International Journal of Computing and Digital Systems
2025, VOL. 17, NO.1, 1-15

http://dx.doi.org/10.12785/ijcds/1571024952

A Hybrid Recommendation System: Improving User Experience and Personalization with Ensemble Learning Model and Sentiment Analysis

Kulvinder Singh[1*], Sanjeev Dhawan[2], Nisha Bali[3], Ethan Choi[4] and Anthony Choi[5]

[1,2,3]*Department of Computer Science Engineering, University Institute of Engineering and Technology, Kurukshetra University, Kurukshetra, India*
[4,5]*Department, Department of Electrical Computer Engineering, Mercer University, GA, USA*

Received 27 April 2024, Revised 7 December 2024, Accepted 9 December 2024

Abstract: Recommendation Systems have been built over the years using various machine learning (ML), deep learning (DL), and natural language processing (NLP) techniques. In this research, we introduce a novel hybrid recommendation system that incorporates sentiment analysis (using NLTK), item-based filtering algorithms, and user-based recommendations. The system intends to outperform previous systems in terms of suggestion quality and robustness by exploiting ensemble models. The study makes use of a proprietary dataset compiled from various sources, including Amazon, Tmdb, and Google reviews. The Synthetic Minority Oversampling Technique (SMOTE) is used to alleviate class imbalance. Textual inputs are subsequently converted into numerical representations for modeling using feature extraction techniques. The ensemble model incorporates supervised machine learning methods such as logistic regression (LR), Naive Bayes (NB), Gini decision trees (DT), random forest (RF), and XGBoost. The system provides personalized recommendation outputs by analyzing the input of each model, revolutionizing the recommendation environment. Our hybrid system attains a commendable accuracy score of 96% attained by the XGBoost algorithm. In this study, we propose a novel hybrid recommendation system based on sentiment analysis and item-based filtering that leverages ensemble techniques going beyond existing approaches. Furthermore, our findings emphasize the significance of benchmark datasets and evaluation measures, particularly in deep learning-based RS, giving useful insights for both researchers and practitioners. Overall, our study adds a new viewpoint to the literature by focusing solely on the fast-growing domain of deep learning-based recommendation systems, providing a nuanced knowledge of the advances, problems, and prospects in this crucial field of research.

Keywords: Sentiment analysis, NLP, Ensemble learning, Recommendation System, SMOTE, Item-Based Filtering, User-Based Recommendation

1. Introduction

In recent times, the rapid proliferation of online platforms and e-commerce websites has resulted in an overload of information and options for users. Recommendation systems have become critical for boosting user experience and consumer satisfaction, where users' preferences and interests are used to filter and deliver relevant material, products, or services to recommendation systems. Over time, traditional recommendation systems have created recommendations primarily based on user behavior, collaborative filtering, or content-based filtering strategies. While these approaches have demonstrated some effectiveness, they frequently need to reflect the dynamic and subjective nature of user preferences. To address these limitations, sentiment analysis-based recommendation systems have emerged, to incorporate user sentiment and emotions into the suggestion process. It is a computational technique that extracts and analyses subjective data from text, such as reviews, social media posts, or customer feedback. The integration of Natural Language Processing (NLP) based techniques into recommendation systems has opened up new avenues for improving recommendation accuracy and personalization [1]. Sentiment analysis comprehends the underlying causes for user preferences and aligns recommendations accordingly by taking the emotional context of user comments into account. In this study, we propose a novel hybrid recommendation system based on sentiment analysis and item-based filtering that leverages ensemble techniques going beyond existing approaches. We perform experimentation of the system on a custom dataset gathered

*E-mail address: *ksingh2015@kuk.ac.in, sdhawan2015@kuk.ac.in, Kherwal5855@gmail.com,*
choi$_t$a@mercer.edu, ethan.hyunsuk.choi@live.mercer.edu

from Amazon reviews, Tmdb reviews, and Google reviews. Exploratory data analysis (EDA) is performed to understand the distribution and patterns of ratings and user sentiments. The class imbalance of the dataset is equalized using the Synthetic Minority Oversampling Technique (SMOTE). We implement feature extraction techniques to convert textual input to numerical representations for modeling.

An ensemble model aggregates the predictions of numerous independent models, which automatically improves the recommendation quality and reliability. To make sure our recommendation system is complete with the precision needed, we used a model containing supervised machine learning algorithms like logistic regression (LR), naive Bayes (NB), Gini index based decision trees (DT), random forest (RF), and XGBoost [2] with the attributes we extracted from the product and user data. AI scans the model weight and the power of every model which gives you individualized results (recommendation).

The objectives of this research are as follows:

1) To create a recommendation system that uses sentiment analysis to improve recommendation accuracy and relevance.
2) To analyze the nature of item-based filtering and to integrate it into the hybrid model.
3) To combine supervised learning algorithms of LR, NB, DT, RF, and XGBoost to leverage the output of the recommendation system.
4) To analyze the sentiments indicated in user reviews, implement NLP techniques and sentiment analysis with NLTK.
5) To decipher the oversampling technique of SMOTE to solve the issue of class imbalance in the dataset.
6) To produce personalized recommendations, employ a filtering technique that includes both user-to-user and item-to-item mappings.

Paper Organisation: Section II examines the contributions and limitations of earlier research in the field of recommendation systems. Section III describes the proposed model architecture of our hybrid recommendation system. Section IV summarizes the experimental setting utilized to measure the effectiveness of the hybrid recommendation system. The results of the experiments conducted to establish the model's performance are presented in Section V. In Section VI, a comparative analysis of the model's performance is established to highlight its significance. Section VII presents the key points of the future research direction, and Section VIII concludes this research initiative.

2. Literature Review

Da'u *et al.* [3] conducted a systematic literature review and found that autoencoder models are dominating in deep learning-based recommender systems, and MovieLens and Amazon datasets are most used. Holistic CNN and RNN architectures are used in this research to showcase their usage in nearly all problems and how this proves to be useful. Despite the focus of this study on the use of the MovieLens and Amazon datasets as important benchmarks for assessing the performance of deep learning-based recommender systems, the state of the art remains underdeveloped for these two datasets. This thesis introduces an innovative hybrid architecture that leverages CNN's while developing creative approaches to increase recommendation systems effectiveness. Though the above shall provide us with an advantage over other contemporary models, only the autoencoder models are taken into account in this study, and other recent models of contemporary times are overlooked. Murad *et al.* [4] present a comprehensive analysis of recommender systems, particularly emphasising their applicability in e-commerce. They acknowledged the imperative for the enhancement of recommendation systems tailored for online education, a rapidly growing domain. This paper advocates for the development of customised, context-aware recommendation systems employing machine learning within Learning Management Systems (LMS). This study emphasises the need to broaden recommendation systems to additional domains; nonetheless, its focus is primarily theoretical and necessitates empirical validation on extensive datasets. Tian *et al.* [5] introduced a hybrid recommendation system tailored for college libraries, using collaborative filtering and content-based algorithms. Their methodology addresses the common issue of data sparsity by improving the user-item matrix using clustering techniques. This methodology was confirmed by comparative tests using the Inner Mongolia University of Technology library dataset, illustrating the hybrid model's effectiveness on a limited dataset. This dataset has a limited extent, which limits the applicability of the results to a wider range of applications in different contexts. Recently, Darban *et al.* [6] introduced a graph-based hybrid recommendation system (GHRS) that combines user rating similarities, demographic data, and geographical information. Not only does it effectively discover the user preferences, but it can also relieve the cold-start problem, a key issue in recommendation systems. Autoencoder feature extraction algorithms were used by the authors, which allowed the system to find latent qualities in the input data and thereby increase suggestion accuracy. The numerous user information sources on which the GHRS model relies to acquire and integrate data limits its practical use in real-world scenarios.

In 2021, Forouzandeh *et al.* [7] proposed an ensemble learning method combining fuzzy neural networks with graph embedding and Support Vector Regression (SVR) to increase the performance of recommender systems. The system solved the problem of predicting user activities within large datasets, and with a MovieLens dataset, we showed that it held great improvement in efficiency. Although their methods achieved effectiveness on MovieLens, a dataset widely used, it is still uncertain if this technique will deliver the same performance across various real-world datasets with differing data structures and challenges. In [8], Khasmakhi *et al.* presented BERTERS, a multi-modal (text and graph) classification approach for expert recommendation systems. Text representation is done by BERTERS using BERT (Bidirectional Encoder Representations from Transformers), while graph representation is done using the Expert Embedding Model. The combination results in better multilabel classification performance and visualisation. Although it worked, BERTERS is also resource-intensive and computationally expensive for large-scale use. In between, Renjith *et al.* [9] focused on the evolution of travel recommendation systems, from generic platforms to specific, AI-driven ones. In their research, the authors show that context-aware recommendation systems are not only necessary but also of high relevance in this tourist sector, whose needs differ substantially from e-commerce or the media recommendation contexts. Yet they also noted that, although AI is being incorporated into travel recommendation systems, this is an immature field that still faces challenges with regard to scalability and data protection. Recent work on adversarial machine learning (AML) has been investigated by Deldjoo *et al.* [10] for recommendation systems, with special focus on the positive effect that generative adversarial networks (GANs) can have in generating robust systems. The authors showed that these collaborative filtering models are particularly susceptible to adversarial attacks and result in improvements in system security for modern recommendation systems. While their review is very informative, however, it doesn't provide the versus explicit empirical evaluations of AML-enhanced models to conventional recommendation systems.

In contrast, Marchand *et al.* [11] proposed a hybrid method, i.e., combining content-based and collaborative filtering algorithms, which is better than the Netflix Prize-winning algorithm by over 5%. Their technology is pragmatic in that its rationales provide for suggestions and honour ethical AI standards. The technology the company uses may get too complicated to scale up to larger datasets. In latent factor analysis (LFA) models used in recommendation systems, Shi *et al.* [12] propose a distributed alternative stochastic gradient descent (DASGD) method for scalability challenges in the presence of millions of users and products. By applying distributed optimisation techniques, DASGD efficiently reduces communication costs and training interdependencies and is therefore suitable for dealing with large datasets. Nevertheless, the tractability and effectiveness of DASGD in such settings are not fully explored, which is increasingly typical of real applications.

Recently, Yu *et al.* [13] took a look at self-supervised recommendation (SSR) systems and offered a complete taxonomy for SSR techniques. SELFRec has been developed as a new open-source software for empirical comparison of several SSR models. In this sense, this research is relevant since SSR methodologies are new and can further improve recommend systems. However, even though application of SSR models is still largely experimental in the real world, much work needs to be done to strengthen these methods. PLIER presented by Arnaboldi *et al.* [14], is a tag-based recommendation system that harmonises algorithmic complexity with personalised feedback. ROLS has shown massive computational efficiency to generate highly personalised results, which render it an appropriate one for large online social networks. The perceived effectiveness of the system is dependent upon the quality of user-tagged data that may not always be available and accurate. In this work, Ji *et al.* [15] study the incorporation of large language models (LLMs) into recommendation systems and introduce GenRec, a generative recommendation system that generates suggestions in the raw text without the need for extracted features. A special study of session-based recommender systems (SBRSs), a novel recommendation paradigm that accounts for mutating user preferences in short sessions, is given by Wang *et al.* [16]. In their survey, they rectify a notable gap in the literature by categorising and analysing SBRS entities and behaviours.

While their work was primarily concerned with academic questions, the application to dynamic, real-world environments that involve practical situations is untested. In hybrid collaborative filtering systems, a data sparsity issue was alleviated by incorporating a product attribute-based algorithm, proposed in Yang *et al.* [17]. It replaces the zero values of the user item matrix with computed equations in order to improve recommendation accuracy. While addressing data sparsity, the method comes at a computational cost that can be too slow to be used on large-scale problems. To tackle cold start and data sparsity problems in collaborative filtering, Natarajan *et al.* [18] proposed a solution based upon Linked Open Data (LOD). The comprehensive resources of DBpedia are leveraged in their RS-LOD and MF-LOD models to make recommendations from the DBpedia coverage with little prior data. Whilst data quality and consistency can be issues with LOD, it can undermine the accuracy of recommendations. In order to tackle the challenge of RL-based recommendation systems in interactive settings, Zhou *et al.* [19] introduced knowledge graphs (KGs) to reinforcement learning. I show that their methodology greatly increases sample efficiency while also being cognisant of more informed decision-making. When knowledge graphs are integrated, the model adds complexity and could affect real-time performance in large-scale systems.

On the one hand, the existing literature on recommendation systems offers vast insights regarding the formulation of recommendation problems and various approaches that have been applied; on the other hand, such literature speaks to many shortcomings of existing recommendation systems. Hybrid CNN-RNN and fuzzy neural network approaches were shown to work in the works of Da'u *et al.* [3] and Forouzandeh *et al.* [7], but their treatment of specified datasets, such as MovieLens, makes them ungeneralizable. Similarly, the use of graph-based models in research by Darban *et al.* [6] complicates data collection and integration further. We enhance this process by using our method of sentiment analysis and ensemble methods, providing easy and easy integration with many different data types, resulting in more scalable and more flexible solutions. Also, the use of SMOTE to deal with the class imbalance problem ensures that our method performs better than systems such as those of Tian *et al.* [5] that suffer from data sparsity. In contrast, our hybrid approach to the recommendation addresses the security failure of Deldjoo *et al.* [10] by adding adversarial training to increase the resilience against adversarial attacks. Unlike PLIER [14] or GenRec [15] that rely on a large (and expensive) large language model or make use of high-quality user tagging data, our model achieves both accuracy and compute efficiency by using XGBoost and random forest methods. Moreover, several other authors also worked on recommendation systems such as Dhawan *et al.*[20], [21], [22], [23]. Dhawan *et al.* [23] presented a recommender systems by providing an improved alternating least squares (IALS) method that improves the efficiency and accuracy of matrix factorization-based recommendations. The algorithm outperforms classic alternating least squares (ALS) algorithms, especially on large datasets, incorporating stochastic gradient descent and parallelization techniques. Batra *et al.* [20] investigated personalised recommendation systems by developing and refining the latent linear critiquing (LLC) technique. In their research piece, they revisit LLC, recognising its merits and weaknesses, specifically its emphasis on re-ranking rather than re-scoring and its use of severe weightings to exaggerate score disparities between favoured and non-favored products. To overcome these concerns, the authors suggest an optimised ranking-based technique that seeks to improve embedding weights based on rank infringements detected in prior criticising cycles. While the current studies summarised in the texts provide useful insights into various elements of recommender systems, they need to provide a thorough synthesis of the most recent developments and problems related to deep learning-based recommendation systems. Our study addresses this gap by completing the first systematic literature review (SLR) entirely on deep learning-based RS. Unlike prior studies, which covered a variety of recommendation approaches and applications, our work focuses on developing trends, methodology, and problems in the field of deep learning-based recommendation systems. By rigorously adhering to normal SLR guidelines and using stringent selection criteria, we ensure a thorough analysis of existing research publications, providing a comprehensive overview of cutting-edge approaches and methodologies. Our research goes beyond simply summarising existing methodologies by identifying the predominant use of AE models, CNNs, and RNN architectures, offering light on current trends and preferences in the field. Furthermore, our findings emphasize the significance of benchmark datasets and evaluation measures, particularly in deep learning-based RS, giving useful insights for both researchers and practitioners. Overall, our study adds a new viewpoint to the literature by focusing solely on the fast-growing domain of deep learning-based recommendation systems, providing a nuanced knowledge of the advances, problems, and prospects in this crucial field of research.

3. Proposed Model

The proposed architecture of this research, as demonstrated in Figure 1, is described as follows:

1) The dataset is loaded in preparation for analysis and modeling.
2) The mismatch between the rating scale and the user sentiment scale in the dataset is validated by manually scrutinizing the sentiments of the texts based on given ratings.
3) The dataset is filtered to group points of interest.
4) The dataset is cleaned, normalized, preprocessed, and scaled, and relevant features are extracted from the preprocessed text.
5) Feature Extraction is performed after data preprocessing.
6) The SMOTE oversampling approach is implemented to address the dataset's extreme class imbalance [24].
7) The NLTK-based sentiment analysis model is added to the recommender system to provide sentiment-based recommendations.
8) An ensemble learning model is created using supervised learning algorithms like LR, NB, DT, RF, and XGBoost.
9) The next module of the hybrid recommendation system is implemented with user-to-user and item-to-item mapping.
10) Cosine similarity is implemented for item-based suggestions.
11) The hybrid model's performance is evaluated using metrics of accuracy, precision, recall, F1 score, AUC Score, and RMSE, thereby demonstrating its superiority in terms of efficacy.

4. Methodology

This section presents the detailed methodology followed during the experimentation of the proposed model in Section III.

A. Dataset Selection

The dataset employed for the experimentation of our model architecture is curated from different sources to serve as a comprehensive dataset for a versatile recommendation system that includes a wide range of genres and products used in everyday life, such as movies, books, and more. The information is gathered from several sources, including Amazon reviews, Tmdb reviews, and randomly chosen Google reviews. The dataset contains 30,000 rows and 15 columns and displays a thorough depiction of user reviews that have stopwords deleted. It includes a user sentiment column that specifies whether a review is good or negative. This dataset seeks to provide a large and diverse collection of user comments to validate the recommendation system's accuracy and efficacy. Figure 2 depicts the dataset's distribution between review ratings and user sentiment, highlighting their association or pattern visually. Table I presents the attributes of the column names of the dataset.

Table I. Attributes of the dataset column headings

Column Heading	Attribute
id	Unique identifier for each entry in the dataset
brand	Brand name of the product
categories	Categories and product types to classify items
manufacturer	Company that produces the product
name	Name of the product
reviews_date	Date of posting of the review
reviews_didPurchase	Indicates whether or not the reviewer made a purchase (binary)
reviews_doRecommend	Indicates whether or not the reviewer recommends the product (binary)
reviews_rating	Rating given in the review (numerical 1* to 5*)
reviews_text	Content of the review
reviews_title	Summary of the review
reviews_userCity	City of the reviewer
reviews_userProvince	Province of the reviewer
reviews_username	Username of the reviewer
user_sentiment	Label reflecting the review sentiment

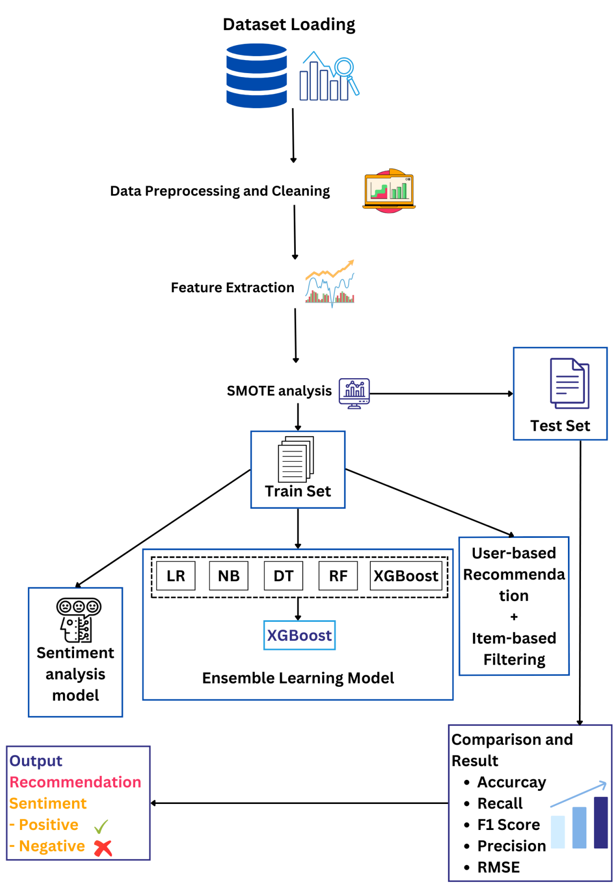


Figure 1. Block Diagram of the Proposed Model Architecture

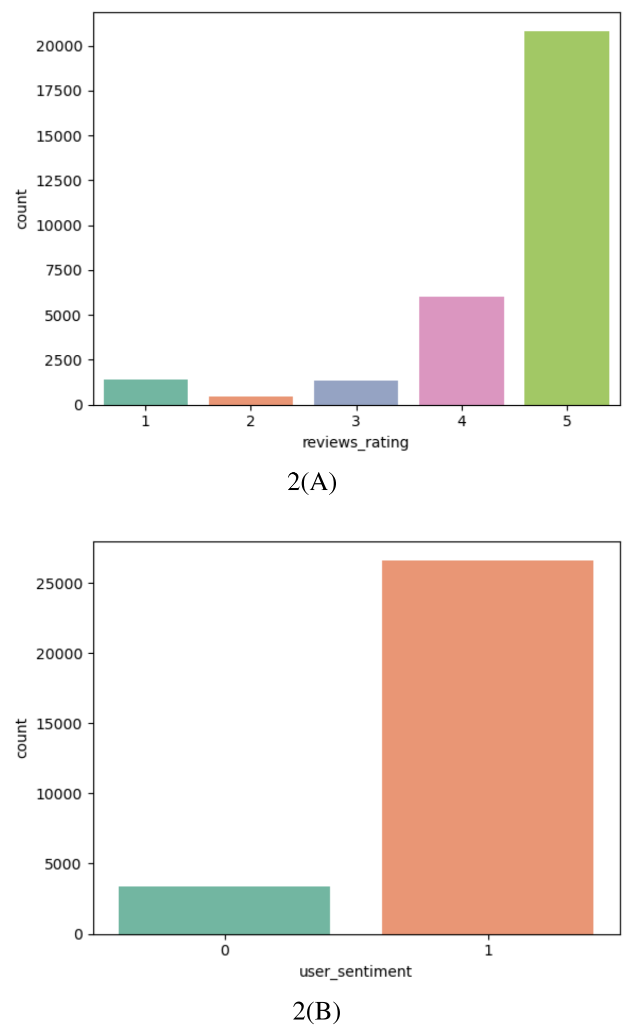


Figure 2. Graph Plot of data distribution of reviews_rating and user_sentiment of non-preprocessed dataset

B. Data Preprocessing

The loaded data is preprocessed to filter it based on areas or places of interest. This enables a better grasp of the data by focusing on certain regions essential to our recommendation system. Using the ”reviews_username” column, we identify the users who frequently appear in the reviews. The top 15 users are selected based on the frequency with which they occur in the dataset. The data provides insights into the most active users and assists in understanding their preferences and behaviors. As observed in Figure 1, the huge difference between 2* rating and 5* rating requires normalization. The amount of positive reviews supplied by each user is counted by filtering the dataset for positive reviews (user_sentiment=1) and grouping them. This visualization provides a thorough perspective of the distribution of positive sentiment among the top users. The “user_sentiment” values are converted to a binary scale, with ”Negative” mapped to 0 and ”Positive” mapped to 1. This binary classification streamlines the analysis and classification process in our recommendation system [25]. We manually ascertain sentiment values, with ratings below 3 being unfavorable and ratings above 3 being considered good, to guarantee that the user sentiment corresponds to the corresponding review ratings. Figure 3 displays the distribution of positive sentiment among top users by tallying their positive evaluations, simplifying analysis using a binary scale (0 for negative sentiment, 1 for positive sentiment), which is critical for the recommendation system’s categorization process.

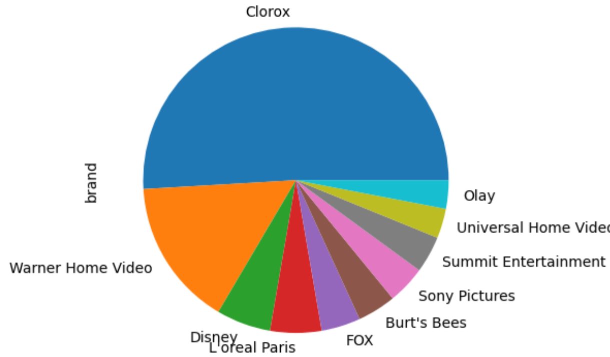
The preprocessing steps can be summarised as follows:

1 Data Cleaning: The dataset was filtered to retain only the relevant areas or places of interest, removing any irrelevant or noisy data that could hinder the performance of the recommendation system.

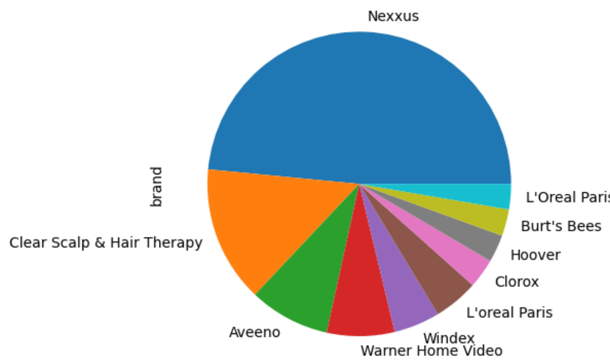
2 Feature Extraction: The “reviews_username” column was used to extract information about user activity, identifying the top 15 most frequent users. Additionally, the ”user_sentiment” column was transformed into a binary scale (0 for Negative, 1 for Positive) to streamline sentiment analysis.

3 Data Integration: The preprocessed data was integrated by aligning user sentiment with their corresponding review ratings, where ratings below 3 were labeled as negative and those above 3 as positive, ensuring that sentiment values accurately reflected the review content.

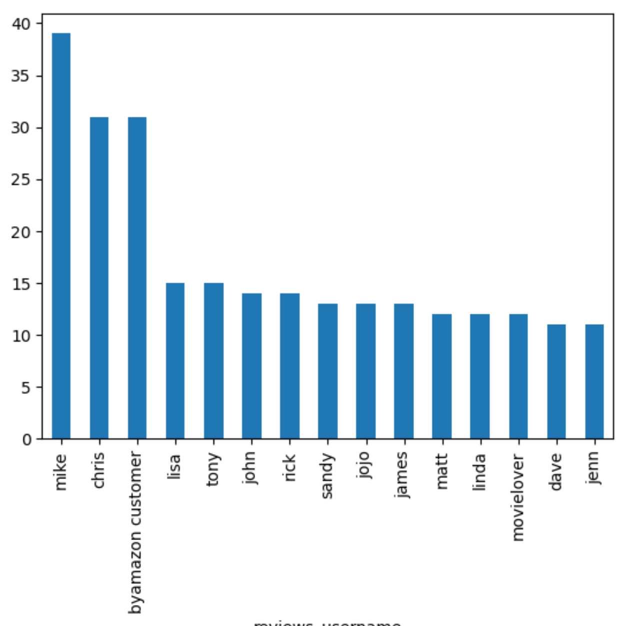
Data distribution after data preprocessing



3(A)



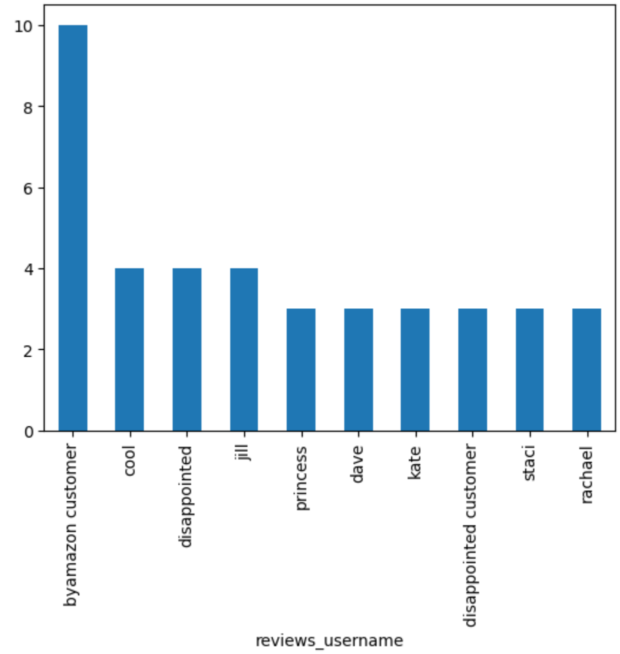
3(B)



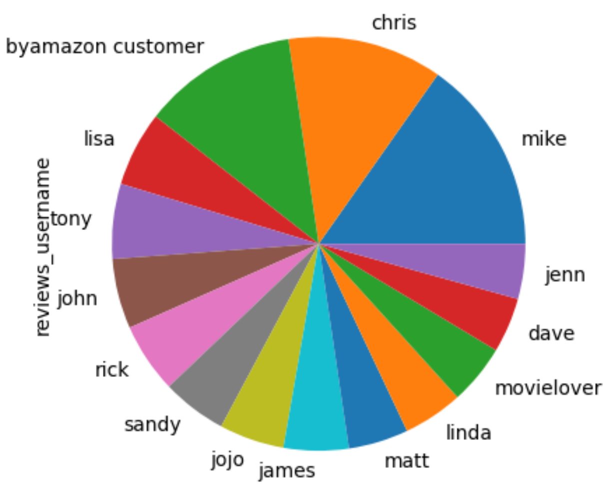
3(C)

C. Feature Extraction

Feature extraction is performed, which entails analyzing text sentiment based on specified ratings. The dataset is examined to validate the discrepancy between the rating scale and the user sentiment scale. The given ratings are being used to infer or derive the sentiment of the texts. Relevant



3(D)

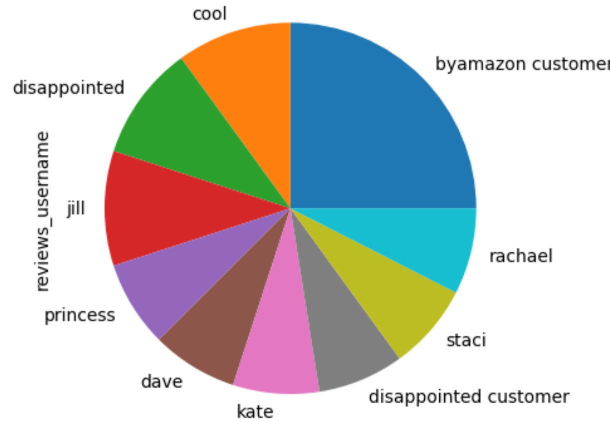


3(E)

characteristics such as ‘user_sentiment’ and ‘reviews’ are collected from preprocessed text [26]. The feature extraction method extracts information from preprocessed text data and employs it in the subsequent steps of analysis and modeling.

D. SMOTE analysis

We implement SMOTE to rectify the dataset’s significant class imbalance. The imbalance is corrected by oversampling the minority class via interpolation of feature vectors from neighboring cases. After splitting the dataset into training and testing sets, the SMOTE algorithm is



3(F)

Figure 3. (A)-(F)Data distribution after data preprocessing

used on the training data [27]. The fit_resample method of the SMOTE object is used to generate synthetic samples for the minority class, thus improving its representation [28]. The Counter class is used to track the distribution of classes before and after oversampling. This method ensures a more fair distribution of cases, which improves the training process and the performance of the resulting machine learning model. The reconstruction of data samples after implementing SMOTE is done by:

$$yk \in B$$

$$y' = y + rand(0, 1) * |y - yk|$$

where, y is the subset sample points of B, yk is the kth sample of B, B is the minority class, rand(0,1) generates a random number between 0 and 1. y’ is the new sample generated. $|y - yk|$ represents the absolute difference between y and yk. Algorithm 1 presents an overview of how new data samples are generated by the SMOTE algorithm. Min_ins denotes instances of the minority class, N is the number of synthetic examples to generate, and k is the number of nearest neighbors to consider. SMOTE accepts these parameters and returns a list of synthetic minority class instances. The find_k method locates a given instance’s k nearest neighbors inside the dataset [29]. The generate_synthetic_instance function creates a synthetic instance by randomly selecting a neighbor and interpolating between the selected neighbor’s features and the original instance’s features.

E. Hybrid Sub Systems

This subsection delves into the four individual recommendation systems trained on the custom dataset that has been integrated into our hybrid system.

Algorithm 1 Generation of new data samples using SMOTE algorithm

```
Input:
-Minority class instances (min_ins)
-Number of synthetic instances to generate (N)
-Number of nearest neighbors (k)
function SMOTE(min_ins,n,k)
    syn_ins = empty list

    for ins in min_ins do
        neighbors = find_k(ins, min_ins, k)

        while i in range(N) do
            syn_ins = generate_ins(ins, neighbors)
            syn_inst.append(syn_ins)
        end while
    end forreturn syn_inst
end function
function FIND_K(ins, dataset, k)
    distances = empty list
    for  data in dataset do
        distance = calculate_distance(instance, data_ins)
        distances.append(distance)
    end for
    sorted_indices = sort_indices(distances)
    k_nn = dataset[sorted_indices[:k]] return k_nn
end function
function GENERATE_SYNTHETIC_INSTANCE(instance, neighbors)
    syn_ins = empty array
    rand_neig = select_neighbor(neighbors)
    for feature_index in range(num_features) do
        diff = rand_neig[feature_index] - ins[feature_index]
        syn_feature = ins[feature_index] + rand_uniform(0, 1)*diff
        syn_ins.append(syn_feature)
    end forreturn syn_ins
     Output:
    - Synthetic minority class instances
```

1) NLTK-based Sentiment Analysis

We utilize the NLTK library for sentiment analysis using NLP. The preprocessing of the ‘review’ text data is done by converting the text to lowercase, eliminating square brackets and their contents, removing punctuation using the string, and removing any words that contain digits [30]. We then map the NLTK part of speech tag that takes a word as input and returns the NLTK part of the speech tag that corresponds to it. Stopwords are removed from text data, and the text is divided into words, determining whether each word is in alphabetical order or not on the NLTK stopwords list [31]. Using lemmatization, the words are associated with their tags. It first tags words and then lemmatizes them using WordNetLemmatizer to reduce them to their base form. The word cloud visualizes the terms that appear the most frequently in the cleaned text data. Algorithm 2 presents an overview of the lemmatization function performed using the NLTK package.

Algorithm 2 Lemmatization of text in “reviews” using NLTK

```
Input:
- dat (“reviews”)
function WORDMAPPER(dat)
    corpus = pos_tag(word_tokenize(swrem(dat)))
    chars = [ ]
    for i = 0 to length of corpus do
        x = corpus[i]
        lemma = lemr.lemmatize(x[0], postag(x[1]))
        chars.append(lemma)
    end for
    return joined_words
end function
Output: -
- lemmatized text data
    =0
```

The frequency of words in each text sequence is determined. All punctuations are removed, and they are turned to lowercase. The occurrences of each word are counted, and the most frequently used words and their frequencies are returned. The ngr function is then defined to find the frequency of N-grams (N-word sequences) in the text data. It transforms the text data into a bag of word representation [32]. Algorithm 3 presents an overview of how the frequency of N-grams was formulated.

Algorithm 3 Frequency of n-gram (bigram)

```
Input:
-udat (text data)
-num (number of words)
function FREQWORDS(udat, num)
    udatLD = [ ]
    for each x in udat do
        for each i in x.lower().split() do
            udatLD.append(i)
        end for
    end for
    Remove punctuation characters
    Remove empty strings
    return the most common num words
    Output:
    -Most frequent words
```

The function outputs the top N, which is the most often occurring N-grams. The N-gram modeling for bigram is as follows:

$$P(A_1A_2A_3 \ldots An) \approx P(A_i|A(i-k) \ldots A(i-1)$$

Where, P is the probability of occurrence of a word (bigram mapping), Ai is the ith word, and k is the mid index.

2) Ensemble Learning Model

The ensemble learning model mixes various algorithms to use their complementing qualities. By combining predictions from multiple models, the ensemble harnesses the varied views and captures a greater range of patterns and relationships in the data, contributing to the recommendation system's resilience, accuracy, and generalizability. The ensemble technique compensates for the shortcomings of individual models, resulting in a more dependable and effective system overall. In our recommendation system, we employ LR, NB, RF, DT, and XG Boost.

3) Logistic Regression (LR)

LR is a classification algorithm that models the link between independent variables and the likelihood of a specific outcome. It is well-suited for binary classification applications and can efficiently handle big datasets. LR maps the input information to the desired output using a logistic function and predicts user preferences in a favorable manner in our recommendation system [33]. It makes use of a sigmoid function, which produces a probability between 0 and 1. It is calculated by

$$f(y^i, \theta) = \frac{1}{1 + e^{\theta T x^i}}$$

Where f is the sigmoid function, and x is the factor that determines whether the sigmoid function will tend to 0 or 1. A value near 1 of the sigmoid function represents the predicted probability of the positive class [34]. The cost function is calculated by

$$M() = \frac{-1}{n} * \Sigma[x(i) log(f(y(i)), \theta) + (1 - x(i)) log(1 - f(y(i)), \theta))]$$

The function θ is improved by

$$\theta_m = \theta_m - \alpha * \frac{\partial m}{\theta_m}$$

Where α is the learning rate.

4) Naive Bayes (NB)

NB is a probabilistic classifier that employs Bayes' theorem with the "naive" assumption of feature independence. The Bayes' theorem is:

$$P(M|N) = \frac{P(N|M)P(M)}{P(N)}$$

NB performs well in text categorization in sentiment analysis. It is computationally efficient and works well even with limited training data [35]. NB's capacity to handle textual input, as well as its quick training and prediction timeframes, make it an important complement to our ensemble recommendation system.

5) Random Forest (RF)

We employ RF in our recommendation system since it improves the accuracy and stability of the ensemble system by using the strength of several decision trees. It is an ensemble learning method that makes predictions by combining many decision trees. Using bootstrapped samples and random feature subsets, it generates a diverse set of decision trees. This contributes to less overfitting and better generalization. RF is durable, scalable, and has the capacity to handle high-dimensional data [36].

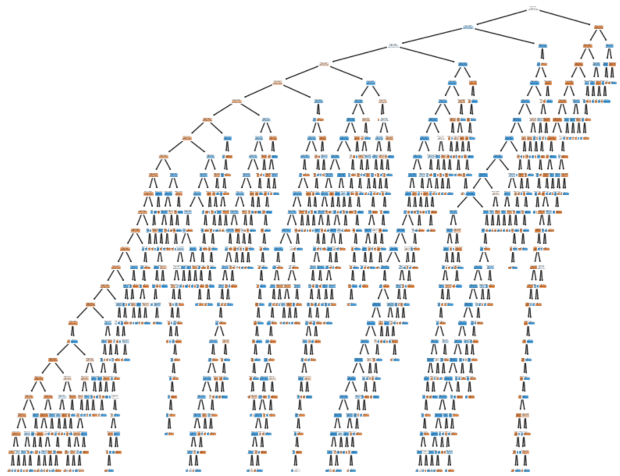


Figure 4. Gini index-based DT plot of the model

6) Decision Tree (DT)

DT constructs a flowchart-like model with each internal node representing a feature, each branch representing a decision rule, and each leaf node representing a predicted outcome. DTs are straightforward to interpret and can handle both numerical and categorical data. They are useful for capturing user preferences and item characteristics in our recommendation system since they can capture complicated linkages and interactions between features [37]. Figure 4 represents a DT visualization that demonstrates how the Gini index is utilized to separate nodes in the model's decision-making process. The Gini index or the cost function, which is evaluated to make splits in the dataset by DT, is calculated using:

$$G = 1 - \sum_{i=1}^{n} pi^2$$

7) XG Boost

XG Boost is a gradient boosting technique that combines gradient boosting concepts with regularisation approaches to generate a strong ensemble model [38]. It allows parallel processing, accommodates missing values, and offers a variety of objective functions and evaluation measures. The tremendous ensemble learning capabilities of XG Boost make it an important component of the recommendation system, enhancing accuracy and predictive performance. The objective function of XG Boost is calculated by:

$o^{(t)} = \sum_{i=1}^{n}\left(x_i - \left(\widehat{x}_i^{(t-1)} + m_t(y_i)\right)\right) + \sum_{i=1}^{t}\Omega(m_i)$

$= \sum_{i=1}^{n}\left[2\left(\widehat{x}_i^{(t-1)} - x_i\right) m_t(y_i)\right] + \Omega(m_i) + constant$

8) Item-based Recommendation

Item-based filtering in our recommendation system allows us to deliver personalized recommendations based on the similarity of items. It functions by analyzing past data from user interactions with goods and discovering common patterns [39]. Based on user reviews, purchase history, or other relevant indicators, the algorithm computes the similarity between items. Using this similarity metric, the algorithm identifies goods that are similar to those in which the user has previously expressed interest [40]. These related things are then recommended to the user, increasing the likelihood that the located items match their interests. Item features are important in assessing user preferences while recommending films, books, or products.

9) User-based Recommendation

We employ user-based recommendations in our system that work on user suggestions. Its primary goal is to find individuals who share similar tastes and preferences. The algorithm examines past data from user interactions, such as ratings, reviews, or purchasing behavior, and detects users with similar tendencies [41]. Based on this data, the algorithm recommends things that have gotten positive feedback from other users. The user-based recommendation gives personalized suggestions based on the preferences of people with similar tastes by utilizing the collective expertise of like-minded individuals. This method is useful when user attributes and preferences are more influential than object qualities in determining recommendations..

10) Evaluation Metrics

Accuracy: Accuracy is the ratio of accurately predicted instances to the total number of cases, including True Positives and True Negatives.

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP}$$

Where, TN is True Negatives, TP is True Positives, FN is False Negatives, FP is False Positives

Precision: By comparing the percentage of accurately predicted positive instances (positive user reviews) to all anticipated positive instances, precision calculates the accuracy of positive predictions.

$$Precision = \frac{TP}{TP + FP}$$

Recall: Recall quantifies the model's ability to identify every instance of a positive. It is the percentage of all positively impacted occurrences (positive user reviews) that were correctly predicted to happen.

$$Recall = \frac{TP}{TP + FN}$$

F1: An impartial model's performance is evaluated by the F1-Score, which is the harmonic mean of recall and precision. It provides beneficial information in cases where there is an uneven distribution of courses or data points.

$$F1 - Score = \frac{2 * Recall * Precision}{c}$$

RMSE: The average of the squared differences between predicted and actual ratings is taken into account in Root Mean Square Error.

$$RSME = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \widehat{y_i})^2}{n}}$$

Where, y_i is the actual or observed value $\widehat{y_i}$ is the predicted value. n are the total number of data points or observations.

5. Results

This section displays the results obtained after conducting experimentation on the dataset, as per the proposed model in Section III. Figure 5 depicts the Confusion Matrix and graphs displaying the True Positive Rate against the False Positive Rate for several Ensemble Learning Model Classifiers (LR, NB, DT, RF, and XG Boost), providing an overview of their classification job performance. The confusion matrix visualizes classifier performance by displaying true positives, false positives, true negatives, and false negatives. The graph depicts the trade-off between the true positive rate and the false positive rate, which aids in model evaluation and comparison. Table II presents the evaluation metric scores obtained by the ensemble learning subsystem of the hybrid recommendation system. XG Boost is the best-performing algorithm in the ensemble model achieving an accuracy of 96% on the test data set. This indicates the model's outstanding capacity in correctly predicting and classifying things in the recommendation system, suggesting its usefulness in producing accurate suggestions or predictions for users based on their preferences or behaviours. This performance demonstrates its potential to improve the system's capacity to recommend relevant products with high precision (98% for XG Boost).The model attains an RMSE Score of 3.55804067664864.

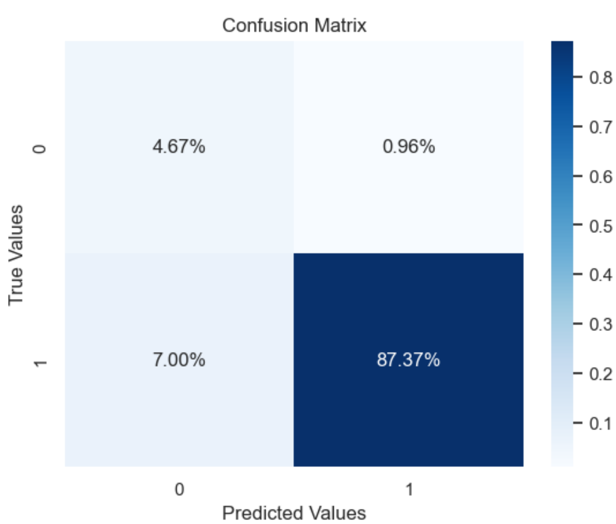
The outputs from different models are aggregated and weighted to generate the final recommendation through an ensemble approach. Each model, including logistic regression (LR), Naive Bayes (NB), decision trees (DT), random forests (RF), and XGBoost, contributes to the final prediction based on its individual strengths. The system assigns weights to each model's output, reflecting its performance on specific tasks, such as sentiment analysis, item-based filtering, or user-based recommendations. XGBoost, given its superior accuracy, is assigned a higher weight, while models like LR and NB, which excel in handling linear relationships and textual data, respectively, are weighted accordingly. The weighted outputs are then combined to produce a single, unified recommendation, ensuring that the system leverages the best attributes of each model to provide robust and accurate suggestions. This ensemble strategy enhances the overall performance by balancing the strengths

Table II. Evaluation Metrics Score of the Ensemble Learning Model

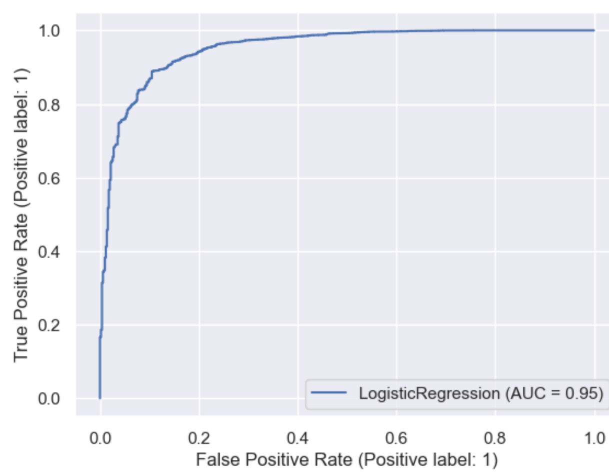
Metric	Logistic Regression	Naive Bayes	Decision Tree	Random Forrest	XG Boost
Accuracy	0.92	0.89	0.91	0.94	0.96
Precision	0.99	0.99	0.98	0.98	0.98
Recall	0.93	0.89	0.92	0.95	0.98
F1Score	0.96	0.94	0.95	0.97	0.98
Auc Score	0.95	0.94	0.84	0.95	0.95

Table III. Comparison Analysis Table

Paper	Model Used	Better Performing Algorithm	Dataset Used	Accuracy
[5]	Hybrid - Collaborative Filtering and Content-based filtering.	Collaborative Filtering Algorithm	Library dataset- Inner Mongolia University of Technology	92%
[7]	Ensemble Learning(fuzzy neural networks, SVC), graph embedding	SVC	MovieLens dataset	76%
Our Model	Hybrid - Sentiment Analysis, Ensemble Learning Model (LR, RF, DT, NB, XG Boost),	XG Boost	Custom - Amazon reviews, Google reviews, Tmdb Reviews	96%



5(A)



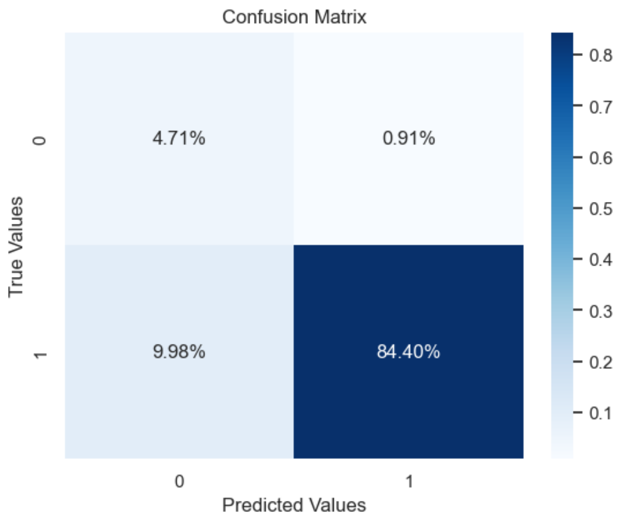
5(B)

and weaknesses of individual models, leading to a more reliable and personalized recommendation system.

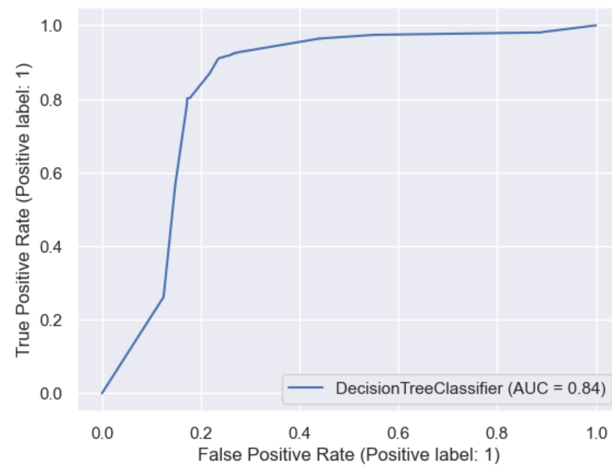
6. Comparative Analysis

This section presents a comparative analysis of the performance of the hybrid recommendation system model used by our research and the models used in papers [5], [7]. Table III presents the comparison analysis. We intricately present a comparative analysis of the algorithms and methods used in these papers and conclude that our model performs consistently well, achieving an accuracy of 96%, which is the highest amongst other models.

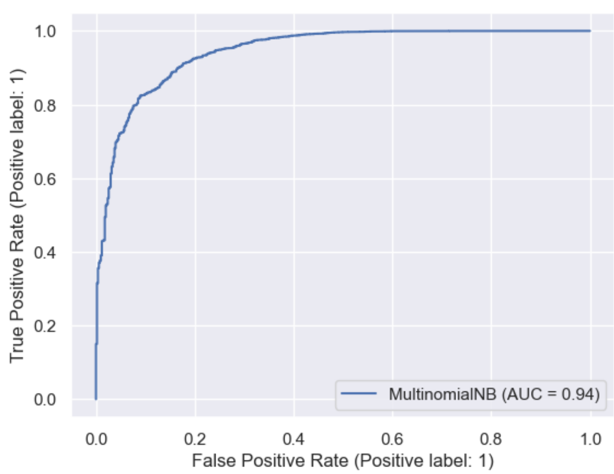
While the XGBoost algorithm significantly contributes to the model's high accuracy (96% on a custom dataset),



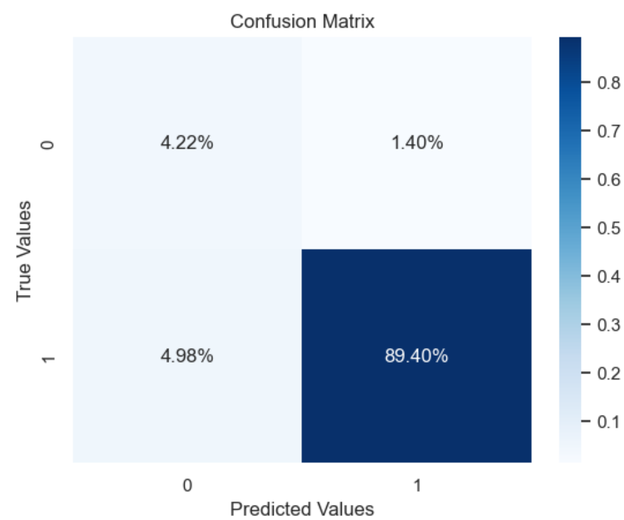
5(C)



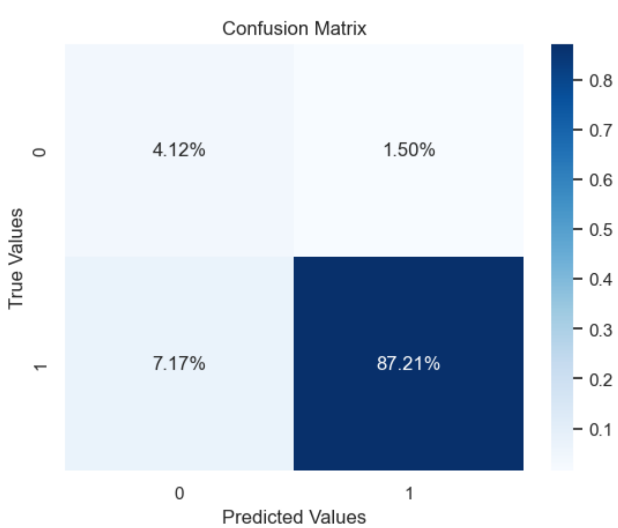
5(F)



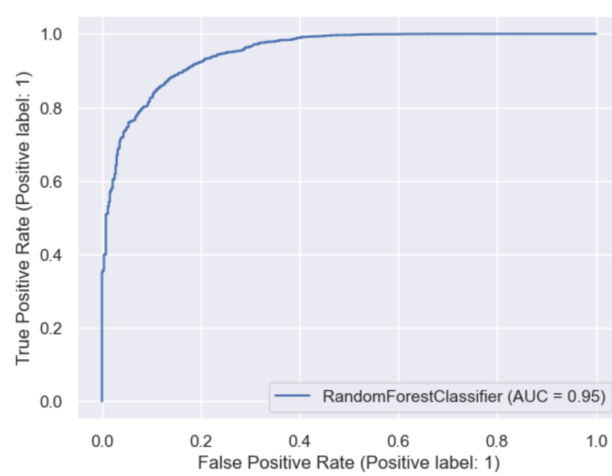
5(D)



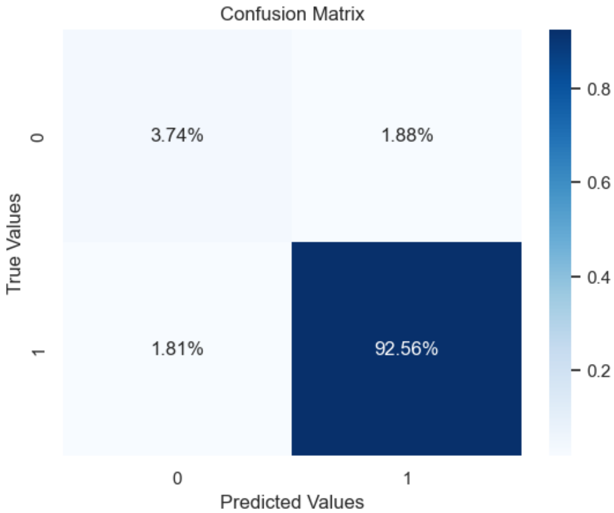
5(G)



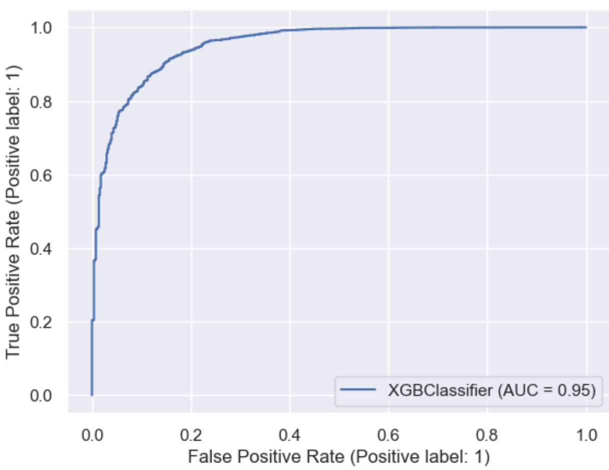
5(E)



5(H)



5(I)



5(J)

Figure 5. (A)-(J)Confusion Matrix and True Positive Rate Vs. False Positive Rate Graphs for the Ensemble Learning Model Classifiers - LR, NB, DT, RF and XG Boost

it also demands substantial computational resources and time for training, especially when compared to simpler models like logistic regression or Naive Bayes. In contrast, models with lower computational overhead, such as SVC or collaborative filtering, might offer faster predictions but at the cost of reduced accuracy. This trade-off highlights the need to balance the system's performance with practical considerations like processing time and resource availability, especially in large-scale or real-time applications. Integrating more efficient models or optimizing ensemble strategies could help mitigate these concerns without compromising accuracy.

7. Future Scope

The proposed hybrid recommendation system serves as a good start for such future developments in recommendation technology. Solutions for improving the system's performance and utility are various. The natural language processing to improve the sentiment analysis model is a vital area for future research. This will allow a more in-depth understanding of the nuances of the emotions expressed in user evaluations and, in turn, improve suggestion customisation and relevance. So, as opposed to numerical static embeddings, contextual embeddings like BERT or GPT-based models are able to detect nuanced sentiments plus complex language and hence generate more accurate sentiment-based recommendations. The second means of another improvement is handling class imbalances using some more sophisticated oversampling techniques. SMOTE is good, but ADASYN or Borderline-SMOTE might provide more sensible synthetic samples. However, for datasets with sparse minority classes, this may lead to better system performance in scenarios with massive imbalance. Deep learning models like recurrent neural networks (RNNs) or transformers could be integrated within the model to significantly improve its capacity to detect complex patterns and sequential relationships among data. It may also be possible to analyse the time dynamics of user preferences and adapt the system to time-varying patterns. By integrating time-series methodologies with recommendation models, the system advances from delivering static recommendations to delivering more dynamic and customised recommendations as user preferences change with time. Future research initiatives in the area of explainability are needed. That said, to gain consumer trust, companies in sensitive industries, e.g., healthcare and finance, must provide transparent and understandable guidance, as consumers become increasingly sophisticated consumers and want to be certain that the advice they follow online is both neutral and free of any ulterior motives. In the future, XAI techniques can be integrated with each recommendation to explain why they were made.

8. Conclusion

The contribution of this study is to demonstrate that our hybrid recommendation system can vastly improve user experience by providing very personalised, robust, and accurate recommendations. Geared towards optimal performance, traditional recommendation models are complemented with the use of ensemble machine learning models such as logistic regression (LR), Naive Bayes (NB), decision trees (DT), random forest (RF), and XGBoost. In some sectors, including e-commerce, media, and social platforms, the system achieved 96% accuracy while predicting user preferences and suggesting appropriate items. We strengthen our model using a diverse dataset spanning from Amazon, TMDb, and Google Reviews, which means our model can be applied to multiple content types. The combination of SMOTE in addressing class imbalance and in feature extraction provides more robustness of the model, making it workable for skewed data distributions. This research has implications beyond the more straightforward accuracy assessment. For this application, we employ a hybrid methodology, which improves the performance and marks an opening up of new ground to combine item-based and user-based filtering with sentiment analysis to produce customised recommendations. This study will lay the groundwork for future systems attempting to provide significant recommendations given the evolution of the digital landscape and the interest in more customised content. However, there are limitations set by the current study that need to be recognised in future research. On the assessed datasets, the system demonstrates robust performance, but the effect of the system in other domains, especially those with less structured datasets, is not explored. Additionally, such a sentiment analysis model relies on traditional NLP techniques, thus limiting the system's ability to grasp all the complexity of modern language like slang and colloquialisms effectively, and hence the relevance of recommendations will vary from context to context. Some of the data sparsity in the regions where there is little user feedback may compromise the efficacy of the model. In such cases, hybrid models based on content-based and collaborative filtering are optimised to mitigate this issue. In conclusion, our work lays the foundation for the generation of future recommendation systems that are much more sophisticated. Using sophisticated machine learning, deep learning, and natural language processing techniques, we show how these technologies are used to implement state-of-the art technology to deliver improved recommendations. Providing a scalable and flexible architecture with flexibility to serve the increasing need for customised user experiences on diverse domains.

References

[1] I. Balush, V. Vysotska, and S. Albota, "Recommendation system development based on intelligent search, NLP and machine learning methods," *CEUR Workshop Proceedings*, vol. 2917, pp. 584–617, 2021.

[2] R. Logesh, V. Subramaniyaswamy, D. Malathi, N. Sivaramakrishnan, and V. Vijayakumar, "Enhancing recommendation stability of collaborative filtering recommender system through bio-inspired clustering ensemble method," *Neural Computing and Applications*, vol. 32, no. 7, pp. 2141–2164, apr 2020.

[3] A. Da'u and N. Salim, "Recommendation system based on deep learning methods: a systematic review and new directions," *Artificial Intelligence Review*, vol. 53, no. 4, pp. 2709–2748, apr 2020.

[4] D. F. Murad, Y. Heryadi, B. D. Wijanarko, S. M. Isa, and W. Budiharto, "Recommendation System for Smart LMS Using Machine Learning: A Literature Review," in *2018 International Conference on Computing, Engineering, and Design (ICCED)*. IEEE, sep 2018, pp. 113–118.

[5] Y. Tian, B. Zheng, Y. Wang, Y. Zhang, and Q. Wu, "College Library Personalized Recommendation System Based on Hybrid Recommendation Algorithm," *Procedia CIRP*, vol. 83, pp. 490–494, 2019.

[6] Z. Zamanzadeh Darban and M. H. Valipour, "GHRS: Graph-based hybrid recommendation system with application to movie recommendation," *Expert Systems with Applications*, vol. 200, p. 116850, 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417422003025

[7] S. Forouzandeh, K. Berahmand, and M. Rostami, "Presentation of a recommender system with ensemble learning and graph embedding: a case on MovieLens," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 7805–7832, 2021. [Online]. Available: http://link.springer.com/10.1007/s11042-020-09949-5

[8] N. Nikzad-Khasmakhi, M. Balafar, M. Reza Feizi-Derakhshi, and C. Motamed, "BERTERS: Multimodal representation learning for expert recommendation system with transformers and graph embeddings," *Chaos, Solitons Fractals*, vol. 151, p. 111260, oct 2021.

[9] S. Renjith, A. Sreekumar, and M. Jathavedan, "An extensive study on the evolution of context-aware personalized travel recommender systems," *Information Processing Management*, vol. 57, no. 1, p. 102078, 2020.

[10] Y. Deldjoo, T. D. Noia, and F. A. Merra, "A Survey on Adversarial Recommender Systems," *ACM Computing Surveys*, vol. 54, no. 2, pp. 1–38, mar 2022.

[11] A. Marchand and P. Marx, "Automated Product Recommendations with Preference-Based Explanations," *Journal of Retailing*, vol. 96, no. 3, pp. 328–343, 2020.

[12] X. Shi, Q. He, X. Luo, Y. Bai, and M. Shang, "Large-scale and Scalable Latent Factor Analysis via Distributed Alternative Stochastic Gradient Descent for Recommender Systems," *IEEE Transactions on Big Data*, pp. 1–13, 2020.

[13] J. Yu, H. Yin, X. Xia, T. Chen, J. Li, and Z. Huang, "Self-Supervised Learning for Recommender Systems: A Survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 1, pp. 335–355, jan 2024.

[14] V. Arnaboldi, M. Giovanni, and F. Delmastro, "PLIERS: a Popularity-Based Recommender System for Content Dissemination in Online Social Networks," in *SAC 2016*, 2016, pp. 671–673.

[15] J. Ji, Z. Li, S. Xu, W. Hua, Y. Ge, J. Tan, and Y. Zhang, "GenRec: Large Language Model for Generative Recommendation," 2024, pp. 494–502.

[16] S. Wang, L. Cao, Y. Wang, Q. Z. Sheng, M. A. Orgun, and D. Lian, "A Survey on Session-based Recommender Systems," *ACM Computing Surveys*, vol. 54, no. 7, pp. 1–38, sep 2022.

[17] X. Yang, S. Zhou, and M. Cao, "An Approach to Alleviate the Sparsity Problem of Hybrid Collaborative Filtering Based Recommendations: The Product-Attribute Perspective from User Reviews," *Mobile Networks and Applications*, vol. 25, no. 2, pp. 376–390, apr 2020.

[18] S. Natarajan, S. Vairavasundaram, S. Natarajan, and A. H. Gandomi, "Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data," *Expert Systems with Applications*, vol. 149, p. 113248, jul 2020.

[19] S. Zhou, X. Dai, H. Chen, W. Zhang, K. Ren, R. Tang, X. He, and Y. Yu, "Interactive Recommender System via Knowledge Graph-enhanced Reinforcement Learning," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM, jul 2020, pp. 179–188.

[20] S. Dhawan, K. Singh, A. Batra, A. Choi, and E. Choi, "An Optimized Ranking Based Technique towards Conversational Recommendation Models," *International Journal of Computing and Digital Systems*, vol. 15, no. 1, pp. 1201–1216, mar 2024. [Online]. Available: https://journals.uob.edu.bh/handle/123456789/5089

[21] S. Dhawan, K. Singh, A. Rabaea, and A. Batra, "ImprovedGCN: An efficient and accurate recommendation system employing lightweight graph convolutional networks in social media," *Electronic Commerce Research and Applications*, vol. 55, p. 101191, sep 2022.

[22] D. Sanjeev, K. Singh, E.-M. Craciun, A. Rabaea, and A. Batra, "Next-Cart Recommendation by Utilizing Personalized Item Frequency Information in Online Web Portals," *Neural Processing Letters*, vol. 55, no. 7, pp. 9409–9434, dec 2023. [Online]. Available: https://link.springer.com/10.1007/s11063-023-11207-2

[23] S. Dhawan, K. Singh, A. Batra, A. Choi, and E. Choi, "A Novel Deep Learning Approach Toward Efficient and Accurate Recommendation Using Improved Alternating Least Squares in Social Media," *Journal of The Institution of Engineers (India): Series B*, feb 2024. [Online]. Available: https://link.springer.com/10.1007/s40031-024-00999-z

[24] B. Ramzan, I. S. Bajwa, N. Jamil, R. U. Amin, S. Ramzan, F. Mirza, and N. Sarwar, "An Intelligent Data Analysis for Recommendation Systems Using Machine Learning," *Scientific Programming*, vol. 2019, pp. 1–20, oct 2019.

[25] H.-W. Chen, Y.-L. Wu, M.-K. Hor, and C.-Y. Tang, "Fully content-based movie recommender system with feature extraction using neural network," in *2017 International Conference on Machine Learning and Cybernetics (ICMLC)*. IEEE, jul 2017, pp. 504–509.

[26] W. Shafqat and Y.-C. Byun, "A Hybrid GAN-Based Approach to Solve Imbalanced Data Problem in Recommendation Systems," *IEEE Access*, vol. 10, pp. 11 036–11 047, 2022.

[27] L. Kumar, S. M. Satapathy, and A. Krishna, "Application of SMOTE and LSSVM with Various Kernels for Predicting Refactoring at Method Level," 2018, pp. 150–161.

[28] C. N. Dang, M. N. Moreno-García, and F. D. la Prieta, "An Approach to Integrating Sentiment Analysis into Recommender Systems," *Sensors*, vol. 21, no. 16, p. 5666, aug 2021.

[29] A. J. Costa, M. S. Santos, C. Soares, and P. H. Abreu, "Analysis of Imbalance Strategies Recommendation using a Meta-Learning Approach," in *7th ICML Workshop on Automated Machine Learning*, no. 9, 2020.

[30] E. Montañés, J. R. Quevedo, I. Díaz, and J. Ranilla, "Collaborative tag recommendation system based on logistic regression," *CEUR Workshop Proceedings*, vol. 497, pp. 173–188, 2009.

[31] L. Shuxian and F. Sen, "Design and Implementation of Movie Recommendation System Based on Naive Bayes," *Journal of Physics: Conference Series*, vol. 1345, no. 4, p. 042042, nov 2019.

[32] G. A. A. MULLA, Y. DEMİR, and M. HASSAN, "Combination of PCA with SMOTE Oversampling for Classification of High-Dimensional Imbalanced Data," *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, vol. 10, no. 3, pp. 858–869, sep 2021.

[33] A. Ajesh, J. Nair, and P. S. Jijin, "A random forest approach for rating-based recommender system," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, sep 2016, pp. 1293–1297.

[34] S. D. Jadhav and H. P. Channe, "Efficient Recommendation System Using Decision Tree Classifier and Collaborative Filtering," *International Research Journal of Engineering and Technology(IRJET)*, vol. 3, no. 8, pp. 2113–2118, 2016.

[35] A. K. Chakraborty, S. Das, and A. K. Kolya, "Sentiment Analysis of Covid-19 Tweets Using Evolutionary Classification-Based LSTM Model," 2021, pp. 75–86.

[36] L. Xu, J. Liu, and Y. Gu, "A Recommendation System Based on Extreme Gradient Boosting Classifier," in *2018 10th International Conference on Modelling, Identification and Control (ICMIC)*. IEEE, jul 2018, pp. 1–5.

[37] A. Dadhich and B. Thankachan, "Sentiment Analysis of Amazon Product Reviews Using Hybrid Rule-Based Approach," 2022, pp. 173–193.

[38] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*. New York: ACM, apr 2001, pp. 285–295.

[39] C. Ajaegbu, "An optimized item-based collaborative filtering algorithm," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 12, pp. 10 629–10 636, dec 2021.

[40] S. Ko and J. Lee, "User preference mining through collaborative filtering and content based filtering in recommender system," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2455 LNCS, pp. 244–253, 2002.

[41] M. Kommineni, P. Alekhya, T. M. Vyshnavi, V. Aparna, K. Swetha, and V. Mounika, "Machine Learning based Efficient Recommendation System for Book Selection using User based Collaborative Filtering Algorithm," in *2020 Fourth International Conference on Inventive Systems and Control (ICISC)*. IEEE, jan 2020, pp. 66–71.