**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

[Cronograma 3](#_Toc150725025)

[Bibliotecas utilizadas no modelo: 4](#_Toc150725026)

[Introdução 5](#_Toc150725027)

[Metodologia do Projeto de Recomendação de Livros 7](#_Toc150725028)

[Importação das Bibliotecas 7](#_Toc150725029)

[Análise Exploratória dos dados 13](#_Toc150725030)

[Construção do modelo 23](#_Toc150725031)

[Treinamento do Modelo 28](#_Toc150725032)

[Framework – Simulação 28](#_Toc150725033)

[Conclusão 31](#_Toc150725034)

[Links 32](#_Toc150725035)

[Bibliografia 33](#_Toc150725036)

# 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 analítica | 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 gráfica dos resultados |
| Etapa 3 | Ajustar o pipeline de treinamento para o resultado | 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 |
| Finalização 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 nos 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

# 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. 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.

Para o projeto proposto foi escolhido o método k-NN, um sistema que 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 o algoritmo busca aprender padrões e relações entre títulos, autores e gêneros nos dados de treinamento para fazer previsões ou classificações precisas em novos dados, com base nas informações aprendidas durante o treinamento.

O k-NN (k-vizinhos mais próximos) é um método utilizado em aplicações de classificação que considera que os registros do conjunto de dados correspondem a pontos no espaço, onde cada atributo representa uma dimensão desse espaço. Quando um novo registro precisa ser classificado, ele é comparado com todos os registros do conjunto de treinamento para identificar os k vizinhos mais próximos. A classe do novo registro é determinada pela classe mais frequente entre esses vizinhos mais próximos, podendo variar de acordo com a métrica escolhida. O valor de k é um parâmetro de entrada do método, e as métricas mais comuns utilizadas são a Euclidiana e a de Manhattan. A escolha do valor de k depende do conjunto de dados e pode ser determinada por técnicas como validação cruzada e bootstrap e precisam ser considerados pois impactam na sensibilidade ao ruído e na definição das fronteiras entre as classes.

# Metodologia do Projeto de Recomendação de Livros

# 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')

**Coleta de Dados:**

Foram coletados três conjuntos de dados em formato CSV: Books.csv, Ratings.csv e Users.csv.

O conjunto de dados Books.csv contém informações sobre os livros, incluindo título, autor, ano de publicação, editora e URLs da imagem, o conjunto de dados Ratings.csv registra as avaliações dos usuários para os livros e o conjunto de dados Users.csv contém informações demográficas sobre os usuários, como localização e idade.

# 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

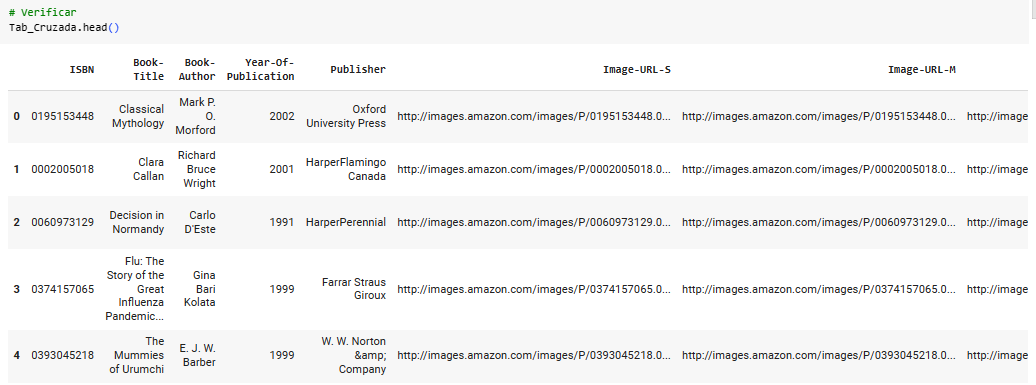
Tab\_Cruzada = Tab\_Cruzada.merge( Dados\_Usuario, how='inner', on='User-ID')

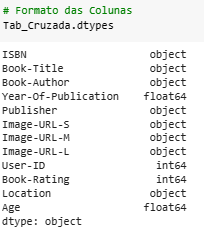
# Dimensão

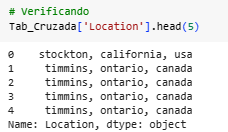
Tab\_Cruzada.shape

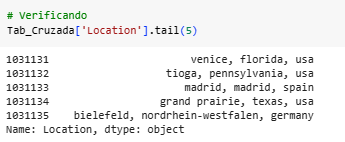
**2. Exploração de Dados**

Uma visualização inicial dos conjuntos de dados foi realizada para entender a estrutura e o conteúdo dos dados. Foi identificado que a abordagem de filtragem colaborativa poderia ser usada para criar o sistema de recomendação, dadas as avaliações dos usuários disponíveis.

****

****

****

****

# 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

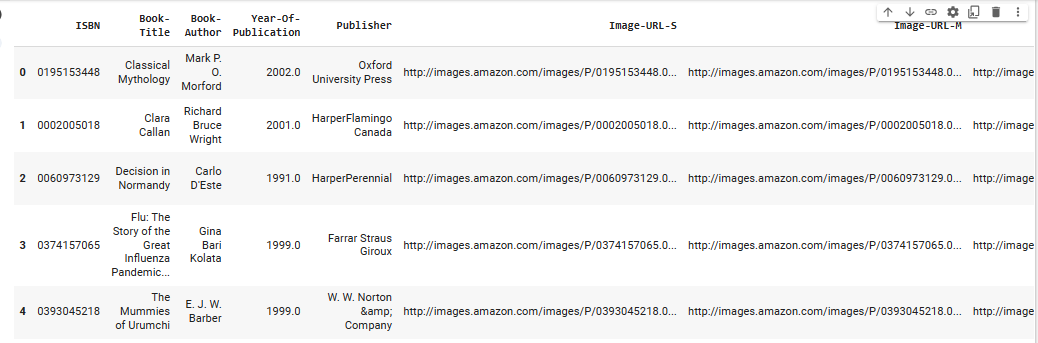
  return Fracao

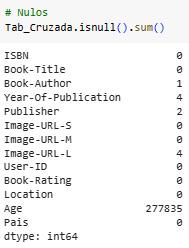
# Criando a coluna

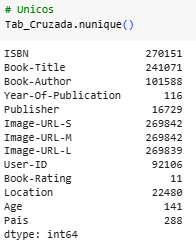
Tab\_Cruzada['Pais'] = Tab\_Cruzada['Location'].apply( Extrair\_Pais )

# Verificando

Tab\_Cruzada.head()

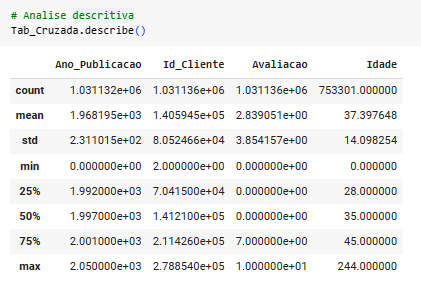
****

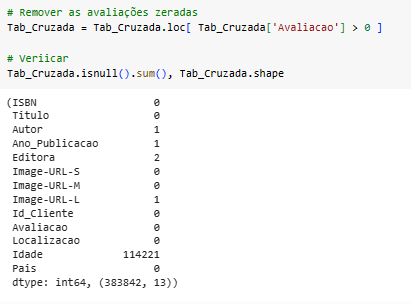
****

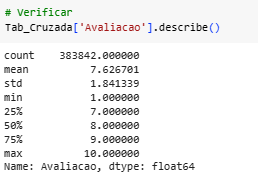
****

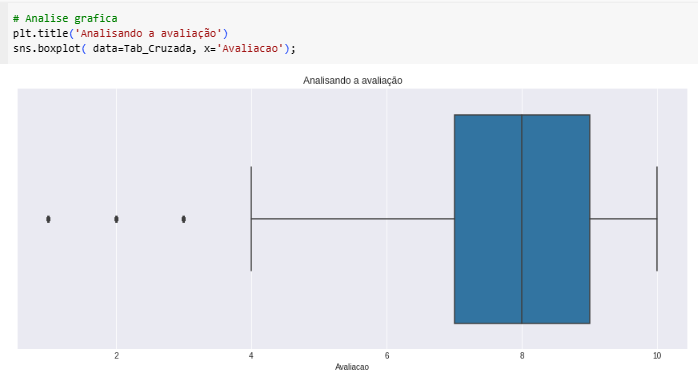
****

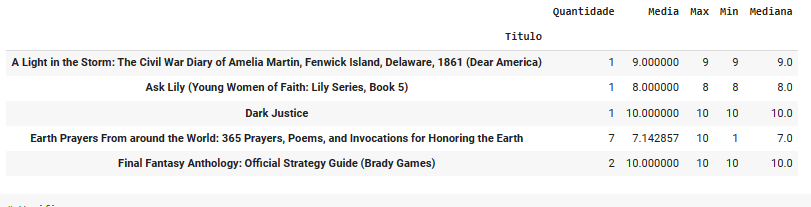
# Análise Exploratória dos dados

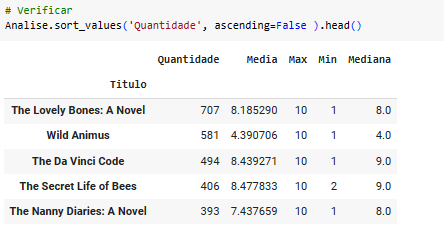


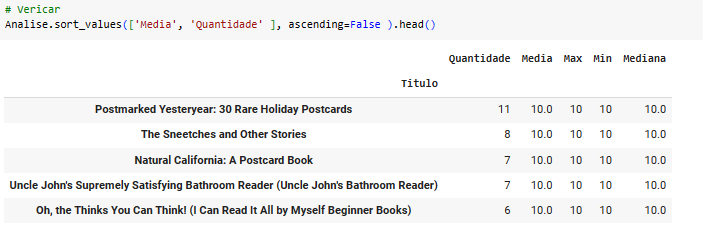


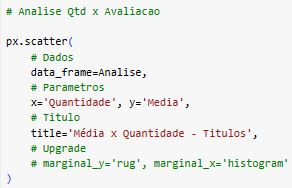
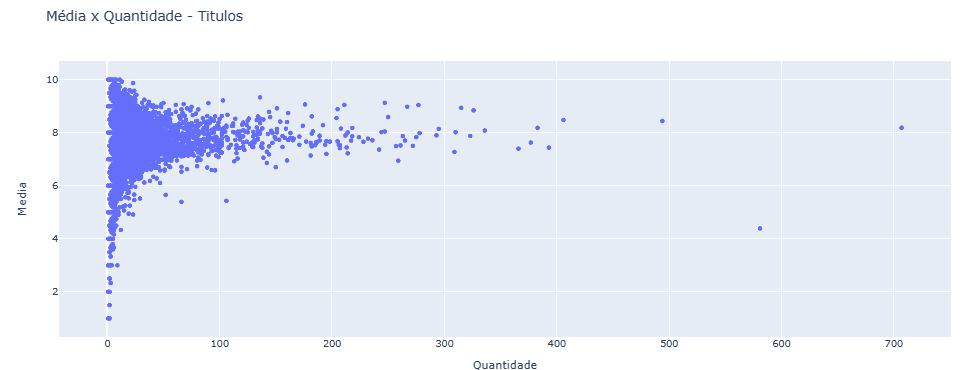
****

****

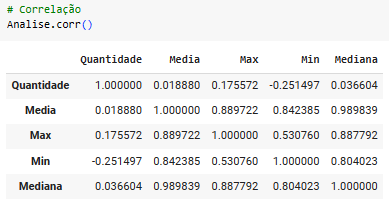
****

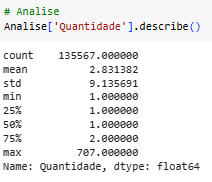
****

****

****

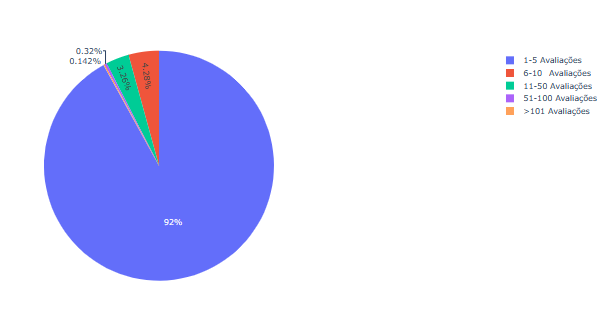
**’**

****

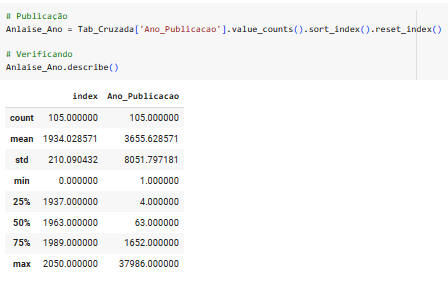
****

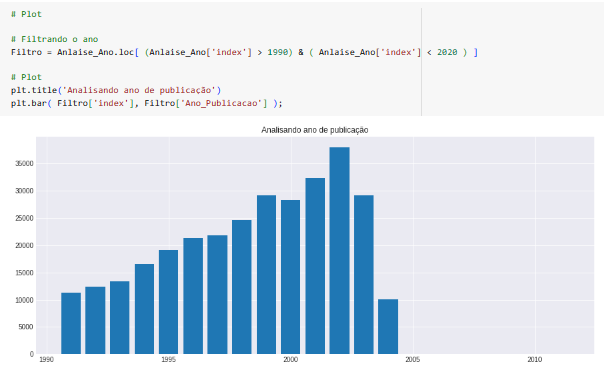
****

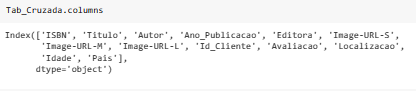
Distribuição das quantidades em cada categoria

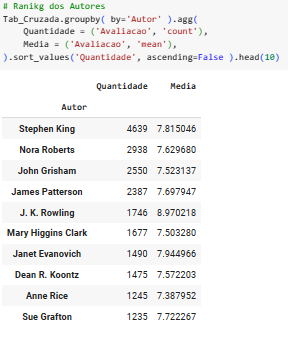
****

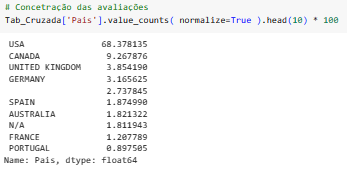
****

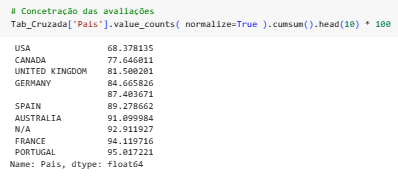
****

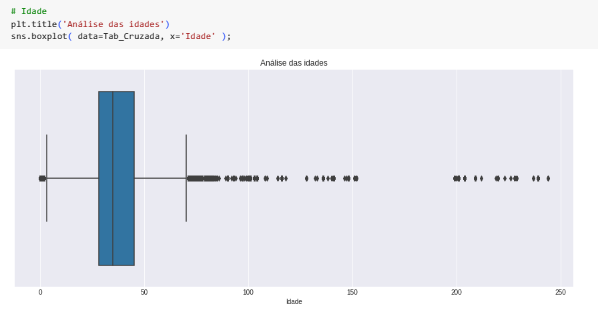
****

****

****

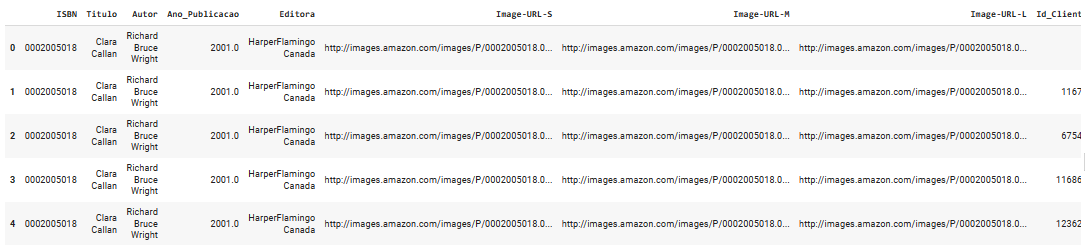
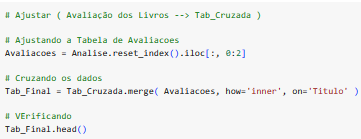
****

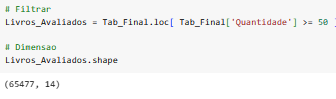
****

****

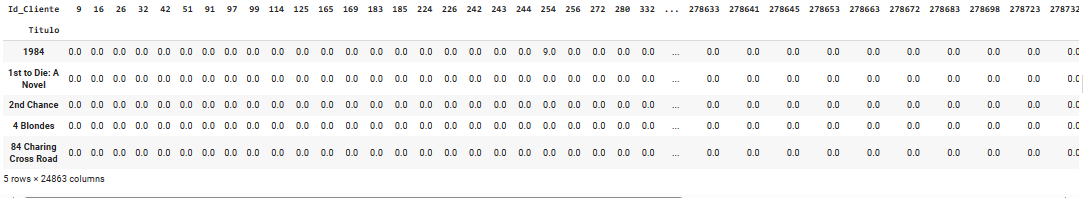
# Construção do modelo

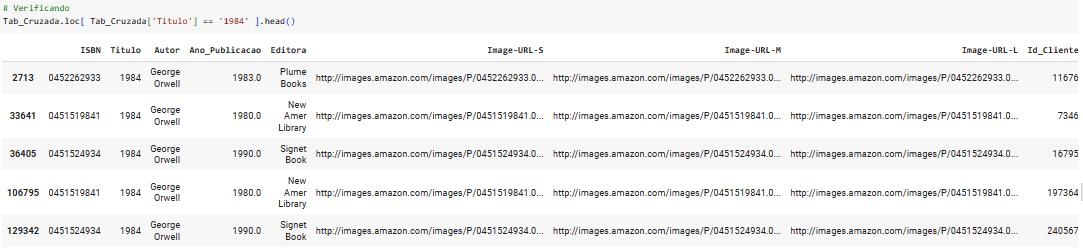
O Aprendizado por Representação é a busca por melhores representações de dados em algoritmos, muitas vezes não supervisionados. Exemplos incluem Análise de Componentes Principais e Clustering. Esses métodos transformam dados de entrada para preservar informações úteis, útil como pré-processamento para 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.

****

****

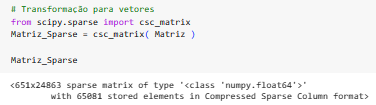
****

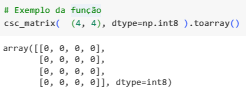
****

****

Uma matriz representativa de dados foi criada, que provavelmente representa características ou avaliações dos livros.

Essa matriz foi visualizada por meio de um gráfico de dispersão para entender a distribuição e relação dos dados.

****

****

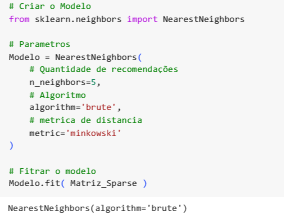
KNN - NEIGHBORS

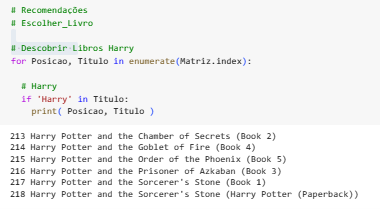
O modelo de vizinhos mais próximos é um tipo de algoritmo de aprendizado não supervisionado usado para tarefas de classificação ou regressão. Ele identifica os 'vizinhos' mais próximos de um ponto de dados em um espaço de características, com base em alguma medida de distância.

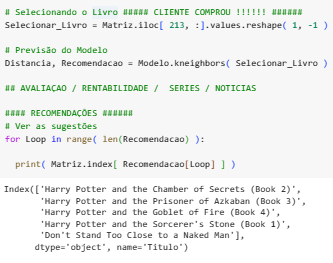
n\_neighbors=5: Este é o número de vizinhos mais próximos a serem considerados. No contexto de recomendação, significa que para cada ponto de dados, o modelo identificará os 5 vizinhos mais próximos.

algorithm='brute': Especifica o algoritmo usado para computar os vizinhos mais próximos. 'Brute' refere-se ao método de força bruta, que envolve o cálculo da distância entre todos os pares de pontos e é simples, mas pode ser ineficiente em grandes conjuntos de dados.

metric='minkowski': Define a métrica de distância utilizada para o cálculo. Minkowski é uma métrica generalizada que inclui a distância Euclidiana (quando o parâmetro de potência p=2) e a distância de Manhattan (quando p=1).

****

****

****

****

# Treinamento do Modelo

O modelo foi treinado a partir da utilização do método Euclidiano.

Aqui, um objeto NearestNeighbors é criado com dois parâmetros principais:

n\_neighbors = 2: Isso especifica que o modelo deve considerar os 2 vizinhos mais próximos de um ponto de dados ao fazer previsões. Em outras palavras, para cada ponto de dados, o modelo identificará os dois pontos mais próximos no conjunto de dados.

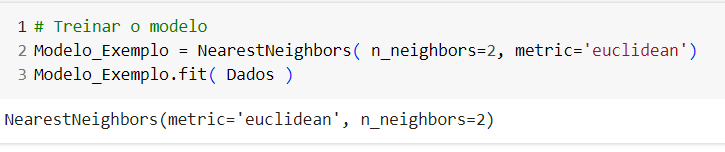
metric='euclidean': Define a métrica de distância a ser utilizada pelo modelo. 'Euclidean' refere-se à distância euclidiana, uma das métricas mais comuns para medir a distância em um espaço multidimensional. A distância euclidiana é a "distância em linha reta" entre dois pontos.

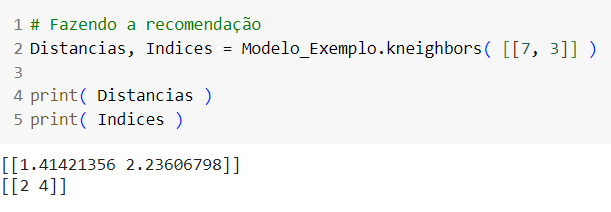
# Framework – Simulação

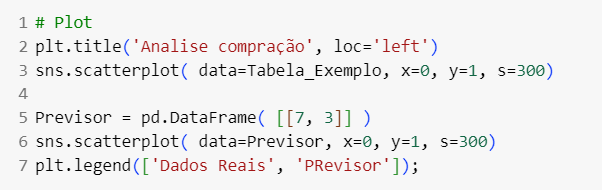
Modelo\_Exemplo.fit(Dados)

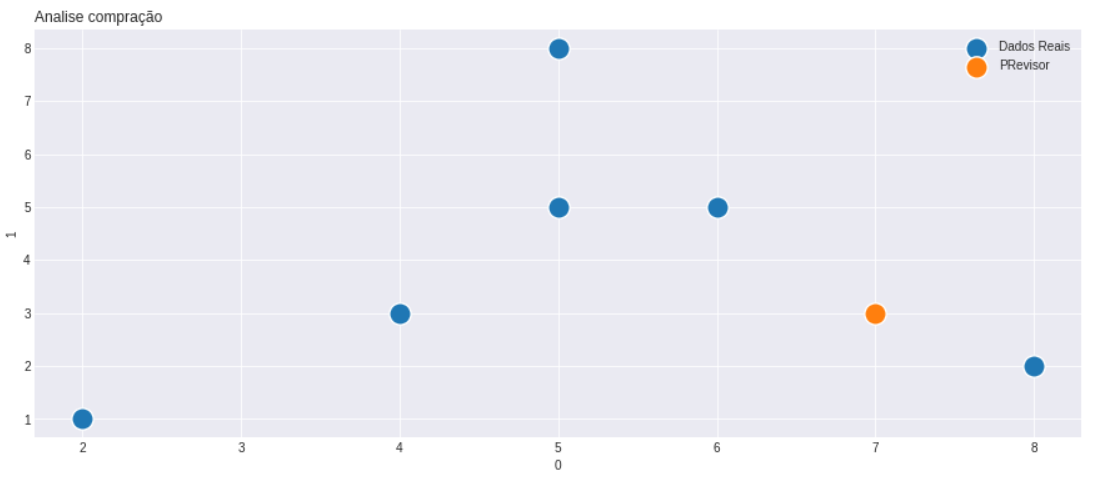
O método fit é usado para treinar o modelo Modelo\_Exemplo com os dados fornecidos. Dados deve ser um conjunto de dados numéricos, como um array do NumPy ou um DataFrame do Pandas, que representa as características dos itens (por exemplo, livros, filmes etc.) ou dos usuários.

Durante o treinamento, o modelo aprende a estrutura dos dados e fica pronto para fazer previsões. Neste contexto, "fazer previsões" significa identificar os vizinhos mais próximos de um determinado ponto.

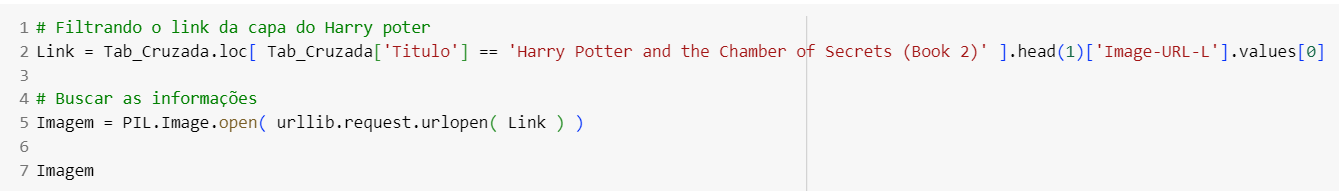
****

****

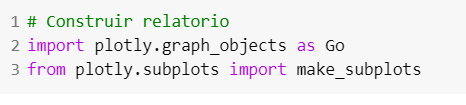
****

****

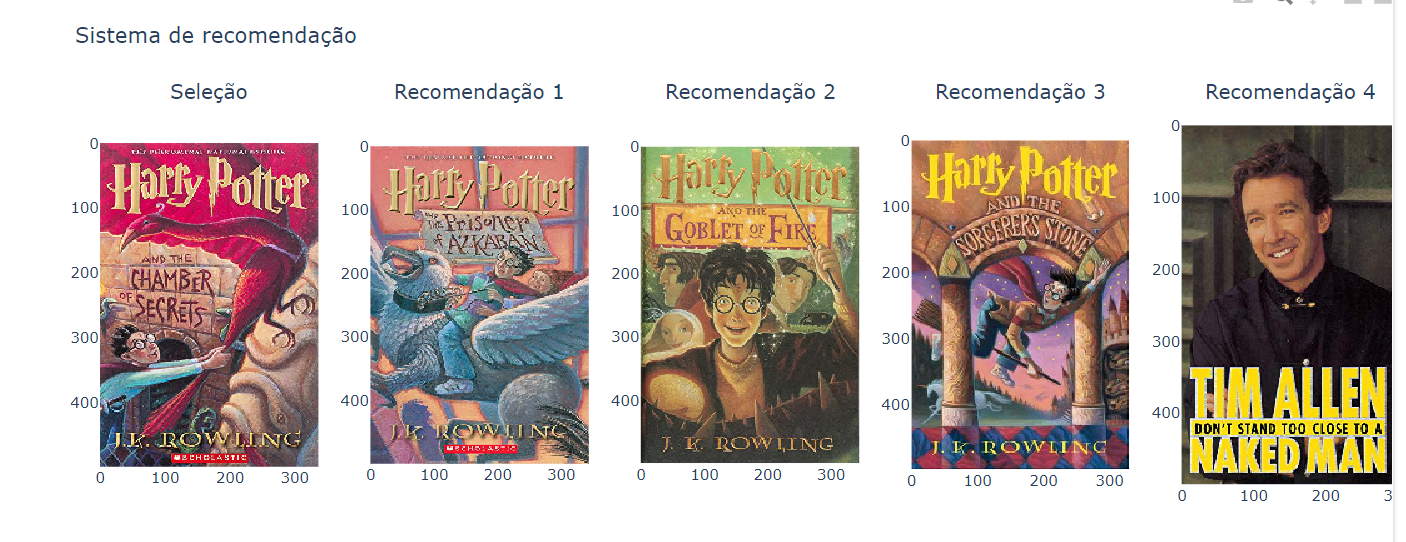
****

****

****

****

****

****

# Conclusão

Esse projeto acadêmico foi um grande passo para o desenvolvimento de um modelo de recomendação de livros usando análise de dados e aprendizado de máquina. A preparação e o enriquecimento cuidadosos dos dados, além da filtragem criteriosa, criaram uma base sólida para o modelo. A implementação do modelo com a classe NearestNeighbors do Scikit-Learn e a manipulação de estruturas de dados complexas mostraram a aplicabilidade das técnicas de aprendizado de máquina em sistemas de recomendação.

No entanto, é importante reconhecer as limitações desse trabalho. A filtragem de livros com base em um número mínimo de avaliações, embora útil para aumentar a confiabilidade das recomendações, pode ter excluído títulos menos conhecidos ou novos, o que pode limitar a diversidade das recomendações. Além disso, o modelo depende fortemente da qualidade e da integridade dos dados disponíveis, o que significa que quaisquer lacunas ou vieses nos dados podem afetar as recomendações finais.

O projeto pode ser expandido e melhorado de várias maneiras. Uma área de interesse é a incorporação de algoritmos mais sofisticados que possam lidar com a chamada "maldição da dimensionalidade", comum em sistemas de recomendação com grandes conjuntos de dados. Além disso, explorar métodos para integrar avaliações qualitativas, além das quantitativas, poderia enriquecer ainda mais as recomendações. Por fim, seria valioso investigar abordagens para reduzir o viés inerente aos dados de avaliação, garantindo assim que o modelo possa oferecer recomendações equitativas e abrangentes.

Apesar das limitações, o projeto oferece insights valiosos e uma base sólida para futuras pesquisas no campo dos sistemas de recomendação. Ele destaca a importância da análise cuidadosa de dados e da aplicação de técnicas de aprendizado de máquina, ao mesmo tempo em que aponta para desafios e oportunidades de crescimento nessa área em constante evolução.

# Links

<https://github.com/gabigermana/projeto_aplicado_III>

<https://github.com/Samuelregis/Projeto_Aplicado_3>

<https://colab.research.google.com/drive/1kb9yZR6q-FTH4fSum1DUXjKFuD-wdIna#scrollTo=-F4_xCwPBoWz>

<https://www.youtube.com/watch?v=1V1Jpp_TKfQ>

# Bibliografia

Book Recommendation Dataset | Kaggle;

https://github.com/Samuelregis/Projeto\_Aplicado\_3/tree/40e189b869705d3c614591fc8691a3b688aca566;

GOLDSCHMIDT, Ronaldo. Data Mining. [Digite o Local da Editora]: Grupo GEN, 2015. E-book. ISBN 9788595156395. Disponível em: https://app.minhabiblioteca.com.br/#/books/9788595156395/. Acesso em: 24 out. 2023;

SILVA, Leandro Augusto da; PERES, Sarajane M.; BOSCARIOLI, Clodis. Introdução à Mineração de Dados - Com Aplicações em R. [Digite o Local da Editora]: Grupo GEN, 2016. E-book. ISBN 9788595155473. Disponível em: https://app.minhabiblioteca.com.br/#/books/9788595155473/. Acesso em: 24 out. 2023;

https://scikit-learn.org/stable/modules/model\_evaluation.html;

https://scikit-learn.org/stable/auto\_examples/model\_selection/plot\_grid\_search\_digits.html#sphx-glr-auto-examples-model-selection-plot-grid-search-digits-py;

https://medium.com/analytics-vidhya/understanding-k-nearest-neighbour-algorithm-in-detail-fc9649c1d196;