International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 04 | April - 2022 Impact Factor: 7.185 ISSN: 2582-3930

CHATBOT MOVIE RECOMMENDER SYSTEM

Akarshita Singh[1], Adarsh Tripathi[2], Ashutosh Mishra[3]

[1]*BTECH(pursuing), Department of Computer Science Institute of Technology and Management(ITM), Gida, Gorakhpur, U.P, 273001, India*

[2] *BTECH(pursuing), Department of Computer Science Institute of Technology and Management(ITM), Gida, Gorakhpur, U.P, 273001, India*

[3] *BTECH(pursuing), Department of Computer Science Institute of Technology and Management(ITM), Gida, Gorakhpur, U.P, 273001, India*

Abstract – This paper deals with the recommender engine and chatbot through python using machine learning, artificial intelligence, to assist movie buffs by recommending what film to watch without requiring them to go through the timeconsuming and complex process of selecting from a vast number of films ranging from thousands to millions. Our goal in this post is to decrease human effort by recommending movies based on the user's preferences. To address these issues, we developed a paradigm that combines both a content-based and a collaborative approach. When compared to other systems that use a content-based approach, it will provide more explicit results. People are limited by content-based recommendation systems; these algorithms do not prescribe things out of the box, restricting your ability to learn more. As a result, when used in a chatbot, it produces reliable results.

Key Words: Machine Learning, Collaborative filtering, similarity, user, Content Based, recommender system.

1. INTRODUCTION

A recommendation system, often known as a recommendation engine, is a paradigm for information filtering that attempts to forecast a user's preferences and provide suggestions based on those choices. These systems have grown in popularity in recent years, and they are now frequently employed in fields such as movies, music, books, videos, apparel, restaurants, food, locations, and other services. These systems gather data on a user's preferences and behaviour, which they then utilise to enhance their recommendations in the future.

The two forms of collaborative recommender systems are further explained in Figure 1. Pure CF is a term used to describe this sort of filtering.

A. Use-based filtering - In the subject of building customized systems, user-based preferences are quite common. If the user's preferences are evaluated historically, this method presupposes that they are not random.

B. Item-based filtering - Item-based filtering, unlike user-based filtering, focuses on the similarity between the things individuals prefer rather than the users themselves. The comparable items are calculated in advance. The things that are comparable to the target item are then recommended to the user.

.

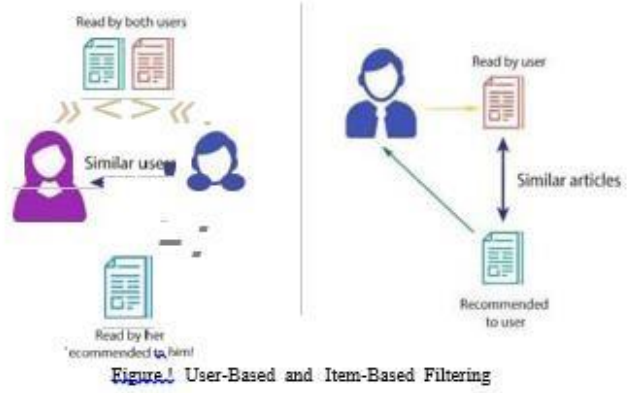

Figure.1 User-Based and Item-Based Filtering

The paper propose a collaborative recommendation system based on the Map Reduce architecture and built to function on the Hadoop platform. The authors built this system using the set-similarity join approach, which incorporates both userbased and item-based collaborative filtering strategies.

The authors suggested a movie recommendation system based on collaborative filtering that relies on user ratings to deliver recommendations. To arrange the movies according to their ratings, the suggested method employs the K-means algorithm.

2. Literature Review

Kumar presented MOVREC, a collaborative filtering-based movie recommendation system. Collaborative filtering gathers data from all users and creates suggestions based on it. Virk have proposed a hybrid system. This system combines contentbased and collaborative methods. De Campos compared and contrasted both classic recommendation methods. Because both of these systems have flaws, he presented a hybrid system that combines Bayesian networks with collaborative procedures. Kuelewska suggested clustering as a method for dealing with suggestions. The centroid-based solution and memory-based approaches for clustering were investigated. As a consequence, precise suggestions were created. Chiru

presented Movie Recommender, a system that generates suggestions based on the user's history. In their work, Sharma and Maan looked at several strategies for making recommendations, including collaborative, hybrid, and content-based suggestions. It also discusses the advantages and disadvantages of different techniques. An inductive learning method was introduced by Li and Yamada.

2.1 Materials and Methods

In this project, mainly uses machine learning using deep learning approaches/algorithm such as collaborative filtering matrix factorization and deep learning.

2.1.1 Generating Dataset

The movies dataset that we utilized in our tests came from Yahoo Research Web scope. Yahoo! Movies User Ratings and Yahoo! Descriptive Content Information, v.l.O. are two files provided by the database. User ID, Movie ID, and Ratings make up the 211231 entries in the Yahoo! Movies Users Ratings file. The 54058 entries in the Yahoo! Movies Descriptive Content Information file include Movie ID, Title, Genre, Directors, Actors, and so on.

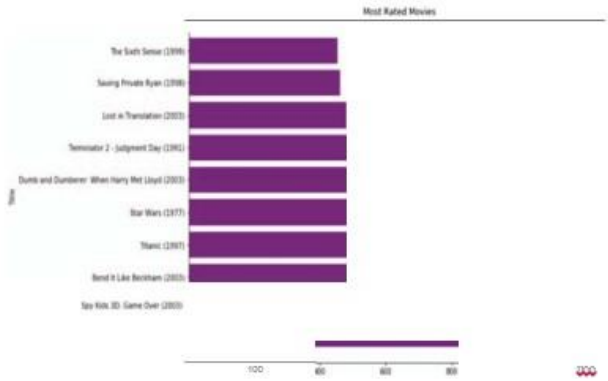

Fig –2: Most Rated Movies

2.1.2 Recommender system technique

Approaches of Recommendation System Recommendation system is usually classified on rating estimation

1. Collaborative Filtering system

2. Content-based system

3. Hybrid system

Comparable things to the ones the user enjoyed in the past will be offered to the user in a content-based approach, whilst items that similar group others with similar likes and preferences will be recommended in a collaborative filtering method. Hybrid systems that incorporate both techniques in some ways have been developed to address the limitations of both methodologies.

Model Construction - The recommender system was built using the Mahout library. We employed the User Similarity class in addition to the PearsonCorrelationSimilarity class for user-based filtering, which employs the Pearson Correlation Coefficient to assess the similarity of users' evaluations; thus the preference.

The Pearson Correlation mathematical formula is shown in Figure 3. The greater the correlation, the more closely the decisions of the two users are connected.

$$r = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2 \sum(y-\bar{y})^2}}$$

Fig -3: Pearson Correlation Coefficient formula.

The User Neighborhood is calculated using a distance-based clustering machine learning technique called NearestNUserNeighborhood, where N is specified in the programme code. The Nearest Neighbor algorithm looks for the most similar data-points among the N closest data-points around each data-point and groups them together.

To have a scalable and fault-resistant storage, the Item Based recommender's findings are put into the Hadoop Distributed File System (HDFS). Because, unlike items, user ratings must be computed every time a recommendation is made, the User Based recommender results must be computed every time a recommendation is made.

2.1.3 Training and Testing

Any system, to be successful, must be thoroughly tested, and well managed test plan should be prepared before actual testing is being performed. “Modules” have been developed and need to be tested in a manner that can reduce occurring of defects as low as possible. Following are the activities we planned to test the system.

1. This system is indeed an evolutionary system so every unit of the system is continuously under testing phase.

2. One test activity “Basis Path Testing” that will try to cover all paths in the system. This activity identifies all paths that provide different functionality of the system, and also other paths to reach at that functionality.

3. Other testing activity is” Control Structure Testing”, which will test each and every condition with positive and negative data combination.

4. This testing activity will also perform “Data Floe Testing” in which it will be tested how the data re following the

system. And will also check whether the data entered from one procedure, is reflected whenever it requires or not.

5. All conditions will be tested with "Boundary Value Analysis" where different input will be given to test whether the system is functioning with boundary values or not.

6. Along with the boundary value analysis, the system is also tested with "Range Value Tested" where editable values will be tested with ranges of values.

7. The system is being tested in "Unit Testing" manner where at the completion of one unit that is tested thoroughly with above mentioned testing activities.

8. The integration testing will also be performed to ensure that the integrated unit is working properly with other units or not.

2.1.4 Performance Evaluation

This paper's movie recommender system makes it easier to comprehend how a recommender system works. We examine these techniques separately to assess the accuracy and relevance of the findings generated by our system.

Movie 1	Movie 2	Similarity
1 80042113 9	1 8003 7 9 216	0.99959636
1 8000616 3 8	1 800111258	0.99959064
1 8 0 0 121659	1 8003 7 9 216	0.9995 54 63
1 807 5 3 746 3	1 8047 3 812 8	0 .9995 9 03
8028 3 191	1 807 858489	0.9995 346
1 800121659	1 800111258	0.999 5051
1 8000616 3 8	1 800121659	0.999477 5
1 8 0 0421 139	1 800121659	0.99944425
1 8 00111258	1 8003 7 9 216	0.99939984
1 800421 139	1 800111258	0.99938726
1 8000616 3 8	1 8003 7 9 216	0.9993557
1 80042113 9	1 8000616 3 8	0.999335
1 8 0 0 080788	1 8000807 9 5	0.9992829
1 8 0 743 259	1 80742 8 853	0.999 2593

Fig -4: Raw Output from Item Based Recommender

By mapping the Movie ID of Movie 1 and Movie 2 to their names, we can compare the Item based similarity coefficient values shown in Fig.4. We acquire the result displayed in Fig.5 by using Python panda's libraries. As the table shows, films that are related are given a higher similarity metric.

The AverageAbsoluteDifferenceRecommenderEvaluator is used to assess the model for a user-based recommender system. The training data is divided into test and train samples. The rating predictions on test data are then compared to the actual ratings indicated in the training data.

The raw result from the user-based filtering strategy is shown in Fig.6. The algorithm suggests ten films to user (User 5) and returns his closest neighbors who share his taste preferences. It also forecasts the user's ratings for each movie suggested (User 5). The average absolute difference is zero, indicating that the forecasts are correct.

The decisions based on the ratings of the suggested goods are completely correct.

```
--- IG.User Based Recommendations for User 5 -----•
RecommendedItem[item: 18G4738128. value: 5. GJ
RecommendedItem[item: 18G8481189, value: 5. GJ
RecommendedItem[item: 18G7537463. value: 5. GJ
RecommendedItem[item: 18G8414381. value:5. GJ
RecommendedItem[item: 18G84G3G3G. value: 5. GJ
RecommendedItem[item: 18G7432594. value:4. 6666665J
RecommendedItem[item: 18G84G4459. value:4. 5J
RecommendedItem[item: 18G84G4659. value:4. 5J
RecommendedItem[item: 18G84G6133. value:4. 5J
RecommendedItem[item: 18G84G4742. value:4. 5J
-----4 Users similar to User 5---- -
58
69
156
236
Average Absolute Difference= 0.00
```

Fig -5: Raw Output Recommender

The Lord of the Rings: The Two Towers (2018)]	5
Freaky Friday (2019)	5
The Lord of the Rings: The Fellowship of the Ring (2019)	5
Bad Boys II (2019)	5
How to Lose a Guy in 10 Days (2019)	4.5

Fig -6: Recommended movies and the predicted rating score (User Based)

Evaluation of Common Issues - The following difficulties are always mentioned with recommender systems. We assess our system in light of these concerns and offer an implementation strategy to address them.

The New User Problem, for starters, is concerned with the situation in which a new user is introduced to the recommender system. He has yet to submit any ratings for any of the movies in the system. This is referred to as a User Cold Start. One easy option is to propose top-rated movies or movies that have just been introduced to this new user.

Second, no new item is suggested, which is a source of worry. It's what's known as an Item Cold Start issue. When a new film is uploaded to the system, it does not yet have any ratings attached to it. What is the best way to find it and suggest it? One option may be to suggest films in the same genre as the top-rated films. If the new film belongs to that genre, it will be noticed. However, in order to achieve this goal, we will need to create a system based on the film's genre.

Keeping these recognized difficulties aside, depending on the infrastructure, any of the two ways may be employed. The similarity calculation for item-based is enormous, on the scale of MxM (M: total movies), but since it is static, we can perform it offline and re-compute it only after a certain amount of time has passed. On the other hand, since each user's neighborhood

is dynamic with fresh ratings from other users, it is costly to execute at runtime. As a result, the former needs cache data storage, while the latter necessitates the use of a specialized processing server.

3. Experimental Result

When the user clicks the "Generate Recommendation" button, a list of movies based on his prior ratings will appear. If he is a new user who has not yet rated any films, he is required to utilise the "search" box to choose a random film or one that piques his interest and rate at least six films. Only then, as seen in Fig.7, will the "Generate Recommendation" button become active.

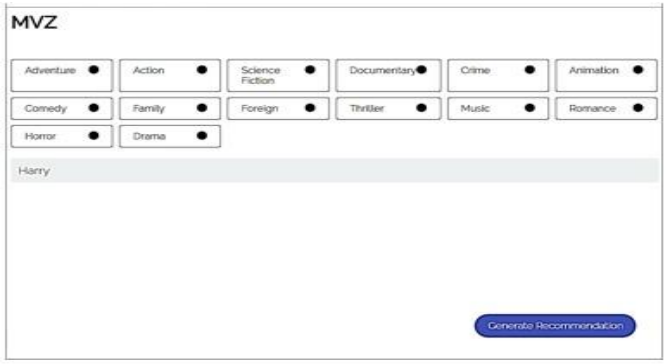

Fig -7: Search

Because the user is new and has not yet reviewed any films, he enters the word "Harry" into the search box, and all films that include the term "Harry" will display on the screen, as shown in Figs. 8.

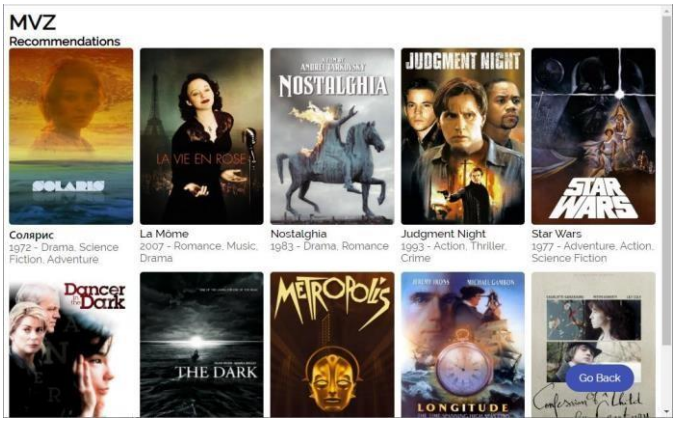

Fig -8: Search result.

The user then assigns ratings to these films based on his preferences, as seen in Fig.9. In order to get suggestions, the user must rate at least six films. The 'Generate Recommendations' option will be activated after he has rated six or more movies; until then, it will stay disabled.

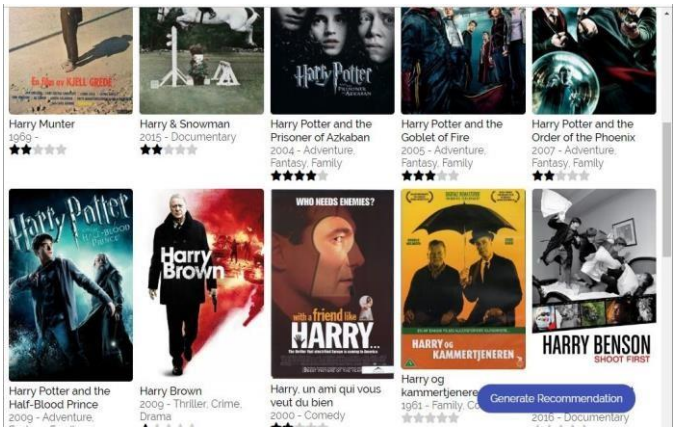

Fig -9: Rating Page

4. CONCLUSIONS

Instead of collaborative, a hybrid filtering strategy might be used to improve the recommender system in the future. According to recent studies, hybrid systems are more successful and deliver more accurate suggestions. As a result, hybrid systems would be preferable. To provide movie recommendations, our technology takes into account user ratings. More features, such as the film's genre, directors, actors, and so on, may be explored in the future to make ideas. In addition, instead of Mahout, a new framework called Apache Prediction 10 might be used to create the system. The Apache Prediction 10 is a machine learning server that builds Universal Recommender System using the Apache Hadoop, Apache Spark, Elastic Search, and Apache Hbase technology stack.

ACKNOWLEDGEMENT

I would like to thank my project guide "**Diksha Sharma",** Assistant Professor/Associate Professor, Department of Computer Science & Engineering, Institute of Technology and Management, Gida, Gorakhpur, U.P. for his valuable guidance and suggestions. I am thankful for his continual encouragement, support, and invaluable suggestions. Without his encouragement and guidance, this project would not have been materialized. Throughout the writing of the project, I have received a great deal of support and assistance.

REFERENCES

1. Mohapatra, H., Panda, S., Rath, A., Edalatpanah, S., & Kumar, R. (2020). A tutorial on powershell pipeline and its loopholes. International journal of emerging trends in engineering research, 8(4), 975-982.

2. Kumar, R., Edalatpanah, S. A., Jha, S., & Singh, R. (2019). A Pythagorean fuzzy approach to the transportation problem. Complex & intelligent systems, 5(2), 255-263.

3. Smarandache, F., & Broumi, S. (Eds.). (2019). Neutrosophic graph theory and algorithms.

Engineering Science Reference.

4. Kumar, R., Edalatpanah, S. A., Jha, S., & Singh, R. (2019). A Pythagorean fuzzy approach to the transportation problem. Complex & intelligent systems, 5(2), 255-263.

5. Mohapatra, H. (2009). HCR using neural network (Doctoral dissertation, Biju Patnaik University of Technology). Retrieved from https://www.academia.edu/39142624/HCR_USING_NEURAL_NETWORK

6. Mohapatra, H., & Rath, A. K. (2019). Detection and avoidance of water loss through municipality taps in India by using smart taps and ICT. IET wireless sensor systems, 9(6), 447-457.

7. Mohapatra, H., & Rath, A. K. (2019). Fault tolerance in WSN through PE-LEACH protocol. IET wireless sensor systems, 9(6), 358-365.

8. Mohapatra, H., Debnath, S., & Rath, A. K. (2019). Energy management in wireless sensor network through EB-LEACH (No. 1192). Easy Chair.

9. Nirgude, V., Mahapatra, H., & Shivarkar, S. (2017). Face recognition system using principal component analysis & linear discriminant analysis method simultaneously with 3d morphable model and neural network BPNN method. Global journal of advanced engineering technologies and sciences, 4(1), 1-6.

10. Panda, M., Pradhan, P., Mohapatra, H., & Barpanda, N. K. (2019). Fault tolerant routing in Heterogeneous environment. International journal of scientific & technology research, 8(8), 1009- 1013.

11. Mohapatra, H., Rath, A. K. (2020). Fault-tolerant mechanism for wireless sensor network. IET Wireless sensor systems, 10(1), 23-30.

12. Swain, D., Ramkrishna, G., Mahapatra, H., Patr, P., & Dhandrao, P. M. (2013). A novel sorting technique to sort elements in ascending order. International journal of engineering and advanced technology, 3(1), 212-126.

13. Xu, X. (2012). From cloud computing to cloud manufacturing. Robotics and computer-integrated manufacturing, 28(1), 75-86.

14. Haenlein, M., & Kaplan, A. (2019). A brief history of artificial intelligence: on the past, present, and future of artificial intelligence. California management review, 61(4), 514.

15. Gayen, S., Smarandache, F., Jha, S., & Kumar, R. (2020). Interval-valued neutrosophic subgroup based on intervalvalued triple T-Norm. In Neutrosophic sets in decision analysis and operations research (pp. 215-243). IGI Global.

Gayen, S., Smarandache, F., Jha, S., Singh, M. K., Broumi, S., & Kumar, R. (2020). Introduction to plithogenic subgroup. In Neutrosophic graph theory and algorithms (pp. 213-259). IGI Global.

16. Gayen, S., Jha, S., Singh, M., & Kumar, R. (2019). On a generalized notion of anti-fuzzy subgroup and some characterizations. International journal of engineering and advanced technology.

17. Zheng, H., Liu, D., Wang, J., & Liang, J. (2019). A QoEperceived screen updates transmission scheme in desktop virtualization environment. Multimedia tools and applications, 78(12), 16755-16781.

18. Broumi, S., Dey, A., Talea, M., Bakali, A., Smarandache, F., Nagarajan, D., & Kumar, R. (2019). Shortest path problem using Bellman algorithm under neutrosophic environment. Complex & intelligent systems, 5(4), 409-416.

19. Kumar, R., Edalatpanah, S. A., Jha, S., Broumi, S., Singh, R., & Dey, A. (2019). A multi objective programming approach to solve integer valued neutrosophic shortest path problems. Neutrosophic sets and systems, 24, 134-149.

20. Kumar, R., Dey, A., Broumi, S., & Smarandache, F. (2020). A study of neutrosophic shortest path problem. In Neutrosophic graph theory and algorithms (pp. 148-179). IGI Global.