

**CENTRO UNIVERSITÁRIO SALESIANO DE SÃO PAULO**

**UNIDADE LORENA**

**CURSO DE ENGENHARIA DE COMPUTAÇÃO**

**INTELIGÊNCIA ARTIFICIAL**

**PROFESSOR ME. WALMIR DUQUE**

Breno Ryan de Andrade Fernandes

Daniel Marton Barbosa

Mariana Gonçalves de Freitas Ribeiro

João Vitor Ferreira Azevedo Pereira

**"Agrupamento (Clustering) /** **Inteligência Artificial Generativa e PLN"**

# Sumário

# Introdução ................................................................................................ 3

# 1. Capítulo 1 - Agrupamento (Clustering) no Python ........................... 3

1.1 - Business Understanding .................................................................... 3

1.2 ­­­­­­­- Data Understanding e Data Preparation ............................................ 4

1.3 – Modeling e Evalution ......................................................................... 4

**2. Capítulo 2 – Inteligência Artificial Generativa – Processamento de Linguagem Natural (PLN) ............................................................................................ 8**

2.1 – Definição de texto a ser trabalhado, explicação do seu gênero textual e objetivos de negócio ................................................................................................. 8

2.2 - PLN - Remoção de Ruídos, Homogeneização Stopwords ............... 8

2.3 - PLN - Stemming / Lemmatization ..................................................... 8

2.4 - Chunk / Embedding .......................................................................... 9

**Conclusão ............................................................................................... 10**

**Referências Bibliográficas .................................................................... 10**

# Introdução

O presente trabalho tem como objetivo explorar e aplicar técnicas de ciência de dados e aprendizado de máquina, com ênfase no método de agrupamento, também conhecido como **clustering**. Essa técnica é amplamente utilizada para identificar padrões em dados não rotulados, organizando-os em grupos (ou clusters) baseados em semelhanças entre suas características. Dentro da vasta gama de algoritmos disponíveis para clustering, optamos pelo uso de **clusters baseados em centróides**, devido à sua simplicidade e eficácia para resolver problemas de classificação e segmentação de dados desconhecidos.

No contexto deste estudo, o clustering foi empregado com uma perspectiva prática e otimista, visando categorizar dados complexos em grupos homogêneos, que possam ser posteriormente analisados e interpretados por especialistas. A ideia central é permitir que os dados, organizados por similaridade, sirvam como base para análises mais profundas, facilitando o trabalho de profissionais ao fornecer insights iniciais de forma automatizada e objetiva. Isso torna o processo mais eficiente e reduz a carga de trabalho em etapas manuais.

Além disso, destacamos que a escolha do método de clustering por centróides, como o algoritmo **KMeans**, se deu pela sua robustez em diversos cenários práticos. Ele permite, por exemplo, a adaptação de parâmetros como o número de clusters e a métrica de distância, o que proporciona maior flexibilidade na análise e maior capacidade de atender às especificidades do conjunto de dados estudado.

Ao longo deste trabalho, buscamos não apenas demonstrar a aplicabilidade do clustering, mas também discutir seus desafios, limitações e potenciais melhorias. O objetivo é contribuir para um entendimento mais aprofundado da técnica, além de explorar como ela pode ser utilizada em cenários reais, como a segmentação de espécies biológicas ou outras aplicações relevantes para a ciência e a indústria.

**1. Capítulo 1 - Agrupamento (Clustering) no Python**

1.1 Business Understanding

Nosso projeto teve como objetivo entender e segmentar pinguins com base em suas características físicas (como comprimento e profundidade do bico), a fim de facilitar a análise biológica, como a diferenciação entre espécies ou populações. Nesse sentido, o modelo de escolhido foi o agrupamento (clustering), que permite identificar padrões de grupos de pinguins com características semelhantes e prever a que cluster um novo pinguim pertencerá.

De acordo com as tarefas de Data Science e de maneira breve, visto que alguns pontos serão mais esclarecidos ao avançar no texto temos:

**1. Pré-Processamento de Dados**

* **Normalização/Padronização**:

O código utiliza o StandardScaler para padronizar os dados. Essa tarefa é essencial para modelos baseados em distância, como o KMeans, pois evita que variáveis com escalas diferentes dominem o cálculo.

* **Divisão de dados**:

A divisão entre conjuntos de treinamento e teste (Percentage Split) é um passo importante na validação de modelos, mesmo para clustering não supervisionado.

* **Transformação dos dados**:

Transformação dos dados em uma forma numérica apropriada para análise.

**2. Clustering (Agrupamento)**

* **Técnicas de Clusterização**:

Implementação do algoritmo KMeans com diferentes métodos de inicialização (random, k-means++).

Uso de diferentes métricas de distância: Euclidiana e Manhattan.

* **Análise do Número Ótimo de Clusters**:

O uso do WCSSE e o método do cotovelo para determinar o número ideal de clusters faz parte de um processo exploratório para avaliação de modelos.

**3. Avaliação do Modelo**

* **Métricas de Avaliação Interna**:

WCSSE (Within-Cluster Sum of Squares): Avalia a compactação dos clusters.

Silhouette Score: mede a separação entre os clusters e o quão bem os dados estão agrupados.

* **Validação Cruzada (Cross-Validation)**:

Ainda que o clustering seja uma técnica não supervisionada, o código adaptou a ideia de validação cruzada para avaliar diferentes configurações de clusterização.

**4. Visualização**

* **Método do Cotovelo (Elbow Method)**:

Produz um gráfico visual para ajudar na seleção do número ideal de clusters.

* **Exploração de Dados**:

O código sugere a visualização de características médias dos clusters, o que pode ser feito por meio de gráficos.

**5. Análise Exploratória dos Clusters**

* **Interpretação das Características dos Clusters**:

Cálculo das médias das variáveis em cada cluster para entender o perfil de cada grupo.

* **Perfilagem dos Grupos**:

Identificação de padrões ou perfis dentro dos dados com base nos clusters.

**6. Aplicação em Cenários Reais**

* **Segmentação de Dados**:

Agrupamento de espécies (como no caso dos pinguins).

* **Detecção de Padrões**:

Identificação de padrões ocultos nos dados para apoio à tomada de decisão.

* **Análise de Dados Multivariados**:

O código pode ser considerado apropriado para explorar e agrupar dados com várias dimensões.

**7. Significado dos Clusters**:

A explicação do significado de cada cluster é baseada nas características médias de cada grupo.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

* Cada cluster representa grupos com diferentes características médias em termos de massa, tamanho do bico e comprimento das asas, provavelmente correspondendo a diferenças entre espécies ou subgrupos de pinguins.

**8. Possíveis melhorias**

Com pequenas adaptações, ele também poderia ser usado para:

* **Detecção de Anomalias**.
* **Redução de Dimensionalidade** (como prévia ao clustering)

Exemplo de como os dados estavam antes do nosso trabalho:

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

**Requisito Funcional 1**: Identificar clusters de pinguins com características semelhantes



**Requisito Funcional 2**: Permitir a previsão de cluster para novos pinguins inseridos

Texto

Descrição gerada automaticamente

**Requisito Funcional 3**: Exibir visualizações gráficas dos clusters formados

Medida de Distância Euclidiana

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Visualização

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

Medida de distância Manhattan

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Visualização:

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

**Requisito Funcional 4:** Determinar o número ideal de clusters usando o método Elbow

Texto

Descrição gerada automaticamente

Para a visualização:

Texto

Descrição gerada automaticamente

Visualização:

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

**Requisito Funcional 5**: Criar clusters e avaliar os resultados com base nas características físicas dos pinguins

Texto

Descrição gerada automaticamente

**Requisito Funcional 6**: Uso de distância de Manhattan para formar cluster alternativos

Texto

Descrição gerada automaticamente

Tela preta com letras brancas

Descrição gerada automaticamente

Requisitos Não Funcionais:

- O modelo precisa ser eficiente para grandes volumes de dados, uma vez que esse número tende a crescer.

- Deve ser escalável para poder agregar mais variáveis ou espécies.

1.2 Data Understanding e Data Preparation

Variáveis (Features):

Valores mínimos

Texto

Descrição gerada automaticamente

Valores maxímos

Texto

Descrição gerada automaticamente

Valores médios

Texto

Descrição gerada automaticamente

- culmen\_length\_mmtipo float — Representando o comprimento do bico.

- culmen\_depth\_mmtipo float — Representando a profundidade do bico.

- flipper\_length\_mm tipo float — Representando o comprimento da barbatana.

- body\_mass\_g tipo float — Peso do pinguim.

- Excluímos da análise a feature “sex”, pois, nesse caso, a diferença entre os sexos não influencia o tipo de espécie.

Target e Classes:

- Não há variável-alvo (target) porque o problema é de agrupamento (clustering).

- Os clusters formados (0, 1 e 2) são as "classes" que o algoritmo deve identificar.

Preparação de Dados:

Verificamos como o estava o dataset com esse input:

Texto

Descrição gerada automaticamente

Obtivemos esse output:

Tela de vídeo game

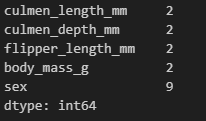
Descrição gerada automaticamente

Como demonstra a imagem existem valores nulos (NaN).

Aplicamos outra verificação:



Output:



Normalizamos utilizando a classe StandardScaller da biblioteca scikit-learn para fazer o tratamento:

Texto

Descrição gerada automaticamente

Nova verificação:



Output:

Texto

Descrição gerada automaticamente

1.3 Modeling e Evaluation

Método do Cotovelo para Determinação de K

O método do cotovelo (Elbow Method) é utilizado para encontrar o número ideal de clusters. Esse método nos revela que, a partir de certo ponto, o aumento de clusters se torna menos significativo do que no início.

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Este gráfico ajuda a identificar o ponto em que a diminuição da WCSS (within-cluster sum of squares) começa a ser mais suave, indicando o número ideal de clusters. Percebe-se que, a partir de 3 clusters, a diferença não diminuiu significativamente, por isso definimos em nosso modelo o K = 3.

Teste de Parâmetros

|  |  |  |
| --- | --- | --- |
| Inicialização | Medida de Distância | Resultado |
| Random | Euclidiana | Texto  Descrição gerada automaticamente |
| Kmeans | Euclidiana | Texto  Descrição gerada automaticamente |
| Random | Manhattan | Texto  Descrição gerada automaticamente |
| Kmeans | Manhattan | Texto  Descrição gerada automaticamente |

Medida de distância Euclidiana (Random e kmeans) :Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Medida de distância Manhattan:

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Tela preta com letras brancas

Descrição gerada automaticamente

Nesse ponto da análise, o que despertou nossa curiosidade foi o fato de que somente com a inicialização Random e a distância Manhattan é que o "Novo pinguim" foi classificado para o cluster 2, sendo esse teste o que teve o melhor índice de WCSS, de ~531.

Tabela de Estratégias de Testes

|  |  |  |  |
| --- | --- | --- | --- |
| Inicialização | Distância | Validação | Resultado |
| Random | Euclidiana | Percentage\_split |  |
| Kmeans | Euclidiana | Percentage\_split |  |
| Random | Manhattan | Percentage\_split |  |
| Kmeans | Manhattan | Percentage\_split |  |
| Random | Euclidiana | Cross\_validation |  |
| Kmeans | Euclidiana | Cross\_validation |  |
| Random | Manhattan | Cross\_validation |  |
| Kmeans | Manhattan | Cross\_validation |  |

**Métrica Silhouette Score**

Utilizamos o Silhouette Score, pois é uma medida que considera tanto a coesão quanto a separação dos clusters. Para isso, foi necessário importar da biblioteca Sklearn. Se, por exemplo, chamarmos coesão de x e separação de y, o Silhouette Score para um único ponto é obtido pela fórmula:

s = (y - x) / (max(x, y))

O valor dessa métrica varia de -1 a 1. Valores próximos de 1 indicam que o ponto está bem ajustado ao seu próprio cluster e mal ajustado aos clusters vizinhos. Se o valor é próximo de zero, temos um ponto que está próximo a um limite de decisão entre dois clusters. Já um valor negativo indica que o ponto pode ter sido atribuído ao cluster errado. Além disso, podemos calcular o Silhouette score para todo o conjunto de pontos.

Testando com um Novo Pinguim (usando um dataset com apenas uma linha)

Texto

Descrição gerada automaticamente

Melhor Modelo ?

Depende.

O WCSSE mede a compactação dos clusters: quanto menor o WCSSE, mais próximos estão os dados dentro de cada um. Um WCSSE menor é, portanto, desejável, mas não deve ser analisado de maneira isolada. Precisamos equilibrá-lo com outras métricas, como o Silhouette Score, que avalia separação e coesão.

Revisão do Modelo Kmeans + Manhattan com validação Cross\_Validation

* Silhouette Score: 0.48807).

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

Se priorizarmos a compactação (WCSSE)

Temos os modelo Random + Manhattan e Kmeans + Euclidean.

Kmeans + Euclidean

* Percentage\_split + WCSSE:47.035

Random + Manhattan

* Cross\_validation + WCSSE:149.81

O melhor modelo depende do objetivo específico:

Se a separação clara entre clusters for mais relevante (por exemplo, para fins interpretativos ou classificatórios), escolha KMeans++ com Euclidean.

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

Se a compactação intra-cluster for mais importante como, minimizar custos ou distância em análises operacionais escolha Random + Manhattan.Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

Em cenários gerais, a separação e a compactação equilibradas fazem do KMeans++ com Euclidean a melhor escolha. No entanto, se o objetivo específico priorizar a compactação extrema, o Random + Manhattan é o vencedor.

Conclusão

Podemos afirmar que nosso trabalho apresentou um bom grau de desafio, mesmo considerando que, em teoria, a técnica de clustering baseada em centróides é uma das abordagens mais acessíveis para implementação e compreensão, especialmente por profissionais de áreas que não sejam diretamente ligadas à ciência de dados ou aprendizado de máquina. Apesar dessa aparente simplicidade, o desenvolvimento deste projeto nos proporcionou uma rica experiência prática e acadêmica, envolvendo desde o tratamento inicial dos dados até a aplicação e avaliação de modelos, passando por diversas etapas críticas que exigiram atenção e ajustes meticulosos.

Durante o processo, enfrentamos diferentes etapas que, embora previsíveis, demandaram um esforço significativo, como a normalização dos dados, a escolha das métricas de avaliação mais adequadas e a adaptação dos algoritmos para melhor desempenho em nossos conjuntos de dados. Além disso, testamos diferentes configurações de parâmetros para identificar aquelas que melhor se adequavam às características do problema abordado, o que nos permitiu compreender mais profundamente os impactos dessas decisões no resultado final.

Embora estejamos satisfeitos com o resultado alcançado, reconhecemos que há melhorias a serem feitas para elevar o desempenho e a qualidade geral da solução. Entre essas possíveis melhorias, podemos destacar o ajuste mais fino de parâmetros, a adoção de técnicas avançadas, como redução de dimensionalidade prévia ao clustering, e a integração de outras métricas de avaliação que complementem as utilizadas. Essas mudanças poderiam ampliar a robustez e a aplicabilidade do modelo, tornando-o ainda mais eficiente e adaptável a novos cenários.

Em síntese, acreditamos que o trabalho foi conduzido de maneira sólida e meticulosa, atendendo aos objetivos propostos e demonstrando a aplicabilidade prática da técnica. Mais do que isso, ele nos proporcionou um aprendizado valioso, reforçando nossa disposição e entusiasmo para continuar explorando e aprimorando técnicas de ciência de dados e aprendizado de máquina no futuro.

Introdução

No trabalho a seguir, buscamos explorar o processamento de linguagem natural (PLN) aplicado ao conjunto de dados “Notícias publicadas no Brasil”, com foco no tema esportes. A escolha desse gênero textual foi motivada por sua acessibilidade e estrutura previsível, facilitando o tratamento e análise do conteúdo. Os objetivos principais incluem a categorização de subtemas, resumos de notícias para facilitar a leitura, identificação de tendências e a personalização de conteúdo para o desenvolvimento de sistemas de recomendação. Além disso, o processo envolveu técnicas robustas de limpeza textual, lematização, tokenização e geração de embeddings, permitindo a criação de uma base de dados vetorial para consulta semântica eficiente. Este documento descreve detalhadamente o fluxo de trabalho, desde o carregamento e filtragem do dataset até a construção de um modelo capaz de identificar similaridades entre textos e fornecer insights relevantes.

# 2. Capítulo 2 - Inteligência Artificial Generativa

## 2.1 - Definição de texto a ser trabalhado, explicação do seu gênero textual e objetivos de negócio

O texto utilizado foi obtido por meio do conjunto de dados “Notícias publicadas no Brasil”, disponibilizado pelo site Kaggle. Esse conjunto foi criado a partir da raspagem de dados (web scraping) em diversos sites, como G1 e SportTV, feita pelo criador do conjunto. Dentro desse conjunto, há alguns temas específicos de notícias (política, esporte, economia e famosos). Optamos por filtrar o tema em "esporte", pois se trata de uma temática mais próxima da grande maioria do público, que não necessita de um repertório mais técnico para lidar com assuntos como "economia", nem causa polêmica com "política" ou textos, geralmente de menor profundidade intelectual, "famosos". Segundo o site Mundo Educação, notícia é um texto jornalístico cuja função é relatar acontecimentos cotidianos, com forte presença de elementos narrativos e descritivos. A notícia é um dos gêneros textuais mais dinâmicos e lidos atualmente. Partindo disso, a escolha do gênero textual se deu justamente por sua estrutura previsível e sua linguagem formal, o que facilitou o processamento (remoção de ruídos). Os objetivos de negócio foram a filtragem e categorização de subtemas para melhor análise, resumo de notícias para facilitar a leitura e obtenção de insights, análise de tendências e identificação de palavras-chave ou temas que estão se destacando nas notícias de esportes ao longo do tempo. Além disso, foi possível personalizar o conteúdo e desenvolver um sistema de recomendação para exibir notícias relevantes aos interesses dos usuários.

### 2.2 - PLN - Tratamento do texto: Remoção de Ruídos, Homogeneização e Stopwords

Como mencionado acima, antes mesmo de iniciarmos o processo de tratamento do texto, precisamos filtrar o conjunto de dados para usar apenas o que, em nosso limitado entendimento, julgamos fundamental para começarmos o trabalho propriamente dito. Com o recurso do framework Pandas, extraímos o CSV e, a partir dele, criamos nosso conjunto de dados filtrado somente com o assunto “esportes”. Para o tratamento de ruídos, foi utilizada a biblioteca nativa do Python Re para lidar com expressões regulares e para formatação de palavras em minúsculas, evitando diferenciação entre palavras como "Tenista" e "tenista" e suas inúmeras variações. Além disso, somado ao processo de tokenização, utilizamos a biblioteca NLTK (Natural Language Toolkit) através do corpus padrão "stopwords", adaptado para a língua portuguesa. Juntamente com a nossa lista de palavras stopwords customizadas, “a, e, i, o, u, né, aí, tá, então”, não tivemos muitos problemas ou necessidades especiais, como gírias, coloquialismostal tal como “né e tu”, pois se trata de um texto jornalístico. Valores numéricos ou caracteres especiais “!,@, $” também foram tratados, assim como outras particularidades utilizadas em nossa linguagem cotidiana sem perceber. Por fim, foram feitos testes para encontrar similaridade entre textos após a integração de modelos de embeddings fornecidos pela biblioteca Sentence Transformers e bancos de dados vetoriais disponibilizados por outra biblioteca chamada ChromaDB. Assim, acreditamos termos produzido um texto com alguma qualidade de homogeneização.

### Carregamento dos dados

Texto

Descrição gerada automaticamente

Teste para conferir as colunas antes da filtragem

Texto

Descrição gerada automaticamente

Criação do novo dataframe somente com os dados relacionados ao tema esportes



Tratamento das stopwords

Texto

Descrição gerada automaticamente

Formatação das palavras em minúsculas, limpeza, lematização e tokenização

Texto

Descrição gerada automaticamente

### Tratamento de valores ausentes

Texto

Descrição gerada automaticamente

### 2.3 - PLN - Stemming / Lemmatization

Quanto à escolha entre Stemming ou Lemmatization, a resposta foi avaliada com certa intuição, uma vez que, ao utilizar o stemming (radicalização), há uma grande redução de palavras, o que resulta em maior desempenho. No entanto, por se tratar de um texto informativo, optamos pela lematização, garantindo uma maior compreensão e integridade da escrita original. Para tal, foi utilizada a biblioteca Spacy com o modelo pré-treinado spacy.load("pt\_core\_news\_sm").

Importação do spacy para lematizar e tokenizar

Texto

Descrição gerada automaticamente

### 2.4 - Chunk / Embedding

Quanto ao limite de chunk, mantivemos a quantidade máxima e o número máximo de sobreposição de caracteres, conforme utilizado em sala de aula, por conta da própria natureza do texto trabalhado. Poderíamos reduzir a redundância apenas reduzindo o overlap ou aumentar a complexidade semântica do texto aumentando o número de chunks. Essa última parte foi mais bem vista em esboços desse trabalho, quando utilizamos texto do tipo romance, com mais figuras de linguagem, metáforas e poesia.

Divisão do texto em chunks

Texto

Descrição gerada automaticamente

# Conclusão

A aplicação de inteligência artificial generativa no contexto de notícias esportivas revelou-se eficaz tanto no processamento quanto na análise de dados textuais. As etapas detalhadas de limpeza, tratamento e geração de embeddings permitiram transformar um grande volume de dados em informações organizadas e relevantes. A escolha por técnicas de lematização e a utilização de ferramentas avançadas como spaCy, Sentence Transformers e ChromaDB garantiram a qualidade dos resultados, permitindo consultas semânticas e a identificação de padrões no conteúdo. Por fim, este trabalho destaca como o uso de IA pode simplificar a análise de textos jornalísticos, oferecendo soluções práticas para problemas reais, como personalização de conteúdo e sistemas de recomendação. As metodologias aqui empregadas podem servir de base para futuras pesquisas ou aplicações em outros contextos e gêneros textuais.

# 6. Referências Bibliográficas

**GLOBO ESPORTE.** Nadal precisa de mais de duas horas para vencer número 74 do mundo. *ge.globo*, 13 jan. 2014. Disponível em: <https://ge.globo.com/tenis/noticia/2014/01/nadal-precisa-de-mais-de-duas-horas-para-vencer-numero-74-do-mundo.html>. Acesso em: 16 nov. 2024.

**DATA GEEKS.** Embeddings. *Data Geeks*, [s.d.]. Disponível em: https://www.datageeks.com.br/embeddings/. Acesso em: 16 nov. 2024.

**ELASTIC.** O que é Word Embedding? *Elastic*, [s.d.]. Disponível em: <https://www.elastic.co/pt/what-is/word-embedding>. Acesso em: 16 nov. 2024.

ANCHIÊTA, Rafael; NETO, Francisco A. R.; MARINHO, Jeziel C.; MOURA, Raimundo. **PLN: das técnicas tradicionais aos modelos de deep learning**. In: **TÍTULO DO LIVRO**. [s.d.]. Capítulo 1. *Sociedade Brasileira de Computação*. Disponível em: <https://books-sol.sbc.org.br/index.php/sbc/catalog/download/79/341/600?inline=1>. Acesso em: 16 nov. 2024.

**BARBOSA, Wellington.** **Processamento de Linguagem Natural (PLN).** *Wellbar - Computação*, 17 jan. 2023. Mini curso de Processamento de Linguagem Natural. Disponível em: <https://www.youtube.com/watch?v=q8NomaZCpjs&list=PLxKmxm_IWVlQCoWxMr-0WoB1gz3o0MhUl> Acesso em: 16 nov. 2024

ESCOLA DNC. **Clusterização: o guia definitivo para análise de dados não supervisionada**. Escola DNC, [s.d.]. Disponível em: <https://www.escoladnc.com.br/blog/clusterizacao-o-guia-definitivo-para-analise-de-dados-nao-supervisionada/>. Acesso em: 18 nov. 2024.

DATA GEEKS. **Clustering**. Data Geeks, [s.d.]. Disponível em: <https://www.datageeks.com.br/clustering/>. Acesso em: 18 nov. 2024.

AWARI. **Aprenda sobre clustering em machine learning**. Awari, [s.d.]. Disponível em: <https://awari.com.br/aprenda-sobre-clustering-em-machine-learning/>. Acesso em: 18 nov. 2024.

MAGALHÃES, Lúcia Helena de; SOUZA, Renato Rocha. **Agrupamento automático de notícias de jornais on-line usando técnicas de machine learning para clustering de textos no idioma português**. *Múltiplos Olhares em Ciência da Informação*, [s.d.]. Disponível em: <https://periodicos.ufmg.br/index.php/moci/article/view/19170/16237>. Acesso em: 18 nov. 2024