UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

SISTEMA DE RECOMENDAÇÃO DE FILMES ATRAVÉS DO ALGORITMO ALS (*ALTERNATING LEAST SQUARES*)

HIAGO LAURENÇO DONHA

GOIÂNIA
2023

HIAGO LAURENÇO DONHA

SISTEMA DE RECOMENDAÇÃO DE FILMES ATRAVÉS DO ALGORITMO *ALS (ALTERNATING LEAST SQUARES)*

Trabalho de Conclusão de Curso apresentado à Escola Politécnica, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação.

Orientador:
Prof. Me. Aníbal Santos Jukemura

Banca Examinadora:
Profa. Me. Fernando Gonçalves Abadia
Prof. Me. Lucília Gomes Ribeiro

GOIÂNIA
2023

HIAGO LAURENÇO DONHA

SISTEMA DE RECOMENDAÇÃO DE FILMES ATRAVÉS DO ALGORITMO *ALS (ALTERNATING LEAST SQUARES)*

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção do título de Bacharel em Ciência da Computação, e aprovado em sua forma final pela Escola Politécnica, da Pontifícia Universidade Católica de Goiás em 12 de Dezembro de 2023.

_____
Orientador: Prof. Me. Aníbal Santos Jukemura

_____
Profa. Me. Fernando Gonçalves Abadia

_____
Prof. Me. Lucília Gomes Ribeiro

GOIÂNIA
2023

AGRADECIMENTOS

Agradeço primeiro a Deus, pois sem ele não teria chegado tão longe nesta caminhada de muitos desafios no qual se passaram mais de 5 anos.

Agradeço a minha família, minha mãe e meu pai por sempre me apoiarem, apesar de já ter pensado em desistir, eles sempre me dando força e incentivo a continuar lutando pelos meus objetivos e sonhos neste longo processo.

Agradeço ao meu professor Anibal Jukemura por aceitar esse desafio que é ser meu orientador acreditando no meu esforço e em meu projeto e que me ajudou muito a evoluir. A professora Carmen que me auxiliou com muita paciência e dicas para minha evolução.

Agradeço ao meu professor e meu primeiro chefe Francisco Calaça no qual acreditou em mim e me deu a oportunidade de estagiar na *GetCoders,* onde pode me abrir portas para o mercado de trabalho.

Agradeço aos meus amigos de curso que espero levar para toda a vida. Manu e Fernando que sempre me apoiaram e conseguiram extrair o melhor de mim, mesmo quando estava prestes a desistir, somos uma grande equipe.

Agradeço ao meu professor Alexandre Ribeiro que teve um papel fundamental na minha aprendizagem em algoritmos e estrutura de dados no qual tive muita dificuldade, porém com sua impecável didática em ensino de tais matérias, mas não só em tais matérias, mas também na capacidade de raciocinar problemas.

Agradeço a todos os professores que contribuíram para minha formação acadêmica.

Agradeço a mim mesmo por sair da zona de conforto e não escolher um apenas projetos que me deixam confortável, mas pode ir atrás de projetos desafiadores que me ensinam sempre coisas novas.

RESUMO

Sistemas de recomendação estão se tornando ferramentas indispensáveis para diversos serviços de *streaming* e *websites*, que buscam oferecer ao seu usuário uma experiência personalizada e simplificando sua utilização devido ao grande volume de dados diante desse ecossistema *big data.* O presente trabalho tem como propósito a aplicação de técnicas como filtragem colaborativa em um conjunto de dados pré-processados para organizar as amostras. O objetivo é propor um sistema de recomendação de filmes utilizando as preferências dos usuários, assim podendo se utilizar como facilitador para manter seus usuários dentro da plataforma. A abordagem teve como premissa a implementação e utilização do método *ALS* (Mínimos Quadrado Alternados). Após o treinamento de dados foi aplicado bateria de teste para garantir as métricas de avaliação como para garantir a qualidade do modelo. Conclui-se que o conjunto de técnicas empregadas neste trabalho pode ser utilizado para recomendar diversos tipos de amostras em áreas correlatas.

Palavras-chave: Filtragem Colaborativa. Sistema de recomendação. Mínimos Quadrado Alternados. *RMSE.* Aprendizado de máquina. *Apache Spark*. *MAE.*

ABSTRACT

Recommendation systems are becoming indispensable tools for various streaming services and websites, aiming to provide users with a personalized experience and simplify their usage amid the vast amount of data in this big data ecosystem. The present work aims to apply techniques such as collaborative filtering to a pre-processed dataset to organize samples. The goal is to propose a movie recommendation system using user preferences, serving as a facilitator to keep users engaged within the platform. The approach is based on the implementation and utilization of the Alternating Least Squares (ALS) method. After training the data, a test battery was applied to ensure evaluation metrics and model quality. It is concluded that the set of techniques employed in this work can be used to recommend various types of samples in related areas.

Keywords: Collaborative Filtering. Recommendation System. Alternate Least Square. *RMSE.* Machine Learning. *Apache Spark*. *MAE.*

LISTA DE ILUSTRAÇÃO

LISTA DE TABELAS

LISTA DE SIGLAS

ALS	*Alternating Least Squares*
FC	Filtragem Colaborativa
IA	Inteligência Artificial
MAE	*Mean Absolute Error*
ML	*Machine Learning*
RMSE	*Root Mean Squared Error*
SR	Sistema de Recomendação
SQL	*Structured Query Language*
JDK	Java *Development Kit*

SUMÁRIO

1 INTRODUÇÃO

Os Sistemas de Recomendação (SR) tornaram-se muito populares nos últimos anos e estão sendo empregados em diversas aplicações web. Trata-se de um tipo específico de sistema de filtragem de informações, cujo propósito é antecipar o comportamento de um usuário com base em suas preferências por determinado item. Por exemplo, o mecanismo de recomendação da Amazon oferece a cada usuário uma resposta personalizada na página inicial. Ademais, empresas como Amazon, YouTube e Netflix utilizam sistemas de recomendação para auxiliar os usuários na descoberta de vídeos novos e relevantes, proporcionando uma experiência do usuário aprimorada e gerando uma receita colossal (PENG, 2022).

O SR emprega uma variedade de tecnologias para filtrar os resultados mais relevantes e disponibilizar aos usuários as informações desejadas. O sistema de recomendação é categorizado em três grandes grupos: o primeiro é o Sistema de Filtragem Colaborativa, o segundo é o sistema Baseado em Conteúdo e o terceiro é o Sistema de Recomendação Híbrida (GOSH et al., 2021).

Os sistemas de recomendação representam ferramentas eficazes para a filtragem de informações online, difundidas em decorrência das mudanças nos hábitos dos usuários, das tendências de personalização e do acesso emergente à internet. Apesar de os sistemas de recomendação mais recentes serem notáveis por fornecerem recomendações precisas, eles enfrentam diversos desafios e limitações, como escalabilidade, *cold start*, entre outros (ROY, D et al, 2022).

A comunidade de pesquisa tem dedicado considerável esforço para aprimorar a aplicabilidade e o desempenho dos sistemas de recomendação nos últimos anos. Novas metodologias e algoritmos foram desenvolvidos para enfrentar muitos dos desafios tecnológicos, visando a produção de recomendações mais precisas e, simultaneamente, a redução do tempo de computação *online*. Diversos algoritmos de recomendação foram propostos e implementados com sucesso em diferentes domínios (SINGH, P. K. et al, 2021).

Diante deste contexto, a filtragem colaborativa utilizando *ALS* (Mínimos Quadrados Alternados), destaca-se como uma ferramenta eficiente para reduzir e evitar pesquisa manual por parte dos usuários. Ao identificar a matriz principal de classificação, é realizado o produto de duas matrizes inferiores, calculando assim,

aproximações chamadas de matriz de fatores (ALEXBUTGIT, 2023).

1.1 Objetivos

Este trabalho tem como principal objetivo a implementação de um sistema de recomendação utilizando algoritmo de *Machine Learning,* para gerar recomendações automáticas relacionadas a filmes, com base no histórico de *ratings* dos usuários. A avaliação dos resultados obtidos ocorrerá por meio de métricas avaliativas para validar o algoritmo, considerando parâmetros de teste e de treino, visando aprimorar todo o processo.

1.1.1 Objetivos Específicos

Além do objetivo principal, este trabalho propõe alcançar os seguintes objetivos específicos:

- Avaliar analiticamente o algoritmo aplicado para o sistema de recomendação, utilizando a técnica *ALS (Alternating Least Squares)* da biblioteca do *Apache Spark.*
- Apresentar resultados que possam contribuir para a melhoria da experiência do usuário ao procurar um novo item para aquisição por meio de compras online.

1.2 Motivação

Os sistemas de recomendação de comércio eletrônico estão se tornando cada vez mais importante no mundo digital. Eles são usados para personalizar a experiência do usuário, ajudar os clientes a encontrarem o que desejam, de forma rápida e eficiente, aumentando a receita para os negócios (SALUNKHE, T, 2023).

Atualmente, os sistemas de recomendação estão sendo cada vez mais utilizados para um grande número de aplicações como *web*, livros, *e-learning*, turismo, filmes, música, comércio eletrônico, notícias, recursos de pesquisa especializados e programas de televisão. (ROY, D et al, 2022). Portanto, estudar sistema de recomendação para aprofundar-se nesse campo reside na constatação de que a sua implementação eficaz pode conferir vantagens substanciais a

empresas e instituições em diversos setores.

1.3 Justificativa

A utilização de um sistema de recomendação pode oferecer várias vantagens às empresas, tais como: Amazon, Netflix, S (Yeung, Chi Ho, 2015).

Tais vantagens incluem o aumento das vendas, a possibilidade de os usuários escolherem produtos de acordo com suas preferências ou de seguirem as recomendações do sistema, além da capacidade desses sistemas em alcançar elevada precisão na correspondência de itens adequados. Observa-se também a sua ampla adoção em sites populares como uma estratégia para manter os usuários engajados e prolongar sua interação. Assim, a pesquisa busca compreender e explorar os mecanismos por trás desses sistemas, visando contribuir para o entendimento de sua aplicação e potenciais benefícios em diferentes contextos

Com a implementação do SR é possível alcançar os objetivos almejados neste trabalho. Sendo assim, o sistema de recomendação se revela como boa alternativa para qualquer negócio, independente do domínio (DONATI, 2018).

1.4 Estrutura do trabalho

Este trabalho apresenta informações que descrevem desde os conceitos básicos sobre SR, até as informações referentes ao funcionamento do sistema com os algoritmos de *Machine Learning*. Desta forma, o trabalho apresenta organizado conforme a seguir:

- O capítulo dois irá tratar os conceitos fundamentais, técnicas aplicadas, algoritmos para a compreensão do trabalho.
- O capítulo três apresenta trabalhos relacionados, no qual, foram fundamentais para o desenvolvimento deste trabalho.
- O capítulo quatro apresenta a metodologia adotada para o desenvolvimento do trabalho.
- O capítulo cinco apresenta testes realizados e seus resultados práticos obtidos no desenvolvimento do trabalho.
- O capítulo seis apresenta as avaliações dos resultados e conclusão deste

trabalho, bem como uma breve descrição dos trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

É importante conceituar alguns termos essenciais no contexto de sistemas de recomendação e aprendizado de máquina, fundamentais para a compreensão e análise crítica dos modelos propostos. O Aprendizado de Máquina é uma disciplina que utiliza algoritmos para capacitar sistemas a aprenderem padrões a partir de dados, aprimorando sua performance ao longo do tempo. No âmbito dos sistemas de recomendação, Modelos de Recomendação são aplicativos que preveem preferências ou comportamentos do usuário. A Filtragem Colaborativa é uma abordagem que recomenda itens com base nas preferências de usuários semelhantes, enquanto a Filtragem Baseada em Conteúdo sugere itens com base nas características do próprio item e nas preferências históricas do usuário. A Filtragem Híbrida combina elementos de ambas as abordagens para otimizar a precisão das recomendações. O desafio do *cold start* ocorre quando há falta de dados sobre um novo usuário ou item, complicando a personalização das sugestões. Por fim, as métricas de avaliação, como o *Mean Absolute Error (MAE)* e o Root *Mean Squared Error (RMSE)*, são cruciais para mensurar a eficácia dos modelos, proporcionando uma base quantitativa para a análise de desempenho em sistemas de recomendação.

2.1 Aprendizado de máquina

É útil apresentar a metodologia de aprendizado de máquina como uma alternativa para abordagem de engenharia convencional. O fluxo de projeto de engenharia convencional começa com a aquisição de conhecimento do domínio: O problema de interesse é estudado detalhadamente, produzindo um modelo matemático que captura a física da configuração em estudo. Baseado no modelo, é produzido um algoritmo otimizado que oferece garantias de desempenho sob a suposição de que o dado modelo baseado em física é uma representação precisa da realidade (SIMEONE, O, 2018).

De acordo com Russell e Norvig (2009), *Machine Learning* é a construção de sistemas que podem aprimorar seu desempenho com base na experiência adquirida, representando uma abordagem fundamental na construção de sistemas

inteligentes. A definição de Flach (2012) e Murphy (2012) destaca a centralidade do aprendizado a partir de dados, enfatizando a capacidade dos algoritmos de generalizar padrões e fazer previsões sem intervenção direta do programador. Essa abordagem, alinhada à perspectiva de Bishop (2006), reforça a importância de sistemas adaptativos capazes de evoluir e melhorar suas habilidades ao longo do tempo. Nesse sentido, o campo de estudo do *Machine Learning* assume uma posição crucial na vanguarda da pesquisa em Inteligência Artificial, influenciando positivamente o desenvolvimento de sistemas mais eficientes e autônomos.

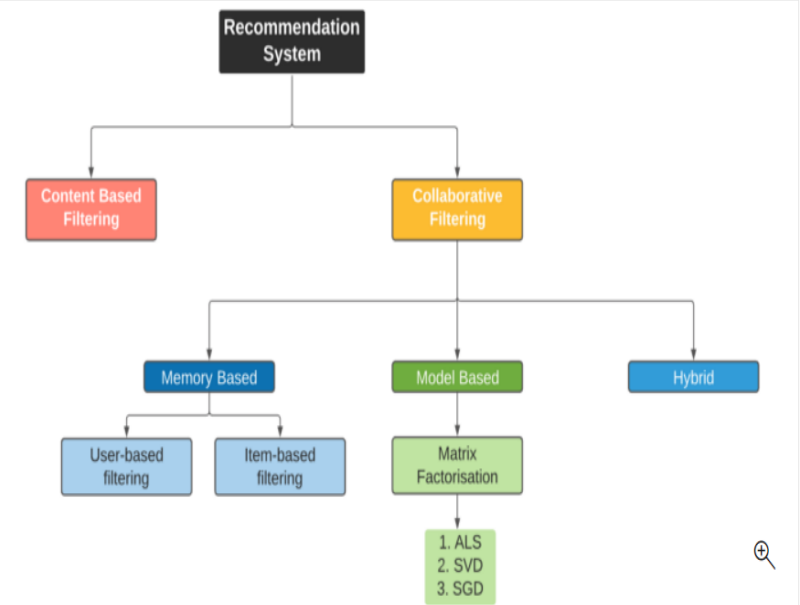
2.1.1 Modelos de Recomendação

Desde o primeiro modelo de Filtragem Colaborativa proposto na década de 1990, os sistemas de recomendação têm sido ativamente estudados, aplicados e expandidos em todos os campos da academia e da indústria até recentemente. Os sistemas de recomendação são sistemas de filtragem que fornecem uma recomendação de item personalizado para um usuário em um serviço ambiente que pode conter ou coletar vários dados (KO, H. et al, 2022).

A fim de aumentar a satisfação do usuário com o serviço do sistema de recomendação, é necessário recomendar vários itens para o usuário através do modelo de recomendação para ampliar o intervalo de seleção destes itens. Ao mesmo tempo, analisando os dados implícitos e explícitos do usuário e os dados de um grupo de usuários. (KO, H. et al, 2022).

A Figura 1 ilustra a visão geral sobre os principais tipos de sistemas de recomendação, com foco principal na estratégia utilizada neste trabalho que é a Filtragem Colaborativa. A Figura 1 também apresenta o contexto em que se inserem a Filtragem Baseada em Conteúdo e a Filtragem Híbrida. Tendo em vista que a Filtragem Colaborativa se utiliza da matriz de fatorização com mínimos quadrados alternados, explorando cada um entende-se melhor como funcionam os modelos descritos a seguir.

Figura 1 - Modelos de Recomendação



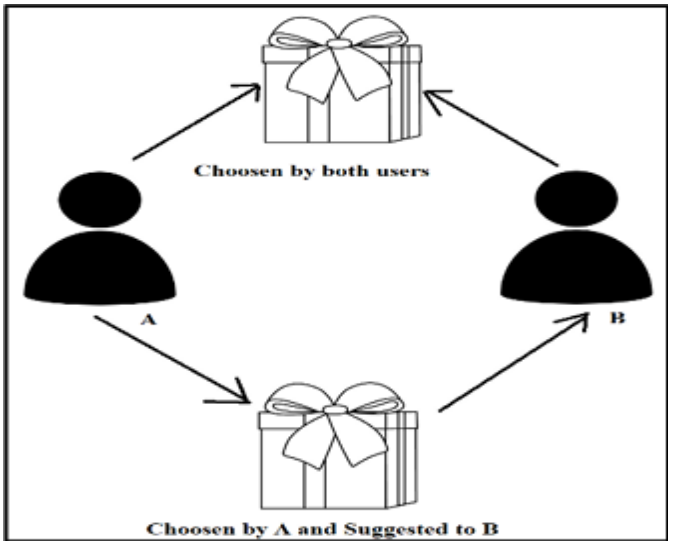
Fonte: ALEXBUTGIT. et al (2023)

2.1.2 Filtragem Colaborativa

Os sistemas de recomendação de Filtragem Colaborativa são aqueles que fazem sugestões feitas a um usuário com base nas preferências de outros usuários. O pressuposto é que se dois usuários tiverem preferências semelhantes, então é mais provável que o façam com os mesmos itens. Estes sistemas normalmente usam algumas estratégias de filtragem colaborativa, que evidenciam um processo de prever os interesses de um indivíduo com base nos interesses de outros usuários (SALUNKHE, T, 2023).

Sistemas de Filtragem Colaborativa produzem previsões ou recomendações para um determinado usuário e um ou mais itens de interesse. Os itens constituem um conjunto que possa fornecer uma classificação variada, como elementos de arte, livros, CDs, artigos de periódicos ou destinos de férias. Classificações em um Sistema de Filtragem Colaborativa podem assumir uma variedade de formas (RESEARCHGATE, 2001).

Figura 2 - Trabalhando com filtragem colaborativa



Fonte: Salunkhe, T (2023)

A Figura 2 ilustra as decisões dos usuários A e B que realizam a escolha de um presente, logo a partir dessas escolhas em comum é feito uma segunda recomendação baseada nas escolhas feitas anteriormente. Apresentado a filtragem colaborativa existem outras formas de recomendar e introduzir o próximo conceito de recomendação que é a filtragem baseada em conteúdo descrito a seguir.

2.1.3 Filtragem Baseada em Conteúdo

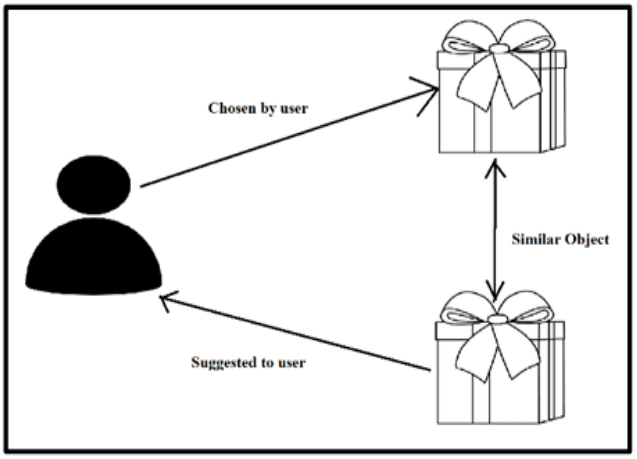
Filtragem Baseada em Conteúdo é uma técnica usada para recomendação de produtos para clientes/usuários de acordo com a similaridade dos itens. É amplamente utilizada em sistemas de recomendação para encontrar itens que vários usuários já demonstraram interesse. O conteúdo é usado para determinar a similaridade dos itens que pode ser qualquer texto para realizar a avaliação do produto (SALUNKHE, T, 2023).

No entanto, cada usuário tem interesses únicos e peculiares, que podem se relacionar com uma pequena porcentagem do conteúdo da *Web*. Portanto, tornou-se ainda mais difícil e demorado para os usuários encontrarem informações de interesse. Para ajudar nesse processo, a *World Wide Web*, popularmente conhecida

por Internet, pode ser personalizada, usando SR (RESEARCHGATE, 2001).

A Figura 3 ilustra a escolha de um usuário para um primeiro item que logo é recomendado um segundo item com similaridade equivalente para este mesmo usuário, ao contrário do que se encontra descrito anteriormente na Figura 2, que se utiliza a recomendação a partir de escolhas de um ou mais usuários para conseguir realizar tal objetivo.

Figura 3 - Trabalhando com filtragem baseada em conteúdo



Fonte: Salunkhe, T (2023)

2.1.4 Filtragem Híbrida

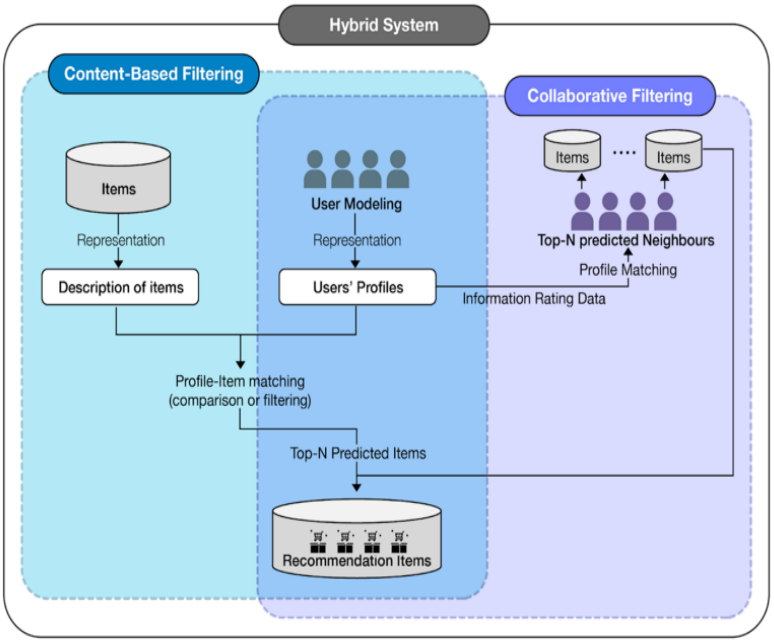
Ambos os modelos de filtragem apresentam limitações. O modelo de Filtragem Baseado em Conteúdo depende de metadados sobre o item do usuário, e o de Filtragem Colaborativa depende do item do usuário. Um modelo de Recomendação Híbrido foi proposto para resolver as limitações de ambos os modelos supracitados e para melhorar o desempenho da recomendação (KO, H. et al, 2022).

Como o modelo de Recomendação Híbrida é projetado principalmente para resolver a escassez, o principal objetivo deste modelo é compensar a falta de dados de classificação, integrando as informações dos modelos de Filtragem Baseada em Conteúdo e Filtragem Colaborativa (KO, H. et al, 2022).

A Figura 4 ilustra a recomendação híbrida utilizando-se de duas bases de dados, sendo a primeira de itens e a segunda de um grupo de usuários. O modelo

gera uma recomendação tanto de usuários quanto de itens. Com isso, é possível diferenciar este modelo dos outros modelos descritos nas Figuras 2 e 3. Porém, vale ressaltar que todos esses modelos de recomendação possuem um problema denominado *cold start* que será descrito a seguir.

Figura 4 - Trabalhando com filtragem híbrida



Fonte: KO, H. et al (2022)

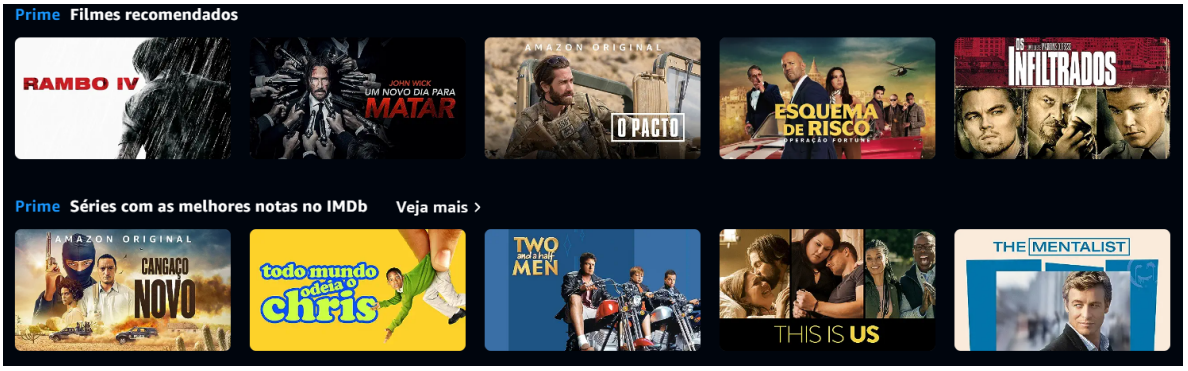
2.2 *Cold start*

O problema definido como *cold start* ocorre quando os sistemas de recomendação recebem novos usuários, com nenhuma ou pouca informação pessoal e de histórico de consumo (SANTANA, M). Tal configuração gera uma falha que ocorre quando não há dados suficientes para tais recomendações precisas ocorram. Esta pode ser uma questão importante para os sistemas de recomendação porque o seu principal propósito é fazer recomendações com base em avaliações e experiências passadas de usuários.

Utilizando um mecanismo de recomendação híbrido que incorpora filtragem baseada em conteúdo e colaborativa, o problema do *cold start* pode ser resolvido (SALUNKHE, T, 2023). Quando se cria uma conta em plataformas como Prime

Vídeo é possível observar o problema de *cold start*, pois o usuário não tem um perfil de recomendação baseado em suas preferências. Isso faz com que a plataforma gere recomendações iniciais aleatórias, para assim poder mapear o perfil e preferências de determinado usuário com o tempo de uso.

Figura 5 - Filmes recomendados na Prime Video



Fonte: Do autor (2023)

A Figura 5 ilustra a plataforma Prime Vídeo que se utiliza da mesma estratégia de recomendação procurando atualizar histórico de preferências do usuário em questão, fazendo recomendações mais genéricas como séries com melhores notas no IMDB ou até mesmo recomendações da própria plataforma. Sabendo-se disso o algoritmo visa preencher esses usuários sem ou com nenhum histórico para realizar recomendações mais assertivas.

Entretanto apenas conceitos não são suficientes para o entendimento completo dessas estratégias de recomendação, a seguir será descrito como o *apache spark* será utilizado com o algoritmo Alternating Least Squares (ALS).

2.3 Alternating Least Squares

ALS é um tipo de método de filtragem colaborativa usado para resolver o problema de *overfitting* em dados esparsos. O *Apache Spark* é usado para implementar o algoritmo ALS. (GOSH, S. et al, 2021).

O *Least Square Alternating* também é um algoritmo de fatoração de matriz e é executado de forma paralela. O ALS é implementado no *Apache Spark ML* é criado para problemas de filtragem colaborativa em grande escala. O *ALS* executa um

trabalho essencial na resolução da escalabilidade e da escassez dos dados de classificação, e é simples e escalável para conjuntos de dados muito grandes (LIAO, K).

Código 1 - Algoritmo ALS

```
Procedure ALS(Xu, Yi)
Initialization Xu <- 0
Initialization matrix Yi with random values
Repeat
    Fix Yi, solve Xu by minimizing's the objective
    function(the sum of squared erros)
    Fix Yi solve Yi by minimizing the objective
    function similarly
Until reaching the maximum iteration
Return Xu, Yi
End procedure
```

Fonte: Do autor (2019)

O Código 1 descreve em linguagem *ALGO* (linguagem de programação) o funcionamento genérico. Esse filtro colaborativo visa preencher as entradas que faltam em uma matriz de associação de usuário-item. Usuários e produtos são descritos por um pequeno conjunto de fatores latentes que podem ser usados para prever entradas ausentes. Assim, o algoritmo de ALS é usado para aprender esses fatores latentes. A ideia é pegar uma matriz grande e fatorá-la em alguma representação menor da matriz original por meio de mínimos quadrados alternados. O resultado da aplicação do algoritmo resume-se em duas ou mais matrizes dimensionais inferiores cujo produto é igual ao original (WALTRICK, C).

Figura 6 - Matriz de fatoração mínimos quadrados

Fonte: ALEXBUTGIT (2023)

Como apresentado na Figura 6, em cada iteração, uma das matrizes de fator é mantida invariável, enquanto a outra é submetida à resolução por meio do método dos quadrados mínimos. Após a conclusão bem-sucedida dessa etapa, a matriz de fatores recém-resolvida é fixada, permanecendo constante durante o processo de resolução aplicado à outra matriz de fatores. Essa abordagem iterativa visa otimizar a resolução de ambas as matrizes de fatores, seguindo as diretrizes e procedimentos (ALEXBUTGIT, 2023).

Utilizando o algoritmo deve-se realizar o trabalho de monitoramento utilizando métricas, descrito a seguir.

2.4 Métricas

O sistema de recomendação é avaliado usando análise *offline*. Durante a análise, não há usuários reais, pois o grande conjunto de dados é dividido em uma base treinamento e uma base de teste. O sistema de recomendação é treinado com o conjunto de dados para prever as classificações dadas pelos usuários diante da base de teste. A avaliação é realizada para comparar a previsão resultante com a previsão real e prevista no conjunto de dados de teste. Métricas populares de precisão preditiva, como Erro Médio Absoluto (MAE), Raiz do Erro Quadrático Médio (RMSE) e Ganho Médio do Usuário (MUG) são usados para mensurar a acurácia da predição feita pelo SR. **(**GOSH, S. et al, 2021).

2.4.1 - *Mean Absolute Error* (MAE)

O Erro Absoluto Médio (*Mean Absolute Error* - MAE) e a Raiz do Erro Quadrático Médio (*Root Mean Squared Error* - RMSE) são medidas utilizadas para avaliar o desvio médio entre uma avaliação prevista e a avaliação real de um usuário, mostrando a quão próxima as previsões do SR se encontra das avaliações reais dos usuários (HERLOCKER, J).

$$MAE = \frac{1}{|Rui|} \sum_{rk \epsilon rui} |ri(rk) - pi(rk)| \quad (1)$$

2.4.2 - *Root Mean Squared Error* (RMSE)

A Raiz do Erro Quadrático Médio (RMSE - *Root Mean Squared Error*) é basicamente o mesmo cálculo de *MSE*, contendo ainda a mesma ideia de penalização entre diferenças grandes do valor previsto e o real. Porém, para lidar com o problema da diferença entre unidades, é aplicada a raiz quadrática como demonstrado na fórmula (2). Assim, a unidade fica na mesma escala que o dado original, resultando em uma melhor interpretação do resultado da métrica (JÚNIOR, C. DE O).

$$RMSE = \sqrt{\frac{1}{|Rui|} \sum rk \; \epsilon r \; ui \; |ri(rk) - pi(rk)|} \quad (2)$$

Para melhor entendimento das duas métricas utilizadas na mensuração de qualidade e monitorização das recomendações realizadas serão utilizados os experimentos realizados nos trabalhos de referência descritos a seguir.

3 ANÁLISE DA LITERATURA

O objetivo deste capítulo é apresentar resultados de outros trabalhos propostos e desenvolvidos com objetivos similares para esses sistemas, abordando uma variedade de implementações. Essa abordagem visa aprofundar a compreensão da filtragem colaborativa, técnica utilizada, sendo essencial, no entanto, primeiro compreender os problemas apresentados e as soluções propostas. Trabalhos relacionados, que utilizam a fatoração matricial para recomendação de vídeos e um sistema de recomendação destinado a sugerir artigos científicos serão utilizados para contextualizar e embasar a discussão.

3.1 Trabalhos relacionados

A seguir, serão apresentados alguns trabalhos relacionados que foram utilizados para estudo e pesquisa antes do desenvolvimento do presente trabalho.

3.1.1 Modelos de fatoração matricial para recomendação de vídeos

O objetivo final do trabalho proposto por (SOUZA et al., 2011) é avaliar modelos de fatoração matricial que têm demonstrado eficácia no problema de recomendação de vídeos, considerando o feedback implícito dos usuários em um domínio temporal. Os modelos de filtragem colaborativa, conforme apontado pelos autores, destacam-se como os mais amplamente utilizados para abordar tal problema. Uma das principais vantagens destes modelos reside na sua capacidade de adaptação a qualquer domínio, possibilitando a abordagem de especificidades nos conjuntos de dados que seriam desafiadoras de explorar por meio de outros modelos (SOUZA et al., 2011).

Considerando o modelo de filtragem colaborativa, duas abordagens comuns para implementar soluções desse modelo são os algoritmos de vizinhança e os modelos de fatoração latente. Os métodos de vizinhança visam

calcular relacionamentos entre itens ou usuários, construindo um modelo baseado em um grafo que descreve a vizinhança.

Os métodos item-item apresentados constroem os grafos de vizinhança com vértices conectando itens similares. Jà os métodos usuário-usuário, em uma abordagem conduzida pelo autor, constroem grafos de vizinhança nos quais os vértices conectam usuários similares.

A escolha deste autor recaiu sobre os modelos de fatoração por matrizes. No trabalho em questão, foi realizada uma avaliação do desempenho de alguns modelos de fatoração matricial otimizados para a tarefa de recomendação, considerando dados implícitos no consumo das ofertas de vídeos da Globo.com. O autor sugere tratar esses dados de consumo como indicativos de intenção por parte do usuário em assistir a um vídeo específico. Além disso, na análise, foram considerados os vieses únicos dos usuários e vídeos, e a variação temporal foi investigada quanto ao seu impacto nos resultados das recomendações.

3.1.2 Sistema de recomendações de artigos científicos utilizando dados sociais.

O objetivo final do trabalho de Grava et al. (2016) foi a proposição de um sistema de recomendação para trabalhos científicos, combinando informações sociais e bibliométricas relacionadas a artigos citados em publicações. Essa proposta visa facilitar a assistência aos pesquisadores, auxiliando-os a responder perguntas específicas, como identificar artigos relevantes em sua área ainda desconhecidos ou descobrir quais trabalhos podem contribuir para suas pesquisas em andamento. Para alcançar esse objetivo, foram desenvolvidas duas abordagens de recomendação:

- A primeira abordagem parte da premissa de que o tempo em que as relações entre os autores foram estabelecidas é determinante para selecionar os autores mais próximos (ou similares). Nesse contexto,

relações mais recentes são consideradas mais relevantes do que relações antigas.

- A segunda abordagem combina os resultados de diferentes técnicas implementadas na proposta, bem como técnicas da literatura correlata, a fim de gerar recomendações de maneira híbrida.

Os resultados obtidos indicaram que a solução baseada no tempo superou as estratégias correlatas, especialmente quando há mais informações disponíveis sobre o autor. Autores com diversas relações de coautoria e um extenso conjunto de artigos citados tendem a obter resultados superiores em comparação com aqueles que possuem poucas relações e citaram poucos artigos. Considerando esses trabalhos como referência para recomendações de vídeos e artigos científicos, juntamente com os conceitos apresentados anteriormente, a metodologia utilizada é descrita a seguir.

4 METODOLOGIA

A filtragem colaborativa, como processo de avaliação de itens com base nas opiniões de outras pessoas, é destacada como uma das técnicas mais indicadas para a criação de Sistemas de Recomendação, uma vez que opera com base nas preferências individuais de cada usuário e item para sugerir novas recomendações. O foco dessas técnicas é preencher as entradas faltantes em uma matriz de associação usuário-item. A biblioteca Spark.ml oferece suporte à filtragem colaborativa baseada em modelo, na qual usuários e produtos são descritos por um conjunto restrito de fatores latentes, utilizados para prever entradas ausentes.

Adicionalmente, a metodologia de pesquisa adotada neste estudo, caracterizada como descritiva e exploratória, complementa a compreensão dos dados coletados, proporcionando uma análise mais abrangente e embasada para a interpretação dos resultados obtidos, como será explicado nos próximos tópicos.

4.1 Metodologia de Pesquisa

O tipo de pesquisa empregado no presente artigo foi de natureza descritiva e exploratória, alinhado aos objetivos propostos, uma escolha fundamentada na proximidade com a questão, conforme destacado por Gil (2010). Nesse contexto, a construção de hipóteses se fez presente. Os procedimentos de coleta dos dados mencionados foram realizados por meio de pesquisa bibliográfica e documental, adotando uma abordagem quantitativa e qualitativa com o propósito de relacionar os dados para fins de interpretação. Além disso, a seção subsequente oferece uma transição natural ao apresentar os principais componentes do ambiente de desenvolvimento do sistema. Esse contexto técnico é essencial para a compreensão e implementação eficaz das soluções propostas, estabelecendo uma conexão entre a fundamentação metodológica e a aplicação prática do sistema em desenvolvimento.

4.2 Ambiente de desenvolvimento

Com o propósito de promover a compreensão tecnológica, a presente seção introduz os principais componentes empregados no ambiente de desenvolvimento do sistema, visando familiarizar o leitor com a implementação do sistema. Inicialmente, optou-se por utilizar o *Google Colab* como a primeira ferramenta para dar início ao desenvolvimento.

4.2.1 *Jupyter notebook*

Esta é uma ferramenta da *Google* que permite a execução de múltiplas linhas/blocos de código em diferentes células, possibilitando o trabalho com dados, sua movimentação para cima ou para baixo, e a obtenção dos resultados imediatamente abaixo da célula correspondente. Trata-se, essencialmente, de um organizador eficiente que atende às necessidades dos Cientistas de Dados e dos profissionais que executam código, representando uma solução há muito desejada. O Jupyter-Notebook, utilizado para escrever em R, SQL, Scala, entre outras linguagens, contribui para tornar o fluxo de trabalho mais fácil e eficiente, conforme destacado por Matos (2019). No entanto, no contexto específico deste trabalho, optou-se por utilizar Python devido à consistência das bibliotecas oferecidas por esta linguagem.

4.2.2 Python

Com Python, é possível acessar uma ampla variedade de bibliotecas de Ciência de Dados, tais como *NumPy*, *SciPy*, *Stats Models*, *scikit-learn*, *pandas*, entre outras, que estão experimentando um crescimento exponencial. Restrições em métodos de otimização e funções que poderiam ter sido limitadas em um ano anterior já não representam um obstáculo, e é possível encontrar soluções robustas e confiáveis para esses desafios (MATOS, 2019).

4.2.3 Bibliotecas

O desenvolvimento da aplicação abordada neste trabalho envolve um conjunto de ferramentas, incluindo a biblioteca *SPARK* do ecossistema *APACHE*, bem como *Spark SQL* e *MLlib*.

4.2.3.1 *Spark*

Apache Spark é uma plataforma de computação em cluster criada para ser rápida e de propósito geral e tem crescido muito em popularidade. O Apache Spark oferece basicamente 3 principais benefícios: facilidade de uso, velocidade e uso geral como diferentes tipos de computação como (*SQL Spark*), *Machine Learning* (*MLlib*). Aprofundando-se no módulo *MLlib* onde técnicas como filtragem colaborativa é capaz de utilizar-se de algoritmos como *ALS*.

4.2.4 *Apache Spark (ALS - Alternating Least Squares)*

No ecossistema Apache, encontra-se um módulo MLlib dedicado a *Machine Learning*, que disponibiliza uma instância do ALS para treinar conjuntos de dados. O Spark.ml atualmente oferece suporte à filtragem colaborativa baseada em modelos, conforme documentação disponível em "Collaborative Filtering - Spark 3.4.0 Documentation" (2023). A implementação desta pesquisa foi direcionada ao aprendizado de máquina não supervisionado, atuando com interface para o ALS.

A grande vantagem dessa esoclha reside na facilidade de implementação, sendo um algoritmo amplamente adotado por grandes empresas para propósitos comuns. Tais sistemas revelam-se cruciais para aumentar as taxas de cliques, gerando receitas substanciais para as organizações. Além disso, possuem mecanismos de recomendação que impactam positivamente as experiências do usuário, contribuindo para maior satisfação e retenção do cliente (TIWARI, 2020).

Há quase duas décadas, a Amazon incorporou o sistema de

recomendação, expandindo seu uso para setores como finanças e viagens. Observa-se que a Netflix, ao invés de exigir que os usuários naveguem por milhares de títulos de filmes, apresenta uma seleção mais restrita, proporcionando uma experiência de usuário aprimorada. Essa abordagem resultou em taxas de cancelamento mais baixas para a Netflix, resultando em economias significativas, estimadas em cerca de um bilhão de dólares anualmente (TIWARI, 2020).

O conjunto de dados utilizado para treinamento e teste do modelo ALS foi o *Movielens*, conforme detalhado na seção subsequente.

4.2.5 Conjunto de dados

O conjunto de dados utilizado para esta implementação foi fornecido pelo *GroupLens*, que coletou e disponibilizou as informações no site *MovieLens*. A escolha desse conjunto de dados fundamentou-se na disponibilidade para estudos e desenvolvimentos, além de ser respaldada pelo caso de sucesso da *Netflix*, que emprega recomendações de filmes.

O GroupLens é um laboratório de pesquisa situado no Departamento de Ciência da Computação e Engenharia da Universidade de Minnesota, especializado em sistemas de recomendação, comunidades online, tecnologias móveis e onipresentes, bibliotecas digitais e sistemas de informação geográfica local (“*What is GroupLens*?”, 2013).

O MovieLens abrange 20.000.000 de registros de classificação, com mais de 50.000 avaliações e cerca de 200.000 usuários, abrangendo o período de 1995 a 2018. O conjunto de dados, coletado, está dividido em dois arquivos: *ratings, movies*.

4.2.5.1 Arquivo *ratings*

Todas as classificações estão registradas no arquivo ratings.csv. Cada linha desse arquivo, após o cabeçalho, representa a avaliação de um filme por parte de um usuário e segue o formato: *userId*, *movieId*, *rating*, *timestamp*. A ordenação das linhas neste arquivo é realizada primeiramente por userId e, em

seguida, por userId e movieId. As classificações consistem nas notas atribuídas por cada usuário.

4.2.5.2 Arquivo *movies*

As informações dos filmes estão contidas no arquivo *movies*.csv. Cada linha deste arquivo após a linha do cabeçalho representa um filme e possui o seguinte formato: *movieId*, *title*, *genres*. Os títulos dos filmes são inseridos manualmente ou importados do site TMDB (“*The Movie Database”*, 2023) e incluem o ano de lançamento entre parênteses. Erros e inconsistências podem existir nesses títulos. Os gêneros são uma lista separada por *pipe* e são selecionados dentre os seguintes: ação, aventura, animação, criança, comédia, crime, documentário, drama, fantasia, musical, mistério, suspense, guerra, romance, ficção científica, gênero não listado.

4.2.6 Implementação

A implementação apresenta a abordagem que engloba, tanto para recomendação *cold start,* quanto para recomendações utilizando a técnica de filtragem colaborativa. A implementação utiliza-se de uma base para treinamento, teste e validação do conjunto de dados apresentado na seção 4.1.5.

O código fonte deste trabalho estará no repositório no *GitHub*: https://github.com/hiagodonha/ALS_Recommendation_TCCII**.**

No primeiro bloco de código foi feito o *download* do *JDK (Java Develop Kit)* necessário para a preparação do ambiente, pois o *Apache Spark* utiliza a *JVM (Java Virtual Machine)* para a compilação do código.

No segundo e terceiro bloco de código é realizada a importação das variáveis de ambientes e *spark* e das bibliotecas do apache spark para a realização da do tratamento e consulta dos dados. No último bloco da Figura 4.1 tem-se uma instância do spark na atual sessão.

Código 4.1 - importação ecossistema spark

```
[ ] !apt-get install openjdk-8-jdk-headless -qq > /dev/null
    !wget -q https://archive.apache.org/dist/spark/spark-3.3.2/spark-3.3.2-bin-hadoop2.tgz
    !tar xf spark-3.3.2-bin-hadoop2.tgz
    !pip install -q findspark

[ ] # configurar as variáveis de ambiente
    import os
    os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
    os.environ["SPARK_HOME"] = "/content/spark-3.3.2-bin-hadoop2"

    # tornar o pyspark "importável"
    import findspark
    findspark.init('spark-3.3.2-bin-hadoop2')

[ ] from __future__ import print_function

    import sys
    if sys.version >= '3':
        long = int

    from pyspark.sql import SparkSession

    from pyspark.ml.evaluation import RegressionEvaluator #evaluation é a biblioteca para verificação da qualidade do modelo
    from pyspark.ml.recommendation import ALS # ALS é o modelo de recomendação que será utilizadp
    from pyspark.sql import Row #row é o formato que o ALS trabalha, row conterá o id do usuario, id filme, nota e timestamp

[ ] spark = SparkSession.builder.master('local[*]').getOrCreate() #criar/iniciar a sessão spark
```

Fonte: Elaborado pelo autor (2023)

Código 4.2 - Filtragem Colaborativa

```
(training, test) = df_result.randomSplit([0.7, 0.3])

[262] als = ALS(maxIter=5, regParam=0.01, userCol="userId", itemCol="movieId", ratingCol="rating", coldStartStrategy="drop")

[263] model = als.fit(training) #treina o modelo com o dataset de treinamento

[ ] predictions = model.transform(test) #aplica o modelo no conjunto de teste para fazer predições
    evaluator = RegressionEvaluator(metricName="rmse", labelCol="rating",
                                    predictionCol="prediction")
    rmse = evaluator.evaluate(predictions)
    print("Erro médio quadrático = " + str(rmse))

    Erro médio quadrático = 0.9649223607971548

[320] predictions = model.transform(test) #aplica o modelo no conjunto de teste para fazer predições
    evaluator = RegressionEvaluator(metricName="mae", labelCol="rating",
                                    predictionCol="prediction")
    mae = evaluator.evaluate(predictions)
    print("Mean absolute error (MAE) on test data = " + str(mae))

    Mean absolute error (MAE) on test data = 0.7159874599152987

[265] userRec = model.recommendForAllUsers(10) #pegar todos os usuários e gerar 10 recomendações

[266] userRec.show()
```

Fonte: Fonte: Elaborado pelo autor (2023)

A partir deste momento, cria-se a instância do ALS com os seguintes parâmetros e técnicas:

- Divisão da base entre 70% destinado a treino e 30% para teste, utilizando o método randomSplit(), como observado na primeira linha do Código 4.2.
- Na linha 262: É onde de fato se inicia a instância do algoritmo ALS passando para ele os devidos hiperparâmetros como:
    - ***userCol*** : especifica o nome da coluna que contém os índices de usuário.
    - ***itemCol***: especifica o nome da coluna que contém os índices de item.
    - ***ratingCol***: especifica o nome da coluna que contém classificações de usuário para os itens.
    - ***coldStartStrategy***: especifica a estratégia para lidar com novos usuários ou itens durante a previsão. Neste caso, "*drop*" indica que os novos usuários ou itens serão ignorados.
    - ***rank***: especifica o número de fatores latentes (também chamados de dimensões) do ALS.
    - ***maxIter***: especifica o número máximo de iterações que o *ALS* pode executar durante o treinamento
    - ***regParam***: específica o termo de regularização que controla a força da penalidade para evitar o *overfitting*.
- Na linha 263 é realizado de fato o treinamento da base de que foi divida utilizando-se do método .fit() do MLlib Spark.
- Na linha 264 é extraído para uma variável predictions a utilização da base de teste. Depois é realizado o *RegressionEvaluator*(), passando como um dos hiperparâmetros para a métrica utilizada. A linha seguinte será explicada no capítulo de descrição e análise dos resultados a seguir.

5 DESCRIÇÃO E ANÁLISE DOS RESULTADOS

Foram realizadas duas baterias de testes para os usuários específicos: *user* 1, *user* 2 e *user* 3. O treinamento do conjunto de dados ocorreu mediante a utilização de diferentes hiperparâmetros, visando a avaliação comparativa dos modelos em relação à base de dados do *MovieLens*. Foram empregadas métricas de avaliação e monitoramento, tais como *RMSE* x *MAE*, para discernir a precisão do conjunto de filmes recomendados para cada usuário.

5.1 Filtragem colaborativa utilizando ALS

Conforme demonstrado na Tabela 1, encontram-se os resultados dos testes realizados para a aplicação da técnica de filtragem colaborativa. Para esta abordagem, adotou-se um limiar em que a nota prevista pelo sistema deve ser igual ou superior a 3. Essa escolha se fundamenta na premissa de que a recomendação deve se concentrar em filmes que o usuário possivelmente apreciará. Observa-se que os *ratings* obtidos conseguem atender ou superar o limiar estabelecido, indicando uma eficácia satisfatória no processo de recomendação.

Tabela 1 - Bateria 1 - usuário 1 x filmes

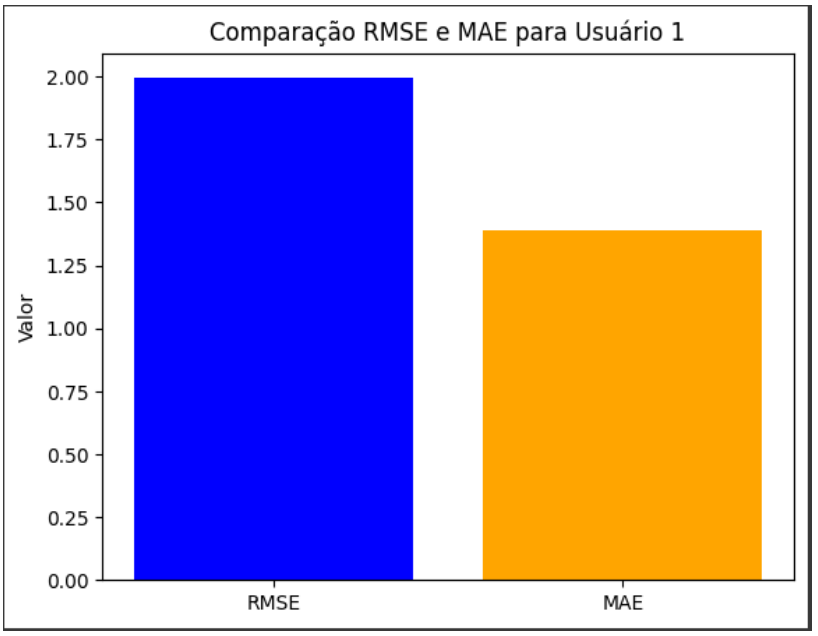
Título	Rating	Nota Mínima	Nota Máxima	Gênero
Infinity	6.0779	1.0	5.0	Drama
Come and See	5.7987	0.5	5.0	Drama | War
For Roseanna	5.7543	1.0	5.0	Comedy | Drama | Romance
Wolf Children	5.7332	0.5	5.0	Animation | Fantasy
Turin Horse	5.4557	4.0	5.0	Drama

Fonte: Elaborado pelo autor (2023)

Na Figura 9, os resultados dos testes para o usuário 1 são apresentados mediante métricas de avaliação RMSE e MAE. Constata-se que o usuário 1 apresenta resultados discrepantes, uma vez que as métricas indicam RMSE =

1,9205 e MAE = 1,2615, distanciando-se significativamente do valor alvo. Esses resultados apontam para possíveis variações nos *ratings* de até 1 ponto, evidenciando a necessidade de uma análise mais aprofundada para compreender e abordar as dificuldades enfrentadas pelo referido usuário.

Figura 9 - RMSE X MAE para usuário 1



Fonte: Elaborado pelo autor (2023)

Conforme descrito na Tabela 2, observam-se *ratings* mais realistas para o conjunto de recomendações, destacando a inexistência de *ratings* iguais ou superiores a 6, assim como a ausência de avaliações menores ou iguais a 3. Esses resultados proporcionam uma percepção mais apurada da correlação entre gêneros, evidenciando a consistência e adequação das recomendações geradas pelo sistema.

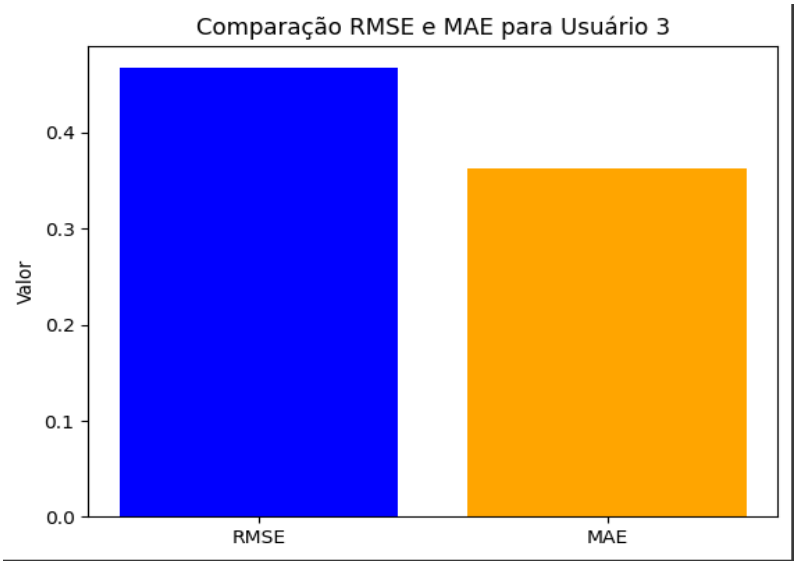
Na Figura 10, evidencia-se que o conjunto de filmes recomendado para o usuário 3 demonstra maior assertividade, conforme análise das métricas. Com valores de RMSE igual a 0,4953 e MAE igual a 0,3840, constatam-se resultados significativos, aproximando-se do desempenho ideal. Essas métricas indicam uma precisão notável na previsão de avaliações para o usuário 3, ressaltando a eficácia do sistema de recomendação na personalização das sugestões de filmes para atender às preferências desse usuário específico.

Tabela 2 - Bateria 1 - usuário 3 x filmes

Título	Rating	Nota Mínima	Nota Máxima	Gênero
Virunga	5.6604	5.0	5.0	Documentário | War
Earth	5.0704	3.0	5.0	Drama | War
Father and Daughter	5.0541	1.0	5.0	Animation | Drama
Road Home	5.0266	1.5	5.0	Drama| Romance
Big Blue	5.4557	1.0	5.0	Adventure| Drama

Fonte: Elaborado pelo autor (2023)

Figura 10 - RMSE X MAE para usuário 3



Fonte: Elaborado pelo autor (2023)

Conforme detalhado na Tabela 3, constata-se a obtenção de gêneros correlatos para o conjunto de filmes recomendado ao usuário 5, com *ratings* superiores em comparação ao usuário 1, conforme evidenciado na Tabela 1. Essa análise indica uma melhoria na qualidade das recomendações, revelando uma adequação mais apurada aos gostos do usuário 5 em relação ao usuário 1. A correlação de gêneros contribui para a personalização efetiva das sugestões, ressaltando a capacidade do sistema em ajustar-se às preferências específicas de cada usuário.

Tabela 3 - Bateria 1 - usuário 5 x filmes

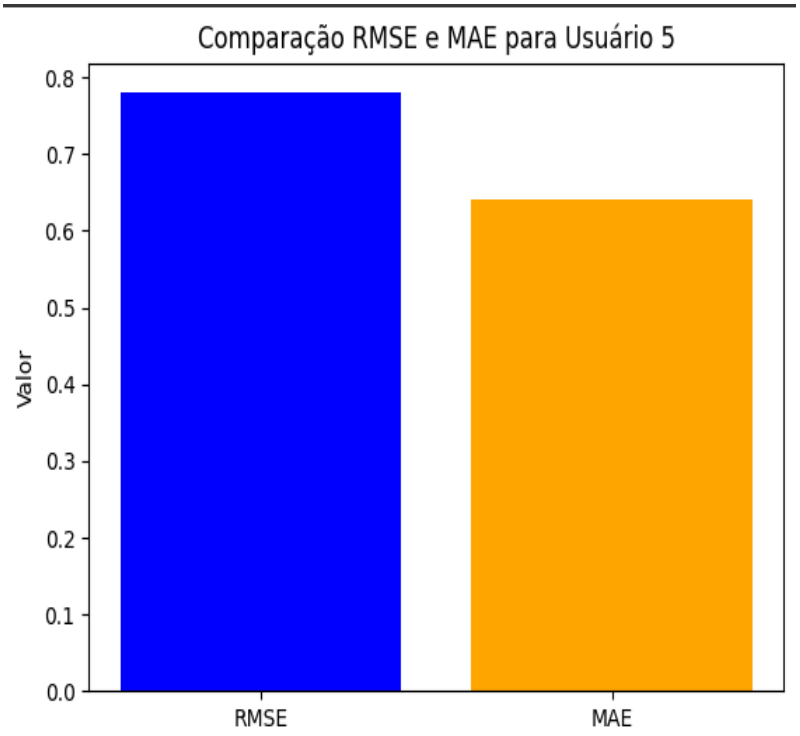
Título	Rating	Nota Mínima	Nota Máxima	Gênero
Monty Python	5.5975	0.5	5.0	Comedy
Fear City	5.7310	2.0	5.0	Comedy
Miss Pettigrew	5.5236	0.5	5.0	Comedy
Waiting for the	5.0362	0	5.0	Comedy
Ben-Hur	5.4883	1.5	5.0	Adventure | Drama

Fonte: Elaborado pelo autor (2023)

Conforme apresentado na Figura 11, a conclusão a ser extraída é que o RMSE apresentou valores discrepantes que não contribuíram para atingir uma média em conformidade com as expectativas. Entretanto, observa-se que o MAE se mantém relativamente estável diante da amostra recomendada ao usuário 5. Essa análise aponta para a necessidade de uma avaliação mais aprofundada do desempenho do sistema em termos de precisão das recomendações, especialmente no que diz respeito à métrica RMSE, a qual requer ajustes para melhor se alinhar com as expectativas estabelecidas.

A Tabela 4 foi elaborada a partir das recomendações sugeridas pelo Sistema de Recomendação (*SR*), incluindo informações como nota mínima, nota máxima, rating e gênero de cada filme recomendado. Com base nos resultados apresentados, pode-se afirmar que a recomendação gerada é considerada positiva. Isso se deve à identificação de uma correlação entre os filmes, fundamentada nos gêneros e *ratings*, indicando uma coerência nas sugestões fornecidas pelo SR. A análise desses elementos na tabela reforça a eficácia do sistema ao proporcionar recomendações que atendem não apenas aos critérios individuais de avaliação, mas também à relação entre gêneros e preferências de rating, contribuindo para uma experiência mais personalizada ao usuário.

Figura 11 - RMSE X MAE para usuário 5



Fonte: Elaborado pelo autor (2023)

Tabela 4 – Bateria 2 - Recomendações filmes x usuário 1

Título	Rating	Nota Mínima	Nota Máxima	Gênero
Fear City	6.8056	2.5	5.0	Comedy
Eddie Izzard: Dress to kill	6.48845	3.5	5.0	Comedy
Phish: Bittersweet	5.8548	1.0	5.0	Documentary
Margaret's Museum	5.8008	1.0	5.0	Drama
Road Home	5.7723	1.5	5.0	Drama | Romance

Fonte: Elaborado pelo autor (2023)

A Tabela 5 foi elaborada seguindo a mesma abordagem adotada nas Tabelas 1 e demais, sendo construída com base em uma recomendação personalizada destinada ao usuário 3. A análise dessa tabela específica permite concluir que a recomendação foi satisfatória para este usuário em particular, dada a similaridade presente em cada filme indicado. A utilização de uma abordagem personalizada evidencia a capacidade do sistema em compreender e atender às preferências

individuais do usuário 3, resultando em sugestões que se alinham de maneira coerente com seus gostos cinematográficos. Este resultado fortalece a assertividade do Sistema de Recomendação ao proporcionar uma experiência de recomendação personalizada e eficaz para usuários específicos.

Tabela 5 – Bateria 2 - Recomendações filmes x usuário 3

Título	Rating	Nota Mínima	Nota Máxima	Gênero
Virunga	5.3901	0	5.0	Documentary| War
Gerhard Richter	5.3415	0.5	5.0	Documentary
Scatter My Ashes	5.3415	1.5	5.0	Documentary
Jim Henson’s	5.0688	0.5	5.0	Fantasy
Dolls	5.065	3.0	5.0	Drama | Romance

Fonte: Elaborado pelo autor (2023)

A Tabela 6 foi elaborada com o intuito de apresentar *ratings* que não se distanciam significativamente do esperado, considerando a amostra de usuários. Isso resulta em recomendações consideradas positivas para o usuário 5, destacando uma alta similaridade entre os gêneros dos filmes recomendados para este usuário na referida amostra. Diante desse contexto, é possível afirmar que, para esse usuário específico, a recomendação foi considerada relevante, uma vez que os filmes indicados demonstram uma coerência e afinidade em relação aos seus interesses cinematográficos. Essa análise reforça a eficácia do Sistema de Recomendação ao proporcionar sugestões alinhadas com as preferências individuais, contribuindo para uma experiência de recomendação personalizada e satisfatória.

Conforme descrito na Tabela 7, foi apresentada a matriz transposta, indicando as recomendações de usuários para um filme específico, neste caso, "*Sense & Sensibility*". Após a realização dos testes aplicados, os *ratings* obtidos se distanciaram ligeiramente dos resultados esperados. Diante desse contexto, para esse filme específico, é possível afirmar que a recomendação não foi tão relevante, uma vez que os *ratings* propostos não alcançaram a proximidade desejada com os resultados apresentados. Essa análise sugere a necessidade de uma revisão ou

ajuste no Sistema de Recomendação para melhor alinhar as sugestões de usuários a filmes específicos, visando aprimorar a precisão e a eficácia das recomendações para esse cenário particular.

Tabela 6 – Bateria 2 – Recomendações filmes x usuário 5

Título	Rating	Nota Mínima	Nota Máxima	Gênero
Earth	5.7326	3.0	5.0	Documentary
Virunga	5.5675	0	5.0	Documentary | War
Scatter My Ashes	5.3617	1.5	5.0	Documentary
Gerhard Richter	5.3617	2.0	5.0	Documentary
Phish	5.3086	1.0	5.0	Documentary

Fonte: Elaborado pelo autor (2023)

Tabela 7 – Bateria 2 – Recomendações usuários x filme 17

sense & sensibility	Drama | Romance

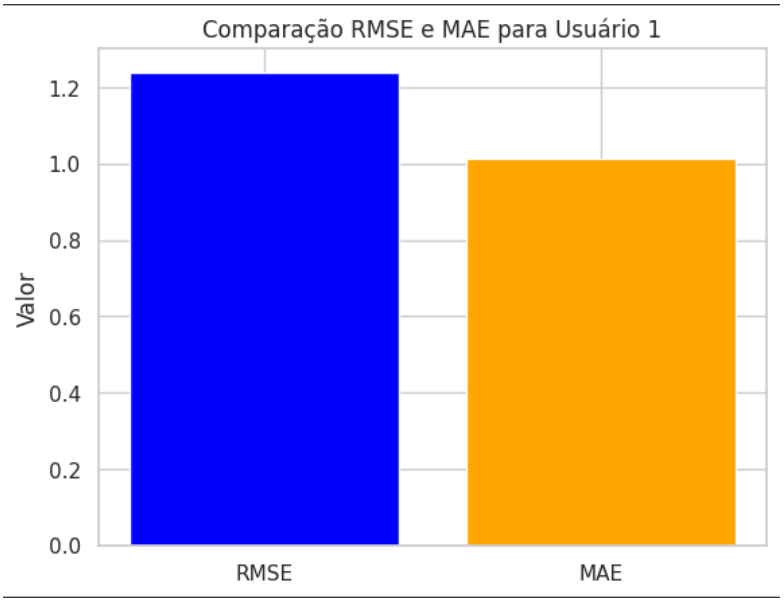
UserId	Rating	Nota Mínima	Nota Máxima	Total de avaliações
869	6.7020	1.0	5.0	20
2912	6.3003	0.5	5.0	24
773	5.9403	1.0	5.0	30
1809	5.9358	1.0	5.0	24
336	5.9047	0.5	5.0	31

Fonte: Elaborado pelo autor (2023)

Conforme evidenciado na Figura 12, observa-se que o conjunto de filmes recomendado para o usuário 1 não foi devidamente assertiva, conforme apresentado pelos resultados das métricas aplicadas. Com um *RMSE* superior a 1,2 (*RMSE* = 1,2623) e um *MAE* de 1,1643, os resultados obtidos se distanciam consideravelmente do ideal, que seria um valor inferior a 1,0. Essa análise aponta para a complexidade enfrentada pelo Sistema de Recomendação ao gerar sugestões para o usuário 1, indicando a necessidade de ajustes ou otimizações para melhorar a precisão das recomendações e se aproximar dos padrões estabelecidos

como ideais.

Figura 12 - Comparação RMSE X MAE para usuário 1



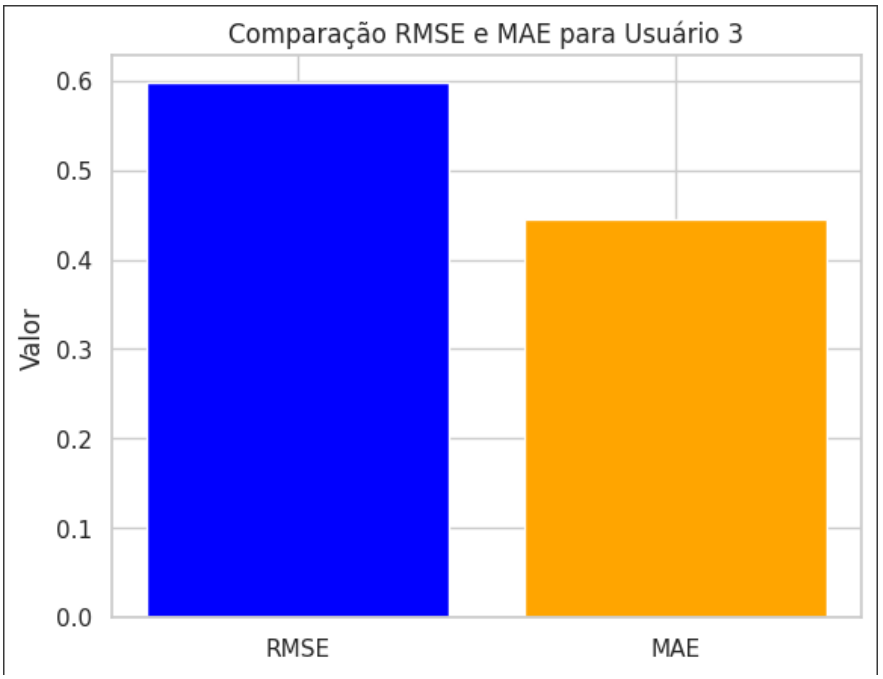
Fonte: Elaborado pelo autor (2023)

Na Figura 13, são apresentados os resultados das métricas avaliativas para o usuário 3, considerando RMSE e MAE. Observa-se que o usuário 3 obteve resultados satisfatórios, com métricas de RMSE igual a 0,5985 e MAE igual a 0,4315, indicando uma proximidade notável de ambos os valores em relação a 0. Esses resultados sugerem que o Sistema de Recomendação alcançou uma precisão elevada ao prever as avaliações para o usuário 3, com variações mínimas nos ratings. Essa análise destaca a eficácia do sistema na personalização de recomendações, proporcionando uma experiência mais consistente e alinhada às preferências do usuário 3.

Na Figura 14, são apresentados os resultados dos testes referentes ao usuário 5, com a análise das métricas de avaliação RMSE e MAE. Constata-se que para o usuário 5 os resultados não foram satisfatórios, uma vez que as métricas revelam valores de RMSE igual a 1,3165 e MAE igual a 0,8495, os quais se distanciam consideravelmente do alvo estabelecido. Essa disparidade sugere que o Sistema de Recomendação presentou dificuldades ao prever com precisão as avaliações do usuário 5, podendo resultar em variações nos *ratings* de até 1 ponto. Diante desse cenário, é recomendável a revisão e ajuste do sistema para melhor

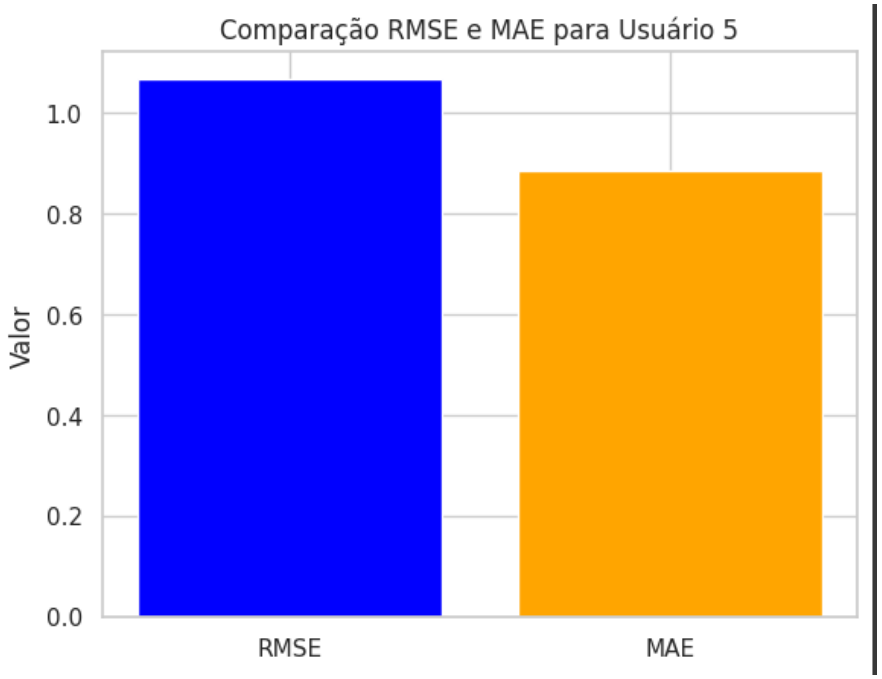
atender às expectativas e padrões desejados de acurácia nas recomendações para o referido usuário.

Figura 13 - Comparação RMSE X MAE para usuário 3



Fonte: Do autor (2023)

Figura 14 - Comparação RMSE X MAE para usuário 5



Fonte: Elaborado pelo autor (2023)

6 CONCLUSÃO

Conclui-se que diante das duas baterias de testes apresentada tendo a primeira utilizando o modelo, a divisão do *dataset* com 70% da base para treinamento e 30% para teste, e empregando os seguintes hiperparâmetros: *maxIter*: 10, *rank*: 50 e *regParam*: 0.15 apresentou uma acurácia satisfatória. As métricas *RMSE* e *MAE* foram registradas como 0,4953 e 0,3840 respectivamente para o usuário 3 aproximando-se consideravelmente do valor ideal 0 para essas métricas.

Já a segunda bateria utilizando-se da divisão do *dataset* com 80% da base para treinamento e 20% para teste, e empregando os seguintes hiperparâmetros: *maxIter*: 5 e *regParam*: 0.01 não apresentando resultados adequados. Com métricas *RMSE* e *MAE* igual à 1,3165 e 0,8495 se distanciando muito em até um ponto do valor desejado. Nota-se que não é utilizado o hipermetro *rank*.

Percebendo que a por meio das métricas de monitoramento e avaliação que o modelo da primeira bateria de testes apresentou uma melhor adaptação aos dados.

O objetivo principal foi alcançado, possibilitando a recomendação personalizada para usuários do conjunto de dados, fundamentada em seus históricos de preferências. A análise foi conduzida mediante o uso de uma tabela de associação de usuário-item. A abordagem adotada consistiu em fatorar uma matriz extensa em uma representação menor por meio de mínimos quadrados alternados. Essa técnica visa preencher os itens não avaliados por determinados usuários com base nas escolhas de usuários similares.

A partir desta abordagem, tornou-se evidente que o Sistema de Recomendação é eficientemente aplicável em diversas áreas, independentemente do produto, podendo abranger filmes, livros, conteúdos, sites e até mesmo algoritmos, como foco principal deste trabalho.

Quanto à usabilidade do usuário, a implementação do *SR* com *Apache Spark* revelou-se capaz de gerar recomendações personalizadas, fundamentadas no histórico e avaliação de um grupo específico. Isso evita a busca manual por novos filmes, uma vez que o sistema pode fornecer novas indicações dentro da plataforma.

Como perspectivas para trabalhos futuros, sugere-se explorar outros algoritmos visando aprimorar os hiperparâmetros, como a aplicação de validação cruzada. Dada a escala de avaliações dos usuários de 1 a 5, é possível aprimorar

esses números ajustando esses valores. Além disso, aprofundar os conhecimentos teóricos sobre técnicas de processamento e tratamento de dados é recomendado. Se possível, a realização de parcerias com instituições detentoras de dados pode contribuir para uma experiência ainda mais aprimorada para o usuário.

7 REFERÊNCIAS

ALEXBUTGIT. **Criar, avaliar e classificar um sistema de recomendação - Microsoft Fabric**. Disponível em: <https://learn.microsoft.com/pt-br/fabric/data-science/retail-recommend-model>. Acesso em: 21 nov. 2023.

Bishop, C. M. (2006). **Pattern Recognition and Machine Learning**. Springer.

SCHAFER, Schafer, Ben & J, Ben & Frankowski, Dan & Dan, & Herlocker, & Jon, & Shilad, & Sen, Shilad. (2007). **Collaborative Filtering Recommender Systems**.

ZISOPOULOS, Charilaos & Karagiannidis, Savvas & Demirtsoglou, Georgios & Antaris, Stefanos. (2008). **Content-Based Recommendation Systems**.

Collaborative Filtering - Spark 2.2.0 Documentation. Disponível em: <https://spark.apache.org/docs/2.2.0/ml-collaborative-filtering.html>. Acesso em: 21 nov. 2023

Collaborative Filtering - **Spark 3.4.0 Documentation**. Disponível em: <https://spark.apache.org/docs/latest/ml-collaborative-filtering.html>. Acesso em: 21 nov. 2023

DONATI, A. **Concepção e Desenvolvimento de um Sistema de Recomendação para o Varejo Físico**. 2018. 83 p. Monografia (Engenharia de Controle e Automação) — Universidade Federal de Santa Catarina. Disponível em:https://repositorio.ufsc.br/bitstream/handle/123456789/200001/PFC%20Andrei%20Donati_2018-2.pdf?sequence=1&isAllowed=y. Acesso em: 10/12/2019. Citado na página 15.

Flach, P. (2012). **Machine Learning: The Art and Science of Algorithms that Make Sense of Data**. Cambridge University Press.

GRAVA, A. P. et al. **Sistema de recomendação de artigos científicos utilizando dados sociai**s. 2016. Dissertação (Mestrado) — Universidade de São Paulo. Disponível em: http://www.teses.usp.br/teses/disponiveis/100/100131/tde-26072016-160726/. Citado 3 vezes nas páginas 13, 25 e 26.

GIL, A. A. C. **Como elaborar projetos de pesquisa**. [s.l.] Éditeur: São Paulo: Atlas, 2010.

GOSH, S. et al. **Recommendation System for E-commerce Using Alternating Least Squares (ALS) on Apache Spark**. Advances in Intelligent Systems and Computing, p. 880–893, 2021.

HERLOCKER, J. et al. **Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems, v. 22, p. 5 – 53, 01 2004**. Citado na página 22.

JÚNIOR, C. DE O. **Prevendo Números: Entendendo as métricas R2, MAE, MAPE, MSE e RMSE**. Disponível em: <https://medium.com/data-hackers/prevendo-n%C3%BAmeros-entendendo-m%C3%A9tricas-de-regress%C3%A3o-35545e011e70>.

KO, H. et al. **A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields**. Electronics, v. 11, n. 1, p. 141, 3 jan. 2022.

LIAO, K. **Prototyping a Recommender System Step by Step Part 2: Alternating Least Square (ALS) Matrix….** Disponível em: <https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-2-alternating-least-square-als-matrix-4a76c58714a1>.

MATOS, D. **Apache Spark e Data Science.** Disponível em: <https://www.cienciaedados.com/apache-spark-e-data-science/>. Acesso em: 30 out. 2023.

MATOS, D. **Por que Cientistas de Dados escolhem Python?** Disponível em: <https://www.cienciaedados.com/por-que-cientistas-de-dados-escolhem-python/>.

MIQUIDO. **We know what you like! Perks of recommendation systems in business**. Disponível em: <https://medium.com/swlh/we-know-what-you-like-perks-of-recommendation-systems-in-business-5f227bb6d09>.

Murphy, K. P. (2012). **Machine Learning: A Probabilistic Perspective**. MIT Press.

O que é Machine Learning? Disponível em: <https://www.oracle.com/br/artificial-intelligence/machine-learning/what-is-machine-learning/>.

PENG, Y. **A Survey on Modern Recommendation System based on Big Data**. arXiv (Cornell University), 30 maio 2022.

RUSSELL, S. J.; **NORVIG, P. Artificial intelligence :** a modern approach. Boston: Pearson, Cop, 2010.

ROY, D.; DUTTA, M. **A systematic review and research perspective on recommender systems.** Journal of Big Data, v. 9, n. 1, 3 maio 2022.

SALUNKE, T.; NICHITE, U. **Recommender Systems in E-commerce**. [s.l: s.n.]. Disponível em: <https://arxiv.org/ftp/arxiv/papers/2212/2212.13910.pdf>. Acesso em: 24 ago. 2023.
SANTANA, M. **Deep Learning para Sistemas de Recomendação (Parte 1) — Introdução**. Disponível em: <https://medium.com/data-hackers/deep-learning-para-sistemas-de-recomenda%C3%A7%C3%A3o-parte-1-introdu%C3%A7%C3%A3o-b19a896c471e>.

SIMEONE, O. **A Very Brief Introduction to Machine Learning With Applications to Communication Systems.** IEEE Transactions on Cognitive Communications and Networking**,** v. 4, n. 4, p. 648–664, dez. 2018.

SINGH, P. K. et al. Recommender systems: an overview, research trends, and future directions. **International Journal of Business and Systems Research**, v. 15, n. 1, p. 14, 2021.

SOUZA, B. D. F. M. E. et al. **MATRIX FACTORIZATION MODELS FOR VIDEO RECOMMENDATION**. 2011. Dissertação (Mestrado) — PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO. Disponível em: http://www.maxwell.vrac.pucrio.br/Busca_etds.php?strSecao=resultado&nrSeq=19273@1. Citado 3 vezes nas páginas 19, 24 e 25.

TIWARI, S. Crafting Recommendation Engine in PySpark. Disponível em: <https://medium.com/analytics-vidhya/crafting-recommendation-engine-in-pyspark-a7ca242ad40a>. Acesso em: 5 nov. 2023.

“The Movie Database.” Themoviedb.org, 2018, www.themoviedb.org/. Accessado e Nov. 2023.

WAZLAWICK,RAUL. **METODOLOGIA DE PESQUISA PARA CIÊNCIA DA COMPUTAÇÃO**. [s.l.] Elsevier, 2009.

What is GroupLens? Disponível em: <https://grouplens.org/about/what-is-grouplens/>. Acesso em: 5 nov. 2023

Yeung, Chi Ho. (2015). Do recommender systems benefit users?.

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
GABINETE DO REITOR

Av. Universitária, 1069 ● Setor Universitário
Caixa Postal 86 ● CEP 74605-010
Goiânia ● Goiás ● Brasil
Fone: (62) 3946.1000
www.pucgoias.edu.br ● reitoria@pucgoias.edu.br

RESOLUÇÃO n° 038/2020 – CEPE

ANEXO I

APÊNDICE ao TCC

Termo de autorização de publicação de produção acadêmica

O(A) estudante Hiago Laurenço Donha
do Curso de Ciência da Computação, matrícula 2017.1.0028.0009-04,
telefone: 62 982527601 e-mail hiagodefenix@gmail.com,
na qualidade de titular dos direitos autorais, em consonância com a Lei nº 9.610/98 (Lei dos Direitos do Autor), autoriza a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o Trabalho de Conclusão de Curso intitulado SISTEMA DE RECOMENDAÇÃO DE FILMES ATRAVÉS DO ALGORITMO ALS
, gratuitamente, sem ressarcimento dos direitos autorais, por 5 (cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial de computadores, no formato especificado (Texto(PDF); Imagem (GIF ou JPEG); Som (WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros, específicos da área; para fins de leitura e/ou impressão pela internet, a título de divulgação da produção científica gerada nos cursos de graduação da PUC Goiás.

Goiânia, 12 de Dezembro de 2023.

Assinatura do autor: Hiago Laurenço Donha

Nome completo do autor: Hiago Laurenço Donha

Assinatura do professor–orientador:

Nome completo do professor-orientador: Anibal Santos Jukemura