Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Departamento de Computação

Trabalho de Aprendizado de Máquina Não Supervisionado

Karys Cristina da Silva Barbosa - 811871
Pedro Marchi Nunes - 801053
Vitor Yuki Inumaru Ferreira - 794041

Julho 2025

Conteúdo

1 Introdução

Este trabalho tem como objetivo utilizar técnicas de aprendizado de máquina aprendidas em aula para analisar um dataset formado por músicas de diversos gêneros presentes no spotify, a partir de um conjunto de dados disponível no Kaggle, que contempla informações diversas sobre cada música, como: dançabilidade, energia, acústica, entre outros.

A relevância do estudo está na análise conjunta de múltiplos fatores já citados acima (acústica, dançablidade, streams, duração em minutos, entre outros) para fins de análise exploratória, mas com grandes possibilidades de aproveitamento comercial. Para isso, será aplicado um processo completo de pré-processamento, tratamento dos dados e visualização das características do dataset, com o intuito de compreender a sua utilização e a aplicação de métodos não supervisionados no processo de agrupamento.

O trabalho possui três etapas principais, sendo estas a justificativa da escolha do dataset, estudo e preparação dos dados, e aplicação dos métodos não supervisionados. O uso dessas técnicas permitirá identificar os principais agrupamento formados a partir de cada métrica usada, auxiliando na interpretação dos resultados e na visualização do conjunto de dados.

Link para acesso ao notebook no Google Colab.

2 Fundamentação Teórica

2.1 Aprendizado de Máquina e Pré-processamento de Dados

O aprendizado de máquina (machine learning) é uma área da inteligência artificial que desenvolve algoritmos capazes de aprender padrões a partir de dados, sem serem explicitamente programados para isso. Entre os tipos de aprendizado, o não supervisionado é amplamente utilizado para problemas de agrupamento (clustering), onde o objetivo é identificar estruturas ocultas nos dados, agrupando itens com características similares em clusters distintos, sem a necessidade de rótulos prévios.

Antes de aplicar algoritmos de aprendizado não supervisionado, é fundamental realizar o pré-processamento dos dados, que inclui a limpeza, tratamento de valores ausentes, normalização e transformação das variáveis. O tratamento adequado dos dados é essencial para garantir a qualidade do modelo, pois dados incompletos, inconsistentes ou outliers mal tratados podem comprometer a eficácia do agrupamento.

A análise exploratória, por meio de técnicas como a análise de distribuição e visualização com histogramas e gráficos de dispersão, ajuda a entender a estrutura dos dados. Como não há rótulos, a ênfase está em identificar padrões, distribuições e possíveis outliers que possam influenciar a formação dos clusters. O uso do histograma permite observar a distribuição das variáveis, enquanto outros modelos de gráficos permitem observar os dados em seu estado natural, antes da aplicação do algoritmo.

Diferentemente do aprendizado supervisionado, que depende de dados rotulados para treinamento, o aprendizado não supervisionado busca descobrir relações intrínsecas nos dados, sem orientação externa. Técnicas como K-means, HDBSCAN* e agrupamento hierárquico são comumente utilizadas para segmentar os dados com base em similaridade, permitindo aplicações como detecção de anomalias, redução de dimensionalidade e exploração de estruturas desconhecidas em conjuntos de dados complexos.

2.2 Bibliotecas Utilizadas

Algumas bibliotecas têm papel fundamental para o desenvolvimento de trabalho de aprendizado de máquina em Python, pois necessita de ferramentas para pré-processamento, modelagem e avaliação de dados. Neste projeto, utilizamos as bibliotecas pandas e numpy para manipulação eficiente dos dados, enquanto o StandardScaler do scikit-learn garantiu a padronização das features. O tratamento adequado dos dados é essencial para garantir a qualidade do modelo, pois dados incompletos ou inconsistentes podem comprometer a eficácia do agrupamento.

A análise exploratória foi realizada combinando técnicas estatísticas com visualizações poderosas. Através do matplotlib, geramos histogramas para entender distribuições e gráficos de dispersão para visualização dos dados preliminarmente. Como não há rótulos, a ênfase estava em descobrir relações intrínsecas e possíveis outliers que poderiam influenciar os resultados.

Para o agrupamento propriamente dito, aplicamos os principais algoritmos não supervisionados do scikit-learn: o k-means para divisão particionada e o HDBSCAN* para detecção de clusters densos. A

qualidade dos agrupamentos foi avaliada rigorosamente usando métricas como o SilhoetteScore (que mede coesão e separação), CalinskiHarabaszScore (razão de dispersão entre clusters) e DaviesBouldinScore (distâncias intra e inter clusters).

Diferentemente do aprendizado supervisionado, que depende de dados rotulados, esta abordagem não supervisionada revelou padrões ocultos e poucas relações naturais nos dados (que se mostraram muito densos após análises iniciais). Todo o processo foi sustentado por um pipeline de pré-processamento, análise e modelagem, garantindo resultados confiáveis e interpretáveis.

3 Metodologia

3.1 Base de Dados

Neste trabalho, utilizamos como material de estudo um dataset retirado da plataforma Kaggle. O conjunto de dados trata sobre músicas disponibilizadas no Spotify, reunindo informações quantitativas e qualitativas sobre faixas, artistas e o desempenho das canções em termos de popularidade em outras plataformas digitais. A base de dados, intitulada *Spotify Dataset* e disponível através do link: Spotify Dataset, contém um total de 20.594 registros e 24 variáveis. Os atributos presentes no conjunto de dados são os seguintes:

- **Artist**: Nome do artista ou banda.
- **Track**: Título da música.
- **Album**: Nome do álbum em que a música foi lançada.
- **Album_type**: Tipo de álbum (por exemplo, álbum completo, single, etc.).
- **Danceability**: Métrica de dançabilidade da música, variando entre 0 e 1.
- **Energy**: Nível de energia da música (0 a 1).
- **Loudness**: Volume médio da faixa em decibéis.
- **Speechiness**: Proporção de conteúdo falado na música.
- **Acousticness**: Grau de sonoridade acústica da música.
- **Instrumentalness**: Probabilidade de a faixa ser instrumental.
- **Liveness**: Probabilidade de a música ter sido gravada ao vivo.
- **Valence**: Medida de positividade emocional da faixa (0 a 1).
- **Tempo**: Tempo da música em batidas por minuto (BPM).
- **Duration_min**: Duração da música em minutos.
- **Title**: Título da música no vídeo correspondente.
- **Channel**: Nome do canal de YouTube que publicou o vídeo da música.
- **Views**: Número de visualizações no YouTube.
- **Likes**: Número de curtidas no vídeo.
- **Comments**: Quantidade de comentários no vídeo.

- **Licensed**: Indica se a música é licenciada (True/False).
- **official_video**: Indica se o vídeo é oficial (True/False).
- **Stream**: Número de vezes que a música foi ouvida no Spotify.
- **EnergyLiveness**: Atributo composto baseado nos valores de *Energy* e *Liveness*.
- **most_playedon**: Plataforma onde a música é mais reproduzida.

O conjunto de dados não apresenta valores ausentes, o que facilita as etapas de pré-processamento. No entanto, por meio da aplicação da função `describe()`, foram observadas algumas irregularidades estatísticas, como valores extremos ou inconsistentes em determinadas colunas, exigindo inspeção e possíveis correções durante a análise exploratória dos dados. Além disso, A Figura 1 demonstra a correlação entre os atributos, demonstrando alguns padrões já esperados como a relação entre os atributos de popularidade.

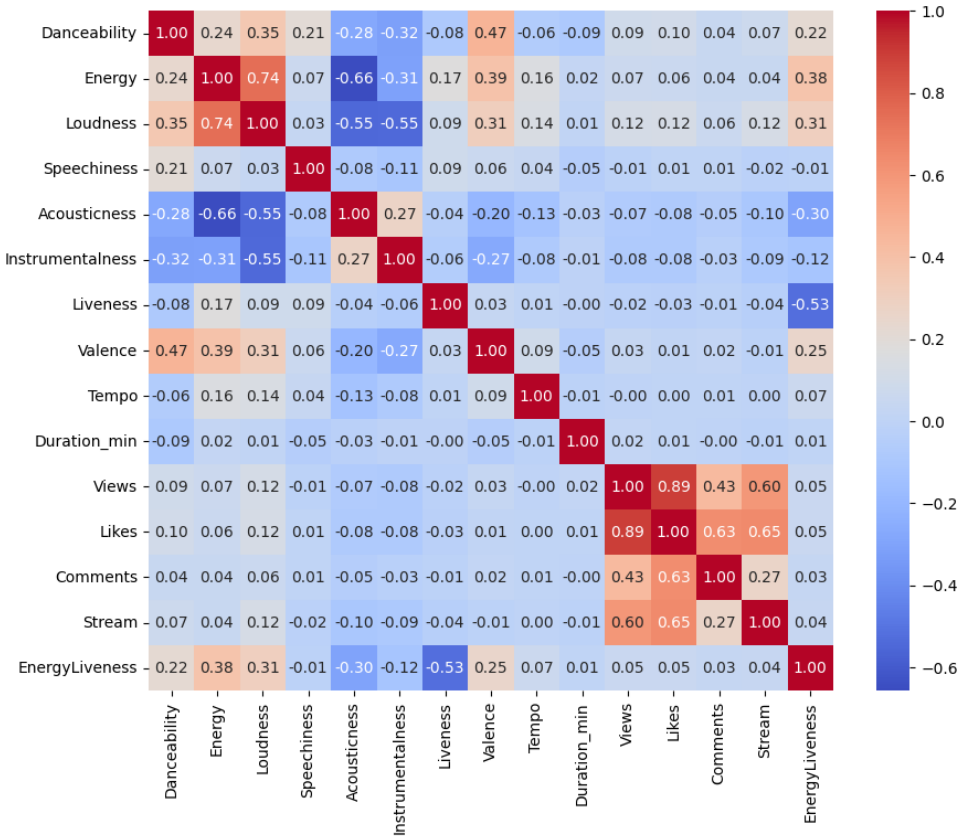

Figura 1: Correlação dos Atributo Numérico

3.2 Pré-processamento dos Dados

Inicialmente, visualizaram-se os histogramas de cada atributo numérico do conjunto de dados, com o objetivo de compreender a distribuição individual das variáveis. Essa etapa foi fundamental para detectar

assimetrias e presença de outliers nos dados.

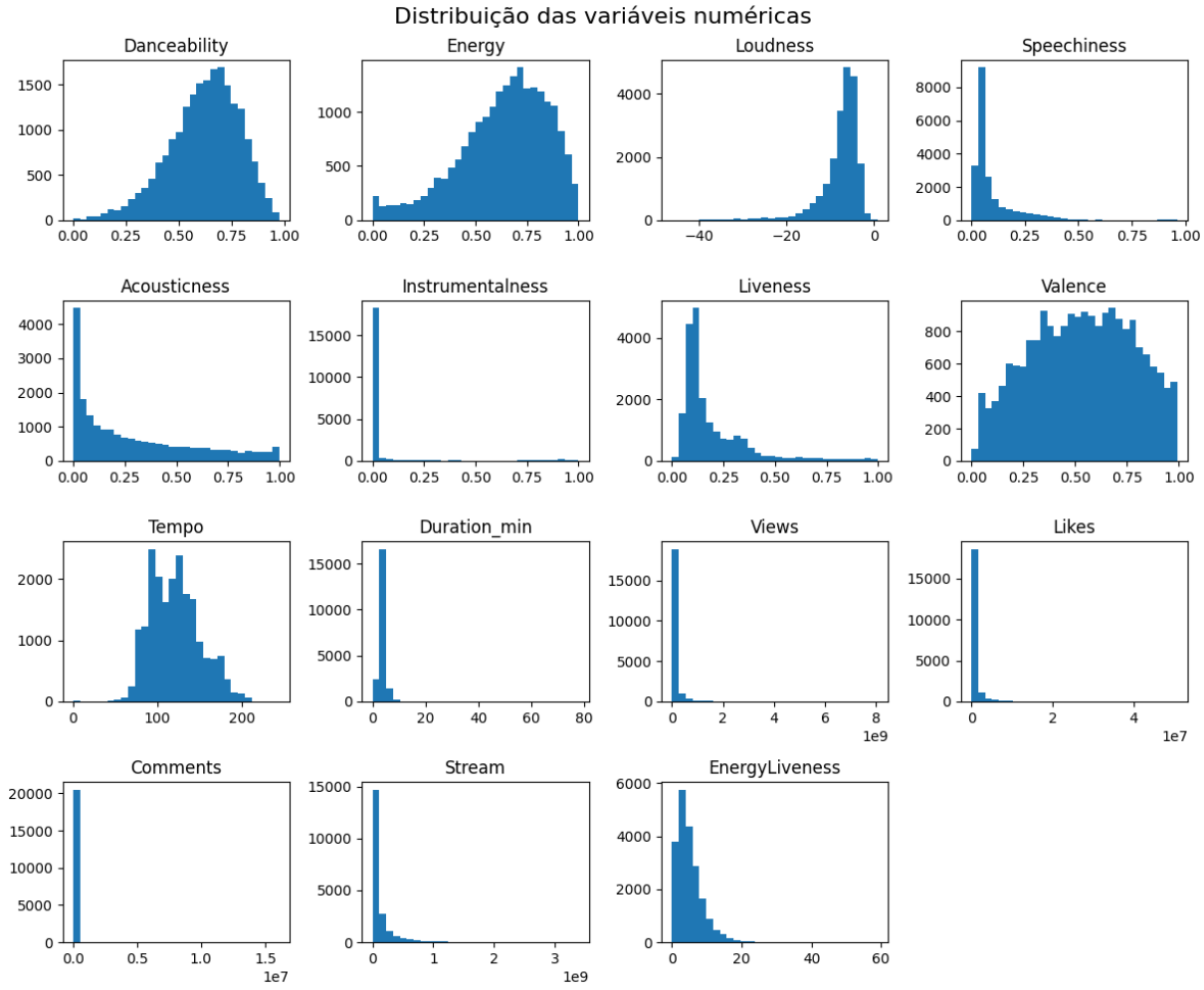

Figura 2: Histograma de Cada Atributo Numérico

Observa-se que diversos atributos apresentam distribuições assimétricas, especialmente aqueles relacionados à popularidade das faixas como *Streams*, *Comments*, *Likes* e *Views*. Esses atributos concentram a maior parte de seus valores em faixas mais baixas, com a presença de poucos registros com valores extremamente elevados, indicando possíveis outliers a serem considerados. Essa característica também é evidenciada pelo boxplot apresentado a seguir, que destaca de forma clara os valores atípicos e a dispersão entre quartis para esses atributos.

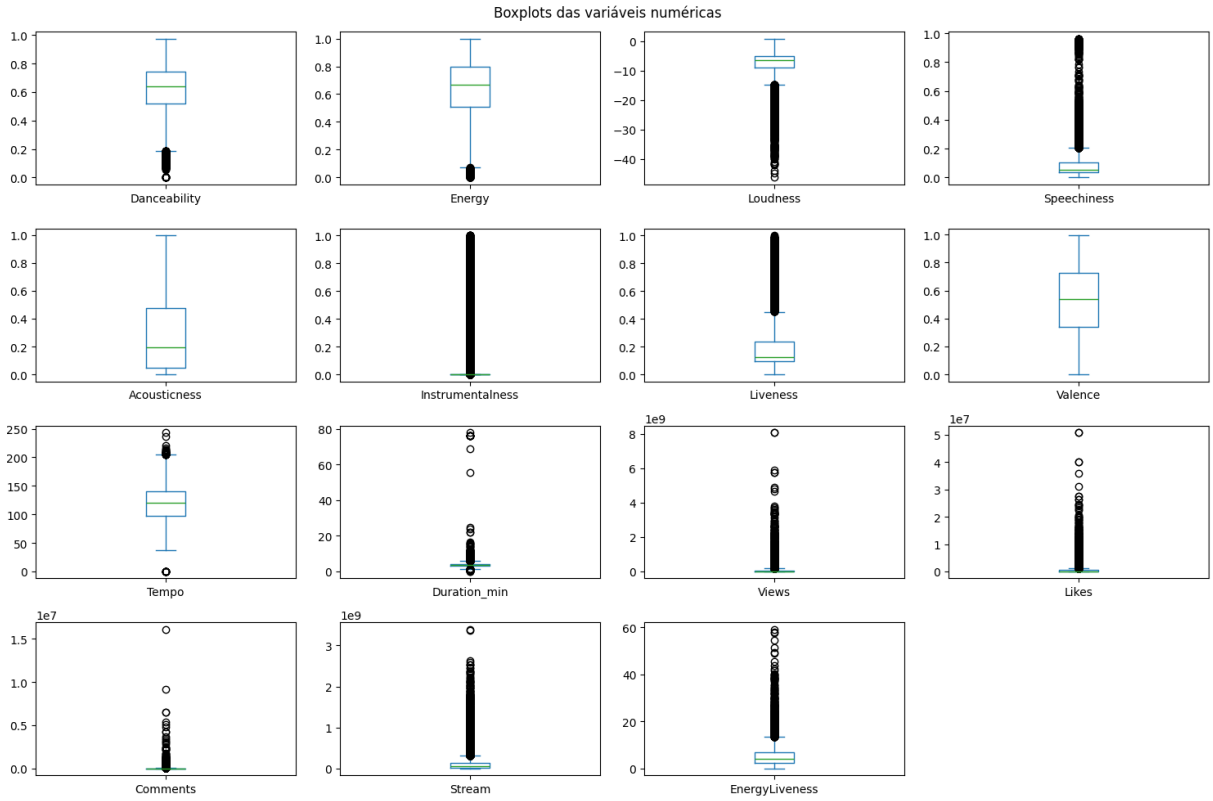

Figura 3: Box Plot de Cada Atributo Numérico

A presença de valores discrepantes nos atributos relacionados à popularidade motivou a criação de uma versão alternativa do conjunto de dados, com a aplicação de um filtro baseado no intervalo interquartil (IQR),tentando diminuir a quantidade de outliers.

Ademais, apesar da descrição fornecida na plataforma Kaggle indicar que o dataset estava previamente limpo, foram identificadas algumas inconsistências, como músicas sem duração ou sem valor de batidas por minuto, além de outros dados incoerentes. Dessa forma, optou-se por visualizar as três versões do conjunto de dados, sendo estas a versão original, a versão filtrada pelo IQR e uma versão com remoção dos dados inconsistentes. A Figura 4 ilustra a análise de componentes principais (PCA) aplicada a cada uma dessas versões.

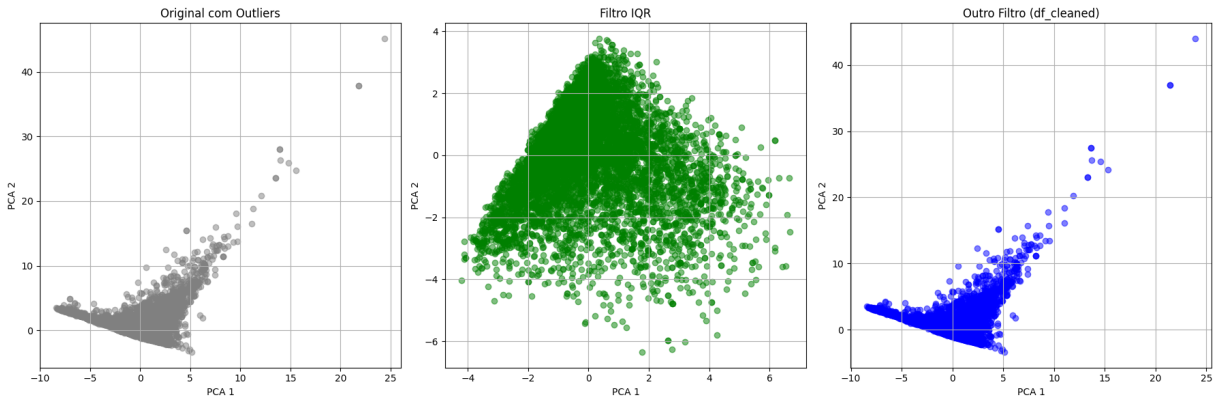

Figura 4: PCA de cada conjunto

É possível observar que a remoção dos valores irregulares não alterou visualmente a distribuição dos dados, uma vez que afetou apenas uma pequena parcela do conjunto total. Por outro lado, a aplicação do filtro do IQR resultou em uma redução significativa da base de dados, com a exclusão de aproximadamente 62,7% dos dados.

Desta forma, optou-se por não utilizar o conjunto de dados com o filtro IQR, porém continuar os testes apenas com os conjuntos dos dados tratados com remoção dos dados inconsistentes e criar um novo conjunto de dados removendo as colunas de popularidade (*Streams*, *Comments*, *Likes* e *Views*).

Por fim, decidiu-se utilizar outro método de visualização dos dados, combinando o algoritmo UMAP com o PCA, com o objetivo de explorar diferentes formas de representação e identificar Analisando A Figura 5possíveis agrupamentos de forma mais clara. apresenta os resultados obtidos com essa abordagem, na qual foi realizada uma variação do parâmetro `n_neighbors` do UMAP. Essa experimentação permitiu avaliar diferentes configurações e selecionar a que proporcionasse uma melhor separação visual dos dados, auxiliando na compreensão da estrutura interna do conjunto.

A partir da análise da Figura 5, observa-se a possível existência de dois grandes grupos no conjunto de dados e um pequeno grupo isolado. No entanto, é importante salientar que a variação do parâmetro `n_neighbors` no algoritmo UMAP influenciou diretamente a forma como esses grupos se relacionam, podendo inclusive resultar em sua aproximação ou até mesmo fusão visual, dependendo do valor escolhido.

3.3 Aplicação dos Algoritmos de Agrupamento

Foram testados diferentes algoritmos de agrupamento não supervisionado com o objetivo de identificar possíveis padrões e agrupamentos naturais dentro do conjunto de dados do Spotify. Optou-se pela aplicação dos algoritmos K-Means e HDBSCAN, com o intuito de comparar uma abordagem baseada em particionamento globular, que requer a definição prévia do número de clusters, com uma abordagem baseada em densidade, que determina automaticamente tanto os agrupamentos quanto os pontos considerados ruído, respectivamente.

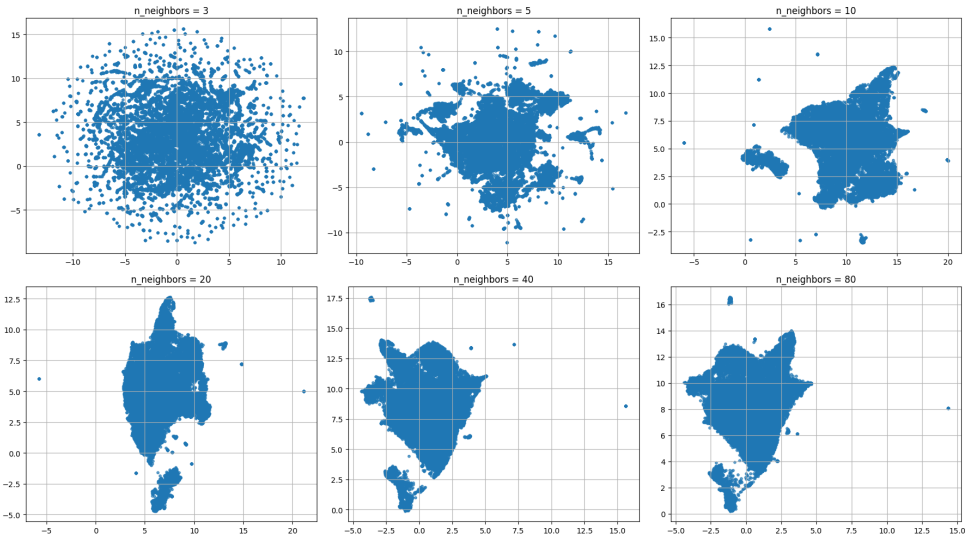

Figura 5: UMAP com diferentes parâmetros

Para todos os métodos foram aplicados os índices de avaliação externos e internos necessários para entender e comparar seus resultados. Além disso, foi visualizado o dendrograma a fim de compreender melhor os resultados e a avaliá-los.

3.3.1 K-Means

Para determinar um número adequado de clusters para o algoritmo K-Means, foram utilizados dois métodos de avaliação, sendo estes o método do cotovelo e o coeficiente de silhueta. A Figura 6 apresenta os resultados dessas duas análises para o conjunto dos dados limpos.

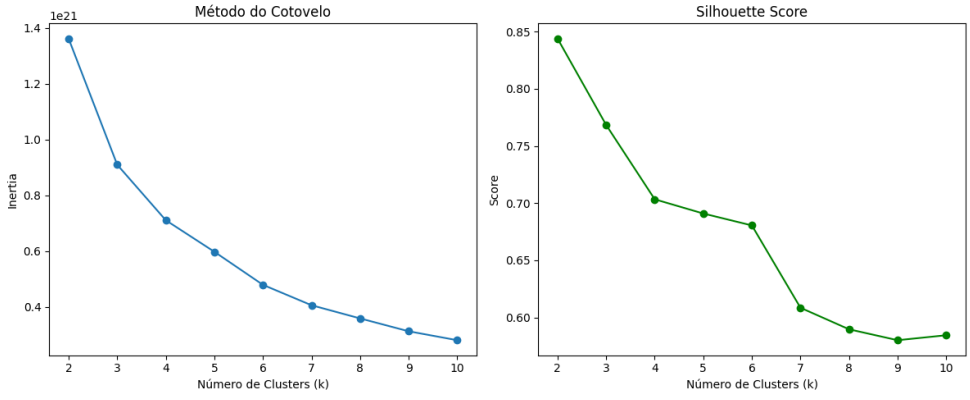

Figura 6: Método do Cotovelo e Silhueta

Com base nesses critérios, foi escolhido o valor de 6 clusters, pois nesse ponto a curva da inércia começa a se estabilizar, indicando um baixo ganho com o aumento do número de agrupamentos. Além disso, o score da silhueta ainda não havia apresentado uma queda significativa. Então, foi aplicado o método de K-Means, obtendo o resultado da Figura 7.

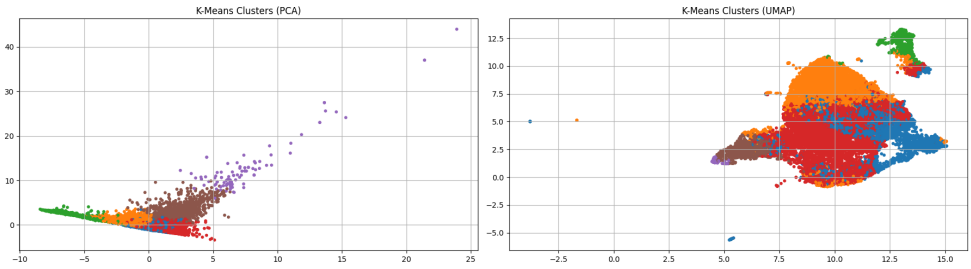

Figura 7: K-Means com 6 Clusters

Ademais, a representação visual dada pelo UMAP indifcava a existência de cerca de dois ou três grupos. Desta maneira, otou-se por também testar o valor de 3 clusters, como visto na Figura 8.

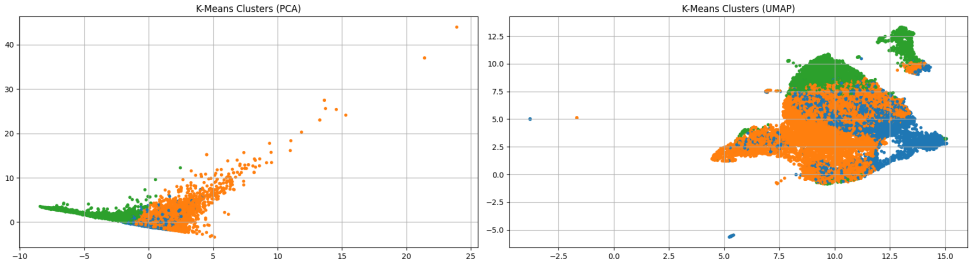

Figura 8: K-Means com 3 Clusters

Para o conjunto dos dados sem as colunas de popularidade obtevê-se valores diferentes no gráfico de cotovelo e índice de silhueta, então foi aplicado o algoritmo de K-Means com 5 clusters, obtendo o resultado da Figura 9.

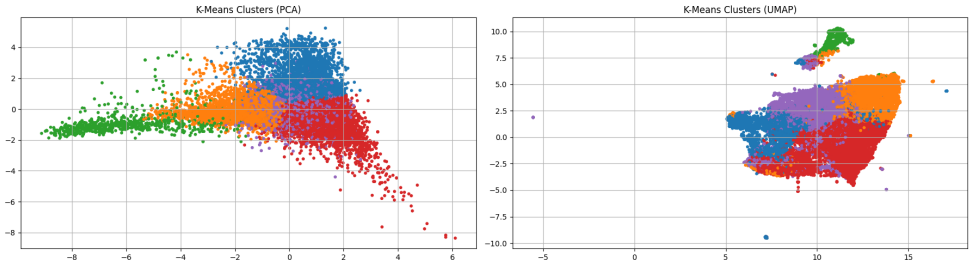

Figura 9: K-Means com conjunto sem colunas de Popularidade

3.3.2 HDBSCAN

Para a aplicação do algoritmo HDBSCAN, foram configurados parâmetros que influenciam diretamente na formação dos agrupamentos, como o `min_cluster_size`, definido como 50, de forma a considerar apenas grupos com pelo menos esse número de elementos. Além disso, foram ajustados os parâmetros `min_samples` e `cluster_selection_epsilon` para avaliar diferentes densidades. A Figura 10 apresenta o resultado visual da aplicação do HDBSCAN sobre o conjunto de dados limpos.

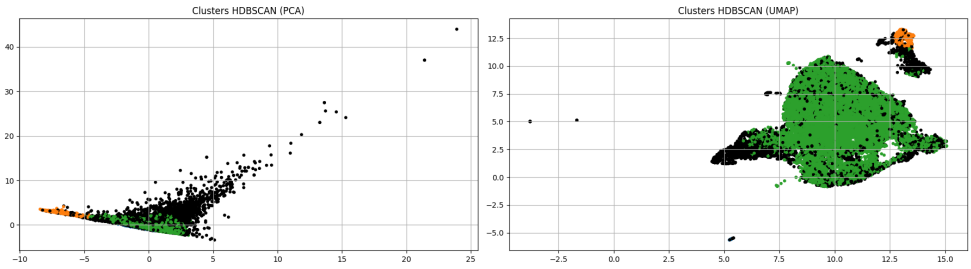

Figura 10: Resultado do HDBSCAN

Em seguida foi aplicado o mesmo algoritmo para o conjunto de dados com as colunas *Streams*, *Comments*, *Likes* e *Views* removidas.

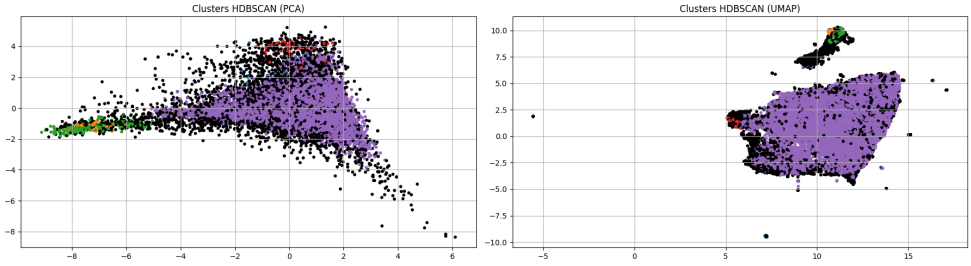

Figura 11: Resultado do HDBSCAN sem colunas de Popularidade

3.3.3 Dendrograma

Por fim, optou-se por realizar uma representação dos dados por meio de um dendrograma, com o objetivo de explorar a estrutura hierárquica de agrupamentos presentes no conjunto de dados sem popularidade.

A Figura 13 apresenta o dendrograma gerado, no qual é possível observar a formação de agrupamentos em diferentes níveis de ligação. Embora essa abordagem se torne mais complexa de interpretar em conjuntos de dados extensos, como o utilizado neste trabalho, ela ainda fornece indícios visuais úteis sobre a proximidade entre grupos de amostras e possíveis divisões naturais.

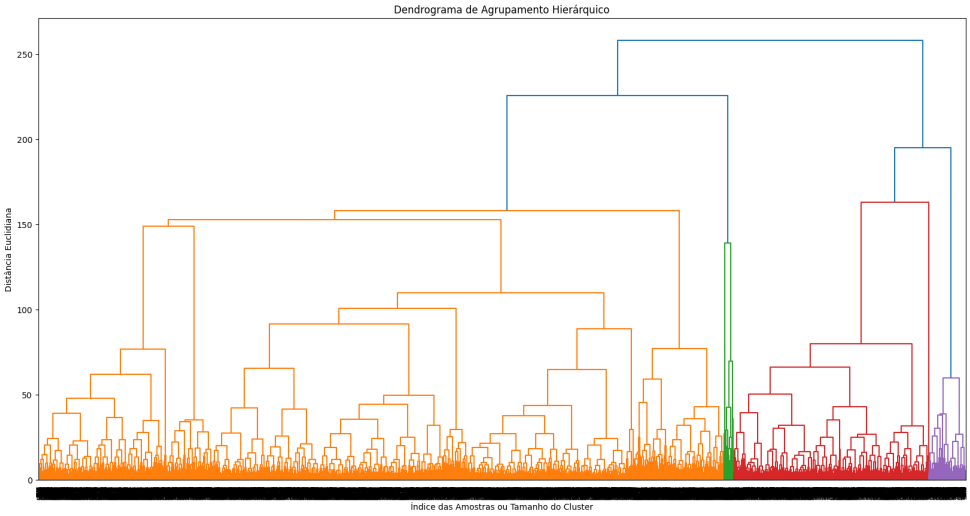

Figura 12: Dendrograma gerado

Ademais, realizou-se o corte do dendrograma com o intuito de gerar 6 clusters, permitindo uma comparação mais direta com os agrupamentos obtidos pelo K-Means e HDBSCAN. A Figura 13 apresenta o dendrograma após o corte, evidenciando os grupos formados a partir da estrutura hierárquica identificada nos dados.

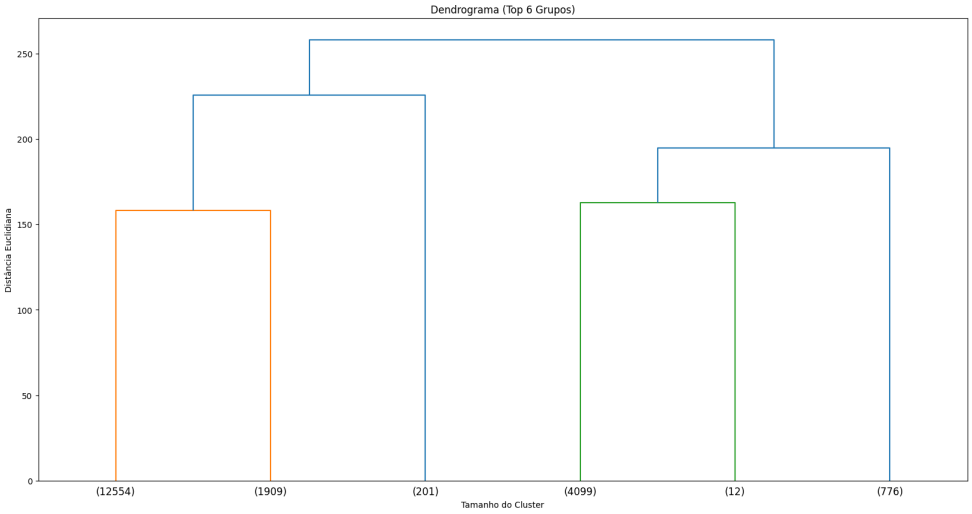

Figura 13: Dendrograma Cortado

4 Discussão dos Resultados

Diversos testes foram conduzidos com os algoritmos K-Means e HDBSCAN aplicados a diferentes versões do conjunto de dados, buscando-se identificar agrupamentos coesos e bem separados. A comparação entre os métodos também considerou métricas internas de avaliação, como *Silhouette Score*, *Calinski-Harabasz Index*, *Davies-Bouldin Index* e métricas externas, como o *Rand Index*.

4.1 Distribuição dos Grupos

A seguir, é apresentado um resumo da distribuição dos clusters gerados por cada modelo:

- **K-Means (3 clusters) — conjunto limpo:**
  - Cluster 0: 5030 elementos
  - Cluster 1: 11140 elementos
  - Cluster 2: 3381 elementos
- **K-Means (6 clusters) — conjunto limpo:**
  - Cluster 0: 4315 elementos
  - Cluster 1: 4613 elementos
  - Cluster 2: 787 elementos
  - Cluster 3: 8651 elementos
  - Cluster 4: 97 elementos
  - Cluster 5: 1088 elementos
- **HDBSCAN — conjunto limpo:**
  - Cluster 0: 408 elementos
  - Cluster 1: 14730 elementos
  - Ruído (`-1`): 4413 elementos
  - Número de clusters válidos (excluindo ruído): 2
- **K-Means (5 clusters) — sem variáveis de popularidade:**
  - Cluster 0: 3468 elementos
  - Cluster 1: 3894 elementos
  - Cluster 2: 767 elementos
  - Cluster 3: 6768 elementos

  - Cluster 4: 4654 elementos

- **HDBSCAN — sem variáveis de popularidade:**
  - Cluster 0: 51 elementos
  - Cluster 1: 53 elementos
  - Cluster 2: 206 elementos
  - Cluster 3: 72 elementos
  - Cluster 4: 13753 elementos
  - Ruído (`-1`): 5416 elementos
  - Número de clusters válidos (excluindo ruído): 5

4.2 Índices Internos

Para o HDBSCAN aplicado ao conjunto limpo, obteve-se um *Silhouette Score* de 0,5539, valor considerado razoável e que sugere uma separação satisfatória entre os grupos formados, desconsiderando os pontos classificados como ruído. No entanto, o índice DBCV, que é mais apropriado para algoritmos baseados em densidade, apresentou um valor negativo (-0,1126), indicando que a estrutura de agrupamento encontrada pode conter sobreposições ou falta de coesão entre os clusters.

Já para o HDBSCAN aplicado à versão do dataset sem as variáveis de popularidade, os resultados foram ligeiramente inferiores. O *Silhouette Score* caiu para 0,3673 e o índice DBCV apresentou um valor ainda mais negativo (-0,3124), sugerindo maior dificuldade do algoritmo em formar agrupamentos densos e bem definidos com essa configuração de dados.

Tabela 1: Métricas do HDBSCAN em diferentes versões do conjunto de dados

Conjunto de Dados	**Silhouette Score**	**DBCV**
Conjunto Limpo	0,5539	-0,1126
Sem Popularidade	0,3673	-0,3124

Já para o K-Means, os resultados variaram conforme o número de clusters:

Tabela 2: Métricas dos modelos K-Means

Configuração	**Silhouette Score**	**Calinski-Harabasz**	**Davies-Bouldin**
K-Means (3 clusters)	0,123	2345,268	2,214
K-Means (6 clusters)	0,158	2502,714	1,613
K-Means (5 clusters, sem popularidade)	0,151	3058,858	1,774

Esses valores indicam que a separação dos clusters melhora com o aumento dos grupos, algo já esperado. Contudo, não há uma diferença significante nas métricas que indique que excluir as colunas de popularidade

melhore o desempenho. Além disso, o valor do índice de silhueta foi relativamente baixo, indicando que o algoritmo K-Means teve dificuldade em encontrar grupos bem difinidos.

4.3 Indice Externo

O Rand Index (não ajustado) entre o agrupamento do K-Means com 3 clusters e o HDBSCAN, ambos aplicados ao conjunto limpo original, foi de 0,4657, sugerindo uma sobreposição parcial entre os agrupamentos. No entanto, o Rand Index ajustado, que leva em consideração a chance, apresentou um valor significativamente mais baixo (0,0528), indicando baixa concordância real entre as partições.

5 Conclusão

De maneira geral, os resultados obtidos pelos algoritmos de agrupamento aplicados ao conjunto de dados do Spotify não apresentaram métricas internas e externas suficientemente elevadas que indiquem a formação de grupos bem definidos. Embora o HDBSCAN tenha apresentado um desempenho visual mais interessante, suas métricas quantitativas sugerem fraca coesão dos clusters. Por outro lado, o K-Means, além de apresentar baixa performance nas métricas de avaliação, resultou em agrupamentos visualmente menos definidos.

Essa limitação nos resultados pode estar associada à própria natureza do dataset, que possui uma grande variedade de atributos musicais contínuos e interdependentes, além de uma possível ausência de estruturas naturais de agrupamento bem definidas. Assim, os algoritmos podem ter encontrado dificuldade em agrupar os dados de maneira significativa.

Portanto, conclui-se que, embora os métodos testados tenham fornecido algumas informações úteis e visualizações interpretáveis, o agrupamento neste caso específico não pareceu eficaz.

6 Referências

KAGGLE. Spotify Dataset. Kaggle, 2022. Disponível em: https://www.kaggle.com/datasets/sanjanchaudhari/spotify-dataset. Acesso em: 01 jul. 2025.

PANDAS. Pandas Documentation. Disponível em: https://pandas.pydata.org/. Acesso em: 30 jun. 2025.

NUMPY. NumPy. Disponível em: https://numpy.org/. Acesso em: 30 jun. 2025.

WASKOM, M. Seaborn: Statistical Data Visualization. Disponível em: https://seaborn.pydata.org/. Acesso em: 03 jul. 2025.

HUNTER, J. D. et al. Matplotlib. Disponível em: https://matplotlib.org/. Acesso em: 03 jul. 2025.

SCIKIT-LEARN DEVELOPERS. Scikit-learn: Machine Learning in Python. Disponível em: https://scikit-learn.org/. Acesso em: 02 jul. 2025.

SCIPY DEVELOPERS. SciPy: Scientific computing tools for Python. Disponível em: https://scipy.org/. Acesso em: 02 jul. 2025.

MCINNES, L.; HEALY, J.; ASTELS, S. hdbscan: Hierarchical Density Based Clustering. Journal of Open Source Software, v. 2, n. 11, p. 205, 2017. Disponível em: https://doi.org/10.21105/joss.00205. Acesso em: 04 jul. 2025.