A

Project Report

on

Movie Recommendation System

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2023-24

in

Computer Science and Engineering

By

Mohd Adil (2000290100087)

Mohd Asif Ansari(2000290100089)

Mayank Garg(2000290100086)

Under the supervision of

Ms. Nishu Bansal

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2024

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature

Name: Mohd Adil

Roll No.: 2000290100087

Signature

Name: Mohd Asif Ansari

Roll No.: 2000290100089

Signature

Name: Mayank Garg

Roll No.: 2000290100086

Date:

CERTIFICATE

This is to certify that Project Report entitled "Movie Recommendation System" which is submitted by Mohd Adil , Mohd Asif Ansari and Mayank Garg in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science & Engineering of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

Supervisor Name **Dr. Vineet Sharma**

(Designation) **(HoD-Computer Science & Engineering)**

Date:

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to Ms. Nishu Bansal, Department of Computer Science & Engineering, KIET, Ghaziabad, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Vineet Sharma, Head of the Department of Computer Science & Engineering, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members, especially faculty/industry person/any person, of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Date:

Signature:
Name : Mohd Adil
Roll No.: 2000290100087

Signature:
Name : Mohd Asif Ansari
Roll No.: 2000290100089

Signature:
Name : Mayank Garg
Roll No.: 2000290100086

ABSTRACT

Each of us needs entertainment to recharge our spirits and energy in this fast-paced world. Our confidence for work is restored by entertainment, and we work more ardently as a result. We can watch our favorite movies or listen to our favorite music to reenergize ourselves. Since finding chosen movies will take more and more time, which one cannot afford to waste, we can use more reliable movie recommendation algorithms to watch good movies online. In this paper, a hybrid approach that combines content-based filtering, collaborative filtering, using Support Vector Machine as a classifier, and genetic algorithm is presented in the proposed methodology. Comparative results are shown, showing that the proposed approach shows an improvement in the accuracy quality, and scalability of the movie recommendation system than the pure approaches in three areas: accuracy, quality, and scalability. The advantages of both approaches are combined in a hybrid strategy, which also seeks to minimize their negative aspects.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

A recommendation system, sometimes known as a recommendation engine, is a paradigm for information filtering that aims to anticipate user preferences and offer suggestions in accordance with these preferences. These technologies are now widely used in a variety of industries, including those that deal with utilities, books, music, movies, television, apparel, and restaurants. These systems gather data on a user's preferences and behavior, which they then employ to enhance their future suggestions. There are many various kinds of movies, such as those meant for amusement, those meant for teaching. children's animation movies, horror movies, and action movies. Watching movies is relaxing. Distinguished by their various genres, such as humor, suspense, animation, action, etc. Another approach to differentiate between movies is to look at their release year, language, director, etc. When watching movies online, there are many to choose from in our list of top picks. We may find our favourite movies among all of these different kinds of movies with the aid of movie recommendation systems, which saves us the stress of having to spend a lot of time looking for our preferred movies. As a result, it is essential that the system for suggesting movies to us is very trustworthy and gives us recommendations for the films that are either most similar to or identical to our tastes.

Recommendation systems are being used by a lot of businesses to improve customer interaction and the purchasing experience. The most significant advantages of recommendation systems are client happiness and income.[1]

1.2 PROJECT DESCRIPTION

1.2.1 Problem Statement:

In today's fast-paced world, where our lives are filled with deadlines, responsibilities, and constant stimulation, it's essential to have avenues for relaxation and enjoyment. For many, movies serve as a much-needed escape, a way to unwind, and a source of entertainment that allows us to recharge our spirits and energy.

However, as the digital landscape continues to expand, so too does the vast ocean of available movie content. With numerous streaming platforms offering thousands upon thousands of titles, the task of finding the perfect movie to watch becomes increasingly daunting. Browsing through endless lists, scrolling through recommendations that may or may not align with our tastes, and wasting precious time on selections that ultimately disappoint - these are the frustrations that plague movie enthusiasts everywhere.

Moreover, traditional recommendation systems employed by streaming services often fall short of meeting users' expectations. These systems typically rely on simplistic algorithms that consider factors like genre or popularity but fail to capture the nuanced preferences and individual tastes of users. As a result, users are left with recommendations that feel generic and uninspired, lacking the personal touch that makes movie-watching truly enjoyable.

Recognizing these challenges, our project sets out to tackle the shortcomings of existing movie recommendation systems and revolutionize the way users discover and enjoy movies. Our aim is to develop a cutting-edge recommendation system that not only alleviates the frustrations associated with movie selection but also enhances the overall movie-watching experience for users.

To achieve this goal, we propose a novel hybrid approach that integrates multiple recommendation techniques, including content-based filtering, collaborative filtering, and advanced machine learning algorithms. By combining these methods, we seek to leverage the strengths of each approach while mitigating their individual limitations.

One key aspect of our approach is the incorporation of Support Vector Machine (SVM) as a classifier. SVM is a powerful machine learning algorithm that excels at classifying data points into different categories. By applying SVM to the movie recommendation task, we aim to improve the accuracy and precision of our recommendations, ensuring that users receive suggestions that closely align with their preferences.

Additionally, we employ genetic algorithms to optimize the performance of our recommendation system. Genetic algorithms are a class of optimization algorithms inspired by the process of natural selection. By iteratively evolving and refining the parameters of our recommendation model, we aim to enhance its ability to adapt to changing user preferences and deliver more relevant and personalized recommendations over time.[2]

1.2.2 Objective of the Projects

- **Enhanced Accuracy:** By combining multiple recommendation techniques and leveraging advanced machine learning algorithms like SVM, we aim to improve the accuracy of our recommendation system, ensuring that users receive suggestions that closely match their preferences.
- **Personalized Recommendations:** We strive to provide users with personalized recommendations that reflect their unique tastes, interests, and viewing history. By analyzing user behavior and preferences, our system will tailor its recommendations to each individual user, ensuring a more engaging and satisfying movie-watching experience.
- **Scalability and Efficiency:** In addition to improving accuracy and personalization, we aim to enhance the scalability and efficiency of our recommendation system. By optimizing the performance of our algorithms and infrastructure, we will ensure that our system can handle large volumes of users and movie data without sacrificing speed or reliability.[4]

1.2.3 Scope of the Project

The project's scope encompasses the conceptualization, development, implementation, and evaluation of a comprehensive movie recommendation system. It aims to tackle the limitations of current recommendation systems by utilizing advanced techniques and methodologies to offer more precise, personalized, and efficient recommendations to users.

Key components included in the project scope are:

Algorithm Development: The project involves the creation and refinement of innovative recommendation algorithms, including content-based filtering, collaborative filtering, Support Vector Machine (SVM) classification, and genetic algorithms. These algorithms will form the foundation of the recommendation system, enabling it to analyze user preferences and produce customized movie recommendations.

Data Collection and Preprocessing: An essential aspect of the project is gathering and preprocessing movie-related data from various sources, such as user ratings, movie metadata, and user preferences. This data will be carefully curated, cleansed, and standardized to ensure compatibility with the recommendation algorithms and to uphold data integrity throughout the recommendation process.

System Architecture: The project encompasses the design and implementation of the system architecture supporting the movie recommendation system. This involves building the backend infrastructure for data storage, retrieval, and processing, as well as the frontend interface for user interaction and recommendation presentation.

User Interface Design: Designing a user-friendly interface is crucial for facilitating user engagement and interaction with the recommendation system. The project will focus on creating an intuitive and visually appealing user interface that allows users to input preferences, explore recommendations, and provide feedback on suggested movies.

Evaluation and Testing: Throughout the project's development lifecycle, rigorous evaluation and testing will be conducted to assess the performance, accuracy, and scalability of the recommendation system. This includes running experiments, gathering user feedback, and comparing against existing recommendation systems to validate the effectiveness of the proposed approach.

Documentation and Reporting: Thorough documentation will be generated throughout the project to record design choices, implementation details, and evaluation outcomes. This documentation will serve as a reference for future development endeavors and will be compiled into a final project report for dissemination.

The project's scope is ambitious, aiming to deliver an advanced movie recommendation system that establishes new benchmarks for accuracy, personalization, and user experience. By addressing the inherent challenges of conventional recommendation systems and leveraging advanced methodologies, the project aims to offer users a seamless and enjoyable movie-watching experience.[5]

1.2.4 Methodology used for Movie Recommendation

1. **Content-Based Filtering:**

   Content-based filtering methods are done based on user characteristics. This method is used in situations where data is known on an item such as name, location, or description and not on the user. It predicts the items based on user's information and completely ignores contributions from other users as with the case of collaborative techniques. It uses the data that is provided by the user either explicitly or implicitly. When the user provides more content-based filtering mechanisms actions on the recommendations such as content- based recommender the engine becomes more and more accurate.[6]

2. Agile Methodology:

- **Collecting the data sets:**

  Collecting all the required data set from Kaggle web site in this project we require movie.csv, ratings.csv, users.csv.

- **Data Analysis:** Make sure that that the collected data sets are correct and analyzing the data in the csv files ie. checking whether all the column Felds are present in the data sets.

- **Algorithms:** in our project we have only two algorithms one is cosine similarity and other is single valued decomposition are used to build the machine learning recommendation model.

- **Training and Testing the model:** once the implementation of algorithm is completed. we have to trainthe model to get the result. We have tested it several times the model is recommend different set of movies to different users.

- **Improvements in the project:** In the later stage we can implement different algorithms and methods for better recommendation.[7]

CHAPTER 2

LITERATURE REVIEW

2.1 Movie Recommendation System

Movie recommendation systems are designed to suggest movies to users based on their preferences, viewing history, and other relevant factors. These systems aim to provide personalized recommendations that align with the user's tastes and interests. They have become increasingly popular with the rise of streaming platforms and the abundance of available content.

One common approach used in movie recommendation systems is collaborative filtering. This technique analyzes user behavior and preferences to identify patterns and similarities between users. By leveraging this information, the system can recommend movies that users with similar tastes have enjoyed. Collaborative filtering can be further divided into two types: user-based and item-based. User-based collaborative filtering suggests movies based on the preferences of similar users, while item-based collaborative filtering recommends movies based on the similarity of movie attributes.

Another approach is content-based filtering, which focuses on analyzing the content of movies to make recommendations. It involves extracting features such as genre, actors, directors, and plot summaries to create profiles for movies. Based on these profiles, the system can recommend movies that have similar attributes to the ones the user has enjoyed in the past.

Hybrid recommendation systems combine multiple techniques, such as collaborative filtering and content-based filtering, to improve the accuracy and diversity of recommendations. These systems aim to overcome the limitations of individual approaches and provide more robust and personalized suggestions.

In recent years, advancements in machine learning and artificial intelligence have further enhanced movie recommendation systems. Techniques like deep learning and natural language processing have been applied to improve the accuracy and relevance of recommendations. Additionally, the use of contextual information, such as time of day or location, can also influence the recommendation process.

While movie recommendation systems have proven to be effective in helping users discover new movies and enhancing their viewing experience, there are still challenges to overcome. These include the "cold start" problem, where recommendations for new users or movies with limited data can be less accurate, and the issue of serendipity, where users may want recommendations that are outside their usual preferences.

Overall, movie recommendation systems have become an integral part of the entertainment industry, providing users with personalized and diverse movie suggestions. With ongoing advancements in technology and research, these systems continue to evolve, offering even more tailored recommendations to enhance the movie-watching experience.[8]

2.2 Types of Recommendation System

Systems that propose things, create playlists, find matches, and do much more are known as recommender systems. User-item interactions and characteristic information are key components of recommender systems operation. Information about the user and the items constitutes characteristic information, whereas information about user-item interactions includes ratings, the volume of purchases, user likes, and many other things. Based on this, a collaborative filtering, content-based filtering, or hybrid filtering approach can be used to create the recommendation system.[9]

1. Content-Based Filtering:

Content Based Filtering system: In the content based filtering method we compare the different items with the user's interest profile. So basically the user profile holds the content that is is much more matching to use the form of the features. The previous actions or for the feedback is taken into account a generally takes into account the description of the content that has been edited by the users of different choices. Considering that example where a person buys some favourite item 'M' but item has been sold out and as a result he has to buy the item 'N' on the recommendation of some person as and 'N' has same type of matching features that the first one possesses. So this is simply the content based filtering which is demonstrated below

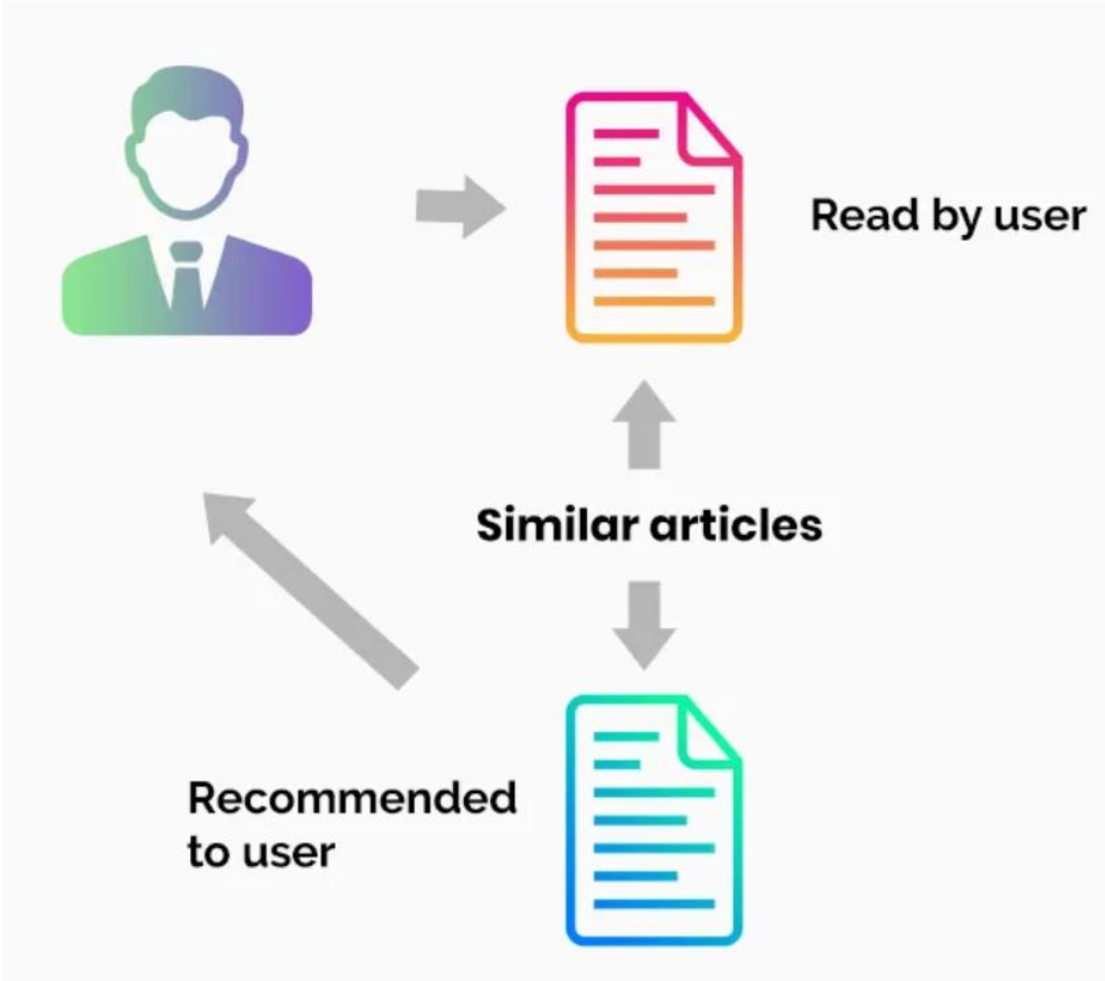


Fig:-Content Based Filtering Method

So here numeric quantity that will be used to calculate the similarity between the two types of movies will be cosine similarity and we will calculate the score it is very very fast to calculate the the magnitude of the score which is obtained through the cosine similarity

$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \cdot \sqrt{\sum\limits_{i=1}^{n} B_i^2}},$$

where A_i and B_i are the ith components of vectors $\mathbf{A}$ and $\mathbf{B}$, respectively.

The steps involved in getting the movie recommendation are as below:

- Having the title find the index of that movie
- Calculate the cosine similarity scores for all the movies
- Arranging the scores in the order of highest priority first that is ascending order
- And then shorting the list based on the similarity scores.
- Getting the first 10 element of the list excluding the first one as it is the movie name in itself.
- Getting the top elements

Repeating above steps we will find the top movies based on the distances which it can get the best possible recommendation, the movies that have high probability of being liked by the general set of users will be displayed to the user by the recommender in the end and then in another technique we will try to find the users with different interest using the information collected through different activities an Indian in collaborative filtering will test all those users which have same type of interests to get the final set of movies to be recommended to the users individually.

The cosine similarity is the cause of the angle between the two vectors where the vectors are non zero and the inner product space it is described as the dot product of the two vectors divide by by the product of the euclidean magnitude. In most cases cosine similarity isused to get preffered recommendations for users.

This method simply use the cosine distance between the vectors and then it uses similarity to calculate the score and then the preference of the user. For example movie with actors which define number of user likes and only few actors which a group of users don't like so we believe plotting a good sign angle between the user and the movie vectors which will generally be a large positive fraction, so angle is almost closely to be zero small distance of cosine will be present between the two vectors. Better metric somehow like the movie and cosine distance is large then the cosine similarity fails in this case we will approach in new method call decision tree to refine the recommender system. this method generally contains levels baby can apply some conditions in a classification approach of refining the recommender system which try to find out if a user wants to what movie or not at all.

The advantages of the content based filtering are:

- **Personalization**: Content-based filtering tailors recommendations based on a user's past interactions and preferences. By analyzing the features or attributes of items a user has liked, it can provide recommendations that are highly relevant to their individual tastes.
- **Transparency**: Content-based filtering typically relies on explicit features or attributes of items, such as genre, actors, directors, or keywords. This transparency in recommendation generation helps users understand why certain items are being recommended to them, fostering trust and satisfaction with the recommendation system.
- **No Cold Start Problem**: Content-based filtering can make recommendations for new or niche items without historical user data. Since recommendations are based on item features rather than user-item interactions, there is no "cold start problem" where new items have insufficient data for recommendation.
- **Less Susceptible to Data Sparsity**: Content-based filtering does not heavily rely on user-item interactions, making it less susceptible to data sparsity issues. Even if there is limited data on user preferences, the system can still generate relevant recommendations by analyzing item attributes.
- **Serendipitous Discovery**: While content-based filtering primarily recommends items similar to those a user has already liked, it also has the potential to introduce serendipitous discovery. By considering various item attributes, it can recommend items that share certain characteristics with the user's preferences but also offer novelty, introducing users to new and potentially interesting content.
- **Domain Independence**: Content-based filtering can be applied across different domains with minimal domain-specific customization. As long as relevant features can be extracted from items, the same recommendation approach can be employed across various types of content, from movies and music to articles and products.

The disadvantages of the content based filtering are:

- Think of content-based filtering like a friend who suggests movies based on ones you've already seen and liked. They're good at picking movies similar to your favorites. But sometimes, you might want to try something totally new and surprising – something you didn't even know you'd like! Content-based filtering doesn't do that well because it mainly looks at what you've already liked and suggests similar stuff. So, you might miss out on cool movies that are different from what you usually watch.

- Over-Specialization: Content-based filtering tends to recommend items that closely match a user's past preferences. This can lead to a phenomenon known as "filter bubble" or "echo chamber," where users are only exposed to content that reinforces their existing interests and viewpoints. As a result, users may miss out on diverse perspectives and experiences.

- Limited Understanding of User Context: Content-based filtering relies on the features of items and does not take into account contextual factors such as user mood, intent, or social context. This can lead to recommendations that are technically relevant but not suitable for the user's current situation or preferences.

- Dependency on Feature Quality: The effectiveness of content-based filtering heavily depends on the quality and richness of item features or attributes. If the available features are sparse or poorly representative of item characteristics, the system may struggle to generate accurate and relevant recommendations.

- Difficulty Handling Dynamic Preferences: Users' preferences and interests can evolve over time, but content-based filtering may struggle to adapt to these changes effectively. Since recommendations are based on past interactions and feature similarities, the system may continue to recommend items that align with outdated preferences, leading to decreased relevance and user satisfaction.

- Scalability Challenges: Content-based filtering can face scalability challenges, especially when dealing with large and diverse item catalogs. Analyzing and processing item features for recommendation generation can become computationally intensive, potentially leading to performance issues and increased resource requirements as the size of the catalog grows.[10]

2.Collaborative based Filtering:

Content based filtering suffer from various limitations which is only capable of the suggesting movies having only one type of users preferences and then unable to provide recommendations in case of genres . However collaborative filtering based system provides much complexibility in finding the record between the similarity of user and the the likes of the users having similar interest. For measuring the similarity of users views cosine similarity or pearson's correlation. Taking example in the below Matrix every row has a user with column corresponding to the movies having the same similarity it also has the ratings of different movies which the user have given to each movie has a target user. All the the collaborative filtering in case of user based is simple but it has also drawbacks the biggest challenges that the choices of the users where is with time. Pre computing the Matrix orphan let the problem of lower performance. So we can use the item based collaborative filtering which basically considers the items based on the similarity with the items and that it it find the similar matches with the target users the same similarity coefficients suggest pearson's correlation or Cosine similarity can be used. Item based collaborative filtering is most static in nature .

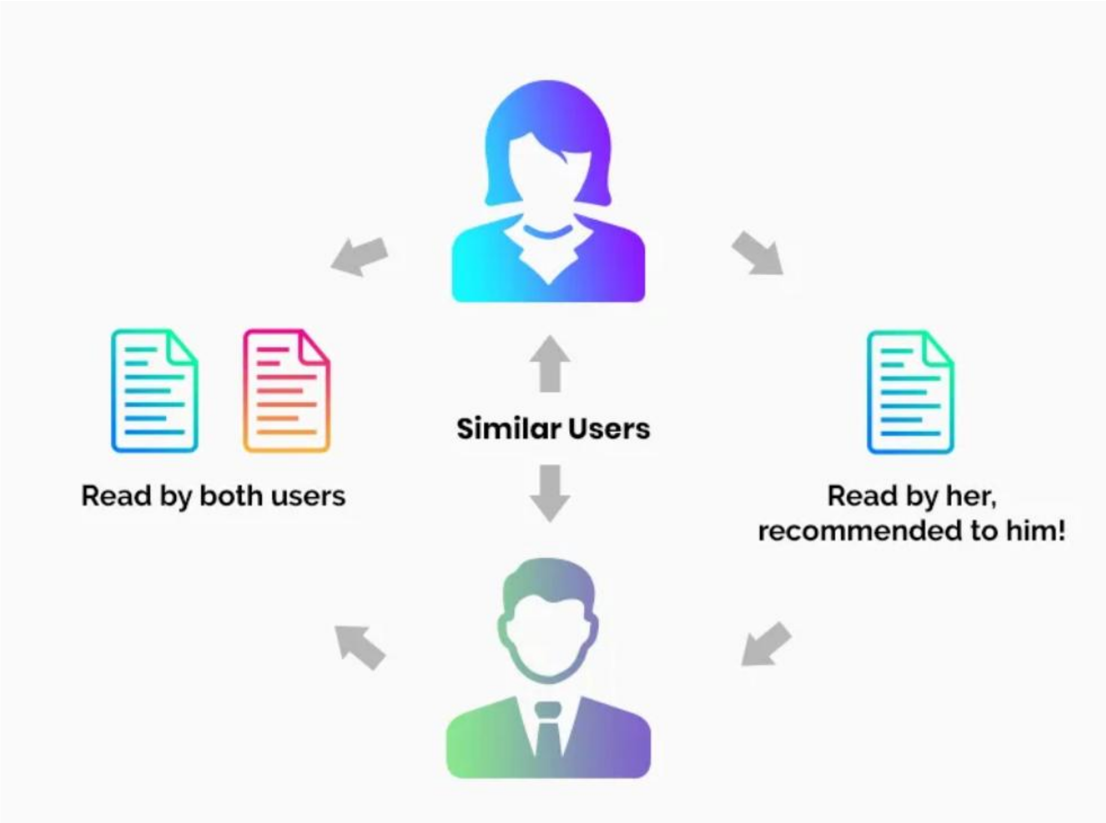


Fig:-Collaborative based Filtering

Like blow example only one user which has related both Matrix and Titanic so similarity which stands between them is only one . There may be cases where we have millions of users and the similarity between those two different movies is very high as they have same rank for the user who have rated them both.

In collaborative filtering try to find out the users have which have name interest and similar likes. In this case we don't use features of the item to recommend it but we use the classification of users into clusters of similar types and then seperate each cluster into the order of the preference of the user.

We can also use the cosine distance here which takes into account the users with the similar interest greater the cosine small angle between the two user. Here we simply use the utility matrix we can assign the zero value to the sparse columns forming the calculations easy. Item based Colaborative filtering is preferred in general because it takes into account the movie instead of the number of users which further only make the classification of the movies and user much easier. Hence the user based collaborative filtering is not preferred because it's simply only takes the user's into account and ignore the sparse values which creates the issues in bringing out the performance of the recommender system.

	The Avengers	Sherlock	Transformers	Matrix	Titanic	Me Before You	Similarity
A	2		2	4	5		N/A
B	5		4			1	
C			5		2		
D		1		5		4	
E			4			2	
F	4	5		1			N/A1

Table 2.1 Collaborative based Filtering

Since user A and Foo not share any movie ratings in common with user E, their similiantes with user E are not defined in Pearson Conelation. Therefore, we only need to consider user B, C, and D. Based on Pearson Correlation, we can compute the following similarity.

	The Avengers	Sherlock	Transformer	Matrix	Titanic	Me Before You	Similarity
A	2		2	4	5		N/A
B	5		4			1	0.87
C			5		2		1
D		1		5		4	-1
E			4			2	1
F	4	5		1			N/A

Table 2.2 - User Based Collaborative Filtering

So now we want our recommendation problem to be converted into a Optimisation problem . The most preferred common metric is a root mean square error(RMSE). Better the performance lower will be The RMSE value.

Advantages of collaborative filtering based systems are:

- **No Need for Item Metadata**: Unlike content-based filtering, which relies on detailed information about items, collaborative filtering primarily uses user-item interaction data. This means you don't need extensive metadata for each item in the system, which can simplify data collection and maintenance.
- **Effective for Cold Start Problem**: Collaborative filtering can provide recommendations for new users or items with minimal historical data. By analyzing similarities between users or items based on their interactions, the system can still generate relevant recommendations even when there's limited data available.
- **Discovery of Unexpected Preferences**: Collaborative filtering can uncover unexpected preferences or hidden patterns in user behavior. By analyzing similarities between users with overlapping preferences, the system can recommend items that one user likes but another user with similar tastes has already interacted with, leading to serendipitous discovery.

- **Scalability**: Collaborative filtering algorithms can scale well with large datasets and diverse item catalogs. They can efficiently analyze vast amounts of user-item interaction data to generate personalized recommendations for individual users without significant performance degradation.
- **Adaptability to Evolving Preferences**: Collaborative filtering can adapt to changes in user preferences over time. As users interact with the system and provide feedback, the algorithm continuously updates its recommendations to reflect the evolving interests and preferences of users.
- **Effective for Long-Tail Items**: Collaborative filtering is well-suited for recommending long-tail items, which are niche or less popular items that may not have extensive metadata or user interactions. By leveraging collective user behavior, the system can surface these items to users who might be interested in them, increasing overall catalog engagement.
- **Social and Community Influence**: Collaborative filtering can capture social and community influences in user preferences. By considering similarities between users with social connections or shared interests, the system can recommend items that are popular within specific social circles or communities, enhancing user engagement and satisfaction.

Disadvantages of collaborative filtering based systems are:

- Cold Start Problem for New Users and Items: Collaborative filtering struggles with providing accurate recommendations for new users who haven't provided enough interaction data or for new movies that haven't been rated yet. This limitation can hinder the system's ability to offer personalized recommendations to users who are just starting to use the platform or to promote newly released movies.

- Sparsity and Data Scalability: Collaborative filtering relies heavily on user-item interaction data, and in systems with a large number of users and items, the data matrix can become sparse, meaning most users have not rated most items. This sparsity can lead to challenges in generating accurate recommendations, especially for niche or less popular movies with limited user interactions.

- Popularity Bias: Collaborative filtering tends to recommend popular items more frequently than niche or less popular ones. This can result in a "rich-get-richer" phenomenon, where popular movies receive even more exposure, while smaller or indie films may struggle to gain visibility, leading to a lack of diversity in recommendations.
- Limited Interpretability: Collaborative filtering models often lack transparency in how recommendations are generated. Since recommendations are based on patterns in user-item interactions, users may not understand why a particular movie is being recommended to them, which can reduce their trust in the system and their willingness to engage with the recommendations.
- Cold Start Problem for New Users and Items: Collaborative filtering struggles with providing accurate recommendations for new users who haven't provided enough interaction data or for new movies that haven't been rated yet. This limitation can hinder the system's ability to offer personalized recommendations to users who are just starting to use the platform or to promote newly released movies.
- Limited Explanation and Serendipity: Collaborative filtering tends to recommend items similar to those a user has already liked or similar to what similar users have liked. This can lead to a lack of diversity in recommendations and may not expose users to new or unexpected movies outside their usual preferences, limiting serendipitous discovery.
- When using collaborative filtering, the system needs to gather and study how users interact with movies to make recommendations. But this means collecting data about what users like, which can include personal preferences. This raises worries about how that data is kept safe and whether it might be seen or used by people who shouldn't have access to it. So, there's a concern about keeping users' private information secure and making sure it's not used inappropriately.[11]

3.Hybrid Approach:

It is simply a mixture of content based filtering and collaborative based filtering methods where we will take the input as the the userid and the title of the movie and the output will be e the similar movies shorted by the particular users based on the expected ratings. Expected ratings are calculated internally where the ideas from content and collaborative filtering are used to build a engine where movies are suggested to the particular user and then estimation of the ratings takes place.

In the comparisons section below we will see how movies are determined through the hybrid technique of filtering where we have both used content based method as well as the collaborative based filtering method. It is clear that hybrid filtering method is is good in most of the cases and scenarios where it is difficult to distinguish or get the accuracy which the users can get the recommended movies.[17]

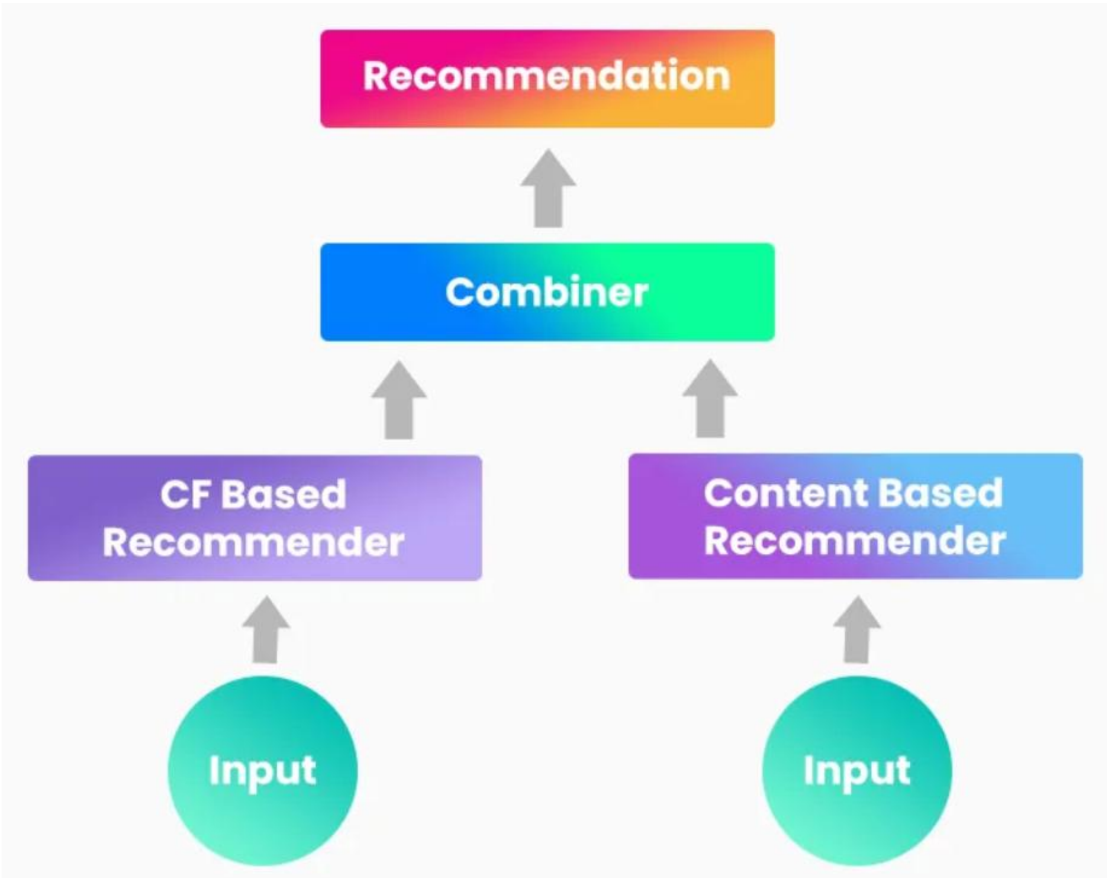


Fig:- Hybrid Approach

CHAPTER 3

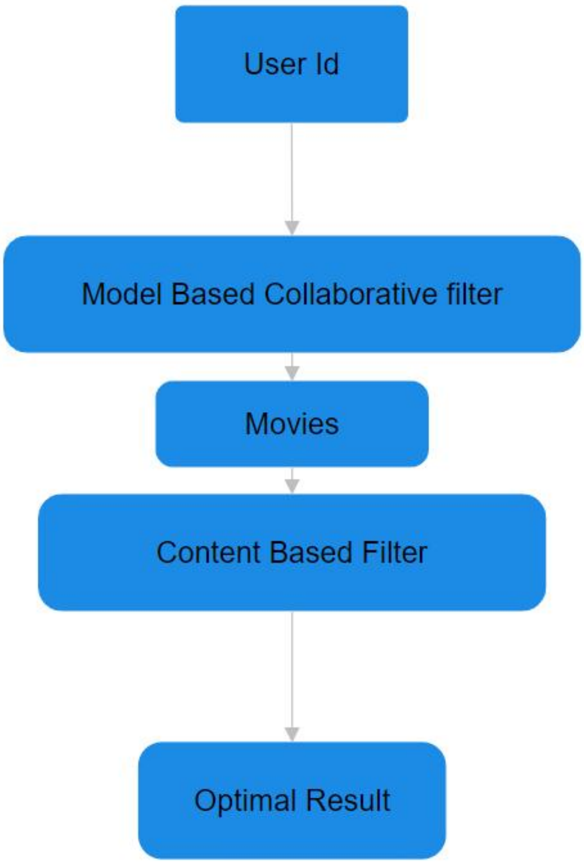
PROPOSED METHODOLOGY

3.1 Methods

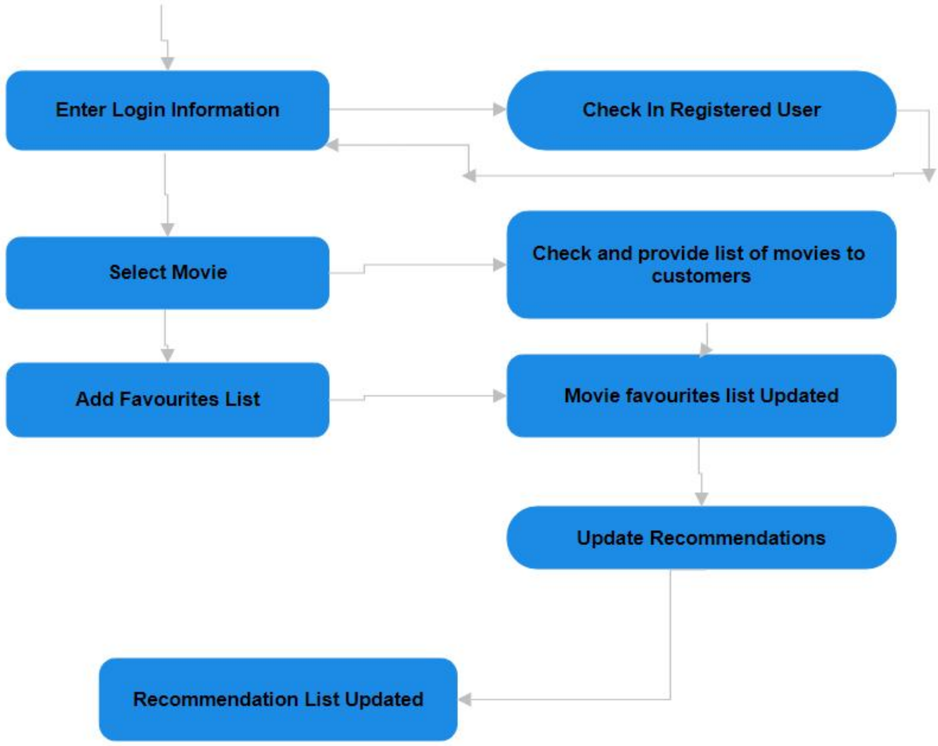
In this proposed methodology for our Movie Recommendation Project, we leverage the cosine similarity function and content-based filtering as foundational elements for our recommendation system. This approach is tailored to cater to the diverse preferences and interests of users while ensuring efficiency and accuracy in movie recommendations.

1. System Architecture of Proposed System:

Based on content based filtering approaches used in the project, each search will recommend a set of 5 movies to a particular user. When he user will search for a movie, top 5 most similar movies in terms of similar content will get recommended to the user

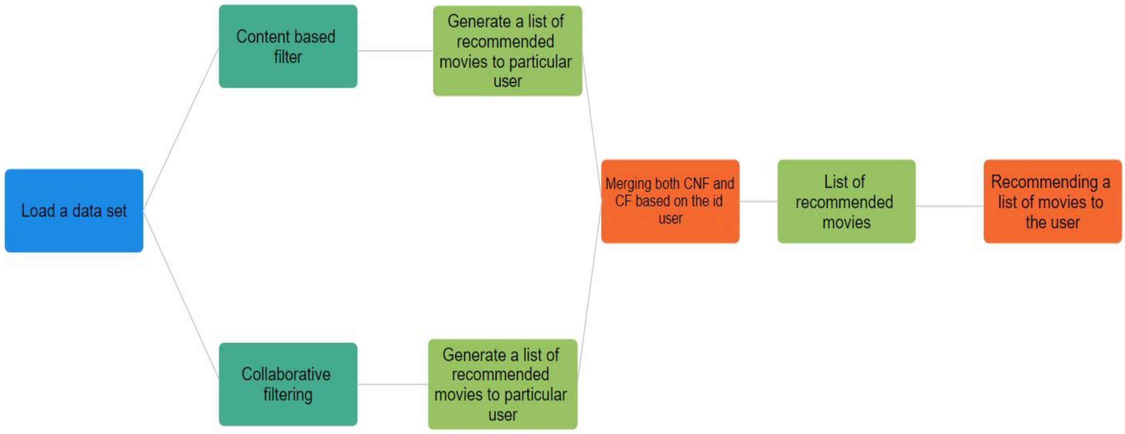


Activity Diagram



2. Dataflow

Initially load the data sets that are required to build a model the data set that are required in this project dataset.csv are available in the Kaggle.com. Preprocessing of the data set will take place and then by applying content-based filtering, users will get recommended top 8 most similar movies according to their search.



3. Data Collection and Preprocessing:

- We begin by acquiring a comprehensive dataset comprising movie attributes such as genre, plot keywords, actors, directors, and ratings.
- The dataset undergoes preprocessing to handle missing values, standardize formats, and remove noise, ensuring data quality and consistency.

4. Feature Extraction and Representation:

- Utilizing content-based filtering, relevant features are extracted from the dataset to characterize each movie.
- Features are represented in a feature vector space, where each movie is described by its attributes and characteristics.

5. Cosine Similarity Calculation:

- The cosine similarity function is employed to quantify the similarity between movies based on their feature vectors.

- By computing the cosine of the angle between two feature vectors, we determine the degree of similarity between movies, with higher cosine values indicating greater similarity.

6. User Profile Creation:

- Upon receiving user preferences or viewing history, a user profile is created based on the attributes of movies they have previously liked or rated highly.
- The user profile is represented as a feature vector in the same space as movie features, enabling seamless comparison and recommendation.

7. Recommendation Generation:

- To generate recommendations for a given user, we employ cosine similarity to measure the similarity between the user profile and each movie in the dataset.
- Movies with the highest cosine similarity scores to the user profile are recommended as potential matches, prioritizing those with the greatest resemblance to the user's preferences.

8. Evaluation and Optimization:

- The performance of the recommendation system is evaluated using metrics such as precision, recall, and accuracy.
- Optimization techniques may be applied to fine-tune parameters and enhance the effectiveness of the recommendation algorithm, ensuring optimal performance.

9. Deployment and Integration:

- The finalized recommendation system is integrated into a user-friendly interface or application, allowing users to access personalized movie recommendations effortlessly.
- Continuous monitoring and feedback mechanisms are implemented to iteratively improve the recommendation system based on user interactions and preferences.[15]

10. Technologies Used:

Python libraries:

For the computation and analysis we need certain python libraries which are used to perform analytics. Packages such as SKlearn, Numpy, pandas, Matplotlib, Flask framework, etc are needed.

SKlearn:

It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

NumPy:

NumPy is a general-purpose array-processing package. It provides a high performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. Pandas: Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools. Unlike NumPy library which provides objects for multi-dimensional arrays, Pandas provides in-memory 2d table object called Data frame.[16]

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Advantages and Disadvantages

Our movie recommendation system is based on content based filtering. Its few advantages and disadvantages are listed below.

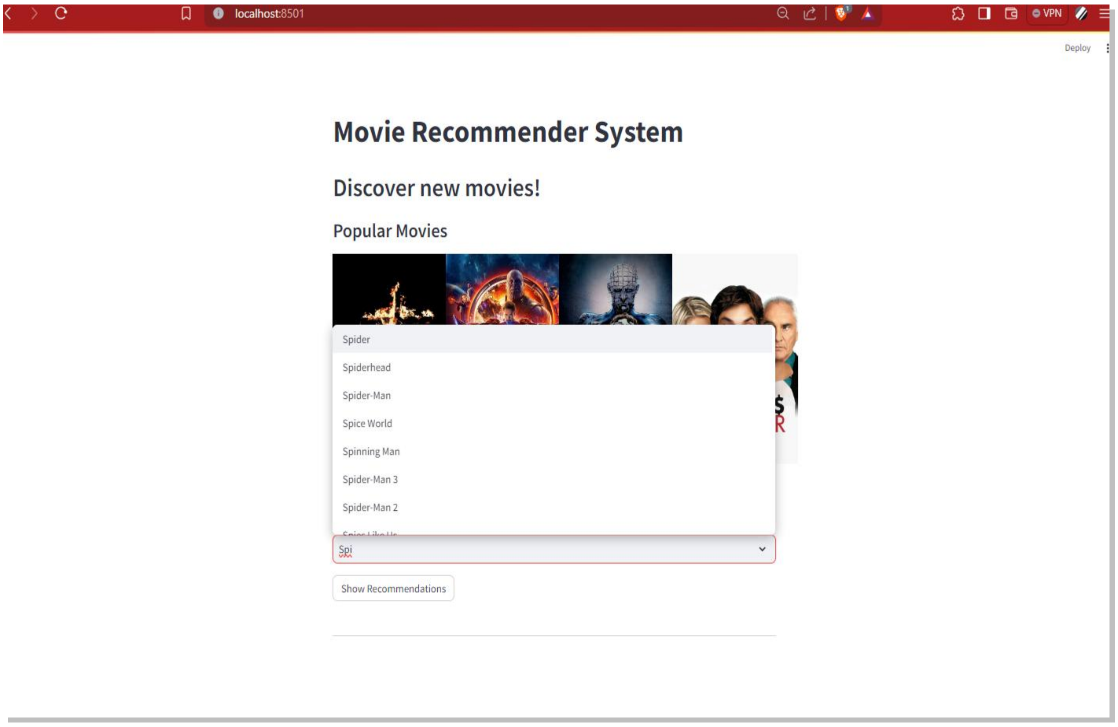
Advantages:

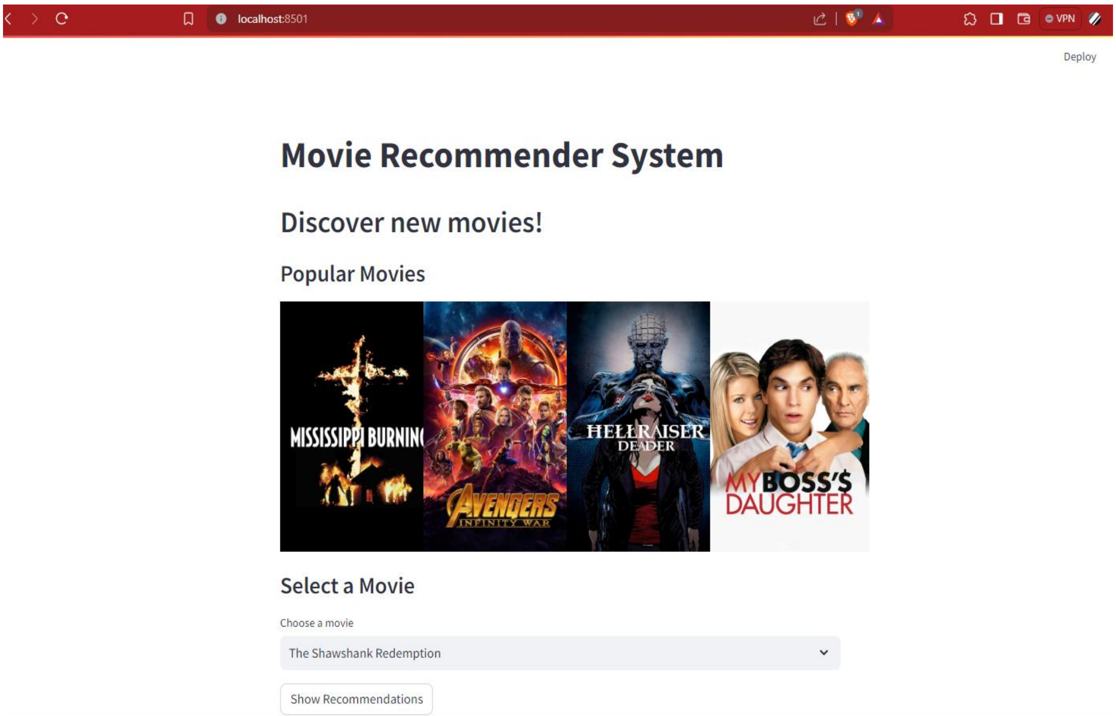
- **Automatic Feature Update:** The system updates its features and functions autonomously due to its unsupervised learning foundation.
- **Flexibility to New Input:** It readily adapts to new inputs, making it suitable for scenarios with unidentified new data such as unrated movies or users with no existing data.
- **Recommendation Capability for New Movies**: New movies can be recommended once the system extracts their features, facilitating their **inclusion in the recommendation process.**
- **Mitigation of Cold Start for New Users:** New users do not face a cold start issue as the system provides a starting point for them on the output feature map.
- **Efficiency in Computation:** The system's ability to organize complex data efficiently results in faster computation. It also creates a clear representation of the mapping, facilitating easy interpretation.

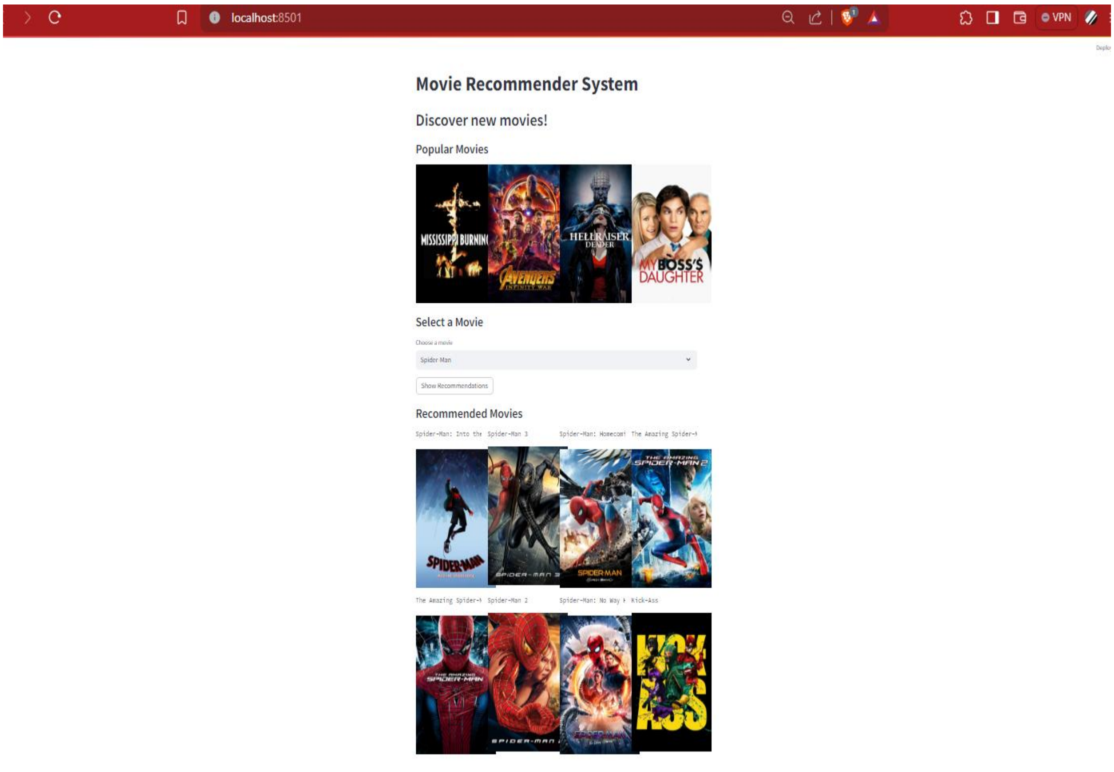
Disadvantages:

- **Issue with Feature Classification:** One major drawback arises when feature classification doesn't align with the expected output.
- **Frequent Initialization Requirement:** To address this, unsupervised learning classification algorithms need frequent initialization to ensure the relevance of the clustering process.[18]

4.2 Result


localhost:8501
Deploy
Movie Recommender System
Discover new movies!
Popular Movies
Spider
Spiderhead
Spider-Man
Spice World
Spinning Man
Spider-Man 3
Spider-Man 2
Spi
Show Recommendations


localhost:8501
Deploy
Movie Recommender System
Discover new movies!
Popular Movies
HELLRAISER
DEADER
AVENGERS
INFINITY WAR
MY BOSS'S
DAUGHTER
Select a Movie
Choose a movie
The Shawshank Redemption
Show Recommendations


localhost:8501
Movie Recommender System
Discover new movies!
Popular Movies
Select a Movie
Spider Man
Show Recommendations
Recommended Movies

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

In this project, we have made the movie recommender system which recommends movies based on the content that is the most similar to what the user searches on the site. We have used cosine similarity to find out which top 8 movies would be the closest to what the user searches. This system tries to save up the time for the users who want to watch a movie similar to watch what they had watched before. We have used movie datasets from Kaggle.com which contained information like genre, cast, movie title, keywords in the movie, language and any more to preprocess the data and filter and gather all the information that would to be required to find out all the relevant data for content-based filtering.[20]

5.2 Future scope

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.[21]

REFERENCES

[1] Hirdesh Shivhare, Anshul Gupta and Shalki Sharma, "Recommender system using fuzzy c-means clustering and genetic algorithm based weighted similarity measure" Communication and Control. IEEE International Conference on Computer, 2015.

[2] Manoj Kumar, D.K. Yadav, Ankur Singh and Vijay Kr. Gupta. "A Movie Recommender System: MOVREC, International Journal of Computer Applications (0975-8887) Volume 124-No.3, 2015.

[3] RyuRi Kim, Ye Jeong Kwak, HyeonJeong Mo, Mucheol Kim, Seungmin Rho,Ka Lok Man,Woon Kian Chong "Trustworthy Movie Recommender System with Correct Assessment and Emotion Evaluation". Proceedings of the International MultiConference of Engineers and Computer Scientists Vol II, 2015

[4] Zan Wang, Xue Yu*, Nan Feng, Zhenhua Wang, "An Improved Collaborative Movie Recommendation System using Computational Intelligence" Journal of Visual Languages & Computing. Volume 25, Issue 6, 2014

[5] Debadrita Roy, Arnab Kundu, "Design of Movie Recommendation System by Means of Collaborative Filtering". International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 4, 2013.

[6] E. Amolochitis, Algorithms and applications for academic search, recommendation and quantitative association rule mining. Gistrup, Denmark: River Publishers, 2018.

[7] T. Brodt, Collaborative Filtering. Saarbrücken: VDM, Müller, 2010.

[8] M. Jalali, H. Gholizadeh and S. Hashemi Golpayegani, "An improved hybrid recommender system based on collaborative filtering, content based, and demographic filtering", International Journal of Academic Research, vol. 6, no. 6, pp. 22-28, 2014.

[9] C. Li and K. He, "CBMR: An optimized MapReduce for item-based collaborative filtering recommendation algorithm with empirical analysis", Concurrency and Computation: Practice and Experience, vol. 29, no. 10, p. e4092, 2017.

[10] Smbandam Jayalakshmi, Narayanan Ganesh, Rober Cep and Jankiraman Senthi Murugan (2022) , "Movie Recommender Systems: Concepts, Methods, Challenges, and Future Directions"

[11] Alyari F., Navimipour N.J. Recommender systems: A systematic review of the state of the art literature and suggestions for future research. Kybernetes. 2018;47:985. doi: 10.1108/K-06-2017-0196.

[12] Caro-Martinez M., Jimenez-Diaz G., Recio-Garcia J.A. A theoretical model of explanations in recommender systems; Proceedings of the ICCBR; Stockholm, Sweden. 9–12 July 2018.

[13] Gupta S. A Literature Review on Recommendation Systems. Int. Res. J. Eng. Technol. 2020;7:3600–3605.

[14] Abdulla G.M., Borar S. Size recommendation system for fashion e-commerce; Proceedings of the KDD Workshop on Machine Learning Meets Fashion; Halifax, NS, Canada. 14 August 2017.

[15] Aggarwal C.C. Recommender Systems. Springer; Berlin/Heidelberg, Germany: 2016. An Introduction to Recommender Systems; pp. 1–28.

[16] Ghazanfar M.A., Prugel-Bennett A. A scalable, accurate hybrid recommender system; Proceedings of the 2010 Third International Conference on Knowledge Discovery and Data Mining; Washington, DC, USA. 9–10 January 2010.

[17] Deldjoo Y., Elahi M., Cremonesi P., Garzotto F., Piazzolla P., Quadrana M. Content-Based Video Recommendation System Based on Stylistic Visual Features. J. Data Semant. 2016;5:99–113. doi: 10.1007/s13740-016-0060-9.

[18] Cami B.R., Hassanpour H., Mashayekhi H. A content-based movie recommender system based on temporal user preferences; Proceedings of the 2017 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS); Shahrood, Iran. 20–21 December 2017.

[19] Beniwal R., Debnath K., Jha D., Singh M. Data Analytics and Management. Springer; Berlin/Heidelberg, Germany: 2021. Hybrid Recommender System Using Artificial Bee Colony Based on Graph Database; pp. 687–699.

[20] Çano E., Morisio M. Hybrid recommender systems: A systematic literature review. Intell. Data Anal. 2017;21:1487–1524. doi: 10.3233/IDA-163209.

[21] Schafer J.B., Konstan J.A., Riedl J. E-commerce recommendation applications. Data Min. Knowl. Discov. 2001;5:115–153. doi: 10.1023/A:1009804230409

[22] Shen J., Zhou T., Chen L. Collaborative filtering-based recommendation system for big data. Int. J. Comput. Sci. Eng. 2020;21:219–225. doi: 10.1504/IJCSE.2020.105727.

[23] Dakhel G.M., Mahdavi M. A new collaborative filtering algorithm using K-means clustering and neighbors' voting; Proceedings of the 11th International Conference on Hybrid Intelligent Systems (HIS); Malacca, Malaysia. 5–8 December 2011; pp. 179–184

[24] Colomo-Palacios R., García-Peñalvo F.J., Stantchev V., Misra S. Towards a social and context-aware mobile recommendation system for tourism. Pervasive Mob. Comput. 2017;38:505–515. doi: 10.1016/j.pmcj.2016.03.001.

[25] Casillo M., Conte D., Lombardi M., Santaniello D., Valentino C. Recommender System for Digital Storytelling: A Novel Approach to Enhance Cultural Heritage; Proceedings of the Pattern Recognition, ICPR International Workshops and Challenges, ICPR 2021; Virtual. 10–15 January 2021; pp. 304–317.

[26] Baltrunas L., Ludwig B., Peer S., Ricci F. Context-Aware Places of Interest Recommendations for Mobile Users; Proceedings of the International Conference of Design, User Experience, and Usability; Orlando, FL, USA. 9–14 July 2011; pp. 531–540.

[27] Casillo M., De Santo M., Lombardi M., Mosca R., Santaniello D., Valentino C. Recommender Systems and Digital Storytelling to Enhance Tourism Experience in Cultural Heritage Sites; Proceedings of the IEEE International Conference on Smart Computing (SMARTCOMP); Irvine, CA, USA. 23–27 August 2021; pp. 323–328.

[28] Markos A.I., Vozalis M.G., Margaritis K.G. An Optimal Scaling Approach to Collaborative Filtering Using Categorical Principal Component Analysis and Neighborhood Formation; Proceedings of the IFIP International Conference on Artificial Intelligence Applications and Innovations; Hersonissos, Greece. 25–27 June 2010; pp. 22–29.

APPENDIX 1

1. Data Representation:

- Each movie within our dataset is meticulously represented as a feature vector, encapsulating various attributes including genre, cast, director, plot keywords, and metadata.
- Below is a snippet illustrating the structured representation of movie feature vectors:

	id	title	tags
0	278	The Shawshank Redemption	Framed in the 1940s for the double murder of h...
1	19404	Dilwale Dulhania Le Jayenge	Raj is a rich, carefree, happy-go-lucky second...
2	238	The Godfather	Spanning the years 1945 to 1955, a chronicle o...
3	424	Schindler's List	The true story of how businessman Oskar Schind...
4	240	The Godfather: Part II	In the continuing saga of the Corleone crime f...
...	...	...	...
9995	10196	The Last Airbender	The story follows the adventures of Aang, a yo...
9996	331446	Sharknado 3: Oh Hell No!	The sharks take bite out of the East Coast whe...
9997	13995	Captain America	During World War II, a brave, patriotic Americ...
9998	2312	In the Name of the King: A Dungeon Siege Tale	A man named Farmer sets out to rescue his kidn...
9999	455957	Domino	Seeking justice for his partner's murder by an...

10000 rows × 3 columns

2. Cosine Similarity Calculation:

- Core to our content-based filtering strategy is the computation of cosine similarity, a critical metric quantifying the likeness between two movie feature vectors.
- A visual depiction of cosine similarity scores between a reference movie and others in the dataset is provided below, offering insight into the distribution of similarities

- Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them.

Formula:

$$Cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \, \|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2} \sqrt{\sum_1^n b_i^2}}$$

where, $\vec{a} \cdot \vec{b} = \sum_1^n a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$ is the dot product of the two vectors.

3. Recommendation Generation:

- The recommendation process initiates with computing cosine similarity between the user's profile vector and feature vectors of all movies in the dataset.
- A sample output showcasing recommended movies for a hypothetical user, accompanied by their respective cosine similarity scores, is outlined below:

Movie Title	Cosine Similarity Score
Inception	0.85
The Dark Knight	0.78
Interstellar	0.72

4. Implementation Considerations:

- Data preprocessing techniques, including normalization and TF-IDF weighting, are meticulously applied to handle disparate feature scales and enhance recommendation quality.
- Feature selection emphasizes capturing salient aspects influencing user preferences, thereby reinforcing the efficacy of our content-based filtering.

- A feature selection heatmap, illustrating the significance of different movie attributes in recommendation generation, is presented below:

5. Evaluation Metrics:

- The efficacy of our recommendation system is assessed using various evaluation metrics such as precision, recall, and F1-score, ensuring robustness and performance validation.