****

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SERGIPE - CAMPUS LAGARTO**

**SISTEMA DE RECOMENDAÇÃO DE MÚSICA BASEADO NOS GÊNEROS MUSICAIS MAIS ESCUTADOS PELO USUÁRIO**

**Kaiki Mello dos Santos**

***Orientador: João Paulo Dias de Almeida***

***Coorientadora: Catuxe Varjão de Santana Oliveira***

**Trabalho de Conclusão de Curso**

**COORDENAÇÃO DO BACHARELADO EM SISTEMAS DE INFORMAÇÃO**



**LAGARTO - SERGIPE**

**ABRIL/2023**

**RESUMO**

A disponibilidade da música digital atualmente é abundante quando comparada ao passado. Os serviços de streaming estão em ascensão, permitindo que músicas possam ser acessadas de diferentes dispositivos (e.g. tablet, computador, celular) em qualquer local, online ou offline. Esse excesso de informação musical causa alguns efeitos colaterais, como a dificuldade na classificação, busca e organização dessa grande quantidade de músicas. Desta maneira, é de grande utilidade o desenvolvimento de um sistema de recomendação de músicas que possa sugerir músicas semelhantes ao perfil do usuário. Os sistemas de recomendação baseiam-se no histórico de preferências do usuário e em seu perfil para prever e recomendar músicas adequadas aos indivíduos levando em conta a semelhança das características das faixas. Assim, esse trabalho tem como objetivo desenvolver um sistema de recomendação de músicas que possa fornecer recomendações tendo como base a semelhança dos metadados que descrevem as mídias e visa utilizar a interação do indivíduo com o software para gerar melhores recomendações personalizadas para cada usuário.

**Palavras-chaves:** Sistemas de recomendação; Recomendação musical; Músicas.

**1 INTRODUÇÃO**

A música é um dos elementos culturais mais antigos da sociedade e está presente em todo mundo em diferentes culturas. Seu conteúdo pode impactar, gerar debates e formar opiniões na vida dos consumidores. Segundo Silva e Domingues (2022), a música está relacionada com a formação do indivíduo e não é meramente uma forma de passatempo. O processo de compor e produzir canções busca criar melodias e arranjos que transmitam sentimentos e sensações que conectem o artista a seu público através da música. Adiyansjah, Gunawan e Suhartono (2019) consideram a música como trabalho criativo humano que busca manifestar ideias e emoções através de sons, e esses podem ser classificados em diversos gêneros (e.g. pop, jazz, forró).

O mercado musical nunca esteve tão imerso no ramo digital quanto atualmente, desde antes do surgimento do MP3 (formalmente MPEG-1 Audio Layer III ou MPEG-2 Audio Layer III) a indústria fonográfica vem se adaptando aos novos meios de criação, composição, consumo e distribuição de obras musicais. A popularização ágil da música em formato digital proporcionou o surgimento dos serviços de streaming de música como Spotify, Tidal, Apple Music entre outros. Estes vêm se destacando por motivarem cada vez mais ganhos no faturamento do setor fonográfico. De acordo com relatório publicado em 2021 pela Federação Internacional da Indústria Fonográfica (IFPI - International Federation of the Phonographic Industry), o comércio global de músicas registrou crescimento de 7,4% em 2020, onde esse valor em receitas totais no mesmo ano foram de 21,6 bilhões de dólares, sendo esse crescimento impulsionado pelo streaming.

A disponibilidade da música na era digital é abundante nos serviços de streaming. Nunca foi tão fácil ouvir música como nos dias atuais em virtude da facilidade de acesso desses serviços através dos diversos dispositivos (e.g. tablet, computador, celular), de qualquer local, online ou offline. A facilidade de acesso a todo esse conteúdo trouxe um novo problema para o usuário, visto que o mesmo, se encontra com dificuldades para decidir o que consumir. Além disso, a simplicidade de acesso e sobrecarga de informação dificulta a descoberta de novas músicas (JORDÃO, 2016). Sendo assim, esse excesso de informação musical causa alguns efeitos colaterais, como dificuldade na classificação, busca e organização dessa grande quantidade de músicas. Então organizar todo esse aglomerado de música é muito custoso e causa fadiga de informação (ADIYANSJAH; GUNAWAN; SUHARTONO, 2019). Desta maneira, é de grande utilidade o desenvolvimento de um sistema de recomendação de músicas que possa sugerir músicas semelhantes ao perfil do usuário.

Segundo Ricci et al. (2011) os sistemas de recomendação são ferramentas e técnicas que provêm sugestão de itens para os clientes. Esses sistemas baseiam-se no histórico de preferências do usuário e em seu perfil para prever e ofertar itens (e.g. músicas, vídeos, produtos) adequadas aos indivíduos levando em conta a semelhança das características para surpreender o usuário com itens que atendam suas necessidades no momento, ou para facilitar a utilização dos serviços. Os sistemas de recomendação encontram sugestões com base na comparação entre itens e/ou usuários, filtram as informações relevantes ordenando itens de acordo com a preferência dos usuários, viabilizando assim a tomada de decisão (SILVA, 2021).

Ao considerar essa problemática acerca da dificuldade que o processo de seleção de músicas gera ao usuário diante da abundância do acervo digital de canções disponíveis, o presente trabalho objetiva o desenvolvimento de um sistema de recomendação de músicas que possa fornecer recomendações tendo como base a semelhança dos metadados que descrevem as mídias e visa utilizar a interação do indivíduo com o software para gerar melhores recomendações personalizadas para cada usuário.

* 1. **JUSTIFICATIVA**

Tendo em vista a evolução tecnológica do mercado fonográfico, a ascensão dos serviços web e de streaming, e a grande quantidade de acervo digital disponível nestas plataformas, os sistemas de recomendação estão cada vez mais presentes no nosso dia-a-dia para que o acesso a estes recursos seja realizado de maneira facilitada. Desta forma, esse projeto irá desenvolver um sistema de recomendação que classifique as informações disponíveis, para facilitar o processo de seleção de músicas pelo usuário diante da abundância de canções disponíveis. O surgimento desses serviços causa uma sobrecarga de informação para o utilizador devido ao amontoado de dados disponível e sua vasta gama de opções. Com essa diversidade de serviços à disposição, buscar novos itens exige esforço e atenção, podendo levar o usuário a tomar decisões por itens que não o agradem (SILVA, 2021).

Os problemas relacionados ao fato da abundância de músicas na era digital vão desde a dificuldade de o sujeito escolher as canções, encontrar novos artistas que se encaixem no seu perfil em meio a esse emaranhado de obras até a demora na busca do conteúdo com informações relevantes para o seu consumo.

Dessa forma, é possível notar que o projeto de desenvolver um sistema de recomendação musical pode impactar diretamente na experiência do usuário, pois este irá despender menos tempo para escolher músicas. Além de que, os sistemas de recomendação em alguns ramos podem ser bastante lucrativos quando efetivos, ou então, um modo de se destacar consideravelmente das plataformas concorrentes (ROCCA, B., 2019; ROCCA, J., 2019).

**1.2 OBJETIVO GERAL**

Este trabalho tem como objetivo implementar um sistema de recomendação de músicas, tendo como base a análise da similaridade dos metadados que descrevem as faixas mais apreciadas pelo usuário. Além disso, o objetivo visa utilizar a interação do indivíduo com o software para gerar melhores recomendações, sendo estas personalizadas para cada usuário.

**1.3 OBJETIVO ESPECÍFICOS**

* Realizar uma revisão da literatura para compreender o estado da arte de sistemas de recomendação no contexto do ramo de músicas.
* Selecionar técnicas e tecnologias para o desenvolvimento de um sistema de recomendação de músicas.
* Definir quais informações compõem o perfil do usuário.
* Estabelecer critérios e métricas para a recomendação de músicas baseadas no perfil do usuário.
* Construir um de sistema de recomendação de músicas que utilize os critérios e métricas estabelecidos para recomendar faixas aos usuários.

**2 METODOLOGIA**

Este estudo propõe o desenvolvimento de um sistema de recomendação de músicas faz uso dos metadados que descrevem as faixas e das interações do usuário com o sistema para sugerir músicas de forma personalizada ao usuário. Considerando o desenvolvimento do software quanto a natureza, a metodologia é aplicada.

Além disso, este trabalho segue a metodologia de pesquisa quantitativa que de acordo com Apuke (2017) é um método de pesquisa que lida com análise de variáveis e de dados usando técnicas estatísticas para obter resultados e responder perguntas. Serão definidas e utilizadas métricas para quantificar o desempenho do sistema de recomendação e do algoritmo de classificação.

Em busca de se familiarizar com os assuntos que são objeto deste trabalho foi realizada uma pesquisa bibliográfica, no que se refere ao conhecimento sobre sistemas de recomendação e sugestão de músicas. As fontes utilizadas na pesquisa foram livros, sites, artigos científicos disponíveis no Google Scholar, realizadas entre maio e junho de 2022.

Os próximos passos para realizar o desenvolvimento desse estudo estão listados nos tópicos metodológicos apresentados a seguir:

* **Definição do Dataset:** os Datasets são conjuntos de dados específicos que servem de amostras para treinamento de algoritmos de Inteligência Artificial e recomendação, nessa etapa será selecionado um ou mais Datasets que possam ser utilizados no processo de desenvolvimento do sistema de recomendação.
* **Limpeza e análise do Dataset:** após a seleção do Dataset serão aplicadas técnicas de limpeza dos dados visando remover possíveis ruídos (e.g. outliers, valores nulos ou duplicados) que prejudiquem no processo de análise do Dataset, em seguida será feita uma análise do Dataset.
* **Modelagem do sistema de recomendação:** por conseguinte será iniciado o processo de modelagem e desenvolvimento do sistema de recomendação.
* **Avaliação do sistema de recomendação:** nesta etapa serão definidas métricas que serão utilizadas para aferir o desempenho do sistema de recomendação.
* **Treinamento de algoritmos de classificação:** os algoritmos de classificação precisam ser ajustados de acordo com os dados, eles precisam ser treinados, para entender os padrões e poder classifica-los.
* **Personalização do sistema de recomendação:** processo de personalização do sistema de recomendação com o algoritmo de classificação treinado.
* **Avaliação do sistema personalizado:** nessa etapa serão utilizadas métricas para aferir o desempenho do sistema de recomendação após a personalização.

**2.1 DATASETS**

A

**3 ALGORITMOS**

Nesta seção são apresentados os algoritmos desenvolvidos para o sistema de recomendação híbrido. Foram combinadas duas estratégias de recomendação (baseado em conteúdo e colaborativo) com o algoritmo de classificação vizinho mais próximo (*Nearest Neighbor* (k-NN)).O conjunto desses algoritmos irão resultar no sistema de recomendação híbrido final (ReKomeNNdador Hydra).

* 1. **ANÁLISE DO CONTEÚDO**

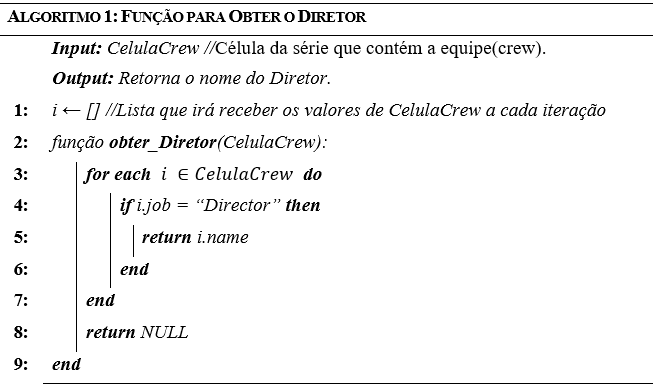
Inicialmente para customizar as recomendações foi criado um mecanismo que calcula a semelhança dos filmes com base nos atributos: elenco, equipe, palavras-chave, e gênero. Estes atributos são utilizados para sugerir filmes similares a um determinado filme que o usuário gostou. Ao usar os metadados dos filmes (conteúdo) para construir esse mecanismo, isso é conhecido como filtragem baseada em conteúdo. Para iniciar a implementação desse mecanismo foram lidos os datasets e criada a variável *ds\_links* (que contém os dados de *links\_small*) que será usada no k-NN.

Durante a análise do conteúdo dos filmes foi realizada uma etapa de pré-processamento dos dados, nessa etapa foi percebido a necessidade de excluir algumas linhas que possuíam dados mal formatados (ex. linhas do dataframe onde na coluna id foi inserido data, além de valores ausentes nos demais atributos), criar colunas ano e organizar dados em formato de lista da coluna de gênero além de remover dados duplicados presentes no dataframe de filmes, como haviam linhas inteiras duplicadas e linhas que só se diferenciavam pela popularidade, foram considerados os dados dos filmes com maior popularidade.

Para implementação deste algoritmo, uniu-se o dataframe filmes (que contém a coluna gêneros) com os dataframes de créditos (contém o elenco e a equipe) e palavras-chave do dataframe de palavras-chave (keywords). Após isso é criada a variável *smd* (similar movie metadata) que será o principal dataframe utilizado nessa etapa. *SMD* (similar movie metadata) é um subconjunto de dados do *md\_filmes* (movie data filmes) filtrado pelo conjunto de dados do dataframe links\_small, também é criado dataframe ds\_filmes que será utilizado na implementação do kNN.

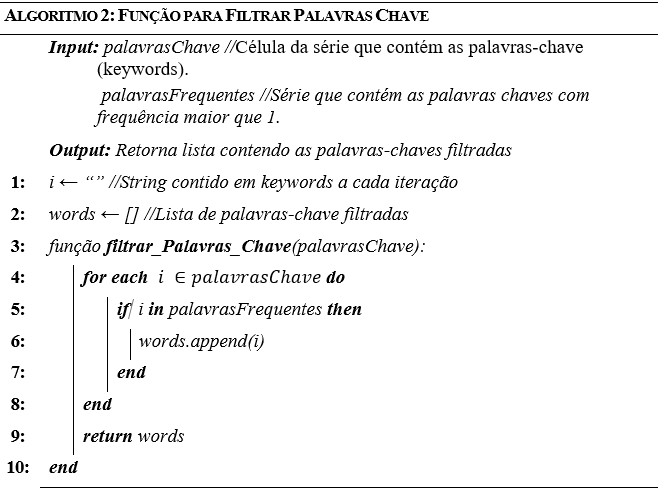
Com todos esses dados em um dataframe foi definido que do atributo equipe (crew) será utilizado apenas informações relacionadas ao diretor, já que os outros não contribuem muito para a impressão do filme (ex. Steven Spielberg, Christopher Nolan, Tim Burton). No caso do atributo elenco (cast) foram selecionados os três principais atores. Logo após, é aplicado nas colunas a função *literal\_eval,* que visa avaliar qualquer string passada para ela e converte em seu objeto python correspondente.

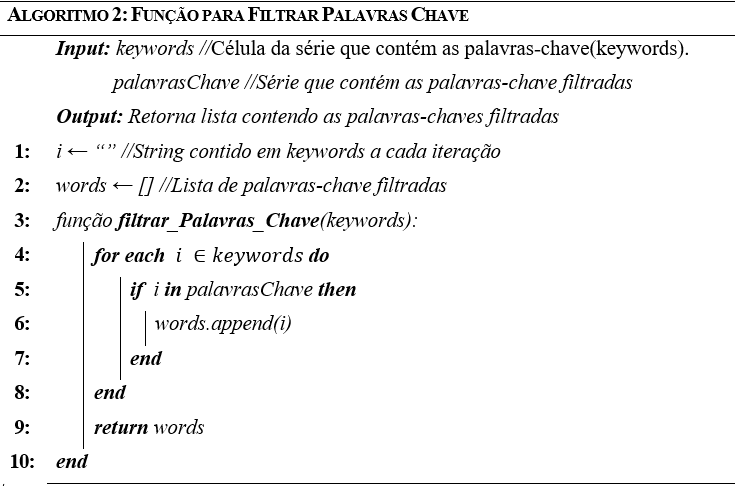
Em seguida é criada a função (*obter\_diretor*) para obter o nome dos diretores que constam na coluna crew e cria uma nova coluna “director” que vai receber esse dado. Posteriormente, os nomes dos atores que constam no elenco são organizados e são selecionados os três principais atores.



A abordagem utilizada é criar um despejo (dump) de metadados que consiste em gêneros, diretor, atores principais e palavras-chave para cada filme. Aplicou-se remoção dos espaços e conversão para minúsculas das características diretor, atores principais e palavras-chave, além de mencionar o diretor três vezes para obter maior peso em relação a todo o elenco, pretendendo aumentar a similaridade de filmes com o mesmo diretor.

Seguidamente é realizado um pré-processamento nas palavras chaves antes de colocá-las em uso para o cálculo da similaridade cosseno, calculada a contagem de frequência de cada palavra e é criada uma série que contém essas informações. As palavras-chave ocorrem em frequências que variam de 1 a 610, palavras que ocorrem uma única vez não tem utilidade e podem ser removidas com segurança resultando numa série contendo as palavras-chaves filtradas. Então é aplicada a função *filtrar\_palavras\_chave* na coluna de palavras-chave do dataframe *smd* visando selecionar apenas palavras que estejam na série filtrada anteriormente.





Logo depois as palavras passam pelo processo de Stemming onde serão convertidas em seu radical para que palavras como Cats e Cat sejam consideradas iguais e para isso é feito o uso do objeto *SnowballStemmer*. E então é criada uma coluna contendo a sopa de palavras (*bag of words*) que é a união das colunas palavras-chave, elenco, diretor e gênero. Em seguida é usado o *CountVectorizer* da biblioteca do Scikit-Learn para criar uma matriz de contagem de tokens, como está descrito no algoritmo abaixo:



Os parâmetro utilizados no *CountVectorizer* são: *analyzer* informando que o recurso deve ser feito de n-grams de palavras, *ngram\_range* se refere ao intervalo de n-gramas do texto que será incluído na sopa de palavras, nesse caso são unigramas e bigramas, *min\_df* que é usado para remover palavras que aparecem com pouca frequência e *stop\_words* que irá remover palavras de parada da língua inglesa dos tokens resultantes (ex. "a", "the", "this", "and")*.*

Em seguida é aplicada a função *fit\_transform* que irá aprender o dicionário de vocabulário e retornar uma matriz documento-termo, posteriormente é calculada a similaridade cosseno e criada uma série chamada de *indices* que contém os títulos e seus respectivos índices no dataframe *smd*, essa série será utilizada para buscar o id do filme através de seu título.

* 1. **ANÁLISE COLABORATIVA**

O algoritmo baseado em conteúdo sofre a limitação de somente ser capaz de sugerir filmes *próximos* a um outro determinado filme, ou seja, não é capaz de capturar gostos e preconceitos pessoais de um usuário, o que torna essa técnica um mecanismo impessoal. Qualquer usuário independente de quem seja, que usar esse mecanismo, irá obter as mesmas recomendações caso realize uma consulta para um determinado filme.

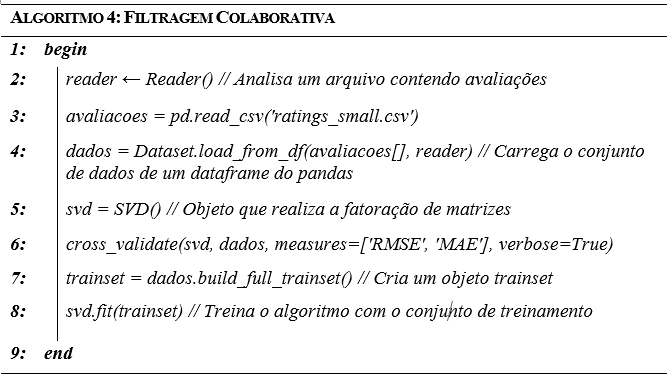
Desse modo, foi implementado um algoritmo que usa a técnica de *Filtragem Colaborativa* para oferecer recomendações mais customizadas e pessoais para o usuário. Essa técnica é baseada na ideia de que indivíduos semelhantes a um usuário, podem ser usados para prever o quanto esse usuário vai gostar de um determinado produto (ex. os filmes) ou serviço que esses indivíduos consumiram/experimentaram, mas o usuário não.

**Figura 1:** Funcionamento da Filtragem Colaborativa baseada em usuários



A implementação do recomendador baseado na técnica de *Filtragem Colaborativa* foi feita utilizando a biblioteca *Surprise* que usou algoritmos como *SVD* (Singular Value Decomposition - Decomposição de Valor Singular) para minimizar o *RMSE* (Root Mean Square Error - Raiz do Erro Quadrático Médio) e fornecer recomendações com maior precisão.

A princípio foi criado um objeto *Reader* que é usado para analisar um arquivo contendo avaliações, e que em resumo define as configurações de leitura dos dados de entrada para o algoritmo de recomendação. Logo após, o código carrega um conjunto de dados de avaliações dos filmes usando a biblioteca Pandas e criado o dataframe ds\_avaliações que será usado na implementação do kNN.



Em seguida é feito o uso da biblioteca Surprise para carregar esses dados em um objeto *Dataset* fazendo o uso da função *load\_from\_df,* que carrega um conjunto de dados de um dataframe pandas, onde são passados como parâmetros as colunas que contém as informações relevantes do dataframe *avaliações*  (‘user\_id’, ‘movie\_id’, ‘rating’) e o objeto *Reader* para que a biblioteca possa ler corretamente os dados do dataframe.

Logo após é criado um objeto *SVD,* que é usado para realizar a fatoração de matrizes, e utilizada função *cross\_validate,* que execute um procedimento de validação cruzada para um determinado algoritmo, passando como parâmetros o objeto *SVD* (algoritmo a ser avaliado)*,* os dados, as medidas (as medidas de desempenho a serem computadas) e a flag verdadeiro para o atributo verbose, essa função é usada para avaliar o desempenho do modelo com base nas métricas de erro *RMSE (Root Mean Square Error - Raiz do erro quadrático médio) e MAE (Mean Absolute Error - Erro Absoluto Médio)*.

Por fim é feito o uso da função *build\_full\_trainset* para criar um objeto *trainset* que contém todos os dados disponíveis de treinamento, logo após o modelo é treinado com esse conjunto de dados completo usando a função *SVD.fit* que treina o modelo de recomendação usando a técnica de decomposição de valores singulares (SVD).

* 1. **NEAREST NEIGHBOR (k-NN) - VIZINHOS MAIS PRÓXIMOS**

Em adicional foi desenvolvido o mecanismo de recomendação usando o K-NN (K - Nearest Neighbors ou K - Vizinhos mais Próximos) que usa a técnica de filtragem colaborativa baseada em usuários. Que tem por objetivo realizar a recomendação de filmes para um usuário específico com base nas avaliações de outros usuários. Esse algoritmo faz uso de alguns dataframes criados nas seções anteriores e que são passados como parâmetros da função.

Para implementação desse mecanismo são verificados se os identificadores *id* e *imdb\_ids* são nulos no *ds\_filmes* (criada na análise de conteúdo que contém o dataset de filmes), em seguida é extraída a coluna imdbId e convertida para o tipo de dado inteiro e então feito o merge com *ds\_links* (criada na análise de conteúdo)*,* além disso são selecionadas as colunas *movieId* e *title* do *ds\_filmes* e do ds\_avaliacoes (dataset de avaliações criado na análise colaborativa) foram selecionadas as colunas *userId*, *movieId* e *rating.*

É realizado um merge entre *ds\_filmes* e *ds\_avaliações* usando a coluna de *movieId* e então realizado um pivot no dataframe onde os dados ficam distribuídos no formato de uma matriz onde as linhas são os usuários, as colunas são os títulos dos filmes e os valores são as avaliações dadas pelos usuários aos filmes, e os valores NA ou NaN(Not Avaliable e Not a Number respectivamente) são substituídos pela nota zero. Em seguida é instanciado o modelo de vizinhos mais próximos *Nearest Neighbors -* que implementa o aprendizado não supervisionado dos vizinhos mais próximos - como mostrado abaixo.

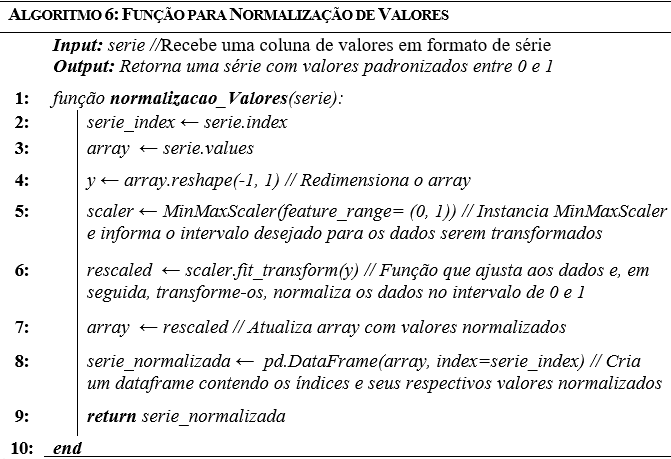


Por conseguinte é passado como hiperparâmetro a métrica de distância do cosseno, em seguida é buscado o índice do usuário no *df\_recommender* que passa pelo *reshape* que reformula o formato da série retornada para um Array 2D pois é isso que a função *kneighbors* espera, logo após é definida a quantidade de vizinhos a serem avaliados que no caso do algoritmo são quatro e então é executada a função *kneighbors* que irá encontrar os vizinhos mais próximos do usuário com base nas avaliações dadas pelos outros usuários.

A função *kneighbors* retorna dois arrays um contendo as distâncias dos vizinhos mais próximos e o outro contendo os índices desses vizinhos, a seguir é criada uma lista de dataframes, onde cada dataframe é composto pelos títulos de filmes e as avaliações dadas pelo usuário e os vizinhos mais próximos. Posteriormente é feito uma mescla entre o dataframe do usuário e seus vizinhos mais próximos utilizando a função *reduce* do módulo *functools* que aplica uma função de dois argumentos cumulativamente aos itens da sequência.

Logo depois, a variável ds\_titulos (dataset títulos) que foi criada após a aplicação do *reduce*,é ordenada de forma decrescente pelos vizinhos mais próximos e então é feito um filtro de todos os filmes que os vizinhos mais próximos assistiram e avaliaram e o usuário não. Após isso, é criada a coluna de média que em *ds\_titulos* que é calculada através da soma das avaliações de cada vizinho dividida pela quantidade de vizinhos mais próximos.

A seguir é resetado o índice do *ds\_titulos* (que antes eram os títulos dos filmes) para inteiros e os títulos passam a ser uma coluna, após isso é realizada a normalização dos valores da média para que os valores fiquem entre 0 e 1 conforme a função *normalizacao\_valores*. Essa função faz uso do *MinMaxScaler* do Scikit Learn que irá transformar os valores dimensionando cada valor para um determinado intervalo, nesse caso deixando os valores padronizados entre o intervalo de 0 e 1, em seguida é criada uma coluna chamada *est* (estimativa) onde os valores retornados pela função são atribuídos a essa coluna. A função de normalização descrita acima está ilustrada a seguir:



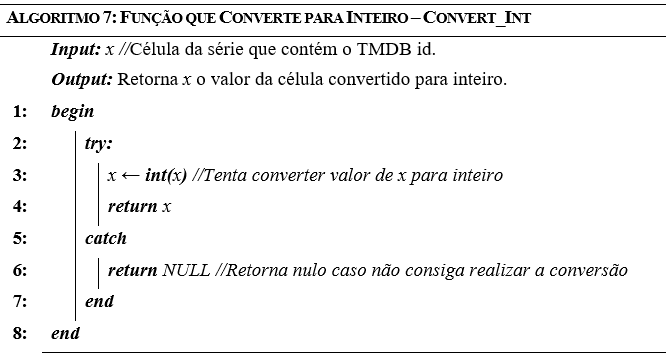
Enfim, são selecionadas as colunas *títulos* e *est* do *ds\_titulos* ordenados os títulos de forma decrescente pelos valores da coluna de estimativas e logo em seguida é retornado o dataframe contendo os dez títulos e as respectivas estimativas dos filmes que foram avaliados pelos os vizinhos mais próximos e que o usuário ainda não assistiu.

* 1. **SISTEMA DE RECOMENDAÇÃO HÍBRIDA**

Para a construção do sistema de recomendação híbrido foram reunidas as técnicas anteriormente implementadas na análise de conteúdo, colaborativa e vizinhos mais próximos. O funcionamento do sistemas de recomendação se dará da seguinte forma:

* **Entrada:** Identificação do usuário (ID) e o título do filme no idioma inglês.
* **Saída:** Dataframe contendo os filmes semelhantes ao informado classificados para um determinado usuário, com base nas classificações deste usuário e ordenados pela coluna de estimativa (‘est’).

Inicialmente o algoritmo começa com a carregando o arquivo CSV *links\_small* que contém o mapeamento entre identificadores dos filmes (IDs), os IDs do The Movie Database (tmdbIds) e os IDs do Internet Movie Database (imdbIds), sendo selecionadas as colunas *movieId* e *tmdbId*, os valores da coluna *tmdbId* são convertidos para inteiros fazendo o uso da função *convert\_int* que consta no algoritmo abaixo. Essa função é responsável por converter um valor em inteiro, ela recebe uma variável “x” e tenta convertê-la, caso a conversão falhe é retornado um valor nulo.



Em seguida, é renomeada a coluna da *tmdbId* para *id* do arquivo carregado e realizado o merge com o dataframe smd (Similar Movie Data) que contém as informações dos filmes, filtrada as colunas título e id, além de definir como identificador do dataframe o título e atribuido a *id\_map*. Logo depois é criada uma variável (*indices\_map*) que irá conter o mesmo dataframe sem a coluna de título e é definida a função que *agrupa\_recomendacoes* que irá juntar o resultado das recomendações das abordagens em conteúdo e filtragem colaborativa(k-NN e SVD) conforme o algoritmo a seguir.

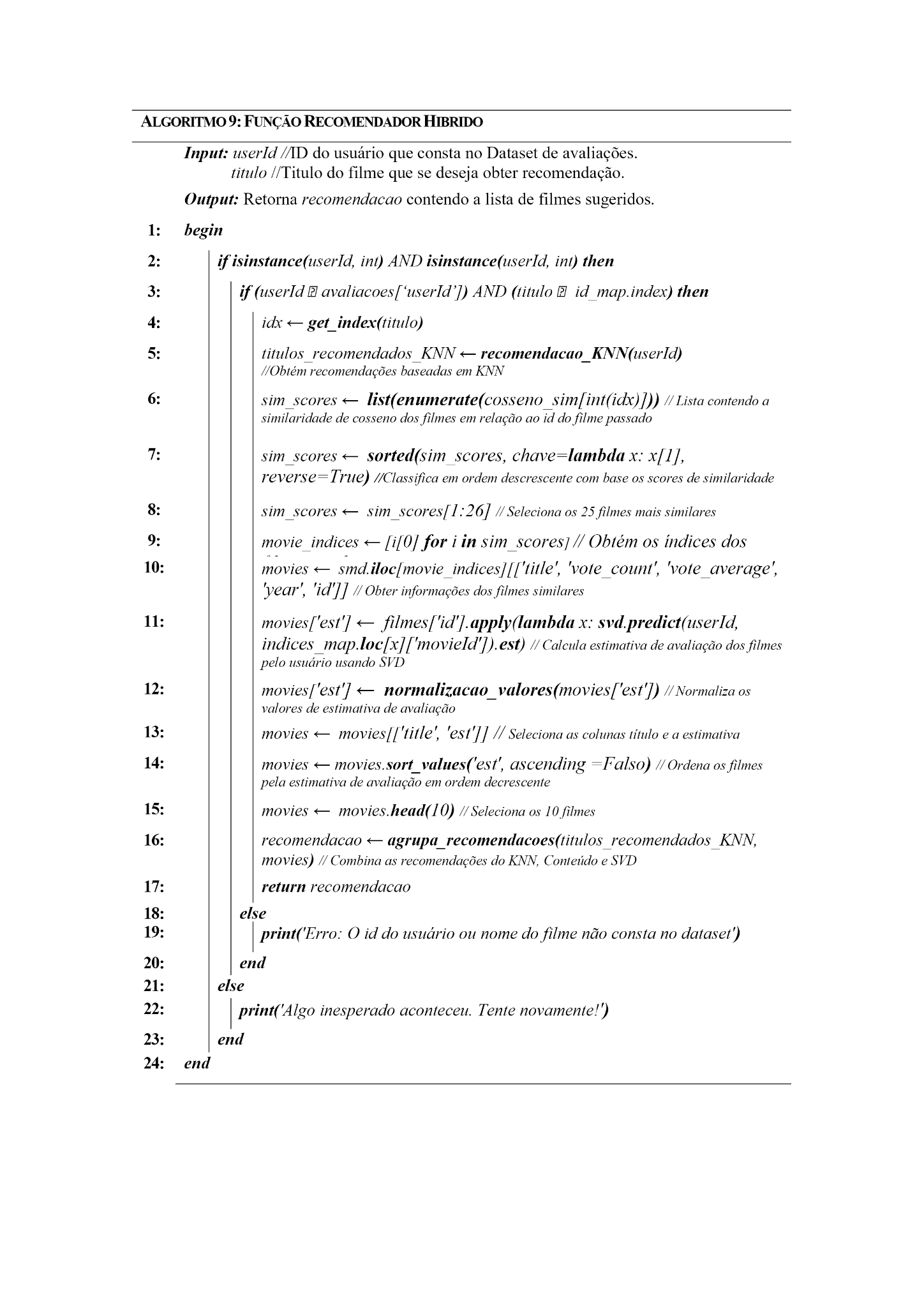
Aqui coloco o algoritmo da função *agrupa\_recomendacoes 8*

A função acima combina os dataframes que contém os resultados de cada abordagem em seguida agrupa as recomendações com base nos títulos e média das estimativas de classificação, isso é feito devido que o mesmo título pode ser retornado pelas diferentes abordagens utilizadas visando evitar que o valor da estimativa (nota) eleve mais determinado filme em consequência do seu título se repetir nas recomendações. Logo após os títulos são ordenados com base nas estimativas em ordem decrescente e o índice é redefinido para valor inteiro. Essa função é utilizada posteriormente para agrupar as recomendações geradas pelas diferentes abordagens.

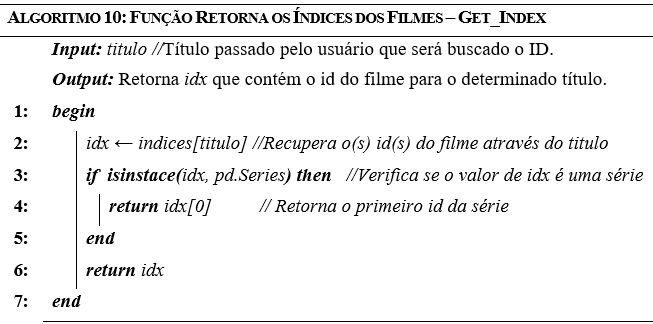
Os sistemas de recomendação híbridos possuem dois design predominantes, o paralelo e o sequencial. Para o desenvolvimento do sistema de recomendação deste trabalho foi utilizada a abordagem sequencial onde são fornecidos os parâmetros de entrada para um único mecanismo de recomendação e a saída é passada para as próximas abordagens de recomendação em uma sequência até que as saídas sejam combinadas e as recomendações fornecidas ao usuário conforme a figura abaixo.



Como demonstrado na imagem, as setas pontilhadas representam os passos (**P1** e **P2)** do percurso percorrido por cada abordagem no sistema de recomendação híbrido, mostrando a forma que as recomendações foram computadas e agrupadas. Após o entendimento do design utilizado para o desenvolvimento do sistemas de recomendação híbrido será descrita a função *recomendador\_hibrido* que é uma das principais funções desse mecanismo de recomendação, nela serão passados o id (código) do usuário e o título do filme para qual deseja obter a recomendação.



A função *recomendador\_hibrido* recebe como parâmetros o código do usuário e o título do filme, esses dados são passados para uma estrutura de controle que verifica se o código do usuário é inteiro e se o título do filme é um texto caso contrário uma mensagem de erro é exibida, então é verificado se o id do usuário está presente no dataframe de avaliações e se o título do filme está presente no dataframe *id\_map,* se ambos os dados estiverem presentes nos dataframes a função prossegue. Depois o título é passado para a função *get\_index* para obter o índice do filme correspondente ao título que foi passado, abaixo a imagem descreve o código da função.



A função *get\_index* recebe o título do filme e será buscado o id do filme na série índices criada no final da abordagem baseada em conteúdo. Em seguida é feita uma verificação se o valor retornado é uma série ou o código o título, como muitos filmes podem possuir o mesmo título mas seus conteúdos, histórias e enredos podem ser totalmente diferentes, em alguns casos podem ser retornados uma série contendo os índices do filmes com aquele determinado título, então foi definido que somente o primeiro índice dessa série será retornado.

Logo após é chamada a função *recomendacao\_KNN* passando como parâmetro para ela o id do usuário que retorna uma lista dos filmes recomendados para o usuário, empregando o método de filtragem colaborativa baseada em usuários e fazendo o uso do algoritmo de k-Nearest Neighbors (k-NN), a lista retornada é armazenada na variável *titulos\_recomendados\_KNN* que será usada logo mais. Depois é criada uma variável chamada *sim\_scores* que contém a lista similaridade do cosseno do filme consultado e os demais filmes, a similaridade do cosseno foi calculada na abordagem baseada em conteúdo.

A lista de *sim\_scores* contém tuplas que comportam os valores do índice dos filmes e suas respectivas similaridades, em seguida os filmes são classificados em ordem decrescente com base nos escores de similaridade e então são selecionados os vinte e cinco filmes mais semelhantes, um detalhe a se notar é que nessa lista de similaridade o primeiro item será o próprio filme e devido a isso o range para selecionar esses filmes inicia do código 1 ao invés de 0. Dos filmes selecionados são extraídos os índices e armazenados na variável *movie\_indices*.

No passo seguinte os índices selecionados contidos em *movie\_indices* são usados para extrair informações relevantes dos filmes no Dataframe smd essas informações são: título, contagem de votos, média de votos, ano de lançamento e ID do filme, esses são armazenados na variável *movies*. E então o ID do filme é utilizado para obter uma estimativa de classificação do usuário para cada um dos filmes selecionados fazendo uso da função “predict” do SVD que foi treinado na abordagem colaborativa, esse método calcula a previsão de classificação para determinado usuário e item, retorna a previsão e então é obtida a classificação estimada.

Em seguida, é utilizada a função listada no algoritmo 6 para normalizar as estimativas de classificação dos filmes para que estejam na escala entre 0 e 1, mesma escala da saída da abordagem colaborativa que usa k-NN. Depois são filtradas as colunas dos títulos e estimativas, esses dados são ordenados de forma decrescente com base nas estimativas de classificação dos filmes e então são selecionados os dez melhores filmes com base nas estimativas de classificação e são armazenados na variável *movies*.

Logo após é chamada a função *agrupa\_recomendacoes* presente no algoritmo 8 desta seção, que irá receber como parâmetros as variáveis contendo recomendações obtidas da abordagem colaborativa que usa o k-NN e da abordagem de conteúdo combinada com a colaborativa que usa SVD. Desse modo são passadas as variáveis contendo as respectivas recomendações *titulos\_recomendados\_KNN* e *movies*, são combinadas as listas recomendadas e então é retornada a recomendação híbrida resultante da combinação dos dois Dataframes de filmes recomendados.

Por fim, é observado que o sistema de recomendação híbrido fornece lista de recomendações distintas para cada usuário, inclusive se os títulos dos filmes passados para o mecanismo de recomendação for o mesmo. A lista de recomendação possui vinte títulos ordenados de forma decrescente por suas respectivas notas (estimativas) para que o usuário possa ter um leque variado de filmes para selecionar. Demonstrando assim, que as recomendações são personalizadas e moldadas para usuários diferentes.

ANÁLISE COLABORATIVA

SISTEMA DE RECOMENDAÇÃO HÍBRIDA

KNN

NOSSO ALGORITMO

**4 CONSIDERAÇÕES FINAIS**

Esse trabalho apresenta a proposta do desenvolvimento de um sistema de recomendação de músicas baseado nos gêneros musicais mais escutados pelos usuários, este sistema será implementado durante a disciplina de Trabalho de Conclusão de Curso II.

Os sistemas de recomendação visam oferecer, com base em uma grande quantidade de dados e análise do perfil do usuário, informações pertinentes que podem interessar de forma específica ou em grupo aos usuários.

Tendo isso em vista o mercado musical, os sistemas de recomendação servem tanto para fortalecer o vínculo entre artista e fã visto que as recomendações se baseiam no perfil do que o usuário ouve, além de ser uma nova forma para descoberta de novos artistas e músicas, que antes só era possível por meios de comunicação como rádio e TV.

Para o autor os principais desafios a serem enfrentados é aprender de forma mais profunda a utilização de Inteligência Artificial (IA), como utilizar a IA aplicada ao contexto de um sistema de recomendação de músicas.

Ao mesmo tempo, o autor deste trabalho espera que com o conhecimento adquirido com o desenvolvimento desse projeto possa contribuir para seu início no mercado de trabalho visto que sistemas de recomendação atualmente são muito usados para auxiliar no processo de tomada de decisão das empresas e contribuem para aumento das vendas direcionando produtos e serviços para compradores em potencial.

Após a revisão bibliográfica esse trabalho segue para etapa de desenvolvimento onde será feita a escolha do Dataset (que são bases de dados específicas que servem de amostras para treinamentos), implementação do sistema de recomendação, escolha dos algoritmos que serão aplicados para recomendação e testes dos mesmos.

Como trabalhos futuros esse trabalho poderá receber o feedback do usuário sobre a sugestão, fazer recomendações de playlists completas para os usuários e fornecer sugestões analisando o sentimento da pessoa obtendo assim um maior nível de personalização na recomendação.

**5 CRONOGRAMA**



**REFERÊNCIAS**

ADIYANSJAH; GUNAWAN, A. A. S.; SUHARTONO, D. Music recommender system based on genre using convolutional recurrent neural networks. Procedia computer science, v. 157, p. 99–109, 2019.

APUKE, O. D. Quantitative research methods: A synopsis approach. Kuwait Chapter of Arabian Journal of Business and Management Review, American University, v. 33, n. 5471, p. 1–8, 2017.

DA SILVA, A. R. O que está por trás dos sistemas de recomendação? , 9 out. 2019. Disponível em: <https://statplace.com.br/blog/o-que-esta-por-tras-dos-sistemas-de-recomendacao/>

DA SILVA, William Rodrigues; DOMINGUES, Marcos Aurélio. Musipath: um sistema para exploração de relacionamentos de artistas em redes de músicas. Brazilian Journal of Development, v. 8, n. 4, p. 27802-27820, 2022.

INTERNATIONAL FEDERATION OF THE PHONOGRAPHIC INDUSTRY. IFPI issues Global Music Report 2021. [S. l.], 23 mar. 2021. Disponível em: https://www.ifpi.org/ifpi-issues-annual-global-music-report-2021/. Acesso em: 31 maio 2022.

JORDAO, Pedro Henrique Ribeiro. SISTEMA DE RECOMENDAÇÃO DE MUSICAS USANDO LDA E ATRIBUTOS DE AUDIO. 2016. Dissertação de Mestrado. Universidade Federal do Rio de Janeiro.

RICCI, F. et al. (EDS.). Recommender Systems Handbook. Boston, MA: Springer US, 2011.

ROCCA, B.; ROCCA, J. Introduction to recommender systems. Disponível em: <https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada>. Acesso em: 9 jun. 2022.

SILVA, D. C. DA. Explorando calibragem ponderada, balanceamentos e métricas para justiça em sistemas de recomendação. 2021.