iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IV **Month of publication:** April 2024

DOI: https://doi.org/10.22214/ijraset.2024.58984

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

A Study on PrediX: Harnessing the power of ML for accurate Predictive Modelling for Movies (MOVIEMAGIC)

Geetika Bhatnagar[1], Ayushi Johari[2], Ankit Chauhan[3], Ms. Nidhi Singh[4]
Department of Computer Science and Engineering, Galgotias College of engineering and Technology, Gautam Budh Nagar,India

Abstract: ***Machine learning is becoming increasingly important in the technical world, and the entertainment business is growing rapidly. The ways in which people consume content are becoming more complex and changing more quickly than in the past. Recommendation engines powered by machine learning create independent systems that grow and learn from their mistakes without requiring explicit coding. It is a system that enables a user to sift through vast volumes of data and identify useful information for themselves. Every entertainment provider displays relevant information to a user according to his tastes using a sophisticated recommendation algorithm. Both their sales and user base retention are aided by it . Different techniques are used by movie recommendation systems. For example, collaborative filtering (CF) compares people based on how similar they are to each other in terms of content consumption, while content-based filtering makes use of the movie's attributes such actors, genre, and year of release. A hybrid technique combines two or more distinct methods for suggesting movies. In this work, we offer an architecture for a movie recommendation system that addresses the cold-start issue by utilizing ML and the MERN stack.***

Keywords: stack, MERN, machine learning cooperative, filtering MongoDB

I. INTRODUCTION

In today's digital age, the entertainment industry is booming with a vast array of movies and TV shows, making it increasingly challenging for viewers to discover content that suits their preferences. To address this challenge, movie recommendation systems powered by machine learning have become invaluable tools. This introduction provides an overview of a Movie Recommendation System built using the MERN (MongoDB, Express, React, Node.js) stack and machine learning techniques.

A. The Challenge

With the explosion of streaming platforms and a seemingly infinite number of movies available, users often find it overwhelming to choose what to watch. This problem calls for a personalized solution, where users receive movie recommendations tailored to their tastes and preferences.

B. The Solution

A Movie Recommendation System utilizes machine learning algorithms to analyze user behavior and movie data to provide personalized movie suggestions. By combining the power of machine learning with the versatility of the MERN stack, we create a comprehensive solution that integrates a database, backend, and frontend to deliver a seamless user experience.

C. Key Components

1) *Data Collection:* The system starts by collecting a vast amount of movie data, including details like titles, genres, ratings, and user interactions.
2) *Data Preprocessing:* This data undergoes preprocessing, where missing values are handled, and user-item interaction matrices are created. Data quality is crucial for the accuracy of recommendations.
3) *Machine Learning Model:* The heart of the system is the recommendation algorithm. Collaborative filtering and matrix factorization are commonly used techniques to make movie suggestions.
4) *API Development (Node.js and Express):* To communicate between the machine learning model, database, and the user interface, an API is created using Node.js and Express.
5) *Database (MongoDB):* User data, movie information, and related data are stored securely in a MongoDB database.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

6) *Frontend (React):* The frontend is built using React, offering an interactive and user-friendly interface where users input their preferences and receive movie recommendations.
7) *User Authentication:* To personalize recommendations and ensure data security, user authentication is implemented.

D. The User Experience

Users experience consists of a simple and easy-to-use interface that allows users to rate movies, list their preferences, or just begin watching movies that are suggested. Over time, the machine learning model refines its recommendations by continuously adapting to user interactions.

II. COLLABORATIVE FILTERING

Predicting systems known as recommendation systems make bold recommendations for items to users or users to the items, and occasionally even users to other users. Similar techniques are used by giants of the internet like Netflix, YouTube, and Amazon Prime to suggest video content based on the user's intended interest. Since there is a ton of information on the internet, it can be challenging and time-consuming to find your material; for this reason, the recommendation is crucial in reducing our effort. These days, these systems are becoming more and more common in a variety of contexts, including books, movies, music, videos, and other social network sites where material is filtered based on recommendations. It's a program that uses user data to enhance suggestion performance and display the most favored choice. In order to construct the tool, user/customer happiness is crucial. Both consumers and businesses benefit from it since happier customers are more likely to want to utilize the system for convenience, which eventually brings in money for the businesses. A recommendation system should always be updated because user preferences may differ from those of other users, and our method has the unfortunate consequence of having people who are dissatisfied with the outcome not using it again. Despite the abundance of algorithms available, collaborative filtering is the most widely utilized algorithm by businesses since it encourages more user engagement. Because collaborative filtering examines the user's surfing history and compares it with other users', it can forecast more accurately than content-based filtering. Conversely, content-based filtering employs cosine similarity to discover related movies and recommends them in decreasing order depending on the information provided by the user. Another technique is called context-based filtering, and it works by gathering additional data from the user, such as genre, release date, and mood, to produce more effective results. In this project, we wanted to keep our system as basic as possible while maintaining a high level of accuracy when compared to other recommendation systems. Content-based filtering is not very precise or accurate, and it has several disadvantages. Thus, the collaborative filtering recommendation system utilizing nearest neighbors is the suggested system.

A. Related Work

Content-based, Collaborative (User-item, User-user), Context-based and Hybrid, and also these days Deep Learning are some of the approaches used to solve the problem of movie recommendation. In, C. S. M. Wu, D. Garg, and U. Bhandary presented a collaborative filtering recommendation system in which the list is suggested based on user ratings. The authors compared the effectiveness and performance of user-based versus item-based suggestions using the Apache Mahout framework. The percentage view method was presented by R. E. Nakhli, H. Moradi, and M. A. Sadeghi in for user movie recommendations. It discovers relevant movies for the user and then assesses the project's correctness by comparing its performance to a random movie recommendation system.

B. Collaborative Filtering (CF)

It essentially suggests products to people with similar tastes by filtering out content based on user interests that are similar to those of other users. In the industries, it is a well-known and well-liked algorithm as well. Within the memory-based methods, two widely used filtering algorithms exist.

In contrast to memory-based strategies, there is another method that is referred to as model-based, but it is less dependable. Figures 1 and 2 presented user- and item-based collaborative filtering, respectively. Figure 1: User-Based CF Demonstration Figure 3: Item Based CF Demonstration It is anticipated that in a user-based system, the user will enjoy goods that other users who share their tastes also enjoy.

Recommended In the item-based like it is assumed that the user will like those items that are similar to the other items liked before. The hybrid approach-This approach provides very accurate results using both collaborative and content-based filtering while removing the drawbacks of the algorithms at the same time. This integrated system is getting more attention nowadays as it is better than both the algorithms .

©IJRASET: All Rights are Reserved

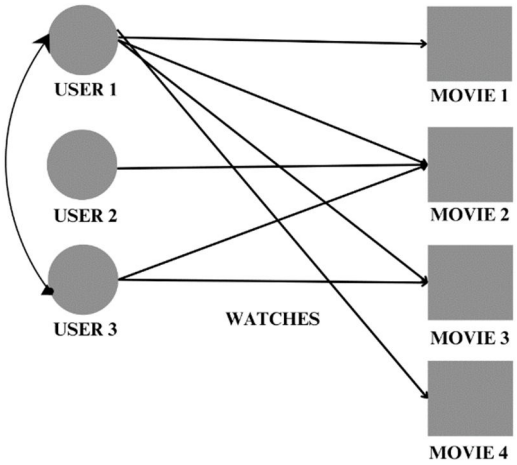


Figure 1: Demonstration of User-Based CF

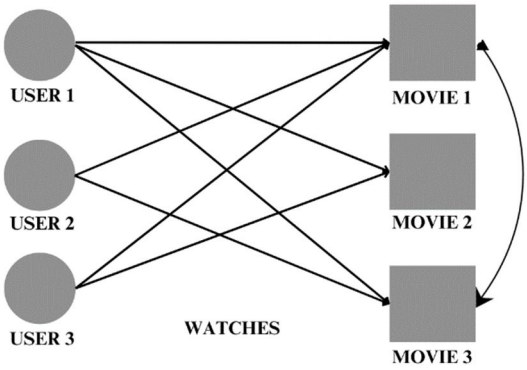


Figure 2: Demonstration of Item-Based CF

C. Proposed Recommendation Engine

The collaborative filtering strategy (item-based approach) utilised by the proposed recommendation system is significantly more accurate and efficient to use because it is non-dynamic and can be done offline, unlike user-based updates. The suggested method ranks the top k most similar movies using cosine angle similarity after utilising the KNN algorithm to calculate the distance between each target movie and every other movie in the dataset . Different techniques used in this proposed algorithm are discussed below:

1) KNN algorithm- In recommendation systems, the KNN algorithm is well-known for its quick calculation times and enhanced prediction capabilities.

2) Cosine similarity- This measure is used to determine how far apart the target movie is from the films in the dataset. Eq. (1) is utilized to determine the cosine similarity of the suggested model. It quantifies the similarity between two documents regardless of how varied they are in size and computes the cosine angle between two vectors in multi-dimensional space[6].

Item-based collaborative filtering-makes the assumption that users will enjoy goods that are comparable to those they have already enjoyed . Here, the goal is to suggest movies by employing the item-based method. The dataset must first be extracted in order to obtain data regarding the intended film and user ratings.

III. CONTENT-BASED MOVIE RECOMMENDATION

A recommendation system is a system that, using a set of data, makes recommendations to users about particular resources, such as books, movies, songs, and so on. Movie recommendation algorithms typically use characteristics of previously enjoyed films to forecast the kind of movies a user will enjoy. These recommendation systems are advantageous to businesses that gather a lot of consumer data and want to efficiently offer the finest recommendations. When creating a movie recommendation system, a variety of elements can be taken into account, such as the movie's genre, the actors in it, or even the director. The algorithms have the ability to suggest movies based on one, two, or more criteria combined . In this paper, the recommendation system has been built on the type of genres that the user might prefer to watch. The approach adopted to do so is content-based filtering using genre correlation. The dataset used for the system is Movie Lens dataset. The data analysis tool used is R.

In this age of the Internet, the number of data transactions that occur every minute has risen. The huge amount of data has grown exponentially along with the number of internet users. However, not all information found on the Internet is useful or provides satisfactory results to users. Such huge amounts of data often turn out to be inconsistent and without proper processing of this data, it goes to waste. In such cases, users have to search several times before they finally find what they are looking for. To solve this problem, researchers have developed recommender systems. The recommendation system provides users with relevant information based on their past preferences. The data is filtered and personalized according to the user's needs. With more and more information on the internet, recommendation systems have become really popular because they are effective in delivering information in a short period of time. Recommender systems have been developed for various domains such as music, movies, news and products in general. Today, most organizations implement recommendation systems to meet customer requirements. LinkedIn, Amazon and Netflix are just a few to mention. LinkedIn recommends relevant connections from people the user may know from among the portal's millions of subscribers. In this way, the user does not have to perform extensive personal searches manually. Amazon's recommendation systems suggest similar products for customers to buy. If a particular customer prefers to buy books from the shopping portal, Amazon offers suggestions for new results in previously suggested categories. In a very similar way, Netflix considers the types of shows a customer watches and makes similar recommendations. The operation of recommender systems can be broadly divided into three categories: content-based, collaborative and hybrid approaches. A content-based recommendation system takes into account the user's past behavior and identifies related patterns, based on which similar items are recommended. Collaborative filtering analyzes a user's past experiences and ratings and associates them with other users. Recommendations are made based on those with the most similarities. Both content-based and collaborative filtering have their limitations. To solve this, the researchers proposed a hybrid approach that would combine the advantages of both methods. This paper proposes a content-based recommendation system using genre correlation. The dataset used for this purpose is the Movie Lens dataset, which contains 9126 movies categorized by genre. There are a total of 11 genres. The ratings for these moves have been collected from 671 users. By taking into account the movies which received high ratings from the users, movies containing similar genres. Recommender systems are broadly classified into three types—collaborative filtering systems, content-based filtering systems, and hybrid systems. Collaborative systems utilize inputs from various users and run various comparisons on these inputs. They build models from the past behaviour of the users. Movie recommendation systems, for example, utilize the ratings of users for various movies , and attempt to find other like-minded users, and recommend movies they have rated well . Collaborative filtering systems have two approaches—memory-based approaches and model-based approaches . Memory-based approaches continuously analyse user data in order to make recommendations. As they utilize the user ratings, they gradually improve in accuracy over time . They are domain-independent and do not require content analysis. Model-based approaches develop a model of a user's behaviour and then use certain parameters to predict future behaviour . The use of partitioning-based algorithms also leads to better scalability and accuracy. Content-based filtering systems examine user-given preferences or documents in an effort to create a model based on this information. They take advantage of a user's specific interests and try to align the user's profile with the characteristics of the different content objects that are going to be suggested. An additional drawback is that they need a sufficient amount of data to construct a trustworthy classifier. There are three types of techniques for content-based filtering systems: wrapper methods, filter methods, and embedding methods. By using wrapper approaches, the features are divided into subsets, these subsets are analysed, and the most promising subset is determined. Heuristic techniques are used by filter algorithms to rank features based on their content. Neither of these approaches depends on the algorithms that are employed. Conversely, embedded techniques are connected with feature selection is carried out by the employed

algorithm during the training stage. To improve recommender systems and lessen the shortcomings of each approach, hybrid systems include content-based and collaborative filtering algorithms. As such, it attempts to leverage the advantages of one approach to offset the drawbacks of the other. Hybrid systems come in three flavours: cross-source, mixed, and weighted hybrids.
A score is kept for every object in weighted hybrid systems, and the weighted aggregate is determined using the different context sources. Depending on the preferences of the user, these are assigned varying weights.In mixed hybrid techniques, the items are ranked using each source, and the top few things are selected from each rank list .Cross-source hybrid methods recommend items that appear in multiple context sources . Hybrid cross-source approaches suggest items that show up in several context sources. These techniques operate under the tenet that an item is more significant the more sources it appears in. Wakil et al. made an effort to enhance their recommendation system by employing emotional filtering. A user experiences specific feelings when they view a particular kind of movie. In a similar vein, a user's feelings may dictate the kind of movie they choose to view. They created an algorithm that makes use of emotion determination after realising that conventional user profiles do not account for the user's emotional state. It evaluates a colour sequence the user selects based on his feelings to ascertain the user's present emotional state. A hybrid recommendation system with feature weighting was proposed by Debnath et al. They evaluated the significance of different aspects for every user and gave these features weights in accordance with their findings. After that, they calculated the weighted aggregate to determine which products would pique the user's interest even more.

IV. HYBRID FILTERING

A recommendation system is a system that offers suggestions to users, leveraging specific data such as books, movies, songs, and other relevant information. Movie recommendation algorithms utilize the attributes of previously enjoyed films to predict the preferences of users and suggest similar movies they might enjoy. Businesses that gather ample customer data and strive to deliver top-notch recommendations can greatly benefit from these recommendation systems. When creating a movie recommendation system, multiple elements such as genre, cast, and even the director of the movie are taken into account. This paper introduces a hybrid movie recommendation system that utilizes a combination of weighted average and min-max scaler to assess movie ratings and popularity. Moreover, TF-IDF is utilized for transforming the data into vectors, while cosine similarity is employed to gauge the resemblance among these vectors. The recommender system is built using the Movies dataset. The results show the top-K recommendation for users as well as the proposed system can provide a prediction of rating for a particular movie.

A. Business Understanding

The initial and one of the most important steps of the research project is the business understanding. The objective of this research is to predict movies according to the user preferences to enhance user viewing experience. Historical data is used for recommending movies to the user according to their preferences. Collaborative Filtering or Content-based filtering are two of the most widely used algorithms used in recommendation systems in the industry, however we try to improve the performances of these models with the inclusion of SVD and SVD++ Algorithms. The goal of this research is to compare various models and algorithms to identify the most efficient recommendation engine. Is the Hybrid recommendation engine the most efficient among all the other recommendation engines, what other models were compared with it? Enhancement of recommendation systems even to a small extent can be very profitable to the businesses as it can make millions of users buy the right product or service and hence increase the overall revenue of the company. In case of online streaming platforms and OTT services, a better movie or video recommendation based on previous ratings could help the company in generating more screen time through the customer and can also convince the potential customer to subscribe to their services for longer period of times. This research study is conducted to answer the research question, ‘To improve the performances of traditional recommender .

B. Modelling

The process of applying various models to our data and eventually obtaining each model's performance for evaluation is known as the "data modeling stage." The models are chosen in accordance with the results of the literature review and the research that has been done on recommender engines. Preprocessing data for modeling purposes is a must at several stages of the data modeling process. However, the modeling step consists entirely of the model production, model evaluation, and model selection. Here, different algorithms are used to compare the model's results and then use those results to determine which model performs the best. Evaluation indicators such as test error, RMSE, hit-miss ratio, and model accuracy are used to assess how well our models perform in our research project. We also use RapidMiner to get the results for Hybrid recommender engine. The following models were used in the research methodology based on the Literature review and research conducted on recommender systems:

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

1) Content-Based Filtering
2) Collaborative Filtering
3) Hybrid Recommender System

C. Hybrid Recommender System

To overcome shortcomings of an individual model, we have developed a hybrid model wherein we stack two different models. The resultant hybrid model gives higher accuracy and more relevant results. The SVD model receives the movie recommendation from Content-Based Filtering and forecasts the user's rating for the recommended film. Lastly, we return the films to their SVD predicted ratings in descending order. Combining features: We combined collaborative filtering methods with content-based filtering approaches to create a hybrid recommender engine using the feature combination technique. The SVD-based rating prediction is essentially used in the implementation as an extra feature data above the content-based movie recommendation. The usage of Feature combination in the implementation of hybrid model helped us in getting more meaningful features out of the traditional features to improve the accuracy and performance of the model. The advantage of implementing feature selection in hybrid model is that is allows the system to consider the data given by SVD approach without completely relying on it, in return it reduces the sensitivity of the engine to the users who have rated the movies.

D. Matrix Factorization

In matrix factorisation, the process is similar to the process used in neighbourhood-based approaches, however there is only one change to the model operator which is Biased Matrix factorisation (BMF) The central operator here is called the BMF and in the collaborative approach, the one and only input needed is the ratings matrix. Here, the set role operator is used to declare the attributes for the purpose of i) user identification and ii) item identification. The data is split between training -95The modelling operator is found in the recommenders under item rating prediction followed by collaborative filtering rating prediction. The parameters are configured according to the modelling operator in order to suit the model

V. CONCLUSION

In conclusion, the Movie Recommendation System developed using the MERN (MongoDB, Express, React, Node.js) stack and machine learning represents a cutting-edge solution to the ever-growing challenge of content discovery in the entertainment industry. This system addresses the fundamental problem of helping users find movies that align with their preferences and interests. By harnessing the power of machine learning algorithms, this system not only simplifies the movie selection process but also enhances the overall user experience. Here are the key takeaways:

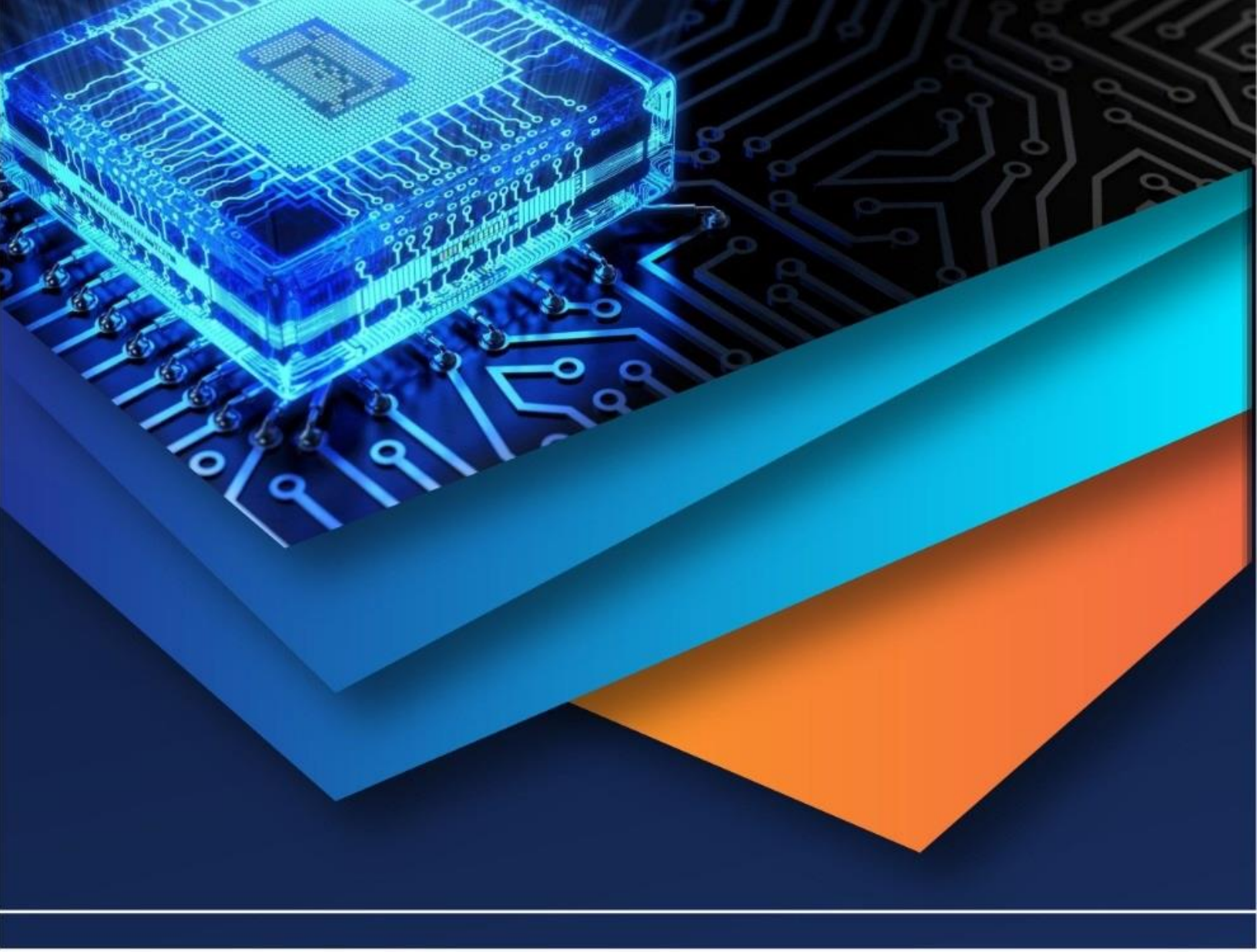
1) *Personalized Recommendations:* The heart of the system lies in its ability to provide personalized movie suggestions. Through collaborative filtering and machine learning, users receive recommendations based on their unique viewing history and preferences.
2) *User-Centric Interface:* The React-based frontend offers an intuitive and user-friendly interface, making it easy for users to interact with the system, rate movies, and explore personalized recommendations.
3) *Data-Driven Decision Making:* The system's ability to analyze user behavior and movie data empowers it to continuously adapt and improve its recommendations, resulting in a dynamic and evolving user experience.
4) *Secure User Authentication:* User data security is a top priority. By implementing robust user authentication mechanisms, the system ensures that user preferences and data remain confidential.
5) *Scalability:* The MERN stack, combined with best practices in database design, enables the system to scale effectively, accommodating a growing user base and expanding movie libraries.
6) *Ongoing Evolution:* As technology advances and user feedback is collected, the Movie Recommendation System has the potential to become even more accurate and personalized, refining its recommendations and providing an ever-improving experience for users.

In an era of information overload, where choices seem limitless, this system offers a crucial solution by simplifying decision-making and enhancing the enjoyment of movie enthusiasts. As technology and machine learning algorithms continue to advance, the future of movie recommendations promises even more accurate and enjoyable viewing experiences, tailored to individual preferences. This project represents an exciting convergence of data science, web development, and entertainment, ultimately improving the way we discover and enjoy movies.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

REFERENCES

[1] Amatriain, X., & Basilico, J. (2013). Netflix Recommendations: Beyond the 5 stars. In Proceedings of the 6th ACM Conference on Recommender Systems (RecSys '12), 89-96.
[2] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. Computer, 42(8), 30-37.
[3] Cremonesi, P., Koren, Y., Turrin, R., & Shapira, M. (2010). Content-based recommendation systems. In Recommender Systems Handbook, 73-105. Springer.
[4] Karatzoglou, A., Amatriain, X., Baltrunas, L., & Oliver, N. (2010). Deep Learning for Recommender Systems. In Proceedings of the 4th International Conference on Web Intelligence, Mining, and Semantics (WIMS '14), 1-5.
[5] Low, Y., Gonzalez, J., Kyrola, A., Bickson, D., et al. (2012). A Fast Parallel Stochastic Gradient Method for Matrix Factorization in Shared Memory Systems. In Proceedings of the 11th International Conference on Data Mining (ICDM '11), 249-258.
[6] Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., & Li, H. (2007). Learning to Rank: From Pairwise Approach to Listwise Approach. In Proceedings of the 24th International Conference on Machine Learning (ICML '07), 129-136.
[7] Sedhain, S., Menon, A. K., Sanner, S., & Xie, L. (2015). AutoRec: Autoencoders Meet Collaborative Filtering. In Proceedings of the 24th International Conference on World Wide Web (WWW '15), 111-112.
[8] Rendle, S. (2012). Factorization Machines. In Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NeurIPS '12), 1278-1286.
[9] Ekstrand, M. D., & Riedl, J. T. (2014). Balancing Prediction and Recommendation Precision and Recall. In Proceedings of the 23rd International Conference on World Wide Web (WWW '14), 49-50.
[10] Hegedüs, I., & Karatzoglou, A. (2015). Probabilistic Matrix Factorization. In Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15), 369-370.
[11] Covington, P., Adams, J., & Sargin, E. (2016). Deep Neural Networks for YouTube Recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16), 191-198.
[12] Drake, E., & Caverlee, J. (2013). SVD-based Collaborative Filtering: A Comparative Analysis. In Proceedings of the 22nd International Conference on World Wide Web (WWW '13), 507-518.
[13] Karnstedt, M., Hayes, C., Kawsar, F., & Adya, A. (2011). Context-aware Event Recommendation in Event-based Social Networks. In Proceedings of the 3rd ACM International Conference on Event-based Systems (DEBS '11), 139-149.
[14] Salakhutdinov, R., & Mnih, A. (2008). Probabilistic Matrix Factorization. In Proceedings of the 21st International Conference on Neural Information Processing Systems (NeurIPS '08), 1257-1264.
[15] Toscher, A., Jahrer, M., & Bell, R. M. (2009). The BigChaos solution to the Netflix Grand Prize. In Proceedings of the 2009 KDD Cup and Workshop (KDD Cup '09), 6-10.
[16] Aggarwal, C. (2016). Towards Next Generation of E-commerce: Recommendation Systems. In Recommender Systems: The Textbook, 333-369. Springer.
[17] Hegedüs, I., & Karatzoglou, A. (2018). Balancing Prediction and Recommendation Precision and Recall. In Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18), 204-212.
[18] Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). BPR: Bayesian Personalized Ranking from Implicit Feedback. In Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI '09), 452-461.
[19] Cantador, I., Brusilovsky, P., & Kuflik, T. (2011). Second Workshop on Information Heterogeneity and Fusion in Recommender Systems. In *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys '11), 397-398.
[20] Kabbur, S., Ning, X., & Karypis, G. (2013). A Unified Framework for Item-based Top-N Recommendation. In Proceedings of the 23rd ACM International Conference on Information and Knowledge Management (CIKM '14), 2421-2426.**
[21] Vozalis, M., & Margaritis, K. (2005). Using an improved Ant Colony System for Collaborative Filtering. Information Sciences, 170(1), 123-141.
[22] Huang, Z., Chung, W., & Ong, T. (2002). Comparative studies of collaborative filtering algorithms. Artificial Intelligence Review, 17(1-2), 143-166.
[23] Desrosiers, C., & Karypis, G. (2011). A Comprehensive Survey of Neighborhood-based Recommendation Methods. In Recommender Systems Handbook, 107-144. Springer.
[24] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. In Proceedings of the 2nd ACM Conference on Electronic Commerce (EC '00), 158-167.
[25] Kantor, P. B., & Kotlyar, M. (1997). Data Mining with Massive Data. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD '97), 43-50.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)