SDSC4016 Project Report

Deep learning based Recommender systems
Using Amazon Reviews

Team 8

Chan Yuk Yee	56230549
Lau Ching Lam	56203744
Leung Chun Wai	56046680

1. Motivation

In this project, a large set of data was given, which contains valuable information about the customer's reviews and product information on Amazon, an e-commerce company that provides online shopping services for customers worldwide (Amazon, n.d.).

One way to utilize the dataset and generate insight is to construct a recommender system, which is a system that produces individualized suggestions as the output that could act as a guideline for users to look for the appropriate items from the massive amount of options (Burke, 2011), by predicting the preference of the consumers based on their consumption habits and product metadata with the use of deep learning techniques.

By building the recommender system, consumers could benefit from finding the right products they need efficiently. At the same time, the platform and product providers could also be favored by matching the correct products with the correct customer and gaining profits from the increased sales.

2. Background

The history of recommender systems can be traced back to the early 1990s when Goldberg and other researchers first proposed collaborative filtering to recommend items based on users' ratings (Dong, 2022). The recommender system grew and evolved rapidly throughout the decades. Most e-commerce and entertainment service providers, like Amazon, Youtube, and Netflix, have applied this technology to enhance user engagement.

Currently, several types of recommender systems are widely used in different platforms, which includes popularity-based recommender system that recommends the most popular products to users, content-based recommender system that recommends similar products to customers, classification model that predicts whether the customer would be interested in the product or not, and some more advanced systems, including collaborative filtering and the hybrid approach (Dwivedi, 2020).

Although the recommender system is a powerful tool that provides advantages to the two ends of the products and services, different issues and challenges have been raised upon its successful development. First, some systems cannot cover new users or products, generally described as the cold-start problem (Lika, 2014). Second, some systems cannot interpret synonyms as the same thing, like they may understand the words 'film' and 'movie' differently (Dwivedi, 2020). Third, shilling attacks in which the misbehaved users may inject misleading comments and ratings into the system (Zhou, 2018). Fourth, when the number of items in a system is vast, it is hard to obtain sufficient data for each item which can lead to data sparsity (Zhang, 2020). Lastly, the evaluation of the performance of the recommender system is complex as there is no set of benchmarks to determine whether the recommendations are appropriate (Dwivedi, 2020).

3. Description

3.1. Dataset

Since this project aims to build a recommender system based on Amazon users' reviews, the dataset is retrieved from the Amazon Review dataset 2018 (Ni, 2018). Instead of using the whole set of data, data from the Amazon Fashion category is mainly used for model training and evaluation in this project. The data is composed of two major parts:

<u>I) The Product metadata</u>
This dataset includes information about 186637 Amazon Fashion products, including the product title, Product IDs, descriptions, categories, prices, brands, image details, and more.

Figure 1. Dataset of Product metadata

title	brand	feature	rank	date	asin	imageURL	imageURLHighRes	description	price	a
Slime Time Fall Fest [With CDROM and Collector...	Group Publishing (CO)	[Product Dimensions:\n \n8....	13,052,976inClothing,Shoesamp;Jewelry(	8.70 inches	0764443682	[https://images-na.ssl-images-amazon.com/image...	[https://images-na.ssl-images-amazon.com/image...	NaN	NaN	
XCC Qi promise new spider snake preparing men'...	NaN	NaN	11,654,581inClothing,Shoesamp;Jewelry(	5 star	1291691480	[https://images-na.ssl-images-amazon.com/image...	[https://images-na.ssl-images-amazon.com/image...	NaN	NaN	

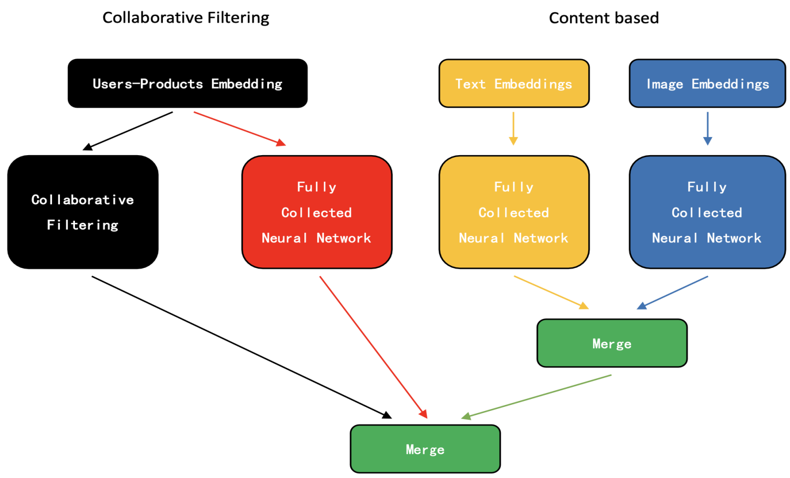
<u>II) User reviews</u>
This dataset includes 883636 records of users' reviews on Amazon Fashion products, comprising the users' ratings, review time, user IDs, review texts, and more.

Figure 2. Dataset of User reviews

overall	verified	reviewTime	reviewerID	asin	reviewerName	reviewText	summary	unixReviewTime	vote	style	image
5	True	10 20, 2014	A1D4G1SNUZWQOT	7106116521	Tracy	Exactly what I needed.	perfect replacements!!	1413763200	NaN	NaN	NaN
2	True	09 28, 2014	A3DDWDH9PX2YX2	7106116521	Sonja Lau	I agree with the other review, the opening is ...	I agree with the other review, the opening is ...	1411862400	3.0	NaN	NaN
4	False	08 25, 2014	A2MWC41EW7XL15	7106116521	Kathleen	Love these... I am going to order another pack	My New 'Friends' !!	1408924800	NaN	NaN	NaN

3.2. The Hybrid model

Figure 3. Overview of the Context Aware Hybrid model



The model built in this project is the Context-Aware Hybrid Model (Figure 3). It is a type of recommender system that combines the power of the traditional Neural Collaborative Filtering network with Content-Based Filtering to provide better recommendations.

The main idea behind the model is to consider both the user's preferences and the context in which the recommendations are being made. The main advantage of this model is that it is highly flexible in the sense that it incorporates a wide range of contextual information, such as product titles, product features, product images, user reviews, user ratings, review time, and more, to generate personalized and relevant recommendations for users. This project will mainly use text and image data as input.

4. Data Analysis

<u>I) Distribution of users' ratings</u>

Figure 3 shows the dataset's overall distribution of the users' ratings. It was found that more than half of the users rated 5, while other ratings only occupied about 7% to 17%. The users' ratings were also grouped by three types of sentiment: positive (rating 4-5), neutral (rating 3), and negative (rating 1-2), and it was found that most customers.

Figure 4. Distribution of the users' ratings

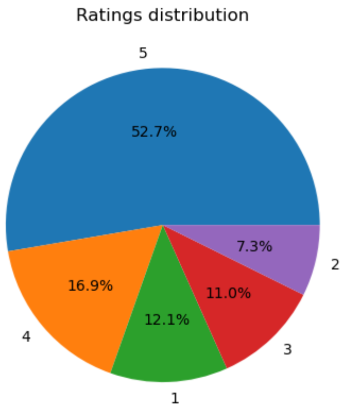
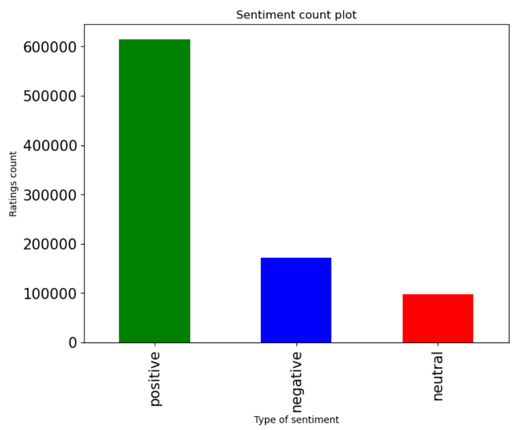


Figure 5. Distribution of sentiment



<u>II) Word cloud of users' review text</u>

Figure 5 below shows the word clouds of the users' review text. For the positive comments (left), some commonly used words include love, comfortable, perfect, and more. For the negative comments (right), some commonly used words include small, disappointed, returned, and more.

Figure 6. Word clouds of users' review text

III) Product images visualization

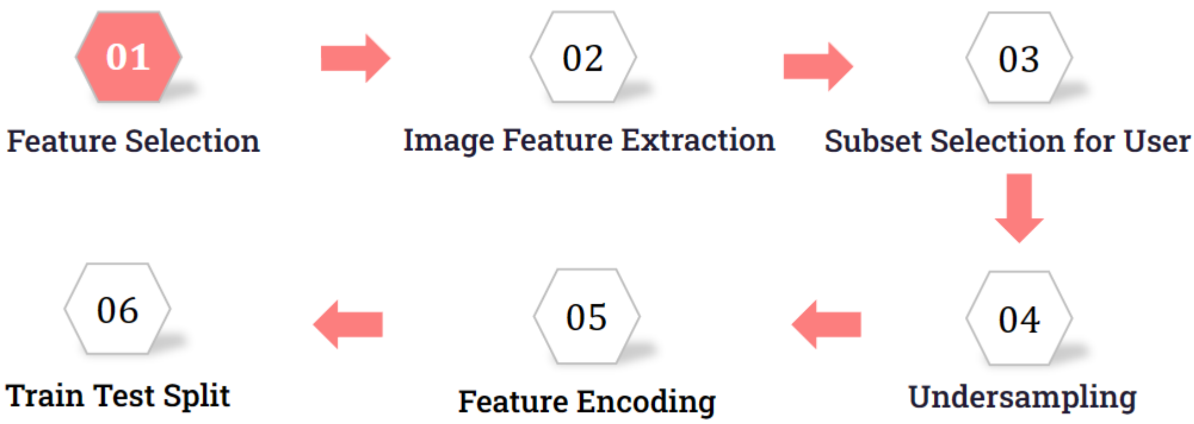
In total, 131,810 product images were downloaded from the URL. After doing some random image visualizations, it is observed that the images generally involve different types of Fashion products, such as T-shirts, dresses, shoes, accessories like bracelets, and the size chart of the products is also included.

Figure 7. Visualization of Product images

5. Implementation details

5.1. Data preprocessing

Figure 8. Visualization of Data Preprocessing



I) Feature Selection

The first step in the data preprocessing pipeline involves choosing the relevant features for Collaborative Filtering and Content-based Filtering. The user and product are selected as input for Collaborative Filtering, while the Product title and image are selected as input for Content-based Filtering.

II) Image Feature Extraction
The next step involves resizing the product images to (32, 32) dimensions and using the Efficientnetv2 model to extract image features. This step helps create a numerical representation of the product images that can be used in machine learning algorithms.

III) Subset Selection for User
Users with only one review are removed from the dataset to improve the data quality and reduce noise, ensuring that the data is more representative of the user population.

IV) Undersampling
The dataset had a significant disproportion in 5-star ratings compared to other rating labels. The 5-star ratings were undersampled to address this imbalance. This step helped balance the classes and improve the performance of machine learning models.

V) Feature encoding
The features were then encoded using different techniques. The reviewer ID and product ID were encoded using LabelEncoder. The product title was encoded using TfidfVectorizer. The product image was encoded using a pre-trained Efficientnetv2 model and converted the categorical features into numerical representations suitable for machine learning models.

VI) Train-test split
After preprocessing, the data was split into 80% for training and 20% for testing. This split helped evaluate the models' performance and generalizability to new data.

5.2. Hybrid recommendation model

Zhang et al. (2017) propose a joint representation learning approach for a recommendation that integrates heterogeneous information sources, including user-item interactions, user profiles, and item attributes. We refer to that idea – integrating heterogeneous information sources and building a hybrid recommendation model that integrates user-item interactions and item attributes. The hybrid model implemented in this project combines a neural matrix factorization model and content-based models. Chakrabarti and Das (2019) propose a general framework for Neural network-based Collaborative Filtering. The architecture of the neural matrix factorization model we implemented in the hybrid model is shown in Figure 9. It refers to the framework for the model of Chakrabarti and Das.

Figure 9. Neural matrix factorization model

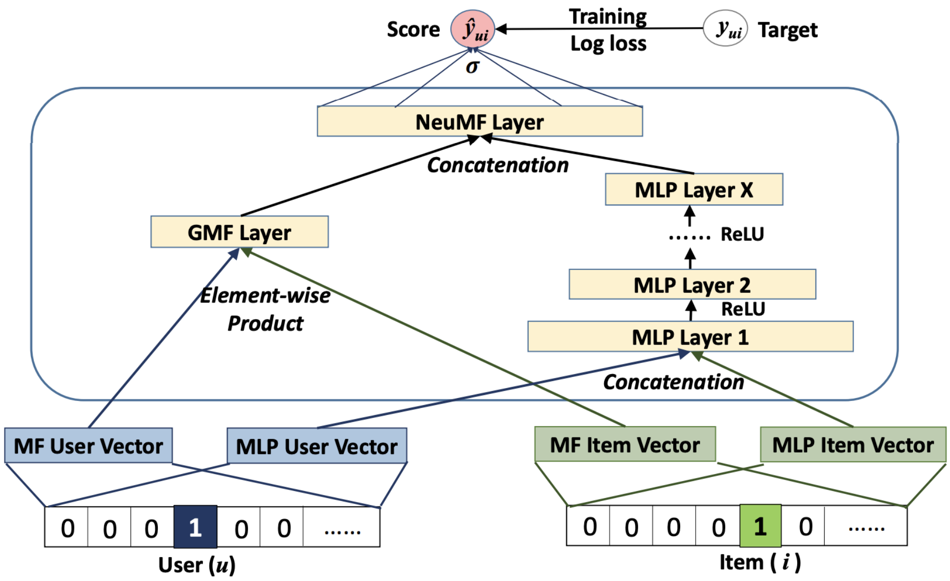
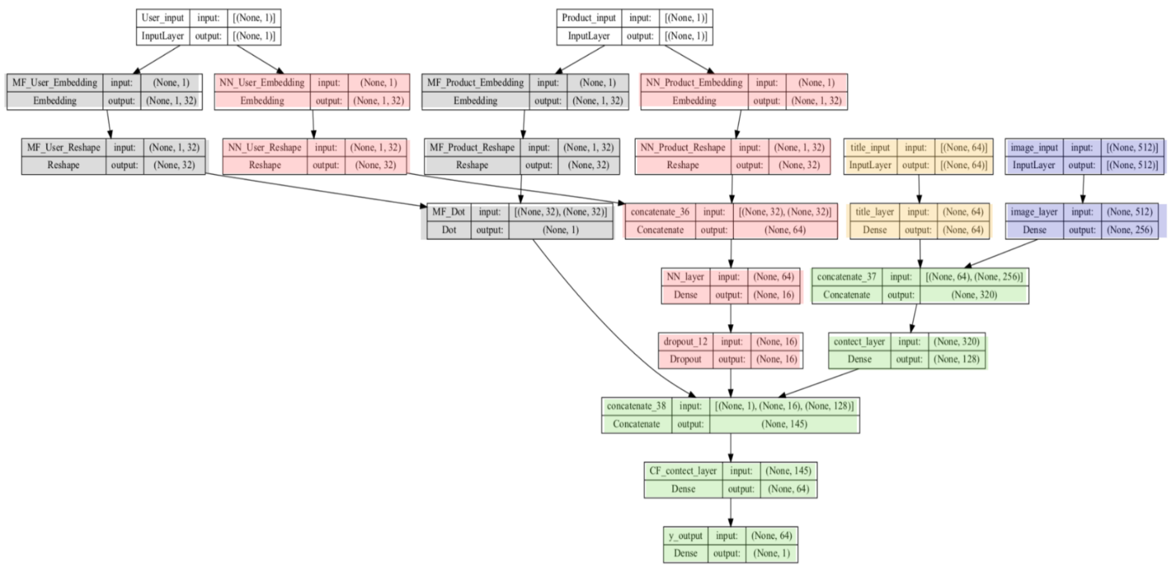


Figure 10. Architecture of the hybrid model



The hybrid model is used for recommendation systems to predict a user's preference for a particular Amazon fashion product based on past interactions with other products. The model inputs the user, product ID, text, and image. The model consists of four main components, as shown in Figure 10.

I) Matrix Factorization Component (Black)

Matrix Factorization Componen uses matrix factorization to learn user and product embeddings. The embeddings are then reshaped and fed into a dot product layer, which predicts the user's preference for a particular product. The dot product layer is normalized to improve the model's performance.

II) Neural Network Component (Red)
This component uses neural networks to learn additional representations of users and products. The embeddings are again reshaped and concatenated before being fed into a dense layer with 16 units and a ReLU activation function. A dropout layer is added to prevent overfitting.

III) Text-Based Component (Yellow)
The model includes a text-based component that inputs the product's title. The text is fed into a dense layer with 64 units and a ReLU activation function.

IV) Image-Based Component (Blue)
The model includes an image-based component that inputs the product's image. The image is flattened into a 1D vector and fed into a dense layer with 256 units and a ReLU activation function.

The outputs of the three components are concatenated and fed into a dense layer with 128 units and a ReLU activation function. The final output is a single scalar value that predicts the user's preference for the product. In the training stage, the model is trained using the Adam optimizer with a learning rate of 0.01 and a decay of 0.0001. The loss function used is mean squared error (MSE), and the metrics used for evaluation are root mean square error (RMSE) and mean absolute error (MAE).

6. Results and Observations

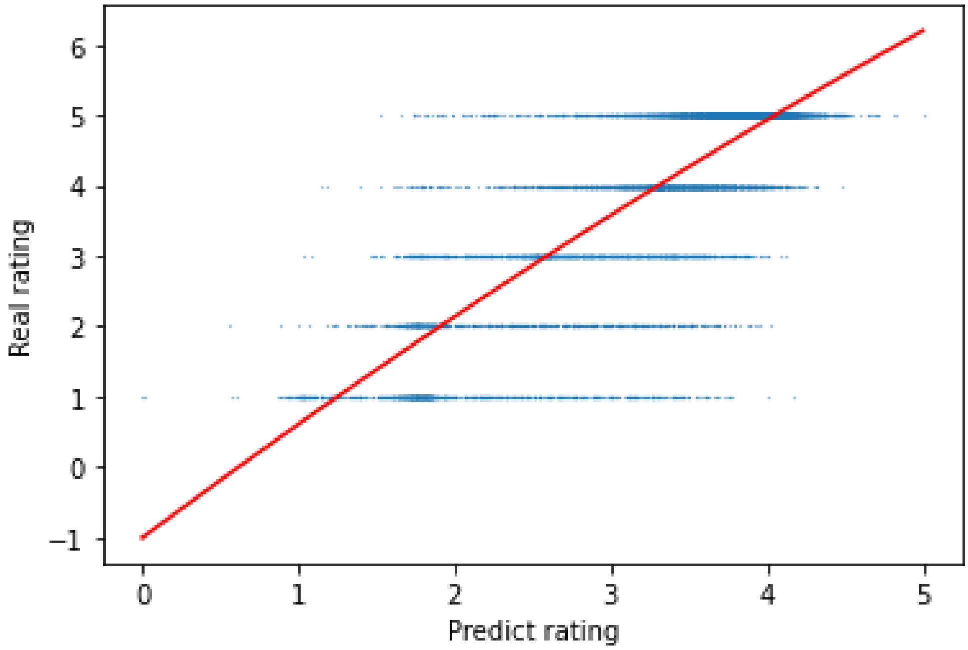
We employ the Pearson correlation coefficient, RMSE, and MAE to evaluate the hybrid recommendation model's effectiveness. Pearson correlation coefficient takes into account the variability of the data. RMSE and MAE provide an intuitive measure of the model's performance in predicting user preferences.

6.1. Pearson Correlation Coefficient

The Pearson correlation coefficient is commonly used in recommendation systems, as seen in Equation 1, which measures the linear correlation between the predicted and actual ratings. It considers that some users may give higher or lower ratings in general and adjusts for this in calculating the correlation coefficient.

$$r = \frac{\sum(x-m_x)\ (y-m_y)}{\sqrt{\sum(x-m_x)^2\ (y-m_y)^2}} \tag{1}$$

Figure 11. Visualization of predicting rating and true rating



After calculating the Pearson correlation coefficient, we obtained a 0.82009 correlation coefficient. It indicated a strong positive correlation between predicted and real ratings. Also, the p-value is extremely small, indicating that the correlation is statistically significant. As shown in Figure 11, we can predict ratings correctly with fewer errors.

6.2. Root Mean Square Error & Mean Absolute Error

RMSE measures the average squared difference between the predicted and actual ratings, as seen in Equation 2. It gives more weight to significant errors and is sensitive to outliers.

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

MAE measures the average absolute difference between the predicted and actual ratings, as seen in Equation 3. It gives equal weight to all errors and is less sensitive to outliers.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i| \tag{3}$$

After model training and prediction, we obtained that the RMSE is 0.75259 and the MAE is 0.51566. The Test RMSE of 0.75259 and Test MAE of 0.51566 indicate that the hybrid model performs reasonably well in predicting user preferences for items in the test set. The RMSE value indicates that, on average, the predicted ratings are off by 0.75259 units from the actual ratings. In contrast, the MAE value indicates that the average absolute difference between the predicted and actual ratings is 0.51566 units.

6.3. Convergence Performance

The hybrid model converges at around 10 epochs, which means that the model has learned to make accurate predictions on the training data after approximately five passes through the training set, as shown in Figure 12 and Figure 13. They indicate that the model is learning quickly and efficiently.

Figure 12. Convergence performance for training

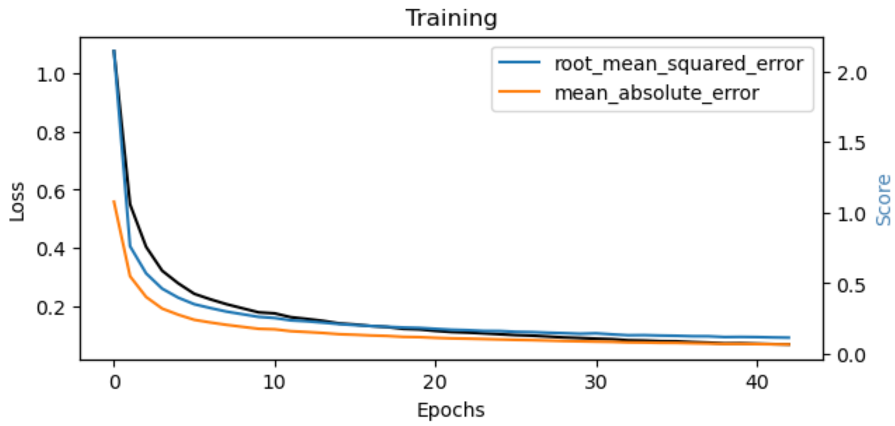
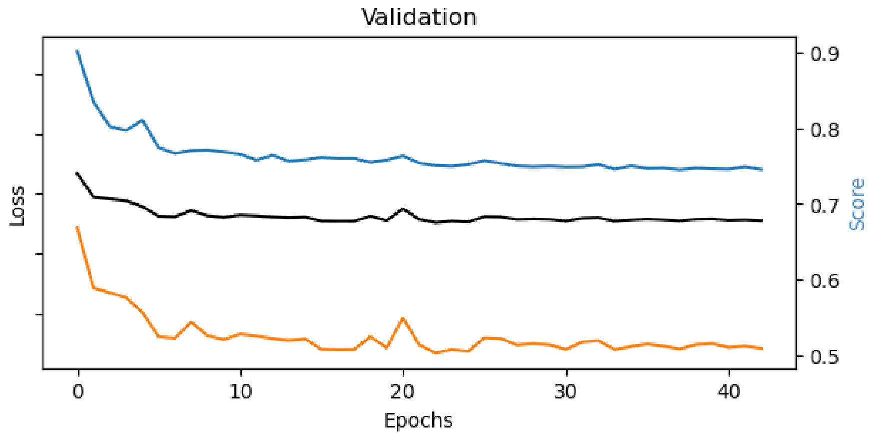


Figure 13. Convergence performance for testing



7. Discussions

A recommender system is a powerful tool that helps users make informed decisions and assists product providers in promoting their products. However, as with many technologies, it has restrictions and challenges. In order to overcome these challenges and improve the overall effectiveness of the recommender systems, there are several possible improvements for future development.

The first challenge recommender systems face is the potential for recommending inappropriate or unsuitable content, which can be handled by adding a rule-based or data-based filtering layer. This layer would prevent the recommendation of content that is considered inappropriate or offensive.

Another challenge is the cold-start issue, where a new user or a new item has no history in the system, making it difficult for the recommender system to make precise predictions. One approach to this problem is to use strategies like clustering to a group of users or items with similar characteristics, enabling the system to make more authentic recommendations even with limited or restricted data.

Finally, integrating sentiment analysis with a recommender system could yield an advantageous approach to improve its effectiveness. It allows the system to consider the user's past behavior and preferences with their present mood and sentiment. For example, suppose a user frequently posts positive comments about vintage fashion. In that case, the recommender system can use sentiment analysis to determine this preference and recommend more vintage-style clothing items. Additionally, sentiment analysis can identify emerging fashion trends and recommend suitable and relevant products to users.

Overall, the future development of recommender systems will require a continued focus on addressing these challenges to improve the accuracy and effectiveness of recommendations. By incorporating new techniques and approaches, we can continue to enhance the power of recommender systems to assist users in making informed decisions and help product providers promote their goods.

8. Conclusion

In conclusion, this project aimed to develop a hybrid recommender system that combines the capability of the traditional Neural Collaborative Filtering network with Content-Based Filtering to provide personalized recommendations for users. We used a dataset of user ratings on fashion and implemented a neural network model that combines both approaches.

The evaluation results show that the hybrid recommender system portrays well in predicting user preferences with a strong positive correlation between predicted and actual ratings with reasonably low RMSE and MAE values. Further, the model converges quickly, indicating efficient learning on the training set.

However, the recommender system still faces several challenges, such as the cold-start problem and the risk of recommending inappropriate content. We suggested adding a filtering layer and applying clustering techniques to address these challenges. We also proposed combining sentiment analysis in the recommender system with emerging fashion trends as an example.

In summary, this project provides an advantageous strategy to improve recommendation systems' accuracy and enhance user experience. The forthcoming development of recommender systems should handle these challenges and incorporate emerging technologies to provide even better-personalized recommendations.

References

Amazon. (2020, September 18). *Who we are*. US About Amazon. Retrieved from https://www.aboutamazon.com/about-us

Burke, R., Felfernig, A., & Göker, M. H. (n.d.). *Recommender Systems: An overview*. AI Magazine. Retrieved from https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/2361

Chakrabarti, N., & Das, S. (2019). Neural Networks and collaborative filtering. 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN).https://doi.org/10.1109/icscan.2019.8878693

Dang, C. N., Moreno-García, M. N., & Prieta, F. D. la. (2021). An Approach to Integrating Sentiment Analysis into Recommender Systems. Sensors, 21(16), 5666. https://doi.org/10.3390/s21165666

Dong, Z., Wang, Z., Xu, J., Tang, R., & Wen, J. (2022, September 5). *A brief history of Recommender Systems*. arXiv.org. Retrieved from https://arxiv.org/abs/2209.01860

Dwivedi, R., Anand, A., Johri, P., Banerji , A., & Gaur, N. K. (2020, June). Product based recommendation system on Amazon Data . Retrieved from https://www.researchgate.net/publication/352815845_Product_Based_Recommendation_System_On_Amazon_Data

Ni, J. (n.d.). *Amazon Review Data (2018)*. Retrieved from https://nijianmo.github.io/amazon/index.html

Panteli, A., & Boutsinas, B. (2023). Addressing the Cold-Start Problem in Recommender Systems Based on Frequent Patterns. Algorithms, 16(4), 182. https://doi.org/10.3390/a16040182

Tang, T. Y., & Winoto, P. (2015). I should not recommend it to you even if you will like it: The ethics of recommender systems. New Review of Hypermedia and Multimedia, 22(1-2), 111–138. https://doi.org/10.1080/13614568.2015.1052099

Zhang, F., Qi, S., Liu, Q., Mao, M., & Zeng, A. (2020). Alleviating the data sparsity problem of recommender systems by clustering nodes in bipartite networks. *Expert Systems with Applications*, *149*, 113346. https://doi.org/10.1016/j.eswa.2020.113346

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep Learning based Recommender System. ACM Computing Surveys, 52(1), 1–38. https://doi.org/10.1145/3285029

Zhang, Y., Ai, Q., Chen, X., & Croft, W. B. (2017). Joint Representation Learning for top-n recommendation with heterogeneous information sources. Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. https://doi.org/10.1145/3132847.3132892

Zhou, W., Wen, J., Qu, Q., Zeng, J., & Cheng, T. (2018). Shilling attack detection for recommender systems based on credibility of group users and Rating Time Series. *PLOS ONE*, *13*(5). https://doi.org/10.1371/journal.pone.0196533