Universidade Presbiteriana Mackenzie

Título: **Livros em Rede: Desvendando Recomendações Inteligentes**
CURSO: **Tecnologia em Ciência de dados**
POLO DE APOIO PRESENCIAL: **Polo EAD SP - Polo EAD Brasília**
SEMESTRE: **04**
COMPONENTE CURRICULAR / TEMA: **Projeto aplicado III**
NOME COMPLETO DOS ALUNOS: Nome : **Gustavo Silva Rios** RA : **10415824** Nome : **Silas de Souza Ferreira** RA: **10414793** Nome : **Israel Soares do N. Viana** RA: **10414894** Nome : **Danilo Brito da Silva** RA: **10415882**

Universidade Presbiteriana Mackenzie

Sumário

Universidade Presbiteriana Mackenzie

1. Lista de Figuras

2.Introdução

No contexto atual, em que as tecnologias digitais moldam grande parte das experiências cotidianas, plataformas como YouTube, Amazon e Netflix popularizaram-se por meio de sistemas de recomendação que personalizam o consumo de conteúdo.

A partir dessa observação, surge o questionamento sobre como essa abordagem pode ser aplicada para incentivar o hábito de leitura. O presente projeto propõe investigar como os sistemas de recomendação de livros podem contribuir para o aumento do interesse pela leitura, especialmente em um cenário onde a oferta de entretenimento digital tem predominado.

A leitura é uma atividade que oferece benefícios profundos e duradouros, tanto para o desenvolvimento pessoal quanto para a formação de uma sociedade mais crítica e informada. No entanto, com o tempo cada vez mais escasso e a vasta oferta de livros, a escolha de uma obra que realmente atraia o leitor pode se tornar um desafio.

A leitura tem o potencial de transformar vidas, oferecendo novas perspectivas, conhecimentos e entretenimento. No entanto, com a crescente popularidade de outras formas de mídia, o hábito de leitura muitas vezes acaba ficando em segundo plano. Neste contexto, sistemas de recomendação de livros podem desempenhar um papel fundamental ao ajudar leitores a encontrar obras que correspondam aos seus gostos e interesses, incentivando-os a ler mais e manter o hábito ao longo da vida.

Nesse sentido, o desenvolvimento de um sistema de recomendação eficaz, que utilize dados demográficos e comportamentais para sugerir leituras personalizadas, pode ser uma solução relevante para promover a leitura de forma mais acessível e satisfatória.

Este projeto surge da constatação de que muitos leitores, após se encantarem com um livro, não sabem quais obras buscar em seguida, o que pode desestimular a continuidade do hábito de leitura. A proposta deste estudo é, portanto, desenvolver um sistema de recomendação que conecte leitores a livros semelhantes aos que já leram, utilizando algoritmos de inteligência artificial. O objetivo é criar uma solução que ajude os leitores a descobrir novos títulos de forma personalizada, promovendo a leitura contínua e enriquecendo a experiência literária.

2.1 Motivação:

- A motivação central deste trabalho é estimular o hábito de leitura, utilizando tecnologias de ciência de dados para oferecer recomendações personalizadas de livros. Em um ambiente onde a concorrência com outras formas de entretenimento é intensa, acredita-se que uma abordagem direcionada e personalizada pode facilitar o acesso dos leitores a obras que atendam suas preferências, resultando em uma experiência literária mais gratificante.

2.2 Justificativa:

- O incentivo à leitura possui um papel fundamental na formação de indivíduos mais críticos e informados, sendo uma ferramenta poderosa de transformação social. A dificuldade em encontrar livros que correspondam aos interesses pessoais pode desmotivar muitos leitores, especialmente aqueles que não possuem um hábito consolidado de leitura. A implementação de um sistema de recomendação de livros, que leve em consideração tanto preferências individuais quanto dados demográficos, é uma proposta relevante para fortalecer o hábito de leitura e aumentar o acesso a conteúdos literários de qualidade, promovendo a inclusão e a democratização da cultura.

2.3 Objetivo Geral:

- Desenvolver um sistema de recomendação de livros que combine métodos de filtragem colaborativa e baseados em conteúdo, com foco em personalização e relevância, visando a promoção da leitura e a valorização do hábito literário.

2.4 Objetivos Específicos:

- Avaliar o impacto de fatores demográficos, como idade e localização, nas preferências de leitura.
- Analisar a qualidade dos dados disponíveis e seus efeitos na precisão das recomendações.
- Implementar e testar diferentes algoritmos de recomendação, comparando seu desempenho com base em métricas de avaliação.
- Desenvolver um sistema híbrido de recomendação, que integre filtragem colaborativa e métodos baseados em conteúdo, garantindo sugestões mais relevantes e personalizadas.
- Explorar o impacto da personalização nas recomendações e sua influência na satisfação dos leitores e no estímulo à leitura.

"Um livro não é apenas um amigo, ele faz amizades para você. Quando você possui um livro com mente e alma, você é enriquecido. Mas quando você o passa adiante, você é triplamente enriquecido."

— Henry Miller Os Livros da Minha Vida (1969)

3.Referencial Teórico

O desenvolvimento de sistemas de recomendação é amplamente fundamentado na literatura de recuperação de informações e aprendizado de máquina. Esses sistemas são ferramentas importantes para filtrar e sugerir itens relevantes com base em características específicas dos usuários ou dos itens.

Um dos métodos mais utilizados é a Filtragem Baseada em Conteúdo (*Content-Based Filtering*), que se concentra na similaridade entre os itens recomendados e os que já foram consumidos pelo usuário. Essa abordagem tem suas raízes em teorias de recuperação da informação, onde a semelhança entre documentos é avaliada de acordo com suas características textuais ou metadados.

Estudos de *Pazzani* e *Billsus* (2007) são referências importantes nesse campo, pois descrevem as melhores práticas e os desafios encontrados na construção de sistemas de recomendação baseados em conteúdo. A avaliação da similaridade entre itens é feita utilizando descritores textuais, como título, descrição, autores e categorias. Assim, o sistema recomenda novos livros com atributos semelhantes aos já avaliados positivamente pelo usuário (*PAZZANI*; *BILLSUS*, 2007).

A transformação de texto em vetores numéricos, necessária para a avaliação de similaridade, é realizada utilizando a técnica de *TF-IDF* (*Term Frequency-Inverse Document Frequency*). O método *TF-IDF*, descrito por Salton e Buckley (1988), permite medir a relevância de cada termo em um documento, considerando tanto a frequência do termo dentro do documento quanto sua raridade no conjunto total de dados. Isso garante que termos mais informativos tenham maior peso no cálculo de similaridade (SALTON; BUCKLEY, 1988; RAMOS, 2003).

Para calcular a similaridade entre os itens, foi utilizada a Similaridade de Cosseno (*Cosine Similarity*). Essa métrica avalia o ângulo entre os vetores *TF-IDF* de dois itens, onde um valor de similaridade de cosseno mais próximo de 1 indica uma maior semelhança entre os itens (SINGHAL, 2001). A Similaridade de Cosseno é uma técnica eficiente e amplamente adotada para mensurar similaridades em sistemas de recomendação baseados em conteúdo.

O embasamento teórico das técnicas utilizadas, como a filtragem baseada em conteúdo, TF-IDF e Similaridade de Cosseno, não só valida a escolha das metodologias aplicadas, mas também orienta a implementação e avaliação do sistema de recomendação. Além disso, práticas como o pré-processamento de dados textuais, incluindo a limpeza e a normalização, garantem a qualidade dos dados, que é essencial para o desempenho do modelo.

4.Metodologia

Nesta seção é apresentada a metodologia que foi utilizada para a construção do modelo e para a execução dos experimentos.
Abaixo na figura 1, está representado o fluxo de todo o processo de forma sequencial, desde a obtenção da base de dados até a análise dos resultados que foram obtidos, juntamente com uma explicação detalhada em cada passo.

Figura 1 – Metodologia utilizada no projeto

4.1 Obtenção dos Dados do Kaggle - 7k Books

Os dados foram obtidos pelo *Kaggle* https://www.kaggle.com/datasets/dylanjcastillo/7k-books-with-metadata.

Trata-se de uma base de dados com aproximadamente 7 mil livros, a fonte está no kaggle e foi construída pela *API* do *Goodreads*, uma plataforma amplamente reconhecida e utilizada por leitores e escritores para catalogar, avaliar e discutir livros. A API permite acessar informações detalhadas sobre livros e autores, facilitando a coleta de dados de forma organizada e limpa.

A escolha dessa fonte foi motivada pela necessidade de obter um conjunto de dados que não apenas fornecesse uma ampla variedade de títulos, mas também garantisse a integridade e a qualidade dos dados, uma vez que muitos conjuntos disponíveis em plataformas como o Kaggle apresentavam informações incompletas ou desatualizadas.

Segue abaixo o dicionário da base de dados:

Data Dictionary

Variable	Data Type	Description
isbn13	Object	Identificador ISBN 13
isbn10	Object	Identificador ISBN 10
title	Object	Título do livro
subtitle	Object	Subtítulo do livro
authors	Object	Autores do livro, separados por ";"
categories	Object	Categorias do livro, separadas por ";"
thumbnail	Object	URL da miniatura do livro
description	Object	Descrição do livro
published_year	Int32	Ano de publicação do livro
average_rating	Float64	Avaliação média no Goodreads
num_pages	Int32	Número de páginas do livro
ratings_count	Int32	Contagem de avaliações recebidas pelo livro
rating_category	Object	Categoria de classificação (ex: Alta, Média, Baixa)
decade	Int32	Década em que o livro foi publicado (ex: 2000)
book_size	Object	Tamanho do livro (ex: Pequeno, Médio, Grande)
published_year_group	Object	Grupo de ano de publicação (ex: 2000-2010)
ratings_per_page	Float64	Média de avaliações por página

Figura 2 - Dicionário de Dados

4.2 Verificação de Dados Nulos, Duplicados e Outliers

Iniciamos o processo de análise realizando uma verificação de integridade dos dados e uma limpeza geral com linguagem Python. Isso incluiu a remoção de valores nulos, o tratamento de dados ausentes e a exclusão de entradas duplicadas. Todas as inconsistências identificadas durante essas etapas foram devidamente tratadas, utilizando a biblioteca Seaborn.

Na identificação e tratamento dos outliers, verificamos valores atípicos que poderiam distorcer a análise utilizando métodos estatísticos (como IQR ou z-score), analisamos o impacto desses outliers na distribuição das variáveis e na modelagem, e com base nisso, decidimos por fazer a remoção ou ajuste desses outliers com base no contexto.

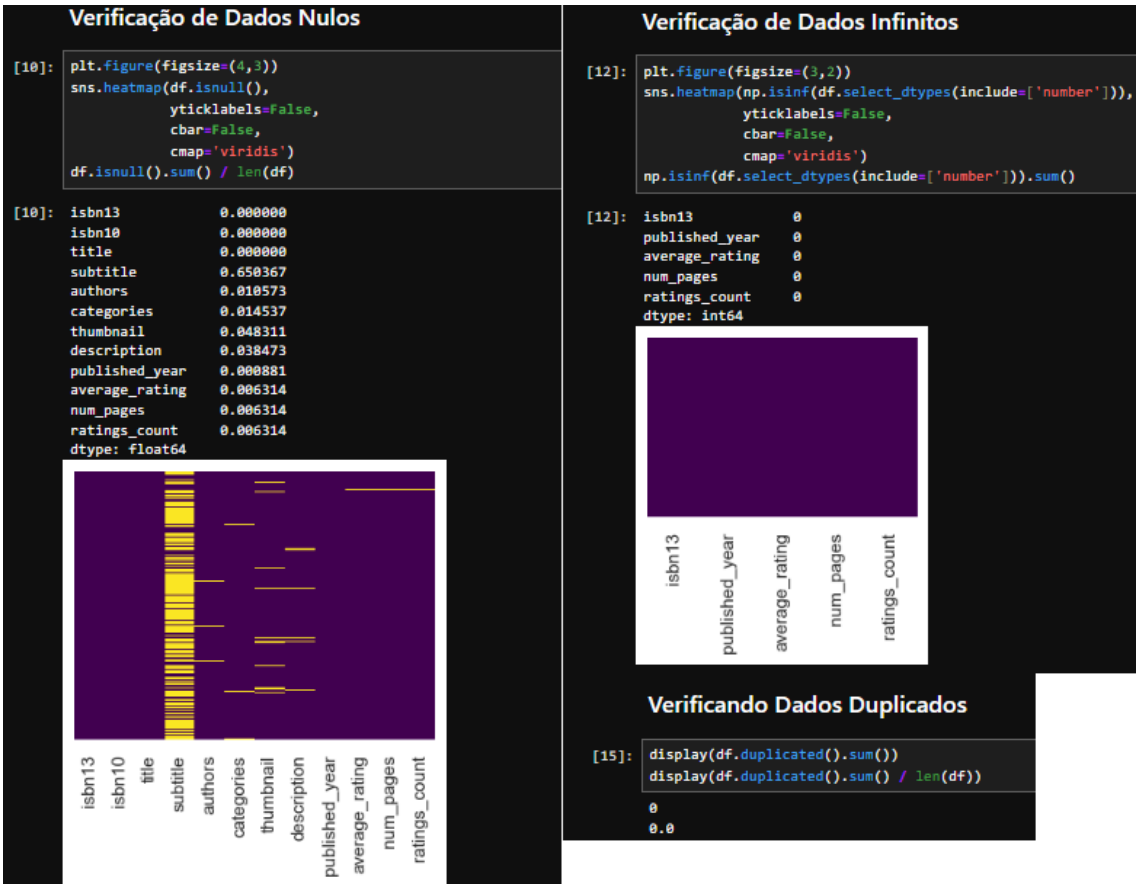


Figura 03: Verificações Iniciais de Integridade dos Dados

[20]:
```
# IQR

# Filtrar o DataFrame original para incluir apenas colunas numéricas
df_numeric = df.select_dtypes(include=['number'])

# Calcular o IQR (Intervalo Interquartil) para cada variável numérica
Q1 = df_numeric.quantile(0.25)
Q3 = df_numeric.quantile(0.75)
IQR = Q3 - Q1

# Identificar outliers em cada coluna numérica usando o método IQR
outliers_indices = []
outliers_columns = []

for col in df_numeric.columns:
    lower_bound = Q1[col] - 1.5 * IQR[col]
    upper_bound = Q3[col] + 1.5 * IQR[col]
    outliers = df[(df[col] < lower_bound) | (df[col] > upper_bound)]
    if not outliers.empty:
        outliers_indices.extend(outliers.index)
        outliers_columns.extend([col] * len(outliers))

# Criar um DataFrame com os índices das instâncias com outliers e as colunas correspondentes
outliers_df = pd.DataFrame({'Index': outliers_indices, 'Outlier_Column': outliers_columns})

# Mostrar o número de outliers identificados e a tabela com os índices das instâncias e colunas com outliers
print("Número de outliers identificados pelo método IQR:", len(outliers_df))
print("\nÍndices das instâncias com outliers e colunas correspondentes:")
print(outliers_df.sample(4))
```

```
Número de outliers identificados pelo método IQR: 2856

Índices das instâncias com outliers e colunas correspondentes:
      Index  Outlier_Column
2044   1040   ratings_count
853     107  published_year
356    6324          isbn13
61     6029          isbn13
```

[22]:
```
# Configuração dos subplots
num_plots = len(numeric_columns)
plots_per_row = 8
num_rows = (num_plots + plots_per_row - 1) // plots_per_row

fig, axes = plt.subplots(nrows=num_rows, ncols=plots_per_row, figsize=(20, 5 * num_rows))
fig.suptitle('Boxplots para Variáveis Numéricas', fontsize=16)

# Se houver apenas uma linha ou coluna, 'axes' será unidimensional
axes = axes.flatten() if num_rows > 1 else axes

# Criar boxplots para cada variável numérica
for i, col in enumerate(numeric_columns):
    sns.boxplot(x=df[col], ax=axes[i])
    axes[i].set_title(col, fontsize=12)

# Remover eixos não utilizados
for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```

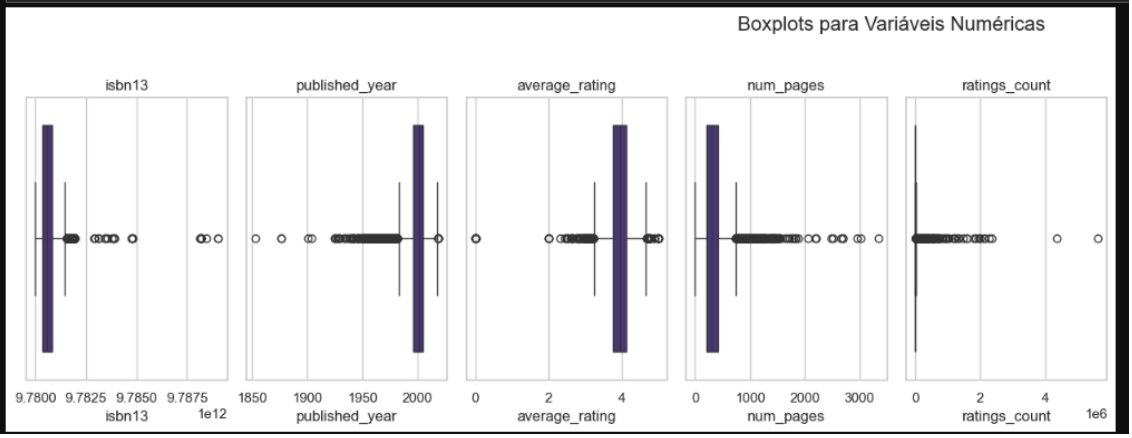


Figura 04 e 05 - Verificações de Outliers por Meio do Método IQR e Gráfico Boxplot

4.3 EDA - Pairplots, Correlações

Após a limpeza e verificação de outliers, foi realizada a análise exploratória com a finalidade de entender as estatísticas descritivas dos dados e sua distribuição nas variáveis. Algumas estatísticas como a distribuição das médias de avaliações dos livros, distribuição dos livros de acordo com a década em que foram publicados, e a contagem de livros de acordo com a categoria de avaliação que foi dividida em 4 níveis, sendo "High", "Excellent", "Low", "Medium".

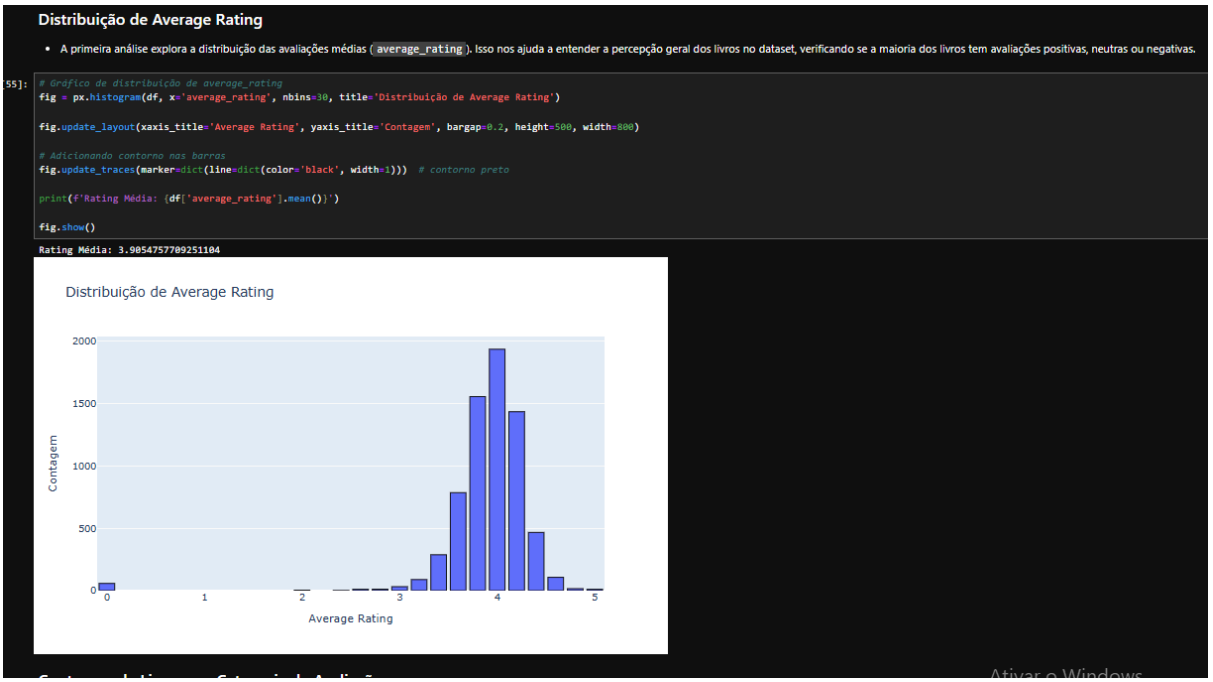
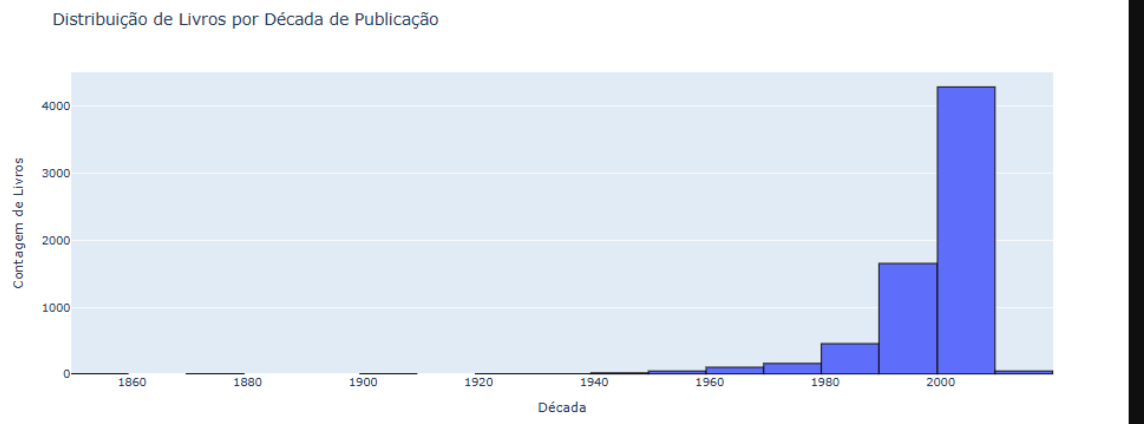


Figura 06 - Distribuição das Avaliações Médias

[60]:
```
# Gráfico de distribuição de livros por década
fig = px.histogram(df[df['decade'] != 0], x='decade', title='Distribuição de Livros por Década de Publicação')
fig.update_layout(xaxis_title='Década', yaxis_title='Contagem de Livros', height=500, width=1200)

# Adicionando contorno nas barras
fig.update_traces(marker=dict(line=dict(color='black', width=1)))  # contorno preto

fig.show()
```



Distribuição por Tamanho de Livro

- Agrupamos os livros em três categorias de tamanho (curto, médio, longo) para entender a distribuição de livros por número de páginas.

[64]:
```
# Gráfico de contagem de Livros por tamanho
book_size_counts = df['book_size'].value_counts().reset_index()
book_size_counts.columns = ['book_size', 'count']  # Renomeando as colunas

fig = px.bar(book_size_counts,
             x='book_size', y='count',
             labels={'book_size':'Tamanho do Livro', 'count':'Contagem'},
             title='Contagem de Livros por Tamanho')

fig.update_traces(marker=dict(line=dict(color='black', width=1)))  # contorno preto
fig.update_layout(height=500, width=800)

fig.show()
```

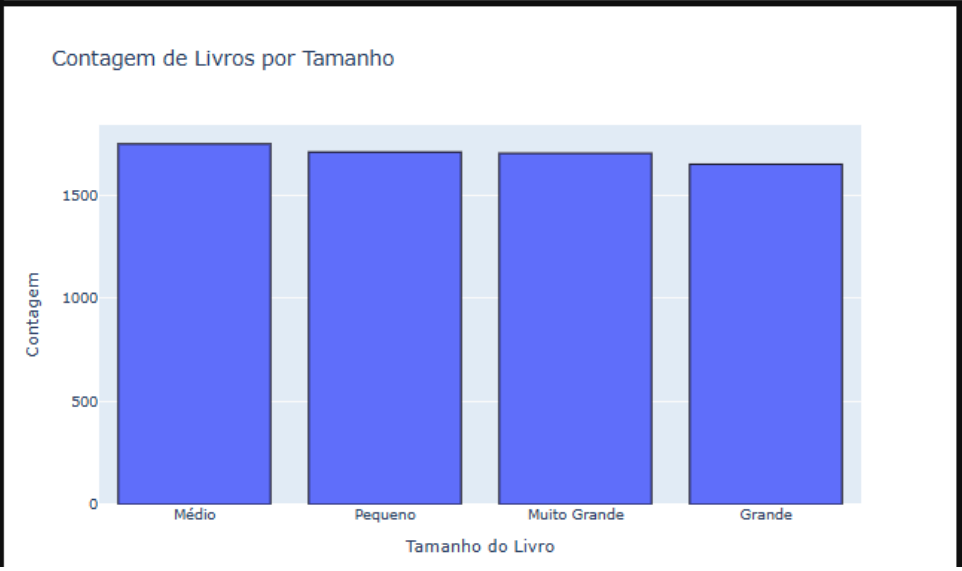


Figura 07 e 08 - Distribuição de Livros por Década de Publicação e por Critérios de Número de Páginas

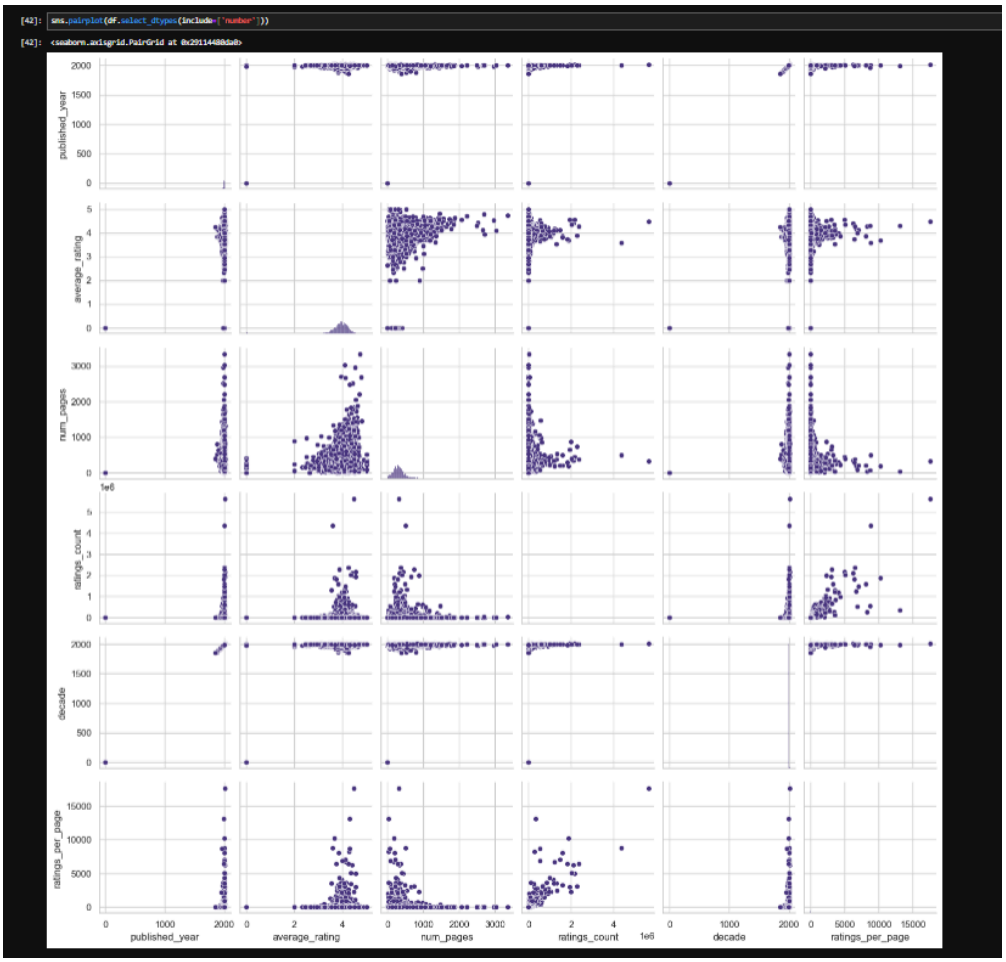


Figura 09 e 10 - Distribuição dos livros por Categorias de Avaliações e Gráfico Pairplot com todas as variáveis numéricas

Com a finalidade de entender possíveis correlações entre as variáveis, utilizamos um heatmap, trazendo o resultado de que pode haver uma forte correlação (cerca de 0.70) entre o ano de publicação e a média de rating. Como correlação fraca (cerca de 0.20) podemos citar como a média de páginas pode estar ou não relacionada com melhores médias de avaliação por parte dos usuários. Essas correlações podem levar a alguns questionamentos como por exemplo:

Livros que foram publicados a mais tempo são considerados melhores do que os livros atuais? Ou o contrário pode ser considerado ?

Livros com maior número de páginas são considerados melhores ? Ou o contrário pode ser considerado ?

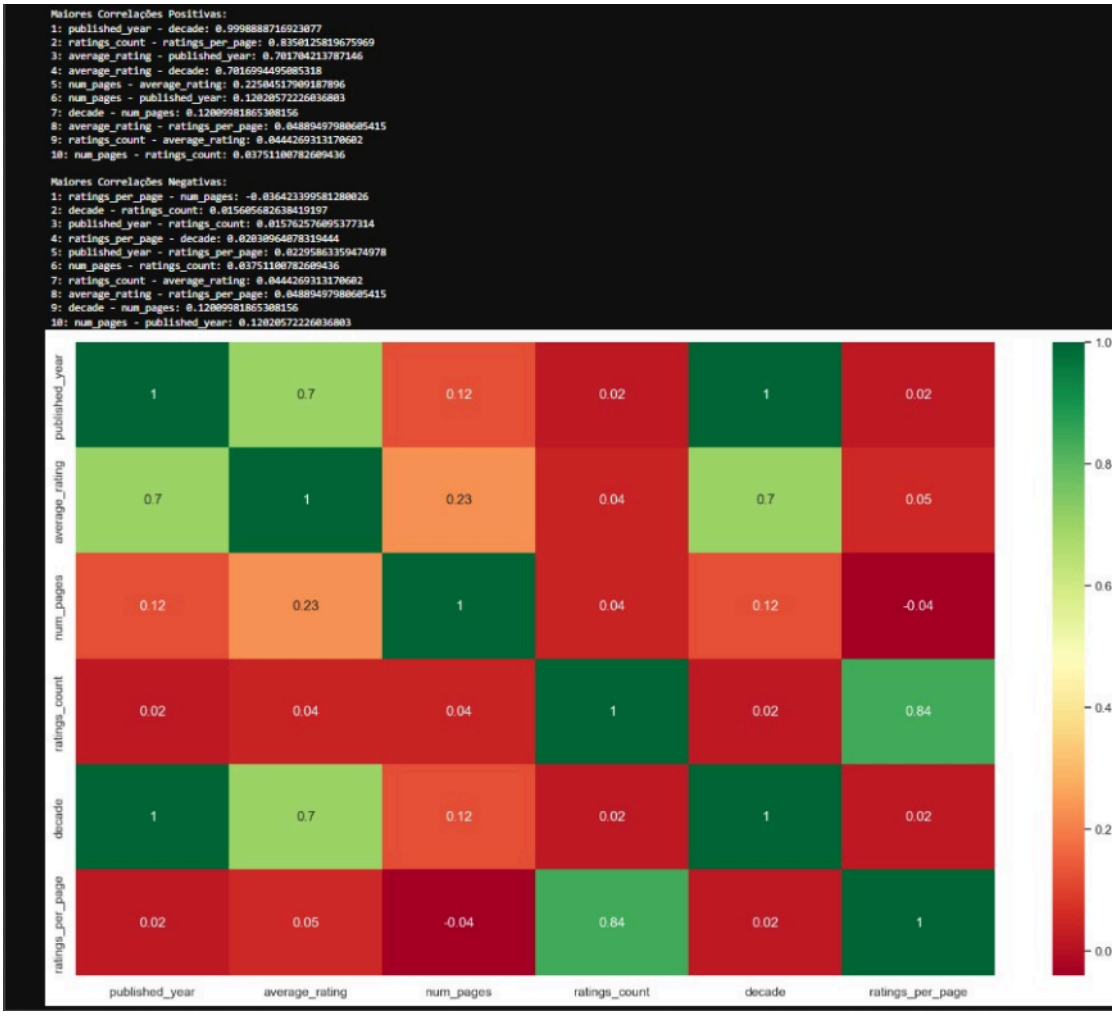


Figura 11: Gráfico Heatmap da Matriz de Correlações

4.4 Vetorização dos Dados, Seleção de Features

No pré-processamento, as variáveis de texto como *description*, *authors*, e *categories* foram preparadas e, em seguida, combinadas em uma única variável chamada *combined_features*.

Este processo unifica as principais características dos itens (neste caso, livros) para que possam ser interpretadas pelo modelo de recomendação.

A etapa de combinação permite ao modelo basear suas recomendações em uma representação mais completa e rica dos itens, incluindo informações sobre o conteúdo, autor e categoria do livro. A **Figura 12** mostra o tratamento do texto e a combinação das features em uma única variável *combined_features*. Este processo facilita a vetorização subsequente dos dados, onde cada palavra ou termo pode ser analisado em relação a seu contexto.

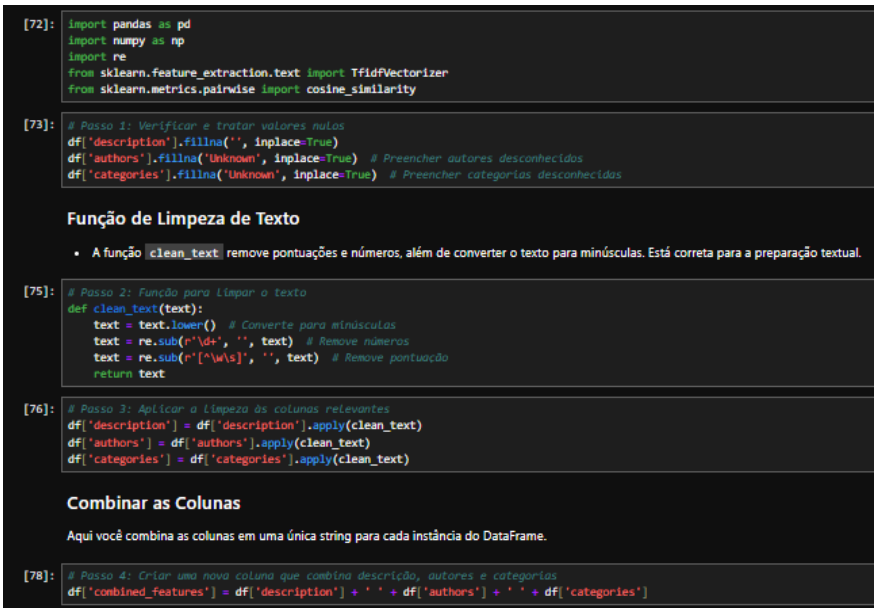


Figura 12 - Tratamento do Texto e Combinação das Features

Com a variável *combined_features* pronta, podemos então utilizar o *TfidfVectorizer* para converter o texto em valores numéricos.

O *TfidfVectorizer* transforma o texto em uma representação vetorial, atribuindo um valor a cada palavra com base em sua frequência inversa no conjunto de dados completo. Esta

vetorização enfatiza palavras específicas de cada item, atenuando palavras muito comuns que pouco contribuem para a diferenciação entre elas.

Na **Figura 13**, a vetorização de *combined_features* utilizando o *TfidfVectorizer* é ilustrada, evidenciando como as palavras são representadas numericamente. Esta etapa é crucial, pois a representação vetorial é o que permite ao modelo calcular similaridades entre os itens.

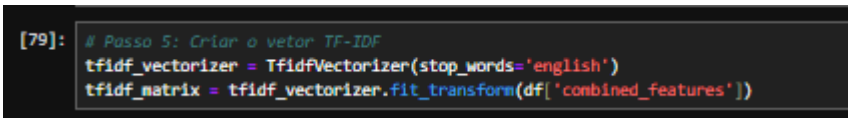


Figura 13 - Vetorização das Features Combinadas utilizando TF-IDF Vectorizer

4.5 Modelagem do Algoritmo de Recomendação

A técnica escolhida para a criação do modelo de recomendação baseou-se em um método de filtragem colaborativa baseada no conteúdo. Essa abordagem utiliza características específicas dos itens (como descrição, autor e categoria) para construir uma recomendação personalizada para cada usuário. Cada item é representado como um vetor em um espaço vetorial, onde os valores numéricos da variável *combined_features* ajudam a posicionar cada livro em relação aos outros.

Para gerar recomendações, o modelo calcula a similaridade entre os vetores dos itens. O cálculo da Similaridade entre Cossenos (em inglês *Cosine Similarity*) foi usado, uma vez que ele mede a proximidade entre vetores e permite encontrar itens com características mais parecidas. Esta métrica é particularmente útil em sistemas de recomendação, pois ela determina quão próximos os itens são entre si em relação às características vetorizadas.

A **Figura 14** abaixo ilustra o cálculo de *Cosine Similarity*, onde o sistema de recomendação personalizado recebe um título de livro e retorna uma lista de recomendações baseadas na proximidade com o vetor do livro selecionado. Esta proximidade é representada pelo valor da Similaridade entre Cossenos, e itens com valores mais altos são considerados mais similares, sendo então recomendados ao usuário.

Figura 14 - Calculando a Similaridade de Cossenos

A função *get_recommendations* (**Figura 15**) foi projetada para receber um título de livro como entrada e, a partir dele, retornar uma lista dos dez livros mais semelhantes.

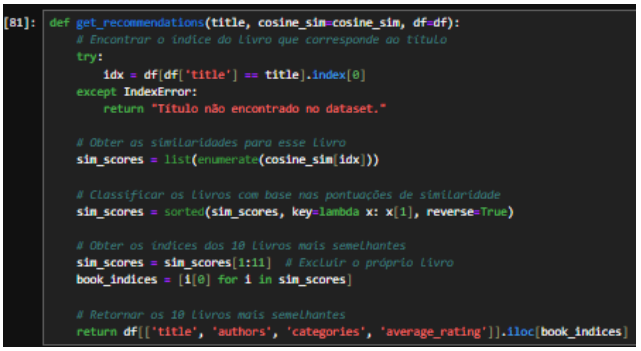
A função segue os seguintes passos:

I. Identificação do Índice do Livro: Primeiro, a função busca o índice do livro no DataFrame com base no título fornecido. Se o título não for encontrado, a função retorna uma mensagem informando que o título está ausente no dataset.

II. Cálculo das Similaridades: Com o índice identificado, a função acessa a matriz de similaridade entre cossenos (*cosine_sim*) e recupera as pontuações de similaridade entre o livro solicitado e os demais.

III. Classificação dos Livros por Similaridade: As pontuações de similaridade são então classificadas em ordem decrescente para identificar os livros mais próximos ao título consultado.

IV. Seleção dos Livros Mais Similares: A função seleciona os dez livros mais semelhantes (excluindo o próprio livro consultado) e retorna uma tabela com os títulos recomendados, incluindo suas colunas principais, como *title*, *authors*, *categories*, e *average_rating*.



```
[81]: def get_recommendations(title, cosine_sim=cosine_sim, df=df):
          # Encontrar o índice do livro que corresponde ao título
          try:
              idx = df[df['title'] == title].index[0]
          except IndexError:
              return "Título não encontrado no dataset."

          # Obter as similaridades para esse livro
          sim_scores = list(enumerate(cosine_sim[idx]))

          # Classificar os livros com base nas pontuações de similaridade
          sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)

          # Obter os índices dos 10 livros mais semelhantes
          sim_scores = sim_scores[1:11]  # Excluir o próprio livro
          book_indices = [i[0] for i in sim_scores]

          # Retornar os 10 livros mais semelhantes
          return df[['title', 'authors', 'categories', 'average_rating']].iloc[book_indices]
```

Figura 15 - Modelagem do Algoritmo de Recomendação

4.6 Testes de Recomendação de Livros pelo Modelo

Para validar o modelo, foi criada uma função chamada *test_recommendations* que testa as recomendações para uma lista de títulos de livros, selecionados aleatoriamente.

A função percorre cada título na lista de testes e exibe as recomendações para cada um deles. As etapas de teste incluíram:

- Geração de Títulos Aleatórios: Usando uma amostra do dataset, cinco títulos de livros foram escolhidos aleatoriamente. Além disso, um título inexistente ("Unknown Book Title") foi adicionado à lista para testar a capacidade da função de lidar com entradas inválidas.
- Execução dos Testes: Para cada título da lista, *test_recommendations* imprime o título do livro consultado e exibe as recomendações fornecidas pela função *get_recommendations*. Essa saída contém uma lista formatada de recomendações para cada título testado.
- Validação da Função com Título Inválido: Ao incluir um título inexistente, verificou-se que a função retorna a mensagem "Título não encontrado no dataset", o que confirma que o modelo possui um mecanismo de validação para lidar com títulos desconhecidos. Esses testes mostraram que o algoritmo é capaz de identificar e recomendar livros semelhantes com base nas features combinadas, e que lida adequadamente com títulos fora do dataset.

Figura 16 - Testando recomendações em amostras aleatórias do dataset

5. Resultados

5.1 Análise dos Resultados Obtidos

Ao solicitar recomendações para o livro "*De Profundis and Other Writings*", o modelo retornou uma lista de dez títulos (**Figura 17)** com temas, autores ou categorias relacionados ao livro consultado.

```
[85]: # Gerar uma lista de 5 títulos aleatórios do dataset usando list comprehension
test_titles = [title for title in df['title'].sample(5).tolist()]

# Adicionar um título inexistente para teste
test_titles.append("Unknown Book Title")

# Exibir a lista de títulos
print(test_titles)

# Testar as recomendações
test_recommendations(test_titles)
```

```
['De Profundis and Other Writings', 'From Nomads to Pilgrims', 'The Forever War', 'All the Names', 'Vita', 'Unknown Book Title']

Recomendações para o livro: De Profundis and Other Writings

                            title                    authors  \
6695          Shakespeare's Sonnets         william shakespeare
812                           Terre                  émile zola
5332  History of Philosophy Volume 1          frederick copleston
6066   The Invisibles, Apocalipstick  grant morrisonjill thompson
3064              Wuthering Heights     emily bronteemily brontë
76     The Real Trial of Oscar Wilde               merlin holland
1196       Complete Shorter Fiction                 oscar wilde
1368   Marketing and the Bottom Line                  tim ambler
6369          The Canterville Ghost                 oscar wilde
6295    The Unmasking of Oscar Wilde               joseph pearce

                    categories  average_rating
6695                     drama            4.25
812                    fiction            4.08
5332                philosophy            4.08
6066                   fiction            4.22
3064                   fiction            3.84
76     biography  autobiography            4.04
1196                   fiction            4.16
1368       business  economics            3.82
6369                   fiction            3.89
6295                  religion            4.25
--------------------------------------------------------------------------------
```

Figura 17 - Output do modelo com a lista de livros recomendados

A análise detalhada do retorno é apresentada abaixo, destacando as semelhanças entre os livros recomendados e o título de entrada.

Título	Autor	Categorias	Média de Avaliação
Shakespeare's Sonnets	William Shakespeare	Drama	4.25
Terre	Émile Zola	Fiction	4.08
History of Philosophy Volume 1	Frederick Copleston	Philosophy	4.08
The Invisibles, Apocalipstick	Grant Morrison, Jill Thompson	Fiction	4.22
Wuthering Heights	Emily Brontë	Fiction	3.84
The Real Trial of Oscar Wilde	Merlin Holland	Biography / Autobiography	4.04
Complete Shorter Fiction	Oscar Wilde	Fiction	4.16
Marketing and the Bottom Line	Tim Ambler	Business / Economics	3.82
The Canterville Ghost	Oscar Wilde	Fiction	3.89
The Unmasking of Oscar Wilde	Joseph Pearce	Religion	4.25

Figura 18 - Lista de Livros Similares

Com a lista de livros similares podemos então avaliar a qualidade das recomendações:

5.2 Similaridade Temática e de Autor:

- A lista de recomendações inclui livros com forte relevância temática e de autor. Por exemplo, *Shakespeare's Sonnets* e *The Real Trial of Oscar Wilde* têm conexões literárias e históricas que ressoam com *De Profundis and Other Writings*, especialmente por ambos explorarem temas profundos e literários e, em alguns casos, serem obras de autores clássicos. Além disso, outros títulos de *Oscar Wilde*, como *Complete Shorter Fiction* e *The Canterville Ghost*, aparecem nas recomendações, indicando que o modelo reconheceu corretamente a similaridade entre obras do mesmo autor.

5.3 Diversidade de Gênero e Categoria:

- Os livros recomendados abrangem uma variedade de gêneros, incluindo *Fiction*, *Drama*, *Philosophy*, e até *Business*/*Economics*.
- Essa diversidade reflete uma amplitude de categorias que, embora ligadas pela profundidade dos temas ou pelo autor, permite recomendações que vão além do gênero único e oferecem ao usuário uma gama de leituras relacionadas, mas não redundantes. Por exemplo, *History of Philosophy Volume 1* e *Marketing* and *the Bottom Line* não são ficções, mas tocam em temas analíticos e reflexivos que podem atrair leitores do livro original.

5.4 Relevância das Recomendações:

- As sugestões mantêm um equilíbrio entre proximidade com o título original e variedade de gêneros e temas. Essa flexibilidade permite que o sistema ofereça recomendações que ampliam as opções do usuário ao mesmo tempo que mantêm a pertinência. Livros como *Wuthering Heights* e *The Invisibles*, *Apocalipstick* oferecem uma experiência literária diferente, mas que provavelmente é apreciada por leitores de Wilde devido aos temas profundos e às narrativas envolventes.

Em última análise o modelo demonstrou um bom desempenho ao retornar recomendações que fazem sentido tanto em termos de similaridade temática quanto de diversidade de gêneros. A presença de obras de *Oscar Wilde* e de outros autores clássicos indica que a função de similaridade capturou adequadamente as características do título consultado. O modelo de recomendação se mostra útil para leitores que buscam tanto expandir quanto aprofundar suas leituras, baseando-se nas qualidades do livro original.

6. *Métricas de Avaliação*

As métricas de avaliação utilizadas foram

1. Cobertura (Coverage)

- Mede a proporção de itens no dataset que foram recomendados pelo sistema.
- **Importância**: Avalia se o sistema está recomendando uma variedade ampla de itens ou se está restrito a uma pequena fração do catálogo.
- **Fórmula**:

$$\text{Cobertura} = \frac{\text{Número de itens únicos recomendados}}{\text{Número total de itens no catálogo}}$$

Figura 19 - Métrica de coverage

---

2. Precisão (Precision)

- Mede a proporção de itens recomendados que são relevantes.
- **Importância**: Indica a relevância imediata das recomendações.
- **Fórmula**

$$\text{Precisão} = \frac{\text{Itens relevantes recomendados}}{\text{Total de itens recomendados}}$$

Figura 20 - Métrica de Precisão

---

3. Recall

- Mede a proporção de itens relevantes que foram recomendados.

- **Importância**: Avalia a capacidade do sistema de encontrar todos os itens relevantes para o usuário.
- **Fórmula**

:

$$\text{Recall} = \frac{\text{Itens relevantes recomendados}}{\text{Total de itens relevantes}}$$

Figura 21 - Métrica de Recall

---

4. F1-Score

- Combina Precisão e Recall em uma única métrica harmônica.
- **Importância**: Equilibra a necessidade de recomendações precisas e completas.
- **Fórmula**

:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}}$$

Figura 22 - Métrica de F1-score

---

5. Taxa de Exploração (Exploration Rate)

- Mede o quão frequentemente o sistema recomenda itens menos populares ou desconhecidos.
- **Importância**: Incentiva a descoberta de itens novos ou menos conhecidos, evitando a sobrecarga de itens populares.
- **Fórmula**:

$$\text{Exploração} = \frac{\text{Itens não populares recomendados}}{\text{Total de itens recomendados}}$$

Figura 23 - Métrica de taxa de exploração

---

6. Diversidade baseada em similaridade (Intra-List Diversity - ILD)

- Mede a dissimilaridade média entre os itens recomendados.
- **Importância**: Avalia se as recomendações cobrem uma variedade de tópicos ou categorias.
- **Fórmula**

: $$\text{ILD} = \frac{\sum_{i=1}^{n} \sum_{j=i+1}^{n} (1 - \text{Sim}(i,j))}{n \cdot (n-1)/2}$$

Figura 24 - Métrica de diversidade

---

7. Serendipidade

- Mede a frequência com que o sistema recomenda itens inesperados, mas agradáveis.
- **Importância**: Proporciona uma experiência mais agradável ao usuário, fugindo do óbvio.
- **Fórmula**: Não tem uma fórmula universal, mas pode ser medida comparando itens recomendados com itens esperados.

---

8. Satisfação do Usuário (User Satisfaction)

- Pode ser coletada por meio de avaliações explícitas (ex.: curtidas, classificações) ou implícitas (ex.: cliques, tempo de interação).
- **Importância**: Direciona o sistema para atender melhor às preferências dos usuários.

---

9. Ranking Percentil (Mean Reciprocal Rank - MRR)

- Mede a posição do primeiro item relevante em uma lista ordenada de recomendações.

- **Importância**: Avalia se itens relevantes aparecem no topo das recomendações.
- **Fórmula**:

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}$$

Figura 25 - Métrica de ranking percentil

---

10. Novidade (Novelty)

- Mede a frequência com que o sistema recomenda itens que o usuário não conhece.
- **Importância**: Promove recomendações que expandem o horizonte do usuário.
- **Fórmula**:
  Baseia-se em medidas de popularidade inversa ou probabilidade de interação.

7. Conclusões e Trabalhos Futuros

7.1 Resumo dos Principais Resultados:

- Neste projeto, desenvolvemos e avaliamos um sistema de recomendação de livros baseado em dados como média de avaliações, número de páginas e contagem de avaliações. Inicialmente, adotamos uma abordagem baseada em similaridade de cosseno utilizando essas características, com o intuito de gerar recomendações personalizadas para os usuários. Os resultados indicaram que a abordagem conseguiu identificar livros semelhantes com base em características quantitativas, mas a recomendação poderia ser aprimorada com o uso de modelos mais avançados.

- A introdução do modelo SVD (Singular Value Decomposition) para reduzir a dimensionalidade e melhorar a qualidade das recomendações mostrou-se eficaz. A análise comparativa entre os métodos de similaridade de cosseno e SVD revelou uma melhoria na diversidade e na personalização das recomendações, com o SVD permitindo uma representação mais rica e compacta dos dados.

7.2 Contribuições do Projeto:

- Este projeto contribui para o avanço de sistemas de recomendação ao explorar diferentes abordagens de modelagem, demonstrando como técnicas de redução de dimensionalidade como o SVD podem ser integradas a modelos tradicionais de recomendação para melhorar a qualidade das sugestões. Além disso, ao aplicar métricas como precisão, recall, F1-score, e diversidade, o projeto proporcionou uma avaliação robusta dos modelos, permitindo compreender não apenas a eficácia das recomendações, mas também aspectos importantes da experiência do usuário, como a novidade e a diversidade dos itens recomendados.

- O uso de similaridade de cosseno e SVD permite que o sistema ofereça recomendações mais personalizadas e escaláveis, atendendo a um público diversificado de usuários com preferências variadas.

7.3 Limitações Identificadas:

- Embora os modelos de recomendação tenham mostrado bons resultados, algumas limitações precisam ser abordadas:

- Escalabilidade: O sistema pode enfrentar desafios de desempenho ao lidar com um grande número de livros e usuários, especialmente quando a base de dados cresce significativamente. A aplicação de técnicas de matriz esparsa e indexação eficiente poderia melhorar a escalabilidade do sistema.

- Simplicidade das Características: O sistema atual utiliza apenas um conjunto limitado de características (média de avaliações, número de páginas, e contagem de avaliações). Características adicionais, como gênero literário, autores ou até mesmo histórico de leitura do usuário, poderiam melhorar a personalização e a precisão das recomendações.

- Recomendações Frias: O sistema pode ter dificuldades em recomendar livros que ainda não possuem muitas avaliações, ou "itens frios", pois a falta de dados dificulta a avaliação precisa da similaridade. A inclusão de técnicas como colaboração baseada em conteúdo poderia ajudar a mitigar essa limitação.

7.4 Impacto Prático:

- O impacto prático deste projeto pode ser significativo em diversos cenários:

- Aumento da Retenção de Usuários: Ao fornecer recomendações mais relevantes e personalizadas, o sistema pode aumentar o engajamento dos usuários, incentivando-os a explorar mais livros e a interagir com a plataforma com maior frequência.

- Melhoria na Experiência do Usuário: A melhoria na precisão das recomendações pode contribuir para uma experiência mais fluida e satisfatória. Usuários que recebem sugestões mais alinhadas com suas preferências tendem a ficar mais satisfeitos com o serviço, resultando em maior fidelização.

- Crescimento de Receita: Para plataformas comerciais, como lojas de livros ou serviços de assinatura, um sistema de recomendação eficiente pode impulsionar as vendas ao sugerir livros com maior probabilidade de serem comprados ou lidos. Além disso, a recomendação de livros pode ser usada para promover novas publicações, aumentando a receita.

7.5 Trabalhos Futuros

- No futuro, o sistema de recomendação pode ser aprimorado com a inclusão de novas técnicas e melhorias nas abordagens existentes:

- Exploração de Modelos Híbridos: combinação de modelos baseados em filtragem colaborativa e filtragem por conteúdo para melhorar a qualidade das recomendações, especialmente em cenários com poucos dados.

- Incorporação de Feedback Explícito e Implícito: Utilizar informações adicionais de interações do usuário, como histórico de navegação, tempo de leitura, ou feedback explícito (avaliações, likes) para aprimorar as recomendações.

- Uso de Deep Learning: Experimentação com redes neurais para modelar de maneira mais complexa as preferências dos usuários e os padrões de consumo de livros.

- Personalização Aprofundada: Exploração de técnicas de análise de sentimentos em resenhas de usuários e utilização de processamento de linguagem natural (NLP) para capturar melhor as preferências de leitura baseadas em temas, gêneros ou estilo literário
- Essas melhorias podem não só aumentar a qualidade das recomendações, mas também ampliar a capacidade de personalização do sistema, trazendo um impacto mais robusto na experiência do usuário e nos resultados práticos para a plataforma.

8. Referências

PAZZANI, M.; BILLSUS, D. *Content-based recommendation systems*. In: The Adaptive Web. Springer, 2007. p. 325-341.

RAMOS, J. *Using TF-IDF to determine word relevance in document queries*. In: *Proceedings of the first instructional conference on machine learning*. 2003.

SALTON, G.; BUCKLEY, C. *Term-weighting approaches in automatic text retrieval*. *Information Processing & Management*, v. 24, n. 5, p. 513-523, 1988.

SINGHAL, A. *Modern information retrieval: A brief overview*. *IEEE Data Engineering Bulletin*, v. 24, n. 4, p. 35-43, 2001.

Link da Apresentação em Vídeo: https://youtu.be/sGOIuVImT0c

Link do repositório: https://github.com/gustavosrios/mack-semestre-4

Link do dataset: https://www.kaggle.com/datasets/dylanjcastillo/7k-books-with-metadata.

Caso você não consiga abrir o link ou tenha problemas para ver as visualizações no arquivo *.ipynb* do *GitHub*, abra esse link abaixo:

https://nbviewer.org/github/gustavosrios/mack-semestre-4/blob/main/Projeto%20Aplicado%20III.ipynb