

# RECOMENDAÇÕES DE PIZZA EM GRUPO: UMA ABORDAGEM PARA APRIMORAR A EXPERIÊNCIA DO USUÁRIO E AS VENDAS EM REFEIÇÕES COLETIVAS

Endrius Ewald - endrius.ewald@edu.pucrs.br, Enrique Dutra - enrique.dutra@edu.pucrs.br  
Orientador: Duncan D. Alcoba Ruiz - duncan.ruiz@pucrs.br

**Resumo**—Em um contexto globalizado, permeado por uma sobrecarga de informações online, os usuários enfrentam desafios consideráveis ao explorar a vasta gama de opções disponíveis. A complexidade desse cenário resulta em experiências de consumo menos eficientes e, por vezes, frustrantes. Diante desse panorama, a personalização e ofertas atrativas emergem como elementos cruciais para aprimorar a conexão entre usuários e itens, visando a melhoria da experiência do consumidor. Empresários, cientes dessa demanda crescente, buscam sistemas capazes de proporcionar recomendações personalizadas, impulsionando tanto as vendas quanto a satisfação do usuário. Este trabalho concentra-se na dinâmica desafiadora das recomendações em atividades coletivas, com foco na sugestão de pizzas. Apresentamos um sistema inovador que propõe um prato coletivo ideal, considerando as preferências individuais de todos os envolvidos. Essa abordagem inovadora utiliza um modelo de recomendação híbrido, incorporando tanto a filtragem colaborativa, através do método de fatorização de matrizes com valores latentes, quanto a filtragem baseada em conteúdo, por meio de um algoritmo de agrupamento. A combinação dessas técnicas, através da agregação de preferências, visa criar a pizza perfeita, promovendo uma experiência culinária única e satisfatória para todos os participantes

**Palavras-chave**—Sistemas de Recomendação, Sistemas de Recomendação em Grupo, Filtragem Colaborativa, Filtragem Baseada em Conteúdo, Agregação de Preferências, Alimentação

**Abstract**—In a globalized context, permeated by an overload of online information, users face considerable challenges when exploring the vast array of available options. The complexity of this scenario results in less efficient and often frustrating consumption experiences. In light of this panorama, personalization and attractive offers emerge as crucial elements to enhance the connection between users and items, aiming to improve the consumer experience. Entrepreneurs, aware of this growing demand, seek systems capable of providing personalized recommendations, boosting both sales and user satisfaction. This work focuses on the challenging dynamics of recommendations in collective activities, with a focus on pizza suggestions. We present an innovative system that proposes an ideal collective dish, considering the individual preferences of all participants. This innovative approach uses a hybrid recommendation model, incorporating both collaborative filtering through the matrix factorization method with latent values, and content-based filtering through a clustering algorithm. The combination of these techniques, through preference aggregation, aims to create the perfect pizza, promoting a unique and satisfying culinary experience for all participants.

**Keywords**—Recommender Systems, Group Recommender Systems, Collaborative Filtering, Content-based Filtering, Preference Aggregation, Food Consumption.

## I. INTRODUÇÃO

Nos últimos anos, o advento da internet e o crescimento exponencial de plataformas digitais têm transformado profundamente a maneira como consumidores interagem com produtos e serviços. A abundância de informações e opções disponíveis pode sobrecarregar os usuários, tornando a escolha de um produto específico uma tarefa desafiadora. Em resposta a essa complexidade, surgem os sistemas de recomendação, ferramentas fundamentais que ajudam a personalizar e simplificar a experiência do usuário, conectando-os de maneira eficaz aos itens que melhor atendem às suas preferências.

A eficácia desses sistemas não se restringe apenas a indivíduos; atividades realizadas em grupo também se beneficiam significativamente de recomendações personalizadas. Uma dessas atividades, que envolve a tomada de decisão coletiva e pode ser especialmente desafiadora, é a escolha de alimentos para consumo em grupo. Este trabalho propõe um sistema inovador para recomendação de pizzas, um alimento versátil e popular, ideal para ilustrar como um recomendador de grupo pode facilitar a escolha e melhorar a experiência gastronômica compartilhada.

O objetivo deste estudo é desenvolver um sistema que não apenas recomende pizzas que satisfaçam as preferências de todos os participantes, levando em consideração suas particularidades individuais, mas também otimize o processo de escolha, reduzindo o tempo dedicado à decisão e aumentando a satisfação geral dos usuários. Este documento está estruturado em diferentes capítulos que abordam desde os fundamentos de machine learning, campo no qual os melhores algoritmos de recomendação estão inseridos, até uma análise detalhada dos algoritmos específicos mais relevantes para nossa proposta.

Ao final deste estudo, espera-se não apenas contribuir para a área de sistemas de recomendação, mas também fornecer uma solução prática e eficaz para uma necessidade cotidiana de consumo alimentar em grupo. Através dessa abordagem, buscamos não só melhorar a experiência do usuário final, mas também impactar positivamente a eficiência e a rentabilidade dos serviços de entrega de alimentos.

O documento apresenta a seguinte estrutura: primeiramente, revisa a literatura existente sobre sistemas de recomendação, discutindo diversas abordagens e técnicas relevantes como filtragem colaborativa, filtragem baseada em conteúdo e métodos híbridos. Em seguida, detalha a metodologia empregada para a recomendação de pizzas em grupos, abordando a construção do modelo de recomendação com uso da filtragem colaborativa e da fatoração de matriz (SVD), além do processo de coleta e simulação de dados de avaliações de pizzas e as métricas de avaliação utilizadas. A seção de testes e experimentos descreve os procedimentos experimentais para validar a eficácia do modelo proposto, incluindo a variação no número de usuários e sabores de pizza, bem como os métodos de avaliação utilizados (RMSE, MAE). Posteriormente, apresenta uma análise detalhada dos resultados obtidos, discutindo as principais descobertas, insights e implicações dos resultados. Finalmente, conclui com um resumo das principais contribuições do estudo, destacando as vantagens da abordagem proposta, além de discutir as limitações do estudo e propor direções para trabalhos futuros que possam aprimorar ainda mais o modelo de recomendação desenvolvido.

## II. TRABALHOS RELACIONADOS

Existem diversas aplicações de sistemas de informação que visam recomendar itens customizados aos usuários, abrangendo desde motores de busca e indexadores até lojas online e aplicativos de streaming. Estes sistemas empregam uma vasta gama de abordagens, técnicas e tecnologias para oferecer os melhores produtos e anúncios aos usuários do sistema.

Esta seção discute alguns dos trabalhos relevantes na área, destacando aqueles que melhor se relacionam com o problema a ser abordado neste documento. Um dos marcos significativos é o *PolyLens: A Recommender System for Groups of Users* [OCO+01], um dos primeiros sistemas de recomendação projetados não apenas para indivíduos, mas para grupos de usuários. Esta abordagem é particularmente útil em atividades grupais como visitas a restaurantes e cinemas.

Outro trabalho relevante é *An Intelligent Multi-Agent Recommender System for Human Capacity Building* [MSM08], que se destaca por sua capacidade de adaptar-se à inclusão de novos itens sem necessidade de re-treinamento do modelo. Apesar de utilizar uma abordagem de sistemas multiagentes, diferentemente da aplicação proposta neste trabalho, onde os agentes são unidades ativas de busca, este trabalho oferece insights valiosos sobre como sistemas adaptativos podem ser implementados.

Além disso, *Matrix Factorization Techniques for Recommender Systems* [KBV09a], de Yehuda e Robert, é outro estudo relevante que explora técnicas avançadas de recomendação, como a filtragem colaborativa e modelos de fatores latentes. Essas técnicas são conhecidas por sua capacidade de inferir preferências de forma autônoma, sem depender de informações explícitas dos usuários.

Esses trabalhos fornecem uma base sólida de conhecimento e inspiração para o desenvolvimento do sistema proposto neste trabalho, enfatizando a importância de adaptar técnicas avançadas de recomendação para contextos específicos, como a seleção de pizzas em grupo.

### A. *PolyLens: A Recommender System for Groups of Users*

[OCO+01]

PolyLens é um sistema de recomendação baseado em filtragem colaborativa desenvolvido para sugerir filmes a grupos de indivíduos que desejam assistir juntos. Como uma extensão do MovieLens [Mov], o PolyLens diferencia-se ao considerar não apenas as preferências individuais dos usuários, mas também as dinâmicas de grupo dentro do GroupLens [Gro], um grupo de pesquisa da Universidade de Minnesota especializado em sistemas de recomendação e comunidades online.

Devido à natureza emergente dos sistemas de recomendação na época de seu desenvolvimento, os autores enfrentaram o desafio de construir o sistema do zero. Eles focaram principalmente em dois aspectos: como integrar as preferências individuais em decisões de grupo e como formar esses grupos, considerando o papel e a interação dos indivíduos, além das questões de privacidade dos dados.

A metodologia central empregada pelo PolyLens são os algoritmos de agrupamento, que comparam um usuário com outros usuários similares em termos de preferências. A premissa é que usuários com gostos similares tendem a apreciar os mesmos filmes, mesmo que ainda não tenham assistido juntos.

Para a tomada de decisão social, os autores escolheram a abordagem de "Menor Miséria", um conceito da teoria da decisão que busca minimizar o desconforto geral entre os membros do grupo, favorecendo o consenso sobre a escolha do filme. Além disso, o sistema evita recomendar filmes que algum dos membros do grupo já tenha visto, considerando que, em contextos como o cinema, repetir a experiência pode não ser desejável para todos os envolvidos.

A avaliação do sistema foi conduzida ao longo de nove meses, envolvendo 819 usuários formando 338 grupos, com mais de 7 mil solicitações de recomendação e 114 mil recomendações fornecidas. Os resultados mostraram que os usuários valorizam e encontram útil os sistemas de recomendação, mesmo que isso implique em uma troca de privacidade por recomendações mais precisas. Contudo, muitos usuários expressaram descontentamento com a abordagem de "Menor Miséria", ressaltando a necessidade de aprimoramento das funções sociais do sistema.

O artigo não detalha as tecnologias específicas utilizadas, mencionando apenas o uso de uma interface web para interação com os usuários e um provedor de e-mail para convites aos grupos.

Embora o PolyLens ofereça insights valiosos sobre sistemas de recomendação para grupos, este estudo concentra-se em aspectos da formação e dinâmica de grupos, não diretamente relacionados ao objetivo deste trabalho: entender como agregar as preferências coletivas de um grupo de usuários para recomendar a pizza ideal.

### B. *An Intelligent Multi-Agent Recommender System for Human Capacity Building* [MSM08]

Este trabalho apresenta um sistema multiagente para recomendar cursos de capacitação a profissionais de engenharia, destacando-se pela sua capacidade de adaptar recomendações a diferentes subáreas dentro do campo da engenharia. Enquanto

algumas áreas podem compartilhar interesses semelhantes em cursos, outras podem ter necessidades específicas e distintas.

Os agentes neste sistema desempenham papéis específicos: um agente extrai informações e palavras-chave dos sites de cursos usando o framework de web scraping scRU-BYit! [Szi]. Estes dados são armazenados em um banco de dados MySQL e posteriormente consultados por outro agente, responsável por classificar e filtrar os cursos com base no perfil do usuário.

O agente de recomendação utiliza o motor de busca Sphinx [Aks] para consultas ao banco de dados, e emprega uma rede neural de classificação perceptron de múltiplas camadas, treinada por 400 épocas com o algoritmo de backpropagation [RM87]. Esta rede neural é fundamental para a inteligência do sistema, utilizando funções de ativação  $F_{outer}$  e  $F_{inner}$  para classificação eficiente dos cursos.

A modelagem dos usuários e dos cursos utiliza palavras-chave para representar os cursos, em vez de identificadores diretos, permitindo a inclusão dinâmica de novos cursos sem a necessidade de treinamento adicional da rede neural. Esta abordagem facilita a escalabilidade do sistema sem comprometer a precisão das recomendações.

A interface com o usuário foi desenvolvida utilizando o framework Symfony [Pot], proporcionando uma interação intuitiva através de uma página web.

O sistema alcançou um sucesso significativo, obtendo uma precisão de 90% com uma única camada oculta na rede neural. Isso resultou em uma redução substancial na sobrecarga de informações enfrentada pelos usuários ao buscar cursos de capacitação específicos em engenharia, demonstrando a eficácia da abordagem multiagente e do uso de redes neurais para recomendação personalizada.

### C. Matrix Factorization Techniques for Recommender Systems [KBV09a]

Este artigo contextualiza a sobrecarga de informações enfrentada pelos consumidores modernos, destacando a necessidade de sistemas de recomendação para filtrar e direcionar escolhas em meio a uma vasta gama de produtos disponíveis. Ele compara o papel desses sistemas ao de um atendente em uma loja física, que auxilia os clientes durante todo o processo de compra, adaptado ao ambiente online.

Os autores discutem duas abordagens principais para sistemas de recomendação: baseada em conteúdo e colaborativa, com esta última sendo preferida devido à sua aplicabilidade genérica e eficácia diante da diversidade de produtos e informações disponíveis. Dentro da abordagem colaborativa, são comparados os algoritmos baseados em vizinhança e os algoritmos de fatores latentes, sendo este último o foco principal do trabalho.

O método de Fatorização de Matrizes é explorado em detalhes, comparando-o com o método SVD (Decomposição de Valores Singulares), ambos pertencentes à categoria de modelos de fatores latentes. O artigo apresenta as fórmulas e explica como o modelo de fatorização de matrizes calcula fatores latentes a partir dos dados do usuário, permitindo inclusive a incorporação de informações externas como dinâmicas temporais e níveis de confiança.

As tecnologias específicas utilizadas não são mencionadas, pois o foco está na descrição matemática e algorítmica dos métodos, permitindo sua implementação em diversas plataformas. O artigo conclui mencionando o impacto da competição da Netflix em 2006, que impulsionou significativamente a pesquisa em sistemas de recomendação, oferecendo um prêmio substancial para melhorias na precisão das recomendações.

A seguir, será realizada uma análise dos trabalhos relacionados citados neste capítulo.

## III. SOLUÇÃO PARA RECOMENDAÇÃO DE PIZZAS EM GRUPO

### A. Base de Dados

Nessa seção serão abordados os dados utilizados, seu processo de escolha, construção, transformação e avaliação.

1) *Busca dos Dados*: Foi realizada uma pesquisa utilizando várias consultas diferentes, como **pizza**, **pizza dataset**, **pizza data**, entre outras, em diversas bases de dados bem conhecidas como Kaggle[Kag], OpenML[ope], Dataset Search do Google[Dat]. Foram enfrentados vários desafios na obtenção de dados específicos de pedidos de pizza. Alguns *datasets* foram encontrados, porém, eles eram principalmente focados em preços[kni], tempos de entrega, imagens de pizza[gau] e informações nutricionais, como exemplificado no *dataset* de pizzas encontrado no data.world[sdh]. Além disso, outros *datasets* encontrados abrangiam dados sobre comidas em geral, mas não forneciam avaliações específicas dos itens de pizza.

Diante desse cenário foram consultadas uma série de pizzarias locais para a obtenção de dados dos seus pedidos, ou a realização de uma parceria no desenvolvimento do sistema proposto. Os estabelecimentos foram contrários a ideia por motivos variados.

Se optou por utilizar o cardápio disponível em um dos *datasets* encontrados, o Maven Pizza Challenge Dataset[Moh23] cuja finalidade também era a análise de preços para se ter pelo menos uma fonte de dados externa.

No fim foi decidido simular os dados das avaliações com o intuito de não se abandonar a ideia de trabalho proposto. À seguir será abordado como um conjunto de dados foi construído com a finalidade de simular o funcionamento do sistema proposto.

2) *Simulação dos Dados*: O dado de entrada necessário em um sistema de recomendação possui um formato simples e prático, é composto necessariamente por uma tripla(3-upla) que contém um identificador do usuário do sistema, um identificador do item avaliado e a nota atribuída por esse usuário ao respectivo item, sendo assim, do seguinte formato ( $\langle user\_id \rangle$ ,  $\langle item\_id \rangle$ ,  $\langle rate \rangle$ ). Esses são os dados essenciais para que qualquer sistema de recomendação opere. Em alguns casos especiais os sistemas de recomendação podem fazer uso de dados extras como marcas temporais, entre outros. Este não é o caso do sistema proposto. A essa tripla se dá o nome de avaliação. Um sistema de recomendação é então alimentado por avaliações. O quanto mais avaliações um sistema de recomendação possuir mais preciso será nas recomendações futuras, ou seja, é necessário uma lista desses dados.

TIPOS DE PIZZA
carne vermelha
carne branca
carne de peixe
vegetais
queijos
frutas
picante
doce
chocolate preto
chocolate branco

Tabela I  
TIPOS DE PIZZA

TIPO DE PIZZA	CRITÉRIO DE ATRIBUIÇÃO
carne vermelha	Possui bovinos.
carne branca	Possui suínos ou frango.
carne de peixe	Possui peixe ou frutos do mar.
vegetais	Possui 3 ou mais vegetais na receita. Alho e Cebola não contam.
queijos	Possui 2 ou mais queijos.
frutas	Possui alguma fruta na composição.
picante	Possui pimenta.
doce	Pizza doce.
chocolate preto	Possui chocolate preto.
chocolate branco	Possui chocolate branco.

Tabela II  
CRITÉRIOS DE ATRIBUIÇÃO

A criação da lista de avaliações simulada foi feita da seguinte maneira: primeiramente foi definido um conjunto de **tipos de pizza**, com base em conhecimento empírico sobre esse tipo de comida, que pode ser observado na Tabela I. Na sequência foram criadas **regras** para atribuição do tipo de pizza a cada receita<sup>1</sup> disponível no cardápio obtido no *dataset* Maven[Moh23] com base nos **ingredientes** de cada receita conforme a Tabela II. Em seguida foram criados diversos usuários simulados(**personas**) e a cada persona foi atribuída duas preferências de tipos de pizza, uma **primária** e outra **secundária**. Foi criado um usuário para cada persona, e uma persona para cada combinação de preferência primária e secundária. E por fim foram atribuídas **notas** as receitas de pizza para cada persona com base no tipo de sua preferência de acordo com os critérios evidenciados na Tabela III.

Sabores doces não foram utilizados devido à ausência deles no *dataset* Maven. Além disso informamos que foram incluídos 4 usuários com as avaliações geradas de maneira totalmente aleatória a fim de se criar um ruído no modelo.

3) *Restrição Alimentar*: Levando em conta que algumas pessoas possuem algum tipo de restrição alimentar, o sistema também contempla uma lista de restrições que o usuário pode sinalizar itens dos quais não poderia comer de jeito

<sup>1</sup>Para fins informativos, nesse trabalho os termos Receita(s), Sabor(es), Item(ns), assim como Pizza(s) são sinônimos. Se referem às receitas que podem ser escolhidas quando uma pedida é pedida. e.g. "4 Queijos".

PREFERÊNCIA	INTERVALO DE NOTAS
Primário	[9,10]
Secundário	[7,8]
Caso contrário	[1,6]

Tabela III  
CRITÉRIO PARA ATRIBUIÇÃO DE NOTAS

algum. Essa lista será utilizada para retirar as pizzas da recomendação do grupo com a finalidade de evitar possíveis incidentes como reações alérgicas alimentares.

4) *Análise Estática dos Dados*: Os dados utilizados nos experimentos apresentam as seguintes características detalhadas:

- Os tipos de pizza mais frequentemente avaliados são: Carne Branca, Vegetais e Carne Vermelha.
- Os tipos de pizza com menor número de avaliações são: Frutas, Picantes e Queijo.

A Figura 1 demonstra a distribuição do número de avaliações por tipo de pizza, evidenciando a predominância de certos tipos sobre outros.

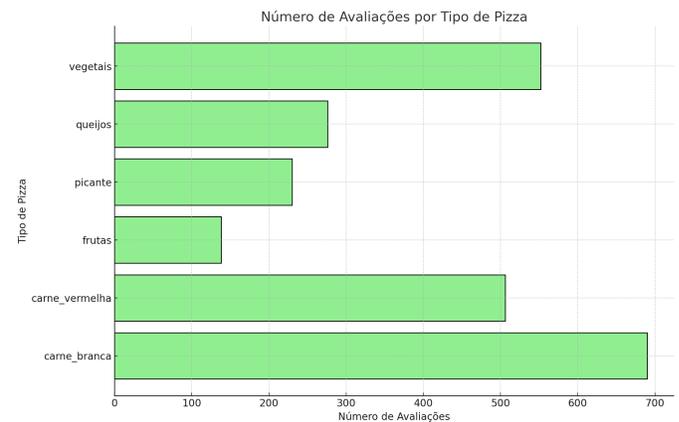


Figura 1. Número de Avaliações por Tipo de Pizza

Além disso, é importante destacar a frequência que cada tipo de pizza está presente no cardápio. Tipos de pizza como Carne Branca, Vegetais e Carne Vermelha são os mais frequentes, conforme ilustrado na Figura 2. Em contraste, os tipos de pizza menos frequentes, como Queijos, Picantes e Frutas, também são apresentados.

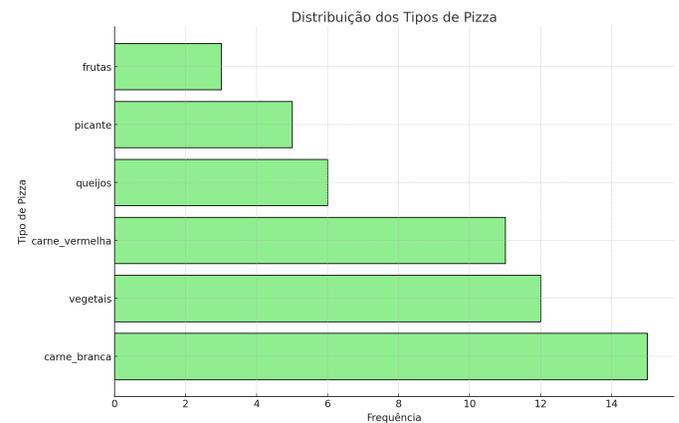


Figura 2. Frequência dos Tipos de Pizza

A média de avaliações por tipo de pizza apresenta valores relativamente similares entre os diferentes tipos, sugerindo que a distribuição das avaliações está equilibrada entre as preferências primárias e secundárias dos usuários. Este equilíbrio

é fundamental para garantir que as avaliações sejam representativas das preferências dos consumidores. A Figura 3 ilustra essa distribuição de forma clara.

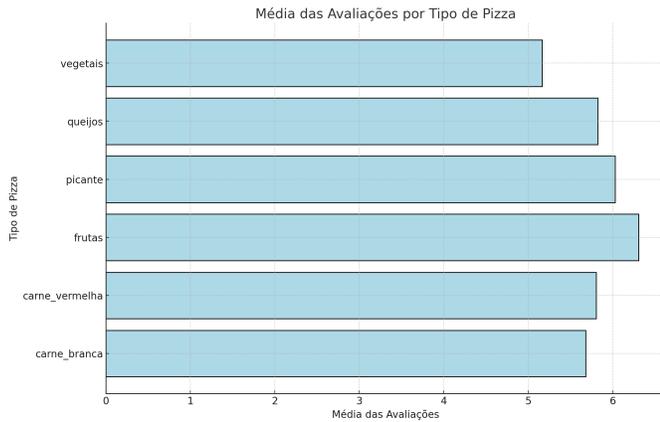


Figura 3. Média das Avaliações por Tipo de Pizza

Adicionalmente, a Figura 4 apresenta a distribuição das médias das avaliações por sabor de pizza. Sabores de pizza com médias de avaliação situadas entre 4 e 6 indicam uma preferência secundária entre os usuários. Por outro lado, sabores de pizza com médias de avaliação mais elevadas correspondem a preferências primárias dos consumidores, evidenciando uma clara distinção entre as preferências de consumo.

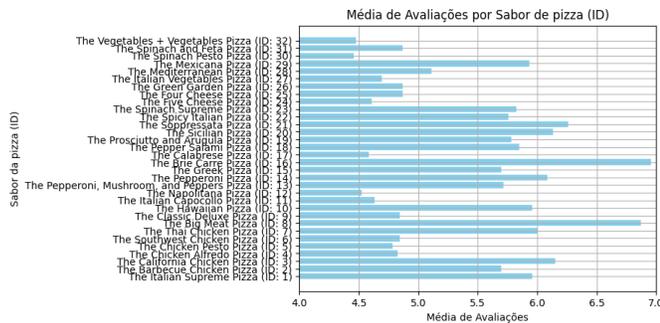


Figura 4. Média das Avaliações por Sabor de Pizza

## B. Sistemas de Recomendação

1) *Filtragem Colaborativa*: A Filtragem Colaborativa é uma abordagem amplamente utilizada em sistemas de recomendação, que utiliza o comportamento passado de um grupo de usuários para prever itens ou serviços que um usuário específico pode gostar. Este método se baseia na ideia de que usuários com preferências semelhantes no passado tendem a ter preferências semelhantes no futuro. Existem duas principais categorias de Filtragem Colaborativa:

- **Filtragem Colaborativa Baseada em Usuários**: Este método recomenda itens para um usuário com base nas preferências de outros usuários semelhantes. Ele identifica usuários que têm históricos de avaliações semelhantes e recomenda itens que esses usuários gostaram e que ainda não foram avaliados pelo usuário-alvo.

- **Filtragem Colaborativa Baseada em Itens**: Em contraste com a abordagem baseada em usuários, este método utiliza as características dos itens para recomendar itens semelhantes aos que um usuário já gostou. Ele identifica itens que são semelhantes aos itens que o usuário já avaliou positivamente e recomenda esses itens como sugestões.

Uma técnica específica dentro da Filtragem Colaborativa é o algoritmo de Singular Value Decomposition (SVD), que é implementado na biblioteca Surprise[Hug20]. O SVD é uma técnica de fatoração de matriz que decompõe a matriz de avaliações dos usuários em três matrizes menores. Isso permite capturar as relações latentes entre usuários e itens, proporcionando recomendações mais precisas.

O SVD funciona da seguinte maneira:

- **Fatoração da Matriz [Sur]**: A matriz de avaliações é decomposta em três matrizes:  $U$  (matriz de usuários),  $\Sigma$  (matriz diagonal de valores singulares) e  $V^T$  (matriz transposta de itens).
- **Redução de Dimensionalidade**: Os valores singulares na matriz  $\Sigma$  representam a importância dos componentes latentes, e a redução de dimensionalidade é aplicada para manter apenas os componentes mais significativos.
- **Reconstrução da Matriz**: As matrizes fatoradas são usadas para reconstruir a matriz original de avaliações, permitindo prever as avaliações ausentes.

Esta técnica é eficaz para lidar com problemas de escalabilidade e esparsidade das matrizes de avaliações, tornando-se uma escolha popular para sistemas de recomendação modernos. Ao utilizar o SVD, é possível capturar a relação complexa entre usuários e itens, melhorando significativamente a precisão das recomendações fornecidas aos usuários finais.

2) *Método de Agregação de Preferências*: Os métodos de agregação de preferência são técnicas utilizadas para combinar múltiplas avaliações ou preferências em uma única métrica representativa. Esses métodos ajudam a resumir as preferências dos usuários, facilitando a análise e a tomada de decisões. A seguir, são descritos alguns dos principais métodos de agregação:

- **Média Aritmética (Mean)** A média aritmética é calculada somando todos os valores e dividindo o total pelo número de valores. É uma medida simples e comum de tendência central, que fornece uma visão geral do conjunto de dados.

$$\text{Mean} = \frac{1}{n} \sum_{i=1}^n x_i$$

- **Mediana (Median)** A mediana é o valor que separa a metade superior da metade inferior de um conjunto de dados ordenado. É uma medida de tendência central que não é influenciada por valores extremos (outliers).

$$\text{Median} = \begin{cases} \frac{x_{(n/2)} + x_{(n/2+1)}}{2} & \text{se } n \text{ é par} \\ x_{((n+1)/2)} & \text{se } n \text{ é ímpar} \end{cases}$$

- **Máximo (Max)** O máximo é o maior valor em um conjunto de dados. É útil para identificar o valor mais extremo em um conjunto de preferências.

$$\text{Max} = \max(x_1, x_2, \dots, x_n)$$

- **Moda (Mode)** A moda é o valor que aparece com mais frequência em um conjunto de dados. Em contextos de preferência, pode indicar a opção mais popular entre os usuários.

$$\text{Mode} = \text{valor mais frequente em } \{x_1, x_2, \dots, x_n\}$$

- **Média Harmônica (Harmonic Mean)** A média harmônica é uma medida de tendência central que é particularmente útil quando os valores são taxas ou razões. É calculada como o número de valores dividido pela soma dos recíprocos dos valores.

$$\text{Harmonic Mean} = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}}$$

3) *Aplicação na Agregação de Preferências:* Para analisar as preferências dos usuários de forma eficaz, utilizamos várias métricas de agregação que nos permitem resumir os dados de avaliações de maneira significativa. Abaixo estão os métodos aplicados:

- **Média Aritmética (Mean):** Calcula a média das avaliações dos usuários, fornecendo uma medida geral das preferências.
- **Mediana (Median):** Determina a mediana das avaliações, oferecendo uma visão das preferências que não é influenciada por outliers.
- **Máximo (Max):** Identifica a avaliação mais alta recebida, destacando a preferência máxima dos usuários.
- **Moda (Mode):** Encontra a avaliação mais frequente, indicando a escolha mais popular entre os usuários.
- **Média Harmônica (Harmonic Mean):** Calcula a média harmônica das avaliações, útil para dados que são taxas ou razões, proporcionando uma perspectiva alternativa da tendência central.

Utilizando esses métodos de agregação, podemos obter uma compreensão mais detalhada e robusta das preferências dos usuários, permitindo a criação de recomendações mais precisas e personalizadas.

4) *Cross-Validation:* Para a avaliação de modelos, um dos métodos mais utilizados é a validação cruzada (*cross-validation*) [HTF09]. Este método divide o conjunto de dados em  $k$  subconjuntos e itera sobre esses subconjuntos, de forma que, a cada iteração, um subconjunto é utilizado como conjunto de teste e o restante como conjunto de treinamento. Ao final desse processo, são calculadas as métricas de Erro Quadrático Médio (*RMSE*) e Erro Absoluto Médio (*MAE*) para cada iteração, permitindo uma avaliação abrangente do desempenho do modelo [Mur12].

5) *Hiperparâmetros:* Os hiperparâmetros são parâmetros do modelo que são definidos antes da etapa de treinamento e não podem ser aprendidos diretamente a partir dos dados de treinamento. É necessário um método externo para sua determinação [BB12]. Exemplos de hiperparâmetros no contexto do algoritmo SVD incluem o número de épocas de treinamento, a taxa de aprendizado e o termo de regularização. O

termo de regularização, em particular, é crucial para equilibrar o sobreajuste (*overfitting*) do modelo [KBV09b].

6) *Grid Search:* Para otimizar um modelo, a definição adequada dos hiperparâmetros é de extrema importância. Uma das técnicas utilizadas para determinar esses hiperparâmetros é a busca em grade (*Grid Search*) [Ped+11]. Esta técnica envolve a aplicação da validação cruzada ao conjunto de dados e, para cada iteração, testar todas as combinações de hiperparâmetros definidas por um intervalo numérico preestabelecido [BB12]. Durante este processo, as métricas de RMSE e MAE são calculadas para cada combinação de hiperparâmetros. Ao final, é identificada a combinação de hiperparâmetros que apresenta as melhores métricas, calculadas a partir da média das métricas obtidas em cada iteração do processo de validação cruzada.

Esses métodos e técnicas são fundamentais para garantir a robustez e a precisão dos modelos de recomendação, permitindo que sejam ajustados e avaliados de maneira sistemática e eficiente [HTF09; Mur12].

### C. Testes

1) *Testes Automatizados:* Para avaliar o desempenho do sistema de recomendação, foram realizados experimentos automáticos variando o número de usuários aleatoriamente e a quantidade de sabores presentes na pizza. Cada experimento envolveu uma combinação distinta de usuários e produtos, totalizando 16 experimentos diferentes. A seguir, detalhamos a configuração dos experimentos:

- Número de Usuários: Variamos o número de usuários de 1 a 4.
- Número de Produtos: Variamos o número de produtos de 1 a 4.

Essa combinação de variáveis resultou nos seguintes cenários experimentais:

- Caso 0 : 1 usuário e 1 produto
- Caso 1 : 1 usuário e 2 produtos
- Caso 2 : 1 usuário e 3 produtos
- Caso 3 : 1 usuário e 4 produtos
- Caso 4 : 2 usuários e 1 produto
- Caso 5 : 2 usuários e 2 produtos
- Caso 6 : 2 usuários e 3 produtos
- Caso 7 : 2 usuários e 4 produtos
- Caso 8 : 3 usuários e 1 produto
- Caso 9 : 3 usuários e 2 produtos
- Caso 10 : 3 usuários e 3 produtos
- Caso 11 : 3 usuários e 4 produtos
- Caso 12 : 4 usuários e 1 produto
- Caso 13 : 4 usuários e 2 produtos
- Caso 14 : 4 usuários e 3 produtos
- Caso 15 : 4 usuários e 4 produtos

Cada uma dessas configurações permitiu testar o sistema de recomendação em diferentes contextos, variando tanto a diversidade de usuários quanto a diversidade de sabores recomendados. Os resultados obtidos em cada experimento foram analisados para entender como o sistema de recomendação se comporta sob diferentes condições, que serão aprofundados na próxima seção.

2) *Testes Manuais*: Além dos 16 experimentos automatizados, foram realizados mais 5 experimentos de maneira manual. Nesses experimentos, os usuários foram escolhidos deliberadamente para investigar o comportamento do sistema de recomendação. A seguir, detalhamos a abordagem desses experimentos:

- **Seleção Deliberada de Usuários**: Os usuários foram selecionados com base em suas preferências primárias e secundárias. Isso permitiu uma análise mais detalhada de como o sistema se comporta.
- **Objetivos Específicos**: Cada experimento manual foi desenhado para avaliar um aspecto particular do sistema, tais como a precisão das recomendações e a diversidade dos itens recomendados.
- **Procedimento Experimental**: Em cada experimento, as interações do sistema foram monitoradas e analisadas detalhadamente. As recomendações fornecidas pelo sistema foram avaliadas quanto à sua relevância e utilidade para os usuários selecionados.

Para os experimentos manuais foi sempre selecionado 4 sabores, a opção mais comum. À seguir detalhes sobre os experimentos:

- **Experimento 1**: Quatro usuários com preferências iguais para verificar se as recomendações funcionam, ou seja, se não é recomendado algo discrepante mesmo que todos os usuários sejam parecidos.

USER ID	PRIMÁRIO	SECUNDÁRIO
1	carne vermelha	carne branca
3	carne vermelha	vegetais
4	carne vermelha	queijos
5	carne vermelha	frutas

Tabela IV  
EXPERIMENTO 1

- **Experimento 2**: Três usuários com preferências iguais e um diferente, para verificar a satisfação do usuário que seria prejudicado pelo grupo.

USER ID	PRIMÁRIO	SECUNDÁRIO
1	carne vermelha	carne branca
4	carne vermelha	queijos
6	carne vermelha	picante
23	vegetais	frutas

Tabela V  
EXPERIMENTO 2

- **Experimento 3**: Quatro usuários com as mesmas preferências secundárias, e nenhuma das primárias iguais, a fim de verificar a dominância da preferência primária sobre a secundária.

USER ID	PRIMÁRIO	SECUNDÁRIO
1	carne vermelha	carne branca
20	vegetais	carne branca
26	queijos	carne branca
38	picante	carne branca

Tabela VI  
EXPERIMENTO 3

- **Experimento 4**: Três usuários com a preferência principal **frutas** que é o tipo de pizza menos presente com apenas

3 itens e 4 sabores, para verificar qual será o sabor extra recomendado.

USER ID	PRIMÁRIO	SECUNDÁRIO
31	frutas	carne vermelha
32	frutas	carne branca
34	frutas	vegetais

Tabela VII  
EXPERIMENTO 4

- **Experimento 5**: Seis usuários sem intersecção para avaliar a satisfação final dos usuários e como os métodos de agregação de preferências se comportam.

USER ID	PRIMÁRIO	SECUNDÁRIO
6	carne vermelha	picante
11	carne branca	frutas
22	vegetais	queijos
28	queijos	vegetais
32	frutas	carne branca
37	picante	carne vermelha

Tabela VIII  
EXPERIMENTO 5

## IV. RESULTADOS

Esta seção descreve o processo acerca dos resultados dos experimentos realizados, as métricas utilizadas para sua avaliação e o momento em que elas foram medidas. Por fim, é feita uma análise detalhada de cada um dos experimentos realizados, destacando os principais achados e *insights* obtidos.

### A. Métricas Utilizadas

Ao avaliar a performance de um sistema de recomendação, diversas métricas podem ser utilizadas para quantificar a qualidade das recomendações geradas. Duas métricas comuns e bem assertivas são o RMSE (Root Mean Square Error) e o MAE (Mean Absolute Error).

1) *RMSE*: O RMSE é uma medida de erro que calcula a raiz quadrada da média dos quadrados dos erros entre os valores previstos e os valores observados. Essa métrica é amplamente utilizada em sistemas de recomendação para avaliar o quão próximo as previsões do sistema estão dos valores reais das avaliações dadas pelos usuários. A fórmula do RMSE é dada por:

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

onde:

- $n$  é o número total de avaliações,
- $y_i$  é o valor real da avaliação do usuário para o item  $i$ ,
- $\hat{y}_i$  é o valor previsto pelo sistema de recomendação para o item  $i$ .

Quanto menor o valor do RMSE, mais preciso é o sistema de recomendação, e a recomendação em si.

2) *MAE*: O MAE é outra métrica utilizada para medir a acurácia de previsões contínuas. Ele calcula a média das diferenças absolutas entre os valores previstos e os valores observados. A fórmula do MAE é definida como:

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

onde:

- $n$  é o número total de avaliações,
- $y_i$  é o valor real da avaliação do usuário para o item  $i$ ,
- $\hat{y}_i$  é o valor previsto pelo sistema de recomendação para o item  $i$ .

O MAE é menos sensível a *outliers* do que o RMSE, pois não considera os quadrados dos erros. Ele fornece uma medida mais direta de quanto o sistema de recomendação está errando em média.

Ambas as métricas, RMSE e MAE, são essenciais para avaliar a precisão e o desempenho de sistemas de recomendação, permitindo ajustes e comparações entre diferentes abordagens e modelos implementados.

## B. Momentos de Avaliação

1) *Avaliação do Modelo*: Foi avaliado o erro do modelo, utilizando a métrica de RMSE e MAE. Esta avaliação nos dá uma perspectiva do erro da representação interna do modelo como um todo. Se este valor for muito grande o modelo não possui uma representação fiel e confiável da preferência dos usuários, ou possui poucas avaliações para gerar as recomendações. Em contrapartida se esse valor for muito pequeno, próximo a 0, significa que ocorreu um *overfit* dos dados do modelo. Para o cálculo foram utilizados os seguintes dados.

$$Y = \bigcup_{i=1}^n \text{aval}(n_i)$$

$$\hat{Y} = \bigcup_{i=1}^n \text{est}(n_i)$$

onde:

- $n$  é o número total de usuários,
- $\text{aval}(x)$  retorna o conjunto de avaliações do usuário  $x$ ,
- $\text{est}(x)$  retorna o conjunto de estimativas do usuário  $x$ .

2) *Avaliação do Usuário*: Foi avaliado o erro do usuário também utilizando as métricas de RMSE e MAE. Este cálculo nos informa o quanto o usuário é decidido das próprias preferências. Um valor alto nessa métrica representa um usuário com várias preferências, ou seja, preferência nenhuma. Para o cálculo desse valor foram utilizado os seguintes dados:

- $u$  é o identificador do usuário,
- $Y = \text{aval}(u)$  retorna o conjunto de avaliações do usuário  $u$ ,
- $\hat{Y} = \text{est}(u)$  retorna o conjunto de estimativas do usuário  $u$ .

3) *Avaliação do Método de Agregação*:

$$\text{scorePizza}(p_i) = \sum_{i=1}^{\text{len}(U)} \sum_{j=0}^{\text{len}(P)} \text{pred}(u_i, p_i)$$

$$\begin{aligned} & \text{aggrScore}(\text{method}) = \\ & \text{sort}(\text{method}(\sum_{i=0}^{\text{len}(\text{scorePizza}()-1)} \text{scorePizza}(i))) \\ & \text{aggrResults}(\text{method}) = \\ & \sum_{m \in M} \sum_{i=0}^{\text{len}(\text{aggrScore}(m))-s} \sum_{u \in U} \text{RMSE}(\text{aval}(u), \text{aggrScore}(m)) \end{aligned}$$

$$\text{finalPizza} = \min_{x \in \text{aggrResults}} \text{aggrResults}(x)$$

onde:

- $s$  é a quantidade de sabores desejado na pizza final.
- $U$  é o conjunto dos usuários do experimento, e  $u_i$  é o usuário;
- $P$  é o conjunto de todas as pizzas,  $p_i$  é a pizza;
- $M$  é o conjunto de todos os métodos de agregação;
- $\text{len}(x)$  retorna o tamanho de um conjunto ou dicionário;
- $\text{scorePizza}(k)$  é um dicionário de lista, cuja chave  $k$  é o identificador da pizza e o valor é uma lista de scores;
- $\text{aggrScore}(k)$  é um dicionário de lista, cuja chave  $k$  é o identificador do método de agregação e o valor é uma lista de scores;
- $\text{aggrResult}(k)$  é um dicionário de lista, cuja chave  $k$  é o identificador do método de agregação e o valor é uma lista de scores;

4) *Avaliação da Satisfação do Usuário*: Foi avaliada a satisfação do usuário utilizando a métrica de RMSE. O cálculo da satisfação é realizado comparando a avaliação real do usuário com a pizza final.

$$Y = \sum_{\text{sabor} \in \text{pizzaFinal}} \text{score}(\text{pizzaFinal}, \text{sabor})$$

$$\hat{Y} = \sum_{p \in \text{aval}(u)} \sum_{\text{sabor} \in \text{pizzaFinal}} p = \text{sabor}$$

onde:

- $u$  é o identificador do usuário,
- $U$  é o conjunto dos usuários do experimento, e  $u_i$  é o usuário,
- $P$  é o conjunto de todas as pizzas,
- $\text{aval}(u)$  retorna o conjunto de avaliações do usuário  $u$ ,
- $\text{score}(p, s)$  retorna o score de um sabor  $s$  de uma pizza  $p$ .

MÉTODO DE AGREGAÇÃO	QUANTIDADE
Média:	8
Mediana:	5
Máximo:	0
Média Harmônica:	2

Tabela IX  
UTILIZAÇÃO DOS MÉTODOS DE AGREGAÇÃO

### C. Resultados Testes Automatizados

O resultado dos experimentos automatizados podem ser visualizados na Tabela X. Os valores em *vermelho* representam a melhor agregação daquele experimento. As agregações podem retornar a mesma combinação de itens, resultando na mesma pontuação da métrica RMSE, vista na Seção IV-A1. Isso fica evidenciado quando o modelo é utilizado para prever a pizza final de apenas um usuário, independentemente do número de sabores a serem utilizados na construção da pizza e pode ser observado nos casos 0, 4, 8 e 12.

Em relação aos métodos de agregação, é possível observar uma dominância do método de agregação por uso da Média Aritmética, ela obtendo o menor erro em 8 dos casos, como pode ser observado na Tabela IX. Seguido pelo método da Mediana, sendo a vencedora em 5 casos e pelo método da Média Harmônica que obteve a melhor pontuação em 2 casos. O método da agregação pelo Máximo não obteve o menor erro em nenhum dos casos. É importante ressaltar que nessa contagem foram excluídos os casos em que havia apenas um usuário.

É possível observar que o quanto mais sabores estão presentes na pizza, maior é a satisfação dos usuários. Isso pode ser visualizado nos casos 3, 7, 11 e 15.

### D. Resultados Testes Manuais

Nesta seção, apresentamos os resultados dos testes manuais realizados para avaliar a qualidade e a performance das pizzas disponíveis no cardápio. Os testes foram conduzidos com um conjunto diversificado de usuários simulados (personas), cada um com preferências específicas para tipos de pizza, tanto primárias quanto secundárias. Essas personas foram desenvolvidas para representar uma ampla gama de gostos e preferências, proporcionando uma visão abrangente sobre a aceitabilidade e a satisfação dos diferentes sabores de pizza.

1) *Experimento 1*: No experimento 1 queríamos observar o funcionamento correto do sistema forçando vários usuários com preferência iguais a fim de ver se o sistema não retornaria uma recomendação fora dos padrões. Para esse experimento foram utilizados vários usuários com a preferência primária *Carne Vermelha*. Conforme observado na Tabela XI todos os sabores recomendados possuem tal tipo demonstrando o funcionamento correto do sistema.

Vale frisar que este foi o único experimento em qual o método de agregação vencedor foi o método da Moda, devido aos vários usuários possuírem avaliações nota 10 para os itens.

PIZZA FINAL			
Método Agregação:	Moda	Score:	0.559
SABORES			
Identificador:	Nome:	Tipos:	Score:
17	The Calabrese Pizza	carne vermelha	9.540
1	The Italian Supreme Pizza	carne vermelha, carne branca	10.000
8	The Big Meat Pizza	carne branca, carne vermelha, picante	10.000
20	The Sicilian Pizza	carne vermelha, carne branca	10.000
SATISFAÇÃO DOS USUÁRIOS			
Identificador:	Score:		
1	0.550		
3	0.568		
4	0.568		
5	0.550		

Tabela XI  
RESULTADO EXPERIMENTO 1

2) *Experimento 2*: Neste experimento gostaríamos de observar a satisfação de um usuário que está em um grupo do qual as preferências divergem das suas. Ao observarmos a satisfação do usuário número 23 nos resultados desse experimento evidentes na Tabela XII constatamos que sua satisfação não destoou muito da dos demais ficando inclusive mais satisfeito que o usuário número 1. Essa satisfação regular se deve ao fato de a preferências primária do usuário 23, vegetais, terem sido satisfeitas pelos sabores 23 - *The Spinach Supreme Pizza* e 15 - *The Greek Pizza*.

PIZZA FINAL			
Método Agregação:	Média Harmônica	Score:	1.704
SABORES			
Identificador:	Nome:	Tipos:	Score:
17	The Calabrese Pizza	carne vermelha	7.322
20	The Sicilian Pizza	carne vermelha, carne branca	7.347
23	The Spinach Supreme Pizza	carne vermelha, vegetais	8.787
15	The Greek Pizza	carne vermelha, vegetais	9.019
SATISFAÇÃO DOS USUÁRIOS			
Identificador:	Score:		
1	1.950		
4	1.280		
6	1.686		
23	1.826		

Tabela XII  
RESULTADO EXPERIMENTO 2

3) *Experimento 3*: Com o experimento 3 pretendemos verificar a dominância da preferência primária sobre a secundária. Conforme evidenciado na Tabela XIII, todas as pizzas recomendadas possuem o tipo *Carne Branca*, tipo escolhido como preferência secundária dos usuários do experimento, evidenciando que o sistema não prioriza as preferências primárias em relação às secundárias. Também é possível destacar que a pontuação de satisfação de todos os usuários foi ótima, nenhuma passando de 2.

PIZZA FINAL			
Método Agregação:	Media	Score:	0.920
SABORES			
Identificador:	Nome:	Tipos:	Score:
16	The Brie Carne Pizza	queijos, carne branca, frutas	7.645
1	The Italian Supreme Pizza	carne vermelha, carne branca	7.646
6	The Southwest Chicken Pizza	carne branca	7.660
8	The Big Meat Pizza	carne branca, carne vermelha, picante	8.278
SATISFAÇÃO DOS USUÁRIOS			
Identificador:	Score:		
1	1.255		
20	0.762		
26	1.004		
38	0.471		

Tabela XIII  
RESULTADO EXPERIMENTO 3

4) *Experimento 4*: No experimento 4 queríamos observar qual seria o item extra recomendado pelo sistema de recomendação, já que o cardápio só possui apenas 3 pratos do tipo **frutas**, e foi solicitado uma pizza com 4 sabores. Como

EXPERIMENTOS			USUÁRIOS								MÉTODO AGREGAÇÃO				RESULT.	SATISFAÇÃO			
id	sabores	n_users	u1		u2		u3		u4		RMSE				pizza	RMSE			
			id	rmse	id	rmse	id	rmse	id	rmse	mean	median	max	h_mean		id	u1_sats	u2_sats	u3_sats
0	1	1	37	0.719							0.0	0.0	0.0	0.0	13	0.0			
1	1	2	42	0.698	11	0.679					0.738	0.738	6.001	0.759	22	0.042	1.042		
2	1	3	22	0.695	39	0.804	15	0.823			1.081	0.885	4.760	1.179	29	1.343	0.343	0.657	
3	1	4	34	0.680	3	0.665	36	0.817	2	0.674	3.113	3.340	4.123	2.675	18	2.926	3.074	1.074	3.074
4	2	1	11	0.679							0.534	0.534	0.534	0.534	22,6	0.534			
5	2	2	6	0.661	28	0.495					0.895	0.895	5.641	0.899	21,14	0.876	0.913		
6	2	3	29	0.875	11	0.679	3	0.665			2.318	3.055	5.339	2.581	14,21	2.221	2.912	1.647	
7	2	4	36	0.817	33	0.735	37	0.719	12	0.845	2.854	3.014	3.446	2.866	10,16	1.322	2.322	4.678	1.884
8	3	1	15	0.823							0.148	0.148	0.148	0.148	23,30,12	0.148			
9	3	2	24	0.705	3	0.665					0.787	0.787	1.413	0.772	28,23,15	0.965	0.511		
10	3	3	37	0.719	41	0.761	12	0.845			0.889	0.848	1.167	0.929	13,8,22	0.822	0.233	1.194	
11	3	4	1	0.758	26	0.687	3	0.665	8	0.667	1.107	1.763	3.041	1.164	8,20,1	1.086	1.384	1.086	0.790
12	4	1	35	0.910							0.746	0.746	0.746	0.746	24,10,7,16	0.746			
13	4	2	33	0.735	5	0.822					1.838	1.838	3.740	1.911	18,7,10,16	2.048	1.600		
14	4	3	40	0.676	30	0.654	8	0.667			2.265	2.457	4.060	2.318	3,16,8,22	2.843	1.974	1.846	
15	4	4	19	0.755	31	0.695	7	0.777	25	0.597	1.200	2.151	2.208	1.251	15,8,14	1.297	0.495	1.205	1.543

Tabela X

RESULTADO DOS TESTES AUTOMATIZADOS.

podemos observar na Tabela XIV foi recomendado o item *The Sicilian Pizza*. A pizza final foi obtida através do método de agregação por média harmônica com pontuação de 0.991. Vale ressaltar também que os 3 itens do tipo **frutas** estão presentes, demonstrando entendimento por parte do algoritmo. Interessante também observar que a pizza extra é dos tipos presentes na preferência secundária de 2 dos 3 usuários, como visto na Tabela VII.

PIZZA FINAL			
Método Agregação:	Média Harmônica	Score:	0.991
SABORES			
Identificador:	Nome:	Tipos:	Score:
20	The Sicilian Pizza	carne vermelha, carne branca	5.748
16	The Brie Carre Pizza	frutas, carne branca, queijos	8.826
10	The Hawaiian Pizza	frutas, carne branca	9.040
7	The Thai Chicken Pizza	frutas, carne branca	9.199
SATISFAÇÃO DOS USUÁRIOS			
Identificador:	Score:		
31	0.888		
32	1.414		
34	0.397		

Tabela XIV

RESULTADO EXPERIMENTO 4

5) *Experimento 5*: Neste experimento pretendíamos avaliar como o sistema se comporta tendo como objeto um conjunto grande de usuários distintos. Podemos observar que sob essas condições o sistema tem dificuldades de agradar os usuários, evidenciado pelas suas satisfações, nenhuma sendo abaixo da pontuação 2. Também é possível observar uma predominância de sabores do tipo **Carne Branca**, possivelmente devido ao tipo **Carne Branca** ser o tipo mais comum no cardápio, como é observado na Figura 2 na Seção III-A4.

PIZZA FINAL			
Metodo Agregação:	Media Harmonica	Score:	2.473
SABORES			
Identificador:	Nome:	Tipos:	Score:
20	The Sicilian Pizza	carne_vermelha, carne_branca	5.967
29	The Mexicana Pizza	picante, vegetais	5.972
19	The Prosciutto and Arugula Pizza	carne_branca, vegetais	6.000
3	The California Chicken Pizza	carne_branca, queijos	6.175
SATISFAÇÃO DOS USUARIOS			
Identificador:	Score:		
6	2.271		
11	2.604		
22	2.728		
28	2.267		
32	2.253		
37	2.663		

Tabela XV

RESULTADO EXPERIMENTO 5

## V. CONCLUSÃO

Neste trabalho, foi desenvolvido um modelo baseado em filtragem colaborativa. Foram realizados diversos experimentos para avaliar a eficácia do modelo.

### A. Contribuições

O trabalho oferece uma abordagem automatizada e eficiente para resolver problemas de recomendação comuns no cotidiano, como a sugestão de sabores de pizza com base em avaliações de usuários. Ao utilizar a filtragem colaborativa, o modelo consegue capturar as preferências dos usuários de maneira eficaz, promovendo uma experiência de consumo mais personalizada. Além disso, o estudo contribui para a literatura existente ao demonstrar a viabilidade e a eficácia da aplicação de modelos de recomendação em contextos práticos.

### B. Limitações

Apesar dos resultados promissores, o estudo apresenta algumas limitações. A principal limitação está relacionada à disponibilidade dos dados, que pode restringir a generalização dos resultados. Além disso, o modelo SVD utilizado não incorpora informações de *timestamp*, o que poderia melhorar a precisão das recomendações ao considerar a evolução das preferências dos usuários ao longo do tempo.

### C. Trabalhos futuros

Para aprimorar o modelo e ampliar sua aplicabilidade, os seguintes passos estão planejados para trabalhos futuros:

- Implementação de Restrições Adicionais: A inclusão de restrições pode ajudar a refinar ainda mais as recomendações, tornando-as mais relevantes para os usuários.
- Desenvolvimento de um Modelo Baseado em Conteúdo: Complementar a filtragem colaborativa com um modelo baseado em conteúdo permitirá combinar as vantagens de ambos os métodos, resultando em recomendações mais robustas.
- Realização de Testes com Dados Reais: Testar o modelo com conjuntos de dados reais fornecerá uma validação

mais rigorosa de sua eficácia e ajudará a identificar possíveis áreas de melhoria.

Essas futuras direções de pesquisa têm o potencial de fortalecer o modelo de recomendação e aumentar sua utilidade prática em diversos contextos de aplicação.

#### REFERÊNCIAS

- [RM87] David E. RUMELHART e James L. MCCLELLAND. “Learning Internal Representations by Error Propagation”. Em: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*. 1987, pp. 318–362.
- [OCO+01] Mark O’CONNOR et al. “PolyLens: A Recommender System for Groups of Users”. Em: *ECSCW 2001: Proceedings of the Seventh European Conference on Computer Supported Cooperative Work 16–20 September 2001, Bonn, Germany*. Ed. por Wolfgang Prinz et al. Dordrecht: Springer Netherlands, 2001, pp. 199–218. ISBN: 978-0-306-48019-5. DOI: 10.1007/0-306-48019-0\_11. URL: [https://doi.org/10.1007/0-306-48019-0\\_11](https://doi.org/10.1007/0-306-48019-0_11).
- [MSM08] V. N. MARIVATE, G. SSALI e T. MARWALA. “An intelligent Multi-Agent recommender system for human capacity building”. Em: *MELECON 2008 - The 14th IEEE Mediterranean Electrotechnical Conference*. IEEE, mai. de 2008. DOI: 10.1109/melcon.2008.4618553. URL: <http://dx.doi.org/10.1109/MELCON.2008.4618553>.
- [HTF09] Trevor Hastie, Robert Tibshirani e Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.
- [KBV09a] Yehuda KOREN, Robert BELL e Chris VOLINSKY. “Matrix Factorization Techniques for Recommender Systems”. Em: *Computer* 42.8 (2009), pp. 30–37. DOI: 10.1109/MC.2009.263.
- [KBV09b] Yehuda Koren, Robert Bell e Chris Volinsky. “Matrix Factorization Techniques for Recommender Systems”. Em: *Computer* 42.8 (2009), pp. 30–37. DOI: 10.1109/MC.2009.263.
- [Ped+11] Fabian Pedregosa et al. “Scikit-learn: Machine Learning in Python”. Em: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- [BB12] James Bergstra e Yoshua Bengio. “Random Search for Hyper-Parameter Optimization”. Em: *Journal of Machine Learning Research* 13 (2012), pp. 281–305.
- [Mur12] Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.
- [Hug20] Nicolas Hug. “Surprise: A Python library for recommender systems”. Em: *Journal of Open Source Software* 5.52 (2020), p. 2174. DOI: 10.21105/joss.02174. URL: <https://doi.org/10.21105/joss.02174>.
- [Moh23] Neethi Mohan. *Maven Pizza Challenge Dataset*. <https://www.kaggle.com/datasets/neethimohan/maven-pizza-challenge-dataset>. Accessed: 2024-07-01. 2023.
- [Aks] Andrew Aksyonoff. *Sphinx — Open Source Search Engine*. URL: <https://sphinxsearch.com/>.
- [Dat] DatasetSearch. *Dataset Search*. <https://datasetsearch.research.google.com/>. Accessed: Jun 2024.
- [gau] gauravduttakiit. *Pizza Toppings Classification*. <https://www.kaggle.com/datasets/gauravduttakiit/pizza-topplings-classification>. Accessed: Jun 2024.
- [Gro] GroupLens. *GroupLens*. URL: <https://grouplens.org/>.
- [Kag] Kaggle. *Kaggle: Your Machine Learning and Data Science Community*. <https://www.kaggle.com/>. Accessed: Jun 2024.
- [kni] knightbarr. *Pizza Price Prediction (Playground Data)*. <https://www.kaggle.com/datasets/knightbarr/pizza-price-prediction>. Accessed: Jun 2024.
- [Mov] MovieLens. *MovieLens*. URL: <https://movielens.org/>.
- [ope] openML. *OpenML*. <https://www.openml.org/>. Accessed: Jun 2024.
- [Pot] Fabien Potencier. *Symfony, High Performance PHP Framework for Web Development*. URL: <https://symfony.com/what-is-symfony>.
- [sdh] sdhilip. *Principal Component Analysis - Pizza Dataset - dataset by sdhilip — data.world*. <https://data.world/sdhilip/pizza-datasets>. Accessed: Jun 2024.
- [Sur] Surprise. *Matrix Factorization — Surprise 1.1.3 documentation*. [https://surprise.readthedocs.io/en/stable/matrix\\_factorization.html](https://surprise.readthedocs.io/en/stable/matrix_factorization.html). Accessed: Jun 2024.
- [Szi] Peter Szinek. *GitHub - scrubber/scrubyt: A simple to learn and use, yet powerful web scraping toolkit!* URL: <https://github.com/scrubber/scrubyt>.