ANÁLISE E FUNCIONAMENTO DE ALGORITMOS DE RECOMENDAÇÃO EM PLATAFORMAS DE STREAMING E REDES SOCIAIS

Wesley Emanuel Da Silva[1], André Faria Ruaro[2]

Resumo: Sistemas de recomendação estão no centro da experiência digital atual, mas equilibrar precisão (acertar o gosto do usuário) com diversidade (oferecer recomendações variadas) continua sendo um desafio técnico e ético relevante, especialmente pelo risco de formação de bolhas de filtro. Este trabalho compara dois algoritmos clássicos: a Filtragem Colaborativa baseada em SVD (*Singular Value Decomposition*) e a Recomendação Baseada em Conteúdo com TF-IDF (*Term Frequency–Inverse Document Frequency*), aplicados ao dataset MovieLens 1M com 1.000.209 avaliações de 6.040 usuários. Os dois modelos foram avaliados simultaneamente com quatro métricas (Precision@10, Recall@10, NDCG@10 e Diversidade intra-lista) sobre os 20 usuários mais ativos, usando protocolo *leave-30%-out* com semente fixa para garantir reprodutibilidade. Os resultados revelaram um achado contra-intuitivo: o SVD superou o TF-IDF não só em precisão (média de 0,705 contra 0,080 em Precision@10), como também em diversidade (0,797 contra 0,087). Em 12 dos 20 usuários, a diversidade do TF-IDF foi zero, ou seja, todos os dez filmes recomendados pertenciam ao mesmo gênero. Esse comportamento ocorre porque o modelo de conteúdo amplifica a concentração do catálogo em Drama e Comédia, criando bolhas de filtro em vez de combatê-las. O trabalho conclui que o *trade-off* entre precisão e diversidade é mais complexo do que a literatura costuma apresentar, e que a escolha do algoritmo deve considerar também a distribuição de gêneros do catálogo. Um protótipo interativo foi desenvolvido para demonstração ao vivo, incluindo simulação de *cold start* com novos usuários.

Palavras-chave: sistemas de recomendação; filtragem colaborativa; recomendação baseada em conteúdo; diversidade; bolhas de filtro; MovieLens.

[1]Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), Criciúma - Santa Catarina - Brasil. wesleyemanueldasilva0@gmail.com

[2]Orientador, Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), Criciúma - Santa Catarina - Brasil. andre.ruaro@unesc.net

ABSTRACT: Recommendation systems are central to today's digital experience, but balancing precision (matching user preferences) with diversity (offering varied recommendations) remains a relevant technical and ethical challenge, especially given the risk of filter bubble formation. This work compares two classical algorithms: Collaborative Filtering based on SVD (Singular Value Decomposition) and Content-Based Recommendation using TF-IDF (Term Frequency–Inverse Document Frequency), applied to the MovieLens 1M dataset with 1,000,209 ratings from 6,040 users. Both models were evaluated simultaneously with four metrics (Precision@10, Recall@10, NDCG@10, and Intra-list Diversity) on the 20 most active users, using a leave-30%-out protocol with a fixed seed to ensure reproducibility. The results revealed a counter-intuitive finding: SVD outperformed TF-IDF not only in precision (mean of 0.705 vs. 0.080 in Precision@10), but also in diversity (0.797 vs. 0.087). In 12 of the 20 users, TF-IDF diversity was zero, meaning all ten recommended films belonged to the same genre. This behavior occurs because the content-based model amplifies the catalog concentration in Drama and Comedy, creating filter bubbles instead of combating them. The work concludes that the precision-diversity trade-off is more complex than the literature typically suggests, and that algorithm selection should also consider the distribution of genres in the catalog. An interactive prototype was developed for live demonstration, including cold start simulation with new users.

Keywords: recommendation systems; collaborative filtering; content-based recommendation; diversity; filter bubbles; MovieLens.

1 INTRODUÇÃO

Nos últimos anos, os sistemas de recomendação tornaram-se parte essencial da experiência digital (Ricci; Rokach; Shapira, 2015). Plataformas como Netflix, TikTok, Spotify, YouTube e Amazon dependem diretamente desses algoritmos para personalizar o que cada usuário vê, ouve e compra. O impacto vai além do engajamento: recomendações eficazes aumentam receita, ampliam a visibilidade de criadores independentes e moldam os padrões de consumo de milhões de pessoas (Ko et al., 2022). Esses sistemas surgiram comercialmente nos anos 2000, quando catálogos digitais cresceram rápido demais para qualquer pessoa navegar manualmente. A Amazon foi pioneira ao usar histórico de compras para sugerir produtos, e desde então a área evoluiu das técnicas clássicas de filtragem

até arquiteturas de *deep learning* capazes de processar texto, imagem e áudio ao mesmo tempo (Ko et al., 2022; Zhou; Xiong; Chen, 2023). Quando funcionam mal, esses mesmos sistemas podem isolar o usuário em bolhas de conteúdo (situação em que a pessoa só vê conteúdos muito parecidos com o que já consome), reproduzir vieses presentes nos dados e prejudicar a descoberta de novos filmes, músicas ou produtos (Pariser, 2011; Chen et al., 2023).

Do ponto de vista técnico, esses algoritmos combinam abordagens clássicas com técnicas mais recentes de aprendizado profundo, grafos de conhecimento e modelos de grande escala (Zhou; Xiong; Chen, 2023; Raza et al., 2026). Entre as abordagens clássicas estão a Filtragem Colaborativa, que aprende com o comportamento coletivo dos usuários, e a Recomendação Baseada em Conteúdo, que analisa os atributos dos itens. A Filtragem Colaborativa parte do princípio de que usuários com comportamentos parecidos tendem a gostar das mesmas coisas: se você e outra pessoa avaliaram os mesmos filmes de forma similar, o sistema infere que você provavelmente vai gostar do que ela gostou e você ainda não viu. A variante mais eficaz dessa abordagem é a fatoração de matrizes (em especial o SVD, *Singular Value Decomposition*), que decompõe a matriz usuário-item em vetores latentes de baixa dimensão, capturando preferências implícitas mesmo quando os dados são esparsos (Koren; Bell; Volinsky, 2009). Os trabalhos de Koren, Bell e Volinsky (2009) estabeleceram as bases dessa técnica e continuam sendo referência central na área. Em avaliações experimentais sobre os datasets MovieLens e Netflix, Cremonesi, Koren e Turrin (2010) confirmaram que os modelos de fatoração de matrizes superam abordagens clássicas de vizinhança em tarefas de recomendação Top-N, cenário idêntico ao adotado neste trabalho. A limitação principal é o *cold start*, ou seja, quando não há dados suficientes sobre um usuário ou item novo, o modelo não tem base para fazer predições.

A Recomendação Baseada em Conteúdo segue uma lógica diferente: analisa os atributos dos próprios itens (gênero, descrição, metadados) e recomenda o que é parecido com o que o usuário já gostou (Lops; Gemmis; Semeraro, 2011). Técnicas como TF-IDF (*Term Frequency–Inverse Document Frequency*) transformam esses atributos em vetores numéricos, e a similaridade de cosseno entre esses vetores determina o quão próximos dois itens são. A vantagem é que funciona mesmo sem histórico colaborativo, o problema é a tendência de recomendar sempre mais do mesmo, o que, paradoxalmente, pode ser uma das principais causas de bolhas de

filtro (Ko et al., 2022; Roy; Dutta, 2022). Sistemas híbridos combinam as duas abordagens para tentar aproveitar o melhor de cada uma, e têm sido amplamente adotados em produção justamente por reduzirem o impacto do *cold start* sem abrir mão da qualidade das predições colaborativas (Burke, 2002; Ko et al., 2022; Raza et al., 2026). Mais recentemente, Zhou, Xiong e Chen (2023) e Raza et al. (2026) exploraram arquiteturas neurais para capturar relações complexas em dados multimodais, com melhor desempenho em cenários de grande escala, embora com custo computacional significativamente maior.

Para avaliar a qualidade das recomendações, a área usa métricas de precisão e de diversidade. As métricas de precisão medem o quanto o sistema acerta: Precision@K calcula a proporção de itens relevantes entre os K recomendados; Recall@K mede quantos dos itens relevantes do usuário aparecem na lista; e NDCG@K (*Normalized Discounted Cumulative Gain*) pondera a posição, de forma que acertar na primeira posição vale mais do que acertar na décima. Tais métricas, contudo, avaliam exclusivamente a relevância dos itens recomendados, sem considerar a variedade do conjunto apresentado ao usuário. Para isso existe a Diversidade intralista, proposta originalmente por Ziegler et al. (2005), que mede a dissimilaridade média entre os itens de uma mesma lista, ou seja, quanto maior o valor, mais variados os gêneros recomendados (Vargas; Castells, 2011; Roy; Dutta, 2022).

O equilíbrio entre precisão e diversidade é um dos desafios centrais da área, pois sistemas muito otimizados para acertar o gosto do usuário tendem a recomendar sempre o mesmo tipo de conteúdo, criando bolhas de filtro (situação em que o usuário deixa de ter contato com conteúdos fora do seu padrão de consumo habitual) (Chen et al., 2023). Além disso, a área enfrenta desafios de esparsidade (a matriz usuário-item é tipicamente preenchida em menos de 1% dos casos) e escalabilidade (sistemas reais lidam com milhões de usuários e itens). Do ponto de vista ético, Abdollahpouri, Burke e Mobasher (2019) quantificaram como o viés de popularidade pode prejudicar a descoberta de conteúdo diversificado e a satisfação do usuário a longo prazo, reforçando que a avaliação de sistemas de recomendação deve ir além das métricas de precisão e incluir indicadores de justiça e diversidade de exposição (Chen et al., 2023; Raza et al., 2026).

Apesar do volume crescente de pesquisas na área, ainda há uma lacuna relevante, pois poucos estudos combinam a avaliação de precisão com a avaliação de diversidade na mesma análise comparativa, e

menos ainda demonstram esse comportamento de forma prática e reprodutível sobre datasets de escala real. A maioria dos trabalhos foca em otimizar uma métrica de cada vez, sem examinar o que acontece com as outras (Roy; Dutta, 2022; Chen et al., 2023). Isso cria uma visão incompleta do comportamento dos algoritmos, especialmente em relação à formação de bolhas de conteúdo, que é um problema ao mesmo tempo técnico e ético.

Diante desse contexto, o objetivo geral deste trabalho é analisar e comparar dois algoritmos clássicos de recomendação, a Filtragem Colaborativa baseada em SVD e a Recomendação Baseada em Conteúdo com TF-IDF, avaliando simultaneamente a precisão e a diversidade das recomendações geradas sobre o dataset MovieLens 1M, que reúne 1.000.209 avaliações de 6.040 usuários sobre 3.706 filmes (Harper; Konstan, 2016). Para alcançar o objetivo geral, foram definidos cinco objetivos específicos:

- revisar as principais abordagens de sistemas de recomendação e seus desafios clássicos, como *cold start*, esparsidade e viés algorítmico;
- implementar os modelos de Filtragem Colaborativa (SVD) e de Recomendação Baseada em Conteúdo (TF-IDF) em Python;
- definir e aplicar as métricas de avaliação Precision@10, Recall@10, NDCG@10 e Diversidade intra-lista;
- avaliar comparativamente os dois modelos sob o mesmo protocolo reprodutível (*leave-30%-out* com semente fixa);
- desenvolver um protótipo interativo para demonstração dos algoritmos em tempo real, incluindo simulação de *cold start* com novos usuários.

O trabalho justifica-se pela lacuna identificada na literatura: poucos estudos avaliam precisão e diversidade na mesma análise comparativa e de forma reprodutível sobre dados de escala real, o que limita a compreensão do comportamento dos algoritmos quanto à formação de bolhas de filtro, um problema simultaneamente técnico e ético. O código, o dataset configurado e o protótipo funcional estão disponíveis para uso em pesquisas posteriores ou como referência para quem desenvolve sistemas de recomendação.

2 MATERIAIS E MÉTODOS

Para verificar se a Filtragem Colaborativa e a Recomendação Baseada em Conteúdo apresentam *trade-offs* distintos entre precisão e diversidade, foi desenvolvido um protótipo experimental em Python que implementou os dois algoritmos sobre o mesmo conjunto de dados e aplicou a mesma bateria de métricas sobre os resultados. Esta seção descreve o conjunto de dados utilizado, a configuração de cada modelo, o protocolo de avaliação e os recursos computacionais empregados.

2.1 CONJUNTO DE DADOS

O conjunto de dados utilizado foi o MovieLens 1M (Harper; Konstan, 2016), mantido pelo GroupLens Research Lab da Universidade de Minnesota e disponibilizado publicamente. É uma das referências mais relevantes em pesquisa de sistemas de recomendação, justamente por combinar volume suficiente de dados com variedade de perfis de usuário. O dataset reúne 1.000.209 avaliações numéricas em escala de 1 a 5 estrelas, atribuídas por 6.040 usuários a 3.706 filmes distintos. Cada usuário avaliou no mínimo 20 filmes, com média de 165,6 avaliações por usuário (máximo de 2.314) e média de 269,9 avaliações por filme.

A distribuição das notas é assimétrica positiva: notas 4 e 5 somam 57,5% do total (348.971 e 226.310 ocorrências, respectivamente), com nota média global de 3,58 e desvio padrão de 1,12. Esse padrão reflete um viés de seleção bastante comum nesse tipo de dado, pois as pessoas tendem a avaliar filmes que escolheram assistir e, por isso, têm mais chance de gostar. Os cinco gêneros mais frequentes no catálogo foram Drama (1.603 filmes), Comédia (1.200), Ação (503), Suspense (492) e Romance (471).

Os dados estão distribuídos em três arquivos: `ratings.dat` (avaliações, com separador `::`), `movies.dat` (metadados dos filmes, com gêneros no formato `"Action|Comedy|Drama"`) e `users.dat` (dados demográficos: gênero, faixa etária, ocupação e CEP). O catálogo completo de filmes em `movies.dat` contém 3.883 títulos, valor ligeiramente superior aos 3.706 com avaliações, pois alguns filmes constam no catálogo sem nenhuma avaliação registrada. A matriz de similaridade de cosseno foi construída sobre o catálogo completo, resultando em dimensões de 3.883 × 3.883.

2.2 MODELO DE FILTRAGEM COLABORATIVA

O algoritmo de Filtragem Colaborativa implementado foi o SVD, técnica de fatoração de matrizes que decompõe a matriz usuário-item em vetores latentes de baixa dimensão para capturar preferências implícitas, mesmo em cenários com muitos dados faltantes, o que é comum nesse tipo de problema, conforme descrito por Koren, Bell e Volinsky (2009). A implementação utilizou a biblioteca *scikit-surprise* 1.1.4.

O modelo foi treinado sobre o conjunto completo de avaliações via `build_full_trainset`, sem divisão prévia; a separação para avaliação foi feita por usuário, conforme explicado na Seção 2.5. Os hiperparâmetros adotados foram os valores padrão da biblioteca, com a semente aleatória fixada em 42 para garantir reprodutibilidade:

- **n_factors = 100**: dimensão dos vetores latentes de usuários e filmes;
- **n_epochs = 20**: iterações de descida de gradiente estocástico (SGD) sobre o conjunto de treino;
- **biased = True**: incorpora termos de *bias* global, por usuário e por item, o que melhora a qualidade das predições ao capturar tendências sistemáticas de avaliação;
- **lr_all = 0,005**: taxa de aprendizado para todos os parâmetros;
- **reg_all = 0,02**: fator de regularização L2, aplicado uniformemente para evitar overfitting.

Para gerar as recomendações Top-10, o modelo calculou a nota prevista para todos os filmes não avaliados pelo usuário e retornou os dez com maior pontuação predita. Com o ML-1M, o treinamento completo levou em média 4,45 segundos, custo pago uma única vez durante a inicialização do sistema e armazenado em cache para todas as interações subsequentes.

2.3 MODELO DE RECOMENDAÇÃO BASEADA EM CONTEÚDO

O modelo de Recomendação Baseada em Conteúdo utilizou vetorização TF-IDF aplicada aos gêneros de cada filme, combinada com similaridade de cosseno para identificar filmes parecidos com o histórico do usuário. A implementação empregou a classe `TfidfVectorizer` da biblioteca *scikit-learn* 1.8.0.

Cada filme foi representado por uma *string* de gêneros separados por espaço, por exemplo, `"Action Comedy Drama"`. O *TF-IDF* funcionou, na prática, como um modelo *bag-of-genres* ponderado por frequência inversa de documento: gêneros muito comuns no catálogo, como Drama e Comédia, receberam peso menor; gêneros raros, como *Film Noir* e Documentário, foram proporcionalmente mais valorizados. Todos os parâmetros do vectorizador foram mantidos nos valores padrão (`ngram_range=(1,1)`, `norm='l2'`, `use_idf=True`, `smooth_idf=True`).

A matriz de similaridade de cosseno foi calculada entre todos os pares de filmes do catálogo (3.883 × 3.883, ou seja, aproximadamente 15 milhões de pares) uma única vez durante a inicialização. O tempo medido foi de aproximadamente 30,6 ms, valor explicado pelo fato de que os vetores TF-IDF de gêneros são extremamente esparsos: cada filme tem no máximo 18 valores não-zero (os 18 gêneros possíveis), e o scikit-learn aproveita essa esparsidade no cálculo. A matriz resultante foi mantida em memória, ocupando cerca de 115 MB de RAM. Essa abordagem de pré-computação é viável para o escopo deste trabalho, em datasets de larga escala como o Netflix (milhões de títulos), seria necessário adotar técnicas de vizinhança aproximada (*approximate nearest neighbors*) para tornar o cálculo praticável.

Na geração das recomendações Top-10, os escores de similaridade dos filmes com nota maior ou igual a 4 no histórico do usuário foram acumulados e os dez filmes não assistidos com maior pontuação agregada foram retornados. Vale destacar que esse acúmulo por loop Python puro se mostrou um gargalo crítico no ML-1M: para o usuário com mais avaliações (2.314 avaliações, das quais 1.210 positivas), o tempo de resposta chegou a aproximadamente 180 segundos. Isso faz sentido quando se contabiliza o custo real: são 1.210 filmes curtidos × 3.883 itens, ou seja, aproximadamente 4,7 milhões de iterações Python, cada uma com acesso a array NumPy, conversão de tipo e atualização de dicionário. A substituição por operação vetorizada NumPy, expressa como `cosine_sim[liked_indices].sum(axis=0)`, reduziu o tempo para cerca de 151 ms, uma diferença de aproximadamente 1.200 vezes, possível porque o NumPy executa o loop em C puro, sem o *overhead* do interpretador. Essa otimização foi aplicada nos experimentos de avaliação em lote, mas representa uma limitação do protótipo interativo que merece atenção em trabalhos futuros.

2.4 TRATAMENTO DO *COLD START*

Ambos os modelos implementaram um mecanismo de *fallback* para o problema de *cold start* (situação em que o usuário não tem histórico de avaliações positivas suficiente para gerar recomendações personalizadas). Quando o conjunto de filmes com nota maior ou igual a 4 está vazio, o sistema recorre à função `get_popular_movies()`, que retorna os filmes com maior média de avaliações dentre os que possuem ao menos 20 avaliações no dataset. Esse limiar evita que filmes com poucas avaliações, e portanto com médias estatisticamente pouco confiáveis, dominem o resultado.

Para novos usuários criados durante a sessão do protótipo, o SVD foi retreinado com as novas avaliações somente quando o usuário forneceu ao menos três avaliações positivas. Abaixo desse limiar, o sistema operou em modo de *cold start* puro, usando popularidade como critério. Na prática, isso permitiu demonstrar ao vivo como a ausência de histórico afeta a personalização, um dos desafios clássicos discutidos na introdução.

2.5 PROTOCOLO DE AVALIAÇÃO OFFLINE

A avaliação comparativa dos modelos seguiu o protocolo *leave-p-out* (Herlocker et al., 2004), com p = 30%: para cada usuário avaliado, 30% dos filmes com nota maior ou igual a 4 foram separados como *ground truth* (o que o usuário realmente gosta, usado para medir o acerto), enquanto os 70% restantes ficaram disponíveis como histórico para que os modelos gerassem as recomendações. A semente aleatória 42 foi fixada em todas as operações de amostragem, garantindo que qualquer pesquisador que repita o experimento obtenha exatamente os mesmos conjuntos de teste.

A avaliação foi conduzida sobre os 20 usuários com maior número de avaliações no dataset (entre 1.216 e 2.314 avaliações cada). A escolha por usuários com histórico extenso garante que o protocolo *leave-30%-out* produza conjuntos de teste representativos, pois com históricos curtos, separar 30% gera um *ground truth* pequeno demais para métricas confiáveis. Por isso, usuários com menos de cinco filmes bem avaliados foram excluídos da avaliação.

Outro ponto importante é que ambos os modelos foram avaliados sobre exatamente os mesmos 20 usuários, com os mesmos conjuntos de *ground truth*. Isso garante que qualquer diferença de desempenho observada venha das estratégias de recomendação em si, não de variações

nos dados de entrada.

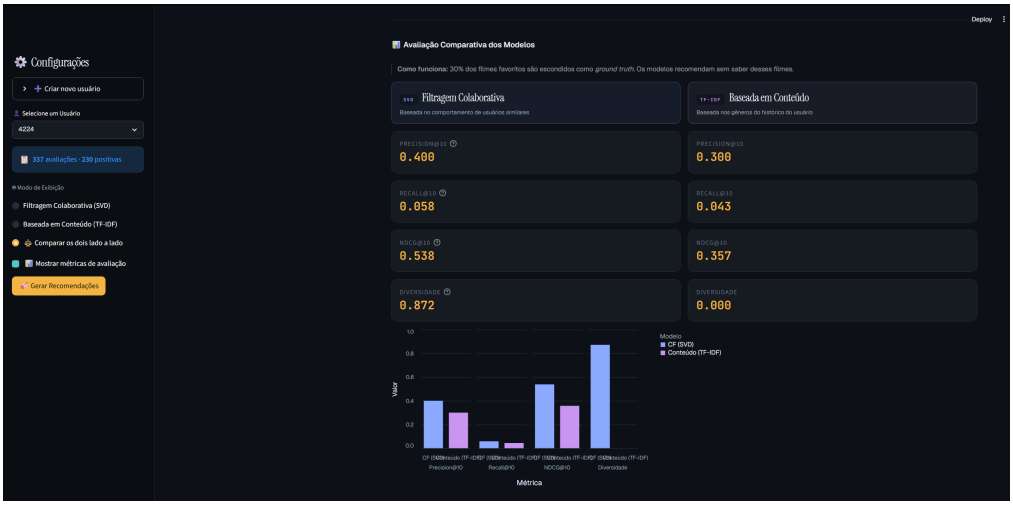
2.6 MÉTRICAS DE AVALIAÇÃO

Quatro métricas foram calculadas para cada usuário avaliado em cada modelo, seguindo as definições apresentadas na introdução:

- **Precision@10**: proporção dos 10 filmes recomendados presentes no *ground truth* do usuário, medindo o quanto a lista acerta o gosto real;
- **Recall@10**: proporção dos filmes do *ground truth* que aparecem entre os 10 recomendados, medindo a cobertura das preferências conhecidas;
- **NDCG@10** (*Normalized Discounted Cumulative Gain*): versão ponderada da precisão que valoriza mais os acertos nas primeiras posições do ranking, de forma que um filme relevante na posição 1 conta mais do que na posição 10;
- **Diversidade intra-lista**: média da dissimilaridade de cosseno entre todos os pares de filmes na lista Top-10, calculada sobre os vetores TF-IDF de gêneros. Valores próximos de 1 indicam lista variada em gêneros; valores próximos de 0 indicam concentração nos mesmos gêneros, o que se traduz, na prática, numa bolha de conteúdo.

Os resultados individuais por usuário, as médias consolidadas e a análise dos *trade-offs* observados são apresentados na seção de Resultados e Discussão. Também são mostrados de forma individual por usuário em tela via protótipo, conforme observado na Figura 1.

Figura 1 - Painel de avaliação comparativa do protótipo, exibindo as quatro métricas lado a lado para os modelos SVD e TF-IDF.



Fonte: Elaborado pelo autor.

2.7 AMBIENTE DA APLICAÇÃO

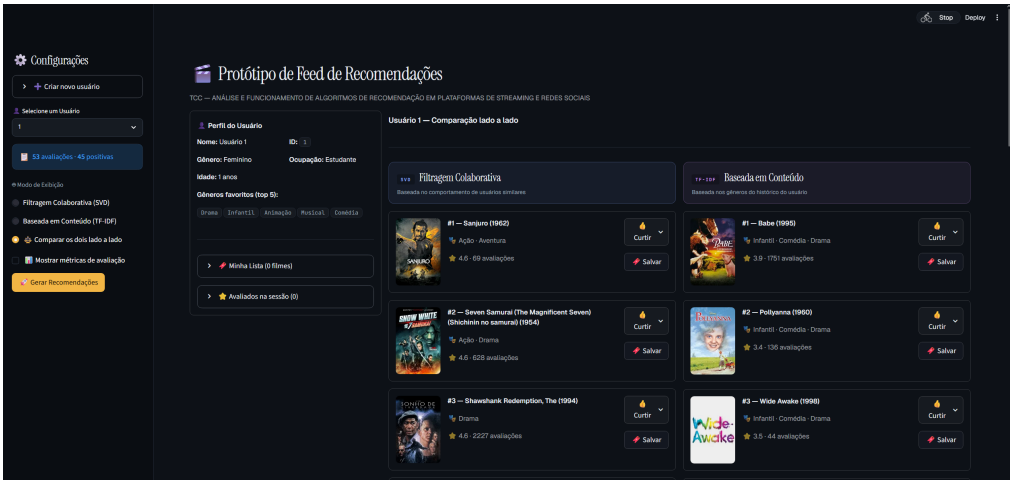
O desenvolvimento e a execução dos experimentos foram realizados em ambiente controlado, isolado, sem conexão externa e sem uso de serviços de computação em nuvem. A configuração de hardware consistiu em processador AMD Ryzen 5 5600 6-Core (3,50 GHz), 32 GB de memória RAM e sistema operacional Windows 11 Home. O protótipo foi implementado em Python 3.12, com ambiente virtual dedicado para isolamento e gerenciamento de dependências.

A biblioteca central para a Filtragem Colaborativa foi a *scikit-surprise* 1.1.4, que fornece a implementação do SVD com suporte nativo a hiperparâmetros configuráveis, descida de gradiente estocástico e termos de *bias*. Para a abordagem Baseada em Conteúdo, foi utilizada a *scikit-learn* 1.8.0, que disponibiliza a classe `TfidfVectorizer` para vetorização dos gêneros e funções otimizadas de similaridade de cosseno, além do cálculo do NDCG@K usado na avaliação. A manipulação e o processamento dos dados do MovieLens 1M ficaram a cargo do *pandas* 3.0.2, responsável pela leitura e filtragem dos arquivos `.dat`, e do NumPy 1.26.4, que viabilizou as operações vetorizadas sobre as matrizes de similaridade, reduzindo o tempo de geração de recomendações de aproximadamente 180 segundos para cerca de 151 ms no caso mais custoso.

A interface interativa do protótipo, conforme podemos observar na Figura 2, foi construída com o *Streamlit* 1.56.0, um *framework* Python que permite criar aplicações web sem necessidade de código front-end se-

parado. Foi por meio dele que a demonstração em tempo real dos algoritmos foi viabilizada, incluindo a simulação de *cold start* com novos usuários e a exibição dos cartazes dos filmes recomendados. Os cartazes foram obtidos dinamicamente via API do TMDB (*The Movie Database*), integrada através da biblioteca *requests* 2.33.1, que gerenciou as requisições HTTP para busca e exibição das imagens em tempo real.

Figura 2 - Interface principal do protótipo desenvolvido para demonstração dos algoritmos de recomendação.



Fonte: Elaborado pelo autor.

3 RESULTADOS E DISCUSSÃO

Esta seção apresenta os resultados da avaliação comparativa entre o SVD e o TF-IDF sobre o MovieLens 1M, seguida da interpretação dos achados com base no referencial teórico apresentado na introdução.

3.1 RESULTADOS

A avaliação foi conduzida sobre os 20 usuários com maior número de avaliações no dataset, usando o protocolo *leave-30%-out* com semente 42, conforme descrito na Seção 2.5. A Tabela 1 apresenta os resultados individuais por usuário para as quatro métricas calculadas.

Tabela 1 - Resultados por usuário: comparação entre SVD e TF-IDF (MovieLens 1M, Top-10)

Usuário	N.º de avaliações	Precision@10		Recall@10		NDCG@10		Diversidade	
		SVD	TF-IDF	SVD	TF-IDF	SVD	TF-IDF	SVD	TF-IDF
4169	2.314	0,900	0,200	0,025	0,006	0,922	0,305	0,766	0,000
1680	1.850	0,900	0,100	0,027	0,003	0,890	0,069	0,834	0,000
4277	1.743	1,000	0,300	0,023	0,007	1,000	0,271	0,760	0,107
1941	1.595	0,900	0,000	0,051	0,000	0,922	0,000	0,753	0,169
1181	1.521	0,900	0,000	0,100	0,000	0,936	0,000	0,809	0,169
889	1.518	0,600	0,000	0,055	0,000	0,672	0,000	0,801	0,000
3618	1.344	0,500	0,100	0,083	0,017	0,570	0,073	0,778	0,000
2063	1.323	0,700	0,000	0,060	0,000	0,763	0,000	0,831	0,000
1150	1.302	0,500	0,000	0,063	0,000	0,373	0,000	0,839	0,000
1015	1.286	0,600	0,100	0,025	0,004	0,669	0,066	0,793	0,000
5795	1.277	0,600	0,100	0,034	0,006	0,715	0,220	0,624	0,000
4344	1.271	0,600	0,000	0,030	0,000	0,712	0,000	0,898	0,000
1980	1.260	0,500	0,100	0,026	0,005	0,493	0,073	0,783	0,000
2909	1.258	0,700	0,200	0,029	0,008	0,688	0,203	0,770	0,373
1449	1.243	0,800	0,000	0,076	0,000	0,849	0,000	0,785	0,000
4510	1.240	0,700	0,200	0,083	0,024	0,801	0,155	0,879	0,244
424	1.226	0,800	0,100	0,034	0,004	0,820	0,110	0,833	0,000
4227	1.222	0,800	0,100	0,064	0,008	0,870	0,110	0,779	0,305
5831	1.220	0,400	0,000	0,016	0,000	0,353	0,000	0,866	0,000
3841	1.216	0,700	0,000	0,044	0,000	0,797	0,000	0,759	0,367
Média	–	**0,705**	**0,080**	**0,047**	**0,005**	**0,741**	**0,083**	**0,797**	**0,087**
Desvio padrão	–	0,167	0,089	0,025	0,006	0,182	0,099	0,058	0,135

Fonte: Elaborado pelo autor.

A Tabela 2 sintetiza as médias e os desvios padrão de cada modelo para facilitar a comparação direta.

Tabela 2 - Comparativo consolidado: médias e desvios padrão

Métrica	SVD (Filtragem Colaborativa)		TF-IDF (Baseado em Conteúdo)	
	Média	Desvio padrão	Média	Desvio padrão
Precision@10	0,705	0,167	0,080	0,089
Recall@10	0,047	0,025	0,005	0,006
NDCG@10	0,741	0,182	0,083	0,099
Diversidade	0,797	0,058	0,087	0,135

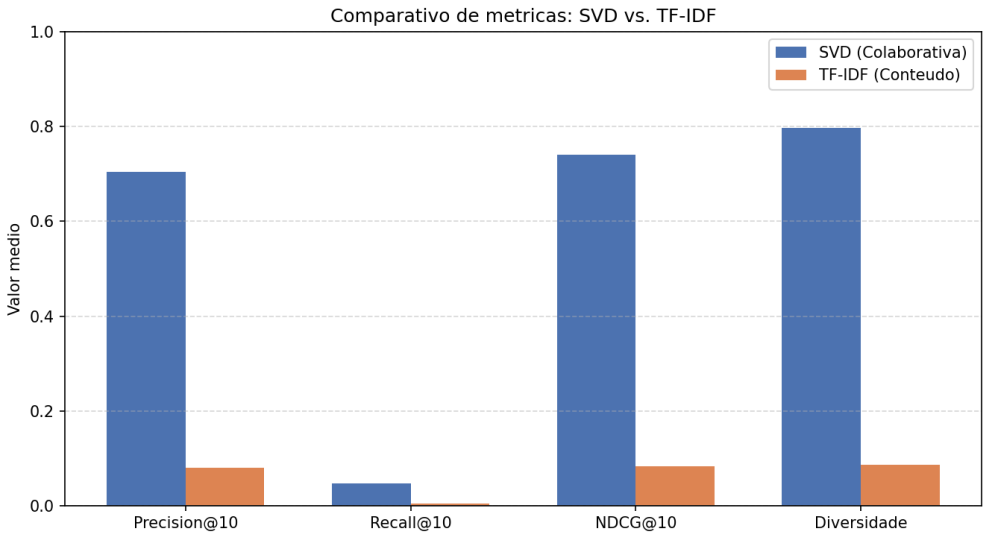
Fonte: Elaborado pelo autor.

3.2 DISCUSSÃO DE RESULTADOS

O SVD superou o TF-IDF em todas as métricas de relevância por margens expressivas. A Precision@10 média do SVD foi 0,705, contra 0,080 do TF-IDF, uma diferença de quase 9 vezes. O NDCG@10 seguiu o mesmo padrão: 0,741 contra 0,083. Na prática, isso significa que, ao receber dez recomendações do SVD, o usuário encontra em média sete filmes que realmente combinam com seu gosto; com o TF-IDF, esse número cai para menos de um. Como pode ser observado na Figura 3, o SVD supera

o TF-IDF de forma consistente em todas as quatro métricas avaliadas.

Figura 3 - Comparativo das médias de Precision@10, Recall@10, NDCG@10 e Diversidade intra-lista entre os modelos SVD e TF-IDF.



Fonte: Elaborado pelo autor.

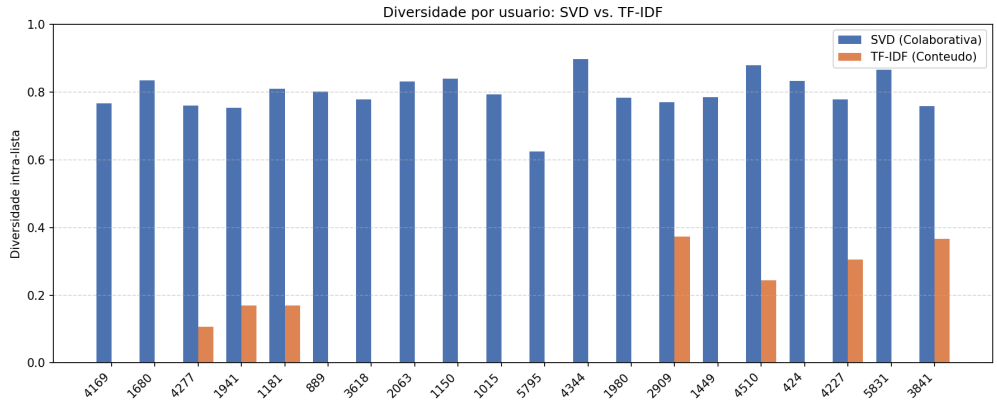
Esse resultado está alinhado com o que Koren, Bell e Volinsky (2009) demonstraram sobre a capacidade das técnicas de fatoração de matrizes de capturar padrões colaborativos latentes, ou seja, preferências que vão além da simples similaridade de gêneros. O SVD aprendeu, a partir de 1.000.209 avaliações, quais combinações de filmes os usuários tendem a apreciar juntos, mesmo quando esses filmes não compartilham gêneros óbvios. O TF-IDF, por não ter acesso a esse sinal colaborativo, fica restrito ao que os metadados dos filmes revelam diretamente.

O Recall@10 foi baixo nos dois modelos (0,047 para SVD e 0,005 para TF-IDF), o que é esperado nesse cenário. Com um catálogo de 3.706 filmes e usuários que têm dezenas a centenas de filmes favoritos, recuperar os filmes certos entre apenas dez recomendações é uma tarefa naturalmente difícil. Vale destacar que o SVD recuperou proporcionalmente dez vezes mais itens relevantes que o TF-IDF mesmo nessa condição adversa.

O resultado mais surpreendente foi o comportamento da métrica de diversidade. Contrariamente ao que se poderia esperar, o SVD apresentou diversidade intra-lista média de 0,797, enquanto o TF-IDF registrou apenas 0,087. Em 12 dos 20 usuários avaliados, a diversidade do TF-IDF foi exatamente zero, ou seja, todos os dez filmes recomendados pertenciam

ao mesmo gênero. A Figura 4 detalha esse comportamento por usuário, tornando visível o padrão de bolha de filtro do TF-IDF ao longo de toda a amostra avaliada.

Figura 4 - Diversidade intra-lista por usuário: SVD vs. TF-IDF. Em 12 dos 20 usuários, o TF-IDF registrou diversidade zero.



Fonte: Elaborado pelo autor.

Esse fenômeno tem uma explicação direta na natureza do modelo. O TF-IDF reconhece os gêneros preferidos no histórico do usuário (na maioria dos casos, Drama e Comédia, que dominam o catálogo com 1.603 e 1.200 filmes respectivamente) e recomenda filmes cada vez mais similares a esses gêneros. Como o catálogo do ML-1M é fortemente concentrado nessas duas categorias, o modelo converge sistematicamente para elas, criando exatamente a bolha de filtro que Chen et al. (2023) descrevem. O usuário que gosta de Drama recebe dez filmes de Drama, sem nenhum espaço para descoberta.

O SVD funciona de forma diferente. Ao identificar usuários com perfis similares, o modelo captura uma diversidade de gostos que vai além dos gêneros explícitos, já que usuários reais têm preferências que cruzam categorias e o SVD aprende essas combinações. Isso explica por que suas recomendações, embora mais precisas, também são mais variadas em gênero, comportamento consistente com as observações de Roy e Dutta (2022) sobre como sistemas colaborativos tendem a promover descoberta por meio da inteligência coletiva do grupo. Em estudo com usuários reais do MovieLens, Ekstrand et al. (2014) observaram padrão semelhante ao comparar algoritmos colaborativos, incluindo o SVD, quanto à diversidade percebida das listas recomendadas.

Esse achado questiona a premissa comum de que aumentar diversidade exige sacrificar precisão. O modelo mais preciso (SVD) também

foi o mais diverso. O modelo que deveria ser mais diversificado por explorar múltiplos gêneros do histórico (TF-IDF) acabou sendo o que mais criou bolhas. A limitação não está na abordagem em si, mas no fato de que, quando o conjunto de atributos descritivos dos itens é dominado por poucos gêneros, o modelo de conteúdo amplifica essa concentração em vez de combatê-la. Isso está diretamente ligado ao viés de popularidade que Abdollahpouri, Burke e Mobasher (2019) quantificaram: Drama e Comédia dominam as recomendações porque dominam o catálogo, e o modelo não tem mecanismo para corrigir esse desequilíbrio.

Outro ponto importante é que o SVD não é uma solução perfeita. O desvio padrão de 0,167 na Precision@10 indica variação considerável entre usuários, onde alguns obtiveram precisão perfeita (1,000, como o usuário 4277), enquanto outros ficaram em 0,400 (usuário 5831). Essa variabilidade sugere que o modelo é mais eficaz para usuários com perfis colaborativos bem definidos, exatamente a limitação de *cold start* que o TF-IDF, em tese, contorna ao usar metadados. Isso reforça o que Ko et al. (2022) apontam como motivação para sistemas híbridos: combinar os dois sinais para reduzir as fraquezas de cada abordagem isolada.

4 CONCLUSÃO

Este trabalho analisou e comparou dois algoritmos de recomendação (a Filtragem Colaborativa baseada em SVD e a Recomendação Baseada em Conteúdo com TF-IDF) sobre o dataset MovieLens 1M, avaliando simultaneamente métricas de precisão e diversidade. O objetivo era verificar se os dois modelos apresentam *trade-offs* distintos entre acertar o gosto do usuário e oferecer recomendações variadas, e o experimento entregou resultados que vão além do esperado.

Os cinco objetivos propostos foram atingidos. A revisão da literatura mapeou as principais abordagens de recomendação (filtragem colaborativa, baseada em conteúdo, sistemas híbridos e modelos de aprendizado profundo), junto com os desafios clássicos de *cold start*, esparsidade e viés algorítmico. As métricas de avaliação (Precision@K, Recall@K, NDCG e Diversidade intra-lista) foram definidas e aplicadas de forma sistemática. Um protótipo funcional foi desenvolvido em Python com interface interativa, incluindo demonstração de *cold start* com novos usuários, e a avaliação comparativa foi conduzida sobre os 20 usuários mais ativos do dataset, com protocolo *leave-30%-out* e semente fixa para reprodutibilidade.

O principal achado do trabalho foi contra-intuitivo: o SVD não só

superou o TF-IDF em precisão por uma margem expressiva (Precision@10 média de 0,705 contra 0,080), como também produziu listas mais diversas em gêneros (diversidade média de 0,797 contra 0,087). Em 12 dos 20 usuários avaliados, a diversidade do TF-IDF foi zero, ou seja, todos os dez filmes recomendados pertenciam ao mesmo gênero, configurando exatamente o problema de bolha de filtro discutido na introdução. Na prática, isso mostra que o *trade-off* entre precisão e diversidade é mais complexo do que a literatura costuma apresentar: não é o modelo mais preciso que cria bolhas, mas o modelo que opera sobre um corpus de metadados concentrado em poucos gêneros dominantes.

Do ponto de vista das contribuições, o trabalho demonstra empiricamente, sobre dados de escala real, que a abordagem colaborativa pode superar a baseada em conteúdo tanto em precisão quanto em diversidade quando o catálogo de metadados é pouco variado. Isso tem implicação direta para quem desenvolve sistemas de recomendação, onde a escolha entre Filtragem Colaborativa e Recomendação Baseada em Conteúdo não deve considerar só a disponibilidade de dados colaborativos, mas também a distribuição de gêneros do catálogo.

O trabalho tem limitações que precisam ser reconhecidas. A avaliação foi restrita aos 20 usuários com mais avaliações, um recorte que favorece perfis bem estabelecidos e pode não representar o comportamento de usuários casuais ou novos. Os hiperparâmetros do SVD foram mantidos nos valores padrão da biblioteca, sem ajuste fino; é possível que parâmetros otimizados alterem os resultados. O modelo de conteúdo usou apenas gêneros como atributos, sendo que descrições, atores, diretores e ano de lançamento poderiam enriquecer os vetores e reduzir o problema da concentração em Drama e Comédia. Por fim, o loop Python na geração de recomendações TF-IDF apresentou desempenho inviável para usuários muito ativos no protótipo interativo (até 180 segundos no pior caso), limitação que foi corrigida nos experimentos em lote, mas não na interface.

A partir dos achados deste trabalho, algumas questões de pesquisa ficam em aberto. Seria interessante investigar se abordagens híbridas, que combinam sinais colaborativos e de conteúdo, conseguem manter a precisão do SVD enquanto reduzem sua dependência de histórico, especialmente em cenários de *cold start*. Outro caminho relevante é examinar como o fenômeno da bolha de filtro do TF-IDF se comporta em catálogos com distribuição de gêneros mais uniforme: o problema persiste, ou é específico da concentração do MovieLens? A incorporação de metadados

textuais ricos (sinopses, palavras-chave) ao modelo de conteúdo também merece investigação, pois com representações semânticas mais densas, a diversidade intra-lista do TF-IDF poderia melhorar significativamente. Por fim, a comparação com arquiteturas de aprendizado profundo, como *Neural Collaborative Filtering*, usando o mesmo protocolo de avaliação deste trabalho, permitiria situar os resultados obtidos dentro do estado da arte atual.

REFERÊNCIAS

ABDOLLAHPOURI, H.; BURKE, R.; MOBASHER, B. **Managing popularity bias in recommender systems with personalized re-ranking**. In: **Proceedings of the Thirty-Second International Florida Artificial Intelligence Research Society Conference (FLAIRS-32)**. [S.l.]: AAAI Press, 2019. 2019. p. 413–418.

BURKE, R. **Hybrid recommender systems: Survey and experiments**. User Modeling and User-Adapted Interaction, v. 12, n. 4, p. 331–370. Acesso em: 06 jun. 2026. 2002, Disponível em: <https://doi.org/10.1023/A:1021240730564>.

CHEN, J. et al. **Bias and debias in recommender system: A survey and future directions**. ACM Transactions on Information Systems, v. 41, n. 3, p. 1–39. Acesso em: 01 jul. 2026. 2023, Disponível em: <https://doi.org/10.1145/3564284>.

CREMONESI, P.; KOREN, Y.; TURRIN, R. **Performance of recommender algorithms on top-n recommendation tasks**. In: **Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys)**. ACM, 2010. 2010. p. 39–46. Acesso em: 01 jul. 2026. Disponível em: <https://doi.org/10.1145/1864708.1864721>.

EKSTRAND, M. D. et al. **User perception of differences in recommender algorithms**. In: **Proceedings of the Eighth ACM Conference on Recommender Systems (RecSys)**. ACM, 2014. 2014. p. 161–168. Acesso em: 01 jul. 2026. Disponível em: <https://doi.org/10.1145/2645710.2645737>.

HARPER, F. M.; KONSTAN, J. A. **The MovieLens datasets: History and context**. ACM Transactions on Interactive Intelligent Systems, v. 5, n. 4, p. 19:1–19:19. Acesso em: 18 mai. 2026. 2016, Disponível em: <https://doi.org/10.1145/2827872>.

HERLOCKER, J. L. et al. **Evaluating collaborative filtering recommender systems**. ACM Transactions on Information Systems, v. 22, n. 1, p. 5–53. Acesso em: 06 jun. 2026. 2004, Disponível em: <https://doi.org/10.1145/963770.963772>.

KO, H. et al. **A survey of recommendation systems: Recommendation models, techniques, and application fields**. Electronics, v. 11, n. 1, p. 141. Acesso em: 18 mai. 2026. 2022, Disponível em: <https://www.mdpi.com/2079-9292/11/1/141>.

KOREN, Y.; BELL, R.; VOLINSKY, C. **Matrix factorization techniques for recommender systems**. Computer, v. 42, n. 8, p. 30–37. Acesso em: 18 mai. 2026. 2009, Disponível em: <https://doi.org/10.1109/MC.2009.263>.

LOPS, P.; GEMMIS, M. de; SEMERARO, G. Content-based recommender systems: State of the art and trends. In: **Recommender Systems Handbook**. Springer, 2011. p. 73–105. Acesso em: 06 jun. 2026. Disponível em: <https://doi.org/10.1007/978-0-387-85820-3_3>.

PARISER, E. **The Filter Bubble: What the Internet Is Hiding from You**. New York: Penguin Press. 2011.

RAZA, S. et al. **A comprehensive review of recommender systems: Transitioning from theory to practice**. Computer Science Review, v. 59, p. 100849. Acesso em: 01 jul. 2026. 2026, Disponível em: <https://www.sciencedirect.com/science/article/pii/S157401372500125X>.

RICCI, F.; ROKACH, L.; SHAPIRA, B. Recommender systems: Introduction and challenges. In: **Recommender Systems Handbook**. Springer, 2015. p. 1–34. Acesso em: 06 jun. 2026. Disponível em: <https://doi.org/10.1007/978-1-4899-7637-6_1>.

ROY, D.; DUTTA, M. **A systematic review and research perspective on recommender systems**. Journal of Big Data, v. 9, p. 59. Acesso em: 18 mai. 2026. 2022, Disponível em: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00592-5>.

VARGAS, S.; CASTELLS, P. **Rank and relevance in novelty and diversity metrics for recommender systems**. In: **Proceedings of the Fifth ACM Conference on Recommender Systems (RecSys)**. ACM, 2011. 2011. p. 109–116. Acesso em: 01 jul. 2026. Disponível em: <https://doi.org/10.1145/2043932.2043955>.

ZHOU, H.; XIONG, F.; CHEN, H. **A comprehensive survey of recommender systems based on deep learning**. Applied Sciences, v. 13, n. 20, p. 11378. Acesso em: 18 mai. 2026. 2023, Disponível em: <https://www.mdpi.com/2076-3417/13/20/11378>.

ZIEGLER, C.-N. et al. **Improving recommendation lists through topic diversification**. In: **Proceedings of the 14th International Conference on World Wide Web (WWW)**. ACM, 2005. 2005. p. 22–32. Acesso em: 01 jul. 2026. Disponível em: <https://doi.org/10.1145/1060745.1060754>.