

**GABRIEL GUIMARÃES MASSOLARI
RAFAEL VARGAS MESQUITA DOS SANTOS**

IMPLEMENTAÇÃO DE UM ALGORITMO DE EQUIDADE PARA SISTEMAS DE RECOMENDAÇÃO



**IMPLEMENTAÇÃO DE UM ALGORITMO DE EQUIDADE
PARA SISTEMAS DE RECOMENDAÇÃO**



Gabriel Guimarães Massolari
Rafael Vargas Mesquita dos Santos

**IMPLEMENTAÇÃO DE UM ALGORITMO DE EQUIDADE
PARA SISTEMAS DE RECOMENDAÇÃO**

1ª Edição

Quipá Editora
2024

Copyright © 2024 dos autores. Todos os direitos reservados.

Esta obra é publicada em acesso aberto. O conteúdo dos capítulos, os dados apresentados, bem como a revisão ortográfica e gramatical são de responsabilidade de seu autor, detentor de todos os Direitos Autorais, que permite o download e o compartilhamento, com a devida atribuição de crédito, mas sem que seja possível alterar a obra, de nenhuma forma, ou utilizá-la para fins comerciais.

Os autores gostariam de expressar sua profunda gratidão ao Instituto Federal do Espírito Santo (IFES) pelo aporte financeiro e apoio contínuo ao longo do desenvolvimento deste trabalho. Este suporte foi fundamental para a realização da pesquisa e os resultados alcançados.

Dados Internacionais de Catalogação na Publicação (CIP)

G963i Guimarães Massolari, Gabriel.

Implementação de um algoritmo de equidade para sistemas de recomendação / Gabriel Guimarães Massolari e Rafael Vargas Mesquita dos Santos. – Iguatu, CE : Quipá Editora, 2024.

49 p. : il.

ISBN 978-65-5376-310-4

1. Programação de Computadores - Software. 2. Sistemas de recomendação . I. Título.

CDD 005

Obra publicada pela Quipá Editora em março de 2024.

Quipá Editora
www.quipaeditora.com.br
@quipaeditora

AGRADECIMENTOS

Agradeço primeiramente a Deus, por iluminar meu caminho e me fortalecer nos momentos de dúvida e incerteza, sendo uma fonte constante de esperança e fé em toda a minha jornada.

À minha família, meu eterno agradecimento pelo apoio incondicional, amor e compreensão. Vocês foram fundamentais em cada etapa deste processo, oferecendo-me a base sólida sobre a qual construí meus sonhos e aspirações.

Agradeço imensamente à minha namorada, Lais Fabiani, por todo apoio durante esses anos de relacionamento. Ela que foi peça fundamental para o meu desenvolvimento pessoal e acadêmico. Obrigado por toda a paciência, carinho e apoio. Sua presença ao meu lado foi fundamental para que eu chegasse até aqui.

Um agradecimento especial ao meu orientador, Rafael Vargas, pela dedicação, paciência e sabedoria compartilhadas. Sua orientação foi essencial para o desenvolvimento deste trabalho, proporcionando-me uma aprendizagem valiosa e um crescimento profissional significativo.

E por último, agradeço a todos os professores que participaram da minha jornada de ensino dentro do Instituto Federal do Espírito Santo. Vocês foram muito importantes para o meu processo de aprendizagem e desenvolvimento profissional.

RESUMO

Os algoritmos dos Sistemas de Recomendação utilizam dados históricos para realizar recomendações para os usuários, porém podem apresentar resultados tendenciosos para determinados grupos devido a desequilíbrios na entrada de dados do sistema, gerando em alguns casos comportamentos antiéticos como o racismo, sendo de suma importância encontrar meios de resolver essa problemática.

Esse trabalho realiza a análise e a implementação de um algoritmo de equidade aplicado a um webservice utilizando métricas de justiça para avaliar e promover a equidade entre os grupos de usuário. O conjunto de dados utilizado para teste foi uma base de dados referente a avaliações de 10 músicas diferentes e para avaliação, foram selecionados 59 avaliadores atribuindo notas em uma escala de 0 a 5.

A validação do algoritmo foi feita utilizando duas estratégias de agrupamento, sendo elas o agrupamento 50-50, que divide os usuários em dois grupos iguais, e o agrupamento 95-5, que divide os 5% dos usuários que mais avaliaram em um grupo e os 95% restantes que menos avaliaram em outro grupo.

O teste foi aplicado as estratégias de recomendação do tipo Filtragem Colaborativa, Filtragem Baseada em Conteúdo e no algoritmo Híbrida Cascata. Em todos os agrupamentos e algoritmos foi possível diminuir a injustiça do grupo e aumentar a eficiência da recomendação. E o melhor resultado na diminuição da injustiça do grupo foi encontrado no agrupamento 50-50 utilizando o algoritmo de Filtragem Baseada em Conteúdo, que foi possível obter uma redução de 99,96% na injustiça do grupo, além de um ganho de eficiência de 56,96%.

Palavras-chave: Sistemas de recomendação, algoritmo de equidade, webservice, métricas de justiça, equidade, recomendação, Filtragem Baseada em Conteúdo, Filtragem Colaborativa, Híbrida Cascata.

LISTA DE FIGURAS

Figura 1 - Diagrama de implementação do algoritmo de equidade no serviço web	15
Figura 2 - Ciclo de vida da recomendação	18
Figura 3 - Representação gráfica de uma função objetivo	26
Figura 4 - Arquitetura do Sistema	30
Figura 5 - Exemplo algoritmo de Equidade	31
Figura 6 - Representação de funcionamento do algoritmo de equidade	37
Figura 7 - Representação da redução da injustiça no agrupamento 50-50	43
Figura 8 - Representação da redução da injustiça no agrupamento 95-5	44

LISTA DE TABELAS

Tabela 1 – Resultados da Injustiça do grupo R_{grp} e do Erro Quadrático Médio $RMSE$ para o agrupamento 50-50 aplicado ao algoritmo de Filtragem Colaborativa	40
Tabela 2 – Resultados da Injustiça do grupo R_{grp} e do Erro Quadrático Médio $RMSE$ para o agrupamento 50-50 aplicado ao algoritmo de Filtragem Baseada em Conteúdo	40
Tabela 3 – Resultados da Injustiça do grupo R_{grp} e do Erro Quadrático Médio $RMSE$ para o agrupamento 50-50 aplicado ao algoritmo de Cascata Híbrida	40
Tabela 4 – Resultados da Injustiça do grupo R_{grp} e do Erro Quadrático Médio $RMSE$ para o agrupamento 95-5 aplicado ao algoritmo de Filtragem Colaborativa	41
Tabela 5 – Resultados da Injustiça do grupo R_{grp} e do Erro Quadrático Médio $RMSE$ para o agrupamento 95-5 aplicado ao algoritmo de Filtragem Baseada em Conteúdo	42
Tabela 6 – Resultados da Injustiça do grupo R_{grp} e do Erro Quadrático Médio $RMSE$ para o agrupamento 95-5 aplicado ao algoritmo de Cascata Híbrida	42

LISTA DE ABREVIATURAS

RMSE(Root Mean-Square Error) - Erro Quadrático Médio

MAE(Mean Absoluto Error) - Erro Absoluto Médio

WSL - Web Service License JVM - Java Virtual Machine

API - Application Programming Interface

HTTP - Hypertext Transfer Protocol

SUMÁRIO

AGRADECIMENTOS

RESUMO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

CAPÍTULO 1 **10**

INTRODUÇÃO

CAPÍTULO 2 **13**

REVISÃO BIBLIOGRÁFICA

CAPÍTULO 3 **26**

METODOLOGIA

CAPÍTULO 4 **35**

RESULTADOS

CAPÍTULO 5

CONFIGURAÇÕES

APÊNDICE A **44**

CÓDIGO DE OTIMIZAÇÃO

REFERÊNCIAS **48**

CAPÍTULO 1

INTRODUÇÃO

Os sistemas de recomendação são importantes ferramentas para fornecer sugestões personalizadas para cada usuário na internet, isso devido a grande quantidade de informações que circulam nessa era da informática, auxiliando assim em escolhas mais efetivas para as pessoas. Atualmente, essa ferramenta está fortemente presente em diversos tipos de serviços, como redes sociais (Facebook), lojas virtuais (Amazon), plataformas de vídeos (YouTube, TikTok) e outros serviços que utilizam desse conceito para melhor atender seus usuários (CHEN et al., 2020).

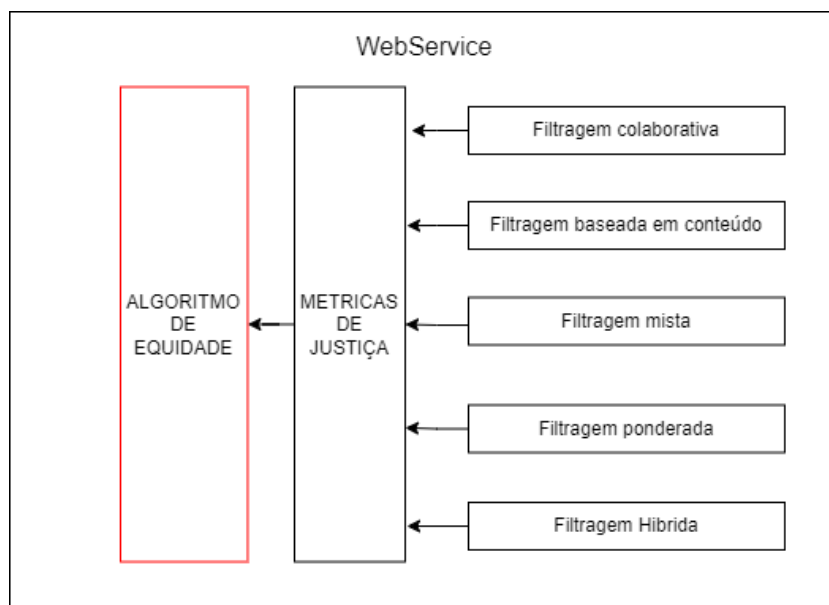
Entretanto, por se tratar de um sistema que aprende de acordo com padrões em dados históricos de entrada, as recomendações podem apresentar comportamentos tendenciosos e injustos para determinados grupos de pessoas. E segundo GÓMEZ, Boratto e SALAMÓ (2022), caso esse comportamento esteja vinculado a características sensíveis como sexo ou etnia, pode ser considerado um resultado discriminante. Um exemplo que demonstra a problemática dessa injustiça é um sistema de recomendação utilizado em uma plataforma de cursos abertos, na qual foi analisada uma inconsistência na recomendação de cursos de professores de certas regiões geográficas, prejudicando assim, os alunos que não tinham acesso a cursos de outras localidades e os professores que tinham a divulgação de seus cursos afetada (GÓMEZ et al., 2022).

Dessa forma, a importância de abordar a injustiça em sistemas de recomendação é crucial, considerando a influência significativa que esses sistemas têm na tomada de decisões e na formação de opiniões dos usuários. Nesta perspectiva, sistemas que recomendam conteúdo de notícias podem inadvertidamente reforçar bolhas de filtro, limitando a exposição dos usuários a uma variedade de perspectivas e opiniões, e potencialmente polarizando ainda mais a sociedade, e essas recomendações injustas podem afetar um determinado grupo de pessoas e a experiência do usuário, prejudicando na utilização do sistema. Portanto, a adoção de uma equidade entre diferentes perfis de

identidade é um importante tema para ser desenvolvido e estudado nessa área (BORATTO et al., 2022)(GÓMEZ; BORATTO; SALAMÓ, 2022).

Neste contexto, este trabalho apresenta a análise e a implementação de um algoritmo de equidade disponibilizado em um serviço web de sistema de recomendação utilizando-se de métricas de justiça para avaliar a justiça do grupo e eficiência do algoritmo. Este poderá ser aplicado a todas as estratégias de recomendação presentes na Figura 1, para assim alcançar a justiça entre os usuários.

Figura 1 - Diagrama de implementação do algoritmo de equidade no serviço web



Fonte: Autor

OBJETIVOS

Objetivo Geral

Esse trabalho possui como objetivo geral elaborar e desenvolver um algoritmo de equidade que utiliza como base métricas de justiça para fornecer um resultado mais justo para os usuários.

Objetivos Específicos

Os objetivos específicos deste trabalho são definidos como:

- Elaborar um algoritmo de equidade utilizando um otimizador matemático para calcular o resultado mais justo;
- Incluir esse algoritmo em um webservice;
- Aplicar e testar o algoritmo de equidade em um estudo de caso aplicado a um contexto de recomendação de músicas, avaliando a justiça do grupo e a eficiência da recomendação.

CAPÍTULO 2

REVISÃO BIBLIOGRÁFICA

No capítulo anterior, foi realizada uma breve introdução ao trabalho. Nesta seção, serão apresentados os conceitos essenciais que fundamentam o desenvolvimento do trabalho, visando proporcionar uma compreensão mais aprofundada.

SISTEMAS DE RECOMENDAÇÃO

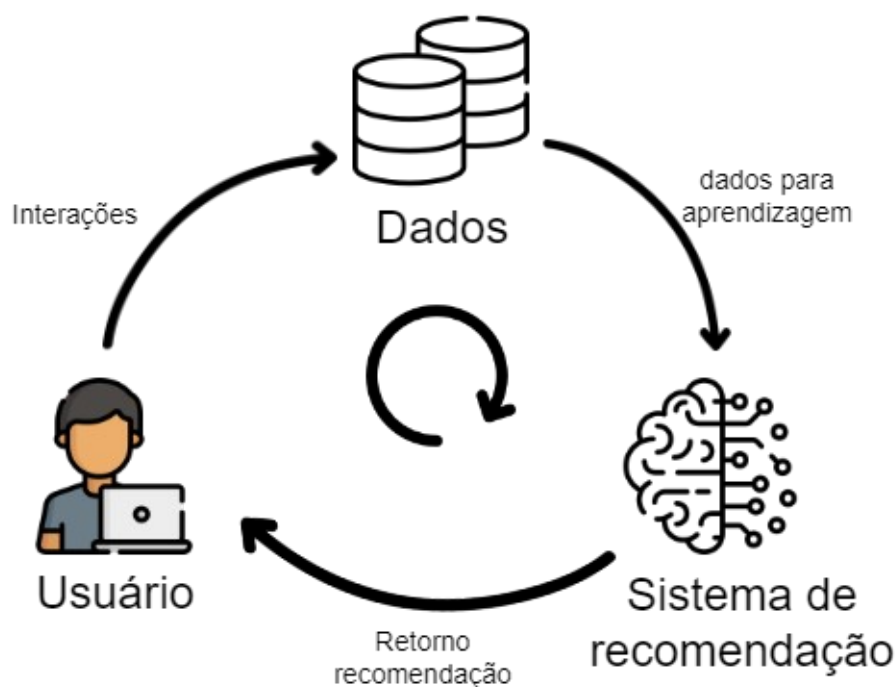
Sistemas de recomendação são definidos como uma estratégia que estima e recomenda qualquer tipo de item para um determinado usuário de acordo com suas preferências, auxiliando em sua tomada de decisão (ZHANG et al., 2019). Atualmente, essa estratégia se torna indispensável para os usuários da internet, devido a alta quantidade de informações online. Algumas de suas aplicações no cotidiano são: a sugestão de filmes, produtos e vídeos em diversas plataformas digitais como Netflix, Amazon e Youtube (AFOUDI; LAZAAR; ACHHAB, 2021).

Esse sistema funciona a partir da entrada de dados dos usuários para prever possíveis gostos e preferências das pessoas, sendo que um dos principais dados de entrada do sistema é a avaliação da pessoa perante a algum item recomendado, o que ajuda a determinar o quão certas estão sendo as recomendações e também a melhorar as sugestões para esse usuário. Sendo assim, um sistema de recomendação possui como principal objetivo entregar a um usuário uma lista de objetos (filmes, livros e etc) que se encaixe com seu perfil para auxiliá-lo a tomar uma decisão na plataforma/sistema que ele esteja utilizando (Lü et al., 2012).

O ciclo de vida de uma recomendação, de acordo com (CHEN et al., 2020), é um loop com três componentes, sendo eles: usuário, dados e modelo. Este ciclo pode ser visualizado na Figura 2. Os três estágios desse loop são:

- **Usuário → Dados:** Fase de coleta dos dados de interação do usuário, como clicks em produtos, avaliações e etc.
- **Dados → Modelo:** Aprendizado do modelo de recomendação baseado nos dados de interação coletados do usuário, além de estimar itens potencialmente interessantes para essa pessoa.
- **Modelo → Usuário:** Retorna a recomendação para o usuário de acordo com suas necessidades, auxiliando suas decisões futuras.

Figura 2 - Ciclo de vida da recomendação



Fonte: Autor

De acordo com Filho et al. (2017), os sistemas de recomendação podem ser divididos em três categorias, sendo elas: filtragem colaborativa, filtragem baseada em conteúdo e híbridos, sendo que cada categoria é aplicada em um contexto específico em cada aplicação.

FILTRAGEM COLABORATIVA

Os algoritmos de filtragem colaborativa são bastante populares, e são responsáveis por gerar recomendações de itens que são conhecidos e relevantes por usuários similares ao perfil da pessoa.

Esse algoritmo ignora os atributos dos itens e costuma se basear no conceito de vizinhança, na qual é definido uma quantidade K de vizinhos e com base no vizinhos mais similares ao usuário, são realizadas as previsões e recomendações (AGUIAR; ARAÚJO; COSTA, 2020).

FILTRAGEM BASEADA EM CONTEÚDO

A filtragem baseada em conteúdo é um modelo que elabora perfis para os usuários e para os itens utilizando como base os atributos dos itens. Esse algoritmo recomenda para o usuário os itens com perfis mais similares aos dele, utilizando como base os itens que o usuário demonstrou interesse no passado.

Sendo assim, para implementação deste modelo é preciso definir quais serão os atributos utilizados para construção dos perfis de usuários e itens. A partir dessa definição, é possível determinar os pesos de cada um desses atributos, para que no final seja realizada a análise de similaridade entre os perfis (FILHO et al., 2017).

FILTRAGEM HÍBRIDA

A Filtragem híbrida combina a Filtragem Colaborativa e a Filtragem Baseada em Conteúdo com o objetivo de extrair as melhores características de cada um. Sendo assim, a hibridização ajuda a resolver problemas específicos de cada um dos modelos anteriores. Filho et al. (2017) também diz que há algumas categorias de sistemas híbridos como:

- **Baseada em pesos:** O resultado da recomendação é obtido de acordo com a pontuação de cada técnica de recomendação;
- **Alternado:** O algoritmo alterna entre filtragem baseada em conteúdo e filtragem colaborativa de acordo com algum critério específico;
- **Misto:** O algoritmo combina o resultado de duas técnicas de recomendação.

Um exemplo de utilização do modelo híbrido de recomendação é a empresa Netflix, que utiliza a Filtragem Colaborativa para comparar as pesquisas e visualizações entre usuários similares, e a Filtragem Baseada em Conteúdo para filtrar filmes que receberam boas avaliações pelo usuário alvo. Assim, de acordo com o resultado desses dois algoritmos, é feita a recomendação de filmes para o usuário (SHOKEEN; RANA, 2020).

Métricas de avaliação

Conforme dito anteriormente, o sistema de recomendação recomenda objetos que potencialmente podem interessar a um usuário. Para avaliar o nível de acerto das recomendações, geralmente os dados são divididos em duas partes: dados de treino e dados de teste, na qual são geradas recomendações na base de treino e os resultados são comparados com os dados de testes que já possuem quais são os interesses de cada usuário. Para avaliar a acurácia das recomendações são utilizadas diversas métricas matemáticas, sendo algumas delas (Lü et al., 2012):

- **Erro Absoluto Médio e Raiz Quadrada do erro médio:** São usados para medir a proximidade das classificações previstas com as classificações verdadeiras, sendo que quanto maior a taxa de erro das recomendações, maiores são os valores dessas métricas.

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

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

- **Correlação de Pearson:** Define se há uma relação linear entre os conjuntos de recomendação avaliados.

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (3)$$

- **Correlação de Spearman:** É semelhante à Correlação de Pearson porém, é avaliado o ranking de recomendação.

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad (4)$$

- **Coefficiente de Kendall:** Mede até que ponto os rankings se equivalem em seus valores exatos.

$$\tau = \frac{2}{n(n-1)} \sum_{i < j} \text{sgn}(x_i - x_j) \cdot \text{sgn}(y_i - y_j) \quad (5)$$

MEDIDAS DE JUSTIÇA

Os sistemas de recomendação encontram padrões comportamentais entre as pessoas para sugerir novos itens para os usuários. Essas recomendações são importantes e auxiliam cada vez mais as decisões diárias da população em geral, haja a vista a grande quantidade de informações que se tem atualmente. Entretanto, podem apresentar um comportamento tendencioso baseados em estereótipos relacionados a diversas características, como gênero ou etnia.

E mesmo que o sistema não leve em consideração dados demográficos dos usuários, ele pode acabar recomendando melhor para um grupo muito específico de usuários. Um exemplo de um algoritmo tendencioso seria a recomendação mais frequente de melhores empregos, com salários mais altos, para homens em comparação às mulheres. Pode-se visualizar que tal algoritmo assume uma postura discriminatória baseada no gênero do usuário em questão, o que prejudica nos aspectos sociais e financeiros dessas mulheres, que acabam não tendo acesso a esses empregos que são divulgados (ZEHLIKE et al., 2022).

Com isso, está acontecendo uma mudança na avaliação das recomendações, na qual além dos métodos citados no **tópico 2.1.4**, também estão sendo adotadas medidas de justiça para avaliar o quão justo estão sendo as recomendações para os indivíduos e para determinados grupos de usuários com características em comum, como pessoas de determinada etnia ou país.

Sendo assim, é muito importante que, além de melhorar a acurácia geral do sistema, também analisar todas as medidas de justiça para avaliar se atributos sensíveis estão sendo considerados no resultado da sugestão ao usuário (DELDJOO; BELLOGIN; NOIA, 2021).

Justiça Individual

A justiça individual estipula que cada usuário com características semelhantes seja tratado de maneira igual, mesmo quando há um erro na predição. Sendo assim, mesmo que a recomendação seja errada para dois indivíduos com perfis semelhantes, caso o sistema erre em uma maior proporção para um usuário e em uma menor proporção para outro, há uma injustiça individual entre eles. Portanto, quanto maior essa desigualdade entre recomendações, menor a justiça individual (LI et al., 2021).

Um exemplo de injustiça individual seria as recomendações de cursos em plataformas de ensino digital, que acaba favorecendo certas pessoas em detrimento as outras, visto que as sugestões de cursos acabam sendo mais assertivas para certas

peças. Nesse cenário, é de extrema importância que seja garantido a igualdade entre os alunos de acordo com as recomendações educacionais, visto que a educação é um meio de mudar significativamente a vida de um indivíduo em relação a oportunidades no mercado de trabalho. Dessa forma, essas oportunidades não podem ser prejudicadas devido a decisões tendenciosas por parte dos sistemas de recomendação educacionais que deve zelar pela equidade de recomendações entre seus usuários para garantir que todos tenham a mesma oportunidade de desenvolvimento educacional e profissional (MARRAS et al., 2022).

Sendo assim, é crucial compreender e medir a injustiça individual, que se refere às diferenças nas experiências negativas que os usuários enfrentam devido ao sistema de recomendação. Ao calcular esse tipo de injustiça, podemos avaliar o nível de equidade entre os usuários do sistema. O objetivo é minimizar essas experiências negativas individuais para promover uma maior equidade entre todos os usuários. Ao fazer isso, buscamos melhorar a justiça e a igualdade no uso do sistema de recomendação.

Justiça do Grupo

A justiça do grupo exige que os grupos devem ser tratados da mesma forma que o grupo privilegiado, sendo o grupo privilegiado o que possui um maior número de interação com o determinado sistema. Assim, as recomendações para esse grupo devem ser justas em comparação aos grupos protegidos (desfavorecidos). Isso implica na definição de uma restrição a taxa de falsos positivos e a taxa de verdadeiros positivos devem ser iguais para o grupo protegido e o grupo favorecido, o que representa o princípio da igualdade de oportunidades (LI et al., 2021).

Um exemplo de injustiça de grupo é uma determinada loja online que possui um maior acesso de pessoas do gênero masculino, na qual esses homens acabam recebendo melhores recomendações em comparação às mulheres, por possuírem uma maior interação no sistema. Neste caso, cria-se uma injustiça de grupo baseada no gênero dos usuários.

E assim como na justiça individual, a injustiça do grupo será a variação de todas as

perdas do grupo. Com esse cálculo é possível medir qual o nível de justiça do grupo do sistema de recomendação. Para melhorar essa justiça, busca-se minimizar a injustiça do grupo para que seja possível alcançar uma maior equidade entre os grupos.

OTIMIZAÇÃO

A otimização é uma área matemática que busca obter a melhor solução dentro de um conjunto de possibilidades em cenários que são matematicamente complexos. O objetivo é conseguir encontrar a melhor resposta possível dentro de um conjunto de regras e limitações buscando uma determinada condição como menor custo e etc. Algumas áreas que possuem esse cenário e que se beneficiam da otimização para a resolução de problemas são: engenharia, economia e diversas áreas científicas que buscam estudar e alcançar o melhor resultado para determinado contexto econômico e científico (KANG et al., 2023).

Devido a isso, esse tema é amplamente útil para sociedade em geral para resolução de problemas de classificação, seleção de recursos e otimização em geral. Na lista abaixo, estão alguns métodos de otimização:

- **Programação Linear:** é um método de otimização que consiste em encontrar a melhor solução de um problema matematicamente linear (POCHMANN, 2022);
- **Algoritmos Genéticos:** são métodos de busca baseados na evolução natural, inspirados na teoria evolutiva, em que é criada uma população inicial e nela são aplicadas as interações genéticas, como o cruzamento e a mutação da população. Com isso, é possível alcançar uma população ideal para resolução de determinada problemática (GRUBISIC et al., 2012);
- **Algoritmos de Enxame de Partículas:** são uma classe de algoritmos, que assim como os algoritmos genéticos, também criam uma população inicial. Cada indivíduo representa uma solução possível, e elas se movem pelo espaço de soluções em busca da melhor solução (GRUBISIC et al., 2012);

- **Métodos de gradiente conjugado:** são um conjunto de técnicas usadas para resolver grandes equações lineares de forma iterativa, além de também ser possível se adaptar para otimização de problemas não lineares (NOCEDAL; WRIGHT, 1999).

Um exemplo prático em que foi utilizado o conceito de otimização para ajudar a sociedade foi um caso em que foi criado um modelo de otimização para auxiliar a localizar hospitais de campanha no Rio de Janeiro durante o período de pandemia. Para tal modelo, foi levado em conta alguns recursos hospitalares como leitos de UTI, leitos de enfermaria e respiradores, base para o modelo estatístico de previsão do número de infectados. Essa otimização buscou entender a evolução da doença no Rio de Janeiro, projetar os números de casos, estimar a demanda de internação e a necessidade de uso da Ventilação Mecânica Invasiva (VMI) e dimensionar a quantidade de recursos a serem alocados por região (HAMACHER, 2020).

Restrições

As restrições são condições em que a solução final de otimização deve se enquadrar, sendo geralmente descritas por meio de equações e inequações lineares. Por exemplo, em um sistema de recomendação que tem como objetivo recomendar dietas alimentares para as pessoas, a restrição seria quais calorias e nutrientes são necessários para atender as quantidades diárias ideais, devendo respeitar os valores mínimos e máximos de cada componente (POCHMANN, 2022).

Variáveis de Decisão

As variáveis de decisão dentro do contexto de otimização são representações dos re- cursos importantes para resolução do problema a ser resolvido. Ou seja, é a abstração dos itens importantes para a otimização em forma matemática. Um exemplo no mesmo contexto de recomendação de dietas citadas anteriormente, são quais alimentos devem ser consumidos, sendo que cada alimento tem um determinado custo associado e uma determinada tabela nutricional (POCHMANN, 2022).

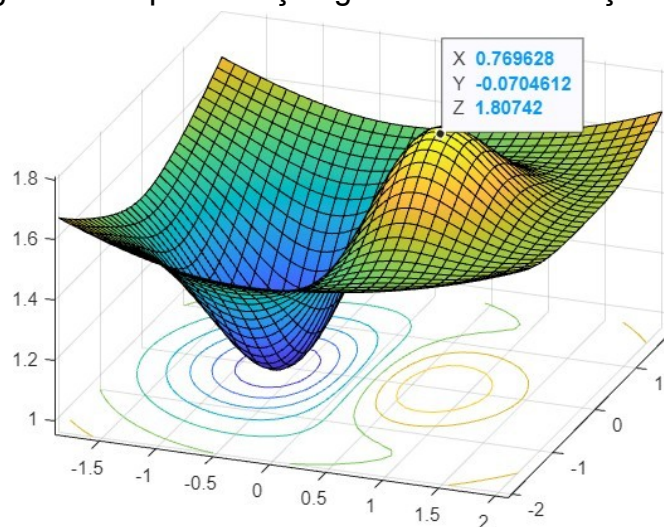
Função Objetivo

A função objetivo é uma função matemática que permite representar a problemática real a ser resolvida em uma fórmula matemática para posteriormente ser processada a fim de encontrar a melhor solução. Vale ressaltar que essa função deve considerar todas as variáveis e pesos possíveis que representem o mais fiel possível a realidade do problema, visto que qualquer erro na função objetivo irá resultar em uma otimização final errada.

E como no exemplo citado no tópico anterior, caso uma função objetivo para representar um modelo de previsão para auxiliar alocações em hospitais e prever número de casos de Covid-19, seja calculada e pensada de forma equivocada, sem considerar todas as variáveis importantes, resulta em um grande impacto geral para população que será alocada de forma inadequada, podendo até gerar vítimas fatais por falha de gerenciamento dos recursos. (HOLGUÍN-VERAS et al., 2013) (HAMACHER, 2020)

De acordo com a Figura 3, que contém uma função objetivo qualquer como exemplo, pode-se visualizar que a função possui uma grande variação de resultados de acordo com os valores de entrada das variáveis. Dessa maneira, é papel do processo de otimização encontrar a solução ideal de acordo com os requisitos do problema e como será resolvido (ELSABAGH; FARHAN; GAFAR, 2021).

Figura 3 - Representação gráfica de uma função objetivo



Fonte: Autor

Otimizadores

Os otimizadores, ou *solvers*, são softwares responsáveis por resolver diretamente os problemas relacionados a otimização matemática, utilizando como base a função objetiva, explicada no tópico anterior, para alcançar os resultados desejados. Essas ferramentas utilizam diversas técnicas de cálculos para encontrar a solução ideal, como a Programação Quadrática Sequencial, que é utilizado por exemplo para otimização de problemáticas não lineares restritas. (VANROYE et al., 2023)

Existem várias classificações de otimizadores matemáticos, que podem ser categorizados de acordo com diferentes critérios. Segundo Zhou, Zhou e Sundmacher (2017) essas são algumas das principais classificações:

- **Otimizadores Determinísticos:** São algoritmos que buscam encontrar a melhor solução de um problema de otimização por meio de um número determinado de iterações. Eles são determinísticos pois, sempre alcançam o mesmo resultado final dado as mesmas condições iniciais;
- **Otimizadores Estocásticos:** Os otimizadores estocásticos usam princípios probabilísticos para encontrar soluções aproximadas ou sub-ótimas. Eles podem ser úteis quando a função objetivo ou as restrições envolvem incertezas ou quando a busca pela solução ótima exige tempo computacionalmente intensivo;
- **Otimizadores Híbridos:** Esses otimizadores buscam encontrar a melhor solução dentro de uma região local do espaço de busca. Geralmente utilizam-se de técnicas de ambas os métodos anteriores.

É importante ressaltar que essas classificações são apenas algumas das formas de categorizar os otimizadores matemáticos, e existem muitos outros critérios e abordagens utilizados na área de otimização. A escolha do otimizador mais adequado depende das características específicas do problema em questão.

Otimizador Gurobi

Um exemplo de *solver* bastante utilizado no mercado e até mesmo no meio acadêmico é o **Gurobi**, que é o software escolhido para ser utilizado nesse trabalho. O Gurobi Optimizer é uma ferramenta de otimização matemática sofisticada, projetada para resolver desafios complexos de negócios. Ele se destaca por sua capacidade de transformar modelos matemáticos complexos em planos de ação detalhados, o que é essencial para tomada de decisões eficazes em diversas aplicações (Gurobi Optimization, LLC, 2021).

Com um ótimo desempenho, o Gurobi é uma das soluções mais rápidas, se comparadas a outros solucionadores em encontrar soluções viáveis e ótimas, especialmente para problemas grandes e complexos. Ele continua aprimorando seu desempenho ao longo do tempo, com melhorias significativas em diferentes tipos de problemas, demonstrando estabilidade numérica e precisão em mais de 10.000 modelos de indústria e academia. Além disso, esse otimizador oferece suporte a uma ampla gama de tipos de problemas, incluindo programação linear, programação linear inteira mista e programação quadrática, tanto convexa quanto não convexa, entre outros (Gurobi Optimization, LLC, 2021).

Além dessas características, o Gurobi também oferece flexibilidade de licenciamento, permitindo que os modelos sejam implantados de várias maneiras, seja localmente, em nuvens internas ou públicas, e com a possibilidade de transferir licenças entre desenvolvedores e usuários finais. Acompanhando as tendências tecnológicas atuais, o Gurobi também oferece um WLS, que permite a implantação quase em qualquer lugar, incluindo ambientes de nuvem e contêineres, sem a necessidade de instalar ferramentas de licenciamento nos clientes (Gurobi Optimization, LLC, 2021).

Em termos de integração, o Gurobi fornece interfaces de programação leves e modernas, facilitando a integração em aplicações existentes. Estas interfaces são compatíveis com várias linguagens de programação, como C++, Java, .NET, Python, C, MATLAB e R, e oferecem ligações para linguagens de modelagem padrão e para o Excel através dos

Frontline Solvers (Gurobi Optimization, LLC, 2021).

Em resumo, o Gurobi Optimizer é uma solução robusta, eficiente e flexível para otimização matemática, oferecendo um bom desempenho, suporte a uma ampla gama de tipos de problemas, suporte especializado, licenciamento flexível e interfaces de programação intuitivas. Ele é uma escolha ideal para organizações que procuram otimizar suas operações e tomar decisões baseadas em dados com precisão e eficiência.

CAPÍTULO 3

METODOLOGIA

No capítulo anterior, foram apresentados os aspectos fundamentais para a compreensão do trabalho atual. Visto isso, juntamente com a teoria necessária, abordada na fundamentação teórica, este capítulo discute a metodologia proposta para guiar o desenvolvimento do projeto.

ARQUITETURA

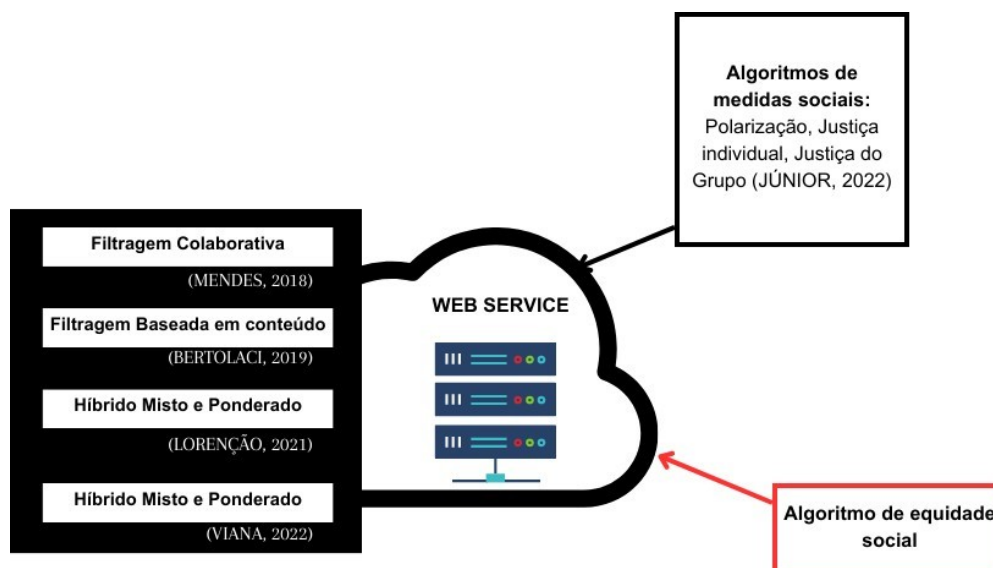
O algoritmo de equidade social para sistemas de recomendação proposto neste trabalho foi implementado em um web service baseado no protocolo HTTP. Neste ambiente estão implementados 5 diferentes algoritmos de recomendação, sendo eles:

1. Sistema de recomendação com Filtragem Colaborativa (MENDES; SANTOS; PICOLI, 2018);
2. Sistema de recomendação com Filtragem Baseado em Conteúdo (BERTOLACI, 2019);
3. Sistema de recomendação com Filtragem Híbrida Ponderada (LORENÇÃO, 2021);
4. Sistema de recomendação com Filtragem Híbrida Mista (LORENÇÃO, 2021);
5. Sistema de recomendação Híbrido Baseado em Combinação Sequencial (VIANA, 2022);

Além desses métodos de recomendação, o serviço web contém a implementação de cálculos de medidas sociais que será utilizado como base para implementação do

algoritmo de equidade, conforme pode ser visualizado na Figura 4.

Figura 4 - Arquitetura do Sistema



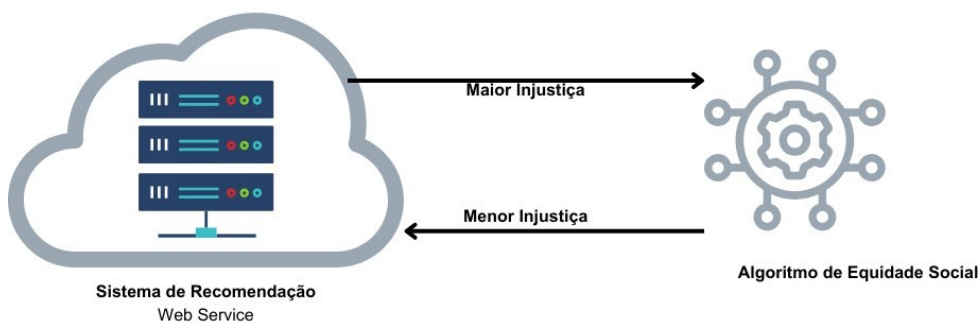
Fonte: Autor

Exemplo de Utilização do Algoritmo de Equidade Social

Esse projeto foi planejado para que recebesse diversos dados de entrada de usuários e grupos de usuários, sendo que esses dados podem apresentar um certo nível de injustiça. Sendo assim, seria aplicado o algoritmo de equidade social para que as recomendações geradas pelo servidor fosse justa entre os grupos/usuários, conforme pode ser visualizado na Figura 5.

Figura 5 - Exemplo algoritmo de Equidade

USUÁRIO	INJUSTIÇA - I	PERDA DO GRUPO - L	INJUSTIÇA DO GRUPO - RGRP
1	3	2,5	0,9506
2	2		
3	0,3	0,55	
5	0,8		



USUÁRIO	INJUSTIÇA - I	PERDA DO GRUPO - L	INJUSTIÇA DO GRUPO - RGRP
1	1,2	1	0,0625
2	0,8		
3	0,4	0,5	
5	0,6		

Fonte: Autor

Ferramentas

Durante a execução do projeto, foram utilizadas as seguintes tecnologias:

- Java: Linguagem de programação multiplataforma, versão OpenJDK (18.0.2.1);
- IntelliJ IDEA: Software de desenvolvimento para desenvolver software de computador escrito em Java, e outras linguagens baseadas em JVM;
- PostgreSQL: Banco de dados relacional para armazenamento dos dados;
- Postman: Software que auxilia no desenvolvimento de APIs, necessário para efetuar testes nas rotas de envio de requisições HTTP;

- H2: Banco de dados em memória, usado na etapa de testes;
- GitHub: plataforma de hospedagem de código-fonte e arquivos com controle de versão . Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou Open Source;
- Gurobi: Otimizador utilizado para realizar as operações matemáticas de otimização;
- Máquina de testes: Os testes serão realizados em uma máquina cuja as configurações são: Windows 10 - 64 Bits, 16 GB de memória ram (DDR4-3000 Mhz), processador amd Ryzen 5 2600 3.4 GHz, ssd 240 GB sandisk;

BASE DE DADOS

O estudo de caso utilizou um conjunto de dados construído por meio de um experimento no contexto musical. Foram registradas cerca de 10 músicas (com diferentes estilos, línguas, cantadas em solo ou banda, cantadas por homem e mulheres, etc.). Para avaliação, foram selecionados 59 avaliadores atribuindo notas em uma escala de 0 a 5;

Em ambos os estudos, foram realizados, após as predições calculadas pelo algoritmo de recomendação, dois tipos de agrupamentos dos usuários: análise de agrupamento hierárquico e análise de agrupamento 95-5. Neste último, consideramos o número de avaliações realizadas pelos usuários.

Por fim, foram apresentados os resultados de medidas de justiça do grupo e de precisão das recomendações. Estas medidas foram comparadas considerando a matriz estimada calculada por um sistema de recomendação tradicional \hat{X} com a matriz estimada calculada pelo algoritmo de equidade \hat{X}_n .

MÓDULO 1 DO ALGORITMO: CÁLCULO DE MEDIDAS SOCIAIS

Tendo em vista todas as especificações e discussões da seção anterior, iremos definir formalmente as métricas que especificam as funções objetivo associadas à justiça individual e justiça do grupo.

Começaremos apresentando a configuração do sistema, notação e a definição do problema. Suponha que $X \in \mathbb{R}^{n \times d}$ seja uma matriz de classificação parcialmente observada de n usuários e d itens, de modo que o elemento x_{ij} denota a classificação dada pelo usuário i para o item j . Seja Ω o conjunto de índices de classificações desconhecidas em X . Além disso, Ω_i denota os índices de classificações de itens desconhecidos para o usuário i , e Ω_j denota os índices de classificações de usuários desconhecidas para o item j .

Para uma matriz A , $P_\Omega(A)$ é uma matriz cujos elementos em $(i, j) \in \Omega$ são a_{ij} e zero nestas posições. Da mesma forma, para um vetor a , $P_{\Omega_j}(a)$ é um vetor cujos elementos em $i \in \Omega_j$ são os elementos correspondentes de a e zero nestas posições. Ao longo do trabalho, denotamos a coluna j de A pelo vetor a_j e a linha i de A pelo vetor a_i . Todos os vetores são vetores de coluna.

Dado um sistema de recomendação tradicional é gerada uma matriz estimada de recomendações $\hat{X} = [\hat{X}_{ij}]_{n \times m}$. Neste problema de recomendação supomos usuários em um conjunto $\{u_1, u_2, \dots, u_n\}$ e itens em um conjunto $\{v_1, v_2, \dots, v_m\}$.

A matriz X_C é uma matriz que possuem as avaliações dos usuários referentes às recomendações fornecidas pelo sistema. Dessa forma, a partir do que foi recomendando e a avaliação do usuário, é possível determinar o nível de justiça individual e o nível de justiça do grupo.

Justiça individual: Para cada usuário i , definimos ℓ_i , a perda de usuário i , como a estimativa do erro quadrático médio sobre as classificações conhecidas do usuário i :

$$\ell_i = \frac{\|P_{\Omega_i}(\hat{x}^i - x_C^i)\|_2^2}{|\Omega_i|} \quad (6)$$

Em seguida, definiremos a injustiça individual como a variação das perdas do usuário:

$$R_{indv}(X_C, \hat{X}) = \frac{1}{n^2} \sum_{k=1}^n \sum_{l>k}^n (l_k - l_l)^2 \quad (7)$$

Para melhorar a justiça individual, buscaremos minimizar a R_{indv} .

Justiça do grupo: Sendo I o conjunto de todos os usuários/itens e $G = \{G_1, G_2, \dots, G_g\}$ seja uma partição de usuários/itens dentro de g grupos, ou seja, $I = \bigcup_{i \in \{1, 2, \dots, g\}} G_i$. Definiremos a perda do grupo como a estimativa do erro quadrático médio sobre todas as classificações conhecidas no grupo i :

$$L_i = \frac{\|P_{\Omega_{G_i}}(\hat{X} - X_C)\|^2}{|\Omega_{G_i}|} \quad (8)$$

Para determinada partição G , a injustiça do grupo será a variação de todas as perdas do grupo:

$$R = \frac{1}{g} \sum_{k=1}^g (X_{\mu_{G_k}} - X_C)^2 = (L) \quad (9)$$

Novamente, para melhorar a justiça do grupo, procuraremos minimizar R_{grp} .

MÓDULO 2 DO ALGORITMO: ALGORITMO DE EQUIDADE SOCIAL

Neste módulo, será apresentada a estrutura de funcionamento do sistema web para que ele seja capaz de gerar recomendações justas, utilizando como base um método de reclassificação com restrições de justiça.

Para novos valores das células x^i de cada uma das matrizes estimadas \hat{X}_p consideram uma estratégia de perturbação baseada na variância das diferenças de avaliações vs recomendações. Sendo assim, é considerada como variância máxima o valor de 16 $(5 - 1)^2$, uma vez que a maior diferença entre um valor avaliado e recomendado seria 4. Um exemplo em que pode ocorrer essa maior diferença é quando o sistema recomenda uma nota 5 para determinado item, porém, o valor real avaliado foi 1. Por isso usa-se a normalização do valor aleatório de recomendação dividindo a injustiça individual l_i por quatro.

O algoritmo de reclassificação é implementado para escolher n linhas $\{v_1, v_2, \dots, v_m | u_i\}$, gerando uma única matriz estimada \hat{X}_n .

Para cada matriz estimada são calculadas n perdas individuais (l_i), referentes aos n usuários. Dessa forma, para cada matriz estimada \hat{X}_p , onde $\{1 \leq p \leq k\}$, teremos uma lista de n perdas individuais $\{l_1, l_2, \dots, l_n | \hat{X}_p\}$.

A matriz de perdas individuais é definida como $Z = [Z_{ij}]_{n \times k}$ e serve para denotar as n perdas individuais calculadas para cada uma das k matrizes estimadas \hat{X}_p , onde $Z_{ij} \in \{R_+\}$, e $\{1 \leq i \leq n\}$, e $\{1 \leq j \leq k\}$, indexa usuários e matrizes estimadas, respectivamente.

A matriz binária é definida como $W = [W_{ij}]_{n \times k}$ para acusar se a perda individual j será considerada para um usuário i na geração da matriz estimada final \hat{X}_n , onde $W_{ij} \in \{0, 1\}$, $\{1 \leq i \leq n\}$, e $\{1 \leq j \leq k\}$ indexa usuários e perdas individuais, respectivamente. Especificamente, se a perda individual j é considerada para o usuário i , então temos $W_{ij} = 1$, senão $W_{ij} = 0$.

A justiça do grupo requer que os grupos protegidos sejam tratados de forma semelhante ao grupo privilegiado. O grupo de usuários pode ser dividido em diferentes requisitos para diferentes tarefas. Nesta proposta serão considerados dois grupos, sendo um simples agrupamento 50-50, na qual foi escolhido os usuários com identificadores de número ímpar em um grupo e usuários com identificadores pares em outro, e o agrupamento 95-5, que divide os usuários a partir de seus níveis de atividade. Em ambos os agrupamentos, serão considerados dois grupos G_1 e G_2 , de forma que $G_1 \cap G_2 = \emptyset$.

Dado um sistema de recomendação tradicional, o qual gera uma matriz estimada de recomendações \hat{X} , cada usuário u_i receberá um conjunto de recomendações $\{v_1, v_2, \dots, v_m|u_i\}$, as quais, na perspectiva de cálculo de justiça, representarão uma perda individual de ℓ_i . Considerando assim as outras k matrizes estimadas, teremos então $\{\ell_1, \ell_2, \dots, \ell_n|\hat{X}_p\}$.

A cada par usuário / matriz estimada é associada a uma pontuação Z_{ij} que representa a perda individual do usuário i nos termos da matriz estimada j . Aqui, aplicaremos o algoritmo de reclassificação para minimizar a soma das pontuações de perdas individuais sob a restrição de justiça. Portanto, podemos formular o procedimento de otimização do problema de recomendação ciente da imparcialidade da seguinte forma:

$$R_{grp}(\hat{X}, X_C, G) = \frac{1}{g} \sum_{k=1}^g (L_k - \mu)^2 \quad (10)$$

$$R_{grp}^{min} = \frac{1}{g} \sum_{k=1}^g (L_k - \mu)^2 \quad (11)$$

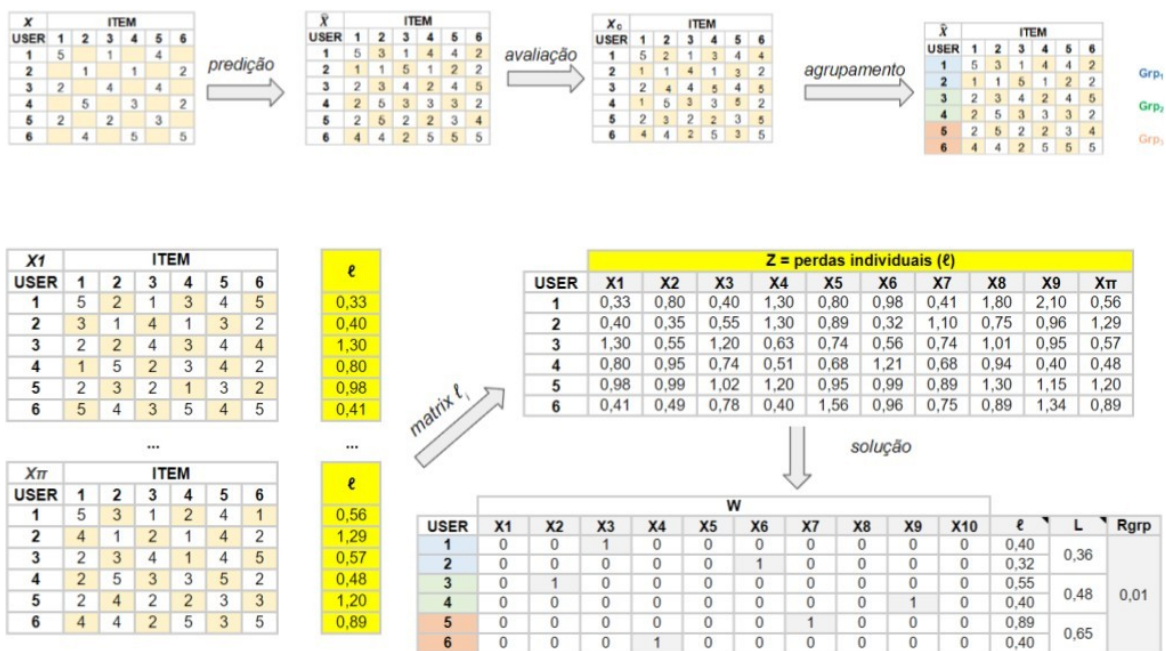
Onde:

$$\ell_i = \sum_{j=1}^n \sum_{k=1}^k W_{ij} Z_{ij} \quad (12)$$

$$L_k = \frac{1}{|\Omega_{G_i}|} \sum_{i=1}^{|\Omega_{G_i}|} \ell_{G_i} \quad (13)$$

Na Figura 6 é possível visualizar de forma simplificada o diagrama do processo de funcionamento do algoritmo de equidade descrito anteriormente:

Figura 6 - Representação de funcionamento do algoritmo de equidade



Fonte: Autor

CAPÍTULO 4

RESULTADOS

Neste capítulo, serão mostrados o desempenho e os resultados alcançados pelo algoritmo de justiça no sistema de recomendação, realizando uma comparação com uma recomendação tradicional e uma recomendação tratada pelo algoritmo.

CODIFICAÇÃO

O código feito para realização deste trabalho está disponível no GitHub¹, e abaixo estão as principais classes elaboradas/alteradas para o funcionamento do algoritmo:

- `RecommendationRatingRepository`: Classe responsável por buscar/filtrar os dados de Avaliação de Recomendações no banco de dados;
- `RecommendationRepository`: Classe responsável por buscar/filtrar os dados de Recomendações no banco de dados;
- `ItemRatingRepository`: Classe responsável por buscar/filtrar os dados de Avaliações dos Itens no banco de dados;
- `RecommendationRepository`: Classe responsável por buscar/filtrar os dados de Recomendações no banco de dados;
- `FairnessRecommendationService`: Classe principal do algoritmo, onde está presente toda a lógica de geração das matrizes, modelagem matemática e retorno da nova matriz de recomendação;

- RgrpController: Classe controladora que recebe a requisição, captura o resultado de FairnessRecommendationService e retorna como resposta;

¹

Link Github: <<https://github.com/gabrielmassolari/srh-backend>>

CAPÍTULO 5

CONFIGURAÇÕES

O agrupamento dos usuários para realização dos testes, considerou duas configurações de grupo:

- **Agrupamento 50-50:** Usuários agrupados em dois grupos, sendo os dois grupos contendo 50% dos usuários igualmente. Para este caso foi considerado os usuários com identificadores de número ímpar em um grupo e usuários com identificadores pares em outro;
- **Agrupamento 95-5:** Usuários agrupados em dois grupos, sendo um grupo contendo 95% dos usuários com menos avaliações e um outro grupo contendo 5% dos usuários com mais avaliações. O grupo com 95% seria desfavorecido visto que representa pessoas com um menor número de avaliações de itens recomendados, enquanto o outro grupo são as pessoas com um maior número de avaliações dentro do sistema. Neste estudo de caso, o grupo de 5% possui 3 usuários e o outro grupo com 95%, possui 56 usuários.

PARÂMETROS

Número de matrizes estimadas calculadas (k): Para a definição do melhor quantitativo de matrizes estimadas a serem calculadas pelo algoritmo de equidade, foram testados cinco possíveis valores de k , a saber: 10, 25, 50, 75 e 100. Em cada um dos testes para k foram realizadas 10 repetições para obtenção da média dos valores resultantes.

Os resultados dos experimentos envolvendo o agrupamento 50-50 estão presentes nas tabelas 1, 2 e 3, e referentes ao agrupamento 95-5 estão presentes nas tabelas 4, 5 e 6. Todos esses resultados contém as seguintes informações:

- k : quantidade de matrizes estimadas calculadas;

- Média: cálculo da média dentre as 10 repetições de execução do algoritmo de equidade;
- Desvio: cálculo do desvio padrão dentre as 10 repetições de execução do algoritmo de equidade;
- (%): percentual de redução ou aumento comparando a média original e a média resultante da execução do algoritmo de equidade.

Tabela 1 - Resultados da Injustiça do grupo R_{grp} e do Erro Quadrático Médio $RMSE$ para o agrupamento 50-50 aplicado ao algoritmo de Filtragem Colaborativa

Estratégia 50-50 para agrupamento dos usuários $\{G_1$ e $G_2\}$ Algoritmo Filtragem Colaborativa Injustiça do grupo (R_{grp}) Média original 5,00393 Eficiência ($RMSE$) Média original 1,27538						
k	$R_{grp}(\mu)$	$R_{grp}(\sigma)$	(%)	$RMSE(\mu)$	$RMSE(\sigma)$	(%)
10	3,53696	1,71284	-29,31	1,01635	0,000027	-20,31
25	3,468908	0,78914	-30,67	0,9899	0,000008	-22,38
50	3,21427	0,11692	-35,76	0,9729	0,00001	-23,71
75	3,2357	0,16397	-35,33	0,9670	0,000007	-24,17
100	3,0948	0,2316	-38,15	0,9652	0,000007	-24,31

Tabela 2 - Resultados da Injustiça do grupo R_{grp} e do Erro Quadrático Médio $RMSE$ para o agrupamento 50-50 aplicado ao algoritmo de Filtragem Baseada em Conteúdo

Estratégia 50-50 para agrupamento dos usuários $\{G_1$ e $G_2\}$ Algoritmo Filtragem Baseada em Conteúdo Injustiça do grupo (R_{grp}) Média original 3,498525 Eficiência ($RMSE$) Média original 2,09386						
k	$R_{grp}(\mu)$	$R_{grp}(\sigma)$	(%)	$RMSE(\mu)$	$RMSE(\sigma)$	(%)
10	0,2007642	0,3969465236	-94,26	1,2243195	0,0009146	-41,52
25	0,00159	0,0000001	-99,95	1,149321	0,0001393	-45,10
50	0,00127	0,00000003	-99,96	1,11909	0,0000632	-46,55
75	0,00122	0,00000002	-99,96	1,110028	0,0000141	-46,98
100	0,001215	0,00000001	-99,96	0,900989	0,179566	-56,96

Tabela 3 - Resultados da Injustiça do grupo R_{grp} e do Erro Quadrático Médio $RMSE$ para o agrupamento 50-50 aplicado ao algoritmo de Cascata Híbrida

Estratégia 50-50 para agrupamento dos usuários $\{G_1$ e $G_2\}$ Algoritmo Cascata Híbrida Injustiça do grupo (R_{grp}) Média original 5,53471 Eficiência ($RMSE$) Média original 1,26593						
k	$R_{grp}(\mu)$	$R_{grp}(\sigma)$	(%)	$RMSE(\mu)$	$RMSE(\sigma)$	(%)
10	4,216007	0,6352425	-23,82	1,022295	0,0008878	-19,24
25	4,029765	0,4517498	-27,19	0,98563	0,0000151	-22,14
50	3,620529	0,1117453	-34,58	0,967855	0,00001735	-23,54
75	3,4074	0,14707870	-38,43	0,96515	0,00001165	-23,75
100	3,458246	0,13203911	-37,51	0,959566	0,00000365	-24,20

Considerando os dados da Tabela 1, Tabela 2 e Tabela 3, referentes ao agrupamento 50-50, podemos observar que à medida que aumentamos o valor de k , conseguimos diminuir a injustiça do grupo R_{grp} . Para um $k = 100$ conseguimos reduzir o valor da R_{grp} para 3, 0948, utilizando o algoritmo de Filtragem Colaborativa, o que representa uma redução de 38, 15% na R_{grp} calculada antes da execução do algoritmo de equidade. Já para o algoritmo de Filtragem Baseada em Conteúdo, considerando o mesmo valor de $k = 100$, o valor de R_{grp} foi reduzido para 0, 001215, representando uma redução de 99, 96%. E por último, o algoritmo Cascata Híbrida obteve um valor final de R_{grp} de 3, 458246, para o mesmo valor de $k = 100$, o que simboliza uma redução de 37, 51%.

A Tabela 1, Tabela 2 e Tabela 3 também mostra a eficiência das recomendações depois da aplicação do algoritmo de equidade. Para os três algoritmos avaliados, quanto maior o valor de k , menor é a $RMSE$, o que simboliza que além de diminuir o valor da Injustiça do Grupo, a eficiência das recomendações aumentaram, devido a diminuição do Erro Quadrático Médio ($RMSE$). O que significa que as recomendações ficaram mais justas e eficientes.

Tabela 4 - Resultados da Injustiça do grupo R_{grp} e do Erro Quadrático Médio $RMSE$ para o agrupamento 95-5 aplicado ao algoritmo de Filtragem Colaborativa

Estratégia 95-5 para agrupamento dos usuários $\{G_1$ e $G_2\}$						
Algoritmo Filtragem Colaborativa Injustiça do grupo (R_{grp}) Média original 0,05498						
Eficiência ($RMSE$) Média original 1,40273						
k	$R_{grp}(\mu)$	$R_{grp}(\sigma)$	(%)	$RMSE(\mu)$	$RMSE(\sigma)$	(%)
10	0,041472	0,0000004	-24,56	1,3065816	0,00002517	-6,85
25	0,037957	0,0000002	-30,96	1,2891	0,00122547	-8,10
50	0,036121	0,0000004	-34,30	1,264389	0,00003060	-9,86
75	0,0351147	0,0000002	-36,13	1,255781	0,00001860	-10,47
100	0,034417	0,0000001	-37,40	1,250007	0,000009	-10,88

Tabela 5 - Resultados da Injustiça do grupo R_{grp} e do Erro Quadrático Médio $RMSE$ para o agrupamento 95-5 aplicado ao algoritmo de Filtragem Baseada em Conteúdo

Estratégia 95-5 para agrupamento dos usuários $\{G_1$ e $G_2\}$						
Algoritmo Filtragem Baseada em Conteúdo Injustiça do grupo (R_{grp}) Média original 0,20933 Eficiência ($RMSE$) Média original 1,9037						
k	$R_{grp}(\mu)$	$R_{grp}(\sigma)$	(%)	$RMSE(\mu)$	$RMSE(\sigma)$	(%)
10	0,053647	0,00000109	-74,37	1,3591	0,00004319	-28,60
25	0,046747	0,00000142	-77,66	1,313822	0,00006710	-30,98
50	0,042978	0,00000103	-79,46	1,287031	0,00005592	-32,39
75	0,041358	0,00000053	-80,24	1,274966	0,00003060	-33,02
100	0,040597	0,00000002	-80,60	1,269205	0,00001680	-33,32

Tabela 6 - Resultados da Injustiça do grupo R_{grp} e do Erro Quadrático Médio $RMSE$ para o agrupamento 95-5 aplicado ao algoritmo de Cascata Híbrida

Estratégia 95-5 para agrupamento dos usuários $\{G_1$ e $G_2\}$						
Algoritmo Cascata Híbrida Injustiça do grupo (R_{grp}) Média original 0,00453 Eficiência ($RMSE$) Média original 1,35301						
k	$R_{grp}(\mu)$	$R_{grp}(\sigma)$	(%)	$RMSE(\mu)$	$RMSE(\sigma)$	(%)
10	0,001461	0,00000017	-67,74	1,152505	0,11262306	-14,81
25	0,001368	0,00000004	-69,80	1,22611	0,00003030	-9,37
50	0,001274	0,00000001	-71,87	1,214875	0,00000272	-10,20
75	0,001156	0,00000001	-74,48	1,208628	0,00000458	-10,67
100	0,001186	0,000000007	-73,81	1,20455	00,0000118	-10,97

Considerando os dados da Tabela 4, Tabela 5 e Tabela 6, referentes ao agrupamento 95-5, podemos observar que à medida que aumentamos o valor de k , conseguimos diminuir a injustiça do grupo R_{grp} . Para um $k = 100$ conseguimos reduzir o valor da R_{grp} para 0,034417, utilizando o algoritmo de Filtragem Colaborativa, o que representa uma redução de 37,40% na R_{grp} calculada antes da execução do algoritmo de equidade. Já para o algoritmo de Filtragem Baseada em Conteúdo, considerando o mesmo valor de $k = 100$, o valor de R_{grp} foi reduzido para 0,040597, representando uma redução de 80,60%. E por último, o algoritmo Cascata Híbrida obteve um valor final de R_{grp} de 0,001186, para o mesmo valor de $k = 100$, o que simboliza uma redução de 73,81%.

Assim como nas tabelas do agrupamento 50-50, A Tabela 1, Tabela 2 e Tabela 3 também mostra a eficiência das recomendações depois da aplicação do algoritmo de

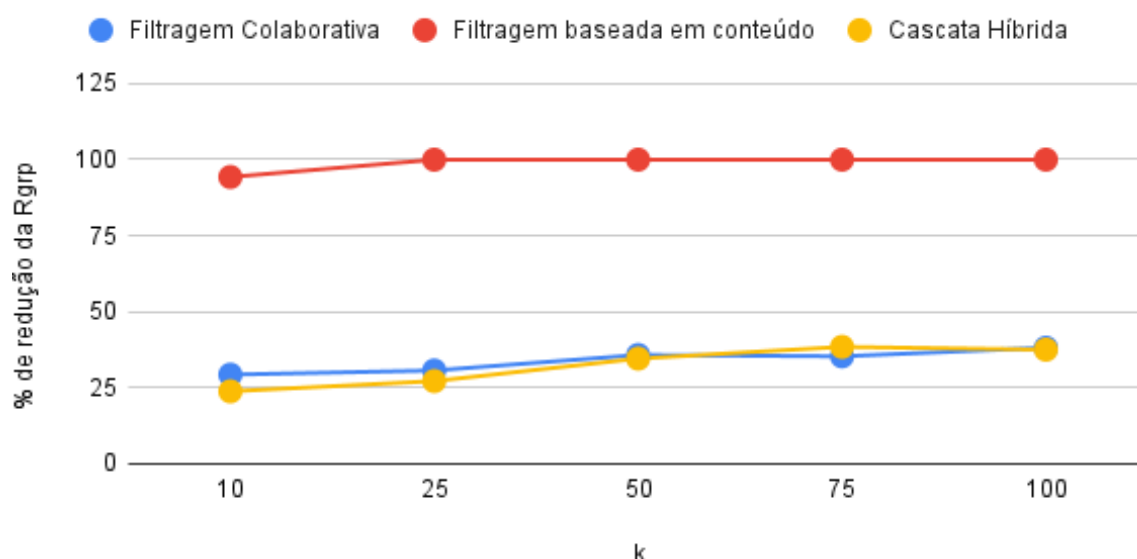
equidade. Para os três algoritmos avaliados, quanto maior o valor de k , menor é a $RMSE$, o que simboliza que além de diminuir o valor da Injustiça do Grupo, a eficiência das recomendações aumentaram, devido a diminuição do Erro Quadrático Médio ($RMSE$), o que significa que as recomendações ficaram mais justas e eficientes assim como nos testes do outro agrupamento.

Ademais, vale destacar que o número máximo de matrizes estimadas foram 100, pois foi observado que, para valores acima disso, os resultados começavam a convergir, ou seja, o ganho de resultado é mínimo ou até mesmo pior, além do custo computacional ser maior. Sendo assim, a observação de convergência sugere que adicionar mais dados ou repetir o processo não traz resultantes melhoras nos resultados.

Na Figura 7 e 8, é possível resumir todos os resultados de diminuição da injustiça do grupo baseado no número de matrizes estimadas obtidos por cada algoritmo nos dois agrupamentos testados.

Figura 7 - Representação da redução da injustiça no agrupamento 50-50

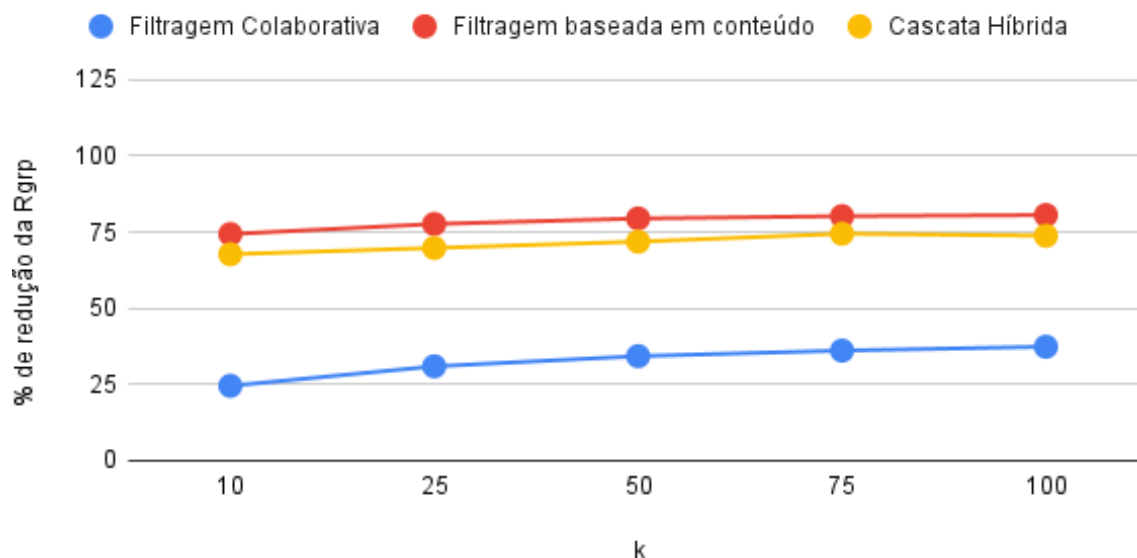
Percentuais de redução da injustiça do grupo (Rgrp) por algoritmo no agrupamento 50-50



Fonte: Autor

Figura 8 - Representação da redução da injustiça no agrupamento 95-5

Percentuais de redução da injustiça do grupo (Rgrp) por algoritmo no agrupamento 95-5



Fonte: Autor

Diante dos resultados, pode-se concluir que, calculando as médias finais de cada agrupamento, considerando $k = 100$, obtivemos uma redução na injustiça do grupo de 58,54% para o agrupamento 50-50 e uma redução de 63,94% no agrupamento 95-5. Isso mostra que, em média, tivemos um resultado melhor na situação em que temos um grupo mais privilegiado (com mais avaliações) e um grupo menos privilegiado (com menos avaliações). Sendo assim, o algoritmo mostra-se efetivo em uma situação clara de injustiça entre os grupos.

CAPÍTULO 6

CONCLUSÃO

O presente trabalho tem como proposta a elaboração e avaliação do algoritmo de equidade em sistemas de recomendação considerando diferentes estratégias de agrupamento de usuários. O algoritmo de equidade conseguiu diminuir a injustiça do grupo em ambas as estratégias de agrupamento nos dois conjuntos de dados.

As maiores reduções foram observadas no algoritmo de Filtragem Baseada em Conteúdos, para ambos os agrupamentos, sendo a maior redução para o agrupamento 50-50. Por exemplo, para um $k = 100$ na estratégia de agrupamento 50-50, foi observada uma redução da injustiça do grupo R_{grp} em 99, 96%. E já para o agrupamento 95-5, foi observada uma redução de R_{grp} em 80, 60%.

Em relação à eficiência das recomendações, pode-se observar que, mesmo considerando reduções expressivas de injustiça, tivemos também um aumento da eficiência das recomendações (diminuição do $RMSE$). E assim como o algoritmo de Filtragem Baseada em Conteúdo obteve a maior redução de Injustiça do Grupo, ele também teve o melhor ganho de eficiência, sendo uma melhora de 56, 96% para o agrupamento 50-50, e um ganho de 33, 32% para o agrupamento 95-5, considerando $k = 100$.

Como trabalhos futuros, existe a possibilidade de testar o algoritmo de equidade para outros conjuntos de dados, testar o algoritmo em sistemas de recomendação baseados em modelo (já que esse trabalho reflete um sistema baseado em memória), melhorar a documentação da utilização dos serviços presentes nesse webservice, além de testar outros otimizadores matemáticos para testes de novos resultados.

APÊNDICE A

CÓDIGO DE OTIMIZAÇÃO

Neste apêndice, será demonstrado como foi feita a otimização matemática do Gurobi para construção do algoritmo de equidade na linguagem Java. Vale ressaltar que possui outras estruturas importantes, que não foram mostradas no código desta seção, porém estão disponíveis no GitHub.

```
1 double [][] matrix_result = new double[linha][coluna];
2
3 try{
4     GRBEnv env = new GRBEnv(true);
5     env.set("logFile", "mip.log");
6     env.start();
7
8     GRBModel m = new GRBModel(env);
9
10    //HashMap<Tuple, GRBVar> xVars = new HashMap<>();
11    ArrayList<GRBVar [][]> grbVarList = new ArrayList<>();
12    ArrayList<GRBVar [][]> grbVarList2 = new ArrayList<>();
13
14
15    //Adicao variaveis
16    for(int i = 0; i < n_groups; i++){
17        int qtdUser = users_g.get(i+1).size();
18        int qtdL = ls_g.get(i+1).size();
19        GRBVar [][] var = new GRBVar[qtdUser][qtdL];
20        grbVarList.add(var);
21        grbVarList2.add(new GRBVar[qtdUser][qtdL]);
22    }
23
24    //Preencher Variaveis
25    for(int g = 0; g < n_groups; g++){
26        int qtdUser = users_g.get(g+1).size();
```

```

27     int qtdL = ls_g.get(g+1).size();
28     for(int i = 0; i < qtdUser; i++){
29         for(int j = 0; j < qtdL; j++){
30             grbVarList.get(g)[i][j] = m.addVar(0.0, 1.0,
31                 0.0, GRB.BINARY, "x_" + i + "_" + j);
32         }
33     }
34
35     ArrayList<GRBLinExpr> grbLinExprList = new ArrayList<>();
36     ArrayList<GRBLinExpr> grbConstList = new ArrayList<>();
37
38     for(int g = 0; g < n_groups; g++){
39         int qtdUser = users_g.get(g+1).size();
40         int qtdL = ls_g.get(g+1).size();
41         grbLinExprList.add(new GRBLinExpr());
42         grbConstList.add(new GRBLinExpr());
43         for(int i = 0; i < qtdUser; i++){
44             GRBLinExpr expr = new GRBLinExpr();
45             for(int j = 0; j < qtdL; j++){
46                 grbLinExprList.get(g).addTerm(list_lix[j][
47                     G_index.get(g+1).get(i)], grbVarList.get(g
48                     ) [i][j]);
49                 expr.addTerm(1.0, grbVarList.get(g)[i][j]);
50             }
51             String constrName = g + "row_sum" + i ;
52             m.addConstr(expr, GRB.EQUAL, 1.0, constrName);
53         }
54     }
55
56     m.update();
57
58     GRBLinExpr LMean = new GRBLinExpr();

```

```

57     GRBVar LMeanVar = m.addVar(-GRB.INFINITY, GRB.INFINITY,
    0, GRB.CONTINUOUS, "avgTotalVar");
58
59     for(int g = 0; g < n_groups; g++){
60         LMean.add(grbLinExprList.get(g));
61     }
62
63     LMean.multAdd(1.0 / usuarios, LMean);
64
65     m.addConstr(LMeanVar, GRB.EQUAL, LMean, "avgTotalConstr")
    ;
66
67     // Variavel auxiliar para o quadrado da media das
    medias
68     GRBVar avgTotalSq = m.addVar(0, GRB.INFINITY, 0, GRB.
    CONTINUOUS, "avgTotalSq");
69
70     // Restricao quadratica para definir avgTotalSq como o
    quadrado de avgTotalVar
71     GRBQuadExpr quad = new GRBQuadExpr();
72     quad.addTerm(1.0, LMeanVar, LMeanVar); // adiciona o
    termo quadr tico
73     m.addQConstr(quad, GRB.EQUAL, avgTotalSq, "
    avgTotalSqConstr");
74
75     // Funcao objetivo: Minimizar a variancia das medias
76
77     GRBLinExpr variance = new GRBLinExpr();
78     variance.addTerm(1.0, avgTotalSq);
79     variance.addTerm(-1.0, LMeanVar);
80     variance.addTerm(-1.0, LMeanVar);
81     m.setObjective(variance, GRB.MINIMIZE);
82

```

```
83     // Otimizar o modelo
84     m.optimize();
85
86     for(int g = 0; g < n_groups; g++){
87         int qtdUser = users_g.get(g+1).size();
88         int qtdL = ls_g.get(g+1).size();
89
90         for(int i = 0; i < qtdUser; i++){
91             for(int j = 0; j < qtdL; j++){
92                 if(grbVarList.get(g)[i][j].get(GRB.DoubleAttr
93                     .X) == 1) {
94                     matrix_result[G_index.get(g+1).get(i)] =
95                         lista_x[j][G_index.get(g+1).get(i)];
96                 }
97             }
98         }
99     }
```


REFERÊNCIAS

- AFOUDI, Y.; LAZAAR, M.; ACHHAB, M. A. Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network. *Simulation modelling practice and theory*, Elsevier B.V, v. 113, p. 102375, 2021. ISSN 1569-190X.
- AGUIAR, J. J. B.; ARAÚJO, J. M. F. R. d.; COSTA, E. d. B. Estudo comparativo de abordagens para sistemas de recomendação baseados em personalidade com uso do serviço ibm watson personality insights. *RISTI : Revista Ibérica de Sistemas e Tecnologias de Informação*, AISTI - Associação Ibérica de Sistemas e Tecnologias de Informação, n. 40, p. 73–88, 2020. ISSN 1646-9895.
- BERTOLACI, L. M. Elaboração de um sistema de recomendação baseado em conteúdo. Cachoeiro de Itapemirim, 2019.
- BORATTO, L. et al. Consumer fairness in recommender systems: Contextualizing definitions and mitigations. In: SPRINGER. *European Conference on Information Retrieval*. [S.l.], 2022. p. 552–566.
- CHEN, J. et al. Bias and debias in recommender system: A survey and future directions. *arXiv preprint arXiv:2010.03240*, 2020.
- DELDJOO, Y.; BELLOGIN, A.; NOIA, T. D. Explaining recommender systems fairness and accuracy through the lens of data characteristics. *Information processing management*, Elsevier Ltd, Oxford, v. 58, n. 5, p. 102662, 2021. ISSN 0306-4573.
- ELSABAGH, M. A.; FARHAN, M. S.; GAFAR, M. G. Meta-heuristic optimization algorithm for predicting software defects. *Expert systems*, Wiley Subscription Services, Inc, Oxford, v. 38, n. 8, p. n/a, 2021. ISSN 0266-4720.
- FILHO, R. J. R. et al. Aprendizado neural de representação de conteúdo para sistema de recomendação de filmes. Pontifícia Universidade Católica do Rio Grande do Sul, 2017.
- GÓMEZ, E.; BORATTO, L.; SALAMÓ, M. Provider fairness across continents in collaborative recommender systems. *Information Processing & Management*, v. 59, n. 1, p. 102719, 2022. ISSN 0306-4573. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S030645732100203X>>.
- GÓMEZ, E. et al. Enabling cross-continent provider fairness in educational recommender systems. *Future Generation Computer Systems*, v. 127, p. 435–447, 2022. ISSN 0167-739X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0167739X21003332>>.
- GRUBISIC, S. et al. Técnica de traçado de raios associada a metaheurísticas para otimização do posicionamento de antenas em ambientes interiores. Florianópolis, 2012.
- Gurobi Optimization, LLC. *Gurobi Optimization*. 2021. Acesso em 05 abr. 2023. Disponível em: <<https://www.gurobi.com/>>.
- HAMACHER, S. *Modelo de otimização para localização de hospitais de campanha para COVID-19*. Tese (Doutorado) — Tese de Doutorado do Curso de Engenharia Industrial da Pontifícia . . . , 2020.

HOLGUÍN-VERAS, J. et al. On the appropriate objective function for post-disaster humanitarian logistics models. *Journal of operations management*, Elsevier B.V, HOBOKEN, v. 31, n. 5, p. 262–280, 2013. ISSN 0272-6963.

KANG, H. et al. Improved harris hawks optimization for non-convex function optimization and design optimization problems. *Mathematics and computers in simulation*, Elsevier B.V, v. 204, p. 619–639, 2023. ISSN 0378-4754.

LI, Y. et al. User-oriented fairness in recommendation. In: *arXiv.org*. Ithaca: Cornell University Library, arXiv.org, 2021. ISSN 2331-8422.

LORENÇÃO, H. d. S. Elaboração de um serviço de recomendação híbrido implantado em webservice restful. Cachoeiro de Itapemirim, 2021.

Lü, L. et al. Recommender systems. *Physics reports*, Elsevier B.V, AMSTERDAM, v. 519, n. 1, p. 1–49, 2012. ISSN 0370-1573.

MARRAS, M. et al. Equality of learning opportunity via individual fairness in personalized recommendations. *International journal of artificial intelligence in education*, Springer New York, New York, v. 32, n. 3, p. 636–684, 2022. ISSN 1560-4292.

MENDES, T. M.; SANTOS, R. V. M. dos; PICOLI, J. G. Algoritmo de recomendação colaborativa aplicado ao contexto educacional/algorithm of collaborative recommendation applied to the educational context. *Brazilian Applied Science Review*, v. 3, n. 1, p. 703–711, 2018.

NOCEDAL, J.; WRIGHT, S. J. *Numerical Optimization*. New York, NY: Springer New York, 1999. (Springer Series in Operations Research and Financial Engineering). ISSN 1431-8598. ISBN 0387987932.

POCHMANN, V. D. O. Sistema de recomendação multiobjetivo para dietas alimentares. 2022.

SHOKEEN, J.; RANA, C. Social recommender systems: techniques, domains, metrics, datasets and future scope. *Journal of intelligent information systems*, Springer US, New York, v. 54, n. 3, p. 633–667, 2020. ISSN 0925-9902.

VANROYE, L. et al. Fatrop : A fast constrained optimal control problem solver for robot trajectory optimization and control. 2023.

VIANA, C. F. Elaboração de um sistema de recomendação híbrido baseado em combinação sequencial. Cachoeiro de Itapemirim, 2022.

ZEHLIKE, M. et al. Fair top-k ranking with multiple protected groups. *Information processing management*, Elsevier Ltd, Oxford, v. 59, n. 1, p. 102707, 2022. ISSN 0306-4573.

ZHANG, S. et al. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys*, ACM, Ithaca, v. 52, n. 1, p. 1–38, 2019. ISSN 0360-0300.

ZHOU, T.; ZHOU, Y.; SUNDMACHER, K. A hybrid stochastic–deterministic optimization approach for integrated solvent and process design. *Chemical Engineering Science*, v. 159, p. 207–216, 2017. ISSN 0009-2509. ICAMD – Integrating Computer-Aided Molecular Design into Product and Process Design. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0009250916301348>>.

ISBN 978-655376310-4



9

786553

763104