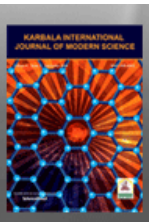

Karbala International Journal of Modern Science

Volume 12 | Issue 1 Article 2

Enhancing Recommendation Performance via Stacking Ensemble and Synthetic Data Augmentation

Zahraa Yaareb Hani
College of Computer Science and Information Technology, University of Kerbala, Karbala, Iraq

Mohsin Hasan Hussein
College of Computer Science and Information Technology, University of Kerbala, Karbala, Iraq.,
mohsin.h@uokerbala.edu.iq

Enhancing Recommendation Performance via Stacking Ensemble and Synthetic Data Augmentation

Abstract

Recommendation systems are essential tools that primarily aim to help users navigate through a large volume of information. They simplify the decision-making process by suggesting relevant items based on users' historical behaviour. However, their performance is often affected by common problems such as data sparsity. This work proposes a stacking-based ensemble recommendation system that integrates multiple machine learning models to enhance the model's predictive performance. A new synthetic data augmentation technique is introduced to address the sparsity issue in the user–item rating matrix. This method uses the Naïve Bayes algorithm to predict additional ratings for each user. These are then added to the original dataset to reduce the sparsity in the user–item rating matrix. Additionally, matrix factorisation models are included as base models to extract latent features. The results of the base models are fed as input to the stacking model to generate the final predictions. Experiments were performed on five benchmark datasets using MAE and RMSE as metrics to assess the performance. The results demonstrate that our proposed StackGBR-SDA model significantly improved the predictive performance compared to the individual models that made up the ensemble. Specifically, it achieved RMSE values of 0.1545, 0.2912, 0.8635, 0.4816, and 0.7163 on the datasets Amazon Food, Yelp, MovieLens100K, CiaoDVD, and FilmTrust, respectively. Moreover, our model outperformed methods from previous studies in terms of RMSE. These findings confirm the effectiveness of ensemble learning, as well as our data augmentation approach, in alleviating the sparsity problem and improving the recommendation performance.

Keywords

Boosting, Data Augmentation, Data Sparseness, Ensemble Learning, Machine Learning, Stacking

This research paper is available in Karbala International Journal of Modern Science: https://kijoms.uokerbala.edu.iq/home/vol12/iss1/2

RESEARCH PAPER

Enhancing Recommendation Performance via Stacking Ensemble and Synthetic Data Augmentation

Zahraa Y. Hani [a], Mohsin H. Hussein [a,b,*]

[a] College of Computer Science and Information Technology, University of Kerbala, Karbala, Iraq
[b] College of Science, University of Warith Al-Anbiyaa, Karbala, Iraq

Abstract

Recommendation systems are essential tools that primarily aim to help users navigate through a large volume of information. They simplify the decision-making process by suggesting relevant items based on users' historical behaviour. However, their performance is often affected by common problems such as data sparsity. This work proposes a stacking-based ensemble recommendation system that integrates multiple machine learning models to enhance the model's predictive performance. A new synthetic data augmentation technique is introduced to address the sparsity issue in the user–item rating matrix. This method uses the Naïve Bayes algorithm to predict additional ratings for each user. These are then added to the original dataset to reduce the sparsity in the user–item rating matrix. Additionally, matrix factorisation models are included as base models to extract latent features. The results of the base models are fed as input to the stacking model to generate the final predictions. Experiments were performed on five benchmark datasets using MAE and RMSE as metrics to assess the performance. The results demonstrate that our proposed StackGBR-SDA model significantly improved the predictive performance compared to the individual models that made up the ensemble. Specifically, it achieved RMSE values of 0.1545, 0.2912, 0.8635, 0.4816, and 0.7163 on the datasets Amazon Food, Yelp, MovieLens100K, CiaoDVD, and FilmTrust, respectively. Moreover, our model outperformed methods from previous studies in terms of RMSE. These findings confirm the effectiveness of ensemble learning, as well as our data augmentation approach, in alleviating the sparsity problem and improving the recommendation performance.

Keywords: Boosting, Data augmentation, Data sparseness, Ensemble learning, Machine learning, Stacking

1. Introduction

The growth of Internet use in recent years has led to a rapid increase in digital information. This has resulted in a problem known as information overload, where users are now facing challenges in finding items of interest amid the abundance of data [1,2]. Consequently, personalised recommendation systems (RSs) have been proposed. These tackle the problem by suggesting relevant items to users. They analyse users' historical data, such as interactions and feedback, to find similar patterns among items or with other users [1,3].

The primary goal of RSs is to suggest relevant items to users. Therefore, they need to collect information and build a profile for each user. This information can be collected either implicitly or explicitly. Systems can obtain implicit feedback indirectly by monitoring user behaviour through their interactions on a website. Examples of such interactions include website clicks, session duration, browsing behaviour, bookmarking a page, and adding items to the shopping cart [4,5]. Explicit feedback, on the other hand, can be collected directly by the users' numerical ratings or textual reviews. Some examples include users rating an

Received 14 July 2025; revised 22 October 2025; accepted 26 October 2025.
Available online 21 November 2025

* Corresponding author.
E-mail address: mohsin.h@uokerbala.edu.iq (M.H. Hussein).

https://doi.org/10.33640/2405-609X.3437

item, adding it to favourites, writing a review, and so on. Many RSs collect user preferences using a combination of explicit and implicit methods [2,4,5].

The demand for accurate RSs has grown significantly over the years, as consumers and businesses increasingly depend on them. RSs have been employed in various domains, such as e-commerce, social media, medical health, and more [1,6]. Among the different types of RSs, the most popular is collaborative filtering (CF). It examines user and item data to identify similarities and predict new items based on the closest neighbours of the target user [7,8]. CF is based on the assumption that users who liked similar things in the past will probably enjoy similar items in the future [6].

However, the algorithms of traditional methods are too simple and, therefore, incapable of capturing complex patterns in user behaviour. Some common issues of RSs include sparsity and cold start [2,7]. Data sparsity occurs due to the huge size of data and the lack of ratings in the user–item matrix, as users often fail to rate items they like or dislike. Thus, it becomes difficult for the model to extract connections or to find similarities between users/items in high sparsity datasets [2,9]. This results in an inaccurate set of similar neighbours, which can negatively affect the model's performance [9,10]. The cold start problem, on the other hand, arises when historical data about the target user is scarce or ratings for a target item are insufficient because it has been newly introduced to the system. Without any feedback data, the system is unlikely to recommend the item, which leads to less accurate predictions for new users or items [2,8,11].

Over the years, researchers have proposed numerous alternative methodologies to address the abovementioned issues. However, developing a robust RS is a complex task and requires combining various techniques. Therefore, researchers have resorted to new, more advanced approaches, such as ensemble learning [12]. Ensemble methods combine multiple models through various techniques, such as simple averaging, to obtain better predictive performance than a single model. These methods have proved highly effective in many RS studies [12,13]. Widely used ensemble methods include voting, bagging, boosting, and stacking [14].

In this paper, we propose a stacking-based ensemble framework to enhance the prediction performance. Additionally, an effective synthetic data augmentation approach is incorporated to handle sparse datasets when there are insufficient user ratings for the model to predict ratings accurately. The strength of the Naïve Bayes (NB) algorithm is utilised to find the top 10 items with the highest probability scores, and the newly generated ratings are added to the original datasets. Multiple experiments were conducted to find the optimal combination of the base and meta models for the stacking framework. The effectiveness of our model was evaluated using common performance measures such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The results demonstrate that the final stacking with Gradient Boosting Regressor (GBR) and data augmentation (StackGBR-SDA) outperformed the individual base models. In addition, the synthetic data augmentation has significantly improved the results by effectively addressing the data sparsity issue. Moreover, a comparative analysis with other literature was conducted. Compared with existing work, our model improved the RMSE performance by 2.37 % for MovieLens100K, 49.89 % for CiaoDVD, and 10.81 % for FilmTrust.

The remainder of this paper is structured as follows: Section 2 reviews related work on ensemble learning or hybrid approaches. Section 3 details the applied methodology, and Section 4 follows with the experimental study, which highlights the obtained results. Finally, Section 5 concludes the paper.

2. Related work

Every day, a vast amount of new content is added to online platforms such as e-commerce websites or music libraries. As a result, the cold start and sparsity problems remain frequently occurring issues. Many studies have proposed new approaches to overcome these problems and improve the model's performance.

For instance, Ref. [15] addressed the aforementioned issues by proposing a hybrid LX recommendation algorithm, which integrates Logistic Regression (LR) and Extreme Gradient Boosting (XGBoost). The latter is a tree model capable of capturing complex high-dimensional data, unlike LR, which struggles with non-linear features. The features were split into continuous and discrete ones. The former were used to train the XGBoost model. The output sparse matrix was combined with the XGBoost prediction scores, as well as the discrete features, and used as the input for the LR model to predict the users' music preferences.

Another solution to the frequent sparsity issue in RS is to leverage latent features. A common approach is matrix factorisation (MF). Major MF techniques in collaborative filtering-based RSs include SVD (Singular Value Decomposition) and NMF (Non-negative Matrix Factorisation).

Reference [16], for example, proposed an ensemble model that leveraged the strengths of two techniques: SVD and SVD++ (SVD with Implicit Feedback). SVD performs well with explicit feedback, while SVD++ extends SVD by adding implicit information. The predictions of both techniques were combined using simple averaging to produce the final results. The results showed that utilizing the complementary strengths of these models can significantly improve the model's performance.

Although MF methods are frequently used to handle sparsity issues, they can be computationally expensive and often require further improvements to boost their predictive performance. Reference [11] utilised an alternative matrix factorisation model called the Truncated-ULV Decomposition model (T-ULVD), a dimension reduction-based method. Additionally, the regularisation process was applied to reduce the semantic losses in the obtained low-dimensional vectors. The authors addressed the data sparsity and cold start problems by taking advantage of the strength of matrix factorisation.

Authors in Ref. [1] further improved traditional matrix factorisation approaches by proposing a Heterogeneous Information-Boosting model (HIBoosting). This model utilised both heterogeneous information and the latent relations in heterogeneous information networks (HINs). Meta-paths were used to define complex relationships from the first to the last object in a sequential path. Next, the random walk technique was employed to discover all possible path instances along the meta-paths. For each generated path instance, the similarity between objects of the same type was calculated. These similarity scores were used to create low-dimensional embedded features of users or items, using matrix factorisation. Finally, the feature vectors generated from different meta-paths were combined as the input feature for the XGBoost model.

In addition to numerical ratings, many studies have also incorporated textual reviews to extract user opinions and sentiments. For instance, Ref. [17] focused on customer reviews and ratings to predict user recommendations. The proposed ensemble approach consisted of two modules. The LSTM (Long Short-Term Memory) model performed sentiment analysis on the review text. The second model, on the other hand, analysed the ratings for different airline services. As the initial step, data preprocessing was performed, including removing missing values and punctuation. Furthermore, tokenisation was implemented to construct a dictionary of words. LSTM was used to perform sentiment analysis on the review texts. For the rating data, SVM (Support Vector Machine) was employed. The outputs of both models, LSTM and SVM, formed the input for the LR model to generate the model's final outputs.

Although many studies have incorporated sentiment analysis, further advancements have integrated emotions to capture deeper psychological context from users. For example, Ref. [18] developed a graph-based movie RS that leveraged sentiment, emotion information, and user ratings. The research states that sentiment and emotions are clearly distinguished by their data types. Sentiment is typically represented as ordinal data types, such as multiple levels of feelings (positive, neutral, negative). On the other hand, emotions are often described as categorical data types such as anger, sadness, fear, joy, and love. The study proposed a multi-relational GCMC-based stacking ensemble RS. The sentiment and emotion information were extracted using fine-tuned BERT models. Subsequently, the model constructed multi-relational graphs that integrated the ratings, sentiment, and emotion values. Finally, Graph Convolution Matrix Completion (GCMC) was employed to predict the user ratings. GCMC is a link prediction approach that transforms the user–item rating matrix into a bipartite graph in order to predict missing ratings using a graph auto-encoder.

With the rise of deep learning, multiple studies have proposed promising solutions to enhance recommendation models. Deep learning techniques can efficiently capture complex patterns through their multi-layer architecture and have therefore been applied across multiple applications [13]. They are considered a powerful tool for solving challenges such as sparsity and the cold-start problem [19]. For instance, autoencoders are often implemented as a solution to both cold start and sparsity. In general, they can be applied in two ways: to learn low-dimensional feature representations or to fill the blanks in the interaction matrix.

Reference [20] proposed a Graph-based Recommendation model (GHRS) incorporating a hybrid approach of content- and collaborative-based methods. It tackled the cold start problem by using autoencoders and integrating users' side information, such as gender and age. First, the authors constructed a graph with nodes representing the users. These nodes were connected to other similar users with common interests. For each user in the similarity graph, a set of information was extracted using multiple metrics such as PageRank, degree

centrality, and others. In the following step, the graph-based features were combined with the side information to create a unified feature matrix for all users. This matrix was then used as the input for the autoencoder to extract new features and for dimension reduction. Next, the K-means algorithm was used to group the users into clusters, utilising the latent features encoded by the autoencoder. The rating predictions were computed based on the cluster's average rating.

Furthermore, Ref. [21] introduced a new hybrid RS that combines content and behaviour-based data to improve accuracy. This approach effectively addressed the cold start and sparsity issues by incorporating content information, thereby reducing the need for historical data. The authors combined the MovieLens100K and MovieLens1M datasets with data from external sources such as Wikipedia to extract additional information, including movie characteristics and summaries. Next, Word2Vec was applied to produce dense feature embeddings. These were used as the input for the Convolutional Autoencoder-based Recommendation System (CAERS) model. At its core, this framework consists of an encoder and a decoder inherited from the autoencoder. Additionally, it leveraged Convolutional Neural Networks (CNNs) to extract meaningful patterns effectively. In particular, CAERS primarily used content data, including movie details, age, and sex. In contrast, Neural Collaborative Filtering (NCF) was used for the behavioural data, namely the ratings and other historical data. NCF is a hybrid framework that consists of a Generalised Matrix Factorisation (GMF) and a Multi-Layer Perceptron (MLP). This technique allows the framework to capture both linear and nonlinear user-item interaction patterns, thereby generating more accurate recommendations. In the final stage, the TriDeepRec hybrid model combined the predictions from both CAERS and NCF through an MLP model. By leveraging the strengths of both models, the model demonstrated a significant improvement in accuracy.

Reference [22] integrated multiple knowledge sources to address the issues of sparsity, scalability, and cold start. It proposed a deep learning and semantic fusion-based system called DLSF to generate top N recommendations. It consisted of three main modules. In the first module, the rating matrix was decomposed into smaller latent factors to deal with the sparsity problem. The learned user/item embeddings were provided as input to the deep feedforward neural network for collaborative filtering. The second module employed content-based filtering to address the cold start problem. It aimed to identify items with contents similar to those the user liked in the past. After preprocessing the text, a bag-of-words representation was generated and then converted into frequency vectors for similarity computation using cosine similarity. For extreme cold-start cases, module three provided popular items as recommendations, which were represented by the most liked or rated items in similar categories. Finally, the decision fusion model combined the recommendations from the previous three models to produce its output. The scores of each module were collected, normalised, and combined using a weighted average.

3. Methodology

In this section, our proposed method is described in detail. The framework consists of four main stages. They are preprocessing, synthetic data augmentation, training of diverse base models, and finally, training the meta learner through stacking. Our model aims to accurately predict ratings of unrated items based on the known user ratings. The model mainly relies on numeric rating information. However, a neutral rating of 3 out of 5 is often considered ambiguous as it does not clearly indicate whether the user would recommend the item. Therefore, sentiment analysis was performed on those ratings only to classify the review as either positive or negative. Next, data augmentation was conducted to solve the data sparsity issue using the Naïve Bayes classifier. Four different base models were examined, and the three best-performing ones were selected for the stacking framework. Furthermore, two different meta learners were tested to find the most effective ensemble strategy. Fig. 1 shows the steps of the proposed model.

3.1. Dataset description

Five baseline datasets available online were implemented in the evaluation step. They are Amazon Food,[1] Yelp,[2] Movielens100K,[3] CiaoDVD,[4] and FilmTrust.[4] Table 1 presents a summary description of each dataset. The Amazon Food dataset is a small subset of the Amazon dataset. It

[1] https://www.kaggle.com/datasets/aistct/amazonfood.
[2] https://www.kaggle.com/datasets/omkarsabnis/yelp-reviews-dataset/data.
[3] https://grouplens.org/datasets/movielens/.
[4] https://guoguibing.github.io/librec/datasets.html.

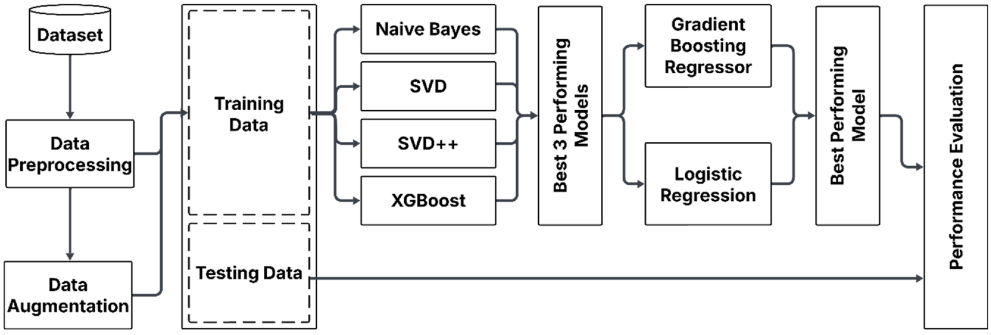

Fig. 1. Proposed model.

Table 1. Statistics of the datasets.

Dataset	#Users	#Items	#Ratings	Sparsity
Amazon Food	34,605	5086	41,710	99.98 %
Yelp	6403	4174	10,000	99.96 %
MovieLens100K	943	1682	100,000	93.70 %
CiaoDVD	17,615	16,121	72,665	99.97 %
FilmTrust	1508	2071	35,497	98.86 %

contains a wide range of food products, including details such as product name, user information, ratings, and plain text reviews. The Yelp dataset provides ratings and reviews for various businesses in different domains, including restaurants, hotels, education, and fashion. It is commonly used in multiple academic fields, such as RSs, NLP, and machine learning. MovieLens100K is a benchmark dataset widely used in many RS studies, which contains movie ratings and metadata. The CiaoDVD dataset contains user movie ratings, helpfulness, and trust relationships collected from the CiaoDVD website in 2013. FilmTrust is another popular dataset that contains movie ratings as well as social trust information. The ratings in all datasets are represented as standard star ratings on a scale of (1–5). FilmTrust is an exception as it follows a (0.5–4.0) rating scale. The text reviews are written in English. For the purpose of this analysis, we focused only on the users' movie ratings and textual reviews, if provided.

3.2. Data preprocessing

The preprocessing step was divided into two main phases: numerical ratings and textual reviews. Processing numerical ratings involved removing any rows with missing values. Additionally, categorical identifiers such as user ID and item ID were encoded into numerical formats using category-based encoding. This was necessary as most models require numerical inputs in order to construct the user–item rating matrix. Furthermore, the rating scale of the FilmTrust dataset was converted to match the standard (1–5) scale of the other datasets. The second phase involved preprocessing the textual reviews. This is an essential step in sentiment analysis as it prepares the data to be fed into sentiment classification models. The raw text data was converted into a structured and clean format for further processing, according to the following steps:

- Removal of hyperlinks (e.g., links starting with 'http' or 'www').
- Removal of email addresses to eliminate irrelevant metadata.
- Removal of HTML tags to retain plain text only.
- Removal of redundant whitespace (multiple whitespace, leading or trailing spaces)

3.3. Synthetic data augmentation

To alleviate the data sparsity problem, an augmentation technique was performed by synthetically generating new rating data for each user. First, the task was converted into a binary classification task. Ratings of 1 or 2 display strong dislike, while ratings of 4 or 5 show great interest. Hence, these ratings were converted to like/dislike without further analysis, where 1 and 2 were labelled as negative, 4 and 5 as positive.

Neutral ratings of 3 were classified based on the sentiment analysis results. Sentiment analysis is an area of Natural Language Processing (NLP) which focuses on classifying textual opinions, such as comments and reviews as positive, negative, or neutral [12,23]. This work employed the pre-trained bert-base-cased model from the HuggingFace Transformers library. As capitalisation can indicate meaningful semantic or emotional information, it is important to retain the reviews' original letter casing. The paper [24] affirms that using the BERT cased model has resulted in better results

compared to the uncased model. For instance, as mentioned by the authors, 'WONDERFUL' conveys much stronger sentiment than 'wonderful'. However, if no review text was available, they were considered positive. In the context of a standard 5-star rating system, a 3-star rating is generally interpreted as neutral or mildly positive, indicating the item or service was acceptable. In addition, the study [25] demonstrated that treating ratings of 3 as positive yielded better results. Based on this insight, our study considered neutral ratings as closer to positive sentiment.

The Naïve Bayes classifier was employed to compute the rating probability for all unrated user–item pairs. For each user, the top k items with the highest predicted probability were selected and assigned the user's mean rating. These newly generated ratings were then added to the original dataset to increase the density of the user–item rating matrix. This model was selected due to its probabilistic and simplistic nature. In addition, it is computationally efficient, which is an important factor as the model was trained on the entire dataset to generate the probability scores. It statistically identifies items a user is more likely to prefer based on the pattern of their existing ratings. Hence, Naïve Bayes is a suitable choice for this task compared to more complex methods, which would add unnecessary overhead without significantly improving the quality of the synthetic ratings. Moreover, several studies have shown that Naïve Bayes can achieve results competitive with other techniques [26].

The top-k items were assigned to the user's mean rating value in order to align the generated ratings with the user's preferences. Thus, the user-item matrix was densified without introducing extreme values. The primary goal was to mitigate sparsity without filling the dataset with too much synthetic information, which could potentially distort the dataset and affect performance. Table 2 compares the sparsity of each dataset before and after augmentation. Furthermore, experiments using k values of 5 and 10 were conducted. Our findings indicate that using $k = 5$ results in less significant improvement compared to $k = 10$. Increasing the value of k would potentially introduce the risk of bias and overfitting in the model. Overall, this strategy significantly improved the model's performance and proved to be an essential step in the preprocessing pipeline.

Table 2. Dataset sparsity before and after synthetic data augmentation.

Dataset	Sparsity before augmentation	Sparsity after augmentation
Amazon Food	99.98 %	99.81 %
Yelp	99.96 %	99.76 %
MovieLens100K	93.70 %	93.10 %
CiaoDVD	99.97 %	99.91 %
FilmTrust	98.86 %	98.38 %

3.4. Base models

This work applied four different base models – Naïve Bayes, SVD, SVD++, and XGBoost. It used the results of the three best-performing models as an input for the stacking framework. Our ensemble framework leveraged the strengths of a diverse set of algorithms, such as probabilistic reasoning, matrix factorisation, and gradient boosting. The following sections provide a detailed overview of each base model.

3.4.1. Naïve Bayes

Naïve Bayes is a fundamental supervised classification algorithm that is widely used in many machine learning and artificial intelligence tasks. Based on Bayes' Theorem, it computes the likelihood that a given sample belongs to each class, and selects the class with the highest probability as the predicted label [26,27]. In this work, Naïve Bayes is employed as a base model to predict user ratings. First, decimal ratings were converted to the closest integer value. Next, the features were encoded using a one-hot encoder. Both user ID and item ID are categorical values that cannot be directly used as input to the Naïve Bayes model. One-hot encoding can convert these values into binary vector representations. The produced high-dimensional, sparse feature matrix was then used to train the NB model. Despite its simplicity, this method served as an effective base learner for the ensemble framework, especially for denser datasets.

3.4.2. SVD & SVD++

Matrix factorisation is a popular method used to address sparsity in RSs. It can capture hidden patterns by extracting latent features from the user–item matrix. Well-known examples include SVD, SVD++, PMF (Probabilistic Matrix Factorisation), and NMF [6,9]. The first two are the most common. Notably, SVD was used in the winning solution of the Netflix Prize competition [28]. It is highly effective in extracting latent features in the underlying structure of the data, even when these features are not explicitly defined [29]. Moreover, SVD++ is an extension of the traditional SVD model, which combines both explicit and implicit feedback to enhance its performance [16]. In our framework, both methods were used to extract the latent features, and thus we profited from the

strengths of each model. This resulted in more robust and reliable recommendations. The models were trained on the training set using tuned hyperparameters to optimise performance. The predicted scores were then fed into our ensemble model.

3.4.3. XGBoost

Extreme Gradient Boosting is a popular supervised learning algorithm based on the Gradient Boosting Decision Tree (GBDT) framework. It is used in many ML domains, such as regression, classification, and ranking tasks, as it provides accurate and efficient results [1]. The fundamental idea of boosting is to iteratively train a set of weak classifiers so that each subsequent model corrects the errors of the previous one, thus resulting in a strong classifier [12]. In recent years, XGBoost has received much attention and has been applied in various studies and competitions, such as Kaggle [12,15]. Its main advantages are extracting high-level features, preventing overfitting, and effectively handling missing values [1,15]. Moreover, XGBoost incorporates regularisation methods and loss function optimisation to improve the model's generalisation ability [15].

In our ensemble framework, the XGBoost model was implemented as a base learner to capture non-linear interactions between users and items. The model was trained on the training set with tuned hyperparameters to optimise performance. Lastly, the model outputs were fed into the stacking framework to generate the final recommendation results.

3.5. Stacking ensemble

While traditional ML methods are popular, they often suffer from sparsity and other limitations that affect their efficiency. However, recent advances in computing power and machine learning techniques have motivated researchers to develop more enhanced solutions, such as ensemble learning [12]. The objective of ensemble learning is to leverage the strengths of multiple base learners to improve the performance of the model. Many studies have revealed that ensemble models generally perform better than individual base models. Well-known ensemble approaches include Voting, Bagging, Boosting, and Stacking [12,23].

Stacking is one of the most effective ensemble techniques. It aims to improve the model's generalisation ability. This approach generates the final recommendation result by first training each base model separately and then combining the results using a meta-model. This meta-model is trained to find the optimal combination of predictions from the base models [12,23]. The strength of stacking lies in combining multiple diverse algorithms to achieve better performance and accuracy than would be possible with a single model. Hence, this technique is popular in many machine learning competitions. The diversity of the base models is one of the main factors contributing to this method's success. For instance, the combination of matrix factorisation models (such as SVD) with tree-based models (e.g., XGBoost) can improve the predictive performance by leveraging their complementary strengths.

Despite these advantages, one drawback of stacking is that each base model must be trained on the entire dataset. This can lead to high computational costs, especially with large datasets [12]. Despite their limitations, stacked models have become a popular choice for many machine learning competitions, as they usually achieve high accuracy [12].

Simple linear classifiers, such as logistic regression, are often selected as meta-learners. However, in our work, LR was compared with a GBR to find the best ensemble strategy. GBR uses mean-squared error loss to minimise the loss function by training successive models on the residual errors of previous models. This enables gradient boosting to effectively learn complex patterns in the data. Additionally, early stopping was applied to the GBR model to halt the training once the validation error stopped improving, thereby preventing overfitting.

After training the base models individually, their prediction scores were used as the input for the stacking framework. Finally, the ensemble framework then generated the final predictions, which were evaluated using the MAE and RMSE metrics to measure the overall performance of the model. Based on the evaluation results, GBR outperformed LR and was therefore selected as the final meta-model.

4. Experimental study

This section describes how the performance of the proposed model was analysed and evaluated in two phases. In the first phase, the datasets were used after preprocessing, while in the second phase, the same steps were repeated on the augmented datasets. The results of both phases were compared in order to demonstrate the effectiveness of our augmentation technique. Notably, in

this study, the datasets are referred to as "original" before applying the synthetic data augmentation technique.

Otherwise, they are "augmented datasets". First, four different base models were trained, and the model with the lowest performance was excluded. Next, the results of the remaining three models were used as inputs for the ensemble framework. Furthermore, two different meta-learners were tested, namely LR and GBR, to find the most effective ensemble strategy. Five-fold cross-validation was employed during training to reduce the risk of overfitting and improve the overall generalisation of the model. The experiments were conducted using five real-world datasets, two of which contain textual reviews as well as numerical ratings. In addition, a comparative analysis with existing methods from other literature was performed to confirm the efficiency of our proposed method. Specifically, we compared our model with four previous studies that used the same datasets, based on the RMSE metric.

All experiments were conducted in a 64-bit Windows 11 Pro system with an 11th Gen. Intel® Core™ i7 2.30 GHz Processor and 16 GB RAM. For the BERT model implementation, Google Colab Pro was used to access a ready-to-use GPU environment. The training time for the augmented datasets is detailed in Table 3. It is important to note that the models were trained sequentially. However, the base models are independent of each other and could therefore be trained in parallel to reduce processing time in large datasets. Although the computational complexity of stacking is higher than that of the individual models, the notable performance gain proves its effectiveness. Furthermore, ensemble learning is an important approach and is frequently applied as a solution in various studies.

4.1. Evaluation metrics

Two common error metrics were used in this work to evaluate the quality of the predictions: MAE and RMSE are very popular choices in many RS studies. MAE is defined as the average of the absolute errors between predictions and actual values, which indicates how close predictions are to the true values. The RMSE metric computes the square root of the mean squared difference between predicted and actual values. For both evaluation measures, smaller values indicate higher accuracy. These metrics can be formulated as follows:

$$MAE = \frac{1}{n}\sum_{j=1}^{n}\left|p_{u,i} - r_{u,i}\right| \tag{1}$$

$$RMSE = \sqrt{\sum_{j=1}^{n}\frac{\left(p_{u,i} - r_{u,i}\right)^2}{n}} \tag{2}$$

Where n is the number of observations in the test set, $p_{u,i}$ denotes the predicted rating by the user u for item i, and $r_{u,i}$ denotes the actual rating of the item.

Table 3. Training time for each dataset.

Dataset	Training Time
Amazon Food	23 m 8.0 s
Yelp	4 m 6.1 s
MovieLens100K	49 m 51.0 s
CiaoDVD	36 m 37.4 s
FilmTrust	7 m 48.9 s

4.2. Hyperparameter settings and tuning

Hyperparameter tuning is an essential step in finding the optimal values to improve the model's performance, in terms of accuracy, generalisation, and overall effectiveness. This step was performed on the SVD, SVD++, and XGBoost algorithms. In our experiment, hyperparameter tuning was performed using the Grid Search approach. This technique exhaustively explores a list of given parameters to find the combination of values that yields the best results. For the SVD and SVD++ models, parameters such as the number of latent factors, the number of training epochs, the learning rate, and the regularisation strength were tuned. The search aimed to minimise both RMSE and MAE using a three-fold cross-validation. Similarly, Grid Search was performed to optimise the XGBoost model, with RMSE as the evaluation metric. The hyperparameter configurations included the tree depth, learning rate, number of trees, fraction of samples, and features to use per tree. The best hyperparameters obtained were then used to train the models. These hyperparameters are listed in Tables 4–8. Incorporating the hyperparameter tuning step in our ensemble framework significantly improved the results obtained.

4.3. Results and discussions

The objective of this work is to improve the performance of RSs through the use of ensemble learning and data augmentation. Extensive experiments were conducted to verify our model's

Table 4. List of selected hyperparameters on the Amazon Food dataset.

Model	Param	Orig. Val.	Aug. Val.
SVD	n_factors	150	50
	n_epochs	40	40
	lr_all	0.01	0.01
	reg_all	0.1	0.02
SVD++	n_factors	50	50
	n_epochs	40	40
	lr_all	0.01	0.01
	reg_all	0.1	0.02
XGBoost	colsample_bytree	1.0	0.8
	learning_rate	0.3	0.01
	max_depth	6	3
	n_estimators	100	50
	Subsample	1.0	0.8

Table 5. List of selected hyperparameters on the Yelp dataset.

Model	Param	Orig. Val.	Aug. Val.
SVD	n_factors	50	50
	n_epochs	30	40
	lr_all	0.005	0.01
	reg_all	0.1	0.02
SVD++	n_factors	50	50
	n_epochs	40	40
	lr_all	0.005	0.01
	reg_all	0.1	0.02
XGBoost	colsample_bytree	0.8	0.8
	learning_rate	0.01	0.01
	max_depth	5	5
	n_estimators	50	50
	Subsample	0.9	0.8

Table 6. List of selected hyperparameters on the MovieLens100K dataset.

Model	Param	Orig. Val.	Aug. Val.
SVD	n_factors	100	150
	n_epochs	40	40
	lr_all	0.01	0.01
	reg_all	0.1	0.1
SVD++	n_factors	100	100
	n_epochs	30	40
	lr_all	0.01	0.01
	reg_all	0.1	0.1
XGBoost	colsample_bytree	0.8	0.8
	learning_rate	0.2	0.2
	max_depth	7	7
	n_estimators	100	200
	Subsample	0.9	0.9

performance using several real-world datasets, and in two different scenarios.

Users who provide very few ratings are unsuitable for evaluation because they do not provide enough information to generate reliable personalised recommendations. Thus, a filtering step was applied to select users who had given five or more ratings, while those with fewer ratings were excluded. The filtered dataset was then divided into five folds for cross-validation. However, the excluded users and their ratings were added back

Table 7. List of selected hyperparameters on the CiaoDVD dataset.

Model	Param	Orig. Val.	Aug. Val.
SVD	n_factors	50	100
	n_epochs	30	40
	lr_all	0.01	0.01
	reg_all	0.1	0.02
SVD++	n_factors	50	100
	n_epochs	30	40
	lr_all	0.01	0.01
	reg_all	0.1	0.02
XGBoost	colsample_bytree	1.0	1.0
	learning_rate	0.3	0.3
	max_depth	6	6
	n_estimators	100	100
	Subsample	1.0	1.0

Table 8. List of selected hyperparameters on the FilmTrust dataset.

Model	Param	Orig. Val.	Aug. Val.
SVD	n_factors	150	150
	n_epochs	40	40
	lr_all	0.01	0.01
	reg_all	0.1	0.1
SVD++	n_factors	150	150
	n_epochs	40	40
	lr_all	0.01	0.01
	reg_all	0.1	0.1
XGBoost	colsample_bytree	1.0	1.0
	learning_rate	0.3	0.3
	max_depth	6	6
	n_estimators	100	100
	Subsample	1.0	1.0

and equally distributed across the training folds. This allows the model to learn additional patterns and improve its generalisation ability.

Next, four base models, Naïve Bayes, SVD, SVD++, and XGBoost, were trained, and the model with the highest RMSE value was excluded. The results of the remaining three models were then used as inputs for the ensemble framework. For the meta-model, two different methods were implemented, namely LR and GBR, to find the optimal stacking strategy. Although the individual base models are well established in existing literature, their integration with synthetic rating augmentation within the stacking framework produced consistent improvement across diverse datasets. The results show that our proposed StackGBR-SDA achieved superior results, which were further enhanced using the augmentation technique. In the following sections, the obtained results are described in detail.

4.3.1. Results of implementing the ensemble approach

In the first scenario, the base and meta models were trained on the original datasets. The results are illustrated in Tables 9–13. Naïve Bayes yielded

the highest RMSE value, therefore, it was excluded from the stacking ensemble on all datasets. Meanwhile, SVD and SVD++ achieved the best performance compared to the other base models in all datasets. This demonstrates their effectiveness in handling sparse data using the extracted latent features. For the meta-model, GBR consistently outperformed LR due to its ability to capture nonlinear patterns in the data, while LR assumes linear relationships, which can lead to underfitting. In addition, the stacking approach using GBR surpassed all base models in both MAE and RMSE.

Compared to the best-performing base model, it achieved improvements in RMSE of 11.69 %, 8.25 %, 2.13 %, 3.15 %, and 2.99 % on the datasets Amazon Food, Yelp, MovieLens100K, CiaoDVD, and FilmTrust, respectively.

4.3.2. Results of implementing ensemble learning with synthetic data augmentation approach

In the second scenario, the synthetic data augmentation approach was implemented on all datasets. The comparison between the original and augmented datasets in terms of sparsity is illustrated in Table 2. The base models were trained on these modified datasets and then fed into the stacking ensemble. Notably, this step significantly improved the results, which demonstrates its effectiveness in handling sparse datasets. Specifically, Naïve Bayes has achieved noticeable performance gain and even outperformed stronger base models. It can be noted that Naïve Bayes performs well on denser datasets. Moreover, it is based on the assumption that all input features are conditionally independent of each other. Therefore, it achieved improved probability scores on diverse and largely independent datasets such as Amazon Food and Yelp. FilmTrust and MovieLens100K, on the other hand, contain similar items, thus the improvement was relatively smaller. SVD and SVD++ consistently outperformed XGBoost. This proves their strength in extracting latent features to alleviate the sparsity issue. Naïve Bayes, SVD, and SVD++ were selected as the final base models for the datasets Amazon Food, Yelp, and CiaoDVD. Meanwhile, XGBoost, SVD, and SVD++ were selected for FilmTrust and MovieLens100K datasets.

Furthermore, stacking using GBR outperformed LR in this scenario as well, and for each base model in terms of RMSE. Compared to the best-performing base model, it achieved improvements in RMSE of 8.04 %, 6.70 %, 2.26 %, 4.27 %, and 3.53 % on the datasets Amazon Food, Yelp, MovieLens100K, CiaoDVD, and FilmTrust, respectively. Moreover, the stacking framework achieved superior results compared to the individual base

Table 9. Recommendation performance on the original Amazon Food dataset.

Model	MAE	RMSE
Naïve Bayes	0.6834	1.2548
SVD	0.748	0.9701
SVD++	0.7239	0.9618
XGBoost	0.8369	1.074
Stacking using LR	0.74	1.3257
Stacking using GBR	**0.6576**	**0.8494**

Bold values in the tables denote the overall best results obtained by our proposed model.

Table 10. Recommendation performance on the original Yelp dataset.

Model	MAE	RMSE
Naïve Bayes	0.8542	1.2246
SVD	0.8096	1.0321
SVD++	0.8124	1.0356
XGBoost	0.92	1.1661
Stacking using LR	0.7587	1.0845
Stacking using GBR	**0.7515**	**0.947**

Bold values in the tables denote the overall best results obtained by our proposed model.

Table 11. Recommendation performance on the original MovieLens100K dataset.

Model	MAE	RMSE
Naïve Bayes	0.7532	1.1037
SVD	0.7377	0.9366
SVD++	0.7218	0.9188
XGBoost	0.82	1.0239
Stacking using LR	0.8942	1.2199
Stacking using GBR	**0.7107**	**0.8992**

Bold values in the tables denote the overall best results obtained by our proposed model.

Table 12. Recommendation performance on the original CiaoDVD dataset.

Model	MAE	RMSE
Naïve Bayes	0.7267	1.1454
SVD	0.7442	0.9579
SVD++	0.7356	0.9527
XGBoost	0.8004	1.02
Stacking using LR	1.0348	1.5124
Stacking using GBR	**0.7186**	**0.9227**

Bold values in the tables denote the overall best results obtained by our proposed model.

Table 13. Recommendation performance on the original FilmTrust dataset.

Model	MAE	RMSE
Naïve Bayes	0.67	1.0307
SVD	0.6767	0.877
SVD++	0.6738	0.8759
XGBoost	0.7556	0.9691
Stacking using LR	0.7133	1.0234
Stacking using GBR	**0.6552**	**0.8497**

Bold values in the tables denote the overall best results obtained by our proposed model.

models, as shown in Tables 14–18. These findings highlight the strength of ensemble learning in enhancing the predictive performance of RSs. Evidently, the use of synthetic data augmentation led to substantial performance gain. This demonstrates its efficiency in mitigating the sparsity issue.

Overall, compared to the same approach without augmentation, the proposed StackGBR-SDA achieved RMSE performance gains of 81.81 %, 69.25 %, 3.97 %, 47.81 % and 15.70 % on the datasets Amazon Food, Yelp, MovieLens100K, CiaoDVD, and FilmTrust, respectively. Figs. 2–6 illustrate the RMSE performance for each base and meta model before and after augmentation. It can be observed that the gain is smaller for MovieLens100K. The reason for this is that this dataset is already relatively dense, so adding a few additional interactions had minimal impact on the model's prediction capability. In addition, the ratings of MovieLens100K and FilmTrust are concentrated around the middle (3–4), with fewer extreme values compared to the other datasets. This narrow distribution reduces variance and makes it difficult for the models to understand user preferences. In contrast, Amazon Food, Yelp, and CiaoDVD are very sparse datasets, and their rating distribution is spread over a broader range. This means the models can clearly distinguish between positive and negative preferences. For this reason, the augmented approach improved the results drastically.

Table 14. Recommendation performance on the augmented Amazon Food dataset.

Model	MAE	RMSE
Naïve Bayes	0.0136	0.168
SVD	0.1287	0.2325
SVD++	0.1181	0.2197
XGBoost	0.5191	0.6421
Stacking using LR	0.0136	0.168
Stacking using GBR	**0.0244**	**0.1545**

Bold values in the tables denote the overall best results obtained by our proposed model.

Table 15. Recommendation performance on the augmented Yelp dataset.

Model	MAE	RMSE
Naïve Bayes	0.0535	0.3121
SVD	0.1706	0.3451
SVD++	0.1529	0.3341
XGBoost	0.5958	0.7249
Stacking using LR	0.0535	0.3121
Stacking using GBR	**0.0763**	**0.2912**

Bold values in the tables denote the overall best results obtained by our proposed model.

Table 16. Recommendation performance on the augmented MovieLens100K dataset.

Model	MAE	RMSE
Naïve Bayes	0.6934	1.0543
SVD	0.6949	0.8997
SVD++	0.68	0.8835
XGBoost	0.7705	0.9879
Stacking using LR	0.8239	1.1691
Stacking using GBR	**0.664**	**0.8635**

Bold values in the tables denote the overall best results obtained by our proposed model.

Table 17. Recommendation performance on the augmented CiaoDVD dataset.

Model	MAE	RMSE
Naïve Bayes	0.173	0.5435
SVD	0.2778	0.5031
SVD++	0.3094	0.5117
XGBoost	0.5978	0.7278
Stacking using LR	0.1732	0.5438
Stacking using GBR	**0.1968**	**0.4816**

Bold values in the tables denote the overall best results obtained by our proposed model.

Table 18. Recommendation performance on the augmented FilmTrust dataset.

Model	MAE	RMSE
Naïve Bayes	0.4678	0.8488
SVD	0.5236	0.745
SVD++	0.518	0.7425
XGBoost	0.6204	0.8462
Stacking using LR	0.4872	0.8672
Stacking using GBR	**0.4819**	**0.7163**

Bold values in the tables denote the overall best results obtained by our proposed model.

Moreover, an additional experiment was performed, where the value of $k = 5$ was set in the data augmentation step. This means that the top five items were selected based on the probability scores and assigned to the mean rating value. The results are shown in Table 19. Our findings indicate that using $k = 5$ results in less significant improvement compared to $k = 10$. Increasing the value of k would likely introduce the risk of bias and overfitting in the model. Additionally, a higher value for k could fill the dataset with too many synthetic ratings, which could decrease the model's performance.

4.3.3. Comparison with other methods

Finally, our proposed method was compared with four previous studies [1,11,20,21], which all used the same datasets. RMSE was used as the evaluation metric, and our method achieved a significantly lower value than the previous studies. As shown in Table 20, our model outperformed the other

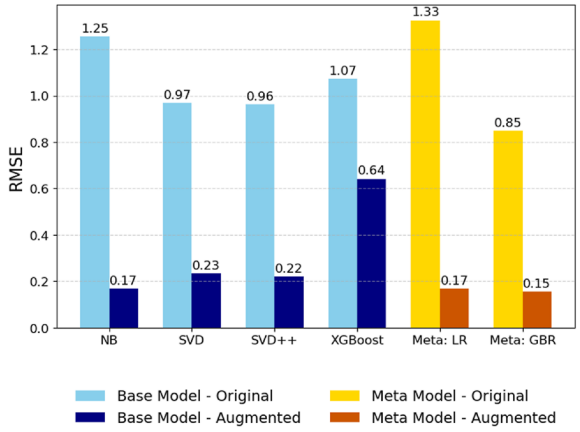

Fig. 2. RMSE comparison between original and augmented models on the Amazon Food dataset.

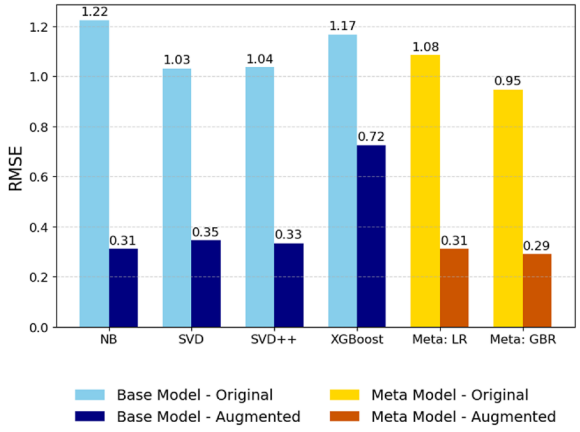

Fig. 3. RMSE comparison between original and augmented models on the Yelp dataset.

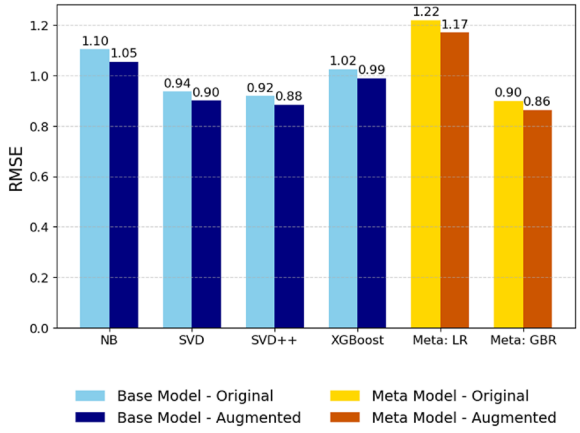

Fig. 4. RMSE comparison between original and augmented models on the MovieLens100K dataset.

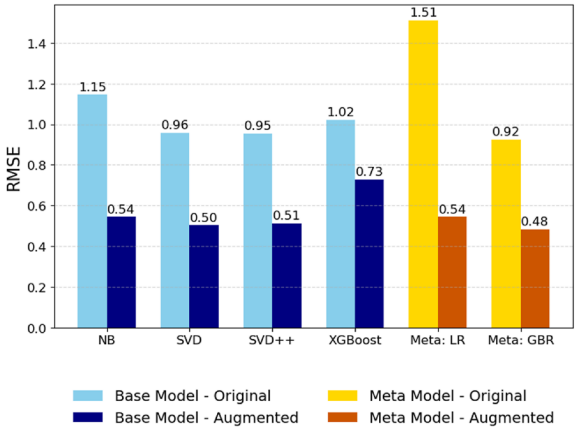

Fig. 5. RMSE comparison between original and augmented models on the CiaoDVD dataset.

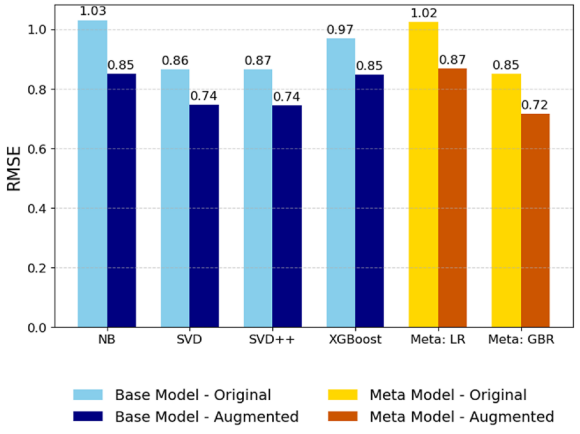

Fig. 6. RMSE comparison between original and augmented models on the FilmTrust dataset.

Table 19. Performance comparison for the StackGBR-SDA model with different k values.

Dataset	MAE (k = 5)	RMSE (k = 5)	MAE (k = 10)	RMSE (10 = 10)
Amazon Food	0.0440	0.2131	0.0244	0.1545
Yelp	0.1273	0.3867	0.0763	0.2912
MovieLens100K	0.6857	0.8790	0.664	0.8635
CiaoDVD	0.3102	0.6035	0.1968	0.4816
FilmTrust	0.5630	0.7796	0.4819	0.7163

Table 20. Recommendation performance of our approach compared to methods from previous studies.

Methodology	MovieLens100K	CiaoDVD	FilmTrust
HIBoosting	/	0.961	/
T-ULVD	0.8923	/	0.8031
GHRS	0.887	/	
TriDeepRec	0.8845	/	/
StackGBR-SDA	**0.8635**	**0.4816**	**0.6536**

Bold values in the tables denote the overall best results obtained by our proposed model.

methods and achieved RMSE improvements of 2.37 %, 49.89 %, and 10.81 % for the datasets MovieLens100K, CiaoDVD, and FilmTrust, respectively. Overall, the results confirm that our stacking framework can effectively enhance the predictions of RSs. In addition, the proposed synthetic data augmentation technique has significantly contributed to promoting performance by successfully alleviating the sparsity issue.

5. Conclusion

This study presents an RS based on stacking technology that combines multiple machine learning models to improve prediction performance. Furthermore, the sparsity issue was addressed in multiple ways. First, matrix factorisation models such as SVD and SVD++ were used to extract latent features. In addition, a synthetic data augmentation technique was proposed to generate 10 additional ratings for each user, based on the highest probability scores generated by Naïve Bayes. Overall, the results illustrated that this method was highly effective and contributed to a significant improvement in the model's performance. The meta-model took advantage of the GBR's ability to capture complex patterns, which further improved the performance. Several experiments were conducted using five benchmark datasets to evaluate the proposed model's efficiency. The results demonstrated that the proposed StackGBR-SDA consistently outperformed the individual base models. Additionally, the data augmentation technique significantly improved the overall performance. This paper confirms that ensemble learning can combine the strengths of multiple models to achieve superior performance.

While StackGBR-SDA shows substantial improvements in recommendation performance and efficiency, there are some limitations that are worth noting. The integration of multiple models can significantly increase computational complexity and resource demands. This increase in complexity can lead to scalability issues, specifically when dealing with large-scale datasets or in environments with limited computational resources.

Additionally, StackGBR-SDA focuses on sparsity only and does not address the cold-start problem, which is an important challenge in RSs. These limitations suggest directions for future research for ongoing improvements in model design, particularly in alleviating the cold-start problem by incorporating content-based features. This can help the model capture additional information to better understand users' preferences. Moreover, deep learning techniques could be utilized to further extend the model. For instance, they could be implemented as base models within the ensemble to extract more complex patterns in the data.

Ethical information

The datasets used are public, freely accessible online, and are dedicated to research studies. The manuscript does not contain any studies involving human participants or animals that require ethical approval.

Funding

No funds.

Conflicts of interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgements

The authors would like to thank the Presidency of the University of Kerbala for their Moral Support and for providing the laboratories to accomplish the practical part of the research.

References

[1] Y. Shao, C. Wang, HIBoosting: a recommender system based on a gradient boosting machine, IEEE Access 7 (2019) 171013–171022, https://doi.org/10.1109/access.2019.2956342.
[2] M.F. Aljunid, M. D.H., M.K. Hooshmand, W.A. Ali, A.M. Shetty, S.Q. Alzoubah, A collaborative filtering recommender systems: survey, Neurocomputing 617 (2025) 128718, https://doi.org/10.1016/j.neucom.2024.128718.
[3] A.A. Mohammed, M.M. Hamad, Recommender systems and machine learning techniques for large educational data: a survey, in: Proceedings of the 16th International Conference on Developments in eSystems Engineering (DeSE), IEEE, Istanbul, Turkiye. 2023, pp. 782–787, https://doi.org/10.1109/dese60595.2023.10469586.
[4] D. Roy, M. Dutta, A systematic review and research perspective on recommender systems, J. Big Data 9 (2022) 59, https://doi.org/10.1186/s40537-022-00592-5.
[5] T.M.A.U. Gunathilaka, P.D. Manage, J. Zhang, Y. Li, W. Kelly, Addressing sparse data challenges in recommendation systems: a systematic review of rating estimation using sparse rating data and profile enrichment techniques, Intell. Syst. Appl. 25 (2025) 200474, https://doi.org/10.1016/j.iswa.2024.200474.
[6] Z. Shahbazi, D. Hazra, S. Park, Y.C. Byun, Toward improving the prediction accuracy of product recommendation system using extreme gradient boosting and encoding approaches, Symmetry 12 (2020) 1566, https://doi.org/10.3390/sym12091566.
[7] K. Ji, Y. Yuan, R. Sun, K. Ma, Z. Chen, J. Liu, A bagging-based ensemble method for recommendations under uncertain rating data, in: Proceedings of the International Conference on Security, Pattern Analysis, and Cybernetics

(SPAC), IEEE, Jinan, China. 2018, pp. 446–450, https://doi.org/10.1109/spac46244.2018.8965431.
[8] F.T. Abdul Hussien, A.M.S. Rahma, H.B. Abdulwahab, An E-Commerce recommendation system based on dynamic analysis of customer behavior, Sustainability 13 (2021) 10786, https://doi.org/10.3390/su131910786.
[9] I. Karabila, N. Darraz, A. El-Ansari, N. Alami, M.E. Mallahi, A hybrid approach combining sentiment analysis and deep learning to mitigate data sparsity in recommender systems, Neurocomputing 636 (2025) 129886, https://doi.org/10.1016/j.neucom.2025.129886.
[10] M. Danlami, A.Y. Gital, K.M. Ibrahim, I.M. Lamir, M.A. Lawal, I.Z. Yakubu, Ensemble-based context-aware recommender system using clustering and singular value decomposition, in: Proceedings of the 3rd Asian Conference on Innovation in Technology (ASIANCON), IEEE, Ravet IN, India. 2023, pp. 1–14, https://doi.org/10.1109/ASIANCON58793.2023.10270346.
[11] F. Horasan, A.H. Yurttakal, S. Gündüz, A novel model based collaborative filtering recommender system via truncated ULV decomposition, J. King Saud Univ. - Comput. Inf. Sci. 35 (2023) 101724, https://doi.org/10.1016/j.jksuci.2023.101724.
[12] I.D. Mienye, Y. Sun, A survey of ensemble learning: concepts, algorithms, applications, and prospects, IEEE Access 10 (2022) 99129–99149, https://doi.org/10.1109/access.2022.3207287.
[13] M.A. Ganaie, M. Hu, A.K. Malik, M. Tanveer, P.N. Suganthan, Ensemble deep learning: a review, Eng. Appl. Artif. Intell. 115 (2022) 105151, https://doi.org/10.1016/j.engappai.2022.105151.
[14] X. Dong, Z. Yu, W. Cao, Y. Shi, Q. Ma, A survey on ensemble learning, Front. Comput. Sci. 14 (2020) 241–258, https://doi.org/10.1007/s11704-019-8208-z.
[15] H. Tian, H. Cai, J. Wen, S. Li, Y. Li, A music recommendation system based on logistic regression and eXtreme gradient boosting, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), IEEE, Budapest, Hungary. 2019, pp. 1–6, https://doi.org/10.1109/ijcnn.2019.8852094.
[16] Nitasha, A.K. Pandey, A. Yadav, S. Mishra, Medhanshi, Enhancing movie recommendation efficiency using ensemble learning techniques, in: Proceedings of the International Conference on Emerging Innovations and Advanced Computing (INNOCOMP), IEEE, Sonipat, India. 2024, pp. 539–544, https://doi.org/10.1109/innocomp63224.2024.00094.
[17] P.K. Jain, G. Srivastava, J.C.-W. Lin, R. Pamula, Unscrambling customer recommendations: a novel LSTM ensemble approach in airline recommendation prediction using online reviews, IEEE Trans. Comput. Soc. Syst. 9 (2022) 1777–1784, https://doi.org/10.1109/tcss.2022.3200890.
[18] C. Lee, D. Han, S. Choi, K. Han, M. Yi, Multi-relational stacking ensemble recommender system using cinematic experience, in: Proceedings of the IEEE International Conference on Big Data and Smart Computing (BigComp), IEEE, Daegu, Korea. 2022, pp. 300–303, https://doi.org/10.1109/bigcomp54360.2022.00064.
[19] H. Ko, S. Lee, Y. Park, A. Choi, A survey of recommendation systems: recommendation models, techniques, and application fields, Electronics 11 (2022) 141, https://doi.org/10.3390/electronics11010141.
[20] Z. Zamanzadeh Darban, M.H. Valipour, GHRS: graph-based hybrid recommendation system with application to movie recommendation, Expert Syst. Appl. 200 (2022) 116850, https://doi.org/10.1016/j.eswa.2022.116850.
[21] A. Ghadami, T. Tran, TriDeepRec: a hybrid deep learning approach to content- and behavior-based recommendation systems, User Model. User-Adapt. Interact 34 (2024) 2085–2114, https://doi.org/10.1007/s11257-024-09418-w.
[22] V. Bhatia, DLSF: deep learning and semantic fusion based recommendation system, Expert Syst. Appl. 250 (2024) 123900, https://doi.org/10.1016/j.eswa.2024.123900.
[23] A.A. Khan, O. Chaudhari, R. Chandra, A review of ensemble learning and data augmentation models for class imbalanced problems: combination, implementation and evaluation, Expert Syst. Appl. 244 (2024) 122778, https://doi.org/10.1016/j.eswa.2023.122778.
[24] B. Ray, A. Garain, R. Sarkar, An ensemble-based hotel recommender system using sentiment analysis and aspect categorization of hotel reviews, Appl. Soft Comput. 98 (2021) 106935, https://doi.org/10.1016/j.asoc.2020.106935.
[25] A.M.A. Al-Sabaawi, M.H. Hussein, M. Dalli, Classifying items with the rating values 3 using text reviews to improve the recommendation accuracy in the collaborative filtering approach, Karbala Int. J. Mod. Sci. 11 (2025) 144–153, https://doi.org/10.33640/2405-609x.3393.
[26] P. Valdiviezo-Diaz, F. Ortega, E. Cobos, R. Lara-Cabrera, A collaborative filtering approach based on Naïve Bayes classifier, IEEE Access 7 (2019) 108581–108592, https://doi.org/10.1109/access.2019.2933048.
[27] A.A. Neamah, A.S. El-Ameer, Design and evaluation of a course recommender system using content-based approach, in: Proceedings of the International Conference on Advanced Science and Engineering (ICOASE), IEEE, Duhok. 2018, pp. 1–6, https://doi.org/10.1109/icoase.2018.8548789.
[28] K. R, P. Kumar, B. Bhasker, DNNRec: a novel deep learning based hybrid recommender system, Expert Syst. Appl. 144 (2020) 113054, https://doi.org/10.1016/j.eswa.2019.113054.
[29] S. Labde, V. Karan, S. Shah, D. Krishnan, Movie recommendation system using RNN and cognitive thinking, in: Proceedings of the 4th Int. Conf. Emerg. Technol., INCET, IEEE, Belgaum, India. 2023, pp. 1–7, https://doi.org/10.1109/incet57972.2023.10170572.