****

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

**SISTEMA DE RECOMENDAÇÃO HIBRIDO**

**REKOMENNDADOR HYDRA**

****

**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 de filmes; Filmes.

**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 que faz uso dos metadados que descrevem as faixas e das interações do usuário com o sistema para sugerir músicas de forma personalizada ao usuário. A natureza do desenvolvimento do software segue a metodologia aplicada.

Em busca de referencial teórico foi realizada uma pesquisa bibliográfica direcionada em sistemas de recomendação de filmes. As fontes utilizadas na pesquisa foram livros, sites, artigos científicos disponíveis no Google Scholar, realizadas entre maio e junho de 2022.

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

* Revisão bibliográfica: ….
* **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.
* **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 classificá-los.
* **Personalização do sistema de recomendação:** processo de personalização do sistema de recomendação com o algoritmo de classificação treinado.

**2.1 DATASETS**

Para o desenvolvimento do sistema de recomendação o presente estudo fez uso do conjunto de dados *The Movies Dataset* para análise e construção do sistema de recomendação. O *The Movies Dataset* é uma fonte abrangente de dados relacionados a filmes compilados por Rounak Banik e disponibilizados no Kaggle (citação). Essa fonte de dados foi criada a partir de dados do Dataset *MovieLens* que é uma base de dados de acesso gratuita e pública mantido pelo GroupLens e utilizada em estudos anteriores, incluindo os trabalhos de Silva (2021), Steck (2018) e Kaya e Bridge (2019).

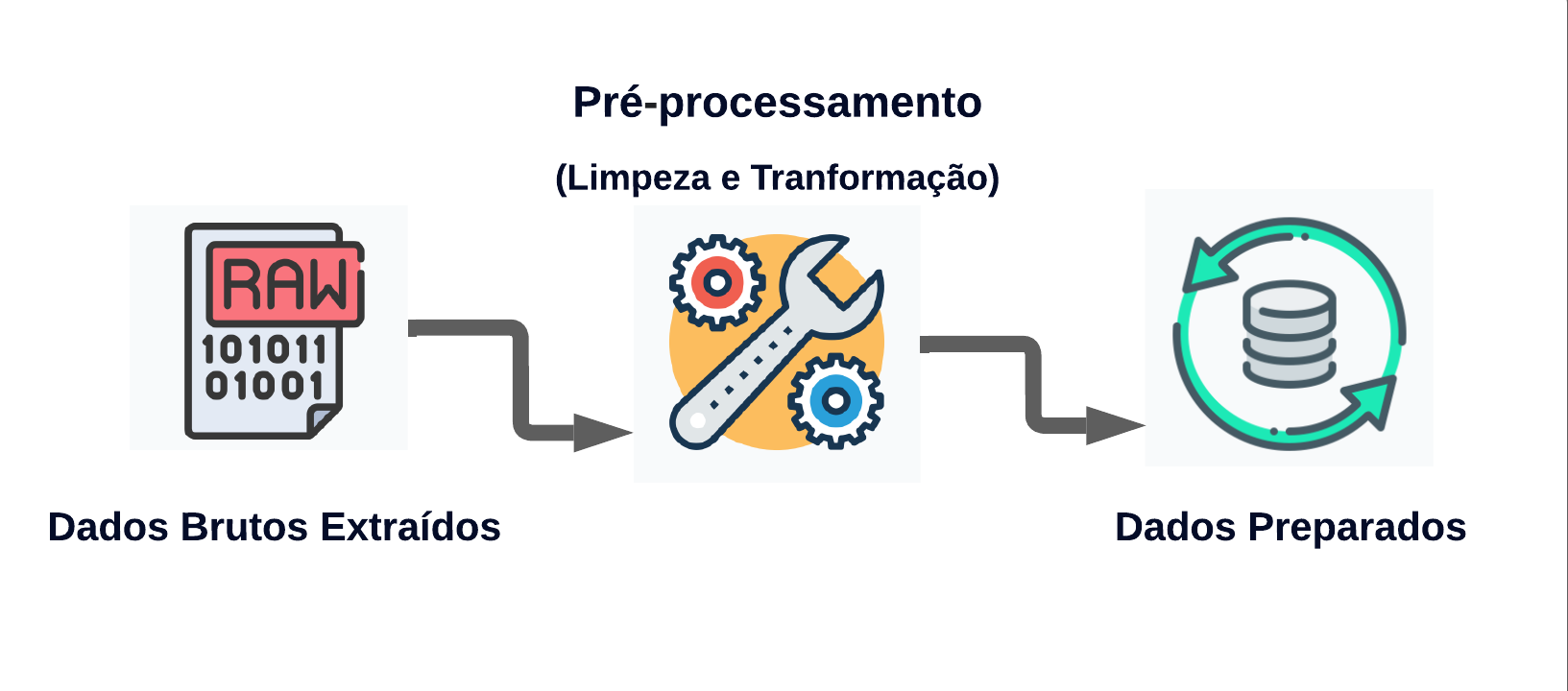
O *The Movies Dataset* contém uma vasta gama de informações sobre filmes, incluindo detalhes sobre títulos, elenco, equipes de produção, gêneros, sinopses, datas de lançamento, idiomas, contagem de votos TMDB, médias de votos, avaliações de usuários, dados de palavras-chave, tags, links para informações em outras bases e muito mais. A base de dados foi coletada do site de competição Kaggle (<https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset>) que oferece um rico conjunto de informações interessantes para pesquisas como: análises exploratórias, estudos de mercado, recomendação de filmes e muitas outras aplicações de pesquisa científica no campo de cinema e indústria cinematográfica.

A base de dados é apresentada em formato CSV (Comma-separated values - valores separados por vírgulas) e é composta por vários arquivos, cada um contendo informações específicas sobre diferentes aspectos e características dos filmes. O conjunto de dados é composto por filmes que foram lançados até julho de 2017 e é dividido em **The Full Dataset** e **The Small Dataset**.

O **The Full Dataset** é o conjunto de dados completo que consiste em mais de 26.000.000 classificações fornecidas por mais de 270.000 usuários para os mais de 45.000 filmes. Já o **The Small Dataset** é um subconjunto composto por mais 100.000 classificações a 9.066 filmes, provenientes de cerca de 700 usuários. Para o desenvolvimento deste trabalho foram utilizados todos os arquivos do **The Small Dataset** e os arquivos de palavras-chave e créditos do **The Full** Dataset pois esses não possuíam um subconjunto menor. Logo abaixo são listadas informações sobre cada Dataset utilizado no trabalho.

* **movies\_metadata:** contém informações gerais sobre os filmes, tais como título, gênero, sinopse, idioma, país de origem, orçamento, receita e diversas outras informações relevantes. Além disso, ele traz detalhes sobre as empresas envolvidas e países de produção. Contém informações de mais de 45.000 filmes.
* **credits:** traz informações sobre o elenco e a equipe de produção de cada filme, como atores, diretores, roteiristas e demais membros da equipe técnica. Nele, são disponibilizados dados como: nome, papel e ID dos profissionais envolvidos. Essas informações estão presentes na forma de uma string JSON codificada.
* **keywords:** contém palavras-chave associadas a cada filme, que podem ser usadas para descrever os temas ou os assuntos abordados nos filmes. Essas informações estão presentes na forma de uma string JSON codificada.
* **links\_small:** contém uma seleção de 9.000 filmes e apresenta informações relativas à identificação de filmes em outros sites de filmes, tais como o ID do IMDB (do Internet Movie Database), o ID do TMDB (do The Movie Database) e o ID do filme no site MovieLens.
* **ratings\_small.csv:** contém uma seleção de mais de 100.000 avaliações, provenientes de cerca de 700 usuários e contém avaliações de usuários para os filmes, incluindo o ID do usuário, o ID do filme, a classificação atribuída pelo usuário e o horário em que a avaliação foi feita.

As bases de dados brutas passaram por um pré-processamento, durante o qual uma sequência de transformações foi aplicada, com o objetivo de torná-las mais limpas e passíveis de filtragem. A figura abaixo exemplifica essa etapa de pré-processamento.



O pré-processamento é uma etapa muito importante, pois envolve a preparação dos dados para qualquer projeto que envolva análise de dados, além de ser uma etapa fundamental para garantir a qualidade dos dados e reduzir erros na análise (citação). A seguir estão listadas algumas etapas de preparação dos dados utilizadas nesse projeto:

* Organização na forma da apresentação dos dados da coluna de gêneros dos filmes. (exemplo: era “assim foi para assim”)
* Criação de uma coluna que possua o dado do ano de lançamento do filme.Acrescentar de onde foi extraída este ano
* Foram removidas as linhas [19730, 29503, 35587], pois possuíam dados mal formatados dos filmes (ex. linhas do dataframe onde foi inserido id no atributo data, além de valores ausentes nos demais atributos).
* Atribuição de tipos de dados corretos as colunas. (exemplo)
* Remoção de linhas duplicadas do Dataframe de filmes.
* Filtragem dos filmes com base nos dados do Dataset de **links\_small**.Explicar melhor o que significa filtrar nesse contexto e o motivo que levou a fazê-la

Após a realização de todas essas etapas, os dados resultantes encontram-se limpos, preparados e foram modelados de acordo com cada modelo de dados, apresentando, portanto, uma estruturação adequada (Silva, 2021). A base de dados resultante possui 9.086 registros de filmes, 100.004 avaliações e 671 registros de usuários.

**3 FUNDAMENTAÇÃO TEÓRICA**

Este capítulo apresenta uma fundamentação teórica com o objetivo de introduzir os conceitos fundamentais que contribuem para a melhor compreensão do trabalho e que foram utilizados como base para o desenvolvimento do sistema de recomendação.

**3.1 SISTEMAS DE RECOMENDAÇÃO**

De acordo com a FACELI et al. (2021), na década de 1990, surgiu o conceito de sistemas de recomendação, que consistem em ferramentas que utilizam informações relacionadas aos usuários e aos itens para encontrar um conjunto de itens que possam agradá-los. Com o amadurecimento desses sistemas ao longo dos anos, o seu uso se tornou cada vez mais comum, sendo difícil navegar pela internet sem encontrar alguma ferramenta que faça uso desse conceito.

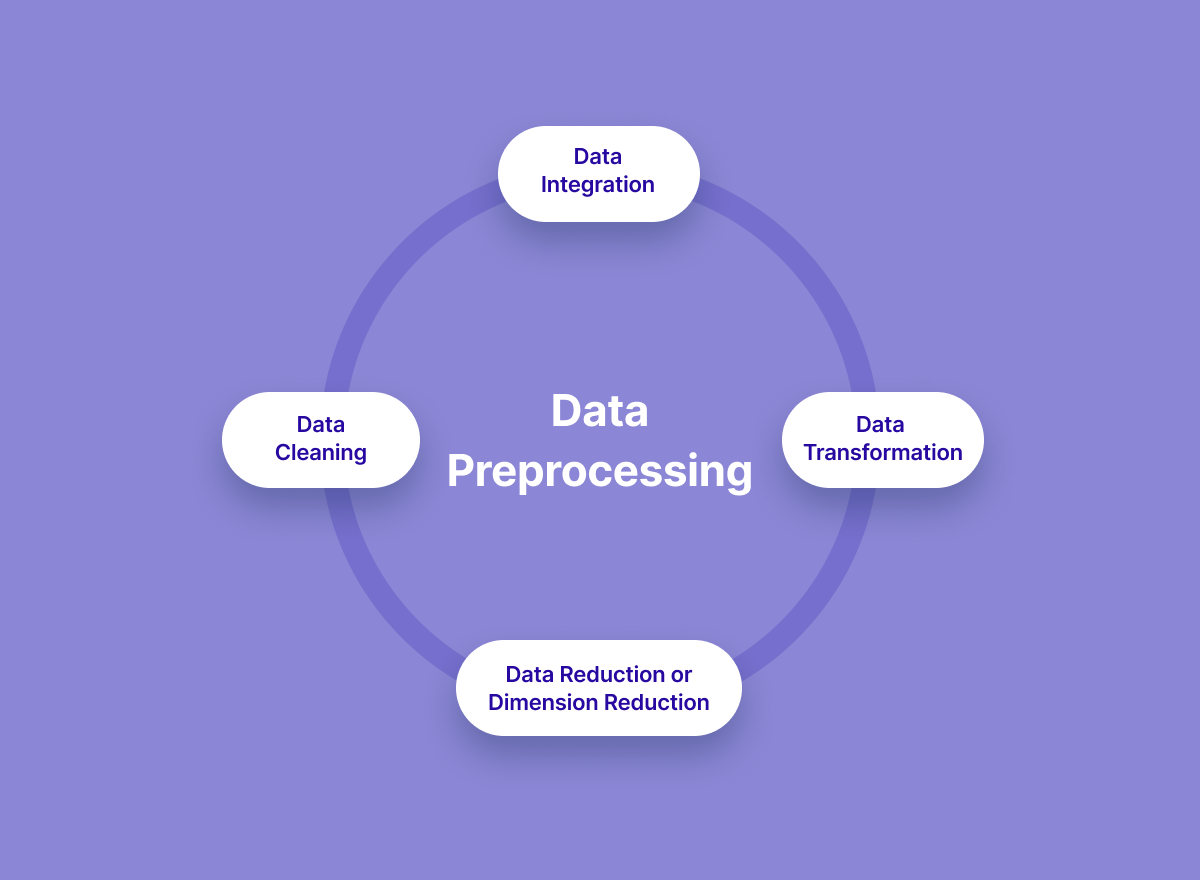
Os sistemas de recomendação são algoritmos que fornecem sugestões personalizadas de itens para usuários com base em suas preferências e comportamentos passados. Para compor a lista de sugestões, os sistemas de recomendação utilizam dados históricos de interação entre usuários e itens para prever e sugerir itens que os usuários possam estar interessados (SILVA, 2021).

Quanto à construção de sistemas de recomendação existem várias abordagens de desenvolvimento, sendo algumas das principais: filtragem baseada em conteúdo, filtragem colaborativa, filtragem híbrida. Cada uma dessas abordagens são brevemente explicadas nas próximas seções.

**3.1.1 PRÉ-PROCESSAMENTO**

A etapa de pré-processamento é uma das etapas essenciais para o desenvolvimento de sistemas de recomendação. Os conjuntos de dados podem apresentar diferentes características, formatos, dimensões e seus atributos podem possuir dados faltantes, duplicados, valores incorretos ou inconsistentes. Além disso, conjunto de dados pode conter um número pequeno ou elevado de atributos e Faceli et al. (2021) afirmam que o estado do dado afeta no desempenho dos algoritmos de aprendizado de máquina.

As técnicas de pré-processamento de dados são frequentemente empregadas para melhorar a qualidade dos dados, seja eliminando ou minimizando problemas existentes citados anteriormente (FACELI et al., 2021). O objetivo dessa etapa é preparar os dados para que possam ser consumidos pelos algoritmos de aprendizado de máquina, de forma a obter resultados mais precisos e relevantes, algumas das tarefas realizadas nessa etapa são: limpeza, normalização, transformação dos dados, além de redução de dimensionalidade e criação de recursos.



Dessa forma, as técnicas de pré-processamento são úteis não apenas por minimizar ou eliminar problemas do conjunto de dados, mas também por tornar os dados mais adequados para serem utilizados por determinados algoritmos de aprendizado de máquina (FACELI et al., 2021). Em resumo, a etapa de pré-processamento é fundamental para a construção de sistemas de recomendação eficazes. Ela ajuda a limpar, normalizar, transformar e resumir os dados, tornando-os prontos para serem usados pelo algoritmo de recomendação.

**3.1.1.1 PRÉ-PROCESSAMENTO TEXTUAL**

O pré-processamento textual é uma técnica específica para tratamento de dados textuais, que visa melhorar a qualidade dos dados antes de serem usados em algoritmos de aprendizado de máquina e mineração de texto. Cazella, Drumm e Barbosa (2010) destacam que, é comum que muitas palavras em um documento não sejam relevantes para representá-lo semanticamente, sendo que, em geral, substantivos ou grupos de substantivos são as palavras mais representativas em um documento. Essa etapa de pré-processamento é importante para garantir que os algoritmos de aprendizado possam entender e extrair informações relevantes dos dados textuais.

Algumas das técnicas comuns de pré-processamento textual incluem:

* Remoção de stopwords: remove palavras comuns que não adicionam significado ao texto, como "a", "e", "o" e "de".
* Tokenização: divide o texto em unidades menores, geralmente palavras, que podem ser processadas pelo algoritmo. Esse processo pode envolver a remoção de pontuação e caracteres especiais.
* Stemming e lematização: reduzem as palavras a sua raiz ou forma básica, para que diferentes formas de uma palavra possam ser tratadas como a mesma. Por exemplo, as palavras "casa", "casas" e "casinha" podem ser reduzidas à raiz "cas".
* Normalização de caso: garante que todas as palavras no texto estejam em maiúsculas ou minúsculas para que não haja ambiguidade nas análises.
* Remoção de caracteres não alfabéticos: remove caracteres não alfabéticos como números e símbolos que não têm relação com o conteúdo do texto.
* Bag-of-words: tem como propósito criar uma representação numérica de um conjunto de palavras em um texto, atribuindo a cada única palavra um número, de acordo com sua frequência no texto. Esse número é calculado com base na quantidade de ocorrências de cada palavra no texto. Essa técnica não leva em conta as características sintáticas das palavras (JÚNIOR, 2021).

Para realizar algumas das etapas de pré-processamento, foram utilizadas funções de bibliotecas específicas. No caso da tokenização, por exemplo, foi empregada a função CountVectorizer da biblioteca Scikit Learn. Conforme descrito por Campos e Figueiredo (2021), essa função realiza a contagem de palavras em cada documento e transforma dados textuais em vetores. Seguindo o mesmo raciocínio, Júnior (2021) descreve que a função CountVectorizer utiliza o conceito de Bag-of-Words para produzir um vetor de características. Já para realização da lematização foi a SnowballStemmer biblioteca da Natural Language Toolkit (NLTK).

As técnicas pré-processamento textual ajudam na melhoria da qualidade dos textos e, consequentemente, aumentam a eficácia dos algoritmos de aprendizado e mineração de texto, essas técnicas permitem que esses algoritmos possam lidar com dados textuais de forma mais eficiente e produzir resultados mais precisos e relevantes.

**3.1.1 COLD START PROBLEM**

Este capítulo apresenta uma fundamentação teórica com o objetivo de introduzir os

**3.1.2 FILTRAGEM COLABORATIVA**

**FATORAÇÃO DE MATRIZES**

**KNN**

**SIMILARIDADE DE COSSENO**

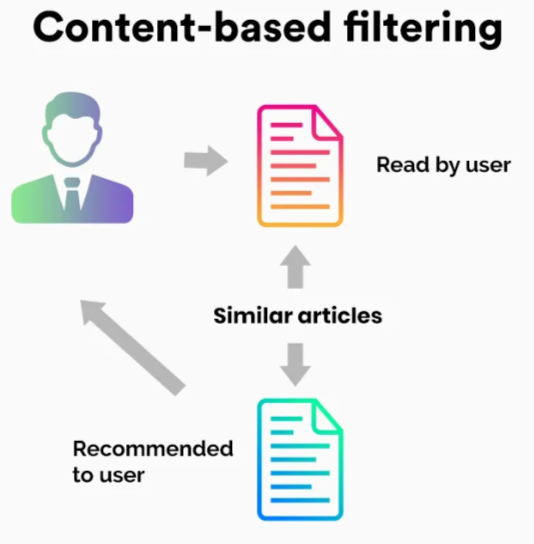
**DISTÂNCIA DE COSSENO**

**3.1.3 FILTRAGEM BASEADA EM CONTEÚDO**

A abordagem baseada em conteúdo é uma técnica de filtragem de informações baseada que usa características dos conteúdos para filtrar para recomendar itens semelhantes. De acordo com Dias et al. (2018), o sistema aprende a recomendar itens semelhantes aos que o usuário apreciou anteriormente, levando em consideração as características associadas a esses itens, por exemplo, se um filme de ação foi avaliado positivamente pelo usuário, o sistema aprende a recomendar outros filmes do mesmo gênero.

A ideia por trás da filtragem baseada em conteúdo é que se um usuário gostou de um item específico, é provável que ele goste de outros itens que possuem características semelhantes. Segundo Silva (2021), os sistemas que adotam essa técnica examinam os metadados que constituem os itens, criando uma estrutura de informações que é utilizada no processo de recomendação. Esse processo se baseia na utilização dos metadados dos itens e das preferências do usuário para encontrar itens desconhecidos com conteúdos semelhantes, classificando-os em uma lista de recomendação.

Os sistemas baseados em conteúdo são amplamente utilizados em várias áreas, como comércio eletrônico, música, filmes e mídia social. Eles podem ser implementados de várias maneiras, fazendo uso dos metadados e analisando seu conteúdo como: palavras-chave, sinopses, tags, gêneros, elencos, músicas entre outros. Como exemplo, na figura é possível verificar que o usuário comprou um determinado livro e devido a similaridade das informações que constituem o livro foi recomendado outra obra similar.



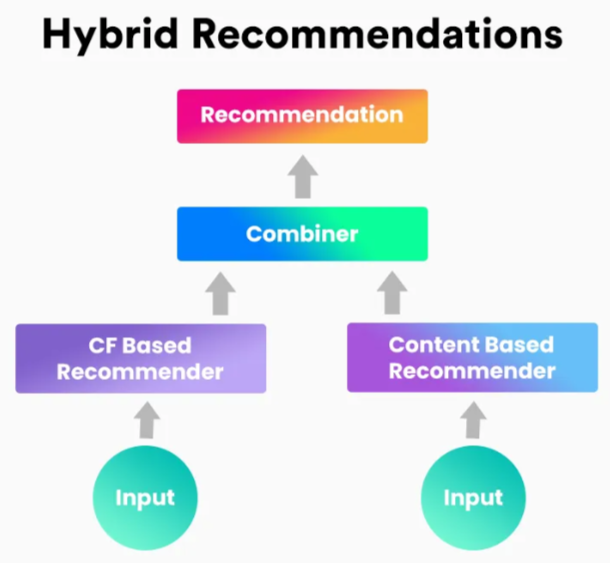
Assim como outras técnicas de recomendação, os sistemas que utilizam filtragem baseada em conteúdo têm suas vantagens e desvantagens. Algumas das desvanatgens são listadas nos estudos de Silva (2021) e Pires (2021). Entre as vantagens temos a rápida adaptação, onde o sistema pode se adaptar rapidamente às mudanças nas preferências do usuário, independentemente da magnitude dessas alterações. Outra vantagem é que o sistema pode fornecer recomendações de novos itens mesmo que o usuário não tenha muitos itens no perfil ou não forneça uma quantidade de feedbacks significativos.

Já entre as desvantagens estão superespecialização onde os itens recomendados têm o máximo de similaridade com os itens já presentes no perfil do usuário, levando o mesmo a ficar preso em uma "bolha de preferências", temos a dependência de informações em que os itens recomendados dependem de metadados que os descrevem, quanto menos informações sobre os itens estiverem disponíveis, mais imprecisa será a recomendação.

A abordagem de filtragem baseada em conteúdo enfrenta menos problemas de partida a frio (cold start problem) do que a abordagem colaborativa. Isso decorre devido que novos usuários ou itens podem ser descritos por suas características (conteúdo), permitindo que sugestões relevantes sejam feitas para essas novas entidades. Com informações sobre as características dos itens ou usuários, o sistema consegue gerar recomendações personalizadas e relevantes desde o início, sem precisar depender de interações anteriores do usuário com o sistema.

**3.1.4 FILTRAGEM HÍBRIDA**

Os sistemas de recomendação que usam a abordagem de filtragem híbrida são aqueles que combinam múltiplas técnicas de recomendação para gerar sugestões personalizadas para os usuários. De acordo com as pesquisas realizadas por Dias et al. (2018) e Silva (2021), um sistema de recomendação híbrido consiste na combinação de duas ou mais técnicas de recomendação, visando melhorar o desempenho e minimizar as desvantagens de cada técnica individualmente. Essa abordagem pode ser realizada de diversas maneiras, como por exemplo, combinando sistemas baseados em conteúdo com filtragem colaborativa.



A precisão e diversidade de itens sugeridos podem ser aprimoradas pela combinação de múltiplos métodos de recomendação, visto que essa abordagem leva em conta informações de diferentes fontes e técnicas de recomendação. Além disso, Dias et al. (2018), em sua pesquisa, explica que o uso dessa abordagem reduz o problema de "cold-start", onde o sistema tem dificuldades para fazer recomendações para novos usuários ou itens. Devido aos resultados superiores em comparação com os métodos utilizados individualmente, essa abordagem tem se tornado mais amplamente utilizada no mercado.

Sistemas de recomendação híbridos são uma abordagem promissora para a geração de recomendações personalizadas e relevantes para os usuários. A combinação de diversas abordagens de recomendação pode resultar em sugestões de melhor qualidade, mas é importante escolher as técnicas apropriadas e integrar as informações de forma eficaz.

.

**3.2 FEEDBACK EXPLÍCITO E IMPLÍCITO**

**4 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. **ESTRATÉGIA BASEADA EM CONTEÚDO**

Inicialmente para customizar as recomendações foi calculada 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. Chama-se de filtragem baseada em conteúdo quando os metadados dos filmes (conteúdo) são utilizados. Nesta etapa foram utilizados os datasets (citar quais 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).. *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. equipe de animação, efeitos especiais, produtores, editores entre outros). 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 calcular a similaridade cosseno. Após calcular a contagem de frequência de cada palavra, é criada uma série com 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 nova série contendo estas 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. **ESTRATÉGIA 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 (citação). 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 (Algoritmo x) 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 estratégia *colaborativa* foi implementada utilizando a biblioteca *Surprise,* usando 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.

Na linha 2 (Algoritmo 4), 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, é carregado 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 (linha 3).



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 algoritmo de recomendação usando o K-NN (K - Nearest Neighbors ou K - Vizinhos mais Próximos) que usa a técnica de filtragem colaborativa baseada em usuários. Este algorimto 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 algoritmo são verificados se os identificadores *id* e *imdb\_ids* são nulos no *ds\_filmes* (criada na Seção 3.1), 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 Seção 3.1)*,* além disso são selecionadas as colunas *movieId* e *title* do *ds\_filmes* e do ds\_avaliacoes (dataset de avaliações criado na Seção 3.1) 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 como parâmetro. Logo após é definida a quantidade de vizinhos a serem avaliados, que no caso deste 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 REKOMENNDADOR HYDRA**

Para a construção do sistema de recomendação ReKomeNNdador Hydra 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 em inglês (EN-US).
* **Saída:** Dataframe contendo os filmes semelhantes ao informado e ranqueados para um determinado usuário com base nas características deste usuário e ordenados pela coluna de estimativa (‘est’).

Inicialmente o algoritmo começa 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 7. 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.



A função agrupaRecomendações() (Algoritmo 7) combina os dataframes que contém os resultados de cada abordagem e 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 algoritmo 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. Em seguida é descrita a função *recomendador\_hibrido* que é responsável por gerar as recomendações do ReKomeNNdador Hydra, nela serão passados o id (código) do usuário e o título do filme para qual se deseja obter a recomendação.



A função *recomendador\_hibrido* (Algoritmo 9)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 (Algoritmo 10)* 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, 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 ReKomeNNdador Hydra 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.

**5 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.

FACELI, K. et al. Inteligência artificial: uma abordagem de aprendizado de máquina. 2021. 2. ed. Rio de Janeiro: LTC, 2021.

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.

JUNIOR, M.; SANTOS, L. D. Comparando predição de popularidade de Podcast a partir de metadados, conteúdo e características de áudio. [s.l.] Universidade Federal de Pernambuco, 9 mar. 2021.

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.