**UNIVERSIDADE PRESBITERIANA MACKENZIE**

**TECNÓLOGO EM CIÊNCIAS DE DADOS**

Caroline Ribeiro Ferreira – 10408052

Lais César Fonseca – 10407066

Leonardo dos Reis Olher – 10407752

Liliane Gonçalves de Brito Ferraz – 10407087

Múcio Emanuel Feitosa Ferraz Filho – 10218691

Otavio Bernardo Scandiuzzi – 10407867

# SISTEMA DE RECOMENDAÇÃO DE LIVROS

**SÃO PAULO**

**2024**

## Sumário

# 1. GLOSSÁRIO

- **Colaboratory:** Conhecido também como “Colab”, é um produto do Google Research, área de pesquisas científicas do Google. O Colab permite que qualquer pessoa escreva e execute código *Python* arbitrário pelo navegador e é especialmente adequado para aprendizado de máquina, análise de dados e educação.

- **Clustering:** É uma forma de organizar dados por meio do agrupamento destes em conjuntos, a partir da maior similaridade existente entre os dados de um mesmo conjunto que os de outro, com base em algum critério pré-determinado.

- **DataFrame:** É uma estrutura de dados bidimensional com os dados alinhados de forma tabular em linhas e colunas.

- **Datasets:** conjuntos de dados organizados em um formato similar ao das tabelas, com linhas e colunas que contém informações sobre determinado tema.

- **GitHub:** GitHub é uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git. Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou Open Source de qualquer lugar do mundo.

- **Kaggle:** É uma plataforma para aprendizado de ciência de dados. É também uma comunidade, a maior da internet, para assuntos relacionados com Data Science.

- ***Machine Learning* (Aprendizado de máquina):** É um subcampo da Engenharia e da ciência da computação que evoluiu do estudo de reconhecimento de padrões e da teoria do aprendizado computacional em inteligência artificial.

- ***Overfitting* (Sobreajuste):** Sobre-ajuste ou sobreajuste é um termo usado em estatística para descrever quando um modelo estatístico se ajusta muito bem ao conjunto de dados anteriormente observado, mas se mostra ineficaz para prever novos resultados. É comum que a amostra apresente desvios causados por erros de medição ou fatores aleatórios.

- **Oversampling**: Consiste em gerar novos exemplos para a classe minoritária, de forma a aumentar sua representatividade no conjunto de dados.

- **Python:** É uma linguagem de programação de alto nível, interpretada de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte. Foi lançada por Guido van Rossum em 1991.

- **PyTorch**: É uma biblioteca de aprendizado de máquina baseada na biblioteca Torch, usada para aplicações como visão computacional e processamento de linguagem natural

- **Streaming:** Fluxo contínuo, fluxo de média, fluxo de mídia ou transmissão contínua, é uma forma de distribuição digital, em oposição à descarga de dados.

## 2. RESUMO

Neste projeto, propomos o desenvolvimento de um sistema de recomendação de livros utilizando técnicas de aprendizado de máquina e análise estatística preditiva. O objetivo principal é melhorar a experiência do usuário ao fornecer recomendações personalizadas individual com base em seu histórico de leituras.

Para o desenvolvimento do projeto utilizaremos um dataset público, que possuem acesso livre no site Kaggle criando um modelo de recomendação de livros, para recomendar livros semelhantes ao leitor com base em seu interesse. Para isto, utilizaremos dos métodos adquiridos nos componentes curriculares de introdução a ciência de dados, como aplicação da linguagem *Python* com o uso do Colab.

A abordagem do projeto visa aplicarmos o método de filtragem baseada em colaboração, no qual precisamos construir uma máquina preditiva que, com base nas escolhas de leituras de outras pessoas, o livro seja recomendado a outras pessoas com interesses semelhantes.

# 3. INTRODUÇÃO

No atual cenário literário, a vasta gama de opções de livros disponíveis desafia os leitores a navegarem por um oceano de informações para encontrar obras que verdadeiramente cativem e ressoem com seus interesses individuais. Diante desse desafio, surge a motivação para o desenvolvimento deste projeto: um “Sistema de Recomendações de Livros”. Esta introdução abordará o contexto do trabalho, a motivação, a justificativa e os objetivos, proporcionando uma visão holística do propósito e das metas deste projeto aplicado.

De acordo com o portal de notícias G1, pertencente ao grupo Globo, apenas 16% da população brasileira comprou algum livro no ano de 2023, verifica-se dessa forma a necessidade de impulsionar um mercado tão importante para o desenvolvimento intelectual das pessoas. Um possível reflexo disso foi o baixo desempenho dos jovens brasileiros no Ranking PISA, acendendo um sinal de alerta para esta situação. Desta forma, buscamos encontrar meios que facilitem o acesso à leitura, de forma que seja otimizada a escolha de bons livros que possam impulsionar o desenvolvimento de hábitos saudáveis naqueles que o utilizarem.

No contexto atual, a abundância de opções literárias é tanto um privilégio quanto um desafio. A tecnologia, no entanto, oferece oportunidades significativas para aprimorar a experiência de leitura, simplificando a escolha de livros por meio de sistemas inteligentes de recomendação. Estes, ao analisarem padrões de leitura, hábitos e preferências individuais, podem fornecer sugestões personalizadas, transformando a busca por livros em uma jornada mais enriquecedora.

A motivação para este projeto emerge da percepção das dificuldades enfrentadas pelos leitores contemporâneos ao selecionar livros entre uma ampla variedade de opções. A escolha deste tema é impulsionada pelo desejo de simplificar e aprimorar a experiência de leitura, tornando-a mais personalizada e acessível. Além disso, a paixão pela literatura e o reconhecimento do impacto positivo que uma recomendação bem-sucedida pode ter na vida do leitor constituem fortes motivadores.

Justificamos o desenvolvimento deste sistema pela sua capacidade potencial de democratizar o acesso à literatura, ampliando o alcance de obras significativas e diversificando o repertório literário dos usuários. Acreditamos que o projeto pode não apenas facilitar a escolha de livros, mas também promover a descoberta de novos gêneros e autores, enriquecendo assim a experiência de leitura de forma significativa.

Os sistemas de recomendação de livros desempenham um papel crucial não só na facilitação da descoberta de novas obras para os leitores, mas também na promoção da diversidade

cultural, no estímulo ao diálogo intercultural e na ampliação do conhecimento sobre uma vasta gama de culturas e perspectivas, com isto atendemos as questões relacionadas aos ODS (Objetivos de Desenvolvimento Sustentável no Brasil).

Ao sugerir livros que se relacionam com interesses específicos ou áreas de estudo, esses sistemas auxiliam os usuários a enriquecer seu entendimento sobre diferentes culturas, períodos históricos e pontos de vista. Além disso, ao possibilitar a descoberta de livros de uma variedade de gêneros, autores e temas, os sistemas de recomendação contribuem para enriquecer a diversidade cultural presente na leitura dos usuários.

A abordagem híbrida foi a escolhida para compor este projeto, aliando a filtragem colaborativa, baseada em interações de usuários, e a filtragem baseada em conteúdo, que foca nas características dos livros presentes nos arquivos. O uso da aprendizagem supervisionada auxiliará na obtenção de informações de múltiplos tipos de dados, como dados textuais, avaliações de usuários e demais metadados editoriais, todos esses divididos entre três datasets escolhidos para compor este projeto. Isso resultará em um modelo final de *Machine Learning* que pode prever diretamente os interesses dos usuários com a adição de itens específicos.

## 4. OBJETIVO

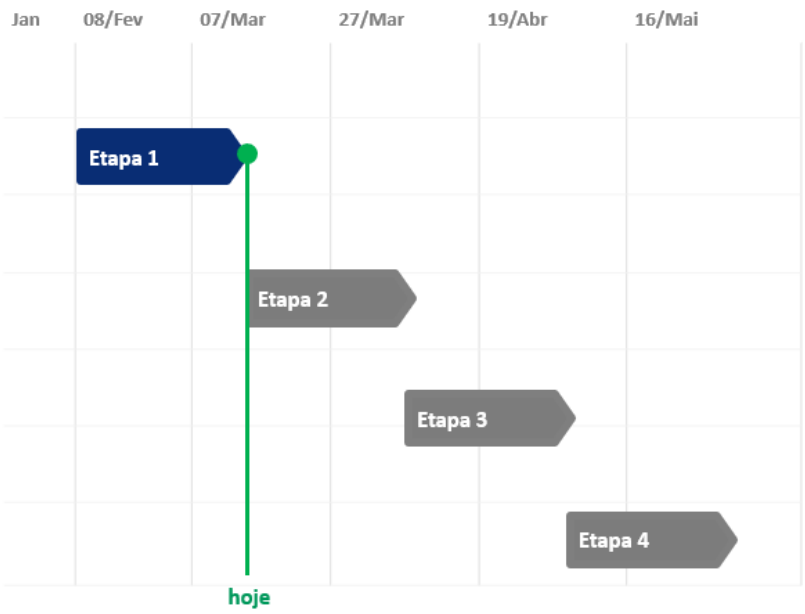
Este projeto tem como principal objetivo desenvolver um Sistema de Recomendações de Livros eficiente e inovador, aplicando técnicas avançadas de recomendação para proporcionar sugestões personalizadas aos usuários. Além disso, busca-se:

- Construir uma plataforma interativa que permita aos usuários interagirem entre si, compartilhando recomendações, resenhas e insights literários.
- Analisar e avaliar criticamente diferentes abordagens de recomendação, propondo melhorias e ajustes conforme necessário.
- Aumentar o engajamento dos usuários e estimular o hábito da leitura, com o intuito de contribuir para a formação de uma comunidade de leitores mais diversificada, tornando a leitura mais inclusiva e abrangente, contribuir para uma sociedade mais informada, empática e culturalmente enriquecida.

Através desses objetivos, almejamos não apenas criar um sistema eficaz de recomendação de livros, mas também fomentar uma comunidade engajada e apaixonada pela leitura. Esses elementos fundamentais formam a base do nosso projeto, delineando a trajetória que percorreremos para alcançar nossos objetivos e oferecer uma solução valiosa no universo literário contemporâneo.

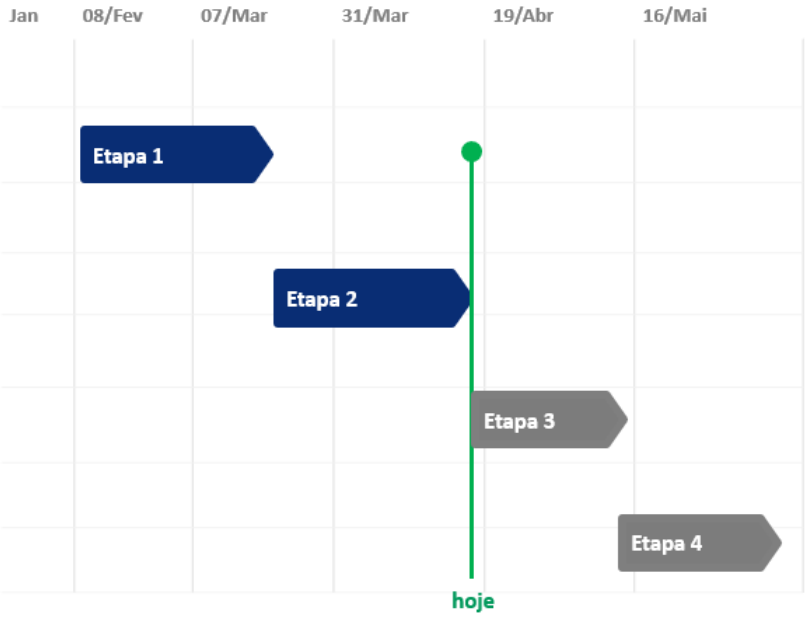
## 5. CRONOGRAMA

Reportar o desenvolvimento das etapas propostas e o percentual de evolução de entregas referente ao componente curricular de Projeto Aplicado III do curso de Tecnologia em Ciências de Dados.



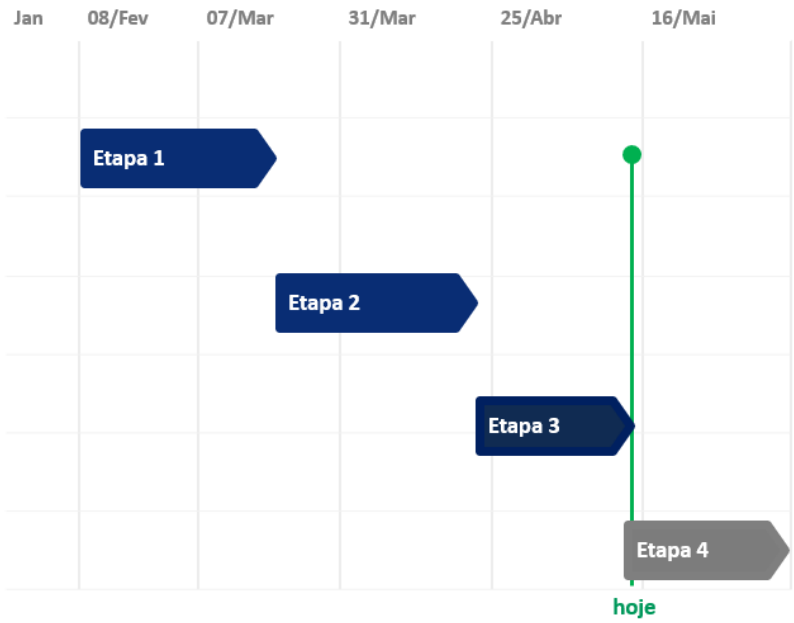

**Entregáveis Etapa 1**

- **Definição do grupo** — Grupo ✓
- **Definição do tema** — Grupo ✓
- **Escolha do Dataset** — Grupo ✓
- **Introdução/Objetivo** — Caroline, Lais, Leonardo e Múcio ✓
- **Cronograma** — Liliane ✓
- **Documento** — Liliane ✓
- **Github** — Otávio ✓



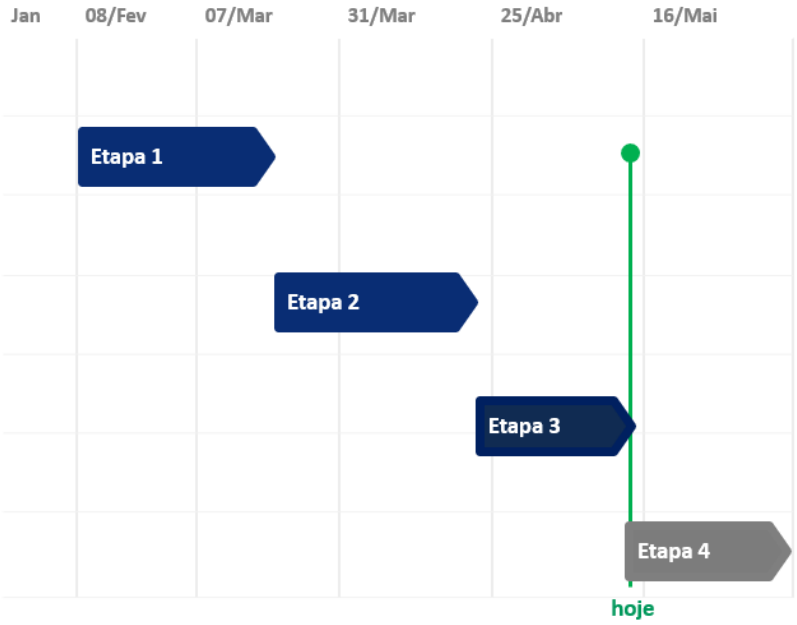

**Entregáveis Etapa 2**

- **Definição de bibliotecas** — Leonardo, Múcio, Otávio ✓
- **Análise Exploratória** — Caroline, Laís, Liliane ✓
- **Tratar e preparar a base** — Caroline, Laís, Liliane ✓
- **Definir a técnica** — Leonardo, Múcio, Otávio ✓
- **Prova de Conceito** — Leonardo, Múcio, Otávio ✓
- **Forma de Avaliação** — Leonardo, Múcio, Otávio ✓
- **Referência Bibliográfica** — Grupo ✓
- **Documento e Github** — Caroline, Laís, Liliane ✓




**Entregáveis Etapa 3**

- **Analisar os resultados** — Leonardo e Múcio ☑
- **Ajustar o pipeline de dados** — Leonardo e Múcio ☑
- **Reavaliar desempenho modelo** — Leonardo ☑
- **Organizar a descrição das técnicas** — Liliane ☑
- **Descrever a metodologia** — Carol e Lais ☑
- **Documento e Github** — Liliane e Otávio ☑




**Entregáveis Etapa 4**

- **Organizar os resultados** — Grupo ☐
- **Analisar os resultados** — Grupo ☐
- **Documentar os resultados** — Grupo ☐
- **Documentar as conclusões** — Grupo ☐
- **Documentação do projeto** — Grupo ☐
- **Vídeo do projeto** — Grupo ☐
- **Artefatos de software** — Grupo ☐

# 6. APRESENTAÇÃO DO METADADOS

Os metadados escolhidos para o desenvolvimento do nosso projeto aplicado de “Sistema de Recomendação de Livros”, tem as características apresentadas a seguir:

## 6.1. Tipo de arquivo

As bases de dados adquirida é de extensão csv. Denominadas de “BX_Books.csv”, “BX-Users.csv” e “Book-Ratings.csv”.

## 6.2. Origem dos dados

Os dados são de domínio público/aberto, do site da Kaggle.

https://www.kaggle.com/code/isaienkov/book-review-ratings-analysis-and-visualization/input

## 6.3. Sensibilidade / LGPD

Esta base de dados não possui dados sensíveis e está de acordo com a Lei Geral de Proteção de Dados Pessoas – LGPD.

## 6.4. Proprietário do dado

KOSTIANTYN ISAIENKOV

## 6.5. Descrição e atributos dos dados

O conjunto de dados BX_Books.csv é o metadados que apresenta informações dos livros, no qual contém 271.379 linhas e 8 colunas, e dispões dos seguintes atributos:

- ISBN: International Standard Book Number (Padrão Internacional de Numeração de Livro) é um padrão numérico criado com o objetivo de fornecer uma espécie de "RG" para publicações de livros.
- Book-Title: Apresenta o Título do livro no qual constam 242.154 livros distintos registrados.
- Book-Author: Apresenta o Nome do Autor do livro o qual constam livros de 102.028 autores distintos registrados.
- Year-Of-Publication: Apresenta o ano de publicação do livro no qual constam publicações de 116 anos cadastrados distintos, buscando os tops 20 anos com mais livros publicados observamos que consta livros de 1985 a 2004.
- Publisher: Apresenta o nome da Editora do livro, no qual apresenta 16.806 nomes distintos de editoras registrada.
- Image-URL: Apresenta a url da imagem/foto da capa do livro.

O conjunto de dados BX-Users.csv é o metadados que apresenta informações dos usuários, no qual contém 278.858 linhas e 3 colunas, e dispões dos seguintes atributos:

- User-ID: Código do usuário no qual contém 278.858 usuários distintos cadastrado.
- Location: Localização do usuário, apresentado por cidade, estado e país. Apresenta 57.339 localizações distintas cadastrada.
- Age: Apresenta a idade do usuário, no qual constam 165 idades distintas cadastrada.

O conjunto de dados BX-Book-Ratings.csv é o metadados que apresenta informações das avaliações dos livros realizadas pelos leitores, no qual contém 1.149.780 linhas e 3 colunas, e dispões dos seguintes atributos:

- User-ID: Código do usuário no qual contém 105.283 usuários distintos cadastrado.
- ISBN: Código de registro do livro, no qual contém 340.556 livros distintos cadastro.
- Book-Rating: Apresenta a nota de avaliação do usuário para cada livro, no qual contém 11 tipos de notas distinta cadastrada.

## 7. BIBLIOTECAS PYTHON

Para a implementação do sistema de recomendação, após estudos e análises realizadas sobre o tema, avaliamos que as bibliotecas da linguagem *Python* que serão utilizadas neste projeto para coletar os dados, processamento e tratamento dos dados, modelagem e avaliação. Selecionamos as bibliotecas a seguir:

### 1. Pandas:

A biblioteca Pandas é uma poderosa ferramenta de linguagem de programação *Python*, de código aberto e gratuito, que desempenha um papel fundamental na análise, limpeza e manipulação de dados. Além disso, ela permite a criação de gráficos e a manipulação de tabelas, tornando-a uma escolha essencial para programadores e cientistas de dados. *Python* é amplamente utilizado em diversas áreas, incluindo aprendizado de máquina, cibe segurança, mineração de dados, ciência de dados, programação web e muitas outras. A biblioteca Pandas é uma das razões pelas quais *Python* é tão popular para lidar com grandes estruturas de dados.

A biblioteca Pandas oferece uma ampla gama de recursos e funcionalidades para programadores e analistas de dados e suas funcionalidades são:

- **Manipulação de Dados**: O Pandas permite importar, manipular e processar dados de diversas fontes, como arquivos CSV, TSV ou bancos de dados SQL. Ele transforma esses dados em objetos *Python* chamados DataFrames, que se assemelham a tabelas, facilitando a análise e a manipulação.
- **Análise de Dados**: Com o Pandas, é possível realizar análises elaboradas dos dados. Ele oferece funções para agregar, agrupar, filtrar e calcular estatísticas, tornando a análise de dados eficiente e poderosa.
- **Limpeza de Dados**: A biblioteca simplifica a tarefa de limpar dados, permitindo a detecção e remoção de valores ausentes, duplicados e inconsistentes. Isso resulta em conjuntos de dados mais confiáveis e prontos para análise.
- **Visualização de Dados**: O Pandas se integra à biblioteca Matplotlib, facilitando a criação de gráficos e visualizações de dados. Isso torna a comunicação dos insights obtidos a partir dos dados mais eficazes.
- **Manipulação de Séries Temporais:** O Pandas oferece suporte robusto para lidar com dados de séries temporais, permitindo análises avançadas de dados ao longo do tempo.
- **Combinação de DataFrames:** É possível combinar DataFrames horizontal ou verticalmente, o que é útil quando se lida com grandes conjuntos de dados fragmentados.
- **Trabalho com Dados Categóricos**: O Pandas facilita a categorização de dados, simplificando a criação de modelos de aprendizado de máquina e a visualização de dados categóricos.

A biblioteca Pandas oferece várias vantagens distintas para os programadores e analistas de dados, são estas as principais vantagens:

- **Produtividade Elevada:** O Pandas é altamente produtivo e eficiente, economizando tempo na análise e manipulação de dados.
- **Facilidade de Acesso:** A biblioteca é conhecida por sua facilidade de uso e acessibilidade, tornando-a adequada para iniciantes e especialistas.
- **Versatilidade:** O Pandas é extremamente versátil e pode ser aplicado em diversas áreas, desde análise de dados até aprendizado de máquina.
- **Comunidade Ativa:** Com uma comunidade de colaboradores ativos, o Pandas está sempre em constante desenvolvimento e melhoria.
- **Integração com Outras Bibliotecas:** A integração com bibliotecas como Matplotlib e NumPy amplia ainda mais suas capacidades.
- **Manipulação de Grandes Dados:** Mesmo em grandes conjuntos de dados, o Pandas mantém seu desempenho e eficiência, tornando-o uma escolha sólida para projetos de qualquer escala.

## 2. Seaborn:

O Seaborn é uma biblioteca de visualização de dados em *Python* que se baseia no popular Matplotlib. Ela foi projetada para criar gráficos estatísticos elegantes e informativos com facilidade, exigindo apenas algumas linhas de código. O Seaborn é particularmente útil ao lidar com dados complexos, fornecendo diversas ferramentas para simplificar o processo de visualização e apresentação de resultados. Suas principais funções são:

- **Gráficos de Barras**: Os gráficos de barras são ideais para visualizar dados categóricos. O Seaborn oferece diversos tipos de gráficos de barras, incluindo gráficos de barras agrupadas, empilhadas e horizontais. Esses gráficos ajudam a representar informações de forma clara e eficaz.
- **Gráficos de Dispersão**: Os gráficos de dispersão são usados para visualizar a relação entre duas variáveis. O Seaborn oferece vários tipos de gráficos de dispersão, como aqueles com linhas de regressão e gráficos de dispersão com hexágonos, que ajudam a identificar tendências e padrões nos dados.
- **Gráficos de Caixa**: Os gráficos de caixa são úteis para representar a distribuição de uma variável numérica. O Seaborn oferece diferentes tipos de gráficos de caixa, incluindo aqueles com distribuição e pontos, permitindo a análise da dispersão e dos valores atípicos nos dados.
- **Gráficos de Densidade**: Os gráficos de densidade ajudam a visualizar a distribuição de uma variável numérica. O Seaborn oferece gráficos de densidade uni variada e bivariada, fornecendo insights sobre a distribuição conjunta de duas variáveis numéricas.

Suas vantagens de utilização são:

- **Facilidade de Uso**: O Seaborn é reconhecido por sua facilidade de uso, permitindo que os usuários criem visualizações complexas com código conciso.
- **Estilo Elegante**: A biblioteca oferece uma ampla variedade de estilos e paletas de cores elegantes, tornando as visualizações atraentes e informativas.
- **Integração com o Matplotlib**: O Seaborn é baseado no Matplotlib, o que significa que você pode combinar as funcionalidades dessas duas bibliotecas, aproveitando o poder do Matplotlib com a simplicidade do Seaborn.
- **Visualização Estatística**: O foco do Seaborn está na visualização estatística, o que o torna uma escolha qualitativa para análises exploratórias de dados e apresentação de resultados em um formato informativo.
- **Flexibilidade:** Apesar de sua simplicidade, a Seaborn oferece opções avançadas de personalização para atender às necessidades específicas de visualização.

### 3. Numpy:

NumPy, abreviatura de "*Numeric Python*", é uma biblioteca poderosa da linguagem de programação *Python* que se destaca por suas estruturas de dados multidimensionais, conhecidas como arrays. Além disso, o NumPy oferece uma extensa coleção de rotinas e funções que facilitam o processamento de arrays,

O NumPy é extremamente reconhecido por fornecer um conjunto abrangente de recursos e operações que simplifica o desenvolvimento de cálculos numéricos. Esses cálculos desempenham um papel fundamental em diversas áreas, incluindo:

- **Modelos de *Machine Learning***: Em algoritmos de *Machine Learning*, é comum realizar uma variedade de cálculos numéricos, como multiplicação de matrizes, transposição e adição. O NumPy oferece uma biblioteca eficiente para executar esses cálculos de maneira fácil e rápida. Os arrays do NumPy são frequentemente usados para armazenar dados de treinamento e intervalos de modelos de *Machine Learning*.
- **Processamento de Imagem e Computação Gráfica**: Para manipular imagens de forma eficiente, o NumPy fornece funções que simplificam tarefas como espelhamento e rotação de imagens, entre outras operações de processamento de imagem.
- **Tarefas Matemáticas**: O NumPy é uma ferramenta útil para executar diversas tarefas matemáticas, incluindo integração numérica, diferenciação, interpolação e extrapolação. Além disso, a biblioteca oferece funções internas para álgebra linear e geração de números aleatórios. O NumPy é frequentemente combinado com outras bibliotecas, como SciPy e Matplotlib, para realizar tarefas complexas de análise e visualização de dados. Ele também é considerado uma alternativa ao MATLAB para aplicações matemáticas.

### 4. Matplotlib:

A biblioteca Matplotlib é uma ferramenta poderosa na linguagem de programação *Python*, voltada para a plotagem de gráficos 2D. Ela foi lançada em 2003 e seu desenvolvimento foi liderado pelo neurologista americano John D. Hunter. A origem do Matplotlib está ligada à pesquisa de pós-doutorado de Hunter, onde ele visualiza dados de eletrocorticografia em pacientes com epilepsia.

O Matplotlib oferece uma ampla gama de funcionalidades e recursos para criar gráficos 2D de alta qualidade e suas funcionalidades são:

- **Visualização de Dados**: A principal função do Matplotlib é criar gráficos e visualizações de dados de maneira eficaz. Ela suporta uma variedade de tipos de gráficos, incluindo gráficos de dispersão, barras, linhas, histogramas, entre outros.
- **Personalização**: A biblioteca permite personalizar todos os aspectos dos gráficos, incluindo núcleos, tamanhos, fontes e estilos. Isso possibilita a criação de visualizações únicas e informativas.
- **Suporte a Diferentes Backends**: A Matplotlib oferece suporte a uma ampla variedade de backends e saídas. Isso significa que os gráficos criados podem ser salvos em diferentes formatos de arquivo e exibições em várias plataformas, tornando-os altamente portáteis.
- **Integração com outras bibliotecas:** O Matplotlib é frequentemente usado em conjunto com outras bibliotecas de análise de dados, como Pandas e NumPy, facilitando a criação de visualizações a partir de dados processados por essas ferramentas.

Existem várias vantagens em escolher a Matplotlib para criar gráficos e visualizações de dados, sendo elas:

- **Facilidade de Uso**: O Matplotlib é conhecido por sua facilidade de uso, tornando-a acessível tanto para iniciantes quanto para profissionais experientes.
- **Ampla Comunidade**: A biblioteca possui uma comunidade ativa de desenvolvedores e usuários.

### 5. SKLearn ou Scikit-Leanr:

O *Scikit-learn*, originalmente chamado de *scikits.learn*, é uma biblioteca de aprendizado de máquina de código aberto para *Python*. Oferece uma ampla variedade de algoritmos para classificação, regressão e agrupamento, como máquinas de vetores de suporte, florestas aleatórias, gradient boosting, k-means e DBSCAN. Essa biblioteca é projetada para integrar-se perfeitamente com as bibliotecas *Python* numéricas e científicas, como NumPy e SciPy. Além disso, o *Scikit-learn* é uma ferramenta gratuita e versátil para modelagem estatística, análise de dados e aprendizado supervisionado e não supervisionado, tornando-o uma escolha popular para *Machine Learning* em *Python*. Suas principais aplicações são:

- **Algoritmos de Classificação**: Identificam categorias associadas aos dados, úteis para tarefas como classificar e-mails como spam ou não.
- **Algoritmos de Regressão**: Criam modelos para compreender a relação entre dados de entrada e saída, usados, por exemplo, para prever o comportamento dos preços das ações.

- **Algoritmos de Agrupamento (*Clustering*):** Agrupam automaticamente dados com características semelhantes, como segmentar clientes por idade ou localização.
- **Redução de Dimensionalidade:** Diminuem o número de variáveis para análise, aprimorando eficiência na visualização e processamento de dados.
- **Seleção de Modelo:** Oferecem ferramentas para comparar, validar e selecionar os melhores modelos e parâmetros para projetos de ciência de dados.
- **Pré-processamento:** Extrai e normaliza recursos nos dados, sendo útil para transformar dados de entrada, como texto, durante a análise.

Principais Recursos são:

- **Pré-processamento:** Realiza transformações e manipulações nos dados brutos, incluindo tratamento de valores ausentes, conversão de valores categóricos em formatos numéricos e seleção de recursos.
- **Estimadores:** Oferece uma variedade de algoritmos predefinidos para aprendizado supervisionado e não supervisionado, como classificadores, regressões, SVM, árvores de decisão e algoritmos de *clustering*.
- **Avaliação do Modelo:** Fornece métricas estatísticas para avaliar o desempenho dos modelos, incluindo validação cruzada e funções de métricas individuais.
- **Otimização do Modelo:** Permite a otimização de hiper parâmetros, incluindo aprendizado conjunto, pesquisa em grade e pesquisa aleatória para melhorar o desempenho dos modelos de aprendizado de máquina

**6. Torch:**

É uma biblioteca de aprendizado de máquina de código aberto para *Python*. Oferece uma ampla variedade de algoritmos para redes neurais, otimização, processamento de tensores e muito mais. Projetado para integração perfeita com bibliotecas *Python* numéricas, como NumPy, e científicas, como SciPy, Torch é amplamente utilizada em pesquisas e aplicações de aprendizado profundo. Suas principais aplicações incluem:

- **Redes Neurais**: Implementação de redes neurais artificiais, incluindo modelos convolucionais, recorrentes e completamente conectados.
- **Processamento de Tensores**: Manipulação eficiente de estruturas de dados multidimensionais, essenciais para operações de álgebra linear e aprendizado profundo.
- **Otimização**: Fornecimento de otimizadores para ajustar os parâmetros dos modelos durante o treinamento, utilizando técnicas como descida de gradiente estocástica e algoritmos de segunda ordem.

- **Visualização**: Ferramentas para visualizar e diagnosticar modelos durante o treinamento, permitindo análise detalhada do comportamento do modelo e do processo de aprendizado.

Principais Recursos são:

- **Módulos e Camadas**: Fornece uma variedade de módulos e camadas predefinidos para construir redes neurais, incluindo convolucionais, lineares, recorrentes e de pooling.
- **Autograd**: Capacidade de calcular automaticamente gradientes para tensores, permitindo o treinamento eficiente de modelos por meio de retro propagação.
- **CUDA e Computação na GPU**: Suporte para computação na GPU por meio do CUDA, permitindo treinamento acelerado de modelos em hardware compatível.
- **API Flexível**: Interface de programação de aplicativos flexível que permite construir e personalizar modelos de aprendizado profundo de acordo com as necessidades específicas do projeto.
- **Integração com Ferramentas Existentes**: Compatibilidade com várias bibliotecas *Python*, incluindo NumPy e SciPy, para facilitar a interoperabilidade com outras ferramentas de análise de dados e aprendizado de máquina.

### 7. Imblearn.under_sampling:

A biblioteca Imblearn.under_sampling em *Python* é uma ferramenta eficaz para lidar com problemas de classificação de dados desbalanceados, onde uma classe possui uma quantidade significativamente maior de exemplos do que as outras. No qual pode ser útil se houver uma grande discrepância na quantidade de avaliações para diferentes livros.

Ela oferece diversas técnicas de subamostragem para balancear a distribuição de classes, melhorando a performance dos modelos de *Machine Learning*.

Principais Funcionalidades são:

- **Diversas Técnicas de Subamostragem**: A biblioteca oferece uma variedade de algoritmos de subamostragem, incluindo:
  - ✓ **Random Under-Sampling**: Remove aleatoriamente exemplos da classe majoritária.
  - ✓ **NearMiss:** Remove exemplos da classe majoritária que são mais próximos de exemplos da classe minoritária.

- ✓ **Tomek Links:** Remove pares de exemplos da classe majoritária que são mais próximos entre si do que de qualquer exemplo da classe minoritária.
- ✓ **Condensed Nearest Neighbour (CNN):** Remove exemplos da classe majoritária que não são necessários para classificar corretamente os exemplos da classe minoritária.

- **Balanceamento de Classes**: Ajusta a distribuição de classes para que todas as classes tenham aproximadamente o mesmo número de exemplos.
- **Preservação de Informação**: Algumas técnicas, como NearMiss e Tomek Links buscam preservar informações importantes da classe majoritária ao remover exemplos menos informativos.
- **Integração Facilitada**: A biblioteca pode ser facilmente integrada com outros pacotes de *Machine Learning*, como scikit-learn.

Pontos Positivos:

- **Melhoria da Performance dos Modelos**: Ao balancear as classes, a biblioteca pode significativamente melhorar a performance dos modelos de *Machine Learning*, especialmente para a classe minoritária.
- **Variedade de Técnicas Disponíveis**: A diversidade de algoritmos de subamostragem permite escolher a técnica mais apropriada para cada problema.
- **Facilidade de Utilização**: A biblioteca é conhecida por ser de fácil utilização e possui uma documentação clara, facilitando sua implementação.

Pontos Negativos:

- **Potencial Perda de Informação**: Algumas técnicas de subamostragem podem remover informações valiosas da classe majoritária, o que pode impactar negativamente o desempenho do modelo.
- **Risco de *Overfitting*:** A subamostragem excessiva pode levar ao *Overfitting,* onde o modelo se adapta demasiadamente aos dados de treinamento e falha em generalizar para novos dados.
- **Não é uma Solução Universal:** Assim como outras técnicas de subamostragem, a Imblearn.under_sampling pode não ser a melhor solução para todos os problemas de dados desbalanceados, sendo necessário considerar alternativas como o *oversampling* ou algoritmos específicos para esse tipo de dado.

Aplicações:

A biblioteca Imblearn.under_sampling é comumente utilizada em problemas de classificação onde as classes estão desbalanceadas, tais como:

- **Detecção de fraude**: Identificar transações fraudulentas em um conjunto de dados onde a maioria das transações são legítimas, como em transações financeiras.
- **Diagnóstico de doenças**: Classificar pacientes com base em seus sintomas, onde a maioria dos pacientes não tem a doença.
- **Análise de sentimento**: Classificar textos como positivos, negativos ou neutros, onde a maioria dos textos são neutros.

## 8. ANÁLISE EXPLORATÓRIA

Aplicando os métodos de análise exploratória dos dados podemos obter o entendimento de todo o conteúdo dos dados presente no dataset, para uma avaliação de como será aplicado o tratamento necessário para a melhor adequação ao modelo proposto e assim obter a medidas de acurácia mais precisas, na implementação do algoritmo de recomendação.

Neste processo de exploração passamos por algumas etapas:

- Importar as bibliotecas de necessário para realização de analise exploratória, gráficos e tratamento, sendo elas: Pandas, Numpy, Seaborn, Matplotlib.pyplot e Matplotlib.ticker.
- Importar os três conjuntos de dados: informações sobre livros, usuários e avaliações.
- Os conjuntos de dados foram unificados, com base nos identificadores únicos de usuários e livros.
- Realizou-se a análise das dimensões dos dataframes, identificação dos atributos e tipos de dados.
- Foram verificadas as distribuições de dados, identificando a quantidade de usuários, livros e avaliações disponíveis.
- Foi realizado o tratamento de valores ausentes, removendo as observações com informações faltantes.
- Calculou-se a distribuição dos usuários por faixa etária e a distribuição da quantidade de leitores por país.
- Exploramos a contagem de livros por ano de publicação, editora e autor, identificando os mais populares.
- Apresentamos as avaliações dos usuários por livro, identificando os mais avaliados.

Os dados foram tratados e preparados para o treinamento do modelo de recomendação. Isso incluiu a união dos conjuntos de dados relevantes, limpeza de dados para remover duplicatas e valores ausentes, bem como a transformação de variáveis categóricas em formato adequado para o treinamento do modelo.

- Realizou-se o tratamento dos dados removendo as observações com valores nulos.
- Selecionou-se apenas os usuários que fizeram mais de 50 avaliações para garantir uma base de dados mais robusta.
- Após o tratamento, o conjunto de dados final possui 765.672 registros e 8 atributos.

O código de análise exploratória consta disponibilizado em nosso diretório do Github: https://github.com/OtavioBer/ProjAplicadoIII/blob/OtavioBer-patch-1/Projeto_Aplicado_III_An%C3%A1lise_Explorat%C3%B3ria.ipynb

## 9. METODOLOGIA

Para o desenvolvimento do sistema de recomendação de livros abordaremos a técnica de “recomendação” através de algoritmos de aprendizado de máquina a DNNs (Deep Neural Networks) e o método de avaliação MSE (Mean Squared Error), para oferecer sugestões personalizadas com base nas preferências do usuário.

Neste processo de implementação seguir um processo bem definido é muito importante para um desenvolvimento bem estruturado e eficaz, e para isto seguimos o modelo tradicional, se apresenta-se na figura 1.

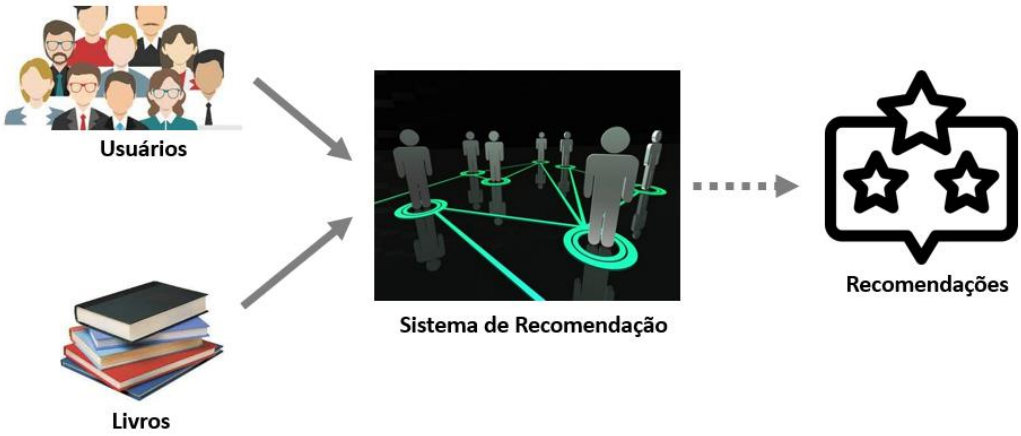
Figura 1: Processo de Recomendação


Elaborado pelo autor, 2024.

Sistemas de recomendação são algoritmos e técnicas que oferecem sugestões personalizadas de itens ou informações para usuários com base em suas preferências, comportamentos passados ou características similares de outros usuários, para exemplificar como esta tecnologia de recomendação funciona, ilustramos na figura 2, este sistema funciona a partir de dois tipos de informações: Características (sobre itens, categorias etc.) e interações

entre usuários e item (avaliações, números de compras, etc.). Com base nisso, temos as recomendações baseada em conteúdo e a filtragem colaborativa.

Figura 2: Sistema de Recomendação




Elaborado pelo autor, 2024.

Um mecanismo de recomendação é uma classe de aprendizado de máquina que oferece sugestões relevantes ao cliente. Antes do sistema de recomendação, a grande tendência para comprar era aceitar sugestões de amigos. Mas agora o Google sabe quais notícias você vai ler, o Youtube sabe que tipo de vídeos você vai assistir com base em seu histórico de pesquisa, histórico de exibição ou histórico de compra.

Um sistema de recomendação ajuda uma organização a criar clientes fiéis e construir a confiança deles nos produtos e serviços desejados para os quais vieram em seu site. Os sistemas de recomendação de hoje são tão poderosos que também podem lidar com o novo cliente que visitou o site pela primeira vez. Eles recomendam os produtos que estão em alta ou com alta classificação e também podem recomendar os produtos que trazem o máximo de lucro para a empresa.

Existem três tipos principais de sistemas de recomendação:

1. **Filtragem Colaborativa:**
    - A filtragem colaborativa é uma abordagem popular em sistemas de recomendação, que faz recomendações com base nas preferências de usuários semelhantes.
    - Existem duas principais técnicas de filtragem colaborativa: baseada em usuário e baseada em item.
    - Na filtragem colaborativa baseada em usuário, recomenda-se itens com base nas preferências de usuários semelhantes.

- Já na filtragem colaborativa baseada em item, os itens são recomendados com base em sua similaridade com outros itens preferidos pelo usuário.

2. **Filtragem Baseada em Conteúdo:**
    - A filtragem baseada em conteúdo recomenda itens com base em características ou atributos dos itens e preferências do usuário.
    - Essa técnica considera tanto as características dos itens quanto o perfil do usuário para fazer recomendações personalizadas.
    - Por exemplo, se um usuário gosta de livros de mistério, a filtragem baseada em conteúdo recomendará outros livros de mistério com características semelhantes.

3. **Sistemas Híbridos:**
    - Os sistemas híbridos combinam diferentes abordagens, como filtragem colaborativa e filtragem baseada em conteúdo, para melhorar a qualidade das recomendações.
    - Essa abordagem pode mitigar as limitações de cada técnica individualmente e fornece recomendações mais precisas e personalizadas.

Os sistemas de recomendação são fundamentais para melhorar a experiência do usuário, aumentar o engajamento e auxiliar os usuários na descoberta de novos produtos, serviços ou conteúdos que possam ser de seu interesse. Eles são aplicados em uma variedade de setores.

Para o nosso modelo de recomendação de livros, optamos por implementar uma abordagem híbrida, combinando elementos de filtragem colaborativa e filtragem baseada em conteúdo. Utilizamos a filtragem colaborativa para identificar padrões de preferências de usuários semelhantes e recomendar livros com base nas avaliações de usuários com perfis semelhantes. Além disso, incorporamos a filtragem baseada em conteúdo para considerar características dos livros, como gênero, autor e editora, e personalizar ainda mais as recomendações com base nas preferências individuais dos usuários. Essa abordagem híbrida permite que nosso modelo aproveite as vantagens de ambas as técnicas, fornecendo recomendações mais precisas e relevantes aos usuários.

# 10. MÉTODO DE RECOMENDAÇÃO

Para o treinamento do modelo de recomendação, optamos por utilizar redes neurais profundas (DNN) com *Embeddings*, uma abordagem que se destaca pela sua capacidade de capturar relações complexas entre os dados e aprender representações eficazes para a recomendação de livros.

**O que são Redes Neurais Profundas (DNN)**

As redes neurais profundas é um tipo de modelo de aprendizado de máquina que consiste em múltiplas camadas de neurônios artificiais. Essas camadas são organizadas em uma arquitetura profunda, permitindo que o modelo aprenda representações hierárquicas dos dados, capturando características complexas e abstratas.

Em nosso modelo, utilizamos *embeddings* para representar usuários e livros de forma vetorizada em um espaço de características latentes.

Os *embeddings* são vetores de números reais de dimensões reduzidas, que são aprendidos durante o treinamento do modelo.

Essas representações densas e de baixa dimensão capturam informações importantes sobre os usuários e os livros, como preferências, padrões de comportamento e características intrínsecas dos itens.

A utilização de redes neurais profundas com *embeddings* oferece várias vantagens para o nosso modelo de recomendação de livros:

- **Captura de Relações Complexas**: As DNNs são capazes de capturar relações não lineares e complexas entre os usuários e os livros, considerando múltiplos fatores e interações entre eles.
- **Representações Eficientes**: Os *embeddings* aprendidos fornecem representações eficazes e compactas dos usuários e dos livros, permitindo uma modelagem mais precisa das preferências e características individuais.
- **Generalização e Personalização**: O modelo é capaz de generalizar padrões de comportamento para fazer recomendações para novos usuários e, ao mesmo tempo, personalizar as recomendações com base nas preferências individuais de cada usuário.

### Avaliação de Desempenho do Modelo

O desempenho do modelo foi avaliado utilizando a métrica de erro quadrático médio (MSE), calculada sobre um conjunto de dados de teste separado. Quanto menor o MSE, melhor é o desempenho do modelo em prever as avaliações dos usuários.

O MSE (Mean Squared Error), ou Erro Quadrático Médio, é uma métrica comumente utilizada para avaliar a precisão de um modelo de regressão. Ele mede a média dos quadrados dos erros entre os valores previstos pelo modelo e os valores reais observados nos dados de teste. Matematicamente, o MSE é calculado pela média dos quadrados das diferenças entre as previsões do modelo ($\hat{y}$) e os valores reais ($y$) para cada amostra nos dados de teste.

### Interpretação do MSE

- Quanto menor o valor do MSE, melhor é o desempenho do modelo. Isso significa que as previsões do modelo estão mais próximas dos valores reais.
- Um MSE igual a zero indicaria um modelo perfeito, onde as previsões do modelo são exatamente iguais aos valores reais para todas as amostras de teste.
- No entanto, um MSE muito baixo nos dados de treinamento pode indicar superajustamento (*Overfitting*), onde o modelo se ajusta muito bem aos dados de treinamento, mas não generaliza bem para novos dados.

### Avaliação no Contexto do Modelo de Recomendação

No contexto do modelo de recomendação apresentado, o MSE é calculado para avaliar a precisão das previsões do modelo em relação às avaliações reais dos usuários para os livros. Uma vez que o modelo é treinado e otimizado para minimizar o MSE durante o treinamento, a avaliação do MSE nos dados de teste fornece uma medida objetiva do quão bem o modelo generaliza para novos dados não vistos. Um MSE baixo indica que o modelo é capaz de fazer previsões precisas das preferências dos usuários para os livros, enquanto um MSE alto indica que o modelo tem dificuldade em fazer previsões precisas e pode precisar de ajustes ou melhorias.

### Utilidade do MSE

O MSE é uma métrica robusta e amplamente utilizada para avaliar a precisão de modelos de regressão, incluindo modelos de recomendação. Ele fornece uma medida quantitativa do desempenho do modelo, facilitando a comparação entre diferentes modelos e ajustes.

No contexto de um sistema de recomendação, um baixo MSE indica que o sistema é capaz de fazer recomendações precisas e relevantes para os usuários, contribuindo para uma melhor experiência do usuário e possivelmente aumentando a satisfação e a retenção do cliente.

**Medida de Acurácia**

Em sistemas de recomendação de livros a medida de acurácia é usada para avaliar o desempenho do sistema, geralmente comparando as recomendações feitas pelo sistema com as classificações reais dos usuários. A acurácia é então calculada dividindo-se o número de predições corretas pelo número total de predições. Essa maneira de cálculo serve para identificar se o modelo previu corretamente os resultados esperados. Ou seja, avalia a exatidão da correspondência entre o valor esperado (ideal) e o valor real ou medido.

A medida de acurácia implementada em nosso algoritmo de recomendação de livros é calculada durante a avaliação do modelo. No código, a precisão é calculada utilizando a seguinte fórmula:

$$Precisão = \frac{Número\ de\ previsões}{Número\ total\ de\ previsões}$$

Essa medida de acurácia é utilizada para avaliar o desempenho do modelo na classificação correta das recomendações de livros como “alta” ou “baixa”. É importante ressaltar que a precisão é apenas uma das métricas que podem ser utilizadas para avaliar o desempenho do modelo de recomendação. Outras métricas, como recall, F1-score, entre outras, também podem ser relevantes dependendo do contexto específico do problema.

Ao avaliar a eficácia de sistemas de recomendação, é fundamental compreender e medir as métricas de acurácia. Essas métricas permitem determinar a precisão das recomendações e sua capacidade de refletir os interesses reais dos usuários.

A aplicação prática das métricas de acurácia envolve o cálculo e a avaliação dessas métricas com base nos dados disponíveis. Utilizar ferramentas e bibliotecas específicas pode facilitar esse processo e fornecer insights valiosos.

A preparação dos dados para calcular as métricas de acurácia envolve a seleção apenas dos itens com avaliações explícitas dos usuários, garantindo a precisão dos resultados.

Ferramentas como a classe EvalRec do HexMax oferecem recursos para calcular e avaliar métricas de acurácia, permitindo uma análise detalhada da precisão das recomendações.

A avaliação das métricas de acurácia pode ser realizada variando os modelos de recomendação, o número de recomendações e aplicando filtros por categoria de item, oferecendo insights sobre os pontos fortes e fracos das abordagens de recomendação.

A avaliação de métricas de acurácia é crucial para identificar áreas de melhoria nos sistemas de recomendação e garantir que as recomendações atendam às expectativas dos usuários.

A compreensão das métricas de acurácia permite identificar os modelos de recomendação mais eficazes e aprimorar a precisão das recomendações oferecidas aos usuários.

A análise das métricas de acurácia em diferentes cenários, como a variação de modelos e o número de recomendações, fornece insights valiosos para otimizar o desempenho dos sistemas de recomendação.

Avaliar as métricas de acurácia em sistemas de recomendação permite uma iteração contínua e aprimoramento dos algoritmos, resultando em recomendações de maior qualidade e relevância para os usuários.

**Matriz de Confusão**

Uma matriz de confusão em um algoritmo de recomendação de livros pode ajudar a avaliar o desempenho do sistema ao comparar as recomendações feitas com as preferências reais dos usuários. Ela permite identificar quantas recomendações corretas foram feitas (verdadeiros positivos), quantas recomendações foram feitas incorretamente (falsos positivos), bem como os livros que foram omitidos e que talvez devessem ter sido recomendados (falsos negativos). Isso ajuda a ajustar e melhorar o algoritmo para fornecer recomendações mais precisas no futuro.

A matriz de confusão é uma ferramenta da área de aprendizado de máquina que mostra a performance de um modelo ao comparar as previsões com os valores reais. No contexto de recomendação de livros, a matriz de confusão pode ajudar a entender quais livros foram recomendados corretamente, quais não foram recomendados quando deveriam e vice-versa. Isso permite ajustar e aprimorar o algoritmo para fornecer recomendações mais precisas aos usuários.

A matriz de confusão é dividida em quatro quadrantes, conforme demonstrado na figura 3.

1. Verdadeiro Positivo (TP): Casos em que o modelo previu corretamente a classe positiva.
2. Falso Positivo (FP): Casos em que o modelo previu incorretamente a classe positiva.
3. Verdadeiro Negativo (TN): Casos em que o modelo previu corretamente a classe negativa.
4. Falso Negativo (FN): Casos em que o modelo previu incorretamente a classe negativa.

Figura 3: Matriz de Confusão

| | | Valor Predito | |
|---|---|---|---|
| | | **Sim** | **Não** |
| **Real** | **Sim** | Verdadeiro Positivo (TP) | Falso Negativo (FN) |
| | **Não** | Falso Positivo (FP) | Verdadeiro Negativo (TN) |

Fonte: https://diegonogare.net/2020/04/performance-de-machine-learning-matriz-de-confusao/#google_vignette

Essa estrutura permite calcular métricas importantes, como precisão, recall, F1-score e AUC-ROC, que são cruciais para avaliar a qualidade de um modelo de classificação.

Em resumo, a matriz de confusão é uma ferramenta fundamental para avaliar o desempenho de algoritmos de classificação em ciência de dados, fornecendo insights sobre onde o modelo está acertando e onde está errando. A matriz é uma tabela que tem as previsões do modelo em uma dimensão (geralmente nas colunas) e as classes reais dos dados na outra dimensão (geralmente nas linhas).

A implementação da matriz de confusão em nosso algoritmo de recomendação de livros tem a finalidade de avaliar o desempenho do modelo de recomendação de forma mais detalhada e entender onde o modelo está acertando e onde está errando. Que foi importante para:

- Avaliação do Desempenho
- Identificação de Padrões de Erro
- Ajuste de Parâmetros
- Validação do Modelo.

# 11. IMPLEMENTAÇÃO DO MODELO

A implementação do modelo de recomendação de livros, foi realizada utilizando a linguagem *Python*, no qual apresenta uma série de etapas necessárias para sua construção, no qual iremos descrever as funcionalidades de cada etapa da implementação deste modelo:

1. **Importação de Bibliotecas**: Importa as bibliotecas necessárias para manipulação de dados, visualização e modelagem, como pandas, numpy, seaborn, matplotlib, torch, scikit-learn, imblearn.under_sampling, onde foi descrito as funcionalidades de cada uma na seção 7. Biblioteca *Python*.

2. **Verificação de GPU**: Verifica se há disponibilidade de processamento da GPU para acelerar a computação, exibindo o nome da GPU se estiver disponível.

3. **Funções de Codificação e Decodificação**: Codificar dados para aplicar ao treinamento do modelo, onde são definidas três funções principais:
   - ✓ encoder_df: Codifica os dados do DataFrame original usando LabelEncoder (usado para normalizar rótulos).
   - ✓ predict_encoder: Codifica os dados de predição usando os mesmos objetos de codificação gerados no treinamento.
   - ✓ predict_uncoder: Decodifica os resultados da predição para torná-los compreensíveis.

Essas funções são cruciais para preparar os dados antes do treinamento do modelo e para interpretar os resultados após a predição.

4. **Carregamento de Dados**: Os conjuntos de dados de usuários, classificações e livros são carregados a partir de arquivos CSV, com delimitadores específicos e codificação ISO-8859-1.

5. **Manipulação de Dados**: Realizar a limpeza, pré-processamento e transformação dos dados brutos em um formato adequado para análise, incluindo remoção de duplicidades e/ou valores ausentes e adequação de variáveis categóricas. Extraímos o país do local de cada usuário e armazenamos em uma nova coluna chamada “Country”. Juntamos os conjuntos de dados de classificações, usuários e livros em um único DataFrame, utilizando as colunas de identificação “User-ID” e “ISBN”. E na etapa de limpeza dos dados, removemos valores ausentes, duplicatas e colunas desnecessárias.

6. **Preparação dos Dados para o Modelo**: Definir as colunas relevantes para o modelo, selecionar apenas as colunas relevantes e redefinir o índice do DataFrame resultante. Com a função “avaliacoes[‘HIGH_RATING’]” criamos uma nova coluna indicando se a avaliação do livro foi alta (>= 8).

7. **Retirada de Avaliações de Baixa Frequência**: Removemos os usuários que avaliaram menos de 2 livros com pontuação alta (8, 9 ou 10) para garantir a qualidade dos dados.

8. **Codificação de Dados**: Utilizamos a função “encoder_df” para codificar os dados antes de alimentar o modelo. Os objetos de codificação são armazenados para uso posterior.

9. **Balanceamento da Base de Dados**: Utilizando a aplicação de “Undersampling” que consiste em remover exemplos da classe majoritária para tornar a proporção entre as classes mais equilibrada, onde atuamos com 3 funções importantes:
   - ✓ “avaliacoes[‘HIGH_RATING’].value_counts()”: trata a contagem de avaliações altas e baixas para verificar o desbalanceamento inicial da base de dados.
   - ✓ “RandomUnderSampler”: Aplica a técnica de “undersampling” para reduzir a classe majoritária, preservando a classe minoritária. Isso é feito para balancear a distribuição das classes.
   - ✓ “train_test_split”: Divide os dados balanceados em conjuntos de treinamento e teste, com 80% dos dados destinados ao treinamento e 20% ao teste. Os dados são embaralhados antes da divisão.

Essas etapas são essenciais para lidar com o desbalanceamento de classes, o que pode impactar negativamente o desempenho do modelo de recomendação.

10. **Definição do Modelo de Recomendação**: Criação do modelo de recomendação com pytorch. Na etapa de “class ModeloRecomendacao”: É uma classe que herda de “nn.Module” (classe base para todos os módulos de rede neural) e define a arquitetura do modelo de recomendação. O modelo utiliza camadas de *embedding* para codificar as diferentes características dos dados, seguidas por várias camadas lineares com ativação ReLU (que plica a função de unidade linear retificada elemento a elemento). A função “forward”: Define o fluxo de passagem direta (forward pass) do modelo, onde as entradas são processadas pelas camadas definidas anteriormente.

11. **Definição do Conjunto de Dados para Treinamento**: A função “class AvaliacoesDataset”: É uma classe de conjunto de dados que herda de “Dataset” e é usada para encapsular os dados de treinamento. Ela implementa os métodos “__len__” e “__getitem__” para permitir o acesso aos dados de forma eficiente durante o

treinamento. Cada item retornado consiste em uma entrada (X) e sua respectiva saída (Y), convertidos para tensores do PyTorch.

12. **Definição do Dispositivo de Computação**: A função “device” verifica se há disponibilidade de GPU (cuda) e define o dispositivo de computação como “cuda” se estiver disponível, caso contrário, usa a CPU (“cpu”).

13. **Hiper parâmetros**: A função “lr” define a taxa de aprendizagem para o otimizador Adam. E o “load_checkpoint”: Booleano que indica se deve carregar um modelo pré-treinado a partir de um arquivo de checkpoint.

14. **Inicialização do Modelo de Recomendação**: Na função “n” calcula o número máximo de categorias para cada uma das cinco características dos dados. Esses valores são usados para inicializar o modelo de recomendação. Já a “model”: Instancia o modelo de recomendação (“ModeloRecomendacao”) com base nos valores máximos calculados anteriormente. O modelo é transferido para o dispositivo de computação definido. Com a função “optimizer” inicializa o otimizador Adam para atualizar os parâmetros do modelo durante o treinamento.

Adam refere-se a um algoritmo de otimização onde “combina técnicas em um algoritmo de aprendizagem eficiente. Como esperado, este é um algoritmo que se tornou bastante popular como um dos algoritmos de otimização mais robustos e eficazes para uso no aprendizado profundo”.

15. **Carregamento de Modelo Pré-Treinado** (se aplicável): Com a função “load_checkpoint” aplica se for verdadeiro, carrega os pesos do modelo e o estado do otimizador de um arquivo de checkpoint previamente salvado.

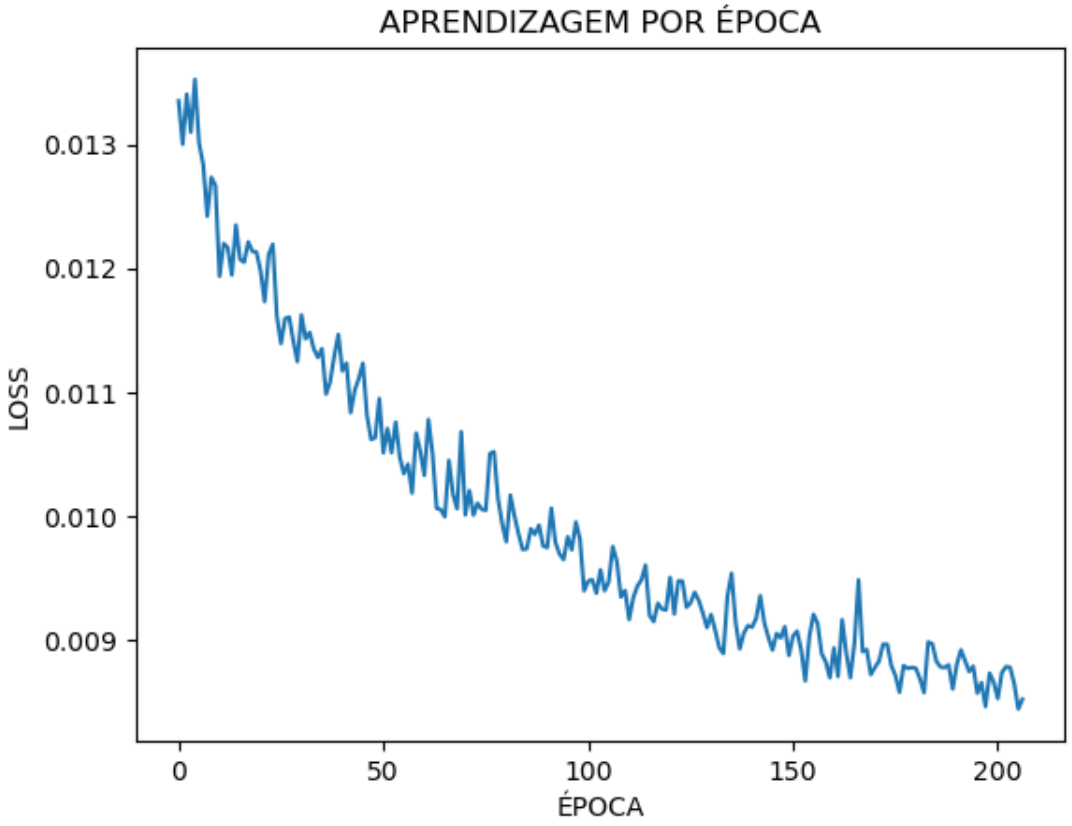
16. **Função de Perda**: A “lossfunc” define a função de perda como o erro médio quadrático (MSE), que será usada para calcular a diferença entre as previsões do modelo e os rótulos verdadeiros durante o treinamento.

17. **Preparação dos Dados para Treinamento e Teste**: Com as funções “train_dataset” e “test_dataset” com instâncias de “AvaliacoesDataset” que encapsulam os conjuntos de dados de treinamento e teste, respectivamente. E a “train_dataloader” e ‘test_dataloader” com o “DataLoader” sua função é para iterar eficientemente sobre os conjuntos de dados de treinamento e teste durante o treinamento e a avaliação do modelo. O conjunto de treinamento é dividido em lotes de tamanho 1000, enquanto o conjunto de teste é avaliado de forma individual (batch_size = 1).

18. **Treinamento do Modelo**: Na função “epochs” informa o número total de épocas de treinamento. E a função “train_results” é o dicionário para armazenar os resultados do treinamento, incluindo o número de épocas e a perda durante o treinamento.
    O loop principal percorre cada época de treinamento:
    - Itera sobre os lotes de dados de treinamento usando a função “train_dataloader”.
    - Realiza a passagem direta (forward pass) através do modelo.
    - Calcula a perda com base na saída do modelo e nos rótulos verdadeiros.
    - Executa a retro propagação (backpropagation) e o ajuste dos pesos do modelo.
    - Calcula a perda média para a época atual e a armazena em “train_results”.
    - A cada 5 épocas, imprime o número da época, a perda média de treinamento e a precisão atual do modelo.
    - Salva um checkpoint do modelo a cada 5 épocas.

19. **Visualização da Evolução da Aprendizagem**: Após o treinamento, os resultados de treinamento são convertidos em um DataFrame do pandas. Plotamos um gráfico de linha para visualizar a evolução da perda ao longo das épocas de treinamento. Isso ajuda a entender como a perda diminui à medida que o modelo é treinado, conforme pode ser analisado na figura 4.

Figura 4: Evolução da Aprendizagem




Fonte: Google Colab – Elaborado pelo autor, 2024.

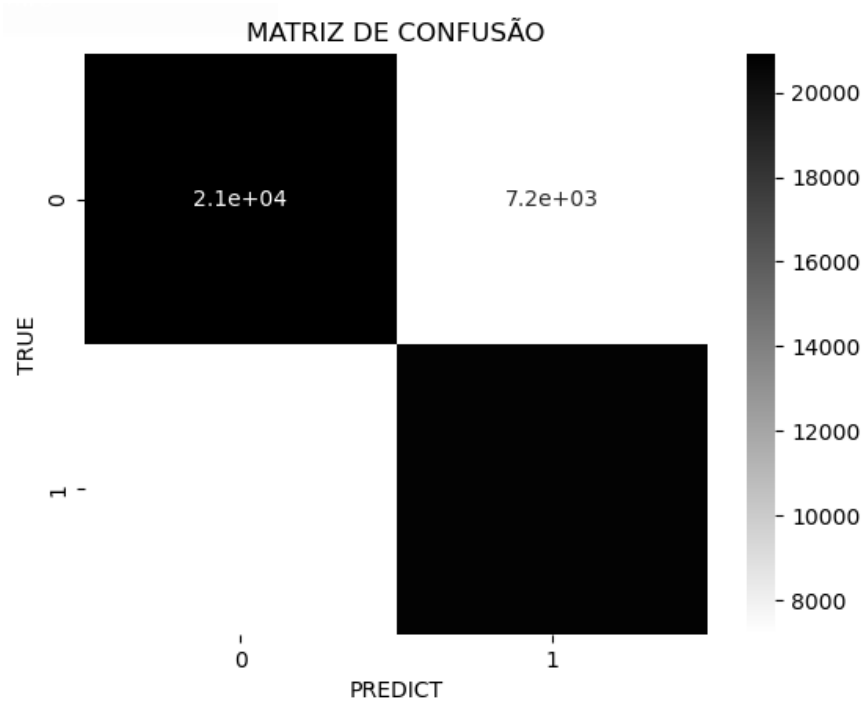
20. **Avaliação do Modelo**: A função “model.eval()” coloca o modelo em modo de avaliação, desativando a camada de *dropout* ou *batch normalization*, se aplicável. O comando “test_loss” é a variável para armazenar a perda média durante a avaliação do modelo. O comando “final” é o dicionário para armazenar os resultados finais da avaliação, incluindo as características do usuário e do livro, a classificação predita e a classificação verdadeira.

    Um loop é executado sobre os dados de teste usando o “test_dataloader”:

    - Realiza a passagem direta (forward pass) através do modelo para obter as previsões.
    - Calcula a perda com base nas previsões e nos rótulos verdadeiros.
    - Armazena as características do usuário e do livro, bem como as classificações preditas e verdadeiras.
    - Calcula a precisão do modelo com base nas previsões e nos rótulos verdadeiros.
    - Os resultados finais são convertidos em um DataFrame do pandas e são impressos o valor da perda de teste e a precisão do modelo.

21. **Matriz de Confusão**: A partir dos resultados finais, é calculada a matriz de confusão para avaliar o desempenho do modelo. A matriz de confusão é impressa e plotada como um mapa de calor, onde as células mostram o número de previsões corretas e incorretas para cada classe. Isso ajuda a visualizar onde o modelo está acertando e errando, conforme é demonstrado na figura 5.

Figura 5: Resultado da Matriz de Confusão




Fonte: Google Colab – Elaborado pelo autor, 2024.

22. **Seleção do Usuário para Recomendação:** Nesta etapa é realizado a implementação do modelo para um determinado usuário, unir dados do User-ID com toda a base de livros e prever quais livros tem maior probabilidade de ter nota 9 ou 10 (High). Com a entrada “user_id” identificamos um usuário para o qual as recomendações serão geradas e também verifica se o “user_id” está presente na base de dados de avaliações.

23. **Filtragem do Usuário na Base de Dados**: Com o comando “user_filter” este extrai os dados do usuário da base de dados de usuários.

24. **Preparação dos Dados para Recomendação**: Na função “predict_data” esta seleciona os dados dos livros que foram treinados pelo modelo. Inclui os dados do usuário em todas as linhas do DataFrame de predição e em seguida remove os livros que o usuário já avaliou da lista de recomendações.

25. **Codificação dos Dados de Recomendação:** No comando “predict_encoded” este codifica os dados dos livros para alimentar o modelo.

26. **Criação do Dataset e DataLoader para Recomendação:** Com as funções: “predict_dataset” é instância do conjunto de dados para predição e a “predict_dataloader” é o DataLoader para iterar sobre os dados de predição.

27. **Geração das Recomendações pelo Modelo**: Nesta etapa se faz a passagem direta pelo modelo para obter as previsões de classificação alta para cada livro. Ordena os resultados pela probabilidade de ser um livro com alta classificação e seleciona os top “n_recomendacoes” livros em seguida decodifica os dados de recomendação para torná-los compreensíveis.

28. **Exibição das Recomendações:** Para cada livro recomendado, exibe o título, ISBN e a capa do livro em HTML, conforme é demonstrado na figura 6.

Para o teste realizado com o usuário de user_id “8680” foram apresentados os 5 livros:

1 - Titulo: A Guided Tour of Rene Descartes' Meditations on First Philosophy with Complete Translations of the Meditations by Ronald Rubin

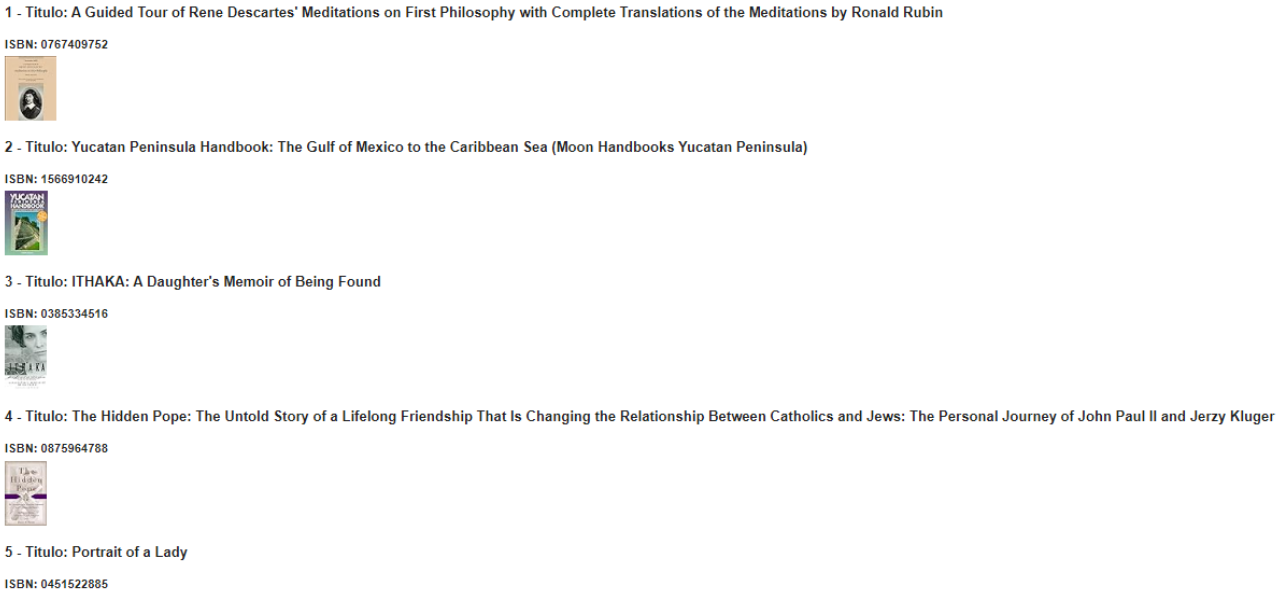
2 - Titulo: Yucatan Peninsula Handbook: The Gulf of Mexico to the Caribbean Sea (Moon Handbooks Yucatan Peninsula)

3 - Titulo: ITHAKA: A Daughter's Memoir of Being Found

4 - Titulo: The Hidden Pope: The Untold Story of a Lifelong Friendship That Is Changing the Relationship Between Catholics and Jews: The Personal Journey of John Paul II and Jerzy Kluger

5 - Titulo: Portrait of a Lady

Figura 6: Livros Recomendados




Fonte: Google Colab – Elaborado pelo autor, 2024.

A demonstração sintetizada do pipeline desta implementação resume com as implementações das etapas apresentadas na figura 7.

Figura 7: Pipeline Sistema de Recomendação


Elaborado pelo autor, 2024.

A implementação da versão 2 do algoritmo de recomendação, com todas as etapas descritas neste documento está disponível em nosso diretório do GitHub, no link: https://github.com/OtavioBer/ProjAplicadoIII/blob/OtavioBer-patch-1/Modelo_Recomendacao_V3.ipynb

# 12. RESULTADOS

Foi avaliado o resultado do test loss de 0.2416 e uma acurácia de 0.7431 indicam o desempenho do modelo de recomendação nos dados de teste.

- **Test Loss (Perda de Teste):** Um valor de perda de teste de 0.2416 indica a média das diferenças entre as classificações previstas pelo modelo e as classificações reais nos dados de teste. Quanto menor o valor de perda, melhor o desempenho do modelo.

- **Acurácia:** Uma acurácia de 0.7431 significa que o modelo classificou corretamente aproximadamente **74,31%** das recomendações nos dados de teste. Significa que das recomendações feitas pelo modelo, cerca de 74,31% delas estão corretas.

A interpretação deste resultado resume que o modelo está realizando recomendações mais precisas e classificando corretamente uma maior proporção das recomendações nos dados de teste, com uma acurácia significativa. Uma acurácia de cerca de 74,31% indica que o modelo está correto na maioria das recomendações. Isso sugere que o modelo implementado está sendo eficaz na recomendação de livros ao usuário. No entanto, na prática, é importante considerar que os modelos de recomendação podem enfrentar desafios devido à natureza subjetiva das preferências dos usuários e à complexidade dos dados. Portanto, alcançar um resultado perfeito pode ser difícil na prática, por não ter dados mais consistentes e um grande volume de avaliações para um maior acervo de livros.

Uma base de dados com um acervo maior de livros e um maior número de avaliações pode ajudar na melhoria do resultado de um modelo de recomendação de livros de várias maneiras:

- **Melhoria da Diversidade de Livros**: Uma base de dados com um acervo maior de livros pode aumentar a diversidade de títulos disponíveis para recomendação. Isso permite que o modelo tenha mais opções para escolher ao fazer recomendações, o que pode levar a sugestões mais personalizadas e relevantes para os usuários.

- **Melhoria da Precisão das Recomendações**: Com um maior número de avaliações disponíveis, o modelo pode aprender com mais exemplos e padrões nos dados. Isso pode resultar em previsões mais precisas e confiáveis sobre quais livros um usuário pode gostar, levando a uma melhor experiência de recomendação.

- **Redução do Desvio do Modelo:** Um maior número de avaliações pode ajudar a reduzir o desvio do modelo, permitindo que ele capture melhor a variabilidade nas preferências

dos usuários. Isso pode resultar em recomendações mais precisas e menos propensas a serem influenciadas por tendências ou comportamentos individuais.

- **Aumento da Robustez do Modelo**: Uma base de dados maior e mais diversificada pode ajudar a aumentar a robustez do modelo, tornando-o mais capaz de lidar com diferentes tipos de usuários e cenários de recomendação. Isso pode levar a um desempenho mais consistente do modelo em uma variedade de situações.

Em resumo, uma base de dados com um acervo maior de livros e um maior número de avaliações pode proporcionar uma fonte de dados mais rica e diversificada para treinar modelos de recomendação de livros, o que pode resultar em recomendações mais precisas, personalizadas e relevantes para os usuários.

# 13. CONCLUSÃO

## 14. DIRETÓRIO GITHUB

Todo o conteúdo do projeto estará disponível no site da GitHub, que poderá ser acessado pelo link:

https://github.com/OtavioBer/ProjAplicadoIII

O diretório está organizado por pastas:

Pasta “Códigos” será disponibilizado os códigos em *Python* que foram utilizados para realizar a análise exploratória, tratamento dos dados e o sistema de recomendação.

Pasta “Dados” temos os arquivos utilizados para o estudo.

Pasta “Documentos” temos o cronograma de entrega do projeto e as versões de entrega deste documento.

Temos também o arquivo README.md com informações relevantes do projeto.

# 15. REFERENCIAL TEÓRICO

## 15.1. Sites de pesquisa:


https://medium.com/tech-grupozap/sistemas-de-recomenda%C3%A7%C3%A3o-5bd1626326fe

https://www.sciencedirect.com/topics/chemical-engineering/deep-neural-network

https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-recommendation-engine-*Python*/

https://learn.microsoft.com/pt-br/azure/architecture/solution-ideas/articles/build-content-based-recommendation-system-using-recommender

https://ieducacao.ceie-br.org/sistemas-recomendacao/

https://www.voitto.com.br/blog/artigo/biblioteca-pandas

https://medium.com/ensina-ai/entendendo-a-biblioteca-numpy-4858fde63355

https://www.crawly.com.br/blog/*Python*-e-big-data-fique-por-dentro-de-3-bibliotecas-essenciais

https://dadosaocubo.com/analise-de-dados-com-seaborn-*Python*/

https://surpriselib.com/

https://awari.com.br/scikit-learn/

https://pt.wikipedia.org/wiki/Scikit-learn

https://aws.amazon.com/pt/what-is/neural-network/#:~:text=As%20redes%20neurais%20profundas%2C%20ou,um%20n%C3%B3%20reprimir%20o%20outro.

https://en.wikipedia.org/wiki/Mean_squared_error

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html

https://scikit-learn.org/stable/modules/model_evaluation.html

https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

https://imbalanced-learn.org/stable/under_sampling.html

https://towardsdatascience.com/

https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/

https://www.kaggle.com/code/residentmario/undersampling-and-oversampling-imbalanced-data

https://acervolima.com/modulo-de-aprendizado-desequilibrado-em-*Python*/

https://pytorch.org/docs/stable/generated/torch.nn.Module.html

https://pytorch.org/docs/stable/generated/torch.nn.ReLU.html

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

https://pt.d2l.ai/chapter_optimization/adam.html

https://www.escoladnc.com.br/blog/avaliacao-de-metricas-em-sistemas-de-recomendacao-acuracia-e-cobertura/

https://diegonogare.net/2020/04/performance-de-machine-learning-matriz-de-confusao/#google_vignette
https://pt.wikipedia.org/wiki/Matriz_de_confus%C3%A3o

## 15.2. Livros da minha biblioteca

FERREIRA, Rogério; Aprendizado Profundo. Editora Saraiva, 2021.

NETTO, Amilcar; NETO, Francisco. *Python* Para Data Science e *Machine Learning* Descomplicado. Rio de Janeiro: Alta Books, 2021.

GRUS, Joel; Data Science do Zero. Editora Alta Books, 2021.

RICCI, F.; ROKACH, L.; SHAPIRA, B. Introdução ao Manual de Sistemas de Recomendação. Springer, 2015.

KOREN, Y.; BELL, R. Avanços na filtragem colaborativa. Springer, 2015.

Su, X.; KHOSHGOFTAAR, TM. Uma pesquisa de técnicas de filtragem colaborativa. Avanços em Inteligência Artificial, 2009.

HERLOCKER, JL; KONSTAN, JA; TERVEEN, LG; RIEDL, JT.Avaliando Sistemas de Recomendação de Filtragem Colaborativa. Transações ACM em Sistemas de Informação, 2004.

RESNICK, P.; VARIAN, HR. Sistemas de recomendação. Comunicações da ACM, 1997.

Material de apoio dos componentes curriculares: Aprendizado de Máquina, Análise Estatística Preditiva, Aquisição e Preparação de Dados