

# **UNIVERSIDADE ESTADUAL PAULISTA - UNESP**

Instituto de Biociências, Letras e Ciências Exatas - campus de São José do Rio Preto

**IGOR LUIZ LAMPA**

**Sistemas de Recomendação:**  
*uma perspectiva orientada ao Deep Learning*

São José do Rio Preto  
2025

**Igor Luiz Lampa**

**Sistemas de Recomendação:**  
uma perspectiva orientada ao *Deep Learning*

Dissertação, apresentada à Universidade Estadual Paulista (UNESP), Instituto de Biociências, Letras e Ciências Exatas, São José do Rio Preto, para obtenção do título de Mestre em Ciência da Computação.

Área de Concentração: Inteligência Computacional

Orientador: Prof. Dr. Geraldo Francisco Donegá Zafalon

**São José do Rio Preto  
2025**

L237s

Lampa, Igor Luiz

Sistemas de Recomendação : uma perspectiva orientada ao Deep Learning / Igor Luiz Lampa. -- São José do Rio Preto, 2025

100 p.

Dissertação (mestrado) - Universidade Estadual Paulista (UNESP),  
Instituto de Biociências Letras e Ciências Exatas, São José do Rio  
Preto

Orientador: Geraldo Francisco Donegá Zafalon

1. Sistemas de Recomendação. 2. Deep Learning. 3. Collaborative Filtering. 4. Content-based. 5. Abordagem Híbrida. I. Título.

## **IMPACTO POTENCIAL DESTA PESQUISA**

A presente pesquisa em sistemas de recomendação baseados em *deep learning* apresenta um impacto potencial significativo em múltiplas dimensões. Do ponto de vista científico e técnico, o trabalho contribui com o avanço do conhecimento na área de Inteligência Artificial e Aprendizado de Máquina no âmbito de Sistemas de Recomendação, explorando arquiteturas modernas capazes de modelar relações complexas entre usuários e itens. Tais avanços possibilitam maior precisão nas recomendações, promovendo inovações aplicáveis a diferentes setores como educação, cultura, comércio eletrônico, saúde e mídia digital, com efeitos positivos na experiência dos usuários e na eficiência dos serviços oferecidos.

No aspecto social e econômico, a pesquisa reforça a importância de soluções personalizadas que atendam a diferentes perfis de usuários, promovendo acessibilidade e inclusão digital. Além disso, estimula a formação de recursos humanos qualificados em ciência de dados e aprendizado profundo, fortalecendo a inserção local e nacional em temas de alta relevância tecnológica. A internacionalização é obtida por meio da participação em eventos científicos e da publicação em periódicos especializados, assim como foi realizada através da participação no evento International Conference on Enterprise Information Systems - ICEIS, em Angers na França, em que foi possível divulgar o posicionamento desta pesquisa no cenário global.

## **POTENTIAL IMPACT OF THIS RESEARCH**

The present research on recommendation systems based on deep learning demonstrates significant potential impact across multiple dimensions. From a scientific and technical perspective, this work contributes to the advancement of knowledge in the fields of Artificial Intelligence and Machine Learning within the context of Recommendation Systems, by exploring modern architectures capable of modeling complex relationships between users and items. These advances enable greater accuracy in recommendations, fostering innovations applicable to various sectors such as education, culture, e-commerce, healthcare, and digital media, with positive effects on user experience and service efficiency.

From a social and economic standpoint, the research reinforces the importance of personalized solutions that address diverse user profiles, promoting accessibility and digital inclusion. Furthermore, it supports the development of qualified professionals in data science and deep learning, strengthening local and national engagement with highly relevant technological topics. Internationalization is achieved through participation in scientific events and publication in specialized journals, as well as through involvement in the International Conference on Enterprise Information Systems (ICEIS), held in Angers, France, where the positioning of this research was shared within the global scientific community.

**IGOR LUIZ LAMPA**

**SISTEMAS DE RECOMENDAÇÃO:**  
uma perspectiva orientada ao *Deep Learning*

Dissertação apresentada à Universidade Estadual Paulista (UNESP), Instituto de Biociências, Letras e Ciências Exatas, São José do Rio Preto, para obtenção do título de Grau acadêmico Mestre em Ciência da Computação.

Área de Concentração: Inteligência Computacional

Data da defesa: 19/03/2025

Banca Examinadora:

---

Prof. Dr. João Paulo Papa  
UNESP – Departamento de Computação e Estatística – *Câmpus* Bauru

---

Prof. Dr. Henrique Dezani  
Faculdade de Tecnologia de São José do Rio Preto – FATEC

*A Deus, minha esposa, pais, irmãos.*

## **AGRADECIMENTOS**

Primeiramente agradeço a Deus por me fornecer a oportunidade de trilhar esta jornada do mestrado que foi tão sonhada e aguardada por anos. Também por me fortalecer com fé, resiliência, foco e muita dedicação para lidar com todos os desafios inerentes ao processo de concepção, elaboração, desenvolvimento, validação e conclusão deste trabalho.

Agradeço à minha esposa Fernanda que me incentivou desde o momento da inscrição no programa do mestrado até o momento presente de conclusão da dissertação. Que esteve junto comigo durante os desafios enfrentados, bem como me acompanhando durante tantas noites em que eu estendia o horário me dedicar ao desenvolvimento, validação e escrita deste trabalho. Enfim, obrigado por todo o suporte, dedicação e companheirismo.

Aos meus pais Luiz e Rosinei, que sempre confiaram e acreditaram nas minhas habilidades, que sempre estiveram presentes e apoiaram muito para que eu pudesse trilhar o caminho do conhecimento e da ciência desde a graduação. Por desde o início terem apoiado e investido na minha capacitação desde muito jovem.

Ao meu orientador Prof. Dr. Geraldo, que desde a graduação em minha primeira disciplina de Programação I em 2011 cursada no Bacharel em Ciência da Computação, pude aprender o início e a base com ele, e atualmente no mestrado, me orientou ao longo de todo o processo, desde as reuniões para definição e refinamento do tema do trabalho, tanto quanto ao longo do desenvolvimento. Também por todo o apoio e revisão do trabalho que apresentamos na conferência internacional ICEIS 2024 na França. Enfim, muito obrigado por toda a orientação e pela amizade construída ao longo destes anos.

*"Meu filho, empenhe-se na disciplina desde a juventude, e até na velhice você terá a sabedoria. Aproxime-se dela como quem ara e semeia, e espero pelos seus frutos saborosos."*

- Eclesiástico 6:7

## **RESUMO**

A utilização massiva das plataformas digitais proporcionou um aumento exponencial no volume de dados consumidos e gerados diariamente. Por conseguinte, nota-se a ocorrência de uma sobrecarga de dados que afetam diretamente a experiência de consumo dos produtos digitais, seja para consultar uma notícia, consumir um produto em *e-commerce* ou a escolha de um filme em uma plataforma de *streaming*. Nesse contexto, surgem os sistemas de recomendação, os quais possuem a finalidade prover uma forma de eficiente de entender as predileções dos usuários e recomendar itens direcionados a eles. Desse modo, este trabalho visa explanar as técnicas clássicas utilizadas, bem como analisar a utilização em conjunto com deep learning, a qual por meio de resultados avaliados possui maior capacidade de obter relações implícitas entre os usuários e itens e, portanto, prover recomendações de maior qualidade e precisão.

**Palavras-chave:** sistemas de recomendação; deep learning; collaborative-filtering; content-based; abordagem híbrida.

## **ABSTRACT**

*The massive use of the digital platforms has provided an exponential increase at the amount of data diary consumed and generated. Thus, there is a data overload which directly affects the consume experience of digital products, whether at find a news, consume an e-commerce product or to choose a movie in a streaming platform. In this context, emerge the recommendation systems, which have the finality of provide an efficient way to comprehend the user predilections and to recommend direct items. In this way, this work aims to explain the classical techniques already used, as well as to analyze the use together with deep learning, which through evaluated results has a grater capability to obtain implicit relationships between users and items, therefore, provide recommendations with quality and accuracy.*

**Keywords:** *recommendation systems; deep learning; collaborative-filtering; content-based; hybrid approach.*

## LISTA DE FIGURAS

2.1	Tipos de Sistemas de Recomendação . . . . .	25
2.2	Arquitetura <i>Autoencoder</i> . . . . .	35
2.3	<i>Arquitetura Restricted-Boltzmann-Machine</i> . . . . .	36
2.4	Arquitetura AE e RBM . . . . .	39
3.1	Arquitetura do Modelo Híbrido proposto . . . . .	42
3.2	Arquitetura Sistema de Recomendação <i>Content-based</i> . . . . .	43
3.3	Modelo Arquitetural MLP . . . . .	50
3.4	Recomendações do módulo <i>content-based</i> . . . . .	51
3.5	Métodos para obter vetor latente de usuários e itens . . . . .	53
3.6	Geração de Preferência DMFL . . . . .	54
3.7	Pseudocódigo do módulo <i>Collaborative-Filtering</i> . . . . .	57
3.8	Recomendações do módulo <i>Collaborative-Filtering</i> . . . . .	58
4.1	Arquivo users.txt . . . . .	68
4.2	Arquivo movies.txt . . . . .	68
4.3	Arquivo ratings.txt . . . . .	68
4.4	Perda por épocas de treinamento dos módulos . . . . .	72
4.5	<i>F1-Score</i> e Acurácia com 10 épocas de treinamento . . . . .	73
4.6	RMSE e MAE com 30 épocas de treinamento . . . . .	75
4.7	Estabilização da perda ao longo do treinamento dos modelos . . . . .	78
4.8	Combinação de Precisão, <i>Recall</i> e <i>F1-Score</i> com 10, 30 e 50 épocas . . . . .	79
4.9	Comparação entre RMSE e MAE após 10 épocas de treinamento . . . . .	80

4.10	<i>F1-Score</i> e Acurácia após 30 épocas de treinamento . . . . .	81
4.11	Comparação dos parâmetros Precisão, <i>Recall</i> e <i>F1-Score</i> . . . . .	85
4.12	Comparação de RMSE e MAE após 50 épocas de treinamento . . . . .	86
4.13	Comparação de <i>Recall</i> , <i>F1-Score</i> e Acurácia, com 100 mil avaliações	88
4.14	Comparação de <i>Recall</i> , <i>F1-Score</i> , Acurácia, RMSE e MAE . . . . .	89
5.1	Comparação geral de Precisão, Recall, F1-Score e Acurácia . . . . .	93
5.2	Resultados de RMSE e MAE para os conjuntos de dados de treino . .	94
5.3	Comportamento dos erros RMSE e MAE conforme volume de dados .	94

## LISTA DE TABELAS

3.1	Modelo de Arquivo de Usuário . . . . .	44
3.2	Modelo de Arquivo de Filmes . . . . .	44
3.3	Modelo de Arquivo de Avaliações . . . . .	45
3.4	Descrição Sub-redes do Módulo <i>Content-based</i> . . . . .	47
3.5	Bibliotecas de apoio . . . . .	64
3.6	Conjunto de dados úteis no desenvolvimento . . . . .	65
4.1	Conjuntos de dados de validação . . . . .	67
4.2	Resultados com 10 mil avaliações e 10 épocas de treinamento . . . . .	72
4.3	Resultados com 10 mil avaliações e 30 épocas de treinamento . . . . .	74
4.4	Resultados com 10 mil avaliações e 50 épocas de treinamento . . . . .	76
4.5	Resultados com 50 mil avaliações e 10 épocas de treinamento . . . . .	76
4.6	Resultados com 50 mil avaliações e 30 épocas de treinamento . . . . .	77
4.7	Resultados com 50 mil avaliações e 50 épocas de treinamento . . . . .	78
4.8	Resultados com 100 mil avaliações e 10 épocas de treinamento . . . . .	80
4.9	Resultados com 100 mil avaliações e 30 épocas de treinamento . . . . .	81
4.10	Resultados com 100 mil avaliações e 50 épocas de treinamento . . . . .	82
4.11	Resultados com um milhão de avaliações e treino de dez épocas . . . . .	83
4.12	Resultados comparativos do Cenário A com 10 mil avaliações . . . . .	84
4.13	Resultados comparativos do Cenário B com 50 mil avaliações . . . . .	85
4.14	Resultados comparativos do Cenário C com 100 mil avaliações . . . . .	87
4.15	Resultados comparativos do Cenário C com um milhão de avaliações . . . . .	88

## **LISTA DE ABREVIATURAS E SIGLAS**

**AE** Autoencoder

**API** Application Programming Interface

**DFM** Deep Factorization Machine

**DL** Deep Learning

**F1-S** F1-Score

**KNN** K-Nearest Neighbors

**MAE** Mean Absolute Error

**ML** Machine Learning

**MLP** Multilayer Perceptron

**NCF** Neural Collaborative Filtering

**NMF** Non-negative Matrix Factorization

**RBM** Restricted Boltzmann Machine

**RMSE** Root Mean Square Error

**RNN** Recurrent Neural Network

**SVD** Singular Value Decomposition

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>18</b>
1.1 CONSIDERAÇÕES INICIAIS . . . . .	18
1.2 MOTIVAÇÃO . . . . .	19
1.3 OBJETIVO . . . . .	19
1.4 JUSTIFICATIVA . . . . .	20
1.5 HIPÓTESE DE PESQUISA . . . . .	20
1.6 METODOLOGIA . . . . .	20
1.7 MÉTRICAS DE AVALIAÇÃO . . . . .	21
1.8 ORGANIZAÇÃO DO TRABALHO . . . . .	22
<b>2 REVISÃO BIBLIOGRÁFICA</b>	<b>24</b>
2.1 CONSIDERAÇÕES INICIAIS . . . . .	24
2.2 SISTEMAS DE RECOMENDAÇÃO . . . . .	24
2.2.1 <i>Content based</i> . . . . .	25
2.2.2 <i>Collaborative Filtering</i> . . . . .	28
2.2.3 Abordagem Híbrida . . . . .	32
2.3 <i>DEEP LEARNING</i> . . . . .	33
2.3.1 Definição . . . . .	33
2.3.2 <i>Generative Deep Architectures</i> . . . . .	34
2.3.3 <i>Discriminative Deep Architectures</i> . . . . .	36
2.3.4 <i>Hybrid Deep Architectures</i> . . . . .	37

2.4	TRABALHOS RELACIONADOS . . . . .	38
2.4.1	Sistemas de recomendações baseados em <i>deep learning</i> . . .	38
<b>3</b>	<b>MODELO HÍBRIDO DO SISTEMA DE RECOMENDAÇÃO</b>	<b>40</b>
3.1	CONSIDERAÇÕES INICIAIS . . . . .	40
3.2	CONCEITUAÇÃO DO MODELO HÍBRIDO . . . . .	40
3.3	MÓDULO <i>CONTENT-BASED DEEP LEARNING</i> . . . . .	42
3.3.1	Carregamento e Recepção dos dados: . . . . .	43
3.3.2	Pré-Processamento . . . . .	45
3.3.3	Módulo de Aprendizado . . . . .	46
3.3.4	Recomendação . . . . .	50
3.4	MÓDULO <i>COLLABORATIVE FILTERING DEEP LEARNING</i> . . .	51
3.4.1	Aprendizado de características de usuários e itens . . . . .	52
3.4.2	Aprendizado do histórico de relações . . . . .	54
3.4.3	Geração das recomendações . . . . .	56
3.5	OTIMIZADOR ADAM . . . . .	58
3.6	UNIFICAÇÃO DOS RESULTADOS . . . . .	59
3.6.1	Abordagem Híbrida de Soma Ponderada . . . . .	60
3.6.2	Pesos Dinâmicos . . . . .	60
3.6.3	Fusão Bayesiana . . . . .	60
3.6.4	Fusão Sigmoide . . . . .	61
3.6.5	<i>Random Forest</i> . . . . .	62
3.6.6	Fusão <i>XgBoost</i> . . . . .	63
3.6.7	<i>Stacking Meta-Learning</i> . . . . .	63

3.7	IMPLEMENTAÇÃO DO ALGORITMO DE RECOMENDAÇÃO HÍBRIDO . . . . .	63
3.8	CONSIDERAÇÕES FINAIS . . . . .	64
<b>4</b>	<b>VALIDAÇÃO DO MODELO HÍBRIDO DE RECOMENDAÇÃO</b>	<b>66</b>
4.1	CONSIDERAÇÕES INICIAIS . . . . .	66
4.2	METODOLOGIA DE VALIDAÇÃO . . . . .	66
4.3	ETAPAS DE VALIDAÇÃO . . . . .	71
4.4	VALIDAÇÃO DOS MÉTODOS DE UNIFICAÇÃO DE RESULTADOS	71
4.4.1	Cenário A: 10 mil avaliações . . . . .	71
4.4.2	Cenário B: 50 mil avaliações . . . . .	75
4.4.3	Cenário C: 100 mil avaliações . . . . .	79
4.4.4	Cenário D: um milhão de avaliações . . . . .	82
4.5	ANÁLISE COMPARATIVA ENTRE SISTEMAS DE RECOMENDAÇÃO . . . . .	83
4.5.1	Cenário A: 10 mil avaliações . . . . .	84
4.5.2	Cenário B: 50 mil avaliações . . . . .	85
4.5.3	Cenário C: 100 mil avaliações . . . . .	86
4.5.4	Cenário D: um milhão de avaliações . . . . .	88
4.6	CONSIDERAÇÕES FINAIS . . . . .	89
<b>5</b>	<b>CONCLUSÕES E PERSPECTIVAS FUTURAS</b>	<b>91</b>
5.1	CONSIDERAÇÕES INICIAIS . . . . .	91
5.2	CONTRIBUIÇÕES DO TRABALHO . . . . .	91
5.3	PROPOSTAS DE TRABALHOS FUTUROS . . . . .	95

5.4	PUBLICAÇÕES OBTIDAS . . . . .	96
5.5	CONCLUSÃO . . . . .	96
	<b>REFERÊNCIAS</b>	<b>98</b>
	<b>DADOS CURRICULARES</b>	<b>101</b>

## 1 INTRODUÇÃO

### 1.1 CONSIDERAÇÕES INICIAIS

O elevado volume de dados digitais disponíveis atualmente sejam estes gerados em mídias sociais, plataformas de comércio eletrônico, mecanismos de busca, agências de notícias entre outras aplicações, gerou uma sobrecarga de informações. Nesse contexto, os usuários ao consumirem serviços disponíveis na web, se deparam com um excesso de opções, as quais em diversos casos são divergentes aos seus interesses e tampouco úteis para o seu perfil pessoal. Logo, se faz necessário o emprego de alguma metodologia que seja eficiente para prover as informações aderentes ao perfil e interesses de quem a solicitou.

Conforme citado por Sarker e Matin (2021), com o aumento no desenvolvimento da web cerca 2,5 quintilhões de bytes são produzidos diariamente, ademais considera que 90% dos dados já produzidos ocorreram nos últimos dois anos. Nesse sentido, é afirmada que essa elevada quantidade de dados prejudica a tomada de decisões. Assim, este problema contribuiu para desenvolvimento de segmento da computação denominada sistemas de recomendação, o qual através de algoritmos viabiliza a personalização de conteúdos direcionados aos usuários e alinhados às suas expectativas.

O objetivo principal de um sistema de recomendação é indicar aos seus usuários o que eles estão mais dispostos a se interessar, propiciando uma experiência personalizada e evitando excesso de informações desnecessárias (NEGI; PATIL, 2021). No cenário econômico, as organizações já empregam este tipo de sistema, como por exemplo, a Netflix, companhia reconhecida mundialmente pelos seus serviços de *streaming*

de filmes e séries, estima que cerca de 80% dos filmes visualizados por seus clientes foram oriundos de indicações de seu sistema próprio de recomendação. Por esse motivo, grandes corporações empregam esta estratégia com o objetivo de aumentar suas receitas e reter mais consumidores para seus empreendimentos (SARKER; MATIN, 2021).

O alicerce básico de um sistema de recomendação é a sua capacidade de compreender os comportamentos e hábitos dos usuários em relação aos itens que estão interagindo (ZHOU, 2020). Isto é, reconhecer quais as tendências de preferências dos usuários e então indicar os itens que combinem com suas expectativas. Com o intuito de efetivar esta tarefa de compreensão de hábitos dos usuários diversos métodos de recomendação foram desenvolvidos.

## 1.2 MOTIVAÇÃO

No contexto contemporâneo, é perceptível o aumento constante no volume de informações que todos precisam lidar diariamente, seja em função de trabalho, negócios, educação e inclusive lazer. Logo, torna-se essencial lançar mão de estratégias que possibilitem filtrar de modo eficiente e personalizado, as informações mais aderentes às necessidades almejadas. Desse modo, a utilização desse tipo de mecanismo é considerada vital para a melhoria no acesso a informação de diversos tipos de sistemas com o objetivo de impulsionar negócios e, primordialmente, apoiar o processo de tomada de decisão (ZHANG et al., 2019; PETTER.; JABLONSKI., 2023).

## 1.3 OBJETIVO

Este trabalho é proposto no sentido de empregar técnicas inovadoras da grande área de aprendizado de máquina, bem como associá-las aos meios tradicionais de sistemas de recomendação, no objetivo de desenvolver um método híbrido. Neste cenário, busca-se reduzir deficiências detectadas a estes métodos quando utilizados

isoladamente, além de somar suas qualidades ao utilizá-los em conjunto. Logo, neste trabalho concentra-se em proporcionar uma estratégia otimizada, que favoreça o processo de recomendação em termos de qualidade e redução de erros.

#### 1.4 JUSTIFICATIVA

Conforme especificam Da'u e Salim (2019), o desenvolvimento de sistemas de recomendação baseados em *deep learning* tornou-se uma tendência crescente na atualidade. Esta condição é justificada pela capacidade desta nova técnica em fornecer uma melhor aprendizagem de representação da interação entre usuários e itens, quando comparados aos tradicionais métodos previamente estabelecidos na literatura. Neste cenário, o uso híbrido de métodos tradicionais aliados ao *deep learning*, mostrou-se inovador na descoberta de relações não lineares e implícitas entre usuários e itens, logo, recomendações de alta qualidade e não triviais podem ser obtidas.

#### 1.5 HIPÓTESE DE PESQUISA

Os métodos clássicos de sistemas de recomendação não são capazes de produzir resultados tão eficientes quanto ao utilizá-los através de abordagens híbridas. Nesse sentido, a hipótese de pesquisa refere-se à unificação de estratégias clássicas *content-based* e *collaborative filtering*, somadas à métodos de *deep learning*, com a finalidade de obtenção de resultados com maior qualidade, diversidade e precisão, como também com menores taxas de erros nas recomendações efetuadas.

#### 1.6 METODOLOGIA

O princípio metodológico deste trabalho se dá pela análise do levantamento bibliográfico realizado em termos da utilização de sistemas de recomendação, com intuito de identificação de aspectos que demandam melhorias ou novas abordagens. Nesse sentido, através destes estudos já previamente realizados foi notada a existência de um

campo aberto e em ascensão quanto à aplicação de sistema de recomendação apoiados por técnicas de *deep learning*.

Neste contexto, a primeira etapa consiste na verificação crítica dos trabalhos atualmente existentes utilizando sistemas de recomendação tradicionais. Na sequência, efetivação de uma análise aprofundada dos atuais algoritmos e técnicas de *deep learning*, com o objetivo de verificar seu funcionamento e quais as possíveis vantagens e desvantagens de aplicá-las para o processo de realização de recomendações.

O estado da arte da aplicação de sistemas de recomendação baseados em *deep learning* também será avaliado, com a finalidade de estabelecer quais foram as estratégias utilizadas. Ademais, compreender especificamente os motivos que geraram esta necessidade e também quais os aspectos positivos e negativos obtidos. Desse modo, constatar com maior precisão quais são os pontos de melhorias e possíveis otimizações e construções de novas soluções neste segmento.

Diante destas prévias etapas apresentadas, este trabalho se propõe à implementar um sistema de recomendação híbrido baseado em abordagens *deep learning*, com o intuito de melhoria de performance e qualidade dos resultados obtidos. A linguagem de programação a ser utilizada na implementação do sistema de recomendação é o Python, visto que foi constatada sua ampla utilização em diversos estudos que foram realizados nessa área, além da existência de bibliotecas de código fonte aberto, cujas funcionalidades possibilitam maior celeridade no processo de desenvolvimento.

## 1.7 MÉTRICAS DE AVALIAÇÃO

Atualmente há diversas bases de dados de acesso livre que são utilizadas com o intuito de prover dados para a avaliação dos modelos de sistema de recomendação propostos. Conforme Huang et al. (2019) e Sarker e Matin (2021), um dos conjuntos de dados mais utilizados é denominado MovieLens, o qual contém informações de usuários (por exemplo, idade, gênero e profissão), filmes (por exemplo, gênero, título

e ano de lançamento) e as avaliações fornecidas pelos usuários aos filmes. Esta base de dados é coletada e mantida pelo grupo de pesquisa GroupLens da Universidade de Minnesota (RAWAT; TOMAR; KUMAR, 2020; HARPER; KONSTAN, 2015).

Desse modo, este trabalho também utilizará a base de dados MovieLens com o objetivo de efetivação da validação do sistema de recomendação a ser proposto. Ademais, ao aplicar o modelo de recomendação serão utilizadas métricas de validação com o objetivo de verificar os resultados obtidos e considerá-los como parâmetros comparativos a outros modelos já existentes. Serão utilizadas as seguintes métricas: Precisão, *Recall*, *F1-Score*, Acurácia, Erro Quadrático Médio (RMSE) e Erro Absoluto Médio (MAE).

## 1.8 ORGANIZAÇÃO DO TRABALHO

Esta seção apresenta uma breve descrição dos capítulos subsequentes que compõem este trabalho, conforme segue:

**Capítulo 2 – REVISÃO BIBLIOGRÁFICA:** este capítulo concentra-se em descrever a fundamentação teórica utilizada para o desenvolvimento deste trabalho. Logo, são abordados tópicos acerca de Sistemas de Recomendação bem como suas abordagens tradicionais e híbridas. Além disso, conceitos e definições relacionados ao *deep learning*, além de apontar relações na aplicação conjunta com sistemas de recomendação. Também são apresentados os principais trabalhos relacionados da atualidade em termos de sistema de recomendação baseados em *deep learning*.

**Capítulo 3 – MODELO HÍBRIDO DO SISTEMA DE RECOMENDAÇÃO:** descreve o modelo de sistema de recomendação proposto, bem como sua implementação em detalhes.

**Capítulo 4 – VALIDAÇÃO DO MODELO HÍBRIDO DE RECOMENDAÇÃO:** etapa de descrição detalhada dos critérios e cenários de validação, bem como a análise dos resultados obtidos e suas respectivas interpretações.

**Capítulo 5 – CONCLUSÕES E PERSPECTIVAS FUTURAS:** são apresentadas as conclusões e perspectivas futuras deste trabalho.

## 2 REVISÃO BIBLIOGRÁFICA

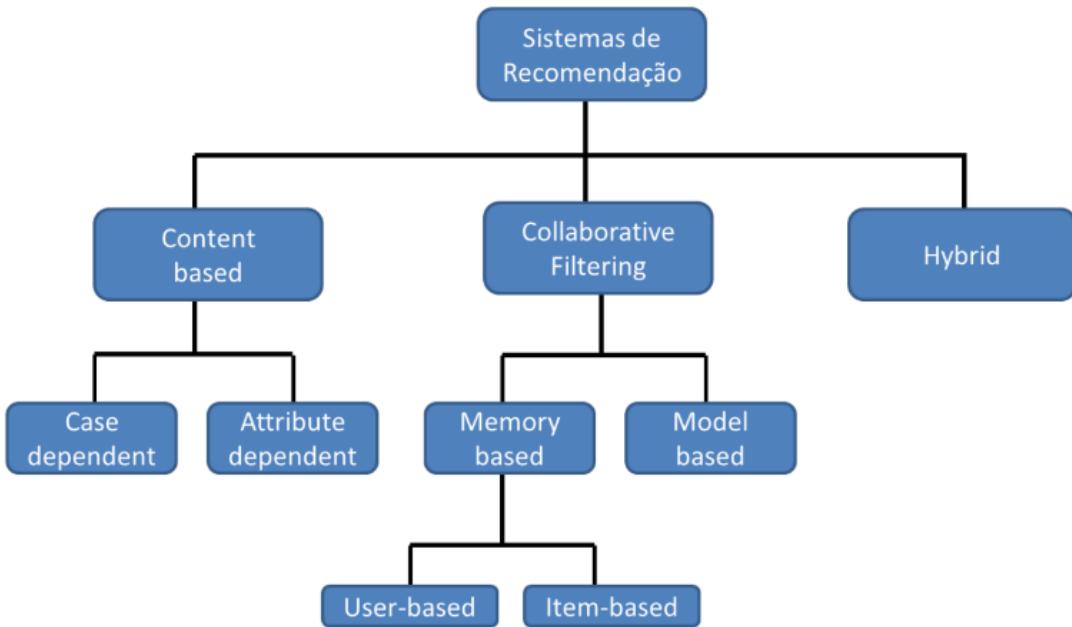
### 2.1 CONSIDERAÇÕES INICIAIS

O objetivo deste capítulo é apresentar a base conceitual teórica utilizada no desenvolvimento deste trabalho. Na Seção 2.2 são abordados os sistemas de recomendação em termos de sua definição e classificação categórica. Na Seção 2.3, o conceito de *deep learning* é definido e classificado em categorias com seus respectivos algoritmos de exemplo. Por fim, na Seção 2.4, os trabalhos relacionados são apresentados.

### 2.2 SISTEMAS DE RECOMENDAÇÃO

Os sistemas de recomendação tradicionais são classificados em três grupos, a saber, *Content based*, *Collaborative Filtering* e Abordagens Híbridas, conforme pode ser visto na Figura 2.1. Nas seções seguintes, estas classificações serão apresentadas em detalhes.

Figura 2.1: Tipos de Sistemas de Recomendação



Fonte: Adaptado de Betru, Onana e Batchakui (2017)

### 2.2.1 *Content based*

A abordagem *content based* depende diretamente de meta-dados referentes aos usuários e itens, isto é, necessita de informações detalhadas relacionadas ao perfil do usuário, bem como dos atributos que descrevem o item. No caso de usuários, podem ser considerados dados como idade, gênero, geolocalização, entre outros dados descritivos. Para os itens, supondo o exemplo de um filme, podem ser observados os atributos de gênero, título, resumo, tags de classificação, tempo de duração, ano de lançamento, atores e demais dados julgados como necessários para descrever o mesmo (SARKER; MATIN, 2021). Conforme Negi e Patil (2021), esta abordagem possui como base características definidas por humanos, as quais auxiliam no processo de recomendação. Desse modo, com estes dados já coletados são realizadas comparações entre o perfil do usuário e os itens, para obtenção das recomendações (BHANUSE; MAL, 2021).

Para estabelecimento do perfil de preferências do usuário, é necessário o emprego de alguma técnica de aprendizado, ou seja, algoritmos que revelem o relacionamento implícito na interação direta entre usuários e itens. Nesse sentido, são geralmente utilizadas, redes neurais artificiais, classificador *Support Vector Machine* – SVM e classificador Bayesiano, os quais são frequentemente empregados para obtenção do perfil do usuário (DA'U; SALIM, 2019).

Um dos fatores positivos da recomendação baseada em conteúdo é independência de usuário, uma vez que, as indicações são realizadas isoladamente, ou seja, desconectada do perfil de preferência dos demais usuários. Outro fator é transparência, denotada pela clareza do processo de obtenção dos itens recomendados, enquanto outros métodos são denominados como caixas-pretas, por não terem uma relação direta entre usuário e itens. Ademais, quando há a inclusão de novos itens a abordagem *content based* é capaz de fazer recomendações, mesmo sem haver prévias avaliações do mesmo, evitando o problema clássico da ausência da primeira avaliação (*first-rater problem*).

Por outro lado, um dos fatores negativos é a ocorrência de obviedade nas recomendações, sendo que esta condição pode resultar em indicações previsíveis as quais não tem variações que são consideradas fundamentais. De acordo com Saat, Noah e Mohd (2018), esse fenômeno é classificado como o problema da bolha do filtro de recomendação, devido o mecanismo desta estratégia ser específico por usuário e não considerar o restante das preferências da comunidade, isto é, forma-se um ciclo de recomendações autorreferenciado pelas próprias predileções do usuário. Logo, com o decorrer da utilização há uma tendência de queda na diversidade dos itens indicados, o que impacta negativamente o processo de recomendação, pois faz com que os usuários sejam privados de novas possibilidades e percam oportunidades que poderiam ser relevantes a eles (GROSSETTI; MOUZA; TRAVERS, 2019).

Apesar da abordagem *content based* ser capaz de lidar com a inserção de novos

itens, quando há inclusão de novos usuários já se torna um obstáculo. Esta condição ocorre em razão da ausência de histórico de interação de novos usuários com os itens disponíveis, o qual seria utilizado para o treinamento e aprendizado de suas preferências. Desse modo, o sistema de recomendação fica em uma condição de impasse por não ter recursos suficientes para sua execução, o que gera a necessidade de uso de alguma outra abordagem mais complexa, ou mesmo, uma indicação generalista e sem critérios personalizados.

Conforme descrito por Bhanuse e Mal (2021), há dois tipos de abordagens *content based*:

- ***Case Dependent Reasoning Method:***

Nesse modelo é feita a associação de itens que estão altamente relacionados com aqueles que já foram apreciados previamente pelo usuário. Há uma tendência de aumento da qualidade das indicações à medida que o usuário interage com os itens disponíveis, uma vez que seu histórico será incrementado e novas conexões poderão ser estabelecidas. Abordagem prática e direta, embora compartilhe do clássico problema da inclusão de um novo usuário, pois não possui referência prévia sobre suas predileções.

Ao considerar um ambiente de *e-commerce* de livros como um exemplo, em que há um usuário que adquiriu um item cuja classificação é ficção científica, então o sistema de recomendação irá considerar que esta categoria de livros está no perfil deste cliente. Por consequência, em futuras interações desse usuário com o sistema, este irá receber recomendações deste gênero literário. É perceptível que este mecanismo depende absolutamente da interação entre o usuário e os itens, por consequência, a ausência deste histórico prévio invalida todo o processo.

- ***Attribute Dependent Method:***

As recomendações são realizadas com base na relação dos atributos dos itens e sua aderência em relação ao perfil do usuário. Nesse sentido, não há dependência em relação à interação dos usuários com os itens, logo a inclusão de novos usuários não geram dificuldades, pois as recomendações utilizarão como base o seu perfil previamente declarado.

Com intuito de exemplificar, considerara-se uma plataforma de *e-learning*, em que há um determinado usuário cadastrado que possui seu perfil com a declaração de área de interesse profissional relacionado à ciência da computação, então o algoritmo de recomendação utilizará este atributo explicitamente definido para lhe recomendar cursos relacionados à sua área. Nota-se que o usuário não precisou interagir com os cursos da plataforma para que esta pudesse lhe fazer recomendações, pois se utilizou apenas dos atributos.

### 2.2.2 *Collaborative Filtering*

*Collaborative Filtering* é um dos métodos mais amplamente utilizados (DA'U; SALIM, 2019) e em oposição à *content based*, esta abordagem não depende dos atributos dos usuários e itens, mas apenas do relacionamento entre eles. Em geral, o mecanismo baseia-se em extrair o histórico de interação de um usuário com os itens e estabelecer comparativos com os registros históricos de interação dos demais usuários com os itens, com a finalidade de obter novas classificações (SARKER; MATIN, 2021). O conceito fundamental empregado por esta abordagem é considerar a suposição de que usuários semelhantes possuem interesses equiparáveis (BHANDUSE; MAL, 2021).

A base desta técnica é a colaboração obtida através da análise de dados de interação dos usuários com os itens, um dos algoritmos mais utilizados é a semelhança de cossenos (GUO; LIU, 2019). Nesse contexto, Obeidat, Duwairi e Al-Aiad (2019) propuseram uma abordagem *collaborative filtering* fundamentada com técnicas clássicas de mineração de dados. Nesse sentido, utilizaram os algoritmos APRIORI e SPADE,

além de técnicas de agrupamento em *cluster*, com a finalidade de obter recomendações a partir da geração de regras de associação obtidas e avaliadas segundo os critérios de suporte, confiança e cobertura ou generalidade.

As recomendações são feitas com base na avaliação do comportamento da comunidade de usuários com os itens, ou mesmos das ligações entre os próprios itens (DA'U; SALIM, 2019). De acordo com Negi e Patil (2021), as características utilizadas como base nesse processo não são definidas por humanos, mas obtidas através de processos de análise de dados, o qual pode ser auxiliado por algoritmos de *machine learning* e *deep learning*.

Em aplicações de sistemas reais é comum a escassez de dados sobre os usuários ou itens, seja pela sua inexistência ou dificuldade de processamento para se obtê-la (PENG et al., 2020). Desse modo, a capacidade deste método em não depender destes atributos para seu funcionamento, é considerada como uma vantagem positiva. Ademais, possui a habilidade de prover indicações com diversidade, uma vez que não considera o usuário com exclusividade e engloba a análise de todos os relacionamentos usuário-item.

Um dos fatores que prejudicam o desempenho desta abordagem é denominado *cold start*, o qual refere-se à condição em que há um novo item ou novo usuário. Neste contexto, há ausência de um histórico do usuário ou do item, que provoca a falta de recursos para reconhecer as preferências. Logo, o sistema será incapaz de gerar indicações assertivas, fato que irá reduzir a performance do sistema de recomendação (FAYYAZ et al., 2020; LIU et al., 2020).

Outro fator que impacta negativamente este método é a escassez de dados, o qual ocorre quando o volume de interação entre usuários e itens é baixo. Essa condição gera uma matriz usuário-item escassa (ISINKAYE; FOLAJIMI; OJOKOH, 2015), que provoca a incapacidade de localizar vizinhos semelhantes e, por conseguinte, um processo de recomendação com baixa assertividade.

O aumento constante do volume de dados empregados no processo de recomendação é também considerado um problema para este método. Desse modo, o problema de escalabilidade, está relacionado com a dificuldade que o método possui ao lidar com o aumento significativo e constante dos dados, uma vez que foi dimensionado para um volume inicial estimado. Logo é fundamental o uso de estratégias que permitam auxiliar na escalabilidade dos métodos, de modo que acompanhe o aumento do volume de dados.

A estratégia *collaborative filtering* é subdividida em dois segmentos, a saber, *memory based* e *model based*. A primeira utiliza de dados referentes às avaliações e classificações registradas, para fazer a análise e então propor novas recomendações, por esse motivo é fundamentada na memória histórica de interações usuário-item. Por outro lado, a segunda abordagem, lança mão de modelos matemáticos e estatísticos, além de métodos de aprendizado de máquina, por exemplo, o *deep learning*, para que possa realizar o treinamento do modelo e, por fim, gerar recomendações (SARKER; MATIN, 2021). Estas duas abordagens são descritas detalhadamente nos tópicos apresentados na sequência.

- ***Memory based:***

*Memory based* também é categorizada em dois tipos, *user based* e *item based*. Na primeira abordagem, o processo de recomendação concentra-se em compreender interesses semelhantes entre os usuários. Em termos práticos, itens são recomendados ao usuário alvo considerando que eles foram classificados ou adquiridos por usuários que se assemelham a ele. Já no método *item based*, o objetivo é estabelecer a relação entre os próprios itens e o interesse por eles, isto é, com base no item que um determinado usuário escolheu realizar recomendações de itens que sejam semelhantes ao mesmo (ANIL et al., 2018).

Nas técnicas baseada em itens e usuários, há fatores que influenciam na escolha entre elas. Primeiramente, a similaridade de itens é considerada com

maior estabilidade, ou seja, a relação estabelecida entre eles é muito provável que esta irá permanecer. Todavia, ao considerar a similaridade entre usuários é o oposto, uma vez que os interesses dos usuários evoluem e se transformam com o tempo, por consequência, demandarão novos cálculos de similaridade com maior frequência. Ademais, outro fator positivo da técnica *item based* é que, em geral, há mais usuários do que itens, logo, a matriz item-item é menor em dimensões do que a usuário-usuário, ou seja, pode significar uma vantagem competitiva em ambientes com limitação de recursos, seja este em relação a tempo ou mesmo *hardware*.

- ***Model based:***

Por outro lado, a categoria de algoritmos de *collaborative filtering model based*, utilizam de técnicas de estatística e aprendizado de máquina para realizar treinamentos do modelo e classificações. Nesse contexto, o cerne da estratégia é detectar qual é a probabilidade de um determinado usuário em classificar um item como de acordo com o seu interesse ou não, com base nas classificações previamente realizadas (BHANUSE; MAL, 2021). A matriz de fatoração é um algoritmo clássico desta área, cuja função básica é a partir de uma matriz escassa de usuário-item, produzir valores probabilísticos para as lacunas sem preenchimento, isto é, itens sem prévia avaliação, as quais serão utilizadas para as recomendações.

Neste ínterim e diante das contribuições científicas na área de aprendizado de máquina e primordialmente avanços na subárea de *deep learning*, possibilitou que este fosse um aliado fundamental nos sistemas de recomendação. Sendo que o emprego de modelos, tais como *Autoencoders*, *Convolution Neural Network – CNN*, *Recurrent Neural Network – RNN*, entre outros, possuem o objetivo de aperfeiçoar o desempenho dos tradicionais algoritmos de recomendação ao prover maior eficiência e menor taxa de erros.

### 2.2.3 Abordagem Híbrida

Os sistemas de recomendação híbridos surgem a partir da combinação de aspectos positivos de uma ou mais estratégias de sistemas de recomendação, com o intuito de prover uma metodologia otimizada, mas também para mitigar as deficiências individuais inerentes a elas quando utilizadas isoladamente (DA'U; SALIM, 2019). A hibridização do processo de recomendação pode ser obtida pela combinação de diversas técnicas, por exemplo, há a clássica união de algoritmos *content based* com os *collaborative filtering*, a qual possui a função de prover maior acurácia das previsões resultantes.

Sarker e Matin (2021) desenvolveram um método híbrido para recomendação, cuja base é composta por um modelo baseado em fatoração de matrizes, isto é, *collaborative filtering model based*, combinado com técnicas de *deep learning*. Conforme divulgaram, foi possível descobrir relações adicionais implícitas a itens e usuário, as quais permitiram inferir com maior precisão novas recomendações. Além disso, ainda empregaram a abordagem *content-based*, em que consideraram informações demográficas disponíveis em relação a usuários e itens, para prover melhores resultados.

Em resumo, a ideia é basicamente unir de modo apropriado técnicas já existentes de sistemas de recomendação, inclusive unindo-as com áreas correlatas e que podem ser utilizadas em conjunto e com a finalidade de aprimorar os resultados. Além disso, diversas pesquisas da atualidade revelam a importância e o aumento da qualidade dos resultados obtidos quando estratégias de recomendação são utilizadas em conjunto. Umas das áreas que mais cresce na aplicação conjunta com algoritmos tradicionais de sistemas de recomendação é denominada *deep learning* (ZHANG et al., 2019), cuja definição e casos de utilização serão discutidos na Seção 2.3.

## 2.3 DEEP LEARNING

### 2.3.1 Definição

A evolução constante da área computacional de aprendizagem de máquina proporcionou o desenvolvimento de uma nova classe de técnicas, denominada *deep learning*. Este conceito é considerado a nova geração de redes neurais artificiais, o qual tradicionalmente é um dos pilares da inteligência artificial e aprendizado de máquina (DA’U; SALIM, 2019). O objetivo de sua aplicação consiste em aperfeiçoar a representação da aprendizagem através de múltiplas camadas e estágios de processamento de dados, sendo capaz de aprender vários níveis de representações e abstração de dados (ZHANG et al., 2019), por conseguinte, ao avançar entre camadas prover maior grau de conhecimento de relações implícitas.

Na atualidade, a aplicação de *deep learning* está em crescimento e expansão, destacando-se em resultados satisfatórios em áreas como visão computacional, processamento de linguagem natural, processamento de imagens e em sistemas de recomendação (XU et al., 2021). Conforme destacou Da’u e Salim (2019), desde as primeiras publicações envolvendo *deep learning* com sistemas de recomendação, houve um crescimento acelerado de estudos correlatos, de modo que a maior parte foi publicada nos últimos seis anos.

Neste ínterim é verificada uma demanda massiva tanto da academia quanto na indústria para a utilização de *deep learning* a uma vasta gama de aplicações, com o intuito de obter proveito de sua intrínseca capacidade de lidar com tarefas complexas e de obter resultados aprimorados. Além disso, esta metodologia surpreende não apenas pelo aumento da performance em diversos aspectos nestas áreas como também pela sua capacidade de aprendizado de representações a partir do início da aplicação, isto é, do estágio zero. (ZHANG et al., 2019)

A efetividade demonstrada em diversos estudos científicos quanto à aplicação desta

abordagem com sistemas de recomendação, possibilitou o desenvolvimento e avanço na utilização deste método. Nesse contexto, foram desenvolvidos diversos algoritmos que combinam estas duas abordagens, favorecendo o surgimento de uma gama de aplicações emergentes e capazes de produzir resultados com maior performance do que comparado aos algoritmos tradicionais. (PENG et al., 2020; SARKER; MATIN, 2021)

As arquiteturas de *deep learning* podem ser categorizadas em três classes, a saber, *generative*, *discriminative* e *hybrid*. Estas serão apresentadas nas seções subsequentes:

### 2.3.2 *Generative Deep Architectures*

São aplicadas essencialmente a métodos de aprendizado não supervisionado. Por definição estas arquiteturas trabalham na tentativa de caracterizar propriedades de alta correlação dos dados visíveis a fim de gerar padrões. Classificados como generativos, pois em grande parte das redes profundas desta classe normalmente utilizam da própria rede para gerar amostras. Exemplos:

#### ***Autoenconders – AE***

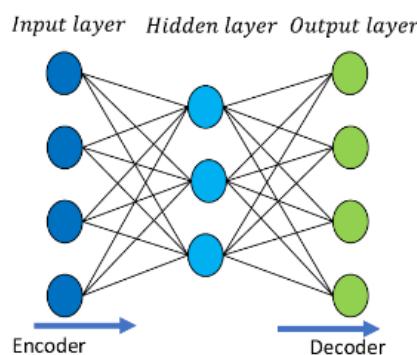
Por definição são treinados para copiar suas entradas de dados para suas saídas, também são compostos basicamente por três camadas, a de entrada e de saída, além de uma camada intermediária oculta. Estes algoritmos não supervisionados são capazes de lidar com dados de entrada que apresentam ruídos, de modo a prover aprendizado de estruturas complexas e hierárquicas. No contexto de sistemas de recomendação são empregados para apoiarem o preenchimento de dados ausentes nas matrizes de interação ou também no aprendizado de aspectos de menor dimensão na camada de saída.

A capacidade do *autoencoder* em favorecer a representação de aprendizado é uma de suas qualidades. Desse modo, há estudos em que utilizam este modelo para apren-

dizado da representação de usuário e itens, com o intuito de melhorar os resultados.

Um dos fatores negativos aliados a este modelo é sua incapacidade em obter uma solução otimizada. Ademais, em razão de seu elevado grau de parametrização, o processo de treinamento deste modelo despende de elevado tempo computacional. A Figura 2.2, ilustra a estrutura base de um autoencoder.

Figura 2.2: Arquitetura *Autoencoder*



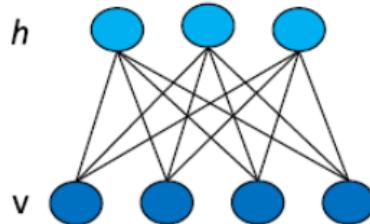
Fonte: Da'u e Salim (2019)

### ***Restricted-Boltzmann-Machine – RBM***

É uma rede neural de duas camadas, a primeira é visível composta pelas variáveis de entrada, e a segunda é denominada oculta, com variáveis também ocultas, de acordo com a Figura 2.3. Há uma conexão completa entre neurônios da camada visível com a oculta, sendo que não há interligações internas à mesma camada. O alto custo computacional relacionado ao treinamento do modelo com elevada parametrização é um aspecto negativo.

Em sistemas de recomendação RBM é utilizado no processo de extração de características de alto nível a partir de dados brutos fornecidos como entrada. Nesse contexto, são úteis para reduzir o problema de *cold start*.

Figura 2.3: Arquitetura Restricted-Boltzmann-Machine



Fonte: DAU; SALIM, 2019

### ***Deep Belief Network – DBN***

É uma estrutura multicamadas que emprega o modelo RBM como base para extrair representação de característica profunda dos dados. No âmbito de sistemas de recomendação é utilizado para redução da dimensionalidade e também para extrair características de dados escassos, com o objetivo de solucionar o problema de *cold start*.

### ***Generative Adversarial Network – GAN***

Constituída por duas redes neurais, sendo um gerador e um discriminador. Essas são treinadas simultaneamente através de um mecanismo de competição entre si, denominado *minimax game*. (ZHANG; YAO; SUN, 2017)

#### **2.3.3 Discriminative Deep Architectures**

São aplicados especificamente a aprendizado supervisionado e baseia-se em prover uma função discriminativa para classificação de padrões. As arquiteturas mais populares são CNN, RNN e MLP, conforme descritas abaixo.

### ***Convolutional Neural Network – CNN***

É uma rede neural artificial direta que baseia-se em camadas de convolução e utiliza operações de agrupamento. Desse modo, é capaz de capturar características globais e locais, além de aumentar consideravelmente a eficiência e precisão do sistema de recomendação. CNN pode ser utilizado no aprendizado da representação de carac-

terísticas oriundas de diversas fontes como imagem, áudio, texto, vídeo, entre outros.

### ***Recurrent Neural Network – RNN***

O funcionamento consiste de laços de repetição e armazenamento de dados anteriores para relembrar computações realizadas previamente. Útil para lidar com comportamento sequencial de usuários e extração de suas características, também na dinâmica temporal na interação usuário-item.

### ***Multilayer Perceptron – MLP***

Considerada a arquitetura de *deep learning* mais simples e uma das mais empregadas em conjunto com sistemas de recomendação. É uma rede neural artificial direta que utiliza o conjunto de entrada para gerar o conjunto de saída, além de camadas intermediárias ocultas. Seu funcionamento baseia-se em transformar modelos lineares em não lineares para obter performance neural. Nesse contexto, é utilizado em sistemas de recomendação para modelar características latentes de usuário e itens, bem como amenizar o problema de *cold start*.

#### **2.3.4 *Hybrid Deep Architectures***

As abordagens híbridas de *deep learning* referem-se à aplicação conjunta dos métodos generativos e discriminativos. Na literatura há estudos que utilizam a combinação de CNN com RNN, CNN com AE e RNN com AE (DA’U; SALIM, 2019). Em resumo, são utilizados em conjunto com a finalidade de realizar a fusão dos aspectos positivos e úteis de cada abordagem, em detrimento aos fatores negativos gerados ao aplicá-los isoladamente.

## 2.4 TRABALHOS RELACIONADOS

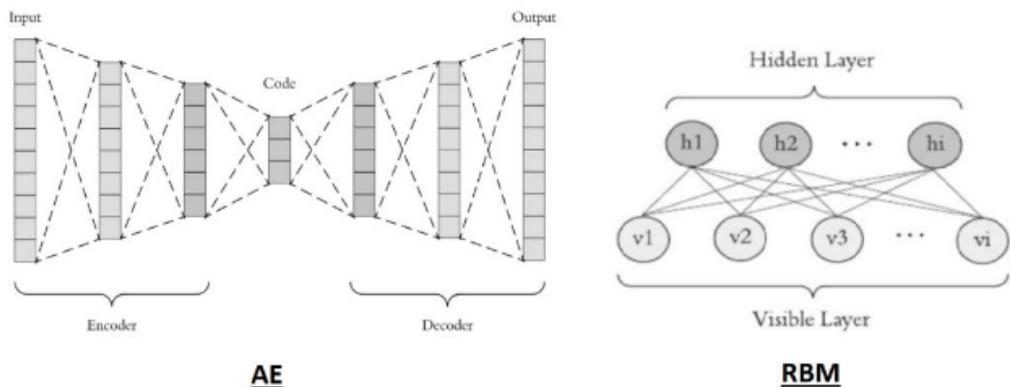
A partir dos estudos das recentes pesquisas envolvendo sistemas de recomendações apoiados por técnicas de *deep learning* foi possível constatar quais as abordagens que estão sendo empregadas. Além disso, identificar quais são os resultados obtidos, melhorias e possíveis desvantagens. Na sequência, serão apresentados alguns destes estudos.

### 2.4.1 Sistemas de recomendações baseados em *deep learning*

Zhou (2020) propôs uma abordagem de sistema de recomendação com o objetivo de reduzir o problema de alto custo computacional inerente aos algoritmos tradicionais quando há inclusão de novos itens ou usuários. Nesse sentido, utilizou de redes neurais profundas para realizar a busca de conteúdos semelhantes para novos usuários ou novos produtos, e então efetivar a inclusão dinamicamente no sistema original. Por meio desta estratégia, reduziu o número de recálculos que seriam necessários e ao comparar com outros algoritmos utilizando o conjunto de dados FilmTrust, verificou-se melhores resultados dos parâmetros erro quadrático médio (RMSE) e erro absoluto médio (MAE).

Uma metodologia *collaborative filtering* aplicada com *deep learning* foi proposta por Negi e Patil (2021), cujo objetivo é unir estas duas áreas e obter melhores resultados nas recomendações. Foram construídos dois modelos, um baseado em *Stacked Autoencoder* – AE, com a função de identificar múltiplas representações comprimidas dos mesmos dados, e outro fundamentado em *Restricted Boltzmann Machine* – RBM, usado na etapa de construir as recomendações, conforme Figura 2.4. A validação utilizou a base de dados MovieLens e constatou que ao analisar as métricas de precisão, recall e erro quadrático médio, a abordagem baseada em *autoencoder* obteve melhores resultados.

Figura 2.4: Arquitetura AE e RBM



Fonte: NEGI; PATIL, 2019

Os autores Sarker e Matin (2021), definiram um sistema de recomendação híbrido baseado em fatoração de matriz e redes neurais profundas, além de utilizar informações auxiliares sobre os usuários e itens. O modelo proposto visa obter relacionamentos internos e implícitos entre os usuários e os itens, para isso utiliza de matriz de fatoração e o modelo *Multilayer Perceptron – MLP*. O processo de validação utiliza a base de dados MovieLens e denota que os resultados obtidos foram melhores do que outros métodos populares, o que indica que sua utilização pode otimizar a experiência de usuário.

Por outro lado, Yuyan, Xiayao e Yong (2019), propuseram um método de recomendação baseado em uma subclasse do *deep learning* denominada *deep reinforcement learning*, com o objetivo de melhorar a acurácia e lidar com a escassez de dados. Baseia-se nas características dos usuários, itens e utiliza uma rede neural convolucional, ademais o processo de validação também emprega a base de dados MovieLens e métrica do erro quadrático médio. Os resultados obtidos denotam melhores resultados do que comparado a abordagens tradicionais de algoritmos colaborativos, o que possibilita maior assertividade nas recomendações realizadas.

### 3 MODELO HÍBRIDO DO SISTEMA DE RECOMENDAÇÃO

#### 3.1 CONSIDERAÇÕES INICIAIS

Este capítulo possui o objetivo de descrever o modelo proposto para o sistema de recomendação. Nesse sentido, serão apresentados os fundamentos utilizados para a elaboração do modelo híbrido para o sistema de recomendação proposto neste trabalho, cujas etapas são baseadas em estratégias *collaborative filtering* e também *content-based*, ambas apoiadas por algoritmos de *deep learning*. Nas seções seguintes serão descritas em detalhes as fases de elaboração e desenvolvimento.

#### 3.2 CONCEITUAÇÃO DO MODELO HÍBRIDO

Conforme observado através da revisão literária, os trabalhos que utilizam *deep learning* como ferramenta de apoio para os sistemas de recomendação, em geral, lançam mão da estratégia *collaborative filtering* (DELLAL-HEDJAZI; ALIMAZIGHI, 2020). Isto é, são abordagens que estão diretamente relacionadas a extrair padrões e inferir previsões com base no histórico de interação entre itens e usuários.

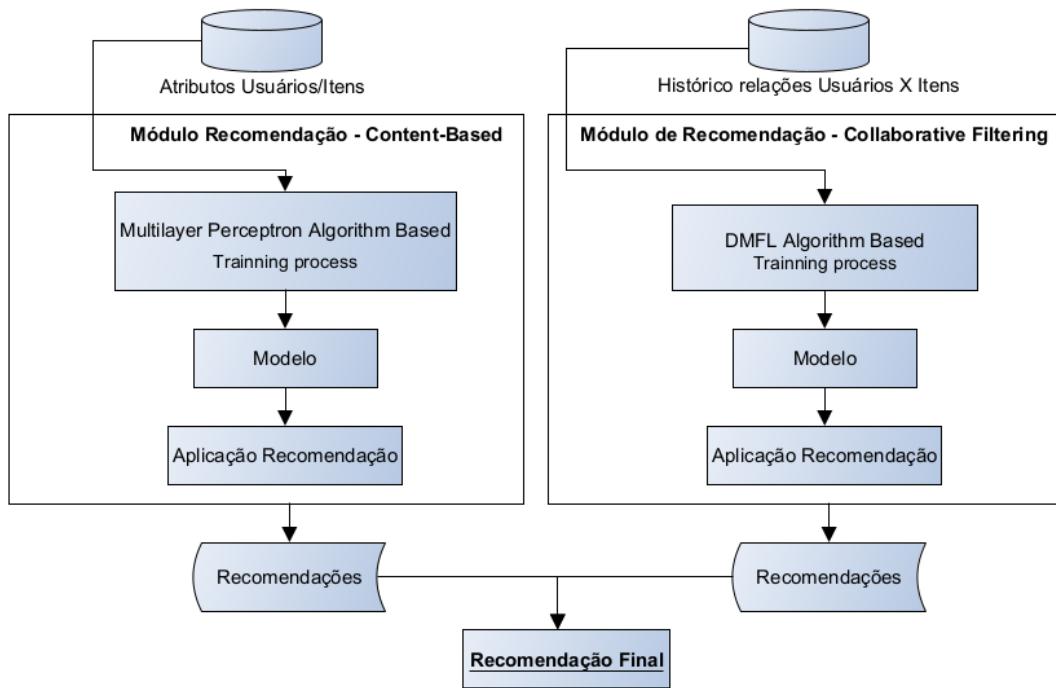
Estas abordagens proveem alta diversificação nos resultados das recomendações e possuem capacidade de lidar com a evolução constante do volume de dados em razão do uso de *deep learning*. Todavia, são impactadas pelo clássico problema de *cold start* e escassez de dados. Logo, quando há nenhum ou pouco histórico de interações, ou mesmo a inclusão de novos usuários e/ou itens esta abordagem, a qual é a mais explorada na previsão baseada em *deep learning*, mostra-se ineficiente para lidar com este problema.

Por outro lado, apesar da abordagem baseada em conteúdo ser pouco explorada por meio da implementação de algoritmos baseados em *deep learning*, ela é fundamental para tratar o problema de *cold start* e escassez de dados. Nesse sentido, é uma técnica capaz de prover predileções, por meio de atributos descritivos individuais, independentemente das relações históricas de interação entre usuários e itens.

Nesse contexto, fundamenta-se a ideia da utilização de um modelo híbrido que seja capaz de agregar as qualidades da estratégia *collaborative filtering* juntamente com a *content-based* e que seja implementado por meio de algoritmos fundamentais de *deep learning*. Ademais, é discutido por Huang et al (2019), que os algoritmos de recomendação híbridos ainda são pouco explorados por abordagens de aprendizado profundo, e, portanto, é uma área com abrangência para ser explorada para contribuições inovadoras e significativas.

De modo geral, a arquitetura do sistema de recomendação proposto é organizada em dois módulos principais, a saber, *Content-Based Deep Learning* e *Collaborative-Filtering Deep Learning*. O primeiro utiliza como entrada os dados de atributos de usuários e itens, e a partir deles gera um modelo que é utilizado para geração das recomendações. Já o segundo utiliza dados históricos de relação entre usuários e itens para treinar o modelo e gerar predileções. Ao final, ambos os resultados são combinados e uma lista final de recomendação é apresentada. Nas seções 3.3 e 3.4 são apresentados os mecanismos de funcionamento interno de cada um dos módulos.

Figura 3.1: Arquitetura do Modelo Híbrido proposto



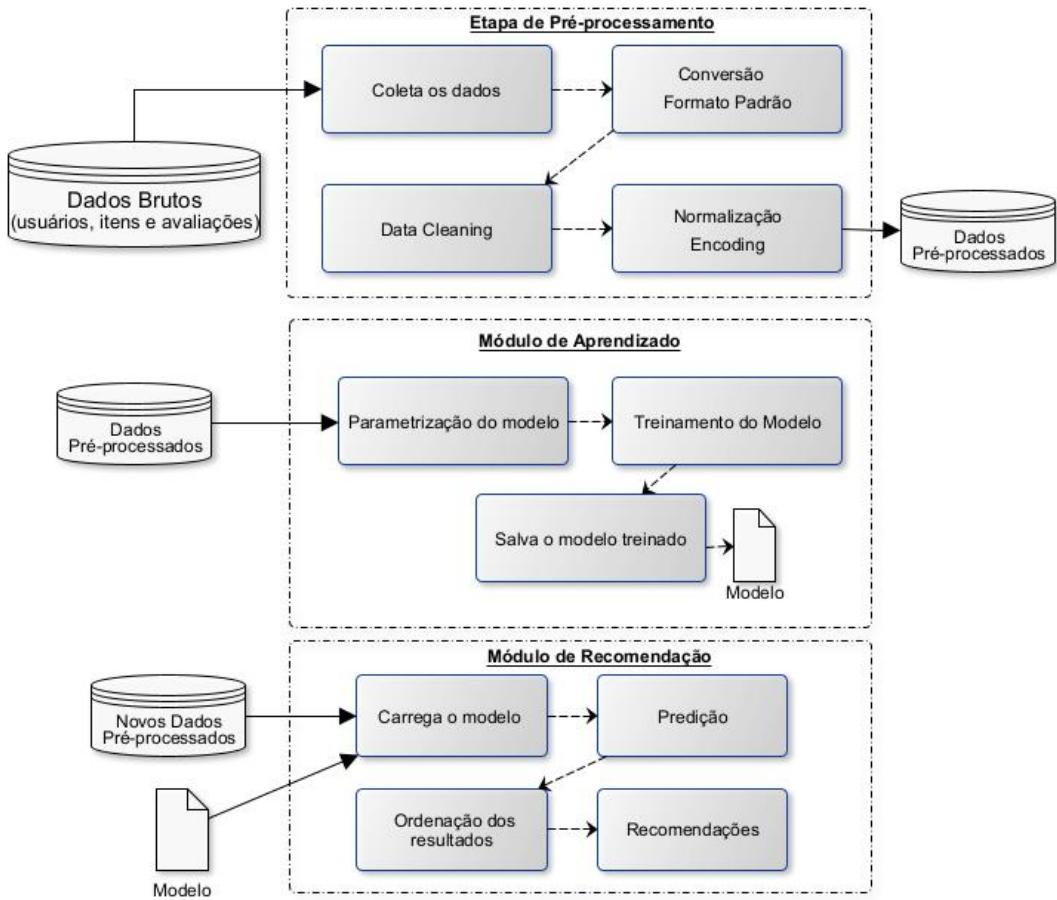
Fonte: Elaboração própria (2024)

A Figura 3.1, representa os dois módulos supracitados que caracterizam o modelo híbrido proposto e como são organizados a fim de que juntos gerem as recomendações.

### 3.3 MÓDULO CONTENT-BASED DEEP LEARNING

Este módulo possui a arquitetura baseada no sistema de recomendação desenvolvido por Dellal-Hedjazi e Alimazighi (2020) e possui dois conceitos primordiais que são a abordagem demográfica e *content-based*. Isto é, considera tanto os atributos que descrevem e qualificam os itens quanto também os que são referentes aos usuários, a fim de serem a base para o treinamento e geração das recomendações.

Figura 3.2: Arquitetura Sistema de Recomendação *Content-based*



Fonte: Adaptado de Dellal-Hedjazi e Alimazighi (2020)

A Figura 3.2 representa as etapas que são executadas para a obtenção da recomendação. As etapas que compõem este módulo são descritas a seguir:

### 3.3.1 Carregamento e Recepção dos dados:

Etapa de coleta dos dados que serão utilizados no processo de treinamento, testes e validação do modelo. Com o intuito de padronizar os dados admitidos como entrada elaborou-se um dicionário de dados, conforme as Tabelas 3.1, 3.2 e 3.3.

Tabela 3.1: Modelo de Arquivo de Usuário

<b>Dados de Usuários</b>		
Nome do arquivo: users.txt		
<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>
user_id	Inteiro	Identificador único do usuário
age	Inteiro	Idade do usuário
gender	Texto (M F)	Gênero do usuário
occupation	Texto	Ocupação do usuário
zip_code	Inteiro	Código postal

Fonte: Elaborado pelo autor, 2025.

Tabela 3.2: Modelo de Arquivo de Filmes

<b>Dados de Filmes</b>		
Nome do arquivo: movies.txt		
<b>Campo</b>	<b>Tipo</b>	<b>Descrição</b>
movie_id	Inteiro	Identificador único do filme
movie_title	Texto	Título do filme
release_date	Data	Data de lançamento do filme
genres	Texto	<b>Gêneros do filme.</b> Possíveis valores: Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western. Os valores devem ser registrados utilizando o número 1 para indicar a presença do gênero e 0 para ausência. Logo, cada filme terá uma sequência conforme o exemplo: 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 . Sendo que cada posição refere-se a um gênero. É necessário informar na mesma ordem de definição dos gêneros.

Fonte: Elaborado pelo autor, 2025.

Uma vez que cada conjunto de dados disponível possuem formatos e conteúdos distintos, é fundamental lançar mão de uma estratégia de padronização da estrutura de dados de entrada. Logo, ao utilizar os padrões previamente descritos torna-se possível a utilização de diferentes massas de teste, desde que as mesmas sejam convertidas e

respeitem as regras definidas para o modelo de dados proposto.

Tabela 3.3: Modelo de Arquivo de Avaliações

Dados de Avaliações		
Nome do arquivo: ratings.txt		
Campo	Tipo	Descrição
user_id	Inteiro	Identificador único do usuário
item_id	Inteiro	Identificador único do filme
rating	Inteiro	Nota de avaliação do filme. Possíveis valores: [1, 5]
timestamp	Inteiro	Tempo em segundos desde 01/01/1970 UTC

Fonte: Elaborado pelo autor (2025).

### 3.3.2 Pré-Processamento

A principal função é fazer um processamento prévio nos dados já coletados pela etapa anterior, com o objetivo de transformar os dados brutos de avaliações, itens e usuários, em representações úteis e bem definidas para o modelo de aprendizado computacional. Primeiramente, é construído o mapeamento dos identificadores de usuários e itens (filmes) para indexadores numéricos sequenciais, pois o conjunto de dados de entrada não garante esta ordenação. Esta estratégia utiliza *hashmaps* para a criação de índices eficientes, os quais reduzem a dimensionalidade do problema e otimizam o acesso aos dados. Desse modo, os identificadores são substituídos por índices mapeados, o que diminui a complexidade computacional do processo de busca para  $O(1)$ , isto é, complexidade constante, que indica a não ocorrência de variação quando há mudança no tamanho do conjunto de dados de entrada.

Em relação às notas de avaliações dos itens pelos usuários, é aplicada a normalização denominada *Min-Max Scaling* para reduzir o intervalo de dados que originalmente é entre 1 e 5 para valores entre 0 e 1. Esta operação é realizada com o intuito de facilitar o processo de aprendizagem da rede neural, pois garante que o treinamento ocorra com maior estabilidade além de reduzir o impacto de valores em escalas subjetivas de

avaliações.

O processo de limpeza de dados é aplicado para a remoção de dados discrepantes (*outliers*) ou desestruturados, e preenchimento de dados ausentes. Essa etapa de *data cleaning* visa evitar que dados nulos causem impacto no processo de treinamento do modelo. Ademais, para extrair alguns dados importantes de conjuntos textuais são utilizadas expressões regulares, por exemplo, para a extração apenas do ano de lançamento do filme.

Há a codificação de dados, etapa em que é feita a conversão de dados do formato original para o formato digital, a fim de que possam ser utilizados nos cálculos matemáticos. Esse procedimento busca identificar dados não numéricos e convertê-los para uma representação ideal, desse modo, os dados de gênero do usuário que podem ser ‘M’ ou ‘F’, são convertidos respectivamente para ‘0’ e ‘1’, e, de modo, semelhante é aplicado para os dados de ocupação do usuário e gênero do filme.

Os títulos dos filmes são processados com a estratégia *One-Hot Encoding* que é uma técnica de representação categórica que permite que cada item seja representado por um vetor numérico binário. Embora esta abordagem aumente a dimensionalidade do problema, é capaz de permitir que o modelo aprenda padrões com base nos títulos descritos por caracteres alfanuméricos.

### 3.3.3 Módulo de Aprendizado

Umas das arquiteturas mais indicadas para problemas de classificação baseado nos atributos e características dos dados de entrada é denominada *Multilayer Perceptron* – MLP, a qual é uma rede neural profunda. Neste caso, foi dimensionada para ser composta por uma camada de entrada, doze camadas ocultas e uma camada de saída.

A camada de entrada será alimentada com as características dos itens e usuários, por exemplo, com a utilização da base de dados MovieLens estão disponíveis os atributos de título, ano de lançamento e gênero, como também os atributos dos usuários,

por exemplo, idade, gênero e ocupação. Em relação à parte intermediária é composta por doze camadas ocultas, às quais são a essência do módulo e são responsáveis pelo processamento e treinamento das redes profundas. Estas camadas internas podem ser redimensionadas para adequação da rede em relação aos resultados obtidos, ou mesmo para suportar diferentes quantidades de atributos disponíveis para os conjuntos de dados em avaliação, e assim permitir a sua otimização.

Em observação, a escolha do número de camadas ocultas é um ponto relevante de decisão do algoritmo, uma vez que pode gerar dois tipos de problemas chamados *overfitting* e *missfitting*, sendo que o primeiro ocorre quando o número de camadas ocultas é alto e isso gera uma adequação muito grande em relação aos dados de entrada, ou seja, o modelo na prática torna-se especializado e não é capaz de generalizar para novos dados avaliados (LAMPA; GOMES; ZAFALON, 2024; SABIRI.; EL ASRI.; RHA-NOUI., 2022). Já, o segundo problema está relacionado à incapacidade do modelo aderir ao comportamento dos dados de entrada e torna-se demasiadamente genérico.

Em resumo, esta camada intermediária é composta por sub-redes do tipo *Multi-layer Perceptron* - MLP, com a função de ativação *Rectified Linear Unit* – ReLU, as quais são responsáveis por utilizar as características dos usuários e itens em suas camadas de aprendizado em profundidade. A Tabela 3.4 descreve cada uma das sub-redes e sua implementação detalhada:

Tabela 3.4: Descrição Sub-redes do Módulo *Content-based*

Sub-rede	Características	Camadas
01	Idade e gênero do usuário	Três camadas conectadas
02	Ocupação do usuário	Uma camada linear
03	Título do Filme	Duas camadas densas para reduzir a dimensionalidade
04	Ano de Lançamento do filme	Três camadas conectadas
05	Gênero do Filme	Uma camada linear

Fonte: Elaborado pelo autor (2025).

O principal objetivo de utilizar cinco subredes específicas, é que cada uma delas torna-se especializada em determinadas características dos filmes e dos usuários. Assim, evita-se o excesso de informações iniciais que poderiam sobrecarregar o processo inicial de aprendizagem. Todavia, após esse etapa inicial, é necessário que os modelos especializados sejam concatenados em uma saída única, e essa operação é feita por uma sexta sub-rede que também utiliza MLP e ReLU, que gera um vetor único como resultado de todas as informações extraídas das demais sub-redes especialistas.

Em observação, durante o processo de implementação foram considerados testes empíricos para analisar o uso de menor quantidade de características consideradas nas subredes de aprendizado. Todavia, os resultados aferidos nestes cenários revelaram menor qualidade das recomendações e maior taxa de erros. Portanto, o módulo construído utiliza todas estas características citadas anteriormente e cada uma das subredes especializadas, pois empiricamente, provaram produzir melhores resultados.

Essa abordagem possibilita que o modelo se especialize individualmente em cada tipo de característica de interesse. Por outro lado, é capaz de combinar a representação dos resultados individuais, com o intuito de obter relações complexas entre as características de usuários e itens. Logo, a partir desta fundamentação, o resultado do treinamento do método é a capacidade de produzir recomendações de alta relevância.

O processo de aprendizado é do tipo supervisionado, visto que se utiliza de dados rotulados, os quais permitem aferir o resultado da predição com o resultado real esperado. Deste modo, o algoritmo trabalha com uma iteração principal, que considera os dados de cada usuário para cada filme e então realiza a predição, que é verificada com o resultado rotulado. A cada iteração o erro é calculado através de uma função de perda denominada erro quadrático médio e descrita pela Equação 3.1, onde  $y_i$  são os valores reais (rótulos),  $\hat{y}_i$  são as previsões do modelo e  $n$  é o número de exemplos:

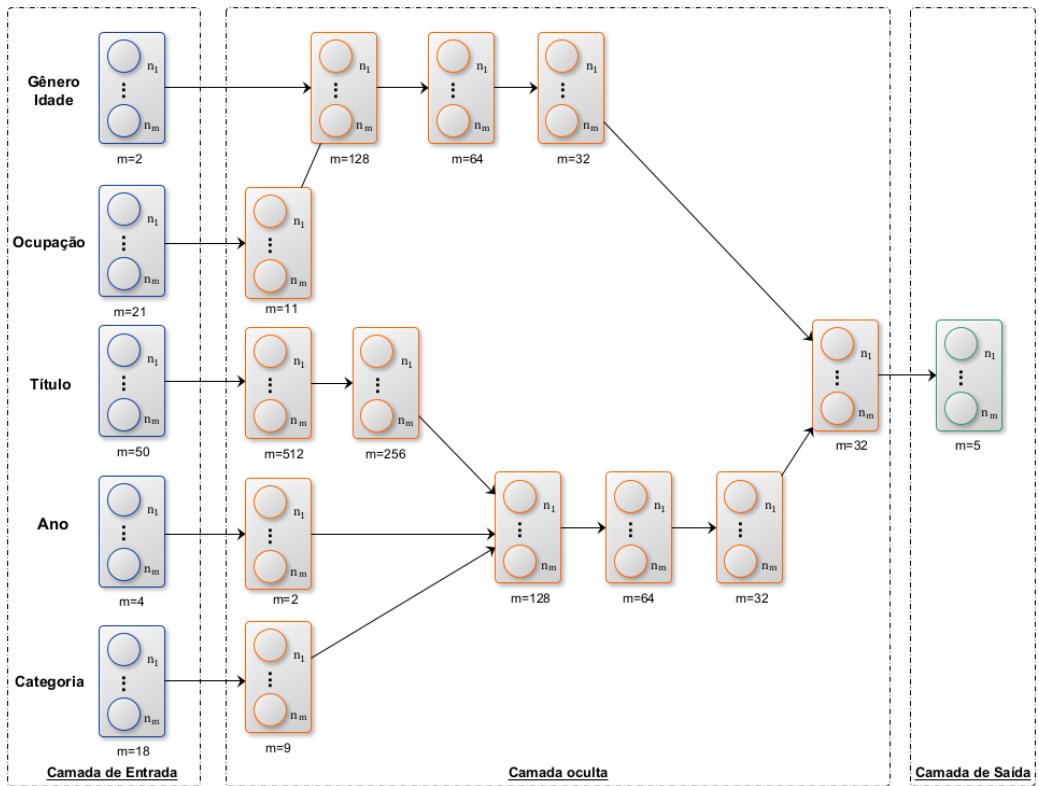
$$\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3.1)$$

Este modelo de treinamento que considera os dados de entrada, os processa em suas camadas, gera uma saída final e calcula o erro, é denominado de propagação à frente (*forward pass*). Ademais, o treinamento também utiliza o conceito de retropropagação de erro (*backpropagation*), o qual refere-se à estratégia de minimização do erro a cada iteração, através do cálculo automático dos pesos da rede, sendo que este cálculo é feito de trás para frente, a cada camada da rede.

O processo de retropropagação utiliza o otimizador *Adaptive Moment Estimation* – Adam, que auxilia no ajuste dos pesos da rede para mitigar os erros. São definidos os parâmetros de taxa de aprendizado com o valor de  $10^{-3}$ , que auxilia na estabilidade do modelo e evita oscilações, como também o fator de decaimento do peso em  $1e^{-4}$ , para evitar *overfitting*.

Além disso, possui um número variável de neurônios, agrupados funcionalmente e totalmente conectados, conforme a Figura 3.3. Basicamente, o processo de aprendizado inicia com os dados pré-processados, dos quais 80% será para treinamento, 10% validação e 10% testes, assim, a aprendizagem ocorre ao passar os dados pela rede neural profunda e calculando-se o erro entre o resultado esperado e o resultado encontrado. Desse modo, a cada iteração o erro é minimizado por meio da retropropagação e os pesos para cada matriz de parâmetros são salvos.

Figura 3.3: Modelo Arquitetural MLP



Fonte: Adaptado de Dellal-Hedjazi e Alimazighi (2020)

### 3.3.4 Recomendação

Por fim, a camada de saída é a responsável por gerar o resultado da recomendação, isto é, o neurônio com a melhor probabilidade é considerado a predição da rede. Etapa que considera o modelo previamente treinado, desse modo, faz o carregamento dos pesos salvos e utiliza-os sob os novos dados pré-processados. O resultado final obtido neste módulo é o próprio modelo treinado, como também uma lista de predileções. O modelo treinado é salvo em um arquivo no formato “.pth” para que possa ficar armazenado no sistema de arquivos e ser reutilizado futuramente em novas gerações de recomendação, ou seja, evita que o modelo tenha que ser treinado todas as vezes que precisar utilizá-lo. Ademais a lista de recomendações é exibida no console onde está sendo executado, conforme a Figura 3.4.

Figura 3.4: Recomendações do módulo *content-based*

Dados do usuário usado para Recomendação			
ID	Idade	Gênero	Ocupação
123	18	F	Estudante
Filmes Recomendados:			
Ordem	Filme ID	Título do Filme	Pontuação
9	00001.	ID: 00174   Casablanca (1942)	Score: 0.8727
10	00002.	ID: 00005   Dr. Strangelove or: How I Learned to Stop Worrying and	Score: 0.8383
11	00003.	ID: 00200   Schindler's List (1993)	Score: 0.8341
12	00004.	ID: 00036   Rear Window (1954)	Score: 0.8330
13	00005.	ID: 00189   Godfather, The (1972)	Score: 0.8278
14	00006.	ID: 00101   Raiders of the Lost Ark (1981)	Score: 0.8257
15	00007.	ID: 00239   Shawshank Redemption, The (1994)	Score: 0.8231
16	00008.	ID: 00260   Citizen Kane (1941)	Score: 0.8169
17	00009.	ID: 00297   Usual Suspects, The (1995)	Score: 0.8168
18	00010.	ID: 00277   Wrong Trousers, The (1993)	Score: 0.8160
19	00011.	ID: 00357   Star Wars (1977)	Score: 0.8091

Fonte: Elaborado pelo autor (2025)

### 3.4 MÓDULO COLLABORATIVE FILTERING DEEP LEARNING

O objetivo principal deste módulo é possibilitar a extração de predileções relevantes que são intrínsecas ao histórico de relações de usuários e itens. Conforme pesquisas da revisão literária, há um modelo que se destaca por sua capacidade extrair a relação entre usuários e itens através de múltiplas perspectivas e que possui eficiência comprovada na obtenção de resultados com maior performance e qualidade. Deste modo, este modelo proposto por Huang et al (2019) é denominado *Deep Matrix Factorization Learning* – DMFL, e é empregado como base de referência para que seja adaptado e acoplado ao algoritmo híbrido proposto neste trabalho.

Os dados de entrada para este módulo são exatamente os mesmos descritos na Subseção 3.3.1. Isto é, os mesmos dicionários de dados para usuários, itens e avaliações, como também os métodos de importação dos dados são compartilhados entre os dois módulos.

A estrutura básica de funcionamento é segmentada em duas etapas, a saber, aprendizado de características e geração de preferências, as quais são descritas nas subseções 3.4.1 e 3.4.2 a seguir.

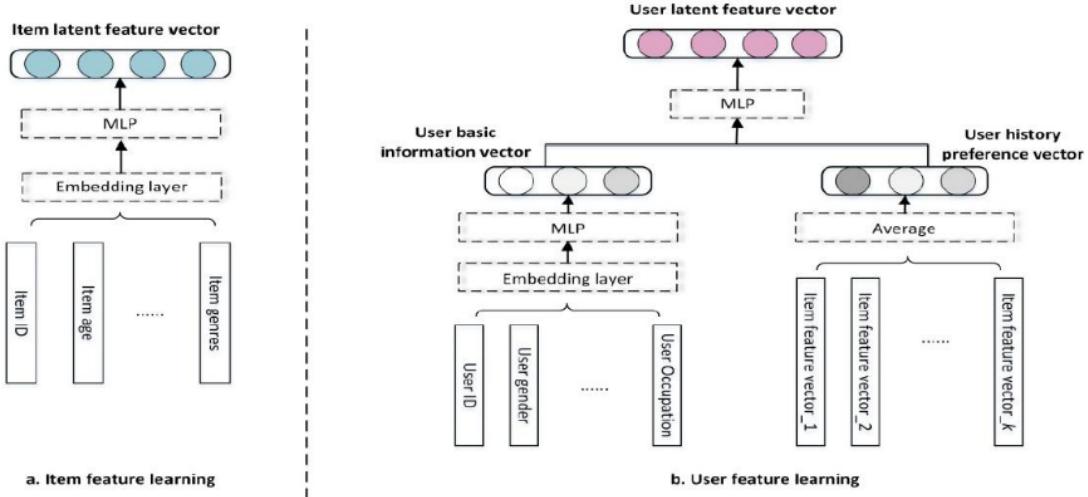
### 3.4.1 Aprendizado de características de usuários e itens

Composto por duas redes neurais paralelas às quais são responsáveis por extrair os vetores de características latentes dos itens e também dos usuários. Neste contexto são utilizadas estruturas *Multilayer Perceptron* – MLP com o objetivo de extrair características profundas e ocultas referentes aos usuários e itens (LAMPA; GOMES; ZAFALON, 2024).

Em razão das características dos itens que serão recomendados serem consideradas estáveis por não mudarem com frequência, é estabelecido um método estático para o treinamento e extração das mesmas. Logo, os dados do vetor latente de características dos itens são extraídos diretamente dos dados estáticos descritivos dos mesmos, conforme Figura 3.5 no item (a). São consideradas as características: título, ano e gênero.

Por outro lado, os dados referentes às características dos usuários podem mudar com maior frequência de acordo com as mudanças de preferências do mesmo. Desse modo, o método representado na Figura 3.5 no item (b) é um processo de aprendizado dinâmico o qual considera tanto os dados de características descritivas, por exemplo, idade, gênero, ocupação e região, como também utiliza o histórico de interações de cada usuário com os itens. Através deste histórico são extraídos os vetores de características dos  $k$  itens que o usuário mais “gostou” recentemente e assim permite considerar a dinâmica natural de evoluções das preferências dos usuários.

Figura 3.5: Métodos para obter vetor latente de usuários e itens



Fonte: HUANG et al, 2019

O vetor de histórico de preferência do usuário é definido através da formulação 3.2, cujo parâmetro que  $k$  representa a quantidade de itens que o usuário gostou recentemente e  $y_t^i$  é o item do histórico de preferência do usuário  $u_i$ :

$$x_i^h = \frac{1}{k} \sum_{t=1}^k y_t^i \quad (3.2)$$

Ademais, com o intuito de obter o vetor final de características do usuário, os vetores  $x_i^c$  e  $x_i^h$ , que se referem respectivamente ao vetor de dados básicos e estáticos, e ao vetor de histórico de preferência, são inseridos na camada de entrada de um *Multilayer Perceptron*. Esse vetor final é descrito pela Equação 3.3:

$$x_i = f \left( W_\alpha \begin{bmatrix} x_i^c : x_i^h \end{bmatrix} + b_\alpha \right) \quad (3.3)$$

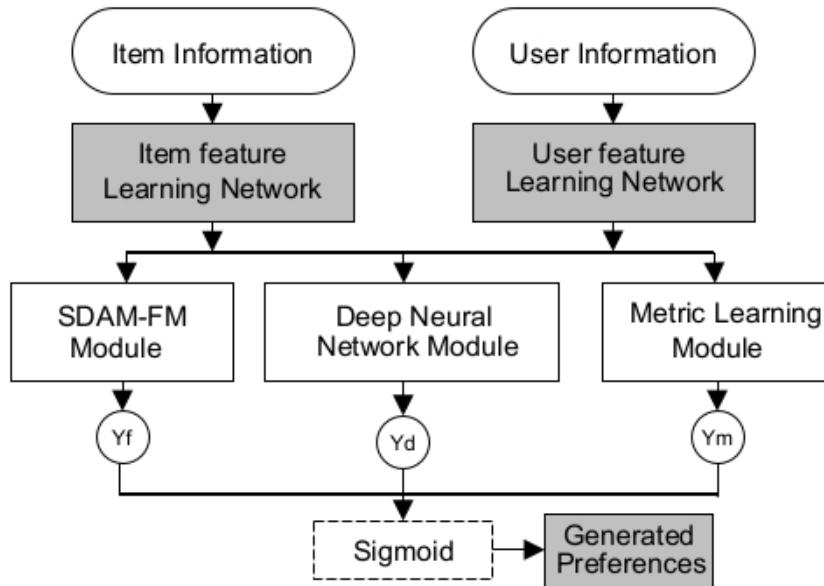
Onde  $f$  é a função de ativação da camada da rede neural,  $W_\alpha$  representa os pesos e  $b_\alpha$  é o *bias* da camada da rede.

### 3.4.2 Aprendizado do histórico de relações

O resultado da etapa descrita na Subseção 3.4.1, que são os vetores de características dos itens e usuários, é utilizado como entrada para o algoritmo responsável por gerar as predileções dos usuários. Sendo que este processo, baseia-se no conceito de combinar simultaneamente vários modelos, a fim de que seja possível extrair características dos dados a partir de múltiplas perspectivas. Deste modo, além de ser capaz de unir as vantagens destes diferentes modelos, também minimiza suas deficiências, logo possibilita prover resultados com maior precisão.

Nesse contexto, há três submódulos que são executados paralelamente e seus resultados são unidos para obter as preferências do usuário. Os módulos são denominados **SDAE-FM**, **Deep Neural Network – DNN**, **Metric Learning** conforme a Figura 3.6.

Figura 3.6: Geração de Preferência DMFL



Fonte: Extraído de Huang et al. (2019)

Os submódulos do processo de geração de preferência são detalhados a seguir:

- **SDAE-FM**: baseado em duas estratégias, Fatoração de Matriz – FM e *Stacked Denoising AutoEncoder* – SDAE, que juntas são responsáveis pela redução

da dimensão das características e extração de características latentes profundas. Desse modo, é utilizado para o aprendizado da importância de cada característica, bem como o relacionamento que ocorre entre as mesmas e, por consequência, obtém as preferências do usuário.

- **Deep Neural Network - DNN:** o objetivo principal deste módulo é explorar as relações profundas e não lineares provenientes da relação entre itens e usuários, e, por fim, gerar as preferências de usuário com maior precisão. Considera uma rede neural com quatro camadas, sendo o cálculo em cada uma delas executado de acordo com a Equação 3.4:

$$a^{(l+1)} = f \left( W^l a^l + b^l \right) \quad (3.4)$$

Onde  $a$  representa a entrada,  $\mathbf{W}$  são os pesos,  $\mathbf{b}$  é o *bias*, sendo todos os parâmetros referentes à  $l$ -ésima camada. A função  $f$  corresponde à função de ativação, que neste caso foi empregado o método *Rectified Linear Units – ReLUs*.

Em observação, nesta etapa é considerado o uso da estratégia de Normalização *Batch* com o intuito de padronizar as entradas em cada camada da rede neural. Além disso, conforme Lampa, Gomes e Zafalon (2024), este procedimento é fundamental para prevenir a condição de *overfitting*, ou seja, evitar que o modelo se torne específico ao conjunto de dados de treinamento e não seja capaz de generalizar para novos casos.

- **Metric Learning:** nesta etapa o objetivo principal é considerar as relações de usuários e itens, e também medi-las por uma perspectiva de distância. Utiliza dois modelos de redes neurais convolucionais paralelas com diferentes parâmetros. O processo computacional destas redes é representado na Equação 3.5 a seguir:

$$h_t = f(x_i * k_t + b_t) \quad (3.5)$$

Onde  $x_i$  refere-se ao dado de entrada,  $k_t$  representa o  $t$ -ésimo filtro,  $b_t$  é o *bias*,  $f$  é a função de ativação **ReLU**s e, por fim, o símbolo  $*$  é a operação de convolução.

Em síntese, cada um dos submódulos visa obter as preferências de usuário por meio de perspectivas específicas: linearidade, não linearidade e distância. Desse modo, ao final da execução das três etapas há três vetores de preferência do usuário que serão processados através da operação sigmoide. Nesse contexto, ao executar esta operação o objetivo principal é obter a combinação dos resultados de modo a efetivamente otimizar a precisão da recomendação. Esta operação é definida pela Equação 3.6, em que os termos referem-se a cada um dos vetores previamente obtidos.

$$z_{ij} = \text{sigmoid}\left(y_{ij}^f + y_{ij}^d + y_{ij}^m\right) \quad (3.6)$$

### 3.4.3 Geração das recomendações

Assim como ocorre no módulo *content-based*, este também possui um processo iterativo sob cada usuário e item, em que utiliza um conjunto de dados de entrada referente a usuários, itens e avaliações, processam o treinamento do modelo e suas sub-redes, calcula o erro e assim sucessivamente, a cada iteração conforme o pseudocódigo descrito na Figura 3.7. Desse modo, utiliza a estratégia de propagação à frente e também retropropagação do erro, a fim de mitigar o erro em cada camada da rede.

Figura 3.7: Pseudocódigo do módulo *Collaborative-Filtering*

```
1 Iniciar modelo CollaborativeFilteringDeepLearning com:
2     - Dois MLPs para extração de características de usuários e itens
3     - Um SDAE-FM (AutoEncoder + Factorização de Matrizes)
4     - Um DNN (com LayerNorm)
5     - Um bloco convolucional para Metric Learning
6
7 Definir função treinar_modelo:
8     - Para cada época do treinamento:
9         - Inicializar `total_loss`
10        - Para cada batch de dados:
11            - Extrair características dos usuários e itens
12            - Passar pelo modelo
13            - Calcular a perda
14            - Retropropagar o erro
15            - Atualizar os pesos do modelo
16            - Acumular a perda total
17        - Calcular perda média da época
18        - Ajustar taxa de aprendizado
19
20 No `forward` do modelo:
21     - Extrair características latentes de usuários e itens
22     - Concatenar os vetores latentes
23     - Passar pelos três submodelos:
24         - SDAE-FM
25         - DNN
26         - CNN (Metric Learning)
27     - Unificar as saídas usando uma função sigmoid
28     - Retornar a predição final
```

Fonte: Elaborado pelo autor (2025)

O otimizador Adam também é utilizado para auxiliar na retropropagação, com o intuito de adequar o gradiente de pesos de modo mais eficiente. É utilizado o fator de taxa de aprendizado com o valor de  $10^{-3}$ , para garantir mais estabilidade no treinamento.

A etapa de treinamento deste modelo também se dá por meio supervisionado, pois lança mão de valores rotulados para a comparação entre a predição realizada e os valores reais. Neste contexto, apesar de serem utilizados modelos *Stacked Denoising AutoEncoder* e redes neurais convolucionais, que são capazes de aprender apenas por meio da obtenção da relação entre os itens e usuários, esses atualmente estão inseridos no mesmo critério de avaliação do restante das sub-redes que são juntamente treinadas e, portanto utilizam os mesmos dados rotulados para calcular o erro na etapa de retropropagação.

Em suma, o resultado obtido ao final da execução deste módulo é o treinamento do modelo *collaborative filtering* e também uma lista de recomendações. O resultado

do modelo treinado é salvo no sistema de arquivos com a extensão “.pth”, com a finalidade de que fique disponível para ser recarregado em utilizações futuras, isto é, sem a necessidade de treinar o modelo todas as vezes que for gerar recomendações. Além disso, os resultados das recomendações para o usuário são exibidas no console de execução, conforme a Figura 3.8.

Figura 3.8: Recomendações do módulo *Collaborative-Filtering*

Dados do usuário usado para Recomendação				
	ID	Idade	Gênero	Ocupação
3	171	36	M	Engenheiro
Filmes Recomendados:				
Ordem	Filme ID	Título do Filme	Pontuação	
9	00001.	ID: 00239   Shawshank Redemption, The (1994)	Score: 0.7868	
10	00002.	ID: 00189   Godfather, The (1972)	Score: 0.7834	
11	00003.	ID: 00174   Casablanca (1942)	Score: 0.7595	
12	00004.	ID: 00153   Face/Off (1997)	Score: 0.7583	
13	00005.	ID: 00216   Pulp Fiction (1994)	Score: 0.7563	
14	00006.	ID: 00357   Star Wars (1977)	Score: 0.7516	
15	00007.	ID: 00031   Silence of the Lambs, The (1991)	Score: 0.7464	
16	00008.	ID: 00049   Fargo (1996)	Score: 0.7436	
17	00009.	ID: 00102   Terminator 2: Judgment Day (1991)	Score: 0.7436	
18	00010.	ID: 00221   Fish Called Wanda, A (1988)	Score: 0.7351	
19	00011.	ID: 00118   2001: A Space Odyssey (1968)	Score: 0.7349	
20	00012.	ID: 00297   Usual Suspects, The (1995)	Score: 0.7340	

Fonte: Elaborado pelo autor (2025)

### 3.5 OTIMIZADOR ADAM

Conforme descrito nas seções anteriores, tanto o módulo *content-based deep learning* quanto o *collaborative filtering deep learning* utilizam uma estratégia de apoio ao mecanismo de retropropagação no treinamento dos modelos, denominada *Adaptive Moment Estimation – Adam*. Desse modo, o objetivo desta seção é conceituar este otimizador e justificar o seu uso no sistema de recomendação desenvolvido.

Neste contexto, a busca contínua por melhores resultados e precisão nos mecanismos dos sistemas de recomendação é um fator que motiva diversas pesquisas e inovações nesta área. Nesse sentido, na literatura científica a estratégia denominada otimizador Adam, é definida como um método estocástico de otimização e que pode ser utilizado nos modelos de aprendizado e treinamento, e por consequência na me-

lhora dos resultados obtidos sob diversos aspectos. (KINGMA; BA, 2017)

Esta estratégia já é utilizada com maior frequência do âmbito do aprendizado de máquina, porém ainda pouco explorada no contexto específico de sistemas de recomendação baseados em *deep learning*. Uma das aplicações, no âmbito de redes neurais profundas é descrita por Jais, Ismail e Nisa (2019), em que através de experimentações aplicou o algoritmo de otimização Adam para melhora de performance de redes neurais amplas e profundas, e por conseguinte obteve resultados que geraram ganho de precisão, redução dos parâmetros de perda e melhorias na predição, quando comparado com os resultados tradicionais sem o uso do otimizador.

Nesse sentido, este trabalho também considera a utilização do otimizador Adam como uma estratégia adicional de melhoria de performance na qualidade dos resultados, portanto é aplicado aos modelos de treinamento do sistema de recomendação híbrido proposto. Assim, esta operação de otimização será inserida em cada uma das redes neurais profundas utilizadas no algoritmo proposto.

### 3.6 UNIFICAÇÃO DOS RESULTADOS

Conforme descrito nas seções prévias o algoritmo do sistema de recomendação proposto baseia-se em dois módulos principais que são processados separadamente e visam ao final o mesmo objetivo, o qual é a produção de recomendações. Em observação, cada um dos módulos considera uma estratégia específica, sendo o primeiro com a abordagem *content-based* e o segundo baseado em *collaborative filtering*, ambos apoiados por algoritmos de *deep learning*.

Neste contexto, os dois modelos gerados ao final das etapas de treinamento, possuem qualidades e funcionalidades distintas, às quais são ainda mais úteis quando combinadas. Logo, foram definidas algumas possíveis abordagens de unificação dos modelos e implementadas como produto final do algoritmo que irá gerar as recomendações de fato. Em observação, as diversas estratégias de unificação propostas são

avaliadas e os resultados são apresentados em detalhes no Capítulo 4.

As estratégias de unificações são descritas nas subseções a seguir.

### 3.6.1 Abordagem Híbrida de Soma Ponderada

A combinação entre os dois modelos de recomendação é realizada através da formulação 3.7, em que  $\alpha$  é o peso atribuído ao modelo *content-based* e  $(1 - \alpha)$  é o peso do modelo *collaborative filtering*.

$$\text{Predição} = \alpha \times P_{\text{content-based}} + (1 - \alpha) \times P_{\text{collaborative-filtering}} \quad (3.7)$$

O valor de  $\alpha$  pode ser definido de modo fixo, ou também ser cálculo de modo dinâmico, e nesse sentido, foi implementada uma função que avalia os modelos individualmente de acordo com alguns dos critérios: precisão, *recall*, *f1-score*, acurácia, RMSE e MAE, e então de acordo com a melhor relevância de cada modelo, o peso de  $\alpha$  é atribuído dinamicamente.

### 3.6.2 Pesos Dinâmicos

Estabelece um modelo híbrido que combina os resultados de ambos os modelos, através da combinação dinâmica de pesos de acordo com a confiabilidade da previsão para o usuário e item. Utiliza o cálculo de erro absoluto de cada modelo individualmente e segue a seguinte regra, se o modelo possui menor erro então recebe um peso maior, caso contrário um peso menor. Na prática, é capaz de ajustar dinamicamente os pesos de acordo com o nível de confiabilidade individual de cada modelo.

### 3.6.3 Fusão Bayesiana

Processo que combina as predições dos dois modelos de acordo com o critério de confiança e probabilidade de cada um. Desse modo, primeiramente é realizada a conversão das previsões em probabilidades que estejam no intervalo entre 0 e 1. Essa

etapa é feita com a aplicação da função sigmoide, onde  $p$  é a probabilidade e *predição* é o valor da predição do item, conforme Equação 3.8.

$$p = \frac{1}{1 + e^{-\text{predição}}} \quad (3.8)$$

A predição final da fusão bayesiana combina os dois modelos com a fórmula 3.9, onde  $p_{content}$  é a probabilidade da predição do módulo *content-based* e  $p_{collab}$  é a probabilidade do *collaborative filtering*.

$$p = \frac{p_{content} \times p_{collab}}{(p_{content} \times p_{collab}) + ((1 - p_{content}) \times (1 - p_{collab}))} \quad (3.9)$$

Esta estratégia possui a capacidade de avaliar a ocorrência de baixa confiança nos modelos individuais, e então reduzir a influência do mesmo na previsão final realizada. Portanto, este método é capaz de considerar a incerteza dos modelos e gerar uma previsão mais confiável.

### 3.6.4 Fusão Sigmoide

Nesta abordagem as predições individuais de cada modelo são convertidas segundo a função sigmoide, sendo esta utilizada para a realização do processo de normalização dos dados obtidos pelos diferentes modelos. Assim, as predições que originalmente estão em escalas distintas, tornam-se valores entre 0 e 1, o que implica em um melhor balanceamento dos valores a serem utilizados na fusão do modelo híbrido.

Na sequência, os valores já normalizados pela função sigmoide são combinados através do cálculo da média aritmética. Essa operação indica que ambos os modelos possuem pesos igualitários na recomendação final, ou seja, ambos os modelos contribuem igualmente para a recomendação final, conforme descrito na Equação 3.10.

$$p = \frac{\text{content sigmoide} + \text{collaborative sigmoide}}{2} \quad (3.10)$$

Ademais, o valor obtido pela média calculada corresponde à probabilidade do usuário gostar de um item. Desse modo, para considerar se o item será ou não recomendado ao usuário, é aplicado um valor de corte (*threshold*) que foi fixado em 0,6 e, então a recomendação final é obtida. A ideia é que através desse limiar seja feita uma comparação e, desde que a probabilidade obtida seja maior que o ponto de corte, então é efetivada a recomendação do item.

Conforme Fang et al. (2020), a escolha do ponto de corte é um desafio pois se um valor baixo for escolhido resultará em um alto número de predições de pouca relevância, por outro lado, um valor alto irá restringir muito os resultados. Desse modo, a escolha do parâmetro com valor de 0,6 foi resultado do processo empírico executado ao longo do desenvolvimento, e possui a finalidade de evitar recomendações irrelevantes, bem como admitir que não apenas itens de alto grau de relevância sejam recomendados, ou seja, prover diversidade.

### 3.6.5 *Random Forest*

Este algoritmo primeiramente, realiza o treinamento de um regressor *random forest* sob as predições dos modelos *content-based* e *collaborative filtering*, a fim de aprender a melhor forma de combinar os resultados dinamicamente. Basicamente, o algoritmo aprende de modo supervisionado, ao iterar sob cada usuário, item e sua respectiva avaliação real, e então realiza seu aprendizado a respeito de qual é a melhor recomendação para cada item.

É uma abordagem que evita o uso de métodos estáticos para a escolha entre as predileções dos modelos, que a torna mais fiel na análise, todavia, demanda o tempo de seu treinamento, para que seja capaz de tomar a melhor decisão. A estratégia *random forest* necessita de um grande conjunto de dados para que seja capaz de realizar um treinamento satisfatório e aprenda a fazer boas combinações considerando cada um dos modelos.

### 3.6.6 Fusão *XgBoost*

A ideia é basicamente utilizar uma técnica para aprender as melhores previsões dos modelos *content-based* e *collaborative-filtering*, de modo a criar um modelo híbrido que faça recomendações mais eficientes. Desse modo, é utilizado um regressor *XGBoost*, que é responsável por utilizar árvores de decisão para aprender a melhor maneira de fazer a combinação das previsões de cada modelo.

O método de regressão utiliza a função de erro quadrático para que dinamicamente faça o ajuste dos pesos da árvore de decisão. O objetivo desse cálculo é reduzir a diferença entre o valor da previsão realizada com o valor rotulado, isto é, possibilitar a melhoria da previsão a cada iteração do treino. Além disso, o mecanismo de treinamento também é do tipo supervisionado e depende de um volume de dados suficiente para aumentar a capacidade de aprendizado.

### 3.6.7 *Stacking Meta-Learning*

Realiza a combinação dos resultados de cada modelo através de uma rede de aprendizado, de modo que a cada iteração de usuário, item e avaliação, o modelo é capaz de aprender a melhor previsão. O modelo resultado é chamado de meta-modelo, e o processo de treinamento caracteriza-se por ser do tipo supervisionado, além disso, utiliza a função de erro quadrático médio para avaliar a perda.

## 3.7 IMPLEMENTAÇÃO DO ALGORITMO DE RECOMENDAÇÃO HÍBRIDO

A implementação do algoritmo proposto neste trabalho foi realizada com a utilização da linguagem de programação Python na versão 3.12.4. Além disso, com o intuito de desenvolver os modelos com as melhores práticas e abordagens, como também obter celeridade, foram utilizadas algumas bibliotecas de código fonte aberto conforme a Tabela 3.5.

O conjunto de dados utilizado durante a etapa de treinamento dos modelos denomina-se MovieLens (RAWAT; TOMAR; KUMAR, 2020; HARPER; KONSTAN, 2015) e é provido pela Universidade de Minnesota. Estes dados são disponibilizados gratuitamente para finalidade de pesquisas científicas e público em geral que necessite de massas de dados para testes. Ademais, os dados estão organizados em cinco principais categorias, as quais se referem à quantidade de avaliações disponíveis nos mesmos.

Tabela 3.5: Bibliotecas de apoio

Biblioteca	Finalidade
<b>os</b>	Interface com o sistema operacional para manipulação de arquivos e diretórios, entre outras funções básicas.
<b>Pytorch</b>	Biblioteca de <i>deep learning</i> para computação tensorial e desenvolvimento de modelos neurais.
<b>Pandas</b>	Manipulação e análise de dados, com estruturas como <i>Data-Frame</i> para facilitar o tratamento e análise de informações.
<b>NumPy</b>	Operações numéricas eficientes com <i>arrays</i> e matrizes.
<b>Scikit-learn</b>	Conjunto de ferramentas para <i>machine learning</i> , incluindo métodos de classificação, regressão e clusterização.
<b>XGBoost</b>	Biblioteca de <i>gradient boosting</i> para regressão e classificação.

Fonte: Elaborado pelo autor (2025).

No decorrer do desenvolvimento, foram utilizadas as bases de dados movieLens-100k, movieLens-1M e movieLens-10M. Em resumo, cada um dos conjuntos de dados é composto por dados de usuários, filmes e as respectivas avaliações dos usuários em relação aos filmes. A terminologia de cada conjunto se refere à quantidade de avaliações conforme a Tabela 3.6.

### 3.8 CONSIDERAÇÕES FINAIS

Neste capítulo foi apresentado o modelo arquitetural proposto, com o detalhamento específico dos principais módulos que são o *content-based deep learning* e o *collaborative filtering deep learning*. Além disso, foram discutidos os aspectos da implemen-

Tabela 3.6: Conjunto de dados úteis no desenvolvimento

<b>Conjunto</b>	<b>Usuários</b>	<b>Filmes</b>	<b>Avaliações</b>
movieLens-100k	943	1.682	100.000
movieLens-1M	6.040	3.900	1.000.209
movieLens-10M	71.567	10.681	10.000.054

Fonte: Elaborado pelo autor (2025).

tação do algoritmo do sistema de recomendação proposto, bem como as estratégias de otimização e de unificação dos resultados. O próximo capítulo será responsável por apresentar todo o processo de validação, considerando os cenários de teste, todas as parametrizações e critérios, além de apresentar e interpretar os resultados obtidos.

## 4 VALIDAÇÃO DO MODELO HÍBRIDO DE RECOMENDAÇÃO

### 4.1 CONSIDERAÇÕES INICIAIS

Este capítulo possui o objetivo de descrever o processo de validação do modelo híbrido proposto. Desse modo, são definidos os cenários considerados para a efetivação dos testes, como a definição dos conjuntos de dados utilizados, escolha dos parâmetros para serem aferidos, fases de execução dos testes e também a análise comparativa dos resultados. Neste contexto, as seções seguintes irão apresentar a metodologia de validação do sistema de recomendação proposto.

### 4.2 METODOLOGIA DE VALIDAÇÃO

Atualmente há diversas bases de dados de acesso livre que são utilizadas com o intuito de prover dados para a avaliação dos modelos de sistema de recomendação. Conforme Huang et al. (2019) e Sarker e Matin (2021), um dos conjuntos de dados mais utilizados é denominado MovieLens (RAWAT; TOMAR; KUMAR, 2020; HARPER; KONSTAN, 2015), o qual contém informações de usuários (por exemplo, idade, gênero e profissão), filmes (por exemplo, gênero, ano de lançamento e título) e as avaliações fornecidas pelos usuários aos filmes. Em observação, o grupo de pesquisa GroupLens da Universidade de Minnesota (RAWAT; TOMAR; KUMAR, 2020) é responsável por coletar e gerenciar estas bases de dados.

Nesta etapa de validação foram consideradas as bases de dados moviesLens100k, movieLens1M e movieLens10M. Além disso, foram extraídos subconjuntos de dados a partir desses conjuntos principais, com o objetivo de avaliar o desempenho do

modelo ao lidar com diferentes volumes de dados. O detalhamento de cada conjunto utilizado está descrito na Tabela 4.1.

Tabela 4.1: Conjuntos de dados de validação

Conjunto	Usuários	Filmes	Avaliações
movieLens-subset-10k	943	1.682	10.000
movieLens-subset-50k	943	1.682	50.000
movieLens-100k	943	1.682	100.000
movieLens-subset-250k	6.040	3.900	250.000
movieLens-subset-500k	6.040	3.900	500.000
movieLens-1M	6.040	3.900	1.000.209
movieLens-10M	71.567	10.681	10.000.054

**Fonte:** Elaborado pelo autor (2025)

Em observação, foi construído um método para converter a estrutura de cada um dos conjuntos de dados descrito na Tabela 4.1 em um padrão específico que foi definido como critério de entrada dos dados para o algoritmo desenvolvido, uma vez que cada um dos conjuntos possuem estruturas distintas. Essa estrutura padronizada foi definida na especificação da arquitetura na Subseção 3.3.1 e está detalhada nas Tabelas 3.1, 3.2 e 3.3, e o objetivo principal é permitir que diferentes conjuntos de dados possam ser utilizados no treinamento do modelo, desde que sigam as instruções do dicionário de dados padrão.

A funcionalidade de padronização inicial dos dados gera três arquivos nomeados de “users.txt”, que possui dados descritivos dos usuários, “movies.txt”, que é composto pelas características dos filmes e, por fim, o arquivo “ratings.txt”, que possui a relação de usuários, filmes e suas respectivas notas de avaliação. Desse modo, estes arquivos são carregados para serem utilizados como entrada no processo de treinamento do algoritmo. A estrutura destes dados pode ser visualizada nas Figuras 4.1, 4.2, 4.3.

Figura 4.1: Arquivo users.txt

1	1 24 M technician 85711	1	16 21 M entertainment 10309
2	2 53 F other 94043	2	17 30 M programmer 06355
3	3 23 M writer 32067	3	18 35 F other 37212
4	4 24 M technician 43537	4	19 40 M librarian 02138
5	5 33 F other 15213	5	20 42 F homemaker 95660
6	6 42 M executive 98101	6	21 26 M writer 30068
7	7 57 M administrator 91344	7	22 25 M writer 40206
8	8 36 M administrator 05201	8	23 30 F artist 48197
9	9 29 M student 01002	9	24 21 F artist 94533
10	10 53 M lawyer 90703	10	25 39 M engineer 55107
11	11 39 F other 30329	11	26 49 M engineer 21044
12	12 28 F other 06405	12	27 40 F librarian 30030
13	13 47 M educator 29206	13	28 32 M writer 55369
14	14 45 M scientist 55106	14	29 41 M programmer 94043
15	15 49 F educator 97301	15	30 7 M student 55436

Fonte: Elaboração própria com base em Harper e Konstan (2015).

Figura 4.2: Arquivo movies.txt

1	50 Star Wars (1977) 01-Jan-1977 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0
2	54 Outbreak (1995) 01-Jan-1995  0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
3	55 Professional, The (1994) 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0
4	56 Pulp Fiction (1994) 01-Jan-1994 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0
5	58 Quiz Show (1994) 01-Jan-1994  0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
6	60 Three Colors: Blue (1993) 01-Jan-1993  0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
7	62 Stargate (1994) 01-Jan-1994 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
8	63 Santa Clause, The (1994) 01-Jan-1994 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
9	64 Shawshank Redemption, The (1994) 01-Jan-1994 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
10	67 Ace Ventura: Pet Detective (1994) 01-Jan-1994 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
11	68 Crow, The (1994) 01-Jan-1994 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0
12	69 Forrest Gump (1994) 01-Jan-1994  0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0

Fonte: Elaboração própria com base em Harper e Konstan (2015).

Figura 4.3: Arquivo ratings.txt

1	387	520	4	886480446	11	246	895	5	884924976
2	119	109	5	874775580	12	231	252	4	888605273
3	346	809	3	874951029	13	144	762	3	888104940
4	339	523	5	891033044	14	254	755	3	886473489
5	222	529	2	881059537	15	5	397	2	875635907
6	230	8	5	880484501	16	303	473	4	879485111
7	329	338	2	891655545	17	151	203	3	879524471
8	184	647	5	889909024	18	168	117	5	884287318
9	292	855	5	881105373	19	323	151	4	878739568
10	60	207	3	883327342	20	90	995	4	891382708

Fonte: Elaboração própria com base em Harper e Konstan (2015).

Desse modo, esta etapa de validação também utilizará a base de dados MovieLens com o objetivo de usufruir desta massa de dados e possibilitar o treinamento e validação do sistema de recomendação proposto. Ademais, ao aplicar o modelo de recomendação serão utilizadas métricas para aferir as previsões realizadas, com o objetivo de verificar os resultados obtidos e considerá-los como parâmetros comparativos a outros modelos já existentes. Serão utilizadas as seguintes métricas:

- Erro quadrático médio (*Root Mean Squared Error – RMSE*): métrica utilizada para calcular a precisão do sistema. Quanto menor o valor de RMSE indica menos erros nas previsões do algoritmo e melhor performance do sistema, logo melhores recomendações (RAWAT; TOMAR; KUMAR, 2020). Considera-se a Equação 4.1, onde  $N$  é o número total de amostras.

$$RMSE = \sqrt{\frac{\sum_{i=0}^N (\text{predição}_i - \text{atual}_i)^2}{N}} \quad (4.1)$$

- Erro absoluto médio (*Mean Absolute Error - MAE*): é calculado pela média dos valores de erro absoluto. Valores baixos de MAE indicam uma menor incidência de erros nas predileções calculas, e então maior qualidade na recomendação efetuada pelo sistema (ALFARHOOD; CHENG, 2018). Considera-se  $N$  o número total de amostras, então MAE pode ser calculado de acordo com a formulação 4.2.

$$MAE = \frac{\sum_{i=0}^N |\text{predição}_i - \text{atual}_i|}{N} \quad (4.2)$$

- Precisão: mensura a porcentagem de acerto das preferências do usuário, logo quanto maior o valor de precisão mais qualificadas são as recomendações. Na prática, denota a quantidade de recomendações que são relevantes entre todas as obtidas. Seja  $TP$  o número de amostras corretamente classificadas como verda-

deiras e  $FP$ , o número de amostras classificadas incorretamente como positivas, assim pode-se calcular a fórmula 4.3 (Fayyaz et al, 2020).

$$P = \frac{TP}{TP + FP} \quad (4.3)$$

- *Recall*: analisa o interesse do usuário de acordo com a recomendação e quanto maior seu valor, mais específica é a recomendação. Desse modo, o *recall* representa a porcentagem de previsões verdadeiras nas amostras, ou seja, dentre o total de itens relevantes qual a proporção que o algoritmo foi capaz de recomendar. Considerando  $TP$  o número de itens positivos que foram corretamente previstos como verdadeiros e  $FN$  o número de itens positivos que foram incorretamente previstos como falsos, então pode-se obter o valor de *recall* através do cálculo da Equação 4.4 (FAYYAZ et al., 2020).

$$Recall = \frac{TP}{TP + FN} \quad (4.4)$$

- *F1-Score*: esta métrica é basicamente a média harmônica entre *recall* e precisão. O máximo valor possível é 1,00 que indica perfeita precisão e *recall*. O cálculo é aplicado de acordo com a Fórmula 4.5 (Fayyaz et al, 2020).

$$F = 2 \cdot \frac{\text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}} \quad (4.5)$$

- Acurácia: é uma métrica básica que avalia a precisão de um método fundamentado no número de recomendações verdadeiras (Fayyaz et al, 2020). É primordial para entender a taxa de acerto global do sistema de recomendação e possibilita uma interpretação macro acerca da performance das previsões obtidas. O cálculo é realizado de acordo com a Equação 4.6.

$$F = \frac{\text{Número de recomendações bem-sucedidas}}{\text{Número total de recomendações}} \quad (4.6)$$

### 4.3 ETAPAS DE VALIDAÇÃO

A validação está segmentada em duas etapas, a primeira é relacionada a validação do sistema de recomendação proposto em relação às suas próprias abordagens de unificação de resultados. Já a segunda etapa, refere-se à validação dos resultados obtidos com este sistema de recomendação proposto com relação a outros sistemas atualmente disponíveis.

As seções 4.4 e 4.5 a seguir, descrevem as etapas, metodologias utilizadas, bem como os resultados obtidos.

### 4.4 VALIDAÇÃO DOS MÉTODOS DE UNIFICAÇÃO DE RESULTADOS

O objetivo principal é analisar o desempenho dos módulos *content-based* e *collaborative filtering* individualmente, bem como cada uma das estratégias de unificação de resultados. Nesse contexto, serão utilizados os conjuntos de dados para treinamento e posterior avaliação das métricas de verificação de resultados.

#### 4.4.1 Cenário A: 10 mil avaliações

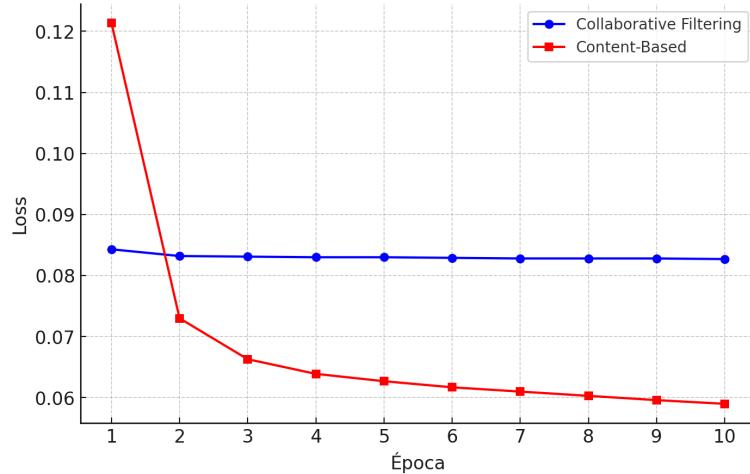
Utiliza a base de dados movieLens-subset-10k para treinamento dos modelos e os seguintes cenários:

- **Treinamento de 10 épocas do modelo:**

A Figura 4.4 exibe a convergência de cada um dos módulos individualmente ao longo do treinamento, em que é possível notar que o *content-based* possui uma variação maior em relação ao *collaborative-filtering*. Todavia, ambos são

capazes de convergir e se estabilizarem em relação ao erro quadrático médio de cada iteração do treinamento do modelo.

Figura 4.4: Perda por épocas de treinamento dos módulos



Fonte: Elaborado pelo autor (2025)

Tabela 4.2: Resultados com 10 mil avaliações e 10 épocas de treinamento

Modelo	Precisão	Recall	F1-S.	Acurácia	RMSE	MAE
<i>Content-based</i>	0.6675	0.7374	0.7007	0.6425	0.2701	0.2164
<i>Collaborative</i>	0.5675	0.9999	0.7241	0.5675	0.2826	0.2365
Soma Ponderada	0.5675	0.9999	0.7241	0.5675	0.2826	0.2365
Pesos Dinâmicos	0.6748	0.8502	0.7524	0.6825	0.2494	0.1928
Fusão Bayesiana	0.5675	0.9999	0.7241	0.5675	0.3079	0.2331
Fusão Sigmoide	0.5745	0.9982	0.7293	0.5795	0.2769	0.229
<i>Random Forest</i>	0.6304	0.6987	0.6628	0.5965	0.3129	0.2466
Fusão <i>XgBoost</i>	0.6612	0.7753	0.7137	0.647	0.2741	0.2173
<i>Stacking Meta-L</i>	0.5965	0.9665	0.7377	0.61	0.2732	0.2259

Fonte: Elaborado pelo autor (2025)

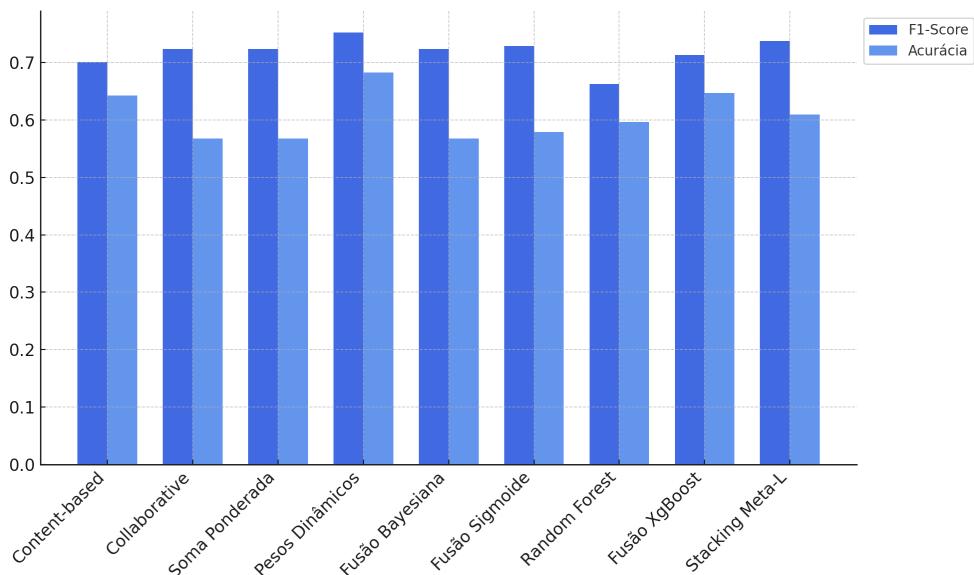
Ao analisar individualmente os módulos *content-based* e *collaborative-filtering*, verifica-se que a precisão é maior no primeiro, o que indica que o modelo tem uma boa taxa de acerto em relação a itens relevantes. Por outro lado, ao ana-

lisar a Tabela 4.2 é perceptível uma precisão menor no colaborativo e um alto *recall*, em que, uma das interpretações é a indicação de recomendações com maior diversidade e não apenas as mais previsíveis. Logo, justifica o uso de uma abordagem híbrida para evitar o clássico problema da bolha do filtro de recomendações discutida no Capítulo 2 deste trabalho.

Ademais, para uma efetiva avaliação da precisão e *recall*, é importante utilizar uma métrica que equilibrar o resultado de ambas, como o *F1-Score*. Nesse sentido, verifica-se que os modelos híbridos de Fusão Sigmóide, *Stacking Meta-Learning* e Pesos Dinâmicos possuem os melhores resultados, conforme Figura 4.5. Logo, possuem um equilíbrio relevante entre a precisão e o *recall*, ou seja, são eficientes em recomendar itens de relevância com precisão e também abrangência.

Acerca da acurácia, o modelo híbrido de pesos dinâmicos obteve o melhor desempenho, o que indica uma boa taxa de recomendações corretas em relação ao total de recomendações realizadas.

Figura 4.5: *F1-Score* e Acurácia com 10 épocas de treinamento



Fonte: Elaborado pelo autor (2025)

Ao analisar sob a perspectiva dos erros RMSE e MAE, verifica-se que o modelo híbrido de pesos dinâmicos possui as menores taxas, e, na sequência, estão os modelos *Stacking Meta-Learning* e Fusão *XgBoost*. Estas métricas com valores baixos indicam que as recomendações do modelo foram próximas dos valores reais.

- **Treinamento de 30 épocas do modelo:**

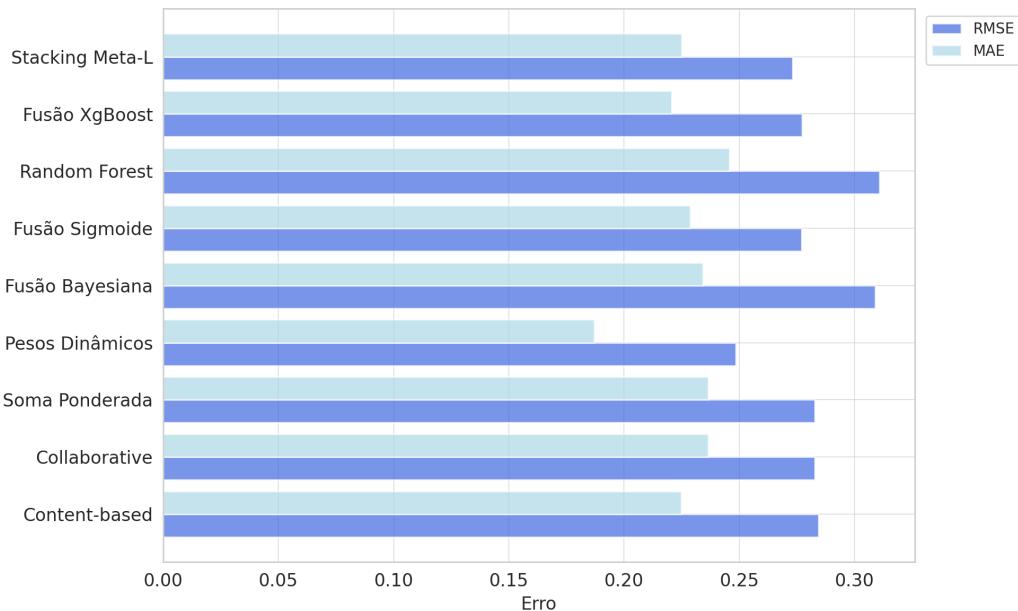
Nesta etapa, é possível verificar que os modelos mais equilibrados em relação à taxa de acertos de recomendações relevantes é com a estratégia de pesos dinâmicos, *Stacking Meta-Model* e Sigmoide, conforme a Tabela 4.3. Em termos de acurácia o melhor desempenho é relacionado à abordagem de pesos dinâmicos, que também possui as menores taxas de erro RMSE e MAE, juntamente com *XgBoost*, conforme a Figura 4.6.

Tabela 4.3: Resultados com 10 mil avaliações e 30 épocas de treinamento

<b>Modelo</b>	<b>Precisão</b>	<b>Recall</b>	<b>F1-S.</b>	<b>Acurácia</b>	<b>RMSE</b>	<b>MAE</b>
<i>Content-based</i>	0.6543	0.7154	0.6835	0.624	0.2844	0.2248
<i>Collaborative</i>	0.5675	0.9999	0.7241	0.5675	0.2826	0.2365
Soma Ponderada	0.5675	0.9999	0.7241	0.5675	0.2826	0.2365
Pesos Dinâmicos	0.6633	0.8140	0.7310	0.6600	0.2484	0.1869
Fusão Bayesiana	0.5675	0.9999	0.7241	0.5675	0.3088	0.2342
Fusão Sigmoide	0.5785	0.9938	0.7313	0.5855	0.2769	0.2287
<i>Random Forest</i>	0.6318	0.7137	0.6703	0.6015	0.3107	0.2456
Fusão <i>XgBoost</i>	0.6278	0.8159	0.7096	0.621	0.2771	0.2207
<i>Stacking Meta-L</i>	0.6004	0.9480	0.7352	0.6125	0.2731	0.2250

**Fonte:** Elaborado pelo autor (2025)

Figura 4.6: RMSE e MAE com 30 épocas de treinamento



Fonte: Elaborado pelo autor (2025)

- **Treinamento de 50 épocas do modelo:**

Os resultados obtidos denotam que a fusão sigmoide, *stacking meta-model* e fusão bayesiana obtiveram relevantes resultados de *F1-Score*, isto é, um bom balanceamento entre precisão e *recall*. Contudo, os métodos bayesiano e sigmoide denotam maiores taxas de erro expressas nos valores de RMSE e MAE. Nesse sentido, ao considerar a combinação de taxas de erros e acurácia, constata-se que a estratégia de pesos dinâmicos, *XgBoost* e *Stacking Meta-Model* obtém resultados melhores, conforme a Tabela 4.4.

#### 4.4.2 Cenário B: 50 mil avaliações

Utiliza a base de dados movieLens-subset-50k para treinamento e validação dos modelos, conforme a seguir:

- **Treinamento de 10 épocas do modelo:**

Avaliação do balanceamento entre maior precisão e acurácia, além das menores

Tabela 4.4: Resultados com 10 mil avaliações e 50 épocas de treinamento

<b>Modelo</b>	<b>Precisão</b>	<b>Recall</b>	<b>F1-S.</b>	<b>Acurácia</b>	<b>RMSE</b>	<b>MAE</b>
<i>Content-based</i>	0.6465	0.7233	0.6827	0.6185	0.2878	0.2257
<i>Collaborative</i>	0.5675	1.0000	0.7241	0.5675	0.2826	0.2365
Soma Ponderada	0.5675	1.0000	0.7241	0.5675	0.2826	0.2365
Pesos Dinâmicos	0.6546	0.8000	0.7200	0.6470	0.2489	0.1882
Fusão Bayesiana	0.5675	1.0000	0.7241	0.5675	0.3087	0.2342
Fusão Sigmoide	0.5782	0.9903	0.7310	0.5845	0.2771	0.2291
<i>Random Forest</i>	0.6305	0.7004	0.6636	0.5967	0.3154	0.2486
Fusão <i>XgBoost</i>	0.6339	0.7489	0.6866	0.6120	0.2761	0.2193
<i>Stacking Meta-L</i>	0.5978	0.9233	0.7258	0.6040	0.2730	0.2243

**Fonte:** Elaborado pelo autor (2025)

taxas de erro, denotam que os modelos de pesos dinâmicos, *XGBoost* e *Stacking Meta Learning* possuem melhor desempenho, conforme Tabela 4.5.

Tabela 4.5: Resultados com 50 mil avaliações e 10 épocas de treinamento

<b>Modelo</b>	<b>Precisão</b>	<b>Recall</b>	<b>F1-S.</b>	<b>Acurácia</b>	<b>RMSE</b>	<b>MAE</b>
<i>Content-based</i>	0.6873	0.6649	0.6759	0.6473	0.2625	0.2129
<i>Collaborative</i>	0.5532	0.9999	0.7123	0.5532	0.2861	0.2411
Soma Ponderada	0.5532	0.9999	0.7123	0.5532	0.2861	0.2411
Pesos Dinâmicos	0.6837	0.7288	0.7055	0.6635	0.2589	0.2054
Fusão Bayesiana	0.5532	0.9999	0.7123	0.5532	0.3127	0.2360
Fusão Sigmoide	0.5532	0.9999	0.7123	0.5532	0.2814	0.2336
<i>Random Forest</i>	0.6250	0.6777	0.6503	0.5968	0.3143	0.2492
Fusão <i>XgBoost</i>	0.6624	0.7681	0.7113	0.6551	0.2614	0.2106
<i>Stacking Meta-L</i>	0.6658	0.7448	0.7031	0.6520	0.2664	0.2187

**Fonte:** Elaborado pelo autor (2025)

- Treinamento de 30 épocas do modelo:

Tabela 4.6: Resultados com 50 mil avaliações e 30 épocas de treinamento

Modelo	Precisão	Recall	F1-S.	Acurácia	RMSE	MAE
<i>Content-based</i>	0.6829	0.6759	0.6794	0.6471	0.2619	0.2122
<i>Collaborative</i>	0.5532	0.9999	0.7123	0.5532	0.2861	0.2411
Soma Ponderada	0.5532	0.9999	0.7123	0.5532	0.2861	0.2411
Pesos Dinâmicos	0.6803	0.7368	0.7074	0.6629	0.2575	0.2038
Fusão Bayesiana	0.5532	0.9999	0.7123	0.5532	0.3125	0.2360
Fusão Sigmoide	0.5532	0.9999	0.7123	0.5532	0.2812	0.2334
<i>Random Forest</i>	0.6277	0.7079	0.6654	0.6061	0.2900	0.2320
Fusão <i>XgBoost</i>	0.6605	0.7657	0.7093	0.6527	0.2612	0.2105
<i>Stacking Meta-L</i>	0.6642	0.7529	0.7058	0.6527	0.2661	0.2185

**Fonte:** Elaborado pelo autor (2025)

Assim como a interpretação anterior, é possível verificar na Tabela 4.6 que os mesmos modelos pesos dinâmicos, *XGBoost* e *Stacking Meta Learning* possuem melhor desempenho, se mantém com os melhores resultados, ao levar em consideração o balanceamento entre os parâmetros.

- **Treinamento de 50 épocas do modelo:**

Neste cenário de testes é possível constatar que os modelos de soma ponderada, bayesiano e *random forest*, possuem tendência a inserir maior quantidade de erros nas previsões realizadas, conforme Tabela 4.7.

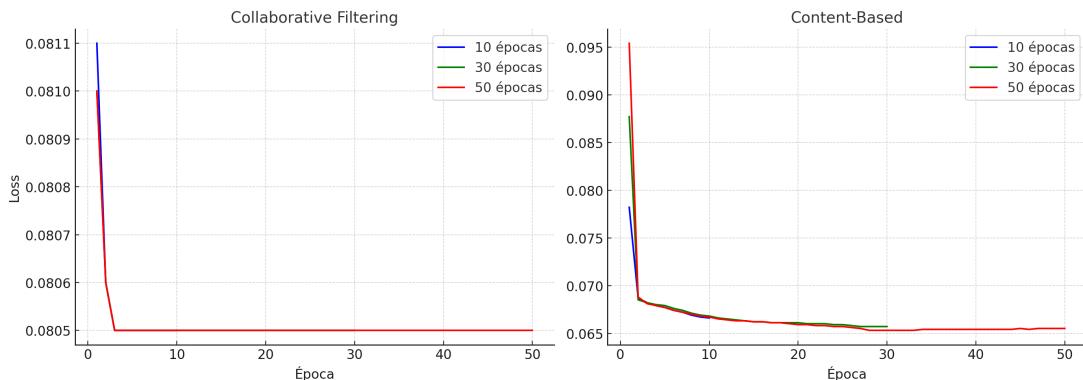
No contexto geral relacionado aos três cenários de validação baseado no conjunto de dados de 50 mil avaliações, verifica-se que o processo de treinamento completo de até cinquenta épocas, demonstra uma rápida convergência em ambas as abordagens. Nesse sentido, a partir da Figura 4.7 pode-se verificar que no *collaborative filtering*, a estabilização inicia-se antes mesmo de dez épocas, já no modelo *content-based* a convergência é a partir de vinte épocas de treinamento.

Tabela 4.7: Resultados com 50 mil avaliações e 50 épocas de treinamento

<b>Modelo</b>	<b>Precisão</b>	<b>Recall</b>	<b>F1-S.</b>	<b>Acurácia</b>	<b>RMSE</b>	<b>MAE</b>
<i>Content-based</i>	0.6811	0.6799	0.6805	0.6468	0.2621	0.2122
<i>Collaborative</i>	0.5532	1.0000	0.7123	0.5532	0.2861	0.2411
Soma Ponderada	0.5532	1.0000	0.7123	0.5532	0.2861	0.2411
Pesos Dinâmicos	0.6793	0.7380	0.7075	0.6624	0.2573	0.2033
Fusão Bayesiana	0.5532	1.0000	0.7123	0.5532	0.3126	0.2362
Fusão Sigmoide	0.5532	1.0000	0.7123	0.5532	0.2812	0.2333
<i>Random Forest</i>	0.6296	0.6750	0.6515	0.6005	0.3108	0.2463
Fusão <i>XgBoost</i>	0.6584	0.7661	0.7082	0.6507	0.2616	0.2108
<i>Stacking Meta-L</i>	0.6590	0.7731	0.7115	0.6532	0.2689	0.2226

**Fonte:** Elaborado pelo autor (2025)

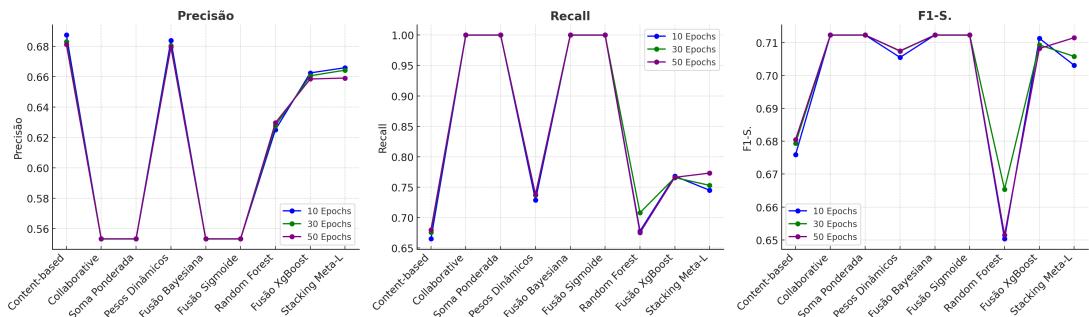
Figura 4.7: Estabilização da perda ao longo do treinamento dos modelos



**Fonte:** Elaborado pelo autor (2025)

A análise comparativa e evolutiva entre as épocas de treinamento possibilita constatar que os resultados obtidos por cada algoritmo se mantém constante ou com brandas alterações no decorrer do ciclo iterativo, conforme a Figura 4.8.

Figura 4.8: Combinação de Precisão, *Recall* e *F1-Score* com 10, 30 e 50 épocas



Fonte: Elaborado pelo autor (2025)

Nesse sentido, pode-se afirmar que há estabilidade no comportamento dos modelos, uma vez que não apresentam variações muito destoantes entre as épocas de treinamento.

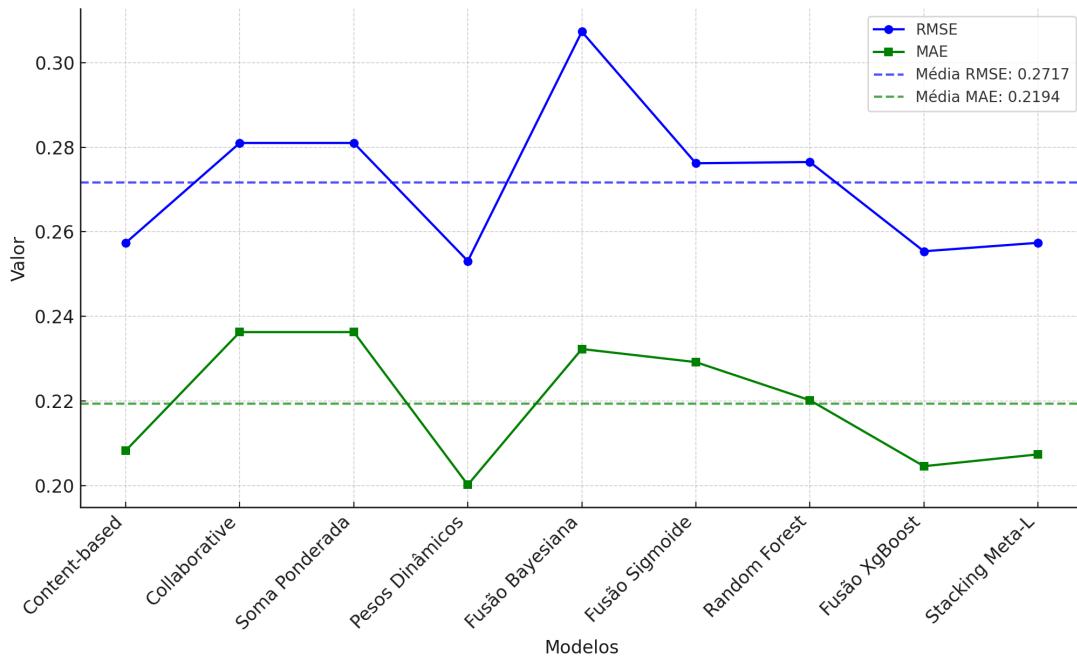
#### 4.4.3 Cenário C: 100 mil avaliações

Utilização da base de dados movieLens-100k para treino e validação dos modelos, conforme demonstrado nas etapas a seguir:

- **Treinamento de 10 épocas do modelo:**

Os resultados obtidos são exibidos na Tabela 4.8, e constatam que em termos de precisão e acurácia, os melhores resultados são referentes a *content-based*, pesos dinâmicos, *Stacking Meta-Model* e Fusão *XGboost*, conforme a Tabela 4.8. Além disso, é possível observar que estes mesmos modelos, possuem os menores erros obtidos nos parâmetros RMSE e MAE, conforme a Figura 4.9.

Figura 4.9: Comparação entre RMSE e MAE após 10 épocas de treinamento



Fonte: Elaborado pelo autor (2025)

Tabela 4.8: Resultados com 100 mil avaliações e 10 épocas de treinamento

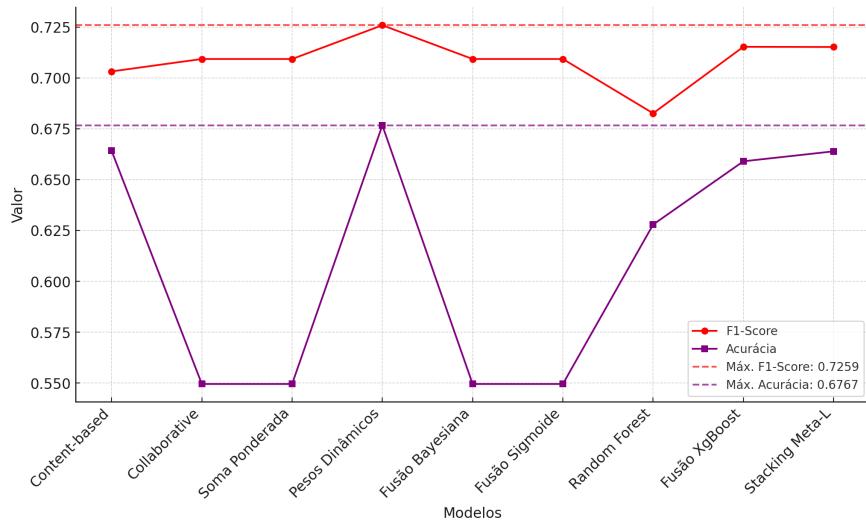
Modelo	Precisão	Recall	F1-S.	Acurácia	RMSE	MAE
<i>Content-based</i>	0.6916	0.6877	0.6897	0.6599	0.2574	0.2083
<i>Collaborative</i>	0.5495	0.9999	0.7093	0.5495	0.2810	0.2363
<i>Soma Ponderada</i>	0.5495	0.9999	0.7093	0.5495	0.2810	0.2363
<i>Pesos Dinâmicos</i>	0.6881	0.7586	0.7216	0.6784	0.2531	0.2002
<i>Fusão Bayesiana</i>	0.5495	0.9999	0.7093	0.5495	0.3073	0.2323
<i>Fusão Sigmoide</i>	0.5495	0.9999	0.7093	0.5495	0.2762	0.2292
<i>Random Forest</i>	0.6421	0.7286	0.6826	0.6277	0.2765	0.2202
<i>Fusão XgBoost</i>	0.6625	0.7826	0.7176	0.6614	0.2554	0.2046
<i>Stacking Meta-L</i>	0.6695	0.7626	0.7130	0.6626	0.2574	0.2074

Fonte: Elaborado pelo autor (2025)

- Treinamento de 30 épocas do modelo:

Os resultados denotam que novamente as estratégias de pesos dinâmicos, Fusão *XgBoost* e *Stacking Meta-Learning* possuem os melhores resultados principalmente na correlação entre *F1-Score* e Acurácia, conforme Figura 4.10 e Tabela 4.9.

Figura 4.10: *F1-Score* e Acurácia após 30 épocas de treinamento



Fonte: Elaborado pelo autor (2025)

Tabela 4.9: Resultados com 100 mil avaliações e 30 épocas de treinamento

Modelo	Precisão	Recall	F1-S.	Acurácia	RMSE	MAE
<i>Content-based</i>	0.6836	0.7240	0.7032	0.6642	0.2560	0.2059
<i>Collaborative</i>	0.5495	0.9999	0.7093	0.5495	0.2810	0.2363
<i>Soma Ponderada</i>	0.5495	0.9999	0.7093	0.5495	0.2810	0.2363
<i>Pesos Dinâmicos</i>	0.6793	0.7794	0.7259	0.6767	0.2506	0.1967
<i>Fusão Bayesiana</i>	0.5495	0.9999	0.7093	0.5495	0.3076	0.2331
<i>Fusão Sigmoidal</i>	0.5495	0.9999	0.7093	0.5495	0.2759	0.2287
<i>Random Forest</i>	0.6423	0.7284	0.6826	0.6279	0.2761	0.2194
<i>Fusão XgBoost</i>	0.6608	0.7796	0.7153	0.6395	0.2550	0.2040
<i>Stacking Meta-L</i>	0.6692	0.7681	0.7152	0.6639	0.2568	0.2066

Fonte: Elaborado pelo autor (2025)

- **Treinamento de 50 épocas do modelo:**

Tabela 4.10: Resultados com 100 mil avaliações e 50 épocas de treinamento

<b>Modelo</b>	<b>Precisão</b>	<b>Recall</b>	<b>F1-S.</b>	<b>Acurácia</b>	<b>RMSE</b>	<b>MAE</b>
<i>Content-based</i>	0.6792	0.7369	0.7068	0.6642	0.2557	0.2057
<i>Collaborative</i>	0.5495	1.0000	0.7093	0.5495	0.2810	0.2363
Soma Ponderada	0.5495	1.0000	0.7093	0.5495	0.2810	0.2363
Pesos Dinâmicos	0.6767	0.7915	0.7296	0.6777	0.2513	0.1976
Fusão Bayesiana	0.5495	1.0000	0.7093	0.5495	0.3080	0.2335
Fusão Sigmoide	0.5495	1.0000	0.7093	0.5495	0.2760	0.2288
<i>Random Forest</i>	0.6433	0.7310	0.6844	0.6294	0.2757	0.2194
Fusão <i>XgBoost</i>	0.6689	0.7696	0.7157	0.6641	0.2552	0.2042
<i>Stacking Meta-L</i>	0.6672	0.7786	0.7186	0.6650	0.2572	0.2075

**Fonte:** Elaborado pelo autor (2025)

A avaliação dos resultados também demonstram o desempenho superior das abordagens pesos dinâmicos, Fusão *XgBoost* e *Stacking Meta-Model*, principalmente nos parâmetros precisão, *F1-score*, Acurácia e menores taxas de erro em RMSE e MAE, conforme a Tabela 4.10.

#### 4.4.4 Cenário D: um milhão de avaliações

Foi utilizado o conjunto de dados movieLens-1M e os resultados obtidos estão na sequência:

- **Treinamento de 10 épocas do modelo:**

A verificação dos dados também confirmam as análises previamente realizadas nos cenários anteriores, com predominância de melhor performance e menos erros com as estratégias de pesos dinâmicos, fusão *XgBoost* e *Stacking Meta-Model*, conforme Tabela 4.11.

Tabela 4.11: Resultados com um milhão de avaliações e treino de dez épocas

<b>Modelo</b>	<b>Precisão</b>	<b>Recall</b>	<b>F1-S.</b>	<b>Acurácia</b>	<b>RMSE</b>	<b>MAE</b>
<i>Content-based</i>	0.7120	0.6678	0.6892	0.6533	0.2566	0.2063
<i>Collaborative</i>	0.5757	0.9999	0.7307	0.5757	0.2800	0.2351
Soma Ponderada	0.5757	0.9999	0.7307	0.5757	0.2800	0.2351
Pesos Dinâmicos	0.7339	0.8727	0.7973	0.7445	0.2535	0.2002
Fusão Bayesiana	0.5757	0.9999	0.7307	0.5757	0.3040	0.2295
Fusão Sigmoide	0.5757	0.9999	0.7307	0.5757	0.2752	0.2285
<i>Random Forest</i>	0.6720	0.7501	0.7089	0.6454	0.2708	0.2149
Fusão <i>XgBoost</i>	0.6784	0.7973	0.7331	0.6657	0.2547	0.2037
<i>Stacking Meta-L</i>	0.7093	0.6799	0.6943	0.6553	0.2561	0.2050

**Fonte:** Elaborado pelo autor (2025)

#### 4.5 ANÁLISE COMPARATIVA ENTRE SISTEMAS DE RECOMENDAÇÃO

O principal objetivo desta seção é considerar as abordagens de maior relevância de acordo com os resultados obtidos na primeira parte da validação e compará-los com clássicas abordagens de sistemas de recomendação. Desse modo, foram selecionados os algoritmos KNN *user-based*, SVD e NMF, os quais são modelos tradicionais de recomendação conforme observado na literatura científica. Além disso, foram incluídos algoritmos que utilizam estratégias de aprendizado profundo, como o *AutoEnconder Based*, NCF e DeepFM.

As abordagens Pesos Dinâmicos, Fusão *XgBoost* e *Stacking Meta-Learning* do algoritmo proposto neste trabalho, foram selecionadas para este comparativo pois obtiveram os melhores resultados na primeira etapa de validação. Além disso, para efeito de comparação, a mesma quantidade de épocas de treinamento do modelo proposto foi utilizada para treinar os modelos comparativos que dependem dessa fase de aprendizado.

O processo de avaliação e resultados obtidos será apresentado a seguir.

#### 4.5.1 Cenário A: 10 mil avaliações

Utilização do conjunto de dados movieLens-subset-10k e aplicação de cinquenta épocas de treinamento.

A partir da análise visual do gráfico de barras da Figura 4.11, fica evidente que as maiores precisões ocorrem no KNN-User, SVD, NMF e *Autoencoder*, porém possuem pouca relevância em *recall* e *F1-Score*. Esse comportamento observado sugere que são eficientes em recomendar os itens mais óbvios, todavia são ineficientes em recomendar com maior grau de diversificação.

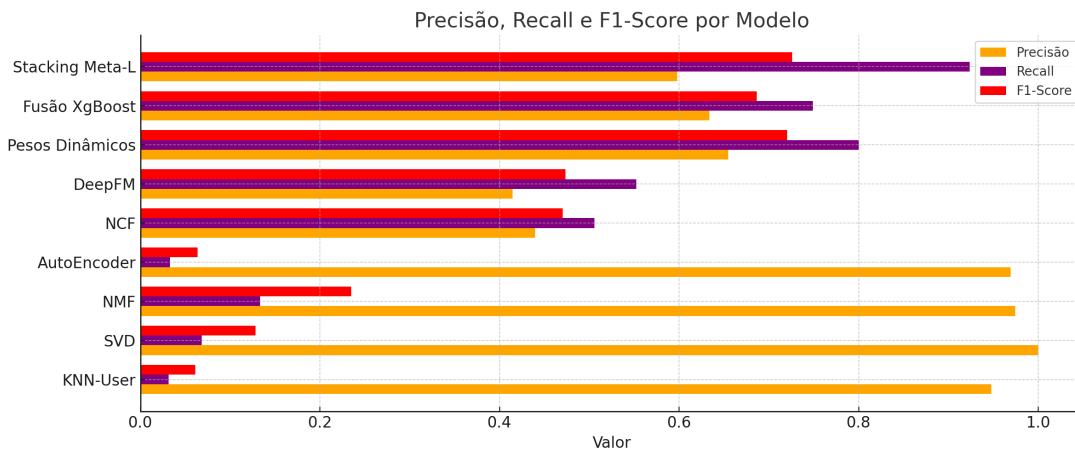
Tabela 4.12: Resultados comparativos do Cenário A com 10 mil avaliações

Modelo	Precisão	Recall	F1-S.	Acurácia	RMSE	MAE
KNN-User	0.9474	0.0316	0.0611	0.447	2.5051	2.3254
SVD	1.0000	0.0684	0.1281	0.469	2.7413	2.4683
NMF	0.9744	0.1333	0.2346	0.504	2.9329	2.5892
<i>AutoEncoder</i>	0.9689	0.0329	0.0636	0.450	2.9644	2.6499
NCF	0.4396	0.5055	0.4703	0.7455	1.0224	0.8080
<i>DeepFM</i>	0.4144	0.5525	0.4736	0.7255	1.0229	0.8086
Pesos Dinâmicos	0.6546	0.8000	0.7200	0.647	0.2489	0.1882
Fusão <i>XgBoost</i>	0.6339	0.7489	0.6866	0.612	0.2761	0.2193
<i>Stacking Meta-L</i>	0.5978	0.9233	0.7258	0.604	0.2732	0.2243

**Fonte:** Elaborado pelo autor (2025)

Por outro lado, a análise comparativa destas métricas da Tabela 4.12 e ao levar em consideração o melhor balanceamento entre as mesmas, nota-se que *Stacking Meta-learning*, Fusão *XgBoost* e Pesos dinâmicos obtém os melhores resultados.

Figura 4.11: Comparação dos parâmetros Precisão, *Recall* e *F1-Score*



Fonte: Elaborado pelo autor (2025)

#### 4.5.2 Cenário B: 50 mil avaliações

Utilização dos dados movieLens-subset-50k e cinquenta épocas de treinamento.

Tabela 4.13: Resultados comparativos do Cenário B com 50 mil avaliações

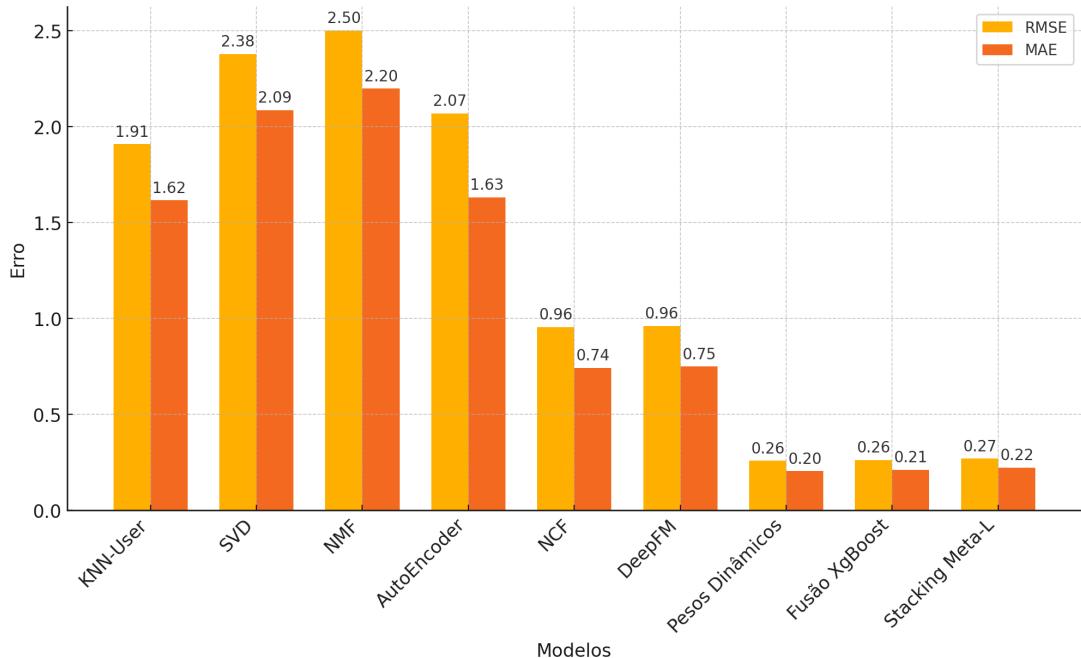
Modelo	Precisão	Recall	F1-S.	Acurácia	RMSE	MAE
KNN-User	0.8952	0.2063	0.3353	0.560	1.9082	1.6158
SVD	0.8986	0.1152	0.2043	0.517	2.3784	2.0869
NMF	0.8833	0.0985	0.1773	0.508	2.5008	2.1985
AutoEncoder	0.96376	0.2492	0.3961	0.57508	2.0688	1.6306
NCF	0.4324	0.5822	0.4962	0.7513	0.9557	0.7427
DeepFM	0.4395	0.5446	0.4865	0.7581	0.9613	0.7490
Pesos Dinâmicos	0.6793	0.7380	0.7075	0.6624	0.2573	0.2033
Fusão XgBoost	0.6584	0.7661	0.7082	0.6507	0.2616	0.2108
Stacking Meta-L	0.6590	0.7731	0.7115	0.6532	0.2689	0.2226

Fonte: Elaborado pelo autor (2025)

A análise dos erros entre os modelos avaliados denota que as menores taxas também ocorrem com a estratégias de Pesos Dinâmicos, Fusão *XgBoost* e *Stacking Meta-Learning*, conforme a Figura 4.12. Por outro lado, nota-se que melhorou a acurácia

dos modelos DeepMF e NCF, nesta etapa com maior volume de dados disponíveis, conforme a Tabela 4.13.

Figura 4.12: Comparação de RMSE e MAE após 50 épocas de treinamento



Fonte: Elaborado pelo autor (2025)

#### 4.5.3 Cenário C: 100 mil avaliações

Aplicação da avaliação com os dados conjunto movieLens-100k e treinamento de 50 épocas. Conforme resultados documentados na Tabela 4.14.

Em geral, todos os algoritmos possuem bom desempenho em relação à acurácia, ou seja, possuem uma boa taxa de acerto em relação ao total de previsões realizadas. Contudo, esta alta taxa analisada isoladamente pode indicar que o método faz apenas recomendações de itens mais populares, ou seja, pode estar no contexto da bolha do filtro de recomendações. Logo, é fundamental que este parâmetro seja analisado em conjunto com *recall* e *F1-Score*, com o intuito de analisar o balanceamento entre eles, cuja interpretação indica um equilíbrio entre previsões relevantes e com grau de

Tabela 4.14: Resultados comparativos do Cenário C com 100 mil avaliações

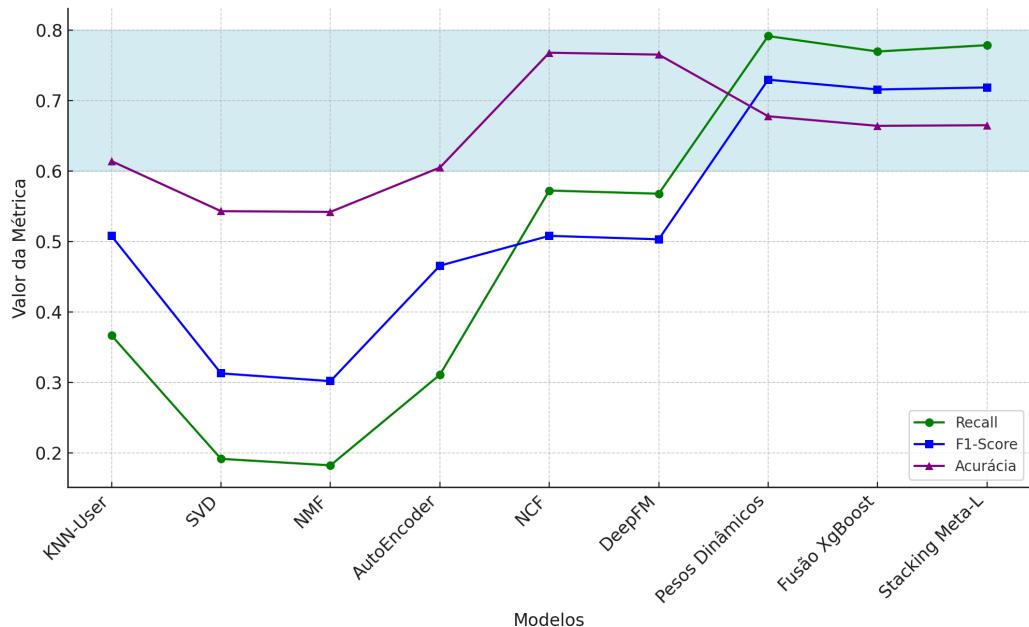
<b>Modelo</b>	<b>Precisão</b>	<b>Recall</b>	<b>F1-S.</b>	<b>Acurácia</b>	<b>RMSE</b>	<b>MAE</b>
KNN-User	0.8257	0.3665	0.5077	0.614	1.586	1.2876
SVD	0.8525	0.1915	0.3128	0.543	2.0488	1.7456
NMF	0.8761	0.1823	0.3018	0.542	2.1595	1.8453
<i>AutoEncoder</i>	0.9273	0.3109	0.4657	0.604	1.7891	1.3998
NCF	0.4566	0.5723	0.5080	0.7679	0.9358	0.7311
<i>DeepFM</i>	0.4516	0.5678	0.5031	0.7652	0.9392	0.7318
Pesos Dinâmicos	0.6767	0.7915	0.7296	0.6777	0.2513	0.1976
Fusão <i>XgBoost</i>	0.6689	0.7696	0.7157	0.6641	0.2552	0.2042
<i>Stacking Meta-L</i>	0.6672	0.7786	0.7186	0.6650	0.2572	0.2075

**Fonte:** Elaborado pelo autor (2025)

diversidade, também chamadas de recomendações ricas.

Nesse sentido, é verificado que as estratégias de Pesos Dinâmicos, Fusão *XgBoost* e *Stacking Meta-learning*, possuem maior estabilidade nos parâmetros, como observado na Figura 4.13. Ademais, os resultados dos erros observados na Tabela 4.14, revelam que também possuem as menores taxas de RMSE e MAE.

Figura 4.13: Comparação de *Recall*, *F1-Score* e Acurácia, com 100 mil avaliações



Fonte: Elaborado pelo autor (2025)

#### 4.5.4 Cenário D: um milhão de avaliações

Utilização dos dados movieLens-1M e treinamento de dez épocas dos modelos.

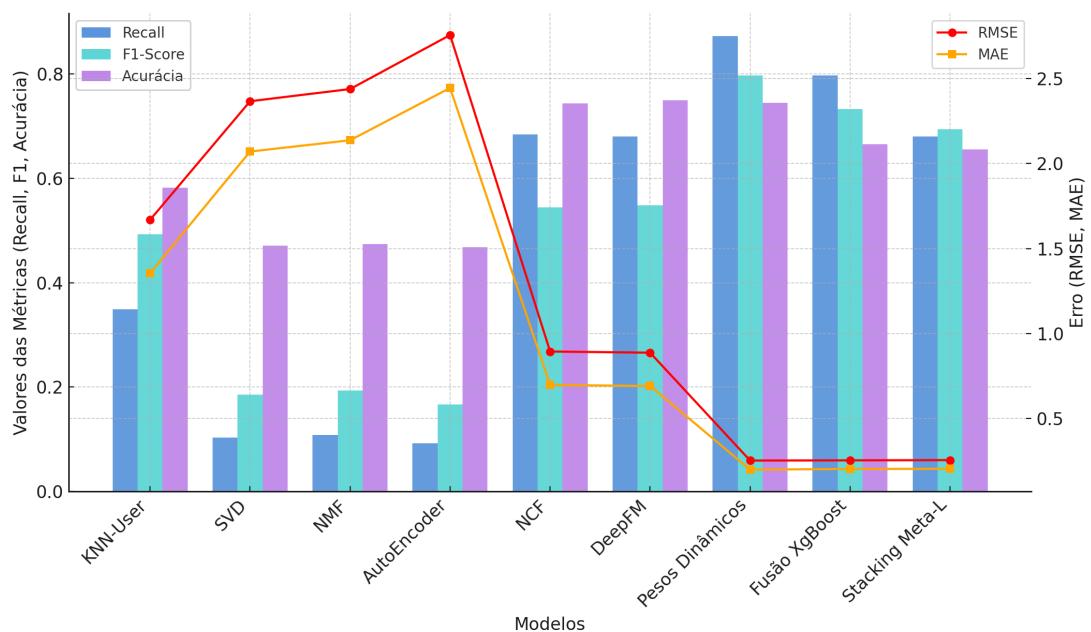
Tabela 4.15: Resultados comparativos do Cenário C com um milhão de avaliações

Modelo	Precisão	Recall	F1-S.	Acurácia	RMSE	MAE
KNN-User	0.8388	0.3488	0.4927	0.582	1.669	1.3548
SVD	0.8955	0.1031	0.1849	0.471	2.3663	2.0706
NMF	0.9000	0.1082	0.1933	0.474	2.438	2.1369
AutoEncoder	0.8437	0.0921	0.1661	0.468	2.7556	2.4445
NCF	0.4522	0.6838	0.5444	0.7438	0.8947	0.6979
DeepFM	0.4599	0.6805	0.5488	0.7495	0.8877	0.6920
Pesos Dinâmicos	0.7339	0.8727	0.7973	0.7445	0.2535	0.2002
Fusão XgBoost	0.6784	0.7973	0.7331	0.6657	0.2547	0.2037
Stacking Meta-L	0.7093	0.6799	0.6943	0.6553	0.2561	0.2050

Fonte: Elaborado pelo autor (2025)

A Figura 4.14 e Tabela 4.15, demonstram visualmente que à medida que os valores de *Recall*, *F1-Score* e Acurácia aumentam e se mantêm equilibrados, observa-se uma redução nos erros (RMSE e MAE). Esse comportamento indica uma relação inversamente proporcional entre o desempenho dos modelos e os erros na predição. Nesse sentido, é observado que nesta etapa, dentre os modelos analisados, as estratégias de Pesos Dinâmicos, Fusão *XgBoost* e *Stacking Meta-Learning* detém os melhores desempenhos, isto é, são capazes de efetuar recomendações relevantes e simultaneamente com menores taxas de erro.

Figura 4.14: Comparação de *Recall*, *F1-Score*, Acurácia, RMSE e MAE



Fonte: Elaborado pelo autor (2025)

## 4.6 CONSIDERAÇÕES FINAIS

Neste capítulo foram descritos todos os processos e etapas que compõe a validação do sistema de recomendação proposto neste trabalho. Desse modo, foram descritos cada um dos cenários utilizados, os resultados obtidos, bem como a interpretação dos mesmos.

Por fim, foi possível constatar que as abordagens propostas para a unificação dos módulos *content-based* e *collaborative filtering* desenvolvidos são capazes de realizar recomendações que atendem aos principais parâmetros de validação presentes na literatura científica. Além disso, por meio da análise dos dados comparativos dos modelos propostos com relação a outras abordagens de sistemas de recomendação, constatou-se que as estratégias de utilização dos módulos *content-based* e *collaborative filtering* com Pesos Dinâmicos, Fusão *XgBoost* e *Stacking Meta-Learning* são eficientes para a geração de previsões com alto grau de relevância, diversidade e ainda com menores taxas de erro. Além disso, a unificação dos resultados, demonstra que os métodos *content-based* e *collaborative filtering* propostas, possuem melhor desempenho em conjunto do que quando analisados isoladamente.

## 5 CONCLUSÕES E PERSPECTIVAS FUTURAS

### 5.1 CONSIDERAÇÕES INICIAIS

Este capítulo é estruturado para apresentar as conclusões finais deste trabalho, logo é composto pela Seção 5.2 que detalha as contribuições efetivamente obtidas com os resultados avaliados, já a Seção 5.3 sugere a continuidade do desenvolvimento deste sistema de recomendação com perspectivas e desafios futuros e, por fim, a Seção 5.5 apresenta a conclusão deste projeto.

### 5.2 CONTRIBUIÇÕES DO TRABALHO

A quantidade de recursos que são oferecidos e consumidos digitalmente através da Internet, aumenta constantemente, envolvendo desde informações de trabalho, lazer, estudos ou simplesmente para comunicação. Neste contexto, é notada uma sobrecarga de informações que são consideradas prejudiciais em relação à experiência dos usuários destes recursos e dos próprios empreendimentos que não são capazes de direcionar os conteúdos que de fato importam aos seus clientes.

Desse modo, justifica-se o quanto fundamental é a utilização de sistemas de recomendação para apoiar a resolução do problema de excesso de informação. Ademais, constatou-se que a utilização de técnicas clássicas de recomendação isoladamente não são eficientes em atender à demanda atual, uma vez que diversas relações implícitas ou mesmo o alto grau de obviedade de estratégias convencionais, afetam diretamente a qualidade das recomendações obtidas.

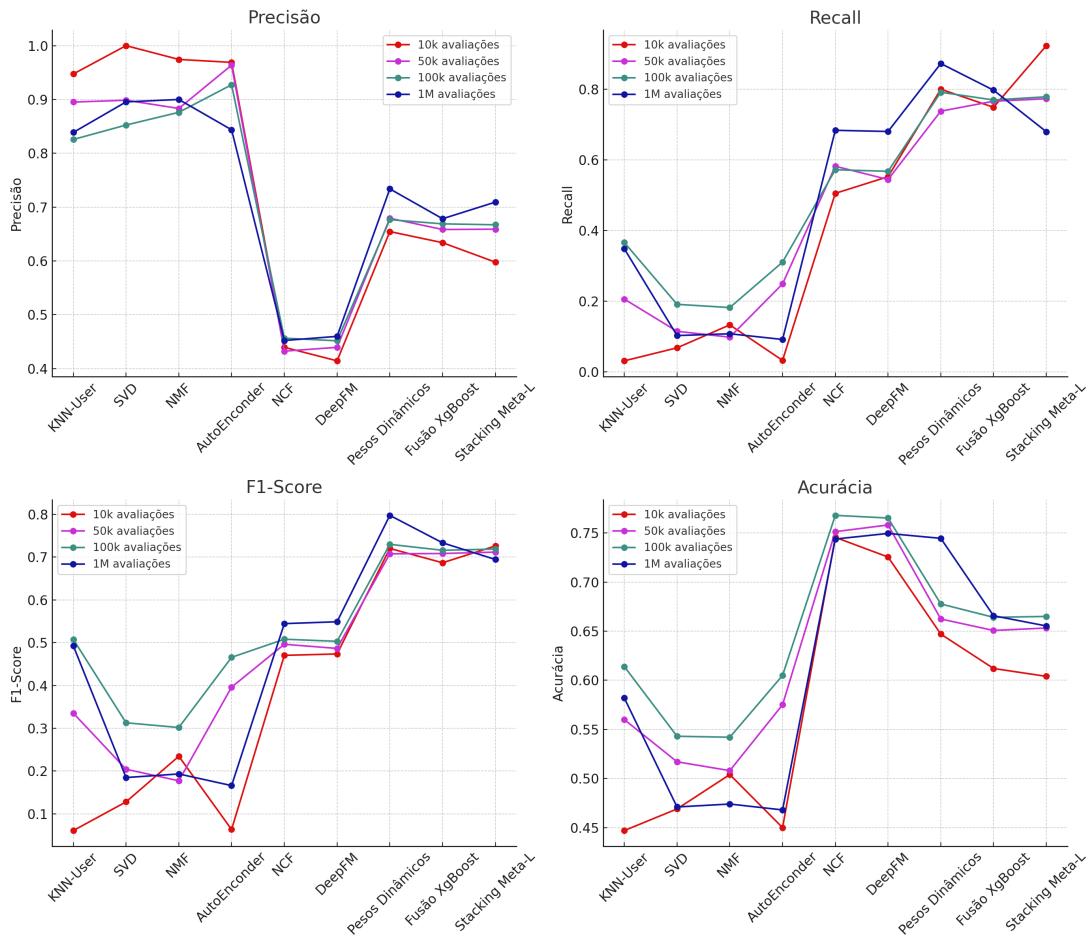
Nesse contexto, foi proposto o método híbrido de recomendação cuja base é fun-

damentada nos módulos clássicos de *content-based* e *collaborative filtering*, ambos apoiados por técnicas de *deep learning* conforme descrito detalhadamente no Capítulo 3, com o intuito de prover previsões de alta qualidade. As principais abordagens de *deep learning* utilizadas são *Multilayer Perceptrons*, *Stacked Denoising Autoencoder*, Fatoração de Matriz, *Deep Neural Network* e Redes Neurais Convolucionais, as quais devidamente organizadas para trabalharem em conjunto, demonstraram a obtenção de resultados satisfatórios em relação aos parâmetros considerados.

A unificação de resultados dos módulos desenvolvidos lançou mão de estratégias para dinamicamente considerar os resultados individuais de predição de cada um e, então efetivar a decisão de escolha da recomendação final baseada em critérios específicos de cada método. Ademais, fundamentado nos resultados providos pelo processo de validação, foi possível identificar que dentre as abordagens de unificação, houve três que se mostraram de maior relevância quanto aos parâmetros e critérios de testes, a saber, Pesos Dinâmicos, Fusão *XgBoost* e *Stacking Meta-Learning*.

Os parâmetros de precisão, *recall* e *F1-score*, quando analisados para os métodos Pesos Dinâmicos, Fusão *XgBoost* e *Stacking Meta-Learning* do sistema de recomendação proposto, indicaram que mesmo com o aumento progressivo na massa de testes utilizada, eles melhoram os resultados dos parâmetros individualmente e os mantém em equilíbrio constante, conforme a Figura 5.1. Por consequência, este comportamento analisado é um fator predominante para que estes métodos produzam recomendações de alto nível de relevância, enquanto são capazes de garantir diversidade na predição, ou seja, evitar o problema da bolha de recomendações.

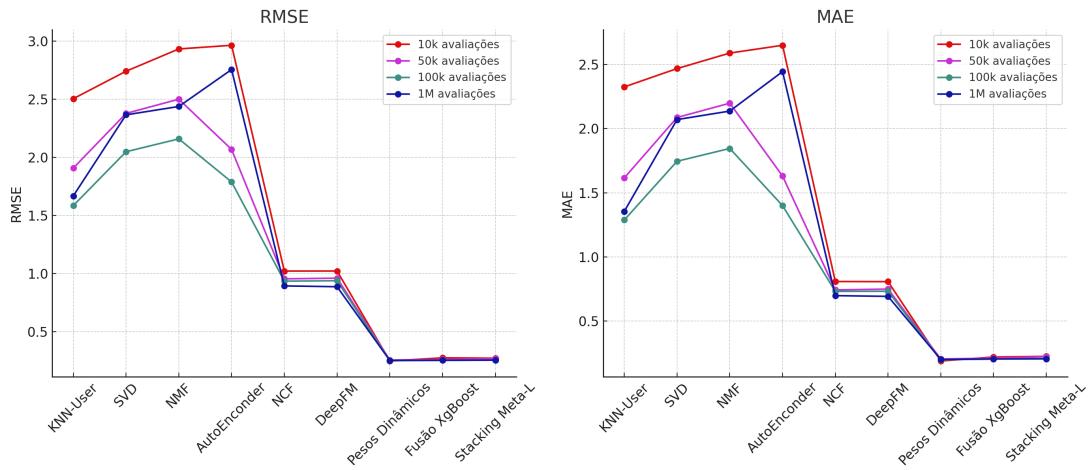
Figura 5.1: Comparação geral de Precisão, Recall, F1-Score e Acurácia



Fonte: Elaborado pelo autor (2025)

A análise dos parâmetros de erro indicados por RMSE e MAE conforme a Figura 5.2, demonstram que algoritmos tradicionais de recomendação como o KNN, SVD, NMF e *Autoencoder* apresentam altas taxas de erro e também maior instabilidade desses indicadores conforme a alteração no volume de dados. Por outro lado, o sistema de recomendação proposto se mostrou eficiente em manter-se com pequenas taxas de erro, as quais se reduzem de modo constante e progressivo mediante o aumento no volume de dados.

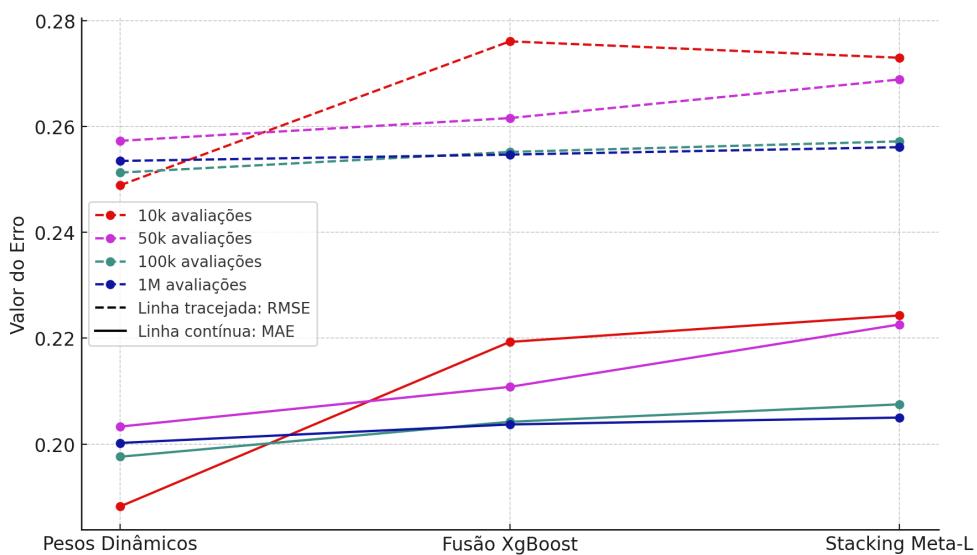
Figura 5.2: Resultados de RMSE e MAE para os conjuntos de dados de treino



Fonte: Elaborado pelo autor (2025)

Neste sentido, essa constatação reforça a capacidade do modelo proposto em lidar com o aumento crescente do volume de dados gerado na atualidade. Por conseguinte, o comportamento é caracterizado por uma inversa proporcionalidade, uma vez que é capaz de reduzir o erro conforme o aumento de dados, de acordo com a indicação gráfica da Figura 5.3.

Figura 5.3: Comportamento dos erros RMSE e MAE conforme volume de dados



Fonte: Elaborado pelo autor (2025)

Em suma, o sistema de recomendação proposto apresentou-se com maior qualidade e equilíbrio em relação aos parâmetros e condições de testes ao qual foi submetido, em detrimento a outras abordagens utilizadas como critério comparativo. Desse modo, a utilização das estratégias clássicas de sistemas de recomendação em conjunto com *deep learning*, proporcionou recomendações com maior grau de eficiência, melhor entendimento das predileções dos usuários por meio da descoberta de relações implícitas, além de prover maior diversificação nas recomendações. Logo, as previsões geradas pelo modelo proposto possuem maior eficiência e assertividade.

### 5.3 PROPOSTAS DE TRABALHOS FUTUROS

A arquitetura atual do modelo proposto neste trabalho foi implementada para execução em processos sequenciais e em CPUs convencionais. Por conseguinte, seu desempenho é impactado pelo aumento do volume de dados, o que gera um maior tempo de execução. Para otimizar esse processamento, uma abordagem alternativa seria a paralelização dos métodos, além da utilização de GPUs, que oferecem maior eficiência em operações que são fundamentais neste algoritmo, como o treinamento dos modelos.

Os modelos que implementam os módulos principais de *content-based* e *collaborative filtering* para a geração das recomendação foram treinados essencialmente com a base de dados movieLens provida pela Universidade de Minneseta. Nesse sentido, uma proposta é a utilização de outras bases de dados com diferentes volumes, a fim de validar o funcionamento do sistema e aferir as métricas utilizadas atualmente. Além disso, este sistema poderá também ser adaptado para se tornar genérico quanto à relação de usuário e qualquer tipo de itens, ou seja, ser aplicado a contextos distintos e que não se restrinja apenas à relação de usuários e filmes, de modo que também possa ser utilizado em outros segmentos e áreas.

Por fim, há a ideia de disponibilização deste sistema de recomendação por meio

de uma API que possibilite interagir com todos os mecanismos que este é capaz de prover. Desse modo, com a disponibilização de *endpoints* para treinamento dos modelos mediante a submissão dos conjuntos de dados e escolhas de parâmetros, e também para consumo das recomendações feitas em tempo real de modo simples e facilmente acessível.

#### 5.4 PUBLICAÇÕES OBTIDAS

Como resultado deste trabalho obteve-se a elaboração de um artigo intitulado "*Recommendation Systems: a Deep Learning Oriented Perspective*" que foi publicado no periódico da conferência *International Conference on Enterprise Information Systems - ICEIS 2024*, bem como apresentado durante este evento que foi sediado em Angers na França. Em observação, o ICEIS é uma conferência internacional reconhecida pela relevância acadêmica, classificada com Qualis A3 e com H5-index de 21 no *Google Scholar*.

Essa publicação denota a relevância do tema abordado neste trabalho bem como uma contribuição para a literatura científica ao abordar desafios e perspectivas para as áreas de sistemas de recomendação e inteligência computacional. Além disso, a apresentação do artigo na conferência possibilitou a disseminação dos resultados desta pesquisa para a comunidade científica, bem como promoveu discussões relevantes que possibilitaram oportunidades para futuras colaborações e novas perspectivas de trabalhos.

#### 5.5 CONCLUSÃO

O resultado conclusivo deste trabalho foi a proposição da arquitetura do modelo híbrido de recomendação que se baseia na estratégia clássica de *content-based* unificado com *collaborative-filtering*, ambas apoiadas por técnicas de *deep learning*. Nesse sentido, a concepção da arquitetura e a implementação do projeto foi alcançada, bem

como o processo de validação efetivado para a análise das métricas comparativas entre as soluções já pré-existentes em relação ao sistema proposto neste trabalho.

Neste contexto, verificou-se que a metodologia proposta é eficiente em lidar com o aumento progressivo de dados, prover recomendações com alto grau de relevância e equilíbrio, que levam em consideração as relações implícitas e não triviais entre usuários e itens. Ademais, é capaz de manter-se com baixos níveis de erro e ainda prover diversidade nos itens recomendados a fim de que evite a obviedade relacionada ao problema da bolha do filtro de recomendações. Em adição, nota-se que por meio dos resultados, foi possível constatar que a utilização unificada dos módulos propostos neste trabalho é capaz de produzir melhores resultados do que quando comparados às previsões geradas isoladamente por estes métodos.

## REFERÊNCIAS

- ALFARHOOD, M.; CHENG, J. Deepfcf: A deep learning based hybrid collaborative filtering approach for recommendation systems. *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, p. 89–96, 2018.
- ANIL, D. et al. Performance analysis of deep learning architectures for recommendation systems. In: *2018 IEEE 25th International Conference on High Performance Computing Workshops (HiPCW)*. [S.l.: s.n.], 2018. p. 129–136.
- BETRU, B. T.; ONANA, C. A.; BATCHAKUI, B. Deep learning methods on recommender system: A survey of state-of-the-art. *International Journal of Computer Applications*, Foundation of Computer Science (FCS), NY, USA, New York, USA, v. 162, n. 10, p. 17–22, Mar 2017. ISSN 0975-8887.
- BHANUSE, R.; MAL, S. A systematic review: Deep learning based e-learning recommendation system. In: . [S.l.: s.n.], 2021. p. 190–197.
- DA'U, A.; SALIM, N. Recommendation system based on deep learning methods: a systematic review and new directions. *Artificial Intelligence Review*, v. 53, p. 2709–2748, 2019.
- DELLAL-HEDJAZI, B.; ALIMAZIGHI, Z. Deep learning for recommendation systems. In: . [S.l.: s.n.], 2020. p. 90–97.
- FANG, H. et al. *Deep Learning for Sequential Recommendation: Algorithms, Influential Factors, and Evaluations*. 2020.
- FAYYAZ, Z. et al. Recommendation systems: Algorithms, challenges, metrics, and business opportunities. *Applied Sciences*, v. 10, n. 21, 2020. ISSN 2076-3417.
- GROSSETTI, Q.; MOUZA, C. du; TRAVERS, N. Community-based recommendations on twitter: Avoiding the filter bubble. In: CHENG, R. et al. (Ed.). *Web Information Systems Engineering – WISE 2019*. Cham: Springer International Publishing, 2019. p. 212–227. ISBN 978-3-030-34223-4.
- GUO, W.-w.; LIU, F. Research on collaborative filtering personalized recommendation algorithm based on deep learning optimization. In: *2019 International Conference on Robots Intelligent System (ICRIS)*. [S.l.: s.n.], 2019. p. 90–93.
- HARPER, F. M.; KONSTAN, J. A. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, Association for Computing Machinery, New York, NY, USA, v. 5, n. 4, dez. 2015. ISSN 2160-6455.
- HUANG, Z. et al. An efficient hybrid recommendation model with deep neural networks. *IEEE Access*, v. 7, p. 137900–137912, 2019.

ISINKAYE, F.; FOLAJIMI, Y.; OJOKOH, B. Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, v. 16, n. 3, p. 261–273, 2015. ISSN 1110-8665.

JAIS, I. K. M.; ISMAIL, A. R.; NISA, S. Q. Adam optimization algorithm for wide and deep neural network. *Knowl. Eng. Data Sci.*, v. 2, p. 41–46, 2019.

KINGMA, D. P.; BA, J. *Adam: A Method for Stochastic Optimization*. 2017.

LAMPA, I.; GOMES, V.; ZAFALON, G. Recommendation systems: A deep learning oriented perspective. In: INSTICC. *Proceedings of the 26th International Conference on Enterprise Information Systems - Volume 1: ICEIS*. [S.l.]: SciTePress, 2024. p. 682–689. ISBN 978-989-758-692-7.

LIU, F. et al. Research on personalized recommendation model and algorithm in deep learning mode. In: *2020 International Conference on Robots Intelligent System (ICRIS)*. [S.l.: s.n.], 2020. p. 116–119.

NEGI, R.; PATIL, A. B. Deep collaborative filtering based recommendation system. In: *12th International Conference on Computing Communication and Networking Technologies, ICCCNT 2021, Kharagpur, India, July 6-8, 2021*. [S.l.]: IEEE, 2021. p. 1–5. ISBN 978-1-7281-8595-8.

OBEIDAT, R.; DUWAIRI, R.; AL-AIAD, A. A collaborative recommendation system for online courses recommendations. In: . [S.l.: s.n.], 2019. p. 49–54.

PENG, M. et al. Comparing recommendation algorithms in session-based e-commerce sites. In: *2020 International Conference on Intelligent Computing, Automation and Systems (ICICAS)*. [S.l.: s.n.], 2020. p. 377–380.

PETTER., S.; JABLONSKI., S. Recommender systems in business process management: A systematic literature review. In: INSTICC. *Proceedings of the 25th International Conference on Enterprise Information Systems - Volume 2: ICEIS*. [S.l.]: SciTePress, 2023. p. 431–442. ISBN 978-989-758-648-4. ISSN 2184-4992.

RAWAT, R.; TOMAR, V.; KUMAR, V. An embedding-based deep learning approach for movie recommendation. In: . [S.l.: s.n.], 2020. p. 1145–1150.

SAAT, N.; NOAH, S. A. M.; MOHD, M. Towards serendipity for content-based recommender systems. *International Journal on Advanced Science, Engineering and Information Technology*, v. 8, p. 1762, 09 2018.

SABIRI., B.; EL ASRI., B.; RHANOUI., M. Mechanism of overfitting avoidance techniques for training deep neural networks. In: INSTICC. *Proceedings of the 24th International Conference on Enterprise Information Systems - Volume 1: ICEIS*. [S.l.]: SciTePress, 2022. p. 418–427. ISBN 978-989-758-569-2. ISSN 2184-4992.

SARKER, M. R. I.; MATIN, A. A hybrid collaborative recommendation system based on matrix factorization and deep neural network. In: *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*. [S.l.: s.n.], 2021. p. 371–374.

XU, G. et al. Physical examination package intelligent recommendation system based on deep learning. In: *2021 6th International Conference on Computational Intelligence and Applications (ICCIA)*. [S.l.: s.n.], 2021. p. 21–25.

YUYAN, Z.; XIAYAO, S.; YONG, L. A novel movie recommendation system based on deep reinforcement learning with prioritized experience replay. In: *2019 IEEE 19th International Conference on Communication Technology (ICCT)*. [S.l.: s.n.], 2019. p. 1496–1500.

ZHANG, S.; YAO, L.; SUN, A. Deep learning based recommender system: A survey and new perspectives. *CoRR*, abs/1707.07435, 2017.

ZHANG, S. et al. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys*, Association for Computing Machinery (ACM), v. 52, n. 1, p. 1–38, fev. 2019. ISSN 1557-7341.

ZHOU, Y. A dynamically adding information recommendation system based on deep neural networks. In: *2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS)*. [S.l.: s.n.], 2020. p. 1–4.

## DADOS CURRICULARES

IDENTIFICAÇÃO	
	Igor Luiz Lampa Data nasc. 25/01/1993
Nacionalidade	Brasileiro
Nome em citações bibliográficas	Lampa, Igor Luiz Lampa, I. L.
Curriculum Lattes	<a href="http://lattes.cnpq.br/5072793254594389">http://lattes.cnpq.br/5072793254594389</a>
ORCID	<a href="https://orcid.org/0009-0005-2099-9020">https://orcid.org/0009-0005-2099-9020</a>
Outro identificador	<a href="https://www.linkedin.com/in/igorluizlampa/">https://www.linkedin.com/in/igorluizlampa/</a>
FORMAÇÃO ACADÊMICA	
2011/2015	Bacharel em Ciência da Computação Universidade Estadual Paulista - UNESP
PRODUÇÃO BIBLIOGRÁFICA	
LAMPA, I.; GOMES, V.; ZAFALON, G. Recommendation systems: A deep learning oriented perspective. In: INSTICC. Proceedings of the 26th International Conference on Enterprise Information Systems - Volume 1: ICEIS SciTePress, 2024, Angers, France. p. 682–689.	
Lampa, I., Godoi Contessoto, A., Rici Amorim, A., Zafalon, G. F. D., Valêncio, C., Souza, R. Project Scope Management: A strategy oriented to the Requirements Engineering. In: Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS), 2017, Porto, Portugal. p. 370-378.	
PARTICIPAÇÃO EM EVENTOS CIENTÍFICOS	
26th International Conference on Enterprise Information Systems - ICEIS, 2024, Angers, France. Recommendation systems: A deep learning oriented perspective. 2024.	
XII Workshop do Programa de Pós-Graduação em Ciência da Computação da Unesp - WPPGCC 2023, 2023, Sistemas de Recomendação: Uma perspectiva orientada ao Deep Learning. 2023.	