CURSO DE SISTEMAS DE INFORMAÇÃO

ANÁLISE DE TÉCNICAS PARA DESENVOLVIMENTO DE SISTEMAS DE RECOMENDAÇÃO

MATHEUS GERMANO MORAIS PIRES

Palmas - TO

2023

CURSO DE SISTEMAS DE INFORMAÇÃO

ANÁLISE DE TÉCNICAS PARA DESENVOLVIMENTO DE SISTEMAS DE RECOMENDAÇÃO

MATHEUS GERMANO MORAIS PIRES

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de Informação da Universidade Estadual do Tocantins - UNITINS, como parte dos requisitos para a obtenção do grau de Bacharel em Sistemas de Informação, sob a orientação do professor Me. Marco Antônio Firmino de Sousa.

Palmas - TO

2023

MATHEUS GERMANO MORAIS PIRES

ANÁLISE DE TÉCNICAS PARA DESENVOLVIMENTO DE SISTEMAS DE RECOMENDAÇÃO

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de Informação da Universidade Estadual do Tocantins - UNITINS, como parte dos requisitos para a obtenção do grau de Bacharel em Sistemas de Informação, sob orientação do professor Me. Marco Antônio Firmino de Sousa.

Aprovado pela Banca examinadora em 24 de Junho de 2023

Prof. Me. Marco Antônio Firmino de Sousa

Profa. Me. Tamirys Virgulino Ribeiro Prado

Prof. Me. Douglas Chagas da Silva

Palmas, 28 de Junho de 2023

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da Universidade Estadual do Tocantins

P667a

PIRES, Matheus Germano Morais

Análise de técnicas para desenvolvimento de sistemas de recomendação. Matheus Germano Morais Pires. - Palmas, TO, 2023

Monografia Graduação - Universidade Estadual do Tocantins – Câmpus Universitário de Palmas - Curso de Sistemas de Informação, 2023.

Orientador: Marco Antonio Firmino de Sousa

1. técnicas sistemas recomendação. 2. sistemas de recomendação. 3. métricas sistemas recomendação. 4. filtragem colaborativa.

CDD 610.7

Elaborado pelo sistema de geração automatica de ficha catalográfica da UNITINS com os dados fornecidos pelo(a) autor(a).

ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO DO CURSO DE SISTEMAS DE INFORMAÇÃO DA UNIVERSIDADE ESTADUAL DO TOCANTINS - UNITINS

Aos **24** dias do mês de **Junho** de **2023**, reuniu-se em sessão não presencial, às **09:15 horas**, por meio do Google Meet sob a Coordenação do Professor **Marco Antonio Firmino de Sousa** a banca examinadora de Trabalho de Conclusão de Curso em Sistemas de Informação, composta pelos examinadores Professor **Marco Antonio Firmino de Sousa** (Orientador), Professor **Douglas Chagas da Silva** e Professora **Tamirys Virgulino Ribeiro Prado**, para avaliação da defesa do trabalho intitulado **"Análise de técnicas para desenvolvimento de sistemas de recomendação"** do acadêmico **Matheus Germano Morais Pires** como requisito para aprovação na disciplina Trabalho de Conclusão de Curso II (TCC II). Após exposição do trabalho realizado pelo acadêmico e arguição pelos Examinadores da banca, em conformidade com o disposto no Regulamento de Trabalho de Conclusão de Curso em Sistemas de Informação, a banca atribuiu a pontuação: 8,5.

Sendo, portanto, o Acadêmico: (X) Aprovado ( ) Reprovado

Assinam esta Ata digitalmente:

Professor Orientador: Marco Antonio Firmino de Sousa

Examinador: Douglas Chagas da Silva

Examinador: Tamirys Virgulino Ribeiro Prado

Marco Antonio Firmino de Sousa

Presidente da Banca Examinadora

Coordenação do Curso de Sistemas de Informação

Este trabalho é dedicado à minha mãe, que sempre foi minha maior apoiadora e sempre quis me ver formado, para todas as pessoas que possuem curiosidade em torno da área da inteligência artificial, e para mim mesmo, Matheus Germano, pois só eu sei a dificuldade colossal que senti ao longo da graduação para que eu concluísse esse trabalho.

Agradecimentos

Agradeço excepcionalmente aos meus familiares, pois se cheguei até aqui, se pude entrar numa universidade e seguir até o fim do curso, é porque tive extensamente o apoio deles nos momentos mais importantes.

Agradeço aos meus colegas de curso que ao longo do tempo se tornaram amigos e sempre prestaram apoio quando precisei. Agradeço especialmente aos meus colegas de curso Gabriel Ferreira, Denis Sousa e Rick Camelo, que eventualmente se tornaram meus amigos muito além dos portões da faculdade. Amizades essas que carregarei sempre com muito carinho e atenção.

Agradeço aos meus amigos que não são do curso mas que sempre deram apoio moral para que eu encerrasse essa jornada de uma vez por todas e me cobravam a entrega do TCC com a melhor das intenções.

Agradeço a cada um dos professores que conheci ao longo do curso. Cada um de vocês, sem exceções, deixaram boas marcas no meu aprendizado. Obrigado a cada um pela paciência e pelos ensinamentos. Um agradecimento especial ao professor Marco Antonio Firmino por ter me aceito como orientando e dar apoio nessa fase final do curso. Você é demais, cara!

Por fim, agradeço de coração a Unitins por sempre ter me recebido tão bem e me preparado como pessoa e principalmente como profissional da área da computação. Essa instituição é parte importante da minha vida, sempre que puder irei revisitá-la com muito carinho, pois ela cresceu junto comigo e foi parte importante da minha vida. Quem sabe um dia estaremos juntos novamente, mas dessa vez voltarei como professor.

Resumo

Ao longo dos anos, a necessidade de recomendar coisas, sejam elas itens, filmes ou pessoas em redes sociais, tornou-se cada vez mais importante a ponto de se perceber que tais recomendações precisam ser personalizadas. Dado o fato de que existem várias técnicas focadas na formulação de mecanismos de recomendação, como machine learning, redes neurais aplicada à pesquisa textual, filtragem colaborativa e entre outras, há a necessidade de pesquisar as técnicas em evidência para um melhor entendimento do assunto. Além disso, existem também a necessidade da compreensão de métricas para avaliar e medir a qualidade das recomendações geradas pelos sistemas de recomendação. O objetivo deste trabalho é examinar as técnicas e métricas em evidência para o desenvolvimento e avaliação de sistemas de recomendação e apresentar aquelas que têm sido mais utilizadas. A abordagem utilizada envolve a comparação de métodos e métricas típicas para entender quando utilizá-los. Em conclusão, constata-se que a escolha de uma técnica depende do domínio ao qual ela é aplicada e que novas técnicas estão integrando a vanguarda das técnicas de sistemas de recomendação. Além disso, foi verificado que métricas são necessárias para avaliar o desempenho do sistema de recomendação.

Palavras-chaves: sistemas de recomendação, filtragem colaborativa, redes neurais, deep neural networks, filtragem baseada em conteúdo, métricas para sistemas de recomendação, recuperação de informação.

Abstract

Over the years, the need for personalized recommendations, whether for items, movies, or individuals on social media, has become increasingly important. As a result, numerous services have emerged to meet this demand, accompanied by a wide range of techniques aimed at achieving accurate recommendations. In light of this, this study aims to present popular techniques in the context of developing recommendation systems, providing beginners in the field with guidance. This work explores diverse techniques surrounding recommendation systems, considering the growing importance of tailored recommendations. The emergence of various services and the development of precise recommendation techniques reflect the need to address this evolving landscape. By examining these techniques, this study offers insights to newcomers in the field, providing them with valuable direction and knowledge.

Key-words: recommendation systems, collaborative filtering, neural networs, deep neural networks, content-based filtering, matrix factorization, information retrieving.

Sumário

1 Introdução

A recomendação de itens tem se tornado uma área de estudo cada vez mais relevante à medida que a quantidade de informações disponíveis cresce exponencialmente. A capacidade de fornecer sugestões personalizadas e relevantes aos usuários tornou-se essencial em diversas aplicações, desde sistemas de recomendação de filmes e produtos em e-commerce até recomendações de conexões sociais em redes sociais. Nesse contexto, a filtragem colaborativa tem se mostrado um método eficaz para desenvolver sistemas de recomendação precisos e personalizados (AGGARWAL, 2016).

Este trabalho tem como objetivo apresentar um estudo abrangente dos métodos utilizados no desenvolvimento de sistemas de recomendação por meio da filtragem colaborativa. Serão explorados diversos tópicos relacionados a esse campo, abrangendo desde conceitos fundamentais de machine learning até a aplicação de redes neurais na pesquisa neural. Além disso, serão discutidas técnicas avançadas, como deep neural networks, collaborative filtering, content-based filtering, matrix factorization e machine learning.

Por meio deste estudo, busca-se fornecer um panorama completo dos métodos e técnicas utilizados no desenvolvimento de sistemas de recomendação por meio da filtragem colaborativa. Espera-se que esse trabalho seja uma fonte valiosa de informações e referência para pesquisadores, profissionais e iniciantes no campo, permitindo uma compreensão aprofundada dos fundamentos e das abordagens avançadas nessa área.

1.1 Objetivos

1.1.1 Objetivo geral

Revisar a literatura a respeito dos métodos e métricas populares para o desenvolvimento de sistemas de recomendação.

1.1.2 Objetivos específicos

- Realizar um levantamento bibliográfico das técnicas mais utilizadas para o desenvolvimento de sistemas de recomendação;
- Explorar os métodos de filtragem colaborativa, passando pelas técnicas baseadas em usuário e item;
- Abordar as métricas mais utilizadas em sistemas de recomendação;
- Apresentar os problemas mais frequentes em sistemas de recomendação;
- Comparar e analisar as vantagens, desvantagens e aplicabilidades das diferentes técnicas apresentadas.

1.2 Justificativa

A área de sistemas de recomendação tem ganhado cada vez mais importância devido à crescente demanda por recomendações personalizadas em diversos domínios, como comércio eletrônico, mídia, redes sociais e serviços online. De acordo com (AGGARWAL, 2016), a filtragem colaborativa é um dos métodos

mais populares e eficazes para desenvolver sistemas de recomendação, pois utiliza a experiência coletiva dos usuários para fornecer sugestões relevantes.

No entanto, o campo dos sistemas de recomendação é vasto e dinâmico, com uma variedade de técnicas e abordagens disponíveis. Diante dessa diversidade, é fundamental realizar um levantamento abrangente acerca dos métodos mais populares para lidar com sistemas desse tipo, a fim de fornecer um direcionamento claro para pesquisadores e profissionais iniciantes na área.

A justificativa para esse estudo reside na necessidade de fornecer uma visão abrangente dos métodos evidentes e relevantes no desenvolvimento de sistemas de recomendação. Com essa análise abrangente, os pesquisadores e profissionais poderão entender melhor as diferentes abordagens, suas vantagens e desafios, e tomar decisões informadas na escolha e implementação dos sistemas de recomendação.

2 Referencial Teórico

Neste capítulo serão apresentados os conceitos que estão envolvidos no desenvolvimento de sistemas de recomendação, descrevendo em tópicos e subtópicos os conceitos, técnicas e métricas que são evidentes no contexto de sistemas de recomendação. No referencial, nós abordaremos o machine learning, suas técnicas relacionadas, técnicas estabelecidas para geração de recomendação e métricas utilizadas para avaliar o desempenho das recomendações.

2.1 Machine Learning

O machine learning (ML) utiliza de recursos computacionais para simular o aprendizado humano ao passo que permite que computadores adquiram conhecimento proveniente do mundo real e melhore determinadas habilidades baseado nesse conhecimento adquirido (PORTUGAL; ALENCAR; COWAN, 2018).

Os humanos aprendem naturalmente a partir da experiência, dada sua capacidade de raciocinar, ao contrário dos computadores que precisam, que dependem de algoritmos. Atualmente existe uma série de algoritmos de ML revisados na literatura (PORTUGAL; ALENCAR; COWAN, 2018).

Os algoritmos de machine learning são classificados a partir da abordagem utilizada no processo de aprendizado. Esses algoritmos possuem duas divisões mais importantes que costumam ser utilizadas na abordagem de sistemas de recomendação, sendo elas: aprendizado supervisionado e aprendizado não supervisionado (PORTUGAL; ALENCAR; COWAN, 2018).

O aprendizado supervisionado ocorre quando há dados de treinamento e conhecimento sobre eles. Neste caso, o algoritmo utiliza o aprendizado do treinamento em um cenário real (NAQA; MURPHY, 2015).

O aprendizado não supervisionado não recebe o treinamento, mas sim um conjunto de dados na qual terá que compreender por si só. Para cada uma de suas tentativas são feitos ajustes até que se obtenha resultados satisfatórios (NAQA; MURPHY, 2015).

O aprendizado semi-supervisionado ocorre quando parte do dado é rotulado e outra parte não é rotulada. Nessa situação o algoritmo aprende sobre a parte não identificada a partir da parte rotulada dos dados (NAQA; MURPHY, 2015).

O aprendizado por reforço se trata de uma abordagem que se baseia em feedback externo, dado por um humano ou até mesmo pelo ambiente (PORTUGAL; ALENCAR; COWAN, 2018).

No contexto de sistemas de recomendação os algoritmos de ML são utilizados principalmente na filtragem colaborativa, abordagens híbridas, deep learning e filtragem baseada em conteúdo, atuando como meio para extrair padrões e extrair relações entre características e preferências de usuário.

2.1.1 Regressão

O método de regressão é um dos mais simples e populares algoritmos de machine learning. Trata-se de uma abordagem matemática para realizar tarefas relativas à predição que possibilita projeções reais/contínuas ou matemáticas de variáveis. De acordo com (MAULUD e ABDULAZEED, 2020), o conceito de regressão

linear foi apresentado em 1894 por Sir Francis Galton como “um teste matemático usado para avaliar e quantificar a relação entre as variáveis consideradas”.

Para que fique mais clara a aplicação da regressão, Wang et. al (2022, p. 13) propõem o seguinte cenário: imagine um cenário onde você pretende treinar um modelo capaz de avaliar o preço de um novo apartamento ou casa baseados nos dados coletados relacionados a informações e preços de uma imobiliária local.

De acordo com (MAULUD e ABDULAZEEZ, 2020), a regressão pode ser aplicada em dois cenários distintos. No primeiro cenário, as análises decorrentes da regressão normalmente são utilizadas para estimativas e palpites, no qual suas aplicações possuem maiores aplicações na área de machine learning.

No segundo cenário, a análise de regressão pode ser utilizada para determinar relações causais entre as variáveis dependentes e independentes. (MAULUD e ABDULAZEEZ, 2020) destacam que a regressão por si só revela apenas as relações entre uma variável dependente e um conjunto fixo de variáveis independentes.

Existem diferentes modelos de regressão que permitem que as variáveis independentes prevejam as variáveis dependentes. A regressão linear simples, mostrada na Equação 1, lida com uma única variável independente, permitindo a distinção da influência das variáveis independentes na interação com as variáveis dependentes. Nessa fórmula, y é a variável dependente, o valor que estamos tentando prever. Os parâmetros $\beta 0$, $\beta 1$ são os coeficientes de regressão que representam a relação e um completo modelo linear entre as entradas independentes x, resultado no valor de y. O representa a variação de y que não são explicadas pelas variáveis independentes que estão no modelo (LIM, 2019).

$$y = \beta 0 + \beta 1 + \varepsilon$$

Equação 1 - Fórmula que define a regressão linear simples. Fonte: (KUMAR et al., 2015)

A regressão linear múltipla é outro método de machine learning bastante utilizado em sistemas de recomendação em razão de sua capacidade de assumir múltiplas variáveis independentes para modelar a variável dependente (MAULUD; ABDULAZEEZ, 2020).

A Equação 2 expressa a definição matemática para esse método. Nessa equação, y é a variável dependente, $\beta 0$ representa o valor esperado de y quando todas as variáveis independentes forem 0,$\beta 1$, $\beta 2 \ldots \beta 1$ representam os coeficientes das variáveis independentes, $x1$, $x2 \ldots xm$ representam as variáveis independentes e o ε representa a variação de y.

$$y = \beta_0 + \beta_1 x_1 \ldots \beta_m x_m + \varepsilon$$

Equação 2 – Fórmula que define a regressão linear múltipla. Fonte: (MAULUD; ABDULAZEEZ, 2020).

O método de regressão linear tem muita utilidade no contexto de filtragem colaborativa, pois pode ser utilizado para identificar fatores latentes, que são características escondidas que possuem a capacidade de influenciar nas preferências de usuários e itens (FALK, 2019).

O uso da regressão linear é exemplificado por (FALK, 2019) ao associar o uso da filtragem baseada em conteúdo com a filtragem colaborativa (abordagem híbrida). A filtragem baseada em conteúdo pode ser ruim para distinguir um item entre bom e ruim, mas a filtragem colaborativa não coloca ênfase nos itens em si, mas no fato de os usuários considerarem-nos bons ou ruins. Em um cenário onde essas duas técnicas

são usadas, a regressão pode ser utilizada para encontrar o peso ideal para itens ao criar uma função que minimiza os erros entre saídas e valores atuais. Se existe a saída de duas recomendações e há a classificação de um usuário, então pode ser utilizada a regressão para calcular o peso de cada recomendação até chegar na recomendação adequada.

2.1.2 Classificação

Outra tarefa muito importante em machine learning é a classificação. A classificação se trata de um método de aprendizado de máquina supervisionado cujos modelos tentam prever o rótulo correto de um determinado dado.

No contexto de sistemas de recomendação, a classificação é bastante aplicada nos métodos de filtragem colaborativa com um método probabilístico a fim de se realizar predições. O problema de predição pode ser visto como um problema de classificação, já que essas tarefas de classificação costumam ser a de atribuir um objeto a uma das várias categorias pré-definidas (JANNACH et al., 2010).

Nesse cenário é comum ocorrer a atuação do classificador de Bayes. Como exemplo, (JANNACH et al., 2010) descreve uma situação onde se é necessária calcular o valor da avaliação para um determinado item a partir de um conjunto de avaliações de usuários. Para isso, o sistema de recomendação utiliza o teorema de Bayes para calcular a probabilidade condicional da avaliação de um determinado item e obter uma classificação ideal para que se possa gerar recomendações mais precisas.

2.1.3 Clusterização

A clusterização, também conhecida como segmentação, é uma técnica utilizada na filtragem colaborativa para otimizar o processo de recomendação. A técnica de clusterização encontra clusters (grupos) de usuários com características semelhantes, reduzindo o cálculo de similaridade entre eles. A clusterização é um algoritmo de aprendizado de máquina não-supervisionado e paramétrico, onde o parâmetro k determina o número de clusters (FALK, 2019).

Os clusters podem ser utilizados para otimizar o algoritmo, restringindo a busca por vizinhos relevantes. No entanto, a clusterização apresenta desafios, como a possibilidade de o usuário ativo estar na borda de um cluster ou de um cluster ter uma forma incomum, o que pode resultar em uma vizinhança subótima. É importante considerar essas limitações ao aplicar a clusterização na filtragem colaborativa (FALK, 2019).

Uma abordagem complementar é combinar a clusterização com algoritmos Top-N ou Threshold, estreitando a área de pesquisa com base na segmentação dos usuários e na seleção dos melhores itens. Essa cooperação pode melhorar a precisão e relevância das recomendações para os usuários.

2.2 Redes neurais na pesquisa neural

Os mecanismos de pesquisa têm desempenhado um papel crucial em vários sistemas ao longo do tempo. Inicialmente, a busca por palavras-chave era suficiente para atender às necessidades básicas dos usuários. Com os recentes avanços em inteligência artificial e deep learning, agora somos capazes de codificar qualquer tipo de dado em vetores e calcular similaridades entre esses vetores, além de classificá-los através de redes neurais. Isso significa que os usuários podem criar consultas com diferentes tipos de dados e obter resultados personalizados, adaptados às suas necessidades (WANG et al., 2022).

No contexto do deep learning, existem arquiteturas de redes neurais são bem sucedidas em tarefas de classificação de dados, reconhecimento de padrões, sistemas de recomendação, entre outras. Conforme descrito por (AGGARWAL, 2020), uma rede neural é um grafo de unidades conectadas que representam um modelo matemático de neurônios biológicos. Essas unidades, popularmente chamadas de "nó"ou "neurônio", são conectadas através de arcos unidirecionais ou bidirecionais com pesos que representam a força das conexões entre as unidades. Esses pesos representam a força das sinapses entre os neurônios, inibindo ou facilitando a passagem de sinais.

A rede neural recebe dados de entrada por meio de unidades de entrada dedicadas e produz sua saída por meio de unidades de saída dedicadas. Cada unidade pode funcionar tanto como uma unidade de entrada quanto como uma unidade de saída. As demais unidades são responsáveis pela lógica computacional, baseada no modelo matemático do neurônio biológico (AGGARWAL, 2020).

2.2.1 Tipos de dados e cenários de pesquisa

Vivemos numa era onde muita informação tem sido produzida, e indivíduos criam grandes quantidades de dados utilizando vários tipos de plataformas que trabalham com diferentes tipos de dados, indo de informações armazenadas como texto a informações no formato multimídia. Tais fenômenos culminam no big data, que segundo (ZULKARNAIN; ANSHARI, 2016), refere-se a conjuntos de dados de tamanho avançado e complexo, os quais vão além das capacidades dos softwares de dados convencionais para análise, gerenciamento, entre outros fins. Esses conjuntos de dados podem conter tanto informações estruturadas como não estruturadas, oriundos de textos, áudios, imagens, vídeos e entre outros que podem alcançar tamanhos de até petabytes.

Para (WANG et al., 2022), de modo geral, existem três tipos de dados, que são:

- Dado estruturado: refere-se a dados que são logicamente expressos e compreendidos por uma estrutura de tabela bidimensional. Esses dados podem ser facilmente organizados e armazenados em bancos de dados relacionais;
- Dado não estruturado: são dados que não possuem uma estrutura comum ou um modelo de dados pré-definido. Esses dados não são facilmente tratados por bancos de dados com estruturas bidimensionais. Exemplos de dados não estruturados incluem documentos de escritório, textos, fotos, arquivos HTML, relatórios, imagens, áudio e informações de vídeo em vários formato;
- Dado semiestruturado: é um tipo de dado que está entre o dado estruturado e o não estruturado. Esse tipo de dado inclui dados como logs, arquivos XML (Extensible Markup Language) e arquivos JSON (JavaScript Object Notation). Os dados semiestruturados não estão em conformidade com as estruturas presentes em bancos de dados relacionais, mas possuem tags significativas que podem ser usadas para separar elementos semânticos e organizar registros e campos.

Segundo (WANG et al., 2022), existem três tipos de pesquisas que podem ser considerados para aplicações, dependendo de seus níveis: web search, enterprise search e personal search.

Na web search, o motor de pesquisa primeiro indexa centenas de milhares de documentos. Os resultados da pesquisa são continuamente retornados ao usuário de modo eficiente enquanto o sistema é continuamente otimizado. Exemplos de web search são Google, Bing, Yahoo Search. No caso da enterprise search, o motor de

pesquisa indexa documentos internos da empresa para fornecer aos empregados e clientes do negócio, como se estivesse criando um índice de busca de patentes. O SoundCloud é uma plataforma onde se verifica esse tipo de pesquisa (WANG et al., 2022).

A personal search pode ser observada, por exemplo, em uma aplicação de email que possibilita que os usuários possam buscar por um histórico de emails.

De acordo com (WANG et al., 2022), é importante destacar a diferença entre pesquisa e correspondência. A pesquisa geralmente é realizada em documentos organizados em um formato não estruturado ou semiestruturado. Já a correspondência, como em uma consulta SQL, é realizada em dados estruturados, como os tabulados em um banco de dados relacional.

Considerando os diferentes tipos de dados e de pesquisa, (WANG et al., 2022) comentam o conceito de modalidade:

> Modalidade refere-se à forma da informação, como texto, imagens, vídeo e arquivos de áudio. Pesquisa de modalidade cruzada (também conhecido como cross-media search) refere-se à recuperação de amostras de diferentes modos com semântica semelhante, explorando a relação entre diferentes modalidades e empregando uma determinada amostra modal.

Para ilustrar o conceito de modalidade, podemos considerar o ato de digitar uma palavra-chave em um campo de busca em um aplicativo de email. Nesse caso, obtemos como resultado uma busca unimodal, na qual são retornados emails (ou uma lista de emails) relacionados à palavra-chave inserida.

Por outro lado, ao inserir uma palavra-chave em uma página de busca de imagens, o mecanismo de busca irá retornar imagens relevantes como resultado de uma busca cross-modal, que utiliza texto para pesquisar imagens. É importante ressaltar que a busca unimodal não se limita apenas a textos. Um exemplo disso é o aplicativo Shazam, amplamente conhecido na App Store, que é capaz de identificar músicas e fornecer seus nomes ao usuário com base em trechos de áudio como entrada. Nesse caso, o conceito de modalidade não se refere mais a texto, mas a áudio. Outro exemplo de busca unimodal é encontrado no aplicativo Pinterest, no qual uma imagem é utilizada para encontrar imagens semelhantes (WANG et al., 2022).

Diante de diversos cenários de pesquisa e da possibilidade de realizar pesquisas combinando múltiplas modalidades, a fim de expor possibilidades (WANG et al., 2022) apresentam como exemplo a seguinte situação:

> [...] um cenário de busca em que um usuário carrega uma foto de roupa e deseja procurar por tipos de roupas semelhantes (normalmente chamamos esse tipo de aplicativo de "comprar o look") e, ao mesmo tempo, insere um parágrafo que descreve as roupas no caixa de pesquisa para melhorar a precisão da pesquisa. Desta forma, nossas palavras-chave de busca abrangem duas modalidades (texto e imagens). Referimo-nos a esse cenário de pesquisa como uma pesquisa multimodal.

2.2.2 Machine learning em pesquisas

A tecnologia de machine learning desempenha um papel significativo em diversos aspectos da sociedade moderna. Conforme destacado por (LECUN, BENGIO e HINTON, 2015), o machine learning já está presente em diversas áreas, desde pesquisas para filtragem de conteúdo em redes sociais até recomendações de e-commerce. Além disso, pode ser encontrada em dispositivos como celulares e câmeras. Os sistemas de machine learning são utilizados para identificar objetos em imagens, transcrever discursos em texto, fazer combinações de itens, entre outras aplicações.

De acordo com (WANG et al., 2022), o aprendizado de máquina refere-se a uma técnica que permite que os computadores tomem decisões de forma similar aos seres humanos, compreendendo as leis subjacentes aos dados e adquirindo conhecimento e experiência. Essa abordagem visa capacitar os computadores a aprenderem e se aprimorarem com base em conjuntos de dados, a fim de tomar decisões informadas e realizar tarefas complexas. (LECUN, BENGIO e HINTON, 2015, p. 1) também afirmam que

> Técnicas convencionais de machine learning foram limitadas em sua habilidade de processar dados naturais em sua forma crua. Por décadas, construir um reconhecimento de padrões ou sistema de machine learning exigiu cuidadosa engenharia e considerável expertise do domínio para desenvolver um extrator de características que transformasse o dado cru (tais como valores de pixel de uma imagem) em uma representação interna adequada ou vetor de características a partir do sistema de aprendizado, frequentemente um classificador, poderia detectar ou classificar padrões na entrada.

Devido à crescente demanda por dados, as empresas estão cada vez mais buscando aprimorar seus processos de extração e análise de dados. Isso tem levado ao surgimento de uma ampla gama de algoritmos de machine learning. Um desses conceitos é o statistical machine learning, que envolve a aplicação de métodos estatísticos e matemáticos para resolver problemas de otimização (WANG et al., 2022).

Por sua vez, o deep learning é composto por métodos de representation learning. Conforme explicado por (LECUN, BENGIO e HINTON, 2015), esse conjunto de métodos permite que uma máquina seja alimentada com dados brutos e, automaticamente, descubra as representações necessárias para a detecção ou classificação de informações. Essa abordagem é especialmente útil para lidar com conjuntos de dados complexos e de grande escala. (LECUN, BENGIO e HINTON, 2015) explicam que

> métodos de deep learning são métodos de representation learning com vários níveis de representação, obtidos pela composição de módulos simples, mas não lineares, que transformam a representação em um nível (começando com a entrada bruta) em uma representação em um nível mais alto e um pouco mais abstrato. Com a composição de tais transformações, funções muito complexas podem ser aprendidas. Para tarefas de classificação, camadas superiores de representação amplificam aspectos da entrada que são importantes para a discriminação e suprimem variações irrelevantes.

Para exemplificar o funcionamento do deep learning (LECUN, BENGIO e HINTON, 2015) abordam o cenário em que uma imagem é representada por um array de valores de pixel. Nas primeiras camadas de um modelo de deep learning, as características aprendidas normalmente representam a presença ou ausência de bordas em orientações e locais específicos da imagem.

A segunda camada do modelo é responsável por detectar motifs, ou seja, padrões específicos observando arranjos particulares de arestas, independentemente de pequenas variações nas posições das arestas. Em seguida, a terceira camada combina esses motifs em combinações maiores que correspondem a partes familiares do objeto.

Conforme avançamos para camadas subsequentes, o modelo de deep learning é capaz de detectar objetos como combinações dessas partes. O aspecto fundamental do deep learning, destacado por (LECUN, BENGIO e HINTON, 2015), é que essas camadas de características não são desenvolvidas por humanos, mas sim aprendidas a partir dos dados utilizando um procedimento de aprendizado de propósito geral. Isso permite que o modelo extraia automaticamente informações relevantes dos dados, sem a necessidade de intervenção manual.

(SHINDE e SHAH, 2018) consideram o deep learning como um subconjunto do machine learning devido ao fato de ser uma rede neural com um grande número de camadas e parâmetros. Essa abordagem,

conhecida como deep neural networks, é amplamente utilizada em métodos de deep learning, já que a maioria desses métodos utiliza arquiteturas baseadas em redes neurais. Por fim, (SHINDE e SHAH, 2018) afirmam que

> [...] o deep learning usa uma cascata de várias camadas de unidades de processamento não linear para extração e transformação de recursos. As camadas inferiores próximas à entrada de dados aprendem recursos simples, enquanto as camadas superiores aprendem recursos mais complexos derivados dos recursos da camada inferior. A arquitetura forma uma representação hierárquica e poderosa de recursos. Isso significa que o aprendizado profundo é adequado para analisar e extrair conhecimento útil de grandes quantidades de dados e dados coletados de diferentes fontes.

O machine learning é amplamente aplicado em áreas como indústria, saúde, finanças e ciência e sistemas de recomendação. Suas aplicações incluem otimização de processos, diagnóstico médico, análise financeira e descoberta científica, entre outros. No caso de sistemas de recomendação, vale destacar que costuma atuar na predição, extração de características, avaliação de modelos, entre outras tarefas (AGGARWAL, 2016).

No contexto deste trabalho, o machine learning destaca-se nas aplicações de sistemas de recomendação, utilizando o estudo de exemplos para realizar previsões e recomendar itens antes mesmo de sermos conscientes do nosso interesse por eles.

2.3 Deep Learning

O deep learning é uma técnica de machine learning que se destaca por sua capacidade de aprender representações úteis de dados de forma autônoma, usando redes neurais e sendo eficaz em lidar com grandes volumes de dados. Isso é feito através de várias camadas que processam os dados que realizam várias transformações lineares e não lineares a fim de modelar conceitos gerais em dados (SERRANO, 2018).

De acordo com (AGGARWAL, 2018), o deep learning utiliza a composição repetida de funções para reduzir a quantidade de unidades computacionais necessárias para aproximar uma função específica. Isso permite o aumento do número de camadas na rede neural enquanto diminui o total de parâmetros necessários, o que resulta em um melhor poder de generalização do modelo. Essa ideia de alavancar repetidas regularidades nos padrões de dados, por meio de arquiteturas profundas, permite a generalização do aprendizado mesmo em espaços de dados sem exemplos disponíveis.

Para (AGGARWAL, 2020), o conceito de deep learning foi introduzido como uma abordagem para explorar redes neurais profundas com múltiplas camadas e um grande número de unidades. Essa arquitetura busca maximizar a extração de características não lineares presentes em cada camada, permitindo a realização de tarefas de classificação complexas. A evolução do hardware de computação de alto desempenho e o emprego de técnicas avançadas de aprendizado modular e seletivo possibilitaram que as redes neurais profundas, com milhões de pesos, alcançassem um desempenho inovador na classificação.

Uma aplicação interessante do deep learning é a arquitetura de classificação de relevância em information retrieval, que simula o processo de julgamento humano. Nessa abordagem, a detecção de contextos relevantes é realizada por meio de uma estratégia de detecção, seguida pela aplicação de uma rede de medida para determinar a relevância local usando uma Convolutional Neural Network (CNN). Em seguida, uma rede de agregação, com integração sequencial e um mecanismo de coleta de termos, é utilizada para produzir uma pontuação global de relevância (SERRANO, 2018).

A utilização de uma rede neural recorrente, com base no conceito de feedback de pseudo relevância, permite aprender um contexto profundo da lista de resultados, ajustando a lista inicialmente ranqueada. A rede neural recorrente codifica sequencialmente os melhores resultados, aprendendo um modelo de contexto local e refinando o ranqueamento dos melhores resultados (SERRANO, 2018).

2.3.1 Deep Neural Netowrk

No contexto das redes neurais, uma variante popular é a rede neural profunda (deep neural network). Essa técnica é reconhecida por sua capacidade de aprender relações complexas e não lineares entre características e rótulos relevantes.

De acordo com (ARULKUMARAN et al., 2017), uma rede neural profunda é capaz de encontrar automaticamente representações compactas de baixa dimensão (características) em dados de alta dimensão, como imagens, textos e áudio. Nesse sentido, (AGGARWAL, 2018) aponta que essa capacidade de reduzir a dimensionalidade dos dados tem sido especialmente útil para lidar com problemas que envolvem um grande número de características ou variáveis.

Conforme apontado por (AGGARWAL, 2018), redes neurais profundas aprendem padrões mais detalhados em camadas anteriores e padrões de nível mais alto em camadas posteriores. Isso reduz o número total de neurônios e a necessidade de dados para o aprendizado. No entanto, redes neurais profundas apresentam desafios como desvanecimento ou explosão do gradiente e seleção de parâmetros. Para enfrentar esses desafios, um design cuidadoso das funções nos neurônios e procedimentos de pré-treinamento são recomendados.

No contexto de sistemas de recomendação, redes neurais profundas são amplamente utilizadas para aprender uma representação de características profundas que capturam o conteúdo da informação e a associação implícita entre clientes e itens. Essas redes mapeiam recursos esparsos de alta dimensão em recursos densos de baixa dimensão, permitindo generalizar combinações ocultas de características por meio de embeddings. No entanto, é importante destacar que essa abordagem pode resultar na recomendação de itens menos relevantes quando as interações entre item e usuário são limitadas (SERRANO, 2018).

2.4 Collaborative Filtering (Filtragem Colaborativa)

Com o aumento significativo dos serviços de marketplace, streaming e pesquisas na internet, a demanda por sistemas de recomendação se tornou cada vez mais evidente. Sites como IMDB (Internet Movie Database), que serve para descoberta e avaliação de filmes, recolhem milhares de classificações de filmes feitas por usuários do site para realizar recomendações precisas para seus usuários (Lü et al., 2012).

A Amazon[1] é uma empresa que adota o modelo de cauda longa, oferecendo grande variedade de produtos de nicho. Segundo (LÜ et al., 2012), uma parcela significativa de suas vendas, que varia de 20% e 40%, é proveniente de produtos que não pertencem aos 100.000 produtos mais vendidos da loja. Em um contexto em que há uma grande diversidade de produtos e clientes, a recomendação personalizada desempenha um papel crucial ao entregar o produto adequado para a pessoa certa.

Um exemplo disso é o serviço da Netflix[2], no qual os usuários podem indicar suas preferências com apenas um clique em botões que funcionam como medidores de curtidas. Essa interação permite que o serviço

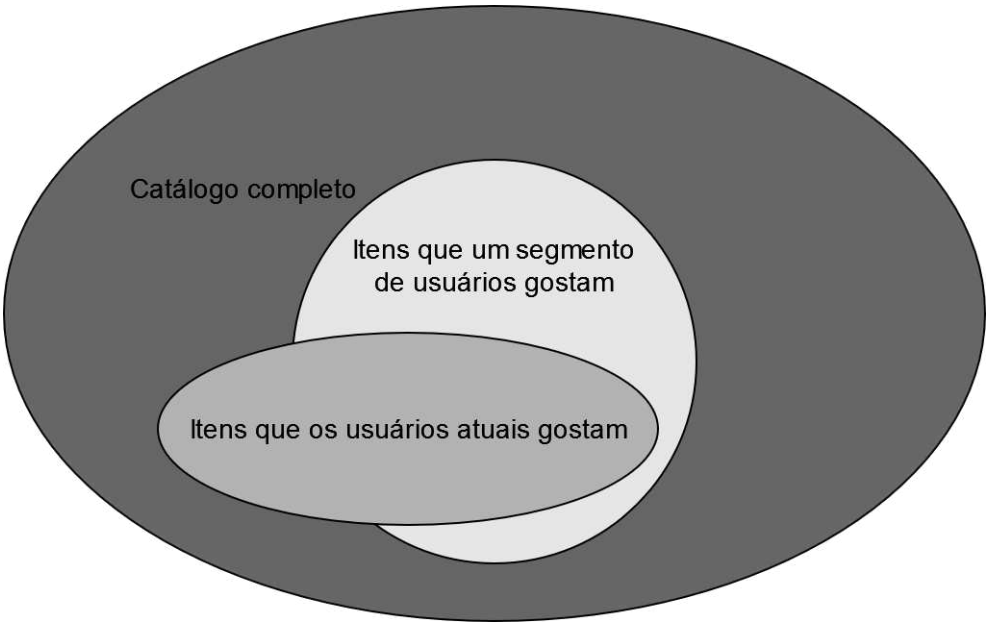
[1] <https://www.amazon.com.br/>
[2] <https://www.netflix.com/br/>

se adapte aos gostos individuais dos usuários e ofereça recomendações mais relevantes com base em suas preferências pessoais.

Dentre as metodologias comumente utilizadas, destaca-se a avaliação, na qual os usuários podem atribuir uma nota numérica (como uma avaliação com estrelas) ou indicar "likes"ou "dislikes"para um determinado conteúdo. Essa forma de avaliação auxilia no processo de personalização das recomendações, permitindo que os algoritmos identifiquem padrões e preferências individuais dos usuários (AGGARWAL, 2016).

A Figura 1 ilustra o funcionamento do processo de filtragem colaborativa. O conjunto externo representa um catálogo completo de itens, enquanto o conjunto intermediário é composto por usuários que consumiram itens semelhantes. O sistema de recomendação gera sugestões com base nesse grupo menor, pressupondo que se os usuários gostaram do mesmo item que o usuário atual, é provável que o usuário atual também goste de outros itens consumidos por esse grupo. Esse grupo é identificado pela sobreposição das preferências entre os usuários individuais e as preferências do usuário atual. Em seguida, o sistema recomenda o conteúdo que o usuário atual ainda não experimentou, ou seja, a parte do círculo intermediário que não é coberta pelo círculo que representa as preferências do usuário atual (FALK, 2019).

Figura 1 – Representação em diagrama do funcionamento da filtragem colaborativa, mostrando a intersecção de conjuntos que auxiliam no desenvolvimento de uma recomendação de item



Fonte: (FALK, 2019)

Essa abordagem visa personalizar as recomendações de acordo com as preferências individuais, proporcionando uma experiência mais relevante e agradável ao usuário.

Os algoritmos de recomendação utilizam dependências significativas na atividade entre usuário e item para fazer previsões precisas sobre o comportamento do usuário, como por exemplo, um usuário interessado em documentários históricos tende a ter interesse em outros documentários históricos ou conteúdos educacionais relacionados. Essas dependências são aprendidas por meio de abordagens orientadas por dados, usando matrizes de classificação. Quanto mais itens o usuário avaliar, melhor será a qualidade das recomendações

e a previsão de seu comportamento futuro. Modelar os interesses dos usuários com base nas características dos itens avaliados ou acessados no passado permite recomendar itens relevantes com base em padrões de comportamento semelhantes (AGGARWAL, 2016).

2.4.1 Modelos de collaborative filtering

Os modelos de filtragem colaborativa são baseados no poder colaborativo das classificações fornecidas por múltiplos usuários para fazer recomendações. Em uma aplicação comum desse modelo, como a recomendação de filmes, os usuários atribuem classificações indicando se gostaram ou não de filmes específicos, conforme mencionado por (AGGARWAL, 2016).

No entanto, é importante ressaltar que a maioria dos usuários assistiu apenas a uma pequena fração da vasta coleção de filmes disponíveis, resultando em muitas classificações não especificadas. Dentro desse contexto, as classificações fornecidas pelos usuários são chamadas de "classificações observadas", enquanto as classificações que não foram fornecidas são denominadas "não classificadas".

Em suma, o algoritmo de filtragem colaborativa parte do pressuposto de que usuários com ideias semelhantes possuem gostos semelhantes. Ou seja, quando dois usuários avaliam itens semelhantes, eles são considerados de gosto semelhante. Uma vez identificados usuários com ideias semelhantes, o sistema recomenda itens que um usuário avaliou positivamente para o outro usuário, e vice-versa (BEEL et al., 2015).

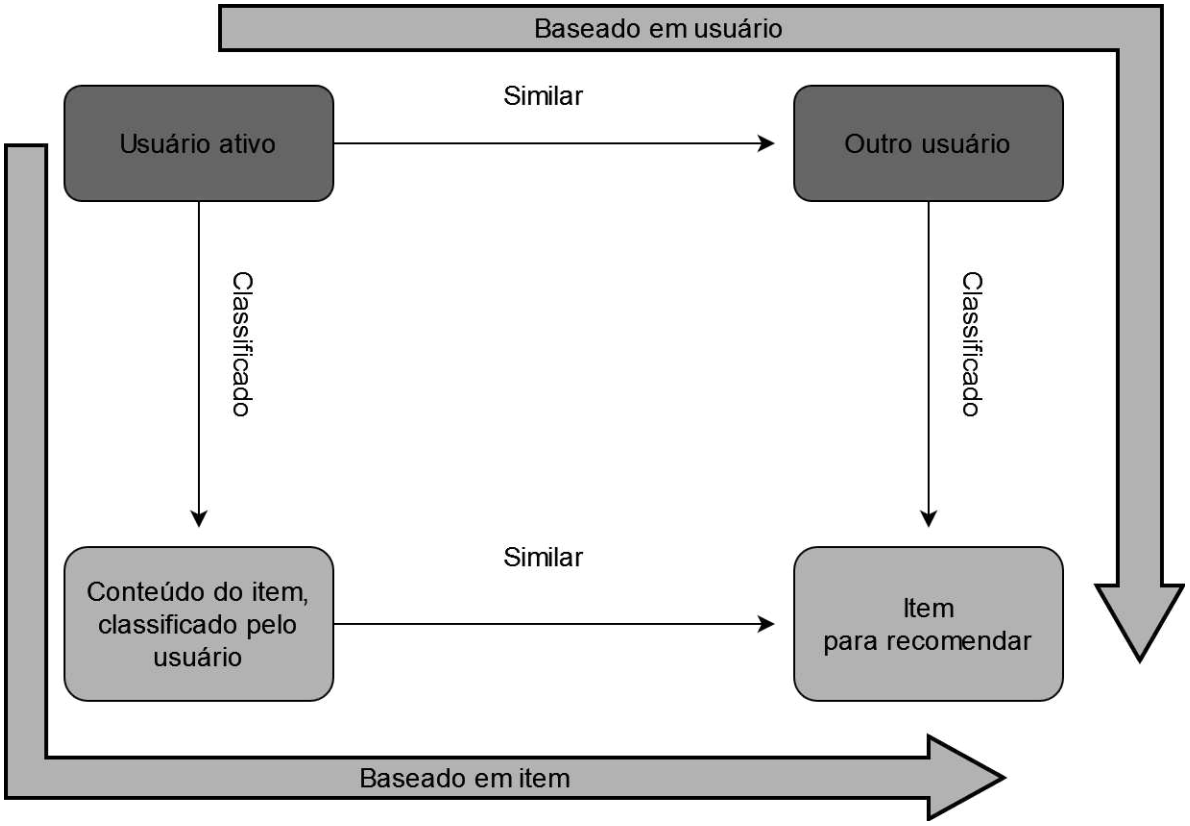
Os métodos de filtragem colaborativa podem ser divididos em duas categorias principais: memory-based methods (métodos baseados em memória) e model-based methods (métodos baseados em modelo).

2.4.2 Memory-based-methods

Memory-based methods, também conhecidos como algoritmos de filtragem colaborativa baseados em vizinhança (neighborhood-based collaborative filtering algorithms), são métodos que se baseiam no conceito de vizinhança para realizar predições. O objetivo é determinar usuários ou itens similares para realizar essas previsões. Esses métodos foram os primeiros a serem desenvolvidos, nos quais a classificação de combinações de usuário-item é prevista com base na vizinhança correspondente (AGGARWAL, 2016).

Na Figura 2 é mostrado os dois modos que a filtragem colaborativa baseada em vizinhança pode ser tratada e que serão explicados nas seguintes seções.

Figura 2 – As duas formas de realizar filtragem realizada em vizinhança. Um método usa usuários similares, enquanto o outro usa itens similares a itens que o usuário gostou



Fonte: (FALK, 2019)

2.4.2.1 User-based Collaborative Filtering

Existem duas maneiras distintas de definir a vizinhança em métodos de filtragem colaborativa. Uma delas é o User-based Collaborative Filtering, no qual as classificações fornecidas por usuários com gostos semelhantes ao usuário alvo A são utilizadas para fazer recomendações para A.

Nesse método, busca-se identificar usuários similares a A e recomendar as classificações não observadas de A, calculando médias ponderadas com base nesse grupo de pares. A premissa é que usuários similares terão interesse em itens semelhantes (AGGARWAL, 2016).

Os algoritmos user-based consistem em três etapas. Primeiramente, é necessário compreender o perfil de cada usuário para encontrar aqueles que são similares ao usuário alvo. Em seguida, calcula-se a união dos itens selecionados por esses usuários, atribuindo um peso a cada item com base em sua importância no conjunto. Por fim, seleciona-se e recomenda-se os itens com maior peso que não foram selecionados pelo usuário em questão (ALMAZRO et al., 2010).

Os algoritmos baseados em usuários, apesar de eficientes, apresentam algumas desvantagens no contexto de sistemas de recomendação. Uma delas é a esparsidade dos dados. Em plataformas como e-commerce, por exemplo, o número de usuários é geralmente grande, mas a maioria deles classifica apenas uma pequena porção dos itens disponíveis. Isso resulta em uma matriz usuário-item esparsa, com muitos elementos zerados, o que dificulta a definição de similaridade entre usuários e torna o algoritmo menos eficaz (SARWAR

et al., 2001).

Uma outra desvantagem dos sistemas de recomendação é a escalabilidade. Em conjuntos de dados extensos, torna-se impraticável encontrar grupos ideais de usuários ou itens com características semelhantes para cada usuário. A maioria dos algoritmos que calculam a similaridade entre vizinhos exigem recursos computacionais que crescem proporcionalmente ao número de usuários e itens. Isso resulta em problemas significativos de escalabilidade, especialmente em sistemas de recomendação web com milhares de usuários. (SARWAR et al., 2001).

2.4.2.2 Item-based Collaborative Filtering

Na abordagem Item-based Collaborative Filtering, a ideia é que um usuário tem maior probabilidade de comprar itens similares aos que já adquiriu no passado, fornecendo assim uma base para prever suas escolhas futuras (ALMAZRO et al., 2010).

Para realizar previsões de classificação para um determinado item alvo B pelo usuário A, o primeiro passo é determinar um conjunto S de itens que sejam mais similares ao item alvo B. As classificações no conjunto S, especificadas pelo usuário A, são utilizadas para prever se o usuário A irá gostar do item B (ALMAZRO et al., 2010).

Esse método é composto por duas etapas. Na primeira etapa, calcula-se a similaridade entre os itens. Para isso, os usuários que classificaram ambos os itens são isolados e técnicas como a similaridade de cosseno e o coeficiente de Pearson são aplicadas para determinar a similaridade $s_{i,j}$ (SARWAR et al., 2001).

Na segunda etapa, selecionam-se os itens mais semelhantes. Os itens mais relevantes identificados na primeira etapa são analisados em relação às classificações dos usuários-alvo, e uma técnica como regressão ou soma de pesos é utilizada para fazer previsões (SARWAR et al., 2001).

2.4.3 Métodos baseados em modelo (model-based)

No método baseado em modelo, são utilizados métodos de mineração de dados e machine learning. A mineração de dados refere-se ao processo de descoberta de padrões e conhecimentos relevantes a partir de grandes quantidades de dados, que podem ser provenientes de bancos de dados, armazéns de dados, a Web, entre outras fontes (Han, 2012). Por sua vez, o machine learning é empregado para criar modelos de predição. No método baseado em modelo, o processamento dos dados brutos ocorre offline, permitindo que o modelo aprendido seja posteriormente utilizado em tempo de execução para realizar predições.(JANNACH et al., 2010).

Existem diversos exemplos de modelos baseados em modelo utilizados em sistemas de recomendação, como árvores de decisão, modelos baseados em regras, métodos Bayesianos, modelos de regressão, máquinas de suporte de vetor, redes neurais e modelos de fator latente. Muitos desses métodos apresentam uma boa capacidade de lidar com matrizes de classificação esparsas, ou seja, mesmo quando há uma quantidade limitada de informações disponíveis sobre itens ou usuários, eles são capazes de gerar um maior número de recomendações a partir desse conjunto de dados, superando o desafio da esparsidade.(AGGARWAL, 2016).

A maioria desses métodos podem ser generalizados para o contexto da filtragem colaborativa, assim como os métodos classificadores baseados em vizinhança. Isso ocorre porque os problemas tradicionais de regressão e classificação são casos especiais do problema de preenchimento de matriz (AGGARWAL, 2016).

A filtragem colaborativa apresenta analogias diretas com algoritmos de machine learning e classificação, o que nos permite considerar a similaridade entre esses campos ao desenvolver algoritmos para problemas de filtragem colaborativa (AGGARWAL, 2016).

Os sistemas de recomendação baseados em algoritmos baseados em modelo oferecem vantagens significativas em relação aos algoritmos baseados em vizinhança, incluindo:

1. Eficiência de espaço: Os modelos aprendidos são geralmente mais compactos do que as matrizes de classificação, o que resulta em requisitos de espaço reduzidos. Em contraste, os algoritmos baseados em vizinhança podem exigir uma quantidade de espaço proporcional ao quadrado do número de usuários ($O(m^2)$), onde m é o número de usuários. Já os métodos baseados em itens têm uma complexidade espacial de $O(n^2)$, onde n é o número de itens (AGGARWAL, 2016);

2. Velocidade de treinamento e predição: os modelos baseados em vizinhança podem apresentar um tempo de processamento quadrático durante a fase de pré-processamento, dependendo do número de itens ou usuários. Por outro lado, os sistemas model-based tendem a ser mais rápidos nessa etapa. Muitas vezes, é possível utilizar modelos compactos e resumidos para fazer previsões de forma eficiente (AGGARWAL, 2016);

3. Prevenção de overfitting: A sumarização utilizada nos métodos model-based frequentemente auxilia na prevenção do overfitting, um problema comum em algoritmos de machine learning. Além disso, esses métodos são capazes de lidar com conjuntos de dados esparsos e permitem a incorporação de informações adicionais nos modelos, proporcionando recomendações mais precisas e personalizadas (AGGARWAL, 2016).

Essas vantagens ressaltam a eficácia e a utilidade dos sistemas de recomendação baseados em algoritmos model-based na filtragem colaborativa.

2.4.4 Considerações sobre seleção de vizinhança

A construção de uma vizinhança é um passo essencial nos algoritmos de filtragem colaborativa baseados em vizinhança. Uma vizinhança consiste em um conjunto de itens que são semelhantes ao conteúdo que um usuário está procurando. O termo "vizinhança"é utilizado porque estamos lidando com itens que possuem uma pequena distância entre si (FALK, 2019).

No entanto, calcular essa distância entre itens e usuários é uma das principais dificuldades encontradas nesses algoritmos. Isso ocorre porque o algoritmo precisa avaliar o grau de similaridade entre os usuários ativos em relação a todos os usuários ou a similaridade entre os itens em relação a todos os outros itens no sistema (FALK, 2019).

Para enfrentar esse desafio, existem algumas técnicas populares que podem ser utilizadas, tais como clusterização, Top-N e método Threshold que serão detalhadas adiante. Essas técnicas são empregadas para aprimorar os resultados e otimizar as soluções mencionadas anteriormente no contexto da filtragem colaborativa. Na próxima subseção, exploraremos essas técnicas em detalhes.

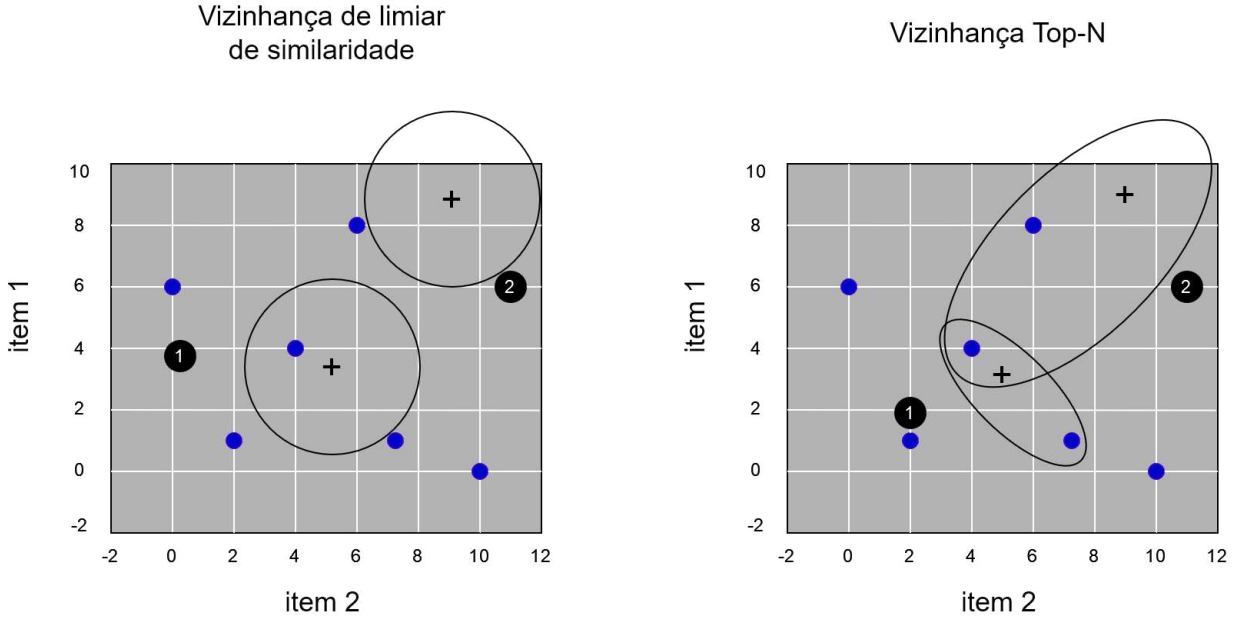
2.4.4.1 Top-N

O algoritmo Top-N é utilizado na construção de modelos de recomendação ao analisar a similaridade entre diferentes itens. O Top-N utiliza esses itens similares para identificar conjuntos de itens a serem recomendados (DESHPANDE; KARYPIS, 2004).

Uma abordagem simples para calcular a vizinhança é definir um número N como a quantidade desejada de vizinhos e considerar que todos os itens possuem N itens similares na vizinhança. No entanto, nem todos esses itens são verdadeiramente similares, o que pode comprometer a qualidade das recomendações. O método Top-N, por exemplo, pode incluir itens distantes do ponto ativo, resultando em recomendações de baixa qualidade, como é demonstrado na Figura 3. Para garantir uma melhor qualidade, é recomendado utilizar o método do threshold, que estabelece um valor mínimo de similaridade. Esse valor é utilizado para selecionar apenas os vizinhos mais relevantes, melhorando a precisão das recomendações (FALK, 2019).

Na Figura 3, no lado esquerdo é mostrado como a vizinhança de limiar de similaridade funciona (similarity threshold neighborhood). Tudo que está dentro do círculo é considerado vizinhança. No lado direito, na abordagem Top-N não ocorre procura pela distância, mas sim uma procura com base na quantidade N de vizinhos, que independe da distância.

Figura 3 – Comparação de algoritmos para encontrar a vizinhança



Fonte: (FALK, 2019)

2.4.4.2 Threshold

O método de seleção de vizinhos baseado em threshold seleciona os vizinhos que pertencem a uma determinada faixa em relação às similaridades das preferências, conforme mostrado na Figura 3. O número de vizinhos selecionados por esse método varia, pois é feita a seleção de vizinhos de acordo com um determinado limite δ. O valor de δ pode ser ajustado conforme as necessidades do recomendador (KIM; YANG,).

A escolha entre o método Top-N e o threshold deve ser feita considerando a prioridade entre qualidade e quantidade. O método do threshold é mais adequado para garantir a qualidade das recomendações, enquanto

o Top-N está mais focado em incluir um maior número de itens na vizinhança (FALK, 2019).

2.4.5 Vantagens e desvantagens de collaborative filtering

Embora os métodos de filtragem colaborativa tenham facilidades em sua implementação, ao longo do tempo foram observadas algumas desvantagens que devem ser consideradas:

- Esparsidade: a maioria dos conjuntos de dados é caracterizada por ter uma quantidade limitada de classificações em relação ao número total de itens. Isso dificulta o cálculo da vizinhança e pode resultar em recomendações limitadas apenas aos itens mais populares (FALK, 2019);
- Grey-sheep: existem usuários com preferências tão únicas e incomuns que se tornam difíceis de relacionar com outros usuários e itens. Esses usuários "grey-sheep"podem ser um desafio para os métodos de filtragem colaborativa, que dependem da similaridade entre usuários (FALK, 2019);
- Cold-start: também chamado de problema de inicialização, esse problema se refere a situação onde existem poucas classificações para se basear as recomendações. Além disso, as taxas de aprendizado são não-lineares e sua qualidade não melhora para sempre, portanto se torna um desafio ainda maior manter a qualidade de recomendações (HERLOCKER et al., 2004);
- Escalabilidade: cada vez mais dados são gerados com facilidade ao longo do tempo, e portanto, o tamanho dos conjunto de dados tem aumentado. Como resultado disso, se tornou essencial desenvolver sistemas de recomendação que podem performar bem na presença de grandes quantias de dados (AGGARWAL, 2016).

Os métodos de filtragem colaborativa oferecem várias vantagens significativas. Uma delas é a sua independência de conteúdo, o que significa que não é necessário adicionar metadados detalhados aos itens ou coletar informações específicas dos usuários. Além disso, a técnica de filtragem colaborativa é caracterizada por algoritmos simples que podem fornecer recomendações de qualidade, mesmo quando as informações sobre um item são limitadas. Esses algoritmos são de fácil implementação e explicação. A natureza colaborativa da filtragem colaborativa também é uma vantagem, pois aproveita o conhecimento coletivo de uma comunidade de usuários (AGGARWAL, 2016).

Para construir um sistema de recomendação eficiente, não é necessário ter um conhecimento especializado no domínio específico. O sistema é desenvolvido com base nas preferências e comportamentos dos usuários. No entanto, é importante que o responsável pelo desenvolvimento do sistema tenha um entendimento adequado do domínio para tomar decisões apropriadas durante o processo de implementação.

2.4.6 Métricas

O desenvolvimento de uma metodologia de avaliação é crucial para compreender a eficácia de um algoritmo de recomendação. A avaliação de um sistema de recomendação é multifacetada, e um único critério pode não ser suficiente para capturar todas as metas do método. Um projeto de avaliação experimental incorreto pode levar a uma subestimação ou superavaliação grosseira da verdadeira precisão do algoritmo ou modelo (AGGARWAL, 2016).

Existem dois principais métodos de avaliação: offline e online. Os métodos offline funcionam sem a interação do usuário em tempo real, e é um dos métodos mais utilizados. O método offline consiste em dividir os dados em dois subconjuntos: treinamento e avaliação.

O subconjunto de treinamento é composto por dados utilizados pelo algoritmo para determinar recomendações ou estimativas. Esses dados são então comparados com os dados do subconjunto de avaliação. Apesar de ser bastante utilizada, pode possuir variações significativas nos dados e modo como é conduzida (NALLAMALA et al., 2020).

Por outro lado, os métodos online são usados para avaliar o desempenho do sistema de recomendação em tempo real. Essa abordagem analisa o comportamento do usuário em relação às recomendações apresentadas em tempo real. Esses métodos também são conhecidos como testes A|B e medem o impacto direto do sistema de recomendação no usuário (AGGARWAL, 2016).

2.4.6.1 Métricas de similaridade

A fim de descobrir quais usuários que possuem gostos semelhantes é necessário realizar o cálculo da similaridade entre esses usuários ou itens para que sejam geradas recomendações adequadas. Na subseção seguinte serão descritas algumas das métricas mais utilizadas em sistemas de recomendação

2.4.6.1.1 Distância euclidiana

A distância entre dois pontos, a e b, com k dimensões é calculada por

$$\sqrt{\sum_{j=1}^{k}(a_j - b_j)^2}$$

Equação 3 - Fórmula para se calcular a distância euclidiana. Fonte: (KUMAR et al., 2015)

A distância euclidiana é sempre maior ou igual a zero. Para pontos idênticos a medida é 0. Para pontos que possuem pouca similaridade a medida é maior (KUMAR et al., 2015).

2.4.6.1.2 Correlação de Pearson

Baseia-se no quanto a classificação por usuários comuns para um par de itens desvia das classificações médias para aquele item. Considerando o conjunto de usuários que classificaram i e j são denotados por U, $\overline{R_i}$ e $\overline{R_j}$ sendo as médias das classificações dos itens i e j, a correlação de similaridade é dada por (KUMAR et al., 2015). É definida pela fórmula:

$$\text{Sim}(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \overline{R_i})(R_{u,j} - \overline{R_j})}{\sum_{u \in U}(R_{u,i} - \overline{R_i})^2 (R_{u,j} - \overline{R_j})^2}$$

Equação 4 - Fórmula para se calcular a correção de Pearson. Fonte: (KUMAR et al., 2015)

2.4.6.1.3 Similaridade do cosseno

Também conhecida como similaridade baseada em vetor, essa fórmula visualiza dois itens e suas classificações como vetores. Tendo $\|\vec{i}\|_2$ e $\|\vec{j}\|_2$ como as normas euclidianas dos vetores $\vec{i}$ e $\vec{j}$, é definida a similaridade entre eles como o ângulo entre esses vetores (KUMAR et al., 2015).

$$\text{Sim}(i,j) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\|_2 * \|\vec{j}\|_2}$$

Equação 5 - Fórmula para se calcular a similaridade de cosseno. Fonte: (KUMAR et al., 2015)

2.4.6.1.4 Mean Average Precision (MAP)

Para calcular o MAP é necessário calcular antes o AP (Average Precision). O AP é usado para medir quão bom o rank é ao executar a precisão de 1 até m, sendo m o número de itens que são recomendados e $P@k(u)$ a precisão no ponto de corte k para o item ou documento u (FALK, 2019). A fórmula do AP é:

$$AP(U) = \frac{\sum_{k=1}^{m} P@k(u)}{m}$$

Equação 6 - Fórmula para se calcular a AP. Fonte: (FALK, 2019)

Na equação 7, $\sum_{u \in U} AP(u)$ é a soma das médias de precisão para todos os usuários em $|U|$. Para obter a média da precisão média (MAP) em todas as recomendações, é utilizada a seguinte fórmula:

$$MAP = \frac{\sum_{u \in U} AP(u)}{|U|}$$

Equação 7 - Fórmula para se calcular a MAP. Fonte: (FALK, 2019)

2.4.6.1.5 Discount Cumulative Gain (DCG)

O DCG (Discounted Cumulative Gain) é uma métrica utilizada em recuperação de informação (IR) e sistemas de recomendação para avaliar a qualidade de uma lista ordenada de itens. Essa métrica atribui pesos aos itens com base em sua relevância e aplica um desconto que reduz a importância dos itens menos relevantes à medida que se desce na lista classificada (JäRVELIN; KEKäLäINEN, 2002). De forma geral, a fórmula do DCG é a seguinte:

$$DCG_{DCG_R} = \sum_{1}^{R} D_r G_r$$

Equação 8 - Fórmula para se calcular a MAP. Fonte: (FALK, 2019)

O D_r é o ganho do documento ou item relevante na posição r e G_r é o ganho alcançado pela apresentação do documento d_r no rank r.

2.4.6.1.6 Normalized Discounted Cumulative Gain (NDCG)

É uma métrica que atribui uma pontuação a qualquer lista de permutações de um conjunto de itens. Seu objetivo é garantir que a lista que coloca os itens de maior relevância nas posições mais altas obtenha a pontuação máxima.

Isso é especialmente relevante em sistemas de recomendação, onde os itens sugeridos devem corresponder aos de maior pontuação para fornecer recomendações precisas e relevantes. O NDCG leva em consideração tanto a relevância dos itens quanto a sua ordem na lista (BALAKRISHNAN; CHOPRA, 2012). A fórmula do NDCG é:

$$nDCG = \frac{DCG@k(y, \pi)}{DCG@k(y, \pi^*)}$$

Equação 9 - Fórmula para calcular o NDCG. Fonte: (BALAKRISHNAN; CHOPRA, 2012)

Nessa fórmula, $DCG@k(y, \pi$ é o DCG até a posição k para o ranking π e $DCG@k(y, \pi^*)$ é o DCG ideal até a posição k, que seria obtido se o ranking ideal fosse utilizado, onde todos os itens relevantes estivessem nas primeiras posições.

2.4.6.2 Métricas de precisão em métodos offline

As métricas a seguir servem para calcular a precisão de modelos de predição ou regressão. Em métodos offline a acurácia da predição pode ser calculada por métodos como RSME e MEA.

2.4.6.2.1 Root Mean Squared Error (RMSE)

Calcula a média da diferença entre os valores previstos e os valores reais, e coloca grandes penalização no erros, fazendo com que os grandes erros contem mais do que vários dos menores (AGGARWAL, 2016). A sua fórmula é:

$$RMSE = \sqrt{\frac{\sum_{(u,j) \in E} {}^{e\,2} uj}{|E|}}$$

Equação 10 - Fórmula para calcular o RMSE. Fonte: (AGGARWAL, 2016)

Na fórmula, (u, j) é um par de usuário-item que pertencem ao conjunto E que estão sendo percoridos durante o cálculo, E representa o conjunto de pares de valores, então $\sum_{(u,j) \in E} {}^{e\,2} uj$ representa a soma dos quadrados dos erros para todos os pares de valores (u, j) no conjunto E.

2.4.6.2.2 Mean Absolute Error (MAE)

Calcula o desvio médio entre as pontuações de recomendação computadas e os valores de avaliação reais para todos os usuários avaliados e todos os itens em seus conjuntos de teste (JANNACH et al., 2010). É calculado com a fórmula:

$$MAE = \frac{\sum_{(u,j) \in H} |(r_{uj} - \hat{r}_{uj})|}{|H|}$$

Equação 11 - Fórmula para calcular o MAE. Os índices de usuário-item são representados por u,j, o conjunto avaliado é representado por H. Fonte: (JANNACH et al., 2010).

Nessa fórmula, (u, j) representa um par de usuário (u) e item (j), que pertencem ao conjunto H. Este conjunto representa as avaliações reais e previstas pelo sistema de recomendação. A parte $|(r_{uj} - \hat{r}_{uj})|$ representa o cálculo da diferença absoluta entre a avaliação real e a prevista. O $|H|$ é o total de avaliações reais e previstas pelo sistema de recomendação.

2.4.6.3 Métricas de performance em métodos offline

De acordo com (FALK, 2019), os métodos a seguir são entendidos como estratégias de suporte à tomada de decisão em sistemas de recomendação. Esses métodos envolvem a análise individual de cada elemento, permitindo avaliar se o sistema está correto ou incorreto em relação a eles.

Ao considerar um sistema de recomendação que examina cada item e o compara com o histórico de consumo do usuário atual, podemos identificar quatro resultados diferentes. São eles:

- Verdadeiro positivo (VP): refere-se aos itens recomendados que foram efetivamente consumidos pelo usuário;
- Falso positivo (FP): ocorre quando o item foi recomendado, mas o usuário não o consumiu;
- Falso negativo (FN): representa a situação em que o recomendador não incluiu o item em uma recomendação, mas o usuário ainda assim o consumiu;
- Verdadeiro negativo (VN): indica os itens que não foram recomendados e que o usuário também não consumiu.

2.4.6.3.1 Acurácia

Essa métrica mede a quão próxima a recomendação está em relação à preferência do usuário. É uma medida fundamental para avaliar a qualidade das recomendações realizadas (KUMAR et al., 2015). A fórmula para calcular a acurácia é definida por:

$$ACC = (VP + VN)/(T + FN) * (VN + FP)$$

Equação 12 - Fórmula para calcular a acurácia. Fonte: (KUMAR et al., 2015).

2.4.6.3.2 Sensibilidade

Também conhecida como taxa de verdadeiros positivos ou taxa de recall, a sensibilidade mede a proporção atual de positivos que são corretamente identificados e é complementar à taxa de falsos negativos (KUMAR et al., 2015). É representada pela fórmula:

$$SEN = VP/(VP + FN)$$

Equação 13 - Fórmula para calcular a sensibilidade. Fonte: (KUMAR et al., 2015).

2.4.6.3.3 Precisão

É a fração de documentos recuperados que são relevantes a necessidade de informação do usuário (KUMAR et al., 2015). É calculada do seguinte modo:

$$PPV = VP/(VP + FP)$$

Equação 14 - Fórmula para calcular a precisão. Fonte: (**??**)

2.4.6.3.4 F1-score

Medida única que sumariza a precisão e recall. Fornece uma melhor quantificação do que a precisão ou recall. Por ser dependente do tamanho da lista de recomendação, não é uma representação completa do trade-off entre precisão e recall (AGGARWAL, 2016).

A fim de resumir o que cada técnica faz, a Tabela 1 contém descrições sobre cada uma das técnicas mencionadas nessa seção.

Tabela 1 – Descrição das métricas de avaliação de desempenho

Métricas	**Descrição**
Acurácia	Utilizada para compreender o quão próximo o resultado de uma recomendação corresponde a preferência de um usuário. É muito importante para avaliar a qualidade das recomendações geradas.
Sensibilidade	Mede a proporção dos positivos que são corretamente identificados. Muito útil para minimizar os falsos negativos e identificar adequadamente as recomendações relevantes ou desejáveis para um usuário.
Precisão	A precisão indica a fração de itens relevantes entre todos os itens recomendados a um usuário. Um item relevante é aquele que o usuário considera atraente.
F1-score	É uma medida que combina precisão e sensibilidade e que é capaz de exibir o comportamento tanto da precisão como da sensibilidade. É útil quando se precisa levar em conta tanto os falsos negativos quanto os falsos positivos.

Fonte: Tabela criada a partir das observações feitas por (KUMAR et al., 2015) e (FAYYAZ et al., 2020)

A Tabela 2 apresenta vantagens e desvantagens entre as métricas apresentadas nessa seção.

Tabela 2 – Apresentação das vantagens e desvantagens de métodos utilizados para criar o perfil de usuário

Nome	Desvantagens	Vantagens
Acurácia	Não leva em conta os falsos positivos, tendendo a ser problemático em situações onde falso positivos não são desejados.	Essa métrica é relevante quando evitar falsos negativos tem um alto custo, pois ela mostra a habilidade do modelo em corretamente reconhecer as ocorrências positivas.
Precisão	Não leva em conta os falsos positivos, tendendo a ser problemático em situações onde falso positivos não são desejados.	Essa métrica é relevante quando evitar falsos negativos tem um alto custo, pois ela mostra a habilidade do modelo em corretamente reconhecer as ocorrências positivas.
F1-score	Essa métrica pode não ser a melhor escolha para todos os cenários, pois pode gerar resultados desfavoráveis em situações em que é crucial ter tanto alta precisão quanto a habilidade de capturar todas as instâncias positivas corretamente.	É uma métrica balanceada que combina precisão e recall, oferecendo uma única medida de desempenho que considera tanto os falsos positivos quanto os falsos negativos.
Acurácia	Essa métrica pode levar a conclusões equivocadas sobre o desempenho do modelo quando as classes estão desbalanceadas, tornando-se enganosa em tais situações.	É uma métrica simples e fácil de entender, representando a proporção de previsões corretas em relação ao total de previsões.

Fonte: Tabela criada a partir das observações feitas por (POWERS, 2020) e (SOKOLOVA; LAPALME, 2009)

2.4.7 Métricas de similaridade e distância

As métricas de similaridade desempenham um papel essencial na filtragem colaborativa, um método amplamente usado em sistemas de recomendação. Essas métricas calculam a proximidade entre itens ou usuários, permitindo identificar padrões e fazer recomendações personalizadas.

Existem várias métricas comumente usadas, como similaridade do cosseno, similaridade de Pearson, similaridade de Jaccard e similaridade Euclidiana, cada uma adequada para diferentes tipos de dados e cenários. Ao usar essas métricas, é possível encontrar itens ou usuários semelhantes, melhorando a precisão das recomendações (KUMAR et al., 2015).

É importante distinguir entre métricas de distância, que quantificam a separação entre objetos, e métricas de similaridade, que medem o grau de semelhança. Dependendo da situação, essas métricas podem ser usadas de forma intercambiável (WANG et al., 2022).

2.4.7.1 Similaridade baseada em cosseno

Essa métrica mede a similaridade entre dois vetores n-dimensionais baseado no ângulo entre eles. Essa medida é comumente usada nos campos de information retrieval e mineração de texto para comparar dois documentos de texto, no qual os documentos são representados como vetores de termos (JANNACH et al., 2010).

A similaridade entre dois itens a e b – compreendidas como os correspondentes vetores de classificação $\vec{a}$ e $\vec{b}$ – pode ser definida como:

$$\text{sim}(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|}$$

Equação 15 - Fórmula para calcular a similaridade baseada em cosseno. Fonte: (JANNACH et al., 2010).

2.4.7.2 Coeficiente de correlação de Pearson (PCC)

O PCC é outra medida para calcular a semelhança entre dois vetores. O resultado varia de -1 a 1, que indicam que o vetor é muito semelhante. Um valor de PCC equivalente a 1 corresponde a forte similaridade.

Por outro lado, um valor próximo a -1 indica uma correlação negativa, indicando que as variáveis estão inversamente relacionadas (ALI; MAJEED, 2021). O PCC é calculado com a seguinte fórmula:

$$r = \sum_{k=1}^{m} \frac{(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{(x_i - \bar{x})^2}\sqrt{(y_i - \bar{y})^2}}$$

Equação 16 - Fórmula para calcular a similaridade através do PCC. Fonte: (ALI; MAJEED, 2021).

Na fórmula da coefieciente de correção de Perason, n é o número de elementos no conjunto de dados, x_i e y_i são observações individuais das variáveis x e y, respectivamente. O $\bar{x}$ e $\bar{y}$ são as médias das variáveis x e y.

2.4.7.3 Coeficiente de Jaccard

O coeficiente de Jaccard é uma estatística para medir a similaridade entre conjuntos finitos. É definido pelo tamanho da intersecção dividido pelo tamanho da união dos conjuntos amostrais (SUN et al., 2020).

$$\text{sim}(u_i, u_j) = \frac{N(u_i) \cap N(u_j)}{N(u_i) \cup N(u_j)}$$

Equação 17 - Fórmula para calcular a similaridade através do coeficiente de Jaccard. Fonte: (SUN et al., 2020).

O $N(u_i)$ representa o conjunto de itens avaliados pelo usuário u_i, $N(u_j)$ representa o conjunto de itens avaliados pelo usuário u_j.

2.4.7.4 Distância de Manhattan

A distância de Manhattan é uma medida de dissimilaridade, ou seja, quanto maior a similaridade entre dois pontos, mais diferentes eles são. Essa métrica pode ser calculada do seguinte modo (WANG et al., 2022):

$$d(vec_1, vec_2) = |p_1 - q_1| + |p_2 - q_2|$$

Equação 18 - Fórmula para calcular a distância através da distância de Manhattan. Fonte: (WANG et al., 2022).

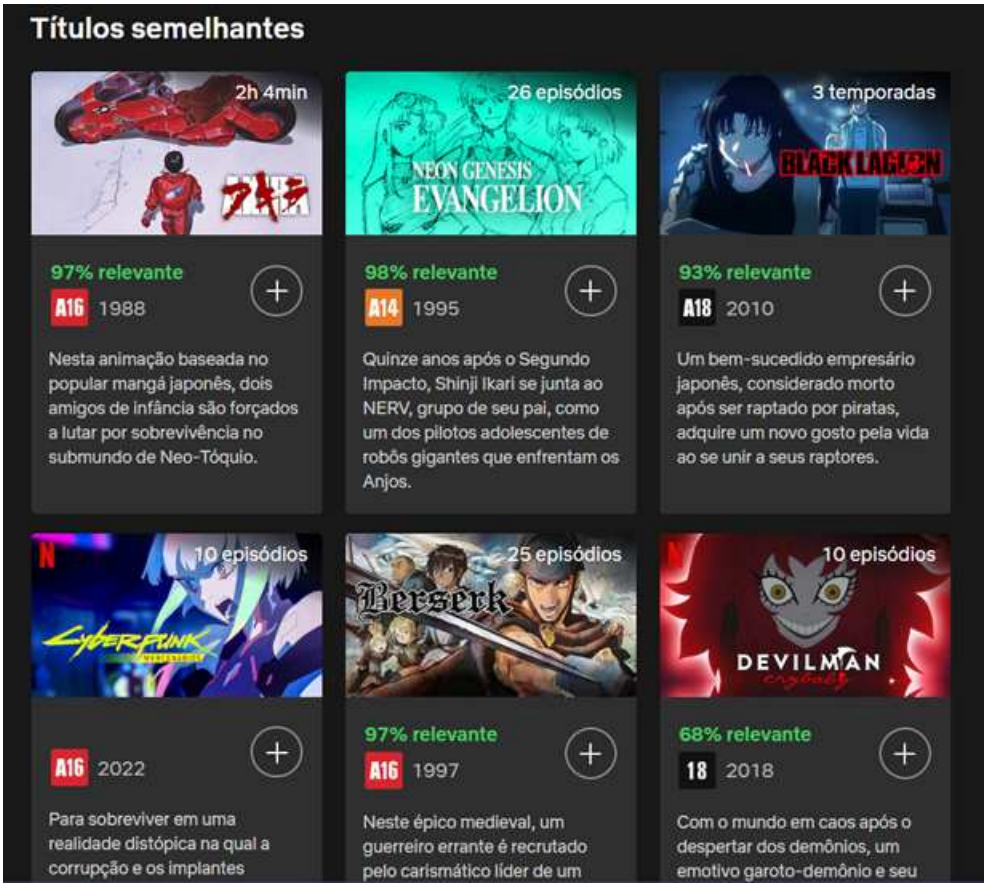
Ela calcula a distância entre dois vetores vec_1 e vec_2 que possuem duas coordenadas: p_1 e p_2 representam as coordenadas do primeiro vetor, enquanto q_1 e q_2 representam as coordenadas do segundo vetor.

2.5 Content-based filtering

Sistemas content-based são desenvolvidos para lidar com cenários em que os itens podem ser descritos por conjuntos de atributos descritivos. Nesse caso, as classificações e ações do usuário em relação a outros itens são suficientes para gerar recomendações significativas. Essa abordagem é especialmente útil quando há itens novos e com poucas classificações disponíveis (AGGARWAL, 2016).

O objetivo de um sistema de recomendação baseado em conteúdo é relacionar usuários a itens que sejam similares aos que eles gostaram no passado. A similaridade não é necessariamente baseada em classificações ou correlações entre usuários, mas sim nos atributos dos itens que o usuário apreciou (AGGARWAL, 2016). Por exemplo, um sistema baseado em conteúdo pode recomendar "títulos semelhantes"na Netflix após o acesso a uma obra, como ilustrado na Figura 4.

Figura 4 – Recomendações da Netflix baseadas no que o usuário assistiu previamente ou procurou



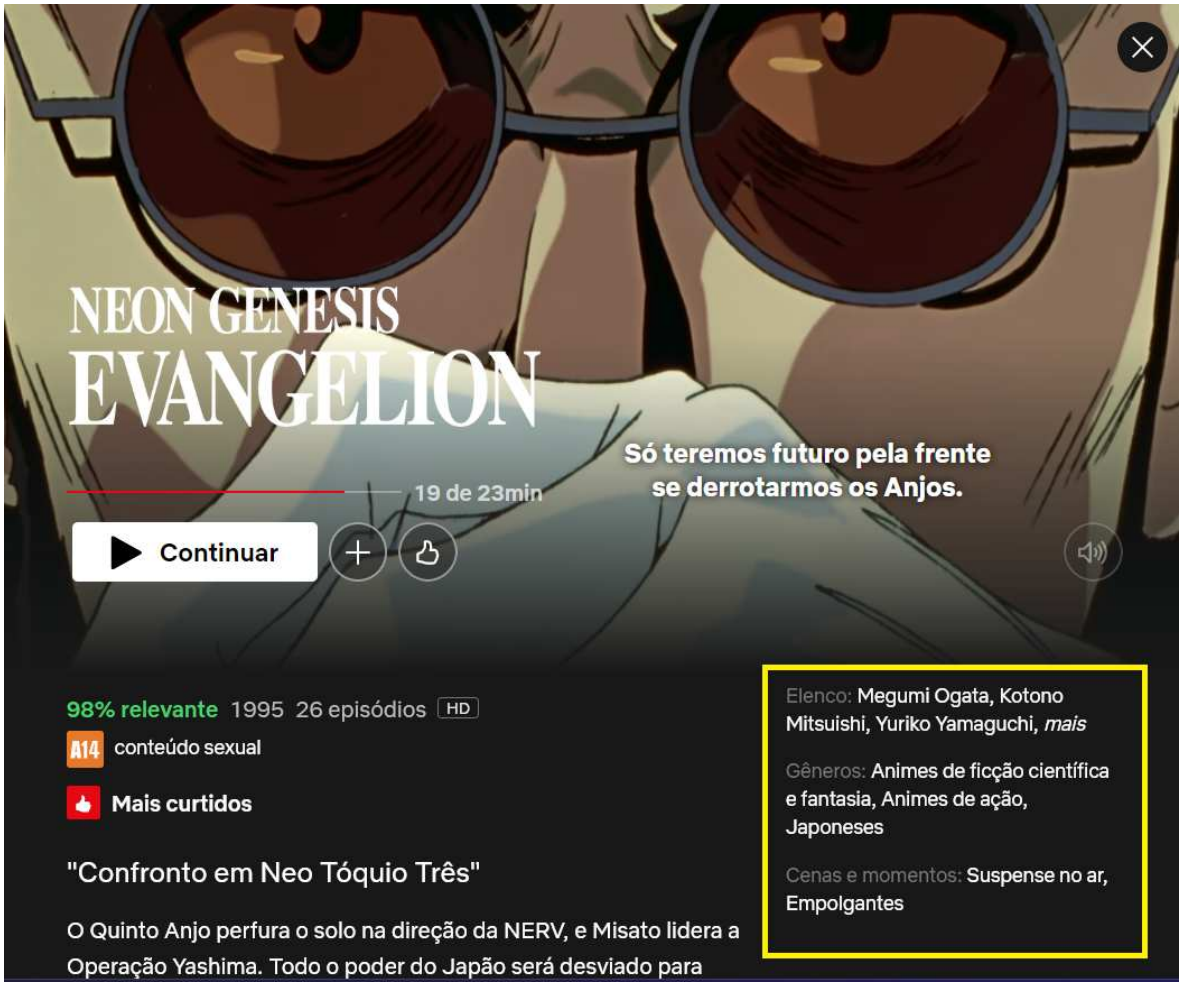
Fonte: Elaborada pelo autor

Ao contrário dos sistemas colaborativos, que utilizam as avaliações de outros usuários além do usuário-alvo, os sistemas baseados em conteúdo concentram-se principalmente nas avaliações do próprio usuário-alvo e nos atributos dos itens que o usuário demonstrou gostar (AGGARWAL, 2016).

Nos sistemas de recomendação baseados em conteúdo, o usuário tem pouca ou nenhuma interação, pois essa metodologia se baseia em uma fonte de dados diferente para o processo de recomendação. Na Figura 5, na área destacada, é possível observar um conjunto de informações de atributos que o usuário pode selecionar para receber recomendações de diferentes obras.

Os sistemas de recomendação baseados em conteúdo utilizam a descrição dos itens e o perfil do usuário como fontes de dados. O perfil do usuário é gerado a partir do feedback do usuário, que pode ser explícito ou implícito. Enquanto isso, os sistemas de recomendação baseados em conteúdo possuem componentes básicos que lidam com a variedade de dados e convertem essas informações em descrições padronizadas (AGGARWAL, 2016).

Figura 5 – No quadrado amarelo estão destacadas as tags de gêneros da obra, que são utilizadas para filtrar conteúdos



Fonte: Elaborada pelo autor

De acordo com (AGGARWAL, 2016), esses componentes podem ser distribuídos do seguinte modo:

- Pré-processamento e extração de características: Os sistemas de recomendação baseados em conteúdo são amplamente utilizados em diversos domínios, como páginas da web, descrições de produtos, notícias e recomendações musicais. Nessa etapa, é necessário pré-processar os dados e extrair características relevantes, coletando informações de diferentes fontes e convertendo-as em representações vetoriais baseadas em palavras-chave. A extração adequada das palavras-chave é fundamental para o bom desempenho do sistema de recomendação.

- Aprendizado de perfis de usuário baseado em conteúdo: Os modelos baseados em conteúdo são personalizados para cada usuário, levando em consideração o seu histórico de compras, classificações e interações com os itens. Esse aprendizado é realizado por meio da análise dos feedbacks explícitos e implícitos fornecidos pelo usuário, combinados com os atributos dos itens. O resultado desse processo é a criação de um perfil de usuário, que relaciona os interesses e preferências do usuário com os atributos dos itens.

- Filtragem e recomendação: Nesta etapa, o sistema de recomendação utiliza o perfil de usuário e as informações dos itens para realizar a filtragem e gerar recomendações personalizadas. É importante

que essa etapa seja realizada de forma eficiente, garantindo tempos de resposta rápidos para que as recomendações sejam feitas em tempo real, atendendo às necessidades e preferências do usuário de forma ágil e precisa.

2.5.1 Cálculo de similaride

A abordagem padrão de recomendações baseadas em conteúdo gira em torno do uso de uma lista de palavras-chave relevantes que aparecem em um documento. O conteúdo do documento pode ser codificado de várias maneiras em uma lista de palavras-chave.

Em uma abordagem simples, essa lista de palavras-chave pode ser representada por um vetor booleano, onde 1 indica que uma palavra aparece no documento e 0 indica que a palavra não aparece. No entanto, essa abordagem é desencorajada devido à existência de abordagens mais eficientes atualmente (JANNACH et al., 2010).

O TF-IDF (term frequency-inverse document frequency) é uma abordagem amplamente utilizada na codificação de documentos, baseada no campo da recuperação de informações. Os documentos de texto são transformados em vetores em um espaço euclidiano multidimensional, onde cada dimensão corresponde a uma palavra-chave presente nos documentos. A codificação é feita considerando a frequência do termo e a frequência inversa do documento. A Frequência do Termo descreve a frequência de ocorrência de um termo específico em um documento, sendo necessário normalizar o tamanho do documento para evitar viés em documentos maiores (JANNACH et al., 2010).

2.5.2 Criação de perfil de usuário

2.5.2.1 Nearest neighbors

A verificação de documentos similares aos que um usuário gostou é um método utilizado para determinar a relevância de um documento para esse usuário. Esse método requer o histórico de preferências do usuário e uma medida de similaridade entre documentos.

Um exemplo de aplicação desse método é encontrado em sistemas personalizados de acesso a notícias, onde o método k-nearest-neighboor (kNN) é usado para modelar os interesses a curto prazo dos usuários. Isso é importante para recomendações de notícias, pois o sistema procura por notícias semelhantes às que foram recentemente classificadas pelo usuário. Com base nessas classificações recentes, o método se adapta rapidamente e direciona o usuário para notícias relevantes e recentes. O método também estabelece um limite para a similaridade dos itens, evitando recomendar itens que o usuário já tenha visto (JANNACH et al., 2010).

Os métodos baseados em kNN têm vantagens, como facilidade de implementação e rápida adaptação a mudanças. Eles também podem fornecer recomendações razoáveis com um número pequeno de classificações do usuário. No entanto, a precisão de predição desses métodos pode ser inferior a técnicas mais avançadas (JANNACH et al., 2010).

2.5.2.2 Classificador de Bayes

Um classificador bayesiano é um método probabilístico usado para resolver problemas de classificação. Esse método se baseia na definição de probabilidade condicional e no teorema de Bayes. A abordagem

bayesiana na estatística utiliza a probabilidade para representar a incerteza nas relações aprendidas a partir dos dados (AMATRIAIN; PUJOL, 2015).

Os classificadores Naive Bayes têm benefícios como robustez a ruídos e atributos irrelevantes, além de lidarem com valores ausentes. No entanto, a suposição de independência pode não ser válida para alguns atributos correlacionados. Nesses casos, as Redes Bayesianas são uma alternativa comumente utilizada (AMATRIAIN; PUJOL, 2015).

O modelo bayesiano é amplamente empregado em sistemas de recomendação de classificação de texto. Tal modelo representa documentos, constrói um modelo de classificação e usa probabilidades para atribuir relevância ou interesse dos documentos aos usuários, possibilitando recomendações mais personalizadas e precisas. Esse modelo é amplamente utilizado em áreas como filtragem de spam, categorização de notícias e análise de sentimentos, devido à sua habilidade de lidar com incerteza e inferir categorias com base em probabilidades (AGGARWAL, 2016).

2.5.2.3 Classificadores baseados em regra

Os classificadores baseados em regra são algoritmos de aprendizado de máquina que utilizam regras explícitas para representar informações ou conhecimentos. Essas regras são expressas no formato SE-ENTÃO, onde a parte do SE é chamada de antecedente e a parte do ENTÃO é chamada de consequente (KUMAR et al., 2015).

SE condição ENTÃO conclusão

Nas regras item-item nos métodos baseados em conteúdo os antecedentes correspondem a presença de itens específicos na descrição. As regras são definidas do seguinte modo:

Item contém conjunto de palavras A $\Rightarrow$ Classificação = Gostei

Item contém conjunto de palavra B $\Rightarrow$ Classificação = Não gostei

No contexto desses classificadores, o antecedente de uma regra é considerado satisfeito para uma determinada linha (representação de palavras-chave do item) se todas as palavras-chave do antecedente estão presentes naquela linha (AGGARWAL, 2016).

O consequente corresponde a várias classificações, que podem ser assumidas como gostei ou não gostei binários. Por sua vez, uma linha é considerada satisfazendo o consequente de uma regra se o valor de classificação no consequente corresponde à variável dependente (classificação) dessa linha (AGGARWAL, 2016).

Nesse método, as métricas utilizadas são suporte e confiança. O suporte refere-se à fração de linhas que satisfazem tanto o antecedente quanto o consequente de uma regra. Já a confiança de uma regra é a fração de linhas que satisfazem o consequente, considerando apenas as linhas que já se sabe que satisfazem o antecedente (AGGARWAL, 2016).

Segundo (AGGARWAL, 2016), a abordagem para classificação baseada em conteúdo pode ser descrita do seguinte modo:

1. Fase de treinamento: determinar todas as regras relevantes a partir do perfil de usuário no desejado nível mínimo de suporte e confiança para o conjunto de treino representado por D_L;
2. Fase de teste: para cada descrição de item no conjunto de dados de teste, representado por D_U, determinar as regras disparadas e uma média de classificação. Ranquear os itens em D_U com base nessa classificação média;

Para que fique clara a compreensão sobre os métodos utilizados para criar o perfil de usuário, na Tabela 3 foram destacadas as vantagens e desvantagens de cada um desses métodos.

Tabela 3 – Apresentação das vantagens e desvantagens de métodos utilizados para criar o perfil de usuário

Nome	Desvantagens	Vantagens
Classificador de Bayes	É ingênuo, pois assume que todos os atributos são independentes entre si	Grande precisão e velocidade quando aplicado a grandes conjuntos de dados
	Sensível a atributos irrelevantes	Tem eficácia comprovada em classificação de texto
	Precisa de uma quantidade considerável de dados de treinamento para realizar recomendações mais precisas	Pontos de ruído individuais nos dados são considerados e atributos irrelevantes têm pouco ou nenhum impacto nas probabilidades posteriores calculadas
		Os componentes do classificador podem ser facilmente atualizados quando novos dados estão disponíveis
		A complexidade do tempo de aprendizado permanece linear ao número de exemplos

Classificador baseado em regra	Pode ter dificuldade em representar relações complexas, dado o fato de que as regras são definidas de modo simplificado Possui problemas com overfit, pois o excesso de regras pode levar a uma modelagem excessiva dos dados ou underfitting (modelagem insuficiente) Exige especialistas no domínio para que as regras sejam apropriadamente elaboradas	Facilmente interpretável pelo ser humano É transparente, dado o fato de que as regras são explicitamente definidas Flexível em razão do fato de que as regras podem ser facilmente modificadas
Vizinho mais próximo	Classificação de registros não conhecidos tende a ser mais cara Conforme o conjunto de treinamento cresce, ocorre maior demanda de recursos computacionais Ruído ou características irrelevantes afetam negativamente a acurácia do método	Facilmente compreensível Pode ser aplicado a problemas de classificação e regressão Se adapta bem a novos dados A construção de seu modelo é fácil

Fonte: Tabela criada a partir das observações feitas por (AGGARWAL, 2016), (JANNACH et al., 2010) e (RAY, 2019)

2.6 Matrix Factorization

Os sistemas de recomendação baseiam-se nas classificações de usuários e itens, representadas em duas matrizes distintas. Essas informações podem ser obtidas de diferentes maneiras, sendo as avaliações com estrelas uma das mais comuns, como ocorre em sites como a Amazon. No entanto, essas matrizes costumam ser esparsas, pois os usuários geralmente avaliam apenas uma fração dos itens disponíveis (SYMEONIDIS; ZIOUPOS, 2016).

Para lidar com os desafios de esparsidade e escalabilidade, os métodos de decomposição de matriz, também conhecidos como métodos de fatorização, são amplamente utilizados em sistemas de recomendação. A fatoração de matriz (matrix factorization) consiste em decompor a matriz original em um produto de outras matrizes, geralmente de dimensões menores. Nesse contexto, a fatoração de matriz é usada para representar a

interação entre itens e usuários como o produto de duas matrizes. Essas matrizes de fatores latentes capturam as características ocultas dos usuários e itens, permitindo inferir preferências e fazer previsões de recomendação com base nessas características (SYMEONIDIS; ZIOUPOS, 2016).

2.6.1 Técnicas baseadas em fatoração de matriz

Devido à alta dimensionalidade dos dados nos sistemas de recomendação, surgiram técnicas para lidar com esse problema. Uma dessas técnicas é a decomposição das matrizes de dados em matrizes menores, a fim de explorar os fatores latentes presentes no conjunto de dados. Essa abordagem permite economizar espaço de armazenamento e reduzir o processamento necessário. Com essa decomposição, é possível descobrir características não observáveis que podem influenciar as interações entre itens e usuários (FALK, 2019).

2.6.1.1 Single Value Decomposition (SVD)

É um método da álgebra linear cuja função é fatoração de uma matriz real ou complexa. O SVD decompõe a matriz em três produtos de matrizes: U, S e V. É uma forma de fatoração de matriz no qual as colunas U e V são ortogonais. A mutualidade ortogonal tem como vantagem que os conceitos podem ser completamente diferentes do outro, e eles podem ser geometricamente interpretados em gráficos de dispersão (AGGARWAL, 2016).

2.6.1.2 Funk SVD

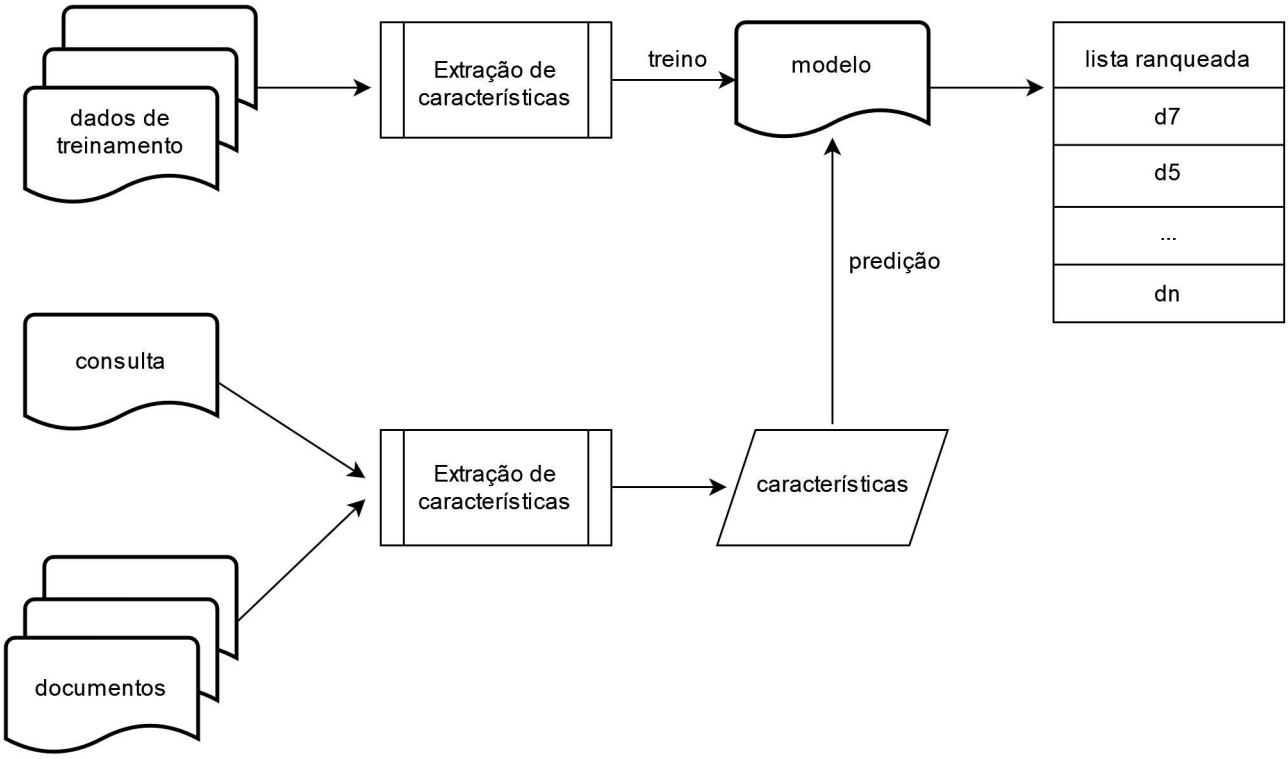
O algoritmo Funk SVD é uma versão simplificada da Decomposição em Valores Singulares (SVD) e foi criado por Simon Funk durante a competição Netflix Prize. Seu objetivo é melhorar a precisão das recomendações na plataforma Netflix, prevendo as classificações dos usuários (FALK, 2019).

O algoritmo utiliza uma abordagem baseada em gradiente descendente estocástico para atualizar iterativamente os fatores latentes de usuários e itens. Esses fatores latentes representam características não observáveis que influenciam as interações entre usuários e itens. Ao minimizar a diferença entre as classificações previstas e observadas, o Funk SVD busca encontrar os melhores parâmetros para gerar recomendações mais precisas e personalizadas (AGGARWAL, 2016).

2.7 Learning to rank

Os métodos de regressão e classificação no aprendizado de máquina supervisionado são amplamente aplicados na otimização de sistemas de pesquisa. Um objetivo importante nesse contexto é prever a taxa de clique em documentos para retornar os mais prováveis de serem clicados. Esse processo é conhecido como "learning to rank"(LtR) e faz parte da abordagem mais abrangente chamada "neural information retrieval". O LtR é amplamente utilizado em áreas como recuperação de informações e filtragem colaborativa. Em sistemas de recuperação de documentos, o LtR envolve atribuir uma pontuação a cada documento com base em uma consulta, classificando-os em ordem decrescente de pontuação, refletindo a relevância em relação à consulta (CAO et al., 2007). O fluxo do LtR é ilustrado na Figura 6.

Figura 6 – Fluxo do LtR



Fonte: (WANG et al., 2022)

Durante o aprendizado, consultas com rankings de documentos são usadas para criar uma função de ranqueamento precisa. Os algoritmos de learning to rank (LtR) são aplicados em problemas de recuperação de informações (IR), como recuperação de documentos, filtragem colaborativa e outros (LIU, 2007).

Esses algoritmos aprendem a combinar características extraídas de documentos e consultas por meio de treinamento discriminativo. Eles utilizam recursos como frequência de termos, pontuações BM25 e PageRank para medir a relevância dos documentos em relação às consultas. A capacidade de combinar diversas características é uma vantagem do LtR, permitindo considerar as complexas necessidades de informação dos usuários (LIU, 2007).

A idéia por trás do BM25 é determinar a relevância de documentos através do cálculo da razão de possibilidades. O BM25 não é apenas um modelo, mas uma família de modelos de ranqueamento. O PageRank é um algoritmo aplicado a busca na Web que faz uso da estrutura de hiperlink da Web para gerar um ranqueamento. Esse algoritmo se baseia na probabilidade de que uma pessoa, ao navegar na internet e clicar em links, chegará em uma página específica. A partir dessas informações, as páginas web são classificadas (LIU, 2007).

A outra propriedade, o discriminative training, está relacionada aos componentes-chave do método de LtR, que são o espaço de entrada, espaço de saída, espaço de hipótese e função de perda, conforme descrito por (LIU, 2007):

- Espaço de entrada: contém os objetos sob análise. Normalmente os objetos são representados por um vetor de características extraído de acordo com a aplicação em questão.

- Espaço de saída: contém o alvo de aprendizado em relação aos objetos de entrada. Existem duas definições relacionadas, mas diferentes, para esse ponto. A primeira definição é o espaço de saída da tarefa, que depende da aplicação. Por exemplo, no problema de regressão, o espaço de saída é o espaço dos números reais. A segunda definição é o espaço de saída usado para facilitar o processo de aprendizado, que pode ser diferente do espaço de saída da tarefa. Por exemplo, pode-se usar algoritmos de regressão para resolver problemas de classificação.

- Espaço de hipótese: define a classe de funções que mapeiam o espaço de entrada para o espaço de saída. Essas funções operam nos vetores de características dos objetos de entrada e fazem previsões de acordo com o formato do espaço de saída.

- Função de perda: para aprender a hipótese ideal, é utilizado um conjunto de treinamento contendo objetos independentes e identicamente distribuídos, juntamente com seus rótulos verdadeiros, selecionados do espaço de entrada e espaço de saída. A função de perda mede o grau de adequação da hipótese em relação ao rótulo verdadeiro. No aprendizado de máquina, a função de perda desempenha um papel importante, pois interpreta a aplicação alvo, ou seja, determina quais previsões são corretas e quais não são. Exemplos de funções de perda amplamente utilizadas para classificação são a perda exponencial, perda de articulação e regressão logística.

O treinamento discriminativo é um processo de aprendizado automático baseado nos dados de treinamento. Portanto, sistemas de pesquisa reais têm uma demanda por esse tipo de abordagem, pois recebem diariamente um feedback significativo dos usuários e logs que indicam um ranqueamento inadequado para determinadas consultas ou documentos.

2.7.1 Abordagem pointwise

Na abordagem pointwise, o espaço de entrada do sistema de recomendação consiste nos vetores de características associados a cada documento. Esses vetores descrevem as propriedades e atributos relevantes de cada item (LIU, 2007).

No problema de ranqueamento, geralmente são aplicados métodos de classificação, regressão ou classificação ordinal para resolver a tarefa. No entanto, uma abordagem comum é negligenciar as estruturas de grupo existentes, que consideram as relações e dependências entre documentos agrupados. Em vez disso, os grupos são tratados como dados de treinamento típicos de aprendizado supervisionado, onde x representa as características e y representa o rótulo de classe, número real ou rótulo de nota. Em seguida, métodos existentes de classificação, regressão ou classificação ordinal podem ser usados para realizar a aprendizagem (LI, 2015).

2.7.2 Abordagem pairwise

Na abordagem pairwise, o espaço de entrada contém um par de documentos, ambos representados por vetores de características. Por sua vez, o espaço de saída contém as preferências entre cada par de documentos. O rótulo de referência no espaço de saída é estabelecido considerando o grau de relevância (LIU, 2007).

Essa abordagem recebe dados de treinamento como entrada. A partir do dado rotulado de uma consulta q_i, representados por pares $(x_{i,1}, y_{i,1}), \ldots, (x_{i,n_i}, y_{i,n_i})$, em que i varia de 1 a m, são criados pares de preferência de vetores de características (documentos). Por exemplo, se x_i, j tem uma nota maior que x_i, k, então $x_i, j \succ x_i, k$, ou seja, o documento x_i, j possui preferência em razão de sua maior relevância. A partir

disso, métodos de classificação podem ser aplicados para treinar um classificador que seja capaz de realizar a classificação de documentos (LI, 2015).

2.7.3 Abordagem listwise

A abordagem listwise é uma estratégia de ranqueamento que utiliza listas de classificação como instâncias tanto no treinamento quanto na predição. Essa abordagem mantém a estrutura do grupo de classificação e permite a incorporação direta das medidas de avaliação de classificação nas funções de perda (LI, 2015).

No contexto do aprendizado de ranqueamento, a abordagem listwise utiliza os dados de treinamento associados a uma consulta específica para aprender um modelo de classificação. Esse modelo atribui pontuações aos vetores de recursos, que representam os documentos, e classifica os documentos com base nessas pontuações, posicionando aqueles com pontuações mais altas em posições superiores na lista (LI, 2015).

2.7.4 Considerações finais sobre métodos de Learning-to-Rank

As diferentes abordagens do Learning-to-Rank são aplicadas em contextos específicos e nem todas podem ser usadas no mesmo cenário. De acordo com (LI, 2015), a abordagem pointwise reduz o ranqueamento a um problema de regressão, classificação ou regressão ordinal. Já na abordagem pairwise, o ranqueamento é tratado como uma classificação de pares.

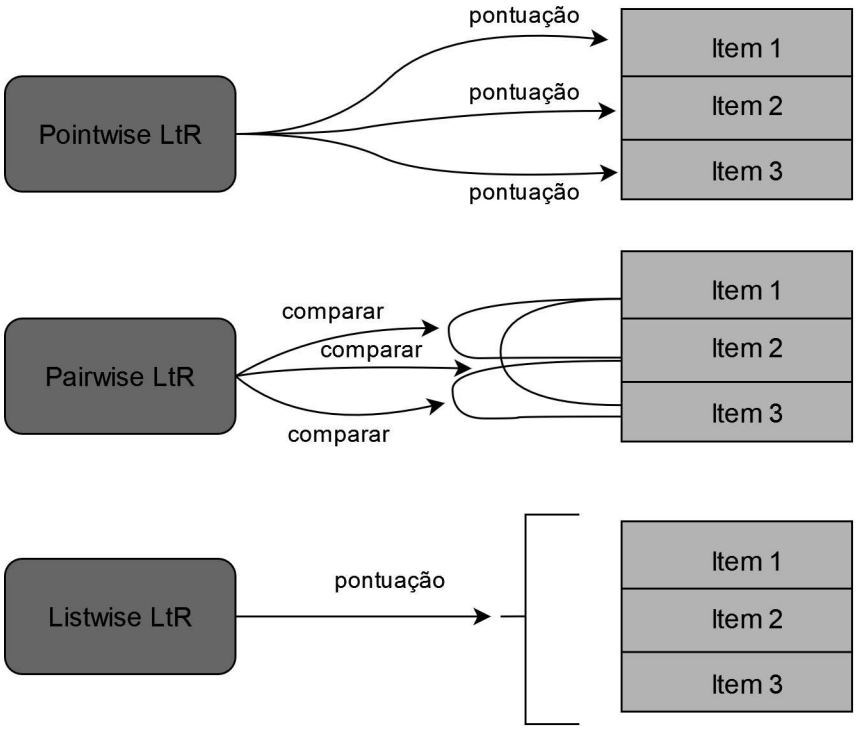
Essas abordagens têm suas vantagens, pois podem aproveitar diversas técnicas e ferramentas de machine learning. Por exemplo, o OC SVM utiliza hiperplanos como modelo de ranqueamento, enquanto o McRank é um algoritmo de machine learning que emprega funções de classificação para construir um modelo robusto (LI, 2015).

Por outro lado, a abordagem listwise trata o problema de ranqueamento como um desafio separado e define algoritmos específicos. A abordagem listwise leva em consideração o conceito de consulta e utiliza informações de posição nos resultados da classificação durante o treinamento do modelo de classificação (LIU, 2007).

Entre os algoritmos comuns na abordagem listwise, destacam-se o LambdaRank, que lida com a classificação de pares otimizando a métrica NDCG, e o ListNet, que utiliza redes neurais para classificar uma lista de itens com base em sua relevância (LI, 2015).

Por fim, a Figura 7 demonstra o funcionamento de cada abordagem para Learning to Rank.

Figura 7 – Demonstração visual do funcionamento de cada abordagem para Learning to rank



Fonte: (FALK, 2019)

2.8 Information retrieval

Nos últimos anos, áreas como visão computacional, aprendizado de máquina e reconhecimento de fala experimentaram avanços significativos, impulsionados pela modelagem de redes neurais, em particular as chamadas deep architectures, que possuem várias camadas ocultas (MITRA; CRASWELL, 2018).

Com base nesses avanços, houve uma expectativa de aplicar métodos neurais no campo da Recuperação de Informação, visando elevar o estado da arte e alcançar performances revolucionárias, assim como observado nas áreas mencionadas anteriormente. Segundo (MITRA e CRASWELL, 2018), Neural Information Retrieval pode ser compreendido como

> aplicações de redes neurais rasas ou profundas para recuperação de informação. Modelos neurais tem sido empregados em vários cenários de IR - incluindo recuperação ad-hoc, sistemas de recomendação, pesquisa multimídia, e até mesmo sistemas conversacionais que geram respostas em resposta a questões em linguagem natural.

De acordo com (MITRA e CRASWELL, 2018), a Information Retrieval pode tomar diferentes formas. Os usuários podem expressar suas necessidades por informação através de consultas textuais, reconhecimento de voz, imagens e em alguns casos até mesmo de modo implícito.

Segundo (MITRA e CRASWELL, 2018),

> uma consulta de pesquisa normalmente contém vários termos, enquanto que o comprimento de um documento pode variar dependendo do cenário, indo de apenas alguns termos a centenas de sentenças ou mais. Dado esse cenário, modelos neurais usam representações vetoriais de texto, que normalmente contém um grande número de parâmetros que precisam ser ajustados.

Dado esse fato, os sistemas de IR devem lidar com consultas que podem conter vocabulário não visto previamente, para corresponder a documentos que variam em tamanho, para encontrar documentos relevantes que também podem conter grandes seções de texto irrelevante.

Sistemas de IR devem aprender padrões em consultas e documentos de texto que indicam relevância, mesmo se a consulta e o documento usam vocabulário diferente, e mesmo se os padrões são específicos de tarefa ou específicos de contexto (MITRA; CRASWELL, 2018).

2.8.1 Abordagens para information retrieval

Os modelos clássicos de Recuperação de Informação (IR) utilizam termos-índice para representar e resumir o conteúdo dos documentos. Esses termos, geralmente substantivos, auxiliam na lembrança dos principais assuntos abordados, enquanto adjetivos, verbos e conectivos têm um papel complementar. No entanto, é importante considerar todas as palavras diferentes presentes no documento para uma representação abrangente (BAEZA-YATES; RIBEIRO-NETO, 1999).

Ao analisar um conjunto de termos-índice para um documento, percebe-se que nem todos têm a mesma relevância na descrição do conteúdo. Decidir quais termos são mais importantes para resumir o assunto de um documento não é uma tarefa fácil. No entanto, existem propriedades em um termo-índice que podem ser medidas para verificar seu potencial (BAEZA-YATES; RIBEIRO-NETO, 1999).

Uma palavra que aparece em todos os documentos não é útil como termo-índice, pois não fornece informações sobre quais documentos podem interessar ao usuário. Por outro lado, uma palavra que aparece em apenas alguns documentos é bastante útil, pois restringe significativamente o conjunto de documentos que podem ser do interesse do usuário.

Dessa forma, fica evidente que diferentes termos-índice possuem relevância variável na descrição dos assuntos de um documento. Esse efeito é observado por meio da atribuição de pesos a cada termo-índice em um documento (BAEZA-YATES; RIBEIRO-NETO, 1999).

Ao longo de várias pesquisas e investigações sobre o funcionamento dos termos dentro de um documento pesquisado, três abordagens tradicionais foram estabelecidas para resolver os problemas de Recuperação de Informação (IR): abordagem booleana, abordagem vetorial e abordagem probabilística.

2.8.1.1 Modelo booleano

O modelo booleano, baseado na teoria de conjuntos e álgebra booleana, é um modelo de recuperação amplamente utilizado no passado. Esse modelo utiliza expressões booleanas para representar consultas, o que garante precisão nos termos utilizados. No entanto, o modelo possui limitações significativas. Sua estratégia de recuperação é baseada em uma classificação binária de documentos como relevantes ou irrelevantes, sem levar em conta critérios adicionais de classificação. Isso resulta em um desempenho limitado na recuperação de informações e torna o modelo mais adequado para recuperação de dados do que para informações em si. Além disso, o modelo enfrenta dificuldades na tradução precisa das necessidades de informação em expressões booleanas, devido à falta de precisão semântica inerente a esse método (BAEZA-YATES; RIBEIRO-NETO, 1999).

No modelo booleano, os termos-índice são considerados como ausentes ou presentes nos documentos, e os pesos atribuídos a esses termos são binários, assumindo valores de zero ou um. Uma consulta é composta por termos-índice conectados por três operadores booleanos: "not", "and"e "or". Assim, a consulta é essencialmente

representada como uma disjunção de um conjunto de vetores. Para fins explicativos, (BAEZA-YATES e NETO, 1999), usa como exemplo a consulta

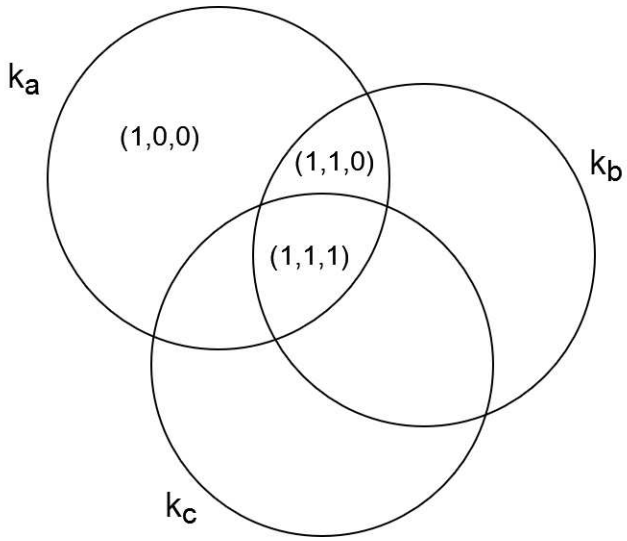
$$[\![q = k_a \wedge (k_b \vee \neg k_c)]\!]$$

Que pode ser escrita na forma normal disjuntiva como

$$[\![\vec{q}_{dnf} = \langle 1,1,1\rangle \vee \langle 1,1,0\rangle \vee \langle 1,0,0\rangle]\!]$$

A Figura 8 ilustra os componentes conjuntivos da consulta q.

Figura 8 – Os três componentes conjuntivos da consulta $[q = k_a \wedge (k_b \vee \neg k_c)]$



Fonte: (BAEZA-YATES; RIBEIRO-NETO, 1999)

O modelo booleano no contexto da recuperação de informações considera cada documento como sendo relevante ou não relevante, sem levar em conta a correspondência parcial entre os termos da consulta e os documentos. Em resumo, uma das principais vantagens desse modelo é seu formalismo claro e sua simplicidade.

No entanto, sua principal desvantagem é que a correspondência exata entre os termos pode resultar em um número excessivo ou insuficiente de documentos recuperados. Atualmente, sabe-se que atribuir pesos aos termos índice pode levar a melhorias significativas no desempenho da busca (BAEZA-YATES; RIBEIRO-NETO, 1999).

2.8.1.2 Modelo vetorial

O modelo vetorial reconhece a importância da correspondência parcial na recuperação de informações, abordando assim a limitação do modelo booleano. Esse modelo utiliza pesos não binários para os termos índice em consultas e documentos, permitindo uma maior flexibilidade.

Segundo (BAEZA-YATES e RIBEIRO-NETO, 1999), pesos são utilizados para calcular o grau de similaridade entre cada documento armazenado no sistema e a consulta do usuário. Ao ordenar os documentos recuperados com base no grau de similaridade, o modelo vetorial leva em consideração termos que atendem

parcialmente à consulta, resultando em um conjunto de documentos resposta ranqueados mais preciso em comparação ao conjunto de documentos resposta obtidos pelo modelo booleano.

De acordo com (BAEZA-YATES e RIBEIRO-NETO, 1999), esse modelo é definido do seguinte modo:

> para o modelo vetorial, o peso w_i, j associado a um par (k_j, d_j) é positivo e não binário. Além disso, os termos índice na consulta são também pesados. Deixe que w_i, q seja o peso associado ao par $[k_i, q]$, onde $w_i, q > 0$. Assim, o vetor de consulta $\vec{q}$ é definido como $\vec{q} = (w_1, q, w_2, q, ..., w_t, q)$ onde t é o número total de termos índice no sistema. Como antes, o vetor para um documento $\vec{d_j}$ representado por $\vec{d_j} = (w_1, j, w_2, j, \ldots, w_t, j)$.

O modelo vetorial utiliza representações em vetor para documentos e consultas, onde cada documento e consulta são vetores de dimensão-t. O objetivo é avaliar a similaridade entre o documento d_j e a consulta q considerando a correlação entre os vetores d e q. A norma dos vetores de documentos ($|\vec{d_j}|$) e de consultas ($|\vec{q_j}|$) não afeta o ranqueamento, com o fator $|\vec{q}|$ sendo o mesmo para todos os documentos. O fator $|\vec{d_j}|$ normaliza o espaço de documentos, considerando a magnitude dos vetores (BAEZA-YATES; RIBEIRO-NETO, 1999).

A correlação entre os vetores $\vec{d_j}$ e $\vec{q}$ pode ser quantificada pelo cosseno do ângulo entre esses dois vetores, sendo um de documentos e outro de consulta:

$$\text{sim}(d_j, q) = \frac{d_j \cdot q}{\|d_j\| \cdot \|q\|}$$

Equação 19 - Cálculo do cosseno entre dois vetores, sendo um vetor $\vec{d_j}$ de documentos e outro vetor $\vec{q}$. Fonte: (BAEZA-YATES; RIBEIRO-NETO, 1999)

O cálculo de $sim(d_j, q)$ tem resultado que varia de de 0 a +1. Ao invés de tentar prever se um documento é relevante ou não, o modelo vetorial ranqueia o documento de acordo com seu grau de similaridade com a consulta, logo um documento será retornado mesmo que haja apenas uma correspondência parcial com a consulta (BAEZA-YATES; RIBEIRO-NETO, 1999).

No modelo vetorial, os pesos são geralmente calculados usando a frequência do termo (tf), que representa a frequência de ocorrência dos termos em um documento ou texto de consulta, e o fator de frequência inversa do documento (idf), que mede o inverso do número de documentos que contêm um determinado termo de consulta ou documento. Isso permite atribuir importância relativa aos termos com base em sua frequência e raridade no corpus de documentos (HIEMSTRA; de Vries, 2000).

Por fim, (YATES e NETO, 1999) afirmam que as principais vantagens do modelo vetorial são:

- boa performance na recuperação em razão de como os termos índice tem seus pesos atribuídos;
- correspondência parcial de termos, permitindo recuperação de documentos que se aproximem as condições da consulta;
- a fórmula de ranqueamento de cosseno ordena os documentos de acordo com o grau de similaridade a consulta. A desvantagem é de que se presume que os termos índice sejam mutuamente independentes. Devido à localização de muitas dependências de termo, a aplicação indiscriminada a todos os documentos na coleção pode ser danosa à performance da recuperação.

2.8.1.3 Modelo probabilístico

O modelo probabilístico é uma abordagem que utiliza métodos probabilísticos para resolver o problema de Recuperação de Informação (IR). No processo de consulta, busca-se especificar as propriedades do conjunto ideal de respostas, que contém exatamente os documentos relevantes e nenhum outro. No entanto, essas propriedades não são conhecidas, e é necessário fazer uma estimativa inicial utilizando termos índice. Essa estimativa é usada para gerar uma descrição probabilística preliminar do conjunto ideal de respostas e recuperar um primeiro conjunto de documentos (BAEZA-YATES; RIBEIRO-NETO, 1999).

Segundo (YATES e NETO, 1999), a maior vantagem do modelo probabilístico, é a de que os documentos são ranqueados na ordem decrescente de sua probabilidade de ser relevante. As desvantagens incluem:

- necessidade de adivinhar a separação inicial dos documentos em conjuntos relevantes e não relevantes;
- o fato de que o método não leva em consideração a frequência com que cada termo índice ocorre em um documento;
- adoção de suposições independentes para termos índice, entretanto vale lembrar de que não é claro que a independência de termos é uma suposição ruim em situações práticas.

3 Metodologia

Este trabalho adotou a metodologia de pesquisa descritiva, utilizando o levantamento de dados bibliográficos para investigar o estado da arte das técnicas de recomendação mais utilizadas atualmente na literatura acadêmica. O objetivo foi compreender e analisar as abordagens mais recentes e relevantes empregadas nessa área de pesquisa.

3.1 Materiais utilizados

Para a metodologia deste trabalho, foram utilizados artigos científicos disponíveis gratuitamente na internet, obtidos por meio do portal de pesquisa acadêmica Google Scholar. A seleção desses artigos foi realizada por meio de consultas específicas, usando consultas com termos como "collaborative filtering", "recommender systems"e "machine learning", a fim de obter uma busca mais refinada sobre as diferentes técnicas de desenvolvimento de sistemas de recomendação.

Alguns dos artigos encontrados foram acessíveis por meio de links fornecidos pelo Google, que nem sempre direcionavam a um portal específico, mas sim ao local onde o artigo estava hospedado. Além disso, foi realizado um levantamento da quantidade de artigos encontrados em cada tema ao longo dos anos, abrangendo o período de 2003 a 2022. Essa análise permitiu compreender a evolução e a relevância dos temas ao longo do tempo.

3.2 Procedimento realizado

Como explicado na seção anterior, a metodologia do trabalho é conduzida no espaço de pesquisa do Google Scholar. Ao longo da pesquisa foi utilizado um padrão de busca para encontrar artigos com assuntos relacionados ao tema do trabalho. Esse padrão de busca se enquadra como um método de pesquisa avançada dentro do Google Scholar, pois é feito uma busca com base em condições específicas.

As palavras-chave utilizada nas pesquisas sempre foi acompanhada do tema em questão seguido do termo “recommender OR recommendation system”. Um exemplo de consulta é a seguinte: "content based recommender OR recommendation system". Ao colocar as palavras-chave entre aspas o mecanismo de busca se certifica de incluir nos resultados exatamente a palavra-chave buscada, letras ou números que o mecanismo de busca do Google normalmente ignoraria.

Além disso sempre é feita a delimitação do ano de pesquisa, como “2020”, e não de um intervalo de anos de pesquisa, como “2003 – 2022”, pois desse modo pode ser analisado o número de publicações relativas ao tema que foram divulgadas a cada ano.

Outro ponto importante da consulta para pesquisa é a utilização de operadores lógicos, como o OR ou AND, pois através deles a pesquisa se torna mais refinada. A opção “incluir citações” também foi marcada durante as pesquisas para que o leque de opções fosse maior.

Por fim, de modo geral, foi escolhido como intervalo de anos para pesquisa, publicações feitas entre 2003 e 2022. O ano de 2003 foi escolhido como ano inicial em razão do baixíssimo número de resultados que havia antes desse ano.

Após a etapa de pesquisa dos dados foi feita uma análise estatística das informações encontradas para entender quais técnicas tem sido mais populares.

4 Resultados

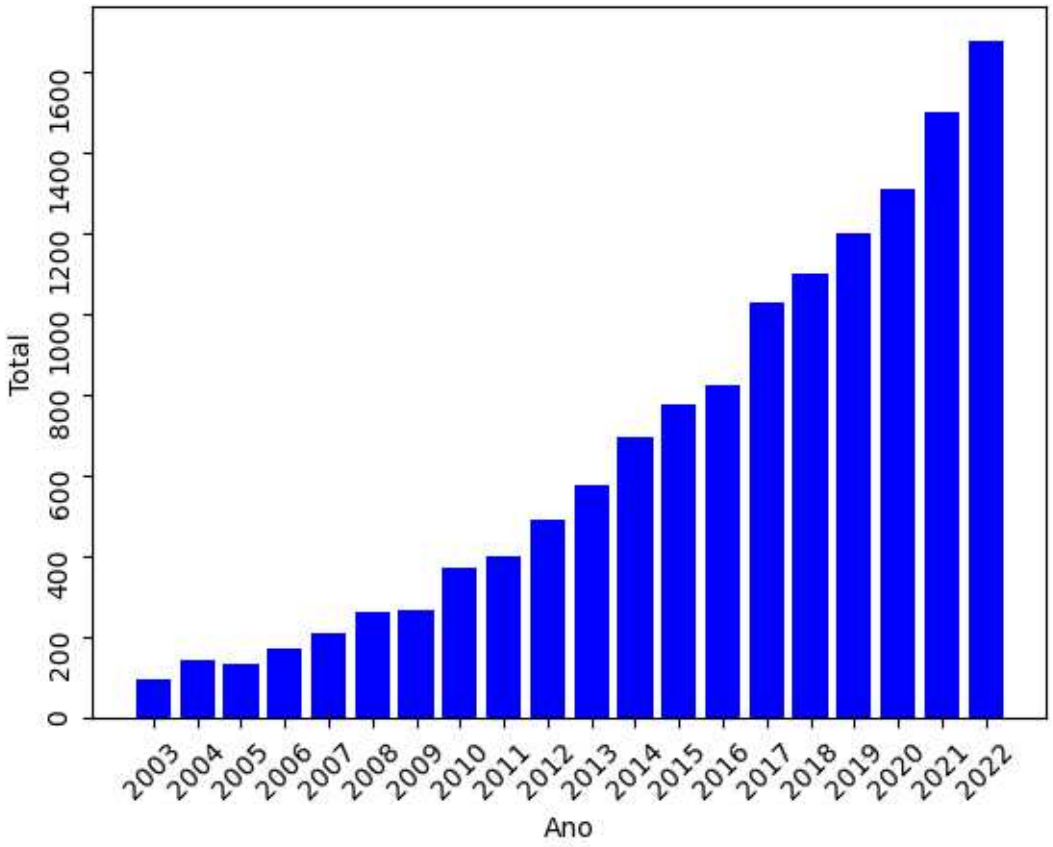
Após realizar pesquisas com as consultas da tabela 4 pode ser observado que o método de filtragem colaborativa, apresentado na Figura 10, tem maior popularidade que o método de filtragem baseada em conteúdo (Figura 9), entretanto, a filtragem colaborativa ao longo do tempo passou a ficar atrás dos métodos oriundos do machine learning.

Tabela 4 – Consultas utilizadas para localizar artigos no Google Scholar sobre sistemas de recomendação envolvendo uma técnica em específico

Consultas
““content-based filtering” “recommender OR recommendation system””
““collaborative filtering” “recommender OR recommendation system””
““machine learning” “recommender OR recommendation system””

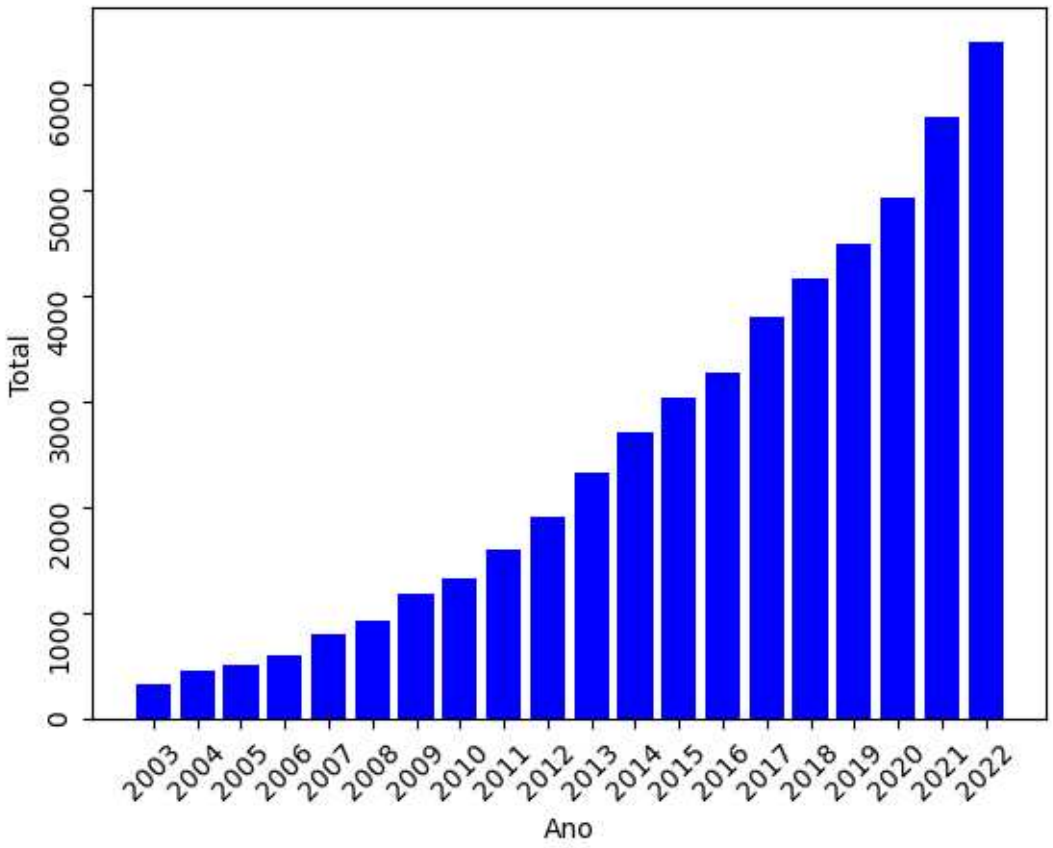
Fonte: Criada pelo próprio autor a partir de dados obtidos

Figura 9 – Distribuição do quantitativo de artigos sobre filtragem baseada em conteúdo encontrados ao longo de 2003 a 2022



Fonte:Criada pelo próprio autor a partir de dados obtidos pelo Google Scholar

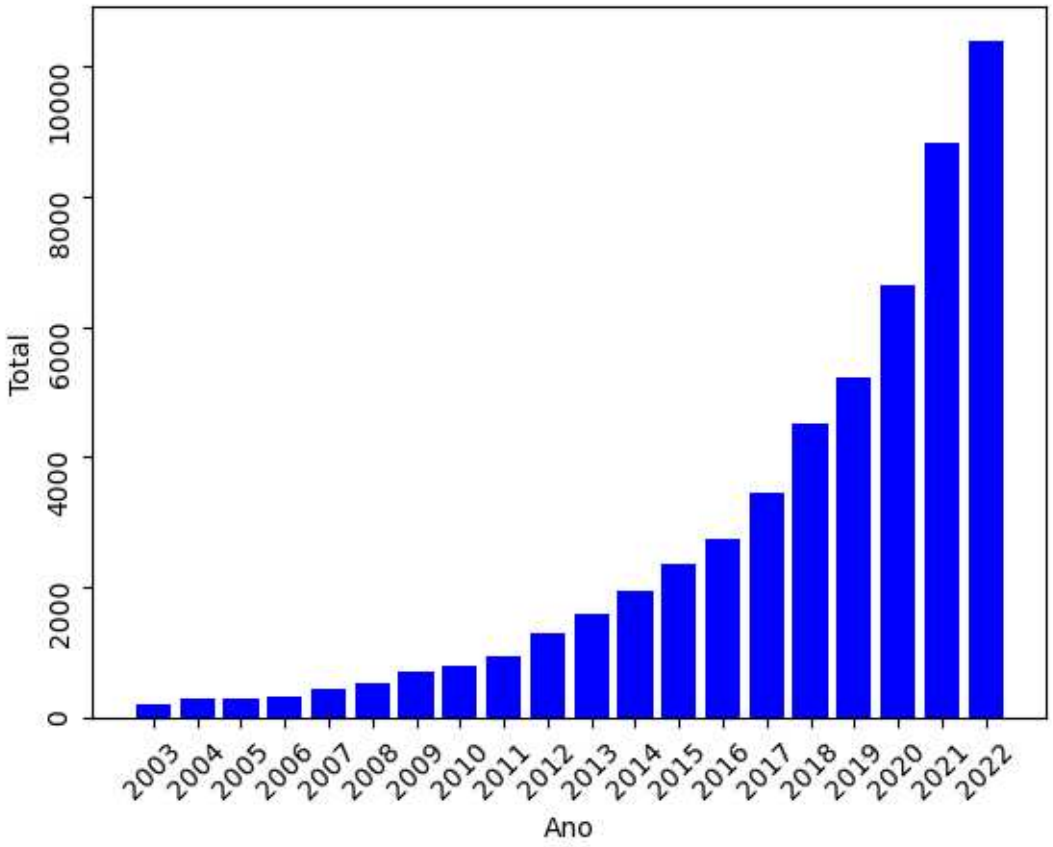
Figura 10 – Distribuição do quantitativo de artigos sobre filtragem colaborativa encontrados ao longo de 2003 a 2022



Fonte:Criada pelo próprio autor a partir de dados obtidos pelo Google Scholar

De acordo com os estudos realizados ao longo deste trabalho, especialmente durante a descoberta do número de trabalhos produzidos envolvendo machine learning e sistemas de recomendação, ficou evidente que os métodos de machine learning aplicados a sistemas de recomendação, atualmente, representam a vanguarda dos algoritmos de recomendação. Isso pode ser afirmado com base no número de trabalhos produzidos sobre machine learning e sistemas de recomendação, conforme mostrado na Figura 11.

Figura 11 – Distribuição do quantitativo de artigos sobre machine learning entre 2003 e 2022



Fonte: Criada pelo próprio autor a partir de dados obtidos pelo Google Scholar

Conforme observado por (GOODFELLOW, BENGIO e COURVILLE, 2016), os sistemas de recomendação têm obtido os melhores resultados por meio de métodos de machine learning, tais como Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Deep Belief Networks (DBN) e entre outros.

Além disso, conforme apontado por (ZHANG et al., 2019), as arquiteturas baseadas em deep learning, também chamadas de arquiteturas neurais, têm se mostrado vantajosas para sistemas de recomendação. Essas arquiteturas possuem características como o funcionamento end-to-end, ou seja, desde o início até o fim do problema de forma distinta, e também apresentam viés indutivo, que são as restrições assumidas pelos algoritmos de machine learning durante o processo de aprendizado.

A escolha de métodos baseados em machine learning e redes neurais é respaldada pelo trabalho de (HE et al., 2017), que comparou o desempenho do método MLP (Multi-layer Perceptron) com a fatoração de matriz, sendo que o MLP obteve resultados superiores.

Outra justificativa apontada por (ZHANG et al., 2019) que impulsiona o uso crescente de deep neural networks é a disponibilidade de grandes volumes de dados com complexidades diversas. Os métodos tradicionais de sistemas de recomendação têm dificuldade em lidar de forma precisa e eficiente com essa quantidade de dados e complexidade.

Além disso, técnicas de classificação supervisionada e regressão têm demonstrado resultados compu-

tacionais e estatísticos superiores. Com o uso de frameworks como Tensorflow ou PyTorch e o treinamento em GPUs, essas técnicas alcançam bom desempenho. Métodos como kNN e classificação Bayesiana são amplamente utilizados devido à sua precisão. Essa preferência tem levado a um aumento significativo na pesquisa em machine learning e sistemas de recomendação, superando numericamente a quantidade de estudos sobre filtragem colaborativa e filtragem baseada em conteúdo.

Para finalizar, dado o fato de que a esparsidade é um dos principais desafios enfrentados pelos sistemas de recomendação, as técnicas estatísticas de machine learning, como a matriz de fatorização, têm se mostrado eficazes para realizar previsões precisas e recomendações, revelando fatores latentes subjacentes que contribuem para a solução desse problema, tornando assim os métodos de machine learning mais atraentes para o desenvolvimento de um sistema de recomendação.

5 Conclusão

O objetivo deste trabalho consistiu em realizar uma revisão de literatura dos métodos amplamente utilizados no desenvolvimento de sistemas de recomendação. Considerando a diversidade de técnicas e algoritmos disponíveis para lidar com os desafios da recomendação de conteúdo ou produtos, é crucial abordar os métodos mais populares e destacar as compensações existentes em diferentes cenários.

Dentro do escopo deste trabalho, foram abordados diversos tópicos relacionados aos sistemas de recomendação, como machine learning, redes neurais aplicadas à pesquisa textual, deep neural networks, filtragem colaborativa, filtragem baseada em conteúdo, fatoração de matriz, aprendizado de classificação e recuperação de informações. Essa abordagem permitiu uma visão abrangente das técnicas existentes sem comprometer a extensão do trabalho.

As abordagens baseadas em machine learning e redes neurais têm ganhado destaque nos últimos anos, tornando-se técnicas populares no desenvolvimento de sistemas de recomendação robustos, eficientes e precisos. Apesar disso, a filtragem colaborativa continua sendo uma abordagem promissora, oferecendo resultados satisfatórios ao levar em consideração as preferências e comportamentos dos usuários para realizar recomendações de itens.

No entanto, é importante ressaltar que esse estudo identificou desafios e limitações no contexto dos sistemas de recomendação. O problema do "cold start"é um desafio frequente na filtragem colaborativa, pois para fazer recomendações é necessário ter um conjunto de dados para trabalhar, mas nem sempre essas informações estão disponíveis.

Além disso, a privacidade e a segurança são preocupações relevantes, uma vez que os sistemas de recomendação geralmente utilizam dados pessoais dos usuários, exigindo a garantia de segurança dessas informações sensíveis.

Quanto ao cálculo da similaridade de itens, a literatura sugere que a similaridade de Pearson possui vantagens em relação a outros métodos de cálculo. Ao longo deste estudo, ficou evidente que a aplicação adequada das métricas de avaliação é fundamental para compreender o desempenho dos sistemas de recomendação.

Essas métricas possibilitam a comparação dos diferentes métodos e técnicas utilizados, permitindo que pesquisadores e profissionais selecionem abordagens mais adequadas para cada contexto específico.

Em última análise, conclui-se que não existe um design único e universalmente adequado para a construção de sistemas de recomendação. Em vez disso, há uma variedade de métodos e técnicas disponíveis. Nesse sentido, os algoritmos de machine learning têm recebido destaque no desenvolvimento desses sistemas, como destacado na seção de resultados deste trabalho. Embora não haja uma fórmula exata para escolher a abordagem a ser utilizada, é valioso compreender esses algoritmos devido às suas vantagens estatísticas e computacionais em relação às técnicas mais comumente aplicadas. Além disso, é recomendável testar os sistemas de recomendação em diferentes configurações, especialmente em cenários específicos, para uma avaliação adequada dos resultados.

5.0.1 Trabalhos futuros

Na área de sistemas de recomendação, as técnicas modernas desempenham um papel crucial em termos de desempenho e precisão. Nos últimos anos, abordagens avançadas de inteligência artificial, como aprendizado de máquina e mineração de dados, têm ganhado destaque devido aos seus resultados impressionantes.

Após a conclusão deste trabalho, observou-se que os métodos baseados em redes neurais profundas são uma tendência atual. Isso se deve às suas capacidades de lidar com dados complexos e oferecer um alto desempenho. Ao utilizar redes neurais profundas, é possível compreender representações complexas dos dados, o que permite a captura de fatores latentes para gerar recomendações mais precisas e direcionadas.

Diante dessa perspectiva, os trabalhos futuros sobre sistemas de recomendação que exploram técnicas baseadas em aprendizado com redes neurais profundas se mostram promissores. Essa abordagem permite continuar explorando melhores alternativas para o desenvolvimento de sistemas de recomendação que ofereçam resultados mais personalizados, confiáveis e que conseguem lidar melhor com problemas típicos de sistemas de recomendação, como escalabilidade e esparsidade dos dados.

Um outro caminho que pode ser considerando para trabalhos futuros é a aplicação de um determinado método baseado em redes neurais profundas na criação de um pequeno sistema de recomendação a fim de identificar performance e qualidade das recomendações dentro de uma determinada base de dados.

Referências

AGGARWAL, C. *Data Classification: Algorithms and Applications*. CRC Press, 2020. (Chapman & Hall/CRC Data Mining and Knowledge Discovery). ISBN 9780367659141. Disponível em: <https://books.google.com.br/books?id=06q3zQEACAAJ>.

AGGARWAL, C. C. *Recommender Systems*. Springer International Publishing, 2016. Disponível em: <https://doi.org/10.1007/978-3-319-29659-3>.

AGGARWAL, C. C. *Neural Networks and Deep Learning*. Springer International Publishing, 2018. Disponível em: <https://doi.org/10.1007/978-3-319-94463-0>.

ALI, S. I. M.; MAJEED, S. S. "a review of collaborative filtering recommendation system. *Muthanna Journal of Pure Science*, Al-Muthanna University, v. 8, n. 1, p. 120–131, jan. 2021. Disponível em: <https://doi.org/10.52113/2/08.01.2021/120-131>.

ALMAZRO, D. et al. *A Survey Paper on Recommender Systems*. arXiv, 2010. Disponível em: <https://arxiv.org/abs/1006.5278>.

AMATRIAIN, X.; PUJOL, J. M. Data mining methods for recommender systems. In: *Recommender Systems Handbook*. Springer US, 2015. p. 227–262. Disponível em: <https://doi.org/10.1007/978-1-4899-7637-6_7>.

ARULKUMARAN, K. et al. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, Institute of Electrical and Electronics Engineers (IEEE), v. 34, n. 6, p. 26–38, nov. 2017. Disponível em: <https://doi.org/10.1109/msp.2017.2743240>.

BAEZA-YATES, R.; RIBEIRO-NETO, B. *Modern Information Retrieval*. New York: Addison-Wesley, 1999.

BALAKRISHNAN, S.; CHOPRA, S. Collaborative ranking. In: *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 2012. Disponível em: <https://doi.org/10.1145/2124295.2124314>.

BEEL, J. et al. Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries*, Springer Science and Business Media LLC, v. 17, n. 4, p. 305–338, jul. 2015. Disponível em: <https://doi.org/10.1007/s00799-015-0156-0>.

CAO, Z. et al. Learning to rank. In: *Proceedings of the 24th international conference on Machine learning*. ACM, 2007. Disponível em: <https://doi.org/10.1145/1273496.1273513>.

DESHPANDE, M.; KARYPIS, G. Item-based top-in/irecommendation algorithms. *ACM Transactions on Information Systems*, Association for Computing Machinery (ACM), v. 22, n. 1, p. 143–177, jan. 2004. Disponível em: <https://doi.org/10.1145/963770.963776>.

FALK, K. *Practical Recommender Systems*. Manning, 2019. ISBN 9781617292705. Disponível em: <https://books.google.com.br/books?id=_dbdnAAACAAJ>.

FAYYAZ, Z. et al. Recommendation systems: Algorithms, challenges, metrics, and business opportunities. *Applied Sciences*, MDPI AG, v. 10, n. 21, p. 7748, nov. 2020. Disponível em: <https://doi.org/10.3390/app10217748>.

GOODFELLOW, I. et al. *Deep Learning*. Cambridge, Massachusetts: The MIT Press, 2016.

HE, X. et al. *Neural Collaborative Filtering*. arXiv, 2017. Disponível em: <https://arxiv.org/abs/1708.05031>.

HERLOCKER, J. L. et al. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, Association for Computing Machinery (ACM), v. 22, n. 1, p. 5–53, jan. 2004. Disponível em: <https://doi.org/10.1145/963770.963772>.

HIEMSTRA, D.; de Vries, A. *Relating the new language models of information retrieval to the traditional retrieval models*. Netherlands: University of Twente, 2000. v. 00. (CTIT Technical report series, 00-09). Imported from CTIT.

JANNACH, D. et al. *Recommender Systems an Introduction*. Leiden: Cambridge University Press, 2010. Disponível em: <http://www.amazon.com/Recommender-Systems-Introduction-Dietmar-Jannach/dp/0521493366>.

JäRVELIN, K.; KEKäLäINEN, J. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, Association for Computing Machinery (ACM), v. 20, n. 4, p. 422–446, out. 2002. Disponível em: <https://doi.org/10.1145/582415.582418>.

KIM, T.-H.; YANG, S.-B. An effective threshold-based neighbor selection in collaborative filtering. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. p. 712–715. Disponível em: <https://doi.org/10.1007/978-3-540-71496-5_75>.

KUMAR, A. et al. Comparison of various metrics used in collaborative filtering for recommendation system. In: *2015 Eighth International Conference on Contemporary Computing (IC3)*. IEEE, 2015. Disponível em: <https://doi.org/10.1109/ic3.2015.7346670>.

LECUN, Y. et al. Deep learning. *Nature*, Springer Science and Business Media LLC, v. 521, n. 7553, p. 436–444, maio 2015. Disponível em: <https://doi.org/10.1038/nature14539>.

LI, H. *Learning to Rank for Information Retrieval and Natural Language Processing*. Springer International Publishing, 2015. Disponível em: <https://doi.org/10.1007/978-3-031-02155-8>.

LIM, H.-I. A linear regression approach to modeling software characteristics for classifying similar software. In: *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE, 2019. Disponível em: <https://doi.org/10.1109/compsac.2019.00152>.

LIU, T.-Y. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, Now Publishers, v. 3, n. 3, p. 225–331, 2007. Disponível em: <https://doi.org/10.1561/1500000016>.

Lü, L. et al. Recommender systems. *Physics Reports*, Elsevier BV, v. 519, n. 1, p. 1–49, out. 2012. Disponível em: <https://doi.org/10.1016/j.physrep.2012.02.006>.

MAULUD, D.; ABDULAZEEZ, A. M. A review on linear regression comprehensive in machine learning. *Journal of Applied Science and Technology Trends*, Interdisciplinary Publishing Academia, v. 1, n. 4, p. 140–147, dez. 2020. Disponível em: <https://doi.org/10.38094/jastt1457>.

MITRA, B.; CRASWELL, N. An introduction to neural information retrieval t. *Foundations and Trends® in Information Retrieval*, Now Publishers, v. 13, n. 1, p. 1–126, 2018. Disponível em: <https://doi.org/10.1561/1500000061>.

NALLAMALA, S. H. et al. A brief analysis of collaborative and content based filtering algorithms used in recommender systems. *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, v. 981, n. 2, p. 022008, dez. 2020. Disponível em: <https://doi.org/10.1088/1757-899x/981/2/022008>.

NAQA, I. E.; MURPHY, M. J. What is machine learning? In: _____. *Machine Learning in Radiation Oncology: Theory and Applications*. Cham: Springer International Publishing, 2015. p. 3–11. ISBN 978-3-319-18305-3. Disponível em: <https://doi.org/10.1007/978-3-319-18305-3_1>.

PORTUGAL, I. et al. The use of machine learning algorithms in recommender systems: A systematic review. *Expert Systems with Applications*, Elsevier BV, v. 97, p. 205–227, maio 2018. Disponível em: <https://doi.org/10.1016/j.eswa.2017.12.020>.

POWERS, D. M. W. *Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation*. 2020.

RAY, S. A quick review of machine learning algorithms. In: *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*. IEEE, 2019. Disponível em: <https://doi.org/10.1109/comitcon.2019.8862451>.

SARWAR, B. et al. Item-based collaborative filtering recommendation algorithms. In: *Proceedings of the 10th international conference on World Wide Web*. ACM, 2001. Disponível em: <https://doi.org/10.1145/371920.372071>.

SERRANO, W. Neural networks in big data and web search. *Data*, MDPI AG, v. 4, n. 1, p. 7, dez. 2018. Disponível em: <https://doi.org/10.3390/data4010007>.

SHINDE, P. P.; SHAH, S. A review of machine learning and deep learning applications. In: *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. IEEE, 2018. Disponível em: <https://doi.org/10.1109/iccubea.2018.8697857>.

SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. *Information Processing &amp Management*, Elsevier BV, v. 45, n. 4, p. 427–437, jul. 2009. Disponível em: <https://doi.org/10.1016/j.ipm.2009.03.002>.

SUN, J. et al. A framework for recommending accurate and diverse items using bayesian graph convolutional neural networks. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery &amp Data Mining*. ACM, 2020. Disponível em: <https://doi.org/10.1145/3394486.3403254>.

SYMEONIDIS, P.; ZIOUPOS, A. *Matrix and Tensor Factorization Techniques for Recommender Systems*. Springer International Publishing, 2016. Disponível em: <https://doi.org/10.1007/978-3-319-41357-0>.

WANG, B. et al. *Neural Search - From Prototype to Production with Jina: Learn How to Build*. S.l.: PACKT PUBLISHING LIMITED, 2022.

ZHANG, S. et al. Deep learning based recommender system. *ACM Computing Surveys*, Association for Computing Machinery (ACM), v. 52, n. 1, p. 1–38, fev. 2019. Disponível em: <https://doi.org/10.1145/3285029>.

ZULKARNAIN, N.; ANSHARI, M. Big data: Concept, applications, &amp challenges. In: *2016 International Conference on Information Management and Technology (ICIMTech)*. IEEE, 2016. Disponível em: <https://doi.org/10.1109/icimtech.2016.7930350>.