

DATA SCIENCE

AULA 5 - Aprendizagem Não- supervisionada

Prof. Gabriel Resende Machado



gabrielmachado@unifeso.edu.com

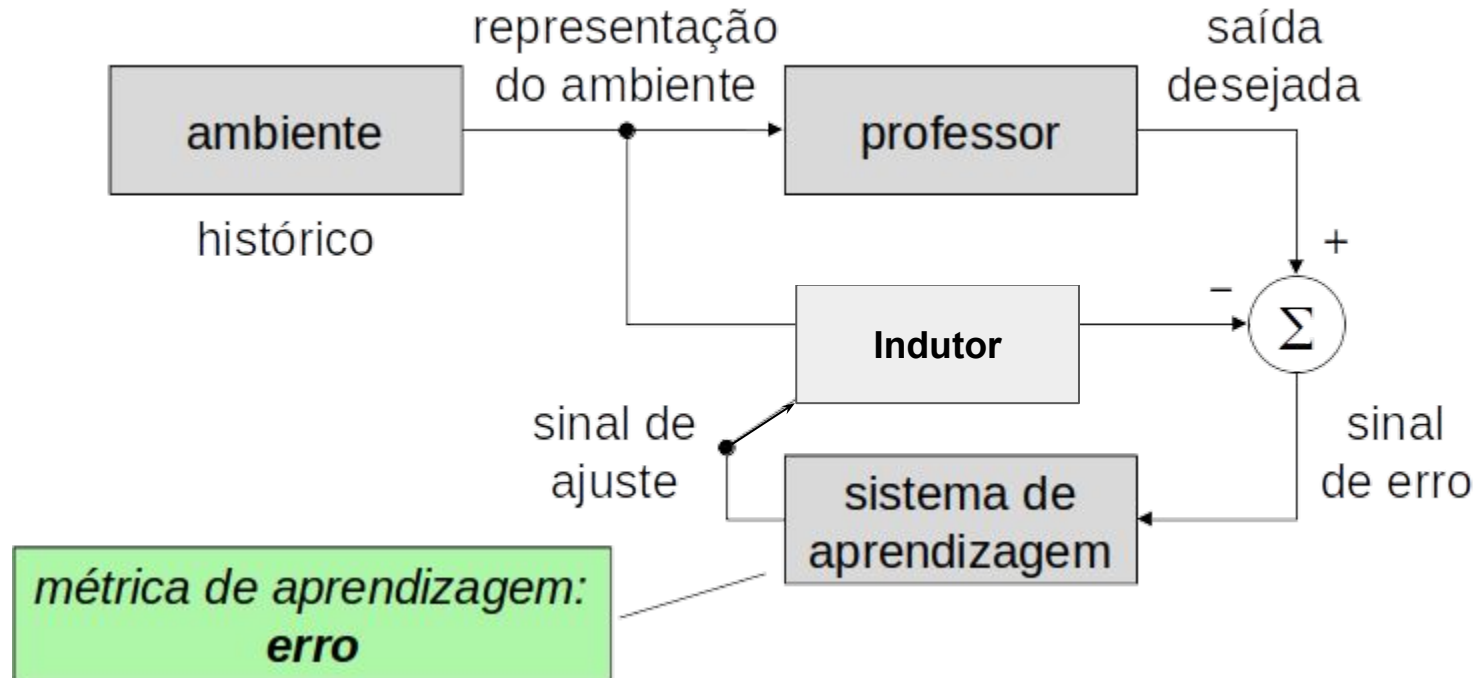


<https://www.linkedin.com/in/machadogabriel>

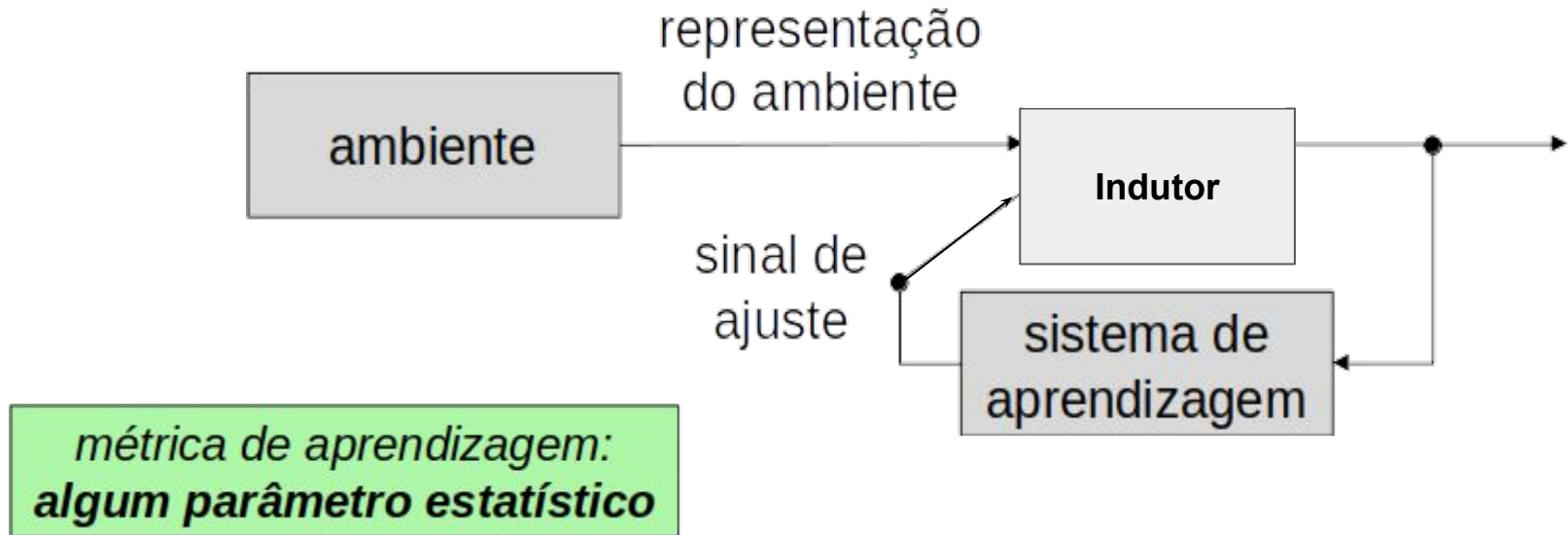


<https://github.com/UNIFESO-Gabriel/data-science>

Aprendizado Supervisionado (revisão)



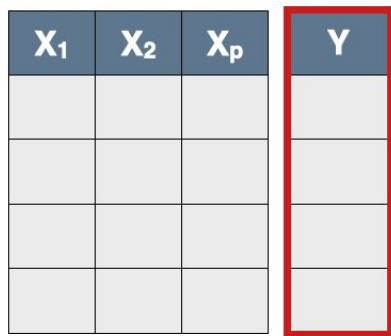
Aprendizado Não-supervisionado



Aprendizado Não-supervisionado

Entradas do modelo.
Ex.: peso, altura,
idade, volume, etc

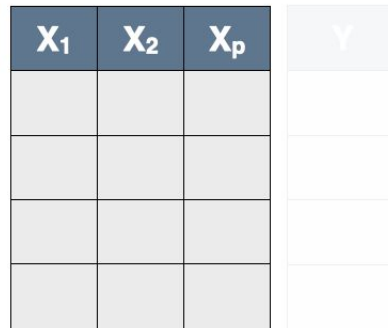
Supervisionado



X ₁	X ₂	X _p	Y

O alvo é aprender a prever Y

Não Supervisionado



X ₁	X ₂	X _p	Y

O objetivo é aprender a diferenciar ou
representar os dados

Aprendizado Não Supervisionado

- **Não exige classificação prévia dos dados** em diferentes categorias;
- Depende **apenas** dos atributos descritivos (não do alvo);
- **Não há exemplos rotulados** da função a ser induzida;

Metodologia baseada em **Aprendizado de Máquina**, na qual o algoritmo de aprendizado não recebe *feedback* do ambiente a respeito da saída desejada para o atributo-alvo.

Tarefas do Aprendizado Não-supervisionado

- **Associação:** avalia o nível de associação entre dados ou conjuntos de dados.
 - Exemplos de algoritmos incluem o ***Apriori***, *FP-Growth*, *Eclat*, entre outros;
- Exemplos de aplicação: Sistemas de recomendação, e-commerce, plataformas de *streaming*, *etc.*;
- **Sumarização:** Geração automática de sumários a partir de textos densos:
 - **Medicina:** análise de prontuários médicos;
 - **Educação:** aprendizado *online*;
 - **Pesquisa:** resumo de artigos científicos;
 - **Mídias audiovisuais, literatura, mercado financeiro, etc;**
- **Agrupamento:** Particiona o conjunto de dados em grupos de características semelhantes.
 - Itens em um mesmo grupo devem ser mais semelhantes (em geral) do que itens em grupos diferentes.

Tarefas do Aprendizado Não-supervisionado



**Algoritmo
aprende o
padrão**



Grupo 1



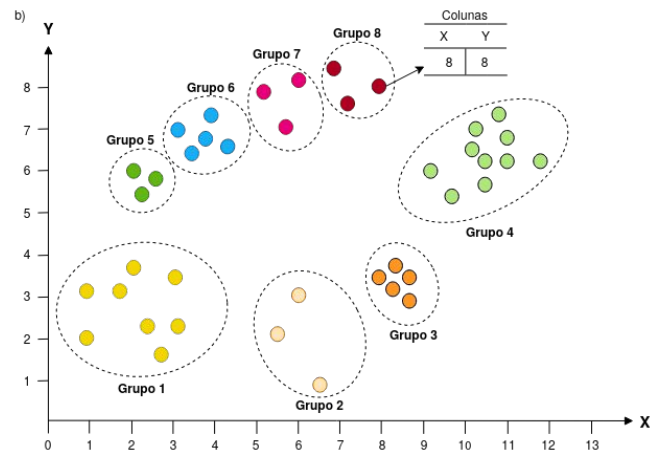
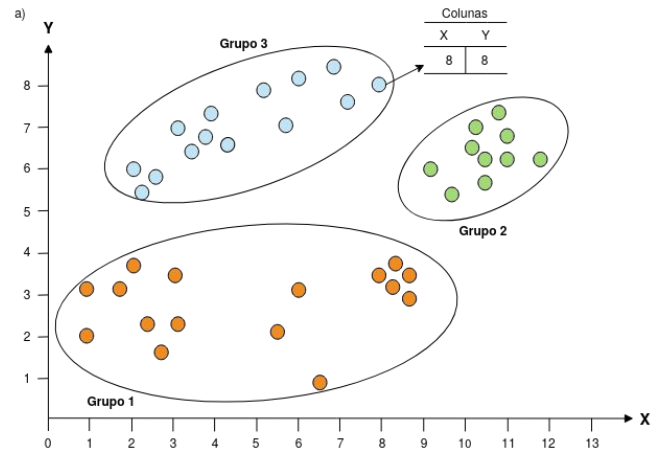
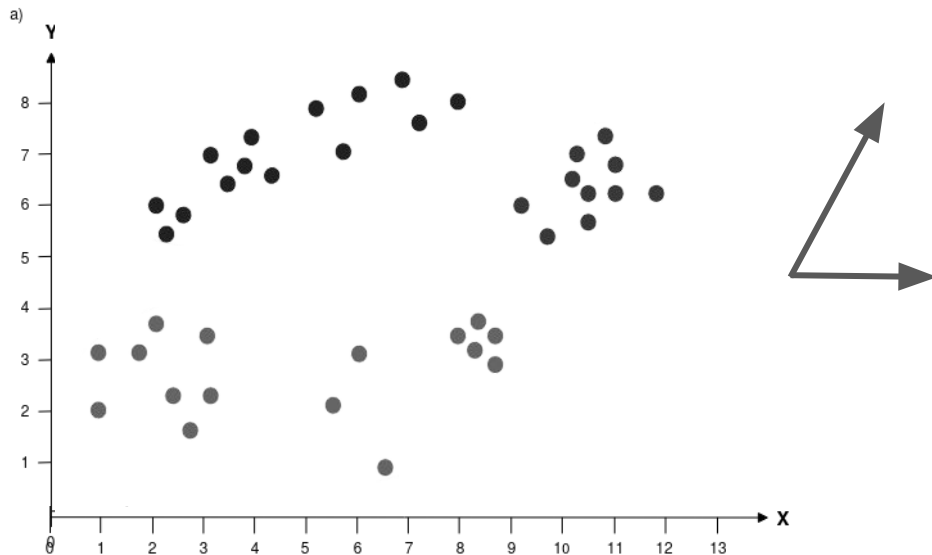
Grupo 2



Grupo 3

Agrupamento (*clustering*)

- **Agrupamento:** um mesmo conjunto de dados pode ser particionado em grupos diferentes.



O Algoritmo *k-Means*

- Algoritmo de **agrupamento particional** mais simples;
- **Particiona o conjunto de dados em $k > 0$ grupos**, onde k é um parâmetro fornecido como entrada;
- Utiliza um processo iterativo para encontrar uma partição com k grupos que minimize um critério de agrupamento;
- O resultado é uma partição com grupos compactos, ou seja, com variância mínima.

O Algoritmo *k*-Means

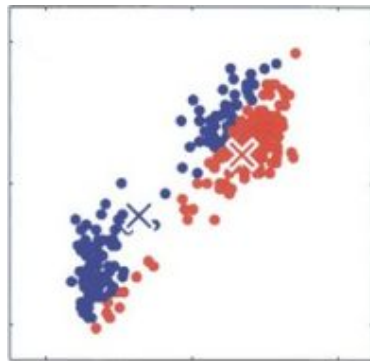
- O critério de agrupamento é dado pelo **erro quadrático**.

$$E = \sum_{j=1}^k \sum_{\mathbf{x}_i \in C^j} d(\mathbf{x}_i, \bar{\mathbf{x}}^j)^2$$

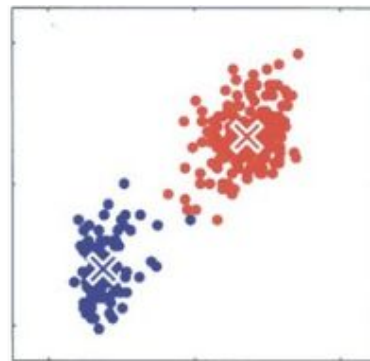
↑
“Erro”

↑

soma das distâncias
de cada objeto até o
respectivo centroide.



valor de E alto



valor de E baixo



*variância mínima
nos grupos*

O Algoritmo *k-Means*

Entrada: Um conjunto de dados $\mathbf{X}_{n \times d}$

Número de *clusters* k

Saída: Uma partição de \mathbf{X} em k *clusters*

Escolher aleatoriamente k valores para centroides dos *clusters*

repita

para cada objeto $\mathbf{x}_i \in \mathbf{X}$ e cluster $\mathbf{C}_j, j = 1, \dots, k$ faça

Calcular a distância entre \mathbf{x}_i e o centroide do cluster $\bar{\mathbf{x}}^{(j)}$: $d(\mathbf{x}_i, \bar{\mathbf{x}}^{(j)})$,
utilizando uma medida de distância

fim

para cada objeto \mathbf{x}_i faça

Associar \mathbf{x}_i ao cluster com centroide mais próximo

fim

para cada cluster $\mathbf{C}_j, j = 1, \dots, k$ faça

Recalcular o centroide

fim

até não haver mais alteração na associação dos objetos aos clusters;

Como recalcular o centroide?

erro quadrático
mínimo

O Algoritmo *k*-Means

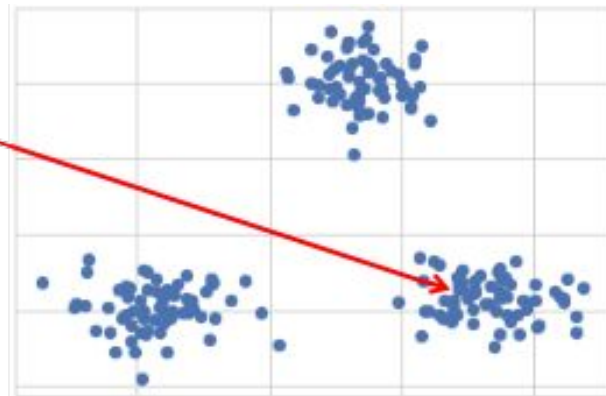
- Como recalcular o centroide?

$$\bar{\mathbf{x}}^k = \frac{1}{n_k} \sum_{\mathbf{x}_i \in C_k} \mathbf{x}_i$$

Somatório dos pontos do *cluster*

Ponto médio (centroide) do agrupamento C^k

centroides



O Algoritmo *k*-Means:

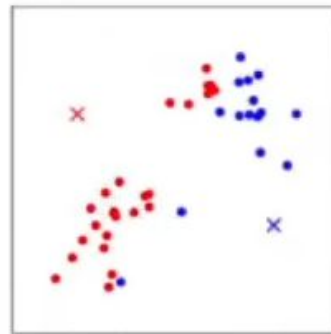
Exemplo de execução



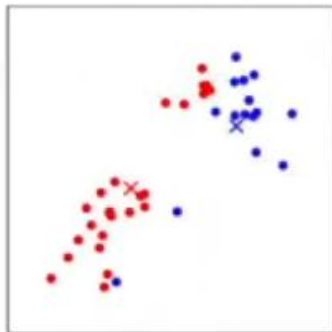
(a)



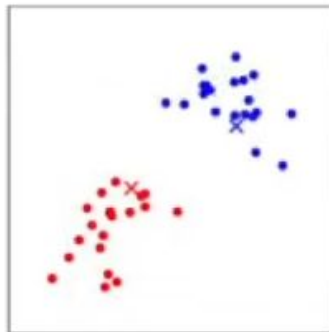
(b)



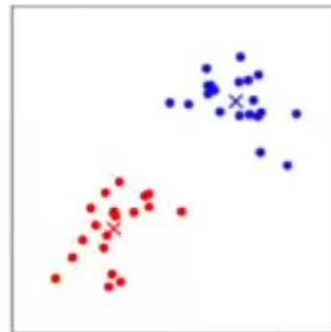
(c)



(d)



(e)



(f)

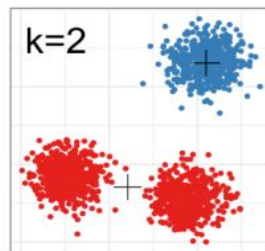
O Algoritmo *k*-Means:

Validação do resultado

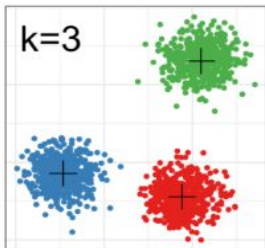
- O que é melhor?

“diferentes agrupamentos são corretos para diferentes propósitos, assim, não podemos dizer que um agrupamento é melhor” (Hartigan, 1985)

menos agrupamentos e *menos* **homogeneidade** interna



k=3



mais agrupamentos e *mais* **homogeneidade** interna

O Algoritmo *k*-Means:

Validação do resultado

- Validação - Como obter o valor ideal de k ?
- Variância *Intracluster* (ou inércia):

$$var(\pi) = \sqrt{\frac{1}{n} \sum_{C_k \in \pi} \sum_{x_i \in C_k} d(\mathbf{x}_i, \bar{\mathbf{x}}^k)} \quad \text{onde:}$$

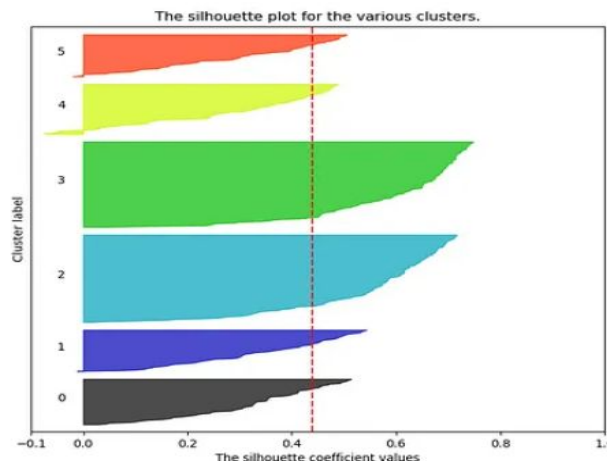
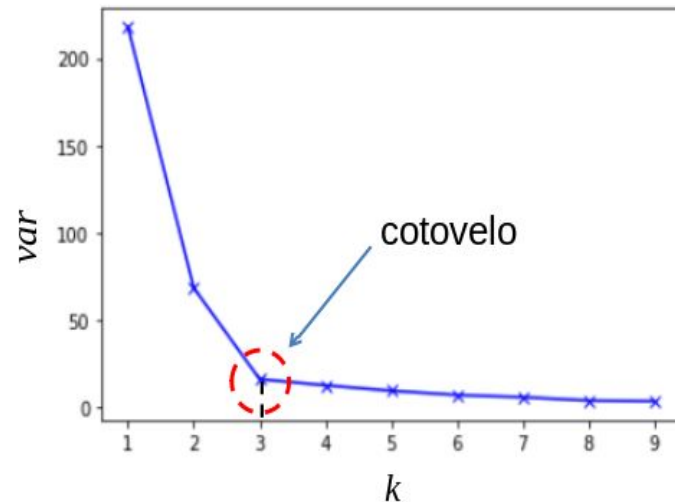
π : partição

- Agrupamentos melhores tem grupos mais compactos;
- Quanto menor a variância, menor a partição.

O Algoritmo *k*-Means:

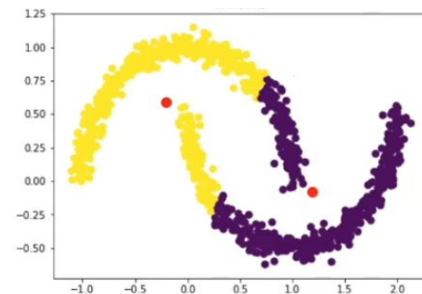
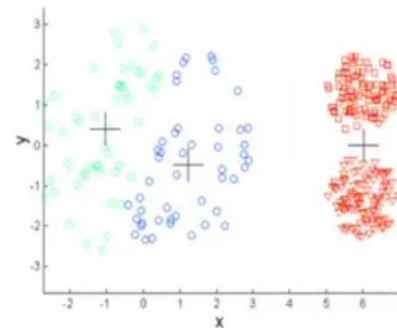
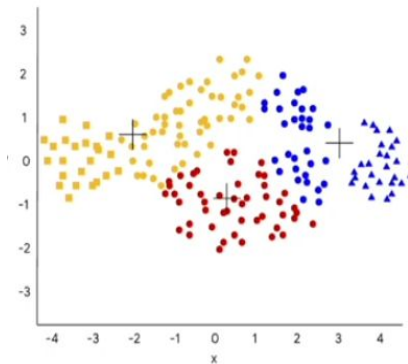
Validação do resultado

- Como obter o valor ideal de *k*?
 - **Método do cotovelo:** Calcula-se a variância *intracluster* (ou inércia) para cada valor de *k*; calcula-se o gráfico entre *k* e o índice. Identifica-se o “**cotovelo**” quando o aumento de *k* não produz efeito significativo no índice; o *k* encontrado tende a ser o melhor;
 - **Coeficiente da silhueta:** mede o quão similar cada ponto em um *cluster* é dos pontos pertencentes aos clusters vizinhos. Varia de -1 (pior) a 1 (melhor).



O Algoritmo *k-Means*: *Pontos Negativos*

- *k-Means* é bastante suscetível a problemas quando os *clusters* são de diferentes tamanhos.
- *k-Means* é também bastante suscetível a problemas quando os *clusters* têm formatos globulares ou diferentes densidades.



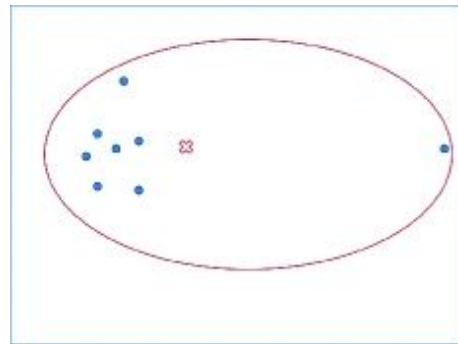
O Algoritmo *k-Means*:

Pontos Positivos

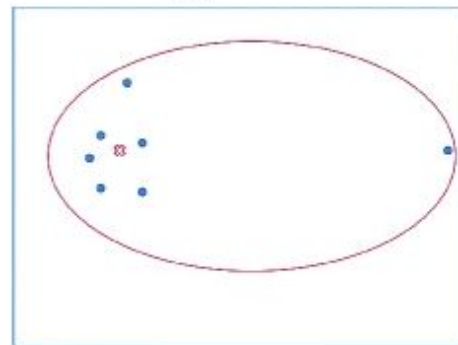
- Relativamente fácil de implementar;
- Possui boa interpretabilidade;
- Pode ser utilizado em grandes bancos de dados;
- Adaptável a novos exemplos;
- A versões aprimoradas que buscam resolver as suas limitações, como número de *clusters*, inicialização, etc). Exemplo: *K-Means++*.

O Algoritmo *k-Medoids*: *Diferenças para o k-Means*

- No K-Means, cada grupo é representado pelo seu centroide, que é a média de todos os pontos de dados nesse grupo;
 - Isso torna o algoritmo suscetível a *outliers*;
- No K-Medoids, cada grupo é representado por um dos pontos de dados reais dentro desse grupo, chamado de medoide.
 - O medoide é o ponto de dados que minimiza a soma das distâncias para todos os outros pontos no mesmo grupo.
 - Isso torna o *k-Medoids* menos suscetível a *outliers* que o *k-Means*.



(a) Mean



(b) Medoid

Clusterização *k*-Medoids:

O algoritmo PAM (Partitioning Around Medoids) - Parte 1

- **Inicialização:**
 - Selecione aleatoriamente, ou por meio de alguma estratégia de inicialização, k medoides iniciais;
- **Atribuição:**
 - Atribua cada ponto de dados ao medoide mais próximo com base em uma métrica de distância (por exemplo, distância euclidiana, distância de Manhattan);

Clusterização *k-Medoids*:

O algoritmo PAM - Parte 2

- **Atualização dos medoides:**
 - Para cada medoide, calcule a dissimilaridade total (frequentemente chamada de custo) somando as distâncias entre o medoide e todos os pontos de dados atribuídos a ele;
 - Para cada ponto de dados que não esteja atualmente servindo como medoide, troque-o por um dos medoides e calcule a dissimilaridade total após a troca;
 - Selecione a troca de medoide que resulta na menor dissimilaridade total para cada cluster.
 - Atualize os medoides com os pontos de dados selecionados.

Clusterização *k-Medoids*:

O algoritmo PAM - Parte 3

- **Convergência:**
 - Repita as etapas de atribuição e atualização até que critérios de convergência sejam atendidos. Critérios comuns de convergência incluem nenhuma ou mínima mudança nos medoides e mínima mudança na dissimilaridade total;
- **Clusterização Final:**
 - A clusterização final é determinada pelos medoides selecionados no final do processo de otimização;
- **Saída:**
 - Retorne os clusters e seus respectivos medoides como resultado da clusterização K-Medoids.

Clusterização *k*-Medoids:

O algoritmo PAM - Exemplo

Passo 1

Selecione dois medoides

- $C1=(3, 4)$
- $C2=(7, 4)$
- *Manhattan Dist* = $|x_1 - x_2| + |y_1 - y_2|$
- $Mdist[(2,6), (3,4)] = |2 - 3| + |6 - 4| = 3$
- $Mdist[(3,4), (3,4)] = |3 - 3| + |4 - 4| = 0$

i	x	y	C1	C2	Cluster
X1	2	6	3	7	C1
X2	3	4	0	4	C1
X3	3	8	4	8	C1
X4	4	7	4	6	C1
X5	6	2	5	3	C2
X6	6	4	3	1	C2
X7	7	3	5	1	C2
X8	7	4	4	0	C2
X9	8	5	6	2	C2
X10	7	6	6	2	C2

Clusterização *k*-Medoids:

O algoritmo PAM - Exemplo

Passo 2

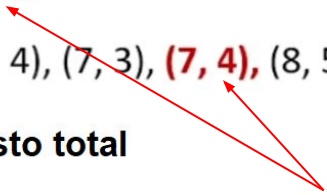
Os clusters são:

- C1: {(2,6), (3,4), (3,8), (4,7)}
- C2: {(6, 2), (6, 4), (7, 3), (7, 4), (8, 5), (7,6)}

i	x	y	C1	C2	Cluster
X1	2	6	3	7	C1
X2	3	4	0	4	C1
X3	3	8	4	8	C1
X4	4	7	4	6	C1
X5	6	2	5	3	C2
X6	6	4	3	1	C2
X7	7	3	5	1	C2
X8	7	4	4	0	C2
X9	8	5	6	2	C2
X10	7	6	6	2	C2

Clusterização *k*-Medoids:

O algoritmo PAM - Exemplo

- C1: {(2,6), **(3,4)**, (3,8), (4,7)}
 - C2: {(6, 2), (6, 4), (7, 3), **(7, 4)**, (8, 5), (7,6)}
 - **Calcule o custo total**
 - $Cost(c, x) = \sum_i |c_i - x_i|$
 - $Custo\ Total = \{Cost((3,4), (2,6)) + Cost((3,4), (3,8)) + Cost((3,4), (4,7)) + Cost((7,4), (6,2)) + Cost((7,4), (6,4)) + Cost((7,4), (7,3)) + Cost((7,4), (8,5)) + Cost((7,4), (7,6))\}$
 - $Custo\ Total = 3 + 4 + 4 + 2 + 3 + 1 + 1 + 2 = 20$
- 
- medoides

Clusterização *k*-Medoids:

O algoritmo PAM - Exemplo

Passo 3

- Escolhe aleatoriamente um ponto não-medoide e recalcula o custo.
- $C1=(3, 4)$ and $C2=(7, 4)$
- $O=(7, 3)$
- Troca $C2$ por O
- **Novos medoides**
- $C1=(3, 4)$ and $O=(7, 3)$

i	x	y	C1	O	Cluster
X1	2	6			
X2	3	4			
X3	3	8			
X4	4	7			
X5	6	2			
X6	6	4			
X7	7	3			
X8	7	4			
X9	8	5			
X10	7	6			

Clusterização *k*-Medoids:

O algoritmo PAM - Exemplo

Passo 3

- **Novos medoides**
- $C1=(3, 4)$ and $O=(7, 3)$
- *Manhattan Dist* = $|x_1 - x_2| + |y_1 - y_2|$
- $Mdist[(2, 6), (7, 3)] = |2 - 7| + |6 - 3| = 8$

i	x	y	C1	O	Cluster
X1	2	6	3	8	C1
X2	3	4	0	5	C1
X3	3	8	4	9	C1
X4	4	7	4	7	C1
X5	6	2	5	2	O
X6	6	4	3	2	O
X7	7	3	5	0	O
X8	7	4	4	1	O
X9	8	5	6	3	O
X10	7	6	6	3	O

Clusterização *k*-Medoids:

O algoritmo PAM - Exemplo

- C1: {(2,6), (3,4), (3,8), (4,7)}
- O: {(6, 2), (6, 4), (7, 3), (7, 4), (8, 5), (7,6)}
- **Calcula o custo total**
- $Cost(c, x) = \sum_i |c_i - x_i|$
- $Custo\ Total\ Atual = \{Cost((3,4), (2,6)) + Cost((3,4), (3,8)) + Cost((3,4), (4,7)) + Cost((7,3), (6,2)) + Cost((7,3), (6,4)) + Cost((7,3), (7,4)) + Cost((7,3), (8,5)) + Cost((7,3), (7,6))\}$
- **Custo Total Atual = 3 + 4 + 4 + 2 + 2 + 1 + 3 + 3 = 22**

Clusterização *k*-Medoids:

O algoritmo PAM - Exemplo

Passo 4

- Custo de trocar o medoide C2 por O
- $S = \text{Total do Custo Atual} - \text{Total do Custo Anterior}$
- $S = 22 - 20 = 2 > 0$
- Portanto, os resultados pioraram após trocar C2 por O
- Medoides finais são **C1=(3, 4) and C2=(7, 4)**
- Os clusters são
- **C1: {(2,6), (3,4), (3,8), (4,7)}**
- **C2: {(6, 2), (6, 4), (7, 3), (7, 4), (8, 5), (7,6)}**

Trabalho 3 - Entrega dia 23/10

Utilize o algoritmo *k-Means* para agrupar os atacadistas em grupos de venda. Utilize o gráfico do cotovelo e da silhueta para definir o “melhor” *k*. Por fim, utilize um gráfico de colunas empilhadas para exibir o quanto cada agrupamento gastou em cada categoria de produto.



DATA SCIENCE

AULA 5 - Aprendizagem Não-Supervisionada

Dúvidas e/ou perguntas?



gabrielmachado@unifeso.edu.com



<https://www.linkedin.com/in/machadogabriel>



<https://github.com/UNIFESO-Gabriel/data-science>