UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

GUSTAVO COUSSEAU

COMPARAÇÃO DE TEMPO E RESULTADO DE ALGORITMOS DE AGRUPAMENTO UTILIZANDO DIFERENTES DISTÂNCIAS E BASES DE DADOS

PATO BRANCO

2023

GUSTAVO COUSSEAU

COMPARAÇÃO DE TEMPO E RESULTADO DE ALGORITMOS DE AGRUPAMENTO UTILIZANDO DIFERENTES DISTÂNCIAS E BASES DE DADOS

Time and result comparison of clustering algorithms using several distances and databases

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia de Computação do Curso de Bacharelado em Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Érick Oliveira Rodrigues

PATO BRANCO

2023

GUSTAVO COUSSEAU

COMPARAÇÃO DE TEMPO E RESULTADO DE ALGORITMOS DE AGRUPAMENTO UTILIZANDO DIFERENTES DISTÂNCIAS E BASES DE DADOS

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia de Computação do Curso de Bacharelado em Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 20/06/2022

Prof. Me. Vinicius Pegorini
Universidade Tecnológica Federal do Paraná

Profª. Drª. Luciene de Oliveira Marin
Universidade Tecnológica Federal do Paraná

Prof. Dr. Luiz Fernando Puttow Southier
Universidade Tecnológica Federal do Paraná

PATO BRANCO

2023

RESUMO

Esse trabalho é uma pesquisa de análise dos algoritmos de aprendizado de máquina não supervisionados DBSCAN, *K-Means* e *Mean-Shift* utilizando as distâncias *Canberra*, *Chebyshev*, Euclidiana, *Minkowski* e Rodrigues com algumas bases de dados que estão contidas no repositório da UCI. Tem como objetivo, comparar a média dos tempos que cada algoritmo demora para processar onze bases de dados com dez distâncias diferentes. Ainda, comparar os agrupamentos obtidos pelo DBSCAN e *Mean-Shift* em relação ao *K-means*. Os resultados mostram que ao utilizar a quantidade e as posições dos agrupamentos obtidos no DBSCAN como entradas para o *K-Means* e os obtidos no *Mean-Shift* como entradas para o *K-Means*, derivam de agrupamentos diferentes mas com alguma igualdade entre os agrupamentos. No entanto, a igualdade é maior com a combinação dos resultados do *Mean-Shift* com o *K-Means*. As menores médias de tempo foram obtidas pelo algoritmo *K-Means* e as maiores pelo algoritmo *Mean-Shift*. E no geral, a distância *Chebyshev* foi responsável pelas menores médias de tempo em 3 dos 4 métodos. Para esses resultados, é necessário a escolha dos parâmetros de entrada adequados para gerar um número considerável de agrupamentos nos algoritmos DBSCAN e *Mean-Shift*.

Palavras-chave: algoritmos de agrupamento; distâncias; aprendizado de mquina.

ABSTRACT

This work is a research analysis of unsupervised machine learning algorithms DBSCAN, K-Means and Mean-Shift using Canberra, Chebyshev, Euclidean, Minkowski and Rodrigues distances with some databases from the UCI repository. It aims to compare the average time that each algorithm takes to process eleven databases with ten different distances. In addition, compare the clusters obtained by DBSCAN and Mean-Shift in relation to K-Means. The results show that when using the number and positions of the clusters obtained in DBSCAN as inputs for K-Means and those obtained in Mean-Shift as inputs for K-means, derive from different clusters but with some equality. However, the equality is higher with the combination of Mean-Shift and K-means results. The lowest time averages were obtained by the K-Means algorithm and the highest by the Mean-Shift algorithm. And overall, the Chebyshev distance was responsible for the lowest time averages in 3 of the 4 methods. For these results, it is necessary to choose the appropriate input parameters to generate a considerable number of clusters in the DBSCAN and Mean-Shift algorithms.

Keywords: clustering algorithm; distances; machine learning.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A construção de computadores digitais possibilitaram a automatização de técnicas de análise de dados. Devido ao aumento do poder computacional e o avanço de algoritmos, ferramentas de *Machine Learning* tornaram-se poderosas na busca de padrões em dados (BIAMONTE *et al.*, 2017).

1.1 Considerações iniciais

Machine Learning é um ramo da computação, em que os algoritmos são projetados para emular a inteligência humana aprendendo o ambiente ao seu redor. Técnicas baseadas neste ramo têm sido aplicada em diversas áreas, desde reconhecimento de padrões e visão computacional até aplicações médicas (NAQA; MURPHY, 2015).

Este ramo pode ser dividido em aprendizado supervisionado, aprendizado não supervisionado, redes neurais, entre outros, sendo que os mais utilizados são aprendizado supervisionado e não supervisionado. No aprendizado supervisionado, os dados de treino são divididos em classes e atributos, e o trabalho da máquina é associar as classes aos dados que estão fora do treinamento. Todos os algoritmos aprendem algum tipo de padrão nos dados de treinamento e aplicam nos dados de teste para predizer ou classificar. Já no aprendizado não supervisionado, os dados não possuem categorias, e o trabalho da máquina é achar categorias naturais em que cada dado de treinamento se enquadra, e depois, categorizar os dados que estão fora do conjunto de treinamento. Os algoritmos de aprendizado não supervisionado são principalmente utilizados para agrupamento e redução de características, ou seja, remove dados redundantes e/ou combina alguns dados se aplicável (MAHESH, 2020).

O objetivo dos algoritmos de agrupamento, ou *clustering*, é descritivo, de maneira que instâncias de dados semelhantes sejam agrupadas e instâncias diferentes pertençam a agrupamentos diferentes. Entretanto, nos algoritmos de classificação, a avaliação é extrínseca, pois os agrupamentos devem refletir alguma classe de referência (ROKACH; MAIMON, 2005).

Já que o agrupamento agrupa instâncias/objetos semelhantes, é necessário uma medida que possa determinar se dois objetos são semelhantes ou diferentes. E para estimar essa relação, pode ser usada uma medida de distância. Muitos métodos de agrupamento usam medidas de distâncias para determinar a similaridade ou dissimilaridade entre dois objetos. Uma medida de distância válida deve ser simétrica e obter um valor mínimo caso os dois objetos sejam similares (ROKACH; MAIMON, 2005).

Aprender uma boa distância no espaço de dados é crucial na aplicação no mundo real. Boas distâncias são muito importantes em visão computacional para o agrupamento de imagens. Uma aplicação importante é no Sistema de Recuperação de Imagem Baseada em Conteúdo (CBIR), que é altamente dependente deste critério para definir a similaridade das imagens (YANG; JIN, 2006).

Com isso, nesse trabalho é feito um estudo das distâncias de *Canberra*, *Chebyshev*, Euclidiana, *Minkowski* e Rodrigues e suas variações de parâmetros utilizando 11 bases de dados do repositório da UCI nos algoritmos de aprendizado não supervisionado DBSCAN, *K-Means* e *Mean-Shift*. Ainda, é feito uma análise desses algoritmos em comparação com cada distância e mostrada a acurácia do DBSCAN e *Mean-Shift* em relação ao *K-means*. Essa comparação nunca foi realizada antes e apenas foi utilizada as distâncias mencionadas acima no algoritmo *KNN* conforme o trabalho realizado por Rodrigues (2018).

1.2 Problemática

Distâncias são de fundamental importância no aprendizado de máquina, pois sua aplicação afeta significantemente a performance de muitos métodos de aprendizado (SHEN *et al.*, 2013). Muitos pesquisadores se esforçaram em achar uma distância para encontrar a similaridade de objetos. As distâncias desempenham um papel crítico em problemas de análise de padrões, como a classificação e o agrupamento. (CHOI; CHA; TAPPERT, 2010).

Um dos problemas que afetam agrupadores e classificadores é a otimização de parâmetros. A combinação de distâncias e outros parâmetros, como a escolha do valor para o parâmetro k dos algoritmos *K-Means* e KNN, funcionam melhor do que outras. A melhor combinação varia de acordo com o problema de agrupamento ou classificação (RODRIGUES, 2018).

Muitos estudos têm demonstrado que uma distância de aprendizado pode aprimorar significativamente o desempenho em tarefas de classificação, agrupamento e recuperação (YANG; JIN, 2006). Rodrigues (2018) realizou um estudo recente, o qual desenvolveu uma distância intermediária entre as distâncias *Chebyshev* e *Minkowsky*. Essa distância intermediária comparada com as distâncias *Canberra*, *Chebyshev*, Euclidiana, *Manhattan*, *Minkowski* e Soma da Diferença dos Quadrados, obteve boas acurácias quando combinado com o classificador KNN.

1.3 Objetivos

A seguir é apresentado o objetivo geral o qual explica de forma sucinta o objetivo do trabalho e na próxima seção as etapas para se alcançar o objetivo geral.

1.3.1 Objetivo geral

Testar a nova distância de Rodrigues nos algoritmos de agrupamento DBSCAN, *K-Means* e *Mean-Shift*, os quais utilizam distâncias como um dos parâmetros de implementação. Comparar com as distâncias *Canberra*, *Chebyshev*, Euclidiana e *Minkowski*. Avaliar as diferenças das distâncias e comportamentos em relação ao tempo e igualdade de agrupamentos.

1.3.2 Objetivos específicos

- Implementar as distâncias no *Python*;
- Implementar os algoritmos do método de como será comparado os algoritmos e a comparação dos agrupamentos;
- Executar testes de tempo e igualdade entre os agrupamentos;

1.4 Justificativa

A distância implementada proposta por Rodrigues (2018) foi testada em um algoritmo de classificação chamado KNN. Os testes feitos mostraram que, em comparação com outras distâncias, como *Chebyshev*, *Manhattan*, *Canberra*, entre outras, a distância de Rodrigues teve resultados significativamente melhores que a distância Euclidiana, e um pouco mais rápida que a distância de *Manhattan*.

Um outro estudo, feito por Perlibakas (2004), mostra que no reconhecimento facial baseado em análise de componentes principais, quatro de catorze distâncias foram as melhores relatadas. Ainda, Lu *et al.* (2016) relatou que a variação do parâmetro *p* na distância de *Minkowski* leva a melhores acurácias em regressões geograficamente ponderadas.

Os estudos ainda são poucos em relação a avaliação de diversas distâncias com um número substancial de conjuntos de dados em tarefas de classificação e agrupamento. Com base nisso, serão feitos testes com diferentes distâncias utilizando três algoritmos de agrupamento, DBSCAN, *K-Means* e *Mean-Shift*. Além disso, os testes serão feitos com diferentes valores dos parâmetros utilizados em cada algoritmo. Para o DBSCAN, a distância máxima entre duas amostras e o número de amostras dentro dessa distância. No caso do *K-Means*, o número de centroides e suas localizações iniciais. E por fim, para o *Mean-Shift*, o tamanho da região de interesse.

As análises serão feitas utilizando 11 *datasets* com diferentes tipos de atributos.

1.5 Estrutura do trabalho

No próximo capítulo são apresentados alguns conceitos técnicos sobre o Aprendizado de Máquina e seus ramos assim como os algoritmos que serão utilizados neste trabalho. Ademais, é feita uma breve comparação entre os algoritmos e descrições das distâncias que serão abordadas. Ainda, é feita uma descrição de como serão utilizadas as bases de dados e alguns dos trabalhos relacionados a esta pesquisa. Para o próximo capítulo é apresentado os materiais e os métodos que serão utilizados para o desenvolvimento da pesquisa. Em seguida, os resultados obtidos e como foram obtidos em relação ao capítulo anterior. Por último, uma conclusão sobre os resultados.

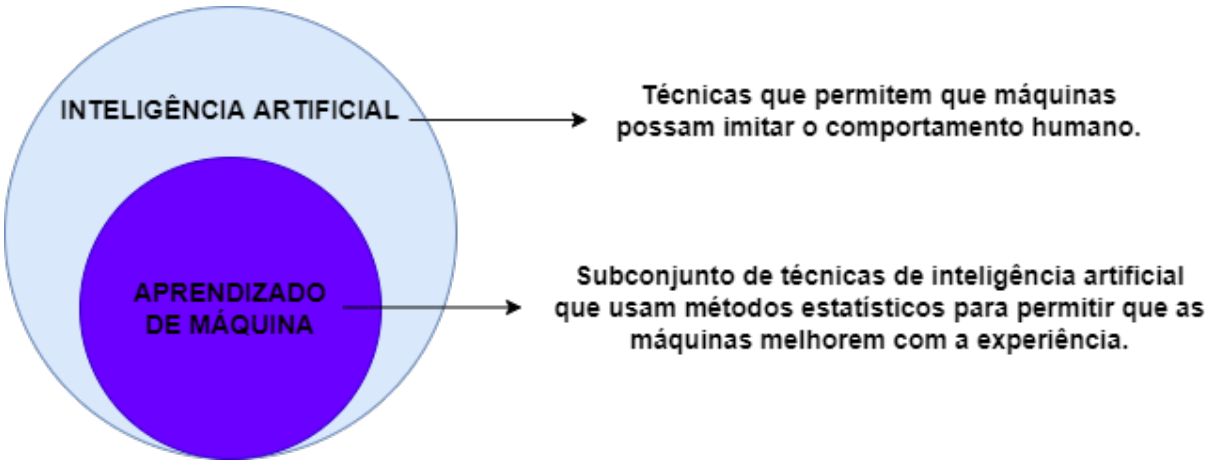
2 REFERENCIAL TEÓRICO

Neste capítulo são apresentados alguns conceitos e fundamentos teóricos. Primeiro, uma breve introdução sobre aprendizado de máquina. Ainda, são apresentados os conceitos e características do aprendizado supervisionado e não supervisionado e quais são suas aplicações. Conceitua-se, também, a teoria dos algoritmos DBSCAN, *K-Means* e *Mean-Shift* e seus diferentes atributos. E, por fim, as distâncias e suas características utilizadas como parâmetros nos algoritmos anteriormente citados.

2.1 Aprendizado de Máquina

Como descrito na Figura 1 o Aprendizado de Máquina é um campo de Inteligência Artificial com o objetivo de desenvolver algoritmos capazes de aprender, automaticamente, a partir de dados. Um agente artificialmente inteligente precisa ser capaz de reconhecer objetos ao seu redor e prever o comportamento do seu ambiente para fazer escolhas informadas (MEHTA *et al.*, 2019).

Figura 1 – Diferença Entre As Duas Áreas



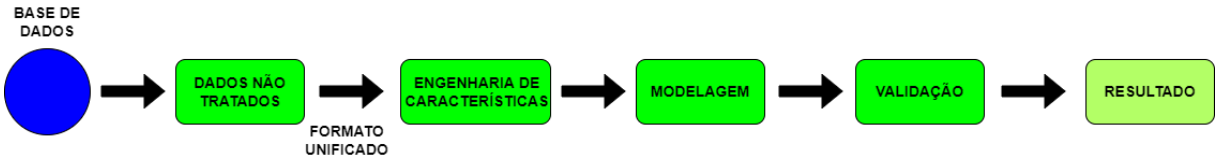
Fonte: Adaptado de Aggarwal *et al.* (2022).

Para Mahesh (2020), o Aprendizado de Máquina é o estudo científico de algoritmos e modelos estatísticos que sistemas de computadores usam para executar uma tarefa específica sem serem programados explicitamente. Esse algoritmos são utilizados para mineração de dados, processamento de imagens, análise preditiva, etc. A principal vantagem do aprendizado de máquina é que, uma vez que o algoritmo aprende o que fazer com os dados, ele pode fazer o trabalho automaticamente.

Ainda, o aprendizado utiliza diferentes tipos de dados para otimizar continuamente os modelos e fazer previsões sob a orientação de algoritmos. A partir desses dados, pode ser

seguido um fluxo de trabalho para chegar nos resultados para as previsões. A Figura 2 mostra um fluxo simplificado do aprendizado de máquina.

Figura 2 – Fluxo de Trabalho do Aprendizado de Máquina



Fonte: Adaptado de Wei *et al.* (2019).

A base de dados pode ser selecionada levando em consideração seu tipo, qualidade e formato. Após a seleção da base de dados, são extraídas as características adequadas para o alvo previsto, mais conhecido como engenharia de características. A extração serve para construir as características dos dados brutos para a aplicação dos algoritmos (WEI *et al.*, 2019).

Com os dados extraídos e bem formatados, é possível construir um modelo para análise. As etapas de modelagem incluem selecionar algoritmos apropriados, treinar a partir de dados de treinamento e fazer previsões precisas (WEI *et al.*, 2019).

Quando o treinamento é concluído, a validação do modelo é aplicada para avaliar a precisão usando os dados não vistos, ou seja, os dados que não foram utilizados para o treinamento. Diversos métodos de aprendizado dividem os dados de entrada em um conjunto de treinamento e um conjunto de testes. Comparado com simulações computacionais, o Aprendizado de Máquina pode identificar padrões em conjuntos de dados de forma eficaz e extrair informações úteis rapidamente (WEI *et al.*, 2019).

Esse processo de identificação de padrões, de acordo com Aggarwal *et al.* (2022), pode ser dividido em três categorias principais:

- **Processo de Decisão:** as técnicas de aprendizado são utilizadas para criar predições ou classificações dos dados, que pode ou não estar dividido em categorias. O objetivo dessas técnicas é estimar um padrão nos dados, o qual auxilia na tomada de decisão.

- **Função de Erro:** esta função faz a previsão de um modelo e decide se são reais ou falsos. Além disso, esta função tem a capacidade de fazer uma comparação para julgar o funcionamento do modelo e verificar o funcionamento das técnicas de Aprendizado de Máquina.

- **Processo de Otimização do Modelo:** se o modelo proposto para implementação se encaixar bem com o conjunto de dados de treinamento, os pesos serão ajustados para reduzir a discrepância entre o trabalho proposto e a estimativa do modelo. A técnica irá se repetir, melhorando o processo de implementação e os pesos de forma independente até que alcance a precisão desejada.

Para esses processos são utilizados algoritmos baseados em distâncias que classificam os dados de teste de acordo com as distâncias calculadas nos dados de treinamento. Os dados que são mais próximos tem uma maior influência na classificação (WALLNÖFER *et al.*, 2020).

As distâncias são conceitos fundamentais no Aprendizado de Máquina e têm efeitos cruciais no desempenho dos algoritmos de aprendizado supervisionado e não supervisionado. Por exemplo, o classificador *KNN* depende de uma função de distância para identificar quais são os vizinhos mais próximos. E o *K-Means*, utiliza as medidas de distâncias para calcular a distância entre pares de dados para o agrupamento (XING *et al.*, 2002).

Na matemática, uma função de distância ou métrica é uma generalização do conceito de distância na física, que se refere a uma medida de tamanho. É um modo de descrever o que significa um elemento no espaço estar próximo ou longe de outro elemento (TROPE; LIBERMAN, 2010).

Para a ciência da computação, há um termo de distância que quantifica a dissimilaridade entre duas palavras, por exemplo “ponto” e “ponta”, a diferença é de apenas uma letra, o que é bem mais próxima de “ponto” e “casco”. Esse termo, conhecido como distância editável, pode ser usado em teoria de códigos e verificador de palavras (RUMMEL, 1976).

Já na geometria, segundo Deza *et al.* (2009), uma métrica de distância, em um determinado espaço de distância (X, d) sendo X com uma certa distância d. Uma função $d : X \times X \to R$ é chamada distância em X se, para todos $x, y \in X$:

- $d(x, y) \geq 0$. (Não negatividade)
- $d(x, y) = d(y, x)$. (Simetria)
- $d(x, x) = 0$. (Reflexividade)
- $d(x, y) \leq d(x, z) + d(y, z)$. (Desigualdade triangular)

Supondo uma quantidade finita de pontos $\{x_i\}_{i=1}^{m} \subseteq R^n$, recebendo informações de que certos pares deles são “semelhantes”:

$S : (x, y) \in$ S se x e y são similares e considerando uma distância de aprendizado, como por exemplo a Euclidiana, é aplicada a seguinte equação:

$$d(x, y) = \|x - y\| = \sqrt{x^2 - y^2} \tag{1}$$

A partir desses conceitos, alguns algoritmos de aprendizado de máquina utilizam distâncias como parâmetros de classificação ou agrupamento de dados. A execução desse aprendizado envolve a criação de um modelo, que é treinado com alguns dados de treinamento e, em seguida, pode fazer previsões com os dados que não foram utilizados no treinamento. Diversos tipos de modelos tem sido utilizados em sistemas de aprendizado de máquina: Redes

Neurais Artificiais, Árvores de Decisão, Máquinas de Suporte de Vetores, Redes Bayesianas, entre outros.

2.1.1 Aprendizado Supervisionado e Aprendizado Não Supervisionado

O aprendizado supervisionado é um paradigma do aprendizado de máquina que tem como relação as informações dos dados de entrada com os de saída em um sistema baseado na quantidade de dados de entrada-saída. O objetivo é construir um sistema artificial que é capaz de aprender um mapeamento entre entrada e saída e possa prever a saída a partir de novos dados. A relação entre as informações de entrada e saída são frequentemente representadas com parâmetros dos modelos de aprendizado (LIU, 2011).

Para resolver um problema do aprendizado supervisionado os seguintes passos são considerados:

- Determinar o tipo dos dados de treinamento.
- Reunir um conjunto de treinamento. Esse conjunto precisa representar um padrão no mundo real.
- Determinar a representação do recurso de entrada da função aprendida. A precisão da função aprendida dependendo fortemente de como o objeto de entrada é representado.
- Determinar a estrutura da função aprendida e o algoritmo de aprendizagem correspondente.
- Executar o algoritmo de aprendizagem no conjunto de treinamento reunido. Alguns algoritmos exigem que o usuário determine certos parâmetros de controle. Esses parâmetros podem ser ajustados otimizando o desempenho em um subconjunto (conjunto de validação) do conjunto de treinamento ou por meio de validação cruzada.
- Avaliar a performance da função aprendida. Após o ajuste e aprendizado dos parâmetros, o desempenho da função resultante deve ser medido em um conjunto de teste separado do conjunto de treinamento.

A relação entrada/saída, segundo Cunningham, Cord e Delany (2008), pode ser representada através de uma função de classificação dada por $f : X \rightarrow Y$, de uma amostra de dados A_n composta por pares de pontos (x, y), sendo x_i pertencente a um conjunto de características X, e $y_i \in Y$:

$$A_n = ((x_1, y_1), ..., (x_n, y_n)) \in (X \times Y)^n \tag{2}$$

onde n representa a dimensão da amostra.

Esse tipo de aprendizado pode aprender certas tarefas do comportamento humano e realizar ações através dos padrões aprendidos. Ainda, como a máquina pode realizar o mapeamento de entradas e saídas muito mais rápido que os humanos, as tarefas são realizadas com mais rapidez e acurácia. Por outro lado, em caso de tarefas mais complexas, por causa das limitações de *hardware* e *software* ainda não é possível igualar o aprendizado humano.

Caso os dados para a realização do aprendizado não tenham saída, ou seja, não tenham sido classificados, é utilizado o aprendizado não supervisionado. A tarefa mais comum desse aprendizado é o agrupamento dos dados de entrada. Nesse caso, o aprendizado aprende padrões na entrada para detectar agrupamentos potencialmente úteis sem classifica-los anteriormente (RUSSELL; NORVIG, 2002).

Essa tarefa pode ser feita por algoritmos de agrupamento e que estão apresentados nas próximas seções. Por primeiro, o algoritmo DBSCAN, baseado em densidade. O próximo, é o *Mean-Shift*, também baseado em densidade mas que tem uma função núcleo que determina o peso dos pontos. Por último, o *K-Means*, iniciando os k agrupamentos com coordenadas aleatórias e atribuindo cada ponto ao agrupamento mais próximo, cada iteração muda os valores dos agrupamentos de acordo com a média de distância calculada até não haver mais a mudança na posição dos agrupamentos (RUSSELL; NORVIG, 2002).

2.2 Algoritmo DBSCAN

Um dos algoritmos de agrupamento é o DBSCAN, publicado na Conferência de Mineração de Dados KDD'96. É baseado em densidade popular implementado independentemente várias vezes e é disponibilizado em kits de ferramenta de agrupamento como o *Scikit-Learn* (SCHUBERT *et al.*, 2017).

As técnicas baseadas em densidade foram introduzidas para o agrupamento de forma arbitrária em banco de dados espaciais com ruído e com um tamanho significativamente grande. Ainda, esse algoritmo aumenta o agrupamento com uma análise de conectividade baseada na densidade (KHAN *et al.*, 2014).

A densidade, segundo Arlia e Coppola (2001), quer dizer que os agrupamentos são definidos como regiões conectadas onde os pontos dados são densos. E caso a densidade não alcance um limite determinado, os dados serão considerados como ruído.

Esse limite de densidade é especificado escolhendo um número mínimo de pontos MinPts em um esfera de raio ϵ. Por definição básica, um ponto central é um ponto que satisfaz a condição dos pontos mínimos da densidade. Para isso um ponto p é escolhido arbitrariamente e uma função de distância é aplicada entre o ponto p e algum outro ponto próximo b, denotada por $distancia(p, b)$. Se a distância for menor ou igual ao raio especificado, o ponto b é incluído na vizinhança de p (ARLIA; COPPOLA, 2001).

Definição 1. *(Vizinhança de um ponto) A vizinhança de um ponto p, denotada por $N(p)$ é definida por* $N(p) = \{q \in D \mid distancia(p, q) \leq \epsilon\}$

Uma abordagem pode exigir que cada ponto em um agrupamento que tenha pelo menos um número de pontos mínimo ($MinPts$) em uma vizinhança. No entanto, essa abordagem falha, pois existem dois tipos de pontos em um agrupamento, os pontos centrais, os quais estão dentro do agrupamento, e os pontos de fronteira, que ficam na borda do agrupamento. Em geral, a vizinhança de um ponto de fronteira possui um número de pontos significantemente menor que a vizinhança de um ponto central. Para isso, podemos definir um número relativamente baixo de $MinPts$ para incluir todos os pontos que pertencem a um mesmo agrupamento.

Esse número, no entanto, não é uma característica para o respectivo agrupamento, principalmente na presença de ruído. Portanto, para cada ponto p em um agrupamento C existe um ponto q em C de modo que p esteja dentro da vizinhança de q e $N(q)$ contenha pelo menos pontos $MinPts$. Para isso, a definição é mostrada a seguir.

Definição 2. *(Diretamente Acessível por Densidade) Um ponto p é diretamente acessível por densidade por um ponto q com respeito a ϵ e $MinPts$ se*

1. $p \in N(q)$ *e*
2. $\mid N(q) \mid \geq MinPts$ *(condição de ponto central)*

Essa definição é simétrica para pares de pontos centrais mas não para um ponto central e outro ponto de fronteira.

Definição 3. *(Acessível por Densidade) Um ponto p é acessível por densidade de um ponto q com respeito a ϵ e $MinPts$ se existe uma cadeia de pontos $p_1, p_2, ..., p_n, p_1 = q, p_n = p$ tal que p_{i+1} é diretamente acessível por densidade por p_i.*

Dois pontos de fronteira de um mesmo agrupamento possivelmente não são densidade alcançável um do outro porque a condição de ponto central pode não ser válida para ambos. Todavia, existe um ponto central a partir do qual ambos os pontos de fronteira do agrupamento sejam alcançáveis pela densidade. Assim, é definido a densidade-conectada que abrange essa relação de pontos de fronteira.

Definição 4. *(Densidade-Conectada) Um ponto p é densidade-conectada a um ponto q com respeito à ϵ e $MintPts$ se existe um ponto o tal que ambos, p e q são acessíveis por densidade por o respeitando o ϵ e $MinPts$.*

Assim, define-se a noção baseada em densidade de um agrupamento. Intuitivamente, um agrupamento é definido como um conjunto de pontos conectados por densidade que é máximo em relação à acessível por densidade. O ruído será definido em relação a um determinado conjunto de agrupamentos. Ruído é um conjunto de pontos do conjunto inteiro que não pertencem a nenhum agrupamento.

Definição 5. *(Agrupamento) Seja D um banco de dados de pontos. Um agrupamento C com respeito a ϵ e $MinPts$ é um sub-conjunto não vazio de D que satisfaz as seguintes condições:*

1. *$\forall\, p, q$: se $p \in C$ e q é acessível por densidade por p com respeito a ϵ e $MinPts$, então $q \in C$. (Maximalidade)*
2. *$\forall p, q \in C$: p é densidade-conectada à q respeitando ϵ e $MinPts$. (Conectividade)*

Definição 6. *(Ruído) Sejam $C_1, ..., C_k$ os agrupamentos da base de dados D respeitando os parâmetros ϵ_i e $MinPts_i$, $i = 1, ..., k$. Então define-se o ruído como um conjunto de pontos na base de dados D não pertencendo a nenhum agrupamento C_i, ou seja, $ruido = \{p \in D \mid \forall i : p \notin C_i\}$.*

Os seguintes lemas são importantes para validar a correção do algoritmo de agrupamento. Dados os parâmetros ϵ e $MinPts$, descobrir um agrupamento em uma abordagem de duas etapas. Primeiro, escolher um ponto arbitrário do banco de dados que satisfaça a condição de ponto central como ponto inicial. Em seguida, recuperar todos os pontos que são alcançáveis por densidade do ponto inicial obtendo o agrupamento que contém o ponto inicial.

Lema 1. *Seja p um ponto em D e $\mid N(p) \mid \geq MinPts$. Então o conjunto $O = \{o \mid o \in D$ e o é alcançável por densidade por p respeitando ϵ e $MinPts\}$ é um agrupamento respeitando ϵ e $MinPts$.*

Cada ponto em C é alcançável por densidade a partir de qualquer um do pontos centrais de C e, portanto, um agrupamento C contém exatamente os pontos que são alcançáveis por densidade a partir de um ponto central arbitrário de C.

Lema 2. *Seja C um agrupamento respeitando ϵ e $MinPts$ e seja p um ponto qualquer $\mid N(p) \mid \geq MinPts$. Então C é igual ao conjunto $O = \{o \mid o \in D$ e o é alcançável por densidade por p respeitando ϵ e $MinPts\}$.*

Idealmente, é necessário saber os parâmetros adequados de ϵ e $MinPts$ de cada agrupamento e pelo menos um ponto do respectivo agrupamento. Então, encontrar todos os pontos que são alcançáveis por densidade através do ponto inicial usando corretamente os parâmetros.

Tendo como base essas informações é possível abstrair o algoritmo DBSCAN de acordo com o Algoritmo 1.

A função de alcance no Algoritmo 1 pode ser definida no Algoritmo 2.

Para uma melhor visualização de como ocorre os passos desse algoritmo, é necessário um conjunto de ponto e os parâmetros ϵ como Eps e $MinPts$. O primeiro passo é escolher um ponto aleatório. A visualização desse conjunto pode ser mostrada na Figura 3.

O próximo passo é definir se esse dado é um ponto central. Para isso, é necessário visitar e contar quantos pontos estão a uma distância de no máximo ϵ em relação ao ponto inicial. Caso a quantidade seja maior ou igual a $MinPts$ esse ponto é marcado como um ponto

Algoritmo 1 – DBSCAN

requer Base de dados b, ϵ, $MinPts$, distância d
1: $C = 0$
2: **para todos** Ponto P em b **faça**
3: Vizinhos $N = alcance(b, d, P, \epsilon)$
4: **se** $|N| < MinPts$ **então**
5: $classe(P) = ruido$
6: continue
7: **finaliza se**
8: $C = C + 1$
9: $S = N \setminus P$
10: **para todos** Ponto Q em S **faça**
11: $Label(Q) = C$
12: $N = Alcance(b, d, Q, \epsilon)$
13: **se** $|N| >= MinPts$ **então**
14: $S = S \cup N$
15: **finaliza se**
16: **finaliza para**
17: **finaliza para**
18: **retorna** Número de agrupamentos C, agrupamentos/clusters S

Fonte: Adaptado de Schubert *et al.* (2017).

Algoritmo 2 – *Alcance - Distância entre dois pontos menor ou igual a* ϵ

requer Base de dados b, ϵ, ponto Q, distância d
1: **para todos** R em b **faça**
2: **se** $distancia(Q, R) <= \epsilon$ **então**
3: $N = N \cup P$
4: **finaliza se**
5: **finaliza para**
6: **retorna** Vizinhos N

Fonte: Adaptado de Schubert *et al.* (2017).

central e é definido o primeiro agrupamento. Os pontos que foram visitados e que atendiam a condição da distância serão verificados novamente.

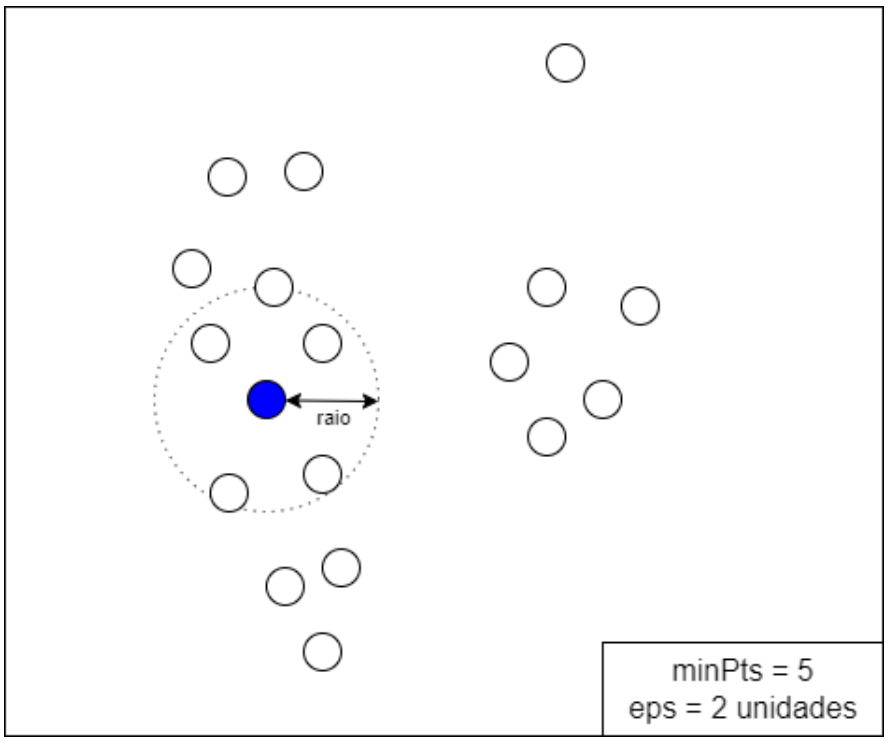
A continuação do algoritmo é mostrada na Figura 4. Os passos anteriores serão repetidos até que todos os pontos sejam verificados. Nesse processo, os pontos que estejam ao alcance do ponto central mas que não atenda as condições de ϵ e $MinPts$ são marcados como pontos de fronteira.

Na Figura 5 é mostrado o passo em que o algoritmo encontra um ponto de fronteira.

Depois que todos os pontos próximos aos pontos centrais foram visitados, é escolhido um ponto aleatório que não tenha sido marcado ainda. Esse processo é repetido até que todos os pontos sejam visitados. Na Figura 6 é observado que o primeiro agrupamento foi definido e que o próximo ponto escolhido pode definir o próximo agrupamento.

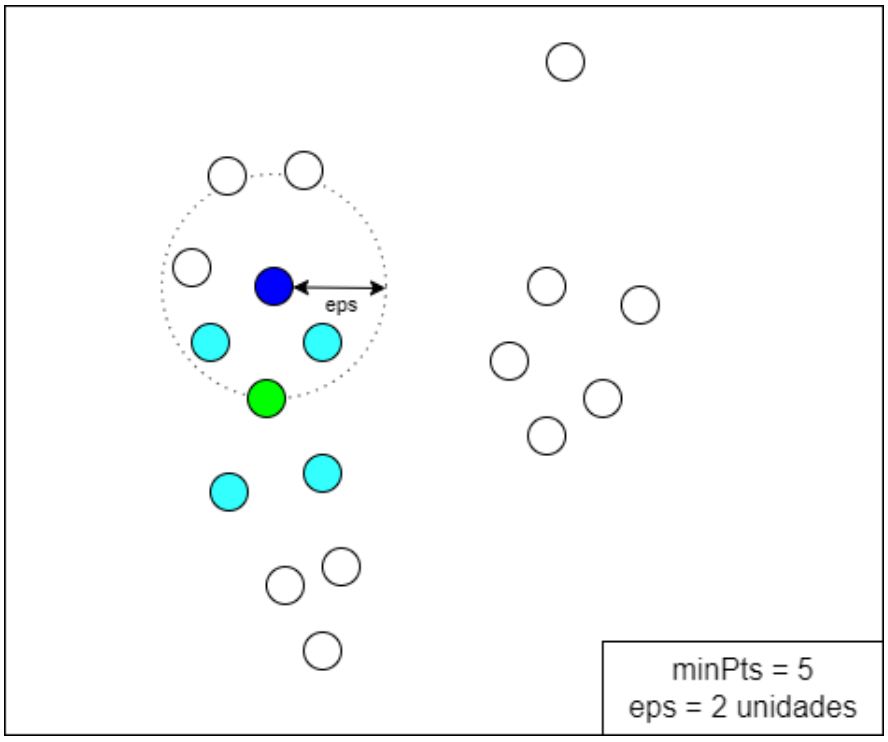
Os pontos em verde são os pontos centrais e os pontos em amarelo são os pontos de fronteira. Os passos se repetem e o resultado após todos os pontos serem visitados pode ser visto na Figura 7. No entanto, ao encontrar um ponto que não atendeu as condições de ϵ e

Figura 3 – Inicialização do algoritmo



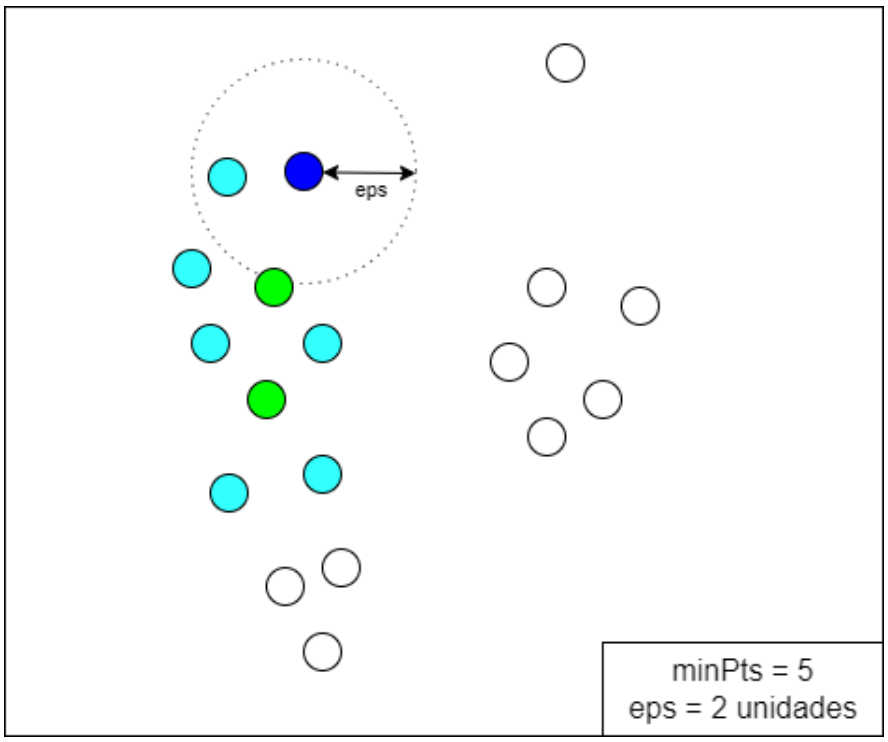
Fonte: Autoria própria (2023).

Figura 4 – Pontos visitados



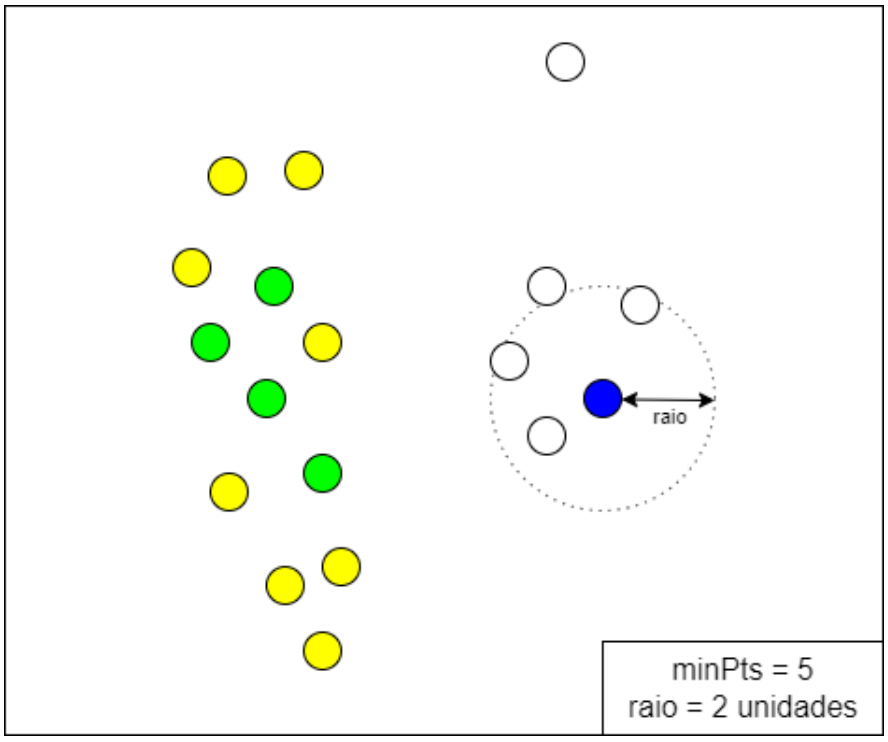
Fonte: Autoria própria (2023).

Figura 5 – Continuação dos pontos visitados (ponto de fronteira)



Fonte: Autoria própria (2023).

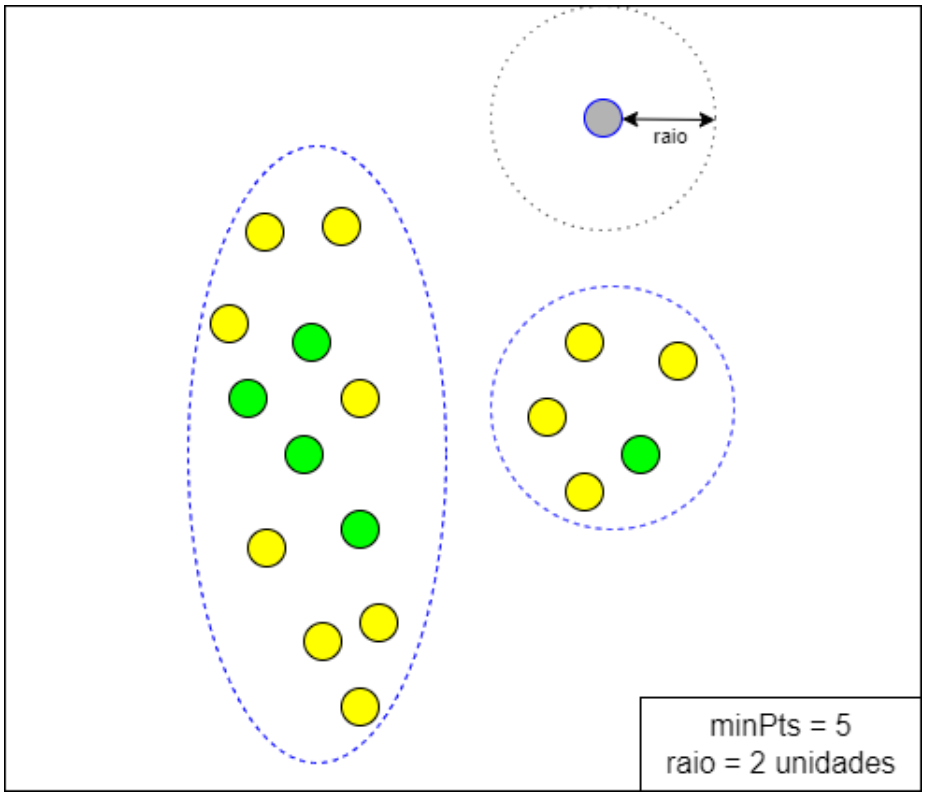
Figura 6 – Continuação dos pontos visitados (próximo grupo)



Fonte: Autoria própria (2023).

$MinPts$ e que não foi alcançado por pontos centrais ou de fronteira, esse ponto é marcado como um ponto de ruído não pertencendo a nenhum agrupamento.

Figura 7 – Grupos encontrados e ruído



Fonte: Autoria própria (2023).

Diante disso, é listado quais são as vantagens na utilização do DBScan:

- Não é necessário definir o número de agrupamentos inicialmente;
- Encontra ruídos no conjunto de dados;
- Requer apenas dois parâmetros e é principalmente insensível à ordenação dos pontos no conjunto de dados;
- Os dois parâmetros, $MinPts$ e ϵ, podem ser definidos por um especialista se o conjunto de dados for bem compreendido.

Além das desvantagens:

- O algoritmo não é totalmente determinístico: os pontos de fronteira que podem ser alcançados a partir de mais de um grupo podem fazer parte de qualquer grupo, dependendo da ordem em que os dados são processados;
- A qualidade do algoritmo depende da medida de distância utilizada na função junto ao ϵ. A distância mais utilizada é a euclidiana;

- DBSCAN não pode agrupar bem conjuntos de dados com grandes diferenças de densidades, uma vez que a combinação de pontos mínimos e ϵ não pode ser escolhida apropriadamente para todos os agrupamentos.

2.3 Algoritmo *K-Means*

O algoritmo de agrupamento *K-Means* agrupa os dados tentando separar as amostras em K agrupamentos com uma variância igual. Esse algoritmo requer que o número de agrupamentos seja especificado. É bem adaptativo com grande número de amostras e tem sido usado em uma grande variedade de áreas de aplicação em muitos campos diferentes (SCULLEY, 2010).

Dada uma base de dados $X = \{x_1, x_2, ..., x_n\}$ de dimensão n em um espaço R^n. Seja $A = \{a_1, a_2, ..., a_c\}$ um número c de agrupamentos. Seja $z = [z_{ik}]_{nxc}$, onde z_{ik} é uma variável binária ($z_{ik} \in \{0, 1\}$) indicando que o dado x_i pertence ao *k-ésimo* agrupamento, $k = 1, 2, ..., c$ e n é a dimensão do dado. A função objetivo do *K-means* é dada por $J(z, A) = \sum_{i=1}^{n} \sum_{k=1}^{c} z_{ik} \times distancia(x_i, a_k)$ (SINAGA; YANG, 2020).

O algoritmo faz as iterações através das condições necessárias para minimizar a função objetivo $J(z,A)$ com equações que atualizam os centros dos agrupamentos e suas associações, respectivamente, como

$$a_k = \frac{\sum_{i=1}^{n} z_{ik} x_{ij}}{\sum_{i=1}^{n} z_{ik}} \text{ e} \tag{3}$$

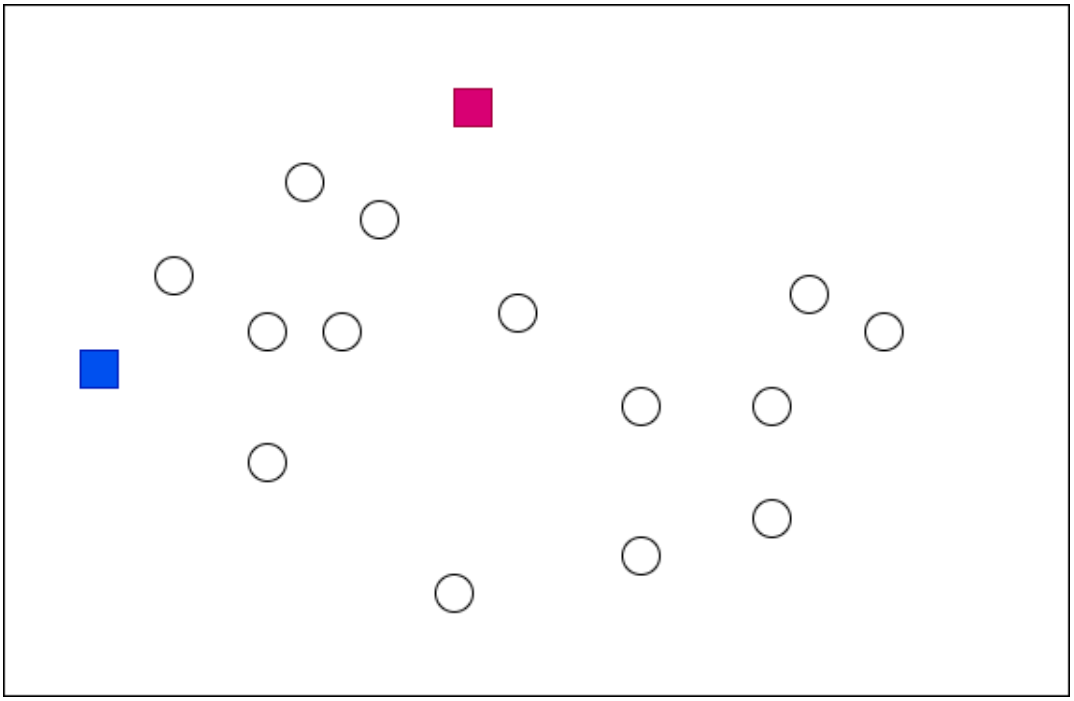
$$z_{ik} = \begin{cases} 1 & \text{se } distancia(x_i, a_k) = \min\limits_{1<=k<=c} distancia(x_i, a_k) \\ 0 & \text{do contrário} \end{cases} \tag{4}$$

onde, $distancia(x_i, a_k)$ é a distância entre o dado x_i e o centro do agrupamento a_k. Porém, determinar o número de agrupamentos ainda é uma dificuldade em aplicações reais. Outro problema, é que a inicialização das posições desses agrupamentos também afeta o algoritmo (SINAGA; YANG, 2020).

Esse algoritmo consiste de três passos a partir da escolha das amostras da base de dados. O primeiro passo é escolher os agrupamentos iniciais. A posição inicial de cada agrupamento pode ser definida aleatoriamente. Assim, na Figura 8 será feito um exemplo ilustrativo apresentando o conceito de agrupamento, e então é escolhido dois agrupamentos.

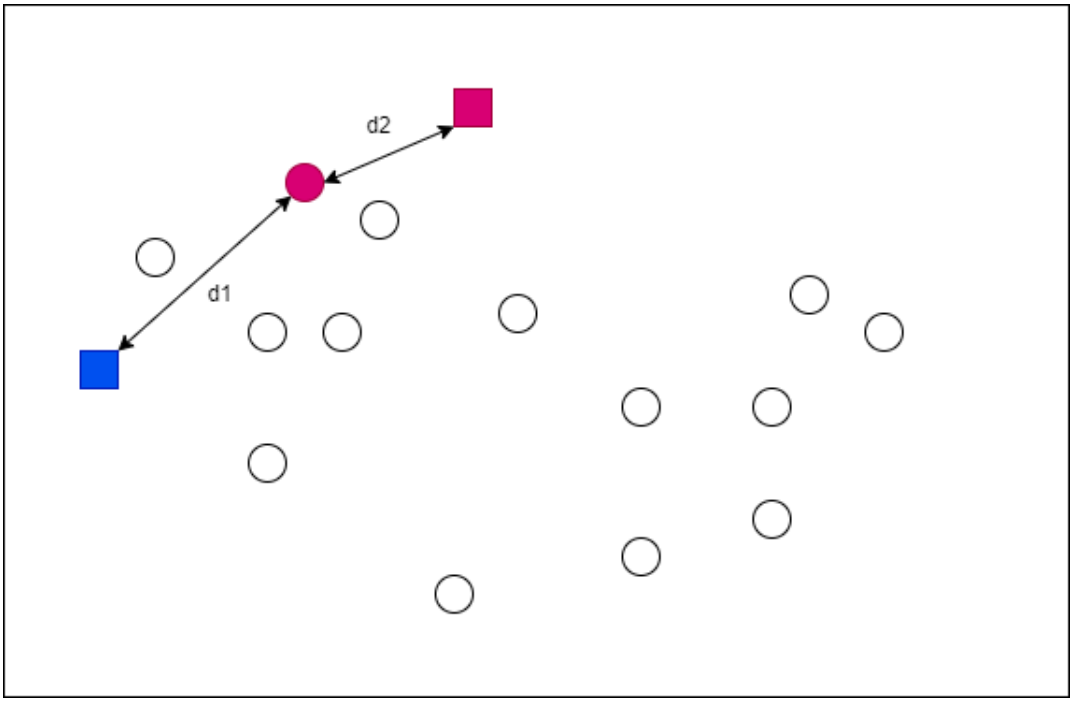
No segundo passo, é escolhido um ponto e é calculada a distância até os dois agrupamentos. Esse ponto irá pertencer ao agrupamento mais próximo. Pode ser visto na Figura 9 que o ponto escolhido pertence ao agrupamento vermelho.

Figura 8 – Posições iniciais de cada agrupamento escolhido



Fonte: Autoria própria (2023).

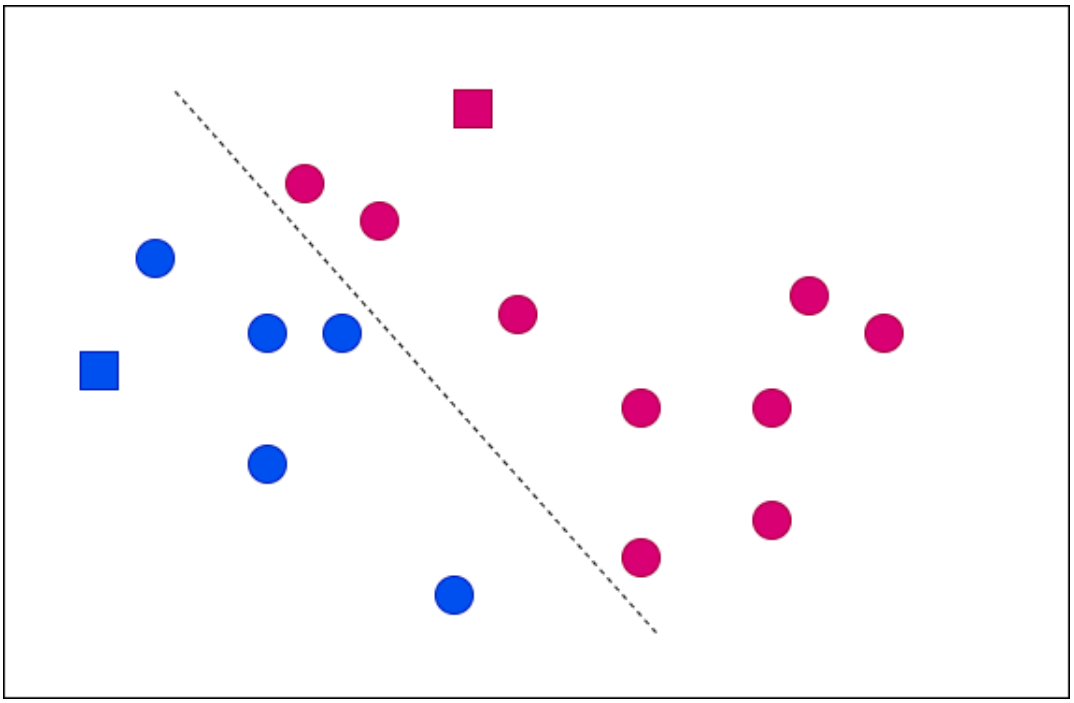
Figura 9 – Ponto pertencente ao agrupamento mais próximo



Fonte: Autoria própria (2023).

Esse cálculo é feito até que todos os pontos pertençam a um agrupamento podendo ser observado na Figura 10.

Figura 10 – Cada ponto é atribuído ao agrupamento mais próximo



Fonte: Autoria própria (2023).

O próximo e último passo é criar novos agrupamentos levando em conta o valor médio de todas as amostras atribuídas a cada agrupamento anterior. A diferença entre o agrupamento antigo e o novo é calculada e o algoritmo repete essas duas últimas etapas até que esse valor seja menor que um limite (Figura 11). Ou seja, o algoritmo repete esses dois últimos passos até que os agrupamentos não se movam significativamente.

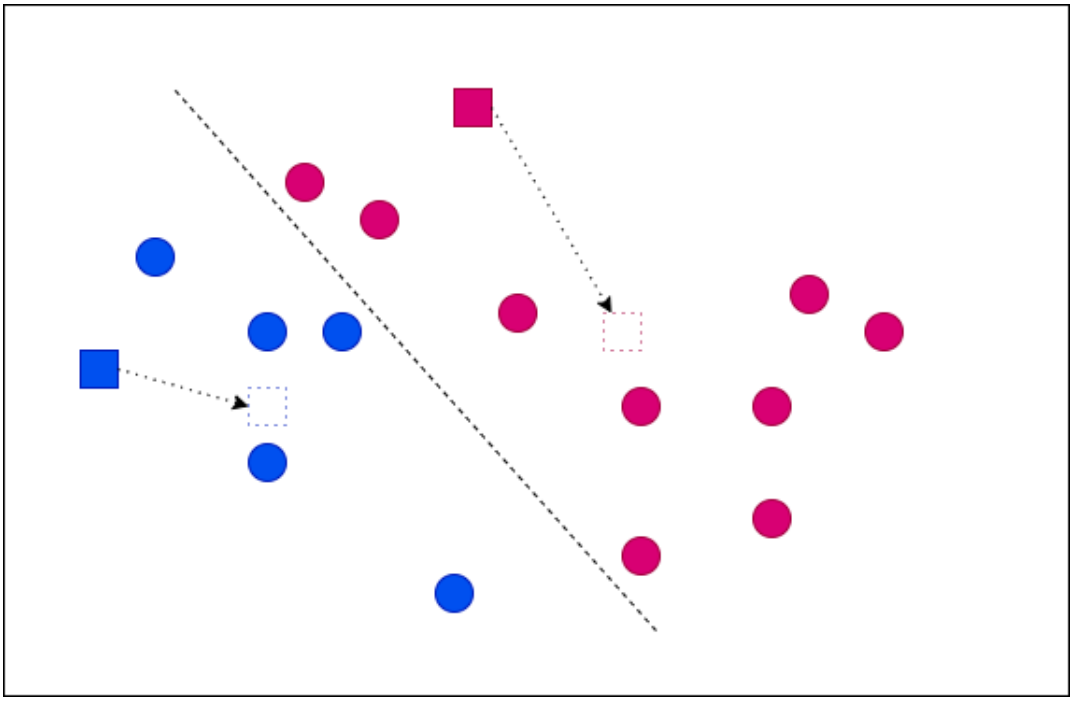
Assim, o resultado do último passo aplicado ao exemplo pode ser visto na Figura 12.

Sendo assim, as vantagens desse algoritmo são: fácil implementação e computacionalmente rápido devido ao número de variáveis. E as desvantagens são que é difícil prever o número de agrupamentos e o resultado final pode ser fortemente impactado pelos centroides iniciais.

2.4 Algoritmo *Mean-Shift*

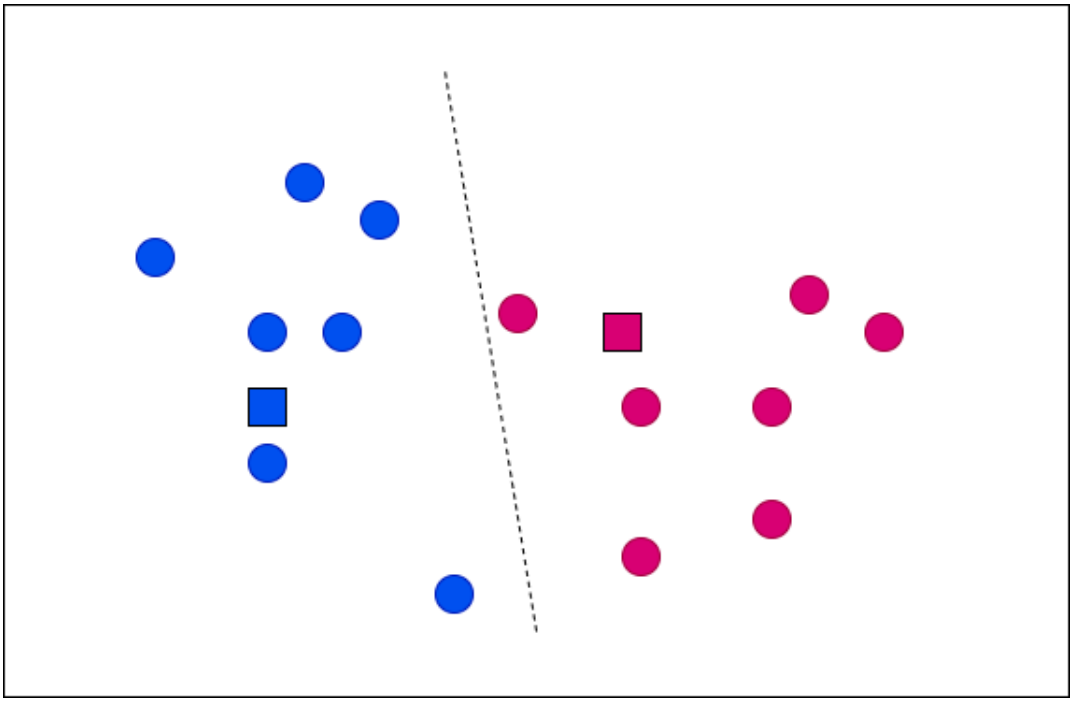
O *Mean-Shift*, ou deslocamento médio, é uma técnica de análise matemática de espaço de características não paramétricas para localizar os máximos de uma função de densidade. Os domínios de aplicação incluem análise de agrupamento computacional e processamento de imagens (CHENG, 1995). Além disso, esse é um método iterativo e começa com um ponto estimado x. Dada uma função do núcleo $K(x_i - x)$. Essa função determina o peso dos pontos próximos para estimar novamente a média.

Figura 11 – Os agrupamentos são deslocados para a média das distâncias dos pontos



Fonte: Autoria própria (2023).

Figura 12 – Novos agrupamentos definidos a partir da média dos pontos de cada agrupamento



Fonte: Autoria própria (2023).

De acordo com Fukunaga e Hostetler (1975), a diferença $m(x) - x$ é chamada de deslocamento médio. O algoritmo *Mean-Shift* agora define $x \leftarrow m(x)$, e repete a estimativa até que $m(x)$ convirja.

A cada iteração do algoritmo, $s \leftarrow m(s)$ é executado para todos $s \in S$ simultaneamente. A primeira questão, então, é como estimar a função densidade dado um conjunto esparso de amostras. Uma das abordagens mais simples é apenas suavizar os dados, por exemplo, convolucionando-os com um núcleo fixo de largura de banda h,

$$f(x) = \sum_i K(x_i - x) = \sum_i k\left(\frac{distancia(x, x_i)}{h^2}\right) \tag{5}$$

onde x_i são as amostras de entrada e $k(r)$ é a função do núcleo. h é o único parâmetro no algoritmo e é chamado de largura de banda. Essa abordagem é conhecida como estimativa de densidade do núcleo. Uma vez calculado o $f(x)$ da função acima, é possível encontrar o máximo local usando o gradiente ascendente ou alguma outra técnica de otimização (SZELISKI, 2010).

Definição 7. *Definição de Núcleo: Seja X um espaço de dimensão n, R^n. A norma de x é um número não negativo, $||x||^2 = x^T x \geq 0$. Uma função $K : X \to R$ é considerada um núcleo se existir um perfil, $k : [0, \inf] \to R$, tal como, $K(x) = k(||x||^2)$ e*

- *k é não negativo*
- *k é não incremental: $k(a) \geq k(b)$ se $a < b$*
- *k é contínua por partes e $\int_0^\infty k(r)dr < \infty$*

Os dois perfis de núcleos mais utilizados para o deslocamento médio são:
Núcleo Plano

$$k(x) = \begin{cases} 1 & \text{se } x \leq \lambda \\ 0 & \text{do contrário } x > \lambda \end{cases} \tag{6}$$

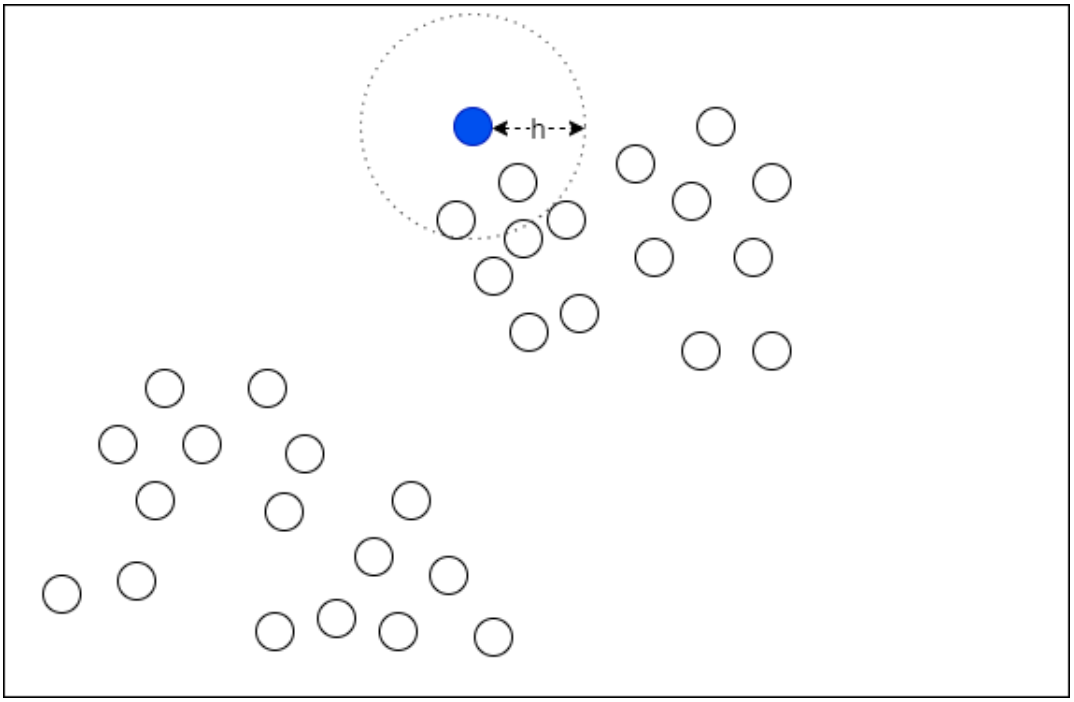
Núcleo Gaussiano

$$k(x) = \epsilon^{-\frac{x}{2\sigma^2}} \tag{7}$$

onde o parâmetro de desvio padrão σ funciona como parâmetro de largura de banda, h (SZELISKI, 2022).

Os passos que consistem esse algoritmo podem ser demonstrados através do exemplo a seguir. Ao definir a largura de banda h é escolhido um ponto aleatório da base de dados (Figura 13). Essa largura de banda pode ser representada por um circulo, o qual seu centro é o ponto escolhido.

Figura 13 – Escolhendo um ponto inicial



Fonte: Autoria própria (2023)..

O próximo passo é calcular a média ponderada desse ponto utilizando a Equação 5. Essa média engloba todos os pontos que estão dentro do círculo. E depois, deslocar esse círculo até a posição da média encontrada. O deslocamento pode ser visto na Figura 14.

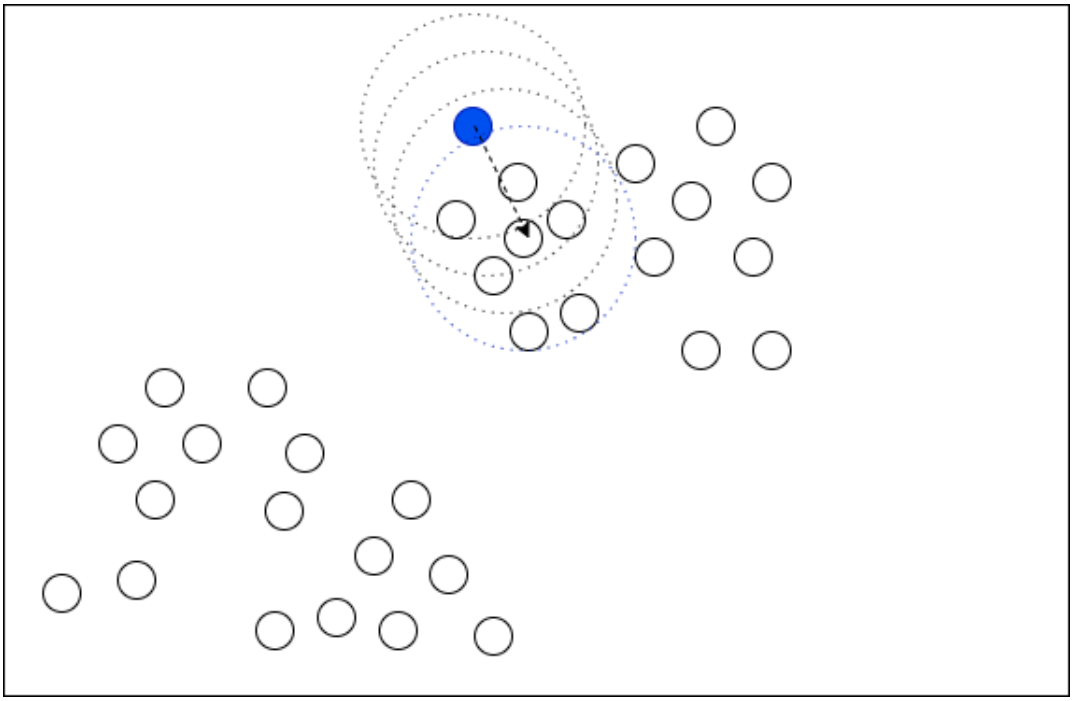
O deslocamento desse circulo irá parar quando atingir o local com mais densidade de ponto. Assim que atingiu o local, é escolhido outro ponto e repetido os passos anteriores. Assim, na Figura 15 é mostrado onde o círculo ficou.

O algoritmo repete o primeiro passo escolhendo aleatoriamente um outro ponto que não havia sido escolhido. A partir do momento que um ponto escolhido e utilizando a função de densidade, não desloque seu centro até o mesmo ponto de maior densidade encontrado anteriormente, é definido um novo agrupamento. Então o deslocamento é feito para o ponto que tem mais densidade. As posições dos pontos mais densos são consideradas as posições dos agrupamentos. A partir disso, o primeiro grupo foi definido e o ponto que não foi visitado ainda será escolhido para o cálculo da média ponderada Figura 16.

Repetindo os passos anteriores é observado o resultado do segundo grupo na Figura 17.

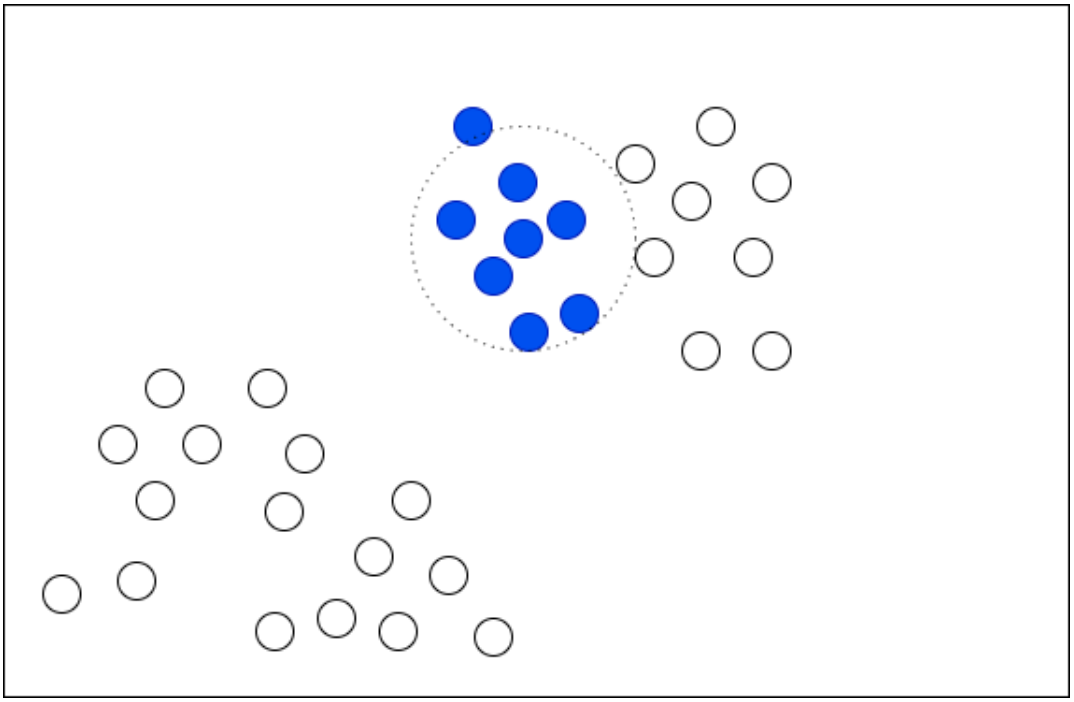
Esse processo se repete para todos os pontos até que os círculos, definidos pela largura de banda, não se movam mais. Assim, o resultado desse exemplo é mostrado na Figura 18

Figura 14 – Deslocamento do círculo



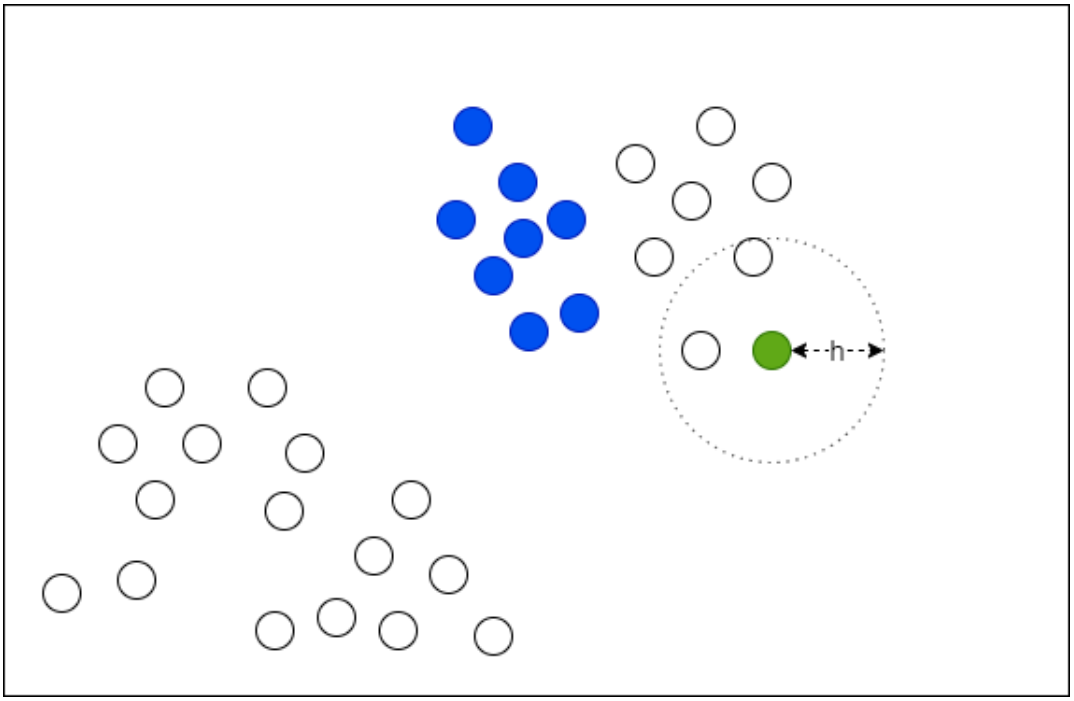
Fonte: Autoria própria (2023).

Figura 15 – Primeiro grupo definido



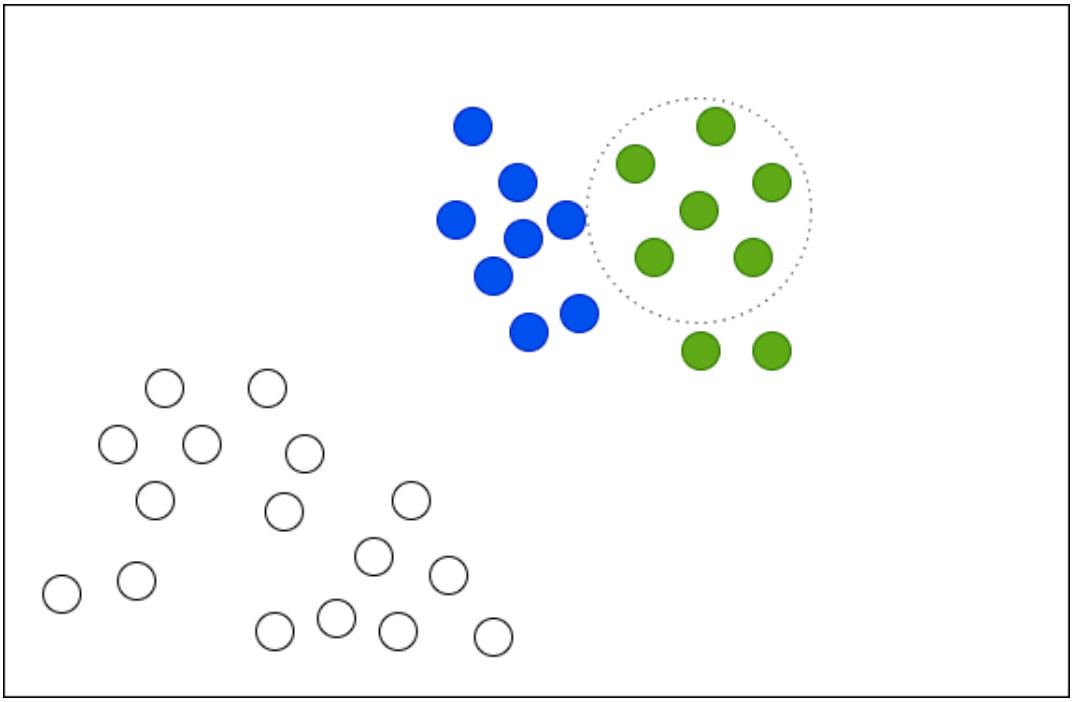
Fonte: Autoria própria (2023).

Figura 16 – Procurar pelo próximo grupo



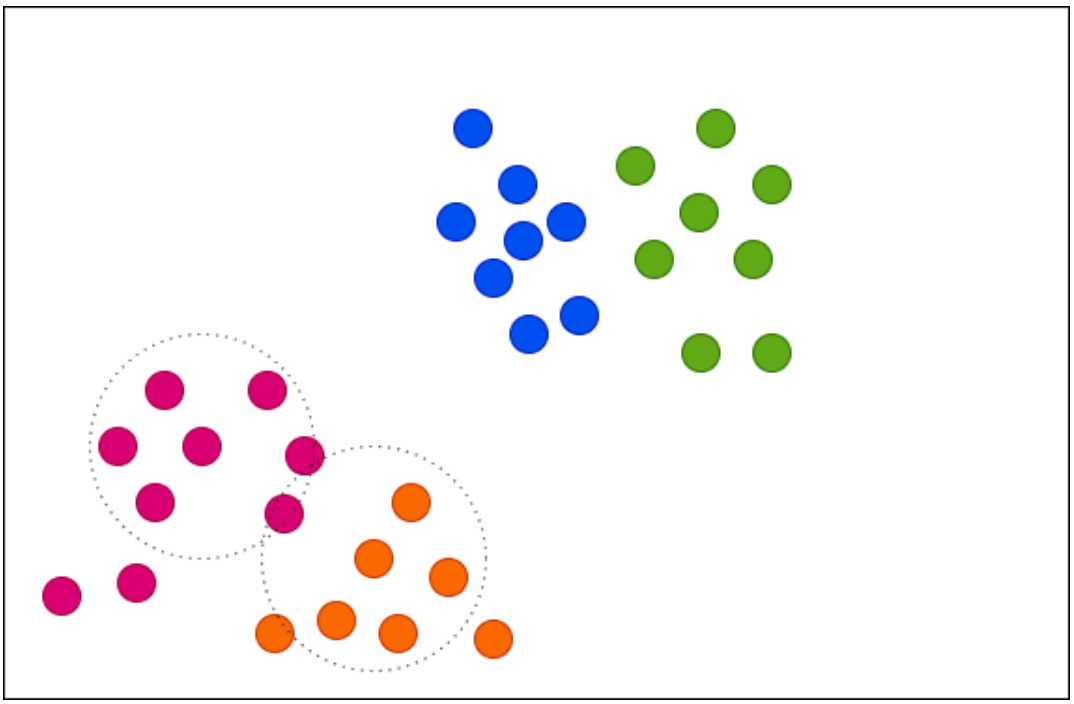
Fonte: Autoria própria (2023).

Figura 17 – Segundo grupo definido



Fonte: Autoria própria (2023).

Figura 18 – Todos os grupos definidos



Fonte: Autoria própria (2023).

Tendo como base essas informações é definido os passos a seguir para a implementação do algoritmo:

1. Escolher um ponto qualquer e definir um raio em torno desse ponto;
2. Calcular a média de todos os pontos que estão dentro do raio;
3. Deslocar o raio de modo que o centro fique na média calculada no passo anterior;
4. Repetir os Passos 2 e 3 até que o raio não seja mais deslocado, ou seja, até a média dos pontos convergir.

Esse convergência irá ser alcançada depois que não tiver mais pontos a serem adicionados dentro desse raio.

2.4.1 Vantagens

- Deslocamento médio é uma ferramenta independente de aplicação adequada para análise de dados reais;
- Não assume nenhuma forma predefinida em agrupamentos de dados;
- É capaz de lidar com espaços de recursos arbitrários;
- O procedimento baseia-se na escolha de um único parâmetro: a largura de banda;

- A largura de banda h tem um significado físico, ao contrário do algoritmo *K-means*.

2.4.2 Desvantagens

- A escolha da largura de banda não é trivial;
- O tamanho inadequado da largura de banda pode fazer com que os agrupamentos sejam mesclados ou gerar agrupamentos adicionais "rasos";
- Muitas vezes requer o uso de largura de banda h variável.

2.5 Comparação entre os três algoritmos

Existem diferentes algoritmos utilizados para agrupar dados com diferentes propriedades, cada problema é diferente e cada algoritmo produzirá resultados diferentes. Para a análise dos algoritmos em cada base será necessário avaliar as vantagens e desvantagens de cada algoritmo.

Tabela 1 – Comparação entre os três algoritmos adaptado de Seif (2018)

	Número de Agrupamentos	**Manipulação de Ruído**	**Complexidade de Tempo**	**Limites**
K-means	Pré-determinado	Não	$O(n)$	Não
Mean-Shift	Baseado em Dados	Sim	$O(n^2)$	Sim (largura de banda)
DBSCAN	Baseado em Dados	Sim	$O(n \log n)$	Sim (distância e pontos mínimos)

Fonte: Autoria própria (2023).

- **Número de Agrupamentos:** Alguns algoritmos de agrupamento exigem que o número de agrupamentos seja conhecido a posteriori e, então, alocarão todos os pontos de dados para um determinado número de agrupamentos. No entanto, neste ponto de análise, não se tem uma estimativa do número de agrupamentos que poderiam ser derivados das atividades.
- **Manipulação de Ruídos:** Nem todas as amostras no determinado conjunto de dados necessariamente é considerado parte de um agrupamento. Para alguns algoritmos é melhor retirar os ruídos do conjunto de dados antes de aplicar o agrupamento. Os algoritmos DBSCAN e *Mean-Shift* tratam os dados de ruídos na hora de utilizar os limites como parâmetros de agrupamento.

- **Complexidade de Tempo:** Apesar de alguns conjuntos de dados serem pequenos e a diferença no tempo de execução dos algoritmos serem pequenas, a estimativa de tempo em conjuntos muito grandes de dados podem ser bem diferentes entre os três algoritmos.
- **Limites:** Alguns algoritmos de agrupamento fornecem parâmetros com os quais orientam o processo de agrupamento e definem alguns limites para as bordas do agrupamento.

2.6 Distâncias

Do ponto de vista científico e matemático, distância é definida como um grau quantitativo de quão distantes dois objetos estão. A escolha de uma distância depende do tipo de medida ou da representação dos objetos (CHA, 2007). Muitas tarefas comuns na Ciência de Dados são baseadas em distâncias entre entidades (LIBERTI, 2020).

A escolha de uma distância é um bom passo nos algoritmos de agrupamento para determinar o cálculo de quão próximo dois objetos estão. Esse passo influencia na forma do agrupamento, pois alguns objetos podem estar próximos um do outro com uma determinada distância e podem estar distantes com outra. Várias distâncias podem ser utilizadas para agrupamento de dados, dentre elas, *Canberra*, *Chebyshev*, *Euclidiana* e *Minkowski* (PANDIT; GUPTA *et al.*, 2011). Ainda, há uma distância intermediária proposta por Rodrigues (2018) que combina as distâncias de *Minkowski* e *Chebyshev*.

2.6.1 *Canberra*

Essa distância é uma versão com pesos da distância de *Manhattan* inicialmente proposta por Lance e Williams (1966).

$$d(P,Q) = \sum_{k=1}^{n} \frac{|x_k - y_k|}{|x_k| + |y_k|} \tag{8}$$

2.6.2 *Chebyshev*

O cálculo dessa distância normalmente é conhecido como a métrica máxima na matemática e segundo Abello, Pardalos e Resende (2013), mede a distância entre dois pontos como a diferença máxima sobre qualquer um dos seus valores de eixo. Em um espaço de duas dimensões, por exemplo, dado dois pontos $P = (x_1, x_2)$ e $Q = (y_1, y_2)$, a distância de *Chebyshev* entre eles é:

$$d(P,Q) = maximo(|x_1 - y_1|, |x_2 - y_2|) \tag{9}$$

E, se o espaço tiver n dimensões, considerando os pontos $P(x_1, x_2, ..., x_n)$ e $Q(y_1, y_2, ..., y_n)$:

$$d(P,Q) = maximo(|x_1 - y_1|, |x_2 - y_2|, ..., |x_n - y_n|) \tag{10}$$

2.6.3 *Euclidiana*

Essa distância representa a linha reta entre dois pontos *P* e *Q* em um plano cartesiano, sendo *P* em (x_1, x_2) e *Q* em (y_1, y_2), a distância entre elas é dada por:

$$d(P,Q) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{11}$$

Se os pontos tiverem dimensão *N* a equação da distância é dada por:

$$d(P,Q) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2} \tag{12}$$

onde p_i e q_i são as coordenadas de p e q na dimensão *i* (TABAK, 2014).

2.6.4 *Minkowski*

A distância de *Minkowski* é métrica de distância generalizada. E o cálculo se dá pela seguinte equação:

$$d(P,Q) = \left(\sum_{k=1}^{n} |x_k - y_k|^P \right)^{\frac{1}{P}} \tag{13}$$

Note que quando o $P = 2$, essa distância se torna a distância *Euclidiana*. A distância de *Chebyshev* é uma variação quando $P = \infty$ (tomando um limite). Ainda, pode-se tomar valores ordinais e quantitativos para os pontos (RODRIGUES, 2018).

2.6.5 Rodrigues

Rodrigues (2018) apresenta uma nova distância. Essa distância é uma combinação das distâncias de *Chebyshev* e *Minkowski*, a qual apresenta a seguinte equação:

$$d(P,Q) = maximo(|x_1 - y_1|, |x_2 - y_2|, ..., |x_n - y_n|) + \left(\sum_{k=1}^{n} |x_k - y_k|^P\right)^{\frac{1}{P}} \quad (14)$$

2.7 Representação de dados

As bases de dados são conjuntos de informações específicas, tais como dados de alunos em uma universidade. Essas informações são organizadas de uma forma que se possa entender e ter sentido em pesquisas ou estudos científicos e podem ser divididas em atributos, como as informações de notas e idades dos alunos (ULLMAN, 2007). Cada informação de cada aluno pode ser chamada de amostra, a qual se refere a um subconjunto da base de dados (BISWAS, 2016). Os atributos podem ser definidos como a característica de cada valor na amostra.

No entanto, ao utilizar bases de dados, alguns valores podem estar faltando ou ainda apresentar valores inconsistentes, o que pode afetar no desempenho dos algoritmos de agrupamento (ALASADI; BHAYA, 2017). Ainda, para uma melhor extração de características, é possível transformar os atributos categóricos em numéricos para a aplicação das distâncias (AHMAD; DEY, 2007).

Diante disso, é necessário o pré-processamento das bases de dados em relação a seus atributos, os quais podem ser divididos em categóricos, inteiros, reais e séries temporais. Esse pré-processamento tem como objetivo transformar os atributos não numéricos de cada amostra em atributos numéricos, preencher atributos faltantes nas amostras e remover amostras redundantes (MALLEY; RAMAZZOTTI; WU, 2016).

As bases de dados multivariáveis consistem de três ou mais tipos de variáveis, ou seja, consistem de medidas individuais que são adquiridas por uma função de três ou mais variáveis. Já as bases de série temporal referenciam uma sequência de dados que são indexados em ordem temporal. Normalmente, esse tipo de base é constituída de amostras registradas em intervalos de tempo consistentes. Além disso, os atributos Inteiro e Real são considerados dados numéricos. Os atributos categóricos são em formas de características não numéricas.

Um exemplo de base de dados, possuindo apenas três atributos, é a base *Haberman*, a qual representa um estudo realizado pela Universidade de Chicago sobre pacientes que sobreviveram a cirurgia de câncer de mama (DUA; GRAFF, 2017). Os atributos representam a idade do paciente, o ano de operação e o número de nódulos auxiliares positivos detectados.

Essa base classifica os pacientes que sobreviveram mais de 5 anos ou morreram antes de completarem 5 anos depois da cirurgia.

Dos 306 pacientes, 81 deles morreram antes de completar 5 anos após cirurgia e 225 sobreviveram mais de 5 anos. Dos que morreram antes de completar os 5 anos, 22 pacientes tinham mais de 60 anos e 55 pacientes que tinham mais de 60 anos viveram mais que 5 anos. Na Tabela 2 é mostrada a relação da idade.

Tabela 2 – Dados dos pacientes em relação a idade

Pacientes	5 anos ou mais	Menos de 5 anos	Total
Mais que 60 anos	55	22	77
Até 60 anos	170	59	229
Total	225	81	306

Fonte: Autoria própria (2023).

Outras informações podem ser retiradas dessa base, as quais mostram a classificação dos pacientes em relação ao número de nódulos auxiliares positivos detectados e podem ser vistos na Tabela 3. Em um total de aproximadamente 45% que não tiveram nódulos detectados apenas 14% deles morreram antes dos 5 anos. Além disso, dos 55% que tiveram pelo menos um nó detectado e não alcançaram os 5 anos pós cirurgia, a porcentagem é maior.

Tabela 3 – Dados dos pacientes em relação ao número de nós

Pacientes	5 anos ou mais	Menos de 5 anos	Total
Nenhum nó auxiliar	117	19	136
Um ou mais nós auxiliares	108	62	170
Total	225	81	306

Fonte: Autoria própria (2023).

Outro exemplo de base é a Iris contida no repositório da UCI, que está balanceada em 50 amostras para cada classe, contendo 3 classes. Pode-se observar algumas características na tabela 4.

Tabela 4 – Tamanhos Máximo e Mínimo das Plantas

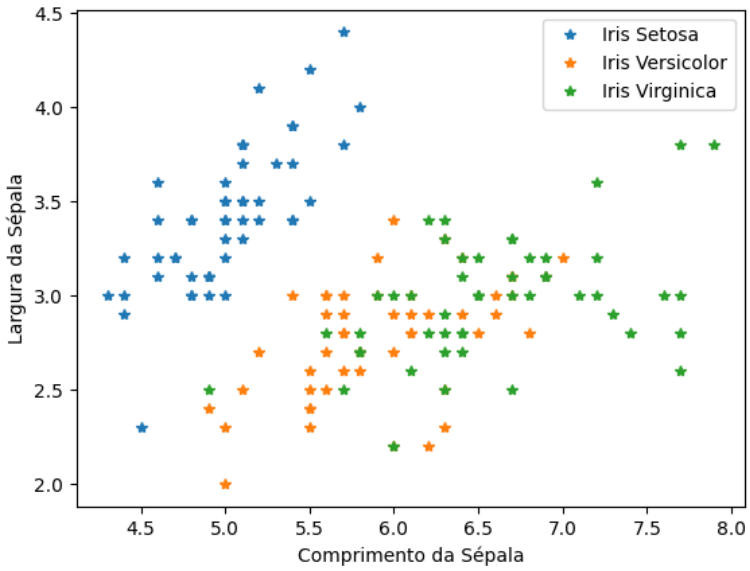
	Tamanho	Comprimento Sépala	Largura Sépala	Comprimento Pétala	Largura Pétala
Íris Setosa	Máximo	5.8 cm	4.4 cm	1.9 cm	0.6 cm
	Mínimo	4.3 cm	2.3 cm	1.0 cm	0.1 cm
Íris Versicolor	Máximo	7.0 cm	3.4 cm	5.1 cm	1.8 cm
	Mínimo	4.9 cm	2.0 cm	3.0 cm	1.0 cm
Íris Virgínica	Máximo	7.9 cm	3.8 cm	6.9 cm	2.5 cm
	Mínimo	4.9 cm	2.2 cm	4.5 cm	1.4 cm

Fonte: Autoria própria (2023).

As sépalas protegem os botões das flores e são o suporte para as pétalas. As pétalas são folhas modificadas que envolvem a parte reprodutiva da flor (BEENTJE *et al.*, 2010).

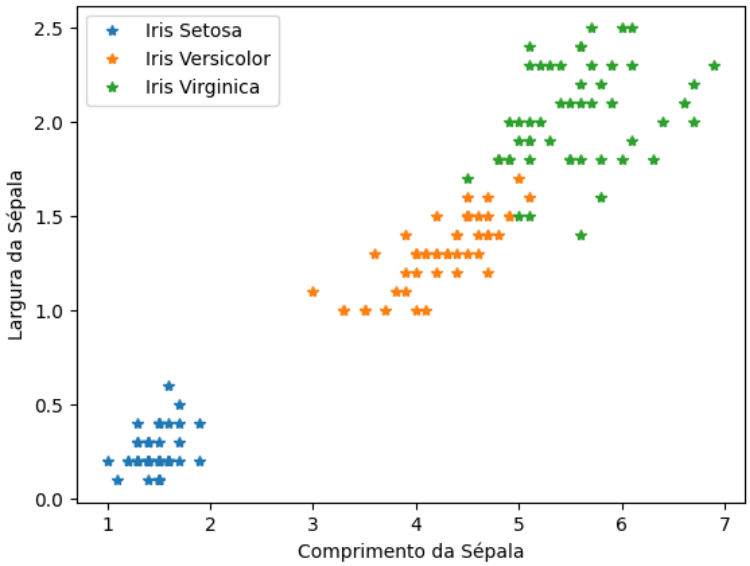
Além disso, na Figura 19 a classe Iris Setosa pode ser linearmente separada das outras através do comprimento e largura da sépala. E também, pelo comprimento e largura da pétala podendo ser visto na Figura 20.

Figura 19 – Separação da classe Iris Setosa comparando a largura e o comprimento da sépala



Fonte: Autoria própria (2023).

Figura 20 – Separação da classe Iris Setosa comparando a largura e o comprimento da pétala



Fonte: Autoria própria (2023).

Após a coleta das bases, há o pré-processamento dos atributos categóricos. Os quais passarão por duas codificações diferentes conforme o tipo do atributo. Para o atributo categórico

que não possui uma ordem, como por exemplo a base da Iris, pode ser utilizada a codificação *One Hot*, transformando cada valor do atributo em um vetor binário. E para os que possuem ordem, pode ser utilizada a codificação de classes, atribuindo valores inteiros entre 0 e quantidade de classes - 1 de acordo com a ordem.

A codificação *One Hot* pode ser aplicada, por exemplo, na base de dados de diabetes, a qual possui um atributo de medição da glucose no sangue. Essa medição foi feita de acordo com os horários de café da manha (08:00), almoço (12:00), jantar (18:00) e hora de dormir (22:00). Um exemplo pode ser observado na Tabela 5.

Tabela 5 – Exemplo de Atributo Categórico

Paciente	**Medida**
Paciente 1	Café
Paciente 2	Almoço
Paciente 3	Jantar
Paciente 4	Hora de Dormir

Fonte: Autoria própria (2023).

Aplicando a codificação *One Hot* na Tabela 5 é observada a transformação do atributo categórico em quatro atributos numéricos mostrados na Tabela 6.

Tabela 6 – Exemplo de Codificação *One Hot*

	Medida			
Paciente	**Café**	**Almoço**	**Jantar**	**Hora de Dormir**
Paciente 1	1	0	0	0
Paciente 2	0	1	0	0
Paciente 3	0	0	1	0
Paciente 4	0	0	0	1

Fonte: Autoria própria (2023).

3 TRABALHOS RELACIONADOS

Há alguns trabalhos na literatura relacionado a comparação de diferentes distâncias aplicadas em algoritmos supervisionados e não supervisionado. Em uma comparação com 14 distâncias, perlibakas2004distance mostrou que em um método de reconhecimento facial baseado na Análise de Componente Principal (PCA) a distância simplificada de *Mahalanobis* teve o melhor resultado. Ainda, propondo uma nova distância baseada na soma dos erros quadráticos, obteve um resultado em que essa distância ficou entre as três melhores, utilizando uma base de dados contendo fotos de 423 pessoas.

Ainda, awasthi2013empirical utilizou o algoritmo de agrupamento, *K-Means*, para comparar as distâncias *Euclidiana* e *Manhattan* em uma base de dados bancária pré-processada. Utilizando 2, 5 e 6 como o valor de agrupamento (k), mostrou que a distância de *Manhattan* teve melhores tempos de execução e soma dos erros quadráticos melhores que a distância Euclidiana em uma base de dados com 600 amostras e 11 atributos.

kokare2003comparison comparou nove medidas de similaridade de imagens, dentre elas, *Manhattan*, Euclidiana, *Chebyshev*, *Mahalanobis*, *Canberra*, *Bray-Curtis*. Utilizando quase duas mil imagens extraídas de uma base de dados com filtro de *Gabor*, obteve o melhor resultado com a distância de *Canberra* e *Bray-Curtis*.

Em relação a algoritmos supervisionados, rodriguescombining propôs uma distância que combina as distâncias *Minkowski* e *Chebyshev* e realizou uma comparação em relação a outras distâncias utilizando o algoritmo *KNN*. Esse estudo mostrou que com sua nova distância, o algoritmo *KNN* obtém um tempo de execução eficiente e também boas acurácias. A análise de acurácia também mostrou que, utilizando um total de 27 bases de dados do repositório da UCI (DUA; GRAFF, 2017) combinadas com as distâncias de *Chebyshev*, *Manhattan*, *Euclidiana*, *Canberra*, Soma das Diferenças dos Quadrados, variações nos parâmetros da distância de *Minkowski*, a distância de Rodrigues foi melhor que a média em 26 bases de dados e obteve a melhor acurácia em 9 das 27 bases.

Portanto, algumas combinações de distâncias e valores de k para o algoritmo *KNN* se destacam em relação a outras combinações. Esse é um problema conhecido na otimização de parâmetros. A melhor combinação varia de acordo com o problema de classificação e se torna difícil antecipar uma combinação sem testes e análises de desempenho.

4 MATERIAIS E MÉTODOS

Neste capítulo são apresentados os passos para o desenvolvimento da análise dos algoritmos de agrupamento variando as bases, as distâncias e seus atributos. O Chapter 3 apresenta o passo-a-passo de como foram medidos os tempos e a taxa de acerto dos agrupamentos. Para cada base b (de um total de 11 bases), e para cada distância d (de um total de 10), foi realizado o agrupamento utilizando o algoritmo *DBSCAN* (Δ), e o algoritmo *Mean-Shift* (M). Para cada um desses dois algoritmos foi medido o tempo de execução ($t\Delta_{b,d}$ e $tM_{b,d}$). Na sequência, a quantidade de agrupamentos que cada um desses dois algoritmos retornaram ($k\Delta_{b,d}$ e $kM_{b,d}$), bem como a posição de cada agrupamento ($c\Delta_{b,d}$ e $cM_{b,d}$), foram armazenados.

Na sequência, foi executado o algoritmo *K-Means* duas vezes. Uma para a quantidade de agrupamentos do DBSCAN ($k\Delta_{b,d}$) e outra para a quantidade de agrupamentos do *Mean-Shift* ($kM_{b,d}$). Como anteriormente, foram calculados os tempos de execução ($tK_{\Delta,b,d}$ e $tK_{M,b,d}$), e os agrupamentos.

Para cada distância d, foi feita a média de tempo obtido considerando todas as bases. Isto é:

- $t\Delta_d$ é a média de tempo de cada distância d para o DBSCAN;
- $tK_{\Delta d}$ é a média de tempo de cada distância d para o *K-Means* usando a quantidade de grupos do DBSCAN;
- tM_d é a média de tempo de cada distância d para o *Mean-Shift*;
- tK_{Md} é a média de tempo de cada distância d para o *K-Means* usando a quantidade de grupos do *Mean-Shift*;

Além disso, os agrupamentos associados a cada ponto foram comparados utilizando o Chapter 4. Os agrupamentos do DBSCAN (Δ) foram comparados com os agrupamentos do *K-Means* usando a quantidade de agrupamentos do DBSCAN (K_Δ). Os agrupamentos do *Mean-Shift* (M) foram comparados com os agrupamentos do *K-means* usando a quantidade de agrupamentos do *Mean-Shift* (K_M). Para cada uma dessas duas comparações, a porcentagem de igualdade entres os grupos foi calculada. Para cada distância d, foram feitas as médias de igualdade considerando todas as bases ($I\Delta_d$ e IM_d).

As distâncias foram *Canberra*, *Chebyshev*, Euclidiana, *Minkowski* e Rodrigues. Para o cálculo das distâncias *Minkowski* e Rodrigues é necessário um parâmetro adicional p_M e p_R respectivamente. Para isso serão utilizados os valores $p_M = 0.75, 3, 4$ e $p_R = 0.75, 1, 2, 3$.

Todos os testes foram feitos utilizando um computador de uso pessoal com processador Intel Core I7-8565U com CPU 1.99 GHz e memória RAM de 8 Gb.

Algoritmo 3 – *Método para o cálculo dos tempos e agrupamentos*

requer Bases b, distâncias d Porcentagem de igualdades $I\Delta_d$, IM_d
para todos d **faça**
  $t\Delta_d$, $tK_{\Delta d}$, tM_d , tK_{Md} = 0
  $I\Delta_d$, IM_d = 0
  para todos b **faça**
    $t\Delta_{b,d}$, $k\Delta_{b,d}$, $c\Delta_{b,d}$, agrupamentos$\Delta = DBScan(b,d)$
    $tM_{b,d}$, $kM_{b,d}$, $cM_{b,d}$, agrupamentos$M = MeanShift(b,d)$
    $tK_{\Delta,b,d}$, $k\Delta_{b,d}$, agrupamentos$K\Delta = Kmeans(b,d,k\Delta_{b,d},c\Delta_{b,d})$
    $tK_{M,b,d}$, $kM_{b,d}$, agrupamentos$KM = Kmeans(b,d,kM_{b,d},cM_{b,d})$
    $t\Delta_d$ += $t\Delta_{b,d}/11$
    $tK_{\Delta d}$ += $tK_{\Delta,b,d}/11$
    tM_d += $tM_{b,d}/11$
    tK_{Md} += $tK_{M,b,d}/11$
    $I\Delta_d$ += comparar(agrupamentosΔ, agrupamentos$K\Delta$)$/11$
    IM_d += comparar(agrupamentos$M$, agrupamentosKM)$/11$
  finaliza para
finaliza para
retorna Médias de tempo $t\Delta_d$, $tK_{\Delta d}$, tM_d e tK_{Md} e Porcentagem de igualdades $I\Delta_d$, IM_d

Fonte: Autoria própria (2023).

Algoritmo 4 – *Comparação entre agrupamentos*

requer agrupamento A e agrupamento B
$acerto = 0$
Tamanho do agrupamento N_A
Tamanho do agrupamento N_B
para todos Ponto P em A **faça**
  se P também está no agrupamento B **então**
    $acerto = acerto + 1$
  finaliza se
finaliza para
$I = (acerto * 100)/N$
retorna Percentual de igualdade I

Fonte: Autoria própria (2023).

4.1 Bases utilizadas

Existem diversas bases de dados utilizadas como dados de entrada para os algoritmos, e estão descritas na Tabela 7.

Todas as bases de dados foram retiradas do repositório da UCI (DUA; GRAFF, 2017) e estão brevemente comentadas logo abaixo.

- *Ecoli*: contém a localização de proteínas em bactérias Gram-negativas, considerando apenas a informação da sequência de aminoácidos;
- *Glass*: estudo de classificação de tipos de vidros motivados por uma investigação criminal do Serviço de Ciência Forense dos Estados Unidos, estão definidos em termos de seus teores de óxido;

Tabela 7 – Comparação entre as diferentes bases de dados

Base de Dados	Características	Atributos	Número Atributos	Número Amostras
Ecoli	Multivariável	Real	8	336
Glass	Multivariável	Real	10	214
Haberman	Multivariável	Inteiro, Real	3	306
Heart-Statlog	Multivariável	Categórico, Real	13	270
Ionosphere	Multivariável	Inteiro, Real	34	351
Iris	Multivariável	Real	4	150
Sonar	Multivariável	Real	60	208
Spect	Multivariável	Categórico	22	267
Spectf	Multivariável	Categórico	44	349
Tae	Multivariável	Categórico, Inteiro	5	151
Wine	Multivariável	Inteiro, Real	13	178

Fonte: Autoria própria (2023).

- *Haberman*: representa os dados de um estudo realizado pelo hospital *Billings* da Universidade de Chicago entre os anos de 1958 e 1970 sobre pacientes que sobreviveram a cirurgias de câncer de mama;
- *Heart-Statlog*: base de dados de doenças cardíacas;
- *Ionosphere*: dados de retornos de radar da ionosfera. Coletados por um sistema em *Goose Bay*, Labrador;
- *Iris*: esta base contém 3 classes com 50 amostras cada representando um tipo de planta iris;
- *Sonar*: sinais entre o sonar refletido em um cilindro metálico e refletido em uma rocha aproximadamente cilíndrica;
- *Spect* e *Spectf*: dados de imagens cardíacas de tomografia computadorizada por emissão de próton único. Cada paciente está classificado em duas categorias: normal e anormal;
- *Tae*: avaliações de desempenho de ensino ao longo de três semestres regulares e dois semestres de verão de 151 atribuições de assistente de ensino no Departamento de Estatística da Universidade de Wisconsin-Madison;
- *Wine*: representa os resultados de uma análise química de vinhos que cresceram na mesma região mas derivados de três cultivares diferentes. As análises determinaram treze atributos encontrados em cada um dos três tipos de vinhos.

A partir dessas informações, as variáveis categóricas foram codificadas de duas formas, para as que possuem ordem e para as que não são ordenadas. Essas codificações foram feitas através da biblioteca *scikit-learn* do *Python*.

5 RESULTADOS

Neste capítulo foram avaliadas as médias dos tempos que cada algoritmo de agrupamento (DBSCAN, *Mean-Shift* e *K-Means*) levou para agrupar 11 bases de dados do repositório da UCI. Para performar os algoritmos foram utilizadas 5 distâncias diferentes, sendo elas, *Canberra*, *Chebyshev*, Euclidiana, *Minkowski* e Rodrigues. Ainda, valores diferentes de p foram passados como parâmetros para as distâncias de *Minkowski* e Rodrigues, totalizando 10 distâncias diferentes.

Além das médias de tempo, foram calculadas as porcentagens de igualdades dos agrupamentos obtidos no DBSCAN em relação ao *K-means* e dos agrupamentos obtidos no *Mean-Shift* em relação ao *K-means* para cada base de dados. E para cada distância, foram calculadas as médias das porcentagens.

Para chegar nesses resultados diversos testes foram feitos para encontrar uma combinação dos parâmetros ϵ e $MinPts$ para o DBSCAN, e h para o *Mean-Shift*. Para o início dos testes, os valores de ϵ, $MinPts$ foram determinados aleatoriamente, se a quantidade de agrupamentos que foram gerados fosse 1, era necessário diminuir os valores para gerar pelo menos dois agrupamentos. E se caso os valores gerassem uma quantidade um pouco acima ou igual a dois eles eram mantidos. Os valores dos parâmetros h foram iguais aos valores dos parâmetros ϵ e podem ser conferidos na Tabela 8 e Tabela 9.

Tabela 8 – Parâmetros utilizados para cada base e distância (parte 1)

	Haberman		***Ecoli***		***Glass***		***Heart-Statlog***		***Ionosphere***	
Parâmetros	ϵ	$MinPts$	ϵ	$MinPts$	ϵ	$MinPts$	ϵ	$MinPts$	ϵ	$MinPts$
Canberra	0.3	5	0.4	29	1.1	5	1.5	5	9	6
Chebyshev	2	12	0.1	5	0.7	5	15	5	0.4	8
Euclidiana	5	5	0.16	10	0.9	6	20	5	1	5
Minkowski (p=0.75)	5	16	0.5	20	2	5	45	5	19	5
Minkowski (p=3)	3	19	0.08	5	0.8	5	17	5	0.9	14
Minkowski (p=4)	4	3	0.07	5	0.6	5	16	5	0.7	13
Rodrigues (p=0.75)	10	3	0.4	3	6	5	45	5	12	5
Rodrigues (p=1)	10	3	0.35	3	4	5	10	5	5	5
Rodrigues (p=2)	8	3	0.25	3	1	5	30	5	1	5
Rodrigues (p=3)	6	3	0.2	3	1	5	30	5	0.9	5

Algumas bases possuem diferenças maiores nos valores das amostras e quantidades de atributos, sendo necessário a utilização de valores diferentes para ϵ e $MinPts$. Observa-se também a necessidade nas mudanças dos parâmetros para cada distância. Caso os mesmos valores fossem utilizados, os algoritmos poderiam gerar apenas um agrupamento ou cada amostra na base seria um agrupamento diferente.

Com base nesses parâmetros, os resultados mostram as diferenças nos tempos de execuções e igualdades nos agrupamentos gerados por cada algoritmo utilizando as diferentes distâncias. As médias de igualdades dos agrupamentos em relação ao DBSCAN e *K-Means* representada pela quarta coluna da Tabela 10 levaram em consideração os pontos de ruídos

Tabela 9 – Parâmetros utilizados para cada base e distância (parte 2)

	Sonar		***Spect***		***Spectf***		***Tae***		***Wine***	
Parâmetros	ϵ	$MinPts$	ϵ	$MinPts$	ϵ	$MinPts$	ϵ	$MinPts$	ϵ	$MinPts$
Canberra	15	5	3.5	3	1.17	3	0.43	3	0.99	10
Chebyshev	0.4	5	0.9	3	20	2	5	3	18	5
Euclidiana	1	5	1.5	5	55	2	7	3	25	5
Minkowski (p=0.75)	20	5	5	7	550	3	13	3	80	5
Minkowski (p=3)	0.6	5	1.5	5	20	3	7	3	25	5
Minkowski (p=4)	0.5	5	1.3	5	15	3	6	3	22	5
Rodrigues (p=0.75)	15	5	5	5	500	3	15	7	150	5
Rodrigues (p=1)	6	5	3	5	170	3	15	7	60	5
Rodrigues (p=2)	1.65	5	2.7	5	45	3	12	7	45	5
Rodrigues (p=3)	0.8	5	2.4	5	23	3	10	7	45	5

gerados pelo DBSCAN. Esses pontos foram considerados como desigualdades em relação ao *K-Means*.

Para as igualdades, dado um ponto A na base b relacionado ao agrupamento k no DBSCAN, esse mesmo ponto deve estar relacionado ao mesmo agrupamento k no *K-Means*. Essa relação foi a mesma para os agrupamentos do *Mean-Shift*.

Tabela 10 – Resultado dos tempos e igualdades de agrupamentos

	Média Tempo DBSCAN (s)	**Média Tempo K-means/ DBSCAN (s)**	**Média Igualdade de Agrupamento (%)**	**Média Tempo MEANSHIFT (s)**	**Média Tempo K-means/ MEANSHIFT (s)**	**Média Igualdade de Agrupamento (%)**
Canberra	0.9043	0.5407	56.1895	60.5897	2.01561	72.4848
Chebyshev	**0.2271**	0.1632	44.2416	**26.7258**	**0.0640**	66.0789
Euclidiana	0.4192	0.1417	**56.4433**	34.2925	0.2475	68.6822
Minkowski (p = 0.75)	0.3420	**0.0538**	55.5625	58.4959	0.1765	67.3168
Minkowski (p = 3)	0.3700	0.1187	42.4154	35.9101	0.4783	65.3603
Minkowski (p = 4)	0.5073	0.1343	40.2378	33.4449	0.6340	**73.6926**
Rodrigues (p = 0.75)	0.6716	0.3820	41.5144	65.1638	0.5234	64.4529
Rodrigues (p = 1)	0.6149	0.3595	48.0192	57.3255	0.6278	65.4701
Rodrigues (p = 2)	0.6719	0.2431	50.1235	54.4946	0.5512	66.6183
Rodrigues (p = 3)	0.7528	0.1779	43.3054	52.7413	2.2310	72.7934

Fonte: Autoria própria (2023).

As médias de tempo que foram obtidas pelos algoritmos DBSCAN e *Mean-Shift*, e a combinação dos parâmetros do *Mean-Shift* com o *K-means* utilizando a distância *Chebyshev* foi a mais baixa em relação as outras. Na média de tempo da combinação do *K-Means* com o DBSCAN, a distancia *Minkowski* tendo como parâmetro $p = 0.75$ obteve o menor valor.

Para o DBSCAN utilizando a *Canberra*, a média retornada foi a mais alta entre as dez. E utilizando a distância de Rodrigues com o parâmetro $p = 0.75$ teve a segunda pior média. No geral, na utilização da distância de Rodrigues, os resultados foram a segunda pior média, as piore médias foram a de *Canberra*.

A quarta e a sétima coluna da tabela representam as médias de igualdades dos agrupamentos obtidos pelo DBSCAN em comparação com o *K-Means* e pelo *Mean-Shift* em comparação com o *K-Means* respectivamente. Com isso, a utilização da distância Euclidiana foi obtida

a maior porcentagem na quarta coluna e da distância de *Minkowski* com $p = 4$ retornou uma porcentagem maior para a sétima coluna.

Para efeito de comparação, os ruídos encontrados pelo algoritmo DBSCAN podem ser descartados. Assim, a comparação entre os agrupamentos não levará em consideração esses ruídos. E os resultados desse efeito podem ser observado na Tabela 11.

Tabela 11 – Igualdades dos agrupamentos do DBSCAN em relação ao *K-Means* desconsiderando os ruídos

Distâncias	**Média de Igualdade dos Agrupamentos (%)**
Canberra	66.8644
Chebyshev	67.3692
Euclidiana	73.3364
Minkowski (p=0.75)	**80.4863**
Minkowski (p=3)	68.3927
Minkowski (p=4)	56.4827
Rodrigues (p=0.75)	67.5934
Rodrigues (p=1)	68.2313
Rodrigues (p=2)	72.9220
Rodrigues (p=3)	67.5773

Fonte: Autoria própria (2023).

Vale notar que ao fazer isso, a combinação dos parâmetros tem um efeito diferente. Sendo assim, a maior média de igualdade é retornada utilizando a distância de *Minkowski* com $p = 0.75$. Nesse caso, pode ser que a combinação dos parâmetros, levou o algoritmo DBSCAN a gerar uma quantidade menor de ruídos.

Como exemplo para o DBSCAN, é possível observar os agrupamentos formados da base de dados *Iris* na Figura 21 e na Figura 22. A distância utilizada nesse exemplo é a distância Euclidiana, tendo como valores $\epsilon = 0.4$ e $MinPts = 5$. Com essas informações, foram gerados 4 agrupamentos, demonstrados por um X na cor preta, e alguns pontos não foram agrupados, sendo eles, os ruídos representados na cor vermelha

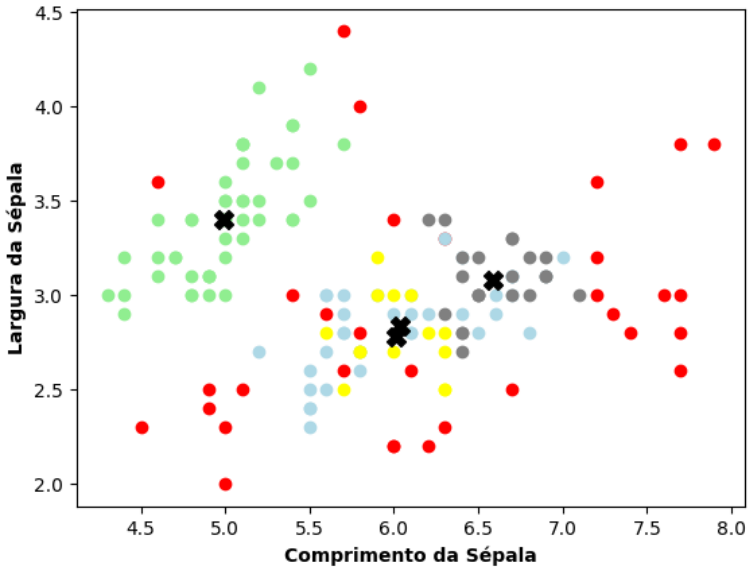
É possível observar uma diferença maior na Figura 22 em relação a posição dos centroides, a qual separa o comprimento e a largura da pétala. Os pontos de ruídos são visivelmente mais separados na Figura 21.

Para uma visualização mais geral, a Figura 23 e Figura 24 mostram em 3 dimensões como os agrupamentos estão distribuídos.

As posições dos agrupamentos e quantidades de agrupamentos que foram encontrados no algoritmo DBSCAN, foram utilizadas como entrada para o *K-means*, incluindo a distância euclidiana. Com as entradas iniciais foram obtidos os agrupamentos conforme Figura 25 e Figura 26.

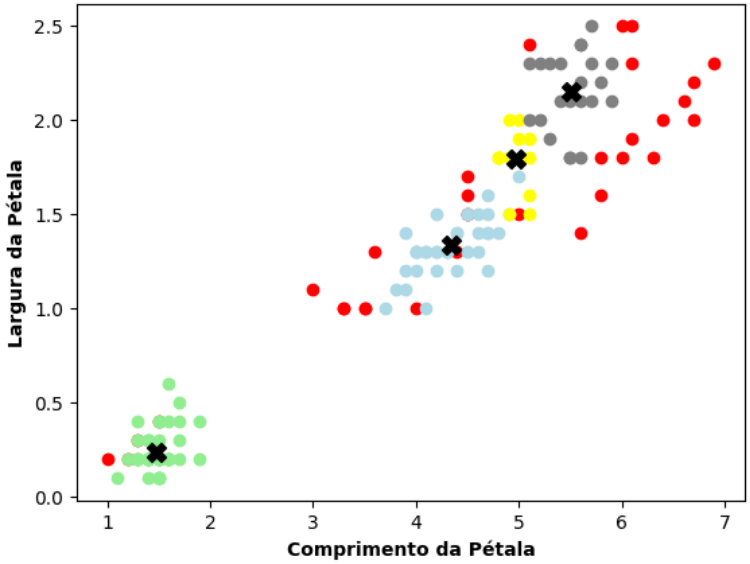
Houve variações nas posições dos centroides e todos os ruídos apresentados pelo DBSCAN agora fazem parte de algum grupo no *K-means*. No entanto, alguns pontos que pertenciam a um grupo agora pertencem a outro.

Figura 21 – Separação dos agrupamentos do DBSCAN da largura e do comprimento da sépala (distância Euclidiana)



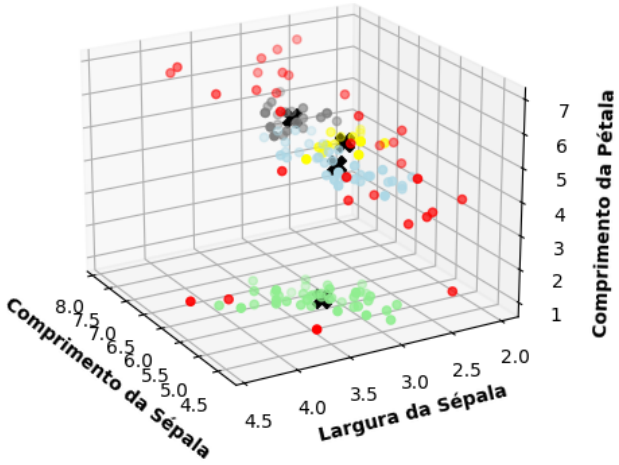
Fonte: Autoria própria (2023).

Figura 22 – Separação dos agrupamentos do DBSCAN da largura e do comprimento da pétala (distância Euclidiana)



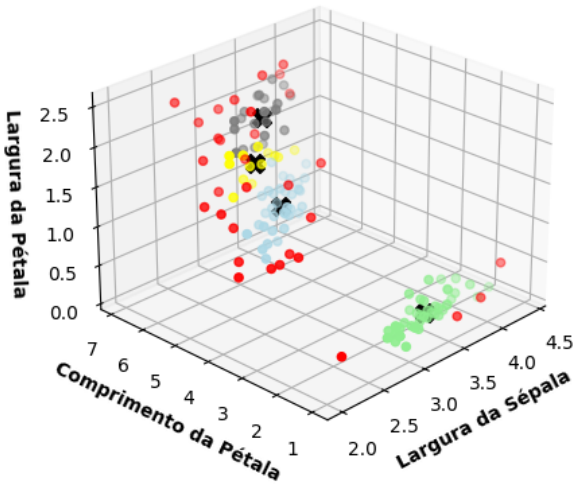
Fonte: Autoria própria (2023).

Figura 23 – Separação dos agrupamentos do DBSCAN da largura e do comprimento da sépala em relação ao comprimento da pétala (distância Euclidiana)



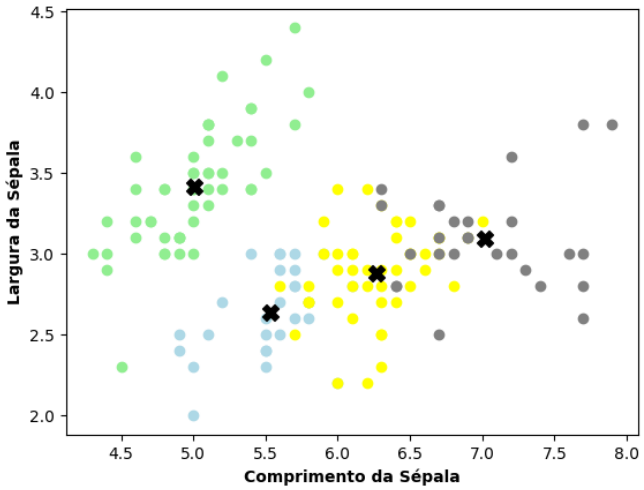
Fonte: Autoria própria (2023).

Figura 24 – Separação dos agrupamentos do DBSCAN da largura e do comprimento da Pétala em relação a largura da sépala (distância Euclidiana)



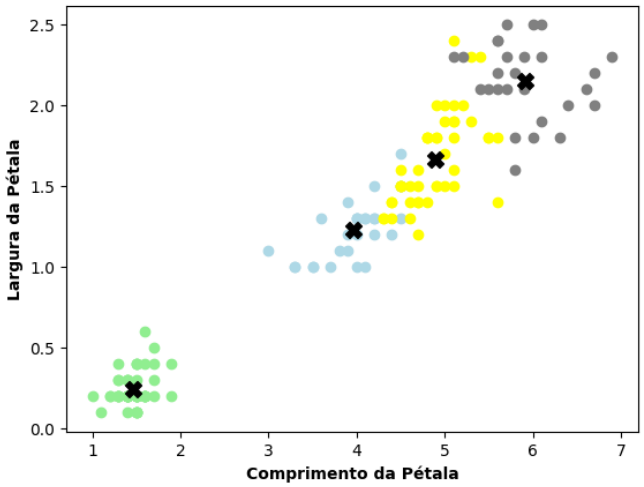
Fonte: Autoria própria (2023).

Figura 25 – Separação dos agrupamentos do *K-means* da largura e do comprimento da sépala (distância Euclidiana)



Fonte: Autoria própria (2023).

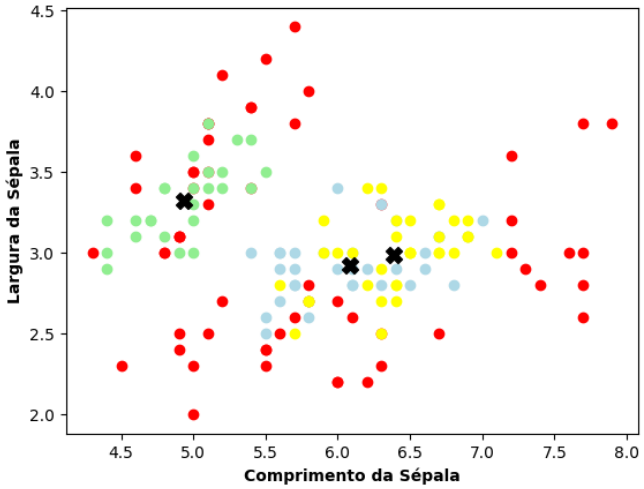
Figura 26 – Separação dos agrupamentos do *K-means* da largura e do comprimento da pétala (distância Euclidiana)



Fonte: Autoria própria (2023).

A distância que foi utilizada afetou significativamente a geração dos agrupamentos, além dos outros parâmetros. Em comparação com a distância de *Canberra*, foram necessários os parâmetros $\epsilon = 0.1$ e $MinPts = 8$ para gerar três agrupamentos.

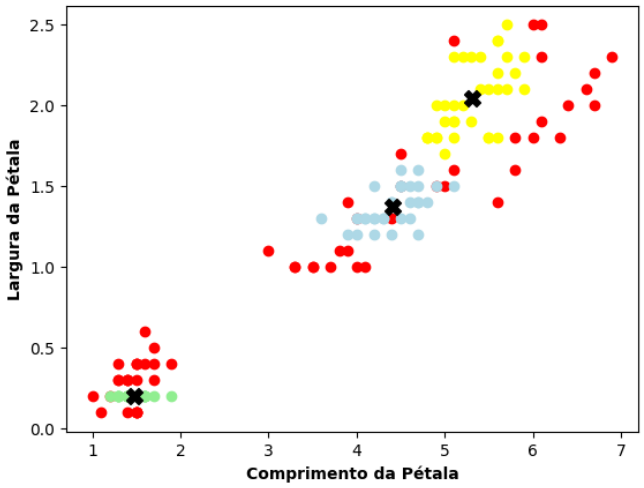
Figura 27 – Separação dos agrupamentos do DBSCAN da largura e do comprimento da sépala (distância *Canberra*)



Fonte: Autoria própria (2023).

Por outro gráfico (Figura 28) é possível observar uma separação mais significativa dos agrupamentos, levando em consideração os dados das pétalas.

Figura 28 – Separação dos agrupamentos do DBSCAN da largura e do comprimento da pétala (distância *Canberra*)

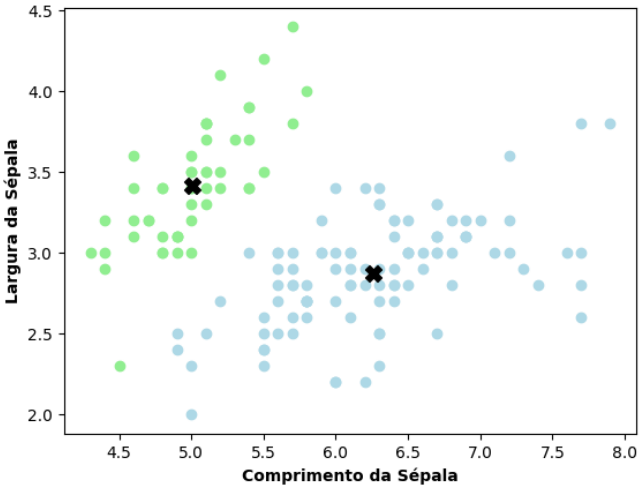


Fonte: Autoria própria (2023).

Comparando a distância de *Canberra* com os mesmos parâmetros utilizados na distância Euclidiana, os agrupamentos gerados foram diferentes, e são mostrados na Figura 29.

Ademais, todos os pontos da base de dados foram atribuídos a um grupo, não sendo gerados os ruídos.

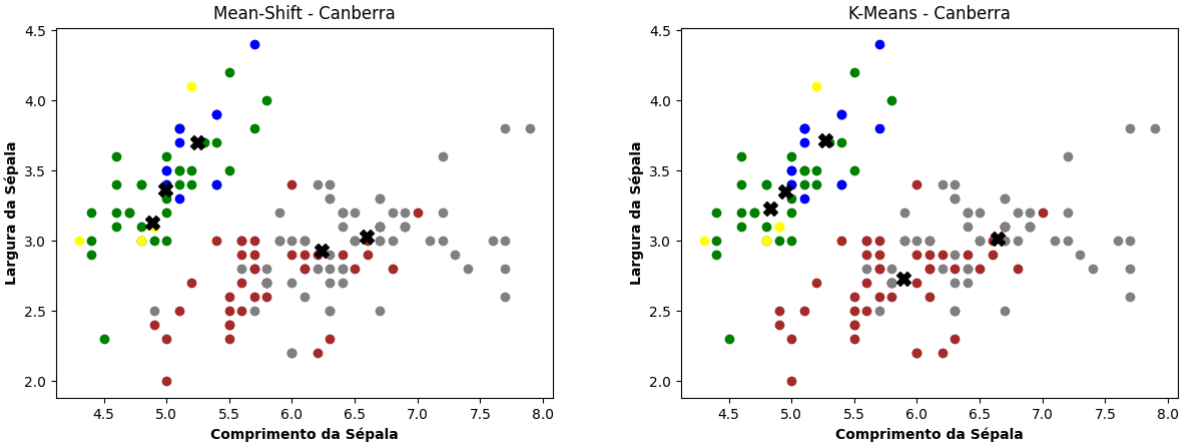
Figura 29 – Separação dos agrupamentos do DBSCAN com a distância *Canberra* ($\epsilon = 0.4$ e $MinPts = 5$)



Fonte: Autoria própria (2023).

Para o algoritmo *Mean-Shift*, são utilizadas as distâncias *Canberra* e Rodrigues com $p = 0.75$ como exemplo. Na primeira, foi gerado cinco agrupamentos com o parâmetro $h = 0.4$ e na segunda, quatro agrupamentos com os parâmetros $h = 1$. A Figura 30 mostra a diferença nas posições dos centroides em relação aos algoritmos e a mudança de alguns pontos para outros grupos.

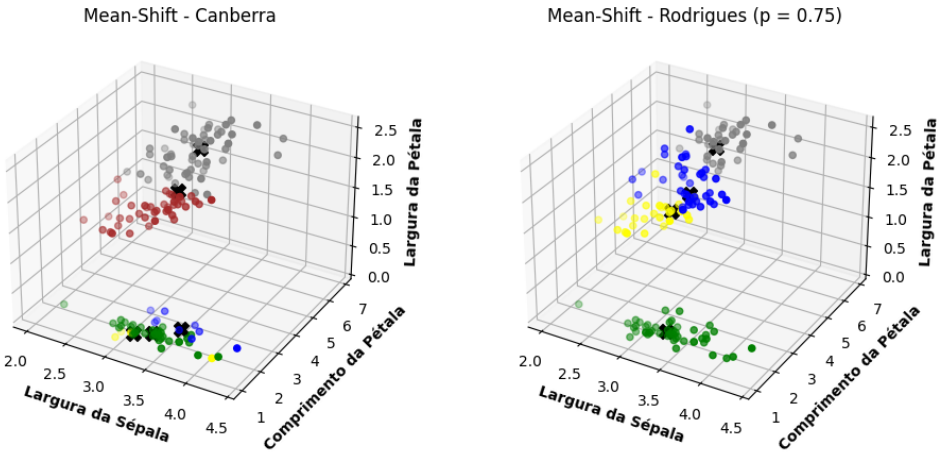
Figura 30 – Comparativo entre os algoritmos *Mean-Shift* e *K-means* com a distância *Canberra*



Fonte: Autoria própria (2023).

Observa-se uma diferença entre as duas distâncias em um gráfico com três dimensões na Figura 31, onde cada grupo está posicionado em lugares com mais densidades de pontos.

Figura 31 – Comparativo entre as distâncias de *Canberra* e Rodrigues (p=0.75) para o *Mean-Shift*



Fonte: Autoria própria (2023).

Há alguns exemplos os quais, as mudanças nas posições dos agrupamentos são bem maiores em relação as outras. Isso se deve ao fato de que as distâncias em conjunto com os parâmetros dos algoritmos influenciam diretamente no comportamento dos grupos. Ainda, a posição desses grupos já definidos nos algoritmos DBSCAN e *Mean-Shift* como entrada para o algoritmo *K-means* pode gerar uma convergência das posições finais com menos iterações no cálculo das novas posições dos agrupamentos.

6 CONCLUSÃO

Esse trabalho teve como objetivo analisar três algoritmos, DBSCAN, *K-Means* e *Mean-Shift*, os quais precisam de uma medida de distância para calcular as similaridades e dissimilaridades entre pares de dados para agrupa-los. As medidas de distâncias utilizadas foram *Canberra*, *Chebyshev*, Euclidiana, *Minkowski* e Rodrigues, sendo as duas últimas dependentes de um parâmetro p. Esse parâmetro foi variado conforme segue: para *Minkowski*, $p = 0.75, 3, 4$ e para Rodrigues, $p = 0.5, 1, 2, 3$. Para realizar os agrupamentos foram utilizadas 11 bases de dados retiradas do Repositório da UCI.

Um total de 10 distâncias foram utilizadas e as análises consistiam em determinar a média de tempo dos algoritmos DBSCAN, *K-Means* e *Mean-Shift* utilizando as 11 bases de dados para cada distância. Ainda, o *K-Means* foi executado duas vezes para cada base, uma vez com a quantidade de agrupamentos e posições dos centros dos agrupamentos do DBSCAN e a segunda vez com a quantidade de agrupamentos e posições dos centros dos agrupamentos do *Mean-Shift*.

Os agrupamentos retornados pelo DBSCAN e pela primeira execução do *K-Means* foram comparados para cada base e foi calculada a média dessa comparação. O mesmo procedimento foi realizado para o *Mean-Shift* e a segunda execução do *K-Means*.

As médias de tempos tiveram uma variação bem grande em relação ao algoritmo *Mean-Shift*, pois para cada ponto visitado, é necessário verificar os pontos que estão dentro da janela (largura de banda) e calcular um peso em relação a todos os pontos dentro da largura de banda. Esses pesos irão determinar o deslocamento que essa janela terá. Esses cálculos são feitos até que não tenha mais deslocamento da janela.

No *K-Means*, para cada ponto foi necessário calcular a distância até os centros dos agrupamentos, sendo que esse cálculo foi feito de acordo com o a quantidade de agrupamentos. O tempo então é medido com base nisso e também na quantidade de iterações que será feito para as posições dos centros dos agrupamentos não mudarem mais.

Já no DBSCAN, o início ocorre escolhendo um ponto aleatório e de acordo com epsilon é definida a sua vizinhança identificando se é um ponto central. Todos os pontos serão visitados e serão definidos como pontos centrais, pontos de fronteira ou ruídos.

Foram testados diversos valores para a largura de banda do algoritmo *Mean-Shift*, para o epsilon e MinPts do algoritmo DBSCAN. Pois para cada distância utilizada, os valores encontrados podem mudar. A utilização de larguras de banda, epsilon e minPts iguais para todas as bases e distâncias pode acarretar em muitos agrupamentos ou apenas um agrupamento.

As escolhas desses valores foram feitas de modo a gerar pelo menos dois agrupamentos para que fossem utilizados no algoritmo *K-means*. A comparação dos agrupamentos também levou em conta os ruídos como um ponto sem grupo, diminuindo a porcentagem de comparação.

As vantagens do DBSCAN e *Mean-Shift* em relação ao *K-Means* é que pode ser que em algumas bases alguns dados que foram coletados podem estar muito fora do esperado. Assim,

o *K-Means* irá associar todos os dados a pelo menos um grupo. O DBSCAN e *Mean-Shift* consideram esses dados como ruídos, dependendo dos parâmetros passados.

Os resultados mostraram que, no geral, ao utilizar a distância de *Chebyshev*, foram obtidos tempos menores que com as outras distâncias. E em seguida, ao utilizar a distância de *Minkowski* com parâmetro $p = 0.75$, as médias de tempo ficaram como a segunda melhor média no geral. Já para a distância Euclidiana, foi possível obter a melhor média de igualdade entre os agrupamentos do *K-Means* tendo como entrada as saídas de quantidade e posições dos agrupamentos do DBSCAN. E para a média de igualdade de agrupamento do *K-Means* em relação ao *Mean-Shift*, o resultado foi o melhor entre as outras distâncias.

As distâncias utilizadas também influenciam no agrupamento e nos ruídos, pois se os parâmetros forem os mesmos pode ser que o ruído encontrado com uma distância seja agrupado em outra.

Os ruídos influenciaram diretamente na comparação da igualdade dos agrupamentos do DBSCAN em relação ao *K-Means*. A porcentagem de igualdade aumenta quando os ruídos não são considerados no cálculo das igualdades.

Um dos problemas encontrados, foi determinar valores para os parâmetros dos algoritmos DBSCAN e *Mean-Shift*, de forma a gerar dois ou mais agrupamentos. Os valores iniciais foram escolhidos aleatoriamente, e para algumas distâncias, a diferença dos valores é muito diferente do que em outras.

Outro problema encontrado foi o tempo que o algoritmo *Mean-Shift* levou para a execução com algumas bases relativamente maiores que outras em relação ao número de atributos e amostras. O tempo de espera combinado com a mudança nos parâmetros de largura de banda se tornam demorados.

Para contornar esses problemas, como trabalho futuro, pode ser feita uma análise das amostras das bases levando em consideração as diferenças de valores, utilizando alguma ferramenta para determinar os valores de ϵ e $MinPts$ para o DBSCAN e largura de banda (h) para o *Mean-Shift*. Com isso, reduzirá o tempo que leva para encontrar valores adequados para esses parâmetros.

Sendo assim, foi possível fazer uma comparação entre os algoritmos de aprendizado de máquina não supervisionados, os quais são utilizados para agrupamento de dados mas com maneiras diferentes de agrupar. Analisando o tempo que cada um consome em relação ao tamanho das bases e se os agrupamentos são parecidos.

REFERÊNCIAS

ABELLO, J.; PARDALOS, P. M.; RESENDE, M. G. **Handbook of massive data sets**. [*S.l.*]: Springer, 2013. v. 4.

AGGARWAL, K. *et al.* Has the future started? the current growth of artificial intelligence, machine learning, and deep learning. **Iraqi Journal for Computer Science and Mathematics**, v. 3, n. 1, p. 115–123, 2022.

AHMAD, A.; DEY, L. A k-mean clustering algorithm for mixed numeric and categorical data. **Data & Knowledge Engineering**, Elsevier, v. 63, n. 2, p. 503–527, 2007.

ALASADI, S. A.; BHAYA, W. S. Review of data preprocessing techniques in data mining. **Journal of Engineering and Applied Sciences**, v. 12, n. 16, p. 4102–4107, 2017.

ARLIA, D.; COPPOLA, M. Experiments in parallel clustering with dbscan. *In*: SPRINGER. **European Conference on Parallel Processing**. [*S.l.*], 2001. p. 326–331.

BEENTJE, H. J. *et al.* **The Kew plant glossary: an illustrated dictionary of plant terms.** [*S.l.*]: Royal Botanic Gardens, 2010.

BIAMONTE, J. *et al.* Quantum machine learning. **Nature**, Nature Publishing Group, v. 549, n. 7671, p. 195–202, 2017.

BISWAS, R. Introducing “nr-statistics”: A new direction in “statistics”. *In*: **Handbook of Research on Generalized and Hybrid Set Structures and Applications for Soft Computing**. [*S.l.*]: IGI Global, 2016. p. 490–535.

CHA, S.-H. Comprehensive survey on distance/similarity measures between probability density functions. **City**, v. 1, n. 2, p. 1, 2007.

CHENG, Y. Mean shift, mode seeking, and clustering. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 17, n. 8, p. 790–799, 1995.

CHOI, S.-S.; CHA, S.-H.; TAPPERT, C. C. A survey of binary similarity and distance measures. **Journal of systemics, cybernetics and informatics**, Citeseer, v. 8, n. 1, p. 43–48, 2010.

CUNNINGHAM, P.; CORD, M.; DELANY, S. J. Supervised learning. *In*: **Machine learning techniques for multimedia**. [*S.l.*]: Springer, 2008. p. 21–49.

DEZA, E. *et al.* **Encyclopedia of distances**. [*S.l.*]: Springer, 2009.

DUA, D.; GRAFF, C. **UCI Machine Learning Repository**. 2017. Disponível em: http://archive.ics.uci.edu/ml.

FUKUNAGA, K.; HOSTETLER, L. The estimation of the gradient of a density function, with applications in pattern recognition. **IEEE Transactions on information theory**, IEEE, v. 21, n. 1, p. 32–40, 1975.

KHAN, K. *et al.* Dbscan: Past, present and future. *In*: IEEE. **The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)**. [*S.l.*], 2014. p. 232–238.

LANCE, G. N.; WILLIAMS, W. T. Computer programs for hierarchical polythetic classification (“similarity analyses”). **The Computer Journal**, The British Computer Society, v. 9, n. 1, p. 60–64, 1966.

LIBERTI, L. Distance geometry and data science. **Top**, Springer, v. 28, n. 2, p. 271–339, 2020.

LIU, B. Supervised learning. *In*: **Web data mining**. [*S.l.*]: Springer, 2011. p. 63–132.

LU, B. *et al.* The minkowski approach for choosing the distance metric in geographically weighted regression. **International Journal of Geographical Information Science**, Taylor & Francis, v. 30, n. 2, p. 351–368, 2016.

MAHESH, B. Machine learning algorithms-a review. **International Journal of Science and Research (IJSR).[Internet]**, v. 9, p. 381–386, 2020.

MALLEY, B.; RAMAZZOTTI, D.; WU, J. T.-y. Data pre-processing. **Secondary analysis of electronic health records**, Springer, p. 115–141, 2016.

MEHTA, P. *et al.* A high-bias, low-variance introduction to machine learning for physicists. **Physics reports**, Elsevier, v. 810, p. 1–124, 2019.

NAQA, I. E.; MURPHY, M. J. What is machine learning? *In*: **machine learning in radiation oncology**. [*S.l.*]: Springer, 2015. p. 3–11.

PANDIT, S.; GUPTA, S. *et al.* A comparative study on distance measuring approaches for clustering. **International journal of research in computer science**, Citeseer, v. 2, n. 1, p. 29–31, 2011.

PERLIBAKAS, V. Distance measures for pca-based face recognition. **Pattern recognition letters**, Elsevier, v. 25, n. 6, p. 711–724, 2004.

RODRIGUES, É. O. Combining minkowski and cheyshev: New distance proposal and survey of distance metrics using k-nearest neighbours classifier. **Pattern Recognition Letters**, Elsevier, v. 110, p. 66–71, 2018.

ROKACH, L.; MAIMON, O. Clustering methods. *In*: **Data mining and knowledge discovery handbook**. [*S.l.*]: Springer, 2005. p. 321–352.

RUMMEL, R. J. Understanding conflict and war: vol. 2: the conflict helix. **Bev-erly Hills: Sage**, 1976.

RUSSELL, S.; NORVIG, P. Artificial intelligence: a modern approach. 2002.

SCHUBERT, E. *et al.* Dbscan revisited, revisited: why and how you should (still) use dbscan. **ACM Transactions on Database Systems (TODS)**, ACM New York, NY, USA, v. 42, n. 3, p. 1–21, 2017.

SCULLEY, D. Web-scale k-means clustering. *In*: **Proceedings of the 19th international conference on World wide web**. [*S.l.*: *s.n.*], 2010. p. 1177–1178.

SEIF, G. The 5 clustering algorithms data scientists need to know. **Towards Data Science**, 2018.

SHEN, C. *et al.* Efficient dual approach to distance metric learning. **IEEE transactions on neural networks and learning systems**, IEEE, v. 25, n. 2, p. 394–406, 2013.

SINAGA, K. P.; YANG, M.-S. Unsupervised k-means clustering algorithm. **IEEE access**, IEEE, v. 8, p. 80716–80727, 2020.

SZELISKI, R. **Computer vision: algorithms and applications**. [*S.l.*]: Springer Science & Business Media, 2010.

SZELISKI, R. **Computer vision: algorithms and applications**. [*S.l.*]: Springer Nature, 2022.

TABAK, J. **Geometry: the language of space and form**. [*S.l.*]: Infobase Publishing, 2014.

TROPE, Y.; LIBERMAN, N. Construal-level theory of psychological distance. **Psychological review**, American Psychological Association, v. 117, n. 2, p. 440, 2010.

ULLMAN, J. D. **A first course in database systems**. [*S.l.*]: Pearson Education India, 2007.

WALLNÖFER, J. *et al.* Machine learning for long-distance quantum communication. **PRX Quantum**, APS, v. 1, n. 1, p. 010301, 2020.

WEI, J. *et al.* Machine learning in materials science. **InfoMat**, Wiley Online Library, v. 1, n. 3, p. 338–358, 2019.

XING, E. *et al.* Distance metric learning with application to clustering with side-information. **Advances in neural information processing systems**, v. 15, 2002.

YANG, L.; JIN, R. Distance metric learning: A comprehensive survey. **Michigan State Universiy**, v. 2, n. 2, p. 4, 2006.