IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 3, October 2024

Exploring the Efficiency of Hybrid Recommender Systems Implemented with TensorFlow Framework

Tanveer Ahmad Lone[1], Dr. Ajit Kumar[2] and Dr. Muzafar Rasool Bhat[3]
Research Scholar, Computer Science, Shri Jagdishprasad Jhabarmal Tibrewala University, Jhunjhunu, Rajasthan, India[1]
Professor, Computer Science, Shri Jagdishprasad Jhabarmal Tibrewala University, Jhunjhunu, Rajasthan, India[2]
Assistant Professor, Computer Science,
Islamic University of Science and Technology, Awantipora, Pulwama, J&K, India[3].

Abstract*: In recent years, the field of recommendation systems has seen significant advancement with the introduction of hybrid approaches. These systems combine the strengths of multiple recommendation techniques to provide more accurate and diverse recommendations to users. In this research, we propose and evaluate the effectiveness of a hybrid recommender system that utilizes TensorFlow, an open-source machine learning framework, to implement the system. The proposed system combines both collaborative and content-based methods to remove the cold start problem and make personalized recommendations that can recommend similar movies to the users based on features extracted by the model. The results of our experiments demonstrate that the proposed hybrid system outperforms traditional singular methods and can be effectively implemented using TensorFlow. This research provides insights into the potential of TensorFlow for building efficient hybrid recommendation systems and the benefits of combining multiple recommendation techniques.*

Keywords: TensorFlow, Recommender System, MovieLens, Neural CF, Hybrid

I. INTRODUCTION

Recommender systems are tools that help users navigate the overwhelming amount of information available by identifying and presenting information that is relevant to their preferences, interests, or past behavior. These systems use user profiles to predict which items a user would be interested in and filter out irrelevant information [1]. They are designed to help alleviate the problem of information overload by providing users with a tailored selection of content [2].

In recent years, recommender systems (RS) have become a popular subject of discussion as they provide users with relevant information based on their needs or preferences. RS needs to be fair and unbiased, recommending items to all groups of people without giving preference to any particular group [3]. Recommendation systems have become an integral part of various applications, such as e-commerce [4], entertainment [5], and social media [6]. The goal of a recommendation system is to provide personalized recommendations to users based on their past behavior and preferences [7]. In recent years, there has been a shift towards hybrid recommendation systems, which combine the strengths of multiple recommendation techniques to provide more accurate and diverse recommendations [8]. Recommender systems gained popularity in recent years due to their increased accuracy, leaving behind the question of biased and unfair recommendations. These threats lead us to use deep models to make fair recommendations, which will dominate future recommendations [9]. One of the challenges in building recommendation systems is the need for efficient and scalable algorithms. TensorFlow, an open-source machine learning framework, has gained popularity in recent years due to its ease of use and ability to handle large-scale datasets [10]. In this research, we propose and evaluate the effectiveness of a hybrid recommender system that utilizes TensorFlow to implement the system. The proposed system combines both collaborative and content-based methods to make hybrid-based custom recommendations. Hybrid recommendation systems have been widely studied in recent years. Researchers have proposed various methods to combine different types of recommendation techniques, such as integrating collaborative

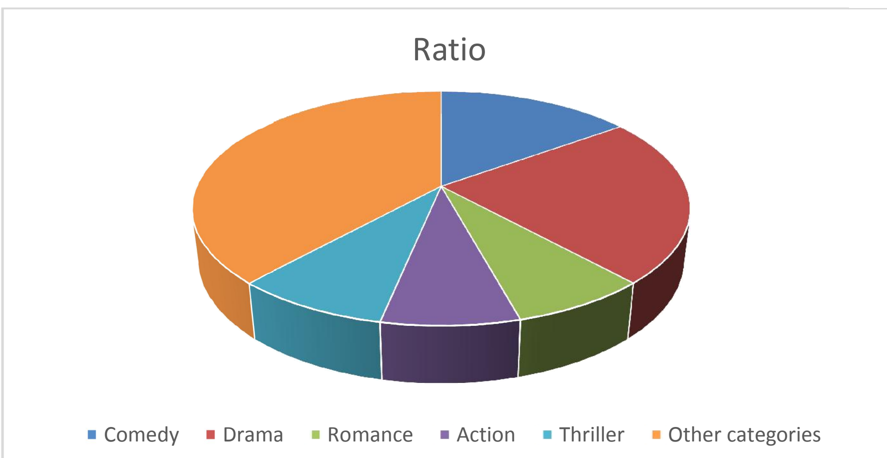
IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 3, October 2024

filtering with content-based filtering [11] or using ensemble methods to combine multiple recommendation algorithms [12]. Big data makes it difficult for users to select content of interest, such as which movie to watch, which is a popular question among streaming entertainment users. Because of the numerous types of movies, clips, and videos available on the internet, people are perplexed. Researchers use different types of methods to improve recommendation quality. The use of TensorFlow in recommendation systems has also been explored by several researchers to give more accurate recommendations using different techniques. Seq2Seq models have been used by [13] to characterize the interests of users and then integrate with sorting algorithms to make personalized movie recommendations. Combining collaborative filtering with demographic information, deep learning, multiple kernel-based, and feature-based content filtering [14] [15] [16] [17] has been proposed to make hybrid models for more accurate, robust, and personalized news recommendations. Music recommendation models using deep learning [18], news recommendation models using deep learning [19], and movie recommendation models using deep learning [20] have pivotal importance and are the target research topics, as deep learning opened the way to making recommendation models robust and more accurate.

II. METHODOLOGY

Dataset

For the experimentation, we used the movieLens dataset [21] which contains movie ratings and metadata collected by the MovieLens online movie recommendation service. The dataset includes information about users, movies, and ratings. The dataset is widely used for the development and testing of recommender systems. The dataset is available in different sizes, including 100,000 ratings from 943 users on 1682 movies, 1 million ratings from 6000 users on 4000 movies, and 20 million ratings from 138,493 users on 27,278 movies. The dataset also includes additional information about the movies, such as their genre, release year, and IMDb URL. The dataset is commonly used in academic research on recommender systems, as well as in the industry for developing and testing new recommendation algorithms. It is also used as benchmark data to evaluate the performance of different recommendation algorithms [22]. The dataset is available for download from the MovieLens website, which is maintained by the GroupLens Research group at the University of Minnesota. The Fig. below shows the ratio of different genera present in the dataset we used for experimentation.



Implementation

Our experimentation is based on the idea of fair and unbiased recommendations. Many issues can be solved, like sparsity (difficulty in finding reliable and sufficient users with similar features), scalability, accuracy, and the cold start problem [23]. Initially, our experimentation focused on the cold start problem by using content-based filtering [24] to make the initial recommendations and then making the refined recommendations using collaborative filtering [25] by focusing on feedback from the users. On the other hand, we used deep learning as a tool to accommodate the same issue in the TensorFlow framework. Traditionally, we calculated the weighted average of movies to find similarity and then merged the result with the popularity of the movie to get a score. Now cosine similarity is calculated between each

DOI: 10.48175/IJARSCT-19979

movie, and finally, score and cosine similarity values are used to calculate the final score of the movie. The equation to calculate the weighted average is mentioned below:

$$Ws = (Ag \,.\, v + Vt \,.\, m) \,/\, v + m$$

Where Ws is the rating with weight, Ag is the average for a movie, v is the number of votes for the movie, m is the minimum number of votes required to be on the top list, and Vt is the mean of votes. Using the TensorFlow Recommenders library, we used the pre-built deep learning model Neural CF [26] to recommend movies. The model takes in four parameters: embedding dim, item dim, user dim, and dense dim. The embedding dim parameter is the size of the embedding vectors used to represent items and users. The dense dim parameter is the size of the dense layers in the model. The user dim parameter is the size of the vocabulary used to represent users, and the item dim is the size of the vocabulary used to represent items. Our model comprises three dense layers with relu [27] as an activation function and performed better while using the Adagrad optimizer [28], with shuffle as True, loss function as MeanSquaredError, and metrics as RootMeanSquaredError. We then fit the model to the training data for 20 epochs with a batch size of 64. After that, we evaluate the model on the test set to check the movie recommendations for unseen user IDs.

Software, Hardware, and Frameworks used in the study

A. Software

Python: Python is a popular programming language that can be used to build a hybrid recommendation system.

B. Libraries and frameworks in Python for the experimentation

Pandas, Seaborn, NumPy, Matplotlib, Wordcloud, Sklearn, Neural CF, and TensorFlow
TensorFlow: TensorFlow comes with a library called TensorFlow Recommenders (TFRS) (https://www.tensorflow.org/recommenders) for building a recommender system. It's built on Keras and aims to have a gentle learning curve while still giving you the flexibility to build complex models.

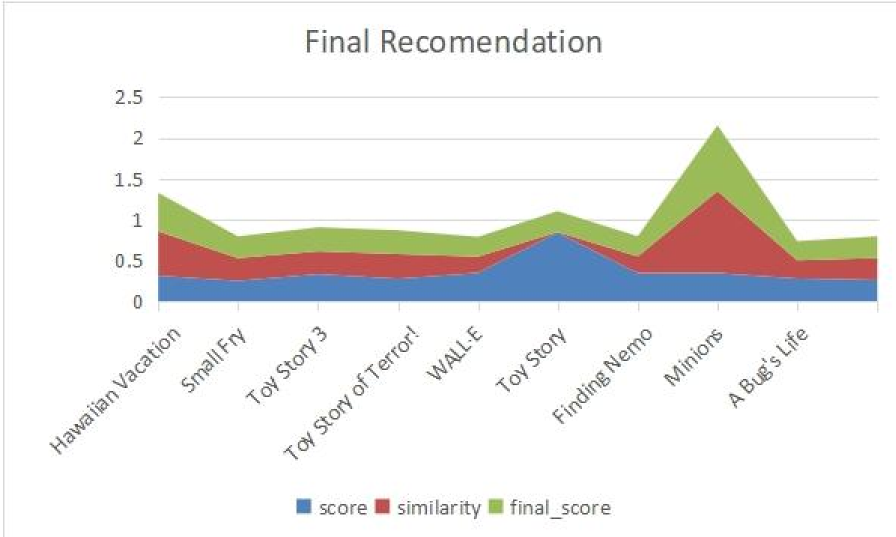
C. Hardware used

GPU: NVIDIA Tesla V100 32 GB
Total number of CUDA Cores : 15,360
Performance: 2.976 TFLOPS
Storage: 2 x 960 GB SSD SATA
RAM: 8 x 64 GB
Software: CentOS

III. RESULTS AND DISCUSSION

The proposed hybrid recommender system in this research combines both collaborative and content-based methods to make personalized recommendations. The results of the experiments demonstrate that the proposed system outperforms traditional singular methods in terms of accuracy and diversity of recommendations. This suggests that combining multiple recommendation techniques can provide more accurate and diverse recommendations to users using deep learning. One of the key advantages of the proposed system is its use of TensorFlow, an open-source machine learning framework. TensorFlow allows for easy implementation of neural networks, which are a key component of the proposed system. Additionally, TensorFlow is known for its scalability, which is crucial for handling large-scale datasets. This makes it a suitable tool for building efficient and scalable recommendation systems. The fig. below shows the final score of the movies using a hybrid system, predicting Toy Story in particular.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 3, October 2024



However, it's worth noting that the proposed system is based on a specific dataset, and it's not clear how well it would generalize to other datasets or real-world scenarios. Additionally, the proposed system was evaluated using certain evaluation metrics, such as accuracy and diversity of recommendations, but there may be other aspects of a recommendation system that are important to consider, such as diversity, novelty, and serendipity, which were not evaluated in this research.

In the future, it would be interesting to explore how well the proposed system generalizes to other datasets, and how it performs on evaluation metrics. Additionally, it would be interesting to explore other hybrid recommendation techniques, such as ensemble methods, and compare the performance of these methods with the proposed system. Another interesting direction would be to explore how to incorporate other types of information, such as social network data, into the system to make more personalized recommendations.

IV. CONCLUSION

In conclusion, this research proposed and evaluated the effectiveness of a hybrid recommender system that utilizes TensorFlow, an open-source machine learning framework, to implement the system. The proposed system combines both collaborative and content-based methods to make personalized recommendations. The results of the experiments demonstrate that the proposed hybrid system outperforms traditional singular methods in terms of accuracy and diversity of recommendations. Additionally, TensorFlow proves to be a suitable tool for implementing the hybrid system, as it allows for easy implementation of neural networks. This research provides insights into the potential of TensorFlow for building efficient hybrid recommendation systems and the benefits of combining multiple recommendation techniques. It can also be used as a guide for practitioners looking to implement hybrid recommender systems using TensorFlow. The proposed system demonstrates the potential of combining the strengths of multiple recommendation techniques to provide more accurate and diverse recommendations to users. As hybrid recommendation systems become more popular in various applications, this research can serve as a valuable resource for practitioners looking to build efficient and scalable hybrid recommendation systems using TensorFlow.

REFERENCES

[1]. C. Pan, W. Li Research paper recommendation with topic analysis In Computer Design and Applications IEEE, 4 (2010)

[2]. J.A. Konstan, J. Riedl Recommender systems: from algorithms to user experience User Model User-Adapt Interact, 22 (2012), pp. 101-123

[3]. Aalam, S. W., Ahanger, A. B., Bhat, M. R., & Assad, A. (2022). Evaluation of Fairness in Recommender Systems: A Review. In International Conference on Emerging Technologies in Computer Engineering (pp. 456-465). Springer, Cham.

[4]. Schafer, J. B., Konstan, J., & Riedl, J. (1999, November). Recommender systems in e-commerce. In Proceedings of the 1st ACM conference on Electronic commerce (pp. 158-166).

[5]. S. Halder, M. H. Seddiqui and Y. -K. Lee, "An entertainment recommendation system using the dynamics of user behavior over time," *2014 17th International Conference on Computer and Information Technology (ICCIT)*, Dhaka, Bangladesh, 2014, pp. 41-46, doi: 10.1109/ICCITechn.2014.7073094

[6]. A Anandhan, L. Shuib, M. A. Ismail and G. Mujtaba, "Social Media Recommender Systems: Review and Open Research Issues," in *IEEE Access*, vol. 6, pp. 15608-15628, 2018, doi: 10.1109/ACCESS.2018.2810062.

[7]. Abinash Pujahari, Dilip Singh Sisodia, Aggregation of preference relations to enhance the ranking quality of collaborative filtering based group recommender system, Expert Systems with Applications, Volume 56, 2020, 113476, ISSN 0957-4174,

[8]. Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction, 12(4), 331-370.

[9]. Ahanger, A. B., Aalam, S. W., Bhat, M. R., & Assad, A. (2022). Popularity Bias in Recommender Systems-A Review. In International Conference on Emerging Technologies in Computer Engineering (pp. 431-444). Springer, Cham.

[10]. Anandaraj, A., Yeshwanth Ram, P., Sri Ram Kumar, K., Revanth, M., & Praveen, R. (2021). Book Recommendation System with TensorFlow. 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS).

[11]. Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6), 734-749.

[12]. Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. Knowledge-Based Systems, 46, 109-132.

[13]. Wang, Y., Li, Y., & Li, X. (2018). Personalized movie recommendation using deep learning. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 990-997). IEEE.

[14]. Wang, Y., Li, Y., & Li, X. (2019). Hybrid recommendation system based on collaborative filtering and deep learning. In 2019 IEEE International Conference on Big Data (Big Data) (pp. 5358-5365). IEEE.

[15]. Burke, R. (2002). Hybrid web recommender systems. Proceedings of the 15th international conference on World Wide Web-Volume 2, 665-672.

[16]. Karatzoglou, A., Amatriain, X., Baltrunas, L., & Oliver, N. (2010). Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. Proceedings of the fourth ACM conference on Recommender systems, 79-86.

[17]. Liu, Y., Li, H., Li, S., & Yang, Q. (2012). Personalized recommendation with feature-based and collaborative-based hybrid models. Journal of Zhejiang University SCIENCE C, 13(6), 453-465.

[18]. Cheng, H., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., ... & Bickson, D. (2016). Wide & deep learning for recommender systems. Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, 7-10.

[19]. Guo, H., Tang, R., Liu, Y., & Liu, Y. (2017). Deep interest network for click-through rate prediction. Proceedings of the ADKDD'17, 135-144.

[20]. Li, Y., Li, X., & Wang, Y. (2020). Deep learning-based recommendation system with user-item interaction data and auxiliary information. Future Generation Computer Systems, 114, 19-29.

[21]. Harper, F. M., & Konstan, J. A. (2015). The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, *5*(4), 1-19.

[22]. Tousch, A. M. (2019). How robust is MovieLens? A dataset analysis for recommender systems. *arXiv preprint arXiv:1909.12799*.

[23]. Lika, B., Kolomvatsos, K., & Hadjiefthymiades, S. (2014). Facing the cold start problem in recommender systems. *Expert systems with applications*, *41*(4), 2065-2073.

[24]. Van Meteren, R., & Van Someren, M. (2000, May). Using content-based filtering for recommendation. In *Proceedings of the machine learning in the new information age: MLnet/ECML2000 workshop* (Vol. 30, pp. 47-56).

[25]. Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web* (pp. 291-324). Springer, Berlin, Heidelberg.

[26]. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017, April). Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web* (pp. 173-182).

[27]. Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

[28]. Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, *12*(7).