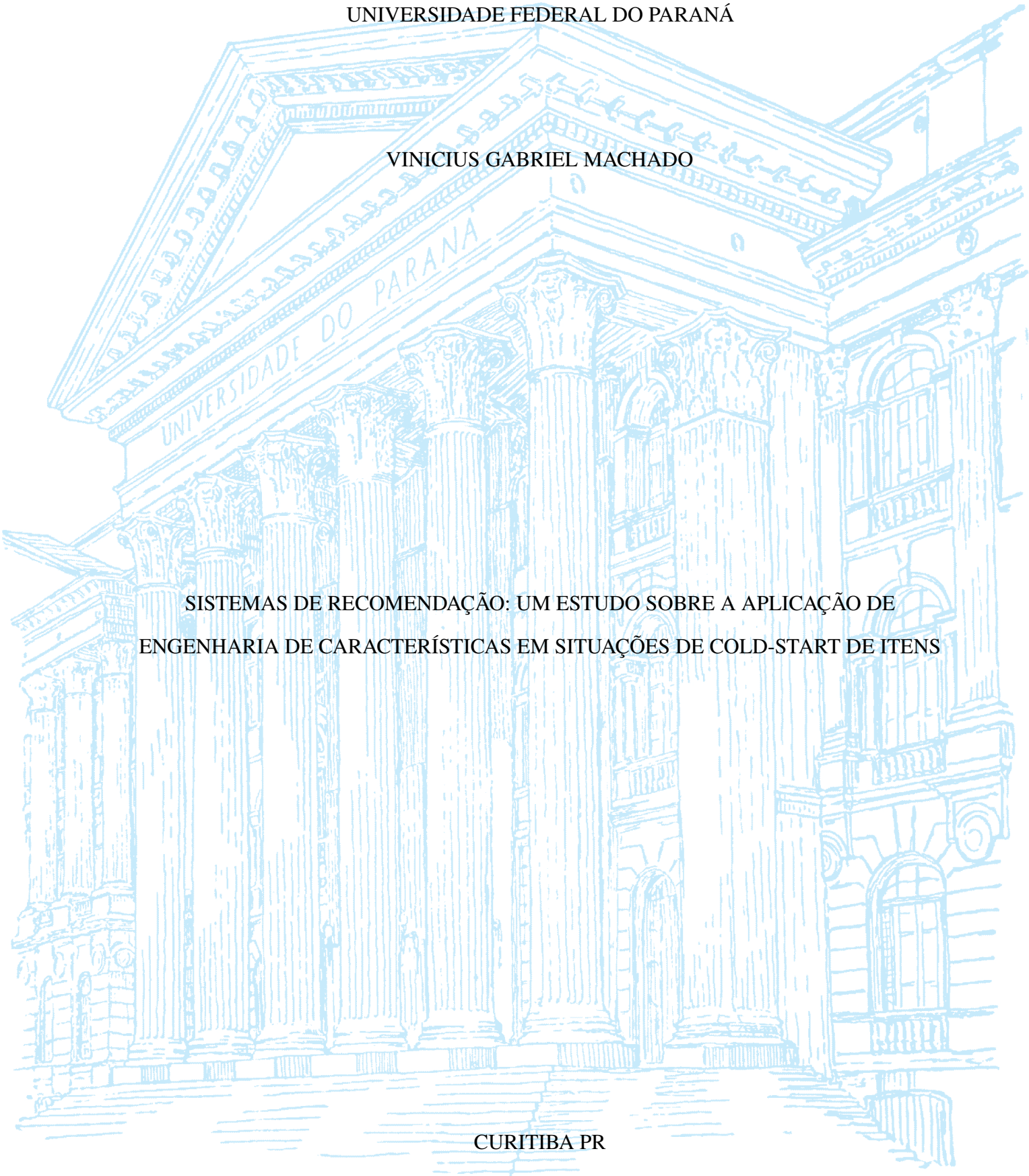
UNIVERSIDADE FEDERAL DO PARANÁ

VINICIUS GABRIEL MACHADO



SISTEMAS DE RECOMENDAÇÃO: UM ESTUDO SOBRE A APLICAÇÃO DE ENGENHARIA DE CARACTERÍSTICAS EM SITUAÇÕES DE COLD-START DE ITENS

CURITIBA PR

2023

VINICIUS GABRIEL MACHADO

SISTEMAS DE RECOMENDAÇÃO: UM ESTUDO SOBRE A APLICAÇÃO DE ENGENHARIA DE CARACTERÍSTICAS EM SITUAÇÕES DE COLD-START DE ITENS

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Eduardo Jaques Spinosa.

CURITIBA PR

2023

Dedico este trabalho de conclusão de curso a minha família e professoras(es) que conheci ao longo da jornada, que lutaram para que eu pudesse ter acesso a uma educação de qualidade.

AGRADECIMENTOS

Primeiramente gostaria de agradecer a minha família pelo suporte incondicional em todos os momentos da minha vida. Em diversos momentos duvidei da minha capacidade, me culpei duramente pelas minhas falhas e por cada acontecimento que não foi de acordo com o planejado. Em todos estes momentos eles estavam lá para me tranquilizar e me incentivar a acreditar em mim mesmo. É graças a eles e para eles que sigo em frente nesta jornada que é a vida, sempre buscando enchê-los de orgulho.

Agradeço ao professor Dr. Eduardo Jaques Spinosa pela sua excepcional orientação neste trabalho. Esta que foi humana quando necessário, compreendendo o estresse e as incertezas de um aluno nesta fase tão importante, mas que também em nenhum momento faltou com o profissionalismo exigido, sendo crítico quando necessário, mas principalmente, me incentivando a sempre fazer mais e melhor.

Gostaria de agradecer também aos meus amigos do curso de Ciência da Computação, com destaque para Christian Debovi e Gabriel Hermida, que me apoiaram, criticaram (merecidamente), mas que sobretudo, estiveram ao meu lado desde o começo, tanto na vida acadêmica quanto na pessoal. Neste momento estou terminando o curso, carregando comigo muitas coisas importantes como conquistas e aprendizados, mas principalmente, a nossa amizade.

Por último, mas não menos importante, gostaria de agradecer aos meus amigos e professores do grupo de pesquisa C3SL, que me proporcionaram nestes últimos anos um ambiente acolhedor, profissional, mas especialmente, um ambiente no qual todos podemos olhar um para o outro como iguais. Além disso, gostaria de também agradecer ao grupo responsável pelo C3HPC (*C3SL High Performance Computer*), que atenderam a minha solicitação de última hora e disponibilizaram acesso ao *cluster* para que eu pudesse expandir os resultados apresentados aqui neste trabalho.

RESUMO

Os meios digitais são praticamente ilimitados quando falamos de itens disponíveis para consumo, sendo assim os usuários estão em constante situação de *over-choice*, devido ao alto número de opções. Idealmente, uma plataforma online filtrará o seu catálogo, priorizando a exibição daquilo que é relevante para o usuário. Desta forma o usuário pode ter uma melhor experiência, a plataforma pode se destacar entre um mar de concorrentes, e o usuário pode ser inclinado a consumir algo desta plataforma. Partindo destas condições e objetivos, surgiram os sistemas de recomendação, que utilizando informações como interações anteriores entre usuários e itens, e dados de usuários e itens, sejam capazes de realizar recomendações apropriadas para cada usuário, realizando a grande filtragem necessária nos catálogos. Normalmente, sistemas que utilizam informações de interações são altamente eficazes em realizar boas recomendações. Porém, em certas situações como alta esparsidade e novos itens e usuários (*cold start*), os sistemas de recomendação são incapazes de realizar recomendações adequadas por falta de informações de interações. Desta forma, surgiu a ideia de utilizar as informações (características) de itens e usuários, buscando estabelecer relações entre os próprios itens e os próprios usuários. Sistemas deste tipo partem da ideia de que um usuário tende a ter interesse por itens semelhantes àqueles que já interagiu e que, usuários com gostos semelhantes tendem a interagir com itens semelhantes, logo, seria possível recomendar um item novo com que outro usuário com mesmos gostos já interagiu. Entretanto, utilizar estas informações brutas nem sempre é a escolha mais adequada já que certas informações podem não contribuir ou podem até mesmo piorar o desempenho dos sistemas de recomendação, impactando negativamente nas métricas ou então aumentando o tempo de execução. Partindo deste fato, iremos conduzir um estudo sobre sistemas de recomendação e *cold start* de itens, realizando experimentos em conjuntos de dados de filmes, livros e receitas, apresentando e utilizando técnicas de engenharia de características para transformar, tratar, selecionar e compreender melhor as características com o objetivo de extrair um maior potencial das informações na atenuação do *cold start* de itens. Tanto os dados originais quanto os modificados serão então utilizados no sistema de recomendação LightFM para realizar predições em conjuntos de teste divididos em itens conhecidos e novos, estas que serão avaliadas utilizando métricas como *auc_score* e *precision@k*. Por fim, iremos demonstrar que o problema de *cold start* de itens é extremamente grave e que é possível sim melhorar os resultados obtidos nesta situação utilizando características, realizando engenharia de características e outros ajustes. Entretanto, soluções eficazes são difíceis de se encontrar, além de impactarem negativamente em cenários de itens já conhecidos (com número significativo de interações), sendo necessário um tratamento diferente de acordo com o tipo de item.

Palavras-chave: Sistemas de Recomendação. Cold Start. Engenharia de Características.

ABSTRACT

Digital media are practically unlimited when talking about the items that we can consume, therefore, users are in a constant over-choice situation because of the high number of options. Consequently, an online platform needs to filter its catalog, displaying to the user only what is relevant to the user so the user can have a better experience, the platform can stand out from a sea of different competitors, and the user can be inclined to consume a product from this platform. With these conditions and objectives in mind, recommender systems emerged, and using information such as previous interactions between users and items, and users and items features, can make recommendations suitable for each user, performing the necessary filtering in the catalogs. Usually, recommender systems that use interaction data are highly effective in making good recommendations. However, in some situations like high sparsity and new items and users (cold start), recommender systems are unable to perform appropriate recommendations because of the lack of interaction data. With this in mind, the idea of using information (features) from items and users emerged, with the objective of better understanding relationships between items and users themselves. Systems of this type assume that a user tends to be interested in items that are similar to those with whom he already interacted and that, users with similar tastes tend to interact with similar items, so, it would be possible to recommend a new item that another user with similar preferences already interacted. However, using this raw information is not always the best decision since certain data may not contribute or may even be harmful to the performance of recommender systems, harming performance metrics or increasing the execution time. Based on this fact, we are going to conduct a study on recommender systems and item cold start, performing experiments on datasets of movies, books, and recipes, presenting and using feature engineering techniques to transform, process, select and get a better understanding of the features to extract greater potential from the information in the process of mitigation of the item cold start problem. Both original and modified data will then be used in the LightFM recommender system to create predictions on test subsets split between known items and new items, which will then be evaluated using metrics such as auc_score and precision@k. Finally, we are going to demonstrate that the item cold start is a serious problem and that it is possible to mitigate the effects using features, performing feature engineering, and other adjustments. However, effective solutions are hard to find and may impact negatively in scenarios of already known items (with a significant number of interactions), thus requiring the system to treat known and unknown items differently.

Keywords: Recommender Systems. Cold Start. Feature Engineering.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ACRÔNIMOS

SR	Sistema(s) de Recomendação
EC	Engenharia de Características
FC	Filtragem Colaborativa
FBC	Filtragem Baseada em Conteúdo
RAM	Random Access Memory

SUMÁRIO

1 INTRODUÇÃO

Ao desenvolvermos um Sistema de Recomendação (SR) precisamos de certas informações para que possamos realizar uma filtragem apropriada para cada usuário. Estas informações idealmente nos ajudam a entender melhor os usuários para os quais estamos recomendando. Em um cenário normal, dados como interações entre usuários e itens são extremamente eficazes em produzir recomendações com alto nível de acurácia (ou seja, recomendar para um usuário algo que ele provavelmente irá gostar), uma vez que estes dados nos ajudam a entender as preferências de cada usuário em um nível mais subjetivo, algo que dificilmente pode ser capturado com precisão nas informações do perfil de um usuário (que costumam ser de caráter genérico, como nome e idade). Com um melhor entendimento das preferências de cada usuário, podemos relacionar usuários pelos seus gostos e recomendar itens que um demonstrou interesse por, para outro que nunca interagiu com aquele item.

Entretanto, em certas situações não possuímos informações de interações, por exemplo, ao introduzirmos um novo item em um plataforma, um sistema baseado puramente em interações dificilmente irá recomendar este novo item, e em uma situação de novo usuário, não sabemos quais são suas preferências pois ele ainda não interagiu com nenhum item, problema conhecido como *cold start* de itens e usuários, respectivamente. Ou então, em um cenário onde uma plataforma possui muitos itens, interações se tornam esparsas já que dificilmente um usuário irá consumir uma quantidade significativa de itens, em comparação com a quantidade total, o chamado problema de esparsidade. Nessas situações, iremos realizar recomendações que possivelmente não serão apropriadas e/ou que não despertarão o interesse do usuário, podendo assim perder um possível cliente ou visualizador.

Nestas condições um SR pode utilizar as informações próprias dos itens e usuários (características) para melhorar as recomendações feitas. Utilizando por exemplo, uma lista de gêneros de um filme, música ou livro para itens e faixa etária para usuários. Com estes dados pode-se estabelecer relações entre os itens e entre os próprios usuários, que apesar de possuírem menor eficácia do que as relações adquiridas por interações, podem servir como solução temporária, até que o usuário ou item tenham adquirido uma quantidade de interações suficiente. Além disso, em um cenário que interações estão disponíveis, estas informações podem ajudar a melhorar as recomendações, mesmo que em uma proporção menor, permitindo realizar um ajuste fino ou contribuindo em situações de empate entre mais de um item.

Infelizmente, acrescentar novas variáveis ao processo de recomendação costuma ser custoso em termos computacionais, já que adicionamos novas camadas de complexidade, como a preparação destas informações para se adequarem ao modelo e a computação da similaridade entre os itens e usuários, que cresce quanto mais características são utilizadas. Desta forma, buscamos um balanço entre custo e benefício, e para isso realizamos uma Engenharia de Características (EC). Por meio deste processo, podemos estudar o custo das informações disponíveis, ou até mesmo entender e/ou melhorar a relação custo-benefício de cada uma. Ao fim de uma EC, podemos ter em mãos um SR com resultados semelhantes aos resultados obtidos com todas as características e com possivelmente um custo computacional menor. Entretanto, como podemos empregar uma EC com o objetivo de atenuar o problema de falta de interações em itens novos?

Este trabalho apresenta um estudo sobre sistemas de recomendação e EC, em um cenário de poucas interações, com o objetivo de adquirir um melhor entendimento sobre os sistemas e processos envolvidos. Para isso, apresentamos estudos realizados em diferentes conjuntos de dados de diferentes áreas de aplicação de SR, estes estudos buscam compreender melhor cada

tipo de dado, melhorar o custo benefício dos resultados e reduzir o impacto de situações de *cold start* de itens.

O documento está organizado da seguinte forma: Capítulo 2 define em maiores detalhes o foco deste documento e os objetivos gerais e específicos. Capítulo 3 provê uma visão geral sobre SR e engenharia de características. Capítulo 4 discute os materiais utilizados, como conjuntos de dados, algoritmo de recomendação e algoritmos de EC. Capítulo 5 por sua vez apresenta os experimentos realizados e seus resultados. Por fim Capítulo 6 conclui este documento.

2 ESTUDO PROPOSTO

Sendo um problema atual e ainda com muito potencial para estudos na área, o problema de *cold start* de itens foi escolhido para ser a área de foco deste trabalho. Como já apresentado de forma breve anteriormente, os problemas de *cold start* se devem à introdução de novos itens nos catálogos (*item cold start*), mas também à introdução de novos usuários nos sistemas (*user cold start*), situações que são extremamente comuns no mundo digital em constante expansão.

Dito isso, este trabalho irá utilizar uma abordagem dirigida ao estudo dos dados, buscando realizar uma EC a fim de entender e melhorar os resultados obtidos em diferentes cenários de *cold start* de itens.

Para isso utilizamos diferentes conjuntos de dados de diferentes domínios, dados de receitas, filmes e livros (conjuntos de dados são descritos detalhadamente na Seção 4.1), contendo características de itens e interações, e simulamos (ou agravamos) condições de *cold start* de itens removendo interações propositadamente (experimentos discorridos no Capítulo 5). Nestas condições, inserimos os dados no SR LightFM (Kula, 2015) – um algoritmo de recomendação híbrido, que combina fatoração de matrizes (FC) com estratégias de FBC, para predizer interações faltantes como produto da média das representações latentes do usuário e do item envolvidos (mais detalhes na Seção 4.2). Observamos os resultados obtidos com os dados originais, com e sem características, para 3 conjuntos diferentes de interações presentes no conjunto de teste, interações de itens frios, de itens quentes e todas interações de teste. Partindo disso, utilizamos de técnicas de EC, para filtrar características, melhorar características existentes e ajustar os parâmetros do algoritmo, a fim de extrair o máximo de potencial das informações dos itens, desejando atenuar o problema de *cold start*. Os resultados são então computados, com os mesmos parâmetros do LightFM utilizados nos resultados originais, analisados e comparados.

De forma concisa os objetivos deste trabalho são:

- Objetivo Geral: Utilizar uma abordagem voltada ao estudo dos dados, usando técnicas e métodos de engenharia de características a fim de entender e melhorar a qualidade de recomendações em cenários de *cold start* de itens;
- Objetivos Específicos:
  - Apresentar sistemas de recomendações e os tipos normalmente observados na literatura;
  - Contextualizar o leitor em situações de *cold start*;
  - Apresentar uma contextualização geral sobre engenharia de características;
  - Utilizar conjuntos de dados de diferentes cenários para promover uma análise do *cold start* de itens em diferentes cenários;
  - Utilizar diferentes técnicas e métodos de engenharia de características para modificar as características dos itens, buscando melhorar os resultados obtidos em *cold start* de itens;
  - Realizar experimentos utilizando diferentes métricas e um sistema de recomendação apropriado, analisando o desempenho em situações de itens conhecidos e não conhecidos, e comparando com os resultados obtidos com os dados originais com e sem características.

3 SISTEMAS DE RECOMENDAÇÃO

Nos dias atuais é impossível negar a grande digitalização que o mundo inteiro sofreu, com tudo que é consumível, desde músicas, livros, filmes, artigos, notícias, e até mesmo supermercados e lojas inteiras, presentes na internet. Além de permitir o fácil acesso a qualquer produto, o mundo digital também possibilitou a expansão ilimitada dos catálogos, já que a limitação de espaço é muito menos restritiva e muito mais fácil de ser contornada no ambiente digital do que no real.

Entretanto, em meio a um mar de opções, os usuários muitas vezes se veem em meio a um ambiente de *over-choice* (Zhang et al., 2019), no português, muitas possibilidades de escolha. Consequentemente, é necessário filtrar de algum modo as opções apresentadas aos usuários, para que eles tenham uma melhor experiência e sejam motivados a consumir algo fornecido por uma plataforma específica. Tal necessidade é decorrente do fato de que juntamente com as bibliotecas *online*, a concorrência também aumenta a cada dia (Vaidya e Khachane, 2017), desta forma, as plataformas estão em constante competição pela atenção de possíveis consumidores e agradá-los é um meio de obter isso.

A filtragem dos catálogos, de preferência, deve extrair itens relevantes para um usuário, levando em conta conhecimentos adquiridos sobre ele, como possíveis interesses, experiências passadas naquele mesmo ambiente ou em outros, localização, idade, poder de consumo, condições sócio-culturais, momento atual, entre outros fatores. Uma filtragem ruim, por outro lado, pode acarretar em um efeito negativo no consumidor, que pode não encontrar nada que lhe agrade, e talvez, optar por não utilizar um serviço, já que não pareceu ter itens relevantes para ele.

Desta forma, surge a ideia de SR que, como o nome sugere, são os responsáveis por, utilizando informações conhecidas, prover recomendações adequadas a cada usuário, realizando a grande filtragem necessária nos catálogos de consumo. Sendo assim, neste capítulo será abordada a teoria por trás destes sistemas. Começando por uma visão geral de SR, a partir de uma análise dos diferentes tipos observados na literatura (Seção 3.1). Em seguida, abordando um dos problemas comumente enfrentado por esses sistemas, recomendação de novos itens e recomendação para novos usuários (Seção 3.2). Finalizando com uma discussão sobre o processo de extração de características (Seção 3.3) que envolve obtenção, extração, engenharia e seleção de características, que serão de fundamental importância no estudo proposto neste trabalho.

3.1 TIPOS DE SISTEMAS DE RECOMENDAÇÃO

Na prática, SR precisam determinar a utilidade de cada item disponível para os usuários e selecionar aqueles que tiverem as melhores pontuações. Um algoritmo de recomendação simples pode, por exemplo, recomendar apenas as músicas mais populares de um aplicativo de *streaming* de músicas. Esta recomendação não está necessariamente errada, mas busca atender um perfil genérico de usuário (Ricci et al., 2015). Entretanto, ela não será capaz de explorar preferências de maior nicho. Um algoritmo mais personalizado, por outro lado, pode perceber o apreço de uma pessoa por músicas de gêneros menos conhecidos e até mesmo de gêneros mais conhecidos, combinando os benefícios de ambas as soluções.

Na literatura, o tema SR é altamente pesquisado, já que é uma necessidade cada vez mais presente no dia-a-dia de todos. Desta forma, diferentes tipos de soluções já foram propostas, cada uma atendendo um diferente domínio, diferente algoritmo utilizado e utilizando diferentes informações, para fornecer uma recomendação personalizada. As próximas seções fornecem

uma visão sobre os tipos de sistemas mais comuns, uma taxonomia proposta por Burke (2007) e compilada por Ricci et al. (2015).

3.1.1 Filtragem Baseada em Conteúdo

Este tipo de recomendação (Filtragem Baseada em Conteúdo ou FBC) se baseia na ideia de que um usuário tem altas chances de gostar de itens semelhantes àqueles pelos quais ele já possui apreço. Neste sentido, a utilidade de um item é determinada pela semelhança com outros, aqueles com os quais o usuário já interagiu, e ela é determinada pela correlação entre as características que definem os itens. No exemplo utilizado anteriormente, de um catálogo de músicas, uma recomendação desse tipo pode perceber que um usuário gosta de músicas do gênero *rock* e recomendar outras músicas do mesmo estilo. Entretanto há alguns problemas, como a necessidade de que o usuário já tenha interagido com uma quantidade significativa de itens para que um padrão possa ser observado, e diferente da filtragem colaborativa (seção seguinte), não há compartilhamento de informações entre usuários. Logo, as recomendações feitas para um usuário estarão restritas aos itens com os quais ele interagiu.

3.1.2 Filtragem Colaborativa

Dentre os tipos de filtragem, este é provavelmente o mais utilizado e o que possui alguns dos melhores resultados em termos de efetividade. Ao realizar uma filtragem deste tipo (Filtragem Colaborativa ou FC), um SR busca relacionar usuários pelas suas preferências aprendidas por meio de interações, logo extraindo informações mais intrínsecas, subjetivas, algo que seria difícil de captar por meio de características no perfil dos usuários. Novamente utilizando o exemplo mencionado anteriormente, podemos imaginar uma correlação de usuários que gostam do gênero *rock*. Após correlacionar um usuário com este grupo, um sistema pode recomendar itens que os outros usuários do grupo gostam como *heavy metal*, partindo do princípio de que se eles possuem uma preferência em comum, eles podem ter outras preferências em comum.

Este modelo normalmente faz uso de uma matriz usuários X itens, onde cada entrada da matriz representa a interação entre um usuário específico e um item específico. Com duas diferentes abordagens sendo utilizadas em associação com esta representação, como apontado por Burke (2002):

- *Memory-based* ou *Neighborhood-based*: A matriz de interações, salva na memória, é utilizada diretamente para predizer interações que não ocorreram. Esta técnica é realizada de duas formas diferentes (Shah et al., 2017):
  - As baseadas em usuários procuram encontrar outros usuários que tenham avaliado os mesmos itens de forma parecida;
  - As baseadas em itens buscam predizer a avaliação de um usuário para um item, utilizando avaliações já feitas de itens semelhantes. Neste modelo, a similaridade entre dois itens é calculada como sendo produto do cálculo das avaliações semelhantes já existentes entre eles.
- *Model-based*: Este método utiliza-se das informações de interações entre usuário e itens para treinar modelos que possam simular estas mesmas interações. Um dos modelos mais utilizados e bem-sucedidos é conhecido por fatoração de matrizes (*matrix factorization*) (Koren et al., 2009). Este modelo representa os usuários e itens em um espaço latente. Um espaço de n dimensões (normalmente de menor dimensionalidade)

onde itens e usuários assumem posições relativas a semelhança entre eles, induzidas por meio de seus fatores latentes, a semelhança entre suas características. As informações de interações são obtidas como o produto de item por usuário. Por fim, funções de aprendizado são utilizadas a fim de predizer as interações faltantes, para isso realizam um processo iterativo incremental onde buscam ajustar seus parâmetros até se tornarem capazes de prever as interações existentes. Ao fim desse processo, o sistema se torna capaz de determinar a probabilidade de um usuário gostar de um item com o qual nunca interagiu.

A falta de muitos registros na matriz de interações, é conhecida como problema de esparsidade e se refere a falta de conhecimento sobre as preferências dos usuários. Normalmente decorre do fato de haver um grande volume de itens (Bobadilla et al., 2013), uma vez que quanto maior a quantidade de itens, menor a probabilidade de um usuário ter interagido com uma proporção significativa dos itens.

Da mesma forma, itens sobre os quais o sistema não possui informações de interação, normalmente itens recém adicionados ao catálogo, são chamados de itens frios (*cold items*) e são a causa do problema de *cold start* de itens. Este problema é muito comum em grandes plataformas, onde itens são adicionados a todo momento e que pela falta de interação não são recomendados aos usuários. Isso por sua vez resulta no problema de cobertura limitada (*limited coverage*), onde o sistema é incapaz de recomendar todos os itens para algum usuário (Nikolakopoulos et al., 2021). No sentido de usuários o mesmo também ocorre, o chamado problema de *cold start* de usuários se refere ao problema de realizar recomendações para um usuário novo em uma plataforma, que pela falta de interações do usuário com itens, são incapazes de realizar uma recomendação adequada.

Nessas 3 situações (esparsidade, *cold start* de item e de usuário) SR baseados em FC são altamente ineficazes, sendo superados pelos sistemas que utilizam FBC.

3.1.3 Demográficos

Uma abordagem mais simples e eficaz é a recomendação baseada em informações demográficas. Esta solução parte da ideia de que o ambiente, condições socioculturais, econômicas, entre outros, tem papel de importância na constituição das escolhas e preferências de um usuário. Logo, é compreensível realizar recomendações baseadas nestas informações de um usuário. Por exemplo, uma boa solução pode perceber que usuários com menos de 20 anos tendem a ter um interesse maior por música eletrônica do que por *jazz*. Ou então, esta solução pode perceber que um usuário de um país que não tem inglês como uma língua materna, tem um interesse maior por músicas em outras línguas, como a sua língua nativa.

3.1.4 Baseados em Conhecimento

SR deste tipo possuem o objetivo de responder a uma necessidade, utilizando conhecimento específico em uma área. Por exemplo, o sistema Entree (Burke, 2002) altamente referenciado na literatura, é um dos melhores exemplos deste tipo de filtragem. Nele um usuário insere quais seriam as características de um restaurante que ele gostaria, por exemplo, um que tenha uma boa atmosfera, um bom preço e que sirva comida mexicana. Além disso, ele também pode informar ao sistema um outro restaurante que seja semelhante ao que ele quer que seja recomendado. Com estas informações, o algoritmo realiza um filtro no catálogo de restaurantes conhecidos, buscando aqueles semelhantes ao que o usuário busca.

Na filtragem baseada em conhecimento há 3 subdivisões: soluções baseadas em caso, baseadas em restrições e baseadas em comunidade. As duas primeiras utilizam das mesmas informações para tomar decisões, mas diferem no método para calcular as soluções ideais. Soluções baseadas em caso utilizam métricas de similaridade para correlacionar os requisitos com as possíveis respostas, já as baseadas em restrições possuem regras explícitas para definir estas mesmas relações. O terceiro tipo, baseadas em comunidade, realizam recomendações utilizando informações sobre os gostos da comunidade na qual o usuário está inserido, como por exemplo, itens que seus amigos de uma rede social demonstraram interesse por. Esta solução tem tido grande desenvolvimento, tendo em vista a popularização das redes sociais, tanto que tem sido referenciada como *Social Recommender Systems* (sistemas de recomendação sociais) (Golbeck, 2006).

3.1.5 Híbrido

Este tipo de SR por sua vez tem como objetivo combinar as diferentes técnicas abordadas nas seções anteriores para fornecer uma recomendação mais robusta e flexível. Utilizando uma solução híbrida é possível, por exemplo, lidar com a dificuldade em recomendar itens novos, sem interações, que os sistemas baseados em FC possuem, utilizando um sistema FBC. Desta forma, um sistema FBC seria utilizado para itens novos e quando estes itens tiverem adquirido uma quantidade suficiente de avaliações, as recomendações passarão a serem feitas pelo sistema baseado em FC.

Existem diversas formas de realizar a combinação de duas ou mais técnicas, Burke (2002) define uma taxonomia para essas formas:

- *Weighted*: Diferentes SR produzem recomendações e ao fim, os resultados são combinados com um peso atribuído a cada sistema;
- *Switching*: Diferentes SR são utilizados, mas o sistema escolhe aquele que for mais apropriado para cada situação;
- *Mixed*: As recomendações produzidas por diferentes sistemas são combinadas em uma única;
- *Feature Combination*: Diferentes SR contribuem para melhorar as características de itens, o resultado é utilizado como a entrada para um **SR** principal;
- *Feature Augmentation*: Um ou mais SR produzem novas características para os itens, que então servem de entrada para o SR principal;
- *Cascade*: Uma estrutura hierárquica de SR é estabelecida, com o sistema mais abaixo sendo utilizado para casos onde há o empate entre 2 ou mais possíveis recomendações;
- *Meta-level*: Diferentes sistemas são combinados de tal forma que, a saída de um é um modelo, esse então é colocado na entrada do sistema principal.

3.2 COLD START

Devido à falta de interações de um novo usuário ou de interações registradas em um novo produto, sistemas baseados em **FC** possuem dificuldade em realizar uma filtragem apropriada. Entretanto, é de extrema importância que ela seja feita. Consequentemente, sistemas são obrigados a explorar diferentes fontes de informação, como as obtidas nas características

de um item (*item cold start*) ou então solicitar informações diretamente para o usuário (*user cold start*). Abordagens que mesmo sendo efetivas, possuem eficácia menor do que as que se aproveitam de interações anteriores, além de serem intrusivas ao usuário em algumas situações, como ao solicitar o preenchimento de um formulário (Bernardis e Cremonesi, 2021).

Diversas estratégias já foram propostas para lidar com o problema de *cold start*, Panda e Ray (2022) apresentam uma análise das abordagens já propostas e estabelecem uma taxonomia para elas:

- *Data-driven Strategies*: Visam extrair o máximo de efetividade das informações dos dados existentes, das relações entre itens e usuários e buscam incrementar as informações com outras fontes, para dessa forma aliviar o problema. Para realizar isso utilizam técnicas de *feature engineering* (EC) e agregam outras informações como localização, redes sociais, dispositivos inteligentes, etc. Deste modo, é possível enriquecer o perfil de um usuário, simular e presumir interações, agrupar usuários, estabelecer correlações entre usuários e medir o nível de confiança das informações que ele agrega a plataforma, entre outras possibilidades.
- *Approach-driven Strategies*: Buscam propor novos modelos para solucionar o problema, utilizando-se da combinação de diferentes métodos e SR. Estes sistemas propostos, podem ser divididos em 5 grupos principais, segundo os autores:
  - *Deep Learning-Based Approaches*: Abordagens que têm se popularizado na última década, que utilizam-se de *deep learning* para extrair correlações e características implícitas – e consequentemente difíceis de serem notadas em outros métodos – entre usuários e itens, e predizer interações entre eles;
  - *Matrix Factorization Based Approaches*: Abordagens que utilizam técnicas de fatoração de matrizes para melhorar o desempenho dos algoritmos. Por meio da decomposição da matriz de interação (como a utilizada em FC) em um espaço latente, é possível extrair características latentes e aliado a uma função de aprendizado, prevendo valores de interação para interações que não ocorreram, é possível gerar recomendações;
  - *Hybrid Recommender System Based Approaches*: Estas utilizam da hibridização de sistemas FBC, baseados em FC e outros, para fornecer um SR mais robusto;
  - *Novel Approaches in Collaborative Filtering*: Abordagens que exploram a aplicação de algoritmos como, *clustering*, KNN e algoritmos genéticos, buscando encontrar novas soluções na área de FC;
  - *Novel Approaches in Content-based Recommender Systems*: Abordagens que exploram relações semânticas, características de personalidade derivadas e diferentes representações do espaço de característica de itens, a fim de definir novas abordagens na área de sistemas FBC.

3.3 EXTRAÇÃO DE CARACTERÍSTICAS

Como mencionado anteriormente, é possível melhorar os resultados de recomendações ao levar em consideração as características dos itens e usuários envolvidos, podendo até mesmo obter resultados melhores do que os sistemas FC em situações de esparsidade e *cold start*. Ao fazer uso de sistemas FBC ou híbridos, que se aproveitem de tais características. Sendo assim, as seguintes subseções irão explorar alguns processos que cooperam com o objetivo de obter o

máximo de potencial de características. Começando pela obtenção de características (Subseção 3.3.1) e finalizando por engenharia de características (Subseção 3.3.2).

3.3.1 Obtenção de Características

Inicialmente é necessário a obtenção de características, que normalmente são binárias, categóricas ou contínuas, e que buscam representar um conjunto de informações da melhor forma possível. Entretanto, o processo de encontrar boas representações é altamente específico ao domínio envolvido e as métricas utilizadas (Guyon e Elisseeff, 2006).

Em termos de características de usuários, por exemplo gêneros preferidos de filmes, podemos obter as características por meio de questionários, solicitando diretamente ao usuário quais seus gêneros ou filmes favoritos. Podemos também presumir preferências com base em informações demográficas ou então podemos extrair de outras fontes, como redes sociais conectadas ao perfil do usuário. Já em relação a itens, muitas vezes há uma curadoria responsável por registrar os itens. Uma solução é utilizar as informações inseridas por essa curadoria como características (extraindo características utilizando a experiência e conhecimento humano). Em cenários onde não há uma curadoria ou a quantidade de itens é elevada, impossibilitando ou inviabilizando a utilização de humanos na obtenção de características, ou até mesmo, quando se deseja complementar o trabalho humano, é possível desenvolver métodos automatizados para extração de características.

Esses métodos de extração de características costumam ser específicos para cada tipo de dado (texto, imagem, vídeo, áudio, etc.) e domínio do SR (recomendação de música, *podcast*, artigo, livro, filme, notícia, etc.). Por exemplo, Deldjoo et al. (2020) cita diferentes abordagens para extração de características de filmes, como análises de características de uma imagem, buscando comparar filmes pelo *poster*, ou então análise *frame* a *frame* dos *trailers* e/ou de trechos de áudio das obras. Outro exemplo mencionado pelos autores é a utilização de redes neurais pré-treinadas, como *ImageNet*, para extrair características visuais de imagens de locais, que aliadas a informações textuais e de localização, obtêm bons resultados na recomendação de pontos turísticos.

3.3.2 Engenharia de Características

Engenharia de características (EC) é o processo de utilizar conhecimento em um domínio para extrair características de dados brutos. O processo, como descrito por Dong e Liu (2018), envolve:

- Transformação de Características: Processo em que se realiza transformações matemáticas, a fim de adquirir diferentes representações para o mesmo dado. Exemplo, em dados de interações explícitas, é possível considerar apenas notas maiores ou iguais a 4, como interações positivas;
- Geração/Extração de Características: Processo em que se gera novas características, que não seriam possíveis de alcançar com transformações. Por exemplo, análise de música, identificando componentes intrínsecos como ritmo e associando com certas emoções provocadas nos usuários;
- Seleção de Características: Processo no qual busca-se extrair um conjunto reduzido de características de um conjunto maior. De forma automatizada, é possível utilizar técnicas *Wrapper*, *Filter* ou *Embedded* (Guyon e Elisseeff, 2006; Venkatesh e Anuradha,

2019) para remover características. Uma filtragem pode ajudar a reduzir a complexidade computacional de um SR, ao custo de uma eventual redução na acurácia;

- Análise e Avaliação de Características: Se refere a conceitos, métodos e métricas para mensurar a relevância e utilidade de conjuntos de características. Normalmente está diretamente relacionado à seleção de características. Por exemplo, análise de correlação;
- Geração Automatizada de Características: Abordagens que permitem gerar um grande volume de características e selecionar um subconjunto delas, de forma automatizada. Como exemplo, podemos pensar em análise de vídeo por meio de aprendizado de máquina, buscando identificar os elementos presentes e catalogar a quantidade de vezes que cada elemento é exibido;
- Análise de Dados: Utilização de métodos de engenharia de características para análise de dados em contextos específicos. Como análise de dados de uma rede social, para medir a resposta do público a determinados acontecimentos, como anúncio de certos fatos.

4 MATERIAIS E MÉTODOS

Este capítulo tem como objetivo explorar os instrumentos utilizados para a realização do estudo proposto, que por sua vez busca analisar o impacto de diferentes características no contexto de *cold start* de itens em SR. As seções estão organizadas da seguinte forma: Seção 4.1 apresenta os conjuntos de dados selecionados. Seção 4.2 discute o algoritmo de recomendação utilizado nos experimentos.

4.1 CONJUNTOS DE DADOS

Para a realização do estudo sugerido é necessário conjuntos de dados adequados. Diversas possibilidades foram avaliadas, muitas falharam em cumprir os seguintes requisitos:

- Quantidade significativa de características: Idealmente deseja-se dados que contenham itens com uma quantidade média para alta de características, para a realização de uma análise mais aprofundada.

- Características apropriadas: Busca-se dados que contenham características que, além de representar o produto real, sejam palpáveis para o ser humano compreender seu impacto em uma recomendação. Por exemplo, em um conjunto de dados sobre livros, dados como título, ano de publicação, autor, gêneros, número de páginas seriam interessantes para o estudo. Enquanto que dados como estatísticas de visualização e compra, ou informações da estética do produto, que apesar de poderem ser importantes em outros estudos, acredita-se que não seriam suficientemente concretos para uma análise humana.

- Interações explícitas e não implícitas: Interações implícitas são altamente importantes para os SR atuais, uma vez que elas são abundantes se comparadas com as explícitas. E que além disso, podem apresentar maiores informações sobre as preferências dos usuários (Jawaheer et al., 2010). Já que o fato de um usuário não avaliar ou avaliar um item, não necessariamente quer dizer que ele não goste de um item que não avaliou, ou que ele goste mais de um que avaliou do que de outros que não avaliou. Entretanto, para fins de delimitação dos experimentos realizados neste trabalho, buscou-se padronizar um tipo de interação, e como a maioria dos conjuntos de dados que julgou-se serem interessantes continham interações explícitas, este foi o padrão adotado.

- Dados apropriados para uma análise de preferências de longo prazo: Como abordado anteriormente, em SR o objetivo é fornecer ao usuário uma experiência concisa e que idealmente seja compatível com suas preferências, para tal, devemos ser capazes de entender essas preferências. Dito isso, devemos entender que suas escolhas são uma representação dos seus gostos de longo prazo e de curto prazo (dependentes do momento). Desta forma, diferentes vertentes surgiram. A primeira e a mais observada na literatura, utiliza-se de todos os dados para definir as preferências dos usuários, partindo do pressuposto de que todas as interações representam os gostos do usuário com mesmo peso, provendo uma visão das escolhas de longo prazo. Isto pode não necessariamente representar a realidade em sua totalidade, por exemplo, ao consumir um serviço de *streaming* de músicas, podemos optar por músicas de diferentes gêneros dependendo da hora do dia. Desta forma, surgiu a ideia de estudar as interações como

sendo dependentes das sessões em que ocorrem, tornando os SR mais flexíveis, sendo capazes de entender as preferências de longo prazo e também de se adaptar ao dinamismo do dia-a-dia de um usuário (Wang et al., 2021). Entretanto, estudos baseados em sessões ainda são recentes e conjuntos de dados deste tipo são escassos, ainda mais escassos são os que atendem as características desejadas, sendo assim, optou-se por utilizar dados apropriados para uma análise de longo prazo.

Considerando todos esses fatores, dezenas de opções de conjuntos de dados foram cogitadas e rejeitadas, algumas por possuírem apenas dados de interações, outras por conterem características inadequadas para uma análise humana (como estatísticas de interação e uso) e o restante por possuírem estritamente interações implícitas, um escopo totalmente diferente do abordado neste trabalho. A seguir serão discutidas aquelas que foram escolhidas. A Subseção 4.1.1 explora um conjunto de dados criado pelo autor, que representa a combinação de dados de interações de usuários com filmes e séries da plataforma MovieLens, com características extraídas do conjunto de dados da plataforma IMDb. Já a Subseção 4.1.2 apresenta o conjunto de dados de receitas Foodcom. Por fim, o terceiro conjunto de dados é apresentado na Subseção 4.1.3, este contêm dados de livros da plataforma Goodreads.

4.1.1 MovieLens + IMDb

Por ser um dos conjuntos de dados mais utilizados na literatura e por conter uma impressionante quantidade de interações, os dados do MovieLens (Harper e Konstan, 2015) se destacam. Entretanto, em seu estado original, os itens não possuem muitas características, possuindo apenas uma lista de gêneros e um título.

Em iterações mais recentes, estes dados são acompanhados pelo chamado Tag Genome (Vig et al., 2012), um conjunto de milhares de rótulos, onde a relevância de cada rótulo é calculada para cada item utilizando técnicas de *Supervised Learning*. O resultado é uma matriz de Rótulos X Itens números reais, variando de 0,0 à 1,0, com valores mais próximos de 1,0 indicando afinidade do rótulo com o item e valores mais próximos de 0,0 indicando a falta de relação entre o item e o rótulo. Porém, pela forma como estes dados são obtidos, isto é, por cálculos de uma rede neural que tenta aproximar a forma como um ser humano rotularia os itens, optou-se por não utilizá-los, já que uma nova camada de dúvida seria adicionada a esse estudo, na qual nos questionaríamos sobre o quanto esses valores representam a real classificação de um humano. Além disso, a complexidade dos experimentos também seria aumentada exponencialmente, já que seria necessário um pré-processamento deste grande volume de informações, agrupando-os em tipos de informações. Por exemplo, no Tag Genome é possível obter a correlação de um item com diferentes gêneros de filmes. Entretanto, estes dados estão espalhados em diferentes rótulos, que deveriam ser agrupados em um único rótulo "Gêneros", para que o impacto da informação de gênero em itens possa ser estudado.

Apesar disso, notou-se que os dados do MovieLens além de conter seus próprios índices de identificação, também continham o índice equivalente na plataforma IMDb, que disponibiliza conjuntos de dados para uso não comercial[1]. Alguns deles contêm diversas informações sobre filmes e programas de TV. Desta forma, surgiu a ideia de combinar os itens do MovieLens com as informações destes mesmos itens no IMDb e para isso foram criados alguns *scripts* para combinação dos dados.

Ao realizar a união dos dados, inicialmente coletou-se dos dados do IMDb apenas as entradas que também estavam presentes no MovieLens. Ao realizar isso notou-se que dos 62.423

[1]https://developer.imdb.com/non-commercial-datasets/

itens do MovieLens, 107 não estavam presentes no IMDb. Logo, estes foram removidos dos dados do MovieLens, informações e interações.

As Tabelas 4.1 e 4.2 exibem em maiores detalhes os dados disponíveis nos dados do MovieLens e do IMDb respectivamente, já a Tabela 4.3 mostra as informações do conjunto de dados resultante da combinação.

MovieLens		
Itens:	Descrição:	Filmes, curtas, etc.
	Quantidade:	62.423
	Características	
	Nome	Tipo
	Título	Textual - String
	Ano	Numérica - Real
	Gêneros	Lista não estruturada - Array de Strings
Interações:	Quantidade:	25.000.095
	Características:	Nota atribuída
	Tipo:	Explícita até 5
Usuários:	Selecionados aleatoriamente, com no mínimo 20 itens avaliados	
Dados de:	Janeiro de 1995 à Novembro de 2019	

Tabela 4.1: Informações sobre o conjunto de dados do MovieLens.

IMDb		
Itens:	Descrição:	Informações sobre filmes, séries, etc.
	Quantidade:	9.826.265
	Características	
	Nome	Tipo
	Tipo de Obra	Categórica - String
	Título Principal	Textual - String
	Título Original	Textual - String
	Destinado a Adultos	Categórica - Boolean
	Ano de Início (ou lançamento)	Numérica - Real
	Ano de Fim	Numérica - Real
	Duração (min)	Numérica - Real
	Gêneros	Lista não estruturada - Array de Strings
	Diretores	Lista não estruturada - Array de Strings
	Escritores	Lista não estruturada - Array de Strings
	Equipe de Produção	Lista não estruturada - Array de Objetos
Dados de:	3 de Maio de 2023	

Tabela 4.2: Informações sobre o conjunto de dados do IMDb.

4.1.2 Foodcom

O segundo conjunto de dados escolhido foi o Foodcom (Majumder et al., 2019). Nestes dados estão presentes informações sobre receitas e avaliações explícitas das mesmas, coletadas da plataforma food.com. A Tabela 4.4 explora as informações disponíveis. Vale observar que a

MovieLens + IMDb		
Itens:	Descrição:	Informações sobre filmes, séries, etc.
	Quantidade:	62.316
	Características	
	Nome	Tipo
	ID, IMDb ID	Categórica - String
	Tipo de Obra	Categórica - String
	Título Principal	Textual - String
	Título Original	Textual - String
	Destinado a Adultos	Categórica - Boolean
	Ano de Início (Lançamento)	Numérica - Real
	Ano de Fim	Numérica - Real
	Duração (min)	Numérica - Real
	Escritores	Lista não estruturada - Array de Strings
	Diretores	Lista não estruturada - Array de Strings
	Gêneros	Lista não estruturada - Array de Strings
	Equipe de Produção	Lista não estruturada - Array de Objetos
Interações:	Quantidade e Esparsidade:	24.983.560 - 99,8%
	Características:	Nota atribuída
	Tipo:	Explícita até 5
Usuários:	162.541 - Selecionados aleatoriamente, com no mínimo 20 itens avaliados	
Dados de:	Janeiro de 1995 à Novembro de 2019 (interações) e 3 de Maio 2023 (itens)	

Tabela 4.3: Informações sobre o conjunto de dados final, resultante da combinação MovieLens + IMDb.

coluna de nutrição é dividida em 7 categorias (7 valores reais): calorias, gordura total, açúcar, sódio, proteína, gordura saturada e carboidratos.

4.1.3 Goodreads

O terceiro e último conjunto de dados escolhido foi o Goodreads (Wan e McAuley, 2018), da plataforma Goodreads. Este contém informações de livros adicionados a estantes públicas de usuários. Os itens contêm diversas características descritivas e estatísticas, as interações são divididas em implícitas e explícitas, essas por sua vez possuem número de votos atribuído a elas. Para este trabalho apenas as interações explícitas serão utilizadas e todas terão o mesmo peso (avaliação de outros usuários não afetará o peso da interação) para padronização. A Tabela 4.5 apresenta em maiores detalhes o conjunto de dados.

4.2 FERRAMENTA DE RECOMENDAÇÃO

Para a realização dos experimentos do estudo, o algoritmo de recomendação LightFM (Kula, 2015) foi escolhido pelas diversas funcionalidades oferecidas. A Subseção 4.2.1 tem como objetivo esclarecer seu funcionamento para prover um melhor entendimento sobre a ferramenta responsável pelas recomendações nos experimentos. Enquanto que a Subseção 4.2.2 apresenta as principais funcionalidades e características da biblioteca.

Foodcom		
Itens:	Descrição:	Receitas com no mínimo 3 passos
	Quantidade:	231.637
	Características	
	Nome	Tipo
	Nome	Textual - String
	Descrição	Textual - String
	ID	Categórica - String
	Tempo de Preparo (min)	Numérica - Real
	Usuário que Publicou	Categórica - String
	Data de Publicação	Data - String
	Rótulos	Lista não estruturada - Array de Strings
	Nutrição	Lista não estruturada - Array de Real
	Número de Passos	Numérica - Real
	Lista de Passos	Lista/Textual - Array de Strings
	Número de Ingredientes	Numérica - Real
	Lista de Ingredientes	Lista/Textual - Array de Strings
Interações:	Quantidade e Esparsidade:	1.132.367 - 99,998%
	Características:	Nota atribuída e avaliação escrita
	Tipo:	Explícita até 5
Usuários:	226.570 - Selecionados aqueles com no mínimo 4 avaliações	
Dados de:	2018 à 2020	

Tabela 4.4: Informações sobre o conjunto de dados do Foodcom.

4.2.1 Visão Geral do LightFM

LightFM foi originalmente proposto para solucionar o problema de recomendação em uma plataforma de itens de moda. Kula (2015) menciona que neste cenário o catálogo de itens está em constante expansão e que, normalmente, os itens mais procurados são os mais recentes, agravando a necessidade de atenuar o problema de *cold start* de itens. Além disso, com o passar do tempo, o tamanho do catálogo chegou perto de 10 milhões de itens, fazendo com que o banco de dados seja altamente esparso. Por fim, uma grande porcentagem dos usuários da plataforma são novos, fazendo com que o *cold start* de usuários também tenha que ser resolvido.

Esta combinação dos dois tipos de *cold start* faz com que métodos normais não sejam capazes de solucionar o problema de forma satisfatória. Uma vez que, segundo o autor, métodos comuns de fatoração de matrizes (sistemas baseados em FC) têm dificuldade em decifrar os fatores latentes de itens e usuários quando interações são esparsas. Por outro lado, métodos de FBC lidam bem com essa situação, mas sofrem com a incapacidade de se beneficiar das interações e com a necessidade de grande volumes de informações para lidar com o *cold start* de usuários.

Para tratar o problema, LightFM foi proposto. Este por sua vez utiliza de um método híbrido, que combina a fatoração de matrizes de sistemas baseados em FC com a exploração de características de itens e usuários de um sistema de FBC.

Da mesma forma que métodos comuns de fatoração de matrizes, LightFM representa itens e usuários como um vetor de fatores latentes. Ele se diferencia no fato de que, ao contrário dos métodos normais que utilizam das interações para determinar os fatores latentes, LightFM utiliza uma abordagem semelhante a de sistemas de filtragem FBC, nessa abordagem os fatores

Goodreads		
Itens:	Descrição:	Livros do tipo quadrinhos
	Quantidade:	89.411
	Características	
	Nome	Tipo
	Título	Textual - String
	Número de Avaliações	Numérica - Real
	Média de Avaliações	Numérica - Real
	Autores	Lista não estruturada - Array de Strings
	(work/book)ID, ISBN(3), (kindle)ASIN	Categórica - String
	Livros Similares	Lista não estruturada - Array de Strings
	Dia, Mês e Ano de Publicação	Numérica - Real
	Informações de Edição	Textual - String
	Série	Categórica - String
	eBook	Categórica - Boolean
	Código da Língua	Categórica - String
	Código do País de Origem	Categórica - String
	Estantes Populares	Lista não estruturada - Array de Strings
	Publicadora	Categórica - String
	Título sem Série	Textual - String
	URL, URL Imagem de Capa, Link	Textual - String
	Número de Avaliações com Texto	Numérica - Real
	Descrição	Textual - String
	Formato	Categórica - String
	Número de Páginas	Numérica - Real
Interações:	Quantidade e Esparsidade:	542.338 - 99,99%
	Características:	Nota atribuída e avaliação escrita
	Tipo:	Explícita até 5
Usuários:	59.347	
Dados de:	Maio 2019	

Tabela 4.5: Informações sobre o conjunto de dados do Goodreads.

são determinados como uma combinação linear das representações latentes (*embeddings*) das características.

Kula (2015) cita que os dois principais objetivos do LightFM são: aprender representações a partir das interações (como um sistema baseado em FC) e ser capaz de realizar recomendações de novos itens e recomendações para novos usuários.

O primeiro é concretizado a partir das representações latentes, que permite que itens bem avaliados pelos mesmos usuários tenham a representação latente de suas características próximas, com o oposto também sendo verdade, com itens que não são bem avaliados pelos mesmos usuários tendo a representação latente de suas características distantes. Desta forma, o sistema é capaz de recomendar itens com características diferentes dos que o usuário já interage, mas semelhantes. Diferente de sistemas FBC, que relacionam itens com base na ocorrência simultânea de características (comparando palavras presentes, por exemplo), não sendo capazes de recomendar itens que não possuem características semelhantes, mesmo que sejam apreciados pelo mesmo tipo de usuário. Já o segundo objetivo é atingido a partir das representações latentes de itens e usuários, que por sua vez são a combinação linear das representações das características.

Isto é efetivo já que mesmo em situações de novos itens e usuários, é possível conhecer suas características com antecedência.

O modo de criar representações se assemelha a técnicas recentes de *word embeddings*, mas difere no fato de que ao invés de realizar uma análise estatística textual, LightFM cria as representações com base nas interações.

Segue as expressões matemáticas para as representações latentes dos usuários e itens respectivamente, para a recomendação e função de otimização, onde:

- e: Representação latente de uma característica;
- j: Uma característica;
- i: Um item;
- u: Um usuário.

$$q_u = \sum_{j \in f_u} e_j^U \tag{4.1}$$

$$p_i = \sum_{j \in f_i} e_j^I \tag{4.2}$$

A recomendação de i para u é computada a partir de:

$$r_{ui} = f(q_u \cdot p_i + b_u + b_i) \tag{4.3}$$

Onde bu e bi são os termos enviesados (*bias term*) do item i e usuário u:

$$b_u = \sum_{j \in f_u} b_j^U \tag{4.4}$$

$$b_i = \sum_{j \in f_i} b_j^I \tag{4.5}$$

A função f é uma função sigmoide e o objetivo de otimização consiste em maximizar a probabilidade (Equação 4.6). As representações das características são aprendidas utilizando descida de gradiente e ADAGRAD (Duchi et al., 2011) para determinar a taxa de aprendizado.

$$L(e^U, e^I, b^U, b^I) = \prod_{(u,i) \in S^+} r_{ui} \times \prod_{(u,i) \in S^-} (1 - r_{ui}) \tag{4.6}$$

4.2.2 Características da Biblioteca LightFM

A seguir estão listadas as principais características do LightFM que influenciaram na sua escolha como algoritmo de recomendação para realização do estudo. As métricas e parâmetros serão abordados em maiores detalhes nos experimentos.

- Documentação clara e concisa, com exemplos de usos;
- Possibilidade de criar conjuntos de dados personalizados, com suporte a inserção de características de usuários e itens;

- Configuração por diversos parâmetros como: função de perda, taxa de aprendizado, número de componentes das representações no espaço latente, diferentes pesos para cada característica, número de *threads*, número de épocas ao realizar treinamento, entre outros;
- Suporte para diferentes métricas de avaliação;
- Artigo com foco em atenuar o problema de *cold start*;
- Utilização de algoritmo semelhante a *word embeddings*.

5 EXPERIMENTOS E RESULTADOS

Para realização dos experimentos, Schifferer et al. (2020) foi utilizado como referência. O artigo em questão é uma apresentação dos temas discutidos em um vídeo tutorial de mesmo nome, que explora técnicas de EC aplicadas a SR. A apresentação discute diferentes tipos de dados e métodos que podem ser aplicados para cada um, a fim de melhorá-los e, consequentemente, obter melhores resultados em uma recomendação, além de abordar problemas comuns ao se trabalhar com análise de dados e discutir possíveis soluções para estes problemas.

Neste capítulo experimentos e análises dos conjuntos de dados serão feitos, explorando cada característica e possibilidades para extrair o pleno potencial de cada uma delas. As seções estão organizadas da seguinte forma: A primeira delas, Seção 5.1, detalha o processo de experimentação realizado, já as outras (Seções de 5.2 à 5.6) detalham diferentes técnicas de EC utilizadas para analisar, transformar, preencher, combinar e filtrar características. Destas, a Seção 5.2 apresenta os conjuntos de dados utilizados, com figuras e informações que servirão de base para a fundamentação de muitas das análises realizadas. Seção 5.3 explora remoções de características que podem ser feitas nos conjuntos de dados sem grandes perdas, filtragens realizadas de forma manual. Seção 5.4 discute o problema de dados faltantes e o que pode ou não ser feito para atenua-lo (como utilização de preenchimento e remoção). Seção 5.5 aborda processos realizados a fim de mudar a representação dos dados, como *binning*, com o objetivo de extrair informações mais relevantes. Seção 5.6 aborda estratégias utilizadas para filtrar características utilizando análise de correlação. Seção 5.7 por sua vez explora os parâmetros padrões e alternativos utilizados no LightFM. Estes parâmetros alternativos têm como objetivo descobrir se é possível obter um melhor desempenho a custo de um maior tempo de execução, extraindo potencial principalmente de características textuais, ou então se é possível manter o desempenho com um menor tempo de execução. Por fim, a Seção 5.8 apresenta e discute os resultados obtidos.

5.1 EXPERIMENTAÇÃO

Neste capítulo serão discutidos diversos procedimentos de EC realizados, e para testar o desempenho destas modificações experimentos serão conduzidos utilizando a ferramenta LightFM. Desta forma, é de vital importância discorrer sobre como estes experimentos foram realizados e as condições sobre as quais foram executados.

Os conjuntos de dados foram obtidos (interações e características de itens) e utilizando Kula (2015) como referência, 80% das interações foram selecionadas para treinamento e 20% para testes (procedimento feito de forma aleatória com uma semente fixada, que por limitações de tempo não pode ser realizado com um *K-Fold* ou com a média de diferentes sementes). Após isso, 20% de todos os itens foram selecionados (também de forma aleatória com uma semente pré-definida), estes tiveram suas interações removidas do conjunto de treinamento e adicionadas ao conjunto de teste. De forma mais sucinta, estamos simulando um *cold start* de itens ou melhor, agravando situações de *cold start* de itens (como veremos na próxima seção). Após este procedimento inicial, as características e interações dos itens são preparados para serem adicionadas ao modelo do LightFM. As interações de teste são divididas em 3 conjuntos, interações de itens frios – itens nunca vistos no treinamento e/ou com menos de n interações no treinamento, interações de itens quentes – com mais de n interações no treinamento – e todas as interações. Nos resultados que serão apresentados, considerou-se n igual a 0 (neste limiar não há

a possibilidade de menos que *n* interações), mas nos Apêndices A e B estão disponíveis testes realizados com *n* igual a 10 e 20.

O sistema foi treinado com todos os itens e interações de treinamento, para testes foi realizada a predição para cada um dos 3 conjuntos de teste, e para cada um destes conjuntos as predições resultantes foram avaliadas utilizando as métricas *auc_score*, *precision@k*, *recall@k* e *reciprocal_rank* (*k* é igual a 10). As notas atribuídas e os tempos de execução foram registrados para cada experimento realizado, e serão mostrados ao final do capítulo.

Os testes foram realizados em máquina pessoal, mas os resultados finais foram obtidos no *cluster* C3HPC (*C3SL High Performance Computer*), executados com 32 *threads* e 16 GB de RAM. Este *cluster* utiliza um sistema de fila de *jobs* para evitar concorrência entre tarefas.

5.2 ANÁLISE DOS CONJUNTOS DE DADOS

Para a realização de uma EC, primeiramente serão apresentadas as distribuições das características principais de cada conjunto de dados, juntamente com uma breve análise. As informações apresentadas aqui servirão de embasamento para muitas das outras análises feitas e conclusões atingidas neste capítulo.

5.2.1 MovieLens + IMDb

Inicialmente, foram feitas análises de dados faltantes, dados distintos e esparsidade para obter uma visão inicial sobre o conjunto de dados. E como podemos observar na Figura 5.1, as características são relativamente bem comportadas, com as mais distintas estando presentes, e como o nome delas e a descrição preliminar sugere (Tabela 4.3), são relativamente simples, sendo este o motivo pela escolha deste conjunto, dados que permitam explorar o cenário de *cold start* de itens em condições relativamente ideais de características de itens.

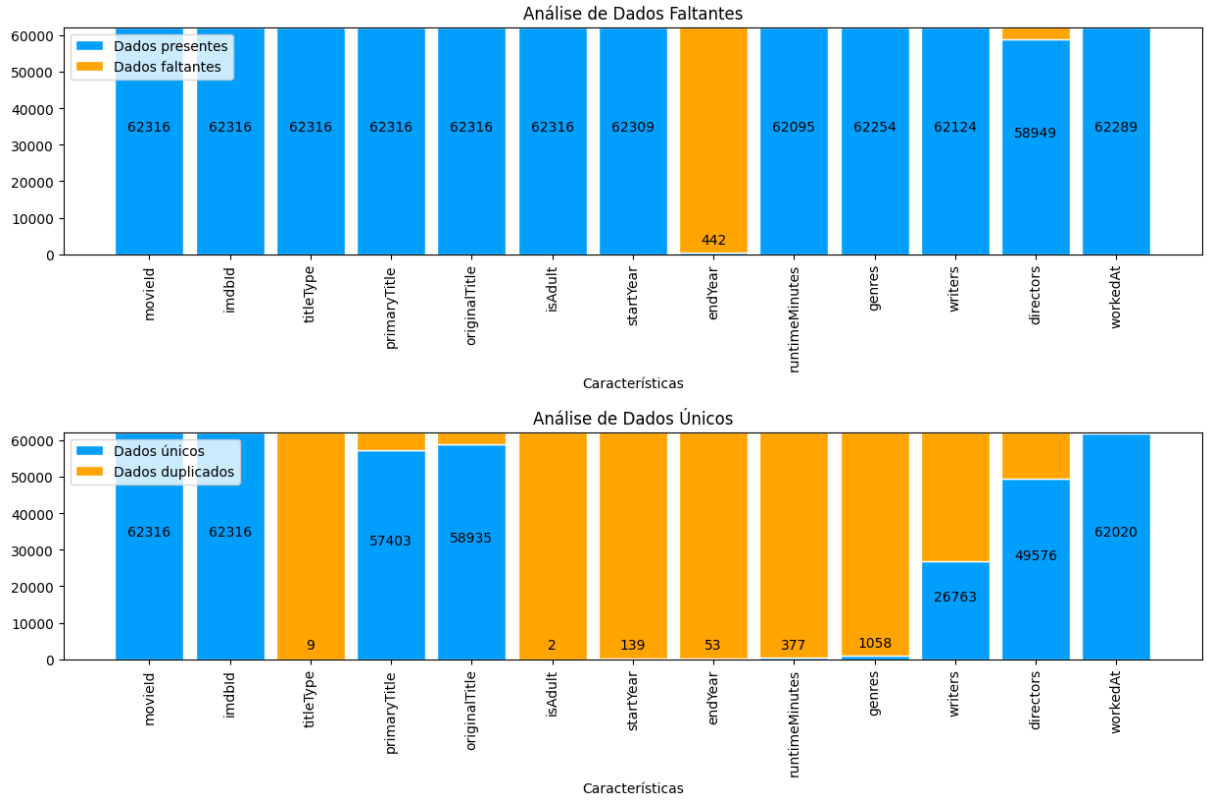


Figura 5.1: Informações de dados faltantes e distribuição dos dados MovieLens + IMDb.

Em relação às interações, na Figura 5.2 podemos observar a distribuição delas em termos de quantidade e valor. Há diversos itens com muitas iterações, mas a grande maioria possui uma baixa quantidade de interações. As avaliações por sua vez, estão distribuídas fortemente entre 3 e 5.

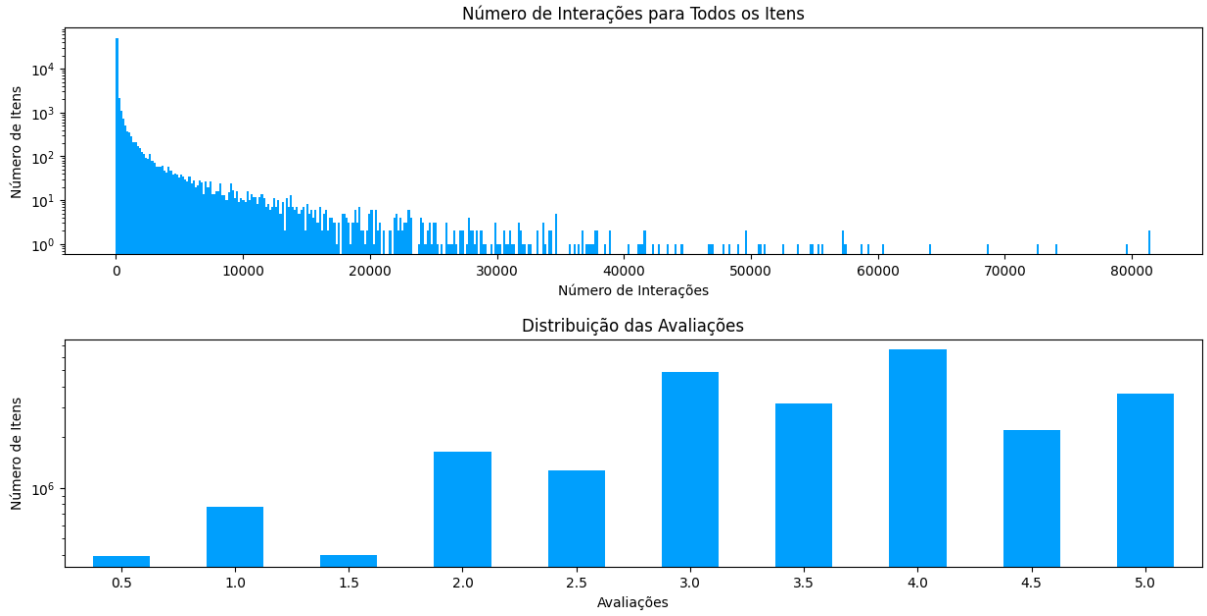


Figura 5.2: Distribuição do número de interações e das avaliações dos dados MovieLens + IMDb.

Nas Figuras 5.3 e 5.4 são apresentadas a interseção entre as características título original e título principal, e a lista de diretores/escritores e equipe de produção. Tais figuras nos ajudam

a entender a diferença entre estas características e serão importantes para, respectivamente, uma filtragem inicial de características e criação de uma nova característica, que representa a combinação.

Figura 5.3: Interseção entre as características título original e título principal.

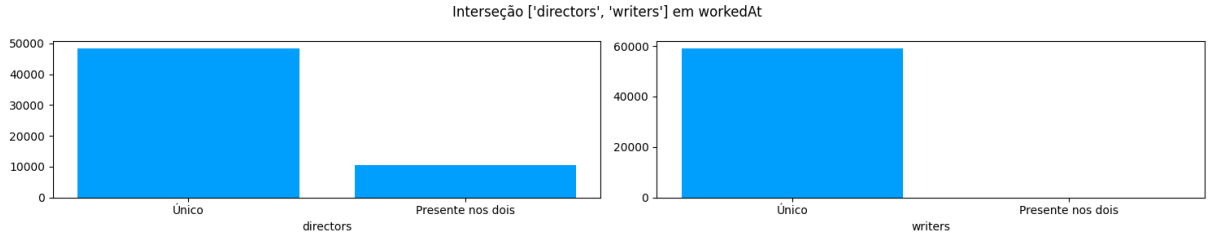


Figura 5.4: Interseção entre as características diretores e equipe de produção/escritores e equipe de produção.

Por fim, as Figuras 5.5, 5.6, 5.7 e 5.8 mostram a distribuição de algumas das características mais importantes para as análises que estão por vir.

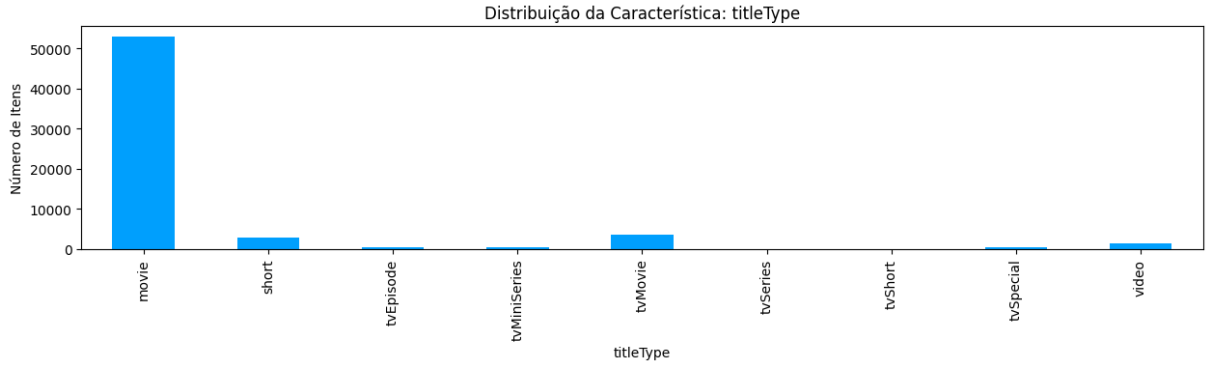


Figura 5.5: Distribuição da característica tipo de obra.

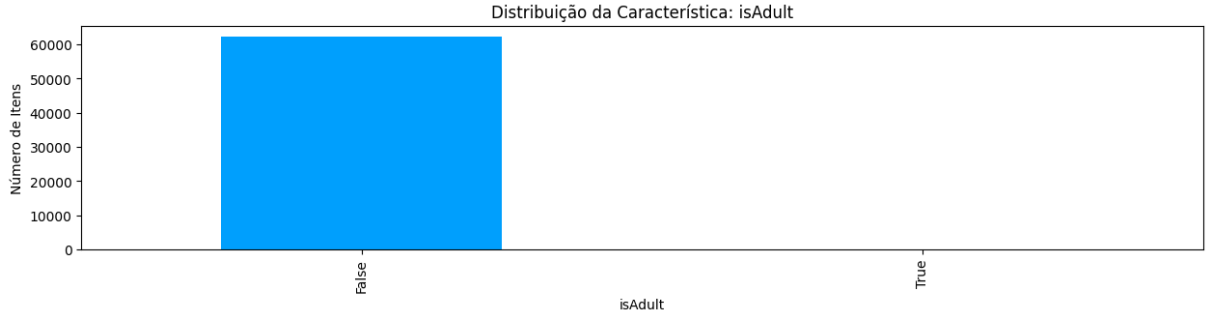


Figura 5.6: Distribuição da característica destinado a adultos.

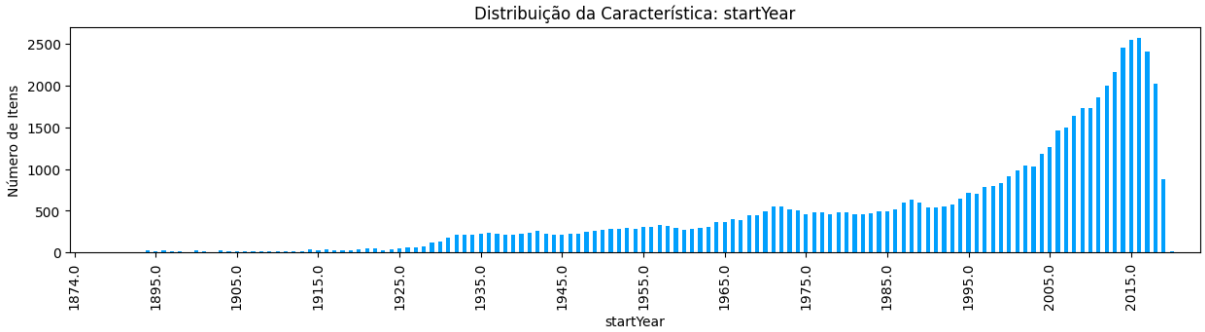


Figura 5.7: Distribuição da característica ano de lançamento.

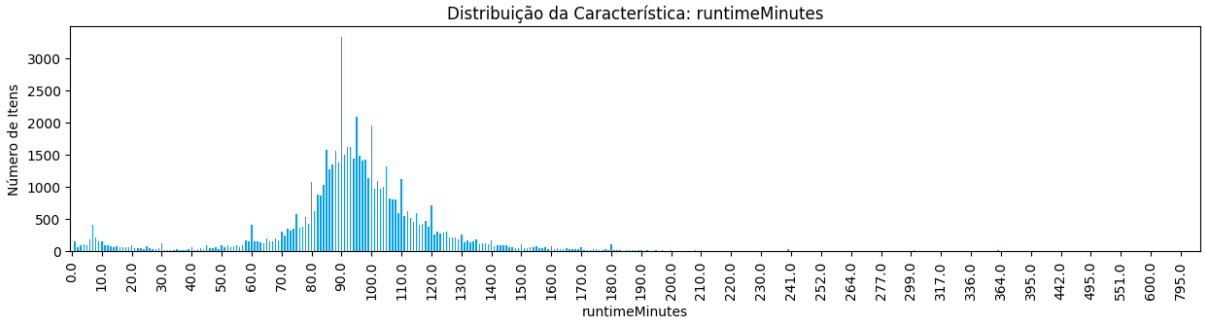


Figura 5.8: Distribuição da característica duração.

5.2.2 Foodcom

Da mesma forma que anteriormente, começamos com uma análise de dados faltantes, dados distintos e esparsidade. E como no MovieLens + IMDb, os dados são relativamente bem comportados (Figura 5.9). Entretanto, algumas das características como "Detalhes"e "Lista de Passos"apresentam uma nova oportunidade para a análise, a exploração de características textuais de tamanho considerável, diferente das outras características de texto que possuem tamanho relativamente pequeno. Para estas informações diferentes abordagens serão utilizadas ao serem analisadas e acredita-se que a configuração do algoritmo de recomendação será essencial para seus desempenhos.

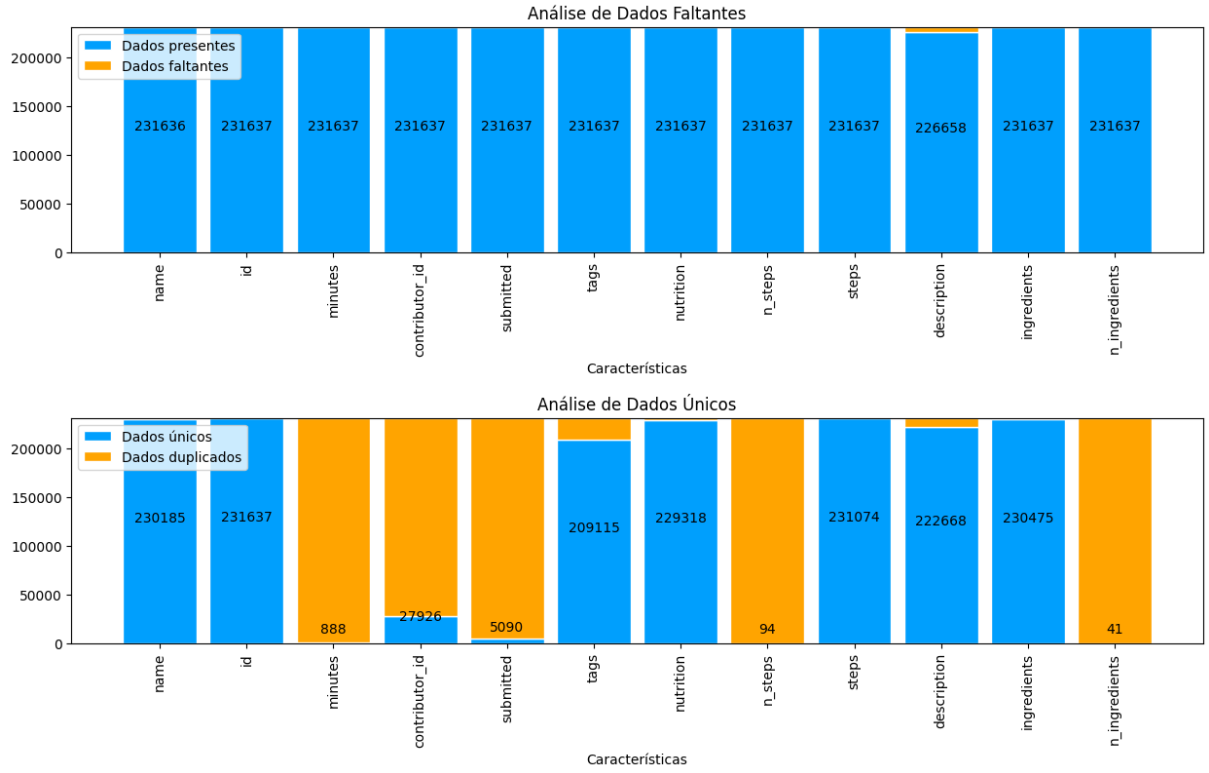


Figura 5.9: Informações de dados faltantes e distribuição dos dados Foodcom.

Novamente, na Figura 5.10 podemos observar a distribuição das interações, desta vez em uma escala menor de número de interações, ainda mais propensa a problemas de *cold start* de itens. As avaliações continuam com um destaque na faixa de 3 a 5, com exceção para avaliações 0 (que indica interações de nota 0 e não a ausência de interação), presentes em número significativo nestes dados.

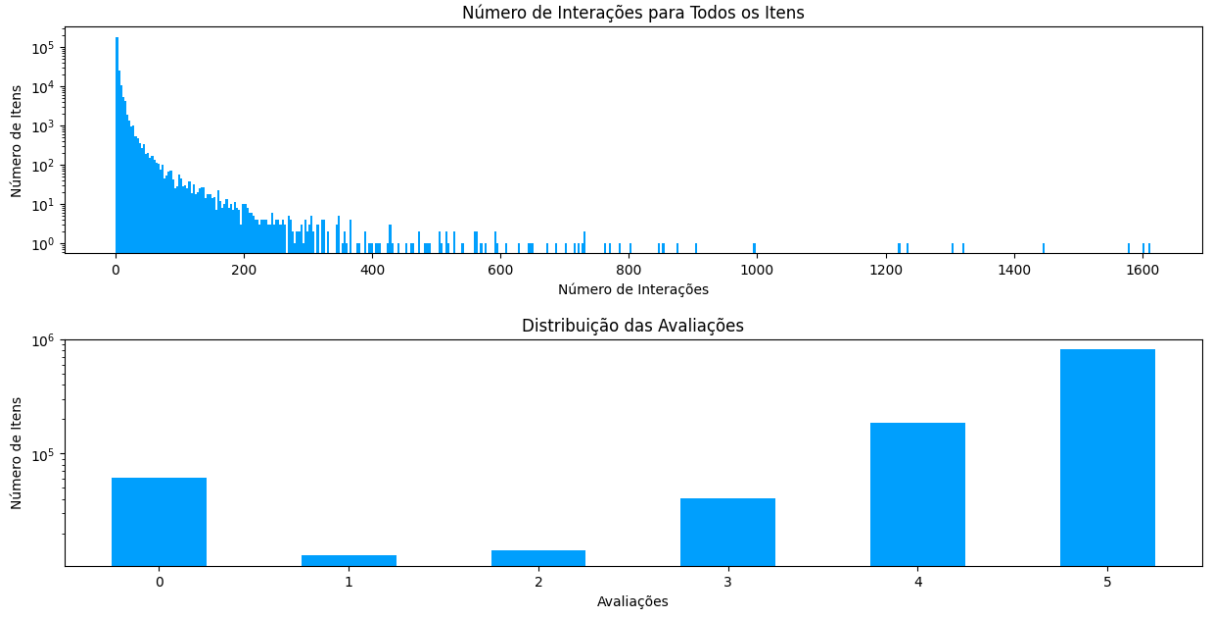


Figura 5.10: Distribuição do número de interações e das avaliações dos dados Foodcom.

Diferente do conjunto de dados do MovieLens + IMDb, não foram notadas interseções entre nenhuma das características dos dados do Foodcom. Porém, vale apresentar a distribuição de algumas das características mais relevantes (Figuras 5.11, 5.12 e 5.13).

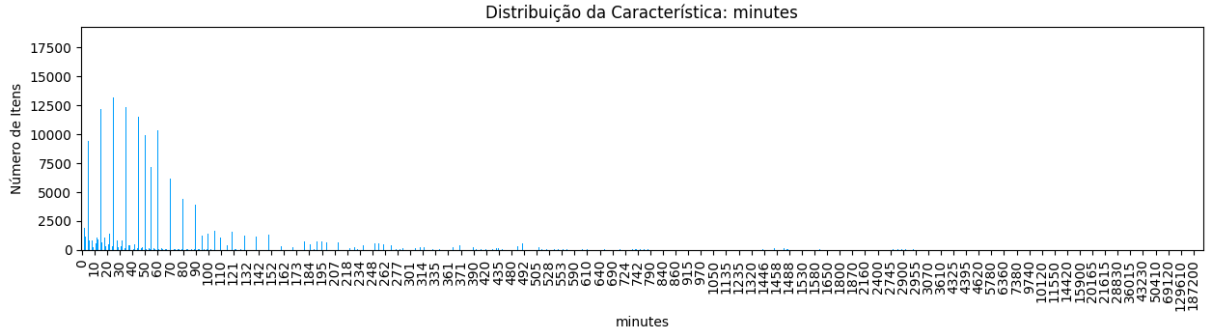


Figura 5.11: Distribuição da característica tempo de preparo.

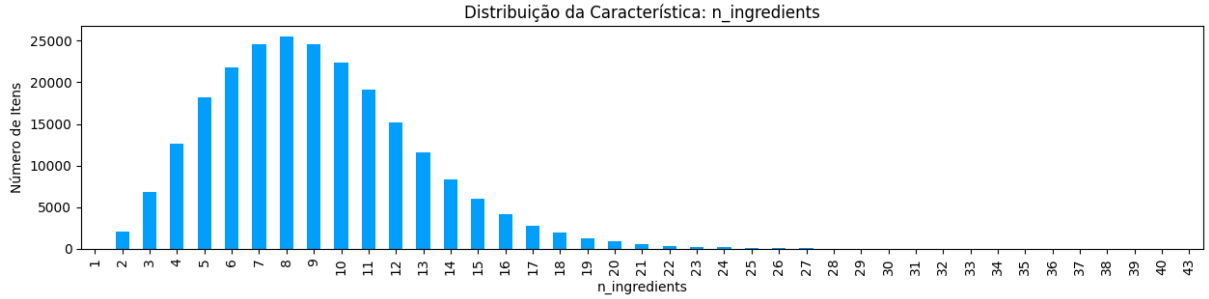


Figura 5.12: Distribuição da característica número de ingredientes.

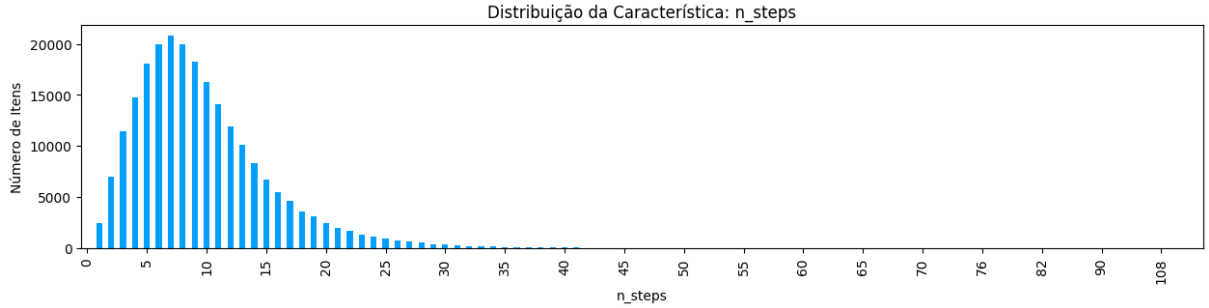


Figura 5.13: Distribuição da característica número de passos.

5.2.3 Goodreads

Como nas duas últimas vezes, a Figura 5.14 apresenta informações de dados faltantes, distintos e esparsidade. Porém, desta vez percebemos que diversos valores estão faltando para diversas características. Além disso, a quantidade de dados distintos também reduziu consideravelmente. Sendo assim, há a oportunidade de observar de forma mais efetiva a correlação entre itens.

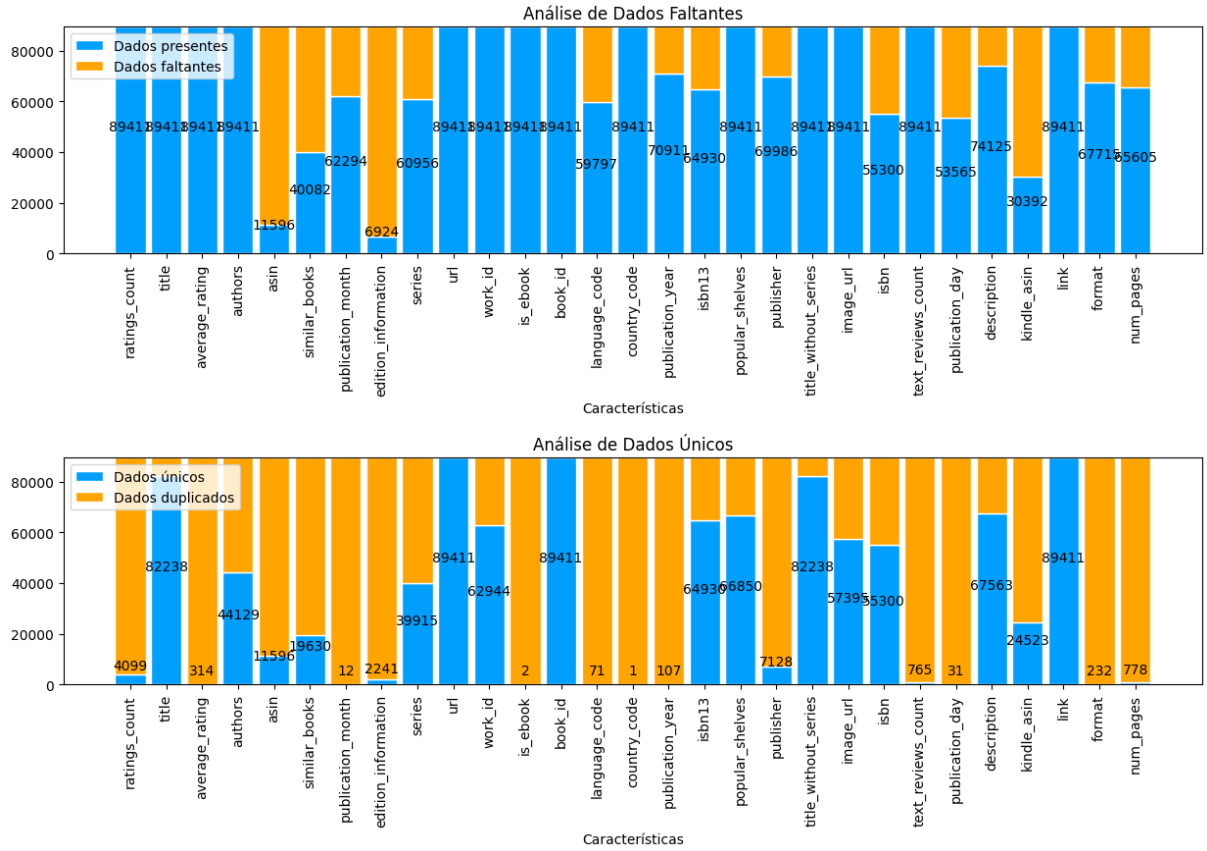


Figura 5.14: Informações de dados faltantes e distribuição dos dados Goodreads.

A Figura 5.15 mais uma vez explora a distribuição das interações, com um comportamento semelhante aos outros conjuntos de dados.

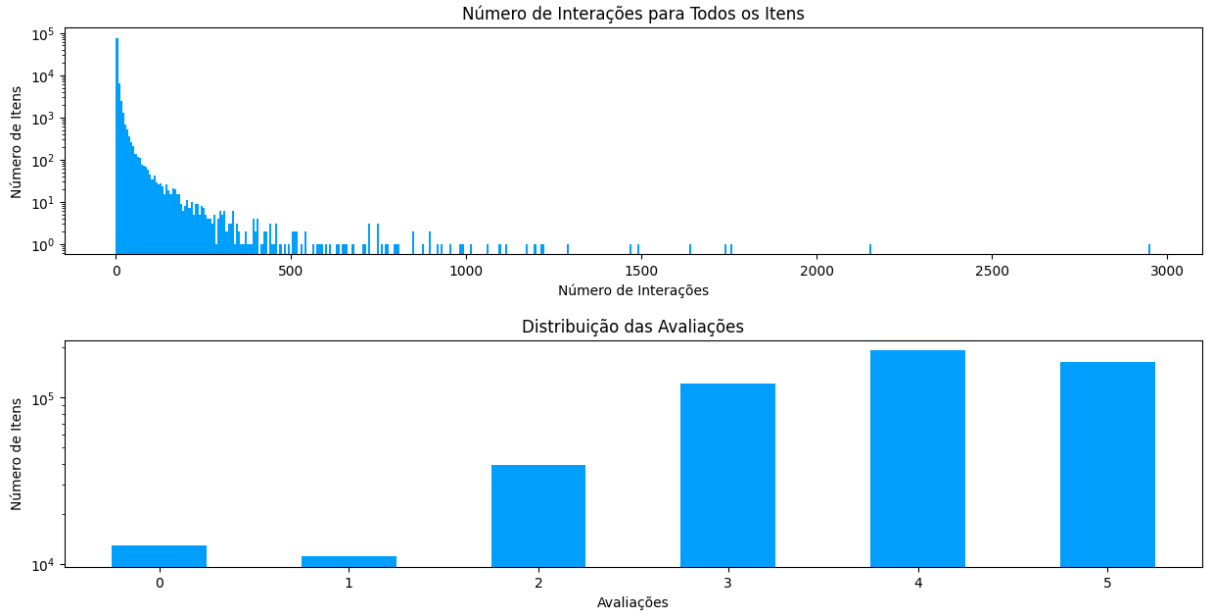


Figura 5.15: Distribuição do número de interações e das avaliações dos dados Goodreads.

Finalizando, as Figuras 5.17, 5.18, 5.19, 5.20, 5.21 e 5.22 apresentam a distribuição de características importantes para as análises que estão por vir. Figura 5.16 exibe a interseção entre as características título e título sem série.

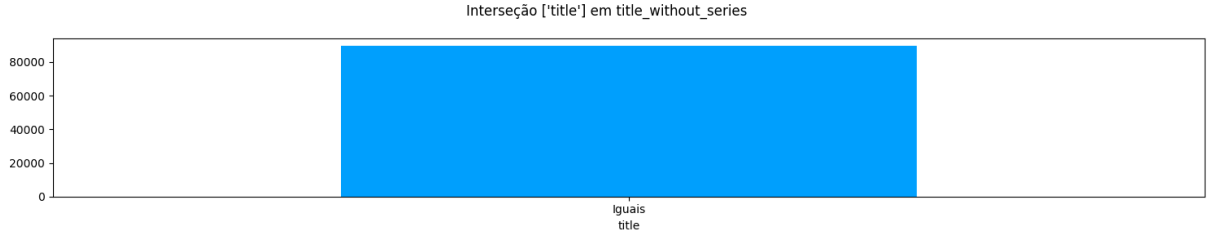


Figura 5.16: Interseção das características título e título sem série.

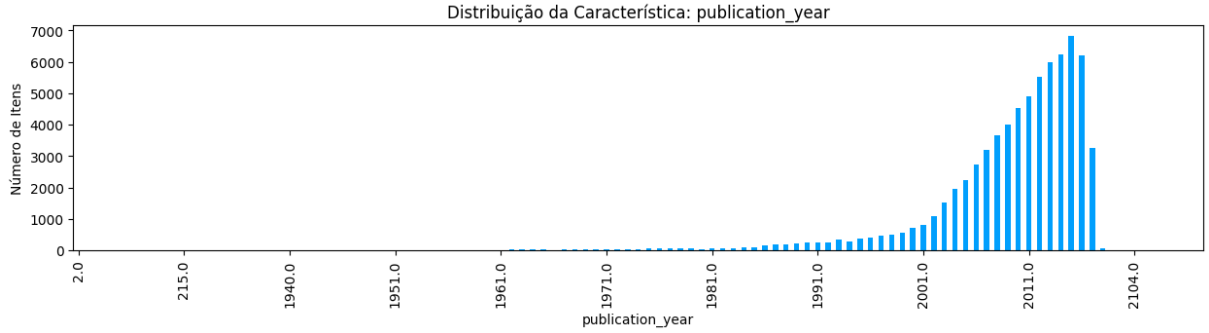


Figura 5.17: Distribuição da característica ano de publicação.

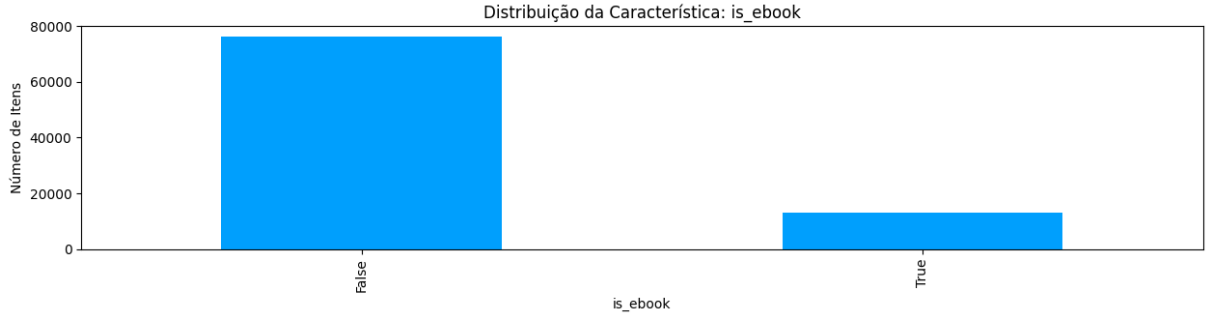


Figura 5.18: Distribuição da característica é ou não eBook.

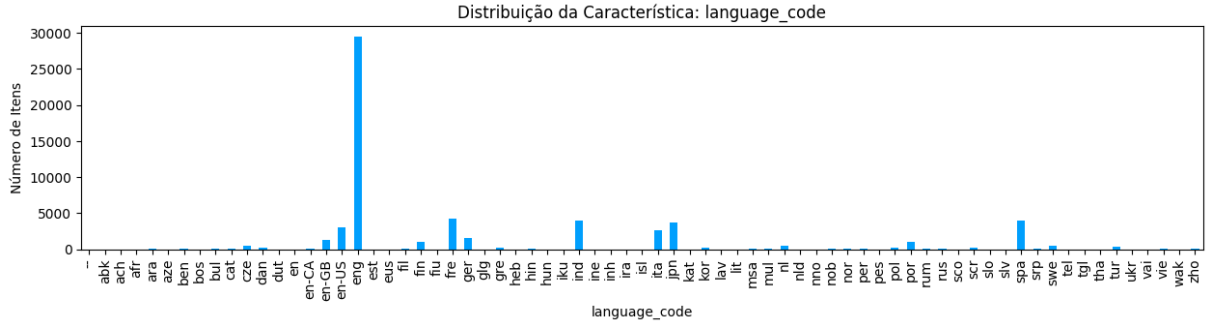


Figura 5.19: Distribuição da característica código da língua.

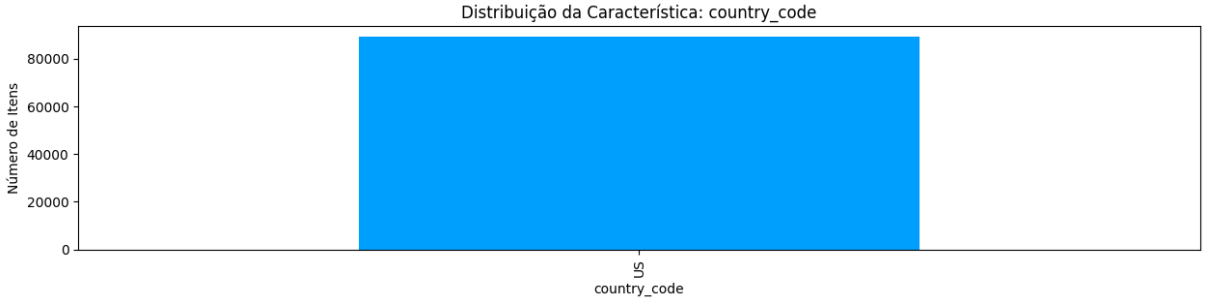


Figura 5.20: Distribuição da característica código do país de origem.

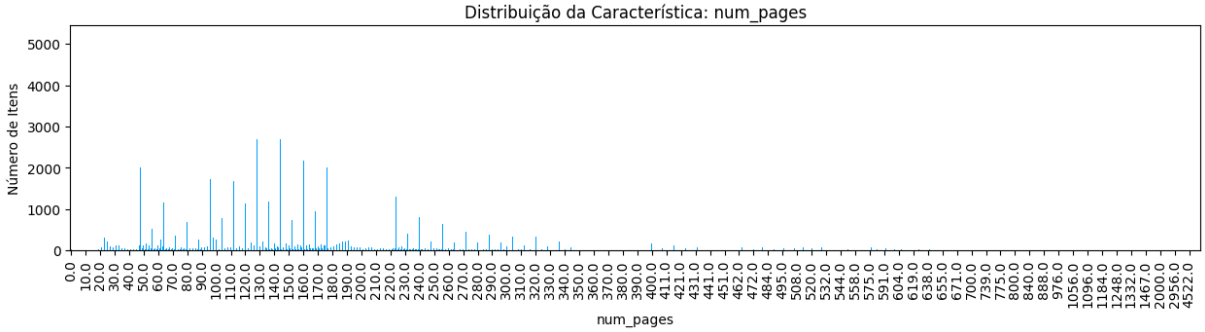


Figura 5.21: Distribuição da característica número de páginas.

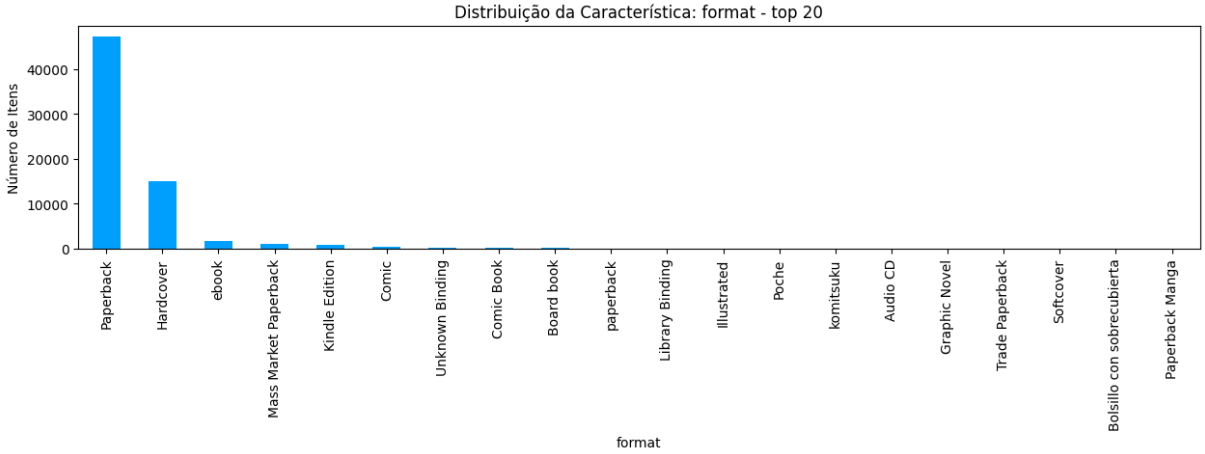


Figura 5.22: Distribuição da característica formato.

5.3 SELEÇÃO MANUAL

Como podemos notar na seção anterior, diversas características possuem altíssima variância, com um valor diferente para cada item, algumas são o oposto disso, tendendo a um único valor. Outras características possuem interseção entre elas, com valores semelhantes frequentemente. Estas características não ajudam o sistema a relacionar itens, uma vez que a representação latente destas características será totalmente diferente ou totalmente igual. Idealmente gostaríamos de manter apenas aquelas que estejam entre os dois opostos, diferentes o suficiente para não dizer que todos os itens são iguais e parecidas o suficiente para permitir o estabelecimento de relações entre itens. Sendo assim, iniciamos a EC com a remoção destas características.

5.3.1 MovieLens + IMDb

Em relação a este conjunto de dados, 3 características podem ser removidas: IMDbId, título original e destinado a adultos. A primeira delas é única para cada item (Figura 5.1). Título original por outro lado possui altíssima interseção com título principal (Figura 5.3), logo não acrescenta novas informações, que possam ajudar o sistema de recomendação. A característica destinado a adultos por outro lado possui apenas um valor para todos os itens (Figura 5.6), desta forma, podemos dizer que ela não possui nenhum impacto no estabelecimento de relações entre itens.

5.3.2 Foodcom

Para o Foodcom duas características podem ser removidas facilmente, estas sendo data de publicação e usuário que publicou. A primeira não possui relevância para o LightFM, uma vez que todas as interações possuem o mesmo peso, não importando quando foram realizadas, logo, a data de publicação de um item também não é relevante. A segunda por sua vez possui alta variabilidade, tanto que dificultou a sua visualização, com um grande número de usuários com poucas publicações e alguns raros *outliers* tendo publicado um número significativo de itens.

5.3.3 Goodreads

Por fim, ao trabalhar com o Goodreads, há um grande número de características que podem ser removidas sem grandes perdas. Inicialmente podemos remover número de avaliações, média de avaliações e número de avaliações com texto, já que temos como objetivo melhorar os resultados de uma recomendação híbrida de sistemas de FC e FBC. Por outro lado, essas 3 características poderiam ser úteis caso estivéssemos trabalhando com uma recomendação baseada em popularidade, estabelecendo diferentes pesos para cada item, baseado em sua média de avaliações, por exemplo. Outras características que podem ser removidas sem nenhuma dúvida são título sem série e código do país de origem, uma vez que a primeira é igual a característica título (Figura 5.16) e a segunda possui um único valor (Figura 5.20).

Em seguida, removemos também as características: códigos ISBN, ISBN3 e ASIN, URL, URL da imagem de capa e link, por serem únicas para cada item (Figura 5.14) e por não estarmos estudando possibilidades como extração de novas características utilizando as referências para o conteúdo original, realizando análise das imagens de capa, por exemplo. Também foi estudada a possibilidade de utilizar as características de ano, mês e dia de publicação, combinando as 3 em uma única característica, porém, uma análise de distribuição notou que o dado teria alta variabilidade, sem nenhum padrão, logo optou-se por mantermos apenas o ano de publicação. Algo que faz sentido se pensarmos que provavelmente um ser humano, no máximo, se importaria com o ano de lançamento de uma obra deste tipo, mas que pode não ser verdade em outros cenários, como observado na discussão sobre o LightFM.

Por fim, se questionou a possibilidade de remover as características se é ou não um eBook e código da língua utilizada (Figuras 5.18 e 5.19) devido a falta de variabilidade. Entretanto, julgou-se que essas características são altamente importantes para o consumidor das obras, já que dizem respeito a forma como o item é consumido e por quem está limitado a ser consumido, logo não foram removidas.

5.4 DADOS FALTANTES

Ao trabalhar com qualquer tipo de dado, um dos problemas mais comuns é a falta dele. Para lidar com isso podemos preencher utilizando diferentes técnicas, como usar a média ou mediana, ou então utilizar outras informações para supor um valor. Uma solução mais extrema seria não utilizar o dado por completo. Na situação aqui discutida, SR, a última opção seria equivalente a remover um item, o que não é possível, já que teremos que recomendar ele de qualquer forma, sendo assim estaríamos reduzindo ainda mais a qualidade das recomendações, ainda mais que optar por utilizar o dado do jeito que está. Como observamos nas seções anteriores, cada um dos conjuntos de dados possui um certo nível de dados faltantes, esta sessão por sua vez irá discutir possibilidades para solucionar cada um destes problemas. Nos casos onde acreditamos que uma solução satisfatória não seja adequada, lembramos que o LightFM realiza a combinação linear das representações latentes das características, logo, a falta de uma característica não deve impactar drasticamente os resultados observados.

5.4.1 MovieLens + IMDb

No MovieLens + IMDb as seguintes características estão faltando em certos itens: ano de início, ano de fim, duração, gêneros, escritores, diretores e equipe de produção (Figura 5.1). Na maioria destes casos o número não é suficientemente grande para ser significativo para os mais de 62 mil itens. Os casos que são significativos são: ano de fim e diretores. Para o primeiro, podemos preencher os dados faltantes com o ano de início, já que a grande maioria dos itens são filmes (Figura 5.5). Porém, optamos por remover esta característica, já que aproximadamente 99% dos itens teriam valor igual a uma outra característica, ano de início. Em relação aos diretores, a situação é menos grave e como veremos na seção 5.5, iremos combinar esta característica e a de escritores com a de equipe de produção, logo, este não será um problema.

5.4.2 Foodcom

Para o Foodcom, apenas uma característica possui dados faltantes, sendo esta a descrição das receitas. Neste caso julgou-se que não há uma solução viável, além de não utilizar a característica, o que julgou-se ser desnecessário, já que são poucos itens, relativamente.

5.4.3 Goodreads

Em relação ao Goodreads temos algumas situações de dados faltantes além daquelas em que as características foram removidas na seção anterior. Para algumas, que não são textuais, podemos pensar em uma solução. Outras como livros similares, série, informações de edição, código de língua e descrição, são características textuais, logo, não há como preenchê-las, pelo menos não de uma forma que esteja dentro do escopo deste trabalho. Também acredita-se que não seja correto removê-las, já que são muitas, poderíamos perder muita informação útil. Por um lado, a falta delas poderia dificultar o sistema de recomendação a estabelecer relações entre certos itens, pelo fato do LightFM ser baseado na média das representações latentes, até onde entendemos. Por outro, o LightFM também diz construir as representações latentes levando em consideração as características serem apreciadas pelo mesmo grupo de usuários, logo, isto poderia equilibrar o problema. Neste sentido, não sabemos ao certo qual seria a melhor escolha, remover as características ou utilizá-las mesmo não estando presentes em diversos itens, neste trabalho optou-se por mantê-las.

As características sobre as quais podemos fazer algo são: ano de publicação e número de páginas. Em ambos os casos podemos utilizar a média ou a mediana para preencher os dados faltantes, entretanto, uma outra dúvida surge. Qual seria o impacto de preencher, respectivamente, 20 e 25 mil itens? Um preenchimento de tamanha escala, comparado com os aproximadamente 90 mil itens totais, poderia prejudicar as relações entre os itens, tornando estas características inúteis como um todo e talvez prejudiciais. Para responder essas dúvidas precisaríamos de testes voltados a essas dúvidas, o que novamente, foge do escopo deste trabalho, sendo assim optou-se por não alterá-las.

5.5 COMBINAÇÃO E TRANSFORMAÇÃO

Ao explorarmos certos dados, percebemos uma distribuição complexa, difícil de identificar um padrão, ou então que possui interseção com outro dado, logo redundante. Tais situações podem dificultar um SR a identificar relações entre itens. Sendo assim, uma possível solução é reduzir a gama de valores. Para realizar isso, podemos combinar informações, como categorias de um dado categórico ou então realizar um agrupamento de dados (*binning*) em uma variável continua, transformando-a em uma variável categórica e buscando facilitar o processo de treinamento, podendo até mesmo reduzir os custos associados a um SR. Ao realizar *binning*, a estratégia utilizada foi manter um número aproximadamente igual de exemplos para cada categoria, com faixas de tamanho variável. Também é possível manter faixas de mesmo tamanho com diferentes números de exemplos, porém, em testes iniciais, esta alternativa pareceu acentuar outro problema, o de que a maioria dos valores se concentra em uma única categoria, logo não foi utilizada. Para a realização deste tipo de agrupamento, foram criadas 10 faixas.

5.5.1 MovieLens + IMDb

Com relação ao conjunto de dados MovieLens + IMDb, observou-se diferentes situações. Primeiramente, notou-se um grau de interseção entre os dados de diretores e equipe de produção (Figura 5.4). Apesar de não terem sido identificadas interseções entre escritores e equipe de produção, percebeu-se que muitos dados de equipe de produção continham valores "escritor:id", logo, estes dados estão presentes na equipe de produção mas não na lista de escritores. A fim de reduzir o número de características e remover a falta de coerência, diretores e escritores foram combinados na equipe de produção.

A segunda situação foi a distribuição da característica tipo de obra (Figura 5.5), esta assume predominantemente o valor de "filme", enquanto que os outros valores estão dispersos entre diversas categorias, muitas delas estando relacionadas a obras destinadas a televisão. Desta forma, foi realizado um agrupamento entre elas (Figura 5.23), com a ideia de que o número de observações de cada uma é tão reduzido, que pode impactar negativamente os itens destas categorias. Sendo assim, agrupar com outras categorias semelhantes pode contribuir positivamente para estes itens.

Por fim, foi realizado um *binning* das características de ano de início (lançamento) e duração (Figuras 5.24 e 5.25). No caso da primeira, o dado foi agrupado por décadas. Esta ideia surgiu de aplicativos de *streaming* de música, que partem do pressuposto de que um usuário não necessariamente estará interessado em itens de um ano específico, mas de itens de uma década em específica.

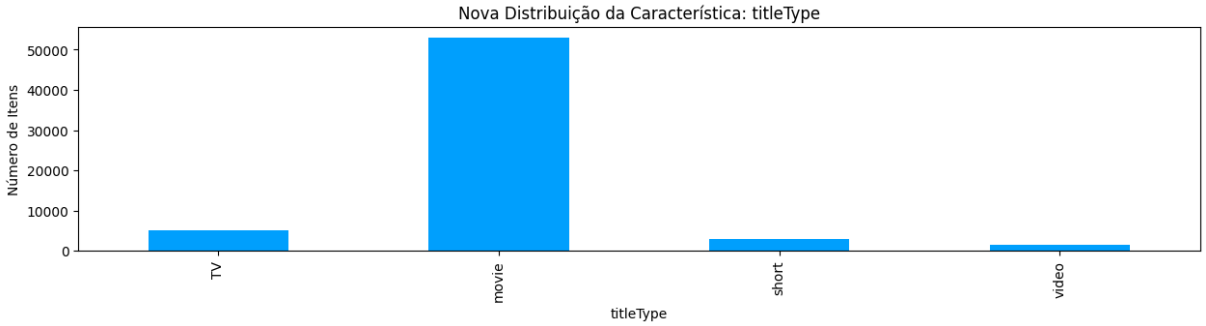


Figura 5.23: Nova distribuição da característica tipo de obra.

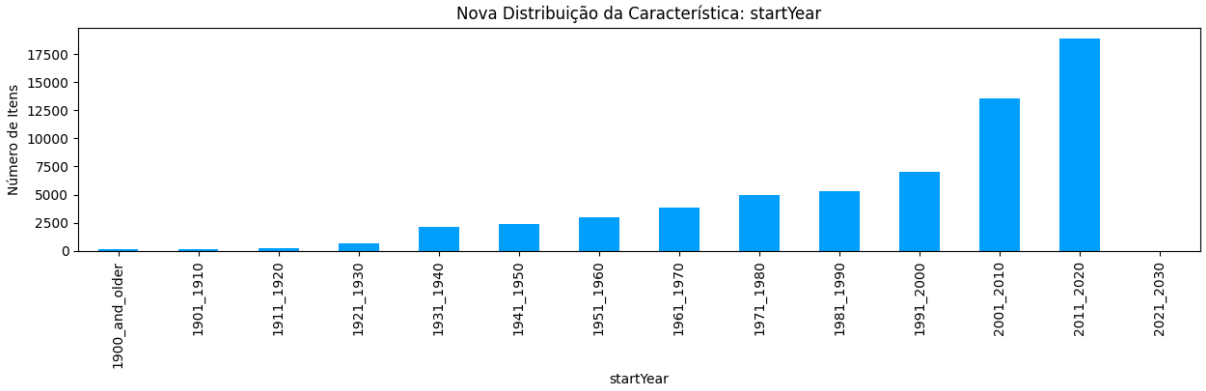


Figura 5.24: Nova distribuição da característica ano de lançamento.

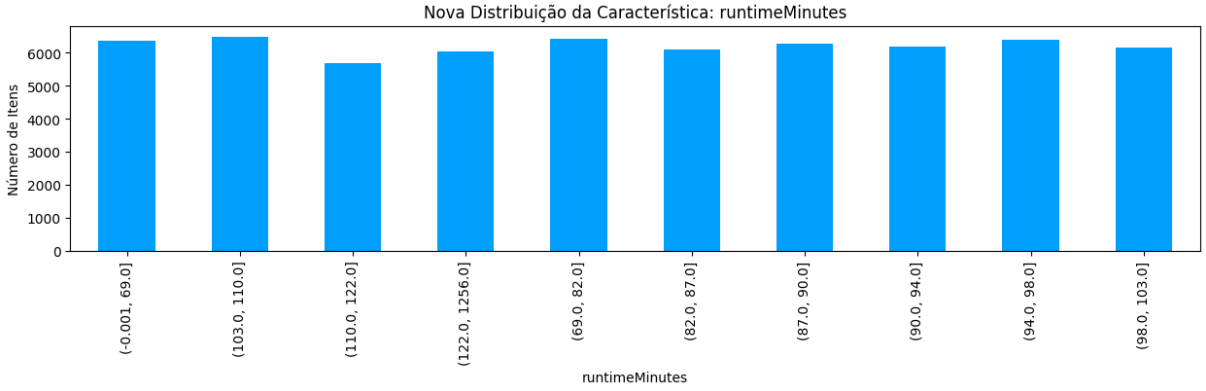


Figura 5.25: Nova distribuição da característica duração.

5.5.2 Foodcom

Da mesma forma que na subseção anterior, foi realizado um *binning* das características tempo de preparo, número de ingredientes e número de passos (Figuras 5.26, 5.27 e 5.28) no Foodcom. Também cogitou-se a possibilidade de combinar as informações de nutrição de uma maneira mais palpável, relacionando os 7 valores nutricionais a "classificações", visando medir o quão saudável uma receita é. Entretanto, por falta de conhecimento nessa área, a ideia foi descartada.

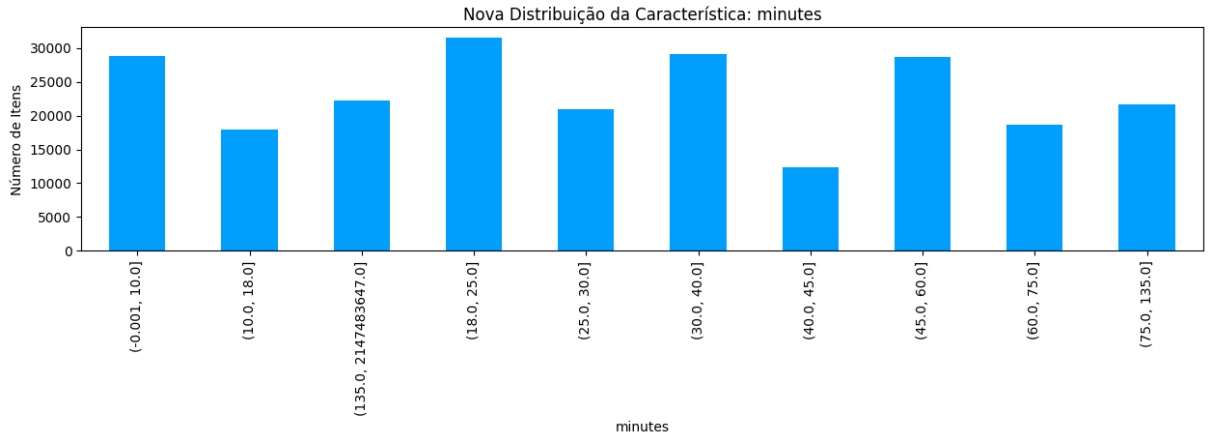


Figura 5.26: Nova distribuição da característica tempo de preparo.

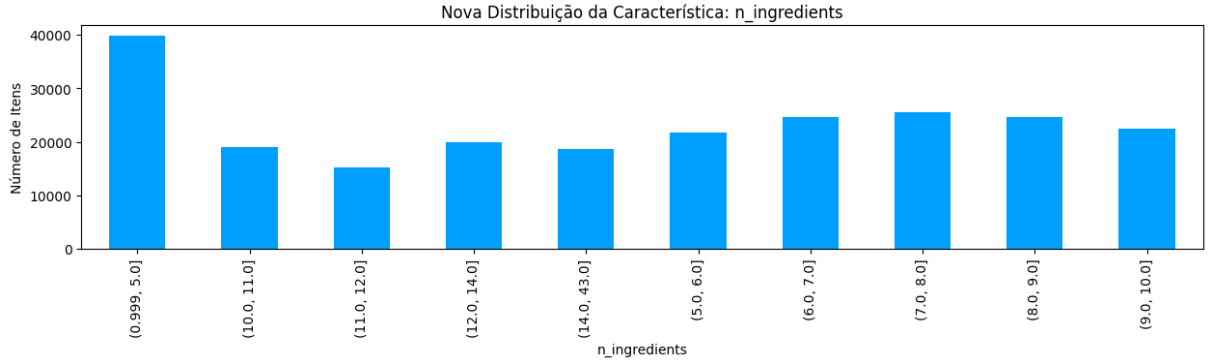


Figura 5.27: Nova distribuição da característica número de ingredientes.

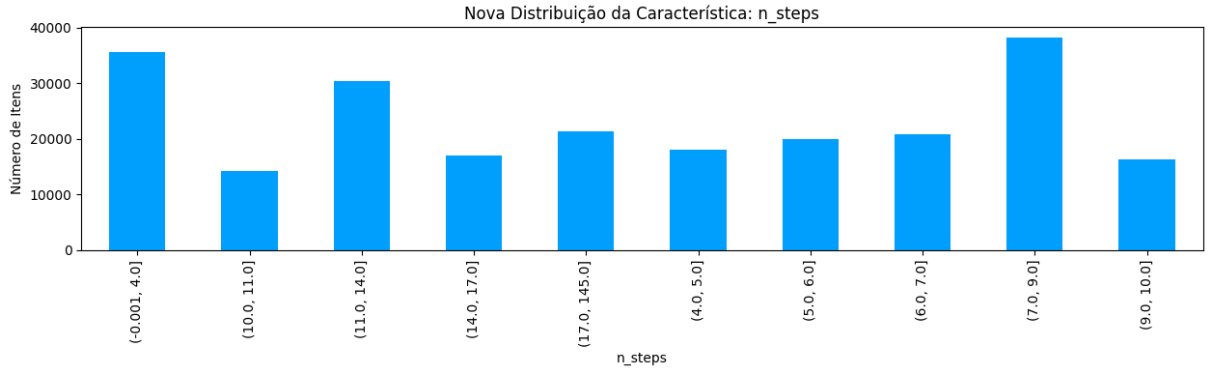


Figura 5.28: Nova distribuição da característica número de passos.

5.5.3 Goodreads

Por fim, considerando os dados do Goodreads. As categorias de menor frequência da característica formato (Figura 5.22) foram combinadas (Figura 5.29), por motivo semelhante ao discutido no MovieLens + IMDb. Também foi realizado um agrupamento das características ano de publicação e número de páginas (Figuras 5.30 e 5.31).

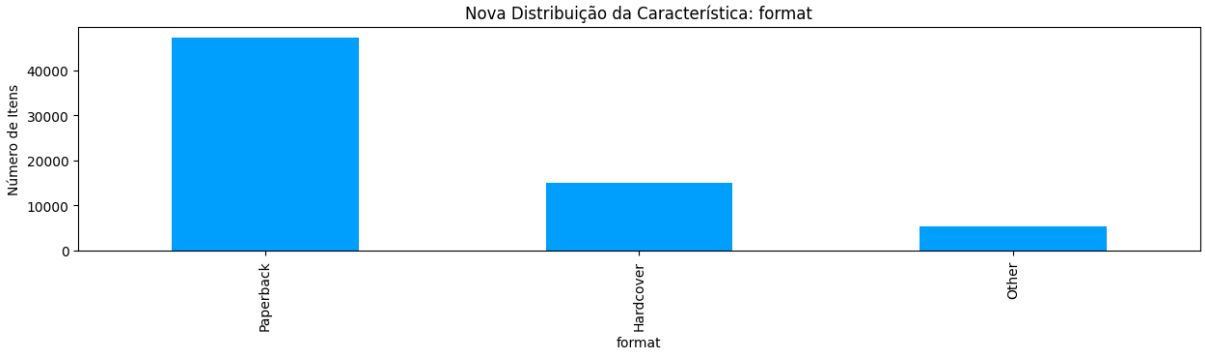


Figura 5.29: Nova distribuição da característica formato.

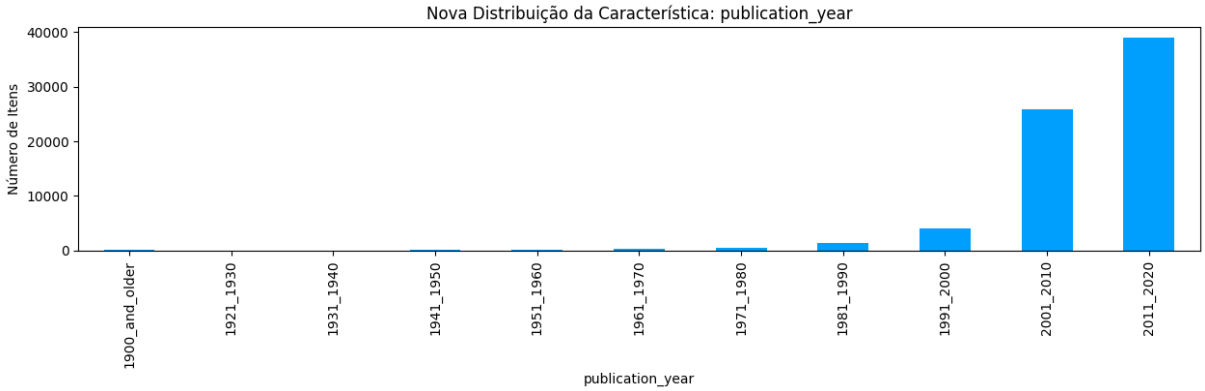


Figura 5.30: Nova distribuição da característica ano de publicação.

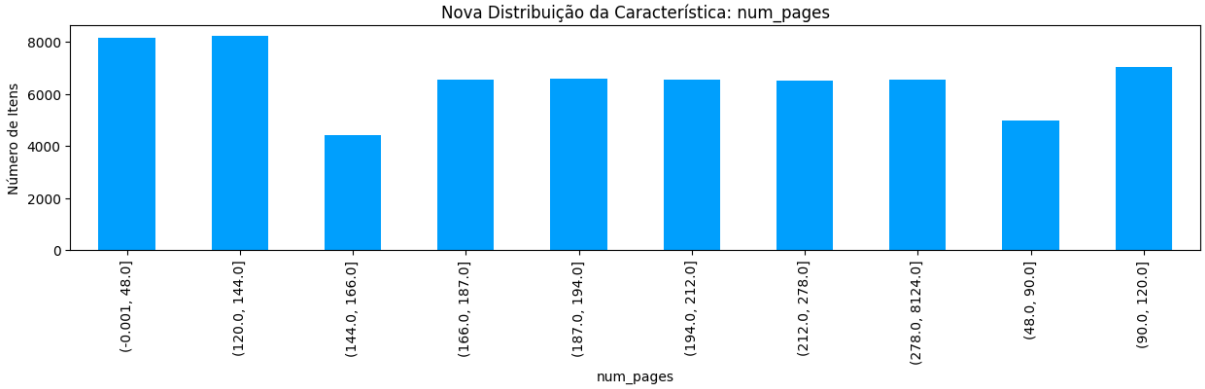


Figura 5.31: Nova distribuição da característica número de páginas.

5.6 ANÁLISE DE CORRELAÇÃO

Para finalizar, foram utilizadas 3 diferentes métricas de correlação para analisar os conjuntos de dados resultantes dos últimos processos, desejando-se remover características que contenham informações semelhantes, que contribuam de forma semelhante ao sistema. Sendo assim, foram removidas características que tivessem uma correlação maior que 0,7 (diretamente relacionadas) ou que tivessem uma correlação menor que -0,7 (inversamente relacionadas) com outra (limiares arbitrários). Para esse processo, a biblioteca *Pandas* foi utilizada, nela os valores de uma características são categorizadas e a correlação é feita entre as categorias resultantes.

5.6.1 MovieLens + IMDb

Nestes dados, observou-se uma correlação de 0,78 entre as características título principal e título original (Figura 5.32), sendo assim, apenas o título principal foi mantido.

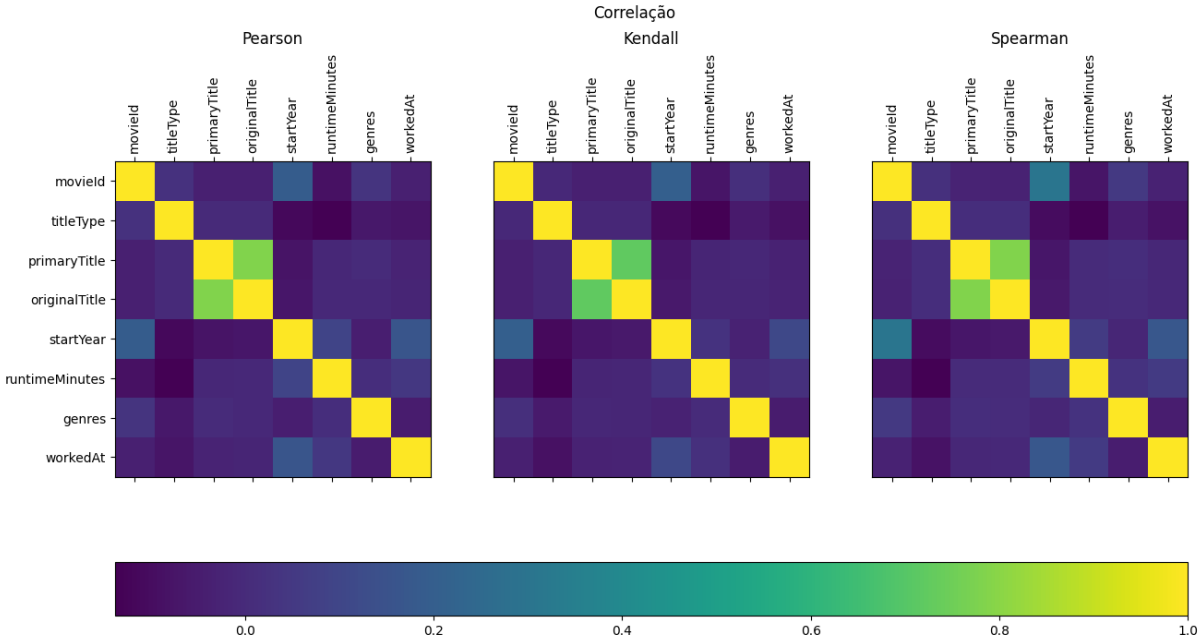


Figura 5.32: Correlação das características do MovieLens + IMDb.

5.6.2 Foodcom

Em relação ao Foodcom não foram observadas características com correlações maiores que 0,7 ou menores que -0,7 (Figura 5.33).

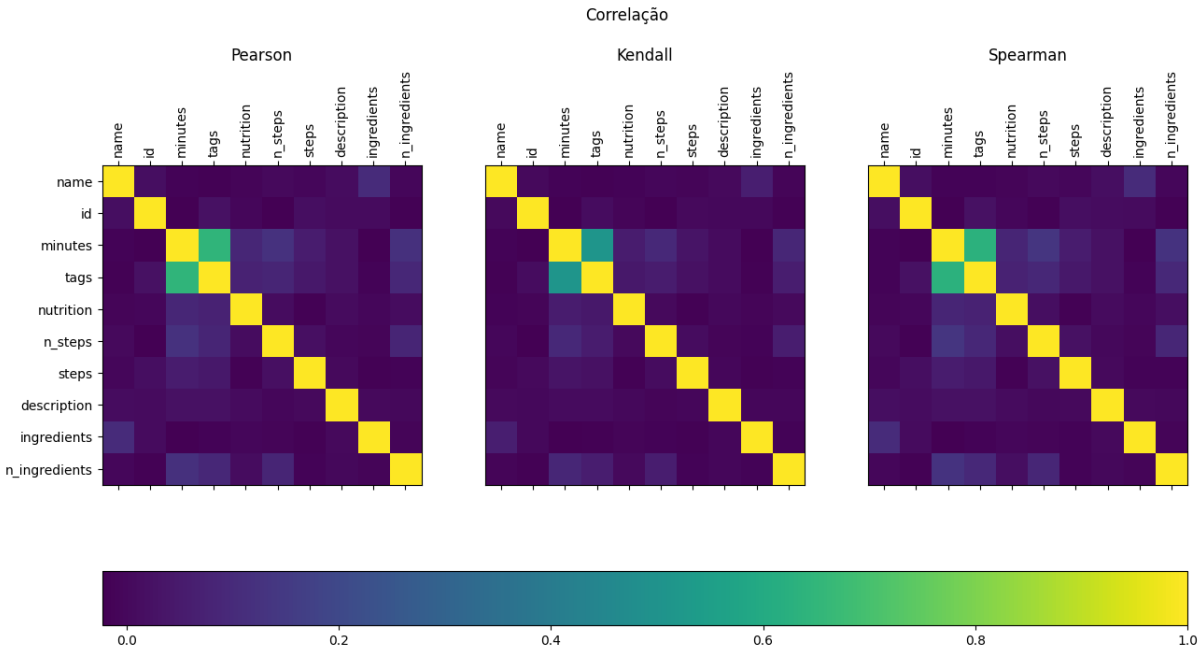


Figura 5.33: Correlação das características do Foodcom.

5.6.3 Goodreads

No Goodreads foi observado uma correlação de 0,81 entre o id de um livro e id da obra (Figura 5.34), logo, a segunda foi removida.

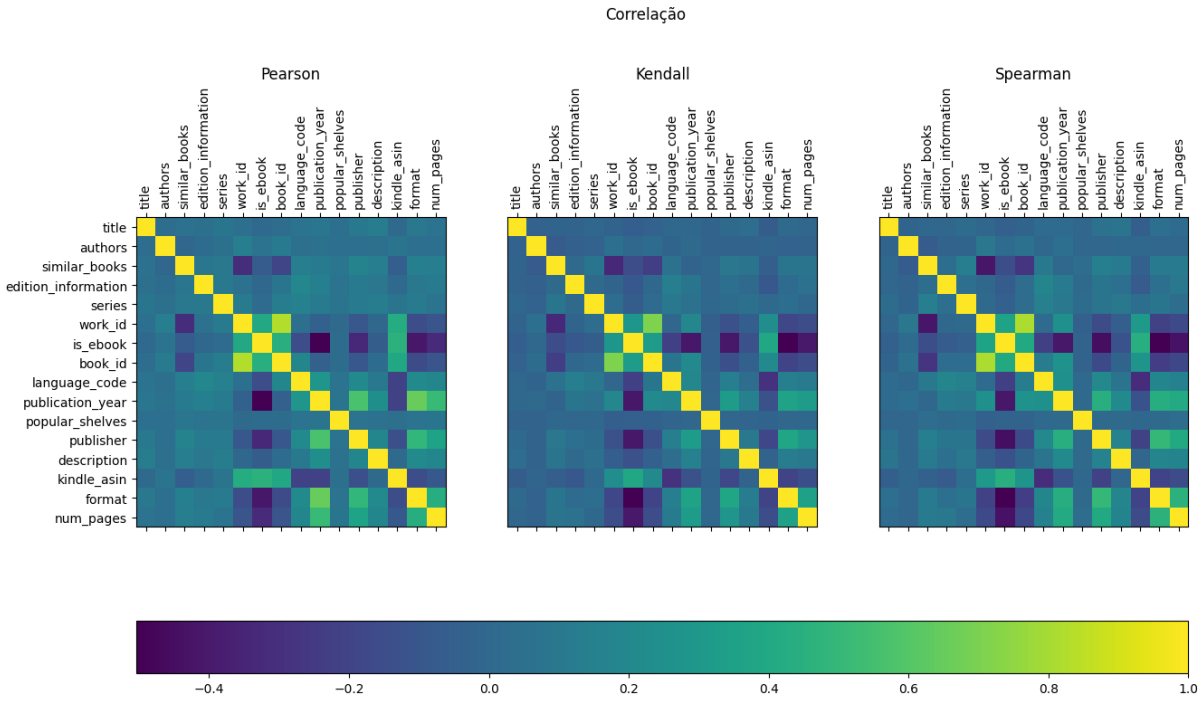


Figura 5.34: Correlação das características do Goodreads.

5.7 CONFIGURAÇÃO LIGHTFM

Alguns dos parâmetros padrões utilizados no LightFM são padrões da biblioteca e outros foram encontrados como sendo bons valores em experimentos preliminares com a biblioteca, nos quais ainda estavam sendo criados os *scripts* dos experimentos finais. Estes parâmetros são:

- Penalidade L2 em características de itens = 0.0 (padrão);
- Número de épocas para treinamento = 5 (encontrada nos experimentos preliminares);
- Função de perda = 'bpr' (encontrada nos experimentos preliminares);
- Normalização da representação latente das características = Verdadeiro (padrão);
- Dimensão das representações latentes = 30 dimensões (encontrada nos experimentos preliminares);
- Número de *threads*: 32.

Além destes valores padrões, foram variados os seguintes parâmetros:

- Dimensão das representações latentes = 10 e 50 dimensões;

Com relação a dimensão utilizada nas representações latentes, acreditamos que este parâmetro possa ter um impacto principalmente em características textuais, já que pelo volume de dados talvez não seja possível representar as características e itens sem remover informações relevantes, logo deseja-se testar o impacto de um menor e um maior número de dimensões.

5.8 ANÁLISE DE RESULTADOS

Nesta seção serão explorados os resultados obtidos para os conjuntos de dados Originais Sem Características (OSC), Originais Com Características (OCC), para os conjuntos de Dados Transformados e Filtrados (DTF) e para os conjuntos de dados transformados e filtrados + Seleção de Correlação ((+) SC). Estes resultados foram obtidos para os 3 conjuntos de interações de teste: interações de itens frios (itens com 0 interações no treinamento), interações de itens quentes (itens com mais de 0 interações no treinamento) e todas as interações de teste com a média de 4 diferentes métricas, *auc_score*, *precision@k*, *recall@k* e *reciprocal_rank*, onde *k* é igual a 10. Estas métricas fazem parte do LightFM. A métrica *auc_score* mede a qualidade da ordenação dos itens recomendados, a probabilidade de um item aleatório positivo (um item que usuário tem interação no teste) estar em uma posição acima de um item aleatório negativo (um item que o usuário não interagiu com no conjunto de teste) no *rank* de itens recomendados. Já *precision@k* e *recall@k*, medem respectivamente, a proporção de itens positivos nos *k* primeiros itens recomendados e o número de itens positivos nos *k* primeiros itens, dividido pela quantidade de itens positivos no teste. Ou seja, ambos analisam a quantidade de itens positivos, mas um deles pode ter uma nota perfeita (1.0) apenas com *k* itens positivos nos *k* primeiros itens recomendados, enquanto que o outro, por exemplo, pode ter uma nota perfeita com apenas 1 item positivo nos *k* primeiros itens, desde que o usuário tenha interagido com apenas 1 item no conjunto de testes. Por fim, *reciprocal_rank* não dá prioridade para a qualidade geral da ordenação dos itens, nem para o número de itens positivos, mas sim para a posição do primeiro item positivo no *ranking*, sendo definido como 1.0 dividido pela posição deste item.

Nas subseções seguintes os resultados serão explorados em maiores detalhes para cada conjunto de dados e ao fim será feita uma comparação entre os resultados de diferentes conjuntos de dados. Os resultados incluem dimensão da representação latente e tempo total de execução, que inclui ajuste do modelo, recomendação para todos os usuários e cálculo das métricas. Lembrando que nos Apêndices A e B estão disponíveis testes realizados com diferentes limiares para considerar um item frio ou quente, entretanto, como são limiares fixos, que não levam em consideração as condições de cada conjunto de dados, não serão apresentados aqui neste capítulo.

5.8.1 MovieLens + IMDb

As Tabelas 5.1 e 5.2 apresentam os resultados obtidos para o MovieLens + IMDb. Na primeira tabela podemos notar que originalmente e sem características (coluna OSC), o sistema de recomendação é incapaz de recomendar os itens frios, com 3 métricas zeradas. Ao utilizar os dados originais com todas as características (coluna OCC) é possível melhorar *precision@k* para 1,2% e *reciprocal_rank* para 4%, o que significa que em média, 1,2% dos itens frios foram recomendados nos 10 primeiros itens e que, em média, o primeiro item frio está na posição 25 da ordenação dos itens (4,6% de *reciprocal_rank*: 1/25 aproximadamente), resultados que não parecem impressionantes mas indicam que, entre mais de 62 mil itens, estes itens que nunca eram recomendados passaram a estar entre os primeiros 25 itens. A qualidade da ordenação dos itens também reflete isso, com *auc_score* aumentando de 0,607 para 0,858. Ao realizar os procedimentos descritos neste trabalho, os resultados se mantiveram aproximadamente iguais ao OCC (colunas DTF e (+) SC), entretanto houve uma redução do tempo de execução em aproximadamente 25%.

Em relação aos itens quentes a situação foi um pouco diferente. Os melhores resultados estão presentes na coluna OSC, o que faz sentido, já que como observado anteriormente, em um cenário onde há muitas interações, um SR que utiliza mecanismos de recomendação com base

em Filtragem Colaborativa (FC) tende a ter melhores resultados do que um que utiliza Filtragem Baseada em Conteúdo (FBC). Os resultados desta coluna, OSC, nos dizem que, em média, entre os 10 primeiros itens 2 são positivos (*precision@k* de 0,21) e que, em média, o primeiro item positivo está na segunda posição da lista (*reciprocal_rank* de 0,47: 1/2 aproximadamente). Comparando estes resultados com os das interações de itens frios e observando a queda de todas as métricas nos itens quentes após adicionar características, é possível chegar a conclusão que idealmente os dois tipos de itens não podem ser tratados da mesma forma, um possui melhores resultados com características e o outro sem elas. Já em relação a todas as interações, houve uma queda dos resultados, uma vez que a queda no desempenho dos itens quentes é mais significativa do que a melhora no desempenho dos itens frios.

Por fim, é possível notar que o acréscimo de características ao SR aumenta drasticamente o tempo de execução. Conscientes deste fato, testes foram realizados variando a dimensão do espaço latente que representa as características, a segunda tabela (Tabela 5.2) apresenta os resultados obtidos variando apenas este parâmetro para um valor menor e um maior (10 e 50 respectivamente). Como podemos observar, nesta situação, de alto número de interações, é possível reduzir ainda mais o tempo de execução, de 24 minutos para 13, mantendo resultados parecidos em situações de *cold start* de itens, sem prejudicar e nem melhorar os resultados de itens quentes, o que nos permite concluir que para essa situação um número alto de dimensões não é necessário e que, possivelmente, para restabelecer o desempenho dos itens quentes, a melhor escolha seria não empregar a mesma metodologia para a recomendação.

Observação sobre o número de interações de cada conjunto:

- Interações de itens frios no teste: 5.064.252;
- Interações de itens quentes no teste: 3.981.602;
- Interações totais no teste: 9.045.854;
- Interações totais no treinamento: 15.937.706.

Execução	OSC	OCC	DTF	(+) SC
dimensão da representação latente	30	30	30	30
AUC frios	0,607	0,859	0,858	0,858
precision@10 frios	0,000	0,012	0,010	0,011
recall@10 frios	0,000	0,006	0,004	0,004
reciprocal rank frios	0,000	0,046	0,036	0,037
AUC quentes	0,950	0,931	0,928	0,927
precision@10 quentes	0,211	0,165	0,126	0,114
recall@10 quentes	0,149	0,117	0,092	0,085
reciprocal rank quentes	0,470	0,367	0,281	0,251
AUC todas interações	0,757	0,890	0,888	0,888
precision@10 todas interações	0,210	0,177	0,135	0,125
recall@10 todas interações	0,066	0,056	0,044	0,041
reciprocal rank todas interações	0,469	0,382	0,289	0,261
tempo de execução (min)	4,338	32,688	26,320	24,618

Tabela 5.1: Média dos resultados do conjunto de dados MovieLens + IMDb para limiar 0.

Onde: OSC = Original Sem Características; OCC = Original Com Características; DTF = Dados Transformados e Filtrados; (+) SC = dados transformados e filtrados + Seleção Correlação.

Execução	(+) SC	(+) SC	(+) SC
dimensão da representação latente	30	10	50
AUC frios	0,858	0,856	0,861
precision@10 frios	0,011	0,012	0,011
recall@10 frios	0,004	0,005	0,004
reciprocal rank frios	0,037	0,041	0,039
AUC quentes	0,927	0,924	0,928
precision@10 quentes	0,114	0,121	0,112
recall@10 quentes	0,085	0,088	0,084
reciprocal rank quentes	0,251	0,272	0,242
AUC todas interações	0,888	0,885	0,890
precision@10 todas interações	0,125	0,132	0,122
recall@10 todas interações	0,041	0,042	0,040
reciprocal rank todas interações	0,261	0,283	0,253
tempo de execução (min)	24,618	13,342	40,020

Tabela 5.2: Média dos resultados do conjunto de dados MovieLens + IMDb para variações de dimensões da representação latente.

Onde: OSC = Original Sem Características; OCC = Original Com Características; DTF = Dados Transformados e Filtrados; (+) SC = dados transformados e filtrados + Seleção Correlação.

5.8.2 Foodcom

Para o conjunto de dados do Foodcom, os resultados estão presentes nas Tabelas 5.3 e 5.4. Como é possível notar novamente, sem uma estratégia para reduzir os efeitos do *cold start* o SR é incapaz de recomendar itens novos, além de ter um *auc_score* menor que 50%. Os resultados de *precision@k*, *recall@k* e *reciprocal_rank* se repetem nos itens quentes e, consequentemente, em todas as interações. Isso, por sua vez, pode ser um reflexo do alto número de itens comparado com o número total de interações e da alta esparsidade das interações, 99,998%, sendo a maior entre os 3 conjuntos de dados (MovieLens + IMDb com 99,8% e Goodreads com 99,99%), dificultando a recomendação de itens quentes e de *cold start*, uma vez que o LightFM utiliza de uma abordagem híbrida. Logo, não é possível recomendar itens quentes apenas com as informações de interações, como podemos observar na coluna OSC nos resultados quentes, e também não é possível construir representações latentes que ajudem a relacionar o gostar de uma característica com o gostar de outra, inclusive, podemos dizer que o SR poderia até ser melhor sem nenhuma interação, já que assim ele estabeleceria relações entre os itens apenas com base nas características.

Apesar dos resultados parecidos, com valores menores que 0,00, em 3 das métricas para os 3 diferentes cenários, ainda é possível notar uma diminuição no desempenho da recomendação de itens quentes após adicionar características e uma leve melhoria na recomendação de itens novos. O *auc_score*, a qualidade geral da ordenação por sua vez melhorou, sendo agora mais próximo da porcentagem de itens frios de teste comparado com os itens quentes. Por outro lado, o tempo de execução aumentou drasticamente, possível consequência do alto número de itens.

Por fim, foram realizados testes com diferentes dimensões, reduzindo consideravelmente o tempo total de execução de 111 minutos para 48 minutos, mantendo resultados parecidos.

Observação sobre o número de interações de cada conjunto:

- Interações de itens frios no teste: 248.462;

- Interações de itens quentes no teste: 162.421;
- Interações totais no teste: 410.883;
- Interações totais no treinamento: 721.484.

Execução	OSC	OCC	DTF
dimensão da representação latente	30	30	30
AUC frios	0,481	0,647	0,643
precision@10 frios	0,000	0,000	0,000
recall@10 frios	0,000	0,001	0,001
reciprocal rank frios	0,000	0,001	0,001
AUC quentes	0,611	0,668	0,661
precision@10 quentes	0,001	0,000	0,000
recall@10 quentes	0,004	0,001	0,001
reciprocal rank quentes	0,006	0,001	0,001
AUC todas interações	0,527	0,651	0,646
precision@10 todas interações	0,001	0,000	0,000
recall@10 todas interações	0,001	0,001	0,001
reciprocal rank todas interações	0,003	0,001	0,001
tempo de execução (min)	11,329	118,790	111,441

Tabela 5.3: Média dos resultados do conjunto de dados Foodcom para limiar 0.

Onde: OSC = Original Sem Características; OCC = Original Com Características; DTF = Dados Transformados e Filtrados; (+) SC = dados transformados e filtrados + Seleção Correlação.

Execução	DTF	DTF	DTF
dimensão da representação latente	30	10	50
AUC frios	0,643	0,636	0,641
precision@10 frios	0,000	0,000	0,000
recall@10 frios	0,001	0,000	0,001
reciprocal rank frios	0,001	0,000	0,001
AUC quentes	0,661	0,653	0,660
precision@10 quentes	0,000	0,000	0,000
recall@10 quentes	0,001	0,001	0,001
reciprocal rank quentes	0,001	0,001	0,001
AUC todas interações	0,646	0,639	0,644
precision@10 todas interações	0,000	0,000	0,000
recall@10 todas interações	0,001	0,001	0,001
reciprocal rank todas interações	0,001	0,001	0,001
tempo de execução (min)	111,441	48,717	178,732

Tabela 5.4: Média dos resultados do conjunto de dados Foodcom para variações de dimensões da representação latente.

Onde: OSC = Original Sem Características; OCC = Original Com Características; DTF = Dados Transformados e Filtrados; (+) SC = dados transformados e filtrados + Seleção Correlação.

5.8.3 Goodreads

Por fim, foram obtidos os resultados para o Goodreads (Tabelas 5.5 e 5.6). Estes que são semelhantes aos do MovieLens + IMDb e do Foodcom, uma vez que o SR deixou de ser totalmente incapaz de ordenar corretamente itens novos, tendo um *auc_score* maior após a utilização de características. Entretanto, os procedimentos realizados parecem não ter tido um impacto significativo, além do tempo de execução, em comparação com a utilização de todas as características originais. Da mesma forma que anteriormente, as interações de itens quentes tiveram valores menores com a adição de características e com os procedimentos feitos.

Novamente, também foram feitos testes de variação do número de dimensões do espaço latente. Entretanto, desta vez foi possível notar uma redução no desempenho da recomendação de itens de *cold start* ao utilizar 10 dimensões, que podemos concluir que este número não é suficiente para extrair todas as informações relevantes das características dos itens deste conjunto de dados. Porém, também não foi possível melhorar os resultados com um número maior de dimensões, sendo assim, seria necessário mais testes para encontrar um melhor número.

Observação sobre o número de interações de cada conjunto:

- Interações de itens frios no teste: 113.209;
- Interações de itens quentes no teste: 79.269;
- Interações totais no teste: 192.478;
- Interações totais no treinamento: 349.860.

Execução	OSC	OCC	DTF	(+) SC
dimensão da representação latente	30	30	30	30
AUC frios	0,496	0,816	0,816	0,817
precision@10 frios	0,000	0,001	0,001	0,001
recall@10 frios	0,000	0,005	0,005	0,006
reciprocal rank frios	0,000	0,005	0,005	0,005
AUC quentes	0,723	0,835	0,833	0,835
precision@10 quentes	0,019	0,002	0,001	0,002
recall@10 quentes	0,070	0,014	0,009	0,009
reciprocal rank quentes	0,077	0,014	0,007	0,006
AUC todas interações	0,592	0,827	0,825	0,826
precision@10 todas interações	0,012	0,002	0,002	0,002
recall@10 todas interações	0,031	0,009	0,007	0,007
reciprocal rank todas interações	0,049	0,012	0,007	0,007
tempo de execução (min)	0,564	14,955	12,916	12,611

Tabela 5.5: Média dos resultados do conjunto de dados Goodreads para limiar 0.

Onde: OSC = Original Sem Características; OCC = Original Com Características; DTF = Dados Transformados e Filtrados; (+) SC = dados transformados e filtrados + Seleção Correlação.

Execução	(+) SC	(+) SC	(+) SC
dimensão da representação latente	30	10	50
AUC frios	0,817	0,811	0,820
precision@10 frios	0,001	0,000	0,001
recall@10 frios	0,006	0,000	0,006
reciprocal rank frios	0,005	0,004	0,005
AUC quentes	0,835	0,828	0,836
precision@10 quentes	0,002	0,001	0,001
recall@10 quentes	0,009	0,005	0,007
reciprocal rank quentes	0,006	0,005	0,006
AUC todas interações	0,826	0,820	0,828
precision@10 todas interações	0,002	0,001	0,002
recall@10 todas interações	0,007	0,003	0,006
reciprocal rank todas interações	0,007	0,005	0,007
tempo de execução (min)	12,611	5,562	20,382

Tabela 5.6: Média dos resultados do conjunto de dados Goodreads para variações de dimensões da representação latente.

Onde: OSC = Original Sem Características; OCC = Original Com Características; DTF = Dados Transformados e Filtrados; (+) SC = dados transformados e filtrados + Seleção Correlação.

5.8.4 Considerações Finais

Ao analisar os 3 conjuntos de dados, alguns padrões foram notados e discutidos, um deles que ainda não foi explorado é o fato de que os resultados para os itens quentes reduziram após os procedimentos realizados, além da inclusão de características original, tanto no MovieLens + IMDb quanto no Goodreads. Acreditamos que o *binning* realizado tenha sido muito severo, que ele tenha ajudado o LightFM a realizar relações que não existiam. Se for esse o caso, poderíamos testar o impacto de diferentes quantidades de grupos, além de 10, porém por falta de tempo não foi possível.

Entre os padrões já relatados está o fato de que o desempenho de itens quentes é reduzido ao utilizar características. Para evitar isso podemos pensar em uma melhor estratégia para realizar recomendações que não tenha um impacto inverso entre os tipos de itens. Esta poderia realizar duas diferentes recomendações para um usuário, uma que envolve apenas itens frios e outra itens quentes, ou então realizar apenas uma, mas dedicar uma porcentagem desta lista de itens para itens frios.

Em relação aos conjuntos de dados, acredita-se que o Goodreads não tenha tido uma grande contribuição para as análises, já que seus resultados se demonstraram ser semelhantes aos outros. Inicialmente havia a ideia de utilizar o Foodcom e o Goodreads para analisar o impacto da dimensionalidade das representações latentes nos resultados, com o diferencial do Goodreads sendo promover uma análise do impacto de dados faltantes em um cenário semelhante. Entretanto, em nenhuma destas situações observou-se uma diferença expressiva, o que acreditamos ter sido causado pela alta esparsidade de ambos os conjuntos de dados.

De forma geral, houve melhorias na qualidade geral da ordenação dos itens (*auc_score*). O que não garante bons resultados nas outras métricas, mas nos diz que, se precisarmos recomendar um item além dos k primeiros, teremos uma lista de itens recomendados que de forma geral é boa.

Acreditamos que uma modificação poderia ter sido feita no processo de avaliação para gerar resultados que se aproximem mais de uma situação real e que possivelmente gere melhores resultados. Esta modificação envolveria alterar o processo de predições e avaliação inteiramente, para que a cada nova interação o modelo fosse atualizado com o novo conhecimento adquirido, esta alteração aproximaria mais o modelo de um SR *online*, em que predições são feitas e novas informações são recebidas (atualizando o modelo) constantemente. Entretanto, por limitações de escopo e de tempo, esta modificação não foi abordada neste trabalho. Preferiu-se utilizar o método de avaliação observado em Kula (2015).

6 CONCLUSÃO

Como pudemos perceber, sistemas de recomendação são altamente relevantes para se obter uma interação de qualidade entre usuário e item no mundo digital em que vivemos. Os usuários estão em contato constante com um grande volume dados e para garantir uma boa experiência, possivelmente incentivando um usuário a consumir algo, é necessário filtrar os catálogos por meio de sistemas de recomendação, estes que utilizarão de diferentes informações, objetivos e métodos para extrair dos catálogos aquilo que se considera mais relevante para um usuário. Em um cenário comum, é possível utilizar informações de interações (sistemas que utilizam filtragem colaborativa) anteriores e obter, de forma simples e rápida, boas recomendações. Entretanto, um dos maiores problemas enfrentados por esses sistemas de recomendação é lidar com a falta de conhecimento prévio sobre a opinião de usuários em relação a um item novo (*cold start* de itens), e sem este conhecimento um sistema de recomendação que utiliza de informações de interações é incapaz de realizar boas recomendações, nem mesmo é capaz de recomendar um novo item, como observamos no capítulo anterior. Desta forma, há a ideia de utilizar as informações próprias dos itens (sistemas que utilizam filtragem baseada em conteúdo) para estabelecer relações entre eles e realizar recomendações utilizando este novo conhecimento. Porém, também deseja-se utilizar as informações de interações quando possível, logo, podemos utilizar um sistema híbrido. Entretanto, como abordamos anteriormente, utilizar todas as informações dos itens da forma que as obtemos, não necessariamente é a melhor escolha, sendo assim, realizamos procedimentos como engenharia de características para transformar, extrair, selecionar e analisar características.

Neste trabalho fizemos o uso da ferramenta de recomendação híbrida LightFM e nela inserimos diferentes conjuntos de dados de cenários de recomendação de livros, filmes e receitas. Então fizemos a divisão dos dados de interação entre treinamento e teste, após ter agravado o *cold start* de itens, e buscamos predizer as interações de teste. Como pudemos observar, ao utilizar apenas informações de interações, o LightFM foi incapaz de recomendar itens novos do teste na forma de avaliação utilizada, mas obteve melhores resultados em interações de itens conhecidos. Ao inserir informações dos itens, notamos melhorias na predição de itens novos e redução da qualidade de itens já conhecidos, porém, também pudemos observar um aumento considerável no tempo de execução. Sendo assim, também tentamos utilizar técnicas de engenharia de características para melhorar os resultados observados, com foco em situações de *cold start* de itens, desta forma, realizamos processos como *binning*, combinação de características e filtragem utilizando dados de correlação, para tentar melhorar as medidas observadas ou então o custo-benefício da solução. Como observamos, em algumas situações, de menor esparsidade, estas modificações podem ter impactado de forma negativa as medidas, possivelmente por ter dificultado a utilização dos dados de interação, já em outras, de mais alta esparsidade, as medidas se mantiveram aproximadamente iguais, porém, em ambas as situações houve redução do tempo de processamento. Também pudemos notar que em alguns casos, o número de dimensões da representação latente pode ser reduzido sem ter impactos significativos nos resultados, melhorando ainda mais o custo-benefício. De forma geral podemos concluir que recomendar itens, especialmente itens novos, em situações de *cold start* e em alta esparsidade é uma tarefa difícil e que ainda há muito a se pesquisar. Entretanto, ficou claro que itens novos e conhecidos devem receber tratamentos diferentes e ajustes poderiam ser realizados para, possivelmente, melhorar os resultados, como ajustar o modelo com cada nova informação de interação.

Concluindo, acreditamos que demonstramos neste trabalho como utilizar técnicas de engenharia de características para aumentar o custo-benefício da solução obtida. Entretanto, não fomos capazes de reduzir ainda mais o impacto do problema de *cold start* de itens utilizando elas, como havíamos proposto na pergunta inicial no Capítulo 1 e em um dos objetivos específicos (Capítulo 2). Também acreditamos que fomos capazes de contextualizar o leitor em assuntos como sistemas de recomendação, *cold start*, engenharia de características (ainda que de forma mais ampla) e realizar experimentos de forma metodológica, realizando a divisão em treino e teste, utilizando diferentes métricas e analisando o problema em diferentes circunstâncias.

Finalizando, para trabalhos futuros há muitas direções para onde expandir, pode-se estudar o impacto de sistemas *online* no problema de *cold start*, pode-se explorar de forma mais apropriada o impacto das representações latentes na qualidade das relações estabelecidas entre itens e como esta informação pode ser utilizada de forma adequada para atenuar o *cold start*, também pode-se aprofundar uma análise do impacto de sistemas de recomendação inteligentes no *cold start*, que utilizem de estratégias híbridas para tomar decisões diferentes para situações diferentes, por fim, também pode-se buscar aprimorar as metodologias para aplicação de engenharia de características para atenuar o *cold start*, desenvolvendo formas para definir o número de grupos utilizado mais adequado ao realizar *binning*, por exemplo.

REFERÊNCIAS

Bernardis, C. e Cremonesi, P. (2021). Nfc: a deep and hybrid item-based model for item cold-start recommendation. *User Modeling and User-Adapted Interaction*, páginas 1–34.

Bobadilla, J., Ortega, F., Hernando, A. e Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-based systems*, 46:109–132.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12:331–370.

Burke, R. (2007). Hybrid web recommender systems. *The adaptive web: methods and strategies of web personalization*, páginas 377–408.

Deldjoo, Y., Schedl, M., Cremonesi, P. e Pasi, G. (2020). Recommender systems leveraging multimedia content. *ACM Computing Surveys (CSUR)*, 53(5):1–38.

Dong, G. e Liu, H. (2018). *Feature engineering for machine learning and data analytics*. CRC Press.

Duchi, J., Hazan, E. e Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

Golbeck, J. (2006). Generating predictive movie recommendations from trust in social networks. Em *Trust Management: 4th International Conference, iTrust 2006, Pisa, Italy, May 16-19, 2006. Proceedings 4*, páginas 93–104. Springer.

Guyon, I. e Elisseeff, A. (2006). An introduction to feature extraction. *Feature extraction: foundations and applications*, páginas 1–25.

Harper, F. M. e Konstan, J. A. (2015). The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19.

Jawaheer, G., Szomszor, M. e Kostkova, P. (2010). Comparison of implicit and explicit feedback from an online music recommendation service. Em *proceedings of the 1st international workshop on information heterogeneity and fusion in recommender systems*, páginas 47–51.

Koren, Y., Bell, R. e Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Kula, M. (2015). Metadata embeddings for user and item cold-start recommendations. *arXiv preprint arXiv:1507.08439*.

Majumder, B. P., Li, S., Ni, J. e McAuley, J. (2019). Generating personalized recipes from historical user preferences. *arXiv preprint arXiv:1909.00105*.

Nikolakopoulos, A. N., Ning, X., Desrosiers, C. e Karypis, G. (2021). Trust your neighbors: a comprehensive survey of neighborhood-based methods for recommender systems. *Recommender Systems Handbook*, páginas 39–89.

Panda, D. K. e Ray, S. (2022). Approaches and algorithms to mitigate cold start problems in recommender systems: a systematic literature review. *Journal of Intelligent Information Systems*, 59(2):341–366.

Ricci, F., Rokach, L. e Shapira, B. (2015). Recommender systems: introduction and challenges. *Recommender systems handbook*, páginas 1–34.

Schifferer, B., Deotte, C. e Oldridge, E. (2020). Tutorial: feature engineering for recommender systems. Em *Proceedings of the 14th ACM Conference on Recommender Systems*, páginas 754–755.

Shah, K., Salunke, A., Dongare, S. e Antala, K. (2017). Recommender systems: An overview of different approaches to recommendations. Em *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, páginas 1–4. IEEE.

Vaidya, N. e Khachane, A. (2017). Recommender systems-the need of the ecommerce era. Em *2017 International Conference on Computing Methodologies and Communication (ICCMC)*, páginas 100–104. IEEE.

Venkatesh, B. e Anuradha, J. (2019). A review of feature selection and its methods. *Cybernetics and information technologies*, 19(1):3–26.

Vig, J., Sen, S. e Riedl, J. (2012). The tag genome: Encoding community knowledge to support novel interaction. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(3):1–44.

Wan, M. e McAuley, J. (2018). Item recommendation on monotonic behavior chains. Em *Proceedings of the 12th ACM conference on recommender systems*, páginas 86–94.

Wang, S., Cao, L., Wang, Y., Sheng, Q. Z., Orgun, M. A. e Lian, D. (2021). A survey on session-based recommender systems. *ACM Computing Surveys (CSUR)*, 54(7):1–38.

Zhang, S., Yao, L., Sun, A. e Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)*, 52(1):1–38.

APÊNDICE A – RESULTADOS PARA O LIMIAR 10

Seguem abaixo os resultados obtidos considerando itens frios como tendo 10 ou menos interações.

A.1 MOVIELENS + IMDB

Número de interações:

- Interações de itens frios no teste: 5.085.877;
- Interações de itens quentes no teste: 3.959.977;
- Interações totais no teste: 9.045.854;
- Interações totais no treinamento: 15.937.706.

Execução	OSC	OCC	DTF	(+) SC	(+) SC	(+) SC
dimensão da representação latente	30	30	30	30	10	50
AUC frios	0,611	0,857	0,858	0,860	0,854	0,862
precision@10 frios	0,000	0,012	0,010	0,011	0,011	0,011
recall@10 frios	0,000	0,006	0,003	0,004	0,004	0,004
reciprocal rank frios	0,000	0,045	0,036	0,038	0,040	0,039
AUC quentes	0,950	0,930	0,928	0,928	0,923	0,929
precision@10 quentes	0,209	0,163	0,124	0,115	0,119	0,113
recall@10 quentes	0,146	0,116	0,091	0,086	0,089	0,085
reciprocal rank quentes	0,468	0,366	0,277	0,251	0,269	0,248
AUC todas interações	0,759	0,889	0,888	0,889	0,884	0,891
precision@10 todas interações	0,208	0,175	0,134	0,126	0,130	0,124
recall@10 todas interações	0,064	0,055	0,043	0,041	0,042	0,041
reciprocal rank todas interações	0,467	0,381	0,286	0,261	0,279	0,259
tempo de execução (min)	4,332	31,570	26,149	25,920	12,981	38,282

Tabela A.1: Resultados do conjunto de dados MovieLens + IMDb para o limiar 10.

Onde: OSC = Original Sem Características; OCC = Original Com Características; DTF = Dados Transformados e Filtrados; (+) SC = dados transformados e filtrados + Seleção Correlação.

A.2 FOODCOM

Número de interações:

- Interações de itens frios no teste: 331.078;
- Interações de itens quentes no teste: 79.805;
- Interações totais no teste: 410.883;

- Interações totais no treinamento: 721.484.

Execução	OSC	OCC	DTF	DTF	DTF
dimensão da representação latente	30	30	30	10	50
AUC frios	0,490	0,635	0,630	0,628	0,632
precision@10 frios	0,000	0,000	0,000	0,000	0,000
recall@10 frios	0,000	0,000	0,000	0,000	0,000
reciprocal rank frios	0,000	0,001	0,001	0,000	0,001
AUC quentes	0,666	0,708	0,697	0,693	0,698
precision@10 quentes	0,002	0,000	0,000	0,000	0,000
recall@10 quentes	0,006	0,001	0,001	0,001	0,001
reciprocal rank quentes	0,008	0,002	0,001	0,002	0,001
AUC todas interações	0,525	0,650	0,643	0,641	0,645
precision@10 todas interações	0,001	0,000	0,000	0,000	0,000
recall@10 todas interações	0,001	0,001	0,000	0,001	0,001
reciprocal rank todas interações	0,003	0,001	0,001	0,001	0,001
tempo de execução (min)	11,455	115,684	108,903	48,381	175,255

Tabela A.2: Resultados do conjunto de dados Foodcom para o limiar 10.

Onde: OSC = Original Sem Características; OCC = Original Com Características; DTF = Dados Transformados e Filtrados; (+) SC = dados transformados e filtrados + Seleção Correlação.

A.3 GOODREADS

Número de interações:

- Interações de itens frios no teste: 139.494;
- Interações de itens quentes no teste: 52.984;
- Interações totais no teste: 192.478;
- Interações totais no treinamento: 349.860.

Execução	OSC	OCC	DTF	(+) SC	(+) SC	(+) SC
dimensão da representação latente	30	30	30	30	10	50
AUC frios	0,508	0,811	0,812	0,806	0,807	0,811
precision@10 frios	0,000	0,001	0,001	0,001	0,001	0,001
recall@10 frios	0,000	0,002	0,005	0,004	0,002	0,005
reciprocal rank frios	0,000	0,004	0,005	0,005	0,004	0,005
AUC quentes	0,758	0,855	0,853	0,849	0,853	0,851
precision@10 quentes	0,023	0,003	0,002	0,001	0,002	0,001
recall@10 quentes	0,092	0,024	0,009	0,007	0,010	0,009
reciprocal rank quentes	0,092	0,017	0,007	0,006	0,008	0,006
AUC todas interações	0,589	0,828	0,828	0,823	0,825	0,827
precision@10 todas interações	0,012	0,002	0,002	0,001	0,001	0,002
recall@10 todas interações	0,031	0,011	0,006	0,005	0,005	0,007
reciprocal rank todas interações	0,049	0,012	0,007	0,006	0,007	0,006
tempo de execução (min)	0,566	14,900	12,837	12,304	5,467	20,019

Tabela A.3: Resultados do conjunto de dados Goodreads para o limiar 10.

Onde: OSC = Original Sem Características; OCC = Original Com Características; DTF = Dados Transformados e Filtrados; (+) SC = dados transformados e filtrados + Seleção Correlação.

APÊNDICE B – RESULTADOS PARA O LIMIAR 20

Seguem abaixo os resultados obtidos considerando itens frios como tendo 20 ou menos interações.

B.1 MOVIELENS + IMDB

Número de interações:

- Interações de itens frios no teste: 5.098.990;
- Interações de itens quentes no teste: 3.946.864;
- Interações totais no teste: 9.045.854;
- Interações totais no treinamento: 15.937.706.

Execução	OSC	OCC	DTF	(+) SC	(+) SC	(+) SC
dimensão da representação latente	30	30	30	30	10	50
AUC frios	0,611	0,860	0,857	0,860	0,856	0,861
precision@10 frios	0,000	0,013	0,010	0,011	0,010	0,011
recall@10 frios	0,000	0,006	0,004	0,004	0,004	0,004
reciprocal rank frios	0,000	0,048	0,036	0,037	0,036	0,038
AUC quentes	0,950	0,931	0,927	0,928	0,924	0,928
precision@10 quentes	0,211	0,166	0,121	0,113	0,119	0,111
recall@10 quentes	0,149	0,118	0,089	0,085	0,088	0,083
reciprocal rank quentes	0,471	0,371	0,267	0,249	0,269	0,241
AUC todas interações	0,758	0,891	0,887	0,889	0,886	0,890
precision@10 todas interações	0,210	0,178	0,131	0,123	0,129	0,121
recall@10 todas interações	0,066	0,056	0,042	0,040	0,042	0,040
reciprocal rank todas interações	0,469	0,387	0,276	0,259	0,277	0,252
tempo de execução (min)	4,022	31,399	28,806	26,476	13,220	39,928

Tabela B.1: Resultados do conjunto de dados MovieLens + IMDb para o limiar 20.

Onde: OSC = Original Sem Características; OCC = Original Com Características; DTF = Dados Transformados e Filtrados; (+) SC = dados transformados e filtrados + Seleção Correlação.

B.2 FOODCOM

Número de interações:

- Interações de itens frios no teste: 353.045;
- Interações de itens quentes no teste: 57.838;
- Interações totais no teste: 410.883;

- Interações totais no treinamento: 721.484.

Execução	OSC	OCC	DTF	DTF	DTF
dimensão da representação latente	30	30	30	10	50
AUC frios	0,496	0,637	0,631	0,629	0,631
precision@10 frios	0,000	0,000	0,000	0,000	0,000
recall@10 frios	0,000	0,001	0,000	0,000	0,000
reciprocal rank frios	0,000	0,001	0,001	0,000	0,001
AUC quentes	0,691	0,729	0,716	0,712	0,713
precision@10 quentes	0,002	0,000	0,000	0,001	0,000
recall@10 quentes	0,010	0,002	0,001	0,003	0,001
reciprocal rank quentes	0,011	0,003	0,002	0,003	0,001
AUC todas interações	0,526	0,653	0,645	0,642	0,645
precision@10 todas interações	0,001	0,000	0,000	0,000	0,000
recall@10 todas interações	0,001	0,001	0,000	0,001	0,000
reciprocal rank todas interações	0,003	0,001	0,001	0,001	0,001
tempo de execução (min)	11,195	114,096	107,204	47,225	172,472

Tabela B.2: Resultados do conjunto de dados Foodcom para o limiar 20.

Onde: OSC = Original Sem Características; OCC = Original Com Características; DTF = Dados Transformados e Filtrados; (+) SC = dados transformados e filtrados + Seleção Correlação.

B.3 GOODREADS

Número de interações:

- Interações de itens frios no teste: 149.000;
- Interações de itens quentes no teste: 43.478;
- Interações totais no teste: 192.478;
- Interações totais no treinamento: 349.860.

Execução	OSC	OCC	DTF	(+) SC	(+) SC	(+) SC
dimensão da representação latente	30	30	30	30	10	50
AUC frios	0,521	0,809	0,806	0,806	0,807	0,811
precision@10 frios	0,000	0,001	0,001	0,001	0,001	0,001
recall@10 frios	0,000	0,002	0,004	0,005	0,002	0,006
reciprocal rank frios	0,001	0,004	0,005	0,005	0,004	0,005
AUC quentes	0,768	0,862	0,857	0,857	0,865	0,861
precision@10 quentes	0,025	0,004	0,002	0,002	0,002	0,002
recall@10 quentes	0,106	0,025	0,010	0,011	0,011	0,010
reciprocal rank quentes	0,103	0,020	0,007	0,007	0,008	0,007
AUC todas interações	0,592	0,827	0,824	0,824	0,826	0,828
precision@10 todas interações	0,013	0,002	0,002	0,002	0,001	0,002
recall@10 todas interações	0,031	0,011	0,007	0,007	0,005	0,007
reciprocal rank todas interações	0,050	0,012	0,006	0,007	0,006	0,007
tempo de execução (min)	0,560	14,689	12,752	12,407	5,278	19,768

Tabela B.3: Resultados do conjunto de dados Goodreads para o limiar 20.

Onde: OSC = Original Sem Características; OCC = Original Com Características; DTF = Dados Transformados e Filtrados; (+) SC = dados transformados e filtrados + Seleção Correlação.