International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. #, No.# (Mon-20..)

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

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

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.

E-mail:author's email

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 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

A systematic literature review on deep learning-based recommendation systems authored by Da'u et al. [3], highlights auto encoder models as dominant, with movielens and Amazon datasets prevailing. It uses hybrid CNN and RNN architecture. The authors conduct a systematic literature review (SLR) to give a thorough overview of deep learning-based recommendation systems. Their findings highlight the importance of autoencoder models in this emerging sector, offering insight into their widespread use and effectiveness in a variety of applications. The authors also explain how the MovieLens and Amazon datasets are commonly used as core standards for measuring the efficacy of deep learning-based recommendation systems. Notably, the study describes a novel hybrid architecture that combines CNNs and RNNs, demonstrating the innovative ways being investigated to improve recommendation system performance

Existing research architectures on recommender systems are summarised by Murad et al. [4] online learning, recognizing the major focus on e-commerce while emphasizing the need for additional development of recommendation methods particular to online education. It lays the framework for future research by proposing the development of a personalized, context-aware recommendation engine for online learning based on ML within a smart Learning Management System (LMS). They highlight the predominance of research efforts in e-commerce sectors, emphasizing the need for additional breakthroughs in recommendation systems adapted exclusively to the realm of online education. Recognizing this deficit, the authors propose a forward-looking agenda for future research projects. They advocate for the creation of a personalized, context-aware recommendation engine designed exclusively for online learning environments. This recommendation engine is intended to use machine learning (ML) techniques in the context of a smart Learning Management System (LMS). A hybrid recommendation system tailored for college libraries, combining collaborative filtering and content-based algorithms to improve book recommendations, is introduced by Tian et al. [5]. One of the key difficulties they address is data sparsity, which is a typical problem in recommendation systems, by employing strategies to improve the user-item matrix. They use clustering approaches to improve accuracy. They confirm the efficiency of their strategy with rigorous comparative studies on the Inner Mongolia

University of Technology library dataset. The authors expand upon their methodology by incorporating the hybrid recommendation algorithm within the Spark big data platform. This integration allows the system to process vast amounts of data efficiently, making it suitable for major university libraries. The authors demonstrate the actual use of their hybrid technique in a real-world situation, highlighting its potential for providing personalized book recommendations based on individual users' tastes and interests. The authors address data sparsity through user-item matrix improvement and clustering for improved accuracy which is validated through comparative experiments on the library dataset at the Inner Mongolia University of Technology.

Darban et al. [6] presented a graph-based hybrid recommendation system (GHRS). This unique recommendation system method incorporates user similarity in ratings, demographic data, and location information into a graph-based model. This novel approach combines many sources of user information, such as ratings, demographic data, and geographical information, into a single graph-based model. By using the graph's intrinsic structure and interactions, GHRS effectively captures user similarities and preferences, improving recommendation accuracy. One crucial element of GHRS is its use of Autoencoder feature extraction algorithms, which allow the system to identify important patterns and latent characteristics from input data. GHRS applies grouping algorithms for coupled qualities, resulting in more robust and personalized suggestions. GHRS outperforms previous algorithms, notably in addressing the cold-start problem, which is a major challenge in recommendation systems. This demonstrates GHRS's ability to provide accurate and relevant recommendations, even for individuals with minimal historical data or diverse platforms.

A novel approach that used group classification, ensemble learning (using fuzzy neural networks and SVR), and graph embedding to improve the accuracy of recommender systems was significantly introduced by Forouzandeh et al. [7]. This addressed the challenge of precise user behavior prediction in large datasets, as evidenced by significant efficiency improvements on MovieLens datasets. Their approach consists of three main components: group classification, ensemble learning with fuzzy neural networks and SVR, and graph embedding approaches. By combining these aspects, the authors hoped to address the challenges involved in forecasting user behaviour across large datasets, with the MovieLens datasets serving as key test beds.

BERTERS, a multimodal classification strategy for expert recommendation systems, was introduced by Khasmakhi et al. [8]. BERTERS translates text into vectors and collects features from co-author networks, which are then concatenated for classification. It performs well in multi-label classification and visualization tasks relevant to academic and community question-answering domains. This novel approach uses the capabilities of both text and graph modalities to improve the recommendation process. BERTERS combines two key components: text representation using BER (Bidirectional Encoder Representations from Transformers), a cutting-edge language model known for its contextual understanding of textual data, and graph embedding with ExEm (Expert Embedding Model), which captures structural information inside co-author networks. By combining these modalities, BERTERS efficiently converts textual inputs to vector representations while also collecting information from co-author networks. The collected characteristics are then concatenated to create a comprehensive representation for classification tasks. BERTERS performs well in cases requiring multi-label classification and visualization, particularly in the academic and community question-answering domains. Multimodal representation learning, Transformer (BERT), and graph embedding (ExEm) are three key strategies. The success of BERTERS can be attributed to its adept use of three key strategies: multimodal representation learning, which exploits various data modalities to enrich the recommendation process; transformer-based text representation via BERT, which allows for nuanced understanding of textual content; and graph embedding facilitated by ExEm, which allows for the extraction of structural information from co-author networks.

The evolution of travel recommendation systems was investigated by Renjith et al. [9], focusing on their features and limits. It examines the transition from generic engines to personalized and contextualized AI-powered systems. The inquiry follows the evolution of crude, one-size-fits-all engines to the introduction of personalized and context-aware AI-driven systems. The study examines important categorization schemes and recommendation algorithms, emphasizing their critical significance in improving prediction accuracy, refining suggestion mechanisms, and optimizing ranking procedures within the dynamic environment of the travel and tourism business.

The vulnerability of collaborative filtering-based recommender systems to adversarial examples was emphasized by Deldjoo et al. [10]. It reviews recent improvements in adversarial machine learning (AML) for increasing recommender system security. It explores the successful use of AML in generative adversarial networks (GANs) to improve the quality of generative models. The survey examines 76 articles from prestigious journals and conferences, providing a significant resource for scholars working on securing recommender systems and improving generative models with GANs. The investigation explores recent advances in adversarial machine learning (AML), with a special emphasis on its use to improve recommender system security. It conducts a comprehensive overview of how AML approaches, particularly generative adversarial networks (GANs), have been used to improve the quality and robustness of generative models. A unique method integrating content and collaborative filtering was introduced by Marchand et al. [11] that beats established recommender approaches in predicted accuracy, outperforming the Netflix Prize-winning algorithm by more than 5%, according to normalized root mean squared error. To address this issue, the authors suggest that explaining these guidelines can reduce the possible harm. They present a novel strategy for combining content-based and collaborative filtering algorithms. Surprisingly, this approach outperforms established recommender systems in terms of predictive accuracy, and the authors emphasize that their strategy not only improves accuracy but also can provide actionable explanations for recommendations, which aligns with ethical criteria for artificial intelligence systems. A distributed alternative stochastic gradient descent (DASGD) solution for latent factor analysis (LFA)--based recommender systems, as proposed by Shi et al. [12], which addresses scalability concerns with large datasets. DASGD separates training dependencies between latent features and uses efficient data partitioning, allocation, and job parallelization algorithms to reduce communication costs. It efficiently reduces the transmission costs associated with distributed optimization. The study's experimental results show that DASGD outperforms existing distributed Stochastic Gradient Descent (SGD) solvers in terms of both prediction accuracy and scalability. This makes DASGD an appealing method for training LFA-based recommenders on large-scale, high-dimensional, and sparsely populated matrices, taking advantage of the resources available in cloud computing settings. Overall, the study emphasizes the importance of DASGD as an efficient and effective approach for tackling the scalability issues inherent in recommender systems that operate on large datasets. The emerging topic of self-supervised recommendation (SSR) is deciphered Yu et al. [13]. They provide a unique definition of SSR and create a comprehensive taxonomy for categorizing existing SSR methods into four categories: contrastive, generative, predictive, and hybrid. They provide a thorough knowledge of the landscape of SSR techniques by explaining the principles, formulations, procedures, and respective advantages and disadvantages of each category. They also distribute an open-source library called SELFRec, which incorporates a wide range of SSR models and benchmark datasets, allowing for empirical comparisons and facilitating future studies on the subject.

A PLIER, a unique tag-based recommender system, is presented by Arnaboldi et al. [14]. It aims to find a balance between algorithmic complexity and personalized recommendations. PLIERS works on the notion that people prefer things and tag that are as popular as those they already own. Using this idea, PLIERS strives to provide highly personalized recommendations while remaining computationally efficient. The authors show that PLIERS surpass existing state-of-the-art systems in terms of recommendation personalization, relevance, and novelty using real online social network (OSN) datasets. This paper advances the subject by proposing a realistic and effective recommendation system that meets the simultaneous challenges of complexity and personalization. A novel approach to recommender systems that leverages the capability of Large Language Models (LLMs) within the generative recommendation framework is introduced by the authors Ji et al. [15]. Their revolutionary technology, known as GenRec, differs from typical discriminative recommendation systems in that it uses raw text data, notably item names or titles, as item IDs. GenRec uses LLMs' expressive capabilities to immediately create recommendations rather than computing ranking scores for individual candidate items. By creating specialized prompts, the authors improve the LLM's ability to interpret recommendation tasks, allowing for the development of more relevant recommendations. They apply LoRA fine-tuning to tailor the LLM to user-item interaction data, which effectively captures user preferences and item characteristics contained in raw text.

Wang et al. [16] provide an important contribution to the field of recommender systems in by addressing the developing paradigm of session-based recommender systems (SBRSs) in their thorough review. In an age of information overload and digitized economies, where quick and accurate recommendations are critical, SBRSs provide a novel solution by collecting dynamic user preferences during short-term sessions. Recognizing the absence of unified issue statements and extensive explanations of SBRS characteristics and problems, the authors conduct a thorough examination of SBRS entities, behaviours, and qualities. An important contribution to the field of recommender systems is developed Yang et al. [17], tackling the widespread sparsity problem that impairs suggestion performance. They note a significant hurdle in their previously proposed Product Attribute Model, namely that the subjectivity of product reviews results in insufficient customer preference information, compounding sparsity issues. To solve this, they suggest a novel sparsity alleviation recommendation strategy that includes an algorithm intended to handle zero values using the Multiplication Convergence Rule and Constraint Condition. This approach effectively replaces zero values with equations, increasing matrix factorization accuracy and minimizing the model's sparsity problem. The authors present a hybrid collaborative formula that combines product attribute information, hence boosting recommendation outcomes. A major contribution to the field of recommender systems was developed by Natarajan et al. [18], tackling two critical issues encountered in collaborative filtering (CF): data sparsity and cold start. Their new method suggests two distinct models for addressing these difficulties effectively. First, they offer the Recommender System with Linked Open Data (RS-LOD) paradigm, which is specifically designed to relieve the cold start problem by utilizing a Linked Open Data (LOD) knowledge base, known as "DBpedia," to obtain necessary information about new entities. This novel technique ensures that personalized recommendations can be made even when insufficient previous data is available for new users or items. They introduce the Matrix Factorization model with Linked Open Data (MF-LOD), which seeks to address the data sparsity issue inherent in existing CF approaches.

The field of interactive recommender systems (IRS) is deciphered Zhou et al. [19], addressing the issues connected with the employment of reinforcement learning (RL) approaches in dynamic user contexts. They suggest a unique strategy that uses knowledge graphs (KG) to improve recommendation decisions. Instead of starting from scratch with RL rules, they employ past knowledge of item correlations learned via KGs to guide candidate selection, enrich item and user state representations, and propagate user preferences among correlated items. By incorporating KGs into the RL framework for IRS, the authors offer a novel solution to the sample efficiency problem, allowing for more effective recommendation strategies without the requirement for large interaction data. Moreover, several other authors also worked on recommendation systems such as Dhawan et al.[20]–[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. Despite the intrinsic temporal complexity of O(n^3) shared by both IALS and ALS due to iteratively solving linear equations, the addition of advanced optimisation methods in IALS highlights its potential for providing

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.

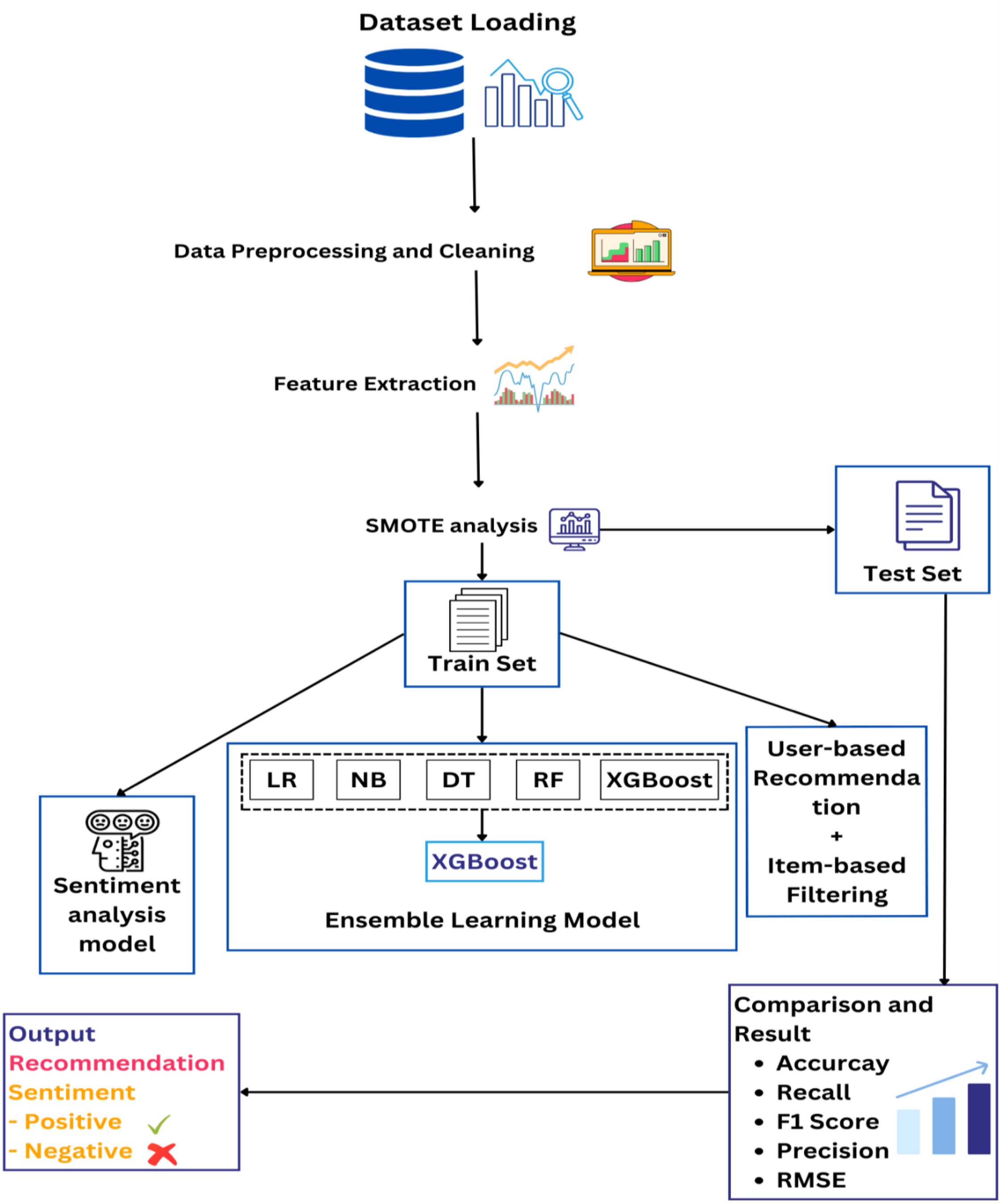


Figure 1. Block Diagram of the Proposed Model Architecture

5. Feature Extraction is performed after data preprocessing.

6. The SMOTE oversampling approach is implemented to address the dataset's extreme class imbalance [24].

E-mail:author's email

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

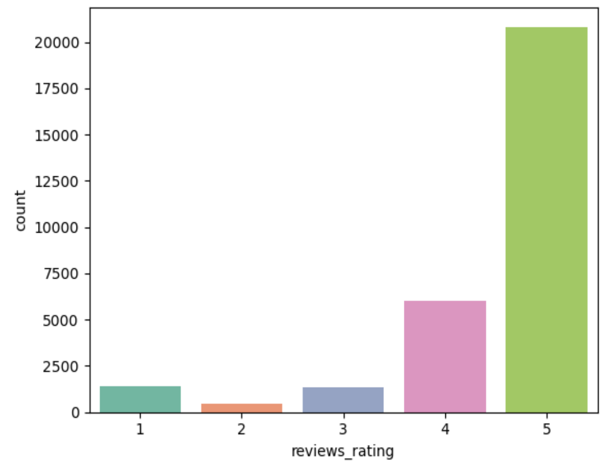
This section presents the detailed methodology followed during the experimentation of the proposed model in Section III.

A. Dataset Selection

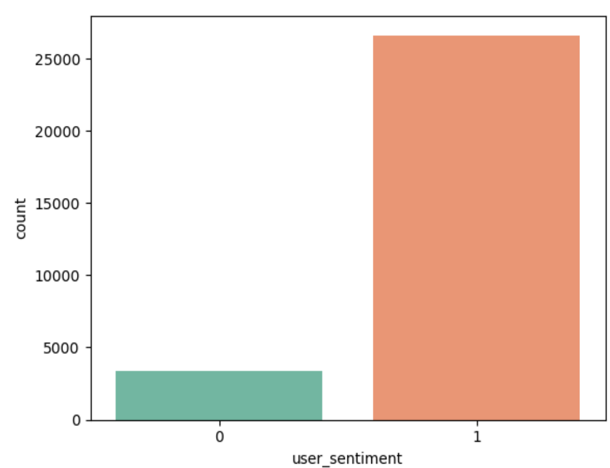
The dataset employed for the experimentation of our model architecture is curated from different sources to serve as a comprehensive dataset for a versatile recommendation system that includes a wide range of genres and products used in everyday life, such as movies, books, and more. The information 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	Attributes
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



2(a)

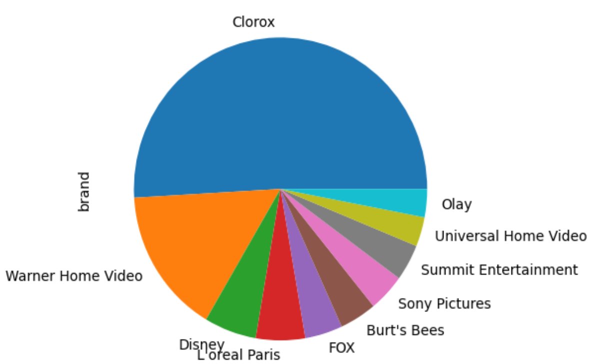


(b)

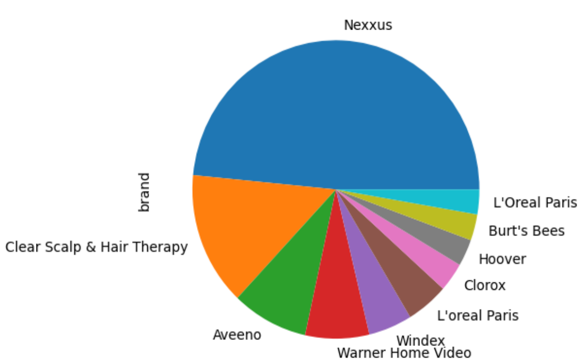
Figure 2. Graph Plot of data distribution of 'reviews_rating' and 'user_sentiment' of non-preprocessed dataset

B. Data Preprocessing

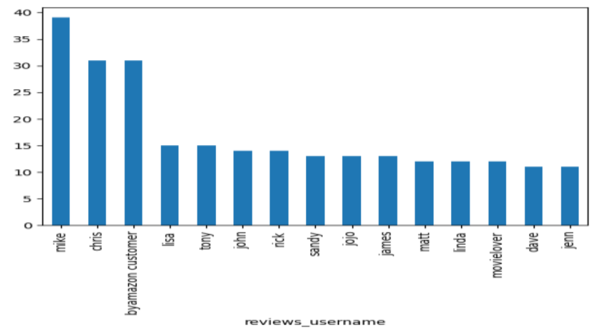
The loaded data is preprocessed to filter it based on areas or places of interest. This enables a better grasp of the data by focusing on certain regions essential to our recommendation system. Using the "reviews_username" column, we identify the users who frequently appear in the reviews. The top 15 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.



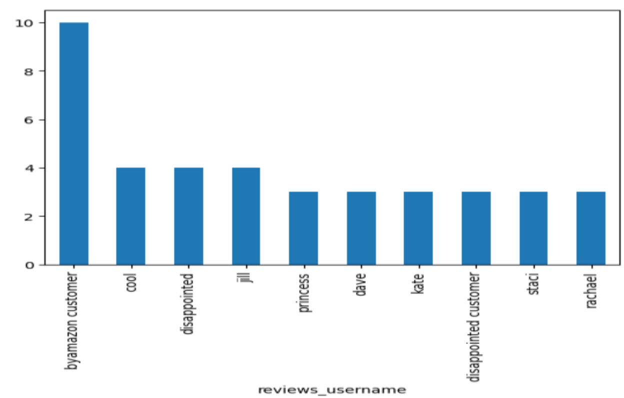
(a)



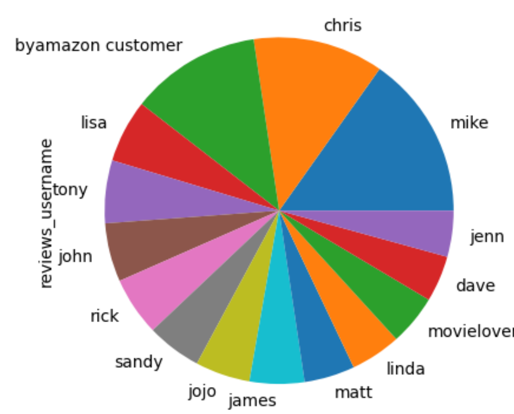
(b)



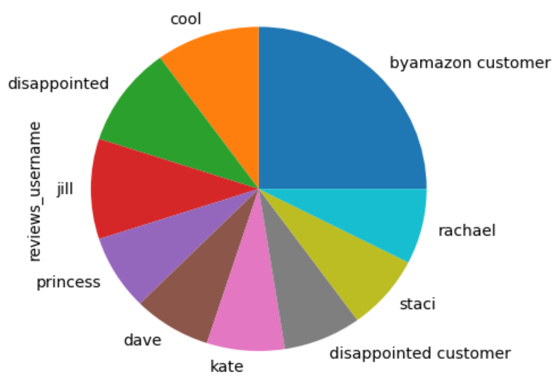
(c)



(d)



(e)



(f)

Figure 3. Data distribution after data preprocessing

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 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 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.

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:
    neighbors = find_k(ins, min_ins, k)
    for i in range(N):
      syn_ins = generate_ins(ins, neighbors)
      syn_inst.append(syn_ins)
  return syn_inst

Function find_k(ins, dataset, k):
  distances = empty list
  for data in dataset:
    distance = calculate_distance(instance, data_ins)
    distances.append(distance)
  sorted_indices = sort_indices(distances)
  k_nn = dataset[sorted_indices[:k]]
  return k_nn

Function generate_synthetic_instance(instance, neighbors):
  syn_ins = empty array
  rand_neig = select_neighbor(neighbors)
  for feature_index in range(num_features):
    diff = rand_neig[feature_index] - ins[feature_index]
    syn_feature = ins[feature_index] + rand_uniform(0, 1) * diff
    syn_ins.append(syn_feature)
  return syn_ins
```

Output:
- Synthetic minority class instances

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.

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:
   x = corpus[i]
   lemma = lemr.lemmatize(x[0], postag(x[1]))
   chars.append(lemma)
return joined_words
```

Output: -
- lemmatized text data

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:
   for each i in x.lower().split():
      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:

$$P(A1\ A2\ A3 \ldots\ An) \approx\ P(Ai\ |A(i-k) \ldots\ A(i-1)$$

Where, P is the probability of occurrence of a word (bigram mapping),

Ai is the ith word,

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.

- 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

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

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

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

The function θ is improved by

$$\theta m = \theta m - \alpha * \partial M(\theta)/\theta m$$

Where α is the learning rate.

- Naive Bayes (NB)

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

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

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.

- 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].

- 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:

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

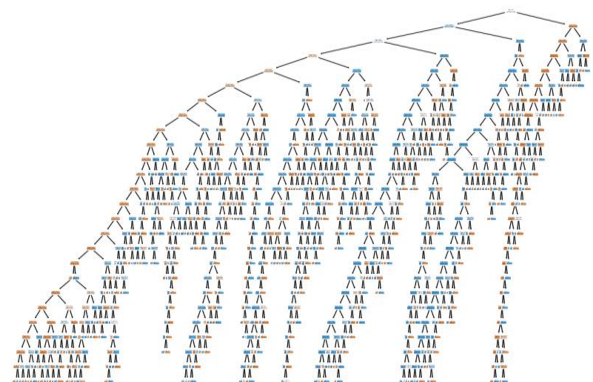


Figure 4. Gini index-based DT plot of the model

- 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:

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

3) 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.

4) 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..

5) Evaluation Metrics

1) *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

2) *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.

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

3) *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.

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

4) *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.

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

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

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

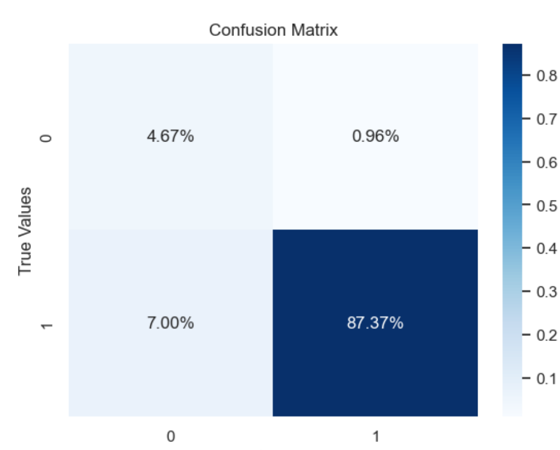
Where, y_i is the actual or observed value

$\hat{y}_i$ is the predicted value.

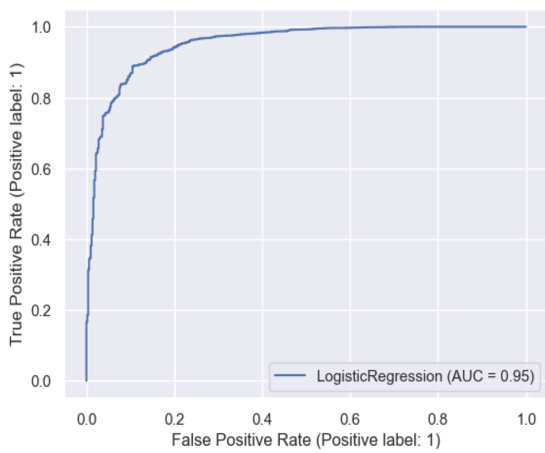
n are the total number of data points or observations.

5. RESULTS

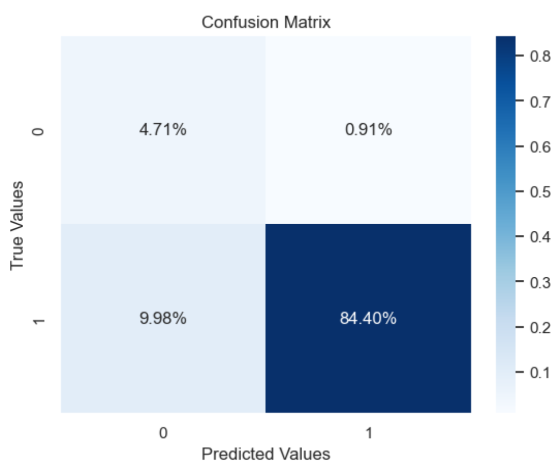
This section displays the results obtained after conducting experimentation on the dataset, as per the proposed model in Section III. Figure 5 depicts the Confusion Matrix and graphs displaying the True Positive Rate against the False Positive Rate for several Ensemble Learning Model Classifiers (LR, 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).



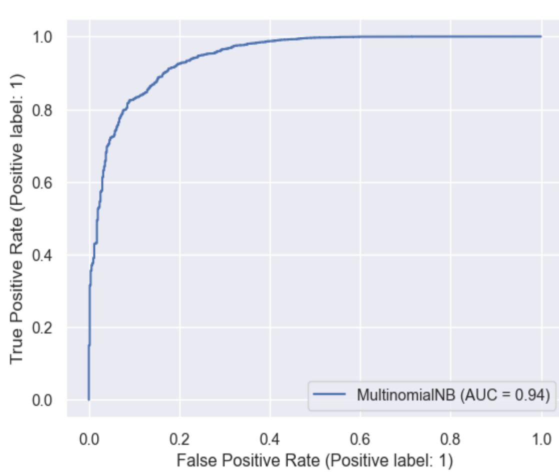
(a)


LogisticRegression (AUC = 0.95)
True Positive Rate (Positive label: 1)
False Positive Rate (Positive label: 1)

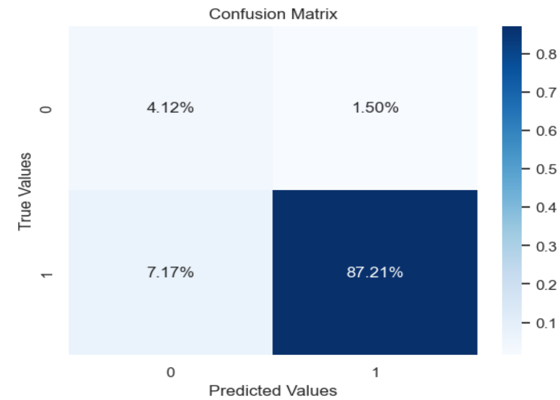
(b)


Confusion Matrix
4.71%
0.91%
9.98%
84.40%
True Values
Predicted Values

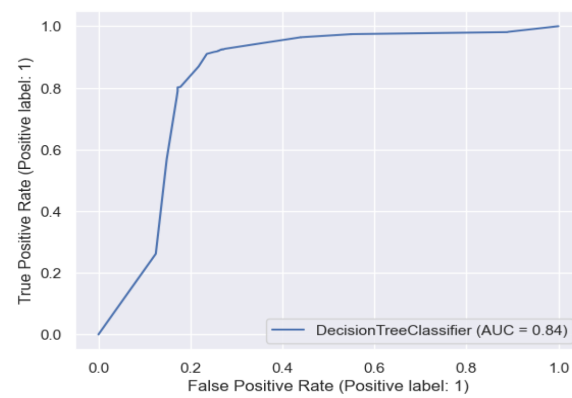
(c)


MultinomialNB (AUC = 0.94)
True Positive Rate (Positive label: 1)
False Positive Rate (Positive label: 1)

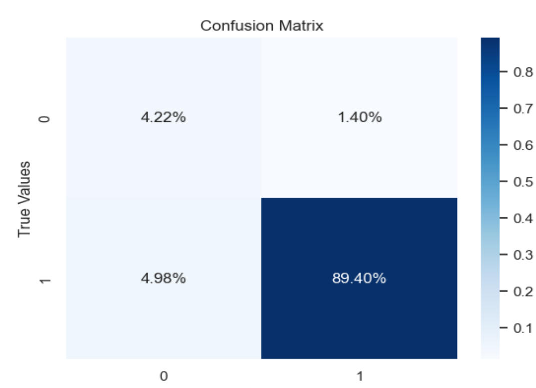
(d)


Confusion Matrix
4.12%
1.50%
7.17%
87.21%
True Values
Predicted Values

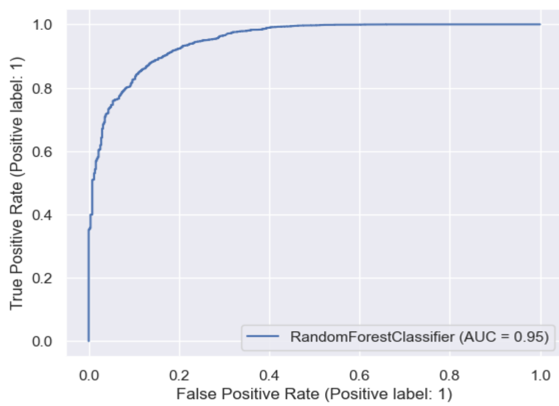
(e)


DecisionTreeClassifier (AUC = 0.84)
True Positive Rate (Positive label: 1)
False Positive Rate (Positive label: 1)

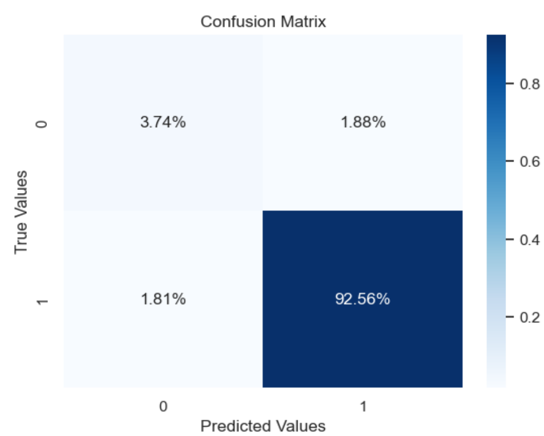
(f)


Confusion Matrix
4.22%
1.40%
4.98%
89.40%
True Values
Predicted Values

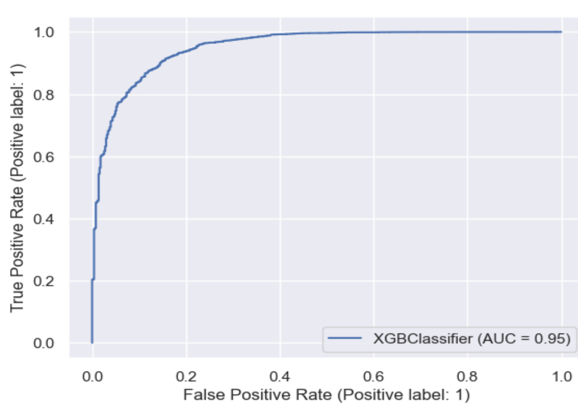
(g)



(h)



(i)



(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

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

The model attains an RMSE Score of 3.55804067664864.

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.

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%

.

7. FUTURE SCOPE

Potential future direction key points can be delved into for additional enhancement and exploration of our proposed hybrid recommendation system. Advanced NLP techniques can be used to enhance the sentiment analysis model, allowing it to capture more subtle sentiments and increase the accuracy of sentiment-based suggestions. Experimentations can be carried out with oversampling strategies such as ADASYN or Borderline-SMOTE, which will handle the class imbalance issue more successfully. Incorporating deep learning (DL) models, such as recurrent neural networks (RNNs) or transformers, will increase recommendation system effectiveness by capturing complex patterns and relationships in data. Taking into account the dataset's temporal component and applying time-series analysis techniques will allow the system to deliver more personalized and dynamic recommendations based on evolving user preferences and trends.

8. CONCLUSION

Our research provides a ground-breaking hybrid recommendation system that has the potential to transform the landscape of recommendation technology. Our approach delivers exceptional suggestion quality and robustness by employing ensemble models, such as sentiment analysis, and including both item-based and user-based filtering techniques. The use of a proprietary dataset compiled from many sources, such as Amazon, Tmdb, and Google reviews, assures broad coverage and relevancy in suggestions. Furthermore, correcting class imbalance with SMOTE and converting textual inputs into numerical representations using feature extraction approaches highlights our dedication to robust and effective modelling strategies.

The ensemble model, which includes LR, NB, DT, RF, and XGBoost, demonstrates our commitment to maximising the potential of machine learning approaches for personalised recommendation results. This combination of cutting-edge approaches not only increases suggestion accuracy to an astonishing 96%, but it also represents a huge step forward in recommendation system architecture. Our hybrid approach transforms user experiences by analysing each model's input, promising individualised recommendations that are relevant to individual tastes and needs.

In summary, our hybrid recommendation system offers a paradigm leap in recommendation technology, with improved performance, accuracy, and user happiness. By combining cutting-edge techniques from machine learning, natural language processing, and ensemble learning, we not only beat prior benchmarks but also lay the framework for future breakthroughs in recommendation systems. As the desire for personalized experiences grows, our technology serves as a beacon of innovation, set to revolutionize recommendation landscapes across several disciplines.

REFERENCES

[1] I. Balush, V. Vysotska, and S. Albota, "Recommendation system development based on intelligent search, NLP and machine learning methods," *CEUR Workshop Proc.*, 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 Comput. Appl.*, vol. 32, no. 7, pp. 2141–2164, Apr. 2020, doi: 10.1007/s00521-018-3891-5.

[3] A. Da'u and N. Salim, "Recommendation system based on deep learning methods: a systematic review and new directions," *Artif. Intell. Rev.*, vol. 53, no. 4, pp. 2709–2748, Apr. 2020, doi: 10.1007/s10462-019-09744-1.

[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)*, Sep. 2018, pp. 113–118. doi: 10.1109/ICCED.2018.00031.

[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, doi: 10.1016/j.procir.2019.04.126.

[6] Z. Zamanzadeh Darban and M. H. Valipour, "GHRS: Graph-based hybrid recommendation system with application to movie recommendation," *Expert Syst. Appl.*, vol. 200, p. 116850, Aug. 2022, doi: 10.1016/j.eswa.2022.116850.

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

[8] N. Nikzad-Khasmakhi, M. A. 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, doi: 10.1016/j.chaos.2021.111260.

[9] S. Renjith, A. Sreekumar, and M. Jathavedan, "An extensive study on the evolution of context-aware personalized travel recommender systems," *Inf. Process. Manag.*, vol. 57, no. 1, p. 102078, Jan. 2020, doi: 10.1016/j.ipm.2019.102078.

[10] Y. Deldjoo, T. Di Noia, and F. A. Merra, "A Survey on Adversarial Recommender Systems," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, Mar. 2022, doi: 10.1145/3439729.

[11] A. Marchand and P. Marx, "Automated Product Recommendations with Preference-Based Explanations," *J. Retail.*, vol. 96, no. 3, pp. 328–343, Sep. 2020, doi: 10.1016/j.jretai.2020.01.001.

[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 Trans. Big Data*, pp. 1–1, 2020, doi: 10.1109/TBDATA.2020.2973141.

[13] J. Yu, H. Yin, X. Xia, T. Chen, J. Li, and Z. Huang, "Self-Supervised Learning for Recommender Systems: A Survey," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 1, pp. 335–355, Jan. 2024, doi: 10.1109/TKDE.2023.3282907.

[14] "No Title", doi: 10.48550/arXiv.2307.02865.

[15] J. Ji *et al.*, "GenRec: Large Language Model for Generative Recommendation," 2024, pp. 494–502. doi: 10.1007/978-3-031-56063-7_42.

[16] S. Wang, L. Cao, Y. Wang, Q. Z. Sheng, M. A. Orgun, and D. Lian, "A Survey on Session-based Recommender Systems," *ACM Comput. Surv.*, vol. 54, no. 7, pp. 1–38, Sep. 2022, doi: 10.1145/3465401.

[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," *Mob. Networks Appl.*, vol. 25, no. 2, pp. 376–390, Apr. 2020, doi: 10.1007/s11036-019-01246-2.

[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 Syst. Appl.*, vol. 149, p. 113248, Jul. 2020, doi: 10.1016/j.eswa.2020.113248.

[19] S. Zhou *et al.*, "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*, Jul. 2020, pp. 179–188. doi: 10.1145/3397271.3401174.

[20] S. Dhawan, K. Singh, A. Batra, A. Choi, and E. Choi, "An Optimized Ranking Based Technique towards Conversational Recommendation Models," *Int. J. Comput. Digit. Syst.*, vol. 15, no. 1, pp. 1201–1216, Mar. 2024, doi: 10.12785/ijcds/150185.

[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," *Electron. Commer. Res. Appl.*, vol. 55, p. 101191, Sep. 2022,

doi: 10.1016/j.elerap.2022.101191.
[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 Process. Lett.*, vol. 55, no. 7, pp. 9409–9434, Dec. 2023, doi: 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," *J. Inst. Eng. Ser. B*, Feb. 2024, doi: 10.1007/s40031-024-00999-z.
[24] B. Ramzan *et al.*, "An Intelligent Data Analysis for Recommendation Systems Using Machine Learning," *Sci. Program.*, vol. 2019, pp. 1–20, Oct. 2019, doi: 10.1155/2019/5941096.
[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)*, Jul. 2017, pp. 504–509. doi: 10.1109/ICMLC.2017.8108968.
[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. 11036–11047, 2022, doi: 10.1109/ACCESS.2022.3141776.
[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. doi: 10.1007/978-3-030-04221-9_14.
[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, doi: 10.3390/s21165666.
[29] A. J. Costa, M. S. Santos, C. Soares, and P. H. Abreu, "Analysis of Imbalance Strategies Recommendation using a Meta-Learning Approach," no. September, 2020.
[30] E. Montañés, J. R. Quevedo, I. Díaz, and J. Ranilla, "Collaborative tag recommendation system based on logistic regression," *CEUR Workshop Proc.*, vol. 497, pp. 173–188, 2009.
[31] L. Shuxian and F. Sen, "Design and Implementation of Movie Recommendation System Based on Naive Bayes," *J. Phys. Conf. Ser.*, vol. 1345, no. 4, p. 042042, Nov. 2019, doi: 10.1088/1742-6596/1345/4/042042.
[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 Bilim. Derg.*, vol. 10, no. 3, pp. 858–869, Sep. 2021, doi: 10.17798/bitlisfen.939733.
[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)*, Sep. 2016, pp. 1293–1297. doi: 10.1109/ICACCI.2016.7732225.
[34] S. D. Jadhav and H. P. Channe, "Efficient Recommendation System Using Decision Tree Classifier and Collaborative Filtering," *Int. Res. J. Eng. Technol.*, 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. doi: 10.1007/978-981-16-1543-6_7.
[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)*, Jul. 2018, pp. 1–5. doi: 10.1109/ICMIC.2018.8529885.
[37] A. Dadhich and B. Thankachan, "Sentiment Analysis of Amazon Product Reviews Using Hybrid Rule-Based Approach," 2022, pp. 173–193. doi: 10.1007/978-981-16-2877-1_17.
[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*, Apr. 2001, pp. 285–295. doi: 10.1145/371920.372071.
[39] C. Ajaegbu, "An optimized item-based collaborative filtering algorithm," *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 12, pp. 10629–10636, Dec. 2021, doi: 10.1007/s12652-020-02876-1.
[40] S. Ko and J. Lee, "User preference mining through collaborative filtering and content based filtering in recommender system," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2455 LNCS, pp. 244–253, 2002, doi: 10.1007/3-540-45705-4_26.
[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)*, Jan. 2020, pp. 66–71. doi: 10.1109/ICISC47916.2020.9171222.

9. ABOUT AUTHORS

Dr. Kulvinder Singh received the M.Tech. degree in Computer Science Engineering, in 2001 and Ph.D. degree in Computer Science, in 2013 from Kurukshetra University Kurukshetra, Haryana India. He is presently working as a Faculty at the Department of Computer Science Engineering, U.I.E.T, Kurukshetra University Kurukshetra, Haryana India. He has already published quality research papers in the SCI, Scopus indexed, and other refereed journals and international conferences. His research interests include software testing, social networks and solid mechanics. He works on various software testing optimization techniques, social networks and mathematical modeling.

Dr. Sanjeev Dhawan received his postgraduate qualifications like the M.Sc. degree in Electronics, in 2000, the M.Tech. degree in Computer Science Engineering, in 2001, the M.C.A degree in Computer Science Applications, in 2005, and the Ph.D. in Computer Science, in 2011, all from the Kurukshetra University, Kurukshetra, Haryana, India. He is presently working as a Faculty of Computer Science Engineering at the University Institute of Engineering Technology (U.I.E.T), Kurukshetra University, Kurukshetra, Haryana, India. He has guided a large number of students in their Master's thesis work and Ph.D. work. He has already published several research papers in top quality journals/conferences of repute. His current research interests fall within the domain of

Computer Science Engineering. More specifically, he has focused and involved in Social networks and data analytics via Recommendation systems, Link prediction, Opinion mining, Sentiment analysis, Healthcare data analytics, Hidden relationship, Spam profiles detection, Communities detection, Web 3.0, Parallel computing, Programming languages, Business intelligence and their correlated areas.

Nisha Bali is currently a research Scholar pursuing her PhD. degree in Computer Science & Engineering at U.I.E.T, Kurukshetra University Kurukshetra, Haryana India. She has received her Master Degree in Computer Engineering from Department of Computer Science Engineering, U.I.E.T, Kurukshetra University Kurukshetra, Haryana, India. She is working on recommender system in social networking sites.

Dr. Anthony Choi received a B.S. in Electrical Engineering from George Washington University in 1991, and his M.S. and Ph.D. in Electrical Engineering from University of Florida in 1995 and 2002, respectively. He is presently a Professor and Director of Machine Intelligence Robotics Laboratory in the Department of Electrical and Computer Engineering at the Mercer University, Macon, GA. He was awarded an ASEE Summer Faculty Fellowship to work at Naval Surface Warfare Center, Panama City Division on Mine Counter Measures using Unmanned Underwater Vehicles. He has served on various boards including the executive board for Georgia Space Grant Consortium. He is also active in consulting. He is an owner/partner in several start-up companies.

Ethan Choi is currently a student working towards his bachelor's degree in Computer Engineering at Mercer University. He has received Mercer Undergraduate Research Scholar (MURS) for the past two summers, working on Natural Language Processing (NLP). He also served as president of the IEEE student chapter at Mercer University. During the summer, he went on a Mercer on Mission trip to South Africa to teach kids the necessary skills for programming in their future, so that they may have a chance to improve their situation and enter a field in which they love at the same time. His current work involves machine learning and NLP research.