UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO

Eduardo Brandão Tavares

Sugestão de livros baseada em algoritmo híbrido de recomendação e grau de interesse recente

Recife

Maio de 2023

Eduardo Brandão Tavares

Sugestão de livros baseada em algoritmo híbrido de recomendação e grau de interesse recente

Artigo apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Aprovado em: 26 de Maio de 2023.

BANCA EXAMINADORA

Silvana Bocanegra (Orientador)
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Marcelo Gama
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Sugestão de livros baseada em algoritmo híbrido de recomendação e grau de interesse recente

Eduardo Brandão [1], **Silvana Bocanegra** [1]

[1]Departamento de Estatística e Informática – Universidade Federal Rural de Pernambuco
Rua Dom Manuel de Medeiros, s/n, - CEP: 52171-900 – Recife – PE – Brasil

eduardo.brandao@ufrpe.br, silvana.bocanegra@ufrpe.br

Resumo. *Com a vasta e crescente gama de livros disponíveis, escolher a próxima leitura se tornou um trabalho complexo em meio a tantas opções. No contexto do Brasil, onde boa parte dos leitores tem que escolher bem qual livro comprar, devido ao baixo poder de compra da nossa população, uma recomendação assertiva se tornou mais valorosa. Neste artigo é apresentado um algoritmo de recomendação de livros baseado em um modelo híbrido, que consiste em utilizar tanto técnicas relacionadas a regras de associação, quanto técnicas que se baseiam no conteúdo dos livros, visando apresentar livros desconhecidos e que acompanhem o interesse recente leitor. O modelo conseguiu atingir uma precisão equiparável a outros modelos nas métricas RMSE e MAE e entrega recomendações bastante relacionadas com as últimas leituras de cada leitor.*

Abstract. *With a vast and growing range of books available, choose your next reading can become a complex job amid so many options. In the context of Brazil, where most readers have to choose well which book to buy, due to the low purchasing power of our population, an assertive recommendation has become more valuable. This article presents a book recommendation algorithm based on a hybrid model, which consists of using both techniques related to association rules and techniques that are based on the content of books, aiming to present unknown books that follow the recent interest of the reader. The model managed to reach an accuracy comparable to other models in the RMSE and MAE metrics and delivers recommendations closely related to the last readings of each reader.*

1. Introdução

No ano de 2021, a porcentagem de pessoas com acesso a internet no Brasil era de 80,2% e essa porcentagem cresceu para 84,3% em 2023 [DataReportal 2023], devido ao crescimento do número de pessoas utilizando a internet, a quantidade de informações disponíveis se torna cada vez maior, o que por sua vez ocasiona a necessidade de uma melhor filtragem do conteúdo consumido.

Em um experimento realizado no ano 2000 pelos psicólogos Sheena Iyengar (Universidade de Columbia) e Mark Lepper (Universidade de Stanford) [Iyengar 2000], foram feitos dois testes num mercado de alimentos. No primeiro, foram exibidos 24 potes de sabores diferentes de geléia e no segundo, foram exibidos apenas 6 sabores. Foi observado que uma grande variedade de escolhas atraiu muitos consumidores, porém, a taxa de compra foi muito baixa, já no segundo teste, menos consumidores se interessaram pelos potes, mas entre os que se interessaram a taxa de compra foi bem expressiva.

Figura 1. Paradoxo da escolha.

Fonte: https://www.yourmarketingrules.com/the-paradox-of-choice.

Essa diferença de comportamento é justificada na psicologia pelo fato de que uma escolha complexa, tende a diminuir as chances de compra, logo uma escolha que pode ser feita de maneira mais simples, obtém melhores resultados.

No contexto atual, uma excelente forma de realizar filtragem de conteúdo assertiva, é através da utilização de sistemas de recomendação [Ricci 2011]. Essa técnica utiliza informações sobre usuários e sobre o conteúdo que eles consomem.

Dentro da abordagem dos sistemas de recomendação, existem três formas mais populares de realizar as recomendações [Ricci 2011]: A primeira é chamada filtragem colaborativa e se baseia em avaliações dos usuários para determinar regras de associação que serão usadas nas recomendações; Na segunda forma é feita a recomendação baseada em conteúdo, que define suas sugestões a partir de características dos usuários e dos itens como por exemplo título do livro e idade do leitor; Por último, temos a abordagem híbrida que usa elementos tanto da filtragem colaborativa quanto da recomendação baseada em conteúdo. Nessa última abordagem são minimizados os problemas de dificuldade de previsão diante da escassez de dados e o "cold start"que é a dificuldade de fazer uma recomendação para um novo usuário da base de dados que avaliou uma quantidade pequena de livros.

Um sistema de recomendação podem ser usados para os mais variados objetivos, neste trabalho será usado para recomendação de livros. A motivação que levou à escolha do problema surgiu a partir da percepção da importância da leitura, pois ela impacta positivamente na capacidade cognitiva [Clark 2013] e na redução de níveis de estresse [Chiles 2009] do leitor. Uma recomendação mais assertiva pode aumentar o interesse pela leitura, segundo pesquisa do instituto Pró-livro: 53% dos leitores desistem do livro antes do fim se não gostarem da leitura e apenas 25% dos leitores afirmam que não receberam nenhum tipo de indicação em relação ao último livro lido [InstitutoProLivro 2022], o que reforça a escolha dos sistemas de recomendação para o contexto do trabalho.

Objetivo geral

O objetivo deste trabalho é desenvolver um modelo de recomendação que permita a um usuário conhecer livros com base em suas preferências. Utilizando um sistema de recomendação híbrido, que combina a abordagem colaborativa com a utilização da recomendação baseada em conteúdo.

Objetivos específicos

1. Atingir o equilíbrio das métricas RMSE, MSE e coeficiente de similaridade recente, para que as recomendações realizadas sejam precisas e tenham correlação com os últimos livros lidos pelo usuário;
2. Desenvolver um modelo com a capacidade de capturar os interesses recentes do usuário, para fazer recomendações que levem em conta o momento atual.

As demais seções do artigo estão organizadas como segue: a Seção 2 apresenta os trabalhos relacionados sobre sistemas de recomendação; A teoria utilizada está descrita na Seção 3; A construção de um algoritmo de recomendação de livros baseada em um modelo híbrido é apresentada na Seção 4; Na Seção 5 estão os resultados experimentais. Por fim são apresentadas as conclusões e trabalhos futuros.

2. Trabalhos relacionados

Diversos esforços têm sido aplicados para melhorar o desempenho de sistemas de recomendação. Em 2009 a Netflix concedeu um prêmio de um milhão de dólares a uma equipe de desenvolvedores por um algoritmo que aumentou a precisão do mecanismo de recomendação da empresa em 10% [JOHNSTON 2012]. A evolução no desempenho de tais sistemas pode ser acompanhada também pelos artigos publicados sobre o tema. Xiaoyuan Su [Su and Khoshgoftaar 2009] atuou na identificação dos principais problemas dos sistemas que usam filtragem colaborativa, como escassez de dados e escalabilidade, além disso, propôs técnicas que podem ser úteis para contornar tais problemas.

Zhi Hui Wang e De Zhi Hou[Wang and Hou 2021] propuseram uma melhoria na qualidade de um modelo que utiliza a abordagem dos filtros colaborativos, através da implementação de um grau de interesse dos livros, que refere-se aos atributos do próprio livro, incluindo tempos de busca, intervalos de empréstimo, tempos de empréstimo e tempos de renovação [Wang and Hou 2021]. Esse grau de interesse é usado em conjunto com os interesses do próprio usuário para fazer as recomendações.

Yonghong Tian [Tian 2019] em um trabalho de comparação entre diferentes modelos, obteve resultados que demonstram que os métodos híbridos podem fornecer recomendações mais precisas do que as abordagens puras. Uma das formas de unir duas abordagens em um sistema híbrido é através da atribuição de pesos às avaliações, Geetha G. propôs em seu trabalho [Geetha 2018] um modelo de recomendação de filmes baseado na combinação da filtragem colaborativa com a baseada em conteúdo. No modelo vetores com as previsões de cada usuário são computadas de acordo com o conteúdo dos filmes. A partir desses vetores são calculadas as similaridades entre os usuários que são aplicadas como pesos sobre as previsões com o objetivo de recuperar o vetor de previsões final para cada usuário.

Em relação aos trabalhos apresentados nesta seção, o trabalho aqui apresentado diferencia-se principalmente no modo como os dados são inseridos no modelo e na maneira como a filtragem baseada em conteúdo é unificada com a filtragem colaborativa.

3. Referencial teórico

Nesta seção são apresentados os conceitos pertinentes aos sistemas de recomendação, bem como suas diversas abordagens. Adicionalmente, técnicas matemáticas são destacadas como formas de extrair informações relevantes dos dados. Por fim, são discutidas as diferentes estratégias para avaliação dos modelos utilizados no contexto dos sistemas de recomendação.

3.1. Sistemas de recomendação

Os sistemas de recomendação são ferramentas e técnicas de software que fornecem sugestões de itens a serem usados por um usuário. As sugestões fornecidas visam apoiar os usuários em vários processos de tomada de decisão, como quais itens comprar, qual música ouvir. Os sistemas de recomendação provaram ser meios valiosos para os usuários lidarem com a sobrecarga de informações e se tornaram uma das ferramentas mais poderosas e populares no comércio eletrônico [Ricci 2011].

O trabalho de recomendação de itens para usuários pode ser feito utilizando o aprendizado de máquina, que por sua vez pode ser dividido fundamentalmente em 3 tipos: aprendizagem supervisionada, não supervisionada e por reforço. Neste trabalho será utilizada a abordagem supervisionada, que ocorre quando o modelo é treinado sobre uma base de dados onde já são apresentadas as respostas desejadas para cada observação, por exemplo:

casaId	**numero**$_{quartos}$	**tamanho**$_{m}2$	**preço**
1	2	50	200000
2	3	37	170000
3	1	43	x

Tabela 1. Exemplo aprendizagem supervisionada.

Na Tabela 1 é apresentado um conjunto de dados ilustrativo com dados de 3 casas em uma mesma vizinhança. Se o objetivo for realizar a previsão do preço para a terceira casa, deve ser implementado um modelo de aprendizagem supervisionada, pois seu treinamento será realizado sobre dados onde a resposta (preço) já foi informada para alguns exemplos, como a casa 1 e a 2.

Para um modelo de recomendação ser bem sucedido é necessária uma série de informações sobre usuários e itens. Algumas informações podem se referir apenas ao usuário como idade ou cidade, outras informações podem se referir apenas ao item, como por exemplo gênero ou número de páginas. No entanto a disposição de informações mais utilizada nesse tipo de modelo é uma matriz que correlaciona os itens com os usuários através de avaliações.

userId	1 bookId)	2 (bookId)	3 bookId)
1	4	5	0
2	0	5	3

Tabela 2. Matriz de avaliações.

Na Tabela 2 são correlacionados usuários (userId) com livros (bookId), os zeros são os livros que usuário da linha correspondente não leu, enquanto os outros valores são as notas dadas pelo usuário. A partir dos dados coletados, é realizado o treinamento do modelo. Nesta etapa os dados são divididos em base de treino e base de teste. A base de treino servirá para o modelo extrair padrões dos dados, enquanto a base de testes servirá para avaliar a performance.

Por último, para avaliar um modelo de aprendizagem de máquina existem diversas formas que serão contempladas em mais detalhes ao decorrer deste artigo, mas falando de maneira geral, a avaliação no caso da aprendizagem supervisionada é positiva quando os resultados obtidos se assemelham com as respostas previamente anotadas.

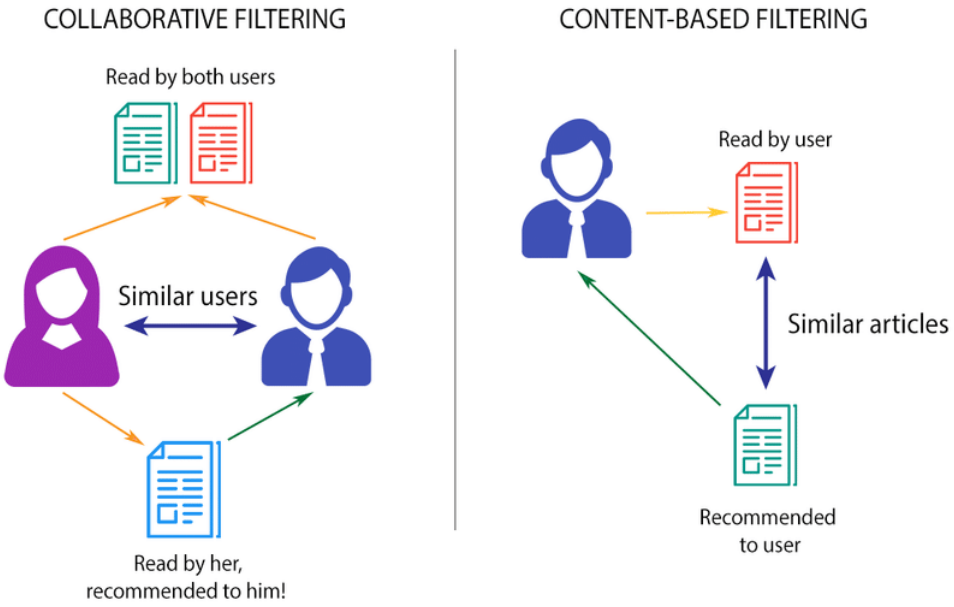
3.2. Técnicas para extração de informação

O funcionamento do sistema de recomendação, depende do quão bem o modelo é capaz de extrair informações dos dados para sugerir itens, esses dados são usados para treinar o modelo para que ele consiga identificar padrões nos dados e assim recomendar itens para novos usuários. As características dos dados dependem muito de qual tipo de sistema será implementado. Para extrair esse conjunto de informações, existem duas principais abordagens [Ricci 2011]:

1. Filtragem baseada em conteúdo: Usa similaridades entre os itens para recomendar ao usuário itens parecidos com os itens que ele já mostrou gostar. Neste trabalho, poderia ser usada a sinopse dos livros, ou até mesmo comentários sobre eles para ajudar na recomendação.

2. Filtragem colaborativa: Usa os comportamentos dos usuários para gerar as recomendações, ou seja, se todos os usuários que gostam do livro A, também gostam do livro B. Essa regra pode ser usada na recomendação mesmo que os livros não tenham conteúdo similar.

Figura 2. Filtragem colaborativa e por conteúdo.



Fonte: https://www.researchgate.net/figure/Content-based-filtering-vs-Collaborative-filtering-Source_fig5_323726564.

No caso da filtragem colaborativa, entre os principais destaques estão a maior presença de itens menos populares nas recomendações e a habilidade de recomendar livros não tão similares em conteúdo em relação aos livros lidos pelo usuário. Em contrapartida, uma grande base de dados precisa ser mantida visto que são necessárias informações históricas das avaliações dos usuários. Outro problema conhecido nessa abordagem é o "cold start", que acontece quando um novo usuário é inserido nos dados. De início ele terá poucas avaliações e para um filtro colaborativo é bem difícil recomendar assertivamente para um usuário que avaliou apenas um ou dois livros.

A filtragem baseada em conteúdo têm vantagens como a menor necessidade de armazenamento de dados, já que não é necessário manter as informações históricas dos usuários; as recomendações são comumente relevantes ao usuário pois conseguem medir com exatidão a semelhança de cada item. Outro ponto importante é que esse método não é tão afetado pelo "cold start", já que com poucas avaliações é possível dar recomendações que façam sentido. Quanto às desvantagens podem ser citadas três principais:

- Falta de novidade nas recomendações: Se um usuário avaliar positivamente o primeiro livro da saga harry potter, é possível que as recomendações nesse modelo sejam todos os outros livros da saga. Provavelmente o usuário já conhecia os outros livros da sequência e não necessitava recomendação, apesar de estar correta.
- Desafio da escalabilidade: Toda vez que um livro novo entrar no catálogo, é necessário coletar e armazenar todas as características dele que são usadas no modelo.
- Características podem estar incorretas: Para exemplificar considere um modelo de filtragem baseada em conteúdo, que utiliza a informação do gênero dos livros como entrada. Pode ser que tenha havido algum erro na hora de anotar os gêneros e no banco de dados agora existe um livro de terror que está marcado com a categoria comédia, inconsistências como essa podem atrapalhar o modelo.

Tendo em vista que ambas as abordagens têm vantagens e desvantagens, é possível uní-las para retirar o melhor de cada uma, usando os chamados modelos híbridos. As abordagens híbridas têm três estratégias, a primeira é fazendo filtragem baseada em conteúdo e colaborativa separadamente e depois combinar os resultados de ambas; a segunda consiste em adicionar recursos da filtragem de conteúdo em uma abordagem de filtragem colaborativa (e vice-versa); na terceira as duas abordagens são unificadas em um só modelo. [Ricci 2011].

Existem diversas técnicas que podem ser utilizadas num modelo baseado em conteúdo, a técnica escolhida dependerá principalmente de quais dados o modelo terá como entrada. Para recomendação de livros e filmes é comum o uso da sinopse como informação contudo, textos não podem servir de entrada para sistemas recomendações, então é necessário transformá-los em entradas numéricas. A técnica escolhida neste trabalho para realizar a transformação foi a TFIDF [Lamblet 2022], sigla para "Term Frequency - Inverse Document Frequency". É uma técnica comumente utilizada em processamento de linguagem natural para calcular a relevância de uma palavra em um documento, em relação a um conjunto de documentos.

A técnica de TFDF leva em consideração a frequência de ocorrência de uma palavra em um documento (TF) e a frequência de ocorrência da palavra em todos os documentos do conjunto (IDF). Isso ajuda a compensar a importância de palavras comuns que aparecem em muitos documentos e, portanto, não são particularmente discriminativas, enquanto dá maior peso a palavras que aparecem com menos frequência em documentos específicos. A TFIDF foi utilizada sobre a base de dados de sinopses dos livros transformando-a em uma matriz com o formato apresentado na Tabela 3, onde cada linha representa um livro e cada coluna uma palavra, essas palavras são todas as que estão presentes em todas as sinopses. Já as células foram calculadas de acordo com a seguinte função:

$$TFIDF = \mathbf{tf(i,j)} \cdot log(\frac{\mathbf{N}}{\mathbf{df(i)}})$$

Onde tfij é o número de vezes que a palavra i aparece na sinopse do livro j, N é o número total de livros e dfi é o número de sinopses que contém a palavra i.

bookId	palavra 0	...	palavra n
0	0.0471	...	0.0701
1	0.0841	...	0.0866
2	0.1122	...	0.1122

Tabela 3. Matriz TFIDF.

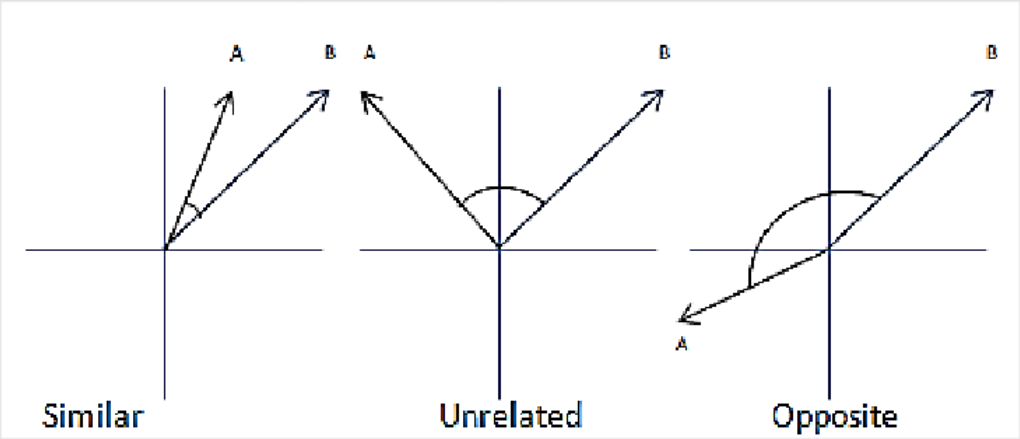
O cálculo do TFDF para uma palavra em um determinado documento é feito multiplicando o TF (frequência da palavra no documento) pelo IDF (frequência inversa da palavra em todos os documentos do conjunto). Quanto maior o resultado do cálculo, mais relevante a palavra é considerada para aquele documento em particular. Para o caso abordado neste trabalho, o cálculo resultará em uma matriz onde cada linha representa um

livro e as colunas são as palavras presentes em todos os livros, com base nessa matriz é possível conseguir as similaridades entre os livros.

Neste trabalho foi aplicada a similaridade cosseno, devido a sua boa performance em matrizes esparsas, já que para um livro existirão várias palavras que não estão presente em sua sinopse, criando a existência de um valor zero na matriz. [1]. A similaridade cosseno é uma medida comumente utilizada para avaliar a semelhança entre vetores em espaços vetoriais. Ela é amplamente aplicada em áreas como recuperação de informações, mineração de texto, aprendizado de máquina e processamento de linguagem natural.

A similaridade cosseno é baseada no conceito de ângulo entre vetores. Ela mede o quanto dois vetores estão alinhados na mesma direção, independentemente de sua magnitude. Essa medida é especialmente útil quando se lida com dados esparsos ou de alta dimensionalidade, como documentos de texto.

Figura 3. Similaridade cosseno.



Fonte: https://www.researchgate.net/figure/Cosine-Similarity-and-Cosine-Distance-Functions_fig2_348968927.

O resultado da similaridade cosseno, é uma matriz que correlaciona os vetores (as sinopses dos livros na forma de matriz TFIDF) conforme apresentado na Tabela 4.

bookId	1 (bookId)	2 (bookId)	3 (bookId)
1	1	0.5	0.9
2	0.5	1	0.6
3	0.9	0.6	1

Tabela 4. Matriz de similaridade.

Para calcular a similaridade cosseno, consideramos dois vetores x e y. Os vetores podem representar documentos, palavras, itens ou qualquer outra entidade que possa ser modelada como um vetor no espaço. A fórmula para o cálculo da similaridade cosseno é a seguinte:

[1]Uma matriz é dita esparsa quando possui uma grande quantidade de elementos com valor zero (ou não presentes, ou não necessários)

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

em que A e B são vetores que representam as linhas na matriz TFIDF. Uma das principais vantagens da similaridade cosseno é a sua invariância em relação às escalas dos vetores. Ela considera apenas a direção e não a magnitude dos vetores, o que a torna robusta em relação a diferenças de escala entre os dados.

Além disso, a similaridade cosseno é computacionalmente eficiente e simples de ser implementada. Ela pode ser facilmente calculada usando operações básicas de álgebra linear, como multiplicação de vetores e cálculo de normas.

Já para a filtragem colaborativa, uma das formas conhecidas de retirar informações é utilizando o SVD (singular value decomposition) [Salakhutdinov and Mnih 2005], um algoritmo que opera sobre matrizes e neste trabalho terá como entrada a matriz de avaliação dos usuários, que passará pelas seguintes etapas:

- Fatoração da matriz: fatorar em três novas matrizes: U, S e V. U e V representam os vetores singulares à esquerda e à direita, respectivamente, enquanto S representa os valores singulares.
- Escolha do número de dimensões: definir o hiperparâmetro do algoritmo que determina quantas dimensões serão mantidas na matriz, visando capturar uma boa gama de informações com um número reduzido de dimensões.
- Reconstrução da matriz: Reconstruir a matriz original usando as dimensões retidas produzirá uma representação de dimensão inferior da matriz original.
- Geração de recomendações: Usar a matriz reconstruída para gerar recomendações personalizadas para cada usuário. Especificamente, o algoritmo prevê a classificação que um usuário daria a um item que ainda não foi avaliado, calculando o produto escalar dos vetores de fator do usuário e do item.

Outras técnicas que podem ser utilizadas são:

- KNN: O algoritmo utiliza um conjunto de dados de treinamento com as avaliações de usuários sobre itens para determinar a semelhança entre eles. Em seguida, ele utiliza essa informação para gerar recomendações para um determinado usuário. O processo de recomendação começa com a identificação dos k usuários mais similares ao usuário alvo, com base em suas avaliações. Em seguida, o algoritmo seleciona os itens mais bem avaliados por esses k usuários e os recomenda ao usuário alvo.
- Slope One: O algoritmo calcula a diferença média de avaliação entre cada par de itens avaliados pelo usuário e utiliza esses valores para gerar previsões de avaliação para itens que o usuário ainda não avaliou. Para isso, ele utiliza uma fórmula simples de ponderação para estimar as avaliações faltantes..
- Co clustering: O algoritmo utiliza uma matriz de avaliações de usuários sobre itens para identificar grupos de usuários e itens que apresentam padrões de avaliação semelhantes. Em seguida, ele gera previsões de avaliação para itens que um usuário ainda não avaliou com base nas avaliações de outros usuários que pertencem ao mesmo grupo.

3.3. Métricas de desempenho

Para avaliar a eficácia do modelo em relação ao seu objetivo proposto, é preciso estabelecer métricas de desempenho. Tais métricas correspondem a cálculos aplicados sobre os resultados gerados pelo modelo, com o intuito de avaliar a precisão, diversidade e relevância dos mesmos.

- MAE: Utilizado para avaliar modelos de regressão calculando a diferença entre o valor predito e o valor real, tomando o valor absoluto dessa diferença e depois calculando a média desses valores ao longo de todas as amostras. Uma pontuação mais baixa de MAE indica que o modelo tem melhor desempenho em termos de precisão das previsões.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|Predicted_i - \hat{Actual}_i|$$

  Onde Predicted é a nota prevista pelo modelo e Actual é a nota real para aquela avaliação. O MAE tem a vantagem de ser menos sensível a outliers em comparação com o RMSE, pois não eleva os erros ao quadrado. No entanto, ele não considera a magnitude dos erros e, portanto, pode não refletir adequadamente a importância relativa dos erros no modelo.
- RMSE: Tomando como base os artigos já referenciados anteriormente a principal métrica de desempenho utilizada foi a RMSE que é uma variação da métrica MSE (Mean squared error). A principal diferença entre as duas é que RMSE pune os erros com maior rigor, pois eleva as diferenças entre os valores previstos e os reais ao quadrado, seguindo a fórmula:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Predicted_i - Actual_i)^2}$$

  Onde Predicted é a nota prevista pelo modelo e Actual é a nota real para aquela avaliação.
- Coeficiente de similaridade recente: Essa métrica serve para identificar se os livros recomendados estão indo de acordo com o interesse recente demonstrado pelo usuário. O coeficiente será obtido ao tirar a média das similaridades de cada livro recomendado ao usuário, em relação aos últimos livros lidos.
- Grau de novidade: se refere à capacidade do sistema de apresentar ao usuário novos itens que ele ainda não conhecia, mas que são relevantes para seus interesses. Um sistema de recomendação eficiente deve ser capaz de equilibrar a oferta de itens já conhecidos pelo usuário com novas opções que possam surpreendê-lo e ampliar seu repertório.

4. Abordagem proposta

Neste trabalho foram usadas trê bases de dados, todas coletadas diretamente do site Skoob [Skoob 2022], que é uma rede social onde os leitores podem compartilhar reviews, notas e comentários sobre livros, além de registrar quais seus livros favoritos, os que eles desejam ler e quais já foram lidos. A primeira base contém informações referentes a nota que cada

usuário deu a cada livro, a segunda, informações sobre a sinopse de cada livro e por último uma com os dados de quais foram os quatro últimos livros lidos de cada usuário.

Para a construção do modelo foi escolhida uma abordagem híbrida, ou seja, que usa elementos tanto da filtragem colaborativa quanto da recomendação baseada em conteúdo. A estrutura geral do modelo pode ser resumida na Figura 4:

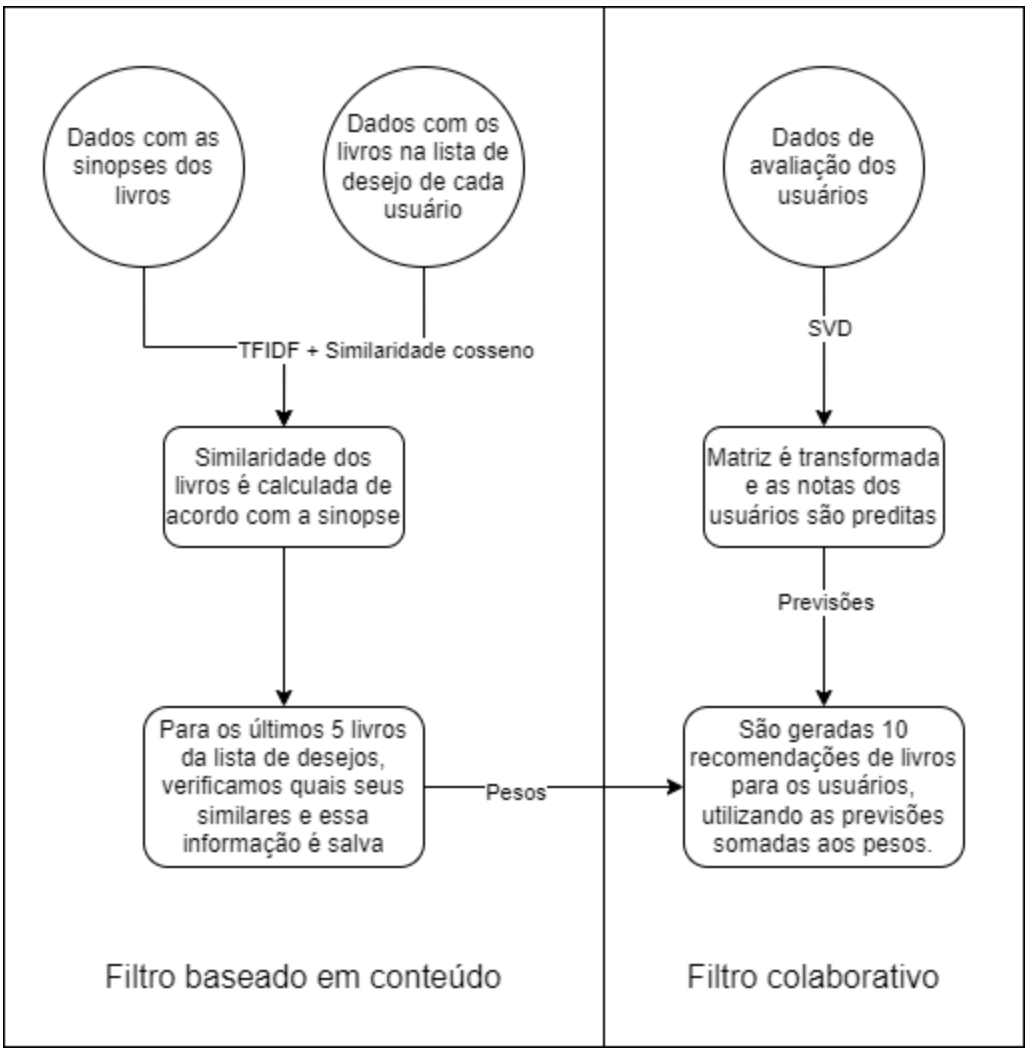


Figura 4. Estrutura do modelo.

A solução pode ser dividida em duas etapas, na primeira as similaridades dos livros serão calculadas através das técnicas TFIDF e similaridade cosseno, que serão aplicadas sobre as sinopses. Os livros que forem similares aos últimos livros adicionados pelo usuário na lista de lidos do Skoob, receberão pesos a serem somados na nota prevista. Na segunda parte, o SVD será aplicado na base de dados com as notas dos usuários, com a finalidade de prever as notas que um usuário daria a cada livro não avaliado, é nessa parte que entram os pesos da primeira etapa. Após a predição das notas, as dez maiores são selecionadas e exibidas como resultado.

O trabalho foi desenvolvido através da utilização do python, uma linguagem de programação amplamente usada em problemas de machine learning [Python 2022], foram

usadas também algumas bibliotecas auxiliares da linguagem, todo o código gerado está disponível em [Brandão 2023]. Na primeira etapa do experimento foram coletados dados do site Skoob. Para a coleta, foram utilizadas as bibliotecas requests e concurrent.features, a primeira delas foi utilizada para fazer requisições ao banco de dados do site.

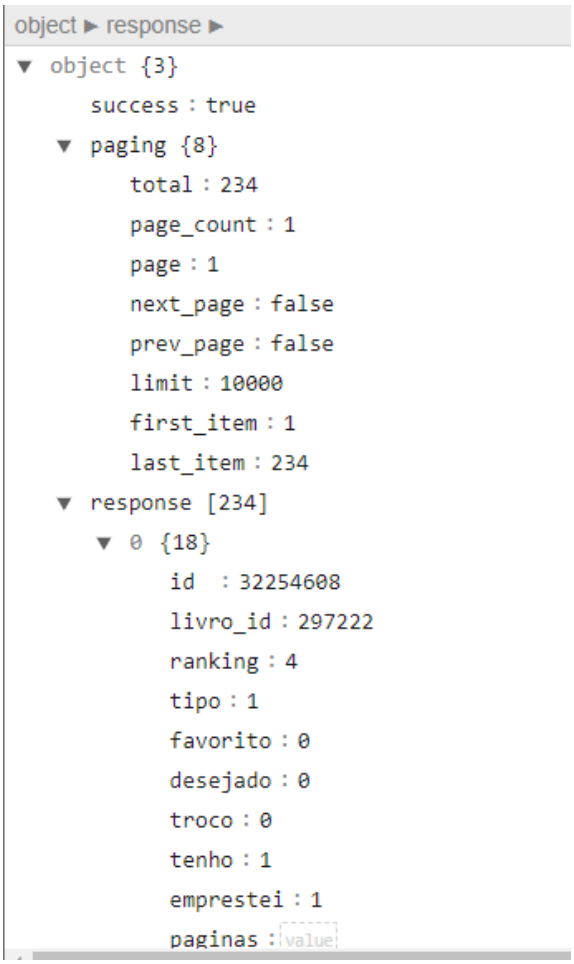


Figura 5. Json resposta.

Cada usuário da plataforma Skoob, têm um Json no formato apresentado na Figura 5. Segundo dados fornecidos pelo próprio site em 2020, existiam sete milhões de usuários cadastrados naquele momento, mesmo com a informação de que nem todos os usuários fizeram avaliações de livros, foi considerada a população como sete milhões.

Para obter as informações necessárias, foi feita uma amostragem aleatória de onde foram obtidas 6337621 avaliações, contendo 74174 usuários e 336432 livros. A partir do Json de cada usuário foram inseridas informações em um csv.

Já a biblioteca concurrent.features permitiu que essas requisições fossem feitas de maneira paralela, reduzindo o tempo necessário para coletar os dados. Nessa coleta foram armazenadas as informações conforme a Tabela 5:

userId	bookId	evaluation
237	523	4
15	436	5

Tabela 5. Csv com dados das avaliações.

Onde o userID é o identificador único de cada usuário da plataforma, o BookId, o identificador único do livro e o evaluation é a nota que o usuário deu ao livro.

userId	bookId	order
37	523	0
37	436	1
37	543	2
37	56	3
37	12	4
10	566	0
10	1	1
10	62	2
10	741	3
10	25	4

Tabela 6. Csv com informações da lista de lidos do usuário.

A Tabela 6 ilustra o arquivo CSV com as informações dos usuários. É possível ver quais foram os últimos livros lidos pelo usuário na plataforma Skoob, uma linha que tem o número 0 na coluna order, significa que aquele foi o último livro lido.

Para finalizar a etapa de coleta de dados, foram recuperadas informações referentes aos livros. As principais informações são o nome do livro, o id e as respectivas sinopses conforme a Tabela 7.

bookId	name	sinopse
27	Duna	livro sobre x
635	Drácula	livro sobre y

Tabela 7. Csv com as informações dos livros.

Após carregados os dados de avaliações dos usuários, foi feita a parte de análise exploratória, de acordo com os seguintes passos:

1. Entendimento dos dados: Para analisar os dados foi necessário ver como eles estavam distribuídos, com isso foi possível notar que alguns usuários avaliaram poucos livros e alguns livros foram avaliados por poucos usuários.
2. Remoção dos livros e dos usuários que foram avaliados menos de 50 vezes: Para que um filtro colaborativo consiga encontrar os padrões de similaridades de um

livro em relação aos outros, é interessante que o mesmo tenha sido avaliado diversas vezes. Para usuários a regra é a mesma, usuários com um maior número de avaliações realizadas são contemplados com melhores recomendações.
A base de dados foi reduzida de 6337621 avaliações para 256057, e passou a contar com 8086 livros e 390 usuários, tornando mais viável também o treinamento do modelo.

Após a estruturação dos dados, o primeiro passo foi gerar a matriz de similaridade entre os livros com base nas sinopses de cada um. Primeiro transformando os dados em uma matriz TFIDF e depois usando a similaridade cosseno sobre a matriz.

Para cada previsão gerada pelo modelo, foi atribuído um peso correspondente à soma das medidas de similaridade entre o livro a ser recomendado e os quatro últimos livros lidos pelo usuário. Tal processo teve como resultado a recomendação de livros com conteúdo mais similar aos últimos livros lidos pelo usuário.

Na segunda etapa ocorre a modelagem do filtro colaborativo, que tem como insumo a base de avaliações que contém apenas dados numéricos, permitindo que a técnica SVD seja aplicada diretamente nos dados.

Com os modelos treinados, a etapa final do experimento foi gerar 10 recomendações para cada usuário, é nessa fase que entra o coeficiente de interesse momentâneo; cada livro tem sua nota prevista aumentada com base no quão similar o livro é em relação aos livros da lista de lidos. Por último, são coletadas as métricas de avaliação.

5. Resultados

Na execução do modelo, foi utilizada uma base de dados com 256057 avaliações de usuários cujo os dados foram divididos em base de treino e base de testes, com 25% dos dados ficando na base de teste. A função da base de treino é fazer com que o modelo seja capaz de identificar padrões nos dados, enquanto a base de testes é utilizada para avaliar o modelo. As métricas apresentadas na sessão 3.3 foram calculadas sobre a base de testes.

O modelo realiza a previsão da nota de um usuário em relação a um livro com base nos dados de teste e com as previsões computadas é possível avaliar a qualidade do modelo. Para escolher o algoritmo utilizado para prever as notas, foram levadas em conta as métricas RMSE e MAE. Quanto mais próximos de zero, melhor a qualidade do modelo.

Algoritmo	RMSE	MAE
SVD	0.8246	0.6556
KNN	0.8318	0.6535
Slope One	0.8422	0.6652
Co-clustering	0.8636	0.6819

Tabela 8. Resultados dos diferentes algoritmos.

A Tabela 8 indica uma superioridade dos modelos SVD e KNN, ambos apresentaram um MAE semelhante, mas quanto a RMSE o SVD se provou superior e portanto, foi o algoritmo escolhido nesse trabalho. Com as métricas calculadas é possível comparar o

algoritmo com os resultados de outros trabalhos para ter um melhor parâmetro em relação aos valores apresentados em cada métrica.

	Modelo	RMSE
1	Bayesian timesSVD++ flipped [Steffen Rendle and Koren 2019]	0.7485
2	Bayesian timesSVD++ [Steffen Rendle and Koren 2019]	0.7523
3	Bayesian SVD++ [Steffen Rendle and Koren 2019]	0.7563
4	MRMA [Dongsheng Li and Chu 2017]	0.7634
5	SparseFC [Lorenz K. Muller and Indiveri 2018]	0.7690

Tabela 9. Melhores trabalhos na base MovieLens.

A Tabela 9 [Mov 2023] exibe os cinco modelos de melhor desempenho na base de dados MovieLens 10M, a qual se encontra previamente depurada, estruturada e pronta para ser empregada como entrada em um sistema de recomendação. Para cada modelo, foi aplicada uma técnica distinta:

1. Os três primeiros modelos foram apresentados no mesmo trabalho e utilizam uma estrutura semelhante com algumas pequenas variações, todos eles utilizam a técnica de fatoração de matriz bayesiana mas apenas os modelos 1 e 2 adicionam uma variável de tempo à fatoração. Nos dois primeiros modelos o timesSVD é aplicado sobre dados de palavras presentes nos filmes, porém no modelo 1 foi utilizada uma matriz de dados mais completa como entrada.
2. Utiliza sobre o conjunto de dados a positive matrix factorization (PMF), que tem como objetivo fatorar matriz de avaliações de usuários por itens em duas matrizes de menor dimensão: uma matriz de usuários por fatores e uma matriz de fatores por itens; juntamente com a PMF é usada a técnica Mixture-Rank Matrix Approximation (MMRA) que assume que as avaliações de usuários por itens podem ser modeladas como uma mistura de distribuições de probabilidade, onde cada distribuição representa um grupo de usuários com preferências semelhantes.
3. Introduz uma arquitetura de redes neurais em que matrizes de peso são reparametrizadas em termos de vetores de baixa dimensão, interagindo através de funções kernel.

1. Atingir o equilíbrio das métricas RMSE, MSE e coeficiente de similaridade recente, para que as recomendações realizadas sejam precisas e tenham correlação com os últimos livros lidos pelo usuário;
2. Desenvolver um modelo com a capacidade de capturar os interesses recentes do usuário, para fazer recomendações que levem em conta o momento atual.

Para a base de dados da Netflix os melhores RMSE são próximos a 0.85 [Steffen Rendle and Anderson 2020], essa diferença ocorre devido a facilidade encontrada na base MovieLens.

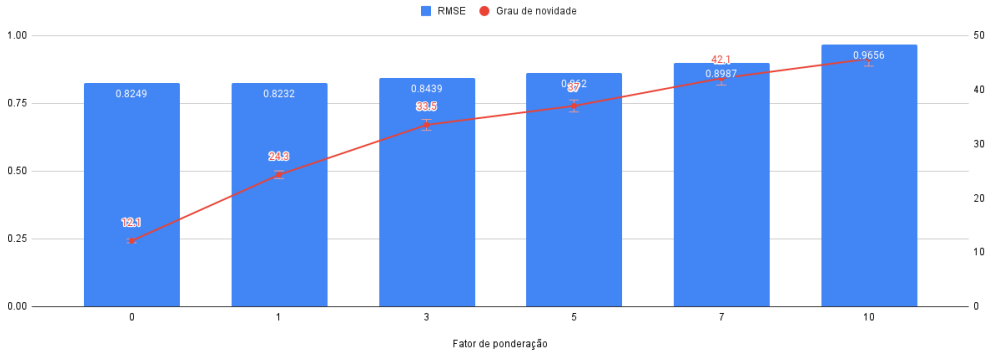
Ao compararmos com modelos propostos em outros artigos, esse trabalho é superado em termos de RMSE e MAE, contudo mantém um valor próximo aos apresentados, como podemos observar na Tabela 10.

Algoritmo	RMSE	MAE
SVD [Paterek 2007]	0.8986	-
timesSVD++ [Koren 2009]	0.8799	-
Matrix Factorization [Shuai Zhang and Zhu 2018]	0.803	0.616
HIN Embedding [Chuan Shi and Yu 2017]	0.7414	0.5741
SVD [Brandão 2023]	0.8246	0.6556

Tabela 10. Comparação com outros trabalhos.

Na segunda fase do modelo, foi realizado um experimento com os pesos do filtro baseado em conteúdo, visando manter o RMSE (Erro Médio Quadrático) e ao mesmo tempo melhorar a similaridade das 10 principais recomendações em relação aos últimos livros lidos pelo usuário. Para aprimorar o modelo, foi implementada uma estratégia em que a nota predita para um livro é acrescida do valor da similaridade desse livro em relação aos quatro últimos livros lidos pelo usuário, multiplicado por um fator de ponderação.

Figura 6. Análise das métricas.



No eixo x da Figura 6 estão presentes os fatores de ponderação testados. É possível observar que entre o fator 0 (sem pesos aplicados) e o fator 1, há um aumento do coeficiente de similaridade sem afetar negativamente o RMSE, já entre os fatores 1 e 3 já é possível ver uma redução do valor da métrica.

6. Conclusões

Este artigo propõe um algoritmo de recomendação baseado em sistema de recomendação híbrido e interesse momentâneo do usuário. Na fase da filtragem colaborativa, é usado o SVD para análise, enquanto no cálculo do interesse momentâneo, é utilizada a similaridade cosseno sobre a matriz com as sinopses de cada livro. A análise foi feita sobre uma base de dados obtida através de um webscraping realizado no site "skoob"e para avaliar o modelo, foram usadas as métricas "RMSE", "MSE"e "coeficiente de similaridade recente".

A partir dos resultados, podemos extrair algumas informações, dentre ela estão o fato de dois modelos podem mesmo obtendo valores semelhantes no que se refere a métricas de avaliação, apresentar resultados completamente diferentes e a avaliação sobre qual dos dois performou melhor, cabe a área de negócio analisar. Além do MAE e

RMSE que foram apresentadas nesse trabalho, é interessante estar atento a métricas como o "grau de novidade", pois com o auxílio delas podemos fazer recomendações de livros que os usuários não conheciam, mas que poderiam gerar interesse a partir do momento que fossem recomendados.

Os resultados apresentados são ultrapassados em questão de métricas por diversos outros artigos que podemos encontrar e que usam mais algoritmos dentro do modelo híbrido, convergindo para a ideia de que modelos híbridos tendem a performar melhor. Ao mesmo tempo, é possível observar uma dificuldade de medir a qualidade real do modelo através das métricas, dois modelos que apresentam o mesmo RMSE podem fazer recomendações bastante diferentes, ou seja, para verificar o real valor de um modelo é necessário inseri-lo no ambiente de produção. O principal benefício que pudemos observar foi uma maior presença de livros similares aos últimos que o usuário demonstrou interesse, tornando o artigo bem sucedido na aplicação do coeficiente que se propôs a fazer.

Para melhorar o trabalho feito, podemos utilizar três caminhos distintos:

- Otimização de hiperparâmetros: Na fase de filtragem colaborativa existem parâmetros que devem ser escolhidos arbitrariamente para ajudar o modelo nos resultados, como por exemplo as dimensões do SVD que serão mantidas.
- Cálculo de similaridade recente: O cálculo usado para definir o quanto uma recomendação está relacionada com o interesse momentâneo do usuário pode ser feita de diversas maneiras, como a aplicação de novas fórmulas ou o ajuste dos pesos a fim de obter melhores recomendações.
- Seleção de modelos, técnicas e baselines: Nesse artigo, foram utilizados apenas dois algoritmos e de maneira concorrente para compará-los, mas em outros artigos podemos observar a melhora de desempenho quando são utilizados diversos algoritmos em conjunto para gerar recomendações. Também é possível inserir outras técnicas, pesos e baselines no cálculo para melhorar o resultado.
- Análise de novos dados: Com a coleta de um número maior de dados, é possível extrair informações diferentes e consequentemente obter uma melhora no resultado, um exemplo de dado novo a ser utilizado são as sinopses de cada livro, que podem ajudar na fase de calcular a similaridade baseada em conteúdo.
- Avaliação online: O melhor meio de garantir que um sistema de recomendação está funcionando bem, é avaliando o mesmo em produção, pois podemos ter dados sobre o quão impactante ele está sendo na escolha dos livros, se os usuários estão avaliando positivamente as recomendações e outros fatores como por exemplo, o quanto demora para um novo usuário começar a ter recomendações de qualidade.

Agradecimentos

Agradeço a Deus por permitir mais essa conquista em minha vida.

Aos meus pais, minha irmã e toda minha família pelo o incentivo durante o curso.

A minha namorada, Cinthia, por toda a companhia e força que me deu.

Aos meus amigos, tanto os que fiz durante a graduação quanto aos que estão comigo desde antes.

A minha orientadora, Silvana, por todo o apoio e todos os conselhos durante o desenvolvimento do trabalho.

Referências

(2023). Modelos novielens. url = https://paperswithcode.com/sota/collaborative-filtering-on-movielens-10m.

Brandão, E. (2023). Codigo git. url = https://github.com/E-Brandao/RecommendationSystem.

Chiles, A. (2009). Reading can help reduce stress, according to university of sussex research. urlhttps://www.theargus.co.uk/news/4245076.reading-can-help-reduce-stress-according-to-university-of-sussex-research/.

Chuan Shi, Binbin Hu, W. X. Z. and Yu, P. S. (2017). Heterogeneous information network embedding for recommendation.

Clark, C. (2013). A novel look at how stories may change the brain. urlhttp://esciencecommons.blogspot.com/2013/12/a-novel-look-at-how-stories-may-change.html.

DataReportal (2023). globalreportoverview. urlhttps://datareportal.com/reports/digital-2023-global-overview-report.

Dongsheng Li, Chao Chen, W. L. W. L. T. L. N. G. and Chu, S. M. (2017). Mixture-rank matrix approximation for collaborative filtering.

Geetha, G. (2018). A hybrid approach using collaborative filtering and content based filtering for recommender system. IOP Publishing Ltd.

InstitutoProLivro (2022). A 5ª edição da retratos da leitura no brasil. urlhttps://www.prolivro.org.br/5a-edicao-de-retratos-da-leitura-no-brasil-2/a-pesquisa-5a-edicao/.

Iyengar, S. S. (2000). When choice is demotivating: Can one desire too much of a good thing? Columbia University.

JOHNSTON, C. (2012). Netflix never used its $1 million algorithm due to engineering costs. urlhttps://www.wired.com/2012/04/netflix-prize-costs/.

Koren, Y. (2009). Collaborative filtering with temporal dynamics.

Lamblet, A. (2022). Medium tfidf. url = https://medium.com/data-hackers/tf-idf-algoritmo-de-recomenda%C3%A7%C3%A3o-6c3cbd55e439.

Lorenz K. Muller, J. N. M. and Indiveri, G. (2018). Kernelized synaptic weight matrices.

Paterek, A. (2007). Improving regularized singular value decomposition for collaborative filtering.

Python (2022). Site python. url = https://www.python.org/.

Ricci, F. (2011). Recommender systems handbook. Springer.

Salakhutdinov, R. and Mnih, A. (2005). Probabilistic matrix factorization. University of Toronto.

Shuai Zhang, Lina Yao, Y. T. X. X. X. Z. and Zhu, L. (2018). Metric factorization: Recommendation beyond matrix factorization.

Skoob (2022). Site skoob. url = https://www.skoob.com.br/.

Steffen Rendle, L. Z. and Koren, Y. (2019). On the difficulty of evaluating baselines a study on recommender systems.

Steffen Rendle, Walid Krichene, L. Z. and Anderson, J. (2020). Neural collaborative filtering vs. matrix factorization revisited.

Su, X. and Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques", advances in artificial intelligence. Hindawi Publishing Corporation.

Tian, Y. (2019). College library personalized recommendation system based on hybrid recommendation algorithm. Elsevier.

Wang, Z. H. and Hou, D. Z. (2021). Research on book recommendation algorithm based on collaborative filtering and interest degree. Hindawi Publishing Corporation.