

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/338614945>

A hybrid recommender system for product sales in a banking environment

Article in *Journal of Banking and Financial Technology* · January 2020

DOI: 10.1007/s42786-019-00014-w

CITATION

1

READS

2,112

2 authors:



[Oladapo Oyebo](#)

Dalhousie University

22 PUBLICATIONS 41 CITATIONS

[SEE PROFILE](#)



[Rita Orji](#)

Dalhousie University

169 PUBLICATIONS 2,135 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Towards Personalizing Persuasive Technologies. [View project](#)



Gas Processing [View project](#)



A hybrid recommender system for product sales in a banking environment

Oladapo Oyeboode¹ · Rita Orji¹

Received: 7 May 2019 / Accepted: 21 December 2019 / Published online: 15 January 2020
© Institute for Development and Research in Banking Technology 2020

Abstract

Recommender systems have been successfully applied in many domains, including in e-commerce and entertainment to boost sales. However, most existing recommender systems employ the collaborative or community-based approach which relies on the preferences or behaviour of other users, in conjunction with that of the target user. This approach may not be applicable to domains such as banking. The two major challenges to using the collaborative approach in the banking domain are: (1) absence of explicit ratings of products by customers for recommendation purpose, (2) cold-start problem which makes it difficult to recommend products to a prospective or new customer who has no preference at all. To tackle the first issue, we develop an algorithm that implicitly infers customer preferences from transaction data. To address the second issue, we propose a hybrid recommender system that combines the item-based collaborative filtering technique (which uses the customer preference data from the algorithm) and the demographic-based approach (which uses customers' demographics). We contribute to knowledge by developing a practical and feasible approach for implementing recommender systems that drive product marketing in the banking sector. We also discuss the performance of this approach based on 393,816 customers dataset. The hybrid approach is applicable to other domains with similar challenges.

Keywords Recommender system · Hybrid approach · Item-based collaborative filtering · Demographic-based technique · Product · Banking · Unsupervised machine learning

1 Introduction

The wide adoption of recommender systems in e-commerce and entertainment domains is aimed at boosting product sales. Amazon and Netflix, for instance, have successfully increased their sales volume and revenues simply by showing users those items that may be of interest to them on their respective sites [2]. This success story indirectly challenges other product-oriented domains, such as banking, to offer personalized recommendations to their customers. Banks offer variety of products to customers, such as savings accounts, checking accounts, investment products (e.g., fixed deposits, call deposits, treasury bills), loans (e.g., mortgage loan, student loan, lease financing, overdrafts, and

others), digital products and services (such as debit cards, credit cards, international fund transfer services, mobile and online banking services, and others), wealth management solutions, currency exchange, private banking products for high net-worth individuals, corporate banking solutions, and so on. Therefore, it is imperative for Banks to sell the right products to customers, and adopting recommender systems to offer personalized recommendations to existing and prospective customers could have a tremendous impact on product sales, which directly influence turnover and income.

Despite the extensive research in the design and evaluation of recommender systems and associated algorithms across various domains [8, 16], there is little knowledge on how to tailor recommender systems to drive product sales in the banking industry. Most recommender systems, including commercial ones and research prototypes, adopt a couple of approaches in providing personalized recommendations. One of the prominent approaches is the collaborative filtering or community-based approach which relies on the preferences or behaviour of other users, in conjunction with that of the target user [8]. Other approaches are content-based,

✉ Oladapo Oyeboode
oladapo.oyebode@dal.ca

Rita Orji
rita.orji@dal.ca

¹ Faculty of Computer Science, Dalhousie University, Halifax, NS, Canada

demographic-based, and knowledge-based approaches [2, 8, 16]. These user preferences are usually explicitly inferred through ratings given to items or by simply liking/disliking the items. In other words, explicit ratings are given by users to express their opinion about an item (such as rating the item on a scale of 1–5). Conversely, implicit ratings are not specified by users but can be determined from user interaction or behavioural data (such as purchases, clicks, product views, etc.). While collaborative filtering approach has proven to be very effective in other domains (such as e-commerce and entertainment) due to the availability of explicit ratings, it is difficult to tailor it to the banking domain since banks generally do not ask customers to rate their products. Another major issue is knowing what to recommend to a prospective or new customer who has not subscribed for or used any banking product. The latter issue is known as the cold-start problem [2, 8, 16]. We addressed these gaps using a hybrid approach that leverages the strengths of collaborative filtering and demographic-based approaches to inform recommender systems that are usable in banking.

Our approach involves five main stages. First, we developed a product rating (PR) algorithm that implicitly infers customer preferences from real-world transaction data. Transaction data has been used by companies and researchers to analyze customer behaviour since it reveals customers' spending patterns [4]. The algorithm generates rating for each product a customer has on the scale of 1–5 based on how often customers use their banking products to carry out personal or business-related transactions. Second, we utilized the ratings dataset (generated by the PR algorithm) to predict ratings for unrated products using the item-based collaborative filtering method [8, 16]. Third, we make use of customer demographics (such as age, gender, marital status, and profession) to group users into clusters using the K-Means unsupervised machine learning algorithm and then generated the cluster-product rating data containing the average of the ratings given to a product by all the customers in that cluster. Fourth, we combine both the predicted and average ratings using a dynamic weighting technique that produces better prediction accuracy. Finally, we recommend top- N unseen products to any given customer (whether new or existing) based on the final ratings.

Our contributions to knowledge are in two folds. First, we developed a practical and feasible approach for implementing recommender systems that drive product sales in the banking sector. We demonstrate the applicability of this approach on real-world datasets and successfully recommended most appropriate and unseen products to target customers. Second, the hybrid approach outperforms other approaches (i.e., item-based collaborative filtering and demographic-based approaches) based on the results of the offline experiments conducted using 415,803 product ratings and demographic data of 393,816 customers.

2 Background

In this section, we described relevant concepts from existing research. We tailored the concepts, including the formulas, to reflect banking terminologies. For instance, most existing research adopt the terms “user” and “item” when discussing recommender systems. Since our target domain is banking, we adopt the terms “customer” and “product” in this paper for clarity.

2.1 Customer-Product rating matrix

Given a set of customers $C = \{c_1, c_2, \dots, c_n\}$ and a set of products $P = \{p_1, p_2, \dots, p_m\}$, where n and m represent number of customers and number of products respectively. The customer-product rating matrix is a $m \times n$ matrix such that r_{ij} is the rating assigned to product p_i by customer c_j , where $1 \leq i \leq m$ and $1 \leq j \leq n$. The rating is within a numerical scale of 1 to 5. Since customers do not explicitly rate products in banking, we infer the ratings from customer's transaction data.

2.2 Item-based collaborative filtering

The collaborative filtering approach can either be neighbourhood-based (also called memory-based) or model-based [16, 18]. The neighbourhood-based method makes use of the customer-product rating matrix to predict ratings for unrated products. Item-based collaborative filtering is one of the neighbourhood-based methods. For a customer c_i , the item-based collaborative filtering algorithm predicts the rating for a product p_i based on c_i 's ratings for similar products. Two products p_i and p_j are said to be similar if they are rated in a similar way by several customers [16]. In this paper, we applied the Cosine Similarity measure for computing the similarity between two products [8].

2.2.1 Cosine similarity measure

The cosine similarity measure is a standard metric, and has been shown to produce the most accurate results when used with the item-based collaborative filtering approach [8]. This metric measures the similarity between two n -dimensional vectors based on the angle between them [8]. Given that n represents the number of customers, the similarity between two products p and q (where p and q correspond to n -dimensional rating vectors \vec{p} and \vec{q}) can be formally defined as follows:

$$\text{sim}(p, q) = \cos(\vec{p}, \vec{q}) = \frac{\vec{p} \cdot \vec{q}}{|\vec{p}| \times |\vec{q}|} = \frac{\sum_{c \in C} r_{cp} r_{cq}}{\sqrt{\sum_{c \in C} r_{cp}^2} \times \sqrt{\sum_{c \in C} r_{cq}^2}} \quad (1)$$

where C is a set of n customers, and r_{cp} and r_{cq} denotes each customer c 's rating for product p and q respectively. The

output of the cosine similarity is a value between 0 and 1. A value of 0 means there is no similarity between the two products. Thus, two products are similar if the similarity value between them is greater than 0. A value close to 1 indicates strong similarity.

2.2.2 Predicting ratings

To predict customer c 's rating for product p , we compute the weighted average of c 's ratings for products in \hat{S} , where \hat{S} denotes a set of products similar to p and rated by c [16]. Formally,

$$pred(c, p) = \frac{\sum_{m \in \hat{S}} sim(m, p) \times r_{cm}}{\sum_{m \in \hat{S}} sim(m, p)} \quad (2)$$

where r_{cm} represents c 's rating for each product m in \hat{S} , and $sim(m, p)$ is the similarity value between m and p .

Research has shown that most of the commercial recommender systems implement the item-based collaborative filtering approach because it supports offline preprocessing of data (such as precomputing the similarity between every pair of products to form product similarity matrix) without losing accuracy [8]. As a result, real-time prediction is possible even for very large rating matrix [8]. The downside of using only the item-based collaborative filtering approach for recommendations is that of the cold-start problem where no product is recommended to any customer without ratings or preferences.

2.3 Demographic-based approach

In this approach, the demographic profile of customers is considered in computing product ratings. This consideration is based on the fact that demographics contribute to differences in people's tastes or preferences [16]. The demographic-based approach, as proposed by Gupta et al. [6], involves three steps:

1. Transform textual demographic features into numeric form.
2. Partition customers into k clusters or groups using an unsupervised machine learning algorithm such as the K-Means clustering algorithm. K in "K-Means" refers to the number of centroids which, in turn, depends on the value of k . The optimal value of k can be determined through the popular Elbow method [10].
3. Given a set of k clusters $G = \{g_1, g_2, \dots, g_k\}$ and a set of n products $P = \{p_1, p_2, \dots, p_n\}$. Create a cluster-product rating matrix such that r_{ij} is the average rating of customers in cluster g_i for product p_j .

A prospective customer, whose demographics is known, is assigned to a cluster with minimum Euclidean distance to the centroid. The benefit of the demographic-based approach is that a prospective customer without preferences or ratings is able to receive product recommendations using his demographic profile, thereby solving the cold-start problem. The demographic-based approach also supports real-time prediction since both steps 2 and 3 are possible offline while new customers are simply added to existing clusters online.

However, demographic data alone is not enough to achieve the level of personalization required in banking. Information about customer behaviour or preferences must also be considered in order to provide more personalized product recommendations. Since the item-based collaborative filtering approach already considers customer preferences, combining the two approaches to form a hybrid approach is the best option.

2.4 Root mean square error (RMSE)

The root mean square error (RMSE) is one of the widely used metrics for measuring the performance or accuracy of recommender systems [8, 16]. The RMSE compares the predicted ratings with the actual and penalizes for larger errors. The popular Netflix recommender system challenge measured performance based on the RMSE [8, 16]. The RMSE formula is given below.

$$RMSE = \sqrt{\frac{\sum_{r_{cp} \in D_{test}} (predictedRating(c, p) - r_{cp})^2}{N}} \quad (3)$$

where D_{test} is the test set used for evaluation, r_{cp} is the actual rating given by customer c to product p in the test set, and $predictedRating(c, p)$ is a function that accepts c and p as parameters and then returns predicted rating for p . N is the number of entries in the test set.

3 Related work

Majority of existing research in recommender systems are targeted at domains, such as e-commerce, entertainment, news media, and others, but less research efforts are seen in banking. For example, Gallego et al. [5] is one of such efforts that partnered with a bank in developing a context-aware recommender system. In their work, they developed a mobile recommender system to recommend places of interest using customer's context information, such as location, current activity and time, credit card purchases, and demographic segmentation. However, their focus is on recommending places of interest (such as restaurants), which differs from banking products, and

the approach they used in implementing the system is not easily adaptable to other tasks, such as recommending banking products.

Another work applied collaborative filtering technique on stock data and user transactions to recommend which company's stock is worth buying for investment purpose [19]. The challenge of this approach is the cold-start problem because users who are new to stock trading (i.e., without past transactions) will not be able to receive recommendations from such system.

Mitra et al. [12] proposed a hybrid recommender system for the insurance domain. Their approach adopts both the preference-based model (which is synonymous to collaborative filtering technique) and the attribute-based model (which is the content-based approach) to recommend insurance policies and riders. Depending on the type of insurance products, their system chooses the appropriate technique to make recommendations. Their proposed approach may not be fit for recommending banking products since it relies on explicit user ratings which is not available in banking. Also, demographic profiles of customers, which would have made the recommendations more personalized, are not considered in their approach.

Zhao et al. [24] developed a demographic-based recommender system that infers users' purchase intents and the product of interest from tweets, extracts the demographics of the various users from their public profiles, detects the demographics of the target consumer from online reviews, and then use the data collected to measure similarity between users and products in order to recommend relevant products to the target user. While this approach demonstrates the significance of demographics to achieving personalized recommendations, it is more suitable to e-commerce than banking.

Vozalis et al. [21], Pazzani [14], and Gupta et al. [6] presented their hybrid frameworks which make use of demographics to enhance recommendations. Gupta et al. applies the clustering technique in finding similarity among users by placing them into clusters or groups based on demographics [6]. We adopt Gupta et al.'s approach to handling demographics since clustering has been proven to be effective in detecting similarity contained in data. However, their approach to combining the multiple approaches in generating recommendations is more tailored to their domain of application, and not directly applicable to a domain such as banking.

Other hybrid techniques combined collaborative filtering and knowledge-based approaches [15], collaborative and content-based approaches [1, 20, 23], as well as collaborative and association rule mining approaches [13].

4 The hybrid approach

Figure 1 shows the architecture of our hybrid recommender system. There are two main datasets used in building and evaluating our system: transaction data and demographic data. We describe both datasets, including datasets derived from them, in Sect. 5.

The main stage of the architecture is the hybridization stage where we combined *predicted ratings* from item-based collaborative filtering approach and *average ratings* from the demographic-based approach. For a target customer, each product's predicted rating and the corresponding average rating are combined in a weighted fashion such that the sum of the weights is 1. We discuss our dynamic weighting method in Sect. 4.1.

4.1 The dynamic weighting method

Since our goal is to recommend products to both existing and prospective/new customers, we adopt a hybrid approach that combines the predicted ratings and average ratings in a weighted fashion for a target customer c (on a per-product basis), as shown in the formula below.

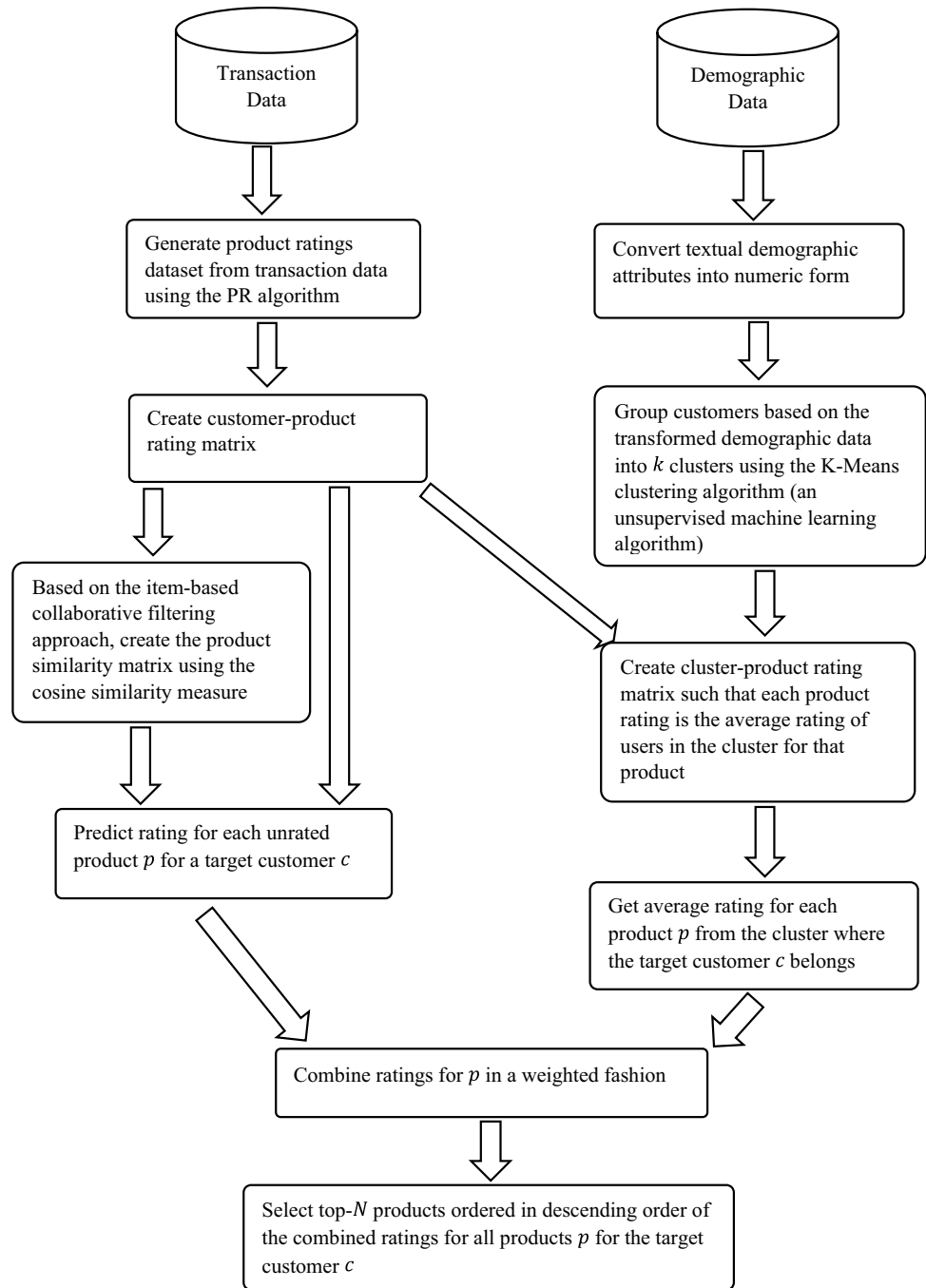
$$finalRating(c, p) = \alpha \times pred(c, p) + \beta \times avgR(g_c, p) \quad (4)$$

where α and β are weights, $\beta = 1 - \alpha$, $pred(c, p)$ is the predicted rating for product p , and $avgR(g_c, p)$ is the average rating of customers in cluster g (where c belongs) for product p .

Existing research adopted different methods for determining α and β [3, 11, 22]. The idea is to come up with optimal values of α and β that improves prediction accuracy significantly. We leverage their works to come up with a suitable weighting method.

In our product ratings dataset (discussed in Sect. 5.1), majority of the customers have 1–3 products (which seems to reflect common practice in the banking domain). This creates sparsity in our customer-product rating matrix and causes the similarity value or score between each pair of products to be low. As customers subscribe to more products, the count of ratings increases and the similarity scores receive a gradual boost as well. We, therefore, propose a *per-product* (hence dynamic) weighting method that gradually increases the value of α as products similar to the target product receive more ratings. We set the initial or base value of α to 0.3, based on the results of the experiment conducted while varying the value of α (as described in Sect. 6.3.1). Formally, we define the value of α as follows:

$$\alpha = 0.3 + \left(\frac{\sum_{m \in S_p} count(r_m)}{count(C) \times count(S_p)} \right) - b \quad (5)$$

Fig. 1 System architecture

S_p is the set of products similar to the target product p , $\text{count}(r_m)$ is the number of ratings for each m in S_p , $\text{count}(C)$ is the number of customers in the customer-product rating matrix, $\text{count}(S_p)$ is number of products in S_p , and b is a constant used for adjusting α such that α does not exceed 1. Thus, if α is less than or equal to 1, the value of b is 0. Moreover, b may be deemed negligible as α 's increase is gradual and unlikely to exceed 1 since sparsity will always exist. The default value of b is 0. The

dynamic weighting method's efficiency is tested in the second experiment (see Sect. 6.3.2).

For a prospective or new customer without preferences or ratings, α is set to 0, so β becomes 1.

5 Dataset

We used transaction data and demographic data of 393,816 customers (of a Bank) for our experiment. The transaction data contains the following attributes: *account number*, *customer number*, *product number*, *transaction type*, *transaction amount*, *transaction date*, *transaction status*, *transaction currency*, and so on. We anonymized both customer number and product number for privacy reasons. To determine how often each customer uses a product, we extracted (from the transaction data) the last transaction date per product for each customer. The extracted data (called product usage data) contains the following attributes: *customer number*, *product number*, and *last transaction date*. This product usage data is used in determining the product ratings, as described in Sect. 5.1. Furthermore, the demographic data contains the following attributes: *customer number*, *age*, *gender*, *marital status*, and *profession*. The customer number was anonymized to protect customers' identity.

5.1 The product ratings dataset

To generate implicit ratings for the products, we applied the product rating (PR) algorithm on the product usage data. The algorithm, a version of which is shown in Table 1 and applied in this work, is based on the relationship between product usage and customer preferences. In other words, if a customer uses a specific product more often, it means he/she is satisfied with that product which, in turn, will be assigned a higher rating implicitly. Thus, the PR algorithm rates a product on a numeric scale of 1–5 per customer based on the number of days the product is not used (i.e., inactive days). A rating of 1 represents low product usage, while five represents high product usage. Table 2 shows an example of the product ratings dataset generated by the algorithm. The PR algorithm can be modified to suit the need of any Bank that will use our system. For instance, a Bank can decide the criteria that determine which rating should be assigned to a product based on the number of inactive days, and can

Table 1 Product Rating (PR) Algorithm: Generate ratings based on Product Usage

```

PR(transaction_data)
begin
    initialize rating_data to empty
    for each customer_number c in transaction_data do
        set product_usage_data to rows in transaction_data for c
        for each product p in product_usage_data do
            set last_usage_date to p's last_transaction_date
            set current_date to today's date
            inactive_days = current_date - last_usage_date
            if inactive_days between 0 and 60
                rating = 5
            else if inactive_days between 61 and 150
                rating = 4
            else if inactive_days between 151 and 240
                rating = 3
            else if inactive_days between 241 and 360
                rating = 2
            else
                rating = 1
            append <c, p, rating> to rating_data
        end for
    end for
end;

```

Table 2 Product ratings dataset (sample)

Customer no.	Product	Rating
3882991	1002	5
3882991	1001	4
3882991	1400	3
6637742	1002	4
6637742	1900	4
6637742	1001	5
5424423	1003	3
5424423	1400	2
8846657	1002	5
8846657	1803	4
8846657	1003	4
3524435	1002	1
3524435	1400	4

Table 3 Demographic data (before transformation)

Customer no.	Age	Gender	Marital status	Profession
3882991	25	Male	Single	Student
6637742	40	Male	Married	Farming
5424423	52	Female	Married	Banking
8846657	32	Male	Single	Clergy
3524435	30	Female	Married	Student

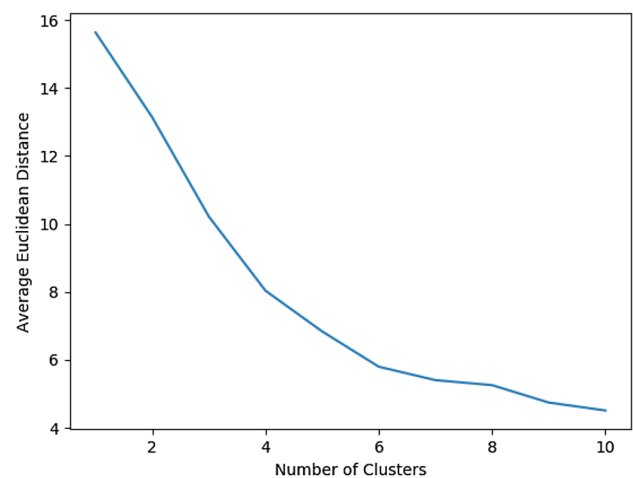
Table 4 Demographic data (after transformation)

Customer no.	Age	Gender	Marital status	Profession
3882991	25	1	1	1
6637742	40	1	2	2
5424423	52	2	2	3
8846657	32	1	1	4
3524435	30	2	2	1

also consider other parameters such as average balance and transaction volume.

5.2 Demographic data

The demographic data contains the customer number, age, gender, marital status, and profession. We converted the gender, marital status, and profession values to numeric form. For instance, we assigned 1 to male and 2 to female. Tables 3 and Table 4 shows a sample demographic data before and after transformation respectively.

**Fig. 2** Detecting the appropriate number of clusters k using Elbow method

5.3 Derived datasets

Prior to predicting ratings for products per customer using our hybrid approach, we created additional datasets required for prediction to take place.

5.3.1 Customer-Product rating matrix

From our product ratings dataset, we generate the customer-product rating matrix (described in Sect. 2.1). The row labels represent unique products and the column labels are unique customers.

5.3.2 Product similarity matrix

Based on the item-based collaborative filtering approach described in Sect. 2.2, we compute the similarity between every pair of products in our product ratings dataset using the cosine similarity measure. Prior to the computation, we set all unrated products to 0 to prevent null values. The result is an $n \times n$ product similarity matrix, where n is the number of products.

5.3.3 Cluster-Product rating matrix

Prior to generating the cluster-product similarity matrix, we partition customers into clusters based on the similarity in their demographic data. To create the clusters, we used the K-Means clustering algorithm which is an unsupervised machine learning algorithm. In determining the appropriate number of clusters k , we applied the Elbow

method [10]. As shown in Fig. 2, the appropriate value for k is 4, which is the elbow point.

Afterwards, we ran the K-Means clustering algorithm on the demographic dataset having specified the number of clusters as 4. To visualize the clusters on a two-dimensional space, we reduced the dimensionality of the data using the Principal Component Analysis (PCA) technique [9]. Figure 3 is the visual representation of the 4 clusters. The clusters were differentiated using distinct colour schemes.

We generate the cluster-product rating matrix by computing the average rating of customers in a cluster for each product.

6 Implementation and evaluation

We implement the hybrid recommender system, based on the system architecture described in Sect. 4, using the Python programming language and execute it on a Windows PC with an installed RAM of 16 GB and 2.60 GHz dual core processor. We generate the product ratings dataset (using the PR algorithm), which contains 415,803 entries.

Afterwards, we run few experiments to verify the performance of the system in terms of prediction accuracy. We discuss the experimental design, evaluation metrics, the experimental results, and the top- N recommendations in subsequent sections.

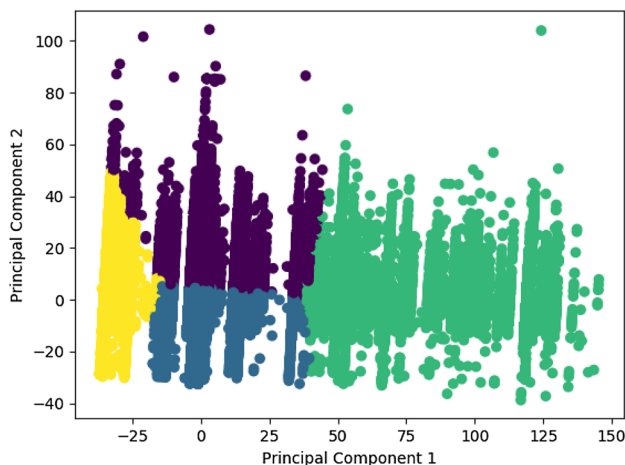


Fig. 3 K-means clustering: visualizing the 4 Clusters

Table 5 Number of entries in each test set used for the experiments

Test set 1	Test set 2	Test set 3	Test set 4	Test set 5	Test set 6	Total
393	393	392	392	392	392	2354

6.1 Experimental design

To evaluate the performance of our hybrid recommender system, we conduct experiments during which we compare the prediction accuracy of our hybrid approach with the prediction accuracy of item-based approach and demographic-based approach respectively. Prior to the experiments, we partitioned our product ratings dataset into training set and test set, and then generated the derived datasets (i.e., customer-product rating matrix, product similarity matrix, and the cluster-product rating matrix) based on the training set. To form the test set, we identified customers with at least three products in the product ratings dataset and randomly picked one of their products and the corresponding rating. In total, we have 413,449 and 2,354 entries in the training set and test set respectively. We divided the test set into 6 unique test sets such that each test set has different entries compared to the other. In other words, the $(customer, product, rating)$ tuple in each test set is different. Table 5 shows the number of entries in each test set. We discussed our experiments, the test set(s) they used, and the results in Sect. 6.3.

6.2 Evaluation metrics

We measured prediction accuracy using the Root Mean Square Error (RMSE), described in Sect. 2.4. The RMSE is one of the most popular and standard metrics for evaluating recommender systems. The lower the RMSE, the better the prediction accuracy.

6.3 Experimental results

In this section, we discuss our experiments and the corresponding results. The experiments involve generating prediction accuracy using RMSE for the three approaches and comparing them. Two experiments were conducted:

1. Using static or fixed weighting technique for the hybrid approach, and comparing the result with the item-based and demographic-based approaches.
2. Using the dynamic weighting method (discussed in Sect. 4.1) for the hybrid approach, and comparing the result with other approaches.

6.3.1 Experiment using static or fixed weighting technique

In this section, we demonstrate the impact or significance of the two weights (α and β) on the hybrid approach's prediction

accuracy. This experiment also helps us to determine the optimal value of α which is then used as the initial or base value of α in our dynamic weighting method (discussed in Sect. 4.1).

We set the weights to fixed values and observe the prediction accuracy of the hybrid approach in comparison with the other two approaches. We perform 9 iterations in this experiment, and we set each value of α to 0.1, 0.2, 0.3, ..., 0.9 respectively to match the number of iterations. In other words, the value of α is 0.1 for the first iteration, 0.2 for the second iteration, 0.3 for the third iteration, 0.4 for the fourth iteration, and so on. The last value of α will be 0.9 after which the iteration stops. Note that the value of β also changes for each value of α since $\beta = 1 - \alpha$.

We observe the prediction accuracy of the hybrid approach for each iteration. We used test set 1 for this experiment. While the RMSE of both item-based and demographic-based approaches remain at 1.988 and 1.748 respectively, the RMSE of the hybrid approach varies in response to changes in the values of α and β , as shown in Table 6 and Fig. 4. From this experiment, the hybrid approach performs better than the demographic-based approach when α is between 0.1 and 0.6. Also, the hybrid approach performs better than the item-based approach for all values of α .

We also observe that the hybrid approach has its lowest RMSE of 1.673 and the best prediction accuracy when α is 0.3. This result informed our decision to set the initial or base value of α in our dynamic weighting method to 0.3.

In the next experiment, we evaluate the hybrid approach using our dynamic weighting method and observe its performance.

6.3.2 Experiment using the dynamic weighting method

The aim of this experiment is to confirm if our hybrid approach will consistently outperform the other two approaches in terms of prediction accuracy. To achieve this, we perform 5 iterations using the remaining five test sets

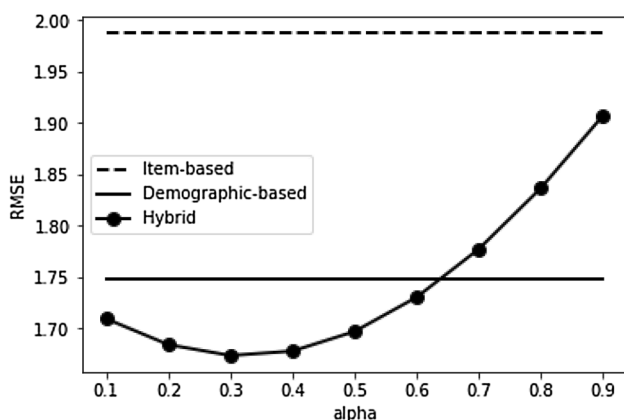


Fig. 4 RMSE of the three approaches as α changes

Table 6 RMSE of the three approaches for each value of α

α	RMSE		
	Hybrid	Item-based	Demographic-based
0.1	1.709	1.988	1.748
0.2	1.684	1.988	1.748
0.3	1.673	1.988	1.748
0.4	1.678	1.988	1.748
0.5	1.697	1.988	1.748
0.6	1.730	1.988	1.748
0.7	1.777	1.988	1.748
0.8	1.836	1.988	1.748
0.9	1.907	1.988	1.748

(i.e., test set 2 to test set 6), one for each iteration. Hence, we use different test set for every iteration. We also observe and compare the RMSE and the Mean Absolute Error (MAE) of the three approaches in this experiment.

As shown in Table 7 and Fig. 5, the hybrid approach using dynamic weighting method consistently achieves a higher prediction accuracy across the five test sets as shown by the lowest RMSE values when compared to item-based and demographic-based approaches. Also, the MAE of the hybrid approach is lower than the corresponding RMSE values. However, the item-based approach achieved the lowest MAE values compared to the other two approaches. This variation between RMSE and MAE suggests why RMSE is more appropriate for measuring prediction accuracy since it penalizes for large errors (such as predicting 1 when actual rating is 5) and does not use absolute values [7, 17]. On the other hand, MAE uses absolute values and assigns equal weights to large and small errors, hence may be inappropriate for evaluating our hybrid system.

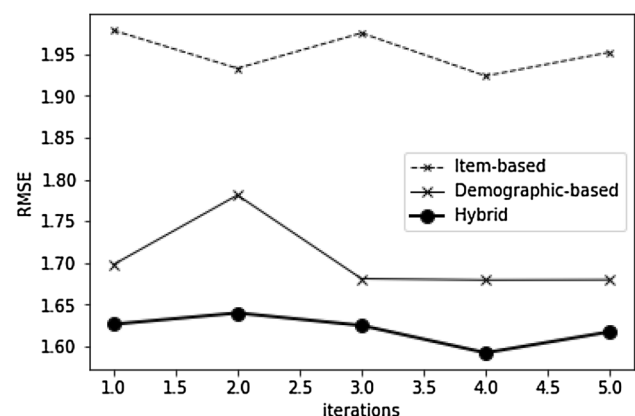


Fig. 5 RMSE of the three approaches for each iteration

Table 7 RMSE and MAE of the three approaches for each iteration

Iteration	RMSE			MAE		
	Hybrid	Item-based	Demographic-based	Hybrid	Item-based	Demographic-based
1	1.626	1.979	1.698	1.576	1.459	1.716
2	1.640	1.933	1.781	1.529	1.350	1.731
3	1.625	1.975	1.681	1.564	1.410	1.754
4	1.592	1.924	1.679	1.532	1.332	1.724
5	1.617	1.952	1.680	1.547	1.437	1.687

6.4 Top- N recommendations

To recommend products for a target customer, the hybrid recommender system checks if the customer has preferences using the customer-product rating matrix. If the customer does, then the system applies the dynamic weighting method to combine the predicted rating and average rating using the item-based and demographic-based approaches respectively, as discussed in Sect. 4.1, for each unseen product. The system then sorts the combined ratings in descending order and recommends the top- N unseen products to the customer. If N is 5, for instance, the system recommends top-5 unseen products to the customer.

On the other hand, if the target customer is a prospective customer, the system generates a unique identifier for that customer and then assigns him to cluster with the least Euclidean distance to the centroid based on his demographic information. Since the customer has no preferences or ratings, the value of α is set to 0 and the value of β becomes 1 (as stated also in Sect. 4.1). Hence, the system applies only the demographic-based approach and retrieves the average rating for each unseen product from the cluster-product rating matrix. The system then sorts the average ratings in descending order and recommends the top- N unseen products to the customer.

Furthermore, the optimal value of N depends on the number of available products in the dataset. While the default value of N can be set to 2 since every bank offers up to five or more products, we allow the value N to be controlled by the user of the proposed system. For instance, if a system user (e.g., product marketer) wants to see the top 4 products to offer his/her customers, he/she will simply set the value of N to 4.

7 Discussion

From our results, it is evident that our hybrid recommender system can provide personalized product recommendations for both prospective and existing customers with better

prediction accuracy than item-based and demographic-based approaches.

The significance of these results is that our hybrid approach has successfully harnessed the strengths of both approaches to provide a better and scalable approach that can be applied even in banks with millions of customers. For instance, the item-based collaborative approach, which is now part of our hybrid system, is mostly used in building commercial recommender systems because of its scalability and the ability to provide real-time recommendations. Furthermore, the item-based approach is built upon customer behaviour which makes it suitable for the banking domain since transaction data is the richest source of customer behaviour in the banking sector and various patterns can be extracted from it. One of the patterns is the product activity we leveraged in defining our product ratings dataset. We can combine multiple patterns to define ratings that strongly reflects customer behaviour by simply modifying the PR algorithm. The other component of our hybrid approach, the demographic-based approach, is also suitable to banking since customers' age, interest, financial status, geographic location, lifestyle, social status can largely influence the products they buy. The demographic-based approach is also scalable and supports real-time computations as well. Hence, we have built a recommender system that is scalable, practical, feasible, and capable of revolutionizing product sales in the banking sector. Future work will perform online experiments in a banking environment. Feedbacks received from selected customers regarding the products the system recommends for them will be used to further enhance the quality of the system.

Our system can also be applied to other domains outside banking with similar problem of lack of explicit customer ratings and cold-start issue. For instance, in the insurance domain, implicit ratings can be defined based on the buying pattern of insurance products by customers, and the PR algorithm can be modified to generate those ratings. Once our system has access to both the ratings data and the customers' demographic data, it can generate personalized product recommendations for both prospective and existing customers of insurance companies.

8 Conclusion and future work

We presented our hybrid approach to generating personalized recommendations to drive product sales in the banking sector. This approach proposed an algorithm for generating implicit ratings from transaction data, and also proposed a dynamic weighting method that combines predicted ratings from item-based collaborative filtering and demographic-based approaches with the aim of improving prediction accuracy. Our results revealed that our hybrid approach consistently achieves higher prediction accuracy than the other two approaches across various datasets, while harnessing the strengths of both approaches in providing personalized recommendations for both existing and prospective customers.

As part of our future work, we plan to perform online experiments with bank customers and record their feedbacks regarding the products recommended to them. Based on customer feedback, we will evaluate the performance of the system in terms of the quality of recommended products using precision, recall and F1 metrics [7]. Furthermore, we intend to compare our approach with existing approaches such as Matrix Factorization and Deep Learning techniques. Also, we will include additional datasets, such as social data, psychographic data, and geographic data to further enhance the system so that product recommendations can be more personalized. Finally, we plan to extend and validate our hybrid approach in domains other than banking.

References

- Aerts L, Claes A, Leeuw D de, Leeuw E de (2019) Implicit hybrid recommender system for informative articles. In: Seminar case studies in business analytics and quantitative marketing (FEM21001), pp 1–52. https://www.researchgate.net/publication/n331716098_Implicit_Hybrid_Recommender_System_for_Informative_Articles
- Aggarwal CC (2016) Recommender systems. Springer, Switzerland. <https://doi.org/10.1007/978-3-319-29659-3>
- Claypool M, Gokhale A, Miranda T, Murnikov P, Netes D, Sartin M (1999) Combining content-based and collaborative filters in an online newspaper. In: Proceedings of the ACM SIGIR '99 workshop on recommender systems: algorithms and evaluation. <https://www.semanticscholar.org/paper/Combining-Content-Based-and-Collaborative-Filters-Claypool-Gokhale/21cfe0b5fbd4a7bd258d25255f6bfce4dca3306>
- EMC Corporation (2013) Banking on customer behaviour. <https://docplayer.net/13040426-Banking-on-customer-behavior.html>
- Gallego D, Huecas G (2012) An empirical case of a context-aware mobile recommender system in a banking environment. In: 2012 third FTRA international conference on mobile, ubiquitous, and intelligent computing. IEEE, Vancouver, pp 13–20. <https://doi.org/10.1109/MUSIC.2012.11>
- Gupta J, Gadge J (2014) A framework for a recommendation system based on collaborative filtering and demographics. In: 2014 international conference on circuits, systems, communication and information technology applications (CSCITA). IEEE, Mumbai, pp 300–304. <https://doi.org/10.1109/CSCITA.2014.6839276>
- Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. *ACM Trans Inf Syst* 22:5–53. <https://doi.org/10.1145/963770.963772>
- Jannach D, Zanker M, Felfernig A, Friedrich G (2011) Recommender systems: an introduction. Cambridge University Press, New York
- Jolliffe IT (2002) Principal component analysis. Springer, New York. <https://doi.org/10.1007/b98835>
- Kodinariya TM, Makwana PR (2013) Review on determining number of Cluster in K-Means clustering. *Int J Adv Res Comput Sci Manag Stud* 1:2321–2782
- Larkey LS, Croft WB (1996) Combining classifiers in text categorization. 19th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, New York, pp 289–297
- Mitra S, Chaudhari N, Patwardhan B (2014) Leveraging hybrid recommendation system in insurance domain. *Int J Eng Comput Sci* 3:8988–8992
- Passi R, Jain S, Singh PK (2019) Hybrid approach for recommendation system. In: 2nd international conference on data engineering and communication technology. Springer, Singapore, pp 117–128. https://doi.org/10.1007/978-981-13-1610-4_12
- Pazzani MJ (1999) Framework for collaborative, content-based and demographic filtering. *Artif Intell Rev* 13:393–408. <https://doi.org/10.1023/A:1006544522159>
- Rahman MM, Zaki ZBM, Alwi NIBM, Monirul Islam M (2019) A hybrid approach to improve recommendation system in E-tourism. In: Emerging technologies in data mining and information security. Springer, Singapore, pp 787–797. https://doi.org/10.1007/978-981-13-1951-8_70
- Ricci F, Rokach L, Shapira B (2015) Recommender systems handbook. Springer, New York. <https://doi.org/10.1007/978-1-4899-7637-6>
- Salam Z, Najafi S (2016) Evaluating prediction accuracy for collaborative filtering algorithms in recommender systems. KTH Royal Institute of Technology, Stockholm. <http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A927356&dsid=537>
- Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: 10th international conference on World Wide Web (WWW '01). ACM, New York, pp 285–295. <https://doi.org/10.1145/371920.372071>
- Sayed FR, Argididi RV, Apte SS (2013) Generating recommendations for stock market using collaborative filtering. *Int J Comput Eng Sci* 3:46–49
- Smith J, Weeks D, Jacob M, Freeman J, Magerko B (2019) Towards a hybrid recommendation system for a sound library. In: ACM IUI workshops. Los Angeles, USA. <http://ceur-ws.org/Vol-2327/IUI19WS-MILC-5.pdf>
- Vozalis M, Margaritis KG (2004) Collaborative filtering enhanced by demographic correlation. In: AIAI symposium on professional practice in AI, part of the 18th World Computer Congress. <https://pdfs.semanticscholar.org/989a/4873e11df7e20cd856c618fce1828e85b489.pdf>
- Xie F, Xu M, Chen Z (2012) RBRA: A simple and efficient rating-based recommender algorithm to cope with sparsity in recommender systems. In: Proceedings of 26th IEEE international conference on advanced information networking and applications workshops, WAINA, pp 306–311. <https://doi.org/10.1109/WAINA.2012.11>
- Ye BK, Tu YJ, Liang TP (2019) A hybrid system for personalized content recommendation. *J Electron Commer Res* 20:91–104
- Zhao XW, Guo Y, He Y, Jiang H, Wu Y, Li X (2014) We know what you want to buy: a demographic-based system for product recommendation on microblogs. *Proc ACM SIGKDD Int Conf Knowl Discov Data Min.* <https://doi.org/10.1145/2623330.2623351>