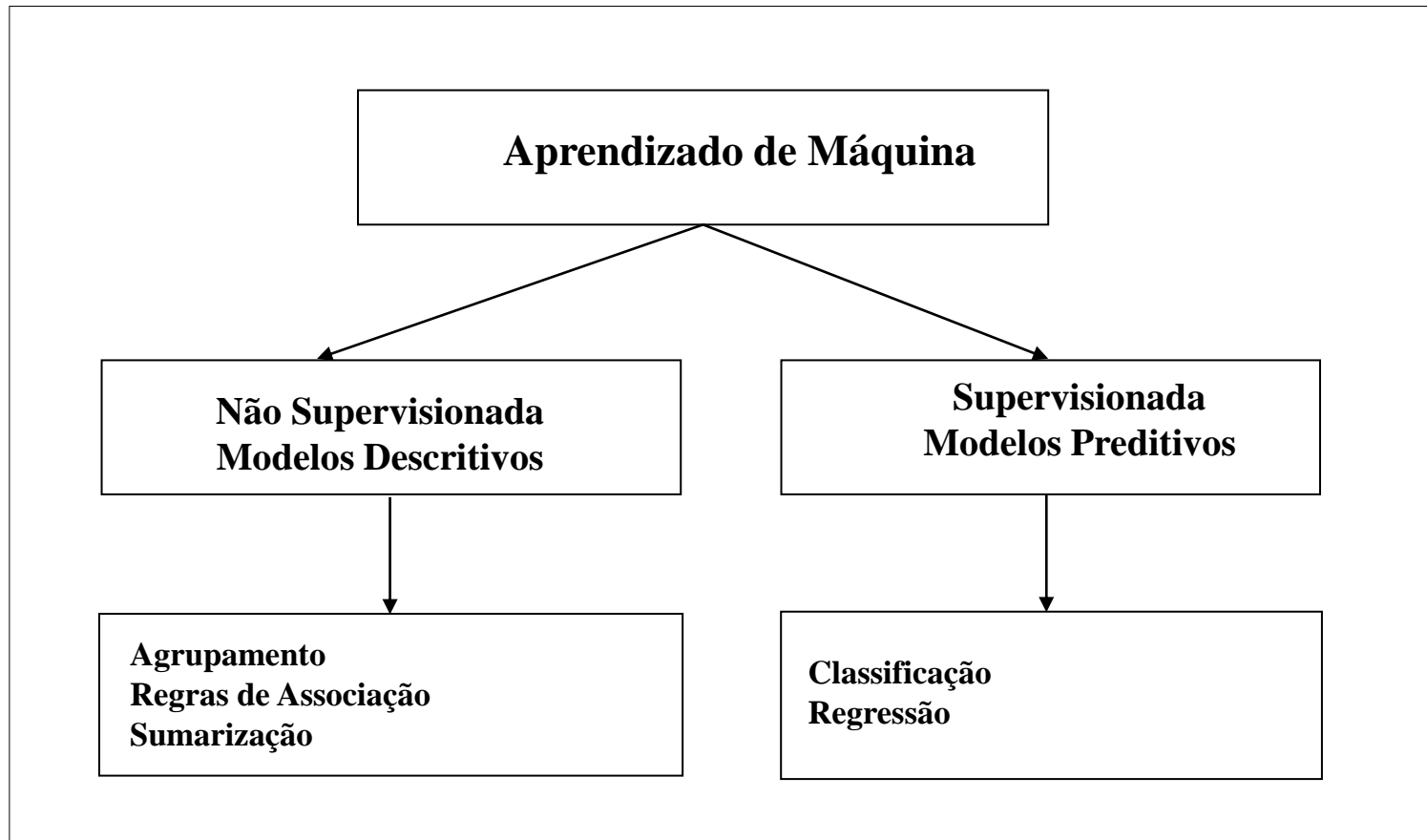


# Aprendizado de Máquina – IMD1101

## Aula 21 – Aprendizado Não Supervisionado 01

# Contextualizando



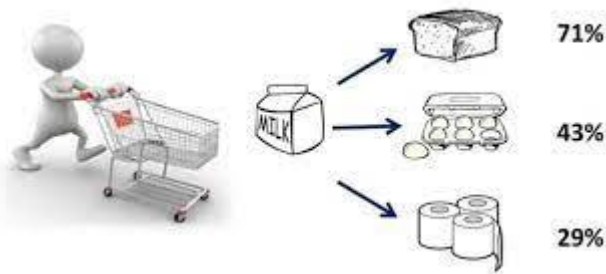
# Modelos Descritivos

- ❑ **Sumarização de Texto:** produção automática de sumários a partir de um ou mais textos.
  - ❖ Extrativa: compõe o resumo a partir de recortes dos textos.
  - ❖ Gerativa: constrói uma síntese dos textos.



# Modelos Descritivos

- ❑ **Associação:** permite identificar relações entre dados a partir da ocorrência desses.
- ❑ **Aplicações:**
  - ❖ Análise da cesta de mercado dos clientes.
  - ❖ Organização dos produtos em uma loja.



<https://diegonogare.net/2020/05/explicando-o-algoritmo-de-regra-de-associacao/>

# Modelos Descritivos

- ❑ **Clustering** ou **agrupamento** é uma técnica de aprendizado **não-supervisionado**, ou seja, quando não há uma **classe associada** a cada exemplo.
- ❑ As instâncias de uma base de dados são colocadas em **clusters** (**grupos**), que normalmente descrevem algum mecanismo existente no processo que as gerou.
- ❑ Dessa forma, algumas instâncias são mais **similares** entre si do que as restantes.

# Agrupamento

- ❑ O principal objetivo do agrupamento é “selecionar” *objetos* (instâncias) de modo que cada objeto seja muito semelhante aos outros no agrupamento (*grupo* ou *cluster*) em relação a algum critério de seleção pré-determinado.
- ❑ Os grupos resultantes de objetos deve exibir *elevada homogeneidade interna* (dentro dos grupos) e *elevada heterogeneidade externa* (entre grupos).

# Agrupamento

- ❑ Dado um conjunto de objetos, colocar os objetos em grupos baseados na similaridade entre eles.
- ❑ Utilizado para encontrar padrões inesperados nos dados.
- ❑ Inerentemente é um problema não **definido claramente**.

- ❑ Como separar os animais abaixo? Qual o critério?



# Descrição do Problema

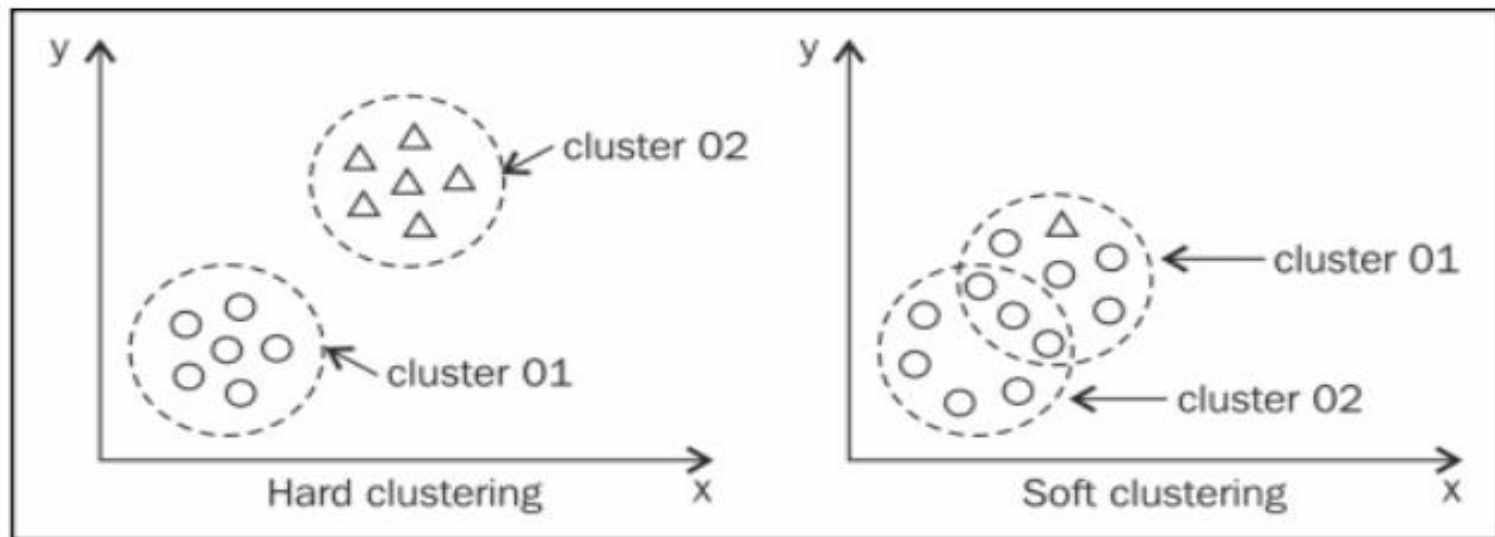
- Dado um conjunto de objetos (instâncias) descritos por múltiplos valores (atributos):
1. Atribuir grupos (clusters) aos objetos particionando-os objetivamente em grupos homogêneos de maneira a:
    - Maximizar a similaridade de objetos dentro de um mesmo cluster.
    - Minimizar a similaridade de objetos entre clusters distintos.
  2. Atribuir uma descrição para cada cluster formado.





# Tipos de Clustering

- Agrupamentos podem ser divididos em:
  - ❖ Hard Clustering;
  - ❖ Soft Clustering.



[https://subscription.packtpub.com/book/big\\_data\\_and\\_business\\_intelligence/9781783554997/2/ch02lv1sec19/types-of-clustering](https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781783554997/2/ch02lv1sec19/types-of-clustering)

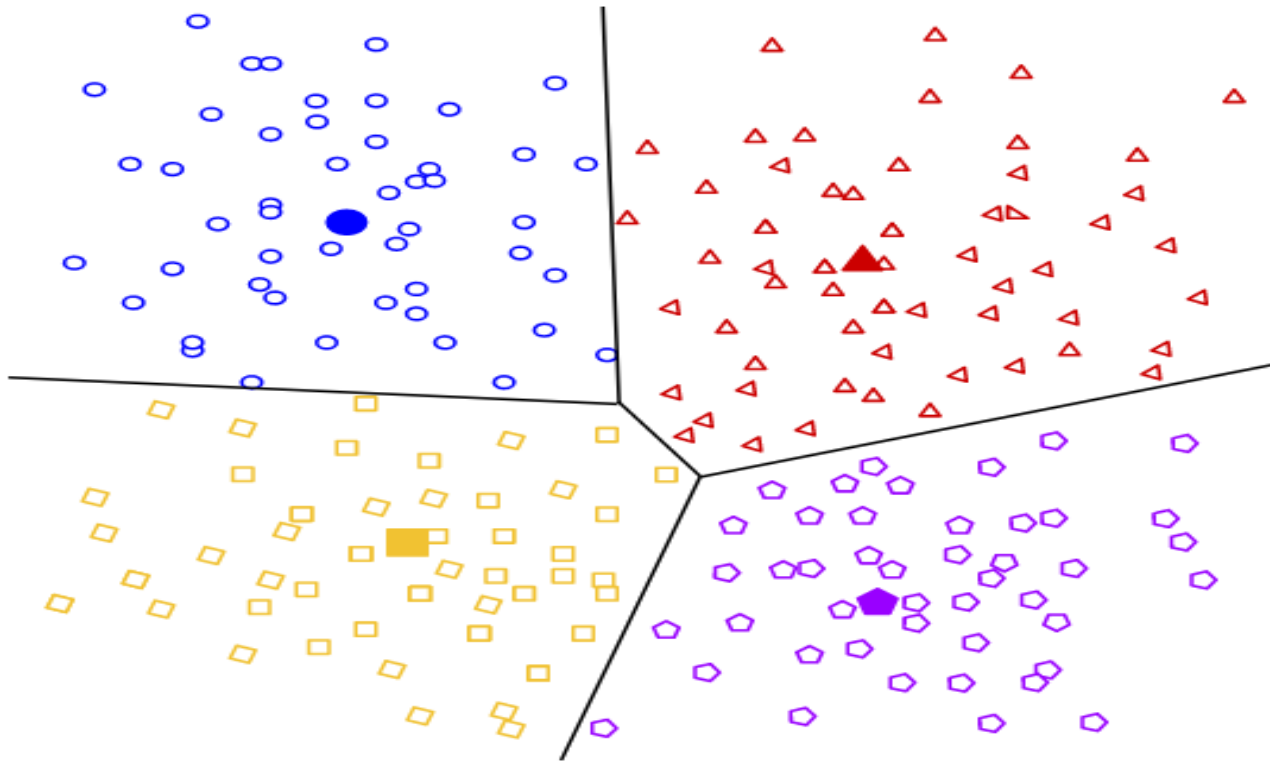
# Algoritmos de Clustering

## □ Tipos:

- ❖ Centroid-based Clustering;
- ❖ Distribution-based Clustering;
- ❖ Hierarchical Clustering.

# Algoritmos de Clustering

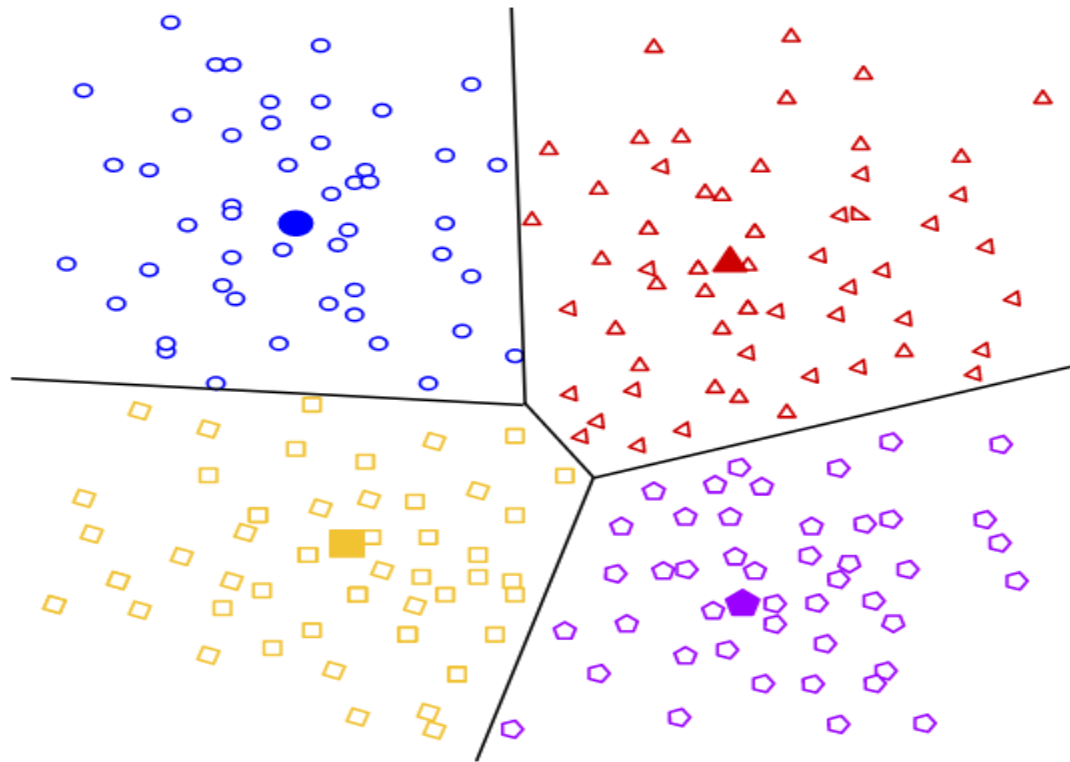
## □ Centroid-based Clustering:



<https://developers.google.com/machine-learning/clustering/clustering-algorithms>

# Algoritmos de Clustering

## □ Centroid-based Clustering:

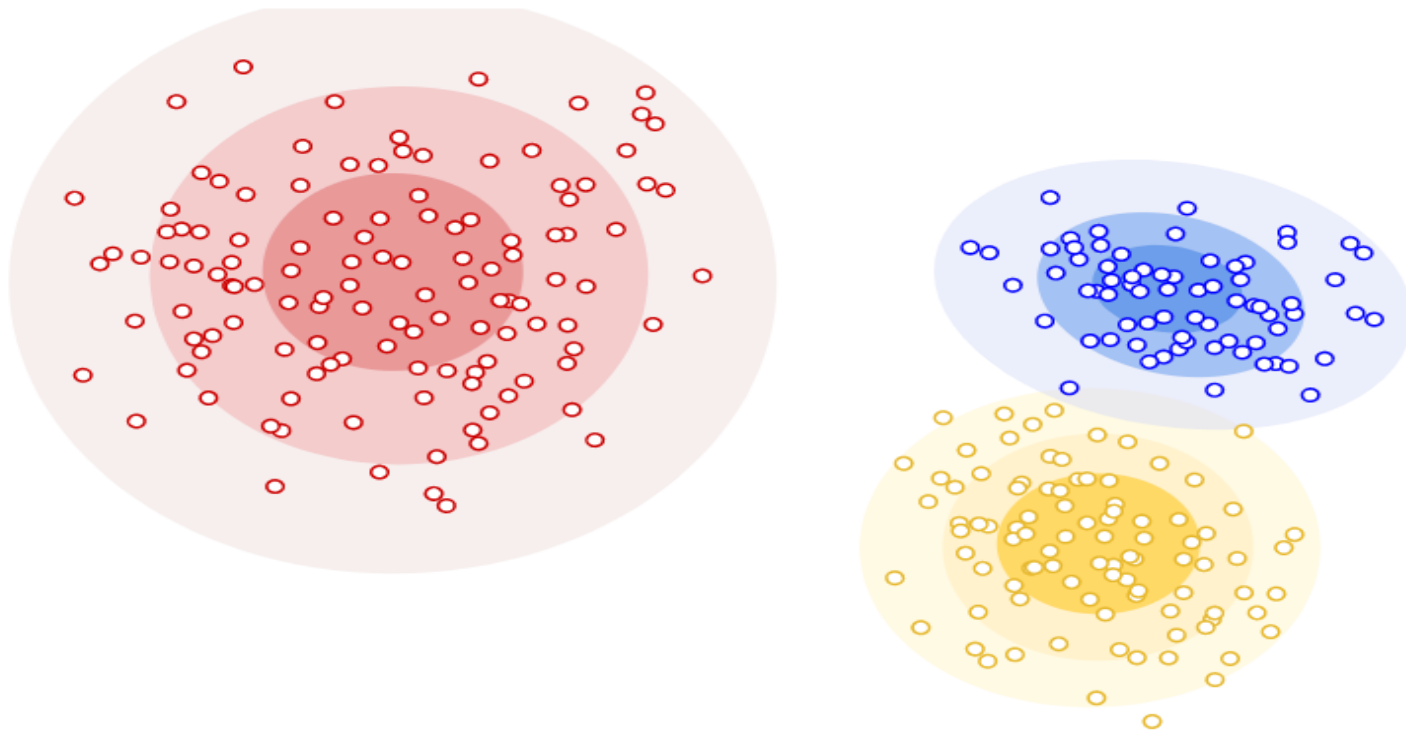


- k-Means
- centróides;
- Medida de distância

<https://developers.google.com/machine-learning/clustering/clustering-algorithms>

# Algoritmos de Clustering

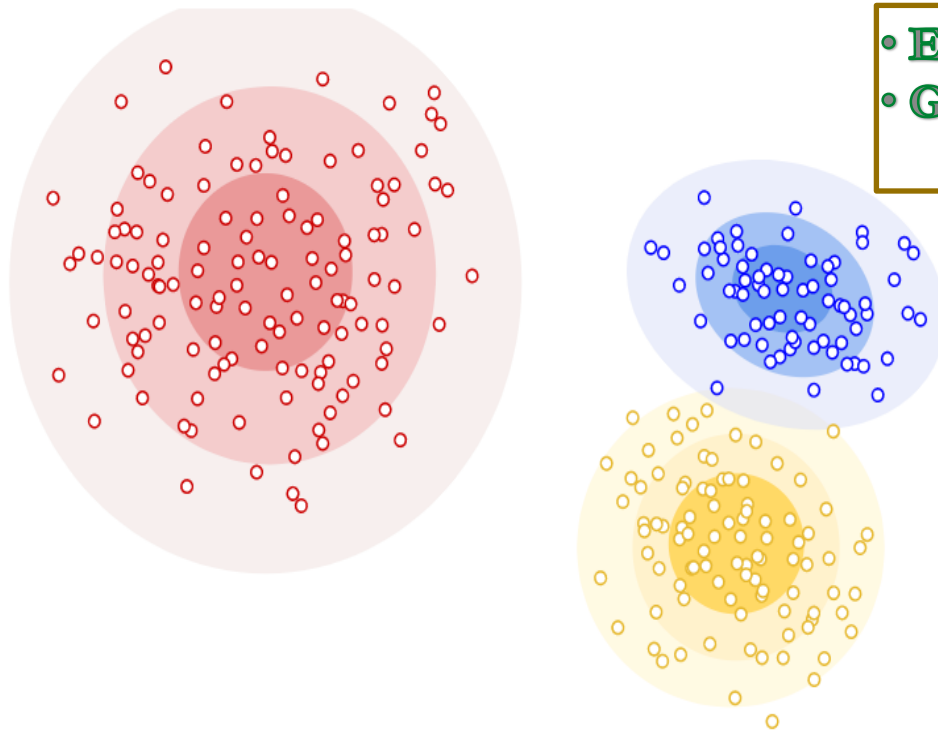
## □ Distribution-based Clustering:



<https://developers.google.com/machine-learning/clustering/clustering-algorithms>

# Algoritmos de Clustering

## □ Distribution-based Clustering:

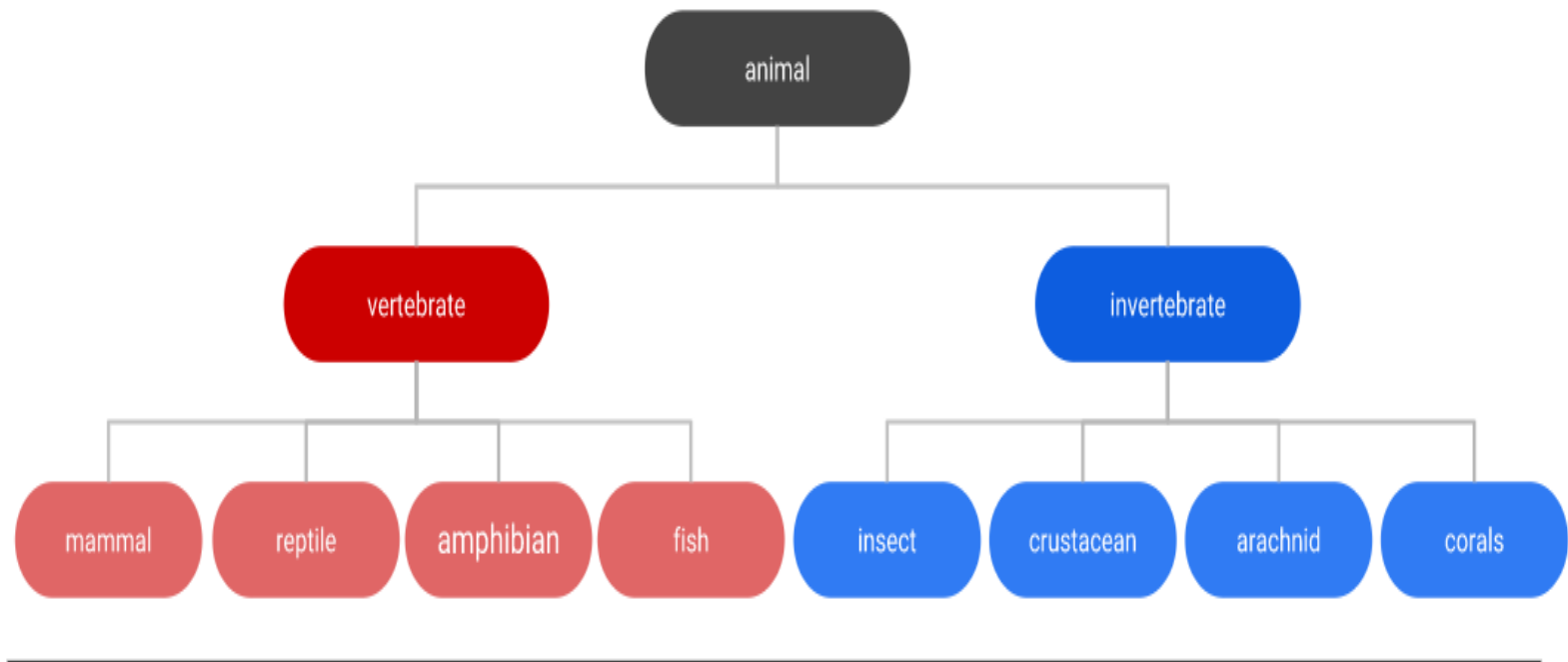


- Expectation–Maximization (EM)
- Gaussian distributions

<https://developers.google.com/machine-learning/clustering/clustering-algorithms>

# Algoritmos de Clustering

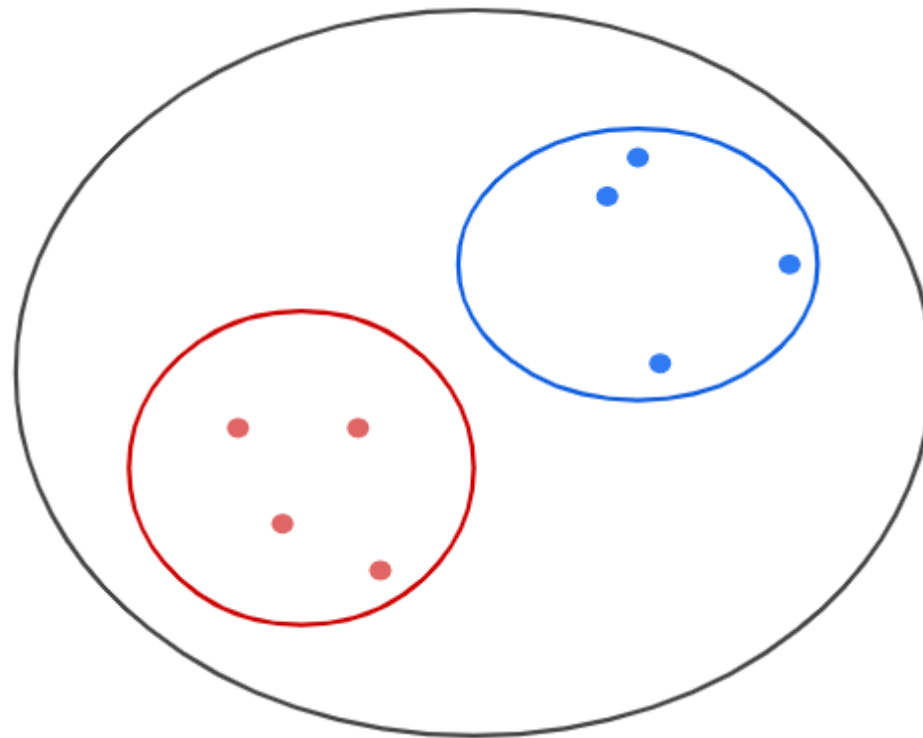
## ❑ Hierarchical Clustering:



<https://developers.google.com/machine-learning/clustering/clustering-algorithms>

# Algoritmos de Clustering

## ❑ Hierarchical Clustering:

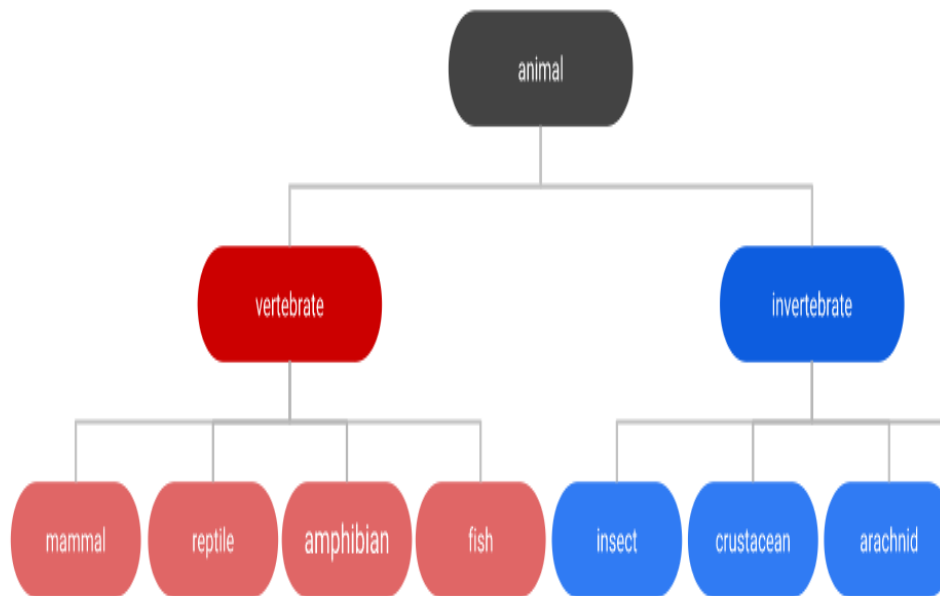


<https://developers.google.com/machine-learning/clustering/clustering-algorithms>

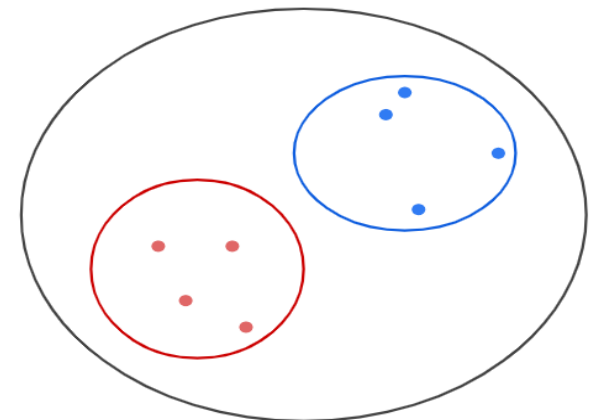


# Algoritmos de Clustering

## ❑ Hierarchical Clustering:



- Hierarchical Clustering
- Medida de distância



<https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68>

# Medidas de Distância

# Calculando a Distância

- ❑ A **distância** é o método mais natural para **dados numéricos**.
- ❑ Valores pequenos indicam **maior similaridade**.
- ❑ Métricas de Distância:
  - ❖ Euclideana
  - ❖ Manhattan
  - ❖ Etc.
- ❑ Não generaliza muito bem para dados não numéricos.
  - ❖ **Qual a distância entre “masculino” e “feminino”?**

# Representação dos Objetos

- ❑ Deve também incluir um método para calcular a similaridade (ou a **distância**) entre os objetos.



# Representação dos Objetos

$$\mathbf{X}_1 = (\mathbf{x}_{1,1}, \mathbf{x}_{1,2}, \mathbf{x}_{1,3}, \dots, \mathbf{x}_{1,m})$$

$$\mathbf{X}_2 = (\mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \mathbf{x}_{2,3}, \dots, \mathbf{x}_{2,m})$$

$$\vdots$$

$$\mathbf{X}_i = (\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \mathbf{x}_{i,3}, \dots, \mathbf{x}_{i,m})$$

$$\vdots$$

$$\mathbf{X}_n = (\mathbf{x}_{n,1}, \mathbf{x}_{n,2}, \mathbf{x}_{n,3}, \dots, \mathbf{x}_{n,m})$$

# Métricas de Distância

- ❑ **Euclidean distance:** é a mais comum entre as distâncias. Ela calcula a raiz quadrada das diferenças entre as coordenadas de um par de objetos:

$$Dist_E = \sqrt{\sum_{r=1}^m (x_{i,r} - x_{j,r})^2}$$

- ❑ **Manhattan distance:** também conhecida por city block distance. Ela calcula as diferenças absolutas entre as coordenadas de um par de objectos:

$$Dist_M = \sum_{r=1}^m |x_{i,r} - x_{j,r}|$$

# Métricas de Distância

- ❑ **Chebyshev distance:** também chamada de distância de valor máximo. Ela calcula a magnitude absoluta das diferenças entre as coordenadas de um par de objetos:

$$Dist_C = \max_{r=1}^m |x_{i,r} - x_{j,r}|$$

- ❑ **Minkowski distance:** é uma distância métrica generalizada. Por exemplo, para  $p=2$ , ela se torna a Distância Euclidiana:

$$Dist_{Mk} = \left( \sum_{r=1}^m |x_{i,r} - x_{j,r}|^{1/p} \right)^p$$

# Métricas de Distância

- O método mais simples para **atributos categóricos** é o seguinte:

$$overlap(x_{i,r}, x_{j,r}) = \begin{cases} 1 & \text{se } x_{i,r} \neq x_{j,r} \\ 0 & \text{se } x_{i,r} = x_{j,r} \end{cases}$$

onde:

$$dist_{Cat} = \sum_{r=1}^m overlap(x_{i,r}, x_{j,r})$$



# Calculando a Distância

- Dada a pequena base de dados abaixo, calcule as distâncias entre as instâncias, usando a distância euclidiana.

| Nº | Idade | Gênero | Estado Civil | Filhos | Escolaridade  | CC  | Renda         | Cartão_Cr. | Imóvel_P |
|----|-------|--------|--------------|--------|---------------|-----|---------------|------------|----------|
| 1  | 45    | Masc   | Divorciado   | 2      | Superior      | Sim | R\$ 5.000,00  | Sim        | Sim      |
| 2  | 37    | Femi   | Solteiro     | 0      | Médio         | Não | R\$ 3.500,00  | Sim        | Não      |
| 3  | 79    | Masc   | Viúvo        | 4      | Fundamental   | Sim | R\$ 10.000,00 | Sim        | Não      |
| 4  | 21    | Femi   | Casado       | 2      | Superior      | Não | R\$ 1.500,00  | Não        | Sim      |
| 5  | 65    | Femi   | Casado       | 1      | Superior      | Sim | R\$ 2.900,00  | Sim        | Sim      |
| 6  | 53    | Masc   | Casado       | 3      | Médio         | Não | R\$ 3.100,00  | Sim        | Não      |
| 7  | 27    | Femi   | Solteiro     | 1      | Superior      | Sim | R\$ 4.200,00  | Sim        | Não      |
| 8  | 33    | Femi   | Casado       | 3      | Pós-graduação | Não | R\$ 7.500,00  | Sim        | Sim      |
| 9  | 41    | Masc   | Divorciado   | 0      | Superior      | Sim | R\$ 5.600,00  | Não        | Não      |
| 10 | 19    | Masc   | Solteiro     | 0      | Médio         | Não | R\$ 800,00    | Não        | Não      |

<https://www.dropbox.com/sh/fhkqy2wybxjl0n5/AAABevgbnnM4HSdPgeUU6tgPa?dl=0>

# Calculando a Distância

□ Base normalizada e binarizada:

| Nº | Idade  | Gen_M | Est_D | Est_S | Est_V | Est_C | Filhos | Esc_S | Esc_M | Esc_F | Esc_P | CC_S | Renda  | Cartão_Cr_S | Imóvel_P_S |
|----|--------|-------|-------|-------|-------|-------|--------|-------|-------|-------|-------|------|--------|-------------|------------|
| 1  | 0,4333 | 1     | 1     | 0     | 0     | 0     | 0,5000 | 1     | 0     | 0     | 0     | 1    | 0,4565 | 1           | 1          |
| 2  | 0,3000 | 0     | 0     | 1     | 0     | 0     | 0,0000 | 0     | 1     | 0     | 0     | 0    | 0,2935 | 1           | 0          |
| 3  | 1,0000 | 1     | 0     | 0     | 1     | 0     | 1,0000 | 0     | 0     | 1     | 0     | 1    | 1,0000 | 1           | 0          |
| 4  | 0,0333 | 0     | 0     | 0     | 0     | 1     | 0,5000 | 1     | 0     | 0     | 0     | 0    | 0,0761 | 0           | 1          |
| 5  | 0,7667 | 0     | 0     | 0     | 0     | 1     | 0,2500 | 1     | 0     | 0     | 0     | 1    | 0,2283 | 1           | 1          |
| 6  | 0,5667 | 1     | 0     | 0     | 0     | 1     | 0,7500 | 0     | 1     | 0     | 0     | 0    | 0,2500 | 1           | 0          |
| 7  | 0,1333 | 0     | 0     | 1     | 0     | 0     | 0,2500 | 1     | 0     | 0     | 0     | 1    | 0,3696 | 1           | 0          |
| 8  | 0,2333 | 0     | 0     | 0     | 0     | 1     | 0,7500 | 0     | 0     | 0     | 1     | 0    | 0,7283 | 1           | 1          |
| 9  | 0,3667 | 1     | 1     | 0     | 0     | 0     | 0,0000 | 1     | 0     | 0     | 0     | 1    | 0,5217 | 0           | 0          |
| 10 | 0,0000 | 1     | 0     | 1     | 0     | 0     | 0,0000 | 0     | 1     | 0     | 0     | 0    | 0,0000 | 0           | 0          |

# Calculando a Distância

❑ Resultado:

|        |        |        |        |        |        |        |        |        |        |        |        |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| [1-2]  | 0,0178 | 1,0000 | 1,0000 | 0,2500 | 1,0000 | 1,0000 | 0,0266 | 0,0000 | 1,0000 | 5,2944 | 2,3009 |
| [1-3]  | 0,3211 | 0,0000 | 1,0000 | 0,2500 | 1,0000 | 0,0000 | 0,2954 | 0,0000 | 1,0000 | 3,8665 | 1,9663 |
| [1-4]  | 0,1600 | 1,0000 | 1,0000 | 0,0000 | 0,0000 | 1,0000 | 0,1447 | 1,0000 | 0,0000 | 4,3047 | 2,0748 |
| [1-5]  | 0,1111 | 1,0000 | 1,0000 | 0,0625 | 0,0000 | 0,0000 | 0,0521 | 0,0000 | 0,0000 | 2,2257 | 1,4919 |
| [1-6]  | 0,0178 | 0,0000 | 1,0000 | 0,0625 | 1,0000 | 1,0000 | 0,0427 | 0,0000 | 1,0000 | 4,1229 | 2,0305 |
| [1-7]  | 0,0900 | 1,0000 | 1,0000 | 0,0625 | 0,0000 | 0,0000 | 0,0076 | 0,0000 | 1,0000 | 3,1601 | 1,7777 |
| [1-8]  | 0,0400 | 1,0000 | 1,0000 | 0,0625 | 1,0000 | 1,0000 | 0,0738 | 0,0000 | 0,0000 | 4,1763 | 2,0436 |
| [1-9]  | 0,0044 | 0,0000 | 0,0000 | 0,2500 | 0,0000 | 0,0000 | 0,0043 | 1,0000 | 1,0000 | 2,2587 | 1,5029 |
| [1-10] | 0,1878 | 0,0000 | 1,0000 | 0,2500 | 1,0000 | 1,0000 | 0,2084 | 1,0000 | 1,0000 | 5,6462 | 2,3762 |
| [2-3]  | 0,4900 | 1,0000 | 1,0000 | 1,0000 | 1,0000 | 1,0000 | 0,4992 | 0,0000 | 0,0000 | 5,9892 | 2,4473 |
| [2-4]  | 0,0711 | 0,0000 | 1,0000 | 0,2500 | 1,0000 | 0,0000 | 0,0473 | 1,0000 | 1,0000 | 4,3684 | 2,0901 |
| [2-5]  | 0,2178 | 0,0000 | 1,0000 | 0,0625 | 1,0000 | 1,0000 | 0,0043 | 0,0000 | 1,0000 | 4,2845 | 2,0699 |
| [2-6]  | 0,0711 | 1,0000 | 1,0000 | 0,5625 | 0,0000 | 0,0000 | 0,0019 | 0,0000 | 0,0000 | 2,6355 | 1,6234 |
| [2-7]  | 0,0278 | 0,0000 | 0,0000 | 0,0625 | 1,0000 | 1,0000 | 0,0058 | 0,0000 | 0,0000 | 2,0961 | 1,4478 |
| [2-8]  | 0,0044 | 0,0000 | 1,0000 | 0,5625 | 1,0000 | 0,0000 | 0,1890 | 0,0000 | 1,0000 | 3,7560 | 1,9380 |
| [2-9]  | 0,0044 | 1,0000 | 1,0000 | 0,0000 | 1,0000 | 1,0000 | 0,0521 | 1,0000 | 0,0000 | 5,0565 | 2,2487 |
| [2-10] | 0,0900 | 1,0000 | 0,0000 | 0,0000 | 0,0000 | 0,0000 | 0,0861 | 1,0000 | 0,0000 | 2,1761 | 1,4752 |
| [3-4]  | 0,9344 | 1,0000 | 1,0000 | 0,2500 | 1,0000 | 1,0000 | 0,8536 | 1,0000 | 1,0000 | 8,0381 | 2,8351 |
| [3-5]  | 0,0544 | 1,0000 | 1,0000 | 0,5625 | 1,0000 | 0,0000 | 0,5956 | 0,0000 | 1,0000 | 5,2125 | 2,2831 |
| [3-6]  | 0,1878 | 0,0000 | 1,0000 | 0,0625 | 1,0000 | 1,0000 | 0,5625 | 0,0000 | 0,0000 | 3,8128 | 1,9526 |
| [3-7]  | 0,7511 | 1,0000 | 1,0000 | 0,5625 | 1,0000 | 0,0000 | 0,3974 | 0,0000 | 0,0000 | 4,7111 | 2,1705 |
| [3-8]  | 0,5878 | 1,0000 | 1,0000 | 0,0625 | 1,0000 | 1,0000 | 0,0738 | 0,0000 | 1,0000 | 5,7241 | 2,3925 |
| [3-9]  | 0,4011 | 0,0000 | 1,0000 | 1,0000 | 1,0000 | 0,0000 | 0,2287 | 1,0000 | 0,0000 | 4,6298 | 2,1517 |
| [3-10] | 1,0000 | 0,0000 | 1,0000 | 1,0000 | 1,0000 | 1,0000 | 1,0000 | 1,0000 | 0,0000 | 7,0000 | 2,6458 |
| [4-5]  | 0,5378 | 0,0000 | 0,0000 | 0,0625 | 0,0000 | 1,0000 | 0,0232 | 1,0000 | 0,0000 | 2,6234 | 1,6197 |
| [4-6]  | 0,2844 | 1,0000 | 0,0000 | 0,0625 | 1,0000 | 0,0000 | 0,0302 | 1,0000 | 1,0000 | 4,3772 | 2,0922 |
| [4-7]  | 0,0100 | 0,0000 | 1,0000 | 0,0625 | 0,0000 | 1,0000 | 0,0861 | 1,0000 | 1,0000 | 4,1586 | 2,0393 |
| [4-8]  | 0,0400 | 0,0000 | 0,0000 | 0,0625 | 1,0000 | 0,0000 | 0,4253 | 1,0000 | 0,0000 | 2,5278 | 1,5899 |
| [4-9]  | 0,1111 | 1,0000 | 1,0000 | 0,2500 | 0,0000 | 1,0000 | 0,1986 | 0,0000 | 1,0000 | 4,5597 | 2,1353 |
| [4-10] | 0,0011 | 1,0000 | 1,0000 | 0,2500 | 1,0000 | 0,0000 | 0,0058 | 0,0000 | 1,0000 | 4,2569 | 2,0632 |

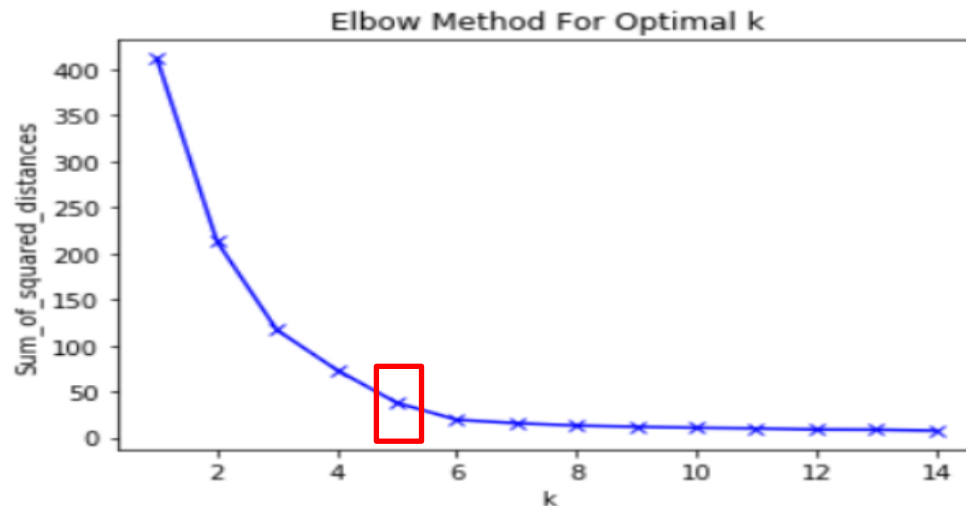
# Calculando a Distância

□ Juntando (merging) objetos:

| Nº | Idade | Idade_N | Gênero | Estado Civil | Filhos | Filhos_N | Escolaridade  | CC  | Renda     | Renda_N | Cartão_Cr. | Imóvel_P |
|----|-------|---------|--------|--------------|--------|----------|---------------|-----|-----------|---------|------------|----------|
| 1  | 45    | 0,4333  | Masc   | Divorciado   | 2      | 0,5000   | Superior      | Sim | 5.000,00  | 0,4565  | Sim        | Sim      |
| 2  | 37    | 0,3000  | Femi   | Solteiro     | 0      | 0,0000   | Médio         | Não | 3.500,00  | 0,2935  | Sim        | Não      |
| 3  | 79    | 1,0000  | Masc   | Viúvo        | 4      | 1,0000   | Fundamental   | Sim | 10.000,00 | 1,0000  | Sim        | Não      |
| 4  | 21    | 0,0333  | Femi   | Casado       | 2      | 0,5000   | Superior      | Não | 1.500,00  | 0,0761  | Não        | Sim      |
| 5  | 65    | 0,7667  | Femi   | Casado       | 1      | 0,2500   | Superior      | Sim | 2.900,00  | 0,2283  | Sim        | Sim      |
| 6  | 53    | 0,5667  | Masc   | Casado       | 3      | 0,7500   | Médio         | Não | 3.100,00  | 0,2500  | Sim        | Não      |
| 7  | 27    | 0,1333  | Femi   | Solteiro     | 1      | 0,2500   | Superior      | Sim | 4.200,00  | 0,3696  | Sim        | Não      |
| 8  | 33    | 0,2333  | Femi   | Casado       | 3      | 0,7500   | Pós-graduação | Não | 7.500,00  | 0,7283  | Sim        | Sim      |
| 9  | 41    | 0,3667  | Masc   | Divorciado   | 0      | 0,0000   | Superior      | Sim | 5.600,00  | 0,5217  | Não        | Não      |
| 10 | 19    | 0,0000  | Masc   | Solteiro     | 0      | 0,0000   | Médio         | Não | 800,00    | 0,0000  | Não        | Não      |

# Como escolher o número de grupos?

- Há uma variedade de métodos utilizados para escolher (identificar) o número ideal de grupos:
  - ❖ O “Elbow” Method (DB)
  - ❖ Silhouette Method
  - ❖ Outros métodos de Clustering Validation.



# Qual o número de grupos?

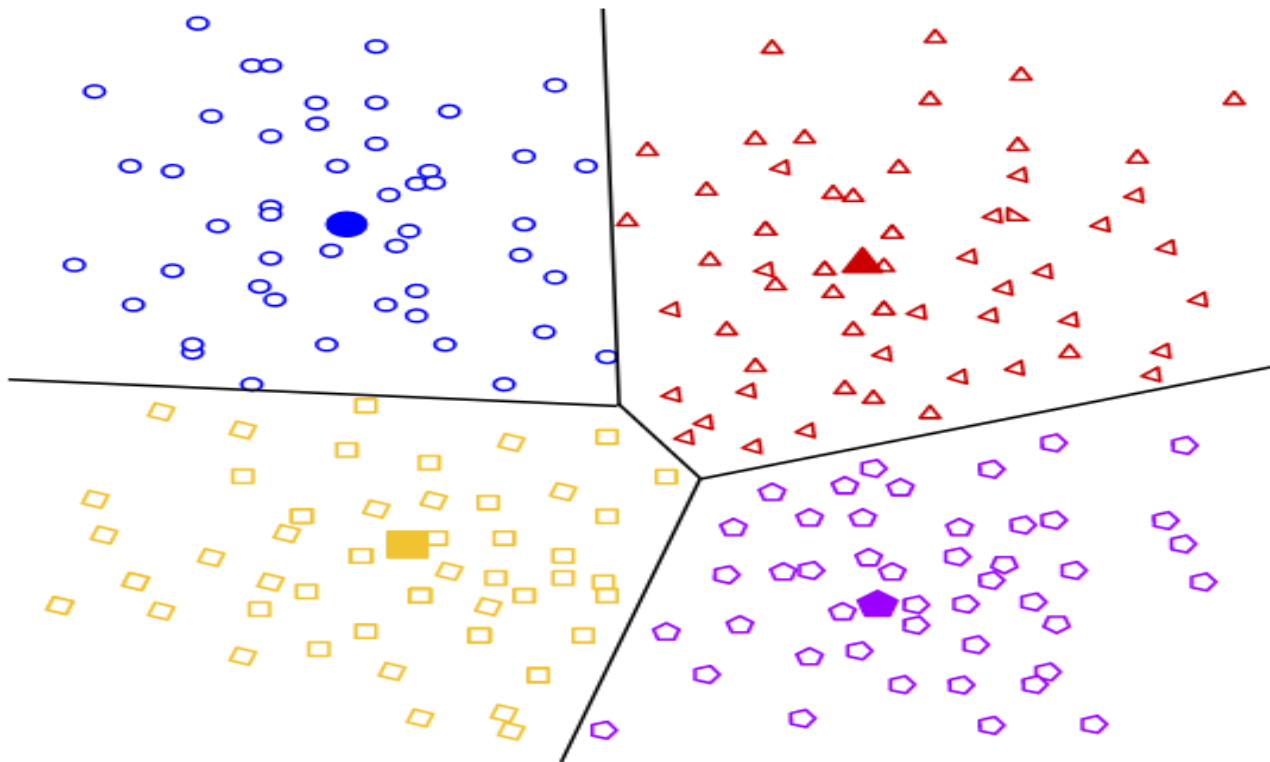
□ Devemos observar:

- ❖ Sabemos que quando nos afastamos de grupos unitários, a **homogeneidade diminui**.
- ❖ Uma instância por grupo **não define** estrutura.
- ❖ Devemos verificar cada solução para a sua descrição de estrutura **versus** a homogeneidade dos grupos.
- ❖ Combinar dois ou mais métodos de validação.

# k-Means

# Algoritmos de Clustering

## □ Centroid-based Clustering:



<https://developers.google.com/machine-learning/clustering/clustering-algorithms>



# História

- ❑ k-Means também chamado de **k-Médias**.
- ❑ É um algoritmo de Agrupamento (***Clustering***) que objetiva **particionar**  $n$  objetos em  $k$  grupos, onde cada objeto pertence ao grupo mais próximo da média.
- ❑ Foi empregado primeiramente por **James MacQueen** em **1967**.

# k-Means

- ❑ Difere do agrupamento hierárquico de várias maneiras.  
Em particular:
  - ❖ Não há hierarquias, os dados são **particionados**.
  - ❖ Ou seja, a solução de seis grupos não é apenas a combinação de dois grupos a partir de uma solução com sete grupos, como no hierárquico.
- ❑ O resultado é apenas a **pertinência** final de cada **padrão** relacionado aos grupos.
- ❑ O número de grupos permitido (**k**) tem que ser **definido a priori**.

# *k*-Means: Algoritmo

- ❑ Passo 1: os primeiros  $k$  centros dos grupos são escolhidos aleatoriamente.
- ❑ Passo2: cada objeto é atribuído ao grupo associado com o centro mais próximo.
- ❑ Passo3: compute um novo centro para cada grupo (média dos valores de todos os objetos - centróide).
- ❑ Passo4: repita passo2 (com os novos centros) e passo3 até que não haja mudança nos centros.

# $k$ -Means: exemplo (1/7)

- ❑ Passo 1: os primeiros  $k$  centros dos grupos são escolhidos aleatoriamente.

|             | X1    | X2     | X3    | X4    | X5     |
|-------------|-------|--------|-------|-------|--------|
| Cliente_1   | 7,000 | 10,000 | 9,000 | 7,000 | 10,000 |
| → Cliente_2 | 9,000 | 9,000  | 8,000 | 9,000 | 9,000  |
| Cliente_3   | 5,000 | 5,000  | 6,000 | 7,000 | 7,000  |
| Cliente_4   | 6,000 | 6,000  | 3,000 | 3,000 | 4,000  |
| → Cliente_5 | 1,000 | 2,000  | 2,000 | 1,000 | 2,000  |
| Cliente_6   | 4,000 | 3,000  | 2,000 | 3,000 | 3,000  |
| Cliente_7   | 2,000 | 4,000  | 5,000 | 2,000 | 5,000  |

K-Means com Correlação de Pearson e  $k=2$ .

# *k*-Means: exemplo (2/7)

- ❑ Passo2: cada objeto é atribuído ao grupo associado com o centro mais próximo.

|           | Cliente_1 | Cliente_2 | Cliente_3 | Cliente_4 | Cliente_5 | Cliente_6 | Cliente_7 |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Cliente_1 | 1,000     |           |           |           |           |           |           |
| Cliente_2 | -0,147    | 1,000     | 0,000     | 0,516     | -0,408    | 0,791     | -0,516    |
| Cliente_3 | 0,000     | 0,000     | 1,000     |           |           |           |           |
| Cliente_4 | 0,087     | 0,516     | -0,824    | 1,000     |           |           |           |
| Cliente_5 | 0,963     | -0,408    | 0,000     | -0,060    | 1,000     | -0,645    | 0,963     |
| Cliente_6 | -0,466    | 0,791     | -0,354    | 0,699     | -0,645    | 1,000     |           |
| Cliente_7 | 0,891     | -0,516    | 0,165     | -0,239    | 0,963     | -0,699    | 1,000     |

# *k*-Means: Exemplo (3/7)

- ❑ Passo3: compute um novo centro para cada grupo (média dos valores de todos os objetos - centróide).

|            | X1    | X2    | X3    | X4    | X5    |
|------------|-------|-------|-------|-------|-------|
| Cliente_2  | 9,000 | 9,000 | 8,000 | 9,000 | 9,000 |
| Cliente_3  | 5,000 | 5,000 | 6,000 | 7,000 | 7,000 |
| Cliente_4  | 6,000 | 6,000 | 3,000 | 3,000 | 4,000 |
| Cliente_6  | 4,000 | 3,000 | 2,000 | 3,000 | 3,000 |
| → Centro_1 | 6,000 | 5,750 | 4,750 | 5,500 | 5,750 |

|            | X1    | X2     | X3    | X4    | X5     |
|------------|-------|--------|-------|-------|--------|
| Cliente_1  | 7,000 | 10,000 | 9,000 | 7,000 | 10,000 |
| Cliente_5  | 1,000 | 2,000  | 2,000 | 1,000 | 2,000  |
| Cliente_7  | 2,000 | 4,000  | 5,000 | 2,000 | 5,000  |
| → Centro_2 | 3,333 | 5,333  | 5,333 | 3,333 | 5,667  |

# *k*-Means: Exemplo (4/7)

- ❑ Passo2: cada objeto é atribuído ao grupo associado com o centro mais próximo.

|           | Cliente_1 | Cliente_2 | Cliente_3 | Cliente_4 | Cliente_5 | Cliente_6 | Cliente_7 | Centro_1 | Centro_2 |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|----------|----------|
| Cliente_1 | 1         | -0,1474   | 0         | 0,087     | 0,9631    | -0,4663   | 0,8913    | -0,1371  | 0,9723   |
| Cliente_2 | -0,1474   | 1         | 0         | 0,516     | -0,4082   | 0,7906    | -0,516    | 0,93     | -0,3498  |
| Cliente_3 | 0         | 0         | 1         | -0,8242   | 0         | -0,3536   | 0,1648    | -0,2599  | 0,068    |
| Cliente_4 | 0,087     | 0,516     | -0,8242   | 1         | -0,0602   | 0,6994    | -0,2391   | 0,737    | -0,0698  |
| Cliente_5 | 0,9631    | -0,4082   | 0         | -0,0602   | 1         | -0,6455   | 0,9631    | -0,3797  | 0,9926   |
| Cliente_6 | -0,4663   | 0,7906    | -0,3536   | 0,6994    | -0,6455   | 1         | -0,6994   | 0,919    | -0,6011  |
| Cliente_7 | 0,8913    | -0,516    | 0,1648    | -0,2391   | 0,9631    | -0,6994   | 1         | -0,4799  | 0,9723   |
| Centro_1  | -0,1371   | 0,93      | -0,2599   | 0,737     | -0,3797   | 0,919     | -0,4799   | 1        | -0,322   |
| Centro_2  | 0,9723    | -0,3498   | 0,068     | -0,0698   | 0,9926    | -0,6011   | 0,9723    | -0,322   | 1        |

# *k*-Means: Exemplo (5/7)

- ❑ Passo3: compute um novo centro para cada grupo (média dos valores de todos os objetos - centróide).

|            | X1    | X2    | X3    | X4    | X5    |
|------------|-------|-------|-------|-------|-------|
| Cliente_2  | 9.000 | 9000  | 8000  | 9000  | 9000  |
| Cliente_4  | 6.000 | 6000  | 3000  | 3000  | 4000  |
| Cliente_6  | 4.000 | 3000  | 2000  | 3000  | 3000  |
| → Centro_1 | 6.333 | 6.000 | 4.333 | 5.000 | 5.333 |

|            | X1    | X2    | X3    | X4    | X5    |
|------------|-------|-------|-------|-------|-------|
| Cliente_1  | 7.000 | 10000 | 9000  | 7000  | 10000 |
| Cliente_3  | 5.000 | 5000  | 6000  | 7000  | 7000  |
| Cliente_5  | 1.000 | 2000  | 2000  | 1000  | 2000  |
| Cliente_7  | 2.000 | 4000  | 5000  | 2000  | 5000  |
| → Centro_2 | 3.750 | 5.250 | 5.500 | 4.250 | 6.000 |



# *k*-Means: Exemplo (6/7)

- ❑ Passo2: cada objeto é atribuído ao grupo associado com o centro mais próximo.

|           | Cliente_1 | Cliente_2 | Cliente_3 | Cliente_4 | Cliente_5 | Cliente_6 | Cliente_7 | Centro_1 | Centro_2 |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|----------|----------|
| Cliente_1 | 1         | -0,1474   | 0         | 0,087     | 0,9631    | -0,4663   | 0,8913    | -0,1106  | 0,9175   |
| Cliente_2 | -0,1474   | 1         | 0         | 0,516     | -0,4082   | 0,7906    | -0,516    | 0,75     | -0,3323  |
| Cliente_3 | 0         | 0         | 1         | -0,8242   | 0         | -0,3536   | -0,6281   | 0,3377   |          |
| Cliente_4 | 0,087     | 0,516     | -0,8242   | 1         | -0,0602   | 0,6994    | -0,2391   | 0,9389   | -0,2939  |
| Cliente_5 | 0,9631    | -0,4082   | 0         | -0,0602   | 1         | -0,6455   | 0,9631    | -0,3062  | 0,9372   |
| Cliente_6 | -0,4663   | 0,7906    | -0,3536   | 0,6994    | -0,6455   | 1         | -0,6994   | 0,8883   | -0,6686  |
| Cliente_7 | 0,8913    | -0,516    | 0,1648    | -0,2391   | 0,9631    | -0,6994   | 1         | -0,4564  | 0,962    |
| Centro_1  | -0,1106   | 0,75      | -0,6281   | 0,9389    | -0,3062   | 0,8883    | -0,4564   | 1        | -0,4473  |
| Centro_2  | 0,9175    | -0,3323   | 0,3377    | -0,2939   | 0,9372    | -0,6686   | 0,962     | -0,4473  | 1        |

# *k*-Means: Exemplo (7/7)

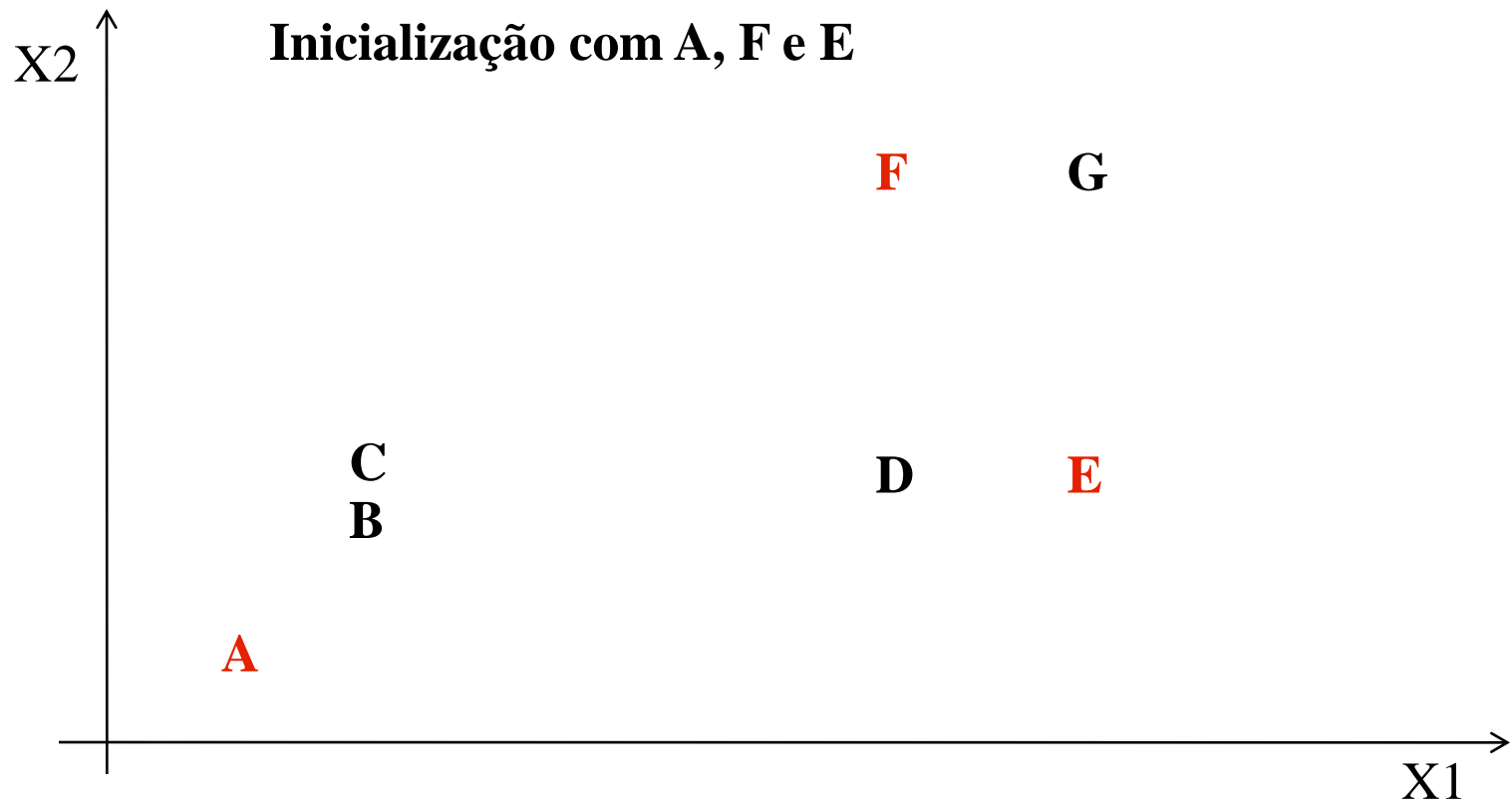
□ Fim, pois não houve mudança nos centros.

|            | X1    | X2    | X3    | X4    | X5    |
|------------|-------|-------|-------|-------|-------|
| Cliente_2  | 9,000 | 9,000 | 8,000 | 9,000 | 9,000 |
| Cliente_4  | 6,000 | 6,000 | 3,000 | 3,000 | 4,000 |
| Cliente_6  | 4,000 | 3,000 | 2,000 | 3,000 | 3,000 |
| → Centro_1 | 6,333 | 6,000 | 4,333 | 5,000 | 5,333 |

|            | X1    | X2     | X3    | X4    | X5     |
|------------|-------|--------|-------|-------|--------|
| Cliente_1  | 7,000 | 10,000 | 9,000 | 7,000 | 10,000 |
| Cliente_3  | 5,000 | 5,000  | 6,000 | 7,000 | 7,000  |
| Cliente_5  | 1,000 | 2,000  | 2,000 | 1,000 | 2,000  |
| Cliente_7  | 2,000 | 4,000  | 5,000 | 2,000 | 5,000  |
| → Centro_2 | 3,750 | 5,250  | 5,500 | 4,250 | 6,000  |

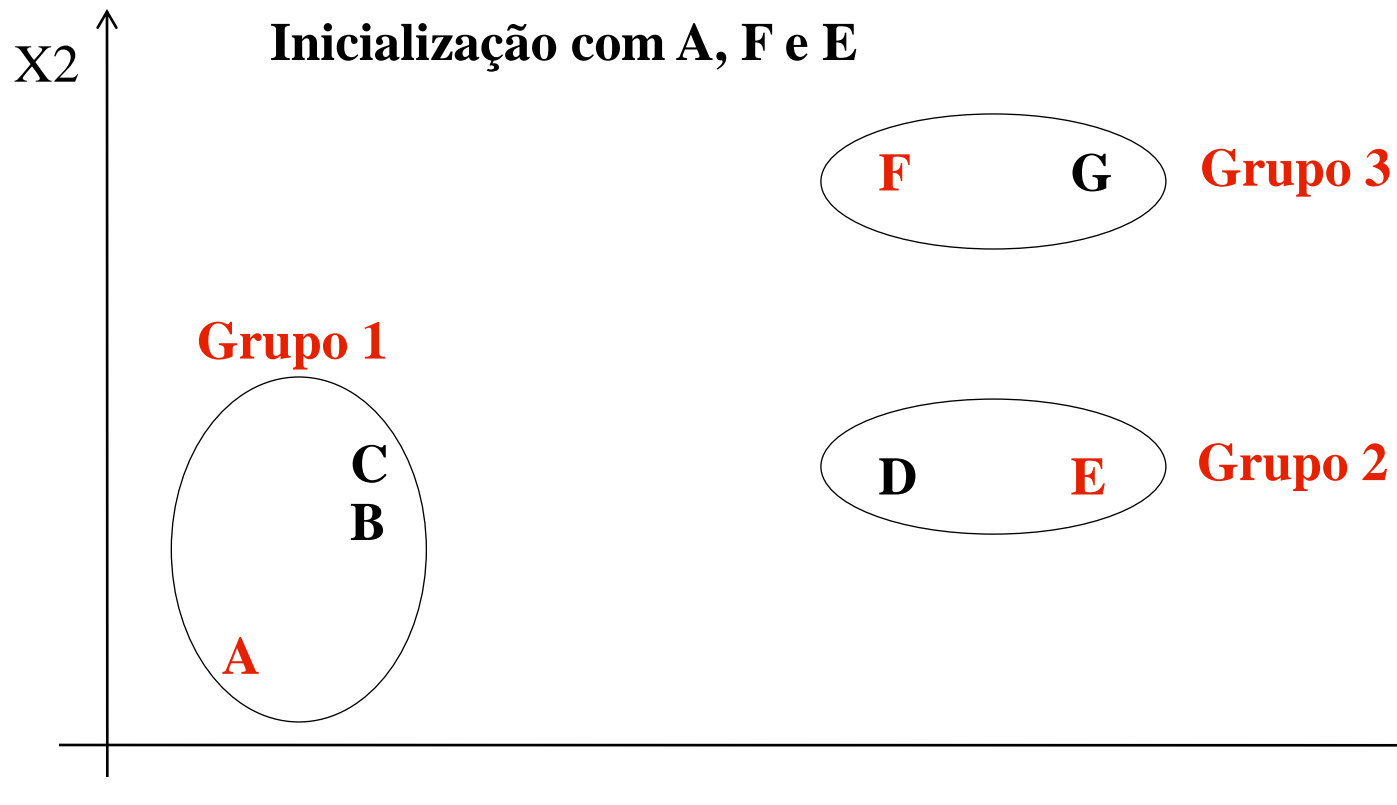
# *k*-Means

## ❑ Sensibilidade à condição inicial:



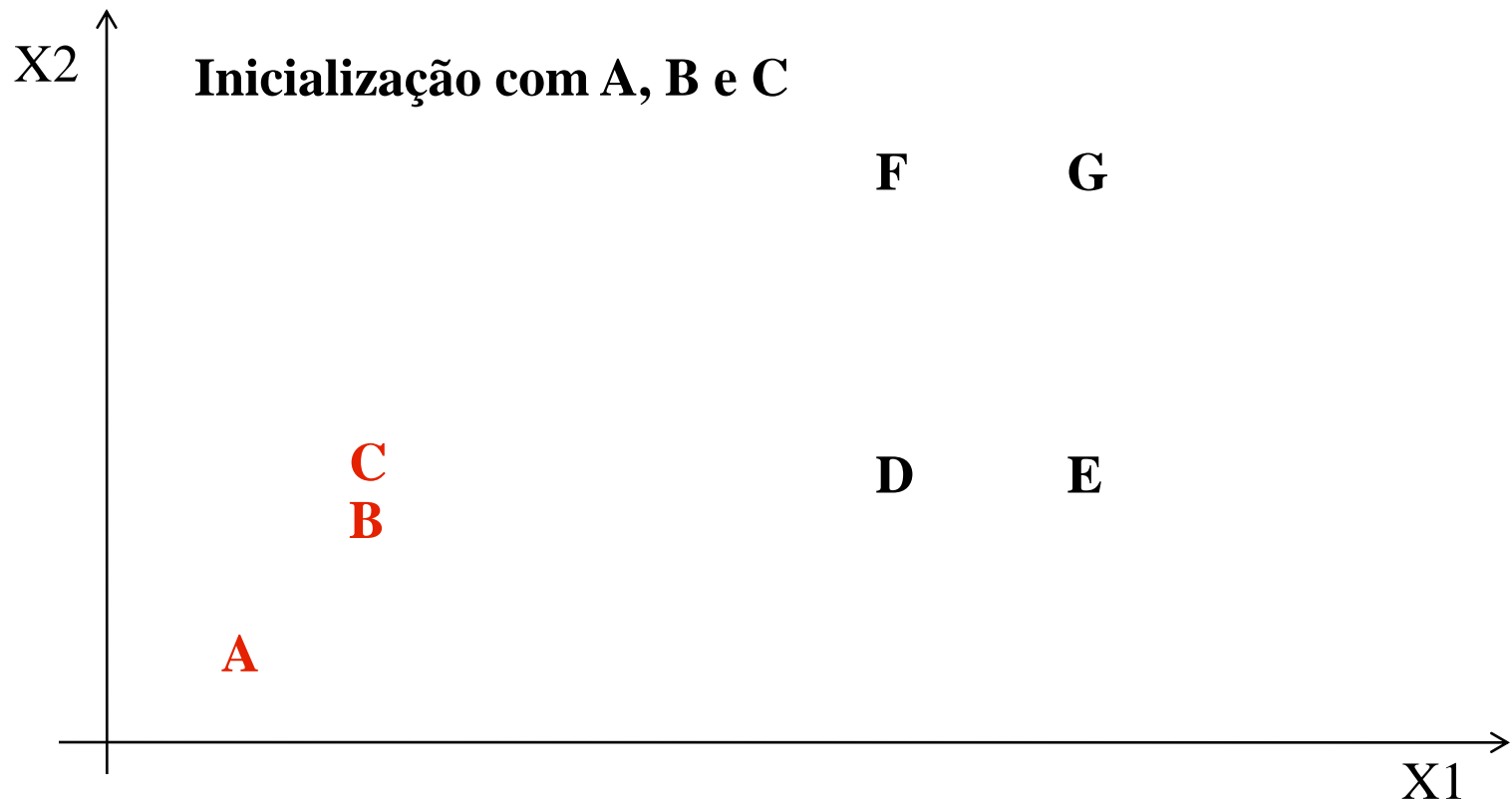
# $k$ -Means

❑ Sensibilidade à condição inicial:



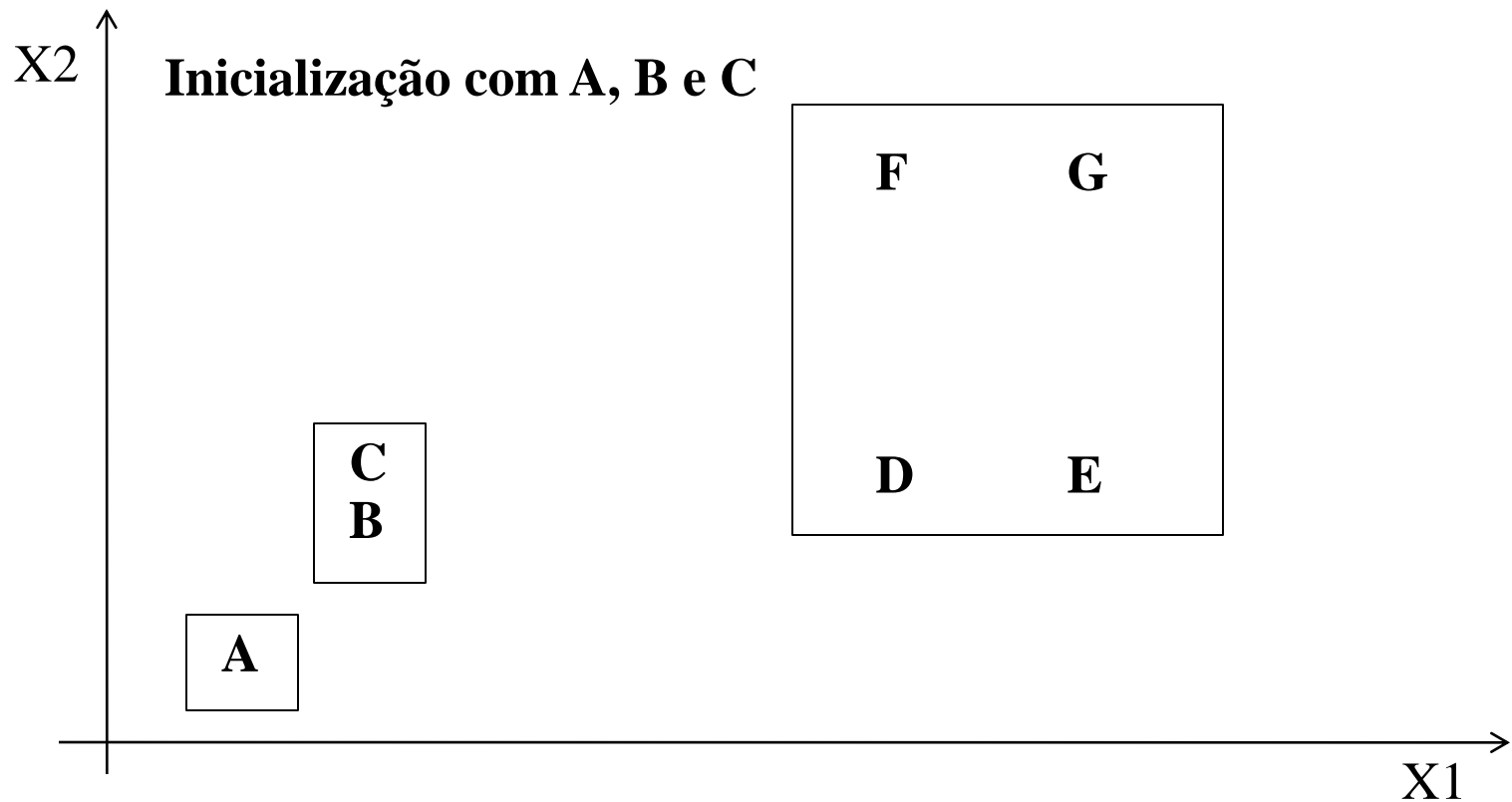
# $k$ -Means

❑ Sensibilidade à condição inicial:



# *k*-Means

## ❑ Sensibilidade à condição inicial:



# $k$ -Means

## ❑ Características:

- ❖ Partição.
- ❖ O número de grupos deve ser definido a priori.
- ❖ Não-determinístico: inicializações aleatórias dos centros.
- ❖ Grupos (clusters) esféricos.

## ❑ Dificuldades:

- ❖ Inicialização dos centros.

# Dúvidas ...



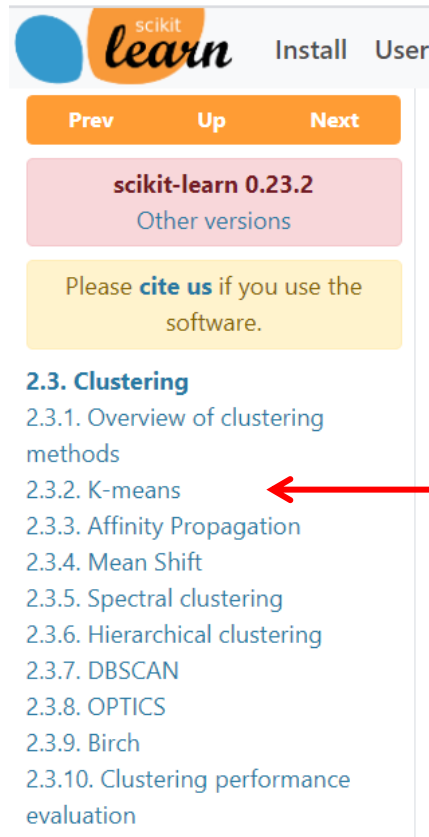


# *k*-Means

- ❑ Abrir dataset **PessoaNormBinary.csv**;
- ❑ Utilizar o algoritmo Simple KMeans
  - ❖ numClusters = 3;
  - ❖ seed= 10;
  - ❖ distanceFunction = EuclideanDistance.
- ❑ Salvar arquivo resultante:
  - ❖ PessoaNormBinary\_kM-s10-3k.arff

# k-Means

## Utilizando k-Means (scikit-learn):



The screenshot shows the scikit-learn documentation interface. At the top, there's a navigation bar with 'scikit-learn', 'Install', and 'User' links. Below this is a sidebar with a 'Prev' button, an 'Up' button, and a 'Next' button. The main content area of the sidebar shows 'scikit-learn 0.23.2' and a link to 'Other versions'. Below that is a yellow box with the text 'Please cite us if you use the software.' The main content area of the page shows a list of clustering methods: 2.3. Clustering, 2.3.1. Overview of clustering methods, 2.3.2. K-means (highlighted with a red arrow), 2.3.3. Affinity Propagation, 2.3.4. Mean Shift, 2.3.5. Spectral clustering, 2.3.6. Hierarchical clustering, 2.3.7. DBSCAN, 2.3.8. OPTICS, 2.3.9. Birch, and 2.3.10. Clustering performance evaluation.

### 2.3.2. K-means

The **KMeans** algorithm clusters data by trying to separate samples in  $n$  groups of equal variance, minimizing a criterion known as the *inertia* or within-cluster sum-of-squares (see below). This algorithm requires the number of clusters to be specified. It scales well to large number of samples and has been used across a large range of application areas in many different fields.

The k-means algorithm divides a set of  $N$  samples  $X$  into  $K$  disjoint clusters  $C$ , each described by the mean  $\mu_j$  of the samples in the cluster. The means are commonly called the cluster “centroids”; note that they are not, in general, points from  $X$ , although they live in the same space.

The K-means algorithm aims to choose centroids that minimise the **inertia**, or **within-cluster sum-of-squares criterion**:

$$\sum_{i=0}^n \min_{\mu_j \in C} (||x_i - \mu_j||^2)$$

Inertia can be recognized as a measure of how internally coherent clusters are. It suffers from various drawbacks:

- Inertia makes the assumption that clusters are convex and isotropic, which is not always the case. It responds poorly to elongated clusters, or manifolds with irregular shapes.
- Inertia is not a normalized metric: we just know that lower values are better and zero is optimal. But in very high-dimensional spaces, Euclidean distances tend to become inflated (this is an instance of the so-called “curse of dimensionality”). Running a dimensionality reduction algorithm such as [Principal component analysis \(PCA\)](#) prior to k-means clustering can alleviate this problem and speed up the computations.

# *k*-Means

## ❑ Utilizando **k-Means** (scikit-learn):

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
import io
from google.colab import files

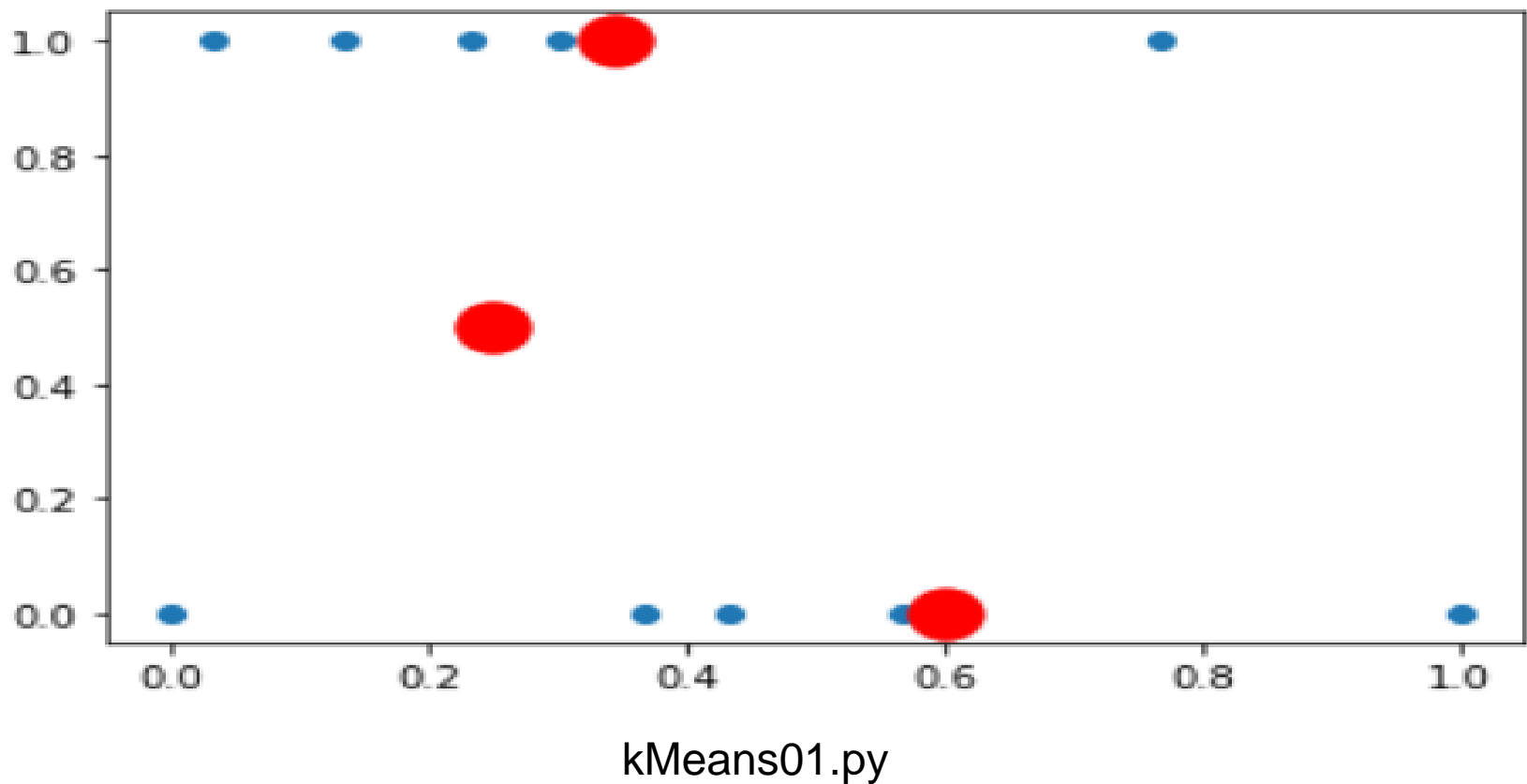
uploaded = files.upload()
dados = pd.read_csv(io.BytesIO(uploaded['PessoaNormBinary.csv']))

# ## kMeans
km = KMeans(n_clusters=3)
km.fit(dados)
centroids = km.cluster_centers_

plt.scatter(dados.iloc[:,0], dados.iloc[:,1])
plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=300)
plt.show()
```

# *k*-Means

❑ Utilizando **k-Means** (scikit-learn):



# *k*-Means

## ❑ Utilizando **k-Means** (scikit-learn):

```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
import io
from google.colab import files

uploaded = files.upload()
dados = pd.read_csv(io.BytesIO(uploaded['PessoaNormBinary.csv']))
dados.info()

# ## kMeans
km = KMeans(n_clusters=3, init='k-means++', max_iter=300, n_init=10,
            random_state=0)
km.fit(dados)
km.fit_predict(dados)
```

# *k*-Means

## ❑ Utilizando *k*-Means (scikit-learn):

```
# juntando os labels com o restante do dataset
dados["Cluster"] = km.labels_
dados["Cluster"] = 'cluster' + dados["Cluster"].astype(str)

# Visualização dos novos atributos
dados.head()

# Salvando Pessoa.csv transformado
df = pd.DataFrame(dados)
df.to_csv('Pessoa_Clustered_kM_k3.csv')

# Download do arquivo transformado
files.download('Pessoa_Clustered_kM_k3.csv')
```

# *k*-Means

❑ Utilizando **k-Means** (scikit-learn):

| N                    | O               | P               | Q        |
|----------------------|-----------------|-----------------|----------|
| RendaBruta           | CartaoCredito_N | ImovelProprio_N | Cluster  |
| 0.456522000000000004 | 0               | 0               | cluster1 |
| 0.293478             | 0               | 1               | cluster2 |
| 1.0                  | 0               | 1               | cluster1 |
| 0.076087             | 1               | 0               | cluster0 |
| 0.228261000000000002 | 0               | 0               | cluster0 |
| 0.25                 | 0               | 1               | cluster2 |
| 0.369565000000000003 | 0               | 1               | cluster1 |
| 0.7282609999999999   | 0               | 0               | cluster0 |
| 0.521739000000000001 | 1               | 1               | cluster1 |
| 0.0                  | 1               | 1               | cluster2 |

# Obrigado!!!

