

BookMatch: Sistema de Recomendação de Livros Baseado em Filtragem Colaborativa por Item

Alécio Ferreira V. Júnior e Fernando Gonzaga M. Oliveira

Resumo - Sistemas de recomendação são utilizados em diversas áreas atualmente e com várias propostas diferentes. Ao selecionar um produto qualquer em um site, por exemplo, o usuário gera uma informação que provavelmente será manipulada a fim estabelecer um perfil individual para ele mesmo. A criação desse perfil possibilita sugestões mais precisas para o cliente final, viabilizando vantagens para todas as partes. A ideia do projeto é utilizar métodos conhecidos de filtragem colaborativa por item, análise de conteúdo, ferramentas web e uma base de dados preestabelecida para: criar um site dinâmico onde o usuário pode aplicar filtros determinados a fim de obter informações sobre os dados presentes; desenvolver um algoritmo de recomendação que seja capaz de sugerir livros que um usuário possa se interessar.

Abstract - Recommender systems are currently used in several areas and with several different proposals. When selecting any product on a website, for example, the user generates information that will probably be manipulated in order to establish an individual profile for himself. The creation of this profile enables more accurate suggestions for the end customer, enabling advantages for all parties. The idea of the project is to use known methods of collaborative filtering per item, content analysis, web tools and a pre-established database to: create a dynamic website where the user can apply certain filters in order to obtain information about the present data; develop a recommendation algorithm that is able to suggest books that a user might be interested in.

Palavras-chave: Sistema; recomendação; dados; usuários; HTML; JavaScript; livros.

I. INTRODUÇÃO

Com o crescente volume de informações disponíveis na internet, encontrar o livro ideal pode ser uma tarefa desafiadora para os leitores. Diante desse cenário, os sistemas de recomendação têm se mostrado uma ferramenta valiosa para auxiliar os usuários a descobrir novas obras com base em seus interesses pessoais e, inclusive, nos próprios produtos/obras. Esses sistemas podem utilizar algoritmos e técnicas de filtragem colaborativa para analisar padrões de avaliação/interesse e fornecer recomendações personalizadas e relevantes.

Neste artigo, apresentamos um sistema de recomendação de livros baseado em filtragem colaborativa por item, que visa oferecer aos usuários sugestões precisas, otimizando a experiência de busca por novas leituras. O sistema proposto combina informações sobre as avaliações de usuários e livros, coletados de uma grande base de dados.

Serão aplicadas técnicas de filtragem colaborativa que se baseiam na similaridade de avaliações entre livros para fazer recomendações. Além disso, é utilizada uma abordagem de filtragem baseada em conteúdo, onde são analisados atributos dos livros, como gênero, autor, sinopse e avaliações de outros leitores. Em suma, este artigo apresenta um sistema de recomendação que oferece uma abordagem para ajudar os usuários a encontrar livros semelhantes com algum título escolhido. Os resultados obtidos mostram o potencial do sistema em fornecer recomendações, contribuindo para aprimorar a descoberta e a experiência de leitura dos usuários.

II. REVISÃO BIBLIOGRÁFICA

Personalized Recommendation System Based on Product Specification values' (Sistema de recomendação personalizado com base nos valores de especificação do produto, 2006), foi o primeiro artigo que utilizamos. No documento, um grupo desenvolve um sistema de recomendação que permite uma relação entre o usuário e o próprio sistema, usando um algoritmo de recomendação baseado em faixa de utilidade para fornecer recomendações mais dinâmicas e personalizadas. O algoritmo faz tomadas de decisão multi-atributo (MADM) para encontrar valores de utilidade de produtos na mesma classe de produtos das empresas, destacando assim, os mais relevantes em cada especificação.

O segundo artigo, 'Intelligent recommender system based on unsupervised machine learning and demographic attributes' (Sistema de recomendação inteligente baseado em aprendizado em máquina não supervisionado e atributos demográficos, 2021), propõe um sistema de recomendação inteligente que relaciona filtragem colaborativa a um algoritmo de aprendizado de máquina não supervisionado (k-means clustering). O trabalho utiliza a idade e sexo como atributos demográficos, criando assim, perfis de usuários segmentados. Os itens de pesquisa foram filmes e, a proposta é que eles sejam agrupados por seus grupos pelo k-means. Os usuários nesse caso serão classificados baseados em sua preferência de gênero cinematográfico. A recomendação de filmes ao usuário é feita através de filtragem colaborativa, que indica um filme para um gênero de afinidade do cliente.

As metodologias descritas acima foram fundamentais para a construção do algoritmo proposto, pois a ideia do projeto

é utilizar métodos conhecidos de filtragem colaborativa por item, análise demográfica - idade e localização como principais - e, assim, desenvolver um algoritmo de recomendação que seja capaz de sugerir livros de interesse ao usuário.

III. METODOLOGIA

BookMatch é um produto final e, neste tópico, detalharemos acerca de todo o processo de estudo e pesquisa necessário para o funcionamento do site.

1 Escolha do Dataset (base de dados)

A base de dados escolhida foi retirada do site Kaggle, coletada por Cai-Nicolas Ziegler durante quatro semanas entre agosto e setembro de 2004, com permissão do CTO (Diretor de Tecnologia - Chief Technology Officer) da Humankind Systems, Ron Hornbaker. No dataset (arquivo onde os dados estão inseridos), existem três arquivos:

- Usuários: onde estão contidos os dados dos usuários de forma anônima, separados em um ID individual, localização (cidade/país) e idade.
- Livros: onde estão contidos os dados de cada um dos livros, separados com um ISBN (Número padrão internacional de livro - International Standard Book Number), título, autor, ano de publicação, editora e imagens da capa das obras, obtidos pela Amazon Web Services. Segue na sequência uma imagem de como os dados eram inicialmente:

	ISBN	book_title	book_author	year_publication	Publisher	image
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press	http://images.amazon.com/images/P/0195153448.0.
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	http://images.amazon.com/images/P/0002005018.0.
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial	http://images.amazon.com/images/P/0060973129.0.
3	0374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Groux	http://images.amazon.com/images/P/0374157065.0.
4	0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton & Company	http://images.amazon.com/images/P/0393045218.0.

Fig. 1. Dados sobre os livros no dataset.

- Avaliações: onde estão contidas as avaliações de cada usuário para seus respectivos livros lidos (a escala de avaliação é de 1-10, onde 0 significa a não leitura da obra).

2 Tratamento dos dados no Colab

Para tratamento dos dados utilizamos a plataforma Colaboratory, desenvolvida pela Google. Os códigos foram elaborados em Python, seguindo os seguintes passos:

1) Importação de bibliotecas

- "numpy" é importado como "np".
- "pandas" é importado como "pd".
- "seaborn" é importado como "sns".

2) Leitura dos conjuntos de dados

Os três conjuntos de dados são lidos a partir do arquivo principal (Books, Ratings e Users). Os dados foram lidos

para novos dataframes e renomeados, a fim de simplificar as sintaxes do código: "df-books", "df-ratings" e "df-users".

3) Encontrando livros mais populares

Os dataframes df-ratings e df-books são mesclados (merged) com base na coluna 'ISBN', criando o dataframe df-ratings-books. O dataframe df-ratings-books é mesclado (merged) com o dataframe df-users com base na coluna 'User-ID', criando o dataframe df-complete. São realizadas operações de agregação para calcular o número de leituras ('num-views-df'), o número de notas ('num-rating-df') e a nota média ('avg-rating-df') para cada livro. Os dataframes são mesclados com base no título do livro ('Book-Title'), criando o dataframe popular-df. Algumas colunas desnecessárias são removidas do dataframe popular-df. A função populares é definida para retornar uma lista dos livros mais populares, com a possibilidade de filtrar por número mínimo de avaliações ('qtd-minima-avaliacoes') e ordenar por métricas como nota média, visualizações ou número de avaliações.

4) Tratando dados para recomendação de livros

O módulo cosineSimilarity é importado da biblioteca sklearn.metrics.pairwise. O dataframe df-ratings é mesclado (merged) com o dataframe df-books com base no 'ISBN', criando o dataframe ratings-with-name. São identificados os usuários que costumam avaliar livros, considerando aqueles que têm pelo menos 200 avaliações ('regular-voters-id'). As avaliações relevantes são filtradas considerando apenas os livros que possuem pelo menos 100 avaliações ('relevant-books'). É criada uma matriz de livros por usuários ('matrix') a partir das avaliações finais. Os valores ausentes na matriz são preenchidos com zero.

Apenas para visualização de padrões na 'matrix', foi gerado um heatmap que demonstra a similaridade entre os livros no intervalo 277° à 297°, onde estão presentes os pontos mais fortes de semelhança no dataset.

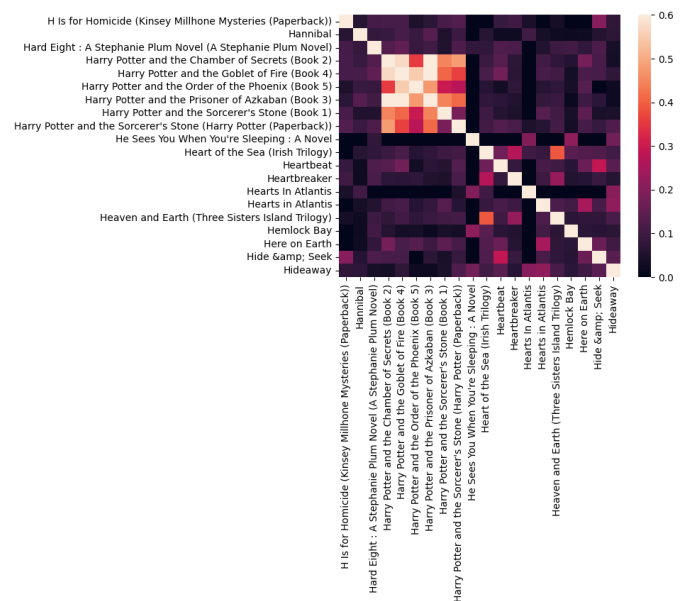


Fig. 2. Heatmap da 'matrix'.

5) Criando a função recomendar()

É definida a função 'recomendar()' que recebe o nome de um livro e a quantidade de recomendações desejadas. A função encontra o índice do livro na matriz e calcula a similaridade desse livro com todos os outros livros. Os livros mais similares são selecionados e suas informações (título, autor(a), ano e imagem) são obtidas do dataframe df-books. As informações dos livros recomendados são retornadas como uma lista de listas.

3 FrontEnd

Com todos os dados devidamente tratados e funções criadas, partimos para a estruturação e implementação do site em si. Utilizando o Visual Studio Code, desenvolvemos o Front End base com HTML (Linguagem de Marcação de Hipertexto - HyperText Markup Language) e CSS (Folhas de Estilo em Cascata - Cascading Style Sheets), criando toda a interface visual. Para permitir uma interação do usuário com o site, implementamos funcionalidades com JavaScript. A figura a seguir demonstra a aparência do site:

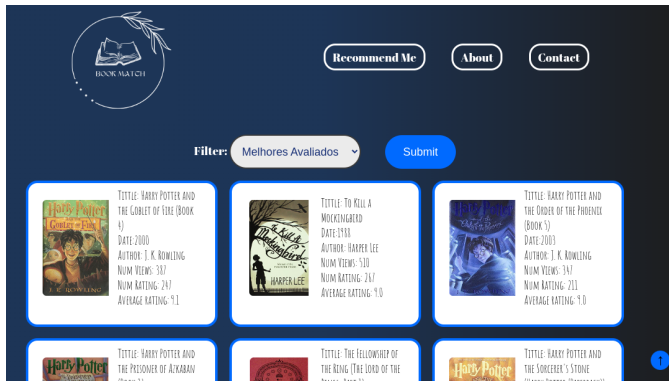


Fig. 3. Página inicial do site BookMatch.

A proposta da **página inicial** é entregar ao usuário uma interface estática e com a possibilidade de filtrar os livros entre: mais avaliados; mais lidos; menos avaliados; menos lidos; melhores avaliados; piores avaliados. Para cada filtro selecionado, existe um arquivo csv que contém 30 livros classificados de acordo com a opção escolhida. Na imagem acima por exemplo, o filtro selecionado solicita os "melhores avaliados", logo, o site recebe o arquivo "melhores-avaliados.csv" e exibe na tela as obras e suas características. Na página "**Recommend Me**", adaptamos o input para receber o nome do livro e inseri-lo no algoritmo de recomendação. O algoritmo explicado anteriormente recomenda três livros semelhantes ao livro digitado, como demonstra a figura a seguir, onde utilizamos o livro "Harry Potter and the Prisoner of Azkaban (Book 3)":

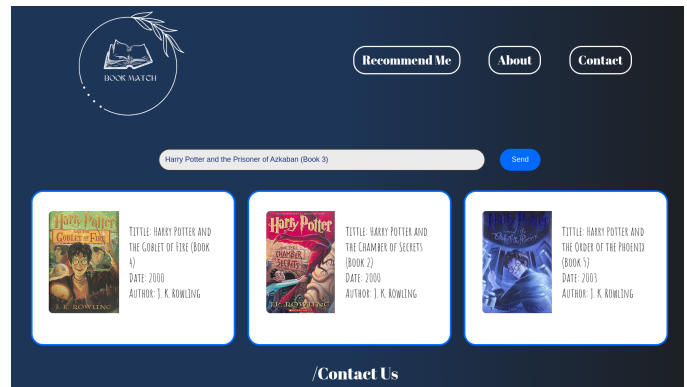


Fig. 4. Recomendação a partir do livro "Harry Potter and the Prisoner of Azkaban (Book 3)"

O restante do site contém uma página **About** dedicada a comentar sobre o projeto em si, e outra página **Contact**, onde inserimos nossas informações de contato.

4 BackEnd

No BackEnd é feita a integração dos códigos gerados em Python no Colab na página. Para a aplicação, foi criado um arquivo principal "app.py" que importa a biblioteca Flask, possibilitando a criação de rotas entre código executável em python e as estruturas em HTML. É viável agora, executar as funções que foram criadas no Colaboratory e utilizá-las no website. Analisando a aba de recomendação, dentro do input existe um script responsável por conectar a string inserida pelo usuário ao backend, que chamará por vez a função de recomendação, entregando ao usuário uma quantidade de livros similares.

IV. CONCLUSÃO

Em resumo, este estudo demonstrou a viabilidade e eficácia da aplicação da filtragem colaborativa por item na construção de um website de recomendação de livros. Através da análise de um dataset predefinido, foi possível identificar padrões de similaridades entre as obras, permitindo a geração de recomendações relevantes. Os resultados obtidos mostraram que o sistema proposto é capaz de fornecer sugestões precisas, possibilitando maior satisfação aos usuários e incentivando a descoberta de novas obras literárias. Com base nesses resultados, é evidente que a utilização da filtragem colaborativa por item em um dataset predefinido pode ser uma abordagem promissora para o desenvolvimento de sistemas de recomendação de livros e pode ser aplicada em diversas outras áreas, proporcionando uma experiência personalizada e altamente relevante para os usuários. O projeto foi disponibilizado e hospedado na plataforma GitHub, e pode ser visualizado a partir do link: <https://github.com/fgonzaga25/BookMatch>.

REFERENCES

- [1] 1. Afoudi Yassine; Lazaar Mohammed; (2021). "Intelligent recommender system based on unsupervised machine learning and demographic attributes".
- [2] Sang Hyun Choi; Sungmin Kang; Young Jun Jeon; (2006). "Personalized recommendation system based on product specification values".
- [3] Seaborn HeatMap. Recuperado de <https://pythonbasics.org/seaborn-heatmap/>.
- [4] Numpy and SciPy. Recuperado de <https://musicinformationretrieval.com/numpybasics.html>.
- [5] CSS Tutorial. Recuperado de <https://www.w3schools.com/css/>.
- [6] GitHub Docs. Recuperado de <https://docs.github.com/pt>.