**Douglas Pereira de Araujo  
Felipe Dal Molin  
Gabriela Germana Da Silva  
Samuel Regis Nascimento Barbosa**

**Projeto Aplicado III**

**Algoritmo de Recomendação para indicação de Livros em Python**

**São Paulo**

**2023**

**Sumário**

Introdução....................................................................................................................................................... 3

Cronograma.................................................................................................................................................... 4

Bibliografia ………………………………………………………………………………..………………..5

Bibliotecas Utilizadas no Projeto ………………………......……………………………...………………..5

Modelagem dos dados ………………………......…………….................………………...………………..6

Análise Exploratória dos dados …………………………….................………………...……………..…..11

Construção do Modelo …..............………………………….................………………...……………..…..21

**Introdução:**

Descoberta de Conhecimento em Bancos de Dados (KDD), ou Knowledge Discovery in Databases, é um processo interdisciplinar que abrange diversas etapas, desde a seleção e preparação dos dados até a interpretação dos resultados obtidos. É um campo que combina elementos da mineração de dados, aprendizado de máquina, estatísticas e outras disciplinas relacionadas, visando extrair informações úteis e ocultas a partir de grandes conjuntos de dados. O processo de KDD envolve a identificação de padrões, tendências e relações nos dados, resultando na geração de conhecimento que pode ser aplicado em tomadas de decisão, previsões e outras análises significativas.

Os algoritmos de recomendação têm um papel fundamental na análise de dados, especialmente no contexto do comércio eletrônico, plataformas de streaming, redes sociais e muitos outros sistemas onde há uma abundância de opções disponíveis para os usuários. Esses algoritmos buscam prever as preferências ou interesses dos usuários e, com base nessa previsão, sugerir itens ou conteúdos que possam ser do interesse deles. A popularidade e a importância dos algoritmos de recomendação cresceram exponencialmente à medida que as empresas buscam personalizar a experiência do usuário e melhorar a satisfação.

Nesse contexto da análise de dados, escolhemos a aplicação de algoritmos de recomendação para livros, que tem desempenhado um papel significativo em aprimorar a experiência dos leitores e promover a descoberta literária. Esse sistema especializado visa entender os gostos e preferências de leitores individuais, utilizando dados como histórico de leitura, classificações e interações com outros

livros. Com base nessas informações esses algoritmos buscam aprender padrões e relações nos dados de treinamento para fazer previsões ou classificações precisas em novos dados, com base nas informações aprendidas durante o treinamento, onde irão identificar padrões e relações entre títulos, autores e gêneros. Esses sistemas de recomendação para livros não apenas facilitam a descoberta de novas obras, mas também contribuem para a construção de uma comunidade de leitores ao redor de interesses comuns. Ao combinar os princípios de KDD e algoritmos de recomendação, esses sistemas proporcionam uma experiência de leitura mais enriquecedora e diversificada, conectando leitores a livros que poderiam passar despercebidos em meio a uma vasta oferta literária.

Os princípios e os processos envolvidos na descoberta de padrões úteis nos dados, abrangem duas técnicas de identificação das informações. Os algoritmos supervisionados são projetados para trabalhar com dados rotulados, ou seja, conjuntos de dados em que cada exemplo é associado a um rótulo ou classe conhecida. Em contraste, os algoritmos não supervisionados lidam com dados não rotulados, explorando a estrutura subjacente dos dados para identificar padrões, agrupamentos e características intrínsecas. Eles buscam organizar os dados de maneira significativa, revelando informações sobre similaridades e diferenças entre os exemplos sem a orientação de rótulos predefinidos. Enquanto os algoritmos supervisionados são aplicados em tarefas de classificação e regressão, os algoritmos não supervisionados são frequentemente utilizados em tarefas de clusterização proporcionando uma compreensão mais profunda dos dados e suas relações.

**Cronograma:**

|  |  |  |
| --- | --- | --- |
| **Etapa** | **Descrição da atividade** | **Prazo** |
| **Etapa 1** | Montagem do grupo | 10/08 a 27/08 |
| Escolha da temática |
| Escolha do Dataset |
| Organização dos documentos no repositório |
| **Etapa 2** | Elaboração da proposta analitica | 28/08 a 21/09 |
| Tratar e preparar a base de dados |
| Definir a técnica para o treinamento do modelo |
| Validação do Modelo |
| Descrever o referencial teórico para a elaboração do projeto. |
| Apresentação dos Scripts da Análise Exploratória em Python |
| Construção grafíca dos resultados |
| **Etapa 3** | Ajustar o pipeline de treinamento para o resultado final | 23/09 a 20/10 |
| Reavaliar o desempenho do modelo. |
| Descrever a metodologia aplicada. |
| **Etapa 4** | Analisar os resultados obtidos | 24/10 a 01/11 |
| Descrever e documentar os resultados |
| Finzalização do data Storytelling |
| Descrever e documentar as conclusões e os trabalhos futuros. |
| Ajuste do relatório final |
| Realizar a gravação da apresentação do projeto em vídeo |
| Organizar todos os documentos no repositórios |
| Entregar todos os arquivos do projeto |
| Vídeo de apresentação do projeto |

**Bibliotecas utilizadas no modelo:**

Pandas;

Numpy;

Matplotlib;

Seaborn;

Scipy;

Requests;

Sklearn;

Pillow;

Warnings

**Importação das Bibliotecas**

# Importação de Bibliotecas

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import plotly.express as px

import warnings

# Configurações Iniciais

warnings.filterwarnings('ignore')

pd.set\_option('display.max\_rows', 100)

pd.set\_option('display.max\_columns', 50)

plt.rcParams['figure.figsize'] = (15, 6)

plt.style.use('seaborn-darkgrid')

**\***

# Ler os dados

Dados\_Livros = pd.read\_csv('Books.csv')

Dados\_Avaliacao = pd.read\_csv('Ratings.csv')

Dados\_Usuario = pd.read\_csv('Users.csv')

# Dimensçao [ Linhas, Colunas ]

Dados\_Livros.shape, Dados\_Avaliacao.shape, Dados\_Usuario.shape

**Modelagem dos dados**

Na fase de modelagem do nosso projeto, um passo crucial foi combinar diferentes conjuntos de dados. Essa etapa, chamada de cruzamento de dados, foi essencial para criar o modelo que recomendará livros aos usuários.

Cruzamento dos Dados

1º Cruzamento:

No primeiro cruzamento de dados, juntamos informações sobre livros e avaliações dos usuários. Esses dados foram combinados para criar uma base que relaciona livros e as opiniões dos leitores. Isso foi feito usando o número de identificação dos livros (ISBN) como chave para conectar as informações.

2º Cruzamento:

No segundo cruzamento de dados, adicionamos informações sobre os próprios usuários. Combinamos os dados já consolidados no primeiro cruzamento com os perfis dos leitores. Ao conectar essas informações usando os IDs dos usuários, conseguimos entender melhor as preferências individuais de leitura. Isso tornou nosso modelo de recomendação altamente personalizado, levando em conta as preferências únicas de cada usuário.

Esses cruzamentos criaram uma tabela única que se tornou a base do nosso modelo de recomendação de livros. Esta tabela contém informações sobre livros, avaliações de usuários e os perfis dos leitores.

**\***

# Cruzamentos dos dados

# 1º Cruzamento

Tab\_Cruzada = Dados\_Livros.merge( Dados\_Avaliacao, how='inner', on='ISBN')

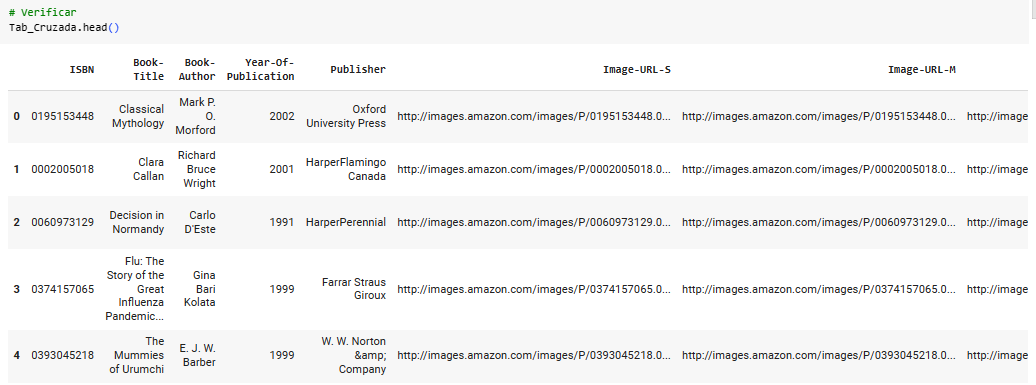
# 2º Cruzamento

Tab\_Cruzada = Tab\_Cruzada.merge( Dados\_Usuario, how='inner', on='User-ID')

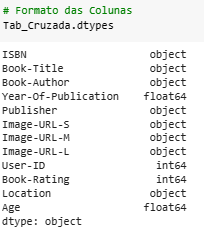
# Dimensão

Tab\_Cruzada.shape

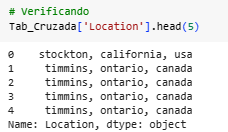
**\***

****

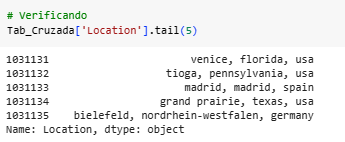
**\***

****

**\***

****

**\***

****

**\***

# Tecnica de tratamento de texto

def Extrair\_Pais( Regiao ):

  '''

    Função para extrair o nome do pais na coluna região

  '''

  # Incluindo a inforção

  Registro = Regiao

  # Fatiar

  Registro = Regiao.split(',')

  # Buscar

  Fracao = Registro[-1].upper()

  #Retorno

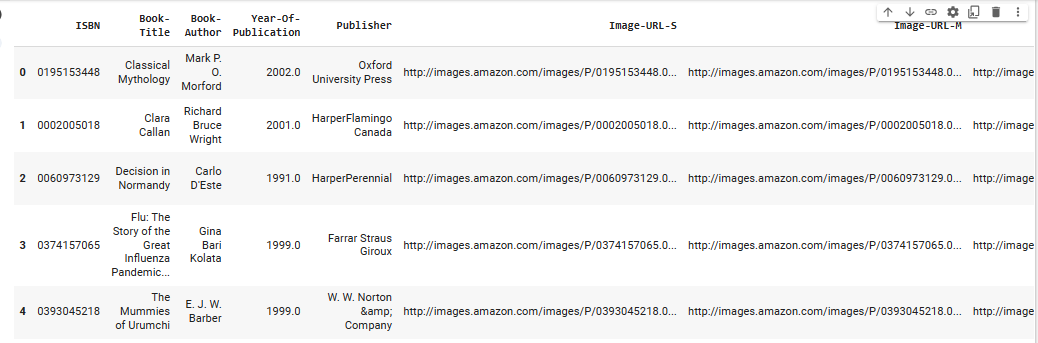
  return Fracao

# Criando a coluna

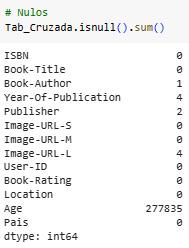
Tab\_Cruzada['Pais'] = Tab\_Cruzada['Location'].apply( Extrair\_Pais )

# Verificando

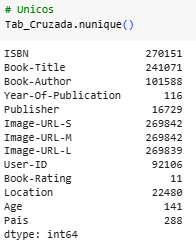
Tab\_Cruzada.head()

****

**\***

****

**\***

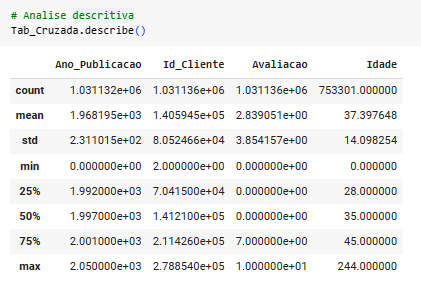
****

**\***

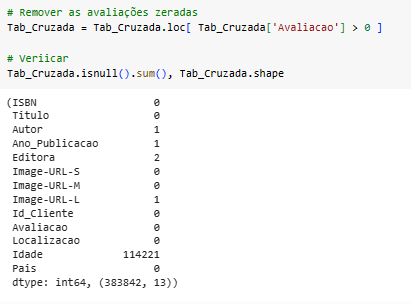
****

**Análise Exploratória dos dados:**

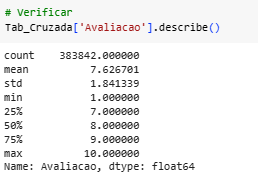
**\***



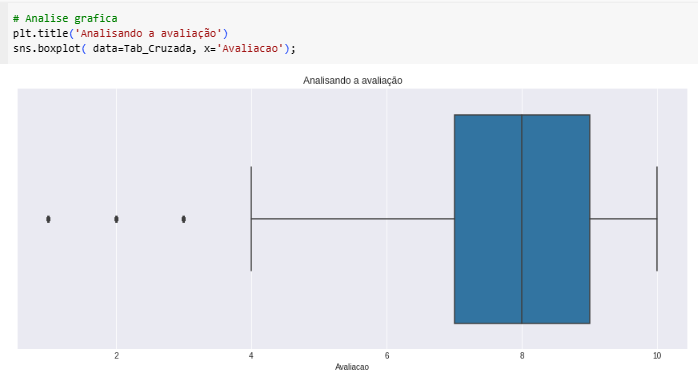
\*



**\***

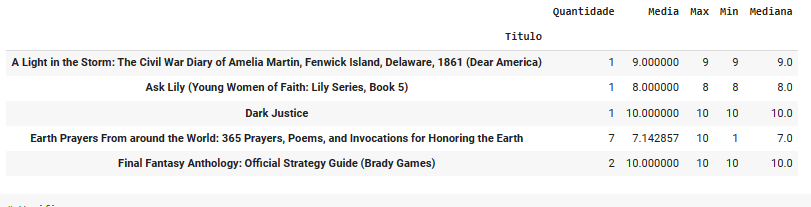
****

**\***

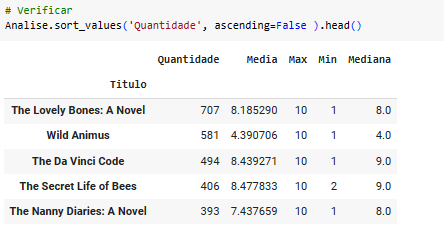
****

**\***

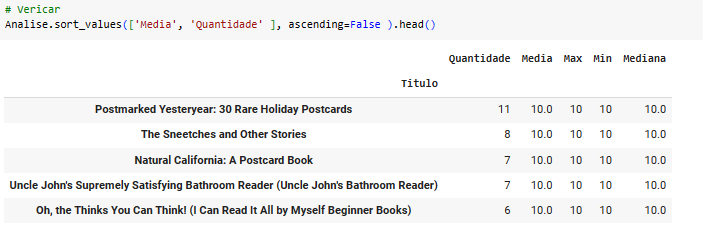
****

****

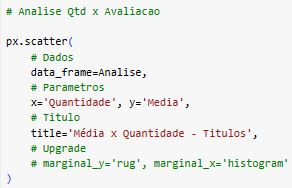
**\***

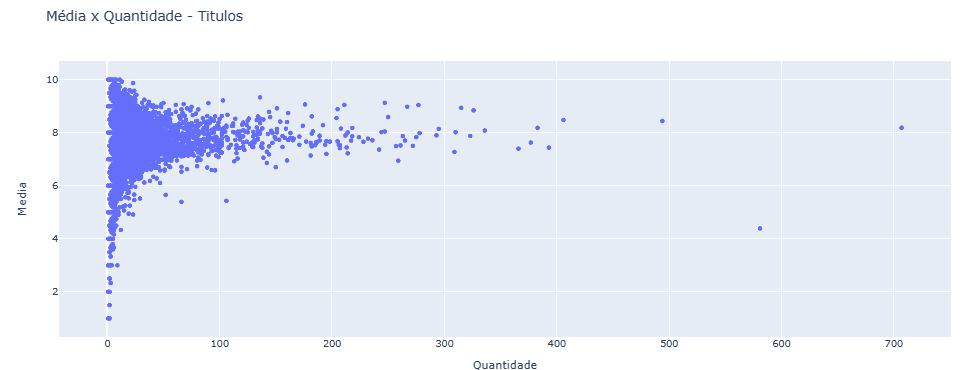
****

**\***

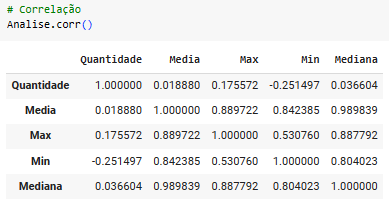
****

**\***

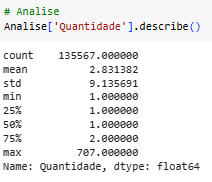
****

****

**\***

****

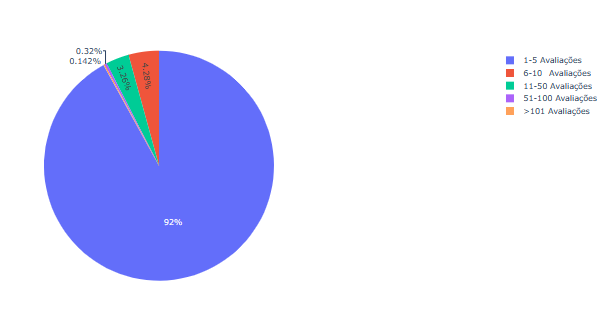
**\***

****

**\***

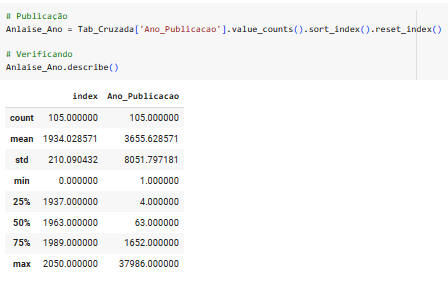
****

Distribuição das quantidades em cada categoria

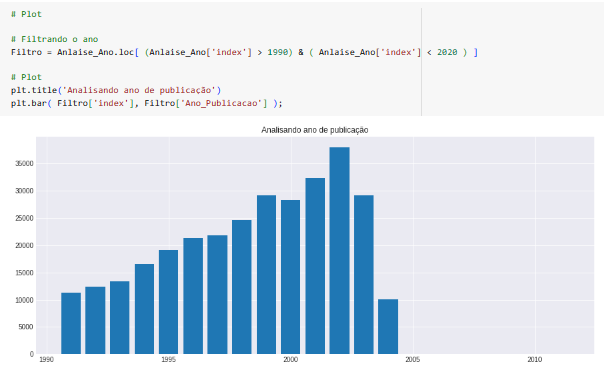
****

****

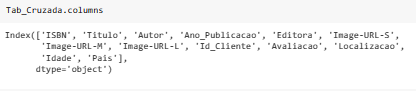
**\***

****

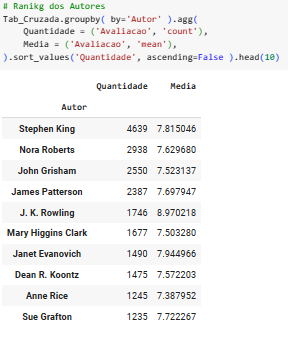
**\***

****

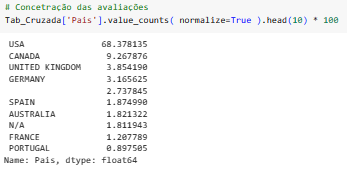
**\***

****

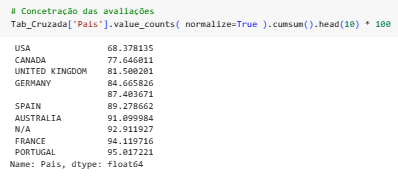
**\***

****

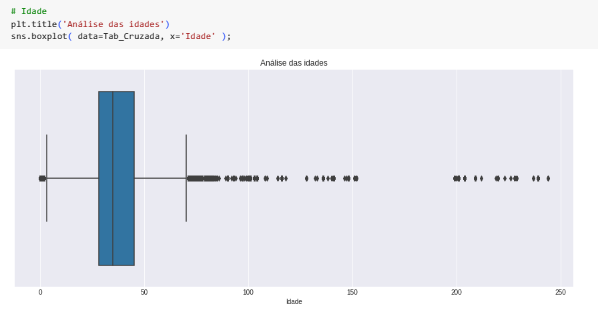
**\***

****

**\***

****

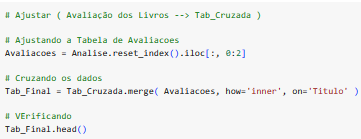
**\***

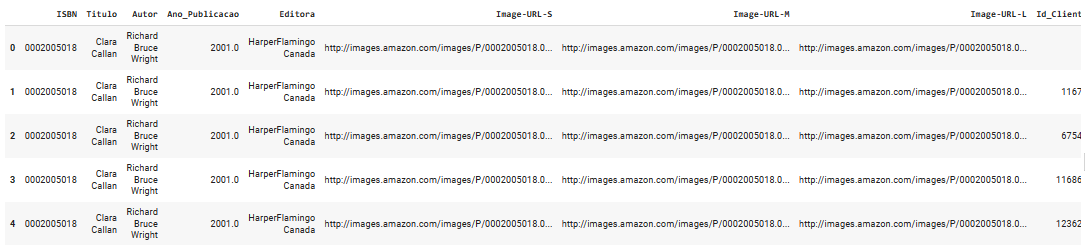
****

**Construção do modelo:**

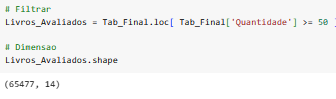
O Aprendizado por Representação é a busca por melhores representações de dados em algoritmos, muitas vezes não supervisionados. Exemplos incluem Análise de Componentes Principais e Clustering. Esses métodos transformam dados de entrada para preservar informações úteis, útil como pré-processamento para classificação e previsão, permitindo reconstrução de dados de origens desconhecidas. Algoritmos buscam representações de baixa dimensão, esparsas ou hierarquias em níveis abstratos. Essencial para máquinas inteligentes que desvendam fatores subjacentes dos dados.

**\***



****

**\***

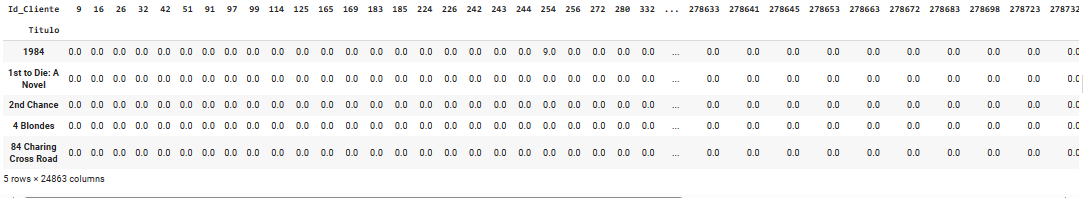
****

**\***

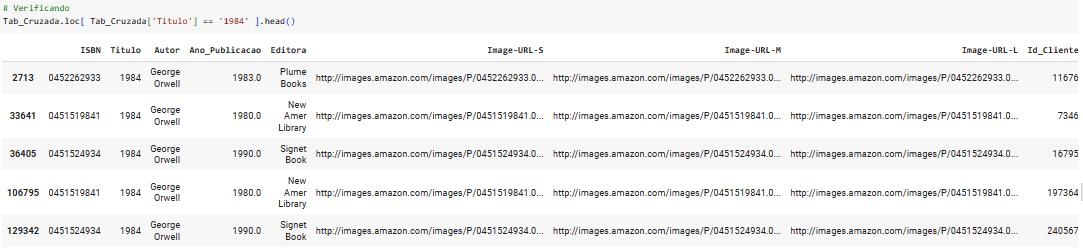
****

**\***

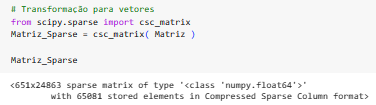
****

****

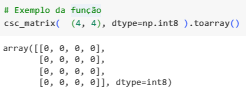
**\***

****

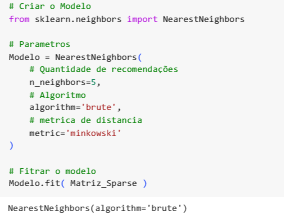
**\***

****

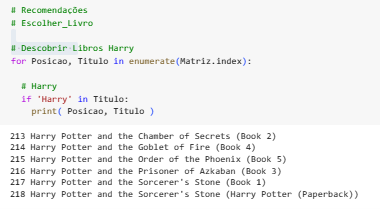
**\***

****

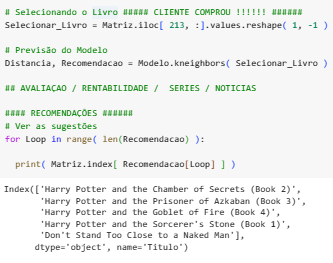
**\***

****

**\***

****

**\***

****

**\***

****

**Bibliografia**:

[Book Recommendation Dataset | Kaggle](https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset)