UNIVERSIDADE FEDERAL DE OURO PRETO

INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS

DEPARTAMENTO DE COMPUTAÇÃO

BRUNA CRISTINA GONÇALVES

IMPACTO DA INTEGRAÇÃO DE META-FEATURES EM MODELOS DE REDES NEURAIS PROFUNDAS PARA SISTEMAS DE RECOMENDAÇÃO HÍBRIDA

Ouro Preto
2026

BRUNA CRISTINA GONÇALVES

IMPACTO DA INTEGRAÇÃO DE META-FEATURES EM MODELOS DE REDES NEURAIS PROFUNDAS PARA SISTEMAS DE RECOMENDAÇÃO HÍBRIDA

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Reinaldo Silva Fortes
Coorientador: Prof. Dr. Pedro Henrique Lopes Silva

Ouro Preto
2026

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E BIOLOGICAS
DEPARTAMENTO DE COMPUTACAO

FOLHA DE APROVAÇÃO

Bruna Cristina Gonçalves

Impacto da Integração de Meta-Features em Modelos de Redes Neurais Profundas para Sistemas de Recomendação Híbrida

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Ciência da Computação

Aprovada em 2 de Março de 2026

Membros da banca

Reinaldo Silva Fortes (Orientador) - Doutor - Universidade Federal de Ouro Preto
Pedro Henrique Lopes Silva (Coorientador) - Doutor - Universidade Federal de Ouro Preto
Jadson Castro Gertrudes (Examinador) - Doutor - Universidade Federal de Ouro Preto
Guilherme Augusto Anício Drummond do Nascimento (Examinador) - Bacharel - Universidade Federal de Ouro Preto

Reinaldo Silva Fortes, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 02/03/2026

Documento assinado eletronicamente por **Reinaldo Silva Fortes**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 03/03/2026, às 09:21, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **1062479** e o código CRC **6C0F2D48**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: 3135591692 - www.ufop.br

Resumo

Sistemas de Recomendação (SR) têm sido amplamente utilizados para auxiliar usuários na descoberta de conteúdos relevantes em cenários caracterizados por grande volume e diversidade de informações. No entanto, esses sistemas ainda enfrentam desafios recorrentes, como a *esparsidade* dos dados e a dificuldade em capturar adequadamente diferentes aspectos do comportamento dos usuários. Nesse contexto, métodos híbridos surgem como uma alternativa promissora ao combinar múltiplas fontes de informação, como os *scores* gerados por algoritmos constituintes e informações adicionais extraídas dos dados, conhecidas como *meta-features*. Este trabalho investiga o impacto da integração de *meta-features* em sistemas de recomendação híbridos baseados em Deep Neural Networks (DNN), analisando como essas características influenciam o desempenho dos modelos. As *meta-features* utilizadas descrevem propriedades estruturais dos dados, permitindo enriquecer o processo de recomendação para além das avaliações explícitas. Para esse fim, foram avaliadas três estratégias de hibridização (HR, STREAM e FWLS) aplicadas a três conjuntos de dados de referência: *BookCrossing*, *Jester* e *MovieLens 1M*. Os experimentos foram conduzidos utilizando métricas de desempenho como RMSE, MAE, PC e R^2, possibilitando uma análise comparativa entre as estratégias híbridas e o impacto da inclusão das *meta-features*. Os resultados indicam que a estratégia FWLS apresentou, de modo geral, o melhor desempenho preditivo, obtendo menores valores de erro nos três *datasets* analisados. Essa abordagem explora de forma mais explícita a interação entre os *scores* dos algoritmos constituintes e as *meta-features*, contribuindo de maneira mais significativa para melhorias na qualidade das recomendações quando comparada à estratégia tradicional baseada apenas nos algoritmos constituintes. No *dataset BookCrossing*, por exemplo, o FWLS alcançou RMSE de 0.1810 e MAE de 0.1388, enquanto o HR obteve RMSE de 0.2058 e MAE de 0.1649. Nos *datasets Jester* e *MovieLens 1M*, os ganhos foram mais sutis, porém consistentes, indicando a estabilidade da abordagem em diferentes domínios. Além disso, o modelo de *Random Forest* foi utilizado como base de comparação por representar uma abordagem consolidada de *Machine Learning* em cenários de recomendação híbrida. A análise comparativa evidenciou que, embora os modelos baseados em DNN apresentem maior capacidade de modelar relações complexas entre as variáveis, o Random Forest (RF) se destaca pela eficiência computacional durante o treinamento. No entanto, é importante destacar que, em sistemas de recomendação, o processo de treinamento normalmente ocorre de forma *offline* e, uma vez que o modelo está ajustado, a geração das recomendações pode ser realizada com baixo custo computacional na fase de inferência. Dessa forma, os resultados obtidos contribuem para uma compreensão mais ampla sobre o papel das *meta-features* e das arquiteturas profundas no contexto de sistemas de recomendação híbridos.

Palavras-chave: Sistemas de Recomendação. Redes Neurais Profundas. Métodos Híbridos. *Meta-features.* Aprendizado de Máquina. Engenharia de *Features*.

Abstract

Recommender systems have been widely used to assist users in discovering relevant content in environments characterized by a large volume and diversity of information. However, these systems still face recurring challenges, such as data sparsity and the difficulty of adequately capturing different aspects of user behavior. In this context, hybrid methods emerge as a promising alternative by combining multiple sources of information, such as the scores generated by constituent algorithms and additional data-driven information known as meta-features. This work investigates the impact of integrating meta-features into hybrid recommender systems based on Deep Neural Networks, analyzing how these characteristics influence model performance. The adopted meta-features describe structural properties of the data, such as rating patterns, content similarity, and temporal information, enabling the recommendation process to be enriched beyond explicit feedback. To this end, three hybridization strategies widely discussed in the literature (HR, STREAM, and FWLS) were evaluated using three benchmark datasets: BookCrossing, Jester, and MovieLens 1M. The experiments were conducted using performance metrics such as RMSE, MAE, and the coefficient of determination (R^2), allowing a comparative analysis of the hybrid strategies and the impact of meta-feature inclusion. The results indicate that the FWLS strategy achieved, overall, the best predictive performance, obtaining lower error values across the three evaluated datasets. This approach explicitly models the interaction between the scores generated by the constituent algorithms and the meta-features, contributing more significantly to improvements in recommendation quality when compared to the traditional strategy based solely on constituent algorithms. In the *BookCrossing* dataset, for instance, FWLS achieved an RMSE of 0.1810 and an MAE of 0.1388, while HR obtained an RMSE of 0.2058 and an MAE of 0.1649. In the *Jester* and *MovieLens 1M* datasets, the improvements were more subtle but remained consistent, indicating the stability of the approach across different domains. Additionally, the Random Forest model was used as a comparison baseline, as it represents a well-established Machine Learning approach in hybrid recommendation scenarios. The comparative analysis showed that, although Deep Neural Network-based models exhibit a greater capacity to model complex relationships among variables, Random Forest stands out for its computational efficiency during the training phase. However, it is important to note that, in recommender systems, model training is typically performed offline and, once the model has been trained, recommendations can be generated with low computational cost during the inference stage. Therefore, the results contribute to a broader understanding of the role of *meta-features* and deep architectures in the context of hybrid recommender systems.

Keywords: Recommender Systems. Hybrid Methods. Deep Neural Networks. Meta-features. Machine Learning. Feature Engineering.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

R^2 Coefficient of Determination. iii, 2, 30, 34, 38, 39, 43

ASYMMF Asymmetric Factorization Model. 14

CB Content-Based Filtering. v, ix, 6, 8, 9, 14, 15

CF Collaborative Filtering. v, vi, ix, 5–9, 14, 15, 20–22, 24–26

CNN Convolutional Neural Networks. 23–25, 44

DAN Deep Association Neural Network. v, 26, 27

DNN Deep Neural Networks. iii, v, ix, 1–4, 9, 11, 17, 19, 20, 22–30, 35–38, 41, 42, 44

FEDNCF Federated Neural Collaborative Filtering. 25

FWLS Feature-Weighted Linear Stacking. iii–vi, 16, 17, 20, 22, 29, 32, 36–38, 40–43

GANs Generative Adversarial Networks. 24, 25

GMF Generalized Matrix Factorization. 26

GNN Gated Neural Networks. v, ix, 1, 17, 18, 29, 30

GRU Gated Recurrent Unit. 12

HR Hybrid Recommender. iii–vi, 2, 16, 17, 20, 22, 29, 31, 32, 36–38, 40–43

HR@k Hit Ratio. 26

IA Inteligência Artificial. 1, 10

IndED Individualized Extreme Dominance. 21, 22

KNN K-Nearest Neighbors. 14

LSTM Long Short-Term Memory Network. 12

MAE Mean Absolute Error. iii, iv, 2, 30, 33, 37, 38, 42, 43

MF Matrix Factorization. 14, 24

MLP Multilayer Perceptron. v, ix, 9–12, 19, 20, 23, 26, 29, 37

MSE Mean Squared Error. 36

NDCG@k Normalized Discounted Cumulative Gain. 26

PC Pearson Correlation. iii, 2, 30, 34, 38, 40, 42, 43

ReLU Rectified Linear Unit. 10, 11, 33, 37

RF Random Forest. iii, ix, 2–4, 13, 17, 20, 21, 28–30, 35, 37, 38, 40–43

RMSE Root Mean Square Error. iii, iv, 2, 30, 33, 36–38, 42, 43

RNAs Artificial Neural Networks. 4, 9–11

RNN Recurrent Neural Network. 12, 23, 24

SR Sistemas de Recomendação. iii, ix, 1–6, 8, 9, 11, 13, 19–29, 33, 42–44

STREAM STacking Recommendation Engines with Additional Meta-data. iii, v, vi, 16, 17, 20, 22, 29, 32, 36–38, 42

SVD Singular Value Decomposition. 14

Sumário

1 Introdução

O avanço das tecnologias de Inteligência Artificial (IA) tem impulsionado significativamente o desenvolvimento de Sistemas de Recomendação (SR), tornando-os indispensáveis em diversas áreas, como *e-commerce*, plataformas de *streaming* e ambientes educacionais. Esses sistemas têm como objetivo auxiliar os usuários na descoberta de conteúdos personalizados, contribuindo para a melhoria da experiência do usuário e para o aumento da eficiência comercial (WANG; WANG; YEUNG, 2015). Apesar dos avanços alcançados, os SR ainda enfrentam desafios recorrentes, como a *esparsidade* dos dados, o problema de *cold-start* e questões relacionadas à *justiça* nas recomendações. Nesse cenário, modelos baseados em Deep Neural Networks (DNN) têm se destacado como alternativas promissoras para lidar com essas limitações, conforme discutido por Wang et al. (2019), que propôs uma arquitetura de Gated Neural Networks (GNN) capaz de ponderar múltiplas fontes de informação no processo de recomendação.

Uma linha de pesquisa amplamente investigada para enfrentar esses desafios é a dos **métodos híbridos de recomendação**, que combinam diferentes técnicas e fontes de informação para explorar suas vantagens e reduzir limitações individuais (BURKE, 2002). De acordo com Fortes, Freitas e Gonçalves (2017), a hibridização em sistemas de recomendação pode ocorrer por meio da integração de **algoritmos constituintes**, responsáveis por gerar previsões individuais, e de ***meta-features***, que descrevem propriedades estruturais dos dados de entrada. Essa combinação permite caracterizar usuários e itens de forma mais abrangente, indo além das avaliações explícitas e possibilitando uma adaptação mais eficaz do modelo às diferentes condições dos dados, como níveis variados de *esparsidade*.

As *meta-features* desempenham um papel central nesse contexto, pois fornecem informações adicionais derivadas dos dados brutos, como estatísticas sobre avaliações, similaridade entre conteúdos e padrões temporais (FORTES; FREITAS; GONçALVES, 2017). Estudos anteriores demonstram que a incorporação dessas características em sistemas de recomendação híbridos pode melhorar significativamente a qualidade das recomendações, especialmente em cenários nos quais os dados disponíveis são limitados ou desbalanceados (ADOMAVICIUS; ZHANG, 2012). No entanto, apesar de sua eficácia em modelos tradicionais de *machine learning*, a investigação do uso de *meta-features* em arquiteturas baseadas em Deep Neural Networks (DNN) ainda é relativamente restrita.

Neste contexto, este trabalho expande a pesquisa realizada por Oliveira (2024), que analisou o impacto das *meta-features* em SR baseados em DNN. Os resultados desse estudo indicaram a necessidade de explorar modelos mais complexos, uma vez que a abordagem experimental adotada utilizou um modelo mais simples. Assim, a presente pesquisa propõe um modelo inspirado no trabalho de Wang et al. (2019), adaptado para integrar, de forma explícita, os *scores* gerados

por algoritmos constituintes e as *meta-features*, caracterizando um método híbrido alinhado às estratégias discutidas por Fortes, Freitas e Gonçalves (2017).

Além disso, para compreender melhor o impacto da adoção de modelos de DNN, este estudo realiza uma comparação de desempenho com o Random Forest (RF), um modelo tradicional de *machine learning*. Essa comparação é essencial, pois permite avaliar não apenas a eficácia do modelo de DNN ao integrar *meta-features*, mas também a viabilidade de soluções alternativas que não recorrem a essa abordagem mais complexa. Dessa forma, busca-se entender se a implementação de um modelo de DNN com *meta-features* resulta em melhorias nas recomendações. Ao explorar a sinergia entre esses elementos, espera-se avançar no campo dos SR, fornecendo soluções para os desafios relacionados à personalização e *justiça*.

No restante desta seção, serão apresentados os objetivos na Seção 1.1 e a organização do trabalho na Seção 1.2.

1.1 Objetivos

O objetivo geral deste trabalho é analisar o impacto da integração de *meta-features* em SR baseados em DNNs. Além disso, o trabalho também realiza comparações entre diferentes estratégias, para verificar se o uso das *meta-features* realmente traz melhorias nos resultados. E, para ampliar essa análise, incluímos ainda uma comparação com uma abordagem tradicional de *machine learning*, que é o Random Forest (RF).

Para alcançar o objetivo geral desta pesquisa, foram estabelecidos os seguintes objetivos específicos:

- Analisar o funcionamento de modelos de recomendação baseados em DNN, avaliando o impacto das *meta-features* e *scores* dos algoritmos constituintes nos resultados;
- Desenvolver um modelo baseado em DNN para integrar *meta-features* e *scores* dos algoritmos constituintes;
- Realizar experimentos para avaliar a eficácia do modelo proposto em comparação com o Hybrid Recommender (HR) e o RF, utilizando métricas quantitativas de desempenho, tais como Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Coefficient of Determination (R^2), Pearson Correlation (PC) e tempo médio de treinamento.

1.2 Organização do Trabalho

Esta monografia está organizada em cinco capítulos, estruturados para garantir uma compreensão lógica e coerente do conteúdo abordado. A seguir, uma breve descrição dos próximos capítulos.

O Capítulo 2 apresenta a Revisão Bibliográfica, onde são discutidos conceitos fundamentais sobre SR, DNN, RF, *meta-features*, estratégias híbridas e algoritmos constituintes. Além disso, são analisados trabalhos relacionados, contextualizando a pesquisa no estado da arte e destacando as contribuições teóricas e práticas que fundamentam este estudo.

O Capítulo 3, detalha o modelo adaptado proposto, descrevendo o conjunto de dados utilizado, a arquitetura da rede neural adotada, os métodos de treinamento e os parâmetros ajustados. Também são apresentadas as métricas empregadas para a avaliação do modelo.

O Capítulo 4 apresenta os experimentos conduzidos, os resultados e uma análise do desempenho do modelo.

Por fim, o Capítulo 5 traz as conclusões e sugestões para trabalhos futuros.

2 Revisão Bibliográfica

Este capítulo apresenta a Revisão da Literatura, que desempenha um papel fundamental no desenvolvimento deste trabalho. A revisão busca proporcionar uma base teórica consistente, reunindo informações relevantes que sustentem a pesquisa e assegurem a qualidade técnica e científica do estudo. Nesse sentido, a Seção 2.1 aborda de maneira abrangente os conceitos e temas que fundamentam este trabalho. Além disso, a Seção 2.2 apresenta os estudos que se relacionam diretamente com os objetivos deste trabalho.

2.1 Fundamentação Teórica

Esta seção tem como objetivo apresentar os conceitos-chave e os assuntos essenciais para a compreensão completa deste trabalho. Assim, a Fundamentação Teórica está organizada para apresentar conceitos relacionados a Sistemas de Recomendação (SR), Artificial Neural Networks (RNAs), Random Forest (RF), *meta-features*, Algoritmos Constituintes e Estratégias Híbridas no contexto dos SR. Além disso, será fornecida uma explicação detalhada do modelo que inspirou esta pesquisa, destacando sua estrutura e funcionamento.

2.1.1 Sistemas de Recomendação (SR)

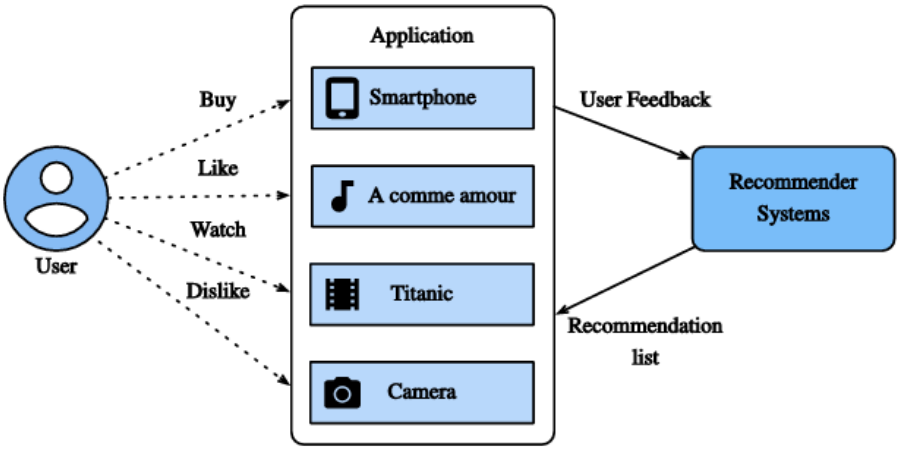
De acordo com Reategui e Cazella (2005), os Sistemas de Recomendação (SR) são ferramentas computacionais desenvolvidas para auxiliar os usuários na escolha de produtos, serviços ou informações em um cenário onde a oferta de opções é muito vasta. Esses sistemas surgiram como uma resposta ao crescimento exponencial da informação disponível, especialmente na Internet, onde os usuários frequentemente enfrentam dificuldades para selecionar as melhores alternativas devido à falta de experiência pessoal ou conhecimento prévio sobre os itens disponíveis.

Reategui e Cazella (2005) explicam que, no dia a dia, as pessoas frequentemente tomam decisões com base em recomendações de terceiros. Essas recomendações podem ser feitas de diversas maneiras, como o boca a boca *(word of mouth)*, cartas de recomendação, críticas de especialistas (como resenhas de livros ou filmes) e artigos em jornais. Os SR buscam automatizar esse processo de recomendação de forma eficiente, maximizando a correspondência entre as preferências dos usuários e as sugestões fornecidas.

O capítulo sobre SR no livro de Toledo (2022) oferece uma abordagem detalhada sobre os sistemas modernos, abordando desde os fundamentos até as aplicações mais avançadas envolvendo DNN. Esses sistemas são fundamentais para diversas plataformas online, como *Amazon*, *Netflix* e *Spotify*, ajudando os usuários a descobrir itens de seu interesse (como produtos, filmes e músicas)

com base em suas preferências. O autor destaca que a evolução da Internet, com o aumento da quantidade de dados e da diversidade de itens disponíveis, trouxe uma nova necessidade de sistemas que ajudem a selecionar e recomendar itens relevantes. Além de aprimorar a experiência do usuário, os SR são uma importante fonte de receita para as empresas ao facilitar o consumo de produtos e serviços personalizados. A Figura 2.1 ilustra o processo de recomendação, destacando o fluxo de informações, desde a entrada de dados até a geração de recomendações personalizadas. Ela também ajuda a visualizar as etapas envolvidas, como a coleta de dados do usuário, o uso de métodos de filtragem e, por fim, a apresentação das recomendações.

Figura 2.1 – Ilustração do Processo de Recomendação.



Fonte: Adaptado de (TOLEDO, 2022).

Outro conceito importante é a distinção entre *feedback* explícito e *feedback* implícito. O *feedback* explícito envolve a coleta de avaliações diretas dos usuários, como classificações ou comentários, enquanto o *feedback* implícito é baseado no comportamento do usuário, como histórico de navegação ou compras. Toledo (2022) enfatiza que, embora o *feedback* implícito seja muitas vezes mais ruidoso e difícil de interpretar, ele é uma valiosa fonte de dados para SR, especialmente em contextos em que os usuários não fornecem *feedback* explícito de forma consistente.

Em um sistema típico de recomendação, os usuários inserem suas preferências, interagem com o sistema e recebem sugestões baseadas em um processamento automático das informações disponíveis. O grande desafio desses sistemas é realizar o casamento correto entre os usuários e os itens recomendados, garantindo que as recomendações sejam relevantes e úteis Toledo (2022).

Reategui e Cazella (2005) citam o *Tapestry*, primeiro sistema de recomendação documentado, desenvolvido por Goldberg et al. (1992) e posteriormente referenciado por Resnick e Varian (1997). Esse sistema introduziu o conceito de Collaborative Filtering (CF), onde as recomendações eram geradas com base na colaboração entre usuários, que avaliavam conteúdos manualmente e compartilhavam suas opiniões.

No entanto, Reategui e Cazella (2005) argumentam que a terminologia “sistema de

recomendação" é mais abrangente do que "CF", pois há outros métodos de recomendação que não dependem da interação direta entre usuários. Isso ocorre porque um usuário pode receber recomendações de itens que nunca foram avaliados explicitamente por outros indivíduos, ou que tenham sido sugeridos com base em padrões ocultos de comportamento.

A filtragem de informações é uma técnica essencial utilizada em SR, cujo objetivo principal é fornecer aos usuários conteúdos ou produtos relevantes com base em suas preferências e interações anteriores. A aplicação dessa técnica tem se expandido para diversas áreas, como comércio eletrônico, *streaming* de vídeo e música, entre outros. A filtragem de informações pode ser dividida em três abordagens principais: Content-Based Filtering (CB), Collaborative Filtering (CF) e filtragem híbrida. Cada uma delas possui características próprias e contribui de maneira distinta para a personalização da experiência do usuário.

A seguir, serão abordados conceitos essenciais sobre os SRs. A Seção 2.1.1.1 discute a filtragem baseada em conteúdo, que utiliza informações sobre os itens para realizar recomendações. Em seguida, a Seção 2.1.1.2 examina a filtragem colaborativa, que se baseia nas interações dos usuários e permite que as recomendações sejam feitas com base nas experiências coletivas. A Seção 2.1.1.3 apresenta a filtragem híbrida, que combina diferentes técnicas para melhorar a precisão das recomendações, e, finalmente, a Seção 2.1.1.4 analisa os principais desafios enfrentados pelos sistemas de recomendação.

2.1.1.1 Content-Based Filtering (CB)

A CB é uma técnica que utiliza informações sobre os itens, como seus atributos e características, para fazer recomendações. A abordagem busca identificar itens relevantes para o usuário com base em um perfil que descreve seus interesses. Segundo Herlocker (2000), essa técnica utiliza descrições automáticas dos itens para compará-los com os interesses do usuário e determinar sua relevância.

Esta filtragem é comum em sistemas como mecanismos de busca e plataformas de recomendação, que analisam o conteúdo dos itens, como palavras-chave, e o comparam com o perfil do usuário, obtido por meio de suas interações com o sistema, como cliques ou aquisições. Técnicas como indexação de frequência de termos e sistemas de busca *booleana* são exemplos de abordagens aplicadas nessa categoria.

Contudo, a CB apresenta desafios, como a dificuldade de analisar dados não estruturados (como vídeos e áudios), o impacto do uso de sinônimos na precisão dos resultados e a possibilidade de superespecialização, na qual o sistema passa a recomendar apenas itens similares ao perfil do usuário, limitando a descoberta de novos conteúdos.

2.1.1.2 Collaborative Filtering (CF)

A CF, por sua vez, se diferencia ao não exigir o conhecimento explícito do conteúdo dos itens. Em vez disso, ela foca nas interações e avaliações dos usuários. Esta abordagem permite que um usuário se beneficie das experiências de outros que compartilham interesses semelhantes, criando um sistema de recomendações baseado na colaboração entre os membros de uma comunidade (HERLOCKER, 2000).

De acordo com Herlocker (2000), os primeiros sistemas colaborativos requeriam que os usuários indicassem suas preferências explicitamente. Com o tempo, sistemas mais avançados automatizaram esse processo, coletando pontuações de usuário. Um exemplo de ambiente baseado em CF é o sistema de recomendação de filmes MovieLens 1M (RIEDL et al., 1999). Nele o usuário insere pontuações para filmes que tenha visto e o sistema utiliza estas pontuações para encontrar pessoas com gostos similares. Dessa forma, o sistema pode recomendar filmes nos quais indivíduos com gostos semelhantes se interessariam, mas que ainda não assistiram.

A Tabela 2.1 mostra na prática como a CFs pode funcionar através de dados fictícios. Por exemplo, se quisermos recomendar um produto ao usuário "Mauro", procuraremos outros usuários com hábitos de consumo semelhantes. No caso, "Paulo" e "João" já compraram produtos que "Mauro" também comprou ("Prod2"). Em seguida, recomendamos a "Mauro" produtos que estes dois outros usuários possuem, mas que "Mauro" ainda não possui, como "Prod1" e "Prod5".

Tabela 2.1 – Recomendação baseada em CFs.

Usuário	Prod1	Prod2	Prod3	Prod4	Prod5	Prod6
Paulo		X			X	
João	X	X				
Márcia			X	X	X	
Carlos			X			
Ana	X			X		
Mauro		X				

Fonte: Adaptado de (HERLOCKER, 2000).

Apesar de suas vantagens, com a descoberta de recomendações inesperadas e a formação de comunidades de usuários, a CF enfrenta desafios como o problema do primeiro avaliador, que ocorre quando um novo item não pode ser recomendado até que mais avaliações sejam feitas, e o problema das pontuações esparsas, que surge quando o número de usuários é pequeno em comparação com a quantidade de informações no sistema (HERLOCKER, 2000).

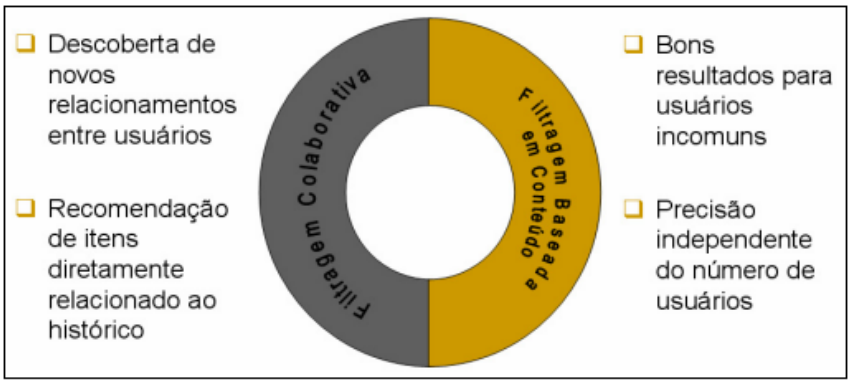
2.1.1.3 Filtragem Híbrida

A filtragem híbrida combina os pontos fortes de diferentes abordagens e algoritmos, ou mesmo diferentes fontes de dados, com o objetivo de mitigar as fraquezas de cada uma. Sistemas híbridos utilizam tanto o conteúdo dos itens quanto as avaliações dos usuários para

criar um sistema mais robusto e preciso de recomendações. Herlocker (2000) e Ansari et al. (2000) defendem que a filtragem híbrida é uma solução eficaz para melhorar a qualidade das recomendações, especialmente em casos em que as técnicas de filtragem isoladas apresentam limitações.

A abordagem híbrida pode ser implementada de diferentes maneiras, combinando os métodos de filtragem de várias formas, como, por exemplo, ponderando as recomendações de cada técnica ou aplicando uma em situações específicas, dependendo do contexto ou da qualidade das informações disponíveis. Essa abordagem é constituída de vantagens proporcionadas por diferentes técnicas e dados, unindo o melhor e mitigando fraquezas de cada uma, conforme expressado pela Figura 2.2 para combinações de CF e CB.

Figura 2.2 – Filtragem híbrida com a combinação de CF e CB.



Fonte: Adaptado de (HERLOCKER, 2000).

2.1.1.4 Desafios dos Sistemas de Recomendação

Um dos principais desafios enfrentados pelos SR é o problema do **Cold-Start**. Esse problema ocorre quando o sistema não dispõe de informações suficientes sobre novos usuários ou itens, o que dificulta a geração de recomendações relevantes. Para mitigar esse problema, algumas abordagens incluem a coleta de informações iniciais do usuário, o uso de *meta-features* e a aplicação de redes neurais recorrentes para modelar padrões de navegação (HAYKIN, 2001).

Outro desafio significativo é a **esparsidade** dos dados. SR frequentemente lidam com interações usuário-item altamente dispersas, pois a maioria dos usuários avalia apenas uma pequena fração do catálogo disponível. Modelos baseados em redes neurais conseguem mitigar esse problema ao aprender representações densas que capturam padrões latentes nos dados, permitindo melhores recomendações mesmo em cenários com poucas avaliações explícitas (HAYKIN, 2001).

Além disso, SR modernos precisam equilibrar **múltiplos objetivos**, como *precisão*, *diversidade* e *novidade*. Enquanto a precisão busca oferecer recomendações mais relevantes, a diversidade e a novidade garantem que os usuários tenham acesso a itens variados e descubram

novas opções. Modelos de DNN, como redes neurais, podem ser ajustados para otimizar múltiplos objetivos simultaneamente, tornando as recomendações mais equilibradas e personalizadas.

Por fim, a ***justiça*** é um aspecto cada vez mais relevante nos SR. Muitas abordagens tradicionais podem gerar viés, favorecendo determinados grupos de usuários ou categorias de itens, o que pode resultar em discriminação e falta de equidade. Redes neurais avançadas permitem a implementação de mecanismos que reduzem esses vieses, promovendo recomendações mais justas e inclusivas para diferentes perfis de usuários. Tais abordagens são exploradas no livro de Haykin (2001).

2.1.2 Redes Neurais em Sistemas de Recomendação

Com os avanços no DNN, as RNAs emergiram como uma solução promissora para superar as limitações dos métodos tradicionais de recomendação. Enquanto abordagens como CFs e CBs dependem de informações prévias do usuário ou dos itens, as redes neurais possibilitam o aprendizado de representações latentes mais expressivas. Isso permite capturar padrões complexos nas interações usuário-item, promovendo recomendações mais personalizadas e precisas (TOLEDO, 2022).

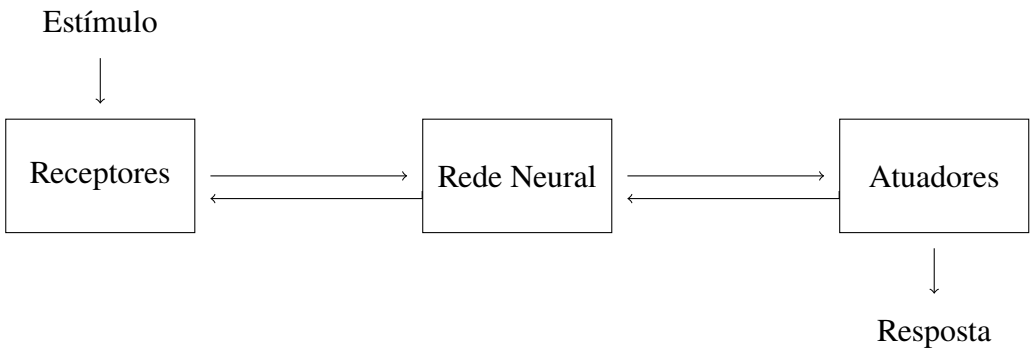
A discussão será dividida em algumas subseções, começando pela definição das redes neurais na Seção 2.1.2.1. Em seguida, na Seção 2.1.2.2, será detalhada a estrutura das Multilayer Perceptron (MLP), ressaltando sua relevância no contexto de sistemas de recomendação. Por fim, a Seção 2.1.2.3 discutirá as aplicações práticas das redes neurais em sistemas de recomendação, destacando as vantagens que esses modelos oferecem em comparação com outras abordagens.

2.1.2.1 O que são Redes Neurais?

Para entender as RNAs, é importante citar com uma breve explicação sobre o cérebro humano, que serve de principal inspiração para esses modelos computacionais (HAYKIN, 2001). O sistema nervoso humano pode ser visto como um processo distribuído de processamento de informações, conforme ilustrado no diagrama apresentado na Figura 2.3 (ARBIB, 1987). Nesse modelo, o cérebro é responsável por três estágios principais: a recepção de informações pelos receptores sensoriais; o processamento dessas informações no próprio cérebro; e a resposta gerada pelos atuadores, como músculos ou órgãos. No diagrama, as setas que apontam da esquerda para a direita indicam a transmissão do sinal, enquanto as setas da direita para a esquerda ilustram o processo de realimentação, essencial para o aprendizado e adaptação contínuos (HAYKIN, 2001).

Essa abordagem do cérebro humano inspirou a criação das RNAs, que são sistemas computacionais projetados para mimetizar o processo de aprendizado e tomada de decisão do cérebro. Segundo Haykin (2001), uma rede neural é composta por unidades de processamento simples, chamadas neurônios artificiais, que se conectam de forma massiva para armazenar e

Figura 2.3 – Representação em diagrama em blocos do sistema nervoso.



Fonte: Adaptado de (HAYKIN, 2001).

processar o conhecimento adquirido por meio da experiência. De forma similar ao cérebro, as RNAs aprendem com os dados ao ajustar suas conexões (ou "pesos sinápticos") com base nas informações recebidas.

A principal vantagem das redes neurais é sua capacidade de processar informações de maneira paralela e não linear, o que as torna extremamente eficientes em tarefas complexas, como o reconhecimento de padrões e a tomada de decisões com base em grandes volumes de dados. Assim como o cérebro humano, que se adapta constantemente para melhorar suas respostas, as RNAs também podem ser treinadas e ajustadas para melhorar sua precisão em diversas tarefas. Essas redes podem ser implementadas em circuitos eletrônicos especializados ou simuladas em computadores convencionais, sendo um modelo promissor para diversas aplicações em aprendizado de máquina e IA, conforme detalhado por Haykin (2001).

2.1.2.2 Multilayer Perceptron (MLP)

A MLP é um dos modelos mais relevantes dentro das RNAs. Segundo Haykin (2001), trata-se de uma rede neural alimentada adiante, composta por três tipos de camadas: entrada, ocultas e saída. O sinal de entrada se propaga camada por camada, permitindo que a rede aprenda padrões complexos nos dados.

As MLPs são treinadas de forma supervisionada, ou seja, utilizando um conjunto de dados rotulado no qual a saída esperada é conhecida. O treinamento ocorre por meio do algoritmo de retropropagação de erro (*backpropagation*). Esse algoritmo ajusta os pesos sinápticos da rede, propagando o erro calculado na saída para as camadas anteriores. Esse processo permite que a rede aprenda a partir de exemplos, tornando-a altamente eficaz para problemas de classificação e predição.

Uma característica essencial das MLPs é a utilização de funções de ativação não lineares, como Rectified Linear Unit (ReLU) ou sigmoide, que permitem modelar relações não triviais nos dados. A função sigmoide mapeia os valores de entrada para um intervalo entre 0 e 1, sendo frequentemente utilizada em problemas de classificação que envolvem a categorização de

dados em classes, mas também pode ser aplicada em regressão, onde o objetivo é prever valores numéricos contínuos. Já a ReLU retorna zero para valores negativos e mantém os positivos inalterados, acelerando o treinamento e mitigando o problema do desaparecimento do gradiente (*vanishing gradient*). Devido à sua flexibilidade, o modelo é amplamente utilizado em áreas como visão computacional, processamento de linguagem natural e, recentemente, em SR (HAYKIN, 2001).

A estrutura de uma MLP é formada por três camadas principais, de acordo com Haykin (2001):

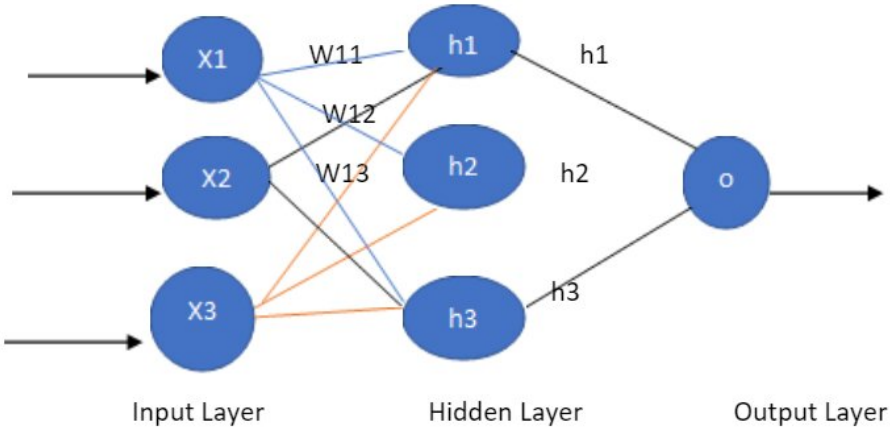
- ***Input Layer* (Camada de entrada)**: Responsável por receber os dados de entrada e repassá-los para a camada oculta. O número de neurônios nessa camada corresponde ao número de atributos do conjunto de dados.
- ***Hidden Layer* (Camada oculta)**: Realiza os cálculos essenciais para a aprendizagem. Os valores de entrada são ponderados por pesos sinápticos e passam por uma função de ativação. O número de camadas ocultas pode variar conforme a complexidade do problema, sendo necessário equilibrar o modelo para evitar *overfitting* ou *underfitting*.
- ***Output Layer* (Camada de saída)**: Responsável por gerar a previsão final do modelo. O número de neurônios nessa camada depende do tipo de problema. Em classificações binárias, por exemplo, há um único nó com ativação sigmoide; para problemas multiclasse, utiliza-se um nó para cada classe com ativação *softmax*.

O funcionamento da MLP ocorre em etapas sucessivas, conforme ilustrado na Figura 2.4. Inicialmente, os dados são recebidos pela camada de entrada, onde cada neurônio representa um atributo do conjunto de dados. Esses valores são então transmitidos para a camada oculta, onde são ponderados por pesos (W) sinápticos e processados por uma função de ativação, como ReLU ou sigmoide. Os neurônios ocultos combinam essas informações e propagam os resultados para a próxima camada. Esse processo se repete até atingir a camada de saída, que gera a previsão final do modelo. Durante o treinamento, a diferença entre a saída prevista e a esperada é calculada, permitindo que os pesos sejam ajustados por meio do algoritmo de retropropagação de erro, aprimorando o desempenho da rede ao longo do tempo (SINGH; TYAGI, 2024).

2.1.2.3 Aplicação das Redes Neurais em Recomendação

As RNAs oferecem uma alternativa promissora para lidar com os desafios dos SR ao processar grandes volumes de dados e aprender representações de características latentes de usuários e itens de forma automática. Modelos baseados em DNN têm demonstrado superioridade em diversos desafios, como a *esparsidade* dos dados e a necessidade de múltiplos objetivos na recomendação, incluindo precisão, novidade e diversidade (SINGH; TYAGI, 2024). Entre os tipos de RNAs utilizados em SR, destacam-se:

Figura 2.4 – Diagrama de uma MLP.



Fonte: Adaptado de (SINGH; TYAGI, 2024).

- **MLPs**: são redes neurais alimentadas adiante que aprendem representações latentes dos usuários e itens, sendo frequentemente utilizadas para modelagem de interações complexas e previsão de preferências.
- **Recurrent Neural Network (RNN)**: são projetadas para lidar com dados sequenciais, como texto ou séries temporais, e podem manter uma "memória" das informações anteriores, permitindo que o modelo capture dependências temporais nos dados.
- **Long Short-Term Memory Network (LSTM)**: uma variação das RNNs, as LSTMs foram desenvolvidas para superar o problema de esquecimento das dependências de longo prazo, conhecido como "desaparecimento do gradiente", permitindo que o modelo se lembre de informações por períodos mais longos.
- **Gated Recurrent Unit (GRU)**: também uma variação das RNNs, as GRUs são mais simples que as LSTMs, mas igualmente eficazes para capturar dependências temporais. Elas utilizam menos parâmetros e podem ser mais rápidas de treinar, mantendo boa performance em tarefas sequenciais.

Essas redes são especialmente úteis em contextos onde a ordem e a sequência dos dados são essenciais para gerar recomendações precisas e contextualizadas. Em aplicações de recomendação baseadas no histórico de navegação, as RNNs são particularmente úteis, pois conseguem modelar padrões temporais nas interações do usuário com um sistema. Um exemplo prático é o uso de redes LSTM e GRU para prever itens que um usuário pode querer visualizar ou comprar com base em seu comportamento anterior. A principal vantagem dessas redes é sua capacidade de manter uma memória interna que armazena informações relevantes sobre estados passados, permitindo que a recomendação seja feita sem depender exclusivamente de avaliações explícitas dos usuários (HAYKIN, 2001).

2.1.3 Random Forest (RF)

O Random Forest (RF) é um algoritmo de aprendizado de máquina que utiliza uma abordagem de *ensemble*, combinando múltiplas árvores de decisão para melhorar a precisão das previsões. De acordo com Breiman (2001), o algoritmo é construído com base no princípio de que uma coleção de modelos fracos pode ser combinada para formar um modelo forte. Cada árvore na floresta é treinada em um subconjunto aleatório do conjunto de dados original, e as previsões são feitas por meio da agregação das saídas de todas as árvores, geralmente por meio de votação (no caso de classificação) ou de média (no caso de regressão).

Uma das principais vantagens do RF é sua robustez contra o *overfitting*, especialmente em conjuntos de dados com muitas características. Isso ocorre porque, ao treinar cada árvore em um subconjunto diferente dos dados, o modelo se torna menos sensível a flutuações e ruídos presentes nos dados de treinamento. Além disso, o algoritmo pode lidar com variáveis de entrada de diferentes escalas e com relações não lineares, tornando-o versátil para uma ampla gama de aplicações (BREIMAN, 2001).

Na prática, o RF é frequentemente utilizado em SR, como demonstrado por Peya et al. (2025), onde a eficácia do algoritmo foi evidenciada em um SR de clientes em potencial, alcançando 100% de acurácia. O uso do RF permitiu explorar as relações entre características de usuários e itens de maneira eficaz, resultando em recomendações mais precisas. Além disso, em Cedro (2025), o RF foi aplicado para desenvolver um SR de investimentos, demonstrando sua capacidade em classificar perfis de risco com uma acurácia de 88,2%. Esse desempenho ressalta a robustez do algoritmo na captura de padrões complexos. Por outro lado, Pinto e Saviniec (2024) e Athayde (2023) também utilizaram o RF para recomendar cultivares de café e para otimizar plantios, respectivamente, demonstrando sua versatilidade em diversos contextos de recomendação. Assim, o RF se mostra não apenas uma alternativa viável, mas também uma escolha competitiva em SR, especialmente nas áreas de comércio eletrônico e agricultura de precisão.

2.1.4 Métodos Híbridos

Um aspecto fundamental neste trabalho é a definição de *meta-features*, que são características extraídas dos dados de entrada para auxiliar na modelagem do sistema de recomendação. Essas *meta-features* desempenham um papel crucial na adaptação dos modelos de recomendação híbridos, pois permitem capturar informações estruturais sobre os dados de usuários e itens, indo além das avaliações explícitas. As informações apresentadas em Seção 2.1.4.1 e Seção 2.1.4.2 baseiam-se no trabalho de Fortes, Freitas e Gonçalves (2017), que analisou a importância das *meta-features* e dos algoritmos constituintes em estratégias de recomendação híbrida.

2.1.4.1 Algoritmos Constituintes

Os algoritmos constituintes são os modelos individuais que compõem um sistema de recomendação híbrido. Cada um desses algoritmos tem a função de gerar previsões de recomendação com base em diferentes abordagens e fontes de dados. A hibridização desses algoritmos visa combinar suas vantagens e mitigar suas limitações, criando um sistema mais robusto e preciso (FORTES; FREITAS; GONçALVES, 2017).

Os algoritmos de CF explorados no trabalho de Fortes, Freitas e Gonçalves (2017) e neste estudo são fornecidos pelo *MyMediaLite* (GANTNER et al., 2011), incluindo técnicas de ponta, como K-Nearest Neighbors (KNN), Matrix Factorization (MF), Singular Value Decomposition (SVD) e Asymmetric Factorization Model (ASYMMF). O KNN é um método que prevê as preferências de um usuário com base nas preferências de usuários semelhantes (RESNICK et al., 1994). A MF decompõe a matriz de avaliações em fatores latentes que capturam as interações entre usuários e itens (KOREN; BELL; VOLINSKY, 2009). A SVD é uma técnica de álgebra linear que decompõe uma matriz em três outras matrizes, facilitando a identificação de padrões e a redução de dimensionalidade (DEERWESTER et al., 1990). O ASYMMF é um modelo que considera fatores assimétricos nas interações entre usuários e itens, permitindo capturar nuances nas preferências dos usuários (KOREN, 2008). Já o algoritmo CB foi implementado utilizando a *Apache Lucene* (MCCANDLESS et al., 2010), indexando o conteúdo dos itens, construindo perfis de usuários a partir de seus itens preferidos e retornando uma lista de itens ranqueados com base na similaridade entre o perfil do usuário e o conteúdo dos itens.

2.1.4.2 *Meta-features*

Meta-features são atributos derivados dos dados brutos que fornecem informações adicionais sobre o conjunto de dados e podem ser utilizados para melhorar a qualidade das recomendações. Elas podem ser agrupadas em diferentes categorias, dependendo da abordagem utilizada (FORTES; FREITAS; GONçALVES, 2017):

- **Collaborative Filtering (CF):** Nos sistemas baseados em CF, os dados de entrada consistem em uma matriz de avaliações, onde os usuários expressam sua satisfação em relação a diferentes itens. Além das notas atribuídas, outras informações, como a data e a frequência das avaliações, podem ser usadas para calcular *meta-features* que caracterizam a distribuição dos dados, a densidade das avaliações e a similaridade entre usuários ou itens.
- **Content-Based Filtering (CB):** Em abordagens CB, os dados de entrada incluem atributos estruturados e não estruturados dos itens recomendados. Nesse contexto, as *meta-features* são calculadas com base no volume de conteúdo disponível e nas medidas de similaridade entre os itens recomendados.

A formalização do cálculo das *meta-features* segue a Equação 2.1:

$$\text{MF}(\Sigma, \delta, \sigma) = \left\{ \sum_{e} (\delta(\sigma_e(D))), \forall e \in D \right\} \tag{2.1}$$

onde:

- D representa os dados de entrada;
- σ_e é uma função de seleção aplicada sobre D para o elemento e, que pode ser um usuário ou um item;
- δ é uma medida calculada para o subconjunto selecionado dos dados de entrada;
- Σ é uma função agregadora, que pode ser contagem, média, logaritmo, entre outras, aplicada para cada elemento e.

Segundo Fortes, Freitas e Gonçalves (2017), *meta-features* são capazes de caracterizar os dados de entrada de forma abrangente, considerando os aspectos mencionados anteriormente para abordagens CFs e CB, mas mantendo um número mínimo de *meta-features*. A Tabela 2.2 lista as *meta-features*, seus tipos e descrições breves.

Tabela 2.2 – *Meta-features*. Tipos: (I) Conteúdo; (II) Número de avaliações; (III) Valores das avaliações; (IV) Datas das avaliações.

Acrônimo	**Tipo**	**Pequena descrição**
COUNTSIZE	I	Número de tokens indexados a partir do conteúdo.
COSINE	I	Média da Similaridade do Cosseno calculada para pares de conteúdo de usuários ou itens.
DICE	I	Média do índice de Dice calculada para pares de conteúdo de usuários ou itens.
ENTROPY	I	Média da Entropia calculada para o conteúdo individual de usuários ou itens.
JACCARD	I	Média do índice de Jaccard calculada para pares de conteúdo de usuários ou itens.
LOGQTDR	II	Logaritmo do número de avaliações.
PCR	II	Porcentagem de usuários ou itens em comum.
PR	II	Porcentagem do número de avaliações.
GINI	III	Índice de Gini calculado para os valores das avaliações.
PEARSON	III	Coeficiente de Variação de Pearson calculado para os valores das avaliações.
PQMEAN	III	Índice pq-mean calculado para os valores das avaliações.
SD	III	Desvio padrão calculado para os valores das avaliações.
LOGDATER	IV	Logaritmo do número de datas de avaliação distintas.
PRDATER	IV	Porcentagem do número de datas de avaliação.

Fonte: Adaptado de (SINGH; TYAGI, 2024).

As *meta-features* utilizadas neste estudo, assim como Fortes, Freitas e Gonçalves (2017), incluem diferentes medidas para quantificação do tamanho do conteúdo (COUNTSIZE), cálculo

de similaridades entre elementos (COSINE, DICE, JACCARD), coesão do conteúdo (ENTROPY), quantidade de avaliações disponíveis (LOGQTDR, PR), proporção de avaliações comuns (PCR, PR), distribuição de valores das avaliações (GINI, PEARSON, PQMEAN, SD) e informações temporais das avaliações (LOGDATER, PRDATER).

2.1.4.3 Estratégias Híbridas

As estratégias híbridas combinam diferentes técnicas de recomendação para melhorar a precisão, a diversidade e a personalização das recomendações. Segundo Fortes, Freitas e Gonçalves (2017), existem muitas estratégias para hibridização (BURKE, 2002), mas os poucos trabalhos que fazem uso de *meta-features* estão focados apenas na estratégia Ponderada.

A hibridização ponderada consiste em combinar diferentes fontes de informação por meio da atribuição de pesos que otimizam o desempenho do sistema. Essa abordagem permite explorar melhor os diferentes aspectos dos dados e aprimorar as recomendações. Três estratégias de hibridização ponderada foram exploradas por Fortes, Freitas e Gonçalves (2017), as descrições e equações que descrevem como as características são construídas para cada estratégia são mostradas nas Equações 2.2, 2.3 e 2.4. Seja S o conjunto de pontuações de recomendação individuais e M o conjunto de *meta-features* extraídas dos dados de entrada:

- **HR**: não utiliza *meta-features*, sendo as *features* definidas apenas pelos *scores* dos algoritmos constituintes.

$$F_{HR} = \{s|\forall s \in S\} \tag{2.2}$$

- **STacking Recommendation Engines with Additional Meta-data (STREAM)**: definida por Adomavicius e Zhang (2012), constrói as *features* pela união de *scores* e *meta-features*.

$$F_{STREAM} = \{m|\forall m \in M\} \cup \{s|\forall s \in S\} \tag{2.3}$$

- **Feature-Weighted Linear Stacking (FWLS)**: proposta por Sill et al. (2009), constrói features a partir do produto entre *scores* e *meta-features*, ideia é capturar a relação não linear entre *scores* e *meta-features* no processo de construção de *features*, permitindo o uso de métodos de aprendizado linear para obter bons resultados consumindo menos tempo e recursos, uma vez que métodos não lineares são caros.

$$F_{FWLS} = \{m \cdot s|\forall m \in M \wedge \forall s \in S\} \tag{2.4}$$

Para otimizar a ponderação dos modelos híbridos, foram utilizados diferentes métodos de aprendizado de máquina disponíveis no *Scikit-Learn* (LEARN, 2020). Esses métodos foram categorizados em:

- **Modelos Lineares**: Incluem técnicas como Regressão *Ridge*, Regressão *Bayesiana Ridge* e Regressão Linear por Vetores de Suporte. Esses modelos são mais apropriados para

FWLS, pois permitem capturar padrões relevantes sem a complexidade computacional dos métodos não lineares.

- **Modelos Não Lineares**: Métodos como RF e *Gradient Boosted Regression Trees*, que são capazes de modelar relações mais complexas entre as variáveis. Esses modelos são mais adequados para a estratégia STREAM, pois permitem a captura de padrões significativos entre os algoritmos constituintes e as *meta-features* de maneira indireta, uma vez que as interações entre eles não são consideradas na construção das *features*.

A principal diferença entre as estratégias HR, STREAM e FWLS está na forma como as *features* são construídas e combinadas. A comparação entre essas estratégias visa avaliar o impacto das *meta-features* na recomendação híbrida. Conforme indicado por Fortes, Freitas e Gonçalves (2017), a escolha por utilizar tanto STREAM quanto FWLS foi motivada pelo fato de que essas abordagens nunca foram comparadas diretamente antes. Dependendo dos conjuntos de dados e das combinações de *meta-features* utilizadas, elas podem apresentar comportamentos distintos.

2.1.5 Modelo de Gated Neural Networks (GNN)

O modelo adotado neste trabalho foi inspirado na abordagem proposta por Wang et al. (2019), que integra técnicas de DNN para recomendações. O estudo introduz um sistema de *robo-advisor* baseado em DNN, combinando um mecanismo de *RankNet* multiobjetivo com uma estrutura de (GNN). O objetivo principal é otimizar a seleção e ranqueamento, fornecendo recomendações personalizadas.

A abordagem proposta por Wang et al. (2019) utiliza três núcleos *RankNet*, cada um treinado para otimizar diferentes objetivos. A rede neural com portas atua como um mecanismo de fusão dessas informações, aprendendo automaticamente a ponderação ideal entre os diferentes núcleos de *RankNet*. Essa estrutura permite que o modelo identifique as informações mais relevantes para a tomada de decisão, proporcionando um sistema de recomendações mais preciso e robusto.

Partes principais do modelo:

- ***RankNet* Profundo**
  - Utiliza o *RankNet*, que é um modelo de aprendizado para ranqueamento. Ele compara pares de ações e tenta aprender qual delas deve ser ranqueada mais alto.
  - A função de perda usada é a entropia cruzada (*cross-entropy loss*), ajustando os pesos do modelo para minimizar a diferença entre as probabilidades previstas e as reais.

- **GNN**

– Essa rede neural aprende pesos automaticamente para combinar diferentes fontes de informação.

– A Equação 2.5 define a forma como os pesos das redes são calculados.

$$\omega_i(Market, Co., Fin.) = \frac{e^{\sum_{j=1}^{J'} \sigma_1(x_1\omega_{1,j}+x_2\omega_{2,j}+x_3\omega_{3,j}+b_i)\omega_{j,i}+b_i}}{\sum_{i=1}^{I'} e^{\sum_{j=1}^{J'} \sigma_1(x_1\omega_{1,j}+x_2\omega_{2,j}+x_3\omega_{3,j}+b_i)\omega_{j,i}+b_i}} \tag{2.5}$$

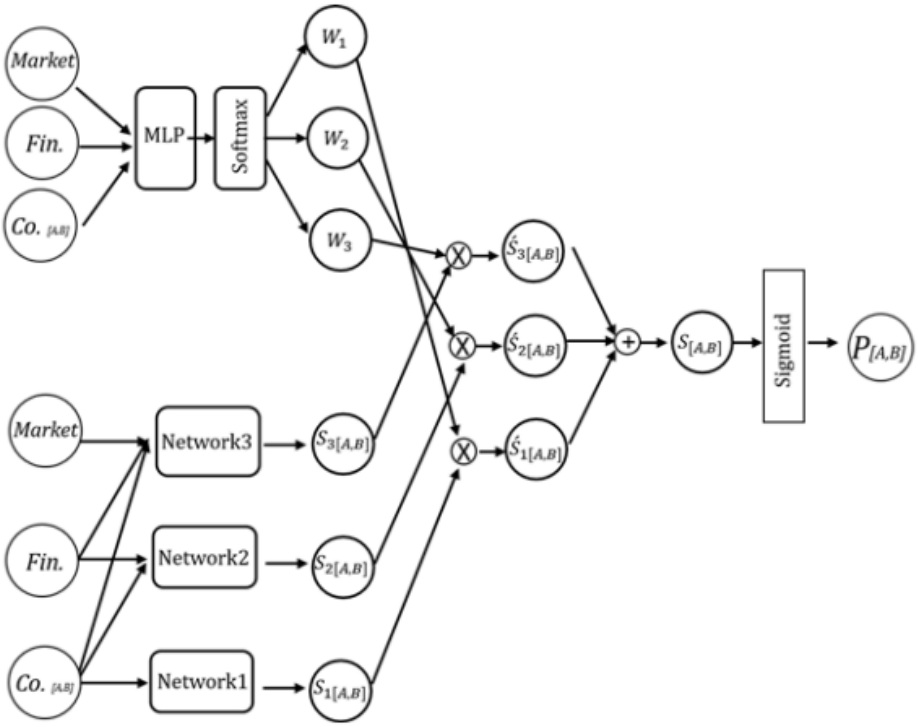
Onde:

* ω_i representa o peso do i-ésimo modelo;
* σ_1 é uma função sigmoide, que normaliza os valores;
* x_1, x_2, x_3 correspondem às características *Market*, *Co.* e *Fin.*;
* $\omega_{1,j}, \omega_{2,j}, \omega_{3,j}$ são os pesos associados a essas características;
* b_i é um viés.

A saída dessa equação representa um *softmax* ponderado dos modelos superiores, que permite que o modelo aprenda automaticamente quais fontes de informação são mais importantes.

A Figura 2.5 ilustra o funcionamento da arquitetura e como os dados fluem pelo modelo:

Figura 2.5 – Gated Neural Networks (GNN) usada para recomendar as principais ações.



Fonte: Adaptado de (WANG et al., 2019).

1. **Extração de características:**

- O modelo recebe os dados de mercado, transações e finanças e passa essas informações por diferentes redes neurais (*Network1, Network2, Network3*).
- Cada rede aprende uma representação diferente dos dados.

2. **Aprendizado de pesos:**
   - Uma MLP processa as informações e gera pesos $\omega_{1,j}, \omega_{2,j}, \omega_3$.
   - Esses pesos são normalizados usando Softmax, garantindo que sua soma seja 1.
3. **Cálculo do Score Final:**
   - Cada saída da rede (S1(A,B), S2(A,B), S3(A,B)) é ponderada pelos pesos $\omega_{1,j}, \omega_{2,j}, \omega_3$.
   - Os valores ponderados são somados e passam por uma função sigmoide para gerar a probabilidade final P(A,B).

2.2 Trabalhos Relacionados

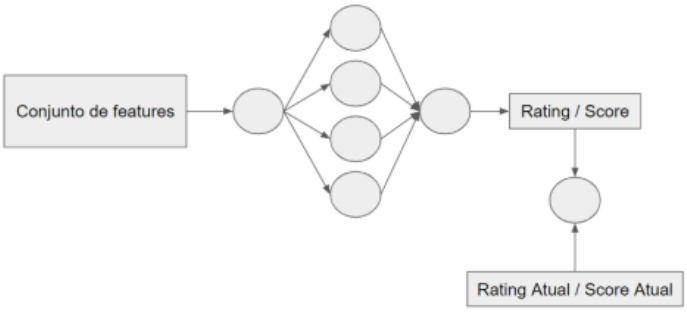
Nesta seção são discutidos estudos relevantes que contribuem para o avanço dos SR, abrangendo tanto abordagens tradicionais quanto técnicas mais recentes baseadas em DNN. A análise desses trabalhos permite compreender os principais desafios da área e as soluções propostas ao longo do tempo. Entre os estudos revisados, destacam-se pesquisas que exploram diferentes metodologias para aprimorar a qualidade das recomendações, incluindo a utilização de *meta-features* e algoritmos constituintes em sistemas híbridos. Trabalhos como os de Fortes, Freitas e Gonçalves (2017), Fortes et al. (2021) e Oliveira (2024) demonstram a importância da incorporação de *meta-features* no processo de recomendação.

Diversos estudos recentes abordam a utilização de *meta-features* em SR baseados em DNN, focando principalmente em desafios como *esparsidade* de dados e *justiça*. O trabalho de Oliveira (2024) apresenta uma contribuição significativa ao explorar como características estatísticas dos dados de entrada podem ser integradas explicitamente em arquiteturas de DNN para melhorar o desempenho dos SR. Os experimentos realizados, utilizando conjuntos de dados clássicos como MovieLens 1M, *BookCrossing* e *Jester*, indicaram melhorias nas métricas de avaliação multi-objetivo, destacando-se o índice de Gini como uma medida importante de *justiça*. O estudo também sublinhou a importância do uso de DNN, como o modelo *DeepFM*, que demonstraram ser eficazes na captura de interações complexas entre *features*, resultando em recomendações mais precisas e diversificadas.

Os resultados observados, no entanto, indicaram que as diferentes configurações de modelos e os conjuntos de *features*, com ou sem o uso de *meta-features*, não apresentaram grandes variações nos desempenhos nas métricas de ranqueamento. Essas melhorias foram mais notáveis em outros aspectos, como a precisão e a diversidade das recomendações.

A Figura 2.6 ilustra o modelo baseado em MLP utilizado nos experimentos realizados por Oliveira (2024). Esse modelo é fundamental para processar as *meta-features* combinadas com os dados de usuários e itens recomendados, capturando interações não-lineares e promovendo maior precisão nas recomendações.

Figura 2.6 – Modelo baseado em MLP utilizado nos experimentos.



Fonte: Adaptado de (OLIVEIRA, 2024).

Além disso, Oliveira (2024) e Fortes, Freitas e Gonçalves (2017) utilizam três abordagens específicas para a construção de conjuntos de *features*: a HR, que utiliza apenas algoritmos constituintes, a STREAM, que combina algoritmos com *meta-features* de forma simplificada, e a FWLS, que adota uma combinação multiplicativa dessas abordagens. Essas estratégias proporcionaram uma análise detalhada sobre como as *meta-features* podem ser aproveitadas para melhorar a precisão e a *justiça* nas recomendações.

Comparativamente, estudos anteriores, como os de Zhang et al. (2021) e Jannach et al. (2017), Fortes et al. (2021), Fortes, Freitas e Gonçalves (2017), também investigaram o uso de *meta-features* em sistemas híbridos tradicionais de recomendação. No entanto, Oliveira (2024) avança o estado da arte ao aplicar *meta-features* diretamente em arquiteturas de DNN, abordando questões de *esparsidade* e *justiça*.

Este estudo também revelou limitações, como a dependência da qualidade das *meta-features* selecionadas e a complexidade computacional envolvida na implementação de DNN. Mesmo assim, os resultados reforçam a eficácia da engenharia de *features* e do meta-aprendizado como estratégias para melhorar os SR em contextos variados.

Por fim, ao comparar o trabalho de Oliveira (2024) com outras pesquisas, é possível observar que as contribuições do estudo se destacam pela inovação na aplicação de *meta-features* em DNN, enquanto os trabalhos anteriores focavam mais em algoritmos híbridos tradicionais. A proposta de explorar modelos mais complexos e robustos e integrar as *meta-features* diretamente nas arquiteturas de DNN é uma contribuição importante para o avanço da área.

O uso de RF em SR também é uma abordagem relevante. O trabalho de Xing, Ma e Ma (2015) propõe um método de recomendação de serviços que combina CF com RF. A pesquisa demonstra que essa combinação pode melhorar a precisão das recomendações, especialmente

em cenários onde os dados são escassos ou onde há alta variabilidade nas preferências dos usuários. Os experimentos realizados com dados de serviços mostram resultados promissores, destacando como o algoritmo pode capturar interações complexas entre usuários e serviços, o que proporciona recomendações mais relevantes.

Outro estudo relevante é o de Haque (2024), que investiga um sistema de recomendação de produtos de *e-commerce* utilizando algoritmos de aprendizado de máquina, incluindo RF. Os autores destacam que o RF, por sua robustez e capacidade de lidar com grandes volumes de dados, é eficaz na personalização de recomendações. A pesquisa enfatiza a importância de interpretar as características que influenciam as recomendações, o que constitui uma vantagem significativa do RF em comparação com outros métodos de aprendizado mais complexos.

O estudo de Rahman, Haque e Ahammad (2024) apresenta um sistema de recomendação de produtos para *e-commerce* que utiliza diversos algoritmos de aprendizado de máquina. Os autores enfatizam a importância de adaptar as estratégias de recomendação às necessidades específicas dos usuários em ambientes de comércio eletrônico, onde a personalização é crucial para melhorar a experiência do cliente. O modelo proposto combina técnicas de CF e algoritmos baseados em conteúdo, avaliando seu desempenho por meio de métricas como precisão, revocação e *F1-score*.

Os resultados obtidos indicam que o sistema de recomendação baseado em *machine learning* não apenas melhora a precisão das recomendações, mas também proporciona uma experiência de compra mais personalizada. Os autores conduziram experimentos com conjuntos de dados reais de *e-commerce*, demonstrando que a integração de múltiplos algoritmos, incluindo RF, aprimora a eficácia do sistema em comparação com abordagens tradicionais. Essa pesquisa reforça a ideia de que a combinação de diferentes técnicas pode oferecer soluções mais robustas e adaptáveis, atendendo às demandas variadas dos consumidores em um mercado dinâmico.

Além disso, o uso de RF neste contexto destaca a capacidade do algoritmo de lidar com dados complexos e de alta dimensionalidade, oferecendo uma interpretação mais clara dos fatores que influenciam as recomendações. A análise da importância das variáveis permite que os desenvolvedores entendam melhor como diferentes características influenciam as preferências dos usuários, contribuindo para a melhoria contínua do sistema de recomendação.

Diversos estudos recentes abordam a otimização multiobjetivo em SR, enfatizando a necessidade de balancear aspectos como precisão, diversidade e novidade das recomendações. O trabalho de Fortes et al. (2021) propõe o Individualized Extreme Dominance (IndED), um método baseado em preferências individuais para otimização multiobjetivo em SR. A abordagem explora uma nova relação de dominância baseada em Pareto, combinada com testes estatísticos para melhorar a qualidade das soluções geradas.

Os experimentos conduzidos pelos autores demonstram que o IndED supera métodos tradicionais ao produzir recomendações mais alinhadas com as preferências individuais dos usuá-

rios. Utilizando três conjuntos de dados representativos (MovieLens 1M, *Jester* e *BookCrossing*), os resultados evidenciaram que o IndED melhora a satisfação dos usuários, equilibrando os *trade-offs* entre objetivos conflitantes. Uma contribuição importante do estudo é a incorporação de *meta-features* no processo de recomendação, permitindo uma otimização mais refinada dos critérios de qualidade.

A abordagem do IndED se diferencia por explorar um modelo evolutivo multiobjetivo, onde soluções são geradas e avaliadas iterativamente para encontrar um conjunto Pareto-ótimo. O modelo se baseia na relação de dominância extrema individualizada, que combina pesos atribuídos a cada objetivo conforme a preferência dos usuários. Essa abordagem melhora a personalização das recomendações, resultando em listas mais relevantes e diversificadas.

Por outro lado, o trabalho de Fortes, Freitas e Gonçalves (2017) analisa a importância das *meta-features* na hibridização de SR, explorando diferentes combinações de CFs e baseada em conteúdo. Fortes, Freitas e Gonçalves (2017) conduzem uma análise multicritério para avaliar o impacto dessas *meta-features* na qualidade das recomendações, considerando critérios como diversidade, novidade e precisão.

Os resultados indicam que a incorporação de *meta-features* melhora significativamente a qualidade das recomendações, reduzindo a tendência dos algoritmos tradicionais de favorecer apenas itens populares. O estudo destaca que a utilização de *meta-features* pode ajudar a equilibrar diferentes objetivos, promovendo um sistema de recomendação mais justo e diversificado.

O modelo proposto em Fortes, Freitas e Gonçalves (2017) considera três abordagens distintas de hibridização: a HR, que utiliza apenas algoritmos constituintes, a STREAM, que combina algoritmos com *meta-features* de forma simplificada, e a FWLS, que adota uma combinação multiplicativa dessas abordagens. Os experimentos demonstraram que a incorporação de *meta-features* melhora significativamente a diversidade e novidade das recomendações, sem prejudicar a precisão.

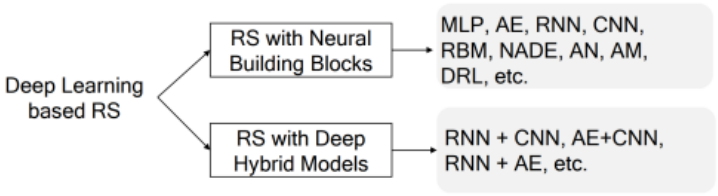
O trabalho de Zhang et al. (2021) fornece uma revisão abrangente sobre SR baseados em DNN, abordando seus avanços, desafios e novas perspectivas. Os autores discutem como as técnicas de DNN têm revolucionado os SR, especialmente no contexto de dados esparsos, aprendizado sequencial e recomendação em tempo real, apresentando uma análise detalhada dos principais modelos e suas aplicações práticas.

Comparativamente, enquanto Fortes et al. (2021) foca em uma abordagem baseada em Pareto para otimização multiobjetivo, Fortes, Freitas e Gonçalves (2017) investiga o impacto das *meta-features* na hibridização de SR. Ambos os estudos reforçam a importância da engenharia de *features* na melhoria da qualidade das recomendações, indicando caminhos promissores para pesquisas futuras na área. A integração dessas técnicas pode representar um avanço significativo na criação de SR mais justos, diversificados e eficientes.

De acordo com Zhang et al. (2021), os modelos baseados em DNN se destacam pela

capacidade de capturar interações complexas entre usuários e itens recomendados. Técnicas como Convolutional Neural Networks (CNN), RNNs e modelos baseados em atenção são frequentemente empregadas para explorar tanto informações estruturadas quanto não estruturadas, como texto, imagens e histórico de interações. O trabalho enfatiza que, além de melhorar a precisão das recomendações, os métodos baseados em DNN têm o potencial de aumentar a diversidade e a personalização das recomendações, aspectos essenciais para promover uma melhor experiência do usuário.

Figura 2.7 – Modelo baseado em MLP utilizado nos experimentos.



Fonte: Adaptado de (ZHANG et al., 2021).

Os resultados apresentados por Zhang et al. (2021) indicam que, embora os sistemas baseados em DNN superem algoritmos tradicionais em muitas métricas de desempenho, como precisão e taxa de cliques, ainda existem desafios significativos a serem superados. Entre esses desafios, destacam-se a alta complexidade computacional, a dificuldade de interpretar os modelos gerados e a necessidade de grandes volumes de dados rotulados para treinamento. Além disso, os autores ressaltam a importância de considerar *justiça* e transparência nos SR baseados em DNN, apontando que esses aspectos ainda são subexplorados na literatura.

Outro ponto importante abordado no estudo é a integração de *meta-features* nos SR. Zhang et al. (2021) destacam que a incorporação de informações adicionais, como características demográficas ou contexto de uso, pode ajudar a mitigar problemas como a *esparsidade* de dados e melhorar a eficácia das recomendações. Modelos híbridos que combinam *meta-features* com DNN são apresentados como uma direção promissora, especialmente em cenários onde a personalização é essencial.

Comparativamente, enquanto estudos como os de Oliveira (2024) se concentram na aplicação direta de *meta-features* em arquiteturas específicas de DNN, Zhang et al. (2021) fornecem uma visão mais ampla, abordando diversas técnicas e *frameworks* que têm sido explorados na literatura. A análise do trabalho reforça a importância de estratégias híbridas, alinhando-se às abordagens propostas por Oliveira (2024), mas vai além ao discutir a aplicabilidade de arquiteturas emergentes, como *Transformers*, para capturar interações sequenciais complexas.

Por fim, o trabalho de Zhang et al. (2021) contribui significativamente para o campo ao mapear o estado da arte em SR baseados em DNN e propor direções futuras para pesquisa, como a integração de *justiça*, transparência e eficiência computacional. Essa análise fornece uma base sólida para a exploração de soluções inovadoras no campo, destacando lacunas que

podem ser abordadas por pesquisas futuras, incluindo este trabalho, que busca aprofundar o uso de meta-aprendizado e engenharia de *features* em SR.

Toledo (2022) destaca os avanços proporcionados pelo DNN no campo dos SR. Com o uso de DNN, é possível melhorar significativamente a precisão das recomendações ao lidar com dados mais complexos e volumosos. O autor discute como CNNs e RNNs têm sido aplicadas para entender as relações sequenciais e estruturais dos dados, como no caso de recomendações baseadas no comportamento anterior do usuário. Além disso, a utilização de mecanismos de atenção tem sido explorada como uma técnica para otimizar o desempenho dos modelos em tarefas que exigem foco em partes específicas dos dados.

A comparação entre os SR abordados por Zhang Zachary C. Lipton e Smola (2021) e a proposta de pesquisa deste trabalho, revela tanto semelhanças quanto diferenciações. Ambas as abordagens compartilham o objetivo de melhorar a personalização das recomendações, utilizando redes neurais para aprender padrões e preferências dos usuários. No entanto, a pesquisa proposta foca na manipulação e otimização das *features* de entrada, com o uso de meta-aprendizado, para aprimorar a eficácia dos SR. A proposta de explorar *meta-features*, como medidas estatísticas sobre os dados, permite uma adaptação mais dinâmica e eficiente dos modelos de recomendação, oferecendo uma contribuição distinta ao campo, ao aplicar essas técnicas de forma prática para melhorar a qualidade das recomendações.

Portanto, enquanto os SR abordados por Zhang Zachary C. Lipton e Smola (2021) se concentram principalmente em DNN e na utilização de *feedback* explícito e implícito, a proposta desta pesquisa se destaca ao aplicar técnicas de engenharia de *features* para otimizar o processo de recomendação, proporcionando uma abordagem mais personalizada e escalável. A pesquisa se posiciona focando na melhoria das entradas de dados para aumentar a precisão e relevância das recomendações em sistemas baseados em redes neurais.

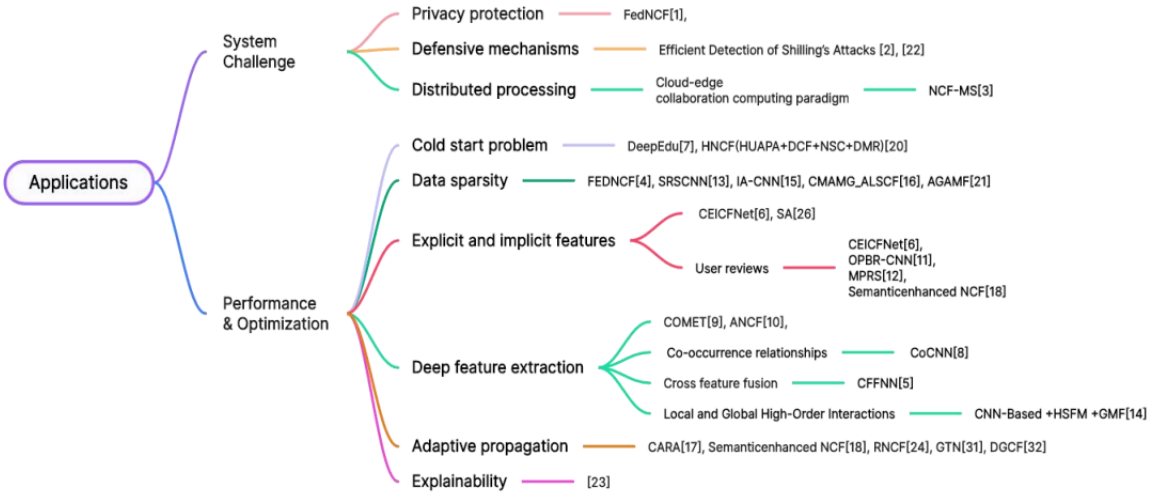
A pesquisa apresentada por Li, Noah e Sarim (2024) aborda o uso de DNNs em SR baseados em CF, destacando os avanços proporcionados por essas arquiteturas na captura de interações complexas e na personalização das recomendações. Diversos estudos revisados pelos autores enfatizam como as DNNs superam limitações de métodos tradicionais, como a *esparsidade* de dados e a dificuldade de modelar relações não-lineares. Entre os conceitos mais importantes abordados no artigo, estão a integração de modelos avançados, como Redes de Grafos , *Autoencoders* e Generative Adversarial Networks (GANs), que são capazes de processar grandes volumes de dados e identificar padrões profundos nas interações usuário-item.

Li, Noah e Sarim (2024) justificam o uso de DNNs pela sua capacidade de capturar características latentes, que métodos tradicionais, como a MF, não conseguem identificar adequadamente. Além disso, a combinação de DNNs com técnicas de CF oferece maior flexibilidade na adaptação a diferentes tipos de dados e melhora a robustez contra problemas recorrentes, como o *cold start* (novos usuários ou itens). As características centrais dessas redes incluem a habilidade de processar relações não-lineares complexas, integrar informações contextuais (como tempo e

localização) e proteger a privacidade dos usuários por meio de estratégias como aprendizado federado.

Li, Noah e Sarim (2024) classificam os trabalhos revisando e separando-os em diferentes tópicos, citando os artigos que utilizam DNN em CFs, conforme apresentado na Figura 2.8. Essa categorização permite uma visão estruturada sobre as abordagens adotadas e reforça as razões pelas quais as DNNs são aplicadas a esse contexto. Os resultados evidenciam as vantagens dessas redes em comparação com métodos tradicionais, demonstrando ganhos significativos na qualidade das recomendações.

Figura 2.8 – Classificação das Aplicações de DNN na CFs.



Fonte: Adaptado de (LI; NOAH; SARIM, 2024).

Os resultados alcançados por Li, Noah e Sarim (2024), com base em uma revisão de aproximadamente 80 artigos publicados entre 2020 e 2024, indicam melhorias significativas na precisão, diversidade e personalização das recomendações. Estudos analisados, como o trabalho de Lin et al. (2023), mostraram que o uso de CNNs em SR aprimorou a extração de padrões locais em interações usuário-item, enquanto modelos como o Federated Neural Collaborative Filtering (FEDNCF), de Perifanis e Efraimidis (2022), combinaram aprendizado federado e CFs neural para melhorar a privacidade e o desempenho preditivo. Além disso, redes como as Redes de Grafos, exploradas por Lin et al. (2022) no modelo GANs, demonstraram eficiência na modelagem de relações em grafos bipartidos, capturando conexões de alta ordem entre usuários e itens.

Apesar dos avanços, Li, Noah e Sarim (2024) destacam limitações importantes, como a alta complexidade computacional das DNNs, especialmente em grandes conjuntos de dados esparsos. Outro ponto abordado é a necessidade de melhorar a interpretação dos resultados gerados por essas redes, um aspecto fundamental para aumentar a confiança dos usuários nos sistemas. Li, Noah e Sarim (2024) também ressaltam que, embora os modelos avançados, como Redes de Grafos e *Autoencoders*, tenham apresentado resultados promissores, a dependência

de grandes volumes de dados rotulados e de alto custo computacional permanece um desafio significativo.

Ao comparar o artigo com outras pesquisas, nota-se que ele se alinha aos esforços de estudos como os de Nguyen et al. (2022) e Manotumruksa, Macdonald e Ounis (2020), que exploraram técnicas avançadas de redes neurais para lidar com dados esparsos e integrar informações sequenciais. Contudo, Li, Noah e Sarim (2024) diferenciam-se ao oferecer uma visão abrangente e estruturada sobre a aplicação de múltiplas arquiteturas de DNN em CFs, estabelecendo um panorama geral sobre os desafios, avanços e oportunidades futuras.

A proposta de Li, Noah e Sarim (2024) avança o estado da arte ao consolidar o conhecimento sobre o impacto das DNNs em SR e sugerir direções claras para pesquisas futuras. A contribuição está em destacar a necessidade de combinar eficiência computacional, proteção de privacidade e modelos mais interpretáveis, além de promover o uso de dados multimodais como um caminho para enriquecer ainda mais as recomendações.

A recomendação personalizada tem sido um dos desafios mais explorados na área de aprendizado de máquina, especialmente com o avanço das DNN. Wang e Tan (2020) propõem um modelo denominado Deep Association Neural Network (DAN), que busca aprimorar os SR ao integrar múltiplas categorias de informações na modelagem das interações usuário-item. O estudo parte da premissa de que métodos tradicionais de CFs e decomposição matricial apresentam limitações ao tratar dados esparsos e não conseguem capturar adequadamente as interações entre diferentes tipos de atributos.

O modelo DAN diferencia-se de abordagens anteriores ao considerar o impacto conjunto de diversas categorias de *features*, como características demográficas dos usuários, informações sobre os itens recomendados e metadados contextuais. Em vez de apenas representar usuários e itens em um espaço vetorial, a rede propõe um mecanismo de combinação entre essas categorias, ampliando a capacidade da rede neural de modelar interações complexas. Essa abordagem evita o problema de superestimação de relações espúrias entre *features* que não deveriam interagir diretamente, conforme destacado pelos autores.

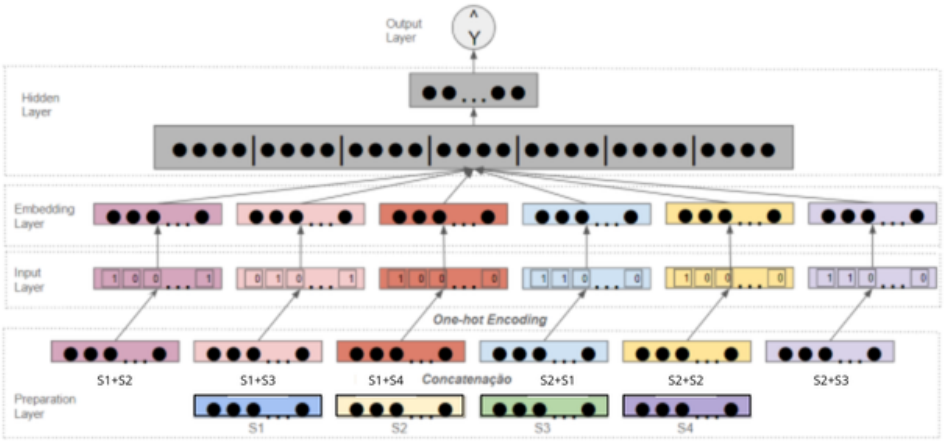
Nos experimentos realizados, os autores utilizam o conjunto de dados MovieLens 1M, que contém avaliações explícitas de filmes, mas que também pode ser transformado em um conjunto de *feedback* implícito, um cenário mais desafiador para os SR. A avaliação do modelo foi conduzida com métricas padrão, como Hit Ratio (HR@k) e Normalized Discounted Cumulative Gain (NDCG@k), demonstrando que a incorporação de múltiplas categorias de *features* melhora significativamente a acurácia das recomendações em comparação com métodos clássicos como (Generalized Matrix Factorization (GMF)) e MLP puro.

Além da versão base do modelo, Wang e Tan (2020) exploram uma variação que também incorpora *meta-features* estatísticas extraídas do conjunto de dados. Essa abordagem visa ajustar dinamicamente os pesos das diferentes *features* com base em suas características estatísticas,

promovendo uma recomendação mais balanceada e justa. Os resultados indicam que a inclusão dessas informações melhora o desempenho do modelo em cenários de *cold start*, onde há poucos dados disponíveis para determinados usuários ou itens.

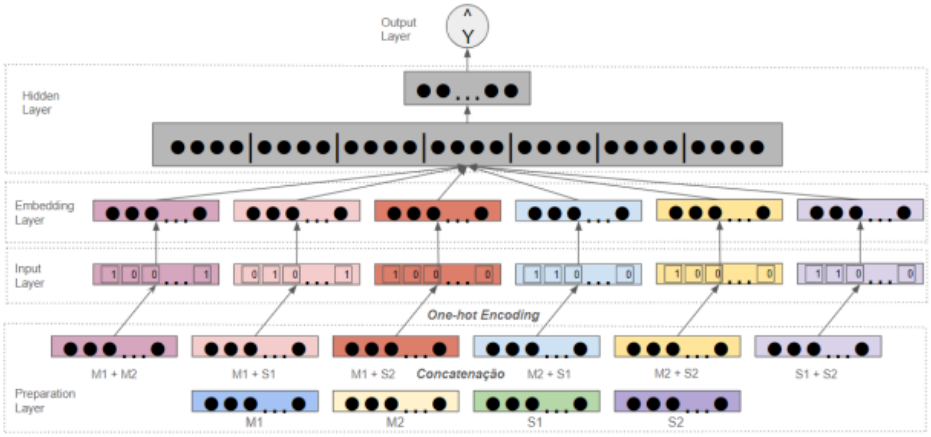
As Figuras 2.9 e 2.10 ilustram o funcionamento dos modelos propostos, destacando as duas variações principais.

Figura 2.9 – Modelo DAN recebendo pontuações de algoritmos constituintes



Fonte: Adaptado de (WANG; TAN, 2020).

Figura 2.10 – Modelo DAN recebendo scores de algoritmos constituintes e *meta-features*



Fonte: Adaptado de (WANG; TAN, 2020).

Ao comparar o trabalho de Wang e Tan (2020) com pesquisas anteriores na área, observa-se que a principal inovação do estudo está na estruturação das *features* em múltiplas camadas, promovendo um aprendizado mais robusto das interações complexas entre usuários e itens. Enquanto abordagens tradicionais utilizam *embeddings* fixos e métodos de fatoração, o DAN se destaca por incorporar relações contextuais e estatísticas para melhorar a personalização das recomendações.

Em síntese, a proposta de Wang e Tan (2020) representa um avanço significativo no campo dos SR baseados em DNN. A inclusão de múltiplas categorias de *features* e *meta-features* amplia

a capacidade dos modelos em oferecer recomendações mais precisas, superando as limitações impostas pela *esparsidade* dos dados. Essa abordagem inovadora abre novas possibilidades para aplicações em *e-commerce*, plataformas de *streaming* e serviços personalizados, sendo um passo importante para o aprimoramento das técnicas de recomendação no contexto de DNN.

A partir da revisão dos trabalhos apresentados, observa-se que a combinação de *meta-features* com técnicas avançadas de DNN tem se mostrado uma abordagem promissora para a melhoria da precisão e da diversidade das recomendações. Além disso, a inclusão do RF como um modelo de *machine learning* no escopo deste estudo proporciona uma oportunidade valiosa para uma comparação abrangente. Essa abordagem não apenas permite analisar o impacto das *meta-features* na performance dos SR baseados em DNN, mas também possibilita investigar se a adoção de um modelo tradicional como o RF resulta em desempenho equivalente ou superior. Assim, o presente estudo se propõe a expandir a aplicação de *meta-features* e a investigar seu impacto em diferentes cenários de recomendação.

3 Desenvolvimento

Este capítulo descreve a metodologia utilizada no desenvolvimento do modelo deste estudo, detalhando as etapas de implementação, o modelo adaptado, as métricas de avaliação e a arquitetura da rede neural. O trabalho é uma continuidade do estudo de Oliveira (2024), que investigou o impacto do uso de *meta-features* em Sistemas de Recomendação (SR) híbridos baseados em redes neurais, utilizando o mesmo conjunto de dados previamente tratados. O modelo proposto aqui foi inspirado na abordagem de Wang et al. (2019), com ajustes para melhor atender aos objetivos desta pesquisa.

3.1 Pipeline de Implementação

O processo de implementação foi estruturado em etapas sequenciais, compreendendo a análise e entendimento dos dados, o treinamento do modelo e a avaliação do desempenho. O pipeline é ilustrado na Figura 3.1 e compreende:

1. **Carregamento dos Dados:** Os *datasets* utilizados (MovieLens 1M, *BookCrossing*, *Jester*) foram carregados e particionados em *folds* para validação cruzada (*K-Folds*), assegurando robustez e generalização do modelo. Foram utilizados os dados já processados, considerando as *features* HR, STREAM e FWLS, já processadas conforme descrito por Fortes et al. (2021).

2. **Pré-processamento:** Constituído dos seguintes tratamentos dos dados:

   - Normalização dos atributos contínuos via *Min-Max Scaling* para o intervalo [0,1].
   - Conversão de atributos categóricos em representações numéricas por meio de *one-hot encoding*.
   - Preparação dos *scores* dos algoritmos constituintes e das *meta-features* para integração no modelo híbrido.

3. **Treinamento do Modelo Adaptado:** O modelo inspirado em Wang et al. (2019) integra redes *RankNet* multiobjetivo e GNN para ranqueamento personalizado. O modelo adaptado, baseado em MLP, recebe os dados processados e é treinado com um conjunto de parâmetros previamente definidos. A saída final combina as funções *Softmax* e *Sigmoid* para gerar probabilidades normalizadas.

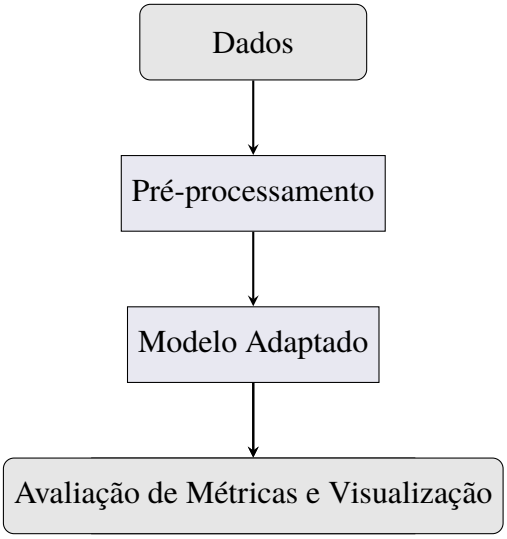
4. **Treinamento do Modelo Random Forest (RF):** Além do modelo de DNN, este estudo também implementou o algoritmo RF para comparação de desempenho. O procedimento de treinamento foi realizado em etapas semelhantes, onde os dados foram carregados e

pré-processados. O modelo foi configurado com o *RandomForestRegressor* da biblioteca *Scikit-Learn* e treinado com os *scores* dos algoritmos constituintes extraídos dos dados. O desempenho do RF foi avaliado utilizando as mesmas métricas que o modelo de DNN, permitindo uma comparação direta entre as abordagens.

5. **Avaliação do Desempenho:** A eficácia do modelo proposto foi avaliada por meio de métricas quantitativas, incluindo RMSE, MAE, R^2, PC e tempo médio de treinamento. Os resultados foram calculados em cada *fold*, com médias para comparação entre estratégias.

6. **Visualização dos Resultados:** Gráficos comparativos foram gerados para facilitar a interpretação do desempenho do modelo.

Figura 3.1 – Fluxo do pipeline de implementação do modelo de recomendação.



Fonte: Elaboração própria.

3.2 Modelo Original

O modelo adotado neste trabalho foi inspirado na abordagem proposta por Wang et al. (2019), que integra técnicas avançadas de DNN para recomendações. O estudo introduz um sistema de *robo-advisor* baseado em DNN, combinando um mecanismo de *RankNet* multiobjetivo com uma estrutura de GNN. O objetivo principal é otimizar a seleção e ranqueamento, fornecendo recomendações personalizadas.

A abordagem proposta por Wang et al. (2019) utiliza três núcleos *RankNet*, cada um treinado para otimizar diferentes objetivos. A rede neural com portas atua como um mecanismo de fusão dessas informações, aprendendo automaticamente a ponderação ideal entre os diferentes núcleos de *RankNet*. Essa estrutura permite que o modelo identifique as informações mais relevantes para a tomada de decisão, proporcionando um sistema de recomendações mais preciso e robusto.

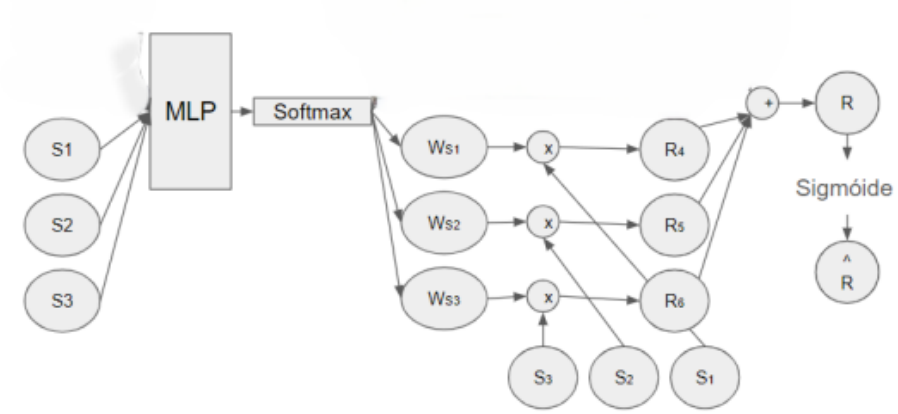
A Figura 2.5 ilustra o funcionamento da arquitetura e como os dados fluem pelo modelo.

3.3 Modelo Adaptado

O modelo utilizado neste trabalho foi desenvolvido com base na abordagem descrita na Seção 2.1.5, onde a estrutura e funcionamento do modelo original são detalhados. No entanto, para atender aos objetivos deste estudo, foram realizadas adaptações que permitiram a incorporação de *meta-features* e dos *scores*, tornando o modelo mais adequado à tarefa proposta. Além disso, foi empregada uma abordagem baseada em regressão para modelar a relação entre as variáveis de entrada e os *scores*, permitindo uma melhor captura das interações entre os dados.

A estrutura do modelo adotado neste trabalho está ilustrada na Figura 3.3. A principal motivação para essa adaptação foi aprimorar a modelagem das interações entre usuários e itens recomendados, explorando um mecanismo de ponderação dos *scores* e das *meta-features*, cujos pesos são ajustáveis ao longo do treinamento da rede.

Figura 3.2 – Estrutura do modelo para a estratégia HR inspirado em Wang et al. (2019).



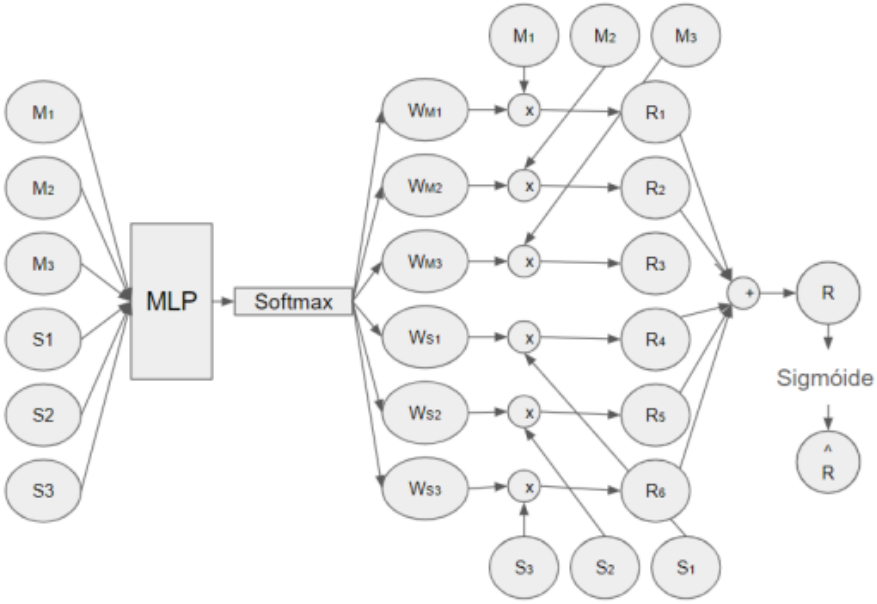
Fonte: Adaptado de (OLIVEIRA, 2024).

O modelo adotado utiliza múltiplos *inputs* representando *meta-features* (*M1, M2, M3*) e (*S1, S2, S3*), combinados por meio de pesos específicos (Wm1, Wm2, Wm3, Ws1, Ws2, Ws3). Esses valores são ajustados por meio desses pesos específicos, que são otimizados durante o treinamento da rede para capturar as interações mais relevantes entre os dados. A última camada emprega as funções de ativação *Softmax* e *Sigmoid*, sendo a primeira responsável por transformar as saídas em probabilidades e a segunda por normalizar os valores finais para um intervalo entre 0 e 1, facilitando a interpretação dos resultados.

3.3.1 Adaptação por Estratégia

Embora a estrutura geral do modelo seja a mesma, as entradas e a forma de combinar os pesos diferem conforme a estratégia de hibridização:

Figura 3.3 – Estrutura do modelo para as estratégias STREAM e FWLS inspirado em Wang et al. (2019).



Fonte: Adaptado de (OLIVEIRA, 2024).

- **HR:** Apenas os *scores* dos algoritmos constituintes (S) são utilizados como entrada, ou seja, as *meta-features* (M) não são incluídas, conforme a Figura 3.2.
- **STREAM:** Tanto *scores* dos algoritmos constituintes (S) quanto *meta-features* (M) são utilizados como entradas. A multiplicação dos pesos é realizada separadamente para cada conjunto, ou seja, os pesos de *scores* multiplicam apenas os *scores* dos algoritmos constituintes, e os pesos das *meta-features* multiplicam apenas as *meta-features*, preservando a contribuição individual de cada tipo de informação, conforme a Figura 3.3.
- **FWLS:** Similar ao STREAM, mas a combinação final é multiplicativa, permitindo capturar interações não lineares entre *scores* dos algoritmos constituintes e *meta-features*, respeitando os pesos de cada entrada de forma independente, conforme a Figura 3.3.

3.3.2 Arquitetura

A rede neural desenvolvida segue uma arquitetura baseada em múltiplas camadas densas, otimizadas para capturar padrões complexos dos dados de entrada. Sua estrutura é composta pelos seguintes elementos:

- **Camada de Entrada**: Contém neurônios correspondentes ao número de *meta-features* e *scores* utilizadas no modelo.
- **Camadas Ocultas**:

  - Primeira camada densa com neurônios e função de ativação ReLU.
  - Segunda camada densa com 4 neurônios e função de ativação *Softmax*, responsável pela normalização dos pesos.
  - Terceira camada densa com 1 neurônio e função de ativação sigmoide, que ajusta os valores finais da recomendação.
  - *Batch Normalization* para estabilizar os valores e melhorar a convergência.
  - *Dropout* (0.1) para evitar *overfitting* e aumentar a generalização do modelo.

- **Camada de Saída**: Gera os pesos ajustados para a combinação dos *scores* dos algoritmos constituintes.

As adaptações realizadas podem permitir explorar novas formas de combinação das *meta-features* e dos *scores*, tornando o modelo flexível para geração de recomendações personalizadas. Dessa forma, a abordagem inspirada no trabalho de Wang et al. (2019) foi aprimorada para melhor atender aos objetivos desta pesquisa, possibilitando uma análise mais aprofundada dos fatores que influenciam a escolha dos itens recomendados.

3.4 Métricas de Avaliação

Para avaliar o desempenho do modelo de recomendação de forma quantitativa e comparativa, considerando métricas de erro, correlação e eficiência computacional, foram utilizadas as seguintes métricas:

- **Root Mean Square Error (RMSE):** Mede a magnitude média dos erros entre as previsões e os valores reais, sendo sensível a grandes discrepâncias. É calculado como:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{3.1}$$

  onde y_i é o valor real do i-ésimo exemplo, $\hat{y}_i$ é a previsão do modelo e N é o número total de exemplos. O RMSE é amplamente utilizado em SR para avaliar a precisão das previsões, conforme discutido por Silveira et al. (2019).

- **Mean Absolute Error (MAE):** Quantifica o erro médio absoluto entre as previsões e os valores reais, fornecendo uma medida direta da discrepância:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{3.2}$$

  O MAE é uma métrica intuitiva que é frequentemente utilizada para medir a precisão em SR, como destacado por Jalili et al. (2018).

- **Coefficient of Determination (R^2):** Avalia a proporção da variância dos dados explicada pelo modelo, indicando a qualidade do ajuste:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{3.3}$$

  onde $\bar{y}$ é a média dos valores reais. Essa métrica é crucial para entender o quanto do comportamento dos dados é capturado pelo modelo, conforme mencionado em Wang, Li e Reddy (2019).

- **Pearson Correlation (PC):** Mede a correlação linear entre as previsões e os valores reais, sendo uma indicação de quão bem o modelo preserva o ranking ou a tendência das avaliações:

$$r = \frac{\sum_{i=1}^{N}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}\sqrt{\sum_{i=1}^{N}(\hat{y}_i - \bar{\hat{y}})^2}} \tag{3.4}$$

  A média é calculada considerando todas as divisões (*folds*) utilizadas na validação cruzada. A PC é uma métrica padrão em recomendações que ajuda a entender a relação entre as avaliações reais e as previstas, como discutido por Nguyen (2024).

- **Tempo de Treinamento:** Avalia a eficiência computacional do modelo, medindo o tempo total necessário para completar o treinamento em cada *fold*. Essa métrica é importante para avaliar a viabilidade do modelo em aplicações práticas (PEYA et al., 2025).

Essas métricas permitem uma avaliação abrangente do modelo, considerando a precisão, a consistência nas previsões e a eficiência computacional.

4 Resultados

Esta seção apresenta os resultados finais obtidos, com foco nas métricas analisadas durante os experimentos realizados. A análise dos resultados é fundamental para entender a eficácia das abordagens testadas. A Seção 4.1 discute a configuração dos experimentos e os dados utilizados, proporcionando um contexto para os resultados. Os resultados fornecem uma visão abrangente das performances das diferentes abordagens testadas, incluindo as estratégias de DNN com *meta-features* e um modelo tradicional de *machine learning*. A Seção 4.2 é fundamental para compreender o desempenho dos modelos, suas limitações e as oportunidades de aprimoramento, contribuindo assim para as conclusões e recomendações da pesquisa.

4.1 Experimentos

Este estudo utilizou a implementação desenvolvida por Oliveira (2024) como base, realizando aprimoramentos para adaptar o modelo às necessidades desta pesquisa. Todo o treinamento e testes foram feitos através do *Google Colab*, em um ambiente de execução com *Python* 3, utilizando uma T4 GPU. O *framework* utilizado para a construção do modelo durante os experimentos foi o *TensorFlow*. A biblioteca utilizada para manipulação e análise de dados foi o *Pandas* 2.1.4. Além disso, parte das análises foi realizada localmente em um computador com processador 12th *Gen Intel(R) Core(TM)* i5-12500H de 2.50 GHz, 16 GB de RAM e sistema operacional de 64 bits.

Em seguida, a Seção 4.1.1 descreve os conjuntos de dados empregados, enquanto a Seção 4.1.2 explica as características utilizadas no modelo. Por fim, a seção 4.1.3 aborda o treinamento dos modelos de DNN e a 4.1.4 discute o treinamento do modelo RF, permitindo uma comparação entre as abordagens.

4.1.1 Configuração do *Dataset*

Os experimentos deste estudo foram conduzidos utilizando um conjunto de dados que engloba três domínios distintos de recomendação:

- *BookCrossing*: Contém avaliações de livros realizadas por usuários, sendo utilizado para analisar preferências em plataformas de leitura.
- *Jester*: Engloba avaliações de piadas, permitindo a análise de padrões de humor e preferências individuais.
- *MovieLens 1M*: Inclui classificações de filmes e é amplamente empregado em pesquisas de recomendação cinematográfica.

Como cada um possui domínios de aplicação diferentes, além de especificidades particulares, como número de usuários, itens e *ratings*, isso faz com que o uso dos três traga contextos diferentes de *esparsidade* aos testes. Todos os valores de *ratings* foram normalizados para faixa de [0,1].

Para garantir a robustez da análise, foi empregada a técnica de validação cruzada com 5 *folds*, que consiste em dividir o dataset em cinco partes iguais. Em cada iteração, quatro partes são utilizadas para o treinamento e uma para teste, repetindo o processo cinco vezes para minimizar a variabilidade dos resultados.

4.1.2 Features

Os conjuntos de *features* utilizados como entradas do modelo de recomendação nos experimentos realizados são compostos por *scores* dos algoritmos constituintes e *meta-features*. Para os algoritmos constituintes e para as *meta-features*, foram utilizados os mesmos *scores* calculados e o mesmo conjunto de medidas estatísticas extraído por Fortes, Freitas e Gonçalves (2017). Todos esses valores foram calculados sobre os três *datasets* mencionados na Seção 4.1.1, já com as avaliações normalizadas.

Os conjuntos de *features* utilizados foram construídos a partir de três estratégias de recomendação: HR, STREAM e FWLS descritas por Fortes, Freitas e Gonçalves (2017), cujas definições detalhadas podem ser consultadas na Seção 2.1 em 2.1.4.3.

Com isso, foi possível verificar se as *meta-features* contribuem para melhores resultados nas recomendações através de análises comparativas, uma vez que duas dessas estratégias utilizam essas *meta-features* no seu conjunto de *features* e outra utiliza apenas algoritmos constituintes.

4.1.3 Treinamento do Modelo de Deep Neural Networks (DNN)

O treinamento foi realizado utilizando o otimizador *Adam*, com função de perda Mean Squared Error (MSE), com o objetivo de minimizar a diferença entre os valores previstos e os reais. Foram configurados limites máximos de 200 épocas para as estratégias HR e STREAM, e 300 épocas para a FWLS. Em todos os casos, aplicou-se a técnica de *Early Stopping*, interrompendo o processo quando a métrica de validação não apresentava melhora após um número definido de épocas (*pacience* = 5 para HR/STREAM e *pacience* = 8 para FWLS). Para acelerar o aprendizado e evitar sobreajuste, reservou-se 15% dos dados de treinamento para validação a cada fold.

As estratégias diferem também nos *batch sizes* e na taxa de aprendizado inicial: HR e STREAM foram treinados com *batch size* de 64 e taxa de aprendizado de 0.0007, enquanto a FWLS utilizou *batch size* de 128 e taxa de aprendizado de 0.001. Além disso, durante o treinamento, foram monitoradas as métricas de *loss* e de RMSE em validação, permitindo acompanhar a evolução do desempenho da rede.

A arquitetura de cada modelo empregou funções de ativação específicas: HR com camadas densas ReLU e saída sigmoide, STREAM com camadas densas e normalização via *softmax* na etapa de ponderação dos *scores*, e FWLS com uma MLP para processamento das *meta-features*, *softmax* para os pesos individuais de cada algoritmo e saída sigmoide. No caso da FWLS, foram ainda analisados os pesos atribuídos a cada algoritmo constituinte e a importância relativa das *meta-features*, possibilitando interpretar como essas variáveis influenciam as recomendações finais.

4.1.4 Treinamento do Modelo de Random Forest (RF)

O processo de treinamento para o modelo de RF é realizado em paralelo ao treinamento do modelo de DNN. Assim como no modelo inspirado por Wang et al. (2019), os dados são carregados e pré-processados de maneira semelhante. O algoritmo RF é configurado para treinar múltiplas árvores de decisão, cada uma alimentada por uma amostra aleatória do conjunto de dados. Isso permite que o modelo capture a variabilidade e as interações complexas entre as características dos dados.

Os dados de entrada para o RF consistem nos *scores* dos algoritmos constituintes. O modelo é treinado com o *RandomForestRegressor* da biblioteca *Scikit-Learn*, que oferece uma implementação eficiente do algoritmo. Durante o treinamento, o modelo ajusta os pesos das árvores com base nas previsões em relação aos valores reais, utilizando medidas de erro, como RMSE e MAE, para avaliar a precisão das recomendações.

Os resultados do treinamento do RF são então comparados aos do modelo de DNN, permitindo uma análise abrangente da eficácia de ambos os métodos. Essa comparação é fundamental para entender as vantagens e desvantagens de cada abordagem em cenários de recomendação, especialmente quanto à precisão e à eficiência computacional.

4.2 Análise dos resultados

A Tabela 4.1 apresenta a comparação média entre as estratégias HR, STREAM, FWLS e RF nos três *datasets* analisados. De modo geral, os resultados indicam que o uso de *meta-features*, especialmente quando incorporado de forma mais expressiva na abordagem FWLS, contribui para melhorias no desempenho preditivo em relação à estratégia HR, que utiliza apenas algoritmos constituintes. A inclusão do RF na comparação fornece uma perspectiva adicional sobre a competitividade de métodos tradicionais de *machine learning* frente às abordagens baseadas em DNN.

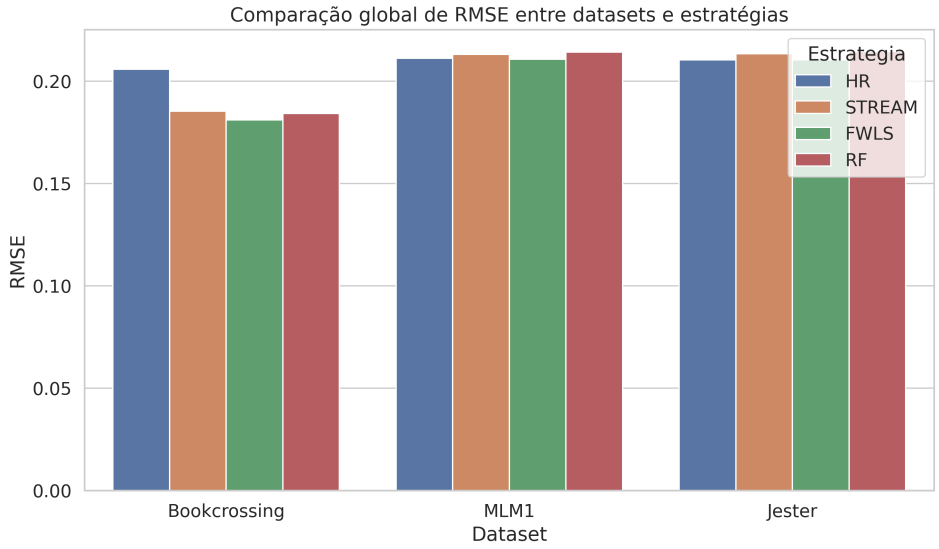
No *dataset* **Bookcrossing**, o FWLS apresentou os menores valores de erro, destacando-se tanto em RMSE (0.1810) quanto em MAE (0.1388), conforme observado na Figura 4.1 e na Figura 4.2. O RF também obteve desempenho competitivo, com valores próximos aos das

Tabela 4.1 – Comparação das métricas de desempenho entre as estratégias HR, STREAM e FWLS nos três *datasets*.

Dataset	Estratégia	RMSE (±)	MAE (±)	R^2 (±)	PC (±)	Tempo (s)
Bookcrossing	HR	0.2058 ± 0.0494	0.1649 ± 0.0505	-0.1090 ± 0.5879	0.3641 ± 0.1353	47.68
	STREAM	0.1852 ± 0.0025	0.1450 ± 0.0015	0.1498 ± 0.0076	0.3924 ± 0.0098	97.12
	FWLS	**0.1810 ± 0.0026**	**0.1388 ± 0.0013**	**0.1878 ± 0.0070**	**0.4369 ± 0.0068**	438.00
	RF	0.1841 ± 0.0025	0.1413 ± 0.0012	0.1601 ± 0.0049	0.4083 ± 0.0053	**27.68**
MLM1	HR	0.2112 ± 0.0008	0.1669 ± 0.0009	0.4226 ± 0.0042	0.6503 ± 0.0034	209.65
	STREAM	0.2131 ± 0.0007	0.1682 ± 0.0008	0.4119 ± 0.0027	0.6425 ± 0.0024	440.60
	FWLS	**0.2106 ± 0.0008**	**0.1666 ± 0.0008**	**0.4258 ± 0.0051**	**0.6534 ± 0.0040**	2071.13
	RF	0.2142 ± 0.0009	0.1686 ± 0.0010	0.4060 ± 0.0037	0.6375 ± 0.0032	**195.18**
Jester	HR	0.2103 ± 0.0011	0.1660 ± 0.0011	**0.3566 ± 0.0069**	**0.5975 ± 0.0061**	135.41
	STREAM	0.2133 ± 0.0007	0.1692 ± 0.0011	0.3380 ± 0.0050	0.5825 ± 0.0052	230.09
	FWLS	**0.2103 ± 0.0013**	**0.1657 ± 0.0015**	**0.3565 ± 0.0082**	**0.5976 ± 0.0070**	659.83
	RF	0.2144 ± 0.0010	0.1689 ± 0.0011	0.3311 ± 0.0045	0.5764 ± 0.0048	**61.48**

Fonte: Elaboração própria.

Figura 4.1 – Comparação do RMSE entre as estratégias avaliadas nos três *datasets*.



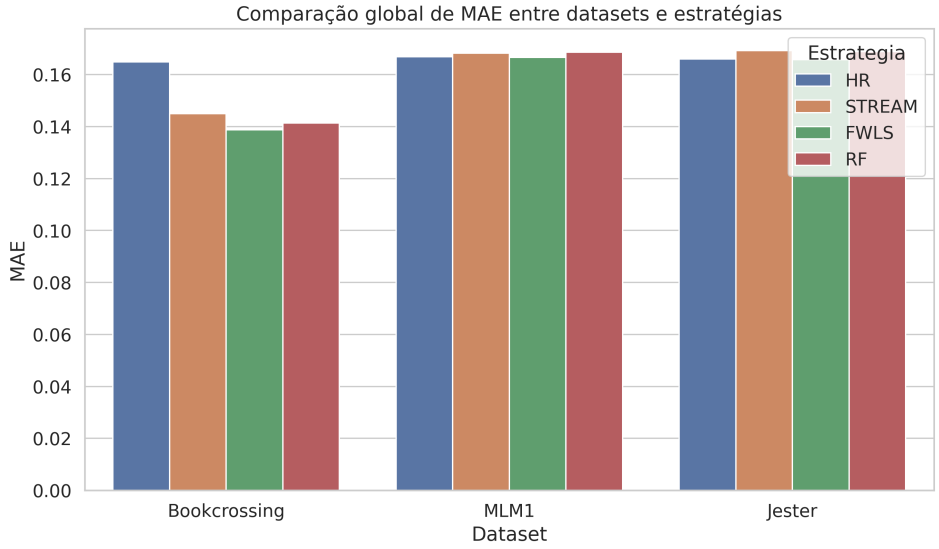
Fonte: Elaboração própria.

abordagens baseadas em DNN, indicando que, embora mais simples, consegue capturar padrões relevantes dos dados.

Resultados semelhantes foram observados nos *datasets* **MLM1** e **Jester**, nos quais o FWLS manteve uma leve vantagem sobre o HR e o STREAM. Ainda que as diferenças absolutas sejam pequenas, elas se mantêm consistentes entre os cenários avaliados, reforçando o impacto positivo da incorporação de *meta-features* na redução do erro preditivo.

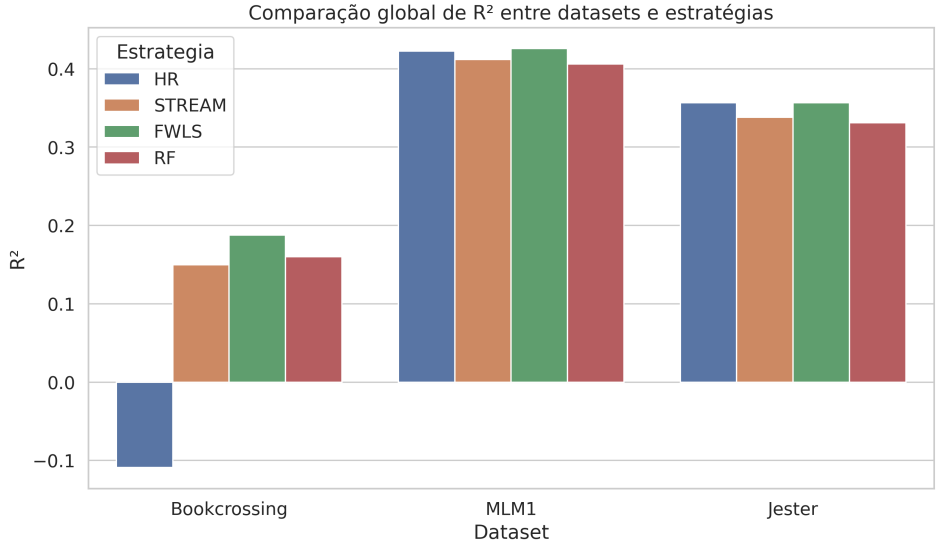
A análise das métricas que avaliam a capacidade explicativa e a correlação entre valores reais e preditos, apresentada nas Figura 4.3 e Figura 4.4. No *dataset* **Bookcrossing**, o FWLS alcançou os maiores valores de R^2 (0.1878) e PC (0.4369), indicando melhor capacidade de explicação da variância dos dados e maior alinhamento entre valores preditos e reais. Observa-se que, na estratégia HR, o valor médio de R^2 no *dataset* **BookCrossing** foi negativo (**-0.1090**).

Figura 4.2 – Comparação do MAE entre as estratégias avaliadas nos três *datasets*.



Fonte: Elaboração própria.

Figura 4.3 – Comparação do coeficiente de determinação (R^2) entre as estratégias avaliadas.

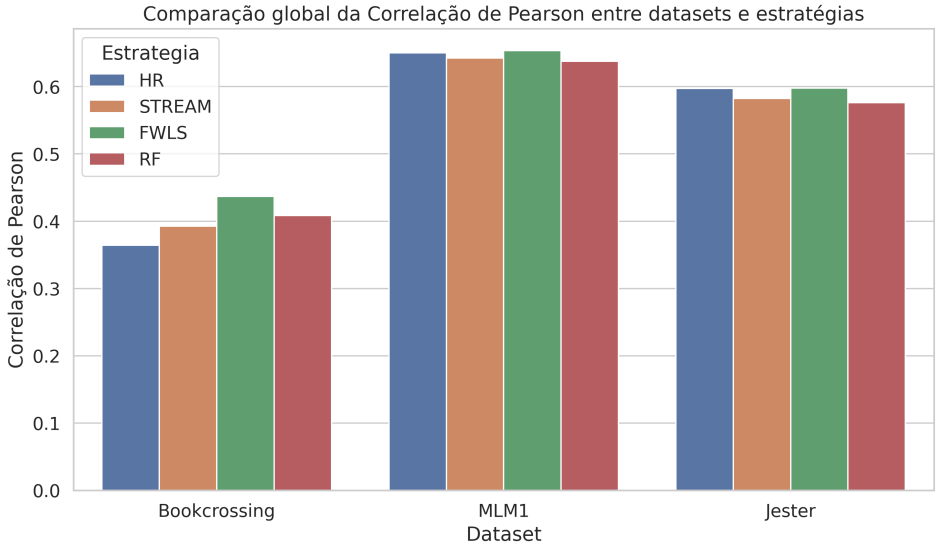


Fonte: Elaboração própria.

Esse comportamento é possível e já discutido na literatura de regressão, ocorrendo quando o erro do modelo é maior do que o erro de um modelo trivial que simplesmente prediz a média dos dados (JAMES et al., 2013) (HASTIE et al., 2009). Em outras palavras, valores negativos de R^2 indicam que o modelo não conseguiu capturar adequadamente a variabilidade presente no conjunto de dados.

Esse resultado pode estar relacionado às características do próprio *dataset BookCrossing*, conhecido por apresentar alta esparsidade e grande variabilidade nas avaliações dos usuários, fatores que tornam a modelagem mais desafiadora em sistemas de recomendação.

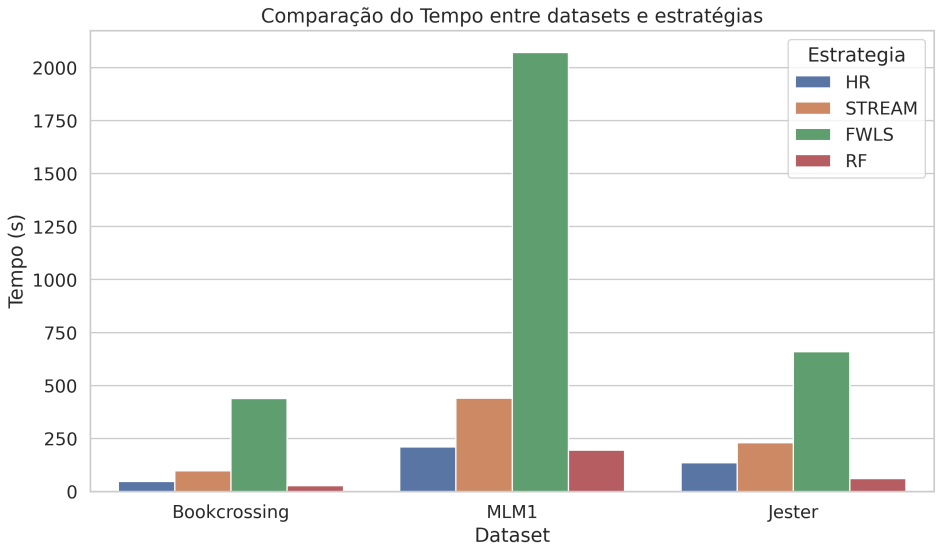
Figura 4.4 – Comparação do coeficiente de correlação de Pearson entre as estratégias avaliadas.



Fonte: Elaboração própria.

Nos *datasets* **MLM1** e **Jester**, as diferenças entre as estratégias foram mais sutis, com valores de R^2 e PC bastante próximos entre FWLS e HR. Ainda assim, o FWLS manteve desempenho superior ou equivalente, sugerindo maior estabilidade do modelo quando aplicado a diferentes domínios.

Figura 4.5 – Comparação do tempo médio de treinamento entre as estratégias avaliadas.



Fonte: Elaboração própria.

A Figura 4.5 evidencia diferenças significativas no custo computacional entre as estratégias. O RF apresentou os menores tempos de treinamento em todos os *datasets*, destacando-se especialmente no **Bookcrossing**, com tempo médio de 27.68 segundos, em comparação com os 438.00 segundos observados para o FWLS. Essa diferença decorre da maior complexidade do

FWLS, que combina múltiplas *meta-features* e algoritmos constituintes em uma arquitetura de DNN.

Embora o FWLS apresente os melhores resultados em termos de precisão, os ganhos observados são relativamente modestos quando comparados ao aumento substancial no tempo de treinamento. Ainda assim, é importante considerar que, em sistemas de recomendação, o treinamento do modelo normalmente ocorre de forma *offline* e, uma vez ajustado, a geração das recomendações pode ser realizada com baixo custo computacional na fase de inferência. Dessa forma, estratégias mais simples, como o RF ou mesmo o HR, continuam sendo alternativas interessantes quando se busca um equilíbrio entre desempenho, complexidade do modelo e custo de treinamento.

5 Considerações Finais

Neste capítulo, serão apresentadas as considerações finais do trabalho, analisando se os objetivos estabelecidos foram alcançados e discutindo a contribuição desta pesquisa para a área de SR híbridos baseados em redes neurais. A Seção 5.1 fornece uma visão geral dos resultados alcançados, enquanto a Seção 5.2 apresenta sugestões para novas investigações.

5.1 Conclusão

Neste trabalho, foi proposta e avaliada uma arquitetura de recomendação baseada em DNN, adaptada da abordagem de Wang et al. (2019), que integra *scores* de algoritmos constituintes e *meta-features* por meio de estratégias híbridas HR, STREAM e FWLS. O objetivo principal foi investigar o impacto da incorporação de *meta-features* no desempenho das recomendações.

Os resultados obtidos demonstram que a utilização de *meta-features*, especialmente quando combinadas de forma multiplicativa como na estratégia FWLS, proporciona melhorias nas métricas de desempenho em relação à abordagem HR, que considera apenas os algoritmos constituintes. Observou-se que, no *dataset BookCrossing*, o FWLS apresentou o menor RMSE (0.1810), menor MAE (0.1388) e maior PC (0.4369), evidenciando que a ponderação dinâmica das *meta-features* permite capturar padrões mais complexos e relevantes nos dados. Nos *datasets* Jester e MovieLens 1M, embora os ganhos tenham sido mais sutis, o FWLS manteve consistência e desempenho superiores ou equivalentes às demais estratégias.

Além disso, ao introduzir o RF como modelo de comparação, a pesquisa ampliou a análise da eficácia das recomendações. O desempenho do RF foi competitivo, especialmente em termos de eficiência computacional. Isso sugere que, em cenários onde a eficiência computacional é um fator relevante, o RF pode ser uma alternativa viável. Entretanto, é importante considerar que, em sistemas de recomendação, o treinamento dos modelos normalmente ocorre de forma offline e, uma vez ajustado, a geração das recomendações pode ser realizada com baixo custo computacional na fase de inferência. Dessa forma, o RF se destaca principalmente pela simplicidade e rapidez no treinamento, mantendo desempenho competitivo em relação às abordagens baseadas em DNN.

Ao comparar esses resultados com trabalhos recentes na área, observa-se que a utilização de *meta-features* e modelos híbridos tem sido considerada uma prática eficiente para melhorar SR. O estudo de Fortes, Freitas e Gonçalves (2017) reporta que a integração de características derivadas dos *algoritmos constituintes* com *meta-features* melhora o RMSE em torno de 5% a 15% em datasets de domínio variado, o que é compatível com os ganhos observados neste estudo. Da mesma forma, abordagens baseadas em DNN, como em Wang et al. (2019), também destacam que o uso de múltiplos núcleos de ranqueamento e de fusão de características permite capturar

interações complexas, corroborando a eficácia da arquitetura adaptada utilizada neste trabalho.

Além das métricas de acurácia, observou-se que o tempo de treinamento aumenta consideravelmente à medida que a complexidade do modelo cresce. O FWLS, embora mais preciso, exigiu até dez vezes mais tempo de processamento do que o HR. Ainda assim, em sistemas de recomendação o treinamento costuma ocorrer offline, de modo que esse custo computacional afeta principalmente a etapa de preparação do modelo, enquanto a geração das recomendações após o treinamento pode ser realizada de forma eficiente. Essa análise é reforçada ao considerar o desempenho do modelo RF, que, apesar de ser menos preciso em alguns casos, apresentou um tempo de treinamento significativamente menor, tornando-se uma alternativa viável em cenários onde a eficiência é crucial.

Com base nos resultados e na análise comparativa com a literatura, pode-se concluir que:

- A inclusão de *meta-features* melhora o desempenho do modelo de recomendação.
- Estratégias híbridas simples, como HR, ainda são úteis como *baseline* devido à sua eficiência computacional e simplicidade de implementação.
- O FWLS, embora mais custoso em termos computacionais, oferece o melhor equilíbrio entre precisão e capacidade de modelar interações complexas.
- O RF, embora menos preciso em alguns casos, apresenta treinamento mais eficiente e desempenho competitivo, destacando a importância de considerar diferentes abordagens em SR.
- As métricas utilizadas (RMSE, MAE, R^2, PC) demonstram a confiabilidade e consistência dos resultados, comparáveis a estudos consolidados na literatura.

5.2 Trabalhos Futuros

Como extensões deste estudo, sugerem-se as seguintes direções:

- Explorar arquiteturas mais profundas ou baseadas em atenção (*attention mechanisms*) para capturar interações ainda mais complexas entre usuários, itens e *meta-features*.
- Avaliar a integração de *embedding layers* para representar itens e usuários, reduzindo a dimensionalidade e, potencialmente, melhorando o tempo de treinamento.
- Investigar estratégias de regularização e de otimização adaptativa para reduzir o custo computacional do FWLS sem comprometer a precisão.
- Implementar uma análise interpretável dos pesos das *meta-features*, permitindo uma compreensão mais detalhada de como cada característica influencia a recomendação final, seguindo tendências de *explainable AI* em SR.

- Realizar comparações adicionais com outros modelos de *machine learning*, como *Support Vector Machines* e *Gradient Boosting*, para avaliar a robustez e a eficácia das recomendações em diferentes contextos.
- Explorar mais experimentações com outros modelos de DNN, como *Transformers* e CNN, para identificar quais arquiteturas oferecem o melhor desempenho em SR.

Portanto, este trabalho reforça que a combinação de DNN com *meta-features* é promissora para SR, oferecendo ganhos em desempenho, com oportunidades futuras para otimização, generalização e interpretação do modelo.

Referências

ADOMAVICIUS, G.; ZHANG, J. Impact of data characteristics on recommender systems performance. *ACM Transactions on Management Information Systems (TMIS)*, ACM New York, NY, USA, v. 3, n. 1, p. 1–17, 2012.

ANSARI, A.; EVGENIOU, T.; XIE, L.; KOHLI, R. Internet recommendation systems. *Journal of Marketing Research*, American Marketing Association, v. 37, n. 3, p. 363–375, August 2000.

ARBIB, M. A. Levels of modeling of mechanisms of visually guided behavior. *Behavioral and Brain Sciences*, Cambridge University Press, v. 10, n. 3, p. 407–436, 1987.

ATHAYDE, M. P. Sistema de recomendação de plantio utilizando aprendizado de máquina. Universidade Federal de Uberlândia, 2023.

BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001.

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

CEDRO, B. R. Desenvolvimento de um sistema fintech de recomendação de investimentos baseado em arquitetura dual de redes neurais. Cachoeiro de Itapemirim, 2025.

DEERWESTER, S.; DUMAIS, S. T.; FURNAS, G. W.; LANDAUER, T. K.; HARSHMAN, R. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, v. 41, n. 6, p. 391–407, 1990.

FORTES, B. et al. Individualized extreme dominance (inded): A new preference-based method for multi-objective recommender systems. *Information Sciences*, v. 572, p. 558–573, 2021. ISSN 0020-0255. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0020025521004977>.

FORTES, R. S.; FREITAS, A.; GONçALVES, M. A. A multicriteria evaluation of hybrid recommender systems: On the usefulness of input data characteristics. In: *International Conference on Enterprise Information Systems*. [s.n.], 2017. Disponível em: <https://api.semanticscholar.org/CorpusID:27078582>.

GANTNER, Z.; RENDLE, S.; FREUDENTHALER, C.; SCHMIDT-THIEME, L. Mymedialite: A free recommender system library. In: *Proceedings of the fifth ACM conference on Recommender systems*. [S.l.: s.n.], 2011. p. 305–308.

GOLDBERG, D.; NICHOLS, D.; OKI, B. M.; TERRY, D. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, ACM, New York, NY, USA, v. 35, n. 12, p. 61–70, December 1992.

HAQUE, M. Z. E-commerce product recommendation system based on ml algorithms. *arXiv preprint arXiv:2407.21026*, 2024.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. H.; FRIEDMAN, J. H. *The elements of statistical learning: data mining, inference, and prediction*. [S.l.]: Springer, 2009. v. 2.

HAYKIN, S. *Redes neurais: princípios e prática*. [S.l.]: Bookman Editora, 2001.

HERLOCKER, J. L. *Understanding and Improving Automated Collaborative Filtering Systems*. Tese (Ph.D. thesis) — University of Minnesota, Minnesota, 2000.

JALILI, M.; AHMADIAN, S.; IZADI, M.; MORADI, P.; SALEHI, M. Evaluating collaborative filtering recommender algorithms: a survey. *IEEE access*, IEEE, v. 6, p. 74003–74024, 2018.

JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. et al. *An introduction to statistical learning: with applications in R*. [S.l.]: Springer, 2013. v. 103.

JANNACH, D.; ADOMAVICIUS, G.; FELFERNIG, A.; LUKIĆ, J. A survey of recommender systems based on deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, v. 29, n. 10, p. 4059–4074, 2017. Disponível em: <https://ieeexplore.ieee.org/document/8529185>.

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.]: ACM, 2008. p. 426–434.

KOREN, Y.; BELL, R.; VOLINSKY, C. Matrix factorization techniques for recommender systems. *Computer*, IEEE, v. 42, n. 8, p. 30–37, 2009.

LEARN, S. Machine learning in python. *Scikit-Learn: Machine Learning in Python—Scikit-Learn*, v. 1, n. 1, 2020.

LI, P.; NOAH, S. A. M.; SARIM, H. M. A survey on deep neural networks in collaborative filtering recommendation systems. December 2024. Disponível em: <https://www.researchgate.net/publication/386374811_A_Survey_on_Deep_Neural_Networks_in_Collaborative_Filtering_Recommendation_Systems>.

LIN, Z.; FENG, L.; GUO, X.; ZHANG, Y.; YIN, R.; KWOH, C. K.; XU, C. Comet: Convolutional dimension interaction for collaborative filtering. *ACM Transactions on Intelligent Systems and Technology*, v. 14, n. 4, p. 1–18, 2023.

LIN, Z.; TIAN, C.; HOU, Y.; ZHAO, W. X. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In: *Proceedings of the ACM Web Conference 2022*. [S.l.]: ACM, 2022. p. 2320–2329.

MANOTUMRUKSA, J.; MACDONALD, C.; OUNIS, I. A contextual recurrent collaborative filtering framework for modelling sequences of venue check-ins. *Information Processing & Management*, v. 57, n. 6, p. 102092, 2020.

MCCANDLESS, M.; HATCHER, E.; GOSPODNETIĆ, O.; GOSPODNETIĆ, O. *Lucene in action*. [S.l.]: Manning Greenwich, 2010. v. 2.

NGUYEN, L. V. Classifications, evaluation metrics, datasets, and domains in recommendation services: A survey. *International Journal of Hybrid Intelligent Systems*, SAGE Publications Sage UK: London, England, v. 20, n. 2, p. 85–100, 2024.

NGUYEN, M.; YU, J.; NGUYEN, T.; YONGCHAREON, S. High-order autoencoder with data augmentation for collaborative filtering. *Knowledge-Based Systems*, v. 240, p. 107773, 2022.

OLIVEIRA, P. H. R. L. de. Trabalho de iniciação científica, *Engenharia de features para meta-aprendizado em sistemas de recomendação baseados em redes neurais*. Centro de convenções da UFOP: [s.n.], 2024. Orientador: Reinaldo Silva Fortes.

PERIFANIS, V.; EFRAIMIDIS, P. S. Federated neural collaborative filtering. *Knowledge-Based Systems*, v. 242, p. 108441, 2022.

PEYA, Z. J.; ISLAM, M. S.; URME, N. J.; AUNY, S. I. Machine learning based potential customer recommendation system for e-commerce using feature selection and xai. In: IEEE. *2025 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. [S.l.], 2025. p. 1–6.

PINTO, E. R.; SAVINIEC, L. Coffee plant: Um sistema de recomendação de cultivares de café baseado em aprendizado de máquina. In: EDITORA CIENTÍFICA DIGITAL. *CIÊNCIAS AGRÁRIAS: TECNOLOGIA, SUSTENTABILIDADE E INOVAÇÃO-VOLUME 2*. [S.l.], 2024. v. 2, p. 46–62.

RAHMAN, A.; HAQUE, Z.; AHAMMAD, M. S. E-commerce product recommendation system using machine learning algorithms. *International Journal of Computer Science and Information Security (IJCSIS)*, v. 22, n. 3, 2024.

REATEGUI, E. B.; CAZELLA, S. C. Sistemas de recomendação. In: CITESEER. *XXV Congresso da Sociedade Brasileira de Computação*. [S.l.], 2005. p. 306–348.

RESNICK, P.; IACOVOU, N.; SUCHAK, M.; BERGSTROM, P.; RIEDL, J. Grouplens: An open architecture for collaborative filtering of netnews. In: *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*. [S.l.]: ACM, 1994. p. 175–186.

RESNICK, P.; VARIAN, H. R. Recommender systems. *Communications of the ACM*, ACM, New York, NY, USA, v. 40, n. 3, p. 55–58, March 1997.

RIEDL, J.; KONSTAN, J. A.; WIERSMA, A.; M., R. C. Combining collaborative filtering with personal agents for better recommendations. In: *Proceedings of AAAI*. [S.l.]: AAAI Press, 1999. v. 35, p. 439–446.

SILL, J.; TAKACS, G.; MACKEY, L.; LIN, D. Feature-weighted linear stacking. arxiv. *arXiv preprint arXiv:0911.0460*, 2009.

SILVEIRA, T.; ZHANG, M.; LIN, X.; LIU, Y.; MA, S. How good your recommender system is? a survey on evaluations in recommendation. *International Journal of Machine Learning and Cybernetics*, Springer, v. 10, n. 5, p. 813–831, 2019.

SINGH, V.; TYAGI, S. Machine learning models for prediction of landslides in the himalayas. In: *Utilizing AI and Machine Learning for Natural Disaster Management*. [S.l.]: IGI Global, 2024. p. 146–174.

TOLEDO, M. *Redes Neurais para Sistemas de Recomendação: uso de Redes Neurais Recorrentes para tratamento de Cold-Start Problem*. [S.l.]: Editora Dialética, 2022.

WANG, H.; WANG, N.; YEUNG, D.-Y. Collaborative deep learning for recommender systems. In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. [S.l.: s.n.], 2015. p. 1235–1244.

WANG, P.; LI, Y.; REDDY, C. K. Machine learning for survival analysis: A survey. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 51, n. 6, p. 1–36, 2019.

WANG, P.-Y.; LIU, C.-S.; YANG, Y.-C.; HUANG, S.-H. A robo-advisor design using multiobjective ranknets with gated neural network structure. In: *2019 IEEE International Conference on Agents (ICA)*. [S.l.: s.n.], 2019. p. 77–78.

WANG, X. n.; TAN, Q. m. Dan: A deep association neural network approach for personalization recommendation. *Frontiers of Information Technology & Electronic Engineering*, v. 21, n. 7, p. 963–980, July 2020. ISSN 2095-9230. Disponível em: <https://doi.org/10.1631/FITEE.1900236>.

XING, L.; MA, D.; MA, B. Service recommendation method based on collaborative filtering and random forest. In: ATLANTIS PRESS. *International Conference on Management, Computer and Education Informatization*. [S.l.], 2015. p. 17–21.

ZHANG, S.; YAO, L.; HE, X.; SONG, L. Deep learning based recommender systems: A survey and new perspectives. *arXiv preprint arXiv:1707.07435*, 2021. Disponível em: <https://arxiv.org/pdf/1707.07435.pdf>.

ZHANG ZACHARY C. LIPTON, M. L. A.; SMOLA, A. J. *Dive into DeepLearning - Release 0.17.1*. Editora, 2021. Disponível em: <https://pt.d2l.ai/d2l-pt.pdf>.

Declaração

Durante a preparação deste documento, eu, **Bruna Cristina Goncalves**, estudante de **graduação** do **Curso de Ciência da Computação**, declaro o uso de Inteligência Artificial Generativa, especificamente da ferramenta **ChatGPT**, desenvolvida pela **OpenAI**, baseada no modelo **GPT 5.2**, versão disponível no período de elaboração deste trabalho (2025-2026), como ferramenta de apoio acadêmico.

A IAGen foi utilizada exclusivamente para fins de revisão textual, tradução de trechos do trabalho e suporte técnico ao *debug* e à organização de código LaTeX, não sendo empregada para a geração de resultados, dados experimentais, análises científicas ou conclusões.

Após o uso da ferramenta, revisei e editei todo o conteúdo gerado, garantindo sua conformidade com os princípios éticos para uso de IAGen (Resolução CONPEP 144) e com os acordos estabelecidos com a pessoa orientadora da pesquisa. Dessa forma, assumo total responsabilidade pelo conteúdo desta publicação.

A IAGen foi utilizada nas seguintes etapas:

- **Revisão de texto**: apoio na correção de erros ortográficos,gramaticais e de pontuação, bem como na melhoria da clareza e coesão da redação acadêmica. O modelo utilizado foi o **ChatGPT (GPT-5.2)**. O *prompt* utilizado foi:

  "Revise este trecho do texto acadêmico, corrigindo erros gramaticais e melhorando a clareza, sem alterar o conteúdo técnico."

- **Tradução**: auxílio na tradução de trechos do trabalho entre os idiomas português e inglês, mantendo o significado técnico original. O modelo utilizado foi o **ChatGPT (GPT-5.2)**. O *prompt* utilizado foi:

  "Traduza este trecho do português para o inglês mantendo o tom acadêmico e o significado técnico."

- **Debug de código LaTeX**: suporte na identificação e correção de erros de compilação, ajustes de formatação e organização estrutural do documento. O modelo utilizado foi o **ChatGPT (GPT-5.2)**. O *prompt* utilizado foi:

  "Identifique e corrija erros neste código LaTeX, mantendo a estrutura original do documento."

[BRUNA CRISTINA GONCALVES]