

ENTERTAINMENT RECOMMENDATION.

P.Hinduja¹, M.Bhargavi², A.Mounica³

Priya Nandini⁴

Department of Computer Science and Engineering, Stanley College of Engineering and Technology for Women, Telangana, India

Abstract. The purpose of a recommender system is to suggest relevant items to users. To achieve this task, there exist two major categories of methods : collaborative filtering methods and content based methods. This project is an entertainment recommendation system. This is a web application where the user can register, and login. After the login , the user will see an dashboard and in the dashboard there are 2 buttons , one is movie recommendation and the other one is music recommendation and.. In the movie recommendation page , top 15 movies based on the ratings will be there and in the music recommendation page , top 15 music based on the atingswill be there. Then the user can logoff from the application.

Keywords: Recommendation system, movies, music

1. Introduction

1.1 About Project

In this hustling world, entertainment is a necessity for each one of us to refresh our mood and energy. Entertainment regains our confidence for work and we can work more enthusiastically. For revitalizing ourselves, we can listen to our preferred music or can watch movies of our choice. For watching favorable movies online we can utilize movie recommendation systems, which are more reliable since searching of preferred movies will require more and more time which one cannot afford to waste. Entertainment item recommendation is a challenging task because of the variance in what an individual perceives as comfortable versus another. For example, children prefer animation, young people like adventure and adult like family movies. Additionally, the preference of entertainment varies in different time intervals or period. For example, listening news or watching sports in the weekdays and enjoying movies in the weekend or reading comics in the childhood and reading novels in adulthood. Entertainment preference also varies from culture to culture. For example, doing religious work or doing social work.

A recommendation system or recommendation engine is a model used for information filtering where it tries to predict the preferences of a user and provide suggests based on these preferences. These systems have become increasingly popular nowadays and are widely used today in areas such as movies, music, books, videos, clothing, restaurants, food, places and other utilities. These systems collect information about a user's preferences and behaviour, and then use this information to improve their suggestions in the future. Movies are a part and parcel

of life. There are different types of movies like some for entertainment, some for educational purposes, some are animated movies for children, and some are horror movies or action films. Movies can be easily differentiated through their genres like comedy, thriller, animation, action etc. Other way to distinguish among movies can be either by releasing year, language, director etc. Watching movies online, there are a number of movies to search in our most liked movies. Movie Recommendation Systems helps us to search our preferred movies among all of these different types of movies and hence reduce the trouble of spending a lot of time searching our favourable movies. So, it requires that the movie recommendation system should be very reliable and should provide us with the recommendation of movies which are exactly same or most matched with our preferences. A large number of companies are making use of recommendation systems to increase user interaction and enrich a user's shopping experience. Recommendation systems have several benefits, the most important being customer satisfaction and revenue. Movie Recommendation system is very powerful and important system.

Although some algorithms show that the user ratings are a powerful parameter for the recommendation system, it solely could not provide a high accurate result especially when the preference of a user changes over time. This project proposes a novel system to recommend popular items to users. We propose an entertainment recommendation algorithm that recommends entertainment items to a group of users. It relies on not only the ratings but also the time of the provided ratings. With the help of three experimental results, we validate the performance of our system.

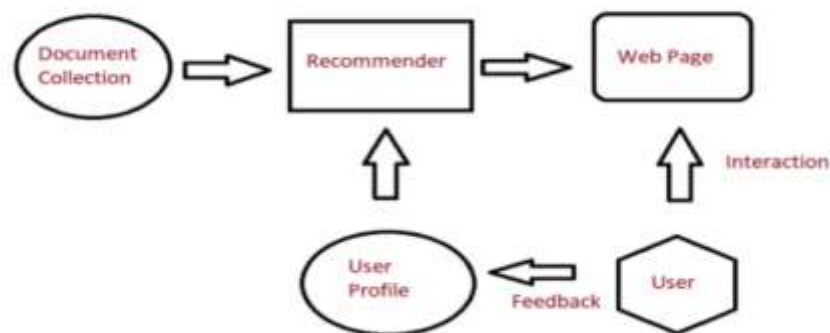
1.2 Objectives of The Project

In the era of information overload, it is very difficult for users to get information that they are really interested in. And for the content provider, it is also very hard for them to make their content stand out from the crowd. That is why many researchers and companies develop Recommender System to solve the contradiction. The mission of Recommender System is to connect users and information, which in one way helps users to find information valuable to them and in another way push the information to specific users. This is the win-win situation for both customers and content providers

- The goal of the project is to recommend a movie or music to the user.
- Providing related content out of the relevant and irrelevant collection of items to users of online service providers.
- Improving the Accuracy of the recommendation system
- Improve the Quality of the movie Recommendation system
- Improving the Scalability.
- Enhancing the user experience.

1.3 Scope of The Project

The objective of this project is to provide accurate movie recommendations to users. The goal of the project is to improve the quality of movie recommendation system, such as accuracy, quality and scalability of system than the pure approaches. This is done using Hybrid approach by combining content based filtering and collaborative filtering, To eradicate the overload of the data, recommendation system is used as information filtering tool in social networking sites. Hence, there is a huge scope of exploration in this field for improving scalability, accuracy and quality of movie recommendation systems Movie Recommendation system is very powerful and important system. But, due to the problems associated with pure collaborative approach, movie recommendation systems also suffers with poor recommendation quality and scalability issues. There are many kinds of Recommender Systems but not all of them are suitable for one specific problem and situation. Our goal is to find a new way to improve the classification of movies, which is the requirement of improving content-based recommender systems.



2. Literature Survey

2.1 Existing System

Ramni Harbir Singh et al. illustrated the modelling of a movie recommendation system by making the use of content-based filtering within the recommendation system [1]. The KNN algorithm is implemented in this model along with the principle of cosine similarity which yields more accuracy than other distance metrics and also the running complexity is comparatively low too.

D. K. Yadav et al. presented MOVREC, a movie recommendation system based on collaborative filtering approach [2]. Collaborative filtering makes use of information provided by the user. That information is analyzed and a movie is recommended to the users which are arranged with the movie with the highest rating first.

Luis M Capos et al. has analyzed two traditional recommender systems i.e. content based filtering and

collaborative filtering [3]. As both of them have their own drawbacks he proposed a new system which is a combination of Bayesian network and collaborative filtering.

A hybrid system has been presented by Harpreet Kaur et al.[4]. The system uses a mix of content as well as collaborative filtering algorithms. The context of the films is additionally considered while recommending. The user - user relationship also as user - item relationship plays a task within the recommendation.

2.2 Proposed System

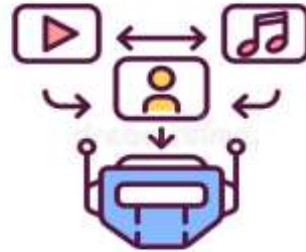
The user specific information or item specific information is clubbed to make a cluster by Utkarsh Gupta et al. using chameleon [5]. This is often an efficient technique based on Hierarchical clustering for the recommender system. To predict the rating of an item electoral system is employed. The proposed system has lower error and has better clustering of comparable items.

Urszula Kuelewska et al. proposed clustering as a way to deal with recommender systems[6]. Two methods of computing cluster representatives were presented and evaluated. Centroid-based solution and memory-based collaborative filtering methods were used as a basis for comparing effectiveness of the proposed two methods. The result was a big increase within the accuracy of the generated recommendations in comparison to only the centroidbased method.

Costin-Gabriel Chiru et al. proposed Movie Recommender, a system which uses the information known about the user to provide movie recommendations.[7] This technique attempts to unravel the matter of unique recommendations which ends up from ignoring the data specific to the user. The psychological profile of the user, their watching history and therefore the data involving movie scores from other websites is collected. They're supported aggregate similarity calculations. The system is a collaborative filtering.

Nicolas Hug proposed a paper on Surprise library in Python [8]. It is used for building and analyzing rating prediction algorithms. Surprise is mainly written in Python, while the computationally intensive parts are optimized with Cython. Internally, Surprise relies on built-in Python data structures and also on numpy arrays. It provides a set of estimators (or prediction algorithms) for rating prediction. Among others, classical algorithms are implemented like the most similarity-based algorithms, and the creation of custom prediction algorithms can also be done.

G. Linden et al. presented about the most popular recommender system Amazon [9]. Amazon is one among the most important sales companies within the United States. Being an internet based retailer, it's a recommendation system as an integrated a part of the marketing tool. This technique basically uses Item-to-Item Collaborative Filtering for personalizing the 5 web site in order that it's suitable for each customer's interests . A characteristic property of Amazon's recommender system is that it's specially designed to scale to many customers and products so as to supply high-quality recommended.



3. Proposed Architecture

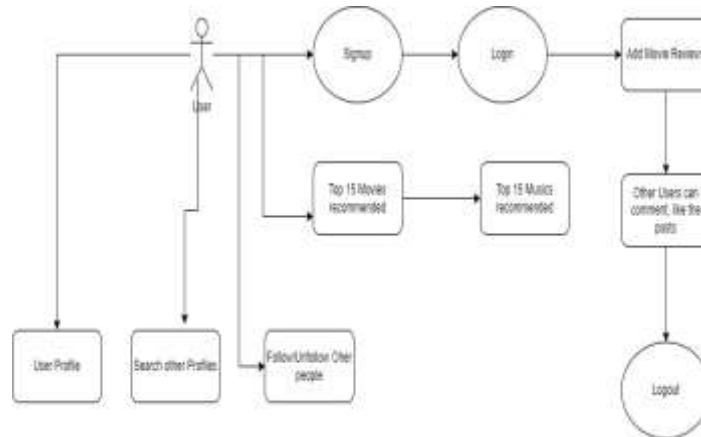


Fig 3.1 Proposed Architecture diagram

Initially the user is directed to the login screen , after entering the password and email , if both the password and email are valid , the user is directed to the home screen. The user have the option to post movie reviews and then they can search for other users and view their profile and follow and unfollow them. After following them , they can view the movie reviews created by them. They also have an option to check the top 15 movies and music recommendations.

4. Implementation

This is the app.js file in which the server will initiate all the functions. The express server is imported and the mongoose library is imported for mongodb database operations. The port number is 5000 and in

that port the server will be running. Since the app.js is the route file , we will be exporting all the routes(endpoints) and models(database schema) to the app.js file.

This is the auth.js file showing sign in route. The major job of the sign in route is to check if both the email and password are not empty and then the query will be passed to check if there is an user document exists with the given email, if exists it will check the password , if both the passwords matched , the route will send 200 request with the “logged in successfully” message This is the user.js (model file). In this file , database schema is defined for the user with mandatory fields. Every field has 1-2 parameters, the first parameter is type that indicates whether the type is String or int and the second type is the required field.

This is the auth.js file showing sign in route. The major job of the sign in route is to check if both the email and password are not empty and then the query will be passed to check if there is an user document exists with the given email, if exists it will check the password , if both the passwords matched , the route will send 200 request with the “logged in successfully” message.

4.1 Code Implementation

Visual Studio Code, also commonly referred to as VS Code,[9] is a source-code editor made by Microsoft for Windows, Linux and macOS.[10] Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

In the Stack Overflow 2021 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python and C++.[16][17][18][19] It is based on the Electron framework,[20] which is used to develop Node.js Web applications

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm,[6] unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts

Python is an interpreted high-level general-purpose programming language. Python's design philosophy

emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

5. Result

Since our project is a movie recommendation system .one can develop a movie recommendation system by using either content-based or collaborative filtering or combining both.

Advantages: it is easy to design and it takes less time to compute

Disadvantages: the model can only make recommendations based on the existing interests of the user. In other words, the model has limited ability to expand on the users' existing interests.

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all the system elements have been properly integrated and perform allocated functions. The testing process is actually carried out to make sure that the product exactly does the same thing what is supposed to do. In the testing stage following goals are tried to achieve: -

- To affirm the quality of the project.
- To find and eliminate any residual errors from previous stages.
- To validate the software as a solution to the original problem.
- To provide operational reliability of the system.

Testing Methodologies

There are many different types of testing methods or techniques used as part of the software testing methodology. Some of the important testing methodologies are:

5.1 SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

5.2 TEST DATA AND OUTPUT

5.2.1 UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

5.2.2 FUNCTIONAL TESTS

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

- Performance Test
- Stress Test
- Structure Test

5.2.3 PERFORMANCE TEST

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

5.2.4 STRESS TEST

Stress Test is those test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

5.2.5 STRUCTURED TEST

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

- Exercise all logical decisions on their true or false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to assure their validity.
- Checking attributes for their correctness.
- Handling end of file condition, I/O errors, buffer problems and textual errors in output information

5.2.6 INTEGRATION TESTING

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

5.3 TESTING TECHNIQUES / TESTING STRATERGIES

5.3.1 TESTING

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet-undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program,

string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

5.3.1.1 WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Cyclometric complexity
- Deriving test cases
- Graph matrices Control

5.3.1.2 BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing

5.3.2 SOFTWARE TESTING STRATEGIES:

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

- Testing begins at the module level and works “outward” toward the integration of the entire computer based system.
- Different testing techniques are appropriate at different points in time.
- The developer of the software and an independent test group conducts testing.
- Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

5.3.2.1 INTEGRATION TESTING:

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together. The problem of course, is “putting them together”- interfacing. There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

5.3.2.2 PROGRAM TESTING:

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

5.3.2.3 SECURITY TESTING:

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

5.3.2.4 VALIDATION TESTING

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

- * The function or performance characteristics confirm to specifications and are accepted.
- * A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic

5.3.2.5 USER ACCEPTANCE TESTING

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required. This is done in regarding to the following points.

- Input screen design.
- Output screen design.

OUTPUT SCREENSHOTS



Fig 4.1 Signup page



Fig 4.2 Login page



Fig 4.3 User Profile page



Fig 4.4 User Profile page



Fig 4.5 Movie review post







 <p>Kanye West, 'Stronger' ❤️ 9.1 2007</p> <p>Explaining the tighter, broader-reaching songs on his third album, Graduation, Kanye West said, "I applied a lot of the things I learned on tour [in 2004] with U2 and the Rolling Stones, about songs that rock stadiums. And they worked!" West found the inspiration for his most grandiose statement to date.</p>	 <p>The Supremes, 'Baby Love' ❤️ 8.5 1964</p> <p>Diana Ross wasn't the strongest vocalist in the Supremes, but as the Motown production team discovered, when she sang in a lower register, her voice worked its sultry magic.</p>	 <p>Townes Van Zandt, 'Pancho and Lefty' ❤️ 8.3 1972</p> <p>When the Riddler, a sadistic serial killer, begins murdering key political figures in Gotham, Batman is forced to investigate the city's hidden corruption and question his family's involvement.</p>	 <p>Lizzo, 'Truth Hurts' ❤️ 7.5 2017</p> <p>"That song is my life and its words are my truth," Lizzo wrote at the time. She had to tack on a writing credit to British singer Nina Lianess, who had tweeted its iconic line "I just took a DNA test, turns out I'm 100 percent that bitch," but the power of this gale-force breakup banger was pure Lizzo, uproariously swaggering and endearingly soulful.</p>	 <p>Miles Davis, 'So What' ❤️ 7.1 1959</p> <p>It's likely that no song on this list has soundtrack more dinner parties than Kind of Blue's warm, welcoming first track. But at the time it was a jarring departure, trading bebop chord changes for a more open-ended modal style.</p>	 <p>Lil Nas X, 'Old Town Road' ❤️ 6.9 2019</p> <p>Montero Hill was an Atlanta college dropout sleeping on his sister's couch and looking to break into music when he came across a track he liked by a Dutch 19-year-old called Youngkio that was based around a banjo sample from a Nine Inch Nails track.</p>
---	--	---	--	---	---

Fig 4.6 Music recommendation list

Entertainment recommendation system

					
<p>The Dark Knight ☆ 9.1 Action/Crime</p> <p>When the menace known as the Joker wreaks havoc and chaos on the people of Gotham, Batman must accept one of the greatest psychological and physical tests of his ability to fight injustice.</p>	<p>Spider-Man: No Way Home ☆ 8.5 Action/Adventure</p> <p>A paired murderer returns to the Staten Island home of his youth, where a poignant friendship blossoms with the restrained wife of his temperamental—and jealous—brother.</p>	<p>The Batman ☆ 8.3 Action/Adventure</p> <p>When the Riddler, a sadistic serial killer, begins murdering key political figures in Gotham, Batman is forced to investigate the city's hidden corruption and question his family's involvement.</p>	<p>Death on the Nile ☆ 7.5 Mystery</p> <p>While on vacation on the Nile, Hercule Poirot must investigate the murder of a young heiress.</p>	<p>Fantastic Beasts: The Secrets of Dumbledore ☆ 7.1 Fantasy/Adventure</p> <p>The third instalment of "Fantastic Beasts and Where to Find Them," which follows the continuing adventures of Newt Scamander.</p>	<p>Murder on the Orient Express ☆ 6.9 Mystery/Crime</p> <p>When a murder occurs on the train on which he's travelling, celebrated detective Hercule Poirot is recruited to solve the case.</p>
					

Fig 4.7 Movie recommendation list

6. Conclusion

Recommender systems main motive is to predict users interests and recommend product items that they may find interesting. They're among the foremost powerful machine learning systems that online retailers implement so as to drive sales. Data required for recommender systems stems from explicit user ratings after watching a movie or taking note of a song, from implicit program queries and purchase histories, or from other knowledge about the users/items themselves. Recommender systems use the opinions of a community of users to assist individuals in that community more effectively identify content of interest from a potentially overwhelming set of choices . Lately, the demand for recommendation services have severely increased thanks to the huge flow of latest content on to the web . In order for users to seek out the content they desire, competent recommendation services are extremely helpful. It's challenging for a user to seek out the acceptable movies suitable for his/her tastes. Different users like different movies or actors. It is important to seek out a way of filtering irrelevant movies and/or find a group of relevant movies. Therefore, automated recommendation services are implemented to ease this task.

7. Future Scope

The recommender system has developed for many years, which ever entered a low point. In the past few years, the development of machine learning, large-scale network and high performance computing is promoting new development in this field. In the proposed approach, It has considered Genres of movies but, in future we can also consider age of user as according to the age movie preferences also changes, like for example, during our childhood we like animated movies more as compared to other movies. There is a need to work on the memory requirements of the proposed approach in the future. The proposed approach has been implemented here on different movie datasets only. It can also be implemented on the Film Affinity and Netflix datasets and the performance can be computed in the future.

We will consider the following aspects in future work.

- Use collaborative filtering recommendation. After getting enough user data, collaborative filtering recommendation will be introduced. As we discussed , collaborative filtering is based on the social information of users, which will be analyzed in the future research.
- Introduce more precise and proper features of movie. Typical collaborative filtering recommendation use the rating instead of object features. In the future we should extract features such as color and subtitle from movie which can provide a more accurate description for movie.
- Introduce user dislike movie list. The user data is always useful in recommender systems. In the future we will collect more user data and add user dislike movie list. We will input dislike movie list into the recommender system as well and generate scores that will be added to previous result. By this way we can improve the result of recommender system.

- Introduce machine learning. For future study, dynamic parameters will be introduced into recommender system, we will use machine learning to adjust the weight of each feature automatically and find the most suitable weights.
- Make the recommender system as an internal service. In the future, the recommender system is no longer a external website that will be just for testing. We will make it as an internal APIs for developers to invoke. Some movie lists in the website will be sorted by recommendation.



8. References

- [1] R. Singh, S. Maurya, T. Tripathi, T. Narula, and G. Srivastav, "Movie recommendation system using cosine similarity and knn," pp. 2249–8958, 06 2020.
- [2] M. Kumar, D. Yadav, A. Singh, and V. K. Gupta, "Article: A movie recommender system: Movrec," International Journal of Computer Applications, vol. 124, pp. 7–11, August 2015.
- [3] L. M. de Campos, J. M. Fernández-Luna, J. F. Huete, and M. A. Rueda-Morales, "Combining content-based and collaborative recommendations: A hybrid approach based on bayesian networks," International Journal of Approximate Reasoning, vol. 51, no. 7, pp. 785 – 799, 2010.
- [4] E. A. S. Harpreet Kaur Virk, Er.Maninder Singh, "Analysis and design of hybrid online movie recommender system," International Journal of Innovations in Engineering and technology, 2015.
- [5] U. Gupta and N. Patil, "Recommender system based on hierarchical clustering algorithm chameleon," International Advance Computing Conference (IACC), 2015.
- [6] U. Kuzelewska, "Clustering algorithms in hybrid recommender system on movielens data," Studies in Logic, Grammar and Rhetoric, vol. 37, 01 2014.
- [7] C. Chiru, C. Preda, V. Dinu, and M. Macri, "Movie recommender system using the user's psychological

- profile,” in 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 93–99, 2015.
8. [8] N. Hug, “Surprise: A python library for recommender systems,” *Journal of Open Source Software*, vol. 5, p. 2174, 08 2020.
 9. [9] G. Linden, B. Smith, and J. York, “Amazon.com recommendations: item-to-item collaborative filtering,” *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.
 10. [10] X. Peng, S. Liangshan, and L. Xiuran, “Improved collaborative filtering algorithm in the research and application of personalized movie recommendations,” in 2013 Fourth International Conference on Intelligent Systems Design and Engineering Applications, pp. 349–352, 2013.
 11. [11] M. Munoz-Organero, G. A. Ramirez-Gonzalez, P. J. Munoz-Merino, and C. Delgado Kloos, “A collaborative recommender system based on space-time similarities,” *IEEE Pervasive Computing*, vol. 9, no. 3, pp. 81–87, 2010.
 12. [12] M. Y. H. Al-Shamri and K. K. Bharadwaj, “Fuzzy-genetic approach to recommender systems based on a novel hybrid user model,” *Expert Systems with Applications*, vol. 35, no. 3, pp. 1386 – 1399, 2008.
 13. [13] Hu Jinming, “Application and research of collaborative filtering in e-commerce recommendation system,” in 2010 3rd International Conference on Computer Science and Information Technology, vol. 4, pp. 686–689, 2010.
 14. [14] C. Davidsson and S. Moritz, “Utilizing implicit feedback and context to recommend mobile applications 301177 REFERENCES from first use,” in *Proceedings of the 2011 Workshop on Context-Awareness in Retrieval and Recommendation, CaRR '11, (New York, NY, USA)*, p. 1922, Association for Computing Machinery, 2011.
 15. [15] A. Bilge, C. Kaleli, I. Yakut, I. Gunes, and H. Polat, “A survey of privacy-preserving collaborative filtering schemes,” *International Journal of Software Engineering and Knowledge Engineering*, vol. 23, 02 2014.
 16. [16] J. A. Calandrino, A. Kilzer, A. Narayanan, E. W. Felten, and V. Shmatikov, ““you might also like:” privacy risks of collaborative filtering,” in 2011 IEEE Symposium on Security and Privacy, pp. 231–246, 2011.
 17. [17] G. Rodríguez, “Introduction to recommender systems in 2019,” 2019. Available: <https://tryolabs>.

com/blog/introduction-to-recommender-systems/.

18. [18] L. Iaquinta, A. Gentile, P. Lops, M. de Gemmis, G. Semeraro, A hybrid content-collaborative recommender system integrated into an electronic performance support system, in: Hybrid Intelligent Systems, 2007. HIS 2007. 7th International Conference on, IEEE, 2007, pp. 47–52.
19. [19] M. Papagelis, D. Plexousakis, Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents, *Engineering Applications of Artificial Intelligence* 18 (7) (2005) 781–789.
20. [20] J. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in: *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, Morgan Kaufmann Publishers Inc., 1998, pp. 43–52.
21. [21] B. M. Sarwar, G. Karypis, J. Konstan, J. Reidl, Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering, in: *Proceedings of the 5th International Conference on Computer and Information Technology (ICIT)*, 2002.
22. [22] Y. Liangxing, D. Aihua, Hybrid product recommender system for apparel retailing customers, in: *Information Engineering (ICIE), 2010 WASE International Conference on*, Vol. 1, IEEE, 2010, pp. 356–360.
23. [23] S. Yun, S. Youn, Recommender system based on user information, in: *Service Systems and Service Management (ICSSSM), 2010 7th International Conference on*, IEEE, 2010, pp. 1–3.
24. [24] H. Kadima, M. Malek, Toward ontology-based personalization of a recommender system in social network, in: *Soft Computing and Pattern Recognition (SoCPaR), 2010 International Conference of*, IEEE, 2010, pp. 119–122.
25. [25] T. Tang, P. Winoto, K. Chan, Scaling down candidate sets based on the temporal feature of items for improved hybrid recommendations, *Intelligent Techniques for Web Personalization* (2005) 169–186.
26. [26] T. Lee, Y. Park, Y. Park, A time-based approach to effective recommender systems using implicit feedback,
27. [27] Y. Koren, Collaborative filtering with temporal dynamics, *Communications of the ACM* 53 (4) (2010) 89–97.