2023 Spring: Product Recommendation System Using NLP
Team 23

19201018 Sadiq Uddin Bhuiyan 19341002 Jubaer Bin Islam
20301054 Shammo Faiyaz Ahmed 20341032 Mir Jadir Siddique
Md Farhadul Islam Md Sabbir Hossain Annajiat Alim Rasel

29th April 2023

Contents

1 Introduction 2

2 Literature Reviews 2
- 2.1 An Innovative Recommender System for eCommerce Websites 3
- 2.2 Sentiment Analysis in Product Reviews 3
- 2.3 Improving Relevance Quality in Product Search using Semantic Similarity . 4
- 2.4 Personalized Recommendations of Products to Users 4

3 Proposed System 5
- 3.1 Model Overview 5
- 3.2 Experiment 5
  - 3.2.1 User Profile 5
  - 3.2.2 Group Rating 6
  - 3.2.3 Similarity Computation 6
  - 3.2.4 Collaborative Prediction 7
  - 3.2.5 New User Predictions 7

4 Result Analysis 7
- 4.1 Data-Set & User Profile 7
- 4.2 Features of our System 7
  - 4.2.1 Combination Coefficient & Number of Clusters 7
  - 4.2.2 Methods for Computing User-User Similarity 8
  - 4.2.3 Neighborhood Size & Construction of User Profiles . . . 8
- 4.3 New User Problem 9
- 4.4 Comparison with Other Works 9

5 Limitations 10

6 Conclusion 10

7 References 11

Abstract

With the continuous development of eCommerce infrastructures, product recommendation systems have become very popular and widely-used in multiple organizations. The purpose is to direct users to products and/or services that might interest them and best meet their preferences. This is demonstrated in eCommerce websites such as Amazon, Daraz, Alibaba, streaming platforms such as Netflix, YouTube, and even social media websites like Facebook, Instagram. To perform appropriate recommendations, two fundamental techniques in NLP are used; Collaborative Filtering (CF), which combines the preferences of other users to make a prediction for the target user and Content-Based Filtering (CBF), which analyses information based on semantic content, using sentiment analysis and comprehension of reviews. In this paper, our goal is to formulate an unsupervised machine and deep learning strategy to combine both these approaches to maximize on their strengths, and dispose of respective drawbacks. We will demonstrate a series of recommendations by studying the techniques for calculating similarities between users based on the information extracted from user profile, user history and user reviews. The objective here is to devise a combination model and evaluate it to demonstrate the effectiveness in providing appropriate recommendations.

1 Introduction

Product recommendations are illustrated when a website delivers a series of personalized product suggestions based on each visitor's profile and browsing history. Recommendation systems are utilized in numerous contexts, such as e-commerce and online advertising. Online firms must sift through millions of pieces of data to present clients with personalized recommendations. It has been demonstrated through study that recommendation systems offer Internet companies and their customers enormous benefits [1]. The engines employed by international technological companies such as YouTube, Netflix, and Amazon are examples of extremely efficient recommendation systems. These systems process vast amounts of information at multiple phases, including the training stages [2].

Search is one of the essential techniques customers use to seek products in e-commerce; hence, it is essential to verify the relevancy and integrity of search results. A search result may be deemed to have poor relevance quality, also known as a search fault, if it does not match the query intent of the user. The severity of these flaws can range from a simple mismatch in brand or color to an entirely irrelevant result of a different product category [3]. It is crucial to address search faults since they can degrade client faith and perception of the e-commerce system, as well as affect the possibility to sell products.

We will investigate the distinctions between content-based and collaborative filtering systems in further depth. Content-based systems generate suggestions based on the user's purchase and preference history. Such algorithms reason that it is highly possible that a user who has previously purchased an item will do so again. Often, these items are grouped according to their features. Collaborative filtering is currently one of the most often used techniques that typically produces better results than the aforementioned content-based recommendation system. It uses user transactions to identify items that may interest them [6].

2 Literature Reviews

Efficiency of recommender systems and their elements have been heavily researched on in the past, thus there are various methodologies that can be used to explore such unique approaches. The subsections below are paper reviews which best contextualizes our model for further comparison and evaluation.

2.1 An Innovative Recommender System for eCommerce Websites

An innovative approach for a recommendation system in e-commerce websites using natural language processing (NLP) has been proposed here [6]. The main objective here is to improve the user experience by providing personalized recommendations based on the user's natural language input.

The proposed system consists of 2 main components: a recommender system (RS) and a sentiment analysis system [6]. The recommender system (RS) is trained via many characteristic data, interactions, transactions, etc. Different algorithms (content-based, model-based, and memory-based) are put into effect here to determine the pattern of the user's inputs [6]. Additionally, many collaborative and filtering systems (CF) are also used to attain such assumptions for different users. In the next part, sentiment analysis emphasizes the feelings, experiences, and feedback of the users. This works for both online and offline methods. By using these components, the recommendation engine then generates personalized recommendations based on the user's query and browsing history.

On the following strands, an experiment has been conducted on a data-set of e-commerce products and user ratings and reviews to evaluate the performance of their proposed system. The results of that experiment assured that the user reviews were far more effective than user ratings. Keyword filtering proved to be critically useful while ratings appeared as volatile for measuring the accuracy of recommendations [6].

Making website recommendations using natural language processing (NLP) using user ratings and reviews improves the user experience and helps to increase sales. Although, this research deduces that additional implementation [6] and groundwork are necessary to evaluate its performance for larger data-sets and real-world applications.

2.2 Sentiment Analysis in Product Reviews

In the field of e-commerce, online platforms provide an avenue for customers to share reviews and evaluations of various products (Mukherjee and Liu, 2012)., which are essential to maintain the reputation and credibility of an e-commerce store. While ratings can provide an overview of the product, reading reviews is the best way to get a complete understanding. However, with the vast number of reviews available, it is not practical to read them all, and there is a need to mine valuable information from reviews to understand the customer's preferences accurately and make informed decisions.

To address this issue, the paper proposes a sentiment-based rating prediction method using sentiment analysis, opinion mining, stemming, VC dimension, and TF-IDF. Sentiment analysis is used to determine the emotional tone behind a series of words, and emotional tones towards the product can be extracted, allowing customers to make informed purchasing decisions. The paper presents a sentiment-based rating prediction and recommendation model that aims to predict the rating of products based on user reviews. The approach uses sentiment similarity, interpersonal sentiment influence, and product reputation similarity in a unified matrix factorization framework to achieve the rating prediction task (Li et al., 2019).

Additionally, the product feature extraction module is a key component of the proposed system, which extracts product features from unstructured reviews (Zhu et al., 2020). The algorithm used for feature extraction separates product features using the combinations of dependencies. The Stanford dependency parser is used to identify conditions in a sentence, and the Stanford deep analyzer is used to determine the sentiment of review sentences. The paper suggests that future research should investigate more complex strategies for opinion and product feature extraction, as well as new classification models that can address the property of arranged names in rating prediction. Additionally, the sentiment lexicons could be enhanced to apply fine-grained sentiment analysis (Chen et al., 2019).

2.3 Improving Relevance Quality in Product Search using Semantic Similarity

Given an arbitrary query q Q and product a A, a paper created a measure of semantic similarity denoted by g: (Q, A) R, assuming the cardinality of Q and A to be infinite [3]. It chose g because it prioritized a model that optimizes predictive performance constrained by offline resources above one that meets stricter online inference demands. It adopted BERT and posed the problem as a two-sentence classification (Devlin et al., 2019) in order to accommodate textual inputs for the query and product. For two-sentence classification utilizing BERT as a cross-encoder, the input sequence is generated by prefixing a 'CLS' token followed by the textual representation of the query, a special separator token ('SEP'), and then the textual representation of the product via the product title. This input sequence is segmented into sub-words using the WordPiece technique (Wu et al., 2016) and given to a pre-trained BERT-based model (12 transformer blocks, 768 hidden units, and 12 self-attention heads). It additionally pre-trained the BERT-base model using Masked LM and Next Sentence Prediction tasks, as stated in (Devlin et al., 2019), on product metadata comprising title and description [3].

The output embedding for the 'CLS' token is fed to a final linear classification layer when generating a classification model. Using binary cross entropy loss, all model weights, including transformer block layers, are learned concurrently. The labeled dataset consisting of judgements tuples (query, product, label) is derived from previous query-product data with relevance judgements (relevant vs. irrelevant) and is divided into train, validation, and test datasets. For measuring relevance quality, the NDCG (Normalized Discount Cumulative Gain) metric was utilized, which obtains a maximum value of 1 when the rank order adheres to the ideal relevance label ordering, and is then averaged across all queries [3]. The GBDT baseline model typically performs effectively for portions of frequent traffic where prior consumer behavior data are available, but will be limited or noisy otherwise. In the end, a GBDT-based predictor is constructed similarly to the baseline, with the addition of the BERT model score as an input feature utilized for feature selection. The final model displays the added value above the existing collection of online-efficient characteristics, where gains may be observed. Utilizing this high-precision predictor, it investigated a number of search ranking applications that used this model with offline-generated scores for their final objective [3].

2.4 Personalized Recommendations of Products to Users

A personalized movie recommendation system is proposed here using both content-based (CBF) and collaborative filtering (CF) approaches [5]. CBF uses TF-IDF and cosine similarity to suggest movies based on user preferences and the similarity between movies. CF utilizes Deep-Learning with memory-based and model-based filtering techniques to suggest movies based on user analysis of previous products.

The user and movie IDs are embedded into continuous vectors and passed through the neural network NN) with dropout layers and ReLu activation features to predict scores [5]. The Root Mean Squared Error (RMSE) is used to evaluate the model's performance, while precision and recall are used to compare between the two approaches. CF was found to be superior with a precision of 57% and recall of 68% compared to 5.6% and 4.3% for CBF [5]. The study incorporates embedding-based deep learning techniques in the collaborative filtering approach to provide better results for personalized movie recommendations. The training and validation loss over a set number of epochs can also be used to further analyze the performance of the model.

Using CF integrated with Neural Networks outperforms CBF, and provides a more thorough representation of why the collaborative approach is better compared to content-based filtering. However, this research concludes by proposing the hybridization of both filtering methods [5] to overcome drawbacks from the techniques, such as a lack of huge data-sets and inefficient data processing.

3 Proposed System

Our model will work directly on the combination of collaborative and content-based filtering into a single system, which will demonstrate significant improvements in a generalized product recommendation prototype. In this approach, we will integrate the semantic contents of user profiles and ratings to calculate the similarity between users.

3.1 Model Overview

1. Extract User information to create their respective profiles.

2. Use Clustering Algorithm to group the user profiles, then use the result to form Group Rating Matrix.

3. Calculate User-User Similarity by using a modified Cosine Similarity Algorithm, User Rating Matrix, Group Rating Matrix and Pearson Correlation-Based Algorithm.

4. Make Predictions for an item by performing a weighted average of deviations from the adjacent item's mean.

Example Procedure: If there are 6 users, we group them into two clusters based on the semantic contents extracted from the user profiles. Then we model the clusters as items to form a new User-Item Matrix for the Collaborative Filter. Finally, the recommendations are made by applying CF Algorithm.

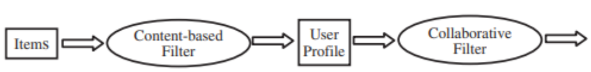

Figure 1: Sequential Combination Model [7]

3.2 Experiment

Here, we will elaborately explain the proposed model and the methods used to optimize an already predefined recommendation system.

3.2.1 User Profile

The User Profile specifies the information needs or item preferences of the user. It contains several profile vectors and each of them represents an attribute of the preferences. For example, in a tech-based eCommerce system, a laptop item contains attributes such as resolution, screen-size, response time, battery life etc. These aspects (attributes) are modelled in the form of profile vectors that contain attribute-value pairs, e.g response time: 1 ms. *Two methods* are used to create the User Profile.

1. ***Manual Weighted Method***: The user expresses preferred aspects of a certain item, but also expresses the degree to which they like these items [7]. An aspect equation can be assembled which contains variables denoting item attribute and weight of the item attribute.

$$\{Aspect_1 : (a_{1,1}(w_{1,1}), ...a_{1,m}(w_{1,m})), ...Aspect_i : (a_{i,1}(w_{i,1}), ...a_{i,m}(w_{i,m}))\}$$

Figure 2: Manual Weighted Method

2. ***Auto Weighted Method***: As user ratings increase, the weight of item attributes in a certain aspect changes, which in turn alters the user profile. This updated user profile can be automatically constructed by following an equation in order to reduce the burden of users [7]. The equation contains number of items, item attribute, threshold and item aspect.

$$W_{n,m} = \frac{Num_{\text{item} \subseteq \text{attribute}_m \text{ of aspect}_n \mid \text{item} > \text{threshold}}}{Num_{item > threshold}}$$

Figure 3: Auto Weighted Method

After a synopsis aspect is formed using these methods, the properties appear to be different from other aspects of the item. Thus, a weighted keyword vector needs to be prepared for each user. Keywords are extracted from the synopsis of e.g the laptop description data-set, for which the user expresses their preference in terms of user rating and threshold [7]. Then, the weight of each keyword is computed using the familiar TF-IDF formula from the Content-Based Filtering Approach.

Finally, we are left with appropriate representations of the user profiles, which will be passed on through to the clustering procedure to group users and provide semantic content information.

3.2.2 Group Rating

The ***K-Means Algorithm*** is a computationally efficient clustering method and provides an output relatively fast. Thus, we use this method along with a couple of adjustments, in order to group the users.

In our case, we apply a fuzzy set algorithm to represent the correlation between an object and a cluster. First, the user profiles are grouped into a given number of clusters. The choosing of the initial cluster proves to be a significant issue, thus the refinement algorithm is used to pick the starting point for the application of the algorithm. Afterward, the probability of an object belonging to a cluster is calculated based on the Euclidean Distance and the Maximum Counter-Similarity between the object and cluster (refer to Figure 4) [7]. However, the fuzzy membership in a cluster is assigned at the last step, thus the ***fuzzy K-Means Algorithm*** is also applied to group the items, where each object is assigned with a fuzzy membership for each iteration in the process.

$$Pro(j,k) = 1 - \frac{CS(j,k)}{MaxCS(i,k)}$$

Figure 4: Probability of 1 Object in a Cluster

After the grouping is completed, we obtain a new Group Rating Matrix which will be sent to the next layer for similarity calculations.

3.2.3 Similarity Computation

Collaborative Filtering (CF) Algorithms are then used to calculate the similarities between users and come up with precise predictions. There are many ways to perform the calculations such as: the Pearson Correlation Algorithm which is most commonly used and the Cosine Correlation Algorithm. Users with similar interests might exhibit different rating patterns and this tends to be a drawback. Thus, a modified version of the Cosine Correlation Algorithm is used to offset this issue.

Since there are differences between the value scale of User Rating and Group Rating Matrix, normalization of values can be used by enlargement of continuous values or decreasing discrete values to calculate the similarity. However, the most efficient way to compute similarities is to use both aforementioned methods.

First, the ***Pearson Correlation-Based Algorithm*** is used to calculate the user-user similarity from the User Rating Matrix, and then the adjusted ***Cosine Correlation Algorithm*** is used to calculate the user-user similarity from the Group Rating Matrix. Finally, the total user similarity is demonstrated as the linear combination of the above two [7] in the form of an equation containing similarity of users using User Rating Matrix and Group Rating Matrix (refer to Figure 5). In addition to that, a combination coefficient (c) is used to compare and contrast between the weights of the respective matrices.

$$sim(k,l) = sim(k,l)_{user} \times (1-c) + sim(k,l)_{group} \times c$$

Figure 5: Total User-User Similarity Calculation

However, when user ratings change, the weights of the matrices will also fluctuate, which makes it difficult to find an optimal and deterministic value for c. For this reason, an adaptive solution has to be implemented to find the value of c.

The solution goes as follows: set the initial value to 0.5, decrease the initial value by a constant value of 0.1; if performance decreases, increase the value by a constant value of 0.1 until performance decreases, else, continue to decrease the the value by a constant value of 0.1 until performance increases. [7]

3.2.4 Collaborative Prediction

Prediction for an item is calculated by performing a weighted average of deviations of the adjacent item's mean. The top N rule is used to choose the nearest N neighbors based on user similarity [7]. A general formula by Resnick is used to devise a prediction on an item by the user, with the use of average ratings (refer to Figure 5). This method can be categorized as a ***Collaborative Prediction*** method because it predicts an item based on a user with similar interests.

$$P_{k,i} = \overline{R}_k + \frac{\sum_{u=1}^{n}(R_{u,i} - \overline{R}_u) \times sim(k,u)}{\sum_{u=1}^{n}|sim(k,u)|}$$

Figure 6: Resnick's General Formula

3.2.5 New User Predictions

In our proposed hybrid approach, we can make predictions for a new user with the manual user profile based on the grouping information. Whereas, in the conventional CF approach, it is difficult to make predictions as the new user has not placed ratings on any item [7].

New users in our system should have their own manual profiles before entering the system, but these user profiles can still be easily constructed by an abstract specification depending on a statement by the user [7]. For new users, a modified version of Resnick's equation (mentioned in 3.2.4) can be used to make predictions. This is done by replacing the standard baseline variable of user rating to an average rating variable of all ratings on the new user's nearest neighbors [7]. The variable can be inferred from the Group Rating Matrix.

Thus, in our system, there is a way to integrate predictions for new users and of course, existing users using the original Resnick equation. Both of the interpretations of this formula use the Collaborative Prediction Method.

4 Result Analysis

4.1 Data-Set & User Profile

We consider a movie data-set to implement our model. It includes 61,265 users, 1,623 movies, and over 2.8 million ratings from users and experiments were run on this data. The discrete rating scale accepts values between one and five. The original ratings, which range from 0 to 1, have been mapped. One nonzero entry out of the total entries is considered a sparse entry in this data set. As a result, each movie data set has a sparsity of 0.9717. The data set is split into a training set and a test set. Only 20% of users are chosen at random to participate in the testing, while the remaining users make up the training set. There are no user preferences included in the data collection for each movie. As a result, we must create users using the auto-weighted approach from the training data set. Each movie's data only has information on the genre, therefore we gather information on the actors, actresses, and directors from the Internet Movie Database (http://www.imdb.com) to add more details to the user profiles.

4.2 Features of our System

4.2.1 Combination Coefficient & Number of Clusters

Depending on the various cluster strengths, the combination coefficient's ideal value varies. For our test data, we employ an adaptive method to determine the best value. For the test data we used, 0.3 is the ideal value. We use the group-rating techniques that were previously developed and run our trials with various cluster densities. The test results are depicted in the figure below. It is evident that the quantity of clusters influences the accuracy of the forecast. As was previously mentioned, it seems that the fuzzy k-means algorithm represents the fuzzy membership more accurately than the adjusted k-means method. Our trials, however, could not clearly demonstrate any benefits. We utilize the modified k-means method instead of the fuzzy k-means algorithm in the next parts because it has a lower computational complexity. Mean Absolute Error is denoted as MAE in the figures below.

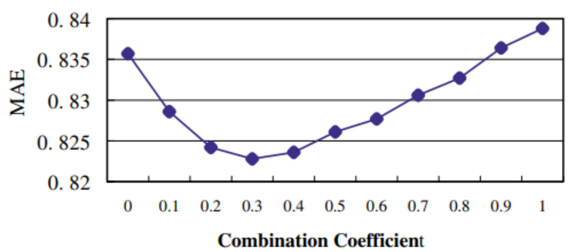

Figure 7: Relationship of MAE vs Combination Coefficient [7]

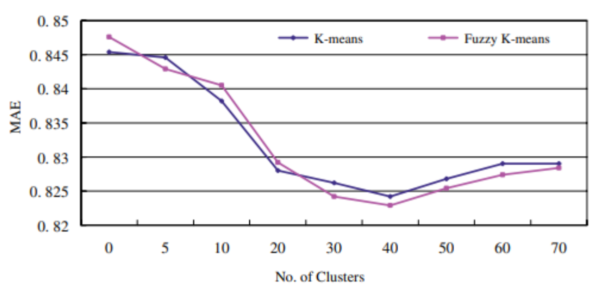

Figure 8: Relationship of MAE vs Number of Clusters; K-Means & Fuzzy K-Means [7]

4.2.2 Methods for Computing User-User Similarity

We have noted that the group-rating matrix and user-rating matrix have different value scales. The value should be changed to the same scale as a result, or the user-user similarity should be computed independently. In our tests, we increase the group-rating matrix's value scale from [0 1] to [0 5], and we use the Pearson correlation-based algorithm to determine how similar the new rating matrix is to the original. This technique is referred to as the expanded Pearson correlation-based approach. In addition, we compute the similarity based on the rating matrix, which is made up of the group-rating matrix and the user rating matrix, using the Pearson correlation-based algorithm. The non-enlarged Pearson correlation-based approach is the name we give to this technique. Finally, the user-user similarity from the user-rating matrix is calculated using the Pearson correlation-based algorithm, and the user-user similarity from the group-rating matrix is calculated using the modified cosine algorithm. The sum of the two previous calculations is then used to get the overall user-user similarity. This method is known as the linear combination strategy. The enlarged Pearson technique, the classic Pearson method, and the linear combination approach are shown to perform best in the figure below. The approach using non-enlarge Pearson correlations performs the worst.

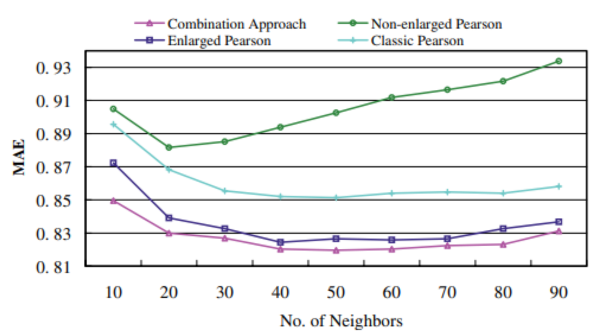

Figure 9: Relationship of MAE vs Number of Neighbors; All Approaches [7]

4.2.3 Neighborhood Size & Construction of User Profiles

The quality of the prediction is significantly impacted by the neighborhood's size. We alter the number of neighbors in our trials and calculate the Mean Absolute Error (Herlocker, Konstan, Borchers, & Riedl, 1999). The ideal Mean Absolute Error (MAE) value is attained in our method when the number of neighbors is increased from 30 to 50. Movie genre, actor, actress, director, and summary are used to generate user profiles. Proper item attributes can help create user profiles that are more accurate, which will enhance the effectiveness of recommendations. In this investigation, we discover that the movie synopsis is the most useful characteristic for effectively identifying consumer desire (refer to Figure 10).

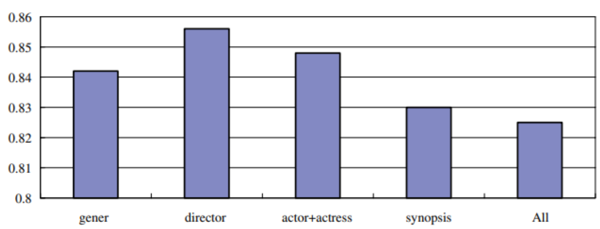

Figure 10: Construction of User Profiles Based on Movie Attributes [7]

4.3 New User Problem

The proposed strategy in this model outperforms previous recommender systems and offers a potential solution to the issue facing new users. Yet, objective evaluations are difficult due to these systems' dynamic nature and variations in decision-making processes.

In an experiment, one user's ratings were removed from the training set, and user profiles were built only using genre information. The average approach was then used to make predictions for this user, which were then compared to the user's actual ratings for 21 items in the test data. The fact that the forecasts only partially matched the user's preferences showed that the method was only marginally successful in resolving the new user issue (refer to Figure 11).

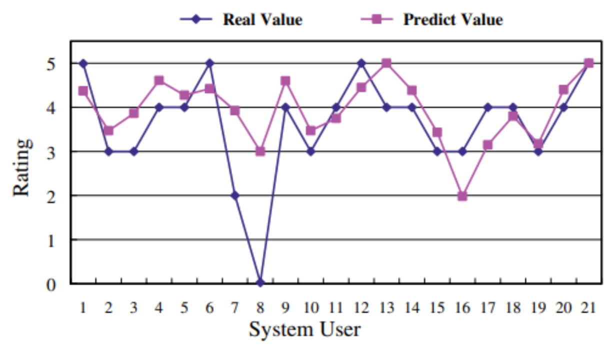

Figure 11: Representation of Predictions for New Users [7]

4.4 Comparison with Other Works

Due to the variations in their methodologies and the dynamic character of these systems, the paper highlights the difficulties in assessing hybrid recommender systems. While some systems employ numeric ratings, others use Boolean values to reflect user preferences. Likewise, while some systems just utilize user profiles, others use item contents in their suggestions. The proposed five-level rating system is thought to reflect the complexity of human preferences more accurately.

The recommended strategy is then contrasted with that of other recommender systems. Due to discrepancies in their decision-making processes or variances in the weights of their content-based filters and collaborative filters, several of these systems are challenging to compare using the recommended technique. A comparison with the Fab system, however, reveals that the recommended strategy outperforms it. The Fab system, a specific version of the recommended technique where the combination coefficient is 1, solely utilizes user profiles to produce suggestions.

According to the article, certain filter-bot concepts that involve rating robots participating as members of a collaborative filtering system enhance performance but do not address the issue with new users. By considering user clusters as items, the recommended method successfully resolves this issue.

5 Limitations

The individual models CBF and CF have some drawbacks which are mentioned below.

1. The biggest limitation of the Content-Based Filtering (CBF) system is that it cannot provide fortuitous results as output [7]. Since every bit of information is carefully selected and categorized, the recommendations are solely based on the content.
2. It takes a significant amount of experience to efficiently utilize Content-Based Filtering systems. As a result, newcomers and beginners might face a lot of difficulties operating these systems [7].
3. Cold start problems and data disparity can be huge problem when using Collaborative Filtering (CF). It can minimize the accuracy of different filtering algorithms as there is much restraint and less data for new users.

Fortunately, this model has been devised in such a way that it prevents encountering the restrictions of their individual models, hence making this combined model optimal and efficient.

6 Conclusion

While Collaborative Filtering (CF) can suggest information based on other users' opinions, Content-Based Filtering (CBF) can choose information based solely on a user's own profile contents. However, a more promising way for information filtering is a hybrid filtering method that combines these two strategies. First, we have implemented a model that groups the users into clusters using a group rating matrix and clustering algorithms. Afterward, by using both CF and CBF, the product predictions for new/current users are created and integrated.

In terms of future research, advanced clustering methods can be developed to group item contents rather than user profiles to get better performance since item-based CF recommendation algorithms can further improve the performance of the suggestions as it outweighs user-based CF algorithms.

Overall, with most recommendation technologies using only one of the models, this combination model would be a significant upgrade in the way a product recommendation system works. It accounts for a larger array of users including new users, is computationally efficient, intelligently, accommodates for user preferences and provides the maximum utility for users browsing marketplaces on the internet.

7 References

[1] Balster, A., Hansen, O., Friedrich, H., & Ludwig, A. (2020, May 11). An ETA Prediction Model for Intermodal Transport Networks Based on Machine Learning. Business & Information Systems Engineering, 62(5), 403–416. https://doi.org/10.1007/s12599-020-00653-0

[2] Rengaraj, N., Mohan, K. C., & N, M. (2019). Movie Recommendation System using Term Frequency-Inverse Document Frequency and Cosine Similarity Method. International Journal of Recent Technology and Engineering, 7(6S3). https://www.ijrte.org/wp-content/uploads/papers/v7i6s3/F1018376S19.pdf

[3] Garakani, A. B., Yang, F., Hua, W., Chen, Y., Momma, M., Deng, J., Gao, Y., & Sun, Y. (2022b, January 1). Improving Relevance Quality in Product Search using High-Precision Query-Product Semantic Similarity. Proceedings of the Fifth Workshop on e-Commerce and NLP (ECNLP 5). https://doi.org/10.18653/v1/2022.ecnlp-1.6

[4] Thomas, K. K. (2019). Sentiment analysis in product reviews using Natural Language Processing and machine learning. International Journal of Information Systems and Computer Sciences, 8(2), 147–151. https://doi.org/10.30534/ijiscs/2019/35822019

[5] Chinivar, S. N. (2022, September 30). Personalized Recommendations of Products to Users. International Journal of Recent Technology and Engineering; Blue Eyes Intelligence Engineering and Sciences Engineering and Sciences Publication - BEIESP. https://doi.org/10.35940/ijrte.c7274.0911322

[6] Bose, S., Choudhary, C., Behra, A., & Kumar Sar, S. (2020). An Innovative Recommender System for E-Commerce Websites using Natural Language Processing. International Journal of Recent Technology and Engineering, 8(6), 4085–4089. https://doi.org/10.35940/ijrte.f7168.038620

[7] Kim, B. C., Li, Q. X., Park, C., Kim, S., & Kim, J. (2006). A new approach for combining content-based and collaborative filters. Journal of Intelligent Information Systems, 27(1), 79–91. https://doi.org/10.1007/s10844-006-8771-2