Projeto Aplicado III

Titulo:

SISTEMA DE RECOMENDAÇÃO DE LIVROS

Informações

Curso: Tecnologia em Ciências de Dados
Semestre: 4º
Componente curricular: Projeto Aplicado III
Professor: Thiago Donizetti dos Santos
Integrantes e TIA:

- **Caroline Ribeiro Ferreira – 10408052**
- **Lais César Fonseca – 10407066**
- **Liliane Gonçalves de Brito Ferraz – 10407087**
- **Leonardo dos Reis Olher – 10407752**
- **Múcio Emanuel Feitosa Ferraz Filho – 10218691**
- **Otavio Bernardo Scandiuzzi – 10407867**

Objetivos do Projeto

1

Desenvolver um modelo de **recomendação robusto usando DNNs** para capturar relacionamentos complexos entre usuários e livros.

2

Empregar a **métrica MSE** para avaliar o desempenho do modelo na previsão de avaliações de usuários para livros recomendados

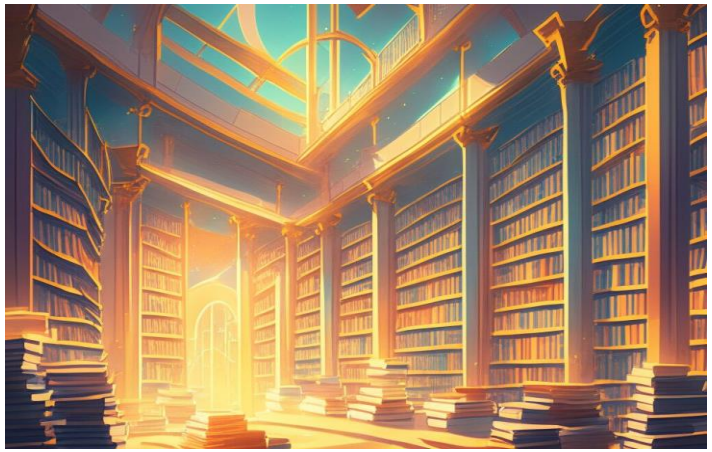
3

Implementar várias técnicas de avaliação, incluindo precisão e **matrizes de confusão**, para avaliar a efetividade do modelo

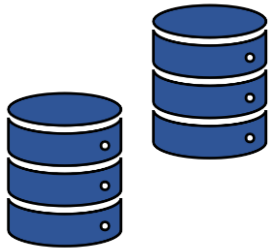
Dataset

Como Base de dados para desenvolvimento do projeto e treinamento dados Públicos, da plataforma **Kaggle**.

O Conjunto de dados escolhidos dispões de dados de qualidade e de relevância para o desenvolvimento de um sistema de recomendação preciso e confiável.



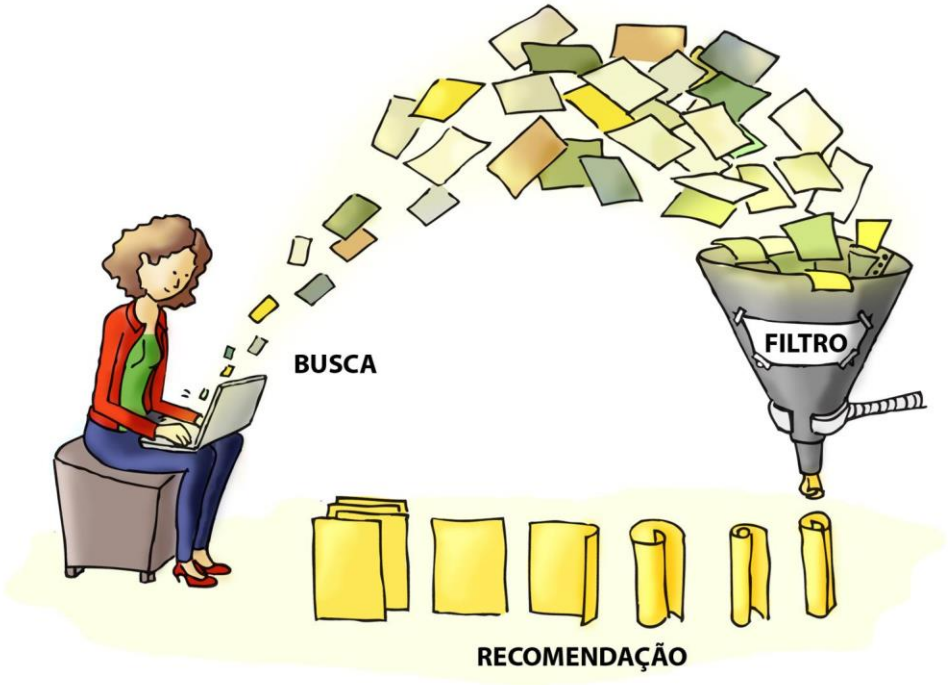
A base contém, 242.154 livros distintos, de 102.028 autores, dos anos de 1985 a 2004, contendo a avaliação de 278.858 usuários, de idades e localidades variadas.



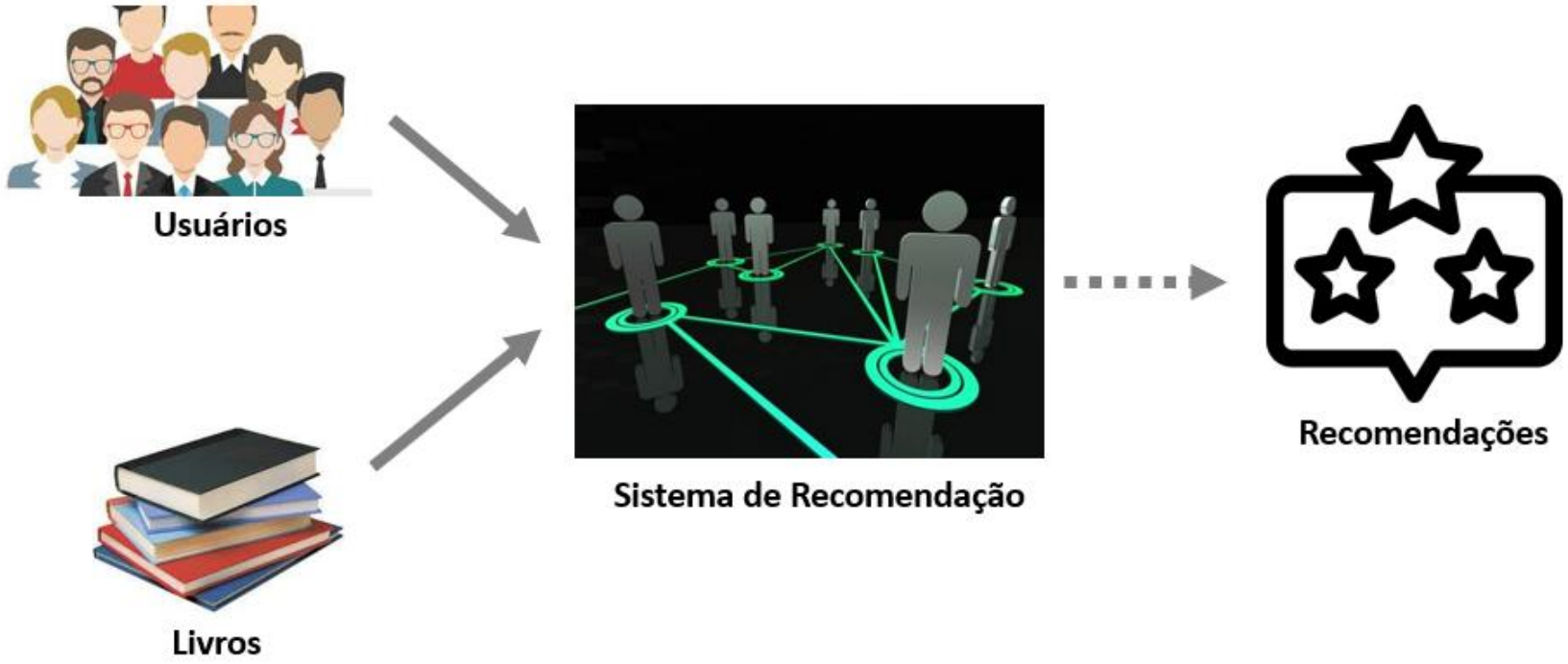
POR QUE UM SISTEMA DE RECOMENDAÇÃO DE LIVROS?

Na era digital de hoje, os sistemas de recomendação se tornaram onipresentes, desempenhando um papel crucial em diversas indústrias, incluindo e-commerce, serviços de streaming e plataformas de mídia social.

Os sistemas de recomendação de livros são particularmente valiosos, guiando os leitores para livros que se alinham com seus interesses e preferências, aprimorando sua experiência geral de leitura.



Afinal, como funciona?



Os dados coletados do dataset, são organizados, processados e analisados e com a implementação de um algoritmo de recomendação utilizando técnicas de aprendizado de máquina geram as recomendações de livros para o usuário de acordo com seu histórico de recomendações.

Análise Exploratória

ETAPAS ANÁLISE EXPLORATÓRIA

1. **Importar as bibliotecas** de necessário para realização de analise exploratória, gráficos e tratamento, sendo elas: Pandas, Numpy, Seaborn, Matplotlib.pyplot e Matplotlib.ticker.

2. **Importar os três conjuntos de dados**: informações sobre livros, usuários e avaliações.

3. Os conjuntos de **dados foram unificados,** com base nos identificadores únicos de usuários e livros.

4. Realizou-se a **análise das dimensões** dos dataframes, identificação dos atributos e tipos de dados.

5. Foram verificadas as **distribuições de dados**, identificando a quantidade de usuários, livros e avaliações disponíveis.

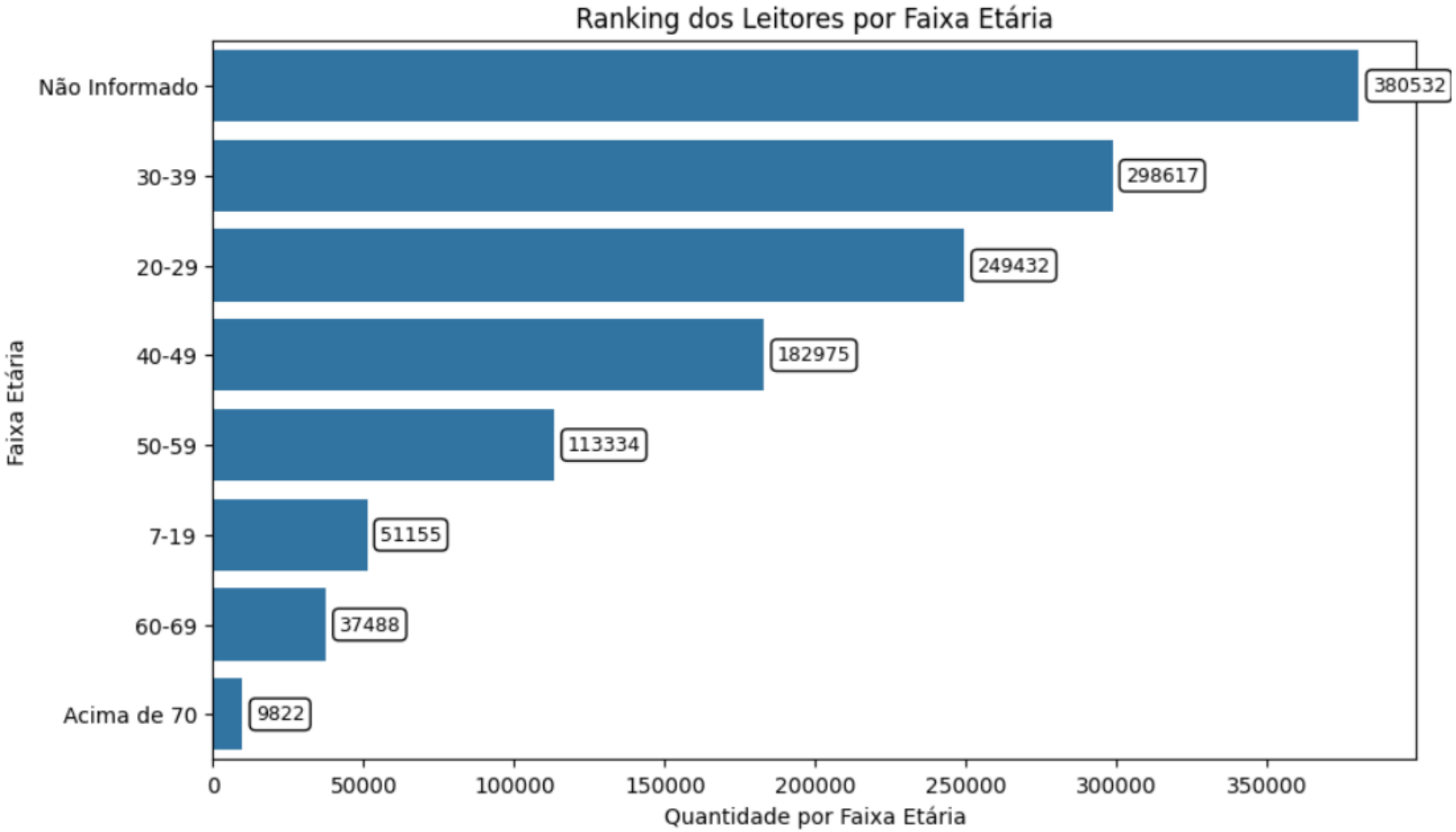
6. Foi realizado o **tratamento de valores ausentes**, removendo as observações com informações faltantes.

7. Calculou-se a **distribuição dos usuários por faixa etária e** a distribuição da quantidade de leitores por país.

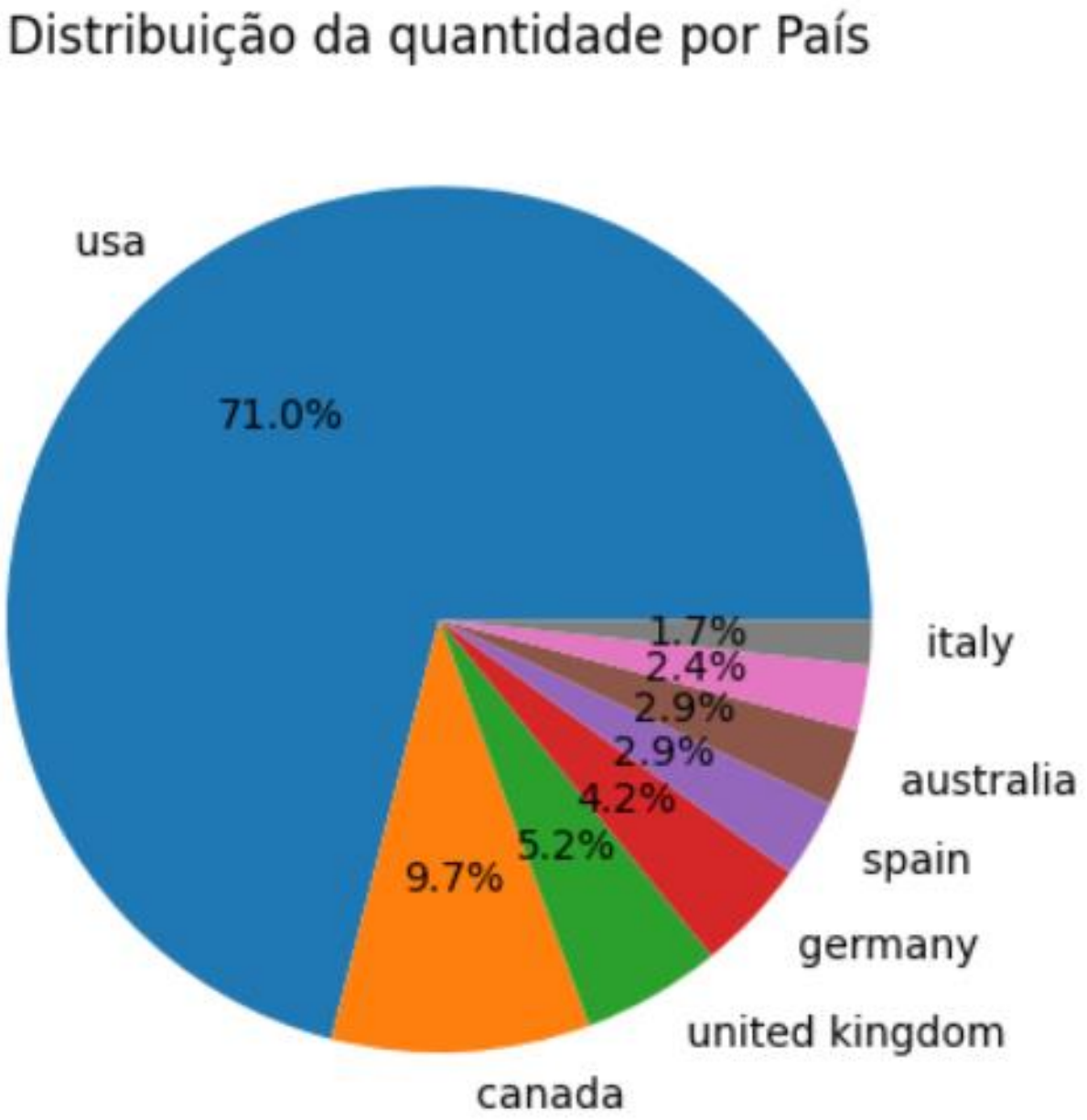
8. **Exploramos a contagem de livros por ano** de publicação, editora e autor, identificando os mais populares.

9. **Apresentamos as avaliações dos usuários por livro**, identificando os mais avaliados.

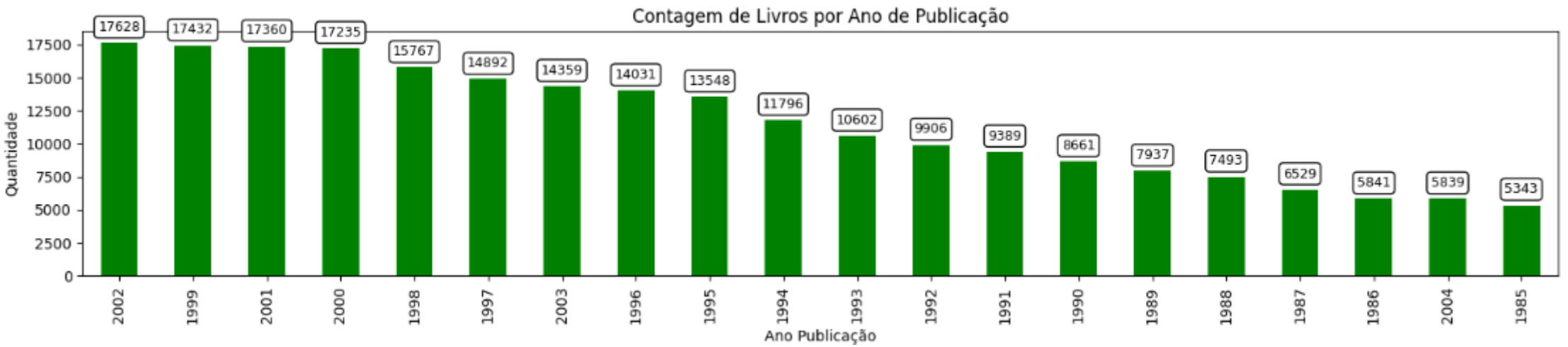
ALGUNS GRÁFICOS APRESENTADOS

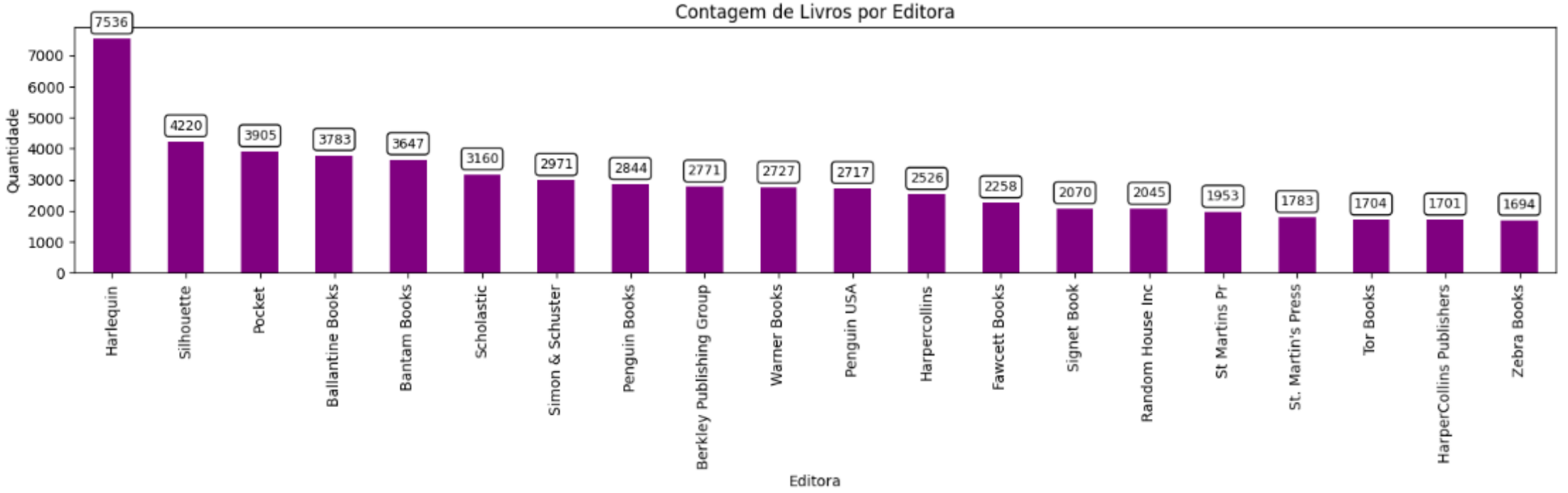


ALGUNS GRÁFICOS APRESENTADOS



ALGUNS GRÁFICOS APRESENTADOS





Limpeza e Preparação dos Dados

LIMPEZA E PREPARAÇÃO DOS DADOS

1. Realizou-se o tratamento dos dados removendo as observações com valores nulos.

2. Selecionou-se apenas os usuários que fizeram mais de 50 avaliações para garantir uma base de dados mais robusta.

3. Após o tratamento, o conjunto de dados final possui 765.672 registros e 8 atributos.

4. Foi feito o balanceamento dos dados para equilibrá-los no modelo de aprendizado de máquina

CÓDIGOS UTILIZADOS

```
# SEPARAR PAIS DO LOCATION
users['Country'] = users['Location'].apply(lambda x: x.split(', ')[-1])
# users = users.sample(10000)

# UNIR BASES
df = ratings\
    .merge(users, how='inner', on='User-ID')\
    .merge(books, how='inner', on='ISBN')

# LIMPEZA DE DADOS
df = df\
    .dropna()\
    .drop_duplicates()\
    .drop(['Location','Image-URL-M','Image-URL-L'], axis=1)

# FILTRAR COLUNAS UTILIZADAS NO MODELO
cols = ['User-ID','ISBN','Book-Author','Publisher','Country','Year-Of-Publication','Age','Book-Rating']
avaliacoes = df[cols]
avaliacoes.reset_index(drop=True, inplace=True)
```

CÓDIGOS UTILIZADOS

```
# CLASSIFICAÇÕES
avaliacoes['HIGH_RATING'] = (avaliacoes['Book-Rating'] >= 8).astype(int)

# RETIRAR RATING
avaliacoes.drop('Book-Rating', axis=1, inplace=True)

# RETIRAR USUARIOS QUE AVALIARAM MENOS QUE 2 LIVROS COM 8,9,10
count_avaliacoes_high = avaliacoes.groupby('User-ID', as_index=False).agg({'HIGH_RATING':sum})
count_avaliacoes_high = count_avaliacoes_high[count_avaliacoes_high['HIGH_RATING'] > 2]
count_avaliacoes_high = count_avaliacoes_high['User-ID'].tolist()
avaliacoes = avaliacoes[avaliacoes['User-ID'].isin(count_avaliacoes_high)]

# CODIFICAR DADOS
enc, df_enc = encoder_df(avaliacoes)
```

CÓDIGOS UTILIZADOS

Balanceamento dos dados

- Undersampling

```
print('BASE DESBALANCEADA:\n')
print(avaliacoes['HIGH_RATING'].value_counts(), end='\n\n\n\n')

# REMOVER PARTE DO RATING MAJORITARIO
x, y = RandomUnderSampler(sampling_strategy='majority').fit_resample(df_enc.drop('HIGH_RATING', axis=1), y=df_enc['HIGH_RATI
x['HIGH_RATING'] = y

print('BASE BALANCEADA:\n')
print(x['HIGH_RATING'].value_counts())

# SPLIT TREINO TESTE
train, test = train_test_split(x, test_size=.2, shuffle=True)
```

```
BASE DESBALANCEADA:

HIGH_RATING
0    486218
1    140091
Name: count, dtype: int64
```

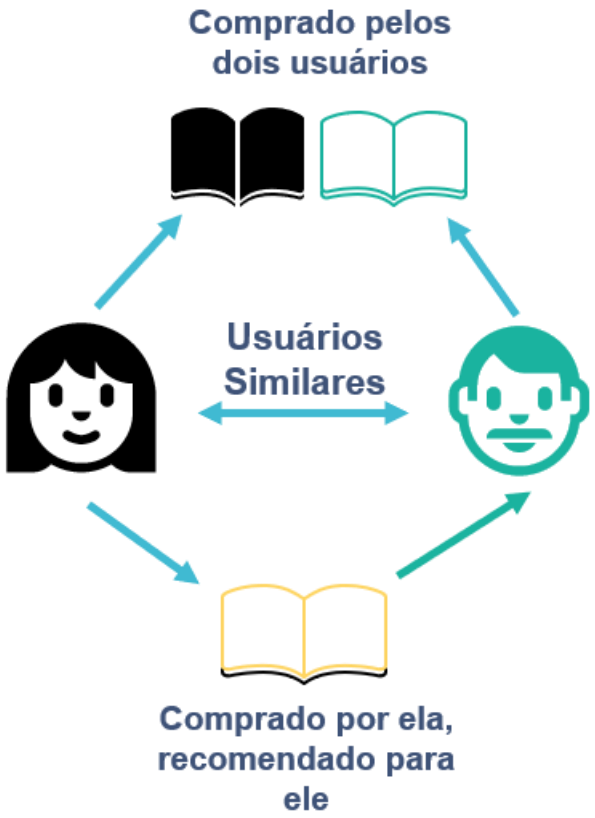
```
BASE BALANCEADA:

HIGH_RATING
0    140091
1    140091
Name: count, dtype: int64
```

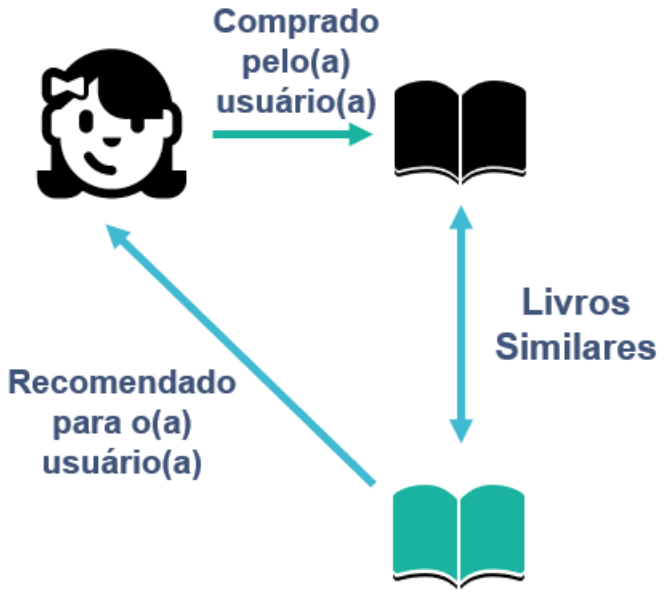
Aprendizado de Máquina

TÉCNICAS APRENDIZADO DE MÁQUINA

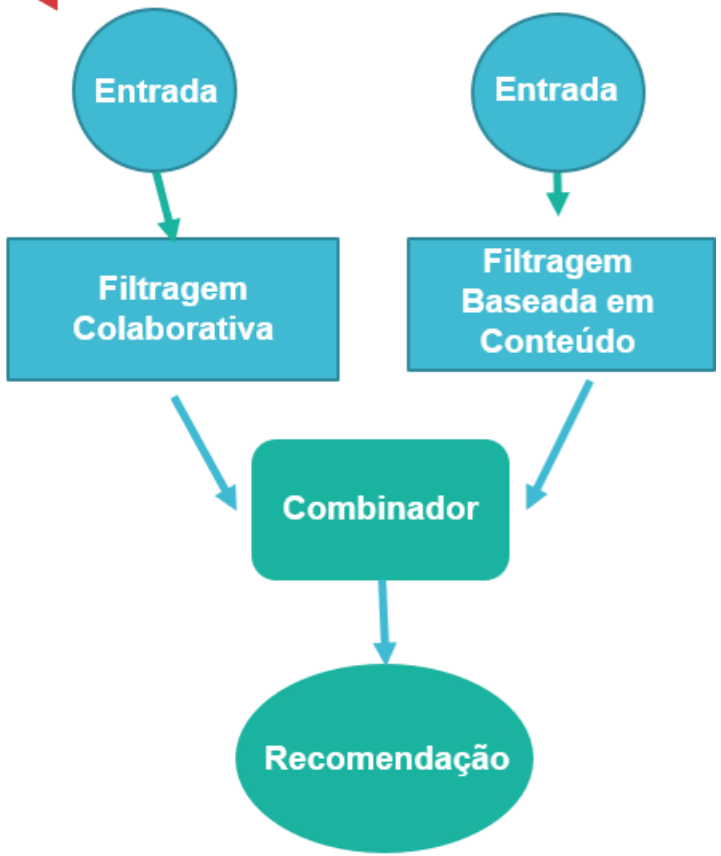
Filtragem Colaborativa



Filtragem Baseada em Conteúdo



Filtragem Híbrida



Modelo de recomendação

Arquitetura da Rede Neural Profunda (DNN)

Camadas de Incorporação: Representam usuários e livros como vetores em um espaço de características latente, capturando suas características únicas.

Camadas Ocultas: Extraem relacionamentos e padrões complexos entre usuários e livros usando várias camadas de neurônios.

Camada de Saída: Prediz as avaliações dos usuários para livros recomendados.

PROCESSO DE TREINAMENTO

Pré-processamento de Dados: Limpa, transforma e prepara os dados para o treinamento do modelo.

Treinamento do Modelo: Otimiza os parâmetros do modelo usando um otimizador e função de perda apropriados (MSE).

Avaliação: Avalia o desempenho do modelo em dados não vistos usando métricas como precisão e matrizes de confusão

Teste e Acurácia

TESTE E ACURÁCIA

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.

MÉTRICAS DE AVALIAÇÃO

➢ **Erro Médio Quadrático (MSE):**

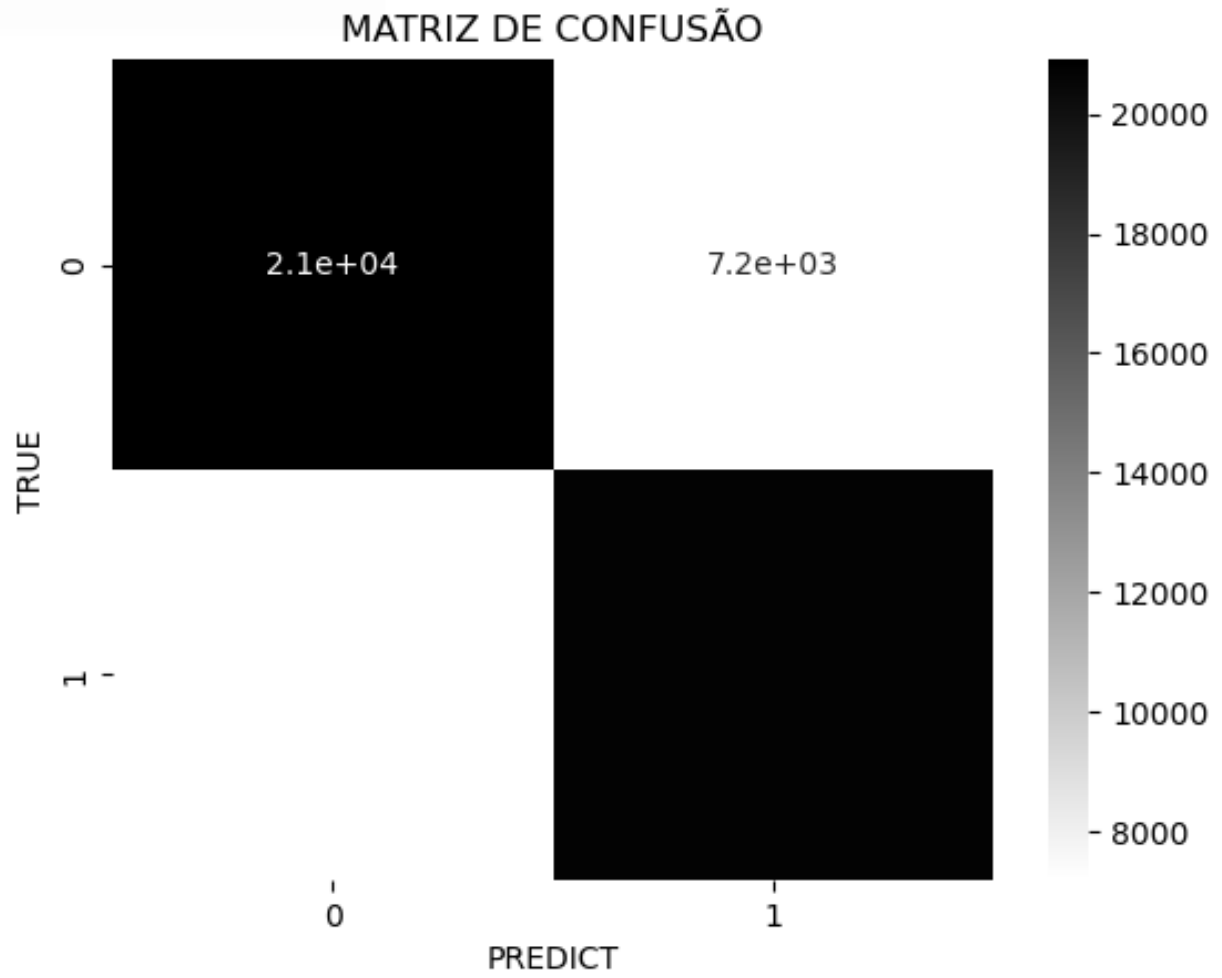
- Quantifica a diferença média quadrática entre as avaliações previstas e as avaliações reais.
- Um MSE menor indica um melhor desempenho do modelo na previsão das preferências do usuário.

➢ **Acurácia:**

- Mede a proporção de recomendações corretas feitas pelo modelo.
- Uma precisão maior sugere que o modelo identifica efetivamente livros que os usuários provavelmente apreciarão.

MATRIZ DE CONFUSÃO

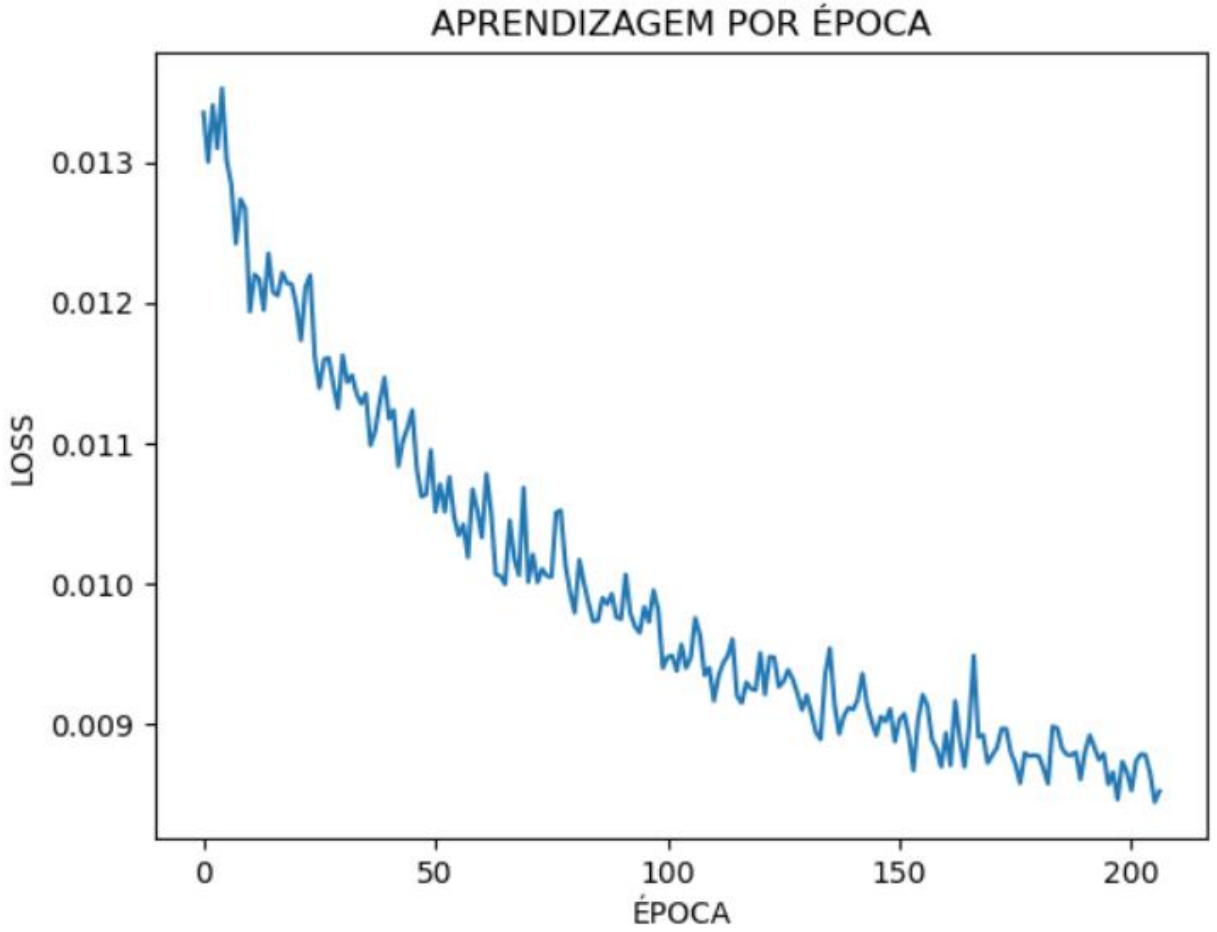
- Visualiza a distribuição de recomendações corretas e incorretas em diferentes categorias.
- Fornece insights sobre os pontos fortes e fracos do modelo na realização de recomendações.



AVALIAÇÃO DO MODELO

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.



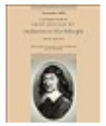
Resultados


affairs
Skewness: 2.34 Kurtosis: 4.21
yearsmarried
Skewness: 0.08 Kurtosis: -1.57
age
Skewness: 0.89 Kurtosis: 0.22
religiousness
Skewness: -0.09 Kurtosis: -1.01
page 2 of 2
religiousness vs daffairs
rating vs daffairs
Target
Percentage (%)
rating
occupation 0.0174
education 0.5336 0.1093
0.0426 -0.0397 0.0243
0.0446 -0.2431
0.1664 -0.199
affairs 0.0952
cluster
fact_1
fact_2
0.25
n=484 100%
rating >= -1.3
0.2
n=414 86%
religiousness >= 0.33
0.26
n=224 46%
rating >= 0.52
0.16
n=103 21%
0.36
n=121 25%
0.5
n=70 14%

RECOMENDAÇÕES SUGERIDAS

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

ISBN: 0767409752



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

ISBN: 1566910242

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

ISBN: 0385334516

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

ISBN: 0875964788

5 - Titulo: Portrait of a Lady

ISBN: 0451522885

PIPELINE

DADOS

Importação de bibliotecas e pacotes, carregamento dos dados e a preparação.

CODIFICAÇÃO

Funções para codificar e decodificar os dados, utilizando LabelEncoder do sklearn.

BALANCEAMENTO

É aplicada uma técnica de balanceamento de dados para lidar com classes desbalanceadas.

MODELO

É definida a arquitetura do modelo de recomendação usando PyTorch, incluindo as camadas de embedding e uma série de camadas lineares.

TREINAMENTO

O modelo é treinado usando os dados preparados e uma função de perda definida.

AVALIAÇÃO

O modelo treinado é avaliado usando o conjunto de teste e são calculadas métricas de desempenho, incluindo a matriz de confusão.

RECOMENDAÇÃO

Com o modelo treinado, recomendações são geradas para um usuário específico com base nos livros que ele ainda não avaliou.

CONCLUSÃO

- ✓ Os resultados do teste mostraram um erro **médio quadrático (MSE) de 0,2416** e uma **precisão de 74.31%**, indicando um desempenho satisfatório do modelo em prever preferências de usuários.
- ✓ A implantação de camadas de incorporação permitiu representar usuários e livros em um espaço de características latentes, capturando suas particularidades de maneira eficiente.
- ✓ O modelo empregou **técnicas de Redes Neurais Profundas** (DNNs) e foi treinado utilizando um conjunto de dados robusto com informações detalhadas sobre livros, autores e avaliações de usuários.
- ✓ Os resultados do teste indicaram um **desempenho satisfatório** do modelo em prever preferências de usuários.
- ✓ Portanto, nosso sistema de recomendação apresenta-se como uma ferramenta valiosa, **pois atende às necessidades específicas para cada usuário** e contribuindo para a descoberta e engajamento dentro da plataforma.

Obrigado(a)!