# 1. INTRODUÇÃO

Para se fazer a recomendação de filme em base do qual for escolhido, foi utilizado o dataset MovieLens 100k, em que contém avaliações de usuários sobre diversos títulos cinematográficos. O sistema foi implementado no Python utilizando a similaridade do cosseno como métrica principal para identificar a semelhança entre filmes. A recomendação é baseada em uma abordagem híbrida, combinando dois métodos: a semelhança de avaliações (filtragem colaborativa) e a semelhança entre os títulos dos filmes (filtragem baseada em conteúdo). O algoritmo calcula a similaridade entre pares de filmes e, a partir de um filme selecionado, retorna aqueles considerados mais semelhantes, com base nas preferências de usuários e nas características textuais dos títulos. Essa abordagem visa melhorar a precisão das recomendações, aproveitando diferentes aspectos dos dados disponíveis no MovieLens.

# 2. OBJETIVO

O principal objetivo deste algoritmo é desenvolver um sistema de recomendação de filmes capaz de sugerir títulos similares a partir de um filme de referência, utilizando técnicas de similaridade do cosseno. A proposta é combinar informações provenientes das avaliações dos usuários (indicando padrões de preferência) com elementos textuais dos títulos dos filmes, resultando em uma abordagem híbrida que potencializa a qualidade das recomendações. Com isso, busca-se oferecer ao usuário sugestões mais relevantes e personalizadas, mesmo quando há pouca informação explícita sobre suas preferências individuais.

# 3. DATASET UTILIZADO

O presente trabalho utiliza o dataset MovieLens 100k, disponível no Kaggle e fornecido pelo GroupLens Research. Trata-se de um conjunto amplamente utilizado em estudos de sistemas de recomendação, contendo 100.000 avaliações explícitas realizadas por 943 usuários em 1.682 filmes, com notas variando de 1 a 5. As informações de avaliação estão no arquivo “u.data”, estruturado com os campos “user\_id“, “item\_id“, “rating“ e “timestamp“. Já os dados dos filmes são obtidos do arquivo “u.item“, contendo “movie\_id“ e “movie\_title“. O dataset é ideal para testes de algoritmos de recomendação por seu tamanho reduzido, padronização e amplo reconhecimento acadêmico.

# 4. ONDE ESTÁ SENDO APLICADO

O código foi desenvolvido utilizando a linguagem Python no ambiente de desenvolvimento Spyder, que faz parte do pacote Anaconda. Esse ambiente foi escolhido por sua praticidade na escrita e execução de scripts científicos, além de oferecer ferramentas integradas para visualização de dados e depuração de código.

Para o funcionamento do algoritmo, foi necessária a instalação da biblioteca scikit-learn (sklearn), utilizada para o cálculo da similaridade do cosseno e para o uso do vetorizador TF-IDF. Essa instalação foi realizada diretamente pelo Anaconda Navigator, facilitando a gestão dos pacotes e garantindo a compatibilidade com o ambiente.

# 5. COMO FUNCIONA O ALGORITMO

O funcionamento do algoritmo pode ser dividido em 6 etapas principais:

5.1. \*\*Carregamento dos Dados\*\*:

são carregados dois arquivos do dataset MovieLens — um com as avaliações dos usuários e outro com os títulos dos filmes.

5.2. \*\*Similaridade pelas Avaliações\*\*:

é construída uma matriz de usuários versus filmes, preenchida com as avaliações. A similaridade do cosseno é calculada entre os filmes com base nas notas dadas pelos usuários.

5.3. \*\*Similaridade pelos Títulos\*\*:

os títulos dos filmes são transformados em vetores numéricos usando a técnica “TF-IDF”. A similaridade do cosseno é então calculada entre esses vetores.

5.4. \*\*Combinação das Similaridades\*\*:

as duas matrizes de similaridade (avaliações e títulos) são combinadas por meio de um fator de ponderação (alpha), permitindo ajustar a influência de cada fonte de informação.

5.5. \*\*Função de Recomendação\*\*:

é definida uma função que, dado um filme, retorna os filmes mais semelhantes a ele com base na matriz de similaridade combinada.

5.6. \*\*Exemplo de Uso\*\*:

o sistema solicita ao usuário o ID de um filme (também pode ser visualizada pela planilha do excell postada no github) e, com base nisso, imprime os cinco filmes mais similares, exibindo também os graus de similaridade correspondentes.

### 6. RESULTADOS

Rodando o algoritmo na prática no Anaconda Navigator para utilizar o Spyder, foram realizados testes utilizando três filmes distintos do conjunto de dados como ponto de partida. Adiante, apresentam-se os resultados obtidos para cada exemplo, considerando a combinação das similaridades por avaliações e títulos com peso de 0.2 para os títulos:

#### Exemplo 1 – Filme ID 1 (Toy Story (1995))

Filmes recomendados mais semelhantes:

* Star Wars (1977) — similaridade: **0.59**
* Return of the Jedi (1983) — **0.56**
* Independence Day (ID4) (1996) — **0.55**
* The Rock (1996) — **0.53**
* Twelve Monkeys (1995) — **0.51**

#### Exemplo 2 – Filme ID 50 (Star Wars (1977))

Filmes recomendados mais semelhantes:

* Return of the Jedi (1983) — **0.71**
* Raiders of the Lost Ark (1981) — **0.61**
* Empire Strikes Back, The (1980) — **0.60**
* Toy Story (1995) — **0.59**
* Star Trek: First Contact (1996) — **0.59**

#### Exemplo 3 – Filme ID 100 (Fargo (1996))

Filmes recomendados mais semelhantes:

* Twelve Monkeys (1995) — **0.55**
* Star Wars (1977) — **0.55**
* The Godfather (1972) — **0.52**
* Jerry Maguire (1996) — **0.52**
* Pulp Fiction (1994) — **0.51**

Esses resultados estão retornando filmes que possuem alta similaridade tanto em termos de preferência dos usuários quanto de características textuais. A coerência dos filmes recomendados — frequentemente populares ou de gêneros semelhantes ao filme de referência — evidencia que a abordagem híbrida implementada é eficaz na geração de sugestões relevantes.