INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS – *CAMPUS* BAMBUÍ
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

Mateus Araújo Cruz

ANÁLISE DE TÉCNICAS DE AGRUPAMENTO DE DADOS PARA NOTÍCIAS DE FUTEBOL

Bambuí - MG
2023

MATEUS ARAÚJO CRUZ

ANÁLISE DE TÉCNICAS DE AGRUPAMENTO DE DADOS PARA NOTÍCIAS DE FUTEBOL

Bambuí - MG
2023

Catalogação na Fonte Biblioteca IFMG - Campus Bambuí

C957a Cruz, Mateus Araújo.
Análise de técnicas de agrupamento de dados para notícias de futebol. / Mateus Araújo Cruz. – 2023.
46 f. : il. ; color.

Orientador: Marcos Roberto Ribeiro.
Trabalho de Conclusão de Curso (graduação) - Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus Bambuí, MG, Curso Bacharelado em Engenharia de Computação, 2023.

1. Agrupamento de dados. 2. Dados textuais. 3. Notícias de futebol. I. Ribeiro, Marcos Roberto. II. Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus Bambuí, MG. III. Título.

CDD 001.61

Elaborada por Douglas Bernardes de Castro- CRB-6/2802

Mateus Araújo Cruz

ANÁLISE DE TÉCNICAS DE AGRUPAMENTO DE DADOS PARA NOTÍCIAS DE FUTEBOL

Aprovado em 30 de novembro de 2023 pela banca examinadora:

Prof. Dr. Marcos Roberto Ribeiro – IFMG – *Campus* Bambuí – (Orientador)
Prof. Me. Cláudio Ribeiro de Sousa – IFMG – *Campus* Bambuí
Prof. Me. Itagildo Edmar Garbazza – IFMG – *Campus* Bambuí

Documento assinado eletronicamente por Marcos Roberto Ribeiro, Professor, em 30/11/2023, às 15:38, conforme Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por Claudio Ribeiro de Sousa, Professor EBTT, em 30/11/2023, às 15:46, conforme Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por Itagildo Edmar Garbazza, Professor, em 30/11/2023, às 15:46, conforme Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade do documento pode ser conferida no site https://sei.ifmg.edu.br/consultadocs informando o código verificador 1750085 e o código CRC 4B3D6529.

Bambuí - MG
2023

AGRADECIMENTOS

Primeiramente, quero expressar minha gratidão ao meu Orientador, o Prof. Dr. Marcos Roberto Ribeiro. Desde o início, ele forneceu todo o suporte necessário para a realização do trabalho, compartilhando conhecimentos valiosos durante esse período. Quero estender meus agradecimentos a todos os professores do IFMG - *Campus* Bambuí. Sua dedicação contribuiu não apenas para o meu crescimento técnico, mas também para o meu desenvolvimento pessoal. Gostaria de agradecer às amizades feitas durante a faculdade, cujo apoio foi fundamental ao longo da jornada acadêmica. Além disso, expresso minha gratidão aos meus demais amigos, que mesmo distantes, sempre transmitiram seu apoio e incentivo. Por fim, dedico um agradecimento sincero à minha família. Aos meus pais, Fernando e Arlete, e à minha irmã Mariana, que não mediram esforços para lutar ao meu lado, tornando possível a realização dos meus sonhos. Estendo os agradecimentos também aos demais familiares que sempre me incentivaram e apoiaram em tudo que foi possível.

RESUMO

Agrupamento de dados é uma técnica de aprendizado não supervisionado que busca padrões ocultos em um conjunto de dados. Para isso, o conjunto é dividido em subgrupos com características semelhantes entre si e distintas dos demais grupos. O presente trabalho investiga as técnicas de agrupamento K-Means, Hierárquico, DBSCAN e mistura gaussiana, aplicadas em notícias do Campeonato Brasileiro de Futebol. A pesquisa tem como objetivo analisar o funcionamento das técnicas e proporcionar possibilidades de identificação de padrões nos dados. No estágio inicial, realizou-se o pré-processamento dos dados, incluindo *tokenização* e remoção de palavras de parada. As notícias foram representadas através da técnica TF-IDF. Em seguida, empregou-se a técnica de redução de dimensionalidade a partir da Análise Semântica Latente. O agrupamento das notícias foi realizado com o número de grupos definido em 21, representando a quantidade de times participantes no campeonato. Os resultados indicaram que tanto o algoritmo K-Means quanto o Modelo de Mistura Gaussiana alcançaram uma acurácia de 75%, demonstrando desempenho superior perante os demais. Adicionalmente, foram realizados experimentos sem a definição prévia do número de clusters, empregando busca em grade para determinar o melhor coeficiente de silhueta. Os algoritmos variaram entre 25 e 32 grupos, sugerindo que essa faixa é apropriada para a divisão da base de dados das notícias.

Palavras-chave: Agrupamento de dados. Dados textuais. Notícias de futebol.

ABSTRACT

Data clustering is an unsupervised learning technique that searches for hidden patterns in a set of data. To do this, the set is divided into subgroups with characteristics similar to each other and different from the other groups. The present work investigates the K-Means, hierarchical, DBSCAN and Gaussian mixture clustering techniques, applied to news from the Brazilian Football Championship. The research aims to analyze the functioning of the techniques and provide possibilities for identifying patterns in the data. In the initial stage, data pre-processing was carried out, including tokenization and removal of stop words. The news was represented using the TF-IDF technique. Next, the dimensionality reduction technique was used using Latent Semantic Analysis. The grouping of news was carried out with the number of groups set at 21, representing the number of teams participating in the championship. The results indicated that both the K-Means algorithm and the Gaussian Mixture Model achieved an accuracy of 75%, demonstrating superior performance compared to the others. Additionally, experiments were carried out without prior definition of the number of clusters, using grid search to determine the best silhouette coefficient. The algorithms varied between 25 and 32 groups, suggesting that this range is appropriate for dividing the news database.

Keywords: Data clustering. Textual data. Soccer news.

LISTA DE FIGURAS

Figura 1 – Etapas do KDD 14
Figura 2 – Etapa de *tokenização* 15
Figura 3 – Etapa de remoção de palavras de parada 16
Figura 4 – Etapa de derivação 16
Figura 5 – Gráficos da execução do algoritmo K-Means com 2 e 3 grupos 21
Figura 6 – Dendrograma 22
Figura 7 – Agrupamento com DBSCAN 24
Figura 8 – Gráfico de distribuição gaussiana 25
Figura 9 – Distribuição da quantidade de notícias por time 30
Figura 10 – Distribuição da quantidade de notícias por time em 2022 30
Figura 11 – Etapas de análise das notícias 33
Figura 12 – Gráfico de dispersão da matriz LSA 37
Figura 13 – Conceito latente 1 38
Figura 14 – Gráfico de valores singulares 38
Figura 15 – Busca em grade para o K-Means 41
Figura 16 – Busca em grade para o Agrupamento Hierárquico Aglomerativo 42
Figura 17 – Busca em grade para o DBSCAN 43
Figura 18 – Busca em grade para o GMM 44

LISTA DE TABELAS

Tabela 1 – Representação da matriz TF-IDF 18
Tabela 2 – Representação da matriz LSA 19
Tabela 3 – Melhor número de componentes para cada algoritmo 39
Tabela 4 – Percentual de acurácia para cada experimento 40
Tabela 5 – Resultados da busca em grade 44

SUMÁRIO

1 INTRODUÇÃO 11
1.1 Objetivos 11
1.2 Justificativa 12
1.3 Resultados esperados 12
1.4 Organização do documento 12

2 FUNDAMENTAÇÃO TEÓRICA 13
2.1 Descoberta de Conhecimento em Banco de Dados 13
2.2 Pré-processamento de texto 14
2.2.1 Tokenização *14*
2.2.2 Remoção de palavras de parada *15*
2.2.3 Derivação *15*
2.2.4 Representação no espaço vetorial e normalização *16*
2.3 Redução de dimensionalidade 18
2.4 Cálculo de similaridade 19
2.5 Agrupamento de dados 20
2.5.1 Agrupamento particional *20*
2.5.2 Agrupamento hierárquico *21*
2.5.3 Agrupamento baseado em densidade *23*
2.5.4 Agrupamento baseado em modelo *23*
2.6 Avaliação de desempenho de técnicas de agrupamento 25
2.7 Trabalhos correlatos 27

3 METODOLOGIA 29
3.1 Classificação da pesquisa 29
3.2 Descrição dos dados 29
3.3 Metodologia de desenvolvimento 31
3.4 Materiais e tecnologias 32
3.5 Métodos e procedimentos 33
3.5.1 Pré-processamento *33*
3.5.2 Representação no espaço vetorial *34*
3.5.3 Redução de dimensionalidade *34*
3.5.4 Categorias de experimentos *35*

4 RESULTADOS E DISCUSSÕES 36
4.1 Descrição dos dados pré-processados 36
4.2 Redução de dimensionalidade 36
4.3 Agrupamento usando medida de validade externa 39

4.4 Agrupamento usando medida de validade interna 41
4.4.1 K-Means 41
4.4.2 Agrupamento Hierárquico Aglomerativo 42
4.4.3 DBSCAN 42
4.4.4 Modelo de Mistura Gaussiana 43
4.4.5 Resumo dos experimentos 44

5 CONSIDERAÇÕES FINAIS 46
5.1 Trabalhos futuros 47

REFERÊNCIAS 48

1 INTRODUÇÃO

A crescente quantidade de dados disponíveis atualmente em páginas da Internet facilitou a coleta, difusão e desenvolvimento de informações nas mais diversas áreas do conhecimento, incluindo o mundo do futebol. Portais de notícias e *blogs* especializados desempenham um papel importante na produção e disseminação de informações sobre jogadores, times, campeonatos e eventos relacionados. Com isso, o acesso a informações ficou mais fácil, contribuindo para a análise e tomada de decisão de forma rápida e segura. No entanto, a localização e leitura dessas informações em tempo hábil tornaram-se um desafio, visto que o volume de dados cresce de forma exponencial (MAGALHÃES, 2020).

A tarefa manual de análise e agrupamento dos dados relacionados ao futebol se torna trabalhosa devido à grande quantidade de notícias disponíveis. Além disso, pode ocorrer perda de informações no processo, dado o padrão possivelmente complexo nos dados. Por esse motivo, é interessante utilizar técnicas de agrupamento, a fim de organizar as informações provenientes das notícias de futebol. Com o uso dessas técnicas, é possível identificar padrões e relacionamentos ocultos nos dados, fornecendo suporte para a tomada de decisões mais embasadas.

A principal motivação deste trabalho foi a falta de publicações que abordem especificamente a aplicação de técnicas de agrupamento de dados em notícias de times de futebol. Essa análise possibilitou o entendimento de quais são as técnicas de agrupamento mais adequadas para o problema específico. Isso pode auxiliar pesquisadores em trabalhos futuros, permitindo aplicação de outras técnicas, tais como análise de sentimentos e predição de resultados.

O objetivo deste trabalho foi analisar e comparar as principais técnicas de agrupamento de dados aplicadas a notícias de times que participam do Campeonato Brasileiro de Futebol. Para alcançar esse objetivo, foi usada uma base de dados que cobre um período razoável de tempo. Além disso, foram aplicadas técnicas de pré-processamento para melhorar a qualidade e representação dos dados. Posteriormente, foram realizados diversos experimentos, com diferentes técnicas de agrupamento, incluindo agrupamento baseado em partições, em hierarquia, em densidade e em modelo. Por fim, foram feitas a análise das métricas obtidas para cada estratégia de agrupamento, a comparação entre os métodos e uma discussão dos resultados obtidos.

1.1 Objetivos

O principal objetivo do presente trabalho foi analisar as principais técnicas de agrupamento de dados aplicadas a notícias de times de futebol. Para atingir o objetivo principal, os seguintes objetivos específicos foram estabelecidos:

- estruturar a base de dados de notícias de times de futebol;

- selecionar as técnicas de agrupamento de dados mais adequadas para a base;
- analisar as técnicas de agrupamento por meio de experimentos utilizando a base de dados.

1.2 Justificativa

De acordo com Dhar *et al.* (2020), poucos estudos exaustivos foram realizados para a tarefa de agrupamento de texto. Esse número é ainda menor para notícias de futebol, com base na revisão feita na literatura.

A Descoberta de Conhecimento em Banco de Dados é uma área de grande importância, que visa extrair informações valiosas a partir de conjuntos de dados. Nesse contexto, este trabalho pode contribuir, uma vez que se concentra na análise de notícias relacionadas a times de futebol brasileiros, utilizando técnicas de Mineração de Texto. Além disso, esta pesquisa pode ajudar os profissionais que atuam em áreas afins ao futebol a tomar decisões mais embasadas e acertadas, economizando tempo e gerando impactos positivos na gestão dos negócios relacionados ao futebol.

1.3 Resultados esperados

Com a conclusão do presente trabalho, espera-se obter contribuições práticas e científicas relevantes. Do ponto de vista prático, a ideia é que este estudo auxilie *sites* a agrupar notícias de forma ágil e eficaz. Isso proporcionará benefícios aos veículos de mídia que lidam com grandes volumes de informação. Por meio da otimização desse processo, pretende-se tornar mais rápida a classificação das notícias, permitindo uma melhor organização e economia de tempo.

Do ponto de vista científico, o objetivo é identificar as técnicas de agrupamento mais adequadas para a base de dados específica de notícias de futebol. Ao alcançar esse objetivo, será possível fornecer um resultado de agrupamento confiável, que poderá ser utilizado como entrada para outras aplicações, como análise de sentimentos, predição de jogos e outras áreas relacionadas ao futebol.

1.4 Organização do documento

Este documento está estruturado em cinco capítulos, incluindo este. No Capítulo 2, são apresentados os fundamentos teóricos necessários, além de uma revisão dos trabalhos correlatos. O Capítulo 3 detalha a metodologia desenvolvida para a execução deste estudo. No Capítulo 4, são expostos os resultados obtidos, seguidos de uma análise e interpretação deles. Por fim, no Capítulo 5, são apresentadas as considerações finais do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os fundamentos necessários para o melhor entendimento deste estudo, bem como os trabalhos correlatos. A Seção 2.1 trata da Descoberta de Conhecimento em Banco de Dados. A Seção 2.2 aborda várias etapas do pré-processamento de texto. A Seção 2.3 trata da redução de dimensionalidade. A Seção 2.4 faz uma apresentação sobre o cálculo de similaridade. A Seção 2.5 discute sobre as principais abordagens de agrupamento de dados. A Seção 2.6 apresenta algumas medidas de avaliação de desempenho para algoritmos de agrupamento de dados. Por fim, a Seção 2.7 trata dos trabalhos correlatos.

2.1 Descoberta de Conhecimento em Banco de Dados

Os avanços constantes na Computação possibilitaram o armazenamento de quantidades cada vez maiores de dados. Tecnologias como a Internet, Sistemas Gerenciadores de Banco de Dados (SGBDs) modernos e dispositivos de memória secundária com alta capacidade e baixo custo permitem a difusão de bases de dados de natureza científica, comercial, administrativa e governamental (GOLDSCHMIDT; PASSOS; BEZERRA, 2015). Essa quantidade massiva de dados é conhecida como Big Data.

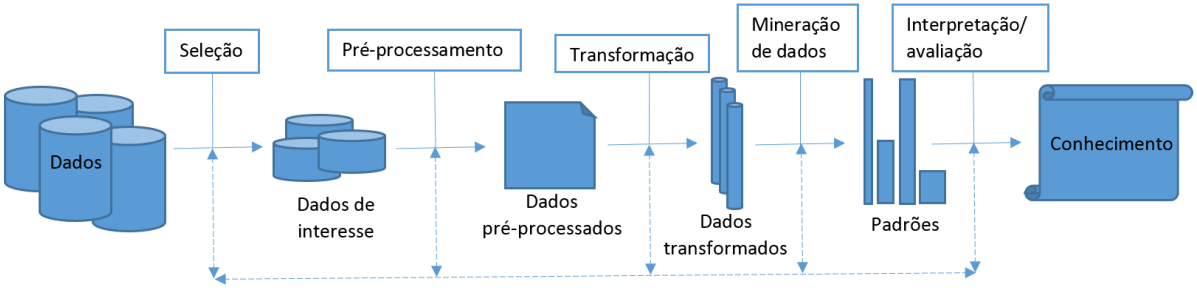
Atualmente, estima-se que sejam gerados, diariamente, terabytes de dados provenientes de fontes comerciais, redes sociais, sensores e outras fontes diversas. Esses dados possuem ampla variedade de formatos, abrangendo desde informações estruturadas, como dados numéricos, até dados não estruturados, como textos, imagens, vídeos e áudios, tornando-os complexos. Por consequência, nem sempre ferramentas tradicionais podem ser usadas no tratamento do Big Data (RIBEIRO, 2022). Considerando esse cenário, é essencial o desenvolvimento de ferramentas automatizadas capazes de analisar, relacionar e interpretar dados, uma vez que essa tarefa seria inviável para ser realizada manualmente por seres humanos.

A fim de auxiliar o estudo dos dados, surgiu um processo denominado *Knowledge Discovery in Databases* (KDD), que, em português, significa Descoberta de Conhecimento em Bancos de Dados. Esse processo tem como finalidade a identificação de padrões úteis a partir de grandes conjuntos de dados (ALLAHYARI *et al.* 2017). O processo de KDD pode ser dividido em cinco grandes etapas. A primeira é a etapa de seleção, que busca entender quais são os dados de interesse e qual o objetivo do processo de KDD. A segunda e a terceira etapas são o pré-processamento e transformação dos dados, respectivamente. Essas etapas buscam eliminar os ruídos e deixar os dados em um formato que possa ser interpretado pelos algoritmos de mineração da próxima etapa.

A quarta etapa é a mineração propriamente dita, em que são aplicadas técnicas que identificam padrões nos dados. Na quinta etapa, são realizadas a avaliação e a interpretação dos dados, na qual é possível obter conhecimento a partir da análise dos

padrões. A Figura 1 mostra as etapas da Descoberta de Conhecimento em Banco de Dados.

Figura 1 – Etapas do KDD



Fonte: FREITAS; MOURA; SILVA, 2015.

2.2 Pré-processamento de texto

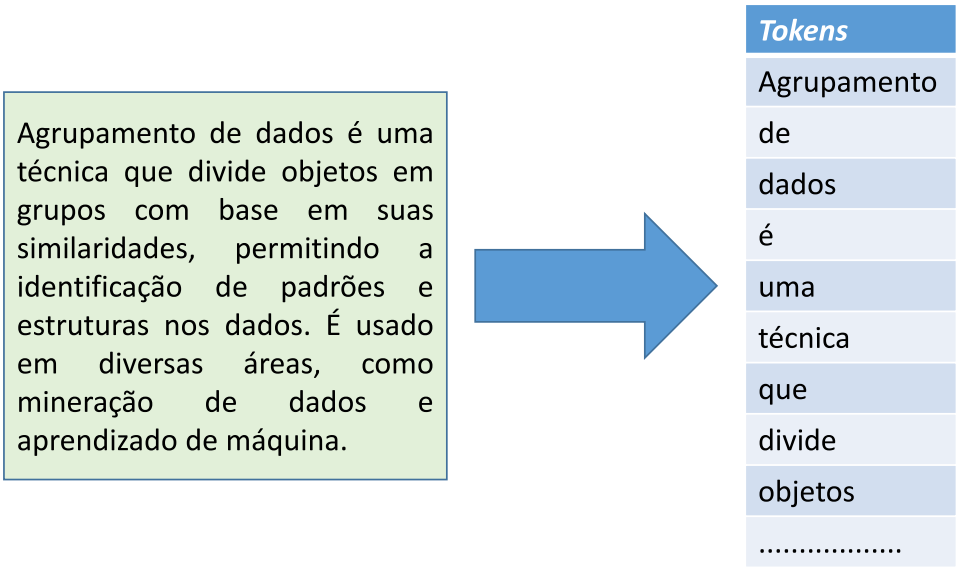
Ao contrário do formato numérico, o texto é essencialmente não estruturado, tornando praticamente impossível processá-lo diretamente em sua forma bruta. Não é viável aplicar operações numéricas a textos, e converter seu conteúdo em um valor numérico não é uma tarefa simples. Portanto, é necessário realizar etapas de pré-processamento no texto, as quais envolvem a separação das frases em uma lista de palavras individuais, a remoção de termos comuns que possuem pouca relevância para o contexto de mineração de texto e a conversão das palavras para sua forma raiz.

2.2.1 Tokenização

De acordo com Aggarwal (2018), a *tokenização* é definida como o processo de converter um texto em uma lista de palavras, os chamados *tokens*. A ocorrência de cada palavra do texto gera um *token*, portanto, três palavras iguais no texto geram três *tokens* correspondentes. O processo de *tokenização* depende de uma análise morfológica do idioma em questão, que define o critério a ser utilizado para segmentar as palavras do texto. Idiomas que têm origem no Latim, como é o caso do português, separam os *tokens* pelo espaço em branco ou pelos sinais de pontuação.

Como processamento subsequente, as palavras que incluem caracteres especiais ou valores numéricos são removidas, e os *tokens* são alterados para seus caracteres minúsculos. A Figura 2 apresenta o processo de *tokenização*, em que um documento é recebido como entrada, e uma lista de *tokens* é devolvida como saída.

Figura 2 – Etapa de *tokenização*



Fonte: Elaborado pelo Autor, 2023.

2.2.2 Remoção de palavras de parada

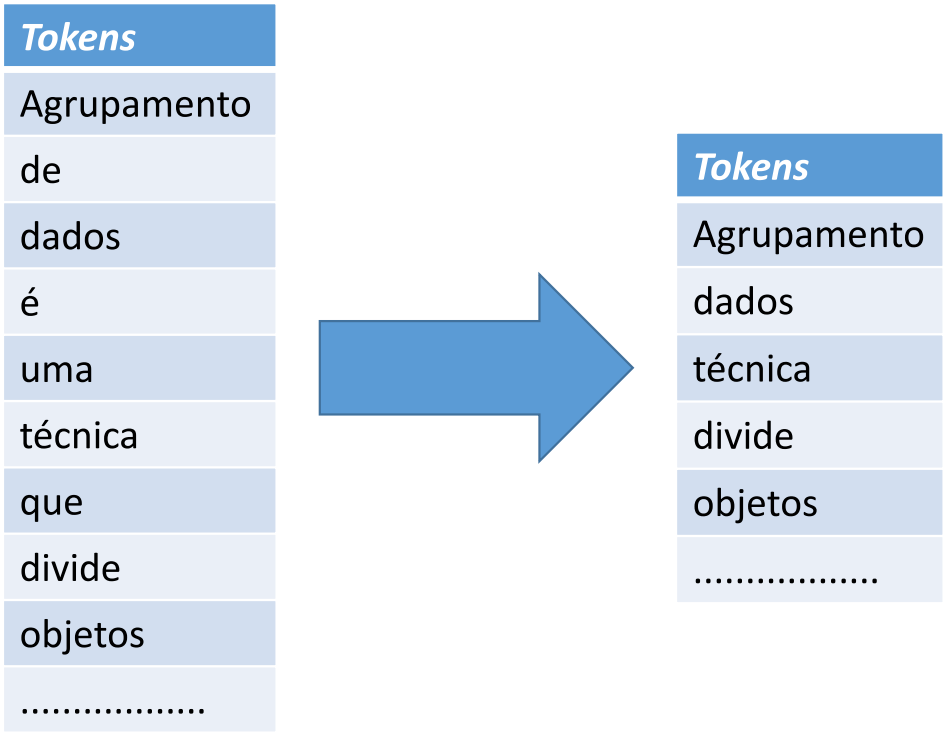
Segundo Aggarwal (2018), as *stop words*, ou palavras de parada, em português, são termos que não carregam relevância para análise do texto e devem ser removidas da lista de *tokens* para maior eficiência do processo. Todos os artigos, preposições, conjunções e, em alguns casos, pronomes são removidos. Além disso, existem dicionários disponíveis com palavras de parada de cada idioma para auxiliar na remoção desses termos. A Figura 3 apresenta o processo de remoção de palavras de parada, em que os *tokens* são filtrados, sendo mantidos apenas os termos mais relevantes para o processo de agrupamento.

2.2.3 Derivação

Stemming, que, em português, significa derivação, é o processo de mapear cada *token* para o seu radical (JO, 2019). Documentos de texto podem conter a forma singular ou plural da mesma palavra, vários tempos verbais e outras variações. Assim, essas palavras são consolidadas em uma única forma, pois tais variações não interferem na interpretação semântica do ponto de vista da mineração. Para a extração da raiz morfológica de uma palavra, são usadas algumas técnicas, como tabelas de consultas semiautomáticas, remoção de sufixos e lematização. As tabelas de consulta semiautomáticas são heurísticas criadas previamente para substituir uma gama de termos pelo seu radical. Um exemplo disso são as palavras “comer”, “comendo” e “comeu”, que possuem o mesmo radical “come”.

Outra estratégia é a remoção de sufixos, em que são armazenadas regras para tentar encontrar a forma raiz da palavra. Sufixos de plural, como “es”; de verbos, como “er” ou “ar”; de adjetivos, como “ão” ou “inho”; e de advérbios, como “mente”, podem ser removidos. Prefixos também podem ser removidos, no entanto, é mais comum apenas a

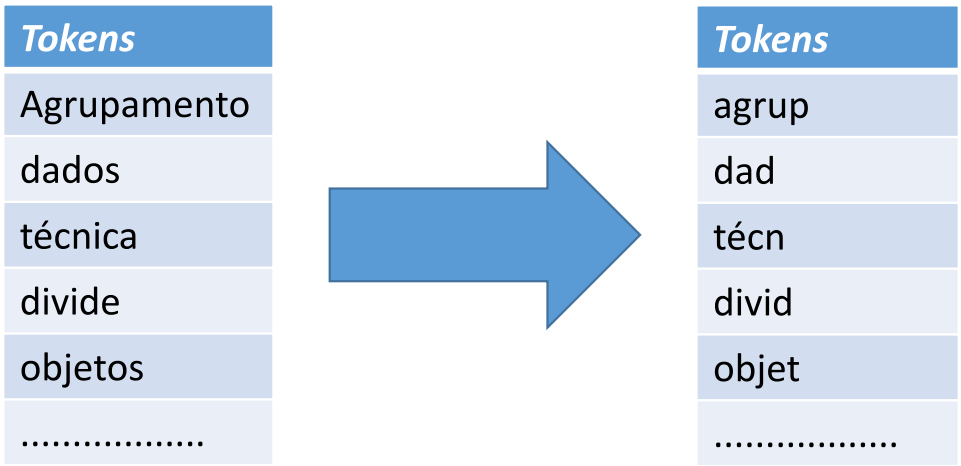
Figura 3 – Etapa de remoção de palavras de parada



Fonte: Elaborado pelo Autor, 2023.

remoção de sufixos. A Figura 4 apresenta o processo de derivação, em que cada *token* da lista é convertido para o seu radical.

Figura 4 – Etapa de derivação



Fonte: Elaborado pelo Autor, 2023.

2.2.4 Representação no espaço vetorial e normalização

Com os termos já extraídos e processados pelas etapas de *tokenização*, remoção de palavras de parada e derivação, o texto precisa ser convertido para uma notação de espaço vetorial, que é uma representação multidimensional e esparsa. Como descrito

por Aggarwal (2018), essa representação contém uma dimensão para cada palavra, e o valor dessa dimensão é positivo somente quando a palavra está no documento; caso contrário, seu valor é 0. O valor positivo pode ser uma frequência normalizada ou um valor de indicador binário 1. Cada documento, normalmente, possui uma quantidade muito pequena de palavras do vocabulário. É comum que o número médio de palavras de cada documento seja de algumas centenas, enquanto a quantidade total do vocabulário seja significativamente maior do que cem mil palavras.

O modelo que envolve a representação simplificada de um documento de texto como um conjunto desordenado de palavras, sem levar em consideração a estrutura gramatical ou a ordem das palavras, é conhecido como *bag of words*. Existem duas representações comumente usadas para esses dados, o modelo binário e o modelo *Term Frequency - Inverse Document Frequency* (TF-IDF). A sigla TF significa frequência do termo, e a sigla IDF, frequência inversa do documento. Para algumas aplicações, a representação binária para retratar a presença ou não do termo no documento é suficiente. Uma das principais vantagens é que ela é compacta, mas, em contrapartida, perde informações por não conter a frequência dos termos individuais nem a importância relativa das palavras no documento.

A maioria das representações não funciona bem com o modelo binário; por esse motivo, usa o modelo TF-IDF, que pondera o impacto de cada termo na representação vetorial do documento. Os pesos das palavras calculadas por esse esquema são proporcionais às ocorrências no texto dado, mas inversamente proporcionais aos de outros textos. Como as frequências das palavras em um documento longo podem variar significativamente, faz sentido usar funções de amortecimento nessas frequências. Nesse caso, a função logarítmica pode ser aplicada para reduzir o efeito do *spam*, que são termos muito frequentes e pouco informativos.

Para o cálculo da frequência inversa do documento, é divido o número total de documentos pelo número de documentos que contêm o termo, e a esse resultado é aplicado o logaritmo. A frequência do termo é obtida como a proporção do número de vezes que um termo aparece em um documento em relação ao número total de termos nele. A frequência TF-IDF é alcançada multiplicando-se a frequência do termo pela frequência inversa do documento.

Para ilustrar o cálculo da frequência TF-IDF, serão apresentados três documentos de texto como exemplo. A Tabela 1 mostra a representação TF-IDF aplicada a esses documentos. Como a tabela possui uma coluna para cada palavra presente nos documentos, são exibidas apenas para cinco termos, para facilitar a visualização.

1. **Documento 1:** Agrupamento de dados é uma técnica que divide objetos em grupos com base em suas similaridades, permitindo a identificação de padrões e estruturas nos dados.

2. **Documento 2:** É usado em diversas áreas, como mineração de dados e aprendizado de máquina.

3. **Documento 3:** Agrupamento de dados é uma técnica amplamente utilizada em análise de dados e reconhecimento de padrões.

Tabela 1 – Representação da matriz TF-IDF

	agrupamento	dados	técnica	mineração	similaridades
Documento 1	0.227959	0.354061	0.227959	0.000000	0.299738
Documento 2	0.000000	0.234400	0.000000	0.396875	0.000000
Documento 3	0.284809	0.442359	0.284809	0.000000	0.000000

Fonte: Elaborado pelo Autor, 2023.

2.3 Redução de dimensionalidade

Os dados textuais, frequentemente, enfrentam o desafio da alta dimensionalidade. Por esse motivo, é comum que a matriz que representa a ocorrência de termos nos documentos contenha milhares de dimensões. *Latent Semantic Analysis* (LSA), que, em português, significa Análise Semântica Latente, é uma técnica empregada para representar termos por meio de componentes latentes ou ocultos. Com isso, é possível obter uma representação mais compacta, uma vez que os termos passam a ser representados por tópicos (LANE; HOWARD; HAPKE, 2019). A LSA é uma técnica popular de redução de dimensionalidade que segue o método de *Singular Value Decomposition* (SVD), ou Decomposição em Valores Singulares.

O SVD é uma técnica matemática que decompõe uma matriz em três outras, sendo uma delas a representação dos conceitos latentes, que é usada no contexto da LSA. Uma propriedade da decomposição SVD é que a multiplicação das três matrizes resulta na matriz original. Uma analogia válida é a decomposição de um número inteiro A, que pode ser fatorado em outros três números inteiros, denominados B, C e D, e a multiplicação desses valores resulta novamente em A.

A maneira padrão de representar dados textuais é através de uma matriz documento-termo, em que as linhas representam os documentos, e as colunas, os termos ou palavras do vocabulário. Já os valores na matriz refletem a importância da palavra no documento. Se o número for zero, essa palavra simplesmente não existe no documento. Diferentes documentos tratarão de diferentes tópicos, por exemplo, saúde, economia e esportes. Algumas notícias podem pertencer unicamente a uma categoria, enquanto outras podem pertencer a duas ou mais.

A primeira matriz obtida pela Decomposição em Valores Singulares fornece justamente essa informação: as linhas representam as notícias; as colunas, os tópicos latentes; e os valores da matriz, o quanto cada termo está presente em determinado tópico.

Já a segunda matriz fornece a informação do peso de cada termo para a formação de um tópico latente. Portanto, as linhas dessa matriz correspondem aos termos; as colunas, aos tópicos; e os valores são os pesos. Por fim, a terceira matriz representa o quanto cada um dos tópicos explica os dados. Isso pode ser usado para entender se a quantidade de componentes escolhida extrai de forma correta os principais tópicos presentes nos documentos.

Ao utilizar LSA, é possível escolher a quantidade de tópicos a serem formados, mas escolher essa quantidade de tópicos para cada conjunto de documentos pode não ser uma tarefa muito simples. Evangelopoulos, Zhang e Prybutok (2012) apresentam uma análise de alguns trabalhos que mostram que o número de tópicos ideal pode variar de 6 a mais de 1000. Apresentam, também, estudos em que esse número ideal varia entre 70 e 100 para coleções com cerca de 1000 documentos e 5000 termos. Por fim, recomendam aos pesquisadores explorarem e reportarem valores de dimensionalidade encontrados em seus estudos. De acordo com Valdez, Pickett e Goodson (2018), decidir quantos tópicos são relevantes para um conjunto de dados representa um equilíbrio delicado entre especificidade e interpretabilidade, sendo a melhor solução aquela mais útil para o projeto em questão. Para fins de exemplificação, a Tabela 2 demonstra uma matriz LSA, que foi reduzida de 5 para 3 componentes.

Tabela 2 – Representação da matriz LSA

	Componente 1	Componente 2	Componente 3
Documento 1	0.77	0.07	0.62
Documento 2	0.80	0.55	0.20
Documento 3	0.97	0.13	0.19

Fonte: Elaborado pelo Autor, 2023.

2.4 Cálculo de similaridade

Uma vez que os documentos são representados no espaço vetorial, a distância euclidiana é comumente utilizada para calcular a similaridade entre documentos. Quanto menor a distância euclidiana entre dois vetores, maior é a similaridade entre os documentos correspondentes e vice-versa. No entanto, os valores da distância euclidiana serão mais altos para distâncias entre pares de documentos mais longos, mesmo que grandes frações desses documentos sejam comuns. Portanto, documentos mais longos e documentos mais curtos são tratados de maneira diferente, o que pode levar a resultados insatisfatórios para a mineração.

Para resolver esse problema, é necessário que o comprimento variável dos documentos seja normalizado. Uma solução é usar o cosseno do ângulo entre os vetores que representam os documentos, já que o cosseno entre dois vetores não depende do comprimento, e sim do ângulo (AGGARWAL, 2018). Essa representação normaliza o

efeito causado pelos comprimentos variáveis dos documentos, pois o denominador contém as normas deles. Além disso, a normalização também garante que o valor do cosseno esteja sempre no intervalo (0, 1).

2.5 Agrupamento de dados

Segundo Rai e Singh (2010), Agrupamento de Dados, também chamado de *Clustering*, é uma técnica que consiste na separação de dados em subgrupos que contêm objetos semelhantes entre si, denominados *clusters*. A modelagem dos dados por meio de agrupamento perde alguns detalhes, mas obtém simplificação, representando muitos objetos de dados por meio de poucos grupos. Na perspectiva de aprendizado de máquina, o agrupamento é um aprendizado não supervisionado que busca por dados ocultos, ou seja, os algoritmos encontram padrões e estruturas por conta própria, sem a orientação explícita de um humano. O agrupamento é frequentemente usado como uma das primeiras etapas na análise de Mineração de Dados, identificando grupos de registros relacionados que podem ser usados como ponto de partida para explorar outros relacionamentos.

De acordo com Kaushik (2016), o agrupamento pode ser dividido em duas categorias: *Hard Clustering* e *Soft Clustering*. No primeiro método, o objeto pertence estritamente a um único grupo ou a nenhum; já no segundo, o objeto pode pertencer simultaneamente a mais de um grupo.

Ghosal *et al.* (2020) explica que o objetivo do agrupamento depende muito da situação em que se deseja aplicá-lo. Para cada uma, é proposta uma metodologia diferente para especificar a proximidade dos pontos dados. Até o momento, mais de 100 algoritmos diferentes de agrupamento foram propostos e estudados (KAUSHIK, 2016). Todos eles podem ser divididos em cinco subconjuntos distintos, discutidos nas seções a seguir.

2.5.1 Agrupamento particional

O agrupamento particional é uma abordagem iterativa que encontra similaridades entre os pontos de dados em um grupo com base na distância até o seu centroide. O algoritmo escolhe pontos iniciais para representar cada grupo. Logo depois, é medida a distância entre cada ponto dos dados e o ponto escolhido como representante de cada grupo. Com base nessas distâncias, é decidido quais pontos dos dados devem pertencer a qual grupo. Em seguida, os pontos centrais de cada grupo são recalculados, e o processo segue de maneira iterativa. O resultado final é uma divisão dos dados em grupos semelhantes entre si e diferentes dos demais grupos. Exemplos de técnicas de agrupamentos particional são o K-Means, o K-Medoids e o K-Modes.

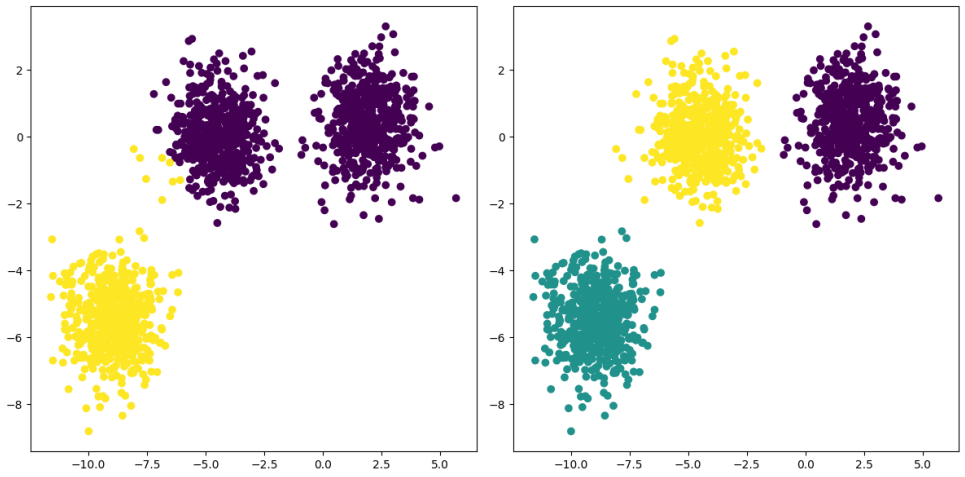
O algoritmo K-Means é uma técnica de agrupamento simples e bastante utilizada que separa o conjunto de dados em k partições, sendo o parâmetro k fornecido como entrada para o algoritmo. O processo inicia com a seleção aleatória de k elementos da base

de dados para serem os centroides iniciais, que representam os grupos (LIKAS; VLASSIS; VERBEEK, 2003). A partir disso, cada elemento é associado ao centroide mais próximo utilizando a distância euclidiana como métrica de proximidade.

Em cada iteração, os pontos são movidos de um grupo para outro, e os centroides são atualizados como a média dos elementos associados a cada um deles, a fim de melhorar a qualidade do agrupamento. A qualidade do agrupamento é avaliada com base nos erros quadráticos, que são a soma das distâncias ao quadrado entre os elementos e o centroide de cada grupo. O objetivo do K-Means é encontrar k grupos, de modo que a soma dos erros quadráticos seja a menor possível. Portanto, as iterações ocorrem até que a soma se estabilize e não haja mais mudanças de elementos entre os grupos (RIBEIRO, 2022).

A Figura 5 apresenta gráficos com resultados da execução do algoritmo K-Means com o valor de k igual a 2 e 3, respectivamente. Cada cor representa um grupo identificado pelo algoritmo.

Figura 5 – Gráficos da execução do algoritmo K-Means com 2 e 3 grupos



Fonte: Elaborado pelo Autor, 2023.

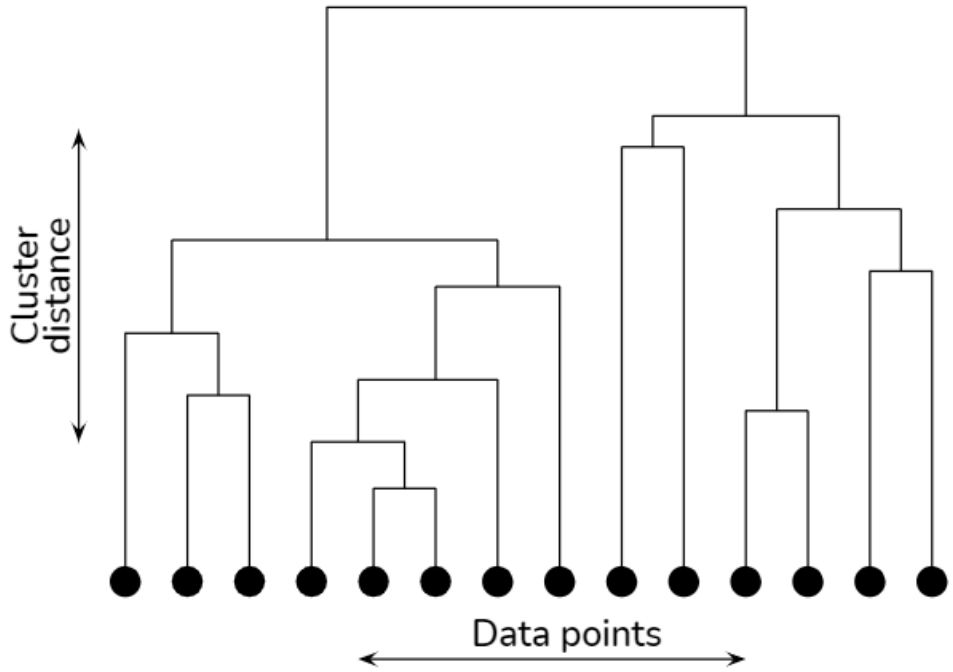
2.5.2 Agrupamento hierárquico

O agrupamento hierárquico é um modelo que tem duas abordagens: aglomerativa e divisiva. Na abordagem aglomerativa, cada ponto de dados começa como seu próprio grupo, e, em seguida, usando uma medida de proximidade, os pontos mais próximos são unidos em um único grupo. Isso é feito repetidamente até que todos os pontos de dados estejam agrupados. Já a abordagem divisiva do agrupamento hierárquico começa com um único grupo, que contém todos os pontos de dados e, à medida que avança, esse grupo é subdividido em grupos menores. Exemplos de algoritmos que utilizam essa abordagem são o Agrupamento Hierárquico Aglomerativo, o Agrupamento Hierárquico Divisivo e o *Balanced Iterative Reducing and Clustering using Hierarchies* (BIRCH).

O Agrupamento Hierárquico Aglomerativo começa tratando cada ponto do conjunto de dados como um grupo separado. Em seguida, a similaridade entre todos os

pares de pontos é calculada (MURTAGH; CONTRERAS, 2011). No próximo passo do algoritmo, ocorre a união dos dois grupos mais semelhantes em um único grupo. Após essa união, a matriz de similaridade é atualizada para refletir as mudanças realizadas nos grupos. Esses passos são repetidos iterativamente até que todos os grupos sejam combinados em um único grande grupo. A Figura 6 apresenta um dendrograma, uma representação gráfica que mostra como os elementos do conjunto de dados são agrupados e ligados em um diagrama de árvore.

Figura 6 – Dendrograma



Fonte: SCIENCE, 2023.

De acordo com Ribeiro (2022), os hiperparâmetros mais importantes do algoritmo aglomerativo são o número de grupos e o critério de ligação. Hiperparâmetro é um parâmetro externo ao modelo de aprendizado de máquina cujo valor é definido antes do treinamento e influencia no desempenho deste. O critério de ligação determina qual métrica de distância usar entre grupos com possibilidade de junção. Normalmente, os valores mais interessantes para o parâmetro do critério de ligação são *single* (simples) e *ward* (similaridade). No critério de ligação simples, a ligação entre grupos ocorre com base na distância mínima entre os pontos mais próximos de diferentes grupos, o que pode resultar na formação de grupos mais alongados. Já a ligação por similaridade busca minimizar a variância dentro de cada grupo, funcionando melhor para agrupamentos esféricos e elípticos. O algoritmo irá mesclar os pares de agrupamento que minimizam esses critérios.

2.5.3 Agrupamento baseado em densidade

Agrupamento baseado em densidade une os pontos de dados em um grupo se eles estiverem próximos uns dos outros em uma região de alta densidade, e usa as regiões de baixa densidade como partição. Ele começa com pontos de dados arbitrários, que ainda não foram visitados, e verifica sua vizinhança. Se houver um número suficiente de pontos dentro de uma determinada distância, um grupo é formado. Essa operação é repetida até que todos os pontos sejam visitados. Alguns exemplos de algoritmos de agrupamento baseado em densidade são o *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN) e o *Ordering Points To Identify the Clustering Structure* (OPTICS).

O algoritmo DBSCAN é uma técnica de agrupamento de dados que se baseia na densidade dos pontos (RIBEIRO, 2022). Isso significa que os grupos são criados considerando-se uma área definida por uma distância máxima entre os pontos. Além disso, é necessário um mínimo de vizinhos dentro dessa área.

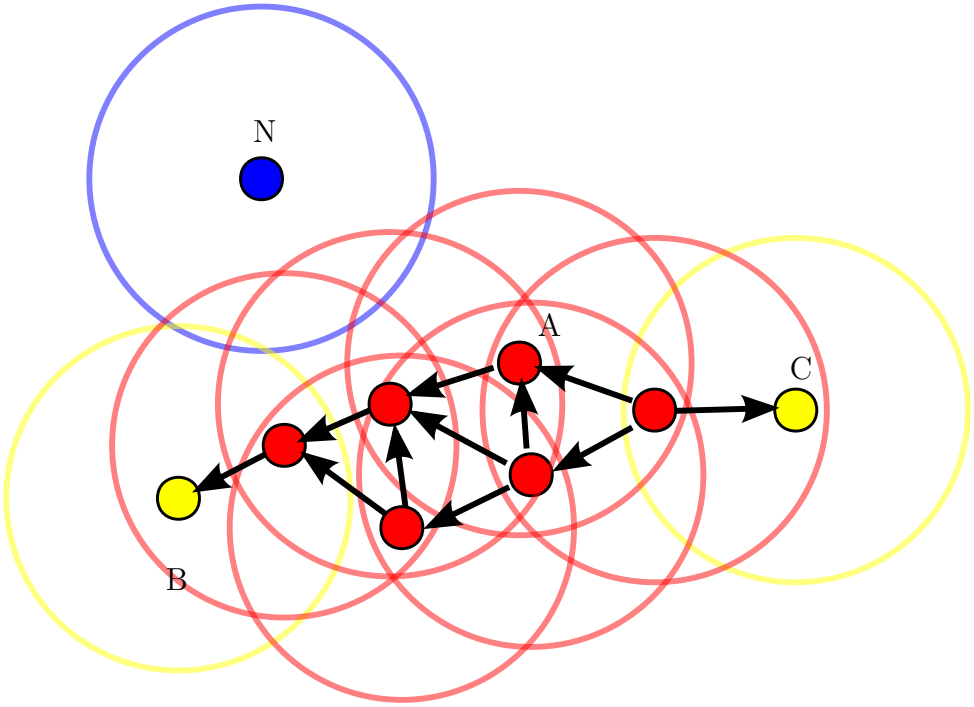
O DBSCAN identifica pontos centrais, pontos de borda e pontos de ruído em um conjunto de dados, formando agrupamentos com base na proximidade e densidade deles. Na Figura 7, os pontos centrais são representados em vermelho; os pontos de borda, em amarelo; e o ruído, em azul. Um ponto é considerado central se o número de pontos em sua vizinhança, dentro de uma distância definida, for igual ou maior que um valor mínimo predefinido. Um ponto de borda é aquele que não é central, mas está dentro da vizinhança de um ponto central. Já um ponto de ruído é aquele que não é um ponto central e não está na vizinhança de qualquer ponto central.

Entre as principais vantagens do DBSCAN, está o fato de que ele consegue identificar agrupamento com variadas formas e tamanhos. Além disso, ele é robusto em relação a ruídos, uma vez que não força a entrada desses pontos em *clusters*, tornando o algoritmo confiável em conjuntos de dados com pontos imprevisíveis. No entanto, esse algoritmo também possui algumas desvantagens. A primeira delas é que ele não se sai bem em situações em que a densidade dos clusters varia muito ao longo do conjunto de dados. Outra desvantagem é possuir uma complexidade computacional alta para conjuntos de dados grandes. Por último, sua eficácia depende muito dos hiperparâmetros informados pelo usuário, o que pode ser uma limitação (SINGH; GIRDHAR; DAHIYA, 2022).

2.5.4 Agrupamento baseado em modelo

Os algoritmos de agrupamento baseados em modelo fazem uso de muitos modelos estatísticos ou matemáticos predefinidos para formar grupos. O algoritmo trabalha com a mistura de probabilidades e cria grupos com base nelas. Essa técnica é útil quando os dados não possuem uma estrutura clara de agrupamento e é necessário encontrar padrões sutis nos dados. Exemplos desses algoritmos são o Modelo de Mistura Gaussiana e o COBWEB.

Figura 7 – Agrupamento com DBSCAN



Fonte: WIKIPEDIA, 2023a.

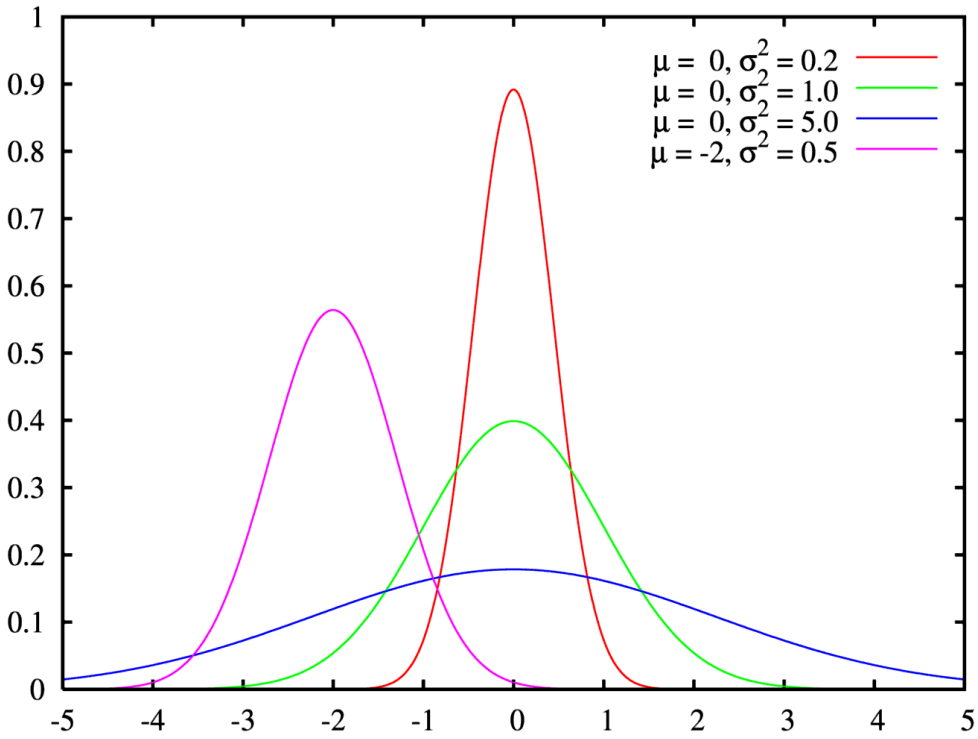
Gaussian Mixture Modelling (GMM), que, em português, significa Modelo de Mistura Gaussiana, é um modelo probabilístico que pode ser usado para o agrupamento de dados. Ele é construído de uma forma um pouco diferente dos algoritmos discutidos até o momento. Os grupos são representados por distribuições de probabilidades, como Distribuição Normal ou Distribuição de Poisson, a depender do tipo de dado. A Figura 8 mostra as curvas de algumas distribuições gaussianas. O algoritmo de mistura considera que cada grupo é representado por uma distribuição normal multivariada com média nos baricentros dos grupos (RELVAS, 2020).

Assim como no K-Means, a determinação da quantidade de grupos a serem criados no GMM é estabelecida como um hiperparâmetro do algoritmo. Em seguida, uma curva gaussiana é gerada para cada grupo, tendo o centroide como valor médio da curva. Assim, a gaussiana representa a probabilidade de cada ponto pertencer a um determinado grupo. O algoritmo itera para modelar a curva gaussiana que melhor se ajuste aos dados, atualizando os parâmetros da distribuição.

Os parâmetros das distribuições são encontrados usando-se o algoritmo *Expectation Maximization* (EM). Esse método consiste em encontrar os melhores valores para as curvas da distribuição. Na primeira etapa, busca-se calcular o valor esperado da função, utilizando-se os valores encontrados no passo anterior como estimativa dos parâmetros.

Na segunda etapa, a meta é maximizar a função obtida na fase de expectativa para gerar novas estimativas dos parâmetros do modelo. O algoritmo encerra sua execução após atingir a convergência, ou seja, até que as mudanças nos parâmetros sejam mínimas. Com os parâmetros estimados, o GMM identifica as distribuições de densidade que

Figura 8 – Gráfico de distribuição gaussiana



Fonte: WIKIPEDIA, 2023b.

representam cada grupo e fornece a probabilidade de cada ponto pertencer a cada grupo.

2.6 Avaliação de desempenho de técnicas de agrupamento

Algoritmos de agrupamento podem ser avaliados através de medidas de validades interna ou externa (AGGARWAL, 2018). As medidas de validade interna usam critérios, como a similaridade média do cosseno, com o centroide do *cluster* mais próximo, para avaliar um agrupamento. Esse critério também é usado como função objetivo por algoritmos de agrupamento, como o K-Means.

A maioria das medidas de validade interna utiliza critérios derivados das funções objetivo de vários algoritmos de agrupamento ou está relacionada a esses critérios de alguma forma. Isso cria um problema no uso de medidas de validade interna para comparar, de forma justa, dois algoritmos de agrupamento com funções objetivo muito diferentes. Por exemplo, se for utilizada uma medida baseada em similaridade média do cosseno com o centroide do *cluster* mais próximo, não é possível que outro algoritmo de agrupamento consiga superar o K-Means para o mesmo número de *clusters*.

Sendo assim, o problema é que a medida não fornece informações sobre a qualidade de um agrupamento específico, mas sim sobre o quão bem o critério de um algoritmo de agrupamento específico corresponde ao critério de avaliação. Portanto, é importante ter atenção ao usar esse tipo de avaliação de agrupamento, pois ela pode ser tendenciosa e levar a interpretações errôneas sobre a precisão dos algoritmos.

O coeficiente de silhueta é definido para cada amostra e é composto por duas

pontuações, sendo a primeira a distância média entre uma amostra e todos os outros pontos da mesma classe, e a segunda, a distância média entre uma amostra e todos os outros pontos no agrupamento mais próximo. Suas vantagens incluem um intervalo limitado entre -1 e 1, em que valores próximos a -1 indicam agrupamento incorreto, e valores próximos a 1, agrupamentos densos e bem separados. Pontuações em torno de zero significam *clusters* sobrepostos. O coeficiente se alinha com o conceito padrão de *clusters*, pois pontuações mais altas estão associadas a *clusters* mais densos e distintos. No entanto, tende a produzir pontuações mais altas para *clusters* convexos em comparação com estruturas não convexas ou complexas (BUITINCK *et al.* 2013).

As medidas de validade externa usam os rótulos para avaliar o agrupamento, assim como os usados em problemas de aprendizado supervisionado. Como a variável dependente não é usada pelo algoritmo de agrupamento, o critério é externo ao algoritmo e ao conjunto de dados usado para o agrupamento. Assim, é verificado se os rótulos estão espalhados aleatoriamente pelos *clusters* ou se cada *cluster* é dominado por um rótulo específico. Uma medida muito empregada é a acurácia, que indica com que precisão um modelo faz previsões corretas. Em termos simples, a função de acurácia pode ser expressa como uma fração ou como a contagem de previsões corretas.

Dado o conhecimento das classes reais e das atribuídas pelo agrupamento, a *adjusted mutual information* (AMI) é uma função que mede a concordância das duas atribuições, ignorando as permutações. Suas vantagens incluem uniformidade, interpretação clara e uma faixa de valores limitada entre 0 e 1, indicando concordância perfeita quando igual a 1. Já as desvantagens do AMI são a dependência de rótulos reais, aplicabilidade limitada em configurações não supervisionadas e falta de ajuste contra o acaso (BUITINCK *et al.* 2013).

O *rand index* é uma medida usada para avaliar a similaridade entre dois conjuntos de rótulos em algoritmos de agrupamento. Suas vantagens incluem interpretabilidade, uniformidade, um intervalo limitado e aplicabilidade a vários algoritmos de agrupamento. No entanto, possui algumas desvantagens, como dependência de rótulos reais, diferenciação limitada entre agrupamentos e a necessidade de técnicas adicionais para seleção de modelo (BUITINCK *et al.* 2013).

O *Fowlkes–Mallows index* (FMI) é calculado como a média geométrica da precisão e da revocação de pares. A precisão é a proporção de pares de pontos que são classificados corretamente em ambos os rótulos. Já a revocação é a proporção de pares de pontos que são classificados corretamente nos rótulos reais. Suas vantagens incluem uniformidade, interpretação clara e ausência de suposições sobre a estrutura do *cluster*. O FMI atribui valores próximos a 0 para atribuições de rótulos aleatórios e valores próximos a 1 para alta concordância. No entanto, sua dependência de rótulos reais é uma desvantagem, pois eles, geralmente, não estão disponíveis ou exigem atribuição manual (BUITINCK *et al.* 2013).

2.7 Trabalhos correlatos

Nesta seção, são apresentados trabalhos realizados na área de agrupamento de dados, como livros, pesquisas e artigos de revisão da literatura. Esses trabalhos fornecem perspectivas teóricas e práticas das principais técnicas e suas aplicações.

Afonso e Duque (2014) relataram, em seu trabalho, os resultados de experimentos sobre agrupamento automático de texto aplicado em artigos científicos e textos de jornais em português brasileiro. Cada experimento do estudo foi separado em quatro procedimentos, sendo eles a seleção do conjunto de documentos, a seleção de classe de palavras, os algoritmos de filtragem e os algoritmos de agrupamento. O objetivo do trabalho foi encontrar o método mais eficaz para o processo de agrupamento, incluindo a correção do agrupamento e o tempo de agrupamento, usando como referência a classificação humana para correção.

Com relação aos resultados do agrupamento, foi observado que os índices de acerto do agrupamento variam de acordo com o número de textos de entrada e seus tópicos. Alguns métodos se saíram melhor na qualidade do agrupamento, mas ao custo de se gastar mais tempo. Outros métodos se saíram melhor com um tipo específico de conjunto de documentos. Portanto, o problema de se avaliar agrupamento de texto é algo relativo, e o conceito de correção e qualidade para agrupamento depende das expectativas de cada um.

Marutho *et al.* (2018) propuseram um método de agrupamento de notícias com base em suas manchetes. Para isso, eles empregaram a ponderação dos documentos utilizando o método TF-IDF e o algoritmo K-Means como estratégia de agrupamento. Além disso, aplicaram o método de cotovelo para determinar o número ótimo de *clusters* e a medida de pureza para avaliar a qualidade do agrupamento. Os autores do trabalho concluíram que o método do cotovelo é eficaz na otimização do número de agrupamentos no algoritmo K-Means.

Tong e Gu (2019) abordam algumas limitações dos métodos tradicionais de agrupamento de texto e propuseram uma abordagem diferente para superar essas questões. O algoritmo proposto utiliza rótulos para representar o conteúdo das notícias, melhorando a questão da alta dimensionalidade dos dados e a dificuldade de expressar os *clusters*. Para lidar com textos longos, o método proposto converte o texto em um código *hash*, que são os rótulos, e calcula a similaridade entre eles. Após isso, é realizado o agrupamento utilizando-se técnicas de agrupamento hierárquico. Os resultados experimentais demonstram que o algoritmo baseado na similaridade de rótulos de texto melhora a qualidade do agrupamento em comparação com os métodos tradicionais.

Aggarwal (2018) abordou, de forma abrangente, os tópicos de Recuperação de Informações e Mineração de Texto, com ênfase no uso de aprendizado profundo para processamento de linguagem natural. O autor conseguiu simplificar a apresentação matemática desses conceitos complexos, oferecendo explicações intuitivas e acessíveis.

O presente trabalho se diferencia dos demais ao aplicar técnicas de agrupamento,

especificamente as notícias do Campeonato Brasileiro de Futebol. Enquanto, alguns estudos abordaram agrupamento de texto em contextos mais amplos, como artigos científicos, de jornais ou manchetes de notícias. Este trabalho se destaca pela aplicação dessas técnicas no cenário esportivo brasileiro, e tal abordagem pode sugerir *insights* únicos para o campo da análise de dados e predição de resultados no contexto específico do futebol brasileiro.

3 METODOLOGIA

Neste capítulo, serão abordados os materiais e métodos adotados para a execução do presente trabalho. A Seção 3.1 apresenta a classificação da pesquisa. A Seção 3.2 descreve a origem e estrutura dos dados utilizados no trabalho. A Seção 3.3 apresenta a metodologia de desenvolvimento, e a Seção 3.4 trata dos materiais e das tecnologias utilizadas. Para finalizar, a Seção 3.5 apresenta os métodos e procedimentos empregados para a construção do trabalho.

3.1 Classificação da pesquisa

Quanto à abordagem, a pesquisa é quantitativa, pois busca analisar os resultados de técnicas de agrupamento de texto aplicadas a um conjunto de notícias de futebol através de métricas e análises estatísticas. Quanto à natureza, a pesquisa é aplicada, já que objetiva conhecer as técnicas mais efetivas para aplicar ao problema de agrupamento de notícias de futebol. Quanto aos objetivos, a pesquisa é exploratória, pois procura explorar as principais técnicas de agrupamento de texto utilizadas para análise de notícias de futebol. Quanto aos procedimentos, a pesquisa é experimental, visto que envolve a aplicação de diferentes técnicas de agrupamento de texto em um conjunto de notícias de futebol, buscando avaliar os resultados obtidos (GERHARDT; SILVEIRA, 2009).

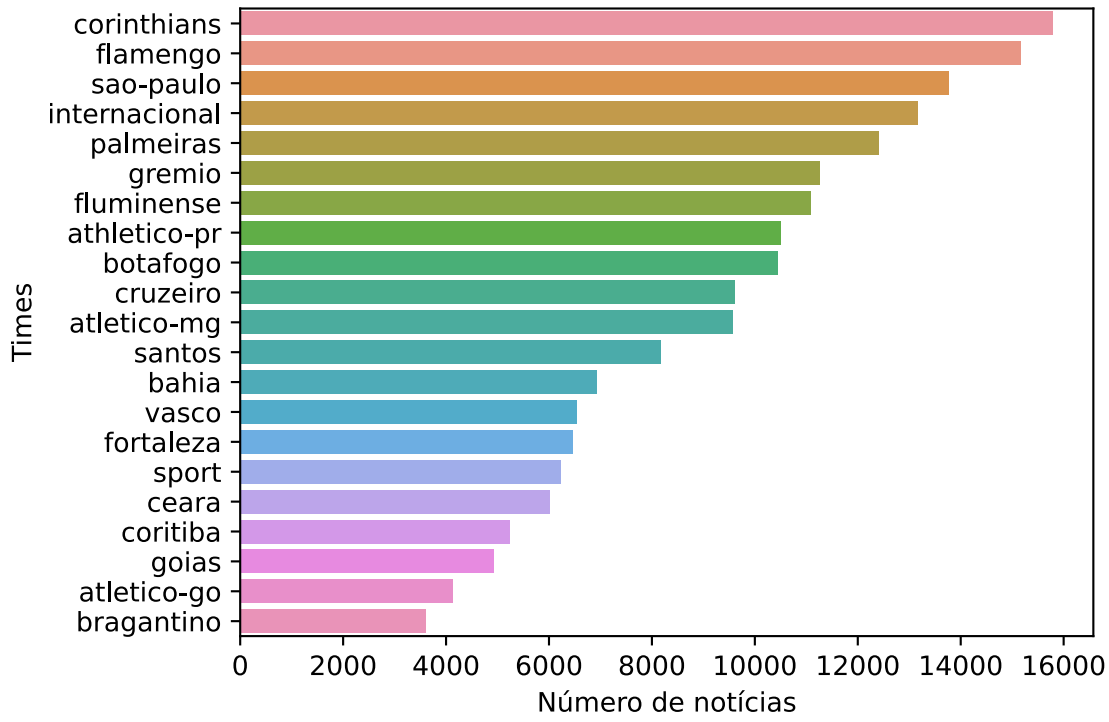
De acordo com Wazlawick (2009), este trabalho tem o estilo "apresentação de algo presumivelmente melhor", uma vez que busca comparar técnicas existentes de forma quantitativa através de testes. Um aspecto importante nesse tipo de pesquisa são as métricas usadas para a comparação, as quais permitem confirmar qual estratégia se mostra mais adequada, dentro da definição estabelecida.

3.2 Descrição dos dados

Para a realização deste trabalho, foi utilizada a base de dados denominada GE *Soccer Clubs News*, a qual foi criada a partir de notícias extraídas de forma aleatória do site do Globo Esporte (GE), abrangendo o período de 2015 a 2022. A base contém um total de 191005 notícias, distribuídas em 21 categorias, sendo que cada categoria representa um time. A média de notícias por time é de 9095. A Figura 9 apresenta um gráfico que ilustra a distribuição da quantidade de notícias por time.

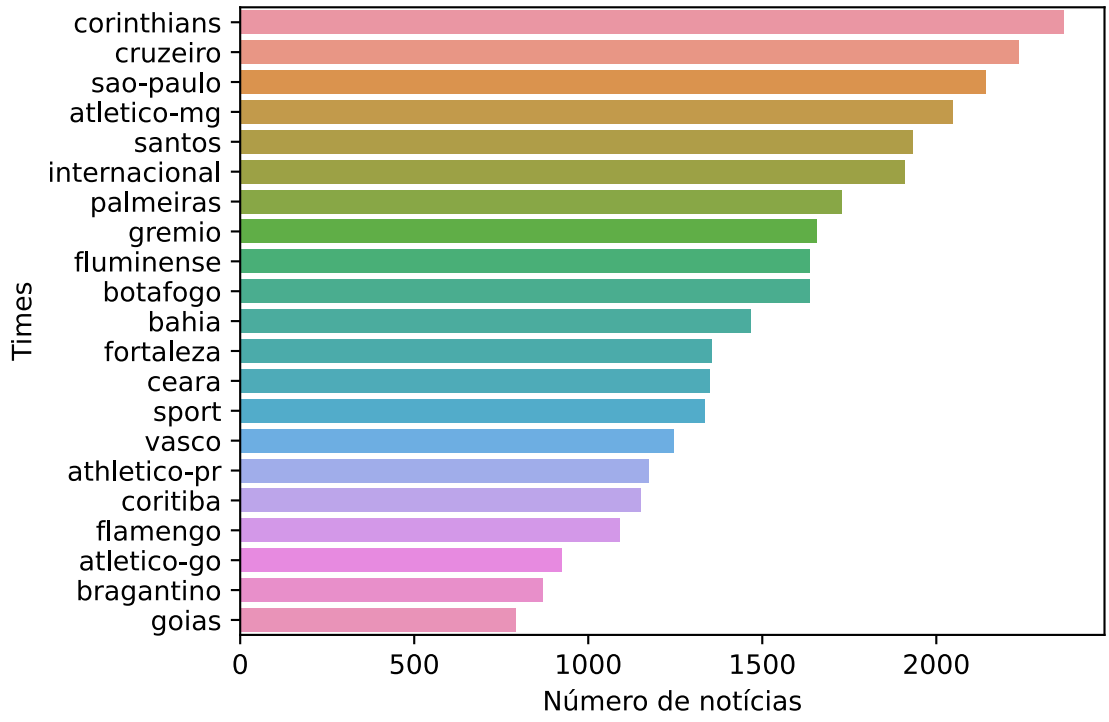
Para a realização do presente trabalho, foram utilizadas apenas as notícias da base de dados pertencentes ao ano de 2022. Essa decisão foi tomada porque esse é o ano mais recente para o qual todas as notícias estão disponíveis. A Figura 10 apresenta um gráfico que exibe a distribuição da quantidade de notícias por time apenas para o ano de 2022.

Figura 9 – Distribuição da quantidade de notícias por time



Fonte: Elaborado pelo Autor, 2023.

Figura 10 – Distribuição da quantidade de notícias por time em 2022



Fonte: Elaborado pelo Autor, 2023.

De acordo com Moneda (2023), autor da base, o conjunto de dados fornece uma amostra de dados do mundo real, com uma coluna contendo o nome do clube, que

pode ser considerada uma classe. O objetivo da base é permitir análises ao longo do tempo, como análise de sentimentos, classificação, entre outras técnicas pertinentes ao tema. Além disso, o autor também compartilhou o *script* de geração da base de dados em um repositório online, permitindo a atualização dos dados. É importante ressaltar que todo o conteúdo das notícias é propriedade da GE, e a disponibilização desse conjunto de dados visa permitir a experimentação e pesquisa no campo da Ciência de Dados.

A base de dados está disponível na plataforma Kaggle[1], reconhecida como uma comunidade online de cientistas de dados e engenheiros de aprendizado de máquina. O Kaggle proporciona aos usuários a oportunidade de encontrar conjuntos de dados relevantes para a construção de modelos de Inteligência Artificial (IA), além de oferecer oportunidades de publicação, interação com profissionais que atuam na área e participação em competições para resolver desafios de Ciência de Dados.

3.3 Metodologia de desenvolvimento

Como metodologia de desenvolvimento do sistema, foi utilizado o modelo Scrum. O foco dessa metodologia é proporcionar uma forma flexível de trabalho para os membros da equipe, permitindo a produção de *software* em um ambiente sujeito a constantes mudanças. É incentivada a estreita colaboração entre a equipe, assim como uma comunicação eficiente, a fim de facilitar a troca de informações e o entendimento por parte de todos.

Alguns dos princípios do Scrum incluem a formação de equipes pequenas, requisitos pouco estáveis ou desconhecidos e utilização de iterações curtas para promover visibilidade no processo de desenvolvimento (SACHDEVA, 2016). Essa abordagem é adequada para este trabalho, uma vez que, nas fases iniciais, os requisitos podem sofrer alterações. Além disso, as iterações curtas permitem identificar rapidamente áreas que precisam de melhorias. No contexto deste trabalho, foram adotadas *sprints* com duração de uma semana, seguidas por reuniões ao término de cada *sprint* para avaliar o progresso. Durante essas reuniões, ocorria uma análise das tarefas concluídas na semana, abordando também as dificuldades enfrentadas e sugestões relevantes. Ao final, uma lista de tarefas a serem realizadas na próxima semana era elaborada.

Para aprimorar e complementar a utilização da metodologia Scrum, foi utilizado um quadro Kanban. Essa estratégia visa otimizar o fluxo de valor por meio de um processo visual facilitado e da limitação do trabalho em progresso, baseado em um sistema de demandas. Nesse sistema, o time inicia o trabalho em um item somente quando percebe que tem a capacidade de fazê-lo (VACANTI; YERET, 2021). O emprego do Kanban é útil, pois proporciona uma visão geral do fluxo de trabalho, auxiliando na priorização e foco. Além disso, a limitação do escopo em cada etapa contribui para um ritmo equilibrado e

[1] https://www.kaggle.com

eficiente de trabalho.

3.4 Materiais e tecnologias

As implementações apresentadas neste estudo foram feitas utilizando-se várias ferramentas e bibliotecas, que são listadas a seguir:

- Python, versão 3.11.0 (https://www.python.org/);
- VSCode, versão 1.84.0 (https://code.visualstudio.com/);
- Pandas, versão 2.1.2 (https://pandas.pydata.org/);
- NumPy, versão 1.26.0 (https://numpy.org/);
- NLTK, versão 3.8.1 (https://www.nltk.org/);
- SpaCy, versão 3.7.2 (https://spacy.io/);
- Scikit-learn, versão 1.3.2 (https://scikit-learn.org/);
- Seaborn, versão 0.13.0 (https://seaborn.pydata.org/);
- Matplotlib, versão 3.8.0 (https://matplotlib.org/);
- GitHub (https://github.com/).

A linguagem de programação utilizada foi o Python, que é versátil e muito usada para se desenvolver aplicativos, *scripts* e soluções em áreas que vão desde desenvolvimento *web* e científico até automação de tarefas e inteligência artificial. Os gráficos gerados foram plotados utilizando-se as bibliotecas Matplotlib e Seaborn, que têm como objetivo criar gráficos e visualizações de dados de forma eficaz e personalizada. Para criação de estruturas de dados, foi utilizado o Pandas, uma biblioteca empregada para manipular e analisar dados de forma eficiente, especialmente em formato de tabelas ou DataFrames. Além disso, foi usado o NumPy, uma biblioteca que permite operações eficientes em matrizes e *arrays* multidimensionais.

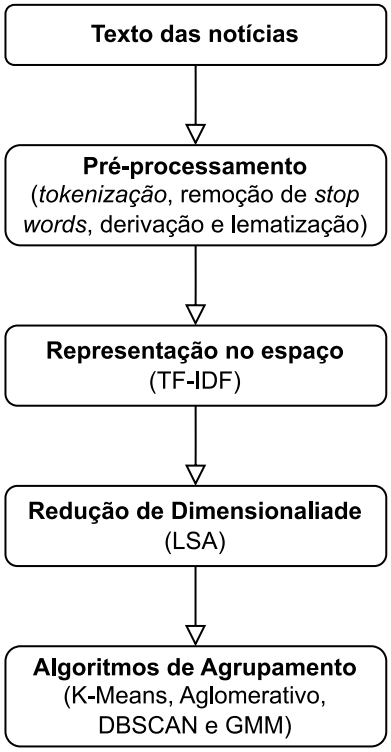
As bibliotecas NLTK e SpaCy foram utilizadas para processamento de linguagem natural, auxiliando no processo de *tokenização*, remoção de palavras de parada, entre outros. O *framework* Scikit-learn foi empregado para auxiliar nas implementações dos modelos de representação de texto no espaço vetorial e algoritmos de agrupamento. O ambiente de desenvolvimento integrado VSCode foi utilizado para codificar e testar o sistema, em ambiente do sistema operacional Linux. Os testes foram feitos utilizando-se o interpretador padrão do sistema operacional. Para fins de versionamento de código, segurança e backup, os protótipos gerados foram persistidos via GitHub.

3.5 Métodos e procedimentos

Esta seção trata das etapas desenvolvidas durante a execução do trabalho. Na Seção 3.5.1, são abordadas as técnicas de pré-processamento utilizadas. Em seguida, a Seção 3.5.2 aborda a implementação do método de representação no espaço vetorial. A Seção 3.5.3 discute sobre o método redução de dimensionalidade. Por fim, a Seção 3.5.4 trata das categorias de experimentos realizados.

A Figura 11 apresenta um fluxograma que descreve todas as etapas pelas quais as notícias passam, desde o processo de pré-processamento até a conclusão dos agrupamentos.

Figura 11 – Etapas de análise das notícias



Fonte: Elaborado pelo Autor, 2023.

3.5.1 Pré-processamento

Na etapa de pré-processamento, as notícias de futebol foram submetidas a um conjunto de técnicas de pré-processamento, com o objetivo de preparar os dados para a etapa de agrupamento. O código foi implementado em Python, e a biblioteca de código aberto Natural Language Toolkit (NTLK) foi usada para auxiliar no processamento de linguagem natural. Essa biblioteca fornece um conjunto de ferramentas para realizar tarefas relacionadas ao processamento de texto e análise de linguagem.

A *tokenização* foi realizada utilizando-se a classe RegexpTokenizer, da biblioteca NLTK. O código de *tokenização* recebeu o documento de notícia como entrada e retornou

uma lista de *tokens*. Esses *tokens* foram obtidos com base em expressões regulares, visando separar o texto em unidades, como palavras e pontuações. Em seguida, os *tokens* foram submetidos a um processo de remoção de palavras de parada. Para isso, foi utilizada a lista de palavras de parada disponibilizada pela biblioteca NLTK. Normalmente, essas palavras não possuem valor semântico significativo para a análise, e, portanto, são removidas para evitar ruído nos resultados.

Após a remoção de palavras de parada, os *tokens* passaram pelo processo de derivação. A classe RSLPStemmer, da biblioteca NLTK, foi usada para realizar essa tarefa. O processo de derivação tem como objetivo converter cada *token* para seu radical, que é o elemento básico e significativo das palavras do ponto de vista gramatical.

Para melhorar ainda mais o pré-processamento, foi realizada uma etapa de remoção de acentos e caracteres especiais. O objetivo foi eliminar sinais de pontuação e outros símbolos não relevantes para a análise.

Após a aplicação dessas etapas de pré-processamento, os dados foram preparados para a próxima etapa da metodologia, que consiste na representação dos documentos no espaço vetorial.

3.5.2 Representação no espaço vetorial

Para representar os documentos no espaço vetorial, foi utilizada a técnica *Bag-of-Words*, a qual consiste em criar um vocabulário único a partir de todas as palavras presentes nos documentos. Dessa forma, foi criada uma matriz onde cada documento é representado por uma linha, e cada termo, por uma coluna. O valor em cada posição representa a frequência das palavras no documento. Essa técnica ignora a ordem das palavras e trata cada documento como um conjunto não estruturado de palavras. Além disso, a matriz resultante com o uso dessa técnica é bastante esparsa, devido à variedade do vocabulário encontrado nos textos.

Para representar a frequência de cada termo nos documentos, foi utilizada a técnica TF-IDF, implementada através da classe TfidfVectorizer, da biblioteca scikit-learn. O TF-IDF calcula um valor ponderado para cada palavra, em cada documento, levando em conta a frequência da palavra no documento e a raridade da palavra em todo o conjunto de documentos. Isso ajuda a identificar palavras-chave mais relevantes para cada documento.

3.5.3 Redução de dimensionalidade

A redução de dimensionalidade permite lidar com a alta complexidade dos vetores TF-IDF resultantes da etapa anterior. Para isso, foi utilizada a Análise Semântica Latente juntamente com a Decomposição em Valores Singulares. O LSA busca capturar a semântica implícita nos documentos, enquanto o SVD decompõe uma matriz em componentes principais.

Para realizar o processo de decomposição, foi utilizada a classe TruncatedSVD, do módulo de decomposição do scikit-learn. O parâmetro *n components* indica o número de componentes principais que serão mantidos na matriz resultante. Em seguida, a função aplica a normalização aos dados utilizando a classe *Normalizer*, também disponibilizada pelo scikit-learn.

3.5.4 Categorias de experimentos

Para a execução do presente trabalho, foram realizados experimentos com a base de dados de notícias e as técnicas de agrupamento previamente discutidas. Os experimentos foram divididos em duas categorias, sendo a primeira com o número de *clusters* predeterminado, e a segunda, onde o número de *clusters* é definido com base na melhor execução de cada algoritmo.

Na primeira categoria de experimentos, o objetivo foi agrupar as notícias pelo nome do time. Esse rótulo é uma informação que já estava presente na base de dados e pôde ser usada após o agrupamento para se calcular a taxa de acerto de cada técnica de agrupamento. Foi usada a métrica de acurácia, que é a proporção entre o número de notícias categorizadas corretamente e o total de previsões realizadas.

Na segunda categoria de experimentos, o objetivo foi agrupar os dados sem a necessidade de especificar previamente o número de *clusters*. Para isso, foi adotada uma técnica chamada de *Grid Search*, ou Busca em Grade, em português. Essa técnica procura encontrar o melhor resultado de agrupamento com base em uma métrica escolhida. A Busca em Grade consiste em explorar diferentes combinações de hiperparâmetros de um algoritmo, visando identificar qual combinação produz os melhores resultados.

Os hiperparâmetros são valores definidos antes da execução do modelo, podendo ser números inteiros, números reais, valores booleanos ou strings. A metodologia da Busca em Grade é realizar uma busca completa em um subconjunto específico do espaço de hiperparâmetros do modelo. Como esse espaço pode incluir valores reais ou ilimitados para alguns hiperparâmetros, pode ser necessário especificar um limite para se aplicar a Busca em Grade (LIASHCHYNSKYI; LIASHCHYNSKYI, 2019).

4 RESULTADOS E DISCUSSÕES

Este capítulo apresenta os resultados da análise de técnicas de agrupamento de dados aplicadas a notícias de futebol. A Seção 4.1 apresenta a descrição dos dados pré-processados. A Seção 4.2 trata dos resultados da etapa de redução de dimensionalidade. A Seção 4.3 discute sobre os experimentos realizados com base nas medidas de validade externa. A Seção 4.4 explora os experimentos que foram conduzidos com base nas medidas de validade interna. Por fim, a Seção 4.4.5 trata dos resumos dos experimentos.

4.1 Descrição dos dados pré-processados

Para os experimentos realizados, foi selecionado um subgrupo de 32042 notícias da base de dados. Essa seleção englobou todas as notícias referentes ao ano de 2022, que representa o ano mais recente na base de dados que contém notícias abrangendo todo o ano. Inicialmente, o comprimento médio dos documentos foi de 454 palavras. Após a *tokenização*, essa média caiu para 169 *tokens*. Esse resultado mostra que a etapa de pré-processamento elimina muito ruído e termos que não são relevantes para o processo de análise de texto.

Foram realizadas algumas etapas de pré-processamento nos dados. A primeira delas foi a *tokenização*, que dividiu todos os documentos por espaços e sinais de pontuação. Foi determinado que os *tokens* deveriam ter ao menos dois caracteres para serem válidos, e, após isso, eles passaram pela remoção de palavras de parada, que eliminou artigos e preposições da lista. Além disso, os *tokens* passaram por uma função que removeu termos que não eram considerados palavras válidas no português brasileiro. Isso foi feito com a ajuda de uma lista de palavras válidas disponibilizada pela biblioteca SpaCy. Esses *tokens* eram resultado da extração de notícias feita pelo autor da base no *site* de notícias do Globo Esporte.

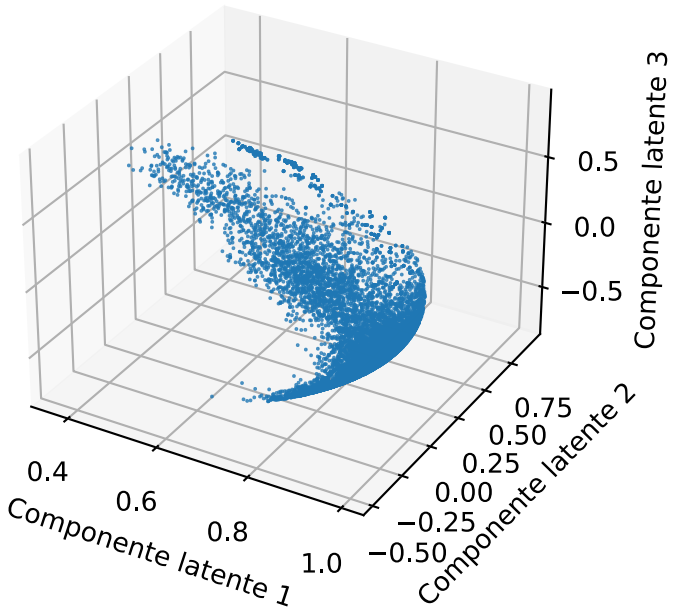
A etapa de pré-processamento e representação no espaço vetorial com TF-IDF resultou em uma matriz de 32042 linhas por 45035 colunas, indicando que os documentos selecionados possuíam 45035 termos únicos em todos eles. Textos possuem a característica da alta dimensionalidade, e algoritmos de agrupamento não funcionam muito bem com uma grande quantidade de dimensões. Por esse motivo, os dados foram submetidos à etapa de redução de dimensionalidade.

4.2 Redução de dimensionalidade

Após as etapas de pré-processamento e representação no espaço vetorial, também foi realizada a etapa de redução de dimensionalidade utilizando-se a técnica LSA. Essa etapa permitiu reduzir a quantidade de atributos da matriz termo-documento, mantendo as informações mais relevantes dos dados.

Para facilitar a visualização, efetuou-se a redução de dimensionalidade com um número de componentes igual a três, o que permitiu plotar os pontos em um espaço tridimensional. A Figura 12 mostra o gráfico de dispersão da matriz reduzida gerada pela decomposição LSA para três componentes, em que cada ponto representa um documento. Foi realizada uma amostragem aleatória, representando 30% dos pontos originais, com o objetivo de melhorar ainda mais a visualização da imagem.

Figura 12 – Gráfico de dispersão da matriz LSA



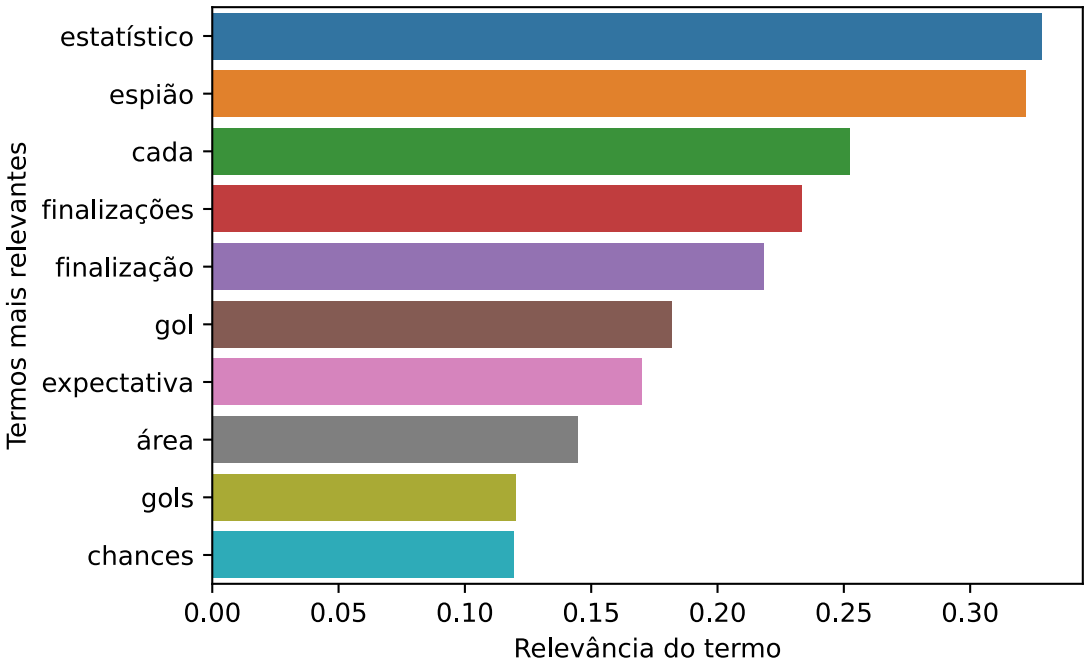
Fonte: Elaborado pelo Autor, 2023.

É importante destacar que a redução para apenas três componentes pode resultar na perda de informações. Isso ocorre porque o modelo perde detalhes da complexidade e a variação do conjunto de dados original. No entanto, foi uma escolha importante para melhorar a compreensão de como os pontos se comportam no espaço. Ao observar o gráfico, fica evidente que a densidade dos pontos é significativa em certas regiões do espaço, o que sugere uma alta sobreposição entre os documentos. Essa sobreposição representa um desafio para os algoritmos de agrupamento, uma vez que torna a diferenciação entre grupos mais complexa.

A técnica de redução de dimensionalidade com LSA resulta na geração de tópicos que condensam e simplificam a representação dos dados originais. Cada tópico é criado considerando como as palavras nos documentos se relacionam. A importância de cada palavra em um tópico depende de quanto ela contribui para o entendimento desse tópico. A Figura 13 mostra os termos que mais contribuíram para a formação do tópico 1. A análise de quais termos contribuem para a formação de cada tópico latente pode ser interessante para entender melhor do que se trata um tópico específico. Isso pode ser útil para se obter *insights* relevantes e aprimorar o modelo.

Por meio da matriz de valores singulares, o LSA também fornece informações sobre a importância relativa de cada tópico na representação global dos dados. A Figura 14

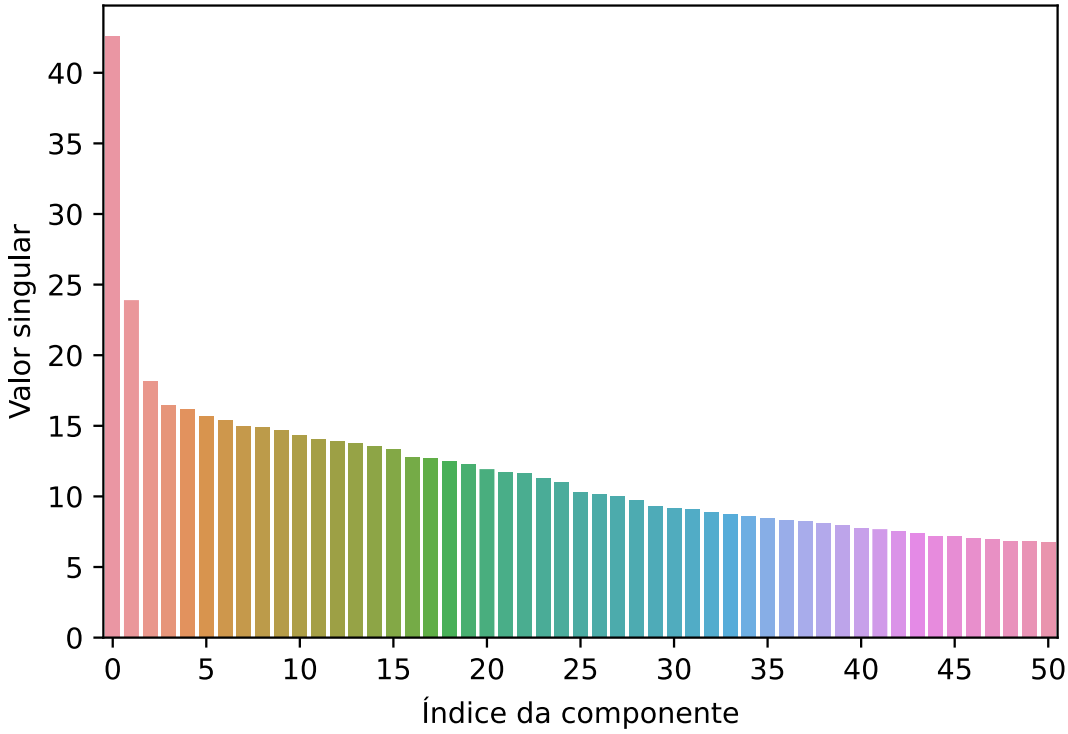
Figura 13 – Conceito latente 1



Fonte: Elaborado pelo Autor, 2023.

apresenta um gráfico dos valores singulares, os quais são dispostos em ordem decrescente, com os maiores valores no topo. Isso implica que, quanto maiores forem os valores singulares, melhor esses tópicos capturam a estrutura predominante nos dados.

Figura 14 – Gráfico de valores singulares



Fonte: Elaborado pelo Autor, 2023.

Determinar o melhor número de componentes para o método de Análise Se-

mântica Latente diretamente não é uma tarefa simples. Uma estratégia, então, é testar o resultado do algoritmo de mineração subsequente para uma gama de números de componentes. Dessa forma, foi realizada uma busca em grade com o número de componentes variando entre 20 e 100 para cada algoritmo de agrupamento. Essa faixa de valores foi escolhida porque a literatura sugere que o número de componentes pode variar de algumas unidades até algumas centenas, dependendo da natureza do problema e da quantidade de dados disponíveis. Portanto, com esses valores, já foi possível observar o comportamento da acurácia e determinar o melhor número de componentes. A Tabela 3 apresenta os resultados do experimento para cada algoritmo.

Tabela 3 – Melhor número de componentes para cada algoritmo

Algoritmo	Número de componentes	Percentual de acurácia
K-Means	32	75
Hierárquico	30	72
GMM	29	74

Fonte: Elaborado pelo Autor, 2023.

Os números de componentes para os algoritmos de agrupamento ficaram bem próximos entre si, variando de 29 a 32. Essa consistência sugere que a faixa 29 a 32 é uma escolha apropriada para a dimensionalidade dos dados após a redução. Uma vez definido esse valor, foi possível seguir em frente com os testes dos algoritmos de agrupamento.

4.3 Agrupamento usando medida de validade externa

Na categoria de experimentos que visava agrupar as notícia pelo nome do time, foi usado o valor 21 como o hiperparâmetro que controla o número de *clusters* gerados pelo algoritmo. É importante destacar que 21 é o número de times que compõem a base de dados usada neste trabalho. Dentre os algoritmos de agrupamento discutidos, o único que não possui um hiperparâmetro para escolher o número de agrupamentos a serem formados é o DBSCAN. Isso ocorre devido à sua estratégia, que define o número de grupos durante a execução com base na densidade dos pontos. Portanto, o DBSCAN não foi usado para esse conjunto de experimentos.

Todos os algoritmos utilizados atribuem números como rótulos para os grupos de dados. Esses números são inteiros, variando de 0 a 20, o que significa que existem 21 grupos diferentes. Para associar os rótulos previstos com os rótulos reais, foi necessário seguir algumas etapas. Inicialmente, a lista de rótulos previstos é atribuída a uma nova coluna da estrutura original da base de dados. As notícias são, então, agrupadas com base nesses rótulos previstos, onde cada *cluster* é composto por notícias que o algoritmo acredita compartilhar características semelhantes. Para cada *cluster* de notícias, determina-se o rótulo real (nome do clube de futebol) mais frequente entre as notícias que o compõem. Logo em seguida, cada rótulo previsto de *cluster* é associado ao rótulo real mais frequente

nesse *cluster*. Por fim, o percentual da taxa de sucesso é calculado como a proporção de notícias em que os rótulos previstos coincidem com os rótulos reais.

Foram realizados diversos experimentos, adicionando ou removendo funções na etapa de pré-processamento. Primeiramente, os algoritmos de agrupamento foram executados usando-se apenas os documentos após a *tokenização* e a remoção das palavras de parada. Em uma segunda variação do experimento, a função de derivação foi introduzida aos *tokens*, resultando na conversão das palavras para seus radicais. Por fim, a terceira variação do experimento substituiu a derivação pela lematização, que, por sua vez, transforma as palavras em seus respectivos lemas. Essas diferentes abordagens foram avaliadas para determinar seu impacto no desempenho dos algoritmos de agrupamento. Os resultados são exibidos na Tabela 4.

Tabela 4 – Percentual de acurácia para cada experimento

Experimentos	K-Means	Hierárquico	GMM
Usando apenas *tokenização*	75	72	74
Usando *tokenização* e derivação	75	71	74
Usando *tokenização* e lematização	74	69	68

Fonte: Elaborado pelo Autor, 2023.

Nos resultados dos experimentos, observou-se que tanto o experimento que usou a *tokenização* quanto aquele que combinou a *tokenização* com a derivação produziram os melhores resultados. A acurácia obtida em ambos os casos foi semelhante, indicando que a simplificação resultante da derivação não apresentou melhorias significativas. Já o uso da *tokenização* em conjunto com a lematização resultou nos valores mais baixos de acurácia. Portanto, a redução das palavras às suas formas básicas pode resultar na perda de informações importantes contidas nas formas originais, e isso pode causar dificuldade de distinção entre os pontos por parte dos algoritmos de agrupamento. A escolha da técnica de processamento de texto melhorou a qualidade dos agrupamentos, e a *tokenização* se destacou como a abordagem mais eficaz para este conjunto de dados.

Durante as etapas de se relacionar os rótulos previstos aos nomes dos times, foi observado um padrão que limitou a acurácia do agrupamento. Foram identificadas muitas notícias de mesmo rótulo previsto associado a dois times. Uma análise mais profunda mostrou que esse padrão era mais frequente em times do mesmo Estado, como Flamengo e Fluminense, Coritiba e Athletico Paranaense, e Fortaleza e Ceará. Esse comportamento está ligado ao fato de que as notícias sobre times do mesmo Estado, frequentemente, apresentam características compartilhadas. Isso ocorre devido a confrontos frequentes entre esses times, resultando em cobertura jornalística para ambos. Além disso, é comum que compartilhem termos em comum, como nome de estádio, cidades e até mesmo jogadores, criando um desafio adicional na tarefa de agrupamento.

4.4 Agrupamento usando medida de validade interna

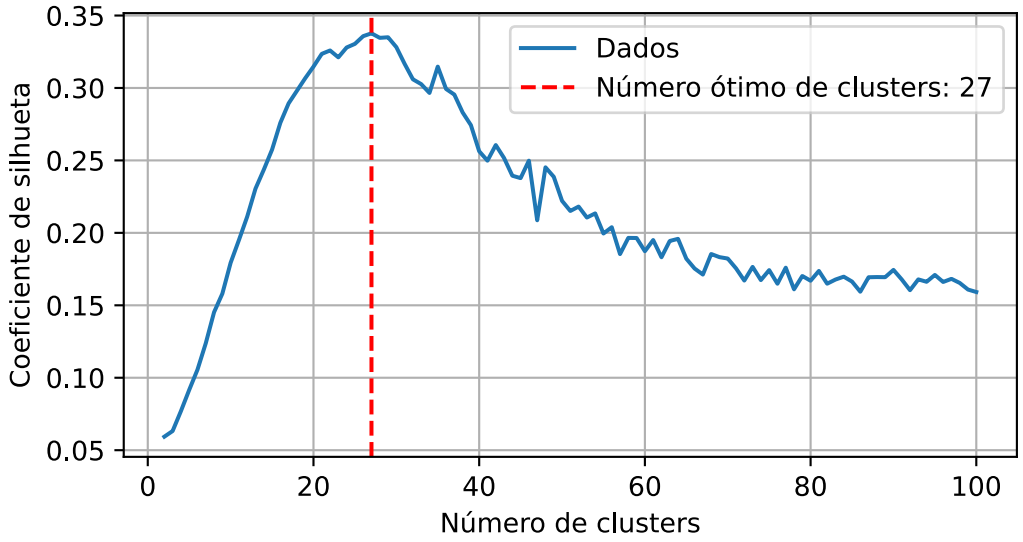
O processo de otimização dos parâmetros para cada algoritmo de agrupamento foi realizado através da execução de uma busca em grade. A cada iteração dessa busca, foi calculado o coeficiente de silhueta como métrica de avaliação, o que possibilitou uma análise dos resultados e a identificação do conjunto de hiperparâmetros que proporciona o melhor desempenho no agrupamento.

4.4.1 K-Means

No caso do algoritmo de agrupamento K-Means, o parâmetro mais importante é o número de *clusters* (*n_clusters*), que define a quantidade de grupos em que os dados serão divididos. Foi explorado um intervalo para esse parâmetro, que variou de 2 até 100. Em cada iteração da busca, foi calculado o coeficiente de silhueta, métrica usada para avaliar a coerência do agrupamento.

Por meio da busca em grade, foi identificado que o número ideal de *clusters* para o conjunto de dados foi 27. O coeficiente de silhueta associado a essa configuração foi 0.337, indicando uma coerência satisfatória para o agrupamento, visto que essa medida varia entre -1 e 1. A Figura 15 apresenta o gráfico da relação entre o número de *clusters* e o coeficiente de silhueta calculado. A análise da curva gerada pela busca em grade revela a capacidade do algoritmo K-Means em capturar a estrutura dos dados de maneira correta. Isso é evidenciado pelo formato da curva, que, inicialmente, exibe uma crescente até atingir o valor máximo, seguida por uma queda em direção a valores baixos de coeficiente de silhueta.

Figura 15 – Busca em grade para o K-Means



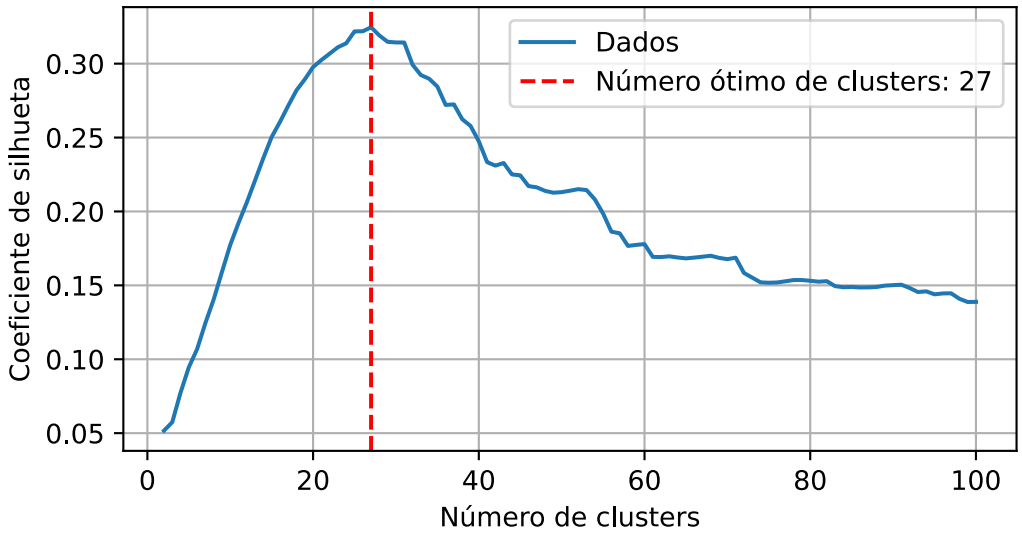
Fonte: Elaborado pelo Autor, 2023.

4.4.2 *Agrupamento Hierárquico Aglomerativo*

No caso do Agrupamento Hierárquico Aglomerativo, também foi realizada a busca em grade para otimização dos parâmetros. Além do número de *clusters* (n_clusters), foram explorados os diferentes critérios de ligação para o agrupamento (*linkage*). Esse parâmetro define qual será o método utilizado para se calcular a distância entre *clusters* durante o processo de aglomeração. Para execução da busca em grade, o número de *clusters* também foi variado de 2 até 100, e os métodos de ligação testados foram *ward*, *complete*, *average* e *single*.

Como resultado da otimização para o Agrupamento Hierárquico Aglomerativo, foi identificado o valor 27 para o número de *clusters*, e o melhor critério de ligação foi o *ward*. O coeficiente de silhueta associado foi 0.324, também indicando uma boa coerência para o agrupamento. A Figura 16 apresenta o gráfico da relação entre o número de *clusters* e o coeficiente de silhueta calculado. A curva resultante da busca em grade destaca a precisão do algoritmo de agrupamento hierárquico em capturar a estrutura dos dados. Este fato é observado no padrão da curva, que apresenta uma fase inicial de ascensão até alcançar o valor máximo, seguida por uma descida em direção a coeficientes de silhueta reduzidos.

Figura 16 – Busca em grade para o Agrupamento Hierárquico Aglomerativo



Fonte: Elaborado pelo Autor, 2023.

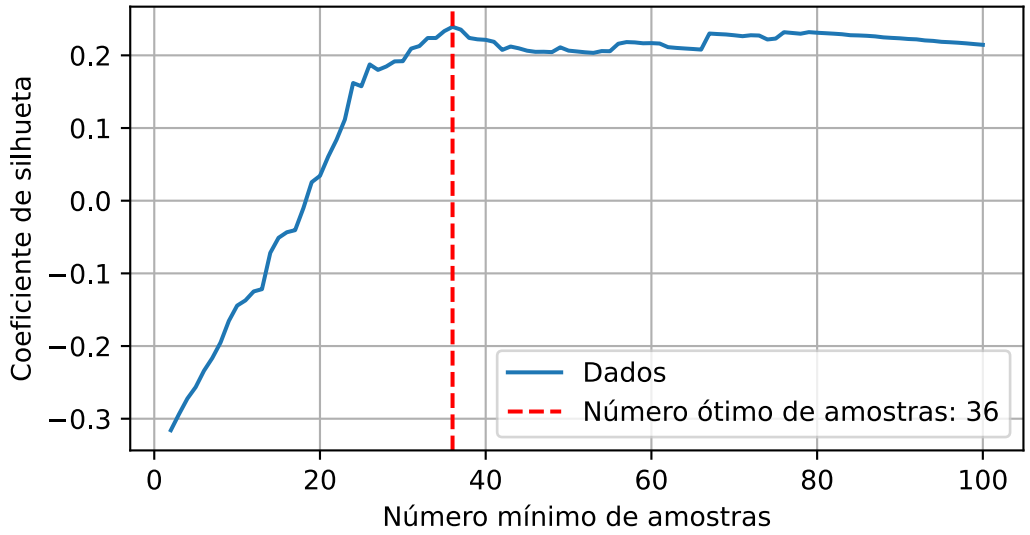
4.4.3 *DBSCAN*

A busca em grade também foi realizada no algoritmo DBSCAN para análise dos parâmetros que resultam no melhor coeficiente de silhueta. O primeiro parâmetro analisado foi distância máxima (*eps*) entre duas amostras, para que uma seja considerada vizinha da outra. O intervalo analisado foi entre 0.1 e 2.0, com os valores sendo incrementados em 0.1. O segundo parâmetro foi o número de amostras (*min_samples*) em uma vizinhança para

que um ponto seja considerado como central. O intervalo analisado para *min_samples* foi entre 2 e 100.

O melhor índice de silhueta encontrado na busca em grade para o DBSCAN foi 0.239, sendo o *eps* igual a 0.4, e *min_samples* igual a 36. Com essa configuração, o modelo identificou 32 grupos no conjunto de dados. A Figura 17 apresenta o gráfico da relação entre o valor de *eps* e o coeficiente de silhueta calculado. A curva gerada pela busca em grade realizada para o DBSCAN mostra que o algoritmo não conseguiu identificar, de forma clara, a estrutura presente nos dados. É possível observar isso através do formato da curva, em que os valores do coeficiente de silhueta aumentam até um ponto máximo, mas continuam em valores altos até o último ponto analisado na busca.

Figura 17 – Busca em grade para o DBSCAN



Fonte: Elaborado pelo Autor, 2023.

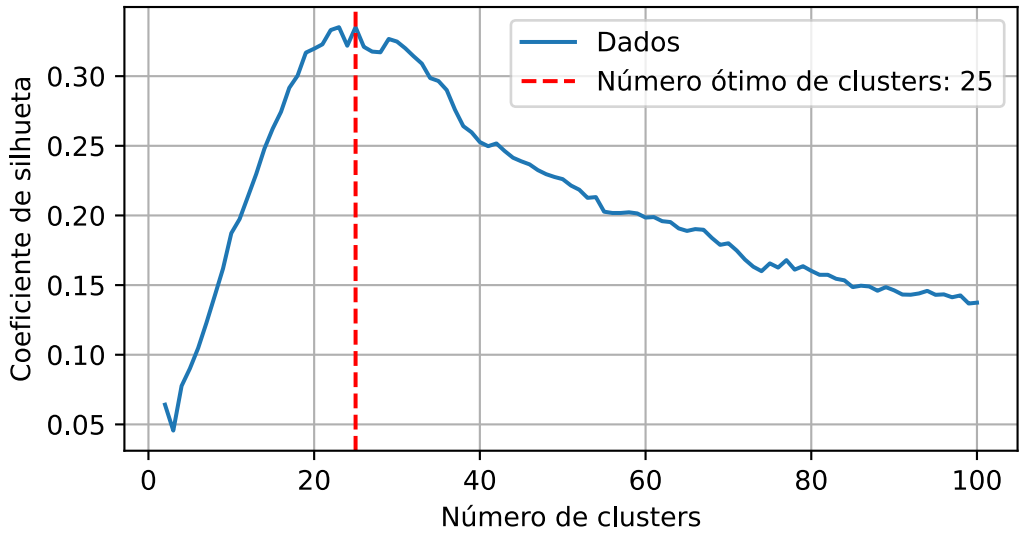
4.4.4 Modelo de Mistura Gaussiana

Para o agrupamento pelo Modelo de Mistura Gaussiana, foi feita a busca em grade variando os principais parâmetros. O primeiro parâmetro variado foi número de componentes (*n_components*), que representam quantos grupos serão formados. O intervalo escolhido também foi de 2 até 100 componentes. O segundo parâmetro foi o tipo de covariância (*covariance_type*), que se refere à maneira como as matrizes de covariância são modeladas para cada componente da mistura. Os valores possíveis para esse parâmetro são *full*, *tied*, *diag* e *spherical*.

Os resultados da busca em grade encontrados para o agrupamento pelo Modelo de Mistura Gaussiana foi 25, para o número de componentes, e 0.335, para o coeficiente de silhueta. A Figura 18 apresenta o gráfico da relação entre o número de componentes e o coeficiente de silhueta calculado. A curva gerada a partir da busca em grade evidencia que o GMM conseguiu aprender de forma adequada a estrutura dos dados. Isso é perceptível

porque os valores de acurácia aumentam até um ponto máximo e, depois, tendem a valores mais baixos de coeficiente de silhueta.

Figura 18 – Busca em grade para o GMM



Fonte: Elaborado pelo Autor, 2023.

4.4.5 Resumo dos experimentos

A Tabela 5 apresenta os resultados da busca em grade realizada para identificar o número mais adequado de *clusters* para cada técnica. Os quatro algoritmos avaliados foram o K-Means, o Hierárquico, o DBSCAN e o GMM. Os resultados incluem o número de agrupamentos identificados por cada algoritmo e seus respectivos coeficientes de silhueta.

Tabela 5 – Resultados da busca em grade

Algoritmo	Número de *clusters*	Coeficiente de Silhueta
K-Means	27	0.337
Hierárquico	27	0.324
DBSCAN	32	0.239
GMM	25	0.335

Fonte: Elaborado pelo Autor, 2023.

As técnicas empregadas pelos algoritmos K-Means, agrupamento hierárquico e GMM revelaram resultados bastante próximos, tanto em relação ao número de *clusters* quanto ao coeficiente de silhueta. Isso sugere que a segmentação das notícias em 25 a 27 grupos pode ser capaz de capturar categorias que expliquem a estrutura da base de dados de forma adequada.

O coeficiente de silhueta médio para esses três algoritmos foi de 0.332, indicando uma divisão eficiente dos pontos em grupos. No entanto, é possível observar que há certa sobreposição entre os grupos, dado que o valor 0 sugere agrupamentos totalmente sobrepostos, enquanto 1 indica um agrupamento perfeito.

Já o algoritmo DBSCAN dividiu o conjunto em 32 *clusters*, apresentando um coeficiente de silhueta um pouco menor, sendo igual a 0.239. Este resultado sugere uma divisão um pouco menos clara em comparação aos outros algoritmos. Essa discrepância pode ser atribuída à maneira como o algoritmo lida com pontos sobrepostos durante a análise.

5 CONSIDERAÇÕES FINAIS

Neste estudo, foram analisadas algumas das principais técnicas de agrupamento de dados aplicadas às notícias de times que participam do Campeonato Brasileiro de Futebol. A intenção foi compreender os padrões e estruturas implícitas nos dados. O principal objetivo foi avaliar o desempenho de diferentes algoritmos de agrupamento, incluindo o K-Means, o agrupamento hierárquico, o DBSCAN e o modelo de mistura gaussiana. Os dados, provenientes de uma base contendo notícias do *site* do Globo Esporte, foram submetidos a várias etapas de pré-processamento antes de serem submetidos ao agrupamento.

A fase de pré-processamento iniciou com a *tokenização* e seguiu com a remoção de palavras de parada, a derivação e outras etapas importantes para preparar as notícias. Além disso, foram utilizadas a representação de saco de palavras e a ponderação TF-IDF para capturar a frequência relativa dos termos. Posteriormente, foi empregada a técnica de redução de dimensionalidade com LSA para reduzir a complexidade dos dados e determinar o número ideal de componentes latentes.

Ao realizar o agrupamento das notícias, foram estabelecidos 21 *clusters*, correspondentes à quantidade de times participantes no campeonato. Os resultados obtidos demonstraram que o K-Means e o modelo de mistura gaussiana alcançaram o índice mais alto de acurácia, atingindo 75%. Já o agrupamento hierárquico obteve uma acurácia de 70%. O algoritmo DBSCAN não foi usado para esse experimento em específico, por não permitir que fosse definido o número de grupos desejados.

Além das análises com número fixo de *clusters*, foram exploradas diferentes configurações de agrupamento, permitindo que o número de *clusters* fosse determinado com base no coeficiente de silhueta mais adequado. Essa abordagem revelou variações entre 25 e 32 grupos, sugerindo que essa faixa de valores pode ser adequada para segmentar a base de dados.

Os resultados obtidos neste trabalho contribuem para uma compreensão mais profunda da aplicação de técnicas de agrupamento em notícias esportivas, capturando suas particularidades e fornecendo *insights* relevantes. Vale ressaltar, também, a significativa importância do processo de representação textual no espaço vetorial e de toda a etapa de pré-processamento na análise.

A transformação de documentos de texto em representações numéricas permitiu que os algoritmos de mineração de dados atuassem de maneira mais eficaz, identificando padrões e estruturas nos dados. Este processo requer um refinamento cuidadoso, uma vez que exerce grande impacto no resultado final do agrupamento.

Vale ressaltar que a etapa de pré-processamento, como discutido anteriormente, é uma tarefa relevante, que apresenta muitas dificuldades. O refinamento dos dados textuais exige a aplicação criteriosa das técnicas de pré-processamento, como *tokenização* e limpeza das informações. A complexidade dos dados textuais é inerente à sua natureza e

ressalta a importância de se investir tempo e esforço na preparação deles antes da aplicação de técnicas de mineração.

A Seção 5.1 discute sobre os trabalhos futuros.

5.1 Trabalhos futuros

Considerando possíveis melhorias para pesquisas futuras, pode ser interessante ampliar o período de tempo selecionado para a análise das notícias. Isso poderia oferecer uma entendimento mais abrangente sobre as tendências ao longo das temporadas do campeonato de futebol. Assim, seria possível compreender, de forma mais profunda, o comportamento do agrupamento aplicado nesse contexto.

Outro ponto relevante é a aplicação de uma abordagem dinâmica na análise de notícias, possibilitando o agrupamento de novas informações com base no modelo já existente. Com essa abordagem, seria possível ajustar o modelo à medida que novas notícias são incorporadas, permitindo uma análise em tempo real.

REFERÊNCIAS

AFONSO, A. R.; DUQUE, C. G. Automated text clustering of newspaper and scientific texts in brazilian portuguese: analysis and comparison of methods. **Journal of Information Systems and Technology Management**, FEA USP, São Paulo, Brasil, v. 11, n. 2, p. 415–436, 2014. Disponível em: https://doi.org/10.4301/S1807-17752014000200011. Acesso em: 18 dez. 2023.

AGGARWAL, C. C. **Machine Learning for Text**: An Introduction. Cham: Springer International Publishing, 2018.

ALLAHYARI, M. *et al.* A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques. In. Disponível em: https://doi.org/10.48550/arXiv.1707.02919. Acesso em: 18 dez. 2023.

BUITINCK, L. *et al.* API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning. 2013. p. 108–122.

DHAR, A. *et al.* Text categorization: past and present. **Artificial Intelligence Review**, v. 54, p. 3007–3054, 2020. Disponível em: https://doi.org/10.1007/s10462-020-09919-1. Acesso em: 18 dez. 2023.

EVANGELOPOULOS, N.; ZHANG, X.; PRYBUTOK, V. R. Latent Semantic Analysis: five methodological recommendations. **European Journal of Information Systems**, v. 21, n. 1, p. 70–86, 2012. Disponível em: https://doi.org/10.1057/ejis.2010.61. Acesso em: 18 dez. 2023.

FREITAS, N.; MOURA, C.; SILVA, M. Sistema multiagente para mineração de imagens de satélite. **XVII Simpósio Brasileiro de Sensoriamento Remoto**, p. 7351–7358, 2015.

GERHARDT, T. E.; SILVEIRA, D. T. **Métodos de Pesquisa**. Porto Alegre: Plageder, 2009.

GHOSAL, A. *et al.* A Short Review on Different Clustering Techniques and Their Applications. Springer Singapore, Singapore, p. 69–83, 2020. Disponível em: https://doi.org/10.1007/978-981-13-7403-6_9. Acesso em: 18 dez. 2023.

GOLDSCHMIDT, R.; PASSOS, E.; BEZERRA, E. **Data mining**: conceitos, técnicas, algoritmos, orientações e aplicações. Campus, Rio de Janeiro, Brasil, 2015.

JO, T. **Machine Learning for Text**: Concepts, Implementation, and Big Data Challenge. Cham: Springer International Publishing, 2019.

KAUSHIK, S. **Clustering Introduction, Different Methods and Applications**. 2016. Disponível em: https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering. Acesso em: 8 abr. 2023.

LANE, H.; HOWARD, C.; HAPKE, H. **Natural Language Processing in Action**. Manning Publications, 2019.

LIASHCHYNSKYI, P.; LIASHCHYNSKYI, P. Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS, 2019. Disponível em: https://doi.org/10.48550/arXiv.1912.06059. Acesso em: 18 dez. 2023.

LIKAS, A.; VLASSIS, N.; VERBEEK, J. J. The global k-means clustering algorithm. **Pattern Recognition**, v. 36, n. 2, p. 451–461, 2003. Biometrics. Disponível em: https://doi.org/10.1016/S0031-3203(02)00060-2. Acesso em: 18 dez. 2023.

MAGALHÃES, L. H. de. **Agrupamento automático de notícias de jornais on-line usando técnicas de machine learning para clustering de textos no idioma português**. 2020. Tese (Doutorado) – Universidade Federal de Minas Gerais, Belo Horizonte.

MARUTHO, D. *et al.* The Determination of Cluster Number at k-Mean Using Elbow Method and Purity Evaluation on Headline News. In: p. 533–538. Disponível em: https://doi.org/10.1109/ISEMANTIC.2018.8549751. Acesso em: 18 dez. 2023.

MONEDA, L. Globo Esporte News Dataset. Version 11, 2023. Disponível em: https://www.kaggle.com/lgmoneda/ge-soccer-clubs-news. Acesso em: 1 abr. 2023.

MURTAGH, F.; CONTRERAS, P. Algorithms for hierarchical clustering: an overview. **WIREs Data Mining and Knowledge Discovery**, v. 2, n. 1, p. 86–97, 2011. Disponível em: https://doi.org/10.1002/widm.53. Acesso em: 18 dez. 2023.

RAI, P.; SINGH, S. A Survey of Clustering Techniques. **International Journal of Computer Applications**, v. 7, n. 12, p. 1–5, 2010. Published By Foundation of Computer Science. Disponível em: https://www.ijcaonline.org/archives/volume7/number12/1326-1808. Acesso em: 18 dez. 2023.

RELVAS, C. E. M. **Agrupamento baseado em modelos de mistura de gaussianas com covariáveis**. 2020. Tese (Doutorado) – Universidade de São Paulo, São Paulo.

RIBEIRO, M. R. **Big Data Avançado e Mineração de Dados**. Belo Horizonte: Instituto Federal de Minas Gerais, 2022.

SACHDEVA, S. Scrum methodology. **International Journal Of Engineering And Computer Science**, v. 5, n. 6, p. 16792–16799, 2016. Disponível em: http://www.ijecs.in/index.php/ijecs/article/view/1989. Acesso em: 18 dez. 2023.

SCIENCE, T. D. **Hierarchical clustering explained**. 2023. Disponível em: https://towardsdatascience.com/hierarchical-clustering-explained-e59b13846da8. Acesso em: 23 ago. 2023.

SINGH, H. V.; GIRDHAR, A.; DAHIYA, S. A Literature survey based on DBSCAN algorithms. In: p. 751–758. Disponível em: https://doi.org/10.1109/ICICCS53718.2022.9788440. Acesso em: 18 dez. 2023.

TONG, Y.; GU, L. A News Text Clustering Method Based on Similarity of Text Labels. Springer International Publishing, Cham, p. 496–503, 2019. Disponível em: https://doi.org/10.1007/978-3-030-19086-6_55. Acesso em: 18 dez. 2023.

VACANTI, D.; YERET, Y. Kanban Guide for Scrum Teams, 2021. Disponível em: https://www.scrum.org/resources/kanban-guide-scrum-teams. Acesso em: 23 mai. 2023.

VALDEZ, D.; PICKETT, A. C.; GOODSON, P. Topic Modeling: Latent Semantic Analysis for the Social Sciences. **Social Science Quarterly**, v. 99, n. 5, p. 1665–1679, 2018. Disponível em: https://doi.org/10.1111/ssqu.12528. Acesso em: 18 dez. 2023.

WAZLAWICK, R. S. **Metodologia de Pesquisa para Ciência da Computação**. Rio de Janeiro: Elsevier, 2009.

WIKIPEDIA. **DBSCAN**. 2023a. Disponível em: https://en.wikipedia.org/wiki/DBSCAN. Acesso em: 18 ago. 2023.

_____. **Mixture model**. 2023b. Disponível em: https://en.wikipedia.org/wiki/Mixture_model. Acesso em: 23 ago. 2023.