

# Comparison of recommendation algorithms for user groups: a food-based case study

Caio Vasconcelos , *Student Member, IEEE* and Douglas Veras , *Senior Member, IEEE*

**Resumo**—There is a rise in the development of platforms that work with the distribution of buying and selling food, and with the increase in food options and the number of users, such platforms use recommendation systems to facilitate the user's choice. These recommendations are usually based on information that the algorithm obtains previously. And increasingly, these recommendations need to be right in specific contexts. This article proposes to compare, through common metrics in the literature, the use of two recommendation algorithms in a context of user groups to make a joint recommendation. One of the algorithms uses a database of groups in neural network training, and the other algorithm uses databases of auxiliary domains with different contexts to perform the prediction. The results indicate that it is possible to perform the prediction for groups of users even if a database with scarce data is used. The article is a theoretical basis to show the efficiency of recommending it to groups in the food domain, and can be incorporated and added to existing platforms.

**Index Terms**—Recommender Systems for Groups, Food Recommender Systems, Cross-domain.

## I. INTRODUÇÃO

Sistemas de recomendação (SR) são sistemas que visam fazer previsões de itens de interesse para um usuário, levando em conta informações obtidas previamente [1] [2] [3] [4]. Essa área tem sido adotada no e-commerce para direcionar as vendas para usuários específicos facilitando a escolha de um produto [5].

Dito isso, os SRs no domínio alimentício têm sido alvo de pesquisas, principalmente devido a uma grande parcela da população estar ficando acima do peso e com doenças decorrentes de distúrbios alimentares [6]. Mas, em comparação a outros domínios estudados utilizando a tecnologia de SR, como sistemas multimídia: filmes, séries e livros, o domínio alimentício ainda é considerado pouco explorado e com muitos desafios a serem superados [4] [6].

Com relação a esses desafios, podemos citar aqui como o principal o fato do cenário alimentício possuir muitas possibilidades de utilização de variáveis extraídas do usuário (pressão arterial, nível de estresse, entre outros) e do contexto do qual está inserido (GPS, clima, entre outros) que devem ser interpretados pelo sistema, a fim de realizar uma predição mais assertiva para o usuário final. Portanto, a chave para uma boa predição é ter uma boa modelagem de dados do usuário e de seu contexto [4].

Ainda assim, os problemas citados acima são encontrados em todos os SRs em um domínio alimentício. Quando levamos em conta um grupo de usuários, surgem outros problemas relacionados, tais como: a forma de combinar/agregar modelos individuais para gerar uma recomendação conjunta, pois cada

grupo possui uma configuração, em que cada preferência individual deve ser levada em conta quando houver a execução do sistema [7].

## II. REVISÃO DA LITERATURA

Dado que trabalhos com foco em grupos de usuários são mais escassos que trabalhos envolvendo apenas um único usuário, alguns autores têm desenvolvido novas abordagens a fim de conseguir o melhor resultado perante o cenário analisado. Dito isso, uma abordagem interessante é a baseada em *feedbacks* [3] [8]. No estudo dessa abordagem foi desenvolvido um protótipo utilizando uma base de dados de receitas com o objetivo de fazer recomendações para um grupo considerando o *feedback* de algumas receitas iniciais já aprovadas pelo usuário. Adicionalmente, foram usadas *tags* associativas para definir categorias às receitas. Porém, o trabalho ainda possui limitações, faltando uma forma de priorização de usuários para definir os que têm maior poder de decisão no grupo (exemplo: pais e filhos), e também por não levar em conta restrições alimentares de usuários individuais no momento de calcular o resultado [8].

Não apenas foram exploradas abordagens novas, como também foram adaptadas abordagens já comuns na literatura no uso de SR para um único usuário. No trabalho de BERKOVSKY e FREYNE [9] há uma análise para definir qual seria a melhor estratégia para um sistema de recomendação de receitas que utiliza como base a filtragem colaborativa (FC), e com foco em um grupo de usuários de estrutura familiar. Assim, ele propõe que a melhor estratégia está relacionada a uma ótima forma de agregação dos dados individuais dos membros do grupo, de forma que os pesos reflitam a interação observada entre esses membros. O trabalho é limitado por ser uma solução para um único caso muito específico [9]. Desta forma, seria necessário estender essa solução para todos os tipos de grupos.

Isto é, um ponto em comum entre as abordagens para SRs desenvolvidas para grupos de usuários é que todas elas estão relacionadas a uma boa forma de agregação de dados dos usuários participantes [10].

Diante dessas limitações e da escassez de trabalhos que solucionem o problema geral, a proposta desse trabalho é realizar um comparativo entre algoritmos que tiveram êxito na tarefa de realizar recomendações com bons resultados nas métricas em outros domínios para usuários considerando informações obtidas previamente. Um deles é o *Cross-Domain Context-Aware Recommendation System* (CD-CARS). A abordagem CD-CARS é uma junção de *Cross-Domain Recommender System* (CDRS) e *Context-Aware Recommender Systems*

(CARS). A abordagem CDRS utiliza um domínio auxiliar para fazer a recomendação para o domínio principal (como por exemplo, recomendar livros baseados em filmes), e a CARS incorpora o contexto do usuário (por exemplo, localização, hora, humor, etc.) para fazer a recomendação [11]. Ademais, o CD-CARS não se limita ao conhecimento do domínio podendo ser alterado conforme a necessidade. Além disso, a maioria das SRs utilizam apenas um único domínio, e não utilizam contexto, para fazerem suas predições [1] [2] [3] [4] [12] [8], enquanto que o CD-CARS utiliza um domínio auxiliar para melhorar a qualidade da predição em cenários em que ocorre tal escassez de dados e variáveis voláteis no contexto do usuário [11]. E o segundo é o algoritmo Attentive Group Recommendation (AGREE) [13] disponibilizado no repositório do github Attentive-Group-Recommendation. O AGREE utiliza um mecanismo de atenção para representar os grupos, onde os membros do grupo e os itens são incorporados à rede de atenção neural, para que os grupos sejam definidos, em seguida é possível dividir as etapas em camadas, sendo os itens enviados para a camada agrupamento, depois para uma camada oculta, e em seguida para a camada de predição. Esse sistema utiliza uma técnica descrita como *neural collaborative filtering* (NCF), para aprender as interações entre grupos e itens por usuário, e utiliza um domínio único.

O AGREE foi selecionado por ser uma solução recente para SRs no estado da arte e com bons resultados. E o CD-CARS, com relação aos trabalhos relacionados, nenhum deles tratou do domínio alimentício para grupos utilizando uma técnica de domínio cruzado e múltiplos contextos.

### III. MÉTODOS

#### A. Base de dados

A base de dados utilizada foi obtida através do processamento da base disponibilizada pela *Amazon* [14]. O conjunto de dados da *Amazon* descreve as notas, que vão de 1 a 5, sendo 1 a menor nota e 5 a maior nota; e os ingredientes que formam a receita descrita. A partir da base disponibilizada, foram feitos agrupamentos aleatórios entre os indivíduos com no mínimo 2 e no máximo 5 pessoas. Para fazer uma análise mais assertiva, foi estabelecido uma regra em que cada grupo deve possuir pelo menos 1 avaliação em comum entre os integrantes.

A base de dados foi dividida, utilizando o método de *hold-out*, de maneira aleatória em treinamento e teste, com uma relação de 70% e 30%, respectivamente.

Também foi realizada uma filtragem em mais duas bases de dados referentes a música e filmes para serem utilizadas como domínio auxiliar no CD-CARS, a filtragem foi realizada de modo que ficassem apenas avaliações realizadas enquanto em companhia de outro usuário, e que a ambas as matrizes individuais de usuários por produtos tivessem tamanhos iguais.

O número de amostras, após o tratamento das bases de dados foram, aproximadamente 3800 avaliações para o domínio de música, 1300 para o domínio de filmes e 4400 para alimentos. Totalizando uma base de dados de aproximadamente 9500 avaliações. A base de dados está representada na representado na Tabela I, também com mais informações.

#### B. Métricas

As métricas utilizadas para a avaliação, exclusivamente no experimento com o AGREE foram *Recall*, *Precision*, *F1-score* e *Normalized Discounted Cumulative Gain* (NDCG), e para o experimento no AGREE e no CD-CARS, foram utilizados: *mean absolut error* (MAE) e *root mean squared error* (RMSE).

As métricas de *Recall* e *Precision* se tratam de uma fração de itens relevantes recuperados nos K elementos recomendados pelo sistema, sendo a *Precision* relativa ao número de acertos previstos e o *Recall* relativo aos acertos reais. A métrica *F1-score* é um valor calculado com base nas métricas *Recall* e *Precision* [15] [16]. A métrica NDCG toma como base o ranqueamento, definido durante os ciclos, dos itens na lista definida por K, cujo K representa K itens recomendados, nesse caso sendo K = 5 [17]. Para as métricas definidas, quanto maiores os valores, melhor a qualidade [13] [18] [19] [20] [21].

As métrica RMSE e MAE são métricas populares para avaliar a performance em sistemas de recomendação. Essas métricas calculam o erro referente a diferença entre as pontuações previstas e as avaliações reais dos usuários [22]. A diferença entre eles é que o RMSE penaliza grandes erros enquanto o MAE dá o mesmo peso a todos os erros [23] [17]. Para as métricas *Recall*, *precision*, *F1-score*, quanto maiores os valores, melhor a qualidade. E para as métricas RMSE e MAE, quanto menores os valores, melhor a qualidade. Todas as métricas possuem valores referentes a uma escala entre 0 e 1.

#### C. Configurações dos algoritmos

As configurações definidas no arquivo `config.py` para o AGREE estão ilustradas abaixo:

```
self.embedding_size = 2
self.epoch = 6
self.num_negatives = 6
self.batch_size = 256
self.lr = [0.000005, 0.000001, 0.0000005]
self.drop_ratio = 0.2
self.topK = 5
```

(a) Configurações do AGREE

Dentre as configurações do AGREE, o tamanho da *embedding* é uma técnica utilizada no aprendizado de linguagens neurais e é relativo ao tamanho do da estrutura de vetores em que será realizado a incorporação [24], neste caso sendo 2 devido ao baixo número de recomendações na base de dados. O número de épocas, é relativo ao número de repetições envolvendo treinamento e avaliação de teste, limitado a 6 devido a não haver grande variação após essa iteração, e ao alto tempo de execução por época. O número de negativos se trata do número de itens avaliados para gerar a lista de K itens ranqueados. O tamanho do *batch* é referente ao espaço que será ocupado pelo dataset no sistema. A variável *learning rate* (LR) é a variação, também utilizada em sistemas multiagentes como técnica de aprendizagem por reforço, é a variação

Tabela I: Informações das bases de dados.

Domínio	Quantidade de avaliações	Quantidade de grupos	Quantidade de produtos	Média de avaliações por grupo
Música	3797	929	929	4.08
Filmes	1294	929	929	1.39
Alimentos	4391	499	1296	8.79
<b>Total</b>	<b>9482</b>	<b>2297</b>	<b>3154</b>	<b>4.12</b>

de aprendizagem do algoritmo ou agente ao realizar uma determinada ação [25]. A taxa de abandono, em uma tradução livre, é utilizada para zerar parte da base de dados durante o treinamento para regularização e prevenção da coadaptação de neurônios [26]. E a variável *topK* é referente ao tamanho da lista de ranqueamento que irá ser utilizado para fazer a recomendação.

O algoritmo do CD-CARS utilizado foi referente à uma modificação desenvolvida de SILVA [27] que dispõe de uma melhor performance e propõe o uso de dois domínios auxiliares para realizar a predição com o intuito de melhorá-la. Com base nisso, também foram utilizados dois domínios auxiliares, sendo eles: Música e filmes. Este algoritmo modificado utiliza a técnica de avaliação *Transfer collaborative filtering via Consensus Regularization (Tracer)* que realiza uma fatoração de matrizes, construindo padrões de avaliações ao nível de *cluster* usuário-item em múltiplos domínios auxiliares, transferindo ao domínio alvo os padrões mais relevantes para a geração das predições, juntamente ao algoritmo de "pós filtragem" que realiza a predição com base nos resultados do *Tracer* e nos critérios contextuais selecionados e disponibilizados na base de dados [27].

```

/* Global Configuration */

double trainingPercentage = 0.70;
double datasetPercentage = 1.0;
boolean enableFixedTestSeed = false;

/* TRACER Configuration */

double userClusterTransfer = 15.0;
double itemClusterTransfer = 15.0;
double trainingIterations = 15;
double alpha = 1.0;
double beta = 0.0001;

```

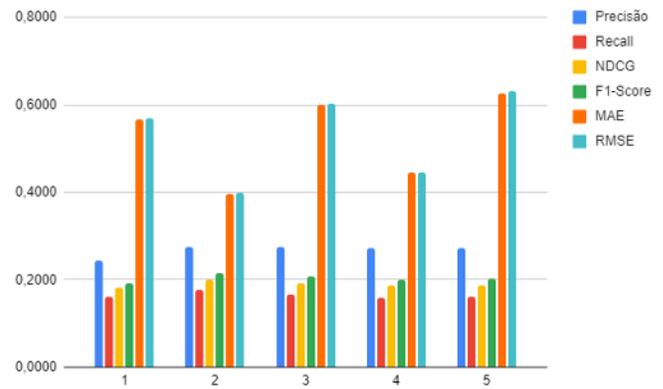
(a) Configurações do CD-CARS

Dentre as configurações do CD-CARS, foi definido a porcentagem que seria destinada a treino e teste da base de dados, e que a base de dados seria usado por completo na soma de treino e teste. Foi definido que os testes seriam definidos aleatoriamente por execução na variável. A transferência no *cluster* foi definida para 15.0 para usuários e itens. O número de iterações para repetições no treinamento foi de 15 rodadas. E as variáveis Beta e Alfa são variáveis utilizados pelo *Tracer* que devem variar entre 0.00001 e 0.1 no Beta, e no Alfa entre 0.05 e 2.

Tanto as configurações do CD-CARS quanto as configurações do AGREE foram definidas com base nos melhores

resultados conforme apresentados nas rodadas preliminares de experimentação.

Os códigos de linha de base e do CD-CARS foram executados em uma *Central Processing Unit (CPU)* 11th Gen Intel(R) Core(TM) i5-11400F @ 2.60GHz 2.59 GHz, possuínte de um memória de 16 Gigabyte (GB) e uma placa de vídeo Nvidia GTX 1660 Super. Python na versão 3.10.2. Nessas configurações, o algoritmo AGREE executou em média uma época a cada 2 horas e 45 minutos, e o algoritmo CD-CARS obteve um resultado por execução a cada 30 minutos.



(a) Tabela de resultados do AGREE

#### IV. RESULTADOS

Nos resultados do AGREE, houve pouca discrepância a partir da 6ª época de treinamento, as mudanças mais relativas começam a ocorrer entre a 1ª e a 3ª época de treinamento. Os resultados do CD-CARS envolveram os contextos de final de semana e dia útil; e os domínios secundários de músicas e filmes. Além disso, foram executadas 5 rodadas de execução no AGREE e no CD-CARS. O CD-CARS utilizou uma divisão aleatória de itens, enquanto que o AGREE foi dividido durante a etapa de filtragem da base de dados, e repetida em número de 5 vezes por execução.

#### V. DISCUSSÃO

Observando os resultados do AGREE na Tabela II, os resultados mais significativos se encontram na primeira rodada onde houveram resultados muito similares ao CD-CARS, com relação as métricas MAE e RMSE, indicando um baixo nível de erro com relação as predições reais, porém os resultados do treinamento não se mantém constantes e tornam-se incertos com base nos dados obtidos. Já ao observar o CD-CARS, os resultados na Tabela III se mantém constantes em cada rodada, tornando a confiança na base mais aceitável que no AGREE.

Tabela II: Resultados do algoritmo AGREE.

Rodada	Iteração	Precisão	Recall	NDCG	F1-Score	MAE	RMSE
1	1	0.1525	0.0856	0.1019	0.1096	0.56774	0.56868
	2	0.2288	0.1512	0.1706	0.1821	0.56764	0.56858
	3	0.2458	0.1600	0.1814	0.1938	0.56757	0.56852
	4	0.2373	0.1569	0.1771	0.1889	0.56753	0.56847
	5	0.2436	0.1615	0.1820	0.1943	0.56741	0.56834
	6	0.2436	0.1603	0.1811	0.1934	0.56736	0.56830
2	1	0.1568	0.0821	0.1005	0.1077	0.39594	0.39890
	2	0.2415	0.1554	0.1768	0.1891	0.39589	0.39885
	3	0.2627	0.1686	0.1921	0.2054	0.39587	0.39883
	4	0.2669	0.1706	0.1947	0.2082	0.39584	0.39880
	5	0.2648	0.1710	0.1945	0.2078	0.39581	0.39877
	<b>6</b>	<b>0.2754</b>	<b>0.1760</b>	<b>0.2008</b>	<b>0.2148</b>	<b>0.39579</b>	<b>0.39875</b>
3	1	0.1695	0.0926	0.1115	0.1198	0.60138	0.60281
	2	0.2606	0.1541	0.1805	0.1937	0.60115	0.60257
	3	0.2733	0.1620	0.1897	0.2034	0.60093	0.60235
	4	0.2797	0.1670	0.1949	0.2091	0.60069	0.60210
	5	0.2797	0.1657	0.1939	0.2081	0.60046	0.60187
	6	0.2754	0.1651	0.1924	0.2065	0.60034	0.60174
4	1	0.1610	0.1042	0.1184	0.1266	0.44464	0.44530
	2	0.2479	0.1579	0.1805	0.1929	0.44464	0.44530
	3	0.2627	0.1539	0.1812	0.1941	0.44447	0.44513
	4	0.2691	0.1593	0.1865	0.2001	0.44437	0.44504
	5	0.2712	0.1619	0.1891	0.2028	0.44427	0.44494
	6	0.2712	0.1599	0.1876	0.2012	0.44423	0.44490
5	1	0.1356	0.0662	0.0831	0.0889	0.62537	0.63034
	2	0.2479	0.1425	0.1685	0.1810	0.62533	0.63030
	3	0.2627	0.1485	0.1768	0.1898	0.62538	0.63034
	4	0.2691	0.1534	0.1820	0.1954	0.62537	0.63033
	5	0.2733	0.1560	0.1849	0.1987	0.62540	0.63035
	6	0.2733	0.1603	0.1881	0.2021	0.62541	0.63037

Tabela III: Resultados do algoritmo CD-CARS.

Rodada	MAE	RMSE
1	0.10663	0.21012
<b>2</b>	<b>0.088434</b>	<b>0.198007</b>
3	0.09749	0.21890
4	0.09509	0.21011
5	0.09340	0.21296

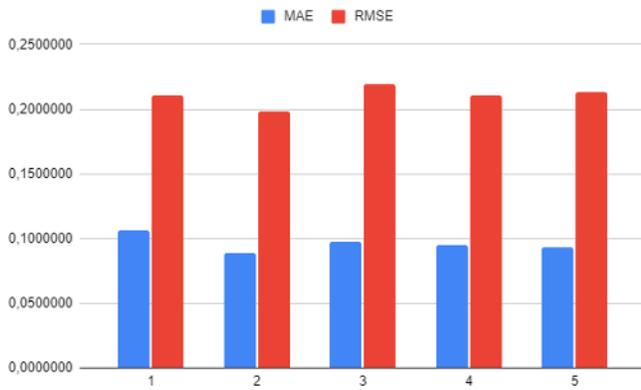
O algoritmo AGREE conseguiu obter resultados bons, mas inconsistentes quando comparados ao algoritmo CD-CARS utilizando uma dupla base de dados auxiliar. O algoritmo CD-CARS teve vantagem no uso de mais de um domínio, obtendo resultados melhores, nesse caso onde foi utilizado uma base de dados com escassez de informações. Estes

resultados, possivelmente, devem ser aprimorados conforme o crescimento da base de dados.

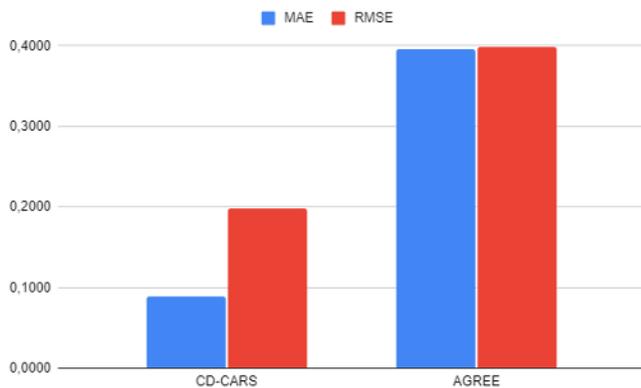
Com relação as métricas, não foi possível realizar as métricas de Precisão, *Recall*, NDCG e *F1-Score*, pelo fato que devido a filtragem do CD-CARS por contexto, o número de K itens por grupo foi reduzido, portanto, é necessário um aumento no número de itens para que seja calculado as métricas no CD-CARS.

## VI. CONCLUSÃO

O artigo se propôs a comparar algoritmos de recomendações adaptados para uma base de dados alimentícia, de forma que obtivesse avaliações para grupos de usuários. Ambos os algoritmos tiveram bons resultados e conseguiram realizar



(a) Tabela de resultados do CD-CARS



(a) Tabela de comparação dos melhores resultados do CD-CARS com o AGREE

predições avaliativas para os grupos, com diferenças de resultados.

Logo, é possível realizar avaliações para grupos de usuários no domínio alimentício com uma boa performance, tendo em vista que mesmo com um número escasso de base de dados que possuam as informações necessárias para uma avaliação em grupo para usuários, existem algoritmos capazes de realizá-los com uma boa qualidade avaliativa. Mas, é necessário avaliar estes resultados em grande escala, com uma base de dados de proporções reais para verificar se os resultados se mantêm em igualdade com o estudo realizado.

Além disso, também é necessário avaliar futuramente quais informações podem ser utilizadas para agregar na avaliação, de forma a aumentar a predição, como já exemplificado anteriormente, é possível utilizar informações obtidas do usuários previamente, como: pressão arterial, clima, região, entre outros. Além de que também é necessário que a avaliação leve em conta restrições e individualidades alimentares, além dos pesos avaliativos para cada indivíduo, como citado no início do artigo.

Também é preciso comparar estes resultados com outros algoritmos que são visados pelo estado da arte, como GroupIM [18] ou *Deep reinforcement learning based group recommender system* (DRGR) [20].

Ademais, em uma pesquisa futura é necessário ser comparado as outras métricas que não foram realizadas no CD-

CARS, para comparar com os algoritmos de base e verificar sua efetividade.

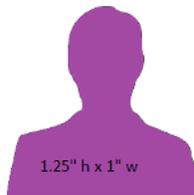
## AGRADECIMENTOS

Os autores agradecem o Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) (Processo xxxxx/20xx-4), a Fundação Araucária e a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelos suportes financeiros para o desenvolvimento do presente trabalho.

## REFERÊNCIAS

- [1] E. Asotic, "Multi-list food recommender systems for healthier choices," Master's thesis, The University of Bergen, 2021.
- [2] S. Bundasak, P. Yoksuriyan, P. Kuntawee, and R. Kotama, "Food recommendation system for the elderly," *NU. International Journal of Science*, vol. 18, no. 1, pp. 152–167, 2021.
- [3] M. Ge, M. Elahi, I. Fernáandez-Tobías, F. Ricci, and D. Massimo, "Using tags and latent factors in a food recommender system," in *Proceedings of the 5th International Conference on Digital Health 2015*, pp. 105–112, 2015.
- [4] W. Min, S. Jiang, and R. Jain, "Food recommendation: Framework, existing solutions, and challenges," *IEEE Transactions on Multimedia*, vol. 22, no. 10, pp. 2659–2671, 2019.
- [5] T. Silveira, M. Zhang, X. Lin, Y. Liu, and S. Ma, "How good your recommender system is? a survey on evaluations in recommendation," *International Journal of Machine Learning and Cybernetics*, vol. 10, pp. 813–831, 2019.
- [6] C. Trattner and D. Elweiler, "Food recommender systems: important contributions, challenges and future research directions," *arXiv preprint arXiv:1711.02760*, 2017.
- [7] J. Masthoff, "Group recommender systems: Combining individual models," in *Recommender systems handbook*, pp. 677–702, Springer, 2010.
- [8] M. Elahi, M. Ge, F. Ricci, D. Massimo, and S. Berkovsky, "Interactive food recommendation for groups," in *Recsys posters*, Citeseer, 2014.
- [9] S. Berkovsky and J. Freyne, "Group-based recipe recommendations: analysis of data aggregation strategies," in *Proceedings of the fourth ACM conference on Recommender systems*, pp. 111–118, 2010.
- [10] T. N. Trang Tran, M. Atas, A. Felfernig, and M. Stettinger, "An overview of recommender systems in the healthy food domain," *Journal of Intelligent Information Systems*, vol. 50, pp. 501–526, 2018.
- [11] D. Vêras, R. Prudêncio, and C. Ferraz, "Cd-cars: Cross-domain context-aware recommender systems," *Expert Systems with Applications*, vol. 135, pp. 388–409, 2019.
- [12] K. Ahuja, M. Goel, S. Sikka, and P. Makkar, "What-to-taste: A food recommendation system," 2020.
- [13] D. Cao, X. He, L. Miao, Y. An, C. Yang, and R. Hong, "Attentive group recommendation," in *The 41st International ACM SIGIR conference on research & development in information retrieval*, pp. 645–654, 2018.
- [14] B. P. Majumder, S. Li, J. Ni, and J. McAuley, "Generating personalized recipes from historical user preferences," *arXiv preprint arXiv:1909.00105*, 2019.
- [15] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proceedings of the fourth ACM conference on Recommender systems*, pp. 39–46, 2010.
- [16] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015.
- [17] Y. Wang, L. Wang, Y. Li, D. He, and T.-Y. Liu, "A theoretical analysis of ndcg type ranking measures," in *Conference on learning theory*, pp. 25–54, PMLR, 2013.
- [18] A. Sankar, Y. Wu, Y. Wu, W. Zhang, H. Yang, and H. Sundaram, "Groupim: A mutual information maximization framework for neural group recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1279–1288, 2020.
- [19] L. Vinh Tran, T.-A. Nguyen Pham, Y. Tay, Y. Liu, G. Cong, and X. Li, "Interact and decide: Medley of sub-attention networks for effective group recommendation," in *Proceedings of the 42nd International ACM SIGIR conference on research and development in information retrieval*, pp. 255–264, 2019.

- [20] Z. Liu, S. Wen, and Y. Quan, "Deep reinforcement learning based group recommender system," *arXiv preprint arXiv:2106.06900*, 2021.
- [21] M. M. Afsar, "Personalized recommendation using reinforcement learning," 2022.
- [22] G. Douglas, "Open access proceedings journal of physics: Conference series," 2019.
- [23] T. Chai and R. R. Draxler, "Root mean square error (rmse) or mean absolute error (mae)," *Geoscientific model development discussions*, vol. 7, no. 1, pp. 1525–1534, 2014.
- [24] W. Gu, A. Tandon, Y.-Y. Ahn, and F. Radicchi, "Principled approach to the selection of the embedding dimension of networks," *Nature Communications*, vol. 12, no. 1, p. 3772, 2021.
- [25] M. Bowling and M. Veloso, "Multiagent learning using a variable learning rate," *Artificial Intelligence*, vol. 136, no. 2, pp. 215–250, 2002.
- [26] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.
- [27] G. M. d. Silva, "Avaliação entre algoritmos de filtragem colaborativa baseada em vizinhança e transferência de conhecimento para cd-cars," B.S. thesis, Brasil, 2019.



**3rd author** Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



**1st author** Caio Giovanni Pereira Vasconcelos, estudante da Universidade Federal Rural de Pernambuco.

## ACRÓNIMOS



**2nd author** Douglas Veras