

Article

Natural Language Processing and Machine Learning-Based Solution of Cold Start Problem Using Collaborative Filtering Approach

Kamta Nath Mishra ¹, Alok Mishra ^{2,*} , Paras Nath Barwal ³ and Rajesh Kumar Lal ⁴

¹ Department of Computer Science & Engineering, Birla Institute of Technology, Ranchi 835215, India; knmishra@bitmesra.ac.in

² Faculty of Engineering, Norwegian University of Science & Technology, 7034 Trondheim, Norway

³ Centre for Development of Advanced Computing, Noida 201307, India; pnbarwal@cdac.in

⁴ Department of Electronics & Communication Engineering, Birla Institute of Technology, Ranchi 835215, India; rklal@bitmesra.ac.in

* Correspondence: alok.mishra@ntnu.no

Abstract: In today's digital era, the abundance of online services presents users with a daunting array of choices, spanning from streaming platforms to e-commerce websites, leading to decision fatigue. Recommendation algorithms play a pivotal role in aiding users in navigating this plethora of options, among which collaborative filtering (CF) stands out as a prevalent technique. However, CF encounters several challenges, including scalability issues, privacy implications, and the well-known cold start problem. This study endeavors to mitigate the cold start problem by harnessing the capabilities of natural language processing (NLP) applied to user-generated reviews. A unique methodology is introduced, integrating both supervised and unsupervised NLP approaches facilitated by sci-kit learn, utilizing benchmark datasets across diverse domains. This study offers scientific contributions through its *novel* approach, ensuring rigor, precision, scalability, and real-world relevance. It tackles the cold start problem in recommendation systems by combining natural language processing (NLP) with machine learning and collaborative filtering techniques, addressing data sparsity effectively. This study emphasizes reproducibility and accuracy while proposing an advanced solution that improves personalization in recommendation models. The proposed NLP-based strategy enhances the quality of user-generated content, consequently refining the accuracy of Collaborative Filtering-Based Recommender Systems (CFBRSs). The authors conducted experiments to test the performance of the proposed approach on benchmark datasets like *MovieLens*, *Jester*, *Book-Crossing*, *Last.fm*, *Amazon Product Reviews*, *Yelp*, *Netflix Prize*, *Goodreads*, *IMDb (Internet movie Database) Data*, *CiteULike*, *Epinions*, and *Etsy* to measure global accuracy, global loss, F-1 Score, and AUC (area under curve) values. Assessment through various techniques such as random forest, Naïve Bayes, and Logistic Regression on heterogeneous benchmark datasets indicates that random forest is the most effective method, achieving an accuracy rate exceeding 90%. Further, the proposed approach received a global accuracy above 95%, a global loss of 1.50%, an F-1 Score of 0.78, and an AUC value of 92%. Furthermore, the experiments conducted on distributed and global differential privacy (GDP) further optimize the system's efficacy.

Keywords: e-commerce; recommender system; collaborative filtering; machine learning techniques



Citation: Mishra, K.N.; Mishra, A.; Barwal, P.N.; Lal, R.K. Natural Language Processing and Machine Learning-Based Solution of Cold Start Problem Using Collaborative Filtering Approach. *Electronics* **2024**, *13*, 4331. <https://doi.org/10.3390/electronics13214331>

Academic Editors: Agnieszka Nowak-Brzezińska and Agnieszka Konys

Received: 5 September 2024

Revised: 24 October 2024

Accepted: 28 October 2024

Published: 4 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The challenge of cold starts in recommender systems presents a significant hurdle, which is particularly noticeable when these systems encounter new users or recently introduced items with limited historical data. Traditional recommendation algorithms heavily depend on past user interactions and item preferences to tailor suggestions. However, when faced with users who have recently joined the platform or entirely new items, the absence

of such data poses a formidable challenge. This data gap complicates understanding user preferences and the ability to provide relevant recommendations, potentially resulting in a suboptimal user experience. Effectively addressing the cold start issue is crucial to meet the demands of both new users and emerging items in recommendation systems.

The influx of new internet users has brought about an unprecedented amount of data. Consequently, users are inundated with a staggering number of choices across various platforms such as Netflix, Spotify, and Amazon. This abundance of options leads to information overload, highlighting the importance of ensuring that users receive tailored recommendations to enhance their experience. Ensuring that users get precisely what they need is critical for improving the user experience of any product. User reviews play a crucial role in addressing the cold start problem within recommender systems for several compelling reasons.

User reviews provide a valuable source of data that empowers recommender systems to offer informed recommendations, especially when dealing with the cold start problem. By leveraging the capabilities of NLP (natural language processing) and meticulously analyzing user sentiments and feedback, these systems can effectively bridge the cold start gap, delivering accurate and highly personalized recommendations from the outset. Hence, recommender systems prove valuable in this scenario by filtering the entire list of items to obtain a smaller subset of items. The classical federated learning architecture can be depicted after implementing a recommender system-based model where an initial list is given as the input and a filtered list is obtained as the output.

A precise mathematical representation of a recommender system entails an algorithm that takes an input set containing “ N ” items and generates an output list comprising “ K ” recommended items. In this context, “ N ” signifies the entire item inventory, and “ K ” represents the specified number of recommendations. To put it another way, the recommender system assists the user in selecting the appropriate product or item, which is crucial for a company. Recommender systems have proven highly beneficial in leading technical organizations worldwide. For example, YouTube reports that 60% of video clicks originate from the home screen, showcasing the impact of their recommendation algorithms [1]. Similarly, Netflix effectively employs recommender systems to suggest shows tailored to users’ preferences [2]. Previous studies offer valuable insights into the varied applications of recommender systems across industries. One notable study by Shuai Zhang et al. provides a thorough analysis of deep learning-based recommender systems, identifying key research trends and proposing strategies for organizing the current literature [1]. Additionally, addressing accuracy improvement remains a central challenge in collaborative filtering recommendation systems, as highlighted in a study by W Lemus Leiva, ML Li, and CY Tsai, which emphasizes the importance of considering rankings’ reliability as input data [2].

In the realm of real-world applications, Kai Wang et al. introduced the RLARS resource, which sets a standard for reinforcement learning-based recommender systems. This resource focuses on creating simulation environments and addressing supposition errors, offline strategy learning, and policy evaluation, essential components in recommendation scenarios [3]. It complements NLP and machine learning methods to tackle the cold start problem by improving collaborative filtering and optimizing recommendations. By learning user preferences over time, reinforcement learning optimizes recommendations in real-world environments, improving accuracy when recommending items to new users or for new products, addressing the cold start challenge more effectively.

Another study by Kumar et al. examines the echo chamber effect in e-commerce platforms, revealing that while tailored e-commerce recommender systems may exhibit echo chamber tendencies in terms of user-click behavior, these tendencies are less pronounced regarding user-buy behavior [4]. The “DNNRec” paper of Kumar et al. [4] proposed a hybrid recommender system using deep learning which complements NLP and machine learning approaches for solving the cold start problem. By combining collaborative filtering with deep neural networks, it improves the understanding of user preferences and

item characteristics, enhancing the accuracy of recommendations for new users or items, thereby addressing cold start challenges effectively. This research underscores the complex dynamics of personalized online retail systems and their impact on user behavior [5].

Building a personalized recommender system involves multiple approaches, broadly categorized into content-based, collaborative, and hybrid filtering techniques. Content-based systems utilize user attributes to match them with relevant products, recommending items with a significant match, typically defined as 80% or higher.

The conceptual construction of this research work focuses on solving the cold start issue in recommendation systems due to sparse data. The NLP-based systems have better performance for language translation that is used for recommendation systems. So, by integrating NLP and machine learning with collaborative filtering, this research aims to enhance prediction accuracy for new users and items, addressing challenges where traditional recommendation techniques fall short due to limited interaction data.

The main research question (RQ) of this paper describes how natural language processing (NLP) techniques, integrating both supervised and unsupervised methods, can improve the effectiveness, scalability, and reliability of collaborative filtering-based recommendation systems while addressing cold start issues, enhancing review quality evaluation, and safeguarding data privacy.

This study makes several key contributions as described below:

- i. Recommender systems have a variety of issues that warrant further study. The cold start problem (CSP), one of the most prevalent issues that collaborative filtering-based recommender systems (CFBRS) encounter is the topic of this research. One of the core contributions of this research work is the further improvement of the effectiveness of problem solutions for the CSP.
- ii. In this study, we have applied an NLP-based approach to improve the quality of user reviews, which will lead to an improvement in the accuracy of recommendations made by the collaborative filtering-based recommender systems.
- iii. Multiple techniques like random forest, Naïve Bayes, Logistic Regressions, etc., are applied to detect the quality of a review submitted by the user.
- iv. Recall, precision, and accuracy scores are computed after using different machine learning methods.
- v. A mix of supervised and unsupervised approaches to NLP are used using the sci-kit to learn the method to further improve the performance of obtained results.
- vi. It is observed that distributed and global differential privacy experiments are possible on both the client and server sides. Further, it has been suggested that modern machine learning models might be compared to dynamic programming dataset-based techniques in further studies.

2. Literature Review

The literature review is divided into four sections, namely, *collaborative filtering*, *content-based filtering*, *deep learning*, and *reinforcement learning-based models*. Further classification is based on specific algorithms employed in the recommender system, with various machine learning and deep learning approaches utilized to enhance recommendation accuracy, as depicted in Figure 1.

2.1. Collaborative Filtering-Based Models

Collaborative filtering neural network topologies are applied by researchers. Three instantiations of the generic framework of NCF, GMF, MLP, and NeuMF, are provided by the researchers and are meant to serve as a roadmap for the development of deep learning recommendation techniques [6–8].

Venil P, G. et al. [9] provide a comprehensive survey of the cold start problem in collaborative filtering systems. The authors explore the various approaches used to address this issue, including content-based filtering, demographic-based filtering, and hybrid approaches. They also discuss the limitations and challenges of these methods and suggest

potential areas for future research. Their paper is an insightful read for anyone interested in collaborative filtering systems and the challenges they face.

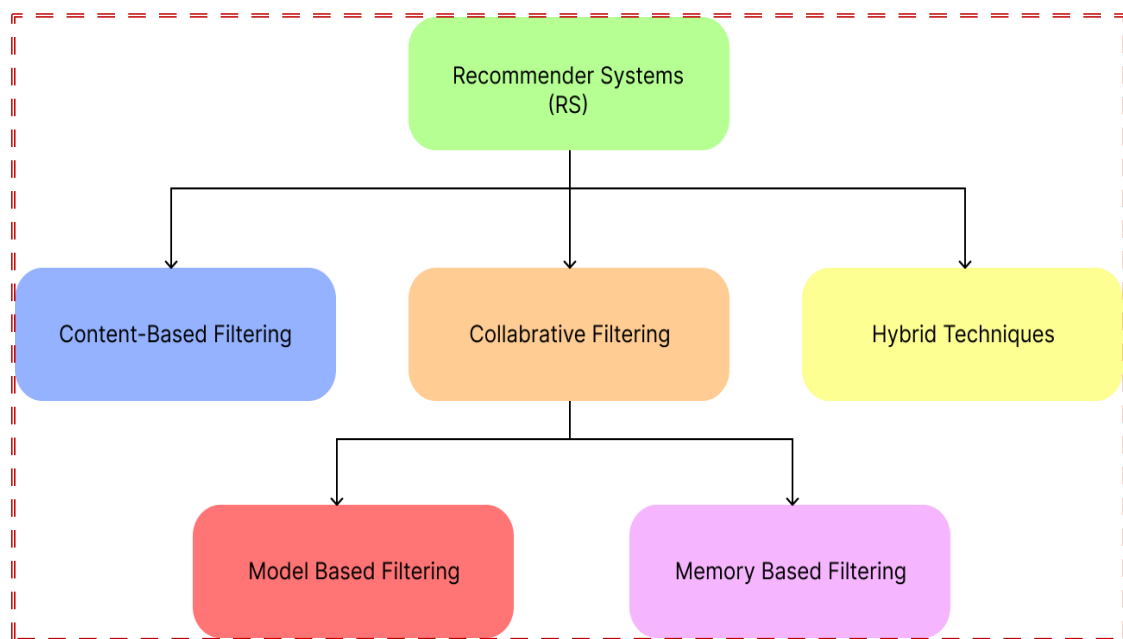


Figure 1. Classification of recommender systems.

In a study carried out by Fu et al. [10], they developed a multi-tasking, multi-policy recommender system. First, the researchers created a task space for each similar user group. BPR was used to pre-train users' embeddings, which were then divided into K clusters using KNN. They assumed that users belonging to the same cluster should be assigned to the same task. To make recommendations for coldstart users, the system employed a universal policy and a task-specific policy. The universal policy was used in the initial recommendation phase, and a discriminator was used to select the most probable task for the user based on the feedback received. Then, the task-specific policy was employed to make the remaining recommendations.

Chen, Li et al. (2015) extensively surveyed the latest advancements in recommender systems using user-generated reviews. Their paper explores sentiment analysis and opinion mining techniques and their integration into recommendation systems, addressing challenges and proposing future research directions. This paper serves as a valuable resource for researchers and practitioners in the field [11]. Wei and Zian [12] present a novel recommendation system. This system effectively addresses the issue of recommending items with minimal user data, often termed "cold start" items. Their approach, which integrates collaborative filtering and deep learning techniques, showcases substantial improvements in recommending these items in real-world scenarios. Further, these methods can potentially be applied in a range of domains, such as e-commerce and online advertising. Furthermore, these approaches shed light on promising ways to improve recommendation systems using deep learning techniques, contributing to the expanding literature on recommender systems [13]. They offer practical guidance for personalized recommendations.

The following are the strengths, weaknesses, and limitations of collaborative filtering:

Strength: Collaborative filtering performs well in designing personalized recommendations by analyzing the interactions between users and items. It enables users to uncover new products based on the preferences of similar individuals.

Weakness: This method may face difficulties with the cold start problem, hinder its ability to suggest items for new users or those with minimum interaction history. It may face scalability issues when dealing with sparse datasets.

Limitation: It primarily concentrates on the behavior of the user, and may overlook the intrinsic characteristics of items.

2.2. Content-Based Filtering Models

By outlining the key characteristics of this type of system, the authors of [14] conducted a field assessment of content-based recommender systems. Their paper examines the key difficulties surrounding the representation of objects, starting with straightforward methods for representing structured data and progressing to more intricate methods, from the field of information retrieval for unstructured data-based items, which makes it unable to directly use all pertinent information.

The researchers of [15] in their study presented a recommender system that uses implicit feedback and is designed to be keyword-aware. The system was implemented on the Hadoop framework and it was tested on a large dataset. The paper described the system's architecture, its algorithm for generating recommendations, and the evaluation results, demonstrating its effectiveness in comparison to other state-of-the-art approaches. The proposed system can be useful for various recommendation applications, especially those that rely on keyword-based searching.

Onestudy [16] address the cold start problem in the insurance domain by proposing a recommendation system that incorporates both user and item information using a heterogeneous information network (HIN). The system is designed to work across multiple domains, including life, health, and automobile insurance, and employs a cross-domain adaptation model, user and item representation learning models, and a recommendation model. The system's performance is evaluated on a large dataset, demonstrating its effectiveness compared to state-of-the-art recommendation methods. The study highlights the potential of HIN-based recommendation systems and provides insights into their effectiveness in addressing cold start problems.

Study [17] focused on the difficulty of offering precise recommendations to new users with minimal or no interaction history in the system. The study compared and reviewed multiple approaches, such as content-based filtering, collaborative filtering, and hybrid techniques, for addressing this challenge. The authors presented an in-depth analysis of the pros and cons of each technique and emphasized the necessity for additional research to generate successful solutions. This analysis can support professionals and researchers in the recommender systems field in determining the optimal approach for managing the new user cold start problem.

In another work by Deepjyoti R. and Mala D. [18], various topics related to recommender systems such as collaborative filtering, content-based filtering, and hybrid approaches were covered. Additionally, they addressed the challenges and limitations of these methods and suggested potential solutions and future directions for research. The paper [19] is useful for researchers and practitioners interested in recommender systems as it provides valuable insights and a comprehensive overview of the field. Overall, it is an essential read for anyone looking to stay up to date with the latest trends and developments in this area.

The following are the strengths, weaknesses, and limitations of content-based filtering approaches:

Strength: Content-based filtering relies on item attributes to recommend similar products, effectively tackling cold start scenarios for new items. This approach operates independently, as it does not depend on data from other users.

Weakness: It may lead to over-specialization, resulting in repetitive suggestions that lack variety.

Limitation: Its performance can be affected when item features are not clearly defined or are insufficient.

2.3. Deep Learning Based Models

This research paper employs a novel approach rooted in NLP to elevate the quality of user reviews, subsequently enhancing the accuracy of recommendations generated by *CFBRS*. The self-generated and standard benchmark datasets utilized in this research encompass diverse user-generated reviews and meticulously pre-processed data from multiple sources. The proposed work carefully applies a blend of supervised and unsupervised NLP approaches through the *sci-kit learn* method. Rigorous experimental analysis is conducted, involving a range of techniques, including random forest, Naïve Bayes, and Logistic Regressions, to assess the quality of user reviews. These techniques yield varying recall, precision, and accuracy scores across different machine learning methods, with the random forest method emerging as the most accurate, achieving remarkable accuracy.

The major objective of this study [20] was to develop a machine learning-based recommender system for Amazon consumers that includes the clothing category. This deep learning algorithm [20] predicts a user's rating on products and recommends the product that is most likely to receive a good rating. The focus of this study [20] is the dilemma of the standard slopes of one algorithm's lower sensitivity and the unreliable rankings in recommender systems. Additionally, the study suggested a gradient methodology based on the combination of reliable data and user-homogenous attributes [20]. The research offers a novel deep-learning structure to improve recommendation system results through collaborative filtering. To retrieve the existing non-linear cognations, researcher presages, reliabilities, and exact suggestions, the authors use the deep learning approach [21]. To provide better and much more accurate homogeneous product recommendations for E-commerce items, the article proposes the *e-SimNet* visual recommendation systems. Such deep learning approaches, as well as the approximation of the most proximate neighbors for retrieving the Top-N suggestions, were used to create their models [22].

The results of the study [22,23] demonstrated that the cross-border e-commerce astute information recommendation system based on deep learning, which is suggested in this study, has a strong selection impact and fits the cross-border e-commerce recommendation requirements.

A deep learning-based hybrid recommender system is suggested in the study [24]. For CF, it makes use of user-item interaction, while CBF makes use of auxiliary data on both users and items. To anticipate the top n-list of things, Deep Hybrid uses MF and DNN connected via a shared common layer. Because there is no explicit feedback, all encounters are considered positive, and examples of negative interaction are picked at random [24].

The researchers suggest using *CF-UIcA*, a neural co-autoregressive model for CF tasks that makes use of structural auto-repressiveness in both the user and item domains. The authors illustrate the efficacy of *CF-UIcA* by evaluating it against two well-known benchmarks, *MovieLens 1M* and *Netflix*, and achieve cutting-edge prediction performance [25]. The study [26] introduced a two-stage CS recommendation engine to address the issues with cold starts. The system is composed of two components, namely the *NNCF* predictor and the *DACR* generator.

The paper [27] proposed a novel model for medical diagnosis that incorporates artificial intelligence and trust mechanisms. They introduced the concept of a *Medical-assisted Diagnosis Model as a Service (MDMaaS)*, which allows healthcare providers to access diagnosis models through a cloud-based platform. The *MDMaaS* employed a multi-stage trust model to ensure the accuracy and reliability of the diagnosis results, as well as a hybrid artificial intelligence model that combines machine learning and rule-based systems. The proposed model was evaluated using a real medical dataset, demonstrating its effectiveness and potential to reduce the workload of medical professionals. The study highlighted the importance of trust mechanisms in medical AI systems and provided valuable insights into their design and implementation.

B. Yi et al. [28] presented an innovative solution to enhance recommendation systems utilizing deep matrix factorization with implicit feedback embedding. By considering

implicit feedback from user–item interactions, the proposed models of [29–31] learn latent representations of users and items via matrix factorization.

The authors evaluated the model on two actual datasets and compared it with various state-of-the-art recommendation algorithms. The experimental outcomes of [32–34] demonstrate that the suggested models outperform existing techniques regarding efficiency and accuracy.

The following are the strengths, weaknesses, and limitations of Deep Learning based approaches:

Strength: Deep learning techniques can identify complicated patterns in data, enhance recommendation accuracy, and effectively process various types of data.

Weakness: These models require large amounts of data and higher computational power.

Limitation: Training deep learning models may be a high time-consuming process, and it may have overfitting risks if not managed and controlled appropriately.

2.4. Reinforcement Learning-Based Model

The paper proposed by G. Zheng et al. [21] suggests a reinforcement learning framework that is based on DQN for online personalized news suggestions. In contrast to earlier approaches, this system effectively models changing news characteristics and user preferences, allowing it to anticipate the future and yield greater payouts (like CTR) in the long run. The studies [33,34] proposed unique framework called DEERS that uses reinforcement learning to model the recommendation session as a Markov Decision Process. They suggested learning approaches for how to automatically identify the best recommendation methods. In [34], the researchers present Deep Page, a unique page-wise recommendation system that uses Deep Reinforcement Learning to simultaneously optimize a page of items and automatically learn the best recommendation tactics. Comprehensive tests are conducted using information from a real e-commerce company to demonstrate the efficacy of our system [34].

In the study [35]. Pseudo Dyna-Q (PDQ) is proposed by the researchers. In PDQ, the World Model is used, a customer simulator that is intended to imitate the environment and handle the selection bias of logged data, in place of dealing with actual consumers. Continually updated and adaptively optimized throughout policy improvement, it uses the most recent suggestion policy [35]. To solve these issues, the study suggests a brand-new *Attribute-aware Neural Attentive Model* (ANAM). To model users' changing appetite for goods, ANAM implements an attention mechanism and employs a hierarchical architecture.

Lei and Li [36] described an interactive recommender system that employs user-specific deep reinforcement learning. This system can generate personalized recommendations in real-time and adjust to the user's feedback and preferences through interaction. The authors [30] reviewed the models in detail and present evaluation outcomes that show the efficacy of the proposed technique in contrast to other advanced recommendation methods. The analysis suggests that the interactive recommender system that employs user-specific deep reinforcement learning has the potential to enhance the accuracy of recommendations and improve the user experience.

The following are the strengths, weaknesses, and limitations of reinforcement learning:

Strength: It is very much effective for optimizing user engagement in changed environments where real-time interactions are required.

Weakness: The implementation of reinforcement learning may be a complex process and it may require a noteworthy amount of user interaction data.

Limitation: It may have lower efficiency with respect to sparse feedback, and it may complicate the whole learning process.

3. Issues with Recommender Systems

There are several problems that industries encounter when attempting to deploy a recommender system on a large scale. A few of these difficulties are addressed in Table 1 of this section.

Table 1. Problems with recommender systems and corresponding important findings.

Problem Name	Authors	Important Findings
1. <i>Cold Start Problem</i>	[20,21,24,26,29–34,36–38]	<ul style="list-style-type: none"> We do not have sufficient information to determine a user's choice for a certain item whenever a new product is added to the platform or a new user registers. If we add a new product and no customers have given it a review as of yet, or if just a negligible number of people have seen the latest movie as of yet, we do not have enough data on the brand-new item. Because of a lack of information, assumptions are made incorrectly, which results in poor recommendations. The effectiveness of customized suggestions increases whenever they have adequate product information and user interaction.
2. <i>Overspecialization Problem (OPS)</i>	[39–45]	<ul style="list-style-type: none"> The OPS becomes evident when the recommended items have a high degree of similarity with each other. The user is bombarded with only a few types of options, in which the user has already shown interest. This leads to the creation of an echo chamber of content. Because of the repeated recommendations of the same items, the user experience deteriorates.
3. <i>Scalability Problem</i>	[46–51]	<ul style="list-style-type: none"> The recommendation system's operation necessitates complicated multidimensional matrix multiplications as well as heavy computations, like discovering an item or user's nearest neighbors. More and more resources are needed as a platform's user base grows to give users appropriate recommendations. The scaling issue becomes so much more significant when considering the affordability and use of recommender system (RS).
4. <i>Sparsity Problem</i>	[52–55]	<ul style="list-style-type: none"> We may have very few ratings from these users regarding many products they have loved, bought, or even despised. Users regularly do not spend enough time giving correct reviews; occasionally, they may provide a random rating without even realizing what kind of item it is. Unwanted and inaccurate results are displayed to the user whenever the RS accepts this incorrect ranking as an input.
5. <i>Insufficient Data Problem</i>	[56–58]	<ul style="list-style-type: none"> The fact that recommender systems require a lot of information to create suggestions is likely their largest problem. The likelihood of receiving useful recommendations would increase with the amount of items and user data a recommendation system has at its disposal.
6. <i>Changing User Preferences</i>	[59–61]	<ul style="list-style-type: none"> The user's preference varies occasionally. A person might visit the same e-commerce site one day to look for a book and another day to look for home goods. Therefore, predicting whatever the user would want to buy is difficult.
7. <i>Gathering Known Ratings for Matrix</i>	[62–64]	<ul style="list-style-type: none"> The majority of users, it has been found, do not rate anything. Therefore, the question of how to determine whether and to what extent people are satisfied with the product emerges. We can extrapolate from the users' choices for other items how users will rate a certain item. This technique is referred to as an implied approach to rating collection.
8. <i>Unpredictable items</i>	[65,66]	<ul style="list-style-type: none"> To determine the sorted list of predictable and unexpected lists of items, an item recommender is utilized. This could include unique products that have existed before or are related to those that already exist; in this case, the system might not be happy to suggest a suitable product to any users, and all those items would end up being discarded for a long time.

4. Proposed Model for Improvement of Existing Solutions

We use an NLP-based pre-processing module on the reviews provided by the users to enumerate the ratings based on their usefulness. This module will also remove any fake or inconsistent ratings. The recommender systems' trained data after applying NLP will have cleaner data for training the machine learning models. Hence, the accuracy of the overall model will increase by using the following algorithmic steps:

The algorithmic steps of Figure 2 can be expressed using a diagram (Figure 3) of different components with a flow of the proposed approach [39,40].

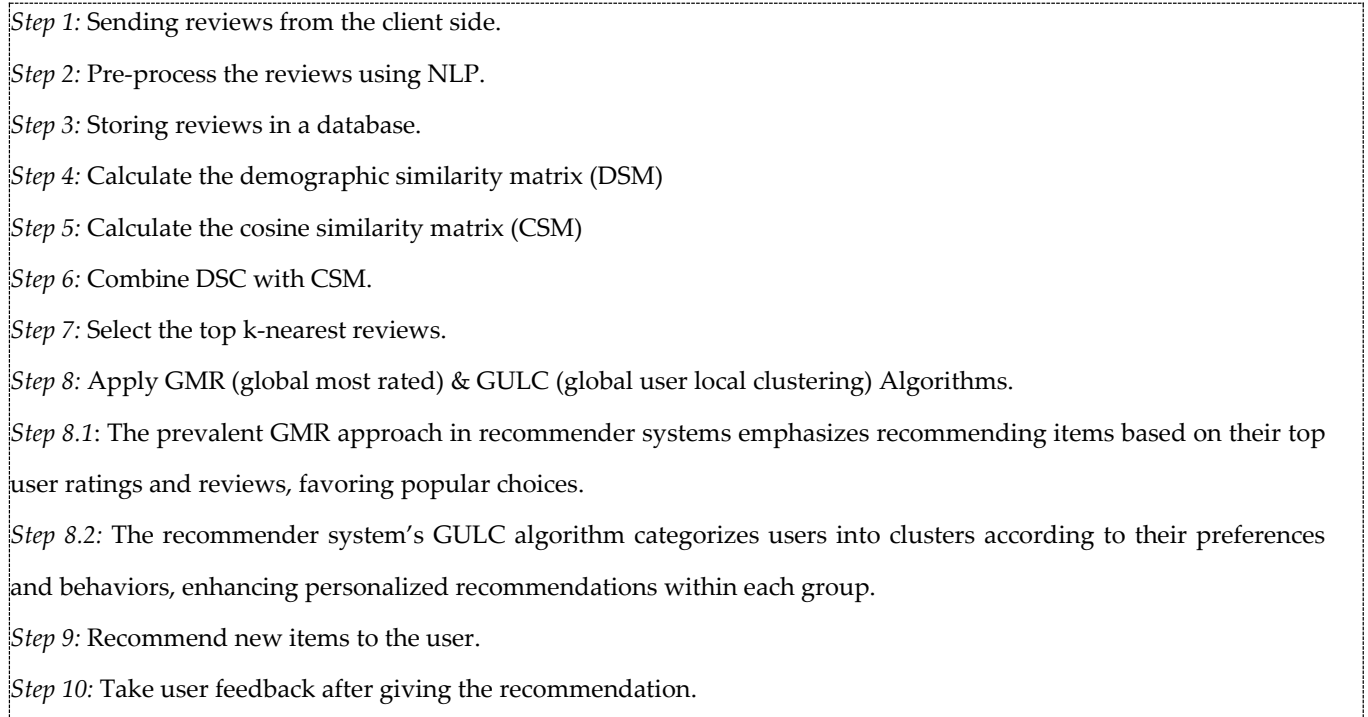


Figure 2. Algorithmic steps of the proposed approach.

4.1. Pre-Processing the User Review and Meta-Data Submitted by the Users

We pre-process the information gathered from the user. This includes their review of an item and some metadata about the user demography. These data will be further used to give the actual recommendations. Using this method, we can recognize the types of views and ideas the review text is conveying, and it can be determined whether the given review is positive or negative. This can further differentiate real reviews from fake reviews. The diagram (Figure 4) represents the general flow of ReSA [41–43]:

The ReSA process can be divided into two phases, namely, *Phase-A* (Pre-process the User-Review-Text), and *Phase-B* (Feature Extraction and Create Document Vectors).

In *Phase-A*, the collected user review goes through a series of operations to fit for ReSA. If we try to include the text and predict its legitimacy, the results obtained will not be satisfactory. Therefore, the contents can be processed [43,44]. Figure 5 represents the flow of review pre-processing for the ReSA approach. The pre-processing involves the following steps:

- Step 1: Input the user review text as *raw_text_data*.
- Step 2: Start Text_Pre-processing using the steps from Step 3 to Step 7.
- Step 3: Remove punctuation marks from *raw_text_data*.
- Step 4: Filter out numbers from *raw_text_data*.
- Step 5: Remove stop words from *raw_text_data*.
- Step 6: Convert all text of *raw_text_data* to lowercase.
- Step 7: Perform stemming on words.

Step 8: Obtain the cleaned_text_data as output.

Step 9: End the text pre-processing task.

Figure 6 describes the flow of preprocessed user review for ReSA approach where indexed data is obtained as the output. Finally, we feed in index data containing the bag of words and create a document vector from the same dataset. This document vector will be used for the processing of the data.

4.2. Proposed Mathematical Model

In this section, the mathematical model is formulated for describing the process of recommendations along with NLP in a collaborative-filtering-based recommender system. The proposed mathematical model is based on the following hypotheses:

H_{yp1}: Pre-processing user reviews will increase the accuracy of recommendations.

H_{yp2}: The user might have given the wrong review.

H_{yp3}: The user might have given the right review.

H_{yp4}: The user might have given a satisfactory review.

H_{yp5}: The accuracy of recommendations might be poor.

H_{yp6}: The accuracy might be satisfactory.

H_{yp7}: The accuracy might be good.

H_{yp8}: The accuracy may be very good.

H_{yp9}: The rate of increase in the number of positive reviews is proportional to the accuracy of predictions.

H_{yp10}: The accuracy might be outstanding.

H_{yp11}: The accuracy might be very poor.

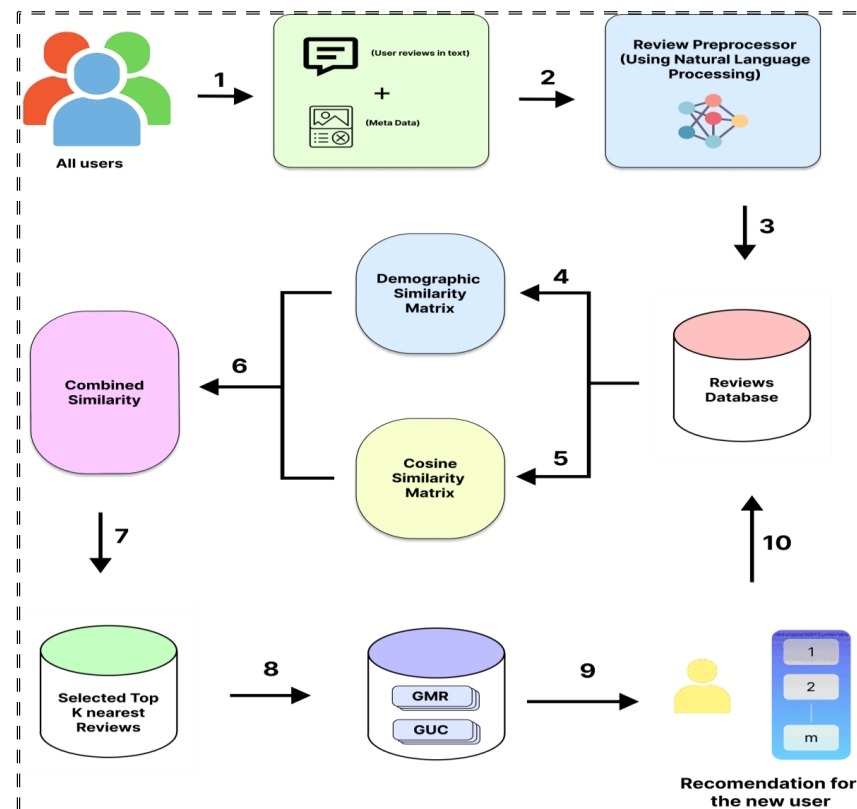


Figure 3. Proposed solution to further improve the solution for the cold start problem.

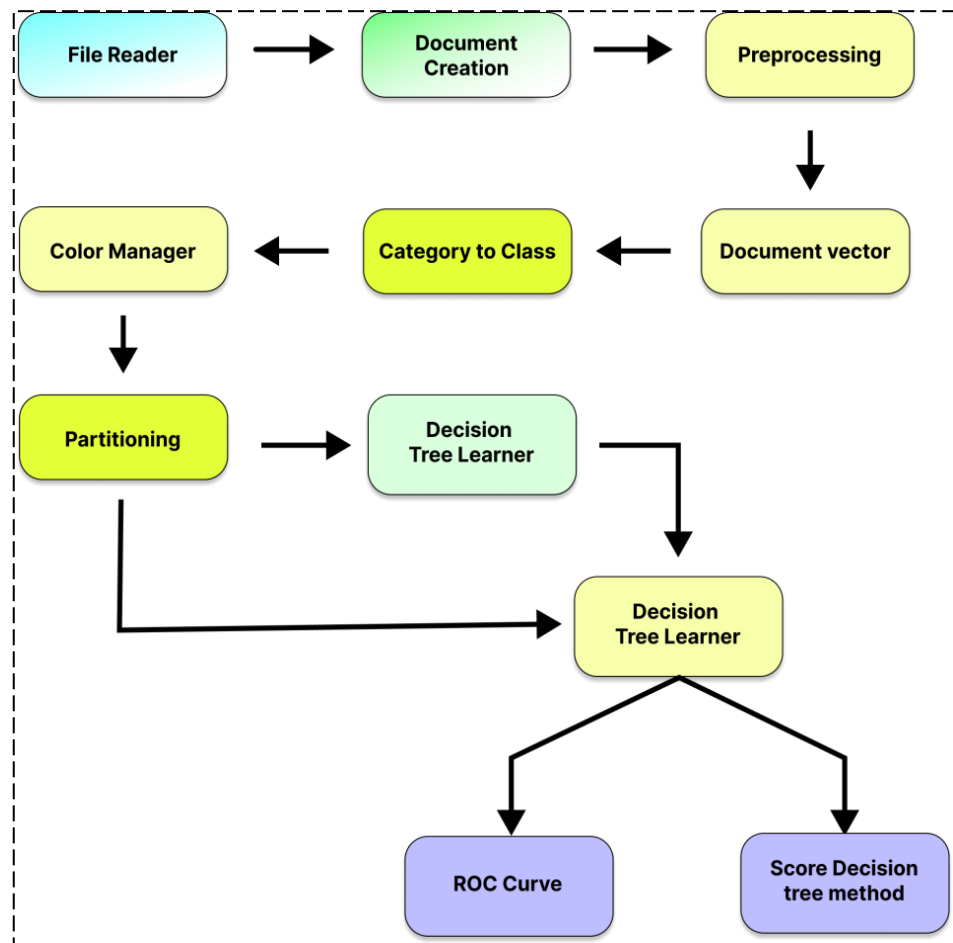


Figure 4. Sentiment analysis from textual review.

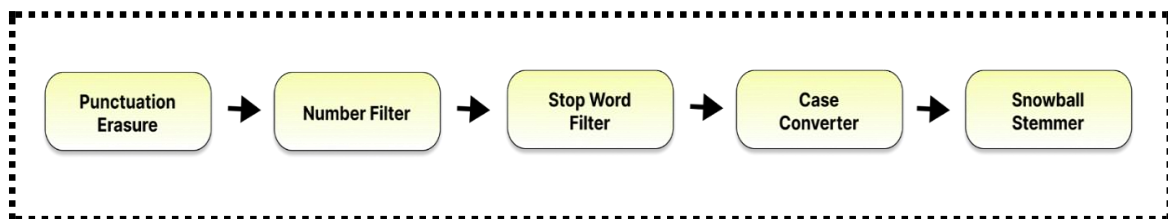


Figure 5. Flow of review "Pre-processing" for ReSA.

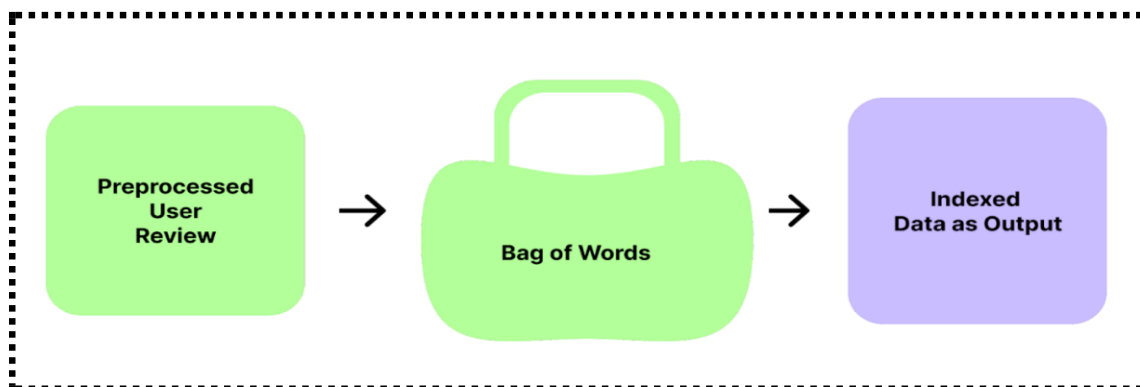


Figure 6. Flow of Preprocessed User Review for ReSA.

Further, Table 2 presents and describes the variables employed in the mathematical model.

Table 2. Nomenclature of the variables used in the research work.

	Symbol	Description
1.	X	The number of accurate recommendations made by the model.
2.	I	The number of inaccurate recommendations made by the model.
3.	U_1	Prediction accuracy after Phase A of review pre-processing.
4.	M	Already given inaccurate recommendations.
5.	U_2	Prediction accuracy after Phase B of the review pre-processing.
6.	V	The number of recommendations changed from incorrect to correct.
7.	E	Data generation rate.
8.	η_1	Rate of general filtering of user reviews.
9.	η_2	Rate of filtering after passing <i>ReSA Phase A</i> .
10.	η_3	Rate of filtering after passing <i>ReSA Phase B</i> .
11.	γ_1	Transfer rate of positive review to negative review at Level 1.
12.	γ_2	Rate of transfer of positive review to filtered data.
13.	δ	Change in the rate of incorrect recommendations.
14.	α_1	Transfer rate of positive review to already filtered data.
15.	α_2	Transfer of positive review to negative review block where a negative review was already stored.
16.	ω_1	The rate at which filtered data are transferred to vulnerable data.
17.	ω_2	Rate of data transfer from filtered to recovered data.
18.	Ψ	The rate of filtered data recovery following NLP.
19.	Z	A matrix with Jacobians for the frequency of additional transitions between filtered data blocks and various blocks.
20.	J	A matrix with the Jacobians for the frequency of fresh reviews appearing inside the filtered data blocks.
21.	M_{RFE}	Jacobian at the equilibrium of review pre-processing.
22.	τ_{ESP}	τ at the Equilibrium at a final block of processing.

Through the above hypothesis (H_{yp1} to H_{yp11}) and Figure 7 of the recommender system review processing, the authors have formulated the equations and theorems described in Appendix A of this paper. Further, the asymptotically stable state described in Appendix A may be useful in rejecting the obtained cold start spam reviews. Hence, the mathematical equations (Equation (A1) to Equation (A17)) of Appendix A are directly connected with various datasets used in conducting experiments in terms of feature matching (feature-based review selection and feature-based fake review rejection). Descriptions of various datasets and the implementation of the proposed ReSA process are presented in the next section (Section 5).

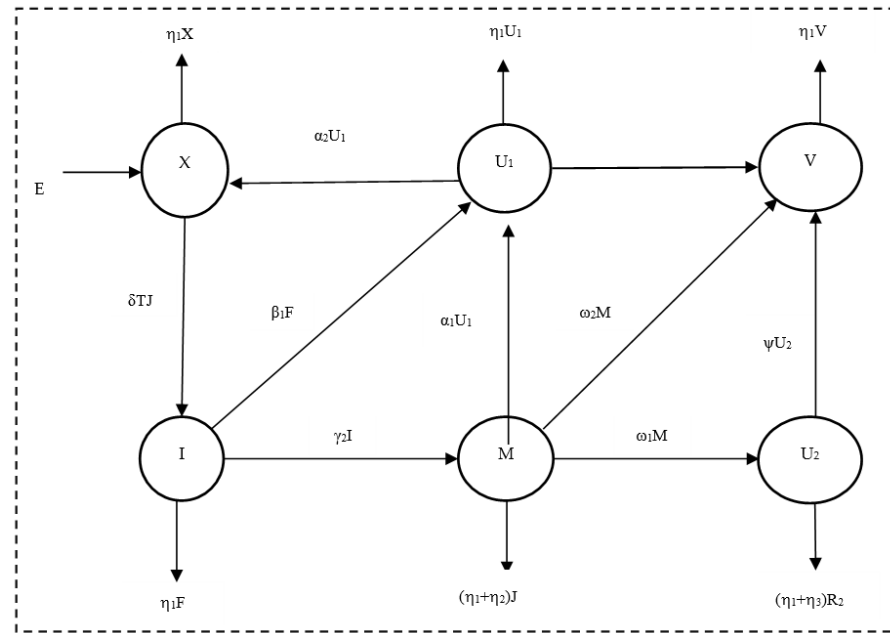


Figure 7. Schematic data flow representing the recommender's system equilibrium states.

5. Results and Discussions

5.1. Datasets Used

The following are the datasets used for enhancing the solutions of cold start problems using recommender system-based machine learning approaches:

MovieLens: This dataset consists of movie ratings provided by users on the MovieLens website and is frequently used to evaluate recommender systems.

Jester: The Jester dataset contains user ratings of jokes on the Jester website and is commonly used to evaluate collaborative filtering-based recommender systems.

Book-Crossing: This dataset comprises book ratings and metadata provided by users of the Book-Crossing website and is often used to evaluate content-based and hybrid recommender systems.

Last.fm: This dataset includes the music listening histories of users on the Last.fm website and is frequently used to evaluate collaborative filtering-based recommender systems.

Amazon Product Reviews: This dataset contains product ratings and reviews provided by users on the Amazon website and is widely used to evaluate recommender systems.

Yelp: The Yelp dataset contains user reviews and ratings of businesses on the Yelp website and is often used to evaluate content-based and hybrid recommender systems.

Netflix Prize Dataset: Renowned for movie recommendations, it contains user ratings for films.

Goodreads Dataset: Invaluable for book recommendations, it is filled with book reviews and ratings.

IMDb Data: Key for media recommendations, it contains user ratings and reviews for movies and TV shows.

CiteULike Dataset: Perfect for academic paper recommendations, it includes academic papers, tags, and user profiles.

Epinions Dataset: Valuable for trust-aware recommendations, it contains user reviews and trust networks.

Etsy Dataset: Beneficial for item recommendations, it comprises user interactions, product listings, and user profiles.

These datasets, varying in size and focus, are essential resources for researchers and practitioners in recommendation systems, tailored to specific research objectives and evaluation criteria.

Field 1: Date of review (which represents the date on which the review was conducted).
Field 2: URL of the product (which represents the Uniform Resource Locator).
Field 3: Review title (which represents the title of the review).
Field 4: Author (which shows the author's name).
Field 5: Rating (which shows the given rating).
Field 6: Review text (which shows the description of the review).
Field 7: Review Sentiment (which represents the linked sentiments related to the review).
Field 8: Review Subjectivity (which represents the mental depth or the seriousness of the review).

5.3. The Logic of the Study's Instrumentation

The following *criteria, indicators, and methods* are instrumentation for our study:

Criteria 1: Mitigation of Cold Start Issues (evaluating how effectively the system generates recommendations for new users or items);

Criteria 2: Performance Metrics (utilizing accuracy, precision, recall, F1-score, and area under the curve to measure model effectiveness);

Criteria 3: Collaborative Filtering Efficacy (assessing the success of user- or item-based filtering techniques);

Criteria 4: NLP Integration (analyzing the impact of text-based reviews in sentiment assessment);

Indicator 1: Effectiveness in Cold Start Resolution (analyzing the quality of initial recommendations);

Indicator 2: Model Evaluation Standards (employing metrics like confusion matrices and AUC for assessment);

Indicator 3: Scalability Considerations (the model's adaptability across various datasets and domains, such as *MovieLens* and *Yelp* etc.);

Indicator 4: Reproducibility Aspects (making datasets and algorithms publicly available for verification);

Method 1: Data Collection (gathering datasets from sources like *IMDb* or *Amazon* for NLP applications);

Method 2: Implementation of Models (using machine learning techniques such as random forest and Naive Bayes);

Method 3: Assessment of Performance (evaluating models through confusion matrices and loss graphs);

Method 4: Visualization Techniques (employing heatmaps and performance charts to illustrate optimization and sentiment analysis).

This structured approach will support the investigation of the cold start problem, enhancing NLP and machine learning solutions while ensuring scalability.

5.4. Results of the Proposed ReSA Process

Numerous factors have a substantial impact on the accuracy of CFBRs. Data sparsity is a prevalent challenge in collaborative filtering, where a significant proportion of users or items possess limited or no interaction data, particularly affecting niche or less popular items. The cold start problem compounds these challenges by emerging when new users or items lack historical interaction data. Collaborative filtering heavily depends on past interactions, rendering recommendations for these "cold start" entities a formidable task. Scalability concerns arise as the user and item populations grow, elevating computational complexity and affecting recommendation speed and efficiency. Data quality is pivotal, with inaccurate or noisy user data, including fake reviews or erroneous ratings, significantly impairing recommendation quality. Detecting and addressing outliers and data inconsistencies is crucial.

Malicious users can manipulate systems through spam and shilling attacks, providing biased ratings or reviews, necessitating the detection and mitigation of such activities. Privacy concerns may result in users withholding preferences, leading to incomplete or

unreliable preference data, undermining recommendation accuracy. Diverse user preferences, termed user heterogeneity, pose challenges, as collaborative filtering assumes user similarity. Varied tastes and preferences can lead to less precise recommendations. Collaborative filtering often exhibits a bias towards popular items, introducing item popularity bias and potentially neglecting niche or lesser-known items. Algorithm scalability is a concern, with some algorithms struggling to process large datasets efficiently. Careful algorithm selection and optimization techniques are vital. Sparse user–item interactions make identifying similar users or items challenging, impacting recommendation accuracy. User preferences evolve over time, affecting recommendation relevance in temporal dynamics. Contextual information, such as location, time, or user behavior, significantly impacts recommendations. Traditional collaborative filtering may not consider these factors, affecting accuracy. Algorithm selection, with different collaborative filtering algorithms having distinct strengths and weaknesses, influences recommendation accuracy. Addressing these factors is crucial to enhancing CFBRs accuracy.

Experiments are conducted on the self-generated and standard benchmark datasets to obtain positive and negative review distribution values. These values are given in Figure 10. After analyzing the results of Figure 10, the authors observed that approx. 80% of the reviews are positive and the rest, 20% of reviews, are negative. Further, it has been observed that there are more positive reviews and fewer negative reviews for the self-generated and benchmark datasets.

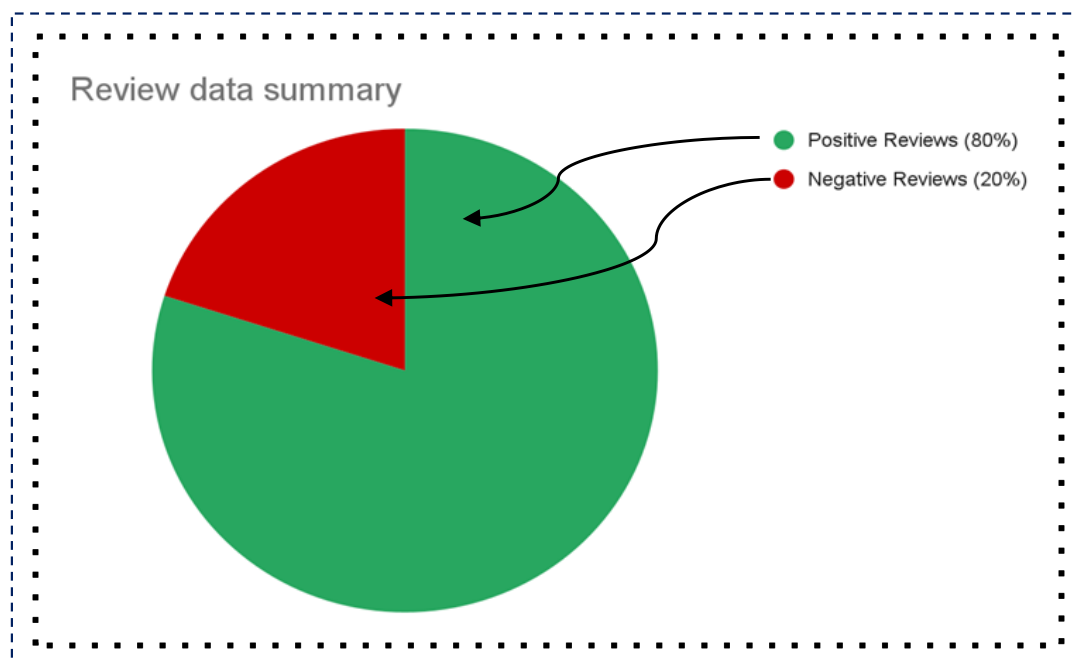


Figure 10. Pie chart of positive and negative review distributions.

5.5. Comparison of Algorithms Using Confusion Matrices

Figure 11 is a description of the performance evaluation of the ReSA process for the improved method to form the recommender cold start problem using machine learning. Two commonly used confusion matrices used in this paper to evaluate performance are the random forest confusion matrix and the Naive Bayes confusion matrix. The random forest confusion matrix in Figure 11 displays the number of true positive (TP), false positive (FP), true negative (TN), and false negative (FN) values obtained by the random forest model. TP represents the number of correctly predicted positive reviews, while FP represents the number of incorrectly predicted positive reviews. TN represents the number of correctly predicted negative reviews, while FN represents the number of incorrectly predicted negative reviews.

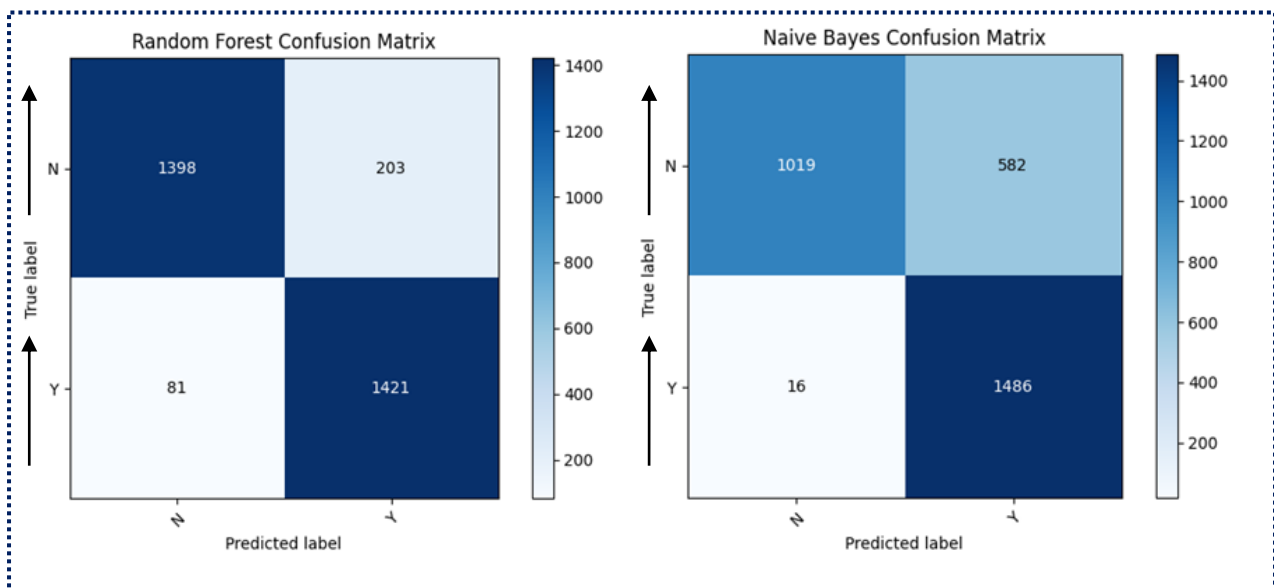


Figure 11. Confusion matrices comparing the performance after the ReSA process.

The Naive Bayes confusion matrix of Figure 11 displays the performance of the Naive Bayes model in predicting the sentiments of reviews. Similar to the random forest confusion matrix, it displays the number of TP, FP, TN, and FN. However, the Naive Bayes model is based on probabilistic principles, which allows it to take into account the probability of each feature in the review being indicative of a positive or negative sentiment [49,50].

By comparing the performance of the ReSA method using both the random forest and Naive Bayes models, it is possible to determine which model is more suitable for the given task. If the random forest model is more accurate, its confusion matrix will show a higher number of TPs and TNs and a lower number of FPs and FNs. Conversely, if the Naive Bayes model is more accurate, its confusion matrix will show a higher F1 score and recall. Other evaluation metrics such as accuracy, precision, recall, and F1-score can also be used to compare the performance of the ReSA method using different models, providing a more detailed view of the model's strengths and weaknesses in predicting the sentiment of reviews.

Figure 11 shows two confusion matrices, one for a *random forest model* and another for a *Naive Bayes model*. These matrices are used to evaluate the performance of classification models.

The following are the key elements of these matrices:

Quadrants: True positive (TP) means correctly predicted positive instances, true negative (TN) means correctly predicted negative instances, false positive (FP) means incorrectly predicted positive instances, and false negative (FN) means incorrectly predicted negative instances.

Labels: "N" represents the *predicted label*, and "Y" represents the *true label*.

Color Scale: A blue-to-white gradient indicates the frequency of observations in each quadrant.

By comparing the distribution of values in the two matrices, we can find the relative performance of both the approaches for classifying the data. Random forest has a *higher precision* and *higher F1 score* than the Naïve-based approach. This means that it is more likely to correctly and equally identify positive instances. Therefore, based on the calculated *precision values* and *F1 score* of Figure 11, it can be said that *random forest* can be a better choice than the *Naïve-based approach* for this particular classification task. But, the best model depends on the specific requirements and trade-offs involved in the application. We may consider the relative importance of precision, recall, F1 score, and accuracy in each context to make an informed decision.

Heat map generation is a powerful visualization tool that can help to identify patterns and relationships in data. In the context of solving the cold start problem using the users' review of the concept of machine learning, a heat map like Figure 12 can be generated to visualize the relationship between the sentiment of user reviews and the performance of the recommendation system [51].

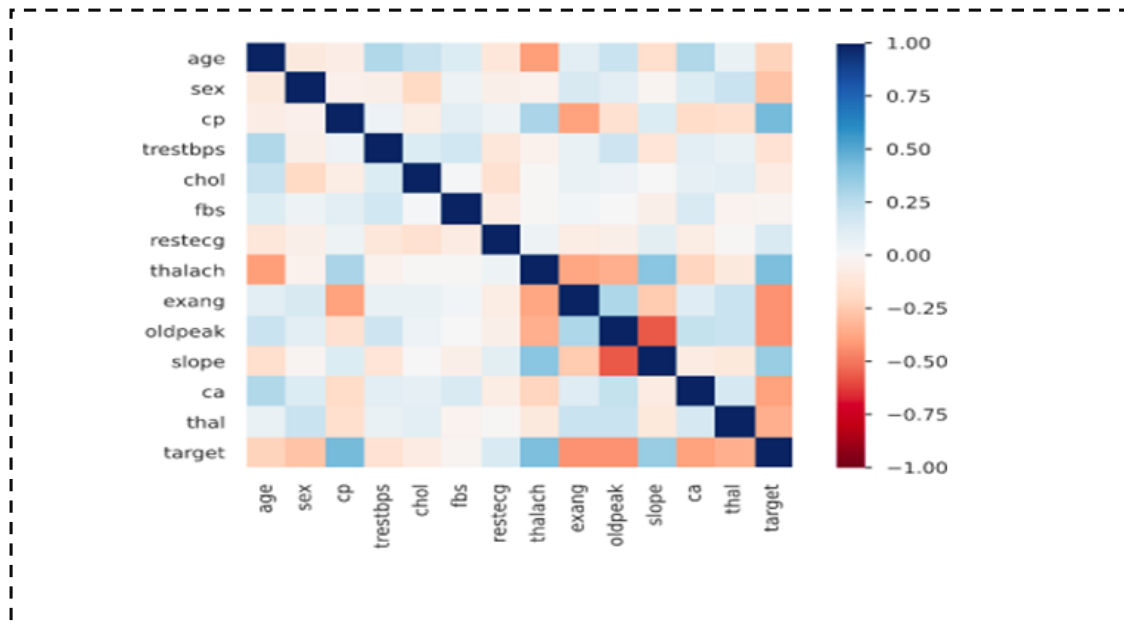


Figure 12. Generating heat-map using Naïve Bayes classification.

Figure 12 describes a correlation matrix, showing the relationships between features within a dataset. The color gradient used in the heatmap visually represents *correlation strength* and *direction*. Dark blue corresponds to a strong negative correlation, while dark red indicates a strong positive correlation. The labels on the 'X' and 'Y' axes mark the dataset's features, with correlation coefficients displayed within each cell to measure these relationships. The analysis of the heatmap identifies three main types of correlations: *strong positive*, where both features increase together; *strong negative*, where one feature decreases as the other increases; and *weak correlations*, indicating minimal or no association between features. These values of *strong positive*, *strong negative*, *weak correlations*, *label 'X'*, and *label 'Y'*, discussed in Figures 11 and 12, are used in computing the precision score, the recall value, the F1 score, and the accuracy.

The heatmap also highlights key aspects like correlations with the target variable, which are particularly significant in predictive modeling tasks. Another important factor discussed in Figure 12 is multiple feature-based correlations, where multiple features exhibit strong correlations with each other.

Further, the heat map of Figure 12 is divided into different sections that represent different levels of sentiment, such as highly positive, moderately positive, neutral, moderately negative, and highly negative. The performance of the proposed ReSA recommendation system can be represented using different colors or shades, with darker shades indicating better performance and lighter shades indicating poorer performance. By analyzing the heat map of Figure 12, the authors have identified the patterns and relationships between the sentiment of user reviews and the performance of the recommendation system [53]. For example, we might find that the recommendation system performs well for highly positive reviews but poorly for highly negative reviews, or that the system performs well for reviews that use certain words or phrases but poorly for reviews that use other words or phrases. Further, Figure 13 shows the best score achieved after using several models

on our self-generated and standard datasets. The authors obtained the following results (Figure 13) using the random forest and Naïve Bayes approaches [54].

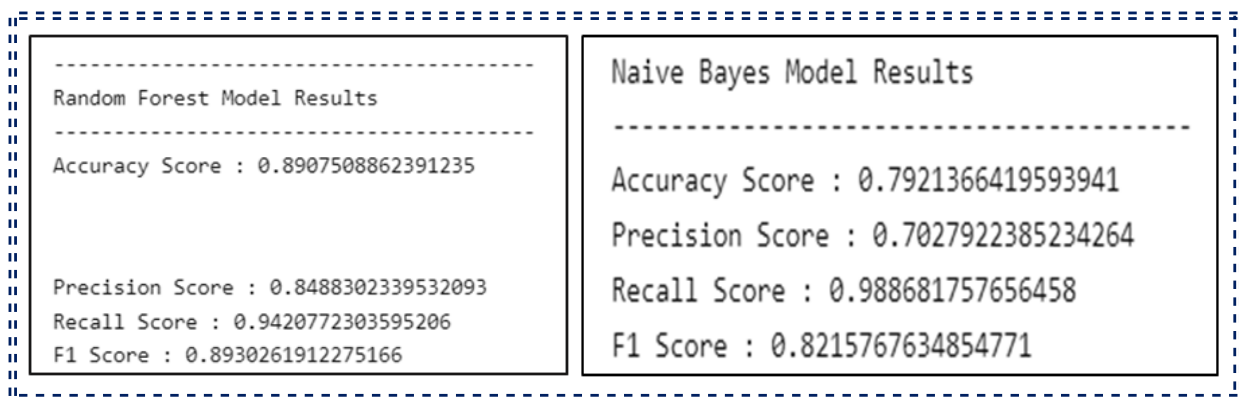


Figure 13. Accuracy, recall, and precision Scores for random forest and Naive Baye’s models.

The graph of Figure 14 represents the training loss versus epoch for different optimizers used in the recommenders’ cold start problem using a machine learning approach. The x-axis of Figure 14 shows the number of epochs, while the y-axis shows the training loss. The graph is composed of multiple lines, each corresponding to a different optimizer [67]. The model/training loss measures how well the model can fit the training data. A lower model/training loss indicates a better fit. The graph of Figure 14 shows how the training loss changes over time as the model is trained for more epochs. The different lines in the graph of Figure 15 represent the training loss for different optimizers, such as *Adam*, *RMSProp*, *SGD*, and so on. By comparing the lines of Figure 14, the authors can see which optimizer produces the best training loss. In general, a lower training loss indicates better performance of the model [55].

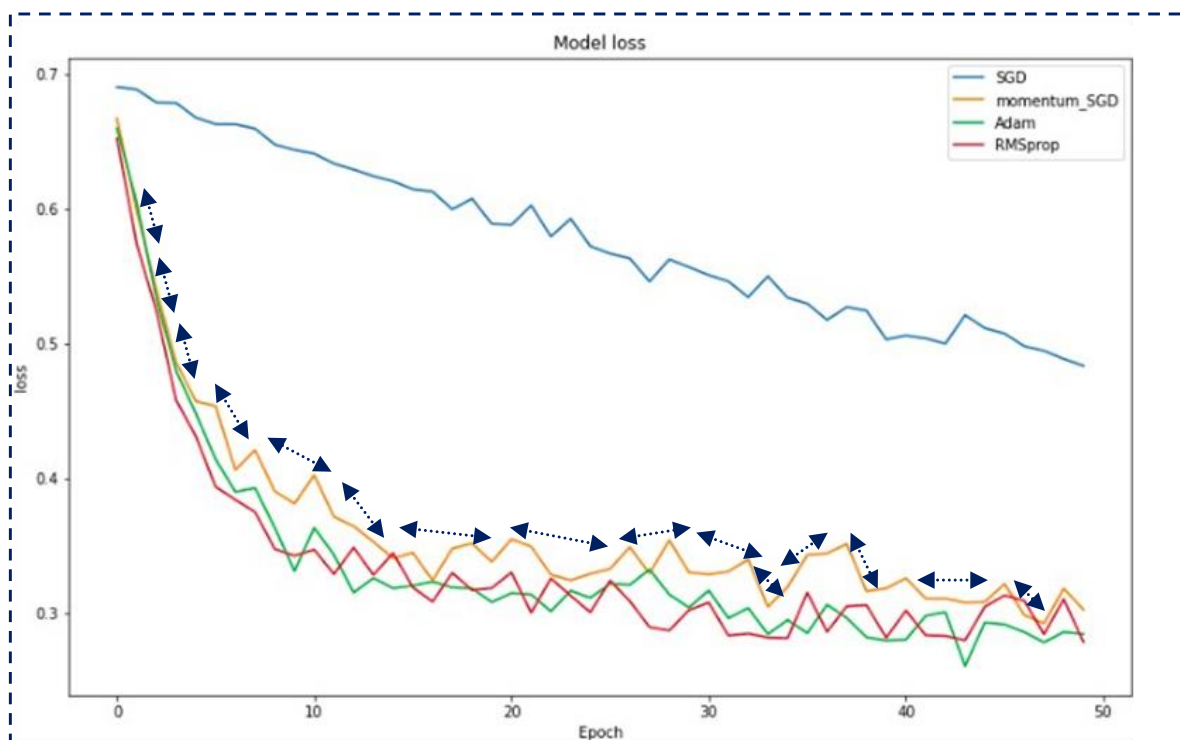


Figure 14. Generating a graph for training loss vs. epoch using the different optimizers.

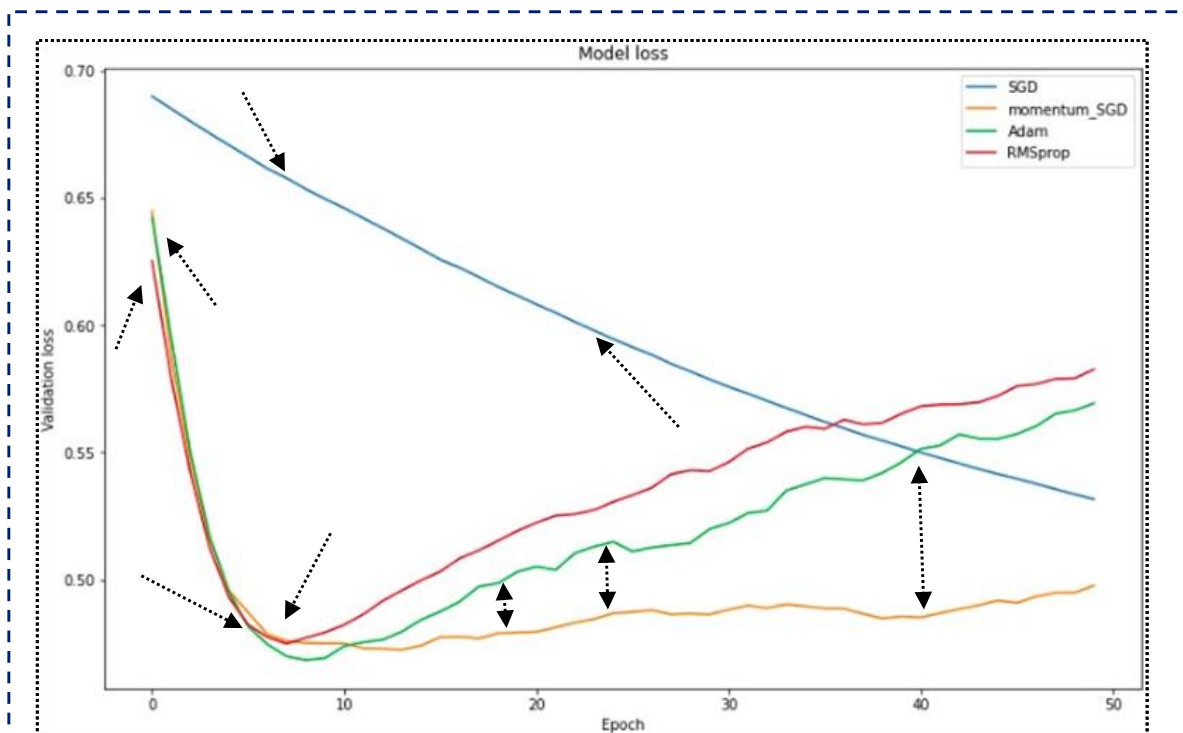


Figure 15. Generating a graph for validation loss vs. epoch using the different optimizers.

The validation loss vs. epoch graph is an essential tool for assessing and selecting the best optimizer for a given task. The validation loss vs. epoch graph of Figure 15 displays the performance of various optimization algorithms in training a model for solving the cold start problem in a machine learning environment while minimizing errors [56]. The y-axis of the graph of Figure 15 represents the validation loss, which indicates the model's error rate on a validation dataset at each epoch during the training process, while the x-axis represents the epoch number. The primary objective is to reduce the validation loss over the epochs to enhance the accuracy of the model. By plotting the validation losses of different optimization algorithms on the same graph, the graph of Figure 15 allows for easy comparison and identification of the superior performing algorithm. The graph of Figure 15 shows trends in validation loss over time, which helps to detect the overfitting or underfitting of the model [57].

The AUC of Figure 16 is a performance metric for evaluating a cold start problem recommendation system in a machine learning environment. It is calculated by plotting the true positive rate (TPR) against the false positive rate (FPR) at various thresholds of the classifier. TPR represents the proportion of true positive recommendations, while FPR represents the proportion of false positive recommendations. The AUC ranges from 0 to 1, with a higher value indicating better performance [58]. A perfect model has an AUC of 1, while a random model has an AUC of 0.5. The AUC of Figure 16 measures the model's ability to differentiate between positive and negative recommendations. Along with other metrics such as precision, recall, and F1-score, it provides a comprehensive evaluation of the model's performance [59].

Table 3 presents the performance of three machine learning algorithms, namely Naïve Bayes, random forest, and Logistic Regression, on a specific task using four evaluation metrics, accuracy, precision, recall, and F1 score. Each row of Table 3 corresponds to a particular metric, while each column of Table 3 represents a specific algorithm. The values listed in the table indicate the scores achieved by each algorithm for each metric. For instance, Naïve Bayes obtained an accuracy score of 0.792136, a precision score of 0.702792, a recall score of 0.988681, and an F1 score of 0.9829470. Similarly, random forest achieved an accuracy score of 0.890750, a precision score of 0.848830, a recall score of 0.942077, and an

F1 score of 0.893026, while Logistic Regression achieved an accuracy score of 0.9464930, a precision score of 0.9457630, a recall score of 0.9357150, and an F1 score of 0.9403620. Finally, Table 3 provides a comparative view of the performance of these algorithms, allowing us to evaluate which algorithm is the most effective for the given task.

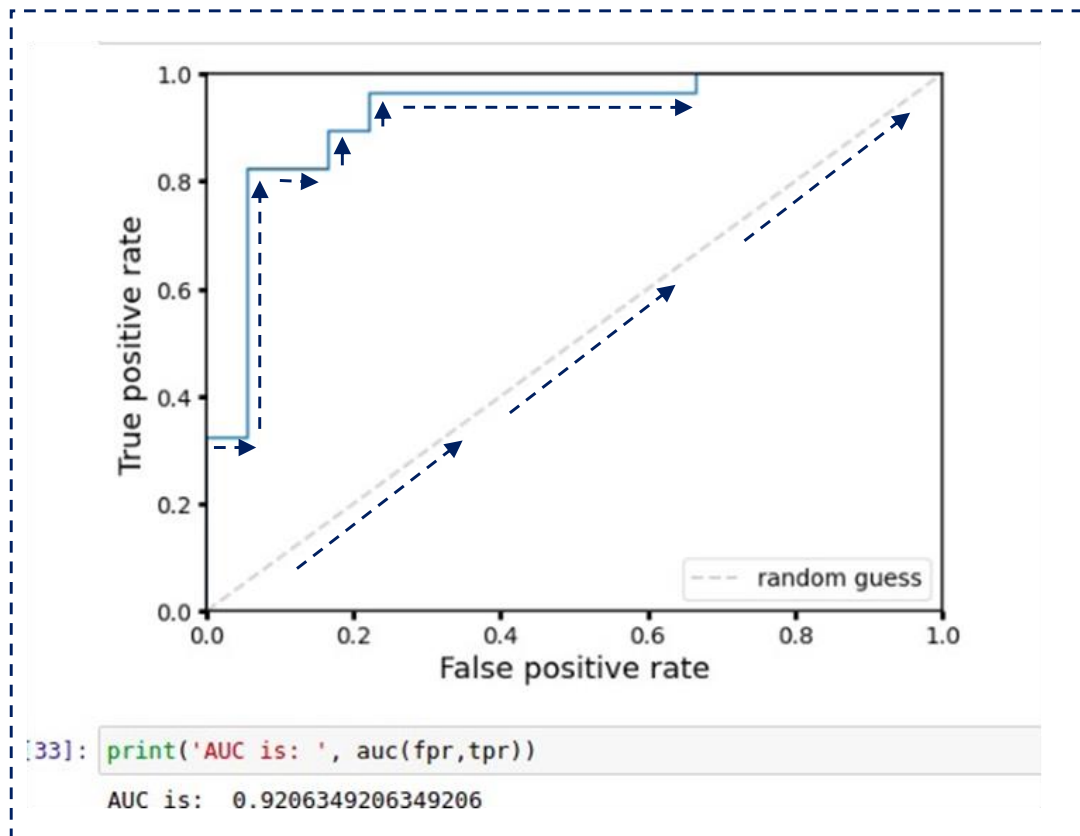


Figure 16. Generation of area under curve (AUC) for the proposed model.

Table 3. Result summary of different algorithms used for proposed ReSA.

	<i>Factors Considered</i>	<i>Naïve Bayes</i>	<i>Random Forest</i>	<i>Logistic Regression</i>	<i>Best Score</i>
1.	Accuracy	0.792136	0.890750	0.9464930	Random Forest
2.	Precision	0.702792	0.848830	0.9457630	Random Forest
3.	Recall	0.988681	0.942077	0.9357150	Naive Bayes
4.	F1 Score	0.9829470	0.893026	0.9403620	Naive Bayes

Table 4 describes the results of various algorithms used for addressing the cold start problem in the domains of blockchain, smart grid, and edge computing. Further, Table 4 lists the performance metrics of 11 different algorithms, out of which the first four and the Hybrid Storage architecture do not have available results for global accuracy, global loss, F1-score, and AUC. The other six algorithms, including Model Chain, Block Deep Net, Deep Coin, Deep Block Scheme, and the Asynchronous advantage actor–critic algorithm, have varying levels of performance in terms of global accuracy and global loss. Finally, the proposed approach that utilizes machine learning for user review pre-processing has achieved a global accuracy of more than 96% and an F1-score of 0.78 and above, with an AUC of 92% and above for almost each of the datasets taken, outperforming the other algorithms. Hence, this approach can be effective in mitigating the cold start problem by utilizing machine learning approaches.

Table 4. Performance comparison of various security provision approaches for edge computing on various datasets [12–36].

	<i>Algorithm Name</i>	<i>Classifier Taken/Technologies Used</i>	<i>Results/Observation</i>			
			<i>Global Accuracy</i>	<i>Global Loss</i>	<i>F1-Score</i>	<i>AUC</i>
1.	SDN Based on Blockchain [12,13]	S.D.N., Blockchain, Fog Nodes	87%	3%	Not Available	Not Available
2.	Edge Chain [14,15]	Edge Computing, Blockchain	Not Available	Not Available	Not Available	Not Available
3.	Smart Grid [16,17]	Smart Grids, Edge Computing, K-Nearest Neighbors	91%	8%	Not Available	Not Available
4.	Hyper Ledger Fabric Blockchain [18,19]	Edge Computing, Hyper ledger Fabric, IoT	Not Available	Not Available	Not Available	Not Available
5.	Hybrid Storage Architecture [20,21]	Blockchain, smart computing, Edge Computing	Not Available	Not Available	Not Available	Not Available
6.	Model Chain [22,23]	Blockchain, Machine Learning	78%	7%	Not Available	Not Available
7.	Block Deep Net [24–26]	Blockchain, Deep Learning	72%	5.4%	Not Available	Not Available
8.	Deep Coin [27,28,30]	Deep Learning, Blockchain, Smart Grid	92%	3.98	0.761	Not Available
9.	Deep Block Scheme [29] [30–32]	Blockchain, Deep Learning	89%	2.65	0.697	Not Available
10.	Asynchronous Advantage Actor–critic Algorithm [30,33–36]	Blockchain, Edge Computing, Deep Reinforcement Learning	76%	1.91%	Not Available	Not Available
11.	Proposed Approach	An Improved Method to Recommender’s Cold Start by User Reviews Pre-processing Using Machine Learning	96%	1.5%	0.78	92%

The comparison between random forest and Naive Bayes in ReSA for sentiment analysis introduces an innovative approach to resolving the cold start problem. This method combines NLP with machine learning techniques, combining both algorithms and advanced evaluation tools such as confusion matrices, F1 scores, and AUC to improve sentiment prediction. The ReSA method's dependence on widely used algorithms and publicly available data ensures that its findings are reproducible. Its accuracy is demonstrated through various metrics like precision, recall, and validation loss graphs, with random forest consistently showing superior performance. Scalability is achieved by applying these machine learning models across diverse domains like blockchain and edge computing. The practical impact of this approach lies in its ability to enhance recommendation systems by addressing performance inconsistencies and mitigating cold start challenges. By doing so, it leads to significant improvements in system performance and user satisfaction. The method has been successfully tested across multiple datasets, including *MovieLens*, *Yelp*, *Amazon Product Reviews*, and *Netflix Prize*, demonstrating its flexibility and effectiveness in different applications. This comprehensive approach not only addresses technical challenges but also improves user experience on a wide range of platforms, making it a robust solution for modern recommendation systems across industries.

The proposed approach of Table 4 demonstrates several strengths when compared to other state-of-the-art approaches. The key strengths, and weaknesses of the proposed approach are described below:

Strength 1: Global Accuracy (the proposed approach achieves a 96% global accuracy, outperforming methods such as SDN, based on Blockchain (87%) and Block Deep Net (72%). Even the Deep Coin method, which reaches 92%, falls short. This high accuracy makes the proposed method particularly effective in providing accurate recommendations).

Strength 2: Global Loss (the method shows a global loss of 1.5%, significantly lower than other methods like Smart Grid (8%) and Deep Block Scheme (2.65%). This low loss highlights the model's precision and ability to minimize errors, which is critical in recommendation systems for better user experience).

Strength 3: F1-Score (reporting an F1-score of 0.78, the proposed approach balances precision and recall well. It slightly outperforms Deep Coin, which has an F1-score of 0.761. This makes the method particularly useful for applications involving imbalanced datasets or the cold start problem, where both precision and recall are critical).

Strength 4: AUC (with an AUC of 92%, the proposed method stands out, showing a strong ability to distinguish between positive and negative outcomes. This makes it more effective for classification tasks compared to other methods, such as SDN and Block Deep Net, which do not provide AUC values for direct comparison).

Weaknesses 1: Limited Use of Emerging Technologies (unlike methods like Edge Chain and Hyperledger Fabric Blockchain, the proposed method does not incorporate technologies like blockchain or edge computing, which are beneficial in decentralized and scalable systems. These methods provide added security, scalability, and data management capabilities, making them better suited for decentralized applications).

Weakness 2: Lack of Real-Time Capabilities (methods like Deep Coin and the Asynchronous Advantage Actor–Critic Algorithm use advanced technologies such as deep reinforcement learning and smart grids, which allow for adaptability in real-time, dynamic environments. The proposed method focuses primarily on cold start and user review processing, making it less versatile in contexts requiring real-time processing and decision-making).

Weakness 3: Absence of Blockchain Integration (Other approaches, such as Model Chain and Hybrid Storage Architecture, combine blockchain for secure, decentralized data handling. The proposed method does not include blockchain technology, which could limit its application in systems that require high security and decentralized data storage and processing).

It can be said that the proposed method excels in key areas like global accuracy, F1-score, and AUC, making it an effective solution for improving recommendation systems

and addressing the cold start problem. However, it lacks the integration of advanced technologies such as blockchain and edge computing. Despite this, the proposed method's superior accuracy and low loss rates make it a highly effective tool, particularly for recommender systems.

6. Conclusions

In this study, an NLP-based approach is applied to improve the quality of user reviews, which will lead to an improvement in the accuracy of recommendations made by the CFBRs. Various datasets obtained in this study contained reviews submitted by different users and pre-processed data, etc. Further, multiple techniques are applied, which include random forest, Naïve Bayes, Logistic Regressions, etc., to detect the quality of a review submitted by the user. They yielded different recall, precision, and accuracy scores after using different machine learning methods.

This study applied a mix of supervised and unsupervised approaches to NLP using the sci-kit learn method. The researchers evaluated their proposed method on several benchmark datasets, including MovieLens, Jester, Book-Crossing, Last.fm, Amazon Reviews, Yelp, Netflix Prize, Goodreads, IMDb, CiteULike, Epinions, and Etsy. They measured performance using metrics such as *Global Accuracy*, *Global Loss*, *Precision Score*, *Recall values*, *F1 score*, and *AUC*. The results from various models like random forest, Naïve Bayes, and Logistic Regression showed that random forest was the most effective, achieving over 90% accuracy. The proposed method recorded a global accuracy of 95%, a global loss of 1.50%, an F1 score of 0.78, and an AUC of 92%.

The results from this study can be further enhanced by doing further trials with other databases that may be gathered using techniques like web-scraping; the authors need to check the compatibility of dynamic programming dataset-based techniques (DPDBTs) with modern machine learning (MML) models.

This study has the following limitations:

- (i) The dataset was gathered from a small number of sources, lacking diversity.
- (ii) This study only concentrated on improving user review quality, ignoring other crucial factors.
- (iii) Other influential factors affecting CFBRs performance were not considered.

To address these limitations, following future research could explore the use of other machine learning techniques and models to improve the accuracy and efficiency of the proposed model:

- (i) Explore the integration of DPDBTs with advanced machine learning techniques, including neural networks, deep learning, and decision-based models, to further improve system accuracy.
- (ii) Implement distributed experiments to enhance performance across client and server environments.
- (iii) The MML models might be compared to DPDBTs in further study. On both the client and server sides, distributed and global differential privacy (GDP) experiments are possible. Hence, the authors may conduct GDP experiments in the DPDBT environment after including and adjusting various result-affecting parameters to further improve the performance of the proposed system.
- (iv) Collect diverse data and address privacy, data quality, and user engagement challenges to refine recommendations.
- (v) Incorporate state-of-the-art NLP frameworks, such as transformers and generators, to elevate analysis capabilities. Gather user feedback to fine-tune recommendation effectiveness.
- (vi) Create hybrid models that integrate collaborative and content-based approaches. Explore real-time data processing methods to boost user engagement and responsiveness.

Author Contributions: Conceptualization, K.N.M., A.M., P.N.B. and R.K.L.; Methodology, K.N.M., A.M., P.N.B. and R.K.L.; Software, K.N.M. and P.N.B.; Validation, K.N.M. and A.M.; Formal analysis, K.N.M., A.M., P.N.B. and R.K.L.; Investigation, K.N.M., A.M., P.N.B. and R.K.L.; Data curation, R.K.L.; Writing—original draft, K.N.M., A.M. and P.N.B.; Writing—review & editing, K.N.M., A.M., P.N.B. and R.K.L.; Visualization, K.N.M., P.N.B. and R.K.L.; Supervision, K.N.M. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by Norwegian University of Science and Technology, Trondheim, Norway.

Data Availability Statement: The data and material used in this paper are appropriately referred to and described in this paper.

Conflicts of Interest: Being the corresponding author, I declare that there are no conflicts of interest with any person or organization for this paper.

Appendix A

The formulated mathematical model for describing the process of recommendations along with NLP in a CFBRs can be represented by Equation (A1).

$$\begin{aligned}\frac{dX}{dt} &= E - \delta TM - \eta_1 X + \alpha_2 U_1 \\ \frac{dI}{dt} &= T\delta M - (\gamma_1 + \gamma_2 + \eta_1) I \\ \frac{dU_1}{dt} &= \gamma_1 I - (\alpha_1 + \alpha_2 + \eta_1) U_1 \\ \frac{dM}{dt} &= \alpha_1 U_1 - (\omega_1 + \omega_2 + \eta_1 + \eta_2) M + \gamma_2 I \\ \frac{dU_2}{dt} &= \omega_1 M - (\eta_1 + \eta_3 + \psi) U_2 \\ \frac{dV}{dt} &= \psi U_2 + \omega_2 M - \eta_1 V\end{aligned}\quad (A1)$$

Appendix A.1. Correlation Between Review Pre-Processing and Accuracy in the Proposed Model

The data and parameters associated with it are assumed to be non-negative. Thus, the rate of change in accuracy concerning the time in pre-processing the user reviews can be given by

$$\frac{dQ}{dt} = \frac{dX}{dt} + \frac{dI}{dt} + \frac{dU_1}{dt} + \frac{dM}{dt} + \frac{dU_2}{dt} + \frac{dV}{dt} \quad (A2)$$

$$\Rightarrow \frac{dQ}{dt} = E - \eta_1 Q - \eta_2 M - \eta_3 U_2 \quad (A3)$$

with $Q = X + I + U_1 + M + U_2 + V$.

Now, it can be concluded that whenever there is no error in recommendations, then ($I = M = U_2 = 0$).

Hence, $\frac{dQ}{dX} = E - \eta_1 Q$, which defined the size 'Q', having the capacity of $\frac{E}{\eta_1}$ As $t \rightarrow \infty$.

Furthermore, the solution of (1) is defined to lie in the range of $\tau = \{(X, I, U_1, M, U_2, V) \in R_6^+ : X(t) \geq 0, I(t) \geq 0, U_1 \geq 0, M(t) \geq 0, U_2(t) \geq 0, V(t) \geq 0, X + I + U_1 + M + U_2 + V \leq \frac{Q}{\eta_1}\}$, because of the positive boundness of the solutions given in the region that is defined by τ .

Appendix A.2. Review Pre-Processing Impact Magnitude (S_0)

We will be applying next-generation matrix manipulations to define the *Review Pre-processing Impact Magnitude (RPIM)* at "Phase A: User-Text-Review-based pre-processing". The RPIM is designated as the mean of all inaccurate re-accommodating after Phase A of review sentiment analysis, i.e., text-based pre-processing when the fake review from one user breached another user's recommended feed and polluted the filtered recommendations. We denote this number as S_0 . This RPIM can be formulated using the spectral radius of the JZ^{-1} matrix. We can obtain this by the linearization of the equation (Equation (A1)).

According to the next-generation matrix principle, RPIM can be deduced as the spectral radius of the next-generation matrix formed as JZ^{-1} of the system of equations (Equation (A1)).

$$j_i - z_i = \begin{pmatrix} \frac{dI}{dt} \\ \frac{dU_1}{dt} \\ \frac{dM}{dt} \\ \frac{dU_2}{dt} \\ \frac{dV}{dt} \end{pmatrix} = \begin{pmatrix} \delta XM - (\gamma_1 + \gamma_2 + \eta_1)I \\ \gamma_1 I - (\alpha_1 + \alpha_2 + \eta_1)U_1 \\ \alpha_1 U_1 - (\omega_1 + \omega_2 + \eta_1 + \eta_2)M + \gamma_2 F \\ \omega_1 M - (\eta_1 + \eta_3 + \psi)U_2 \\ \psi U_2 + \omega_2 M - \eta_1 V \end{pmatrix} \tag{A4}$$

Furthermore,

$$j_i = \begin{pmatrix} \delta XM \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, z_i = \begin{pmatrix} (\gamma_1 + \gamma_2 + \eta_1)I \\ (\alpha_1 + \alpha_2 + \eta_1)U_1 - \gamma_1 I \\ (\omega_1 + \omega_2 + \eta_1 + \eta_2)M - (\alpha_1 U_1 + \gamma_2 I) \\ (\eta_1 + \eta_3 + \psi)U_2 - \omega_1 M \\ \eta_1 V - (\psi U_2 + \omega_2 M) \end{pmatrix} \tag{A5}$$

Here, in the above equation, j_i is the rate of submission of new fake reviews for a given product, and z_i is the rate of other reviews posted on the product page of the same product. Following is the formation of the matrices J and Z:

$$J = \begin{pmatrix} \frac{\partial j_1}{\partial I} & \frac{\partial j_1}{\partial U_1} & \frac{\partial j_1}{\partial M} & \frac{\partial j_1}{\partial U_2} & \frac{\partial j_1}{\partial V} \\ \frac{\partial j_2}{\partial I} & \frac{\partial j_2}{\partial U_1} & \frac{\partial j_2}{\partial M} & \frac{\partial j_2}{\partial U_2} & \frac{\partial j_2}{\partial V} \\ \frac{\partial j_3}{\partial I} & \frac{\partial j_3}{\partial U_1} & \frac{\partial j_3}{\partial M} & \frac{\partial j_3}{\partial U_2} & \frac{\partial j_3}{\partial V} \\ \frac{\partial j_4}{\partial I} & \frac{\partial j_4}{\partial U_1} & \frac{\partial j_4}{\partial M} & \frac{\partial j_4}{\partial U_2} & \frac{\partial j_4}{\partial V} \\ \frac{\partial j_5}{\partial I} & \frac{\partial j_5}{\partial U_1} & \frac{\partial j_5}{\partial M} & \frac{\partial j_5}{\partial U_2} & \frac{\partial j_5}{\partial V} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \delta X_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{A6}$$

where, $X_0 = \frac{F}{\eta_1}$, and,

$$Z = \begin{pmatrix} \frac{\partial z_1}{\partial I} & \frac{\partial z_1}{\partial U_1} & \frac{\partial z_1}{\partial M} & \frac{\partial z_1}{\partial U_2} & \frac{\partial z_1}{\partial V} \\ \frac{\partial z_2}{\partial I} & \frac{\partial z_2}{\partial U_1} & \frac{\partial z_2}{\partial M} & \frac{\partial z_2}{\partial U_2} & \frac{\partial z_2}{\partial V} \\ \frac{\partial z_3}{\partial I} & \frac{\partial z_3}{\partial U_1} & \frac{\partial z_3}{\partial M} & \frac{\partial z_3}{\partial U_2} & \frac{\partial z_3}{\partial V} \\ \frac{\partial z_4}{\partial I} & \frac{\partial z_4}{\partial U_1} & \frac{\partial z_4}{\partial M} & \frac{\partial z_4}{\partial U_2} & \frac{\partial z_4}{\partial V} \\ \frac{\partial z_5}{\partial I} & \frac{\partial z_5}{\partial U_1} & \frac{\partial z_5}{\partial M} & \frac{\partial z_5}{\partial U_2} & \frac{\partial z_5}{\partial V} \end{pmatrix} \tag{A7}$$

$$\text{Hence, } Z = \begin{pmatrix} (\gamma_1 + \gamma_2 + \eta_1) & 0 & 0 & 0 & 0 \\ -\gamma_1 & (\alpha_1 + \alpha_2 + \eta_1) & 0 & 0 & 0 \\ -\gamma_2 & -\gamma_1 & (\omega_1 + \omega_2 + \eta_1 + \eta_2) & 0 & 0 \\ 0 & 0 & -\omega_1 & (\eta_1 + \eta_3 + \psi) & 0 \\ 0 & 0 & -\omega_2 & -\psi & \eta_1 \end{pmatrix} \tag{A8}$$

Now, by the definition of inverse of matrices, we know that $Z^{-1} = \frac{AdjZ}{|Z|}$

$$Z^{-1} = \begin{pmatrix} \frac{1}{(\gamma_1 + \gamma_2 + \eta_1)} & 0 & 0 & 0 & 0 \\ \frac{1}{(\gamma_1 + \gamma_2 + \eta_1)(\alpha_1 + \alpha_2 + \eta_1)} & \frac{1}{(\alpha_1 + \alpha_2 + \eta_1)} & 0 & 0 & 0 \\ \frac{\gamma_1 \alpha_1 + \gamma_2 (\alpha_1 + \alpha_2 + \eta_1)}{(\gamma_1 + \gamma_2 + \eta_1)(\alpha_1 + \alpha_2 + \eta_1)(\omega_1 + \omega_2 + \eta_1 + \eta_2)} & \frac{\alpha_1}{(\alpha_1 + \alpha_2 + \eta_1)(\omega_1 + \omega_2 + \eta_1 + \eta_2)} & \frac{1}{(\omega_1 + \omega_2 + \eta_1 + \eta_2)} & 0 & 0 \\ \frac{\gamma_2 \omega_1 (\alpha_1 + \alpha_2 + \eta_1) + (\omega_1 \alpha_1 \gamma_1)}{(\gamma_1 + \gamma_2 + \eta_1)(\alpha_1 + \alpha_2 + \eta_1)(\omega_1 + \omega_2 + \eta_1 + \eta_2)(\eta_1 + \eta_3 + \psi)} & \frac{\omega_1 \alpha_1}{(\alpha_1 + \alpha_2 + \eta_1)(\omega_1 + \omega_2 + \eta_1 + \eta_2)(\eta_1 + \eta_3 + \psi)} & \frac{-\omega_1}{(\omega_1 + \omega_2 + \eta_1 + \eta_2)(\eta_1 + \eta_3 + \psi)} & \frac{-\omega_2}{(\omega_1 + \omega_2 + \eta_1 + \eta_2)(\eta_1 + \eta_3 + \psi)} & 0 \\ \left(\frac{(\eta_1 + \eta_3 + \psi)\omega_2 \gamma_2 (\alpha_1 + \alpha_2 + \eta_1) + \psi \omega_1 \gamma_1 \alpha_1}{(\gamma_1 + \gamma_2 + \eta_1)(\alpha_1 + \alpha_2 + \eta_1)(\omega_1 + \omega_2 + \eta_1 + \eta_2)(\eta_1 + \eta_3 + \psi)} \right) & \frac{(\eta_1 + \eta_3 + \psi)\alpha_1 \omega_2 + \omega_1 \psi \alpha_1}{(\alpha_1 + \alpha_2 + \eta_1)(\omega_1 + \omega_2 + \eta_1 + \eta_2)(\eta_1 + \eta_3 + \psi)\eta_1} & \frac{\omega_2 (\eta_1 + \eta_3 + \psi) + \psi \omega_1}{(\omega_1 + \omega_2 + \eta_1 + \eta_2)(\eta_1 + \eta_3 + \psi)\eta_1} & \frac{\psi}{\eta_1} & \frac{1}{\eta_1} \end{pmatrix} \tag{A9}$$

Thus, the RPIM can be denoted by V_0 , given by:

$$V_0 = \kappa(JZ^{-1}) = \frac{\gamma X_0(\gamma_1\alpha_1 + \gamma_2(\alpha_1 + \alpha_2 + \eta_1))}{(\gamma_1 + \gamma_2 + \eta_1)(\alpha_1 + \alpha_2 + \eta_1)(\omega_1 + \omega_2 + \eta_1 + \eta_2)} \tag{A10}$$

$$V_0 = \kappa(JZ^{-1}) = \frac{\delta E(\gamma_1\alpha_1 + \gamma_2(\alpha_1 + \alpha_2 + \eta_1))}{\eta_1(\gamma_1 + \gamma_2 + \eta_1)(\alpha_1 + \alpha_2 + \eta_1)(\omega_1 + \omega_2 + \eta_1 + \eta_2)} \tag{A11}$$

Furthermore, we can conclude that our system of Equation (A9) becomes the following for the steady-state condition:

$$\begin{aligned} E - \delta XM - \eta_1 X + \alpha_2 U_1 &= 0 \\ \delta XM - (\gamma_1 + \gamma_2 + \eta_1)I &= 0 \\ \gamma_1 I - (\alpha_1 + \alpha_2 + \eta_1)U_1 &= 0 \\ \alpha_1 U_1 - (\omega_1 + \omega_2 + \eta_1 + \eta_2)M + \gamma_2 I &= 0 \\ \omega_1 M - (\eta_1 + \eta_3 + \psi)U_2 &= 0 \\ \psi U_2 + \omega_2 M - \eta_1 V &= 0 \end{aligned} \tag{A12}$$

Here, we have discussed the local stability of a risk-free equilibrium and the equilibrium state of the accurate recommendations for a given recommender system can be denoted by Equation (A12) after analyzing the characteristics equations.

Theorem A1. *If $V_0 < 1$, then the no-risk equilibrium state is asymptotically in the stable state in the region τ_0 for recommending a recommender system, otherwise it is in an unstable state.*

Proof. We can establish the Jacobian Matrix as $\tau_0(X = 1, I = U_1 = M = U_2 = 0)$, in a no-risk equilibrium condition.

$$M_{RFE}(\tau_0) = \begin{pmatrix} -\eta_1 & 0 & -\alpha_2 & -\delta & 0 \\ 0 & -(\gamma_1 + \gamma_2 + \eta_1) & 0 & \delta & 0 \\ 0 & \gamma_1 & -(\alpha_1 + \alpha_2 + \eta_1) & 0 & 0 \\ 0 & 0 & \alpha_1 & -(\omega_1 + \omega_2 + \eta_1 + \eta_2) & 0 \\ 0 & 0 & 0 & \omega_1 & -(\psi + \eta_1 + \eta_3) \end{pmatrix} \tag{A13}$$

This corresponding Eigenvalues of $M_{DIE}(\tau_0)$ are as follows:

$$\begin{aligned} \lambda_1 &= -\eta_1 \\ \lambda_2 &= -(\psi + \eta_1 + \eta_3) \end{aligned} \tag{A14}$$

Similarly, by using the equation of the cubic polynomial that we have found, we can calculate the other three Eigenvalues:

$$\lambda^3 + e_1\lambda^2 + e_2\lambda + e_3 = 0$$

Here,

$$\begin{aligned} e_1 &= (\gamma_1 + \gamma_2 + 3\eta_1 + \alpha_1 + \alpha_2 + \omega_1 + \omega_2 + \eta_2) \\ e_2 &= (\gamma_1 + \gamma_2 + \eta_1)(\alpha_1 + \alpha_2 + \eta_1) + (\gamma_1 + \gamma_2 + \eta_1)(\omega_1 + \omega_2 + \eta_1 + \eta_2) + (\omega_1 + \omega_2 + \eta_1 + \eta_2)(\alpha_1 + \delta + \eta_1) \\ e_3 &= (\gamma_1 + \gamma_2 + \eta_1)(\alpha_1 + \alpha_2 + \eta_1)(\omega_1 + \omega_2 + \eta_1 + \eta_2) - \delta_1\gamma_1\alpha_1 \\ &= (\gamma_1 + \gamma_2 + \eta_1)(\alpha_1 + \alpha_2 + \eta_1)(\omega_1 + \omega_2 + \eta_1 + \eta_2)(1 - V_0) \end{aligned}$$

If $V_0 < 1$, then we can deduce that $e_1 > 0; e_2 > 0, e_3 > 0$ and $e_1.e_2 > e_3$. Thus, we can say that $e_1 > 0; e_2 > 0, e > 0$ and $e_1.e_2 > e_3$. \square

Appendix A.3. Balancing the State of User Reviews and Product Reviews Using NLP Algorithms

We can obtain a balance in user reviews in collaborative filtering-based recommender systems using NLP algorithms with $\tau^*(X^*, I^*, U_1^*, M^*, U_2^*, V^*)$, by finding a solution to Equation (A12). Finally, we obtain the following equations:

$$\begin{aligned}
 V^* &= \frac{EV_0\alpha_1 - (\delta\alpha_1 + V(\omega_1 + \omega_2 + \eta_1 + \eta_2))M^*}{V_0\eta_1\alpha_1} \\
 I^* &= \frac{\delta M^*}{V_0(\gamma_1 + \gamma_2 + \eta_1)} \\
 U_1^* &= \frac{(\eta_1 + \eta_2 + \omega_1 + \omega_2)M^*}{\alpha_1 U_1^*} \\
 M^* &= \frac{\alpha_1 U_1^*}{(\omega_1 + \omega_2 + \eta_1 + \eta_2)} \\
 U_2^* &= \frac{\omega_1 M^*}{(\psi + \eta_1 + \eta_3)} \\
 V^* &= \frac{(\psi\omega_1 + \omega_2(\psi + \eta_1 + \eta_3))M^*}{\eta_1(\psi + \eta_1 + \eta_3)} \tag{A15}
 \end{aligned}$$

Local stability of balanced state for improving accuracy in recommender systems using a combined approach of user review processing and NLP-based algorithms.

Theorem A2. *The equilibrium for positive and negative recommendations in a collaborative recommender system, the natural language process approach, and algorithms with $\tau^*(X^*, I^*, U_1^*, M^*, U_2^*, V^*)$, is in an asymptotically stable state when $V_0 > 1$; otherwise, it is in an unstable state.*

Proof. At the equilibrium for positive and negative user devices using collaborative filtering and natural language processing algorithms with $\tau^*(X^*, I^*, U_1^*, M^*, U_2^*, V^*)$, the matrix of Variations Equation (A12) becomes

$$\tau_{EVP}^*(X^*, I^*, U, M^*, U_2^*, V) = \begin{pmatrix} -(\delta M^* + \eta_1) & 0 & \alpha_2 & -\delta X^* & 0 & 0 \\ \delta M^* & -(\gamma_1 + \gamma_2 + \eta_1) & 0 & \delta X^* & 0 & 0 \\ 0 & \gamma_1 & -(\alpha_1 + \alpha_2 + \eta_1) & 0 & 0 & 0 \\ 0 & 0 & \alpha_1 & -(\omega_1 + \omega_2 + \eta_1 + \eta_2) & 0 & 0 \\ 0 & 0 & 0 & \omega_1 & -(\psi + \eta_1 + \eta_2) & 0 \\ 0 & 0 & 0 & \omega_2 & \psi & -\eta_1 \end{pmatrix} \tag{A16}$$

The equivalent Eigenvalues are

$$\lambda_1 = -\eta_1$$

$$\lambda_2 = -(\psi + \eta_1 + \eta_2)$$

After finding the solutions to the fourth-degree polynomial, the other Eigenvalues are yielded as $\lambda^4 + E_1\lambda^3 + E_2\lambda^2 + E_3\lambda + E_4 = 0$, in which

$$\begin{aligned}
 E_1 &= (\delta M^* + \eta_1 + \eta_2 + \alpha_1 + \alpha_2 + \omega_1 + \omega_2 + 4\eta_1 + \eta_2) \\
 E_2 &= ((\delta M^* + \eta_1)(\gamma_1 + \gamma_2 + \eta_1) + (\delta M^* + \eta_1)(\eta_1 + \eta_2 + \omega_1 + \omega_2) + (\alpha_1 + \alpha_2 + \eta_1)(\delta M^* + \eta_1) \\
 &\quad + (\gamma_1 + \gamma_2 + \eta_1)(\omega_1 + \omega_2 + \eta_1 + \eta_2) + (\gamma_1 + \gamma_2 + \eta_1)(\alpha_1 + \alpha_2 + \eta_1) \\
 &\quad + (\alpha_1 + \alpha_2 + \eta_1)(\omega_1 + \omega_2 + \eta_1 + \eta_2)) \\
 E_3 &= ((\delta M^* + \eta_1)(\gamma_1 + \gamma_2 + \eta_1)(\omega_1 + \omega_2 + \eta_1 + \eta_2) + (\delta M^* + \eta_1)(\gamma_1 + \gamma_2 + \eta_1)(\alpha_1 + \alpha_2 + \eta_1) \\
 &\quad + (\delta M^* + \eta_1)(\omega_1 + \omega_2 + \eta_1 + \eta_2)(\alpha_1 + \alpha_2 + \eta_1) \\
 &\quad + (\omega_1 + \omega_2 + \eta_1 + \eta_2)(\gamma_1 + \gamma_2 + \eta_1)(\alpha_1 + \alpha_2 + \eta_1) - \delta M^* \gamma_1 \alpha_2 - \delta X^* \alpha_1 \gamma_1) \\
 E_4 &= ((\delta M^* + \eta_1)(\gamma_1 + \gamma_2 + \eta_1)(\omega_1 + \omega_2 + \eta_1 + \eta_2)(\alpha_1 + \alpha_2 + \eta_1) - (\delta M^* + \eta_1)\delta X^* \alpha_1 \gamma_1 + \\
 &\quad \delta^2 X^* M^* \gamma_1 \alpha_1 - \delta M^* \alpha_2 \gamma_1 (\omega_1 + \omega_2 + \eta_1 + \eta_2)) \tag{A17}
 \end{aligned}$$

Through this, we find out

$$\begin{aligned} E_1 > 0, E_3 > 0, E_4 > 0 \\ E_1 E_2 E_3 > E_3^2 + E E_4 \\ \implies E_1 E_2 E_3 - (E_3^2 + E_1^2 E_4) > 0 \end{aligned} \quad (\text{A18})$$

Thus, by applying the Routh–Hurwitz criterion for the above equations, if $V_0 > 1$, then the equation for the positive and negative recommendations in a collaborative recommender system, the NLP approach, and the proposed algorithm of Figure 3 are asymptotically stable. \square

References

- Zhang, S.; Yao, L.; Sun, A.; Tay, Y. Deep Learning-based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.* **2018**, *51*, 1–38. [CrossRef]
- Leiva, W.L.; Li, M.L.; Tsai, C.Y. A Two-Phase Deep Learning-Based Recommender System: Enhanced by a Data Quality Inspector. *Appl. Sci.* **2021**, *11*, 9667. [CrossRef]
- Wang, K.; Zhang, Y.; Yin, D.; Zheng, C.; Liu, B. RL4RS: A Real-World Benchmark for Reinforcement Learning-based Recommender System. In Proceedings of the 15th ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September–1 October 2021; pp. 444–448.
- Kiran, R.; Kumar, P.; Bhasker, B. DNNRec: A novel deep learning-based hybrid recommender system. *Expert Syst. Appl.* **2020**, *144*, 113054.
- Ge, Y.; Zhao, S.; Zhou, H.; Pei, C.; Sun, F.; Ou, W.; Zhang, Y. Understanding Echo Chambers in E-commerce Recommender Systems. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, 25–30 July 2020; pp. 2261–2270.
- Bobadilla, J.; Ortega, F.; Hernando, A.; Bernal, J. Deep Learning Architecture for Collaborative Filtering Recommender Systems. *Appl. Sci.* **2020**, *10*, 2241. [CrossRef]
- Addagarla, S.K. e-Sim Net: A Visual Similar Product Recommender System for E-Commerce. *Indones. J. Electr. Eng. Comput. Sci.* **2021**, *22*, 563–570. [CrossRef]
- He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T.S. Neural Collaborative Filtering. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 173–182.
- Venil, P.; Vinodhini, G.; Suban, R. A State of the Art Survey on Cold Start Problem in a Collaborative Filtering System. *Int. J. Sci. Technol. Res.* **2020**, *9*, 2606–2612.
- Fu, M.; Huang, L.; Rao, A.; Irissappane, A.A.; Zhang, J.; Qu, H. A Deep Reinforcement Learning Recommender System with Multiple Policies for Recommendations. *IEEE Trans. Ind. Inform.* **2023**, *19*, 2049–2061. [CrossRef]
- Chen, L.; Chen, G.; Wang, F. Recommender systems based on user reviews: The state of the art. *User Model. User-Adapt. Interact.* **2015**, *25*, 99–154. [CrossRef]
- Wei, J.; He, J.; Chen, K.; Zhou, Y.; Tang, Z. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Syst. Appl.* **2017**, *69*, 29–39. [CrossRef]
- Camacho, L.A.G.; Alves-Souza, S.N. Social network data to alleviate cold-start in recommender system: A systematic review. *Inf. Process. Manag.* **2018**, *54*, 529–544. [CrossRef]
- Lops, P.; de Gemmis, M.; Semeraro, G. Content-based Recommender Systems: State of the Art and Trends. In *Recommender Systems Handbook*; Ricci, F., Rokach, L., Shapira, B., Kantor, P., Eds.; Springer: Boston, MA, USA, 2011; pp. 73–105. [CrossRef]
- Hsieh, M.-Y.; Weng, T.-H.; Li, K.-C. A keyword-aware recommender system using implicit feedback on Hadoop. *J. Parallel Distrib. Comput.* **2018**, *116*, 63–73. [CrossRef]
- Bi, Y.; Huang, H.; Hu, Y.; Zhu, X.; Xiong, Y. A Heterogeneous Information Network based Cross Domain Insurance Recommendation System for Cold Start Users. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'20), Virtual Event, 25–30 July 2020; pp. 2211–2220.
- Son, L.H. Dealing with the new user cold-start problem in recommender systems: A comparative review. *Inf. Syst.* **2016**, *58*, 87–104. [CrossRef]
- Deepjyoti, R.; Mala, D. A systematic review and research perspective on recommender systems. *J. Big Data* **2023**, *9*, 59. [CrossRef]
- Wei, Y.; Wang, X.; Li, Q.; Nie, L.; Li, Y.; Li, X.; Chua, T.S. Contrastive Learning for Cold-Start Recommendation. In Proceedings of the 29th ACM International Conference on Multimedia, Chengdu, China, 20–24 October 2021; pp. 5382–5390.
- Jiang, L.; Cheng, Y.; Yang, L.; Li, J.; Yan, H.; Wang, X. A trust-based collaborative filtering algorithm for E-commerce recommendation system. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 3023–3034. [CrossRef]
- Zheng, G.; Zhang, F.; Yuan, N.J.; Xie, X.; Ma, W.-Y. DRN: A Deep Reinforcement Learning Framework for News Recommendation. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1671–1680.
- Li, L. Cross-Border E-Commerce Intelligent Information Recommendation System Based on Deep Learning. *Comput. Intell. Neurosci.* **2022**, *2022*, 6602471. [CrossRef]

23. Smith, B.; Linden, G. Two Decades of Recommender Systems at Amazon.com. *IEEE Internet Comput.* **2017**, *21*, 12–18. [[CrossRef](#)]
24. Liu, D.; Li, J.; Du, B.; Chang, J.; Gao, R.; Wu, Y. A Hybrid Neural Network Approach to Combine Textual Information and Rating Information for Item Recommendation. *Knowl. Inf. Syst.* **2022**, *63*, 621–646. [[CrossRef](#)]
25. Du, C.; Zhang, L.; Han, X.; Li, D. Collaborative Filtering with User-Item Co-Autoregressive Models. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI-18), New Orleans, LA, USA, 2–7 February 2018; pp. 291–298. [[CrossRef](#)]
26. Tsai, C.-Y.; Chiu, Y.-F.; Chen, Y.-J. A Two-Stage Neural Network-Based Cold Start Item Recommender. *Appl. Sci.* **2021**, *11*, 4243. [[CrossRef](#)]
27. Guo, K.; Zhang, Z.; Liu, S.; Liu, Y.; Wang, L. MD Maa S: Medical-assisted diagnosis model as a service with artificial intelligence and trust. *IEEE Trans. Ind. Inform.* **2020**, *16*, 2102–2114. [[CrossRef](#)]
28. Yi, B.; Shen, X.; Liu, H.; Zhang, Z.; Zhang, W.; Liu, S.; Xiong, N. Deep matrix factorization with implicit feedback embedding for recommendation system. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4591–4601. [[CrossRef](#)]
29. Rashid, M.; Javed, M.A.; Ahmad, J.; Lee, S. A cold-start recommender system based on clustering and deep learning. *J. Ambient Intell. Humaniz. Comput.* **2020**, *11*, 1183–1198.
30. Panteli, A.; Boutsinas, B. Addressing the Cold-Start Problem in Recommender Systems Based on Frequent Patterns. *Algorithms* **2023**, *16*, 182. [[CrossRef](#)]
31. Chen, J.; Lu, Y.; Shang, F.; Zhu, T. A novel recommendation scheme with multi-factorial weighted matrix decomposition strategies via forgetting rule. *Eng. Appl. Artif. Intell.* **2021**, *101*, 104191. [[CrossRef](#)]
32. AlRossais, N.; Kudenko, D.; Yuan, T. Improving cold start recommendations using item based stereo types. *User Model. User-Adapt. Interact.* **2021**, *31*, 867–905. [[CrossRef](#)]
33. Zhao, X.; Zhang, L.; Ding, Z.; Xia, L.; Tang, J.; Yin, D. Recommendations with Negative Feedback via Pairwise Deep Reinforcement Learning. In Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 17–20 November 2018; pp. 1040–1048.
34. Zhao, X.; Xia, L.; Zhang, L.; Ding, Z.; Yin, D.; Tang, J. Deep Reinforcement Learning for Page-wise Recommendations. In Proceedings of the 12th ACM Conference on Recommender Systems, Vancouver, BC, Canada, 2–6 October 2018; pp. 95–103.
35. Zou, L.; Xia, L.; Du, P.; Zhang, Z.; Bai, T.; Liu, W.; Nie, J.; Yin, D. Pseudo Dyna-Q: A Rein for cement Learning Framework for Interactive Recommendation. In Proceedings of the 13th International Conference on Web Search Data Mining, Houston, TX, USA, 3–7 February 2020; pp. 816–824.
36. Lei, Y.; Li, W. Interactive recommendation with user-specific deep reinforcement learning. *ACM Trans. Knowl. Discov. Data* **2019**, *13*, 1–15. [[CrossRef](#)]
37. Singh, P.K.; Pramanik, P.K.D.; Dey, A.K.; Choudhury, P. Recommender systems: An overview, research trends, and future directions. *Int. J. Bus. Syst. Res.* **2021**, *15*, 14–52. [[CrossRef](#)]
38. Rahman, M.; Shama, I.A.; Rahman, S.; Nabil, R. Hybrid Recommendation System to Solve Cold start Problem. *J. Theor. Appl. Inf. Technol.* **2022**, *100*, 3562–3578.
39. Abdullah, N.A.; Rasheed, R.A.; Nasir, M.H.N.M.; Rahman, M.M. Eliciting Auxiliary Information for Cold Start User Recommendation: A Survey. *Appl. Sci.* **2021**, *11*, 9608. [[CrossRef](#)]
40. Davidson, J.; Liebald, B.; Liu, J.; Nandy, P.; Van Vleet, T.; Gargi, U.; Gupta, S.; He, Y.; Lambert, M.; Livingston, B.; et al. *The You Tube Recommender System*; IAEME Publication: Chennai, India, 2019.
41. Reddy, S.R.S.; Nalluri, S.; Kuniseti, S.; Ashok, S.; Venkatesh, B. Content-Based Movie Recommendation System Using Genre Correlation. In *Smart Intelligent Computing and Applications*; Satapathy, S.C., Bhateja, V., Das, S., Eds.; Springer: Singapore, 2019; pp. 391–397.
42. Tabianan, K.; Velu, S.; Ravi, V. K-Means Clustering Approach for Intelligent Customer Segmentation Using Customer Purchase Behavior Data. *Sustainability* **2022**, *14*, 7243. [[CrossRef](#)]
43. Kang, S.; Chung, K. Preference-Tree-Based Real-Time Recommendation System. *Entropy* **2022**, *24*, 503. [[CrossRef](#)]
44. Karimova, F. A Survey of e-Commerce Recommender Systems. *Eur. Sci. J. ESJ* **2016**, *12*, 75. [[CrossRef](#)]
45. Tawiq, F.; Munir, S.; Akhtar, N.; Ali, A.; Aslam, W. An E-Commerce Recommendation System Based on Dynamic Analysis of Customer Behaviour. *Sustainability* **2021**, *13*, 11818. [[CrossRef](#)]
46. Hussien, F.T.A.; Rahma, A.M.S.; Wahab, H.B.A. Recommendation Systems For E-commerce Systems An Overview. In Proceedings of the Sixth International Scientific Conference for Iraqi Al Khwarizmi Society (FISCAS) 2021, Cairo, Egypt, 22–23 November 2021.
47. Addagarla, S.K.; Amalanathan, A. Probabilistic Unsupervised Machine Learning Approach for a Similar Image Recommender System for E-Commerce. *Symmetry* **2020**, *12*, 1783. [[CrossRef](#)]
48. Pandey, S.; Khanna, P. Content-Based Image Retrieval Embedded with Agglomerative Clustering Built on Information Loss. *Comput. Electr. Eng.* **2016**, *54*, 506–521. [[CrossRef](#)]
49. Okon, E.U.; Eke, B.O.; Asagba, P.O. An Improved Online Book Recommender System using Collaborative Filtering Algorithm. *IJCA* **2018**, *179*, 41–48. [[CrossRef](#)]
50. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Item-based collaborative filtering recommendation algorithms Web Recommendation System for E-Commerce Applications. In Proceedings of the 10th International Conference on World Wide Web, Lyon, France, 23–27 April 2018.
51. Sharma, S.N.; Sadagopan, P. Influence of conditional holo entropy- based feature selection on automatic recommendation system in E-commerce sector. *J. King Saud Univ.—Comput. Inf. Sci.* **2022**, *34*, 5564–5577.

52. Zhang, Z.; Xu, G.; Zhang, P. Research on E-Commerce Platform-Based Personalized Recommendation Algorithm. *Appl. Comput. Intell. Soft Comput.* **2016**, *2016*, 5160460. [[CrossRef](#)]
53. Burke, R. Integrating Knowledge-based and Collaborative-filtering Recommender Systems. In Proceedings of the AAAI Conference on Artificial Intelligence, Orlando, FL, USA, 18–22 July 1999.
54. Sharma, R.S.; Saikh, A.A.; Li, E. Designing Recommendation or Suggestion Systems: Looking to the Future. *Electron. Mark.* **2021**, *31*, 243–252. [[CrossRef](#)]
55. Hu, Y.; Yi, X.; Davis, L.S. Collaborative fashion recommendation: A functional tensor factorization approach. In Proceedings of the 23rd ACM International Conference on Multimedia, ACM, Brisbane, Australia, 26–30 October 2015.
56. Abdulla, G.M.; Arif, S.; Jassim, F. Size recommendation system for fashion e-commerce. *J. Soft Comput. Decis. Support Syst.* **2017**, *4*, 12–22.
57. E, R.K.; Venkatesan, R.; Suresh Kumar, V. E-Commerce Recommender System Using Product Data. *Int. J. Sci. Technol. Res.* **2020**, *9*, 5811–5816.
58. Lee, D.; Hosanagar, K. Impact of Recommender Systems on Sales Volume and Diversity. *MIS Q.* **2014**, *38*, 143–166.
59. Khalid, A.; Lundqvist, K.; Yates, A. Recommender Systems for MOOCs: A Systematic Literature Survey (January 1, 2012–July 12, 2019). *Int. Rev. Res. Open Distrib. Learn.* **2020**, *21*, 255–291. [[CrossRef](#)]
60. Salunke, J.; Chaudhari, A. Classification of Recommendation System for E-commerce Application. *J. Comput. Sci. Eng. Softw. Test.* **2017**, *3*, 1–10.
61. Paniello, U. Developing a price-sensitive recommender system to improve accuracy and business performance of e-commerce applications. *Int. J. Electron. Commer. Stud.* **2015**, *6*, 57–66.
62. Aamir, M.; Bhusry, M. Recommendation system: State of the art approach. *Int. J. Comput. Appl.* **2015**, *120*, 25–32. [[CrossRef](#)]
63. Alamdari, P.M.; Navimipour, N.J.; Hosseinzadeh, M.; Safaei, A.A.; Darwesh, A. A systematic study on the recommender systems in the e-commerce. *IEEE Access* **2020**, *8*, 115694–115716. [[CrossRef](#)]
64. Houshmand-Nanehkaran, F.; Lajevardi, S.M.; Mahlouji-Bidgholi, M. Optimization of Fuzzy Similarity by Genetic Algorithm in User-Based Collaborative Filtering Recommender Systems. *Expert Syst.* **2022**, *40*, e12893. [[CrossRef](#)]
65. Hazrati, N.; Elah, M. Addressing the New Item Problem in Video Recommender Systems by Incorporation of Visual Features with Restricted Boltzmann Machines. *Expert Syst.* **2021**, *38*, e12645. [[CrossRef](#)]
66. Santos, R.D.; Aguilar, J. A Hybrid Recommender System Based on Description/Dialetheic Logic and Linked Data. *Expert Syst.* **2022**, *40*, e13143. [[CrossRef](#)]
67. Sohail, S. Recommender Systems for Ecommerce: In perspective of Business Strategies. *Int. J. Latest Technol. Eng. Manag. Appl. Sci.* **2013**, *8*, 165–169.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.