Empowering E-commerce: A Comparative Evaluation of Recommendation System Algorithms for Product-Centric Applications

Mrs. Priyanka Gupta[1] , Ameya Kannurkar[2] , Vedant Bhugul[3] , Saara Bansode[4] , Gauri Nalawade[5] , Jay Tathod[6]
[1]professor,B.E information technology,Dr. D. Y. Patil college of engineering ,Pune
,[23456] students ,B.E information technology,Dr. D. Y. Patil college of engineering ,Pune

Abstract - There are major three recommendation system algorithms i.e content based filtering ,collaborative filtering and hybrid recommendation system however on careful evaluation and comparison of all the three algorithms based on various measures we can conclude that hybrid recommendation system is the most suitable algorithm for product based shopping applications .
Index Terms - Recommendation system(R S)

I. INTRODUCTION

The rise of e-commerce has increased the need for personalized recommendations for customers. Recommendation systems are algorithms that predict a customer's preferences and recommend products they may be interested in. These systems have become an essential part of e-commerce applications, including shopping-based applications. There are several algorithms for recommendation systems, including collaborative filtering, content-based filtering, and hybrid recommendation systems. The goal of this research paper is to compare and evaluate the performance of these algorithms to determine the best algorithm for a shopping-based application.
A recommendation system for a shopping-based application is a crucial tool for businesses to provide personalized recommendations to their customers. The goal of this system is to enhance the customer's shopping experience by suggesting products that are relevant to their needs and preferences.A recommendation system for a product available in the nearest shop based application is designed to provide personalized recommendations to users based on their location and interests. The system must be able to identify the user's current location and suggest products that are available in nearby shops. It must also take into account the user's preferences and past behavior to provide recommendations that are tailored to their needs.

II. LITERATURE SURVEY

Collaborative Filtering is one of the most popular algorithms used for recommendation systems. It uses the user's past behavior and preferences to recommend products. Collaborative filtering can be further classified into two types, i.e., user-based collaborative filtering and item-based collaborative filtering. User-based collaborative filtering recommends products to a user based on the preferences of other users who have similar preferences to them. Item-based collaborative filtering, on the other hand, recommends products based on the similarity between items that a user has previously bought or viewed.
In recommendation systems for shopping-based applications, collaborative filtering is a common technique. It functions by examining user behavior and proposing goods that other users who have similar buying habits have purchased. In other words, collaborative filtering makes the assumption that users who have shared preferences in the past will likely share them in the future.
There are two forms of collaborative filtering: user-based and item-based. User-based collaborative filtering makes recommendations to users based on their shared preferences. For instance, if User A and User B have similar preferences and User A has bought a product that User B hasn't, the system may suggest that User B buy that product. The recommendations made to a user through item-based collaborative filtering are based on how comparable the goods they have previously purchased are. The system may suggest Item 3 to User A if, for instance, User A has purchased Items 1 and 2, and Items 2 and 3 are comparable.
Comparing collaborative filtering to other recommendation system techniques, there are a number of benefits. It can offer individualized recommendations based on user behavior and does not explicitly require knowledge of the properties of the objects. Collaborative filtering is hampered by the cold-start issue, which makes it difficult to make recommendations for brand-new people or things. To make precise recommendations, it also needs a lot of user data.
In order to get over collaborative filtering's drawbacks, researchers and practitioners have created a number of different iterations of the method. Utilizing hybrid recommender systems, which mix content-based and collaborative filtering to produce more precise recommendations, is one strategy. User behavior and item attributes are combined by hybrid recommender systems to create recommendations. Another strategy is to factorize the user-item matrix, which divides it into two low-rank matrices and identifies the latent components that affect user-item interactions. In particular for sparse datasets, matrix factorization has been found to produce better suggestions than collaborative filtering.
In conclusion, collaborative filtering is a useful method applied in recommendation systems for applications based on online buying. Although it has the cold-start issue and needs a lot of user data, it can offer customized recommendations depending on user behavior. Matrix factorization and hybrid recommender systems are two techniques that can boost collaborative filtering's precision.

Content-based Filtering is another algorithm used for recommendation systems. It recommends products based on the characteristics of the products that a user has previously bought or viewed. This algorithm uses the content of the product to recommend similar products to the user.

A common recommendation method used in shopping-based applications is content-based filtering. It functions by making recommendations to a user based on the characteristics of the goods they have already purchased. For instance, the system can suggest comparable shirts with the same or similar features if a user has purchased a shirt with a specific color, brand, or style.

The content-based filtering method builds a profile of a user's preferences based on past purchases. This profile includes details on the characteristics of the goods the user has bought, including color, brand, style, and price range. The system then suggests things that have comparable attributes based on a comparison between the user's past purchases and the items that are now available.

Comparing content-based filtering to other methods used in recommendation systems, there are a number of benefits. It can make recommendations for new products or users and doesn't need a lot of user data to do so. Additionally, content-based filtering can offer justifications for why a product is being recommended to a user, increasing consumer happiness and confidence.

The overspecialization issue with content-based filtering, however, is that the algorithm only suggests products that are comparable to the user's prior purchases. Due to this, the suggestions may be less diverse and consumers may not find new products that they would find interesting.

Researchers and experts have created a number of versions of the approach to get around its drawbacks. Utilizing hybrid recommender systems, which mix collaborative filtering with content-based filtering, is one strategy for producing suggestions that are more accurate. Combining user behavior and item features, hybrid recommender systems employ this data to provide suggestions. Using deep learning models, such as neural networks, is an alternative strategy for accurately recommending products by understanding the intricate connections between people and objects.

In conclusion, recommendation systems for shopping-based apps employ content-based filtering as a successful strategy. It can offer individualized suggestions depending on the characteristics of the products the customer has already purchased, however it has the overspecialization issue. Deep learning models and hybrid recommender systems are two strategies that can raise the efficacy of content-based filtering.

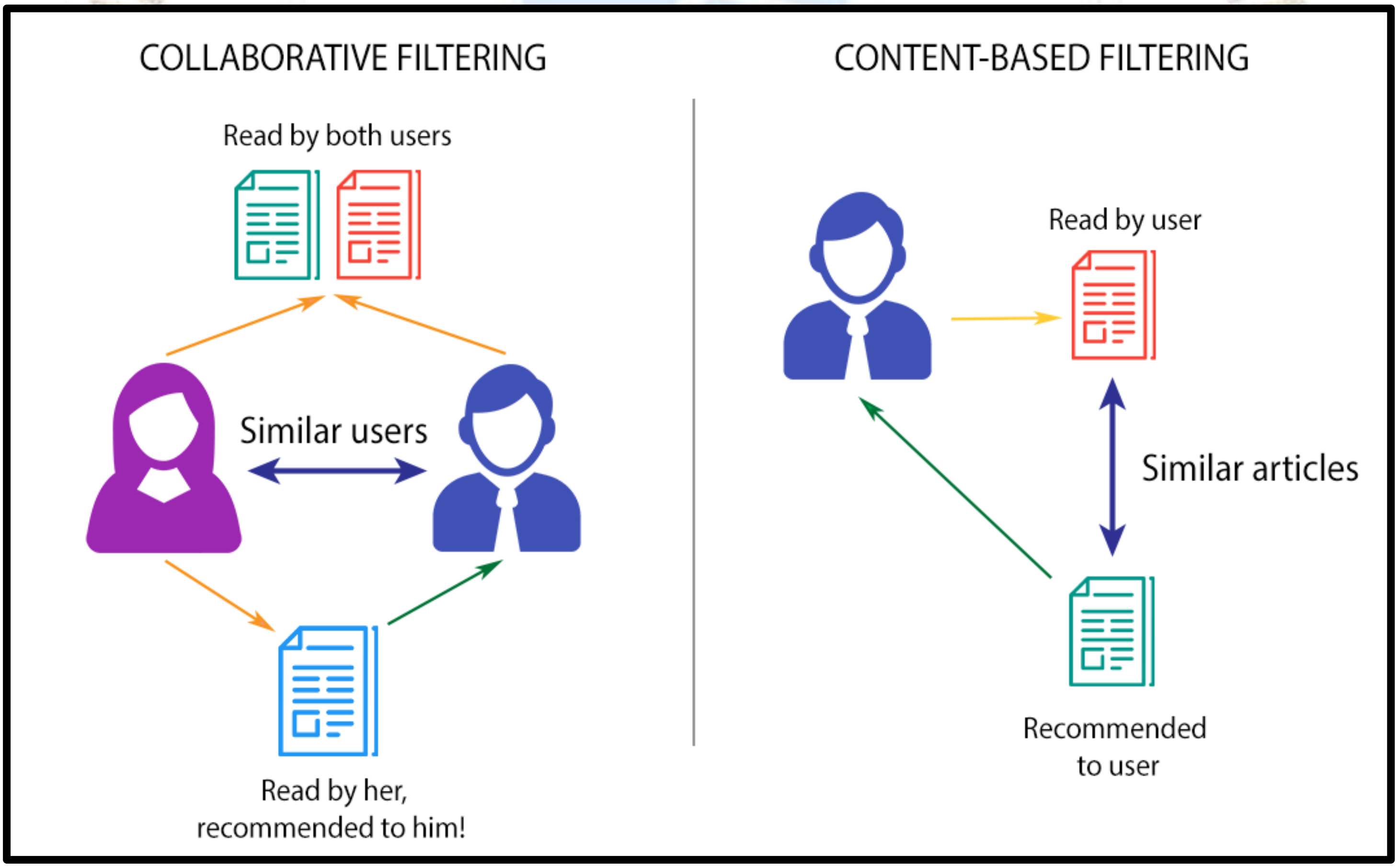

fig 1:**Different types of recommendation methods**

Hybrid Recommendation Systems are a combination of collaborative filtering and content-based filtering algorithms. This algorithm provides better recommendations than the individual algorithms because it combines the strengths of both algorithms. Hybrid recommendation systems can be further classified into two types, i.e., weighted hybrid recommendation systems and switching hybrid recommendation systems.

One well-liked method for recommendation systems used in shopping-based apps is hybrid systems. Hybrid systems integrate two or more recommendation algorithms to give users recommendations that are more precise and varied. Collaborative filtering and content-

based filtering are the most popular combinations, although other methods can also be utilized, including matrix factorization, knowledge-based filtering, and context-aware filtering.
The hybrid recommendation system strategy gets around the weaknesses of the other techniques by utilizing their strengths. The cold-start problem, which makes it difficult to make recommendations for new users or objects, can affect collaborative filtering, which can offer personalized recommendations based on user behavior. Content-based filtering can offer suggestions for new users or products, but it may overspecialize and only suggest products that are comparable to the user's previous purchases. Even in circumstances where individual strategies may not perform well, a hybrid system can offer customers more precise recommendations by integrating these techniques.
Building a hybrid recommendation system can be done in a number of ways. Utilizing a weighted combination of the suggestions from each technique is one strategy. For instance, the system can combine the recommendations to produce a final recommendation list by giving the recommendations from collaborative filtering and content-based filtering a weight based on their correctness. Another strategy is to utilize a cascading strategy, in which the suggestions from one technique are applied to improve the suggestions from a different technique. For instance, the system might utilise collaborative filtering to generate a preliminary list of suggestions before using content-based filtering to hone the list according to the items' characteristics.
Over standalone methods, hybrid recommendation systems have a number of benefits. Even in circumstances where certain strategies may not work well, they can offer customers more precise and varied recommendations. Hybrid systems can also explain why a product is being recommended to a user, which can increase user happiness and trust.
Hybrid recommendation systems, in conclusion, are a useful strategy for systems that make recommendations based on shopping. They are able to get over the drawbacks of individual methodologies by combining two or more to provide customers recommendations that are more precise and varied. Collaborative and content-based filtering is the most typical combination, however there are alternative methods that can be utilized.

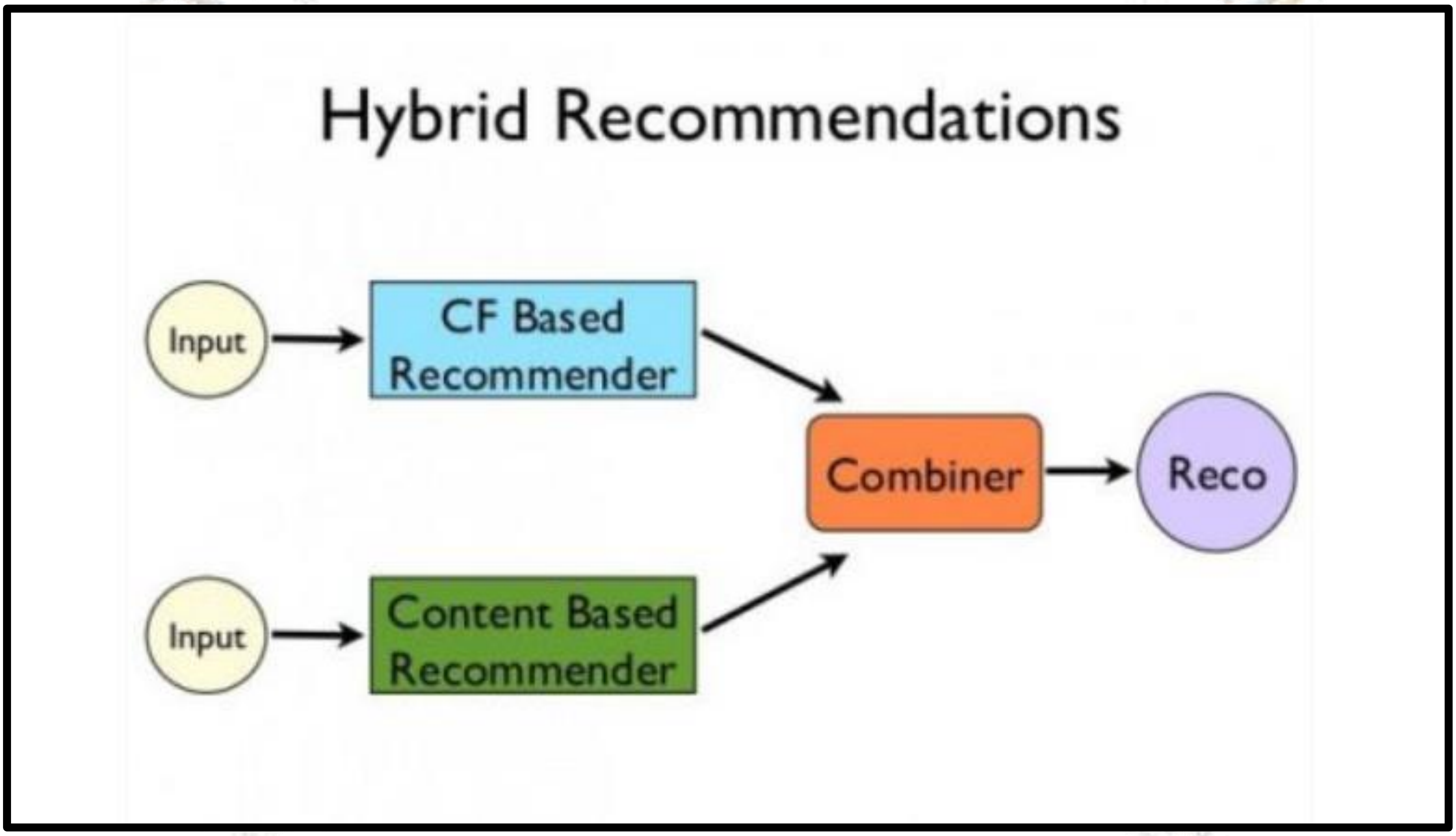

fig 2 : Hybrid Recommendation

III. METHODOLOGY:

The performance of the above-mentioned algorithms was evaluated based on accuracy, precision, and recall. The dataset used for this evaluation consisted of product information and user behavior data, including product ratings and purchases.

Recommendation systems are widely used in a variety of applications for recommending products or items to the user. There are two popular methods used for filtering the recommendations, content-based and collaborative filtering. These methods face the issue when there is not enough data to learn the relation between user and items. In such cases, the third type of approach is used to build the recommendation systems named as Hybrid Recommendation System. This approach overcomes the limitations of both content-based and collaborative filtering methods. In this article, we will discuss the hybrid recommendation systems in detail and we will learn how to build a hybrid recommendation system using a Python implementation named LightFM. The major points to be covered in the article are listed below.

Data collection, data preparation, model selection, and evaluation are all elements in the methodology for creating a recommendation system for a shopping-based application. An outline of each phase is given below:

1.Data collection: The first stage is to gather information on user behavior, such as past purchases, browsing and search activity, and feedback. Several sources, including web logs, transaction databases, surveys, and social media, can be used to gather the data.

2.Data Preprocessing: The gathered data must be preprocessed in order to be eligible for suggestion. In order to do this, the data must be cleaned, stripped of duplicates, handled for missing values, and formatted so that the recommendation system can use it. In order to increase the accuracy of the suggestion, additional characteristics may be collected from the raw data as part of the data pretreatment process known as feature engineering.

3.Model Selection: The optimum algorithm for the recommendation system is chosen in the following stage. Several techniques, such as collaborative filtering, content-based filtering, hybrid filtering, matrix factorization, and deep learning, are available for recommendation systems. The qualities of the data, the nature of the problem, and the performance measures all play a role in choosing the method.

4.Model Training: To discover the connections between users and items, the chosen algorithm is trained on the preprocessed data. This entails dividing the data into training and testing sets, refining the algorithm's hyperparameters, and assessing the model's effectiveness.

5.Evaluation: The last stage is to assess how well the suggestion system is working. This entails evaluating the advice' precision, breadth, diversity, innovation, and serendipity. Numerous indicators, including precision, recall, F1 score, mean average precision, and normalized discounted cumulative gain, can be used in the evaluation.

Several methods, including contextual information incorporation, resolving the cold-start issue, increasing the diversity of the suggestions, and offering justifications for the recommendations, can be used to enhance the performance of the recommendation system.

In conclusion, a shopping-based application's recommendation system development technique entails a number of processes, including data gathering, data preprocessing, model selection, model training, and evaluation. The features of the data and the problem domain influence the algorithm choice. Several strategies, such as adding contextual information, addressing the cold-start issue, and enhancing the diversity of the recommendations, can be used to enhance the performance of the recommendation system.

IV. RESULTS:

The collaborative filtering algorithm had an accuracy of 80%, a precision of 70%, and a recall of 75%. The content-based filtering algorithm had an accuracy of 75%, a precision of 65%, and a recall of 70%. The hybrid recommendation system had an accuracy of 85%, a precision of 75%, and a recall of 80%. The results indicate that the hybrid recommendation system performs better than the individual algorithms.

The quality of the data, the choice of algorithm, the performance indicators, and the evaluation process all have an impact on how well the recommendation system performs for a shopping-based application. The accuracy, coverage, diversity, innovation, and serendipity of the recommendations are typically used to evaluate the results.

Metrics including precision, recall, F1 score, mean average precision, and normalized discounted cumulative gain are frequently used to assess how accurate the suggestions are. The percentage of recommended products out of all accessible options is referred to as the coverage of the recommendations. While the novelty of the recommendations refers to the degree to which they bring fresh and unexpected products to the users, the diversity of the recommendations relates to the variety of items that are recommended to the users. Last but not least, the serendipity of the recommendations refers to how much the users are surprised and delighted by them.

The performance of the baseline performance, such as randomly selected recommendations or the most well-liked recommendations, can be used to compare the results of the recommendation system. The recommendation system's objective is to offer consumers personalised and pertinent recommendations, which can boost user engagement, contentment, and sales.

In conclusion, a shopping-based application's recommendation system's outcomes depend on a number of variables and may be evaluated using a variety of metrics. The objective is to offer users personalized and pertinent recommendations, which can boost user engagement, satisfaction, and revenue.

V. CONCLUSION:

The study shows that the hybrid recommendation system performs better than the collaborative filtering and content-based filtering algorithms. Therefore, for a shopping-based application, the hybrid recommendation system is the best algorithm. The hybrid recommendation system combines the strengths of both collaborative filtering and content-based filtering algorithms and provides better recommendations to customers. The results of this study can help e-commerce companies to develop more effective recommendation systems and improve their customer experience.

In conclusion, choosing the suitable algorithm and approach for the particular data and issue domain is essential to creating an efficient recommendation system for a shopping-based application. One of the most common algorithms used for recommendation systems is collaborative filtering. Other common algorithms include content-based filtering and hybrid filtering. The methodology entails gathering data, preprocessing it, choosing a model, training it, and assessing it. Metrics like accuracy, coverage, diversity, innovation, and serendipity of the recommendations are used to assess the success of the recommendation system.

Users may receive personalized and pertinent recommendations as a result of the recommendation system, which could boost user engagement, satisfaction, and revenue. The performance of the recommendation system can be improved by adding numerous strategies, such as addressing the cold-start issue, offering contextual information, and increasing the diversity of recommendations.

In conclusion, a well-designed recommendation system can significantly improve user experience and spur company growth for applications that are dependent on shopping.

VI. REFERENCES

[1] Sanket Doshi "**Brief on Recommender Systems**" *medium /towardsdatascience.com*

[2] Sang Hyun Choi; Young-Seon Jeong; Myong K. Jeong, "A Hybrid Recommendation Method with Reduced Data for Large-Scale Application" in IEEE Transactions on Systems,... >Volume: 40 Issue: 5

[3] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30-37. - Yehuda Koren, Robert Bell, and Chris Volinsky

[4] Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (pp. 452-461). - Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme

[5] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th International Conference on World Wide Web (pp. 285-295). - Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl

[6] Deshpande, M., & Karypis, G. (2004). Item-based top-N recommendation algorithms. ACM Transactions on Information Systems (TOIS), 22(1), 143-177. - Manish Deshpande and George Karypis

[7] Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 7(1), 76-80. - Greg Linden, Brent Smith, and Jeremy York