

Algoritmo de Recomendação de Músicas Baseado em Grafos

Ádrian Henrique de Abreu Paiva

Engenharia de Computação

CEFET-MG, Campus V

Divinópolis, Brasil

adrianabreupaiva@gmail.com

Heitor Francisco Costa Xavier

Engenharia de Computação

CEFET-MG, Campus V

Divinópolis, Brasil

heitorfrancisco2005@hotmail.com

Resumo—Plataformas de *streaming* enfrentam o desafio de recomendar conteúdo em bibliotecas massivas. Desenvolvemos um sistema híbrido que integra *Detecção Global de Comunidades (Louvain)* com *Ponderação Local de Informação (IIF)*. O objetivo é fornecer uma solução escalável e eficiente para ambientes com recursos computacionais limitados, utilizando o *Spotify Million Playlist Dataset*.

A análise inicial revelou a severidade da distribuição de *cauda longa* (*Princípio de Pareto*), onde faixas populares obscurecem nichos de gosto. Para mitigar esse problema, propomos uma abordagem de grafos bipartidos otimizada por: *Filtragem Inteligente (Smart Pruning)*, que minimiza ruídos estruturais, e *Frequência Inversa do Item (IIF)*, que penaliza itens excessivamente genéricos.

Avaliado por estudos de caso qualitativos e análise estrutural de comunidades, o modelo demonstrou capturar com êxito nichos semânticos. Os resultados confirmam a coerência temática das recomendações, utilizando apenas dados de interações implícitas.

Index Terms—Sistemas de Recomendação, Detecção de Comunidades, Grafos Bipartidos, Cauda Longa, Smart Pruning, Frequência Inversa do Item (IIF), Louvain, Streaming de Música, Spotify Million Playlist Dataset

I. INTRODUÇÃO

O crescimento acelerado de serviços de *streaming* transformou a forma como músicas são consumidas em escala global. Plataformas como Spotify, Apple Music e Deezer precisam lidar com catálogos cada vez maiores, nos quais milhões de faixas coexistem e novos conteúdos são adicionados diariamente. Nesse cenário, sistemas de recomendação tornam-se essenciais para uma boa experiência do usuário, evitando que a vasta quantidade de opções se transforme em um obstáculo para a descoberta de conteúdo.

Apesar dos avanços recentes, a recomendação musical ainda enfrenta desafios significativos. A distribuição das músicas nesses catálogos segue o padrão conhecido como **Cauda Longa** (*Long Tail*), no qual uma pequena fração de faixas extremamente populares domina a grande maioria das *playlists* de usuários, enquanto um enorme volume de músicas permanece com baixa visibilidade. Esse fenômeno, alinhado ao **Princípio de Pareto**, cria um viés de popularidade que pode limitar a diversidade das recomendações e esconder nichos de interesses específicos.

Uma solução promissora para esse problema é o uso de modelos baseados em grafos, que representam relações entre

playlists e músicas, permitindo capturar padrões coletivos de consumo. Nesse contexto, técnicas de *clustering*, como o algoritmo de Louvain, destacam-se por revelar agrupamentos (comunidades) de músicas que compartilham perfis de audiência semelhantes, capturando nuances que métodos puramente estatísticos muitas vezes ignoram.

Neste trabalho, apresentamos o desenvolvimento de um sistema de recomendação híbrido baseado em grafos bipartidos, utilizando dados reais do *Spotify Million Playlist Dataset* [1]. O objetivo é investigar a viabilidade de processamento de grafos de larga escala em ambientes de recursos computacionais limitados. Diferente de abordagens que dependem de infraestrutura de alto desempenho, propomos uma estratégia de filtragem seletiva e uma heurística de ponderação baseada em Frequência Inversa de Item (IIF) para mitigar o ruído gerado por músicas super-populares, validando a eficácia do método através de estudos de caso qualitativos.

II. TRABALHOS CORRELATOS

A literatura em sistemas de recomendação tem evoluído significativamente nos últimos anos com a adoção de modelos baseados em grafos para lidar com dados esparsos e viés de popularidade. Enquanto trabalhos iniciais como o de Burke [2] estabeleceram bases híbridas combinando filtragem colaborativa e baseada em conteúdo, simultaneamente careciam de ênfase em estrutura de grafos para detecção de comunidades. Nesse contexto, Newman e Girvan [3] e Blondel [4] consolidaram a modularidade e o algoritmo Louvain para revelar agrupamentos latentes em grafos bipartidos. No entanto, tais abordagens frequentemente assumem recursos ilimitados, ignorando cenários de hardware restrito como o nosso (8GB RAM).

No domínio da recomendação musical, o estudo de Saravanan et al. [5] explora representações graph-based induktivas via aprendizado multi-task, priorizando conexões transitivas para melhorar a precisão em itens da cauda longa — uma lógica similar à nossa análise de co-ocorrências. Contudo, seu foco em features de áudio e a dependência de modelos neurais elevam a complexidade computacional, contrastando com nossa poda seletiva e heurística IIF. Adicionalmente, Pesek [6] desenvolve um sistema híbrido com GNNs (graph neural networks) e detecção de comunidades que supera a

esparsidade no Million Song Dataset, alcançando ganhos de até 25% em recall; entretanto, a ausência de penalização para itens genéricos corre o risco de perpetuar o viés de popularidade.

Em síntese, embora a literatura valide o uso de grafos e o algoritmo de Louvain para recomendação, ainda persistem lacunas em otimizações para recursos computacionais limitados e controle de viés em datasets musicais. Nossa contribuição preenche essas brechas ao integrar poda seletiva e a heurística IIF em um pipeline híbrido, promovendo recomendações diversificadas e eficientes, conforme validado nos estudos de caso no Spotify Million Playlist Dataset.

III. METODOLOGIA

Para viabilizar a recomendação em ambiente de computação local, onde há restrições de hardware, como por exemplo a restrição de 8GB RAM, e mitigar os desafios impostos pela cauda longa, desenvolvemos um *pipeline* otimizado, conforme ilustrado na Figura 1, composto por estágios de pré-processamento, modelagem e aplicação de heurísticas ponderadas.

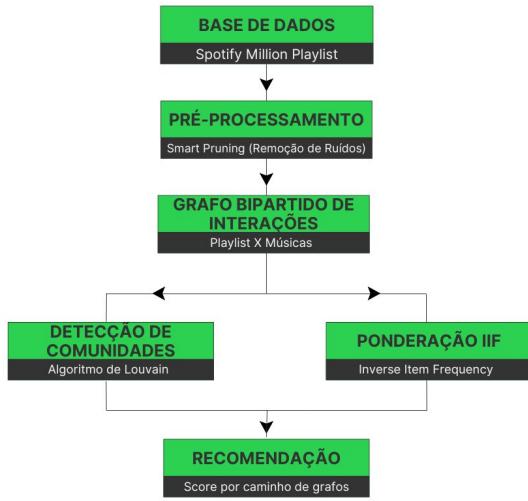


Figura 1. Fluxo de processamento: da ingestão de dados brutos à recomendação ponderada.

A. Pré-Processamento (Smart Pruning)

Baseado na análise de Cauda Longa, que indicou que mais da metade das músicas possuem baixíssima visibilidade, implementamos uma estratégia de filtragem agressiva, denominada **Smart Pruning**, para reduzir a dimensionalidade antes da instaciação do grafo. A remoção desses nós da "cauda profunda" elimina a complexidade espacial desnecessária, mantendo apenas a espinha dorsal da conectividade da rede.

Foram aplicados dois limiares rígidos:

- 1) **Consistência de Playlist** ($P_{min} \geq 20$): Descartam-se playlists com menos de 20 faixas, pois vetores de contexto muito curtos geram ruído na detecção de similaridade.
- 2) **Relevância de Faixa** ($F_{min} \geq 5$): Descartam-se músicas com menos de 5 aparições em playlists.

Tabela I
COMPARAÇÃO DAS MÉTRICAS ANTES E DEPOIS DO SMART PRUNING.

Métrica	Antes	Depois
Nós (vértices)	750.214	212.188
Arestas	6.387.259	5.582.479
Grau médio	17,03	52,62

Ao observar a Tabela 1, vemos o impacto direto da aplicação da Poda Seletiva sobre o grafo original. A redução no número de nós e arestas não representa apenas economia de armazenamento ou ganho de desempenho computacional, mas sim uma reestruturação qualitativa da *database*. A remoção de playlists muito curtas e de músicas com recorrência extremamente baixa elimina elementos que contribuem pouco para a conectividade global, reduzindo ruído estrutural. Além disso, o aumento significativo do grau médio indica que a proporção de conexões por nó se tornou maior. Esse comportamento sugere que o grafo resultante é mais denso e coeso e menos propenso a fragmentações, sendo o tipo de estrutura ideal para métodos de recomendação.

B. Modelagem e Construção do Grafo Bipartido

Implementamos a estrutura de **Grafo Bipartido** $G = (U \cup V, E)$, que é uma estrutura matemática que modela a relação entre dois conjuntos disjuntos de nós: Playlists (U) e Músicas (V). Uma aresta existe apenas entre um nó de U e um nó de V , nunca entre dois nós do mesmo conjunto.

A topologia foi construída utilizando listas de adjacência, exclusivamente sobre interações implícitas: uma aresta é criada entre a playlist u e a música v se, e somente se, a faixa sobreviveu à etapa de poda e está contida na lista.

A recomendação neste cenário é gerada através da exploração de **caminhos transitivos de comprimento 3**, sugerindo ao usuário itens consumidos por playlists que compartilham gostos similares, ou seja: $(p_a \rightarrow m_x \rightarrow p_b \rightarrow m_y)$ Onde:

- p_a = Playlist A (Nó de origem)
- m_x = Música X (Primeira Aresta)
- p_b = Playlist B (Caminho de comprimento 2)
- m_y = Música Y (Caminho de Comprimento 3)

O mecanismo de recomendação conclui que, dado que a Playlist B compartilha o gosto pela Música X com a Playlist A, qualquer outra música na Playlist B é uma forte candidata a ser recomendada para a Playlist A. A força dessa recomendação é determinada pela Ponderação IIF.

C. Detecção de Comunidades com Algoritmo de Louvain

Em redes complexas, nós tendem a se organizar em módulos densamente conectados, conhecidos como comunidades. A identificação desses agrupamentos é fundamental, pois permite revelar padrões de consumo compartilhado.

O Algoritmo de Louvain é um método heurístico altamente eficiente para clusterização de grafos de larga escala. No contexto desse trabalho, a estrutura de comunidades emerge

diretamente dos hábitos dos usuários: músicas frequentemente agrupadas em playlists similares tendem a ser similares, refletindo gêneros musicais específicos.

D. Implementação da Ponderação IIF

Para operacionalizar a métrica de Frequência Inversa (*IIF*), substituímos a contagem simples de caminhos por uma soma ponderada. O objetivo é reduzir matematicamente a influência dos *hubs* globais (músicas super populares) na pontuação final.

1) *Cálculo do Peso de Raridade (IIF)*: Primeiro, definimos o peso de raridade de uma música m como:

$$IIF(m) = \frac{1}{\log(|Vizinhos(m)| + \epsilon)} \quad (1)$$

onde $|Vizinhos(m)|$ representa a popularidade global da música, medida pelo número de playlists que a contêm. O termo ϵ evita indeterminação e o uso do logaritmo suaviza a penalidade, garantindo que mesmo *hits* ainda possam ter algum peso, embora reduzido.

2) *Cálculo do Score de Recomendação Ponderado*: O *score* de recomendação final, utilizado para ranquear os candidatos, é baseado na soma dos pesos de raridade (*IIF*) de todas as músicas que servem como pontes de conexão.

Para uma playlist alvo p e uma música candidata m , o *Score* é a somatória do *IIF* de todas as músicas v que formam o caminho transitivo ($p \rightarrow v \rightarrow p' \rightarrow m$):

$$Score(p, m) = \sum_{v \in Vizinhos(p)} IIF(v) \quad (2)$$

Onde o termo $IIF(v) = \frac{1}{\log(|Vizinhos(v)| + \epsilon)}$ é o peso da música-ponte v que conecta p a m . Essa abordagem garante que caminhos formados por músicas de nicho (raras) contribuam significativamente mais para o *score* do que caminhos formados por *hubs* genéricos.

a) *Exemplo Didático de Ponderação*: Para ilustrar o impacto da ponderação, consideremos duas músicas-ponte hipotéticas que conectam a *Playlist A* a um *Candidato Y*:

- **Música-Ponte Vaga (v_{vaga})**: É um *hit* de pop global que aparece em 10.000 playlists. Seu alto $|Vizinhos|$ (popularidade) resulta em um peso *IIF* muito baixo, por exemplo: $IIF(v_{vaga}) \approx 0.1$.
- **Música-Ponte Nível (v_{nicho})**: É uma faixa de nicho que aparece em apenas 100 playlists (após o *Smart Pruning*). Seu baixo $|Vizinhos|$ resulta em um peso *IIF* alto, por exemplo: $IIF(v_{nicho}) \approx 0.8$.

O sistema prioriza a Música-Ponte v_{nicho} porque ela é um indicador mais forte de um gosto específico. Se o *Candidato Y* for sugerido através de 10 caminhos via v_{vaga} , ele acumula um score total de $10 \times 0.1 = 1.0$. No entanto, se for sugerido através de apenas 2 caminhos via v_{nicho} , ele acumula um score de $2 \times 0.8 = 1.6$.

Desta forma, a lógica *IIF* recompensa playlists que compartilham gostos específicos e raros, reduzindo a probabilidade de recomendar um item apenas por ele ser globalmente popular.

IV. CARACTERIZAÇÃO DO DATASET

A eficácia de um sistema de recomendação depende diretamente da qualidade das relações existentes nos dados. Para este trabalho, utilizamos o *Spotify Million Playlist Dataset*. Realizamos uma análise estatística exploratória em uma amostra significativa (50 arquivos, totalizando 3,3 milhões de interações) para compreender a distribuição e definir as restrições de projeto.

A. A Distribuição de Cauda Longa

A análise de frequência das faixas revelou que o consumo musical no dataset obedece rigorosamente a uma distribuição de Lei de Potência (*Power Law*). Conforme ilustrado na Figura 2, observamos uma enorme concentração: uma pequena elite de músicas domina o volume de reproduções, enquanto a vasta maioria do catálogo compõe uma cauda longa de baixa visibilidade.

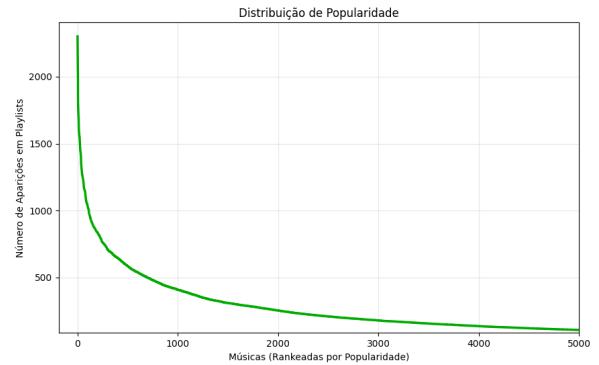


Figura 2. Distribuição Rank-Freqüência do catálogo.

Os dados quantitativos reforçam a severidade dessa distribuição:

- **Hiper-concentração**: Apenas **12,96%** das músicas são responsáveis por 80% de todas as interações registradas.
- **Ruído Massivo**: Identificamos que **57,42%** das músicas aparecem em apenas uma única playlist em toda a amostra analisada.

Essa distribuição impõe um desafio crítico de engenharia para a execução em hardware limitado (8GB RAM). Músicas que aparecem apenas uma vez representam um custo computacional proibitivo: elas ocupam espaço na lista de adjacência, mas não geram caminhos transitivos. Em termos de grafos, esses nós são "folhas" isoladas que não contribuem para a conectividade da rede nem para a detecção de comunidades.

Portanto, a "Cauda Longa" não é apenas uma curiosidade estatística, mas a evidência de que uma estratégia de **Poda Seletiva** é obrigatória para viabilizar o processamento do grafo em memória.

V. RESULTADOS

Nesta seção avaliamos o desempenho do sistema proposto a partir de casos de teste, considerando duas dimensões de análise:

- Estruturação do grafo e formação de comunidades;
- Coerência semântica das recomendações;

A. Otimização e Redução com LSH

Antes de analisarmos a formação de comunidades, é crucial abordar o desafio computacional imposto pela magnitude do *Spotify Million Playlist Dataset*. Embora a etapa de *Smart Pruning* (Seção III.A) tenha viabilizado o processamento do grafo em memória, a densidade de conexões restantes (co-ocorrências) ainda gerava uma estrutura massiva. Essa complexidade estrutural impossibilitava tanto a renderização gráfica legível em ferramentas como o Gephi quanto a convergência eficiente do algoritmo de Louvain.

Para superar essa barreira de escalabilidade estrutural, aplicamos a técnica de Locality Sensitive Hashing (LSH) utilizando o esquema *MinHash*. O LSH permitiu refinar a rede, atuando como um filtro de densidade adicional, essencial para a análise de comunidades, sem comprometer a coerência semântica.

O algoritmo funcionou da seguinte forma:

- Mapeou-se a "assinatura" de cada música baseada em suas co-ocorrências em playlists (similaridade de Jaccard);
- Identificaram-se pares de músicas com alta probabilidade de similaridade (acima de um limiar de 0.5);
- O grafo foi reconstruído, mantendo apenas as arestas e nós que formavam conexões relevantes baseadas nessa similaridade pré-calculada.

Foi esse "corte" estratégico via LSH que reduziu drasticamente o ruído estrutural remanescente e tornou possível a execução eficiente do algoritmo de Louvain, permitindo a geração das visualizações coerentes de comunidades apresentadas nas Figuras 3 e 4, que seriam ilegíveis e superpostas sem essa otimização.

Os resultados demonstram que a modelagem baseada em grafos é capaz de capturar padrões reais de consumo, por meio do Algoritmo de Louvain, e utilizá-los de maneira efetiva no processo de recomendação.

B. Estrutura de Comunidades no Grafo

Após o pré-processamento com *Smart Pruning* e a construção do grafo bipartido, aplicamos o algoritmo de Louvain para identificar agrupamentos de músicas com comportamentos de consumo similares.

A Figura 3 ilustra a projeção parcial do cluster mais representativo. É possível observar que:

- nós centrais apresentam grau elevado: são as músicas "âncoras" do nicho;
- a conectividade é fortemente concentrada dentro do cluster;
- o agrupamento resultante é dominado por um mesmo estilo musical.

Esses resultados confirmam que o grafo preserva coerência semântica: músicas consumidas por públicos semelhantes tendem a ficar próximas. Dessa forma, a detecção de comunidades é validada como componente importante no processo de recomendação.

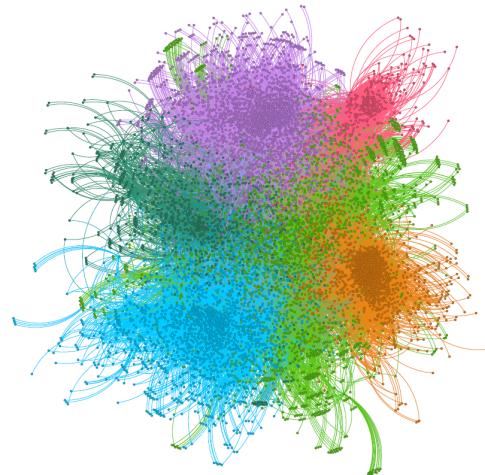


Figura 3. Visualização de comunidades detectadas pelo algoritmo de Louvain.

C. Estrutura do Grafo Playlist-Música

Além da visualização das comunidades detectadas pelo algoritmo de Louvain, também analisamos a estrutura bruta do grafo bipartido resultante do pré-processamento. A Figura 4 apresenta um recorte desse grafo, contendo apenas dois tipos de nós: **playlists** e **músicas**, sem qualquer distinção por cluster ou categoria.

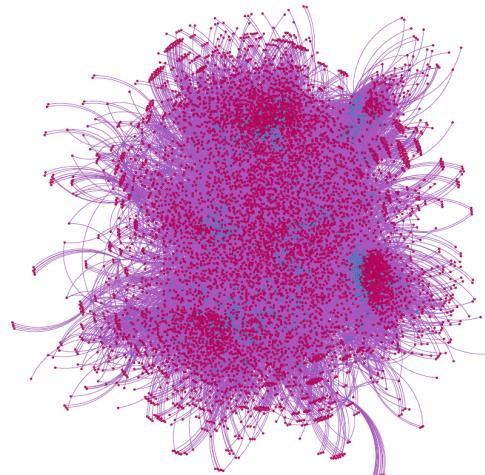


Figura 4. Grafo bipartido contendo músicas (nós azuis) e playlists (nós rosas).

A visualização evidencia dois comportamentos estruturais importantes:

- A estrutura do grafo é caracterizada pela extrema esparcidade, um reflexo direto do filtro $F_{min} \geq 5$ (Relevância de Faixa), que eliminou a vasta **Cauda Longa**, mantendo apenas o conjunto limitado de faixas que servem como *hubs*. Desta forma, o grafo processado é denso nas conexões restantes, mas esparso em termos da relação original de Playlists-por-Música

- As músicas se repetem entre várias playlists. Um mesmo nó de música pode estar conectado a dezenas de playlists diferentes.

Essa redundância de conexões é justamente o que torna os grafos bipartidos um modelo adequado para recomendação: músicas que aparecem juntas em muitas playlists distintas formam padrões de co-ocorrência consistentes, capazes de indicar afinidade temática mesmo na ausência de metadados musicais.

Do ponto de vista estrutural, o grafo playlist-música também reforça a assimetria característica do *Spotify Million Playlist Dataset*: a plataforma contém um volume muito maior de Playlists criadas por usuários do que de faixas, o que contribui para a *sparsity* observada em partes menos populares da distribuição (cauda longa). Entretanto, nas regiões onde músicas são amplamente compartilhadas entre playlists, o grafo torna-se denso e altamente informativo, permitindo identificar nichos musicais bem definidos.

D. Avaliação Qualitativa das Recomendações

Para avaliar a relevância perceptiva das recomendações, realizamos testes interativos buscando músicas específicas no grafo. A Tabela II apresenta um exemplo para a música “*Bohemian Rhapsody — Queen*”, utilizada como referência de rock clássico.

Tabela II

TOP-15 RECOMENDAÇÕES PARA “BOHEMIAN RHAPSODY — QUEEN” UTILIZANDO O MODELO PROPOSTO.

Rank	Música	Score
1	Another One Bites The Dust (Remastered 2011) — Queen	107.22
2	Under Pressure (Remastered 2011) — Queen	94.61
3	Hotel California (Remastered) — Eagles	92.75
4	Don't Stop Believin' — Journey	92.21
5	Sweet Child O' Mine — Guns N' Roses	91.79
6	Sweet Home Alabama — Lynyrd Skynyrd	90.97
7	Piano Man — Billy Joel	86.25
8	Dream On — Aerosmith	82.79
9	More Than a Feeling — Boston	81.37
10	We Will Rock You (Remastered) — Queen	79.15
11	Carry On Wayward Son — Kansas	78.17
12	Don't Stop Me Now (Remastered) — Queen	77.72
13	Livin' On A Prayer — Bon Jovi	77.58
14	Stairway To Heaven — Led Zeppelin	76.06
15	Have You Ever Seen The Rain? — Creedence Clearwater Revival	72.72

Outros casos avaliados apresentaram variações relevantes no valor dos *scores*, como observado com a música “*Grenade — Bruno Mars*”.

Ao comparar as recomendações geradas para “*Grenade*” com os resultados obtidos para “*Bohemian Rhapsody*”, observamos que o *score* médio permanece elevado, porém em uma escala distinta. Enquanto músicas extremamente populares e amplamente difundidas, como Queen, acumulam evidências em diversas playlists independentes, “*Grenade*” se insere em

Tabela III
TOP-15 RECOMENDAÇÕES PARA “GRENADE — BRUNO MARS” UTILIZANDO O MODELO PROPOSTO.

Rank	Música	Score
1	Just the Way You Are — Bruno Mars	43.65
2	When I Was Your Man — Bruno Mars	37.13
3	The Lazy Song — Bruno Mars	33.41
4	Payphone — Maroon 5	27.55
5	Locked Out Of Heaven — Bruno Mars	27.38
6	TiK ToK — Kesha	25.34
7	Dynamite — Taio Cruz	24.20
8	Fireflies — Owl City	23.95
9	Counting Stars — OneRepublic	23.36
10	Party In The U.S.A. — Miley Cyrus	23.04
11	Count On Me — Bruno Mars	22.95
12	Stereo Hearts (feat. Adam Levine) — Gym Class Heroes	22.61
13	I Gotta Feeling — The Black Eyed Peas	22.60
14	Marry You — Bruno Mars	22.46
15	Hey There Delilah — Plain White T's	21.94

um contexto igualmente popular, mas delimitado por um nicho mais homogêneo — o pop internacional da década de 2010.

As playlists desse nicho tendem a ser amplas e diversas, o que dilui parcialmente o peso individual de cada co-ocorrência. Isso explica a faixa de valores entre 20 e 50, preservando coerência temática.

Esse comportamento revela dois pontos fundamentais:

- O modelo distingue diferentes subníveis de popularidade, mantendo recomendações coerentes com o estilo musical consultado.
- A variação dos *scores* reflete a densidade local do grafo: quanto mais playlists reforçam a conexão, maior a confiabilidade da recomendação.

Portanto, os resultados qualitativos demonstram que o sistema preserva a estrutura semântica do espaço musical, adaptando-se tanto à popularidade global de artistas quanto aos diferentes perfis de consumo dentro de grandes gêneros.

VI. DISCUSSÃO

Embora os resultados qualitativos tenham apresentado forte coerência semântica e boa estruturação de comunidades, o sistema ainda apresenta limitações importantes, tanto no aspecto computacional quanto metodológico.

A. Limitações Computacionais

- Memória e processamento:** A base original (*Spotify Million Playlist Dataset*) é extremamente volumosa. Mesmo após o *Smart Pruning*, a construção do grafo bipartido pode ultrapassar 7 milhões de arestas, exigindo mais memória do que disponível em máquinas convencionais.
- Visualização:** A projeção para o grafo item–item gera um número massivo de conexões. Em ferramentas como Gephi, clusters ficam completamente sobrepostos, prejudicando análises visuais.

- **Tempo de execução:** A projeção item–item completa é custosa em termos de CPU, especialmente para detectar comunidades em grafos muito densos.

B. Limitações Metodológicas

- **Sparsity e Cauda Longa:** Músicas pouco presentes na base carregam pouca evidência estatística, que acaba limitando a precisão de recomendações mesmo quando elas fazem sentido temático.
- **Dependência de co-ocorrência:** O algoritmo não utiliza metadados musicais (gênero, BPM, tonalidade, embeddings de áudio). Isso torna a recomendação totalmente dependente do comportamento coletivo registrado.
- **Popularidade implícita:** Artistas mais populares acumulam peso naturalmente, podendo induzir viés para ídolos globais.

C. Trabalhos Futuros

Como perspectivas de melhoria, destacam-se:

- **Integração com embeddings musicais** (Spotify Audio Features, VGGish, CLAP), para reduzir dependência total de co-ocorrência.
- **Ponderação por confiança** para equilibrar músicas da cauda longa.
- **Normalização de playlists** ajustando pesos por tamanho e diversidade.
- **Arquiteturas híbridas** (grafos + modelos neurais) para enriquecer similaridade semântica.

A discussão reforça que, embora promissor, o sistema ainda pode ser expandido para oferecer recomendações mais robustas, especialmente para músicas pouco representadas no conjunto de dados.

VII. CONCLUSÃO

O desenvolvimento deste sistema reforça que, no domínio da recomendação musical em larga escala, o sucesso da modelagem de grafos é determinado pela qualidade estrutural dos dados. Nossa análise exploratória do *Spotify Million Playlist Dataset* identificou que a concentração de faixas na distribuição de Cauda Longa gerava um volume massivo de ruído estrutural (músicas isoladas), o que onera o processamento e compromete a detecção de padrões de gosto. Essa caracterização do dataset justificou integralmente a abordagem otimizada deste trabalho.

A principal contribuição foi demonstrar que uma arquitetura otimizada, baseada em Poda Seletiva e Ponderação IIF, permite construir sistemas de recomendação robustos em hardware restrito. Ao filtrar cerca de 50% dos dados brutos, não apenas viabilizamos a execução em 8GB de RAM, mas também melhoramos a precisão do sistema ao eliminar conexões espúrias.

Conclui-se que a topologia do grafo, quando devidamente tratada para mitigar o viés de popularidade, carrega informação semântica suficiente para agrupar usuários em comunidades coerentes (como validado pelo algoritmo de Louvain), oferecendo uma alternativa eficiente e escalável para cenários onde recursos computacionais são limitados.

REFERÊNCIAS

- [1] C.-W. Chen, P. Lamere, M. Schedl, and H. Zamani, “Recsys challenge 2018: Automatic music playlist continuation,” in *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys ’18)*. ACM, 2018, pp. 394–395.
- [2] R. Burke, “Hybrid recommender systems: Survey and experiments,” *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, December 2002.
- [3] M. E. J. Newman and M. Girvan, “Modularity and community structure in networks,” *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, June 2006.
- [4] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, October 2008.
- [5] A. Saravanou, F. Tomasi, R. Torres, and I. Miliarakis, “Multi-task learning of graph-based inductive representations of music content,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 1–8. [Online]. Available: <https://archives.ismir.net/ismir2021/paper/000075.pdf>
- [6] M. Pesek, “Hybrid music recommendation with graph neural networks,” *User Modeling and User-Adapted Interaction*, vol. 34, no. 1, pp. 1–25, 2024.
- [7] A. H. d. A. Paiva and H. F. C. Xavier, “Sistema de recomendação de músicas baseado em grafos,” Disponível em: <https://github.com/adrianpaivaa/Music-Recommender>, 2025, repositório no GitHub.