UNIVERSIDADE ESTADUAL DO PIAUÍ
CENTRO DE TECNOLOGIA E URBANISMO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

João Manoel Carvalho Borges Cunha Oliveira

TRANSFORMERS EM SISTEMAS DE RECOMENDAÇÃO: UM ESTUDO COMPARATIVO COM MODELOS DE MACHINE LEARNING

TERESINA
2025

João Manoel Carvalho Borges Cunha Oliveira

TRANSFORMERS EM SISTEMAS DE RECOMENDAÇÃO: UM ESTUDO COMPARATIVO COM MODELOS DE MACHINE LEARNING

Monografia de Trabalho de Conclusão de Curso apresentado na Universidade Estadual do Piauí – UESPI como parte dos requisitos para conclusão do Curso de Bacharelado em Ciência da Computação.

Orientador: Pr. Dr. Sérgio Barros de Sousa

TERESINA

2025

O48t Oliveira, Joao Manoel Carvalho Borges Cunha.
Transformers em sistemas de recomendação: um estudo comparativo com modelos de machine learning / Joao Manoel Carvalho Borges Cunha Oliveira. - 2025.
47f.: il.

Monografia (graduação) - Universidade Estadual do Piauí - UESPI, Bacharelado em Ciências da Computação, Campus Poeta Torquato Neto, 2025.
"Orientador: Prof. Dr. Sérgio Barros de Sousa".

1. Transformers. 2. Sistemas de Recomendação. 3. Machine Learning. 4. MovieLens. 5. Avaliação de Ranking. I. Sousa, Sérgio Barros de . II. Título.

CDD 006.31

Ficha elaborada pelo Serviço de Catalogação da Biblioteca da UESPI
JOSÉ EDIMAR LOPES DE SOUSA JÚNIOR (Bibliotecário) CRB-3ª/1512

TRANSFORMERS EM SISTEMAS DE RECOMENDAÇÃO: UM ESTUDO COMPARATIVO COM MODELOS DE MACHINE LEARNING

João Manoel Carvalho Borges Cunha Oliveira

Monografia de Trabalho de Conclusão de Curso apresentado na Universidade Estadual do Piauí – UESPI como parte dos requisitos para conclusão do Curso de Bacharelado em Ciência da Computação.

Pr. Dr. Sérgio Barros de Sousa , Dsc.
Orientador

Nota da Banca Examinadora: 9,00

Banca Examinadora:

Documento assinado digitalmente
gov.br
SERGIO BARROS DE SOUSA
Data: 07/07/2025 18:37:30-0300
Verifique em https://validar.iti.gov.br

Pr. Dr. Sérgio Barros de Sousa , Dsc.
Presidente

Documento assinado digitalmente
gov.br
MARCUS VINICIUS RIBEIRO DE CARVALHO
Data: 07/07/2025 15:38:45-0300
Verifique em https://validar.iti.gov.br

Pr. Dr. Marcus Vinícius Carvalho, Dsc.
Membro

Documento assinado digitalmente
gov.br
Reginaldo Rodrigues das Gracas
Data: 07/07/2025 15:45:06-0300
Verifique em https://validar.iti.gov.br

Pr. Me. Reginaldo Rodrigues das Graças, Dsc.
Membro

Dedico este trabalho aos meus pais,
meus maiores apoiadores e inspirações de vida.

“A única jornada impossível é aquela que você nunca começa.“
(Tony Robbins)

.

RESUMO

Os Sistemas de Recomendação (SR) são fundamentais para personalizar a experiência do usuário em plataformas digitais, mas enfrentam desafios como *cold-start* e esparsidade de dados. Este trabalho realiza uma análise comparativa da capacidade de ranking entre um modelo baseado na arquitetura *Transformers* e algoritmos de *Machine Learning* (ML) estabelecidos — SVD++, ItemKNN e um modelo baseado em Popularidade — para a recomendação de filmes. Utilizando os datasets MovieLens (100K e 1M), o estudo avalia a eficácia dos modelos em gerar rankings relevantes, especialmente em cenários críticos como cold-start. As métricas de avaliação focam na qualidade do ranking, como NDCG, Precisão@k e Recall@k, complementadas por métricas de erro (*Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE)) dos scores que geram esses rankings. A metodologia emprega validação cruzada k-fold (k=10). Os resultados indicam que o modelo Transformers, embora apresente maior *recall* em alguns cenários, foi consistentemente superado pelos modelos tradicionais nas métricas de ordenação, como NDCG e Precisão@k. No entanto, apresentou maior robustez em situações de *cold-start* para itens com poucas interações. Conclui-se que, apesar de ainda não superar métodos consolidados em todas as métricas, o *Transformers* mostra potencial para ser explorado em contextos específicos de recomendação, contribuindo para a literatura com insights sobre a aplicabilidade e o desempenho de modelos de *Deep Learning*.

Palavras-chaves: Transformers, Machine Learning, Sistemas de Recomendação, Avaliação de Ranking, MovieLens.

ABSTRACT

Recommender Systems (RS) are essential for personalizing the user experience on digital platforms, but they face challenges such as cold-start and data sparsity. This work performs a comparative analysis of the ranking capability between a model based on the Transformers architecture and established *Machine Learning* (ML) algorithms — SVD++, ItemKNN, and a Popularity-based model — for movie recommendation. Using the MovieLens datasets (100K and 1M), the study evaluates the effectiveness of the models in generating relevant rankings, especially in critical scenarios such as cold-start. The evaluation metrics focus on the quality of the ranking, such as Normalized Discounted Cumulative Gain (NDCG), Precision@k, and Recall@k, complemented by error metrics (*Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE)) of the scores that generate these rankings. The methodology employs k-fold cross-validation (k=10). The results indicate that the Transformers model, although presenting higher recall in some scenarios, was consistently outperformed by traditional models in ranking metrics, such as NDCG and Precision@k. However, they presented greater robustness in cold-start situations for items with few interactions. It is concluded that, although it does not yet surpass consolidated methods in all metrics, the Transformers shows potential to be explored in specific recommendation contexts, contributing to the literature with insights on the applicability and performance of Deep Learning models.

Keywords: Transformers, Machine Learning, Recommendation Systems, Ranking Evaluation, MovieLens.

LISTA DE ABREVIATURAS E SIGLAS

BERT4Rec	BERT *for Sequential Recommendation*
DCG	*Discounted Cumulative Gain*
GNNs	Redes Neurais Grafônicas
IDCG	*Ideal Discounted Cumulative Gain*
KNN	*K-Nearest Neighbors*
LLMs	Modelos de Linguagem Grandes
MAE	*Mean Absolute Error*
ML	*Machine Learning*
NDCG	*Normalized Discounted Cumulative Gain*
RMSE	*Root Mean Squared Error*
RNNs	Redes Neurais Recorrentes
RS	Recommender Systems
SASRec	*Sequential Recommendation with Self-Attentive Networks*
SR	Sistemas de Recomendação

LISTA DE QUADROS

SUMÁRIO

1 INTRODUÇÃO

Os Sistemas de Recomendação (SR) têm se demonstrado cada vez mais importantes para diversos setores da sociedade, como o entretenimento e comércio eletrônico, impulsionados pela crescente quantidade de dados disponíveis nas plataformas digitais e pelo desejo contínuo de personalizar a experiência do usuário, facilitando com que o mesmo encontre conteúdos e produtos relevantes. Nesse contexto, os SR são utilizados por grandes empresas, como Netflix e Amazon, as quais utilizam informações, como o histórico no site, para fazer recomendações que se assemelham aos gostos do usuário (Segaran, 2007), colaborando para manter o engajamento e estimular o consumo contínuo na plataforma.

Porém, a escolha e o desempenho do modelo de ML que será empregado nesses sistemas afeta diretamente em sua eficácia em trazer uma boa experiência para o usuário, pois ele precisará lidar com desafios complexos, como a necessidade de maior personalização, sua escalabilidade e a escassez de dados (*cold-start*). Tendo a capacidade de um modelo em gerar um *ranking* de itens verdadeiramente relevante para cada usuário como um diferencial crítico.

Graças aos avanços nos métodos de DP, o modelo *Transformers* oferece uma abordagem diferenciada, onde apresenta um mecanismo chamado de “atenção” para processar tanto dados sequenciais e contextuais, diferente dos modelos de ML tradicionais como o *K-Nearest Neighbors* (KNN) item-*based*, que se baseia na similaridade entre itens, ou modelos de fatoração de matrizes como o SVD++, o *Transformers* têm o potencial de capturar dependências de longo alcance e nuances contextuais nas sequências de interações do usuário. Essa capacidade pode se traduzir em uma melhor compreensão das preferências do usuário e, consequentemente, em *rankings* de recomendação superiores.

Assim, o presente trabalho realiza uma análise comparativa da eficácia de *ranking* entre um modelo baseado na arquitetura *Transformers* e algoritmos estabelecidos de ML — especificamente SVD++, ItemKNN (item-*based*) e um modelo de Popularidade — no contexto de sistemas de recomendação de filmes. Utilizando os *datasets MovieLens* (100K e 1M), o estudo avalia a capacidade dos modelos em gerar rankings relevantes, com atenção a cenários críticos como o *cold-start*. Baseado na avaliação dessa análise comparativa, espera-se fornecer *insights* sobre a aplicabilidade e o desempenho de modelos de *Deep Learning* na tarefa de otimização de *ranking* em SR, visando avanços na precisão e relevância das recomendações.

1.1 Objetivos

1.1.1 *Objetivo geral*

Comparar a eficácia de *ranking* do modelo *Transformers* com algoritmos de *Machine Learning* estabelecidos (SVD++, ItemKNN e um modelo de Popularidade) em sistemas de recomendação de filmes, utilizando os datasets MovieLens (100K e 1M).

1.1.2 *Objetivos específicos*

- Avaliar os modelos utilizando métricas de qualidade de *ranking*, como NDCG@k, Precisão@k e Recall@k, além de analisar métricas de erro (RMSE, MAE) dos scores preditos que dão origem aos *rankings*, para caracterizar o comportamento dos modelos.
- Avaliar a eficiência computacional (tempo de treinamento) de cada modelo.
- Identificar cenários (ex.: *cold-start*, aproveitamento de dados sequenciais) onde o modelo *Transformers* demonstra vantagens ou desvantagens em relação aos demais na geração de *rankings* relevantes.

Para alcançar esses objetivos, se torna fundamental o referencial teórico que fundamenta a aplicação da análise comparativa de desempenho entre os modelos.

1.2 Justificativa

A experiência do usuário é algo de suma importância para as plataformas digitais, as quais estão fortemente presentes na sociedade. Além disso, outros fatores, como a escalabilidade do sistema e a qualidade do *ranking* dos resultados, dependem diretamente da escolha do modelo certo para o sistema de recomendação da plataforma. Por exemplo, um modelo que lida bem com pequenos volumes de dados pode não ser adequado para grandes volumes, da mesma forma que modelos que incorporam aprendizado contínuo podem se adaptar melhor ao usuário e as suas tendências de consumo que mudam com o tempo.

Representando uma nova alternativa para os modelos que já são mais consolidados no mercado, o modelo *Transformers* apresentam uma abordagem que permite capturar interações complexas e contextuais, mas trazem desafios em termos de custo computacional por causa do mecanismo de ”atenção”(Nauen et al., 2023). Apesar dos avanços observados em trabalhos como (Kang; McAuley, 2018) e (Sun et al., 2019), ainda há uma lacuna na comparação direta entre *Transformers* e métodos clássicos com foco em métricas de *ranking*. Este trabalho busca contribuir preenchendo essa

lacuna, especialmente avaliando cenários críticos como o *cold-start* e contrastando o custo computacional dos modelos. Espera-se que esses resultados ofereçam base sólida para decisões práticas sobre a adoção de modelos avançados em SR.

1.3 Estrutura dos capítulos

O presente trabalho está estruturado como se segue abaixo.

O Capítulo 2 apresenta o referencial teórico, com conceitos sobre SR, discutindo abordagens como a Filtragem Baseada em Conteúdo, Filtragem Colaborativa e Filtragem Híbrida, assim como os modelos que serão utilizados para a análise comparativa (*Transformers*, SVD++, ItemKNN, Popularidade), explorando também os desafios e as métricas de avaliação de *ranking* presentes no estudo. Além disso, apresenta trabalhos relacionados, os quais foram de extrema importância para o desenvolvimento dos conceitos discutidos ao longo deste capítulo.

No Capítulo 3, é descrita a metodologia utilizada nos experimentos, abordando detalhadamente as bibliotecas, parâmetros, base de dados e a técnica de validação dos resultados utilizadas para o estudo.

Em seguida, o Capítulo 4 discute os resultados e suas implicações, avaliando os experimentos junto de suas métricas, as quais vão proporcionar uma visão de como cada modelo se comporta na tarefa de recomendação e o impacto que sua aplicação terá na qualidade do *ranking*.

Ao final, o Capítulo 5 apresentará as conclusões acerca dos resultados e recomendações para trabalhos futuros.

2 REFERENCIAL TEÓRICO

2.1 Sistemas de Recomendação

De acordo com (Sharma; Gera, 2013), esses sistemas são capazes de sugerir itens que possam ser de interesse do usuário, utilizando dados históricos, preferências e comportamentos anteriores para gerar recomendações personalizadas. Esses itens podem incluir produtos, serviços, filmes, músicas, entre outros, e a análise pode se basear em compras realizadas, avaliações fornecidas, tempo de visualização ou cliques em determinados conteúdos.

A importância dos SR se dá principalmente pelo alto volume de informação presente em ambientes digitais, como plataformas de *streaming* e redes sociais. Segundo (Ricci; Rokach; Shapira, 2010b), os SR podem ser agrupados em três categorias principais: Filtragem Baseada em Conteúdo, Filtragem Colaborativa e Abordagens Híbridas. A escolha entre essas abordagens depende de fatores como o domínio da aplicação, o tipo e a quantidade de dados disponíveis, bem como as características do público-alvo.

Atualmente, com os avanços em aprendizado de máquina, tem-se observado a incorporação de técnicas mais sofisticadas, como redes neurais profundas e modelos baseados em atenção (Zhang et al., 2019), que potencializam a precisão e a personalização das recomendações. O objetivo central permanece o mesmo: fornecer sugestões que agreguem valor ao usuário e melhorem sua experiência com o sistema.

2.1.1 Filtragem Baseada em Conteúdo

A Filtragem Baseada em Conteúdo é uma abordagem que recomenda itens similares àqueles que o usuário demonstrou interesse no passado, com base nas características ou atributos dos próprios itens (Pazzani; Billsus, 2007). Por exemplo, se um usuário assistiu a vários filmes de ficção científica com um determinado ator, o sistema baseado em conteúdo pode recomendar outros filmes de ficção científica ou filmes com o mesmo ator, independentemente do que outros usuários assistiram.

A essência dessa abordagem reside na criação de perfis para usuários e itens. O perfil de um item é construído a partir de seus atributos, enquanto o perfil de um usuário é derivado dos atributos dos itens que ele avaliou positivamente ou interagiu. A recomendação é então gerada calculando a similaridade entre o perfil do usuário e os perfis dos itens ainda não vistos por ele (Lops; Gemmis; Semeraro, 2011). Dados como seleção e aquisição de itens (Reategui; Cazella, 2005), por exemplo, são alguns

dos utilizados para construir o interesse do usuário.

Um dos pontos fortes dessa abordagem é a personalização intrínseca às recomendações, uma vez que cada sugestão é baseada exclusivamente no histórico individual de cada usuário. Contudo, essa característica também representa uma limitação, já que o sistema pode apresentar uma falta de diversidade e redundância nas recomendações, oferecendo sempre itens muito semelhantes aos já consumidos, fenômeno conhecido como *overspecialization* (Lops; Gemmis; Semeraro, 2011).

Outro desafio enfrentado é a incapacidade do sistema de recomendar itens que não compartilham características explícitas com o histórico do usuário, o que compromete a descoberta de novos conteúdos (Pazzani; Billsus, 2007) . Para lidar com essas limitações, estratégias como a incorporação de aprendizado de representação e o uso de *embeddings* semânticos têm sido propostas (Zhang et al., 2019), permitindo identificar similaridades mais profundas entre os itens.

2.1.2 Filtragem Colaborativa

A Filtragem Colaborativa é uma das abordagens mais utilizadas em SR e se baseia na interação entre os usuários e os itens. A lógica é a de que usuários com comportamentos semelhantes no passado tendem a manter preferências semelhantes no futuro. Percebe-se então que é uma abordagem em que sua base está na troca de experiências entre os usuários com gostos semelhantes (Reategui; Cazella, 2005). O estado da arte em Filtragem Colaborativa abrange desde métodos baseados em vizinhança até modelos de fatoração de matrizes e, mais recentemente, abordagens baseadas em aprendizado profundo (Su; Khoshgoftaar, 2009).

Segundo (Su; Khoshgoftaar, 2009), existem duas grandes categorias: baseada em usuários e em itens. A técnica baseada em usuários encontra outros com gostos similares e recomenda itens que esses usuários gostaram, mas que o usuário ativo ainda não viu. Já a técnica baseada em itens identifica itens que são similares àqueles que o usuário ativo gostou no passado e os recomenda (Sarwar et al., 2001). Ambas as estratégias podem ser implementadas com técnicas de vizinhança ou de aprendizado latente, como fatoração de matrizes (Koren; Bell; Volinsky, 2009).

Apesar da sua eficácia, a Filtragem Colaborativa sofre com problemas significativos, como esparsidade e o *cold-start*. Tais desafios têm motivado o uso de métodos. Pesquisas recentes exploram o uso de *embeddings* de grafos e técnicas de aprendizado auto-supervisionado para capturar padrões complexos em dados de interação esparsos (Wang et al., 2019).

2.1.3 Filtragem Híbrida

Essa abordagem combina as duas abordagens anteriores, visando superar suas limitações, como o *cold-start* para a Filtragem Colaborativa (Alam et al., 2021) e a limitação das recomendações aos itens já consumidos para a Filtragem Baseada em Conteúdo. A Filtragem Híbrida representa o estado da arte em SR ao buscar alavancar os pontos fortes de múltiplas técnicas (Burke, 2002) , utilizando o melhor delas, como bons resultados para usuários incomuns e recomendações relacionadas diretamente ao histórico (Reategui; Cazella, 2005), para mitigar as fraquezas das abordagens individuais.

Os sistemas híbridos podem ser implementados de diferentes maneiras: por meio de combinação de pontuações, modelo unificado ou abordagem sequencial, onde uma técnica alimenta a outra (Reategui; Cazella, 2005). Um sistema pode, por exemplo, usar a Filtragem Colaborativa para gerar uma recomendação inicial e, em seguida, aplicar um filtro baseado em conteúdo para refinar a sugestão.

A utilização de abordagens híbridas tem ganhado ainda mais força com a ascensão de modelos de *Deep Learning*, que permitem a criação de arquiteturas mais complexas e robustas. Segundo (Zhang et al., 2019), redes neurais híbridas podem aprender representações conjuntas de usuários e itens, explorando tanto os padrões de interação quanto os atributos semânticos dos dados. Essa integração tem se mostrado eficaz em cenários com alta variabilidade e esparsidade, além de oferecer uma experiência de recomendação mais precisa e personalizada.

Pesquisas recentes exploram arquiteturas neurais complexas que combinam módulos de Filtragem Colaborativa e Baseada em Conteúdo, como o *Deep*FM, para capturar interações de baixo e alto nível entre features (Guo et al., 2017). Outras técnicas incluem a construção de um modelo único que incorpora características de ambas as abordagens, como modelos de fatoração de matrizes que integram atributos de itens e usuários (Zhang et al., 2019).

2.2 Desafios em *Machine Learning*

2.2.1 *Cold-start*

O problema de *cold-start* ocorre quando as informações sobre um novo usuário ou um novo item que o SR possui não são suficientes, dificultando a geração de recomendações personalizadas. Isso impacta negativamente a experiência do usuário, especialmente em fases iniciais de uso da plataforma. Esse cenário é comum em sistemas que dependem fortemente de interações prévias para realizar predições (Schein et al., 2002). O *cold-start* é um desafio inerente a muitos sistemas baseados em dados

de interação e representa um obstáculo significativo para a adoção inicial de novos usuários e a promoção de novos itens (Park; Chu, 2009).

Uma solução comum para o *cold-start* de usuários é o uso de perfis demográficos ou questionários iniciais que ajudam a construir uma base mínima de preferências. Segundo (Schein et al., 2002), a combinação de modelos baseados em conteúdo com informações demográficas permite gerar recomendações iniciais mais relevantes, mesmo com poucas ou nenhuma interação anterior do usuário. Para o *cold-start* de itens, estratégias incluem a análise de conteúdo textual, metadados, imagens ou até mesmo informações de fornecedores para criar representações iniciais dos itens (Park; Chu, 2009).

Além disso, técnicas de *transfer learning* e *meta-learning* também têm sido propostas como alternativas emergentes, especialmente em contextos com escassez extrema de dados (Vartak et al., 2017). Elas permitem aproveitar conhecimentos adquiridos em domínios semelhantes ou aprender a adaptar-se rapidamente a novos usuários ou itens com poucos exemplos. Pesquisas recentes exploram o uso de redes neurais grafônicas para incorporar informações de grafos externos para enriquecer as representações de usuários e itens frios (Fan et al., 2019).

2.2.2 Esparsidade

A esparsidade é um dos principais desafios enfrentados por sistemas de recomendação. Ela ocorre quando há uma quantidade limitada de interações entre usuários e itens, resultando em uma matriz de interação altamente esparsa, o que dificulta a identificação de padrões confiáveis de preferência (Bobadilla et al., 2013). Isso é especialmente problemático em plataformas com muitos usuários e itens, como serviços de *streaming* ou *e-commerce*. Este fenômeno ocorre porque a maioria dos usuários interage apenas com uma pequena fração do total de itens disponíveis, resultando em matrizes de interação usuário-item predominantemente preenchidas com valores ausentes (Ricci; Rokach; Shapira, 2010a).

Uma consequência direta da esparsidade é a dificuldade em realizar inferências precisas sobre as preferências do usuário, principalmente para itens menos populares. Métodos tradicionais de Filtragem Colaborativa, como KNN ou SVD, tendem a apresentar desempenho inferior nesses cenários (Sarwar et al., 2001). A escassez de dados para muitos itens, conhecida como o problema da cauda longa, dificulta a geração de recomendações relevantes para itens de nicho, impactando diretamente a diversidade e a serendipidade das sugestões. A esparsidade também aumenta a dificuldade de treinar modelos complexos que exigem grandes volumes de dados para aprender padrões significativos.

Para contornar esse problema, técnicas baseadas em fatoração de matrizes, como SVD++, foram amplamente utilizadas, pois conseguem estimar as interações ausentes por meio de representações latentes (Koren; Bell; Volinsky, 2009). Posteriormente, modelos baseados em Redes Neurais Profundas e *Autoencoders* Variacionais passaram a ser aplicados, permitindo a extração de padrões mais complexos em dados altamente esparsos (Sedhain et al., 2015).

Pesquisas mais recentes têm explorado o potencial das Redes Neurais Grafônicas (GNNs), como o *Light*GCN, que modelam as interações usuário-item como um grafo, capturando relações de alta ordem e aliviando os efeitos da esparsidade (He et al., 2020). Além disso, técnicas como o *Dropout*Net têm sido propostas para lidar com o problema de *cold-start*, intrinsecamente ligado à esparsidade, especialmente para novos usuários ou itens (Volkovs; Yu; Poutanen, 2017).

2.3 Métricas de Avaliação

2.3.1 *Root Mean Squared Error* (RMSE)

O RMSE é uma métrica de erro de predição amplamente utilizada, especialmente em tarefas de predição de rating, onde o objetivo é estimar o valor numérico que um usuário daria a um item (Shani; Gunawardana, 2011). Num contexto de SR de filmes, é esperado que recomendações que prejudicassem a experiência do usuário sejam evitadas o máximo possível, e o RMSE carrega um diferencial de penalizar erros grandes de forma mais severa, já que eleva o erro ao quadrado antes da média, o que amplia o impacto de desvios elevados (Koren; Bell; Volinsky, 2009).

Em avaliações comparativas de modelos, o RMSE é frequentemente utilizado como critério principal em competições e benchmarks, como a *Netflix Prize* (Bennett; Lanning, 2007), o que reforça sua relevância histórica e prática na área de sistemas de recomendação. A popularidade do RMSE se deve, em parte, à sua interpretabilidade e à sua relação com a otimização por mínimos quadrados, comum em muitos algoritmos de aprendizado de máquina (James et al., 2013). A sua vasta aplicação estende-se para além dos SR, sendo uma métrica padrão para avaliar modelos de regressão em diversas áreas da ciência de dados e aprendizado de máquina (James et al., 2013).

A característica de penalizar erros maiores mais significativamente torna o RMSE particularmente útil em cenários onde a magnitude do erro é crítica. Por exemplo, prever um *rating* de 1 estrela para um filme que o usuário avaliaria com 5 estrelas é um erro consideravelmente pior do que prever 4 estrelas, e o RMSE reflete essa diferença de forma mais evidente do que métricas lineares. Essa sensibilidade, no entanto, também significa que o RMSE pode ser desproporcionalmente influenciado por um pequeno

número de previsões muito ruins, o que deve ser considerado na interpretação dos resultados (Chai; Draxler, 2014). No geral, um RMSE menor indica que o modelo está sendo mais preciso em suas predições, ou seja, quanto mais próximo de 0 o RMSE for, melhor.

2.3.2 *Mean Absolute Error* (MAE)

O MAE mede a média das diferenças absolutas entre as predições do modelo e os valores reais, o que proporciona uma métrica intuitiva do erro em termos absolutos. Essa métrica, assim como o RMSE, é uma métrica de erro de predição, mas sua natureza linear o torna mais robusto a valores extremos, entrega uma perspectiva diferente sobre o desempenho do modelo (Shani; Gunawardana, 2011), pois é menos sensível a *outliers* e fornece uma visão clara e direta do erro médio nas predições (Willmott; Matsuura, 2005).

Diversos estudos mostram que modelos que apresentam RMSE e MAE baixos tendem a ser bem avaliados pelos usuários, mas há casos em que o modelo com menor RMSE não é o mais bem avaliado na prática, destacando a importância da análise complementar de métricas (Gunawardana; Shani, 2009). A escolha entre MAE e RMSE muitas vezes depende da importância relativa atribuída a erros grandes versus erros pequenos no contexto específico da aplicação. No geral, um MAE menor indica que o modelo está acertando mais de suas predições, ou seja, quanto mais próximo de 0 o MAE for, melhor.

A sua robustez a *outliers* é uma de suas principais vantagens, pois cada erro contribui para a média de forma proporcional à sua magnitude, sem o efeito de amplificação dos erros quadrados presente no RMSE. Isso significa que o MAE pode oferecer uma representação mais estável do desempenho médio do modelo quando o conjunto de dados contém valores anômalos ou previsões excepcionalmente imprecisas (Hodson, 2022). Além disso, a interpretabilidade direta do MAE facilita a comunicação dos resultados para um público mais amplo, que pode não estar familiarizado com as nuances de métricas baseadas em erros quadráticos (Legates; Jr, 1999).

Em avaliações comparativas de sistemas de recomendação, o MAE é frequentemente reportado junto ao RMSE para fornecer uma imagem mais completa da performance preditiva (Herlocker et al., 2004). Estudos que realizam *benchmarking* de diferentes *frameworks* de recomendação também costumam incluir o MAE como uma métrica fundamental (Said; Bellogín, 2014).

2.3.3 *Normalized Discounted Cumulative Gain* (NDCG@k)

O NDCG@k é uma métrica que avalia a qualidade de *ranking* em sistemas de recomendação, especialmente quando a relevância é graduada (Järvelin; Kekäläinen, 2002). Ela é uma métrica considerada padrão para avaliar a qualidade de listas ordenadas, pois leva em conta tanto a relevância dos itens quanto suas posições na lista de recomendação (Järvelin; Kekäläinen, 2002).

Essa métrica é particularmente relevante em cenários de recomendação *top*-N, onde a ordem dos itens apresentados é fundamental (Cremonesi; Koren; Turrin, 2010). Neste trabalho, os *ratings* originais dos usuários foram utilizados como os scores de relevância para o cálculo do *Normalized Discounted Cumulative Gain* (NDCG).

A fórmula do NDCG@k combina três componentes essenciais: ganho cumulativo, ganho descontado e normalização. Esse último componente compensa a variabilidade no número de itens relevantes por usuário (Wang et al., 2013). O componente de ”ganho descontado”é crucial, pois atribui maior peso aos itens relevantes que aparecem nas primeiras posições da lista de recomendação, refletindo o comportamento do usuário, que tende a prestar mais atenção ao topo da lista (Croft; Metzler; Strohman, 2010). O ganho é descontado logaritmicamente com a posição do item, refletindo a menor probabilidade de o usuário examinar itens em posições inferiores.

Para calcular o *Discounted Cumulative Gain* (DCG), somam-se esses ganhos descontados para os k primeiros itens. A normalização é realizada dividindo o DCG pelo *Ideal Discounted Cumulative Gain* (IDCG), que representa o DCG máximo possível para aquele usuário, obtido ao ordenar perfeitamente os itens por relevância. Essa normalização garante que o NDCG@k varie entre 0 e 1, facilitando comparações entre diferentes usuários e *queries* (Schütze; Manning; Raghavan, 2008).

A eficácia do NDCG@k em tarefas de recomendação *top*-N, onde o objetivo é apresentar uma lista curta dos itens mais promissores, tem sido consistentemente demonstrada em diversos estudos (Cremonesi; Koren; Turrin, 2010). Modelos de aprendizado profundo, como os baseados em *autoencoders* variacionais para filtragem colaborativa, frequentemente utilizam NDCG@k como uma das principais métricas para avaliar a qualidade das recomendações de *ranking* geradas (Liang et al., 2018). Pesquisas recentes continuam a explorar variações do NDCG ou métricas complementares para capturar nuances da qualidade do *ranking* (Zhou et al., 2010).

2.3.4 Precisão@k (*Precision*@k)

A Precisão@k é uma métrica de avaliação fundamental em sistemas de recomendação e na recuperação de informação, projetada para medir a proporção de itens relevantes dentro de um subconjunto do topo da lista de resultados rankeados. Essa

métrica quantifica quantos dos k itens mais bem posicionados na lista de recomendação são de fato pertinentes ao usuário (Shani; Gunawardana, 2011). Por exemplo, se um sistema recomenda 10 itens (k=10) e 6 deles são considerados relevantes pelo usuário, a Precisão@10 seria de 0.6.

Essa métrica é frequentemente preferida pela sua interpretabilidade direta e foco na relevância dos itens que são mais proeminentes para o usuário, ou seja, aqueles no topo da lista (Ricci; Rokach; Shapira, 2010b). A seleção de um k apropriado é crucial e geralmente alinhada com o número de itens que um usuário realisticamente visualizaria ou consideraria em uma única interação.

Embora a Precisão@k forneça uma medida intuitiva da utilidade imediata das recomendações, ela não considera a posição dos itens relevantes dentro do *top*-k, nem o número total de itens relevantes que poderiam ter sido recomendados (Herlocker et al., 2004). Dito isso, sua utilização em conjunto de outras métricas, como Recall@k e NDCG@k, ajuda fornecer uma visão mais ampla sobre o desempenho do modelo. Desafios na avaliação de métricas *top*-K em grande escala, especialmente com amostragem de itens negativos, são áreas de pesquisa ativa (Krichene; Rendle, 2020).

2.3.5 *Recall*@k

Diferentemente da Precisão@k, que foca na exatidão dos itens recomendados, o Recall@k avalia a lista como um todo, ou seja, qual proporção do total de itens que o usuário consideraria relevantes que efetivamente foi mostrada entre os k primeiros itens (Sarwar et al., 2001). Um valor de *Recall*@k mais alto indica que o sistema é mais eficiente em não deixar de fora itens que seriam do interesse do usuário.

Um *Recall*@10, por exemplo, mede quantos dos itens relevantes para um usuário estão presentes entre as 10 primeiras recomendações. Segundo (Schütze; Manning; Raghavan, 2008), aumentar o número de itens recomendados pode levar a um *recall* maior, mas potencialmente a uma precisão menor, pois itens em que o usuário teria menos interesse podem ser incluídos para garantir a cobertura dos relevantes.

O *Recall*@k é particularmente importante em cenários onde é crucial não perder itens relevantes, mesmo que isso signifique incluir alguns itens menos relevantes na lista de recomendações. Por exemplo, em sistemas de alerta médico ou na detecção de fraudes, um alto *recall* é fundamental (Powers, 2020). A interdependência entre Precisão@k e *Recall*@k é um aspecto central na avaliação de SR.

Recentemente, com o advento de Modelos de Linguagem Grandes (LLMs) aplicados à recomendação, a avaliação do *recall* continua sendo uma métrica chave para medir a cobertura e a eficácia desses novos modelos em identificar o espectro de interesses do usuário (Wu et al., 2024).

2.4 *MovieLens*

Os *datasets MovieLens*, mantidos pelo *GroupLens Research Lab* da Universidade de Minnesota, são amplamente utilizados como *benchmarks* para algoritmos de recomendação devido à sua disponibilidade pública e estrutura bem definida, que tipicamente inclui identificadores de usuário, identificadores de item, avaliações em uma escala numérica e, em algumas versões, *timestamps* das avaliações (Harper; Konstan, 2015). Essa riqueza de informações permite a exploração de diversos aspectos da recomendação, desde a predição de *ratings* até a modelagem de sequências de interações do usuário.

Segundo (Said; Bellogín, 2014), a base de dados de classificação de filmes MovieLens tem grande reconhecimento no meio acadêmico, tendo um total de classificações que passa de 25 milhões de avaliações. A utilização consistente desses *datasets* facilita a comparabilidade entre diferentes algoritmos e abordagens propostas na literatura (Ekstrand et al., 2011).

Tomando por base estudos como os de (Alam et al., 2021) e (Anwar; Uma, 2021), a versão escolhida para o *dataset* foi o MovieLens 100k, o qual contém 100000 classificações de 943 usuários em 1682 filmes. Visando explorar a capacidade de adaptação do modelo *Transformers* com grandes volumes de dados e expandir os trabalhos citados, também será usado a versão MovieLens 1M, o qual contém 1 milhão de classificações de 6040 usuários em 3706 filmes.

A escolha desses *datasets* não é aleatória, pois eles são frequentemente empregados em pesquisas que introduzem novos modelos, incluindo aqueles baseados em *Deep Learning*, devido à sua capacidade de testar a escalabilidade e o desempenho em diferentes densidades de dados e cenários de mundo real (Batmaz et al., 2019). A escolha das versões 100k e 1M permite uma análise comparativa do desempenho dos modelos em diferentes escalas de dados, investigando como a esparsidade e o volume de interações afetam a precisão e a capacidade de generalização das recomendações (Ekstrand et al., 2011).

2.5 Validação Cruzada K-Fold

A técnica de validação cruzada *k-fold* (*k-Fold Cross Validation*) é uma das mais utilizadas para selecionar modelos e estivar erros em classificadores (Anguita et al., 2012). Este método é fundamental para obter uma estimativa mais confiável do desempenho de um modelo em dados não vistos, em comparação com uma única divisão treino-teste, sendo crucial para avaliar a capacidade de generalização. Ao invés de somente separar os dados em conjuntos de treino e teste, ela os divide em k sub-

conjuntos (ou *folds*) e faz o treinamento múltiplas vezes, isso garante que todos os subconjuntos sejam usados como conjunto de teste ao menos uma vez.

Essa abordagem reduz a variabilidade associada a métodos simples de divisão treino-teste e minimiza o risco de *overfitting* (Yates et al., 2022), ou seja, o risco de perder sua capacidade de generalizar para novos dados. O procedimento padrão, conforme estabelecido por (Kohavi et al., 1995), envolve particionar o conjunto de dados original aleatoriamente em k subconjuntos de tamanhos aproximadamente iguais. Desses k *folds*, um é retido como o conjunto de teste (ou validação), e os k-1 *folds* restantes são usados como dados de treinamento.

O processo de treinamento e teste é repetido k vezes, com cada um dos k subconjuntos usado exatamente uma vez como dados de teste. As k métricas de desempenho resultantes das k iterações são então tipicamente agregadas, por exemplo, calculando-se a média, para produzir uma única estimativa da performance do modelo (Kohavi et al., 1995). Essa metodologia é particularmente valiosa quando o tamanho do conjunto de dados é limitado, pois garante que cada observação seja utilizada tanto para treinamento quanto para teste, fornecendo uma avaliação mais robusta e menos enviesada do desempenho do modelo em dados não vistos (Liu; Özsu, 2009). Além disso, a validação cruzada é uma etapa crucial na otimização de hiperparâmetros, onde diferentes configurações de modelo são avaliadas para encontrar aquela que generaliza melhor para dados futuros (Bergstra; Bengio, 2012).

2.6 Modelos

2.6.1 *Modelo de Popularidade*

A premissa central desse modelo é que que os itens mais populares entre todos os usuários são também os mais propensos a serem relevantes para um usuário individual. Em sua forma mais básica, o modelo gera a mesma lista de itens mais populares para todos os usuários, não oferecendo personalização, mas, como demonstrado por (Ji et al., 2020) e observado neste trabalho, pode ser surpreendentemente eficaz.

Embora robusto e computacionalmente barato, o modelo de popularidade demonstra uma incapacidade de descobrir itens novos ou de *long-tail*, que podem ser de grande interesse para subconjuntos específicos de usuários, mas não para a maioria (Aggarwal et al., 2016),o que também não é o ideal para um SR.

Apesar de sua simplicidade e pouca personalização, os modelos baseados em popularidade servem como um importante *baseline* para avaliar o desempenho de algoritmos de recomendação mais sofisticados. Se um modelo complexo não consegue superar significativamente um *baseline* de popularidade em métricas chave, sua utili-

dade prática e o custo computacional adicional podem ser questionados (Herlocker et al., 2004).

2.6.2 *ItemKNN (K-Nearest Neighbors Item-Based)*

A abordagem ItemKNN é um método de filtragem colaborativa baseado em vizinhança. A premissa fundamental é que se um usuário gostou de um determinado item, ele provavelmente também gostará de itens que são ”vizinhos”ou similares a ele (Sarwar et al., 2001). A similaridade entre dois itens é tipicamente calculada analisando os padrões de co-avaliação de todos os usuários. Por exemplo, se muitos usuários que avaliaram bem o item A também avaliaram bem o item B, então A e B são considerados similares. Esta abordagem foi popularizada por sua eficácia e interpretabilidade, especialmente em cenários com mais itens do que usuários, onde a matriz de similaridade item-item é mais estável.

Uma vez que a matriz de similaridade item-item é construída, a predição de um *rating* para um par usuário-item não avaliado é feita considerando os k itens mais similares ao item em questão que já foram avaliados pelo usuário ativo. Esses *ratings* dos vizinhos são então agregados, muitas vezes de forma ponderada pela similaridade, para gerar a predição (Linden; Smith; York, 2003). Algoritmos como o KNN*Baseline* na biblioteca *surprise* aprimoram essa abordagem ao incorporar também desvios de usuários e itens, o que pode levar a predições mais acuradas.

A construção da matriz de similaridade item-item pode ser computacionalmente intensiva, especialmente com um grande número de itens, mas uma vez calculada, ela pode ser reutilizada para gerar recomendações rapidamente, tornando o ItemKNN adequado para cenários onde a latência da recomendação é crítica (Ricci; Rokach; Shapira, 2010b). A escolha do número de vizinhos também é um hiperparâmetro crucial que afeta o equilíbrio entre precisão e diversidade das recomendações. Sua eficácia em tarefas de recomendação *top*-N também foi extensivamente estudada, focando na geração de listas de itens relevantes (Deshpande; Karypis, 2004) .

2.6.3 *SVD++*

Segundo (Jiao et al., 2019), o modelo SVD++ é amplamente aplicado em SR, sendo ele uma versão aprimorada do modelo SVD tradicional. Essa abordagem estendida da fatoração de matrizes se destaca por incorporar informações adicionais, como interações implícitas dos usuários com os itens, que não são consideradas no SVD tradicional. Essas interações incluem comportamentos como cliques, visualizações ou quaisquer outras formas de engajamento indireto com os itens, mesmo sem avaliações explícitas. O SVD++ ganhou notoriedade, especialmente após o Netflix *Prize*, por sua

capacidade de melhorar significativamente a acurácia das predições (Koren, 2008).

A incorporação do feedback implícito no SVD++ é realizada ao modelar o perfil do usuário não apenas com base nos itens que ele avaliou explicitamente, mas também considerando o conjunto de todos os itens com os quais ele interagiu implicitamente. Isso é feito adicionando um termo ao modelo que representa a soma dos vetores latentes dos itens com os quais o usuário teve alguma interação implícita, normalizada pelo número dessas interações (Koren, 2008). Essa extensão permite que o modelo capture uma visão mais holística das preferências do usuário, potencialmente melhorando a qualidade das recomendações, especialmente para usuários com poucas avaliações explícitas ou para itens que raramente são avaliados explicitamente, mas frequentemente consumidos (Hu; Koren; Volinsky, 2008).

No entanto, essa complexidade adicional traz o ponto negativo significativo de ter um tempo de treinamento maior (Jiao et al., 2019). O modelo SVD++ exige maior poder computacional e tempo para processar as informações adicionais, o que pode ser um desafio em sistemas de grande escala ou em cenários que demandam atualizações frequentes das recomendações.

Apesar disso, a melhoria na acurácia frequentemente justifica seu uso. *Frameworks* modernos de recomendação podem integrar características de modelos como SVD++ em arquiteturas mais complexas e híbridas, demonstrando a influência duradoura de suas ideias na modelagem de preferências do usuário (Wu et al., 2022).

2.6.4 *Transformers*

Introduzido por (Vaswani et al., 2017), a arquitetura do modelo *Transformers* é baseada principalmente no mecanismo de atenção. Com esse mecanismo, o modelo consegue enfrentar melhor os desafios em que outros modelos têm dificuldade, como o processamento sequencial de texto e o custo computacional de obter relacionamentos notáveis entre palavras em uma frase (Nikzad-Khasmakhi et al., 2021).

No contexto de sistemas de recomendação, os *Transformers* podem ser adaptados para processar sequências de interações do usuário (por exemplo, filmes assistidos em ordem). Ao aprender a partir dessas sequências, o modelo pode prever um score de relevância (como um *rating*) para itens futuros ou itens mascarados dentro da sequência. Esses scores são então utilizados para gerar um *ranking* de recomendações. A capacidade do mecanismo de atenção de ponderar a importância de diferentes itens no histórico do usuário é o que o torna promissor para gerar recomendações contextuais e personalizadas, potencialmente levando a *rankings* de maior qualidade.

De acordo com (Liang, 2024), o modelo *Transformers*, aplicado a recomendações multimodais, é capaz de processar dados provenientes de diversas modalidades,

como texto, imagens e áudio. Ele extrai correlações potenciais entre essas modalidades, gerando representações de recursos mais ricas para o sistema de recomendação.

O estado da técnica em recomendação sequencial tem sido amplamente influenciado pelos *Transformers*, com modelos como SASRec (Kang; McAuley, 2018) e BERT4Rec (Sun et al., 2019) demonstrando resultados superiores em capturar dependências de longo alcance nas sequências de interação. Pesquisas atuais exploram a eficiência e a escalabilidade dos *Transformers* para lidar com sequências muito longas, a incorporação de informações contextuais e a aplicação em diferentes domínios de recomendação (Zhou et al., 2020).

2.7 Trabalho Relacionados

A aplicação de modelos sequenciais para capturar a dinâmica das interações do usuário tem sido uma área de intenso desenvolvimento. No entanto, a arquitetura *Transformers*, introduzida em (Vaswani et al., 2017), revolucionou o processamento de sequências, principalmente em Processamento de Linguagem Natural, devido ao seu eficaz mecanismo de auto-atenção que permite modelar dependências de longo alcance de forma mais eficiente que as Redes Neurais Recorrentes (RNNs). Inspirados por esse sucesso, pesquisadores adaptaram os *Transformers* para SR. Modelos como *Sequential Recommendation with Self-Attentive Networks* (SASRec), introduzido em (Kang; McAuley, 2018), e BERT *for Sequential Recommendation* (BERT4Rec), introduzido em (Sun et al., 2019), demonstraram a capacidade dos *Transformers* de modelar as dependências entre itens no histórico de interações do usuário. BERT4Rec, em particular, utiliza uma abordagem análoga ao BERT, mascarando itens na sequência de interações e treinando o modelo para prevê-los. Essa técnica de aprendizado auto-supervisionado a partir de sequências mascaradas serviu como uma inspiração fundamental para a adaptação da tarefa de predição de *ratings* de itens mascarados no modelo *Transformers* desenvolvido neste trabalho, cujos scores preditos são subsequentemente utilizados para gerar *rankings*.

Para contextualizar o desempenho do modelo *Transformers*, este estudo o compara com algoritmos de filtragem colaborativa conhecidos e utilizados como *baselines*. O SVD++, uma extensão do SVD que incorpora tanto *feedback* explícito quanto implícito, além de vieses de usuário e item (Koren, 2008), é um método de fatoração de matrizes reconhecido por sua precisão e robustez em diversos cenários de recomendação. Da mesma forma, o ItemKNN, popularizado por trabalhos como os de (Sarwar et al., 2001) e aplicada com sucesso em sistemas comerciais de grande escala como o da Amazon.com (Linden; Smith; York, 2003), serve como um importante *benchmark*. A variante KNN*Baseline*, empregada neste estudo através da biblioteca scikit-*surprise*,

combina a abordagem de vizinhança com estimativas de *baseline* para aprimorar as predições de *rating*.

Além dos modelos de filtragem colaborativa mais elaborados, este trabalho também inclui um Modelo de Popularidade como *baseline* fundamental. Conforme argumentado em (Ji et al., 2020), a avaliação criteriosa contra um baseline robusto é essencial para justificar a complexidade e o custo computacional de modelos mais avançados. Os achados deste TCC, onde o Modelo de Popularidade demonstrou um desempenho notavelmente competitivo em várias métricas, especialmente no *dataset* MovieLens 1M, reiteram a importância dessas discussões e a força de *baselines* simples bem implementados.

A avaliação da eficácia de sistemas de recomendação, especialmente aqueles focados na apresentação de listas ordenadas de itens, requer métricas que capturem a qualidade do *ranking*. Métricas como Precisão@k, Recall@k e NDCG@k são amplamente adotadas para esse fim, conforme discutido em trabalhos clássicos sobre avaliação em recuperação de informação e sistemas de recomendação, como discutido em (Herlocker et al., 2004) e (Schütze; Manning; Raghavan, 2008). Este trabalho utiliza essas métricas, juntamente com métricas de erro de predição como RMSE e MAE, e o coeficiente de determinação R^2, para fornecer uma análise comparativa abrangente da capacidade dos diferentes modelos em posicionar itens relevantes no topo das listas de recomendação e em prever as avaliações dos usuários. A metodologia de validação cruzada k-fold é empregada para garantir a robustez e a generalização dos resultados.

Finalmente, os datasets MovieLens (Harper; Konstan, 2015) são *benchmarks* padrão na pesquisa em sistemas de recomendação, utilizados em inúmeros estudos para avaliar e comparar algoritmos devido à sua disponibilidade pública, inclusão de *ratings* explícitos e *timestamps*, e características bem documentadas. A utilização das versões 100k e 1M neste trabalho permite analisar o desempenho e a escalabilidade dos modelos em diferentes volumes e densidades de dados.

2.7.1 *Comparação com Trabalho Atual*

O presente trabalho se insere no contexto da contínua exploração de arquiteturas de aprendizado profundo para Sistemas de Recomendação (SR), com um foco particular na avaliação da arquitetura *Transformers*. Enquanto a literatura existente, como (Kang; McAuley, 2018) e (Sun et al., 2019), demonstraram o potencial dos *Transformers* para modelagem sequencial e predição do próximo item, este estudo adota uma abordagem levemente distinta. Inspirado pela capacidade de aprendizado de representações contextuais ricas desses modelos, o *Transformers* aqui desenvolvido é adaptado para a tarefa de predição de *ratings* de itens mascarados dentro de uma sequência de interações do usuário. Esses *ratings* preditos são utilizados como sco-

res para gerar e avaliar *rankings* de recomendação, permitindo uma comparação direta com modelos tradicionais de filtragem colaborativa que também operam na predição de *ratings*.

A principal contribuição e diferenciação deste TCC reside na análise comparativa direta e detalhada do desempenho de *ranking* do modelo *Transformers* frente a um conjunto de *baselines* fortes e diversificados: SVD++, ItemKNN e um Modelo de Popularidade. Enquanto muitos trabalhos focam em comparar novos modelos de DP apenas com outros modelos de DP ou com um conjunto limitado de *baselines*, este estudo enfatiza a importância de contextualizar o desempenho do *Transformers* em relação a algoritmos clássicos e eficientes. A inclusão e a análise aprofundada do Modelo de Popularidade, em particular, são informadas por discussões recentes na literatura que argumentam por uma ”revisita”a este *baseline*, destacando sua potencial força e a necessidade de modelos mais complexos justificarem seu custo (Ji et al., 2020).

Adicionalmente, este estudo fornece uma análise do desempenho dos modelos em cenários de *cold-start* para usuários e itens, um desafio persistente em SR. Ao avaliar todos os modelos sob essa perspectiva, busca-se oferecer *insights* sobre qual abordagem pode ser mais robusta em situações de escassez de dados. Assim, este trabalho complementa a literatura existente ao oferecer uma perspectiva comparativa focada no *ranking*, utilizando uma adaptação da arquitetura *Transformers* para predição de *ratings* e confrontando-a com *baselines* tradicionais e simples em *datasets* de diferentes escalas, com uma atenção particular aos desafios práticos como o *cold-start* e a interpretação da eficácia de modelos complexos.

3 DESENVOLVIMENTO

Este capítulo detalha o processo metodológico adotado para atingir os objetivos propostos nesta pesquisa. Caracterizando-se como uma pesquisa aplicada de natureza experimental e abordagem quantitativa, o foco reside na investigação e comparação do desempenho de diferentes modelos de recomendação. O objetivo geral é comparar a eficácia de *ranking* do modelo *Transformers* com algoritmos de *Machine Learning* estabelecidos (SVD++, ItemKNN e Popularidade) em SR de filmes, utilizando os *datasets* MovieLens. As seções abaixo descrevem as bases de dados utilizadas, os modelos implementados e o fluxo de treinamento e avaliação experimental desenhado para abordar cada um dos objetivos específicos.

3.1 Base de Dados

Foram utilizados dois conjuntos de dados públicos amplamente reconhecidos da plataforma *MovieLens*. O primeiro deles, o *MovieLens* 100K, contém 100.000 avaliações realizadas por 943 usuários em cerca de 1.700 filmes. O segundo, o *MovieLens* 1M, oferece 1.000.000 de avaliações feitas por 6.040 usuários em aproximadamente 3.900 filmes. A escolha destes *datasets* visa fornecer um ambiente robusto e padronizado para a avaliação comparativa dos modelos, conforme o objetivo geral.

Durante a etapa de pré-processamento, os dados foram preparados para atender aos requisitos específicos de cada modelo e para permitir a análise de cenários particulares. Para o modelo *Transformers*, os valores de *rating* (originalmente de 1 a 5) foram normalizados para uma escala entre 0 e 1. Os demais modelos (Popularidade, ItemKNN e SVD++) utilizaram os *ratings* na sua escala original. Essa diferenciação é importante para o correto funcionamento e otimização do modelo *Transformers*.

Visando o objetivo específico de ”Identificar cenários onde o modelo *Transformers* demonstra vantagens ou desvantagens”, foram definidas condições para análise de *cold-start*: usuários com 20 ou menos interações e itens com 5 ou menos avaliações em seu histórico. Essa separação permite uma avaliação focada do desempenho dos modelos em situações de escassez de dados. Adicionalmente, para garantir a viabilidade da construção de sequências para o modelo *Transformers* e uma comparação justa, usuários com um número muito baixo de interações foram filtrados dos conjuntos de dados antes da divisão em folds de validação cruzada.

3.2 Modelos Implementados

Para alinhar-se com o objetivo geral de avaliar o modelo *Transformers* frente a modelos estabelecidos, quatro abordagens distintas foram implementadas e comparadas. A primeira abordagem é o modelo de Popularidade, utilizado como *baseline*. Este modelo não personalizado recomenda itens com base em sua popularidade global, definida como a média das avaliações (*ratings*) por item, calculada a partir do conjunto de treinamento de cada *fold*, utilizando a biblioteca Pandas.

Em seguida, implementou-se o ItemKNN, um algoritmo clássico de Filtragem Colaborativa baseada em item. Neste método, a predição de notas para um par usuário-item considera os k itens mais similares ao item em questão que já foram avaliados pelo usuário.

A terceira abordagem é o SVD++, uma extensão do algoritmo SVD para fatoração de matrizes, que incorpora tanto *feedback* explícito quanto implícito, além dos vieses de usuários e itens. A implementação utilizou a classe SVDpp da biblioteca *surprise*. Para o *MovieLens* 100K, foram configurados 20 fatores latentes e 75 épocas de treinamento, enquanto para o *MovieLens* 1M, utilizaram-se 30 fatores latentes e 150 épocas.

Por fim, foi implementado um modelo baseado na arquitetura *Transformers*, adaptado para a tarefa de predição de *ratings* em sequências de interações de usuários. Este modelo passou por um pré-processamento específico, no qual foram geradas sequências de interações de tamanho fixo para cada usuário: 30 itens para o *MovieLens* 100K e 100 itens para o *MovieLens* 1M. A arquitetura do modelo utilizou quatro camadas de atenção *multi-head*, com dimensões dos *embeddings* de 128 para o *dataset* 100K e 256 para o 1M, aplicando-se um *dropout* de 0,2 para regularização. O treinamento foi realizado com o otimizador Adam, taxa de aprendizado de 0,001, *batch size* de 128 e *early stopping* com paciência de 10 épocas. A natureza sequencial deste modelo é central para o objetivo específico de avaliar o aproveitamento de dados sequenciais.

3.3 Fluxo de Treinamento

3.3.1 Preparação dos Dados e Validação Cruzada

Após o carregamento e pré-processamento inicial dos *datasets* MovieLens 100K e 1M, foi adotada uma estratégia de validação cruzada com 10 *folds*.

No caso específico do *Transformers*, 20% dos itens em cada sequência foram mascarados para permitir que o modelo aprendesse a prever valores ausentes com

base no contexto anterior. Além disso, para o *Transformers*, além da normalização dos *ratings*, foi feito o mapeamento de identificadores de filmes para IDs internos e a construção das sequências temporais de interação dos usuários.

3.3.2 Treinamento e Avaliação por Modelo em Cada *Fold*

No modelo de Popularidade, o treinamento consistiu no cálculo da média de *ratings* para cada item presente no conjunto de treino do *fold* corrente. Para a predição e avaliação, os *ratings* dos itens avaliados por cada usuário no conjunto de validação do *fold* foram previstos como a média de *ratings* do item ou uma média global do treino, caso o item fosse desconhecido. Com base nesses *scores*, as métricas de erro (RMSE, MAE) e, após ordenar os itens por score, as métricas de *ranking* (NDCG@10, Precisão@10, *Recall*@10) foram calculadas. O tempo para o cálculo das médias foi registrado para conferir a eficiência.

Para os modelos ItemKNN e SVD++, o treinamento foi realizado utilizando as respectivas classes da biblioteca *surprise* (KNN*Baseline* e SVDpp). Os modelos treinados foram, então, utilizados para prever os *ratings* dos pares usuário-item no conjunto de validação do *fold*. De forma semelhante ao modelo de Popularidade, essas predições foram usadas para calcular RMSE, MAE e, após ranqueamento, NDCG@10, Precisão@10 e *Recall*@10. O tempo de treinamento de cada modelo no *fold* também foi cronometrado.

O modelo *Transformers* foi treinado com as sequências de interações dos usuários do conjunto de treino do *fold*. A tarefa de treinamento envolveu a predição de *ratings* para itens artificialmente mascarados dentro das sequências, o que força o modelo a aprender a prever valores ausentes com base no contexto sequencial anterior e posterior. Na etapa de predição e avaliação, para cada sequência de um usuário no conjunto de validação, itens foram mascarados um a um, e o modelo previu o *rating* para esse item mascarado. As predições normalizadas foram convertidas de volta à escala original antes do cálculo das métricas. RMSE e MAE foram calculados sobre essas predições denormalizadas, e os *rankings* para NDCG@10, Precisão@10 e Recall@10 foram gerados ordenando os itens com base nesses scores preditos. Este processo avalia diretamente a capacidade do *Transformers* de utilizar o contexto sequencial. O tempo de treinamento no *fold* foi igualmente registrado.

3.3.3 Cálculo das Métricas de Avaliação

Para avaliar o desempenho dos modelos, foram calculadas diversas métricas em cada *fold* do processo de validação cruzada, sendo posteriormente considerada a média dos resultados obtidos. A primeira métrica utilizada foi o RMSE, que mensura a

magnitude média dos erros de predição, penalizando mais severamente erros maiores. Em complemento, foi calculado o MAE, que avalia a média das diferenças absolutas entre os valores preditos e os reais, fornecendo uma visão mais direta e menos sensível a *outliers* do erro médio.

A métrica R^2 também foi empregada, sendo responsável por indicar a proporção da variância dos *ratings* reais que é explicada pelas predições do modelo. Quanto mais próximo de 1 for o valor de R^2, melhor será o ajuste dos scores preditos em relação aos valores reais.

Além dessas métricas de erro e ajuste, foram adotadas métricas específicas para a avaliação da qualidade das listas de recomendação. A Precisão@10 representa a proporção de itens relevantes entre os dez primeiros itens recomendados ao usuário, considerando um item como relevante quando seu rating real é maior ou igual a 4,0. Já o *Recall*@10 mensura a proporção desses itens relevantes recomendados entre os dez primeiros em relação ao total de itens relevantes disponíveis no conjunto de validação para aquele usuário.

A partir da precisão e do *recall*, foi calculado o F1-*score*, que consiste na média harmônica dessas duas métricas, oferecendo uma avaliação balanceada entre a capacidade de recomendação de itens relevantes e a abrangência sobre o total de itens relevantes. Por fim, foi utilizada a métrica NDCG@10, que avalia a qualidade do *ranking* atribuindo pesos maiores às posições mais altas da lista, de forma que recomendações corretas nas primeiras posições impactem mais positivamente o resultado.

3.3.4 Análise de Cenários Específicos

Para identificar cenários onde o modelo *Transformers* demonstra vantagens ou desvantagens, foram definidas diretrizes para a análise de cenários de *cold-start*.

Usuários em *cold-start* foram caracterizados como aqueles com 20 ou menos interações totais no *dataset*. Similarmente, itens em *cold-start* foram definidos como aqueles com 5 ou menos avaliações no *dataset*. As métricas de avaliação foram calculadas separadamente para as predições envolvendo esses usuários e itens específicos no conjunto de validação de cada *fold*. Esta abordagem permitiu uma análise focada do desempenho dos modelos em condições de esparsidade de dados.

Ao final do processo de validação cruzada, os resultados das métricas de cada *fold* foram agregados para fornecer uma estimativa robusta do desempenho de cada modelo, permitindo as comparações e análises necessárias para responder aos objetivos da pesquisa.

3.4 Especificações do Ambiente de Execução

Com o objetivo de permitir a replicação dos experimentos apresentados neste trabalho, esta seção descreve as especificações do ambiente computacional utilizado durante o desenvolvimento, treinamento e avaliação dos modelos.

Todos os experimentos foram realizados em um ambiente local com as seguintes configurações de *hardware* e *software*:

- **Processador (CPU):** AMD Ryzen 7 5700X3D
- **Placa de Vídeo (GPU):** AMD Radeon RX 6750 XT
- **Memória RAM:** 32 GB DDR4
- **Sistema Operacional:** Windows 11 64 bits
- **Linguagem de Programação:** Python 3.10
- **Bibliotecas principais utilizadas:** pandas, numpy, scikit-learn, surprise, matplotlib, torch, transformers

É importante ressaltar que, embora os modelos tenham sido implementados de forma a não depender exclusivamente de GPU, o uso da aceleração gráfica contribuiu para a redução dos tempos de treinamento, especialmente no caso do modelo baseado em *Transformers*.

4 RESULTADOS DOS MODELOS

Os Quadros 1 e 2 apresentam os resultados médios obtidos pelos modelos avaliados (*Transformers*, SVD++, ItemKNN e Popularidade) nos *datasets MovieLens* 100K e 1M, considerando as métricas RMSE, MAE, R^2, NDCG@10, Precisão@10, *Recall*@10 e tempo de treinamento.

Quadro 1: Resultados médios no *dataset MovieLens* 100K

Métrica	**Transformers**	**SVD++**	**ItemKNN**	**Popularidade**
RMSE	1.1400	1.0289	1.0282	1.0230
MAE	0.9207	0.8242	0.8165	0.8163
R^2	0.0354	0.1618	0.1630	0.1709
NDCG@10	0.7563	0.8571	0.8608	0.8580
Precisão@10	0.5286	0.7796	0.7855	0.7828
Recall@10	0.3517	0.2564	0.2606	0.2589
Tempo (s)	127.31	72.15	1.14	0.0027

Quadro 2: Resultados médios no *dataset MovieLens* 1M

Métrica	**Transformers**	**SVD++**	**ItemKNN**	**Popularidade**
RMSE	1.0885	0.9899	0.9806	0.9790
MAE	0.8649	0.8011	0.7810	0.7821
R^2	0.0487	0.2143	0.2289	0.2313
NDCG@10	0.7303	0.8808	0.8819	0.8820
Precisão@10	0.5789	0.8413	0.8429	0.8424
Recall@10	0.1665	0.2003	0.2012	0.2010
Tempo (s)	1739.60	4058.28	24.75	0.0240

Analisando as métricas de erro, os modelos tradicionais consistentemente superaram o *Transformers*. No *MovieLens* 100K, o modelo de Popularidade (RMSE 1.0230, MAE 0.8163) e ItemKNN (RMSE 1.0282, MAE 0.8165) obtiveram resultados próximos, seguidos pelo modelo SVD++ (RMSE 1.0289, MAE 0.8242). O *Transformers* obteve os maiores erros (RMSE 1.1400, MAE 0.9207). Tendência similar foi observada no MovieLens 1M, onde o modelo de Popularidade (RMSE 0.9790) e ItemKNN (RMSE 0.9806) foram os melhores, enquanto o *Transformers* (RMSE 1.0885) novamente apresentou o maior erro. Os valores de R^2 corroboram essa observação, indicando uma maior capacidade dos modelos tradicionais em explicar a variabilidade das avaliações.

Isso sugere que, para a predição direta de *ratings*, a complexidade adicional do modelo *Transformers*, na configuração utilizada, não se traduziu em uma redução dos erros de predição, e modelos mais simples ou com diferentes mecanismos de regularização e captura de vieses foram mais eficazes.

Em relação à qualidade do *ranking*, fundamental para sistemas de recomendação, as métricas NDCG@10, Precisão@10 e *Recall*@10 revelam que os modelos tradicionais também geraram listas de recomendação mais relevantes. No *MovieLens* 100K, o ItemKNN destacou-se com o maior NDCG@10 (0.8608) e Precisão@10 (0.7855). O *Transformers*, em contrapartida, apresentou o menor NDCG@10 (0.7563) e Precisão@10 (0.5286), embora seu Recall@10 (0.3517) tenha sido superior aos demais, sugerindo que, apesar de encontrar mais itens relevantes, sua ordenação e a precisão entre os *top*-10 foram inferiores. No *MovieLens* 1M, o modelo de Popularidade e o ItemKNN apresentaram os melhores e muito próximos valores de NDCG@10 (0.8820 e 0.8819, respectivamente) e Precisão@10 (0.8424 e 0.8429). O *Transformers* novamente obteve os piores indicadores de *ranking* (NDCG@10 0.7303, Precisão@10 0.5789, *Recall*@10 0.1665). Esses resultados indicam uma dificuldade do modelo *Transformers*, na configuração testada, em posicionar os itens mais relevantes no topo das listas de recomendação em comparação com abordagens mais estabelecidas.

Apesar da capacidade teórica dos *Transformers* em modelar sequências, essa característica não garantiu uma performance superior em métricas de *ranking* tradicionais neste estudo, levantando questões sobre a adequação da arquitetura ou dos dados para esta tarefa específica sem ajustes mais finos ou um volume de dados ainda maior.

4.1 Desempenho em Cenários Específicos

Os quadros abaixo apresentam os resultados dos modelos em situações de *cold-start*.

Quadro 3: Desempenho em *cold-start* (*MovieLens* 100K)

Modelo	RMSE (Usuário)	RMSE (Item)	F1 (Usuário)	Precision	Recall
Transformers	1.1766	1.3651	0.1846	0.6221	0.1084
SVD++	1.1057	1.5337	0.2892	0.7340	0.1801
ItemKNN	1.1022	1.5560	0.3488	0.7368	0.2284
Popularidade	1.1223	N/A	0.3309	0.7449	0.2127

Quadro 4: Desempenho em *cold-start* (*MovieLens* 1M)

Modelo	RMSE (Usuário)	RMSE (Item)	F1 (Usuário)	Precision	Recall
Transformers	1.1193	1.3572	0.5204	0.7035	0.4130
SVD++	1.0217	1.5560	0.4299	0.8496	0.2877
ItemKNN	1.0107	1.3835	0.6047	0.8189	0.4793
Popularidade	1.0136	N/A	0.5895	0.8235	0.4590

4.1.1 Análise do contexto *Cold-Start*

Em situações de *cold-start*, os modelos enfrentam o desafio de fazer recomendações com pouca ou nenhuma informação prévia sobre usuários ou itens. Os resultados apresentados nos Quadros anteriores permitem analisar o comportamento dos algoritmos nesses dois cenários distintos.

Para *usuários cold-start*, os resultados do *dataset* 100K mostram que o modelo ItemKNN apresentou o melhor desempenho geral, com maior valor de F1 (0.3488), seguido pelo modelo de Popularidade (0.3309). O modelo *Transformers* obteve o menor valor de F1 (0.1846) e o maior RMSE (1.1766), indicando menor capacidade de gerar recomendações precisas para novos usuários. No *dataset* 1M, observou-se uma melhora geral nos indicadores, com o ItemKNN novamente se destacando (F1 = 0.6047), enquanto o *Transformers* apresentou F1 intermediário (0.5204), à frente do SVD++ (0.4299), mas ainda com RMSE mais elevado (1.1193).

Já para *itens cold-start*, os resultados mostram que o modelo *Transformers* apresentou o menor RMSE em ambos os *datasets*, com valores de 1.3651 no *MovieLens* 100K e 1.3572 no 1M. Os modelos tradicionais, como SVD++ e ItemKNN, obtiveram RMSE acima de 1.53 no dataset 100K e permaneceram acima de 1.38 no 1M. Esses resultados indicam que o *Transformers*, embora não tenha se destacado na recomendação para novos usuários, demonstrou maior capacidade de generalização em situações de *cold-start* para itens. Isso pode ser atribuído à sua habilidade de capturar relações contextuais e dependências sequenciais nas interações dos usuários, permitindo que o modelo infira relevância mesmo com informações escassas sobre os itens.

4.1.2 Aproveitamento de Dados Sequenciais

Uma das principais hipóteses desta pesquisa era que a capacidade intrínseca dos modelos *Transformers* de capturar dependências sequenciais e contextuais complexas nas interações dos usuários se traduziria em uma vantagem significativa na qualidade do *ranking* das recomendações. No entanto, os resultados gerais apresentados, particularmente as métricas NDCG@10 e Precisão@10, indicam que, para as configurações e os *datasets MovieLens* testados, essa capacidade não resultou em uma superioridade consistente frente aos modelos tradicionais mais simples, como ItemKNN e Popularidade.

Diversos fatores podem contribuir para essa observação. Primeiramente, a complexidade inerente aos *Transformers* pode demandar volumes de dados consideravelmente maiores para um treinamento eficaz, superando a esparsidade e permitindo que o modelo aprenda padrões sequenciais verdadeiramente generalizáveis; os *datasets* utilizados, embora padrões, podem não atingir esse limiar para a arquitetura especí-

fica empregada. Em segundo lugar, um ajuste de hiperparâmetros mais exaustivo e específico para a tarefa de *ranking* sequencial poderia ser necessário para otimizar o desempenho do *Transformers*. Adicionalmente, é possível que a natureza das sequências de interação nos *datasets MovieLens*, focados em avaliações de filmes, não possua a densidade ou o comprimento de dependências contextuais que permitiriam ao *Transformers* explorar plenamente seu potencial em comparação com domínios onde o comportamento sequencial é mais pronunciado.

Apesar disso, a vantagem notável do *Transformers* no cenário de *cold-start* para itens sugere que o aprendizado contextual da sequência, mesmo que não otimizado para o *ranking* geral, de fato auxilia na generalização para itens com poucas interações. Isso indica que, embora a superioridade geral no *ranking* não tenha sido alcançada, a arquitetura *Transformers* possui mecanismos valiosos para lidar com a escassez de dados em itens, inferindo características relevantes a partir do contexto sequencial em que aparecem, um aspecto que modelos puramente colaborativos ou baseados em popularidade têm maior dificuldade em abordar.

4.2 Eficiência Computacional

A avaliação da eficiência computacional, medida pelo tempo de treinamento em segundos por *fold*, é um fator crucial para determinar a viabilidade de implementação e manutenção de modelos de recomendação em ambientes de produção. Os resultados demonstram uma variação considerável no custo computacional entre as abordagens. No dataset *MovieLens* 100K, os modelos de Popularidade (0.0027s) e ItemKNN (1.14s) destacaram-se pela extrema rapidez de treinamento, tornando-os ideais para cenários com recursos limitados ou necessidade de re-treinamentos frequentes. O SVD++ (72.15s) demandou um tempo consideravelmente maior, enquanto o *Transformers* (127.31s) foi o mais lento neste *dataset* menor, refletindo sua maior complexidade arquitetural.

Essa disparidade de tempos tornou-se ainda mais pronunciada no *dataset MovieLens* 1M. Enquanto Popularidade (0.0240s) e ItemKNN (24.75s) mantiveram uma eficiência notável, o tempo de treinamento do *Transformers* aumentou para aproximadamente 29 minutos (1739.60s). Assim como o SVD++, com cerca de 67 minutos (4058.28s), tornando-se o mais custoso computacionalmente. Este último resultado evidencia que, embora o *Transformers* seja intrinsecamente complexo, algoritmos iterativos como o SVD++, dependendo do número de fatores latentes e épocas de treinamento configurados, podem apresentar um custo de escalabilidade ainda maior para *datasets* mais volumosos. A escolha entre SVD++ e *Transformers*, do ponto de vista da eficiência, pode depender do ponto de equilíbrio entre a complexidade do modelo

e o custo de suas iterações de treinamento em grandes volumes de dados.

Em suma, os modelos de Popularidade e ItemKNN oferecem uma eficiência imbatível, sendo excelentes escolhas para baselines ou sistemas que priorizam velocidade e baixo custo computacional. Modelos mais sofisticados como SVD++ e *Transformers*, embora potencialmente capazes de capturar padrões mais complexos, impõem um ônus computacional significativamente maior, exigindo uma análise cuidadosa do *trade-off* entre o ganho esperado em acurácia e os recursos de treinamento disponíveis.

5 CONCLUSÕES

Com base nos resultados obtidos nos experimentos com os datasets *MovieLens* 100K e 1M, os modelos tradicionais de recomendação (SVD++, ItemKNN e Popularidade) demonstraram superioridade em métricas como RMSE, MAE e R^2 em comparação com o *Transformers*. No entanto, o *Transformers* mostrou potencial em cenários específicos, como *cold-start* para itens, onde obteve o menor RMSE (1.3572 no *dataset* 1M). Esses resultados reforçam a importância de escolher o modelo adequado para cada contexto, considerando *trade-offs* entre desempenho computacional e qualidade das recomendações.

5.1 Contribuições

Este trabalho busca contribuir para o avanço da área de sistemas de recomendação por meio de:

- Análise comparativa entre abordagens clássicas e modernas, evidenciando que modelos simples como o de Popularidade podem ser altamente competitivos, especialmente em *datasets* maiores (RMSE de 0.9790 no MovieLens 1M).
- Destaque para cenários onde modelos complexos se sobressaem, como no *cold-start* de itens, onde o *Transformers* superou os demais.
- Documentação detalhada de métricas de *ranking* (NDCG@10, Precisão@10), essenciais para avaliar a relevância prática das recomendações.

Além disso, os resultados obtidos podem auxiliar profissionais e pesquisadores na seleção de algoritmos, equilibrando precisão e custo computacional – como no caso do ItemKNN, que combinou bom desempenho (RMSE de 0.9806 no 1M) com tempo de execução baixo (24.75s).

5.2 Limitações

As principais limitações deste estudo incluem:

- Restrição a dados de avaliações explícitas, o que pode não refletir cenários do mundo real com interações implícitas (como cliques ou tempo de visualização).

- Hiperparâmetros fixos, que podem não representar o desempenho ótimo dos modelos em todas as situações.
- Escala dos experimentos, limitada a *datasets* de tamanho médio (100K e 1M), enquanto aplicações industriais frequentemente exigem análise de volumes maiores.

Outro desafio foi a interpretação das métricas de *cold-start*, onde mesmo o *Transformers* – embora melhor em RMSE para itens – apresentou valores absolutos altos (acima de 1.35), indicando espaço para melhorias.

5.3 Trabalhos Futuros

Como direções para pesquisas futuras, recomenda-se:

- Explorar arquiteturas *Transformers* especializadas, como as que incorporam informações contextuais (ex.: tempo, localização), para melhorar seu desempenho em *ranking*.
- Combinações híbridas, como integrar a capacidade do *Transformers* em *cold-start* com a eficiência do ItemKNN em recomendações convencionais.
- Avaliação em *datasets* mais complexos, incluindo dados implícitos, multimodais (ex.: texto, imagem) ou de domínios específicos (e-*commerce*, *streaming*).
- Estudos de escalabilidade, testando os modelos em ambientes distribuídos para reduzir tempos de treinamento (como os 1739.60s do *Transformers* no 1M).

REFERÊNCIAS

AGGARWAL, C. C. et al. *Recommender systems*. [S.l.]: Springer, 2016. v. 1. Citado na página 25.

ALAM, M. T. et al. Comparative analysis of machine learning based filtering techniques using movielens dataset. *Procedia Computer Science*, Elsevier, v. 194, p. 210–217, 2021. Citado 2 vezes nas páginas 18 e 24.

ANGUITA, D. et al. The'k'in k-fold cross validation. In: *ESANN*. [S.l.: s.n.], 2012. v. 102, p. 441–446. Citado na página 24.

ANWAR, T.; UMA, V. Comparative study of recommender system approaches and movie recommendation using collaborative filtering. *International Journal of System Assurance Engineering and Management*, Springer, v. 12, p. 426–436, 2021. Citado na página 24.

BATMAZ, Z. et al. A review on deep learning for recommender systems: challenges and remedies. *Artificial Intelligence Review*, Springer, v. 52, p. 1–37, 2019. Citado na página 24.

BENNETT, J.; LANNING, S. The netflix prize. Netflix, 2007. Citado na página 20.

BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. *The journal of machine learning research*, JMLR. org, v. 13, n. 1, p. 281–305, 2012. Citado na página 25.

BOBADILLA, J. et al. Recommender systems survey. *Knowledge-based systems*, Elsevier, v. 46, p. 109–132, 2013. Citado na página 19.

BURKE, R. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, Springer, v. 12, p. 331–370, 2002. Citado na página 18.

CHAI, T.; DRAXLER, R. R. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, Copernicus Publications Göttingen, Germany, v. 7, n. 3, p. 1247–1250, 2014. Citado na página 21.

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*. [S.l.: s.n.], 2010. p. 39–46. Citado na página 22.

CROFT, W. B.; METZLER, D.; STROHMAN, T. *Search engines: Information retrieval in practice*. [S.l.]: Addison-Wesley Reading, 2010. v. 520. Citado na página 22.

DESHPANDE, M.; KARYPIS, G. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, ACM New York, NY, USA, v. 22, n. 1, p. 143–177, 2004. Citado na página 26.

EKSTRAND, M. D. et al. Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction*, Now Publishers, Inc., v. 4, n. 2, p. 81–173, 2011. Citado na página 24.

FAN, W. et al. Graph neural networks for social recommendation. In: *The world wide web conference*. [S.l.: s.n.], 2019. p. 417–426. Citado na página 19.

GUNAWARDANA, A.; SHANI, G. A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research*, v. 10, n. 12, 2009. Citado na página 21.

GUO, H. et al. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017. Citado na página 18.

HARPER, F. M.; KONSTAN, J. A. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, Acm New York, NY, USA, v. 5, n. 4, p. 1–19, 2015. Citado 2 vezes nas páginas 24 e 29.

HE, X. et al. Lightgcn: Simplifying and powering graph convolution network for recommendation. In: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. [S.l.: s.n.], 2020. p. 639–648. Citado na página 20.

HERLOCKER, J. L. et al. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, ACM New York, NY, USA, v. 22, n. 1, p. 5–53, 2004. Citado 4 vezes nas páginas 21, 23, 26 e 29.

HODSON, T. O. Root mean square error (rmse) or mean absolute error (mae): When to use them or not. *Geoscientific Model Development Discussions*, Göttingen, Germany, v. 2022, p. 1–10, 2022. Citado na página 21.

HU, Y.; KOREN, Y.; VOLINSKY, C. Collaborative filtering for implicit feedback datasets. In: IEEE. *2008 Eighth IEEE international conference on data mining*. [S.l.], 2008. p. 263–272. Citado na página 27.

JAMES, G. et al. *An introduction to statistical learning*. [S.l.]: Springer, 2013. v. 112. Citado na página 20.

JÄRVELIN, K.; KEKÄLÄINEN, J. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, ACM New York, NY, USA, v. 20, n. 4, p. 422–446, 2002. Citado na página 22.

JI, Y. et al. A re-visit of the popularity baseline in recommender systems. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. [S.l.: s.n.], 2020. p. 1749–1752. Citado 3 vezes nas páginas 25, 29 e 30.

JIAO, J. et al. A novel learning rate function and its application on the svd++ recommendation algorithm. *IEEE Access*, IEEE, v. 8, p. 14112–14122, 2019. Citado 2 vezes nas páginas 26 e 27.

KANG, W.-C.; MCAULEY, J. Self-attentive sequential recommendation. In: IEEE. *2018 IEEE international conference on data mining (ICDM)*. [S.l.], 2018. p. 197–206. Citado 3 vezes nas páginas 14, 28 e 29.

KOHAVI, R. et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: MONTREAL, CANADA. *Ijcai*. [S.l.], 1995. v. 14, n. 2, p. 1137–1145. Citado na página 25.

KOREN, Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.: s.n.], 2008. p. 426–434. Citado 2 vezes nas páginas 27 e 28.

KOREN, Y.; BELL, R.; VOLINSKY, C. Matrix factorization techniques for recommender systems. *Computer*, IEEE, v. 42, n. 8, p. 30–37, 2009. Citado 2 vezes nas páginas 17 e 20.

KRICHENE, W.; RENDLE, S. On sampled metrics for item recommendation. In: *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. [S.l.: s.n.], 2020. p. 1748–1757. Citado na página 23.

LEGATES, D. R.; JR, G. J. M. Evaluating the use of "goodness-of-fit" measures in hydrologic and hydroclimatic model validation. *Water resources research*, Wiley Online Library, v. 35, n. 1, p. 233–241, 1999. Citado na página 21.

LIANG, A. Enhancing recommendation systems with multi-modal transformers in cross-domain scenarios. *Journal of Computer Technology and Software*, v. 3, n. 7, 2024. Citado na página 27.

LIANG, D. et al. Variational autoencoders for collaborative filtering. In: *Proceedings of the 2018 world wide web conference*. [S.l.: s.n.], 2018. p. 689–698. Citado na página 22.

LINDEN, G.; SMITH, B.; YORK, J. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, IEEE, v. 7, n. 1, p. 76–80, 2003. Citado 2 vezes nas páginas 26 e 28.

LIU, L.; ÖZSU, M. T. *Encyclopedia of database systems*. [S.l.]: Springer New York, NY, USA, 2009. v. 6. Citado na página 25.

LOPS, P.; GEMMIS, M. D.; SEMERARO, G. Content-based recommender systems: State of the art and trends. *Recommender systems handbook*, Springer, p. 73–105, 2011. Citado 2 vezes nas páginas 16 e 17.

NAUEN, T. C. et al. Which transformer to favor: A comparative analysis of efficiency in vision transformers. *arXiv preprint arXiv:2308.09372*, 2023. Citado na página 14.

NIKZAD-KHASMAKHI, N. et al. Berters: Multimodal representation learning for expert recommendation system with transformers and graph embeddings. *Chaos, Solitons & Fractals*, Elsevier, v. 151, p. 111260, 2021. Citado na página 27.

PARK, S.-T.; CHU, W. Pairwise preference regression for cold-start recommendation. In: *Proceedings of the third ACM conference on Recommender systems*. [S.l.: s.n.], 2009. p. 21–28. Citado na página 19.

PAZZANI, M. J.; BILLSUS, D. Content-based recommendation systems. In: *The adaptive web: methods and strategies of web personalization*. [S.l.]: Springer, 2007. p. 325–341. Citado 2 vezes nas páginas 16 e 17.

POWERS, D. M. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020. Citado na página 23.

REATEGUI, E. B.; CAZELLA, S. C. Sistemas de recomendação. In: CITESEER. *XXV Congresso da Sociedade Brasileira de Computação*. [S.l.], 2005. p. 306–348. Citado 3 vezes nas páginas 16, 17 e 18.

RICCI, F.; ROKACH, L.; SHAPIRA, B. Introduction to recommender systems handbook. In: *Recommender systems handbook*. [S.l.]: Springer, 2010. p. 1–35. Citado na página 19.

RICCI, F.; ROKACH, L.; SHAPIRA, B. Recommender systems handbook. In: ____. [S.l.: s.n.], 2010. v. 1-35, p. 1–35. ISBN 978-0-387-85819-7. Citado 3 vezes nas páginas 16, 23 e 26.

SAID, A.; BELLOGÍN, A. Comparative recommender system evaluation: benchmarking recommendation frameworks. In: *Proceedings of the 8th ACM Conference on Recommender systems*. [S.l.: s.n.], 2014. p. 129–136. Citado 2 vezes nas páginas 21 e 24.

SARWAR, B. et al. Item-based collaborative filtering recommendation algorithms. In: *Proceedings of the 10th international conference on World Wide Web*. [S.l.: s.n.], 2001. p. 285–295. Citado 5 vezes nas páginas 17, 19, 23, 26 e 28.

SCHEIN, A. I. et al. Methods and metrics for cold-start recommendations. In: *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. [S.l.: s.n.], 2002. p. 253–260. Citado 2 vezes nas páginas 18 e 19.

SCHÜTZE, H.; MANNING, C. D.; RAGHAVAN, P. *Introduction to information retrieval*. [S.l.]: Cambridge University Press Cambridge, 2008. v. 39. Citado 3 vezes nas páginas 22, 23 e 29.

SEDHAIN, S. et al. Autorec: Autoencoders meet collaborative filtering. In: *Proceedings of the 24th international conference on World Wide Web*. [S.l.: s.n.], 2015. p. 111–112. Citado na página 20.

SEGARAN, T. *Programming collective intelligence: building smart web 2.0 applications*. [S.l.]: O'Reilly Media, Inc, 2007. Citado na página 13.

SHANI, G.; GUNAWARDANA, A. Evaluating recommendation systems. *Recommender systems handbook*, Springer, p. 257–297, 2011. Citado 3 vezes nas páginas 20, 21 e 23.

SHARMA, L.; GERA, A. A survey of recommendation system: Research challenges. *International Journal of Engineering Trends and Technology (IJETT)*, v. 4, n. 5, p. 1989–1992, 2013. Citado na página 16.

SU, X.; KHOSHGOFTAAR, T. M. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, Wiley Online Library, v. 2009, n. 1, p. 421425, 2009. Citado na página 17.

SUN, F. et al. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In: *Proceedings of the 28th ACM international conference on information and knowledge management*. [S.l.: s.n.], 2019. p. 1441–1450. Citado 3 vezes nas páginas 14, 28 e 29.

VARTAK, M. et al. Towards visualization recommendation systems. *Acm Sigmod Record*, ACM New York, NY, USA, v. 45, n. 4, p. 34–39, 2017. Citado na página 19.

VASWANI, A. et al. Attention is all you need. *Advances in neural information processing systems*, v. 30, 2017. Citado 2 vezes nas páginas 27 e 28.

VOLKOVS, M.; YU, G.; POUTANEN, T. Dropoutnet: Addressing cold start in recommender systems. *Advances in neural information processing systems*, v. 30, 2017. Citado na página 20.

WANG, X. et al. Neural graph collaborative filtering. In: *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. [S.l.: s.n.], 2019. p. 165–174. Citado na página 17.

WANG, Y. et al. A theoretical analysis of ndcg type ranking measures. In: PMLR. *Conference on learning theory*. [S.l.], 2013. p. 25–54. Citado na página 22.

WILLMOTT, C. J.; MATSUURA, K. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, v. 30, n. 1, p. 79–82, 2005. Citado na página 21.

WU, L. et al. A survey on large language models for recommendation. *World Wide Web*, Springer, v. 27, n. 5, p. 60, 2024. Citado na página 23.

WU, S. et al. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, ACM New York, NY, v. 55, n. 5, p. 1–37, 2022. Citado na página 27.

YATES, L. et al. Cross validation for model selection: a primer with examples from ecology. *arXiv preprint arXiv:2203.04552*, 2022. Citado na página 25.

ZHANG, S. et al. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)*, ACM New York, NY, USA, v. 52, n. 1, p. 1–38, 2019. Citado 3 vezes nas páginas 16, 17 e 18.

ZHOU, K. et al. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In: *Proceedings of the 29th ACM international conference on information & knowledge management*. [S.l.: s.n.], 2020. p. 1893–1902. Citado na página 28.

ZHOU, T. et al. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences*, National Academy of Sciences, v. 107, n. 10, p. 4511–4515, 2010. Citado na página 22.