

Recomendação Musical Em Tempo Real: Algoritmo Contextualizado Para Personalização De Listas De Reprodução

Lécio Oliveira Gonçalves de Assis
Universidade Federal de São Paulo
Instituto de Ciência e Tecnologia
São José dos Campos, São Paulo 12247-014
Email: lecio.assis@unifesp.br

Resumo—Este documento apresenta a formalização do projeto de um algoritmo contextualizado para personalização de listas de reprodução, com o objetivo de proporcionar recomendações musicais mais precisas e envolventes para os usuários do Spotify. O projeto visa romper com a abordagem atual de aleatoriedade aparente e criar uma nova forma de ordenação de músicas, levando em consideração critérios como histórico de reprodução, músicas recentemente ouvidas, contexto audiovisual e popularidade. A implementação será feita em Python, utilizando a API do Spotify e bibliotecas de inteligência artificial.

Palavras-chave—Sistemas de Recomendação Musical, Algoritmo Contextualizado, Personalização de Listas de Reprodução, Plataformas de Streaming, Associação de Músicas, Contexto Audiovisual, Controle de Reprodução, Qualidade das Recomendações, Filtragem Colaborativa, Filtragem Baseada em Conteúdo.

I. INTRODUÇÃO

A música desempenha um papel fundamental na vida das pessoas e plataformas de streaming de música, como o Spotify, têm ganhado cada vez mais popularidade. No entanto, muitos usuários enfrentam problemas ao usar a funcionalidade de reprodução aleatória, pois o algoritmo subjacente não oferece uma experiência verdadeiramente aleatória. Em vez disso, o algoritmo tende a espalhar músicas do mesmo artista ou álbum ao longo da playlist, resultando em repetição constante e falta de diversidade musical. Outro problema comumente reportado está nas recomendações. Boa parte dos algoritmos são baseados em popularidade e não trazem novidades no geral ou falham em ser relevantes para aqueles que apreciam músicas fora do mainstream. Isso leva a uma experiência decepcionante para todos que desejam descobrir novas músicas e desfrutar de uma playlist variada.

II. FUNDAMENTOS

A. Recomendação de Música

A recomendação de música é um campo de estudo que visa fornecer aos usuários sugestões personalizadas de músicas com base em suas preferências e características musicais. Existem diferentes abordagens para a recomendação de música, sendo as mais comuns a filtragem colaborativa e a filtragem baseada em conteúdo.

A filtragem colaborativa utiliza o comportamento passado do usuário e de outros usuários para identificar padrões e sugerir músicas com base em interesses semelhantes. A filtragem baseada em conteúdo analisa as características musicais das faixas, como gênero, artista, ritmo, melodia e harmonia, para fazer recomendações com base em similaridade.

B. Análise de Dados Musicais

A análise de dados musicais é essencial para compreender e extrair informações relevantes das músicas. Neste projeto, utilizaremos a API do Spotify como fonte principal de informações e análise das faixas musicais. A API do Spotify fornece dados como título, artista, álbum, gênero, popularidade e características musicais, como ritmo, energia e valência. Essas informações serão utilizadas para identificar padrões, similaridades e preferências dos usuários, sem a necessidade de ferramentas adicionais de análise de áudio.

C. Aprendizado de Máquina (Filtragem Colaborativa, Agrupamento e Regras de Associação)

O aprendizado de máquina desempenha um papel fundamental na construção de modelos de recomendação. Um método comum é a filtragem colaborativa, que utiliza algoritmos de aprendizado de máquina para identificar padrões e relações entre os usuários e as músicas. Essa abordagem busca encontrar semelhanças entre os usuários com base em seus históricos de preferências e recomendar músicas com base nas preferências de usuários semelhantes.

Algoritmos de agrupamento são técnicas utilizadas para identificar grupos de usuários com preferências semelhantes e itens (músicas) com características similares. Esses algoritmos analisam os dados coletados, como características musicais ou histórico de preferências dos usuários, e os agrupam com base em sua similaridade. Essa abordagem permite criar segmentos de usuários com interesses musicais parecidos, o que é valioso para personalizar as recomendações e oferecer sugestões relevantes aos usuários.

Por sua vez, as regras de associação são utilizadas para descobrir padrões de comportamento e preferências dos usuários. Essas regras identificam associações frequentes entre itens com

base nos dados observados. Por exemplo, se muitos usuários que gostam de um determinado gênero musical também costumam ouvir um artista específico, essa associação pode ser capturada por uma regra de associação. Essas regras fornecem insights valiosos sobre as preferências dos usuários e podem ser usadas para gerar recomendações mais precisas.

D. Processamento de Linguagem Natural (NLP)

O processamento de linguagem natural (NLP) é relevante para a análise de letras de músicas e textos relacionados. Técnicas de NLP podem ser aplicadas para extrair tópicos, detectar sentimentos e gerar representações textuais que enriqueçam o perfil do usuário e melhorem a precisão das recomendações.

O uso de NLP poderá ser explorado por permitir considerar não apenas as características musicais, mas também as preferências e interesses expressos pelos usuários em texto e levar em consideração as letras das músicas.

E. Bibliotecas e Tecnologias

O projeto faz uso de várias bibliotecas e tecnologias para implementar as funcionalidades propostas. Além da API do Spotify, que fornece acesso aos dados musicais, são utilizadas as seguintes ferramentas:

- PySpark [4]: uma biblioteca para processamento distribuído de dados, que permite lidar com grandes volumes de informações de forma eficiente.
- PyAudioAnalysis [5]: uma biblioteca em Python para análise de áudio, que possibilita a extração de características musicais relevantes.
- ImageAI [6]: uma biblioteca em Python para análise de dados de vídeo, que permite identificar objetos, reconhecer padrões e analisar conteúdo visual.
- TensorFlow [7]: uma biblioteca popular de aprendizado de máquina e deep learning, amplamente utilizada para construir e treinar modelos de recomendação.
- scikit-learn [8]: uma biblioteca em Python para aprendizado de máquina, que fornece uma ampla gama de algoritmos e ferramentas para a construção de modelos de recomendação e análise de dados.

Essas tecnologias e bibliotecas são essenciais para implementar as funcionalidades do sistema de recomendação de música proposto, permitindo o processamento e a análise eficiente dos dados musicais e de vídeo, a construção e treinamento dos modelos de recomendação. Ao longo do projeto é provável a utilização de ferramentas alternativas, dependendo do direcionamento que as explorações tomarem.

III. TRABALHOS RELACIONADOS

Visando facilitar a abordagem aos diversos trabalhos sobre sistemas de recomendação, a literatura será dividida em dois grupos. O primeiro grupo consiste em trabalhos que abordam conceitos básicos, soluções estabelecidas, problemas, desafios e o contexto histórico dos sistemas de recomendação. O segundo grupo abrange trabalhos que apresentam possíveis soluções, novos desafios, uma evolução em relação a modelos

anteriores e análises desses modelos. O projeto atual se enquadra na categoria de literatura experimental, pois busca apresentar uma nova abordagem e soluções para desafios existentes. Os trabalhos citados a seguir fornecem uma base sólida de conhecimento teórico e exploram soluções inovadoras no campo de sistemas de recomendação de músicas.

A. Literatura Conceitual

[9] discute os tipos de sistemas de recomendação, como "lean-in" e "lean-back", e apresenta os desafios associados a esses conceitos.

[10] explora diferentes modelos de recomendação, incluindo modelos baseados em contexto e emoção, destacando os potenciais problemas e desafios dessas abordagens.

[11] trata especificamente do quão injusto é o viés de popularidade no contexto dos sistemas de recomendação.

[12] aborda a playlist adaptativa e os critérios de qualidade que podem ser considerados nesse tipo de recomendação.

[13] discute métodos de recomendação, como filtragem colaborativa, filtragem baseada em conteúdo, filtragem híbrida e outros, fornecendo uma visão geral dos conceitos por trás dessas abordagens.

[14] destaca os maiores desafios enfrentados pelos sistemas de recomendação, levando em consideração as limitações e questões em aberto na área.

[15] apresenta conceitos relevantes no contexto de sistemas de recomendação.

B. Literatura Experimental

[16] propõe um modelo de individualização utilizando big data como fonte de informação.

[17] propõe uma abordagem híbrida que busca trazer diversidade e evitar o problema do "cold-start" na recomendação de músicas.

[18] apresenta vários problemas identificados em uma abordagem anterior e propõe um modelo aprimorado para superar essas limitações.

IV. OBJETIVO

O objetivo deste projeto é desenvolver um algoritmo contextualizado para personalização de listas de reprodução no Spotify, que leve em consideração diferentes critérios, como histórico de músicas ouvidas, músicas recentemente ouvidas, contexto audiovisual e popularidade. O algoritmo buscará proporcionar recomendações musicais mais precisas e envolventes, equilibrando a familiaridade com a descoberta de novas músicas, a fim de oferecer uma experiência musical personalizada e satisfatória para os usuários, resolvendo ambos os problemas citados anteriormente.

V. METODOLOGIA EXPERIMENTAL

A. Base de Dados

1) *Histórico Ampliado de Streaming*: Esses dados são solicitados diretamente na sessão de privacidade do Spotify. Possui o histórico de streaming anual do usuário. Possui vários detalhes sobre cada streaming:

- Data e hora, no formato UTC (Tempo Universal Coordenado), do fim do último streaming.
- Seu nome de usuário do Spotify.
- Plataforma usada para tocar a faixa (por exemplo, Android OS, Google Chromecast).
- Por quantos milésimos de segundos você ouviu a faixa.
- Código do país onde a faixa foi ouvida.
- Endereço IP usado para tocar a faixa.
- Agente do usuário utilizado para tocar a faixa (por exemplo, um navegador, como o Mozilla Firefox ou o Safari).
- Nome da faixa.
- Nome do artista, banda ou podcast.
- Nome do álbum da faixa.
- Uma URI de faixa do Spotify, que identifica apenas essa música.
- Nome do episódio do podcast.
- Nome do programa do podcast.
- Uma URI de episódio do Spotify, que identifica apenas esse episódio do podcast.
- Por que a faixa começou (por exemplo, a faixa anterior terminou ou você escolheu na playlist).
- Por que a faixa terminou (por exemplo, a faixa tocou até o fim ou você clicou em “Avançar”).
- Se a ordem aleatória foi usada quando a faixa estava tocando.
- Se o usuário pulou para a próxima música.
- Se a faixa foi ouvida no modo offline.
- Marcação de tempo de quando e se o modo offline foi usado.
- Se a faixa foi ouvida em uma sessão particular. [1]

Estes dados foram solicitados por alguns usuários selecionados, com um período de até 30 dias para retorno a partir da solicitação. Mais amostras serão adquiridas, de forma a ampliar estas amostras para treinamento.

2) *Análise de Áudio da Faixa*: “Obtenha uma análise de áudio de baixo nível para uma faixa no catálogo do Spotify. A análise de áudio descreve a estrutura da faixa e o conteúdo musical, incluindo ritmo, altura e timbre.” [2] Cada faixa possui 6 sessões de informações diferentes relacionadas ao áudio, a mais relevante é chamada “track”. Se trata de uma análise relativamente menos técnica que as outras e fornecer, de forma simplificada, os mesmos dados. Portanto será a única a ser utilizada em qualquer situação. Estas análises podem ser obtidas em forma de requisições na API do Spotify, porém como forma de simplificar o processo e poder analisar o conjunto como um todo, será usada uma amostra [3], referentes a músicas lançadas entre 1921 e 2021. Ela possui aproximadamente 600 mil músicas, sendo uma representação de 0.5% de todas as faixas disponíveis na plataforma. Também serão feitas requisições de forma inteligente, de acordo com os conhecimentos adquiridos na amostra e através da requisição de outros dados, como Top Artistas, Top Músicas e Itens favoritos.

3) *Músicas Tocadas Recentemente*: Neste endpoint será obtido o histórico recente de streaming do usuário, limitado a

50 itens. Seu uso principal será na aplicação do modelo e não no treinamento ou teste. Poderá também ser a fonte de um histórico ampliado artificial, armazenando o histórico recente ao longo do tempo, tornando o modelo mais específico para o usuário.

4) *Recomendações*: Este endpoint da API será usado como forma de validação das recomendações obtidas com o modelo. Aqui temos os mesmos parâmetros de tolerância relativos aos encontrados na Análise de Áudio da Faixa e Artistas, Gêneros e Músicas de origem.

B. Coleta de Dados

A coleta de dados será realizada por meio da API de desenvolvimento do Spotify. A API fornecerá acesso a informações relevantes relacionadas ao usuário, como histórico de reprodução, e relacionadas às músicas, como a análise aprofundada do áudio, detalhes sobre o álbum e o artista e índices de popularidade. Esses dados serão essenciais para alimentar o algoritmo de recomendação e fornecer insights sobre as preferências musicais dos usuários.

C. Pré-processamento de Dados

Os dados coletados da API do Spotify passarão por um processo de pré-processamento para remover informações irrelevantes, tratar dados ausentes e normalizar formatos. Serão aplicadas técnicas de limpeza de dados, como remoção de duplicatas, tratamento de valores nulos e padronização de atributos, garantindo que os dados sejam consistentes e adequados para uso no algoritmo de recomendação. Devido à grande quantidade de dados existentes, essa etapa será fundamental para que os objetivos sejam alcançados, utilizando o PySpark como ferramenta para lidar com grandes volumes de informações de forma eficiente.

D. Desenvolvimento da Recomendação

O algoritmo de recomendação será desenvolvido em Python, utilizando bibliotecas e frameworks adequados, como scikit-learn e TensorFlow. Serão exploradas técnicas de aprendizado de máquina, como filtragem colaborativa, agrupamento, busca de informação, regras de associação e processamento de linguagem natural, para identificar padrões nos dados e realizar recomendações personalizadas com base nos critérios estabelecidos. O algoritmo levará em consideração o histórico de músicas ouvidas, músicas recentemente ouvidas, contexto audiovisual e popularidade para criar listas de reprodução mais coesas e atraentes para os usuários.

E. Integração do Contexto Audiovisual

Uma inovação deste projeto é a incorporação do contexto audiovisual externo ao algoritmo. Por meio da captura de imagens e vídeos de um dispositivo, o algoritmo terá a capacidade de “ver” e “ouvir”, permitindo a distinção de elementos presentes no ambiente. Essa percepção audiovisual adiciona uma camada adicional de informações ao processo de seleção de músicas, melhorando ainda mais a qualidade das recomendações musicais. Para a execução deste processo

serão exploradas bibliotecas para análise de áudio, como PyAudioAnalysis, e bibliotecas para interpretação de imagens, como ImageAI, sendo que ambas já implementam diferentes modelos de inteligência artificial.

F. Interface de Usuário

Será desenvolvida uma interface de usuário que permitirá aos usuários fazer login com sua conta do Spotify e conceder as permissões necessárias para uma experiência personalizada. A interface também possibilitará que os usuários interajam com a lista de reprodução recomendada, fornecendo feedback sobre as músicas sugeridas e permitindo alterações na lista de reprodução de acordo com suas preferências. O desenvolvimento de uma interface gráfica não possui prioridade dentre os objetivos, ao longo do projeto decisões serão tomadas quando a sua implementação, sendo essencial apenas uma interface de linha de comando.

VI. RESULTADOS

Espera-se que o algoritmo contextualizado para personalização de listas de reprodução proporcione recomendações musicais mais precisas e envolventes para os usuários do Spotify. Ao considerar critérios como histórico de músicas ouvidas, popularidade, contexto audiovisual e parâmetros adicionais, o algoritmo buscará equilibrar familiaridade e descoberta, promovendo a satisfação do usuário e elevando a reprodução de músicas a um novo patamar de qualidade. O algoritmo também busca evitar problemas comuns como o “cold-start” e uma demanda computacional excessiva, uma vez que a próxima música deve ser definida antes do final da música atual. A avaliação da qualidade das recomendações será feita por meio de testes e coleta de feedback dos usuários. A validação do modelo será obtida analisando as recomendações da própria plataforma de streaming. Será verificado o motivo das semelhanças e diferenças, buscando acertar onde a plataforma acerta e mostrar melhor desempenho do que as falhas do algoritmo existente.

VII. CONCLUSÃO

O projeto de desenvolvimento de um algoritmo contextualizado para personalização de listas de reprodução no Spotify representa uma abordagem inovadora, centrada no usuário e focada em proporcionar uma experiência musical mais coesa, envolvente e personalizada. Ao considerar critérios como histórico de músicas ouvidas, popularidade, contexto audiovisual e parâmetros adicionais, o algoritmo busca equilibrar familiaridade e descoberta, promovendo a satisfação do usuário e elevando a reprodução de músicas a um novo patamar de qualidade. A implementação em Python, utilizando a API do Spotify e técnicas de inteligência artificial, permitirá a criação de recomendações mais precisas e relevantes. Com esse projeto, busca-se aprimorar a qualidade das recomendações musicais e proporcionar uma experiência única aos usuários do Spotify.

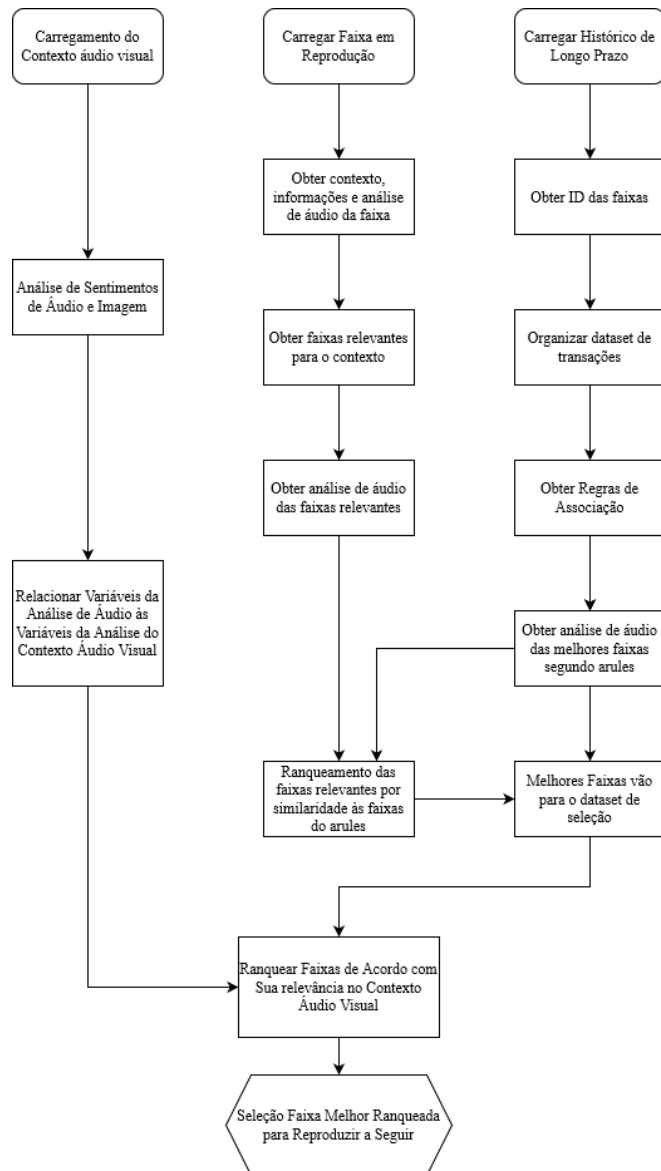


Figura 1. Diagrama de Blocos do Pipeline Experimental. Fonte: Autor

REFERÊNCIAS

- [1] Como Interpretar Meus Dados. Disponível em <https://support.spotify.com/br-pt/article/understanding-my-data/>
- [2] Análise de Áudio da Faixa. Disponível em <https://developer.spotify.com/documentation/web-api/reference/get-audio-analysis>
- [3] Spotify Dataset 1921-2020, 600k+ Tracks. Disponível em https://www.kaggle.com/datasets/yamaerenay/spotify-dataset-19212020-600k-tracks?resource=download&select=dict_artists.json
- [4] Apache Spark. Disponível em: <https://github.com/apache/spark/tree/master/python>.
- [5] pyAudioAnalysis. Disponível em: <https://github.com/tyiannak/pyAudioAnalysis>.
- [6] ImageAI. Disponível em: <https://github.com/OlafenwaMoses/ImageAI>.
- [7] TensorFlow. Disponível em: <https://www.tensorflow.org/?hl=pt-br>.
- [8] scikit-learn. Disponível em: <https://scikit-learn.org/stable/>.
- [9] Schedl, M., Knees, P., McFee, B. & Bogdanov, D. Music recommendation systems: Techniques, use cases, and challenges. *Recommender Systems Handbook*. pp. 927-971 (2021)

- [10] Song, Y., Dixon, S. & Pearce, M. A survey of music recommendation systems and future perspectives. *9th International Symposium On Computer Music Modeling And Retrieval*. **4** pp. 395-410 (2012)
- [11] Kowald, D., Schedl, M. & Lex, E. The unfairness of popularity bias in music recommendation: A reproducibility study. *Advances In Information Retrieval: 42nd European Conference On IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II* **42**. pp. 35-42 (2020)
- [12] Jannach, D., Kamehkhosh, I. & Bonnin, G. Music Recommendations. (2018)
- [13] Qin, Y. A historical survey of music recommendation systems: Towards evaluation. (McGill University (Canada),2013)
- [14] Schedl, M., Zamani, H., Chen, C., Deldjoo, Y. & Elahi, M. Current challenges and visions in music recommender systems research. *International Journal Of Multimedia Information Retrieval*. **7** pp. 95-116 (2018)
- [15] Zhang, Q., Lu, J. & Jin, Y. Artificial intelligence in recommender systems. *Complex & Intelligent Systems*. **7** pp. 439-457 (2021)
- [16] Sun, P. Music Individualization Recommendation System Based on Big Data Analysis. *Computational Intelligence And Neuroscience*. **2022** (2022)
- [17] Kaufman, J. A Hybrid Approach to Music Recommendation: Exploiting Collaborative Music Tags and Acoustic Features. (2014)
- [18] Hu, Y. A model-based music recommendation system for individual users and implicit user groups. (University of Miami,2014)