Universidade Presbiteriana Mackenzie

Tecnologia em Ciência de Dados - Projeto Aplicado 3

Matheus Vinícius Gomes

Leandro Rodrigues Dos Santos

Luiz Eduardo de Mei Salvador Coelho

Matheus Neves de Castro

Projeto Aplicado III

Sistema de Recomendação de Filmes para Plataforma de Streaming

SÃO PAULO

2º Semestre de 2024

Autores:

Matheus Vinícius Gomes / 10408179 / mathvgomes@gmail.com

Leandro Rodrigues Dos Santos / 23019689 / lers.138@gmail.com

Luiz Eduardo de Mei Salvador Coelho / 23024585 / Luiz02coelho@gmail.com

Matheus Neves de Castro / 10415190 / 10415190@mackenzista.com.br

Projeto Aplicado III

Sistema de Recomendação de Filmes para Plataforma de Streaming

Professora: CAROLINA TOLEDO FERRAZ

SÃO PAULO

2º Semestre de 2024

Resumo (deverá ser elaborado apenas no final do projeto.)

Sumário

Introdução

Com o crescimento vertiginoso das plataformas de streaming, a experiência do usuário se torna cada vez mais desafiadora à medida que a biblioteca de conteúdo se expande continuamente. Os usuários enfrentam uma diversidade de gêneros, estilos e diretores, que, embora enriquecedora, pode levar à **sobrecarga de opções**. Esse fenômeno, conhecido como **paradoxo da escolha**, muitas vezes dificulta que os usuários encontrem conteúdos que realmente correspondam aos seus gostos e preferências. Nesse contexto, os **sistemas de recomendação de filmes** surgem como uma solução fundamental, direcionando o usuário de maneira prática e personalizada para os títulos que mais se adequam ao seu perfil.

Um sistema de recomendação bem estruturado pode transformar a experiência do usuário, facilitando a descoberta de novos conteúdos e promovendo o engajamento. Ao fornecer sugestões baseadas em preferências e comportamentos anteriores, esses sistemas buscam não apenas atender às expectativas imediatas do usuário, mas também encorajá-los a explorar filmes que poderiam, de outra forma, passar despercebidos. Em um mercado altamente competitivo, onde a retenção de usuários é essencial para o sucesso, esses sistemas se tornam um diferencial estratégico para as plataformas de streaming, **aumentando tanto a satisfação quanto a lealdade dos usuários.**

Este projeto, portanto, visa desenvolver um sistema de recomendação de filmes que ofereça sugestões personalizadas com base no comportamento e nas preferências dos usuários. A implementação de um sistema desse tipo requer a utilização de técnicas de **Machine Learning** e a análise profunda dos dados de consumo, para garantir que as recomendações sejam precisas e relevantes.
Para alcançar esse objetivo, estabelecemos os seguintes objetivos específicos:

1. **Analisar e explorar dados de consumo de filmes**: Examinar dados de interações dos usuários com o catálogo de filmes, realizando uma análise exploratória dos padrões de consumo.
2. **Preparar e limpar os dados**: Implementar técnicas de pré-processamento e limpeza dos dados, assegurando uma base confiável e estruturada para a modelagem.
3. **Selecionar e aplicar técnicas de Machine Learning**: Escolher algoritmos adequados para a criação de um modelo de recomendação eficaz e personalizado.
4. **Avaliar o desempenho do modelo**: Definir e aplicar métricas para medir a precisão e a relevância das recomendações, refinando o modelo conforme necessário.

5. **Documentar e apresentar os resultados**: Registrar todas as etapas e os resultados do projeto, ressaltando os impactos positivos na experiência do usuário e as possibilidades de melhorias futuras.

Dessa forma, o presente trabalho busca não apenas otimizar o processo de recomendação de filmes, mas também contribuir para uma experiência de uso mais agradável, estimulando a **exploração contínua de novos conteúdos** e ampliando o valor percebido pelo usuário nas plataformas de streaming.

Referencial Teórico

O aumento no uso de plataformas de streaming trouxe à tona a importância de sistemas de recomendação personalizados, que não apenas facilitam a escolha de conteúdo, mas também aprimoram a experiência do usuário ao sugerir filmes que correspondam aos seus interesses. Diversos estudos científicos têm investigado formas eficazes de desenvolver e aprimorar esses sistemas. Nesta seção, abordaremos os principais trabalhos relacionados ao tema e os desafios que ainda existem na literatura sobre recomendações de filmes.

Abordagens em Sistemas de Recomendação

Os sistemas de recomendação de filmes têm sido amplamente estudados, com diferentes abordagens propostas para otimizar a personalização e precisão. Filtragem colaborativa, baseada em interações dos usuários, é uma técnica comum em sistemas de recomendação (Su et al., 2020), utilizando métodos como decomposição em valores singulares (SVD) e Redes Neurais, que correlacionam usuários com preferências similares. Embora amplamente adotada, a filtragem colaborativa sofre de problemas de sparsity (sparsidade), onde a ausência de avaliações suficientes para certos filmes ou novos usuários limita a qualidade das recomendações (Schafer et al., 2007).

A filtragem baseada em conteúdo é outra técnica frequentemente utilizada, que recomenda filmes com características semelhantes aos que o usuário já avaliou positivamente (Pazzani & Billsus, 2007). No entanto, este método enfrenta o desafio da dependência de dados: é necessário que as características do filme (gênero, diretor, elenco) estejam bem definidas, o que nem sempre é possível ou suficiente para capturar nuances de preferências individuais.

Recentemente, os modelos híbridos têm ganhado destaque ao combinar filtragem colaborativa e baseada em conteúdo para superar as limitações de ambas as abordagens. Burke (2002) demonstrou que esses modelos conseguem reduzir a sparsidade e melhorar a precisão da recomendação. Contudo, desafios como a complexidade computacional e a necessidade de calibragem entre diferentes métodos ainda persistem.

Desafios no Desenvolvimento de Sistemas de Recomendação de Filmes

Apesar do progresso nas abordagens de recomendação, alguns desafios críticos ainda permanecem. Um dos principais desafios é o problema do cold start para novos usuários e filmes, que ocorre quando o sistema não possui informações suficientes para fazer recomendações iniciais eficazes (Park et al., 2021). Várias técnicas têm sido exploradas para mitigar esse problema, como o uso de aprendizado de transferência e dados externos (Camacho & Alves-Souza, 2018), mas ainda não há uma solução universalmente eficaz.

Outro desafio significativo é a explicabilidade das recomendações. Como muitos sistemas de recomendação de filmes utilizam algoritmos complexos, os resultados podem parecer uma "caixa-preta" para os usuários. Trabalhos de Tintarev e Masthoff (2011) exploram formas de tornar os sistemas de recomendação mais transparentes e justificáveis, porém, em alguns casos, a explicação pode prejudicar a simplicidade e a eficiência do modelo.

Além disso, há questões de equidade e diversidade nas recomendações. Estudos mostram que recomendações tendem a reforçar padrões existentes, recomendando frequentemente filmes populares e ignorando opções menos conhecidas (Jannach et al., 2015). Modelos que priorizam filmes amplamente avaliados tendem a obscurecer conteúdos de nicho, o que representa um obstáculo tanto para a diversidade das recomendações quanto para a satisfação do usuário.

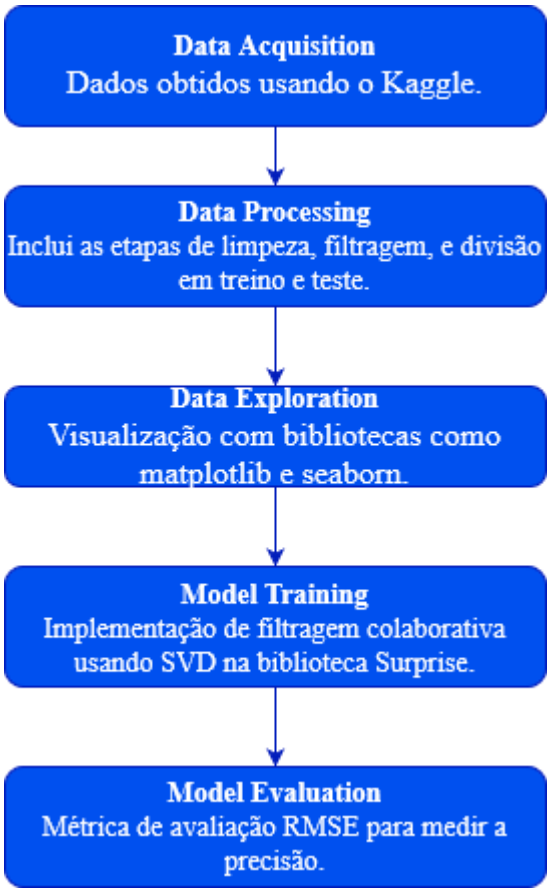
Perspectivas Futuras

O campo dos sistemas de recomendação de filmes segue evoluindo, especialmente com o uso de modelos baseados em Deep Learning e aprendizado de representação. Esses modelos conseguem capturar padrões mais complexos de preferência, mas ainda apresentam limitações, como a necessidade de grandes volumes de dados rotulados e o alto custo computacional (Zhang et al., 2019).

Em síntese, enquanto os sistemas de recomendação têm avançado significativamente, muitos desafios permanecem, principalmente em relação à complexidade dos algoritmos, à personalização para novos usuários e à equidade nas recomendações. Estudos futuros poderão focar na criação de modelos mais explicáveis e adaptáveis, capazes de combinar precisão e diversidade em recomendações mais satisfatórias e inclusivas.

Metodologia

O desenvolvimento de um sistema de recomendação eficiente exige uma abordagem estruturada, desde a aquisição e tratamento de dados até a modelagem e avaliação do modelo. Este projeto utiliza o **Netflix Prize Dataset**, disponível publicamente no Kaggle, como fonte primária de dados, contendo informações sobre filmes e avaliações de usuários. Abaixo, detalhamos as fases da metodologia aplicada neste projeto.



A metodologia adotada neste projeto compreendeu diversas etapas estruturadas, desde a aquisição de dados até a avaliação do modelo. Inicialmente, os dados foram adquiridos a partir do Netflix Prize Dataset, seguido de um rigoroso processo de limpeza e preparação. A separação dos dados em conjuntos de treinamento e teste foi realizada utilizando o arquivo "probe", assegurando que o modelo pudesse ser validado em condições realistas. Após a modelagem utilizando o SVD, a precisão do modelo foi avaliada com base no RMSE. Com os resultados preliminares, foram realizados ajustes no pipeline, e uma nova rodada de avaliação demonstrou melhorias significativas no desempenho.

O modelo de recomendação foi desenvolvido utilizando a técnica de filtragem colaborativa baseada em SVD, que permite identificar relações ocultas entre usuários e filmes. Essa abordagem se destaca pela sua capacidade de lidar com grandes conjuntos de dados e pela eficácia em ambientes esparsos. A implementação do algoritmo foi realizada com a biblioteca Surprise, que oferece suporte a otimizações específicas para o SVD. Além disso, a integração de um sistema híbrido, que combina filtragem colaborativa e baseada em conteúdo, visa aumentar a diversidade e a relevância das recomendações.

1. Aquisição de Dados

Para adquirir os dados, utilizamos a biblioteca kaggle, que permite o download direto do dataset da competição Netflix Prize. Este conjunto de dados compreende um histórico robusto de avaliações, contendo:

- **ID do usuário**
- **ID do filme**
- **Data da avaliação**
- **Nota dada (0 a 5)**

Esse histórico é essencial para a criação de um sistema de recomendação que capture padrões de comportamento e preferências ao longo do tempo.

2. Preparação e Tratamento dos Dados

Para manipular e estruturar os dados, empregamos as bibliotecas os, shutil, pandas, e numpy. O tratamento dos dados foi dividido em três principais etapas:

2.1. Limpeza e Filtragem de Dados

Inicialmente, limpamos e formatamos os dados para remover inconsistências e garantir a homogeneidade. Dados incompletos ou errôneos foram identificados e removidos conforme necessário. Com o uso de pandas e numpy, filtramos usuários com, no mínimo, 10 avaliações, reduzindo a **sparsity** (escassez de dados) e melhorando a confiabilidade do modelo.

2.2. Separação em Conjuntos de Treinamento e Teste

Dividimos o conjunto de dados em treino e teste utilizando o arquivo **"probe"** fornecido pelo Netflix Prize Dataset. Este conjunto contém clientes com filmes em comum no conjunto de treinamento, mas sem a coluna de avaliação visível, permitindo a validação futura do modelo em dados reais. Essa abordagem proporciona uma base sólida para que o modelo aprenda padrões de recomendação em um ambiente de treinamento e seja testado em condições controladas, representando o ambiente real.

2.3. Conversão para o Formato Esperado pelo Modelo

O conjunto de dados foi convertido para um formato compatível com o algoritmo SVD, usado posteriormente na modelagem. Este processo envolveu a transformação de tabelas e a normalização das variáveis de interesse, garantindo que o modelo receba dados consistentes e otimizados.

3. Análise Exploratória e Visualização

Para obter insights e observar tendências nos dados, utilizamos matplotlib e seaborn para a visualização gráfica. Foram explorados aspectos como:

- **Distribuição das avaliações**: identificação de notas predominantes, variabilidade entre elas, e evolução ao longo do tempo.
- **Satisfação dos usuários**: análise do comportamento das avaliações e identificação de possíveis mudanças de satisfação dos usuários com o tempo, influenciadas pela qualidade das recomendações.

Essas análises preliminares forneceram um contexto sobre o conjunto de dados e direcionaram as próximas etapas da modelagem.

4. Modelagem

Para o treinamento do modelo de recomendação, utilizamos a biblioteca Surprise, que fornece uma implementação otimizada do algoritmo **SVD (Singular Value Decomposition)**. A escolha do SVD se deve ao seu desempenho comprovado na modelagem de recomendações baseadas em **fatores latentes**, o que permite identificar relações ocultas entre usuários e filmes, mesmo em conjuntos de dados esparsos.

- **Configuração do SVD**: configuramos o SVD para ajustar os fatores latentes conforme os padrões de avaliação dos usuários, buscando associar perfis similares e recomendações mais precisas.
- **Treinamento e Teste**: o conjunto de dados de treinamento foi utilizado para que o modelo aprenda as relações entre usuários e filmes, enquanto o conjunto de teste serviu para avaliar a qualidade das previsões.

5. Avaliação do Desempenho do Modelo

A precisão do modelo foi avaliada pela métrica **Root Mean Square Error (RMSE)**, que mede a discrepância entre as avaliações reais e as previsões do modelo. O RMSE penaliza erros maiores, proporcionando uma métrica rigorosa para a qualidade das recomendações.

- **Métrica de RMSE**: Após a execução da fase inicial de modelagem, foi realizado um levantamento dos resultados preliminares, o modelo apresentou um RMSE de 1.0565, indicando margem significativa para otimizações. Este valor oferece uma referência inicial e será utilizado para ajustar hiperparâmetros e avaliar a eficácia de modificações no modelo.

6. Considerações sobre Melhorias Finais

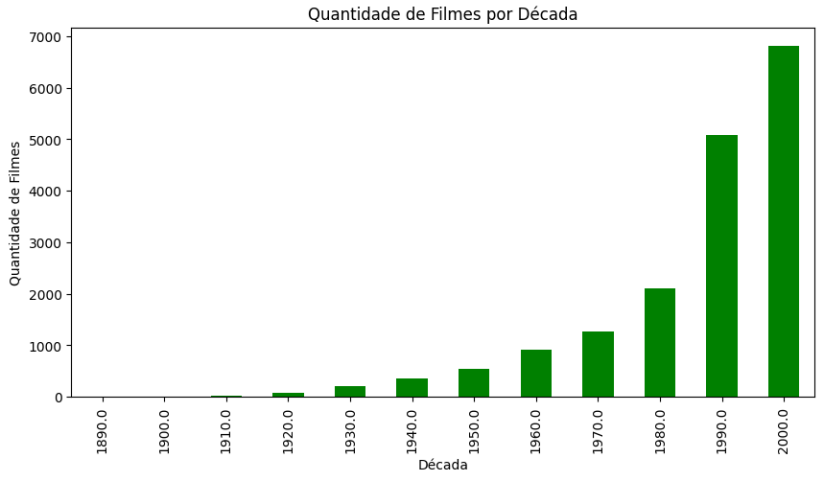
Com base nos resultados obtidos, identificamos a necessidade de otimizações adicionais no modelo. Entre as possibilidades estão o ajuste de hiperparâmetros do SVD, a integração de abordagens híbridas (que combinam filtragem colaborativa e métodos baseados em conteúdo) e a avaliação do modelo em diferentes faixas de dados temporais.

Resultados

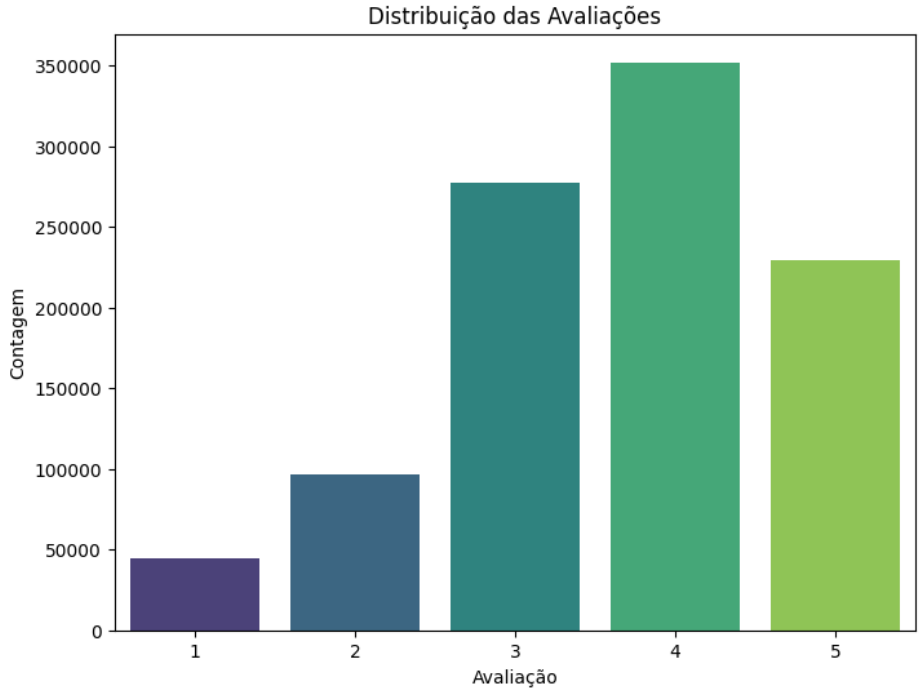
Explorando os dados

A Netflix revolucionou o mundo do entretenimento, inicialmente enviando DVDs às casas dos clientes e evoluindo para uma plataforma de streaming de conteúdo, e posteriormente produtora e desenvolvedora de filmes e roteiros originais. Em uma competição do kaggle, a Netflix decidiu abrir os seus dados com o desafio de quem conseguia fazer o melhor algoritmo de recomendação com um prêmio de $1MM de dólares. Esses dados estão divididos entre avaliações dos usuários aos filmes, uma tabela de informações sobre o filme e podem ser adquiridos publicamente no site do kaggle:
https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data/data

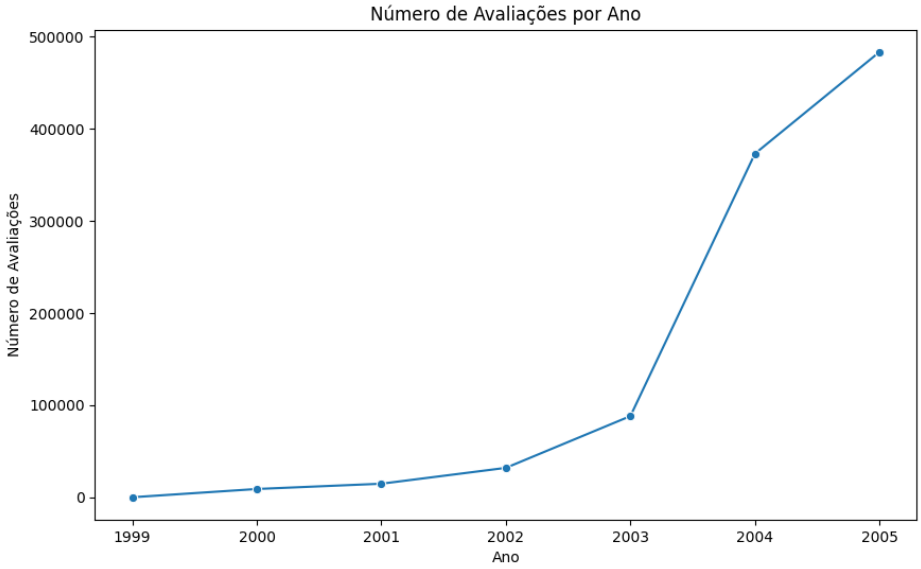
A empresa, lançou muitos filmes (17.026 até a época do lançamento do dataset) ao longo do tempo, com conteúdo de diversas eras tentando atingir todos os públicos, das crianças aos idosos:



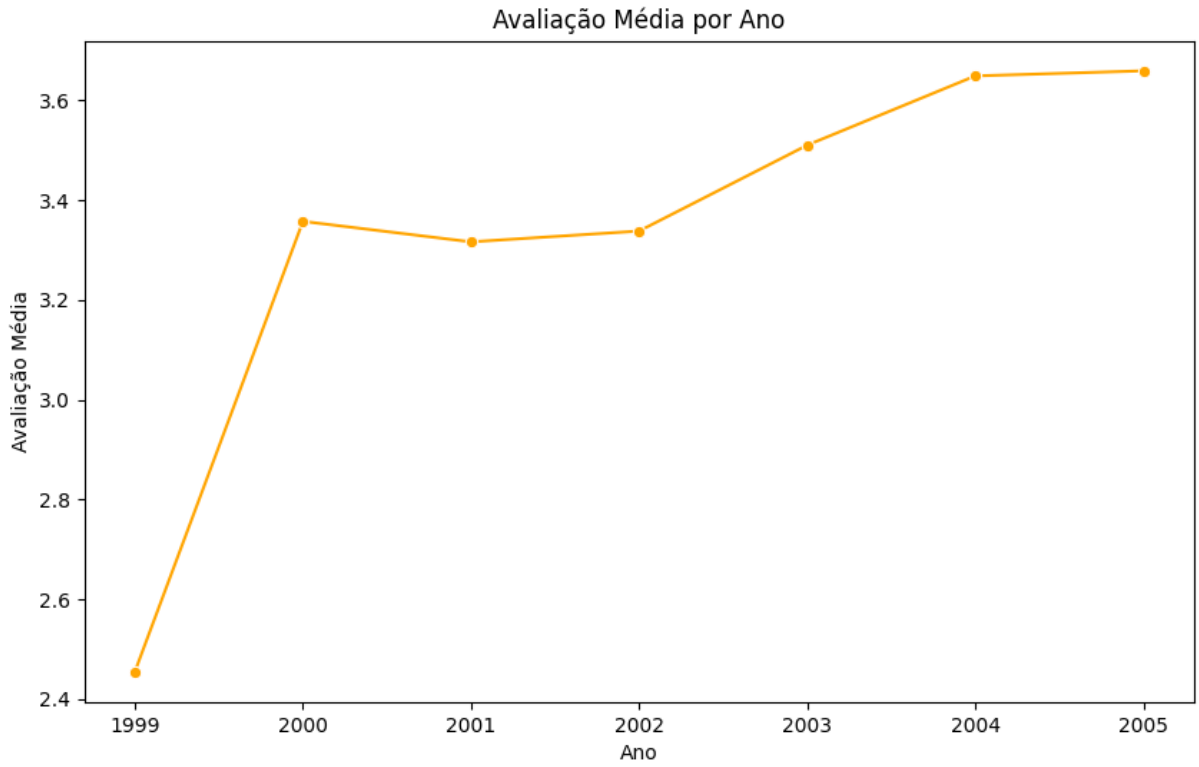
Os dados de avaliações dos filmes são gigantescos e contém, por ID de usuário, o filme a data da avaliação e a nota em si. Os usuários podem dar notas de 0 a 5 e essa é a distribuição:



No começo do serviço, as avaliações eram menos comuns, mas, isso mudou com o tempo:



Parece também que os usuários ficaram mais satisfeitos com o conteúdo ao longo do tempo, indicando uma possível causa como a qualidade das ofertas personalizadas que os algoritmos fazem em detrimento a escolhas aleatórias:



Sabendo da quantidade massiva de informações que temos a disposição, vamos seguir para o treinamento e teste de performance do modelo.

Tratar e preparar a base de dados para o treinamento.

Sabendo da quantidade massiva de informações que temos a disposição, vamos seguir para o tratamento da base para o treinamento do modelo.

Os dados do Netflix Prize contam com um conjunto chamado “probe” que consiste em clientes com filmes que estão no conjunto de treinamento, mas sem a coluna de avaliação visível, que servirá como amostra de testes.

Já para o treinamento, usaremos a combinação de dados dos clientes com mais de 10 avaliações, para evitar sparsity e melhorar a performance do modelo.

Tratando os dados txt:

```
# Carregando o arquivo de avaliações (combined_data_1.txt)
data_path = 'combined_data_1.txt'

# Criar um DataFrame vazio para armazenar os dados
data = {'MovieID': [], 'CustomerID': [], 'Rating': [], 'Date': []}

# Lendo o arquivo linha por linha
with open(data_path, 'r') as file:
    movie_id = None
    for line in file:
        line = line.strip()
        if line.endswith(':'):
            # Linha com o MovieID
            movie_id = int(line.replace(':', ''))
        else:
            # Linha com avaliação (CustomerID, Rating, Date)
            customer_id, rating, date = line.split(',')
            data['MovieID'].append(movie_id)
            data['CustomerID'].append(int(customer_id))
            data['Rating'].append(float(rating))
            data['Date'].append(date)

# Convertendo o dicionário para um DataFrame
df = pd.DataFrame(data)

# Exibir as primeiras linhas
df.head()
```

Filtrando usuários com, no mínimo, 10 avaliações:

```
# Definindo limites mínimos
min_ratings_per_user = 10
min_ratings_per_movie = 10

# Filtrando usuários que avaliaram menos de 10 filmes
filtered_users = df['CustomerID'].value_counts()[df['CustomerID'].value_counts() >= min_ratings_per_user].index
df = df[df['CustomerID'].isin(filtered_users)]

# Filtrando filmes que receberam menos de 10 avaliações
filtered_movies = df['MovieID'].value_counts()[df['MovieID'].value_counts() >= min_ratings_per_movie].index
df = df[df['MovieID'].isin(filtered_movies)]

# Exibindo as dimensões após o filtro
df.shape
```

Aplicando a separação dos conjuntos de teste e treinamento com o arquivo probe:

```python
# Carregar o arquivo probe.txt
probe_path = 'probe.txt'

# Lendo o arquivo probe
probe_data = {'MovieID': [], 'CustomerID': []}

with open(probe_path, 'r') as file:
    movie_id = None
    for line in file:
        line = line.strip()
        if line.endswith(':'):
            # Linha com o MovieID
            movie_id = int(line.replace(':', ''))
        else:
            # Linha com o CustomerID
            customer_id = int(line)
            probe_data['MovieID'].append(movie_id)
            probe_data['CustomerID'].append(customer_id)

# Convertendo para DataFrame
df_probe = pd.DataFrame(probe_data)

# Criar uma coluna que indique se os dados estão no probe
df['is_probe'] = df[['MovieID', 'CustomerID']].apply(
    lambda x: 1 if ((x['MovieID'], x['CustomerID']) in zip(df_probe['MovieID'], df_probe['CustomerID'])) else 0, axis=1)

# Dividir em treino e teste
df_train = df[df['is_probe'] == 0]
df_test = df[df['is_probe'] == 1]

# Remover a coluna auxiliar
df_train.drop(columns='is_probe', inplace=True)
df_test.drop(columns='is_probe', inplace=True)

# Verificar tamanhos dos conjuntos de treino e teste
df_train.shape, df_test.shape
```

Esse tratamento nos permitiu ter 968858, 4 em treino e 30919, 4 em teste.

Técnica de treinamento

Para o treinamento, vamos usar a técnica de filtragem colaborativa baseada em fatores latentes, com o algoritmo SVD. O conjunto de treinamento será usado para o modelo aprender as relações entre usuários e filmes. Essa abordagem é baseada em parelhar usuários com perfis semelhantes (em termos de padrões de avaliação).

Para isso, com a biblioteca surprise, implementaremos sobre os dados tratados e separados entre treino e teste, o seguinte algoritmo:

```
from surprise import SVD, Dataset, Reader
from surprise.model_selection import train_test_split, accuracy

# Configurar os dados para a biblioteca Surprise
reader = Reader(rating_scale=(1, 5))

# Preparar o conjunto de treino para Surprise
train_data = Dataset.load_from_df(df_train[['CustomerID', 'MovieID', 'Rating']], reader)

# Preparar o conjunto de teste para Surprise
test_data = list(zip(df_test['CustomerID'], df_test['MovieID'], df_test['Rating']))

# Carregar o conjunto de treino
trainset = train_data.build_full_trainset()

# Inicializar o modelo SVD
model = SVD()

# Treinar o modelo
model.fit(trainset)

# Fazer previsões para o conjunto de teste
predictions = model.test(test_data)

# Avaliar o desempenho do modelo com RMSE
accuracy.rmse(predictions)
```

Avaliando o desempenho do modelo.

Escolhemos como métrica de avaliação o RMSE, que avalia a diferença entre as previsões e as avaliações reais feitas pelo modelo. Ele se baseia no cálculo da raiz quadrada da média dos erros ao quadrado, penalizando erros maiores em detrimento aos menores.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - p_i)^2}$$

Como resultados iniciais, nosso modelo alcançou uma acurácia de 1.0565, revelando uma margem significativa para otimização e obtenção de resultados mais precisos.

Conclusão e trabalhos futuros