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, publicadora 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 Giroux	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 Escolha do tipo de Algoritmo de Recomendação ideal

A filtragem colaborativa é uma abordagem amplamente utilizada em sistemas de recomendação que se baseia na análise das interações entre usuários e itens. A premissa subjacente é que usuários semelhantes tendem a ter preferências semelhantes, e itens que foram bem avaliados por usuários semelhantes a um determinado usuário têm maior probabilidade de serem bem avaliados por esse usuário também. Considerando as propriedades da base de dados utilizada no projeto, um Algoritmo de Recomendação baseado em Filtragem Colaborativa acaba se tornando uma das opções mais viáveis para a criação do sistema.

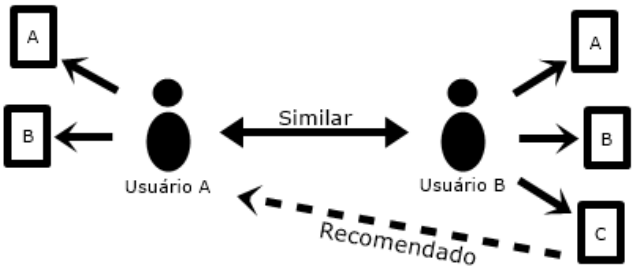

Fig. 2. Ilustração da abordagem por Filtragem Colaborativa

Tendo em vista que o Algoritmo baseado em Filtragem Colaborativa foi escolhido para este projeto, é necessário manipular os dados para atender a um aspecto fundamental deste tipo de algoritmo: obter uma representação matricial das interações entre usuário-item. A representação matricial proposta consiste em uma matriz onde as linhas correspondem aos nomes dos livros, as colunas correspondem aos IDs dos usuários e os valores presentes indicam as avaliações atribuídas por cada usuário a um determinado livro.

Após organizar os dados para atender melhor às necessidades do projeto, é necessário definir uma medida de similaridade que seja capaz de definir, a partir de um livro-alvo, quais seriam os livros mais semelhantes a este. Para o desenvolvimento deste projeto, a medida de similaridade escolhida foi a da Similaridade do Cosseno.

A similaridade de cosseno é uma medida que quantifica a similaridade entre dois vetores em um espaço vetorial. Ela é amplamente utilizada em algoritmos de recomendação baseados em filtragem colaborativa. A ideia por trás dessa medida é calcular o cosseno do ângulo formado pelos dois vetores, onde valores próximos a 1 indicam alta similaridade e valores próximos a 0 indicam baixa similaridade.

Para ilustrar, vamos supor que exista dois usuários (leitores) U_1 e U_2 que tenham lido e avaliado os mesmos três Livros L_1, L_2 e L_3. A nota do primeiro usuário U_1 para os três livros foi 3, 4 e 2, respectivamente. Já o segundo usuário U_2 avaliou os três livros com as notas 1, 4 e 3, respectivamente. Dessa forma, é possível representar cada livro como sendo um conjunto de avaliações (vetor), sendo que L_1 = [3,1], L_2 = [4,4] e L_3 = [2,3]. Fazendo a representação de cada livro no plano como um vetor, é possível encontrar a variação de ângulo entre eles.

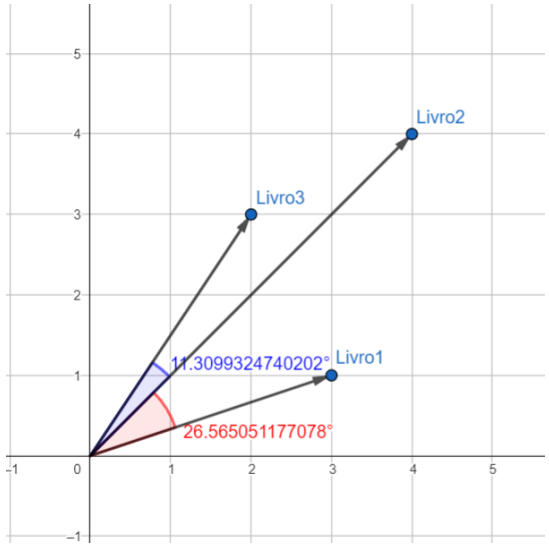

Fig. 3. Representação dos livros no plano e cálculo do ângulo entre eles

Obtendo o valor do cosseno dos ângulos formados entre L_1 e L_2, temos que cos(26.565051177078°) = 0.8944271909999, enquanto para L_2 e L_3, o valor encontrado é de cos(11.3099324740202°) = 0.9805806756909. Dessa forma é possível concluir que L_2 é mais semelhante ao L_3 do que ao L_1, de acordo com as avaliações fornecidas pelos usuários. Esta abordagem será utilizada no decorrer do projeto, com a diferença de que uma representação gráfica seria n-dimensional, sendo n a quantidade de usuários que avaliaram uma quantidade mínima de livros.

3 **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 demostra a similaridade entre os livros no intervalo 277° à 297°, onde estão presentes os pontos mais fortes de semelhança no dataset.

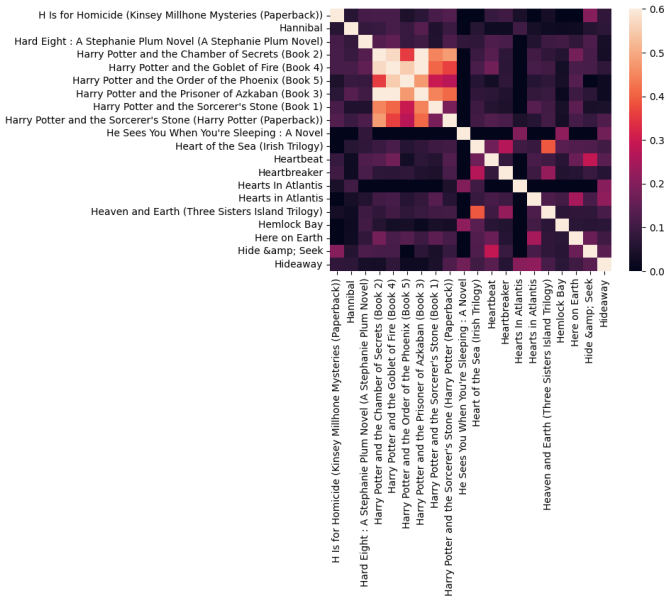

Fig. 4. Heatmap da ’matrix’.

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 **Front-end**

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:

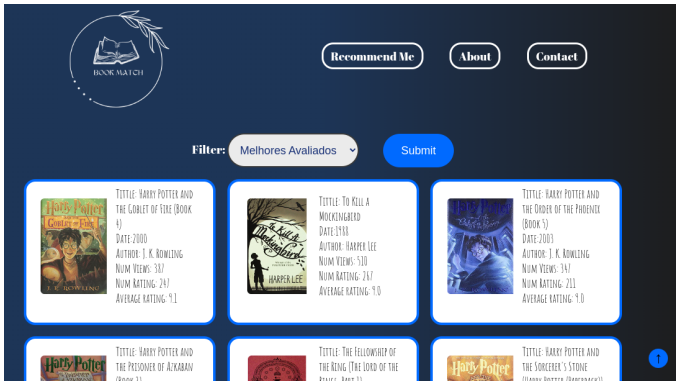

Fig. 5. Página inicial do site BookMatch.

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)”:

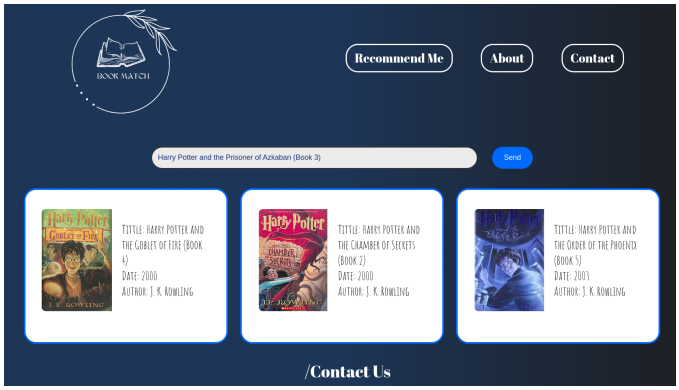

Fig. 6. Recomendação a partir do livro ”Harry Potter and the Prisoner of Azkaban (Book 3)”

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 Back-end

Para atingir as recomendações personalizadas desejadas e fornece-lás ao front-end, foi necessário realizar um desenvolvimento lógico em Python 3.11.3. Com isso, alguns arquivos em ”.py”foram gerados, cada um com uma função específica:

- **”app.py”**: representa um aplicativo web que utiliza o Framework Flask 2.3.2 para lidar com diferentes rotas (URLs) e renderizar os templates desenvolvidos no front-end correspondentes (/index, /recommend, /about, /contact) a essas rotas. Sua simplicidade, documentação completa, flexibilidade e capacidade de padronizar um projeto foram alguns dos motivos que levaram à escolha do Flask.
- **”readCsv.py”**: responsável por ler os arquivos da base de dados (Books.csv, Ratings.csv e Users.csv) armazená-los em objetos DataFrame da biblioteca Pandas 2.0.2. No mesmo código, são geradas cópias desses dataframes, que são atribuídas a novas variáveis, o que possibilita preservar o conjunto de dados originais e realizar alterações apenas em suas cópias.
- **”populars.py”**: realiza várias manipulações e transformações nos dados para obter uma lista dos livros mais populares com base em diferentes critérios de classificação. Ele mescla os dataframes de avaliações, livros e usuários, e em seguida calcula o número de leituras, número de notas e nota média para cada livro. É nesse código que a função ’populares()’ é desenvolvida, possibilitando enviar parâmetros como quantidade de livros a obter, quantidade mínima de avaliações e um critério de filtragem, retornando um dataframe contendo os livros especificados, ordenados de forma crescente ou decrescente (conforme enviado por parâmetro).
- **”generateFilteredBooks.py”**: nesse código, a função ’populares()’ desenvolvida anteriormente é chamada 6 vezes com pequenas diferenças nos parâmetros. O objetivo dessas chamadas é gerar resultados filtrados para seis tipos de filtros diferentes: ”Melhores Médias”, ”Piores Médias”, ”Mais Lidos”, ”Menos Lidos”, ”Mais Avaliados”e ”Menos Avaliados. Logo após, para cada um dos resultados é gerado em um arquivo em formato .csv (*Comma-Separated Values*) que será posteriormente utilizado no front-end. A escolha por essa metodologia se sustenta na ideia de que, devido ao fato de que o sistema não possui a possibilidade do usuário fornecer um feedback, esses dados seriam fixos e não precisariam ser gerados toda vez que um filtro fosse aplicado. Com essa escolha, foi possível reduzir significativamente o tempo de carregamento da página inicial.
- **”generateMatrix.py”**: é a codificação da etapa de representar matricialmente os dados obtidos da base de dados, através de diversas manipulações dos dados. Além da matriz ”usuárioXlivro”, esse código

também é reponsável por realizar a vetorização dos valores da matriz e aplicar a medida de similaridade da Similaridade do Cosseno. Essa operação é realizada importando a função *cosine_similarity()*, da biblioteca Scikit-Learn, em sua versão 1.2.2. Aplicando a função *cosine_similarity()* e passando a matriz ”usuárioXlivro”como parâmetro, o resultado é uma nova matriz ”livroXlivro”, onde cada valor indica o valor da similaridade do cosseno entre quaisquer dois livros da base de dados.

- **”recommend.py”**: apresenta a definição da função *recomendar()*, sendo essa o foco principal deste projeto. Na função, existem dois parâmetros:
  - *nome_livro*: parâmetro obrigatório que representa o nome do livro que será a base das recomendações
  - *qtd_recomendacoes*: parâmetro opcional, indica a quantidade de livros recomendados que deverão ser retornados. Por padrão, três livros são recomendados.

  No escopo da função, o algoritmo irá identificar o *index* do livro enviado por parâmetro e reordenar a matriz ”livroXlivro”em ordem decrescente, para obter os livros que obtiveram a maior pontuação de similaridade. Para cada um dos livros selecionados, será gerado um objeto que contém algumas informações úteis para visualização no front-end, como o Título do Livro, Autor do Livro, Ano de Publicação e a URL contendo uma imagem da capa do livro. O retorno da função é uma lista de todos esses objetos gerados.

Além dos arquivos em Python, também foram implementados arquivos em JavaScript (.js), responsáveis por promover a integração do conteúdo recebido das rotas do Flask e atualizar o conteúdo dinamicamente na página inicial e na aba de recomendações do projeto.

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. Pelas configurações estabelecidas, a recomendação por padrão exibe as três obras mais próximas calculadas por similaridade de cosseno. A figura abaixo mostra a recomendação feita para o livro ”Rising Tides”, que faz parte de uma triologia feita pela autora Nora Roberts. A eficácia do algoritmo é demonstrada por dois dos três livros recomendados, fazerem parte da triologia em que a obra inserida integra.

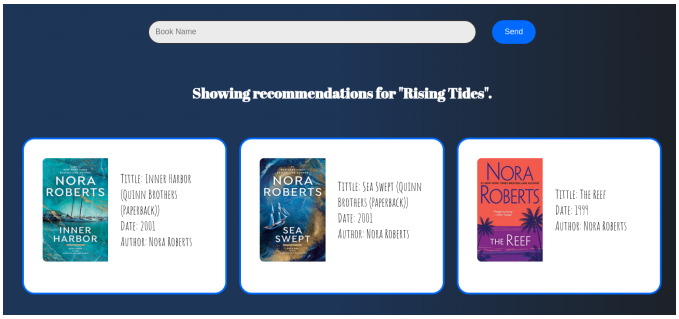

Fig. 7. Recomendação feita para o livro ”Rising Tides”

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.

V. Trabalhos Futuros

A princípio, a busca por um livro na página de recomendação deve conter o título exato da obra como ela está no dataset. Uma ferramenta que poderia auxiliar na busca e evitar erros de pesquisa seria aplicar um autocomplete no input, sugerindo livros de acordo com os caracteres inseridos pelo cliente. Mecanismos de feedback seriam importantes para manter o site atualizado e satisfatório. Pela limitação de tempo, informações adicionais da base de dados que poderiam ser integradas ao projeto foram movidas para essa seção, como a idade dos usuários e localização, criando outros parâmetros para observação e recomendação.

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 roduct 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.