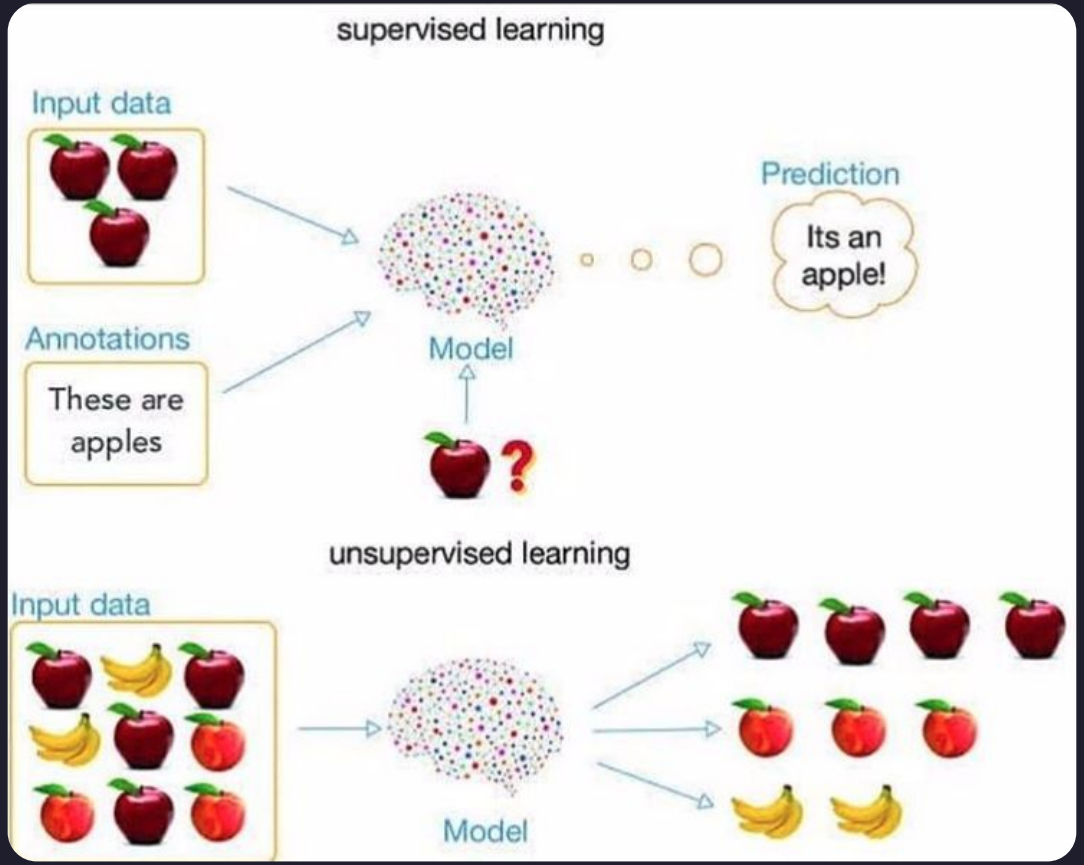
MACHINE LEARNING: CLUSTERING

01

INTRODUÇÃO


supervised learning
Input data
Prediction
Its an apple!
Annotations
These are apples
Model
unsupervised learning
Input data
Model

INTRODUÇÃO

- No aprendizado supervisionado, todas os exemplos de treinamento eram rotulados.
- Estes exemplos são ditos “supervisionados”, pois os dados de treinamento contém tanto a entrada (atributos), quanto a saída (classe).

INTRODUÇÃO

O que podemos fazer quando temos em frente um conjunto de dados sem rótulo?

Utilizaremos técnicas não supervisionadas!

INTRODUÇÃO

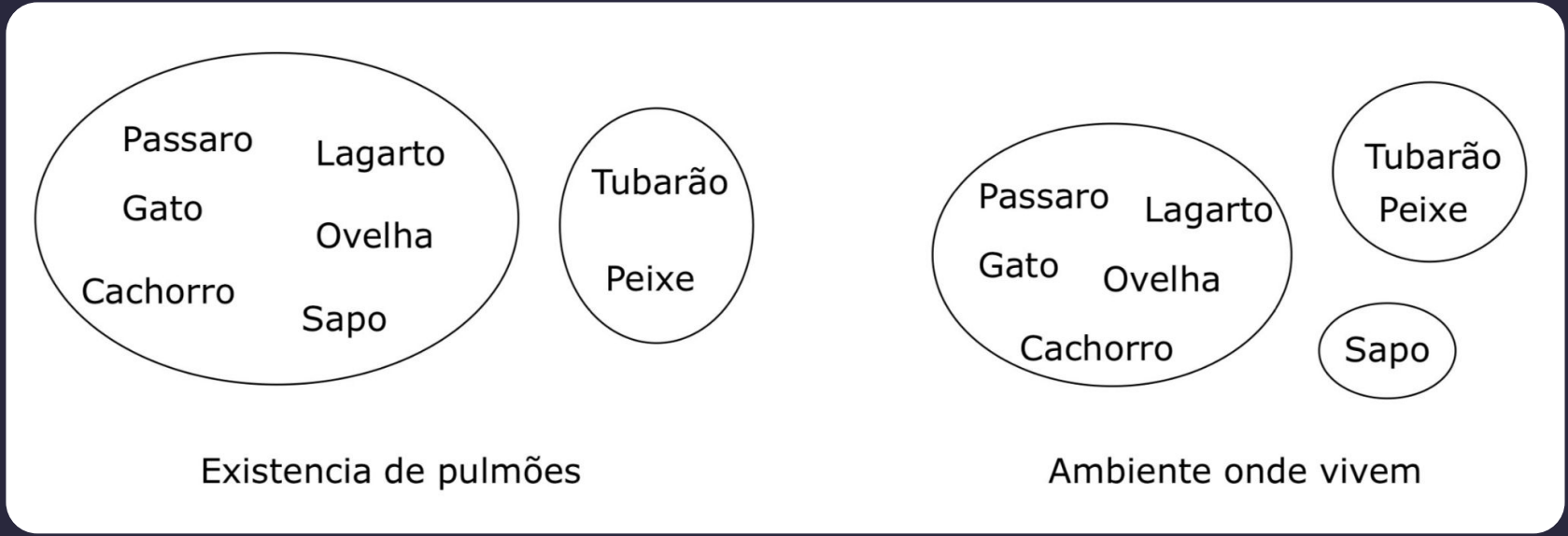
- Entretanto, podemos utilizar grandes quantidades de dados não rotulados para encontrar padrões existentes nestes dados.
  - E somente depois supervisionar a rotulação dos agrupamentos encontrados.
- Esta abordagem é bastante utilizada em aplicações de mineração de dados (data mining), no qual o conteúdo de grandes bases de dados não é conhecido antecipadamente.
- Técnica não-paramétrica.
  - Isso será muito útil na prática, onde a maioria dos conjuntos de dados do mundo real não segue pressupostos teóricos matemáticos

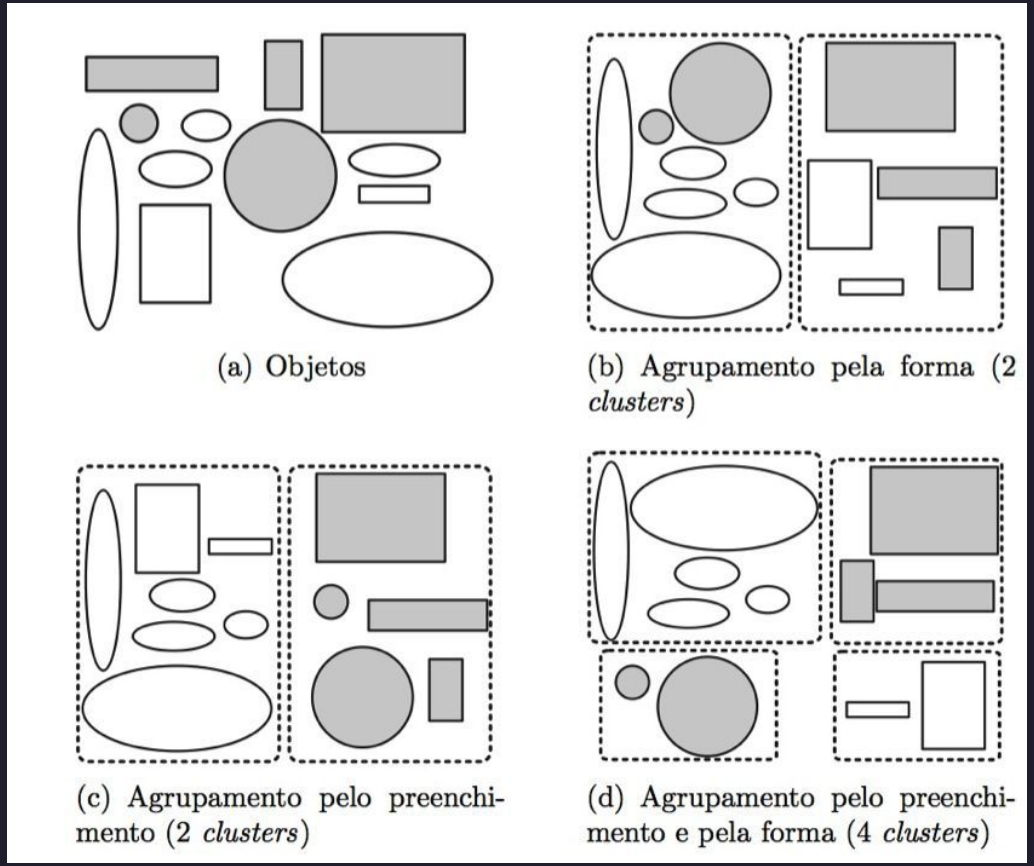
INTRODUÇÃO

- O principal interesse do aprendizado não supervisionado é desvendar a organização dos padrões existentes nos dados por meio de clusters (agrupamentos) consistentes.
- Com isso, é possível descobrir similaridades e diferenças entre os padrões existentes, assim como derivar conclusões úteis a respeito deles.

INTRODUÇÃO

Exemplos de agrupamentos (*clusters*)





(a) Objetos

(b) Agrupamento pela forma (2 *clusters*)

(c) Agrupamento pelo preenchimento (2 *clusters*)

(d) Agrupamento pelo preenchimento e pela forma (4 *clusters*)

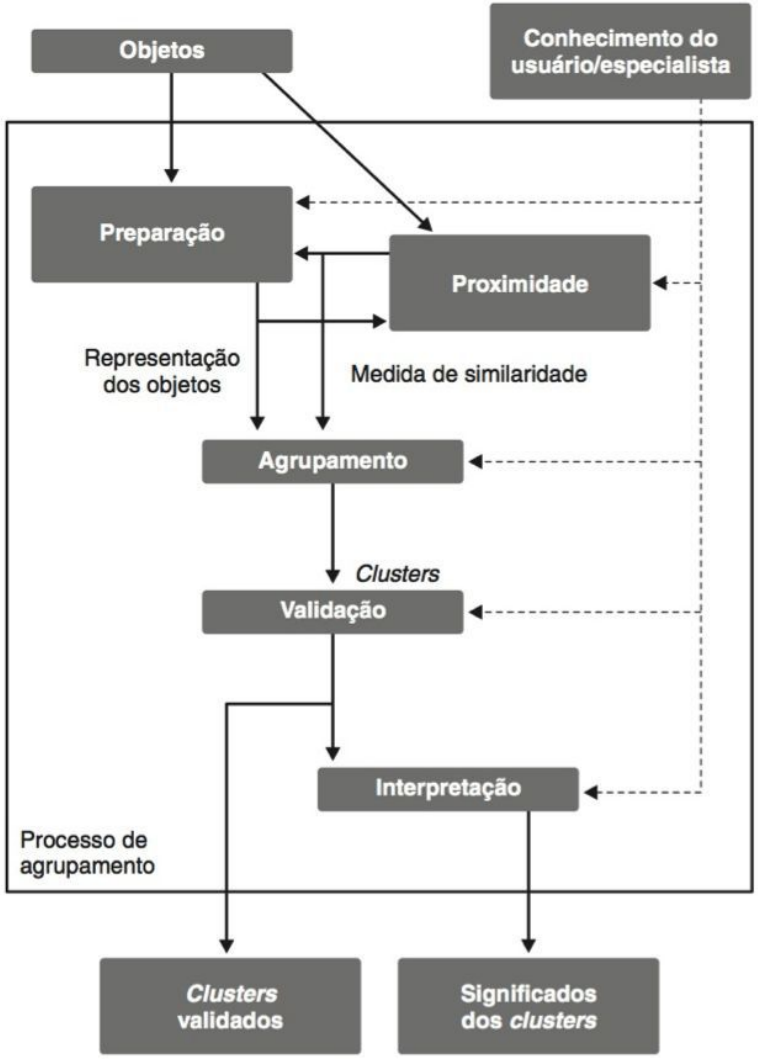
CLUSTERIZAÇÃO

- A clusterização é o processo de agrupar um conjunto de objetos físicos ou abstratos em classes de objetos similares.
- Um cluster é uma coleção de objetos que são similares uns aos outros (de acordo com algum critério de similaridade pré-definido) e dissimilares a objetos pertencentes a outros clusters.

CRITÉRIO DE SIMILARIDADE

Similaridade é fácil de ser definida?

VISÃO GERAL - APRENDIZADO NÃO-SUPERVISIONADO



ETAPAS DO PROCESSO DE APRENDIZADO NÃO SUPERVISIONADO

1. Seleção de atributos (preparação)
2. Medida de proximidade (proximidade)
3. Critério de agrupamento (agrupamento)
4. Algoritmo de agrupamento (agrupamento)
5. Verificação dos resultados (validação)
6. Interpretação dos resultados (interpretação)

PROCESSO DE APRENDIZADO NÃO-SUPERVISIONADO

Seleção de Atributos:

- Atributos devem ser adequadamente selecionados de forma a codificar a maior quantidade possível de informações relacionada a tarefa de interesse.
- Os atributos devem ter também uma redundância mínima entre eles.

PROCESSO DE APRENDIZADO NÃO-SUPERVISIONADO

Medida de Proximidade:

- Medida para quantificar quão similar ou dissimilar são dois vetores de atributos.
- É ideal que todos os atributos contribuam de maneira igual no cálculo da medida de proximidade.
  - Um atributo não pode ser dominante sobre o outro, ou seja, é importante normalizar os dados.

PROCESSO DE APRENDIZADO NÃO-SUPERVISIONADO

Medidas de Dissimilaridade

- Métrica lp ponderada;
- Métrica Norma l∞ ponderada;
- Métrica l2 ponderada (Mahalanobis);
- Métrica lp especial (Manhattan);
- Distância de Hamming.

Medidas de Similaridade

- Produto interno (inner);
- Medida de Tanimoto.

PROCESSO DE APRENDIZADO NÃO-SUPERVISIONADO

Critério de Agrupamento

- Depende da interpretação que o especialista dá ao termo sensível com base no tipo de cluster que são esperados.
- Por exemplo, um cluster compacto de vetores de atributos pode ser sensível, de acordo com um critério, enquanto outro cluster alongado pode ser sensível, de acordo com outro critério.

PROCESSO DE APRENDIZADO NÃO-SUPERVISIONADO

Algoritmo de Agrupamento

- Tendo adotado uma medida de proximidade e um critério de agrupamento devemos escolher um algoritmo de clusterização que revele a estrutura agrupada do conjunto de dados.
- A seguir, os tipos de algoritmos serão comentados na aula.

PROCESSO DE APRENDIZADO NÃO-SUPERVISIONADO

Validação dos Resultados:

- Uma vez obtidos os resultados do algoritmo de agrupamento, devemos verificar se o resultado está correto.
- Isto geralmente é feito por meio de testes apropriados.

PROCESSO DE APRENDIZADO NÃO-SUPERVISIONADO

Interpretação dos Resultados:

- Em geral, os resultados da clusterização devem ser integrados com outras evidências experimentais e análises para chegar às conclusões corretas.

PROCESSO DE APRENDIZADO NÃO-SUPERVISIONADO

- Diferentes escolhas de atributos, medidas de proximidade, critérios de agrupamento e algoritmos de clusterização levam a resultados totalmente diferentes.
  - Qual resultado é o correto?
  - Quais os atributos são corretos?

CLUSTERIZAÇÃO

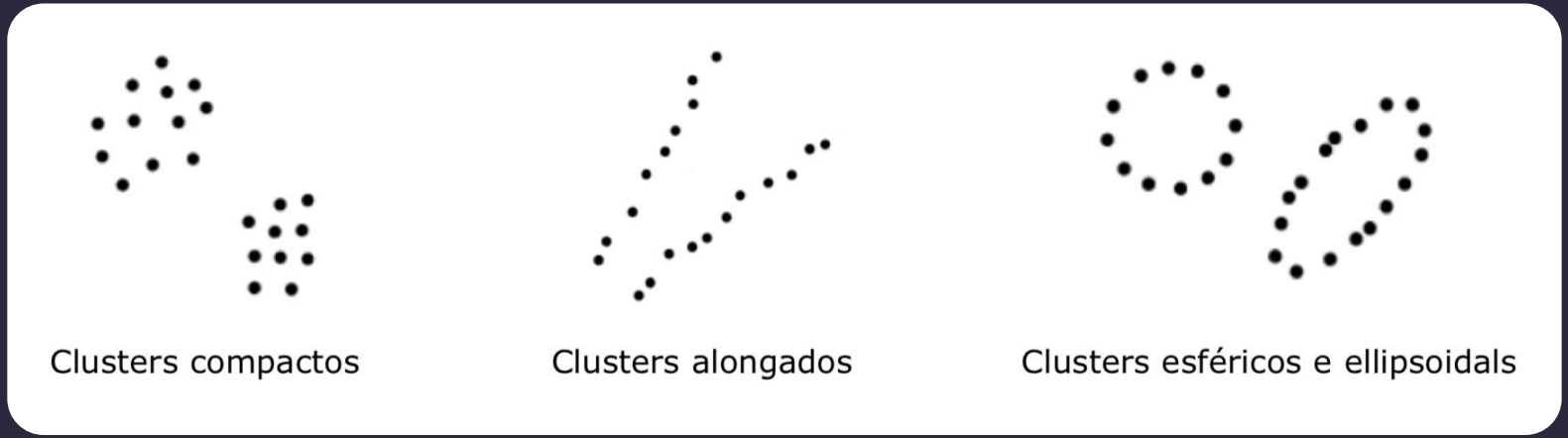
Dado um conjunto de dados X: $X = \{x_1, x_2, \ldots, x_n\}$

Definimos como um m–agrupamento de X a partição de X em m conjuntos (clusters ou grupos) C_1, C_2, ... , C_m tal que as três condições seguintes sejam satisfeitas:

- Nenhum cluster pode ser vazio ($C_i \neq \emptyset$).
- A união de todos os cluster deve ser igual ao conjunto de dados que gerou os clusters, ou seja, X.
- A interseção de dois clusters deve ser vazio, i.e., dois clusters não podem conter vetores em comum ($C_i \cap C_j = \emptyset$).

CLUSTERIZAÇÃO

- Os vetores contidos em um cluster Ci devem ser mais similares uns aos outros e menos similares aos vetores presentes nos outros clusters.
- Tipos de Clusters:



ALGORITMOS DE CLUSTERING

- Os algoritmos de clusterização buscam identificar padrões existentes em conjuntos de dados.
- Os algoritmos de clusterização podem ser divididos em várias categorias:
  - Sequenciais;
  - Hierárquicos;
  - Baseados na otimização de funções custo;
  - Outros: Fuzzy, SOM, LVQ ...

ALGORITMOS DE CLUSTERING

- São algoritmos diretos e rápidos.
- Geralmente, todos os vetores de características são apresentados ao algoritmo uma ou várias vezes (até 5 ou 6 vezes).
- O resultado final geralmente depende da ordem de apresentação dos vetores de características.

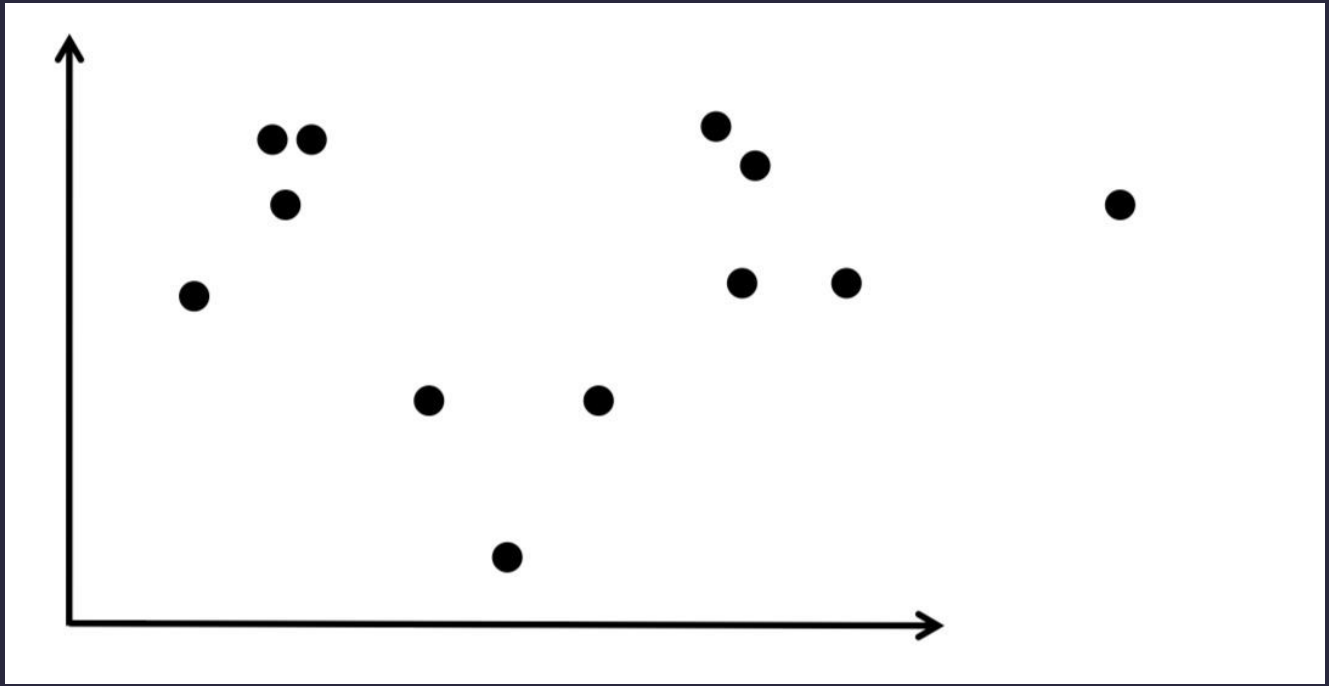
ALGORITMOS DE CLUSTERING

- Basic Sequential Algorithmic Scheme (BSAS)
  - Todos os vetores são apresentados uma única vez ao algoritmo.
  - Número de clusters não é conhecido inicialmente.
  - Novos clusters são criados enquanto o algoritmo evolui.

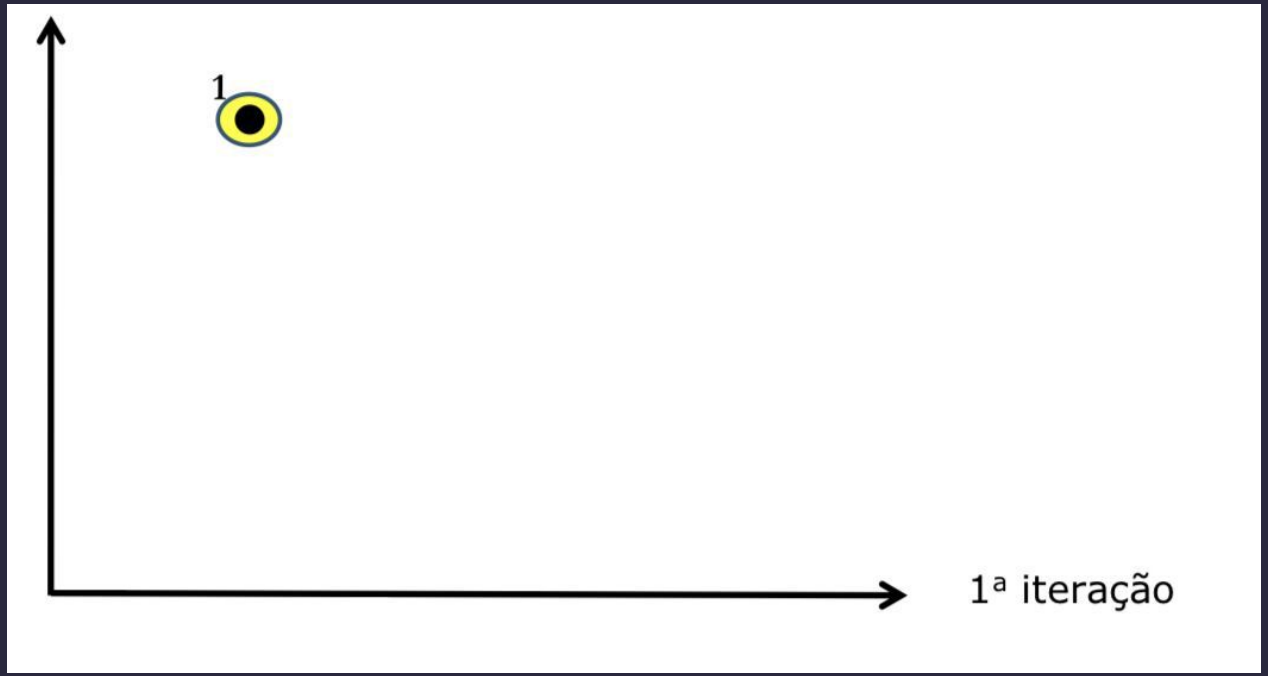
BASIC SEQUENTIAL ALGORITHMIC SCHEME (BSAS)

- Parâmetros do BSAS:
  - d(x, C): métrica de distância entre um vetor de características x e um cluster C.
  - θ: limiar de dissimilaridade.
  - q: número máximo de clusters.
- Idéia Geral do Algoritmo:
  - Para um dado vetor de características, designá-lo para um cluster existente ou criar um novo cluster (depende da distância entre o vetor e os clusters já formados).

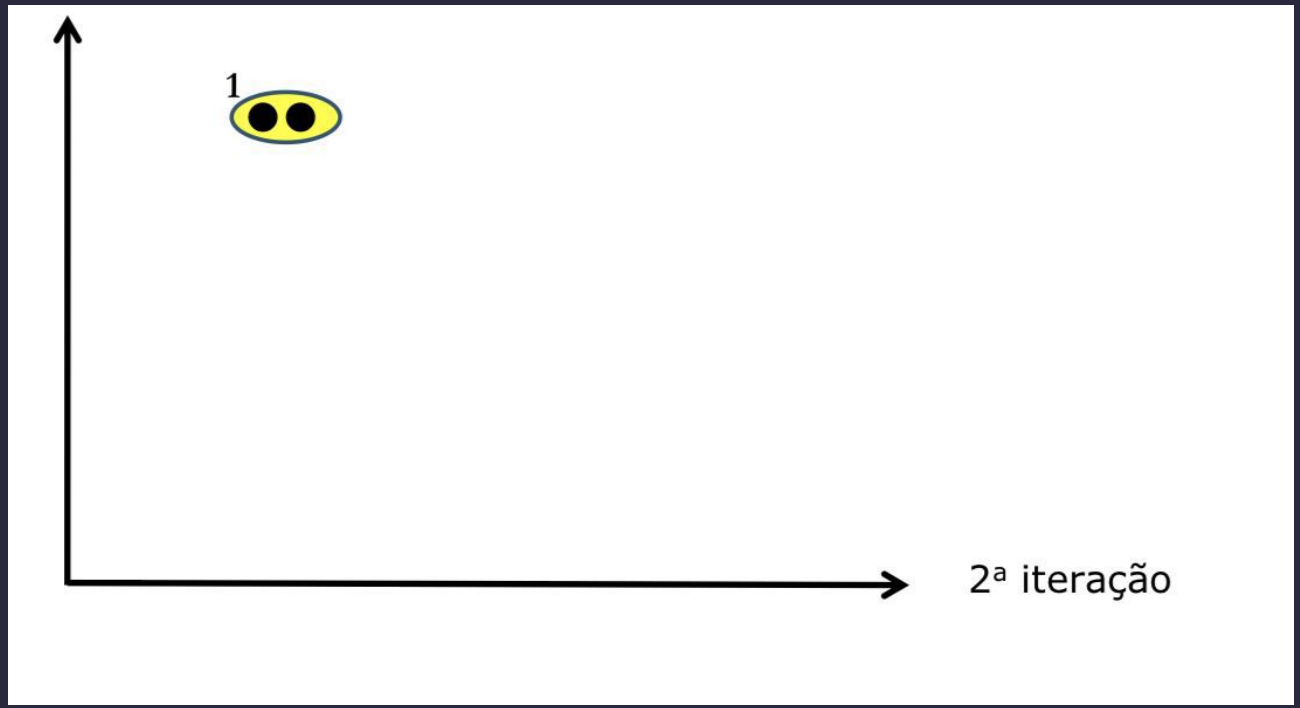
BASIC SEQUENTIAL ALGORITHMIC SCHEME (BSAS)



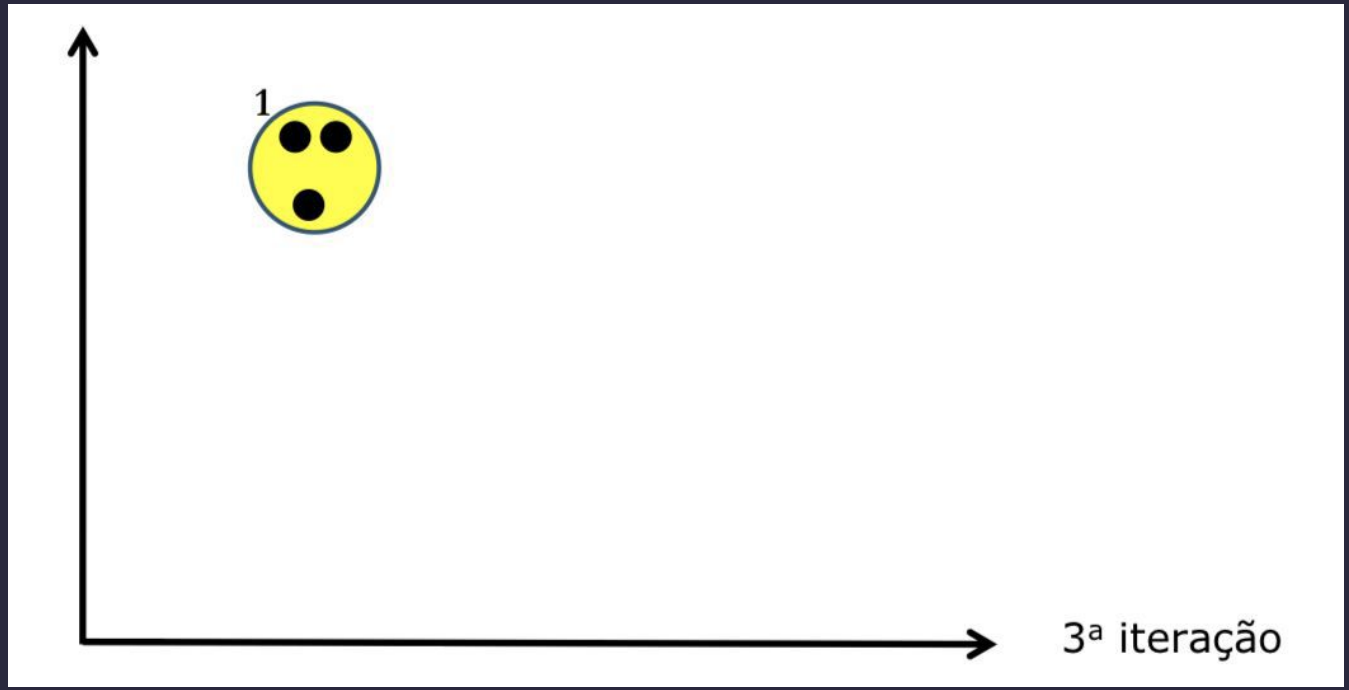
BASIC SEQUENTIAL ALGORITHMIC SCHEME (BSAS)



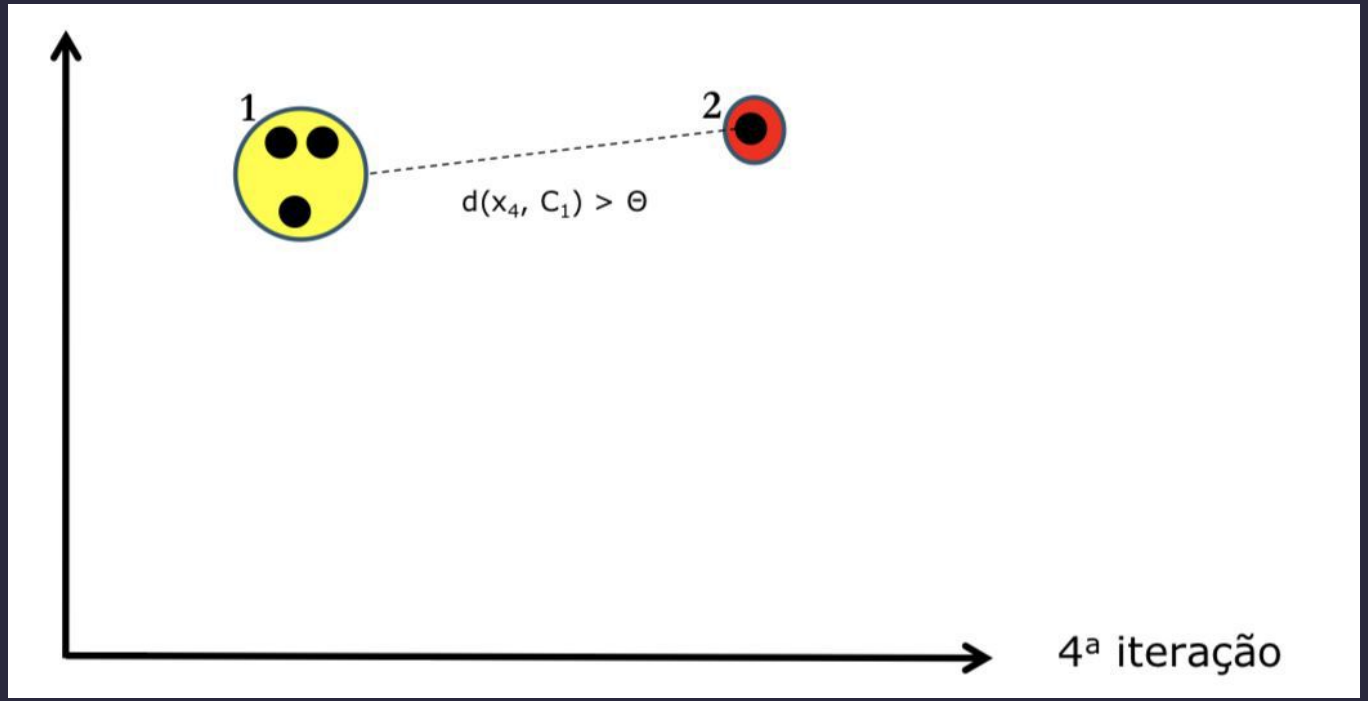
BASIC SEQUENTIAL ALGORITHMIC SCHEME (BSAS)



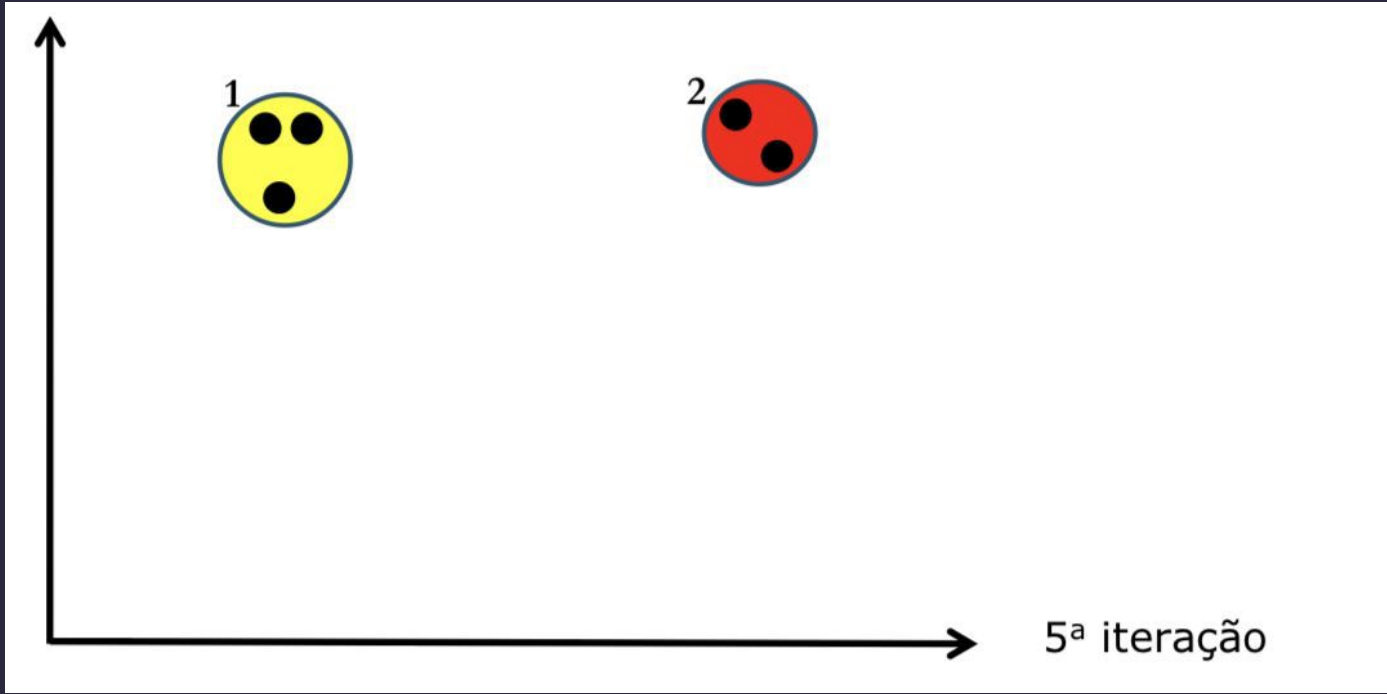
BASIC SEQUENTIAL ALGORITHMIC SCHEME (BSAS)



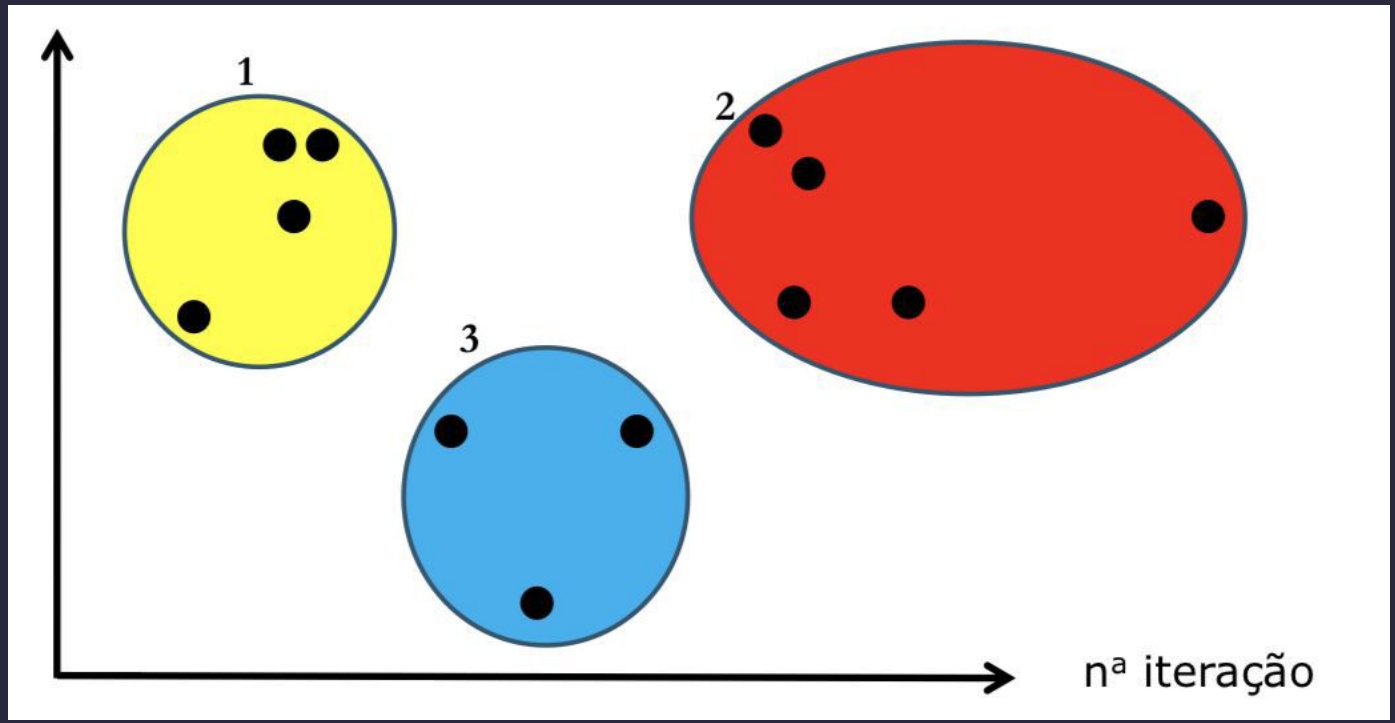
BASIC SEQUENTIAL ALGORITHMIC SCHEME (BSAS)



BASIC SEQUENTIAL ALGORITHMIC SCHEME (BSAS)



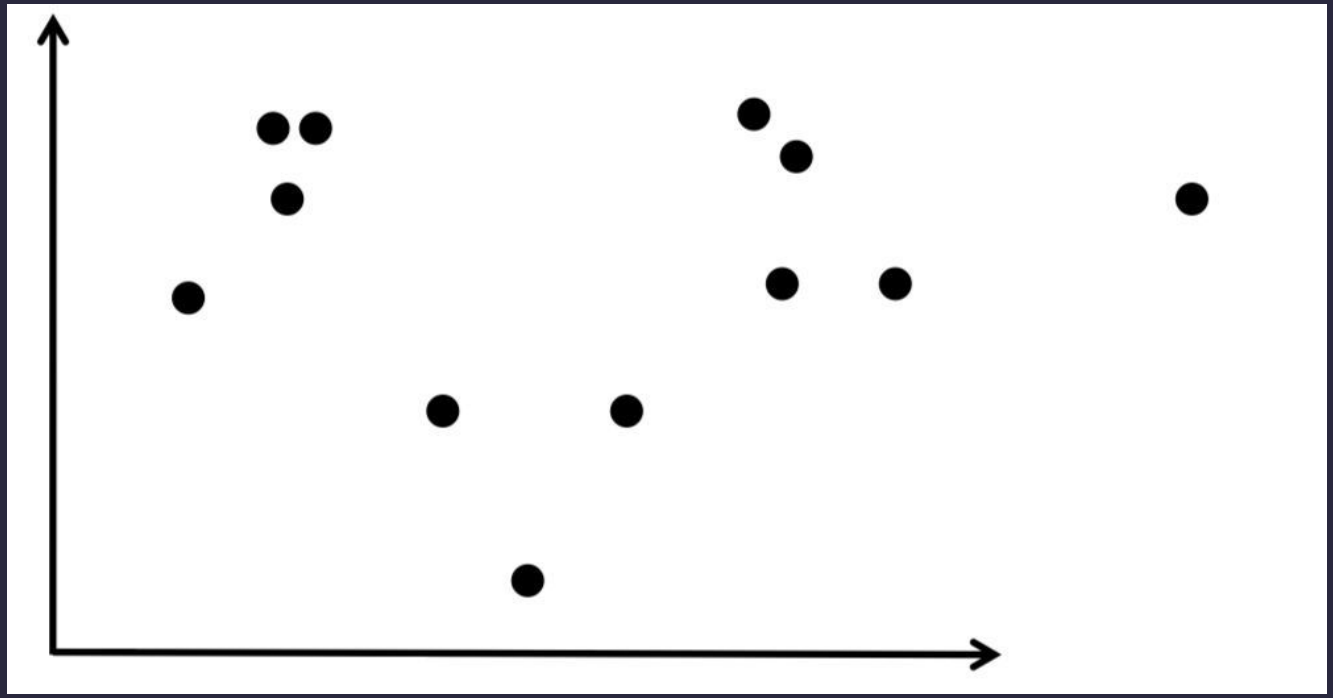
BASIC SEQUENTIAL ALGORITHMIC SCHEME (BSAS)



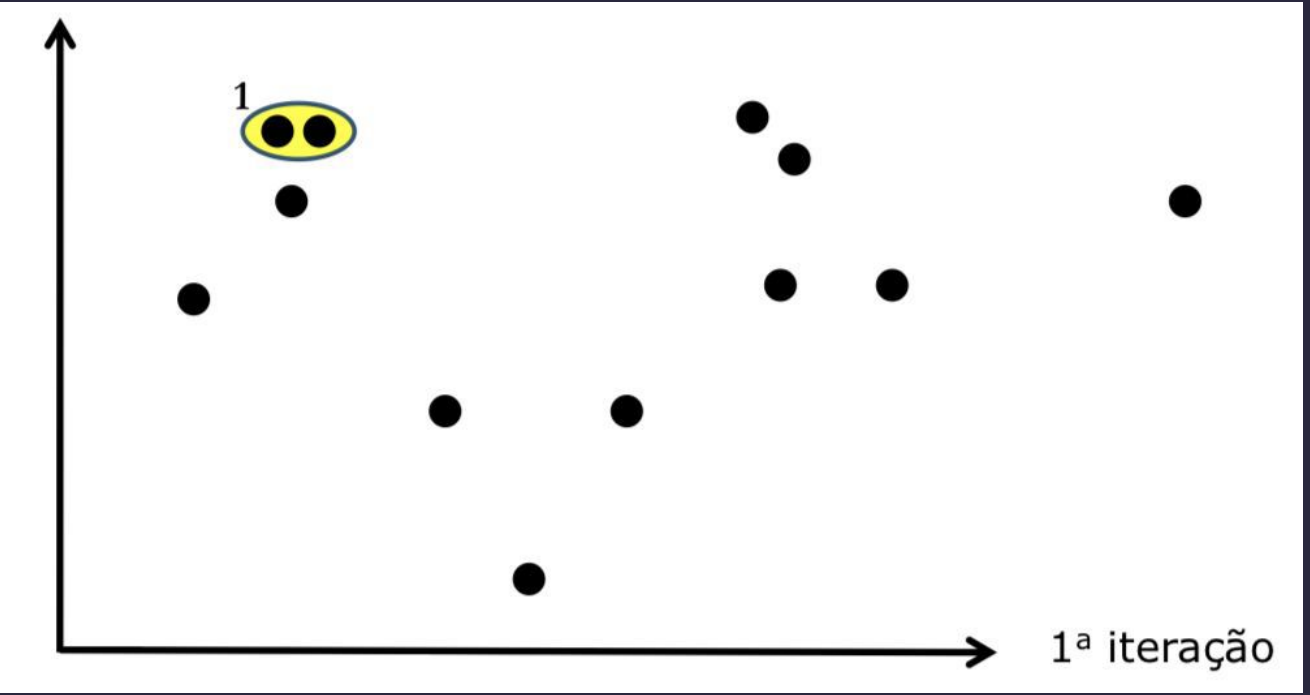
CLUSTERIZAÇÃO HIERÁRQUICA

- Os algoritmos de clusterização hierárquica pode ser divididos em 2 subcategorias:
- Aglomerativos:
  - Produzem uma sequência de agrupamentos com um número decrescente de clusters a cada passo.
  - Os agrupamentos produzidos em cada passo resultam do anterior pela fusão de dois clusters em um.
- Divisivos:
  - Atuam na direção oposta, isto é, eles produzem uma sequência de agrupamentos com um número crescente de clusters a cada passo.
  - Os agrupamentos produzidos em cada passo resultam da partição de um único cluster em dois.

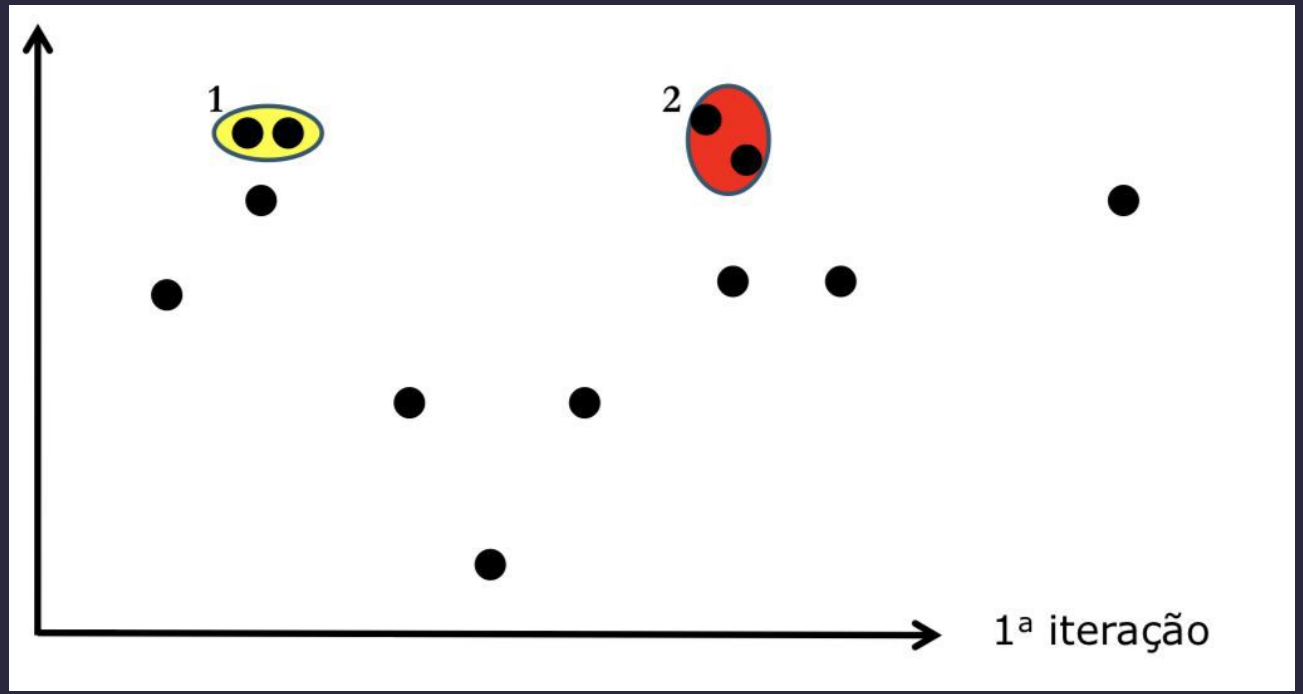
CLUSTERIZAÇÃO HIERÁRQUICA - AGLOMERATIVOS



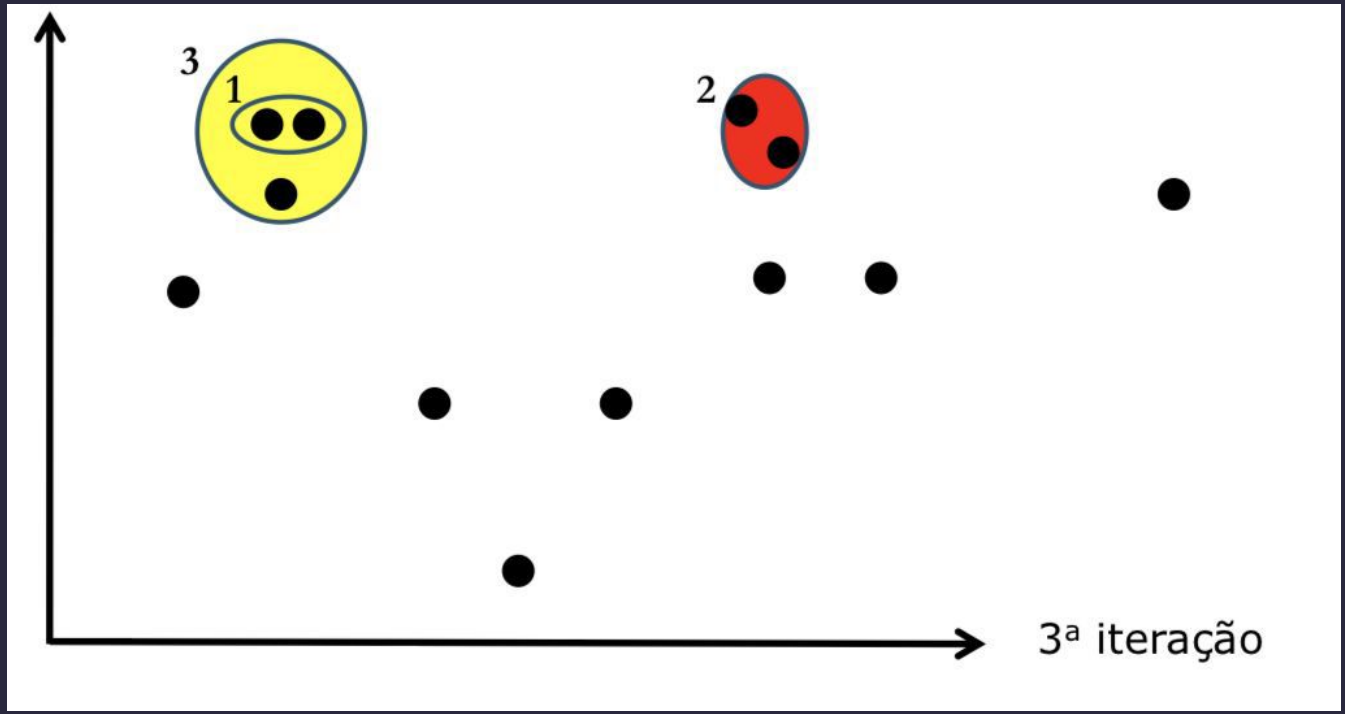
CLUSTERIZAÇÃO HIERÁRQUICA - AGLOMERATIVOS



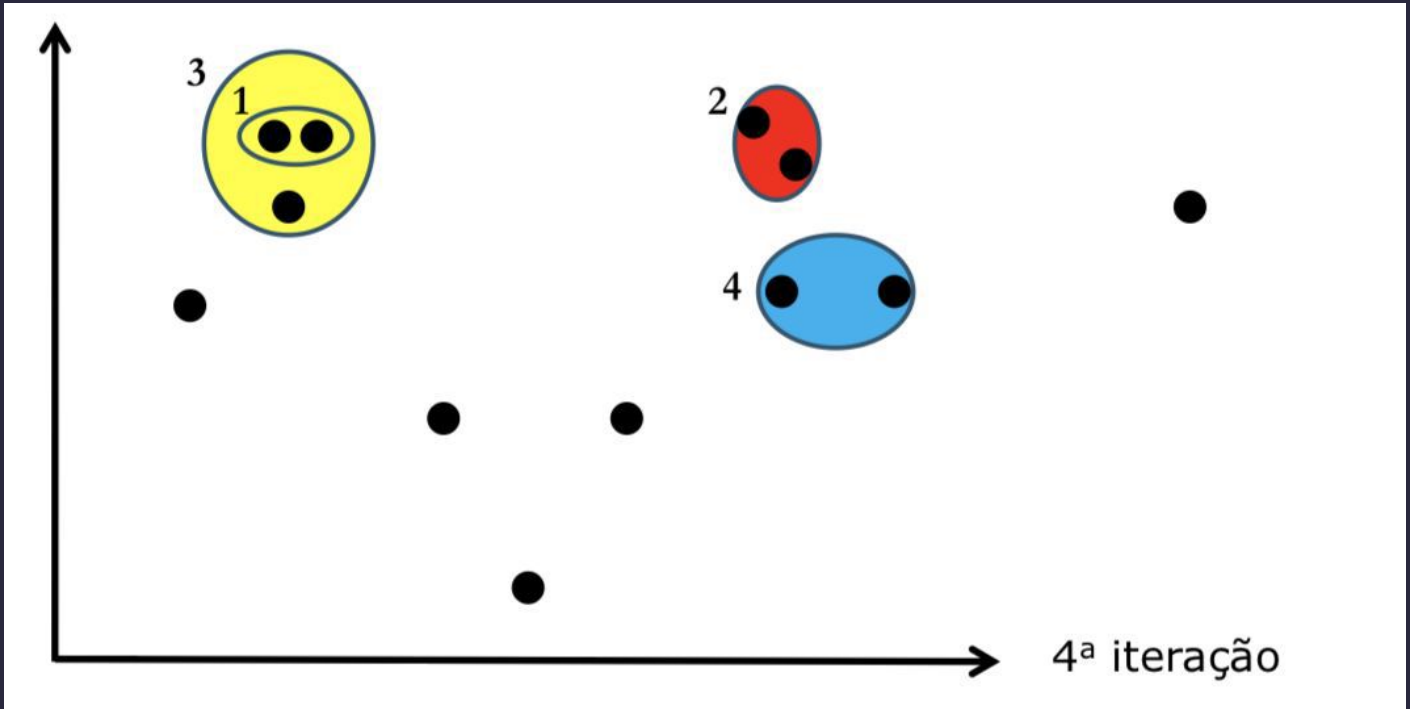
CLUSTERIZAÇÃO HIERÁRQUICA - AGLOMERATIVOS



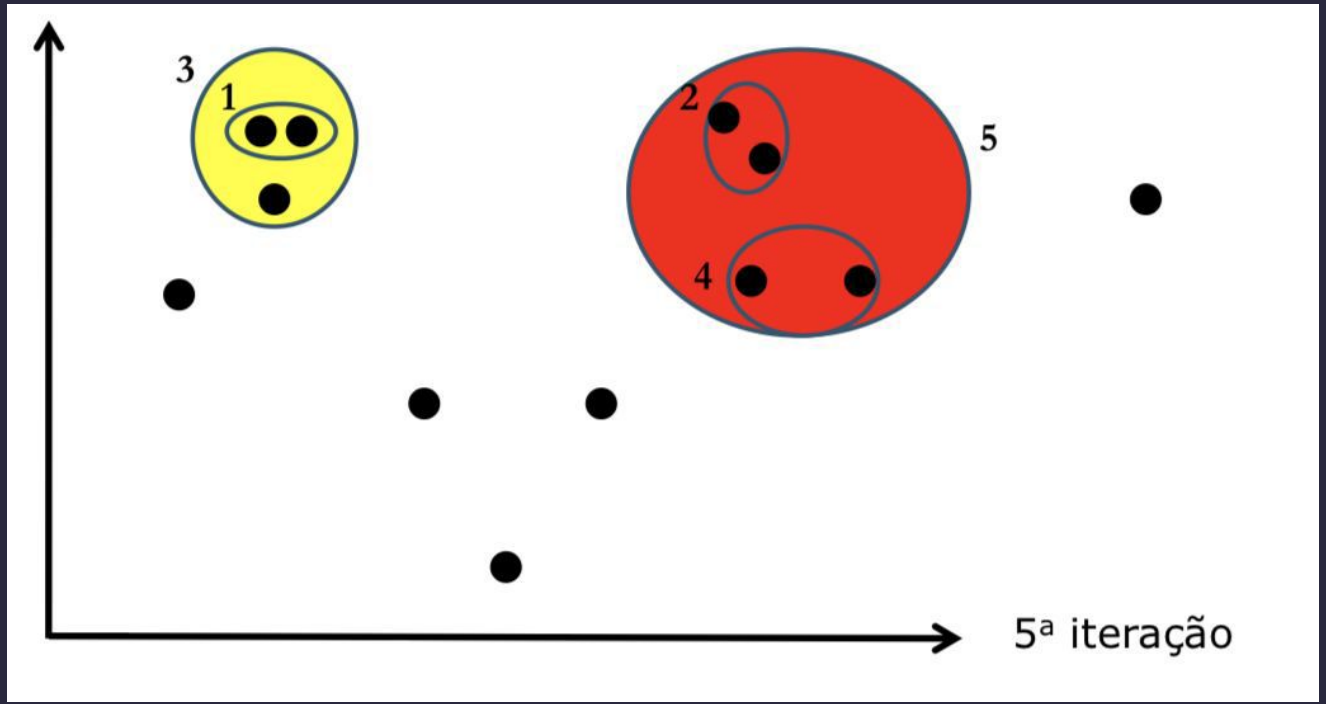
CLUSTERIZAÇÃO HIERÁRQUICA - AGLOMERATIVOS



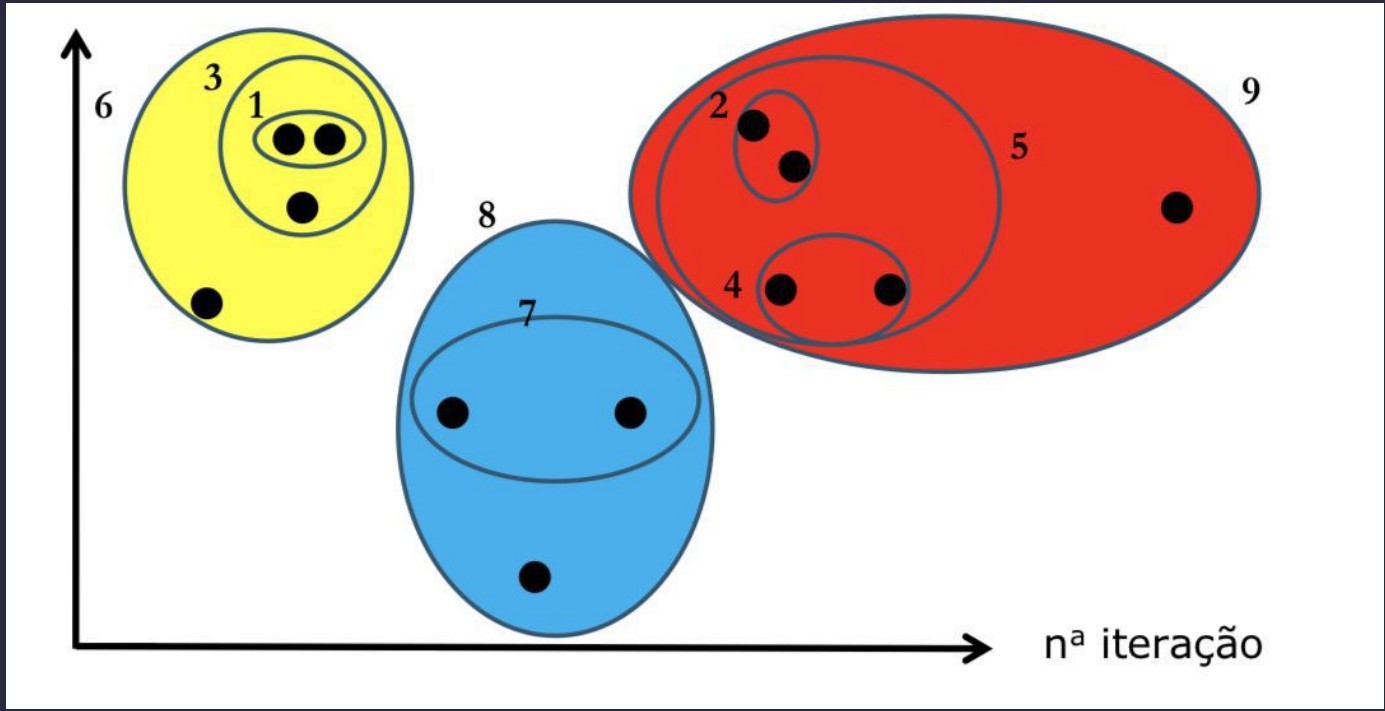
CLUSTERIZAÇÃO HIERÁRQUICA - AGLOMERATIVOS



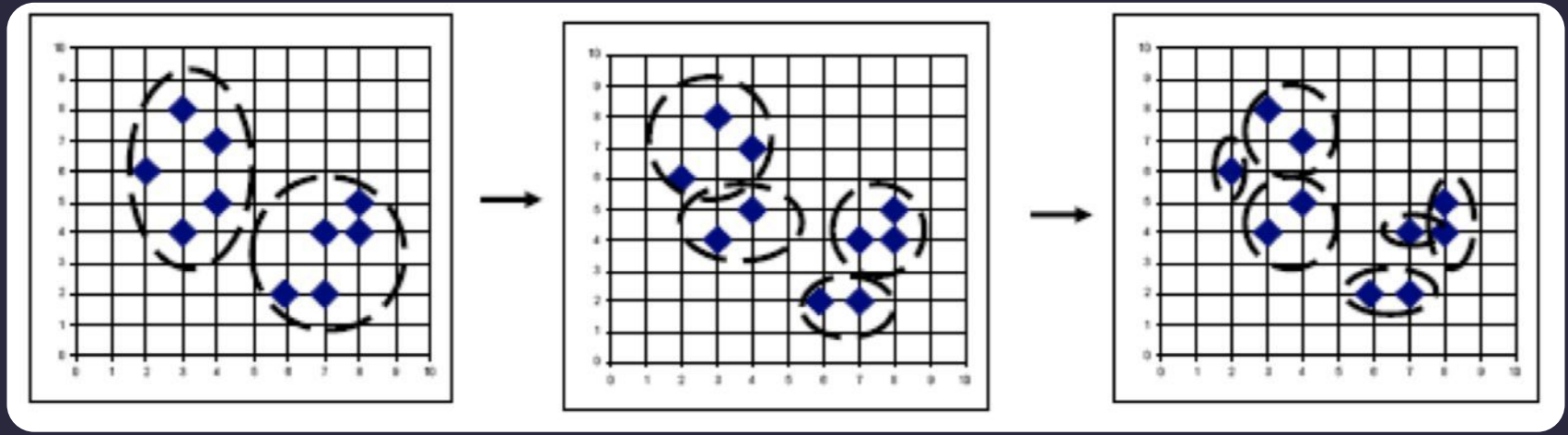
CLUSTERIZAÇÃO HIERÁRQUICA - AGLOMERATIVOS



CLUSTERIZAÇÃO HIERÁRQUICA - AGLOMERATIVOS



CLUSTERIZAÇÃO HIERÁRQUICA - DIVISIVO



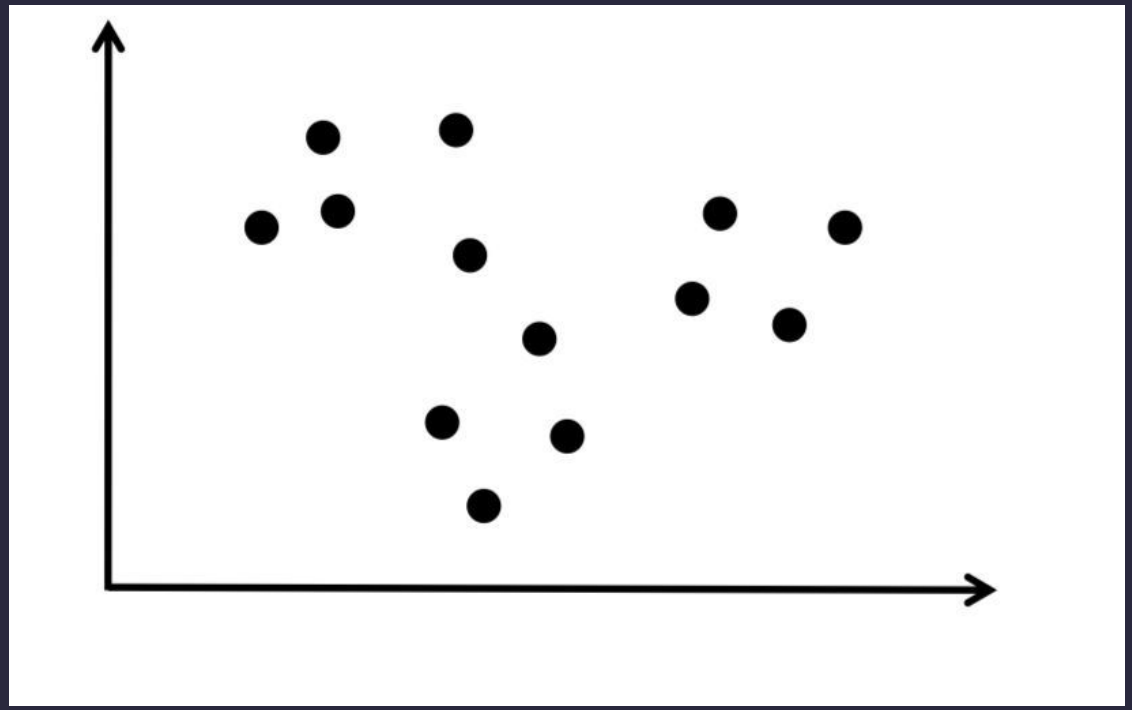
K-MEANS

- É a técnica mais simples de aprendizagem não supervisionada.
- Consiste em fixar k centróides (de maneira aleatória), um para cada grupo (clusters).
  - Há diversas estratégias para definir o número ideal de centróides
- Associar cada indivíduo ao seu centróide mais próximo.
- Recalcular os centróides com base nos indivíduos classificados.

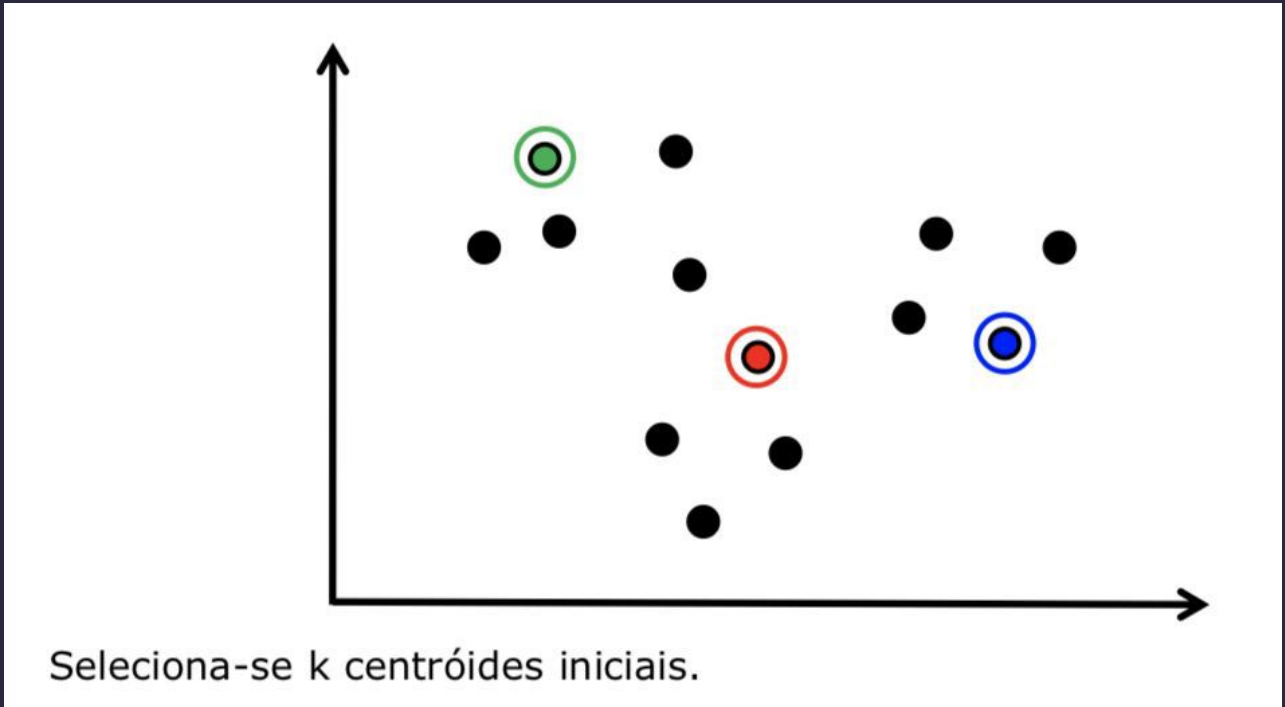
ALGORITMO K-MEANS

1. Selecione k centróides iniciais.
2. Forme k clusters associando cada exemplo ao seu centróide mais próximo.
3. Recalcule a posição dos centróides com base no centro de gravidade do cluster.
4. Repita os passos 2 e 3 até que os centróides não sejam mais movimentados.

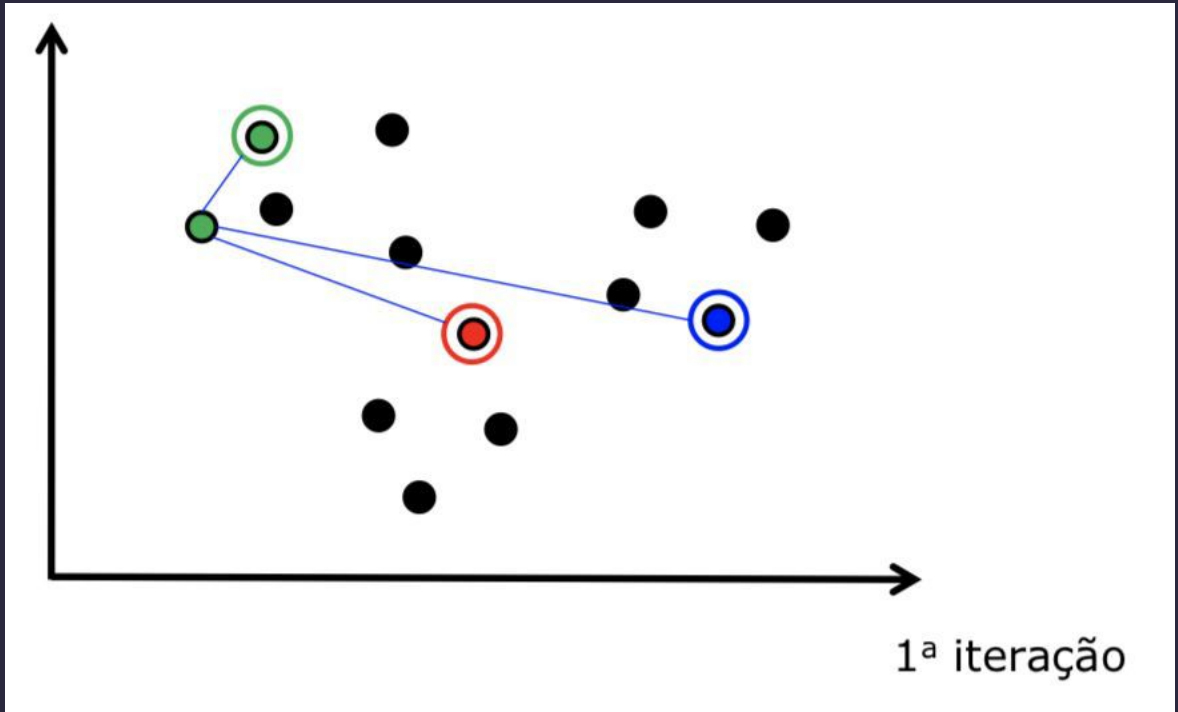
ALGORITMO K-MEANS



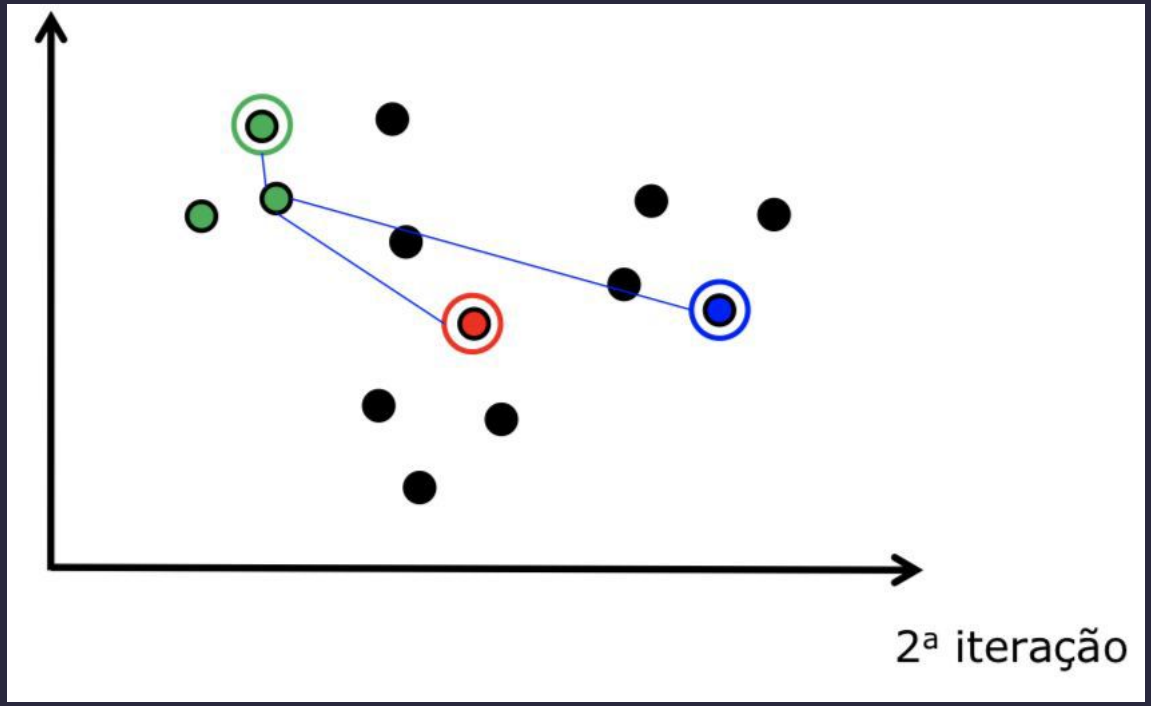
ALGORITMO K-MEANS



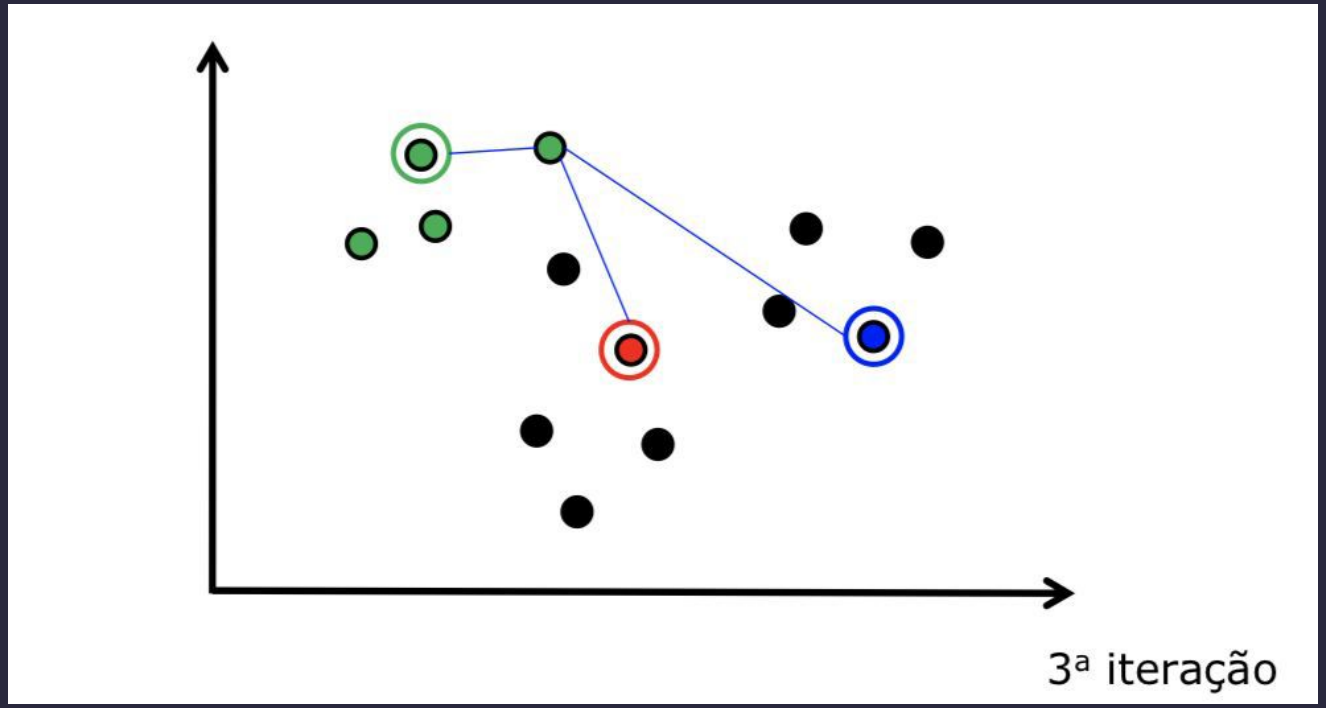
ALGORITMO K-MEANS



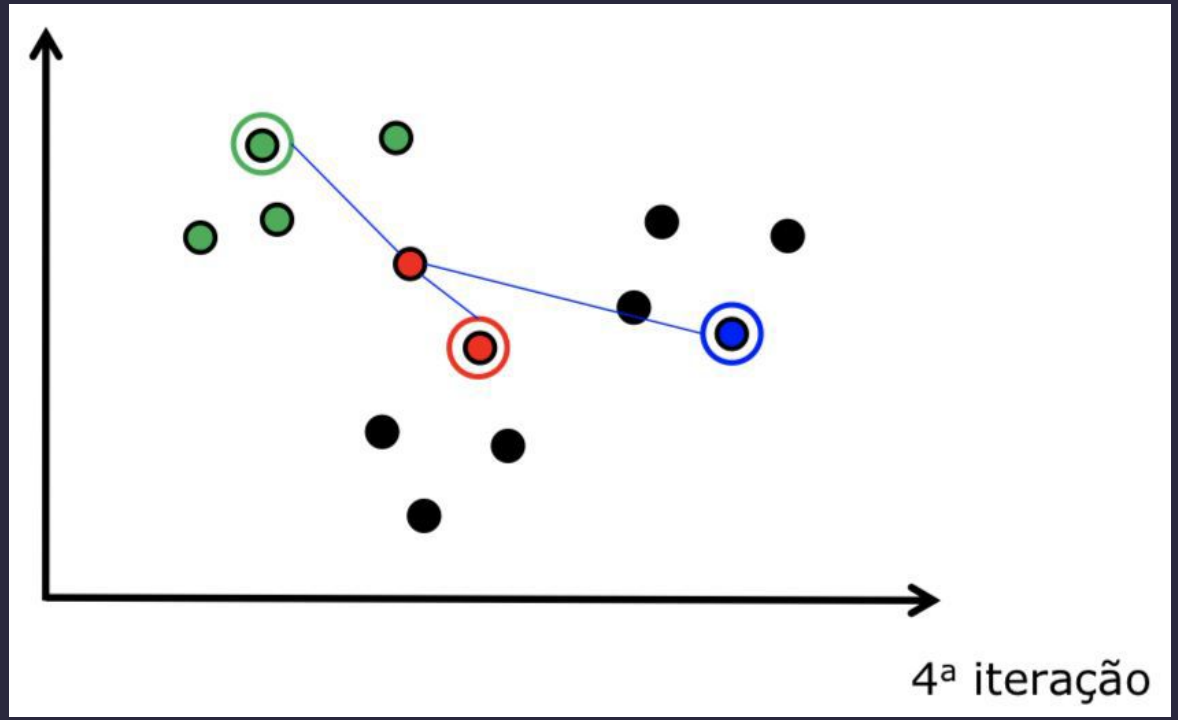
ALGORITMO K-MEANS



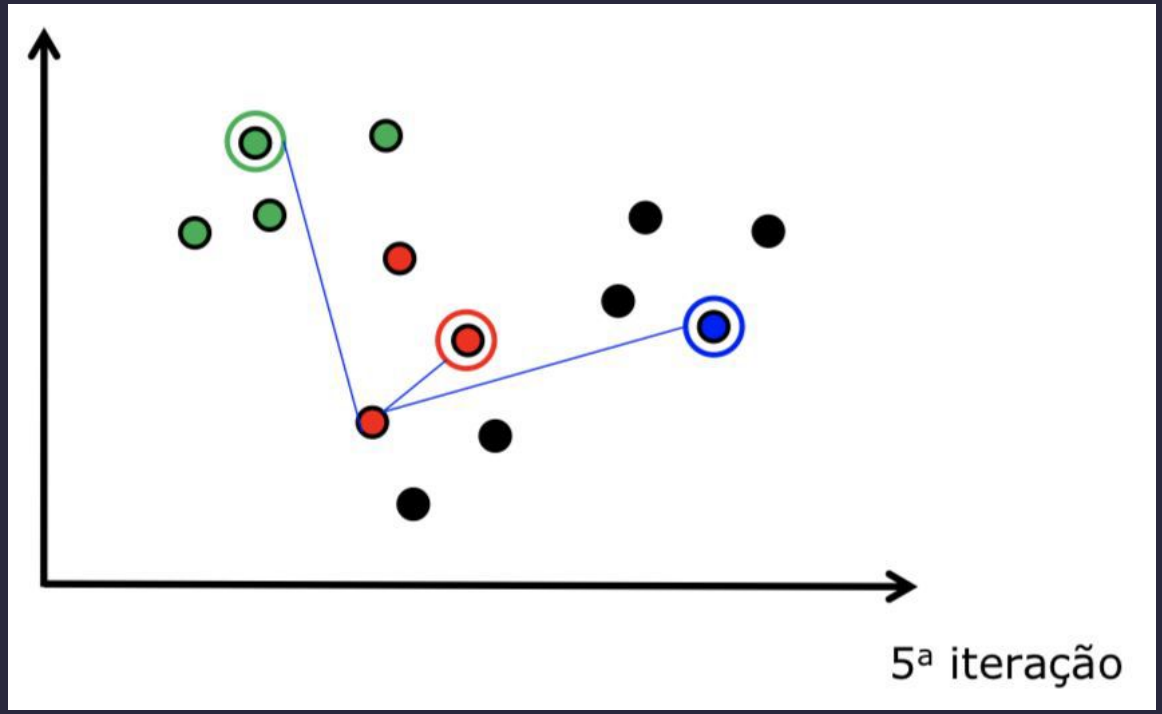
ALGORITMO K-MEANS



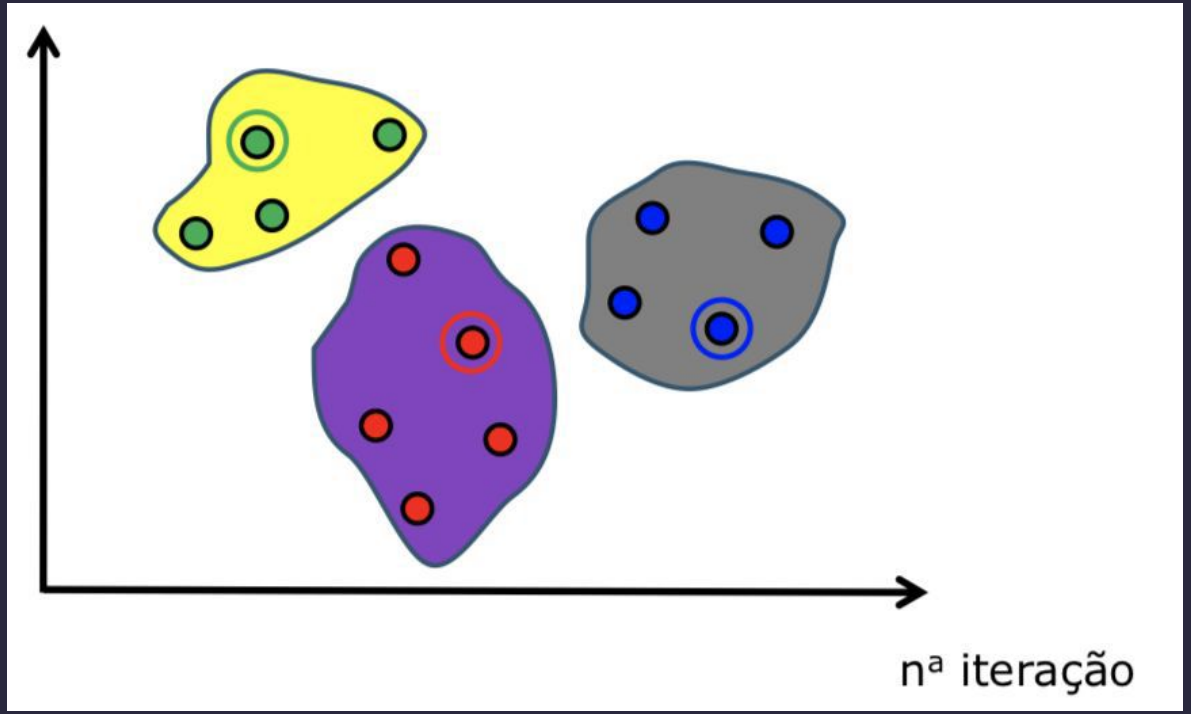
ALGORITMO K-MEANS



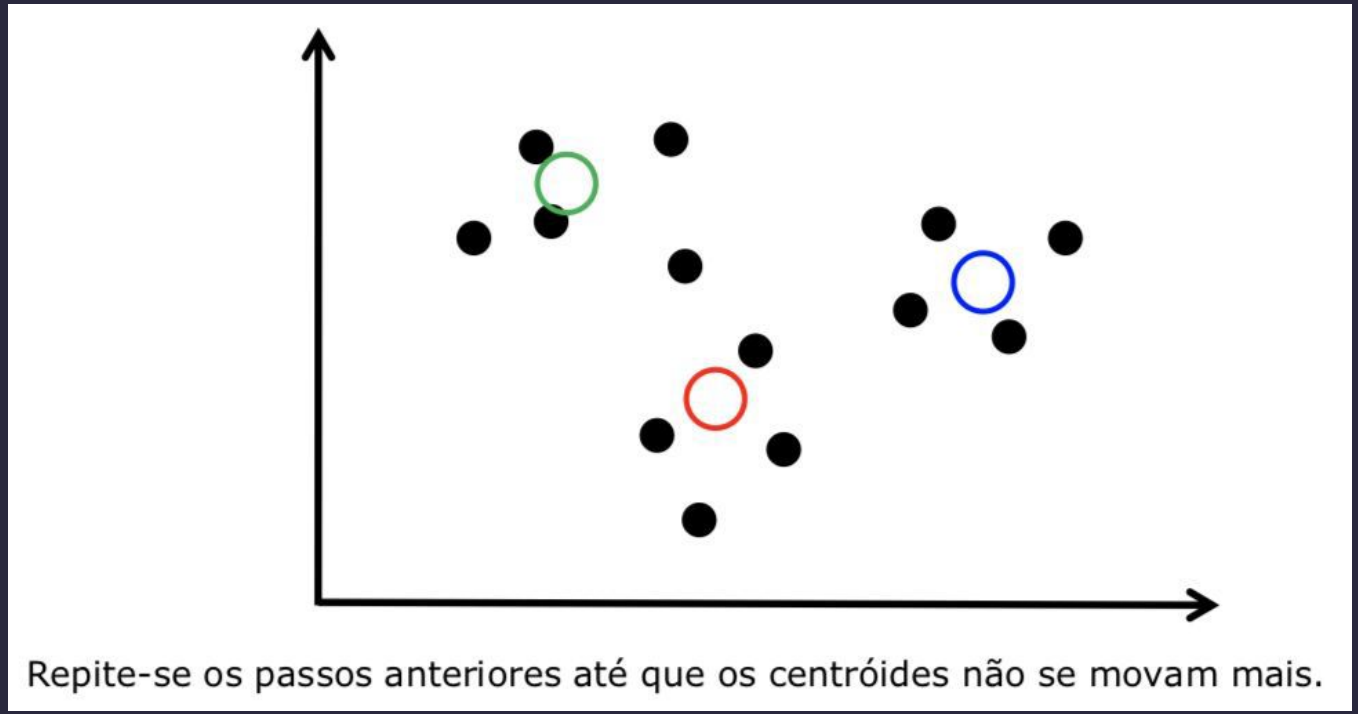
ALGORITMO K-MEANS



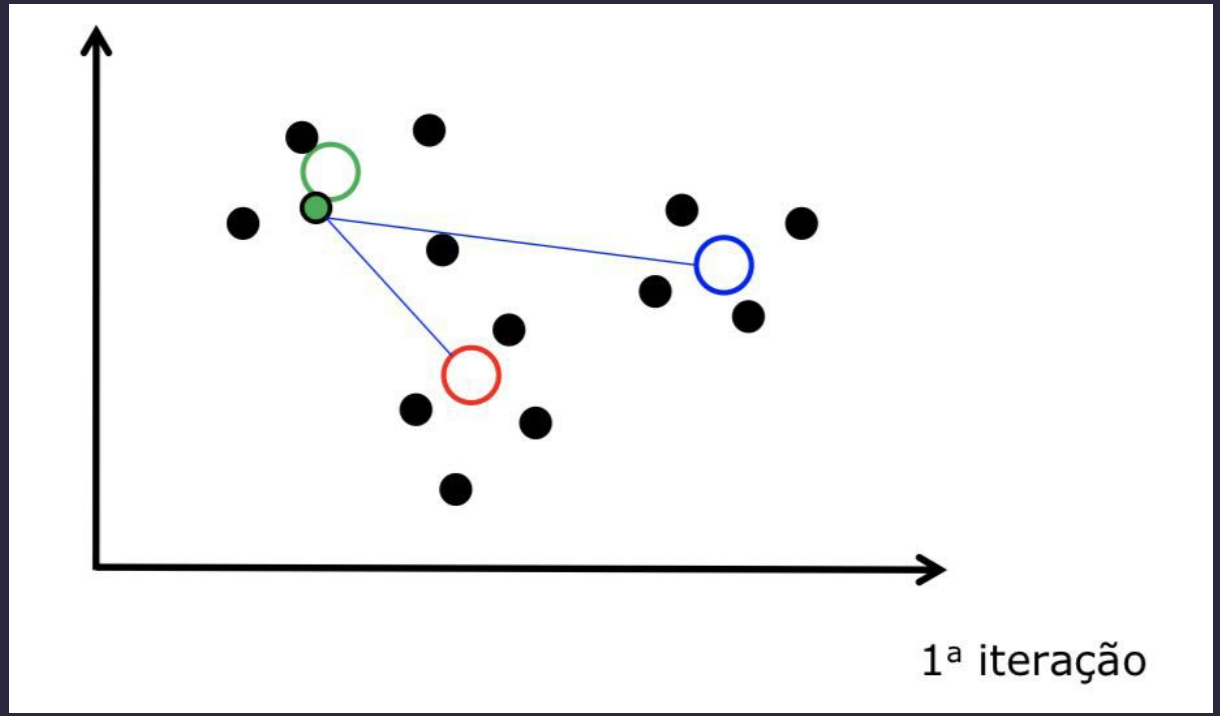
ALGORITMO K-MEANS



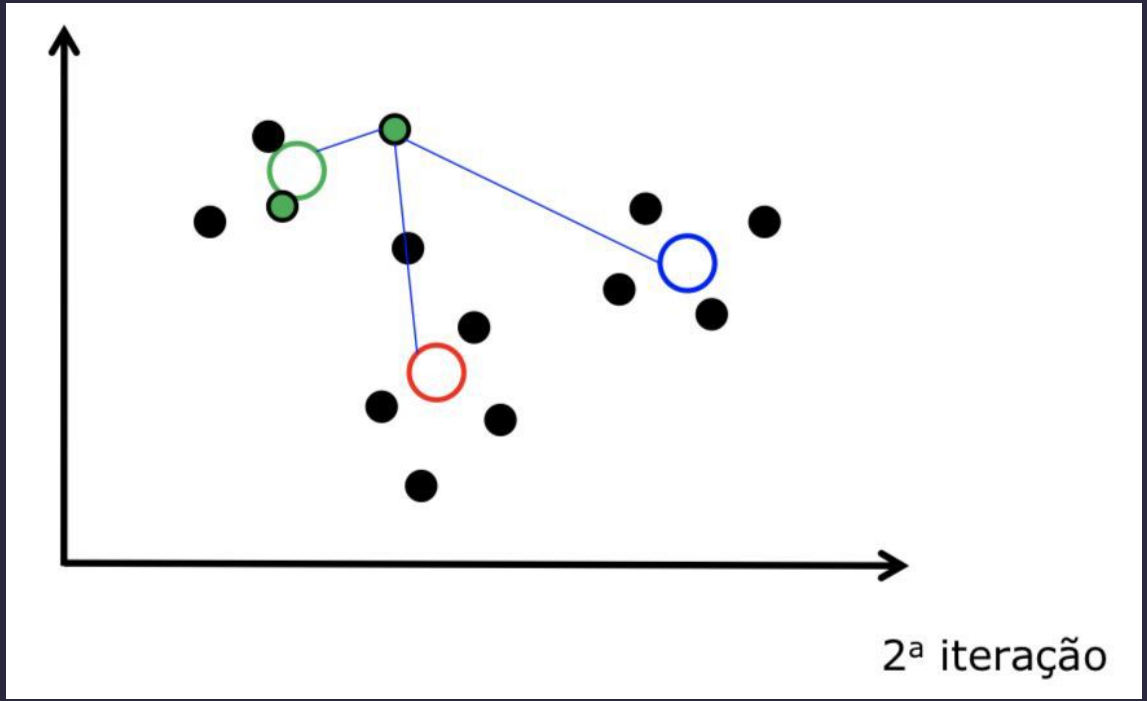
ALGORITMO K-MEANS



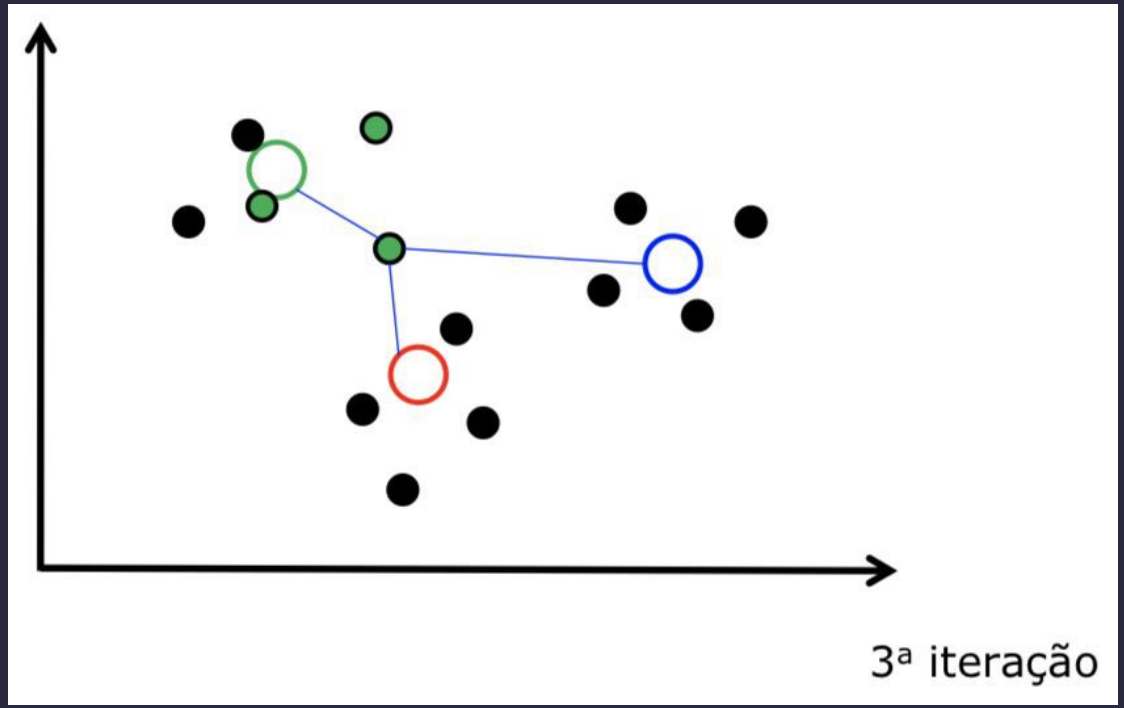
ALGORITMO K-MEANS



ALGORITMO K-MEANS

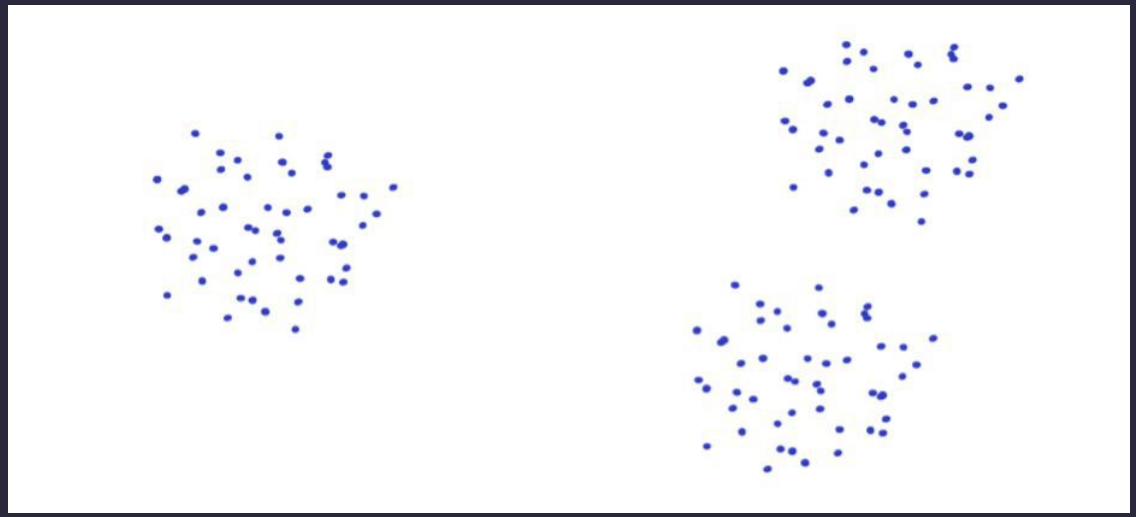


ALGORITMO K-MEANS



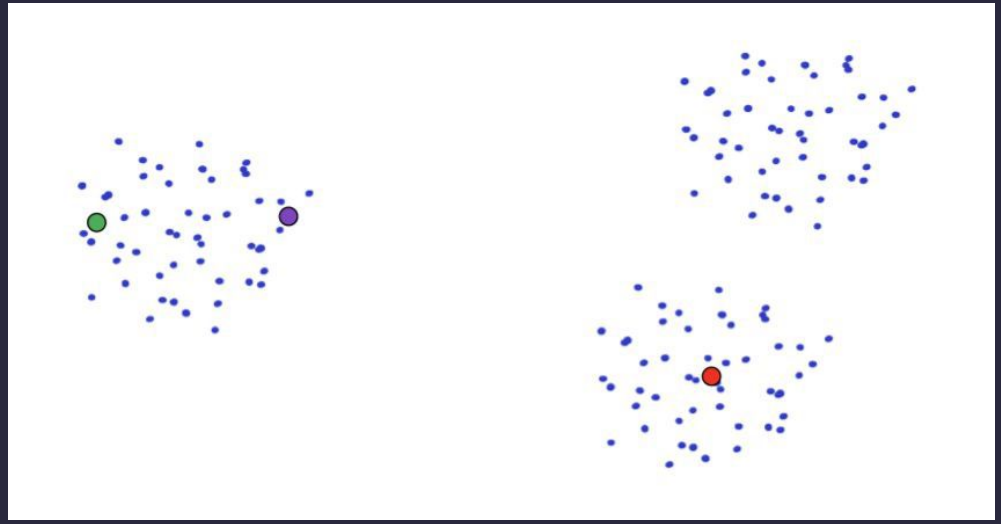
PROBLEMAS DO K-MEANS

O principal problema do K-Means é a dependência de uma boa inicialização.



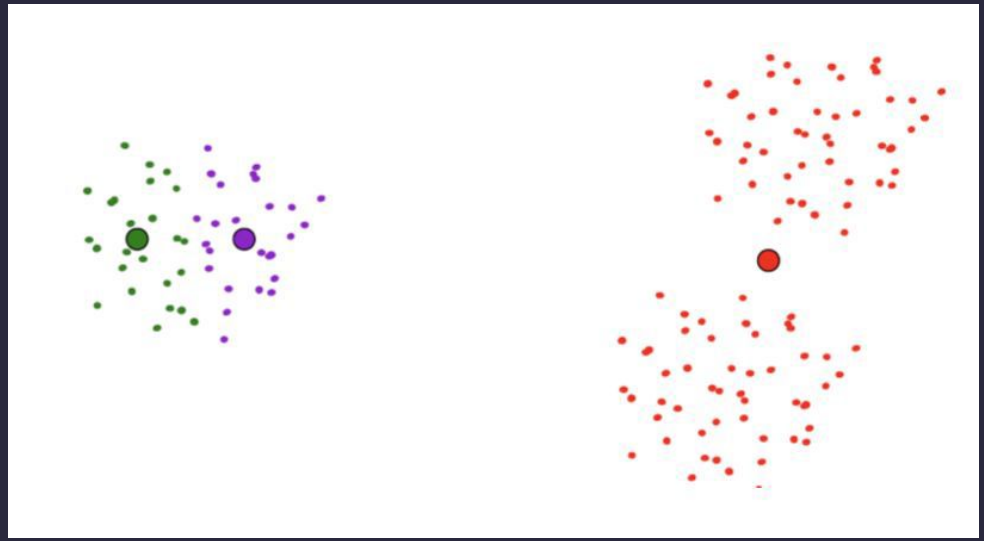
PROBLEMAS DO K-MEANS

O principal problema do K-Means é a dependência de uma boa inicialização.



PROBLEMAS DO K-MEANS

O principal problema do K-Means é a dependência de uma boa inicialização.



APRENDIZADO NÃO-SUPERVISIONADO

- O aprendizado não-supervisionado ou clusterização (agrupamento) busca extrair informação relevante de dados não rotulados.
- Existem vários algoritmos agrupamento de dados.
- Diferentes escolhas de atributos, medidas de proximidade, critérios de agrupamento e algoritmos de clusterização levam a resultados totalmente diferentes.

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large `n_samples`, medium `n_clusters` with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with `n_samples`	Many clusters, uneven cluster size, non-flat geometry	Distances between points
Spectral clustering	number of clusters	Medium `n_samples`, small `n_clusters`	Few clusters, even cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters or distance threshold	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints	Distances between points
Agglomerative clustering	number of clusters or distance threshold, linkage type, distance	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints, non Euclidean distances	Any pairwise distance
DBSCAN	neighborhood size	Very large `n_samples`, medium `n_clusters`	Non-flat geometry, uneven cluster sizes	Distances between nearest points
OPTICS	minimum cluster membership	Very large `n_samples`, large `n_clusters`	Non-flat geometry, uneven cluster sizes, variable cluster density	Distances between points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation	Mahalanobis distances to centers
Birch	branching factor, threshold, optional global clusterer.	Large `n_clusters` and `n_samples`	Large dataset, outlier removal, data reduction.	Euclidean distance between points

EXEMPLO

NEAREST NEIGHBORS

Problema: Identifique se existem tipos de flor de íris de acordo com informações relacionadas ao tamanho da pétala e sépala da flor.

Fonte:

http://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_iris.html

Bibliotecas Principais: SKLearn e Yellowbrick

(https://www.scikit-yb.org/en/latest/index.html)

NEAREST NEIGHBORS

```
from sklearn.neighbors import NearestNeighbors
import numpy as np

X = np.array([[-1, -1], [-2, -1], [-3, -2], [1, 1], [2, 1], [3, 2]])
nbrs = NearestNeighbors(n_neighbors=2, algorithm='ball_tree').fit(X)
distances, indices = nbrs.kneighbors(X)
```

MACHINE LEARNING: CLUSTERING