

Tera

AULA 27

Unsupervised Learning:
Clustering

Instrutor: [Raphael Ballet](#)

Background:

- Engenheiro de Controle e Automação (IMT)
- Mestre em Sistemas Aeroespaciais e Mecatrônica (ITA)
- Data Scientist - Elo7

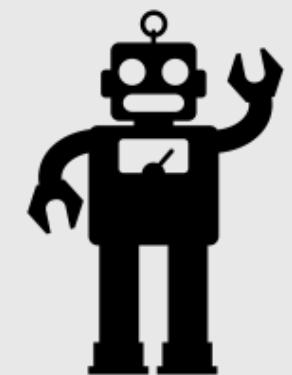
Interesses:



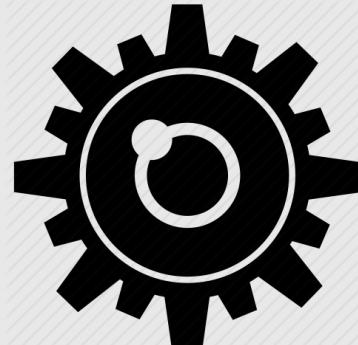
Drones



Aprendizado
de Máquina



Robótica



Visão
Computacional



Processamento de
Linguagem Natural



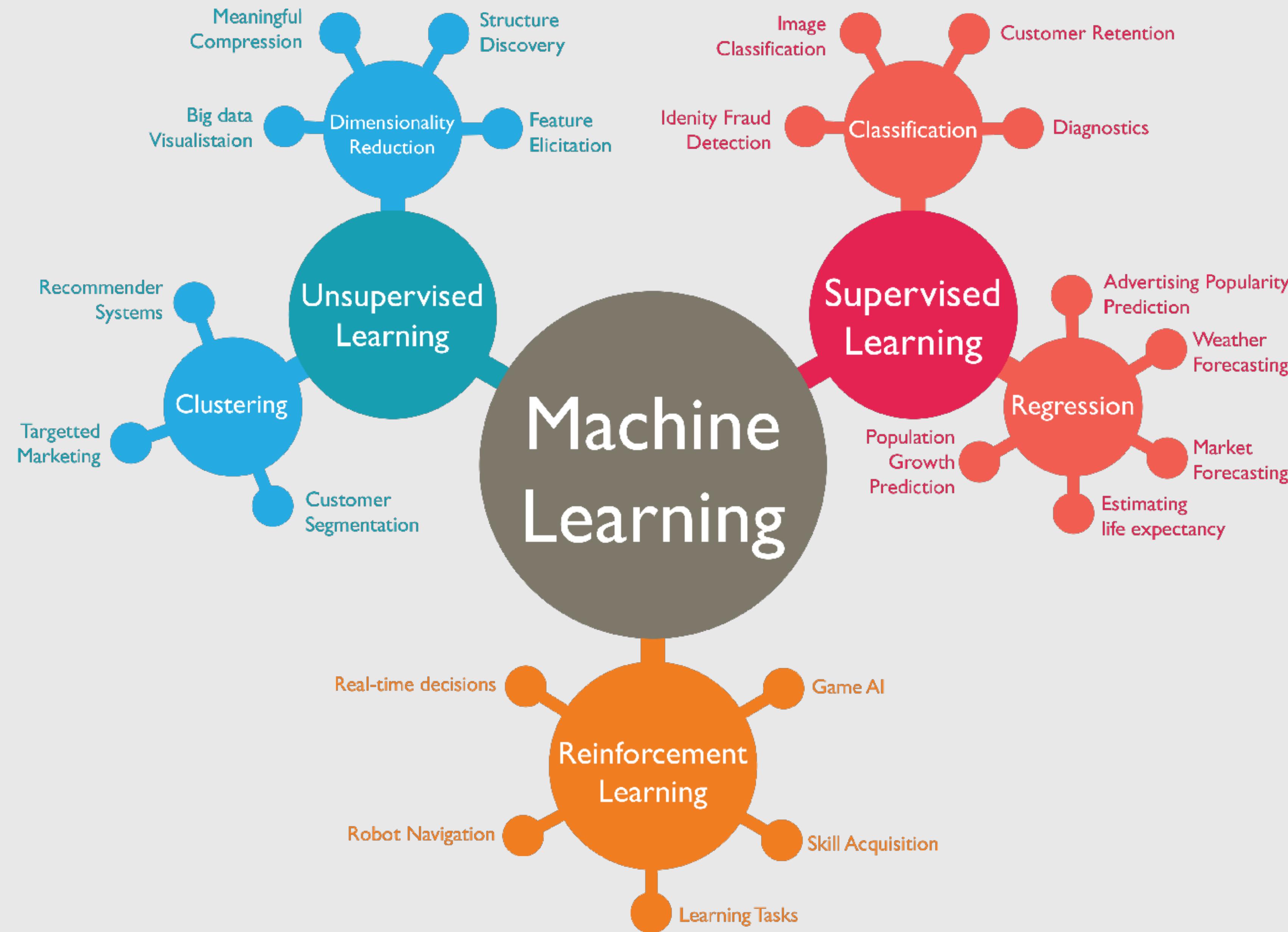
Sistemas de
recomendação

Planejamento:

1. Introdução a clustering
2. Clustering: K-Means
3. Clustering: Hierarchical Clustering
4. Clustering: DBSCAN
5. Maldição da dimensionalidade
6. Redução de dimensionalidade
7. Clustering em NLP

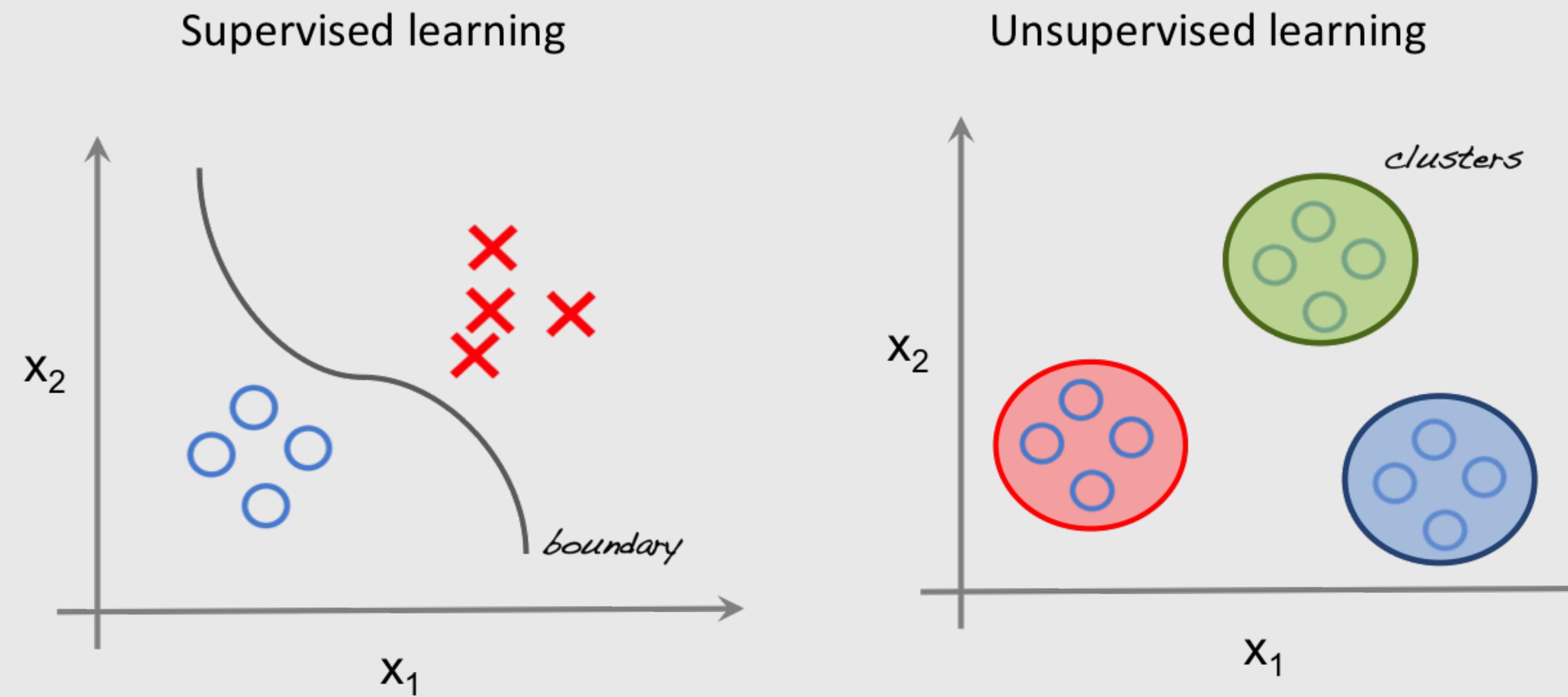
T

1. INTRODUÇÃO



1. INTRODUÇÃO:

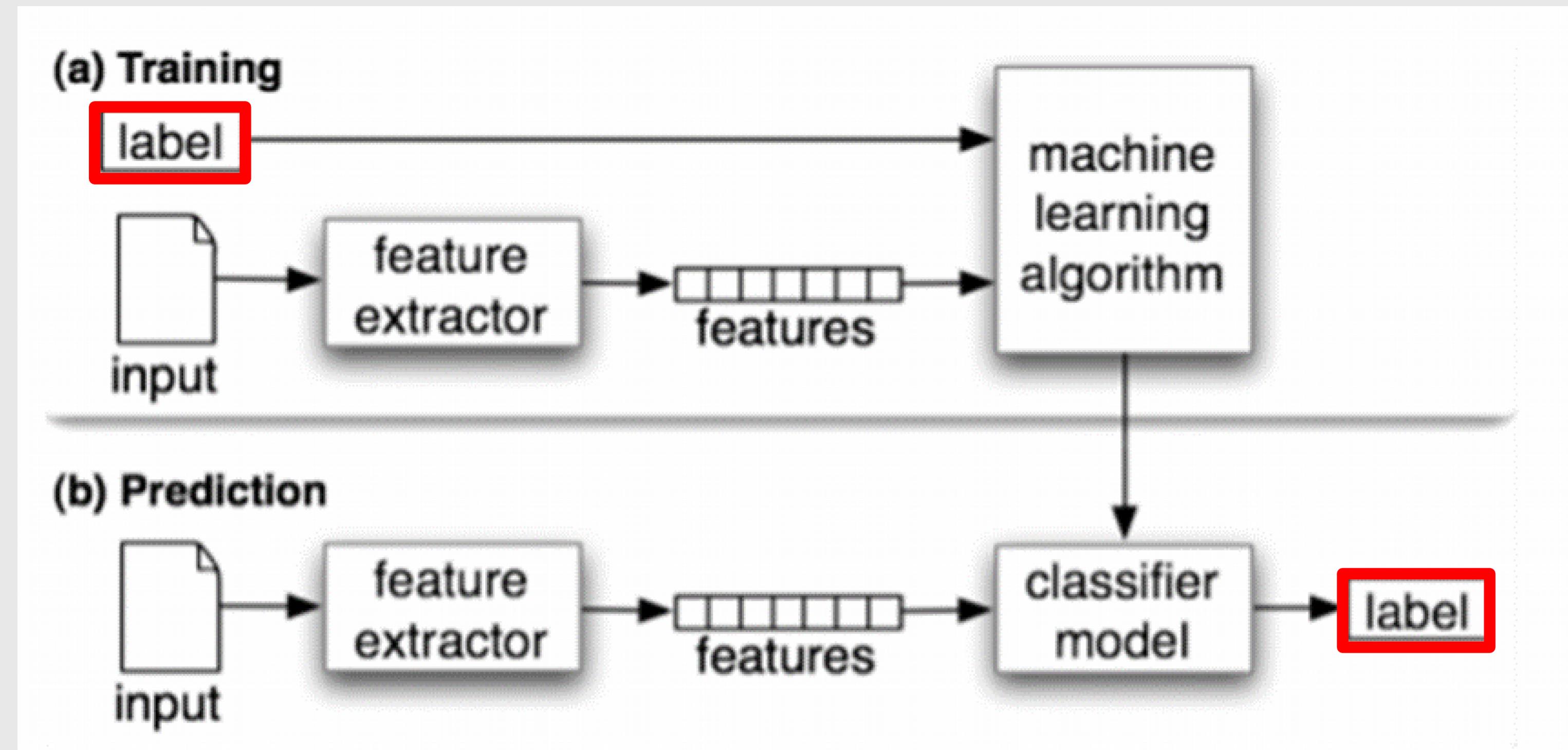
- Aprendizagem supervisionada vs.
Aprendizagem não supervisionada



I

Aprendizagem supervisionada:

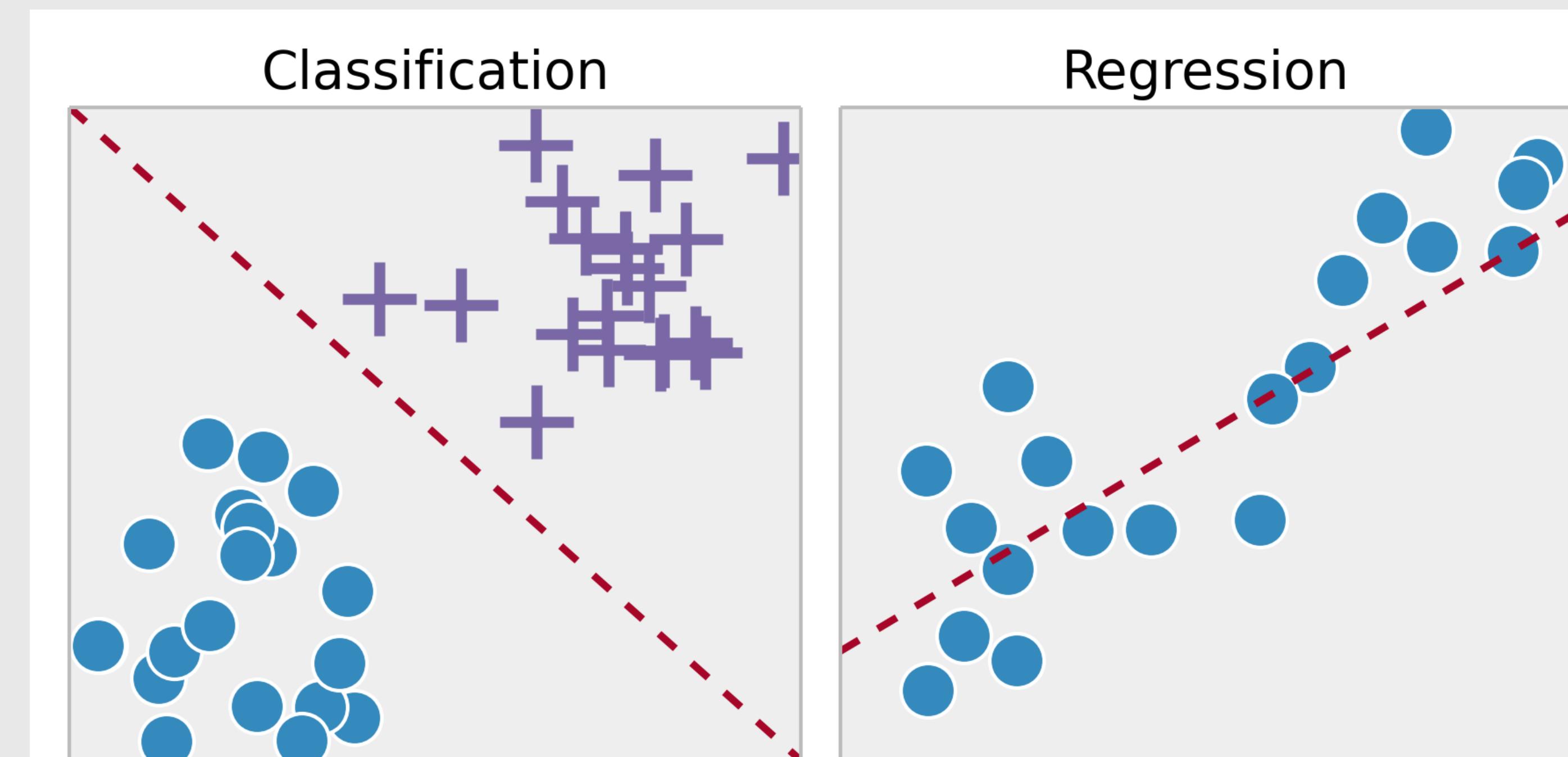
- Dados possuem **rótulo** (label)
- Treinamento com rótulo = Supervisão
- Métodos: Classificação / Regressão



I

Aprendizagem supervisionada:

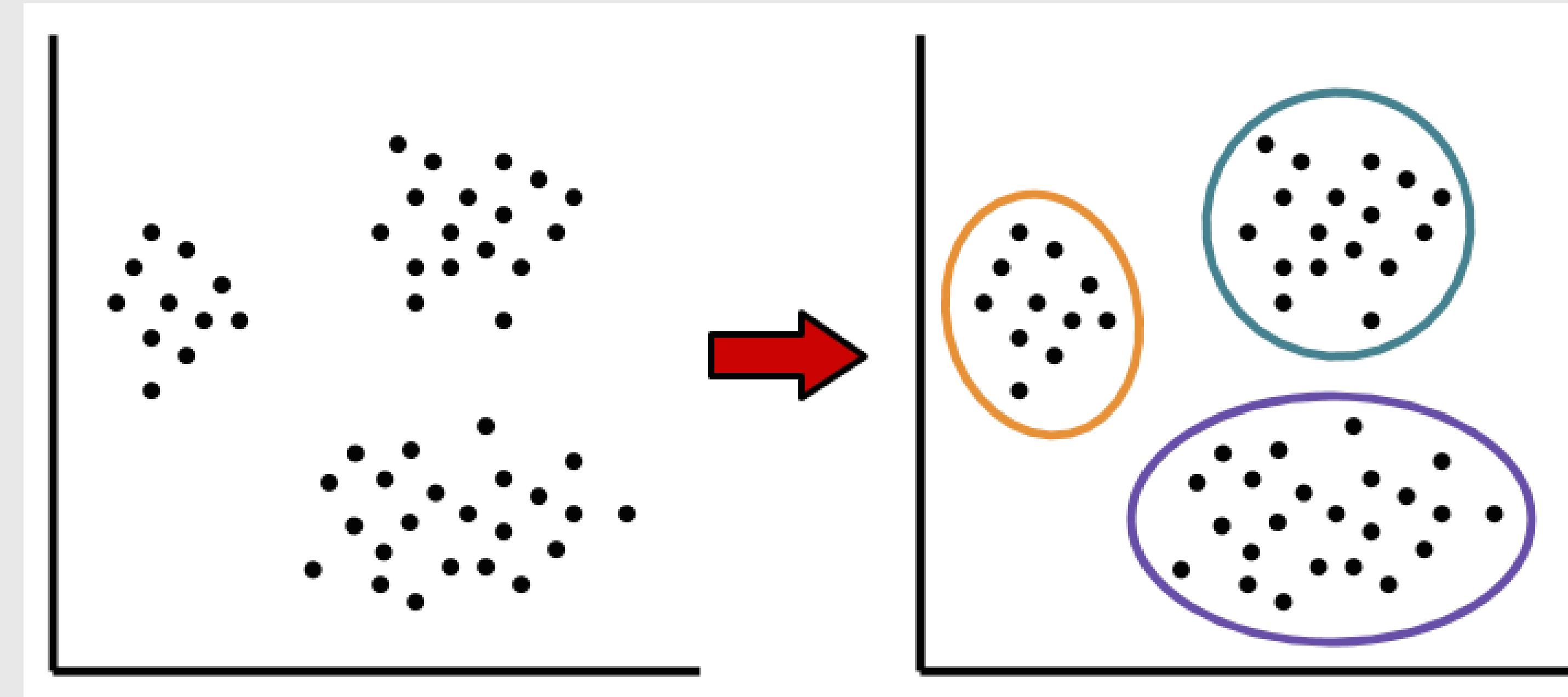
- Análises determinísticas
- Métricas de sucesso bem definidas
- Ex: Crédito, identificação de câncer, previsão de séries temporais financeiras etc.



T

Aprendizagem não supervisionada:

- Dados não rotulados (sem supervisão)
- Métodos: Clustering / density estimation
- Análises exploratórias
- Métricas de sucesso qualitativas e não global
- Ex: padrões de usuários, cluster de produtos / clientes etc.



I

Aprendizagem não supervisionada:

- Exemplo:

Entrada (X)

usa barba	usa óculos	fala estranho
Não	Sim	Sim
Não	Sim	Não
Sim	Não	Sim
Sim	Não	Sim
Sim	Não	Não
Sim	Sim	Sim

I

Aprendizagem supervisionada:

- Exemplo:

Entrada (X)			Saída (Y)
usa barba	usa óculos	fala estranho	cientista de dados
Não	Sim	Sim	Sim
Não	Sim	Não	Não
Sim	Não	Sim	Sim
Sim	Não	Sim	Sim
Sim	Não	Não	Não
Sim	Sim	Sim	Sim

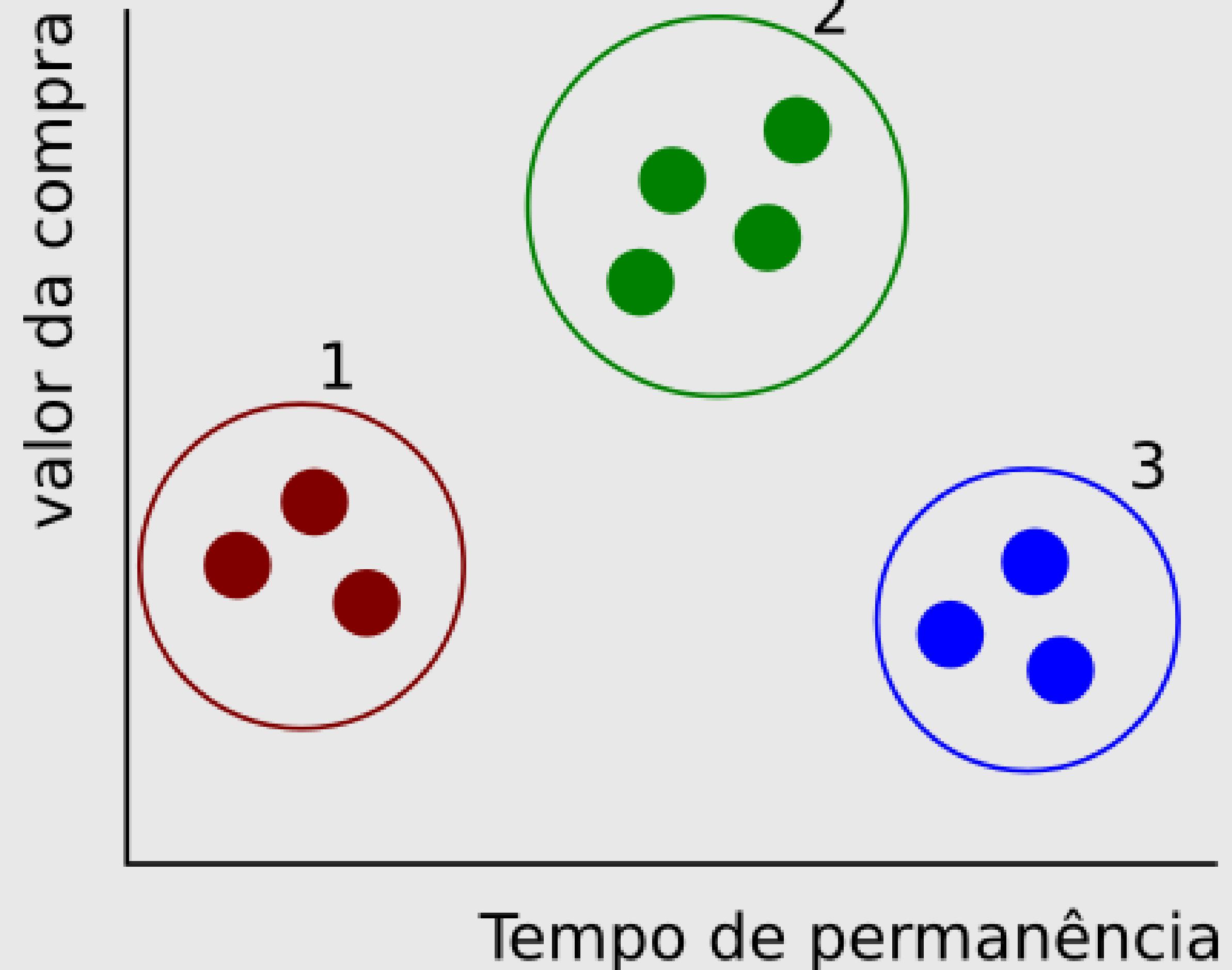
INTRODUÇÃO:

- **Objetivos:**
 - Análise exploratória
 - Encontrar padrões nos dados
 - Resumir dados



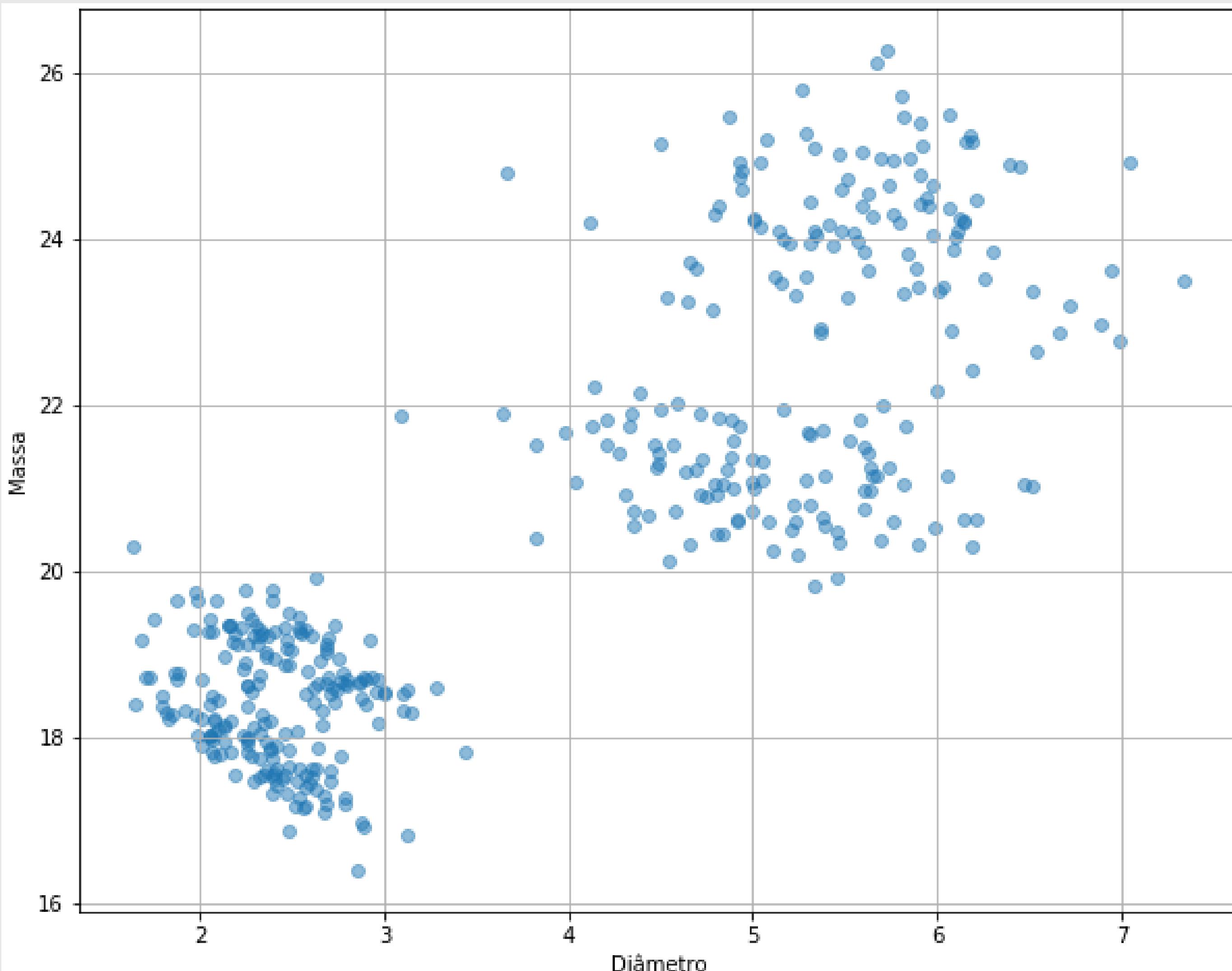
Exemplo:

- **Segmentação usuários:**
 - 1: Compras específicas (Ex: compras para bebês)
 - 2: Compras para eventos (Ex: preparação de casamento)
 - 3: Exploração do site - Não procura nada em específico



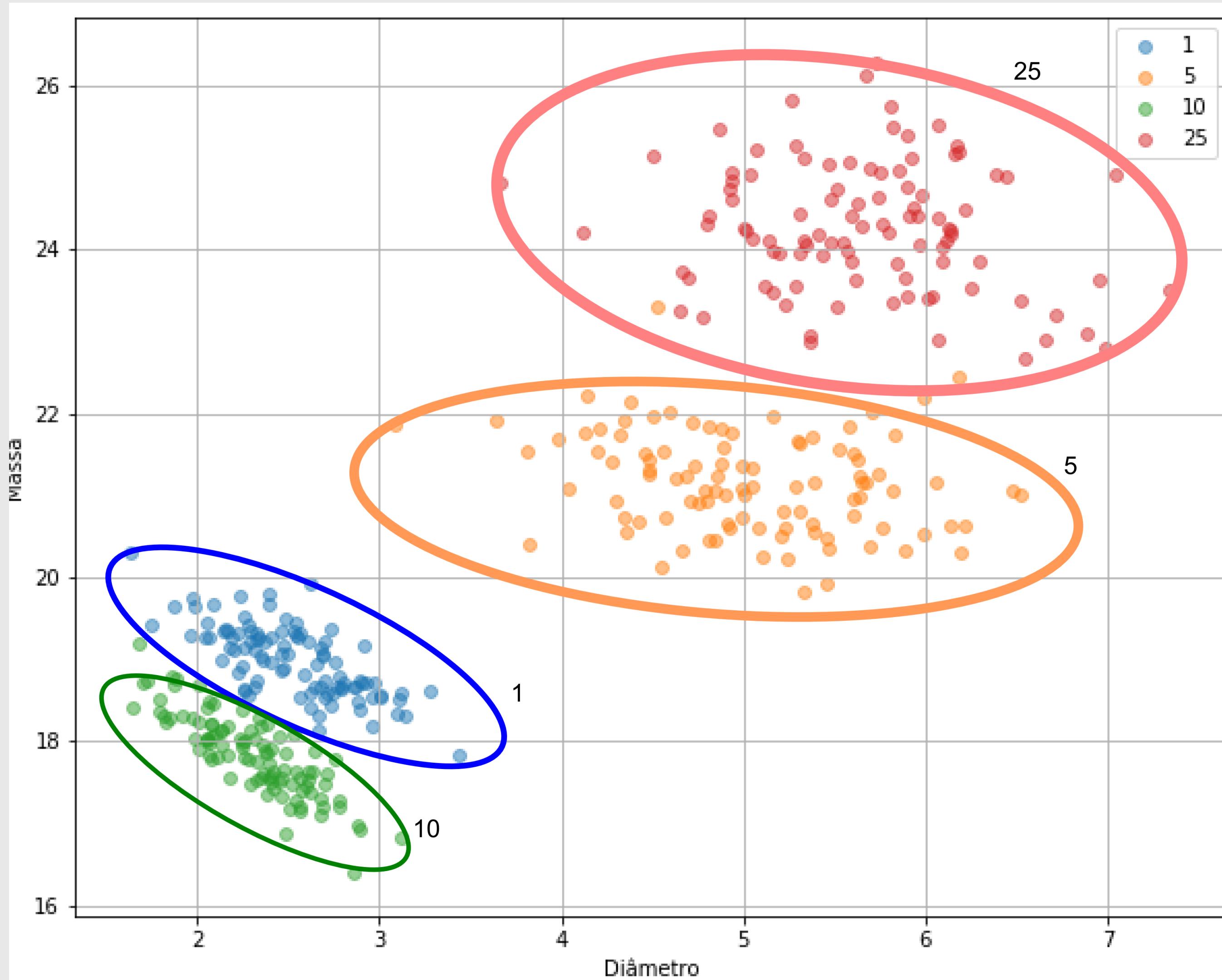
Exemplo:

- **Análise de clusters naturais:**
 - Distribuição de massa e diâmetro de moedas norte-americanas



Exemplo:

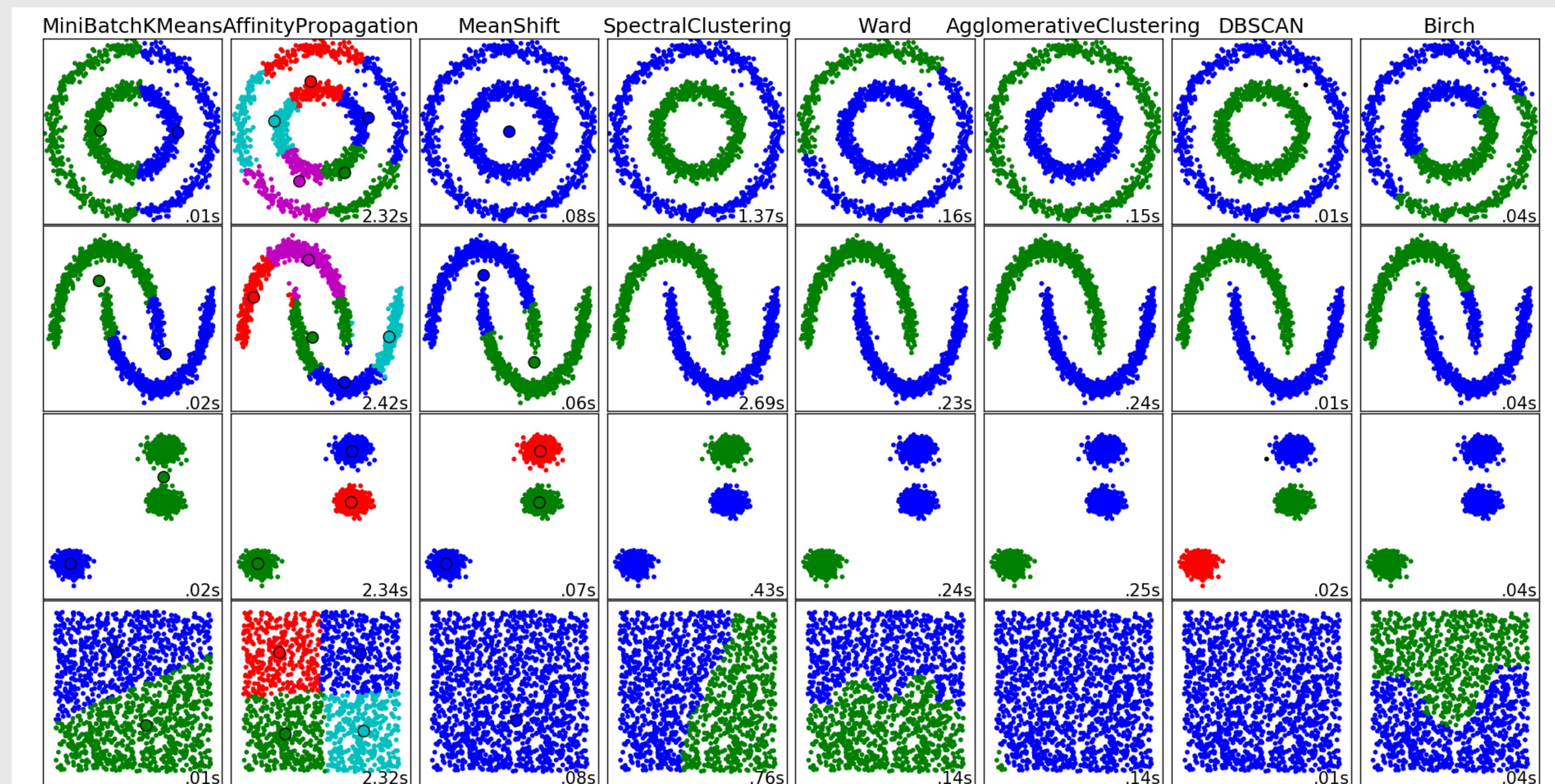
- Análise de clusters naturais:
 - Distribuição de massa e diâmetro de moedas norte-americanas



T

2. Clustering:

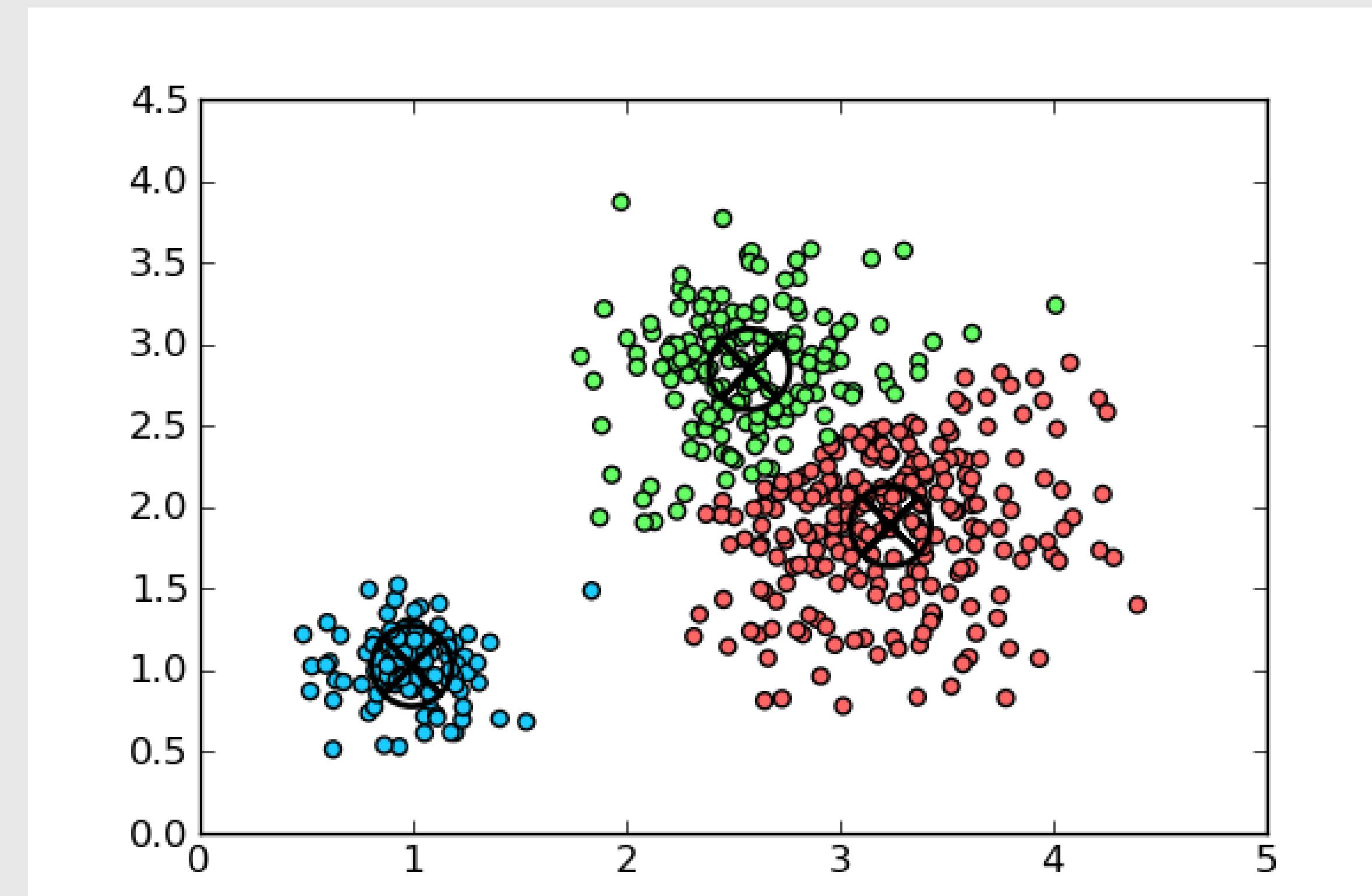
- Métodos populares:
 - K-Means
 - Hierarchical Clustering



T

2. Clustering: K-Means

- Gera clusters iterativamente
- Precisa escolher o número de clusters
- Agrupa dados pela distância do centroide



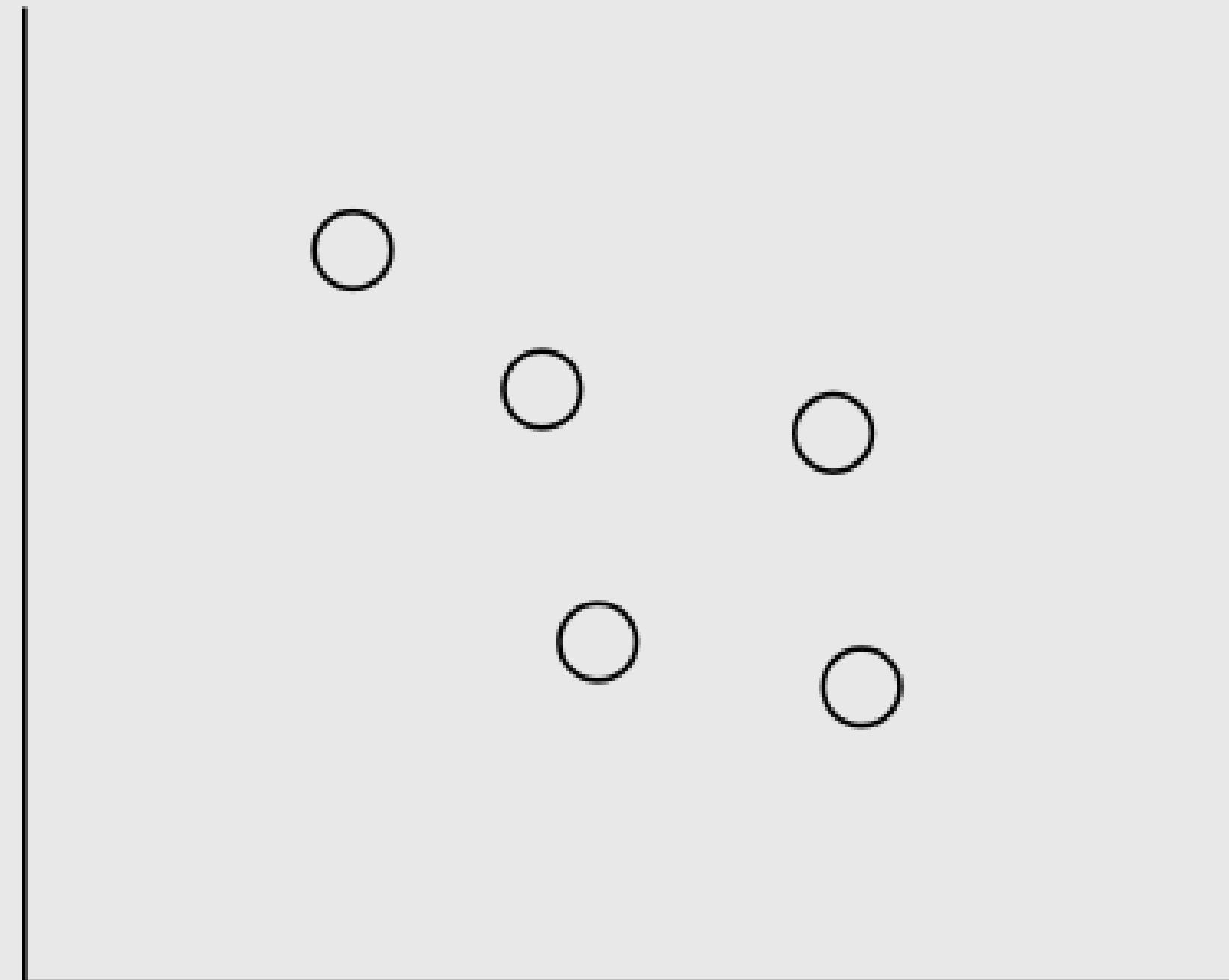
2. K-Means

- Algoritmo:
 - **Passo 1:** Defina o número de clusters
 - **Passo 2:** Atribua aleatoriamente um cluster para cada ponto
 - **Passo 3:** Calcule o centroide de cada cluster
 - **Passo 4:** Calcule a distância entre cada ponto e os centroides dos clusters
 - **Passo 5:** Atribua cada ponto a seu cluster cujo centroide está mais próximo
 - **Passo 6:** Repita os passos 3-5 até não haver mais variação

T

2. K-Means

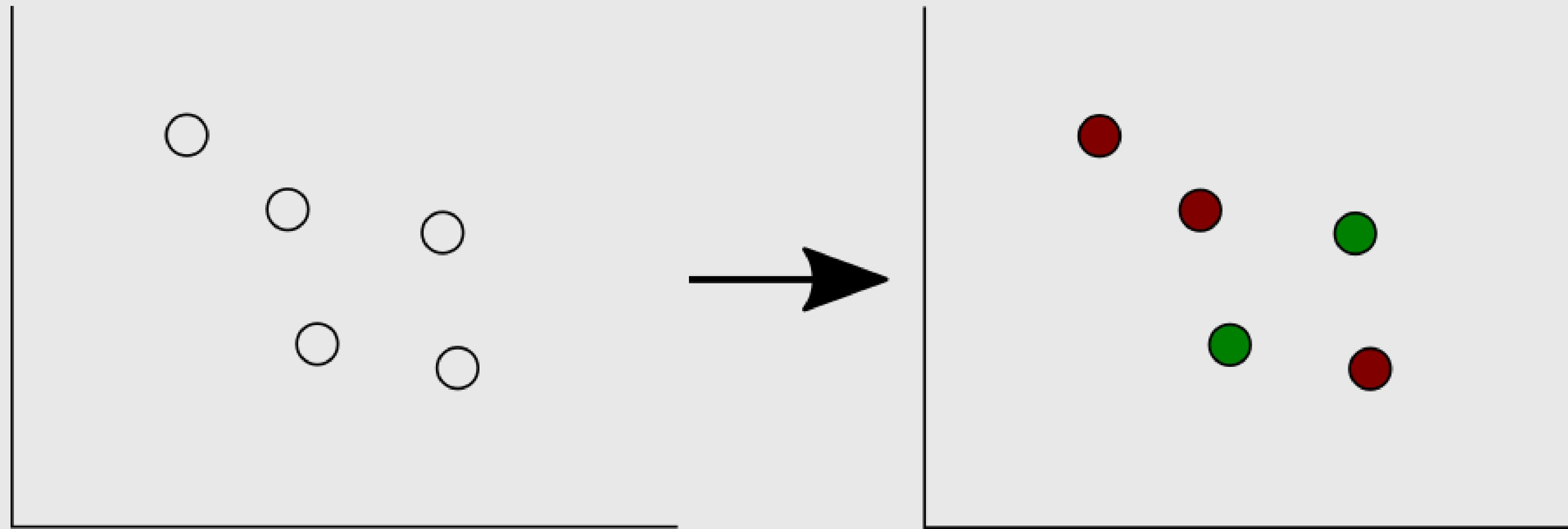
- **Passo 1:** Defina o número de clusters (k)
 - $K = 2$ (Arbitrário, por enquanto)



T

2. K-Means

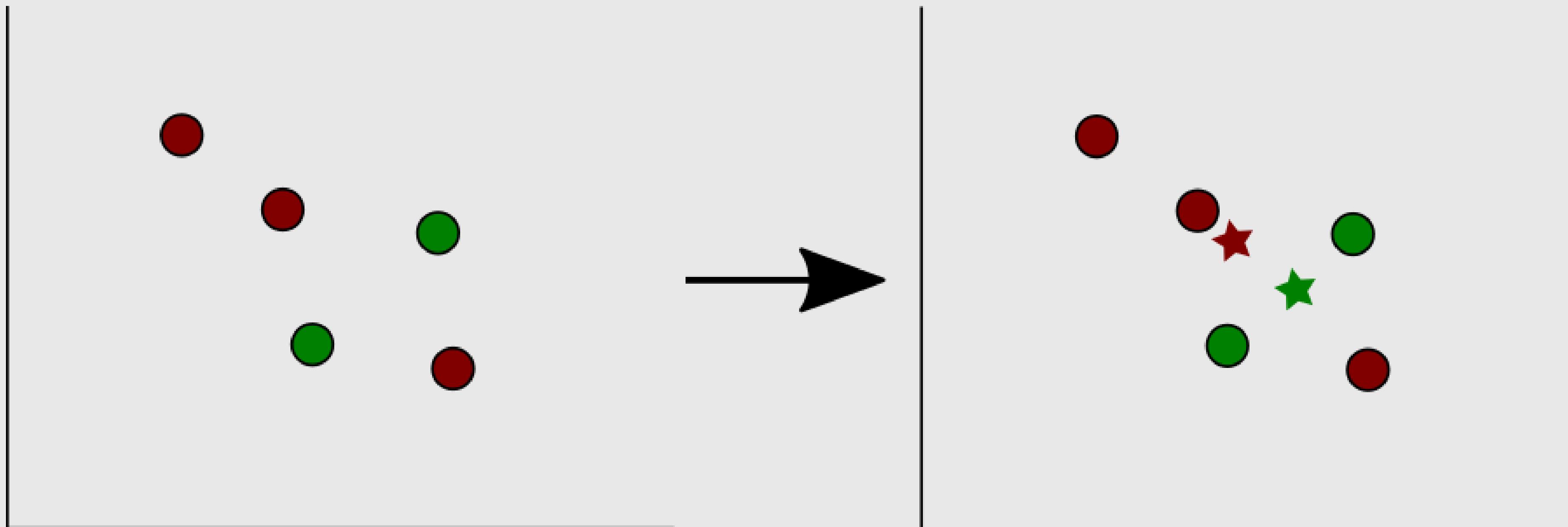
- **Passo 2:** Atribua aleatoriamente um cluster para cada ponto



T

2. K-Means

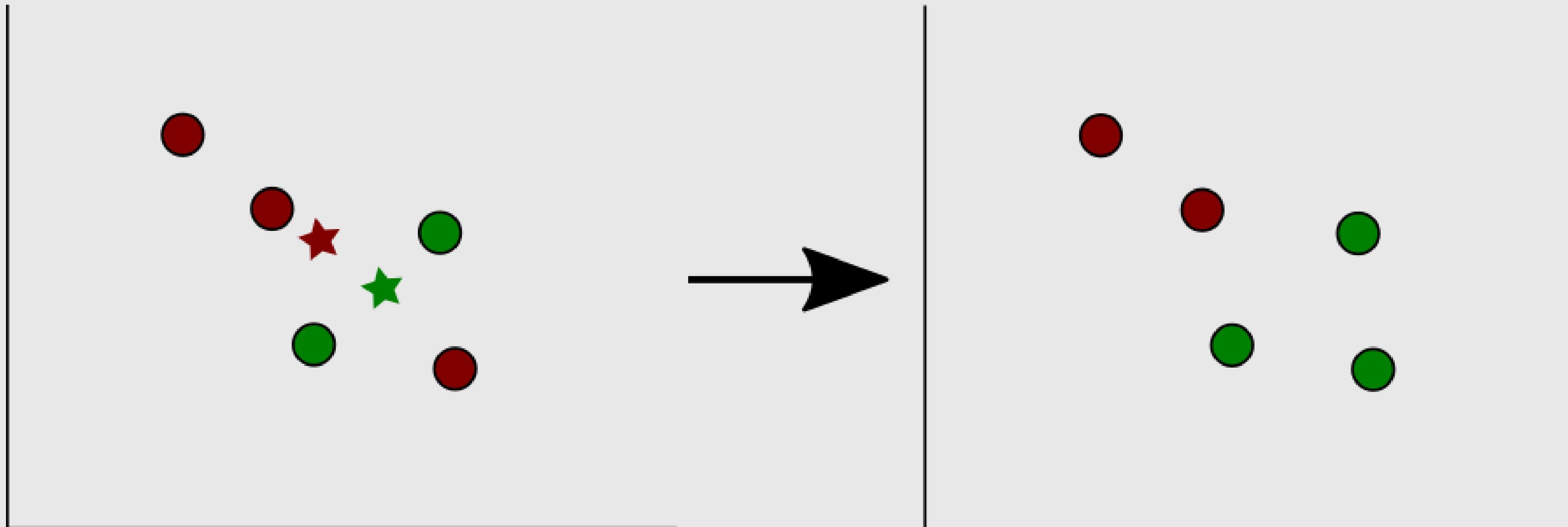
- **Passo 3:** Calcule o centroide de cada cluster



T

2. K-Means

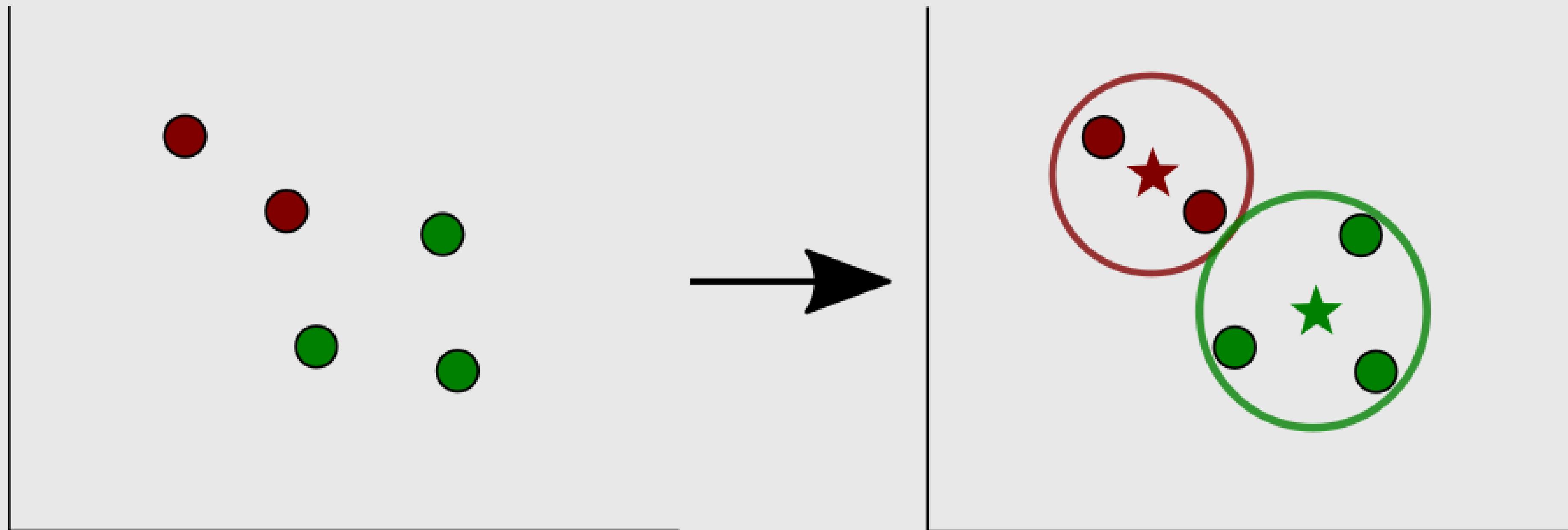
- **Passo 5:** Atribua cada ponto a seu cluster mais próximo



T

2. K-Means

- **Passo 6:** Repita os passos 3-5 até não haver mais variação



T

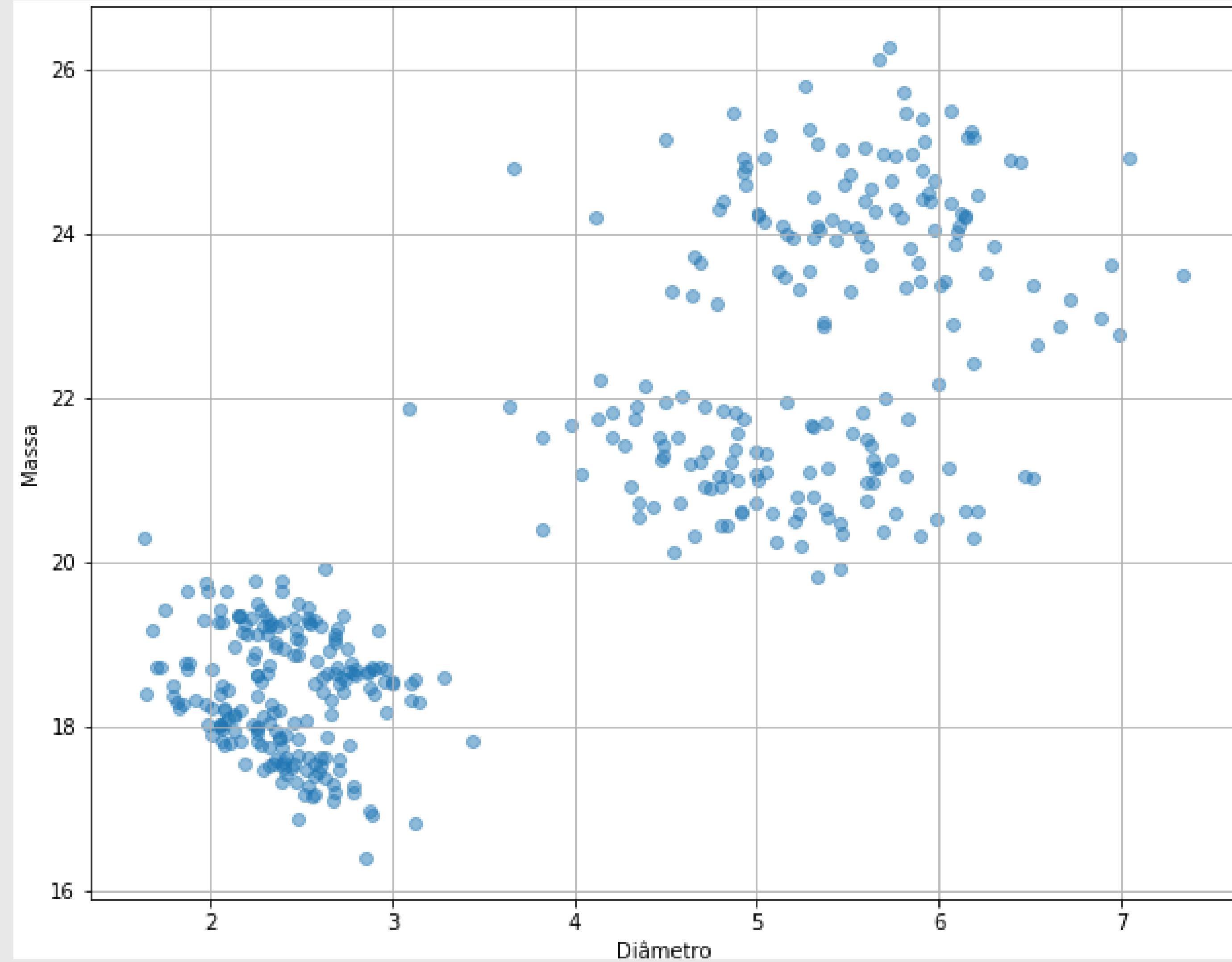
2. K-Means

- **Exemplo:** <https://www.youtube.com/watch?v=5I3Ei69I40s>

I

2. K-Means

- **Exemplo aplicado:**
 - Cluster de moedas



2. K-Means

- **Avaliar um bom cluster:**

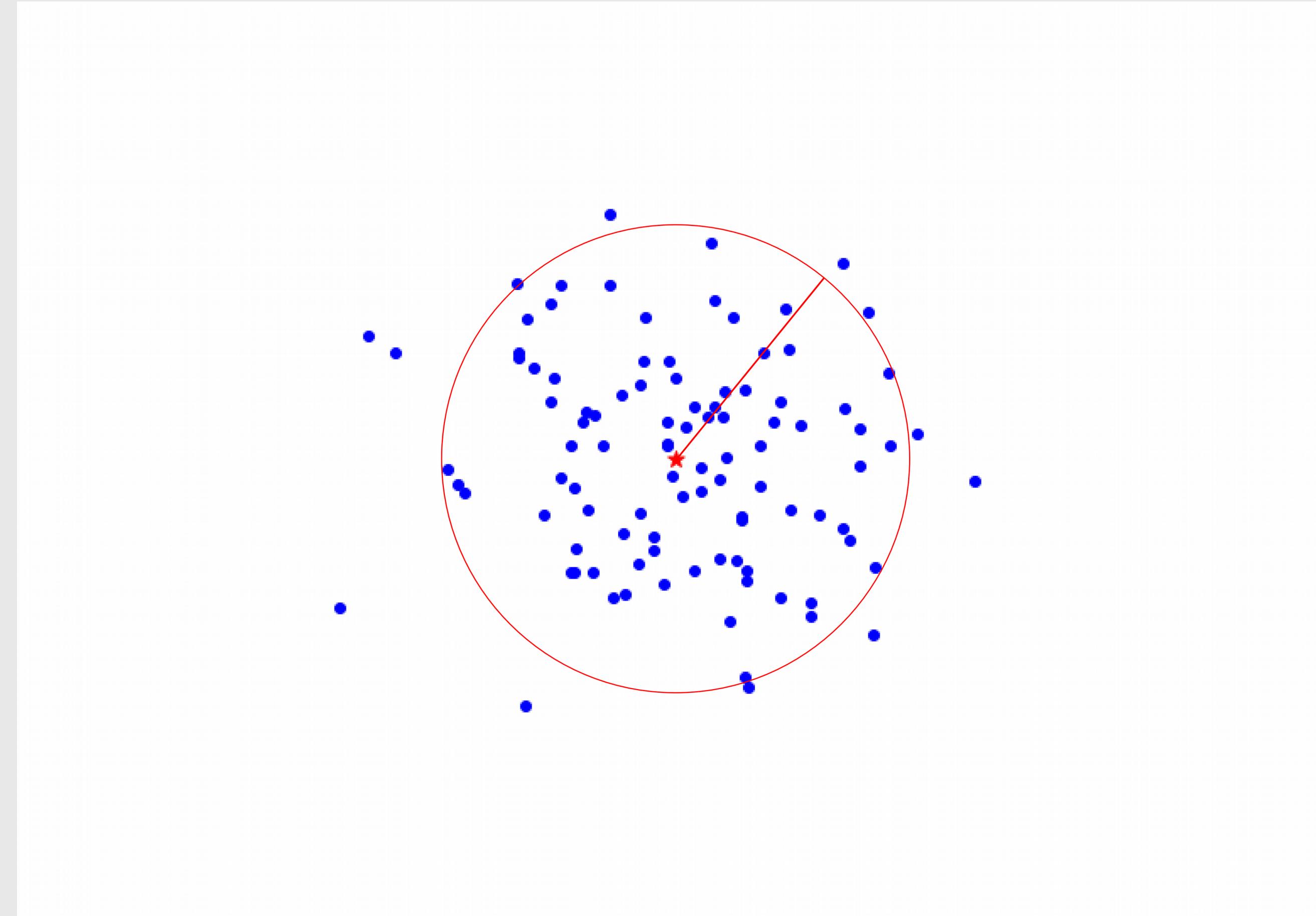
- Dados dentro do cluster possuem perfil semelhante
- Dados bem distribuídos nos clusters (balanceados)
- Não muito disperso



Inércia

2. K-Means

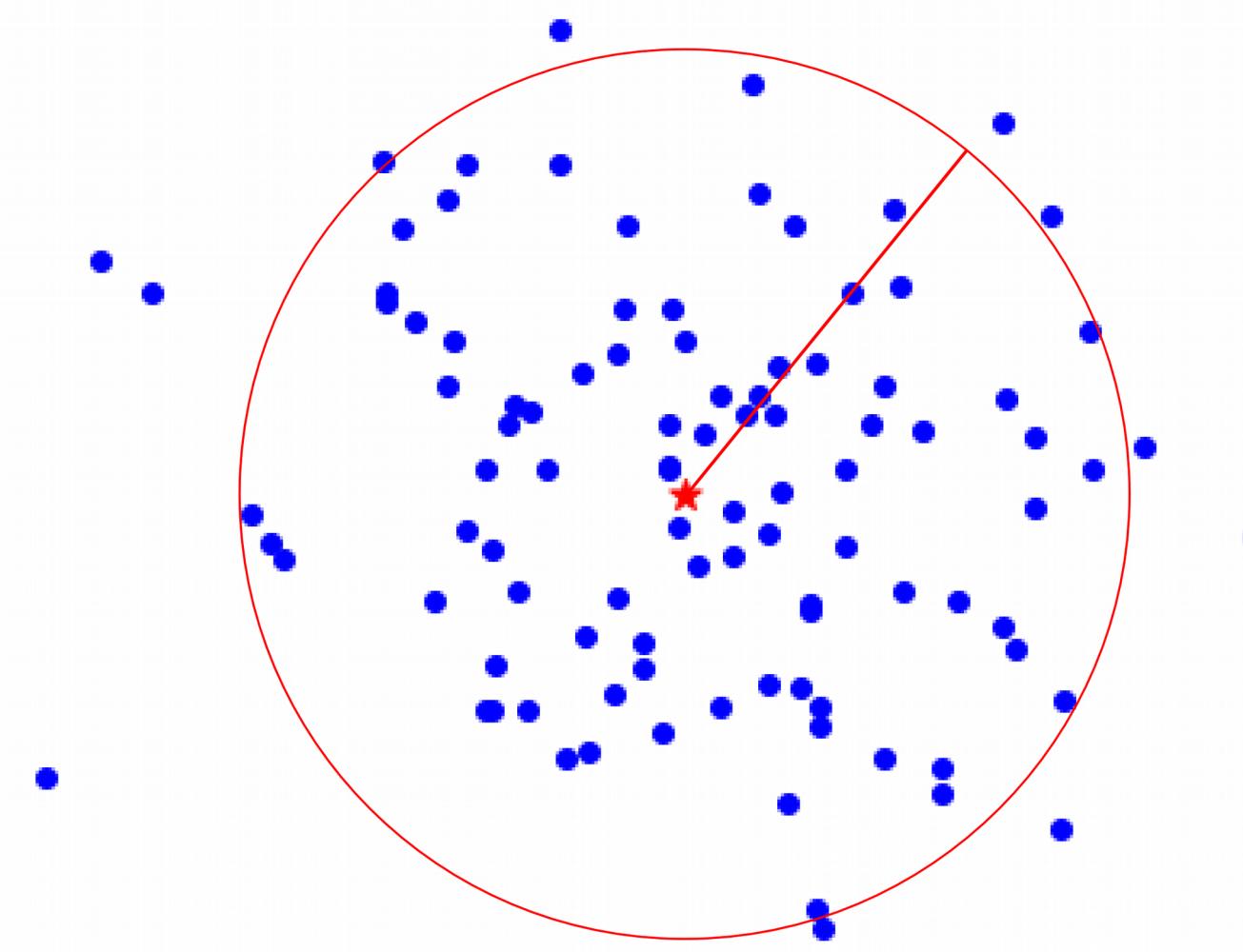
- **Inércia:**
 - Nível de dispersão dos pontos em relação ao centroide



I

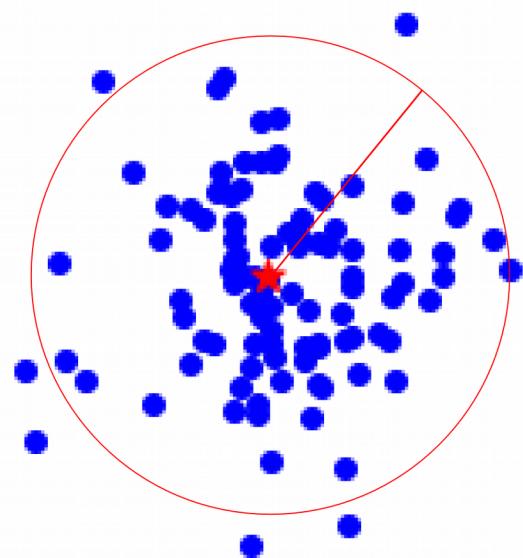
2. K-Means

- Inércia:
 - Elevada



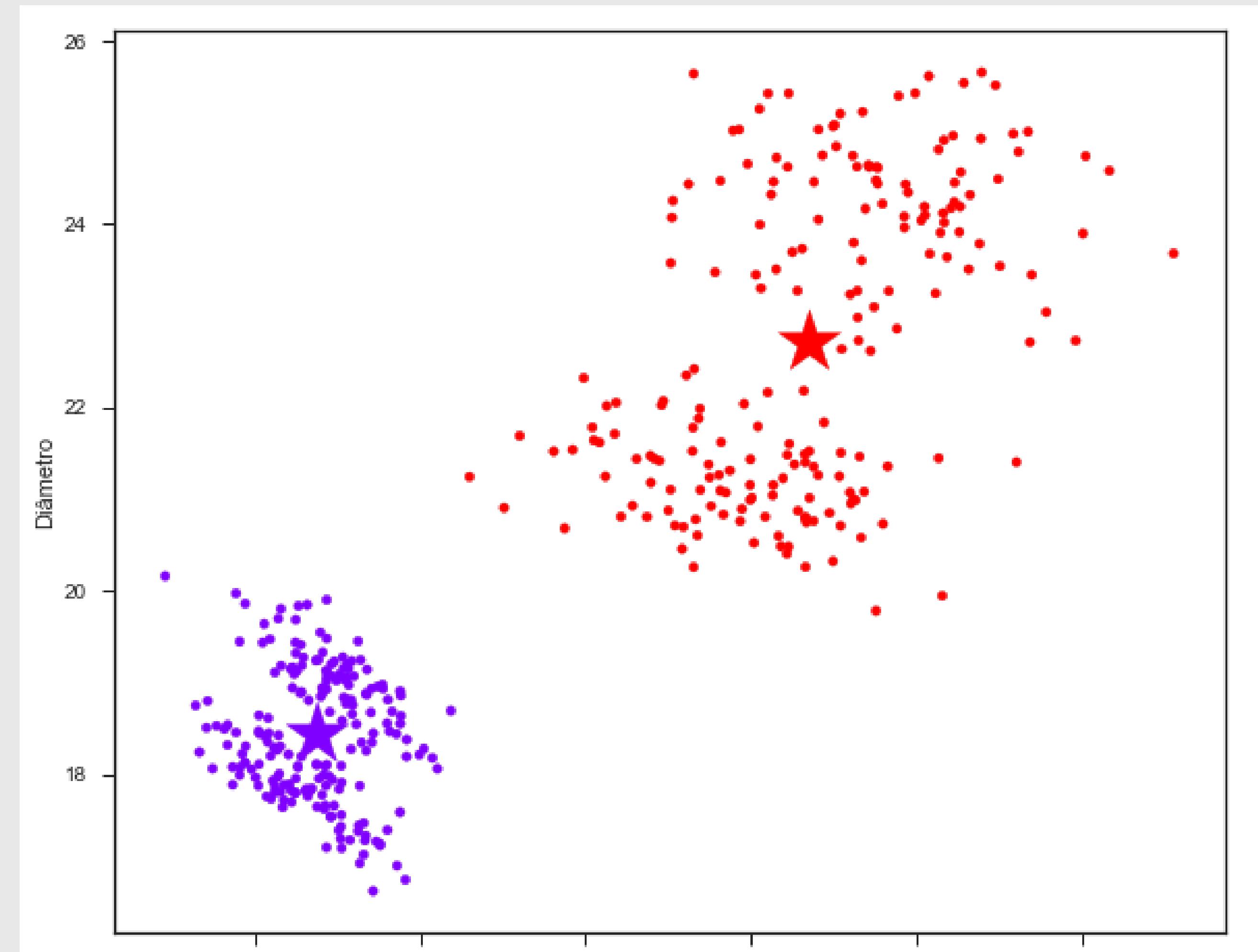
2. K-Means

- Inércia:
 - Baixa



2. K-Means: Inércia

- Tem relação com o número de clusters
- Ex: $k=2$
- Inércia alta

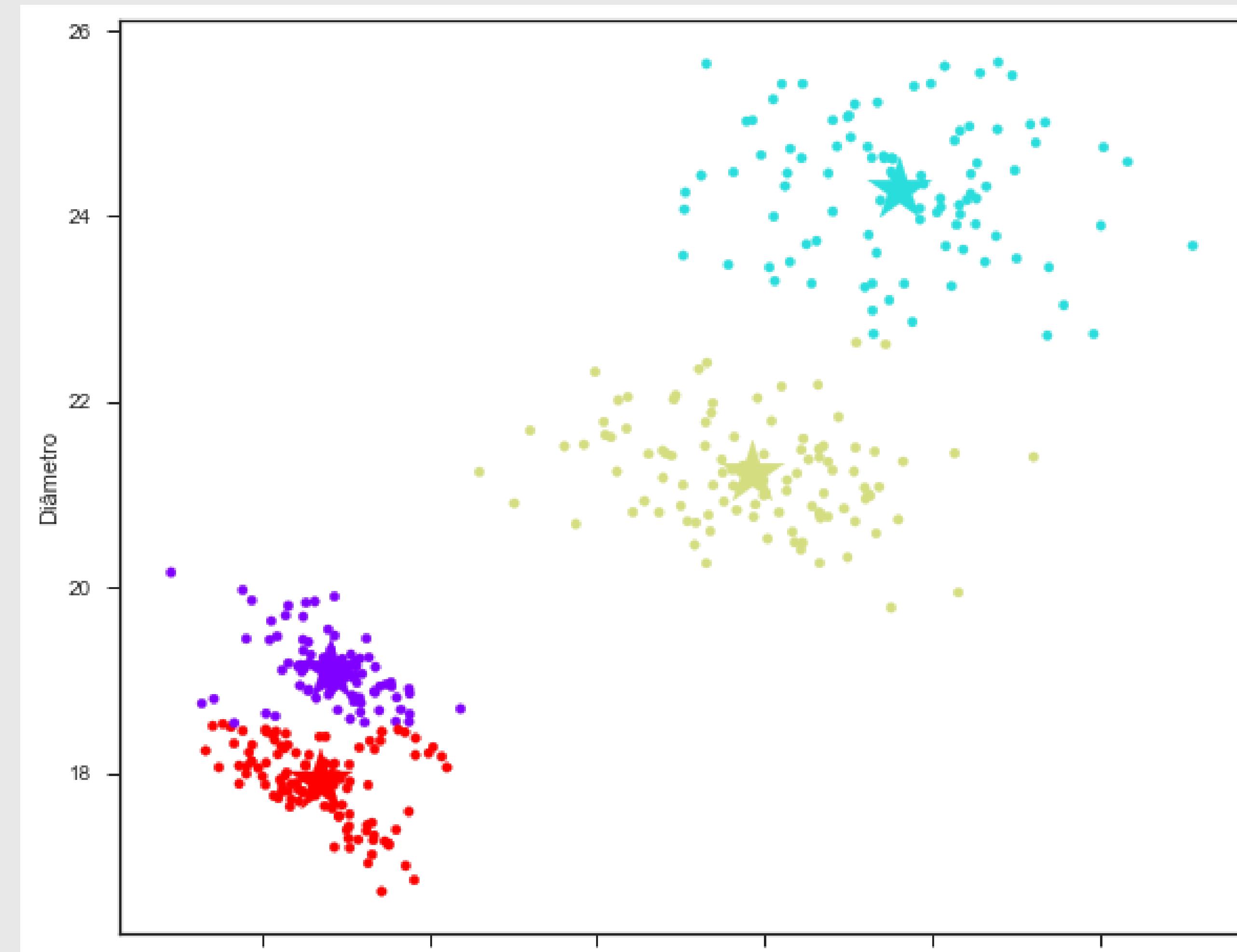


Inércia = 786

T

2. K-Means: Inércia

- Tem relação com o número de clusters
- Ex: $k=4$



Inércia = 211

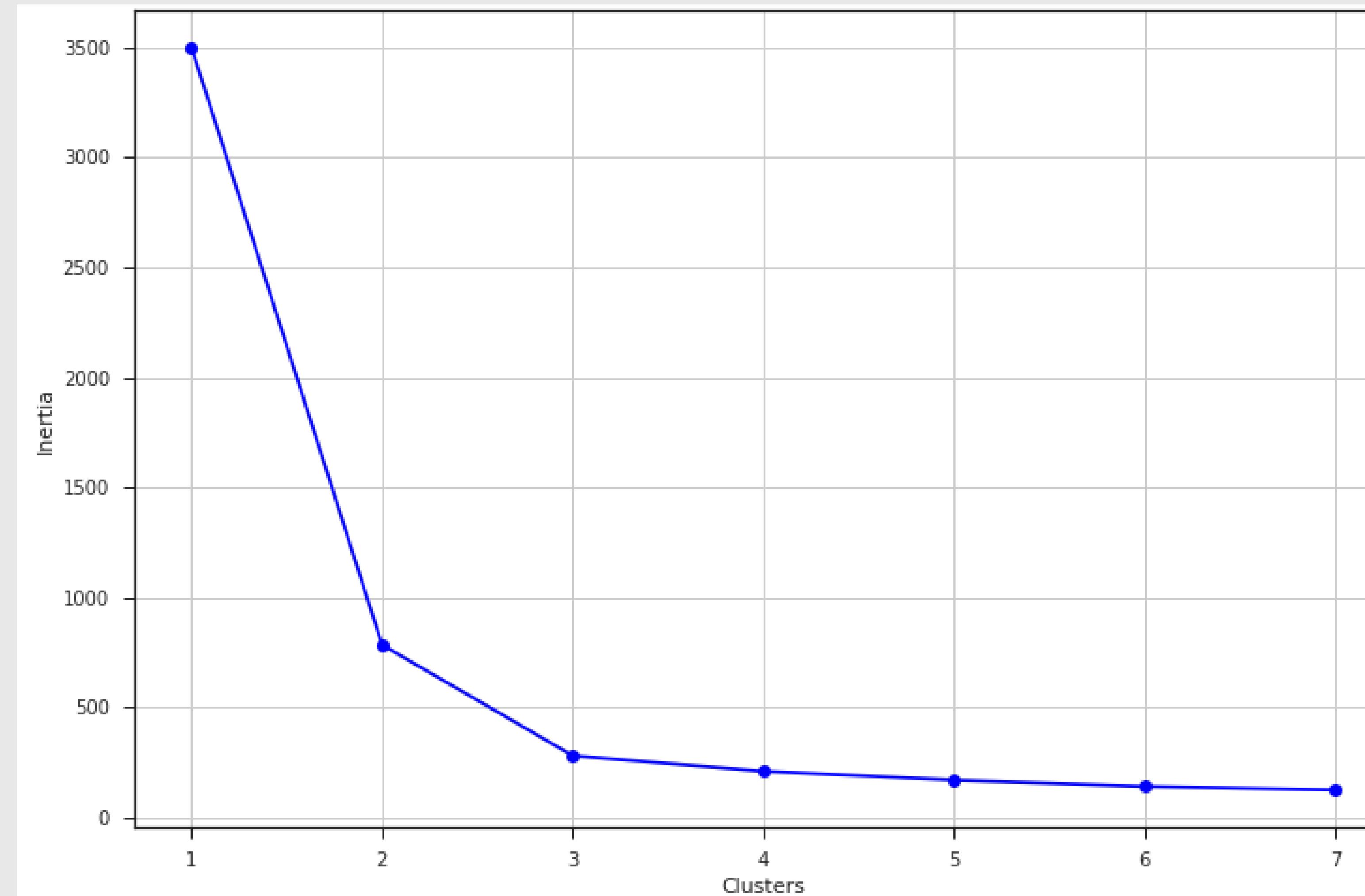
T

2. K-Means: Inércia

- Exemplo: Notebook

2. K-Means: Inércia

- Tem relação com o número de clusters



2. K-Means

- **Vantagens K-Means**
 - Algoritmo simples
 - Resultado intuitivo
 - Funciona bem na prática
 - Pode ser utilizado para conjunto grande de dados (tempo de execução cresce linearmente com tamanho do dataset)

2. K-Means

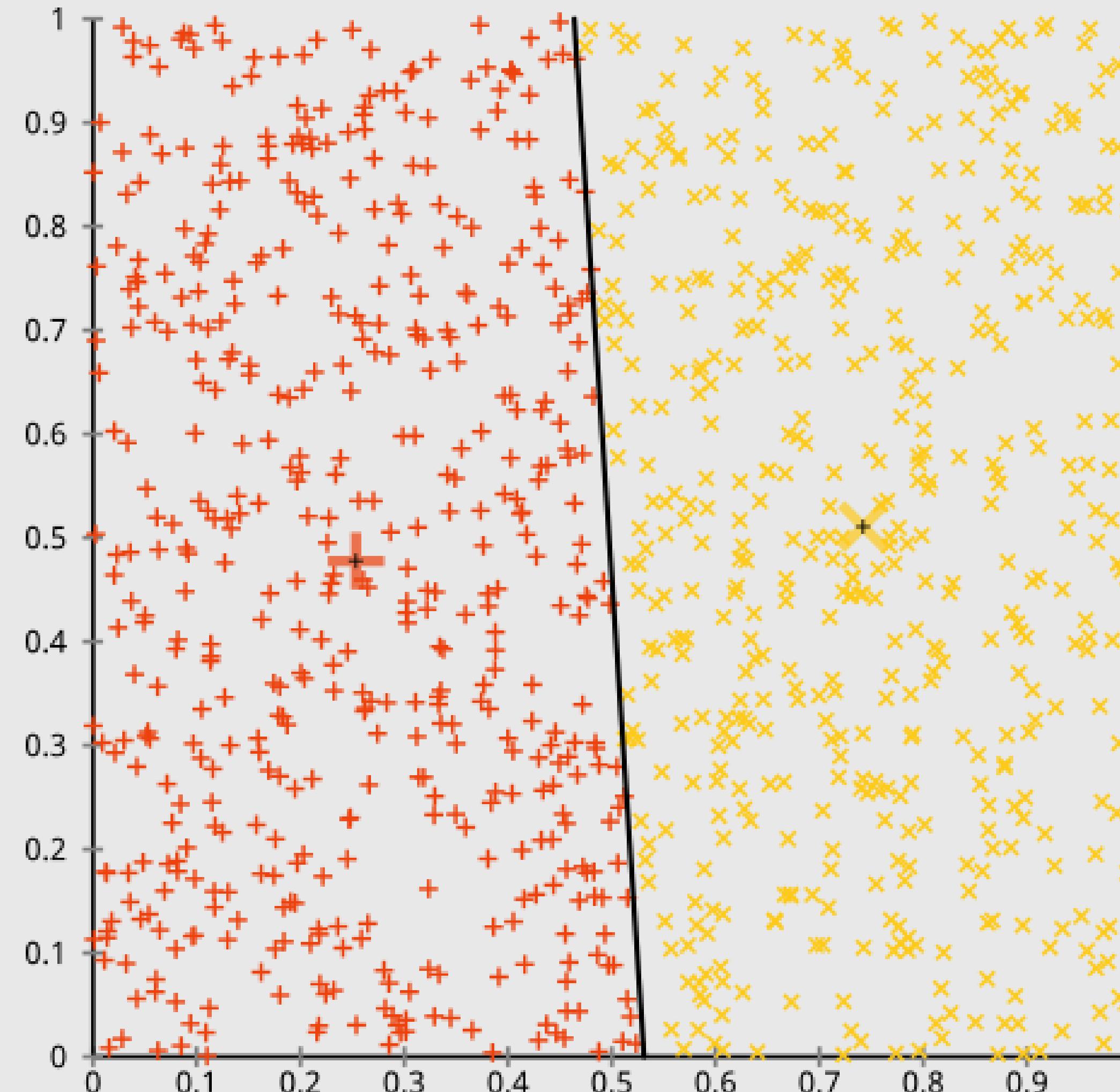
- **Desvantagens K-Means**

- Necessita escolher o número de clusters
- Pode convergir para resultados indesejados
- Resultados podem divergir a cada repetição (inicialização aleatória)
- Algoritmo “cego”: Encontra clusters até em locais onde não há

I

2. K-Means

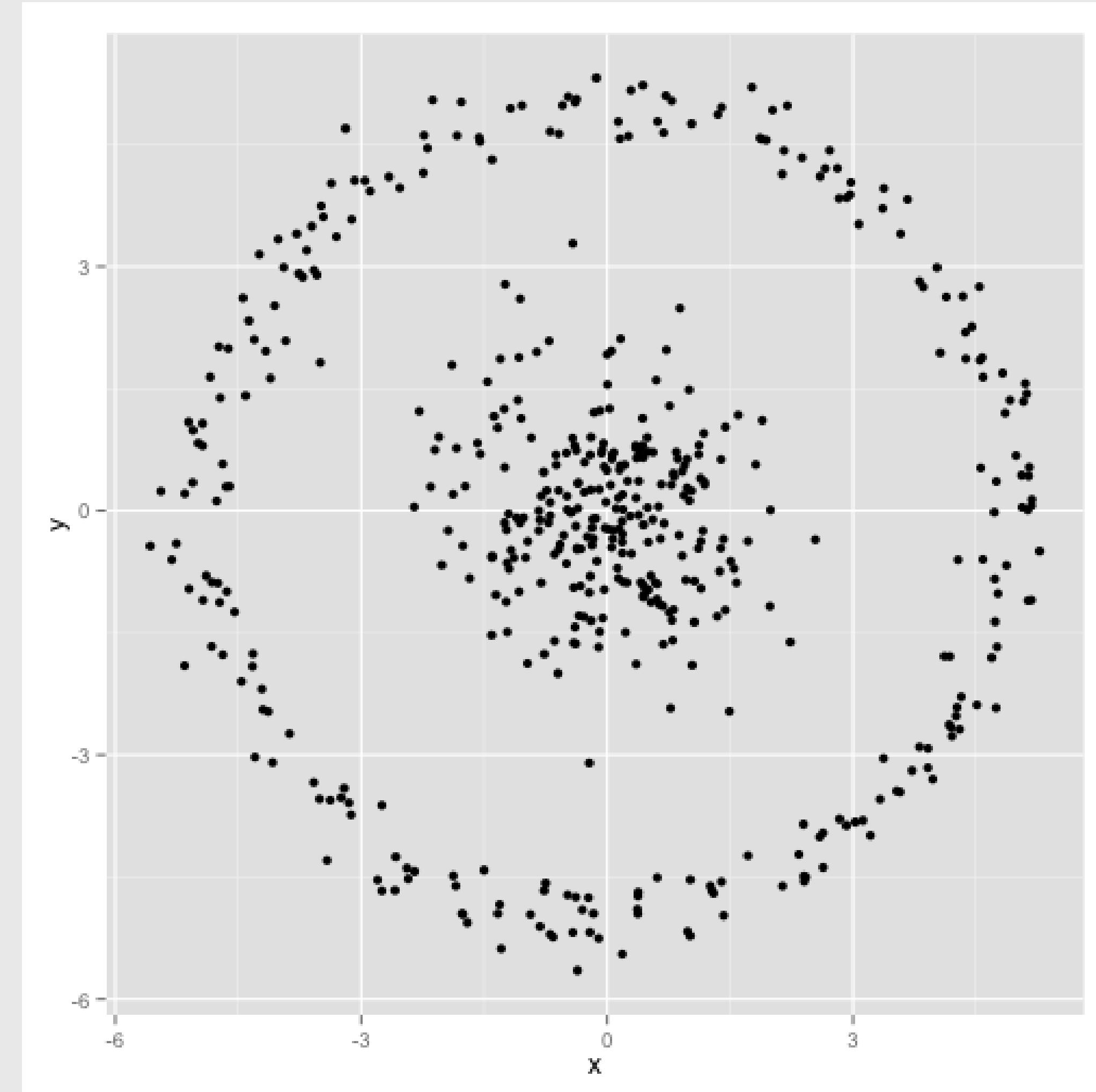
- Desvantagens K-Means



I

2. K-Means

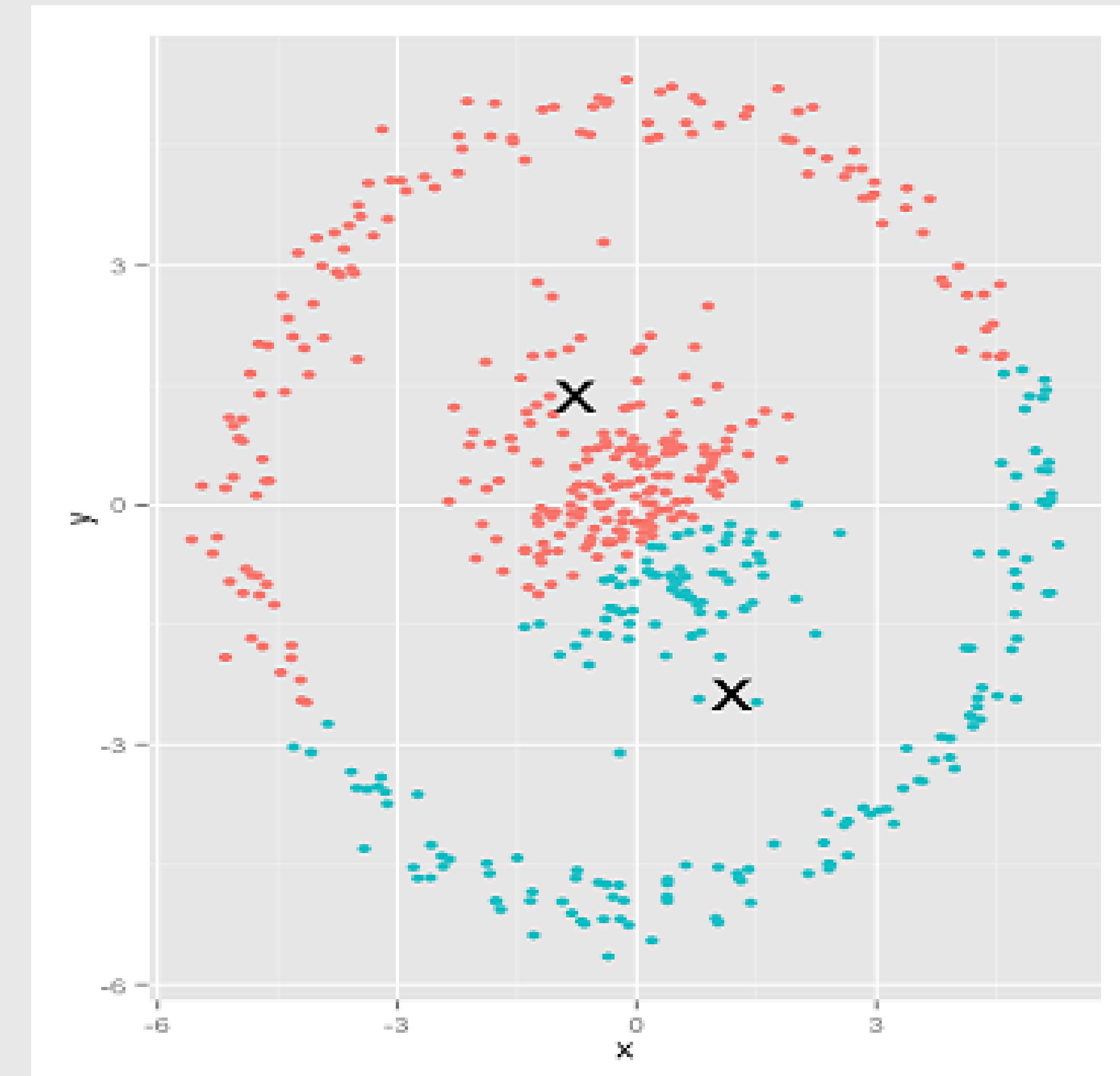
- Desvantagens K-Means



I

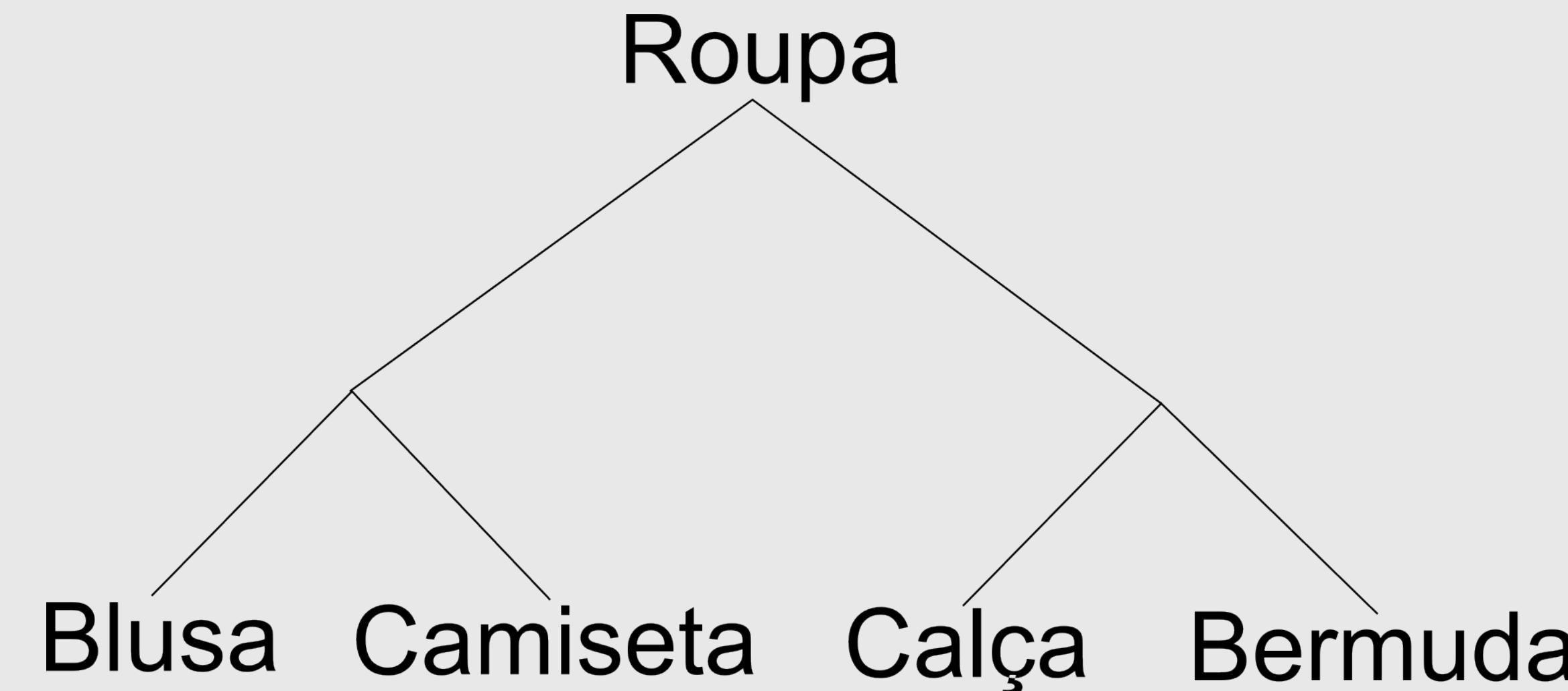
2. K-Means

- Desvantagens K-Means



3. Clustering: Hierarchical Clustering

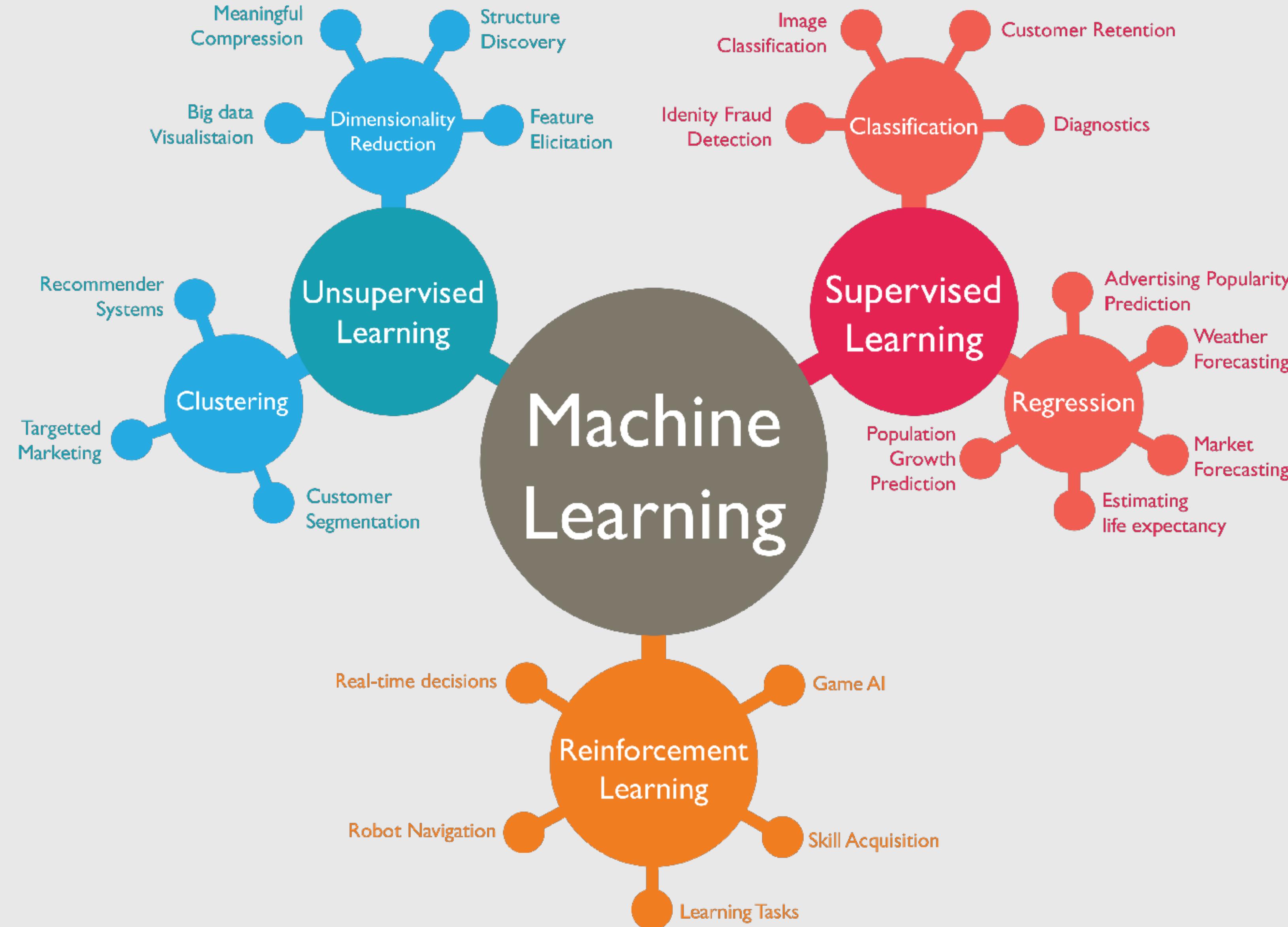
- Método iterativo
- Se baseia no princípio da conectividade das observações
- Depende da definição de distância / similaridade



T

3. Clustering: Hierarchical Clustering

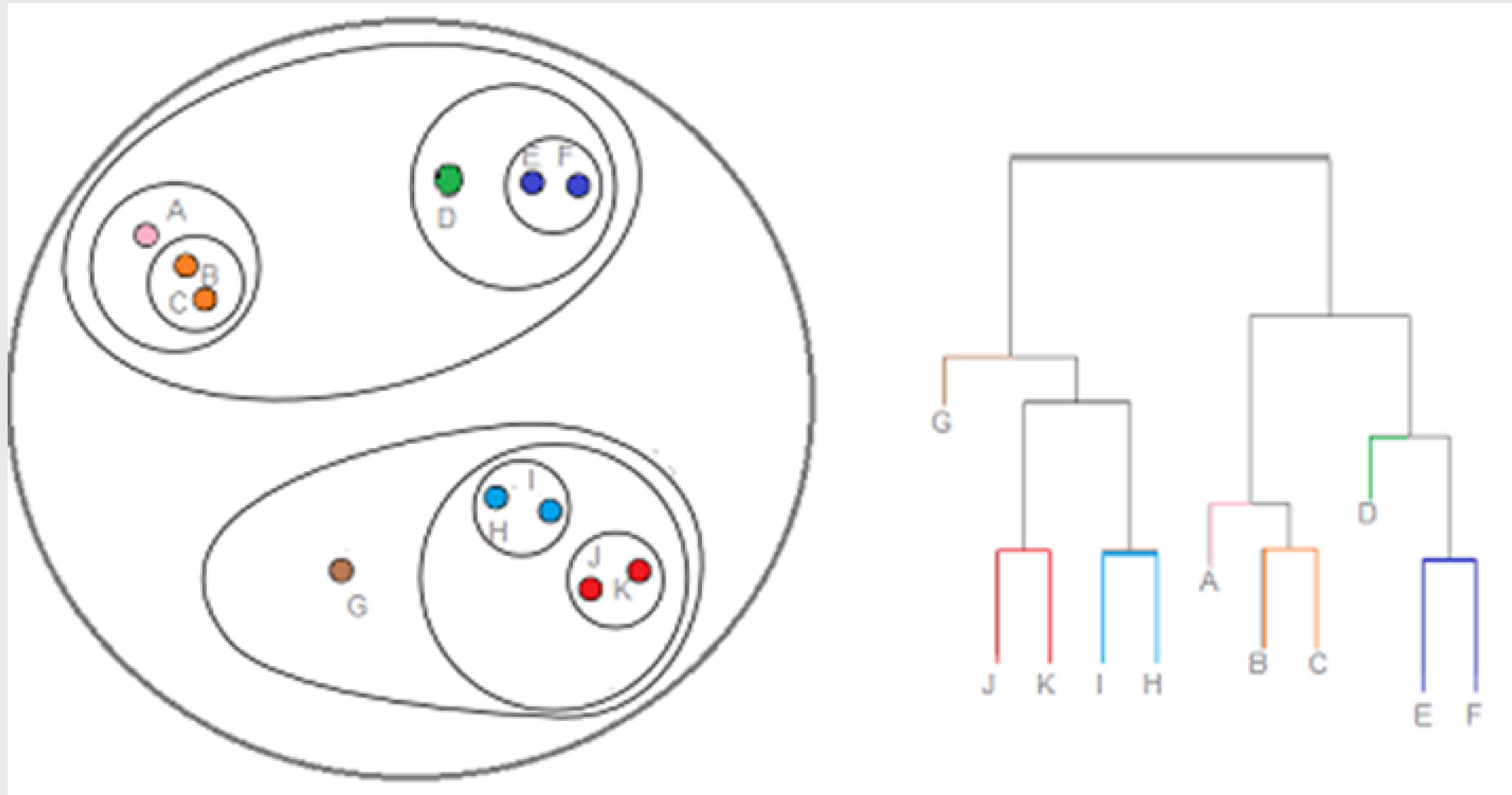
- Exemplo:



I

3. Clustering: Hierarchical Clustering

- Visualizar histórico de divisão: **Dendrograma**

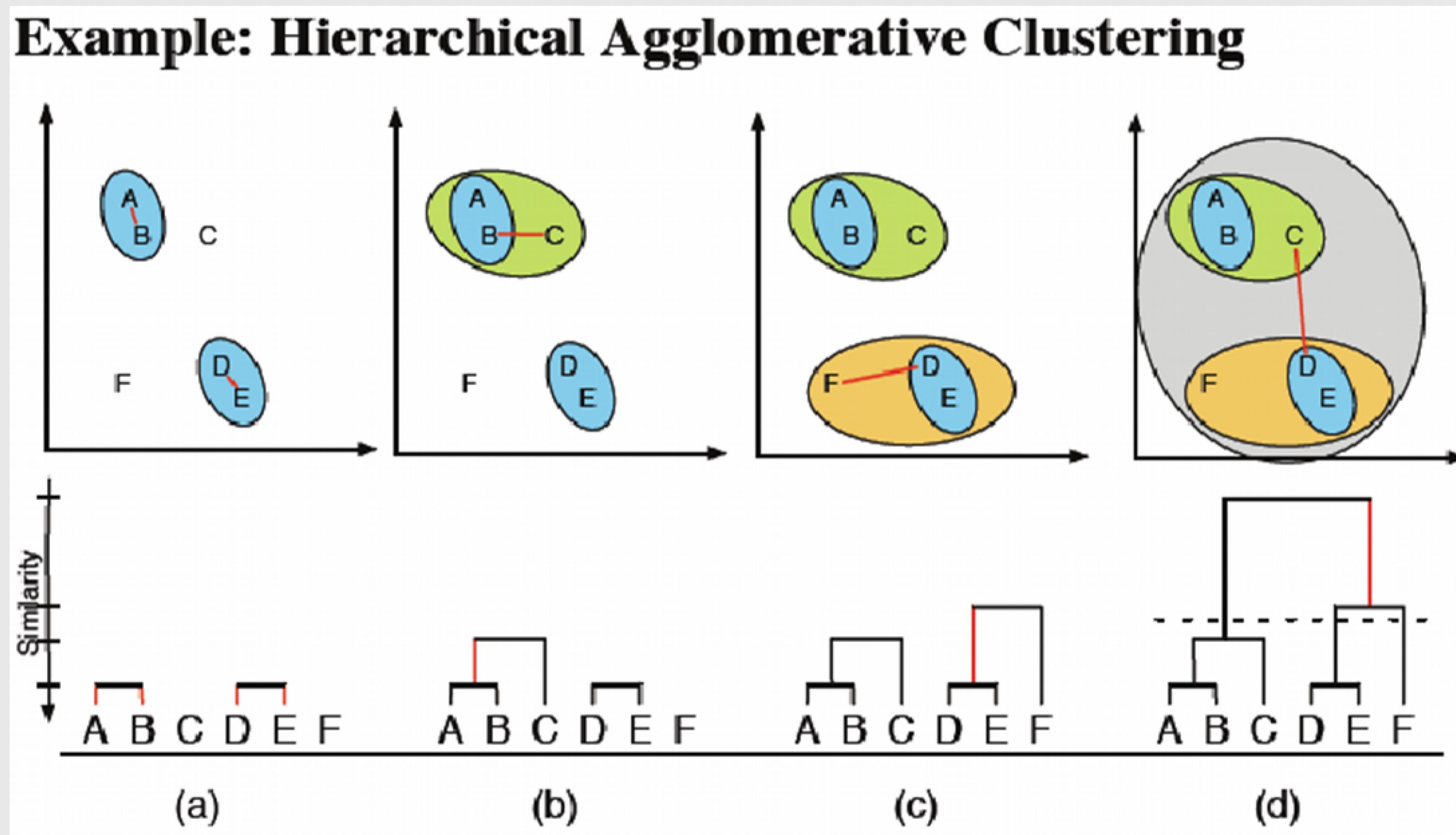


3. Clustering: Hierarchical Clustering

- 2 métodos principais:
 - Aglomerativo (Mais utilizado)
 - Por divisão

3. Clustering: Hierarchical Clustering

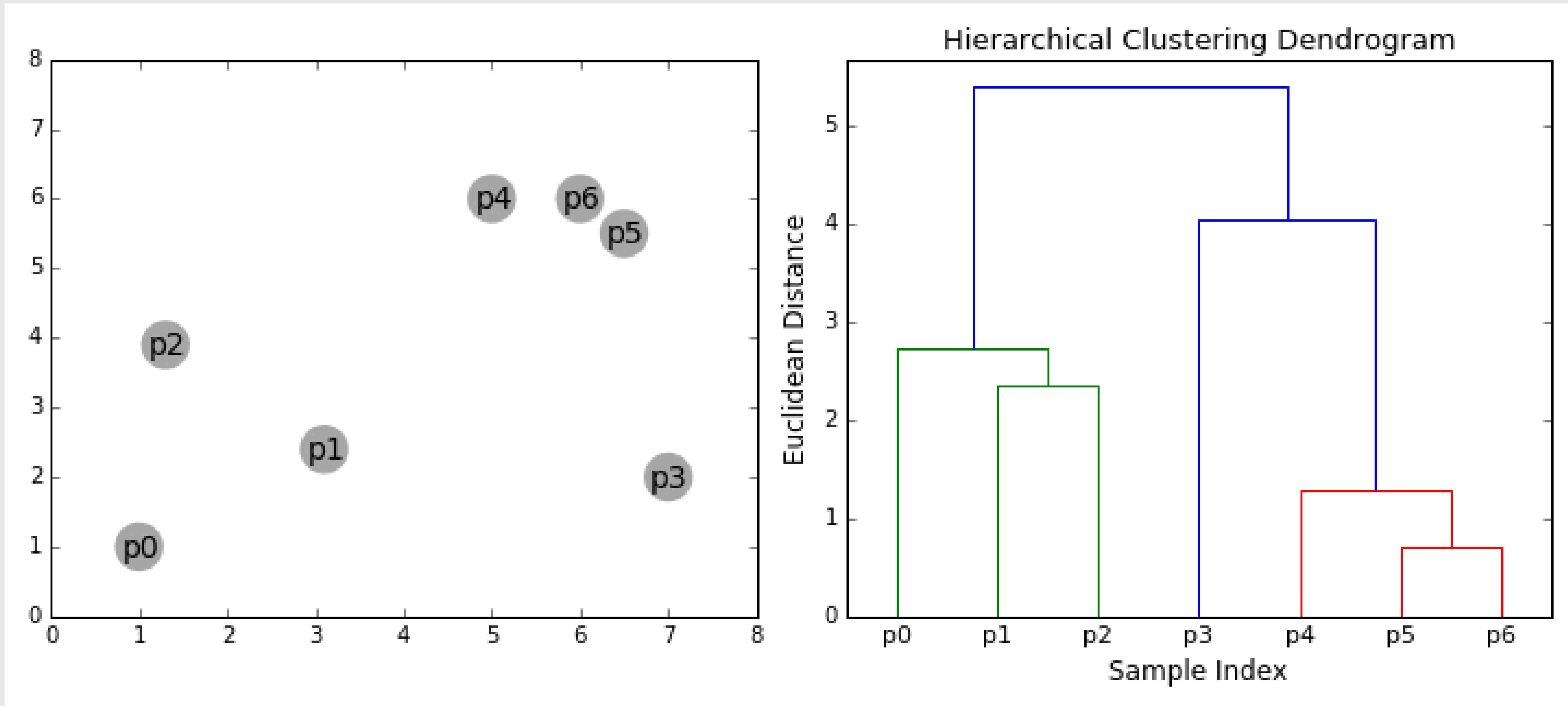
- Método aglomerativo:



I

3. Clustering: Hierarchical Clustering

- Método aglomerativo:



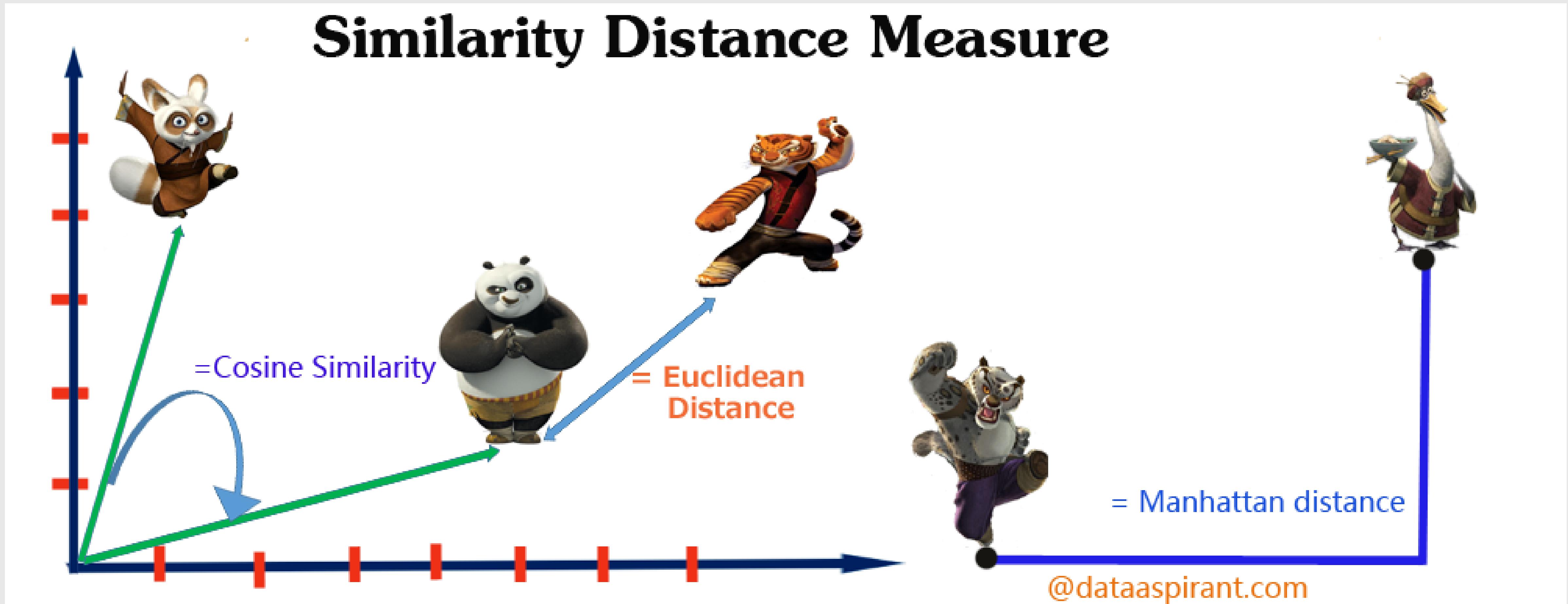
3. Clustering: Hierarchical Clustering

- Como agrupar clusters?
 - Distância (afinidade) entre clusters:
 - Euclidiana
 - Manhattan
 - Cosseno

I

3. Clustering: Hierarchical Clustering

- Tipos de distância:



3. Clustering: Hierarchical Clustering

- Tipos de distância:

- Euclidean Distance → $d(p, q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$
- Manhattan Distance → $d(p, q) = \sum_{i=1}^n |q_i - p_i|$
- Cosine Similarity → $similarity = \cos(\phi) = \frac{A \cdot B}{\|A\| \|B\|}$

3. Clustering: Hierarchical Clustering

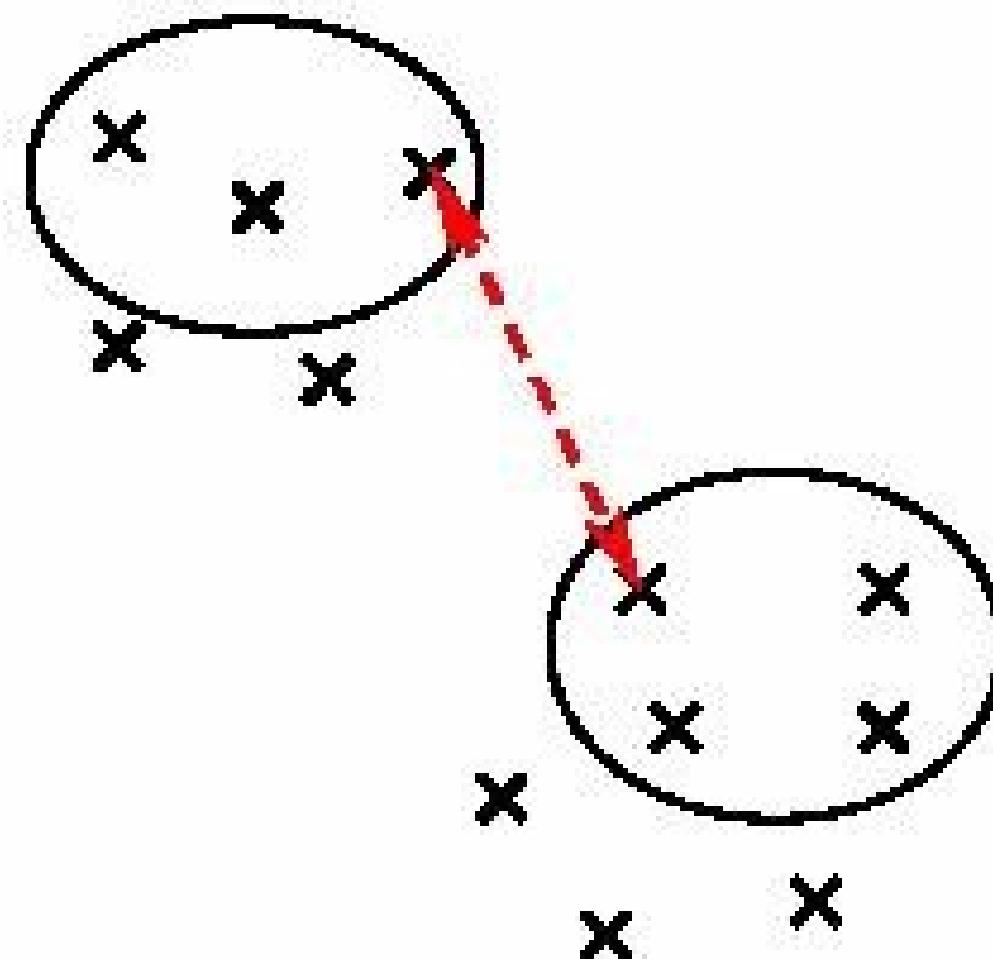
- Como agrupar clusters?
 - Tipos de ligação:
 - **Ward**: Mínima variância
 - **Completa**: Máxima distância
 - **Média**: Distância média
 - **Simples**: Menor distância

I

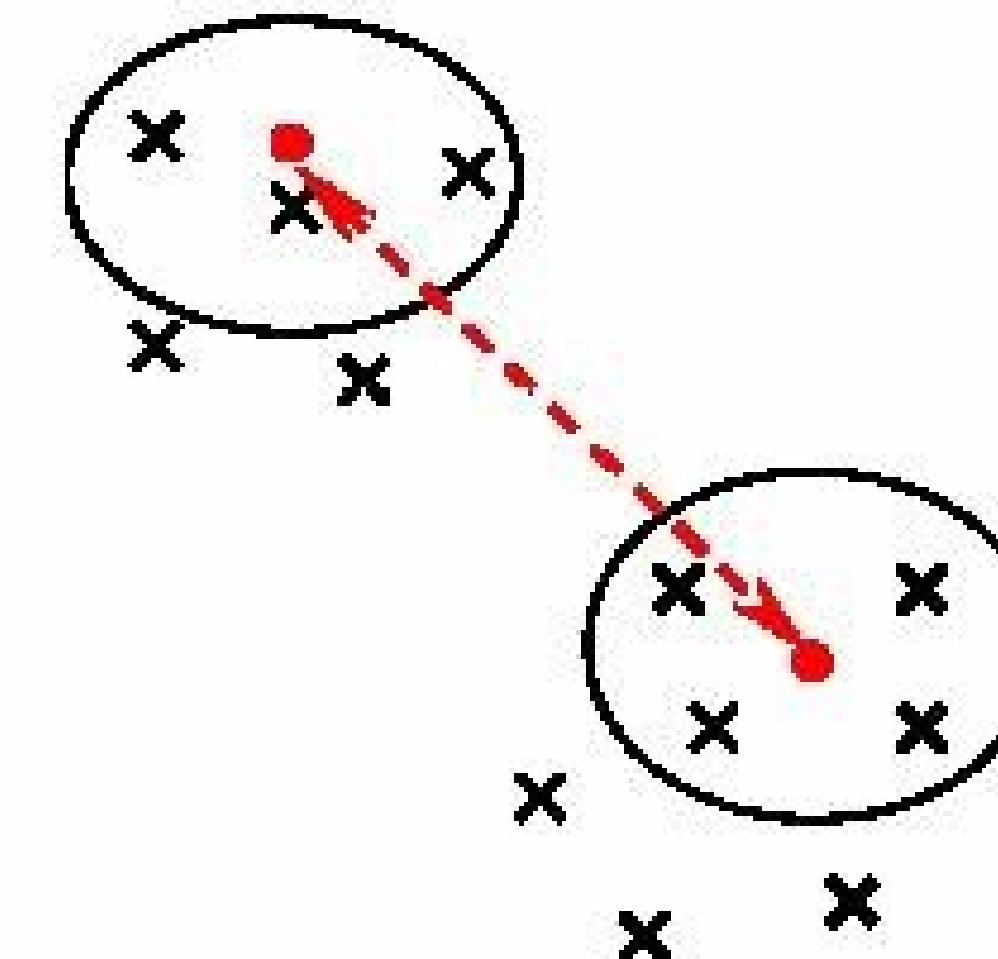
3. Clustering: Hierarchical Clustering

- Tipos de ligação:

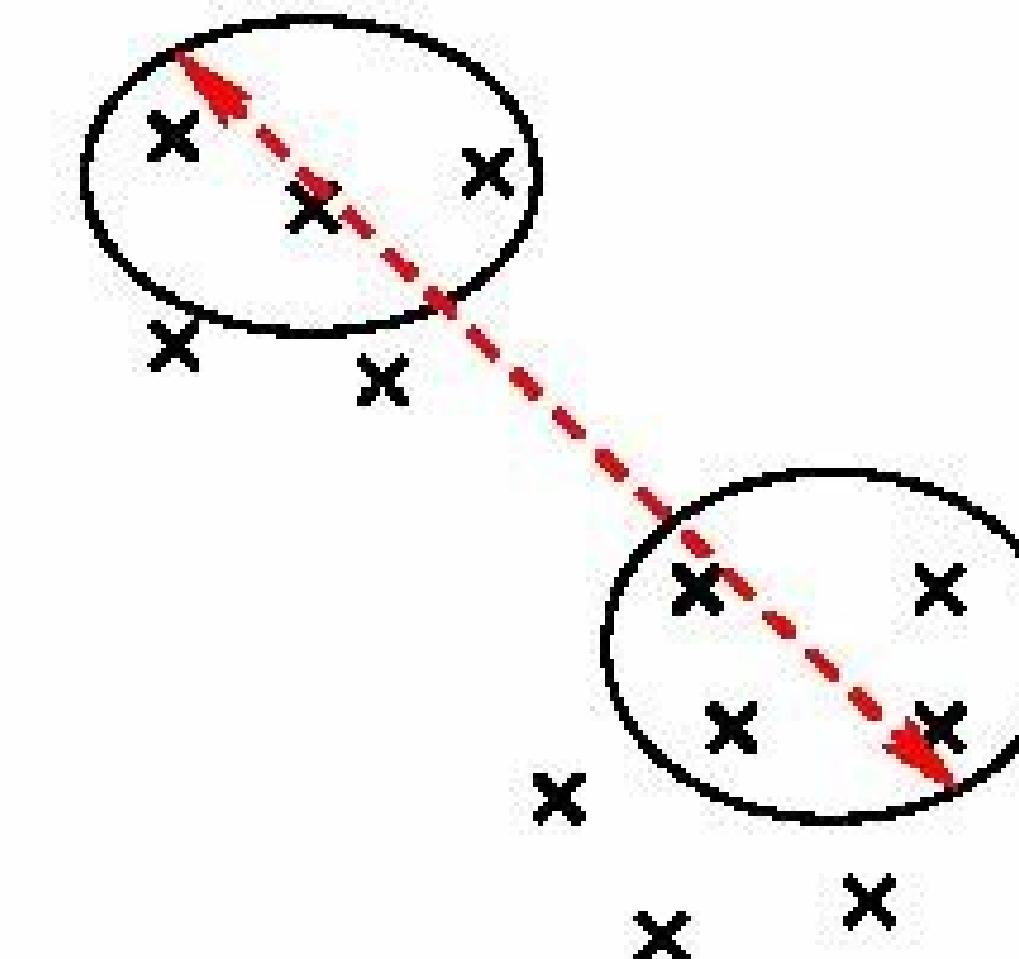
- Simple linkage



- Average linkage



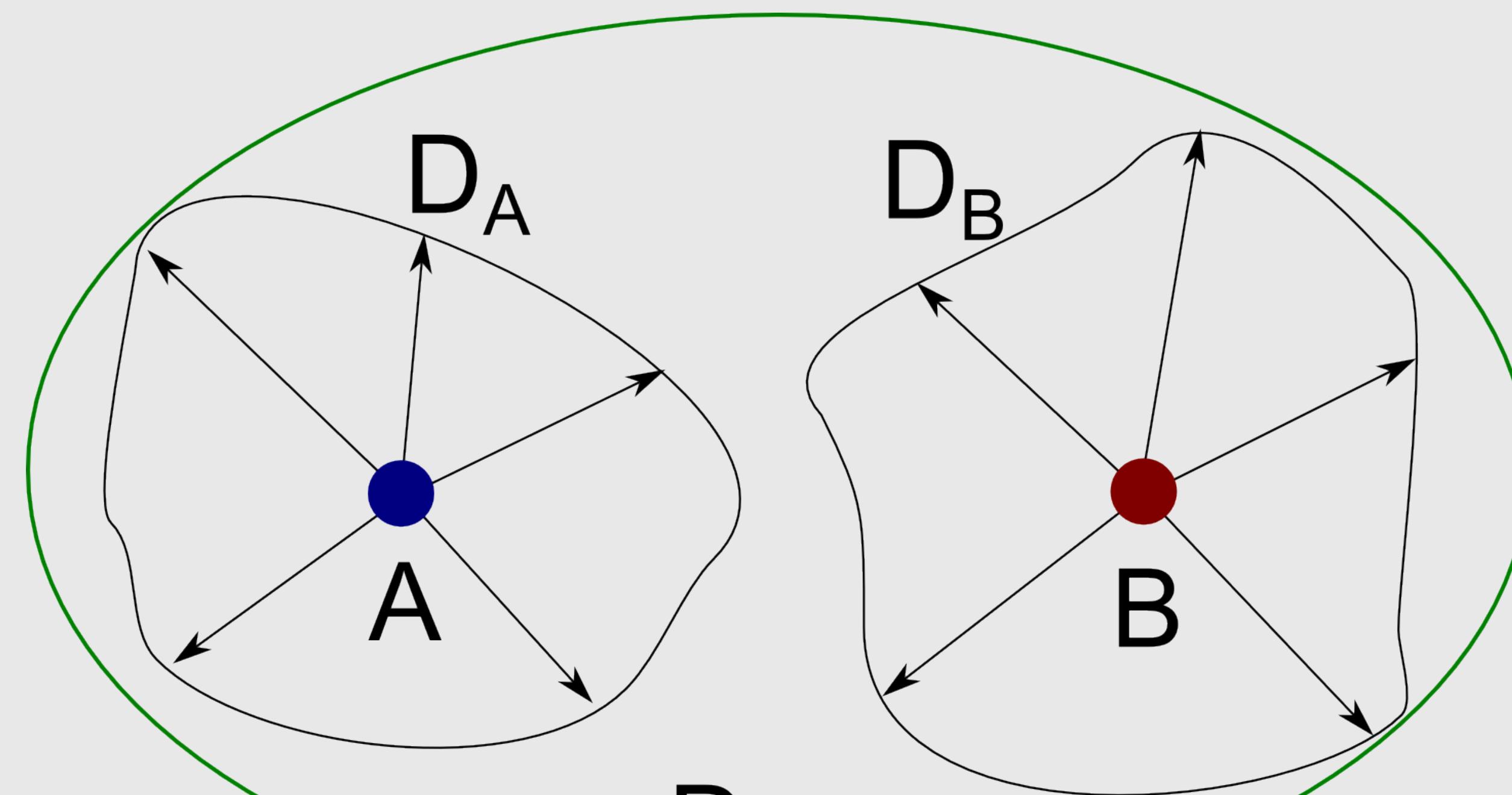
- Complete linkage



T

3. Clustering: Hierarchical Clustering

- Tipos de ligação: **Ward**

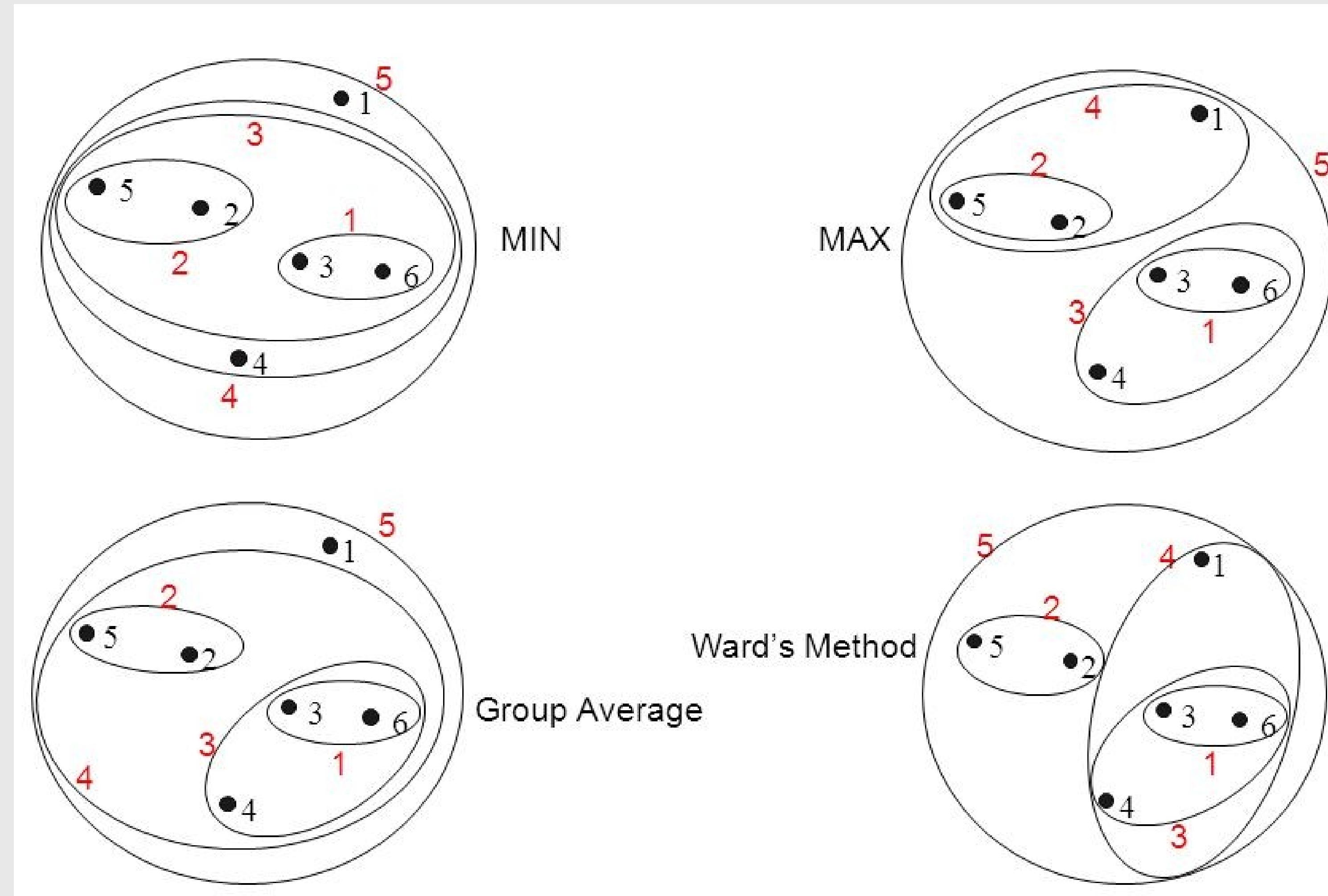


$$\text{Ward} = D_A + D_B - D_{AB}$$

I

3. Clustering: Hierarchical Clustering

- Tipos de ligação: Resultados distintos!



T

3. Clustering: Hierarchical Clustering

- Exemplo: notebook

3. Hierarchical Clustering

- **Vantagens**
 - Algoritmo simples
 - Resultado fácil de ser explicado
 - Não precisa definir o número de clusters
 - Permite analisar as relações entre clusters (hierarquia)
 - Resposta determinística

3. Hierarchical Clustering

- **Desvantagens**
 - Inviável com datasets grandes (cresce exponencialmente)
 - Pode convergir para decisões indesejadas
 - Difícil definir melhor região de corte
 - Depende da escolha de parâmetros de distância e ligação

T

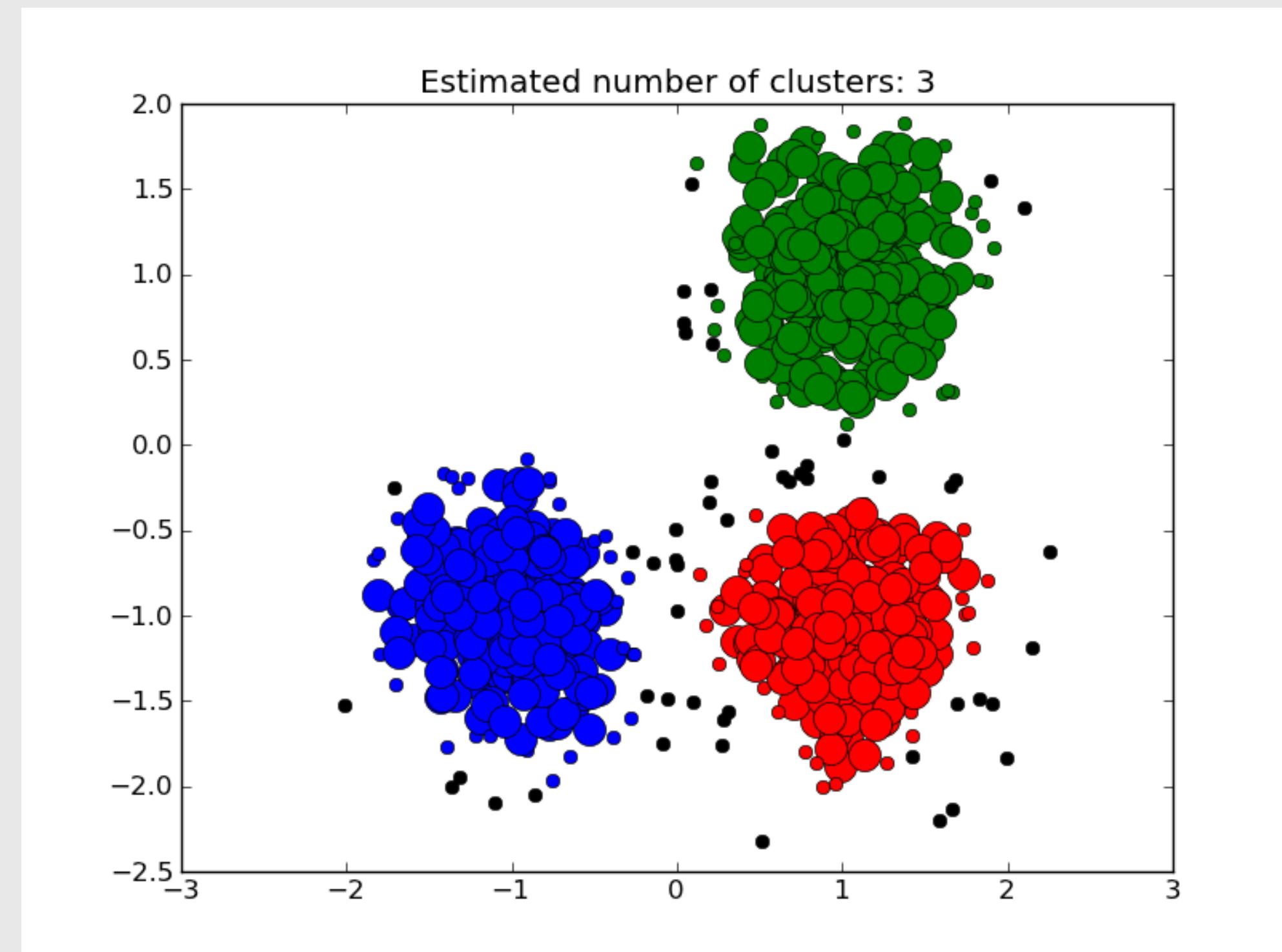
Clustering

- Outros algoritmos:
 - **Expectation-Maximization (EM)**
 - **Birch**
 - **Spectral Clustering**
 - **DBSCAN ***

T

4. Clustering: DBSCAN

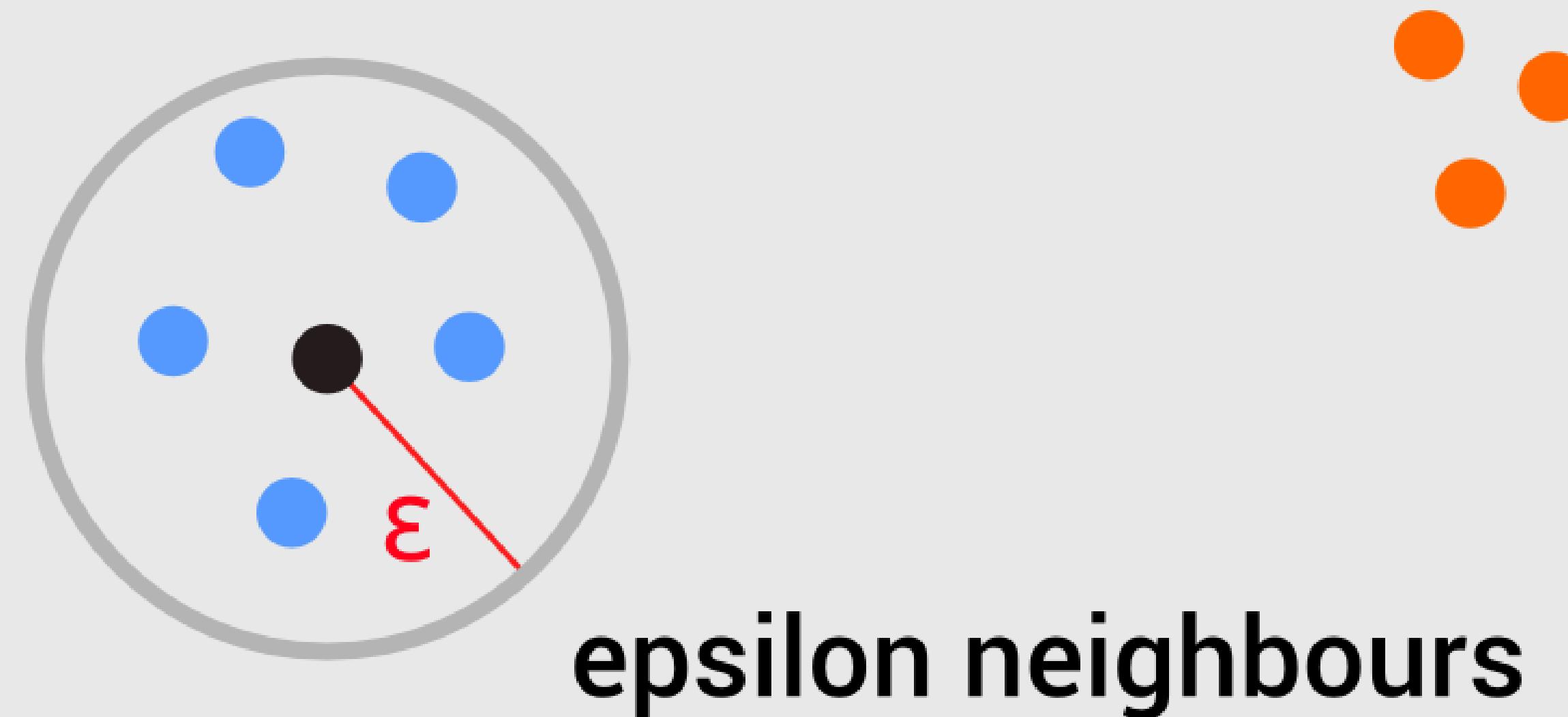
- Density-Based Spatial Clustering Algorithm with Noise
 - Utiliza o conceito de conectividade por densidade



T

4. Clustering: DBSCAN

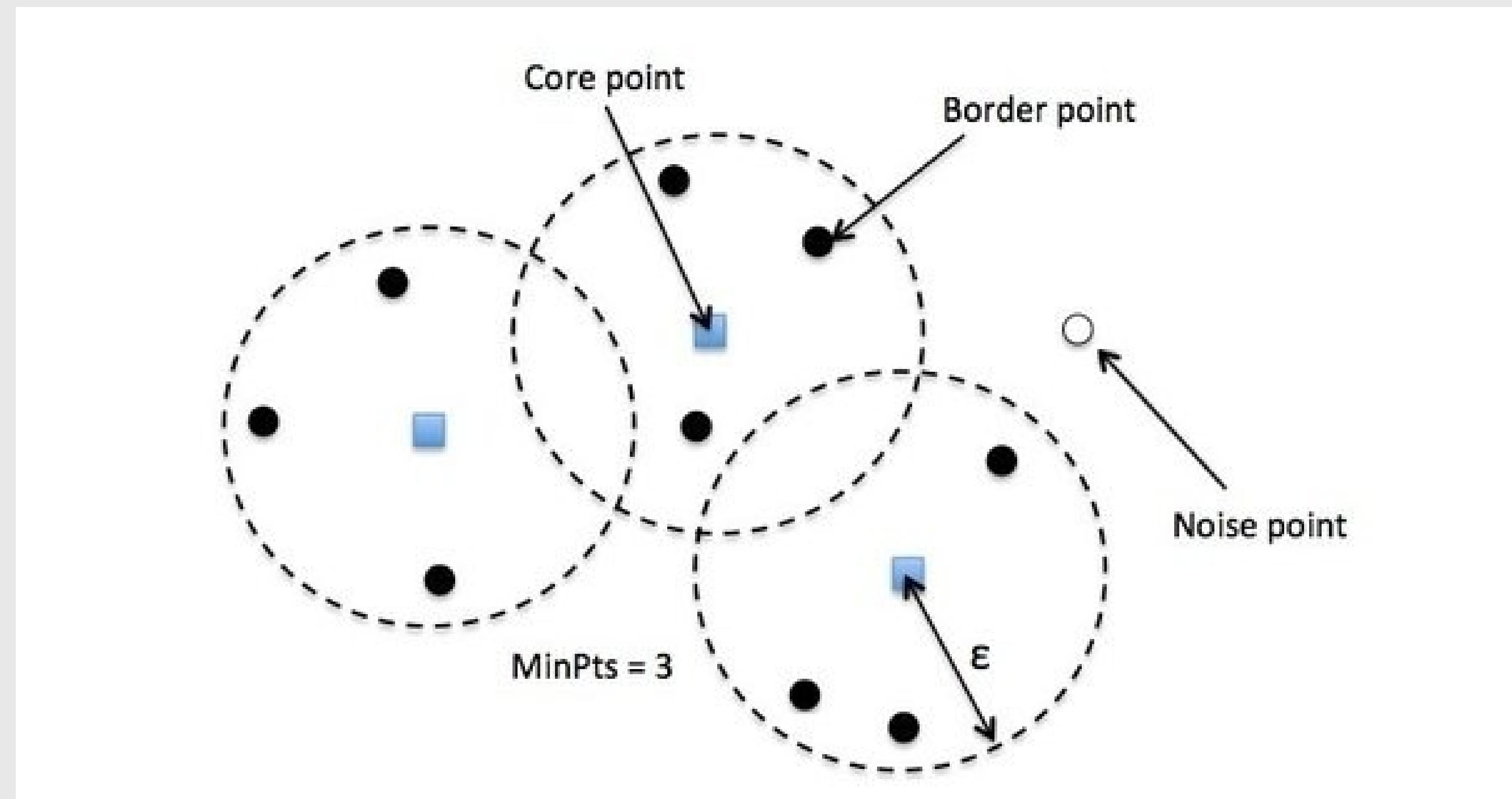
- Precisa escolher dois parâmetros: epsilon, min_points



I

4. Clustering: DBSCAN

- Conecta os pontos a partir de regiões de densidade



4. Clustering: DBSCAN

- **Vantagens:**

- Não precisa escolher o número de clusters
- Consegue eliminar possíveis outliers
- Escala facilmente para datasets grandes
- Funciona facilmente na prática com mínimos ajustes

T

4. Clustering: DBSCAN

- **Desvantagens:**

- Não há

4. Clustering: DBSCAN

- **Desvantagens:**
 - Mentira... A escolha dos parâmetros é crucial e depende de ajustes finos (“No free lunch”)
 - Sofre com clusters de diferentes densidades

T

4. Clustering

- Comparação métodos: Notebook

T

5. Maldição da Dimensionalidade



5. Maldição da Dimensionalidade

- Dimensão (d) do espaço de atributos (X):
 - Número de atributos
- Exemplo atributos: caracterizar gatos
 - Comprimento
 - Largura
 - Cor
 - Malhado ou não
 - Cor do olho
 - ...



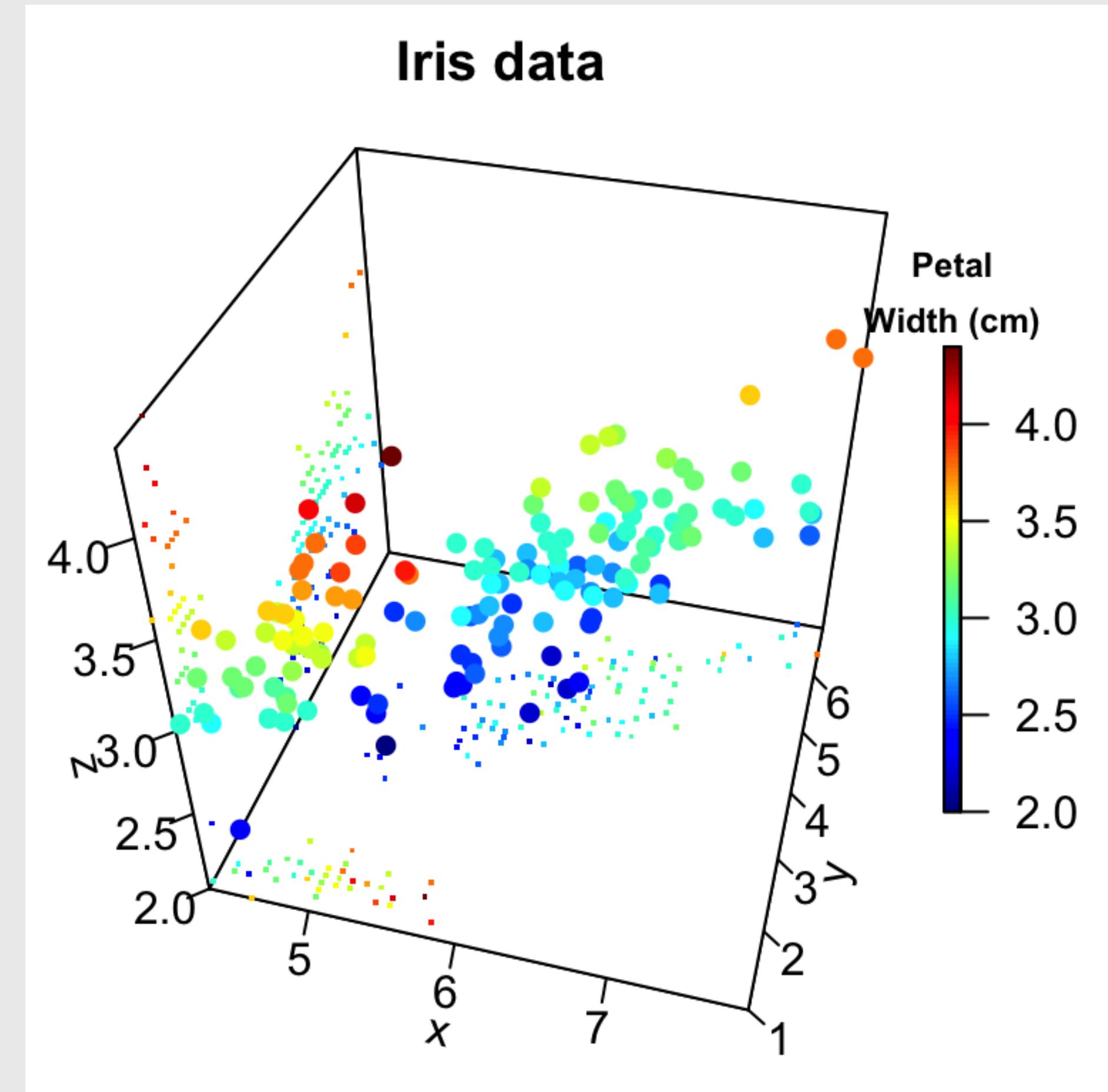
5. Maldição da Dimensionalidade

- Alto número de dimensões = problema:
 - Visualização / Intuição (> 3D)
 - Quantidade de dados necessários aumenta exponencialmente
 - Métricas de distância se tornam inúteis

I

5. Visualização

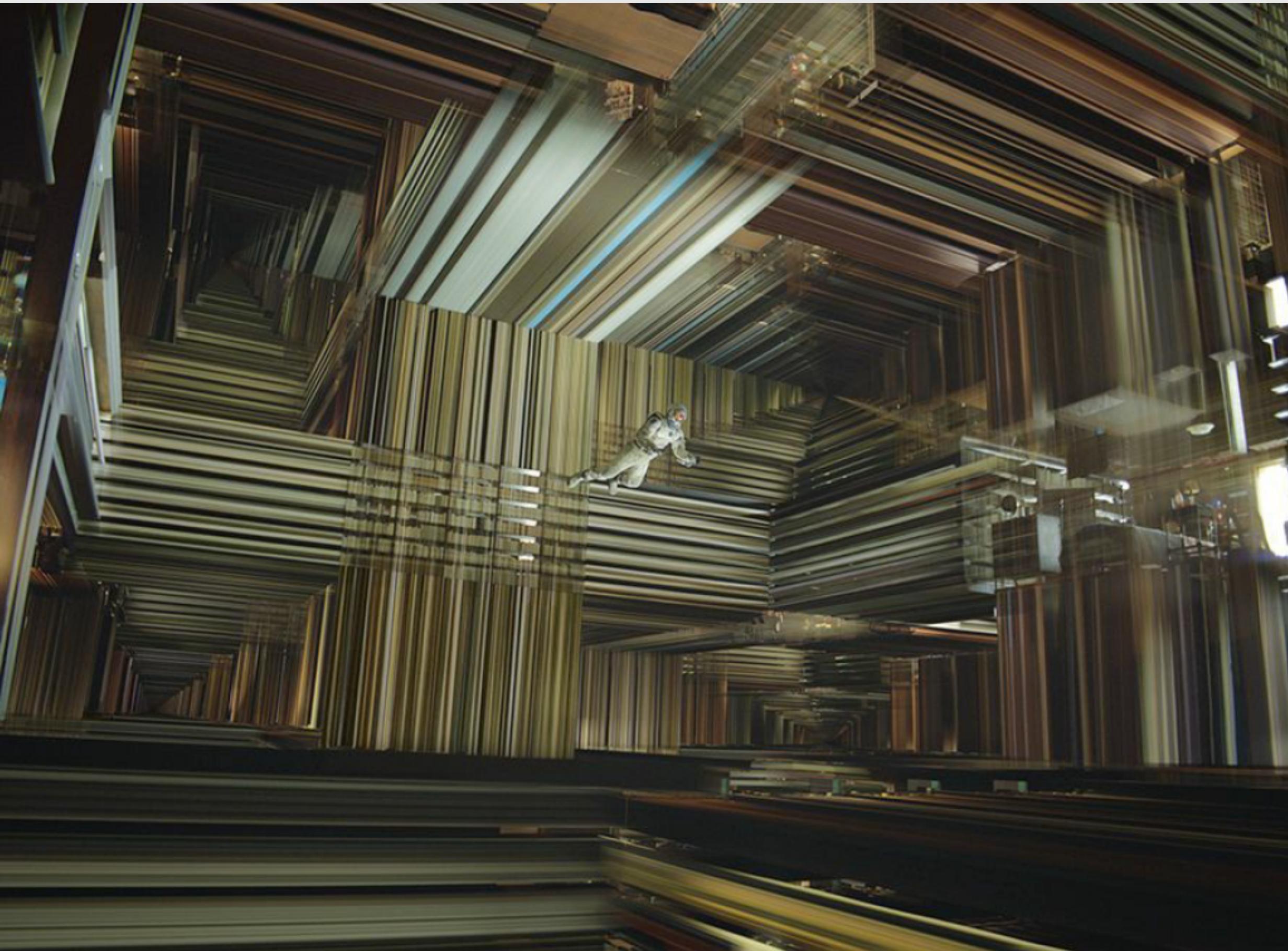
- Conseguimos visualizar bem até 3 dimensões



T

5. Visualização

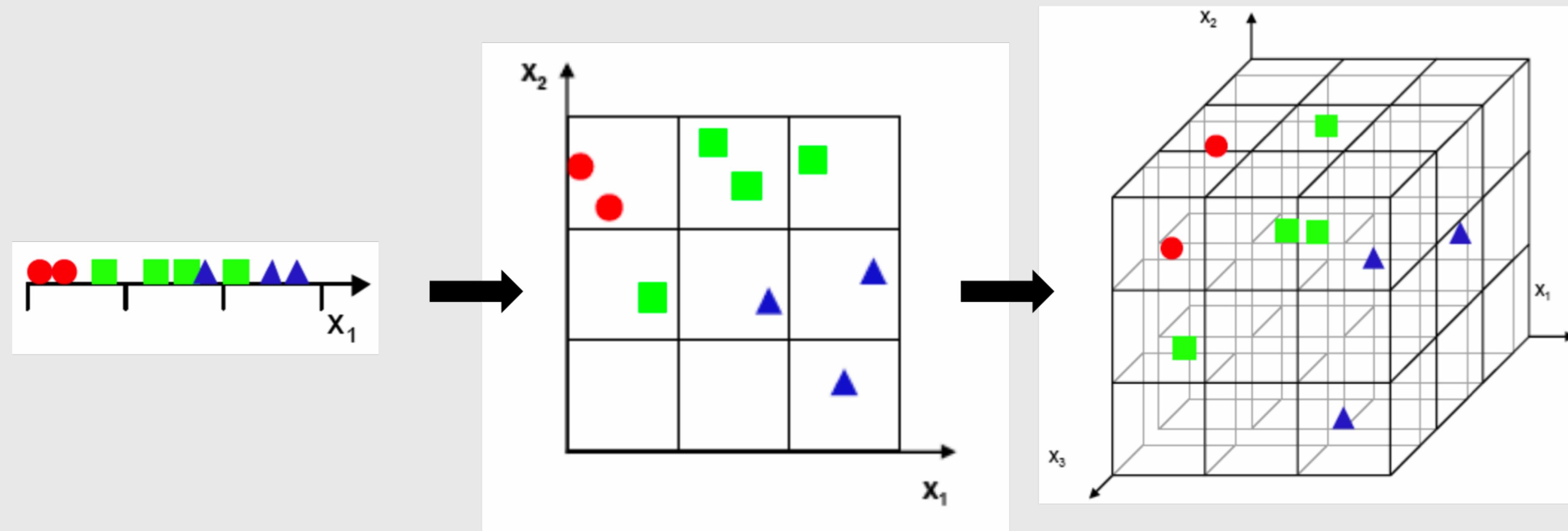
- Mas e mais do que 3 dimensões?



I

5. Maldição da Dimensionalidade

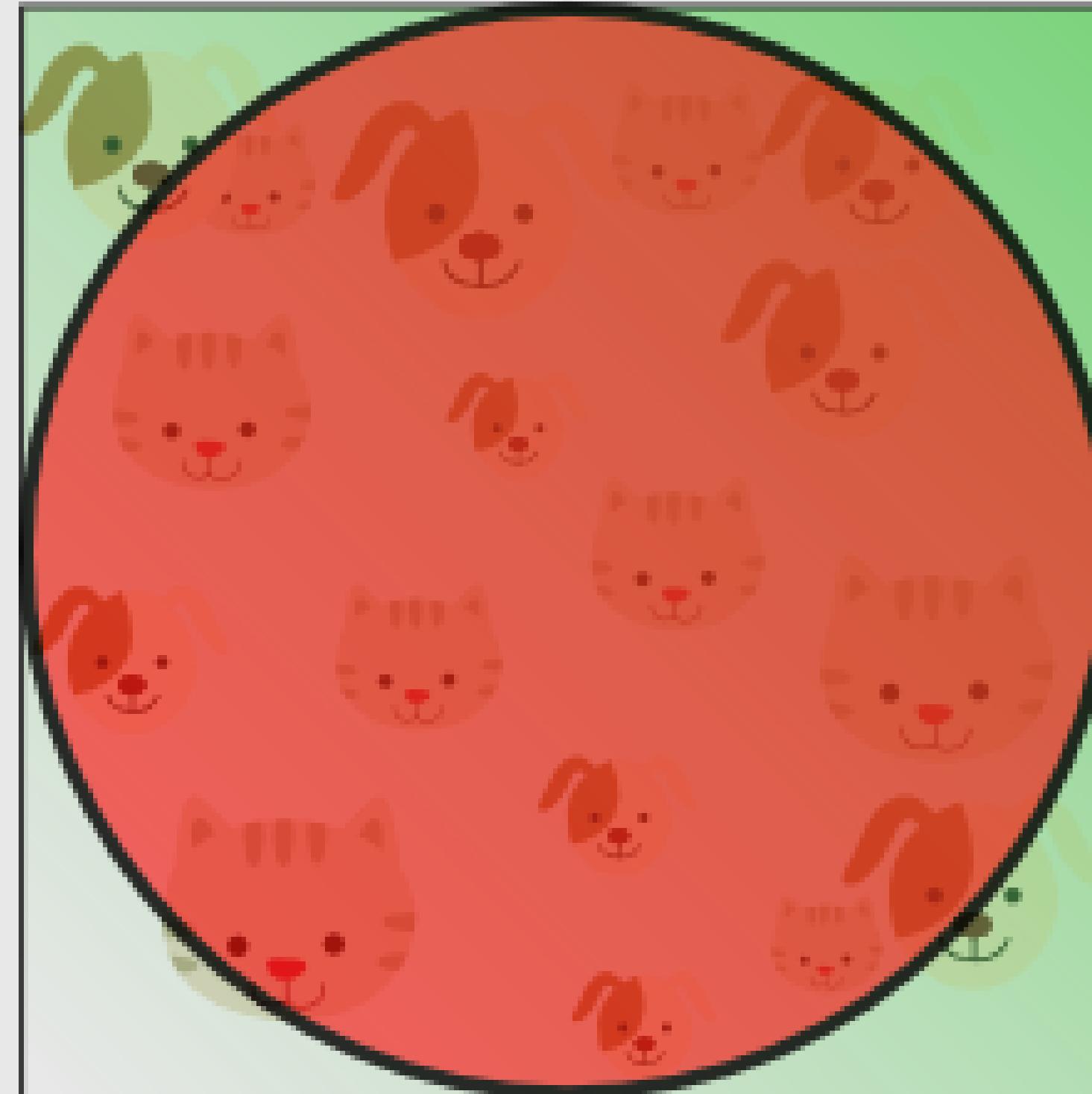
- Observações fixas distribuídas uniformemente
 - Mais dimensões → Dados mais esparsos (menos densos)



I

5. Maldição da Dimensionalidade

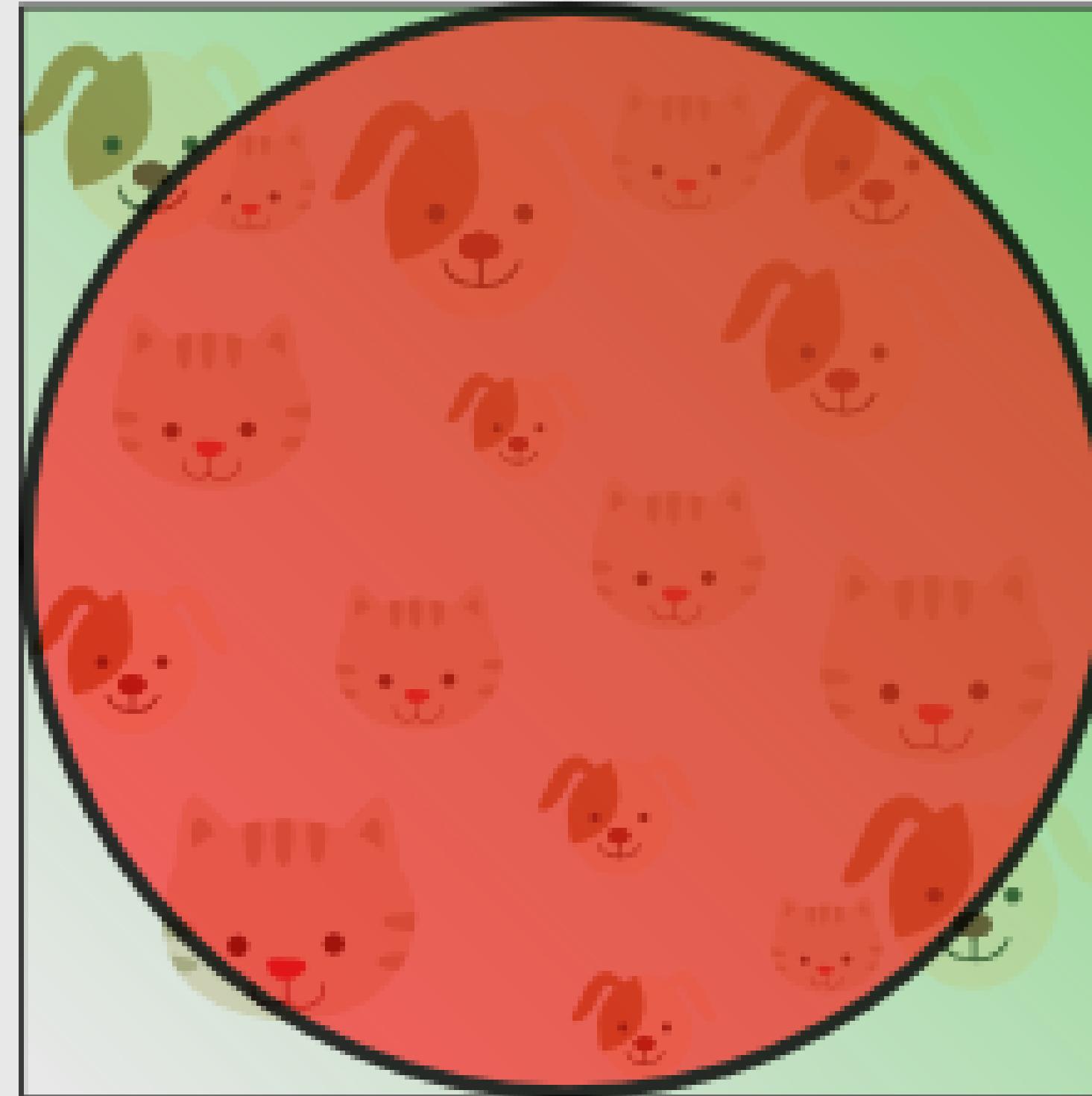
- Mais dimensões → Distâncias sem significado
 - Suponha que o espaço de atributos seja um quadrado
 - O círculo inscrito representa distâncias r do centro



I

5. Maldição da Dimensionalidade

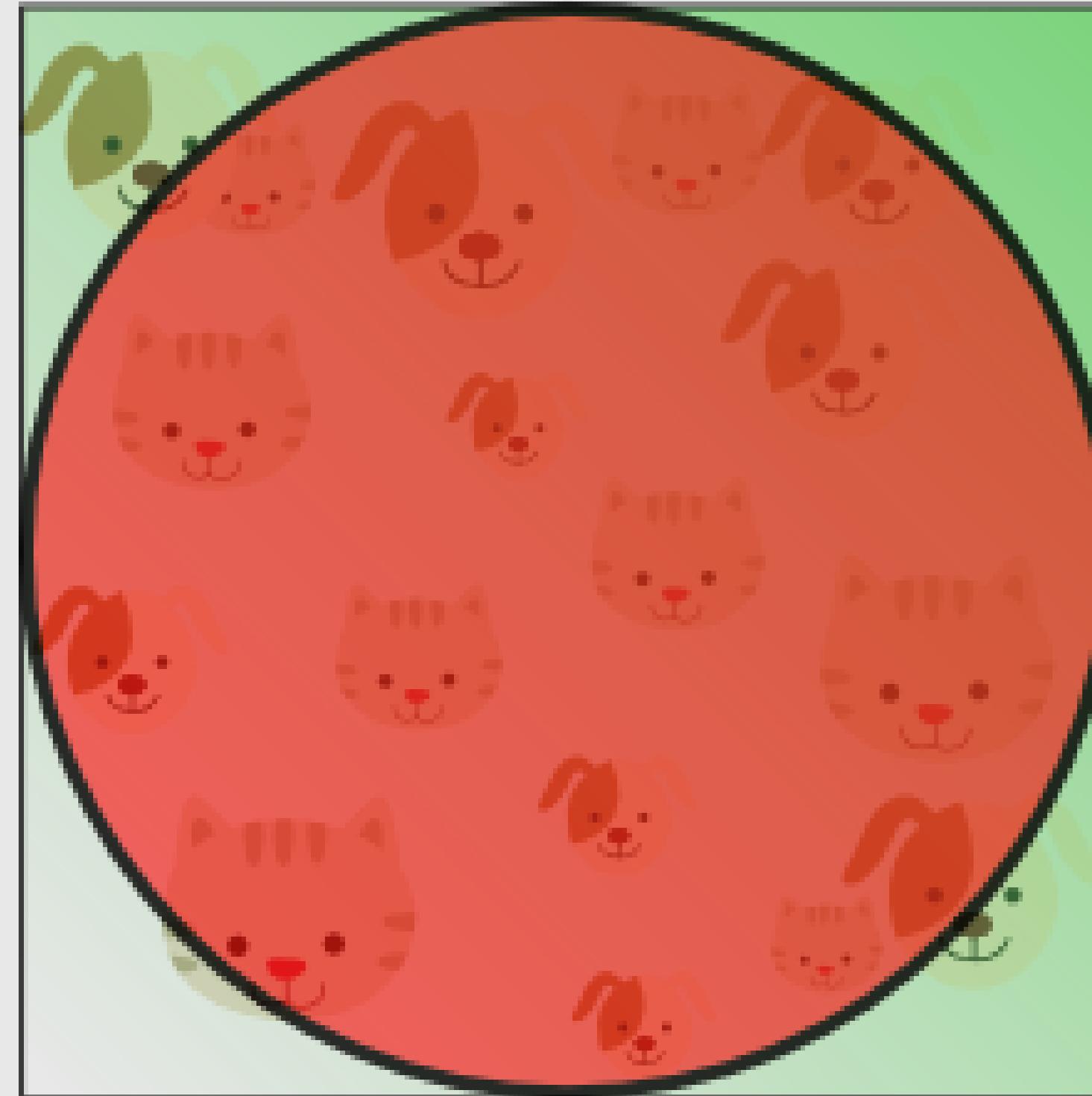
- Fora do círculo → Mais próximo do canto do que do centro
- Pontos nos cantos têm mesma distância



I

5. Maldição da Dimensionalidade

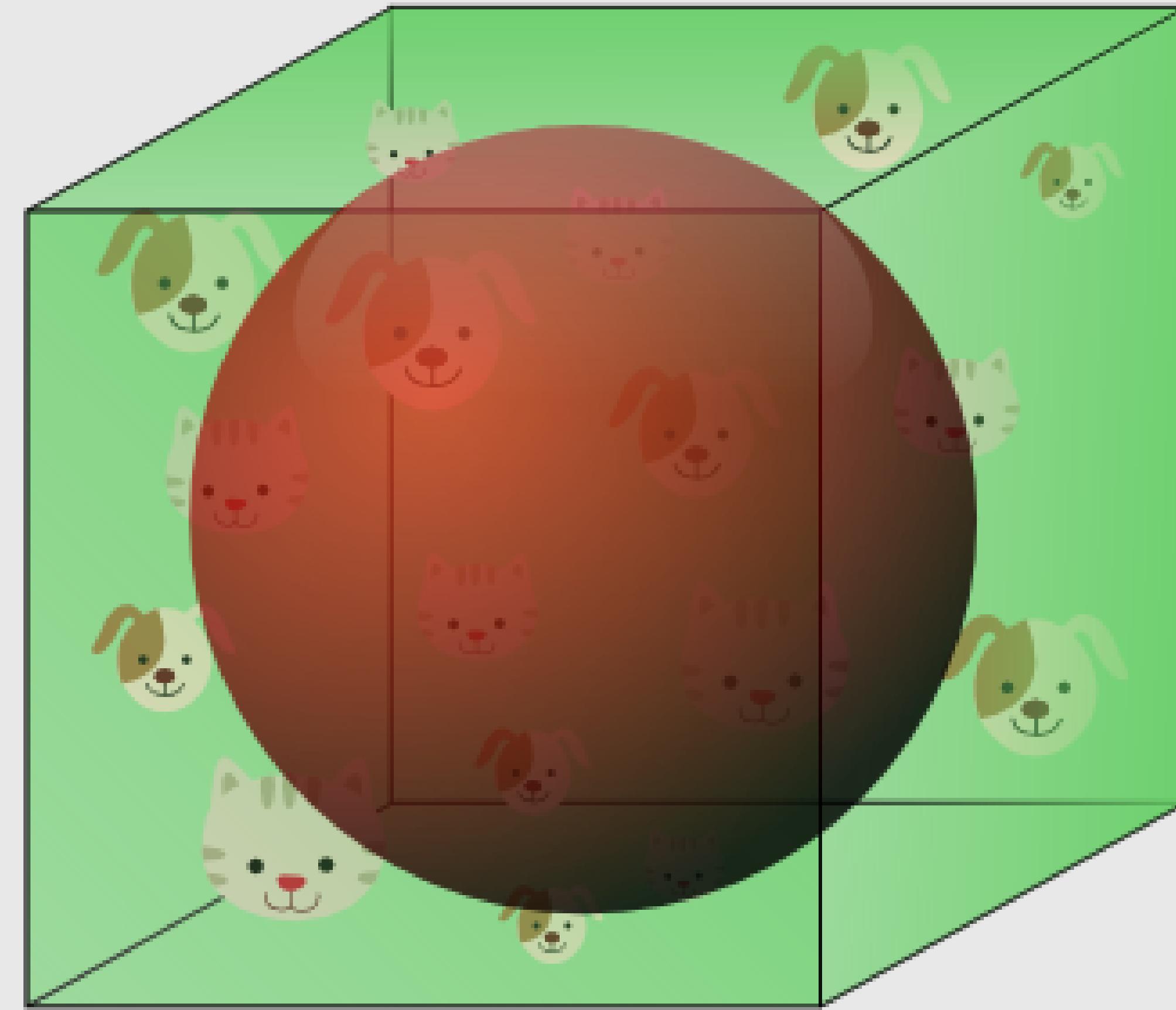
- Dados distribuídos uniformemente
 - Probabilidade de cair dentro do círculo: 78,5%



I

5. Maldição da Dimensionalidade

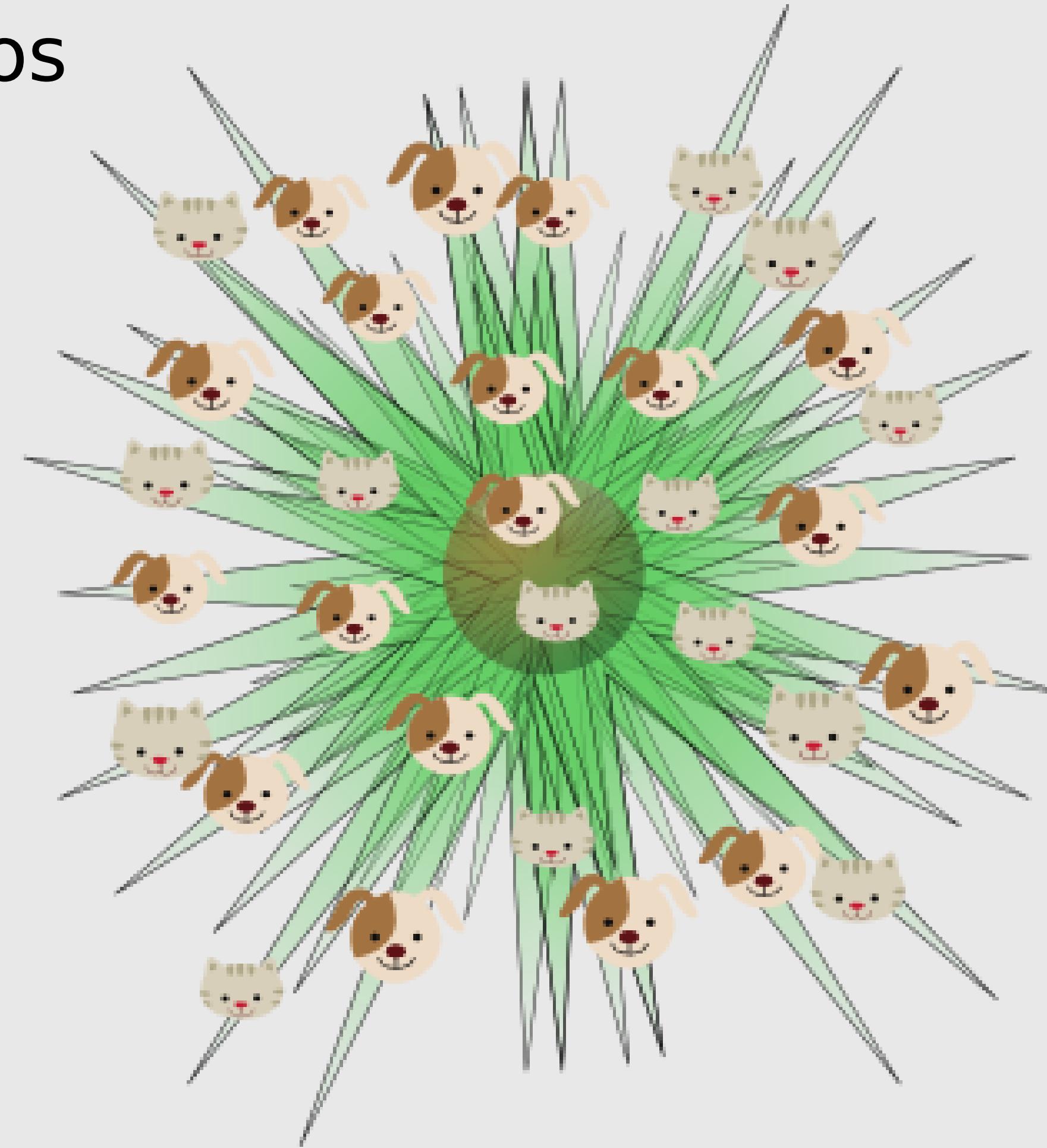
- 3D → Mais cantos: 8
- Probabilidade de cair dentro da esfera: 52,3%



I

5. Maldição da Dimensionalidade

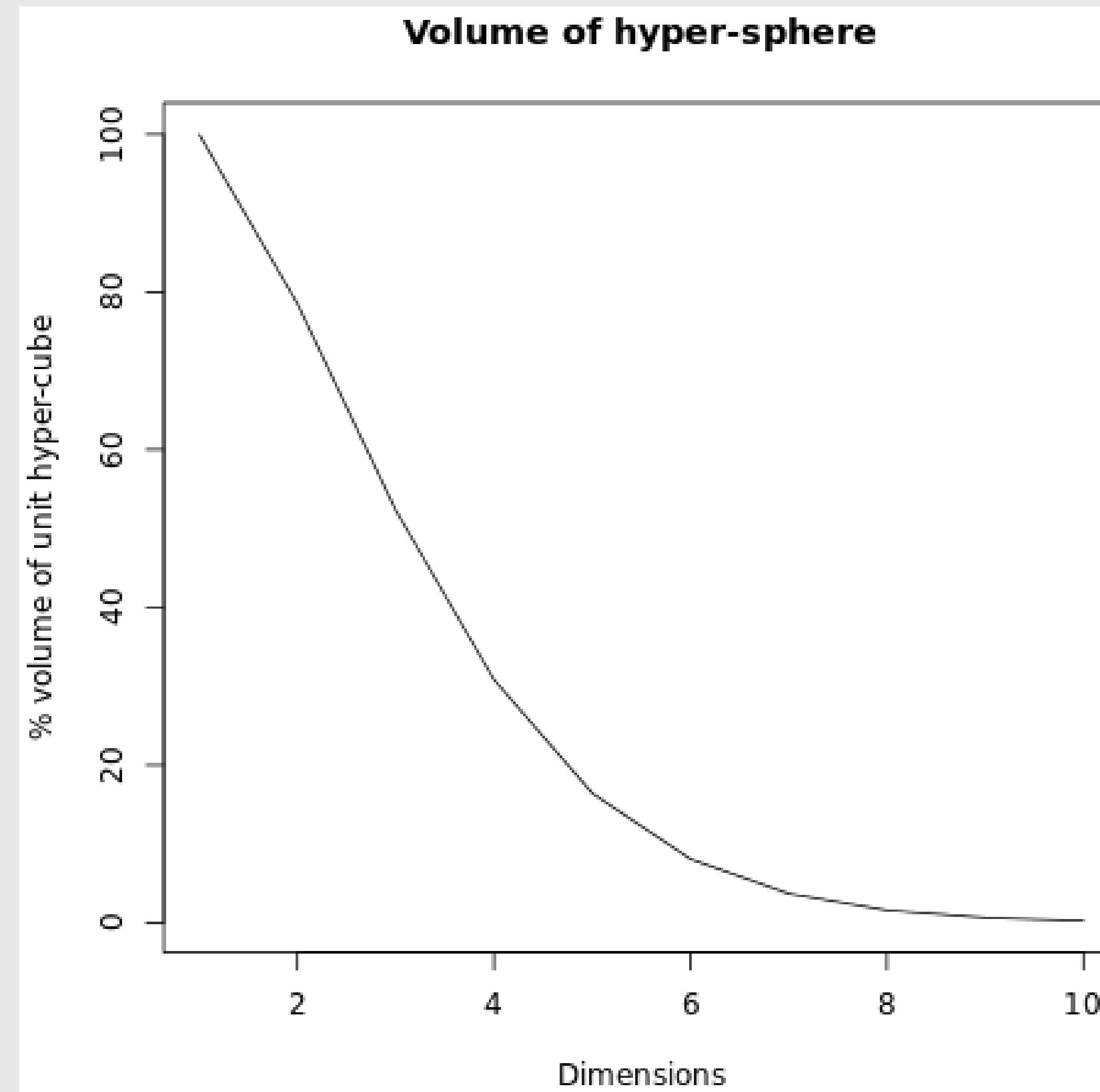
- 8D → Muitos cantos: 256
- Probabilidade de cair dentro da (hiper)esfera: ~2%
- Dados nos cantos



I

5. Maldição da Dimensionalidade

- Probabilidade de dados dentro da hiperesfera



I

5. Maldição da Dimensionalidade

- Resultado:
 - Todas as observações estão nos cantos
 - Pontos equidistantes
 - Não temos mais noção de similaridade!

I

5. Maldição da Dimensionalidade

- Solução:
 - Torcer para haver correlação forte entre dados
 - Redução de dimensionalidade

I

6. Redução de Dimensionalidade

- Dois objetivos principais:
 - Facilitar visualização e intuição
 - Amenizar o problema de similaridade entre observações

I

6. Redução de Dimensionalidade

- Técnicas principais:
 - PCA
 - T-SNE (Visualização apenas)

6. PCA

- Principal Component Analysis:
 - Encontra atributos “mais importantes” → Maior variação
 - Elimina atributos “menos úteis”

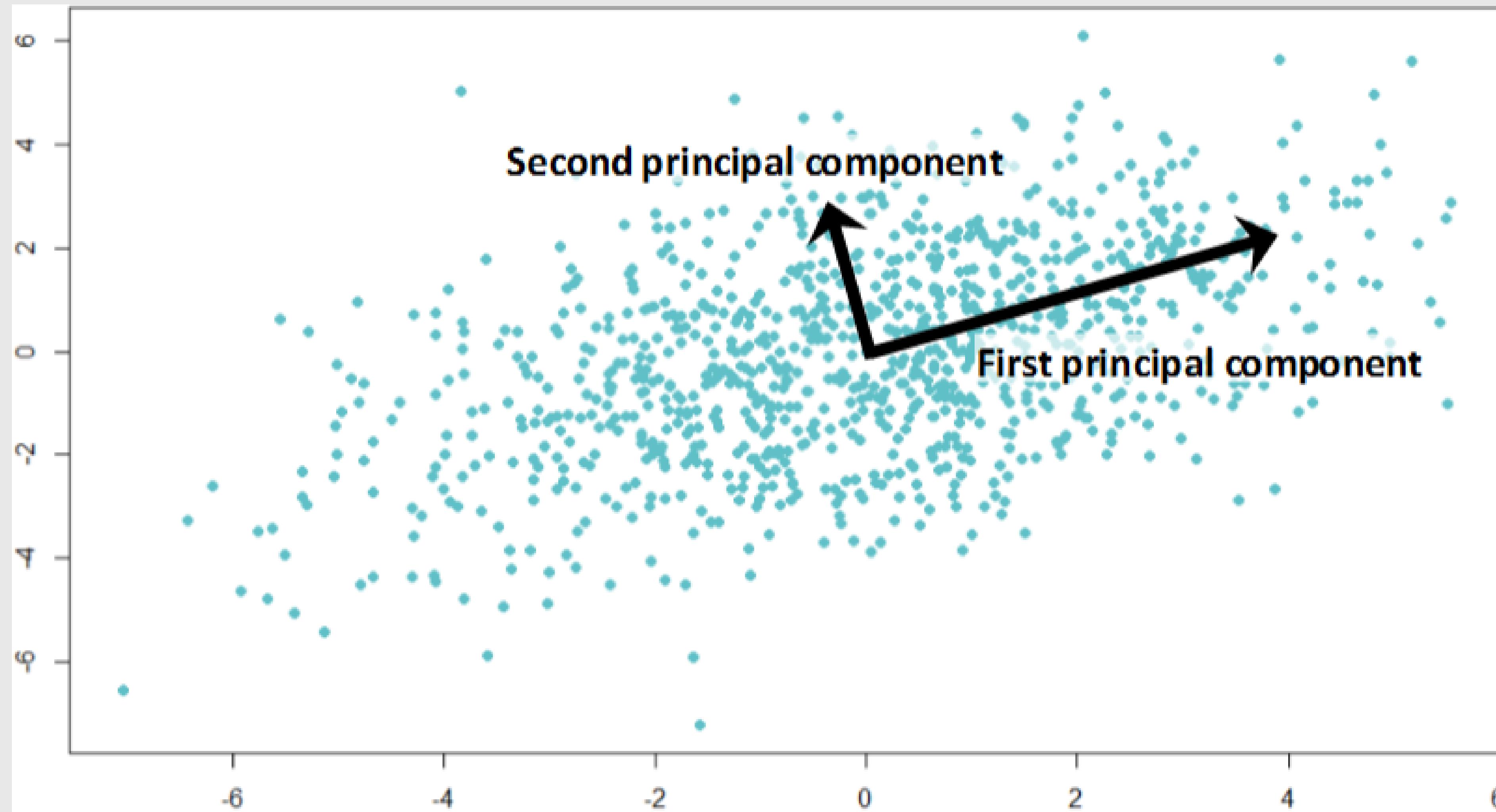
6. PCA

- PCA: Componentes principais:
 - **Primeiro componente:**
 - Direção (combinação linear) de maior variação nos dados
 - **Segundo componente:**
 - Direção da segunda maior variação e ortogonal ao primeiro (descorrelacionado)
 - **N-ésimo componente:**
 - Direção da n-ésima maior variação e ortogonal aos n-1 primeiros componentes

I

6. PCA

- Componentes principais:



6. PCA

- Passos do algoritmo PCA:
 - 1) Remove média amostral dos dados
 - 2) Rotaciona os eixos para descorrelacionar os atributos
 - 3) Ordena os componentes principais em nível de variância
 - 4) Remove os componentes menos variantes (Opcional)

I

6. PCA

- Exemplo: notebook

6. PCA

- **Vantagens:**
 - Permite reduzir dimensionalidade do problema sem perder informação
 - Menor dimensionalidade → Maior velocidade e menos memória para algoritmos de ML
 - Resultados determinísticos

6. PCA

- **Desvantagens:**

- Dimensões resultantes (componentes principais) não representam os atributos
- Perde a “explicabilidade” do algoritmo
- Má escolha de número de componentes pode prejudicar análise
- Encontra apenas relações lineares

5. Visualização: T-SNE

- **T-distributed Stochastic Neighbor Embedding**
 - Mapeia N-dimensões em 2 ou 3 dimensões
 - Distância é proporcional a probabilidade de proximidade entre pontos (afinidade)
 - Método iterativo baseado em otimização (gradiente descendente)
 - Procura manter a estrutura dos dados

T

6. T-SNE

- Mas, nós já temos o PCA. Por que usar T-SNE?

T

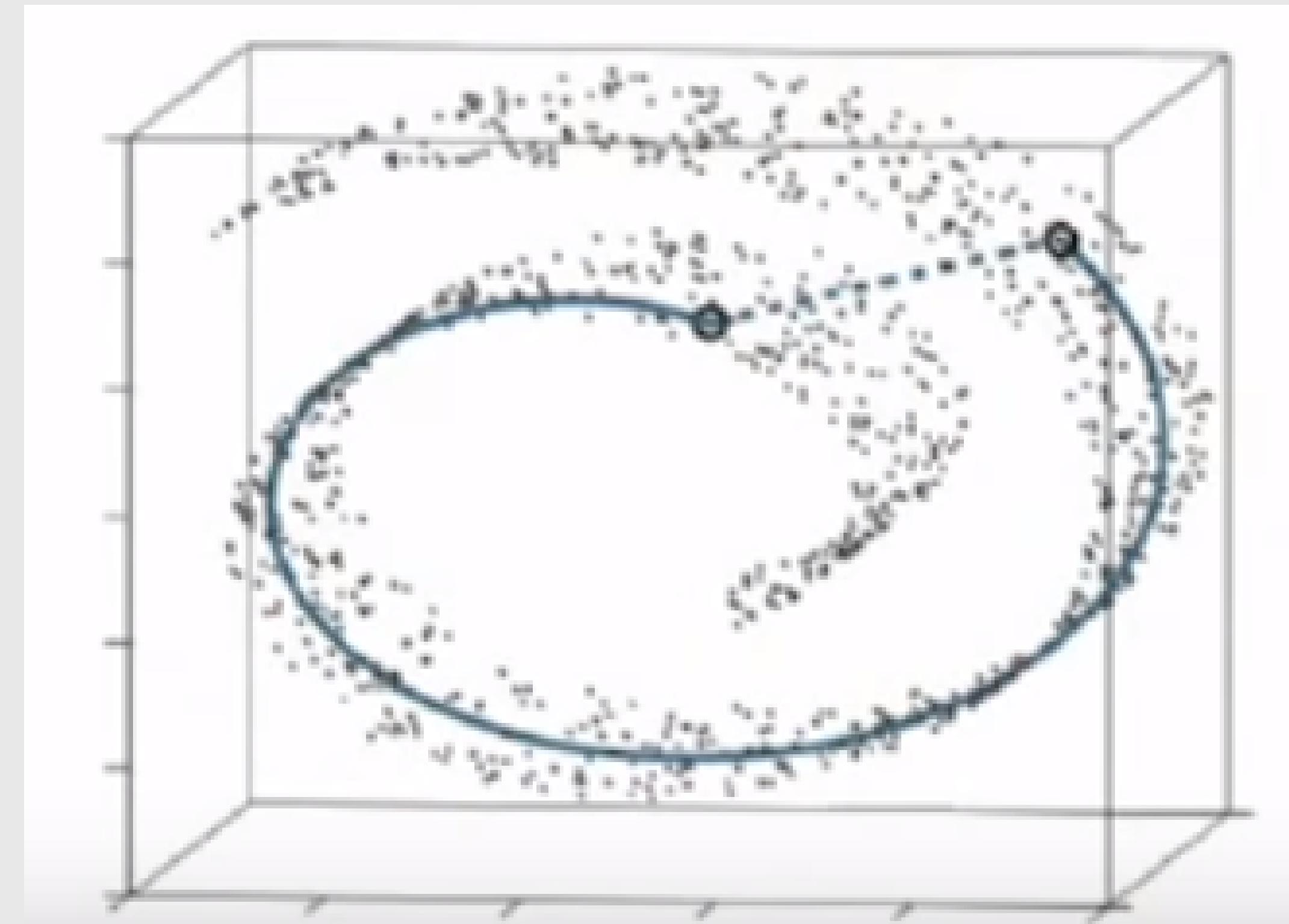
6. T-SNE

- Mas, nós já temos o PCA. Por que usar T-SNE?
 - PCA encontra apenas relações lineares
 - Falha ao encontrar estruturas complexas

I

6. T-SNE

- Mas, nós já temos o PCA. Por que usar T-SNE?
 - PCA encontra apenas relações lineares
 - Falha ao encontrar estruturas complexas



6. T-SNE

- Existem basicamente 2 parâmetros:
 - **Learning rate:** Taxa de aprendizado - gradiente descendente
 - **Perplexity:** ~ Número aproximado de vizinhos de um ponto (observação) – Entre 5 e 50

T

6. T-SNE

- Exemplo: notebook
- Projetor T-SNE - Tensorflow

6. T-SNE

- **Vantagens:**
 - Permite a visualização de relações entre dados multidimensionais
 - Mantém a estrutura dos dados (não-linear)
 - Rápido e eficiente mesmo para grandes dimensões e grande quantidade de observações

6. T-SNE

- **Desvantagens:**
 - Depende da escolha dos parâmetros (nem sempre fácil)
 - Não possui repitibilidade dos resultados
 - Distâncias entre clusters não significam nada
 - Interpretação dos resultados não trivial

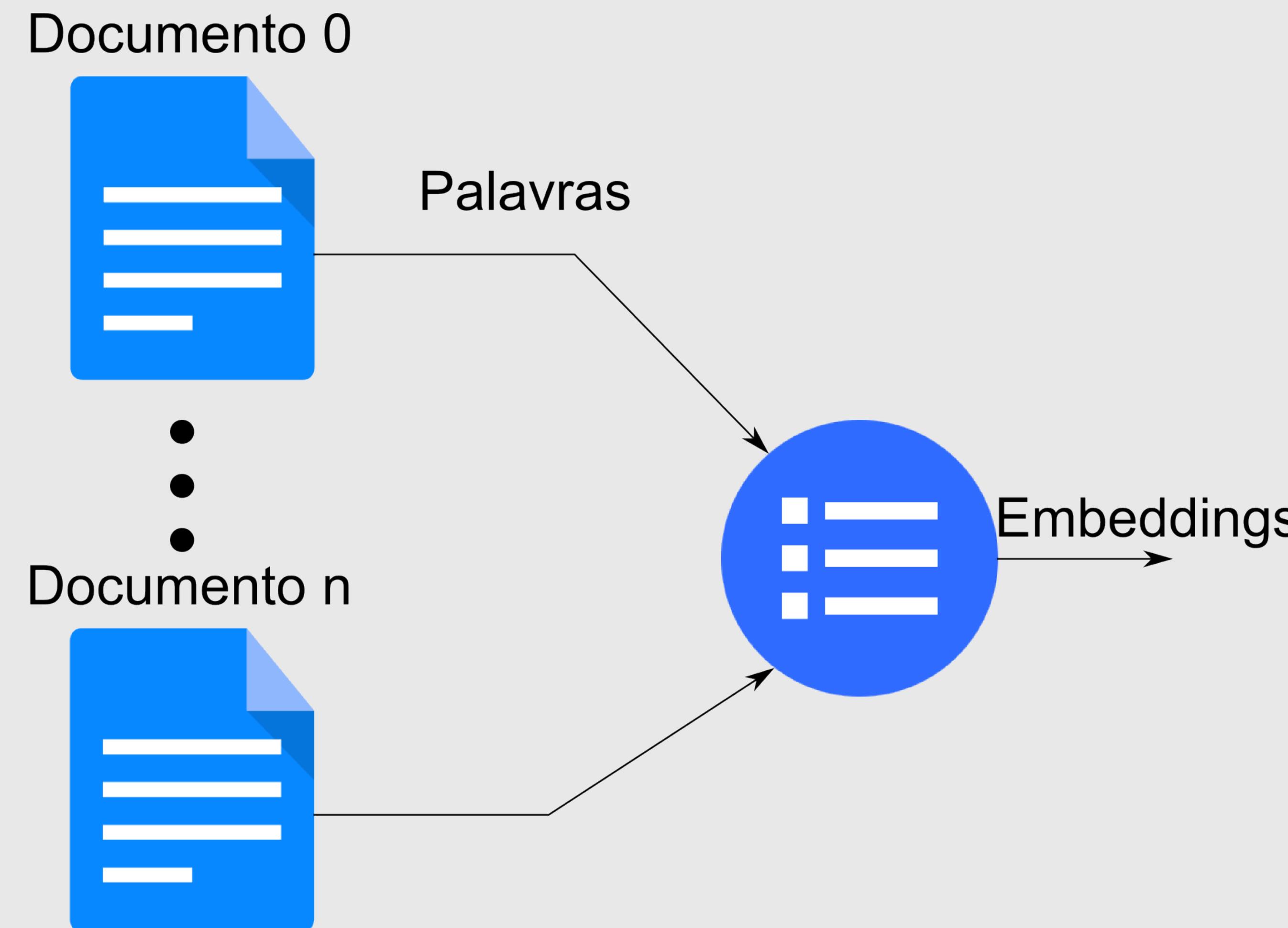
7. Clustering NLP

- Objetivos:
 - Encontrar relações entre documentos
 - Relações entre palavras
 - Documentos semelhantes possuem conjuntos de palavras semelhantes

T

7. Clustering NLP

- Vetor de atributos de um texto:



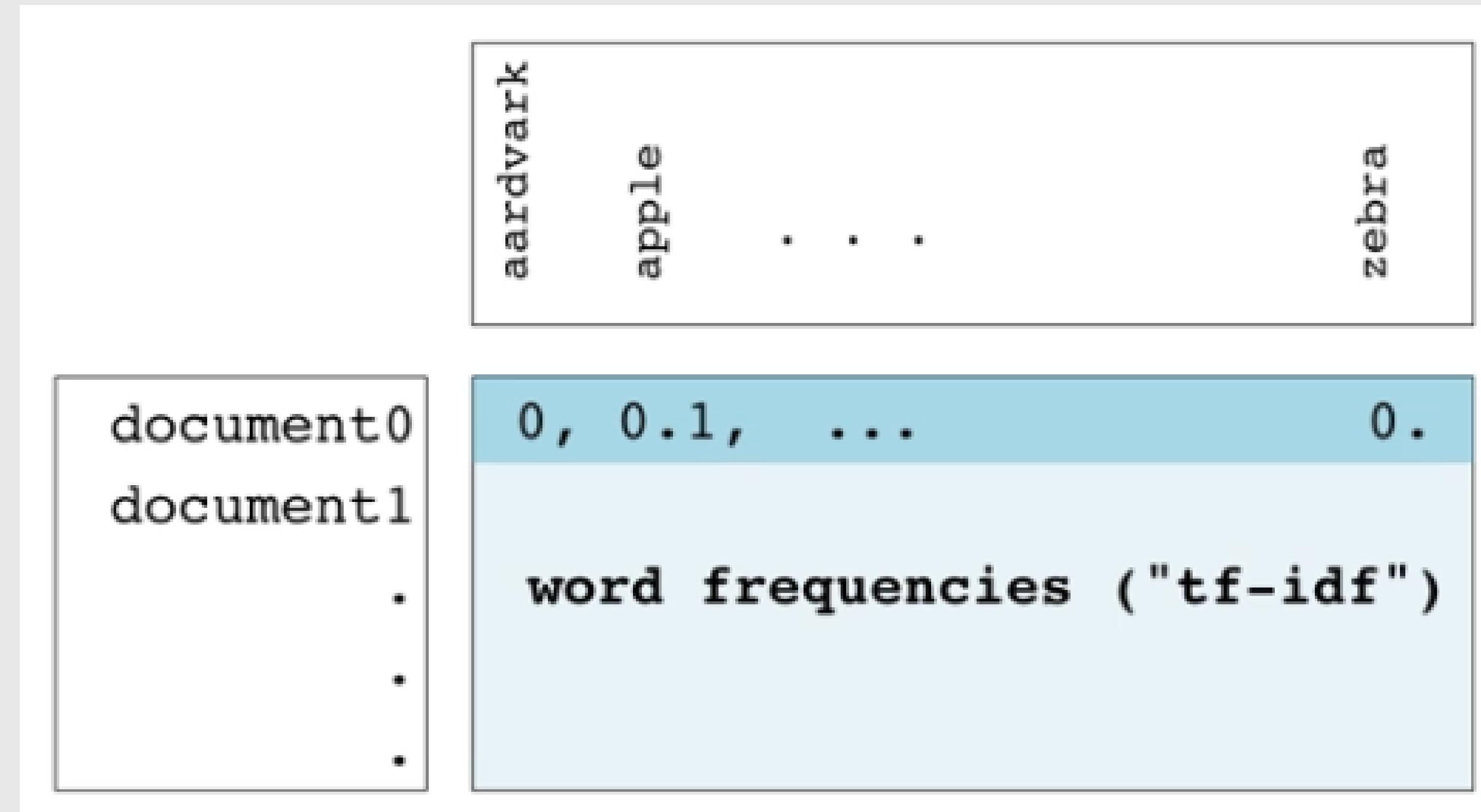
7. Clustering NLP

- Vetor de atributos de um texto (embedding) - X:
 - Bag-of-Words
 - TF-IDF
 - Word2Vec
 - Glove
 - FastText

I

7. Clustering NLP

- Vetor de atributos de um texto:



T

7. Clustering NLP

- Vetor de atributos de um texto:
 - Cada palavra é uma dimensão :-)
 - Matriz esparsa :-)

	aardvark	apple	...	zebra
document0	0,	0.1,	...	0.
document1
.
.
.
	word frequencies ("tf-idf")			

T

7. Clustering NLP

- Solução: **PCA!**
- Vamos a um exemplo prático no notebook

T

OBRIGADO!