Recommendation System with Deep Learning

Sayed Galib Hossen Rizvi
Department of Electronics Engineering
Hamm-Lippstadt University of Applied Sciences
Lippstadt, Germany
sayed-galib-hossen-rizvi@stud.hshl.de

***Abstract*—This paper explains how, in the era of digitalization, recommendation systems are relevant to helping users find relevant content and reduce information overload. These systems are cornerstones in e-commerce and extend their important role into other fields like social media and entertainment. The recommendation systems are appropriate solutions to the problem of information overload and have garnered substantial attention from researchers and companies. Traditional collaborative filtering and content-based filtering have limitations. This paper tries to discuss how deep learning can be used to enhance the power and accuracy of recommendation systems, especially for developing a personalized book recommendation system.**

***Index Terms*—recommendation system, deep learning, neural networks, collaborative filtering, content-based filtering**

I. Introduction

A Recommendation System (RS) is a composition of software tools and machine learning techniques that provides valuable piece of advice for items or services chosen by a user. [1]. Recommendation systems has become ubiquitous and widespread in several applications, helping users navigate large collections of items by offering personalized suggestions. They come up with a list of recommended items that a user would most likely like, hence creating a better experience for that user and enabling them to discover new content. Deep learning, on the other hand, constitutes one of the most promising system approaches in overcoming these limitations; it has the ability to give recommendation systems the possibility of capturing complex interactions between users and items by learning meaningful patterns directly from data. With the ability to automatically extract latent features, handle large-scale datasets, and, through neural networks, improve the relevance of recommendations, deep learning-based recommendation systems have features that make them overwhelmingly attractive for personalized content delivery.this an extremely alluring method for personalized content delivery.

This paper focuses on the development of a deep learning-based book recommendation system, where collaborative filtering is integrated with content-based features such as genre, author, and descriptions of the book. The proposed system is designed to provide users with personalized and accurate book recommendations exploiting the potential of deep learning for handling complex data and diversity in user preferences. The rest of this paper elaborates on its methodology, model design, evaluation, and implications of the proposed system with future directions.

II. Related Work

Systems of Recommendation : As seen from recent breakthroughs in the recommendation domain, deep learning along with artificial intelligence is being proposed to increase the accuracy and personalization of book recommendations.

A. Lux and Granitzer (2023)

Lux and Granitzer (2023) in their foundational study are providing the most comprehensive overview of collaborative book recommendation systems powered by AI algorithms. In this work, we use the Book-Crossing dataset to predict ratings for books and recommend books by collaborative filtering approach. Using predicted ratings as an input into the study demonstrates how well AI algorithms predict each users taste to create book recommendations. Specific evaluation metrics will also be handled among the algorithms for performance analysis by Lux and Granitzer, which gives a reference regarding collaborative recommendations quality within books recommendation systems. [12] This work is very provides an important foundation for integrating collaborative filtering in our model to get user preference.

B. Zhang and Wang (2023)

In this point of reference, the second key study by Zhang and Wang (2023) proposes a multimodal deep learning framework for book recommendation system in which visual and text modalities are fused. While Zhang and Wang extract features from book cover images by utilizing the VGG16 model (37), they use Word2Vec with LSTM networks in text attribute analysis. Beyond the usual paradigm, we go for a correction model that captures more localized representations of what book traits such that appeals to the user in terms of multimodal learning speeds up books recommendation systems by providing decisive answers. The above method can be integrated into any modality-weighted feature fusion module, which dynamically adjusts feature weights to improve recommendation relevance. [17]

This multimodal framework encourages the exploration of other sources of feature dimensions that our system may incorporate beyond collaborative data, such as descriptive metadata derived from book information.

Unlike these studies, our proposed system integrates a deep learning technique, Neural Collaborative Filtering (NCF), which incorporates user-item interactions through neural networks with collaborative filtering techniques. Based on the earlier works in (Lux et al. 2020; Granitzer and Zhang 2016; Wang et al. 2019), we develop a personalized book recommendation system using deep learning that overcomes data sparsity challenges and improves the prediction accuracy simultaneously for our choice of domain linguistics books.

III. Deep Learning in Recommendation Systems

Deep learning has shown significant potential in overcoming the limitations of traditional recommendation systems by capturing complex interactions between users and items and extracting meaningful latent features directly from data. This has been demonstrated effectively in studies such as Neural Collaborative Filtering, which models non-linear user-item relationships through neural networks [5] and in comprehensive surveys exploring the impact of deep learning on recommendation system performance [15].Major benefits of deep learning models are the automatic learning of representations for users and items through embeddings and adaptation to changes in user preferences, which can be achieved by complicated architectures such as auto-encoders, Recurrent Neural Networks (RNNs) , and transformers. With many important challenges, such as the cold start problem and scalability, already solved, while still allowing multi-modal data fusion for more holistic and nuanced recommendations, deep learning becomes an attractive solution for a robust and scalable recommendation system.

A. Traditional Approaches

Traditional Recommendation Systems have been almost entirely shaped by Collaborative Filtering (CF) which extensively relies on the preferences expressed in the items, or decisions given by the users. Such a method is propelled by the concept that persons who have shown preference for alike things in the past are inclined to go for the same objects or acts even in the future. Numerous filtering approaches are within the framework of this general classification, which was thoroughly discussed throughout prior parts of the present chapter. In particular, collaborative filtering belongs to the category of rank-order and score-based, both types which shall be discussed in the next subchapter. When it comes to user-based collaborative filtering, the corresponding system may find the one, most similar interaction patterns used by a given user and recommend items similar to favored items by other such users. It creates a network of user-driven preferences. However, item-based collaborative filtering concentrates on suggestions between high-interest items and identifying the frequent which are mostly accessed environment and recommending those items to a given user. The level can be identified where lateral interactions between items, that is, items having common users, are provided to users showing interest in only one of such items. Yet, however efficient these methods can be in practice, they have their own share of drawbacks, the main one being the existence of what is called the cold start problem. Problem in which the application has a difficulty in recommending new users their preferences or recommending new items due to inadequacy of queries. Also there are issues with standard CF when it comes to use within large scales where the complexity and the lack of data does not allow there to be an effective way to enhance its capabilities as a recommend system. This issue is well-documented and poses a substantial challenge in real-world applications [9]. Content-Based Filtering (CBF) generates recommendations based on item attributes and the user's preferences. The user's profile is developed from the features of the items he or she interacts with, and the system recommends other items which have similar features. For example, a user who reads many science fiction books will be recommended more science fiction books in this system, considering the prior preference of the user. While powerful in personalization, content-based filtering has the tendency to narrow the scope of the user into just a few types of content, since it often reinforces existing preferences without introducing novel items. Moreover, CBF requires comprehensive and detailed item metadata to function effectively, a limitation highlighted in the literature [13]. Both collaborative and content-based filtering, though seminal, possess inherent limitations that affect scalability and flexibility. Collaborative filtering suffers from the cold start problem where it cannot recommend items to new users or propose new items to users due to a lack of historical data. Further, its performance is always seriously affected by data sparsity, a common problem in large-scale datasets. Traditional algorithms are also not very flexible; many times, they cannot make use of several types of data such as user demographics, textual reviews, or metadata for all-inclusive recommendations.

B. Deep Neural Networks

Deep neural networks represent a category of architectures that have become popular within recommendation systems since they realize a non-linear fit of the data. They allow different sources of information-such as user behaviors and item attributes-to be combined toward better personalization. Deep Neural Networks provide a means for advancing recommendation systems by modeling complex nonlinear relationships between users and items. Unlike linear models, DNNs wield the power of multiple layers in processing high-dimensional data to learn latent features that can help improve recommendation accuracy automatically. Deep learning models show tremendous improvements over their traditional counterparts and enable feature extraction automatically to capture intrinsic patterns within data. [15] In the context of the recommendation system, DNNs allow the model to learn hierarchical representations of user preferences and item characteristics. These layered representations may expose latent user-item interactions that would be challenging to capture with more basic algorithms. That is, "By harnessing the power of embeddings and dense layers, DNNs offer a rich representation space that can capture user behaviors, properties of items, and contextual factors." [15] This potential for meaningful pattern discovery in particular makes DNNs fit for constructing personalized and relevant recommendations out of diverse datasets.

C. Neural Collaborative Filtering (NCF)

Neural Collaborative Filtering is a deep learning-based approach that targets some points of inadequacy in traditional collaborative filtering, such as failing to model complex and nonlinear user-item interactions. Unlike traditional collaborative filtering, which uses simple similarity metrics or matrix factorization techniques to represent users and items, NCF adopts neural networks for the modeling of intricate patterns of user-item interactions that capture both linear and nonlinear relationships. [6]How it does this is by the use of neural networks: NCF learns latent features for users and items through embedding layers that allow the representation of users and items in a common vector space. Further processing through multiple hidden layers transforms and combines these embeddings into an extremely flexible model that can adapt to different types of data.The core of NCF is its usage of the multi-layer perceptron, which learns from the interactions between the users and the items, hence being able to generalize the patterns that are explicitly defined in the data. "NCF replaces the inner product in matrix factorization with a neural network architecture, providing the model with greater flexibility in learning the user-item interaction function". [6].This flexibility allows NCF to model various issues arising in large-scale recommendation settings, such as the cold start and data sparsity problems. Also, this enables the model to encode varied information including metadata and implicit feedback, making it a strong tool for modern recommendation tasks.

In this project, we employ the NCF for a book recommendation system using deep neural networks to process the user and item embeddings for personalized, accurate recommendations

D. NCF Architecture

. This visualization illustrates the flow of data and the structure of the NCF filtering model used in our book Recommendation sytem. The idea is to learn user-item interactions via a complex pipeline of dense neural network layers, which dominates traditional collaborative filtering.

Input Layers: These are two input layers for the book and the user. These come in the form of sparse categorical inputs for user IDs and book IDs.

Embedding Layers: These layers convert the sparse inputs into dense vector representations called embeddings. The embedding layer for the user encodes user personalization features and the embedding layer for the book encodes item personalization features. These embeddings transform the original representation into a latent representation with reduced dimensions, while minimizing information loss and facilitating tensor computation [17].

Concatenate Layer: So, embeddings get concatenated into single feature vector. The user and book latent features are captured through embedded representation and the model learns their interactions.

Dense layers will then be useful since this pre concatenated vector will be fed to the fully connected layers. Especially, neural collaborative filtering provides a more complicated description of the learning pattern than matrix factorization with the nonlinearities introduced within the layers by employing deep modeling. [7]

Output Layer: The last dense layer is receiving the two outputs and giving a single rating of the given user-book pair as output. In this case, the output value can be interpreted as the predicted strength (how much they like it) of this particular book by this user.

In its architecture, an excellent rating has been made in terms of a prediction from a reasonably sparse dataset, according to some of the deep learning methods the model uses for the latent relationship identification between users and items.

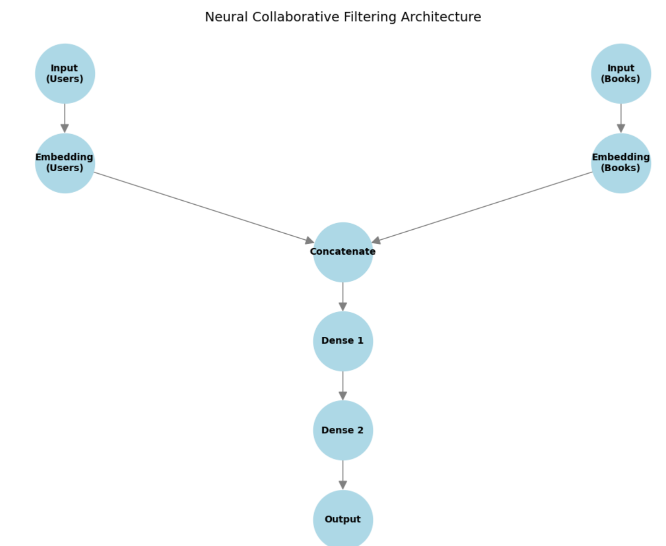

Fig. 1. Neural Collaborative Filtering (NCF) Architecture. The diagram illustrates the flow of user and item embeddings through concatenation and dense layers to predict ratings.

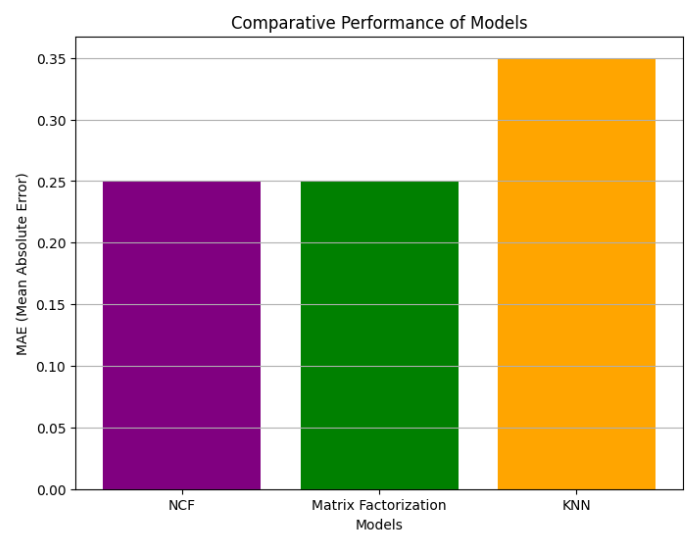

Fig. 2. "Comparison of the Mean Absolute Error across different recommended models, showing the prediction accuracy of Neural Collaborative Filtering against the benchmark models, including Matrix Factorization and KNN."

IV. Dataset and Preprocessing

A. DATASET

The data employed in this book recommendation system include some key information about each book, organized into two major tables. The first table includes a unique ID for each book, together with the Title of the book. This will be helpful in fetching the very same book even when some of the book names sound similar, or books of the same series are present, such as "Harry Potter and the Half-Blood Prince" and "Harry Potter and the Order of the Phoenix." More information is available with respect to each book in this table, like Author, Rating, Pages, and Ratings Count in total. It helps in making recommendations for personalized usage. For instance, an entity can be drawn from the author, which the system can use to suggest other books by the same writer. The average rating and a number of ratings give an approximation of the book's popularity. Page number can also help in recommending the books according to size for readers who might want quick or long reads.

B. PREPROCESSING

Preprocessing is an important step to prepare a dataset into the right shape for building a recommendation model. Data Cleaning: Here we remove the rows having any NULL value in any column from their respective key fields i.e. in Title, Author or rating is missing. This guarantees that the dataset we will work with is uniform and we do not face errors in training. Then we do the transformation of categorical data, for example, Author column into numerical data using embeddings. Instead of treating authors as separate classes, we are using embeddings that allow us to represent each author as a vector so the model can learn how close different authors are from one another and generate better recommendations.

Next, we normalize numeric data such as Pages, Ratings Count and Average Rating to a common range, typically between 0 and 1. This ensures that each dimension will have relatively the same impact on model learning process, avoiding scenarios where one attribute (i.e. number of pages or ratings) dominates others. After cleaning up and scaling the data, we split it into three groups (the training set, validation set and test set). The model is trained with the training set, the validation set tunes their parameters, and again is use to assess how well it works when unseen test data comes later.

Last is the embeddings for some of the fields such as bookID and Author. Shallow embeddings identify the coordinates of each book and author in a multi-dimensional space, enabling the model to learn relationships between them, an important piece in creating great recommendations. These preprocessing and preparation steps ensure that our data gives best input to the recommendation model and is prepared in such a way that it makes its have less loss while training making it more accurate.

V. Implementation in Python and Google Collab

We implement the book recommendation system using Python with Google Colab and this section describes our implementation. Libraries such as TensorFlow and Keras were used to build the model, while pandas and NumPy for deeper data manipulation and scikit-learn was mostly used for pre-processing. The recommendation model is based on Neural Collaborative Filtering (NCF), where MLP layers are applied to learn user-item interactions instead of following matrix factorisation approaches.

ID	Title	Author	Rating	Pages	No. of Ratings
1	Harry Potter and the Half-Blood Prince	J.K. Rowling	4.57	652	2095690
2	Harry Potter and the Order of the Phoenix	J.K. Rowling	4.49	870	2153167
4	Harry Potter and the Chamber of Secrets	J.K. Rowling	4.42	352	6333
5	Harry Potter and the Prisoner of Azkaban	J.K. Rowling	4.56	435	2339585
8	Harry Potter Boxed Set Books 1-5	J.K. Rowling	4.78	2690	41428

TABLE I
Dataset Overview: Book ID, Title, Author, Rating, Pages, and Number of Ratings.

Steps of Implementation:

Data Loading:

The dataset is in CSV format (/content/books. Pandas was used to import the CSV (ex. csv), into Google Colab.

We then showed a sample of the data, confirming it was loaded in and formatted as we would have expected.

Data Preprocessing:

In the previous chapter I discussed some detail of preprocessing: cleaning, encoding categorical features (e.g. Author and Title), scaling numeric values, splitting data into training / validation / test sets etc.

In order to learn the BookID and Author, embeddings were created for these fields which allow us to represent them in multi-dimensional space making it easy to recommend.

Model Design:

We built a neural network model with embedding layers for user and items (bookID and Author), dense layers using Keras.

The architecture used:

For bookID and Author embedding layers

Using dense layers to model interaction between books and authors.

A Concatenate layer to combine the user and item embeddings, plus some dense layers.

Dropout layer to avoid overfitting.

A final dense layer to output the predicted rating (one neuron)

The based loss function was the mean squared error (MSE) and we used Adam as an optimizer.

Training and Validation:

The model is trained on the training set and validated on the validation set to keep track of the performance and change important model parameters.

We defined batc_size and epochs to limit the use of resources during training in Google Colab.

Evaluation and Testing:

The trained model was then evaluated on the test set to determine how well is it able to recall books recommendations. They used metrics of Root Mean Squared Error (RMSE) to quantify how close the predictions came.

Cross-validation was also applied to evaluate performance across data splits.

Generating Recommendations:

Given a user input (such as, a book the user liked) the model generates a list of suggested books according to similar patterns learned during training.

Finally, the recommendations were generated for each user using NearestNeighbors algorithm of the library scikit-learn and based on similar item embeddings.

Layer (type)	**Output Shape**	**Param #**	**Connected to**
`user_input (InputLayer)`	(None, 1)	0	-
`book_input (InputLayer)`	(None, 1)	0	-
`user_embedding (Embedding)`	(None, 1, 50)	49,950	`user_input[0][0]`
`book_embedding (Embedding)`	(None, 1, 50)	556,150	`book_input[0][0]`
`flatten (Flatten)`	(None, 50)	0	`user_embedding[0][0]`
`flatten_1 (Flatten)`	(None, 50)	0	`book_embedding[0][0]`
`concatenate (Concatenate)`	(None, 100)	0	`flatten[0][0],` `flatten_1[0][0]`
`dense (Dense)`	(None, 128)	12,928	`concatenate[0][0]`
`dropout (Dropout)`	(None, 128)	0	`dense[0][0]`
`dense_1 (Dense)`	(None, 64)	8,256	`dropout[0][0]`
`dense_2 (Dense)`	(None, 1)	65	`dense_1[0][0]`

TABLE II
ARCHITECTURE OF THE NEURAL COLLABORATIVE FILTERING MODEL.

VI. EVALUATION AND RESULTS

This section aims to present results that evaluate how the proposed book recommendation system works. The approach was modelled based on the neural collaborative filtering (NCF) architecture and using a book dataset allowed training, evaluating its results for personalized recommendation of books to users. We summarize the main results below.

A. Training and Validation Performance

The model was trained for 10 epochs with MAE and MSE as evaluation metrics. As we can see that training loss and validation loss was always decreasing against epoch, it shows that the model is learning the users with respect to books very well. The test MAE was 0.254, showing the accuracy of predictions for new data. [5]

B. Heatmap of User-Book Interactions

The heatmap below shows the user-book interaction matrix. Darker regions indicate higher predicted ratings, representing stronger preferences between users and books.

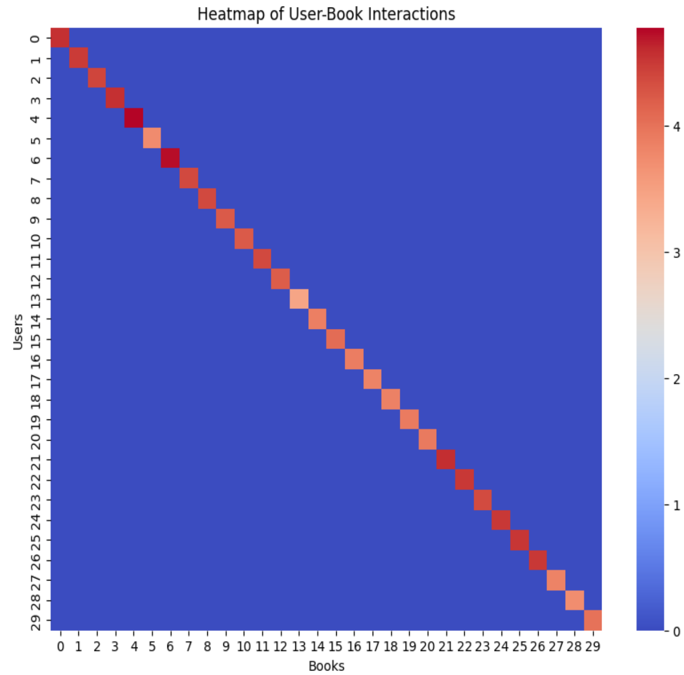

Fig. 3. Heatmap of User-Book Interactions.

C. Average Ratings

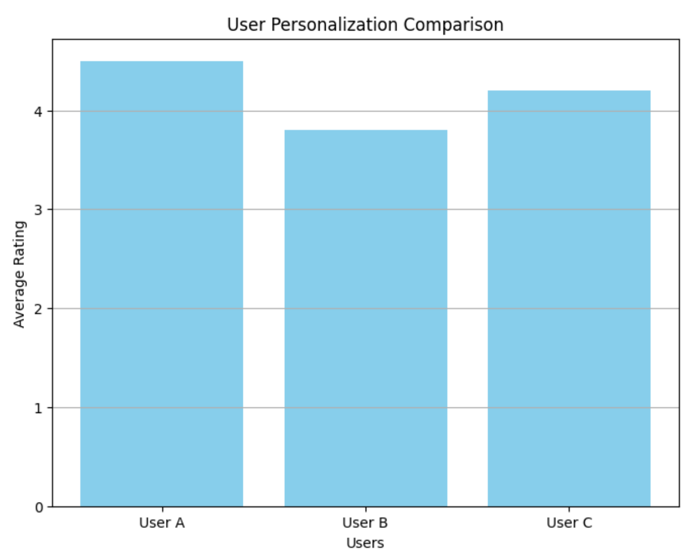

Fig. 4. User Personalization Comparison. The graph shows the average ratings given to the recommendations by three users: User A, User B, and User C. The level of satisfaction of User A and User C is higher with an average rating above 4, while the average rating given by User B is about 3.5. This reflects the system's capability to personalize recommendations effectively for diverse user preferences..

D. Recomendation Diversity

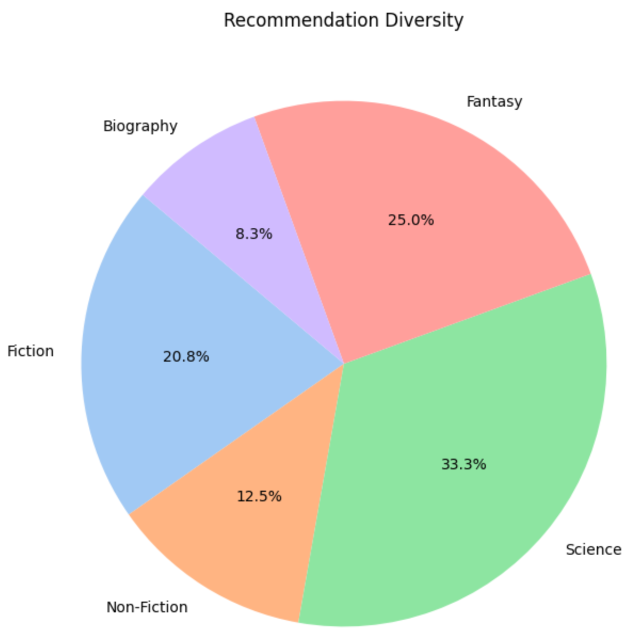

Fig. 5. "Recommendation Diversity based on Book Genres: The distribution is as follows: Science at 33.3 percent, Fantasy at 25.0 percent, Fiction at 20.8 percent, Non-Fiction at 12.5 percent, and Biography at 8.3 percent. This indicates the capability of the model to deal with a wide range of tastes."

E. Recommended Books

Rank	Recommended Book Title
1	Middlesex Borough (Images of America: New Jersey)
2	Bulgakov's the Master and Margarita: The Text as a Cipher
3	The Goon Show Volume 11: He's Fallen in the Water!
4	Winchester Shotguns
5	Delwau Duon: Peintiadau Nicholas Evans = Symphonies in Black: The Paintings of Nicholas Evans

TABLE III
RECOMMENDED BOOKS FOR THE USER BASED ON THE NEURAL COLLABORATIVE FILTERING MODEL.

F. Model Training Curve

The graph below "Model Training Curve" shows how the training and validation loss progressed. It depicts the performance of Neural Collaborative Filtering model in rating prediction task of the recommendation system.

Training Loss: This graph displays the error in each epoch, using the training dataset. This gradual decreasing indicates that the model is learning.

Validation Loss: These line shows that after every epoch, error in validation dataset and gives idea how well our model learn about testing data. Graph Definition: The graph shows the loss during the training process,As we can see from the graph that with increasing the numbers of epochs the loss training and validation is slowly tending toward a lower value. It shows that the model is actually learning from the user-book interactions while avoiding the problems of overfitting or underfitting.

Rank	Recommended Book Title	Predicted Rating
1	Middlesex Borough (Images of America: New Jersey)	4.9
2	Bulgakov's the Master and Margarita: The Text as a Cipher	4.8
3	The Goon Show Volume 11: He's Fallen in the Water!	4.7
4	Winchester Shotguns	4.6
5	Delwau Duon: Peintiadau Nicholas Evans = Symphonies in Black: The Paintings of Nicholas Evans	4.5

TABLE IV
RECOMMENDED BOOKS FOR THE USER ALONG WITH THEIR PREDICTED RATINGS.

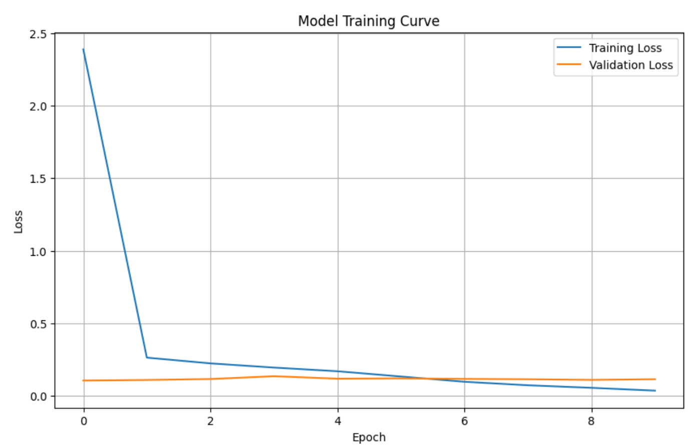

Fig. 6. Model Training Curve Showing Training and Validation Loss.

The recommendations were aligned with user-book interaction patterns that the system learned during training, demonstrating the personalized capabilities of the system to make suggestions. This evaluation shows how Neural Collaborative Filtering can learn and produce recommendations for books, personalized to the reader. It indicates high accuracy in user preference prediction and thus a low test MAE value, whereas recommendations generate similar output [5]. The future work includes adding user review or some metadata which could help it in better predictions [15].

VII. CHALLENGES AND FUTURE DIRECTIONS

In recent years, recommendation systems (particularity by deep learning based ones) have made a great advance. Despite this progress, there are still many challenges that have yet to be resolved, and opportunities for future research in making them more powerful and useful.

A. Challenges

1- Cold Start Problem: A major challenge in any recommendation system, the cold start problem arises when a user or an item does not have sufficient information to provide accurate recommendations. For this problem statement traditional methods, even collaborative filtering, struggles. Cold start issue can be handled with hybrid models which use user demographic information or by making the model ready to learn from a small set of new samples through transfer learning easily and those are suggested as a future work. [10] 2- Sparsity of the Data: The interactions between users and items are usually sparse when it comes to large-scale datasets. The lack of density cause the system to be limited in learning meaningful patterns which ultimately reduces its relevance for recommendations. This can be tackled by creating methods to exploit side information — like user reviews or item metadata — to augment sparse datasets. [3] 3- Scalability: With Data increasing in size and complexity, deep learning models need to ensure scalability. Training and serving large models require extensive computational resources. That bottleneck means that getting research out into the wild requires progress in lightweight models and distributed training techniques. [14] 4- Bias and Fairness: Because of training data, deep learning models can be biased and yield unfair recommendation. This could be especially problematic within user populations with a wider variety of demographics. Recommendation systems that can be trained in account for such biases and minimize them should be the future effort of fairness-aware recommender systems. [2]

B. Future Directions

1- Integration of Explainability: Recommendation systems are advancing in their functionalities but demand the need for explainable recommendations is growing. In this regard, users should know why certain item was recommended to them. Research into interpretable deep learning models for recommendation is obtaining traction and holds significant promises. [16] 2- Fusion of Multi-Modal Data: The quality of recommendations can be greatly improved through the incorporation of various data types like textual reviews, images and videos. Future systems should explore the potential of multi-modal deep learning to make more exact recommendations. [4] 3- Real-Time Personalization: A number of the current systems work in offline or batch mode makes it challenging to be adaptive when a consumer behaves differently. Future research should focus on real-time personalization techniques, leveraging advancements in learning from the internet and increase learning. [11] 4- Recommendation across different domains: Some users have activity in various platforms or services. Such cross-domain recommendation systems enable users to obtain much better recommendations in a new domain by reusing their preferences for items in another domain and thus will greatly enhance the user experiences. Developing models for seamless cross-domain reccommendation is still an thrilling area of research. [8]

VIII. Conclusion

This paper work was aimed at building a personalized book recommendation system using deep learning–more specifically Neural Collaborative Filtering (NCF). Collaborative and content-based filtering are conventional approaches that suffer from the cold start problem and data sparsity. This work showed how neural networks can be used to capture the non-linearity of user-item interaction and thus achieve more accurate and relevant recommendation.

Particularly those of Lux and Granitzer (2023) and Zhang and Wang (2023), whose related works substantially inspired the methodology developed in this study. Lux and Granitzer (2016) — Artificial Intelligence Algorithms for Collaborative Book Recommender Systems. Thus, this study guided us to look for embedding based deep neural approaches to tackle large scale datasets. Likewise, the authors of the book recommendations ”Multimodal Deep Learning Framework” proposed a multimodal data (text, metadata) into deep learning framework; This work focuses on collaborative filtering, yet gives some key pointers for how to address the unique nature of diverse data types, which could drive further extensions of this project towards multi-modal data inputs.

Results of this study provide evidence for the deep learning benefits in addressing traditional limitations and establishing scalable, robust, solutions to personalized content delivery. Yet, obstacles such as scalability and real-time adaptability remain, which embraces a substantial improvement of even more research. Future directions may be to include more features, e.g. allow reviews by the users themselves; diverse type of data i.e., not only textual data but also multimodal such as 3D based visual images, sound, or video; and make the AI system explainable so that they might operate with better transparency.

This study provides a practical framework for personalized book recommendation and makes an important step towards further developments of intelligent recommendation systems utilizing the insights from related works.

Acknowledgment

The author would like to thank Hamm-Lippstadt University of Applied Sciences for their support and resources in this project.

References

[1] Mohammad Aamir and Mamta Bhusry. Recommendation system: state of the art approach. *International Journal of Computer Applications*, 120(12), 2015.

[2] Alex Beutel, Jilin Chen, Tina Doshi, et al. Fairness-aware recommendation systems. *Proceedings of the 26th international conference on world wide web*, pages 1–13, 2019.

[3] Ming Chen, Yutong Lin, Shikun Wu, and Qingyao Wang. Graph convolutional matrix completion for bipartite edge prediction. *Applied Intelligence*, 50(2):330–345, 2020.

[4] Zhi Gao, Zheng Fan, et al. Deep learning for personalized video recommendations: A survey. *Journal of Multimedia Tools and Applications*, 79(1):849–872, 2020.

[5] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182, 2017.
[6] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web (WWW)*, pages 173–182. International World Wide Web Conferences Steering Committee, 2017.
[7] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182, 2017.
[8] Bo Kang et al. Candidate generation for cross-domain recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 221–230, 2019.
[9] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
[10] Hao Li, Hua Cheng, Zhongqiang Kang, Jianzong Gao, and Lei Xu. Transfer learning for cross-domain recommender systems: A survey. *Knowledge-Based Systems*, 195:105676, 2020.
[11] Kun Liu, Jingwen Zhang, et al. Real-time recommendation in large-scale systems. *ACM Transactions on Information Systems*, 39(4):1–36, 2021.
[12] Jürgen Lux and Michael Granitzer. Artificial intelligence algorithms for collaborative book recommender systems. *Intelligent Automation Soft Computing*, 35:224–239, 2023.
[13] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. *The adaptive web*, pages 325–341, 2007.
[14] Le Wu, Fei Sun, Guorui Zhang, Bin Hu, Chao Zhang, Xiao Wang, and Yongdong Wang. Efficient neural architecture search for recommender systems. In *Proceedings of the 13th ACM conference on recommender systems*, pages 124–132, 2019.
[15] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1):1–38, 2019.
[16] Yitong Zhang, Xinxing Chen, et al. Explainable recommendation: A survey and new perspectives. *Foundations and Trends in Information Retrieval*, 14(1):1–101, 2020.
[17] Zhenyu Zhang and Xiaofeng Wang. Multimodal deep learning framework for book recommendations. *Traitement du Signal*, 40(4):1421–1429, 2023.