Recommendation System in Business Intelligence Solutions for Grocery shops: Challenges and Perspective

Atul Mohan, Anas Abdelrazeq, and Frank Hees
Chair of Information Management in Mechanical Engineering (IMA) - RWTH Aachen University
Dennewartstr. 27
52068 Aachen, Germany
+49 241 80 91131
{atul.mohan, anas.abdelrazeq, frank.hees}@ima-ifu.rwth-aachen.de

ABSTRACT

With the surge in digital platforms and extension of e-commerce, the field of recommendation has been a topic of interest not only for the data scientist but deemed important by the business experts to enhance the user-centric services. A large number of retail & service-oriented companies such as Amazon, Netflix, Goodreads, and Spotify etc. use Business Intelligence (BI) and recommendation systems to provide users with various choices of products based on their interest. Evidently, such a customized user-experience not only provide them with a better service, but also enables the companies to understand customer behavior and enhance their business. The aim of this paper is to introduce a recommendation system in the business intelligence platform to a *new-system* where no user's previous interaction information is available. We present an exploratory study of implementing *recommendation system* in the project SmartEmma, a grocery shop application in Aachen, funded by EFRE.NRW, European Union and WIRTSCHAFT.NRW.

CCS Concepts

• Information systems → Information Systems Applications → Electronic Commerce → E-commence Infrastructure • Information systems → Information Systems Applications → Electronic Commerce → Online Shopping

Keywords

E-Commerce; Business Intelligence; Recommendation System; Collaborative Filtering.

1. INTRODUCTION

SmartEmma is a research project that attempts to build a digital marketplace for online food shopping from various small-scale grocery retailers in the city of Aachen, Germany. This platform not only assists these retailers by bridging the gap between their current business model and e-commerce but also allows the customers to shop different products from a variety of stores through a smartphone application. Moreover, it provides these retailers with a business intelligence platform that supplies various analytics to support them in keeping track and enhancing their sales. With such a digital platform, SmartEmma aims to preserve the diversity in area of stationary food retailing and setup an eco-friendly distribution channel based on the quality of offline trading. The application provide the customers a flexibility of customizing their shopping cart with products from the pool of retailers with various offers. Furthermore, within the competition of increasing the sale by enhancing user's engagement with the application and their experience of shopping, the use of recommendation systems that guide them towards the product of their interest and provide them with more choices, has become obligatory.

BI platforms, on the other hand, allows the retailer to understand the customer's behavior and patterns within their sales. However, for such an enterprise, setting up a BI platform for the retailers and incorporating a recommendation system for the customer spawns' various challenges. In addition to well-recognized issues of *cold-start* and *data sparsity*, the application has overcome challenges of integrating data from different retailers and *new-system* as no previous transaction is provided.

This paper aims towards discussing the setup of a recommendation system within the BI platform of project SmartEmma and methods used to overcome the challenges of data integration and cold-start during the implementation. Section 2 presents a summarized background of different filtering methods used in the recommendation system along with their challenges and then proceeds to describe the implementation details used to overcome them in SmartEmma, in Section 3. Finally, Section 4 presents the evaluation of the system and Section 5 concludes the discussions by summarizing the key points of the findings.

2. Background

Recommendation Systems (R.S) are filtering methods that understand the preference of a group and then recommend the most relevant items or products to users[1]. In a business context, this can also be defined as the likelihood of a user buying a certain product or a single value on a certain scale that depicts the user's preference over different products. In eq. 1, this concept is depicted as a function of user and products, which results in a score that quantifies user's interest toward the product.

$$\boldsymbol{R}\text{:}\,\boldsymbol{User} \times \boldsymbol{Item} \rightarrow \boldsymbol{Rating} \qquad \text{eq. 1}$$

Over the years, recommendation systems have been used to solve the problem of connecting current users to the most relevant products over the millions of items in an inventory [2], [3]. For example, recommendations of books, products and various items from Amazon [4], [5], movies by Netflix, personalized newspaper applications or music on Spotify are a few successful examples recommendation systems being used in the e-commerce application today. Although the designs and working of these systems are dependent on the domain of use and the data available, most of them rate the likelihood of user buying an item on a scale of 1 (least likely) to 5(most likely). Additionally, the filtering of items or user can be performed by providing user-specific or item-specific information such as contextual-demography, features and utility of products etc. to enhance the results of these system. The following paragraphs presents a few of the currently used R.S that were considered for this project and their limitations.

Association rule mining has been used in the field of data mining over the years to discover frequently occurring attribute-value relationships of items in a given data set [6], [7]. Oftenly used in *market-basket analysis,* this method aims to idetify rules to predict the occurance of an item based on the other items currently present in the basket. As argued in [8], although identifing association rules mining might not be actively used in very large product inventories or in case of a new system, it has been successfully used in recommending website and organization of webpages[9], [10] .

Collaborative-Filtering (CF) was first introduced as a commertial recommendation system to recommend news articles to a collection of users in "Tapestry" [1]. It analyses historical interaction and usage data across users to identify user/item groups that are well matched. For instance, if a group of people are found to like chocolates and an individual is identified to frequently buy a certain brand of chocolate, then it is safe to assume that the individual might also like other chocolates frequently bought by group of people [11]. The CF methods are broadly classified into two types, namely, a) Memory based approaches [12] and b) Model Based approaches [13], [14]. The difference between these approaches is in the method used to compute the similarity between the customer's group i.e. statistical methods in the case of memory based approach and learning the parameters using gradient descent methods [15], [16], matrix factorization [13], [17] or multi-layered neural networds [14], [18] for model based methods. However, as this method is based on the similarity of taste among users, it has major challenges while dealing with data sparcity, cold-start problem.

The *Content-based filtering (CB)* method is based on the profile attributes of the users as well as products. It recommends items that are similar in content to the products that user has rated high in the past or matches with the marker attributes of the user [3], [19]. A profile created for users and items describes various inherent characteristics, which helps the R.S. in associating users to the best matching items. On one hand where this method allows for independence of users, transparency in rating and overcomes the cold-start problem when a new item is introduced to the system, it has to deal with issues of limited-content analysis, overspecialization and new user being added to the system [20].

Demography-based R.S produce different recommendations to user based on his/her demography, *Knowledge-based* R.S rely on the specific domain knowledge of item features to map them to user's requirement [21]. Similar to these is *Utility-based* R.S, which base their recommendation on assessment between the requirements of user and properties of various items available in the inventory. As a result, these systems overcome the challenge of cold-start and data sparsity but have to deal with issues of acquiring extensive user & catalogue knowledge, various features of item and functions that map user's need to items.

Evidently, the task of selecting a recommendation algorithm is influenced by various factors such as the amount of information available on users and items, the amount of user's transactional data available on which the recommendation scoring is learned and scalability of the algorithm as the platform and inventory grows. Considering the case of SmartEmma, where the platform serves various small-scale retailers, the challenges are many folds.

Firstly, due to the lack of a common digital platform earlier, the inventories do not concur to a single schema resulting in a variety of schemas and varying degree of information for users and products. Moreover, many of these retailers do not categorize or provide a description of their products, which makes categorization of products and subsequently segregation of users very difficult. As a result, the application needs a mediated schema that retains maximum information while reducing sparsity within data fields. Secondly, as the application would be a 'new-system", it does not provide the historical transactions of the users. This results in the *cold-start* problem where no recommendation for a new item to the users or recommendation of products to a new user can be provided.

Furthermore, most of the retailers only have a very basic information of their user such as age, gender and language. Coupled with the lack of historical transactions, rules out using methods like CB filtering, KB filtering and Association rule mining in absence of user profiles and product catalogs. Thirdly, the recommendation system faces the problem of *gray-sheep,* which defines a group of users who provide no ratings or do not consistently agree or disagree with any user groups [22]. Such users lead to the problem of data sparsity where insufficient data leads to an incorrect rating to the product. As a consequence of these challenges, the use of collaborative filtering suits the best for the case of SmartEmma. This decision is further supported by the fact of a large product bases because of conglomerating various retailer for the same platform and the benefits of CF in [4], [5].

3. Implementation

The R.S in SmartEmma consist of two modules, a pre-processor to integrate and clean the data from various retailers and an Item-Item collaborative filtering to assign a rating to product as well as stores. The pre-processor module is an *extract-transform-load* (ETL) [23], [24] scenario, that is responsible for cleaning and transforming inventory and transactional data into a star schema for warehousing and BI analytical purposes. The CF module is a prediction algorithm that predicts the rating of a product and a store (retailer) by a given user, using Item-Item CF method. This section presents the implementation details of these modules.

3.1 Data Pre-processing

A data model is a concept of Database Management Systems (DBMS) that provides the conceptual representation of the entire schema and describes the structure of database, depicted in form of an ER diagram [25], [26]. Modelling of data in different schemes of every retailer is necessary in order to create a mediated platform before the RS can process it to generate recommendations. For such a transformation of data, ETL tools are used for extraction of relevant data from the source and then transform into the defined models. Therefore, based on the initial information provided by the retailers, the key tables and their transformation are described below:

- Customers/Users: This model consist of basic information of the users. For all the retailers, a maximal set of all the attributes was taken to form the dataset. Considering the scenario where most of the customers don't prefer sharing various details online or over digital platforms, the current customer table comprise of fields *birthdate, gender, language, created date* and *updated date*.

- Store: The store details for most of the retailers were similar to one-another. Therefore, the maximal attribute set from all retailers is taken as the attribute set. However, to flatten the information, the details for location and schedule were stored in separate weakly connected tables, as such information is not present for all retailers.

- Products: The product table comprise of various features that provide varying level details over the product. For example, where one retailer provides extensive details over the size of product and subsequent prices for its products, the other retailer provides only very basic detail of the product with no updates on these information. Therefore, a maximal set of these information is taken with the consideration that the missing information would be provided later as the application matures.

- Orders/Transactions: In case of SmartEmma, transactions are associated with a user and a single transaction comprise of multiple smaller transactions for each retailer, from which customer has bought a product. The nested structure of each order comprise of details such as size, weight, actual price, applied price etc., which needs to be flattened. Thus, each transaction record is segregated into two tables namely, *order-header & order-details*, where the header table contains exactly one record of each record comprising of meta-information of the transaction whereas the detail contains multiple entries of each record comprising of product and store detail of every bought product. This flattening not only provides opportunity to maintain a clean record set, it allows specific utilization of each information within the BI module of the application over OLAP.

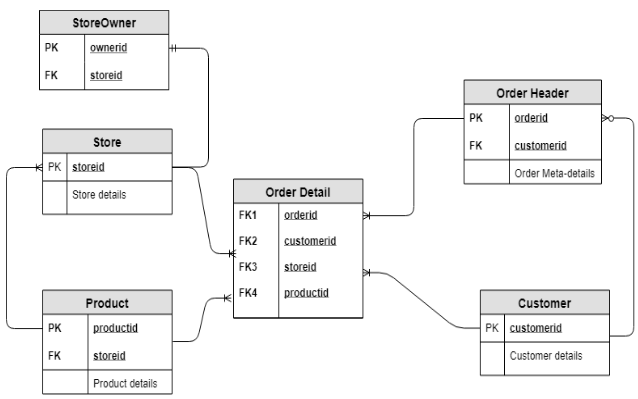

Figure 1: ER Diagram for the SmartEmma core tables

Figure 1 depicts the conceptual model of the data in a star-schema, which is frequently used in data warehousing and OLAP, where Order-Detail serves as a fact table and Customer, Product, Store & Order-Header are fact tables. The benefit of using this schema in the application is to provide OLAP analytics in the BI as well as reduce the number of *read/write* operations in R.S.

Furthermore, the data is first staged in a MongoDB collection and then flattened by a transformation step into multiple MySQL tables. This ensures a *fault tolerance* within the pipeline of the application and parallel processing of data. Thereby allowing the application to process transactional details for the previous business day while injection of current business day details into the system. In the next section, we state the details for incremental item-item collaborative filter to generate the recommendations.

3.2 Item - Item Collaborative Filtering

As described in the Background section, collaborative filtering is used to identify the the likelihood of an item being bought by a customer given that the same item is bought and rated by his peers. In this case, *peers* signify a group of customer that have already bought the given item and have similar market basket as that of the customer. Such a method to produce a rating for an active user based on its similarity with a setset of users and weighting of their rating is also called *neighbourhood-based* or more generally *memory-based* CF. Its algorithm can be summarized into following steps:

1. Based on the similarity of the rating of the active user, assign a weight to all other user as $w_{a,u}$ where *a* denotes an active user and *u* is any other user in the customer base.
2. Selecting a neighboorhood of *K* users who depict the highest similarity with the active user.
3. Generate a score as prediction from a weighted combination of the *neighbooring* user's rating.

In step 1, the weight $w_{a,u}$ signifies the similarity between the users a and u and is commonly computed with a variety of similarity measures such as, *cosine similarity, pearson correlation coffecient, mean squared difference etc*. as presented in [22], [27]. However, as the rating in our system are always positive and orthogonal items are rated with zero for simplicity and lack of item description, *cosine similarity* is used to identify neighboors of the active users, in eq. **2**. Here, the rating of two users, *a* & *u* , can be considered as an *m-dimensional* vector, $\overrightarrow{R_a}$ & $\overrightarrow{R_u}$, and then the similarity is formulated as the cosine of the angle between these vectors.

$$w_{a,u} = \cos(\vec{R}_a, \vec{R}_u) \qquad \text{eq. 2}$$

Finally, in step 3, the prediction for the rating of an active user for an item is computed as the weighted average of deviation from the neighboor's mean ratings(eq. 3). Here, $\overline{r_a}$ defines the average rating of user a and $p_{a,i}$ is the predicted rating of user a for an item i.

$$p_{a,i} = \overline{r_a} + \frac{\sum_{u\in K}\left(r_{u,i} - \overline{r_u}\right) \times w_{a,u}}{\sum_{u\in K} w_{a,u}} \qquad \text{eq. 3}$$

However, when this method is applied to an inventory with large number of user & item, the computational complexity to search for similar users grows extremely large in order to $(n^2 - n)$ where *n* is the number of items/users, as eq. **2** is aggregated over the entire user inventory. Therefore, instead of using traditional neighboorhood-based CF, *item-item* based CF is used as proposed by [4]. In this variation of CF, rather than matching or identifing similarity between the users, only the user's who have already rated similar items as the active user are considered as neighbours. Consequently, the similarity between item *i* and *j*, is the cosine angle between the rating of these items for user's who rated these items (eq. 4) and then rating for item *i* of user *a* can be given by eq. 5. Such a modification within the algorithm results in faster recommender systems and oftenly leads to impoved recommendation, as presented in [4], [28].

$$w_{a,u} = \frac{\sum_{i=1}^{m} r_{a,i} \cdot r_{u,i}}{\sqrt{\sum_{i=1}^{m} r_{a,i}^2} \cdot \sqrt{\sum_{i=1}^{m} r_{u,i}^2}} \qquad \text{eq. 4}$$

$$p_{a,i} = \frac{\sum_{j\in K} r_{a,j}\, w_{i,j}}{\sum_{j\in K}\left|w_{i,j}\right|} \qquad \text{eq. 5}$$

Along with the increasing complexity of operation with growth of inventory sizes, the second major challenge faced in the R.S. of SmartEmma is of "*new-system*", which signifies the absence of any rating of products from the user base. As a result, the average rating of every user would be returned as zero in eq. **2**. To overcome this situation, *default-voting* method is used, as proposed by [29]. Here, the initial rating of a product and store is defined as a function of user's frequency and loyalty, descibed by eq. 6 and eq. 7 respectively.

$$p_{r,a} = \frac{F_p^a \times quantity}{\sum_{i \in U}(F_p^i \times quantity)} \qquad \text{eq. 6}$$

$$s_{r,a} = \frac{F_s^a \times \sum I_s^a}{\sum_{i \in U}(F_s^i \times \sum I_s^i)} \qquad \text{eq. 7}$$

where:

- $p_{r,a}$ & $s_{r,a}$: default rating for product p and store *s* for user *a*,
- F_p^a : scaler value depicting the number of times user *a* bought product *p*,
- F_s^a : the number of times user a visited the store *s*,
- I_s^a : the number of items bought by user *a* from store *s*.

Such a voting scheme tries to simulate a logical rating value based on the shopping and preferences of an item for a user as well as provides an initial point for the R.S. while computing the rating of the item. However, as soon as a user provides his/her actual rating for the given product, this voting scheme for that user-item combination is deactivated, and the actual rating is used instead.

4. Evaluation

Generally, the evaluation of a recommendation is performed by comparing the rating generated by the system either against the actual rating provided by the user or by comparing the error rate of mis-classification with a benchmark system [30], [31]. However, such an evaluation of SmartEmma's RS is not possible as this project refers to the problem of a new system in recommendation where no previous information or historical transaction of user data is made available, thereby making the evaluation of the platform a challenge in itself. Therefore, in the case of SmartEmma, the evaluation of the system is performed on actual data, collected from the alpha-test users of the application.

In this evaluation, the business platform was made available to 10 test users who were tasked with purchasing various products over the platform. The user-base for the testing phase where divided into three categories on the basis of purchasing capacities as follows:

- Small Purchase Group: Orders contain 3 – 5 different products with order amount not exceeding 10 euros.
- Medium Purchase Group: orders containing 10 - 15 different products with order amount not exceeding 30 euros.
- Large Purchase Group: Orders containing 10 - 20 different products with order amount not exceeding 50 euros.

The basic idea of segregating the user base on this criteria is bi-fold:

a) to simulate users purchasing products for short or week long duration
b) to identify the limitation of system over different range of sparcity.

Furthermore, this test group is asked to rate the products they buy between the range of 1 - 5. In this alpha-testing of the platform, 253 transactional orders where placed for 135 products from 14 stores, where a user rating for 7 products was provided.

However, this collection of transactional data was found to be significantly smaller than the actual product base of the application and as a result, the averaging of the weight and similarity score for 95% of the users were significantly small as well. Such a small score of similarity can be attributed to the similarity function used in eq. 2 and initial rating fomula used for default voting in eq. 6 & eq. 7, as it scale of the score is directly propotional to the number of users contributing to the transactional base of the application. Therefore, the evaluation of our R.S. platform requires further extensive testing with a larger pool of beta-test users that can produce an even bigger pool of transactional data. Such a testing phase is planned in the later stage of the project and would be covered in a separate paper.

5. Conclusion and Outlook

This paper presents the use-case of SmartEmma where multiple small-scale retailers collaborated over a digital platform to provide customers with a food and grocery retailing application. The retailer view of the application equips them with a BI platform to provide an overview of their sale and identify products that are highly rated among the users. Such an application not only provides customers an alternative to pre-established grocery applications but also uplifts these retailers to the current platform of e-commerce in the city of Aachen, Germany.

The paper further describes the current alternatives in the field of information retrieval and recommendation and discusses their implementation challenges, and of item-item collaborative filtering in the use-case. Moreover, the current implementation fine-tunes the recommendation with default-voting method to resolve the challenge of new-system, where it provides no historical transaction information initiate the system. Evidently, the extensive testing of R.S in the application would be done with actual data as the user-base and consequently the transaction-base & user rating is expected to surge.

Additionally, based on the further results of the R.S on the beta-testing of the application, a comparative study of current algorithm with other R.S could be done to establish its performance. Furthermore, the item-item CF could be replaced by a *hybrid recommendation system* [32], in order to incorporate the benefits of previously mentioned R.S and obtain the recommendation from majority voting. The idea would be to overcome the challenge of stagnation within the recommendation of products and improve the quality of recommendation even further.

6. REFERENCES

[1] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using Collaborative Filtering to Weave an Information Tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70, Dec. 1992.

[2] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, *Recommender Systems Handbook*, 1st ed. Berlin, Heidelberg: Springer-Verlag, 2010.

[3] P. Melville and V. Sindhwani, "Recommender Systems," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds. Boston, MA: Springer US, 2010, pp. 829–

838.

[4] G. Linden, B. Smith, and J. York, "Amazon.Com Recommendations: Item-to-Item Collaborative Filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan. 2003.

[5] B. Smith and G. Linden, "Two Decades of Recommender Systems at Amazon.Com," *IEEE Internet Comput.*, vol. 21, no. 3, pp. 12–18, May 2017.

[6] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

[7] R. Agrawal, T. Imieliński, and A. Swami, "Mining Association Rules Between Sets of Items in Large Databases," *SIGMOD Rec.*, vol. 22, no. 2, pp. 207–216, Jun. 1993.

[8] X. Amatriain, A. Jaimes, N. Oliver, and J. M. Pujol, "Data Mining Methods for Recommender Systems," in *Recommender Systems Handbook*, 2011, pp. 39–71.

[9] B. Mobasher, R. Cooley, and J. Srivastava, "Automatic Personalization Based on Web Usage Mining," *Commun. ACM*, vol. 43, no. 8, pp. 142–151, Aug. 2000.

[10] R. Cooley, B. Mobasher, and J. Srivastava, "Web mining: information and pattern discovery on the World Wide Web," in *Proceedings Ninth IEEE International Conference on Tools with Artificial Intelligence*, 1997, pp. 558–567.

[11] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," in *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, 1994, pp. 175–186.

[12] J. S. Breese, D. Heckerman, and C. M. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," *CoRR*, vol. abs/1301.7, 2013.

[13] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *Computer (Long. Beach. Calif).*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[14] D. Billsus and M. J. Pazzani, "Learning Collaborative Information Filters," in *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998, pp. 46–54.

[15] F. Niu, B. Recht, C. Re, and S. J. Wright, "HOGWILD!: A Lock-free Approach to Parallelizing Stochastic Gradient Descent," in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, 2011, pp. 693–701.

[16] R. Salakhutdinov and N. Srebro, "Collaborative Filtering in a Non-uniform World: Learning with the Weighted Trace Norm," in *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2*, 2010, pp. 2056–2064.

[17] Y. Koren and R. Bell, "Advances in Collaborative Filtering," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Boston, MA: Springer US, 2011, pp. 145–186.

[18] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural Collaborative Filtering," in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 173–182.

[19] R. Van Meteren, "Using Content-Based Filtering for Recommendation," 2000.

[20] M. J. Pazzani and D. Billsus, "Content-based," P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 325–341.

[21] D. Bridge, M. H. Göker, L. McGinty, and B. Smyth, "Case-based Recommender Systems," *Knowl. Eng. Rev.*, vol. 20, no. 3, pp. 315–320, Sep. 2005.

[22] X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," *Adv. Artif. Intell.*, vol. 2009, p. 4:2--4:2, Jan. 2009.

[23] P. Vassiliadis, A. Simitsis, and E. Baikousi, "A Taxonomy of ETL Activities," in *Proceedings of the ACM Twelfth International Workshop on Data Warehousing and OLAP*, 2009, pp. 25–32.

[24] P. Vassiliadis, A. Simitsis, P. Georgantas, M. Terrovitis, and S. Skiadopoulos, "A Generic and Customizable Framework for the Design of ETL Scenarios," *Inf. Syst.*, vol. 30, no. 7, pp. 492–525, Nov. 2005.

[25] R. A. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 3rd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.

[26] H. Garcia-Molina *et al.*, *Database Systems: A Practical Approach to Design, Implementation, and Management*, vol. 49, no. 4. Pearson, 2010.

[27] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An Algorithmic Framework for Performing Collaborative Filtering," in *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999, pp. 230–237.

[28] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based Collaborative Filtering Recommendation Algorithms," in *Proceedings of the 10th International Conference on World Wide Web*, 2001, pp. 285–295.

[29] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," in *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, 1998, pp. 43–52.

[30] F. Hill, R. Reichart, and A. Korhonen, "SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation."

[31] A. Said and A. Bellog\'\in, "Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks," in *Proceedings of the 8th ACM Conference on Recommender Systems*, 2014, pp. 129–136.

[32] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," *User Model. User-adapt. Interact.*, vol. 12, no. 4, pp. 331–370, Nov. 2002.

Columns on Last Page Should Be Made As Close As Possible to Equal Length

Authors' background

Your Name	Title*	Research Field	Personal website
Atul Mohan	Masters	Data Analysis	
Anas Abdelrazeq	Masters	Data Science	
Frank Hees	Dr. rer. Nat.	Data Science	

***This form helps us to understand your paper better, the form itself will not be published.**

***Title can be chosen from: master student, Phd candidate, assistant professor, lecture, senior lecture, associate professor, full professor**