LEONARDO STALL

CLUSTERING

K-MEANS

OUTUBRO 2024

LEONARDO STALL

Introdução

01

O que é K-means?

É um algoritmo de aprendizado não supervisionado utilizado para agrupar dados em diferentes clusters com base em similaridade

02

Importância

É amplamente utilizado em aplicações como segmentação de mercado, pois ajuda a identificar padrões ocultos em grandes conjuntos de dados

03

Como o funciona?

Atribui pontos de dados a clusters e ajustando as posições desses clusters até que a diferença entre os grupos seja minimizada

04

Quando usar?

K-means é eficaz quando temos grandes quantidades de dados não rotulados e queremos encontrar grupos naturais nos dados

Formula matemática

K-means é baseado no conceito de minimizar a soma das distâncias quadradas entre os pontos de dados e o centroide (o centro do cluster).

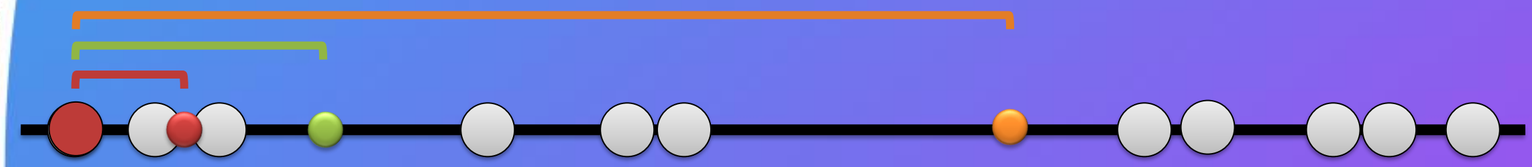
Fórmula da Função de Custo (Inércia)

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2$$

J: A função de custo que queremos minimizar.
k: O número de clusters.
Ci: O i-ésimo cluster.
x: Um ponto de dado pertencente ao cluster Ci.
μi: O centroide do cluster Ci.
$||x - \mu_i||^2$:A distância euclidiana quadrada entre o ponto x e o centroide μi .

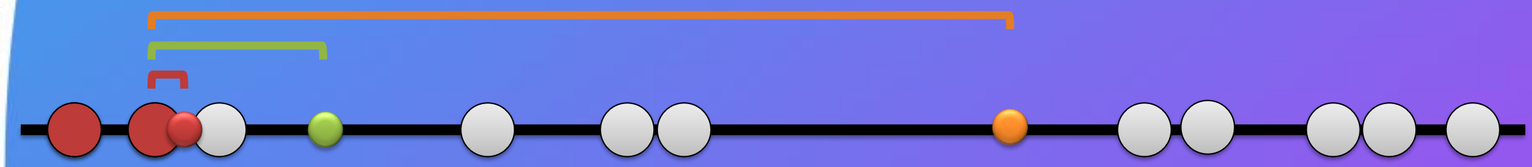
Funcionamento

O algoritmo K-means começa escolhendo k centroides aleatórios. Cada ponto de dado é atribuído ao centroide mais próximo. Em seguida, os centroides são recalculados como a média dos pontos de cada cluster. Os pontos são reatribuídos aos novos centroides e o processo se repete até que os centroides se estabilizem ou o número máximo de iterações seja atingido. O resultado final são k clusters com pontos de dados similares.



Funcionamento

O algoritmo K-means começa escolhendo k centroides aleatórios. Cada ponto de dado é atribuído ao centroide mais próximo. Em seguida, os centroides são recalculados como a média dos pontos de cada cluster. Os pontos são reatribuídos aos novos centroides e o processo se repete até que os centroides se estabilizem ou o número máximo de iterações seja atingido. O resultado final são k clusters com pontos de dados similares.

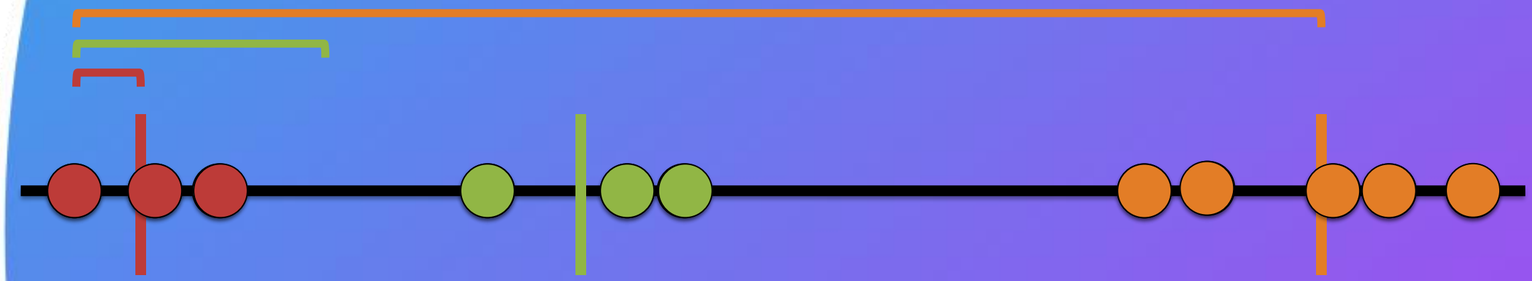


Funcionamento

O algoritmo K-means começa escolhendo k centroides aleatórios. Cada ponto de dado é atribuído ao centroide mais próximo. Em seguida, os centroides são recalculados como a média dos pontos de cada cluster. Os pontos são reatribuídos aos novos centroides e o processo se repete até que os centroides se estabilizem ou o número máximo de iterações seja atingido. O resultado final são k clusters com pontos de dados similares.

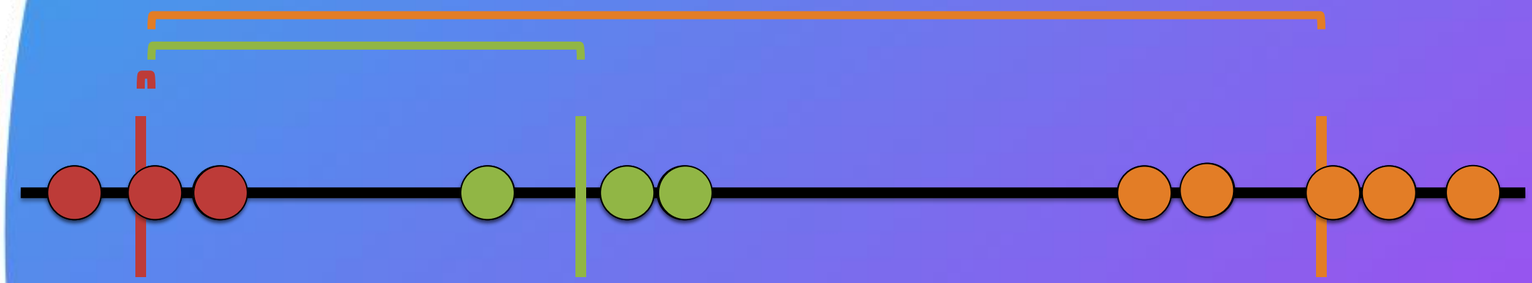
Funcionamento

O algoritmo K-means começa escolhendo k centroides aleatórios. Cada ponto de dado é atribuído ao centroide mais próximo. Em seguida, os centroides são recalculados como a média dos pontos de cada cluster. Os pontos são reatribuídos aos novos centroides e o processo se repete até que os centroides se estabilizem ou o número máximo de iterações seja atingido. O resultado final são k clusters com pontos de dados similares.



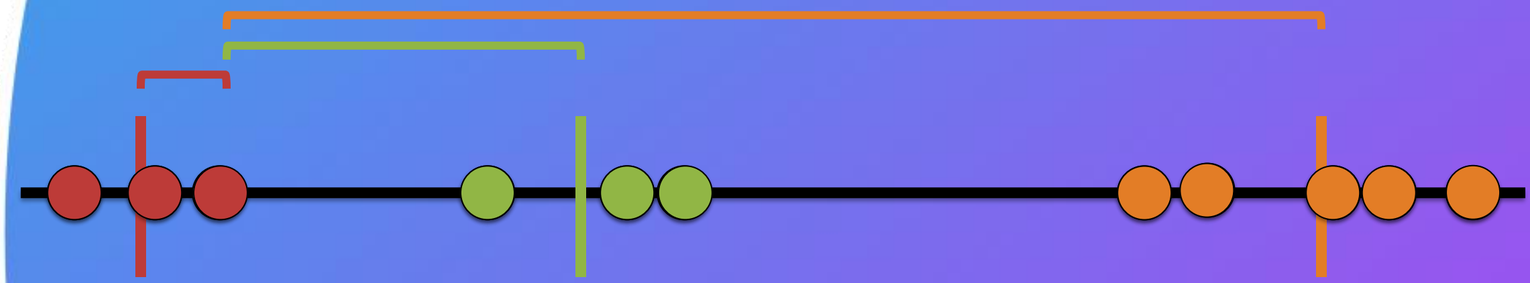
Funcionamento

O algoritmo K-means começa escolhendo k centroides aleatórios. Cada ponto de dado é atribuído ao centroide mais próximo. Em seguida, os centroides são recalculados como a média dos pontos de cada cluster. Os pontos são reatribuídos aos novos centroides e o processo se repete até que os centroides se estabilizem ou o número máximo de iterações seja atingido. O resultado final são k clusters com pontos de dados similares.



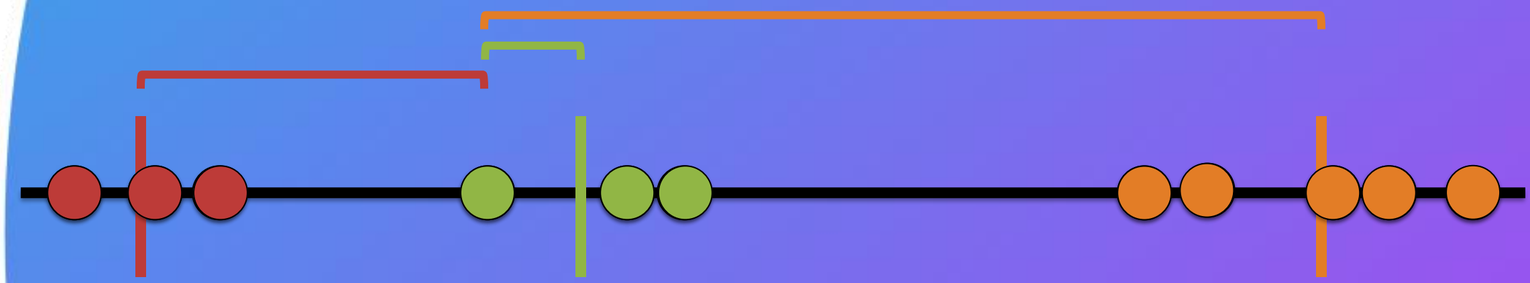
Funcionamento

O algoritmo K-means começa escolhendo k centroides aleatórios. Cada ponto de dado é atribuído ao centroide mais próximo. Em seguida, os centroides são recalculados como a média dos pontos de cada cluster. Os pontos são reatribuídos aos novos centroides e o processo se repete até que os centroides se estabilizem ou o número máximo de iterações seja atingido. O resultado final são k clusters com pontos de dados similares.



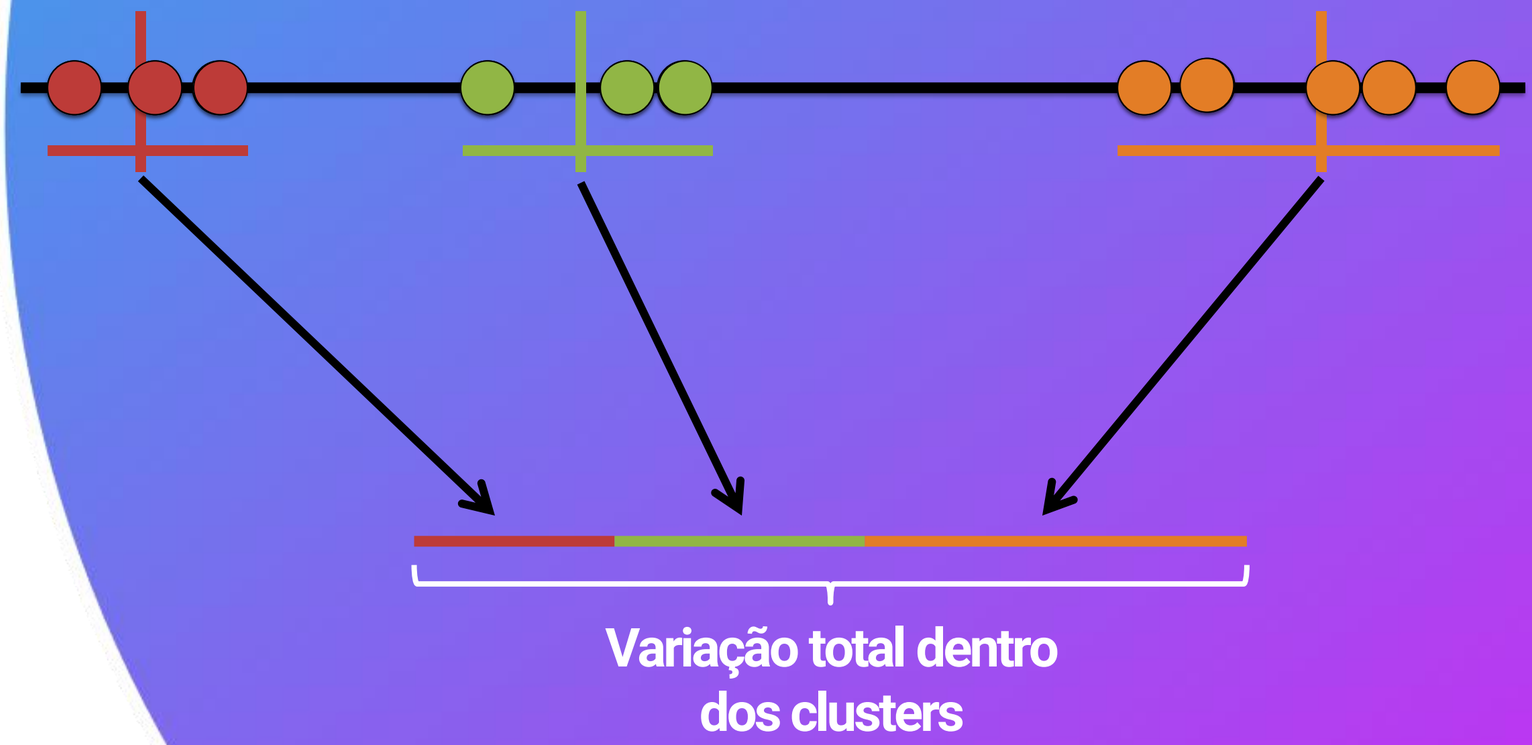
Funcionamento

O algoritmo K-means começa escolhendo k centroides aleatórios. Cada ponto de dado é atribuído ao centroide mais próximo. Em seguida, os centroides são recalculados como a média dos pontos de cada cluster. Os pontos são reatribuídos aos novos centroides e o processo se repete até que os centroides se estabilizem ou o número máximo de iterações seja atingido. O resultado final são k clusters com pontos de dados similares.



Funcionamento

O algoritmo K-means começa escolhendo k centroides aleatórios. Cada ponto de dado é atribuído ao centroide mais próximo. Em seguida, os centroides são recalculados como a média dos pontos de cada cluster. Os pontos são reatribuídos aos novos centroides e o processo se repete até que os centroides se estabilizem ou o número máximo de iterações seja atingido. O resultado final são k clusters com pontos de dados similares.



Funcionamento

O algoritmo K-means começa escolhendo k centroides aleatórios. Cada ponto de dado é atribuído ao centroide mais próximo. Em seguida, os centroides são recalculados como a média dos pontos de cada cluster. Os pontos são reatribuídos aos novos centroides e o processo se repete até que os centroides se estabilizem ou o número máximo de iterações seja atingido. O resultado final são k clusters com pontos de dados similares.

Funcionamento

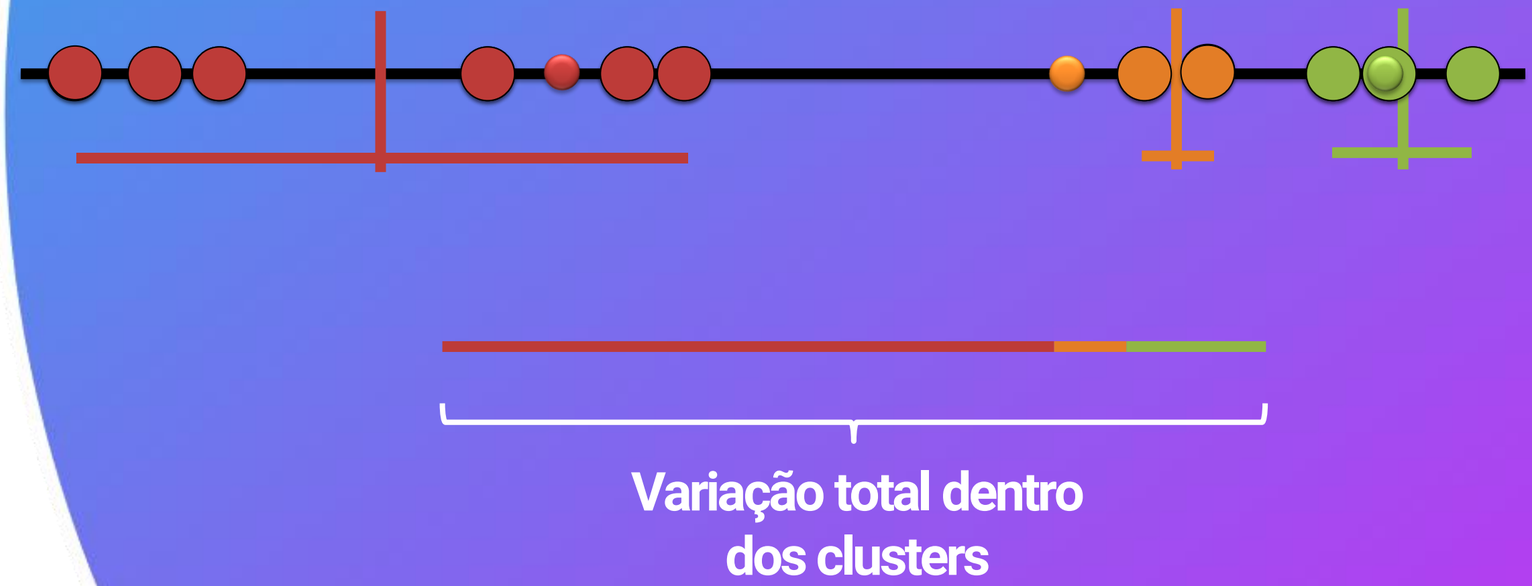
O algoritmo K-means começa escolhendo k centroides aleatórios. Cada ponto de dado é atribuído ao centroide mais próximo. Em seguida, os centroides são recalculados como a média dos pontos de cada cluster. Os pontos são reatribuídos aos novos centroides e o processo se repete até que os centroides se estabilizem ou o número máximo de iterações seja atingido. O resultado final são k clusters com pontos de dados similares.

Funcionamento

O algoritmo K-means começa escolhendo k centroides aleatórios. Cada ponto de dado é atribuído ao centroide mais próximo. Em seguida, os centroides são recalculados como a média dos pontos de cada cluster. Os pontos são reatribuídos aos novos centroides e o processo se repete até que os centroides se estabilizem ou o número máximo de iterações seja atingido. O resultado final são k clusters com pontos de dados similares.

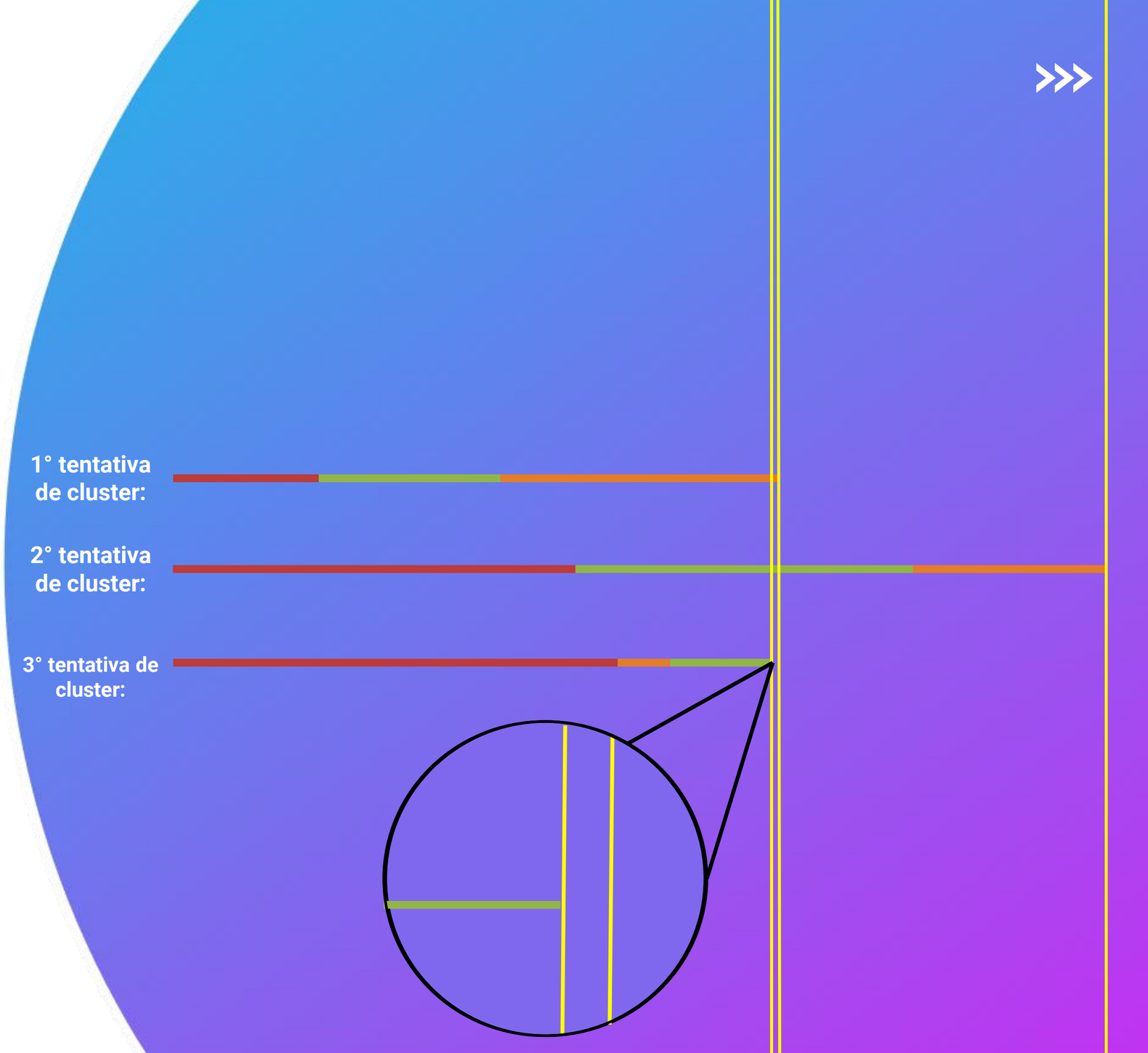
Funcionamento

O algoritmo K-means começa escolhendo k centroides aleatórios. Cada ponto de dado é atribuído ao centroide mais próximo. Em seguida, os centroides são recalculados como a média dos pontos de cada cluster. Os pontos são reatribuídos aos novos centroides e o processo se repete até que os centroides se estabilizem ou o número máximo de iterações seja atingido. O resultado final são k clusters com pontos de dados similares.



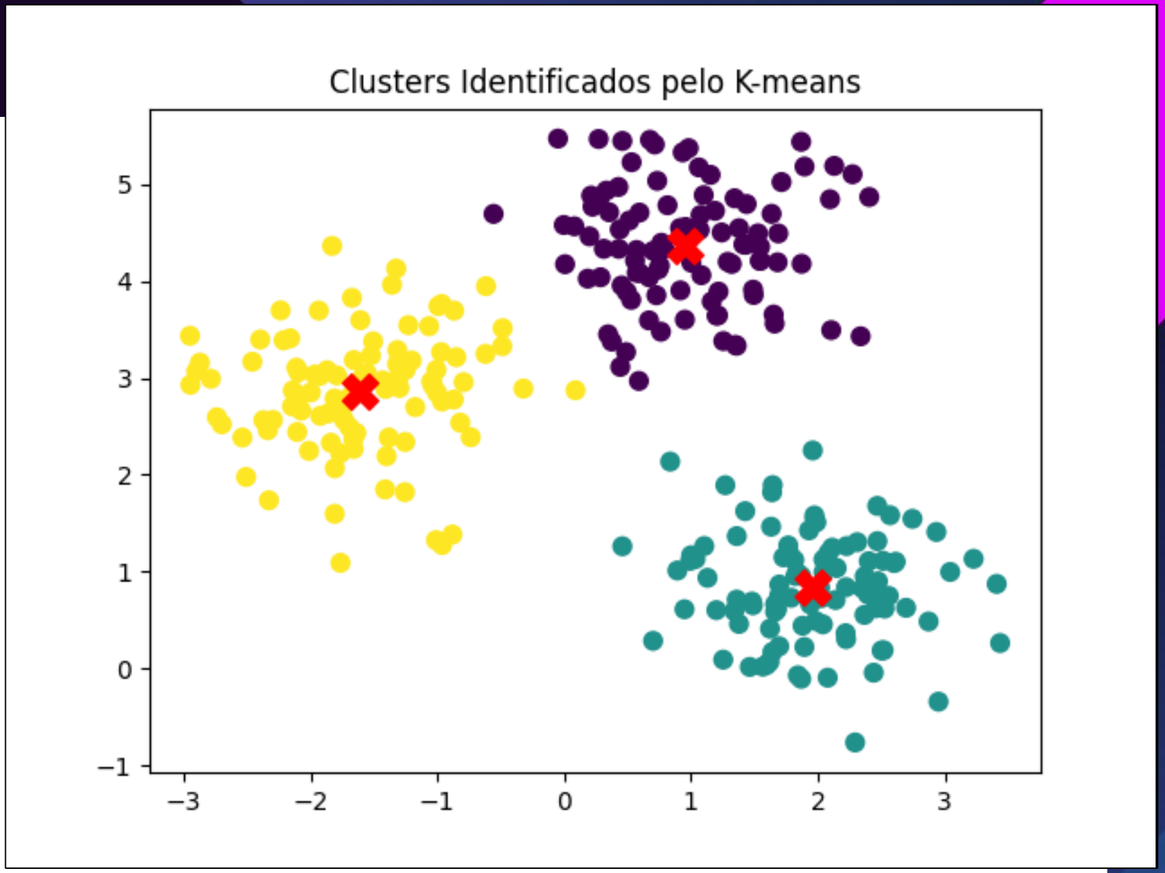
Funcionamento

O algoritmo K-means começa escolhendo k centroides aleatórios. Cada ponto de dado é atribuído ao centroide mais próximo. Em seguida, os centroides são recalculados como a média dos pontos de cada cluster. Os pontos são reatribuídos aos novos centroides e o processo se repete até que os centroides se estabilizem ou o número máximo de iterações seja atingido. O resultado final são k clusters com pontos de dados similares.

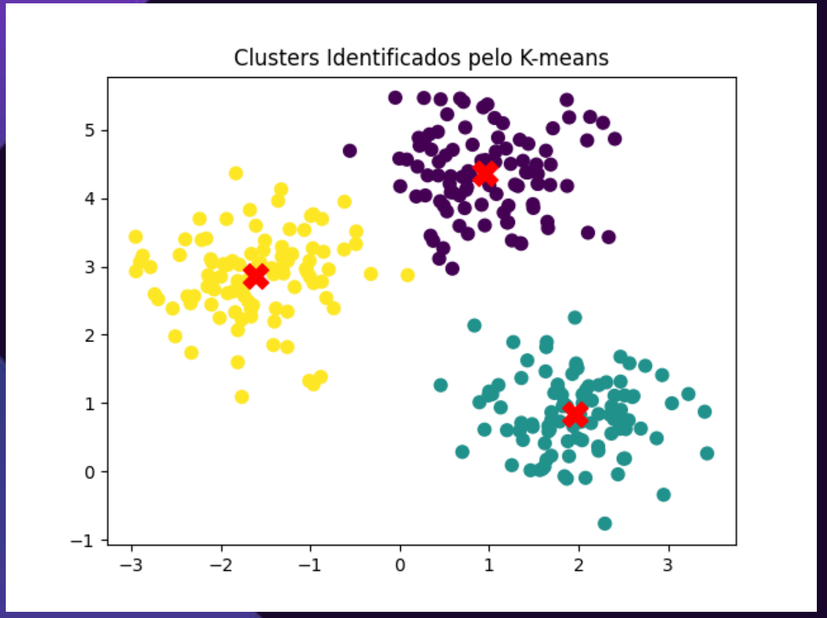


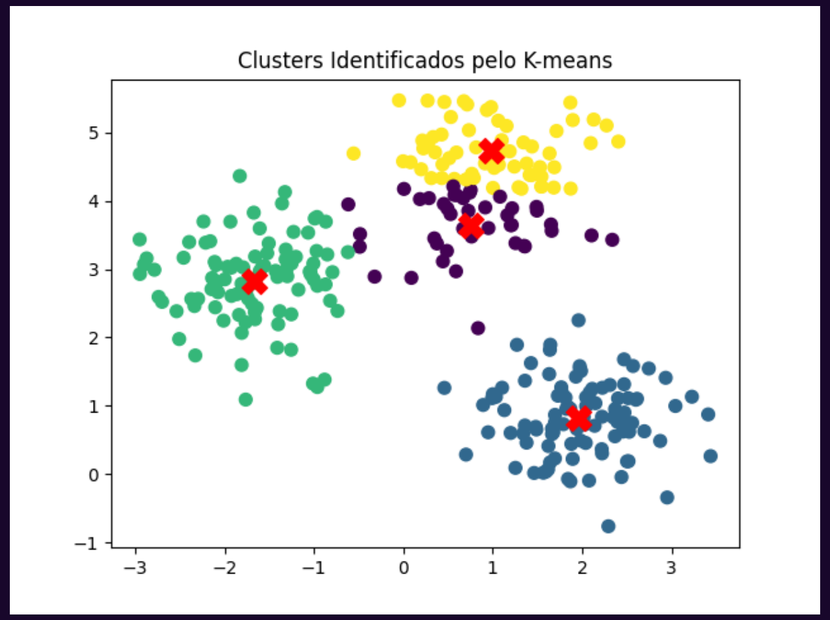
2° Período > PMC105A - Inteligência Artificial e Aprendizagem de Máquina > apresentação.py > ...

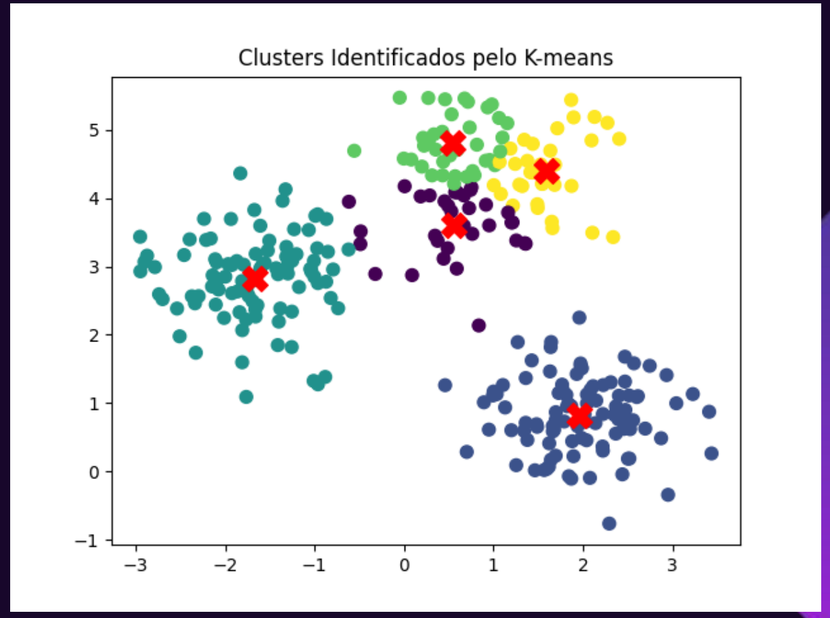
```python
# Importando bibliotecas necessárias
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs

# Gerando dados fictícios
X, y = make_blobs(n_samples=300, centers=3, cluster_std=0.60, random_state=0)

# Visualizando os dados gerados
plt.scatter(X[:, 0], X[:, 1], s=50)
plt.title("Dados Gerados")
plt.show()

# Aplicando K-means com k = 3 (número de clusters)
kmeans = KMeans(n_clusters=3)
kmeans.fit(X)

# Obtendo os centroides e os rótulos (clusters) para cada ponto de dado
centroides = kmeans.cluster_centers_
rótulos = kmeans.labels_

# Visualizando os clusters com seus centroides
plt.scatter(X[:, 0], X[:, 1], c=rótulos, s=50, cmap='viridis')

# Plotando os centroides
plt.scatter(centroides[:, 0], centroides[:, 1], s=200, c='red', marker='X')
plt.title("Clusters Identificados pelo K-means")
plt.show()
```



Como saber o melhor valor de k?





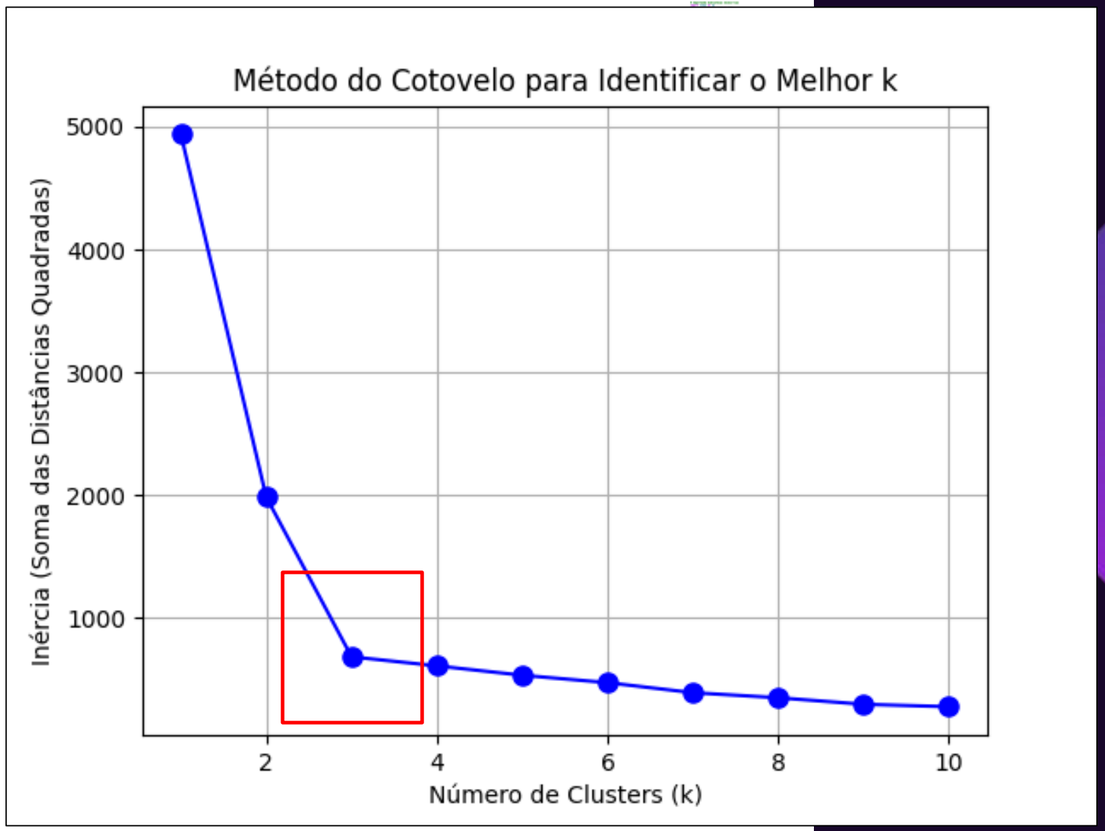


Como saber o melhor valor de k?

Método do Cotovelo

2° Período > PMC105A - Inteligência Artificial e Aprendizagem de Máquina > apresentação.py > ...

```python
# Importando bibliotecas necessárias
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs

# Gerando dados fictícios (mesmo procedimento)
X, y = make_blobs(n_samples=1000, centers=3, cluster_std=0.60, random_state=0)

# Lista para armazenar os valores de inércia para diferentes k's
inercias = []

# Testando diferentes valores de k (de 1 a 10)
k_values = range(1, 11)
for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=0)
    kmeans.fit(X)
    # Guardando o valor da inércia (soma das distâncias quadradas)
    inercias.append(kmeans.inertia_)

# Plotando o gráfico do Método do Cotovelo
plt.plot(k_values, inercias, 'bo-', markersize=8)
plt.xlabel('Número de Clusters (k)')
plt.ylabel('Inércia (Soma das Distâncias Quadradas)')
plt.title('Método do Cotovelo para Identificar o Melhor k')
plt.grid(True)
plt.show()
```



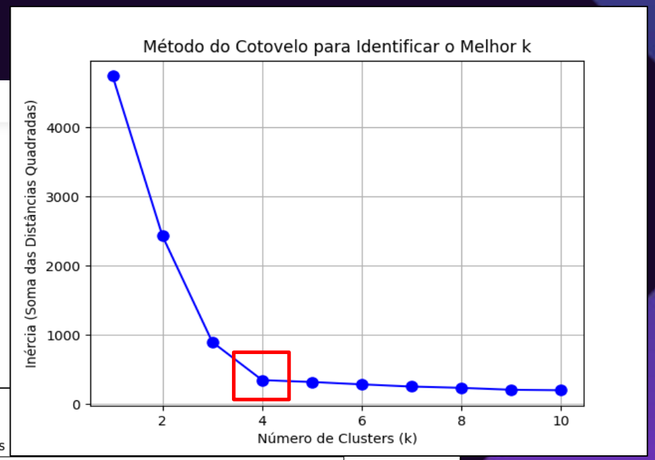
LEONARDO STALL

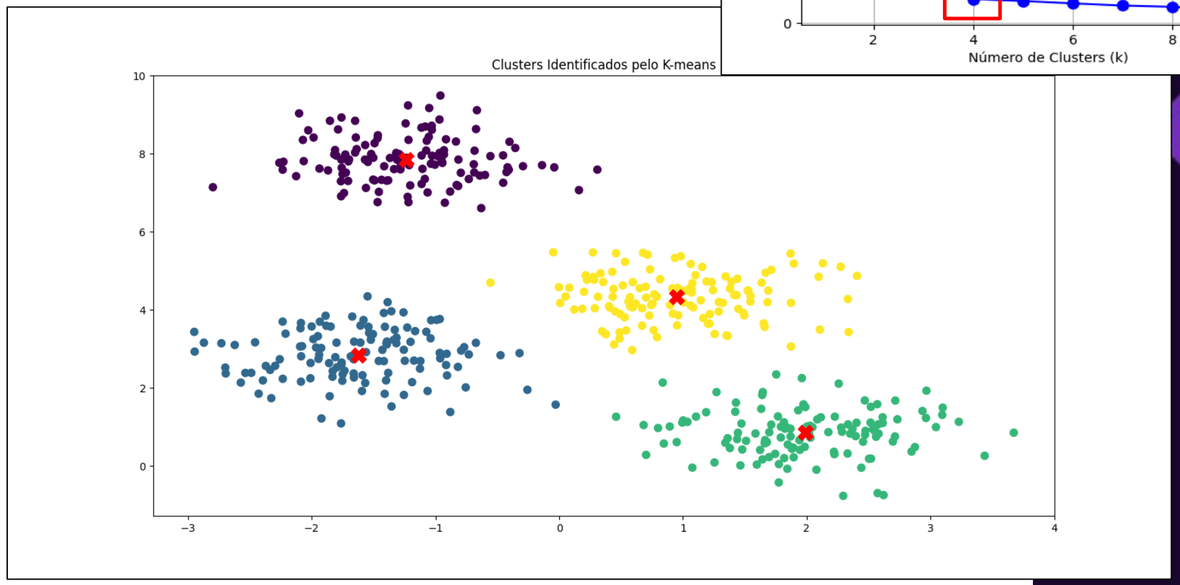
Como saber o melhor valor de k?

Método do Cotovelo

2° Período > PMC105A - Inteligência Artificial e Aprendizagem de Máquina > k-means cotovelo.py > ...

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs

X, y = make_blobs(n_samples=500, centers=4, cluster_std=0.60, random_state=0)

inercias = []

# Testando diferentes valores de k (de 1 a 10)
k_values = range(1, 11)
for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=0)
    kmeans.fit(X)
    # Guardando o valor da inércia (soma das distâncias quadradas)
    inercias.append(kmeans.inertia_)

plt.plot(k_values, inercias, 'bo-', markersize=8)
plt.xlabel('Número de Clusters (k)')
plt.ylabel('Inércia (Soma das Distâncias Quadradas)')
plt.title('Método do Cotovelo para Identificar o Melhor k')
plt.grid(True)
plt.show()
```





stall.leonardo@pucpr.edu.br

OBRIGADO