Universidade Presbiteriana Mackenzie

ANDREI SOUZA DE OLIVEIRA – RA: 10408496

DANIELE DOS SANTOS ROSA – RA: 10407781

GABRIELA OHASHI DE SOUZA – RA: 10110022

MARINA OHASHI DE SOUZA – RA: 10161483

MIGUEL MAURÍCIO TADEU PITALI DA SILVA – RA: 10407541

# PROJETO APLICADO III: BOA LEITURA

São Paulo

2024

# RESUMO (Em construção)

**SUMÁRIO**

INTRODUÇÃO…… 4

OBJETIVOS E METAS.....5

FUNDAMENTAÇÃO TEÓRICA…….6

METODOLOGIA……...10

RESULTADO……..…11

ANÁLISE EXPLORATÓRIA…….....11

PRÉ-PROCESSAMENTO E LIMPEZA DOS DADOS……15

CRIAÇÃO UM SISTEMA DE RECOMENDAÇÃO DE LIVROS…….....18

AVALIAÇÃO DO DESEMPENHO DO MODELO…….....……21

CONCLUSÃO E TRABALHOS FUTUROS…….....22

BIBLIOGRAFIA……..23

# INTRODUÇÃO

Os sistemas de recomendação são ferramentas cruciais nos dias atuais, especialmente diante da abundância de informações disponíveis. Eles desempenham um papel fundamental ao ajudar os usuários a filtrar e descobrir conteúdos relevantes em meio a um mar de opções.

Esses sistemas enfrentam desafios de escalabilidade e aprimoramento das recomendações, mas são essenciais para facilitar a tomada de decisões dos usuários e melhorar sua experiência de consumo (Medeiros, 2013).

Nesse contexto, propostas de melhoria, como recomendações baseadas em competências e análises de comportamento dos usuários, ganham destaque. A constante evolução desses sistemas é crucial para acompanhar as necessidades dinâmicas dos usuários e garantir recomendações cada vez mais precisas e personalizadas.

O presente trabalho apresenta um projeto de recomendação de livros. Além de contribuir para a filtragem e descoberta de conteúdo relevante, a pesquisa busca aprimorar a precisão das recomendações e a adaptação às preferências individuais dos usuários, fortalecendo ainda mais a relevância desses sistemas na era da informação.

**Conjunto de dados escolhido**

“Book-Crossing: User review ratings”, disponível em: Book-Crossing: User review ratings (kaggle.com)

**Repositório Projeto:**

OhashiMarina/Projeto-Aplicado-III (github.com)

# OBJETIVOS E METAS

Com os dados obtidos sobre os usuários, títulos de livros e avaliações dos usuários as obras, com o intuito de otimizar e melhorar a experiência do cliente Amazon, criamos um sistema de recomendação através de algoritmos de aprendizagem de máquina.

O propósito deste projeto é sugerir livros de forma colaborativa e examinar as características dos diferentes grupos de usuários em suas leituras. Os objetivos específicos deste trabalho incluem:

- Criar um sistema de recomendação de livros.
- Analisar os grupos de usuários e descrever suas relações, destacando seus padrões de comportamento.

# FUNDAMENTAÇÃO TEÓRICA

A Análise Exploratória de Dados (EDA, *Exploratory Data Analysis*) é usada para analisar e investigar conjuntos de dados e resumir suas características principais, empregando métodos quantitativos e de visualização dos dados. Dentro do modelo CRISP-DM, a EDA compreende as fases de Entendimento e Preparação dos Dados, incluindo também a fase de Modelagem.

O modelo CRISP-DM (*Cross-Industry Standard Process for Data Mining*) é uma metodologia abrangente de mineração de dados e um modelo de processo que oferece modelo para a execução de um projeto de mineração de dados. Ele é estruturado em seis etapas distintas: compreensão do negócio, compreensão dos dados, preparação dos dados, modelagem, avaliação e implantação (Shearer, 2000).

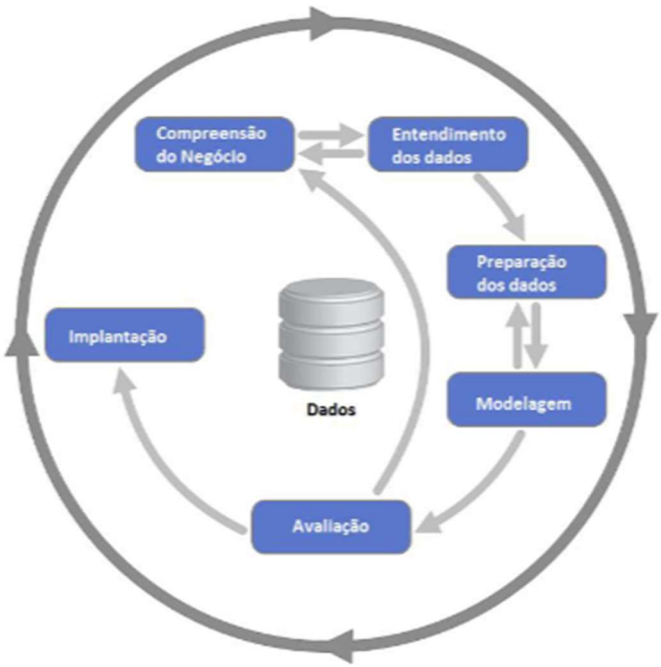



**Figura 1 - Fases do CRISP-DM (Shearer, 2000).**

A sequência de fases no CRISP-DM não segue uma ordem rígida. Na maioria dos projetos, há um movimento flexível entre as etapas, permitindo retornos quando necessário. Esta metodologia abrange descrições das fases típicas de um projeto, as tarefas necessárias em cada fase e explicações das relações entre essas tarefas.

Como modelo de processo, o CRISP-DM oferece uma visão geral do ciclo de vida da mineração de dados (Chapman, 2000).

A fase inicial envolve a aquisição do conhecimento do domínio de negócios, ou seja, compreender os objetivos do projeto de mineração de dados a partir da perspectiva do negócio. Esse entendimento se transforma em um problema de mineração de dados. É amplamente reconhecido que essa etapa é uma das mais cruciais do processo, pois é a base para o desenvolvimento de um plano preliminar do projeto de mineração, direcionado para a realização dos objetivos (Shearer, 2000).

A segunda fase, que envolve a compreensão dos dados, inicia-se com a coleta inicial de dados, com o propósito de desenvolver uma familiaridade com os dados, identificar possíveis problemas de qualidade dos dados, obter *insights* iniciais e identificar subconjuntos de interesse que possam levar à formulação de hipóteses sobre informações ocultas. Esse estágio é de fundamental importância para prevenir surpresas indesejadas durante a fase subsequente, a preparação de dados, que frequentemente é a etapa mais extensa de um projeto (Shearer, 2000).

Na fase subsequente, ocorre a preparação dos dados, que engloba todas as atividades necessárias para construir o conjunto de dados final a partir dos dados brutos iniciais. Esses dados preparados servirão como entrada para a ferramenta de modelagem na etapa seguinte. As tarefas de preparação de dados são flexíveis e podem ser realizadas em várias iterações, sem uma ordem estritamente definida. Essas atividades envolvem a seleção de tabelas, registros e atributos, bem como a realização de transformações e a limpeza dos dados (Chapman et al., 2000).

A etapa de preparação de dados é a mais crítica do processo e frequentemente a que demanda maior tempo em projetos de mineração de dados. Estima-se que, em geral, essa fase absorva entre 50-70% do tempo e dos recursos de um projeto. Alocar recursos adequados para as fases iniciais de compreensão do negócio e esforços de tratamento de dados pode ajudar a minimizar a carga relacionada a essa etapa, mas, ainda assim, será necessário um esforço substancial para a preparação e formatação dos dados para fins de mineração (IBM, 2016).

A etapa de modelagem ocorre na quarta fase do processo. Dependendo da natureza do problema de mineração, diversas técnicas podem ser aplicadas. Tipicamente, a modelagem envolve várias iterações, nas quais o analista de dados executa múltiplos modelos, inicialmente com as configurações padrão e, em seguida, ajustam os parâmetros para obter valores otimizados. Além disso, é comum retornar à fase de preparação de dados, se necessário, para realizar manipulações específicas exigidas pelos modelos (Shearer, 2010; IBM, 2016).

A sexta e última fase é a etapa de implantação, na qual os novos *insights* e conhecimentos descobertos são aplicados para promover melhorias na organização. Durante essa etapa, é fundamental que todo o conhecimento adquirido seja organizado e apresentado de maneira que o cliente possa utilizá-lo eficazmente no processo de tomada de decisão. (Shearer, 2000; Chapman et al., 2000).

## APRENDIZADO DE MÁQUINA

Nos últimos anos, houve um notável crescimento na pesquisa em aprendizado de máquina. O aprendizado de máquina é uma disciplina da inteligência artificial que se concentra no desenvolvimento de técnicas computacionais para a aquisição automática de conhecimento e na construção de sistemas capazes de aprender com base em experiências adquiridas por meio da resolução bem-sucedida de problemas anteriores (Rezende, 2005). Essa área representa a interseção entre estatística, inteligência artificial e ciência da computação, e tem aplicação significativa no reconhecimento de padrões (Guido, 2016).

O Aprendizado de Máquina (*Machine Learning*) é uma disciplina que emprega uma ampla gama de procedimentos e algoritmos para a identificação automatizada de padrões, agrupamentos e tendências nos dados, com o propósito de extrair informações valiosas para análise. Em termos simples, pode ser descrito como o uso de métodos matemáticos para treinar algoritmos a fim de reconhecer padrões (Nelli, 2015).

## APRENDIZADO DE MÁQUINA SUPERVISIONADO

O aprendizado supervisionado é um processo que envolve a extração de um modelo de conhecimento a partir de dados apresentados na forma de pares ordenados, consistindo em uma entrada e uma saída desejada. A entrada representa o conjunto de atributos ou características que são fornecidos ao algoritmo para um caso específico, enquanto a saída desejada corresponde ao valor de uma característica-alvo que se espera que o algoritmo possa prever sempre que receber determinados valores de entrada (Goldschmidt, 2015). Alguns exemplos de algoritmos que se encaixam nesse modelo incluem K-Nearest Neighbors (KNN), Modelos Lineares, Classificador Naive Bayes, Support Vector Machines e Redes Neurais Artificiais.

No presente trabalho, optou-se pela utilização do modelo KNN, como será descrito posteriormente.

# METODOLOGIA

Foi utilizada a base de dados "*Book-Crossing: User review ratings"*, disponível no *Kaggle*, na qual são apresentados conjuntos de dados que relacionam livros e avaliações de leitores. No contexto deste trabalho, entende-se por dados as principais características dos livros (ID_LIVRO, TITULO, AUTOR) e dos leitores (ID_USUARIO, ID_LIVRO, AVALIACAO).

Para realizar a análise de dados, criar modelos de aprendizado de máquina e avaliar seu desempenho, foram utilizadas as seguintes bibliotecas Python:

- Matplotlib: para criação gráficos e visualizações de dados, ajudando na interpretação e comunicação de resultados;
- Numpy: para operações numéricas, incluindo manipulação de arrays multidimensionais;
- Pandas: para realização de operações como leitura, filtragem, agregação e transformação de dados;
- Profile Report: ferramenta do *pandas_profiling* que gera um relatório detalhado sobre um *DataFrame* do pandas, fornecendo insights estatísticos e visuais sobre seus dados.
- Scikit-learn (também conhecida como *sklearn*): para implementações eficientes de uma ampla gama de algoritmos de aprendizado de máquina, incluindo *Nearest Neighbors*;
- Scipy: para criação de uma matriz esparsa (*sparse matrix*) a partir de um DataFrame do pandas; e
- Seaborn: para visualização de dados estatísticos e gráficos informativos.

Através da técnica K-Nearest Neighbors (KNN), o algoritmo KNN um método não paramétrico que se baseia na proximidade dos exemplos de treinamento para tomar decisões de classificação ou regressão. Basicamente esse algoritmo identifica através da similaridade para qual classe um ponto/dado pertence. Dentro do ambiente Machine Learning, o algoritmo de KNN é considerado um algoritmo de fácil entendimento, muitas vezes aplicado para reconhecimento de padrões, detectação de fraude e, como o caso apresentado, para sistemas de recomendações.

# RESULTADO

## Análise Exploratória dos Dados

A base de dados **BX_Books** (“Livros”) contém informações sobre as obras (livros), composto por código do livro, título da obra, autor da obra, Ano da publicação da obra, editora e outras três variáveis com link da imagem do livro, cada coluna indica um tamanho da imagem, podemos entendo S-Small (pequeno), M-Medium (médio) e L-Large (grande).

- ISBN = International Standard Book Number (Número Padrão Internacional de Livro): Variável contendo um padrão numérico criado com o objetivo de fornecer um número de identificação para cada publicação monográfica;
- Book-Title = Título do livro: Variável que informa o nome da obra/livro;
- Book-Author = Nome do autor: Coluna informando o nome do autor responsável pela obra;
- Year-Of-Publication = Ano da publicação: Dados referente ao ano em que o livro/obra foi publicado;
- Publisher = Editora: Nome da editora responsável pela obra/livro;
- Image-URL-S = URL da imagem do livro tamanho pequeno;
- Image-URL-M = URL da imagem do livro tamanho médio; e
- Image-URL-L = URL da imagem do livro tamanho grande.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 271379 entries, 0 to 271378
Data columns (total 5 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   ISBN                 271379 non-null  object
 1   Book-Title           271379 non-null  object
 2   Book-Author          271378 non-null  object
 3   Year-Of-Publication  271379 non-null  int64
 4   Publisher            271377 non-null  object
dtypes: int64(1), object(4)
memory usage: 10.4+ MB
```

A base de dados **BX-Books-Ratings** (“avaliações”) contém as avaliações dos usuários sobre as obras (livros), composto por Identificação do usuário, código do livro e avaliação do livro.

- User-ID = ID-Usuario: Variável contendo um código identificador de cada usuário;
- ISBN = International Standard Book Number (Número Padrão Internacional de Livro): Variável contendo um padrão numérico criado com o objetivo de fornecer um número de identificação para cada publicação monográfica; e
- Book-Rating = Classificação do Livro: Variável responsável pelos dados de avaliação do usuário (User-ID) sobre as obras (ISBN).

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1149780 entries, 0 to 1149779
Data columns (total 3 columns):
 #   Column       Non-Null Count    Dtype
---  ------       --------------    -----
 0   User-ID      1149780 non-null  int64
 1   ISBN         1149780 non-null  object
 2   Book-Rating  1149780 non-null  int64
dtypes: int64(2), object(1)
memory usage: 26.3+ MB
```

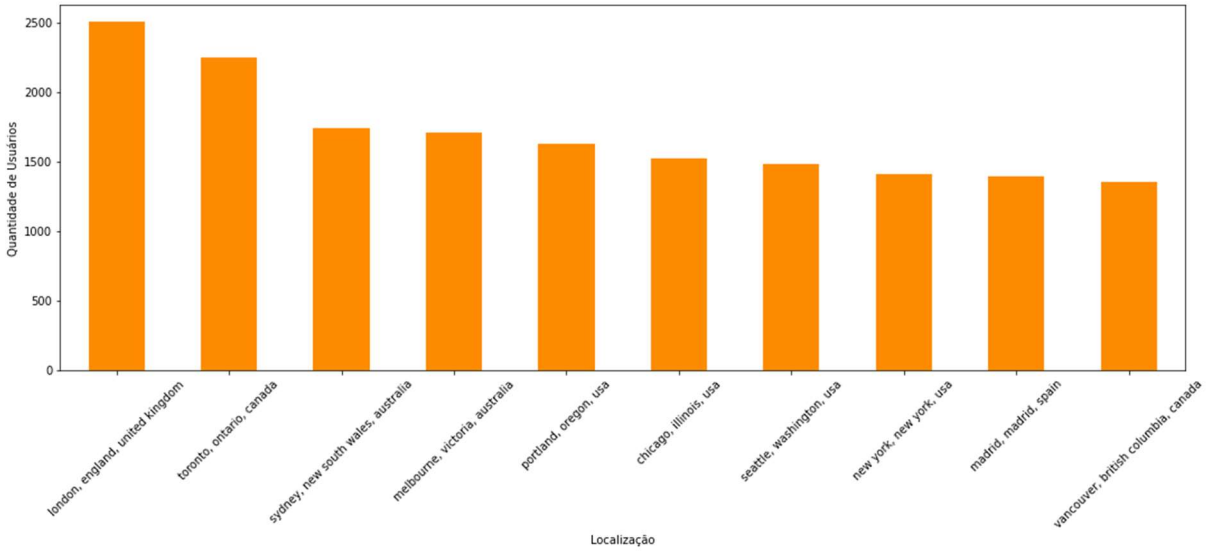
Já a base de dados **BX_Users** (“usuários”) contém informações sobre os usuários Amazon (clientes), contendo o User-ID, Localidade do usuário e idade

- User-ID = ID-Usuário: Variável contendo um código identificador de cada usuário;
- Location = Localização: Estado onde o usuário está localizado; e
- Age = Idade: Idade do usuário.

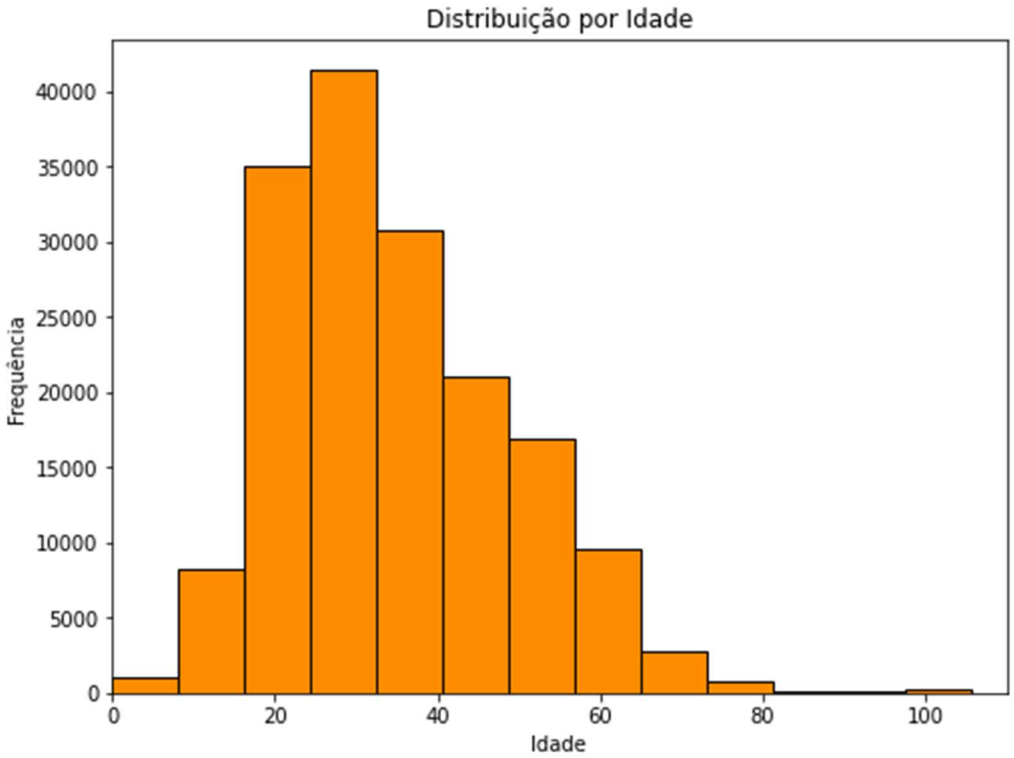
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 278858 entries, 0 to 278857
Data columns (total 3 columns):
 #   Column    Non-Null Count   Dtype
---  ------    --------------   -----
 0   User-ID   278858 non-null  int64
 1   Location  278858 non-null  object
 2   Age       168096 non-null  float64
dtypes: float64(1), int64(1), object(1)
memory usage: 6.4+ MB
```

Para a melhor avaliação dos dados da base de dados, foram feitos os gráficos:

1) Contagem das cidades com maior quantidade de usuários:




2) Distribuição dos usuários por idade:

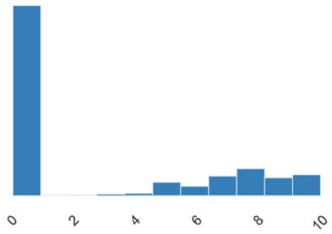


3) Avaliação dos livros:

Book-Rating
Real number (ℝ)

ZEROS

| | | | |
|---|---|---|---|
| **Distinct** | 11 | **Minimum** | 0 |
| **Distinct (%)** | < 0.1% | **Maximum** | 10 |
| **Missing** | 0 | **Zeros** | 647323 |
| **Missing (%)** | 0.0% | **Zeros (%)** | 62.8% |
| **Infinite** | 0 | **Negative** | 0 |
| **Infinite (%)** | 0.0% | **Negative (%)** | 0.0% |
| **Mean** | 2.8390215 | **Memory size** | 15.7 MiB |



## PRÉ-PROCESSAMENTO E LIMPEZA DOS DADOS

Inicialmente, foram realizadas operações para filtrar e renomear colunas no DataFrame livros. Primeiro, foram selecionadas apenas as colunas necessárias, que são 'ISBN', 'Book-Title' e 'Book-Author'. Em seguida, as colunas foram renomeadas para 'ID_LIVRO', 'TITULO' e 'AUTOR', respectivamente, usando o método 'rename' com um dicionário de mapeamento de nomes de colunas. O parâmetro 'inplace' foi definido como “True” para aplicar as mudanças diretamente ao DataFrame livros.

| | ID_LIVRO | TITULO | AUTOR |
|---|---|---|---|
| 0 | 0195153448 | Classical Mythology | Mark P. O. Morford |
| 1 | 0002005018 | Clara Callan | Richard Bruce Wright |
| 2 | 0060973129 | Decision in Normandy | Carlo D'Este |
| 3 | 0374157065 | Flu: The Story of the Great Influenza Pandemic... | Gina Bari Kolata |
| 4 | 0393045218 | The Mummies of Urumchi | E. J. W. Barber |

De forma semelhante, no DataFrame avaliações, foram selecionadas apenas as colunas necessárias, que são 'User_ID’, ‘ISBN' e 'Book-Rating' que foram renomeadas ID_USUARIO', 'ID_LIVRO' e 'AVALIACAO’, respectivamente. Da mesma forma, o DataFrame avaliações foi ajustado.

| | ID_USUARIO | ID_LIVRO | AVALIACAO |
|---|---|---|---|
| 0 | 276725 | 034545104X | 0 |
| 1 | 276726 | 0155061224 | 5 |
| 2 | 276727 | 0446520802 | 0 |
| 3 | 276729 | 052165615X | 3 |
| 4 | 276729 | 0521795028 | 6 |

Considerando a grande quantidade de avaliações classificadas como com nota zero (716.109 observações), essas instâncias foram excluídas e o DataFrame avaliações foi novamente ajustado, resultando em uma nova base de dados com 433.671 avaliações não nulas.

```
         ID_USUARIO     ID_LIVRO  AVALIACAO
1            276726   0155061224          5
3            276729   052165615X          3
4            276729   0521795028          6
6            276736   3257224281          8
7            276737   0600570967          6
...             ...          ...        ...
1149773      276704   0806917695          5
1149775      276704   1563526298          9
1149777      276709   0515107662         10
1149778      276721   0590442449         10
1149779      276723  05162443314          8

[433671 rows x 3 columns]
```

Foi realizada a contagem de avaliações por ID de livro, com a criação de uma Coluna de quantidade de avaliações no DataFrame ‘avaliacoes’ para contar o número de avaliações em cada grupo. O resultado foi armazenado em um novo DataFrame chamado 'contagem_avaliacoes'. A função 'reset_index()' foi usada para redefinir o índice do DataFrame, e o nome da coluna de contagem foi definido como 'QTDE_AVALIACOES'.

| | ID_LIVRO | TITULO | AUTOR | QTDE_AVALIACOES |
|---|---|---|---|---|
| 0 | 0195153448 | Classical Mythology | Mark P. O. Morford | NaN |
| 1 | 0002005018 | Clara Callan | Richard Bruce Wright | 9.0 |
| 2 | 0060973129 | Decision in Normandy | Carlo D'Este | 2.0 |
| 3 | 0374157065 | Flu: The Story of the Great Influenza Pandemic... | Gina Bari Kolata | 6.0 |
| 4 | 0393045218 | The Mummies of Urumchi | E. J. W. Barber | NaN |
| 5 | 0399135782 | The Kitchen God's Wife | Amy Tan | 17.0 |
| 6 | 0425176428 | What If?: The World's Foremost Military Histor... | Robert Cowley | 1.0 |
| 7 | 0671870432 | PLEADING GUILTY | Scott Turow | 1.0 |
| 8 | 0679425608 | Under the Black Flag: The Romance and the Real... | David Cordingly | NaN |
| 9 | 074322678X | Where You'll Find Me: And Other Stories | Ann Beattie | 1.0 |

Foram verificadas a existência de valores nulos dos DataFrames ‘livros’ e ‘avaliacoes’ e essas observações foram excluídas da base de dados, resultando em uma base de dados de 149.841 livros e 77.805 avaliações.

Posteriormente, nos DataFrames ‘livros’ e ‘avaliacoes’ foram selecionados apenas as avaliações cujos clientes tenham realizado mais de 9 avaliações, resultando em 5.444 observações livros e 295.561 observações em avaliações:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5444 entries, 5 to 131842
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   ID_LIVRO         5444 non-null   object
 1   TITULO           5444 non-null   object
 2   AUTOR            5444 non-null   object
 3   QTDE_AVALIACOES  5444 non-null   float64
dtypes: float64(1), object(3)
memory usage: 212.7+ KB
<class 'pandas.core.frame.DataFrame'>
Int64Index: 295561 entries, 133 to 1149747
Data columns (total 3 columns):
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
 0   ID_USUARIO  295561 non-null  int64
 1   ID_LIVRO    295561 non-null  object
 2   AVALIACAO   295561 non-null  int64
dtypes: int64(2), object(1)
memory usage: 9.0+ MB
```

Para a concatenação dos DataFrames, precisamos remover as letras do ‘ID_LIVRO’ de ambas as bases de dados.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5444 entries, 5 to 131842
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   ID_LIVRO         5444 non-null   int64
 1   TITULO           5444 non-null   object
 2   AUTOR            5444 non-null   object
 3   QTDE_AVALIACOES  5444 non-null   float64
dtypes: float64(1), int64(1), object(2)
memory usage: 212.7+ KB
```

```
ID_USUARIO    int64
ID_LIVRO      int64
AVALIACAO     int64
dtype: object
```

Foi realizada a concatenação dos DataFrames considerando o 'ID_LIVROS' para tal concatenação:

| | ID_USUARIO | ID_LIVRO | AVALIACAO | TITULO | AUTOR | QTDE_AVALIACOES |
|---|---|---|---|---|---|---|
| 0 | 276822 | 60096195 | 10 | The Boy Next Door | Meggin Cabot | 53.0 |
| 1 | 278554 | 60096195 | 9 | The Boy Next Door | Meggin Cabot | 53.0 |
| 2 | 7125 | 60096195 | 8 | The Boy Next Door | Meggin Cabot | 53.0 |
| 3 | 7346 | 60096195 | 8 | The Boy Next Door | Meggin Cabot | 53.0 |
| 4 | 8067 | 60096195 | 10 | The Boy Next Door | Meggin Cabot | 53.0 |

Ainda para a limpeza dos dados, foi realizada a verificação e exclusão de dados duplicados para que houvesse problemas de termos o mesmo usuário avaliando o mesmo livro mais de uma vez. Nossa base de dados resultante ficou com 88.579 observações.

```
# Visualizando se houve alteração na quantidade de registros
avaliacoes_e_livros.shape
```

```
(88579, 6)
```

## CRIAÇÃO UM SISTEMA DE RECOMENDAÇÃO DE LIVROS

Foi realizado um processo de pivoteamento (ou pivot) dos dados. A função `pivot_table()` foi utilizada para reorganizar o DataFrame `avaliacoes_e_livros` de modo que cada `ID_USUARIO` passasse a ser uma coluna, e o valor de nota (`AVALIACAO`) para cada livro avaliado fosse disposto em células correspondentes. A coluna `TITULO` foi utilizada como índice para as linhas do DataFrame resultante.

| ID_USUARIO | 242 | 243 | 254 | 388 | 446 | 503 | 505 | 507 | 625 | 638 | ... | 278314 | 278356 | 278390 | 278418 | 278535 | 278554 | 27858 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TITULO | | | | | | | | | | | | | | | | | | |
| 'Salem's Lot | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | Nal |
| 10 Lb. Penalty | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | Nal |
| 100 Selected Poems by E. E. Cummings | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | Nal |
| 14,000 Things to Be Happy About | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | NaN | NaN | NaN | Nal |

Os valores nulos foram substituídos por zero para que todas as células tenham um valor numérico válido, facilitando operações e cálculos subsequentes.

| ID_USUARIO | 242 | 243 | 254 | 388 | 446 | 503 | 505 | 507 | 625 | 638 | ... | 278314 | 278356 | 278390 | 278418 | 278535 | 278554 | 278582 | 27 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TITULO | | | | | | | | | | | | | | | | | | | |
| 'Salem's Lot | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| 10 Lb. Penalty | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| 100 Selected Poems by E. E. Cummings | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| 14,000 Things to Be Happy About | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |

O DataFrame livros_pivot é passado como argumento para a função csr_matrix() DO SciPY, que o converte em uma matriz esparsa no formato CSR. Uma matriz esparsa é uma matriz que possui a maioria de seus elementos iguais a zero. Essa matriz esparsa permite uma representação eficiente de conjuntos de dados que contenham principalmente zeros, especialmente em problemas de análise de dados e aprendizado de máquina, economizando memória e operações

```
# Vamos importar o csr_matrix do pacote SciPy
# Esse método possibilita criarmos uma matriz sparsa
from scipy.sparse import csr_matrix


# Vamos transformar o nosso dataset em uma matriz sparsa
livros_sparse = csr_matrix(livros_pivot)
```

```
# Tipo do objeto
type(livros_sparse)
```

Foi realizado um processo de recomendação de livros com base no modelo treinado `modelo`. O livro de interesse no exemplo foi o "The Boy Next Door". Primeiro, foi feita uma filtragem do DataFrame `livros_pivot` usando o método `filter()`, selecionando apenas as informações referentes ao livro "The Boy Next Door".

Essas informações são passadas para o método `kneighbors()` do modelo treinado, que retorna as distâncias e as sugestões dos livros mais similares ao livro fornecido. O resultado é armazenado nas variáveis `distances` e `sugestions`. Em seguida, um loop é utilizado para iterar sobre as sugestões e imprimir os títulos dos

livros mais similares, acessando os índices do DataFrame `livros_pivot` através dos índices armazenados em `sugestions`.

Dessa forma, o código está recomendando os livros mais similares ao livro "The Boy Next Door" com base no modelo treinado.

```
#The Boy Next Door
distances, sugestions = modelo.kneighbors(livros_pivot.filter(items = ['The Boy Next Door'], axis=0).values.reshape(1, -1))

for i in range(len(sugestions)):
    print(livros_pivot.index[sugestions[i]])
```

```
Index(['The Boy Next Door', 'Confessions of an Ex-Girlfriend',
       'The Awakening (Dover Thrift Editions)', 'Due di due (Bestsellers)',
       'Engaging Men (Red Dress Ink (Paperback))'],
      dtype='object', name='TITULO')
```

Posteriormente, foram testadas recomendações a partir de outros livros:

```
#Artemis Fowl (Artemis Fowl, Book 1)
distances, sugestions = modelo.kneighbors(livros_pivot.filter(items = ['Artemis Fowl (Artemis Fowl, Book 1)'], axis=0).value

for i in range(len(sugestions)):
    print(livros_pivot.index[sugestions[i]])
```

```
Index(['Artemis Fowl (Artemis Fowl, Book 1)',
       'The Arctic Incident (Artemis Fowl, Book 2)', 'Genome',
       'Due di due (Bestsellers)',
       'All I Know About Animal Behavior I Learned in Loehmann's Dressing Room'],
      dtype='object', name='TITULO')
```

```
#Hoot (Newbery Honor Book)
distances, sugestions = modelo.kneighbors(livros_pivot.filter(items = ['Hoot (Newbery Honor Book)'], axis=0).values.reshape(

for i in range(len(sugestions)):
    print(livros_pivot.index[sugestions[i]])
```

```
Index(['Hoot (Newbery Honor Book)', 'Due di due (Bestsellers)', 'Seta',
       'Jade Peony', 'Garzanti - Gli Elefanti: Gabbiano Jonathan Livingston'],
      dtype='object', name='TITULO')
```

# AVALIAÇÃO DO DESEMPENHO DO MODELO

# (Em construção)

Um bom método de avaliação de desempenho para o modelo criado seria a validação cruzada, especialmente a validação cruzada k-fold.

Na validação cruzada k-fold, o conjunto de dados é dividido em k partes (ou "dobras"). O modelo é treinado k vezes, cada vez usando k-1 partes como dados de treinamento e a parte restante como dados de teste. Isso garante que cada parte do conjunto de dados seja usada tanto para treinamento quanto para teste.

Após cada execução do modelo, métricas de desempenho como precisão, recall, F1-score, ou RMSE (Root Mean Squared Error) para modelos de regressão podem ser calculadas usando os dados de teste. Em seguida, a média dessas métricas pode ser calculada para fornecer uma estimativa geral do desempenho do modelo.

# CONCLUSÃO E TRABALHOS FUTUROS

## (Em construção)

# BIBLIOGRAFIA


CHAPMAN, P.et al.CRISP-DM 1.0: Step-by-step data mining guide.SPSS inc, v. 9, p. 13, 2000. Disponível em: https://www.kde.cs.uni-kassel.de/wp-content/uploads/lehre/ws2012-13/kdd/files/CRISPWP-0800.pdf. Acesso em: 31 mar. 2024.

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: 30 mar. 2024.

GUIDO, A. C. S. Introduction to Machine Learning with Python. [S.l.]: O'Reilly Media, 2016.

IBM SPSS. IBM SPSS modeler text analytics 16 user guide. 2016. Disponível em: https://public.dhe.ibm.com/software/analytics/spss/documentation/modeler/16.0/en/ta_guide_book.pdf. Acesso em: 29 mar. 2024.

MEDEIROS, I. Estudo sobre sistemas de recomendação colaborativos. [S.l.]: Recife, 2013.

REZENDE, S. O. Sistemas inteligentes: fundamentos e aplicações. [S.l.]: Manole, 2005.

SHEARER, C. The CRISP-DM Model: The New Blueprint for Data Mining. Journal of Data Warehousing, v. 5, n. 4, 2000. Disponível em: https://www.academia.edu/42079490/CRISP_DM_The_New_Blueprint_for_Data_Mining_Colin_Shearer_Fall_2000. Acesso em: 29 mar. 2024.