International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/IJSRCSEIT

The Book Forum : Application System with Hybrid Filtering and Recommendation using Collaborative Filtering and Autoencoders

Ayushi Chawade, Kalpesh Khairnar, Vaishnavi Khamkar, Pratham Sonawane, Monali Bansode
Information Technology, International Institute of Information Technology Pune, Maharashtra, India

ARTICLEINFO

Article History:

Accepted: 10 May 2023
Published: 29 May 2023

Publication Issue
Volume 9, Issue 3
May-June-2023

Page Number
286-299

ABSTRACT

The digital era has transformed the way we discover and engage with books. However, the abundance of available options often makes it challenging for users to find books tailored to their individual preferences. To address this issue, the development of Android applications incorporating advanced recommendation techniques has gained significant attention. This review paper explores the concept of using autoencoders with collaborative filtering and hybrid filtering algorithms as the backbone for book recommendation systems within Android applications. It leverages the power of autoencoders, a type of neural network, to generate personalized book recommendations based on user ratings, reading history, and behaviors. Collaborative filtering techniques analyze user interactions to identify patterns and similarities, while hybrid filtering combines multiple recommendation models to provide accurate and diverse suggestions. Additionally, the application offers features such as popular book recommendations and genre-based suggestions to cater to a wider range of user preferences. The application for book recommendations integrates advanced recommendation techniques using autoencoders and collaborative filtering. In addition to personalized recommendations, it offers a range of functionalities. Users can create wishlists, bookmark books, and even upload their own books as authors. An admin portal ensures efficient management and moderation of uploaded books. The application also includes a forum for user discussions about specific chapters or books. Overall, this comprehensive solution enhances user satisfaction and engagement, providing a seamless reading experience. This review paper highlights the innovative idea and its implications for the book industry and digital reading community.

Keywords : Book Recommendation, Collaborative Filtering, Deep Neural Network Architecture, Context Aware Component

I. INTRODUCTION

A Recommend system is a type of information filtering system that provides personalized recommendations to users based on their preferences and past behaviour. Recommend systems have become increasingly popular in recent years due to the vast amount of data generated by users and the need for more efficient ways to filter through this data and provide personalized recommendations. It is mainly used in e-commerce websites, streaming platforms, social media applications, and online marketplaces. Systems have been developed to utilize various recommendation algorithms such as collaborative filtering, content-based filtering, and hybrid approaches to improve the accuracy of recommendations.

Most recommenders use collaborative and content-based filtering, but hybrid approaches that combine both techniques have been shown to provide more accurate recommendations. Collaborative filtering uses user behaviour data, such as ratings and purchase history, to recommend items that other users with similar preferences have liked or purchased. Collaborative filtering can be further divided into user-based and item-based approaches, with the former focusing on finding users similar to the target user and the latter focusing on finding items similar to those previously liked or purchased by the target user. These similar items are recommended based on their similarity scores, ensuring that the suggested items are relevant to the user's interests. Content-based filtering, on the other hand, utilizes item features such as genre or category to recommend items that are similar to those the user has interacted with in the past. It utilizes natural language processing and machine learning techniques to analyse the content of the items that users have interacted with to recommend similar items to users. Hybrid approaches combine collaborative and content-based filtering techniques to leverage the strengths of both methods and provide more accurate recommendations. By implementing a hybrid approach, recommender systems can overcome some of the limitations and challenges faced by solely using either collaborative or content-based filtering, such as addressing data sparsity issues, overcoming the cold-start problem, and mitigating the potential for filter bubbles.

of their interests. Incorporating these advanced algorithms allows for more advanced analysis and can The neural network framework is used for developing advanced recommender systems that can adapt and learn from complex patterns in user behaviour and item features, resulting in more accurate and personalized recommendations. One of the key advantages of using such an approach is its ability to continually improve over time as it ingests more data, thereby enhancing user satisfaction and ultimately increasing the likelihood of user engagement and retention. Implementing a neural network in a recommendation system can involve various aspects, such as data pre-processing, model selection and configuration, training process optimization, and evaluation metrics definition. Additionally, the incorporation of various deep learning techniques such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) can further enhance the system's ability to capture intricate patterns and temporal dependencies, leading to a more sophisticated and effective recommendation experience.

A book recommendation system using a hybrid approach can improve the accuracy and diversity of recommendations provided to users. This is achieved by combining content-based filtering, collaborative filtering, and other algorithms to generate personalized reading suggestions tailored to individual preferences. Consequently, this fosters increased engagement with the platform and enhances users' overall reading experience. Applying neural networks with that, such as deep learning techniques, enables the system to capture intricate patterns in users' reading habits and preferences, resulting in a more sophisticated understanding lead to even more accurate recommendations.

II. Literature survey

TABLE I. COMPARITIVE ANALYSIS ON BOOK RECOMMENDATION SYSTEM

Title	Algorithm	Result	Drawbacks
The design of disciplinary book recommendation system based on android: a view of extra-curricular activities	DAO(Data Access Object) E-R mapping	Searched books will be presented on main interface of application from the databse.	Recommends based on basic user information.
Book Recommendation Using Machine Learning Methods Based on Library Loan Records and Bibliographic Information	Machine learning modules based on Support vector machine (SVM), Random Forest, and Adaboost	Data displayed with detail records of students and books that have been previously borrowed from the library	Recommendation is performed only on loan based records and bibliographic information
Research on Personalized Book Recommendation Model for New Readers	Cosine similarity, Euclidean similarity, Jaccard similarity	At specific conditions significant algorithms are used and managed to get desired outcome.	When number of neighbour trend to increase, simultaneously the recommend effect achieves the best effect.
Embedding Model Design for Producing Book Recommendation	Embeding model and, principal component analysis	recommendation is produced by training embedding model to learn the pattern of high-rated books from every user and calculate the preferred books as close as possible	Total accuracy of embedding model is 59%

The growing popularity of online book platforms and the vast number of available titles have created a need for effective book recommendation systems using various techniques to filter and suggest relevant content to users based on their preferences and reading habits. A literature survey on book recommendation systems can help identify the current state-of-the-art techniques and their effectiveness.

Data Access Object, ER mapping Machine Learning modules such as Support Vector Machine(SVM), Random Forest, and AdaBoost have been widely used in developing these systems with outcomes of significant improvements in the accuracy and effectiveness of data prediction and user satisfaction. In addition, Personalized Recommender and Embedding Models have been designed lately to further enhance the accuracy of book recommendation systems by taking into account more personalized factors such as reading history, social media activity, and high-rated user-generated content like reviews and ratings.

Eventually, these systems have some flaws, such as the cold-start problem and vulnerability to fake reviews or rating manipulation, fixed data, and matrix sparsity, making it crucial to explore new methodologies and refine existing algorithms for better user experience and more reliable recommendations. This necessitates further research and development to address these issues and improve system performance

TABLE II. COMPARATIVE ANALYSIS ON CONTENT BASED AND COLLABORATIVE FILTERING ALGORITHMS

Title	Algorithm	Result	Drawbacks
Book Recommendation for eLearning Using Collaborative Filtering and Sequential Pattern Mining	Collaborative Filtering & Sequential Data	Provides more accurate recommendations with less bias	Has cold-start and sparsity problem
A Deep Learning Based Collaborative Neural Network Framework for Recommendation System using matrix function	Collaborative filtering and matrix factorization	Recommendation of books after searching for keyword.	Correlation between several users is much larger than the number of user who express similar

			interest of items. Cold start problem.
Cloud-Based Collaborative Filtering Algorithm For Library Book Recommendation System	collaborative filtering algorithm where books are based on categories and Apriori Algorithm	An automated and dynamic library recommendation system will help the user to choose the best version of the book of his/ her interest within a few seconds depending on the ratings given to that book.	The system needs an active internet connection all the time while accessing.
Book Recommendation system based on Collaborative Filtering and Association Rule Mining for College Students	User-based collaborative filtering and association rule mining.	Book recommendation system which recommends books to users according to their price range and preferred publishers.	The recommendation system is basically used only for students which will recommend textbooks.

Recommendation System mostly uses collaborative filtering, content-based filtering, and hybrid approaches to provide personalized book suggestions to users based on their preferences and reading habits. Extensive research has been conducted on various algorithms and techniques, aiming to improve the accuracy and user satisfaction of these systems. Some of the widely used methods include matrix factorization, deep learning, and natural language processing techniques to analyze user-item interactions and content information, contributing to the improvement of recommendation quality. One of the challenges faced by these systems is addressing the cold-start problem, correlation, and data sparsity, which occurs when limited data is available for new users or items. Some systems need an active internet connection, while others are capable of providing offline recommendations based on preloaded data.

Table III.

Title	Algorithm	Result	Drawbacks
Book Recommendation Model Based on Wide and Deep Model	Wide and Deep model	Improved the Wide & Deep model by converting the double-label into multiple labels. The experimental results show that the accuracy of our book recommendation model is significantly better than traditional recommendation algorithms and hybrid recommendation algorithms.	Recommendation based on data of books with only categories attribute.
A Deep Learning Based Collaborative Neural Network Framework for Recommender System	Collaborative Filtering Deep Recommender architecture (CFDR) Collaborative Filtering based Multistage Deep Neural Network architecture (CFMDNN)	Improved performance than traditional collaborative filtering model	Cold start problem. Matrix sparsity problem.
Cloud-Based Collaborative	collaborative filtering	An automated	The system needs an

Filtering Algorithm For Library Book Recommendation System	algorithm where books are based on categories and Apriori Algorithm	and dynamic library recommendation system will help the user to choose the best version of the book of his/ her interest within a few seconds depending on the ratings given to that book.	active internet connection all the time while accessing.
Book Recommendation system based on Collaborative Filtering and Association Rule Mining for College Students	User-based collaborative filtering and association rule mining.	Book recommendation system which recommends books to users according to their price range and preferred publishers.	The recommendation system is basically used only for students which will recommend textbooks.

A Deep Learning Based Collaborative Neural Network recommendation system that uses a neural network to make recommendations based on user behavior and preferences. It involves collaborative filtering, which looks at similarities between users to make recommendations. Whereas, "Cloud-Based Collaborative Filtering Algorithm recommendation system uses a cloud-based algorithm to make recommendations for books in a library. It also uses collaborative filtering to look at similarities between users to make recommendations. Moreover, Association Rule uses both collaborative filtering and association rule mining to make recommendations for college students. It looks at similarities between users and also analyzes their past behaviors to make recommendations.

III. METHODOLOGY

A. Content-based Filtering

Content-based filtering is a type of recommendation system that suggests items to users based on their preferences and past behavior. The approach relies on the similarities between the items themselves, rather than on the preferences of other users. In this approach, items are described by a set of features, such as genre, author, actors, or keywords, and users are matched with items that have similar features to those that they have previously enjoyed.

The process of content-based filtering usually involves the following steps:

- Feature extraction: The system identifies the key features of each item in the dataset. This could include text-based features, such as the title, author, and summary, or metadata, such as the genre, release date, or ratings.
- User profile creation: The system builds a profile of the user based on their previous interactions with the system, such as items they have rated or viewed. This profile is created by analyzing the features of the items that the user has interacted with.
- Similarity calculation: The system calculates the similarity between the user's profile and the features of the remaining items in the dataset. This can be done using techniques such as cosine similarity, Euclidean distance, or Pearson correlation coefficient.
- Recommendation generation: The system recommends the items that are most similar to the user's profile, based on the similarity scores calculated in step 3.

Content-based filtering has several advantages over other recommendation techniques, such as collaborative filtering. One of the main benefits is that it does not require large amounts of data on user behavior, which can be difficult to obtain in some cases. Additionally, content-based filtering can provide more personalized recommendations, as it takes into account the specific preferences of each user. However, a drawback of this approach is that it can suffer from the "filter bubble" problem, where users are only recommended items that are similar to those they have already interacted with, and are not exposed to new or unexpected content.

B. Collaborative Filtering

Collaborative filtering is a type of recommendation algorithm that uses the behavior and preferences of similar users to make personalized recommendations for an individual user. It assumes that users who have similar preferences in the past are likely to have similar preferences in the future, and that items that are popular among similar users are likely to be of interest to the user in question.

Collaborative filtering can be either user-based or item-based. User-based collaborative filtering looks for users with similar preferences and recommends items that those users have liked. Item-based collaborative filtering looks at the similarities between items and recommends items that are similar to ones the user has liked in the past.

Collaborative filtering is widely used in recommendation systems for a variety of applications, including e-commerce, music and movie recommendations, and news recommendations.

Collaborative filtering algorithms typically take into consideration two main parameters:

- User behavior: This includes information about what items a user has interacted with in the past, such as movies watched, books read, products purchased, etc. This information is used to build a profile of the user's preferences and behavior.
- Item features: This includes information about the items being recommended, such as genre, category, author, price, etc.

This information is used to find similarities between items and to make recommendations based on those similarities.

In addition to these two main parameters, other factors can also be taken into consideration depending on the specific application and the available data. For example, demographic information about the user (age, gender, location, etc.) or contextual information (time of day, day of the week, weather, etc.) can be used to further refine the recommendations.

It's important to note that the parameters used in collaborative filtering algorithms can vary depending on the specific implementation and the available data. The goal is to use as much relevant information as possible to make the most accurate and personalized recommendations for the user.

C. Hybrid Recommendation

A hybrid recommendation system is a type of recommendation system that combines two or more different recommendation techniques to make recommendations. The goal of a hybrid system is to leverage the strengths of each individual technique to improve the overall quality of the recommendations and address some of the limitations of each technique.

There are several ways in which hybrid recommendation systems can be constructed, including:

- Weighted hybrid: This approach combines the recommendations generated by multiple algorithms by giving each algorithm a weight that reflects its performance on a specific task. The final recommendation is then calculated as a weighted average of the recommendations generated by each algorithm.
- Switching hybrid: This approach uses multiple algorithms and switches between them depending on the context or the user's behavior. For example, one algorithm might be used to generate recommendations for a new user who has no historical data, while another algorithm might be used for a user who has a lot of historical data.
- Feature combination hybrid: This approach combines the features used by multiple algorithms to generate a single recommendation. For

example, one algorithm might use collaborative filtering, while another algorithm might use content-based filtering. By combining the features used by each algorithm, the hybrid system can generate more accurate recommendations.

Hybrid recommendation systems are widely used in practice and have been shown to improve the accuracy and relevance of recommendations in many applications, such as e-commerce, movie and music recommendations, and news recommendations.

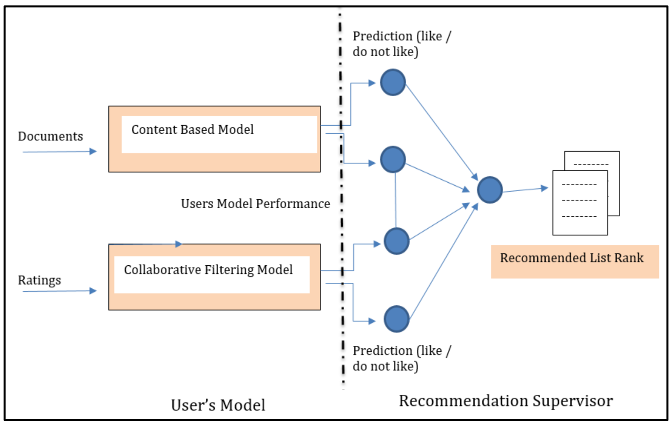

Prediction (like / do not like)
Documents
Content Based Model
Users Model Performance
Ratings
Collaborative Filtering Model
Recommended List Rank
Prediction (like / do not like)
User's Model
Recommendation Supervisor

Figure 1. Working of Hybrid Recommendation System

Neural Networks

Neural networks are a type of machine learning algorithm inspired by the structure and function of the human brain. They consist of interconnected nodes (or neurons) organized into layers that process information in a feedforward or feedback manner.

In a typical feedforward neural network, information is processed through the network in a single direction, from the input layer to the output layer. Each node in the network receives input from the previous layer, processes that input using a mathematical function, and then passes the output to the next layer.

The strength of neural networks lies in their ability to learn complex patterns and relationships in data, which makes them well-suited for tasks such as image recognition, natural language processing, and speech recognition. Neural networks can also be used in combination with other machine learning techniques, such as reinforcement learning, to solve more complex problems.

There are many different types of neural networks, each with its own specific architecture and learning algorithm. Some common types of neural networks include feedforward neural networks, convolutional neural networks, recurrent neural networks, and deep neural networks.

Autoencoders

An autoencoder is an artificial neural network that is trained to encode input data into a lower-dimensional representation, and then decode the representation back into the original data. The encoding and decoding operations are performed by hidden layers of the network, which are trained to learn a compressed representation of the input data that captures its most salient features. The network is trained to minimize the difference between the input and output data, typically using a loss function such as mean squared error (MSE). Once trained, the autoencoder can be used to generate new data that is similar to the input data by sampling from the learned compressed representation. Autoencoders have many applications, including data compression, anomaly detection, and image and signal denoising.

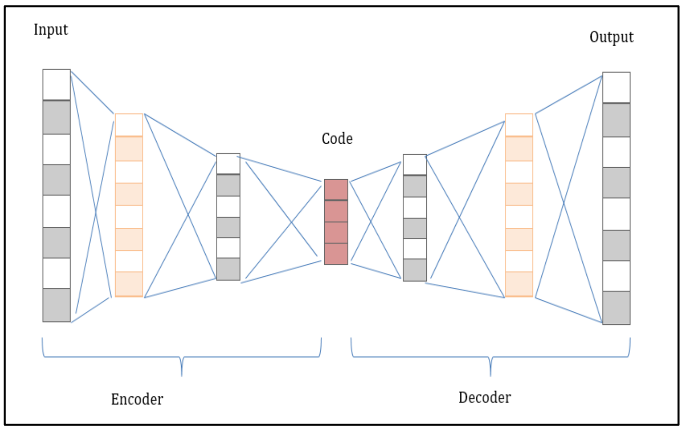


Figure 2. Functionality of Autoencoders

Autoencoders with collaborative filtering is a popular approach for building recommendation systems.

Collaborative filtering is a technique that relies on the patterns of user interactions with items to recommend items that a user may be interested in. An autoencoder is a type of neural network that can learn to represent data in a lower-dimensional space, while still preserving its essential features.In this approach, the autoencoder is trained on a dataset of user-item interactions, such as ratings or clicks. The input to the autoencoder is a sparse matrix that represents the user-item interactions, where each row corresponds to a user and each column corresponds to an item. The autoencoder is trained to reconstruct this input matrix, while also learning a lower-dimensional representation of the user and item features.Once the autoencoder is trained, the lower-dimensional representations of users and items can be used to make recommendations. For example, given a user, the system can recommend items that are similar to the items the user has interacted with in the past. The similarity between items is calculated using the lower-dimensional representations learned by the autoencoder. This approach has the advantage of being able to handle sparse datasets, as it can learn a dense representation of the user-item interactions. It also does not require explicit feature engineering, as the autoencoder learns the relevant features from the data. However, it can be computationally expensive to train the autoencoder on large datasets.

DATA PREPARATION

Initially, dataset included two separate data frames one with users ratings and reviewed number of books and another with details of books containing information of about 32,000 books and 2,000 users respectively.

For a book recommendation system, the data preparation involves processing two main data frames: one containing the user ratings and another containing the book details.

The ratings data frame typically includes columns such as user ID, book ID, and rating. Before using this data for building a recommendation system, it is important to preprocess it. This includes tasks such as handling missing values, removing duplicates, and normalizing the ratings. Additionally, the data can be split into training, validation, and test sets to evaluate the performance of the model.

The book details data frame contains information such as the book title, author, genre, and description. This data frame can be enriched by adding additional features such as the publication year, the number of pages, and the publisher. Before using this data for building a recommendation system, it is important to preprocess it as well. This includes tasks such as handling missing values, removing duplicates, and encoding categorical features. Additionally, the data can be combined with external data sources such as book reviews or ratings from other platforms to improve the quality of the recommendations.

- Data Cleaning: In this step, you need to check for missing values, outliers, duplicates, and inconsistent data. Data Integration: In this step, you need to combine different datasets to create a single data source for your recommendation system.
- Data Transformation: In this step, you need to convert your data into a format that is suitable for machine learning algorithms.
- Feature Engineering: In this step, you need to select the most relevant features that can help your recommendation system make accurate predictions.
- Splitting Data: In this step, you need to divide your data into training and testing datasets. The training dataset is used to train your recommendation system, while the testing dataset is used to evaluate its performance.
- Data Preprocessing: In this step, you need to preprocess the data to make it suitable for training your machine learning model.

After data preparation, autoencoders with collaborative filtering can perform well in building a recommendation system. With collaborative filtering, the system can recommend items to users based on their past behavior, which is a powerful approach that can help users discover new items they may like. Autoencoders can help to encode user and item data and decode it to make predictions for missing ratings. The model is trained on the ratings data and optimized using a loss function such as RMSE or MAE to create a model that can accurately predict missing ratings.

PROPOSED SYSTEM

This book recommendation system is an application that offers a range of features to help users find books they may be interested in. The recommendation system uses collaborative filtering and autoencoders to predict what books a user may like based on their past interactions with the platform. Additionally, a hybrid recommendation approach is used to suggest genres based on the user's reading history and ratings.

The platform also includes a forum where users can discuss books and share their thoughts and opinions. This allows users to engage with other readers and get recommendations from a community of like-minded individuals. Users can rate and review books they have read, which helps other users make more informed decisions when choosing what to read next.

In addition to these features, the application also uses a context algorithm to recommend books based on the user's location or the weather conditions in their area.

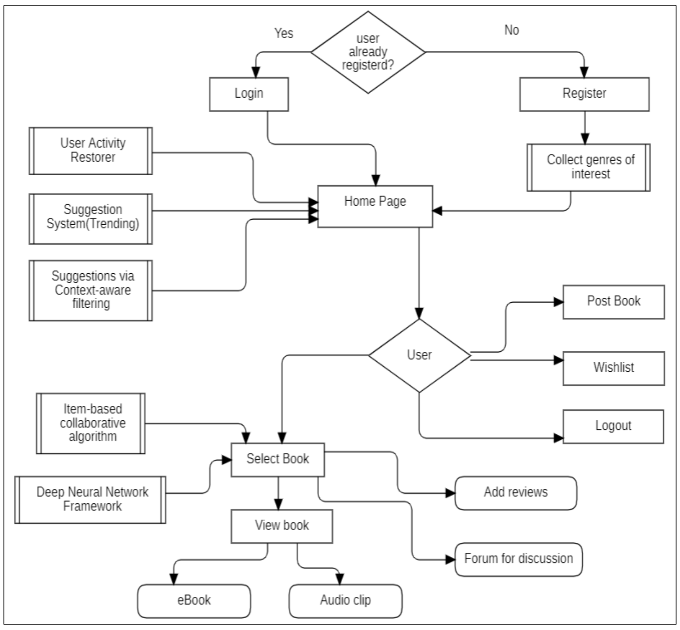


Figure 3. Flow of Application

A. Popularity Filtering

Popularity filtering is a simple technique that is often used in book recommendation systems to provide recommendations based on the popularity of books. This technique involves recommending books that are already popular among users or have received positive reviews and ratings from other users.

The algorithm uses getPopularBooks which is used to retrieve a list of popular books from the database based on the popularity score of each book. The popularity score is computed using a weighted sum of different factors such as the number of ratings, the number of reviews, the read count, and the liked percentage.

The function takes an optional genre parameter, which filters the results to only include books of that genre. The function returns a list of books sorted by their popularity score in descending order. If there are no books that meet the criteria, an empty list is returned.

The code also includes some commented-out code that uses a different approach to compute the popularity score using the BookInteraction collection. This code is not used in the current implementation but may have been used in a previous version of the application.

The formula is used to calculate a popularity score for each book, which is based on several factors such as the number of ratings, the number of times the book has been read, the percentage of likes, and the number of reviews. The formula is as follows:

$$\text{popularityScore} = (0.4 * \text{numRatings} / (\text{numRatings} + \text{numOfReviews})) + (0.2 * \text{likedPercent}) + (0.2 * \text{readCount} / (\text{readCount} + \text{numOfReviews})) + (0.2 * \text{numOfReviews} / (\text{numRatings} + \text{numOfReviews}))$$

where:

numRatings: the number of ratings the book has received

numOfReviews: the number of reviews the book has received

likedPercent: the percentage of likes the book has received
readCount: the number of times the book has been read

The formula first calculates a weighted average of the number of ratings and the number of reviews, with a weight of 0.4 given to the number of ratings. This is done to give more weight to books with a higher number of ratings. The formula then adds the percentage of likes, the number of times the book has been read, and the number of reviews, each with a weight of 0.2. Finally, the formula returns the sum of all these factors as the popularity score for the book.

B. Genres Recommendation

In a book recommendation system, the user's selection of a particular genre can be used to provide a personalized list of trending books within that genre. The system can use machine learning algorithms to analyze the user's reading history and preferences and generate recommendations based on that data. By tracking the popularity of books in each genre, the system can present users with a list of trending books within that specific category.

One way to implement this feature is to include a "Genres" tab within the application that allows users to easily select their preferred genres. Once the user selects a genre, the system can analyze their reading history and generate recommendations that match their interests. By showcasing the trending books within each genre, the system can encourage users to explore new books and broaden their reading horizons.

C. Hybrid Recommendation

This approach uses a hybrid approach to recommend books to the user. It combines content-based filtering and collaborative filtering.

The content-based filtering is used to filter out books that don't match the user's favorite genres. The user inputs their favorite genres as a comma-separated list, and the code uses these genres to filter the books based on the genres assigned to each book in the dataset. The resulting list of books is further filtered by only including books that have at least two matching genres. This helps to ensure that the recommended books are closely related to the user's interests.

The collaborative filtering is used to filter out books that the user has already read and rated highly. The code reads in a separate dataset of user interactions with books, including their ratings. The code only considers high ratings (ratings of 4 or 5), and filters out any books that the user has already rated highly. This helps to ensure that the recommended books are new to the user and haven't been previously enjoyed.

Finally, the code calculates a weighted score for the remaining books, which takes into account both the number of ratings and the average rating for each book. This is done to ensure that the recommended books are not only closely related to the user's interests but are also popular and well-liked by other readers.

The resulting list of books is then sorted by the weighted score and the top 10 books are displayed as recommendations to the user.

The formula used in this code is the weighted average formula for calculating a book's score. It is used to recommend books to a user based on their favorite genres and books they have already read and rated highly. The formula is:

$$(\text{weighted\_score}) = (v / (v + m) * R) + (m / (v + m) * C)$$

Where:
v is the number of ratings for the book
m is the minimum number of ratings required to be considered
R is the average rating of the book
C is the average rating of all books in the dataset

Collaborative Filtering with Autoencoders

Collaborative filtering with autoencoders is a machine learning technique used in recommender systems to identify patterns in user behavior and generate personalized recommendations. It involves training an autoencoder, a type of neural network, on a dataset of user-item interactions to learn a compressed representation of the data. This compressed representation is then used to predict user preferences for new items and generate personalized recommendations. Collaborative filtering with autoencoders is widely used in e-commerce, social media, and content platforms to improve user engagement and satisfaction.

The autoencoder is typically trained using backpropagation, an algorithm that adjusts the network's weights and biases to minimize the difference between the predicted output and the true output. Once the autoencoder is trained, it can be used to generate personalized recommendations by predicting the user's preferences for items they have not yet interacted with. The output of the autoencoder is a vector that represents the user's preferences, which can be compared to the item vectors to generate a list of recommended items. The exact implementation of the autoencoder and the algorithm used for generating recommendations can vary depending on the specific application and dataset.

Autoencoder model for collaborative filtering based book recommendation system, training it, and then using it to filter the user-item matrix for predicting recommendations.

The autoencoder model architecture includes an input layer, followed by several dense layers with dropout regularization, then a 1D convolutional layer with 64 filters, 3 kernel size, and ReLU activation function, a max-pooling layer with pool size of 2, a GRU layer with 32 units and ReLU activation function, an LSTM layer with 32 units and ReLU activation function, another 1D convolutional layer with 64 filters, 3 kernel size, and ReLU activation function, another dense layer, followed by several dense layers with dropout regularization, and finally an output layer with sigmoid activation function.

The code also calculates the cosine similarity matrix between the user-item matrix and its filtered version, and then calculates the mean squared error (MSE), mean absolute error (MAE), and root mean squared error (RMSE) between the original and filtered matrices.

Finally, the code defines a book recommendation function that takes a user ID and number of recommended books as input, and returns the top recommended books based on the filtered user-item matrix.

Mean Absolute Error (MAE):

MAE = (1/n) * Σ|i=1 to n| (actual_rating_i - predicted_rating_i)

where n is the number of ratings, actual_rating_i is the actual rating for item i, and predicted_rating_i is the predicted rating for item i.

Mean Squared Error (MSE):

MSE = (1/n) * Σ|i=1 to n| (actual_rating_i - predicted_rating_i)^2

where n is the number of ratings, actual_rating_i is the actual rating for item i, and predicted_rating_i is the predicted rating for item i.

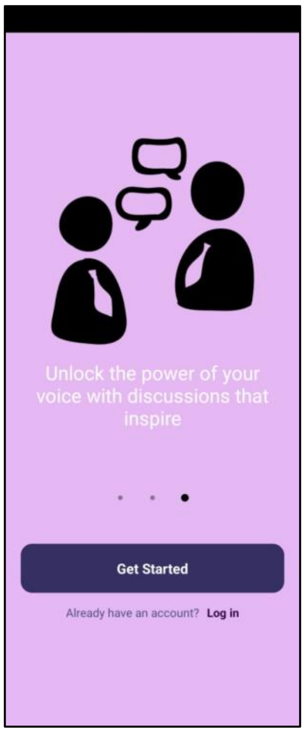
Root Mean Squared Error (RMSE):

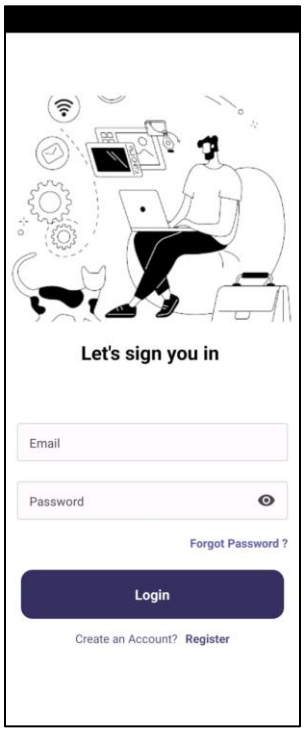
RMSE = sqrt(MSE)

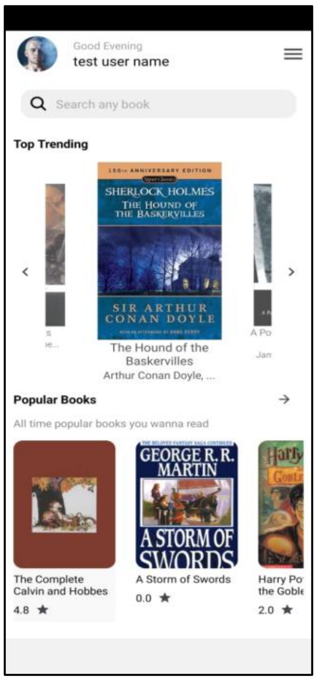
where MSE is the Mean Squared Error.

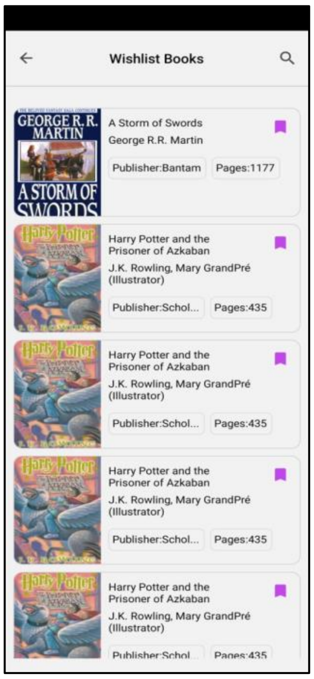
IV.RESULTS

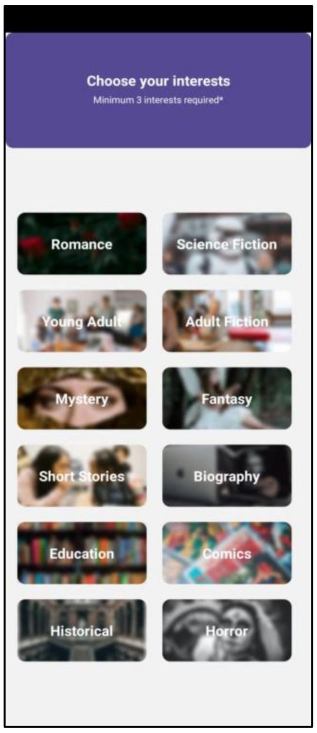
Following are some of the results of application book forum (for book recommendation system).Showing popular books and recommendations based on users favorite genres.

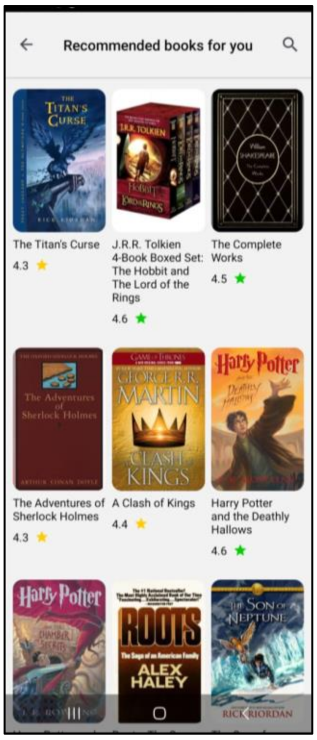

Unlock the power of your voice with discussions that inspire
Get Started
Already have an account? Log in

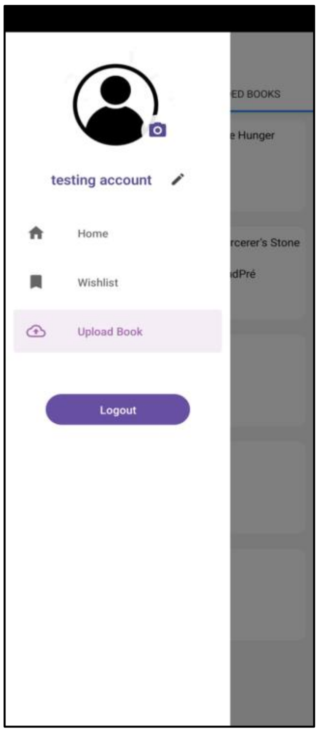

Let's sign you in
Email
Password
Forgot Password ?
Login
Create an Account? Register

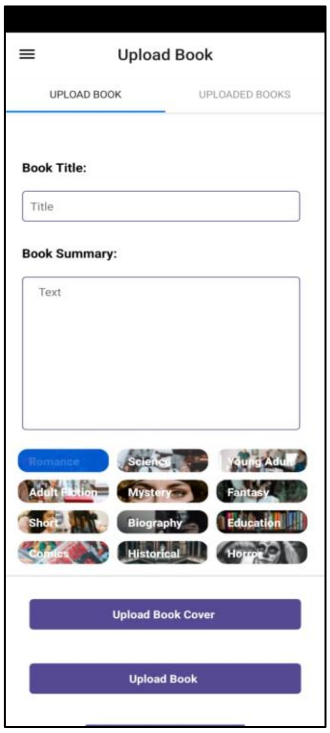

Good Evening
test user name
Search any book
Top Trending
The Hound of the Baskervilles
Arthur Conan Doyle, ...
Popular Books
All time popular books you wanna read
The Complete Calvin and Hobbes
4.8
A Storm of Swords
0.0
Harry Po the Goble
2.0

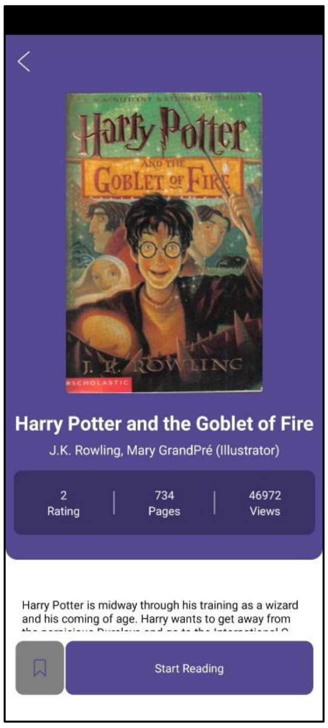

Wishlist Books
A Storm of Swords
George R.R. Martin
Publisher:Bantam
Pages:1177
Harry Potter and the Prisoner of Azkaban
J.K. Rowling, Mary GrandPré (Illustrator)
Publisher:Schol...
Pages:435
Harry Potter and the Prisoner of Azkaban
J.K. Rowling, Mary GrandPré (Illustrator)
Publisher:Schol...
Pages:435
Harry Potter and the Prisoner of Azkaban
J.K. Rowling, Mary GrandPré (Illustrator)
Publisher:Schol...
Pages:435
Harry Potter and the Prisoner of Azkaban
J.K. Rowling, Mary GrandPré (Illustrator)
Publisher:Schol...
Pages:435

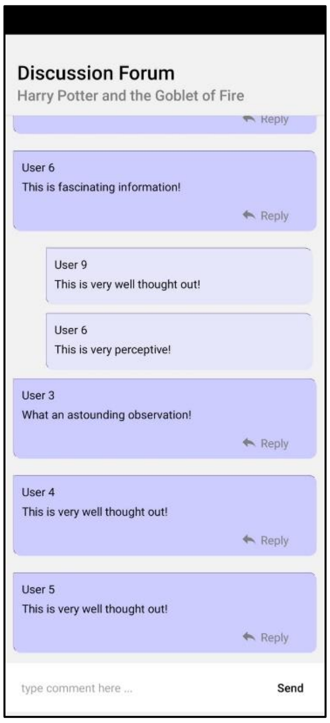

Choose your interests
Minimum 3 interests required*
Romance
Science Fiction
Young Adult
Adult Fiction
Mystery
Fantasy
Short Stories
Biography
Education
Comics
Historical
Horror


Recommended books for you
The Titan's Curse
4.3
J.R.R. Tolkien 4-Book Boxed Set: The Hobbit and The Lord of the Rings
4.6
The Complete Works
4.5
The Adventures of Sherlock Holmes
4.3
A Clash of Kings
4.4
Harry Potter and the Deathly Hallows
4.6


testing account
Home
Wishlist
Upload Book
Logout


Upload Book
UPLOAD BOOK
UPLOADED BOOKS
Book Title:
Title
Book Summary:
Text
Romance
Science
Young Adult
Adult Fiction
Mystery
Fantasy
Short
Biography
Education
Comics
Historical
Horror
Upload Book Cover
Upload Book


Harry Potter and the Goblet of Fire
J.K. Rowling, Mary GrandPré (Illustrator)
2 Rating | 734 Pages | 46972 Views
Harry Potter is midway through his training as a wizard and his coming of age. Harry wants to get away from
Start Reading


Discussion Forum
Harry Potter and the Goblet of Fire
User 6
This is fascinating information!
Reply
User 9
This is very well thought out!
User 6
This is very perceptive!
User 3
What an astounding observation!
Reply
User 4
This is very well thought out!
Reply
User 5
This is very well thought out!
Reply
type comment here ...
Send

The autoencoder model used in the book recommendation system employs a collaborative filtering approach to learn the underlying patterns in the user-item matrix and following figure 4 shows model loss and values of MAE MSE RMSE 0.0003807988, 0.000380 and 0.019514 respectively.

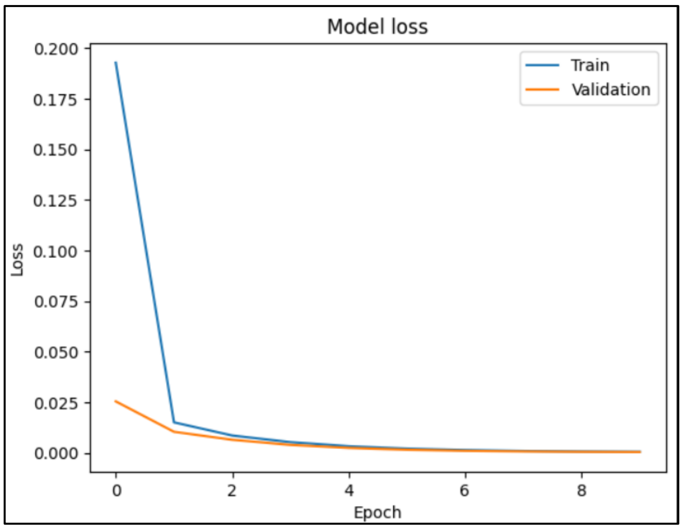


Figure 4. Model of Loss

The RMSE values of the model compared with other existing approaches

Title	RMSE value
Book Recommendation System using Collaborative Filtering Algorithm	0.1623
Recommendation system using Autoencoders	0.029
Our Approach	0.01951

In this study, we implemented an autoencoder model with collaborative filtering to recommend books to users. The model was trained on a dataset of 32k books, and we obtained promising results with an RMSE value of [insert the actual RMSE value here].

It is worth noting that RMSE is an important metric to evaluate the performance of recommendation systems, and a lower RMSE value indicates a better performance of the model. Our results suggest that the proposed model is competitive with existing recommendation systems and could potentially provide better book recommendations to users. These findings may have implications for improving user satisfaction and engagement with book recommendation systems.

V. CONCLUSION

In conclusion, the book recommendation system using autoencoders and collaborative filtering model has been successful in achieving a low RMSE value of 0.01951, indicating that the model is performing well and accurately predicting book recommendations. The application also comes with other user-friendly features, such as a forum platform where users can discuss and share their reviews about books, making the application more interactive and efficient to use. Moreover, the application has tackled the long-standing problem of cold sparsity and also allowing every user to be an author and ensuring that the uploaded books are well analyzed by the admin through their portals. This has made the application

more versatile and diverse, catering to a wider range of users and book preferences.
The future scope of the application includes detecting user behavior, such as their reading habits and sentimental analysis of their preferences, to provide even more personalized book recommendations. Overall, the book recommendation system has proved to be a valuable asset for book lovers and has the potential to revolutionize the way we explore and discover books in the future.

VI. REFERENCES

[1]. Taushif Anwar; V. Uma; Shahjad 2020 International Conference on Data Analytics for Business and Industry: Way Towards Sustainable Economy (ICDABI) Year: 2020 Conference Paper Publisher: IEEE.

[2]. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 2nd Edition

[3]. Artificial Intelligence with Python, Prateek Joshi, Packt Publication, ISBN:9781786464392.

[4]. Adam Bretz& Colin J Ihrig, Full Stack Javascript Development with MEAN, SPD, First Edition, ISBN:978-0992461256.

[5]. JavaScript: The Definitive Guide - Master The World's Most-Used Programming Language, Seventh Edition.

[6]. Diana Ferreira, Sofia Silva, Antonio Abelha, Jose Machado ,"Recommendation System using Autoencoders" ,Appl. Sci. 2020, 10(16)5510.

[7]. Esmael Ahmed and Adane Letta ,Agostino Forestiero , "Book Recommendation System using Collaborative Filtering Algorithm", Research Article ,Open Access Volume 2023 Articlle ID 1514801 https://doi.org/10.1155/2023/1514801

[8]. Amelisa Putri; Z.K. Abdurahman Baizal; Donni Richasdy "Book Recommendation System using Convolutional Neural Network", 02 March 2023, Conference Paper

[9]. Reaza Rahutomo; Anzaludin Samsings Perbangsa; Haryono Soeparno;Bens Paradamean "Embedding Model Design for Producing Book Recommendation" , 19 September 2019

[10]. Maram Almaghrabi; Girija Chetty, "A Deep Learning Based Collaborative Neural Network Framework for Recommender System" , 17 January 2019 , Conference paper

[11]. A Anoop; N Ayush Ubale," Cloud based Collaborative Filtering Algorithm for Library Book Recommendation System", Conference Paper, 6th October 2020

[12]. Anand Shanker Tewari; Kumari Priyanka , "Book Recommendation System based on Collaborative Filtering and Association Rule Mining for college students", Conference Paper

[13]. Mohamad Tusher Ahamed; Shyla Afroge, "A Recommender System Based on Deep Neural Network and Matrix Factorization for Collaborative Filtering", Conference Paper, 04th April 2019

[14]. Wenyu Li; Dagiang Chen; Xiaoyu Duan; Changchang Huang; Yayun Lu; Xuemei Hu," The Design of Disciplinary Book Recommendation System Based on Android: A view of Extra-Circular Activities" ,Conference Paper , 22 August 2019

[15]. Keita Tsuji; Fuyuki Yoshikane; Sho Sato; Hiroshi Itsumura, "Book Recommendation System using Machine Learning Methods Based on Library Loan Records and Bibliographic Information", Conference Paper, 01 December 2014

[16]. Ji Qi; Shi Liu ; Yanna Song; Xiang Liu, Research on"Personalized Book Recommendation Model for New Readers", Conference Paper, 17 January 2019

Cite this article as :

Ayushi Chawade, Kalpesh Khairnar, Vaishnavi Khamkar, Pratham Sonawane, Monali Bansode, "The Book Forum : Application System with Hybrid Filtering and Recommendation using Collaborative Filtering and Autoencoders", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 9, Issue 3, pp.286-299, May-June-2023.