UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Mitigação de Vieses em Sistemas de Recomendação: Uma Abordagem Híbrida de Calibração de Popularidade com LLMs e Otimização de Prompts

Rodrigo Ferrari de Souza

Monografia - MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

Rodrigo Ferrari de Souza

Mitigação de Vieses em Sistemas de Recomendação: Uma Abordagem Híbrida de Calibração de Popularidade com LLMs e Otimização de Prompts

Monografia apresentada ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Prof. Dr. Marcelo Garcia Manzato

Versão original

São Carlos

2025

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi, ICMC/USP, com os dados fornecidos pelo(a) autor(a)

S856m	Souza, Rodrigo Ferrari de Mitigação de Vieses em Sistemas de Recomendação: Uma Abordagem Híbrida de Calibração de Popularidade com LLMs e Otimização de Prompts / Rodrigo Ferrari de Souza ; orientador Marcelo Garcia Manzato. – São Carlos, 2025. 49 p. : il. (algumas color.) ; 30 cm. Monografia (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2025. 1. LaTeX. 2. abnTeX. 3. Classe USPSC. 4. Editoração de texto. 5. Normalização da documentação. 6. Tese. 7. Dissertação. 8. Documentos (elaboração). 9. Documentos eletrônicos. I. Manzato, Marcelo Garcia, orient. II. Título.

Rodrigo Ferrari de Souza

Bias Mitigation in Recommender Systems: A Hybrid Popularity Calibration Approach with LLMs and Prompt Optimization

Monograph presented to the Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, as part of the requirements for obtaining the title of Specialist in Artificial Intelligence and Big Data.

Concentration area: Artificial Intelligence

Original version

São Carlos

2025

RESUMO

Souza, R.F. de **Mitigação de Vieses em Sistemas de Recomendação: Uma Abordagem Híbrida de Calibração de Popularidade com LLMs e Otimização de Prompts**. 2025. 49 p. Monografia (MBA em Inteligência Artificial e Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2025.

Sistemas de Recomendação (SR) são amplamente utilizados para personalizar conteúdos em plataformas digitais. Contudo, frequentemente reforçam vieses de popularidade, bolhas de filtro e polarização, limitando a diversidade e comprometendo a justiça. Avanços recentes em Large Language Models (LLMs) abrem novas oportunidades para enfrentar esses desafios por meio de interação em linguagem natural e técnicas avançadas de engenharia de prompts. Este trabalho investiga o desempenho de estratégias de recomendação baseadas em LLMs em comparação com métodos tradicionais de calibração, bem como os efeitos de diferentes estratégias de otimização de prompts na qualidade de recomendações multiobjetivo. Foram conduzidos experimentos no conjunto de dados MovieLens, avaliando oito modelos baseline (incluindo calibração por popularidade e por gênero, Bayesian Personalized Ranking e abordagens híbridas) em relação a um recomendador baseado em LLM (LLaMa 3.1-8b-instruct), com e sem otimização de prompts. A avaliação considerou múltiplas métricas — MAP, NDCG@10, Long Tail Coverage (LTC), F1 Score para justiça, RMSE para descalibração de popularidade e uma métrica agregada baseada na Multi-Attribute Utility Theory (MAUT) — ao longo de seis execuções independentes. A significância estatística foi verificada por meio do teste não paramétrico de Wilcoxon, com nível de significância de 5%. Os resultados mostram que a estratégia baseada em LLM, mesmo sem otimização, supera a maioria dos métodos tradicionais em acurácia de ranqueamento e alcança um equilíbrio favorável entre diversidade e justiça. A otimização de prompts não gerou ganhos uniformes, mas se mostrou eficaz para ajustar as prioridades do sistema: algumas configurações preservaram a precisão, enquanto outras ampliaram a diversidade e a exposição à cauda longa. Esses achados indicam que LLMs podem atuar como componentes robustos e adaptáveis em SRs multiobjetivo, com a engenharia de prompts funcionando como mecanismo de ajuste fino para atender a metas específicas de aplicação. Como trabalhos futuros, propõe-se realizar testes A/B em ambiente online, ampliar a análise para outros domínios e conjuntos de dados com diferentes graus de popularidade e esparsidade, e explorar novas técnicas de otimização de prompts integradas a métricas de justiça, visando aumentar a adaptabilidade e o impacto social de recomendadores baseados em LLMs.

Palavras-chave: Sistemas de Recomendação. Large Language Models. Otimização de Prompts. Viés de Popularidade. Justiça. Calibração.

ABSTRACT

Souza, R.F. de **Mitigação de Vieses em Sistemas de Recomendação: Uma Abordagem Híbrida de Calibração de Popularidade com LLMs e Otimização de Prompts**. 2025. 49 p. Monograph (MBA in Artificial Intelligence and Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2025.

Recommender Systems (RS) are widely used to personalize content across digital platforms. However, they often reinforce popularity bias, filter bubbles, and polarization, limiting diversity and fairness. Recent advances in Large Language Models (LLMs) open new opportunities for addressing these challenges through natural language interaction and advanced prompt engineering. This work investigates the performance of LLM-based recommendation strategies compared to traditional calibration methods, as well as the effects of different prompt optimization strategies on multi-objective recommendation quality. We conducted experiments on the MovieLens dataset, evaluating eight baseline models (including popularity- and genre-based calibration, Bayesian Personalized Ranking, and hybrid approaches) against an LLM-based recommender (LLaMa 3.1-8b-instruct), both with and without prompt optimization. The evaluation considered multiple metrics—MAP, NDCG@10, Long Tail Coverage (LTC), F1 Score for fairness, RMSE for popularity miscalibration, and a Multi-Attribute Utility Theory (MAUT) aggregate score—over six independent runs. Statistical significance was assessed using the Wilcoxon signed-rank test at a 5% significance level. Results show that the LLM-based strategy, even without optimization, outperforms most traditional methods in ranking accuracy and achieves a balanced trade-off between diversity and fairness. Prompt optimization did not yield uniform improvements but proved effective for tailoring system priorities: some configurations preserved accuracy while others enhanced diversity and long-tail exposure. These findings suggest that LLMs can serve as robust and adaptable components in multi-objective RS, with prompt engineering acting as a fine-tuning mechanism to meet specific application goals. Future work will explore online A/B testing, domain generalization to datasets with varying sparsity and popularity distributions, and novel prompt optimization techniques integrated with fairness-aware metrics to further enhance the adaptability and social impact of LLM-based recommenders.

Keywords: Recommender Systems. Large Language Models. Prompt Optimization. Popularity Bias. Fairness. Calibration.

SUMÁRIO

1 INTRODUÇÃO

Sistemas de recomendação são fundamentais para a personalização de conteúdos em plataformas digitais, permitindo que cada usuário encontre filmes, músicas ou outros produtos alinhados às suas preferências (Zangerle; Bauer, 2022). Apesar de seu impacto positivo, tais sistemas frequentemente apresentam desafios como vieses, polarização e bolhas de filtro, que limitam a diversidade e comprometem a justiça tanto para usuários quanto para itens. Embora existam técnicas isoladas para mitigação de vieses, aumento da transparência e análise de comportamento, ainda há uma lacuna na integração desses aspectos em uma abordagem holística (Chen *et al.*, 2023).

Uma técnica promissora para lidar com esses problemas é a calibração das recomendações (Steck, 2018; Sacilotti; Souza; Manzato, 2023; Souza; Manzato, 2024), que busca alinhar os resultados sugeridos pelo sistema às preferências expressas ou implícitas do usuário. Um sistema calibrado tende a refletir de forma proporcional o histórico de consumo do indivíduo, evitando que apenas um gênero ou categoria domine as recomendações. Tradicionalmente, essa técnica é aplicada como etapa de pós-processamento, por meio de reponderações, interpolação de rankings ou ajustes nos *scores* (Sacilotti; Souza; Manzato, 2023; Atauchi *et al.*, 2025; Silva; Jannach, 2025).

Nos últimos anos, avanços em arquiteturas neurais profundas, especialmente os Grandes Modelos de Linguagem (Large Language Models – LLMs), como GPT e LLaMA, abriram novas possibilidades para a calibração de sistemas de recomendação (Ortega; Souza; Manzato, 2024; Touvron *et al.*, 2023). Esses modelos permitem o uso de engenharia e otimização de prompts, bem como de cadeias de raciocínio (chain-of-thought), para reclassificação personalizada e calibrada (Gao *et al.*, 2025), possibilitando também o controle de objetivos do sistema em linguagem natural (Ortega; Souza; Manzato, 2024). Estudos indicam ganhos expressivos — entre 5,6% e 20,7% em métricas como NDCG@10 — ao otimizar prompts para reclassificação (Wang *et al.*, 2025). Entretanto, a maioria desses trabalhos mantém foco quase exclusivo na melhoria da precisão do ranking, negligenciando aspectos como diversidade, cobertura, justiça e viés de popularidade (Bittencourt *et al.*, 2023).

Diante desse cenário, este trabalho propõe investigar o potencial dos LLMs na calibração de sistemas de recomendação, com foco no equilíbrio entre precisão, diversidade e justiça. Diferentemente de estudos prévios, será avaliado não apenas o desempenho dos LLMs frente a métodos tradicionais de calibração, mas também o impacto da otimização de prompts em múltiplas estratégias. A abordagem será dividida em duas etapas: (i) comparação direta entre modelos tradicionais e LLMs na tarefa de recomendação/calibração e (ii) avaliação do efeito da otimização de prompts sobre métricas de precisão, diversidade,

cobertura e justiça.

Assim, este trabalho busca responder às seguintes perguntas de pesquisa:

RQ1: *Em que medida estratégias baseadas em LLMs superam métodos tradicionais de recomendação em métricas como precisão, diversidade, cobertura e justiça?*

RQ2: *Qual é o impacto da otimização de prompts em LLMs na qualidade das recomendações considerando múltiplas métricas e objetivos?*

Para responder a essas questões, serão testadas diferentes estratégias de otimização de prompts, avaliadas por métricas como MAP, NDCG@10, MRMC (*Mean Rank Miscalibration*) de Gêneros, MRMC de Popularidade, LTC (*Long Tail Coverage*) e GAP (*Group Average Popularity*). Também será utilizada uma métrica agregada baseada na Teoria da Utilidade Multiatributo (MAUT) (Neiva; Gomes, 2007; Carvalho; Rocha, 2020), permitindo analisar trade-offs entre precisão, popularidade e diversidade de forma integrada. Os resultados serão comparados com abordagens clássicas de recomendação, como BPR (*Bayesian Personalized Ranking*) (Rendle *et al.*, 2012), calibração por gêneros (Steck, 2018) e calibração por popularidade (Abdollahpouri *et al.*, 2021).

Espera-se que a pesquisa contribua para o avanço no desenvolvimento de sistemas de recomendação mais equilibrados e explicáveis, promovendo experiências mais justas e personalizadas para usuários e provedores de conteúdo.

2 FUNDAMENTAÇÃO TEÓRICA

O desenvolvimento de um sistema para gerar recomendações de itens aos usuários envolve conceitos relacionados a sistemas de recomendação, vieses, calibração, LLMs e otimização de LLMs. Este capítulo discute os conceitos fundamentais na literatura e visa ajudar a construção desta pesquisa de forma a avançar o estado da arte.

2.1 Sistemas de Recomendação

Com o avanço da internet, os sistemas de recomendação evoluíram para sugerir itens em diversos domínios, como compras, filmes, notícias e buscas, auxiliando na tomada de decisão dos usuários (Ricci; Rokach; Shapira, 2015). Esses sistemas buscam reduzir o esforço na descoberta de itens que correspondam às preferências do usuário, utilizando seu histórico de interações ou o de outros usuários com perfis semelhantes, a partir de feedback explícito ou implícito (Ricci; Rokach; Shapira, 2011).

A área ganhou destaque com o *Netflix Prize* em 2006, que impulsionou o desenvolvimento de novos algoritmos e metodologias (Bennett; Lanning *et al.*, 2007). Entre as principais abordagens estão a filtragem colaborativa, baseada em preferências de usuários similares, e a filtragem por conteúdo, que utiliza informações dos itens e do histórico do usuário. Outras técnicas incluem métodos baseados em conhecimento, regras, contexto, dados demográficos e híbridos.

2.2 Abordagens Clássicas

As abordagens em sistemas de recomendação se fundamentam em três artefatos principais (Ricci; Rokach; Shapira, 2011):

- **Interações:** registros das ações dos usuários (explícitas ou implícitas) que auxiliam na melhoria contínua das recomendações.
- **Itens:** os objetos sugeridos, cujo valor depende do interesse do usuário.
- **Usuários:** cada usuário possui um perfil único que orienta as recomendações.

Com base nesses elementos, as abordagens podem ser divididas em (Aggarwal, 2016):

- **Filtragem Colaborativa:** utiliza o histórico e as preferências de usuários semelhantes, com métodos baseados em memória ou modelo.

- **Filtragem Baseada em Conteúdo:** associa interações passadas a metadados dos itens.
- **Filtragem Baseada em Conhecimento:** recomenda itens avaliando sua utilidade sem recorrer ao histórico do usuário, por meio de regras ou casos.
- **Filtragem Híbrida:** combina múltiplos algoritmos, como Filtragem Colaborativa e Filtragem Baseada em Conteúdo (perfis de gêneros, elenco, descrições), podendo ainda integrar modelos de popularidade ou técnicas de aprendizado profundo. Essa combinação reforça a eficácia das recomendações, explorando tanto padrões de uso coletivo quanto características individuais dos itens.
- **Abordagem Demográfica:** emprega informações como idade, gênero e localização para agrupar usuários com perfis semelhantes e recomendar itens consumidos dentro do mesmo grupo demográfico.
- **Abordagem Baseada em Comunidade:** integra dados do usuário e de seus contatos em redes sociais.

Essas técnicas permitem a personalização das recomendações e servem como base para a avaliação e comparação dos sistemas, tema abordado na próxima seção.

2.3 Avaliação dos Sistemas de Recomendação

2.3.1 Formas para Avaliação dos Sistemas de Recomendação

Para validar os sistemas de recomendação, geralmente são utilizados três métodos (Ricci; Rokach; Shapira, 2015; Aggarwal, 2016):

- ***Offline:*** Avaliação baseada em conjuntos de dados de teste, que mede a capacidade preditiva, mas não capta a reação dos usuários nem aspectos como novidade.
- **Experimentos com usuários:** Envolvem a participação de usuários para analisar interações e comportamentos, embora possam sofrer vieses na seleção dos participantes.
- ***Online:*** Realizada em sistemas já implementados, avalia a influência do sistema no comportamento dos usuários, reduzindo o viés decorrente da seleção artificial de participantes.

2.3.2 Características dos Sistemas de Recomendação

As principais métricas para avaliar os sistemas de recomendação são (Aggarwal, 2016):

- **Acurácia:** Capacidade de prever preferências e avaliações dos usuários.
- **Cobertura:** Proporção de itens ou usuários que podem ser recomendados.
- **Confiança:** Nível de satisfação dos usuários com as recomendações.
- **Diversidade:** Grau de dissimilaridade entre as recomendações, que promove novidade e surpresa, embora possa reduzir a acurácia.
- **Escalabilidade:** Capacidade do sistema de gerar recomendações de forma eficiente mesmo em alta demanda.
- **Novidade:** Aptidão para sugerir itens ainda desconhecidos pelo usuário.
- **Robustez:** Estabilidade do sistema diante de muitas requisições e avaliações incorretas.
- **Surpresa:** Habilidade de apresentar recomendações inesperadas e atrativas.

2.3.3 Métricas para Avaliação dos Sistemas de Recomendação

Para avaliar algumas das características de sistemas de recomendação apresentadas anteriormente, utilizaremos neste trabalho as seguintes métricas:

- ***Mean Average Precision* (MAP).** Conforme a Equação 2.1, mede a acurácia global, calculando a média da precisão ($AveP(n)$) para todos os itens recomendados (Parra; Sahebi, 2013). Seus valores variam de 0 a 1, onde quanto maior, melhor.

$$MAP = \frac{1}{|N|} \sum_{n}^{N} AveP(n) \tag{2.1}$$

- ***Mean Rank Miscalibration* (MRMC).** Métrica que computa o grau de calibração da lista conforme as preferências do usuário (Silva; Manzato; Durão, 2021). A Equação 2.2 obtém o valor de justiça, sendo ele normalizado a partir do pior caso de divergência da lista, dependendo da medida de divergência utilizada e representada por $F(p, q(\{\}))$. Na Equação 2.3 é calculada a soma das médias dos valores dos erros de calibração para cada posição da lista. Por fim, a Equação 2.4 obtém o valor médio para todos os usuários. Seus valores variam de 0 a 1, sendo que quanto menor, melhor.

$$MC(p, q) = \frac{F(p, q)}{F(p, q(\{\}))} \tag{2.2}$$

$$RMC(u) = \frac{\sum_{j=1}^{N} MC(p, q(R^{*}@J))}{N} \tag{2.3}$$

$$MRMC(u) = \frac{\sum_{u \in U} RMC(u)}{|U|} \tag{2.4}$$

A métrica foi utilizada para avaliar tanto a distribuição dos itens com base nos seus gêneros, bem como com base no aspecto de popularidade. Como o objetivo desta pesquisa é baseado em dois tipos de justiça (gênero e popularidade), propomos neste trabalho usar a média harmônica (ou pontuação F1) entre MRMC de gêneros e popularidade, onde os valores mais altos são melhores:

$$F1 = 2\frac{(1 - MRMC\ Genre) * (1 - MRMC\ Pop)}{(1 - MRMC\ Genre) + (1 - MRMC\ Pop)} \tag{2.5}$$

- ***Long-Tail Coverage* (LTC).** Mede a proporção de itens pouco populares (cauda longa) efetivamente recomendados aos usuários, variando de 0 (apenas itens populares) a 1 (apenas itens de nicho pouco consumidos)(Abdollahpouri; Burke; Mobasher, 2018). A métrica é representada pela Equação 2.6, onde $\bigcup_{u \in Ut}$ é o conjunto de usuários da lista de recomendação, L_u é a lista de itens recomendados e Φ é o conjunto de itens pertencentes a cauda longa.

$$LTC = \frac{|(\bigcup_{u \in Ut} L_u) \bigcap \Phi|}{|\Phi|} \tag{2.6}$$

- ***Group Average Popularity* (Δ GAP).** Essa métrica avalia a variação da popularidade dos itens para os usuários, onde valores mais altos indicam maior preferência por itens de nicho (Abdollahpouri *et al.*, 2019). Variando em $[-1, \infty]$, valores negativos reduzem o viés de popularidade e positivos aumentam-no. Os usuários são divididos em grupos segundo suas preferências (itens populares, nicho ou ambos). A popularidade média dos itens do perfil do usuário u é calculada pela Equação 2.7, onde ϕ representa a popularidade de um item e pu a lista de itens do perfil do usuário.

$$GAP(g) = \frac{\sum_{u \in g} \frac{\sum_{i \in pu} \phi(i)}{|pu|}}{|g|} \tag{2.7}$$

A partir disso, é possível representar o valor do GAP para o perfil dos usuários por meio de $GAP(g)_r$ e o GAP das recomendações por meio de $GAP(g)_p$. Assim, é possível calcular a variação de popularidade, como representada na Equação 2.8.

$$\Delta GAP(g) = \frac{GAP(g)_r - GAP(g)_p}{GAP(g)_p} \tag{2.8}$$

Por fim, como os valores ótimos de ΔGAP devem ser próximos de zero, propomos neste trabalho a utilização do Root Mean Squared Error (RMSE) entre os três grupos de usuários, onde os valores mais baixos são melhores:

$$RMSE = \frac{\sqrt{\Delta GAP_{BB}^2 + \Delta GAP_N^2 + \Delta GAP_D^2}}{3} \tag{2.9}$$

- ***Normalized Discounted Cumulative Gain* (nDCG).** Utilizamos o *Normalized Discounted Cumulative Gain* (NDCG) para avaliar a qualidade da classificação das listas recomendadas. O NDCG compara o ganho cumulativo descontado do *ranking* produzido com o ganho de um *ranking* ideal, penalizando itens relevantes aparecendo em posições baixas (Jadon; Patil, 2024). O DCG no corte N é dado pela Eq. 2.10, normalizado pelo $IDCG@N$, que é o DCG obtido ao ordenar os itens por relevância decrescente (Eq. 2.11). O NDCG varia em $[0, 1]$, sendo 1 um *ranking* perfeito:

$$DCG@N = \sum_{i=1}^{N} \frac{2^{\mathrm{rel}_i} - 1}{\log_2(i+1)} \tag{2.10}$$

$$NDCG@N = \frac{DCG@N}{IDCG@N} \tag{2.11}$$

  onde i é a posição do item na lista e rel_i é sua relevância na posição i.

- ***Teoria da Utilidade Multiatributo* (MAUT).** A métrica MAUT permite condensar várias métricas de qualidade em um único escalar, facilitando a comparação global entre diferentes modelos de recomendação (Neiva; Gomes, 2007; Carvalho; Rocha, 2020).

  Seja $M = \{MAP,\ NDCG,\ RMSE,\ LTC,\ F1\ Score\}$ o conjunto de métricas consideradas. Para cada modelo e para cada métrica $j \in M$, denotamos por m_{ij} o valor bruto obtido.

  Como as métricas possuem escalas diferentes, primeiro aplicamos *min-max scaling* para torná-las comparáveis; se para alguma métrica os valores menores forem melhores, basta inverter o sinal antes de normalizar:

$$u_{ij} \;=\; \frac{m_{ij} - \min(m_j)}{\max(m_j) - \min(m_j)}, \tag{2.12}$$

  onde $\min(m_j)$ e $\max(m_j)$ representam, respectivamente, o pior e o melhor valor observado para a métrica j entre todos os métodos avaliados.

  Definindo um vetor de pesos $\mathbf{w} = (w_1, \ldots, w_{|M|})$ tal que $\sum_{j=1}^{|M|} w_j = 1$, a utilidade global do método i é:

$$U_i \;=\; \sum_{j=1}^{|M|} w_j \, u_{ij}. \tag{2.13}$$

  Quando não há preferência por nenhuma métrica específica, utiliza-se $w_j = \frac{1}{|M|}$, atribuindo importância igual a todas elas. Valores mais altos de U_i indicam melhor

capacidade do método em equilibrar, de forma simultânea, todos os critérios de qualidade considerados. No nosso cenário, todas terão os mesmos pesos. A métrica U_i situa-se no intervalo $[0, 1]$. Assim, um método com $U_i \approx 1$ aproxima-se do melhor desempenho observado em todas as métricas, enquanto valores próximos de 0 revelam fraco desempenho global. Essa métrica permite uma avaliação abrangente e alinhada às questões de pesquisa, uma vez que reflete diretamente os *trade-offs* entre precisão, diversidade e justiça, aspectos centrais deste trabalho.

Embora a acurácia seja o objetivo principal, outras características também devem ser consideradas na escolha da abordagem. No entanto, os sistemas de recomendação ainda enfrentam limitações, como os vieses. A próxima seção abordará os principais vieses e seus impactos.

2.4 Vieses e Calibração

Os sistemas de recomendação enfrentam o desafio do viés, que limita as interações ao apresentar apenas determinados itens, podendo gerar recomendações distantes das reais preferências dos usuários e prejudicar a eficácia dos sistemas (Elsweiler; Trattner; Harvey, 2017). Diversos vieses têm sido identificados na literatura (Chen *et al.*, 2023):

- **Viés de Confirmação.** Nesse viés, os usuários consomem itens conforme suas crenças prévias.
- **Viés de Conformidade.** As avaliações se ajustam ao comportamento de outros usuários.
- **Viés de Exposição.** O usuário visualiza apenas parte do feedback disponível.
- **Viés Indutivo.** O modelo incorpora informações dos dados de treinamento.
- **Viés de Posição.** A interação se concentra nos primeiros itens da lista.
- **Viés de Popularidade.** Esse viés faz com que os itens populares sejam recomendados mais frequentemente do que os itens não populares, diminuindo a serendipidade e personalização, além de afetar a experiência do usuário.
- **Viés de Seleção.** Ocorre quando o usuário avalia apenas os itens que já lhe agradam.

Esses vieses resultam do desequilíbrio de classes no aprendizado de máquina, gerando classificações injustas e impactando a consistência do desempenho entre diferentes grupos (Abdollahpouri *et al.*, 2020; Ekstrand *et al.*, 2018). Além disso, eles podem causar bolhas de filtro, isolando os usuários de diversas perspectivas e promovendo a polarização,

o que representa um risco à democracia (Nguyen *et al.*, 2014; Gelfert, 2018; Pariser, 2011; Lunardi *et al.*, 2020).

Para combater esse problema, estudos sugerem a implementação de algoritmos que promovam a diversidade e visualizações interativas que ampliem o leque de recomendações (Munson; Resnick, 2010; Helberger; Karppinen; D'acunto, 2018; Nagulendra; Vassileva, 2014). Apesar do foco na acurácia (Konstan; Riedl, 2012), é crucial que os sistemas considerem também aspectos como novidade, serendipidade e diversidade (Ricci; Rokach; Shapira, 2015).

Para combater a injustiça em sistemas de recomendação, uma estratégia eficaz é a calibração, que ajusta a geração das recomendações para que a proporção de itens reflita as reais preferências dos usuários (Steck, 2018). Segundo (Kaya; Bridge, 2019), a calibração pode ser aplicada tanto durante o processamento quanto no pós-processamento, permitindo sua implementação de forma independente ao sistema que gerou as recomendações.

2.5 Otimização de LLMs

A união entre sistemas de recomendação e LLMs foi explorada no trabalho (Zhao *et al.*, 2024), destacando como os LLMs aprimoram a compreensão do usuário ao analisar linguagem natural, geram recomendações ricas com explicações contextuais e permitem interações conversacionais, superando limitações tradicionais como o cold start. No entanto, o artigo também alerta para os desafios éticos e de privacidade, enfatizando a necessidade de um uso responsável dessa tecnologia em constante evolução. Apesar disso, o uso de LLMs em sistemas de recomendação tem crescido, especialmente para mitigar vieses. Há três abordagens principais:

- **Como modelo de recomendação:** LLMs podem ser discriminativos, reclassificando itens para reduzir vieses, ou gerativos, sugerindo novos itens que promovam diversidade e equidade (Kang *et al.*, 2023; Liu *et al.*, 2023).
- **Como aprimoradores:** Enriquecem as representações de itens e usuários com informações contextuais e semânticas ou geram dados adicionais para complementar modelos tradicionais, evitando favoritismos (Li *et al.*, 2023a; Li *et al.*, 2023b).
- **Como simuladores:** Criam ambientes interativos que modelam comportamentos de usuários e itens para detectar e corrigir fontes de viés, ajustando os sistemas para maior justiça e equilíbrio (Du *et al.*, 2024; Wang *et al.*, 2023).

Além disso, trabalhos como (Lichtenberg; Buchholz; Schwöbel, 2024; Sah; Xiaoli; Islam, 2024; Lin *et al.*, 2025; Hou *et al.*, 2024) mostram que apesar de também terem vieses nas recomendações, os LLMs apresentaram uma redução nos vieses em comparação

aos modelos tradicionais e também possibilitam reduzir ainda mais os vieses por meio de novas instruções inseridas nos prompts.

A otimização de prompts tem ganhado destaque por explorar LLMs para gerar soluções adaptativas. Três técnicas relevantes são o OPRO (Yang *et al.*, 2023a), o EvoPrompt (Guo *et al.*,) e o OptFormer (Chen *et al.*, 2022), que diferem no uso de trajetórias de aprendizado, exemplares e estratégias de exploração.

O OPRO utiliza o próprio LLM como otimizador, gerando soluções iterativas a partir de descrições textuais com um meta-prompt que armazena as melhores instruções. EvoPrompt, inspirado em algoritmos evolutivos, aplica mutações e cruzamentos para explorar o espaço de prompts, embora não use exemplares para orientar a otimização, o que pode limitar sua precisão. Por sua vez, o OptFormer treina um transformer com dados de otimização de hiperparâmetros, identificando padrões para guiar soluções de forma estratégica com base no histórico de otimizações.

Combinando essas estratégias, LLMs podem aumentar a diversidade, descoberta e novidade em sistemas de recomendação, reduzindo bolhas de filtro, polarização e injustiça. Sua flexibilidade para atuar como modelos, aprimoradores ou simuladores permite explorar diversas linhas de pesquisa e modelar características complexas dos usuários, gerando recomendações mais justas. Além disso, os LLMs podem aprimorar a transparência das recomendações, auxiliando técnicas de explicação e abordagens conversacionais.

2.6 Considerações Finais

Este capítulo apresentou os conceitos e métricas necessárias para a compreensão deste trabalho. Nota-se que, embora a acurácia seja um objetivo central, aspectos como diversidade, justiça e cobertura são igualmente importantes para garantir recomendações mais equilibradas e eficazes.

Mostramos também como a calibração se estabelece como uma estratégia fundamental para mitigar problemas como o viés de popularidade. Além disso, exploramos o papel dos LLMs, que introduzem novas possibilidades ao permitir representações semânticas mais ricas, interações em linguagem natural e mecanismos de otimização de *prompts*, aspectos que ampliam o potencial dos sistemas de recomendação. Dessa forma, esses elementos servem como base para a proposta desenvolvida no próximo capítulo, que busca avaliar a capacidade dos LLMs para produzir sistemas de recomendação mais justos, diversos e equilibrados.

3 PROPOSTA

Este capítulo descreve a metodologia adotada para a condução dos experimentos e a proposta de otimização de prompts aplicada a sistemas de recomendação baseados em LLM. Serão detalhadas as abordagens comparadas, incluindo métodos tradicionais, o modelo baseado em LLM e o processo de otimização por OPRO.

3.1 Justificativa

O Capítulo 2 mostrou que, apesar dos avanços em *calibração* dos sistemas tradicionais, os LLMs e a otimização de prompts podem colaborar na evolução dos sistemas de recomendação principalmente por conta dos seguintes fatores (Lin *et al.*, 2025):

- Extraem *representações semânticas ricas* dos perfis de usuário e metadados de itens, possibilitando ajustes finos na distribuição recomendada;
- Uso de interface por meio de linguagem natural possibilita uma maior facilidade para solicitar as recomendações conforme necessidade;
- Por conta do pré-treino massivo, funcionam em domínios pouco rotulados e permitem lidar com o problema de *cold-start*, que é quando não se tem muitas informações históricas sobre o domínio, e *long-tail*, que é a capacidade de recomendar itens menos populares;

Objetivo específico. Diante deste cenário, este trabalho propõe uma análise de LLMs e otimização de prompts no processo de calibração de Sistemas de Recomendação. A proposta será avaliada em comparação direta com calibradores tradicionais, tais como abordagens baseadas em regularização da popularidade (Abdollahpouri *et al.*, 2021), calibração por gênero (Steck, 2018) e estratégias de reordenação em múltiplas etapas (Souza, 2024), utilizando as métricas apresentadas no Capítulo 2 a fim de verificar se a métrica MAUT apresenta diferença estatisticamente significativa em relação as abordagens tradicionais de forma a trazer um desempenho mais balanceado em termos de popularidade, precisão e diversidade das recomendações.

3.2 Estratégia de Calibração

A literatura mostra que uma forma de trazer recomendações mais coerentes para um Sistema de Recomendação é por meio da calibração, sendo que esse processo pode ser feito por meio de três estratégias: calibração em etapa de pré-processamento. calibração em

etapa de processamento e calibração em etapa de pós-processamento (Pitoura; Stefanidis; Koutrika, 2022).

O trabalho (Souza, 2024) apresentou modelos de recomendação baseados em estratégias de calibração em etapa de processamento (BPR e BPR com calibração) e em etapa de pós-processamento ((Steck, 2018) e calibração por gênero e por popularidade). Essas mesmas abordagens foram implementadas nesse trabalho, sendo adaptadas para aplicação nas abordagens tradicionais em comparação com as recomendações geradas por meio de LLMs e otimização de prompts.

Supondo que há um conjunto de itens $I = \{i_1, i_2, ..., i_{|I|}\}$, um conjunto de usuários $U = \{u_1, u_2, ..., u_{|U|}\}$ e um conjunto de itens candidatos para cada usuário $CI_u = \{i_1, i_2, ..., i_N\}$, onde N é o número de itens sugeridos pelo sistema de recomendação. Além disso, existem as informações dos usuários sobre as preferências de popularidade. A tarefa é explorar essas preferências para gerar uma lista de recomendações que aumente a justiça em relação a popularidade dos itens.

Para tanto, propõe-se uma abordagem de calibração, na prática, o método utiliza medidas de divergência na etapa de geração de recomendações para realizar uma calibração de acordo com diferentes níveis de popularidade de interesse do usuário. Como resultado, os usuários recebem uma lista de recomendações próxima ao seu perfil de interesse em termos de popularidade.

3.2.1 Divisão de Popularidade

A calibração da lista de recomendações com base na popularidade dos itens já consumidos pelo usuário é feita por meio de uma divisão de popularidade para agrupar os itens com base na quantidade de avaliações recebidas na base de dados. A divisão de popularidade, introduzida em (Abdollahpouri *et al.*, 2021), é baseada no conceito de cauda longa dos sistemas de recomendação, conforme pode ser visualizado na Figura 1. A curva foi dividida em três partes. O **Head (H)**, com itens representando 20% do total de interações. A **Tail (T)** com itens que somam menos de 20% das interações, e o grupo **Mid (M)**, que contém itens que não são nem **Head (H)** nem **Tail (T)**. Vale ressaltar que esta divisão por percentual foi escolhida com base no princípio de Pareto (Newman, 2005).

3.2.2 Calibração

A calibração por popularidade foi uma adaptação da fórmula proposta por (Steck, 2018). Seu trabalho pressupõe que os itens podem ter mais de um gênero, o que não é válido no contexto de popularidade, onde um item possui apenas um nível de popularidade. Então, ao invés disso, foram calculadas as somas dos pesos de cada tipo de popularidade sobre a soma de todos os pesos.

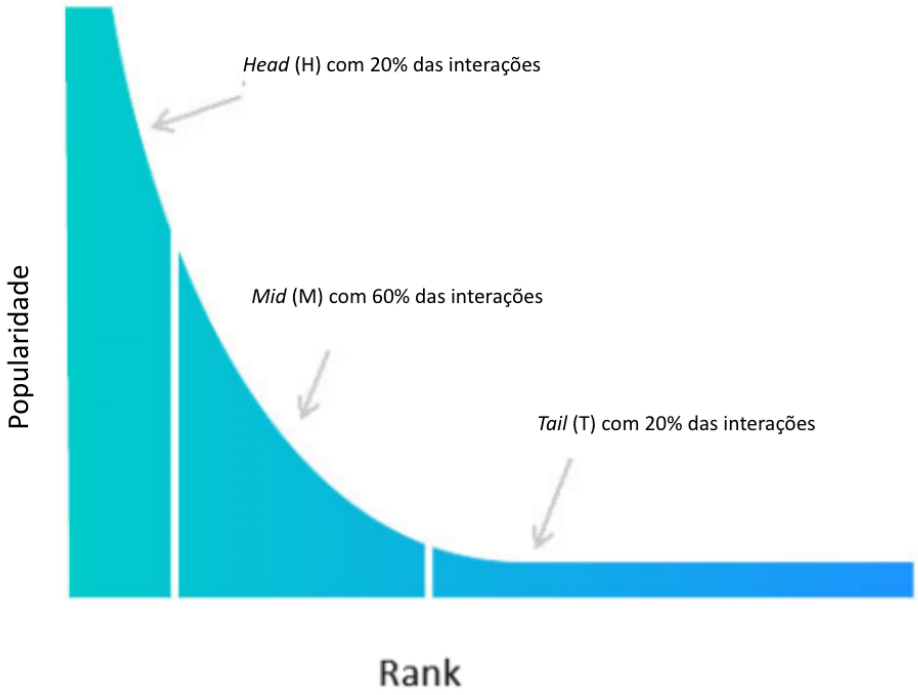

Figura 1 – Curva representando a divisão dos itens em grupos de popularidade.

Assim, $x(t|u)$ é definido como a distribuição alvo baseada na popularidade dos itens com os quais o usuário interagiu no passado. Na Equação 3.1 os pesos r_{ui} são definidos como a classificação explícita ou implícita que o usuário u deu ao item i:

$$x(t|u) = \frac{\sum_{i \in I_u} r_{ui} \cdot x(t|i)}{\sum_{i \in I_u} r_{ui}} \tag{3.1}$$

onde I_u é o conjunto de itens interagidos pelo usuário u, e $x(t|i)$ é definido como 1 se o item i estiver na categoria de popularidade t. Então, para lidar com a distribuição de lista recomendada, a Equação 3.2 define $y(t|u)$ como:

$$y(t|u) = \frac{\sum_{i \in R_u^*} w_p(u,i) \cdot x(t|i)}{\sum_{i \in R_u^*} w_p(u,i)} \tag{3.2}$$

Neste caso, usamos os pesos $w_p(u,i)$ como a posição de classificação do item i na lista reordenada recomendada R_u^* para o usuário u.

Utiliza-se a medida de divergência Kullback-Leibler pelas mesmas razões apontadas por (Steck, 2018) e exploradas por (Souza; Manzato, 2024). O Kullback-Leibler quantifica a desigualdade no intervalo $[0, \infty]$, onde 0 significa que ambas as distribuições são quase iguais e valores mais altos indicam injustiça.

Adicionalmente, é adotada a regularização proposta por (Steck, 2018), que definiu $\alpha = 0.01$ como uma variável de regularização para evitar divisão por zero quando $y(t|u)$

vai para zero.

$$D_{KL}(x\|y) = \sum_t x(t|u) \cdot log \frac{x(t|u)}{(1-\alpha) \cdot y(t|u) + \alpha \cdot x(t|u)} \tag{3.3}$$

A divergência de Kullback-Leibler é uma medida que quantifica a diferença entre duas distribuições de probabilidade, neste caso, entre a distribuição observada $x(t|u)$ e a distribuição de referência $y(t|u)$. No contexto da calibração por popularidade, $x(t|u)$ representa a distribuição empírica dos itens observados pelo usuário u, enquanto $y(t|u)$ representa uma distribuição de referência desejada, que é baseada na popularidade dos itens na base de dados.

3.3 Abordagens Tradicionais

3.3.1 O Método BPR

O método BPR (Rendle *et al.*, 2012) visa aprender vetores latentes de usuários e itens que expressem suas preferências particulares. Para isso, o algoritmo otimiza uma função de perda que recompensa, para cada usuário, a ordenação correta entre itens preferidos (positivos) e não-preferidos (negativos). Essa otimização é conduzida por gradiente descendente estocástico, que ajusta gradualmente os vetores latentes de usuários e itens a partir dos gradientes dessa função.

Nesse modelo, um usuário é indicado como u e um item é referido como i, j; r_{ui} refere-se ao feedback explícito ou implícito de um usuário u para um item i. No primeiro caso, é um número inteiro fornecido pelo usuário indicando o quanto ele gostou do conteúdo; no segundo caso, é apenas um booleano mostrando se o usuário consumiu ou visitou o conteúdo ou não. A predição do sistema sobre a preferência do usuário u para o item i é representada por $\hat{r}_{ui}$, que é um valor de ponto flutuante estimado pelo algoritmo de recomendação. O conjunto de pares (u, i) para os quais r_{ui} é conhecido é representado por $K = \{(u, i)|r_{ui}\}$.

Em um modelo de fatorização tradicional, cada usuário u é associado a um vetor de fatores $p_u \in \mathbb{R}^f$ e cada item i com um vetor de fatores $q_i \in \mathbb{R}^f$. Uma regra de previsão seria:

$$\hat{r}_{ui} = p_u^T q_i \quad (1) \tag{3.4}$$

Seja $N(u)$ o conjunto de itens para os quais o usuário u já forneceu algum tipo de feedback implícito, e $\overline{N}(u)$ o conjunto complementar, isto é, itens ainda não observados pelo mesmo usuário. Um aspecto crucial desse cenário é que apenas interações positivas são registradas; pares usuário-item ausentes são interpretados como evidência negativa.

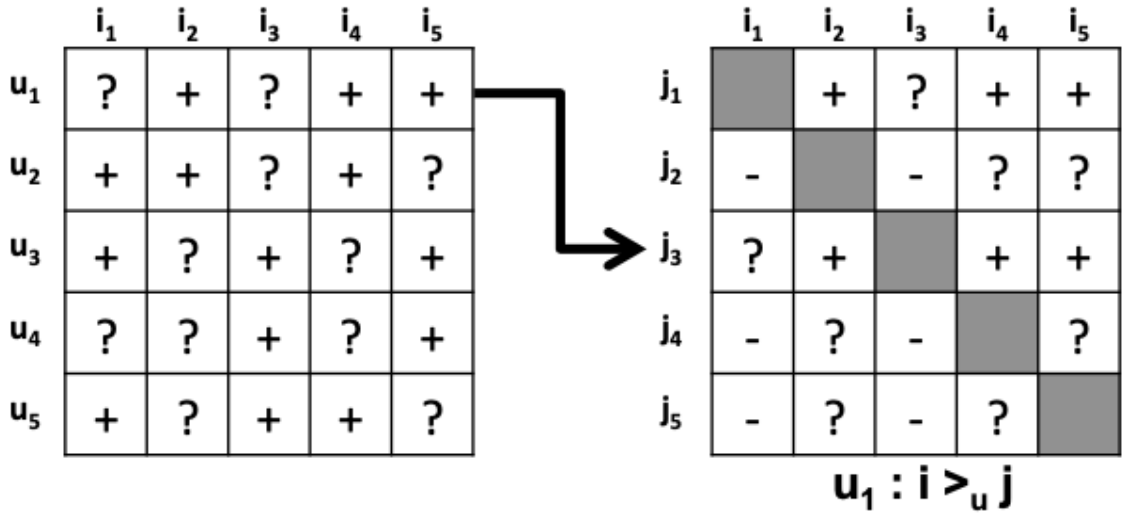

Figura 2 – O quadro à esquerda representa os dados observados. A abordagem cria uma relação par de itens específica para o usuário $i \succ_u j$ entre dois itens. No lado direito da tabela, o sinal de mais indica que o usuário u está mais interessado no item i do que no item j; o sinal de menos indica que o usuário prefere o item j ao i; o ponto de interrogação indica que não se pode inferir nenhuma conclusão entre os itens.

Em (Rendle *et al.*, 2012), discute–se um efeito indesejado que surge quando o modelo é ajustado exclusivamente com essa divisão "observado = positivo / ausente = negativo". Como apenas itens de $N(u)$ recebem pontuações elevadas, o algoritmo tende a empurrar as predições dos demais itens para zero, impossibilitando ordenar adequadamente aqueles que poderiam interessar ao usuário.

Para contornar esse problema, os autores propuseram um procedimento geral de aprendizado de preferências baseado em ordenação. Em vez de considerar só os pares (u, i), avalia-se também a relação relativa entre dois itens quaisquer para o mesmo usuário. Se $i \in N(u)$ e $j \in \overline{N}(u)$, então assume-se que u prefere i a j. A Figura 2 ilustra essa ideia.

Para estimar essa preferência relativa, (Rendle *et al.*, 2012) uma análise Bayesiana, formulando a probabilidade $\text{prob}(i \succ_u j \mid u, \Theta)$ e atribuindo a probabilidade anterior para o parâmetro do modelo $\text{prob}(\Theta)$. O critério de otimização resultante, denominado *BPR-Opt* é:

$$BPR\text{-}Opt := \sum_{(u,i,j) \in S_K} \ln \sigma(\hat{s}_{uij}) - \Lambda_\Theta \|\Theta\|^2$$

onde $\hat{s}_{uij} := \hat{r}_{ui} - \hat{r}_{uj}$ e S_K é o conjunto de triplas (u, i, j) onde i está em $N(u)$ e j não está. O símbolo Θ representa os parâmetros do modelo, Λ_Θ é o conjunto de constantes de regularização, e σ é a função logística definida como $\sigma(x) = \frac{1}{1+e^{-x}}$.

Os autores também propuseram uma variação na técnica de descida de gradiente estocástico, denominada LearnBPR, que amostra aleatoriamente de S_K para ajustar Θ. O Algoritmo 1 mostra uma visão geral do método de aprendizagem, onde α é a taxa de aprendizado.

Algoritmo 1: Aprendizado via LearnBPR.

Input: D_K
Output: Parâmetros ajustados Θ
Inicializar Θ com valores aleatórios
for *cont = 1,...,#Iterações* **do**
  obtenha (u, i, j) a partir de S_K
  $\hat{s}_{uij} \leftarrow \hat{r}_{ui} - \hat{r}_{uj}$
  $\Theta \leftarrow \Theta + \alpha \left(\frac{e^{-\hat{s}_{uij}}}{1+e^{-\hat{s}_{uij}}} . \frac{\partial}{\partial \Theta} \hat{s}_{uij} - \Lambda_\Theta \Theta \right)$
end

No presente estudo, definimos a abordagem BPR para considerar a regra de predição $\hat{r}_{ui}$ do modelo de fatorização simples definido na Equação 3.4. Portanto, aplicar a Equação 3.4 em $\hat{s}_{uij}$ resulta em $\Theta = \{p_u, q_i, q_j\}$, que devem ser aprendidos. Calculamos as derivadas parciais em relação a $\hat{s}_{uij}$:

$$\frac{\partial}{\partial \Theta} \hat{s}_{uij} = \begin{cases} q_i - q_j & \text{quando } \Theta = p_u \\ p_u & \text{quando } \Theta = q_i \\ -p_u & \text{quando } \Theta = q_j \\ 0 & \text{caso contrário} \end{cases}$$

Esses gradientes são então usados para atualizar os fatores de usuário e item em direção ao mínimo da função de perda, iterativamente, até que a convergência seja alcançada ou um número fixo de iterações seja concluído. Desse modo, o SGD permite ajustar os fatores de usuário e item de forma a maximizar a diferença entre as pontuações dos itens positivos e negativos, resultando em recomendações mais precisas e personalizadas.

3.3.2 BPR com Calibração por Popularidade

Para incorporar a calibração de popularidade já na fase de aprendizado propõe-se modificar o LearnBPR (Algoritmo 1), como feito em (Souza; Manzato, 2024). A ideia é acrescentar, à atualização dos vetores de usuário, um termo de divergência de Kullback-Leibler (usado na calibração de popularidade), de modo que a função de perda passe a penalizar rankings que se afastem do perfil de popularidade desejado.

A única mudança ocorre quando o parâmetro atualizado é $\Theta = p_u$, resultando em:

$$
\begin{aligned}
p_u \leftarrow p_u + \alpha \left(\frac{e^{-\hat{s}_{uij}}}{1 + e^{-\hat{s}_{uij}}} . (q_i - q_j) \right. \\
\left. + \lambda \left(1 - \frac{D_{KL}(x \| y)}{D_{KLvoid}} \right) - \Lambda_{p_u} p_u \right)
\end{aligned}
\tag{3.5}
$$

onde λ é utilizado como coeficiente do impacto que a divergência terá no sistema, e D_{KLvoid} é definido como:

$$
D_{KLvoid} = \sum_t x(t|u) \cdot log \frac{x(t|u)}{\alpha \cdot x(t|u)} \tag{3.6}
$$

Com isso, o modelo passa a considerar simultaneamente a ordem implícita de preferência entre itens e a distância entre a distribuição de popularidade recomendada e a observada, promovendo listas mais justas e, potencialmente, uma experiência de recomendação mais equilibrada e satisfatória para o usuário.

3.3.3 Popularity

Esse trabalho (Sacilotti; Souza; Manzato, 2023) apresenta uma abordagem de calibração das recomendações com base na popularidade dos itens e no nível de interesse dos usuários por esse aspecto e por isso foi incluído em nossos experimentos, o que possibilita avaliar separadamente a nossa proposta com um trabalho com calibração somente em popularidade.

3.3.4 Personalized

Uma abordagem de calibração personalizada das recomendações para cada usuário (Sacilotti; Souza; Manzato, 2023), o qual pode receber uma lista recomendada com base na proporção de interesse nos gêneros ou na popularidade dos itens, dependendo se o interesse dele na popularidade está acima de um limite definido. Foi selecionado para ser comparado por conta de possibilitar uma análise em relação a seleção de calibração por gênero ou por popularidade.

3.3.5 Steck - Calibração por gêneros

Trabalho clássico de calibração por conteúdo com base nos gêneros dos itens para alinhar a distribuição de categorias do ranking ao perfil do usuário (Steck, 2018). Foi selecionado como uma alternativa a calibração por popularidade e poder validar comparar com a nossa proposta em relação a esse aspecto.

3.3.6 Two-Stage

Abordagem em duas etapas de calibração, aplicando primeiro a calibração por popularidade e depois a calibração com base nos gêneros (Souza; Manzato, 2024), repre-

sentando estratégias que combinam múltiplos objetivos de balanceamento. Permite uma comparação com uma abordagem que calibra as recomendações com base no gênero e na popularidade.

3.3.7 Abdoullahpouri

Baseline inspirado em avaliações centradas no usuário para lidar com o viés de popularidade (Abdollahpouri *et al.*, 2021), incluído como representante de intervenções voltadas a reduzir a exposição desbalanceada entre itens populares e de nicho. Utiliza uma medida de divergência diferente da implementada pelo *baseline Popularity*.

3.4 Recomendação baseada em LLM

Em nossa proposta de recomendação com LLM, adotamos o *Llama3.1-8b-instruct*, quantizado em 4 *bits*, o que reduz significativamente o consumo de memória da GPU sem causar prejuízos relevantes no desempenho (Hu *et al.*, 2021). Optamos por esse modelo por ser aberto e amplamente utilizado na literatura (Fonseca *et al.*, 2025; Cunha; Rocha; Gonçalves, 2025; Fonseca *et al.*, 2024). O procedimento adotado consistiu na solicitação de recomendações personalizadas por meio do *prompt* apresentado na Figura 3, instanciado para uma coleção de dados relacionada a filmes. Esse *prompt* é composto por duas partes: o `System Prompt`, que fornece o contexto ao modelo; e o `User Prompt`, que representa a entrada fornecida ao modelo, composta por uma solicitação de recomendação personalizada construída a partir dos 20 itens previamente consumidos pelo usuário, extraídos do conjunto de treino. As saídas geradas pelo modelo foram limitadas a até 1000 novos *tokens*, com amostragem ativada e configuração de *temperature* para 0.7 e *top-p* em 0.9. O parâmetro *use_cache* foi mantido ativado para otimizar o desempenho da inferência.

Dado o caráter livre da linguagem natural gerada por LLMs, podem surgir desafios relacionados à padronização e interpretação das respostas. Para garantir a conformidade com o conjunto de dados a serem utilizados nos experimentos, nossa abordagem precisa de etapas de pré-processamento e filtragem. Um dos principais possíveis problemas é a diferença entre descrições de itens na coleção com a descrição retornada pelo LLM. Por exemplo, em uma coleção de filmes, como a utilizada em nossa avaliação experimental (i.e. MovieLens), pode haver diferenças nas representações dos títulos dos filmes. Títulos iniciados com o artigo “The” são frequentemente registrados no conjunto de dados com o artigo ao final, como em “Godfather, The (1972)”, enquanto o LLM normalmente retorna no formato canônico “The Godfather (1972)”, o que pode gerar conflitos na validação. Assim, foi implementada uma etapa de normalização para tratar essas variações, assegurando que recomendações semanticamente equivalentes não fossem descartadas.

Em cenários onde os itens recomendados são baseados em descrições (e.g. filmes, pontos de interesse, etc.), outro tipo comum de erro está relacionado a pequenas variações

***Prompt* Inicial**

System Prompt:
Given movies/tv shows titles, provide the recommendations following pattern:
1. title (release year)
2. title (release year)
...
User Prompt:
I need exactly 10 movies or TV shows (based on the MOVIELENS 1M dataset) and your recommendations must be based on cast members.
Follow this strict format and do not include any explanations, duplicates, or corrections in your response:
1. title (release year)
2. title (release year)
3. title (release year)
...
I've watched:
{movies}

Figura 3 – *Prompt* para gerar recomendações. O campo `{movies}` é substituído pelos filmes assistidos pelo usuário.

na grafia dos itens, como "Pleasentville (1998)" e "Pleasantville (1998)" no cenário de filmes. Apesar de referirem-se ao mesmo filme, essas diferenças impedem o reconhecimento da correspondência. Para lidar com esses casos, propomos a utilização de um algoritmo de comparação de similaridade baseado na distância de *Levenshtein*, capaz de identificar e dimensionar discrepâncias ortográficas. Além desses casos, nossa abordagem também está preparada para lidar com outros tipos de erros, como a presença de itens com múltiplas descrições ou a repetição de recomendações para um mesmo usuário. Além disso, recomendações que não correspondessem a itens presentes no conjunto de dados foram descartadas. Por fim, o procedimento de solicitação é iterado até que sejam obtidas 10 recomendações válidas por usuário ou até o limite de cinco iterações consecutivas sem sucesso. Com o conjunto final de recomendações, realiza-se a comparação com o conjunto de teste individual de cada usuário, possibilitando a avaliação de desempenho do modelo. Desse modo, nossa abordagem permite validar as recomendações geradas pelo LLM da forma mais precisa possível, minimizando inconsistências decorrentes da linguagem natural.

3.4.1 Processo de Otimização de Prompt

A sensibilidade dos LLMs ao *prompt* é uma questão crucial que precisa ser considerada durante sua utilização, uma vez que *prompts* semanticamente semelhantes podem resultar em desempenhos significativamente diferentes (Yang *et al.*, 2023b). Dessa forma, a engenharia de *prompt* tornou-se uma etapa fundamental para obter o melhor desempenho desses modelos. Para lidar com esse desafio, os autores de (Yang *et al.*, 2023b) propõem o método *OPRO* (***O**ptimization by **PRO**mpting*), que utiliza o próprio LLM para otimizar o *prompt* a ser utilizado em uma tarefa.

O *OPRO* transforma os LLMs em otimizadores iterativos baseados em linguagem natural. Nesse processo, o problema de otimização, como encontrar um *prompt* que maximize a acurácia em determinada tarefa, é descrito textualmente e apresentado ao modelo

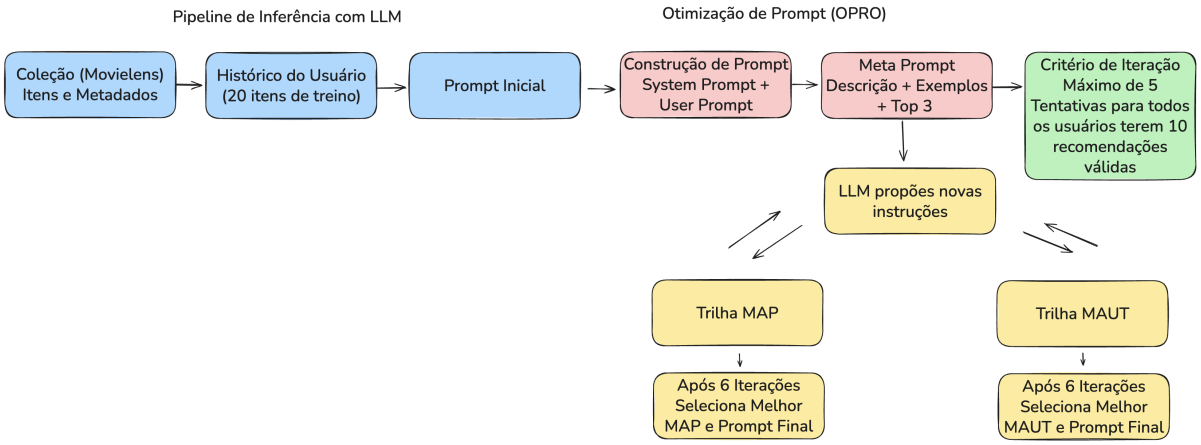

Figura 4 – Figura representando o processo de otimização de prompt.

por meio de um *meta-prompt*. Esse *meta-prompt* inclui três elementos principais: uma descrição da tarefa, exemplos ilustrativos e um histórico das instruções previamente geradas com seus respectivos desempenhos. A cada iteração, o LLM propõe novas instruções, que são avaliadas e reincorporadas ao *meta-prompt*. O ciclo se repete até que não haja mais melhorias relevantes. Os resultados mostram que o *OPRO* gera *prompts* que superam instruções criadas manualmente.

Assim, inspirados nesse trabalho, propomos uma adaptação do método *OPRO* para o contexto de sistemas de recomendação. Nossa proposta parte de um *prompt* inicial, elaborado manualmente (e.g. Figura 3), cuja efetividade é avaliada por uma métrica específica. A partir dessa avaliação, iniciamos um processo iterativo de refinamento por meio de um *meta-prompt*, que inclui a descrição da tarefa, que consiste em gerar instruções para orientar a geração de recomendações, e um conjunto de instruções previamente criadas, acompanhadas de seus respectivos desempenhos. Esse processo também é conduzido pelo LLM *Llama3.1-8b-instruct*, quantizado em 4 *bits*. A cada iteração, o modelo recebe o *meta-prompt* como entrada e propõe novas instruções. Na primeira iteração, como ainda não há histórico, o *meta-prompt* inclui apenas a descrição da tarefa e o *prompt* inicial. A partir da segunda, as melhores instruções geradas anteriormente passam a ser incorporadas ao *meta-prompt*, enriquecendo o contexto fornecido ao modelo com o objetivo de que o refinamento ocorra progressivamente ao longo das etapas. A geração foi configurada para produzir até 1000 novos *tokens*. A amostragem foi ativada, com temperatura 1.6, *top-p* igual a 0.9 e *top-k* limitado a 40, controlando a aleatoriedade e a diversidade da saída.

O número total de iterações, a quantidade de novas instruções geradas em cada rodada e o número de instruções mantidas no *meta-prompt* são definidos como parâmetros do processo. Neste trabalho, realizamos seis iterações; em cada uma delas, o modelo gera quatro novas instruções, que são avaliadas individualmente e, em seguida, comparadas

System Prompt:
Your task is to create a clear and effective instruction for a movie recommendation task using the MovieLens 1M dataset. The instruction you generate will be used by a model to recommend movies based on user preferences, past ratings, or relevant contextual signals such as genre affinity, viewing history, or similar user behavior.

To help you write a stronger instruction, we provide examples of previous instructions along with quality scores. These are ordered from worst to best in terms of recommendation relevance, clarity, and alignment with the dataset. Use them as inspiration, but do not copy them.

{instructions}

Your instruction must explicitly include the following constraints:
- The output must be a numbered list in this exact format:
  1. title (release year)
  2. title (release year)
  3. title (release year)
  ...
- The model must output **only** the list — no additional text, explanations, or commentary.

To maximize recommendation quality:
- Encourage the model to consider patterns in user preferences, ratings, and genres.
- Make it clear that the recommendations should be personalized and relevant to the user context.
- Ensure that the instruction clearly connects the task to the MovieLens 1M dataset (which includes user ratings, genres, and timestamps).
- Encourage the model to suggest items that are both relevant and meaningfully ordered according to the user's preferences.
- Balance recommendations to not overly favor the most popular items, promoting a more diverse and personalized experience.
- Ensure that the instruction leads the model to capture deeper patterns in user behavior, including preferences across popularity levels and genre variety.
- The goal is to produce recommendations that are not only accurate, but also calibrated, diverse, and well-aligned with each user's unique profile.

Focus on crafting an instruction that is practical, specific, and optimized for generating relevant, high-quality recommendations in the correct format.
Focus on crafting an instruction that is practical, specific, and optimized for generating high-quality recommendations in the correct format.

User Prompt:
Create a new instruction that is concise and highly effective for the recommendation task using the MovieLens 1M dataset.
The instruction must explicitly state that the model's response should consist only of a numbered list of movie titles in the format:
1. title (release year)
2. title (release year)
3. title (release year)
...
It must also make clear that no additional text, explanations, or extra information should be included in the output. Ensure that the instruction is presented directly, without mentioning that it is being generated or evaluated, and do not include any reference to performance metrics or scoring.

Figura 5 – *Meta-prompt* para otimização, com trechos comuns e variações específicas destacadas para as métricas *MAP* (em azul) e *MAUT* (em laranja). O campo `{instructions}` é substituído, em cada iteração, pelas melhores instruções geradas até o momento, e suas respectivas pontuações.

com as três melhores instruções já presentes no *meta-prompt*. A partir disso, as três com melhor desempenho são selecionadas para compor o *meta-prompt* da próxima iteração. A métrica utilizada na avaliação é a mesma aplicada ao *prompt* inicial e permanece constante ao longo de todo o processo.

Um aspecto fundamental da proposta é que as instruções geradas são utilizadas como conteúdo do campo `system` do *prompt* final. Assim, estamos otimizando o próprio contexto fornecido ao LLM, buscando instruções que o guiem da forma mais eficaz possível

na geração das recomendações. Logo, a otimização ocorre sobre a formulação da tarefa e influencia diretamente o comportamento do modelo.

Para avaliar a eficácia do método, apresentamos os *meta-prompts* utilizados em duas estratégias de otimização distintas. Na primeira, adotamos a métrica *MAP* como critério principal, selecionando ao final das iterações a instrução que alcançou o maior valor nessa métrica. Na segunda estratégia, o *meta-prompt* é baseado na métrica *MAUT* (Carvalho; Rocha, 2020), que agrega os indicadores *MAP*, *LTC*, *RMSE*, *NDCG@10* e *F1-score* em uma única medida composta (todas as métricas estão detalhadas na Seção 4.2). Para isso, todas as métricas foram normalizadas, com inversão do *RMSE* (uma vez que valores menores são preferíveis), e calculada sua média aritmética. Como o *MAUT* é sensível à composição do conjunto avaliado devido à normalização, ele foi recalculado a cada iteração considerando as sete instruções em avaliação (as três melhores previamente incluídas no *meta-prompt* e as quatro geradas na iteração atual) garantindo assim uma comparação justa e consistente. Ao final, selecionamos a instrução com maior valor de MAUT dentro da iteração em que essa métrica atingiu seu valor máximo, uma vez que, diferentemente do MAP, seus valores não são diretamente comparáveis entre iterações. A Figura 5 ilustra nossa proposta para o cenário de filmes.

Além das métricas, realizamos uma avaliação preliminar para observar o impacto do perfil dos usuários no processo de otimização. Em uma das versões, selecionamos aleatoriamente 100 usuários, sendo que 89 deles já recebiam 10 recomendações com o *prompt* inicial. Na outra, os 100 usuários escolhidos não atingiam esse mínimo. Essa diferenciação nos permitiu observar como tanto o critério de avaliação quanto as características do conjunto de usuários podem influenciar o resultado final. Ao todo, foram realizadas quatro execuções, variando entre a métrica utilizada e o perfil dos usuários selecionados, cada uma resultando em uma instrução final otimizada, utilizada posteriormente para a geração de recomendações.

No próximo Capítulo será detalhado como os experimentos foram executados e os resultados obtidos para cada modelo avaliado.

3.5 Base de Dados

A base de dados utilizada no experimento foi a MovieLens 1M, que contém 1.000.209 avaliações de 6.040 usuários em 3.952 filmes, com notas variando de 1 a 5. Foram removidos os usuários com menos de 30 interações, resultando em um conjunto final com 5.289 usuários e 3.883 filmes. Para cada abordagem foram realizadas 6 execuções independentes. No caso do OPRO, cada execução envolveu 6 iterações de otimização, com avaliação de 7 prompts candidatos por rodada (3 herdados + 4 novos). Todas as recomendações foram comparadas com os conjuntos de teste dos usuários para cálculo das métricas *MAP, NDCG@10, LTC, F1 Score, RMSE e MAUT.*

3.6 Considerações Finais

Este capítulo apresentou a proposta deste trabalho, fundamentada na utilização de LLMs aliados a técnicas de calibração e à otimização de *prompts*. Foram descritos tanto os métodos tradicionais de calibração, bem como a incorporação do processo iterativo de otimização de *prompts*. Dessa forma, a proposta não apenas amplia o espaço de comparação com técnicas consolidadas, mas também introduz uma abordagem para alinhar a distribuição de recomendações às preferências dos usuários.

Espera-se que a combinação entre calibração e LLMs, especialmente com o apoio de um processo de engenharia e otimização de *prompts*, resulte em sistemas de recomendação mais equilibrados entre precisão, diversidade e justiça. No próximo capítulo, será detalhado a execução dos experimentos comparativos, permitindo avaliar seu desempenho em relação aos modelos de referência.

4 AVALIAÇÃO EXPERIMENTAL

Neste capítulo, apresentamos os resultados obtidos, divididos em duas subseções, cada uma diretamente relacionada às questões de pesquisa previamente definidas. Todos os experimentos relatados nesta seção foram executados com seis repetições visando garantir a consistência nos resultados obtidos. Para assegurar a robustez das comparações, aplicamos o teste estatístico não paramétrico de Wilcoxon (Rey; Neuhäuser, 2011), com nível de significância de 5%, para verificar se as diferenças entre os métodos são estatisticamente significativas.

4.1 RQ1: LLM versus Métodos Tradicionais

Com relação às métricas de **precisão** (MAP e NDCG@10), a estratégia baseada em LLM apresentou ganhos expressivos em comparação aos métodos tradicionais, conforme ilustrado na Tabela 1. O MAP médio dos métodos tradicionais variou entre 0,008 e 0,037, ao passo que a estratégia LLM atingiu 0,059. Resultados similares foram observados para o NDCG@10, com valores entre 0,003 e 0,013 nos métodos tradicionais e 0,021 na estratégia LLM. Esses números indicam que a estratégia baseada em LLM é mais eficaz em posicionar itens relevantes nas primeiras posições da lista de recomendação, fator crítico para a experiência do usuário.

Modelo	**MAP**	**NDCG**	**LTC**	**F1 Score**	**RMSE**	**MAUT**
Métodos Tradicionais						
Popularity (Sacilotti; Souza; Manzato, 2023)	0,027	0,010	0,048	**0,539**	0,210	0,511
Personalized (Sacilotti; Souza; Manzato, 2023)	0,035	0,013	0,080	0,257	1,498	0,287
Steck (Steck, 2018)	0,037	0,013	0,075	0,243	1,130	0,320
Two-stage (Souza; Manzato, 2024)	0,034	0,013	0,080	0,266	0,960	0,342
Abdoullahpouri (Abdollahpouri *et al.*, 2021)	0,026	0,010	0,046	0,537	1,115	0,412
BPR (Rendle *et al.*, 2012)	0,008	0,003	**0,542**	0,502	0,623	0,472
BPR Calibrado (Souza; Manzato, 2024)	0,008	0,003	0,530	0,499	**0,044**	0,527
Modelo baseado em LLM (Llama)						
Sem otimização	**0,059**	**0,021**	0,517	0,296	0,367	**0,741**

Tabela 1 – Comparação entre modelos tradicionais e o modelo baseado em LLM (LLaMa). Os melhores valores por métrica estão em negrito. Todos os resultados apresentaram significância estatística ($p\text{-}value < 0{,}05$), conforme o teste de Wilcoxon.

No que tange à **diversidade** e **cobertura**, mensuradas pela métrica LTC, a estratégia LLM também se destacou. Embora alguns métodos tradicionais, como o BPR e BPR Calibrado, tenham alcançado os melhores valores de LTC, o LLM ainda obteve um desempenho elevado (0,517), demonstrando sua capacidade de conciliar precisão com maior variedade e alcance nas recomendações, equilibrando adequadamente o *trade-off* entre

precisão e diversidade (Zanon; Rocha; Manzato, 2022). Isso evidencia seu potencial para mitigar a concentração em itens populares, ampliando a exposição a conteúdos diversos.

Quanto à métrica **RMSE**, que avalia o desequilíbrio de popularidade entre grupos (quanto menor, melhor), o LLM superou a maioria dos métodos tradicionais, cujos valores oscilaram entre 0,044 e 1,498. Embora o BPR Calibrado e o Popularity tenham registrado RMSE inferiores, ambos apresentaram sérios compromissos em outras dimensões: o BPR Calibrado teve os menores valores de MAP e NDCG@10, enquanto o Popularity exibiu cobertura limitada e precisão modesta (MAP de 0,027). Já o LLM conseguiu manter um RMSE competitivo sem abrir mão da qualidade do ranking e da diversidade, como refletido também em seu alto valor de MAUT.

No aspecto da **justiça**, o LLM apresentou desempenho inferior aos modelos tradicionais, com um F1 Score de 0,296, ante valores entre 0,243 e 0,539 nos demais métodos. Esse resultado ilustra o conhecido *trade-off* entre diversidade e justiça (Treuillier *et al.*, 2024; Zhao *et al.*, 2025). Ainda assim, ao se considerar a métrica MAUT — que integra múltiplas dimensões como precisão, diversidade e justiça — o modelo LLM demonstrou desempenho notavelmente superior. Enquanto os métodos tradicionais variaram entre 0,287 e 0,527, o LLM alcançou 0,741, sinalizando uma solução mais equilibrada e robusta.

Assim, em resposta à RQ1, os resultados indicam que estratégias baseadas em LLM superam consistentemente os métodos tradicionais em precisão e diversidade, mantendo níveis competitivos de justiça. Sua capacidade de otimizar múltiplos objetivos simultaneamente torna-as alternativas mais eficazes e balanceadas para sistemas de recomendação modernos.

4.2 RQ2: Efeitos da Otimização de *Prompts*

4.2.1 Prompts Resultantes da Otimização

Conforme descrito na Seção 3.4.1, aplicamos o processo de otimização em diferentes configurações, gerando *prompts* otimizados para cada cenário. As variações consideraram o uso das métricas *MAP* e *MAUT*, bem como diferentes conjuntos de usuários: *MAP* com 100 usuários aleatórios (*MAP_random* - Figura 6); *MAP* com 100 usuários com menos de 10 recomendações (*MAP_below_10* - Figura 7); *MAUT* com 100 usuários aleatórios (*MAUT_random* - Figura 8); e *MAUT* 100 usuários com menos de 10 recomendações (*MAUT_below_10* - Figura 9). Em todos os casos, o conteúdo do campo `User Prompt` utilizado para solicitar as recomendações ao modelo foi o mesmo apresentado na Figura 3, garantindo assim consistência na forma de interação do usuário com o modelo ao longo dos diferentes experimentos.

4.2.2 Impactos da Otimização de *Prompts* no Desempenho do LLM

Após demonstrarmos, em RQ1, que a estratégia baseada em LLM (LLaMa) supera as abordagens tradicionais em termos de balanceamento entre as métricas de precisão,

System Prompt:
Generate Personalized Movie Recommendations Using the MovieLens 1M Dataset

Use the MovieLens 1M dataset to generate a list of personalized movie recommendations based on the user's context (if provided). This list should consider patterns in user ratings, genres, and timestamps to create unique and relevant suggestions tailored to each viewer.

Your response should be a numbered list in the exact format:

1. Title (release year)
2. Title (release year)
3. Title (release year)
...

This list **must consist only of**:

* Numbered movie titles with release years in the specified format
* Exactly one title per entry
* Release years included for each title

Do not include:

* Any additional text, explanations, or comments in your response
* Extra information or unnecessary details about the movies or the users
* Non-movie titles, incomplete, or missing metadata

Provide your output exactly as specified above, following the specified format for each entry.

Figura 6 – *Prompt* gerado pelo processo de otimização com a métrica *MAP* utilizando 100 usuários aleatórios.

System Prompt:
Recommend Movies from the MovieLens 1M Dataset.

Given the MovieLens 1M dataset and a relevant user context, output a **complete** list of personally recommended movies in the following format:

1. title (release year)
2. title (release year)
3. title (release year)
...

Your response should consist **solely** of this numbered list. Please adhere strictly to the specified format.

You may draw upon patterns in user ratings, genres, preferences, and contextual clues from the dataset to generate your recommendations.

Output the recommendation list exactly as instructed; ensure that it includes the title and release year for each movie.

Figura 7 – *Prompt* gerado pelo processo de otimização com a métrica *MAP* utilizando 100 usuários com menos de 10 recomendações.

diversidade e justiça, avançamos para investigar se e como a otimização de *prompts* pode melhorar ainda mais o desempenho desses modelos.

Iniciamos nossa análise nas métricas de precisão (MAP e NDCG@10). Tomando o LLM sem otimização como referência (MAP = 0,059; NDCG@10 = 0,021), na Tabela 2, as variantes *MAP_below_10* e *MAUT_below_10* apresentaram queda estatisticamente significativa em MAP (0,056 e 0,055), enquanto *MAP_random* e *MAUT_random* foram estatisticamente equivalentes. Para o NDCG@10, houve um empate estatístico para as

System Prompt:
Personalized Movie Recommendation

Your task is to generate a list of movie recommendations based on user preferences and past ratings from the MovieLens 1M dataset. Each movie should be accompanied by its release year. Follow these rules exactly:

* Your response should consist only of a numbered list of movie titles, one movie per line, in the exact format:
1. title (release year)
2. title (release year)
3. title (release year)
...

* The list should contain multiple movie entries; there is no minimum or maximum length for the list.
* Leave no blank lines between the list entries.
* Do not include any additional text, headings, or information in your response; the list should be the only thing present.
* Each title should be a string enclosed in parentheses, indicating its release year, or left uncaptioned but present only as part of a title. For example:
1. "Inception (2010)"
2. "The Dark Knight (2008)"

* Provide movie titles that accurately reflect the preferences and habits of the users in the MovieLens 1M dataset.

Ensure your recommendations are calibrated and balance diverse film genres, as well as offer unique profiles that account for relevance, alignment with user profiles, and genre affinity.

Include only these necessary items:

1. title (release year)
2. title (release year)
3. title (release year)
...

Your response will take the form of a numbered list with the movie titles presented in the above format, offering a set of personalized and well-calibrated movie suggestions for viewers interested in a varied selection of films from the MovieLens dataset.

Figura 8 – *Prompt* gerado pelo processo de otimização com a métrica *MAUT* utilizando 100 usuários aleatórios.

System Prompt:
Recommend Movie Titles Based on User Behavior

Respond with a numbered list of recommended movie titles and release years.

The format for the response is:

1. title (release year)
2. title (release year)
3. title (release year)
...

Include the exact movie title and release year for each item.

The response must be a standalone numbered list of movie titles and dates. Provide only this list - do not include any additional text or information.

Figura 9 – *Prompt* gerado pelo processo de otimização com a métrica *MAUT* utilizando 100 usuários com menos de 10 recomendações.

diferentes otimizações, com exceção de *MAP_below_10* (0,019), que apresentou redução estatisticamente significativa. Em síntese, é possível otimizar sem degradar a precisão

Modelo	MAP@10	NDCG@10	LTC	F1 Score	RMSE	MAUT
Modelos LLM (LLaMA) com otimização						
MAP_below_10	0,056 ▼	0,019 ▼	**0,592** ▲	0,292 •	**0,350** •	0,392 ▼
MAP_random	0,057 •	**0,021** •	0,512 •	0,300 •	0,490 ▼	0,458 •
MAUT_below_10	0,055 ▼	0,020 •	0,583 ▲	0,297 •	0,406 •	0,445 •
MAUT_random	0,055 •	0,020 •	0,486 ▼	**0,302** •	0,433 ▼	0,401 •
Modelo LLM (LLaMA) sem otimização						
Sem otimização	**0,059**	**0,021**	0,517	0,296	0,367	**0,518**

Tabela 2 – Comparação entre o modelo LLM (LLaMa) sem otimização e quatro variações com diferentes estratégias de otimização. Os melhores valores em cada métrica estão em negrito. O símbolo ▲ indica que a otimização apresentou melhora estatisticamente significativa em relação ao modelo sem otimização (*p-value* < 0.05, teste de Wilcoxon); • indica ausência de diferença significativa; e ▼ indica que o modelo sem otimização foi estatisticamente superior.

quando se utiliza *MAP_random* ou *MAUT_random*.

Em relação ao viés de popularidade, utilizando a métrica RMSE (menor é melhor), o LLM sem otimização obteve 0,367. As estratégias *MAP_random* (0,490) e *MAUT_random* (0,433) apresentaram valores estatisticamente inferiores, ao passo que *MAP_below_10* (0,350) e *MAUT_below_10* (0,406) apresentaram empate estatístico com LLM sem otimização. Em outras palavras, otimizações "*random*" tendem a aumentar o erro agregado entre grupos, enquanto os processos "*below_10*" preservam esse equilíbrio.

Os processos *MAP_below_10* e *MAUT_below_10* elevaram o LTC de forma significativa (0,592 e 0,583) em relação ao *baseline* (0,517), reforçando que esses processos de otimização aumentam a diversidade das recomendações. Já as otimizações *MAP_random* (0,512) e *MAUT_random* (0,486) apresentaram empate e perdas estatísticas, respectivamente. Não houve diferenças estatisticamente significativas entre a estratégia baseada em LLM e suas variantes otimizadas em termos de justiça, o que mostra que as otimizações não afetaram a justiça do sistema em termos de gênero e popularidade.

Em relação à métrica MAUT, as estratégias *MAP_random* (0,458), *MAUT_random* (0,401) e *MAUT_below_10* (0,445) não apresentaram alterações significativas em comparação ao modelo não otimizado, enquanto *MAP_below_10* (0,392) teve redução estatisticamente significativa.

Em essência, há espaço para otimizar sem afetar o sistema como um todo e ainda há a possibilidade de ganhos em termos de diversidade e justiça. Os resultados mostram que os processos podem ser selecionados conforme o objetivo do sistema. Se o objetivo é aumentar diversidade/cobertura e reduzir miscalibração, *MAP_below_10* e *MAUT_below_10* são preferíveis, aceitando-se um *trade-off* de precisão/MAUT. Se a prioridade é preservar

precisão e qualidade global com ganhos em calibração, *MAP_random* é a melhor opção. Por fim, *MAUT_random* é uma opção quando se busca estabilidade geral, mas não é indicada quando LTC e RMSE são preferências centrais.

Em suma, respondendo à RQ2, os resultados demonstram que a otimização de *prompts* permite ajustar o sistema de recomendação a diferentes objetivos, embora não produza ganhos uniformes em todas as métricas. Algumas estratégias mantêm altos níveis de precisão, mas com prejuízos em termos de equidade, enquanto outras favorecem a diversidade de exposição, ainda que com uma leve redução no desempenho. Portanto, a seleção da estratégia de otimização deve ser guiada pelas prioridades específicas de cada contexto de aplicação, considerando os trade-offs envolvidos.

4.3 Considerações Finais

Neste capítulo, avaliamos comparativamente os métodos tradicionais e a abordagem baseada em LLM, bem como os impactos da otimização de *prompts*. Os resultados da RQ1 mostraram que o LLM superou de forma consistente os métodos tradicionais em termos de precisão e diversidade, alcançando ainda um desempenho competitivo em justiça. Apesar de a métrica F1 ter indicado limitações, o valor agregado da métrica MAUT evidenciou a robustez do modelo como solução mais equilibrada.

Na RQ2, observamos que a otimização de *prompts* não produziu ganhos uniformes, mas possibilitou ajustes conforme os objetivos de cada cenário. Estratégias como *MAP_below_10* e *MAUT_below_10* ampliaram diversidade e cobertura, ao passo que *MAP_random* preservou a precisão e a qualidade global. Assim, a seleção da estratégia de otimização deve ser orientada pelas prioridades do sistema de recomendação em uso.

De forma geral, os achados deste capítulo reforçam o potencial dos LLMs como alternativas eficazes e flexíveis para recomendação, ao mesmo tempo em que evidenciam a importância de escolhas conscientes no processo de otimização. Estes resultados servirão de base para a conclusão a ser apresentada no próximo capítulo, bem como para propor direções futuras de pesquisa.

5 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho investigamos o desempenho de sistemas de recomendação baseados em LLMs em comparação com métodos tradicionais, bem como os efeitos de diferentes estratégias de otimização de *prompts*. A avaliação foi conduzida com rigor estatístico, utilizando múltiplas métricas que contemplam precisão, diversidade, justiça, equilíbrio entre grupos e desempenho agregado.

Os resultados indicam que estratégias baseadas em LLM, mesmo sem otimizações específicas, apresentam desempenho superior à maioria dos métodos tradicionais, especialmente em termos de precisão e equilíbrio geral. Além disso, demonstra boa capacidade de diversificação das recomendações e cobertura de itens, mantendo um desempenho competitivo também no controle de discrepâncias entre grupos. Embora alguns métodos tradicionais tenham obtido valores melhores em métricas pontuais, como o RMSE, esses ganhos geralmente vieram acompanhados de perdas significativas em outras dimensões, como precisão ou diversidade. Dessa forma, estratégias baseadas em LLM se destacam por oferecer um equilíbrio mais favorável entre os diferentes objetivos do sistema. Por outro lado, as estratégias de otimização de *prompts* não proporcionaram melhorias universais, mas funcionam como mecanismos eficazes para calibrar as prioridades do sistema.

Estratégias que buscam preservar a precisão mostraram desempenho sólido nas métricas clássicas de ranqueamento, embora com aumento nas discrepâncias entre grupos. Por outro lado, abordagens voltadas à ampliação da diversidade conseguiram promover maior exposição à cauda longa, ainda que com leves perdas em precisão. A métrica de justiça apresentou variações discretas entre as estratégias, e o desempenho agregado foi inferior principalmente nas estratégias que não consideraram limites inferiores nas recomendações. Esses achados reforçam que a escolha da estratégia de otimização deve estar alinhada aos objetivos específicos da aplicação, em vez de se pautar por expectativas de ganhos uniformes.

Como direções para pesquisas futuras, destacam-se três frentes principais. Primeiramente, propõe-se a realização de experimentos em ambientes *online*, por meio de testes A/B, para avaliar o impacto das recomendações em métricas de engajamento e satisfação dos usuários em cenários reais de uso. Em segundo lugar, recomenda-se expandir a análise para outros domínios e conjuntos de dados, incluindo contextos com diferentes graus de popularidade e esparsidade, a fim de verificar a robustez dos achados. Por fim, sugere-se explorar variações arquiteturais de LLMs e novas estratégias de construção e otimização de *prompts*, assim como métricas mais refinadas de justiça e impacto social, visando aprofundar a compreensão sobre os limites e potencialidades desses modelos em sistemas de recomendação multiobjetivo.

REFERÊNCIAS

ABDOLLAHPOURI, H.; BURKE, R.; MOBASHER, B. Popularity-aware item weighting for long-tail recommendation. **arXiv preprint arXiv:1802.05382**, 2018.

ABDOLLAHPOURI, H. *et al.* The unfairness of popularity bias in recommendation. **arXiv preprint arXiv:1907.13286**, 2019.

ABDOLLAHPOURI, H. *et al.* The connection between popularity bias, calibration, and fairness in recommendation. *In*: **Fourteenth ACM conference on recommender systems**. New York, NY, USA: Association for Computing Machinery, 2020. p. 726–731.

ABDOLLAHPOURI, H. *et al.* User-centered evaluation of popularity bias in recommender systems. *In*: MASTHOFF, J. *et al.* (ed.). **Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization, UMAP 2021, Utrecht, The Netherlands, June, 21-25, 2021**. ACM, 2021. p. 119–129. ISBN 978-1-4503-8366-0. Disponível em: https://doi.org/10.1145/3450613.3456821.

AGGARWAL, C. C. **Recommender Systems: The Textbook**. 1st. ed. New York, NY, USA: Springer Publishing Company, Incorporated, 2016. ISBN 3319296574.

ATAUCHI, P. D. F. *et al.* Do calibrated recommendations affect explanations? a study on post-hoc adjustments. v. 16, p. 441–460, Jun. 2025. Disponível em: https://journals-sol.sbc.org.br/index.php/jis/article/view/5563.

BENNETT, J.; LANNING, S. *et al.* The netflix prize. *In*: **Proceedings of KDD cup and workshop**. New York, NY, USA: ACM, 2007. p. 35.

BITTENCOURT, G. *et al.* A survey on review-aware recommendation systems. *In*: **Proceedings of the 29th Brazilian Symposium on Multimedia and the Web**. [*S.l.: s.n.*], 2023. p. 198–207.

CARVALHO, R.; ROCHA, L. Estratégias para aprimorar a diversidade categórica e geográfica de sistemas de recomendação de pois. *In*: SBC. **Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)**. [*S.l.: s.n.*], 2020. p. 23–26.

CHEN, J. *et al.* Bias and debias in recommender system: A survey and future directions. **ACM Transactions on Information Systems**, ACM New York, NY, v. 41, n. 3, p. 1–39, 2023.

CHEN, Y. *et al.* Towards learning universal hyperparameter optimizers with transformers. **Advances in Neural Information Processing Systems**, v. 35, p. 32053–32068, 2022.

CUNHA, W.; ROCHA, L.; GONÇALVES, M. A. A thorough benchmark of automatic text classification: From traditional approaches to large language models. **arXiv preprint arXiv:2504.01930**, 2025.

DU, Y. *et al.* Enhancing job recommendation through llm-based generative adversarial networks. *In*: **Proceedings of the AAAI Conference on Artificial Intelligence**. [*S.l.: s.n.*], 2024. v. 38, n. 8, p. 8363–8371.

EKSTRAND, M. D. *et al.* All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation and effectiveness. **Proceedings of Machine Learning Research**, PMLR, v. 81, p. 172–186, 23–24 Feb 2018.

ELSWEILER, D.; TRATTNER, C.; HARVEY, M. Exploiting food choice biases for healthier recipe recommendation. *In*: **Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: Association for Computing Machinery, 2017. (SIGIR '17), p. 575–584. ISBN 9781450350228. Disponível em: https://doi.org/10.1145/3077136.3080826.

FONSECA, G. *et al.* Instance-selection-inspired undersampling strategies for bias reduction in small and large language models for binary text classification. *In*: **Proceedings of the 63rd ACL**. [*S.l.: s.n.*]: Association for Computational Linguistics, 2025. p. 9323–9340. ISBN 979-8-89176-251-0.

FONSECA, G. *et al.* Estratégias de undersampling para redução de viés em classificação de texto baseada em transformers. *In*: SBC. **Brazilian Symposium on Multimedia and the Web (WebMedia)**. [*S.l.: s.n.*], 2024. p. 144–152.

GAO, J. *et al.* Llm4rerank: Llm-based auto-reranking framework for recommendations. *In*: **Proceedings of the ACM on Web Conference 2025**. [*S.l.: s.n.*], 2025. (WWW '25), p. 228–239. ISBN 9798400712746. Disponível em: https://doi.org/10.1145/3696410.3714922.

GELFERT, A. Fake news: A definition. **Informal logic**, Informal Logic, v. 38, n. 1, p. 84–117, 2018.

GUO, Q. *et al.* Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. arxiv 2023. **arXiv preprint arXiv:2309.08532**.

HELBERGER, N.; KARPPINEN, K.; D'ACUNTO, L. Exposure diversity as a design principle for recommender systems. **Information, Communication & Society**, Taylor & Francis, v. 21, n. 2, p. 191–207, 2018.

HOU, Y. *et al.* Large language models are zero-shot rankers for recommender systems. *In*: SPRINGER. **European Conference on Information Retrieval**. [*S.l.: s.n.*], 2024. p. 364–381.

HU, E. J. *et al.* **LoRA: Low-Rank Adaptation of Large Language Models**. 2021. Disponível em: https://arxiv.org/abs/2106.09685.

JADON, A.; PATIL, A. A comprehensive survey of evaluation techniques for recommendation systems. *In*: SPRINGER. **International Conference on Computation of Artificial Intelligence & Machine Learning**. [*S.l.: s.n.*], 2024. p. 281–304.

KANG, W.-C. *et al.* Do llms understand user preferences? evaluating llms on user rating prediction. **arXiv preprint arXiv:2305.06474**, 2023.

KAYA, M.; BRIDGE, D. A comparison of calibrated and intent-aware recommendations. *In*: **Proceedings of the 13th ACM Conference on Recommender Systems**. New York, NY, USA: Association for Computing Machinery, 2019. (RecSys '19), p. 151–159. ISBN 9781450362436. Disponível em: https://doi.org/10.1145/3298689.3347045.

KONSTAN, J. A.; RIEDL, J. Recommender systems: from algorithms to user experience. **User modeling and user-adapted interaction**, Springer, v. 22, n. 1, p. 101–123, 2012.

LI, J. *et al.* Gpt4rec: A generative framework for personalized recommendation and user interests interpretation. **arXiv preprint arXiv:2304.03879**, 2023.

LI, L. *et al.* Large language models for generative recommendation: A survey and visionary discussions. **arXiv preprint arXiv:2309.01157**, 2023.

LICHTENBERG, J. M.; BUCHHOLZ, A.; SCHWÖBEL, P. Large language models as recommender systems: A study of popularity bias. **arXiv preprint arXiv:2406.01285**, 2024.

LIN, J. *et al.* How can recommender systems benefit from large language models: A survey. **ACM Transactions on Information Systems**, ACM New York, NY, v. 43, n. 2, p. 1–47, 2025.

LIU, J. *et al.* Is chatgpt a good recommender? a preliminary study. **arXiv preprint arXiv:2304.10149**, 2023.

LUNARDI, G. M. *et al.* A metric for filter bubble measurement in recommender algorithms considering the news domain. **Applied Soft Computing**, Elsevier, v. 97, p. 106771, 2020.

MUNSON, S. A.; RESNICK, P. Presenting diverse political opinions: How and how much. *In*: **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. New York, NY, USA: Association for Computing Machinery, 2010. (CHI '10), p. 1457–1466. ISBN 9781605589299. Disponível em: https://doi.org/10.1145/1753326.1753543.

NAGULENDRA, S.; VASSILEVA, J. Understanding and controlling the filter bubble through interactive visualization: A user study. *In*: **Proceedings of the 25th ACM Conference on Hypertext and Social Media**. New York, NY, USA: Association for Computing Machinery, 2014. (HT '14), p. 107–115. ISBN 9781450329545. Disponível em: https://doi.org/10.1145/2631775.2631811.

NEIVA, S. B.; GOMES, L. F. A. M. A aplicação da teoria da utilidade multiatributo à escolha de um software de e-procurement. **Revista Tecnologia**, v. 28, n. 2, 2007.

NEWMAN, M. Power laws, pareto distributions and zipf's law. **Contemporary Physics**, Informa UK Limited, v. 46, n. 5, p. 323–351, sep 2005. Disponível em: https://doi.org/10.1080%2F00107510500052444.

NGUYEN, T. T. *et al.* Exploring the filter bubble: The effect of using recommender systems on content diversity. *In*: **Proceedings of the 23rd International Conference on World Wide Web**. New York, NY, USA: Association for Computing Machinery, 2014. (WWW '14), p. 677–686. ISBN 9781450327442. Disponível em: https://doi.org/10.1145/2566486.2568012.

ORTEGA, G. M.; SOUZA, R. F. de; MANZATO, M. G. Evaluating zero-shot large language models recommenders on popularity bias and unfairness: A comparative approach to traditional algorithms. *In*: SBC. **Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)**. [*S.l.: s.n.*], 2024. p. 45–48.

PARISER, E. **The Filter Bubble: What the Internet Is Hiding from You**. London, UK: Penguin Group , The, 2011. ISBN 1594203008.

PARRA, D.; SAHEBI, S. Recommender systems: Sources of knowledge and evaluation metrics. *In*: **Advanced techniques in web intelligence-2**. New York, NY, USA: Springer, 2013. p. 149–175.

PITOURA, E.; STEFANIDIS, K.; KOUTRIKA, G. Fairness in rankings and recommendations: an overview. **The VLDB Journal**, Springer, p. 1–28, 2022.

RENDLE, S. *et al.* Bpr: Bayesian personalized ranking from implicit feedback. **arXiv preprint arXiv:1205.2618**, 2012.

REY, D.; NEUHÄUSER, M. Wilcoxon-signed-rank test. *In*: **International encyclopedia of statistical science**. [*S.l.: s.n.*]: Springer, 2011. p. 1658–1659.

RICCI, F.; ROKACH, L.; SHAPIRA, B. Introduction to recommender systems handbook. *In*: RICCI, F. *et al.* (ed.). **Recommender Systems Handbook**. Boston, MA: Springer US, 2011. p. 1–35. ISBN 978-0-387-85820-3. Disponível em: https://doi.org/10.1007/978-0-387-85820-3_1.

RICCI, F.; ROKACH, L.; SHAPIRA, B. Recommender systems: Introduction and challenges. *In*: _____. **Recommender Systems Handbook**. Boston, MA: Springer US, 2015. p. 1–34. ISBN 978-1-4899-7637-6. Disponível em: https://doi.org/10.1007/978-1-4899-7637-6_1.

SACILOTTI, A.; SOUZA, R. F. d.; MANZATO, M. G. Counteracting popularity-bias and improving diversity through calibrated recommendations. *In*: **In Proceedings of the 25th International Conference on Enterprise Information Systems**. Prague, Czech Republic: Scitepress, 2023. v. 1.

SAH, C. K.; XIAOLI, L.; ISLAM, M. M. Unveiling bias in fairness evaluations of large language models: A critical literature review of music and movie recommendation systems. **arXiv preprint arXiv:2401.04057**, 2024.

SILVA, D. C. da; JANNACH, D. **Calibrated Recommendations: Survey and Future Directions**. 2025. Disponível em: https://arxiv.org/abs/2507.02643.

SILVA, D. C. da; MANZATO, M. G.; DURÃO, F. A. Exploiting personalized calibration and metrics for fairness recommendation. **Expert Systems with Applications**, Elsevier, v. 181, p. 115112, 2021.

SOUZA, R.; MANZATO, M. A two-stage calibration approach for mitigating bias and fairness in recommender systems. *In*: **Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing**. New York, NY, USA: ACM, 2024. p. 1659–1661.

SOUZA, R. F. d. **Explorando Formas de Calibração e Redução do Viés de Popularidade em Sistemas de Recomendação**. 2024. Dissertação (Dissertação de Mestrado em Ciências de Computação e Matemática Computacional) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brazil, 2024. Acesso em: 2025-07-02. Disponível em: https://doi.org/10.11606/D.55.2024.tde-11072024-151014.

SOUZA, R. F. de; MANZATO, M. G. Uma abordagem em etapa de processamento para redução do viés de popularidade. *In*: SBC. **Brazilian Symposium on Multimedia and the Web (WebMedia)**. [*S.l.: s.n.*], 2024. p. 310–317.

STECK, H. Calibrated recommendations. *In*: **Proceedings of the 12th ACM Conference on Recommender Systems**. New York, NY, USA: Association for Computing Machinery, 2018. (RecSys '18), p. 154–162. ISBN 9781450359016. Disponível em: https://doi.org/10.1145/3240323.3240372.

TOUVRON, H. *et al.* Llama: Open and efficient foundation language models. **arXiv preprint arXiv:2302.13971**, 2023.

TREUILLIER, C. *et al.* Beyond trade-offs: Unveiling fairness-constrained diversity in news recommender systems. *In*: . [*S.l.: s.n.*], 2024. (UMAP '24), p. 143–148. ISBN 9798400704338.

WANG, J. *et al.* **Automating Personalization: Prompt Optimization for Recommendation Reranking**. 2025.

WANG, Y. *et al.* Enhancing recommender systems with large language model reasoning graphs. **arXiv preprint arXiv:2308.10835**, 2023.

YANG, C. *et al.* Large language models as optimizers. **arXiv preprint arXiv:2309.03409**, 2023.

YANG, C. *et al.* Large language models as optimizers. *In*: **The Twelfth International Conference on Learning Representations**. [*S.l.: s.n.*], 2023.

ZANGERLE, E.; BAUER, C. Evaluating recommender systems: survey and framework. **ACM computing surveys**, ACM New York, NY, v. 55, n. 8, p. 1–38, 2022.

ZANON, A. L.; ROCHA, L. C. D. da; MANZATO, M. G. Balancing the trade-off between accuracy and diversity in recommender systems with personalized explanations based on linked open data. **Knowl. Based Syst.**, v. 252, p. 109333, 2022.

ZHAO, Y. *et al.* Fairness and diversity in recommender systems: A survey. **ACM Trans. Intell. Syst. Technol.**, v. 16, n. 1, jan. 2025. ISSN 2157-6904.

ZHAO, Z. *et al.* Recommender systems in the era of large language models (llms). **IEEE Transactions on Knowledge and Data Engineering**, IEEE, 2024.