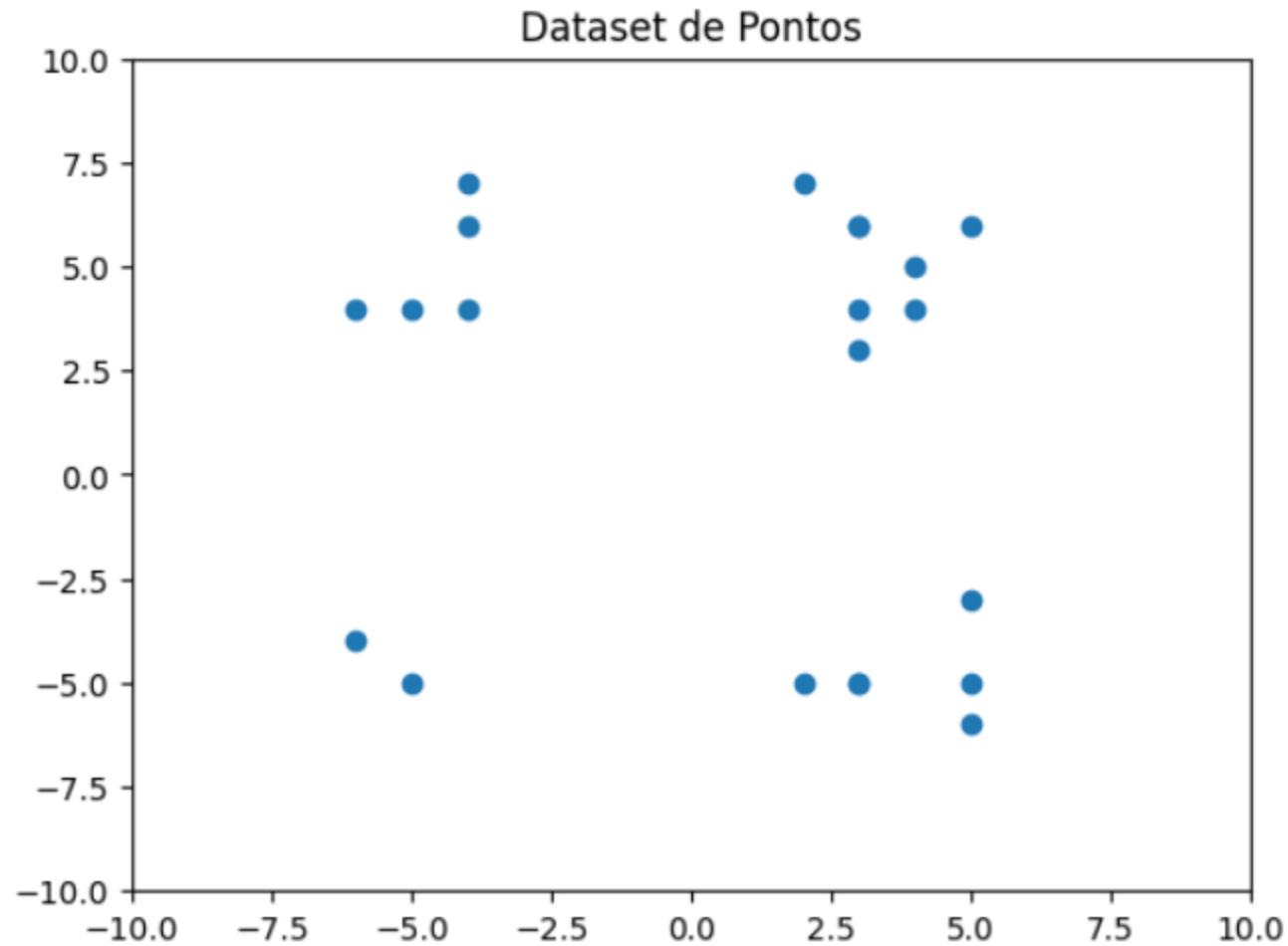
- Inércia: É a soma das distâncias quadradas das amostras ao centro do cluster mais próximo.

$$\text{Inertia} = \sum (\text{distance}(\text{point}, \text{centroid})^2)$$

- Para determinar o número ideal de clusters, temos que selecionar o valor de k no “cotovelo”, ou seja, o ponto após o qual a distorção/inércia começa a diminuir de forma linear.

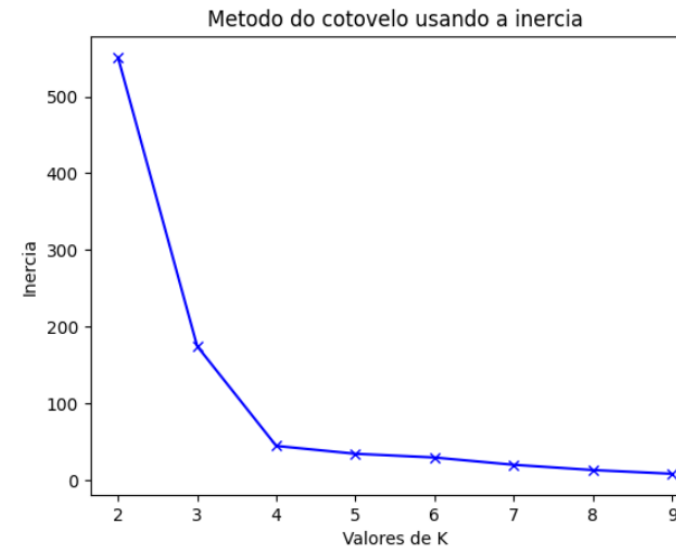
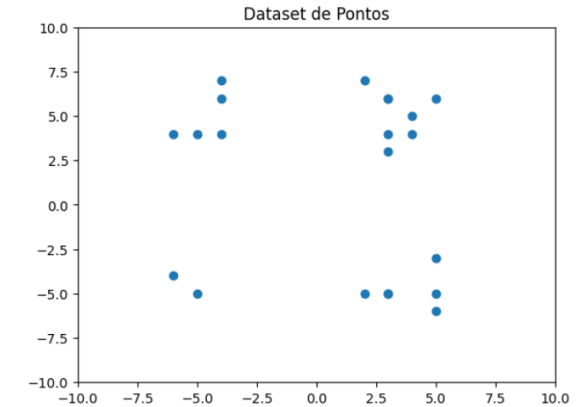
# K-Means e o método Elbow

- Exemplo:



# K-Means e o método Elbow

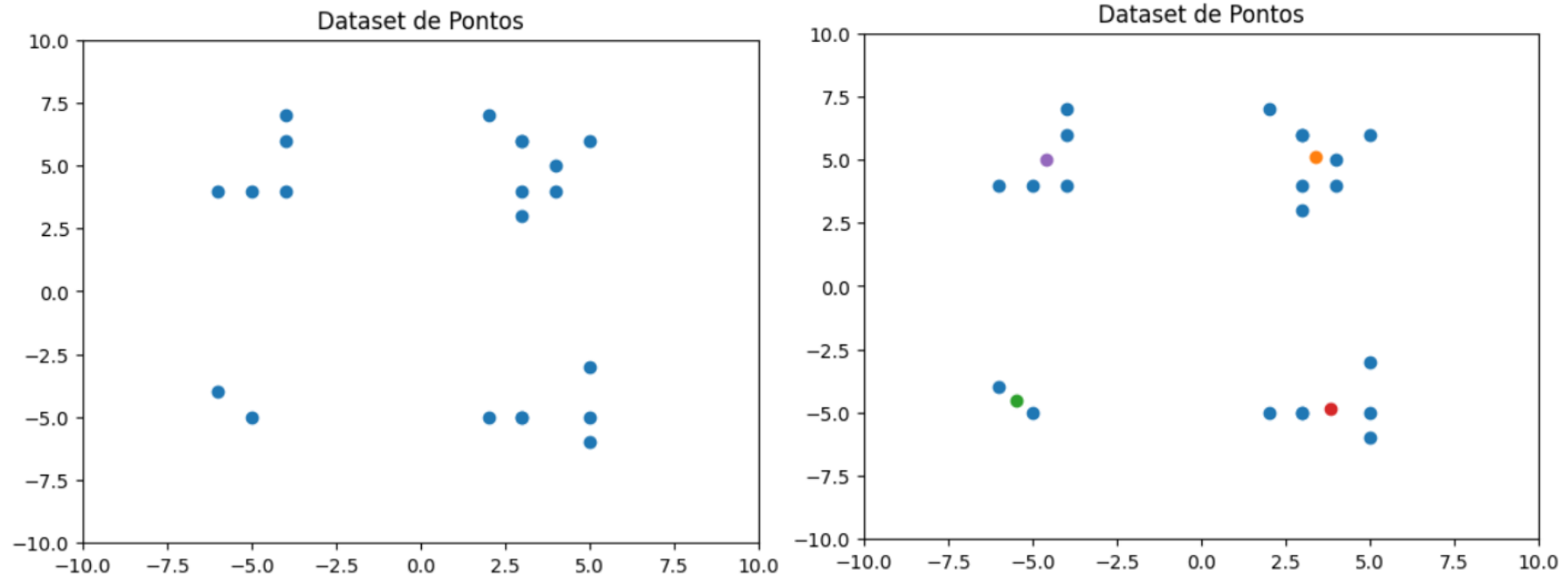
- Exemplo:



Observando os gráficos, concluímos que o número ideal de clusters para os dados é 4.

# K-Means e o método Elbow

- Exemplo:



Observando os gráficos, concluímos que o número ideal de clusters para os dados é 4.



# Dinâmica

- Exemplos de uso do k-Means